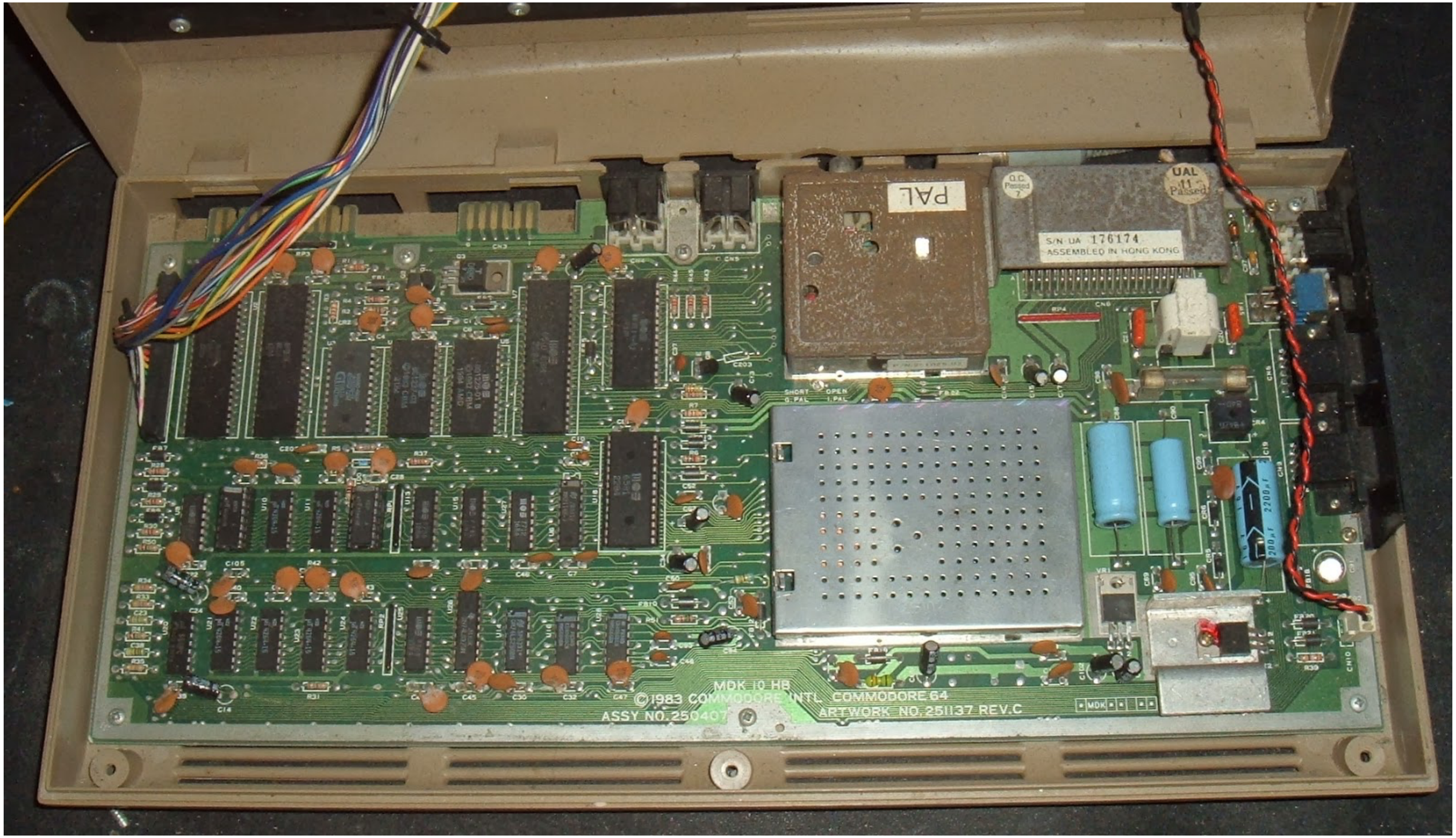
THE PERFECT APP

Me: Preben Thorö

- Software Pilot
- CTO
- Product Manager
- Coach
- Speaker
- Hobby Nerd
- …

Being a Hobby Nerd



Purpose

To show one of the easiest ways to make digital products

Arduino Introduction

Colombian student Hernando Barragán’s Master’s thesis in 2004: The Wiring Platform

“The Wiring platform consisted of a hardware PCB with an ATmega128 microcontroller, an integrated development environment (IDE) based on Processing and library functions to easily program the microcontroller.”
(from Wikipedia)

