

Apresentação da Melhoria de Projeto

RFID DOOR ACCESS

Bruno Lemos 98221
Tiago Marques 98459

Arquiteturas para Sistemas Embutidos
Professor Arnaldo Oliveira

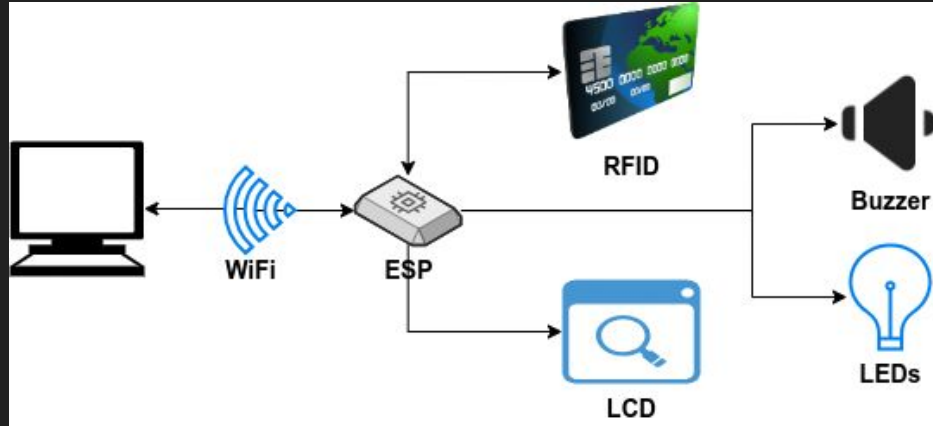
Recap

- Utilizamos a tecnologia de identificação por RFID para controlar o acesso a portas ou entradas específicas.
- Através da leitura da tag RFID, a aplicação pode autorizar ou não o acesso.
- Os registos dos logs e permissões são guardados numa aplicação externa ao esp32
- É usada uma dashboard para permitir ver o estado do sistema e controlar a operação do esp32

Recap - O que era possível

- Adicionar acessos, através da leitura da tag RFID e adicionando-a à base dados da aplicação no PC
- Remover acessos, através da dashboard, eliminando a tag RFID da base dados da aplicação no PC
- Autorizar/Negar acessos, através da verificação da presença da tag RFID na base de dados
- Histórico de operações, logs de operações como:
 - Leitura de cartão com acesso permitido.
 - Leitura de cartão com acesso negado.
 - Adição de cartão à base de dados.
 - Remoção de cartão da base de dados
- Alternar entre o modo de configuração e de leitura do esp32
- Feedback do estado do sistema, através do LCD, LEDs e buzina

Recap - Arquitetura e Comunicações



Config Users Logs Home

💡 Config MODE Write Card Connect

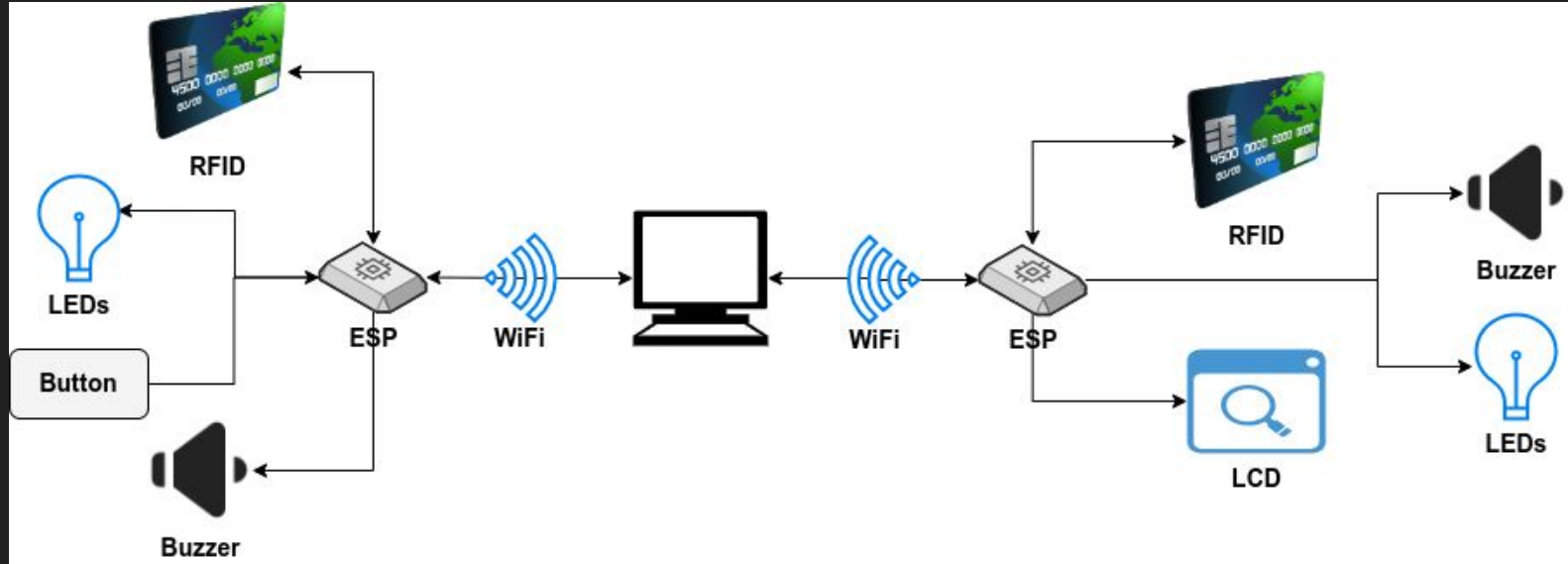
Name	RFID	Time	Action
andante	763221640328	06-06-2023 00h23	Remove
tag	230530204918	05-06-2023 22h07	Remove
SIGA 1230	658377737352	31-05-2023 01h41	Remove
SIGA 1292	38966592648	31-05-2023 01h40	Remove
Cartao unico	305883318121	30-05-2023 23h12	Remove

