Introduction to Arduino

Week 4

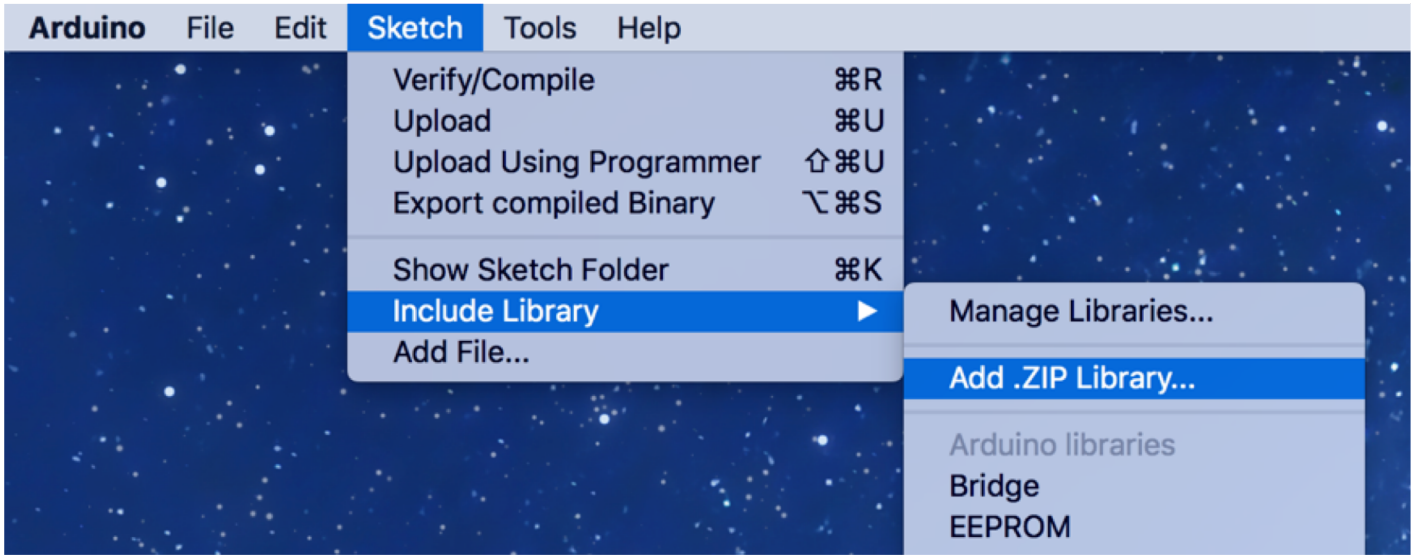
Questions about last 3 weeks

Experiments

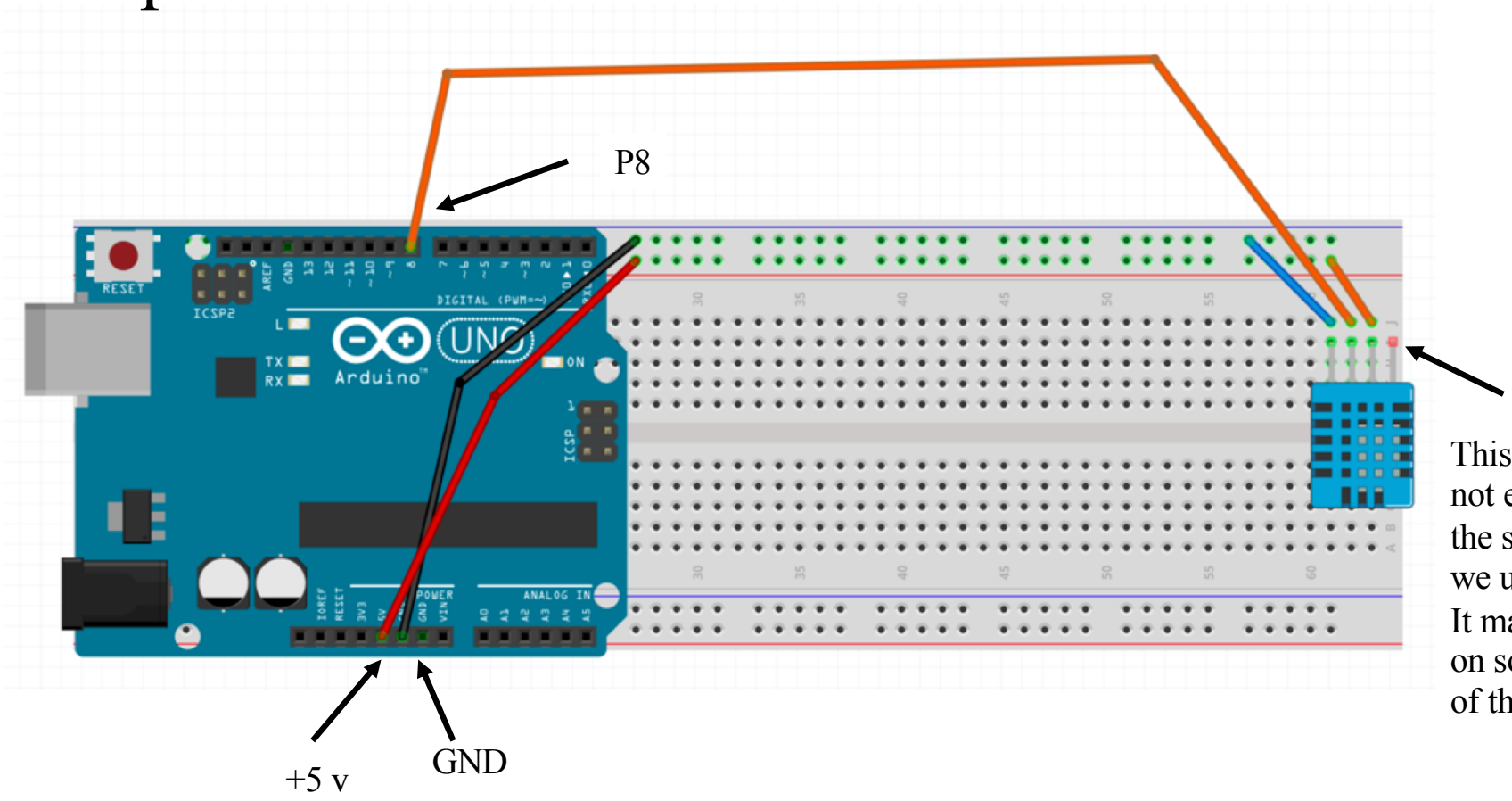
- Experimental test setup
- Temperature and Humidity Sensor using serial output
- Temperature and Humidity Sensor with LCD display
- Servo
- Ultra-sonic ranging using serial output
- Ultra-sonic ranging with LCD display
- Other (if time permits)
  - Buzzer
  - Relay

Temperature & Humidity Sensor experiment

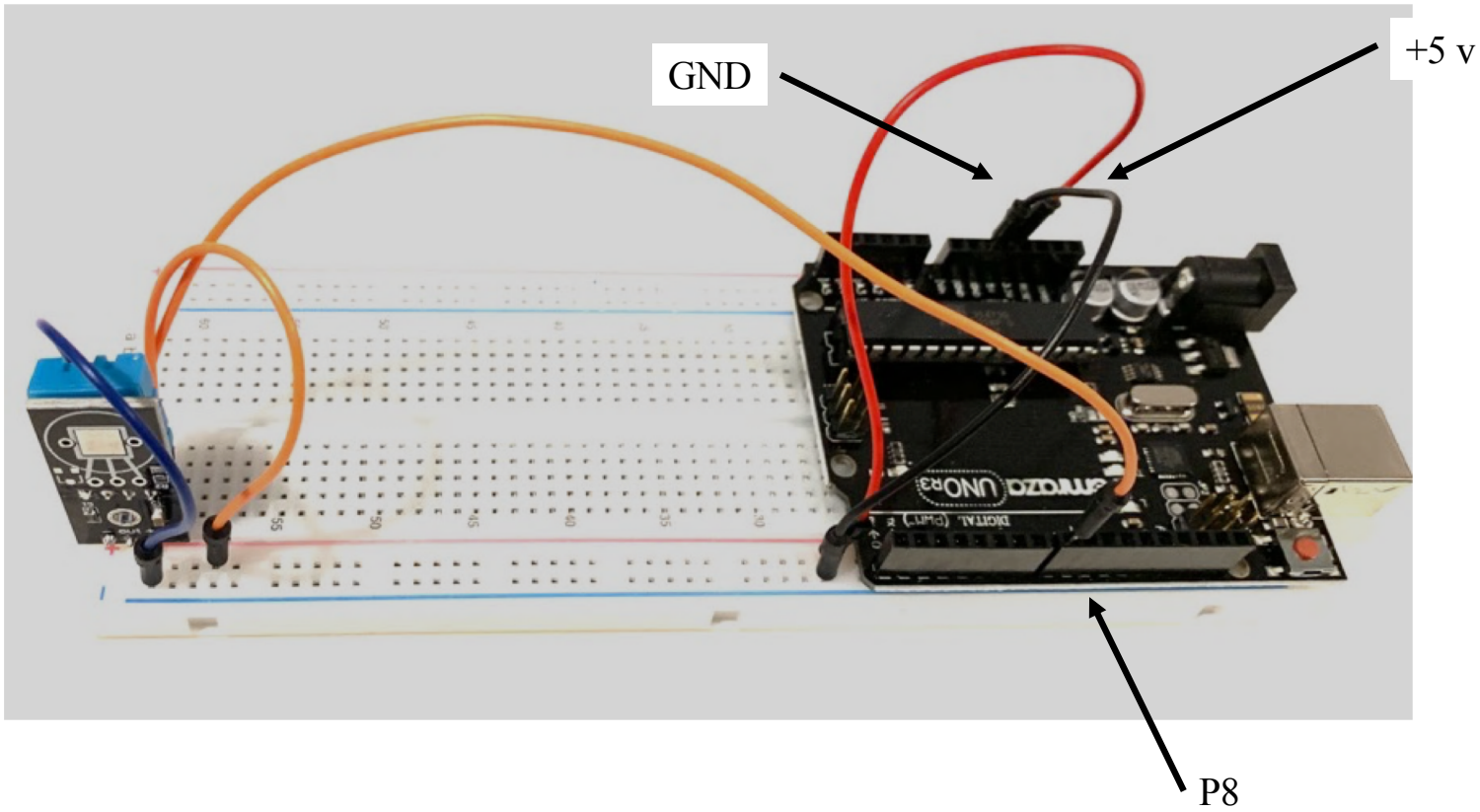
- The supplied kit comes with a basic temperature and humidity sensor
- We have to install the driver for the sensor
- We then use the driver by:



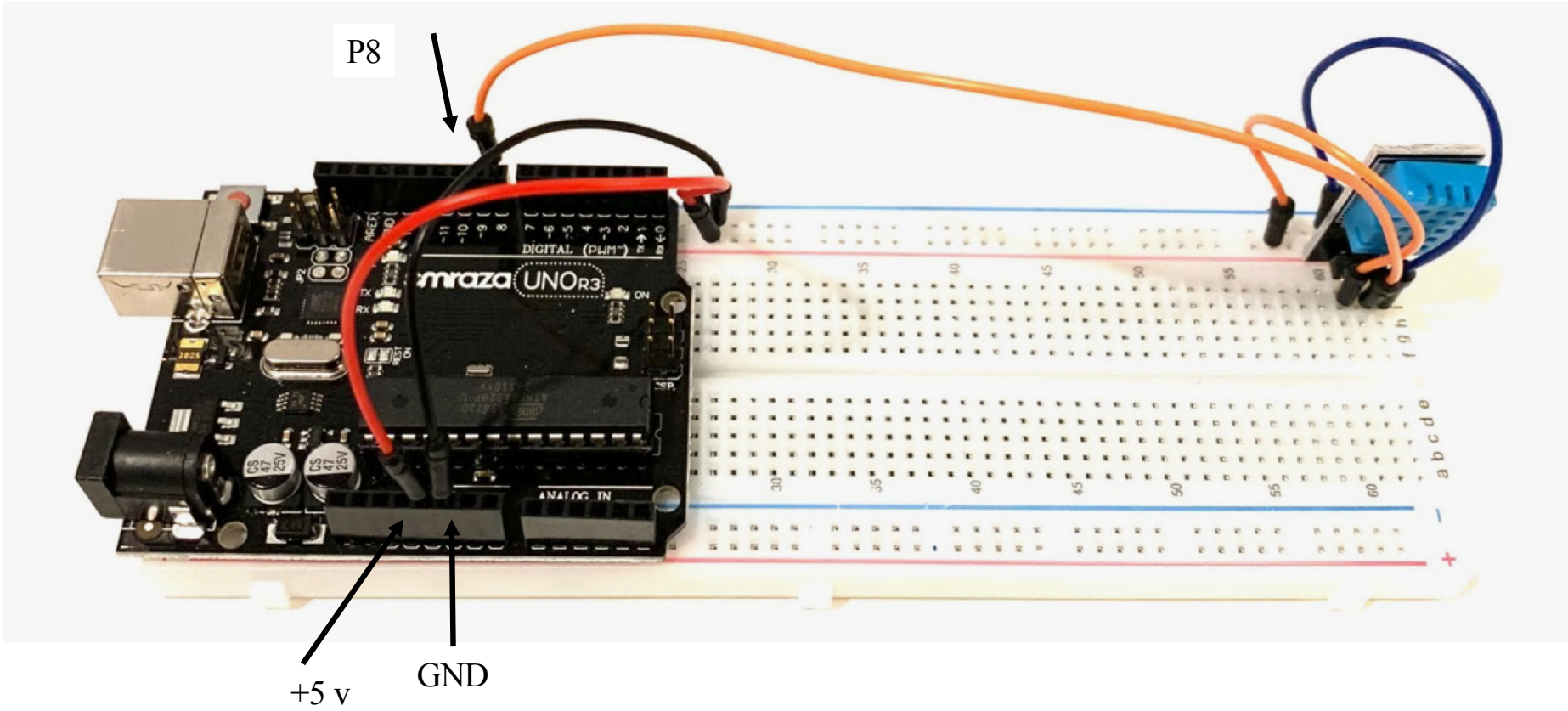
Temperature & Humidity Sensor experimental setup



