



Universidade do Minho
Escola de Engenharia

SOLAR TRACKER SYSTEM

Diego Soares Brandão	97769
Henrique Nogueira Magalhães	94939
João Pedro Machado da Silva	95610
José Pedro Azevedo Leite	95247
Rui Pedro Fernandes Pedroso	96868
Tiago Leal Pereira	96008

Projeto Integrador em Engenharia Eletrónica Industrial e Computadores 2

Projeto orientado por:

Professor Dr. Jaime Francisco Cruz Fonseca

Índice:

1 - Introdução:.....	4
2 - Arquitetura de Hardware:.....	5
2.1 - Estudo e seleção dos sensores:.....	6
2.1.1 - Dimensionamento dos sensores de corrente e tensão.....	6
2.2 - Controlo dos motores:	9
2.3 - Interface PV - ESS	10
3 - Arquitetura de <i>software</i> :.....	14
3.1 - Acionamento dos motores a partir dos LDRs:	14
3.2 - Algoritmo MPPT (Perturbação e Observação).....	15
3.3 - Servidor MQTT e módulo <i>Wi-Fi</i> a partir da ESP32:	17
4 - Conclusão e trabalho futuro:.....	20
5 - Referências:.....	21

Índice de figuras

Figura 2.1 - Hardware do STS	5
Figura 2.2 - Circuito final do sensor de corrente.....	8
Figura 2.3 - Circuito final do sensor de tensão	9
Figura 2.4 - Conversor de tensão para -5V	9
Figura 2.5 - Diagrama de blocos do acionamento dos motores	10
Figura 2.6 - Diagrama de blocos da interface PV-E.S.S.....	10
Figura 2.7 - Valores calculados do conversor CC-CC boost (4)	11
Figura 2.8 - Circuito do conversor CC-CC Boost.....	11
Figura 2.9 - Circuito com o MOSFET e opto acoplador.....	12
Figura 2.10 - Conversor CC-CC Buck.....	12
Figura 2.11 - Battery Management System (BMS)	13
Figura 3.1 - Diagrama de blocos do sistema desenvolvido	14
Figura 3.2 - Fluxograma da função de comparação dos LDRs	15
Figura 3.3 - Curva potência - tensão do PV ()	16
Figura 3.4 - Fluxograma do algoritmo MPPT	17
Figura 3.5 - Fluxograma do algoritmo desenvolvido na ESP 32.....	19

Lista de Siglas e Acrónimos

STS – *Solar Tracker System*

MQTT – *Message Queuing Telemetry Transport*

MPPT – *Maximum Power Point Tracking*

HMI – *Human Machine Interface*

LDR – *Light Dependent Resistor*

PWM – *Pulse Width Modulation*

ADC – *Analog to Digital Converter*

Wi-Fi – *Wireless Fidelity*

PV – *Photovoltaic Panels*

CMRR – *Common-Mode Rejection Ratio*

IC – *Integrated Circuit*

1 - Introdução:

Nos últimos anos, temos testemunhado uma crescente preocupação global com a questão da sustentabilidade e a necessidade de reduzir a dependência de fontes de energia não renováveis e altamente poluentes. Nesse contexto, as energias renováveis emergem como uma solução promissora para enfrentar os desafios do aquecimento global, da escassez de recursos e da degradação ambiental.

A energia solar tem se destacado como uma das principais fontes de energia renovável e sustentável disponíveis atualmente. Derivada do sol, uma fonte inesgotável e amplamente acessível, sem exceções, a energia solar oferece uma solução promissora para reduzir a dependência de combustíveis fósseis e mitigar os impactos negativos das mudanças climáticas.

Com o propósito de fazer um estudo do território para uma instalação de painéis fotovoltaicos em grande escala, no âmbito da unidade curricular de Projeto Integrador em Engenharia Eletrônica Industrial e Computadores 2, propôs-se criar um *Solar Tracker System*. Este sistema tem a capacidade de seguir perpendicularmente a orientação solar e obter dados provenientes de sensores para concluir qual a melhor posição e disposição dos painéis fotovoltaicos a ser instalados no terreno. Os dados recolhidos pelos sensores são enviados para um servidor MQTT e apresentados numa HMI, onde é possível verificar a variação dos valores na última hora bem como o último valor obtido pelos mesmos. Uma vez que os painéis fotovoltaicos utilizados possuem um baixo rendimento, foi utilizado um algoritmo MPPT (Perturbação e Observação) de forma a extrair a máxima potência possível. A manutenção necessária aos painéis fotovoltaicos e a necessidade de utilizar uma fonte para alimentação dos motores correspondem a alguns entraves do sistema.

Para o projeto, foram delineados alguns objetivos gerais:

- ✓ Orientar os painéis perpendicularmente à radiação solar;
- ✓ Colocar os painéis numa posição de referência quando a luminosidade se encontra abaixo de um certo nível (como é o caso do período noturno);
- ✓ Implementação de um algoritmo MPPT para extrair a máxima potência dos painéis;
- ✓ Aproveitamento da energia fornecida pelos painéis para alimentação do circuito de controlo do sistema;
- ✓ Desenvolvimento de uma interface entre um utilizador e sistema para análise dos dados fornecidos pelo sistema.

2 - Arquitetura de Hardware:

O hardware do STS envolve a integração de componentes eletrônicos e mecânicos num sistema coordenado capaz de seguir o movimento do sol. O objetivo principal é direcionar os painéis fotovoltaicos de forma a ficarem perpendiculares à radiação solar, aumentando assim a quantidade de energia capturada em comparação com sistemas estáticos.

O hardware deste sistema é composto por:

- ✓ 4 painéis fotovoltaicos policristalinos de 5W cada (1);
- ✓ 2 motores de passo Nema 17(2);
- ✓ 3 sensores de intensidade luminosa;
- ✓ 1 sensor de temperatura e humidade (3);
- ✓ 2 sensores de presença com contacto;
- ✓ 1 sensor de posição;
- ✓ 1 microcontrolador ESP32 DevKit V4;
- ✓ 1 microcontrolador STM32 F767ZI;

Nota: Em anexo, seguem 2 imagens do hardware do STS, com a legenda dos respetivos elementos constituintes.

