

09\_2019

[Sistema de Monitorização de Canais de Rega]

[Dissertação de Mestrado]

Manuel Lameira – nº 4829 |

**MESTRADO “INTERNET DAS COISAS” – IP BEJA – 2018/2019**

Índice

[1. Introdução 3](#_Toc520746515)

[2. Estado da Arte 3](#_Toc520746516)

[3. Análise do Problema 3](#_Toc520746517)

[4. Arquitetura do Sistema 4](#_Toc520746518)

[4.1 Hardware 7](#_Toc520746519)

[4.1.1 Sensores 10](#_Toc520746520)

[4.1.2 Plataforma Computacional 11](#_Toc520746521)

[4.1.3 Módulo de Comunicações 12](#_Toc520746522)

[4.1.4 Custos do Projeto 13](#_Toc520746523)

[4.2 Software 14](#_Toc520746524)

[5. Desenvolvimento 14](#_Toc520746525)

[5.1 Hardware 15](#_Toc520746526)

[5.2 Software 17](#_Toc520746527)

[6. Teste e Implementação do Sistema 19](#_Toc520746528)

[7. Conclusão 21](#_Toc520746529)

[8. Referências bibliográficas 22](#_Toc520746530)

[9. Apêndices 24](#_Toc520746531)

[Figura 1 - Esquema de montagem dos equipamentos na Obra de Entrada 5](#_Toc520746501)

[Figura 2 - Hipótese 1 - Funcionamento normal da obra de entrada 6](#_Toc520746502)

[Figura 3 - Hipótese 2 - Funcionamento pelo bypass "ladrão" 6](#_Toc520746503)

[Figura 4 - Hipótese 3 - Funcionamento pela abertura da comporta do bypass 6](#_Toc520746504)

[Figura 5 - Diagrama de Blocos da estação recetora 7](#_Toc520746505)

[Figura 6 - Diagrama de Blocos da estação emissora 8](#_Toc520746506)

[Figura 7 - Data Flow Graph 9](#_Toc520746507)

[Figura 8 - Esquema de montagem do circuito a implementar 15](#_Toc520746508)

[Figura 9 - Esquema de montagem alternativo do circuito a implementar 16](#_Toc520746509)

[Figura 10 - Esquema físico com LED ligado e potenciómetro a 30% 16](#_Toc520746510)

[Figura 11 - Código Python 17](#_Toc520746511)

[Figura 12 - Excerto do código C 18](#_Toc520746512)

[Figura 13 - Ciclo do Produto 19](#_Toc520746513)

[Figura 14 - Consume de energia - EnergyTrace™ Technology MSP430FR2433 20](#_Toc520746514)

[Tabela 1 - Dimensões padronizadas do canal Parshall (mm) 4](#_Toc520746532)

[Tabela 2 - Sensor Temperatura 10](#_Toc520746533)

[Tabela 3 - Sensor Altura de Água 10](#_Toc520746534)

[Tabela 4 - Sensor Partículas em Suspensão e Turbidez 11](#_Toc520746535)

[Tabela 5 - Sensor Pulley Switch 11](#_Toc520746536)

[Tabela 6 - Microcontroladores 11](#_Toc520746537)

[Tabela 7 - Fornecimento de energia 12](#_Toc520746538)

[Tabela 8 - Comunicação e transmissão de dados 12](#_Toc520746539)

[Tabela 9 - Sistemas GPS 13](#_Toc520746540)

[Tabela 10 - Custos totais do projeto 13](#_Toc520746541)

# Introdução

A proposta de criação do Mestrado em Internet das Coisas (doravante designado por MIOT) apresenta, entre outras justificações, o facto do Instituto Politécnico de Beja integrar a infraestrutura de investigação científica Engage-SKA. Neste contexto há um tema comum identificado em reuniões com membros do projeto Engage-SKA pertencentes à Escola Superior Agrária do IPBeja: **a água** [1].

No enquadramento dos trabalhos já realizados, nomeadamente nas disciplinas de Dispositivos para a Internet das Coisas e Sistemas Embebidos, onde se escolheu o Sistema de monitorização de Estações de Tratamento de Águas Residuais (ETARs), e é sobre estes que o presente relatório assenta. Descreve-se a metodologia da execução, assim como a sua implementação e as alterações sofridas.

Foram atualizados alguns aspetos do relatório face ao anterior, nomeadamente a adição das figuras no capítulo 3 e a substituição do capitulo 5, 6 e 7.

# Estado da Arte

O único sistema semelhante que foi possível encontrar foi o sistema de medidor de caudal ultrassónico da empresa ecodepur [2]. A empresa faz questão de omitir diversos detalhes sobre o sistema, como a arquitetura do sistema, que tipos de sensores usam no total e como é feita a comunicação entre o equipamento e a base. No entanto, pelo que é possível analisar parte-se do princípio que este sistema é bastante semelhante ao que é idealizado neste projeto, uma vez que a forma como está desenhado o modelo indica-nos que é utilizado um sensor ultrassónico a uma distância fixa e conhecida acima da superfície da água, podendo assim ser medido o volume de água a passar pelo afluente.

Este sistema opera em uma gama de temperatura de -40ºC a +90ºC com uma frequência de 41.5KHz. É possível ser programado através de um teclado de 5 botões e contém um display de 2 linhas de 16 dígitos LCD. Funciona ainda com alimentação 230V AC.

