Quang Nguyen

CPE 409 Lab

# Goals

* To learn to how interface with LCD display using C code

# Equipment used

## Hardware

* Microchip Explorer 16 board
* PIC kit 3

## Software

* MPLAB X IDE 2.00

# Design Specifications

### Design 1

* Must use the onboard LCD to display a message
  + Must display “Embedded Systems” on the first line and “Are Great” on the second line

### Design 2

* The program must starts with the text of design 1
* On first press of S3
  + The display must be cleared
* Presses of S3 after the first press must result in the program printing a new character on the LCD display.
  + The program must reprint the text of design 1
* Presses of s6 will causes the display to lose one character.
  + The program will continues to erase a character on the press of S6 until the display is completely cleared.

# Design

* Refer to Figure 1 and 2 for the flow diagram of the program

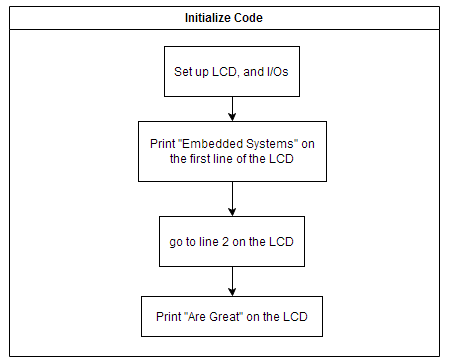


Figure 1: Flow diagram of the Initialize Code

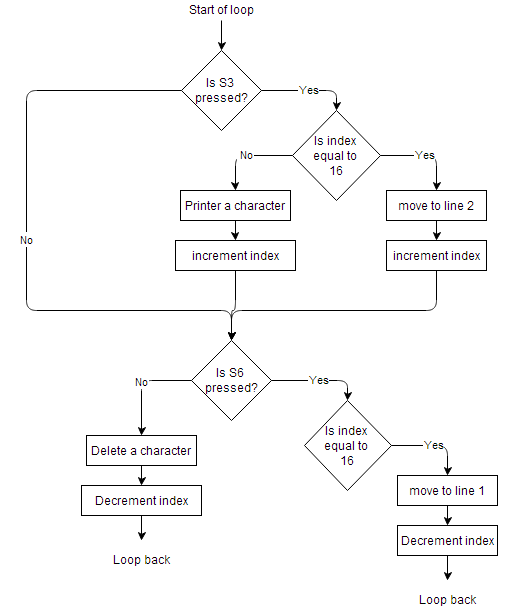


Figure 2: Flow diagram of the main loop

# Verification

* Upon giving power to the board, the LCD display was verified to display “Embedded Systems Are Great”
* S3 was then pressed
  + It was verified that the LCD display does indeed clear it self
* S3 was then pressed until all the characters are printed on the display
* S6 was then pressed until all the characters are cleared away

# Conclusions and Limitations

* No limitations were found
* The program works exactly as expected.

# Programming Code

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

\*

\* Library includes

\*

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

#include <p33FJ256GP710A.h>

#include "lcd.h"

#include <delay.h>

#include <string.h>

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

\*

\* Constant Declarations

\*

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

const char LcdData1[] = "Embedded Systems"; //sizeof() will result in 17

const char LcdData2[] = "Are Great"; //sizeof() will result in 10

const char LcdSpace[] = " ";

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

\*

\* Global Variable Declarations

\*

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

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

\*

\* Function Prototype

\*

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

void initialize();

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

\*

\* Main Function

\*

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

int main() {

// setting up everything

initialize();

int LcdCounter = 0;

int LcdFlag = 0, LcdLine = 1;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Main Loop \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

while (1) {

// Next press of RD6 = the printing of one char

if ((\_RD6 == 0) && (LcdFlag == 1))

{

// If the character can still be printed on the 1st line

if (LcdCounter < (sizeof(LcdData1)-1))

{

// Print the current character and increment

puts\_lcd(&LcdData1[LcdCounter], 1);

LcdCounter++;

}

// if the character is on the last (16th) position of the 1st line

if (LcdCounter == (sizeof(LcdData1)-1))

{

// increment

// puts\_lcd(&LcdData1[LcdCounter], 1);

LcdCounter++;

// Move to line 2 and indicate that we are now in line 2

line\_2();

LcdLine = 2;

}

// If the Lcd is pointing to LcdData2

else if (LcdCounter >= (sizeof(LcdData1)- 1))

{

// if ((LcdCounter == (sizeof(LcdData1) - 1 )) && (LcdLine == 1))

// {

// line\_2();

// LcdLine = 2;

// }

// If The pointer is still pointing at legitimate data

if (LcdCounter <= ((sizeof(LcdData1) + sizeof(LcdData2)) -2))

{

// Print the current character and increment

puts\_lcd(&LcdData2[(LcdCounter - sizeof(LcdData1))], 1);

LcdCounter++;

}

// Else don't do anything. Just sit and wait

}

Delay\_Us(1000); // Debouncing

}

// First press of RD6 = clear the screen

if ((\_RD6 == 0) && (LcdFlag == 0))

{

home\_clr();

home\_it();

LcdFlag = 1;

Delay\_Us(1000); // Debouncing

}

// If RD7 was pressed

if ((\_RD7 == 0))

{

if ((LcdCounter == sizeof(LcdData1)) && (LcdLine == 2))

{

lcd\_cmd(0x8F);

puts\_lcd(&LcdSpace, 1);

lcd\_cmd(0x8F);

LcdCounter--;

LcdCounter--;

// LcdLine = 1;

}

else

{

cursor\_left();

puts\_lcd(&LcdSpace, 1);

cursor\_left();

LcdCounter--;

}

Delay\_Us(1000); // Debouncing

}

} // End of the infinite While loop

return 1;

}

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

\*

\* Initialize Function

\*

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

void initialize() {

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

\* Setting up for Clock (PLL, M, N1, N2)

\* for 32 MHz and Fcy = 16 MHz

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

// Fosc = Fin(M/(N1\*N2)) = 8 MHz (32/(2\*4)) = 32 MHz

PLLFBD = 30; // M = 32

// N1 default is 2

// N2 default is 4

// Fcy = Fosc/2 by default

// Setting up RD6 and RD7

\_TRISD6 = 1;

\_TRISD7 = 1; // Set RD6 & 7 to inputs

// Initialize LCD

Init\_LCD();

lcd\_cmd(0xD);

// Making sure that we are starting at line 1, column 0

home\_it();

// Print the LcdData1

puts\_lcd(LcdData1, sizeof(LcdData1)-1);

// Move to next line

line\_2();

// Print the LCDData2

puts\_lcd(LcdData2, sizeof(LcdData2)-1);

}