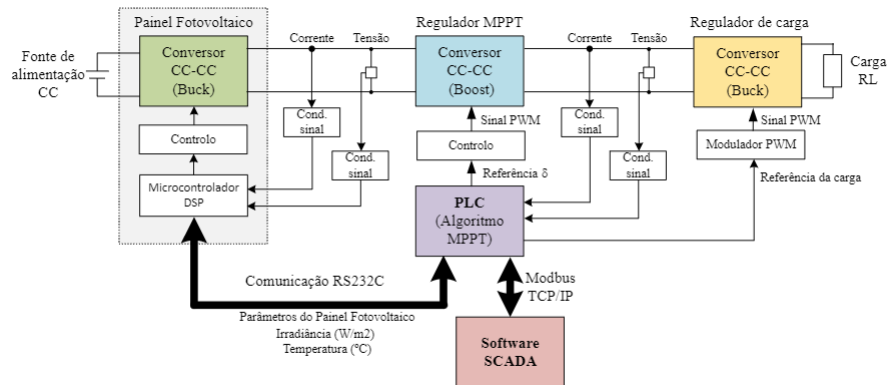


**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**  
**Departamento de Engenharia Eletrotécnica de Energia e Automação**



## Simulador dinâmico para painéis fotovoltaicos

HÉLIO ALEXANDRE VELOSO LOPES  
(Licenciado em Engenharia Eletrotécnica)

Dissertação de natureza científica para obtenção do grau de Mestre  
em Engenharia Eletrotécnica – Ramo de Automação e Eletrónica Industrial

Orientador:

Prof. Doutor Armando José Leitão Cordeiro

Júri:

Presidente: Prof. Doutor João Hermínio Ninitas Lagarto

Vogais:

Prof. Doutor José Gabriel da Silva Lopes

Prof. Doutor Armando José Leitão Cordeiro

Maio de 2023



Dissertação realizada sob orientação de

**Doutor Armando José Leitão Cordeiro**

Professor Coordenador do Departamento de  
Engenharia Eletrotécnica Energia e Automação

**Instituto Superior de Engenharia de Lisboa**



## Resumo

Esta dissertação apresenta o trabalho desenvolvido para a colocação em funcionamento de um protótipo de simulador de sistemas de produção de energia solar fotovoltaica. A solução baseia-se num conjunto de conversores estáticos de potência apoiados por um microcontrolador, por um autómato programável e por um *software* de parametrização e acompanhamento dos resultados. Foram ainda desenvolvidos diversos circuitos que complementar as funcionalidades do trabalho apresentado.

Ao longo do documento são abordados vários temas de relevância: a temática das energias renováveis numa perspetiva atual, com enfoque no potencial da energia solar; as soluções existentes e em estudo para o aproveitamento da energia solar; a caracterização de funcionamento de um painel fotovoltaico, tendo em consideração o ambiente dinâmico em que este naturalmente se insere, assim como vários algoritmos de maximização da potência aplicados em conversores estáticos de potência.

São também apresentados conceitos das áreas de eletrónica de potência e de automação aplicados na construção, dimensionamento e colocação em funcionamento do simulador, onde são abordadas as diversas topologias que constituem o simulador, a forma como se interligam e o seu princípio de funcionamento. Para comprovar o desempenho da solução proposta apresentam-se diversas simulações e resultados experimentais obtidos com o protótipo desenvolvido. Os testes e resultados obtidos com esta solução revelam um bom desempenho com limitações pontuais em alguns aspetos, os quais se descrevem ao longo dos respetivos capítulos. A solução serviu essencialmente para aprofundar os conhecimentos em diversas áreas da engenharia e ao mesmo tempo afigura-se como uma possível ferramenta pedagógica que permita reforçar o ensino integrado de energias renováveis, eletrónica de potência e automação.

**Palavras-chave:** Simulador de Painel Fotovoltaico; Controlo por modo deslizante; Conversor CC-CC Redutor-Elevador (Buck-Boost); Controlador Lógico Programável; Sistema SCADA.



# Abstract

This dissertation presents the work developed for the commissioning of a simulator prototype of photovoltaic solar energy production systems. The solution is based on a set of static power converters supported by a microcontroller, a programmable logic controller and software for parameterization and monitoring of results. Several circuits were also developed to complement the functionalities of the presented work.

Throughout the document, several relevant topics are addressed: the theme of renewable energies in a current perspective, with a focus on the potential of solar energy; existing solutions and those under study for harnessing solar energy; the characterization of the operation of a photovoltaic panel, taking into account the dynamic environment in which it is naturally inserted, as well as several power maximization algorithms applied in static power converters.

Concepts from the areas of power electronics and automation applied in the construction, design and commissioning of the simulator are also presented, where the various topologies that make up the simulator are addressed, the way in which they are interconnected and their operating principle. To prove the performance of the proposed solution, several simulations and experimental results obtained with the developed prototype are presented. The tests and results obtained from this solution reveal a good performance with occasional limitations in some aspects, which are described throughout the respective chapters. The solution essentially served to deepen knowledge in different areas of engineering and at the same time appears as a possible pedagogical tool that allows reinforcing the integrated teaching of renewable energies, power electronics and automation.

**Keywords:** PV simulator; Sliding mode controller; Buck-Boost DC-DC converter; Programmable Logic Controller; SCADA system.



## **Agradecimentos**

Em primeiro lugar, gostaria de expressar o meu mais sincero agradecimento aos meus familiares, pelo seu apoio constante ao longo desta jornada. Sem eles, não teria sido possível concluir este trabalho.

Gostaria de agradecer ao meu orientador, Professor Doutor Armando José Leitão Cordeiro, pela orientação valiosa e confiança depositada em mim. O seu conhecimento e experiência foram fundamentais para a realização desta tese.

Quero agradecer também aos meus colegas e amigos que sempre me apoiaram durante este processo.

Gostaria de deixar também um agradecimento ao Instituto Superior de Engenharia de Lisboa, pelas oportunidades e recursos disponibilizados para a realização desta tese, assim como os conhecimentos que me foram passados ao longo de todo o meu percurso académico nesta instituição.

Este trabalho foi apoiado pelo Instituto Politécnico de Lisboa, código de referência: IPL/2021/ATS2SPV\_ISEL. Agradeço à Phoenix Contact Portugal no âmbito da parceria EDUNET pelo fornecimento de dispositivos de automação para este trabalho.

Agradeço a todos por terem contribuído para que eu pudesse concluir este trabalho.



# Índice

1. Introdução .....	1
1.1. Enquadramento e Motivação .....	2
1.2. Objetivos.....	3
1.3. Estrutura da dissertação .....	3
2. Estado da arte.....	5
2.1. Considerações sobre energias renováveis.....	5
2.1.1. Enquadramento atual.....	5
2.1.2. Principais fontes de energia renovável.....	7
2.2. Recurso Solar.....	8
2.2.1. Disponibilidade deste recurso .....	8
2.2.2. Geometria Sol-Terra.....	9
2.2.3. Fatores meteorológicos .....	10
2.3. Energia Solar Fotovoltaica .....	11
2.3.1. Modelação da célula fotovoltaica.....	11
2.3.2. Curvas características I-V e P-V .....	21
2.3.3. Variação da irradiância solar incidente e temperatura .....	23
2.3.4. Painel Fotovoltaico.....	24
2.3.5. Associação de painéis fotovoltaicos.....	26
2.3.6. Tipos de sistemas fotovoltaicos.....	28
2.3.7. Simuladores de sistemas fotovoltaicos.....	32
2.4. Conversores Eletrónicos de Potência .....	33
2.4.1. Topologias de conversores CC-CC adotados em sistemas solares fotovoltaicos .....	34
2.4.2. Controlo de conversores CC-CC adotados em sistemas solares fotovoltaicos .....	41

2.4.3.	Os Algoritmos <i>MPPT</i> .....	42
2.5.	Sistemas Embebidos .....	49
2.5.1.	Microcontroladores .....	49
2.5.2.	Comunicações .....	51
2.6.	Controladores industriais e supervisão .....	55
2.6.1.	Controladores Lógicos Programáveis ( <i>PLC</i> ) .....	55
2.6.2.	Norma IEC 61131-3 .....	57
2.6.3.	Sistemas de supervisão ( <i>SCADA</i> ).....	58
3.	Trabalho desenvolvido.....	60
3.1.	Arquitetura proposta .....	60
3.2.	Painel Fotovoltaico a simular .....	63
3.3.	Simulação do sistema proposto .....	64
3.4.	Conversores CC-CC .....	72
3.4.1.	Dimensionamento do Conversor CC-CC <i>Buck</i> .....	74
3.4.2.	Dimensionamento do Conversor CC-CC <i>Boost</i> .....	76
3.4.3.	Desenvolvimento do circuito de potência e disparo do conversor....	79
3.5.	Controlo do Painel Fotovoltaico.....	81
3.6.	Transdutores .....	83
3.6.1.	Transdutor de tensão .....	83
3.6.2.	Transdutor de corrente .....	84
3.6.3.	Filtro passa-baixo .....	85
3.7.	Microcontrolador dsPIC30f4011 .....	87
3.7.1.	Ligações físicas .....	87
3.7.2.	Programação.....	88
3.7.3.	Velocidade de funcionamento e comunicação .....	89
3.7.4.	Aplicação do controlo em tensão .....	91
3.8.	PLC Phoenix Contact ILC 131 .....	93
3.8.1.	Ligações físicas .....	93

3.8.2.	Aplicação como controlador MPPT .....	94
3.8.3.	Aplicação como regulador da carga .....	95
3.8.4.	Comunicação <i>RS-232-C</i> .....	96
3.9.	Interface homem-máquina ( <i>HMI</i> ).....	96
4.	Resultados.....	99
4.1.	Modo I (Painel <i>SPV</i> ).....	99
4.1.1.	Curvas características para diferentes valores de irradiância incidente e temperatura .....	103
4.2.	Modo II (Algoritmo <i>MPPT</i> ) .....	106
5.	Conclusões .....	110
5.1.	Trabalho futuro .....	111
6.	Referências.....	113
7.	Anexos .....	121

## Lista de figuras

Figura 2.1 – Crescimento da Energia Renovável face à energia total consumida entre 2009 e 2019 [3].	6
Figura 2.2 – Percentagem de Energia Renovável usada no consumo total de energia, dividido pelo tipo de energia final, em 2018 [3].	6
Figura 2.3 – Incremento anual de capacidade de produção de Energia Elétrica Renovável, por tecnologia, de 2014 a 2020 [3].	7
Figura 2.4 – Energia Solar Fotovoltaica, capacidade global e crescimento anual, 2010-2020 [3].	8
Figura 2.5 – Potencial fotovoltaico no mundo [7].	9
Figura 2.6 – Rotação da Terra em torno do Sol ao longo do ano [10].	10
Figura 2.7 – Junção p-n de uma célula fotovoltaica.	12
Figura 2.8 – Circuito elétrico equivalente do modelo ideal de uma célula solar.	13
Figura 2.9 – Formação da característica I-V.	13
Figura 2.10 – Circuito elétrico equivalente do modelo de 4 parâmetros de uma célula solar fotovoltaica.	17
Figura 2.11 – Circuito elétrico equivalente do modelo de 5 parâmetros de uma célula solar fotovoltaica.	19
Figura 2.12 – Circuito elétrico equivalente de dois díodos de uma célula solar fotovoltaica.	21
Figura 2.13 – Curvas características I-V e P-V de um gerador solar fotovoltaico.	22
Figura 2.14 – Modos de funcionamento de uma célula fotovoltaica.	23
Figura 2.15 – Exemplo de curvas características I-V e P-V com diferentes valores de irradiância solar incidente e temperatura STC ( $T=25^{\circ}\text{C}$ ) [16].	24
Figura 2.16 – Exemplo de curvas características I-V e P-V com diferentes valores de temperatura e irradiância solar incidente STC ( $G=1000\text{W/m}^2$ ) [16].	24
Figura 2.17 – Módulo fotovoltaico constituído por 36 células ligadas em série [26].	25
Figura 2.18 – Hierarquia do agrupamento de células fotovoltaicas [26].	26
Figura 2.19 – Associação em série de painéis fotovoltaicos [19].	27
Figura 2.20 – Associação em paralelo de painéis fotovoltaicos [17].	27
Figura 2.21 – Associação em série e paralelo de painéis fotovoltaicos [17].	28

Figura 2.22 – Esquema simplificado de um sistema solar fotovoltaico ligado à rede elétrica. ....	30
Figura 2.23 – Esquemas simplificados de um sistema isolado da rede elétrica: (a) com recurso a um carregador unidirecional; (b) com recurso a um carregador bidirecional. ....	31
Figura 2.24 – Layout de um sistema híbrido. ....	32
Figura 2.25 – Conversores CC-CC unidirecionais utilizados para carregar baterias em sistemas solares fotovoltaicos: (a) conversor redutor ou <i>Buck</i> ; (b) conversor elevador ou <i>Boost</i> . ....	36
Figura 2.26 – Conversores CC-CC redutores-elevadores ( <i>Buck-Boost</i> ) unidirecionais utilizados para carregar baterias em sistemas solares fotovoltaicos: (a) conversor clássico; (b) exemplo de nova topologia com extensão do ganho de tensão [56]. ....	36
Figura 2.27 – Conversor CC-CC redutor-elevador ( <i>Buck-Boost</i> ) bidirecional utilizado para carregar e descarregar baterias em sistemas isolados da rede ou sistemas híbridos [57]. ....	37
Figura 2.28 – Modos de funcionamento do conversor CC-CC redutor ( <i>Buck</i> ) unidirecional: (a) – topologia; Modo I – Interruptor S1 ligado; Modo II – Interruptor S1 desligado. ....	38
Figura 2.29 – Forma de onda típicas de um conversor CC-CC <i>Buck</i> unidirecional [23]. ....	39
Figura 2.30 – Modos de funcionamento do conversor CC-CC elevador ( <i>Boost</i> ) unidirecional: (a) – topologia; Modo I – Interruptor S1 ligado; Modo II – Interruptor S1 desligado. ....	40
Figura 2.31 – Forma de onda típicas de um conversor CC-CC <i>Boost</i> unidirecional [23]. ....	41
Figura 2.32 – Diagrama de fluxo do método P&O. ....	45
Figura 2.33 – Regiões possíveis do ponto de operação de um gerador fotovoltaico, relativas ao PMP. ....	47
Figura 2.34 – Diagrama de fluxo do método da Condutância Incremental. ....	48
Figura 2.35 – Componentes de um microcontrolador. ....	49
Figura 2.36 – Microchip dsPIC30F4011. ....	50
Figura 2.37 – Características principais dos microcontroladores da gama dsPIC30F. ....	50
Figura 2.38 – Dispositivo PICkit 3. ....	51

Figura 2.39 – Exemplo de mensagem série em RS-232 (em cima) e TTL (em baixo).....	52
Figura 2.40 – Conversor USB – UART. ....	52
Figura 2.41 – Conversor UART (TTL) - RS232.....	53
Figura 2.42 – Esquema de ligações de uma comunicação <i>SPI</i> .....	53
Figura 2.43 – DAC de 12-bits “MCP4922”. ....	54
Figura 2.44 – Composição típica de um <i>PLC</i> .....	55
Figura 2.45 – Modelo básico de comunicações industriais. ....	56
Figura 2.46 – <i>Layout</i> do <i>PLC</i> ILC 131 ETH da Phoenix Contact.....	57
Figura 2.47 – Exemplo de configuração de um sistema SCADA. ....	59
Figura 3.1 – Diagrama de blocos do modo I (modo painel <i>SPV</i> ).....	61
Figura 3.2 – Diagrama de blocos do modo II (modo de teste <i>MPPT</i> ).....	62
Figura 3.3 – Dados do Painel Fotovoltaico ASE-100-DG-UR recolhidos do <i>software PVsyst</i> [73].....	64
Figura 3.4 – Esquema da simulação da solução idealizada do painel <i>SPV</i> na ferramenta <i>SIMULINK</i> . ....	64
Figura 3.5 – Grandezas elétricas da simulação da solução idealizada do painel <i>SPV</i> na ferramenta <i>SIMULINK</i> .....	65
Figura 3.6 – Variação da irradiância incidente da simulação da solução idealizada do painel <i>SPV</i> na ferramenta <i>SIMULINK</i> . ....	66
Figura 3.7 – Variação da temperatura da simulação da solução idealizada do painel <i>SPV</i> na ferramenta <i>SIMULINK</i> .....	66
Figura 3.8 – Esquema da simulação do painel <i>SPV</i> na ferramenta <i>SIMULINK</i> .....	67
Figura 3.9 – Grandezas elétricas da simulação do painel <i>SPV</i> na ferramenta <i>SIMULINK</i> .....	68
Figura 3.10 – Variação da irradiância incidente da simulação do painel <i>SPV</i> na ferramenta <i>SIMULINK</i> . ....	69
Figura 3.11 – Variação da temperatura da simulação do painel <i>SPV</i> na ferramenta <i>SIMULINK</i> .....	69
Figura 3.12 – Curvas características I-V (esquerda) e P-V (direita) para diferentes valores de irradiância incidente. ....	70
Figura 3.13 – Curvas características I-V (esquerda) e P-V (direita) para diferentes valores de temperatura.....	70
Figura 3.14 – Esquema da simulação do modo <i>MPPT</i> na ferramenta <i>SIMULINK</i> . ....	71

Figura 3.15 – Variação de carga da simulação do modo <i>MPPT</i> na ferramenta <i>SIMULINK</i> .....	71
Figura 3.16 – Topologia base do Conversor CC-CC Redutor-Elevador ( <i>Buck-Boost</i> ).....	72
Figura 3.17 – Esquema de funcionamento do Conversor CC-CC como Elevador ( <i>Boost</i> ): (a) com o semicondutor $T_2$ à condução; (b) com o semicondutor $T_2$ ao corte. 73	73
Figura 3.18 – Esquema de funcionamento do Conversor CC-CC como Redutor ( <i>Buck</i> ): (c) com o semicondutor $T_1$ à condução; (b) com o semicondutor $T_1$ ao corte. . 73	73
Figura 3.19 – Esquema elétrico do circuito de potência do conversor CC-CC construído. ....	79
Figura 3.20 – Esquema elétrico do circuito de disparo do conversor CC-CC construído. ....	79
Figura 3.21 – Placa de circuito impresso construída. ....	80
Figura 3.22 – Esquema de aquisição de tensão do transdutor de tensão. ....	83
Figura 3.23 – Esquema de tratamento e envio de sinal do transdutor de tensão. ....	84
Figura 3.24 – PCB do transdutor de tensão. ....	84
Figura 3.25 – Transdutor de corrente LEM LA25-NP. ....	84
Figura 3.26 – Esquema de ligações do LEM LA25-NP. ....	85
Figura 3.27 – Ligações para uma escala máxima de 5A no LEM LA25-NP. ....	85
Figura 3.28 – Ligações do microcontrolador <i>dsPIC30f4011</i> . ....	87
Figura 3.29 – Projeto na linguagem C construído na ferramenta MPLAB X. ....	89
Figura 3.30 – Configuração da velocidade de <i>clock</i> e velocidade de comunicação série do microcontrolador. ....	90
Figura 3.31 – Equações do controlo em tensão aplicado pelo microcontrolador. ....	91
Figura 3.32 – Configuração física do PLC Phoenix Contact ILC 131 ETH. ....	93
Figura 3.33 – Ligações à carta de entradas analógicas. ....	94
Figura 3.34 – Ligações à carta de saídas analógicas. ....	94
Figura 3.35 – Cabo para comunicação RS-232 através da porta DIN-8 do PLC. ....	94
Figura 3.36 – Sequência de instruções do algoritmo <i>MPPT</i> aplicado pelo <i>PLC</i> . ...	95
Figura 3.37 – Bloco de configuração da comunicação <i>RS232</i> do <i>PLC</i> . ....	96
Figura 3.38 – Página principal do <i>HMI</i> implementado. ....	97
Figura 3.39 – Página de leitura e edição de parâmetros do <i>HMI</i> implementado. ....	98
Figura 4.1 – Software SCADA no modo I, conversor CC-CC regulador da carga com duty-cycle de 0,7. ....	99
Figura 4.2 – Tensão e corrente enviadas por <i>RS232C</i> para o <i>PC</i> . ....	100

Figura 4.3 – Formas de onda dos transdutores de tensão (CH1) e corrente (CH2). .....	100
Figura 4.4 – Formas de onda da mensagem enviada para o <i>DAC</i> , pins 24 (SCK1, CH1) e 25 (SDO1, CH2) do microcontrolador.....	101
Figura 4.5 – Formas de onda do comparador, CH1: sinal <i>PWM</i> gerado como resultado da comparação, CH2: referência de tensão gerada pelo <i>DAC</i> , CH3: onda triangular com frequência 60 kHz. ....	101
Figura 4.6 – Formas de onda, CH1: sinal <i>PWM</i> de comutação do <i>MOSFET</i> , CH4: corrente na bobine (100mV/A). ....	102
Figura 4.7 – Formas de onda do comparador responsável pela geração do sinal <i>PWM</i> do conversor CC-CC <i>Buck</i> regulador da carga, CH1: sinal <i>PWM</i> gerado, CH2: referência de tensão do <i>PLC</i> , CH3: onda triangular com frequência de 100 kHz.....	102
Figura 4.8 – Formas de onda da comunicação <i>RS232C</i> entre o microcontrolador e o <i>PLC</i> registadas nos pins do microcontrolador, esquerda: pin 27 ( <i>UART2 Tx</i> ), direita: pin 28 ( <i>UART2 Rx</i> ). ....	103
Figura 4.9 – Curvas características I-V para diferentes valores de irradiância incidente com pontos experimentais. ....	104
Figura 4.10 – Curvas características P-V para diferentes valores de irradiância incidente com pontos experimentais. ....	104
Figura 4.11 – Curvas características I-V para diferentes valores de temperatura com pontos experimentais. ....	105
Figura 4.12 – Curvas características P-V para diferentes valores de temperatura com pontos experimentais. ....	105
Figura 4.13 – Formas de onda do modulador <i>PWM</i> do conversor <i>Boost</i> , CH1: sinal <i>PWM</i> , CH2: tensão de referência do algoritmo <i>MPPT</i> em funcionamento no <i>PLC</i> , CH3: onda triangular de 100 kHz. ....	106
Figura 4.14 – Formas de onda, CH1: sinal <i>PWM</i> de comutação do <i>MOSFET</i> do conversor <i>Boost</i> , CH4: corrente na bobine (100mV/A). ....	107
Figura 4.15 – Pontos experimentais da curva I-V do algoritmo <i>MPPT</i> ensaiado.	108
Figura 4.16 - Pontos experimentais da curva P-V do algoritmo <i>MPPT</i> ensaiado.	108
Figura 4.17 – Sistema <i>SCADA</i> no ponto de máxima potência. ....	109

## **Lista de tabelas**

Tabela 2.1 – Metodologia do método <i>P&amp;O</i> . .....	46
Tabela 3.1 – Características do Painel Fotovoltaico ASE-100-DG-UR [72]. .....	63
Tabela 3.2 – Descrição das ligações do microcontrolador <i>dsPIC30f4011</i> . .....	87

## Lista de acrónimos

ADC	–	Conversor Analógico-Digital
AM	–	Massa de Ar ( <i>Air Mass</i> )
CA	–	Corrente Alternada
CC	–	Corrente Contínua
CC-CC	–	Corrente Contínua – Corrente Contínua
CPU	–	<i>Central Processing Unit</i>
DAC	–	Conversor Digital-Analógico
FF	–	Fator de forma ( <i>Fill factor</i> )
IGBT	–	<i>Insulated Gate Bipolar Transistor</i>
MISO	–	<i>Master In Slave Out</i>
MOSFET	–	<i>Metal–oxide–semiconductor Field-effect Transistor</i>
MOSI	–	<i>Master Out Slave In</i>
MPPT	–	<i>Maximum Power Point Tracking</i>
P&O	–	<i>Perturb and Observe</i>
PLC	–	<i>Programmable Logic Controller</i>
PLL	–	<i>Phase-locked Loop</i>
PMP	–	Ponto de máxima potência
ppm	–	Parte por milhão
PWM	–	<i>Pulse-width modulation</i>
Rx	–	<i>Receive signal</i>
SCADA	–	<i>Supervisory Control And Data Acquisition</i>
SPI	–	<i>Serial Peripheral Interface</i>
SPV	–	Solar Fotovoltaico
SS	–	<i>Slave Select</i>
STC	–	<i>Standard Test Conditions</i>
TTL	–	<i>Transistor-transistor Logic</i>
Tx	–	<i>Transmit signal</i>
UART	–	<i>Universal Asynchronous Receiver-Transmitter</i>
USB	–	<i>Universal Serial Bus</i>

## Lista de variáveis

$A$ [ $m^2$ ]	–	Área
$C_p$	–	Passo da perturbação
$D$	–	Fator de ciclo ( <i>Duty cycle</i> )
$G$ [ $W/m^2$ ]	–	Irradiância incidente
$I_{cc}$ [A]	–	Corrente de curto-circuito
$I_d$ [A]	–	Corrente que atravessa o diodo
$I_{mp}$ [A]	–	Corrente no ponto de máxima potência
$I_o$ [A]	–	Corrente de saturação do diodo
$I_s$ [A]	–	Corrente gerada pelo efeito fotovoltaico na célula solar
$m$	–	Fator de idealidade
$N_p$	–	Número de células solares em paralelo
$N_s$	–	Número de células solares em série
$P_{dc}$ [W]	–	Potência Contínua
$P_p$ [W]	–	Potência de pico
$R_p$ [ $\Omega$ ]	–	Resistência paralelo
$R_s$ [ $\Omega$ ]	–	Resistência série
$SYB$ [ $^\circ$ ]	–	Declinação solar
$T$ [K]	–	Temperatura
$u_{Icc}$ [ $A/^\circ C$ ]	–	Coefficiente de variação da corrente de curto-circuito
$u_{Vca}$ [ $V/^\circ C$ ]	–	Coefficiente de variação da tensão em circuito aberto
$V_{ca}$ [V]	–	Tensão de circuito aberto
$V_{mp}$ [V]	–	Tensão no ponto de máxima potência
$V_t$ [V]	–	Tensão térmica do diodo



# 1. Introdução

Ao longo dos últimos anos têm vindo a acentuar-se de forma generalizada as preocupações com as alterações climáticas decorrentes da quantidade excessiva de dióxido de carbono na atmosfera, perturbando o efeito de gases de estufa com consequente impacto no aquecimento global [1]. Para mitigar este problema global, vários países e organizações estão promovendo atualmente uma suave transição para fontes de energia limpas ou com pegada ecológica reduzida. Essa transição pode ser constatada em diferentes setores, desde a produção industrial até diversos serviços públicos e privados, passando pelos transportes e construção de edifícios. É interessante perceber que as mudanças climáticas são muito provavelmente consequência dos avanços tecnológicos alcançados ao longo do último século (acompanhado de um acelerado crescimento da população mundial), e agora, parecem ser necessários novos avanços tecnológicos para superar os problemas gerados por eles mesmos. De acordo com alguns relatórios mais relevantes [2] [3], o uso de fontes de energia renováveis, especialmente através de geradores eólicos e painéis solares fotovoltaicos (*SPV*) tem vindo a crescer em todo o mundo e cada vez mais países apostam na eletrificação das suas infraestruturas públicas e privadas com recurso a energias renováveis. Os investimentos em fontes de energia renováveis batem ano após ano novos recordes por diversos motivos. Por um lado, impulsionados pelos menores custos de produção como consequência da produção massiva e pelo crescimento de novos mercados apoiados por políticas governamentais mais amigas do ambiente. Por outro lado, devido também à elevada variação dos preços da energia em consequência de diversas instabilidades geopolíticas mundiais acompanhadas de elevadas especulações. Todos estes motivos têm contribuindo para que muitas empresas e indivíduos se tornem produtores-consumidores (*prosumers*) com recurso a alguma forma de energia renovável. De acordo com o relatório da Agência Internacional de Energia [2], a energia *SPV* e eólica contribuíram com cerca de dois terços do crescimento das energias renováveis em 2021. A República Popular da China sozinha alcançou quase metade do aumento global de energia renovável em 2021, seguida pelos Estados Unidos, União Europeia e Índia. Ao longo de 2021, a geração de energia *SPV* subiu quase 145 TWh, ou quase 18%, aproximando-se de 1000 TWh.

Existem, no entanto, sempre novos aspectos a serem investigados e aperfeiçoados no projeto de componentes e matérias-primas para sistemas renováveis, com especial enfoque nos solares fotovoltaicos. Para impulsionar continuamente novos desenvolvimentos e novas instalações *SPV* adicionais, é também necessário compreender princípios de funcionamento, estratégias de controle, instalação/interligação, eficiência energética e aspectos de manutenção de modo a integrá-las de forma mais acelerada nas mais diversas aplicações. Neste sentido, para incentivar o desenvolvimento e instalação de painéis *SPV*, é também necessário criar novas ferramentas destinadas à formação de técnicos e jovens engenheiros. Foi neste sentido que se pretendeu conduzir este trabalho final de mestrado, através da criação de uma solução que permitisse simular o princípio de funcionamento dos sistemas *SPV* e dos algoritmos associados aos seus conversores estáticos de potência para maximização da potência produzida. Este trabalho é também importante no sentido em que existem geralmente algumas restrições em relação ao teste de painéis *SPV*. Alguns deles estão relacionados com a variabilidade das condições de operação (já que a irradiação e a temperatura não são constantes ao longo do dia) e a posição relativa entre o sol e os painéis solares muda constantemente [4]. Para fins educacionais, os testes de campo de painéis *SPV* geralmente são bastante caros e dependem fortemente das condições climáticas. Por esta razão, diversas ferramentas e métodos de baixo custo desenvolvidos para simular o comportamento dos painéis têm vindo a ser propostos na literatura.

## **1.1. Enquadramento e Motivação**

A área da automação industrial é fundamental nos dias que correm. A sua importância está associada à industrialização de processos e aliada à automatização de tarefas, permitindo assim reduzir a mão-de-obra humana, reduzir custos de produção, melhorar a eficácia na sua produção e até mesmo a eficiência de processos.

Atualmente a automação industrial desempenha um papel preponderante em diversos contextos e, como tal, é natural que esta área funcione como agregadora de diversas outras áreas técnicas afins à Engenharia Eletrotécnica como é o caso da eletrónica, mecânica ou informática, tornando assim possível desenvolverem-se de forma combinada sistemas mais ou menos complexos que simplificam determinado processo ou resolvem determinado problema que dificilmente seria possível sem estes recursos.

Uma finalidade menos comum da automação industrial é a utilização de um sistema como o proposto neste trabalho final de mestrado para efeitos de simulação do seu comportamento elétrico, onde para além da eletrónica de potência e de regulação e comando necessários se integram aspetos de automatização dos ensaios com recurso a soluções e equipamentos usados na automação industrial. A motivação para a realização deste trabalho é bastante evidente dada a elevada importância das energias renováveis e em particular do contributo da energia solar fotovoltaica num contexto global de descarbonização, mostrando que é possível criar soluções e dotá-las de mecanismos automáticos de ensaio e validação usando ferramentas usadas na automação industrial combinadas com outras áreas do conhecimento.

## **1.2. Objetivos**

O objetivo principal deste Trabalho Final de Mestrado é o estudo e a construção de um simulador de um sistema de produção de energia fotovoltaica integrando mecanismos automáticos de realização de ensaios com recurso a soluções industriais, tanto quanto possível, criando uma ferramenta testes do funcionamento destes sistemas de forma conjugada e com componentes de baixo custo.

É ainda objetivo deste trabalho usar esta ferramenta como meio didático para demonstração em laboratório dos princípios de funcionamento e comportamento dos sistemas *SPV* de modo a cimentar os conhecimentos dos alunos de mestrado em Engenharia Eletrotécnica numa área tão relevante como as energias renováveis aliadas aos conhecimentos adquiridos em eletrónica de potência, eletrónica de regulação e comando, aquisição e processamento de sinais e automação industrial.

Foi também objetivo dotar o autor deste trabalho de conhecimentos adicionais em diversas áreas onde pudesses existir algumas lacunas de modo a preparar-se para um mercado de trabalho cada vez mais exigente e abrangente.

Por forma a avaliar o seu comportamento, serão apresentados diversos ensaios realizados, para simular e experimentar situações distintas, quer no funcionamento do painel fotovoltaico, quer no funcionamento de algoritmos de maximização da potência fornecida pelo simulador *SPV* mediante diversas condições atmosféricas impostas.

## **1.3. Estrutura da dissertação**

O presente documento está organizado em cinco capítulos. O primeiro capítulo apresenta o enquadramento, motivação e objetivos do projeto, assim como a estrutura

do documento. O segundo capítulo enquadra a importância atual da produção de Energia Solar Fotovoltaica, aborda diversos aspetos relativos ao funcionamento dos painéis fotovoltaicos, sistemas fotovoltaicos, assim como aspetos relativos às outras áreas científicas aplicadas na construção do simulador. O terceiro capítulo descreve o desenvolvimento do simulador, abordando em maior detalhe os componentes aplicados, assim como a metodologia aplicada na integração do sistema desenvolvido. O quarto capítulo apresenta os resultados experimentais obtidos nos ensaios efetuados em laboratório, a sua avaliação e análise. Finalmente o quinto capítulo é dedicado às conclusões finais, juntamente com algumas propostas de trabalho futuro a desenvolver.

De forma a facilitar a leitura e compreensão deste documento foram adotadas as seguintes convenções:

- Sempre que possível, recorreu-se a conceitos presentes na Língua Portuguesa, mas em determinados casos, para uma compreensão mais fácil do seu significado, estes conceitos são acompanhados pela designação em Língua Inglesa, colocada entre parêntesis e itálico;
- As siglas de equipamentos, conceitos e termos serão utilizadas na Língua Inglesa por forma a garantir um maior contacto com as siglas utilizadas;
- A numeração de figuras e tabelas foi efetuada de forma sequencial ao longo de cada capítulo, referenciadas por dois números separados por um ponto, sendo que o primeiro número refere-se ao capítulo e o segundo à sequência ordenada dentro do capítulo;
- As referências bibliográficas foram ordenadas pela ordem que surgem ao longo da dissertação. A citação das referências bibliográficas ao longo dos diversos capítulos foi realizada de acordo com as normas IEEE, da forma que normalmente aparece em publicações científicas.

## **2. Estado da arte**

Neste capítulo abordam-se aspetos relacionados com o estado da arte relativo ao panorama atual das energias renováveis, caracterização dos recursos solares mundiais, tipificação da energia solar, modelos dos painéis fotovoltaicos e suas curvas características. Apresentam-se as aplicações e instalações dos painéis solares fotovoltaicos em diferentes vertentes e alguns tipos de conversores adotados quer para ajustar, baixar ou subir a tensão de modo a compatibilizar os sistemas fotovoltaicos com a carga bem como soluções de conversores que permitam extrair a máxima potência da conversão solar fotovoltaica de acordo com alguns dos algoritmos mais conhecidos e usados. Dado que estes últimos assuntos são de natureza muito vasta e amplamente estudados ao longo dos últimos anos, apresentam-se essencialmente algumas das topologias que estão diretamente ou indiretamente relacionadas com o trabalho desenvolvido neste Trabalho Final de Mestrado.

### **2.1. Considerações sobre energias renováveis**

#### **2.1.1. Enquadramento atual**

A utilização de energia na sociedade em termos globais está em constante crescimento, de forma a garantir a crescente melhoria na qualidade de vida dos países desenvolvidos e a suportar o crescimento populacional nos países em desenvolvimento. O tema da exploração dos recursos naturais usados para produção de energia, é considerado um dos temas mais importantes da era moderna. A produção de energia com base em combustíveis fósseis é reconhecida atualmente como a causa de diversos problemas da sociedade, tais como: a depleção do ozono, alterações climáticas, problemas ambientais e riscos acrescidos para a saúde pública. A quantidade de CO<sub>2</sub> na atmosfera aumentou de 280 ppm (parte por milhão) na era pré-industrial para 409,8 ppm em 2019 [5].

Segundo o Global Status Report 21, verificou-se um aumento da necessidade global de energia de 320 ExaJoules em 2009 para 381 ExaJoules em 2019. Este aumento de necessidade energética foi garantido apenas em 25% por fontes de energia renováveis. Os restantes 75% foram garantidos por energia proveniente do consumo de combustíveis fósseis, tendo este ainda muito peso no conjunto energético global [3].

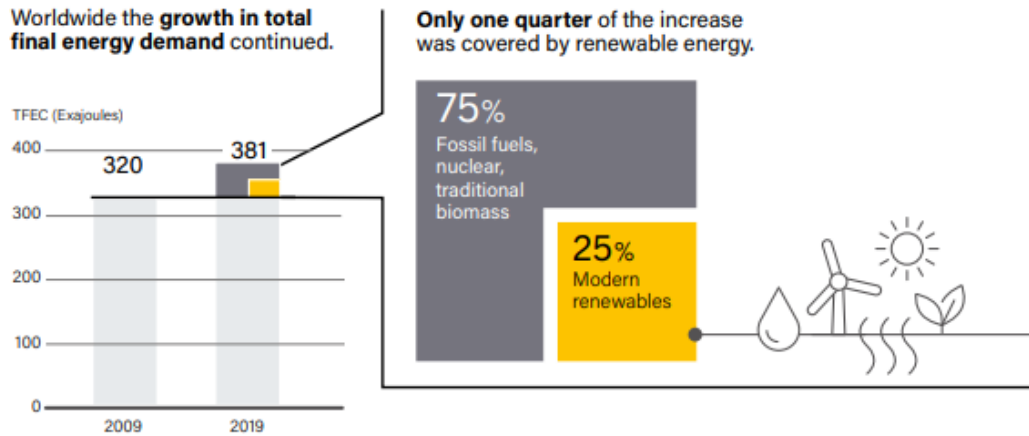


Figura 2.1 – Crescimento da Energia Renovável face à energia total consumida entre 2009 e 2019 [3].

Perante os efeitos adversos das tecnologias de produção de energia com base em combustíveis fósseis, existe atualmente uma forte tendência de investimento em tecnologias limpas, sem emissões de carbono. Esta tendência é impulsionada pelas políticas verdes e rápidos avanços científicos em determinadas tecnologias. A energia é utilizada em 3 formas principais: a energia térmica, na forma de calor; a energia relativa aos transportes; e a energia elétrica. Pela Figura 2.2, verifica-se que é no setor da produção de energia elétrica que se reflete o maior impacto da transição para fontes de energia renovável.

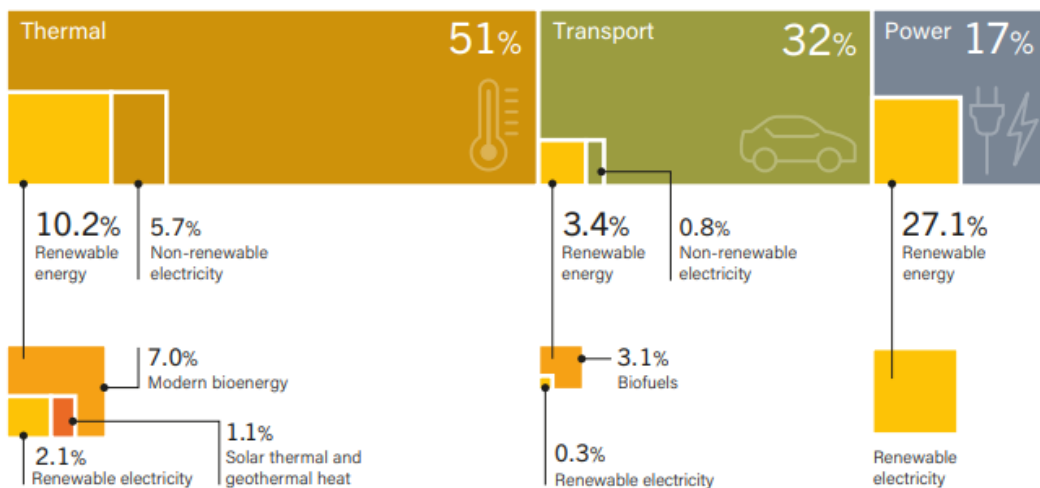


Figura 2.2 – Percentagem de Energia Renovável usada no consumo total de energia, dividido pelo tipo de energia final, em 2018 [3].

## 2.1.2. Principais fontes de energia renovável

No setor da produção de energia elétrica, a produção com recurso a fontes renováveis, estabeleceu, globalmente, um novo recorde anual de potência instalada em 2020. A capacidade de produção, face ao ano anterior cresceu cerca de 257 GW [3]. Continuando uma tendência desde 2012, a maioria da potência instalada em 2020 foi de origem renovável, representando cerca de 83% de toda a potência instalada. Verifica-se que, nos últimos anos, a energia solar fotovoltaica e a energia eólica são as tecnologias que mais contribuíram para o aumento da potência instalada.

A energia solar fotovoltaica contribuiu com cerca de 139 GW de potência instalada, mais de metade do valor total (ver Figura 2.3). Cerca de 93 GW devem-se à energia eólica e cerca de 20 GW à energia hidroelétrica. Os restantes acréscimos de capacidade representam valores marginais, e devem-se a outras tecnologias ainda em desenvolvimento ou que não apresentam uma fiabilidade comercial para um crescimento em massa, tais como a energia das marés, a energia geotérmica ou a energia solar térmica concentrada.

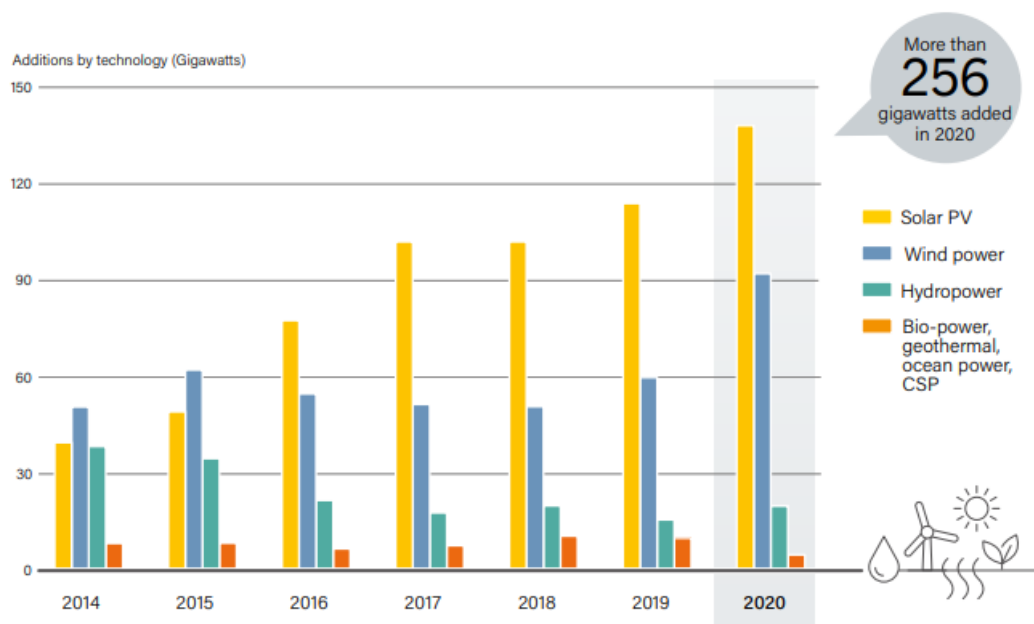


Figura 2.3 – Incremento anual de capacidade de produção de Energia Elétrica Renovável, por tecnologia, de 2014 a 2020 [3].

Tanto a energia solar fotovoltaica como a energia eólica estabeleceram novos recordes de potência total instalada em 2020. A primeira, com uns estimados 760 GW de potência global total instalada, como se pode verificar pela Figura 2.4, e a segunda chegou aos 743 GW instalados globalmente. Estes valores são notáveis, particularmente

para a energia solar fotovoltaica, dado que esta tecnologia ultrapassou pela primeira vez a energia eólica em termos de potência total instalada, sendo esta uma derradeira prova de um exponencial crescimento nos últimos anos, considerando que em 2010 existiam apenas 39 GW de potência total global instalada [3].

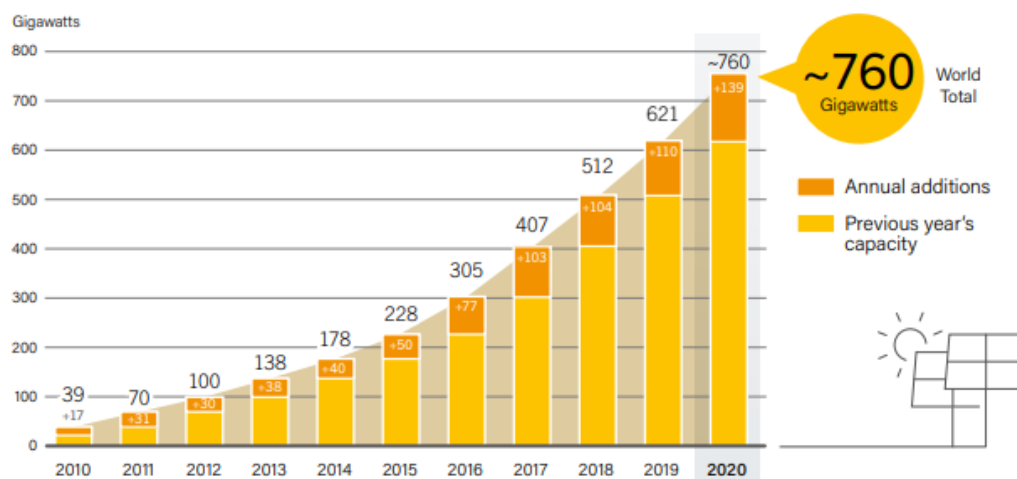


Figura 2.4 – Energia Solar Fotovoltaica, capacidade global e crescimento anual, 2010-2020 [3].

Pelo seu enorme potencial em captar a vasta energia do Sol que atinge a superfície terrestre e pelas suas possibilidades de desenvolvimento na melhoria de eficiência, a tecnologia solar fotovoltaica é um tema de elevada importância na atualidade. São exemplos, o estudo de formas de modelar e simular os sistemas fotovoltaicos, de forma a ter cada vez melhores previsões da sua produção em determinados locais, assim como de melhorias nos seus componentes, de forma a assegurar um aumento na eficiência da transformação de energia captada proveniente do Sol.

## 2.2. Recurso Solar

### 2.2.1. Disponibilidade deste recurso

A atmosfera terrestre recebe, anualmente, cerca de  $1,5 \times 10^{18}$  kWh de energia do Sol [6]. Observa-se na Figura 2.5 o mapa de potenciais recursos solares distribuídos mundialmente de acordo com [7]. A origem da maioria das fontes de energia existentes na Terra é proveniente do Sol. A geração de energia hidroelétrica, que acontece por via da evaporação, responsável pelo ciclo das águas e que possibilita o enchimento dos rios para a consequente geração de energia. A energia eólica é proveniente dos ventos e é a radiação solar que induz a circulação atmosférica em larga escala. Também a energia da biomassa (lenha, carvão vegetal, álcool, etc.) depende da radiação solar para o processo

de fotossíntese, em que é diretamente absorvida e armazenada nas ligações químicas de moléculas orgânicas que sustenta toda a cadeia alimentar do planeta [8].

A radiação solar, além de ser responsável pela manutenção da vida na Terra, também se constitui enquanto uma inesgotável fonte energética, em que há um enorme potencial de utilização por meio de sistemas de captação e conversão em outra forma de energia, são exemplos a energia térmica e a energia elétrica.

A energia elétrica pode ser obtida através de uma geração direta (energia solar fotovoltaica) e geração indireta (conversão de irradiação solar em calor para geração de energia elétrica). A energia térmica pode também ser obtida e aplicada diretamente (geração de calor para aquecer fluidos, bem como secagem e aquecimento industrial) [9]. No presente Trabalho Final de Mestrado será apenas abordada a energia solar fotovoltaica (*SPV*).

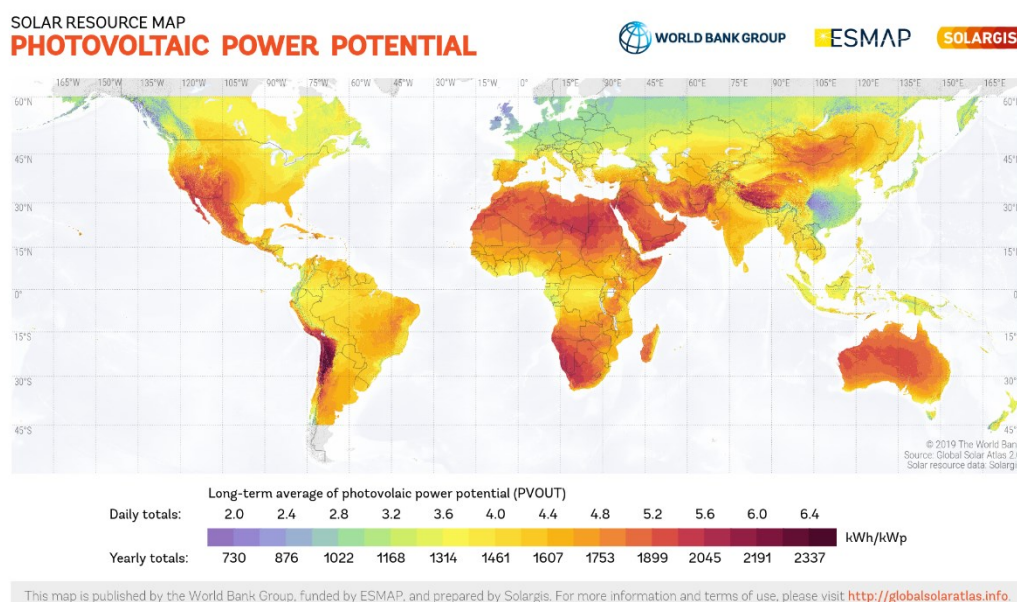


Figura 2.5 – Potencial fotovoltaico no mundo [7].

Apesar da enorme quantidade de radiação solar que incide sobre a atmosfera, verifica-se que o potencial fotovoltaico ao longo do ano é muito variável ao longo da superfície terrestre. Este potencial é essencialmente influenciado pela geometria Sol-Terra e por fatores meteorológicos [10].

### 2.2.2. Geometria Sol-Terra

Conhecer o ângulo de incidência dos raios solares é crucial para quantificar a energia proveniente do Sol. A Terra descreve em torno do Sol uma trajetória elíptica ao longo do ano, vulgarmente referido como movimento de translação (ver Figura 2.6). O

eixo rotacional da Terra, em relação ao plano normal da elipse apresenta uma inclinação de aproximadamente  $23,45^\circ$ . Esta inclinação e o movimento de translação são os responsáveis pela variação de radiação solar que incide sobre a superfície da Terra ao longo do ano [10].

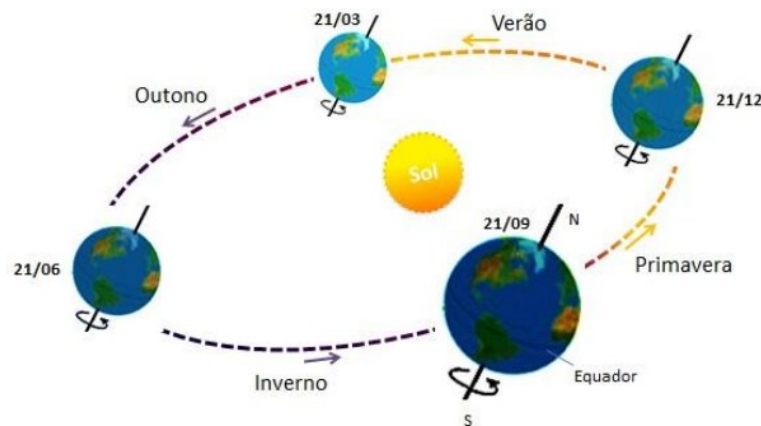


Figura 2.6 – Rotação da Terra em torno do Sol ao longo do ano [10].

Ao observar o movimento aparente do Sol, ao meio-dia solar e ao longo do ano, verifica-se que o ângulo entre os raios solares e o plano do Equador varia entre  $-23,45^\circ$  no solstício de inverno (21 de dezembro) e  $+23,45^\circ$  no solstício de verão (21 de junho). Este ângulo, denominado de declinação solar ( $SYB$ ) é positivo no hemisfério Norte e negativo no hemisfério Sul. Nos equinócios da primavera (21 de março) e outono (21 de setembro), os raios solares alinham-se com o plano do Equador ( $SYB=0$ ).

Considerando  $n$  o dia do calendário, de 1 a 365, a declinação solar pode ser calculada através de [11]:

$$\delta = 23,45 \operatorname{sen} \left[ 360 \times \frac{284 + n}{365} \right] [^\circ] \quad (2.1)$$

### 2.2.3. Fatores meteorológicos

A densidade média anual de irradiância solar, quando medida num plano perpendicular à direção da propagação dos raios solares no topo da atmosfera terrestre denomina-se de “constante solar” e corresponde a  $1.367 \text{ W/m}^2$  [8]. A irradiância solar que atinge a superfície terrestre e incide sobre uma superfície é constituída por uma componente direta, uma componente difusa e uma componente devido ao albedo (quando a superfície está inclinada). A soma das parcelas denomina-se por radiação global.

A radiação direta vem diretamente da direção do Sol e produz sombras nítidas. A radiação difusa é proveniente de todas as direções e atinge a superfície num valor condicionado pela atmosfera terrestre. O coeficiente de reflexão das superfícies envolventes (solo, vegetação, obstáculos, terrenos rochosos, etc.) denomina-se por albedo. Antes de atingir o solo, as características da radiação solar (intensidade, distribuição espectral e angular) são afetadas por interações com a atmosfera, devidas aos efeitos de absorção e espalhamento. Estas interações são dependentes da espessura da camada atmosférica, um coeficiente denominado Massa de Ar (AM, do inglês *Air Mass*), que depende do ângulo zenital do Sol, da distância Terra-Sol e das condições atmosféricas e meteorológicas [8]. A distribuição espectral padrão da radiação solar engloba a radiação solar direta e difusa incidente, ao nível do mar, num dia de céu limpo, numa superfície perpendicular aos raios solares e inclinada, relativamente à horizontal, de  $37^\circ$ , e a que corresponde uma irradiância total de  $1000 \text{ W/m}^2$  [12].

A corrente elétrica gerada por um painel fotovoltaico depende da irradiância solar incidente e da temperatura da célula. Como forma de uniformização das condições de medida dos parâmetros característicos das células, foram estabelecidas um conjunto de condições de referência (*Standard Test Conditions - STC*), nomeadamente:

- Temperatura da célula,  $\theta^r = 25^\circ\text{C} \equiv T^r = 298,16 \text{ K}$ ;
- Irradiância incidente na célula,  $G^r = 1000 \text{ W/m}^2$ ;
- Distribuição espectral padrão da radiação solar  $AM = 1,5$ .

As grandezas indicadas com o índice superior  $r$  consideram-se medidas nas condições de referência [12].

## **2.3. Energia Solar Fotovoltaica**

### **2.3.1. Modelação da célula fotovoltaica**

A célula fotovoltaica é o menor elemento de um painel fotovoltaico. A célula é constituída por materiais semicondutores que conseguem produzir energia elétrica quando incide a irradiância solar à sua superfície, ao que usualmente chamamos de efeito fotovoltaico. Para representar o efeito fotovoltaico provocado pelas células fotovoltaicas, recorre-se a modelos equivalentes de circuitos elétricos. A partir de um modelo deste tipo, sabendo algumas das suas propriedades elétricas, assim como as

condições atmosféricas, é possível modelar uma célula, módulo ou painel fotovoltaico. Para a modelação de, por exemplo, um determinado painel fotovoltaico, disponível comercialmente, é possível conhecer parte dos parâmetros necessários à sua modelação através das informações do fabricante. Para a obtenção dos restantes parâmetros é necessário um cálculo iterativo computadorizado ou a análise de dados reais [12].

A modelação é importante no estudo de aplicações fotovoltaicas, pois permite simular o comportamento de um determinado gerador fotovoltaico em condições normais, como em condições de temperatura e irradiância adversas. Os modelos aqui apresentados são os mais comuns da literatura deste tema. Quando aplicados com a melhor informação possível conseguem apresentar erros relativos na ordem dos 3% [13]. Apresentam-se seguidamente alguns dos modelos mais comuns encontrados na literatura da especialidade: o modelo ideal (ou modelo de 3 parâmetros), o modelo de 4 parâmetros, o modelo de 5 parâmetros e o modelo de 2 díodos.

### 2.3.1.1. Modelo ideal

A célula fotovoltaica comporta-se como uma fonte de corrente elétrica quando incide radiação proveniente do Sol na sua superfície. Nas referências bibliográficas [14], [15], [16], [17] podemos encontrar dados sobre os vários tipos de materiais usados na construção de células fotovoltaicas. O material mais usado é o silício, que na sua forma pura é estruturalmente estável ao nível atómico. Através da sua dopagem, é possível a criação de 2 camadas na célula: uma do tipo p com excesso de cargas positivas; outra do tipo n com excesso de cargas negativas. O resultado da sobreposição das 2 camadas forma uma junção p-n, com propriedades semicondutoras, conhecida do funcionamento dos díodos [12].

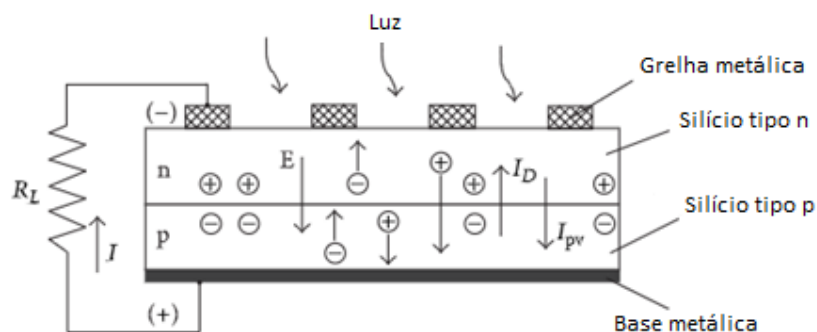


Figura 2.7 – Junção p-n de uma célula fotovoltaica.

O modelo ideal representa uma célula fotovoltaica como uma fonte de corrente  $I_S$  (dependente da irradiância incidente) em paralelo com um diodo pelas suas

características semicondutoras, tal como representado na Figura 2.8. **Erro! A origem da referência não foi encontrada..**

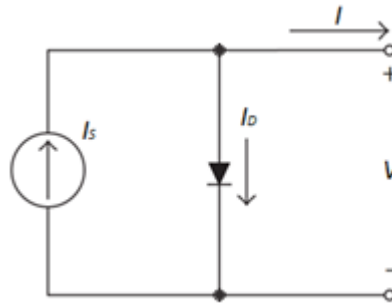


Figura 2.8 – Circuito elétrico equivalente do modelo ideal de uma célula solar.

Este modelo denomina-se como ideal pois assume o funcionamento de um díodo ideal e despreza perdas que se verificam num sistema real que, ao serem consideradas, aumentariam a complexidade do modelo. Com base na equação do díodo, o modelo ideal tem a seguinte característica de saída:

$$I = I_S - I_D = I_S - I_0(e^{V/mV_t} - 1) \text{ [A]} \quad (2.2)$$



Figura 2.9 – Formação da característica I-V.

A corrente  $I_S$  representa a corrente elétrica gerada através do efeito fotovoltaico, constante na característica I-V da célula proporcionalmente à irradiação incidente. A corrente  $I_0$  representa a corrente de saturação do díodo,  $m$  é o fator de idealidade do díodo (depende das suas características físicas) e a tensão  $V_t$  a tensão térmica do díodo, dada por:

$$V_t = \frac{kT}{q} \text{ [V]} \quad (2.3)$$

A tensão térmica do díodo é obtida em função de  $k$  que representa a constante de Boltzmann ( $1,381 \times 10^{-23}$  J/K),  $q$  que representa o valor de carga de um eletrão

$(1,602 \times 10^{-19} \text{ C})$  e  $T$  é a temperatura da célula em graus Kelvin, assumindo que esta é igual à temperatura da junção p-n.

Os três parâmetros a determinar para poder aplicar este modelo são a corrente  $I_S$ , a corrente de saturação do diodo  $I_o$  e o fator de idealidade  $m$ , sendo isto possível através de uma metodologia algébrica [12]. Para tal, é necessário observar alguns pontos de operação particulares:

1. Curto-circuito ( $V = 0; I = I_{cc}$ )
2. Circuito aberto ( $V = V_{ca}; I = 0$ )
3. Ponto de máxima potência ( $V = V_{mp}; I = I_{mp}$ )

### 1. Curto-circuito ( $V = 0; I = I_{cc}$ )

Substituindo os termos da equação para uma situação de curto-circuito, deduz-se que a corrente de saída será igual à corrente de curto-circuito  $I_{cc}$  da célula:

$$I_{cc} = I_S - I_o(e^{0/mV_t} - 1) [A] \quad (2.4)$$

Simplificando (2.4), vem:

$$I_{cc} = I_S [A] \quad (2.5)$$

A corrente de curto-circuito é o valor máximo da corrente de carga, ou seja, igual à corrente gerada por efeito fotovoltaico. O seu valor é uma característica da célula, dependente das condições de irradiância solar incidente e da temperatura.

### 2. Circuito aberto ( $V = V_{ca}; I = 0$ )

Substituindo os termos da equação para uma situação de circuito aberto, com o intuito de determinar a tensão de saída:

$$0 = I_S - I_o(e^{V_{ca}/mV_t} - 1) [A] \quad (2.6)$$

Resolvendo (2.6) em ordem a  $V_{ca}$ :

$$V_{ca} = mV_T \ln\left(1 + \frac{I_S}{I_o}\right) [V] \quad (2.7)$$

A tensão de circuito aberto é o valor máximo da tensão aos terminais da célula. Tal como a corrente de curto-circuito, o seu valor é uma característica da célula, também dependente das condições da irradiância incidente e da temperatura.

### 3. Potência máxima ( $V = V_{mp}$ ; $I = I_{mp}$ )

Para uma situação de potência máxima em determinadas condições de irradiância solar e temperatura:

$$I_{mp} = I_S - I_o(e^{V_{mp}/mV_t} - 1) \quad [A] \quad (2.8)$$

É também importante referir outros conceitos relacionados com o gerador fotovoltaico a modelar. O ponto de máxima potência para quaisquer condições de irradiância solar e temperatura é dado por:

$$P_{mp} = V_{mp}I_{mp} \quad [W] \quad (2.9)$$

Em condições de referência ou *STC*, no ponto de máxima potência, o valor de potência designa-se por potência-pico  $P_p$ :

$$P_p = P_{DC}^r = V_{mp}^r I_{mp}^r \quad [W] \quad (2.10)$$

O rendimento nas condições de referência é determinado pela relação entre a potência-pico e a potência correspondente à irradiância solar incidente na área da célula denominada por  $A$ :

$$\eta^r = \frac{P_p}{AG^r} \quad (2.11)$$

Em outras condições de funcionamento, o rendimento, com irradiância solar incidente  $G$ , é dado por:

$$\eta = \frac{P_{DC}}{AG} \quad (2.12)$$

O fator de forma (*Fill Factor*) é outro parâmetro relevante e é dado pela relação entre a potência-pico  $P_p$  e o produto  $V_{ca}^r I_{cc}^r$ . Para células do mesmo tipo, os valores de  $V_{ca}^r$  e  $I_{cc}^r$  são aproximadamente constantes, mas a forma da curva I-V pode variar consideravelmente. Naturalmente, pretende-se que o fator de forma seja o maior possível, mas este é uma consequência das características da célula.

$$FF = \frac{P_{DC}^r}{V_{ca}^r I_{cc}^r} \quad (2.13)$$

A fim de definir as equações que determinam os três parâmetros necessários para este modelo, sabe-se que para as condições *STC*, os valores de  $I_{cc}^r$ ,  $V_{ca}^r$ ,  $I_{mp}^r$ ,  $V_{mp}^r$  e  $P_{mp}^r$  são conhecidos para as células comercializadas, sendo dados pelo fabricante. Logo, reescrevendo as equações dos pontos de operação descritos anteriormente para as condições *STC*, obtemos:

$$I_{cc}^r = I_S^r [A] \quad (2.14)$$

$$0 = I_S^r - I_o^r (e^{V_{ca}^r/mV_t^r} - 1) [A] \quad (2.15)$$

$$I_{mp}^r = I_S^r - I_o^r (e^{V_{mp}^r/mV_t^r} - 1) [A] \quad (2.16)$$

Ao resolver a equação (2.15) em ordem a  $I_o^r$  e considerando a igualdade de (2.14) obtemos a equação (2.17):

$$I_o^r = \frac{I_{cc}^r}{e^{V_{ca}^r/mV_t^r} - 1} [A] \quad (2.17)$$

Considerando que nas equações (2.15) e (2.16), os termos seguintes são muito superiores à unidade então por simplificação podemos eliminar o termo mais pequeno:

$$e^{V_{ca}^r/mV_t^r} \gg 1 \quad (2.18)$$

$$e^{V_{mp}^r/mV_t^r} \gg 1 \quad (2.19)$$

Finalmente, relacionando as equações (2.15) e (2.16) após estas simplificações é possível obter o fator de idealidade  $m$  do díodo:

$$m = \frac{V_{mp}^r - V_{ca}^r}{V_t^r \ln\left(1 - \frac{I_{mp}^r}{I_{cc}^r}\right)} \quad (2.20)$$

Este modelo, pela sua simplicidade, permite representar uma célula solar de forma relativamente eficaz e simples, porém, ao desprezar os efeitos resistivos, as características I-V e P-V não são as mais corretas, particularmente em condições de baixa irradiância solar [18].

### 2.3.1.2. Modelo de 4 parâmetros

O modelo de 4 parâmetros complementa o modelo ideal ao incluir uma resistência série  $R_s$  ao circuito equivalente como indicado na Figura 2.10. A resistência série (idealmente com valor nulo) representa quedas de tensão devidas á resistência do material semiconductor, dos contactos metálicos e dos contactos metal-semiconductor [19]. O maior impacto acontece quando a célula solar apresenta valores de tensão mais elevados, ou seja, na zona em que a célula solar se comporta como uma fonte de tensão.

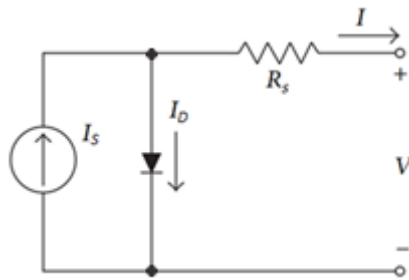


Figura 2.10 – Circuito elétrico equivalente do modelo de 4 parâmetros de uma célula solar fotovoltaica.

