

Instituto Superior de Engenharia de Lisboa

Sistemas Embebidos II

Projeto – Fechadura/Controlo de acessos eletrónico

Ivo Pereira - 42172

Docente:

Pedro Sampaio

Índice

Introdução	.3
Esquema do projeto	.4
Periféricos	
Sistema Operativo	.4
Máquina de estados	.5
Utilização de memoria não volátil	.5
App Memory Managment	.6
Estruturas de dados	.6
Funcionalidades	.6
Inicialização da Aplicação	.6
App	.7
Leitura de Cartões	.7
Validar Cartões	.7
Registar acessos	.7
-	
App Leitura de Cartões Validar Cartões	

Introdução

Este projeto tem como objetivo a criação de um sistema de controlo de uma fechadura. O sistema é baseado no microcontrolador da NXP LPC1769, e tem como periféricos um display LCD, um teclado, um leitor de cartões wireless, uma EEPORM e um módulo de wifi ESP.

As chaves da fechadura são cartões que são lidos pelo leitor de cartões wireless e de seguida validados pelo sistema de controlo. O LCD fornece ao utilizador uma interface com o sistema, mostrando ao utilizador se o cartão foi ou não validado. Para além disso esta também presente um modo de manutenção, acedido e navegado pelo teclado, que permite a opção de adicionar um cartão como chave e a opção de verificar os registos de acesso. Para aceder a esse modo é também necessário a presença de um cartão que esta definido no sistema como cartão com permissões para manutenção.

O sistema tem uma data e hora que são mostrados no LCD e guardados como informação nos registo de acesso. Esta hora e data de sistema são adquiridos na inicialização do sistema através de um pedido em rede no protocolo NTP.

É utilizada uma biblioteca (SE1819) criada e documentada anteriormente que permite o controlo dos periféricos mencionados, assim como de alguns já persentes no chip.

Esquema do projeto

Periféricos

Para além dos periféricos já presentes no LPC como o RTC, SysTick, GPIO, SPI, I2C e UART, estas quatro últimas que servem para fazer interface com outros periféricos externos; utilizamos também como periféricos externos um LCD, um teclado 4x3, um leitor de cartões, uma EEPROM e um modulo de *wifi* ESP.

O esquema elétrico da montagem encontra-se na documentação do projeto com todos os pins usados e conceções entre periféricos e microcontrolador. Mas são também necessários controladores para estes periféricos. Foi usada a biblioteca SE1819 para controlar estes periféricos.

Foram usadas entradas e saídas GPIO para a ligação entre microcontrolador e teclado, assim como LCD; para o leitor de cartões a interface é baseada em SPI; Para a EEPROM o I2C; e para o modulo *wifi* ESP a UART.

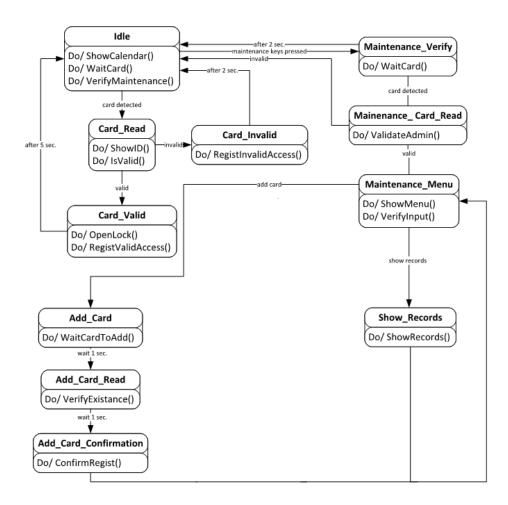
Para informações mais detalhadas sobre a interface e controlo dos periféricos a documentação da biblioteca SE1819 pode ser consultada.

Sistema Operativo

Foi utilizado também um sistema operativo FreeRTOS que permite tirar partido de *multitasking*. O projeto utiliza 3 *tasks* diferentes que comunicam através de objetos do sistema operativo. Uma *task* para o *keypad* que está bloqueada num semáforo e é desbloqueada quando uma tecla for pressionada, descobre que tecla foi pressionada e coloca o código dessa tecla numa fila que pode ser então acedida por outras *tasks* para obter as teclas pressionadas. Uma *task* para o display que recebe numa fila comandos para mandar ao LCD, interpreta esses comandos e controla o LCD de acordo. E por fim a task principal que controla o modulo ESP para acesso á rede e onde corre a principal logica da aplicação.

