Instituto Superior de Engenharia de Lisboa

Sistemas Embebidos I

Projeto – Fechadura/Controlo de acessos eletrónico

Ivo Pereira

João Raimundo

Mariana Bento

Docente:

Pedro Sampaio

06/01/2019

Índice

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 e um leitor de cartões wireless.

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 configuração da data e hora presentes no LCD, assim como opção de adicionar um cartão como chave, e por fim, 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.

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

Interface com periféricos

Para além dos periféricos já presentes no LPC como o RTC, SysTick, GPIO e SPI, estas duas últimas que servem para fazer interface com outros periféricos externos; utilizamos também como periféricos externos um LCD, um teclado 4x3 e um leitor de cartões.

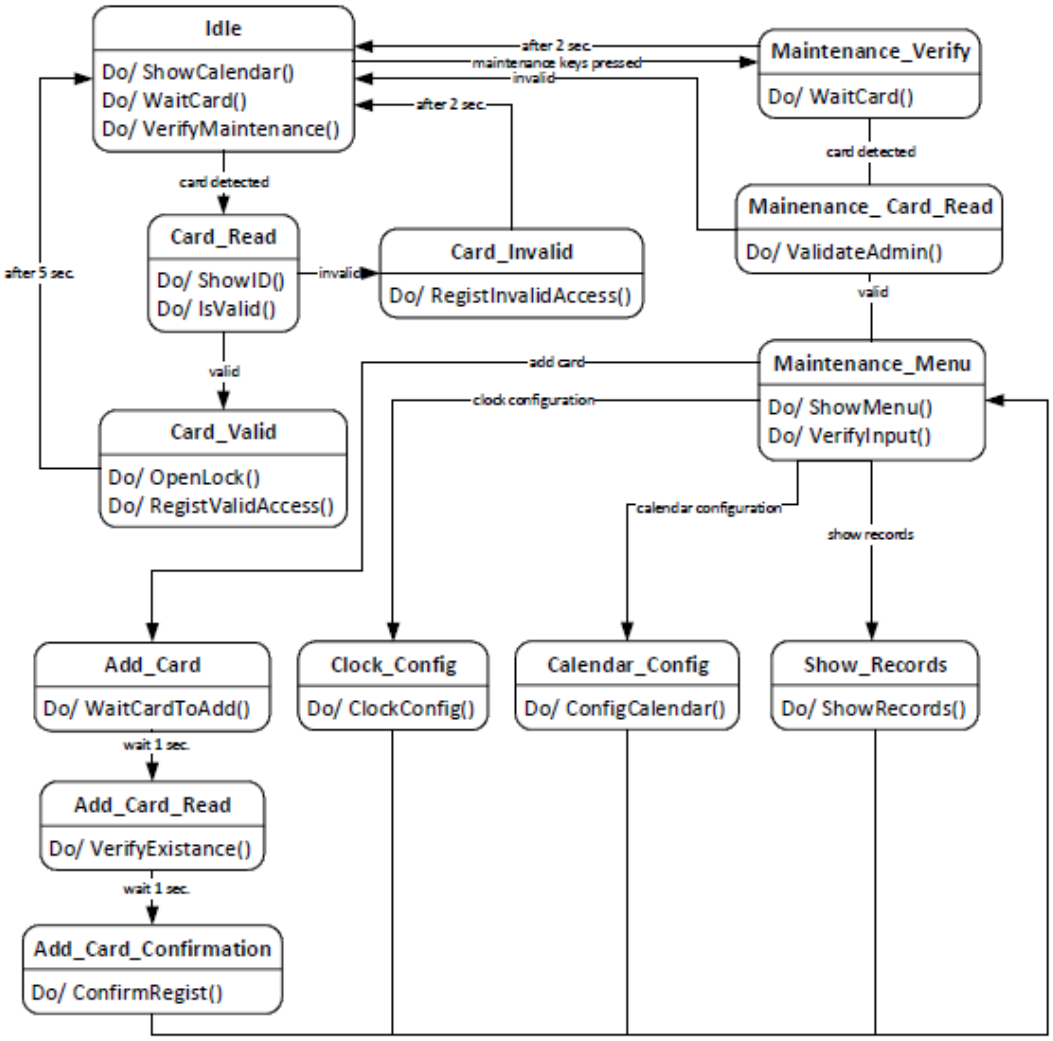
O esquema elétrico da montagem encontra-se na documentação do projeto com todos os pins usados e conceções entre periféricos e microcontrolador. Mas são também necessários controladores para estes periféricos. Foi usada a biblioteca SE1819 para controlar estes periféricos.