A característica de saída deste modelo é dada por:

$$I = I_s - I_o \left( e^{\frac{V+R_s I}{mV_T}} - 1 \right) [A] \quad (2.21)$$

Em condições *STC*, a equação anterior representa-se da seguinte forma:

$$I = I_s^r - I_o^r \left( e^{\frac{V+R_s I}{mV_T^r}} - 1 \right) [A] \quad (2.22)$$

Os quatro parâmetros a determinar neste modelo são a corrente  $I_s$ , a corrente de saturação do díodo  $I_o$ , o fator de idealidade  $m$  e a resistência série  $R_s$ . Alguns pontos de operação importantes, tal como no modelo anterior, podem ser descritos da seguinte forma em condições *STC*:

#### 1. Curto-circuito ( $V = 0$ ; $I = I_{cc}^r$ )

$$I_{cc}^r = I_s^r - I_o^r \left( e^{\frac{R_s I_{cc}^r}{mV_T^r}} - 1 \right) [A] \quad (2.23)$$

#### 2. Circuito aberto ( $V = V_{ca}^r$ ; $I = 0$ )

$$0 = I_S^r - I_o^r (e^{\frac{V_{ca}^r}{mV_t^r}} - 1) [A] \quad (2.24)$$

### 3. Ponto de máxima potência ( $V = V_{mp}^r$ ; $I = I_{mp}^r$ )

$$I_{mp}^r = I_S^r - I_o^r (e^{\frac{V_{mp}^r + R_s I_{mp}^r}{mV_t^r}} - 1) [A] \quad (2.25)$$

Por forma simplificar a obtenção dos parâmetros, é uma prática comum na literatura a aproximação da corrente fotovoltaica à corrente de curto-circuito ( $I_S \approx I_{cc}$ ), ao considerar que a diferença entre ambas é geralmente muito pequena [20]. Desta forma obtemos o primeiro parâmetro, com a equação (2.23) a tornar-se em:

$$I_S^r = I_{cc}^r [A] \quad (2.26)$$

Nestas condições, a corrente de saturação do díodo  $I_o$  pode ser obtida através das equações (2.24) e (2.26) [21]:

$$I_o = I_{cc}^r e^{-\frac{V_{ca}^r}{mV_t^r}} [A] \quad (2.27)$$

Partindo da equação anterior, a equação (2.22) pode ser rescrita:

$$I = I_{cc}^r (1 - e^{\frac{V - V_{ca}^r + R_s I}{mV_t^r}}) [A] \quad (2.28)$$

A equação do ponto de máxima potência torna-se assim:

$$I_{mp}^r = I_{cc}^r (1 - e^{\frac{V_{mp}^r - V_{ca}^r + R_s I_{mp}^r}{mV_t^r}}) [A] \quad (2.29)$$

É assim possível deduzir o valor da resistência série  $R_s$ :

$$R_s = \frac{mV_t^r \ln \left( 1 - \frac{I_{mp}^r}{I_{cc}^r} \right) + V_{ca}^r - V_{mp}^r}{I_{mp}^r} [\Omega] \quad (2.30)$$

O último parâmetro a obter é assim o fator de idealidade  $m$ , que pode ser obtido em consideração ao facto de a derivada de potência em ordem à tensão ser nula no ponto de máxima potência e através da equação (2.22) [21]:

$$m = \frac{2V_{mp}^r - V_{ca}^r}{mV_t^r \left[ \frac{I_{mp}^r}{I_{cc}^r - I_{mp}^r} + \ln \left( 1 - \frac{I_{mp}^r}{I_{cc}^r} \right) \right]} \quad (2.31)$$

Este modelo apresenta um bom compromisso entre complexidade e precisão, sendo o aplicado no desenvolvimento desta dissertação.

### 2.3.1.3. Modelo de 5 parâmetros

O modelo de 5 parâmetros, também conhecido como modelo real, complementa os modelos anteriores ao incluir uma resistência paralelo/shunt  $R_p$  ao circuito equivalente, tal como ilustrado na Figura 2.11. A resistência paralelo/shunt na célula solar representa as correntes de fuga quando o diodo está inversamente polarizado e pode derivar de defeitos construtivos no dispositivo fotovoltaico. Se esta for baixa, resulta em perdas de potência ao formar um caminho de corrente alternativo ao da corrente gerada. Assim, diminui a corrente que flui na junção p-n, o que reduz a tensão da célula fotovoltaica. O impacto deste elemento é mais acentuado quando a célula apresenta tensões menores, ou seja, na zona em que a célula fotovoltaica se comporta como uma fonte de corrente [22].

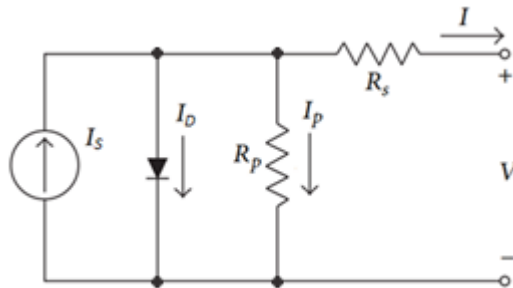


Figura 2.11 – Circuito elétrico equivalente do modelo de 5 parâmetros de uma célula solar fotovoltaica.

A característica de saída deste modelo é dada por:

$$I = I_s - I_D - I_p = I_s - I_o \left( e^{\frac{V+R_s I}{mV_t}} - 1 \right) - \frac{V + R_s I}{R_p} \quad [A] \quad (2.32)$$

Em condições *STC*, a equação representa-se da seguinte forma:

$$I = I_s^r - I_o^r \left( e^{\frac{V+R_s I}{mV_t^r}} - 1 \right) - \frac{V + R_s I}{R_p} \quad [A] \quad (2.33)$$

Os cinco parâmetros a determinar neste modelo são a corrente  $I_{pV}$ , a corrente de saturação do diodo  $I_o$ , o fator de idealidade  $m$ , a resistência série  $R_s$  e a resistência paralelo  $R_p$ . É possível determinar estes parâmetros através de uma metodologia algébrica e aplicar um cálculo iterativo a um sistema de equações ou partindo da análise de dados reais. Alguns pontos de operação importantes, tal como no modelo anterior, podem ser descritos da seguinte forma em condições *STC*:

**1. Curto-circuito ( $V = 0$ ;  $I = I_{cc}^r$ )**

$$I_{cc}^r = I_S^r - I_o^r e^{\frac{R_s I_{cc}^r}{mV_t^r}} - \frac{R_s I_{cc}^r}{R_p} [A] \quad (2.34)$$

**2. Circuito aberto ( $V = V_{ca}^r$ ;  $I = 0$ )**

$$0 = I_S^r - I_o^r e^{\frac{V_{ca}^r}{mV_t^r}} - \frac{V_{ca}^r}{R_p} [A] \quad (2.35)$$

**3. Ponto de máxima potência ( $V = V_{mp}^r$ ;  $I = I_{mp}^r$ )**

$$I_{mp}^r = I_S^r - I_o^r e^{\frac{V_{mp}^r + R_s I_{mp}^r}{mV_t^r}} - \frac{V_{mp}^r + R_s I_{mp}^r}{R_p} [A] \quad (2.36)$$

De acordo com a metodologia algébrica demonstrada em [12], partindo das equações que caracterizam estes pontos de operação particulares é possível obter os parâmetros  $m$ ,  $R_s$ ,  $R_p$ ,  $I_o^r$  e  $I_S^r$  e assim aplicar o modelo. Este modelo, tal como o anterior, é amplamente usado no estudo da célula solar pela sua boa aproximação à célula solar fotovoltaica real.

**2.3.1.4. Modelo de 2 díodos**

O modelo de 2 díodos está referenciado na literatura como o mais preciso, no entanto são necessários 7 parâmetros para a sua aplicação, o que torna a tarefa muito complexa e apenas possível com recurso a métodos alternativos ao algébrico [18].

Este modelo admite que existem perdas de recombinação de portadores de carga na região de depleção da junção p-n, ou seja, a corrente de saturação da célula solar ( $I_o$ ) é o resultado de uma sobreposição linear da difusão das cargas e da sua recombinação. Numa célula solar fotovoltaica, a recombinação de portadores representa perdas

significativas, especialmente quando em operação a tensões baixas, facto que é desprezado nos modelos com apenas um díodo [23]. Este modelo apresenta os mesmos elementos que os modelos anteriores, e um díodo adicional em paralelo que representa a recombinação de portadores de carga (ver Fig. 2.12).

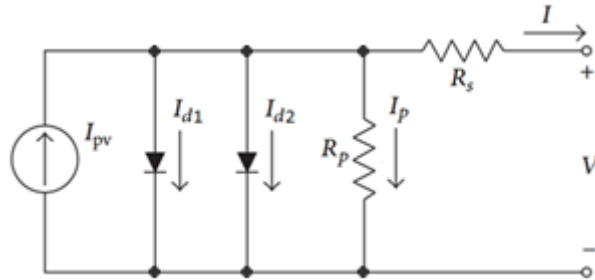


Figura 2.12 – Circuito elétrico equivalente de dois díodos de uma célula solar fotovoltaica.

A equação característica deste modelo é assim dependente de uma nova corrente que atravessa o segundo díodo. Logo, os 2 novos parâmetros a determinar são a corrente de saturação do segundo díodo  $I_{o2}$  e o seu fator de idealidade  $m_2$ :

$$I = I_{pv} - I_{o1} \left( e^{\frac{V+R_s I}{m_1 V_{T1}}} - 1 \right) - I_{o2} \left( e^{\frac{V+R_s I}{m_2 V_{T2}}} - 1 \right) - \frac{V + R_s I}{R_p} \quad [A] \quad (2.37)$$

De forma resumida, os sete parâmetros a determinar para a aplicação deste modelo no estudo de um gerador fotovoltaico são a corrente  $I_{pv}$ , as correntes de saturação dos díodos ( $I_{o1}$  e  $I_{o2}$ ), os fatores de idealidade dos díodos ( $m_1$  e  $m_2$ ), a resistência série  $R_s$  e a resistência paralelo  $R_p$ .

### 2.3.2. Curvas características I-V e P-V

As células fotovoltaicas quando agrupadas formam os painéis fotovoltaicos que apresentam aos seus terminais de saída uma tensão e corrente não lineares, que depende das propriedades físicas e elétricas do material que os constitui, assim como do ponto de funcionamento em que se encontram. Adicionalmente, e como descrito anteriormente, um gerador solar fotovoltaico produz também energia de acordo com as condições de irradiância e temperatura. Esta variabilidade é descrita pela sua característica de saída I-V (corrente-tensão). Esta característica representa todas as combinações de corrente e tensão a que o gerador pode operar ou a carga que este pode alimentar em termos instantâneos, considerando que a irradiância e a temperatura se mantêm constantes [8], [24]. A Figura 2.13 demonstra uma curva I-V típica, assim como a curva P-V que dela

provém e os pontos chave de funcionamento mencionados anteriormente na descrição dos modelos.

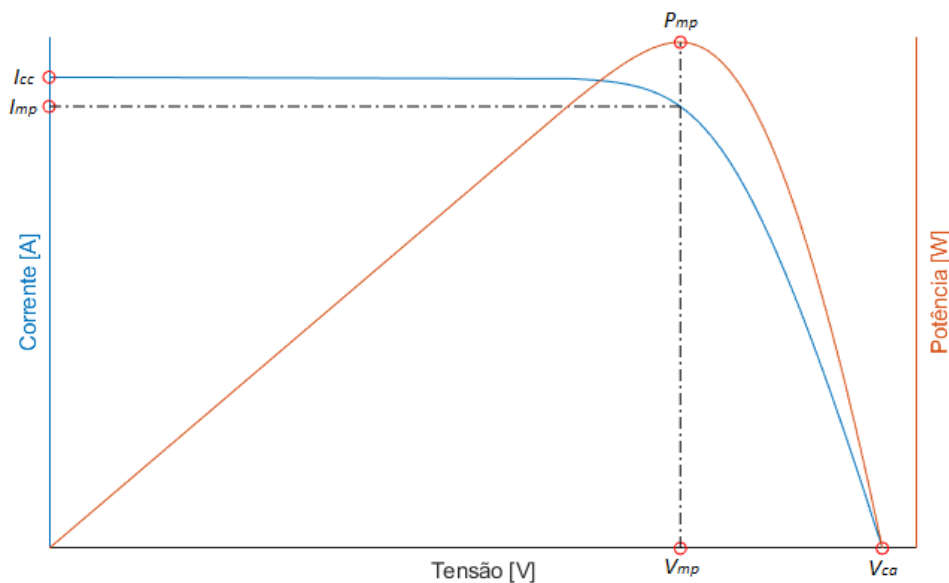


Figura 2.13 – Curvas características I-V e P-V de um gerador solar fotovoltaico.

A curva I-V inicia no ponto de curto-circuito  $I_{cc}$  com a tensão a 0V e termina no ponto em que a corrente é nula, apresentando a tensão de circuito aberto  $V_{ca}$ . O ponto intermédio de maior relevância é onde se situa o ponto de máxima potência, onde o sistema fotovoltaico produz a sua potência elétrica máxima, caracterizado por  $I_{max}$  e  $V_{max}$  ou  $I_{mp}$  e  $V_{mp}$ . Num sistema fotovoltaico, uma das funções do conversor estático de potência conectado aos seus terminais é a de ajustar o valor da carga aos terminais do painel fotovoltaico de forma contínua de modo a extrair a máxima potência do painel em todos os momentos. Esta tarefa baseia-se num algoritmo que se denomina usualmente de *MPPT* (*Maximum Power Point Tracking*) [25]. A curva I-V evidencia alguns aspetos importantes do seu funcionamento. Pode comprovar-se que nos painéis, para tensões abaixo de  $V_{mp}$ , a corrente gerada pelo painel é praticamente independente da tensão de saída, o que significa que o painel funciona de forma semelhante a uma fonte de corrente. Após o ponto de máxima potência, o seu comportamento altera, e a percentagem de cargas elétricas que se recombina nas células solares em vez de alimentar a carga, aumenta com o aumento da tensão. No ponto de tensão aberta do painel, todas as cargas elétricas se recombina internamente. Por outras palavras, pode-se dizer que o painel se comporta como uma fonte de tensão em tensões de operação superiores a  $V_{mp}$ . O ponto de máxima potência define a fronteira entre os modos de operação e é onde o produto entre a corrente e a tensão atingem o valor máximo [8].

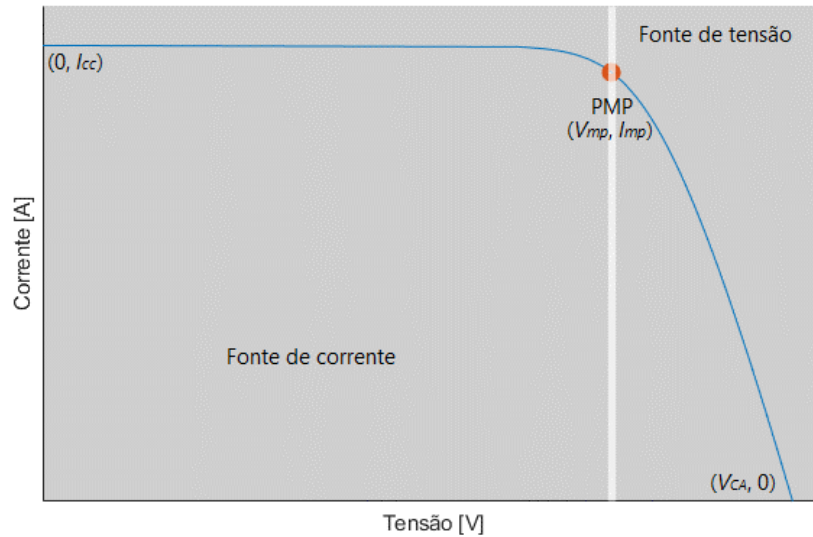


Figura 2.14 – Modos de funcionamento de uma célula fotovoltaica.

### 2.3.3. Variação da irradiância solar incidente e temperatura

Os parâmetros da irradiância solar incidente e temperatura fornecidos pelos fabricantes são em condições STC. Condições essas que dificilmente serão as condições reais de funcionamento de um painel fotovoltaico. Para se processar este ajuste, as seguintes expressões determinam os valores de corrente de curto-circuito  $I_{cc}$  e tensão em circuito aberto  $V_{ca}$  para quaisquer condições de irradiância  $G$  e temperatura do módulo  $T$  [12].

$$I_{cc}(G, T) = \frac{G}{G^r} [I_{cc}^r + \mu_{I_{cc}}(T - T^r)] \text{ [A]} \quad (2.38)$$

$$V_{ca}(G, T) = V_{ca}^r + \mu_{V_{ca}}(T - T^r) + mV_t \ln\left(\frac{G}{G^r}\right) \text{ [V]} \quad (2.39)$$

O coeficiente de temperatura da corrente de curto-circuito é dado por  $\mu_{I_{cc}}$  (A/°C) e o coeficiente de temperatura da tensão em circuito aberto é dado por  $\mu_{V_{ca}}$  (V/°C). Estes coeficientes traduzem a variação das respectivas grandezas com a temperatura e são geralmente fornecidos pelos fabricantes.

Considera-se assim que as grandezas afetadas pela temperatura são apenas as tensões e correntes, ou seja, os parâmetros  $m$ ,  $R_s$  e  $R_p$  são considerados independentes da temperatura.

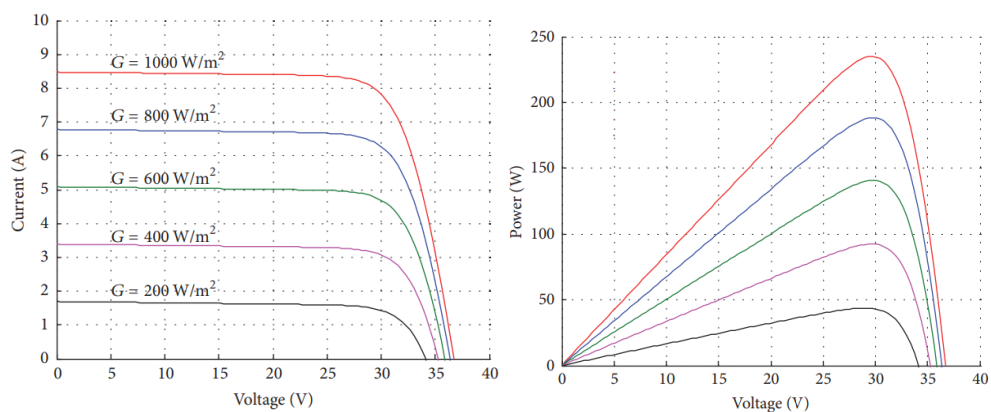


Figura 2.15 – Exemplo de curvas características I-V e P-V com diferentes valores de irradiância solar incidente e temperatura STC ( $T=25^\circ\text{C}$ ) [16].

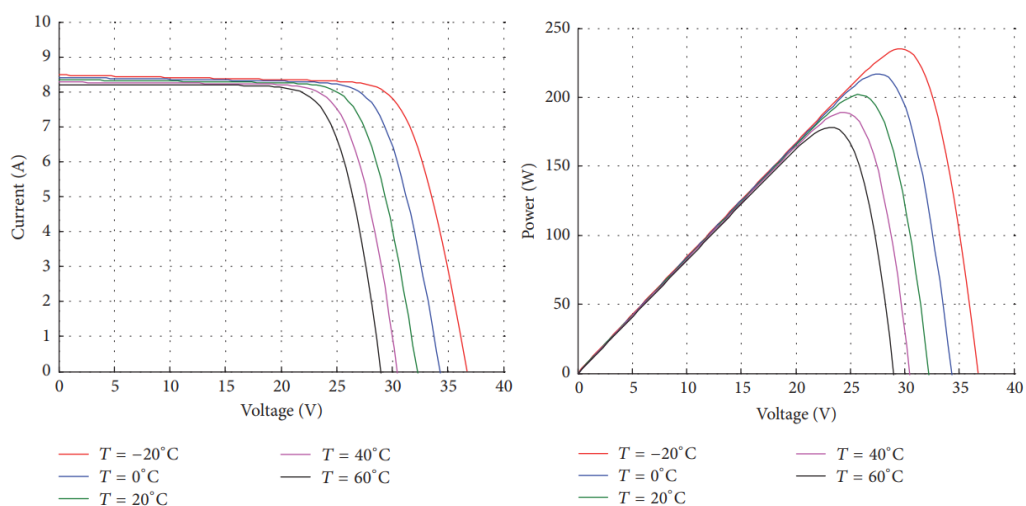


Figura 2.16 – Exemplo de curvas características I-V e P-V com diferentes valores de temperatura e irradiância solar incidente STC ( $G=1000\text{W/m}^2$ ) [16].

### 2.3.4. Painel Fotovoltaico

A potência máxima de uma única célula fotovoltaica não excede 2W, o que é obviamente insuficiente para a maioria das aplicações [12]. De forma a atingir uma potência mais elevada, as células são agrupadas em série, aumentando a tensão de saída para a soma das tensões das células, constituindo assim o que se pode designar de módulos fotovoltaicos [26], tal como ilustrado na Figura 2.17.

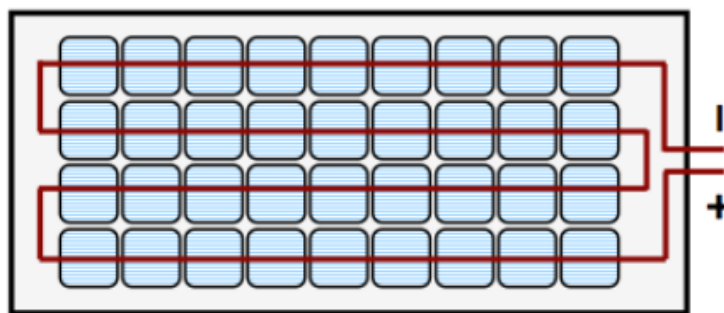


Figura 2.17 – Módulo fotovoltaico constituído por 36 células ligadas em série [26].

No processo de construção dos módulos, as células são normalmente embebidas numa película de etileno acetato de vinilo, um material flexível, translúcido e não refletor da radiação solar, que ainda assegura o isolamento elétrico entre as células. As células são ainda fechadas em caixilharia de alumínio e protegidas frontalmente por vidro temperado, o que lhes garante uma proteção contra esforços mecânicos, adversidades atmosféricas e humidade [26]. Por sua vez, o agrupamento de módulos fotovoltaicos através de ligações em série, paralelo ou mistas, forma o painel fotovoltaico (ver Figura 2.18). Os painéis fotovoltaicos têm geralmente um encapsulamento, por forma a garantir proteção mecânica e durabilidade perante adversidades naturais ao longo da sua vida útil [12]. Os modelos apresentados anteriormente são referentes a uma célula solar, para o estudo do funcionamento de um painel fotovoltaico é necessário fazer alguns ajustes em consideração ao número de células em série e em paralelo. As equações seguintes resumem os ajustes necessários para os modelos de um diodo [27].

$$V_{S(painel)} = N_s V_{S(célula)} \quad (2.40)$$

$$I_{S(painel)} = N_p I_{S(célula)} \quad (2.41)$$

$$I_p(painel) = N_p I_p(célula) \quad (2.42)$$

$$I_o(painel) = N_p I_o(célula) \quad (2.43)$$

$$V_t(painel) = N_s V_t(célula) \quad (2.44)$$

$$R_{S(painel)} = \frac{N_s}{N_p} R_{S(célula)} \quad (2.45)$$

$$R_{p(painel)} = \frac{N_s}{N_p} R_{p(célula)} \quad (2.46)$$

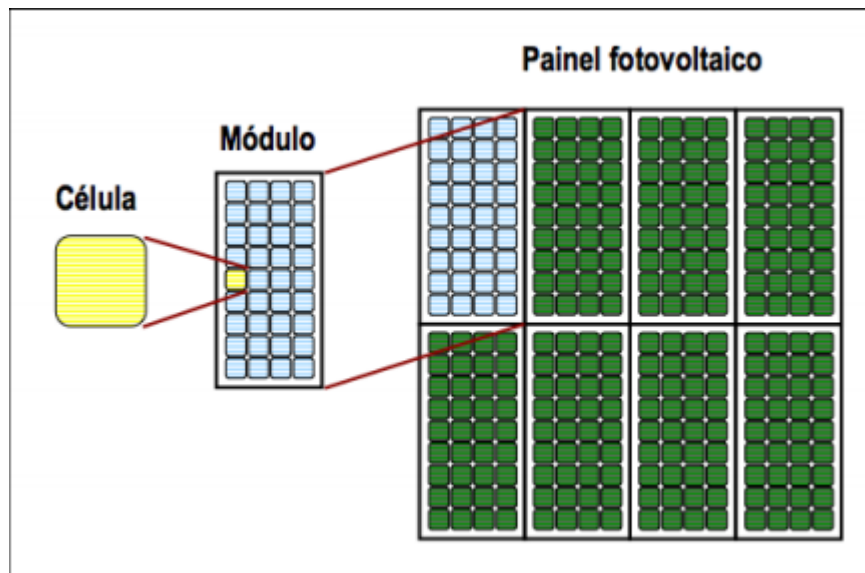


Figura 2.18 – Hierarquia do agrupamento de células fotovoltaicas [26].

Sendo o modelo de 4 parâmetros o aplicado no desenvolvimento desta dissertação, é possível deduzir a equação que caracteriza o funcionamento de um painel partindo da equação que caracteriza o funcionamento de uma célula (Equação (2.21)), considerando as equações (2.40) - (2.46):

$$I = N_p I_S - N_p I_o \left( e^{\frac{V + \frac{N_s}{N_p} R_s I}{m N_s V_T}} - 1 \right) [A] \quad (2.47)$$

### 2.3.5. Associação de painéis fotovoltaicos

Nos sistemas fotovoltaicos que utilizem mais do que um painel fotovoltaico, pode-se ainda associar os vários painéis de diferentes formas, consoante as necessidades externas de tensão e corrente. Neste tipo de associação é importante que os painéis sejam do mesmo tipo, de forma a minimizar perdas de potência por desequilíbrios no sistema [12]. A ligação em série de painéis, tal como na associação de células fotovoltaicas para a formação de módulos, permite obter tensões mais elevadas, mantendo a corrente estipulada do painel. Este tipo de associação pode ser designado por fileiras ou *stack* [26].

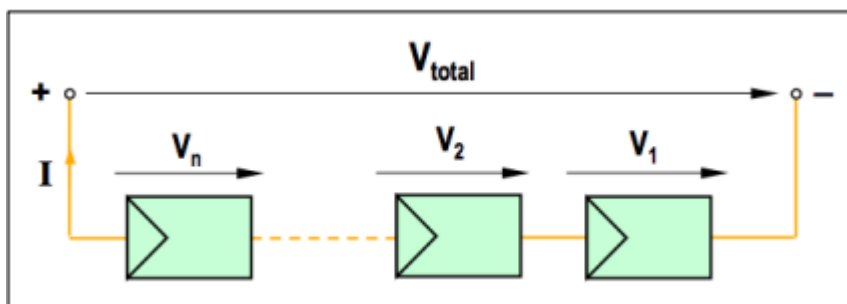


Figura 2.19 – Associação em série de painéis fotovoltaicos [19].

Neste caso, podem ser estabelecidas as seguintes relações para a tensão e corrente:

$$V_1 = V_2 = \dots = V_n = V \Rightarrow V_{total} = V_1 + V_2 + \dots + V_n = n \times V \quad (2.47)$$

$$I_1 = I_2 = \dots = I_n = I \quad (2.48)$$

A ligação em paralelo de painéis fotovoltaicos permite que se obtenha correntes mais elevadas, mantendo a tensão do painel. Este tipo de ligação é usado tipicamente em sistemas autónomos, isolados ou *off-grid* [26].

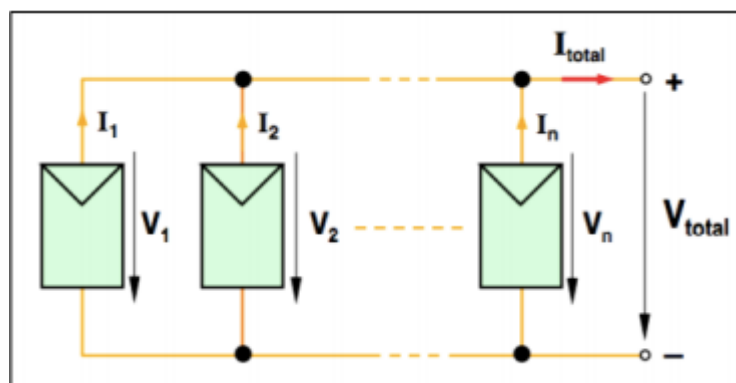


Figura 2.20 – Associação em paralelo de painéis fotovoltaicos [17].

Assim, podem ser descritas da seguinte forma as relações de corrente e tensão para este caso:

$$I_1 = I_2 = \dots = I_n = I \Rightarrow I_{total} = I_1 + I_2 + \dots + I_n = n \times I \quad (2.49)$$

$$V_{total} = V = V_1 = V_2 = \dots = V_n \quad (2.50)$$

Em sistemas ligados à rede, é comum a associação de várias fileiras de painéis ligados em paralelo. Verifica-se assim uma associação  $n \times m$  de painéis, na qual  $n$

representa o número de fileiras de painéis associados em paralelo e  $m$  representa o número de painéis associados em série em cada fileira [26].

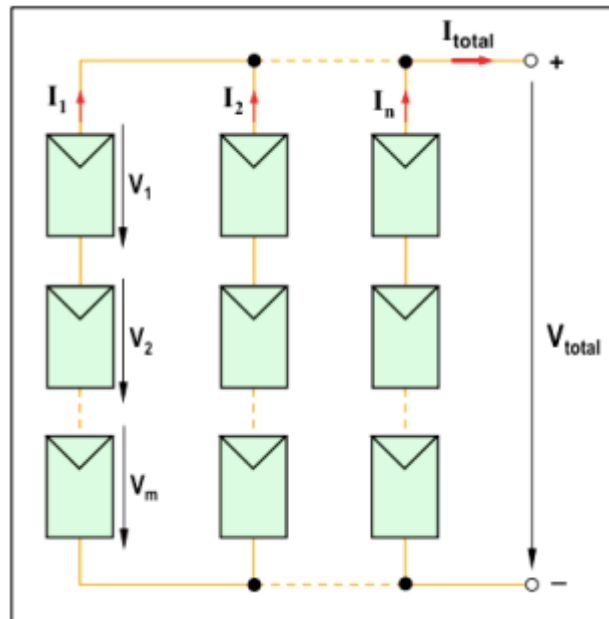


Figura 2.21 – Associação em série e paralelo de painéis fotovoltaicos [17].

Neste tipo de associação, conseguem-se obter tensões e correntes mais elevadas. Se todos os painéis forem iguais, a partir da figura é possível obter a seguinte relação para a corrente total de saída:

$$I = I_1 = I_2 = \dots = I_n \Rightarrow I_{total} = n \times I \quad (2.51)$$

Sendo a tensão de todos os painéis fotovoltaicos igual, é possível obter a tensão total da fileira, que por sua vez, é a tensão de saída pois as fileiras estão ligadas em paralelo:

$$V = V_1 + V_2 + \dots + V_n \Rightarrow V_{total} = m \times V \quad (2.52)$$

### 2.3.6. Tipos de sistemas fotovoltaicos

Para obter o melhor aproveitamento de um sistema fotovoltaico, existem várias configurações possíveis que determinam o tipo de sistema. A sua escolha depende de vários fatores, como localização, possibilidade de ligação à rede elétrica, necessidades dos consumidores, entre outros. Estes sistemas podem ser sistemas ligados à rede (*on-grid*), sistemas isolados (*off-grid*) ou sistemas híbridos [28].

### **2.3.6.1. Sistemas ligados à rede elétrica**

As instalações cujos sistemas fotovoltaicos se encontram ligados à rede elétrica, permitem utilizar energia proveniente da rede elétrica nos momentos em que o sistema fotovoltaico não produza energia suficiente para colmatar as necessidades da carga. Quando o oposto acontece, a energia produzida pelo sistema fotovoltaico superar as necessidades de consumo locais, é possível vender a energia produzida em excesso à rede elétrica [28]. Estes sistemas devem ser conectados à rede de acordo com as normas técnicas e legislação aplicável a sistemas fotovoltaicos, atualmente baseadas nas normas IEC 60364-7-712 e EN 50438. A Figura 2.22 ilustra o conceito de sistema solar fotovoltaico ligado à rede. Estes sistemas carecem também de cuidado no seu dimensionamento face aos consumos do local evitando o produtor privado estar a fornecer energia à rede e a pagar por ela caso não tenha contrato de venda da energia produzida. Para efetuar esta ligação é necessário o uso de um inversor para converter a energia produzida em CC (Corrente Contínua) para CA (Corrente Alternada) ao nível de tensão da rede elétrica. Quando a tensão gerada pelo sistema fotovoltaico não é suficiente para alimentar o inversor é habitual que este possua um conversor CC-CC (Corrente Contínua - Corrente Contínua) que permite subir a tensão para os níveis desejados. Neste tipo de sistemas o armazenamento de energia não é forçosamente necessário, sendo apenas necessários painéis fotovoltaicos e inversor (ou conversor CC-CC + inversor). Como a fiabilidade do fornecimento de energia elétrica está assegurada pela rede elétrica, o dimensionamento destes sistemas é considerado menos crítico do que o dos sistemas isolados da rede [8].

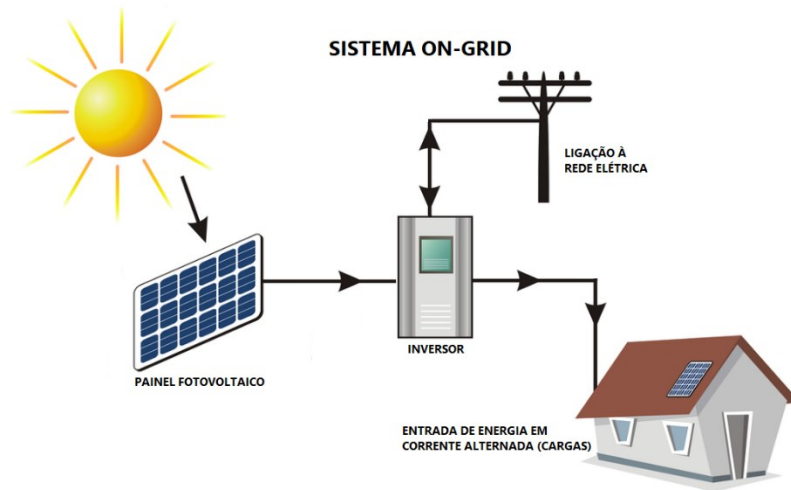


Figura 2.22 – Esquema simplificado de um sistema solar fotovoltaico ligado à rede elétrica.

### 2.3.6.2. Sistemas isolados da rede elétrica

Este tipo de instalações é aplicado tipicamente em regiões remotas, sem acesso à rede elétrica, ou locais onde não é economicamente viável a ligação devido a problemas com os terrenos, dificuldades técnicas com as passagens da rede elétrica ou devido a preocupações ambientais. Num sistema fotovoltaico autónomo toda a energia gerada é consumida instantaneamente pela carga ou armazenada, sem ligação a uma rede elétrica [28]. Alguns sistemas isolados não necessitam forçosamente de armazenamento. É o caso da irrigação, onde toda a água bombeada é diretamente consumida ou colocada em reservatórios [10]. O sistema de armazenamento é fundamental na maioria das aplicações com sistemas fotovoltaicos isolados da rede, garantindo que o excedente de produção de energia é armazenado, podendo esta energia armazenada ser consumida em outros períodos (noite ou dias chuvosos e nublados, com baixos níveis de irradiância solar). Em teoria existem diversas formas de armazenar energia elétrica, por exemplo através do campo elétrico com supercondensadores ou como energia mecânica com volantes de inércia ou bombagem de água. No entanto, a bateria eletroquímica continua a ser o dispositivo mais usado neste contexto, por ser conveniente e eficiente no armazenamento de energia elétrica [8]. A Figura 2.23 ilustra dois processos de utilização do armazenamento em sistemas isolados da rede. No primeiro caso (Fig. 2.23 (a)) possui um carregador de baterias que carrega constantemente as baterias sendo que o inversor procede à sua descarga para alimentar as cargas em corrente alternada. Este sistema desfavorece a duração das baterias pelos ciclos de carga e descarga. No segundo caso (Fig. 2.23 (b)) os painéis fotovoltaicos são ligados diretamente ao inversor (que

pode ter também um conversor CC-CC para subir a tensão proveniente dos painéis solares) que usa a energia proveniente dos painéis para alimentar diretamente a carga, enviando o excesso para as baterias. Se necessário durante os períodos em que o painel fotovoltaico produz energia insuficiente poderá obter-se a restante energia das baterias. Neste caso é também necessário a existência de um conversor CC-CC bidirecional que carrega a bateria e permite descarregar a mesma de forma controlada.

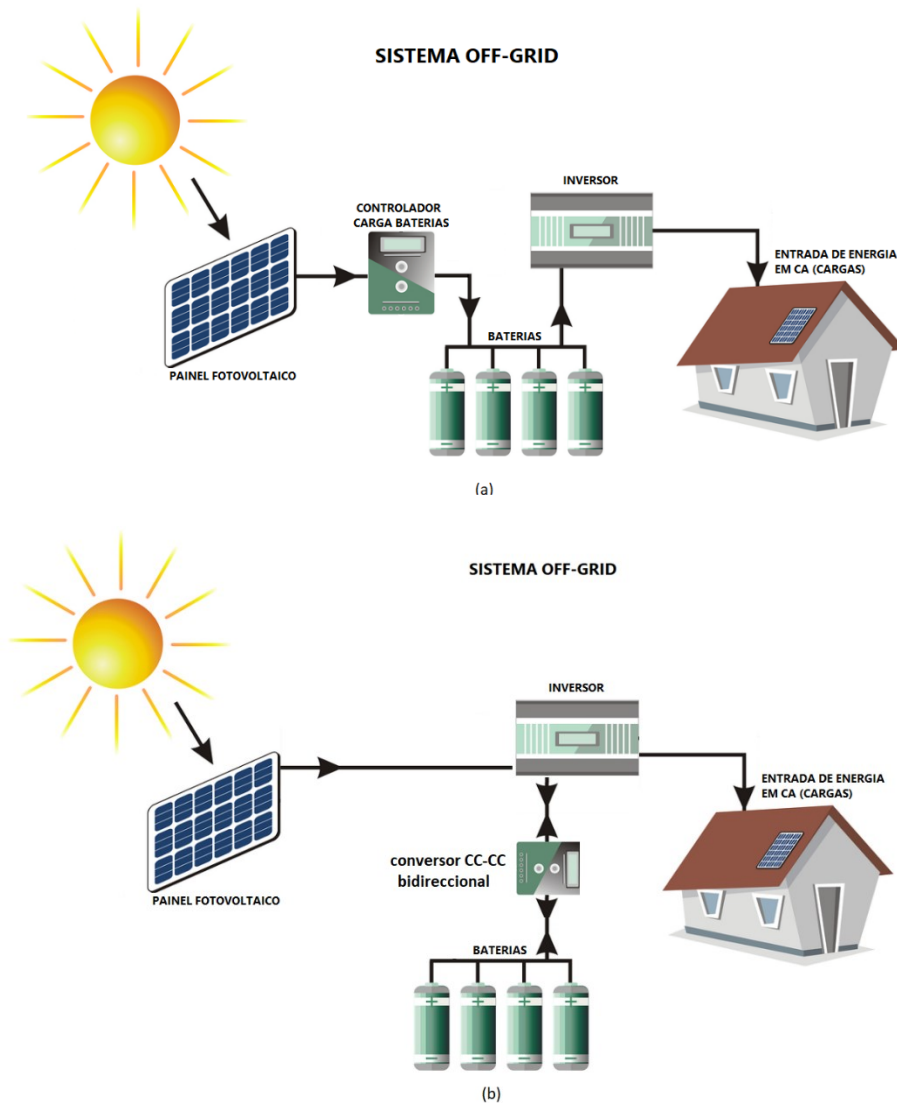


Figura 2.23 – Esquemas simplificados de um sistema isolado da rede elétrica: (a) com recurso a um carregador unidirecional; (b) com recurso a um carregador bidirecional.

### 2.3.6.3. Sistemas híbridos

Este tipo de sistemas combina os descritos anteriormente, ou seja, um sistema ligado à rede elétrica, com armazenamento de energia (tipicamente baterias) normalmente usados em sistemas isolados da rede. Esta combinação de sistemas

permite que a energia produzida em excesso pelos módulos fotovoltaicos seja armazenada nas baterias, e que, posteriormente as baterias descarreguem para alimentar a carga quando os módulos fotovoltaicos não estão a produzir energia suficiente [28]. Existe ainda uma garantia de fornecimento, pois a qualquer momento que a carga necessite de mais energia que a produzida instantaneamente pelos painéis fotovoltaicos e armazenada nas baterias, esta pode sempre ser fornecida pela rede elétrica. A energia produzida pode ainda ser injetada na rede, caso seja em excesso e que as baterias estejam totalmente carregadas. Na Figura 2.24 apresenta-se um exemplo de um sistema híbrido onde, para além do que já foi referido anteriormente, o inversor poderá também acomodar internamente o carregador bidirecional de baterias.

Em sistemas híbridos, o armazenamento de energia também pode ser aplicado com o mesmo fundamento, mas não é considerado crítico, pois o fornecimento de energia está garantido pela rede em caso de falha. Em locais onde o fornecimento de energia pela rede elétrica não é considerado estável, um sistema de armazenamento pode ser aplicado como reserva de energia para quando a rede elétrica falha. Esta última situação não é comum nos países desenvolvidos, é até comum a regulamentação de exigência de proteções que desliguem a produção em casos de falha da rede elétrica [29].

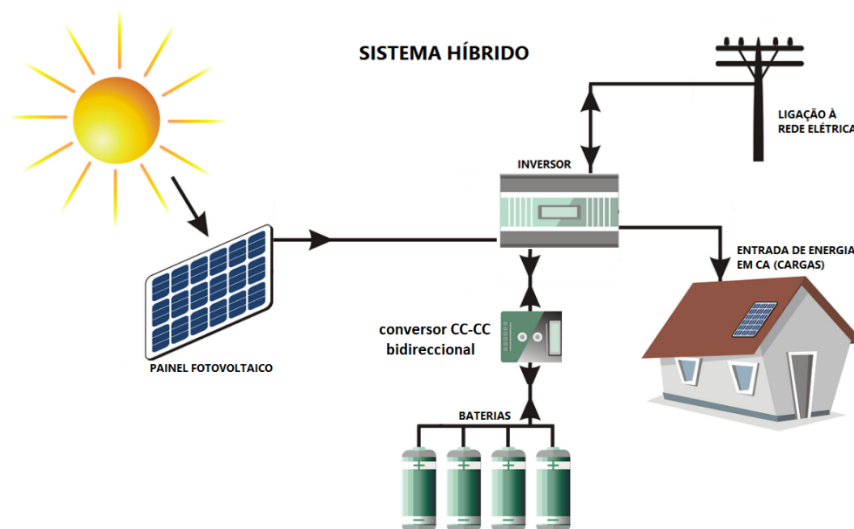


Figura 2.24 – Layout de um sistema híbrido.

### 2.3.7. Simuladores de sistemas fotovoltaicos

Existem geralmente algumas restrições em relação ao teste de campo de painéis *SPV*. Alguns delas estão relacionadas com a variabilidade das condições de funcionamento (já que a irradiância solar e a temperatura não são constantes ao longo

do dia como já foi descrito anteriormente) e a posição relativa entre o sol e os painéis solares muda constantemente [4]. Para fins educacionais, os testes de campo de painéis SPV são geralmente bastante caros e dependem fortemente das condições climáticas. Por esta razão, diversas ferramentas e métodos de baixo custo desenvolvidos para simular o comportamento dos painéis têm sido propostos na literatura por diversos autores. A maioria das ferramentas para simular painéis *SPV* são soluções baseadas em *software* dedicadas aos modelos teóricos (alguns já descritos nas subsecções anteriores). Alguns dos modelos mais comuns são os algoritmos de simulação de tensão dependente da temperatura [30], simulador diferenciável de pico-a-pico [31], modelo de sete parâmetros [32], modelo de circuito equivalente de um diodo [33], [34], modelo de diodo duplo [35] ou modelo de quatro díodos [36]. Outras ferramentas de simulação são dedicadas à investigação de painéis *SPV* em relação a dados operacionais para análise da sua fiabilidade [37]. Alguns autores propõem soluções simples utilizando fonte de alimentação CC em tensão constante e resistências variáveis ajustadas às condições operacionais [38]. Um simulador de painel *SPV* baseado em uma fonte de energia não linear para aplicação de alta potência também foi sugerido por [39]. Existem também soluções baseadas em *hardware* baseadas em conversores de potência comutados a alta frequência, como os propostos por [40], [41] e [42]. Outras soluções de *hardware* e *software* de baixo custo também foram desenvolvidas. Algumas dessas soluções são soluções baseadas em microcontroladores programáveis usando diferentes conversores CC-CC, como *Buck*, *Boost* ou *Buck-Boost* [43], [44], [45] e [46]. Apesar das características dessas soluções, a maioria delas não é flexível o suficiente para aceitar diferentes tipos de módulos *SPV* ou não são capazes de redefinir as condições de operação como irradiância solar e/ou temperatura ou não são capazes de selecionar e implementar diferentes algoritmos *MPPT* quando conectado a aplicações ligadas ou isoladas da rede elétrica. Neste trabalho é usada uma solução com componentes de baixo custo, sendo a base do *hardware* conversores CC-CC (*Buck* e *Boost*) apoiados por soluções de *software* construídas para correr num microcontrolador e num PLC. Permite alterar os parâmetros da simulação de forma a simular diferentes módulos SPV e definir quaisquer condições ambientais em termos de irradiância solar incidente e temperatura.

## 2.4. Conversores Eletrônicos de Potência

É necessário começar por referir a importância que têm os conversores estáticos de potência para os sistemas baseados em energias renováveis e, em especial para os

sistemas solares fotovoltaicos. Dependendo das respectivas aplicações e dadas as características dos sistemas solares fotovoltaicos em termos de tensões e correntes, torna-se normalmente necessário ajustar os níveis de tensão de modo a compatibilizar os painéis e a carga, seja ela uma bateria enquanto sistema isolado (ou híbrido) ou para estabelecer ligação direta à rede de distribuição através de um inversor que é também um conversor estático de potência que passa de uma tensão contínua para tensão alternada à frequência da rede de alimentação. As soluções mais comuns de inversores baseiam-se em topologias de dois níveis de tensão, como por exemplo as apresentadas em [47], [48] ou topologias multiníveis [49], [50]. A maioria das soluções propostas são controladas de modo a extrair a máxima potência dos sistemas solares fotovoltaicos. No entanto estas topologias bem como as suas técnicas de controlo não foram exploradas neste capítulo por não fazerem parte dos objetivos deste trabalho na medida que não se pretende usar a solução desenvolvida para estabelecer ligação à rede elétrica de distribuição.

#### **2.4.1. Topologias de conversores CC-CC adotados em sistemas solares fotovoltaicos**

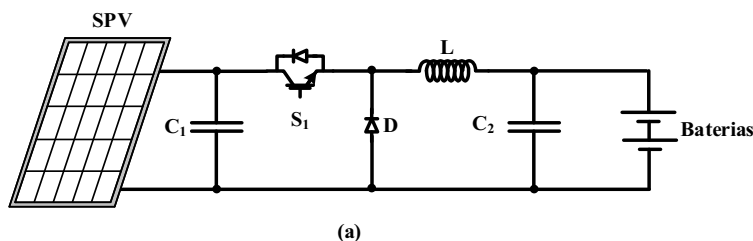
Os conversores estáticos de potência nos sistemas solares fotovoltaicos são dispositivos cujo objetivo é transferir a energia produzida pelos painéis para as baterias (ou banco de baterias, supercondensadores, barramento CC ou diretamente para outras cargas) garantindo a sua proteção contra cargas e/ou descargas excessivas e sobretensões (dependendo da solução utilizada, como descrito na secção 2.3.6.2). São muitas vezes considerados componentes críticos, pois caso falhem, as baterias (ou outros elementos) podem sofrer danos irreversíveis ou simplesmente deixar de conseguir transferir energia. Devem ser dimensionados considerando-se as características das baterias, nomeadamente o seu tipo e curvas de carga e descarga [8] [51] e dos parâmetros dos sistemas solares fotovoltaicos. Os conversores são compostos por um circuito de controlo e por um circuito de potência. O circuito de controlo monitoriza grandezas, tais como a tensão, corrente e temperatura das baterias (ou outros elementos) e gera sinais de comando para o circuito de potência.

O conversor estático de potência é constituído por dispositivos semicondutores de potência que controlam a tensão e/ou corrente da carga ou descarga (dependendo da topologia e método de controlo) das baterias [10]. Os conversores CC-CC apresentam normalmente alta eficiência, controlo relativamente simples e a possibilidade de atingir elevados ganhos de tensão. Os conversores apresentam também algumas desvantagens

relativas ao facto de tensões e correntes quadradas provocarem perdas nos semicondutores ao ligar/desligar, o que limita normalmente as frequências de comutação a algumas centenas de kHz. Estas formas de onda e frequências de comutação podem provocar também interferências eletromagnéticas. Os semicondutores de potência usados atualmente neste tipo de aplicação são os *MOSFET* de potência e os *IGBT*. Estes semicondutores são operados com base num fator de ciclo ( $D$ ), que define o rácio entre os estados ligado e desligado perante o tempo total de operação. Este rácio está sempre compreendido entre 0 e 1. Para uma frequência de operação constante, o fator de ciclo ( $D$ ) entende-se pela relação entre o tempo que o dispositivo semiconductor está à condução face ao período total de operação ( $T$ ).

$$D = \frac{t_{on}}{t_{on} + t_{off}} = \frac{t_{on}}{T} \quad (2.53)$$

Quando aplicados em contexto de sistemas fotovoltaicos, os conversores CC-CC podem classificar-se como elevadores ou redutores, podendo ainda ser unidirecional ou bidirecionais (como exemplificado em 2.3.6.2). A Figura 2.25 (a) ilustra um esquema de um conversor CC-CC redutor (*Buck*) unidirecional utilizado para carregar baterias e é normalmente adotado apenas quando a tensão das baterias é inferior à tensão dos painéis solares. A Figura 2.25 (b) ilustra um esquema de um conversor CC-CC elevador (*Boost*) unidirecional utilizado para carregar baterias e é adotado quando a tensão dos painéis solares é inferior à tensão das baterias. Estes esquemas clássicos são considerados a base para muitas outras topologias que surgiram posteriormente, tal como o *Cúk* [52], *Sepic* [53], *Zeta* [54] ou *Luo* [55], entre outros. Apesar de utilizarem um maior número de componentes estas topologias apresentam normalmente ganhos de tensão mais elevados. A sua escolha depende dos níveis de tensão desejados.



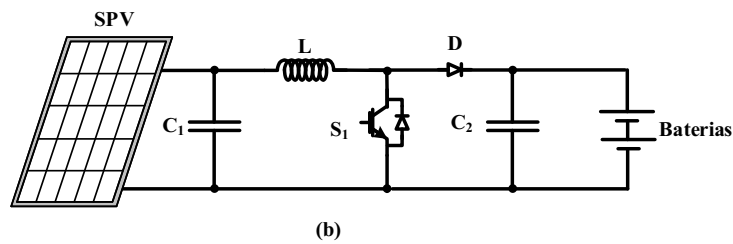


Figura 2.25 – Conversores CC-CC unidirecionais utilizados para carregar baterias em sistemas solares fotovoltaicos: (a) conversor redutor ou *Buck*; (b) conversor elevador ou *Boost*.

Note-se que estes conversores também podem surgir associados aos inversores para fazer subir ou descer a tensão do barramento CC na entrada dos inversores de modo a ser possível ligar à rede de distribuição ou mesmo a carga CA isoladas.

Uma das desvantagens do uso das soluções propostas na Figura 2.25 é o facto de estas só serem adotadas quando se pretende subir ou descer a tensão individualmente não garantindo uma combinação destas duas. Como tal surgiram também soluções que permitem simultaneamente subir ou descer a tensão gerada pelos painéis dependendo do ponto de funcionamento dos painéis solares e das condições climáticas. A Figura 2.26 ilustra dois exemplos de conversores CC-CC redutor-elevador (*Buck-Boost*) unidirecionais.

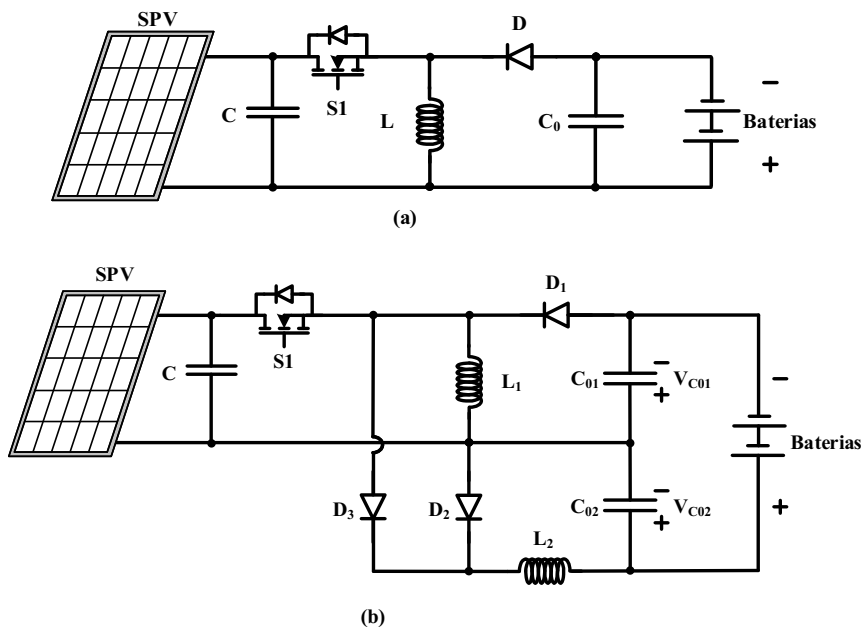


Figura 2.26 – Conversores CC-CC redutores-elevadores (*Buck-Boost*) unidirecionais utilizados para carregar baterias em sistemas solares fotovoltaicos: (a) conversor clássico; (b) exemplo de nova topologia com extensão do ganho de tensão [56].

Os conversores apresentados na figura anterior, pelo facto de serem unidirecionais, ainda constituem uma desvantagem relativamente à possibilidade de carga e descarga das baterias com a mesma topologia como é exigido em soluções como a proposta na

Figura 2.24. Deste modo foram também desenvolvidas soluções de conversores *Buck-Boost* bidirecionais como o que se apresenta na Figura 2.27, permitindo controlar o fluxo de carga e descarga de acordo com o ponto de funcionamento dos sistemas solares fotovoltaicos e carga no instante.

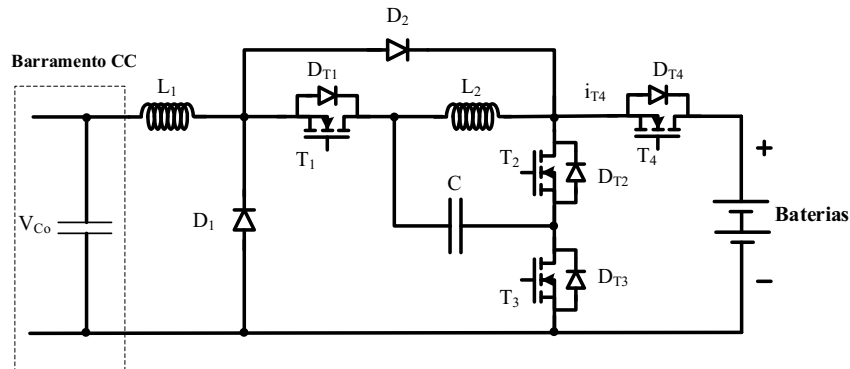
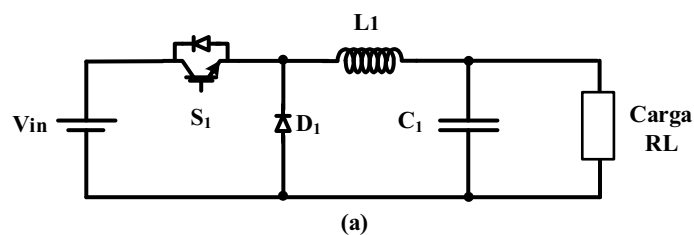


Figura 2.27 – Conversor CC-CC redutor-elevador (*Buck-Boost*) bidirecional utilizado para carregar e descarregar baterias em sistemas isolados da rede ou sistemas híbridos [57].

Foi considerado que o sistema construído neste trabalho seria uma simulação de um sistema isolado com uma carga simples, sem recurso a baterias, de forma a aplicar uma montagem unidirecional que é suficiente para demonstrar o comportamento do módulo *SPV* e do algoritmo *MPPT*. Descreve-se seguidamente o modo de funcionamento dos conversores CC-CC *Buck* e *Boost* unidirecionais cujas topologias são utilizadas nas simulações e resultados experimentais realizados neste trabalho, permitindo assim uma melhor compreensão do seu funcionamento.

#### 2.4.1.1. Conversor CC-CC *Buck* unidirecional

A topologia do conversor CC-CC *Buck* unidirecional está representada na Figura 2.28 (a). O semicondutor é representado como um interruptor ideal ( $S_1$ ), os outros elementos que o compõem são um diodo ( $D_1$ ), uma indutância ( $L_1$ ), um condensador ( $C_1$ ) e uma carga ( $RL$ ).



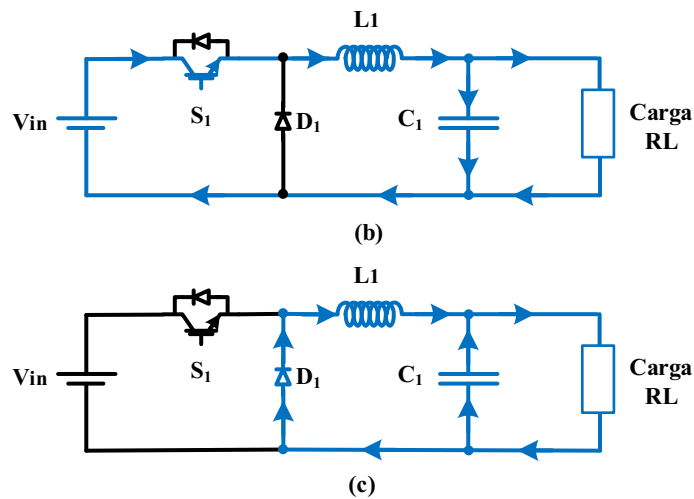


Figura 2.28 – Modos de funcionamento do conversor CC-CC redutor (*Buck*) unidirecional: (a) – topologia; Modo I – Interruptor  $S_1$  ligado; Modo II – Interruptor  $S_1$  desligado.

Neste conversor existem essencialmente dois modos de funcionamento, o modo I quando o interruptor de potência  $S_1$  é ligado e o modo II quando o interruptor de potência é desligado. No modo I, quando o interruptor  $S_1$  é ligado, a tensão aplicada na carga sobe lentamente de acordo com a carga do condensador  $C_1$  através da indutância  $L_1$ . Quando o interruptor  $S_1$  é desligado no modo II, a energia armazenada no campo magnético da indutância é descarregada para a carga através do diodo  $D_1$ , fazendo a tensão da carga e do condensador  $C_1$  baixar lentamente. A tensão média aplicada à carga é a relação entre a abertura e o fecho de  $S_1$ , denominada por fator de ciclo (*duty-cycle*) multiplicada pela tensão de entrada [58].

Assumindo que o circuito funciona no modo contínuo, isto é, que a corrente na bobine ( $L_1$ ) nunca é nula, as formas de onda típicas deste conversor estão representadas na Figura 2.29.

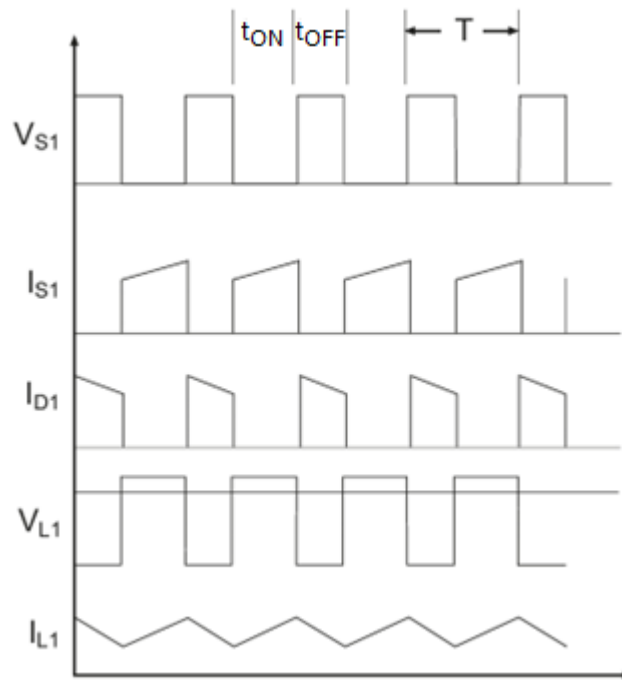


Figura 2.29 – Forma de onda típicas de um conversor CC-CC *Buck* unidirecional [23].

A função de transferência deste circuito é determinada pela análise dos dois modos de funcionamento admitindo componentes ideais. Pelo princípio da conservação de energia, a energia que entra na indutância tem de igualar a energia que sai, ou seja, a relação tensão-tempo dos dois modos de funcionamento tem de ser igual [58].

Com  $S_1$  fechado:

$$Energia_{IN} = (V_{IN} - V_{OUT}) \times t_{ON} \quad (2.54)$$

Com  $S_1$  aberto:

$$Energia_{OUT} = V_{OUT} \times t_{OFF} \quad (2.55)$$

Sabendo que

$$t_{OFF} = T - t_{ON} \quad (2.56)$$

E que:

$$D = \frac{t_{ON}}{T} \quad (2.57)$$

Ao igualar as equações da energia:

$$(V_{IN} - V_{OUT}) \times t_{ON} = V_{OUT} \times t_{OFF} \leftrightarrow \frac{V_{OUT}}{V_{IN}} = D \quad (2.58)$$

Conclui-se que a função de transferência é o fator de ciclo ( $D$ ), ou seja, este representa a relação entre a tensão de saída e entrada. A tensão de saída é igual à tensão

de entrada no caso em que  $S_1$  está fechado e diminui proporcionalmente durante o tempo em que  $S_1$  está aberto.

### 2.4.1.2. Conversor CC-CC *Boost* unidirecional

A topologia do conversor *Boost* está representado na Figura 2.30 (a). Tal como no conversor *Buck*, o semicondutor é representado como um interruptor ideal ( $S_1$ ) e os outros elementos que o compõem são um diodo ( $D_1$ ), uma indutância ( $L_1$ ), um condensador ( $C_1$ ) e uma carga ( $RL$ ).

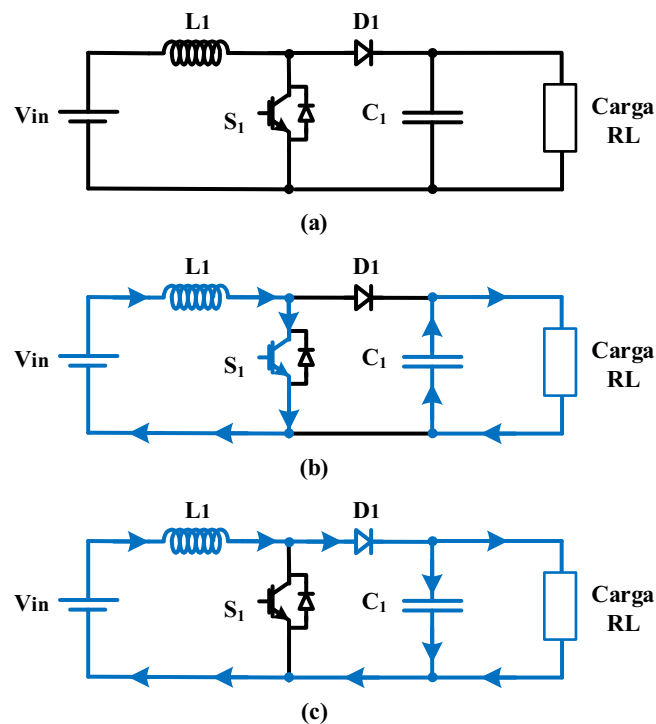


Figura 2.30 – Modos de funcionamento do conversor CC-CC elevador (*Boost*) unidirecional: (a) – topologia; Modo I – Interruptor  $S_1$  ligado; Modo II – Interruptor  $S_1$  desligado.

Tal como no conversor *Buck*, também neste conversor existem dois modos de funcionamento, o modo I quando o interruptor de potência  $S_1$  é ligado e o modo II quando o interruptor de potência  $S_1$  é desligado. Com  $S_1$  ligado, a corrente que atravessa a indutância  $L_1$  aumenta linearmente na relação  $\frac{V_{IN}}{L_1}$ . Durante este período a carga é alimentada com a energia armazenada no condensador  $C_1$ . Quando o interruptor  $S_1$  abre, a energia armazenada na indutância provoca uma inversão da tensão aos seus terminais que somada com a tensão de entrada ultrapassa a tensão de saída colocando o diodo  $D_1$  em condução alimentando a carga e carregando o condensador  $C_1$ . A corrente

que atravessa a indutância  $L_1$ , reduz-se linearmente de acordo com  $\frac{(V_{OUT}-V_{IN})}{L_1}$  [58]. Assumindo que o circuito funciona no modo contínuo, isto é, que a corrente na bobine ( $L_1$ ) nunca é nula, as formas de onda típicas deste conversor estão representadas na Figura 2.31.

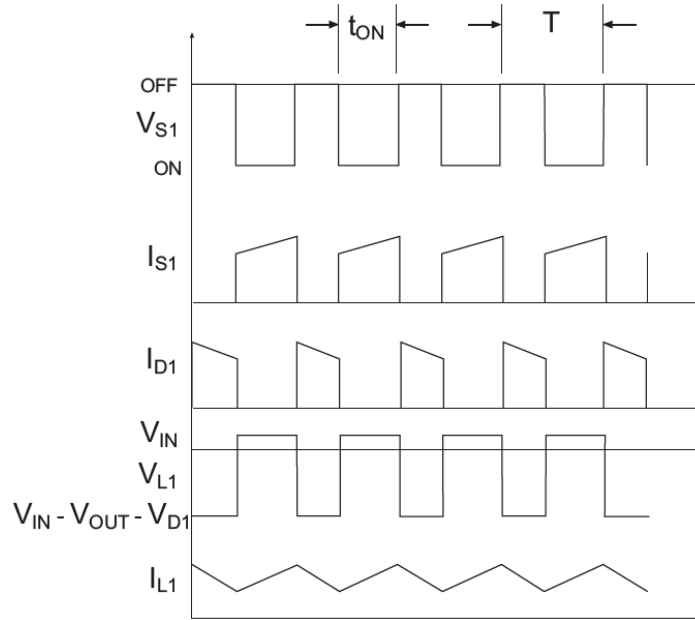


Figura 2.31 – Forma de onda típicas de um conversor CC-CC *Boost* unidirecional [23].