Arduino Introduction

2005: Some modifications (cheaper chip set),
and open source:

Arduino was born

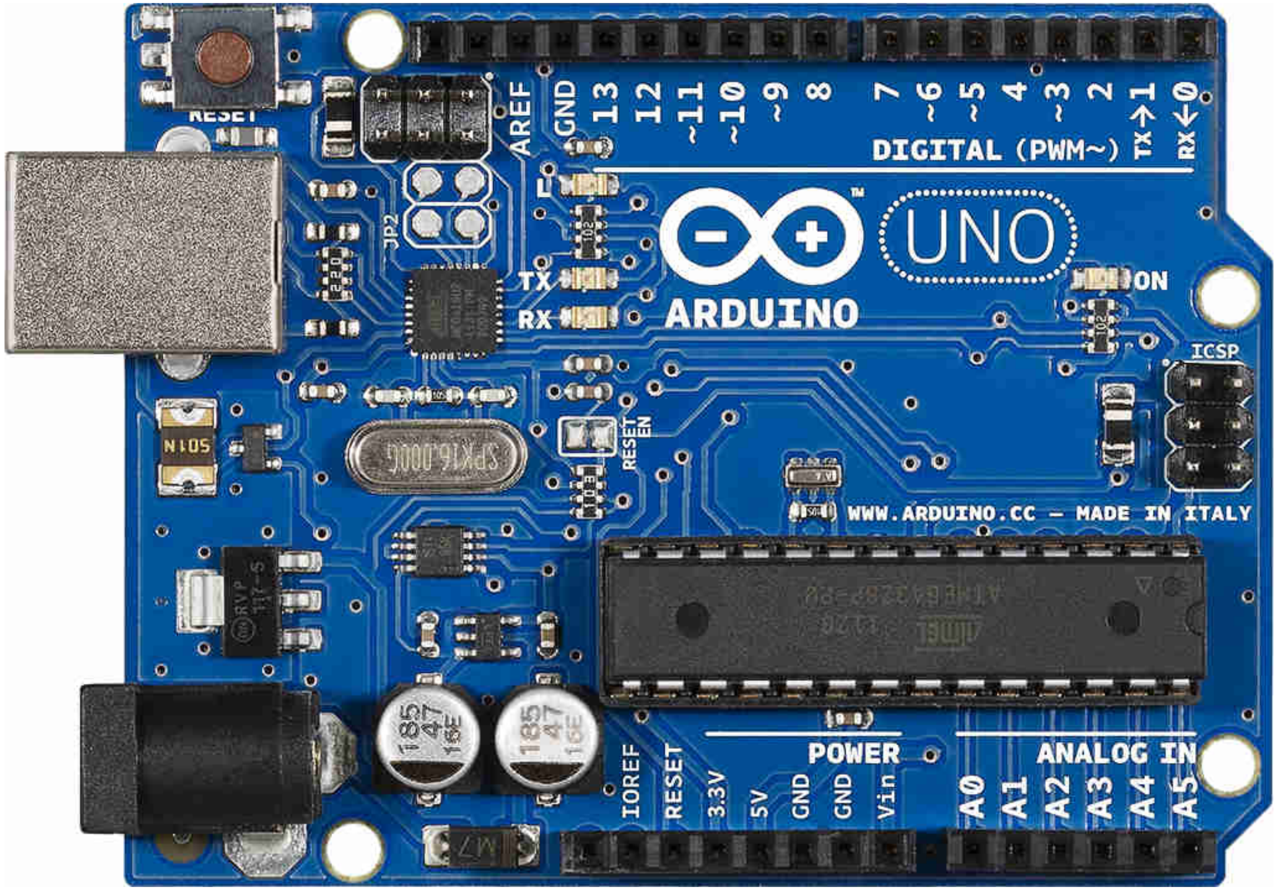
Arduino Introduction