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

Máquina de estados

A aplicação deste projeto 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 a memoria Flash disponibilizada no LPC1769.

App Memory Managment

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

Sectores

A memoria Flash do LPC funciona por sectores, foi então definido um sector para guardar as **estruturas de dados com Ids de cartões**, tanto para acesso à fechadura como à manutenção, e um sector para guardar a **estrutura de dados com os registos de acesso** à fechadura.

Apenas é possível escrever em memoria "vazia", e para apagar dados em flash (tornar uma parte de flash "vazia") é necessário apagar todo o setor. Por essas razões e por ser necessário ter em Flash os índices ou tamanhos atuais das estruturas de memoria que estão sempre a incrementar utilizamos também um terceiro **sector de controlo** que contem os índices para as estruturas de dados em memoria.

Do sector de controlo é apenas utilizada uma pequena fração do seu tamanho e cada vez que o valor dos índices guardados é alterado é necessário passar a informação deste setor para RAM, manipular os índices necessários, apagar o sector e voltar a escrever em flash os valores.

Estruturas de dados

São guardados em flash três estruturas de dados que são *arrays* alinhados que alojam os valores de Ids de cartões e os registos de acesso. Dois destes *arrays*, os que guardam Ids de cartões, encontram-se no mesmo sector com tamanhos definidos em app_mem_managment.h, o último que guarda os registos pode ocupar todo um sector.

Funcionalidades

Este modulo disponibiliza quatro funções à aplicação. Três destas funções são para adicionar dados às três estruturas de dados, AddRecord, AddLockCard e AddMaintenanceCard. Nestas funções são escritos os dados em flash e é também incrementado o índice da respetiva estrutura no sector de controlo.

A outra função disponibilizada é uma função de preparação HardResetFlash. Esta função apaga todos os conteúdos nos três sectores utilizados pela aplicação e prepara as estruturas de dados para serem utilizadas. Por fim também adiciona um cartão *default* que já irá ter acesso ao modo de manutenção da aplicação.

Inicialização da Aplicação

Antes de funcionar corretamente pela primeira vez a aplicação necessita de executar dois métodos que preparam as suas funcionalidades. Um deles irá correr o HardResetFlash mencionado anteriormente. O outro método irá executar os métodos de inicialização necessários das bibliotecas utilizadas, nomeadamente a biblioteca SE1819 e a biblioteca MFRC522.

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 flash 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.

Configuração de Relógio e Data

Para a configuração do relógio ou data do sistema a aplicação corre um ciclo com o número de iterações igual ao número de campos em que o índice será representante do campo que o utilizador está a configurar no momento, por exemplo quando o i é 0 representa horas, 1 representa minutos e 2 segundos.

Dentro desse ciclo corre outro ciclo que espera o input do utilizador para cada algarismo do campo alternando entre os dois algarismos até que o utilizador mude para o próximo campo.

Quando o utilizador muda de campo é armazenado e validado o valor do número dado por o conjunto dos algarismos, se for valido é inserido numa estrutura de data o valor desse campo e o ciclo avança para o próximo campo. Se não for validado o ciclo não avança e o utilizador tem de voltar a inserir um valor valido nesse campo.

Por fim o valor da data ou hora é inserido no RTC do microcontrolador através da função RTC_SetValue do controlador do RTC.