A função transferência obtém de forma semelhante à do conversor *Buck*, partindo de equações iniciais diferentes:

Com  $S_1$  fechado:

$$Energia_{IN} = V_{IN} \times t_{ON} \quad (2.59)$$

Com  $S_1$  aberto:

$$Energia_{OUT} = (V_{OUT} - V_{IN}) \times t_{OFF} \quad (2.60)$$

A função transferência é assim dada por:

$$\frac{V_{OUT}}{V_{IN}} = \frac{1}{1 - D} \quad (2.61)$$

## 2.4.2. Controlo de conversores CC-CC adotados em sistemas solares fotovoltaicos

Neste contexto, o controlo dos conversores CC-CC mais básicos têm um funcionamento básico de liga-desliga (*on-off*). Este tipo de controlador aplica

diretamente a tensão e a corrente do painel sobre a bateria (ou outro dispositivo). Neste caso, o painel fotovoltaico funciona como uma fonte de corrente limitada pela sua corrente de curto-circuito  $I_{CC}$  nas condições instantâneas de operação (irradiância solar e temperatura), com o valor de tensão estabelecido pela bateria. A estratégia de controlo dos controladores *on-off* é baseada na tensão instantânea nos terminais da bateria, que é comparada a dois limites. No limite superior, a bateria é desconectada do painel fotovoltaico. No limite inferior, a bateria é conectada novamente ao sistema fotovoltaico [8].

Os controladores mais modernos e disponíveis atualmente a nível comercial para este efeito partem das características básicas apresentadas anteriormente, mas apresentam funcionalidades adicionais e podem ser categorizados em dois tipos distintos. O tipo de controlador moderno mais simples é denominado usualmente de “Controlador de carga *PWM*” e recorre a um microprocessador que, por monitorização das grandezas elétricas da bateria, gera sinais *PWM* e efetua a carga das baterias em vários estágios. Os controladores de carga *PWM* são muito comuns e representam uma solução relativamente simples e de baixa complexidade, embora não seja a mais eficaz para extração de potência do painel. Os controladores mais sofisticados, denominados de “Controlador de carga *MPPT*”, aplicam uma estratégia de seguimento do ponto de máxima potência ao painel fotovoltaico de forma a aumentar a eficácia da produção de energia e do processo de carga. O equipamento consegue operar com uma ampla gama de tensão de entrada, e incluem normalmente um conversor CC-CC elevador (Corrente Contínua - Corrente Contínua) como primeiro estágio, de forma a ter uma melhor combinação entre a curva I-V do painel e as baterias. Este tipo de controlo também pode ser aplicado em inversores nos sistemas ligados à rede elétrica sem armazenamento de energia.

### **2.4.3. Os Algoritmos *MPPT***

Os algoritmos *MPPT* servem para maximizar a produção de potência de um gerador fotovoltaico em todas as condições de funcionamento. Os vários tipos de sistemas fotovoltaicos podem apresentar configurações distintas adequadas à sua utilização final. Independentemente do destino da energia produzida, o problema que o *MPPT* pretende colmatar é a variação da eficácia de produção de energia das células solares que é dependente da irradiância solar incidente no painel fotovoltaico, da temperatura do painel fotovoltaico e das características elétricas da carga. Perante a variação destas

características, a característica da carga que causa a máxima eficiência do painel também varia, que pode ser observado ao analisar a sua curva I-V. A eficácia do sistema encontra o seu ponto ótimo quando a característica da carga se altera de forma consistente para manter o painel fotovoltaico a operar no ponto de máxima potência [8], [59].

O algoritmo *MPPT* traduz o processo de encontrar o ponto de máxima potência (PMP) (*MPP – Maximum Power Point*) e manter lá a característica da carga. Podem ser usados determinados circuitos elétricos para apresentar um valor de carga arbitrário às células solares por forma a extrair a máxima potência do painel fotovoltaico e em seguida converter a energia produzida para alimentar a carga pretendida, carregar baterias ou injetar na rede elétrica [8], [12]. Tipicamente, o *MPPT* está integrado no primeiro estágio do conversor de potência que se apresenta aos terminais de saída do painel fotovoltaico, seja este um controlador de carga (em sistemas isolados ou híbridos) ou um inversor (em sistemas ligados diretamente à rede elétrica) [8]. Segue-se uma descrição de algumas das abordagens mais comuns de algoritmos *MPPT*. Uma revisão de este e outros métodos pode ser encontrado em [60], [61], [62], [63].

#### **2.4.3.1. Método de tensão constante**

Este método tem como base a observação de que o ponto de máxima potência ocorre entre 72% e 78% da tensão de circuito aberto, para condições normais de funcionamento [64]. O módulo solar fotovoltaico opera a uma tensão constante dentro da referida gama, ou seja, o fator de ciclo (ou *duty-cycle*) do conversor CC-CC que garante que a tensão de saída é igual a:

$$V_{REF} = K_1 \times V_{ca} \quad (2.62)$$

Com  $K_1$  compreendido entre 0,72 e 0,78. O método é simples, rápido e fácil de implementar, mas demonstra uma precisão limitada, a tensão em circuito aberto tem de ser medida regularmente e é apenas aplicado em locais onde a variação de temperatura é residual [64].

#### **2.4.3.2. Método da célula piloto**

Este método segue o princípio do método anterior, mas tenta colmatar a necessidade de desconectar o módulo fotovoltaico da rede para obter uma medição da

tensão de circuito aberto. O procedimento consiste em desconectar apenas uma célula solar, designada por célula piloto, para obter a medição necessária [65]. A tensão correspondente ao ponto de máxima potência pode assim ser obtida por:

$$V_{REF} = K_2 \times V_{ca(célula\ piloto)} \quad (2.63)$$

Com  $K_2 < 1$ .

### 2.4.3.3. Método da corrente constante

O mesmo princípio dos métodos anteriores pode ser aplicado à corrente de curto-circuito, ao invés da tensão de circuito aberto. O módulo fotovoltaico irá assim operar a uma corrente constante [66]. Empiricamente, sabe-se que o ponto de máxima potência ocorre entre 78% e 92% da corrente de curto-circuito por:

$$I_{REF} = K_3 \times I_{CC} \quad (2.64)$$

Com  $K_3$  compreendido entre 0,78 e 0,92.

### 2.4.3.4. Método de Perturbar e Observar (*P&O*, *Perturb and Observe*)

O algoritmo de *P&O* é um dos aplicados de forma mais comum na prática, pela sua facilidade de implementação. Este algoritmo consiste na aplicação de uma perturbação periódica na tensão de operação e seguidamente ao verificar a variação de potência, deduz a direção a atribuir à tensão de referência (ver Figura 2.32). Então, se a tensão de operação é perturbada numa dada direção e a potência extraída do painel fotovoltaico aumenta, isto significa que o ponto de operação se aproximou do PMP e a tensão de operação tem de continuar a ser perturbada na mesma direção. De outra forma, se a potência extraída do painel diminui, o ponto de operação afastou-se do PMP e a direção da perturbação tem de ser revertida. Desta forma, a tensão de operação é perturbada de forma sucessiva, e, assim que é atingido o PMP, irá oscilar em torno da tensão ideal  $V_{mpp}$ . Isto causará uma perda de potência que será tanto maior quanto a dimensão do passo de perturbação  $C_p$ .

A figura seguinte ilustra o funcionamento deste algoritmo para um determinado instante  $k$ . São medidos os valores de tensão ( $V(k)$ ) e corrente ( $I(k)$ ) nesse instante e

calculada a respetiva potência ( $P(k)$ ). A perturbação é aplicada na zona inferior do diagrama, de acordo com a dimensão do passo de perturbação ( $C_p$ ) pré-estabelecida.

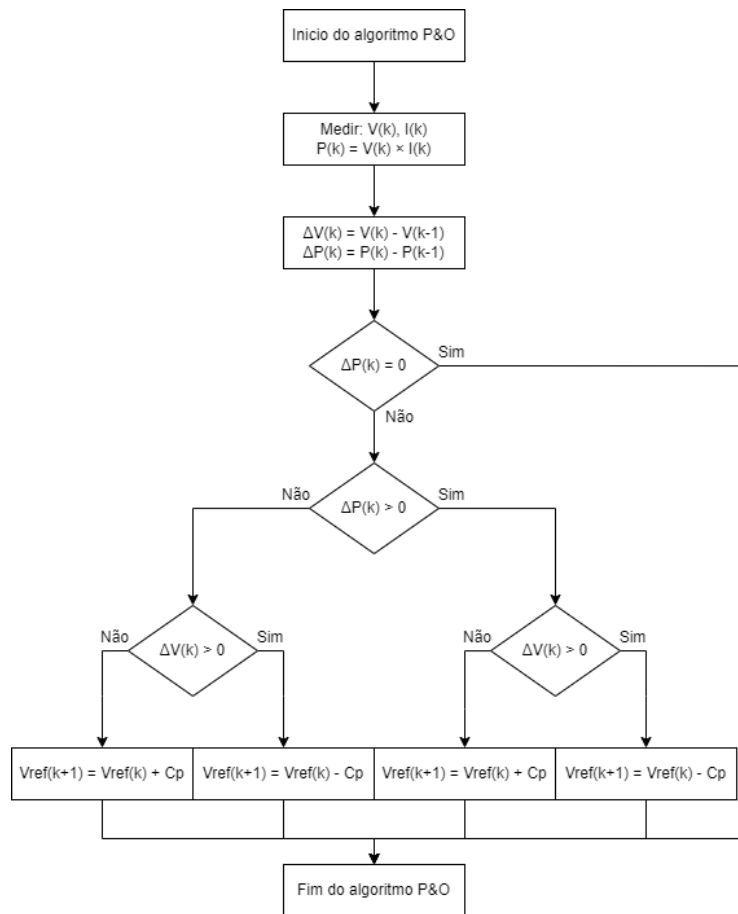


Figura 2.32 – Diagrama de fluxo do método P&O.

De acordo com a Figura 2.32 é possível enumerar as 5 hipóteses possíveis a cada ciclo. Estas hipóteses estão descritas de seguida e resumidas na Tabela 2.1.

- Hipótese 1: após uma perturbação de tensão positiva (incremento da tensão) e exista um aumento de potência, significa que o ponto de operação está à esquerda do PMP e a aproximar-se do PMP, a próxima perturbação será positiva novamente;
- Hipótese 2: após uma perturbação de tensão positiva (incremento da tensão) e exista um decréscimo de potência, o ponto de operação situa-se a direita do PMP e a afastar-se do PMP, a próxima perturbação terá de ser negativa para ir de encontro ao PMP;
- Hipótese 3: após uma perturbação de tensão negativa e ocorra uma alteração positiva na potência, significa que o ponto de operação está à

direita do PMP e a aproximar-se do PMP, a próxima perturbação será negativa novamente;

- Hipótese 4: após uma perturbação de tensão negativa e ocorra uma alteração negativa na potência, o ponto de operação situa-se a esquerda do PMP e a afastar-se do PMP, a próxima perturbação terá de ser positiva para ir de encontro ao PMP;
- Hipótese 5: para uma qualquer perturbação anterior, caso não exista alteração na potência, significa que o ponto de operação está no PMP e não é necessária qualquer perturbação neste ciclo.

Tabela 2.1 – Metodologia do método *P&O*.

Hipótese	Perturbação	Alteração na potência	Próxima perturbação
1	Positiva	Positiva	Positiva
2	Positiva	Negativa	Negativa
3	Negativa	Positiva	Negativa
4	Negativa	Negativa	Positiva
5	Positiva/Negativa	Constante	Sem perturbação

A maior desvantagem do método *P&O* comparativamente com outros métodos, nomeadamente o método da condutância incremental descrito de seguida e outros métodos mais complexos não descritos no presente trabalho, é um ocasional desvio do ponto de máxima potência quando as condições atmosféricas se alteram de forma rápida. Logo, é fundamental que a dimensão do passo de perturbação e a velocidade do passo de cálculo seja adequada para garantir um bom desempenho na resposta dinâmica e estática.

#### **2.4.3.5. Método da condutância incremental**

O método da condutância incremental segue o princípio de tentativa e erro do método *P&O* simples. Este método baseia-se no facto que o declive é nulo no PMP na curva *P-V* (ver Figura 2.33). Tal como no método *P&O*, a tensão e a corrente são monitorizadas à saída do gerador fotovoltaico pelo controlador *MPPT*. Porém, neste caso, de forma a encontrar o PMP são feitas comparações sucessivas entre a condutância instantânea e a condutância incremental, com o objetivo que estas se igualem.

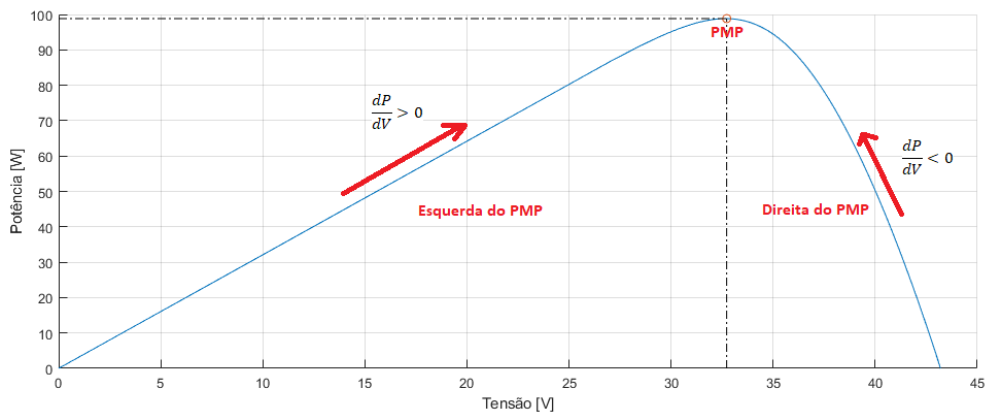


Figura 2.33 – Regiões possíveis do ponto de operação de um gerador fotovoltaico, relativas ao PMP.

O método P&O na prática, ao atingir o PMP fica constantemente em oscilações em torno desse ponto. Por sua vez, o método de condutância incremental reduz essas oscilações, pois ao atingir o PMP não é aplicada qualquer perturbação. As comparações sucessivas permitem que o algoritmo saiba qual a zona de operação em que o painel SPV está a funcionar (ver Figura 2.33 e Equação 2.65). O diagrama de fluxo deste método é apresentado na Figura 2.34.

$$\left\{ \begin{array}{l} \frac{dP}{dV} > 0 \text{ se } \frac{dI}{dV} > -\frac{I}{V}, \quad \textit{esquerda do PMP} \\ \frac{dP}{dV} < 0 \text{ se } \frac{dI}{dV} < -\frac{I}{V}, \quad \textit{direita do PMP} \\ \frac{dP}{dV} = 0 \text{ se } \frac{dI}{dV} = -\frac{I}{V}, \quad \textit{PMP} \end{array} \right. \quad (2.65)$$

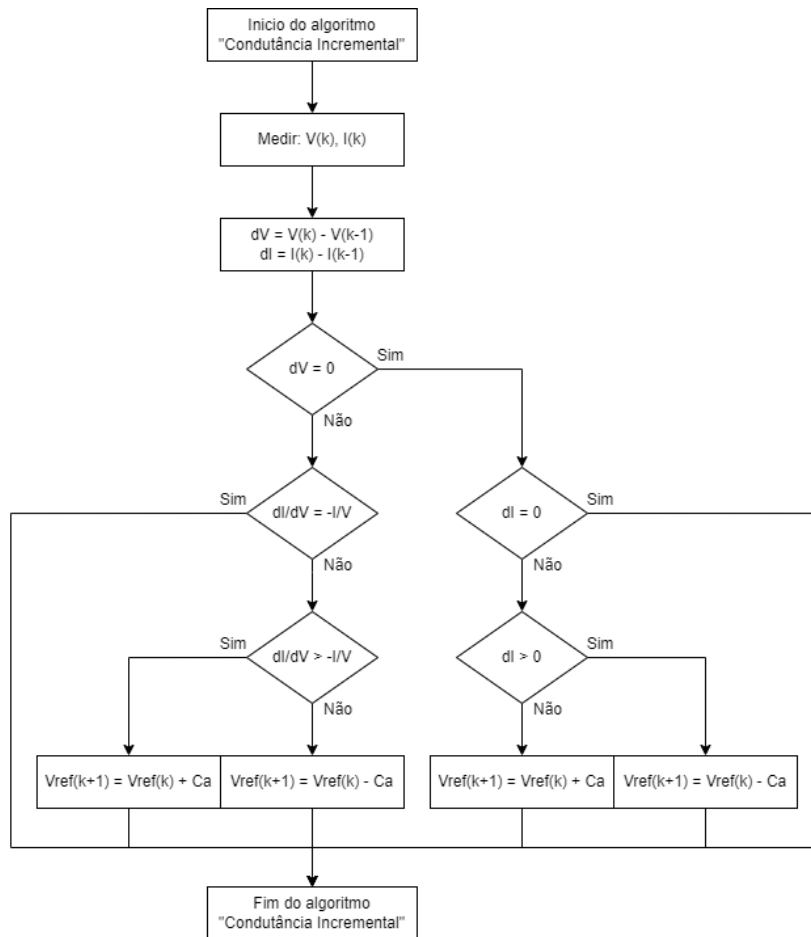


Figura 2.34 – Diagrama de fluxo do método da Condutância Incremental.

## 2.5. Sistemas Embebidos

Um sistema embebido é a designação de um sistema no qual se combina em termos de *hardware* um microprocessador com um *software* destinado a desempenhar uma tarefa específica. O sistema pode ser independente ou como parte englobante de um sistema maior. A complexidade destes sistemas pode ir desde a utilização de um microcontrolador até a utilização de vários microcontroladores com periféricos e redes de comunicação, com ou sem interface gráfica. A complexidade varia com as tarefas a desempenhar [67], [68]. O microprocessador por si só, não é, muitas vezes, suficiente para implementar um sistema e requer muitas vezes outros sistemas auxiliares. No contexto desta dissertação procurou-se informação sobre sistemas embebidos e equipamentos periféricos dado que estes são parte integrante da solução proposta neste trabalho e como um grau de relevância significativo.

### 2.5.1. Microcontroladores

Os microcontroladores são tipicamente dispositivos que combinam um ou mais microprocessadores e outros elementos periféricos que combinados formam um computador miniatura, constituído por núcleos de processamento, memórias, temporizadores e entradas e saídas. Estas entradas e saídas (I/O) podem ser digitais, analógicas e, podem ainda, disponibilizar outras funções específicas como interrupções ou portas de comunicação. Estes dispositivos programáveis foram desenvolvidos para diversas aplicações, estando presentes na maioria dos equipamentos eletrónicos. Ao nível do mercado, encontram-se disponíveis com uma grande diversidade de especificações e preços [68].

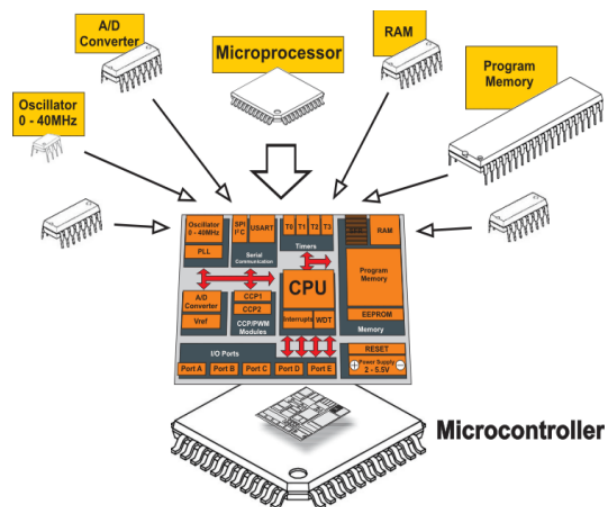


Figura 2.35 – Componentes de um microcontrolador.

A programação de um microcontrolador consiste na criação de uma estrutura lógica, geralmente em linguagem Assembly ou C/C++, das instruções que permitem ao microcontrolador desempenhar as tarefas a que se destina. Após isto é necessário efetuar a compilação e a transferência do programa desenvolvido, do compilador para a memória Flash do microcontrolador. O compilador consiste numa aplicação que fornece um ambiente de desenvolvimento, teste e depuração de programas, onde é gerado um ficheiro que contém as instruções a serem executadas pelo microcontrolador. A transferência do programa para o microcontrolador é usualmente feita recorrendo a um software específico que varia de acordo com o fabricante e o modelo a programar. Após isto, o microcontrolador ficará a executar as tarefas definidas, permitindo, de acordo com o restante hardware, desempenhar as funções pretendidas.

Para a presente dissertação será aplicado um microcontrolador da Microchip, modelo dsPIC30F4011 (ver Figura 2.36). Este microcontrolador é de 16bits e funciona com uma tensão de alimentação de 5V, as suas características principais encontram-se descritas na Figura 2.37.



Figura 2.36 – Microchip dsPIC30F4011.

**dsPIC30F Motor Control and Power Conversion Family**

Device	Pins	Program Mem. Bytes/Instructions	SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	Motor Control PWM	10-bit A/D 1 Msp	Quad Enc	UART	SPI	I <sup>2</sup> C™	CAN
dsPIC30F4012	28	48K/16K	2048	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	1
dsPIC30F4011	40/44	48K/16K	2048	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	1

Figura 2.37 – Características principais dos microcontroladores da gama dsPIC30F.

A construção do programa do microcontrolador será feita em C, usando o software da Microchip concebido para o efeito, o “MPLAB X”. A programação do microcontrolador será efetuada por intermédio de um “PICkit 3” (Figura 2.38), dispositivo que permite a interação entre PC e dsPIC cujas funções principais são: converter as instruções do programa construído e gerar os sinais necessários para a sua escrita no microcontrolador; permitir a depuração das instruções do dispositivo através do software de desenvolvimento da Microchip “MPLAB X”.



Figura 2.38 – Dispositivo PICkit 3.

## 2.5.2. Comunicações

No decorrer do projeto, serão usados vários tipos de comunicações envolvendo o microcontrolador dsPIC. Segue-se uma descrição generalizada dos tipos de comunicação aplicados, assim como o âmbito específico para esta aplicação.

### 2.5.2.1. *UART*

A UART (Universal Asynchronous Receiver/Transmitter) consiste num transmissor-recetor de comunicação série bidirecional assíncrono entre dois ou mais dispositivos presentes na maior parte dos microcontroladores. Para tal, cada dispositivo necessita efetuar a conversão de paralelo para série dos dados a enviar (Tx) e efetuar o processo inverso na receção (Rx). Esta conversão consiste em enviar a informação pretendida bit a bit a um determinado débito binário (Baud Rate) pré-estabelecido entre o emissor e o recetor.

O facto de ser uma comunicação assíncrona indica que não existe um sinal de relógio independente do sinal de dados, deste modo, para que haja sincronismo e que a mensagem seja interpretada pelo recetor são enviadas tramas curtas e dados adicionais no início de cada transmissão com o intuito de permitir o sincronismo por parte do recetor. Do mesmo modo, o fim da mensagem e o bit de paridade, podem ser adicionados no fim da trama, sendo o fim da mensagem obrigatório.

Este método de comunicação é, por vezes, referido como “TTL serial” (transistor-transistor logic). A comunicação série ao nível TTL aplica os limites de 0V e Vcc (tipicamente 5V ou 3,3V). O nível lógico “1” é representado por Vcc e o nível lógico “0” é representado por 0V, podendo por vezes ser invertido este esquema (ver Figura 2.39).

Uma das vantagens da utilização deste tipo de portas de comunicação em microcontroladores deve-se à simplicidade na sua adaptação para implementar uma comunicação com computadores. A porta série dos computadores cumpre o standard RS-232-C e apresenta diferenças face ao TTL. Segundo o standard RS-232-C, o nível lógico “1” é representado por uma tensão negativa (entre -3V e -15V), enquanto o nível

logico “0” é representado por uma tensão positiva (entre 3V e 15V) (ver Figura 2.39). Tipicamente, nos computadores, as tensões na porta série variam entre -13V e 13V.

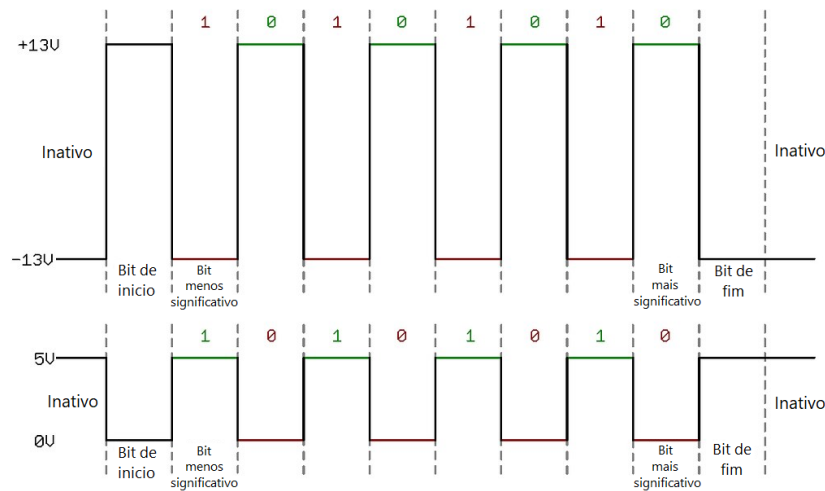


Figura 2.39 – Exemplo de mensagem série em RS-232 (em cima) e TTL (em baixo).

Neste projeto, este tipo de comunicação será usado para a comunicação entre o microcontrolador e o computador, assim como para comunicação entre o microcontrolador e um Autômato Programável, vulgarmente conhecido como PLC.

Atualmente, face à raridade de portas RS-232-C nos computadores, é possível efetuar este tipo de comunicação emulando uma porta RS-232-C via Universal Serial Bus (USB). Para tal, basta converter, a porta série (UART) para o standard USB, recorrendo a conversores USB – UART, sendo interpretada pelo computador como uma porta série standard RS-232-C (ver Figura 2.40).



Figura 2.40 – Conversor USB – UART.

Para satisfazer a comunicação entre o microcontrolador e um PLC, foi aplicado um outro tipo de conversor, que tal como os conversores USB – UART, é baseado no circuito integrado “MAX232”, visto que o PLC, tal como os computadores, apresenta um funcionamento seguindo o standard RS-232-C (ver Figura 2.41).



Figura 2.41 – Conversor UART (TTL) - RS232.

### 2.5.2.2. Rede de comunicação *SPI*

Na grande maioria dos microcontroladores atuais está disponível uma rede de comunicação denominada de SPI (Serial Peripheral Interface). Este tipo de comunicação série síncrona possui uma topologia mestre – escravo de 4 sinais, sendo amplamente usado em sistemas embebidos para comunicar com diversos componentes periféricos que não estão contemplados nos microcontroladores e permitem fazer a sua expansão a outros dispositivos através de comunicação digital. As aplicações comuns desta tecnologia são a comunicação entre o microcontrolador e dispositivos periféricos como conversores analógico-digital (ADC) e digital-analógico (DAC), sensores, cartões de memória, ecrãs, entre outros. A rede SPI permite a implementação de comunicações bidirecionais sendo o sinal MOSI referente à comunicação do mestre para o escravo e o MISO o inverso. Neste protocolo, a comunicação inicia-se quando o dispositivo mestre da rede coloca o sinal SS (Slave Select) do respectivo escravo ao nível lógico “0”, sendo o dispositivo mestre também responsável por gerar o sinal de relógio SCLK, necessário para o sincronismo da comunicação (ver Figura 2.42).

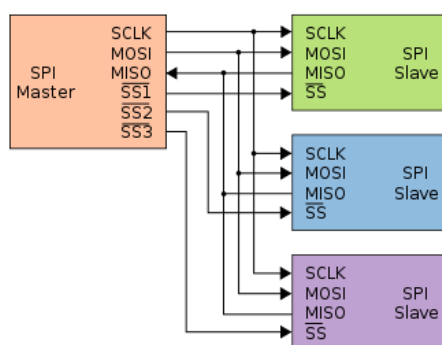


Figura 2.42 – Esquema de ligações de uma comunicação *SPI*.

Uma vez que o microcontrolador aplicado neste projeto não dispõe de saídas analógicas, será necessária a utilização de um dispositivo adicional para este efeito. Será

assim aplicado na forma de circuito integrado o DAC de 12-bits “MCP4922” que comunica através da rede SPI com o microcontrolador. Neste caso, o microcontrolador será o mestre e enviará um sinal digital para o “MCP4922” para que este o converta em sinal analógico e disponibilize o respetivo nível de tensão em uma das suas duas saídas.



Figura 2.43 – DAC de 12-bits “MCP4922”.

## 2.6. Controladores industriais e supervisão

### 2.6.1. Controladores Lógicos Programáveis (PLC)

Um controlador lógico programável (*PLC*) é um tipo de controlador com um microprocessador que usa um espaço de memória programável para guardar instruções e implementar funções lógicas, sequenciais, temporizadas e aritméticas de forma a controlar máquinas ou processos. São desenhados de forma a serem programados e operados de forma simples, através de linguagens de programação intuitivas. Foram desenvolvidos na década de 60 do século passado, nos Estados Unidos, com a finalidade de substituir painéis de relés que eram muito utilizados na indústria automóvel [69].

Semelhante a um microcontrolador, tipicamente um *PLC* é constituído por alguns componentes fundamentais, tais como, uma unidade de processamento central (*CPU*), memória, uma unidade de alimentação, uma secção de interface com *inputs* e *outputs* (*I/O*) com isolamento galvânico, uma interface de comunicação e um *software* de programação (ver Figura 2.44).

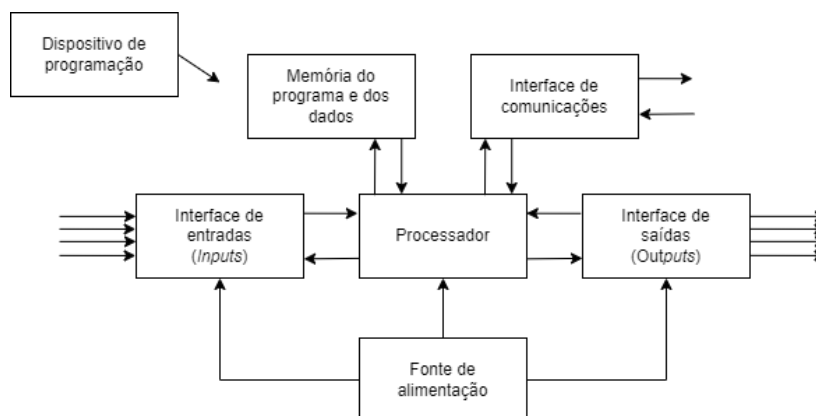


Figura 2.44 – Composição típica de um *PLC*.

A *CPU* contém o microprocessador, que interpreta os sinais de input e desempenha as ações de controlo programadas, comunicando as decisões tomadas como sinais de output. A unidade de alimentação é necessária para converter a tensão de entrada de 230V em CA para a tensão CC necessária para o processador e os circuitos nos módulos de interface dos inputs e outputs. Existem também variantes com alimentação em 24VCC e 24VCA.

O *software* de programação é usado para construir e carregar o programa para a memória do *PLC*. É tipicamente usado um computador para esta tarefa, com um *software* específico do fabricante do *PLC*.

A memória é onde o programa é carregado, onde são armazenados os *inputs* para processamento e de onde os valores de *output* são enviados. Também determinadas áreas da memória podem ser usadas para guardar registos do programa, necessários para o processamento das tarefas a desempenhar.

A secção dos *inputs* e *outputs* é onde o processador recebe e transmite informação com dispositivos externos, normalmente com isolamento galvânico. Os *inputs* podem ser de diversos tipos, como interruptores ou uma grande variedade de sensores. Os *outputs* podem ser também muito variados, sendo os mais comuns, sinalizadores ou relés e contactores para o arranque de motores. Estes *inputs* e *outputs* podem ser classificados como binários (apenas 2 estados: *on* e *off*) ou analógicos (o sinal é proporcional à variável monitorizada).

A interface de comunicação é usada para receber e enviar dados em redes de comunicação com determinados protocolos, tipicamente com outros *PLC*, mas também pode ser usada com outro tipo de controladores, tais como microcontroladores. Serve para transmitir dados entre dispositivos, verificações e sincronização entre aplicações do utilizador (ver Figura 2.45).

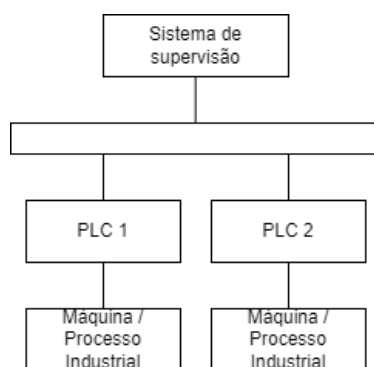


Figura 2.45 – Modelo básico de comunicações industriais.

O *PLC* usado nesta dissertação é o *ILC 131 ETH*, do fabricante Phoenix Contact (ver Figura 2.46). A gama *ILC 1X1* deste fabricante caracteriza-se por suportar os protocolos *Modbus/TCP* e *PROFINET Ethernet*. O seu processador tem uma velocidade de 64 MHz e é programável através do *software PC WORX*, seguindo a norma *IEC 61131-3*. Este *PLC* é modular e o seu módulo base inclui 8 entradas digitais e 4 saídas digitais. As interfaces de comunicação deste módulo incluem:

- *Ethernet* 10/100 Mbps, através da ficha RJ45;
- *INTERBUS* local bus (master), para ligação em barramento;
- Porta de parametrização/operação/diagnostico (comunicação série), através da ficha *MINI DIN* de 6 pinos (PS/2).

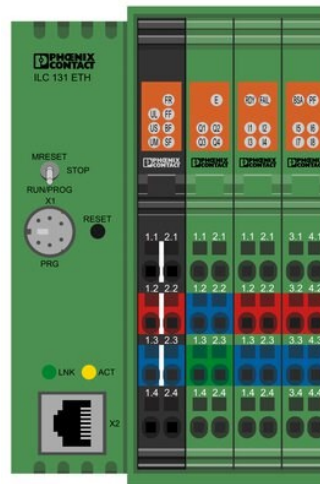


Figura 2.46 – *Layout* do *PLC* ILC 131 ETH da Phoenix Contact.

### 2.6.2. Norma IEC 61131-3

As cinco partes do standard *IEC 61131* definem os requisitos dos sistemas de *PLC* modernos. Estes requisitos referem-se ao *hardware* do *PLC* e à sua forma de programação.

A terceira parte do standard, *IEC 61131-3*, é vista como um guia para a programação de *PLC* e tem como objetivos:

- Fornecer metodologias de construção de lógicas de programação de forma estruturada e modular, permitindo a quebra dos programas em partes gerenciáveis;
- Definir 5 linguagens de programação, cada uma com suas características, de forma a cobrir a maioria das necessidades de controlo atuais;
- Permite o uso de outras linguagens de programação, desde que obedecidas as mesmas formas de chamadas e trocas de dados (*Visual Basic*, *Flow Chart*, *C++*, etc.).

Dentro destes aspetos, a *IEC 61131-3* define cinco linguagens de programação:

- *ST* (*Structured Text*) - Texto Estruturado;
- *IL* (*Instruction List*) - Lista de Instruções;

- *LD (Ladder)* - Diagrama de contactos;
- *FBD (Function Block Diagram)* - Diagrama de blocos;
- *SFC (Sequential Flow Chart)* - Diagrama de Fluxo.

As linguagens *ST* e *IL* são construídas na forma de texto. As duas seguintes (*LD* e *FBD*) são ditas gráficas por possuírem representação na forma de símbolos. A linguagem *SFC* é normalmente tida como linguagem gráfica, mas também permite programação na forma de texto. Com a possibilidade de combinar todas estas linguagens, é possível controlar um processo de uma forma simples e estruturada, assim como vários processos simultaneamente.

Na presente dissertação foram usadas as linguagens *ST* e *FBD*. A primeira para a criação de uma função principal no programa, que de uma forma sequencial corre as instruções necessárias para aplicação do algoritmo *MPPT*. A segunda para a criação de funções adicionais, com o propósito da leitura e escrita respetivamente das entradas e saídas analógicas do *PLC* e também para a comunicação série em *RS-232-C*.

### **2.6.3. Sistemas de supervisão (SCADA)**

*SCADA* é um acrónimo formado pelas primeiras letras da expressão “*Supervisory control and data acquisition*”, em português: Supervisão do controlo e aquisição de dados. Um sistema *SCADA* permite que um operador, numa localização centralizada, de um grande sistema distribuído possa ter visualização e controlo em tempo real sobre os controladores em funcionamento, tipicamente autómatos programáveis, enquanto estes automatizam um determinado processo industrial através de dispositivos de campo (ver Figura 2.47) [70].

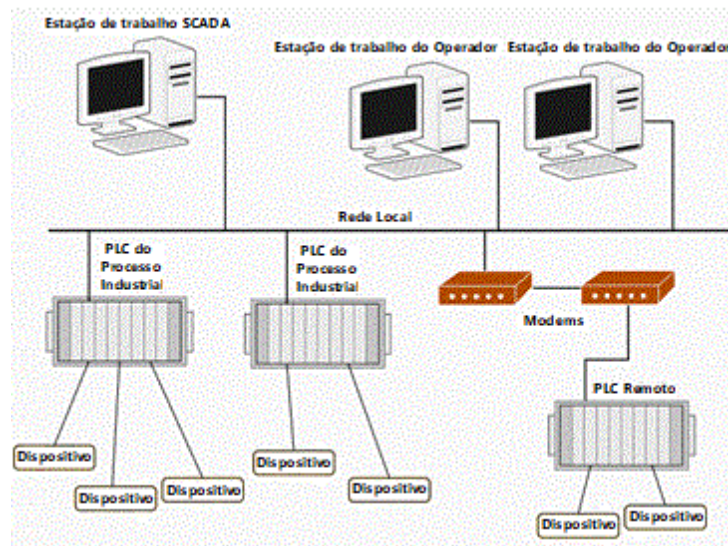


Figura 2.47 – Exemplo de configuração de um sistema SCADA.

As principais vantagens da utilização de um sistema *SCADA* em associação com um determinado processo controlado por *PLCs* são a centralização da informação, permitindo ao operador visualizar possíveis falhas dos processos em tempo real, assim como atuar manualmente nos dispositivos de campo. Os sistemas de supervisão são também uma matéria vasta pelas diversas arquiteturas que podem existir, diferentes formas de armazenar e tratar informação e mesmo de estabelecer comunicações a diversos níveis.

### 3. Trabalho desenvolvido

Neste capítulo é apresentado o trabalho desenvolvido para a construção do simulador. É demonstrado o sistema implementado em termos de simulações e desenvolvimento de *hardware* e descritas as interações entre os vários elementos que o compõem.

#### 3.1. Arquitetura proposta

O sistema implementado tem o propósito de funcionar como uma simulação de um sistema de produção de energia fotovoltaica. Para tal, os elementos típicos deste tipo de sistemas foram simulados por outros componentes. Os elementos principais são três conversores CC-CC *Buck-Boost* clássicos, uma fonte de alimentação CC regulada, um microcontrolador *dsPIC30f4011*, um *PLC Phoenix Contact ILC 131 ETH* com um sistema *SCADA* embebido que comunica através do protocolo *Modbus TCP/IP*, transdutores analógicos de condicionamento de sinal (corrente e tensão), circuitos analógicos de modo deslizante para realização da modulação *PWM* e uma carga RL. O sistema foi desenvolvido para operar em 2 modos diferentes: o primeiro modo (modo I) ou modo painel *SPV* e o segundo modo (modo II) ou modo de teste *MPPT*.

O diagrama de blocos do modo I é apresentado na Figura 3.1. O funcionamento do sistema neste modo pretende testar e simular a curva I-V do painel fotovoltaico de acordo com os parâmetros do fabricante considerando diversos valores de irradiância solar e temperatura. Neste modo são apenas usados dois conversores CC-CC *Buck-Boost* (ambos a funcionar como conversores *Buck*; o primeiro simula o painel *SPV*; o segundo varia o valor da carga de forma linear através de modulação *PWM*).

Inicialmente a fonte de alimentação CC é ajustada para 60V para alimentar o primeiro conversor CC-CC. Nos seus terminais de saída estão ligados um transdutor de tensão e um transdutor de corrente que enviam a informação destas grandezas para duas entradas analógicas do microcontrolador. Este, através de um cálculo interno baseado no modelo de 4 parâmetros do painel *SPV*, define o valor de tensão correto a impor. Seguidamente envia este valor para um *DAC* externo por comunicação *SPI*. O *DAC* envia a respetiva tensão de controlo para o circuito de modulação *PWM*. Neste circuito, a tensão de controlo ao ser comparada com uma onda triangular de 60 kHz é gerado um

o sinal *PWM* que serve de sinal de disparo para a gate do *MOSFET* presente no primeiro conversor CC-CC.

Os parâmetros do painel solar e condições atmosféricas podem ser ajustados através da página web do sistema *SCADA* [71]. Ao realizar qualquer ajuste, o sistema *SCADA* envia a informação para o *PLC* que por sua vez transmite os dados necessários para o microcontrolador através de comunicação *RS232C*. Através do sistema *SCADA*, é também possível de regular o valor da carga através de uma barra deslizante. Tendo em conta o valor submetido, o *PLC* envia um sinal de controlo através de uma saída analógica que, após comparação com uma onda triangular de 100 kHz, resulta num sinal *PWM*. É também possível automatizar a recolha de resultados através da aplicação de valores crescentes de sinal para o circuito de modulação *PWM* o que leva a um aumento gradual do valor da carga RL. Os valores de corrente e tensão à saída do primeiro conversor CC são registados em tempo real e apresentados ao utilizador de forma gráfica, com a possibilidade de exportação de resultados via ficheiro “csv” para posterior análise. A figura seguinte ilustra o diagrama de blocos do funcionamento do modo I.

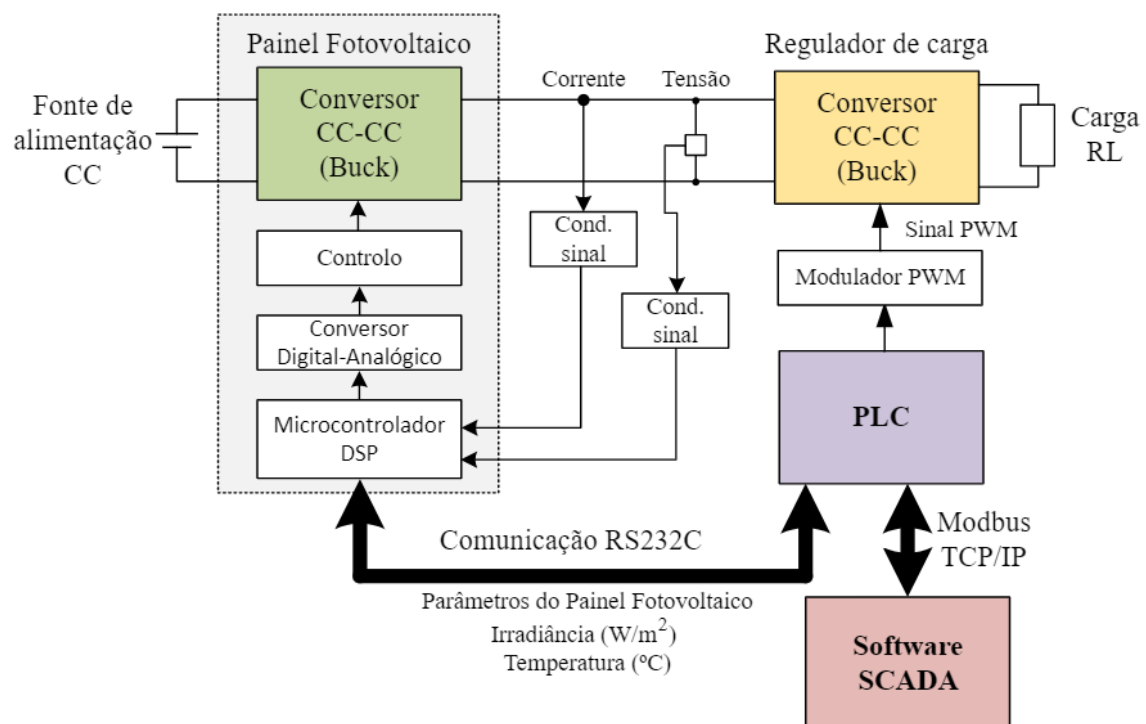


Figura 3.1 – Diagrama de blocos do modo I (modo painel SPI).

O diagrama de blocos do modo II é apresentado na Figura 3.2. Esta configuração é usada para testar o algoritmo *MPPT* e avaliar a potência entregue à carga em diversas condições. Neste modo são usados três conversores CC-CC *Buck-Boost*, em que dois

operam no modo *Buck* (com a mesma funcionalidade e comportamento do modo I) e um no modo *Boost*. A operação é semelhante à do modo I, mas o conversor adicional permite a implementação do algoritmo *MPPT* no sistema. A energia atravessa o conversor *Boost* e aos seus terminais de saída estão presentes um transdutor de tensão e um transdutor de corrente, que, de forma análoga aos transdutores do primeiro conversor, recolhem informação sobre as grandezas elétricas (tensão e corrente) e enviam, neste caso, para duas entradas analógicas do *PLC*. Ao processar os valores obtidos, é enviado um sinal de controlo através de uma saída analógica que, após comparação com uma onda triangular de 100 kHz, resulta num sinal *PWM* para o disparo do *MOSFET* presente no conversor *CC-CC Boost*.

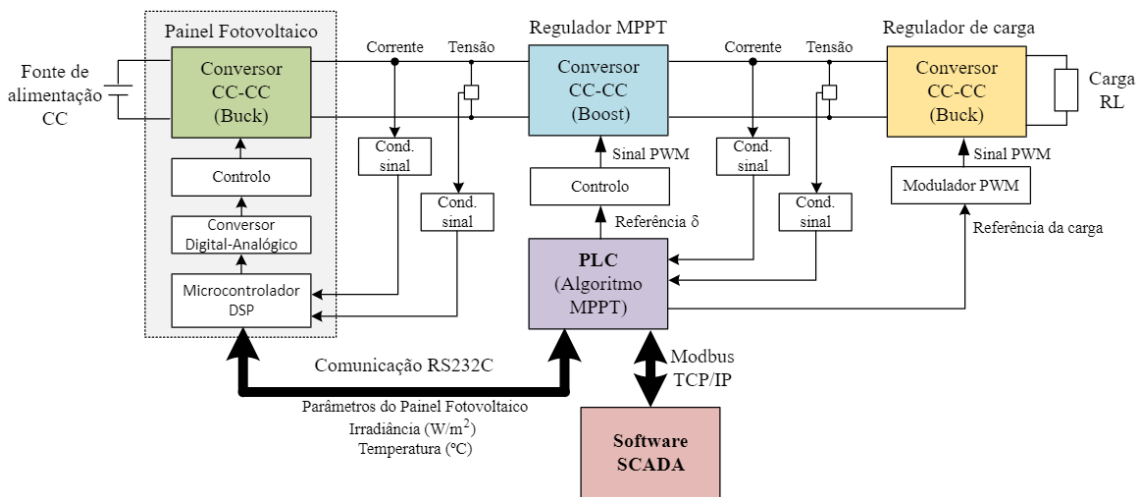


Figura 3.2 – Diagrama de blocos do modo II (modo de teste *MPPT*).

### 3.2. Painel Fotovoltaico a simular

O painel fotovoltaico a simular neste estudo será o modelo *ASE-100-DG-UR* [72] da marca *Schott Solar*. O dispositivo é constituído por células de silício monocristalino e apresenta uma potência nominal de 100W. Seguidamente apresentam-se as suas principais características.

Tabela 3.1 – Características do Painel Fotovoltaico ASE-100-DG-UR [72].

<b>Electrical Characteristics</b>	
<b>STC Power Rating <math>P_{mp}</math> (W)</b>	100
<b>Open Circuit Voltage <math>V_{oc}</math> (V)</b>	43.2
<b>Short Circuit Current <math>I_{sc}</math> (A)</b>	3.22
<b>Voltage at Maximum Power <math>V_{mp}</math> (V)</b>	35.7
<b>Current at Maximum Power <math>I_{mp}</math> (A)</b>	2.81
<b>Panel Efficiency</b>	11.2%
<b>Fill Factor</b>	71.9%
<b>Power Tolerance</b>	-10.00% ~ 10.00%
<b>Maximum System Voltage <math>V_{max}</math> (V)</b>	1000
<b>Mechanical Characteristics</b>	
<b>Cell Type</b>	Monocrystalline Cell
<b>Cells</b>	72
<b>Dimensions</b>	1017.0 × 881.0 × 11.0mm (34.7 × 40.0 × 0.4 inch)
<b>Weight</b>	20.0Kg (44.1 lbs)

### 3.3. Simulação do sistema proposto

A partir da base de dados do *software PVsyst*, [73] foram recolhidos dados adicionais para a simulação (Figura 3.3).

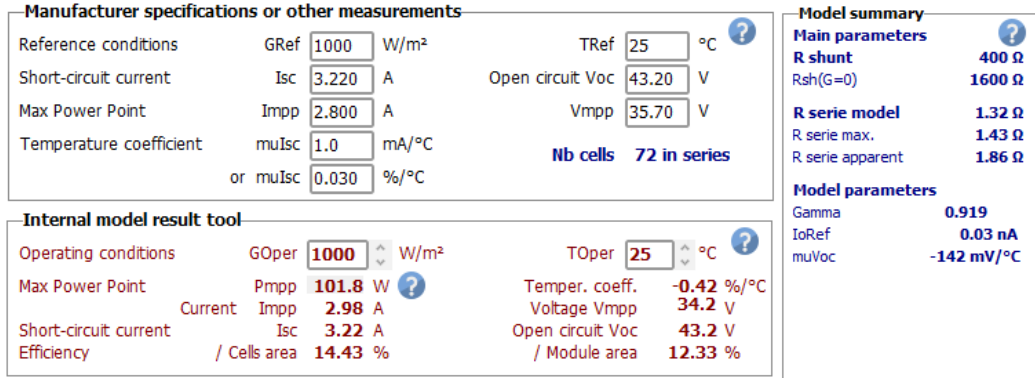


Figura 3.3 – Dados do Painel Fotovoltaico ASE-100-DG-UR recolhidos do *software PVsyst* [73].

Com os dados do fabricante e os dados recolhidos do *software PVsyst*, foi possível simular o sistema por recurso às ferramentas *MATLAB-SIMULINK*. A primeira simulação teve o propósito de simular o modo I de funcionamento do sistema aplicando o método de controlo pretendido inicialmente (ver Figura 3.4). Este método de controlo é apresentado no subcapítulo 3.5.

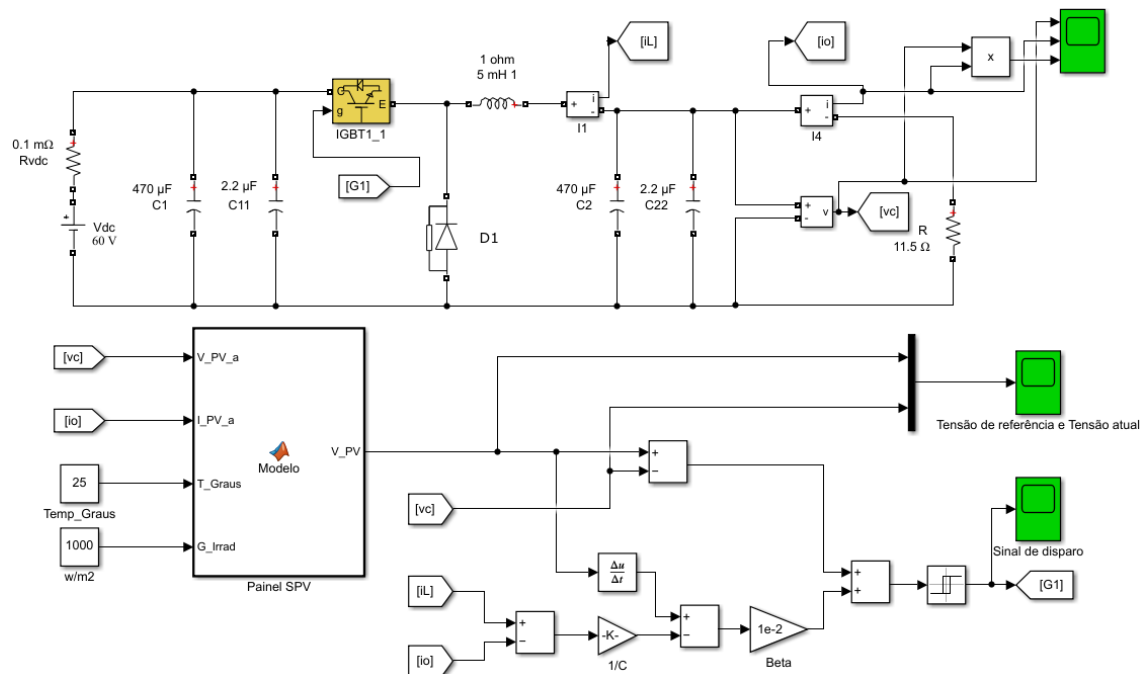


Figura 3.4 – Esquema da simulação da solução idealizada do painel *SPV* na ferramenta *SIMULINK*.

Foram recolhidos dados das grandezas elétricas do circuito para várias situações de funcionamento. Na Figura 3.5 é possível observar o painel *SPV* a funcionar próximo do ponto de máxima potência.

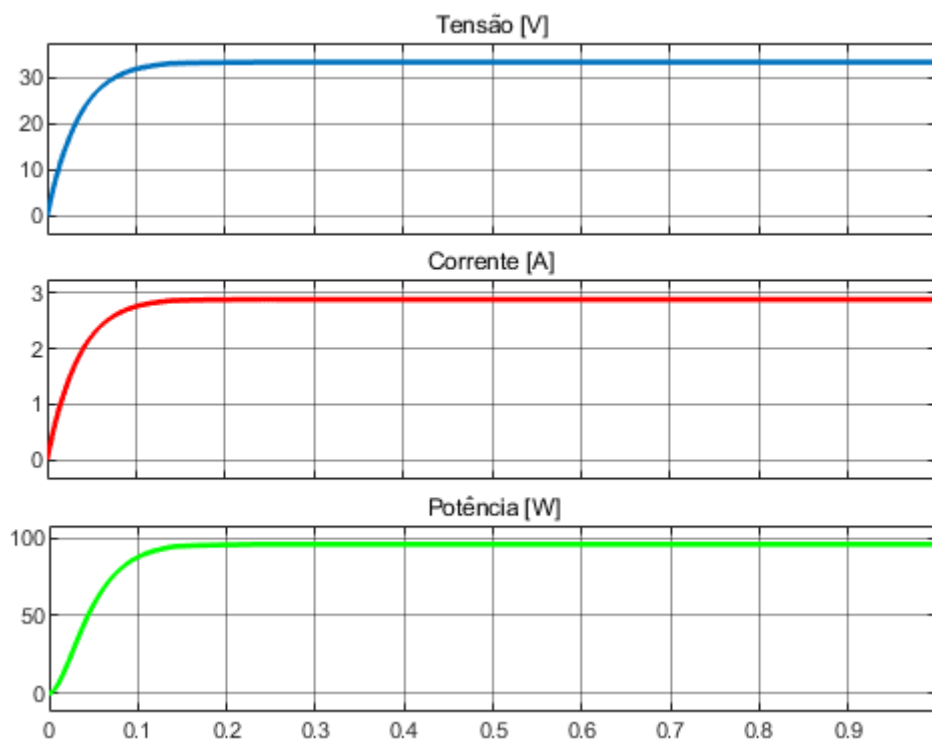


Figura 3.5 – Grandezas elétricas da simulação da solução idealizada do painel *SPV* na ferramenta *SIMULINK*.

Partindo deste ponto de funcionamento, foram também simuladas algumas alterações de irradiância incidente e temperatura, que podem ser observadas respetivamente na Figura 3.6 e Figura 3.7.

Na Figura 3.6 a irradiância incidente inicialmente a  $1000 \text{ W/m}^2$  altera para  $600 \text{ W/m}^2$  aos 0,33 segundos e aos 0,66 segundos altera para  $400 \text{ W/m}^2$ .

Na Figura 3.7 a temperatura inicialmente a  $25^\circ\text{C}$  aumenta para  $40^\circ\text{C}$  aos 0,33 segundos e seguidamente aos 0,66 segundos aumenta para  $70^\circ\text{C}$ .

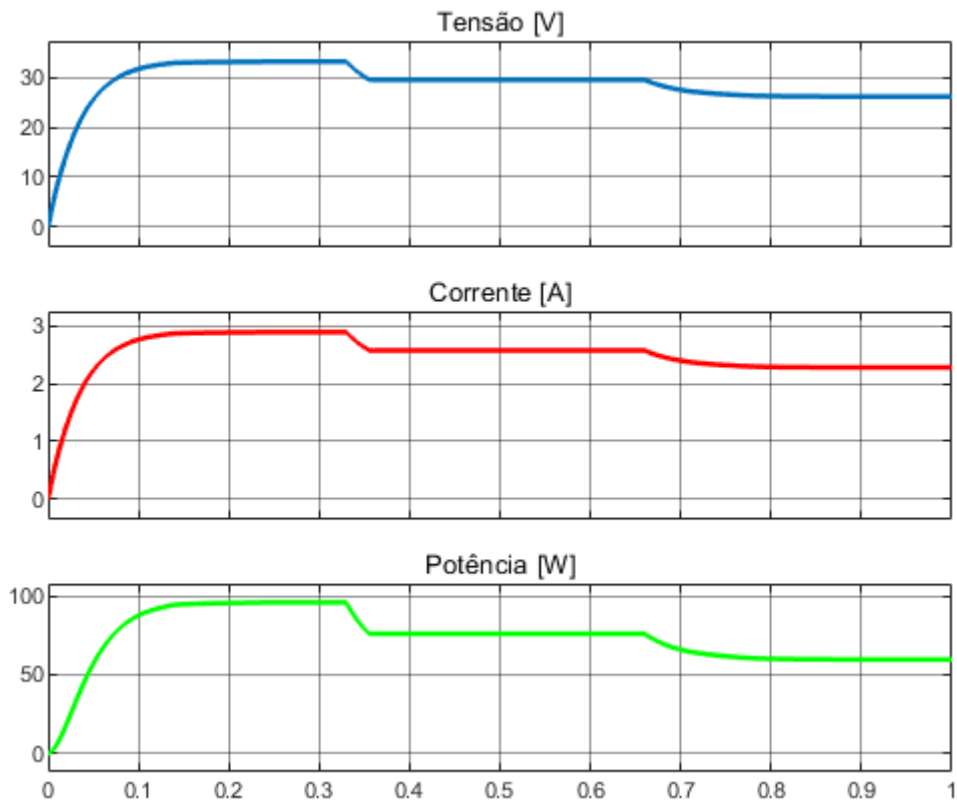


Figura 3.6 – Variação da irradiância incidente da simulação da solução idealizada do painel *SPV* na ferramenta *SIMULINK*.

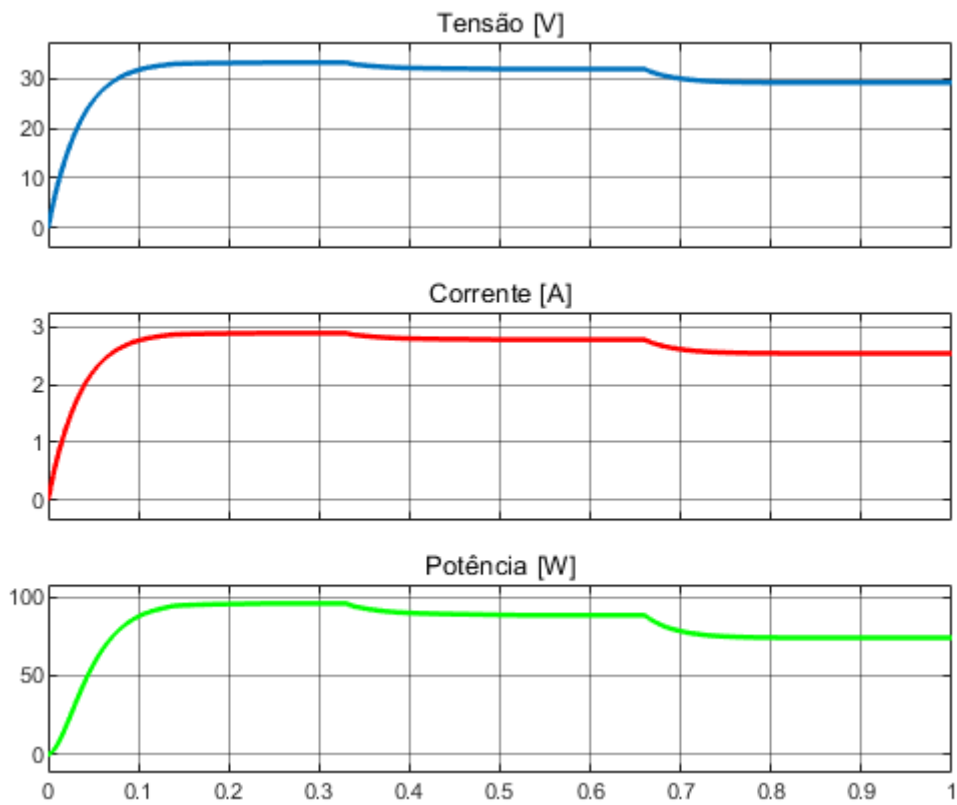


Figura 3.7 – Variação da temperatura da simulação da solução idealizada do painel *SPV* na ferramenta *SIMULINK*.

Esta solução idealizada apresentava bons resultados em simulação, porém não foi possível a sua implementação na prática devido à variação rápida das correntes medidas e ao comportamento da montagem derivativa que tinha resultados longe dos esperados e simulados. A instabilidade provocada pelos comportamentos descritos danificou alguns componentes e optou-se por seguir um método de controlo alternativo (ver Figura 3.8).

Seguindo as equações do modelo de 4 parâmetros, a partir da tensão e corrente num dado instante e respetivas condições de temperatura e iluminância incidente, é imposta uma nova tensão para o instante seguinte. Este sinal de tensão calculado é em seguida comparado com a tensão atual para gerar o sinal de disparo aplicado ao semicondutor do conversor *CC-CC Buck*.

A simulação da Figura 3.8 foi a implementada neste Trabalho Final de Mestrado.

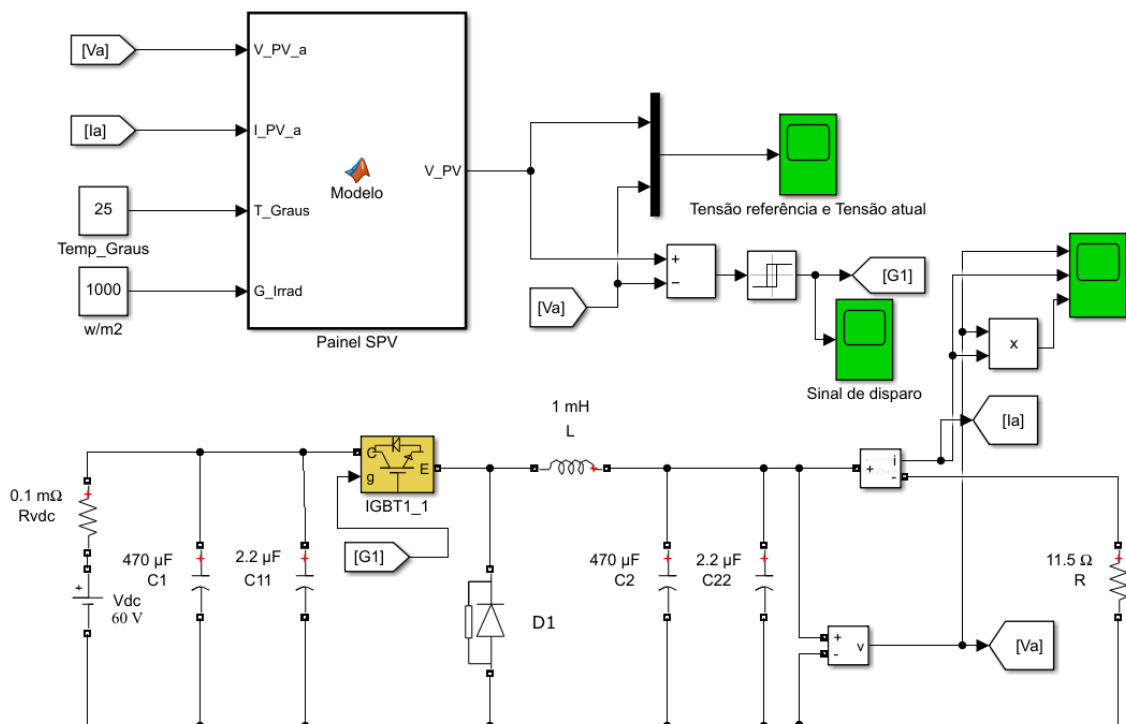


Figura 3.8 – Esquema da simulação do painel *SPV* na ferramenta *SIMULINK*.

Seguem-se os resultados desta simulação, recolhidos de forma idêntica à simulação idealizada anteriormente. Na Figura 3.9 é possível observar o painel *SPV* a funcionar próximo do ponto de máxima potência. Neste caso, como a simulação apresenta uma resposta mais rápida às variações impostas, o tempo de simulação foi de 0,2 s.

