

*Desenvolvimento de uma estação
meteorológica em bare-metal e RTOS.*

1º Trabalho prático de Sistemas Embebidos e de Tempo Real

Rúben Guimarães nº11156

Kyrylo Yavorenko nº10355

Escola Superior de Tecnologia, IPCA

Barcelos

06 de Maio de 2018

Conteúdo

Introdução	2
Resumo	3
Objectivos	4
Arquitectura	5
Recursos usados no projecto	6
Esquema do projecto	8
Desenvolvimento	10
Conclusão	13
Bibliografia	15

Introdução

O trabalho prático abordado neste relatório foi desenvolvido no âmbito da unidade curricular Sistemas Embebidos e de Tempo Real do curso de Engenharia de Sistemas Informáticos, lecionada pelo docente António Moreira. O docente desafiou os alunos a aplicarem os conceitos de programação de sistemas embebidos e de sistemas embebidos adquiridos durante o decorrer da unidade curricular. Desenvolvendo um projeto onde o desenvolvimento estivesse dividido em duas partes (baremetal e usando o sistema operativo FreeRTOS) e que adquirisse de diversas fontes de sinais analógicos e digitais para poder replicar o funcionamento de uma estação meteorológica.

Resumo

Neste trabalho desenvolvemos uma pequena estação meteorológica recorrendo a plataforma de prototipagem eletrónica open-source Arduino. Esta estação consiste num conjunto de sensores que obtém dados sobre o estado do tempo que depois são enviados para o Arduino para serem processados e por fim são mostrados ao utilizador quer através de um lcd de 16x2 quer através da consola do IDE do Arduino.

Objectivos

Os objetivos definidos para o projecto pelo docente foram:

- Desenvolvimento de um programa usando a tipologia baremetal.
- Desenvolvimento de um programa usando o sistema operativo FreeR-TOS usando 3 tasks.
- Utilizar os seguintes sensores:
 - Sensor de água.
 - 3 LDR's para calcular a posição do sol.
 - Sensor de humidade.
 - Dois sensores de temperatura (para o ar e o solo).
 - Barómetro.
 - Anemómetro

Arquitectura

Podemos consultar na figura seguinte um diagrama com a arquitectura do projecto.

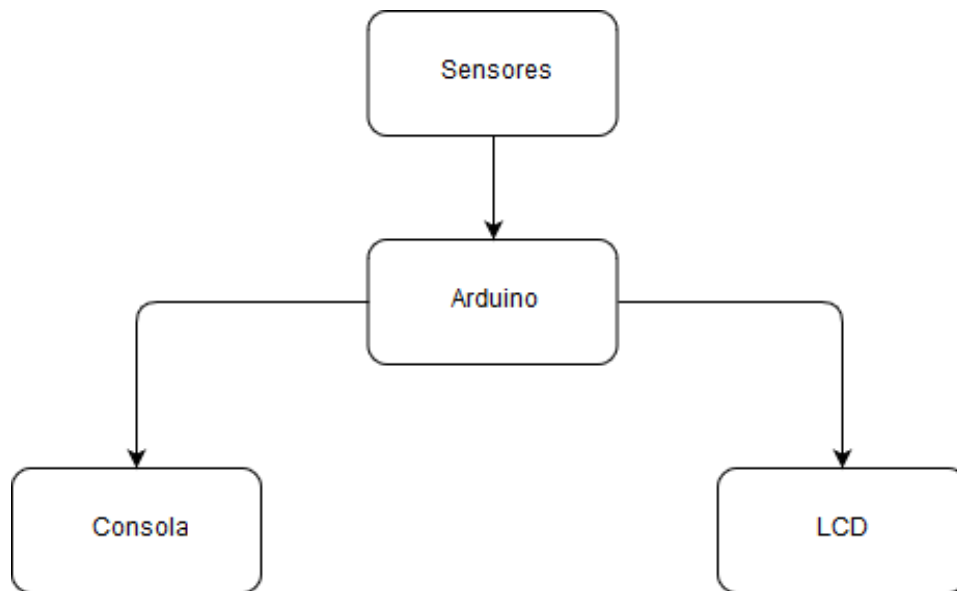


Figura 1: Diagrama da arquitectura do projecto

Recursos usados no projecto

Para o desenvolvimento do projecto foram utilizados os seguintes recursos:

Software:

- *Arduino IDE 1.8.5* para o desenvolvimento do código usado.
- *GitHub Desktop* para atualizar o repositório com o código do projecto.
- *Fritzing 0.9.3b* para criar os protótipos dos esquemas do projecto.

Hardware:

- *Arduino Mega 2560*, placa que contém o microcontrolador que controla tudo o nosso projecto.
- *DHT11*, sensor usado para obter leituras da humidade e da temperatura do ambiente.
- *DS18B20*, sensor à prova de água usado para obter temperaturas de água.
- *BMP180*, sensor que obtém leituras da pressão atmosférica e da temperatura ambiente, com os valores obtidos conseguimos fazer uma estimativa da altitude.
- *Water Sensor*, sensor usado para saber se existe água ou não.

