LED

Source:https://en.wikipedia.org/wiki/Light-emitting_diode

Library Overview

The LED Control Library provides a simple interface for controlling two status LEDs used in the RFID Security Access System. The library manages red and green LEDs to provide visual feedback about system state, including pin entry status and successful RFID card scanning.

Key Features

- **Dual LED Control:** Independent control of RED and GREEN status LEDs
- **Visual Feedback:** Clear indication of system states (locked/unlocked/success)
- **Simple API:** Easy-to-use on/off functions for each LED
- **Low Power:** Uses current-limiting resistors for safe LED operation
- **Direct GPIO Control:** Fast, efficient LED switching

Use Cases

- Pin entry status indication (red LED)
- Successful card scan confirmation (green LED)
- System state visualization
- User feedback in access control systems
- Status indication for embedded applications

Function Summary

Function Name	Description

initLEDs(void)	Configures the LED pins as outputs and establishes the initial state with the Red LED on and Green LED off. Just activating the TRISB bits for RB5 and RA4 set as outputs
	Input: none
	Return: none
	Usage:
	  int main(void) {
	  initLEDs(); // Call once during system initialization
	
	  // RED LED is now ON, GREEN LED is now OFF
	
	  while(1) {
	  // Main program loop
	 }
	}

redLED_ON(void)	Activates the Red LED (RB5) to visually indicate that the system is locked and waiting for pin entry. By setting LATBto high
	Input: none
	Return: none
	Usage:
	 // at system startup
	 redLED_ON(); // indicate system is locked
	
	 // after wrong pin entry
	 if (!validatePIN()) {
	   redLED_ON(); // keep LED on (still locked)
	 }
	
	 // after reset
	 resetSystem();
	 redLED_ON(); // ready for next user

redLED_OFF(void)	Deactivates the Red LED (RB5) to signal that the correct pin has been entered by setting LATB to low.
	Input: none
	Return: none
	Usage:
	  // after correct pin entered
	  if (getPinAccepted()) {
	 redLED_OFF(); // Turn off RED LED (unlocked)
	 }
	
	 // during protected operation
	 if (getPinAccepted()) {
	 redLED_OFF();
	 // allow RFID scanning
	}

greenLED_ON(void)	Activates the Green LED (RA4) to visually indicate a successful RFID card scan. By setting LATA to high
	Input: none
	Return: none
	Usage:
	// after successful card scan
	if (rfid_read_card_uid(cardID) == 0) {
	lcd_printStr("Access Granted!");
	
	greenLED_ON(); // turn on green led
	__delay_ms(3000); // keep on for 3 seconds
	greenLED_OFF(); // turn off after indication
	}

greenLED_OFF(void)	Deactivates the Green LED (RA4) to end the successful scan indication. LATA set to low
	Input: none
	Return: none
	Usage:
	// after success indication period
	greenLED_ON();
	__delay_ms(3000); // Display for 3 seconds
	greenLED_OFF(); // Turn off after indication
	
	// during system reset
	resetSystem();
	greenLED_OFF(); // ensure green LED is off
	redLED_ON(); // turn red LED back on

