



Universidade do Minho
Escola de Engenharia

Laboratórios de Telecomunicações e Informática II

ENGENHARIA DE TELECOMUNICAÇÕES E INFORMÁTICA
2020/2021

(Docentes: Bruno Daniel Mestre Viana Ribeiro, Vadym Serhiyovych Hapanchak,
José Augusto Afonso, Sérgio Adriano Fernandes Lopes)

12 de março de 2021

Relatório de especificação

FASE A

Rui Filipe Ribeiro Freitas - a84121@alunos.uminho.pt

Sandro Teixeira Ribeiro – a85316@alunos.uminho.pt

Tiago João Pereira Ferreira - a85392@alunos.uminho.pt

Índice

Índice de figuras	4
Índice de tabelas.....	5
Lista de abreviaturas	6
Introdução	7
1. Planeamento do projeto	8
1.1. Planeamento temporal	8
1.2. Tecnologias/Ferramentas necessárias	9
1.2.1. Ao nível do <i>hardware</i>	9
1.2.2. Ao nível do <i>software</i>	10
2. Fundamentos.....	11
2.1. Sistema sensor	11
2.1.1. Sensor de movimento.....	11
2.1.2. Sensor de luminosidade.....	12
2.1.3. LED	13
2.2. Placa ESP32.....	14
2.3. Concentrador	14
3. Desenvolvimento.....	15
3.1. Arquitetura do sistema	15
3.1.1. Arquitetura de <i>hardware</i>	15
3.1.2. Arquitetura de <i>software</i>	16
Referências	17

Índice de figuras

Figura 1 - Planeamento temporal tabela.....	8
Figura 2 - Planeamento temporal gráfico.....	8
Figura 3 - Esquema PIR HC-SR501.	11
Figura 4 - LDR (Sensor de luminosidade).....	12
Figura 5 - Esquema divisão de tensão.	12
Figura 6 - Circuito LED.....	13
Figura 7 - Esquema placa ESP32.	14
Figura 8 - Circuito do esquema geral.....	15
Figura 9 - Circuito Sistema Sensor.	15
Figura 10 - Trama START.....	16
Figura 11 - Trama STOP.	16
Figura 12 - Trama DATA.....	16
Figura 13 - Trama ERROR.....	16

Índice de tabelas

Tabela 1 - Tecnologias ao nível do hardware.	9
Tabela 2 - Tecnologias ao nível do software.	10

Lista de abreviaturas

LTI II - Laboratórios de Telecomunicações e Informática II

MQTT - *Message Queuing Telemetry Transport*

LED – Light-Emitting Diode

LDR - Light Dependent Resistor

IDE - Integrated Development Environment

CSV – Comma-Separated Values

PS – Porta *Socket*

FA – Frequência de amostragem

TC – Tipo de comunicação

IESP – Identificador da placa ESP

TA – *Timestamp* das amostras

ESM – Estado do sensor de movimento

ESL – Estado do sensor de luminosidade

EL – Estado da lâmpada

Introdução

No âmbito da Unidade Curricular de LTI II (Laboratórios de Telecomunicações e Informática II) foi-nos proposto o desenvolvimento de um sistema de iluminação inteligente que permita monitorizar a ocupação e a luminosidade em diferentes áreas de uma residência, bem como controlar a luminosidade desses ambientes através do acionamento de lâmpadas.

Este projeto está dividido em 3 Fases e uma fase extra onde a fase A passa pelo planeamento, desenvolvimento e teste de todo o *hardware* e *software* que compõem o sistema geral necessários para a implementação dos sensores reais e a sua comunicação com os concentradores. Na fase B o objetivo é implementar um sistema central básico e o protocolo de comunicação entre este sistema e os concentradores. Em relação à fase C, muito resumidamente passa por completar o sistema central elaborado na fase B através de uma base de dados relacional e acrescentar funcionalidades de monitorização e administração que possamos achar necessárias. Na fase extra do projeto, que se trata de uma fase opcional é nos sugerida a implementação de funcionalidades mais avançadas através do uso do MQTT (*Message Queuing Telemetry Transport*). Cada fase do projeto terá uma entrega de um relatório de especificação passada uma semana do início dessa fase e uma demonstração acompanhada de um relatório de demonstração.