# Análise do Problema

O objetivo do trabalho **sistema para a monitoração de incidentes em estações de tratamento de águas residuais com georreferenciação da informação** assenta numa plataforma de *hardware* e *software* composta por microcontrolador, sistemas de comunicação de dados, a sua programação, sensores, atuadores e a eletrónica de aquisição e condicionamento de sinal.

Este conjunto de equipamento será montado na obra de entrada, mais precisamente no canal Parshall. Deste modo é permitido um controlo contínuo do afluente para a fossa, em situação de normal funcionamento. Em caso excecional de descarga direta através do *By-Pass* (situação de incidente)para a linha de água, é acionado um sensor de contacto localizado na comporta do *By-Pass*, possibilitando quantificar com precisão o caudal, volume e tempo de descarga direta. Para este trabalho iremos focar-nos apenas na situação de incidente, ou seja, descargas diretas para a linha de água.

O sistema permite um maior controlo ambiental uma vez que é autónomo e de comunicação instantânea.

# Arquitetura do Sistema

O sistema será montado na obra de entrada, mais precisamente no canal Parshall.

O canal Parshall é um dispositivo de medição de caudal na forma de um canal aberto com dimensões padronizados. A água é forçada por uma garganta (W) relativamente estreita, sendo que o nível da água a montante da garganta é o indicativo do caudal a ser medido, independentemente do nível de água a jusante. A Tabela 1 [3] mostra os valores padronizados da largura da garganta do canal de Parshall, bem como de outras dimensões do mesmo.

## Hardware

Neste subcapítulo expomos o hardware utilizado para este projeto. O equipamento físico é dividido em dois módulos, o que fica montado na obra de entrada na ETAR a efetuar a recolha e envio dos dados (emissor) e o segundo que recebe e trata os dados (recetor).

Apresenta-se também de seguida os diagramas de blocos dos dois módulos com as respetivas ligações entre os seus componentes e descreve-se as suas constituintes nos pontos seguintes. Tentou-se, sempre que possível, encontrar equipamentos com uma ótima qualidade preço para englobar no projeto de modo a manter os custos o mais baixo possível.

Como referido no artigo anterior, a estação emissora será instalada na obra de entrada, no entanto de modo a economizar energia das baterias o sistema só é ativado quando a comporta do *By-Pass* é acionada.

Para que o sistema funcione, idealizou-se um *switch* mecânico que opere por contacto.

A estação recetora será ligada a um computador (PC ou Raspberry Pi) para tratamento de dados e fornecer uma interpretação gráfica dos mesmos.

Figura 1 - Diagrama de Blocos da estação emissora

Figura 2 - Data Flow Graph

* Os Retângulos representam componentes de hardware e os ovais módulos de software.
* As setas apontam da fonte da informação para o destino

### 4.1.1 Sensores

Medidor de temperatura [4] – A função do medidor de temperatura é como o nome indica, medir a temperatura do afluente. Propomos estes sensores de temperatura:

Tabela 1 - Sensor Temperatura

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Equipamento** | **Marca/local** | **Compativel** | **Voltagem de funcionamento** | **Amper de funcionamento** | **Amplitude de medição** | **Precisão** | **A prova de agua** | **Preço** |
| **DS18B20** | Chines / eBay | Arduino | 3,2 ~ 5,25VDC | 2mA (max) | -55 ~ 110 ℃ | ±0.5ºC@-10~+80'C | sim | 2,89 € |
| **Danfoss MBT 153** | DANFOSS | - | sob consulta | sob consulta | -50~200 ºC | +/- 1°C | sim | sob consulta |
| **ALTA MNS2-9-IN-TS-WT-L03** | MONNIT | - | 3,6V | - | -40°C to +100°C | +/- 1°C | sim | sob consulta |

Ir-se-á utilizar o **sensor DS18B20**, por questões de logística e aquisição.

Medidor de altura de água [5] – Para o medidor de altura utilizaremos um medidor ultrassónico. Após uma pesquisa de mercado encontrou-se os seguintes:

Tabela 2 - Sensor Altura de Água

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Equipamento** | **Marca/local** | **Voltagem de funcionamento** | **Amper de funcionamento** | **Distancia de Medição** | **Precisão** | **A prova de agua** | **Preço** |
| **HC-SR04 Ultrasom** | chines ebay | 5V | 2mA | 2cm ate 450cm | Até 0,3cm | Não | 0,83 € |
| **JSN SR04T Ultrasom** | chines ebay | 5V | 30mA | 25cm ate 4500m | 0,5cm | sim | 6,69 € |
| **VDM18-300/20/88/122/151 LASER** | Pepperl+Fuchs | 30V |  | 80mm ate 300mm | 0,10% | sim | sob consulta |
| **Ultrasom mb1043 EZ4** | Maxbotix | 2.5 – 5.5 V | 2.5mA | 30cm ate 500cm | 1mm | não | 28,10 € |

Por questões de logística e aquisição, optou-se pelo sensor **Ultrasom MB1043 EZ4**.

Sensor de partículas em suspensão [6] – Como sensor de partículas em suspensão iremos utilizar um sensor de turbidez (encontrado nas maquinas de lavar loiça e maquinas de lavar roupa).