Temperature & Humidity Sensor experimental setup – View 1

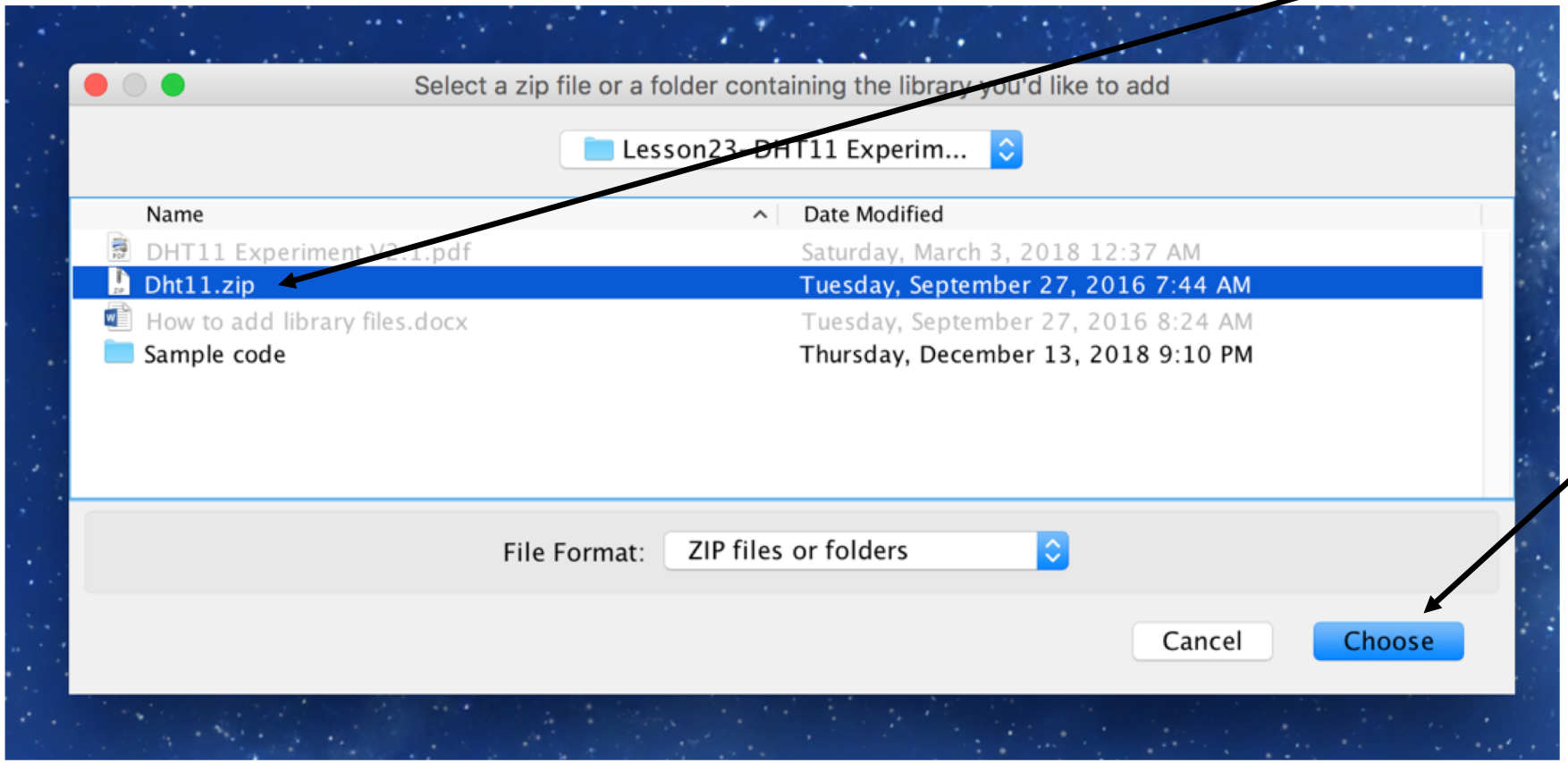


Temperature & Humidity Sensor experimental setup – View 2



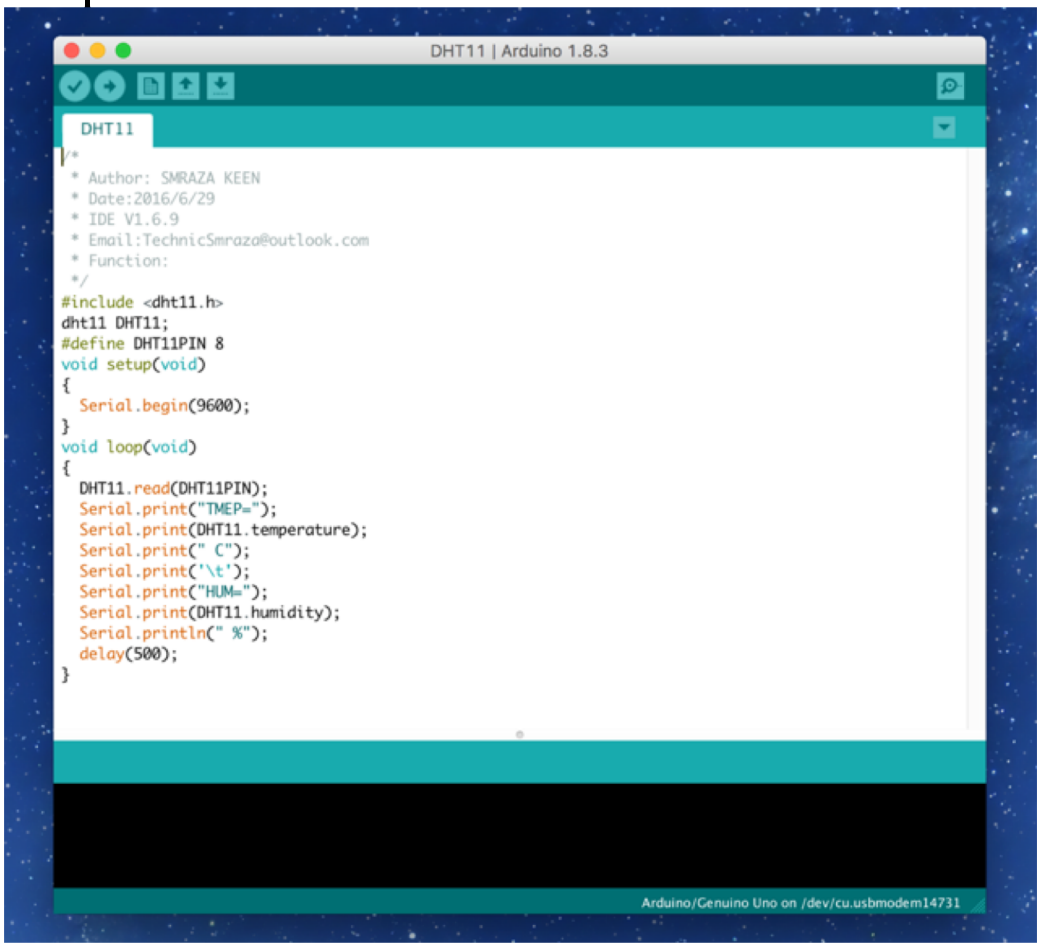
Adding DHT library

- Find the less23-DHT Experiment on your hard drive

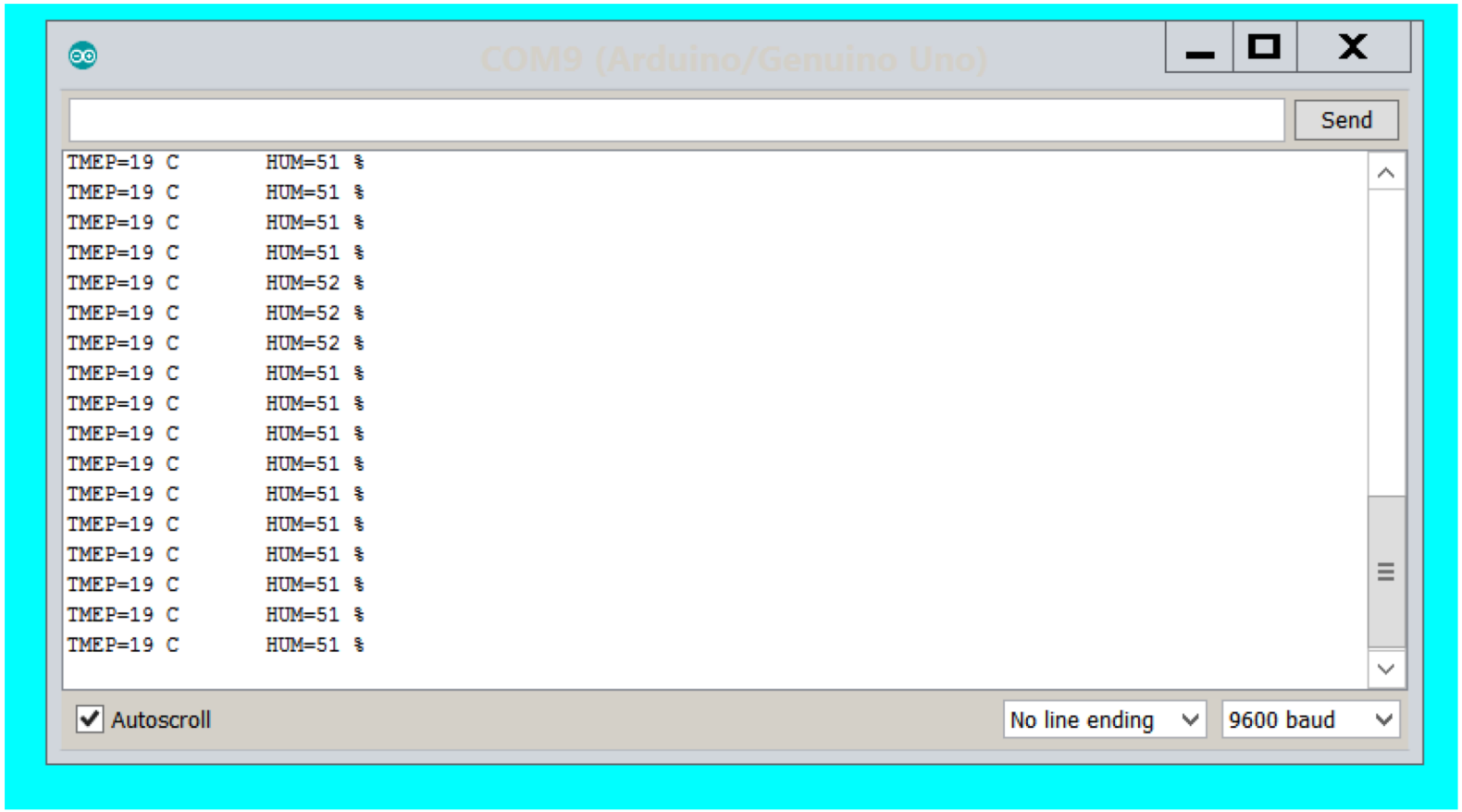


Temperature & Humidity Sensor experiment

- Open the sample program from the downloaded zip file
- Lesson 23



Sample Output of Temperature and Humidity Measure



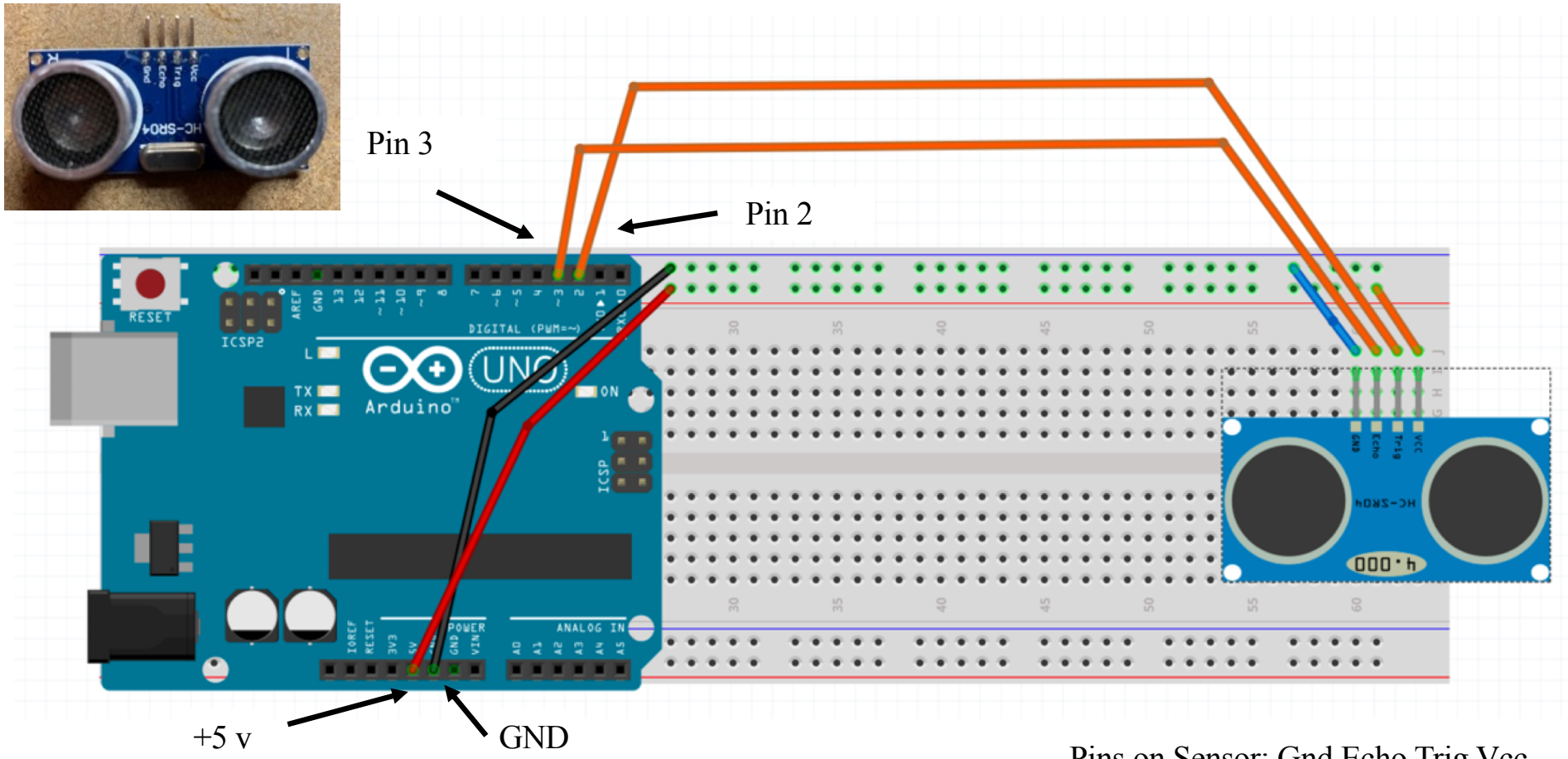
Ultra-sonic ranging Experiment

- The ultrasonic sensor has a
  - Transmitter (Trigger pin)
  - Receiver (Echo pin)
- Over a range of 0 – 1 meter (or so),
  - The sound waves will be transmitted by the sensor
  - Reflected off an object
  - Received at the sensor
- It functions like a radar using sound rather than radio waves
- It is useful to measure distance
- It can also be used on a robot to avoid collisions with objects