Máquina de estados

A aplicação deste projeto, apos a inicialização, segue uma máquina de estados que representa o funcionamento e o fluxo da aplicação.



Todos estes estados estão então definidos num *header file* e a aplicação navega através deles tendo uma variável global que define o estado atual e alterando esse estado quando as condições para tal se verificarem.

A aplicação corre num ciclo infinito que contem um *switch case*, cada caso representa um estado.

Utilização de memoria não volátil

É necessária a utilização de memoria persistente para guardar os Ids dos cartões que têm acesso à fechadura assim como Ids dos cartões que têm acesso à manutenção e por fim também dos registos de acesso à fechadura. Para guardarmos estes dados utilizamos então uma EEPROM que é acedida através do protocolo I2C.

App Memory Managment

Para o controlo da memoria presistente e definição de como esta será usada criamos um módulo de controlo de memória. Este modulo utiliza o modulo E2P da biblioteca SE1819 para manipular dados.

Estruturas de dados

São guardados na EEPROM três estruturas de dados. Uma primeira estrutura que é a estrutura de controlo e duas outras estruturas que são *arrays*. Um *array* guarda Ids de cartões que têm acesso à fechadura. O outro *array* guarda estruturas que representam os registos de acesso, que têm um Id de cartão, se a abertura foi válida ou não, a data e hora do acesso. A primeira estrutura de controlo guarda os índices de ambos os *arrays* assim como 4 *slots* para Ids de cartões que têm acesso à manutenção.

Funcionalidades

Este modulo disponibiliza sete funções à aplicação. Duas destas funções são para adicionar dados aos dois arrays de estruturas de dados, AddRecord e AddLockCard. Nestas funções são escritos os dados na EEPROM e é também incrementado o índice da respetiva estrutura no sector de controlo.

Outra função disponibilizada é uma função de configuração SetupControl. Esta função é para ser utilizada a primeira vez que o sistema for utilizado. E cria a estrutura de controlo com os índices a zero e os cartões que têm acesso à manutenção e guarda-a na EEPROM.

Duas outras funções são de verificação de cartões, CardExists e AdminCardExists, que retornam se um certo cartão tem acesso à fechadura ou ao modo de manutenção respetivamente.

E por fim, as últimas duas funções dão suporte ao acesso dos registos. A primeira GetRecordsLength que retorna a dimensão do *array* de registos; e GetRecord, que dado um índice retorna o registo desse índice.

Inicialização da Aplicação

Antes de funcionar corretamente pela primeira vez a aplicação necessita de executar alguns métodos que preparam as suas funcionalidades. Um deles irá correr o SetupControl mencionado anteriormente. O outro método irá executar os métodos de inicialização necessários das bibliotecas utilizadas, nomeadamente a biblioteca SE1819 e a biblioteca MFRC522. E por fim será realizada uma rotina que faz a conexão à rede através do modulo ESP e adquire a hora e data atual. O valor da data e hora é inserido no RTC do microcontrolador através da função RTC_SetValue do controlador do RTC.

App

Leitura de Cartões

Para ler os Ids de cartões que dão acesso à fechadura e ao modo de manutenção utilizamos métodos biblioteca MFRC522, primeiro o PICC_IsNewCardPresent para detetar um cartão e logo de seguida se for detetado um cartão o PICC_ReadCardSerial(Uid uid) que coloca em *uid* o id do cartão lido. Este *uid* é de seguida transformado da estrutura Uid para um inteiro e guardado para poder ser utilizado em diferentes estados se necessário.

A leitura de cartões é realizada para validar o acesso à fechadura, à manutenção e também para adicionar um cartão ao sistema.

Validar Cartões

Para validar um cartão é necessário percorrer a estrutura de dados com cartões que esta em memoria flash e verificar se o Id do cartão presente encontrasse guardado.

Registar acessos

Cada vez que um cartão é apresentado para abrir ou tentar abrir a fechadura é registada esta ação. Uma estrutura de dados que guarda a data e hora, id do cartão e também se foi válido ou não é preenchida e escrita em memoria na estrutura de dados de registos.

Menu de Manutenção

Para apresentar o menu de manutenção a app guarda a *string* de cada uma das opções num *array*. Mostra duas opções de cada vez no LCD utilizando o método LCDText_Printf do controlador de LCD da biblioteca SE1819 e espera o input do utilizador, adquirido através do teclado com o método KEYPAD_Read do controlador de teclado, para percorrer e mostrar outras opções ou para adquirir a opção escolhida. Processando assim a opção mudando o estado da aplicação para o que corresponde à opção escolhida.