Tabela 3 - Sensor Partículas em Suspensão e Turbidez

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Equipamento** | **Marca/local** | **Voltagem de funcionamento** | **Amper de funcionamento** | **Insulation Resistance** | **Preço** |
| **SEN0189** | chines ebay | 5V | 40mA | 100 M | 15,10 € |

Não existe muita escolha no sensor de partículas, dos encontrados, todos se baseiam na base do **SEN0189**.

Sensor de contacto [7] – Para diferenciar se o afluente esta encaminhado pelo *By-Pass* ou diretamente para a fossa, instalamos um sensor de contacto na comporta do *By-Pass* para sinalizar ao microcontrolador a utilização da via alternativa. O microcontrolador indica nas leituras de caudal que o sensor foi adicionado (por exemplo colocando um caractere “B” após a s leituras) e simultaneamente contabiliza o tempo em que o sensor esta acionado, permitindo calcular o fluxo de afluente descarregado pelo *By-Pass*.

Tabela 4 - Sensor Pulley Switch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Equipamento** | **Marca/local** | **Voltagem de funcionamento** | **Amper de funcionamento** | **Preço** |
| **Conjunto de Switch Magnético para Portas** | Boxelectronica | ate 100V | ate 500mA | 5,89 € |
| **Pulley Switch** | Funcionamento Mecânico | | | |

Após uma análise mais cuidada, concluiu-se que o sensor proposto não é funcional. Para o estudo em causa, apenas interessa o afluente que é encaminhado para o *By-Pass*. Utilizando o sensor inicialmente proposto, existe a necessidade de este estar constantemente ligado e consumindo energia valiosa das baterias. Optou-se por colocar um sensor de contacto do tipo ***Pulley Switch*** na comporta do *By-Pass*. Quando a comporta é aberta, o sensor abre o fornecimento de energia ao microcontrolador e consequentemente inicia o sistema.

### 4.1.2 Plataforma Computacional

Microcontrolador MSP430 [8] – De modo a monitorizar os vários componentes do sistema utiliza-se um microcontrolador MSP430 da Texas Instrumentes. Este microcontrolador permite, entre outras funções, a otimização do consumo de energia *(Low Power Mode)*. Microcontroladores adquiridos e disponíveis:

Tabela 5 - Microcontroladores

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Equipamento** | **Quantidade** | **BIT** | **RAM** | **TIMERS** | **PINS** |
| **MSP430G2553** | 2 | 16 | 512 B | 2 | 20 |
| **MSP430F5529** | 2 | 16 | 128 KB | 4 | 40 |
| **MSP430FR5969** | 1 | 16 | 64 KB | 5 | 20 |
| **MSP430FR6989** | 1 | 16 | 128 KB | 5 | 40 |
| **MSP430FR2433** | 1 | 16 | 16 KB | 4 | 20 |

Optou-se por escolher o **Microcontrolador MSP430F5529**, uma vez que satisfaz as necessidades deste projeto, tanto como estação emissora como para a estação recetora dos dados.

Fornecimento de energia [9] – A alimentação do sistema irá ser por meio de baterias e painéis solares. Como os componentes definidos, conseguimos averiguar a necessidade energética do sistema.

Tabela 6 - Fornecimento de energia

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Equipamento** | **Marca/local** | **Voltagem de funcionamento** | **Amper de funcionamento** | **Preço** |
| **Battery Holder Pro KIT 15pcs** | chines ebay | - | - | 8,81 € |
| **3.7V 4000mAh Li Po 125054** | chines ebay |  |  | 7.64 € |

Por questões de logística e aquisição, optou-se pelo sensor **3.7V 4000mAh Li Po 125054**.

Computador – De modo a conseguir tratar os dados transmitidos pelo equipamento instalado na ETAR utiliza-se um computador com capacidade de correr uma base de dados. Quer-se poder averiguar as leituras e executar comparações como por exemplo leituras por data ou valores máximos.

### Módulo de Comunicações

Comunicação e transmissão de dados [10] – Para a transmissão de dados optamos por um sistema repartido por emissão/ receção via radio e uma central para tratamento dos mesmos do tipo *Raspberry Pi*. Propomos:

Tabela 7 - Comunicação e transmissão de dados

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Equipamento** | **Marca/local** | **Voltagem de funcionamento** | **Amper de funcionamento** | **Antena** | **Distancia** | **Preço** |
| **HC-12** | chines ebay | 3,2 - 5,5V | 1uA | 914MHz | 1000m | 2,86 € |
| **Nrf24l01 Radio Transceiver** | chines ebay | 1,9 - 3,6V | 1uA | 2,4GHz | 15 - 610 m | 1,29 € |
| **LoRa SX1278** | chines ebay | 1,8 - 3,6V | 1uA | 868MHz | 2000m | 10,44 € |
| **LoRA Gamma 868** | chines ebay | 3.6 – 15V | 155mA | 869.5MHz | 16000 m | 33,50 € |

De modo a garantir um bom funcionamento da unidade escolheu-se o equipamento **LORA GAMMA 868** com um alcance de 16km.

Sistema de georreferenciação [11] – Para o sistema de georreferenciação ponderou-se os seguintes elementos:

Tabela 8 - Sistemas GPS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Equipamento** | **Marca/local** | **Voltagem de funcionamento** | **Amper de funcionamento** | **Preço** |
| **SIM900** | chines ebay | 4,8 ate 5V | 1.5mA | 14,38 € |
| **GPS Mini NEO 7M** | chines ebay |  |  | 6,17 € |

Por questões de logística e aquisição, optou-se pelo sensor **GPS MINI NEO 7M**.