- *Buzzer*, usado para dar um feedback sonoro caso seja detectada água no sensor de água.
- *QTR-8RC*, sensor usado para contar as rotações por minuto de um cata-vento.
- *LCD 16x2*, usado para mostrar os valores obtidos pelos sensores.
- *Light Dependent Resistor*, usado para descobrir qual dos lados (este, sul ou oeste) o sol de encontra.
- *Breadboard*, usada para montar o circuito.
- *Diversas resistências*, usadas para proteger componentes ou para servir de pull-up.
- *Diversos fios*, usados para efetuar as ligações entre componetes, breadboard e o arduino.

Esquema do projecto

Na imagem seguinte podemos verificar o esquema do projecto desenvolvido na plataforma fritzing.

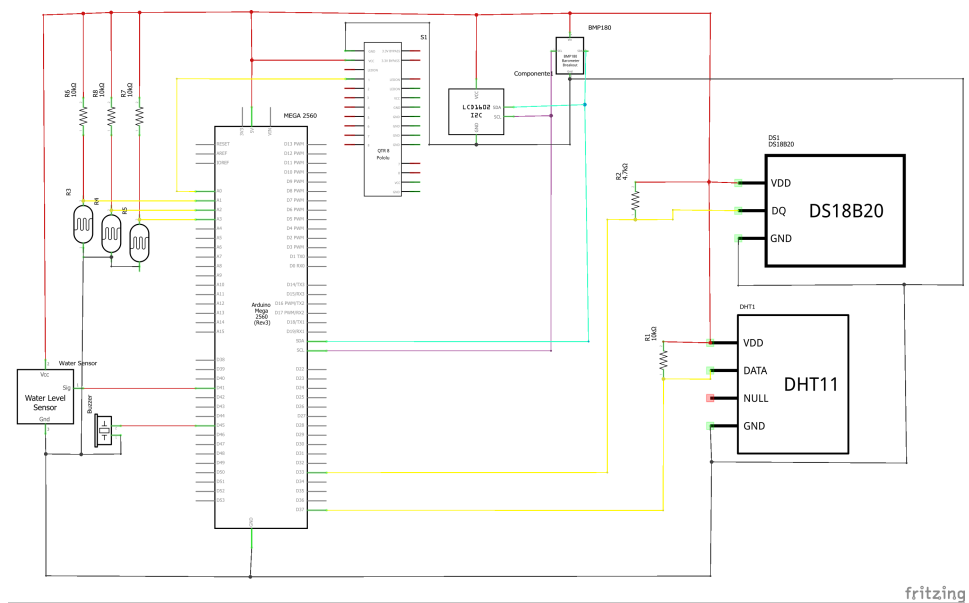


Figura 2: Esquema do projecto

Na imagem seguinte podemos ver o projecto totalmente finalizado.

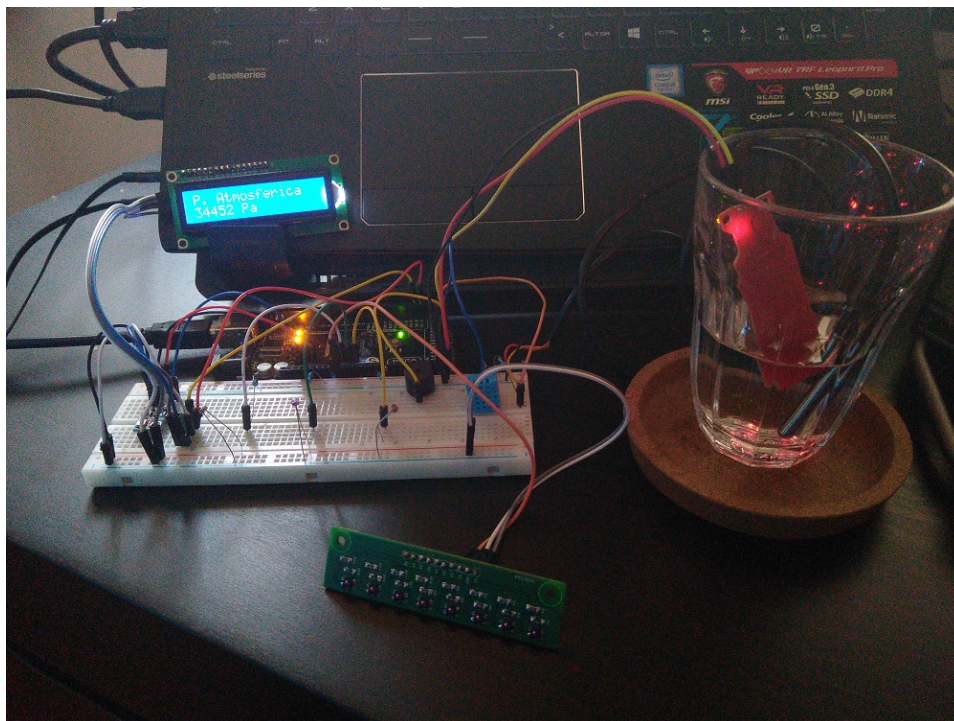


Figura 3: Foto do projecto finalizado.