A figura seguinte representa uma visão geral do STS.



Figura 2.1 - Hardware do STS

Neste capítulo será demonstrada a abordagem das diferentes partes de hardware do sistema.

2.1 - Estudo e seleção dos sensores:

No STS são utilizados 6 tipos de sensores: sensores de luminosidade, sensor de temperatura e humidade, sensor de corrente, sensor de tensão, sensores de presença e sensores de posição.

Para os sensores de luminosidade, optou-se por utilizar LDRs, sensores estes que variam o seu valor de resistência consoante a intensidade de luz incidente nos mesmos (quanto maior a intensidade da luz menor o valor de resistência e vice-versa). Estes sensores são utilizados neste projeto com o objetivo de acionar os motores de passo de modo a orientar os painéis perpendicularmente à incidência da luz solar, de acordo com as diferenças entre os valores de resistência entre os LDRs (se a resistência de um LDR for substancialmente maior que a de outro ocorrerá o movimento num eixo).

Para o sensor de temperatura e humidade, utilizou-se o DHT11, um módulo preparado para obter as duas grandezas físicas (temperatura e humidade ambientes) que são usados como estudo para inferir quais os impactos destas variáveis na produção de energia elétrica.

O sensor de tensão foi dimensionado de forma que, colocado em paralelo com os terminais do PV, consiga representar uma tensão entre 0 – 22.6V, numa escala suportável por um pino de um canal de um ADC, isto é, no caso, uma escala de 0 - 3V.

No que diz respeito aos sensores de presença optou-se por dois interruptores de fim de curso que embora esteja associado a um desgaste mecânico por se tratar de um sensor de presença com contacto, a sua aplicação no STS não provoca um desgaste significativo. Entre outros, este sensor foi escolhido uma vez que é de fácil utilização, manutenção e devido ao seu baixo custo.

No que se refere ao sensor de posição, utilizou-se um potenciómetro multivoltas (dado o curso do movimento no eixo altitude) preso ao eixo do motor superior com o propósito de obter a posição dos painéis no eixo altitude e definir os limites de curso do mesmo.

Relativamente ao sensor de corrente, foi utilizada uma resistência com um valor óhmico baixo (0.1Ω) que será colocada em série com o PV permitindo que, na presença de uma corrente, seja gerada uma tensão aos terminais da mesma. Essa tensão ditará o valor da corrente que é fornecida pelo PV. No entanto, devido ao seu valor óhmico, quando a corrente presente é na ordem dos amperes, a tensão gerada é na ordem dos milivolts, sendo necessária amplificação da mesma para posterior leitura a partir de um canal de um ADC do microcontrolador.

2.1.1 - Dimensionamento dos sensores de corrente e tensão

Primeiramente, dimensionou-se o sensor de corrente. Sabemos, a partir do *datasheet* dos painéis presentes, que a corrente máxima fornecida por um único painel, I_{SC} , é de 0.31A. Visto que os quatro painéis estão ligados em paralelo, a corrente resultante, I_{TOTAL} , é de 1.24A ($0.31A * 4$). Como a resistência *shunt* que mede a corrente é de 0.1Ω , a tensão máxima aos terminais da mesma, $V_{MÁX,SHUNT}$, é de **124mV** ($0.1\Omega * 1.24A$).

Esta diferença de potencial tem de ser convertida por um ADC. No entanto, qualquer ADC da STM32 tem dificuldade em ler qualquer tensão inferior a 0.3V com bastante definição, isto é, com pouca margem de erro. Para contornar este problema, a

tensão da resistência *shunt* tem de possuir um estágio intermédio de amplificação, de modo a conseguir ser lida corretamente pelo ADC. Este estágio de amplificação é feito com recurso a um INA126P e a um LM741 para correção de *offset* do anteriormente mencionado. Definiu-se, que para uma tensão de entrada de 124mV, o INA126P teria uma tensão de saída de 3V. Podemos determinar qual o valor do ganho que será aplicado a este CI segundo as expressões:

$$V_{out, INA} = (V_{+, INA} - V_{-, INA}) * G \quad (1)$$

Se queremos um V_{OUT} de 3V, para um V_{IN} de 124mV ($V_{+, INA} - V_{-, INA}$), o valor do ganho terá de ser igual a, aproximadamente, **24.1935**.

Neste circuito integrado, o ganho é definido colocando uma resistência entre os pinos 1 e 8. Do *datasheet*, sabemos que o valor da mesma pode ser calculado pela seguinte expressão:

$$G = 5 + \frac{80000}{R_G} \quad (2)$$

Se o ganho pretendido é o mencionado acima, a resistência de ganho tem o valor de, aproximadamente, **4168Ω**. De forma a minimizar erros e obter o valor de ganho mais próximo possível do calculado, usou-se um potenciômetro de 5kΩ, regulando-o de acordo com a saída e a entrada prevista.

Como nenhum componente eletrónico é perfeito, e o INA126P não é exceção, é necessário corrigir algumas características intrínsecas aos AmpOps que produzem um efeito indesejado na saída. Neste caso, o ganho em modo comum, A_{CM} . Quando as entradas são iguais, idealmente não deveria existir uma tensão de saída, no entanto, os AmpOps amplificam mínimas diferenças residuais, gerando um *offset DC*.

No caso desta aplicação, este último afeta bastante o comportamento do sistema: em termos práticos, a STM32 controla o conversor CC-CC elevador conforme o valor de tensão (representativo de uma corrente) que recebe e, com este *offset*, acaba por o controlar de uma maneira que não corresponde ao previsto.

