Embedded Systems (CSCE 4114)

Lab 6

Zack Fravel

Kaelon Scott

11/14/16

zpfravel@uark.edu

ksscott@uark.edu

Abstract

The purpose of our final main lab project was to get in groups of two and design a C++ program to run on the spider that mimics the functionality of the second lab we completed, the shifting LED's. We successfully designed a C++ program to be loaded on the spider that shifts through the LED's at 8Hz and is able to increase to 16 Hz or decrease to 1 Hz stepping up/down the integer values.

Introduction

In order to design a C++ program to run on the spider, we are taking advantage of the spider manufactures' pre existing libraries and functions that are built for basic functions. The lab walked us through setting up the C++ development environment, which involved connecting the spider to the network as well as connecting the spider to the computer through the UART connection at a baud rate of 115,200 bits/sec. Once we set up this environment, we're able to make changes to our main.cpp, upload it to the spider, and run it by sending a command through the console (./hello_world).

Design and Implementation

The program we are aiming to implement has a couple of features we have to meet. First, the program should initialize the LED's to shift at a speed of 8 Hz. So, in our design we declare a "hertz" integer variable and set it to 8. Along with this, we declare a few other variables (period, count, shift) that are going to help us make the shift calculations later on in the program. The logic is all contained in a while(1) loop, to ensure the program runs until a certain action is taken to stop the program.

The functionality of our program is split up into two main blocks within the while loop. The first block is checking whether or not the buttons are being pressed within an if statement and sets the values of "hertz" accordingly. Below that, we set the period of the shifting using the "hertz: we just determined with the push button. The final part of the program is a case statement contained in an if statement that shifts the LED's continuously at the rate determined by the period. The program worked successfully and was demonstrated in lab.

Results

I have included a screenshot of the main.cpp code we used to implement the desired functionality. The program does precisely what the lab laid out in its description, it shifts through LED's at a continuous rate which can be changed by the pressing of either of the push buttons.

```
#include <stdio.h>
#include <stdint.h>
#include "PIO_LED.h"
#include "PIO_BUTTON.h"
#include <time.h>


int main(int argc, char *argv[]){

    CPIO_LED LED_PIO;
    CPIO_BUTTON BUTTON_PIO;

        /* Hello World! */
        printf("Hello World!\r\n");

        int period = 0;
        int count = OS_GetTickCount();
        int shift = 0;
        int shiftMod;
        int hertz = 8;
        int old_buttons, buttons;

        // run program until suspended
        while (1){

                buttons = BUTTON_PIO.GetBUTTON();

                if (old_buttons != buttons){
                        if ((BUTTON_PIO.GetBUTTON() & 0x01) == 0){
                                if (hertz > 1)
                                hertz--;
                        } else if ((BUTTON_PIO.GetBUTTON() & 0x02) == 0) {
                                if (hertz < 16)
                                hertz++;
                        }
                }

                old_buttons = buttons;

                period = OS_TicksPerSecond() / hertz;

                if (OS_GetTickCount()-count > period) {
                        shift++;
                        shiftMod = shift % 8;
                        switch (shiftMod){
                                case 0: LED_PIO.SetLED(0x01);
                                        break;
                                case 1: LED_PIO.SetLED(0x02);
                                        break;
                                case 2: LED_PIO.SetLED(0x04);
                                        break;
                                case 3: LED_PIO.SetLED(0x08);
                                        break;
                                case 4: LED_PIO.SetLED(0x10);
                                        break;
                                case 5: LED_PIO.SetLED(0x20);
                                        break;
                                case 6: LED_PIO.SetLED(0x40);
                                        break;
                                case 7: LED_PIO.SetLED(0x80);
                                        break;
                        }
                        count = OS_GetTickCount();
                        printf("shift %d\r\n",shiftMod);
                } else { period = 0; }
        }

        return 0;
}
```