

main.c

//Evan Garcia

//Professor Bolding

//EE 3280

//April 19,2016

//main.c

#include <project.h>

#include <stdio.h>

//Global count and mystring variables

int Count = 0;

char mystring[20];

int main()

{

GLCD\_Start();

Backlight\_Write(1);

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

//Start my interrupts

A\_Interrupt\_Start();

B\_Interrupt\_Start();

C\_Interrupt\_Start();

D\_Interrupt\_Start();

Joystick\_Interrupt\_Start();

GLCD\_Clear(GLCD\_BLACK);

for(;;)

{

//Stop Flashing Screen

A\_Interrupt\_Disable();

B\_Interrupt\_Disable();

//Update mystring and print it to the LCD

sprintf(mystring,"Count:%5d", Count);

GLCD\_PrintString(mystring, 60, 15, GLCD\_WHITE, GLCD\_BLACK);

A\_Interrupt\_Enable();

B\_Interrupt\_Enable();

}

}

A\_Interrupt.c

//This interrupt shall increment the counter by one, and includes code that updates the display.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Place your includes, defines and code here

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* `#START A\_Interrupt\_intc` \*/

//Access global variables

extern int Count;

extern char mystring[20];

#include <stdio.h>

#include <project.h>

/\* `#END` \*/

extern cyisraddress CyRamVectors[CYINT\_IRQ\_BASE + CY\_NUM\_INTERRUPTS];

/\* Declared in startup, used to set unused interrupts to. \*/

CY\_ISR\_PROTO(IntDefaultHandler);

CY\_ISR(A\_Interrupt\_Interrupt)

{

#ifdef A\_Interrupt\_INTERRUPT\_INTERRUPT\_CALLBACK

A\_Interrupt\_Interrupt\_InterruptCallback();

#endif /\* A\_Interrupt\_INTERRUPT\_INTERRUPT\_CALLBACK \*/

/\* Place your Interrupt code here. \*/

/\* `#START A\_Interrupt\_Interrupt` \*/

//Increment counter

Count++;

//Update mystring and LCD in IRS

sprintf(mystring,"Count:%5d", Count);

GLCD\_PrintString(mystring, 60, 15, GLCD\_WHITE, GLCD\_BLACK);

//Debounce

CyDelay(200);

A\_Interrupt\_ClearPending();

/\* `#END` \*/

}

B\_Interrupt.c

//This interrupt shall increment the counter 10 times per second, and includes code that updates the display.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Place your includes, defines and code here

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* `#START B\_Interrupt\_intc` \*/

//Access global variables

extern int Count;

extern char mystring[20];

#include <stdio.h>

#include <project.h>

/\* `#END` \*/

extern cyisraddress CyRamVectors[CYINT\_IRQ\_BASE + CY\_NUM\_INTERRUPTS];

/\* Declared in startup, used to set unused interrupts to. \*/

CY\_ISR\_PROTO(IntDefaultHandler);

CY\_ISR(B\_Interrupt\_Interrupt)

{

#ifdef B\_Interrupt\_INTERRUPT\_INTERRUPT\_CALLBACK

B\_Interrupt\_Interrupt\_InterruptCallback();

#endif /\* B\_Interrupt\_INTERRUPT\_INTERRUPT\_CALLBACK \*/

/\* Place your Interrupt code here. \*/

/\* `#START B\_Interrupt\_Interrupt` \*/

//Increment counter

Count++;

//Update mystring and LCD in IRS

sprintf(mystring,"Count:%5d", Count);

GLCD\_PrintString(mystring, 60, 15, GLCD\_WHITE, GLCD\_BLACK);

//Ten iterations a second

CyDelay(100);

/\* `#END` \*/

}

C\_Interrupt.c

//This interrupt shall decrement the counter by one, but does not include code that updates the display.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Place your includes, defines and code here

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* `#START C\_Interrupt\_intc` \*/

//Access global variable

extern int Count;

#include <stdio.h>

#include <project.h>

/\* `#END` \*/

extern cyisraddress CyRamVectors[CYINT\_IRQ\_BASE + CY\_NUM\_INTERRUPTS];

/\* Declared in startup, used to set unused interrupts to. \*/

CY\_ISR\_PROTO(IntDefaultHandler);

CY\_ISR(C\_Interrupt\_Interrupt)

{

#ifdef C\_Interrupt\_INTERRUPT\_INTERRUPT\_CALLBACK

C\_Interrupt\_Interrupt\_InterruptCallback();

#endif /\* C\_Interrupt\_INTERRUPT\_INTERRUPT\_CALLBACK \*/

/\* Place your Interrupt code here. \*/

/\* `#START C\_Interrupt\_Interrupt` \*/

//Decrement counter

Count--;

//Debounce

CyDelay(200);

C\_Interrupt\_ClearPending();

/\* `#END` \*/

}

D\_Interrupt.c

//This interrupt shall decrement the counter 10 times per second, but does not include code that updates the display.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Place your includes, defines and code here

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* `#START D\_Interrupt\_intc` \*/

//Access global variable

extern int Count;

#include <stdio.h>

#include <project.h>

/\* `#END` \*/

extern cyisraddress CyRamVectors[CYINT\_IRQ\_BASE + CY\_NUM\_INTERRUPTS];

/\* Declared in startup, used to set unused interrupts to. \*/

CY\_ISR\_PROTO(IntDefaultHandler);

CY\_ISR(D\_Interrupt\_Interrupt)

{

#ifdef D\_Interrupt\_INTERRUPT\_INTERRUPT\_CALLBACK

D\_Interrupt\_Interrupt\_InterruptCallback();

#endif /\* D\_Interrupt\_INTERRUPT\_INTERRUPT\_CALLBACK \*/

/\* Place your Interrupt code here. \*/

/\* `#START D\_Interrupt\_Interrupt` \*/

//Decrement Counter

Count--;

//Ten iterations a second

CyDelay(100);

/\* `#END` \*/

}

Joystick\_Interrupt.c

//This interrupt resets the counter to zero

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Place your includes, defines and code here

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* `#START Joystick\_Interrupt\_intc` \*/

//Access global variable

extern int Count;

#include <stdio.h>

#include <project.h>

/\* `#END` \*/

extern cyisraddress CyRamVectors[CYINT\_IRQ\_BASE + CY\_NUM\_INTERRUPTS];

/\* Declared in startup, used to set unused interrupts to. \*/

CY\_ISR\_PROTO(IntDefaultHandler);

CY\_ISR(Joystick\_Interrupt\_Interrupt)

{

#ifdef Joystick\_Interrupt\_INTERRUPT\_INTERRUPT\_CALLBACK

Joystick\_Interrupt\_Interrupt\_InterruptCallback();

#endif /\* Joystick\_Interrupt\_INTERRUPT\_INTERRUPT\_CALLBACK \*/

/\* Place your Interrupt code here. \*/

/\* `#START Joystick\_Interrupt\_Interrupt` \*/

//Reset counter to 0

Count = 0;

/\* `#END` \*/

}