### Custos do Projeto

Apresenta-se na tabla seguinte o material escolhido para o projeto assim como o seu custo total e sites para fornecimento.

Tabela 9 - Custos totais do projeto

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Equipamento** | **Descrição** | **Valor Unitário** | **Qt.** | **Total** | **Stock Pessoal** | **link vendedor** |
| **Microcontrolador** | MSP430FR5969 | 12,90 € | 1 | 12,90 € | sim |  |
| **Sensor Temperatura** | Waterproof DS18B20 Sensor | 6,50 € | 1 | 6,50 € | sim |  |
| **LoRa** | Gamma-868-SO | 15,15 € | 2 | 30,30 € | não | [ebay](https://www.ebay.com/itm/LORA-TELEMETRY-MODEM-MODULE-16KM-RF-SOLUTIONS-GAMMA-868R/142542083117?epid=2220897919&hash=item21302b942d:g:kHMAAOSwi0xZ5cg5) |
| **Ultrasons** | MB1043 HRLV-MaxSonar-EZ4 | 28,10 € | 1 | 28,10 € | não | [maxbotic](https://www.maxbotix.com/Ultrasonic_Sensors/MB1043.htm) |
| **Sensor de partículas  em suspensão** | SEN0189 | 15,15 € | 1 | 15,15 € | não | [dfrobot](https://www.dfrobot.com/wiki/index.php/Turbidity_sensor_SKU:_SEN0189) |
| **Sensor de contacto** | Pulley Switch | 5,89 € | 1 | 5,89 € | sim |  |
| **Bateria** | 3.7V 4000mAh Li Po 125054 | 7,64 € | 1 | 7,64 € | sim |  |
| **GPS** | GPS Mini NEO-7M | 6,17 € | 1 | 6,17 € | sim |  |
| ***TOTAL*** |  |  |  | *112,65 €* |  |  |
| **TOTAL IPBEJA** |  |  |  | **73,55 €** |  |  |

Estima-se um custo total para o projeto de 112,65€, no entanto o grupo dispõe de algum material necessário (como indicado na tabela) pelo que o custo reduz para **73,55€ (setenta e três euros e cinquenta e cinco cêntimos).**

## Software

O software previsto para este projeto, nesta fase de desenvolvimento, serão os seguintes:

* Criação do código em Code Composer Studio [12] (Texas Instrumentes) para os MSP430;
* Exportação dos dados brutos para ficheiro Excel;
* Criação de base de dados tipo MongoDb [13] para consulta e manuseamento dos dados.

O IDE da Texas Instrumentes permite um controlo a nível de bit nas placas MSP430, o que possibilita colocar as mesmas em baixo consumo de energia (Low Power Mode). Também possibilita uma otimização a nível de linguagem de programação o que consequentemente é mais uma forma para a otimização do consumo energético.

Idealiza-se uma base de dados pensada para “Big Data” como o MongoDb pela incerteza do volume de dados a tratar. O MongoDB é *schemalees* mas estruturado. Quer isto dizer que não ficamos restringidos a uma organização rígida de estrutura para os dados. Esta ultima afirmação pode demonstrar-se particularmente vantajosa em situações de adição de sensores com outros parâmetros de leitura.

# Desenvolvimento

Neste capítulo do relatório pretende-se demonstrar a implementação do sistema a montar na obra de entrada da ETAR (estação emissora) para monitorizar o caudal do afluente, temperatura e turbidez. A localização é obtida pelo modulo GPS. As medições serão enviadas por LoRa para o servidor que se encontra no Instituto Politécnico de Beja (estação recetora) onde os valores serão tratados e armazenados.

Os dados enviados pela estação emissora não são tratados, i.e., apenas envia o valor da voltagem de cada sensor. Com este método, não necessitamos de efetuar cálculos no microcontrolador MSP430 e conseguindo-se uma poupança no consumo de energia do sistema.

Do lado da estação recetora, esta um modulo LoRa ligado ao Raspberry Pi 3, que trata os dados recebidos.

No entanto, com a implantação do sistema surgiram alterações ao equipamento a aplicar, nomeadamente, não se conseguiu adquirir os módulos LoRa e assim substituíram-se pelos módulos GSM/GPRS Sim900 [14]. Esta alteração permite a eliminação do GPS, uma vez que a localização é obtida por triangulação, e representa mais uma economia de energia no sistema.

## 5.1 Hardware

Como *hardware* utilizámos uma placa MSP430G2553, um sensor de turbidez, um sensor analógico de temperatura (NTC thermistor) e um HC-SR04 Ultrasom como medidor de distância.

Por questões praticas alterou-se a placa MSP430FR2433 pela MSP430G2553 e não se implementou o sensor de ultrassom. Este sensor ira depender de uns LED’s que indicam o estado do sistema, a cor vermelha significa comporta acionada, amarelo representa o bypass ladrão e o LED verde indica o estado normal de funcionamento do sistema. No esquema da figura 8 o sensor da comporta é representado pelo botão.

