
1º TRABALHO

INSTRUMENTAÇÃO E AQUISIÇÃO DE DADOS

Ana Sofia Sousa
96508

Duarte Marques
96523

João Chaves
96540

Objetivos

Este trabalho teve como objetivo a conexão entre um Raspberry Pi 4 Model B e um Arduino UNO, a fim de efetuar aquisição de dados, mais concretamente a variação de uma tensão ao longo do tempo. A aquisição deverá ser realizada pelo Arduino (porta analógica **A5**) e comandada pelo Raspberry Pi. Este último apresentará ainda os dados via *plot*.

Trabalho realizado

Para este fim, foram elaborados dois programas: um *script* em Python3 (`main_raspberry.py`) para executar no Raspberry Pi e um programa para executar no Arduino (`main.ino`).

No código elaborado para o Arduino, seleciona-se o *baud rate* a utilizar para a comunicação como sendo de 9600, na função **setup**. Este espera continuamente por um *input* do Raspberry Pi, ligado via porta *serial*. Quando recebido este comando, é comparado com o comando esperado ('1'). Se for o correto, é retornado para o Raspberry Pi o valor da porta analógica A5. Se não for o correto, é enviada uma mensagem de erro (neste caso, a letra 'e').

No Raspberry Pi, é executado um programa em Python3, o qual produz uma janela utilizando a biblioteca gráfica PyQt5. Esta janela inclui três botões e duas *spinboxes*:

- **Start** - botão para começar a aquisição periódica de dados
- **Stop** - botão para terminar a aquisição periódica de dados
- **Command** - botão para pedir apenas 1 dado ao Arduino
- **Rate** - *spinbox* para controlar a periodicidade da aquisição de dados, em milissegundos
- **Window width** - *spinbox* para controlar a quantidade de dados mostrados no ecrã, em segundos (mostra os últimos n segundos)

Em primeiro lugar, o programa procura a entrada à qual estará conectado o Arduino, terminando a sua execução caso esta não seja encontrada. De seguida, desenha a janela, estabelece a ligação (com o mesmo *baud rate* do Arduino) e fica à espera de ações (via cliques nos botões ou alterações nas *spinboxes*). Clicando no botão **Start**, começa a aquisição com a periodicidade selecionada. Clicando no **Stop**, esta termina. A qualquer altura é possível clicar no **Command** e requerir um dado adicional. À medida que os dados são recebidos, são impressos na consola.

É de notar que alterar qualquer uma das *spinboxes* quando o programa está a adquirir dados resulta na paragem desta aquisição. Quando a aquisição é terminada e recomeçada, os dados são eliminados. Existe ainda um mecanismo para evitar a ocupação de demasiada memória: quando elapsou o tempo correspondente a dez larguras de janela, o programa para e recomeça a aquisição automaticamente (efetivamente apagando os dados anteriores). Os dados não são realmente armazenados, mas podem ser visualizados ao arrastar a janela do *plot*.

No *plot*, os dados são apresentados em função do tempo passado desde o início da aquisição (em segundos) e como diferenças de potencial. O Arduino interpreta as tensões de 0 a 1023, o que corresponde na realidade a tensões entre 0 e 5 V, pelo que esta correspondência é realizada quando os dados são recebidos. Os dados recebidos são também impressos no terminal, bem como algumas indicações do programa (início ou fim da aquisição, erros, etc.).

Conclusões

Foi estabelecida com sucesso a ligação entre um Arduino UNO e um Raspberry Pi. Este *hardware* foi utilizado para realizar um *plot* dos valores de tensão (obtidos de um pino analógico do Arduino) ao longo do tempo. A utilização das bibliotecas 'PyQt5' e 'pyqtgraph' foi necessária à criação da janela na qual é apresentado o gráfico $V(t)$ e os *widgets* que permitem controlar a aquisição de dados. Considera-se que os objetivos a que nos propusemos foram cumpridos.