Warning

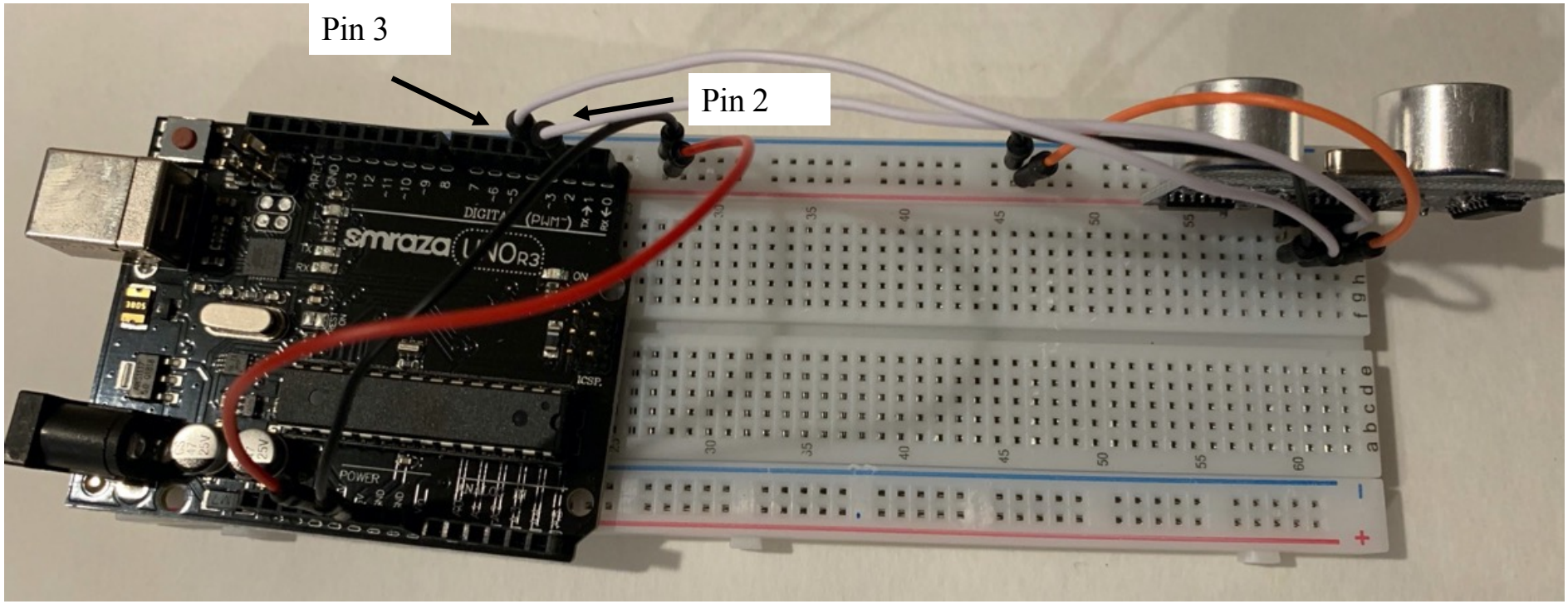
- The HC-SR04 ultrasonic sensor is a 5 volt device
- The Arduino Uno is a 5 volt device
- Other Arduino types use 3.3 volts
  - On them we must use a level converter on the Echo line
- Not required for the Uno