Apesar de todos os ICs com AmpOps possuírem métodos para aumentar o CMRR, é ainda necessário minimizar o ganho em modo comum com uma tensão de *offset* recebida no pino 5 do INA126P. Para isso é usado um LM741 em configuração de seguidor de tensão com um barramento que permite regular esta tensão de *offset*. Este circuito foi dimensionado seguindo os seguintes cálculos:

$$V_{OS, INA} = 250 \mu V \quad (3)$$

$$G_{min} = 5 \text{ (defeito)} \quad (4)$$

$$G_{max} = 100 \quad (5)$$

$$V_{OS, INA} \times G_{mas} = 25 \text{ mV} \quad (6)$$

Para $R8 = 20k\Omega$ (potenciômetro) e definindo um gama de regulação de 100mV a -100mV:

$$I = \frac{200mV}{20k\Omega} = 10 \mu A \quad (7)$$

$$R9 = R10 = \frac{5 - 0.1}{0.00001} = 490k\Omega \quad (8)$$

O circuito final fica como na figura 2.2:

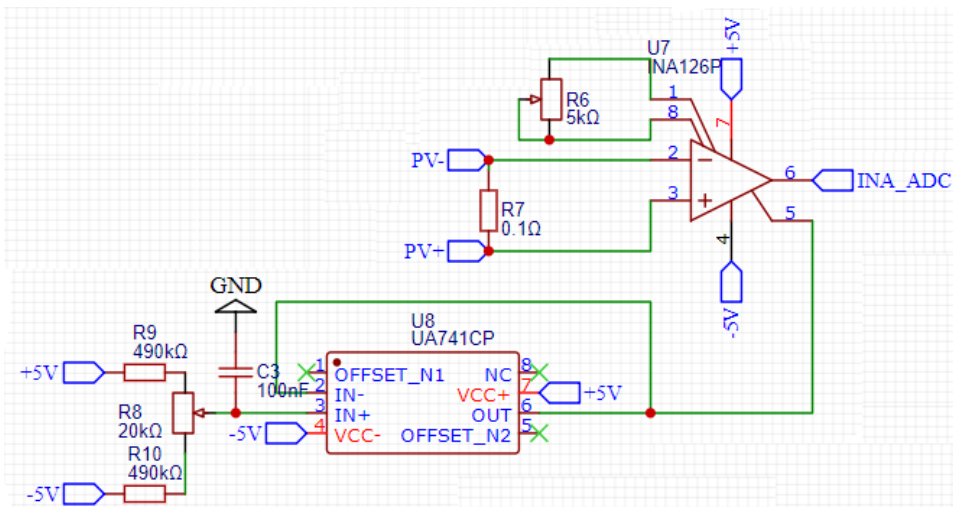


Figura 2.2 - Circuito final do sensor de corrente

Seguidamente, dimensionou-se o sensor de tensão. Sabemos que a tensão máxima fornecida pelos PV, V_{oc} , é de 22.6V e pretendemos que a representação dessa tensão seja 3V para poder ser lida pelo ADC.

Com efeito, foi dimensionado um divisor resistivo tal que cumprisse os requisitos acima, segundo os seguintes cálculos:

$$V_{OUT} = \frac{R_2}{R_1 + R_2} \times V_{IN} \quad (9)$$

Para $V_{OUT} = 3V$ e $V_{IN} = 22.6$, a relação das resistências dá-se pela seguinte expressão:

$$R_2 = 6.53 \times R_1 \quad (10)$$

As resistências escolhidas têm de ter um valor óhmico elevado pois, para medir um sinal, a impedância tem de ser alta. Para tal, para um potenciômetro (R_1) = 100kΩ, R_2 tem o valor de 650kΩ. Ajustando o potenciômetro R_1 , conseguimos obter $V_{OUT} = 3V$.

O circuito final apresenta-se como na figura 2.3:

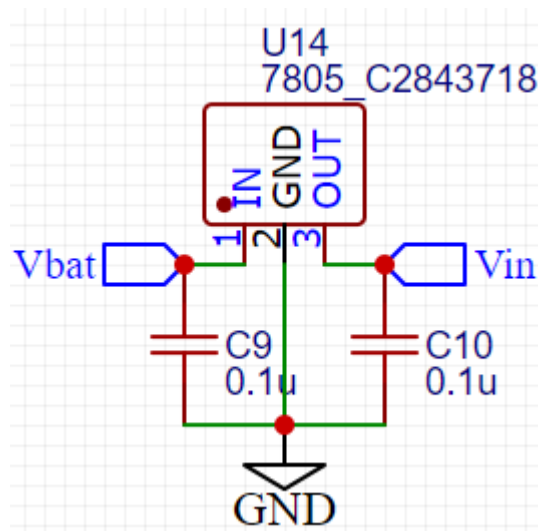


Figura 2.3 - Circuito final do sensor de tensão

Adicionalmente, para acrescentar ao circuito do INA126P e do LM741, nomeadamente na alimentação, foi utilizado o integrado LMC7660, que permite converter uma tensão de entrada, no seu simétrico na saída, isto é, converte $V_{IN} = 5V$, $V_{OUT} = -5V$. O circuito foi desenhado conforme recomendado pelo fabricante no respetivo *datasheet*.