Arduino is OpenSource Electronics

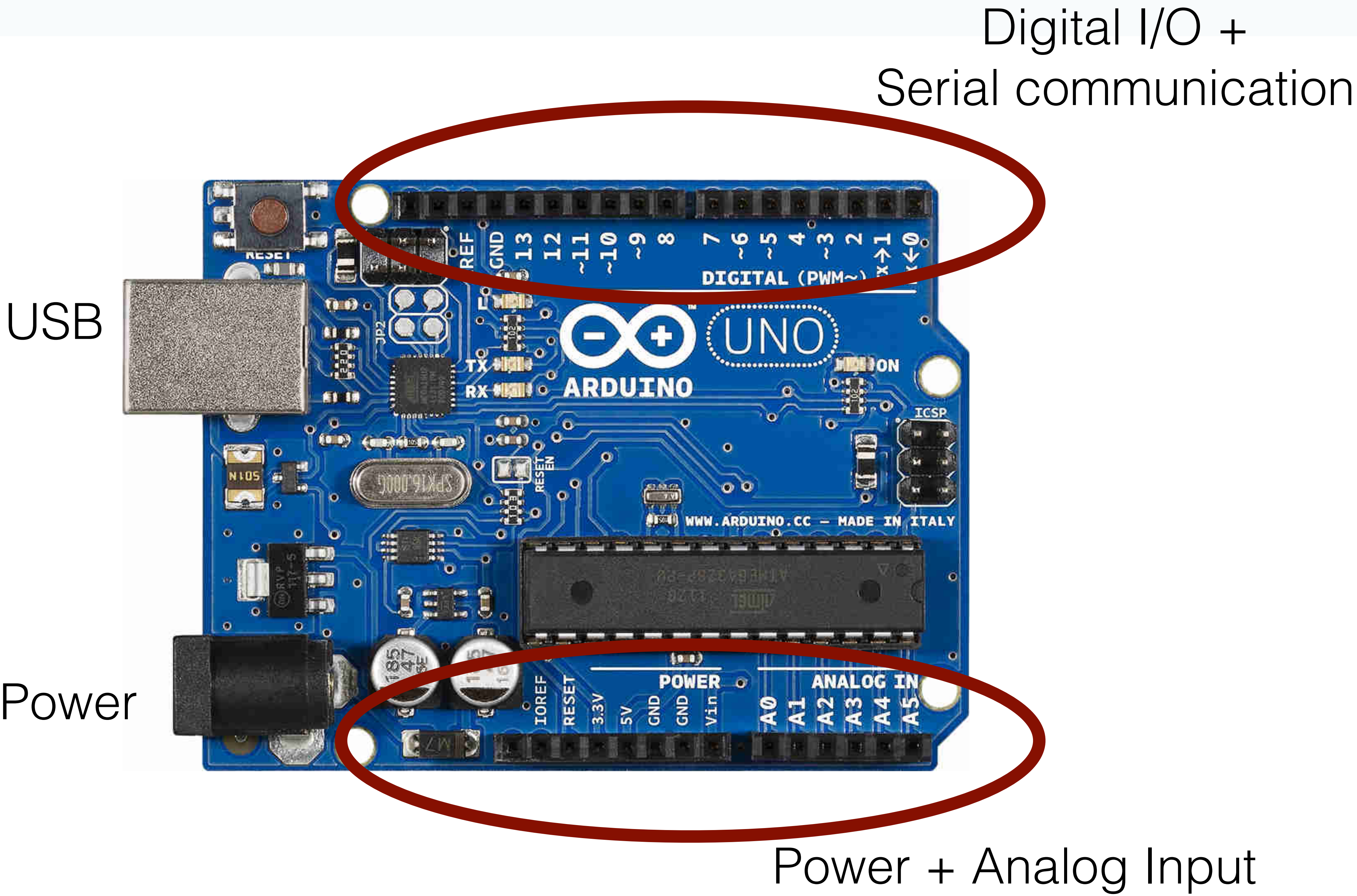
- has a huge community
- is used for educational purposes

It is so easy to get started!