Distance Measure Setup

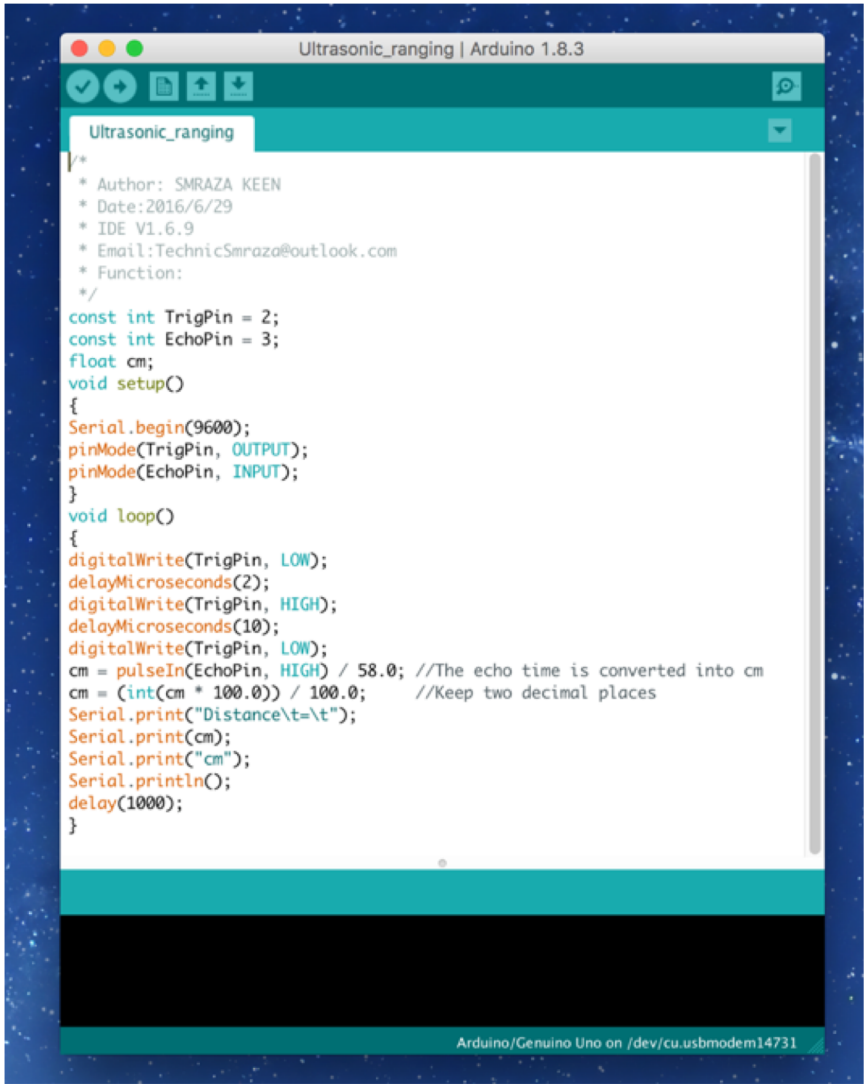


Pins on Sensor: Gnd Echo Trig Vcc

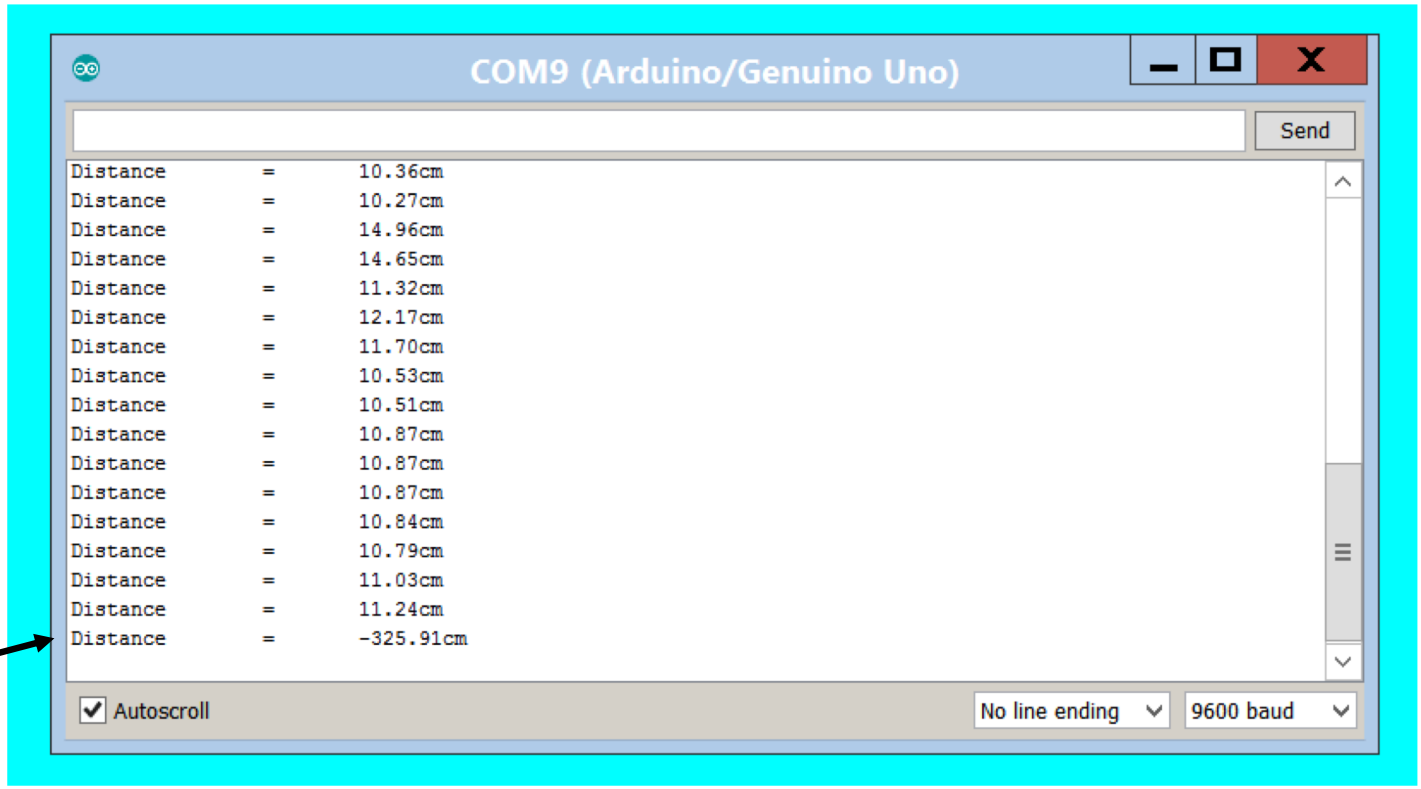
Distance Measure Setup



Ultrasonic Ranging Code



Example Output

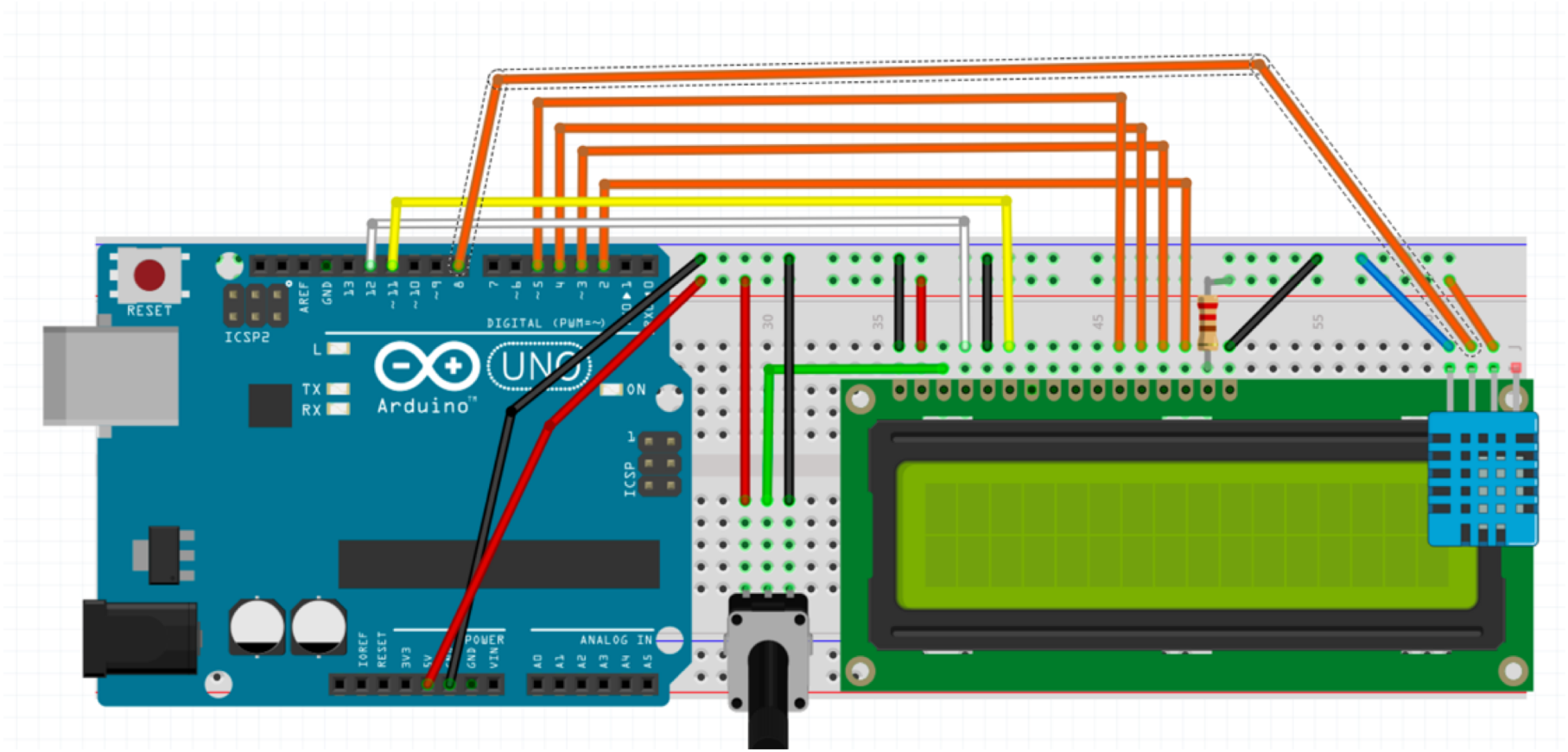


Anomalous Reading

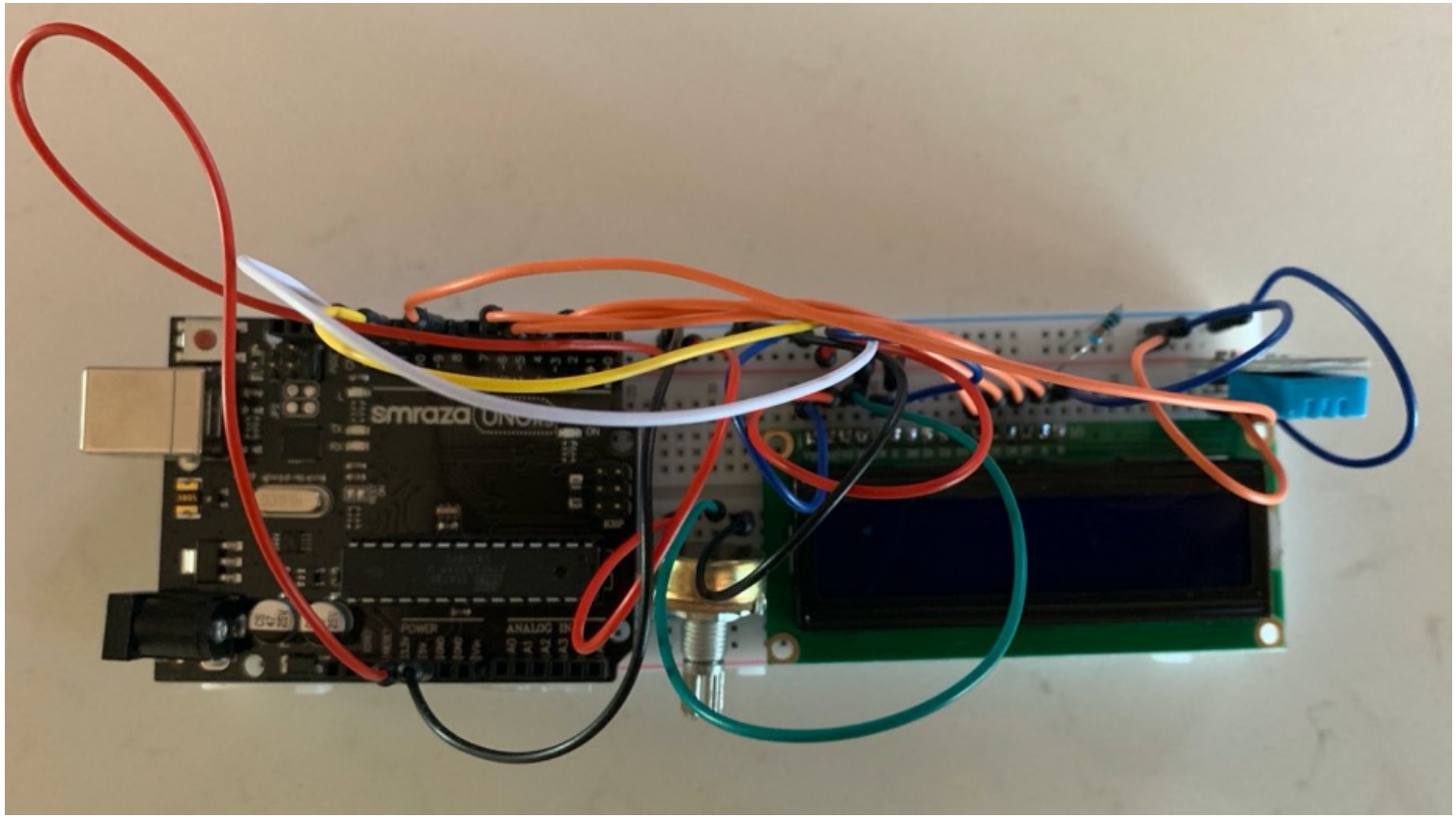
Combining Temperature and Humidity Sensor with LCD

- Instead of displaying the temperature and humidity on the serial terminal will we now display it on the LCD display