De modo a sermos capazes de cumprir com os objetivos deste projeto semestral devemos pôr em prática conhecimentos adquiridos noutras unidades curriculares, nomeadamente Redes de Computadores, Sistemas Operativos, Sistemas Distribuídos, Eletrónica e Laboratórios de Telecomunicações e Informática I.

1. Planeamento do projeto

1.1. Planeamento temporal

De modo a que o grupo se mantenha focado no trabalho e com um compromisso para cumprir horários resolvemos planear as tarefas a fazer nesta fase. Na figura 1 observamos em forma de tabela os vários assuntos a ser tratados nesta fase com um período dado por nós para cumprir. Em relação à figura 2 demonstramos em forma de gráfico o tempo despendido nas várias tarefas. Ambas as figuras foram retiradas do programa *Gantt*.


		
Nome	Data de início	Data de fim
• Fase A	22-02-2021	07-04-2021
• Planeamento Temporal	22-02-2021	26-02-2021
• Definição da Arquitetura do Sistema	01-03-2021	01-03-2021
• Início da elaboração do relatório	02-03-2021	02-03-2021
• Ligações físicas entre o esp e sensores	04-03-2021	04-03-2021
• Desenvolvimento código esp	05-03-2021	15-03-2021
• Desenvolvimento código concentrador	18-03-2021	29-03-2021
• Realização de Testes	01-04-2021	02-04-2021
• Discussão dos resultados obtidos	05-04-2021	05-04-2021
• Conclusão e revisão do relatório	07-04-2021	07-04-2021

Figura 1 - Planeamento temporal tabela.

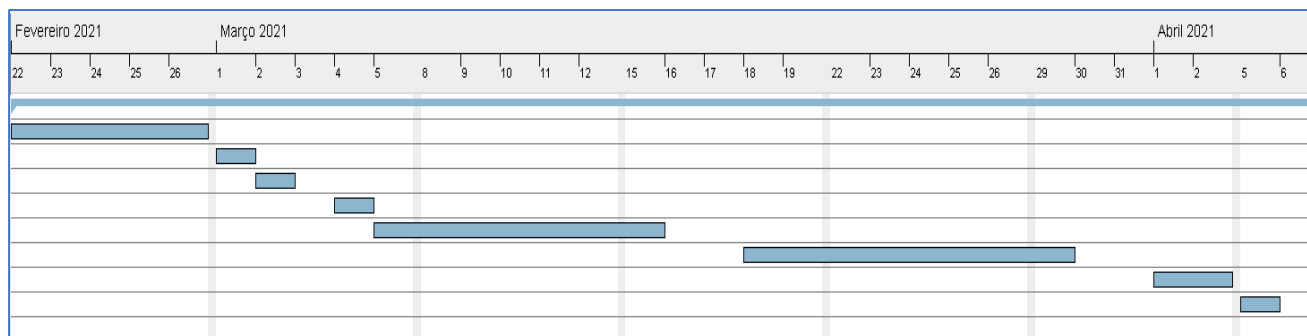


Figura 2 - Planeamento temporal gráfico.

1.2. Tecnologias/Ferramentas necessárias

Em qualquer projeto são necessárias certas tecnologias/ferramentas que nos facilitem o trabalho e ao mesmo tempo que aumentem a nossa produtividade. Para este trabalho em específico teremos de recorrer a tecnologias específicas tanto ao nível do *hardware* como ao nível do *software*, de modo a cumprir com os objetivos propostos.

1.2.1. Ao nível do *hardware*

No que toca à parte do *hardware* deste nosso trabalho as tecnologias por nós utilizadas estão presentes na tabela 1, sendo que estas podem vir a ser alteradas no futuro.









Tabela 1 - Tecnologias ao nível do *hardware*.