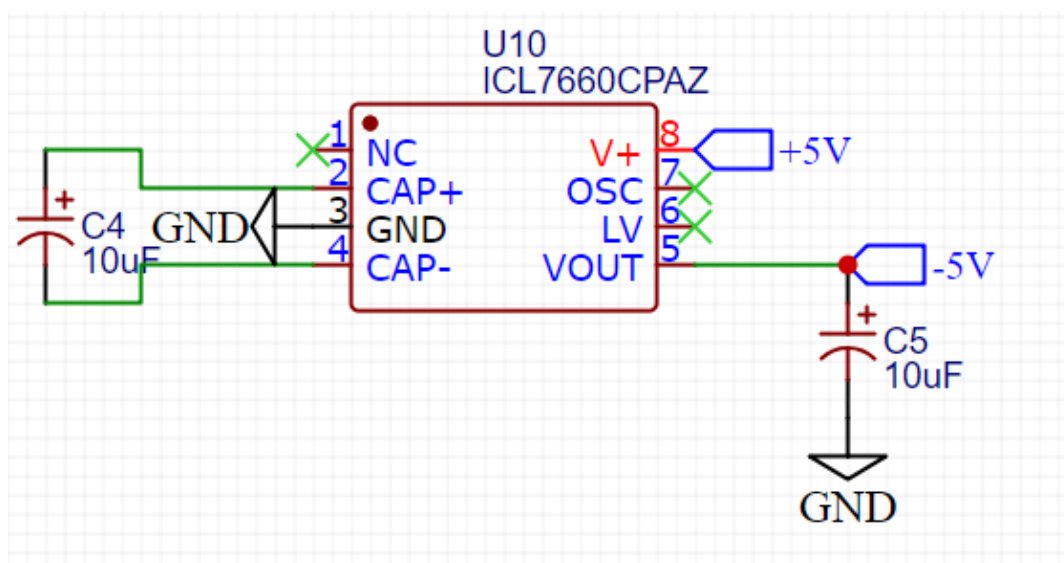


Figura 2.4 - Conversor de tensão para -5V

2.2 - Controlo dos motores:

O módulo responsável para o controlo é constituído por vários sensores. São estes: 3 sensores de luminosidade LDRs divididos por uma barreira, 2 sensores fim de curso, 1 potenciómetro multivoltas, 2 motores de passo e 2 drivers de motor de passo DRV8825.

Os sensores de luminosidade estão ligados a um divisor resistivo de forma que a variação da resistência provoque uma variação na tensão, que será posteriormente

lida pelo microcontrolador. A diferença do valor resistivo entre 2 LDRs, obtida por dois canais do ADC presente no microcontrolador STM32, faz o acionamento do respetivo motor.

Para que o motor que se encontra na parte inferior, que representa o eixo azimute, não dê mais que uma volta, são utilizados dois sensores de fim de curso (um para cada sentido), ligados em modo *Pull-up*, de forma a serem lidos a nível lógico '0' quando estimulados.

Já para o motor que se encontra na parte superior, que representa o eixo altitude, os limites da rotação são impostos por software. Para esse fim é utilizado um potenciômetro multivoltas como forma de medição da posição dos painéis. Assim, para além de determinar os limites de movimento, também é possível estabelecer uma posição de referência para que, sempre que os sensores de luminosidade não atingirem um certo valor de intensidade luminosa, o painel fique nessa mesma posição (correspondente ao período noturno). O motor de passo utilizado na parte superior possui caixa redutora, uma vez que este necessita um maior torque, de forma a conseguir aguentar o peso da parte superior (onde se encontram os painéis).

Ambos os motores são controlados por 2 drivers DRV8255, que acionam o respetivo motor de acordo com o sinal PWM enviado pelo microcontrolador.

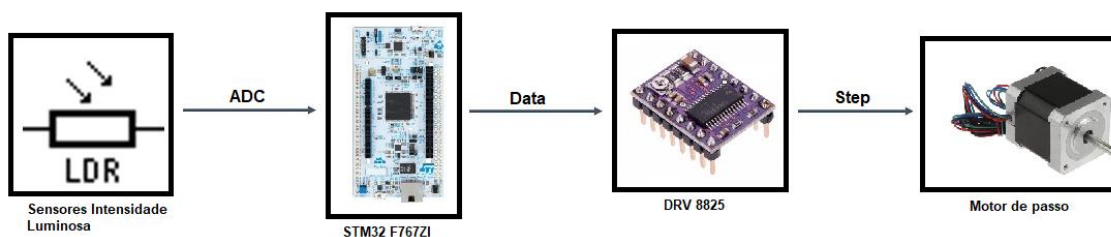


Figura 2.5 - Diagrama de blocos do acionamento dos motores

2.3 - Interface PV - ESS

De forma a alimentar o circuito de controlo, aproveitando a energia produzida pelos painéis fotovoltaicos, foi desenvolvido um circuito de alimentação constituído por três módulos: um conversor de corrente contínua *step-up* onde é implementado o algoritmo MPPT, um conversor corrente contínua *step-down* e um sistema de armazenamento de energia controlado.

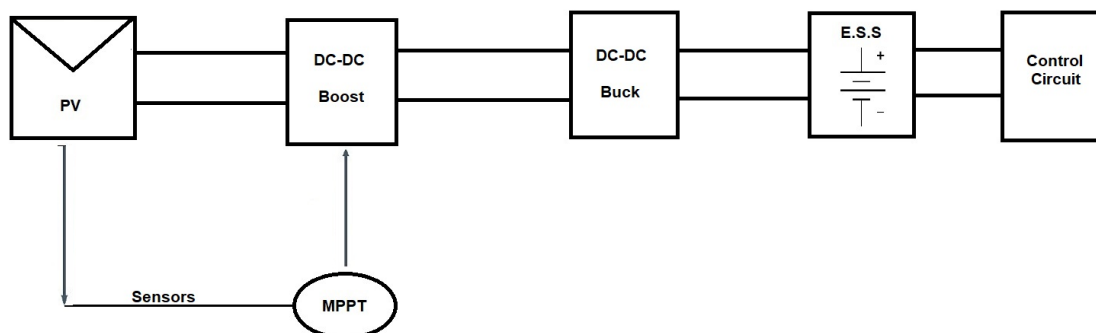


Figura 2.6 - Diagrama de blocos da interface PV-E.S.S

A decisão de utilizar essa constituição de conversores deve-se ao facto de que o carregamento das baterias ser dado por uma tensão inferior à tensão máxima fornecida pelos painéis. No entanto, utilizar um conversor *Buck* nas saídas dos painéis iria afetar o estudo do desempenho dos mesmos, pois a corrente seria reduzida e os valores medidos não seriam relativos à capacidade máxima dos painéis fotovoltaicos. Para isso, optou por se utilizar um conversor *Boost* aos terminais dos painéis e estabelecer uma tensão de referência mais alta e estável, sem que haja perdas na produção. Assim, o *Buck* faria então a regulação de tensão para o carregamento das baterias.