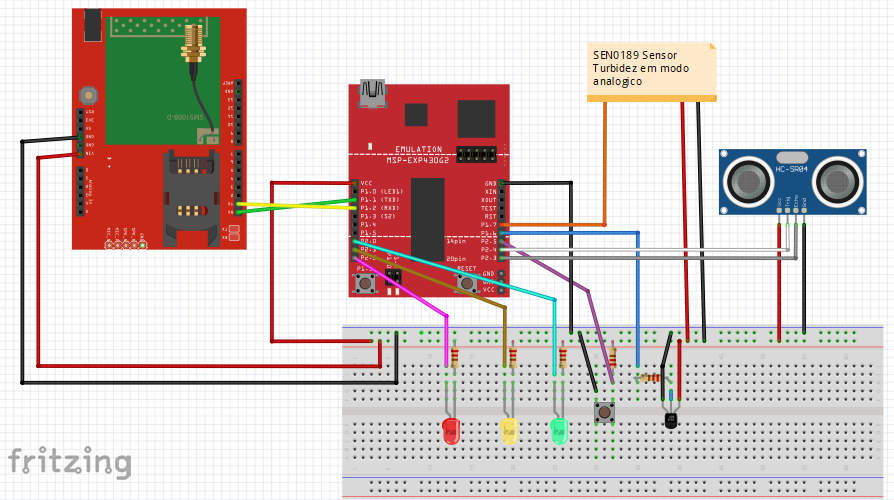


Figura 3 - Esquema de montagem do circuito a implementar

Todo o sistema foi testado em plataforma Arduíno, uma vez que permite um *Debug* mais rápido e intuitivo. Após a realização das ligações na placa de desenvolvimento MSP430G2553 e verificada a consistência do sistema, efetuada por confirmação de receção de dados via USB-TTL na interface *PuTTY [14]*, procedeu-se aos testes de comunicação com o modulo SIM900. Lamentavelmente um desses testes provocou um curto-circuito no modulo de comunicação, o que impossibilitou a implementação estudada do sistema.

Como forma de poder concluir o presente trabalho em tempo útil e por falta de módulos de comunicação *wireless* suplentes ou alternativos *(como por exemplo modulo Bluetooth ou RF433)*, decidiu-se avançar com uma ligação direta TX-RX ao Raspberry Pi 3, conforme apresentado na figura 9.

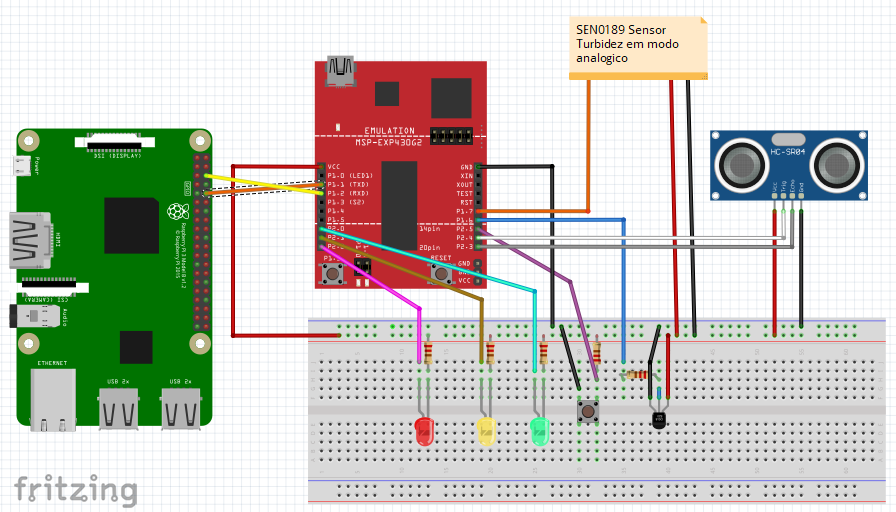


Figura 4 - Esquema de montagem alternativo do circuito a implementar

Os resultados obtidos por meio desta ligação foram os esperados, uma vez que a ligação é bastante básica.