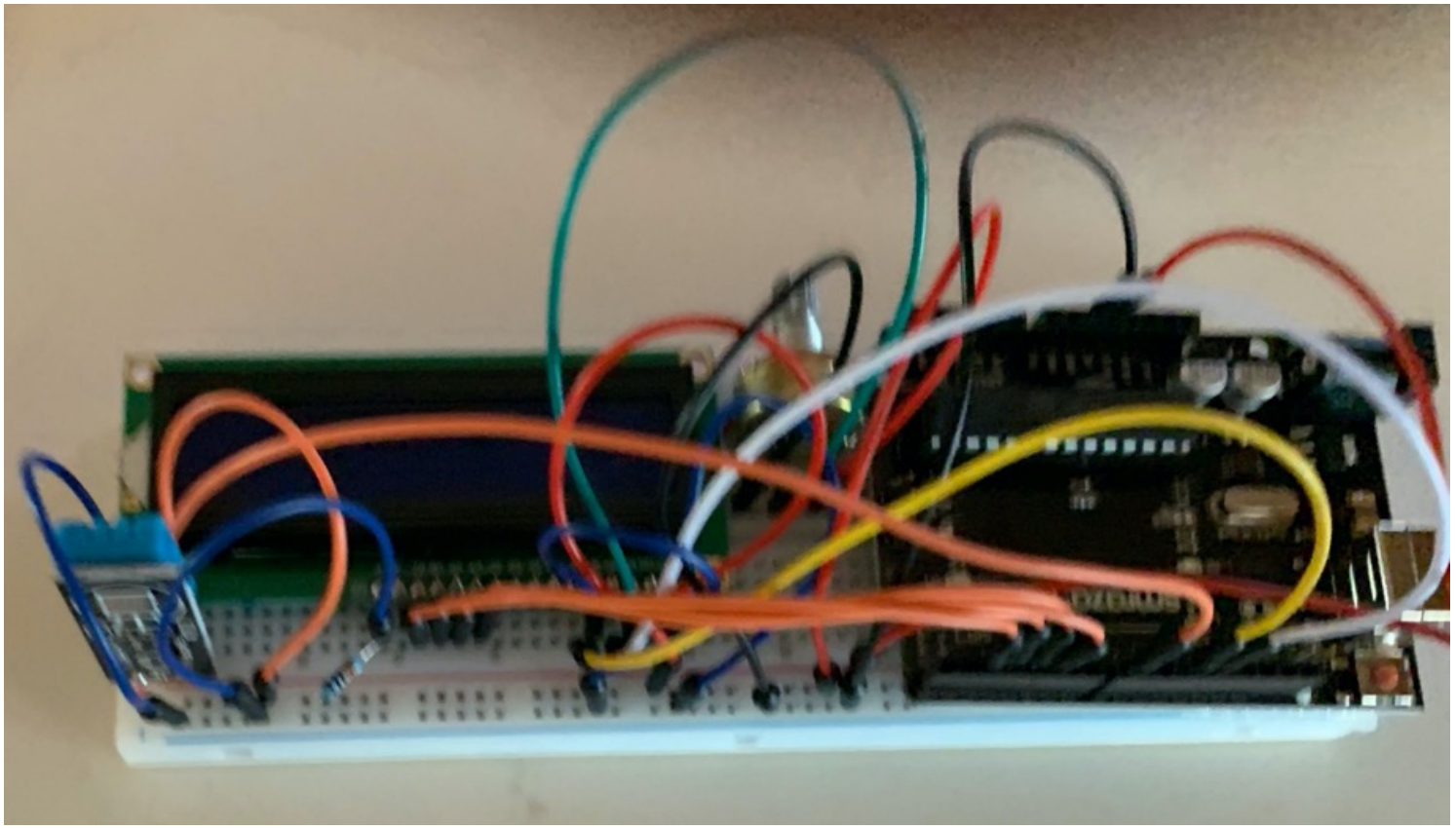
Temperature and Humidity Sensor with LCD of Wiring



Temperature and Humidity Sensor with LCD experimental setup



Temperature and Humidity Sensor with LCD experimental setup (another view)



Code

lcd_w_dht11 | Arduino 1.8.3

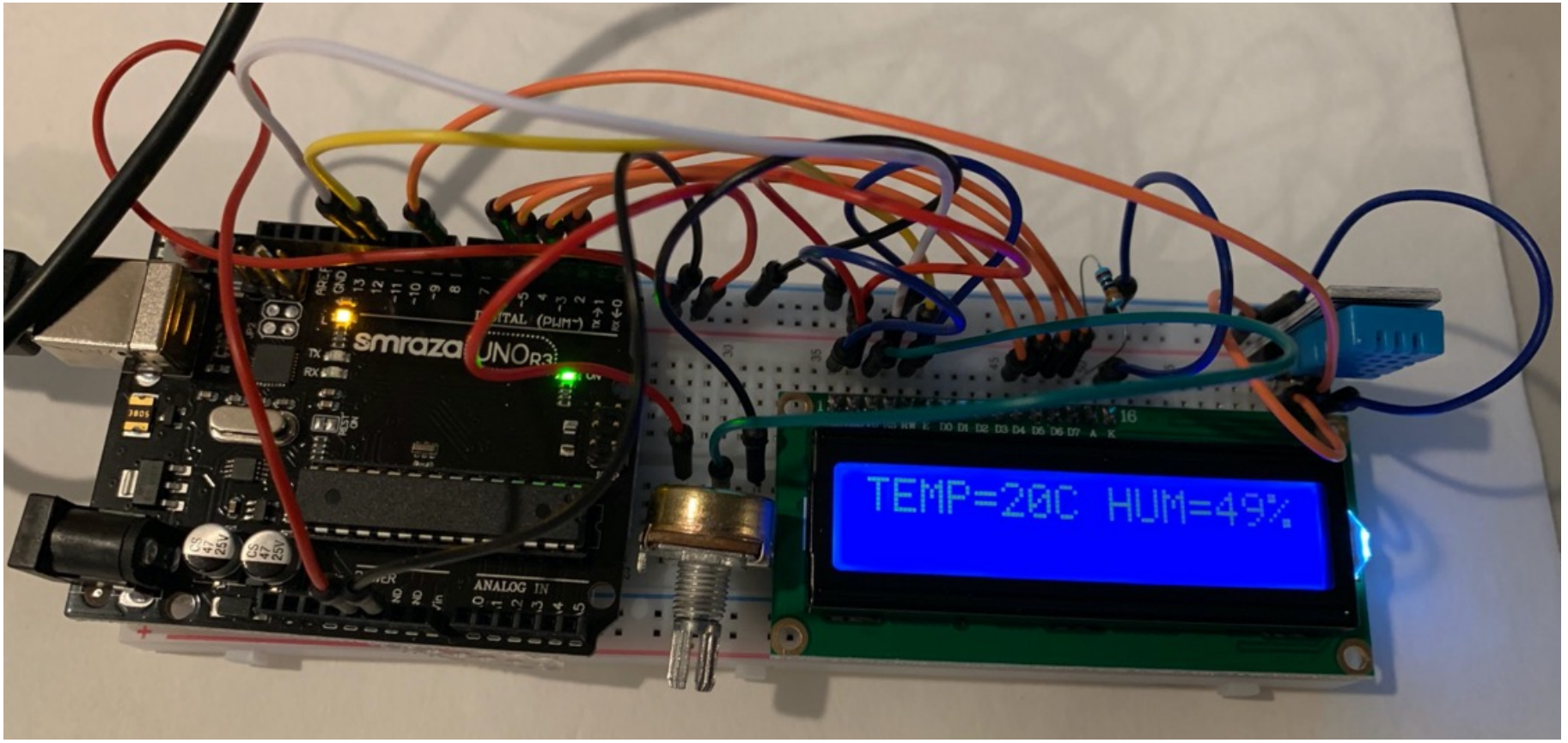
lcd_w_dht11

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
#include <dht11.h>
dht11 DHT11;
#define DHT11PIN 8
void setup()
{
// set up the LCD's number of columns and rows:
  lcd.begin(16,2);
 // Serial.begin(9600);