Para o conversor DC-DC *step-up* (*Boost*), desenvolveu-se um circuito próprio. O dimensionamento dos componentes foi feito de acordo com os parâmetros previamente medidos das saídas dos 4 painéis fotovoltaicos em paralelo, bem como frequência de comutação.

Frequency	18000	This is the boost converter frequency. For microcontrollers its often the CPU clock / 256
	Hz	
Min Vin	10	The lowest expected input voltage
	V	
Max Vin	22.6	The highest expected input voltage
	V	
Min Vout	24	The lowest desired output voltage
	V	
Max Vout	33	The highest desired output voltage
	V	
Iout	0.37	Output current draw
	Amps	
Vripple	0.063	Maximum allowable voltage ripple
	V	

Calculate!		
Min. Duty Cycle	$D_{min} = 1 - (V_{imax}/V_{omin})$	5.83 %
Max. duty cycle	$D_{max} = 1 - (V_{imin}/V_{omax})$	69.1 %
Min. Inductor size	$L > D * V_{in} * (1-D) / (freq * 2 * I_{out})$	366.19 μ H
Peak inductor current	$I_{pk} = (V_{inmax} * D) / (f * L)$	1.0805 A
Minimum capacitor	$Cap > I_{out} / (V_{ripple} * freq)$	326.27 μ F
Minimum Schottky diode	$V_{breakdown} \geq V_{outmax} \ \& \ I_{diode} \geq I_{pk}$	33 V 1.0E A

Figura 2.7 - Valores calculados do conversor CC-CC boost (4)

Com os parâmetros gerados, foi feita a escolha de componentes de acordo com a disponibilidade e o que se adequaria melhor ao projeto com segurança.

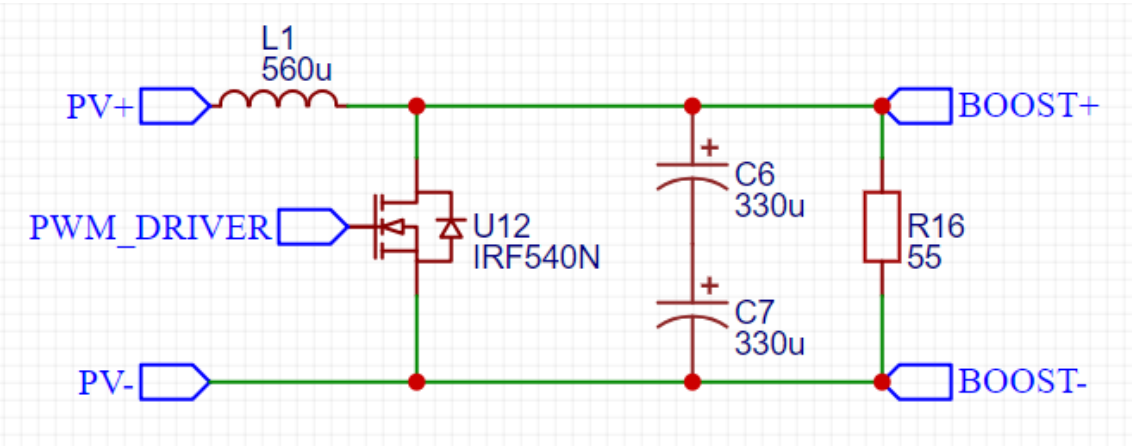


Figura 2.8 - Circuito do conversor CC-CC Boost

Para funcionar como interruptor do circuito, foi utilizado o MOSFET IRF540N. O sinal de PWM fornecido pelo microcontrolador utilizado oferece uma tensão máxima de 3.3V, como a tensão mínima da gate do MOSFET é na ordem dos 4V, foi necessário utilizar um driver para a gate. Para esse fim, foi escolhido um opto acoplador 4N25 para amplificar o PWM de saída do microcontrolador para 15V de forma isolada para que não se danifique o equipamento.

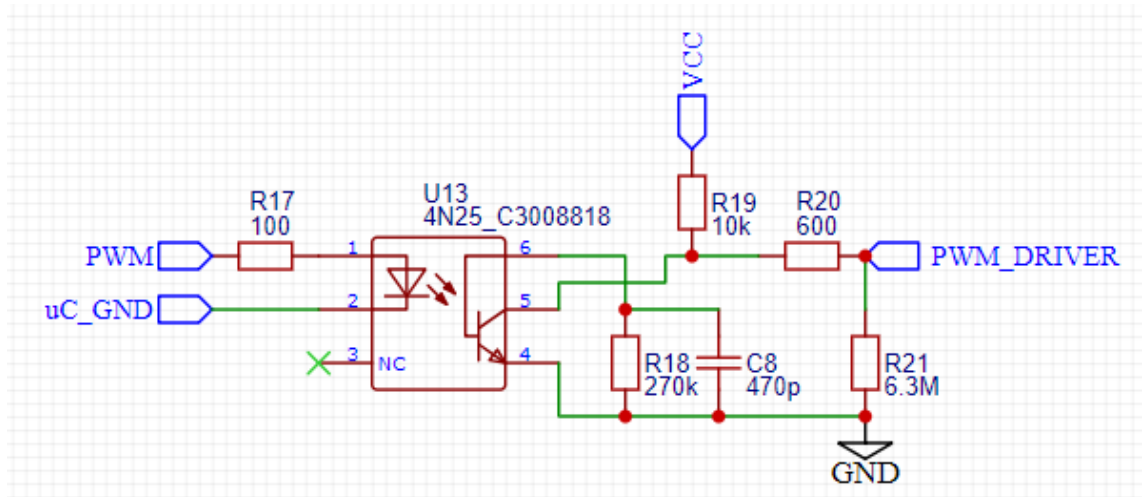


Figura 2.9 - Circuito com o MOSFET e opto acoplador

Para o conversor DC-DC Buck foi adquirido um módulo Buck XL4015 (5) que permite a conversão DC-DC com saída fixa por controlo com feedback. O módulo suporta entradas de até 38V e saídas até 35V 5A, suficiente para o carregamento das baterias. Por meio de um potenciômetro presente no circuito, a saída é regulada para limite seguro de carregamento e independente da tensão fornecida pelos painéis, assim as baterias irão carregar e descarregar de forma segura.