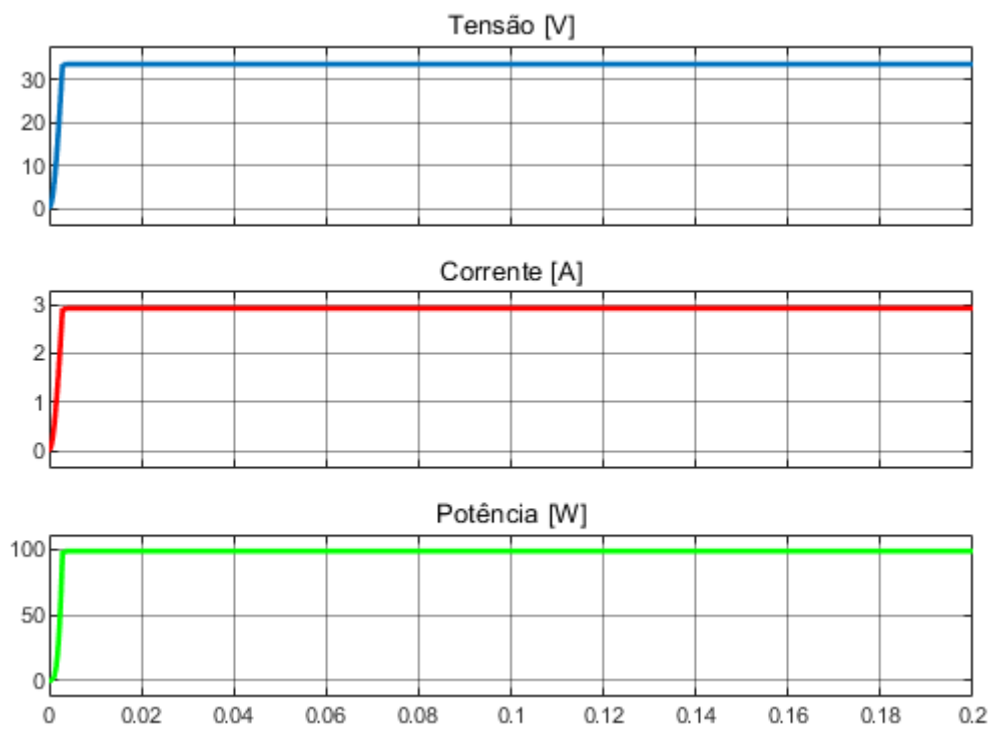


Figura 3.9 – Grandezas elétricas da simulação do painel *SPV* na ferramenta *SIMULINK*.

Tal como na simulação anterior, foram simuladas algumas alterações de irradiância incidente e temperatura, que podem ser observadas respetivamente na Figura 3.10 e Figura 3.11.

Na Figura 3.10 a irradiância incidente inicialmente a  $1000 \text{ W/m}^2$  altera para  $600 \text{ W/m}^2$  aos 0,06 segundos e aos 0,13 segundos altera para  $400 \text{ W/m}^2$ .

Na Figura 3.11 a temperatura inicialmente a  $25^\circ\text{C}$  aumenta para  $40^\circ\text{C}$  aos 0,06 segundos e seguidamente aos 0,13 segundos aumenta para  $70^\circ\text{C}$ .

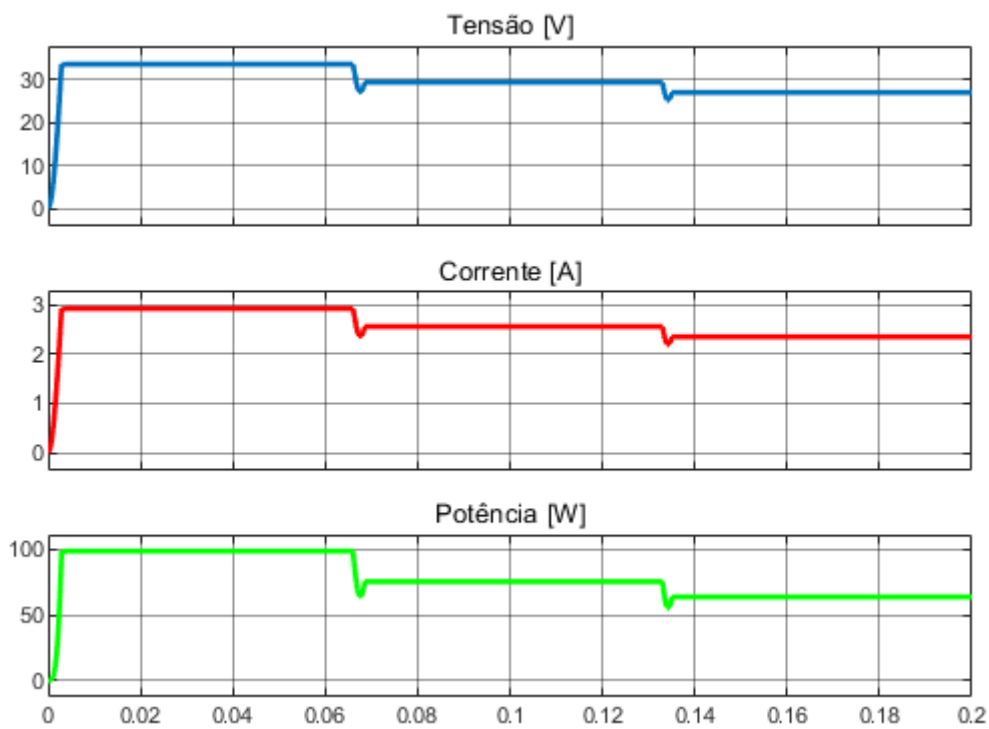


Figura 3.10 – Variação da irradiância incidente da simulação do painel *SPV* na ferramenta *SIMULINK*.

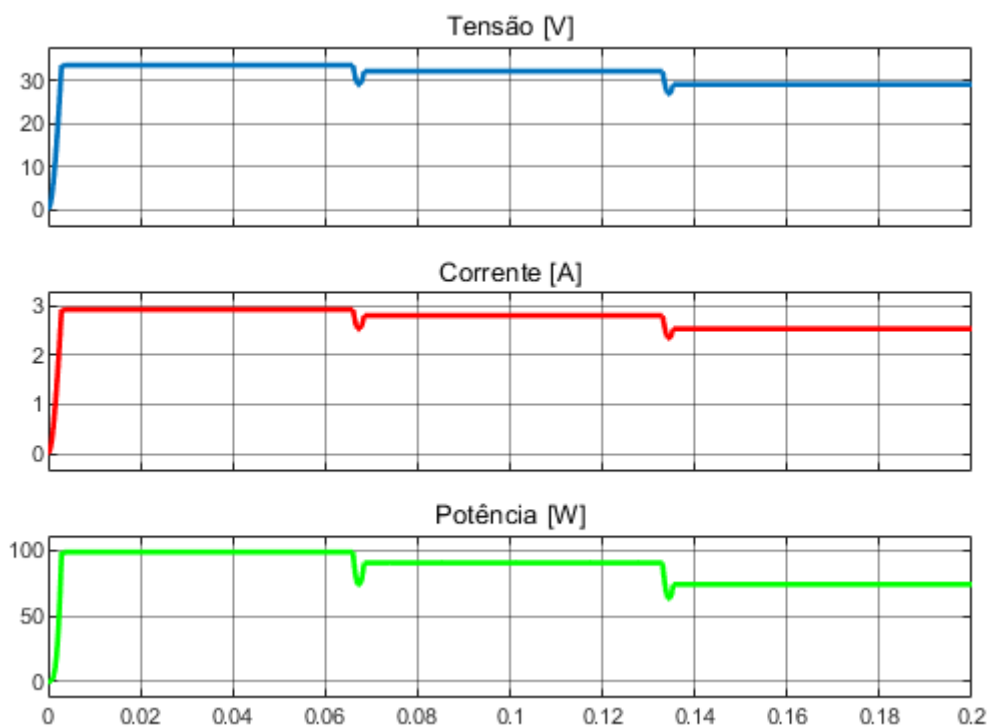


Figura 3.11 – Variação da temperatura da simulação do painel *SPV* na ferramenta *SIMULINK*.

Apesar do método de controlo ser mais simplista, verifica-se que a resposta do sistema é mais rápida que a verificada na anterior simulação. Apesar da pequena oscilação de valores quando existem variações dos parâmetros ambientais, logo de seguida a resposta torna-se estável, o que leva a concluir que a simulação corresponde

ao pretendido, pois permite demonstrar o comportamento do sistema em vários cenários reais.

De acordo com os dados disponíveis, desta vez, por recurso à ferramenta MATLAB, foi possível traçar as curvas características I-V e P-V para diferentes valores de irradiância incidente, assim como para diferentes valores de temperatura de funcionamento.

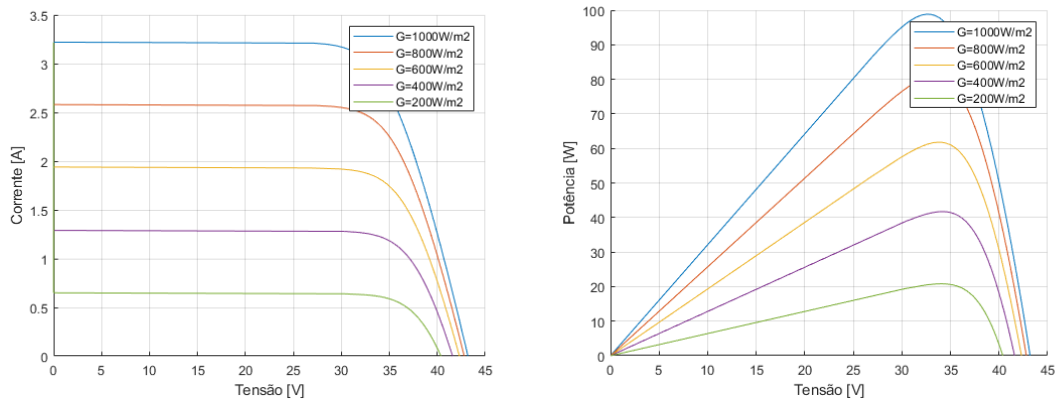


Figura 3.12 – Curvas características I-V (esquerda) e P-V (direita) para diferentes valores de irradiância incidente.

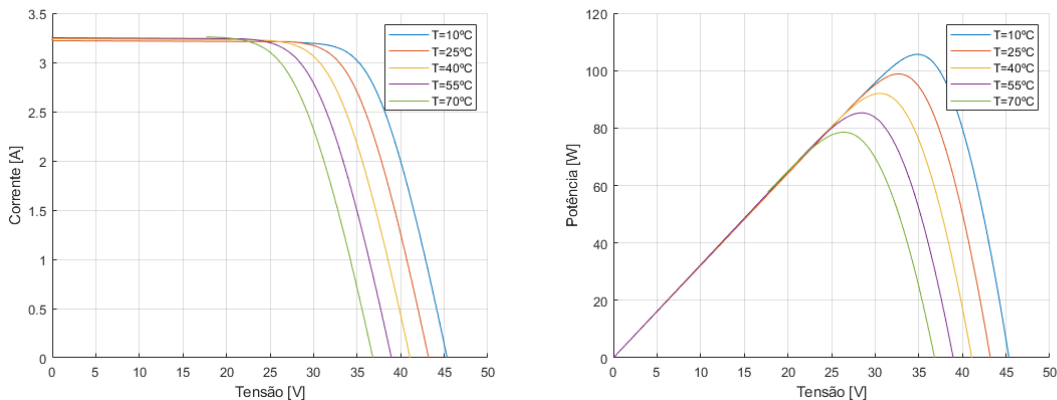


Figura 3.13 – Curvas características I-V (esquerda) e P-V (direita) para diferentes valores de temperatura.

A seguinte simulação na ferramenta *SIMULINK* teve o propósito de simular o comportamento do sistema no modo II (ver Figura 3.14). Estão presentes dois conversores CC-CC, um a funcionar como *Buck* e o outro a funcionar como *Boost*. O primeiro, tal como na primeira simulação, simula o comportamento do painel *SPV* e o segundo simula o funcionamento do algoritmo *MPPT*. Aos terminais do conversor *Boost*, como carga, está presente uma resistência fixa e duas resistências adicionais que são ligadas sequencialmente de forma a simular variações de carga.

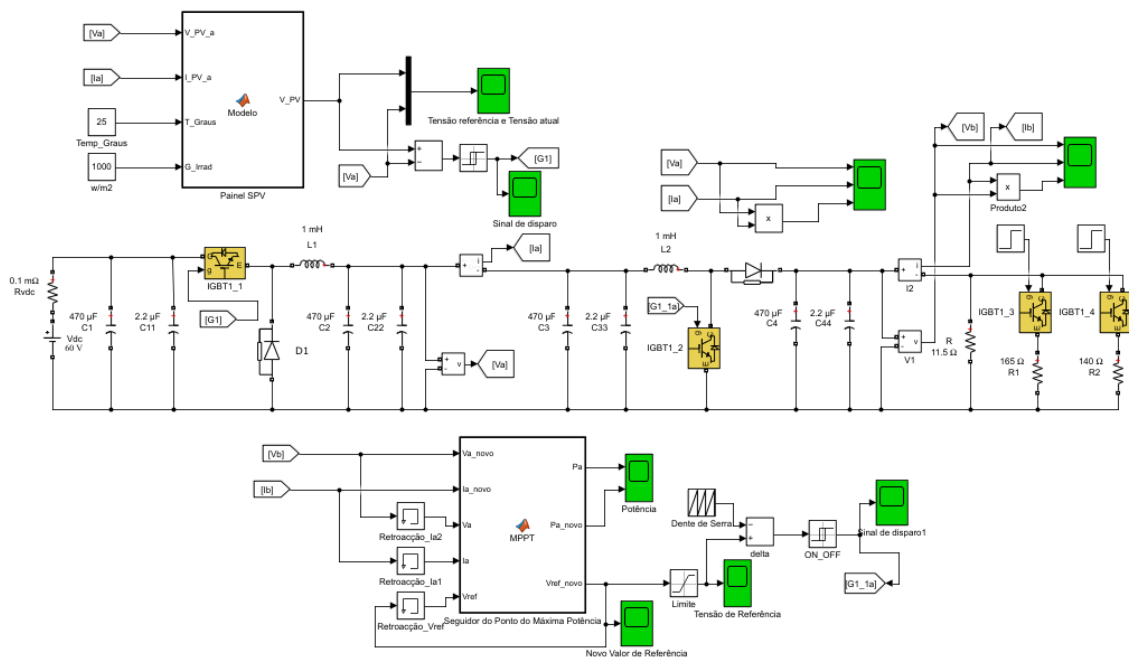


Figura 3.14 – Esquema da simulação do modo *MPPT* na ferramenta *SIMULINK*.

Com esta simulação verifica-se que após as variações de carga nos instantes 0,33 s e 0,66 s (Figura 3.15), graças ao algoritmo *MPPT*, é fornecida uma potência de cerca de 98 W à carga, aproximadamente a potência máxima que o painel pode fornecer.

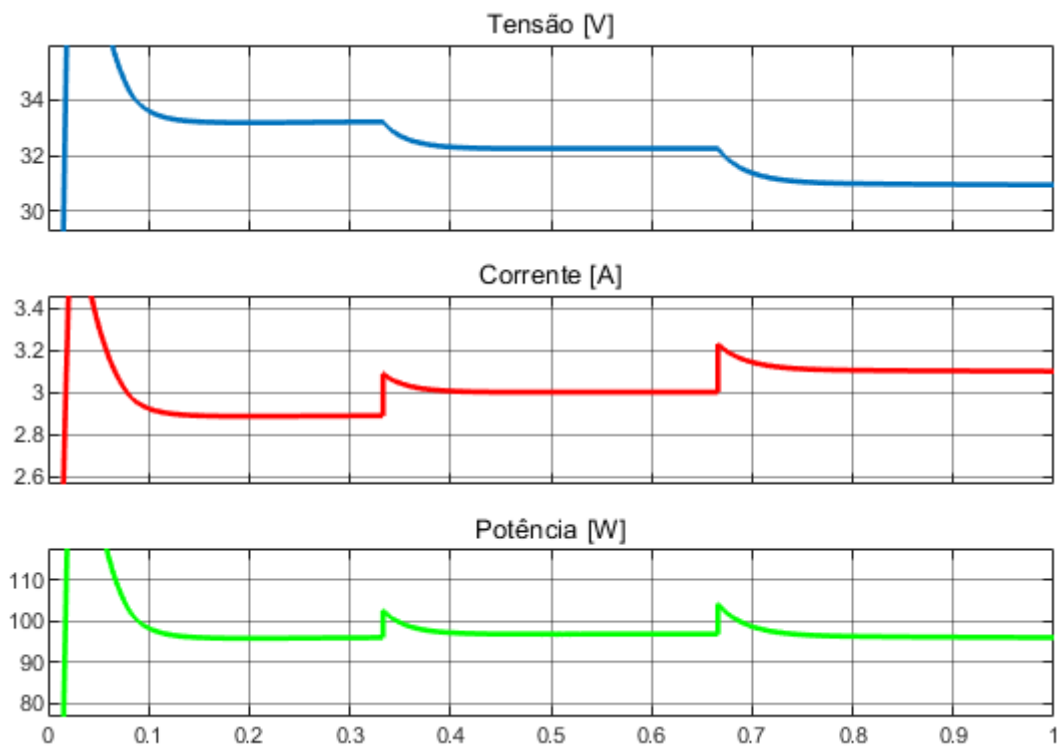


Figura 3.15 – Variação de carga da simulação do modo *MPPT* na ferramenta *SIMULINK*.

### 3.4. Conversores CC-CC

Para a representação das grandezas elétricas do painel fotovoltaico, do conversor de potência associado e da sua carga foi usado um conversor CC-CC *Buck-Boost* clássico em placa de circuito impresso, cujo diagrama do circuito de potência se pode ver na figura seguinte.

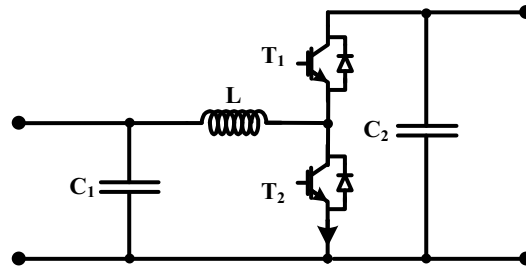


Figura 3.16 – Topologia base do Conversor CC-CC Redutor-Elevador (*Buck-Boost*).

Este conversor permite um funcionamento como conversor redutor (ou *Buck*) ou como conversor elevador (ou *Boost*), dependente da forma como são efetuadas as ligações externas. Ao ligar este conversor com a alimentação à esquerda e a carga à direita (usando  $T_2$  e desligando  $T_1$ ), irá operar em modo *Boost*. Ao trocar as posições respetivas da carga e da alimentação, este opera no modo *Buck* (usando  $T_1$  e desligando  $T_2$ ).

No funcionamento como *Boost*, quando  $T_2$  está a conduzir (Figura 3.17

Figura 3.17 – Esquema de funcionamento do Conversor CC-CC como Elevador (*Boost*): (a) com o semicondutor  $T_2$  à condução; (b) com o semicondutor  $T_2$  ao corte.

a), a bobine é carregada com energia magnética e a energia existente na carga flui através do condensador  $C_2$ . Quando  $T_2$  está ao corte, a energia magnética armazenada na bobine é descarregada na carga e no condensador  $C_2$  através do diodo em anti-paralelo existente em  $T_1$  (Figura 3.17 b). Esta desmagnetização da bobine, carrega o condensador  $C_2$  e provoca um aumento do nível de tensão aos seus terminais, naturalmente aumentando o nível de tensão na carga. A frequência de disparo de  $T_2$  define o tempo que o circuito está em cada modo de funcionamento, este rácio define o ganho de tensão.

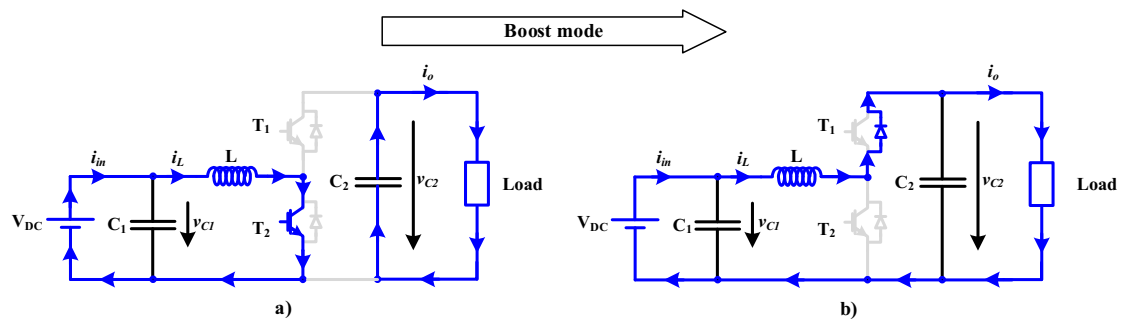


Figura 3.17 – Esquema de funcionamento do Conversor CC-CC como Elevador (*Boost*): (a) com o semicondutor  $T_2$  à condução; (b) com o semicondutor  $T_2$  ao corte.

Ao funcionar como *Buck*, com  $T_1$  à condução (Figura 3.18 c), a energia flui da alimentação para carga através da bobine, mantendo-a carregada com o nível de tensão estabilizado pelo condensador  $C_1$ . Quando  $T_1$  entra ao corte (Figura 3.18 d), a energia magnética armazenada na bobine flui através do díodo em anti-paralelo de  $T_2$ , decrescendo gradualmente o nível de tensão. Neste modo de funcionamento,  $T_1$  está à condução apenas o tempo necessário para carregar a bobine de energia suficiente para manter um nível de tensão mais baixo que o da fonte. Naturalmente, a frequência de disparo de  $T_1$  define o ganho de tensão.

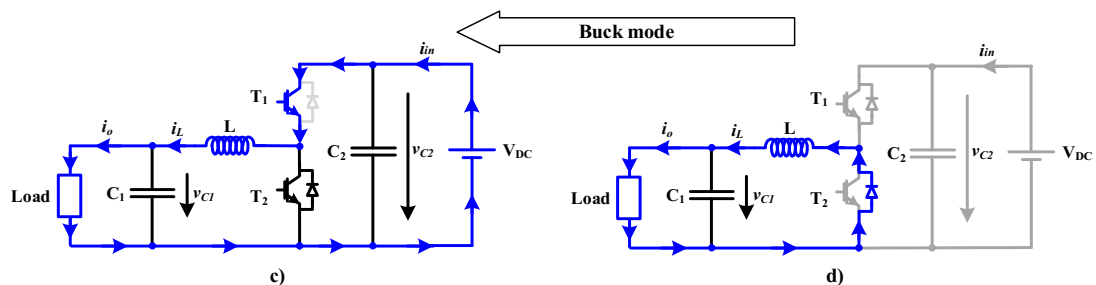


Figura 3.18 – Esquema de funcionamento do Conversor CC-CC como Redutor (*Buck*): (c) com o semicondutor  $T_1$  à condução; (b) com o semicondutor  $T_1$  ao corte.

Para o mesmo propósito podiam ter sido aplicadas outras topologias, com maiores ganhos de tensão e outras características, porém esta foi considerada adequada para este projeto conforme demonstrado pelos resultados experimentais (Capítulo 4).

De seguida apresentam-se os dimensionamentos para os conversores CC-CC considerando os dados do painel fotovoltaico a simular e os diagramas das placas construídas com os elementos aplicados. Com os dimensionamentos dos dois tipos de conversor, é possível escolher elementos que satisfaçam todos os casos e assim optar por elementos comuns na sua construção.

### 3.4.1. Dimensionamento do Conversor CC-CC *Buck*

Esta metodologia de dimensionamento seguiu o procedimento e indicações dados por [74]. Para dimensionar os componentes do conversor a usar, é necessário saber 4 parâmetros:

- Tensão de entrada  $V_{IN(\min)}$  e  $V_{IN(\max)}$ ;
- Tensão de saída nominal  $V_{OUT}$ ;
- Corrente máxima de saída  $I_{OUT(\max)}$ ;
- Semicondutor a aplicar (Wolfspeed, modelo C2M0080120D).

Para efeitos de dimensionamento, considerou-se que a tensão de entrada  $V_{IN(\min)} = V_{IN(\max)} = 60$  V, a tensão de saída  $V_{OUT} = 43,2$  V (tensão em circuito aberto do painel *SPV*) e a corrente máxima de saída  $I_{OUT(\max)} = 3,22$  A (corrente de curto-circuito do painel *SPV*).

O primeiro passo é o cálculo da corrente máxima que atravessa o semicondutor e para tal é necessário conhecer o *duty-cycle* máximo em operação. Na equação seguinte é considerada a eficiência, pelo facto de o conversor ter de fornecer também a energia dissipada, desta forma é obtido um valor de *duty-cycle* mais realista. Estima-se um valor de 90% de eficiência, pois é um valor típico de *worst-case scenario* para conversores CC-CC *Buck*.

$$D_1 = \frac{V_{OUT}}{V_{IN(\max)} \times \eta} = \frac{43,2}{60 \times 0,90} = 0,8 \quad (3.1)$$

Sabendo o *duty-cycle*, podemos calcular o *ripple* da corrente na indutância. Considerou-se o valor da frequência de comutação do semicondutor  $f_S = 60$  kHz e um valor de indutância da bobine  $L = 1$  mH.

$$\Delta I_L = \frac{(V_{IN(\max)} - V_{OUT}) \times D_1}{f_S \times L} = \frac{(60 - 43,2) \times 0,8}{60 \times 10^3 \times 1 \times 10^{-3}} = 0,224 \text{ A} \quad (3.2)$$

Assim é possível determinar se o semicondutor escolhido consegue entregar a corrente necessária para o funcionamento do conversor. Ao verificar o *datasheet* do semicondutor [75], considerou-se que o seu valor de corrente limite  $I_{LIM(\min)} = 24$  A.

$$I_{MAXOUT} = I_{LIM(min)} - \frac{\Delta I_L}{2} = 24 - \frac{0,224}{2} = 23,888 \text{ A} \quad (3.3)$$

Como a corrente a aplicar no circuito é inferior à calculada, pode-se assim determinar a corrente máxima de pico que o semiconductor, a bobine e o díodo terão de suportar:

$$I_{SW(max)} = \frac{\Delta I_L}{2} + I_{OUT(max)} = \frac{0,224}{2} + 3,22 = 3,332 \text{ A} \quad (3.4)$$

Para determinar o díodo a aplicar, deve-se ter em conta três aspetos: a tensão inversa suportada, a corrente média que atravessa o díodo e a sua dissipação de potência.

A tensão inversa suportada pelo díodo é de 1200 V, dada pelo *datasheet* do semiconductor [75]. Os restantes dois aspetos são avaliados pelas equações seguintes, o valor da tensão do díodo à condução ( $V_F$ ) é dado pelo *datasheet* [75].

$$I_F = I_{OUT(max)} \times (1 - D_1) = 3,22 \times (1 - 0,8) = 0,644 \text{ A} \quad (3.5)$$

$$P_D = I_F \times V_F = 0,644 \times 4,3 \approx 2,77 \text{ W} \quad (3.6)$$

Para determinar a capacidade mínima do condensador de saída do conversor, aplica-se a seguinte fórmula, de forma a manter a tensão de saída do conversor com uma oscilação reduzida, considerou-se uma tensão máxima de *ripple*  $\Delta V_{OUT} = 0,01$ :

$$C_{OUT(min)} = \frac{\Delta I_L}{8 \times f_S \times \Delta V_{OUT}} = \frac{0,224}{8 \times 60 \times 10^3 \times 0,01} \approx 47 \mu\text{F} \quad (3.7)$$

Sendo necessário impor um valor máximo de *overshoot* do condensador, é necessário também aplicar a fórmula seguinte. Para este caso, considerou-se o mesmo valor da tensão máxima de *ripple*  $\Delta V_{OUT}$ :  $V_{OS} = 0,01$ . Considerou-se que  $\Delta I_{OUT} = \Delta I_L$ , pois a aplicação em causa quando em funcionamento tem variações relativamente lentas de corrente:

$$C_{OUT(min),OS} = \frac{\Delta I_{OUT}^2 \times L}{2 \times V_{OUT} \times V_{OS}} = \frac{0,224^2 \times 1 \times 10^{-3}}{2 \times 43,2 \times 0,01} \approx 58 \mu\text{F} \quad (3.8)$$

### 3.4.2. Dimensionamento do Conversor CC-CC *Boost*

Analogamente ao dimensionamento anterior, esta metodologia de dimensionamento seguiu o procedimento e indicações dados por [76].

Para dimensionar os componentes do conversor CC-CC *Boost* a usar, de forma análoga ao dimensionamento do conversor CC-CC *Buck*, é necessário saber 4 parâmetros:

- Tensão de entrada  $V_{IN(min)}$  e  $V_{IN(max)}$ ;
- Tensão de saída nominal  $V_{OUT}$ ;
- Corrente máxima de saída  $I_{OUT(max)}$ ;
- Semicondutor a aplicar.

Para efeitos de dimensionamento, considerou-se que a tensão de entrada  $V_{IN(min)} = V_{IN(max)} = 26,7 \text{ V}$  (ponto de operação com tensão mínima, obtido através do MATLAB), a tensão de saída  $V_{OUT} = 33,76 \text{ V}$  (tensão do PMP, obtido através do MATLAB) e a corrente máxima de saída  $I_{OUT(max)} = 3,22 \text{ A}$  (corrente de curto-circuito do painel *SPV*).

Para se realizarem os cálculos do conversor CC-CC a funcionar em modo *Boost* será necessário estabelecerem-se algumas considerações prévias. Considerou-se aqui também uma eficiência de 90%, sendo este o valor típico de *worst case scenario* para conversores CC-CC *Boost*. Note-se que este conversor *Boost* funciona conectado na saída de um conversor *Buck*. Assim, deste modo, podemos considerar que a energia de desmagnetização do indutor do conversor *Buck* (quando o transístor  $T_1$  está ao corte) é igual à energia magnética armazenada pelo indutor do conversor *Boost* (quando o transístor  $T_2$  está à condução), como mostra a equação seguinte. Nesta equação assume-se ainda o mesmo valor de corrente médio genérico para ambas as situações e um indutor semelhante.

$$\frac{1}{2}L(I_{av})^2(1 - D_1) = \frac{1}{2}L(I_{av})^2D_2 \quad (3.9)$$

Da fórmula anterior podemos obter o valor do fator de ciclo para este conversor.

$$(1 - D_1) = D_2 \Rightarrow D_2 = (1 - 0,8) = 0,2 \quad (3.10)$$

O valor da tensão de saída do conversor *Boost* nestas condições é dado pela equação seguinte.

$$V_{OUT} = \frac{V_{IN} \times \eta}{(1 - D_2)} = \frac{26,7 \times 0,9}{(1 - 0,2)} \approx 30 \text{ V} \quad (3.11)$$

Sabendo o *duty-cycle*, podemos calcular o *ripple* da corrente na bobine. Considerou-se o valor da frequência de comutação do semiconductor  $f_s = 100 \text{ kHz}$  e um valor de indutância da bobine  $L = 1 \text{ mH}$ .

$$\Delta I_L = \frac{V_{IN(\min)} \times D_2}{f_s \times L} = \frac{26,7 \times 0,367}{100 \times 10^3 \times 1 \times 10^{-3}} \approx 0,098 \text{ A} \quad (3.12)$$

Assim é possível determinar se o semiconductor escolhido consegue entregar a corrente necessária para o funcionamento do conversor. Ao verificar o *datasheet* do semiconductor, considerou-se que o seu valor de corrente limite  $I_{LIM(\min)} = 24 \text{ A}$ .

$$I_{MAXOUT} = \left( I_{LIM(\min)} - \frac{\Delta I_L}{2} \right) \times (1 - D_2) = \left( 24 - \frac{0,098}{2} \right) \times (1 - 0,367) \quad (3.13)$$

$$\approx 15,16 \text{ A}$$

Como a corrente a aplicar no circuito é inferior à calculada, pode-se assim determinar a corrente máxima de pico que o semiconductor, a bobine e o diodo terão de suportar:

$$I_{SW(\max)} = \frac{\Delta I_L}{2} + \frac{I_{OUT(\max)}}{1 - D_2} = \frac{0,098}{2} + \frac{3,22}{1 - 0,367} \approx 5,136 \text{ A} \quad (3.14)$$

A tensão inversa suportada pelo diodo é de  $1200 \text{ V}$ , dada pelo *datasheet* do semiconductor [75]. Os restantes dois aspetos são avaliados pelas equações seguintes, o valor da tensão do diodo à condução ( $V_F$ ) é dado pelo *datasheet* [75].

$$I_F = I_{OUT(\max)} = 3,22 \text{ A} \quad (3.15)$$

$$P_D = I_F \times V_F = 0,644 \times 4,3 = 2,77 \text{ W} \quad (3.16)$$

Para determinar a capacidade mínima do condensador de saída do conversor, aplica-se a seguinte fórmula, arbitrando para este caso uma tensão máxima de *ripple*  $\Delta V_{OUT} = 0,05$ :

$$C_{OUT(\min)} = \frac{\Delta I_{OUT(\max)} \times D_2}{f_s \times \Delta V_{OUT}} = \frac{3,22 \times 0,367}{100 \times 10^3 \times 0,05} \approx 236,35 \text{ } \mu\text{F} \quad (3.17)$$

Perante os cálculos efetuados para o funcionamento do conversor *Boost* assim como do conversor *Buck*, optou-se pela utilização de condensadores eletrolíticos de 470  $\mu\text{F}$  para todos os conversores. Por questão de uniformização das placas construídas, era o valor de capacidade que estava disponível para colocar em todas sem colocar em causa os limites de *ripple* de tensão de saída considerados.

### 3.4.3. Desenvolvimento do circuito de potência e disparo do conversor

Considerando os dimensionamentos efetuados, os componentes do circuito de potência do conversor foram selecionados de acordo com a Figura 3.19. Foi também elaborado o circuito de disparo dos semicondutores de acordo com a Figura 3.20. Os esquemas apresentados foram elaborados no *software* “KiCad” e posteriormente foram fabricados em placas de circuito impresso. Os componentes necessários foram soldados à placa, assim como os terminais para ligações de entrada e saída de energia, e sinais externos.

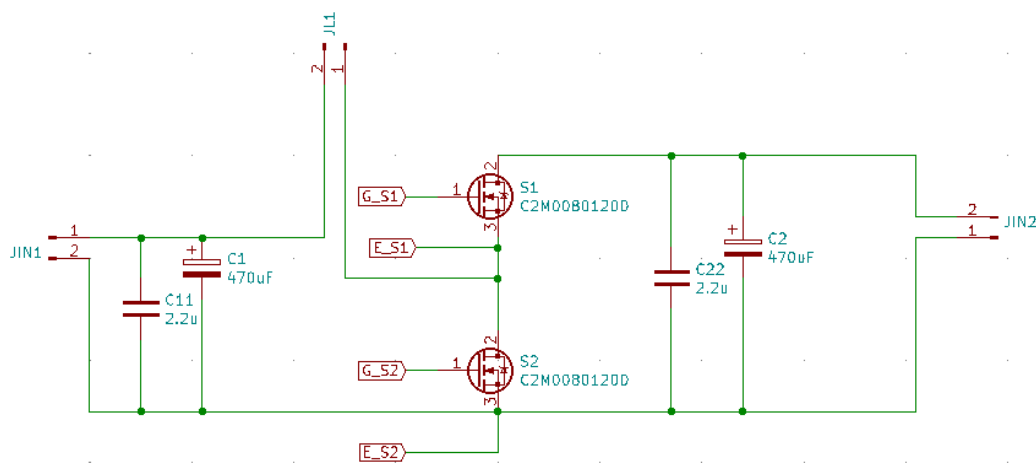


Figura 3.19 – Esquema elétrico do circuito de potência do conversor CC-CC construído.

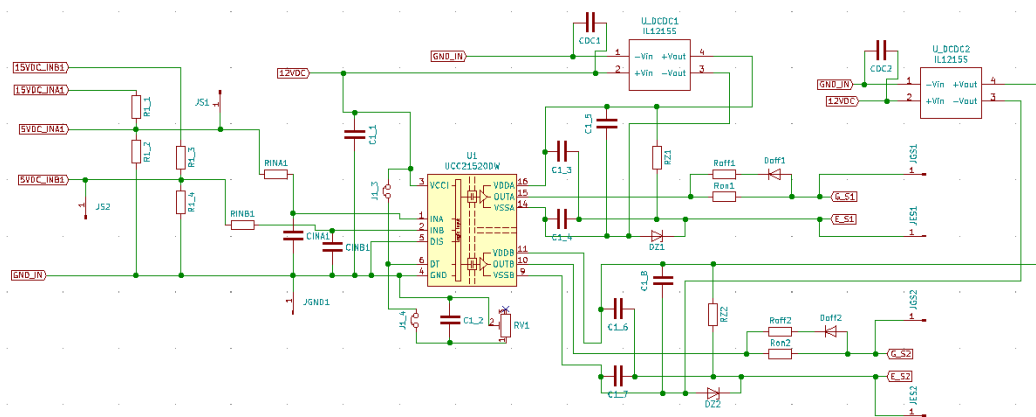


Figura 3.20 – Esquema elétrico do circuito de disparo do conversor CC-CC construído.

No circuito de potência (Figura 3.19) foram aplicados capacitores eletrolíticos (C1 e C2) de capacidade 470  $\mu\text{F}$  para a conversão de energia. Em paralelo com estes foram aplicados capacitores de película (C11 e C22) de capacidade 2,2  $\mu\text{F}$  para filtrar altas frequências parasitas. Como elemento semiconductor foram escolhidos MOSFETs de potência (S1 e S2) da marca Wolfspeed, modelo C2M0080120D,

baseados em Carbetto de Silício. Por motivos de otimização de custos, a bobine usada com este conversor não foi soldada à placa e foram colocados terminais para a sua ligação como um elemento externo (“JL1”), tendo esta um valor aproximado de 1 mH.

O conversor funciona como *Buck* ao controlar o semicondutor “S1” e ao usar os terminais “JIN2” como entrada e os terminais “JIN1” como saída. Neste modo de funcionamento, o semicondutor “S2” fica sem qualquer função e pode ser substituído por um diodo.

Para um funcionamento do conversor como *Boost*, deverá ser controlado o semicondutor “S2” e ser utilizados os terminais “JIN1” como entrada e os terminais “JIN2” como saída. Pelo mesmo motivo do caso anterior, o semicondutor “S1” pode ser substituído por um diodo.

Para o disparo dos semicondutores (Figura 3.20), o conversor é dotado de uma *drive* de disparo com sinais isolados e tempos mortos (marca Texas Instruments, modelo UCC21520DW). Através de divisores de tensão é possível aplicar sinais de disparo à *drive* de 15V para além dos 5V previstos nas suas características. O esquema da *drive* de disparo demonstra a entrada dos sinais nos respetivos terminais até à sua aplicação às *gates* dos semicondutores.

Na Figura 3.21 está demonstrada a placa de circuito impresso construída, com os respetivos componentes escolhidos já soldados e a bobine externa utilizada, ligada aos terminais (“JL1”).

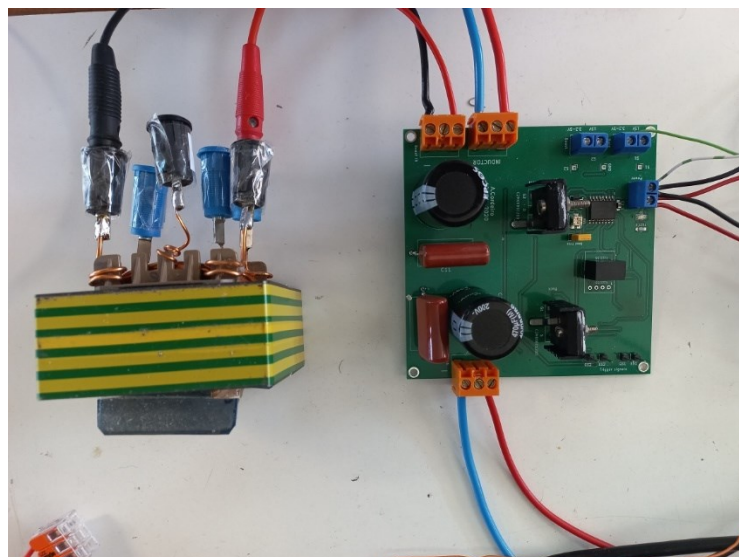


Figura 3.21 – Placa de circuito impresso construída.

### 3.5. Controlo do Painel Fotovoltaico

O conversor CC-CC *Buck* responsável pela simulação da curva I-V do painel fotovoltaico é controlado através de uma técnica de controlo de modo deslizante. Esta técnica apresenta uma rápida resposta dinâmica e robustez face à variação de parâmetros e da carga [77], [78] e foi usada em diversas aplicações de energias renováveis [79], [80]. Para desenhar este controlador, é considerado um modelo dinâmico com componentes ideais, nomeadamente o transístor de potência, a bobine e os condensadores. O estado do semicondutor  $T_1$  da Figura 3.16 é representado pela variável dependente do tempo  $\lambda$ , como descrito pela Equação 3.18.

$$\lambda = \begin{cases} 1, T_1 \rightarrow \text{Turned On} \\ 0, T_1 \rightarrow \text{Turned Off} \end{cases} \quad (3.18)$$

Ao examinar a correspondência entre as tensões e correntes apresentadas na Figura 3.17, é possível obter um modelo de estado simplificado do conversor CC-CC em ordem às variáveis de controlo  $i_L$  e  $v_{C1}$ , Equação 3.19.

$$\begin{cases} \frac{di_L}{dt} = -\frac{1}{L}v_{C1} + \frac{\lambda}{L}v_{DC} \\ \frac{dv_{C1}}{dt} = \frac{1}{C1}i_L - \frac{1}{C1}i_o \end{cases} \quad (3.19)$$

Considerando a tensão do condensador  $v_{C1}$  como a variável de output a controlar, as equações de estado na forma canónica controlável podem ser obtidas pelas Equações 3.20 e 3.21.

$$\frac{d}{dt} \begin{bmatrix} v_{C1} \\ \phi \end{bmatrix} = \begin{bmatrix} \phi \\ -\frac{1}{LC1}v_{C1} - \frac{1}{C1}\frac{di_o}{dt} + \frac{\lambda}{LC1}v_{DC} \end{bmatrix} \quad (3.20)$$

$$\phi = \frac{i_L - i_o}{C1} \quad (3.21)$$

Da equação de estado apresentada em (3.20), é possível analisar que o *output* de corrente tem um grau relativo de dois [78] (pois apenas a segunda derivada contém a variável controlável). Assim, nesta condição, a superfície deslizante definida em (3.22) é suficiente para assegurar a necessária robustez e estabilidade do sistema em cadeia

fechada conforme apresentado em [79]. O parâmetro  $\beta$  definido em (3.22) está relacionado com a constante de tempo da desejada resposta de primeira ordem da tensão a controlar ( $\beta > 0$ ). Baseado em (3.20) e (3.22) é possível definir teoricamente uma superfície deslizante descrita pela equação (3.23).

$$S(e_{v_{C1}}, e_{\phi}) = (v_{C1ref} - v_{C1}) + \beta(\phi_{ref} - \phi) \quad (3.22)$$

$$S(e_{v_{C1}}, e_{\phi}) = (v_{C1ref} - v_{C1}) + \beta \frac{dv_{C1ref}}{dt} - \frac{\beta}{C1}(i_L - i_o) \quad (3.23)$$

Contudo, relativamente ao conversor CC-CC, é necessário impor uma limitação na largura de banda do sinal aplicado à gate devido ao limite da frequência de disparo dos transístores de potência. Em aplicações práticas isto é feito tipicamente por recurso a comparadores histeréticos que mantêm a superfície deslizante próxima de zero com uma frequência de disparo limitada. Em conclusão, a lei de controlo para o transístor de potência T1 é dada pela equação (3.24), onde  $\Delta$  introduz o limite na frequência de disparo.

$$\begin{cases} \text{if } S(e_{v_c}, e_{\theta}) > \Delta \text{ or } [\dot{S}(e_{v_c}, e_{\theta}) < 0 \text{ and } |S(e_{v_c}, e_{\theta})| < \Delta] \text{ then } T_1 \rightarrow \text{Turns On;} \\ \text{if } S(e_{v_c}, e_{\theta}) < -\Delta \text{ or } [\dot{S}(e_{v_c}, e_{\theta}) > 0 \text{ and } |S(e_{v_c}, e_{\theta})| < \Delta] \text{ then } T_1 \rightarrow \text{Turns Off;} \end{cases} \quad (3.24)$$

Esta solução, já referida no subcapítulo 3.3 (Figura 3.4), foi impossível de implementar na prática por diversos problemas com os componentes do circuito de potência. Foi assim aplicada uma solução alternativa baseada em modulação *PWM*.

### 3.6. Transdutores

Para a leitura das grandezas elétricas foram aplicados transdutores de tensão e corrente, em ambos os modos de funcionamento para o processamento dos dados pelo microcontrolador e no modo II para o processamento dos dados pelo *PLC*.

São de seguida apresentados os transdutores aplicados para ambos os casos.

#### 3.6.1. Transdutor de tensão

Os transdutores de tensão foram desenhados no *software* “KiCad” (Figura 3.22 e Figura 3.23) e posteriormente impressos em placas de circuito.

O transdutor de tensão, a ser colocado em paralelo com a tensão a medir, tem múltiplas entradas (“JS1”), dotadas internamente de resistências entre elas, de forma a funcionar como divisor de tensão e permitir escolher a escala de tensão a medir. Nesta cadeia, a resistência “RVA8” é fisicamente um potenciômetro, de modo a permitir regular um “offset” ao sinal. Desta forma o nível de tensão a medir é convertido para a gama 0V-5V e aplicado ao integrado AMC1100, sendo este um amplificador com isolamento galvânico.

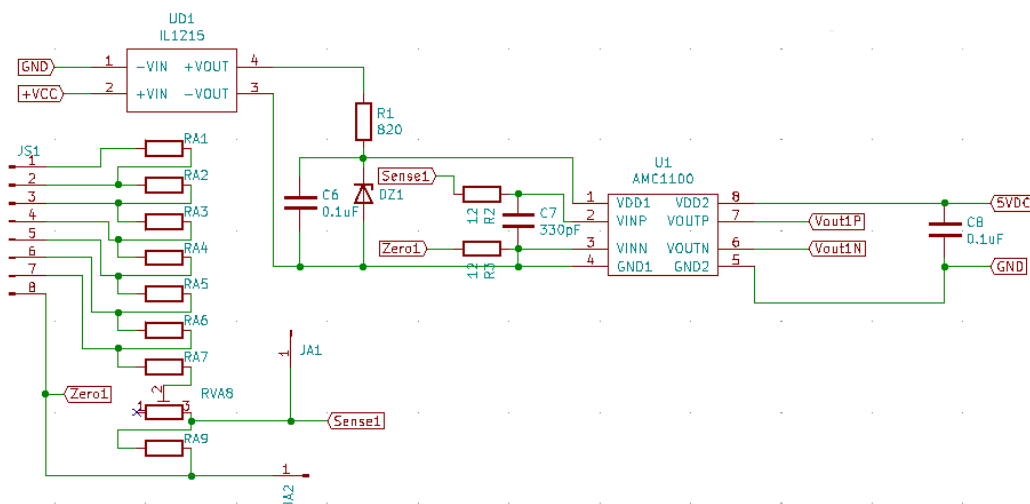


Figura 3.22 – Esquema de aquisição de tensão do transdutor de tensão.

O sinal à saída do amplificador é em seguida aplicado a uma cadeia de amplificadores operacionais (TL084), com o intuito de reduzir ruído e permitir a aplicação de um determinado ganho através da regulação de um potenciômetro. No fim da cadeia do sensor de tensão é possível obter um nível de tensão dentro da gama -Vcc e +Vcc proporcional ao nível de tensão a medir.

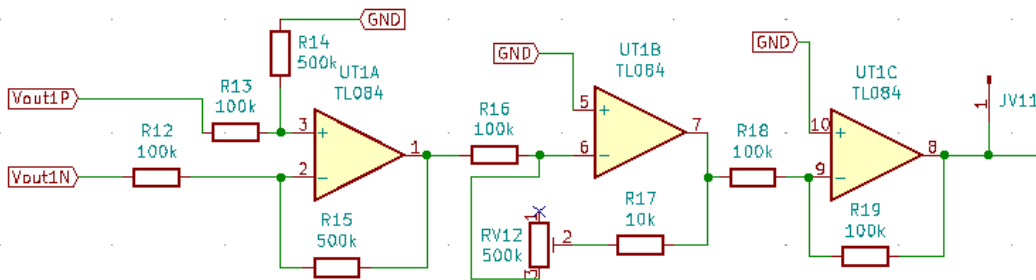


Figura 3.23 – Esquema de tratamento e envio de sinal do transdutor de tensão.

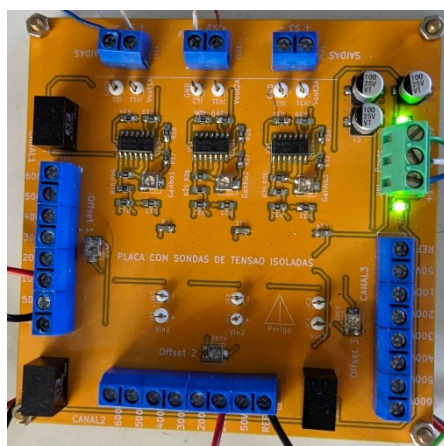


Figura 3.24 – PCB do transdutor de tensão.

### 3.6.2. Transdutor de corrente

O transdutor de corrente aplicado no projeto foi o LEM LA 25-NP, devido à sua simplicidade de aplicação, elevada precisão e linearidade. À sua saída gera um sinal isolado com nível de tensão proporcional à corrente que o atravessa.



Figura 3.25 – Transdutor de corrente LEM LA25-NP.

O transdutor é colocado em série no circuito do qual se pretende medir a corrente e necessita de ser alimentado externamente por uma tensão de  $\pm 15V$  ( $V_c$  - Figura 3.26).

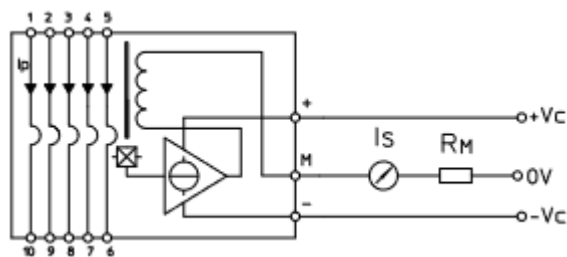


Figura 3.26 – Esquema de ligações do LEM LA25-NP.

Permite a medição de correntes até 25A, com a possibilidade de alteração da sua escala de medição para valores máximos inferiores. De acordo com o seu *datasheet*, as ligações necessárias para uma escala máxima de 5A são as apresentadas na Figura 3.27. A corrente a medir é assim ligada entre os pontos 1 e 6.

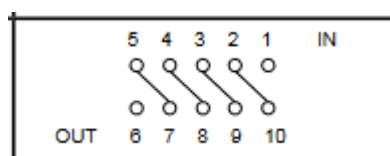


Figura 3.27 – Ligações para uma escala máxima de 5A no LEM LA25-NP.

### 3.6.3. Filtro passa-baixo

As entradas analógicas do microcontrolador não têm qualquer tipo de filtragem de ruído e verificou-se que devido às comutações do sistema os transdutores estavam com problemas nas leituras. Por forma a minimizar o ruído provocado pelas comutações constantes foi implementado um filtro passa-baixo aos sinais provenientes dos transdutores de tensão e corrente que estão ligados ao microcontrolador.

Para dimensionar o filtro a aplicar foi seguida a Equação 3.25.

$$f_c = \frac{1}{2\pi RC} \quad (3.25)$$

Considerando que o ruído elétrico existente provém de comutações a 60 kHz e 100 kHz, considerou-se uma frequência de corte  $f_c = 10$  kHz e foi aplicado um condensador de 2,2 nF. Reescrevendo a Equação 3.25 em ordem a  $R$ , Equação 3.26, é assim possível determinar o valor de resistência a aplicar.

$$R = \frac{1}{2\pi f_c C} \Leftrightarrow R = \frac{1}{2 \times \pi \times 10^4 \times 2,2 \times 10^{-9}} \Leftrightarrow R \approx 7234 \Omega \quad (3.26)$$



## 3.7. Microcontrolador dsPIC30f4011

### 3.7.1. Ligações físicas

As ligações físicas do microcontrolador foram efetuadas conforme observado na Figura 3.28. Segue-se também uma listagem dos pins utilizados e a descrição da sua função.

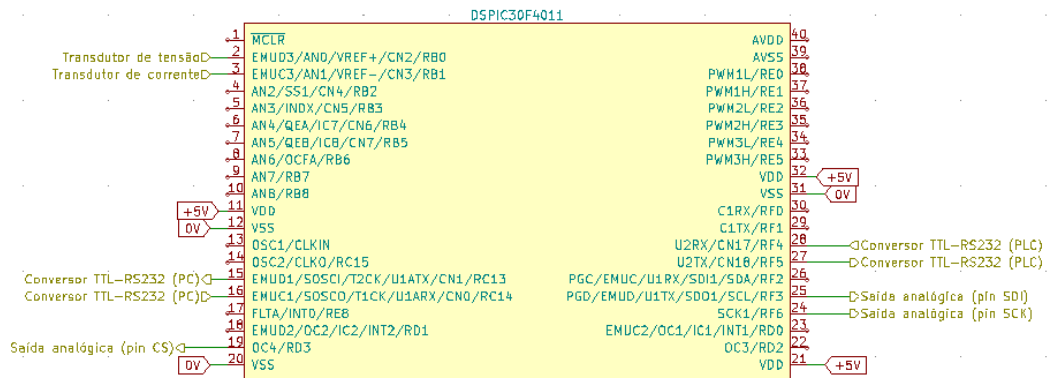


Figura 3.28 – Ligações do microcontrolador *dsPIC30f4011*.

Tabela 3.2 – Descrição das ligações do microcontrolador *dsPIC30f4011*.

Pin #	Função	Descrição
11, 21, 32	Vdd	Alimentação positiva (+5V)
12, 20, 31	Vss	Alimentação negativa (0V)
2	AN0	Entrada analógica do valor medido pelo transdutor de tensão
3	AN1	Entrada analógica do valor medido pelo transdutor de corrente
15	U1ATX	Pin alternativo para transmissão (TX) do módulo UART1 (PC)
16	U1ARX	Pin alternativo para recepção (RX) do módulo UART1 (PC)
19	RD3	Sinal digital para comando da transmissão no protocolo SPI para o DAC
24	SCK1	Sinal de relógio para a transmissão no protocolo SPI para o DAC
25	SDO1	Transmissão de dados no protocolo SPI para o DAC
27	U2TX	Pin para transmissão (TX) do módulo UART2 (PLC)
28	U2RX	Pin para recepção (RX) do módulo UART2 (PLC)

De notar que foram utilizados os pins alternativos para a comunicação *UART1* pois um dos pins normais (pin 25) estava ocupado com a comunicação no protocolo *SPI* com o *DAC*.

### 3.7.2. Programação

A programação do microcontrolador foi construída na linguagem *C* através da ferramenta MPLAB X (Figura 3.29). O programa está segmentado em vários ficheiros:

- *main.c* – contém os bits de configuração base do microcontrolador, inicializa as funções auxiliares contidas nos outros ficheiros, contém o ciclo principal da simulação (lê as entradas analógicas, calcula e envia para o *DAC* o valor de tensão de referência) e inclui as ações a tomar quando recebe dados do *PLC* via *UART2*.
- *analog.c* – contém a função de inicialização das entradas analógicas e a função de leitura das mesmas.
- *analog.h* – contém os protótipos das funções definidas em *analog.c*.
- *comm.c* – contém as funções de inicialização das comunicações *UART* e *SPI*, assim como as funções de interrupção de cada *UART* e a função de escrita através do protocolo *SPI*.
- *comm.h* – contém as definições relativas à velocidade do processamento de instruções e velocidade de comunicação por *UART*, assim como os protótipos das funções definidas em *comm.c*.
- *timer.c* – define um temporizador para enviar a tensão e corrente medidas através das comunicações *UART* de forma constante.
- *timer.h* – contém os protótipos das funções definidas em *timer.c*.

```

1  // DSPIC30F4011 Configuration Bit Settings
2
3  // 'C' source line config statements
4
5  // FOSC
6  #pragma config FPR = FRC_PLL16      // Primary Oscillator Mode (FRC w/ PLL 16x)
7  #pragma config FOS = PRI            // Oscillator Source (Primary Oscillator)
8  #pragma config FCKSMEN = CSW_FSCM_OFF // Clock Switching and Monitor (Sw Disabled, Mon Disabled)
9
10 // FWDI
11 #pragma config FWFSB = WDTFSB_16   // WDT Prescaler B (1:16)
12 #pragma config FWPSA = WDTFSA_512  // WDT Prescaler A (1:512)
13 #pragma config WDT = WDT_OFF        // Watchdog Timer (Disabled)
14
15 // FBORPOR
16 #pragma config FWRTI = FWRTI_64     // POR Timer Value (64ms)
17 #pragma config BODENV = BORV42     // Brown Out Voltage (4.2V)
18 #pragma config BOREN = FBOR_ON      // FBOR Enable (Enabled)
19 #pragma config LPOL = PWMxL_ACT_HI  // Low-side PWM Output Polarity (Active High)
20 #pragma config HPOL = PWMxH_ACT_HI  // High-side PWM Output Polarity (Active High)
21 #pragma config PWMpin = RST_IOPIN   // PWM Output Pin Reset (Control with PORT/TRIS regs)
22 #pragma config MCLR = MCLR_EN       // Master Clear Enable (Enabled)
23
24 // FGS
25 #pragma config GWRP = GWRP_OFF      // General Code Segment Write Protect (Disabled)
26 #pragma config GCP = CODE_PROT_OFF  // General Segment Code Protection (Disabled)
27
28 // FICD
29 #pragma config ICS = ICS_PGD        // Comm Channel Select (Use PGC/EMUC and PGD/EMUD)
30
31 // #pragma config statements should precede project file includes.
32 // Use project enums instead of #define for ON and OFF.
33
34 #include <xc.h>
35 #include <p30F4011.h>
36 #include <math.h>
37 #include <uart.h>
38 #include <stdlib.h>
39 #include "comm.h"
40 #include "analog.h"
41 #include "timer.h"
42
43 float VI, n, VOC_STC, ISC_STC, IO, RS, VOC, ISC, Ri, Rv, dT, T, G, Noell, VPVref, VPV, IPV, VPVrefSV;
44 int VPVrefWord;
45 int DataReceived=0;
46 char buffer[80], rxbuffer[80], txbuffer[80];

```

Figura 3.29 – Projeto na linguagem C construído na ferramenta MPLAB X.

### 3.7.3. Velocidade de funcionamento e comunicação

Por forma a tirar partido do melhor desempenho possível do microcontrolador optou-se por usar o seu oscilador interno e usando a função *PLL* (*Phased-locked Loop*) interna, foi atingida uma velocidade de processamento de instruções muito próxima do seu máximo.

A frequência do oscilador interno é de 7,3728 MHz e o seu fator de multiplicação *PLL* máximo é de 16x. Partindo destes valores, as seguintes equações definem a frequência por instrução e o tempo de processamento de cada instrução.

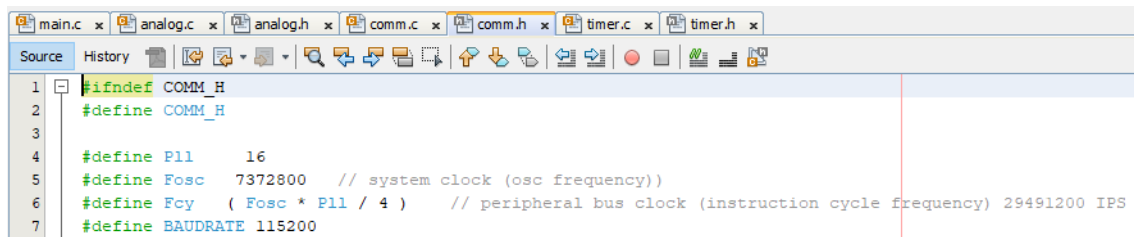
$$F_{cy} = \left( \frac{F_{osc} * PLL}{4} \right) = \left( \frac{7372800 * 16}{4} \right) = 29491200 \text{ Hz} \quad (3.27)$$

$$T_{cy} = \frac{1}{29491200} \approx 33,9 \text{ ns} \quad (3.28)$$

A frequência de comunicação (*Baud Rate*) *UART* escolhida para ambos os canais foi de 115200 bits/s. Para tal é necessário definir os registos U1BRG e U2BRG de acordo com a frequência do oscilador interno.

$$U_{xBRG} = \left( \frac{F_{cy}}{BAUDRATE} \right) - 1 = \left( \frac{29491200}{115200} \right) - 1 = 15 \quad (3.29)$$

Estes registos apenas aceitam valores inteiros, nos casos de arredondamento a velocidade de comunicação terá um erro associado. Verifica-se que usar o oscilador interno é uma boa opção pois o cálculo resulta num valor inteiro. Estas configurações definem a base para as comunicações do microcontrolador (Figura 3.30).



```

1  #ifndef COMM_H
2  #define COMM_H
3
4  #define Pll      16
5  #define Fosc    7372800 // system clock (osc frequency)
6  #define Fcy    ( Fosc * Pll / 4 ) // peripheral bus clock (instruction cycle frequency) 29491200 IPS
7  #define BAUDRATE 115200

```

Figura 3.30 – Configuração da velocidade de *clock* e velocidade de comunicação série do microcontrolador.

Em resumo, as configurações de cada canal *UART* foram as seguintes:

### *UART1 (dsPIC – PC)*

- *Baud Rate*: 115200 kbits/s;
- 8 bits de dados;
- 1 stop bit;
- Sem bit de paridade;
- Utilização de pins alternativos.

### *UART2 (dsPIC – PLC)*

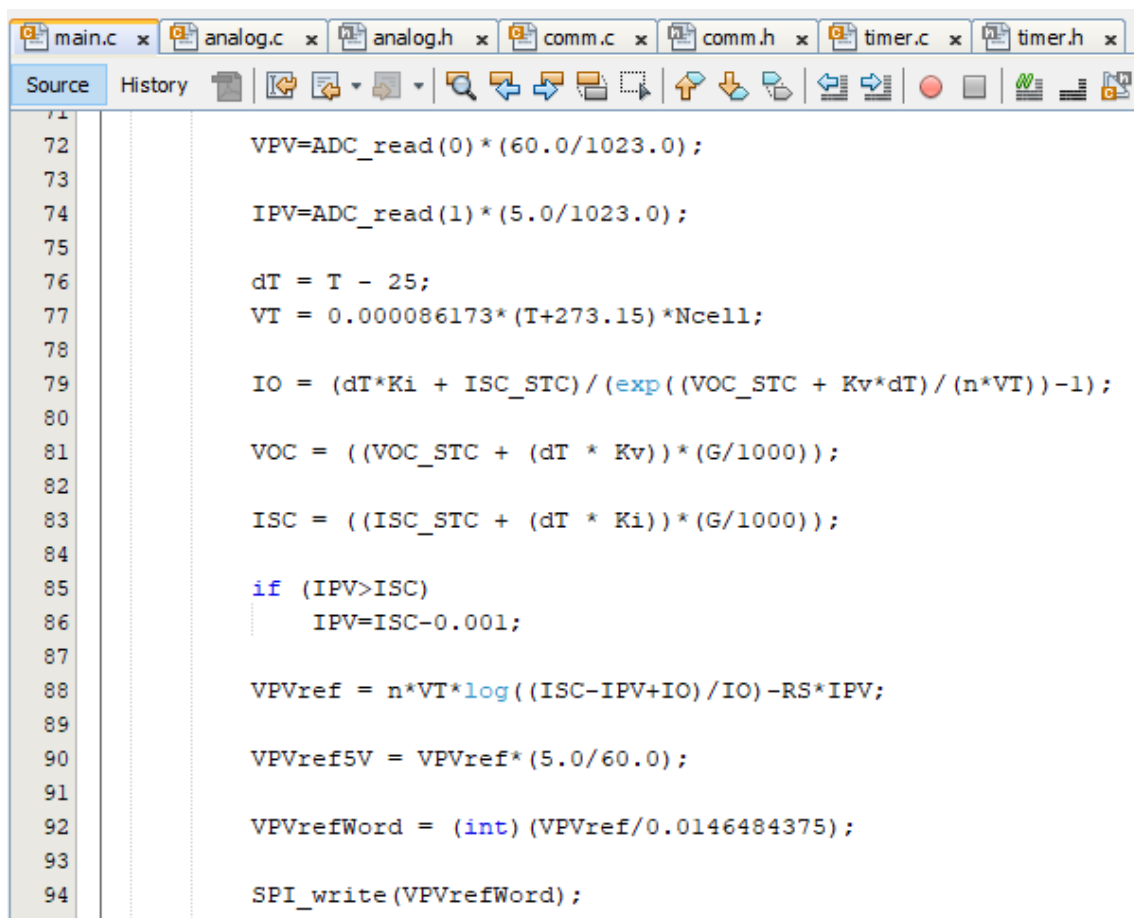
- *Baud Rate*: 115200 kbits/s;
- 8 bits de dados;
- 1 stop bit;
- Bit de paridade *even*.

### 3.7.4. Aplicação do controlo em tensão

O conversor CC-CC *Buck* associado ao microcontrolador é controlado em tensão. Para tal, foi necessária alguma adaptação matemática à equação característica do modelo de 4 parâmetros (Equação 2.21). Reescrevendo a equação, em ordem à tensão, obtém-se a Equação 3.30.

$$V = mV_T \ln\left(\frac{I_S - I + I_o}{I_o}\right) - R_S I \text{ [V]} \quad (3.30)$$

O microcontrolador lê as entradas analógicas com os valores de tensão e corrente a jusante do conversor, aplica a Equação 3.28, define um novo valor de tensão e envia-o para o *DAC* por comunicação *SPI* (Figura 3.31). Por sua vez, através de um amplificador operacional com função de comparador (TL084), ao comparar esta tensão de referência e a tensão atual e posteriormente com uma onda triangular de 60 kHz, é gerado um sinal de disparo para o semicondutor, de forma a variar o seu *duty cycle* e levar a tensão de saída para o valor de referência.



```
71
72     VPV=ADC_read(0)*(60.0/1023.0);
73
74     IPV=ADC_read(1)*(5.0/1023.0);
75
76     dT = T - 25;
77     VT = 0.000086173*(T+273.15)*Ncell;
78
79     IO = (dT*Ki + ISC_STC)/(exp((VOC_STC + Kv*dT)/(n*VT))-1);
80
81     VOC = ((VOC_STC + (dT * Kv))*(G/1000));
82
83     ISC = ((ISC_STC + (dT * Ki))*(G/1000));
84
85     if (IPV>ISC)
86         IPV=ISC-0.001;
87
88     VPVref = n*VT*log((ISC-IPV+IO)/IO)-RS*IPV;
89
90     VPVref5V = VPVref*(5.0/60.0);
91
92     VPVrefWord = (int)(VPVref/0.0146484375);
93
94     SPI_write(VPVrefWord);
```

Figura 3.31 – Equações do controlo em tensão aplicado pelo microcontrolador.



