```
#include <avr/io.h> // import avr library for registers
#include <avr/interrupt.h> // import avr library for interrupt function
#include <LiquidCrystal.h> // import the library for LCD character display
// initialize the library and define pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int AnalogIn_0 = A0;
float sensorVal;
int led = 9;
void setup(){
 lcd.begin(16, 2); // set up the LCD
 pinMode(led, OUTPUT); // set up the led pin

 /* Configure interrupt on Timer1 */
 cli(); // disable global interrupts before configuring interrupt
 TCCR1A = 0; // set TCCR1A register to 00000000
 TCCR1B = 0; // set TCCR1B register to 00000000
 OCR1A = 15624; // set compare match register
 /* Notice that (15624+1)*1/16000000*1024 = 1 second */

 TCCR1B |= (1 << WGM12); // Set WGM12 bit to 1, turns on CTC mode:
 TCCR1B |= (1 << CS12)|(1 << CS10); // Set CS10 and CS12 bits to 1
 /* Notice that When CS12=1, CS11=0 and CS12=1, pre-scaler is 1024 */

 TIMSK1 |= (1 << OCIE1A); // enable timer compare interrupt:
 sei(); // enable global interrupts after configured interrupt
}
//Interrupt Service Routine
ISR(TIMER1_COMPA_vect){ // This function runs once every time timer compare matches
 digitalWrite(led, !digitalRead(led)); // toggle led pin
}
void loop(){
 // Convert 0 to 1023 ADC Reading Value to 0 to 5V voltage value
 sensorVal = analogRead(AnalogIn_0)*5/1024.000;

 lcd.clear(); // start with a blank screen (refresh)
 lcd.setCursor(0, 0); // set the cursor to column 0, line 0
 lcd.print("Saurav");
 lcd.setCursor(0, 1);
 lcd.print(sensorVal);

 delay(100);
}
```

Done uploading.

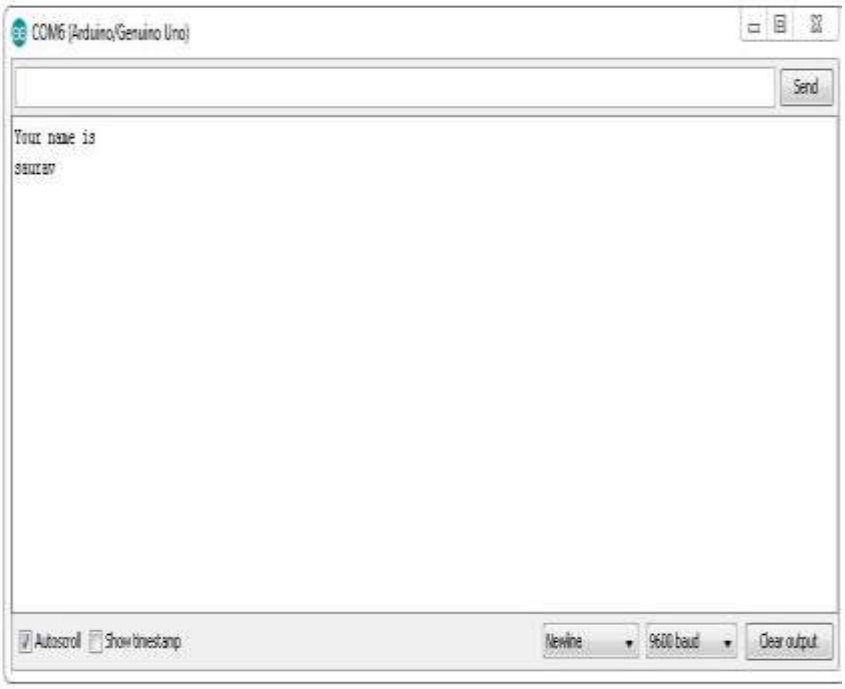
```
Sketch uses 3672 bytes (11%) of program storage space. Maximum is 32256 bytes.
Global variables use 71 bytes (3%) of dynamic memory, leaving 1977 bytes for local variables. Maximum is 2048 bytes.
```

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

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4,3,2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("saurav");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```