Figura 2.10 - Conversor CC-CC Buck

Para as baterias, foram utilizadas 2 células 18650 com 2500mAh e 3.7V ligadas em série que fornecem uma tensão máxima, quando totalmente carregadas, de 8.2V. Ainda é ligado às cargas um circuito BMS (6) que fará a proteção de carga e descarga das baterias, o que irá permitir que as baterias estejam sempre conectadas aos conversores sem que haja problemas de sobrecarga, além de evitar dano contra picos de tensão ou corrente.



Figura 2.11 - Battery Management System (BMS)

Como uma última camada de proteção, os terminais da bateria estão ligados a um regulador de tensão 7805 (7), capaz de regular os 8.2V fornecidos pela bateria para 5V. Isso é feito para que não haja problemas de alimentação da STM32-F767ZI, dado que a alimentação externa é feita nominalmente por 5V.

3 - Arquitetura de software:

O software desempenha um papel fundamental neste sistema, permitindo o acompanhamento preciso do movimento do sol ao longo do dia, bem como a extração da potência máxima dos painéis fotovoltaicos. Ao direcionar painéis fotovoltaicos na direção correta, bem como a utilização de um controle MPPT, o sistema, aumenta significativamente a eficiência energética. Além disso, determinadas grandezas físicas como a temperatura, a humidade, até a própria inclinação dos painéis são importantes para a eficiência energética.

A figura 3.1 corresponde ao diagrama de blocos do sistema implementado:

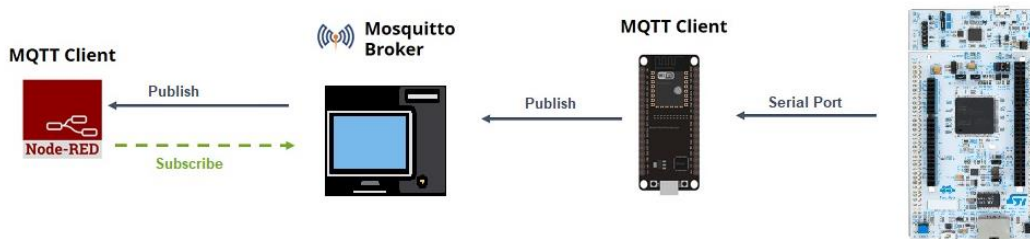


Figura 3.1 - Diagrama de blocos do sistema desenvolvido

Neste capítulo, serão abordados os programas desenvolvidos na STM 32 (acionamento dos motores a partir dos LDRs, obtenção dos valores dos sensores e algoritmo MPPT) e na ESP 32 (servidor MQTT e módulo Wi-Fi).

Nota: Segue em anexo um documento que explica como foi efetuada a configuração dos periféricos da STM.

3.1 - Acionamento dos motores a partir dos LDRs:

No que diz respeito à codificação para o acionamento dos motores de passo foi configurado o Timer 2 para gerar uma interrupção a cada 100ms e na sua respetiva callback é apenas ativa uma flag (adc_flag) para posteriormente ser efetuado o processamento onde serão lidos os valores dos LDRs. Já na main, quando é verificada a ativação da flag, é chamada uma função (ADC_Read_Channel) responsável por ler os valores de cada sensor em diferentes canais.

Assim que concluída a leitura, é feita a comparação dos valores de dois sensores luminosos na função (ldrComparison) que posteriormente comandará um motor, de seguida é chamada novamente a mesma função. Porém, desta vez, é feita a comparação entre o valor de um LDR já lido e o que ainda não foi lido para comandar o outro motor.

Na comparação foi definido um valor de sensibilidade para que haja uma histerese. Quando a diferença entre os 2 valores medidos for maior que o valor

estabelecido para a sensibilidade, o sinal de PWM será ativado de forma a acionar o motor e o sentido de rotação será definido pelo sinal proveniente do resultado da diferença.

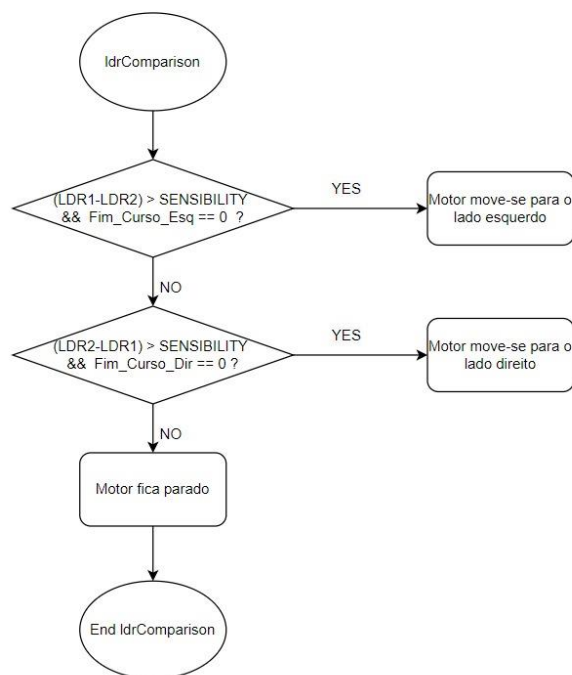


Figura 3.2 - Fluxograma da função de comparação dos LDRs

Em paralelo são verificados os estados dos sensores de fim de curso. Caso estejam acionados, a rotação naquele sentido é interrompida. Como o motor do eixo superior não possui fins de curso, o controlo da posição é feito pelo potenciômetro.

No final do dia, quando todos os sensores se encontrarem abaixo de um valor de referência imposto, os motores serão controlados de modo a posicionar os painéis na posição de referência (paralelo em relação a um eixo horizontal) para que no nascer do próximo dia os sensores de luminosidade sejam capazes de detetar os primeiros raios de sol.

