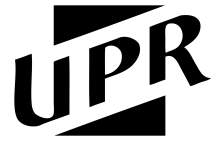




University of Puerto Rico  
Mayagüez Campus  
Mayagüez, Puerto Rico  
Department of Electrical and Computer Engineering



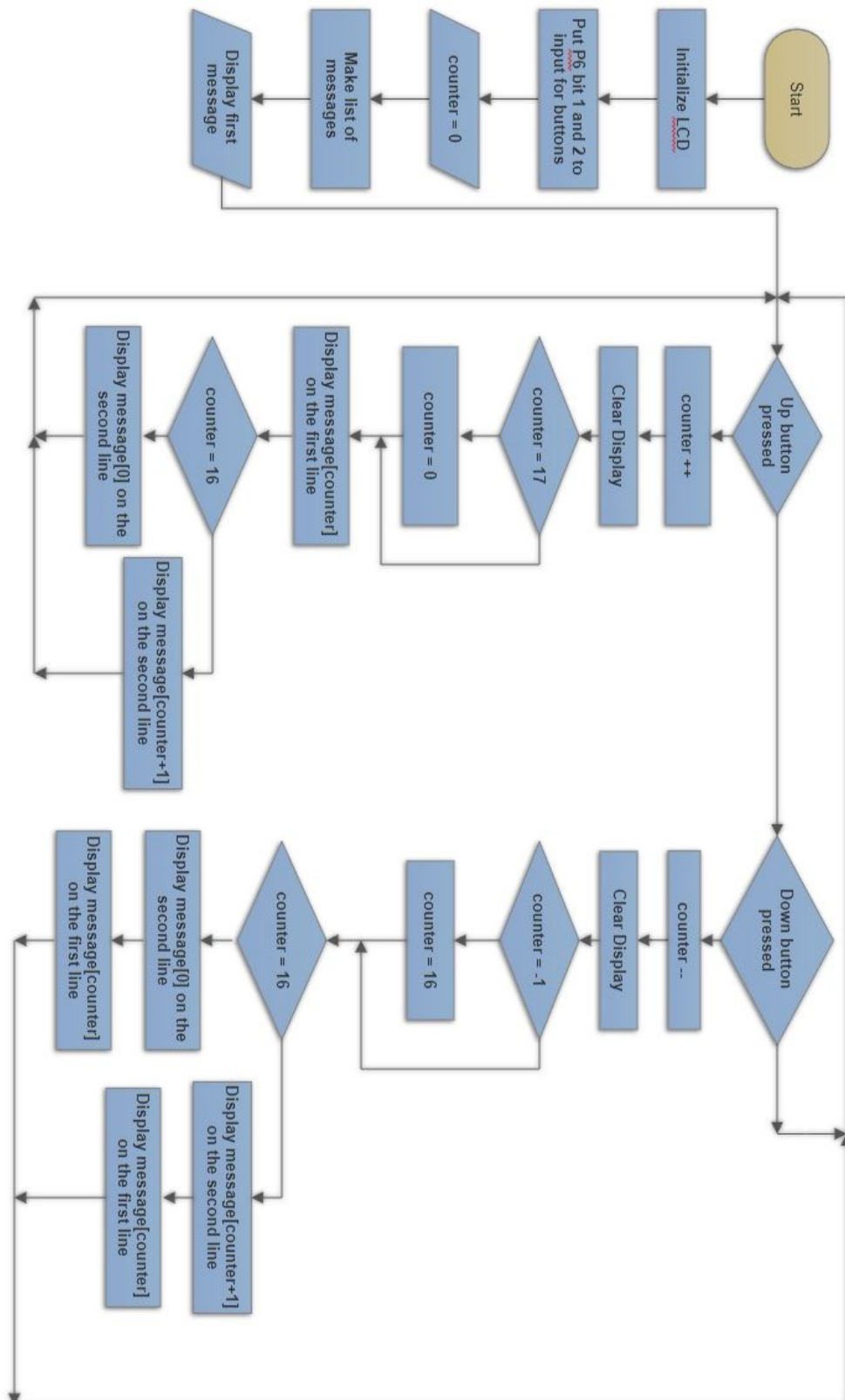
## Experiment 2: IDE, GPIOs and LCD

*by The Chicken Coop Team*

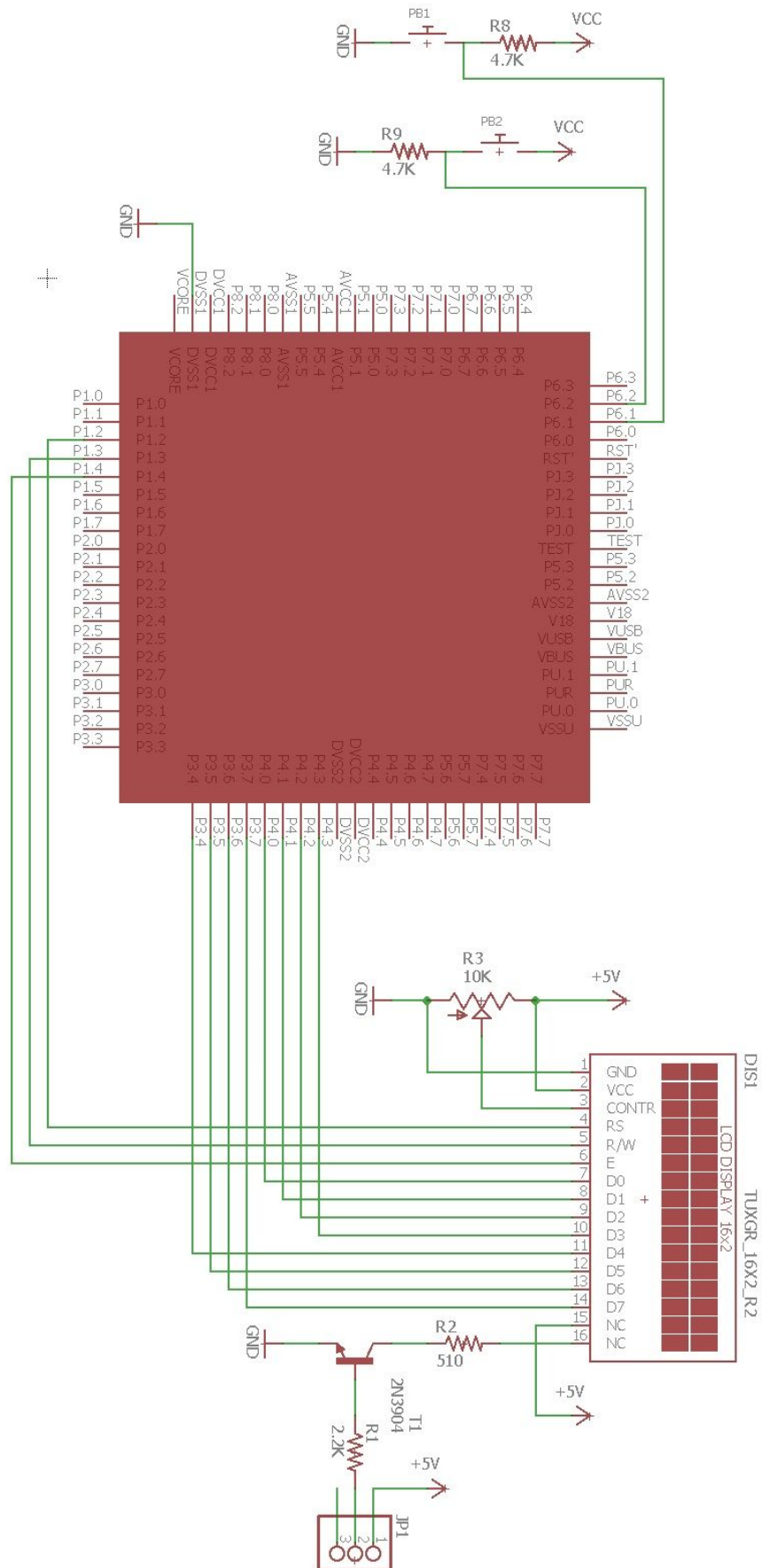
*Orlando Y. Alverio, Giovanni Gardón & Lianne Sánchez*

For: Dr. Manuel Jimenez  
Course: ICOM 4217, Section 090  
Date: September 11, 2017.

- Software plan and explanation (pseudocode and flowchart)



- Schematic



## • Code Listing

```

#include <lcd.h> // Including our own lcd header file programmed in Lab1 part 3
#include <msp430.h> // Including MSP430 family library

/**
 * main.c
 */
void main(void)
{
    WDCTL = WDTPW | WDTHOLD; // Stop the watchdog timer
    initialize(); // Calling the function "initialize" from our lcd.h header class
    // This initialize function establishes output registers for the
    LCD
    P6DIR= 0xF9; // The connections for the LCD are in P1, P4 and P3.
    // Establishing the input pins for the push buttons in port P6
    // which are P6.1 & P6.2

    unsigned char *ListOfMessages[] = {"1.Hola","2.somos","3.Giovanni","4.Lianne", //Creating an unsigned
character
    "5.Yamuel","6.estudiantes","7.de","8.Micro 2.", //pointer array with 17
messages
    "9.Nos","10.gusta","11.el","12.software","13.y",
    "14.el","15.hardware","16.Adios", "17.lol"};

    int counter=0; // Initializing a counter to zero, this counter is the one
that moves

    writeMessage(ListOfMessages[counter]); // through the array
