

Escola Politécnica da Universidade de São Paulo

GPI²O Simulator

PCS3732 - Laboratório de Processadores
Gregory do Amaral, 11375354
Miguel Velasques Abilio Piola Alves, 11807601
Pedro Henrique Machado Almeida, 12553559

São Paulo
Agosto/2024

Sumário

Objetivo	3
Protocolo de comunicação	3
Desenvolvimento da Aplicação alto nível do Simulador	3
GUI	3
Monitoramento das GPIO	4
Integração entre UART e SERIAL	4
Integração entre UI e UART	5
Demonstração do projeto	6
Documentação do projeto	6
Referências	6

Objetivo

O objetivo do nosso projeto consiste em implementar um monitor de pinos GPIOs da placa Raspberry Pi, que consiste em fazer com que a placa envie informações constantemente para um computador através de comunicação serial UART. O computador irá receber e interpretar essas informações recebidas e então exibirá no monitor uma interface gráfica representando o estado de cada pino da placa. Esse projeto foi motivado pela necessidade de testar códigos sem que houvesse a necessidade de componentes eletrônicos adicionais a Raspberry Pi. Por conta de ser um monitor de GPIO de uma Raspberry Pi, decidimos batizar o projeto como GPI²O Simulator.

Protocolo de comunicação

Para a realização desse projeto, decidimos implementar um protocolo de comunicação por meio da UART, para garantir comunicação eficiente entre todas as partes do nosso projeto. O protocolo consiste em enviar inicialmente uma palavra definindo o início da mensagem, em seguida enviar o número de gpios ativos, em seguida inúmeras palavras de 8 bits com a seguinte distribuição:

- 5 bits para identificação da GPIO
- 1 bit para representação de saída ou entrada
- 1 bit para representação de nível lógico alto ou baixo
- 1 bit reservado

Por convenção, decidimos que o início seria representado pelo caractere “S” de start e o fim seria representado pelo caractere “E” de end.

Desenvolvimento da Aplicação alto nível do Simulador

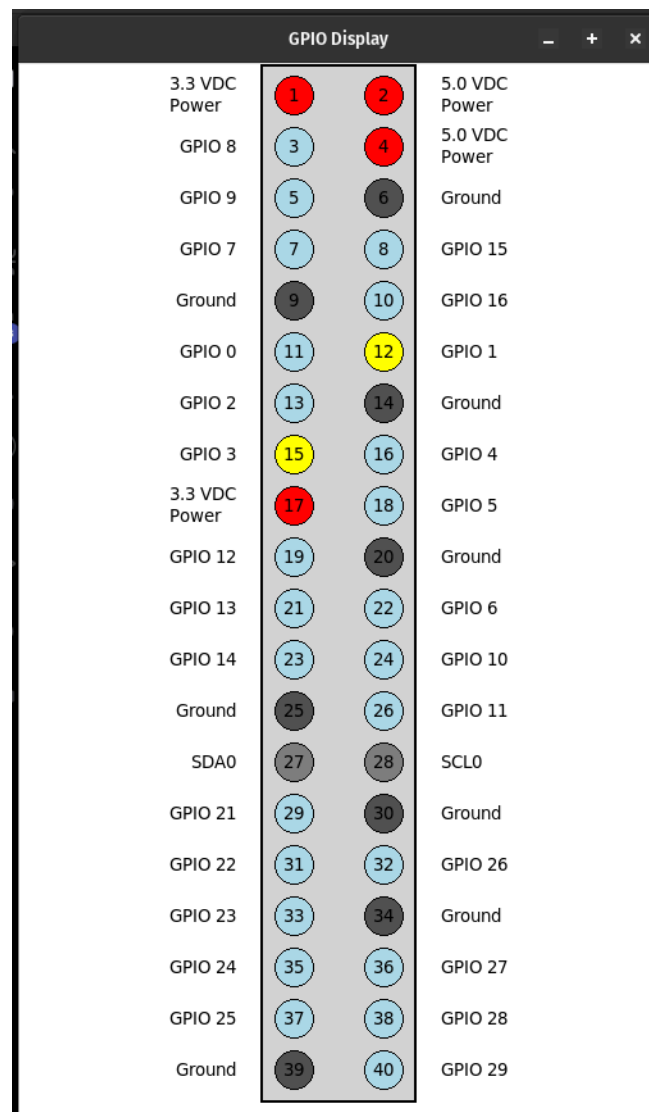
Uma vez que o protocolo de comunicação foi convencionado, foi desenvolvido uma GUI da aplicação, assim como o código responsável por receber e tratar os dados enviados pela uart, tentando abstrair ao máximo a parte de baixo nível do transmissor. Para isso, utilizou-se a linguagem de programação Python, juntamente da biblioteca Tkinter, para o desenvolvimento da GUI, e PySerial, para a parte de recepção dos dados pela UART.

GUI

A GUI desenvolvida para o projeto tem o propósito de apresentar todos os 40 pinos externos da placa, simplificando todos os pinos que podem ser configurados como GPIO como GPIOs, além dos pinos de 3v3 e GND. Para isso temos a seguinte disposição:

- Pinos Vermelhos: 3v3
- Pinos Cinza-Escuros: GND
- Pinos Azul-Claros: GPIOs em baixo/desativados
- Pinos Amarelos: GPIOs em alto/ativos

Na figura 1, é possível ver a GUI para um exemplo onde os GPIOs 1 e 3 estão em alto,



enquanto todos os outros estão desligados.

