NAMES:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ECE-C304 Lab Assignment # 5

(5A,5B,5C due 9th Week)

Submit code for all parts

*5A* (LCD Character String Display)

Using sleep timer interrupt, set up a shifting display of “ROBOTS” on the LCD. The display should start on Row 0 , Col 0 and move rightward on Row 0 as far as possible without losing any letters and then shift to Row 1, Col 0. It should then move rightward on Row 1 as far as possible and then move back to Row 0, Col 0 and repeat the cycle.

Initials\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_ Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5B (GPIO Interrupts, PWM and Bar Graph)

The objective is to use P0[0] as a GPIO pin , generate interrupts through this pin and let the GPIO ISR increment PWM pulse width and also display the current pulse width in the form of a horizontal bar graph on the LCD. When the Pulse width reaches 255, it should be reset to 0.

5B-1

1. Create new project 5B with C main
2. Connect P0[0] to the switch and P1[4] to one of the LEDs
3. Place a PWM8 and connect its Compare Out to P1[4] as done in the previous lab
4. Place a LCD as done in the previous lab and assign Port 2 to it
5. Select settings for P0[0] as follows:

* Mode StdCPU ,Drive Pull Down, Interrupt on Rising Edge, Initial Value 0

1. Locate GPIO interrupt vector location in the boot.tpl file and write one line assembly code to increment Pulse Width by 5 for each interrupt generated
2. Write C main to start PWM and LCD as also enable GPIO interrupts on P0[0]
3. In the control loop apply the current Pulse Width to the PWM and also display it as a horizontal bar graph

Initials\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_ Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5B-2

In this part you should not use the GUI to set up rising edge interrupts on P0[0] , to set up its drive mode etc. Use the appropriate registers to do all this

Initials\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_ Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5B-3

In this part, when the Pulse Width reaches 255, it should start decrementing by 5 , rather than reset to 0. A one line ISR cannot do this. Write a C language ISR to achieve this functionality and place a ljmp at the GPIO interrupt vector location

Initials\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_ Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5C-1 Using UART to transmit data from the Microcontroller to the host PC

In this part, you will have to modify the code developed in 5B-3 to display the pulse width on the Hyperterminal window of the host PC in addition to the LCD. To transmit the data to the PC, a UART transmitter is set up at 115,200 bps. (Study the datasheet of the UART and get familiar with the APIs to start the transmitter, send a byte over it and so on). The clock for the transmitter is generated by a PWM (named BaudClock) whose clock source, period and pulse width are adjusted to generate a waveform at 8\*115,200 Hz with duty cycle 50% (calculate these parameters assuming the clock source for Baudclock is SYSCLK\*2)

Procedure

1. Make a copy of the project used for part 5B-3 and name it 5C
2. Delete the PWM and the code associated with it
3. Place a PWM8 in DBB01 and name it BaudClock and set its parameters as calculated
4. Connect its output to Broadcast Row 0
5. Place a UART transmitter TX8 in DCB02 . Name it TX and parametrize as follows:

Clock – Broadcast Row 0

Output – Row 0 Output 3

Clock Sync – Sync to Sysclk

1. Connect Row 0 Output 3 to P2[7] thru global interconnect
2. In the main program start TX and in the ISR add code to transmit pulse width followed by CR+LF
3. On the board, connect P2[7] to TX on the 4-pin serial connector and connect COM1

connector of the PC to the 9-pin port on the board

1. Build and generate the application and power on the board (You can use high level APIs)
2. Open Hyperterminal on the PC, select the correct COM port and set the baud rate to 115,200.
3. Every time you press the switch you will see pulse width displayed as a bargraph on the LCD . At the same time, the hex value of the pulse width will be displayed in the Hyperterminal
4. As you keep pressing and releasing the switch, in the Hyperterminal capture the data and save it in .csv format. After one complete cycle 0-255-0, stop the capture
5. Open EXCEL or MATLAB and show the graph of the captured data in the form of a sawtooth wave (0 to 255 and back to 0)

Initials \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5A Code**

#include <m8c.h> // part specific constants and macros

#include "PSoCAPI.h" // PSoC API definitions for all User Modules

#include "LCD.h"

unsigned char row = 0x00;

unsigned char col = 0;

void main(void)

{

LCD\_Start();

LCD\_Init();

LCD\_Position(row,col);

INT\_MSK0 |= 0x40;

M8C\_EnableGInt; // Uncomment this line to enable Global Interrupts

while(1){

M8C\_Sleep;

LCD\_Init();

LCD\_Position(row,col);

LCD\_PrCString("Robots");

col++;

if(col==16){

col = 0;

row ^= 0x01;

}

}

}

**5B-3 Code**

#include <m8c.h> // part specific constants and macros

#include "PSoCAPI.h" // PSoC API definitions for all User Modules

#include "PWM8.h"

#include "LCD.h"

unsigned char pulsewidth = 0x00;

unsigned char pwDir = 0;

void main(void){

LCD\_Start();

LCD\_Init();

LCD\_InitBG(LCD\_SOLID\_BG );

PWM8\_Start();

M8C\_EnableGInt ; // Uncomment this line to enable Global Interrupts

INT\_MSK0 |= 0x20;

PRT0IE |= 0x01;

PRT0DM0 = 0x01; //DM0[0-3] = 1

PRT0DM1 = 0x01; //DM1[0-3] = 0

PRT0DM2 = 0x00; //DM1[0-3] = 0

PRT0IC1 = 0x01;

PRT0IC0 = 0x00;

while(1){

if(pulsewidth==255){

pwDir = 1;

}

PWM8\_WritePulseWidth(pulsewidth);

LCD\_DrawBG(0,0,16,pulsewidth/5);

LCD\_DrawBG(1,0,16,72);

}

}

#pragma interrupt\_handler gpioISR

void gpioISR(void){

if(pwDir == 0){

pulsewidth += 5;

}

else{

pulsewidth -= 5;

}

}

**5C Code**

#include <m8c.h> // part specific constants and macros

#include "PSoCAPI.h" // PSoC API definitions for all User Modules

#include "BaudClock.h"

#include "LCD.h"

unsigned char pulsewidth = 0x00;

unsigned char pwDir = 0;

void main(void)

{

LCD\_Start();

LCD\_Init();

LCD\_InitBG(LCD\_SOLID\_BG );

BaudClock\_Start();

TX\_Start(0x00);

M8C\_EnableGInt ;

INT\_MSK0 |= 0x20;

PRT0IE |= 0x01;

PRT0DM0 = 0x01; //DM0[0-3] = 1

PRT0DM1 = 0x01; //DM1[0-3] = 0

PRT0DM2 = 0x00; //DM1[0-3] = 0

PRT0IC1 = 0x01;

PRT0IC0 = 0x00;

while(1){

if(pulsewidth==255){

pwDir = 1;

}

LCD\_DrawBG(0,0,16,pulsewidth/5);

LCD\_DrawBG(1,0,16,72);

}

}

#pragma interrupt\_handler gpioISR

void gpioISR(void){

if(pwDir == 0){

pulsewidth += 5;

}

else{

pulsewidth -= 5;

}

TX\_PutSHexByte(pulsewidth);

while( !( TX\_bReadTxStatus() & TX\_TX\_BUFFER\_EMPTY ) );

TX\_PutCRLF();

}