FERRAMENTA	QUANTIDADE	UTILIZAÇÃO
Computador Portátil	1	Desenvolvimento de código, elaboração do relatório, pesquisa e testes
Placa ESP32	1	Comunicação com os sensores e os LED's
Sensor de movimento (PIR HC-SR501)	1	Deteção de movimento
Sensor de luminosidade (LDR 5...10kOhm)	1	Medição da luminosidade
LED	3	Simulação de lâmpadas
Cabo Micro USB	1	Ligações físicas entre o PC e a placa ESP32
Fios de ligação	Vários	Efetuar as ligações entre as placas ESP32 e os sensores e LED's
Resistência	Várias	Controlo de fluxo

1.2.2. Ao nível do *software*

Em relação ao *software* que irá ser utilizado na execução do nosso projeto este passará maioritariamente pelas aplicações apresentadas na tabela 2. Nesta realçamos as mais importantes e fundamentais, sabendo que, no entanto, poderão sofrer alterações no futuro dado ainda nos encontrarmos numa fase mais introdutória do trabalho.

Tabela 2 - Tecnologias ao nível do *software*.

LOGO	DESCRIÇÃO	UTILIZAÇÃO
	Microsoft Word	Realização e sincronização do relatório do projeto
	GanttProject	Planeamento do projeto através de diagramas de Gantt
	Arduino IDE	Software para a conexão com a placa ESP32 e comunicação com o mesmo
	Visual Studio Code	Execução e compilação de código
	GitHub	Sincronização do código da aplicação
	Visual Paradigm	Implementação de fluxogramas para a elaboração da aplicação e do projeto
	Google Drive	Sincronização de ficheiros essenciais à realização do trabalho
	Discord	Comunicação entre o grupo

2. Fundamentos

2.1. Sistema sensor

O sistema sensor que temos de implementar na elaboração deste projeto é constituído por uma placa ESP32 que comunica através de Wi-Fi ou Bluetooth com um concentrador. Esta ligação é importante para a aquisição de dados por parte do concentrador através de uma conexão sem fios fiável. Ligados ao dispositivo de RF estão conectados sensores de movimento e de luminosidade e LED's através de cabos devidamente conectados. Todos os elementos que constituem o sistema sensor serão explicados de seguida.

2.1.1. Sensor de movimento

O sensor de movimento utilizado pelo nosso grupo foi o recomendado pelos docentes, o PIR HC-SR501. Este sensor tem como principal função a deteção de movimento. Na placa é possível alterar os valores de tempo e de sensibilidade do sensor através de dois potenciômetros. O valor de tempo altera o tempo que passa até o output passar de HIGH para LOW após a deteção de movimento. O mínimo é 3 segundos após deteção de movimento pelo que este sensor não é o mais eficaz se quisermos um sensor que funcione em tempo real. Em relação à sensibilidade este serve para aumentar ou diminuir a distância de deteção, que varia entre os 3 e os 7 metros. [1]

Na figura 3 podemos observar o *pinout* do sensor de movimento, onde observamos os potenciômetros que nos permitem ajustar os parâmetros enunciados anteriormente assim como 2 formas de disparo, único ou repetido, em que no modo único o sensor mantém-se HIGH até deixar de detetar movimento e no modo repetido em que o sensor mantém-se HIGH até o tempo definido pelo potenciômetro que ajusta o tempo ser atingido. Na elaboração do nosso projeto iremos utilizar o modo de disparo repetido. Para além destes parâmetros observamos na imagem 3 pinos que vamos ligar à placa ESP32, o 3.3 V de modo a fornecer energia ao sensor, o SIGNAL que liga à placa de modo a que possamos controlar o sensor e o GND.