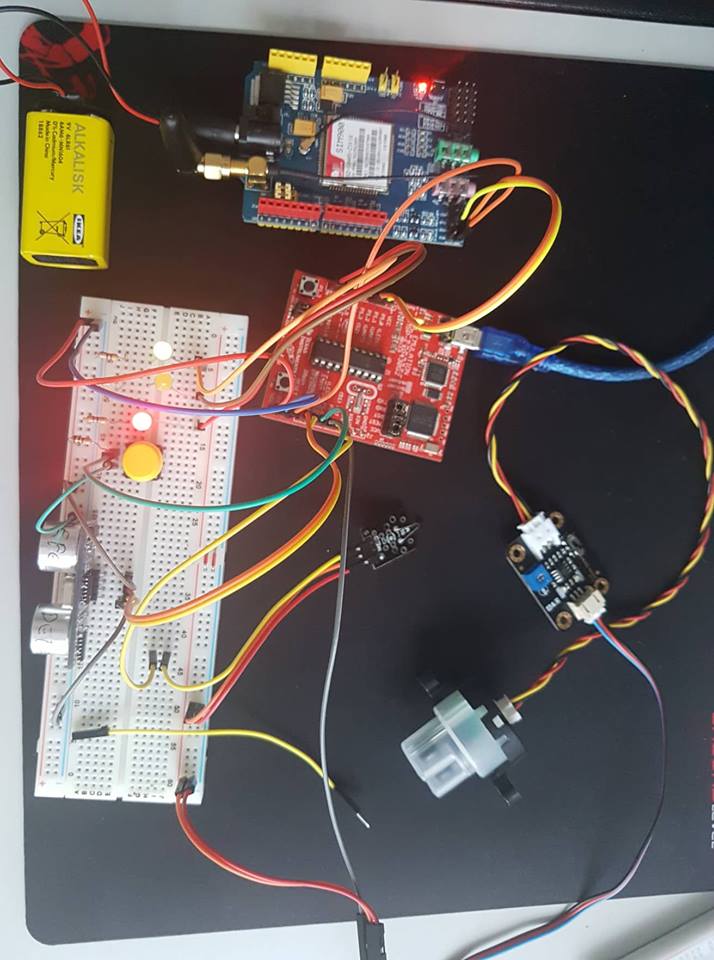


Figura 5 - Esquema físico com LED ligado e potenciómetro a 30%

## 5.2 Software

Para que este exemplo funcione de maneira pretendida foi utilizado a funcionalidade *Comparator*, sendo que foi escolhido como *input* o CA6 (pin P1.6) e como output o pin P1.3, este opera da seguinte forma: sempre que a corrente ultrapasse um valor referência (que neste exemplo será 0.5\*VCC) a placa envia um sinal de HIGH para o pin P1.3, fazendo com que o LED acenda. O código completo utilizado para este exemplo encontra-se nos apêndices.

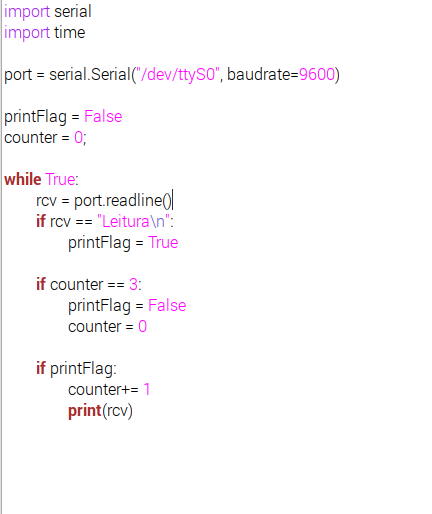


Figura 6 - Código Python

Na figura 11 está presente o código desenvolvido para o equipamento recetor (Raspberry PI). Este código foi produzido utilizando a linguagem Python. Dado o tempo disponível apenas foi possível efetuar uma comunicação básica entre o dispositivo emissor (MSP430) e o recetor, sendo que este código está sempre a correr e por sua vez o equipamento recetor estará sempre “a ouvir” por um sinal. Em ambiente real a comunicação poderia não ser constante, havendo falhas e interrupções no sinal, pelo que seria necessário implementar um ou mais sistemas de *failsafe* tanto no lado do emissor como do recetor.

O objetivo deste código é simplesmente ouvir todas as comunicações no serial “ttyS0” (ou Serial0) e filtrar as informações recebidas de forma a ser possível trata-las posteriormente. Sempre que é recebido uma comunicação “Leitura” o sistema irá preparar-se para ler esta mesma mensagem e as duas seguintes, neste caso irá fazer apenas um print, em ambiente real estas informações seriam enviadas para uma base de dados depois de tratadas.

**Nota**: Para conseguir colocar o Raspberry Pi a ler as comunicações pelo Serial 0 foi necessário alterar o ficheiro de configuração *config.txt* e *cmdline.txt*, de acordo com a documentação do Raspberry Pi, que se pode consultar aqui: <https://www.raspberrypi.org/documentation/configuration/uart.md>.

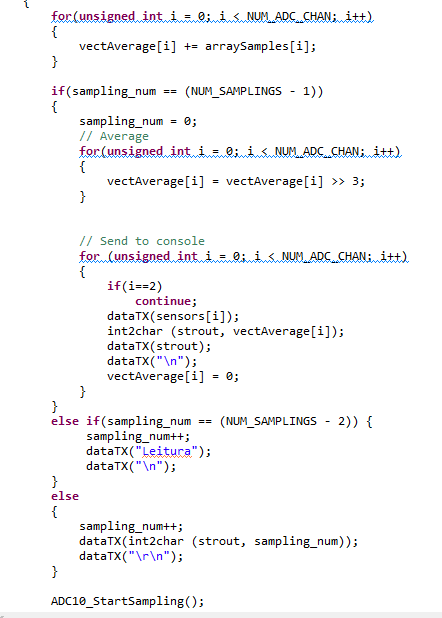


Figura 7 - Excerto do código C

O código para o equipamento emissor (MSP430), excerto presente na figura 12, foi baseado no código exemplo disponibilizado pelo professor, uma vez que este já estava configurado com as funções base para o projeto, nomeadamente a utilização do ADC10 com interrupções e a comunicação UART.

No excerto de código que se consegue visualizar na figura 2 está então programado a recolha de oito leituras dos dois sensores que estão ligados ao MSP430, sendo que no fim de cada ciclo o MSP430 é colocado em *low power mode*. No fim dos oito ciclos é então calculada a média das leitras e feito o envio dos valores através de comunicação UART.

# Teste e Implementação do Sistema

O desenvolvimento de um produto segue um ciclo de analise-design-desenvolvimento-teste-produção a semelhança do indicado na figura 13.

Neste capítulo focamo-nos principalmente na última fase do ciclo de vida de um projeto, a fase de teste e implementação.

