

SISTEMA DE CONTROL DE ACCESO BASICO

Casas Bruno, Juan Martin
e-mail: jcasasbruno@frba.utn.edu.ar
Fontalva, Ramiro
e-mail: rfontalva@frba.utn.edu.ar
Pérez del Cerro, Josefina
e-mail: jprezdelcerro@frba.utn.edu.ar

ABSTRACT: Se desarrolló un sistema de control de acceso básico mediante el cual con un teclado matricial se ingresa una clave que se almacena en la memoria flash del LPC1769 y muestra lo escrito en el display LCD.

PALABRAS CLAVE: LCD, teclado matricial, memoria flash.

1 INTRODUCCIÓN

Se solicitó realizar un sistema de control de acceso básico por entrada de clave de 4 dígitos programable por usuario, que funcione con un teclado matricial y un display LCD 2x16. Se utilizó la memoria Flash del LPC1769 para almacenar la clave ingresada.

2 FUNDAMENTACION TEORICA

2.1 HARDWARE

Para poder realizar el proyecto se utilizó

- LPC1769
- Display LCD 2x16
- Teclado matricial 4x4
- Resistencia 1 kΩ
- Resistencia 270 Ω

El LPC1769 es un stick de desarrollo con un microprocesador Cortex M3, el cual es el encargado de correr el sistema operativo FreeRtos. El LPC provee también una memoria flash que se utilizará para persistir la clave aún cuando el dispositivo no esté alimentado.

El display muestra los datos ingresados e informa la validez de estos. Tiene 16 pines, de los cuales se utilizan 6 conectados como outputs al microprocesador en los pines P0_15, P0_16, P0_23, P0_24, P0_25, P0_26, 2 conectados a tierra, 2 alimentados con 5V, 4 no conectados y uno alimentado con 1V para regular el contraste del display. Para esto se utilizaron las resistencias, las cuales construyeron un divisor resistivo que entrega en su nodo $V_o = 5V \cdot 250\Omega / 1250\Omega = 1V$.

Los datos se ingresan por el teclado matricial, el tiene 8 pines, 4 para sus filas (conectados a P0_06,

P0_07, P0_08, P0_09) y 4 para sus columnas (P0_0, P0_01, P0_17 y P0_18).

El proyecto se construyó sobre una placa perforada, alimentada con 5V.

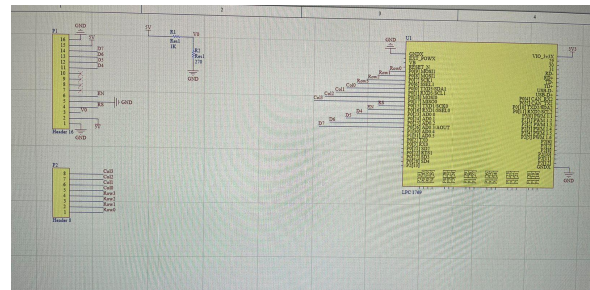


Fig 1. Esquemático del proyecto. Una foto de mayor tamaño puede encontrarse en el apartado 5. Esquemático.

2.2 SOFTWARE

Se utilizó el sistema operativo FreeRtos, el cual permite crear tareas que son administradas por un dispatcher en tiempo real.

Antes de inicializar el scheduler de tareas se hace la configuración del micro, definiendo los pines P0_06, P0_07, P0_08, P0_09, P0_15, P0_16, P0_23, P0_24, P0_25, P0_26 como outputs y los pines P0_0, P0_01, P0_17 y P0_18 como inputs. En la función configuración también se guarda en la variable global clave el valor que se encuentra en la memoria flash del stick.

El código se encuentra compuesto por 3 tareas esenciales, todas con la misma prioridad:

La tarea "Polling", que hace el barrido del teclado matricial, activando secuencialmente las columnas del teclado y comprobando si alguna de las filas se encuentra en estado bajo.

La segunda tarea es "Lcd", la cual espera a que haya un string en la cola queueLcd, en ese momento se desbloquea y muestra por pantalla el valor.

Finalmente la tercer tarea, EscribirEnMemoria, tiene una pequeña diferencia en su creación y es que a parte del nivel de prioridad, se le pasa el macro portPRIVILEGE_BIT, el cual cambia el modo de tarea de User (default) a Privileged. FreeRtos tiene una

MPU (Memory Protection Unit), la cual restringe las direcciones de RAM a las cuales pueden acceder las tareas corriendo en modo User. Para poder escribir en la memoria Flash del LPC1769 se requiere una tarea con modo Privileged ya que esas direcciones de memoria si no quedan en read-only.

La tarea Polling, hace el barrido por filas (las cuales estan definidas como outputs) y por columnas (definidas como inputs), como se mencionó anteriormente, y si encuentra en estado bajo una de las filas se mapea el valor de la columna y fila actual a la constante matriz, una matriz de 4x4 que equivale a los botones del teclado.

En el caso de que se haya presionado cualquier botón que no sea A, B ni C en el teclado, este aparece en el display Lcd. Si en cualquier momento se presiona B se borra la clave que se estaba escribiendo.

Al haber escrito 4 caracteres el usuario puede presionar A para validar lo escrito contra la clave guardada en memoria y el display dirá “Clave Correcta” o “ClaveIncorrecta” según corresponda. Si en cambio el usuario presiona C, el valor que se había ingresado se guarda como la nueva clave, tanto en memoria como en tiempo de ejecución.

Se implementó también una función debounce que hace que solo se pueda ingresar un carácter por vez que se presiona un botón del teclado, donde se guarda el último estado de cada uno de los botones y solo valida que se presionó un botón si el estado anterior en el que estaba era diferente.

Para el manejo de la memoria flash se utilizó código de la biblioteca iap.h que permite reservar y leer espacio en las direcciones de memoria deseadas. En este caso se utilizó el sector 16 de memoria, empezando en 0x00010000.

3 ALCANCE LOGRADO

Se logró ingresar el código mediante el teclado matricial, almacenar la clave en la memoria flash y mostrar la validación y código en el display LCD.

Además, se agregó funcionalidad para que el usuario pueda definir la clave que desee y también cambiarla. Esta clave persistirá en memoria aún cuando el dispositivo se desconecte de la alimentación.

4 DIAGRAMA EN BLOQUES

En la figura 2 se explica la lógica básica del software del proyecto.

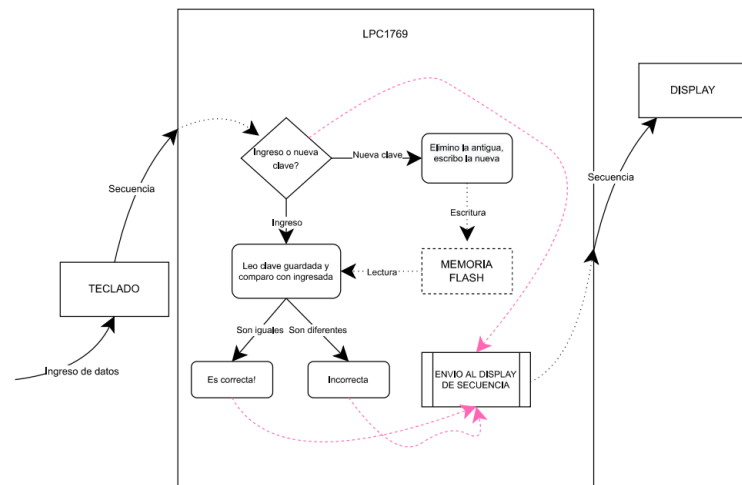


Fig 2. Diagrama de estados de la lógica de funcionamiento

5 ESQUEMÁTICO