## 3.8. PLC Phoenix Contact ILC 131

### 3.8.1. Ligações físicas

O PLC utilizado foi o ILC 131 ETH da Phoenix Contact, com as seguintes cartas adicionais:

- IB IL 24 DI 8/HD-PAC
- IB IL 24 DI 8/HD-PAC
- IB IL 24 DO 8/HD-PAC
- IB IL 24 DO 4-ME
- IB IL AO 2/UI-PAC
- IB IL AI 2/SF-ME

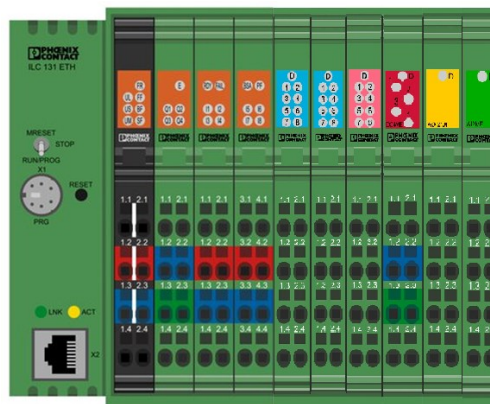


Figura 3.32 – Configuração física do PLC Phoenix Contact ILC 131 ETH.

Verificou-se que apesar de todas as cartas aplicadas, apenas se verificou necessária a utilização da porta DIN-8 para a comunicação *RS-232* com o microcontrolador, a carta de entradas analógicas (IB IL AI 2/SF-ME) para receber as leituras de tensão e corrente (Figura 3.33), e a carta de saídas analógicas (IB IL AO 2/UI-PAC) para enviar os sinais de disparo dos semicondutores dos conversores *CC-CC Boost (MPPT)* e *Buck* (controle da carga) (Figura 3.34).

Tal como mencionado no subcapítulo 3.6, as leituras de tensão e corrente provenientes dos transdutores para a carta de entradas analógicas foram dadas por sinais de tensão proporcionais, daí a representação da grandeza elétrica de ambos os casos ser um nível de tensão na Figura 3.33.

A ligação através da porta DIN-8 para comunicação *RS-232* é possível através da adaptação de um cabo conforme observado na Figura 3.35. A ficha DB9 na outra ponta liga a um conversor *TTL-RS232* para permitir a comunicação com o microcontrolador.

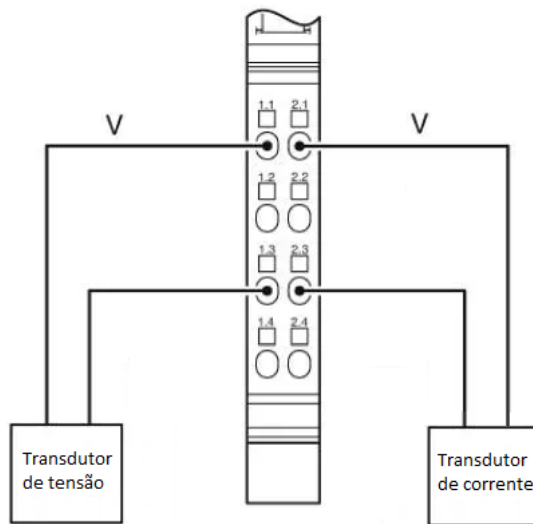


Figura 3.33 – Ligações à carta de entradas analógicas.

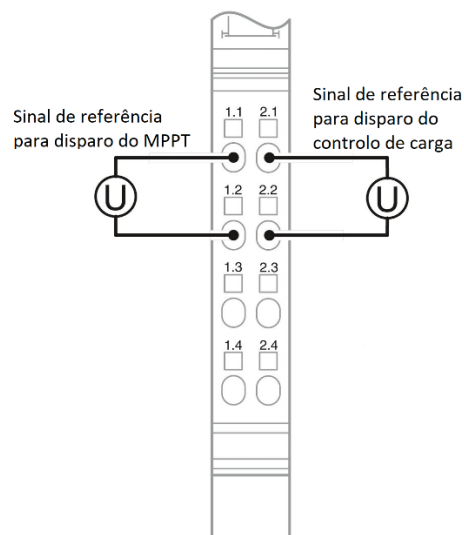


Figura 3.34 – Ligações à carta de saídas analógicas.

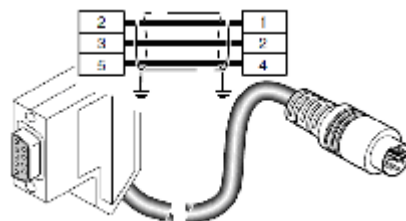


Figura 3.35 – Cabo para comunicação RS-232 através da porta DIN-8 do PLC.

### 3.8.2. Aplicação como controlador MPPT

O controlo do conversor CC-CC *Boost* é realizado pelo *PLC* através de um algoritmo *MPPT*, método *P&O*.

A cada ciclo do programa, o *PLC* recolhe os valores de tensão e corrente à saída do conversor através das entradas analógicas. Por multiplicação direta obtém a potência. Ao comparar os valores de tensão e potência do ciclo atual com o ciclo anterior, a referência de tensão aumenta ou diminui, de acordo com a Figura 2.32. Esta tensão de referência é imposta por uma saída analógica num amplificador operacional com função de comparador (TL084), para comparação com uma onda triangular de frequência 100 kHz. O resultado da comparação é usado como sinal de disparo para o semiconductor do conversor CC-CC *Boost* de forma a variar a tensão de saída. O ciclo principal do programa pode ser observado na Figura 3.36.

```

8
9      (*% Inicio do Algoritmo P&O %*)
10
11     IF deltaP > 0.0 AND deltaV > 0.0 THEN
12     Vref := Vref + C;
13     END_IF;
14
15     IF deltaP > 0.0 AND deltaV < 0.0 THEN
16     Vref := Vref - C;
17     END_IF;
18
19     IF deltaP < 0.0 AND deltaV > 0.0 THEN
20     Vref := Vref - C;
21     END_IF;
22
23     IF deltaP < 0.0 AND deltaV < 0.0 THEN
24     Vref := Vref + C;
25     END_IF;
26
27     IF deltaP = 0.0 THEN
28     Vref := Vref;
29     END_IF;
30
31     IF Vref < 0.0 THEN
32     Vref := 0.0;
33     END_IF;
34
35     IF Vref > 1.0 THEN
36     Vref := 1.0;
37     END_IF;
38
39     Va := V;
40     Pa := P;
41

```

Figura 3.36 – Sequência de instruções do algoritmo *MPPT* aplicado pelo *PLC*.

### 3.8.3. Aplicação como regulador da carga

O último conversor CC-CC da cadeia funciona como *Buck* e tem como único propósito a regulação da potência da carga.

Através de uma barra deslizante e de uma caixa de texto na interface homem-máquina (*HMI*) é possível alterar o valor da tensão de referência para o controlo da carga. Este sinal, de forma análoga aos anteriores, é comparado num amplificador operacional (TL084) com uma onda triangular de frequência 100 kHz e o resultado da

comparação serve como sinal de disparo para o semicondutor do conversor CC-CC *Buck*.

### 3.8.4. Comunicação *RS-232-C*

Conforme mencionado no subcapítulo 3.8.1, a ligação através da porta DIN-8 permite a implementação do protocolo *RS-232-C* para comunicação série. Esta comunicação é configurada internamente por recurso ao bloco “RS232\_INIT” (Figura 3.37). Nesta configuração verificou-se a particularidade de que o *PLC* permite apenas que o bit de paridade seja *even*, excluindo assim a opção *odd* ou mesmo de não o aplicar.

Em resumo, as configurações da configuração foram as seguintes:

- *Baud Rate*: 115200 kbits/s;
- 8 bits de dados;
- 1 stop bit;
- Bit de paridade *even*.

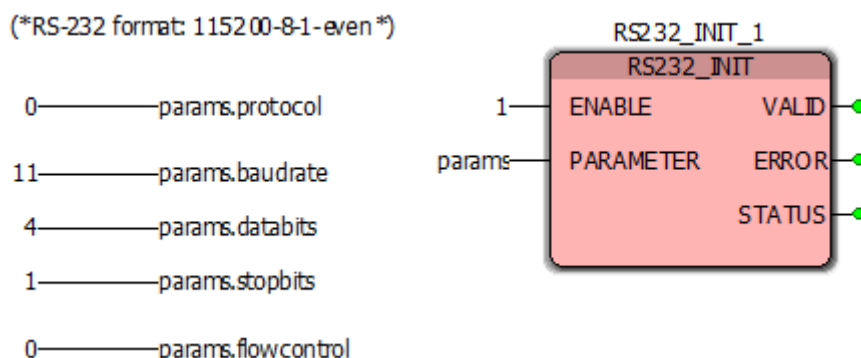


Figura 3.37 – Bloco de configuração da comunicação *RS232* do *PLC*.

## 3.9. Interface homem-máquina (*HMI*)

A partir do *PLC* foi possível construir uma interface homem-máquina que corre através de um servidor web embutido no próprio dispositivo. O *software* disponibilizado pela Phoenix Contact para a construção deste tipo de interfaces é o *WebVisit*. Porém, neste trabalho foi usada outra ferramenta compatível, o *Scada Editor* da empresa *SpiderControl* por esta apresentar algumas funções adicionais.

Foi assim desenhada uma página na qual são apresentados os valores de medição instantânea para as grandezas elétricas (tensão, corrente e potência) em ambos os controladores (ver Figura 3.38). Estas grandezas são também apresentadas na forma de

gráfico, com possibilidade de gravação de um ficheiro “.csv” para posterior análise de dados. Existem também barras deslizantes e caixas de texto onde é possível regular o sinal de controlo do conversor CC-CC *Boost* e do conversor CC-CC *Buck* associado à regulação da carga.

É também possível escolher entre vários modos de operação do *PLC*. Existe um modo manual para numa fase inicial efetuar testes, regulando os sinais de controlo, sendo este o único que permite alteração ao sinal do conversor CC-CC *Boost*. O modo *MPPT* coloca o *PLC* a funcionar com o algoritmo *MPPT*, ajustando automaticamente o *duty-cycle* do conversor CC-CC *Boost*.

De forma a automatizar a recolha de dados, e tornar o processo mais rápido e preciso, foi criada uma função para alterar progressivamente os valores do sinal de disparo para o conversor CC-CC *Buck* associado à regulação da carga. Desta forma, o sinal incrementa a cada 10 segundos, colocando o sistema a alternar entre vários pontos de funcionamento fixos, e a recolher dados referentes a estes pontos.

Outro modo de operação implementado no *HMI*, foi o modo de baterias. Neste modo o *PLC* impõe e mantém uma tensão definida pelo utilizador à saída do conversor CC-CC *Boost*, de forma a simular que o sistema está a carregar uma bateria ou um módulo de baterias. Sendo esta uma prática comum em sistemas de produção de energia *SPV* e de fácil implementação no *PLC* considerou-se que seria uma boa adição ao projeto. Porém, como este modo não era o propósito principal do presente Trabalho Final de Mestrado, não foram retirados resultados com o mesmo. Ficou apenas implementado para futuras demonstrações ou desenvolvimentos.

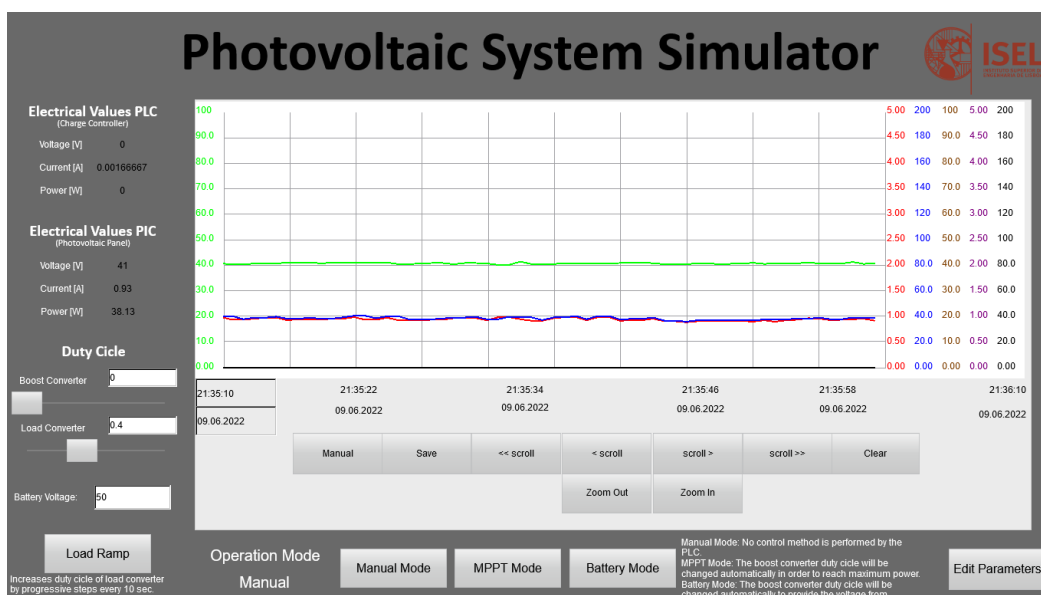
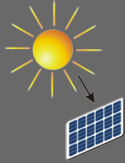


Figura 3.38 – Página principal do *HMI* implementado.

A partir da página principal do *HMI*, é possível aceder a uma outra página que permite ao utilizador alterar os valores das condições ambientais, assim como os parâmetros do painel *SPV* (ver Figura 3.39). A partir desta página é possível interagir com o *dsPIC* de forma a ler os parâmetros atuais em funcionamento, assim como editar esses parâmetros. Este funcionamento deve-se à interação de comunicação *RS-232-C* do *PLC* com a porta *UART* do *dsPIC* em lógica *TTL* através dos conversores *TTL-RS232* implementados.

## Parameter Adjustment



Environment Parameters

Temperature	25	<input type="text" value="25"/>	Write T
Irradiance	1000	<input type="text" value="1000"/>	Write G

Read All

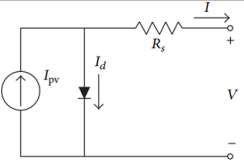
Reads all parameters currently running on the microcontroller.

Copy All

Copies all parameters from static text to text boxes.

Write All

Writes all parameters to the microcontroller.



Photovoltaic Panel Parameters

Voc [STC]	43.2	<input type="text" value="43.2"/>	Write Voc
Isc [STC]	3.22	<input type="text" value="3.22"/>	Write Isc
Series Resistance	1.86	<input type="text" value="1.86"/>	Write Rs
Kv coefficient	-0.142	<input type="text" value="-0.142"/>	Write Kv
Ki coefficient	0.001	<input type="text" value="0.001"/>	Write Ki
Diode ideality factor	0.919	<input type="text" value="0.919"/>	Write N
Number of cells	72	<input type="text" value="72"/>	Write Nce

Main Screen

Figura 3.39 – Página de leitura e edição de parâmetros do *HMI* implementado.

## 4. Resultados

Neste capítulo são demonstrados os resultados obtidos nos vários ensaios efetuados para os dois modos de funcionamento do conversor.

### 4.1. Modo I (Painel *SPV*)

Partindo da configuração da Figura 3.1, o primeiro conversor CC-CC simulou o funcionamento do painel *SPV* com os parâmetros do fabricante demonstrados na Tabela 3.1 e os demais recolhidos no *software* PVsyst, Figura 3.3. A tensão de entrada foi regulada para 60V CC e foi escolhido um ponto de funcionamento do regulador da carga no *software* SCADA, Figura 4.1.

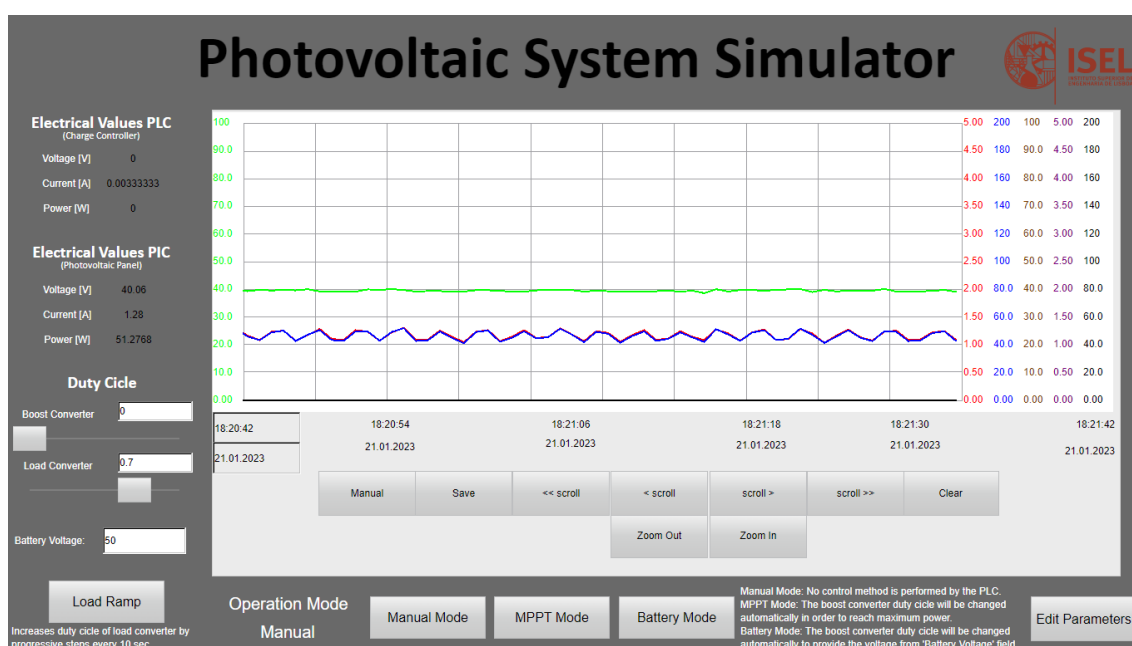


Figura 4.1 – Software SCADA no modo I, conversor CC-CC regulador da carga com duty-cycle de 0,7.

É possível verificar na Figura 4.1 as oscilações nas medições de tensão e corrente, sendo mais notório nesta última, propagando este erro para o cálculo da potência. Estas oscilações são devidas às comutações dos semicondutores no sistema.

Simultaneamente, a comunicação *RS232C* com o *PC* demonstra o estado atual das grandezas elétricas, Figura 4.2.

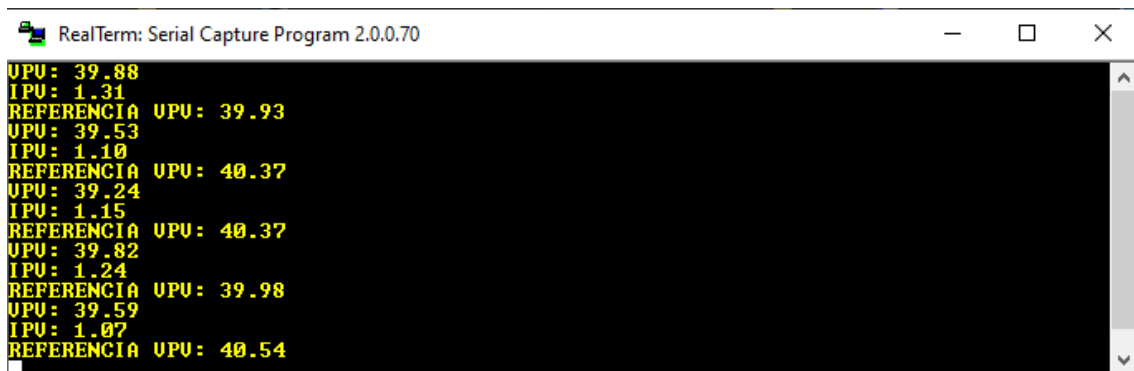


Figura 4.2 – Tensão e corrente enviadas por RS232C para a PC.

Os transdutores de tensão e corrente, após os filtros passa-baixo, registavam as formas de onda demonstradas na Figura 4.3, que, como já reportado, é evidente o ruído elétrico nas medições de corrente.

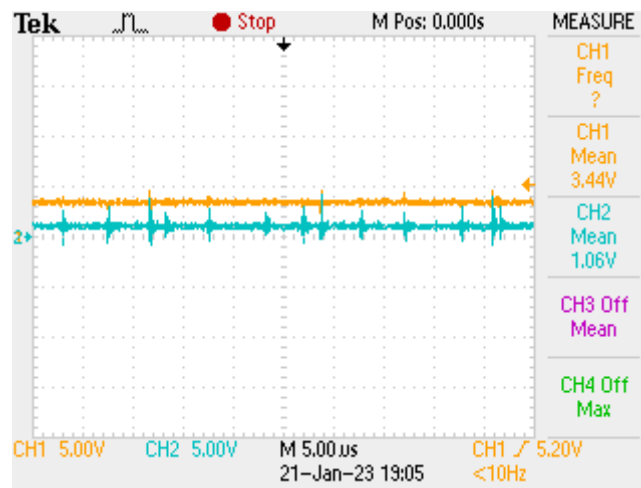


Figura 4.3 – Formas de onda dos transdutores de tensão (CH1) e corrente (CH2).

Através do cálculo efetuado pelo microcontrolador *DSP*, é possível observar a cadeia de controlo da tensão de referência, em primeiro lugar a mensagem enviada para o *DAC* (ver Figura 4.4) e em seguida a comparação do sinal com a onda triangular a 60 kHz que leva à geração do sinal *PWM* da mesma frequência. Verifica-se neste caso, também algum ruído elétrico propagado pelos circuitos do sistema.

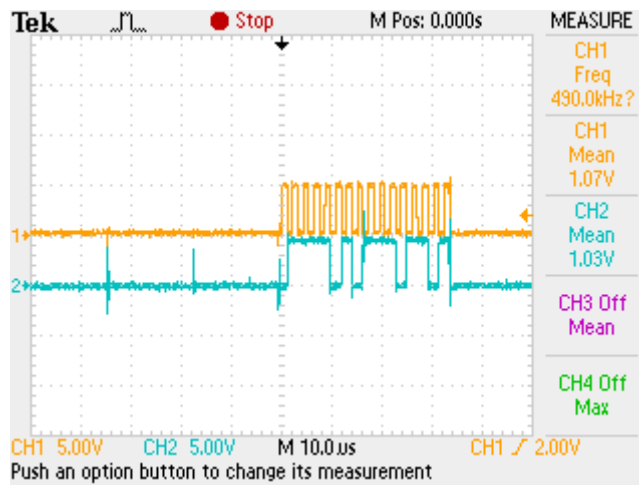


Figura 4.4 – Formas de onda da mensagem enviada para o DAC, pins 24 (SCK1, CH1) e 25 (SDO1, CH2) do microcontrolador.

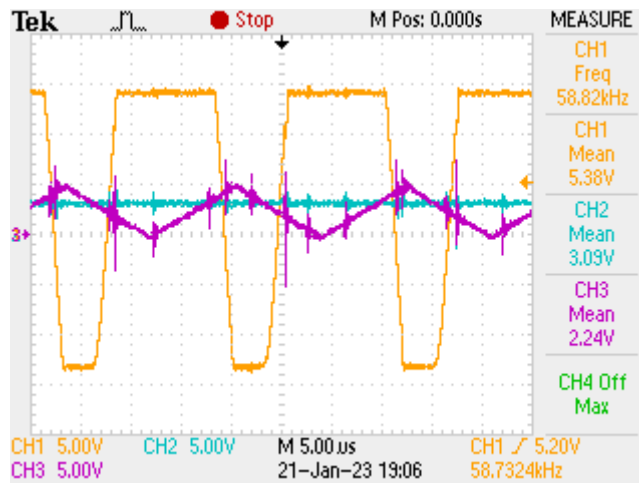


Figura 4.5 – Formas de onda do comparador, CH1: sinal *PWM* gerado como resultado da comparação, CH2: referência de tensão gerada pelo DAC, CH3: onda triangular com frequência 60 kHz.

A bobine do primeiro conversor, responsável pelo armazenamento de energia magnética e posterior descarga ao longo dos ciclos de comutação do *MOSFET* tem o seu comportamento demonstrado na Figura 4.6. De forma a evidenciar melhor o seu funcionamento, na mesma figura está novamente a forma de onda do sinal *PWM* de disparo do *MOSFET*. Como esperado, é perceptível o aumento de corrente na bobine quando o *MOSFET* está à condução e a sua redução quando o *MOSFET* está ao corte.

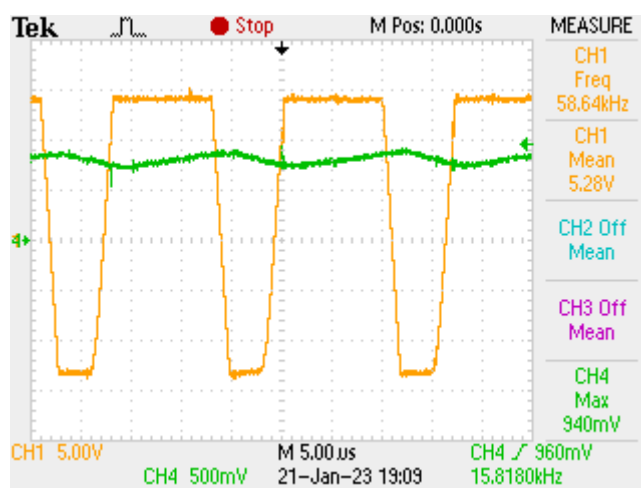


Figura 4.6 – Formas de onda, CH1: sinal *PWM* de comutação do *MOSFET*, CH4: corrente na bobine (100mV/A).

Relativamente ao controlo do regulador de carga por parte do *PLC*, as formas de onda do comparador respetivo podem ser observadas na Figura 4.7. Novamente é evidente alguma perturbação provocada por ruído elétrico, nos sinais de referência e triangular.

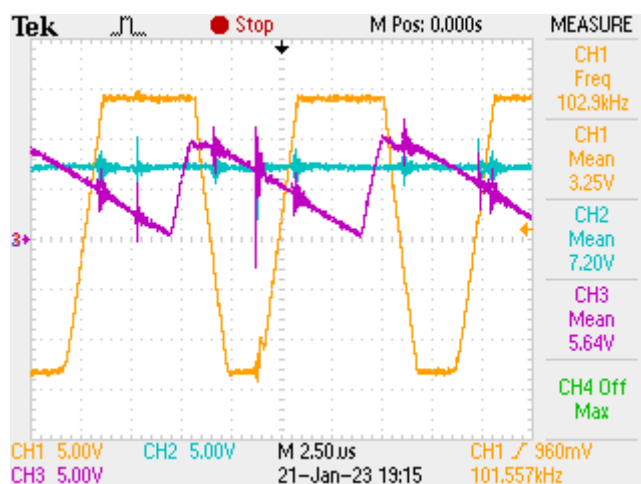


Figura 4.7 – Formas de onda do comparador responsável pela geração do sinal *PWM* do conversor *CC-CC Buck* regulador da carga, CH1: sinal *PWM* gerado, CH2: referência de tensão do *PLC*, CH3: onda triangular com frequência de 100 kHz.

Resta também demonstrar a comunicação *RS232C* entre o *PLC* e o microcontrolador.

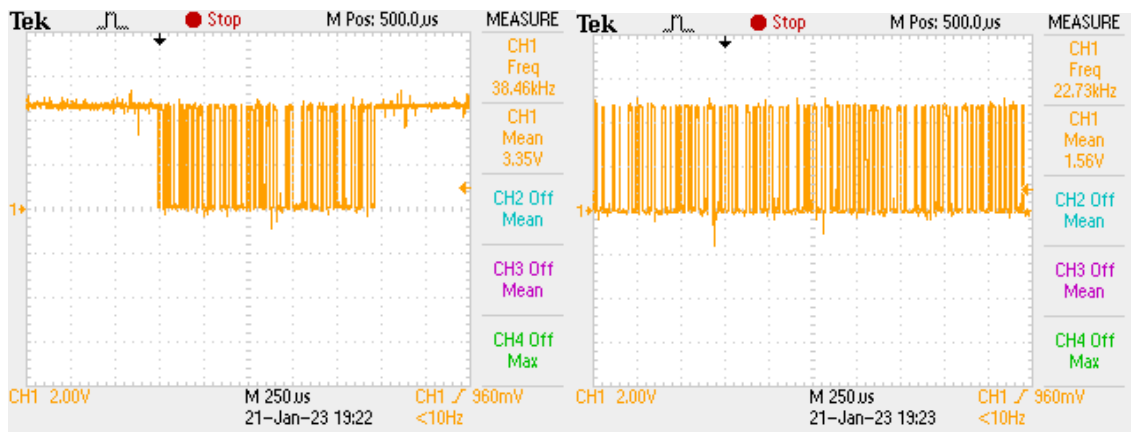


Figura 4.8 – Formas de onda da comunicação *RS232C* entre o microcontrolador e o PLC registadas nos pins do microcontrolador, esquerda: pin 27 (*UART2 Tx*), direita: pin 28 (*UART2 Rx*).

#### 4.1.1. Curvas características para diferentes valores de irradiância incidente e temperatura

Tendo sido comprovado o funcionamento do sistema no modo I para um ponto de funcionamento, de forma a consolidar os resultados pretende-se também simular vários pontos de funcionamento em condições de irradiância incidente e temperatura variáveis. Esta recolha de resultados foi simplificada através de uma função implementada para gerar um ciclo de alterações progressivas à tensão de referência imposta pelo *PLC* ao regulador de carga.

Foram assim recolhidos vários pontos de funcionamento para diversos valores de irradiância incidente: 1000 W/m<sup>2</sup>, 800 W/m<sup>2</sup>, 600 W/m<sup>2</sup>, 400 W/m<sup>2</sup> e 200 W/m<sup>2</sup>. O valor de temperatura foi mantido conforme as condições *STC*: 25°C.

Seguidamente, de forma análoga, foram recolhidos vários pontos de funcionamento para diversos valores de temperatura: 10°C, 25°C, 40°C, 55°C e 70°C. Identicamente, o valor de irradiância incidente foi mantido conforme as condições *STC*: 1000 W/m<sup>2</sup>.

Os resultados foram exportados e inseridos no *MATLAB*, de forma a ter uma comparação com as curvas teóricas. Os pontos assinalados nas figuras são provenientes dos resultados experimentais, e as curvas são geradas pelo *MATLAB* com a mesma cor para uma fácil associação. Para a variação de irradiância incidente, ver Figura 4.9 e Figura 4.10. Para a variação de temperatura, ver Figura 4.11 e Figura 4.12.

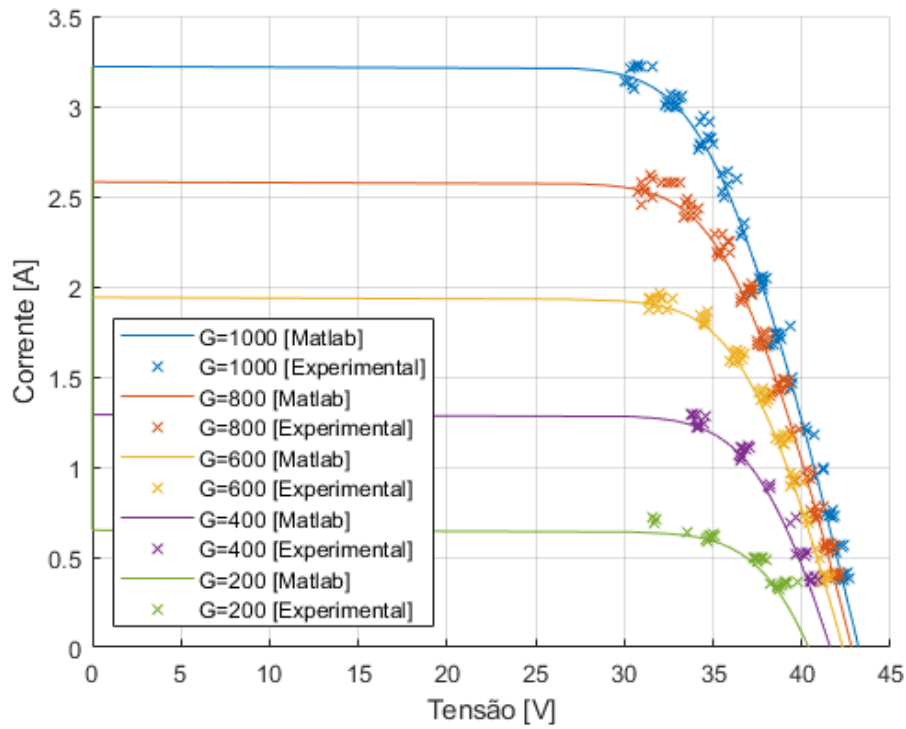


Figura 4.9 – Curvas características I-V para diferentes valores de irradiância incidente com pontos experimentais.

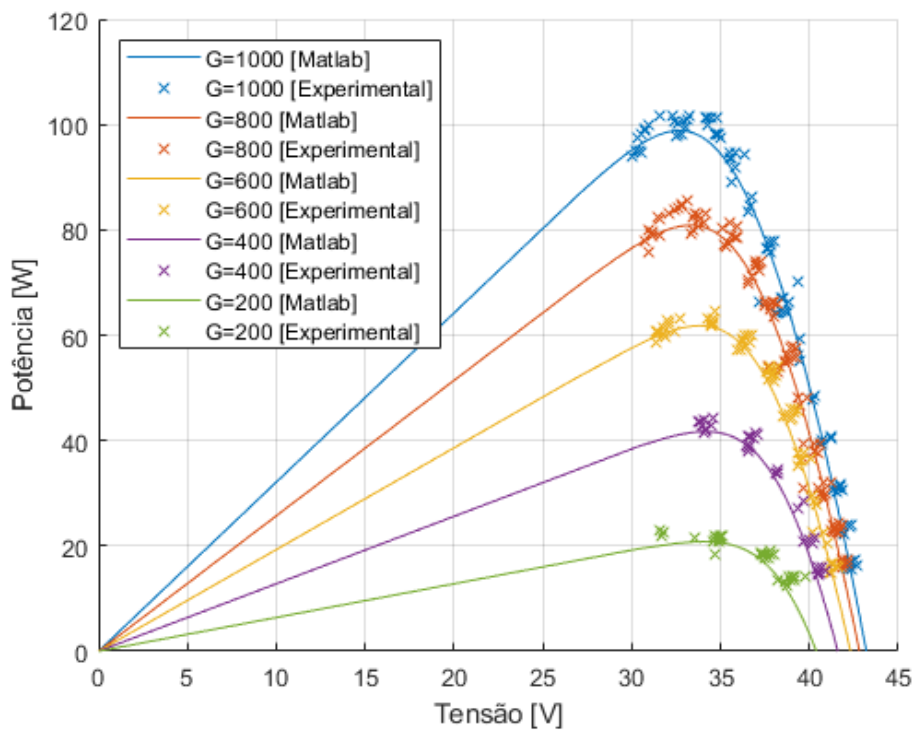


Figura 4.10 – Curvas características P-V para diferentes valores de irradiância incidente com pontos experimentais.

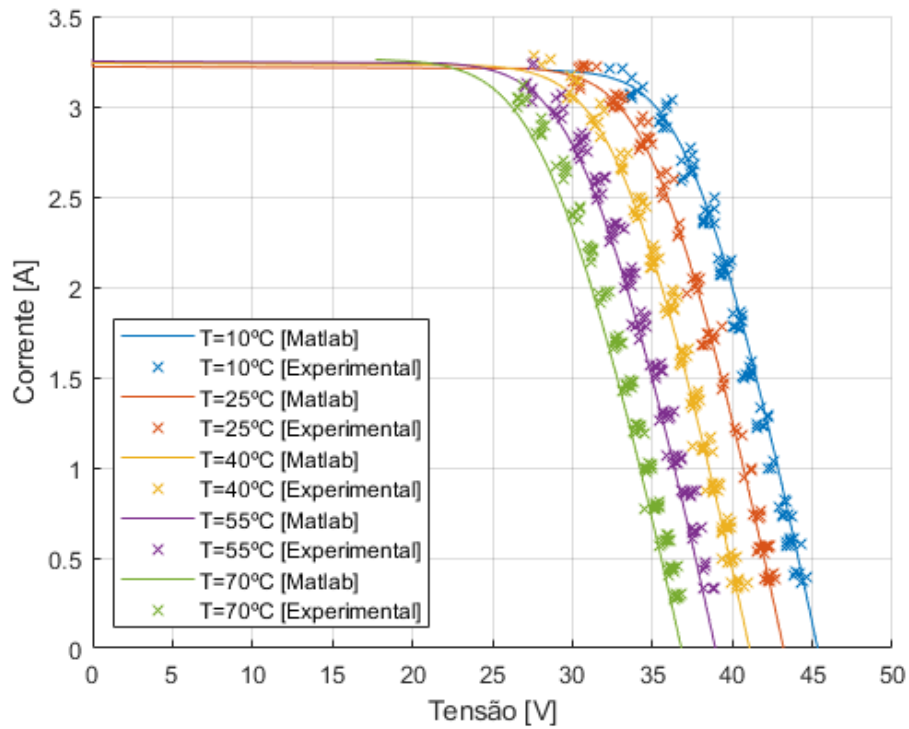


Figura 4.11 – Curvas características I-V para diferentes valores de temperatura com pontos experimentais.

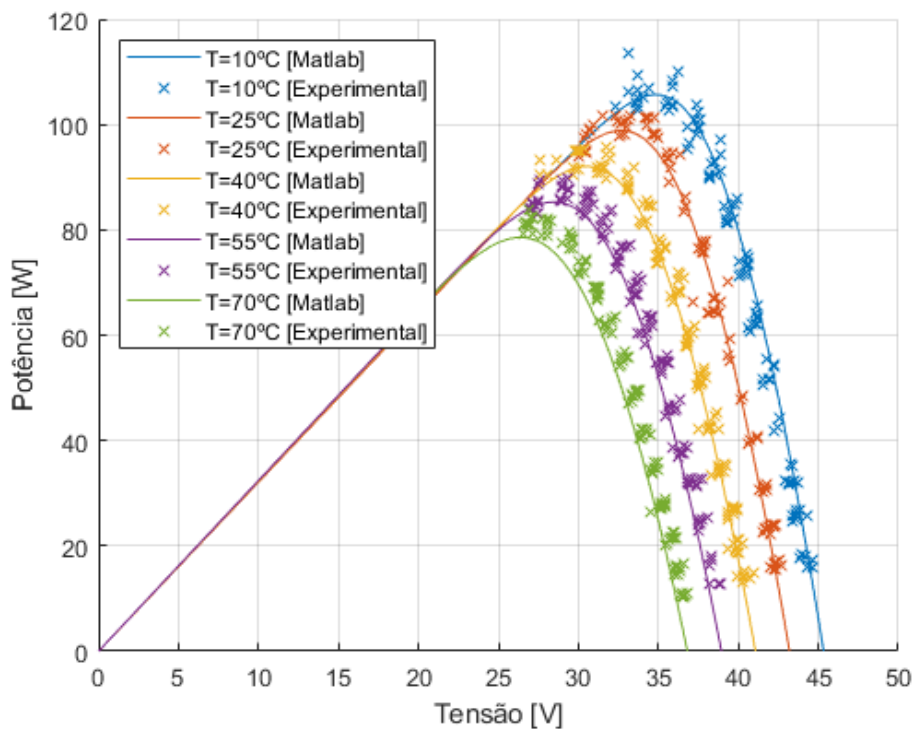


Figura 4.12 – Curvas características P-V para diferentes valores de temperatura com pontos experimentais.

## 4.2. Modo II (Algoritmo *MPPT*)

Para o ensaio do modo II, foi escolhido um ponto de operação para o painel *SPV* próximo à corrente de curto-circuito. Isto para o conversor *Boost* poder elevar a tensão e consequentemente reduzir a corrente, levando o sistema para o ponto de máxima potência. Foi também limitada a corrente da fonte, de forma a limitar a potência à potência aproximada do painel, pois sem esta limitação o algoritmo *MPPT* iria caminhar até ao limite de potência da fonte que ultrapassa a potência do painel *SPV* simulado. Os parâmetros do painel *SPV* assim como as condições ambientais simuladas são apresentadas de seguida.

Parâmetros do painel *SPV*:

- $V_{ca} = 43,20 \text{ V}$
- $I_{cc} = 3,22 \text{ A}$
- N° células em série = 72
- Resistência série = 1,86
- $n = 0,919$

Condições ambientais do ensaio:

- $G = 1000 \text{ W/m}^2$  (*STC*)
- $T = 25^\circ\text{C}$  (*STC*)

Na Figura 4.13 podem ser observadas as formas de onda do modulador *PWM* do conversor *Boost*.

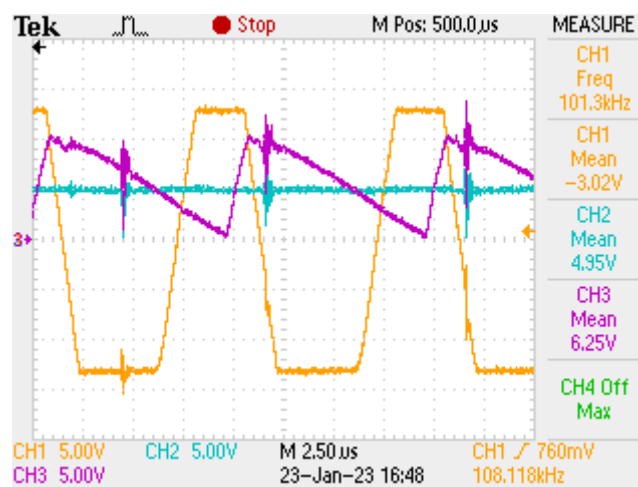


Figura 4.13 – Formas de onda do modulador *PWM* do conversor *Boost*, CH1: sinal *PWM*, CH2: tensão de referência do algoritmo *MPPT* em funcionamento no *PLC*, CH3: onda triangular de 100 kHz.

Na Figura 4.14 pode ser observada a imagem da corrente na bobine em conjunto com o sinal *PWM* gerado. Verifica-se que a bobine opera no modo contínuo, existindo diminuição da corrente quando o *MOSFET* conduz e aumento da mesma quando este está ao corte.

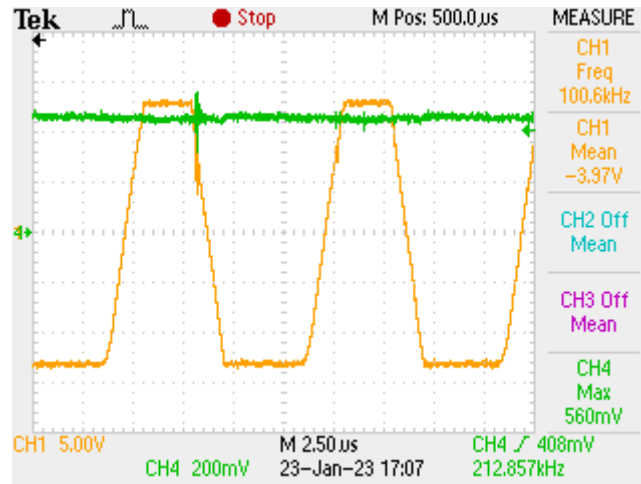


Figura 4.14 – Formas de onda, CH1: sinal *PWM* de comutação do *MOSFET* do conversor *Boost*, CH4: corrente na bobine (100mV/A).

Foram exportados os pontos do ensaio e colocados em sobreposição com as curvas correspondentes simuladas no MATLAB, que podem ser observados na Figura 4.15 e Figura 4.16.

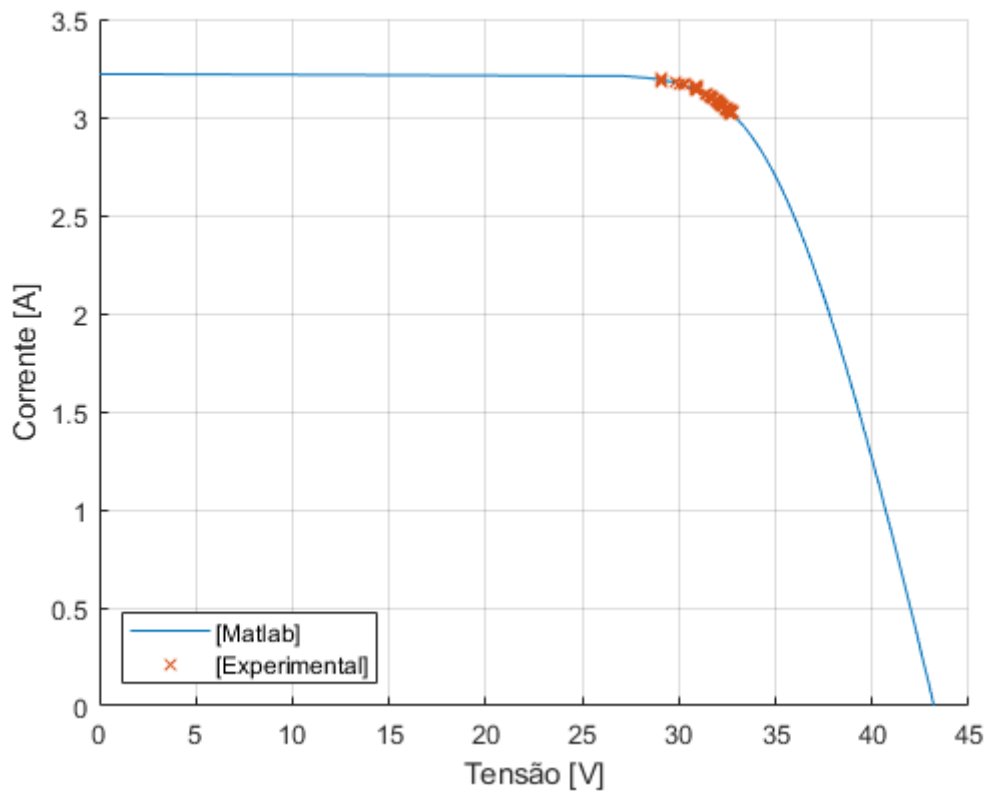


Figura 4.15 – Pontos experimentais da curva I-V do algoritmo *MPPT* ensaiado.

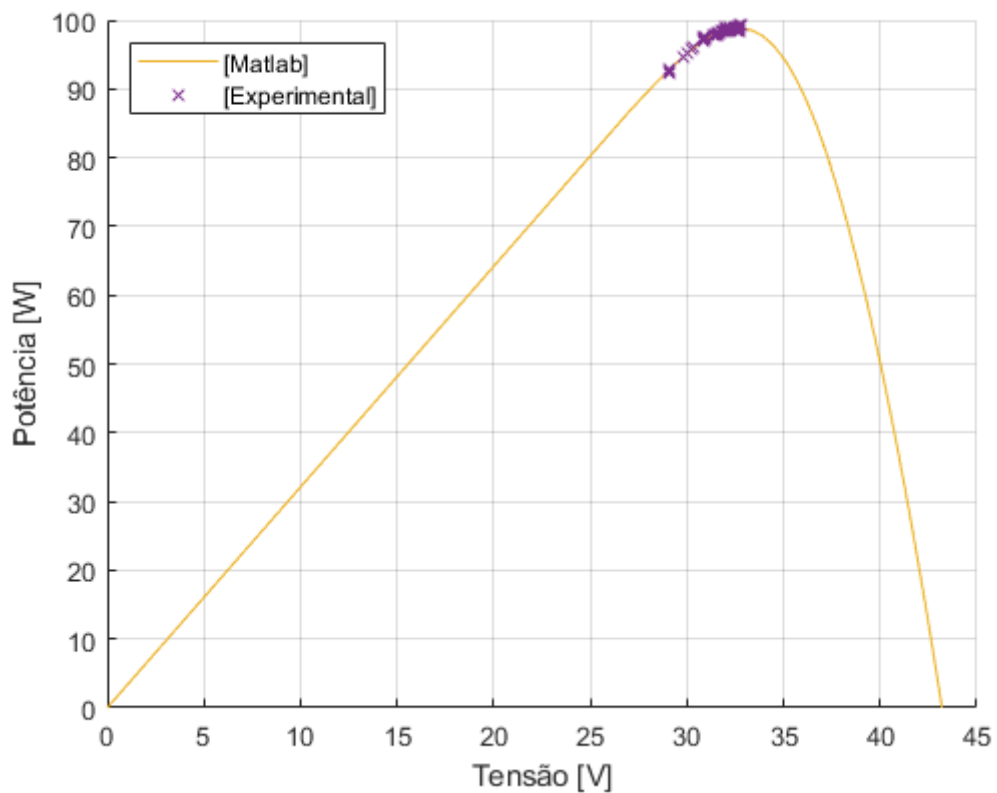


Figura 4.16 - Pontos experimentais da curva P-V do algoritmo *MPPT* ensaiado.

De acordo com a Figura 4.15 e a Figura 4.16, pode ser observado o comportamento do algoritmo *MPPT*, levando o painel *SPV* de um determinado ponto inicial para o ponto de máxima potência. Os pontos estão com alguma sobreposição, facto que comprova que o passo de perturbação é reduzido. A sequência dos pontos ao apresentar pouca variação face às curvas simuladas em MATLAB comprova que o algoritmo funciona de forma correta, com uma taxa de erros reduzida tendo em conta as oscilações dos circuitos do sistema.

Quando o ponto de máxima potência é atingido, é observada uma oscilação em torno do mesmo. Comportamento que se deve ao funcionamento do algoritmo, visto estar continuamente à procura de variações, e a introduzir perturbações para caminhar para o ponto de máxima potência. Na Figura 4.17 está demonstrado o estado do sistema *SCADA* no ponto de máxima potência.

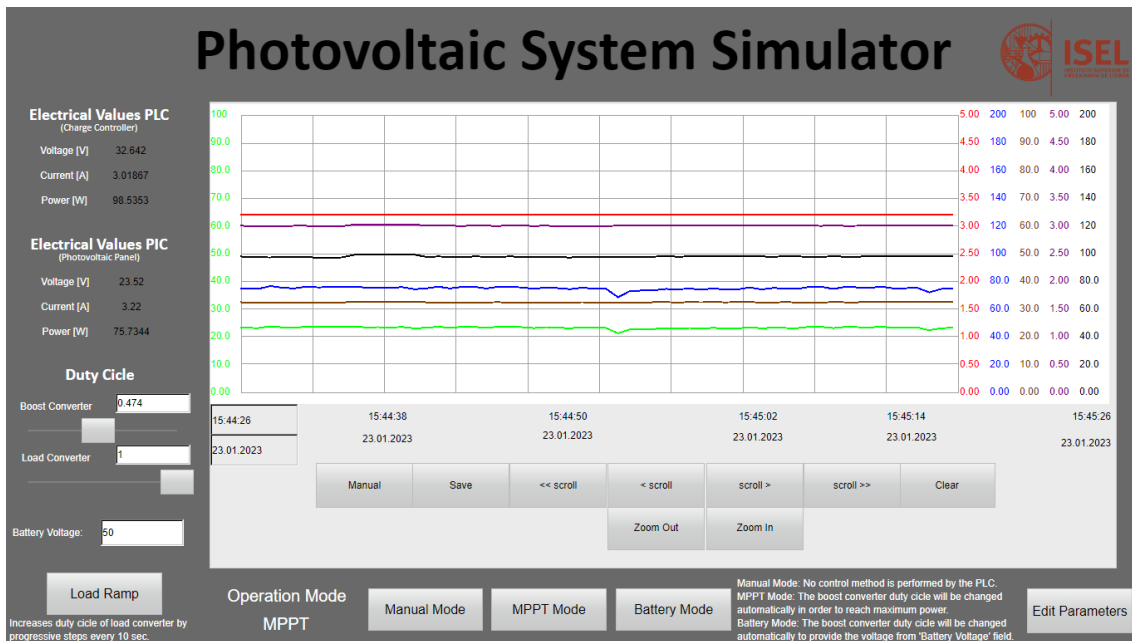


Figura 4.17 – Sistema *SCADA* no ponto de máxima potência.

## 5. Conclusões

Era pretendido como objetivo deste Trabalho Final de Mestrado a construção de um simulador de um sistema de produção de energia fotovoltaica integrando mecanismos automáticos de realização de ensaios.

A intenção inicial era de implementar uma técnica de controlo de modo deslizante com conversores CC-CC *Buck-Boost*. Esta técnica funcionava em simulação, porém, por problemas com os componentes dos circuitos, não foi possível implementar na prática. Foi assim adotada uma estratégia alternativa de controlo em malha fechada por *PWM*.

A solução de controlo implementada simplificou o desenvolvimento e apresentou comportamentos idênticos e resultados muito próximos aos esperados no que diz respeito ao painel *SPV*, assim como ao controlador com algoritmo *MPPT*.

Foi observada uma desvantagem nesta montagem: as comutações “em cadeia” dos *MOSFET* usados nos conversores CC-CC provocaram algum ruído elétrico, que se intensificava com o aumento da corrente. Este ruído afetou particularmente as leituras do transdutor de corrente associado ao *dsPIC*, pois as entradas analógicas do mesmo não carecem de qualquer tipo de filtragem, facto que não acontece com o *PLC*. Isto levou a uma oscilação nos valores lidos, o que em consequência levou a que os valores de tensão impostos pelo *dsPIC* para simulação do painel *SPV* tivessem um ligeiro desvio em torno do valor correto.

Com o decorrer do projeto houve diversas tentativas de mitigação do ruído elétrico das comutações: foi escolhido outro transdutor de corrente, que se verificou em testes práticos que é mesmo suscetível às variações bruscas de corrente; foram colocados filtros passa-baixo entre as saídas dos transdutores e as entradas analógicas do *dsPIC*; a bobine da carga foi substituída por outra de indutividade mais elevada de forma a tentar que a corrente em termos globais tivesse menos variações bruscas. Após todas estas alterações, verificou-se uma melhoria significativa, porém não foi possível eliminar por completo este efeito negativo.

O desenvolvimento do programa do *dsPIC* foi um tema desafiante, por falta de experiência com este dispositivo em particular, foi necessário despende de um tempo considerável de estudo e testes práticos. Considero que foi bem-sucedido, pois foi possível colocar o dispositivo a funcionar próximo à sua velocidade de *clock* máxima,

implementar as funções de cálculo, usufruir das entradas e saídas analógicas assim como implementar a comunicação *SPI* e comunicação série.

A interface homem-máquina (*HMI*) foi implementada com sucesso para efeitos de supervisão (visualização das grandezas elétricas e recolha de dados), como para efeitos de controlo (alteração de parâmetros de funcionamento e automatização de testes). Demonstrou ser uma mais-valia como uma ferramenta integradora de sistemas e comprovou a implementação bem-sucedida de comunicação série entre o *PLC* e o *dsPIC*.

De uma forma geral o simulador foi concluído com sucesso cumprindo os objetivos propostos. O sistema desenvolvido permite de forma rápida e automática recolher dados de um determinado painel *SPV* e de um algoritmo *MPPT*, sendo também intuitivo e prático ao utilizador de alterar os parâmetros de funcionamento.

Os ensaios apresentados comprovam o funcionamento do simulador do painel fotovoltaico, assim como do algoritmo de maximização de potência para diversas condições atmosféricas.

Com o simulador desenvolvido foi possível estudar e aplicar conhecimentos de diversas áreas, nomeadamente eletrónica de potência, eletrónica de regulação e comando, aquisição e processamento de sinais e automação industrial. O trabalho desenvolvido foi sem dúvida uma mais-valia em termos de conhecimentos adquiridos, especialmente na aplicabilidade de conceitos de áreas distintas e sua interação para um objetivo final.

## **5.1. Trabalho futuro**

Seguem-se algumas sugestões de trabalho futuro, partindo do simulador construído neste Trabalho Final de Mestrado.

Sem mexer na composição do simulador criado, podem ser testados outros painéis *SPV* de forma a verificar a aplicabilidade do simulador para vários equipamentos existentes comercialmente. Pode ser desenvolvida a função de teste automático para simular outros tipos de cenários de condições ambientais, que consistam em variações de temperatura ou irradiância incidente. A página do sistema *SCADA* pode ser alterada para se adequar a outros tipos de simulações.

Podem ser aplicadas outras topologias de conversores *CC-CC*, tais como as mencionadas no subcapítulo 2.4.1. Outros métodos *MPPT* podem também integrar facilmente o projeto, tais como os descritos em 2.4.3 ou outros mencionados em

referência no mesmo subcapítulo. Também podem ser explorados outros modelos de painéis *SPV*, tais como os apresentados em 2.3.1 ou outros existentes na literatura. Podem ser também feitas modificações que permitam testes do simulador com ligação à rede elétrica.

## 6. Referências

- [1] F. M. Baena-Moreno, M. Rodríguez-Galán, F. Vega, B. Alonso-Fariñas, L. F. V. Arenas e B. Navarrete, “Carbon capture and utilization technologies: a literature review and recent advances,” *Energy sources, Part A: Recovery, Utilization, and Environmental Effects*, vol. 41:12, pp. 1403-1433, 2019.
- [2] IEA, “International Energy Agency Report, IEA. Global Energy Review,” IEA, 2021.
- [3] REN, “Renewables 2021 Global Status Report – a comprehensive annual overview of the state of renewable energy,” REN21, 2021.
- [4] P. Gangwar, R. P. Tripathi e A. K. Singh, “Solar photovoltaic tree: a review of designs, performance, applications, and challenges,” *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*.
- [5] J. & D. S. A. Blunden, State of the Climate in 2019, Bulletin of the American Meteorological Society, 2020.
- [6] M. F., Energia Solar Passiva, Argumentum, 2010.
- [7] The World Bank, “Global Solar Atlas 2.0,” 2020.
- [8] J. & A. G. M. Tavares Pinho, Manual de Engenharia de Sistemas Fotovoltaicos, Rio de Janeiro: Cepel, 2014.
- [9] M. d. O. L. Alves, “Energia Solar: Estudo da geração de energia elétrica através dos sistemas fotovoltaicos on-grid e off-grid,” 2019.
- [10] S. S. A. Freitas, “Dimensionamento de sistemas fotovoltaicos,” 2008.
- [11] S. & Kagali, “European Journal of Physics Education,” *Determination of the Declination of the Sun on a Given Day*, pp. 17-22, 2012.
- [12] R. Castro, Uma introdução às energia renováveis: Eólica, Fotovoltaica e Mini-Hídrica, IST Press, 2011.
- [13] C. P. Cameron e W. E. & R. D. M. Boyson, “Comparison of PV system performance-model predictions with measured PV system performance,” em *33rd IEEE Photovoltaic Specialists Conference*, 2008.
- [14] A. Goetzberger e C. Hebling, “Photovoltaic materials, past, present, future,”

*Solar Energy Materials and Solar Cells*, Vols. %1 de %262(1-2), pp. 1-19, 2000.

- [15] B. P. Singh, S. K. Goyal e P. Kumar, “Solar PV cell materials and technologies: Analyzing the recent developments,” *Materials Today: Proceedings*, vol. 43, p. 2843–2849, 2021.
- [16] A. Polman, M. Knight, E. C. Garnett, B. Ehrler e W. C. Sinke, “Photovoltaic materials: Present efficiencies and future challenges,” *Science*, vol. 352(6283), p. aad4424, 2016.
- [17] N. M. Kumar, S. S. Chopra, A. K. V. de Oliveira, H. Ahmed, S. Vaezi, U. E. Madukanya e J. M. Castañón, “Solar PV module technologies,” *Photovoltaic Solar Energy Conversion*, p. 51–78, 2020.
- [18] R. Castro e M. Silva, “Experimental and Theoretical Validation of One Diode and Three Parameters-Based PV Models”.*Energies 2021, 14, 2120*.
- [19] J. C. V. Madeira, “Caracterização de Silício por Fotoluminescência e Electroluminescência: Implementação de um Sistema de Medida,” Dissertação de Mestrado, Faculdade de Ciências da Universidade de Lisboa, 2011.
- [20] D. Sera, R. Teodorescu e P. Rodriguez, “34th Annual Conference of IEEE Industrial Electronics,” *Photovoltaic Module Diagnostics by Series Resistance Monitoring and Temperature and Rated Power Estimation*, 2008.
- [21] R. Khezzer, Z. M. e K. A., “Modeling improvement of the four parameter model for photovoltaic modules,” *Elvesier - Solar Energy Volume 110*, Dezembro 2014.
- [22] D. A.D., N. E. e L. Ponnusamy, “Influence of Shunt Resistance on the Performance of Solar Photovoltaic Cell,” 2012.
- [23] B. Alsayid, “Modeling and Simulation of Photovoltaic Cell/Module/Array with Two-Diode Model,” *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, vol. 1, nº 3, pp. 6-11, 2012.
- [24] M. Â. S. Valente, “Caracterização Automática de um Pannel Fotovoltaico,” 2011.
- [25] R. A. O. Barros, “Interface entre Painéis Solares Fotovoltaicos e uma Carga Monofásica,” 2012.
- [26] J. Carneiro, Módulos Fotovoltaicos - Características e Associações, 2010.
- [27] J. Bastidas, C. Ramos e E. Franco, “Modeling and parameter calculation of photovoltaic fields in irregular weather conditions,” *Ingeniería, Vol. 17, No. 1*, pp.

37-48, 2012.

- [28] I. M. Mendes, “Dimensionamento de um Sistema de Energia Fotovoltaica para Autoconsumo,” 2019.
- [29] EDP Distribuição, “Manual de ligações à rede elétrica de serviço público,” 2018.
- [30] Y. Hishikawa, T. Doi, M. Higa, K. Yamagoe, H. Ohshima, T. Takenouchi e M. Yoshita, “Voltage-Dependent Temperature Coefficient of the I–V Curves of Crystalline Silicon Photovoltaic Modules,” *IEEE Journal of Photovoltaics*, vol. 8(1), pp. 48-53, 2018.
- [31] S. Mann, E. Fadel, S. S. Schoenholz, E. D. Cubuk, S. G. Johnson e G. Romano, “ $\partial$ PV: An end-to-end differentiable solar-cell simulator,” *Computer Physics Communications*, vol. 272, n° 108232, 2022.
- [32] M. S. Khalid e M. Abido, “A novel and accurate photovoltaic simulator based on seven-parameter model,” *Electric Power Systems Research*, vol. 116, pp. 243-251, 2014.
- [33] M. Ünlü e S. Çamur, “A simple Photovoltaic simulator based on a one-diode equivalent circuit model,” *2017 4th International Conference on Electrical and Electronic Engineering (ICEEE)*, pp. 33-36, 2017.
- [34] N. M. A. A. Shannan, N. Z. Yahaya e B. Singh, “Single-diode model and two-diode model of PV modules: A comparison,” *2013 IEEE International Conference on Control System, Computing and Engineering*, pp. 210-214, 2013.
- [35] S. M. H. Hosseini e A. A. Keymanesh, “Design and construction of photovoltaic simulator based on dual-diode model,” *Solar Energy*, vol. 137, pp. 594-607, 2016.
- [36] B. Singh, M. Singla e P. Nijhawan, “Parameter estimation of four diode solar photovoltaic cell using hybrid algorithm,” *Energy Sources, Part A: Recovery, Utilization and Environmental Effects*, vol. 44(2), p. 4597–4613, 2022.
- [37] S. Sousa, M. Onofre, T. Antunes, C. Branco, J. Maia, J. I. Rocha e V. F. Pires, “Implementation of a low cost data acquisition board for photovoltaic arrays analysis and diagnostic,” *2013 International Conference on Renewable Energy Research and Applications (ICRERA)*, pp. 1084-1088, 2013.
- [38] C. Roncero-Clemente, E. Romero-Cadaval, V. M. Minambres, M. A. Guerrero-Martinez e J. Gallardo-Lozano, “PV Array Emulator for Testing

- Commercial PV Inverters,” *Elektronika ir Elektrotechnika*, vol. 19, n° 10, pp. 71-75, 2013.
- [39] L. Bun, B. Raison, G. Rostaing, S. Bacha, A. Rumeau e A. Labonne, “Development of a real time photovoltaic simulator in normal and abnormal operations,” *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, pp. 867-872, 2011.
- [40] N. Kahoul, M. Houabes e A. Neçaibia, “A comprehensive simulator for assessing the reliability of a photovoltaic panel peak power tracking system,” *Frontiers in Energy*, vol. 9, pp. 170-179, 2015.
- [41] W. Subsingha, “Real-time Photovoltaic Simulator Using Current Feedback Control,” *Energy Procedia*, vol. 89, pp. 160-169, 2016.
- [42] K.-H. Tang, K.-H. Chao, Y.-W. Chao e J.-P. Chen, “Design and Implementation of a Simulator for Photovoltaic Modules,” *International Journal of Photoenergy*, vol. 2012, 2012.
- [43] H. Qi, Y. Bi e Y. Wu, “Development of a photovoltaic array simulator based on buck convertor,” *2014 International Conference on Information Science, Electronics and Electrical Engineering*, pp. 14-17, 2014.
- [44] A. Cordeiro, D. Foito e V. F. Pires, “A PV panel simulator based on a two quadrant DC/DC power converter with a sliding mode controller,” *International Conference on Renewable Energy Research and Applications (ICRERA)*, pp. 928-932, 2015.
- [45] W. Zhang e J. W. Kimball, “DC-DC Converter Based Photovoltaic Simulator with a Double Current Mode Controller,” *IEEE Trans. Power Electro*, vol. 33, n° 7, pp. 5860-5868, 2018.
- [46] I. Chakraborty, S. Sekaran e S. K. Pradhan, “An enhanced DC-DC boost converter based stand-alone PV-Battery OFF-Grid system with voltage balancing capability for fluctuating environmental and load conditions,” *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, vol. 44:3, pp. 8247-8265, 2022.
- [47] V. V. S. P. Kumar e B. G. Fernandes, “Fault tolerant topology for two level VSI and transformerless grid connected inverters,” *2014 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, pp. 1-6, 2014.

