```
/*
Timing Measurement
The circuit:
 * LDR_start  attached from pin 13 to ground
 * LDR_stop  attached from pin 12 to ground
 *   The circuit:
 * LCD RS pin to digital pin 1
 * LCD Enable pin to digital pin 2
 * LCD D4 pin to digital pin 3
 * LCD D5 pin to digital pin 4
 * LCD D6 pin to digital pin 5
 * LCD D7 pin to digital pin 6
 * LCD R/W pin to ground
 * 10K resistor:
 * ends to +5V and ground
 * wiper to LCD VO pin (pin 3)
 */

// include the library code:
#include <LiquidCrystal.h>

// constants won't change. They're used here to
// set pin numbers:
const int LDR_start = 13;     // the number of the pushbutton pin
const int LDR_stop =  12;      // the number of the LED pin

// Some variables to read the input digital information
int start_var = 0;
int stop_var = 0;

// variables will change:
int State = 0;         // variable for reading the pushbutton status
// 0 is waiting/armed
// 1 is time started
// 2 is time stop

// Start and stop times
unsigned long start_time_us;
unsigned long stop_time_us;
float delta_time;
float speed_var;
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(1, 2, 3, 4, 5, 6);

void setup() {
  // initialize the LDR_start:
  pinMode(LDR_start, INPUT);
  // initialize the LDR_stop:
  pinMode(LDR_stop, INPUT);
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  lcd.clear();
```

```
    lcd.setCursor(0, 0);

  lcd.print("SpeeDe is ready");// Print a message to the LCD.
}
void loop(){
  // Serial.print("State = ");
  // Serial.println(State);
  switch(State){
    case 0:
      // 0 is waiting/armed
       start_var = digitalRead(LDR_start);
       if(start_var == 1){
         start_time_us = micros();
         State = 1;
        }
       break;
    case 1:
       stop_var = digitalRead(LDR_stop);
       if(stop_var == 1){
         stop_time_us = micros();
         State = 2;
        }
       break;
   default:
     lcd.clear();
     lcd.setCursor(0, 0);
     delta_time = (stop_time_us - start_time_us)/1e6;
     lcd.print("Time Delta in s");
     lcd.setCursor(0,1);
     lcd.print(delta_time);
     delay(3000);

     speed_var = 5.5e-2/delta_time;
     lcd.clear();
     lcd.setCursor(0, 0);
     lcd.print("Speed in m/s");
     lcd.setCursor(0,1);
     lcd.print(speed_var);
     delay(3000);
     lcd.clear();

     speed_var = speed_var*3.6;
     lcd.clear();
     lcd.setCursor(0, 0);
     lcd.print("Speed in km/hr");
     lcd.setCursor(0,1);
     lcd.print(speed_var);
     delay(3000);

    }}
```