

```
// SpeedTrap by Sumner Patterson
// v. 24-07-16
```

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
/*****/
char array1[]=" SpeedTrap Speedometer  "; //the string to print on the LCD on startup
char array2[]=" By Sumner Patterson    "; //the string to print on the LCD on startup
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display //
was 0x27 tried 0x3F
```

```
#define rightSensorPin 27
#define leftSensorPin 26
#define reSetPin 32
```

```
unsigned long time1 = 0;
unsigned long rightSensorMillis=0;
unsigned long leftSensorMillis=0;
unsigned long trapTime=0;
unsigned long constantMph = 0;
unsigned long MPH = 0;
```

```
// variables will change:
```

```
int previousMph = 0;
int currentMph = 0;
int lastMph = 000;
int reSet = 0; // reads the rest button switch
int rightSensorState = 0; // right sensor beam.
int leftSensorState = 0; // left sensor beam.
int rightSenFlag = 1; // Have it only read one time and then change value of rightSenFlag until x is
reset to 1.
int leftSenFlag = 1; // Have it only read one time and then change value of leftSenFlag until x is
reset to 1.
int showDisplayFlag = 1; // I think I'm actually not using this to do anything at the moment.
int mphFlag = 0; // I think I'm actually not using this to do anything at the moment.
int tim = 300; //the value of delay time
long Time = 0;
long debounce = 700; // Stops reading the reset button more than once....increase if that is a problem.
int resetFlag = 1; // show waiting for loco
```

```
void setup()
{
  Wire.begin(18, 22); // 18 is SDA and 22 is SCL
```

```
  pinMode(reSetPin, INPUT_PULLUP); // initialize the reSetPin as an input
  pinMode(rightSensorPin, INPUT); // initialize the right sensor pin as an input:
  pinMode(leftSensorPin, INPUT); // initialize the left sensor pin as an input:
  pinMode(rightSensorPin, INPUT_PULLUP); // turn on the pullup for right sensor
```

```
pinMode(leftSensorPin, INPUT_PULLUP); // turn on the pullup for left sensor***** This did not work and was from Adafruit
```

```
constantMph = 54500; // Initially 54500 -- Increase to make mph lower -- Decrease to make mph higher.
```

```
rightSenFlag = 1;  
leftSenFlag = 1;  
mphFlag = 1;  
lastMph = 000;  
previousMph = 000;
```

```
lcd.init(); //initialize the lcd  
lcd.backlight(); //open the backlight
```

```
Serial.begin(9600);
```

```
lcd.setCursor(15,0); // set the cursor to column 15, line 0  
for (int positionCounter1 = 0; positionCounter1 < 26; positionCounter1++)  
{  
  lcd.scrollDisplayLeft(); //Scrolls the contents of the display one space to the left.  
  lcd.print(array1[positionCounter1]); // Print a message to the LCD.  
  delay(tim); //wait for 250 microseconds  
}  
lcd.clear(); //Clears the LCD screen and positions the cursor in the upper-left corner.  
lcd.setCursor(15,1); // set the cursor to column 15, line 1  
for (int positionCounter2 = 0; positionCounter2 < 26; positionCounter2++)  
{  
  lcd.scrollDisplayLeft(); //Scrolls the contents of the display one space to the left.  
  lcd.print(array2[positionCounter2]); // Print a message to the LCD.  
  delay(tim); //wait for 250 microseconds  
}  
lcd.clear(); //Clears the LCD screen and positions the cursor in the upper-left corner.  
delay (1000);  
lcd.setCursor(4,0); // set the cursor to column 4, line 0  
  lcd.print("SpeedTrap");  
lcd.setCursor(0,1); // set the cursor to column 0, line 1  
  lcd.print("Waiting on Loco");  
delay (1000);  
// lcd.clear();  
}  
/*****
```

```
void loop()
```

```
{  
  // ***** read the state of the right sensor *****  
  
  rightSensorState = digitalRead(rightSensorPin);  
  if (rightSenFlag == 1) { // Have it only read one time and then change value of rightSenFlag until x is reset to 1  
    if (rightSensorState == LOW) { // check if the sensor beam is broken. If it is, the sensorState is
```

LOW:

```
rightSenFlag = 2;
rightSensorMillis = millis();
Serial.println(" .....");
Serial.println((String) " Right Sensor Triggered....."+"rightSenFlag = "+rightSenFlag);
}
```

```
// ***** read the state of the left sensor
```

```
*****
```

```
leftSensorState = digitalRead(leftSensorPin);
if (leftSenFlag == 1) { // Have it only read one time and then change value of leftSenFlag until
x is reset to 1)
if (leftSensorState == LOW) { // check if the sensor beam is broken. If it is, the sensorState is
LOW:
```

```
leftSenFlag = 2;
leftSensorMillis = millis();
Serial.println(" .....");
Serial.println((String) " Left Sensor Triggered....."+"leftSenFlag = "+leftSenFlag);
}
```

```
// *****Determine the MPH
```

```
*****
```

```
if (rightSenFlag == 2 && leftSenFlag == 2) {
if (rightSensorMillis < leftSensorMillis){
trapTime = (leftSensorMillis - rightSensorMillis);
} else if (rightSensorMillis > leftSensorMillis){
trapTime = (rightSensorMillis - leftSensorMillis);
}
MPH = constantMph / trapTime; // calculates mph.
mphFlag = 2;
}
```

```
//***** Reads the reset button
```

```
*****
```

```
reSet = digitalRead(reSetPin); //Reads the button to refresh the display
if (( reSet == LOW) && ( millis() - Time) > debounce) { // Button == LOW && ( millis() -
Time) > debounce &&
```

```
Time = millis();
```

```
lcd.clear(); //Clears the LCD screen and positions the cursor in the upper-left corner.
```

```
rightSenFlag = 1; // reset right sensor so it can turn on.
leftSenFlag = 1; // reset left sensor so it can turn on.
showDisplayFlag = 1;
mphFlag = 1;
rightSensorMillis=0;
leftSensorMillis=0;
lastMph = MPH;
```

```

MPH = 0;
resetFlag = 1;

if (resetFlag = 1) { // show waiting for loco
  lcd.setCursor(0,1); // set the cursor to column 4, line 1
  lcd.print("      ");
  lcd.setCursor(0,1); // set the cursor to column 4, line 1
  lcd.print("Previous MPH ");
  int previousMphInt = lastMph;
  char previousMph[10];
  sprintf(previousMph, "%d", previousMphInt);
  lcd.print(previousMph);
  showDisplayFlag = 0;

lcd.setCursor(0,0); // set the cursor to column 4, line 1
  lcd.print("Waiting on Loco");
  resetFlag = 2;
  }
}
//***** Displays Current MPH *****

if (rightSenFlag == 2 && leftSenFlag == 2)
{
  lcd.setCursor(0,0); // set the cursor to column 4, line 1
  lcd.print("      ");
  lcd.setCursor(0,0); // set the cursor to column 4, line 1
  lcd.print("Current MPH ");
  int currentMphInt = MPH;
  char currentMph[10];
  sprintf(currentMph, "%d", currentMphInt);
  lcd.print(currentMph);

//***** Displays Previous MPH *****

  lcd.setCursor(0,1); // set the cursor to column 4, line 1
  lcd.print("      ");
  lcd.setCursor(0,1); // set the cursor to column 4, line 1
  lcd.print("Previous MPH ");
  int previousMphInt = lastMph;
  char previousMph[10];
  sprintf(previousMph, "%d", previousMphInt);
  lcd.print(previousMph);
  showDisplayFlag = 0;
}

// ***** End Show Display
*****
  delay (1);
}

```