3.2 - Algoritmo MPPT (Perturbação e Observação)

Como foi abordado anteriormente, os painéis fotovoltaicos possuem um baixo rendimento. De forma a extrair a máxima potência, além de os colocar de forma perpendicular à radiação solar, foi implementado um algoritmo MPPT.

O MPPT (8), como o nome indica, é um algoritmo muito utilizado quando se trata da otimização de extração de potência de um painel solar. Em aplicações de potência, este algoritmo encontra-se facilmente em inversores solares. Este permite ajustar a tensão e a corrente fornecidas pelos painéis solares em qualquer condição de operação, para garantir que a máxima quantidade de energia é extraída. Ao otimizar a operação no ponto de máxima potência maximiza a eficiência do sistema de energia solar. De entre as diversas técnicas de implementação do MPPT, optou-se por utilizar a técnica de Perturbação e Observação (P&O) que se destaca pela sua fácil implementação e

eficácia. Esta técnica tem por base a alteração dos tempos de comutação do interruptor analógico, para extrair mais ou menos tensão ou corrente.

Como referido anteriormente no estudo dos sensores, foram dimensionados dois tipos: um de corrente e um de tensão. Recorreu-se a este dimensionamento pois é necessária uma boa definição e uma mínima margem de erro para que a leitura dos valores e posterior atuação sejam de acordo com o previsto.

Inicialmente, com recurso da STM32, são obtidos os valores de tensão e corrente e, posteriormente, calculada a potência. Depois de obtidas essas variáveis, são comparadas com os valores obtidos anteriormente (na última amostragem).

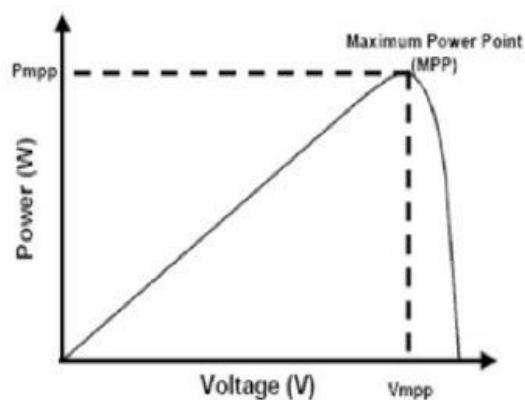


Figura 3.3 - Curva potência - tensão do PV (13)

O algoritmo MPPT implementado é representado no fluxograma seguinte:

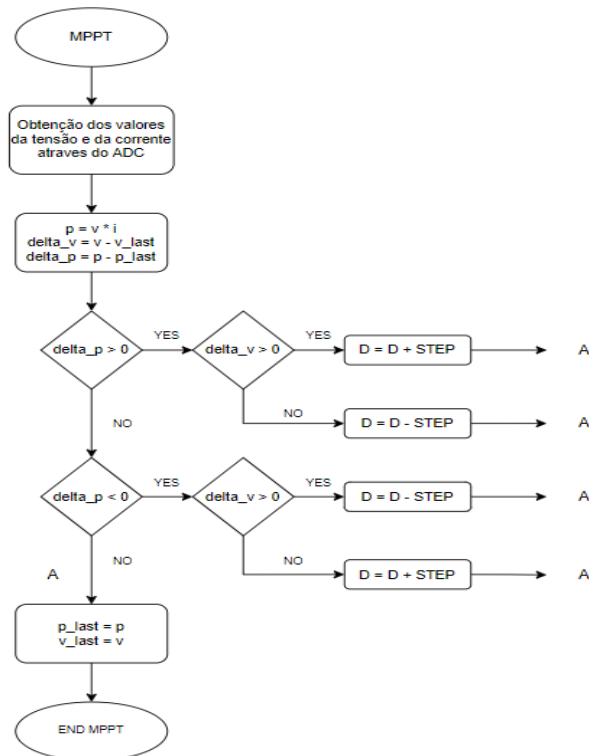


Figura 3.4 - Fluxograma do algoritmo MPPT

3.3 - Servidor MQTT e módulo Wi-Fi a partir da ESP32:

Como forma de analisar os dados recolhidos pelos sensores e assim conseguir perceber qual a melhor disposição dos painéis, decidiu-se recorrer a um software capaz de criar uma interface com esses dados. Foi utilizado o software de programação visual *Node-Red*, capaz de ligar dispositivos de hardware, APIs e serviços online de diversas formas. Para este projeto, decidiu-se usar uma *ESP 32 DevKit V4* como módulo *Wi-Fi* (utilizou-se o *Arduino IDE* para a programar) para comunicar com este software através do protocolo MQTT. O MQTT (*Message Queuing Telemetry Transport*)(9) é um protocolo de mensagens para redes restritas de baixa largura de banda e dispositivos IoT de latência extremamente elevada. Uma vez que o Message Queuing Telemetry Transport é especializado para ambientes de baixa largura de banda e alta latência, é um protocolo ideal para a comunicação máquina-a-máquina (M2M). O MQTT funciona segundo o princípio do *publisher/subscriber* e é operado através de um *broker* central (corretor ou intermediário), isto significa que o *publisher* (emissor de dados) e o *subscriber* (recetor dos dados) não têm uma ligação direta. As fontes de dados comunicam os seus dados através de uma publicação e todos os destinatários com interesse em determinadas mensagens ("marcadas pelo tópico") recebem os dados entregues porque se registaram como assinantes.

Para a *ESP 32* conectar-se ao servidor broker, é necessário, numa primeira instância, que esta se conecte à mesma rede *Wi-Fi* e ao IP do mesmo.

Para isso, são necessárias incluir algumas bibliotecas específicas ("WiFi.h" e "PubSubClient.h").

De forma a conectar a ESP à rede *WiFi*, utiliza-se a função "Wifi.begin" e nos parâmetros da mesma coloca-se o *SSID* e a *password* da rede a que se pretende conectar. Enquanto essa ligação não for efetuada, o programa não avança, ficando preso num *loop* (a função *WiFi.status* é responsável por fazer essa verificação como parâmetro num ciclo *while*).