- Um único esp32 para acessos (autorizar/negar) e configurações (adicionar tags RFID)
- Dashboard usada para ver os logs, ver tags autorizadas (users) e removê-las se preciso, alterar o esp32 entre modo de leitura ou de configuração
- RFID usa o protocolo I2C
- LCD usa o protocolo SPI
- Configurado um servidor HTTP no esp32 para comunicar com Wi-Fi para a dashboard no PC
- É usado um servidor SNTP para obter o tempo em que foram realizadas as operações
- Usada a UART para a introdução das credenciais para aceder à Wi-Fi

Melhorias

- De forma a aproximar o projeto a um sistema real, adicionamos um segundo esp32 com o papel de nó de controlo
- É possível agora também remover um cartão através da leitura da sua tag RFID pelo novo esp32 'central' adicionado
- Novo modo de operação no qual não é permitido ler cartões, é um modo de bloqueio
- Alterações necessárias à dashboard de forma a implementar as alterações
- É possível adicionar mais esp32 para realizar leituras, podendo ter vários

Melhorias - Arquitetura



- 2 esp32:
 - Um para configurações de acessos (adicionar e remover tags RFID)
 - LED para mostrar o modo (adicionar ou remover)
 - Um para leituras de acessos (autorizar e negar acesso)
 - Output no LCD e nos LEDs

Melhorias - Dashboard


DOOR RFID MANAGER

Config

Users

Logs

Home

 **Config MODE**

Write Card

Connect

Name	RFID	Time	Action
Bruno Lemos	38966592648	28-06-2023 18h20	<div>Remove</div>
Tiago Marques	658377737352	28-06-2023 18h20	<div>Remove</div>
CEO	763221640328	28-06-2023 18h20	<div>Remove</div>
Cartao único	305883318121	28-06-2023 18h20	<div>Remove</div>

10.112.40.122

Disconnect

10.112.40.123

Disconnect

O que fizemos

- led
- lcd
- buzzer
- server
- uart
- api

Adaptamos

- rc522
 - disponível com I2C e SPI
 - no nosso código adaptamos para apenas usarmos SPI

<https://github.com/abobija/esp-idf-rc522>

Check list

Os requisitos obrigatórios do mini-projeto são os seguintes:

- - Utilizar o ESP32DevKitC como base do embedded system : Feito
- - A aplicação a executar no kit ESP32DevKitC deve ser desenvolvida em C/C++ e tirar partido do FreeRTOS: Feito
- - Devem ser explorados os periféricos do ESP32 que fizerem sentido no contexto do projeto, incluindo aspetos de interrupções e DMA: Feito
- - Os dados recolhidos do sensor e processados no ESP32 devem ser apresentados num dashboard remoto, sendo para tal necessária conectividade de rede (WiFi / BT): Feito
- - Deve ser disponibilizada uma ligação por Terminal; independente do dashboard remoto: Feito
- - Devem ser exploradas as várias funcionalidades das ferramentas de desenvolvimento, incluindo debug: Feito

Os aspetos opcionais a incluir no mini-projeto são os seguintes (não sendo uma lista fechada):

- Podem ser explorados os modos de baixo consumo energético do ESP32: Não era possível devido ao projeto
- Podem ser suportadas atualizações remotas (Over-the-Air) do sistema: Não
- Pode ser incluído algum tipo de atuador cuja utilização faça sentido com o sensor usado (de forma a criar um loop de controlo; ou que seja controlado através do dashboard): Feito
- Pode ser suportado um sistema de ficheiros para armazenar dados localmente: Não
- Dados persistentes: Feito
- É escalável, podendo adicionar mais esp32 (tanto nó de controlo como de acesso a portas): Sim
- É possível desativar todas as funções de um esp32 remotamente e consequentemente ativá-lo remotamente: Sim

Divisão de trabalho

Bruno Lemos - 50%

Tiago Marques - 50%

A photograph of an Arduino Uno microcontroller board on a breadboard. The board is populated with various components: a black integrated circuit (likely a motor driver), a small black component (possibly a sensor or relay), and several colored jumper wires (red, blue, yellow, green, white) connecting different pins. A 16x2 LCD display is connected to the bottom of the breadboard. The display shows the text "DOOR ACCESS" on the first line and "Status: scan" on the second line. The word "DEMO" is overlaid in large white letters in the center of the image.

DEMO

DOOR ACCESS
Status: scan