

Instruções para montagem do PiLock

Um projeto da disciplina de redes de computadores

Matheus Alves, Claudio Lima, Pedro Neto, Pedro Almeida

¹Instituto metrópole digital – Universidade Federal do Rio Grande do Norte (UFRN)
Av. Capitão Mor Golveia - Lagoa Nova, Natal - RN, 59078-970

{matheuseejam, pedropaivaneto}@hotmail.com, cadoafb@gmail.com, pedrohenriqueg@yahoo.com

Abstract. *This paper describes the work of PiLock: remote physical access control with RPi of the Matter of Computer Networks. The objective of the work was to simulate the reading of magnetic cards using an RFID reader on an Arduino platform.*

Resumo. *Este artigo descreve o trabalho de PiLock: controle de acesso físico remoto com RPi da matéria de Redes de Computadores. O objetivo do trabalho foi simular a leitura de cartões magnéticos utilizando um leitor de RFID em uma plataforma arduino.*

1. Informações Gerais

O projeto foi implementado com uso os de um Leitor RFID integrado no Arduino Mega e um Raspberry pi.

Para a elaboração deste projeto alguns materiais são necessários. Um arduino Mega foi usado para a conexão e gerenciamento da parte física do projeto. Ele controla os leds além do próprio leitor RFID.

Um leitor RFID será foi usado para ler os cartões magnéticos.

Um raspberry pi 2 será o responsável por energizar através da interface serial o Arduino além de se comunicar com o mesmo também pela interface de comunicação

Outras peças necessárias são jumpers(fios elétricos), leds e cartões magnéticos.

2. Preparando o hardware

Para que os componentes funcionem juntos será necessária uma montagem que segue padrões importantes.

O raspberry deve estar conectado a internet para que possa buscar as chaves autorizadas em um servidor SQL. Uma biblioteca que se conecte em um servidor MySQL também se torna necessária. O interpretador de python deve estar instalado e é recomendável estar em sua versão 2.x

O arduino vai conter o código descrito na próxima sessão. O mesmo deve estar conectado no raspberry via interface serial. Para isso será necessário um cabo usb. Conecte o arduino através de Jumpers no leitor RFID. Recomendamos uma protoboard para organizar melhor as conexões.

O leitor RFID Usado nesse trabalho deve ser um modelo Rfid Mfrc522 Mifare. Capaz de ler cartões magnéticos a uma frequência de 13,56MHz. Este leitor deve ser

alimentado com 3,3v pelo arduino. Tome cuidado para não conectar um jumper de um dos pinos do arduino na entrada de energia, pois os pinos do arduino tem 5v e a entrada de energia do leitor deve receber apenas 3,3 o que pode acarretar em danos ao leitor.

3. Preparando o Software

Esta seção dedica-se apenas a descrever superficialmente o funcionamento dos códigos c++ e Python que controlam o arduino e raspberry respectivamente. Para mais detalhes consulte os códigos completos no github.

Para controlar o arduino um código c++ deverá ser escrito. Este código deve incluir duas bibliotecas através da instrução `#include`. São elas: a biblioteca `SPI.h` e `MFRC522.h` que controlam a comunicação serial provida pela conexão USB e abstraem o funcionamento do leitor RFID respectivamente.

O código c++ deve definir os pinos que acendem os leds(verde, amarelo e vermelho) como OUTPUT através da função nativa `pinMode()`. Os pinos SS e RST também devem ser configurados para que a biblioteca MFRC possa se comunicar corretamente com o leitor RFID.

No raspberry um código Python será escrito para se comunicar com o código executado no arduino. Este código deverá importar duas bibliotecas: `Serial` para realizar uma conexão com o arduino via interface serial(USB) e `MySQLdb` para realizar a conexão e busca das chaves autorizadas do banco de dados. O código Python precisa aguardar uma chave vinda da interface serial para, só então, checar se essa chave está contida na lista de chaves do banco de dados. Se a chave estiver contida o código deverá enviar(também pela interface serial) a instrução de autorização para o arduino que, por sua vez, irá autorizar a entrada do usuário.

4. Montando e executando

Monte o circuito conectado o Raspberry a periféricos necessários para sua utilização(teclado, mouse, monitor...).

Guarde o código python em uma pasta de sua preferencia no raspberry. Mas apenas o execute quando os próximos passos estiverem concluídos.

Conecte o leitor RFID no arduino através das conexões dos pinos usando os jumpers.

