

```

// Modified and added LCD display by Glen Popiel - KW5GP

/*
  Uses MORSE ENDECODER Library by raronzen
  Copyright (C) 2010, 2012 raron
  GNU GPLv3 license (http://www.gnu.org/licenses)
  Contact: raronzen@gmail.com (not checked too often..)
  Details: http://raronoff.wordpress.com/2010/12/16/morse-endecoder/
*/

#include <avr/pgmspace.h>
#include <MorseEnDecoder.h> // Morse EnDecoder Library
#include <Wire.h> // I2C Library
#include <LiquidCrystal_I2C.h> // Liquid Crystal I2C Library

#define morseInPin 2
#define Speed_Pin A0

const int lcd_end = 16; // set width of LCD
const int lcd_address = 0x27; // I2C LCD Address
const int lcd_lines = 2; // Number of lines on LCD
String text; // Variable to hold LCD scrolling text
char cw_rx; // Variable for incoming Morse character
int read_speed; // Variable for desired CW speed setting
int current_speed=-1; // variables to track speed pot

MorseDecoder morseInput(morseInPin, MORSE_KEYER, MORSE_ACTIVE_LOW); //
Define the Morse objects

LiquidCrystal_I2C lcd(lcd_address,lcd_end,lcd_lines); // set the LCD I2C
address to 0x27 for a 16 chars and 2 line display

void setup()
{
  lcd.init(); // initialize the LCD
  lcd.backlight(); // Turn on the LCD backlight
  lcd.home(); // Set the cursor to line 0, column 0

  lcd.print("KW5GP CW Decoder");
  delay(3000);
  lcd.clear();
} // End Setup Loop

void loop()
{
  read_speed = map(analogRead(Speed_Pin),10,1000,5,35); // Read the
potentiometer to determine code speed
  if (current_speed != read_speed) // If the Speed Pot has changed,
update the speed and LCD
  {
    current_speed = read_speed; // Set the current speed to the desired
speed
    morseInput.setspeed(read_speed); // Call the set speed function

```

```

        text = String(current_speed) + " wpm"; // set up the LCD display
with the current speed
        lcd.clear(); // Clear tHE LCD
        lcd.setCursor(5,1); // Set the cursor to 5,1
        lcd.print(text); // Display the CW text
        text = "";
    }

    morseInput.decode(); // Decode incoming CW

    if (morseInput.available()) // If there is a character available
    {
        cw_rx = morseInput.read(); // Read the CW character

        // Display the incoming character - Set the text to display on line 0
only.
        if (text.length() >15) // When length = 15, trim and add to new
character so display appears to scroll left
        {
            text = text.substring(1,16); // Drop the First Character
        }
        text = text + cw_rx; // Set up the text to display on the LCD
        lcd.setCursor(0,0); // Set the cursor to 0,0
        lcd.print(text); // Display the CW text
    }
} // End Main Loop

```