**LCD MODULE**

*Provides functions to display messages on the LCD screen.*

**LCD\_Initialize:**

*Runs through initialization sequence as listed in datasheet*

select the control register by lowering the appropriate pin

turn on display

set to 4 line mode, 5x10 pixel chars

set to 8 bit mode (data-bus size)

set cursor invisible, cursor shift left

clear screen

**Put\_Char:**

*Writes and ASCII character to the LCD screen*

Selects the data register, by raising the appropriate pin

write char variable onto data-bus pins

Pulse enable line

**LCD\_Display\_Message:**

*Outputs a specific message (There are several versions of this type of function in the code, eg. LCD\_Update, LCD\_Waterlogged, LCD\_Unload, LCD\_Wait\_for\_Team, etc)*

Setup char arrays of 20 chars long for each of the four lines

For 1-20 put the ith character in the string array onto the LCD screen

If the message to display has variables to deal with (e.g crab pot ID) use the INT\_to\_LCD function to put that text onto the LCD screen in the proper place

**INT\_to\_LCD:**

*outputs a decimal representation of a char variable and puts the corresponding ACSII characters on the LCD screen*

take number and divide by 1000, store in thousands variable

Set number to remainder of number / 1000

number /100 store in hundreds variable

modulo number by 100

number /10 store in tens variable

set remainder in ones variable  
 switch on throusands

case #:

put the ascii char value of # onto the LCD screen

case 0:

put whitespace on LCD Screen

switch on hundreds

case #:

put the ascii char value of # onto the LCD screen

case 0:

if this number is less than 1000, put whitespace on LCD Screen

switch on tens

case #:

put the ascii char value of # onto the LCD screen

case 0:

if this number is less than 100, put whitespace on LCD Screen

switch on ones

case #:

put the ascii char value of # onto the LCD screen

**HEX\_to\_LCD:**

*outputs a hex representation of a char variable and puts the corresponding ACSII characters on the LCD screen*

isolate the number’s most significant nibble by anding with 0xf0

switch on that mask

case0xX0:

put ascii char of X on screen

isolate the number’s least significant nibble by anding with 0x0f

switch on that mask

case0x0X:

put ascii char of X on screen