    // Calling the function "writeMessage" from our lcd.h header
    // class, giving as parameter
    setCursorDown(); // The first char pointer of the pointer char array
    // Calling the function "seCursorDown" from out lcd.h header
    // class that sets cursor in
    // position 0x40 from the LCD screen.
    writeMessage(ListOfMessages[counter+1]); // Calling the function "writeMessage" from our lcd.h header
    // class, giving as parameter
    // The second char pointer of the pointer char array.

    while(1){ // While 1, being while true is a loop that does not have end.
        if((P6IN & 0x02)==0x00){ // When PB1 is pressed go into the block of if statement, PB1
is
        // located in P6.1 and is connected to a pull up resistor so its
        // reading a logic '1' all the time and a logic '0' when pressed,
        // when PB1 is pressed we will get xxxx00x, which means
        // that if we perform and with 0x02 the result when PB1 is
being
        // pressed has to be 0x00.
        counter++; // We increment the counter by the value of one.
        if(counter==17){ // If counter equals 17, a non-valid position in array
            counter=0; // Then, counter is set to 0 (starts from the beginning of list).
        }
        clear(); // Calling "clear" function from lcd.h that clears the display.
        setCursorUp(); // Calling "setCursorUp" function from lcd.h that sets cursor in
        // position 0x00.
        writeMessage(ListOfMessages[counter]); // Calling function "writeMessage" to write the
message on first