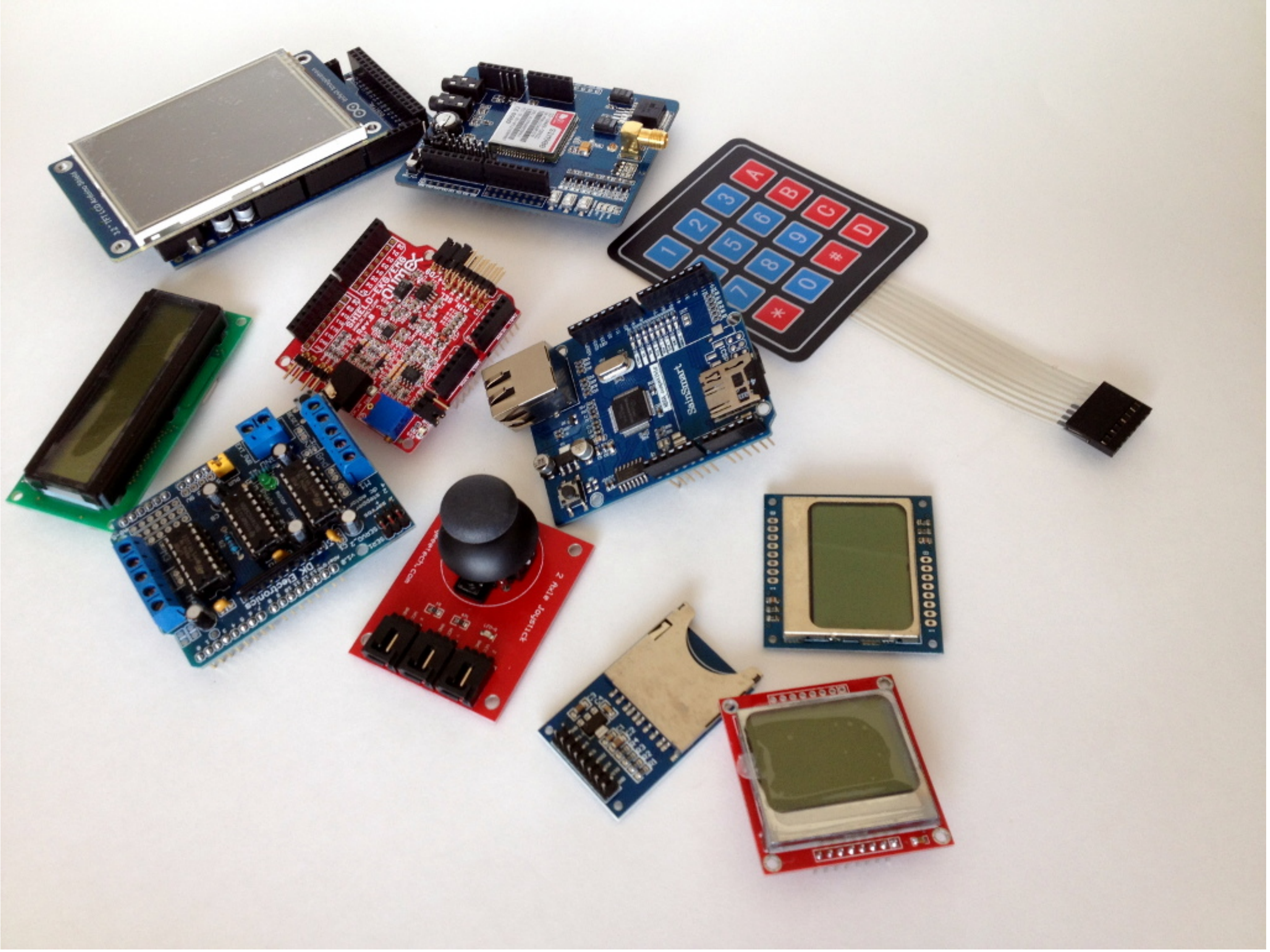
Arduino Introduction



Arduino Introduction



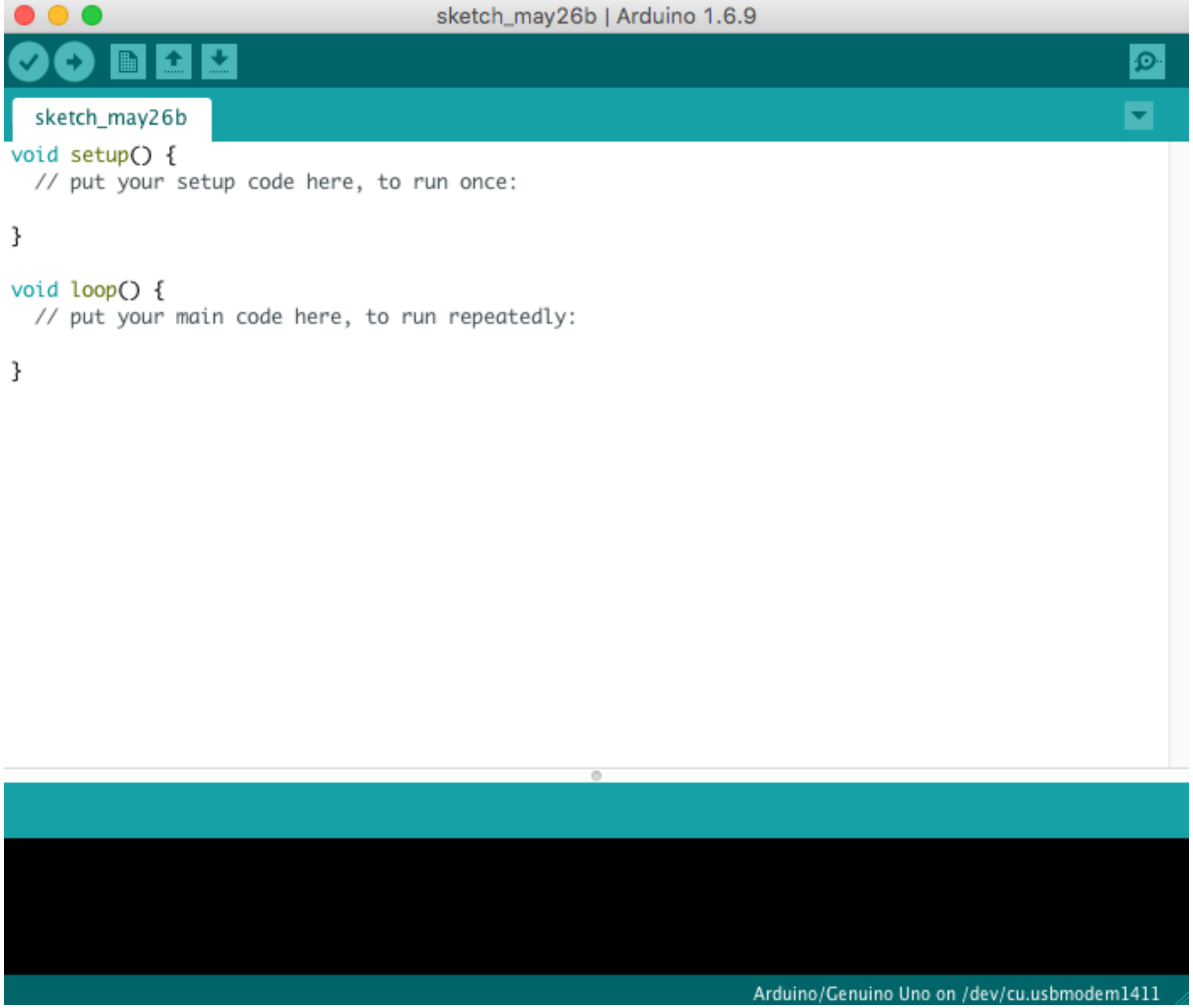
Arduino Introduction



Shields for

Network
GPS
High voltage
Bluetooth
Joystick
Display
Keyboard
SD cards
…