}
void loop()
{
  DHT11.read(DHT11PIN);
  lcd.begin(16,2);
  lcd.print("TEMP=");
  lcd.print(DHT11.temperature);
  lcd.print("C");
  lcd.print(' ');
  lcd.print("HUM=");
  lcd.print(DHT11.humidity);
  lcd.println("%");
  delay(500);
 }
```

Arduino/Genuino Uno on /dev/cu.usbmodem14731

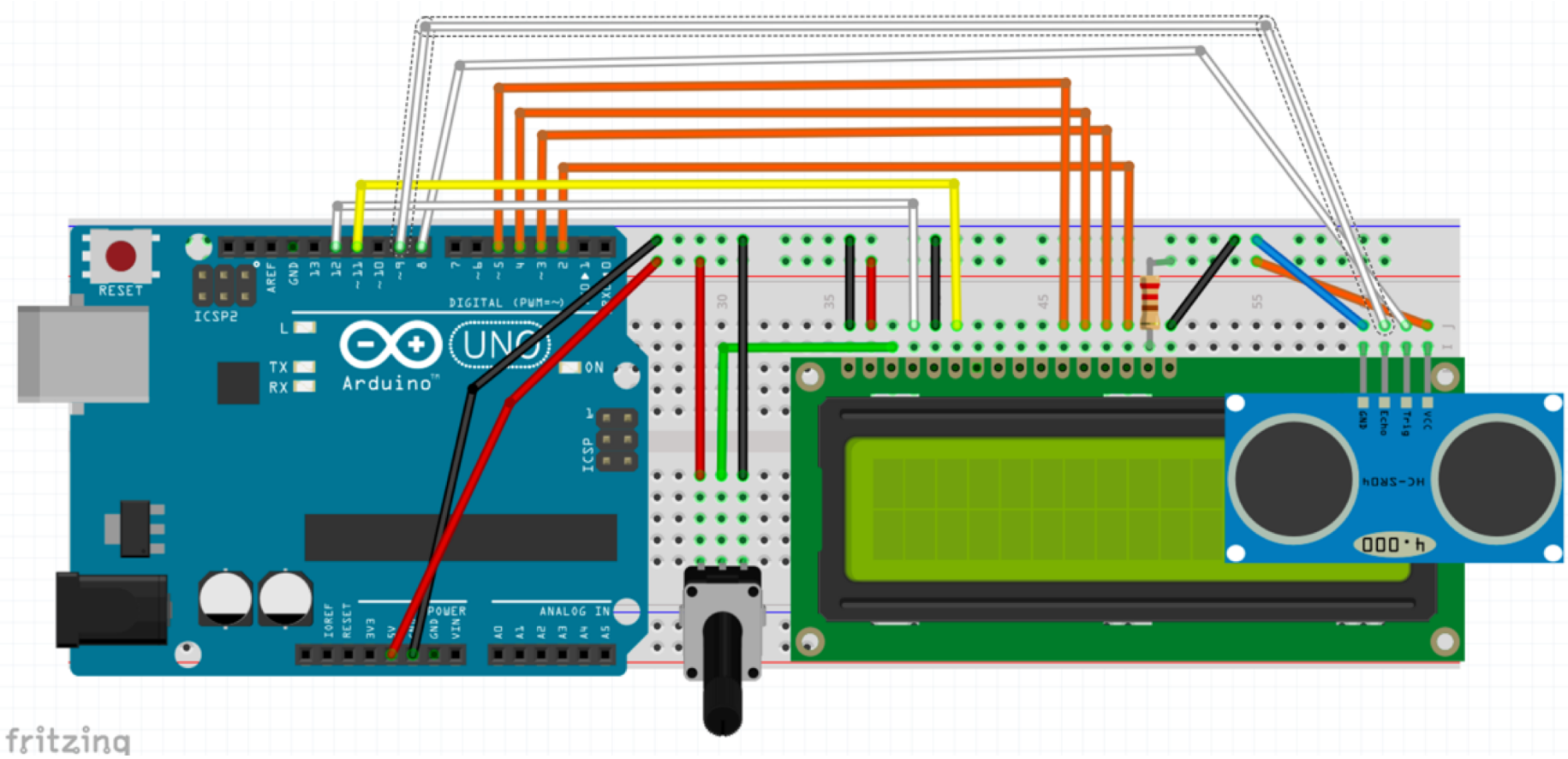
Sample Output



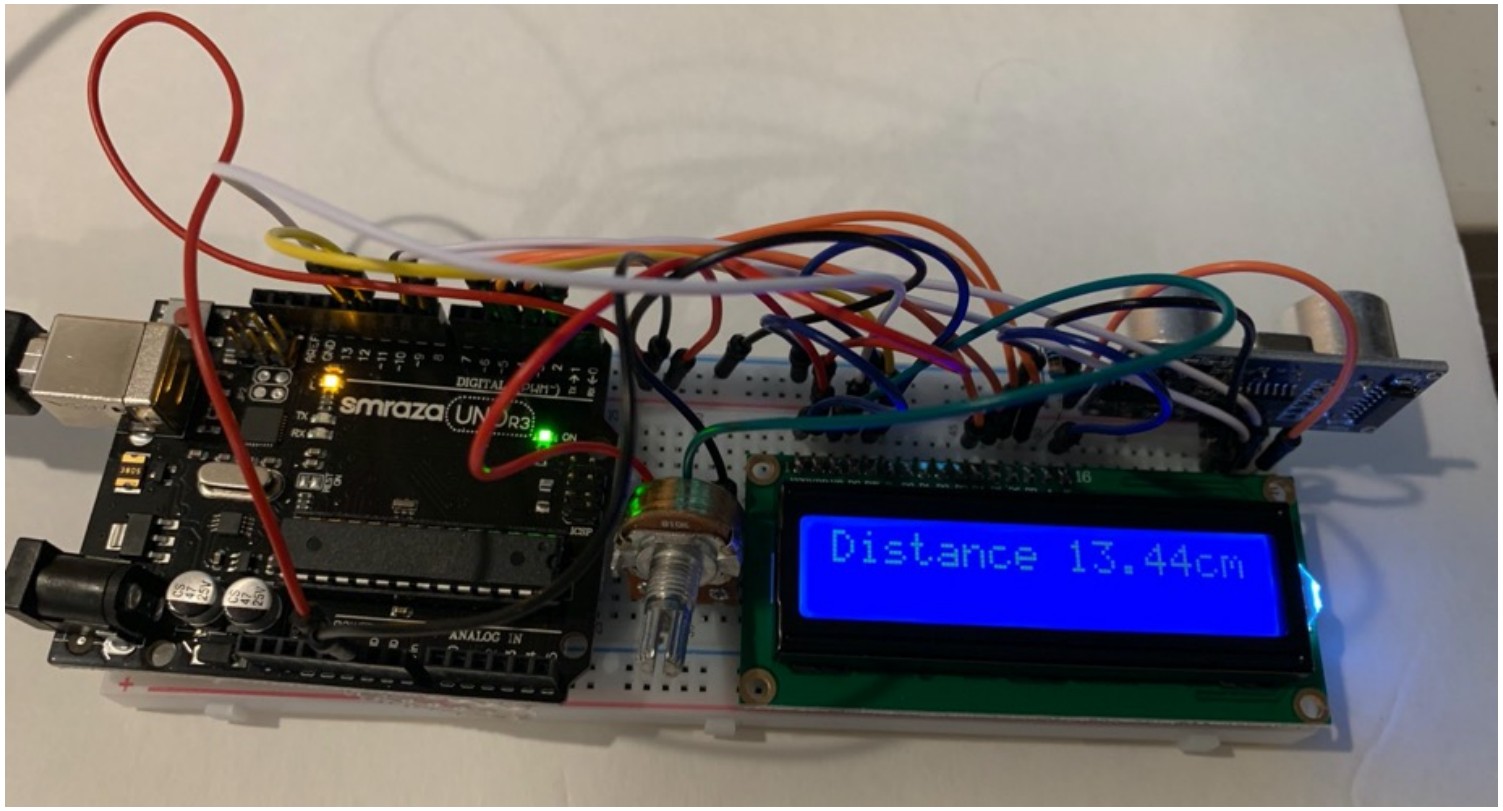
Display Distance Sensor on LCD

- We will combine ultra sonic sensor with LCD display
- The ultra sonic sensor uses pins 2 and 3
- The LCD display uses pins 2 and 3
- So we will change the ultra sonic sensor to use pins 8 and 9

LCD display of range experimental setup



LCD display of range experimental setup



LCD display of range code

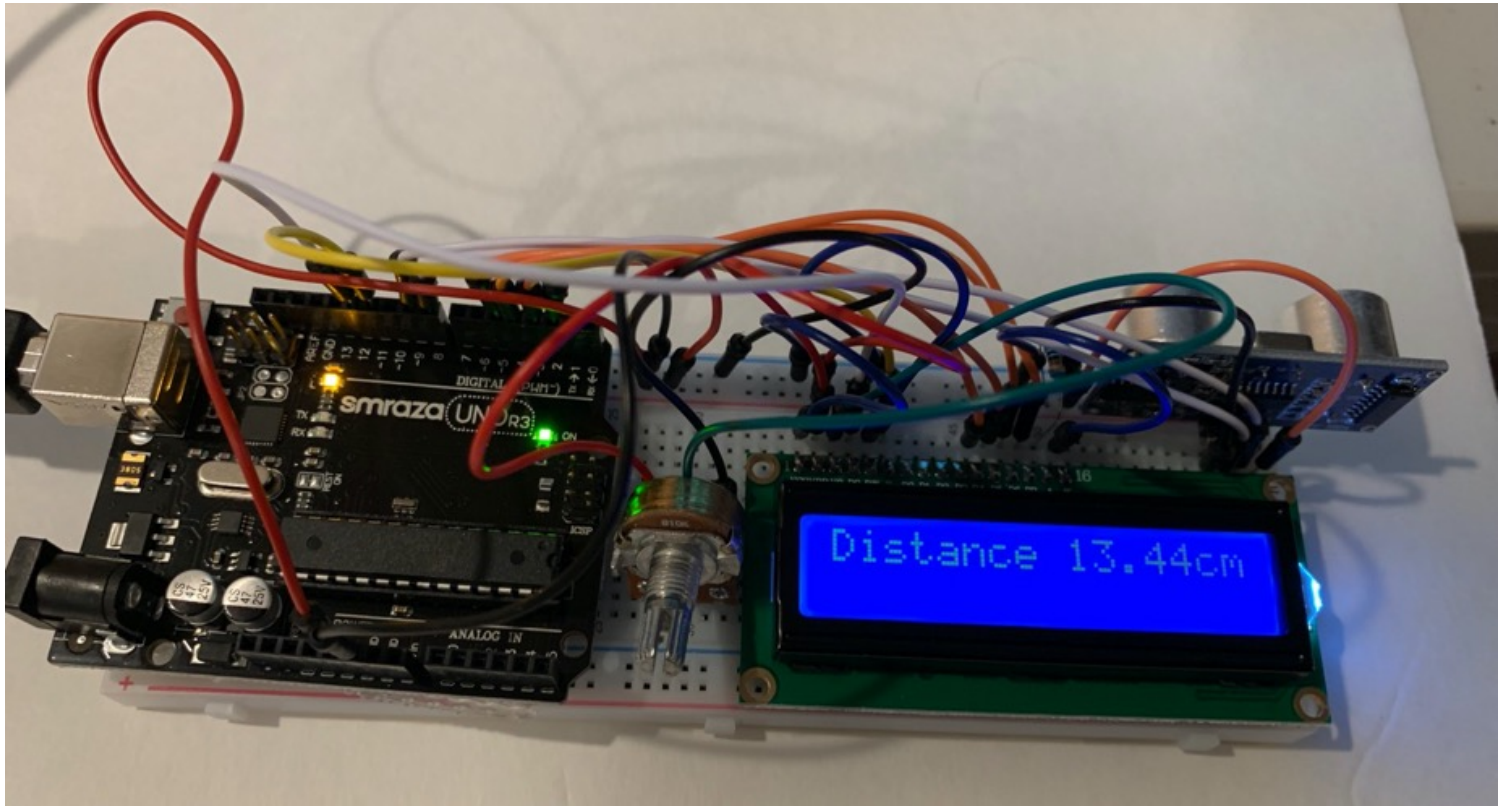
LCD1602_ultrasonic | Arduino 1.8.16

LCD1602_ultrasonic