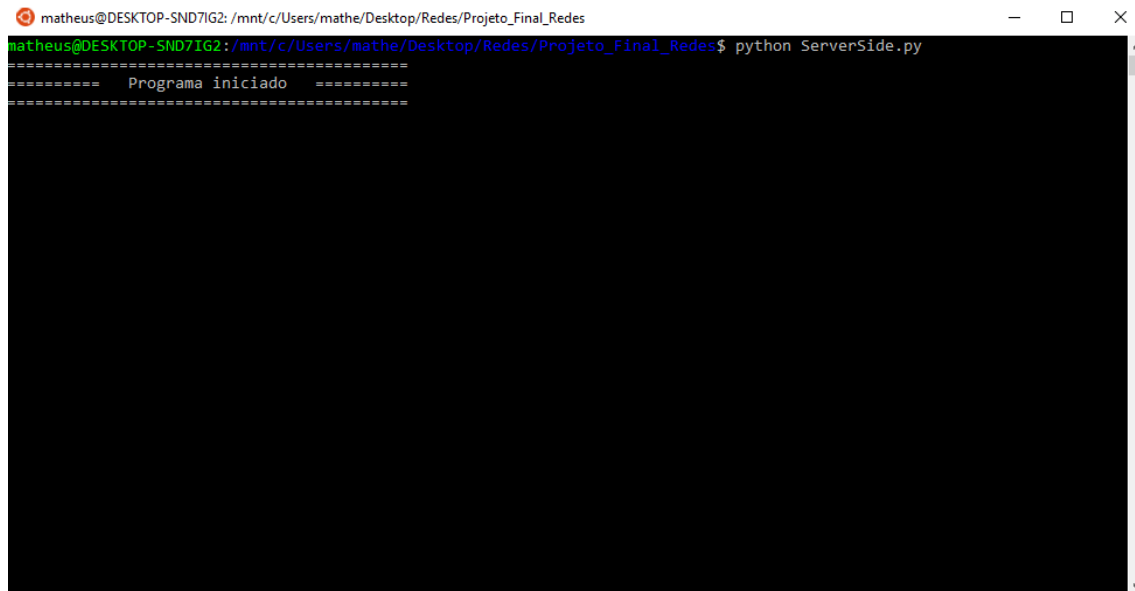
Insira o código c++ através da IDE do arduino e o conecte no raspberry. Após inicializar o arduino execute o código Python no raspberry.

O código no arduino irá se conectar com o raspberry e o sistema estará funcionando.

Na figura 1 é possível observar o código python que, uma vez no raspberry, estará aguardando uma chave ser enviada pelo cabo USB para realizar a autenticação.

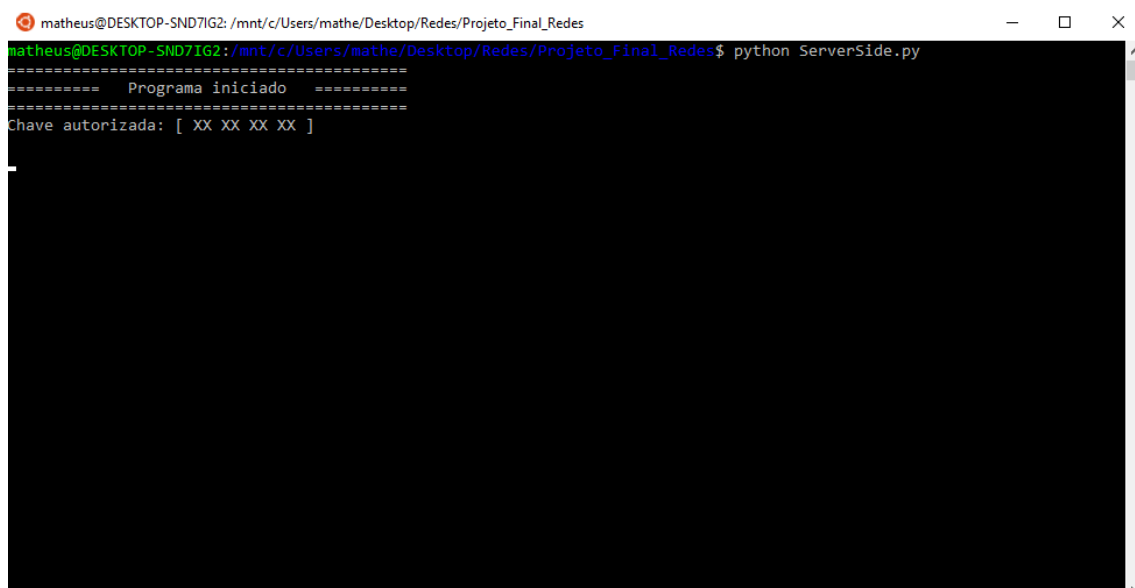
Na figura 2 observe uma chave de teste recebida e o retorno dado pelo código é que a chave está autorizada.

Na figura 3 observe uma chave de teste recebida e o retorno dado pelo código é que a chave não é autorizada.



```
matheus@DESKTOP-SND7IG2: /mnt/c/Users/mathe/Desktop/Redes/Projeto_Final_Redes
matheus@DESKTOP-SND7IG2:/mnt/c/Users/mathe/Desktop/Redes/Projeto_Final_Redes$ python ServerSide.py
=====
Programa iniciado
=====
```

Figure 1. O programa Python



```
matheus@DESKTOP-SND7IG2: /mnt/c/Users/mathe/Desktop/Redes/Projeto_Final_Redes
matheus@DESKTOP-SND7IG2:/mnt/c/Users/mathe/Desktop/Redes/Projeto_Final_Redes$ python ServerSide.py
=====
Programa iniciado
=====
Chave autorizada: [ XX XX XX XX ]
```

Figure 2. Uma chave autorizada

```
matheus@DESKTOP-SND7IG2: /mnt/c/Users/mathe/Desktop/Redes/Projeto_Final_Redes
matheus@DESKTOP-SND7IG2:/mnt/c/Users/mathe/Desktop/Redes/Projeto_Final_Redes$ python ServerSide.py
=====
Programa iniciado
=====
Chave negada: [ XX XX XX XY ]
```

Figure 3. Uma chave negada