Após a conexão à rede *Wi-Fi*, é efetuada a conexão ao servidor *broker* através do protocolo *MQTT*. Para isso, recorre-se à função "Client.setServer" que recebe de parâmetros o IP do servidor bem como a porta onde será conectado. Tal como na conexão à rede *Wi-Fi*, também na conexão ao servidor é feito um *loop* para conectar, impedindo que o programa avance sem a conexão ser efetuada. Para esta conexão e, à semelhança da conexão *Wi-Fi*, a função responsável por verificar o estado da mesma é a função "cliente.connected".

Quando as conexões forem efetuadas, o programa começa a processar os dados recebidos por porta série, provenientes da STM. A ESP recebe uma string onde tem os dados dos vários sensores (separados por um espaço) e, no último dado um "\n". Primeiro, é chamada a função "read_line" que guarda numa variável do tipo *char**, a string recebida por porta série. Depois disso, é chamada a função "decode_vars" que separa a string recebida nas respetivas variáveis, com recurso à função "strtok" (da biblioteca *stdlib*) e armazena os valores separados num *array* de strings.

Para finalizar o *loop*, publica-se cada variável no respetivo tópico com a função "client.publish" e chama-se a função "client.loop" para que o *client* (ESP) seja capaz de processar a mensagem e restabelecer a ligação ao servidor.

A figura 3.5 corresponde ao fluxograma principal do programa implementado na ESP:



Figura 3.5 - Fluxograma do algoritmo desenvolvido na ESP 32

O Node-Red recebe as variáveis enviadas pela ESP (para isso necessita de subscrever aos respetivos tópicos no *broker*) e apresenta-as numa *dashboard* (que se acede em qualquer browser em “ip_do_broker”:1880/ui). Para ser possível a visualização do HMI, tanto o *broker* quanto o Node-Red e a ESP devem estar conectados à mesma rede *Wi-Fi*.

4 - Conclusão e trabalho futuro:

Desde o momento que nos propusemos a desenvolver este projeto tínhamos a percepção dos problemas e dificuldades que a realização do mesmo envolveria e para tal seria necessário dedicar tempo e esforço para que fosse possível obtermos o resultado pretendido.

Durante todo o projeto foram postas à prova algumas competências adquiridas ao longo do curso. Deste modo, aprofundamos conhecimentos já adquiridos de unidades curriculares anteriormente lecionadas, desde Sistemas Embebidos e Energias Renováveis e Mobilidade Elétrica, passando por Máquinas Elétricas, entre outras. Foi um projeto ímpar comparativamente a projetos já realizados em anos anteriores porque envolveu um maior conhecimento de várias áreas e requereu muito trabalho autónomo.

Ao longo do desenvolvimento do projeto, encontramos inúmeras adversidades tanto no desenvolvimento do design como na elaboração e testes dos circuitos elétricos que compunham o STS. Até ao resultado final, todos os circuitos passaram por um processo de simulação, montagem e teste até serem obtidos os resultados projetados nas simulações.

Uma das aprendizagens mais importantes a reter da execução deste projeto e que, por vezes, não recebe a devida importância, está relacionada com a comunicação e o crescimento de cada um como pessoa e futuros engenheiros, bem como o desenvolvimento de um ambiente de trabalho saudável.

Todo este trabalho realizado não pode ser visto como uma ideia final e sim uma forte base para um projeto mais elaborado no futuro, já sem as limitações do mundo analógico.

Para desenvolvimento futuro, seria de considerar implementar uma base de dados para o HMI, de forma a guardar os dados recolhidos pelos sensores. Além disso, seria plausível desenvolver um sistema capaz de detetar problemas ou falhas associados à descida do rendimento dos paines (limpeza e manutenção).

Em suma, este projeto foi uma mais valia para o conhecimento e o desenvolvimento de competências importantes para nós como futuros engenheiros, tanto técnicas quanto trabalho em grupo. Os resultados atingidos apresentam um nível bastante bom, revelador do empenho e do trabalho conjunto de todos os integrantes do grupo.

5 - Referências:

1 – Datasheet painéis fotovoltaicos

URL: <https://www.tme.eu/Document/bab057a8a844396ef052688645558ec2/CL-SM5P.pdf>

2 – Datasheet motores de passo

URL: <https://www.omc-stepperonline.com/download/17HS19-0406S-PG14.pdf>

3 – Datasheet DHT11

URL: https://image.dfrobot.com/image/data/DFR0067/DFR0067_DS_10_en.pdf

4 – Boost Converter Calculator

URL: <https://learn.adafruit.com/diy-boost-calc/the-calculator>

5 – Conversor CC-CC buck

URL: <https://www.botnroll.com/pt/conversores-dcdc/2998-step-down-dc-dc-1-25-35vdc-c-display-5a-75w.html>

6 – Features and Specifications BMS

URL: <https://www.botnroll.com/pt/carregadores/3119-bms-para-proteccao-baterias-18650-2s-74v-4a.html>

7 – Voltage Regulator

URL: <https://www.electronicshub.org/understanding-7805-ic-voltage-regulator/>

8 – What is MPPT?

URL: <https://www.solar-electric.com/learning-center/mppt-solar-charge-controllers.html/>

9 – What is MQTT? – OPC Router

URL: <https://www.opc-router.com/what-is-mqtt/>

10 – Slides de Sistemas de Automação

11– ESP 32 / STM 32 Power Supply System

URL:<https://www.elektormagazine.com/labs/esp32stm32-power-supply-system>

12 - ESP 32 MQTT Publish/Subscribe with Arduino IDE

URL:<https://randomnerdtutorials.com/esp32-mqtt-publish-subscribe-arduino-ide/>

13 - Curva potência-tensão de um PV

URL:https://paginas.fe.up.pt/~ee03195/Carro_Solar/PaineisCurvasdeFuncionamento.html