```
// initialize the library with the numbers of the interface pins
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
const int TrigPin = 8;
const int EchoPin = 9;
float cm;
void setup()
{
// set up the LCD's number of columns and rows:
  lcd.begin(16,2);
pinMode(TrigPin, OUTPUT);
pinMode(EchoPin, INPUT);
}
void loop()
{
// Turn onf the display:
  lcd.display();
  lcd.begin(16,2);
digitalWrite(TrigPin, LOW);
delayMicroseconds(2);
digitalWrite(TrigPin, HIGH);
delayMicroseconds(10);
digitalWrite(TrigPin, LOW);
cm = pulseIn(EchoPin, HIGH) / 58.0; //The echo time is converted into cm
cm = (int(cm * 100.0)) / 100.0;     //Keep two decimal places
lcd.print("Distance ");
lcd.print(cm);
lcd.print("cm");
lcd.println();
delay(1000);
 }
```

1 Arduino Uno on /dev/cu.usbmodem147501

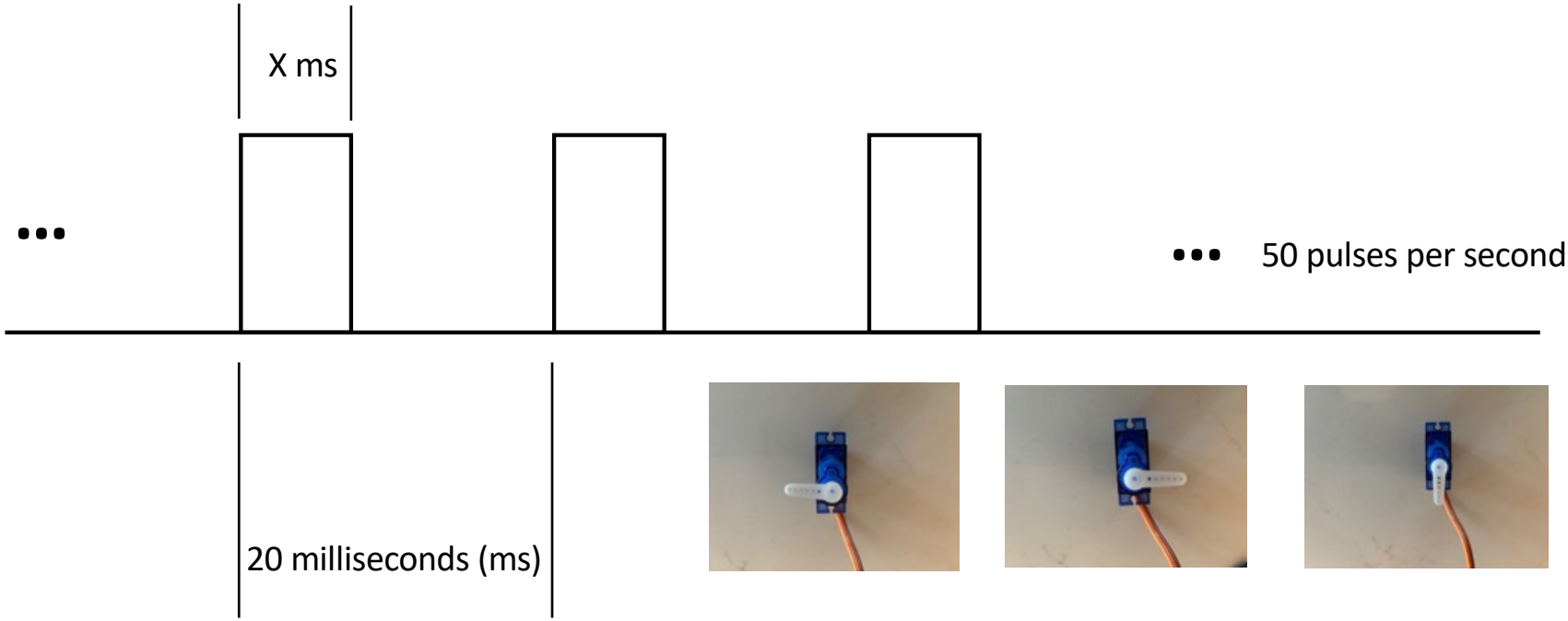
Sample Output of LCD display of range



Example controlling a Servo

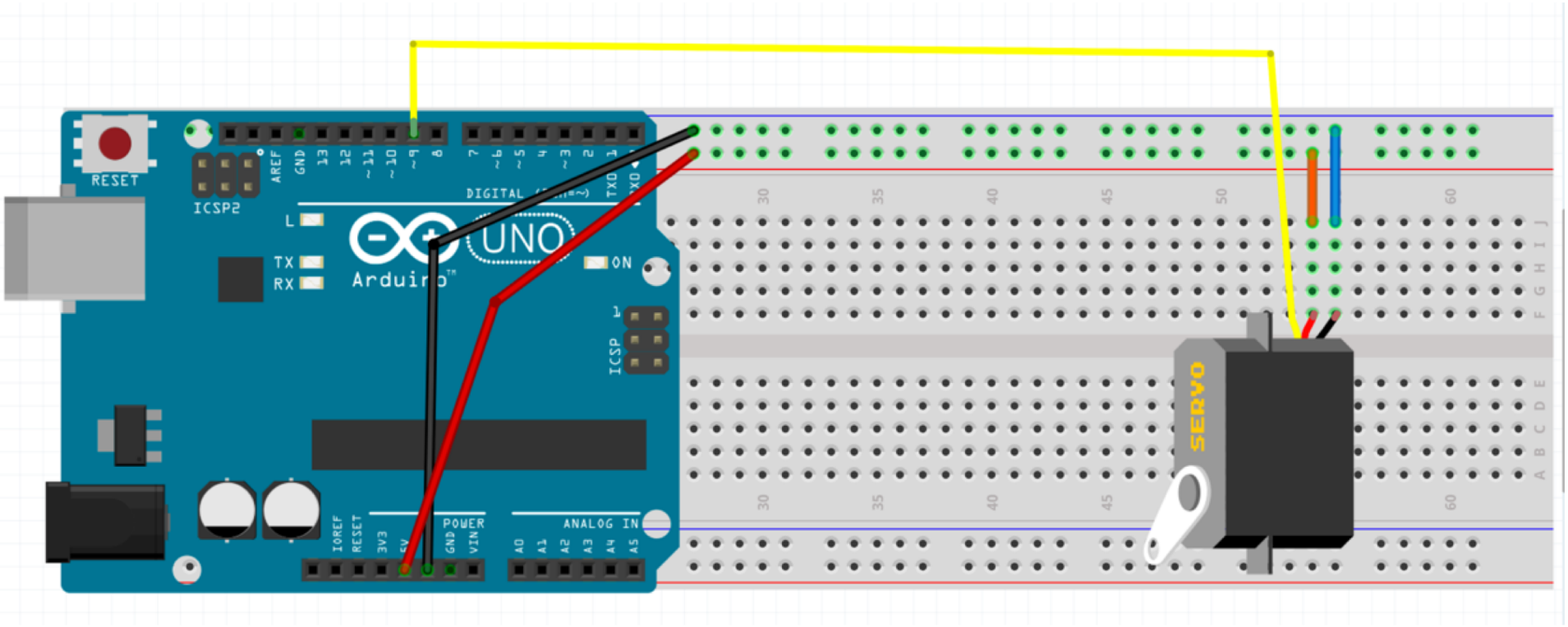
- Servos are used in radio controlled airplanes, cars and robots
- There is a standard way of controlling the servo
- Most Servos meet this standard
- Some deviate slightly from the standard
- Servos change their position based on a signal sent to them
  - The position can range from -90 to 0 to +90 degrees
- Some servos are modified to continuously rotate and thus become motors
  - The range of speed is from full backwards to stop to full forward
- The Arduino has commands to control servos

Standard Servo Operation



Pulse width	servo position	motor operation
X = 1.0 ms	-90 degrees	backwards - full speed
X = 2.0 ms	+90 degrees	forwards - full speed
X = 1.5 ms	0 degrees	stop

Servo Test Setup



Servo Sweep Code

Questions?

Post Class Projects

- If you haven't completed any of the experiments in class, try them after the class
- Look at the experiments for:
  - Buzzer
  - Speaker
  - Relay
  - RGB LED
- Try to:
  - If you know C, do the experiment
  - Try to combine individual experiments in one experiment