
SISTEMA COM ARDUINO UNO E RASPBERRY PI 3

Ana Martins, Daniela Cardoso, Inês Moreira
Prof. Luís Melo
Instrumentação e Aquisição de Dados
Instituto Superior Técnico - LEFT

1 Objetivos

O objetivo principal deste trabalho foi estabelecer uma ligação via USB entre o Arduino Uno e o Raspberry Pi 3, criando um sistema com os dois microcontroladores capaz de comunicar entre si lendo e enviando sinais.

2 Arduino

O Arduino está programado de modo a que fique à espera de um comando do Raspberry Pi para adquirir um valor de uma das suas entradas analógicas. Caso o comando não seja o correto, o Arduino envia uma mensagem de erro ao Raspberry Pi. Para o comando correcto envia o valor adquirido de volta ao Raspberry Pi por USB.

2.1 Código Arduino

Utiliza-se uma baud rate de 9600 bits por segundos, esta tem de ser a mesma a utilizar no Raspberry Pi para que se possa ler e enviar informação de forma correta.

Utiliza-se a função `Serial.available()` para saber o número de caracteres disponíveis para leitura da porta serial. Caso haja caracteres para leitura, o Arduino lê com a função `Serial.readString()` e guarda a informação na `string data`.

Posteriormente compara-se a `string data` com o comando correto pretendido - "start" - através da função `equalsIgnoreCase()`. Se o comando for correto, o Arduino envia pela porta serial um valor entre 0 e 1023 tem correspondência a uma tensão lida na porta analógica A0 - este valor foi anteriormente guardado utilizando a função `analogRead()`. Caso o comando não seja o correto, o Arduino devolve a mensagem "fatal error".

Em ambos os casos utilizamos a função `Serial.flush()` para garantir que a transmissão dos dados é terminada antes do programa prosseguir.

3 Raspberry Pi

O Raspberry Pi corre um programa Python3 que utiliza a biblioteca PyQt5 para criar uma janela. Nesta janela está um gráfico feito com a biblioteca pyqtgraph, este gráfico xy apresenta os valores recebidos do Arduino em tempo real. A janela tem ainda dois push buttons, o start/stop e o clean.

3.1 Código Raspberry Pi Python

Definiu-se a classe `MainWindow()` que foi utilizada para criar a janela e todos os métodos necessários para o gráfico e botões. A janela tem ainda definida a função `initialQuestion()` responsável por garantir que o utilizador, ao inicializar o programa, fornece o comando correto para leitura e aquisição de dados.

3.1.1 Plot xy

Os arrays `x` e `y` têm, cada um deles, 100 pontos e vão sendo atualizados pela função `update_plot_data()`. Esta função remove o primeiro elemento de cada array, manda ao Arduino o comando dado pelo utilizador e lê a resposta do Arduino que fica guardada na variável `aux`. Finalmente utiliza-se `y.append(int(aux))` para adicionar o novo valor lido ao plot.

3.1.2 Botões Start/Stop e Clean

Criaram-se dois botões com a função `QPushButton()`, um com função de paragem de desenho do gráfico e retoma quando este está pausado e outro com função de limpeza do canvas, um espécie de reset ao gráfico. As funções que permitem fazer isto são `startMethod()` e `cleanMethod()`.