/\* Ultra Sonic LCD Tape Measure

\* Arduino NANO with 16x2 LCD and Ultrasonic Sensor HC-SR04

\* Author: Richard Attenborough

\* Modified by Pat McMahon 15/9/2018

\* A007

\*This complete code has been tested and works with An Arduino Nano

\*

\*

\* Reworked from several sources.

\* Ultrasonic Sensor HC-SR04 and Arduino Tutorial - Created by Dejan Nedelkovski @ www.HowToMechatronics.com

\* Arduino NANO setup for LCD , Vasco Ferraz - Contact: http://vascoferraz.com/contact/ http://vascoferraz.com/tutorials/hitachi-hd44780-lcd/

\*/

// Add in the LiquidCrystal Library to interface with the 16x2 LCD

#include <LiquidCrystal.h> // includes the LiquidCrystal Library

// Initialize the LiquidCrystal library with the following Arduino's interface pins: 2,3,4,5,11,12

// Number 11 means Arduino's digital I/O port number 11.

// Number 12 means Arduino's digital I/O port number 12.

// LCD PIN - connection

// 1 VSS GROUND

// 2 VDD +5

// 3 VO 10k POTentiometer Output middle pin - Use this to adjust the contrast on the LCD Screen

// 4 RS Arduino NANO Digital Pin 12

// 5 RW GROUND

// 6 E/Enable Arduino NANO Digital Pin 11

// 7 D0 Nothing

// 8 D1 Nothing

// 9 D2 Nothing

// 10 D3 Nothing

// 11 D4 Arduino NANO Digital Pin 5

// 12 D5 Arduino NANO Digital Pin 4

// 13 D6 Arduino NANO Digital Pin 3

// 14 D7 Arduino NANO Digital Pin 2

// 15 A 48 Ohm Resistor then +5 V (Was originally specced at 220 Ohm but that was too high. The backlight did not glow).

// The resistor is present to protect the LED Backlight. Differnt screen sources may require different resistor values.

// 16 K GROUND

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Creates an LCD object. Parameters: (rs, enable, d4, d5, d6, d7)

//-----------------------------------------------------------------------

// Ultrasonic sensor setup . Only 2 data pins TRIG and ECHO.

// Sensor Pin - Connection

// 1 VCC +5

// 2 TRIG Arduino NANO Digital Pin 9

// 3 ECHO 1K Resistor then Arduino Nano Digital Pin 8

// 4 GND Ground

const int trigPin = 9; // for Ultrasonic sensor TRIG to Arduino NANO Digital Pin 9

const int echoPin = 8; // for Ultrasonic sensor ECHO to Arduino NANO Digital Pin 8

long duration; // define DURATION variable as LONG Integer

int distanceCm; // Define distance Variables as integers

//-----------------------------------------------------------------------

void setup()

{

// LCD INITIALIZE

lcd.begin(16,2); // Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display

// Ultrasonic Pins on Arduino SETUP

pinMode(trigPin, OUTPUT); // TRIGger is OUTPUT TO the Ultrasonic Sensor

pinMode(echoPin, INPUT); // ECHO is the returned INPUT FROM the Ultrasonic Sensor in Milliseconds

}

void loop()

{

//

// GET information From the Ultrasonic Sensor

//

// CLEAR the sensor settings

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

// TRIGGER the pulse

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH); // Measure the return time on the sensor

// CALCULATE the distance based on the speed of sound. Distance has to be divided by TWO because the sound has to go TO the destinatino then BACK. Twice the distance.

distanceCm= duration\*0.034/2; // This is the same as alternate formula ==> (duration/2) / 29.1 . This formula returns a value in CM

// 340 M/sec is the speed of sound at 15deg C. (0.034 CM/Sec)

// 331.5 M/sec @ 0 deg C (0.0331.5 CM/Sec)

// 343 M/Sec @ 20 deg C (0.0343 CM/Sec)

// 346 M/Sec @ 25 deg C (0.0346 CM/Sec)

// A more advanced configuration would add in a DHT11 or similar sensor to measure the temperature and alter the equation accordingly.

// DISPLAY the information on the LCD Screen

//

lcd.clear(); // Clear the screen

lcd.setCursor(0,0); // Sets the location at which subsequent text written to the LCD will be displayed

lcd.print("Dist: "); // Prints string "Distance" on the LCD

lcd.print(distanceCm); // Prints the distance value from the sensor (This is appended to the previous output)

lcd.print(" cm"); // Append the term CM to the output to make clear the units that are being used. (This is appended to the previous output)

delay(100); // Need to have a delay for screen stability . Any lower delay than this and the screen may get scrambled

}