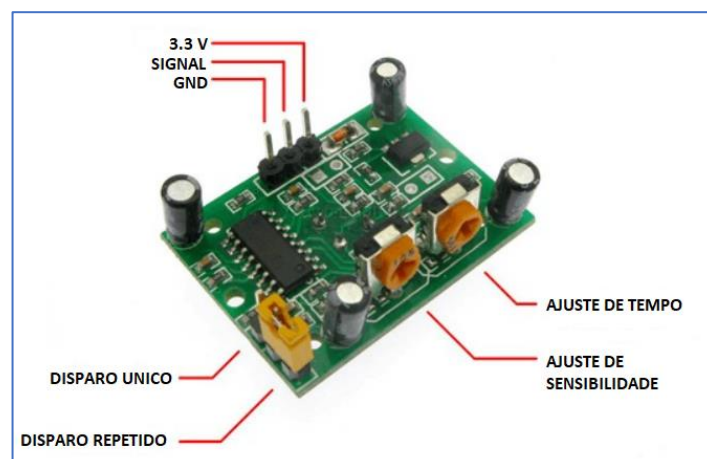


Figura 3 - Esquema PIR HC-SR501.

2.1.2. Sensor de luminosidade

O sensor de luminosidade escolhido pelo grupo foi o LDR (Light Dependent Resistor) com resistência na luz de 5-10k Ω . Um dos materiais presentes é um semicondutor de alta resistência e, como mostra a figura 4, possui apenas dois terminais e não tem polaridade.

Este sensor é uma resistência, a qual varia conforme a intensidade de luz que incide sobre ele. Quando temos pouca luminosidade a resistência terá um valor elevado e quanto maior for a luminosidade menor será a resistência. O sensor LDR pode apresentar um atraso até 10 milissegundos para detectar a mudança entre pouca e muito luminosidade, ou seja, entre um ambiente escuro e um ambiente claro. [2]

Na montagem do hardware foi necessária a utilização de uma resistência de 1k Ω junto ao sensor LDR para preservar o bom funcionamento deste. Na figura 5 podemos observar o esquema a que nos baseamos para obter o valor da resistência em que o valor apresentado no pino 32 da placa ESP corresponde à tensão entre a resistência e o LDR.

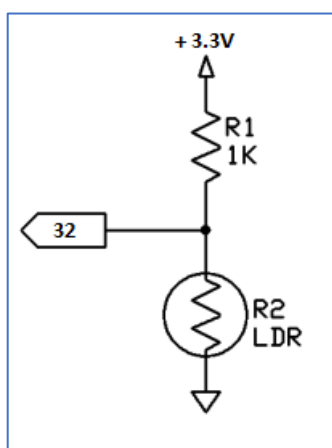


Figura 5 - Esquema divisão de tensão.

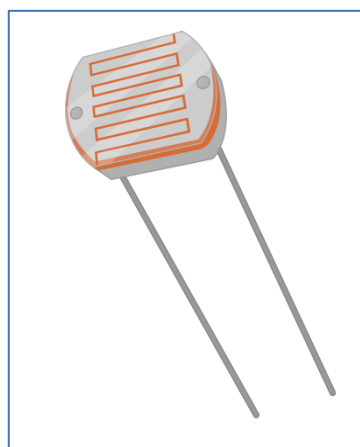


Figura 4 - LDR (Sensor de luminosidade).

2.1.3. LED

Na elaboração deste projeto temos por objetivo utilizar 3 LED's, no entanto como ainda estamos numa fase inicial poderemos vir a utilizar mais ou menos. A nossa perspectiva é utilizar um LED para servir como lâmpada em que podemos acender sempre que for conveniente, outro é para caso o sensor de movimento detete movimento o LED acende e apaga caso não seja detetado mais movimento. E o último para efeitos de iluminação caso o sensor de luminosidade atinja valores perto de 0 indicando que a área se encontra em escuridão. De modo a acender o LED é necessário que por ele passe corrente. No entanto é necessário controlar a quantidade de corrente e para isso utilizamos uma resistência entre o LED e a fonte de alimentação. O valor desta resistência foi calculado através da equação 1.1. Através da observação da figura 6 e da equação retiramos vários valores dependendo dos LED's usados pois a tensão do LED varia com a cor. Pelas contas efetuadas pelo grupo a resistência que devíamos utilizar andaria na casa dos 50-250 Ohms. Como o recurso a resistências era limitado decidimos escolher uma resistência de 470 Ohms das disponíveis visto acharmos não haver grande problema ter uma resistência um pouco superior.

