

*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
API's externas usadas	7
Desenvolvimento	8
Conclusão	11
Bibliografia	12

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.

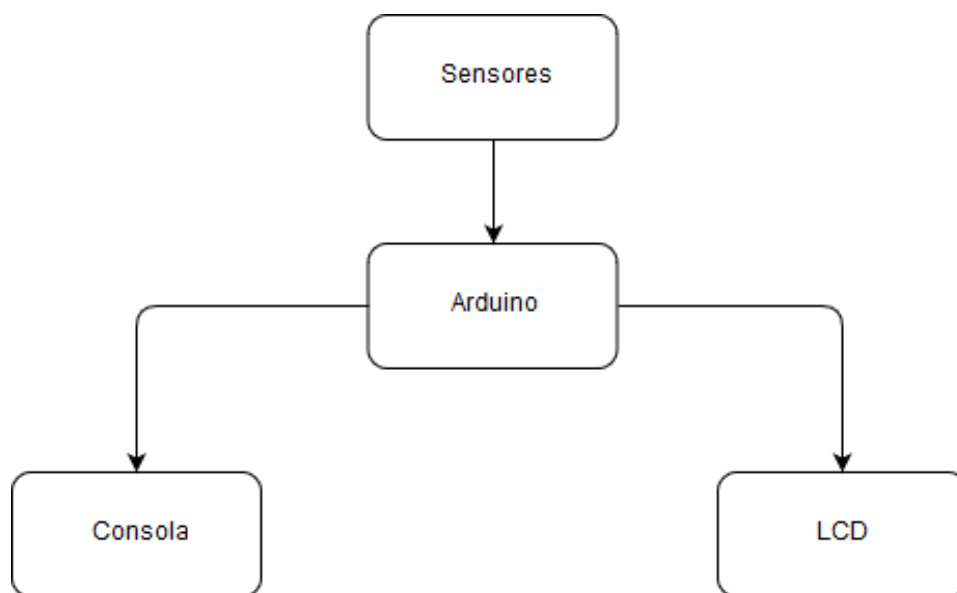


Figura 1: Diagrama da arquitectura do projecto

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.

Outros:

- *json2csharp* para a criação de classes dos ficheiros *JSON*. [1]
- *TinyTwitter* usada para facilitar a comunicação com o *Twitter*. [2]
- *Azure* usado para alojar o serviço e base de dados. [2]

API's externas usadas

Foram usadas 4 API's externas:

- *ipapi.co* - Usada para receber um ficheiro *JSON* com as informações de um dado endereço IP.[4]
- *ipify A Simple IP Address API* - Usada para receber uma string com o endereço IP da nossa ligação.[5]
- *Twitter Developer Platform* - Usada para publicar *Tweets* no *Twitter*. [6]
- *OpenWeatherMap* - Usada para receber um ficheiro *JSON* com a informação meteorológica de uma dada cidade. [7]

Desenvolvimento

O serviço foi desenvolvido recorrendo a um serviço do tipo *Windows Communication Foundation* (WCF). Este efetua a comunicação com as API's externas, trabalha os dados recebidos (se necessário) e disponibiliza serviços para um ou mais cliente usarem. Os serviços que disponibiliza são os seguintes:

- **GetIPInfo/{enderecoIP}** do tipo GET que envia a informação do endereço IP recebido no campo `enderecoIP`, num ficheiro *JSON*
- **MyIp** do tipo GET que envia a informação do endereço IP da ligação numa string.
- **Tweet** do tipo POST que recebe uma mensagem e publicar essa mensagem na conta <https://twitter.com/trabalhoisi>. Este recorre a uma biblioteca[2] para facilitar a comunicação com o *Twitter*.
- **Weather/{nomeCidade}** do tipo GET que envia a informação meteorológica da cidade recebida no campo `nomeCidade`, num ficheiro *JSON*

O serviço também contém objectos para guardar as respostas recebidas em *JSON* criados no serviço *json2csharp* tal como podemos verificar na imagem seguinte:

Este serviço atualmente está publicado no *Azure* e pode ser chamado usando o seguinte link `http://wcfrest20180109101801.azurewebsites.net/Service.svc/`.

O cliente foi desenvolvido recorrendo ao *Windows Presentation Foundation* (WPF) que recorre a linguagem de marcação *Extensible Application Markup Language* (XAML) . Este é composto por 3 abas (Endereços de IP/ Twitter / Meteorologia) onde existe uma interface onde podemos testar os serviços desenvolvidos e ver os resultados. Podemos ver a aba do *Twitter* na imagem seguinte.

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] *json2csharp* . 01 Janeiro, 2018. <http://json2csharp.com/>
- [2] jmhdez. *TinyTwitter*. 01 Janeiro, 2018. <https://github.com/jmhdez/TinyTwitter>
- [3] *Azure*. 01 Janeiro, 2018. <https://portal.azure.com/>
- [4] *ipapi.co*. 01 Janeiro, 2018. <https://ipapi.co/>
- [5] *ipify* A Simple IP Address API. 01 Janeiro, 2018. <https://www.ipify.org/>
- [6] *Twitter* Developer Platform. 03 Janeiro, 2018. <https://developer.twitter.com/>
- [7] *OpenWeatherMap* Current weather and forecasts in your city. 05 Janeiro, 2018. <http://openweathermap.org/current>