```

```

// line.
setCursorDown(); // Calling "setCursorDown" function from lcd.h that sets cursor
in
// position 0x40.
if (counter==16){ // SPECIAL CASE, when counter is 16 the next position is
// invalid
writeMessage(ListOfMessages[0]); // then, "writeMessage" in position 0 of array.
}
else{ // Else, all counters are valid
writeMessage(ListOfMessages[counter+1]); // "writeMessage" in the position after counter.
}
}
else if((P6IN & 0x06)==0x06){ // When PB2 is pressed go into the block of if statement, PB2
is
// located in P6.2 and is connected to a pull down resistor so
its
// reading a logic '0' all the time and a logic '1' when pressed,
// when PB2 is pressed we will get xxxxx11x, which means
// that if we perform an and with 0x06 the result when PB2 is
// being pressed has to be 0x06.
counter--; // We decrement the counter by the value of one.
if(counter==(-1)){ // If counter equals -1 i went out of range because the lowest
// index in array is 0.
counter=16; // then, set counter to 16 which is the highest index in the
array.
}
clear(); // Calling "clear" method from lcd.h header class to
clear display.
setCursorDown(); // Calling "setCursorDown" from lcd.h to set cursor in position
// 0x40.
if(counter==16){ // If counter equals 16 the next value is invalid
// then, we write the next available value which is position 0 of
// array.
setCursorUp(); // set cursor in position 0x40.
writeMessage(ListOfMessages[counter]); // and write on the top row the current element.
}
else{ // Else, is a valid index/counter
writeMessage(ListOfMessages[counter+1]); // Writing in lower row the next element in list
setCursorUp(); // set cursor in position 0x00
writeMessage(ListOfMessages[counter]); // writing the message of the current element in list
}
}
// DO NOTHING IF BUTTONS NOT PRESSED
}
}

```

- References

-www.tutorialspoint.com, "C Data Types". [Online]. Available: [https://www.tutorialspoint.com/cprogramming/c\\_data\\_types.htm](https://www.tutorialspoint.com/cprogramming/c_data_types.htm). [Accessed: September 11, 2017].

-“HD44780 pdf, HD44780 description, HD44780 datasheets, HD44780 view :: ALLDATASHEET ::”. [Online]. Available: <http://pdf1.alldatasheet.com/datasheet-pdf/view/63673/HITACHI/HD44780.html>. [Accessed: September 11, 2017].

-www.ti.com, "Digital I/O Configuration". [Online]. Available: [http://www.ti.com/sc/docs/products/micro/msp430/userguid/ag\\_08.pdf](http://www.ti.com/sc/docs/products/micro/msp430/userguid/ag_08.pdf). [Accessed: September 11, 2017].

-“MSP430F5529 LaunchPad - Texas Instruments Wiki”. [Online]. Available: [http://processors.wiki.ti.com/index.php/MSP430F5529\\_LaunchPad](http://processors.wiki.ti.com/index.php/MSP430F5529_LaunchPad). [Accessed: September 11, 2017].

-“Programming in C with CCS”. [Online]. Available: [http://software-dl.ti.com/trainingTTO/trainingTTO\\_public\\_sw/MSP430\\_LaunchPad\\_Workshop/v4/Chapters/MSP430m02\\_CCS.pdf](http://software-dl.ti.com/trainingTTO/trainingTTO_public_sw/MSP430_LaunchPad_Workshop/v4/Chapters/MSP430m02_CCS.pdf). [Accessed: September 11, 2017].

-www.ti.com, "MSP430F5229". [Online]. Available: <http://www.ti.com/product/msp430f5529?qgpn=msp430f5529>. [Accessed: September 11, 2017].

-www.ti.com, "MSP430x5xx and MSP430x6xx Family User's Guide". [Online]. Available: <http://www.ti.com/lit/ug/slau208p/slau208p.pdf>. [Accessed: September 11, 2017].