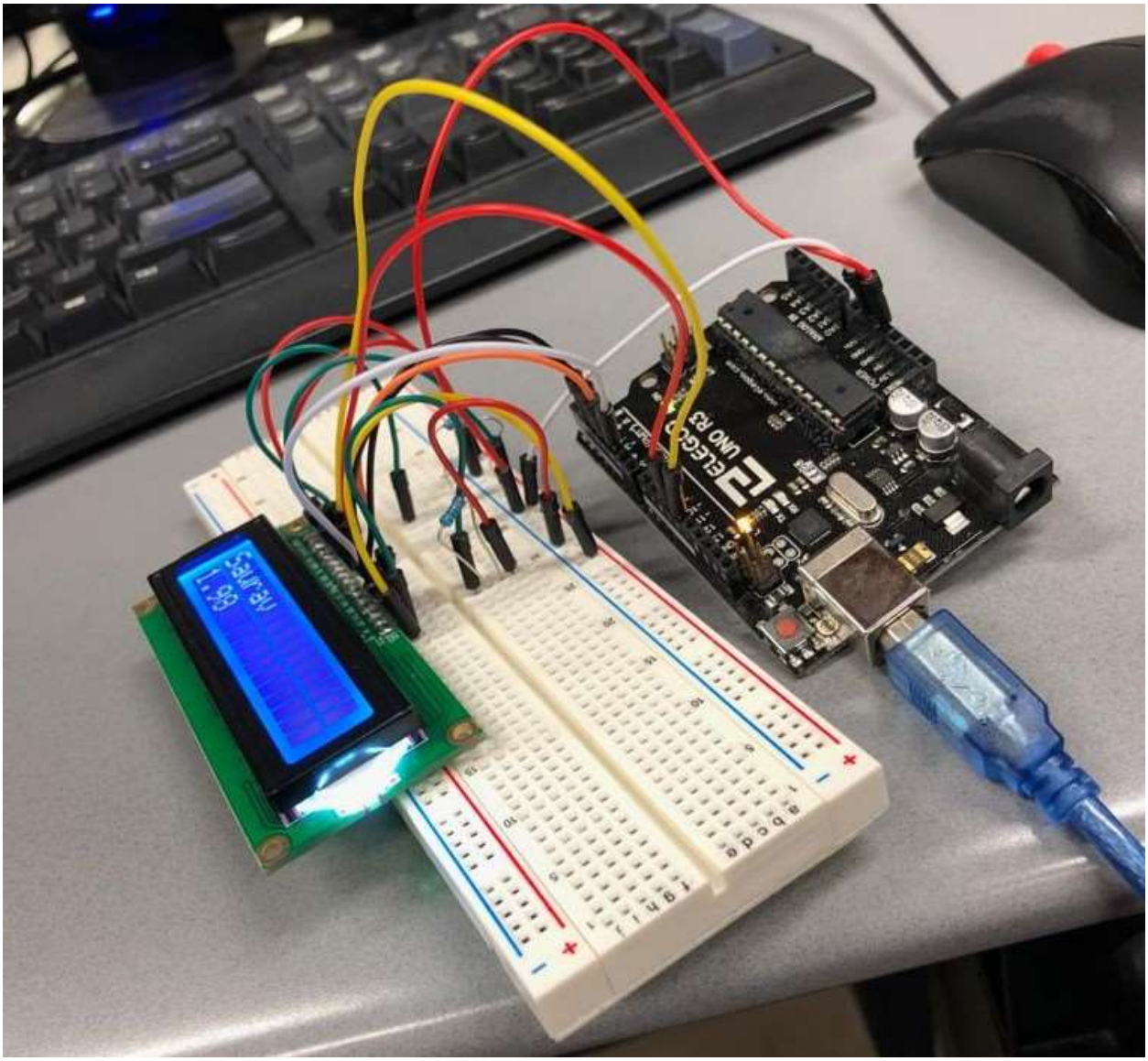
Done uploading.