Programming the Arduino


sketch_may26b | Arduino 1.6.9
sketch_may26b
Arduino/Genuino Uno on /dev/cu.usbmodem1411

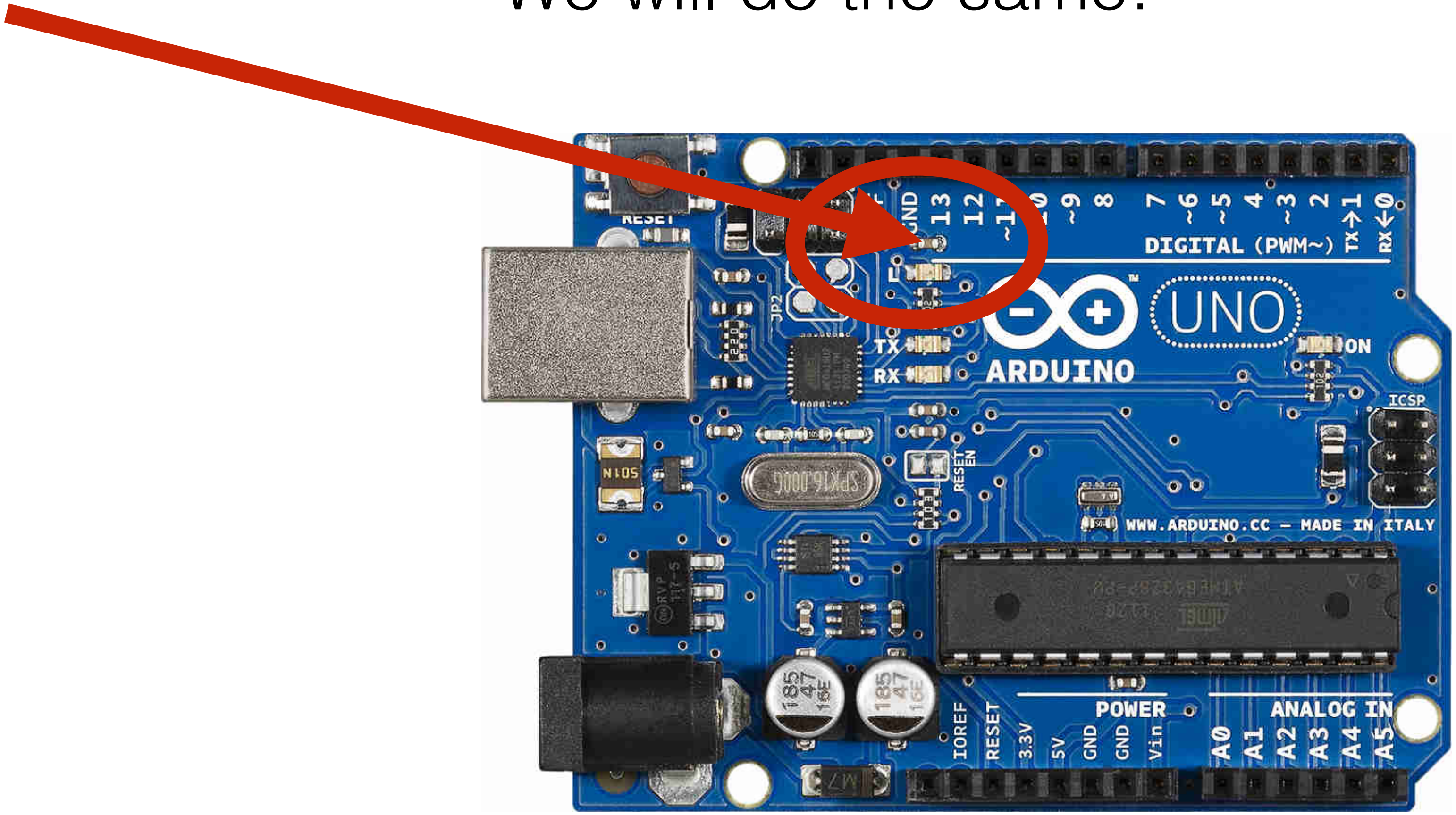
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Download from www.arduino.cc