- [48] A. Aghazadeh, M. Davari, H. Nafisi e F. Blaabjerg, “Grid Integration of a Dual Two-Level Voltage-Source Inverter Considering Grid Impedance and Phase-Locked Loop,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 9, n° 1, pp. 401-422, 2021.
- [49] N. Shah, “Multilevel Inverter Based Single-Stage Grid Connected Photovoltaic System using Cascaded Two-Level Inverter,” *2018 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, pp. 1-6, 2018.
- [50] D. K. Singh, S. Manna e A. K. Akella, “Grid Connected PV System Using Multilevel Inverter,” *2021 7th International Conference on Electrical Energy Systems (ICEES)*, pp. 346-351, 2021.
- [51] R. Costa, “Comparação de Métodos de Seguimento Solar para Sistemas Fotovoltaicos,” 2011.
- [52] P. K. Maroti, S. Padmanaban, P. Wheeler, F. Blaabjerg e M. Rivera, “Modified high voltage conversion inverting cuk DC-DC converter for renewable energy application,” *2017 IEEE Southern Power Electronics Conference (SPEC)*, pp. 1-5, 2017.
- [53] M. S. B. Ranjana, N. SreeramulaReddy e R. K. P. Kumar, “A novel sepic based dual output DC-DC converter for solar applications,” *2014 POWER AND ENERGY SYSTEMS: TOWARDS SUSTAINABLE ENERGY*, pp. 1-5, 2014.
- [54] F. Lima de Sa, C. Dal Agnol, W. Raphael, D. R. Caballero e S. A. Mussa, “A New DC-DC Double Zeta Quadratic Converter,” *2020 IEEE International Conference on Industrial Technology (ICIT)*, pp. 426-431, 2020.
- [55] F. L. Luo, “Luo-Converters, a series of new DC-DC step-up (boost) conversion circuits,” *Proceedings of Second International Conference on Power Electronics and Drive Systems*, pp. 882-888 vol.2, 1997.
- [56] V. F. Pires, D. Foito, A. Cordeiro e J. F. Silva, “A Single-Switch DC/DC Buck-Boost Converter with Extended Output Voltage,” *2018 7th International Conference on Renewable Energy Research and Applications (ICRERA)*, pp. 791-796, 2018.
- [57] V. F. Pires, A. Cordeiro, D. Foito e J. F. Silva, “Control of Bidirectional Quadratic DC-DC Converters for Storage Support of DC Power Grids,” *2018 7th International Conference on Renewable Energy Research and Applications (ICRERA)*, pp. 227-232, 2018.

- [58] S. Roberts, *DC/DC Book of Knowledge*, Austria: RECOM Engineering GmbH & Co KG, 2016.
- [59] F. F. d. Santos, “Utilização de Energia Fotovoltaica para a eficiência energética de uma moradia,” 2011.
- [60] A. Ahmad, F. Murtaza e H. A. Sher, “Power tracking technique for efficient operation of photovoltaic array in solar applications - A review,” *Renewable and Sustainable Energy Reviews*, vol. 101, pp. 82-102, 2019.
- [61] M. A. G. Brito, L. Galotto, L. P. Sampaio, G. d. A. e. Melo e C. A. Canesin, “Evaluation of the Main MPPT Techniques for Photovoltaic Applications,” *IEEE Transactions on Industrial Electronics*, vol. 60, n° 3, pp. 1156-1167, 2013.
- [62] D. Verma, S. Nema, A. M. Shandilya e S. K. Dash, “Maximum power point tracking (MPPT) techniques: Recapitulation in solar photovoltaic systems,” *Renewable and Sustainable Energy Reviews*, vol. 54, p. 1018–1034, 2016.
- [63] J. P. Ram, T. S. Babu e N. Rajasekar, “A comprehensive review on solar PV maximum power point tracking techniques,” *Renewable and Sustainable Energy Reviews*, vol. 67, p. 826–847, 2017.
- [64] D.-Y. Lee, H.-J. Noh, D.-S. Hyun e I. Choy, “An Improved MPPT Converter Using Current Compensation Method for Small Scaled PV-Applications,” em *18th Annual IEEE Applied Power Electronics Conference and Exposition*, 2003.
- [65] Z. M. Salameh, F. Dagher e W. A. Lynch, “Step-down maximum power point tracker for photovoltaic systems,” *Solar Energy*, vol. 46:5, p. 279–282, 1991.
- [66] S. M. Alghuwainem, “Matching of a DC motor to a photovoltaic generator using a step-up converter with a current-locked loop,” *IEEE Transactions on Energy Conversion*, vol. 9:1, p. 192–198, 1994.
- [67] G. Gridling e B. Weiss, *Introduction to Microcontrollers*, Vienna: Vienna University of Technology, 2007.
- [68] J. Crisp, *Introduction to Microprocessors and Microcontrollers*, Newnes, 2003.
- [69] F. d. O. C. Lemos, “Controlo remoto de conversor de potência no âmbito da indústria 4.0,” Instituto Superior de Engenharia de Lisboa, Lisboa, 2018.
- [70] P. M. B. Moreira, “Sistema de Supervisão e Controlo em Instrumentação Distribuída,” Faculdade de Engenharia da Universidade do Porto, Porto, 2011.
- [71] Phoenix Contact, “Webvisit 6 Pro,” [Online]. Available:

- <https://www.phoenixcontact.com/pt-pt/produtos/visualization-webvisit-6-pro-2700949>. [Acedido em 19 5 23].
- [72] P. Inc., “ASE-100-DG-UR mono Solar Panel from Schott Solar,” Possharp, The Source for Renewables, [Online]. Available: [http://www.posharp.com/ase-100-dg-ur-mono-solar-panel-from-schott-solar\\_p1976046512d.aspx](http://www.posharp.com/ase-100-dg-ur-mono-solar-panel-from-schott-solar_p1976046512d.aspx). [Acedido em 8 Março 2022].
- [73] P. SA, “Download/PVSystem,” PVsystem SA, [Online]. Available: <https://www.pvsyst.com/download-pvsyst/>. [Acedido em 13 Novembro 2021].
- [74] B. Hauke, “Basic Calculation of a Buck Converter's Power Stage,” Texas Instruments, 2015.
- [75] Wolfspeed, “Datasheet C2M0080120D,” [Online]. Available: <https://assets.wolfspeed.com/uploads/2020/12/C2M0080120D.pdf>. [Acedido em 19 5 2023].
- [76] B. Hauke, “Basic Calculation of a Boost Converter's Power Stage,” Texas Instruments, 2022.
- [77] W. Gao e J. C. Hung, “Variable Structure Control of Nonlinear Systems: A New Approach,” *IEEE Transactions on Industrial Electronics*, vol. 40, n° 1, pp. 45-55, 1993.
- [78] J. F. Silva e S. F. Pinto, “Linear and Nonlinear Control of Switching Power Converters,” em *Power Electronics Handbook 4th Ed.*, 2018, p. 1141–1220.
- [79] V. Fernão Pires, J. F. Martins e C. Hao, “Dual-inverter for grid-connected photovoltaic system: Modeling and sliding mode control,” *Solar Energy*, vol. 86, n° 7, p. 2106–2115, 2012.
- [80] J. Mérida, L. T. Aguilar e J. Dávila, “Analysis and synthesis of sliding mode control for large scale variable speed wind turbine for power optimization,” *Renewable Energy*, vol. 71, p. 715–728, 2014.
- [81] J. M. R. Vaz, “Dimensionamento de um sistema fotovoltaico,” 2017.
- [82] J. S. Stein, S. McCaslin, C. W. Hansen, W. E. Boyson e C. D. Robinson, “Measuring PV System Series Resistance Without Full IV Curves”.
- [83] C. Nituca, G. Chiriac, D. Cuciureanu, G. Zhang, D. Han e A. Plesca, “Numerical Analysis of a Real Photovoltaic Module with Various Parameters,” *Modelling and Simulation in Engineering*, vol. 2018, 2018.
- [84] T. Majaw, R. Deka, S. Roy e B. Goswami, “Solar Charge Controllers using

MPPT and PWM: A Review,” *ADBU Journal of Electrical and Electronics Engineering*, vol. 2, n° 1, pp. 1-4, 2018.

## **7. Anexos**

Apresentam-se como anexos, os programas desenvolvidos para este trabalho no *dsPIC30f4011*, no *PLC Phoenix Contact ILC 131* e no *HMI*.

C:/Users/helio/Desktop/MPLABX/Tese30.X/main.c

```
1 //// DSPIC30F4011 Configuration Bit Settings
2
3 // 'C' source line config statements
4
5 // FOSC
6 #pragma config FPR = FRC_PLL16           // Primary Oscillator Mode (FRC
w/ PLL 16x)
7 #pragma config FOS = PRI                 // Oscillator Source (Primary
Oscillator)
8 #pragma config FCKSMEN = CSW_FSCM_OFF   // Clock Switching and Monitor
(Sw Disabled, Mon Disabled)
9
10 // FWDT
11 #pragma config FWPSB = WDTPSB_16        // WDT Prescaler B (1:16)
12 #pragma config FWPSA = WDTPSA_512      // WDT Prescaler A (1:512)
13 #pragma config WDT = WDT_OFF           // Watchdog Timer (Disabled)
14
15 // FBORPOR
16 #pragma config FPWRT = PWRT_64          // POR Timer Value (64ms)
17 #pragma config BODENV = BORV42         // Brown Out Voltage (4.2V)
18 #pragma config BOREN = PBOR_ON         // PBOR Enable (Enabled)
19 #pragma config LPOL = PWMxL_ACT_HI      // Low-side PWM Output Polarity
(Active High)
20 #pragma config HPOL = PWMxH_ACT_HI      // High-side PWM Output
Polarity (Active High)
21 #pragma config PWMPIN = RST_IOPIN       // PWM Output Pin Reset
(Control with PORT/TRIS regs)
22 #pragma config MCLRE = MCLR_EN         // Master Clear Enable
(Enabled)
23
24 // FGS
25 #pragma config GWRP = GWRP_OFF         // General Code Segment Write
Protect (Disabled)
26 #pragma config GCP = CODE_PROT_OFF     // General Segment Code
Protection (Disabled)
27
28 // FICD
29 #pragma config ICS = ICS_PGD           // Comm Channel Select (Use
PGC/EMUC and PGD/EMUD)
30
31 // #pragma config statements should precede project file includes.
32 // Use project enums instead of #define for ON and OFF.
33
34 #include <xc.h>
35 #include <p30F4011.h>
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/main.c

```
36 #include <math.h>
37 #include <uart.h>
38 #include <stdlib.h>
39 #include "comm.h"
40 #include "analog.h"
41 #include "timer.h"
42
43 float VT, n, VOC_STC, ISC_STC, IO, RS, VOC, ISC, Ki, Kv, dT, T, G,
Ncell, VPVref, VPV, IPV, VPVref5V;
44 int VPVrefWord;
45 int DataReceived=0;
46 char buffer[80], rxbuffer[80], rxbuffer2[80];
47
48 int main(void) {
49
50     uart1_start();
51     uart2_start();
52     timer1_start();
53     ADC_start();
54     SPI_start();
55
56     putsUART1("Startup ok\r\n");
57
58     VPVref = 0;
59     T = 25;
60     G = 1000;
61     VOC_STC = 43.2;
62     ISC_STC = 3.22;
63     RS = 1.86;
64     Ki = 0.001;
65     Kv = -0.142;
66     n = 0.919;
67     Ncell = 72;
68
69     while (1)
70     {
71
72         VPV=ADC_read(0) * (60.0/1023.0);
73
74         IPV=ADC_read(1) * (5.0/1023.0);
75
76         dT = T - 25;
77         VT = 0.000086173 * (T+273.15) * Ncell;
78
79         IO = (dT*Ki + ISC_STC) / (exp((VOC_STC + Kv*dT) / (n*VT)) - 1);
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/main.c

```
80
81     VOC = ((VOC_STC + (dT * Kv)) * (G/1000));
82
83     ISC = ((ISC_STC + (dT * Ki)) * (G/1000));
84
85     if (IPV>ISC)
86         IPV=ISC-0.01;
87
88     VPVref = n*VT*log((ISC-IPV+IO)/IO) -RS*IPV;
89
90     VPVref5V = VPVref*(5.0/60.0);
91
92     VPVrefWord = (int) (VPVref/0.0146484375);
93
94     SPI_write(VPVrefWord);
95
96     if(DataReceived == 1)
97     {
98         putsUART1("ESCREVEU:\r\n");
99         putsUART1(rxbuffer2);
100        putsUART1("\r\n");
101
102        if(rxbuffer2[0]=='R' && rxbuffer2[1]=='P' &&
rxbuffer2[2]=='A')
103            {
104                sprintf(buffer,
"PA%6.2f#%6.0f#%6.2f#%6.2f#%6.2f#%6.3f#%6.3f#%6.3f#%6.0f#", T, G, VOC_STC,
ISC_STC, RS, Kv, Ki, n, Ncell);
105                putsUART2(buffer);
106            }
107        else if(rxbuffer2[0]=='R' && rxbuffer2[1]=='T' &&
rxbuffer2[2]=='T')
108            {
109                sprintf(buffer, "TT%6.2f", T);
110                putsUART2(buffer);
111            }
112        else if(rxbuffer2[0]=='R' && rxbuffer2[1]=='G' &&
rxbuffer2[2]=='G')
113            {
114                sprintf(buffer, "GG%6.0f", G);
115                putsUART2(buffer);
116            }
117        else if(rxbuffer2[0]=='R' && rxbuffer2[1]=='V' &&
rxbuffer2[2]=='O')
118            {
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/main.c

```
119             sprintf(buffer, "VO%6.2f", VOC_STC);
120             putsUART2(buffer);
121         }
122         else if(rxbuffer2[0]=='R' && rxbuffer2[1]=='I' &&
rxbuffer2[2]=='S')
123         {
124             sprintf(buffer, "IS%6.2f", ISC_STC);
125             putsUART2(buffer);
126         }
127         else if(rxbuffer2[0]=='R' && rxbuffer2[1]=='R' &&
rxbuffer2[2]=='S')
128         {
129             sprintf(buffer, "RS%6.2f", RS);
130             putsUART2(buffer);
131         }
132         else if(rxbuffer2[0]=='R' && rxbuffer2[1]=='K' &&
rxbuffer2[2]=='V')
133         {
134             sprintf(buffer, "KV%6.3f", Kv);
135             putsUART2(buffer);
136         }
137         else if(rxbuffer2[0]=='R' && rxbuffer2[1]=='K' &&
rxbuffer2[2]=='I')
138         {
139             sprintf(buffer, "KI%6.3f", Ki);
140             putsUART2(buffer);
141         }
142         else if(rxbuffer2[0]=='R' && rxbuffer2[1]=='N' &&
rxbuffer2[2]=='N')
143         {
144             sprintf(buffer, "NN%6.3f", n);
145             putsUART2(buffer);
146         }
147         else if(rxbuffer2[0]=='R' && rxbuffer2[1]=='N' &&
rxbuffer2[2]=='C')
148         {
149             sprintf(buffer, "NC%6.0f", Ncell);
150             putsUART2(buffer);
151         }
152         else if(rxbuffer2[0]=='W' && rxbuffer2[1]=='P' &&
rxbuffer2[2]=='A')
153         {
154             T=atof(rxbuffer2+3);
155             G=atof(rxbuffer2+10);
156             VOC_STC=atof(rxbuffer2+17);
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/main.c

```
157             ISC_STC=atof(rxbuffer2+24);
158             RS=atof(rxbuffer2+31);
159             Kv=atof(rxbuffer2+38);
160             Ki=atof(rxbuffer2+45);
161             n=atof(rxbuffer2+52);
162             Ncell=atof(rxbuffer2+59);
163             sprintf(buffer,
"PA%6.2f#%6.0f#%6.2f#%6.2f#%6.2f#%6.3f#%6.3f#%6.3f#%6.0f#", T, G, VOC_STC,
ISC_STC, RS, Kv, Ki, n, Ncell);
164             putsUART2(buffer);
165         }
166         else if(rxbuffer2[0]=='W' && rxbuffer2[1]=='T' &&
rxbuffer2[2]=='T')
167         {
168             T=atof(rxbuffer2+3);
169             sprintf(buffer, "TT%6.2f", T);
170             putsUART2(buffer);
171         }
172         else if(rxbuffer2[0]=='W' && rxbuffer2[1]=='G' &&
rxbuffer2[2]=='G')
173         {
174             G=atof(rxbuffer2+3);
175             sprintf(buffer, "GG%6.0f", G);
176             putsUART2(buffer);
177         }
178         else if(rxbuffer2[0]=='W' && rxbuffer2[1]=='V' &&
rxbuffer2[2]=='O')
179         {
180             VOC_STC=atof(rxbuffer2+3);
181             sprintf(buffer, "VO%6.2f", VOC_STC);
182             putsUART2(buffer);
183         }
184         else if(rxbuffer2[0]=='W' && rxbuffer2[1]=='I' &&
rxbuffer2[2]=='S')
185         {
186             ISC_STC=atof(rxbuffer2+3);
187             sprintf(buffer, "IS%6.2f", ISC_STC);
188             putsUART2(buffer);
189         }
190         else if(rxbuffer2[0]=='W' && rxbuffer2[1]=='R' &&
rxbuffer2[2]=='S')
191         {
192             RS=atof(rxbuffer2+3);
193             sprintf(buffer, "RS%6.2f", RS);
194             putsUART2(buffer);
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/main.c

```
195         }
196         else if(rxbuffer2[0]=='W' && rxbuffer2[1]=='K' &&
rxbuffer2[2]=='V')
197         {
198             Kv=atof(rxbuffer2+3);
199             sprintf(buffer, "KV%6.3f", Kv);
200             putsUART2(buffer);
201         }
202         else if(rxbuffer2[0]=='W' && rxbuffer2[1]=='K' &&
rxbuffer2[2]=='I')
203         {
204             Ki=atof(rxbuffer2+3);
205             sprintf(buffer, "KI%6.3f", Ki);
206             putsUART2(buffer);
207         }
208         else if(rxbuffer2[0]=='W' && rxbuffer2[1]=='N' &&
rxbuffer2[2]=='N')
209         {
210             n=atof(rxbuffer2+3);
211             sprintf(buffer, "NN%6.3f", n);
212             putsUART2(buffer);
213         }
214         else if(rxbuffer2[0]=='W' && rxbuffer2[1]=='N' &&
rxbuffer2[2]=='C')
215         {
216             Ncell=atof(rxbuffer2+3);
217             sprintf(buffer, "NC%6.0f", Ncell);
218             putsUART2(buffer);
219         }
220
221         memset(rxbuffer2,0,80*sizeof(char));
222         DataReceived = 0;
223     }
224
225 }
226
227 }
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/analog.c

```
1 #include <xc.h>
2 #include "analog.h"
3
4 void ADC_start(void)
5 {
6     TRISB = 0xFF;           //Input pins
7     ADPCFG = 0x00;        //Analog pins
8
9     ADCON1bits.ADSIDL = 0; //Continue operation in idle mode
10    ADCON1bits.FORM = 0b000; //Integer format
11    ADCON1bits.SSRC = 0b111; //Enable auto-convert
12    ADCON1bits.SIMSAM = 0; //Sample multiple channels individually
in sequence
13    ADCON1bits.ASAM = 0; //Sampling begins when SAMP bit set
14
15    ADCON2bits.VCFG = 0b000; //AVdd AVss
16    ADCON2bits.CSCNA = 0; //Do not scan inputs
17    ADCON2bits.CHPS = 0b00; //Converts CH0
18    ADCON2bits.SMPI = 0b0000; //Interrupt at completion of each
sample/convert
19    ADCON2bits.BUFG = 0; //Buffer configured as one 16-word buffer
20    ADCON2bits.ALTS = 0; //Use MUX A input multiplexer settings
21
22    ADCON3bits.SAMC = 0b00001; //Auto-sample time 1 Tad
23    ADCON3bits.ADRC = 0; //Clock derived from system clock
24    ADCON3bits.ADCS = 0b001001; //9 - Tcy = 33,33nseg -> Tad = 166,65nseg
25
26    ADCHSbits.CH0NA = 0; //CH0 Negative input AVss
27
28    ADCSSL = 0; //Do not scan inputs
29
30    ADCON1bits.ADON = 1; //A/D converter module is operating
31 }
32
33 int ADC_read(int ch)
34 {
35     ADCHSbits.CH0SA = ch; //Select input channel
36     ADCON1bits.SAMP = 1; //Start sampling
37     __delay32(10); //Wait 10 cycles (333,33nseg at 30MIPS)
38     ADCON1bits.SAMP = 0; //Stop sampling
39     while(!ADCON1bits.DONE); //Wait for conversion to complete
40     return ADCBUF0; //Read conversion result
41 }
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/analog.h

```
1 #ifndef ANALOG_H
2 #define ANALOG_H
3
4 void ADC_start(void);
5
6 int ADC_read(int ch);
7
8 #endif
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/comm.c

```
1 #include <xc.h>
2 #include <stdlib.h>
3 #include <uart.h>
4 #include "comm.h"
5
6 void __attribute__((__interrupt__)) _U1RXInterrupt(void)
7 {
8     putsUART1("Interrupt ok\r\n");
9
10    getsUART1(5, rxbuffer, 123);
11
12    if(rxbuffer[0]=='m')
13    {
14        memset(rxbuffer,0,5*sizeof(char));
15        putsUART1("MENU TERMINAL RS-232\r\n");
16        putsUART1("1 - Ver condicoes solares e temperatura\r\n");
17        putsUART1("2 - Ver caracteristicas do painel\r\n");
18        putsUART1("3 - Alterar condicoes solares e
temperatura\r\n");
19        putsUART1("4 - Alterar caracteristicas do painel\r\n");
20
21        while(rxbuffer[0]!='1' && rxbuffer[0]!='2' &&
rxbuffer[0]!='3' && rxbuffer[0]!='4')
22        {
23            memset(rxbuffer,0,5*sizeof(char));
24            getsUART1(5, rxbuffer, 123);
25        }
26        if(rxbuffer[0]=='1')
27        {
28            memset(rxbuffer,0,5*sizeof(char));
29            putsUART1("VALOR ATUAL DE TEMPERATURA (Celsius)
[T]: ");
30            sprintf(buffer, "%.2f\r\n", T);
31            putsUART1(buffer);
32
33            putsUART1("VALOR ATUAL DE IRRADIACAO (w/m2) [G]:
");
34            sprintf(buffer, "%.2f\r\n", G);
35            putsUART1(buffer);
36
37        }
38
39        if(rxbuffer[0]=='2')
40        {
41            memset(rxbuffer,0,5*sizeof(char));
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/comm.c

```
42         putsUART1("Tensao em circuito aberto [VOC_STC]: ");
43         sprintf(buffer, "%.2f\r\n", VOC_STC);
44         putsUART1(buffer);
45
46         putsUART1("Corrente de curto-circuito [ISC_STC]:
");
47         sprintf(buffer, "%.2f\r\n", ISC_STC);
48         putsUART1(buffer);
49
50         putsUART1("Resistencia Serie [RS]: ");
51         sprintf(buffer, "%.2f\r\n", RS);
52         putsUART1(buffer);
53
54         putsUART1("Coeficiente Corrente-Temperatura [Ki]:
");
55         sprintf(buffer, "%.2f\r\n", Ki);
56         putsUART1(buffer);
57
58         putsUART1("Coeficiente Tensao-Temperatura [Kv]: ");
59         sprintf(buffer, "%.2f\r\n", Kv);
60         putsUART1(buffer);
61
62         putsUART1("Fator de idealidade [n]: ");
63         sprintf(buffer, "%.2f\r\n", n);
64         putsUART1(buffer);
65
66         putsUART1("Numero de celulas do painel [Ncell]: ");
67         sprintf(buffer, "%.2f\r\n", Ncell);
68         putsUART1(buffer);
69
70     }
71
72     if(rxbuffer[0]=='3')
73     {
74         memset(rxbuffer,0,5*sizeof(char));
75         putsUART1("VALOR ATUAL DE TEMPERATURA (Celsius
[T]: ");
76         sprintf(buffer, "%.2f\r\n", T);
77         putsUART1(buffer);
78         putsUART1("INTRODUZA NOVO VALOR:\r\n");
79         while(rxbuffer[0]==0)
80         {
81             getsUART1(5, rxbuffer, 12300);
82         }
83         T=atof(rxbuffer);
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/comm.c

```
84         memset(rxbuffer,0,5*sizeof(char));
85         putsUART1("VALOR NOVO DE TEMPERATURA (Celsius) [T]:
");
86         sprintf(buffer, "%.2f\r\n", T);
87         putsUART1(buffer);
88
89         memset(rxbuffer,0,5*sizeof(char));
90         putsUART1("VALOR ATUAL DE IRRADIACAO (w/m2) [G]:
");
91         sprintf(buffer, "%.2f\r\n", G);
92         putsUART1(buffer);
93         putsUART1("INTRODUZA NOVO VALOR:\r\n");
94         while(rxbuffer[0]==0)
95         {
96             getsUART1(5, rxbuffer, 12300);
97         }
98         G=atof(rxbuffer);
99         memset(rxbuffer,0,5*sizeof(char));
100        putsUART1("VALOR NOVO DE IRRADIACAO (w/m2) [G]: ");
101        sprintf(buffer, "%.2f\r\n", G);
102        putsUART1(buffer);
103    }
104
105    if(rxbuffer[0]=='4')
106    {
107
108        memset(rxbuffer,0,5*sizeof(char));
109        putsUART1("Valor atual de tensao em circuito aberto
(V) [VOC_STC]: ");
110        sprintf(buffer, "%.2f\r\n", VOC_STC);
111        putsUART1(buffer);
112        putsUART1("INTRODUZA NOVO VALOR:\r\n");
113        while(rxbuffer[0]==0)
114        {
115            getsUART1(5, rxbuffer, 12300);
116        }
117        VOC_STC=atof(rxbuffer);
118        memset(rxbuffer,0,5*sizeof(char));
119        putsUART1("Valor novo de tensao em circuito aberto
(V) [VOC_STC]: ");
120        sprintf(buffer, "%.2f\r\n", VOC_STC);
121        putsUART1(buffer);
122
123
124        putsUART1("Valor atual de corrente de
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/comm.c

```
curto-circuito (I) [ISC_STC]: ");
125         sprintf(buffer, "%.2f\r\n", ISC_STC);
126         putsUART1(buffer);
127         putsUART1("INTRODUZA NOVO VALOR:\r\n");
128         while(rxbuffer[0]==0)
129         {
130             getsUART1(5, rxbuffer, 12300);
131         }
132         ISC_STC=atof(rxbuffer);
133         memset(rxbuffer,0,5*sizeof(char));
134         putsUART1("Valor novo de corrente de curto-circuito
(I) [ISC_STC]: ");
135         sprintf(buffer, "%.2f\r\n", ISC_STC);
136         putsUART1(buffer);
137
138         putsUART1("Valor atual de resistencia serie (ohm)
[RS]: ");
139         sprintf(buffer, "%.2f\r\n", RS);
140         putsUART1(buffer);
141         putsUART1("INTRODUZA NOVO VALOR:\r\n");
142         while(rxbuffer[0]==0)
143         {
144             getsUART1(5, rxbuffer, 12300);
145         }
146         RS=atof(rxbuffer);
147         memset(rxbuffer,0,5*sizeof(char));
148         putsUART1("Valor novo de resistencia serie (ohm)
[RS]: ");
149         sprintf(buffer, "%.2f\r\n", RS);
150         putsUART1(buffer);
151
152         putsUART1("Valor atual do coeficiente
corrente-temperatura [Ki]: ");
153         sprintf(buffer, "%.2f\r\n", Ki);
154         putsUART1(buffer);
155         putsUART1("INTRODUZA NOVO VALOR:\r\n");
156         while(rxbuffer[0]==0)
157         {
158             getsUART1(5, rxbuffer, 12300);
159         }
160         Ki=atof(rxbuffer);
161         memset(rxbuffer,0,5*sizeof(char));
162         putsUART1("Valor novo do coeficiente
corrente-temperatura [Ki]: ");
163         sprintf(buffer, "%.2f\r\n", Ki);
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/comm.c

```
164             putsUART1(buffer);
165
166             putsUART1("Valor atual do coeficiente
tensao-temperatura [Kv]: ");
167             sprintf(buffer, "%.2f\r\n", Kv);
168             putsUART1(buffer);
169             putsUART1("INTRODUZA NOVO VALOR:\r\n");
170             while(rxbuffer[0]==0)
171             {
172                 getsUART1(5, rxbuffer, 12300);
173             }
174             Kv=atof(rxbuffer);
175             memset(rxbuffer,0,5*sizeof(char));
176             putsUART1("Valor novo do coeficiente
tensao-temperatura [Kv]: ");
177             sprintf(buffer, "%.2f\r\n", Kv);
178             putsUART1(buffer);
179
180             putsUART1("Valor atual de fator de idealidade [n]:
");
181             sprintf(buffer, "%.2f\r\n", n);
182             putsUART1(buffer);
183             putsUART1("INTRODUZA NOVO VALOR:\r\n");
184             while(rxbuffer[0]==0)
185             {
186                 getsUART1(5, rxbuffer, 12300);
187             }
188             n=atof(rxbuffer);
189             memset(rxbuffer,0,5*sizeof(char));
190             putsUART1("Valor novo de fator de idealidade [n]:
");
191             sprintf(buffer, "%.2f\r\n", n);
192             putsUART1(buffer);
193
194             putsUART1("Valor atual do numero de celulas solares
[Ncell]: ");
195             sprintf(buffer, "%.2f\r\n", Ncell);
196             putsUART1(buffer);
197             putsUART1("INTRODUZA NOVO VALOR:\r\n");
198             while(rxbuffer[0]==0)
199             {
200                 getsUART1(5, rxbuffer, 12300);
201             }
202             Ncell=atof(rxbuffer);
203             memset(rxbuffer,0,5*sizeof(char));
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/comm.c

```
204             putsUART1("Valor novo do numero de celulas solares
[Ncell]: ");
205             sprintf(buffer, "%.2f\r\n", Ncell);
206             putsUART1(buffer);
207
208         }
209     }
210
211     else
212     {
213         putsUART1("ESCREVEU:\r\n");
214         putsUART1(rxbuffer);
215         putsUART1("\r\n");
216         putsUART1("ERRO - USE 'm' PARA O MENU\r\n");
217         memset(rxbuffer,0,5*sizeof(char));
218     }
219
220     _U1RXIF = 0; // clear interrupt flag
221 }
222
223 void __attribute__((__interrupt__)) _U2RXInterrupt(void)
224 {
225     if(DataReceived == 0)
226     {
227         getsUART2(80, rxbuffer2, 12300);
228
229         putsUART1("Interrupt PLC ok\r\n");
230
231         DataReceived = 1;
232     }
233
234     _U2RXIF = 0; // clear interrupt flag
235 }
236
237 void uart1_start(void)
238 {
239     U1BRG = ((Fcy/BAUDRATE)/16)-1;
240
241     U1MODEbits.ALTIO = 1; //Communicate with U1ATX and U1ARX pins
242
243     U1MODEbits.PDSEL = 0b00; //8 bit - no parity (PC)
244     U1MODEbits.STSEL = 0; //1 stop bit
245
246     _U1RXIP = 7; //max priority interrupt
247     _U1RXIF = 0; //clear interrupt flag
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/comm.c

```
248     _U1RXIE = 1;           //enable receive interrupt
249
250     U1STAbits.URXISEL = 0b00; //Receive interrupt flag bit set when a
character is received
251
252     U1MODEbits.UARTEN = 1;   //set uart 1 on
253     U1STAbits.UTXEN = 1;     //transmit enable
254 }
255
256 void uart2_start(void)
257 {
258     U2BRG = ((Fcy/BAUDRATE)/16)-1;
259
260     U2MODEbits.PDSEL = 0b01; //8 bit - even parity (PLC)
261     U2MODEbits.STSEL = 0;     //1 stop bit
262
263     _U2RXIP = 6;              //2nd max priority interrupt
264     _U2RXIF = 0;             //clear interrupt flag
265     _U2RXIE = 1;            //enable receive interrupt
266
267     U2STAbits.URXISEL = 0b00; //Receive interrupt flag bit set when a
character is received
268
269     U2MODEbits.UARTEN = 1;   //set uart 2 on
270     U2STAbits.UTXEN = 1;     //transmit enable
271 }
272
273 void SPI_start(void)
274 {
275     _TRISF3 = 0; //SDO1
276     _TRISF6 = 0; //SCK1
277     _TRISD3 = 0; //CS
278     _LATD3 = 1;
279
280     SPI1CONbits.MODE16 = 1;
281     SPI1CONbits.CKE = 1;
282     SPI1CONbits.CKP = 0;
283     SPI1CONbits.MSTEN = 1;
284     SPI1CONbits.SPRE = 0b100; //4:1
285     SPI1CONbits.PPRE = 0b01; //16:1
286
287     SPI1STATbits.SPIROV = 0;
288
289     SPI1STATbits.SPIEN = 1;
290 }
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/comm.c

291

292 void SPI\_write(int ref)

293 {

294 \_LATD3 = 0;

295 SPI1BUF = (0x7000 | ref);

296 while(SPI1STATbits.SPITBF);

297 \_\_delay32(1500);

298 \_LATD3 = 1;

299 }

C:/Users/helio/Desktop/MPLABX/Tese30.X/comm.h

```
1 #ifndef COMM_H
2 #define COMM_H
3
4 #define Pll      16
5 #define Fosc    7372800 // system clock (osc frequency)
6 #define Fcy     ( Fosc * Pll / 4 ) // peripheral bus clock (instruction
cycle frequency) 29491200 IPS
7 #define BAUDRATE 115200
8
9 extern int DataReceived;
10 extern char buffer[80], rxbuffer[80], rxbuffer2[80];
11 extern float VT, n, VOC_STC, ISC_STC, IO, RS, VOC, ISC, Ki, Kv, dT, T,
G, Ncell, VPVref, VPV, IPV;
12
13 void __attribute__((__interrupt__)) _U1RXInterrupt(void);
14
15 void __attribute__((__interrupt__)) _U2RXInterrupt(void);
16
17 void uart1_start(void);
18
19 void uart2_start(void);
20
21 void SPI_start(void);
22
23 void SPI_write(int ref);
24
25 #endif
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/timer.c

```
1 #include <xc.h>
2 #include "timer.h"
3
4 char tbuffer[80];
5 int countT1=0;
6
7 void __attribute__((__interrupt__)) _T1Interrupt(void)
8 {
9     countT1++;
10    if (IPV>ISC)
11        IPV=ISC-0.001;
12    if(countT1 % 2 == 0)
13    {
14        if(DataReceived == 0)
15        {
16            sprintf(tbuffer, "ST%5.2f %5.2f", VPV, IPV);
17            putsUART2(tbuffer);
18        }
19    }
20    if(countT1>=4)
21    {
22        sprintf(tbuffer, "%.2f", VPV);
23        putsUART1("VPV: ");
24        putsUART1(tbuffer);
25        putsUART1("\r\n");
26
27        sprintf(tbuffer, "%.2f", IPV);
28        putsUART1("IPV: ");
29        putsUART1(tbuffer);
30        putsUART1("\r\n");
31
32        sprintf(tbuffer, "%.2f", VPVref);
33        putsUART1("REFERENCIA VPV: ");
34        putsUART1(tbuffer);
35        putsUART1("\r\n");
36
37        countT1=0;
38    }
39    _T1IF = 0;
40 }
41
42 void timer1_start(void)
43 {
44     T1CON = 0; // Clear Timer 1 configuration
45     T1CONbits.TCKPS = 3; // Set timer 1 prescaler (0=1:1, 1=1:8,
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/timer.c

2=1:64, 3=1:256)

```
46     PR1 = 57600;           // Set Timer 1 period (max value is 65535) //  
29491200 IPS -> 0,1s = 11520  
47     _T1IP = 1;           // Set Timer 1 interrupt priority  
48     _T1IF = 0;           // Clear Timer 1 interrupt flag  
49     _T1IE = 1;           // Enable Timer 1 interrupt  
50     T1CONbits.TON = 1;    // Turn on Timer 1  
51 }
```

C:/Users/helio/Desktop/MPLABX/Tese30.X/timer.h

```
1 #ifndef TIMER_H
2 #define TIMER_H
3
4 extern float VT, n, VOC_STC, ISC_STC, IO, RS, VOC, ISC, Ki, Kv, dT, T,
G, Ncell, VPVref, VPV, IPV, VPVref5V;
5 extern int VPVrefWord;
6 extern int DataReceived;
7
8 void timer1_start(void);
9
10 void __attribute__((__interrupt__)) _T1Interrupt(void);
11
12 #endif
```

## Data types: rs232types

TYPE

```
T_RS232 :   STRUCT
  protocol   :INT; (* 0: transparent, data transmission without protocol *)
  baudrate   :INT; (* 3: 1200kBit, 5: 2400kBit, 6: 4800kBit, 7: 9600kBit, 8: 19200k
Bit, 9: 38400kBit, 10: 57600kBit, 11: 115200kBit *)
  databits   :INT; (* 4: 8 data bits, even parity *)
  stopbits   :INT; (* 1: 1 stop bit *)
  flowcontrol :INT; (* 0: not supported *)
  error_pattern :INT; (* not supported *)
  first_delimiter :INT; (* not supported *)
  second_delimiter:INT; (* not supported *)
  XON_pattern :INT; (* not supported *)
  XOFF_pattern :INT; (* not supported *)
END_STRUCT;
END_TYPE
```

TYPE

```
Tmessage : array [1..80] of BYTE;
END_TYPE
```

## Data types: sys\_flag\_types

TYPE

```
Task_Name_Typ : ARRAY[0..9] OF BYTE;

Extended_Task_Info : STRUCT
  TaskName           : Task_Name_Typ; (* Name of the Task as ARRAY OF BYTE, NULL te
rminated *)
  TaskPrio           : INT;           (* Priority of the task *)
  undocumented_0     : INT;
  TaskPeriod         : INT;           (* Period of the task in milliseconds *)
  TaskStack          : INT;           (* Stack size of the task *)
  unused_1           : INT;
  TaskWatchdog       : INT;           (* Watchdog time in milliseconds *)
  undocumented_2     : INT;
  undocumented_3     : INT;
  undocumented_4     : INT;
  CurDuration        : INT;           (* Current task duration in ticks including p
reemption *)
  MinDuration        : INT;           (* Minimum task duration in ticks including p
reemption *)
  MaxDuration        : INT;           (* Maximum task duration in ticks including p
reemption *)
  undocumented_5     : INT;
  CurDelay           : INT;           (* Current task delay in ticks including pree
mption *)
  MinDelay           : INT;           (* Minimum task delay in ticks including pree
mption *)
  MaxDelay           : INT;           (* Maximum task delay in ticks including pree
mption *)
END_STRUCT;
END_TYPE
```

TYPE

```
Task_Name_eCLR : ARRAY[0..35] OF BYTE;

Task_Info_eCLR : STRUCT
  TaskStack          : INT;           (* Stack size of the task *)
  TaskPrio           : INT;           (* Priority of the task *)
  TaskPeriod_us      : DINT;          (* Period of the task in microseconds *)
  TaskWatchdog_us    : DINT;          (* Watchdog time in microseconds *)
  TaskPeriod         : INT;           (* Period of the task in milliseconds *)
  TaskWatchdog       : INT;           (* Watchdog time in milliseconds *)
  MinDuration_us     : DINT;          (* Minimum task duration in microseconds includi
ng preemption *)
  MaxDuration_us     : DINT;          (* Maximum task duration in microseconds includi
ng preemption *)
  CurDuration_us     : DINT;          (* Current task duration in microseconds includi
ng preemption *)
  MinDelay_us        : DINT;          (* Minimum task delay in microseconds including
```

<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG          Flachsmarktstraße 8          32825 Blomberg          Germany</p>	<p>Sheet number          1</p>
<p>Current POU:</p>	<p><b>Tese7</b></p>	<p>This copy printed out at:          26/02/2023 19:30:46</p>

```

preemption *)
    MaxDelay_us      : DINT;          (* Maximum task delay in microseconds including
preemption *)
    CurDelay_us     : DINT;          (* Current task delay in microseconds including
preemption *)
    MinDuration     : INT;           (* Minimum task duration in milliseconds includi
ng preemption *)
    MaxDuration     : INT;           (* Maximum task duration in milliseconds includi
ng preemption *)
    CurDuration     : INT;           (* Current task duration in milliseconds includi
ng preemption *)
    MinDelay        : INT;           (* Minimum task delay in milliseconds including
preemption *)
    MaxDelay        : INT;           (* Maximum task delay in milliseconds including
preemption *)
    CurDelay        : INT;           (* Current task delay in milliseconds including
preemption *)
    unused_1        : DINT;
    unused_2        : DINT;
    unused_3        : DINT;
    unused_4        : DINT;
    unused_5        : DINT;
    unused_6        : DINT;
    unused_7        : DINT;
    unused_8        : DINT;
    unused_9        : DINT;
    TNameMaxSize    : INT;
    TNameSize       : INT;
    TaskName        : Task_Name_eCLR;
END_STRUCT;
END_TYPE

TYPE
    PND_IO_6       : ARRAY[0..5] OF BYTE;
    PND_IO_7       : ARRAY[0..6] OF BYTE;
    PND_IO_32      : ARRAY[0..31] OF BYTE;
    PND_IO_64      : ARRAY[0..63] OF BYTE;
    PND_IO_128     : ARRAY[0..127] OF BYTE;
    PND_IO_256     : ARRAY[0..255] OF BYTE;
    PND_IO_512     : ARRAY[0..511] OF BYTE;
END_TYPE

TYPE
    Redundancy_OPC_Struct : ARRAY[0..2] OF INT;
END_TYPE

TYPE
    MBT_STATION_DIAG : STRUCT
        Status          : DWORD; (* Bit 0 = Station is online*)
                        (* Bit 1 = Exception exits *)

        IP              : DWORD;
        OfflineCounter  : DWORD; (* Incremented by every shutdown of this station. *)

        NetCycleAvg    : DWORD; (* ms. NetInCycleAvg + NetOutCycleAvg *)
        NetCycleMin    : DWORD; (* ms. *)
        NetCycleMax    : DWORD; (* ms. *)

        NetInCycleAvg  : DWORD; (* ms. *)
        NetInCycleMin  : DWORD; (* ms. *)
        NetInCycleMax  : DWORD; (* ms. *)

        NetOutCycleAvg : DWORD; (* ms. *)
        NetOutCycleMin : DWORD; (* ms. *)
        NetOutCycleMax : DWORD; (* ms. *)

        InternalError1 : DWORD; (* For internal use of Phoenix Contact! *)
        InternalError2 : DWORD; (* For internal use of Phoenix Contact! *)

        ExceptionCounter : DWORD; (* Incremented by every Modbus exception received on thi
s station. *)
        DiagReserved1   : BYTE;  (* Reserved for a later use! *)
    END_STRUCT;
END_TYPE

```

<b>PC WORX</b>	Phoenix Contact GmbH & Co. KG Flachsmarktstraße 8 32825 Blomberg Germany	Sheet number 2
Current POU:	<b>Tese7</b>	This copy printed out at: 26/02/2023 19:30:46

```

        DiagReserved2      : BYTE;  (* Reserved for a later use! *)
        UnitID             : BYTE;  (* Modbus Unit Identifier *)
        DiagReserved3     : BYTE;  (* Reserved for a later use! *)
    END_STRUCT;

    MBT_STATION_CONTROL : STRUCT
        ctrlStatistics     : BYTE;  (* Bit 0 = Clear statistics *)
        ControlReserved1   : BYTE;  (* Reserved for a later use! *)
        ControlReserved2   : WORD;  (* Reserved for a later use! *)
        ControlReserved3   : DWORD; (* Reserved for a later use! *)
    END_STRUCT;
END_TYPE

TYPE
    MBS_SERVER_STATUS : STRUCT
        AcceptCnt          : DWORD;
        AcceptErrorCnt     : DWORD;
        ReceiveErrorCnt    : DWORD;
        ModbusRequestCnt   : DWORD;
        ModbusExceptionCnt : DWORD;

        SocketErrorCode    : WORD;
        TcpConnectedClients : WORD;
        LastExceptionRegister : WORD;
        LastExceptionRegLength : WORD;

        LastExceptionNode  : BYTE;
        LastExceptionFunction : BYTE;
        LastExceptionCode  : BYTE;
        reserved            : BYTE;
    END_STRUCT;
END_TYPE

TYPE
    MBS_SERVER_CONTROL : STRUCT
        Flags              : BYTE;
        reserved1          : BYTE;
        reserved2          : BYTE;
        reserved3          : BYTE;
    END_STRUCT;
END_TYPE

TYPE
    MBS_NODE_STATUS : STRUCT
        ModbusRequestCnt   : DWORD;
        ModbusExceptionCnt : DWORD;
        Flags               : DWORD;
        reserved            : DWORD;
    END_STRUCT;
END_TYPE

TYPE
    OPCUA_STATUS_INFO : STRUCT
        restartCount       : UDINT; (* each reset and clear cache is counted *)
        monitoredVariablesCount : UDINT; (* use IdataSource.GetSubscribedVariablesCount *)
        sessionsCount      : UDINT; (* <-- get Data from ServerDiagnostic method GetTotalSubscribedVariablesCount *)
        isServerEnabled     : UDINT; (* fetched by conf API *)
        isLicensePresent    : UDINT; (* fetched by lic API *)
    END_STRUCT;
END_TYPE

TYPE
    IEC61850_TIMESTAMP : STRUCT
        Seconds            : DWORD;
        Fractions          : DWORD;
        TimeQuality        : DWORD;
    END_STRUCT;
END_TYPE

TYPE
    PROFISAFE_DIAG_HEADER : STRUCT

```

<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG                  Flachsmarktstraße 8                  32825 Blomberg                  Germany</p>	<p>Sheet number                  3</p>
<p>Current POU:</p>	<p>Tese7</p>	<p>This copy printed out at:                  26/02/2023 19:30:46</p>

```

        NumberOfFConn      : UINT; (* number of F-Connections, max 256 *)
        PLCHostStatusByte  : BYTE; (* status byte of Standard PLC *)
        SPNSAgentStatusByte : BYTE; (* status byte of SPNS agent *)
        CycleCount        : UINT;  (* cycle count off the spns agent (update the shm
diag area) *)
    END_STRUCT;

    F_MESSAGE : ARRAY[0..15] OF BYTE; (* F-message: 16 bytes *)

    F_DEVICE_DIAG : STRUCT
        FAddress      : WORD;      (* F address of device *)
        F_WD_Time     : WORD;      (* F WD time of device *)
        F_DataLenIn   : BYTE;      (* length of IN F data *)
        F_MessageIn   : F_MESSAGE; (* IN F data array *)
        F_DataLenOut  : BYTE;      (* length of OUT F data *)
        F_MessageOut  : F_MESSAGE; (* OUT F data array *)
    END_STRUCT;

    DEVICE_DIAG_LIST : ARRAY[0..255] OF F_DEVICE_DIAG; (* array of F_DEVICE_DIAG structur
: 256 elem *)

    PROFISAFE_DIAG : STRUCT
        ProfisafeDiagHeader : PROFISAFE_DIAG_HEADER; (* header *)
        FDeviceDiagnostics  : DEVICE_DIAG_LIST;      (* list of diagnostics *)
    END_STRUCT;
END_TYPE

TYPE
    PROFISAFE_RECORD : STRUCT
        CodeName          : DWORD; (* F_S/D_Address *)
        ProfiSafeDiagBitField : DWORD;
        SrtMin            : UINT;   (* minimal safety round trip time
of the connection *)
        SrtMax            : UINT;   (* maximal safety round trip time
of the connection *)
        SrtCur            : UINT;   (* current safety round trip time
of the connection *)
        FwdTime           : UINT;   (* parameterized f-wd-time *)
        ValidReg          : UINT;   (* the values in this profisafe r
econd are balid or not *)
        Reserved          : UINT;   (* to fill up the structure to a
size x*4 *)
    END_STRUCT;

    PROFISAFE_RECORD_ARRAY : ARRAY[0..299] OF PROFISAFE_RECORD;

    PROFISAFE_DIAG_OUT : STRUCT
        MaxNumberOfProfiSafeRecords : UINT; (* max. number of possible profisafe records *)
        NumberOfProfiSafeRecords    : UINT; (* current number of possible profisafe record
s in array *)
        ProfiSafeRecord              : PROFISAFE_RECORD_ARRAY;
        ProfiSafeGlobalRecord        : DWORD;
    END_STRUCT;
END_TYPE

```

### Data types: compile

