Lab 3 Joystick and LED Programming With CubeMX

Lab Report

Student: Jeremiah Webb

Student ID: 2545328

Instructor: Dr. Jianhua Liu

Section #2

Introduction

This lab was used to understand and program a Joystick in Keil and understand how to set up pins and the functions that affect those pins in CubeMx. This lab shows the students the correlation of programming in C, and programming based on already generated code from CubeMX. This baseline to edit LEDs allows students to advance into more intricate programs later using the pins edited in CubeMx.

Code Snippets

```
#include "usr_tasks.h"
#include "stdbool.h"
// 2 seconds = 2000 ms
///////////////////////////////////////////////

#if defined(STM32L476xx)
#define LED_ON GPIO_PIN_SET
#define LED_OFF GPIO_PIN_RESET
#elif defined(STM32F412Zx)
#define LED_ON GPIO_PIN_RESET
#define LED_OFF GPIO_PIN_SET
#endif

//Learned what Macros are by Troy Neubauer, extremely helpful, made this a
lot simpler
//Macro to receive information of pin, put into Hal function
#define ReadPin(pin) HAL_GPIO_ReadPin(pin##_GPIO_Port, pin##_Pin)
//Macro to Write a specific thing to the pin, as received from the if
statements below
#define WriteToPin(pin, value)HAL_GPIO_WritePin(pin##_GPIO_Port, pin##_Pin,
value)
//Macro to Toggle pin, depending on the input using Hal Function
#define TogglePin(pin) HAL_GPIO_TogglePin(pin##_GPIO_Port, pin##_Pin)
void USR_Task_RunLoop(void) {
      uint8_t up = ReadPin(JOY_U);
      uint8_t center = ReadPin(JOY_C);
      uint8_t down = ReadPin(JOY_D);
  uint8_t left = ReadPin(JOY_L);
  uint8_t right = ReadPin(JOY_R);
```

```
      //if up is pressed, Green on, red On
       if (up) {
    WriteToPin(LD_R, LED_ON);
    WriteToPin(LD_G, LED_ON);
  }
       //If down is pressed, both are off
       else if (down) {
    WriteToPin(LD_R, LED_OFF);
    WriteToPin(LD_G, LED_OFF);
  }
       //If Center is pressed, both turn on and off every 2 seconds
       else if (center) {
    WriteToPin(LD_R, LED_ON);
    WriteToPin(LD_G, LED_ON);
    for (uint32_t i = 0; i < 10; i++) {
      TogglePin(LD_R);
      TogglePin(LD_G);
      HAL_Delay(200);
    }
    // Toggle Green to begin alternation
    TogglePin(LD_G);
            }
       //If right is pressed, Red is off, green is on for two seconds
            else if (right) {
    WriteToPin(LD_R, LED_OFF);
    WriteToPin(LD_G, LED_ON);
            }
            //If Left is pressed, Red is on for two seconds, green is off
            else if (left) {
            WriteToPin(LD_R, LED_ON);
    WriteToPin(LD_G, LED_OFF);
            }
            else {
    TogglePin(LD_R);
    TogglePin(LD_G);
    HAL_Delay(100);
  }
      //Needed some help with this one, Wouldn't work with the
Hal_Delay(2000), this seemed to work, closest I could get it to
  if (up || down || left || right) {
    HAL_Delay(1999);
    WriteToPin(LD_R, LED_ON);
    WriteToPin(LD_G, LED_OFF);
    HAL_Delay(1);
  }
}

//Turn Red on, Green off.

void USR_Task_LD_R_on_LD_G_off(void) {

  HAL_GPIO_WritePin(LD_R_GPIO_Port, LD_R_Pin, GPIO_PIN_SET);

  HAL_GPIO_WritePin(LD_G_GPIO_Port, LD_G_Pin, GPIO_PIN_RESET);

}
```

Questions

What board you are using? What mode is being used to drive the LEDs
(e.g. Push-pull or Open-drain)?

I am using the STM32F412 board. The mode being used is Pull-Down.

Can we change the output mode above to the other mode? Why or why not?

We cannot change the mode of the LEDs, using Pull down allows us to connect the signal to ground and have a default state of 0 with the signal is floating. When connected to High, the pull down will be overridden, and the input pin will read a 1.

Narrative

Overall, in the lab, I learned how to use macros to my benefit. Additionally, learning how to use CubeMx more in-depth and being able to physically "see" the programming onto the board is interesting. Understanding the objectives of the lab was good to see that as a class we are going down the rabbit hole of what is microprocessor systems and how to program them. Seeing how C programming can impact the physical shows visualization and importance of the skills we are learning are.

Results

Now I can understand how to use pins from the STM32 chip to do specific functions and use CubeMX to properly implement the base for them. Using CubeMx in this basic example has helped visualize how pins and the functions that use them are implemented in C code. With the help of previous debugging skills, the lab objectives of the joystick functions are doable.