$$Resistance = \frac{Supply\ Voltage - Forward\ Voltage}{Current} \quad (1.1)$$

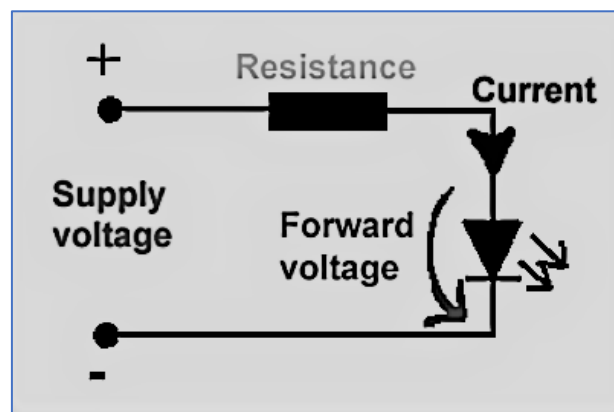


Figura 6 - Circuito LED.

3. Desenvolvimento

3.1. Arquitetura do sistema

3.1.1. Arquitetura de *hardware*

A arquitetura de *hardware* do sistema representa a estrutura dos componentes utilizados na elaboração do projeto e a implementação dos mesmos através de ligações físicas. O sistema que vamos organizar tem como principais componentes o PC onde estão implementados o concentrador e o sistema sensor que envia e recebe informações dos vários sensores e LED's. Na figura 8 apresentamos um esquema que traduz todo o processo elaborado no que diz respeito ao *hardware* por nós utilizado. A ligação entre o concentrador e o sistema sensor será efetuado através de Wi-Fi. O nosso grupo decidiu escolher esta forma de ligação pois achamos ser uma tecnologia recente e em constante desenvolvimento assim como o facto de na nossa pesquisa termos encontrado mais informações sobre o mesmo.

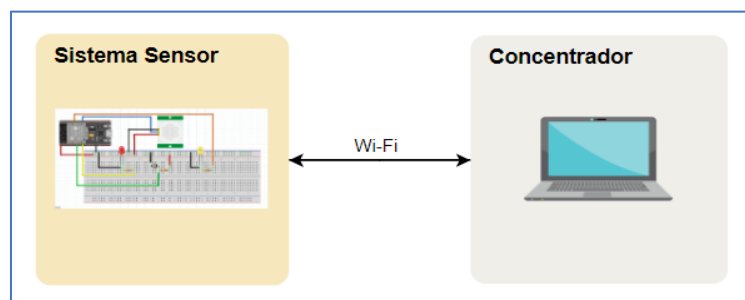


Figura 8 - Circuito do esquema geral.

Na figura 9 apresentamos mais detalhadamente o esquema do sistema sensor composto pela placa ESP32 e pelos componentes que caracterizam o sistema sensor. Salientamos as 2 resistências de 470 Ohms para os LED's e uma de 1k Ohm para o sensor de movimento. Na figura observamos também o sensor de movimento (PIR) conectado à porta 18 do ESP e o sensor de temperatura (LDR) à porta 32.