```

TYPE
    ensure_compiled : ARRAY[0..1] OF BOOL;
END_TYPE

```

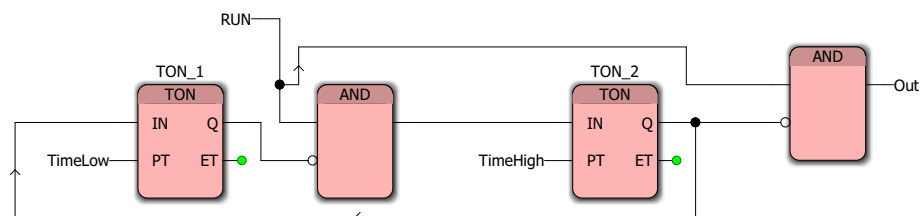
<b>PC WORX</b>	Phoenix Contact GmbH & Co. KG Flachsmarktstraße 8 32825 Blomberg Germany	Sheet number 4
Current POU:	<b>Tese7</b>	This copy printed out at: 26/02/2023 19:30:46

POU: MUV, Description: MUVT

POU: MUV, Variables: MUVV

Name	Type	Usage	Description	Init	sta	DDP	TB	Hidden	as	ult	Hidden	ant	pt	OnF	Device	Signal	Alias	
Default																		
RUN	BOOL	VAR I...																
TimeLow	TIME	VAR I...																
TimeHigh	TIME	VAR I...																
Out	BOOL	VAR ...																
TON_1	TON	VAR																
TON_2	TON	VAR																

POU: MUV, Worksheet: MUV



<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG          Flachsmarktstraße 8          32825 Blomberg          Germany</p>	<p>Sheet number 5</p>
<p>Current POU: MUV</p>	<p>Tese7</p>	<p>This copy printed out at: 26/02/2023 19:30:46</p>

POU: MUV, Local Cross References

Variable	Worksheet	Access	Command	I/O Addr.	Type	Init.Value	Comment	Count	Page
Out	MUV	Write			BOOL			1	5
RUN	MUV	Read			BOOL			1	5
TimeHigh	MUV	Read			TIME			1	5
TimeLow	MUV	Read			TIME			1	5
TON_1	MUV	Call			TON			1	5
TON_2	MUV	Call			TON			1	5

<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG                  Flachsmarktstraße 8                  32825 Blomberg                  Germany</p>	<p>Sheet number 6</p>
<p>Current POU: MUV</p>	<p><b>Tese7</b></p>	<p>This copy printed out at: 26/02/2023 19:30:46</p>

POU: Communication, Description: CommunicationT

POU: Communication, Variables: CommunicationV

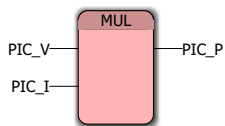
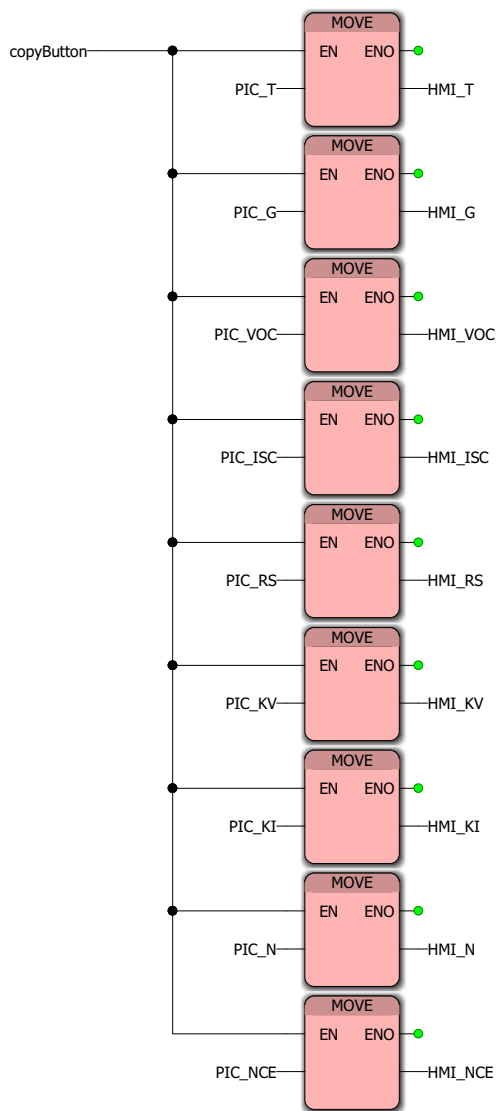
Name	Type	Usage	Description	Init	eta	DDP	TB	Hidden	as	ult	Hidden	ant	ptOnF	Device	Signal	Alias
Default																
RS232_INIT_1	RS232...	VAR														
RS232_SEND...	RS232...	VAR														
RS232_RECE...	RS232...	VAR														
params	T_RS2...	VAR														
MsgSend	STRING	VAR														
dat_count	INT	VAR														
var1	INT	VAR														
PIC_T	REAL	VAR				x										
PIC_G	REAL	VAR				x										
PIC_VOC	REAL	VAR				x										
PIC_ISC	REAL	VAR				x										
PIC_RS	REAL	VAR				x										
PIC_KV	REAL	VAR				x										
PIC_KI	REAL	VAR				x										
PIC_N	REAL	VAR				x										
PIC_NCE	REAL	VAR				x										
MsgSendB	Tmess...	VAR														
MsgReceiveB	Tmess...	VAR	E...													
HMI_T	REAL	VAR				x										
HMI_G	REAL	VAR				x										
HMI_VOC	REAL	VAR				x										
HMI_ISC	REAL	VAR				x										
HMI_RS	REAL	VAR				x										
HMI_KV	REAL	VAR				x										
HMI_KI	REAL	VAR				x										
HMI_N	REAL	VAR				x										
HMI_NCE	REAL	VAR				x										
MsgSendLen	INT	VAR														
MUV_1	MUV	VAR														
doneReceive	BOOL	VAR														
PIC_V	REAL	VAR				x										
PIC_I	REAL	VAR				x										
PIC_P	REAL	VAR				x										
copyButton	BOOL	VAR				x										
BUF_TO_ST...	BUF_T...	VAR														
MsgReceive	STRING	VAR	E...													
TOF_1	TOF	VAR														
R_TRIG_1	R_TRIG	VAR														
MsgCommand	STRING	VAR														
writeT	BOOL	VAR				x										
writeG	BOOL	VAR				x										
writeVOC	BOOL	VAR				x										
writeISC	BOOL	VAR				x										
writeRS	BOOL	VAR				x										
writeKV	BOOL	VAR				x										
writeKI	BOOL	VAR				x										
writeN	BOOL	VAR				x										
writeNCE	BOOL	VAR				x										
readT	BOOL	VAR				x										
readG	BOOL	VAR				x										
readVOC	BOOL	VAR				x										
readISC	BOOL	VAR				x										
readRS	BOOL	VAR				x										
readKV	BOOL	VAR				x										
readKI	BOOL	VAR				x										
readN	BOOL	VAR				x										
readNCE	BOOL	VAR				x										
writeALL	BOOL	VAR				x										
R_TRIG_2	R_TRIG	VAR														
TOF_2	TOF	VAR														
R_TRIG_3	R_TRIG	VAR														
TOF_3	TOF	VAR														
R_TRIG_4	R_TRIG	VAR														
TOF_4	TOF	VAR														
R_TRIG_5	R_TRIG	VAR														
TOF_5	TOF	VAR														
R_TRIG_6	R_TRIG	VAR														
TOF_6	TOF	VAR														
R_TRIG_7	R_TRIG	VAR														

<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG                  Flachsmarktstraße 8                  32825 Blomberg                  Germany</p>	<p>Sheet number 7</p>
<p>Current POU: Communication</p>	<p>Tese7</p>	<p>This copy printed out at: 26/02/2023 19:30:46</p>



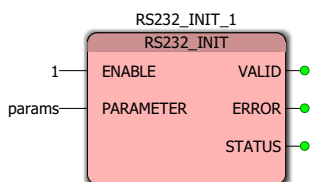




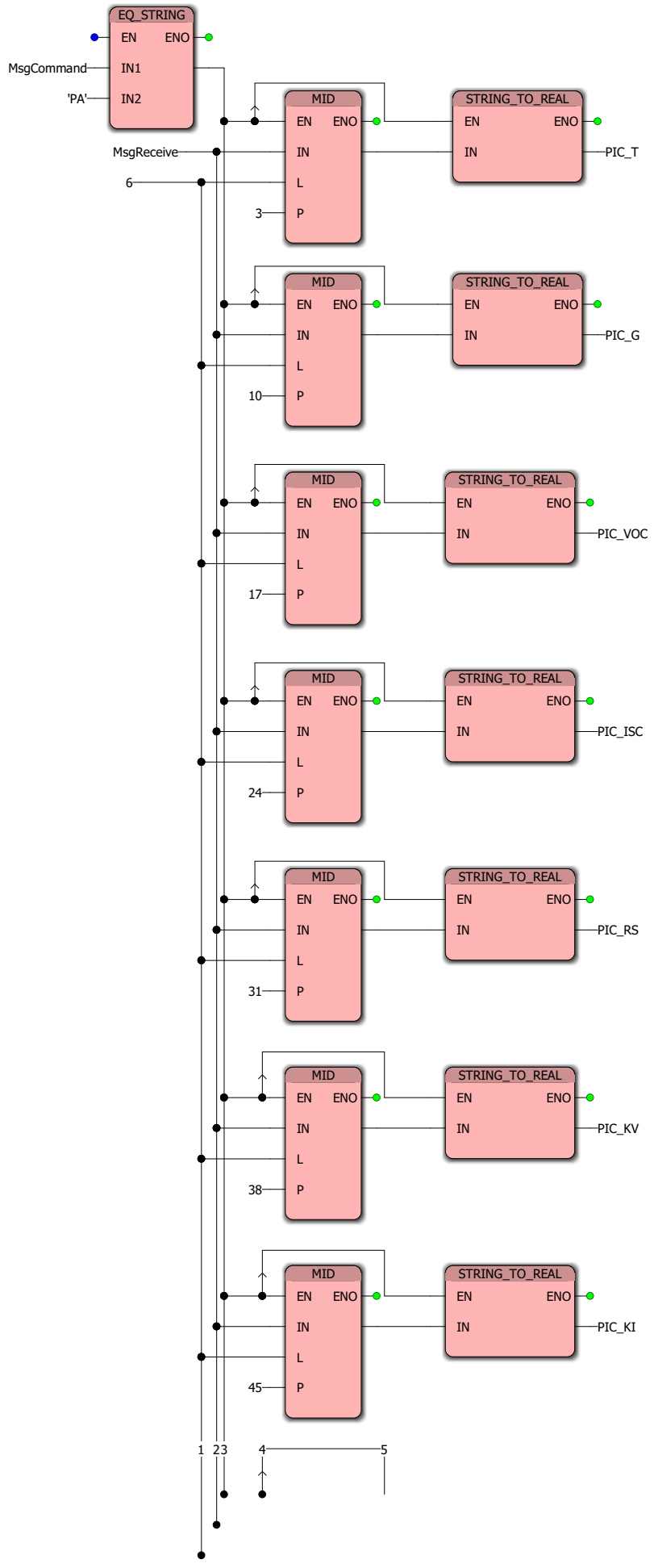


(\*RS-232 format: 115200-8-1-even\*)

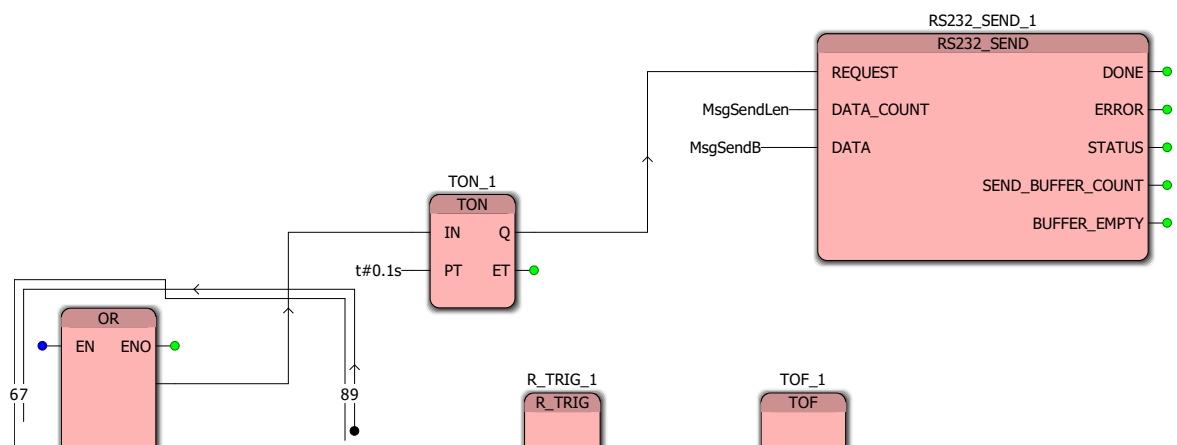
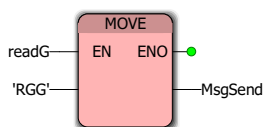
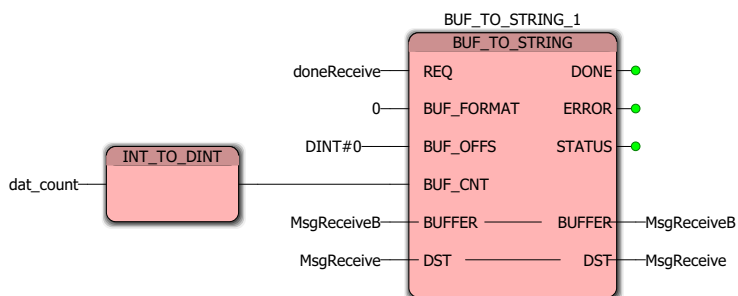
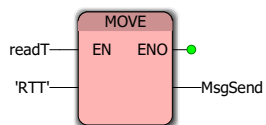
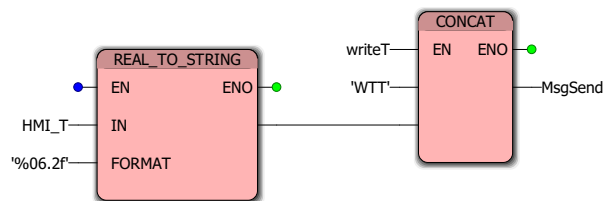
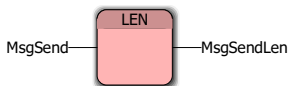
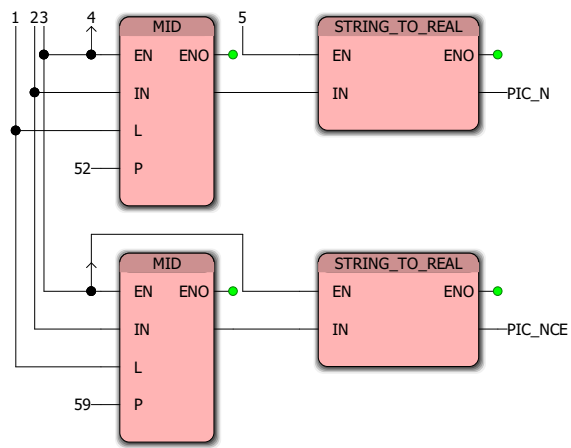
- 0 ——— params.protocol
- 11 ——— params.baudrate
- 4 ——— params.databits
- 1 ——— params.stopbits
- 0 ——— params.flowcontrol



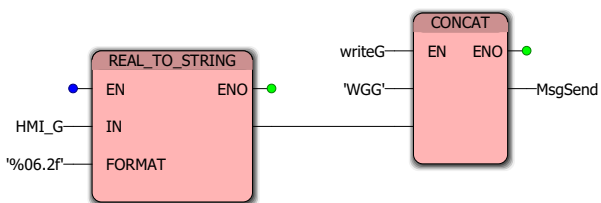
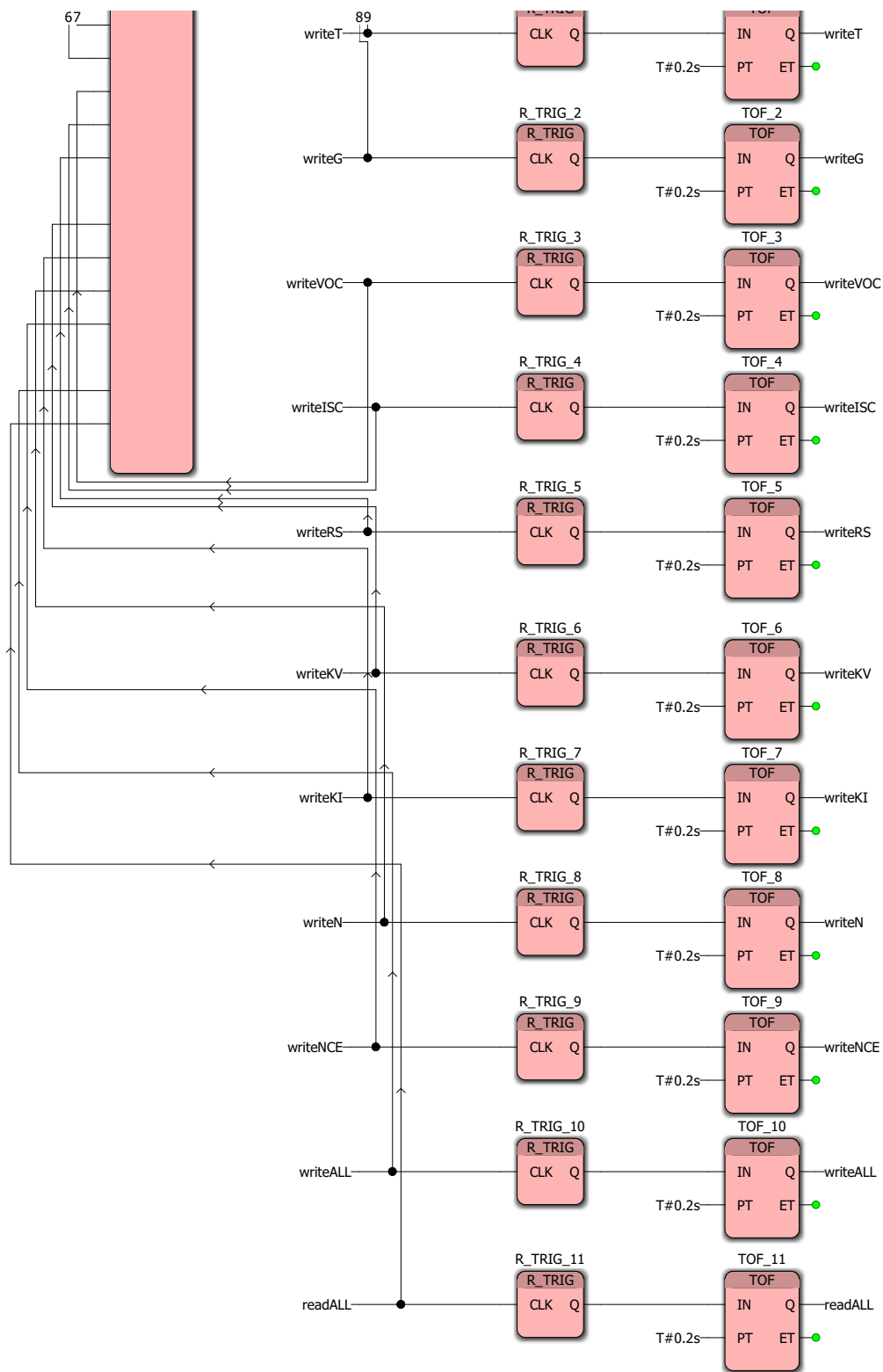
<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG          Flachsmarktstraße 8          32825 Blomberg          Germany</p>	<p>Sheet number          11</p>
<p>Current POU:          Communication</p>	<p><b>Tese7</b></p>	<p>This copy printed out at:          26/02/2023 19:30:47</p>



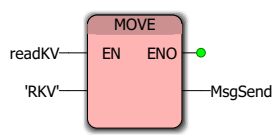
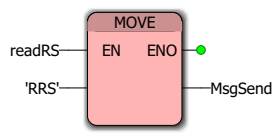
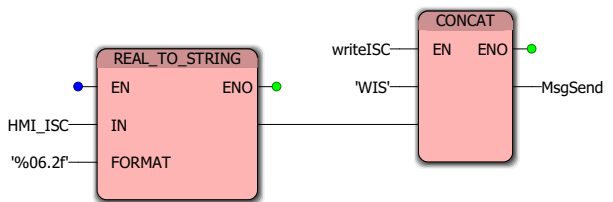
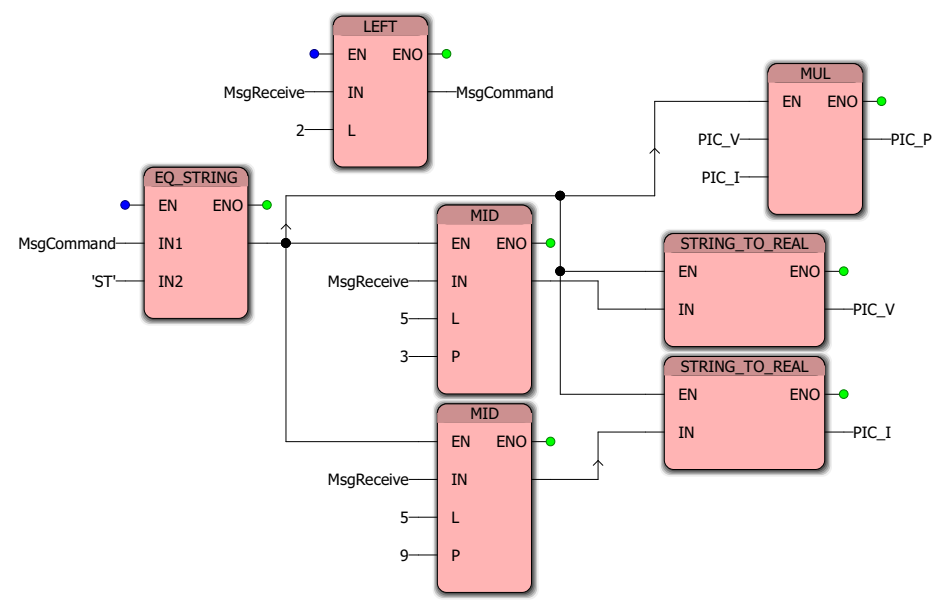
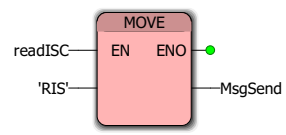
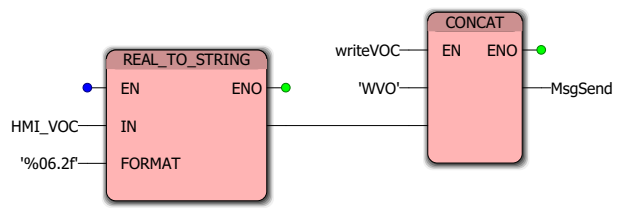
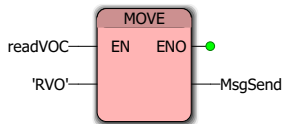
<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG          Flachsmarktstraße 8          32825 Blomberg          Germany</p>	<p>Sheet number          12</p>
<p>Current POU:          Communication</p>	<p>Tese7</p>	<p>This copy printed out at:          26/02/2023 19:30:47</p>



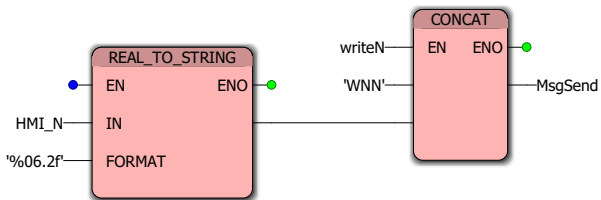
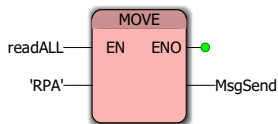
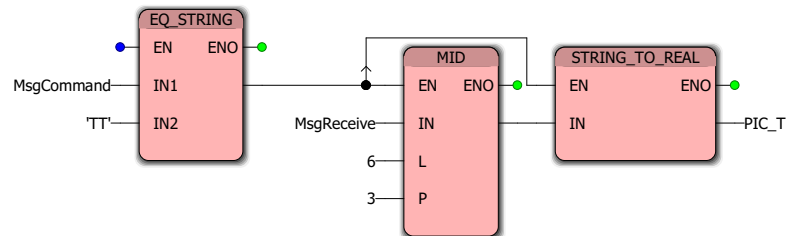
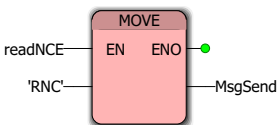
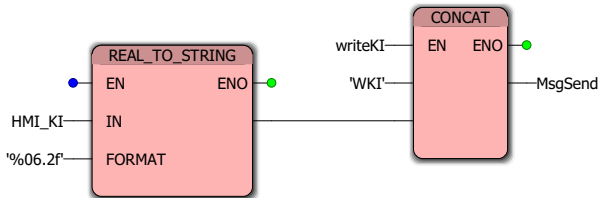
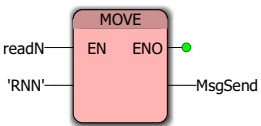
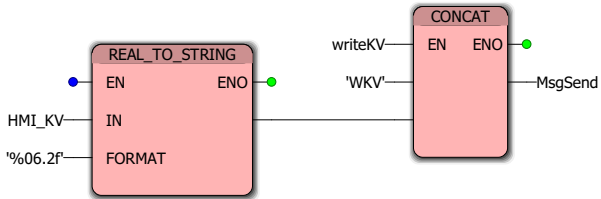
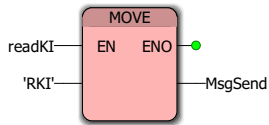
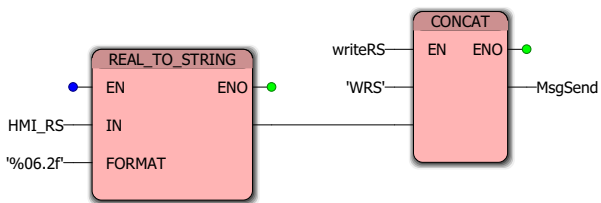
<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG          Flachsmarktstraße 8          32825 Blomberg          Germany</p>	<p>Sheet number          13</p>
<p>Current POU:          Communication</p>	<p><b>Tese7</b></p>	<p>This copy printed out at:          26/02/2023 19:30:47</p>



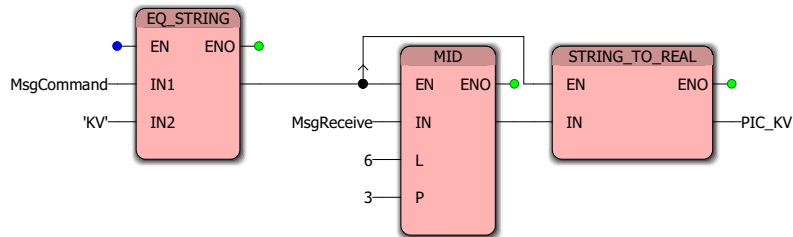
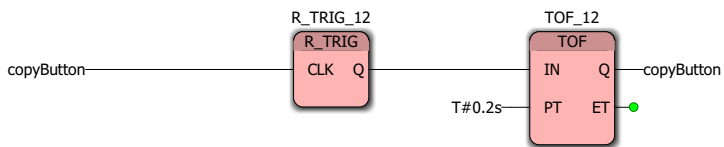
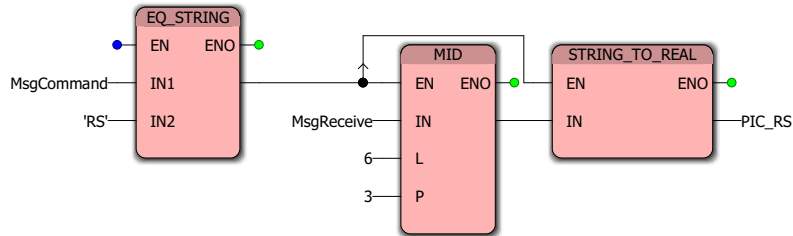
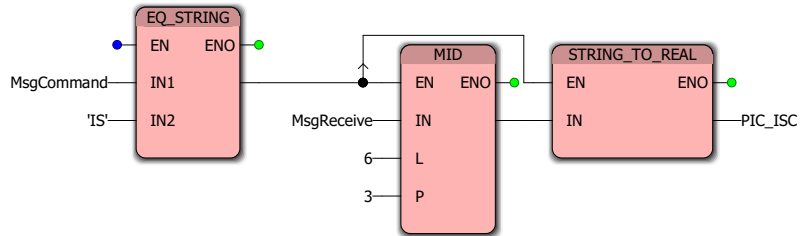
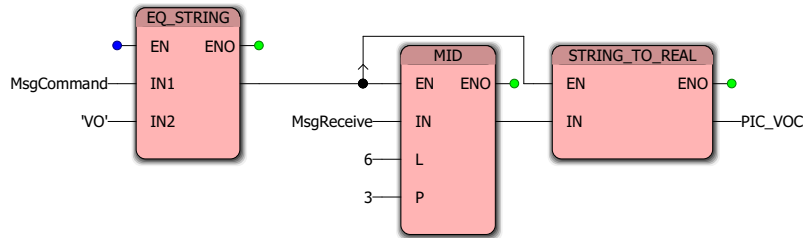
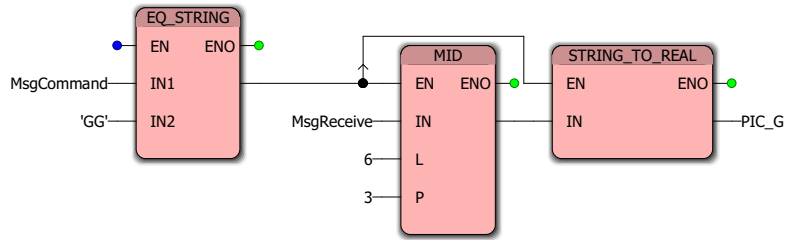
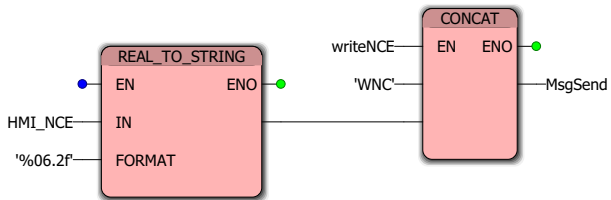
<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG          Flachsmarktstraße 8          32825 Blomberg          Germany</p>	<p>Sheet number          14</p>
<p>Current POU:          Communication</p>	<p><b>Tese7</b></p>	<p>This copy printed out at:          26/02/2023 19:30:48</p>

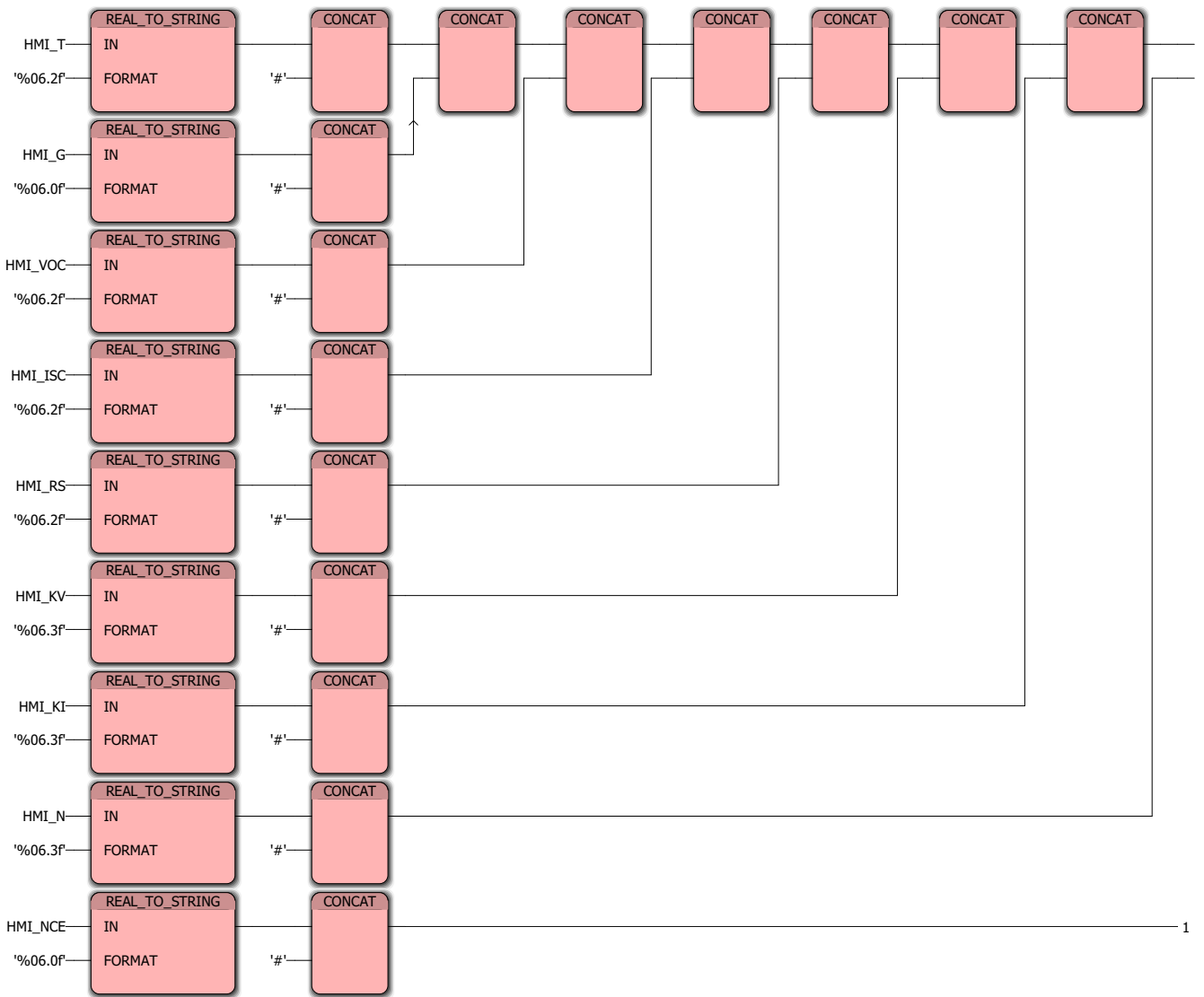


<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG          Flachsmarktstraße 8          32825 Blomberg          Germany</p>	<p>Sheet number          15</p>
<p>Current POU:          Communication</p>	<p>Tese7</p>	<p>This copy printed out at:          26/02/2023 19:30:48</p>

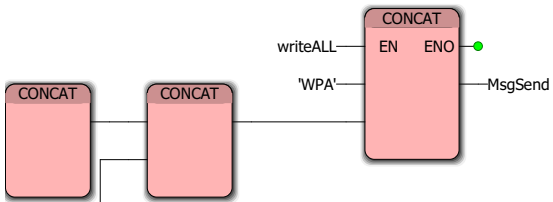


<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG          Flachsmarktstraße 8          32825 Blomberg          Germany</p>	<p>Sheet number          16</p>
<p>Current POU:          Communication</p>	<p>Tese7</p>	<p>This copy printed out at:          26/02/2023 19:30:48</p>

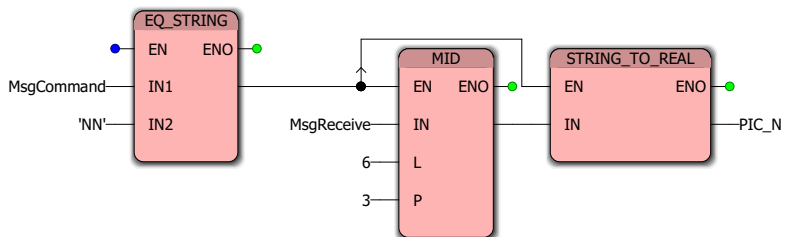
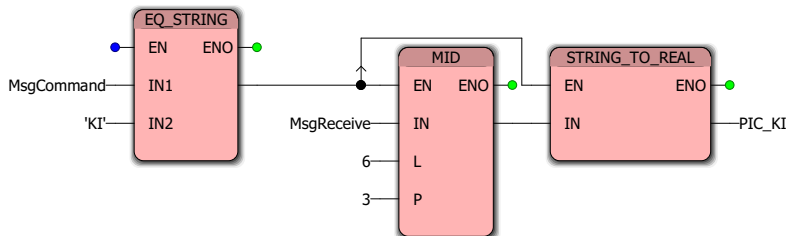




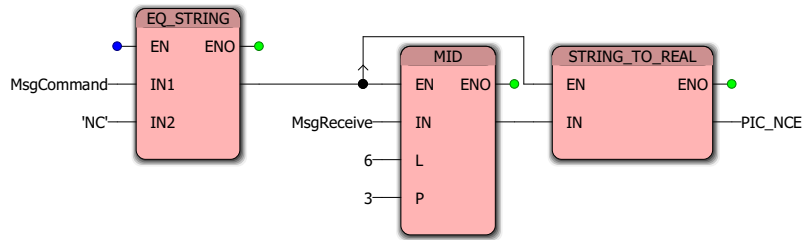
<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG          Flachsmarktstraße 8          32825 Blomberg          Germany</p>	<p>Sheet number          18</p>
<p>Current POU:          Communication</p>	<p>Tese7</p>	<p>This copy printed out at:          26/02/2023 19:30:48</p>



1



<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG          Flachsmarktstraße 8          32825 Blomberg          Germany</p>	<p>Sheet number          19</p>
<p>Current POU:          Communication</p>	<p><b>Tese7</b></p>	<p>This copy printed out at:          26/02/2023 19:30:48</p>



<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG          Flachsmarktstraße 8          32825 Blomberg          Germany</p>	<p>Sheet number          20</p>
<p>Current POU:          Communication</p>	<p><b>Tese7</b></p>	<p>This copy printed out at:          26/02/2023 19:30:48</p>

POU: Communication, Local Cross References

Variable	Worksheet	Access	Command	I/O Addr.	Type	Init.Value	Comment	Count	Page
BUF_TO_S...	Communic...	Call			BUF_T...			1	16
copyButton	Communic...	Read			BOOL			1	12
copyButton	Communic...	Write			BOOL			1	19
copyButton	Communic...	Read			BOOL			1	14
dat_count	Communic...	Write			INT			1	13
dat_count	Communic...	Read			INT			1	15
doneReceive	Communic...	Read			BOOL			1	20
doneReceive	Communic...	Write			BOOL			1	19
HMI_G	Communic...	Read			REAL			1	15
HMI_G	Communic...	Read			REAL			1	17
HMI_G	Communic...	Write			REAL			1	18
HMI_ISC	Communic...	Read			REAL			2	15
HMI_ISC	Communic...	Write			REAL			1	19
HMI_KI	Communic...	Read			REAL			1	13
HMI_KI	Communic...	Read			REAL			1	18
HMI_KI	Communic...	Write			REAL			1	12
HMI_KV	Communic...	Write			REAL			1	12
HMI_KV	Communic...	Read			REAL			1	14
HMI_KV	Communic...	Read			REAL			1	13
HMI_N	Communic...	Read			REAL			1	13
HMI_N	Communic...	Write			REAL			1	12
HMI_N	Communic...	Read			REAL			1	18
HMI_NCE	Communic...	Read			REAL			1	12
HMI_NCE	Communic...	Read			REAL			1	13
HMI_NCE	Communic...	Write			REAL			1	12
HMI_RS	Communic...	Read			REAL			1	14
HMI_RS	Communic...	Read			REAL			1	15
HMI_RS	Communic...	Write			REAL			1	12
HMI_T	Communic...	Read			REAL			1	15
HMI_T	Communic...	Read			REAL			1	16
HMI_T	Communic...	Write			REAL			1	18
HMI_VOC	Communic...	Read			REAL			1	14
HMI_VOC	Communic...	Write			REAL			1	18
HMI_VOC	Communic...	Read			REAL			1	15
MsgComm...	Communic...	Write			STRING			1	12
MsgComm...	Communic...	Read			STRING			1	12
MsgComm...	Communic...	Read			STRING			2	14
MsgComm...	Communic...	Read			STRING			1	11
MsgComm...	Communic...	Read			STRING			3	13
MsgComm...	Communic...	Read			STRING			1	17
MsgComm...	Communic...	Read			STRING			1	18
MsgComm...	Communic...	Read			STRING			1	19
MsgComm...	Communic...	Read			STRING			1	20
MsgSend	Communic...	Write			STRING			5	14
MsgSend	Communic...	Write			STRING			2	18
MsgSend	Communic...	Write			STRING			1	12
MsgSend	Communic...	Write			STRING			1	15
MsgSend	Communic...	Write			STRING			6	11
MsgSend	Communic...	Write			STRING			1	17
MsgSend	Communic...	Read			STRING			1	19
MsgSend	Communic...	Write			STRING			1	19
MsgSend	Communic...	Write			STRING			2	16
MsgSend	Communic...	Read			STRING			1	16
MsgSend	Communic...	Write			STRING			1	10
MsgSendB	Communic...	Read			Tmess...			1	17
MsgSendB[1]	Communic...	Read			Tmess...			1	17
MsgSendLen	Communic...	Read			INT			2	17

<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG                  Flachsmarktstraße 8                  32825 Blomberg                  Germany</p>	<p>Sheet number                  21</p>
<p>Current POU:                  Communication</p>	<p><b>Tese7</b></p>	<p>This copy printed out at:                  26/02/2023 19:30:48</p>

Variable	Worksheet	Access	Command	I/O Addr.	Type	Init.Value	Comment	Count	Page
MsgSendLen	Communic...	Write			INT			1	19
MUV_1	Communic...	Call			MUV			1	19
params	Communic...	Read			T_RS232			1	11
params.bau...	Communic...	Write			T_RS232			1	10
params.dat...	Communic...	Write			T_RS232			1	13
params.flo...	Communic...	Write			T_RS232			1	16
params.pro...	Communic...	Write			T_RS232			1	16
params.sto...	Communic...	Write			T_RS232			1	16
PIC_G	Communic...	Write			REAL			1	14
PIC_G	Communic...	Write			REAL			1	13
PIC_G	Communic...	Read			REAL			1	12
PIC_I	Communic...	Read			REAL			1	14
PIC_I	Communic...	Read			REAL			1	17
PIC_I	Communic...	Write			REAL			1	15
PIC_ISC	Communic...	Write			REAL			2	14
PIC_ISC	Communic...	Read			REAL			1	14
PIC_KI	Communic...	Write			REAL			1	14
PIC_KI	Communic...	Write			REAL			1	13
PIC_KI	Communic...	Read			REAL			1	12
PIC_KV	Communic...	Read			REAL			1	12
PIC_KV	Communic...	Write			REAL			1	14
PIC_KV	Communic...	Write			REAL			1	11
PIC_N	Communic...	Write			REAL			1	14
PIC_N	Communic...	Read			REAL			1	12
PIC_N	Communic...	Write			REAL			1	11
PIC_NCE	Communic...	Write			REAL			1	17
PIC_NCE	Communic...	Read			REAL			1	12
PIC_NCE	Communic...	Write			REAL			1	14
PIC_P	Communic...	Write			REAL			1	14
PIC_P	Communic...	Write			REAL			1	15
PIC_RS	Communic...	Read			REAL			1	12
PIC_RS	Communic...	Write			REAL			2	14
PIC_T	Communic...	Write			REAL			1	14
PIC_T	Communic...	Write			REAL			1	13
PIC_T	Communic...	Read			REAL			1	18
PIC_V	Communic...	Write			REAL			1	19
PIC_V	Communic...	Read			REAL			1	12
PIC_V	Communic...	Read			REAL			1	14
PIC_VOC	Communic...	Write			REAL			1	18
PIC_VOC	Communic...	Write			REAL			1	14
PIC_VOC	Communic...	Read			REAL			1	12
R_TRIG_1	Communic...	Call			R_TRIG			1	12
R_TRIG_10	Communic...	Call			R_TRIG			1	16
R_TRIG_11	Communic...	Call			R_TRIG			1	11
R_TRIG_12	Communic...	Call			R_TRIG			1	13
R_TRIG_2	Communic...	Call			R_TRIG			1	17
R_TRIG_3	Communic...	Call			R_TRIG			1	17
R_TRIG_4	Communic...	Call			R_TRIG			1	17
R_TRIG_5	Communic...	Call			R_TRIG			1	17
R_TRIG_6	Communic...	Call			R_TRIG			1	17
R_TRIG_7	Communic...	Call			R_TRIG			1	17
R_TRIG_8	Communic...	Call			R_TRIG			1	17
R_TRIG_9	Communic...	Call			R_TRIG			1	19
readALL	Communic...	Read			BOOL			1	17
readALL	Communic...	Read			BOOL			1	15
readALL	Communic...	Write			BOOL			1	15
readG	Communic...	Read			BOOL			1	10
readISC	Communic...	Read			BOOL			1	11
readKI	Communic...	Read			BOOL			1	15

<b>PC WORX</b>	Phoenix Contact GmbH & Co. KG Flachsmarktstraße 8 32825 Blomberg Germany	Sheet number 22
Current POU: Communication	<b>Tese7</b>	This copy printed out at: 26/02/2023 19:30:49

Variable	Worksheet	Access	Command	I/O Addr.	Type	Init.Value	Comment	Count	Page
readKV	Communic...	Read			BOOL			1	15
readN	Communic...	Read			BOOL			1	15
readNCE	Communic...	Read			BOOL			1	15
readRS	Communic...	Read			BOOL			1	11
readT	Communic...	Read			BOOL			1	11
readVOC	Communic...	Read			BOOL			1	16
RS232_INI...	Communic...	Call			RS232...			1	17
RS232_RE...	Communic...	Call			RS232...			1	18
RS232_SE...	Communic...	Call			RS232...			1	17
TOF_1	Communic...	Call			TOF			1	13
TOF_10	Communic...	Call			TOF			1	16
TOF_11	Communic...	Call			TOF			1	10
TOF_12	Communic...	Call			TOF			1	13
TOF_2	Communic...	Call			TOF			1	17
TOF_3	Communic...	Call			TOF			1	17
TOF_4	Communic...	Call			TOF			1	17
TOF_5	Communic...	Call			TOF			1	17
TOF_6	Communic...	Call			TOF			1	17
TOF_7	Communic...	Call			TOF			1	17
TOF_8	Communic...	Call			TOF			1	17
TOF_9	Communic...	Call			TOF			1	19
TON_1	Communic...	Call			TON			1	19
var1	Communic...	Write			INT			1	13
writeALL	Communic...	Write			BOOL			1	15
writeALL	Communic...	Read			BOOL			1	19
writeALL	Communic...	Read			BOOL			1	15
writeG	Communic...	Read			BOOL			1	16
writeG	Communic...	Write			BOOL			1	15
writeG	Communic...	Read			BOOL			1	10
writeISC	Communic...	Read			BOOL			1	13
writeISC	Communic...	Write			BOOL			1	11
writeISC	Communic...	Read			BOOL			1	16
writeKI	Communic...	Read			BOOL			1	14
writeKI	Communic...	Write			BOOL			1	11
writeKI	Communic...	Read			BOOL			1	13
writeKV	Communic...	Read			BOOL			1	13
writeKV	Communic...	Read			BOOL			1	14
writeKV	Communic...	Write			BOOL			1	11
writeN	Communic...	Read			BOOL			1	14
writeN	Communic...	Write			BOOL			1	13
writeN	Communic...	Read			BOOL			1	13
writeNCE	Communic...	Read			BOOL			1	13
writeNCE	Communic...	Read			BOOL			1	14
writeNCE	Communic...	Write			BOOL			1	13
writeRS	Communic...	Read			BOOL			1	13
writeRS	Communic...	Read			BOOL			1	14
writeRS	Communic...	Write			BOOL			1	11
writeT	Communic...	Read			BOOL			1	12
writeT	Communic...	Write			BOOL			1	15
writeT	Communic...	Read			BOOL			1	10
writeVOC	Communic...	Write			BOOL			1	11
writeVOC	Communic...	Read			BOOL			1	16
writeVOC	Communic...	Read			BOOL			1	10

<b>PC WORX</b>	Phoenix Contact GmbH & Co. KG Flachsmarktstraße 8 32825 Blomberg Germany	Sheet number 23
Current POU: Communication	<b>Tese7</b>	This copy printed out at: 26/02/2023 19:30:49

POU: AnalogOutput, Description: AnalogOutputT

POU: AnalogOutput, Variables: AnalogOutputV

Name	Type	Usage	Description	Init. value	DDP	TB	Hidden	as	ult	Hidden	ant	ptOnF	Device	Signal	Alias
Default															
IL_AO_2_UI_...	IL_AO...	VAR													
IL_AO_2_UI_...	IL_AO...	VAR													
IL_AO_2_UI_...	IL_AO...	VAR													
a_in	ANL_ARR_W_1_4	VAR													
a_out	ANL_ARR_W_1_4	VAR													
ONBOARD_O...	BOOL	VAR	Local o...												
Vref	REAL	VAR													
Vcarga	REAL	VAR													
arrPara	ANL_ARR_Para_1_2	VAR													
arrDiag	ANL_ARR_Diag	VAR													
IL_AO_2_UI_...	IL_AO...	VAR													
IL_AO_2_UI_...	IL_AO...	VAR													
testeCarga	BOOL	VAR			x										
TP_1	TP	VAR													
TP_2	TP	VAR													
TP_3	TP	VAR													
TP_4	TP	VAR													
TP_5	TP	VAR													
TP_6	TP	VAR													
TP_7	TP	VAR													
TP_8	TP	VAR													
TP_9	TP	VAR													
TP_10	TP	VAR													
TON_1	TON	VAR													
TON_2	TON	VAR													
TON_3	TON	VAR													
TON_4	TON	VAR													
TON_5	TON	VAR													
TON_6	TON	VAR													
TON_7	TON	VAR													
TON_8	TON	VAR													
TON_9	TON	VAR													
TON_10	TON	VAR													
TON_11	TON	VAR													
TP_11	TP	VAR													
TON_12	TON	VAR													
TP_12	TP	VAR													
TON_13	TON	VAR													
TP_13	TP	VAR													
TON_14	TON	VAR													
TP_14	TP	VAR													

Name	Type	Description	Init. value
a_in	ANL_ARR_W_1_4		
-	WORD		
-	WORD		
-	WORD		
-	WORD		

Name	Type	Description	Init. value
a_out	ANL_ARR_W_1_4		
-	WORD		
-	WORD		
-	WORD		
-	WORD		

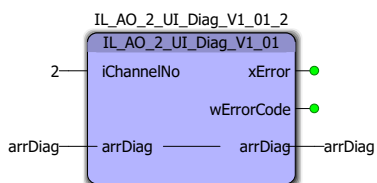
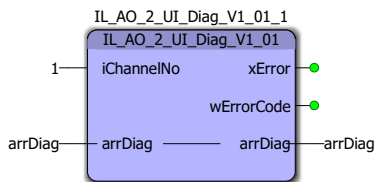
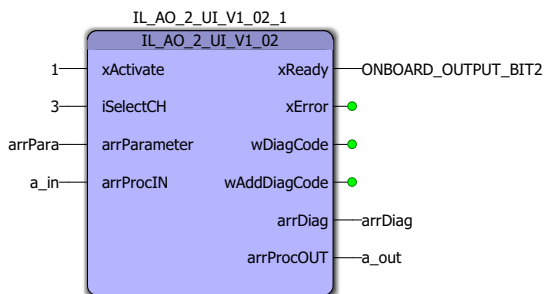
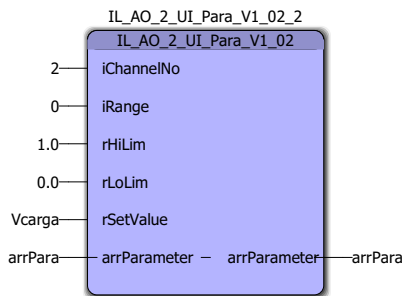
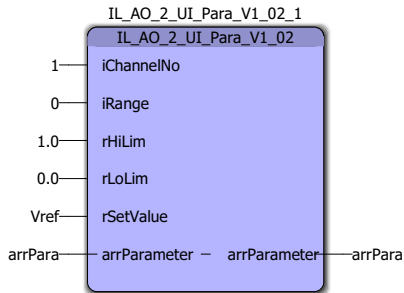
Name	Type	Description	Init. value
arrPara	ANL_ARR_Para_1_2		
-	udtPara		
--iRange	INT		
--rHiLim	REAL		
--rLoLim	REAL		
--rSetValue	REAL		
--xError	BOOL		
--wDummy	WORD		
-	udtPara		
--iRange	INT		
--rHiLim	REAL		
--rLoLim	REAL		
--rSetValue	REAL		

<b>PC WORX</b>	Phoenix Contact GmbH & Co. KG Flachsmarktstraße 8 32825 Blomberg Germany	Sheet number 24
Current POU: AnalogOutput	<b>Tese7</b>	This copy printed out at: 26/02/2023 19:30:49

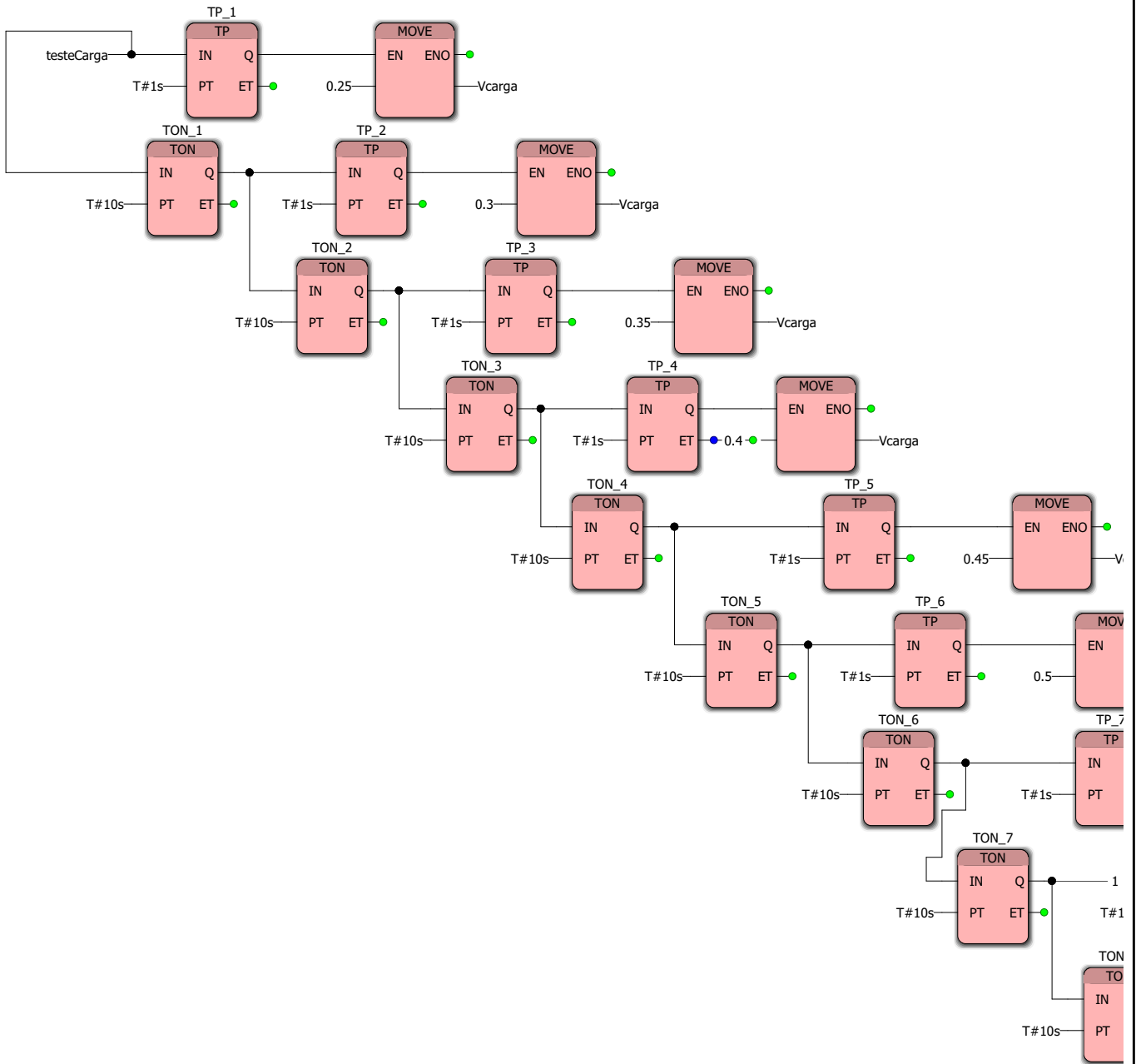
Name	Type	Description	Init. value
--xError	BOOL		
--wDummy	WORD		

Name	Type	Description	Init. value
arrDiag	ANL_ARR_Diag_1_2		
-	udtDiag		
--xError	BOOL		
--wErrorCode	WORD		
--iDelay	INT	Delay of arrInputPD[3/4] to arrOutputP...	
-	udtDiag		
--xError	BOOL		
--wErrorCode	WORD		
--iDelay	INT	Delay of arrInputPD[3/4] to arrOutputP...	

### POU: AnalogOutput, Worksheet: AnalogOutput

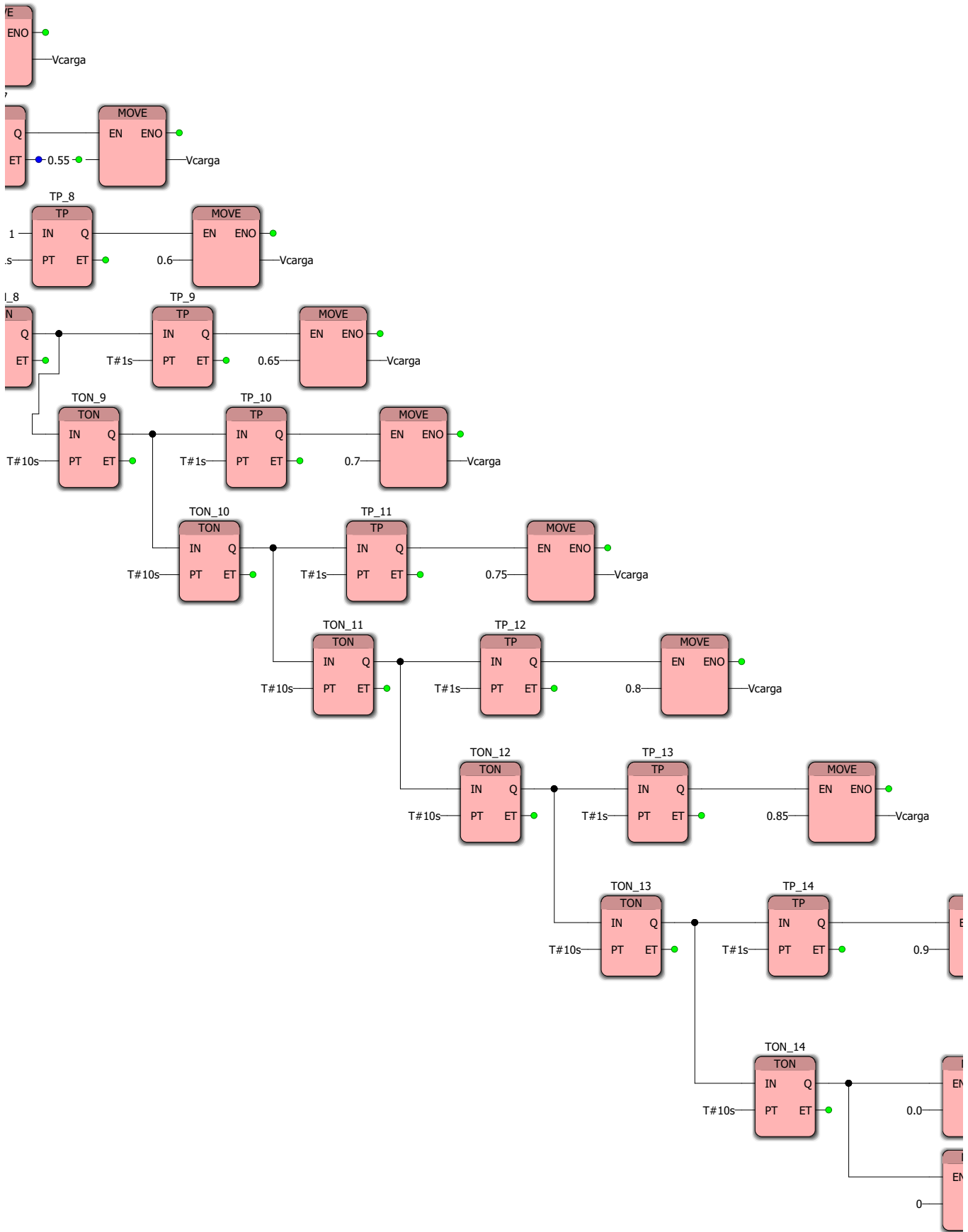


<b>PC WORX</b>	Phoenix Contact GmbH & Co. KG Flachsmarktstraße 8 32825 Blomberg Germany	Sheet number 25
Current POU: AnalogOutput	<b>Tese7</b>	This copy printed out at: 26/02/2023 19:30:50



<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG          Flachsmarktstraße 8          32825 Blomberg          Germany</p>	<p>Sheet number          26</p>
<p>Current POU:          AnalogOutput</p>	<p><b>Tese7</b></p>	<p>This copy printed out at:          26/02/2023 19:30:50</p>

carga



PC WORX

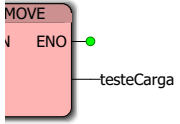
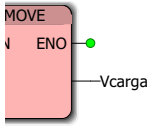
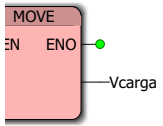
Phoenix Contact GmbH & Co. KG  
Flachsmarktstraße 8  
32825 Blomberg  
Germany

Sheet number  
27

Current POU:  
AnalogOutput

Tese7

This copy printed out at:  
26/02/2023 19:30:50



**PC WORX**

Phoenix Contact GmbH & Co. KG  
Flachsmarktstraße 8  
32825 Blomberg  
Germany

Sheet number  
28

Current POU:  
AnalogOutput

**Tese7**

This copy printed out at:  
26/02/2023 19:30:50

POU: AnalogOutput, Local Cross References

Variable	Worksheet	Access	Command	I/O Addr.	Type	Init.Value	Comment	Count	Page
arrDiag	AnalogOutput	Read			ANL_A...			2	27
arrDiag	AnalogOutput	Write			ANL_A...			3	27
arrPara	AnalogOutput	Write			ANL_A...			2	26
arrPara	AnalogOutput	Read			ANL_A...			1	27
arrPara	AnalogOutput	Read			ANL_A...			2	26
IL_AO_2_UI_Dia...	AnalogOutput	Call			IL_AO_...			1	26
IL_AO_2_UI_Dia...	AnalogOutput	Call			IL_AO_...			1	27
IL_AO_2_UI_Par...	AnalogOutput	Call			IL_AO_...			1	26
IL_AO_2_UI_Par...	AnalogOutput	Call			IL_AO_...			1	27
IL_AO_2_UI_V1...	AnalogOutput	Call			IL_AO_...			1	26
testeCarga	AnalogOutput	Write			BOOL			1	26
testeCarga	AnalogOutput	Read			BOOL			1	27
TON_1	AnalogOutput	Call			TON			1	27
TON_10	AnalogOutput	Call			TON			1	27
TON_11	AnalogOutput	Call			TON			1	27
TON_12	AnalogOutput	Call			TON			1	26
TON_13	AnalogOutput	Call			TON			1	27
TON_14	AnalogOutput	Call			TON			1	26
TON_2	AnalogOutput	Call			TON			1	25
TON_3	AnalogOutput	Call			TON			1	25
TON_4	AnalogOutput	Call			TON			1	27
TON_5	AnalogOutput	Call			TON			1	26
TON_6	AnalogOutput	Call			TON			1	26
TON_7	AnalogOutput	Call			TON			1	27
TON_8	AnalogOutput	Call			TON			1	27
TON_9	AnalogOutput	Call			TON			1	27
TP_1	AnalogOutput	Call			TP			1	26
TP_10	AnalogOutput	Call			TP			1	25
TP_11	AnalogOutput	Call			TP			1	27
TP_12	AnalogOutput	Call			TP			1	27
TP_13	AnalogOutput	Call			TP			1	26
TP_14	AnalogOutput	Call			TP			1	27
TP_2	AnalogOutput	Call			TP			1	26
TP_3	AnalogOutput	Call			TP			1	26
TP_4	AnalogOutput	Call			TP			1	28
TP_5	AnalogOutput	Call			TP			1	27
TP_6	AnalogOutput	Call			TP			1	27
TP_7	AnalogOutput	Call			TP			1	25
TP_8	AnalogOutput	Call			TP			1	25
TP_9	AnalogOutput	Call			TP			1	25

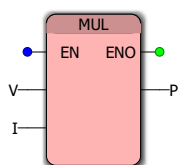
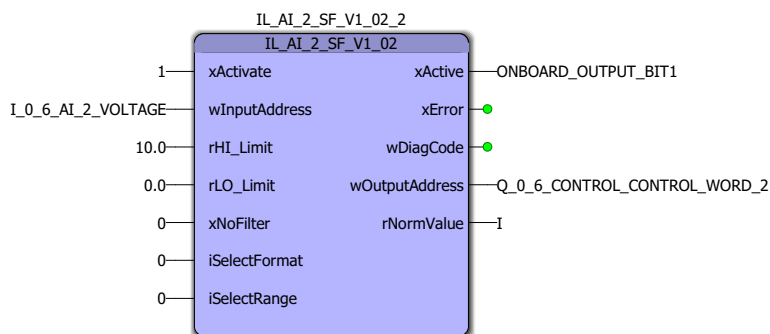
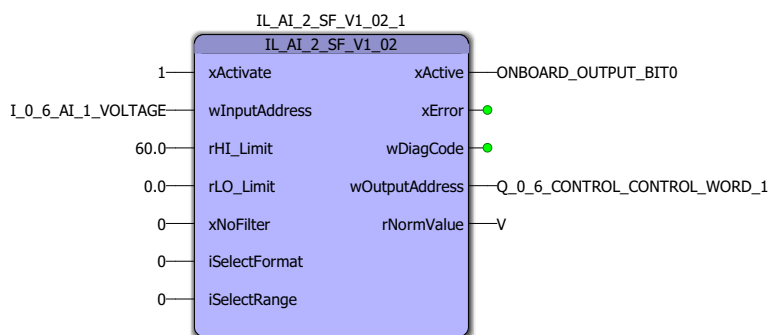
<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG                  Flachsmarktstraße 8                  32825 Blomberg                  Germany</p>	<p>Sheet number                  29</p>
<p>Current POU:                  AnalogOutput</p>	<p><b>Tese7</b></p>	<p>This copy printed out at:                  26/02/2023 19:30:50</p>

POU: AnalogInput, Description: AnalogInputT

POU: AnalogInput, Variables: AnalogInputV

Name	Type	Usage	Description	Init	eta	DDP	TB	Hidden	as	ult Hidden	ant	ptOnF	Device	Signal	Alias
Default															
IL_AI_2_SF_...	IL_AI_...	VAR													
IL_AI_2_SF_...	IL_AI_...	VAR													
ONBOARD_O...	BOOL	VAR_E...	Local o...												
ONBOARD_O...	BOOL	VAR_E...	Local o...												
I_0_6_AI_1_V...	WORD	VAR_E...													
Q_0_6_CONT...	WORD	VAR_E...													
Q_0_6_CONT...	WORD	VAR_E...													
I_0_6_AI_2_V...	WORD	VAR_E...													
V	REAL	VAR_E...													
I	REAL	VAR_E...													
P	REAL	VAR_E...													

POU: AnalogInput, Worksheet: AnalogInput



POU: AnalogInput, Local Cross References

Variable	Worksheet	Access	Command	I/O Addr.	Type	Init.Value	Comment	Count	Page
IL_AI_2_SF_V1_...	AnalogInput	Call			IL_AI_2...			1	30
IL_AI_2_SF_V1_...	AnalogInput	Call			IL_AI_2...			1	30

**PC WORX**

Phoenix Contact GmbH & Co. KG  
 Flachsmarktstraße 8  
 32825 Blomberg  
 Germany

Sheet number  
 31

Current POU:  
 AnalogInput

**Tese7**

This copy printed out at:  
 26/02/2023 19:30:50

POU: Main, Description: MainT

POU: Main, Variables: MainV

Name	Type	Usage	Description	Init	eta	DDP	TB	Hidden	as	Ult Hidden	Plant	OnF	Device	Signal	Alias
Default															
deltaV	REAL	VAR													
deltaP	REAL	VAR													
Vref	REAL	VAR E...													
I	REAL	VAR E...													
V	REAL	VAR E...													
P	REAL	VAR E...													
C	REAL	VAR													
Pa	REAL	VAR													
Va	REAL	VAR													
modo	INT	VAR		0		x									
Vbat	REAL	VAR		5..		x									

POU: Main, Worksheet: Main

```

C := 0.001; (*Dimensão do salto para modificação da ref.a da tensão (V)*)

IF modo = 1 THEN

deltaP := P - Pa;
deltaV := V - Va;

(*% Inicio do Algoritmo P&O %*)

IF deltaP > 0.0 AND deltaV > 0.0 THEN
Vref := Vref + C;
END_IF;

IF deltaP > 0.0 AND deltaV < 0.0 THEN
Vref := Vref - C;
END_IF;

IF deltaP < 0.0 AND deltaV > 0.0 THEN
Vref := Vref - C;
END_IF;

IF deltaP < 0.0 AND deltaV < 0.0 THEN
Vref := Vref + C;
END_IF;

IF deltaP = 0.0 THEN
Vref := Vref;
END_IF;

IF Vref < 0.0 THEN
Vref := 0.0;
END_IF;

IF Vref > 1.0 THEN
Vref := 1.0;
END_IF;

Va := V;
Pa := P;

END_IF;

IF modo = 2 THEN

IF V < Vbat THEN
Vref := Vref + C;
END_IF;

IF V > Vbat THEN
Vref := Vref - C;

```

<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG          Flachsmarktstraße 8          32825 Blomberg          Germany</p>	<p>Sheet number          32</p>
<p>Current POU:          Main</p>	<p>Tese7</p>	<p>This copy printed out at:          26/02/2023 19:30:51</p>

```
END_IF;

IF V = Vbat THEN
Vref := Vref;
END_IF;

IF Vref < 0.0 THEN
Vref := 0.0;
END_IF;

IF Vref > 1.0 THEN
Vref := 1.0;
END_IF;

END_IF;
```

<b>PC WORX</b>	Phoenix Contact GmbH & Co. KG Flachsmarktstraße 8 32825 Blomberg Germany	Sheet number 33
Current POU: Main	<b>Tese7</b>	This copy printed out at: 26/02/2023 19:30:51

POU: Main, Local Cross References

Variable	Worksheet	Access	Command	I/O Addr.	Type	Init.Value	Comment	Count	Page
C	Main	Write	:=		REAL			1	32
C	Main	Read			REAL			6	32
deltaP	Main	Read	<		REAL			2	32
deltaP	Main	Read	>		REAL			2	32
deltaP	Main	Read	=		REAL			1	32
deltaP	Main	Write	:=		REAL			1	32
deltaV	Main	Write	:=		REAL			1	32
deltaV	Main	Read	<		REAL			2	32
deltaV	Main	Read	>		REAL			2	32
modo	Main	Read	=		INT	0		2	32
Pa	Main	Write	:=		REAL			1	32
Pa	Main	Read			REAL			1	32
Va	Main	Write	:=		REAL			1	32
Va	Main	Read			REAL			1	32
Vbat	Main	Read			REAL	50.0		2	32
Vbat	Main	Read			REAL	50.0		1	33

Configuration: STD\_CNF, Resource: STD\_RES, Variables: MainV

Name	Type	Usage	Description	Init	DDP	TB	Hidden	as	Hidden	Want	OnF	Device	Signal	Alias
Default														

Configuration: STD\_CNF, Resource: STD\_RES, Variables: AnalogInputV

Name	Type	Usage	Description	Init	DDP	TB	Hidden	as	Hidden	Want	OnF	Device	Signal	Alias
Default														

Configuration: STD\_CNF, Resource: STD\_RES, Variables: AnalogOutputV

Name	Type	Usage	Description	Init	DDP	TB	Hidden	as	Hidden	Want	OnF	Device	Signal	Alias
Default														

Configuration: STD\_CNF, Resource: STD\_RES, Variables: CommunicationV

Name	Type	Usage	Description	Init	DDP	TB	Hidden	as	Hidden	Want	OnF	Device	Signal	Alias
Default														

Configuration: STD\_CNF, Resource: STD\_RES, Variables: Global\_Variables

Name	Type	Usage	Description	Init	DDP	TB	Hidden	as	Hidden	Want	OnF	Device	Signal	Alias
Auto														
P	REAL	VAR ...			x									
I	REAL	VAR ...			x									
V	REAL	VAR ...			x									
Vcarga	REAL	VAR ...		0..	x									
Vref	REAL	VAR ...		0..	x									
MsgReceive	STRING	VAR ...		'...'										
MsgReceiveB	Tmess...	VAR ...												
doneReceive...	BOOL	VAR ...												
aux	BOOL	VAR ...												
Default														
I 0 6 AI 1 V...	WORD	VAR ...										# 6 IB I...	AI 1 Vo...	
Q 0 6 CONT...	WORD	VAR ...										# 6 IB I...	Control...	
a in	ANL A...	VAR ...										# 5 IB I...	~AI 64	
a out	ANL A...	VAR ...										# 5 IB I...	~AO 64	
Q 0 6 CONT...	WORD	VAR ...										# 6 IB I...	Control...	
I 0 6 AI 2 V...	WORD	VAR ...										# 6 IB I...	AI 2 Vo...	
System Variables														
PLCMODE_ON	BOOL	VAR ...	PLC st...		x									
PLCMODE_L...	BOOL	VAR ...	PLC st...											
PLCMODE_R...	BOOL	VAR ...	PLC st...											
PLCMODE_S...	BOOL	VAR ...	PLC st...											
PLCMODE_H...	BOOL	VAR ...	PLC st...											
PLCDEBUG_...	BOOL	VAR ...	Breakp...											
PLCDEBUG_...	BOOL	VAR ...	Variabl...											
PLCDEBUG_...	BOOL	VAR ...	Addres...											
PLC TICKS_...	INT	VAR ...	Syste...											
PLC SYS TI...	DINT	VAR ...	Numbe...											
PLC TASK A...	INT	VAR ...	Numbe...											
PLC SYSTAS...	INT	VAR ...	Numbe...											
PLC MAX E...	DINT	VAR ...												
PLC ERRORS	DINT	VAR ...												
PLC TASK D...	INT	VAR ...	Numbe...											
PLC TASK 1	Task I...	VAR ...	Task 1 ...											
PLC TASK 2	Task I...	VAR ...	Task 2 ...											
PLC TASK 3	Task I...	VAR ...	Task 3 ...											
PLC TASK 4	Task I...	VAR ...	Task 4 ...											
PLC TASK 5	Task I...	VAR ...	Task 5 ...											
PLC TASK 6	Task I...	VAR ...	Task 6 ...											
PLC TASK 7	Task I...	VAR ...	Task 7 ...											
PLC TASK 8	Task I...	VAR ...	Task 8 ...											
LICENSE	DWORD	VAR ...	Licens...											
MASTER DIA...	BYTE	VAR ...	Master...											
MASTER DIA...	BYTE	VAR ...	Master...											
MASTER DIA...	BOOL	VAR ...	Bus se...											
MASTER DIA...	BOOL	VAR ...	Syste...											
MASTER DIA...	BOOL	VAR ...	Standa...											
MASTER DIA...	BOOL	VAR ...	Synchr...											
MASTER DIA...	BOOL	VAR ...	Faulty ...											
MASTER DIA...	BOOL	VAR ...	Define...											
MASTER DIA...	BOOL	VAR ...	Define...											
MASTER DIA...	BOOL	VAR ...	Pendin...											
MASTER DIA...	BOOL	VAR ...	User er...											
MASTER DIA...	BOOL	VAR ...	Periph...											

<p><b>PC WORX</b></p>	<p>Phoenix Contact GmbH &amp; Co. KG                  Flachmarktstraße 8                  32825 Blomberg                  Germany</p>	<p>Sheet number 35</p>
<p>Current POU: STD_CNF:STD_RES</p>	<p>Tese7</p>	<p>This copy printed out at: 26/02/2023 19:30:51</p>

Name	Type	Usage	Description	Init. value	DDP	TB	Hidden	as	ult	Hidden	Want	OnF	Device	Signal	Alias
MASTER DIA...	BOOL	VAR ...	Bus error												
MASTER DIA...	BOOL	VAR ...	Fault o...												
MASTER DIA...	BOOL	VAR ...	Diagno...												
MASTER DIA...	BOOL	VAR ...	Data tr...												
MASTER DIA...	BOOL	VAR ...	Select...												
MASTER DIA...	BOOL	VAR ...	Control...												
MASTER DIA...	BYTE	VAR ...	Master...												
MASTER DIA...	BYTE	VAR ...	Master...												
MASTER DIA...	BYTE	VAR ...	Extend...												
MASTER DIA...	BYTE	VAR ...	Extend...												
COP_DIAG S...	BOOL	VAR ...	Runtim...												
COP_DIAG S...	BOOL	VAR ...	Fatal e...												
COP_DIAG S...	BOOL	VAR ...	Warnin...												
COP_DIAG S...	BOOL	VAR ...	Power ...												
COP_DIAG S...	BOOL	VAR ...	Runtim...												
COP_DIAG S...	BOOL	VAR ...	Runtim...												
COP_DIAG S...	BOOL	VAR ...	Runtim...												
COP_DIAG S...	BOOL	VAR ...	Runtim...												
COP_DIAG S...	BOOL	VAR ...	Runtim...												
COP_DIAG S...	BOOL	VAR ...	Runtim...												
COP_DIAG P...	WORD	VAR ...	Diagno...												
COP_DIAG P...	WORD	VAR ...	Extend...												
COP_CPU L...	BOOL	VAR ...	The co...												
ONBOARD I...	WORD	VAR ...	Local i...												
ONBOARD I...	BOOL	VAR ...	Local i...												
ONBOARD I...	BOOL	VAR ...	Local i...												
ONBOARD I...	BOOL	VAR ...	Local i...												
ONBOARD I...	BOOL	VAR ...	Local i...												
ONBOARD I...	BOOL	VAR ...	Local i...												
ONBOARD I...	BOOL	VAR ...	Local i...												
ONBOARD I...	BOOL	VAR ...	Local i...												
ONBOARD O...	BOOL	VAR ...	Local o...												
ONBOARD O...	BOOL	VAR ...	Local o...												
ONBOARD O...	BOOL	VAR ...	Local o...												
ONBOARD O...	BOOL	VAR ...	Local o...												
RTC BATTE...	BOOL	VAR ...	Realti...												
RTC DATA I...	BOOL	VAR ...	Realti...												
FLASHCARD...	BOOL	VAR ...	Phoeni...												
ONBOARD O...	BOOL	VAR ...	Local I/...												
POWER SUP...	BOOL	VAR ...	24 V m...												
POWER SUP...	BOOL	VAR ...	24 V lo...												
POWER SUP...	BOOL	VAR ...	24 V lo...												
KEY SWITC...	BOOL	VAR ...	Start/st...												
KEY SWITC...	BOOL	VAR ...	Start/st...												
KEY SWITC...	BOOL	VAR ...	Start/st...												
RTC HOURS	INT	VAR ...	Syste...												
RTC MINUTES	INT	VAR ...	Syste...												
RTC SECON...	INT	VAR ...	Syste...												
RTC DAY	INT	VAR ...	Syste...												
RTC MONTH	INT	VAR ...	Syste...												
RTC YEAR	INT	VAR ...	Syste...												
PND S1 PLC...	BOOL	VAR ...	Status ...												
PND S1 VAL...	BOOL	VAR ...	IO Con...												
PND S1 OU...	BOOL	VAR ...	IOP st...												
PND S1 DAT...	WORD	VAR ...	Proces...												
PND S1 OU...	PND I...	VAR ...	Output ...												
PND S1 INP...	PND I...	VAR ...	Input p...												
PND IO DRI...	INT	VAR ...	Numbe...												
PND RESET...	WORD	VAR ...	shows ...												
IB DEVICE ...	BOOL	VAR ...	Interbu...												
IB DEVICE ...	BOOL	VAR ...	Interbu...												
IB DEVICE ...	BOOL	VAR ...	Interbu...												
APPMGR AP...	STRING	VAR ...			x										
APPMGR TE...	STRING	VAR ...													
APPMGR AP...	STRING	VAR ...			x										

Name	Type	Description	Init. value
MsgReceiveB	Tmessage		
-	BYTE		
-	BYTE		
-	BYTE		
-	BYTE		
-	BYTE		
-	BYTE		
-	BYTE		
-	BYTE		
-	BYTE		

<b>PC WORX</b>	Phoenix Contact GmbH & Co. KG Flachsmarktstraße 8 32825 Blomberg Germany	Sheet number 36
Current POU: STD_CNF:STD_RES	<b>Tese7</b>	This copy printed out at: 26/02/2023 19:30:51































Configuration: STD\_CNF, Resource: STD\_RES, I/O-Configuration

```
PROGRAM IN01 : INPUT
( (* @OWNER := Auto_MB *)
  VAR_ADR := 0,
  END_VAR_ADR := 0,
  DEVICE := DRIVER,
  DRIVER_NAME := 'eclriodrv_mbc',
  DRIVER_PAR1 := 0,
  DRIVER_PAR2 := 0,
  DRIVER_PAR3 := 0,
  DRIVER_PAR4 := 0,
  DATA_TYPE := WORD
);

PROGRAM OUT01 : OUTPUT
( (* @OWNER := Auto_MB *)
  VAR_ADR := 0,
  END_VAR_ADR := 0,
  DEVICE := DRIVER,
  DRIVER_NAME := 'eclriodrv_mbc',
  DRIVER_PAR1 := 0,
  DRIVER_PAR2 := 0,
  DRIVER_PAR3 := 0,
  DRIVER_PAR4 := 0,
  DATA_TYPE := WORD
);

PROGRAM IN00 : INPUT
( (* @OWNER := Auto_INTERBUS *)
  VAR_ADR := 1,
  END_VAR_ADR := 16,
  DEVICE := DRIVER,
  DRIVER_NAME := 'eclriodrv_ib',
  DRIVER_PAR1 := 0,
  DRIVER_PAR2 := 0,
  DRIVER_PAR3 := 0,
  DRIVER_PAR4 := 0,
  DATA_TYPE := BYTE
);

PROGRAM OUT00 : OUTPUT
( (* @OWNER := Auto_INTERBUS *)
  VAR_ADR := 1,
  END_VAR_ADR := 16,
  DEVICE := DRIVER,
  DRIVER_NAME := 'eclriodrv_ib',
  DRIVER_PAR1 := 0,
  DRIVER_PAR2 := 0,
  DRIVER_PAR3 := 0,
  DRIVER_PAR4 := 0,
  DATA_TYPE := BYTE
);

(* PROFINET Device input-output range *)
PROGRAM IN_PNP : INPUT
( (* @OWNER := Auto_Slave *)
  VAR_ADR := 63972,
  END_VAR_ADR := 64245,
  DEVICE := DRIVER,
  DRIVER_NAME := 'eclrIoDrv_PND30',
  DRIVER_PAR1 := 1,
  DRIVER_PAR2 := 1,
  DRIVER_PAR3 := 0,
  DRIVER_PAR4 := 0,
  DATA_TYPE := BYTE
);

(* PROFINET Device input-output range *)
PROGRAM OUT_PNP : OUTPUT
( (* @OWNER := Auto_Slave *)
  VAR_ADR := 63972,
```

<b>PC WORX</b>	Phoenix Contact GmbH & Co. KG Flachsmarktstraße 8 32825 Blomberg Germany	Sheet number 52
Current POU: STD_CNF:STD_RES	<b>Tese7</b>	This copy printed out at: 26/02/2023 19:30:54