Figura 1 - Interface das GPIOs

Monitoramento das GPIO

O sistema de monitoramento das GPIOs consiste em detectar as mudanças nos registradores responsáveis por cada GPIO e a partir dessa detecção, enviar diversas mensagens para o computador através de comunicação serial. Essa detecção é feita em com duas frentes:

- Um loop verifica o registrador *gplev* para detectar o nível de pinos de input
- O sistema provê um método void `monitored_gpio_set(uint32_t gpio, uint32_t value)` para que pinos de output possam ser escritos e seus valores salvos no sistema de monitoramento

Os dados dos pinos são salvos conforme mostrado abaixo:

Número do GPIO	Direção	Valor	Forçado
Valor de 0 a 29	0 - output 1 - input	0 - low 1 - high	1 - forçado pelo <i>monitored_gpio_set</i>
5 bits	1 bit	1 bit	1 bit

Tabela 1: Modelo de dados dos pinos

Integração entre UART e SERIAL

O desenvolvimento do receptor serial também inclui a implementação de um decodificador das mensagens enviadas, em acordo com o protocolo de comunicação desenvolvido. Para a validação desta parte, foi criado um Mock para o transmissor serial, a fim de emular a interface UART. A figura 2 que segue apresenta uma saída de console que demonstra que a informação enviada de que o GPIO 1 e 3 estão ligados está sendo corretamente recebida e decodificada pelo algoritmo.

```

GPIOs ativos: 2
GPIO Número: 1, Direção: out, Estado: high
GPIO Número: 3, Direção: out, Estado: high
GPIOs ativos: 2
GPIO Número: 1, Direção: out, Estado: high
GPIO Número: 3, Direção: out, Estado: high
GPIOs ativos: 2
GPIO Número: 1, Direção: out, Estado: high
GPIO Número: 3, Direção: out, Estado: high
GPIOs ativos: 2
GPIO Número: 1, Direção: out, Estado: high
GPIO Número: 3, Direção: out, Estado: high
GPIOs ativos: 2
GPIO Número: 1, Direção: out, Estado: high
GPIO Número: 3, Direção: out, Estado: high

```

Figura 2 - Saída do console recebendo dados

Integração entre UI e UART

No laboratório, o grupo conseguiu testar a integração entre a UI e o transmissor UART na Raspberry Pi. Tudo funcionou como esperado: a Raspberry enviou os bytes de forma correta e o algoritmo do simulador no computador recebeu, decodificou a mensagem e apresentou na tela a disposição em questão, podemos observar o resultado dessa integração na Figura 3.

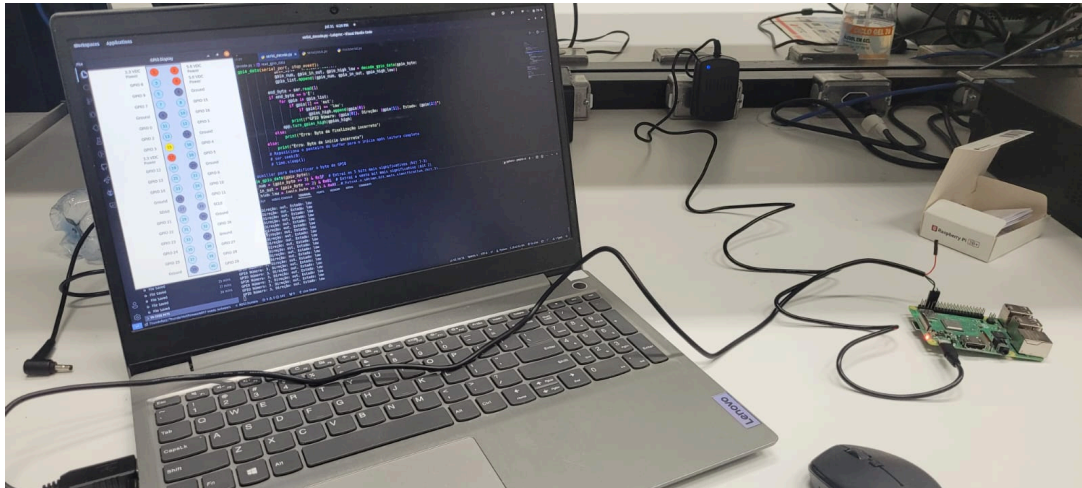


Figura 3 - Testes no laboratório

Demonstração do projeto

A demonstração do nosso projeto com alguns programas exemplos do usuário pode ser vista através do seguinte vídeo: <https://youtu.be/J80HAff5Vio>

Documentação do projeto

O código e implementação prática do nosso projeto pode ser visualizado no seguinte repositório no github: <https://github.com/miguel-usp/PCS3732-Projeto-Gpi2oSim>

Referências

1. <https://www.wise-ware.com.br/pcs3732/t/geral>
2. <https://www.wise-ware.com.br/poli/wiki/>
3. <https://docs.python.org/pt-br/3/library/tkinter.html>
4. <https://s-matyukevich.github.io/raspberry-pi-os/docs/lesson01/rpi-os.html>