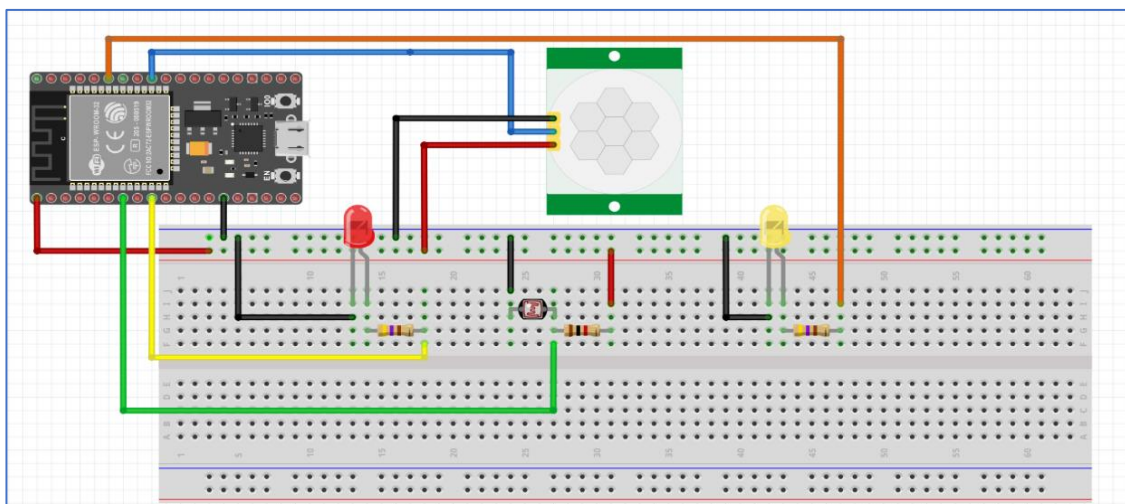


Figura 9 - Circuito Sistema Sensor.

3.1.2. Arquitetura de *software*

No que toca à arquitetura de *software* esta passa pela discussão de certas medidas a tomar relativamente ao protocolo de comunicação entre o concentrador e a placa ESP32. Decidimos utilizar diversas tramas de dados cada uma com informações para o funcionamento eficiente do nosso projeto.

- **START** – Na trama *START* é enviada a informação do concentrador para o sistema sensor para este começar o seu funcionamento, ou seja, diz ao sistema sensor que este pode começar a recolher e a enviar amostras para o concentrador. Esta trama contém os campos apresentados na figura 10.

Tipo de mensagem (1 Byte)	Identificador do sistema (1 Byte)	Timestamp (1 Byte)
------------------------------	--------------------------------------	-----------------------

Figura 10 - Trama *START*.

- **STOP** – Relativamente à trama *STOP* é enviada a informação para o sistema sensor cessar o seu funcionamento, ou seja, diz ao sistema sensor para este não enviar mais amostras e para parar de as recolher. Esta trama contém os campos apresentados na figura 11.

Tipo de mensagem (1 Byte)	Identificador do sistema (1 Byte)	Timestamp (1 Byte)	Razão para a paragem (1 Byte)
------------------------------	--------------------------------------	-----------------------	----------------------------------

Figura 11 - Trama *STOP*.

- **DATA** – Na trama *DATA* é enviado para o concentrador o valor das amostras obtidas pelos sensores assim como o estado das lâmpadas num certo período. Esta trama contém os campos apresentados na figura 12.

Tipo de mensagem (1 Byte)	Identificador do sistema (1 Byte)	Timestamp (1 Byte)	Tipo de grandeza medida (1 Byte)	Número de amostras (1 Byte)	Valor medido (4 Bytes)
------------------------------	--------------------------------------	-----------------------	-------------------------------------	--------------------------------	------------------------

Figura 12 - Trama *DATA*.

- **ERROR** – A trama *ERROR* é enviada da placa ESP32 para o concentrador a indicar que houve um erro no sistema. Esta trama contém os campos apresentados na figura 13.

Tipo de mensagem (1 Byte)	Identificador do sistema (1 Byte)	Timestamp (1 Byte)	Tipo de erro (1 Byte)
------------------------------	--------------------------------------	-----------------------	-----------------------

Figura 13 - Trama *ERROR*.

Referências

- [1] <https://lastminuteengineers.com/pir-sensor-arduino-tutorial/>, Last Minute Engineers.
- [2] <http://mundoprojetado.com.br/ldr-o-que-e-e-como-funciona/>
- [3] <https://www.geeksforgeeks.org/relational-database-from-csv-files-in-c/>