* Especificações
* Restrições
* Diagrama de bloco
* Gráfico de fluxo de dados
* Hardware
* Software
* Conclusão
* Novos Requisitos
* Novas Restrições
* Requisitos
* Restrições

Figura 8 - Ciclo do Produto

É importante saliente que apenas utilizamos a fase de teste para o exemplo demonstrado no capítulo 5, uma vez que não temos todo o material necessário para construir o projeto.

A analise e resultados dos parâmetros de otimização são os seguintes:

**Eficiência estática** (utilização da memoria) = 1006kb;

**Eficiência dinâmica** (velocidade de execução) = N.D.;

**Precisão** (diferença entre a verdade esperada e medida) = Preciso;

**Estabilidade** (operação consistente) = Estável

Os valores apurados indicam, no nosso entender, uma boa eficiência. No entanto, o próximo passo será a colocação do resto do código do projeto e depois otimizar o programa como um todo.

Para determinar os valores supracitados utilizou-se a informação apresentada no Code Composer Studio, também se efetuou uma análise de consumo de energia utilizando-se para esse efeito o modo *EnergyTrace™ Technology* do microcontrolador MSP430FR2433.

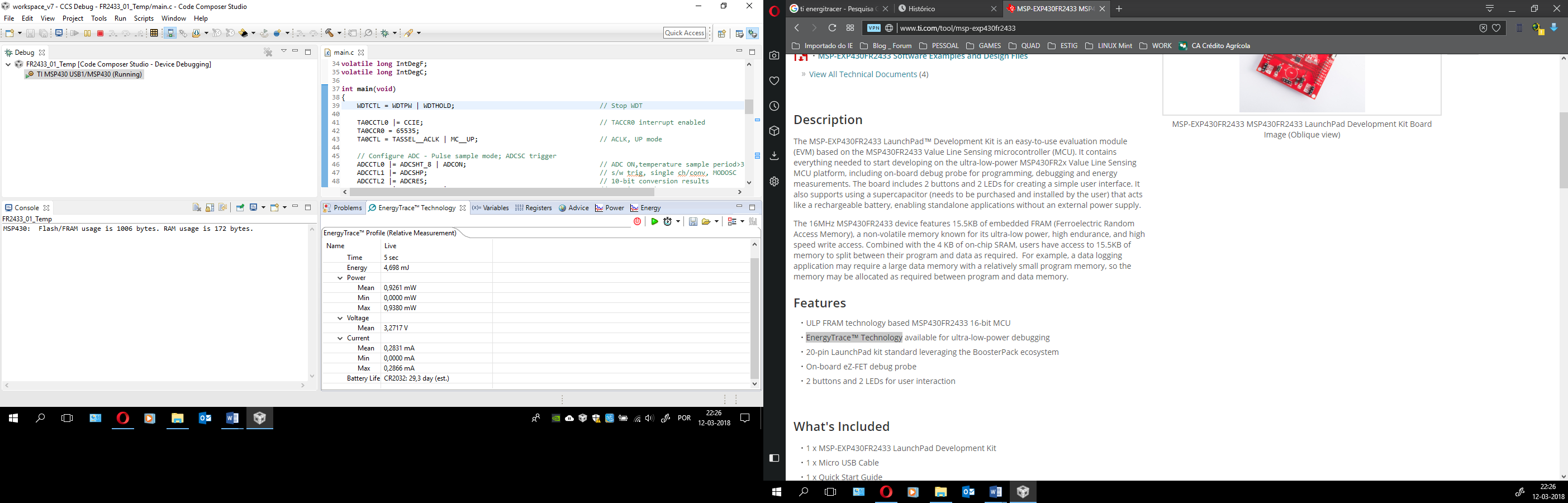


Figura 9 - Consume de energia - EnergyTrace™ Technology MSP430FR2433

É de referir que os valores demonstrados não contemplam o consumo de energia geral do sistema, uma vez que apenas conseguimos medir o consumo do MSP430. Os restantes consumos retiramos dos respetivos *datasheets* dos módulos a incluir no sistema de monotorização.

# Conclusão

Durante a realização deste projeto foram sentidas algumas dificuldades, nomeadamente na implementação das comunicações UART e configurações dos *timers*. A conjugação e definição certa entre os vários parâmetros do código a nível de bit é notoriamente complexa. As constantes mudanças entre manual de utilizador e manual específico não ajudam a uma rápida organização de tarefas a implementar. No entanto serviu para nos familiarizarmos com o mundo dos microcontroladores avançados. Designamos avançados porque em comparação com outros microcontroladores mais *user friendly* (como por exemplo o Arduino), os MSP430 permitem afinações quase cirúrgicas, criando assim sistemas altamente precisos e adaptados ás necessidades dos projetos. Outra grande vantagem é o sistema de *Low Power Mode* que permite “hibernar” o controlador quando este não é utilizado e assim poupar as baterias. Para a implementação deste projeto, foram sacrificados dois integrados MSP430, dois USB-TTL e um modulo GSM/GPRS.

Tendo em conta as dificuldades supramencionadas e o tempo para entrega do trabalho, apenas conseguimos ligar o sistema por comunicação física ao Raspberry Pi e apresentar as leituras na linha de comandos.