Desenvolvimento

Inicialmente decidimos que a nossa abordagem ao projecto seria começarmos por bare metal, testar todos os sensores e as ligações nesta metodologia de código e só depois passar a parte que iríamos recorrer ao sistema operativo FreeRTOS.

Começamos por testar o sensor de temperatura DS18B20, para este recorremos a a biblioteca OneWire.h [1] para facilitar a leitura dos dados dos sensores. [2] [3]. De seguida passamos ao sensor DHT11 que nos permite medir a percentagem de humidade do ambiente e a temperatura. Para este recorremos a biblioteca DHT.h [4]. [5]. Depois passamos ao barómetro BMP180, este sensor permite-nos medir a pressão atmosférica, a temperatura e com a pressão atmosférica podemos calcular aproximadamente a altitude que encontra. Para obter os valores recorremos a biblioteca Wire.h (nativa do arduino) e a Adafruit_BMP085.h [6]. [7]. Depois do circuito destes sensores montados e o código escrito, passamos a imprimir o valor deste na consola para verificar se estava tudo a funcionar correctamente.

O próximo passo foi testar o sensor de detecção de água, este foi bastante simples tendo em conta que devolve um sinal HIGH (5V) caso seja

encontrada água nos contactos do sensor. [8]. Depois montamos os 3 LDR's que servem para determinar a posição do sol, este foram também bastaste simples tendo em conta que já tínhamos feito circuitos com a utilização de LDR's no decorrer da cadeira. Só tivemos que os ligar e determinar no código qual dos LDR's estava a obter um valor de resistência mais baixo.

Por fim faltava utilização de LED's infravermelhos para ler as rotações de um cata-vento, primeiro testamos utilizar simples LED's IR e contar as vezes que este era interrompido num certo período de tempo. Não obtimos sucesso, por isso recorremos ao sensor QTR-8RC mas utilizamos apenas um dos conjuntos de LED's. Para obter a leitura das Rotações por minuto fizemos uma função que durante 5 segundos consulta o valor obtido na porta analógica do LED receptor de 50 em 50 milissegundos. Caso este valor seja inferior ao do threshold definido (800) conta uma volta. Também tivemos que definir uma flag para evitar as mesmas leituras (caso o cata-vento estivesse parado). Para calcular um valor das rotações por minuto efectuamos uma regra de 3 simples para efectuar uma estimativa das rotações num minuto.

Como podemos verificar na imagem seguinte, estávamos a obter os valores dos sensores todos na consola.

```
*** Valores Lidos ***  
Humidade ambiente: 30%  
Temperatura ambiente: 19°C  
Temperatura água: 17.00°C  
Temperatura BMP: 17.60°C  
Pressão atmosférica: 34460Pa  
Altitude: 111m  
Água no sensor ? Não!  
Velocidade vento: 36 RPM  
Posição do sol: Sul
```

Figura 4: Valores obtidos na consola do IDE.

O único aspecto que nos faltava para terminar a implementação em bare metal era mostrar os valores dos sensores num LCD 16x2.

Conclusão

Este trabalho permitiu-se aplicar os conhecimentos adquiridos durante o desenvolver da unidade curricular de Integração de Sistemas de Informação e explorar e desenvolver processos de interoperabilidade entre sistemas, assentes em serviços web. Uma das partes que correu mal no trabalho foi o uso da base de dados alojada no *Azure* que por algum motivo não mantinha a ligação aberta quando a tentava usar no serviço. De qualquer forma acho que este trabalho foi um sucesso tendo conseguido alcançar os meus objetivos e ficando a conhecer mais sobre serviços RESTful.

Bibliografia

- [1] PaulStoffregen. *OneWire.h*. 06 Abril, 2018. <https://github.com/PaulStoffregen/OneWire>
- [2] dfrobot. *Waterproof DS18B20 Digital Temperature Sensor*. 06 Abril, 2018. <https://github.com/PaulStoffregen/OneWire>
- [3] sparkfun. *DS18B20*. 06 Abril, 2018. https://github.com/sparkfun/simple_sketches/blob/master/DS18B20/DS18B20.ino
- [4] Adafruit. *Adafruit Unified Sensor Driver*. 06 Abril, 2018. https://github.com/adafruit/Adafruit_Sensor
- [5] lady ada. *Connecting to a DHTxx Sensor*. 06 Abril, 2018. <https://learn.adafruit.com/dht/connecting-to-a-dhtxx-sensor>
- [6] Adafruit. *A powerful but easy to use BMP085/BMP180 Library*. 06 Abril, 2018. <https://github.com/adafruit/Adafruit-BMP085-Library>
- [7] Adilson Thomsen. *Controlando temperatura e pressão com o BMP180*. 06 Abril, 2018. <https://www.filipeflop.com/blog/temperatura-pressao-bmp180-arduino/>

[8] tutorialspoint. *Arduino - Water Detector / Sensor*. 20 Abril, 2018. https://www.tutorialspoint.com/arduino/arduino_water_detector_sensor.htm