```
END_VAR_ADR := 64245,  
DEVICE := DRIVER,  
DRIVER_NAME := 'eclrIoDrv_PND30',  
DRIVER_PAR1 := 1,  
DRIVER_PAR2 := 1,  
DRIVER_PAR3 := 0,  
DRIVER_PAR4 := 0,  
DATA_TYPE := BYTE  
);
```

<b>PC WORX</b>	Phoenix Contact GmbH & Co. KG Flachsmarktstraße 8 32825 Blomberg Germany	Sheet number 53
Current POU: STD_CNF:STD_RES	<b>Tese7</b>	This copy printed out at: 26/02/2023 19:30:54

Configuration: STD\_CNF, Resource: STD\_RES, Global Cross References

Variable	Worksheet	Access	Command	I/O Addr.	Global Path	Type	Init.Value	Comment	Count	Page
a_in	AnalogOut...	Read		%IB5	STD_C...	ANL_A...			1	27
a_out	AnalogOut...	Write		%QB5	STD_C...	ANL_A...			1	27
aux	STD_CNF...				STD_C...	BOOL			1	
COP_CPU...	STD_CNF...			%MX1.60	STD_C...	BOOL		The controller is ap	1	
COP_DIA...	STD_CNF...			%MW1.60	STD_C...	WORD		Extended diagnosti	1	
COP_DIA...	STD_CNF...			%MW1.60	STD_C...	WORD		Diagnostic paramet	1	
COP_DIA...	STD_CNF...			%MX1.60	STD_C...	BOOL		Fatal error (control	1	
COP_DIA...	STD_CNF...			%MX1.60	STD_C...	BOOL		Runtime system DE	1	
COP_DIA...	STD_CNF...			%MX1.60	STD_C...	BOOL		Runtime system HA	1	
COP_DIA...	STD_CNF...			%MX1.60	STD_C...	BOOL		Runtime system LC	1	
COP_DIA...	STD_CNF...			%MX1.60	STD_C...	BOOL		Runtime system RE	1	
COP_DIA...	STD_CNF...			%MX1.60	STD_C...	BOOL		Runtime system RL	1	
COP_DIA...	STD_CNF...			%MX1.60	STD_C...	BOOL		Runtime system ST	1	
COP_DIA...	STD_CNF...			%MX1.60	STD_C...	BOOL		Power on (control p	1	
COP_DIA...	STD_CNF...			%MX1.60	STD_C...	BOOL		Runtime error (cont	1	
COP_DIA...	STD_CNF...			%MX1.60	STD_C...	BOOL		Warning (control pr	1	
doneRecei...	STD_CNF...				STD_C...	BOOL			1	
FLASHCA...	STD_CNF...			%MX1.60	STD_C...	BOOL		Phoenix Contact SE	1	
I	AnalogInpu...	Read			STD_C...	REAL			1	30
I	AnalogInpu...	Write			STD_C...	REAL			1	30
I	Main.MainV				STD_C...	REAL			1	
I_0_6_AI_1...	AnalogInpu...	Read		%IW13	STD_C...	WORD			1	30
I_0_6_AI_2...	AnalogInpu...	Read		%IW15	STD_C...	WORD			1	30
IB_DEVIC...	STD_CNF...			%MX1.60	STD_C...	BOOL		Interbus device con	1	
IB_DEVIC...	STD_CNF...			%MX1.60	STD_C...	BOOL		Interbus device con	1	
IB_DEVIC...	STD_CNF...			%MX1.60	STD_C...	BOOL		Interbus device con	1	
KEY_SWIT...	STD_CNF...			%MX1.60	STD_C...	BOOL		Start/stop switch in	1	
KEY_SWIT...	STD_CNF...			%MX1.60	STD_C...	BOOL		Start/stop switch in	1	
KEY_SWIT...	STD_CNF...			%MX1.60	STD_C...	BOOL		Start/stop switch in	1	
LICENSE	STD_CNF...			%MD1.60	STD_C...	DWORD		License information	1	
MASTER_...	STD_CNF...			%MB1.60	STD_C...	BYTE		Extended master di	1	
MASTER_...	STD_CNF...			%MB1.60	STD_C...	BYTE		Extended master di	1	
MASTER_...	STD_CNF...			%MB1.60	STD_C...	BYTE		Master diagnostic p	1	
MASTER_...	STD_CNF...			%MB1.60	STD_C...	BYTE		Master diagnostic p	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		Selected configurati	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		Bus segment discor	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		Bus error	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		Fault on the controll	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		Faulty data cycles	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		Diagnostic routine a	1	
MASTER_...	STD_CNF...			%MB1.60	STD_C...	BYTE		Master diagnostic s	1	
MASTER_...	STD_CNF...			%MB1.60	STD_C...	BYTE		Master diagnostic s	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		Peripheral fault	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		Defined error densit	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		Controller board rea	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		Standard function p	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		Data transmission a	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		Pending message	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		Synchronization err	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		System failure	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		User error/paramete	1	
MASTER_...	STD_CNF...			%MX1.60	STD_C...	BOOL		Defined warning tim	1	
MsgReceive	Communic...	Read			STD_C...	STRING	'0000000'		4	14
MsgReceive	Communic...	Read			STD_C...	STRING	'0000000'		1	17
MsgReceive	Communic...	Read			STD_C...	STRING	'0000000'		1	19
MsgReceive	Communic...	Read			STD_C...	STRING	'0000000'		1	11
MsgReceive	Communic...	Read			STD_C...	STRING	'0000000'		1	10

<b>PC WORX</b>	Phoenix Contact GmbH & Co. KG Flachsmarktstraße 8 32825 Blomberg Germany	Sheet number 54
Current POU: STD_CNF:STD_RES	<b>Tese7</b>	This copy printed out at: 26/02/2023 19:30:54

Variable	Worksheet	Access	Command	I/O Addr.	Global Path	Type	Init. Value	Comment	Count	Page
MsgReceive	Communic...	Read			STD_C...	STRING	'0000000'		1	15
MsgReceive	Communic...	Read			STD_C...	STRING	'0000000'		1	18
MsgReceive	Communic...	Read			STD_C...	STRING	'0000000'		1	13
MsgReceive	Communic...	Read			STD_C...	STRING	'0000000'		2	12
MsgReceive	Communic...	Write			STD_C...	STRING	'0000000'		1	20
MsgReceive	Communic...	Read			STD_C...	STRING	'0000000'		1	20
MsgReceiv...	Communic...	Write			STD_C...	Tmess...			1	17
MsgReceiv...	Communic...	Read			STD_C...	Tmess...			1	17
MsgReceiv...	Communic...	Read			STD_C...	Tmess...			1	20
MsgReceiv...	Communic...	Write			STD_C...	Tmess...			1	20
ONBOARD...	STD_CNF...			%MW1.6C	STD_C...	WORD		Local inputs	1	
ONBOARD...	STD_CNF...			%MX1.60	STD_C...	BOOL		Local input IN1	1	
ONBOARD...	STD_CNF...			%MX1.60	STD_C...	BOOL		Local input IN2	1	
ONBOARD...	STD_CNF...			%MX1.60	STD_C...	BOOL		Local input IN3	1	
ONBOARD...	STD_CNF...			%MX1.60	STD_C...	BOOL		Local input IN4	1	
ONBOARD...	STD_CNF...			%MX1.60	STD_C...	BOOL		Local input IN5	1	
ONBOARD...	STD_CNF...			%MX1.60	STD_C...	BOOL		Local input IN6	1	
ONBOARD...	STD_CNF...			%MX1.60	STD_C...	BOOL		Local input IN7	1	
ONBOARD...	STD_CNF...			%MX1.60	STD_C...	BOOL		Local input IN8	1	
ONBOARD...	AnalogInpu...	Write		%MX3.61	STD_C...	BOOL		Local output OUT1	1	30
ONBOARD...	AnalogInpu...	Write		%MX3.61	STD_C...	BOOL		Local output OUT2	1	30
ONBOARD...	AnalogOut...	Write		%MX3.61	STD_C...	BOOL		Local output OUT3	1	27
ONBOARD...	STD_CNF...			%MX3.61	STD_C...	BOOL		Local output OUT4	1	
ONBOARD...	STD_CNF...			%MX1.60	STD_C...	BOOL		Local I/O overload	1	
P	AnalogInpu...	Write			STD_C...	REAL			1	30
P	Main.Main	Read	-		STD_C...	REAL			1	32
P	Main.Main	Read			STD_C...	REAL			1	32
PLC_ERR...	STD_CNF...			%MD1.20	STD_C...	DINT			1	
PLC_MAX...	STD_CNF...			%MD1.20	STD_C...	DINT			1	
PLC_SYS_...	STD_CNF...			%MD1.0	STD_C...	DINT		Number of system t	1	
PLC_SYST...	STD_CNF...			%MW1.2C	STD_C...	INT		Number of available	1	
PLC_TASK...	STD_CNF...			%MB1.50	STD_C...	Task_In...		Task 1 information	1	
PLC_TASK...	STD_CNF...			%MB1.51	STD_C...	Task_In...		Task 2 information	1	
PLC_TASK...	STD_CNF...			%MB1.52	STD_C...	Task_In...		Task 3 information	1	
PLC_TASK...	STD_CNF...			%MB1.53	STD_C...	Task_In...		Task 4 information	1	
PLC_TASK...	STD_CNF...			%MB1.55	STD_C...	Task_In...		Task 5 information	1	
PLC_TASK...	STD_CNF...			%MB1.56	STD_C...	Task_In...		Task 6 information	1	
PLC_TASK...	STD_CNF...			%MB1.57	STD_C...	Task_In...		Task 7 information	1	
PLC_TASK...	STD_CNF...			%MB1.58	STD_C...	Task_In...		Task 8 information	1	
PLC_TASK...	STD_CNF...			%MW1.2C	STD_C...	INT		Number of available	1	
PLC_TASK...	STD_CNF...			%MW1.4	STD_C...	INT		Number of tasks us	1	
PLC_TICK...	STD_CNF...			%MW1.2C	STD_C...	INT		System ticks per se	1	
PLCDEBU...	STD_CNF...			%MX1.20	STD_C...	BOOL		Breakpoint set	1	
PLCDEBU...	STD_CNF...			%MX1.20	STD_C...	BOOL		Variable(s) forced	1	
PLCDEBU...	STD_CNF...			%MX1.20	STD_C...	BOOL		Address status ena	1	
PLCMODE...	STD_CNF...			%MX1.8.C	STD_C...	BOOL		PLC status HALT	1	
PLCMODE...	STD_CNF...			%MX1.20	STD_C...	BOOL		PLC status LOADIN	1	
PLCMODE...	STD_CNF...			%MX1.20	STD_C...	BOOL		PLC status ON	1	
PLCMODE...	STD_CNF...			%MX1.7.C	STD_C...	BOOL		PLC status RUN	1	
PLCMODE...	STD_CNF...			%MX1.6.C	STD_C...	BOOL		PLC status STOP	1	
PND_IO_D...	STD_CNF...				STD_C...	INT		Number of the PLC	1	
PND_RES...	STD_CNF...			%MW3.61	STD_C...	WORD		shows the bits of th	1	
PND_S1_...	STD_CNF...				STD_C...	WORD		Process data length	1	
PND_S1_I...	STD_CNF...				STD_C...	PND_I...		Input process data	1	
PND_S1_...	STD_CNF...				STD_C...	BOOL		IOP status of the hig	1	
PND_S1_...	STD_CNF...				STD_C...	PND_I...		Output process dat	1	
PND_S1_P...	STD_CNF...				STD_C...	BOOL		Status of the higher	1	
PND_S1_V...	STD_CNF...				STD_C...	BOOL		IO Controller has es	1	
POWER_S...	STD_CNF...			%MX1.60	STD_C...	BOOL		24 V local input sup	1	

<b>PC WORX</b>	Phoenix Contact GmbH & Co. KG Flachsmarktstraße 8 32825 Blomberg Germany	Sheet number 55
Current POU: STD_CNF:STD_RES	<b>Tese7</b>	This copy printed out at: 26/02/2023 19:30:55

Variable	Worksheet	Access	Command	I/O Addr.	Global Path	Type	Init. Value	Comment	Count	Page
POWER_S...	STD_CNF...			%MX1.60	STD_C...	BOOL		24 V main supply v	1	
POWER_S...	STD_CNF...			%MX1.60	STD_C...	BOOL		24 V local output su	1	
Q_0_6_CO...	AnalogInpu...	Write		%QW13	STD_C...	WORD			1	30
Q_0_6_CO...	AnalogInpu...	Write		%QW15	STD_C...	WORD			1	30
RTC_BATT...	STD_CNF...			%MX1.60	STD_C...	BOOL		Realtime clock batt	1	
RTC_DATA...	STD_CNF...			%MX1.60	STD_C...	BOOL		Realtime clock data	1	
RTC_DAY	STD_CNF...			%MW1.6C	STD_C...	INT		System time (day)	1	
RTC_HOU...	STD_CNF...			%MW1.6C	STD_C...	INT		System time (hours	1	
RTC_MINU...	STD_CNF...			%MW1.6C	STD_C...	INT		System time (minut	1	
RTC_MON...	STD_CNF...			%MW1.6C	STD_C...	INT		System time (month	1	
RTC_SEC...	STD_CNF...			%MW1.6C	STD_C...	INT		System time (secor	1	
RTC_YEAR	STD_CNF...			%MW1.6C	STD_C...	INT		System time (year)	1	
V	AnalogInpu...	Write			STD_C...	REAL			1	30
V	AnalogInpu...	Read			STD_C...	REAL			1	30
V	Main.Main	Read	>		STD_C...	REAL			1	32
V	Main.Main	Read	<		STD_C...	REAL			1	32
V	Main.Main	Read			STD_C...	REAL			1	32
V	Main.Main	Read	=		STD_C...	REAL			1	33
V	Main.Main	Read	-		STD_C...	REAL			1	32
Vcarga	AnalogOut...	Write			STD_C...	REAL	0.0		3	25
Vcarga	AnalogOut...	Write			STD_C...	REAL	0.0		3	27
Vcarga	AnalogOut...	Read			STD_C...	REAL	0.0		1	26
Vcarga	AnalogOut...	Write			STD_C...	REAL	0.0		8	26
Vcarga	AnalogOut...	Write			STD_C...	REAL	0.0		1	28
Vref	AnalogOut...	Read			STD_C...	REAL	0.0		1	26
Vref	Main.Main	Write	:=		STD_C...	REAL	0.0		9	32
Vref	Main.Main	Read			STD_C...	REAL	0.0		1	33
Vref	Main.Main	Read	+		STD_C...	REAL	0.0		3	32
Vref	Main.Main	Read	<		STD_C...	REAL	0.0		1	33
Vref	Main.Main	Write	:=		STD_C...	REAL	0.0		3	33
Vref	Main.Main	Read	>		STD_C...	REAL	0.0		1	32
Vref	Main.Main	Read	<		STD_C...	REAL	0.0		1	32
Vref	Main.Main	Read			STD_C...	REAL	0.0		1	32
Vref	Main.Main	Read	-		STD_C...	REAL	0.0		3	32
Vref	Main.Main	Read	>		STD_C...	REAL	0.0		1	33

**PC WORX**

Phoenix Contact GmbH & Co. KG  
 Flachsmarktstraße 8  
 32825 Blomberg  
 Germany

Sheet number  
 56

Current POU:  
 STD\_CNF:STD\_RES

**Tese7**

This copy printed out at:  
 26/02/2023 19:30:55

<b>Data types</b> .....	1
Data types: rs232types .....	1
Data types: sys_flag_types .....	1
Data types: compile .....	4
<b>Logical POU's</b> .....	5
<b>POU: MUV</b> .....	5
POU: MUV, Description: MUVT .....	5
POU: MUV, Variables: MUVV .....	5
POU: MUV, Worksheet: MUV .....	5
POU: MUV, Local Cross References .....	6
<b>POU: Communication</b> .....	7
POU: Communication, Description: CommunicationT .....	7
POU: Communication, Variables: CommunicationV .....	7
POU: Communication, Worksheet: Communication .....	10
POU: Communication, Local Cross References .....	21
<b>POU: AnalogOutput</b> .....	24
POU: AnalogOutput, Description: AnalogOutputT .....	24
POU: AnalogOutput, Variables: AnalogOutputV .....	24
POU: AnalogOutput, Worksheet: AnalogOutput .....	25
POU: AnalogOutput, Local Cross References .....	29
<b>POU: AnalogInput</b> .....	30
POU: AnalogInput, Description: AnalogInputT .....	30
POU: AnalogInput, Variables: AnalogInputV .....	30
POU: AnalogInput, Worksheet: AnalogInput .....	30
POU: AnalogInput, Local Cross References .....	31
<b>POU: Main</b> .....	32
POU: Main, Description: MainT .....	32
POU: Main, Variables: MainV .....	32
POU: Main, Worksheet: Main .....	32
POU: Main, Local Cross References .....	34
<b>Physical Hardware</b> .....	35
<b>Configuration: STD_CNF</b> .....	35
Configuration: STD_CNF, Resource: STD_RES .....	35
Configuration: STD_CNF, Resource: STD_RES, Variables: MainV .....	35
Configuration: STD_CNF, Resource: STD_RES, Variables: AnalogInputV .....	35
Configuration: STD_CNF, Resource: STD_RES, Variables: AnalogOutputV .....	35
Configuration: STD_CNF, Resource: STD_RES, Variables: CommunicationV .....	35
Configuration: STD_CNF, Resource: STD_RES, Variables: Global_Variables .....	35
Configuration: STD_CNF, Resource: STD_RES, I/O-Configuration .....	52
Configuration: STD_CNF, Resource: STD_RES, Global Cross References .....	54

<b>PC WORX</b>	Phoenix Contact GmbH & Co. KG Flachsmarktstraße 8 32825 Blomberg Germany	Sheet number 1
Current POU: Table of Contents	<b>Tese7</b>	This copy printed out at: 26/02/2023 19:30:57

17

# Photovoltaic System Simulator

**51 Electrical Values PLC**  
(Charge Controller)

4 Voltage [V] 1 @GV.V

5 Current [A] 2 @GV.I

6 Power [W] 9 @GV.P

---

**67 Electrical Values PIC**  
(Photovoltaic Panel)

6 Voltage [V] 6 @GV.V

6 Current [A] 6 @GV.I

6 Power [W] 6 @GV.P

---

**52 Duty Cycle**

7 Boost Converter @GV.Vref

57-59-58-60

8 Load Converter @GV.Vcarga

53-55-54-56

---

Battery Voltage: 48 @GV.Vbat

---

68 Load Ramp

increases duty cycle of load converter by progressive steps every 10 sec.

20-22-44

21

33 @MACROXMinTim

35 @MACROXMinDat

34 @MACROXTTime[2].61

ROONTXDate[2].61

34 @MACROXTTime[4].61

ROONTXDate[4].61

36 @MACROXTTime[6].61

ROONTXDate[6].61

37 @MACROXTTime

@MACROXTDate

37 Manual

35 Save

40 << scroll

41 < scroll

42 scroll >

43 scroll >>

36 Clear

11-12-13-14-15-16

39 Zoom Out

38 Zoom In

19

51 Operation Mode

45 Manual Mode

46 MPPT Mode

47 Battery Mode

3 Edit Parameters

Manual Mode: No control method is performed by the PLC.

MPPT Mode: The boost converter duty cycle will be changed automatically in order to reach maximum power.

Battery Mode: The boost converter duty cycle will be changed automatically to provide the voltage from 'Battery Voltage' field.

TEQ Information for Main.teq

- 1 -PAINTER StaticText
  - r0 - EDIT SOURCE[[PPO "@GV.V" ]]
  - r1 - TEXT HEIGHT CENTERED
  - r2 - TEXT WIDTH CENTERED
- 2 -PAINTER StaticText
  - r0 - EDIT SOURCE[[PPO "@GV.I" ]]
  - r1 - TEXT HEIGHT CENTERED
  - r2 - TEXT WIDTH CENTERED
- 3 -PAINTER Button
  - r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_7f\_36.png" ]]
  - r1 - TEXT HEIGHT CENTERED
  - r2 - TEXT WIDTH CENTERED
  - r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_7f\_36.png" ]]
  - r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_7f\_36.png" ]]
  - r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Edit Parameters" ]]
  - a0 - SOURCE TO DEST[[STRING "Parameters.teq" ]]TO [[STRING "Parameters.teq" ]] ON MOUSE\_UP\_EVENT
- 4 -PAINTER StaticText
  - r0 - EDIT SOURCE[[HTML TAG "Voltage [V]" ]]
  - r1 - TEXT HEIGHT CENTERED
  - r2 - TEXT WIDTH CENTERED
- 5 -PAINTER StaticText
  - r0 - EDIT SOURCE[[HTML TAG "Current [A]" ]]
  - r1 - TEXT HEIGHT CENTERED
  - r2 - TEXT WIDTH CENTERED
- 6 -PAINTER StaticText
  - r0 - EDIT SOURCE[[HTML TAG "Power [W]" ]]
  - r1 - TEXT HEIGHT CENTERED

	Date	Name	File : C:\Users\helio\Desktop\WEBVIEW2\Main.teq
Creation	Monday, June 06, 2022		
Last modified	Thursday, June 09, 2022		Project : C:\Users\helio\Desktop\WEBVIEW2\Tese.prj
			Page : 1/9

r2 - TEXT WIDTH CENTERED

7 -PAINTER StaticText  
r0 - EDIT SOURCE[[HTML TAG "Boost Converter" ]]  
r1 - TEXT HEIGHT CENTERED  
r2 - TEXT WIDTH CENTERED

8 -PAINTER StaticText  
r0 - EDIT SOURCE[[HTML TAG "Load Converter" ]]  
r1 - TEXT HEIGHT CENTERED  
r2 - TEXT WIDTH CENTERED

9 -PAINTER StaticText  
r0 - EDIT SOURCE[[PPO "@GV.P" ]]  
r1 - TEXT HEIGHT CENTERED  
r2 - TEXT WIDTH CENTERED

10 -PAINTER StaticText  
r0 - TEXT HEIGHT CENTERED  
r1 - TEXT WIDTH CENTERED  
r2 - EDIT SOURCE[[CONTAINER "activemode" ]]

11 -PAINTER Event  
a0 - END PLCACTIONS LIST ON NO\_EVENT  
a1 - SOURCE TO DEST[[STRING "Manual" ]]  
TO [[STRING "Manual" ]] ON REPAINT\_EVENT  
ON CONDITION [[[ PPO "Main.mod0" ] == [STRING "0" ] ] ]

12 -PAINTER MacroNameHiddenLabel  
r0 - HIDE PAINTER  
r1 - EDIT SOURCE[[STRING "EventP\_writeSrc2Dst\_isEqual\_6\_30\_02" ]]  
r2 - TEXT HEIGHT CENTERED

13 -PAINTER Event  
a0 - END PLCACTIONS LIST ON NO\_EVENT  
a1 - SOURCE TO DEST[[STRING "MPPT" ]]  
TO [[STRING "MPPT" ]] ON REPAINT\_EVENT  
ON CONDITION [[[ PPO "Main.mod0" ] == [STRING "1" ] ] ]

14 -PAINTER MacroNameHiddenLabel  
r0 - HIDE PAINTER  
r1 - EDIT SOURCE[[STRING "EventP\_writeSrc2Dst\_isEqual\_6\_30\_02" ]]  
r2 - TEXT HEIGHT CENTERED

15 -PAINTER Event  
a0 - END PLCACTIONS LIST ON NO\_EVENT  
a1 - SOURCE TO DEST[[STRING "Battery" ]]  
TO [[STRING "Battery" ]] ON REPAINT\_EVENT  
ON CONDITION [[[ PPO "Main.mod0" ] == [STRING "2" ] ] ]

16 -PAINTER MacroNameHiddenLabel  
r0 - HIDE PAINTER  
r1 - EDIT SOURCE[[STRING "EventP\_writeSrc2Dst\_isEqual\_6\_30\_02" ]]  
r2 - TEXT HEIGHT CENTERED

17 -PAINTER StaticText  
r0 - EDIT SOURCE[[HTML TAG "Photovoltaic System Simulator" ]]  
r1 - TEXT HEIGHT CENTERED  
r2 - TEXT WIDTH CENTERED

18 -PAINTER Image  
r0 - PAINT SOURCE[[FILE "logo-isel.png" ]]  
r1 - RESIZE IMAGE

19 -PAINTER StaticText  
r0 - TEXT HEIGHT CENTERED  
r1 - TEXT WIDTH CENTERED  
r2 - HIDE PAINTER  
r3 - END PLCREPAINTS LIST  
r4 - USE EXTERN REPAINTS LIST[[FILE "macroExtRepaints.xml" ]]  
r5 - EDIT SOURCE[[CONTAINER "@MACROShowGrid.61f434aa" ]]  
r6 - EDIT SOURCE[[CONTAINER "@MACROAutoYAxisDescr.61f434aa" ]]  
r7 - EDIT SOURCE[[CONTAINER "@MACRODisplayedTime.61f434aa" ]]  
r8 - EDIT SOURCE[[CONTAINER "@MACROMaxStoreTime.61f434aa" ]]  
r9 - EDIT SOURCE[[CONTAINER "@MACROAvailableTrends.61f434aa" ]]  
r10 - EDIT SOURCE[[CONTAINER "@MACROAvailableTrendsColored.61f434aa" ]]  
r11 - EDIT SOURCE[[PPO "Communication.PIC\_V" ]]

	Date	Name	File : C:\Users\helio\Desktop\WEBVIEW2\Main.teq
Creation	Monday, June 06, 2022		
Last modified	Thursday, June 09, 2022		Project : C:\Users\helio\Desktop\WEBVIEW2\Tese.prj
			Page : 2/9

```

r12 - EDIT SOURCE[[STRING "0" ]]
r13 - EDIT SOURCE[[STRING "100" ]]
r14 - EDIT SOURCE[[STRING "0,255,0" ]]
r15 - EDIT SOURCE[[PPO "Communication.PIC_I" ]]
r16 - EDIT SOURCE[[STRING "0" ]]
r17 - EDIT SOURCE[[STRING "5" ]]
r18 - EDIT SOURCE[[STRING "255,0,0" ]]
r19 - EDIT SOURCE[[PPO "Communication.PIC_P" ]]
r20 - EDIT SOURCE[[STRING "0" ]]
r21 - EDIT SOURCE[[STRING "200" ]]
r22 - EDIT SOURCE[[STRING "0,0,255" ]]
r23 - EDIT SOURCE[[PPO "@GV.V" ]]
r24 - EDIT SOURCE[[STRING "0" ]]
r25 - EDIT SOURCE[[STRING "100" ]]
r26 - EDIT SOURCE[[STRING "128,64,0" ]]
r27 - EDIT SOURCE[[PPO "@GV.I" ]]
r28 - EDIT SOURCE[[STRING "0" ]]
r29 - EDIT SOURCE[[STRING "5" ]]
r30 - EDIT SOURCE[[STRING "128,0,128" ]]
r31 - EDIT SOURCE[[PPO "@GV.P" ]]
r32 - EDIT SOURCE[[STRING "0" ]]
r33 - EDIT SOURCE[[STRING "200" ]]
r34 - EDIT SOURCE[[STRING "0,0,0" ]]
r35 - EDIT SOURCE[[STRING "" ]]
r36 - EDIT SOURCE[[STRING "0" ]]
r37 - EDIT SOURCE[[STRING "100" ]]
r38 - EDIT SOURCE[[STRING "255,0,0" ]]
r39 - EDIT SOURCE[[STRING "" ]]
r40 - EDIT SOURCE[[STRING "0" ]]
r41 - EDIT SOURCE[[STRING "100" ]]
r42 - EDIT SOURCE[[STRING "255,128,0" ]]
r43 - EDIT SOURCE[[STRING "" ]]
r44 - EDIT SOURCE[[STRING "0" ]]
r45 - EDIT SOURCE[[STRING "100" ]]
r46 - EDIT SOURCE[[STRING "0,255,0" ]]
r47 - EDIT SOURCE[[STRING "" ]]
r48 - EDIT SOURCE[[STRING "0" ]]
r49 - EDIT SOURCE[[STRING "100" ]]
r50 - EDIT SOURCE[[STRING "128,0,0" ]]
r51 - EDIT SOURCE[[CONTAINER "@MACROAutoYAxisDescrAutoMinMax.61f434aa" ]]
r52 - EDIT SOURCE[[CONTAINER "@MACROVersion.61f434aa" ]]
r53 - EDIT SOURCE[[STRING "NDIGIT3" ]]
r54 - END EXTERN REPAINTS LIST
a0 - DISABLE ON CONDITION ON NO_EVENT
20 -PAINTER Rectangle
21 -PAINTER Event
a0 - END PLCACTIONS LIST ON NO_EVENT
a1 - DISABLE ON CONDITION[[STRING "" ]]TO [[STRING "" ]] ON NO_EVENT
ON CONDITION [( ( [CONTAINER "@MACROActiveTrends.61f434aa" ] != [STRING "0" ] ) ) AND ( ( [CONTAINER "@MACROActiveTrends.61f434aa" ] != [STRING "0" ] ) ) ]
a2 - SOURCE TO DEST[[STRING "#sp;120,0,30,50,70\n" ]]TO [[STRING "#sp;120,0,30,50,70\n" ]] ON GAINED_FOCUS_EVENT
a3 - SOURCE TO DEST[[CONTAINER "@MACROActiveTrendsHeaderSP.61f434aa" ]]TO [[CONTAINER "@MACROActiveTrendsHeaderSP.61f434aa" ]]
a4 - ADD SOURCE TO DEST[[CONTAINER "@MACROAvailableTrendsColored.61f434aa" ]]TO [[CONTAINER "@MACROAvailableTrendsColored.61f434aa" ]]
22 -PAINTER MacroOnLineTrend
r0 - EDIT SOURCE[[CONTAINER "@MACROActiveTrends.61f434aa" ]]
r1 - END PLCREPAINTS LIST
r2 - USE LOCAL TCR FORMAT[[STRING "NDIGIT3" ]]
23 -PAINTER EditBox
r0 - END PLCREPAINTS LIST
r1 - EDIT SOURCE[[CONTAINER "@MACROXMinTime.61f434aa" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - USE AWT COMPONENT

```

	Date	Name	File : C:\Users\helio\Desktop\WEBVIEW2\Main.teq
Creation	Monday, June 06, 2022		
Last modified	Thursday, June 09, 2022		Project : C:\Users\helio\Desktop\WEBVIEW2\Tese.prj
			Page : 3/9

```

r4 - USE INTERIOR COLOR[[COLOR "(255 , 255 , 255)" ]]
  ON CONDITION [[[ [CONTAINER "@MACROXAutoUpdate.61f434aa" ] == [STRING "MANUAL" ] ) ]]
r5 - USE 3D FRAME
a0 - END PLCACTIONS LIST ON NO_EVENT
a1 - POPUP DIALOG[[STRING "" ]]TO [[STRING "" ]] ON MOUSE_UP_EVENT
  ON CONDITION [[[ [CONTAINER "@MACROXAutoUpdate.61f434aa" ] == [STRING "MANUAL" ] ) ]]
24 -PAINTER EditBox
r0 - END PLCREPAINTS LIST
r1 - EDIT SOURCE[[CONTAINER "@MACROXMaxTime.61f434aa" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - TEXT RIGTH JUSTIFIED[[STRING "" ]]
a0 - END PLCACTIONS LIST ON NO_EVENT
a1 - POPUP DIALOG[[STRING "" ]]TO [[STRING "" ]] ON MOUSE_UP_EVENT
a2 - DISABLE ON CONDITION GROUP[[STRING "" ]]TO [[STRING "" ]] ON MOUSE_DOWN_EVENT
25 -PAINTER EditBox
r0 - END PLCREPAINTS LIST
r1 - EDIT SOURCE[[CONTAINER "@MACROXMinDate.61f434aa" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - USE AWT COMPONENT
r4 - USE INTERIOR COLOR[[COLOR "(255 , 255 , 255)" ]]
  ON CONDITION [[[ [CONTAINER "@MACROXAutoUpdate.61f434aa" ] == [STRING "MANUAL" ] ) ]]
r5 - USE 3D FRAME
a0 - END PLCACTIONS LIST ON NO_EVENT
a1 - POPUP DIALOG[[STRING "" ]]TO [[STRING "" ]] ON MOUSE_UP_EVENT
  ON CONDITION [[[ [CONTAINER "@MACROXAutoUpdate.61f434aa" ] == [STRING "MANUAL" ] ) ]]
26 -PAINTER EditBox
r0 - END PLCREPAINTS LIST
r1 - EDIT SOURCE[[CONTAINER "@MACROXMaxDate.61f434aa" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - TEXT RIGTH JUSTIFIED[[STRING "" ]]
a0 - END PLCACTIONS LIST ON NO_EVENT
a1 - POPUP DIALOG[[STRING "" ]]TO [[STRING "" ]] ON MOUSE_UP_EVENT
a2 - DISABLE ON CONDITION GROUP[[STRING "" ]]TO [[STRING "" ]] ON MOUSE_DOWN_EVENT
27 -PAINTER EditBox
r0 - END PLCREPAINTS LIST
r1 - EDIT SOURCE[[CONTAINER "@MACROONTXTime[8].61f434aa" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
28 -PAINTER EditBox
r0 - END PLCREPAINTS LIST
r1 - EDIT SOURCE[[CONTAINER "@MACROONTXDate[8].61f434aa" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
29 -PAINTER EditBox
r0 - END PLCREPAINTS LIST
r1 - EDIT SOURCE[[CONTAINER "@MACROONTXTime[6].61f434aa" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - TEXT WIDTH CENTERED[[STRING "" ]]
30 -PAINTER EditBox
r0 - END PLCREPAINTS LIST
r1 - EDIT SOURCE[[CONTAINER "@MACROONTXDate[6].61f434aa" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - TEXT WIDTH CENTERED[[STRING "" ]]
31 -PAINTER EditBox
r0 - END PLCREPAINTS LIST
r1 - EDIT SOURCE[[CONTAINER "@MACROONTXTime[4].61f434aa" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - TEXT WIDTH CENTERED[[STRING "" ]]
32 -PAINTER EditBox
r0 - END PLCREPAINTS LIST
r1 - EDIT SOURCE[[CONTAINER "@MACROONTXDate[4].61f434aa" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - TEXT WIDTH CENTERED[[STRING "" ]]

```

	Date	Name	File : C:\Users\helio\Desktop\WEBVIEW2\Main.teq
Creation	Monday, June 06, 2022		
Last modified	Thursday, June 09, 2022		Project : C:\Users\helio\Desktop\WEBVIEW2\Tese.prj
			Page : 4/9

## 33 -PAINTER EditBox

```
r0 - END PLCREPAINTS LIST
r1 - EDIT SOURCE[[CONTAINER "@MACROONTXTime[2].61f434aa" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - TEXT WIDTH CENTERED[[STRING "" ]]
```

## 34 -PAINTER EditBox

```
r0 - END PLCREPAINTS LIST
r1 - EDIT SOURCE[[CONTAINER "@MACROONTXDate[2].61f434aa" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - TEXT WIDTH CENTERED[[STRING "" ]]
```

## 35 -PAINTER Button

```
r0 - END PLCREPAINTS LIST
r1 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Save" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - TEXT WIDTH CENTERED[[STRING "" ]]
r4 - PAINT SOURCE[[FILE "btn_simple_3.svg|btn_simple_3_75_36.png" ]]
r5 - PAINT A SOURCE ON PRESSED[[FILE "btn_simple_p_3.svg|btn_simple_p_3_75_36.png" ]]
r6 - PAINT A SOURCE ON DISABLED[[FILE "btn_simple_d_3.svg|btn_simple_d_3_75_36.png" ]]
a0 - END PLCACTIONS LIST ON NO_EVENT
a1 - SOURCE TO DEST[[STRING "SAVELOGSTOFILE" ]]TO [[STRING "SAVELOGSTOFILE" ]] ON MOUSE_UP_EVENT
ON CONDITION [[[ CONTAINER "@MACROAvailableTrendsAction.61f434aa" ] == [STRING "0" ] ) ]]
```

## 36 -PAINTER Button

```
r0 - END PLCREPAINTS LIST
r1 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Clear" ]]
ON CONDITION [[[ CONTAINER "@MACROAvailableTrendsAction.61f434aa" ] == [STRING "0" ] ) ]]
```

```
r2 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Start" ]]
ON CONDITION [[[ CONTAINER "@MACROAvailableTrendsAction.61f434aa" ] == [STRING "LOGSWAITING" ] ) ]]
```

```
r3 - TEXT HEIGHT CENTERED[[STRING "" ]]
r4 - TEXT WIDTH CENTERED[[STRING "" ]]
r5 - PAINT SOURCE[[FILE "btn_simple_3.svg|btn_simple_3_75_36.png" ]]
r6 - PAINT A SOURCE ON PRESSED[[FILE "btn_simple_p_3.svg|btn_simple_p_3_75_36.png" ]]
r7 - PAINT A SOURCE ON DISABLED[[FILE "btn_simple_d_3.svg|btn_simple_d_3_75_36.png" ]]
a0 - END PLCACTIONS LIST ON NO_EVENT
a1 - SOURCE TO DEST[[STRING "CLEARNSTOPLOGS" ]]TO [[STRING "CLEARNSTOPLOGS" ]] ON MOUSE_UP_EVENT
ON CONDITION [[[ CONTAINER "@MACROAvailableTrendsAction.61f434aa" ] == [STRING "0" ] ) ]]
```

```
a2 - SOURCE TO DEST[[STRING "STARTLOGGING" ]]TO [[STRING "STARTLOGGING" ]] ON MOUSE_UP_EVENT
ON CONDITION [[[ CONTAINER "@MACROAvailableTrendsAction.61f434aa" ] == [STRING "LOGSWAITING" ] ) ]]
```

## 37 -PAINTER Button

```
r0 - END PLCREPAINTS LIST
r1 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Manual" ]]
ON CONDITION [[[ CONTAINER "@MACROXAutoUpdate.61f434aa" ] == [STRING "1" ] ) ]]
```

```
r2 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Auto" ]]
ON CONDITION [[[ CONTAINER "@MACROXAutoUpdate.61f434aa" ] == [STRING "MANUAL" ] ) ]]
```

```
r3 - TEXT HEIGHT CENTERED[[STRING "" ]]
r4 - TEXT WIDTH CENTERED[[STRING "" ]]
r5 - PAINT SOURCE[[FILE "btn_simple_3.svg|btn_simple_3_75_36.png" ]]
r6 - PAINT A SOURCE ON PRESSED[[FILE "btn_simple_p_3.svg|btn_simple_p_3_75_36.png" ]]
r7 - PAINT A SOURCE ON DISABLED[[FILE "btn_simple_d_3.svg|btn_simple_d_3_75_36.png" ]]
a0 - END PLCACTIONS LIST ON NO_EVENT
a1 - SOURCE TO DEST[[STRING "SETAUTO" ]]TO [[STRING "SETAUTO" ]] ON MOUSE_UP_EVENT
ON CONDITION [[[ CONTAINER "@MACROXAutoUpdate.61f434aa" ] == [STRING "MANUAL" ] ) ]]
```

```
a2 - SOURCE TO DEST[[STRING "MANUAL" ]]TO [[STRING "MANUAL" ]] ON MOUSE_UP_EVENT
ON CONDITION [[[ CONTAINER "@MACROXAutoUpdate.61f434aa" ] == [STRING "1" ] ) ]]
```

## 38 -PAINTER Button

```
r0 - END PLCREPAINTS LIST
r1 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Zoom In" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - TEXT WIDTH CENTERED[[STRING "" ]]
r4 - PAINT SOURCE[[FILE "btn_simple_3.svg|btn_simple_3_75_36.png" ]]
r5 - PAINT A SOURCE ON PRESSED[[FILE "btn_simple_p_3.svg|btn_simple_p_3_75_36.png" ]]
r6 - PAINT A SOURCE ON DISABLED[[FILE "btn_simple_d_3.svg|btn_simple_d_3_75_36.png" ]]
```

	Date	Name	File : C:\Users\helio\Desktop\WEBVIEW2\Main.teq
Creation	Monday, June 06, 2022		
Last modified	Thursday, June 09, 2022		Project : C:\Users\helio\Desktop\WEBVIEW2\Tese.prj
			Page : 5/9

```

a0 - END PLCACTIONS LIST ON NO_EVENT
a1 - SOURCE TO DEST[[STRING "50" ]]TO [[STRING "50" ]] ON MOUSE_DOWN_EVENT
39 -PAINTER Button
r0 - END PLCREPAINTS LIST
r1 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Zoom Out" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - TEXT WIDTH CENTERED[[STRING "" ]]
r4 - PAINT SOURCE[[FILE "btn_simple_3.svg|btn_simple_3_75_36.png" ]]
r5 - PAINT A SOURCE ON PRESSED[[FILE "btn_simple_p_3.svg|btn_simple_p_3_75_36.png" ]]
r6 - PAINT A SOURCE ON DISABLED[[FILE "btn_simple_d_3.svg|btn_simple_d_3_75_36.png" ]]
a0 - END PLCACTIONS LIST ON NO_EVENT
a1 - SOURCE TO DEST[[STRING "-100" ]]TO [[STRING "-100" ]] ON MOUSE_DOWN_EVENT
40 -PAINTER Button
r0 - END PLCREPAINTS LIST
r1 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "<< scroll" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - TEXT WIDTH CENTERED[[STRING "" ]]
r4 - PAINT SOURCE[[FILE "btn_simple_3.svg|btn_simple_3_75_36.png" ]]
r5 - PAINT A SOURCE ON PRESSED[[FILE "btn_simple_p_3.svg|btn_simple_p_3_75_36.png" ]]
r6 - PAINT A SOURCE ON DISABLED[[FILE "btn_simple_d_3.svg|btn_simple_d_3_75_36.png" ]]
a0 - END PLCACTIONS LIST ON NO_EVENT
a1 - SOURCE TO DEST[[STRING "-80" ]]TO [[STRING "-80" ]] ON MOUSE_DOWN_EVENT
41 -PAINTER Button
r0 - END PLCREPAINTS LIST
r1 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "< scroll" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - TEXT WIDTH CENTERED[[STRING "" ]]
r4 - PAINT SOURCE[[FILE "btn_simple_3.svg|btn_simple_3_75_36.png" ]]
r5 - PAINT A SOURCE ON PRESSED[[FILE "btn_simple_p_3.svg|btn_simple_p_3_75_36.png" ]]
r6 - PAINT A SOURCE ON DISABLED[[FILE "btn_simple_d_3.svg|btn_simple_d_3_75_36.png" ]]
a0 - END PLCACTIONS LIST ON NO_EVENT
a1 - SOURCE TO DEST[[STRING "-10" ]]TO [[STRING "-10" ]] ON MOUSE_DOWN_EVENT
42 -PAINTER Button
r0 - END PLCREPAINTS LIST
r1 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "scroll >" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - TEXT WIDTH CENTERED[[STRING "" ]]
r4 - PAINT SOURCE[[FILE "btn_simple_3.svg|btn_simple_3_75_36.png" ]]
r5 - PAINT A SOURCE ON PRESSED[[FILE "btn_simple_p_3.svg|btn_simple_p_3_75_36.png" ]]
r6 - PAINT A SOURCE ON DISABLED[[FILE "btn_simple_d_3.svg|btn_simple_d_3_75_36.png" ]]
a0 - END PLCACTIONS LIST ON NO_EVENT
a1 - SOURCE TO DEST[[STRING "10" ]]TO [[STRING "10" ]] ON MOUSE_DOWN_EVENT
43 -PAINTER Button
r0 - END PLCREPAINTS LIST
r1 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "scroll >>" ]]
r2 - TEXT HEIGHT CENTERED[[STRING "" ]]
r3 - TEXT WIDTH CENTERED[[STRING "" ]]
r4 - PAINT SOURCE[[FILE "btn_simple_3.svg|btn_simple_3_75_36.png" ]]
r5 - PAINT A SOURCE ON PRESSED[[FILE "btn_simple_p_3.svg|btn_simple_p_3_75_36.png" ]]
r6 - PAINT A SOURCE ON DISABLED[[FILE "btn_simple_d_3.svg|btn_simple_d_3_75_36.png" ]]
a0 - END PLCACTIONS LIST ON NO_EVENT
a1 - SOURCE TO DEST[[STRING "80" ]]TO [[STRING "80" ]] ON MOUSE_DOWN_EVENT
44 -PAINTER MacroNameHiddenLabel
r0 - HIDE PAINTER
r1 - EDIT SOURCE[[STRING "OnlineTrendMinimal_7_00_00" ]]
r2 - TEXT HEIGHT CENTERED
45 -PAINTER Button
r0 - PAINT SOURCE[[FILE "btn_simple_3.svg|btn_simple_3_8b_33.png" ]]
r1 - TEXT HEIGHT CENTERED
r2 - TEXT WIDTH CENTERED
r3 - PAINT A SOURCE ON PRESSED[[FILE "btn_simple_p_3.svg|btn_simple_p_3_8b_33.png" ]]

```

	Date	Name	File : C:\Users\helio\Desktop\WEBVIEW2\Main.teq
Creation	Monday, June 06, 2022		
Last modified	Thursday, June 09, 2022		Project : C:\Users\helio\Desktop\WEBVIEW2\Tese.prj
			Page : 6/9

r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_8b\_33.png" ]]  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Manual Mode" ]]  
 a0 - SOURCE TO DEST[[STRING "0" ]]TO [[STRING "0" ]] ON MOUSE\_DOWN\_EVENT  
 a1 - SOURCE TO DEST[[STRING "0" ]]TO [[STRING "0" ]] ON MOUSE\_UP\_EVENT  
 46 -PAINTER Button  
 r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_8b\_33.png" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_8b\_33.png" ]]  
 r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_8b\_33.png" ]]  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "MPPT Mode" ]]  
 a0 - SOURCE TO DEST[[STRING "0" ]]TO [[STRING "0" ]] ON MOUSE\_DOWN\_EVENT  
 a1 - SOURCE TO DEST[[STRING "1" ]]TO [[STRING "1" ]] ON MOUSE\_UP\_EVENT  
 47 -PAINTER Button  
 r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_8b\_33.png" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_8b\_33.png" ]]  
 r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_8b\_33.png" ]]  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Battery Mode" ]]  
 a0 - SOURCE TO DEST[[STRING "0" ]]TO [[STRING "0" ]] ON MOUSE\_DOWN\_EVENT  
 a1 - SOURCE TO DEST[[STRING "2" ]]TO [[STRING "2" ]] ON MOUSE\_UP\_EVENT  
 48 -PAINTER EditText  
 r0 - EDIT SOURCE[[PPO "Main.Vbat" ]]  
 r1 - USE AWT COMPONENT  
 r2 - TEXT HEIGHT CENTERED  
 r3 - USE 3D FRAME  
 a0 - POPUP DIALOG ON MOUSE\_DOWN\_EVENT  
 49 -PAINTER StaticText  
 r0 - EDIT SOURCE[[HTML TAG "Battery Voltage:" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 50 -PAINTER StaticText  
 r0 - EDIT SOURCE[[HTML TAG "Electrical Values PLC" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 51 -PAINTER StaticText  
 r0 - EDIT SOURCE[[HTML TAG "Operation Mode" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 52 -PAINTER StaticText  
 r0 - EDIT SOURCE[[HTML TAG "Duty Cicle" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 53 -PAINTER Rectangle  
 a0 - END PLCACTIONS LIST[[STRING "" ]]TO [[STRING "" ]] ON NO\_EVENT  
 a1 - GET PAINTER ORG W IN DESTINATION[[STRING "" ]]TO [[STRING "" ]] ON GAINED\_FOCUS\_EVENT  
 54 -PAINTER Polyline  
 55 -PAINTER Button  
 r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_28\_1e.png" ]]  
 r1 - TEXT HEIGHT CENTERED[[STRING "" ]]  
 r2 - TEXT WIDTH CENTERED[[STRING "" ]]  
 r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_28\_1e.png" ]]  
 r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_28\_1e.png" ]]  
 a0 - END PLCACTIONS LIST[[STRING "" ]]TO [[STRING "" ]] ON NO\_EVENT  
 a1 - GET PAINTER ORG W IN DESTINATION[[STRING "" ]]TO [[STRING "" ]] ON GAINED\_FOCUS\_EVENT  
 a2 - WRITE OPERATION RESULT IN DESTINATION[[STRING "SUB" ]]TO [[STRING "SUB" ]] ON GAINED\_FOCUS\_EVENT  
 a3 - REGISTER DRAG EVENT[[STRING "" ]]TO [[STRING "" ]] ON MOUSE\_DOWN\_EVENT  
 56 -PAINTER MacroNameHiddenLabel  
 r0 - HIDE PAINTER  
 r1 - EDIT SOURCE[[STRING "HorizontalThinSlider6\_50\_00" ]]

	Date	Name	File : C:\Users\helio\Desktop\WEBVIEW2\Main.teq
Creation	Monday, June 06, 2022		
Last modified	Thursday, June 09, 2022		Project : C:\Users\helio\Desktop\WEBVIEW2\Tese.prj
			Page : 7/9

r2 - TEXT HEIGHT CENTERED

57 -PAINTER Rectangle  
a0 - END PLCACTIONS LIST[[STRING "" ]]TO [[STRING "" ]] ON NO\_EVENT  
a1 - GET PAINTER ORG W IN DESTINATION[[STRING "" ]]TO [[STRING "" ]] ON GAINED\_FOCUS\_EVENT

58 -PAINTER Polyline

59 -PAINTER Button  
r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_28\_1e.png" ]]  
r1 - TEXT HEIGHT CENTERED[[STRING "" ]]  
r2 - TEXT WIDTH CENTERED[[STRING "" ]]  
r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_28\_1e.png" ]]  
r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_28\_1e.png" ]]  
a0 - END PLCACTIONS LIST[[STRING "" ]]TO [[STRING "" ]] ON NO\_EVENT  
a1 - GET PAINTER ORG W IN DESTINATION[[STRING "" ]]TO [[STRING "" ]] ON GAINED\_FOCUS\_EVENT  
a2 - WRITE OPERATION RESULT IN DESTINATION[[STRING "SUB" ]]TO [[STRING "SUB" ]] ON GAINED\_FOCUS\_EVENT  
a3 - REGISTER DRAG EVENT[[STRING "" ]]TO [[STRING "" ]] ON MOUSE\_DOWN\_EVENT

60 -PAINTER MacroNameHiddenLabel  
r0 - HIDE PAINTER  
r1 - EDIT SOURCE[[STRING "HorizontalThinSlider6\_50\_00" ]]  
r2 - TEXT HEIGHT CENTERED

61 -PAINTER StaticText  
r0 - EDIT SOURCE[[PPO "Communication.PIC\_V" ]]  
r1 - TEXT HEIGHT CENTERED  
r2 - TEXT WIDTH CENTERED

62 -PAINTER StaticText  
r0 - EDIT SOURCE[[PPO "Communication.PIC\_I" ]]  
r1 - TEXT HEIGHT CENTERED  
r2 - TEXT WIDTH CENTERED

63 -PAINTER StaticText  
r0 - EDIT SOURCE[[HTML TAG "Voltage [V]" ]]  
r1 - TEXT HEIGHT CENTERED  
r2 - TEXT WIDTH CENTERED

64 -PAINTER StaticText  
r0 - EDIT SOURCE[[HTML TAG "Current [A]" ]]  
r1 - TEXT HEIGHT CENTERED  
r2 - TEXT WIDTH CENTERED

65 -PAINTER StaticText  
r0 - EDIT SOURCE[[HTML TAG "Power [W]" ]]  
r1 - TEXT HEIGHT CENTERED  
r2 - TEXT WIDTH CENTERED

66 -PAINTER StaticText  
r0 - EDIT SOURCE[[PPO "Communication.PIC\_P" ]]  
r1 - TEXT HEIGHT CENTERED  
r2 - TEXT WIDTH CENTERED

67 -PAINTER StaticText  
r0 - EDIT SOURCE[[HTML TAG "Electrical Values PIC" ]]  
r1 - TEXT HEIGHT CENTERED  
r2 - TEXT WIDTH CENTERED

68 -PAINTER Button  
r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_8b\_33.png" ]]  
r1 - TEXT HEIGHT CENTERED  
r2 - TEXT WIDTH CENTERED  
r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_8b\_33.png" ]]  
r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_8b\_33.png" ]]  
r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Load Ramp" ]]  
a0 - TOGGLE SOURCE[[PPO "AnalogOutput.testeCarga" ]] ON MOUSE\_UP\_EVENT

69 -PAINTER StaticText  
r0 - EDIT SOURCE[[HTML TAG "(Charge Controller)" ]]  
r1 - TEXT HEIGHT CENTERED  
r2 - TEXT WIDTH CENTERED

70 -PAINTER StaticText  
r0 - EDIT SOURCE[[HTML TAG "(Photovoltaic Panel)" ]]

	Date	Name	File : C:\Users\helio\Desktop\WEBVIEW2\Main.teq
Creation	Monday, June 06, 2022		
Last modified	Thursday, June 09, 2022		Project : C:\Users\helio\Desktop\WEBVIEW2\Tese.prj
			Page : 8/9

r1 - TEXT HEIGHT CENTERED

r2 - TEXT WIDTH CENTERED

71 -PAINTER MultilineLabel

r0 - EDIT SOURCE[[STRING "Manual Mode: No control method is performed by the PLC.\nMPPT Mode: The boost converter duty cycle will

r1 - ENABLE AUTOADJUST MULTILINE TEXT

72 -PAINTER MultilineLabel

r0 - EDIT SOURCE[[STRING "Increases duty cycle of load converter by progressive steps every 10 sec." ]]

r1 - ENABLE AUTOADJUST MULTILINE TEXT

73 -PAINTER EditText

r0 - EDIT SOURCE[[PPO "@GV.Vcarga" ]]

r1 - USE AWT COMPONENT

r2 - TEXT HEIGHT CENTERED

r3 - USE 3D FRAME

a0 - POPUP DIALOG ON MOUSE\_DOWN\_EVENT

74 -PAINTER EditText

r0 - EDIT SOURCE[[PPO "@GV.Vref" ]]

r1 - USE AWT COMPONENT

r2 - TEXT HEIGHT CENTERED

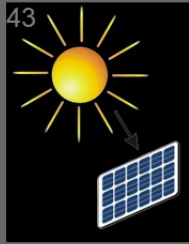
r3 - USE 3D FRAME

a0 - POPUP DIALOG ON MOUSE\_DOWN\_EVENT

	Date	Name	File : C:\Users\helio\Desktop\WEBVIEW2\Main.teq
Creation	Monday, June 06, 2022		
Last modified	Thursday, June 09, 2022		Project : C:\Users\helio\Desktop\WEBVIEW2\Tese.prj
			Page : 9/9

44

# Parameter Adjustment



## 4 Environment Parameters

5	Temperature	Communication.PIC_T	Communication.HMI_T	32	Write T
6	Irradiance	Communication.PIC_G	Communication.HMI_G	33	Write G

2 Read All

Reads all parameters currently running on the microcontroller.

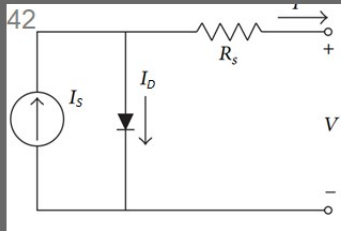
31 Copy All

Copies all parameters from static text to text boxes.

41 Write All

Writes all parameters to the microcontroller.

## 45 Photovoltaic Panel Parameters



7	Voc [STC]	Communication.PIC_V	Communication.HMI_V	34	Write Voc
8	Isc [STC]	Communication.PIC_I	Communication.HMI_I	35	Write Isc
9	Series Resistance	Communication.PIC_R	Communication.HMI_R	36	Write Rs
10	Kv coefficient	Communication.PIC_K	Communication.HMI_K	37	Write Kv
11	Ki coefficient	Communication.PIC_N	Communication.HMI_N	38	Write Ki
12	Diode ideality factor	Communication.PIC_N	Communication.HMI_N	39	Write N
13	Number of cells	Communication.PIC_N	Communication.HMI_N	40	Write Nce

30 Main Screen

### TEQ Information for Parameters.teq

#### 1 -PAINTER EditBox

r0 - EDIT SOURCE[[PPO "Communication.HMI\_T" ]]  
 r1 - USE AWT COMPONENT  
 r2 - TEXT HEIGHT CENTERED  
 r3 - USE 3D FRAME  
 a0 - POPUP DIALOG ON MOUSE\_DOWN\_EVENT

#### 2 -PAINTER Button

r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_d2\_30.png" ]]  
 r1 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_d2\_30.png" ]]  
 r2 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_d2\_30.png" ]]  
 r3 - TEXT WIDTH CENTERED  
 r4 - TEXT HEIGHT CENTERED  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Read All" ]]  
 a0 - TOGGLE SOURCE[[PPO "Communication.readALL" ]] ON MOUSE\_UP\_EVENT

#### 3 -PAINTER StaticText

r0 - EDIT SOURCE[[PPO "Communication.PIC\_T" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED

#### 4 -PAINTER StaticText

r0 - EDIT SOURCE[[HTML TAG "Environment Parameters" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED

#### 5 -PAINTER StaticText

r0 - EDIT SOURCE[[HTML TAG "Temperature" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED

#### 6 -PAINTER StaticText

	Date	Name	File : C:\Users\helio\Desktop\WEBVIEW2\Parameters.teq
Creation	Monday, June 06, 2022		
Last modified	Thursday, June 09, 2022		Project : C:\Users\helio\Desktop\WEBVIEW2\Tese.prj
			Page : 1/5

r0 - EDIT SOURCE[[HTML TAG "Irradiance" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 7 -PAINTER StaticText  
 r0 - EDIT SOURCE[[HTML TAG "Voc [STC]" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 8 -PAINTER StaticText  
 r0 - EDIT SOURCE[[HTML TAG "Isc [STC]" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 9 -PAINTER StaticText  
 r0 - EDIT SOURCE[[HTML TAG "Series Resistance" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 10 -PAINTER StaticText  
 r0 - EDIT SOURCE[[HTML TAG "Kv coeficient" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 11 -PAINTER StaticText  
 r0 - EDIT SOURCE[[HTML TAG "Ki coeficient" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 12 -PAINTER StaticText  
 r0 - EDIT SOURCE[[HTML TAG "Diode ideality factor" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 13 -PAINTER StaticText  
 r0 - EDIT SOURCE[[HTML TAG "Number of cells" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 14 -PAINTER StaticText  
 r0 - EDIT SOURCE[[PPO "Communication.PIC\_G" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 15 -PAINTER StaticText  
 r0 - EDIT SOURCE[[PPO "Communication.PIC\_VOC" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 16 -PAINTER StaticText  
 r0 - EDIT SOURCE[[PPO "Communication.PIC\_ISC" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 17 -PAINTER StaticText  
 r0 - EDIT SOURCE[[PPO "Communication.PIC\_RS" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 18 -PAINTER StaticText  
 r0 - EDIT SOURCE[[PPO "Communication.PIC\_KV" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 19 -PAINTER StaticText  
 r0 - EDIT SOURCE[[PPO "Communication.PIC\_KI" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 20 -PAINTER StaticText  
 r0 - EDIT SOURCE[[PPO "Communication.PIC\_N" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 21 -PAINTER StaticText  
 r0 - EDIT SOURCE[[PPO "Communication.PIC\_NCE" ]]

	Date	Name	File : C:\Users\helio\Desktop\WEBVIEW2\Parameters.teq
Creation	Monday, June 06, 2022		
Last modified	Thursday, June 09, 2022		Project : C:\Users\helio\Desktop\WEBVIEW2\Tese.prj
			Page : 2/5

r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 22 -PAINTER EditText  
 r0 - EDIT SOURCE[[PPO "Communication.HMI\_G" ]]  
 r1 - USE AWT COMPONENT  
 r2 - TEXT HEIGHT CENTERED  
 r3 - USE 3D FRAME  
 a0 - POPUP DIALOG ON MOUSE\_DOWN\_EVENT  
 23 -PAINTER EditText  
 r0 - EDIT SOURCE[[PPO "Communication.HMI\_VOC" ]]  
 r1 - USE AWT COMPONENT  
 r2 - TEXT HEIGHT CENTERED  
 r3 - USE 3D FRAME  
 a0 - POPUP DIALOG ON MOUSE\_DOWN\_EVENT  
 24 -PAINTER EditText  
 r0 - EDIT SOURCE[[PPO "Communication.HMI\_ISC" ]]  
 r1 - USE AWT COMPONENT  
 r2 - TEXT HEIGHT CENTERED  
 r3 - USE 3D FRAME  
 a0 - POPUP DIALOG ON MOUSE\_DOWN\_EVENT  
 25 -PAINTER EditText  
 r0 - EDIT SOURCE[[PPO "Communication.HMI\_RS" ]]  
 r1 - USE AWT COMPONENT  
 r2 - TEXT HEIGHT CENTERED  
 r3 - USE 3D FRAME  
 a0 - POPUP DIALOG ON MOUSE\_DOWN\_EVENT  
 26 -PAINTER EditText  
 r0 - EDIT SOURCE[[PPO "Communication.HMI\_KV" ]]  
 r1 - USE AWT COMPONENT  
 r2 - TEXT HEIGHT CENTERED  
 r3 - USE 3D FRAME  
 a0 - POPUP DIALOG ON MOUSE\_DOWN\_EVENT  
 27 -PAINTER EditText  
 r0 - EDIT SOURCE[[PPO "Communication.HMI\_KI" ]]  
 r1 - USE AWT COMPONENT  
 r2 - TEXT HEIGHT CENTERED  
 r3 - USE 3D FRAME  
 a0 - POPUP DIALOG ON MOUSE\_DOWN\_EVENT  
 28 -PAINTER EditText  
 r0 - EDIT SOURCE[[PPO "Communication.HMI\_N" ]]  
 r1 - USE AWT COMPONENT  
 r2 - TEXT HEIGHT CENTERED  
 r3 - USE 3D FRAME  
 a0 - POPUP DIALOG ON MOUSE\_DOWN\_EVENT  
 29 -PAINTER EditText  
 r0 - EDIT SOURCE[[PPO "Communication.HMI\_NCE" ]]  
 r1 - USE AWT COMPONENT  
 r2 - TEXT HEIGHT CENTERED  
 r3 - USE 3D FRAME  
 a0 - POPUP DIALOG ON MOUSE\_DOWN\_EVENT  
 30 -PAINTER Button  
 r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_8b\_39.png" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_8b\_39.png" ]]  
 r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_8b\_39.png" ]]  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Main Screen" ]]  
 a0 - SOURCE TO DEST[[CONTAINER "@LAST\_PAGE" ]]TO [[CONTAINER "@LAST\_PAGE" ]] ON MOUSE\_UP\_EVENT  
 31 -PAINTER Button  
 r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_d2\_30.png" ]]  
 r1 - TEXT HEIGHT CENTERED

	Date	Name	File : C:\Users\helio\Desktop\WEBVIEW2\Parameters.teq
Creation	Monday, June 06, 2022		
Last modified	Thursday, June 09, 2022		Project : C:\Users\helio\Desktop\WEBVIEW2\Tese.prj
			Page : 3/5

r2 - TEXT WIDTH CENTERED  
 r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_d2\_30.png" ]]  
 r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_d2\_30.png" ]]  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Copy All" ]]  
 a0 - TOGGLE SOURCE[[PPO "Communication.copyButton" ]] ON MOUSE\_UP\_EVENT

## 32 -PAINTER Button

r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_8c\_20.png" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_8c\_20.png" ]]  
 r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_8c\_20.png" ]]  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Write T" ]]  
 a0 - TOGGLE SOURCE[[PPO "Communication.writeT" ]] ON MOUSE\_UP\_EVENT

## 33 -PAINTER Button

r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_8c\_20.png" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_8c\_20.png" ]]  
 r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_8c\_20.png" ]]  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Write G" ]]  
 a0 - TOGGLE SOURCE[[PPO "Communication.writeG" ]] ON MOUSE\_UP\_EVENT

## 34 -PAINTER Button

r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_8c\_20.png" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_8c\_20.png" ]]  
 r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_8c\_20.png" ]]  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Write Voc" ]]  
 a0 - TOGGLE SOURCE[[PPO "Communication.writeVOC" ]] ON MOUSE\_UP\_EVENT

## 35 -PAINTER Button

r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_8c\_20.png" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_8c\_20.png" ]]  
 r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_8c\_20.png" ]]  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Write lsc" ]]  
 a0 - TOGGLE SOURCE[[PPO "Communication.writeISC" ]] ON MOUSE\_UP\_EVENT

## 36 -PAINTER Button

r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_8c\_20.png" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_8c\_20.png" ]]  
 r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_8c\_20.png" ]]  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Write Rs" ]]  
 a0 - TOGGLE SOURCE[[PPO "Communication.writeRS" ]] ON MOUSE\_UP\_EVENT

## 37 -PAINTER Button

r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_8c\_20.png" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_8c\_20.png" ]]  
 r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_8c\_20.png" ]]  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Write Kv" ]]  
 a0 - TOGGLE SOURCE[[PPO "Communication.writeKV" ]] ON MOUSE\_UP\_EVENT

## 38 -PAINTER Button

r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_8c\_20.png" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_8c\_20.png" ]]  
 r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_8c\_20.png" ]]  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Write Ki" ]]  
 a0 - TOGGLE SOURCE[[PPO "Communication.writeKI" ]] ON MOUSE\_UP\_EVENT

	Date	Name	File : C:\Users\helio\Desktop\WEBVIEW2\Parameters.teq
Creation	Monday, June 06, 2022		
Last modified	Thursday, June 09, 2022		Project : C:\Users\helio\Desktop\WEBVIEW2\Tese.prj
			Page : 4/5

## 39 -PAINTER Button

r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_8c\_20.png" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_8c\_20.png" ]]  
 r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_8c\_20.png" ]]  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Write N" ]]  
 a0 - TOGGLE SOURCE[[PPO "Communication.writeN" ]] ON MOUSE\_UP\_EVENT

## 40 -PAINTER Button

r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_8c\_20.png" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_8c\_20.png" ]]  
 r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_8c\_20.png" ]]  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Write Nce" ]]  
 a0 - TOGGLE SOURCE[[PPO "Communication.writeNCE" ]] ON MOUSE\_UP\_EVENT

## 41 -PAINTER Button

r0 - PAINT SOURCE[[FILE "btn\_simple\_3.svg|btn\_simple\_3\_d2\_30.png" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED  
 r3 - PAINT A SOURCE ON PRESSED[[FILE "btn\_simple\_p\_3.svg|btn\_simple\_p\_3\_d2\_30.png" ]]  
 r4 - PAINT A SOURCE ON DISABLED[[FILE "btn\_simple\_d\_3.svg|btn\_simple\_d\_3\_d2\_30.png" ]]  
 r5 - BLEND A SOURCE OVER PREVIOUS[[HTML TAG "Write All" ]]  
 a0 - TOGGLE SOURCE[[PPO "Communication.writeALL" ]] ON MOUSE\_UP\_EVENT

## 42 -PAINTER Image

r0 - PAINT SOURCE[[FILE "4p.png" ]]  
 r1 - RESIZE IMAGE

## 43 -PAINTER Image

r0 - PAINT SOURCE[[FILE "sun-panel-removebg-preview.png" ]]

## 44 -PAINTER StaticText

r0 - EDIT SOURCE[[HTML TAG "Parameter Adjustment" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED

## 45 -PAINTER StaticText

r0 - EDIT SOURCE[[HTML TAG "Photovoltaic Panel Parameters" ]]  
 r1 - TEXT HEIGHT CENTERED  
 r2 - TEXT WIDTH CENTERED

## 46 -PAINTER MultilineLabel

r0 - EDIT SOURCE[[STRING "Reads all parameters currently running on the microcontroller." ]]  
 r1 - ENABLE AUTOADJUST MULTILINE TEXT

## 47 -PAINTER MultilineLabel

r0 - EDIT SOURCE[[STRING "Copies all parameters from static text to text boxes." ]]  
 r1 - ENABLE AUTOADJUST MULTILINE TEXT

## 48 -PAINTER MultilineLabel

r0 - EDIT SOURCE[[STRING "Writes all parameters to the microcontroller." ]]  
 r1 - ENABLE AUTOADJUST MULTILINE TEXT

	Date	Name	File : C:\Users\helio\Desktop\WEBVIEW2\Parameters.teq
Creation	Monday, June 06, 2022		
Last modified	Thursday, June 09, 2022		Project : C:\Users\helio\Desktop\WEBVIEW2\Tese.prj
			Page : 5/5