Uma possível continuidade deste trabalho, devia prever outros sistemas acessórios interligados (por exemplo comunicação wireless) em todas as comportas que compõem o sistema de bypass, para se poder monitorizar qualquer descarga de afluente sem tratamento ou com tratamento insuficiente para as linhas de água.

É de salientar que o desenvolvimento e implementação deste sistema, no mínimo em todas as obras de entrada das ETAR’s a nível nacional, é urgente e de extrema necessidade para um controlo rigoroso das descargas para as linhas de água. Também é uma forma de salvaguardar o meio ambiente e garantir a qualidade da água nos meios recetores.

# Referências bibliográficas

[1] - EnunciadoTrabalho\_SE\_-2017-2018.pdf

[2] - <http://www.ecodepur.pt/pt/4/produtos>

[3] - <http://www.dec.ufcg.edu.br/saneamento/PARSHALL.html>

[4] - Sensor Temperatura:

* <https://www.ebay.com/itm/DS18B20-Waterproof-Digital-Temperature-Sensor-With-Adapter-Module-for-Arduino-HM/201639236501?hash=item2ef2a29395:g:ziYAAOSwCfdXpFMi>
* <http://www.directindustry.com/industrial-manufacturer/water-temperature-sensor-161988.html>
* <https://www.monnit.com/Product/MNS2-9-IN-TS-WT-L03>

[5] - Sensor de distância:

* <https://www.ebay.com/itm/1pcs-Ultrasonic-Module-HC-SR04-Distance-Measuring-Transducer-Sensor-for-Arduino/400985326881?epid=1638465117&hash=item5d5c968521:g:kLQAAOxyNyFS-xFw>
* <https://www.ebay.com/itm/Arduino-Ultrasonic-Ranging-Measuring-Transducer-Sensor-Waterproof-JSN-SR04T/272266517229?hash=item3f64595aed:g:VvcAAOSwtJZXVnh8>
* <http://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>
* <https://www.pepperl-fuchs.com/global/en/classid_53.htm?view=productdetails&prodid=34337>

[6] - Sensor de partículas em suspensão

* <https://www.dfrobot.com/wiki/index.php/Turbidity_sensor_SKU:_SEN0189>
* <https://www.usinainfo.com.br/outros-sensores-arduino/sensor-de-turbidez-arduino-para-monitoramento-da-agua-4539.html>

[7] - Sensor de contacto

* <https://www.boxelectronica.com/pt/varios-sensores/838-conjunto-de-switch-magnetico-para-portas.html>

[8] - Microcontroladores:

* <http://processors.wiki.ti.com/index.php/MSP430F5529_LaunchPad>
* <http://www.ti.com/tool/MSP-EXP430F5529LP>
* <http://www.ti.com/tool/MSP-EXP430G2>
* <http://www.ti.com/product/msp430fr2422>
* <http://www.ti.com/tool/MSP-EXP430FR5969>
* <http://www.ti.com/product/MSP430FR6989>

[9] - Fornecimento de energia

* <https://www.ebay.com/itm/Battery-Holder-Pro-KIT-15pcs-Power-Supplies-for-Arduino-Solar-4AA-3AA-2AA-DC-EU/222242451205?hash=item33beaeaf05:g:gQgAAOSw-0xYOwYo>

[10] - Comunicação e transmissão de dados:

* <https://www.ebay.com/itm/433Mhz-HC-12-SI4463-Wireless-Serial-Port-Module-1000m-Replace-Bluetooth-TOP/401051275954?epid=26007567495&hash=item5d6084d2b2:g:JawAAOSwMHdXS9OD>
* <https://www.ebay.com/itm/Wireless-Radio-Transceiver-Module-Communication-Control-for-Arduino-DIY/322668815062?_trkparms=aid%3D555018%26algo%3DPL.SIM%26ao%3D2%26asc%3D44040%26meid%3Dbf00b29db36c45de9590955d7d66ce48%26pid%3D100005%26rk%3D1%26rkt%3D6%26sd%3D262966959176%26itm%3D322668815062&_trksid=p2047675.c100005.m1851>
* <https://arduino-info.wikispaces.com/Nrf24L01-2.4GHz-HowTo>
* <https://www.cooking-hacks.com/blog/send-data-at-extreme-long-range-using-lora-with-arduino-raspberry-pi-and-intel-galileo/>
* <https://www.cooking-hacks.com/blog/send-data-at-extreme-long-range-using-lora-with-arduino-raspberry-pi-and-intel-galileo/>

[11] - Sistema de georreferenciação

* <https://www.ebay.com/itm/SIM900-Module-Quad-Band-Development-Board-GSM-GPRS-for-Arduino-Raspberry-Pi-E7G2/253162539156?epid=2248191795&hash=item3af1a9c894:g:HhUAAOSwFuxZwP7n>

[12] – <http://www.ti.com/tools-software/ccs.html>

[13] – <https://www.mongodb.com/?_ga=2.110403590.208858189.1520880704-1686614365.1516833254>

[13] – <https://www.ebay.com/itm/SIM900-850-900-1800-1900-MHz-GPRS-GSM-Development-Board-Module-Kit-For-Arduino/263106376867?hash=item3d425cb4a3%3Ag%3AwJMAAOSwzXBZhuns&_sacat=0&_nkw=sim900&_from=R40&rt=nc&_trksid=p2050601.m570.l1313.TR8.TRC1.A0.H0.Xsim900.TRS0>

[14] – <https://putty.org/>

# Apêndices