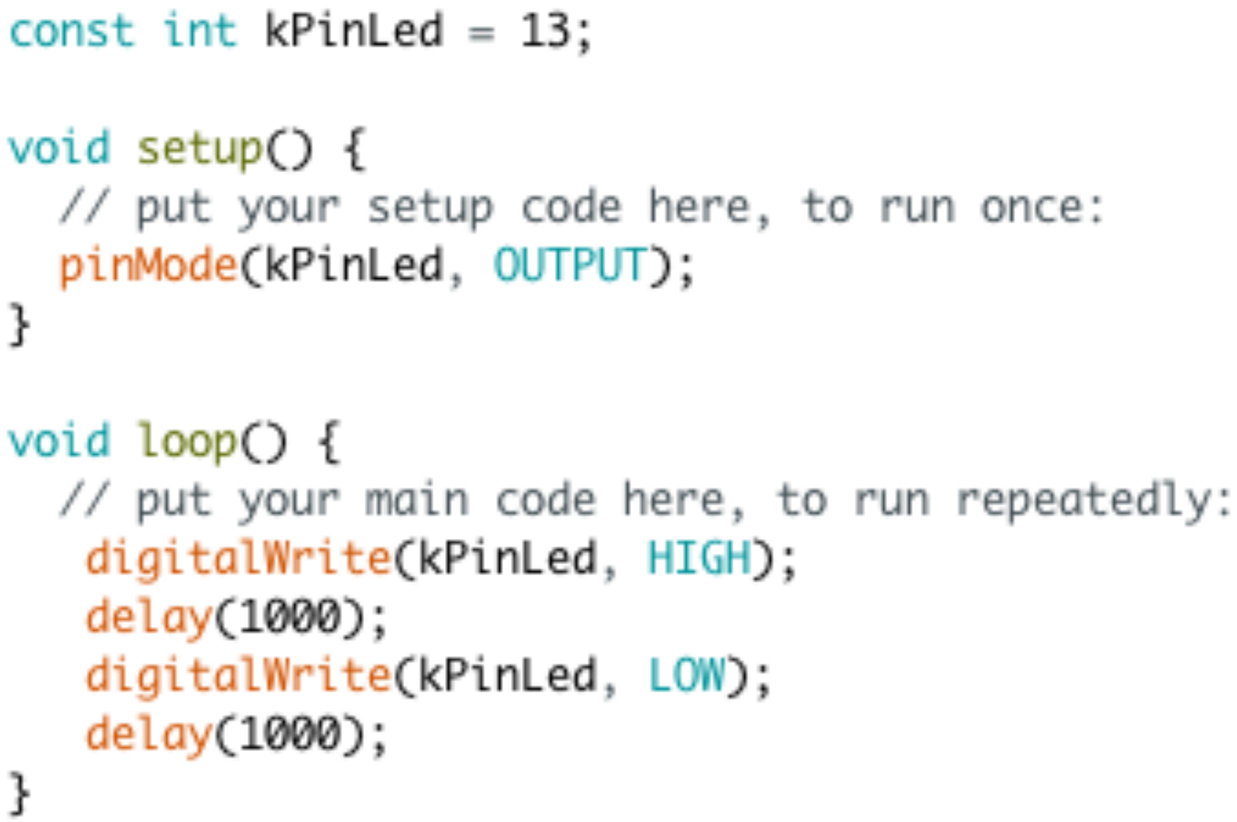
Hello World

Arduino ‘Hello World’ almost always control pin #13

We will do the same!