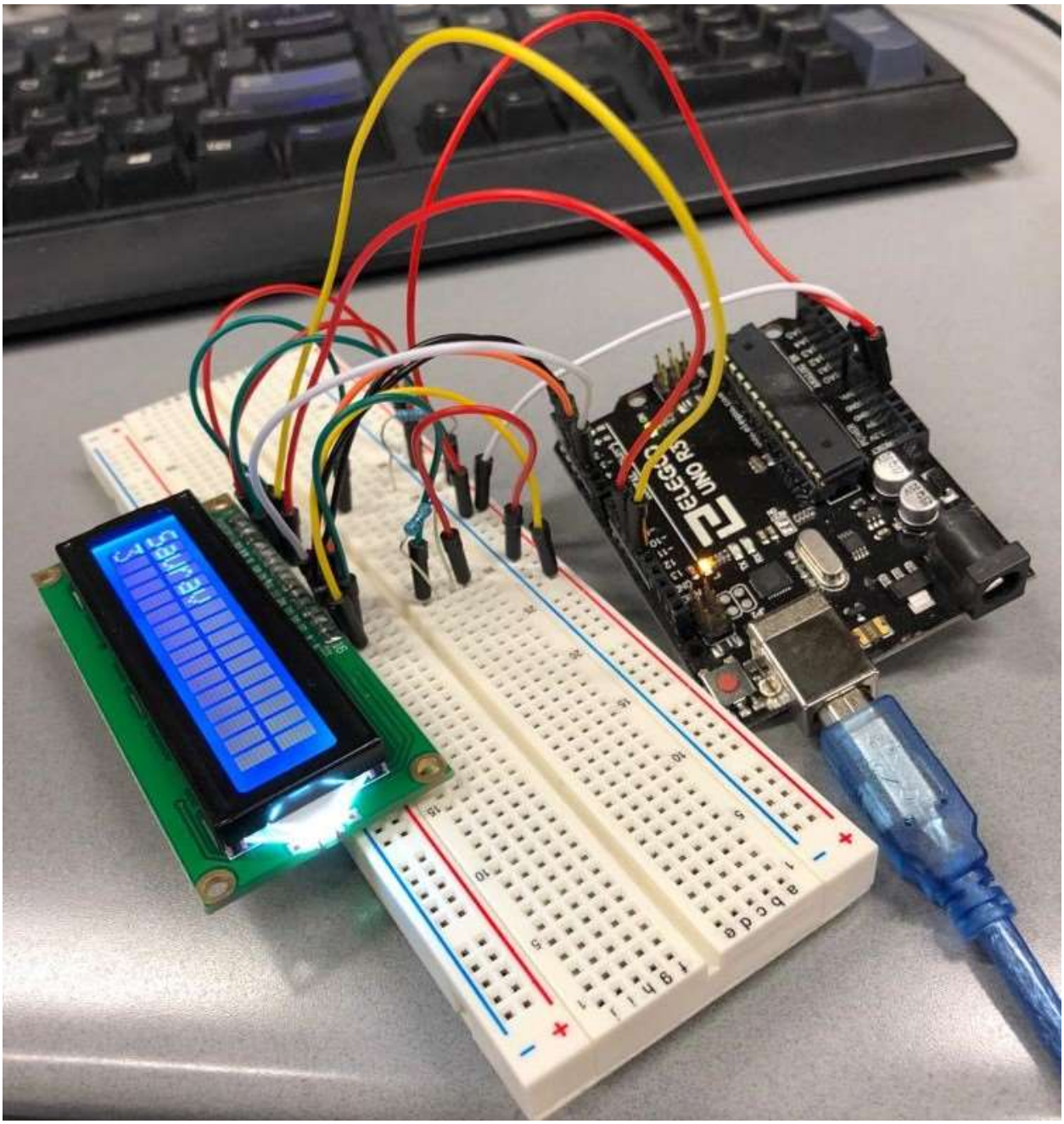
```
Sketch uses 1894 bytes (5%) of program storage space. Maximum is 32256 bytes.
Global variables use 51 bytes (2%) of dynamic memory, leaving 1997 bytes for local variables. Maximum is 2048 bytes.
```