Wiring Diagram

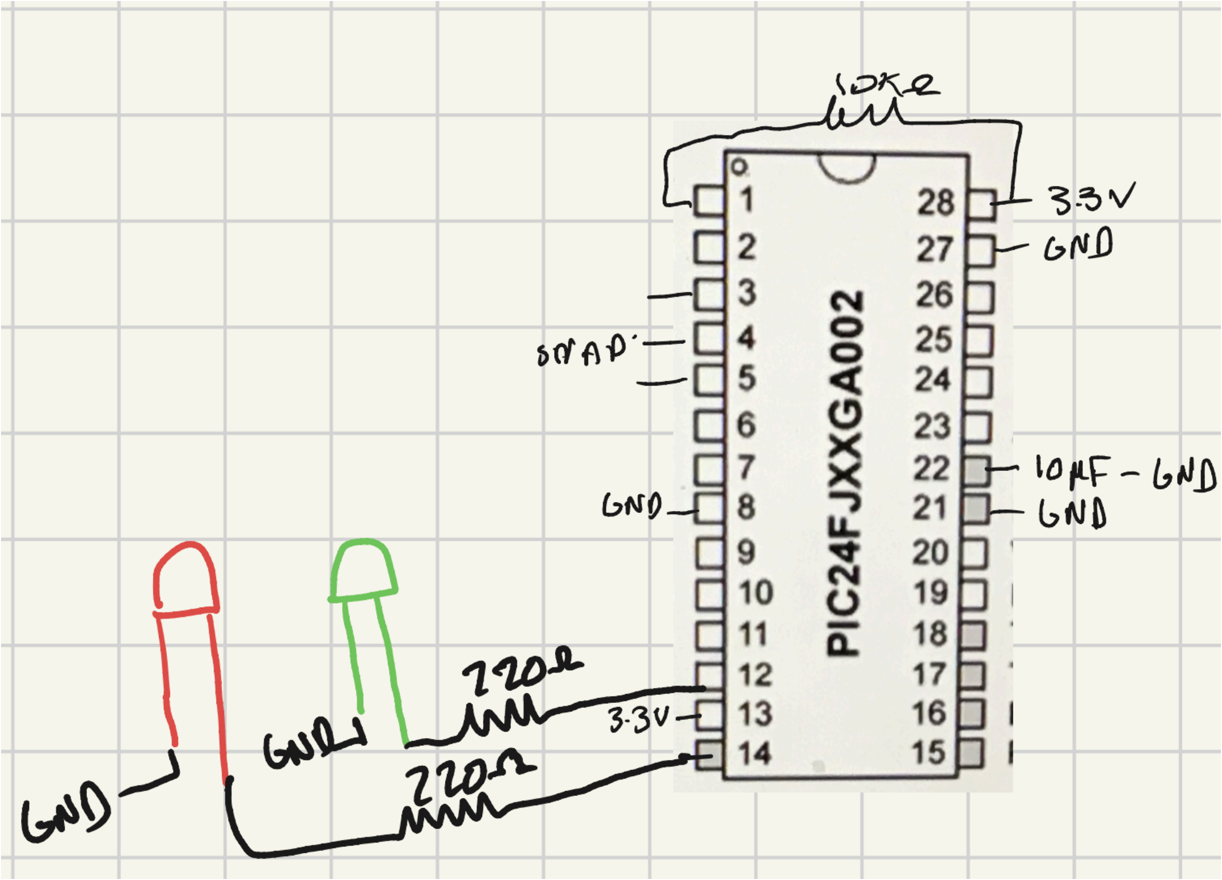

Figure 1: Configuration for the led_control feedback diagram

Notes: only external component need was the 220Ω resistors and the pins used for the led were RB5(red LED) and RA4(green LED)

Code

Source file

```
/*
 * File: led_control.c
 * Author: Alamin Suliman
 * Lab: RFID Security Access System
 * Description: LED control functions for visual feedback red LED(RB5) indicates waiting for correct pin while green LED(RA4) indicate sucessful RFID card scan
 */
```

```
#include "xc.h"
#include "led_control.h"
#define FCY 16000000UL
#include <libpic30.h>               //for tghe __delay_ms() function
#define RED_LED LATBbits.LATB5       //define red led pin (RB5)
#define GREEN_LED LATAbits.LATA4     //define green led pin (RA4)

/*
 * Function: initLEDs
 * Description: initializes led pins as outputs and sets initial states
 * red led starts on (waiting for pin) and green led is off
 */
void initLEDs(void)
{
    TRISBbits.TRISB5 = 0;           // Configure RB5 (red led pin) as output
    TRISAbits.TRISA4 = 0;           // Configure RA4 (greeb led pin) as output
    greenLED_OFF();                 // make sure green ledis off initially
    redLED_ON();                    // turn on red led to indicate waiting for pin

}

/*
 * Function: redLED_ON
 * Description: turns on the red led (RB5)
 * red led indicates system is locked and waiting for correct pin entry
 */
void redLED_ON(void)
{
    RED_LED = 1;                    // Set RB5 high

}

/*
 * Function: redLED_OFF
 * Description: turns off the red led(RB5)
 * red led turns off when correct pin is entered
 */
void redLED_OFF(void)
{
    RED_LED = 0;                    // Set RB5 low

}

/*
 * Function: greenLED_ON
 * Description: turns on the green led (RA4)
 * green led indicates successful RFID card scan
 */
void greenLED_ON(void)
{
   GREEN_LED = 1;                  //et RA4 high

}

/*
 * Function: greenLED_OFF
 * Description:turns off the green led (RA4)
 * green led turns off after indication period
 */
void greenLED_OFF(void)
{
   GREEN_LED = 0;                  //set RA4 low

}
```

Header file

```
/*
 * File: led_control.c
 * Author: Alamin Suliman
 * Lab: RFID Security Access System
 * Description: Header file for LED control functions
 */

#ifndef LED_CONTROL_H
#define LED_CONTROL_H

#include <stdint.h>

void initLEDs(void);
void redLED_ON(void);
void redLED_OFF(void);
void greenLED_ON(void);
void greenLED_OFF(void);


#endif // LED_CONTROL_H
```