

**Main.C**

/\* ========================================

\*

\* Copyright YOUR COMPANY, THE YEAR

\* All Rights Reserved

\* UNPUBLISHED, LICENSED SOFTWARE.

\*

\* CONFIDENTIAL AND PROPRIETARY INFORMATION

\* WHICH IS THE PROPERTY OF your company.

\*

\* ========================================

\*/

#include "project.h"

#include <stdio.h>

#define PUSHED 0

#include "stdlib.h"

char mystring[20];

int count=0;

int Acount = 0;

int main(void)

{

CyGlobalIntEnable; /\* Enable global interrupts. \*/

/\* Place your initialization/startup code here (e.g. MyInst\_Start()) \*/

GLCD\_Start();

A\_Interrupt\_ClearPending();

A\_Interrupt\_Start();

B\_Interrupt\_ClearPending();

B\_Interrupt\_Start();

C\_Interrupt\_ClearPending();

C\_Interrupt\_Start();

D\_Interrupt\_ClearPending();

D\_Interrupt\_Start();

StickButton\_Interrupt\_ClearPending();

StickButton\_Interrupt\_Start();

Backlight\_Write(1);

while(1) {

//This loop doesn't need to do anything

// sprintf(mystring, "%5d", count);

//GLCD\_PrintString(mystring, 10, 10, GLCD\_WHITE, GLCD\_BLACK);

}

}

/\* [] END OF FILE \*/

**INTERUPTS**

**Includes for all:**

#include "project.h"

#include <stdio.h>

#define PUSHED 0

extern int count;

extern char mystring[20];

**A\_Interrupt.c**

while (A\_Button\_Read() == !PUSHED);

count++; //Increment the counter

//Format the string and print to the screen

sprintf(mystring, "%5d", count);

GLCD\_PrintString(mystring, 10, 10, GLCD\_WHITE, GLCD\_BLACK);

//CyDelay(25);

//While that holds and no increment of the counter

while (A\_Button\_Read() == PUSHED);

CyDelay(25);

A\_Interrupt\_ClearPending(); //Needed to make other interrupts work

**B\_Interrupt.c**

//CyDelay(25);

//While the button is pushed increment the counter and print to the screen

//The CyDelay sets how the fast the counter will go up

while (B\_Button\_Read() == PUSHED)

{

count++;//Increment the counter

sprintf(mystring, "%5d", count);

GLCD\_PrintString(mystring, 10, 10, GLCD\_WHITE, GLCD\_BLACK);

CyDelay(10);

}

B\_Interrupt\_ClearPending(); //Needed to make other interupts work

//Decrements the pointer 10 times a second when pressed

while (D\_Button\_Read() == !PUSHED);

//CyDelay(25);

while (D\_Button\_Read() == PUSHED)

{

count--;

//sprintf(mystring, "%5d", count);

//GLCD\_PrintString(mystring, 10, 10, GLCD\_WHITE, GLCD\_BLACK);

CyDelay(10);

}

D\_Interrupt\_ClearPending();

**C\_Interrupt.c**

//Decrements the counter by 1 once when the button is pushed

//Does not update the screen

while (C\_Button\_Read() == !PUSHED);

count--;//Decrement the counter

//CyDelay(25);

while (C\_Button\_Read() == PUSHED);

CyDelay(25);

C\_Interrupt\_ClearPending(); //Needed for other interupts to work

**D\_interrupt.c**

//Decrements the pointer 10 times a second when pressed

while (D\_Button\_Read() == !PUSHED);

//CyDelay(25);

while (D\_Button\_Read() == PUSHED)

{

count--;

//sprintf(mystring, "%5d", count);

//GLCD\_PrintString(mystring, 10, 10, GLCD\_WHITE, GLCD\_BLACK);

CyDelay(10);

}

D\_Interrupt\_ClearPending();

**StickButton\_interrupt.c**

//When the joystick is pushed set the count to 0

//This does not update the screen

while (StickButton\_Read() == !PUSHED);

count=0;

//CyDelay(25);

while (StickButton\_Read() == PUSHED);

CyDelay(25);

StickButton\_Interrupt\_ClearPending();