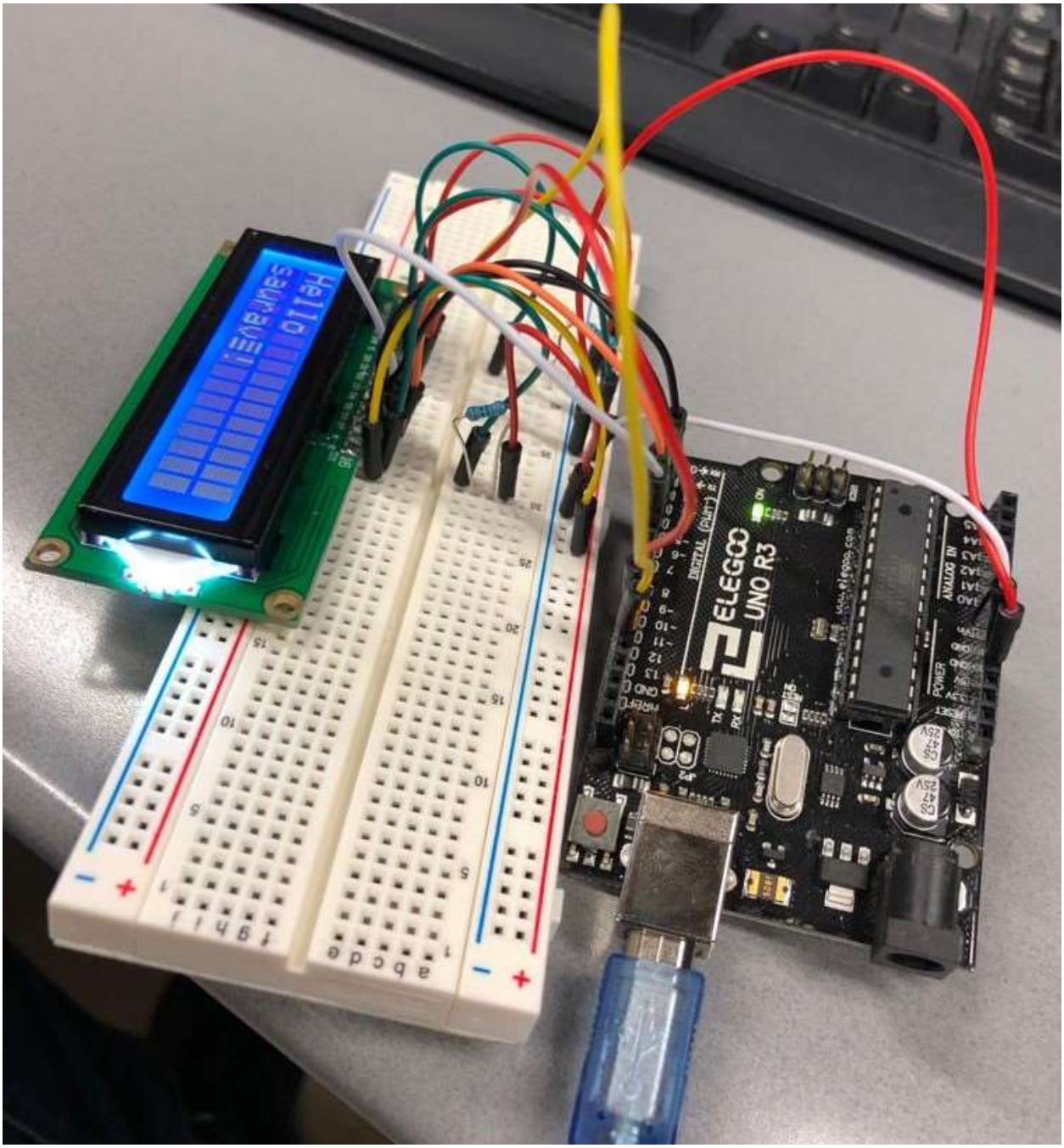
```
sketch_jul10
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
char name [15]; // Declare Array of Characters
int index = 0; // Declare Array Index
void setup(){
 lcd.begin(16, 2); // Set up the LCD
 Serial.begin(9600); // Set up Serial Port
 Serial.println("Your name is "); // Ask user for input
}

void loop(){
 // Wait for Serial Input
 while(Serial.available()==0){
 } /* Note that Serial.available() function
returns the number of chars in serial port buffer.
 as you call the Serial.read(), the value returned from
 Serial.available() function will decrease accordingly.*/

 // LCD Init
 lcd.clear(); // start with a blank screen (refresh)
 lcd.setCursor(0, 0); // set the cursor to column 0, line 0
 lcd.print("Hello");
 lcd.setCursor(0, 1);

 // Capture Serial Input and fill the name character array
 index = 0;
 while(Serial.available()>0){
 if(index < 15){
 name[index] = Serial.read();
 Serial.print(name[index]);
 index+=1;
 delay(100); //Delay for 0.1s for reliable serial capture
 }
 }

 // LCD Output
 for (int i=0; i<index; i++) {
 lcd.print(name[i]);
 lcd.print("!");

 delay(1000);
}
```




```
Sketch uses 2992 bytes (9%) of program storage space. Maximum is 32256 bytes.
Global variables use 263 bytes (12%) of dynamic memory, leaving 1785 bytes for local variables. Maximum is 2048 bytes.
```


UNO R3


UNO R3
ELEGOO


Hello
ELEGOO
UNO R3