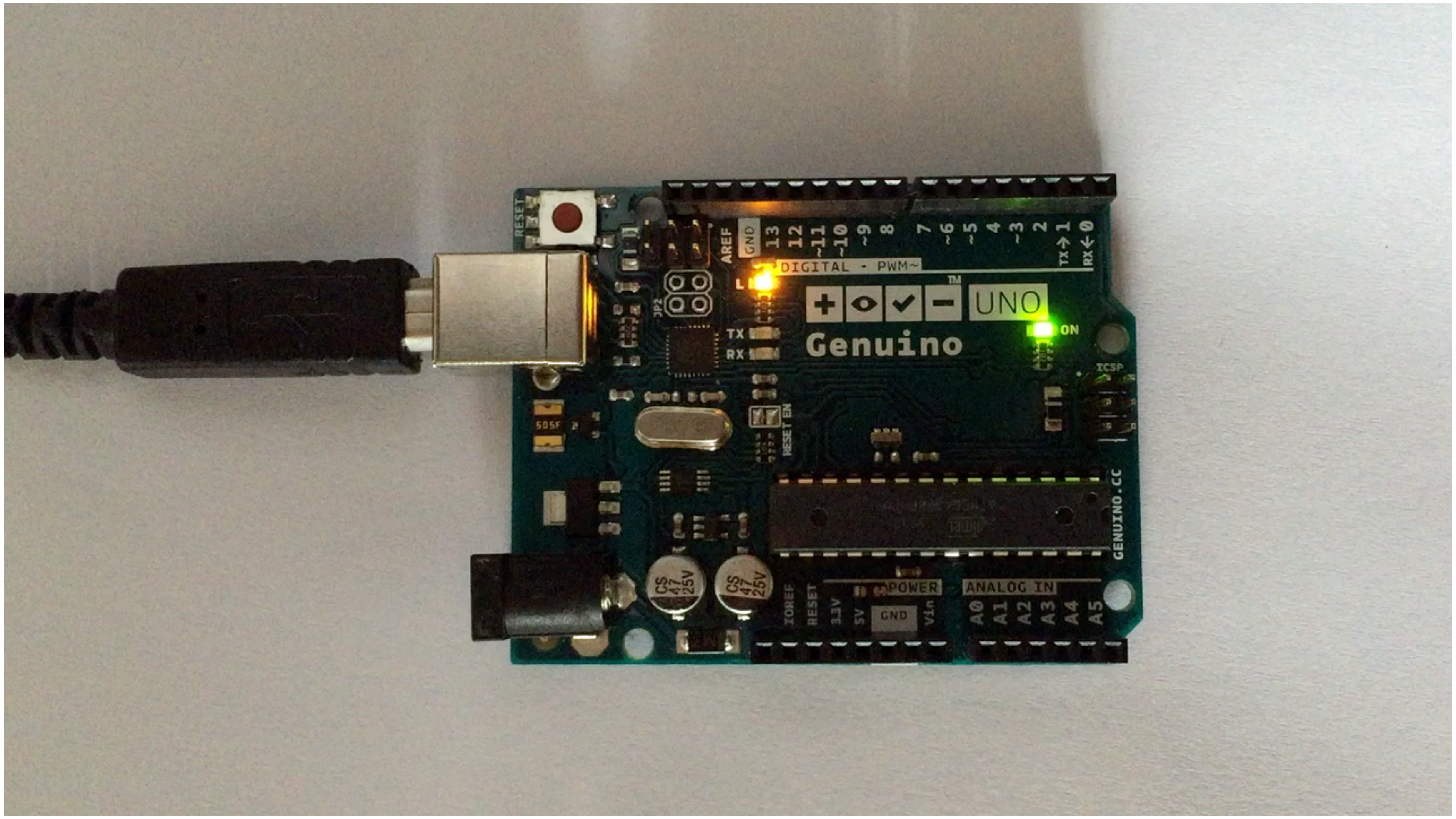


Hello World

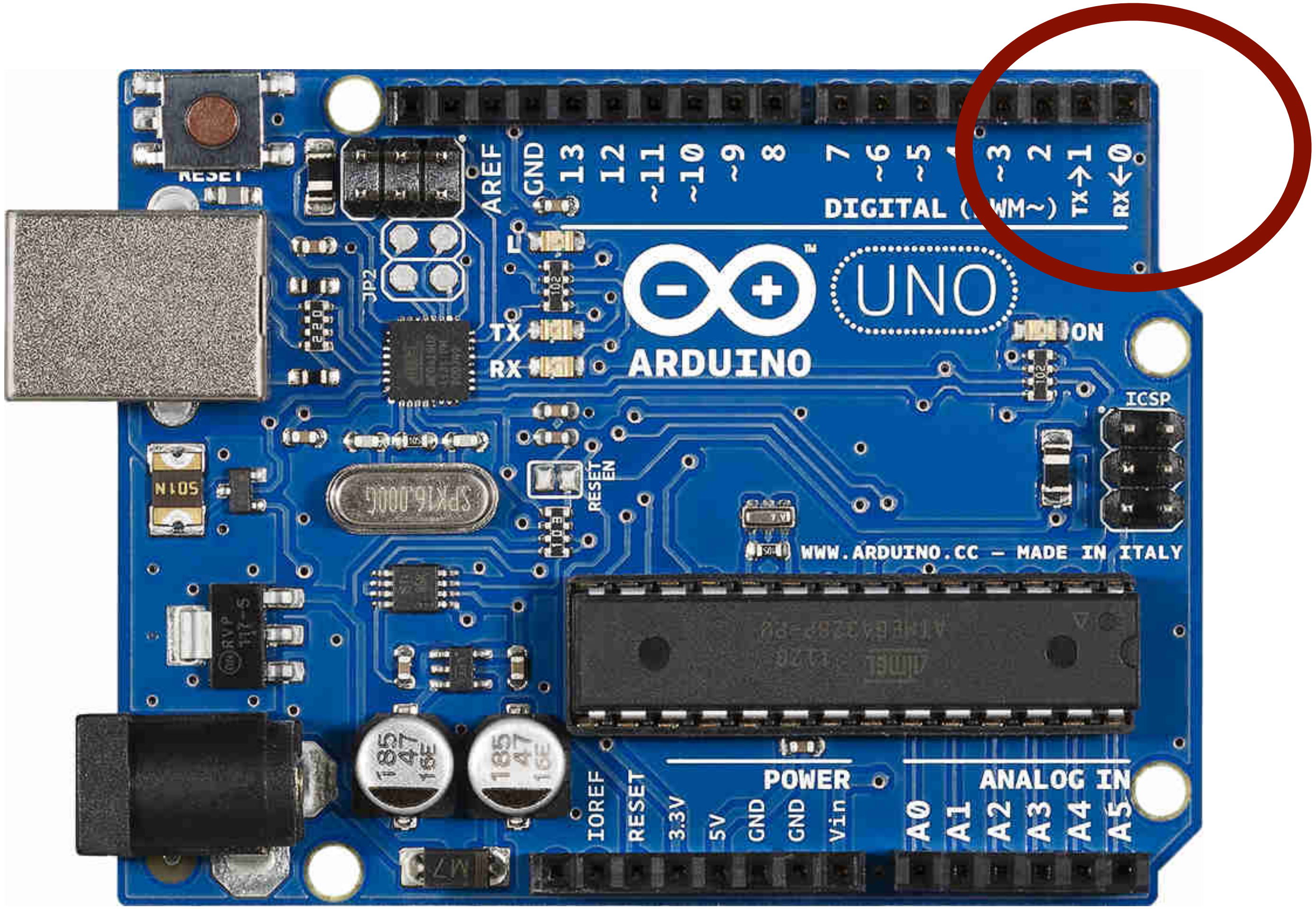


```
const int kPinLed = 13;

void setup() {
  // put your setup code here, to run once:
  pinMode(kPinLed, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
   digitalWrite(kPinLed, HIGH);
   delay(1000);
   digitalWrite(kPinLed, LOW);
   delay(1000);
}
```

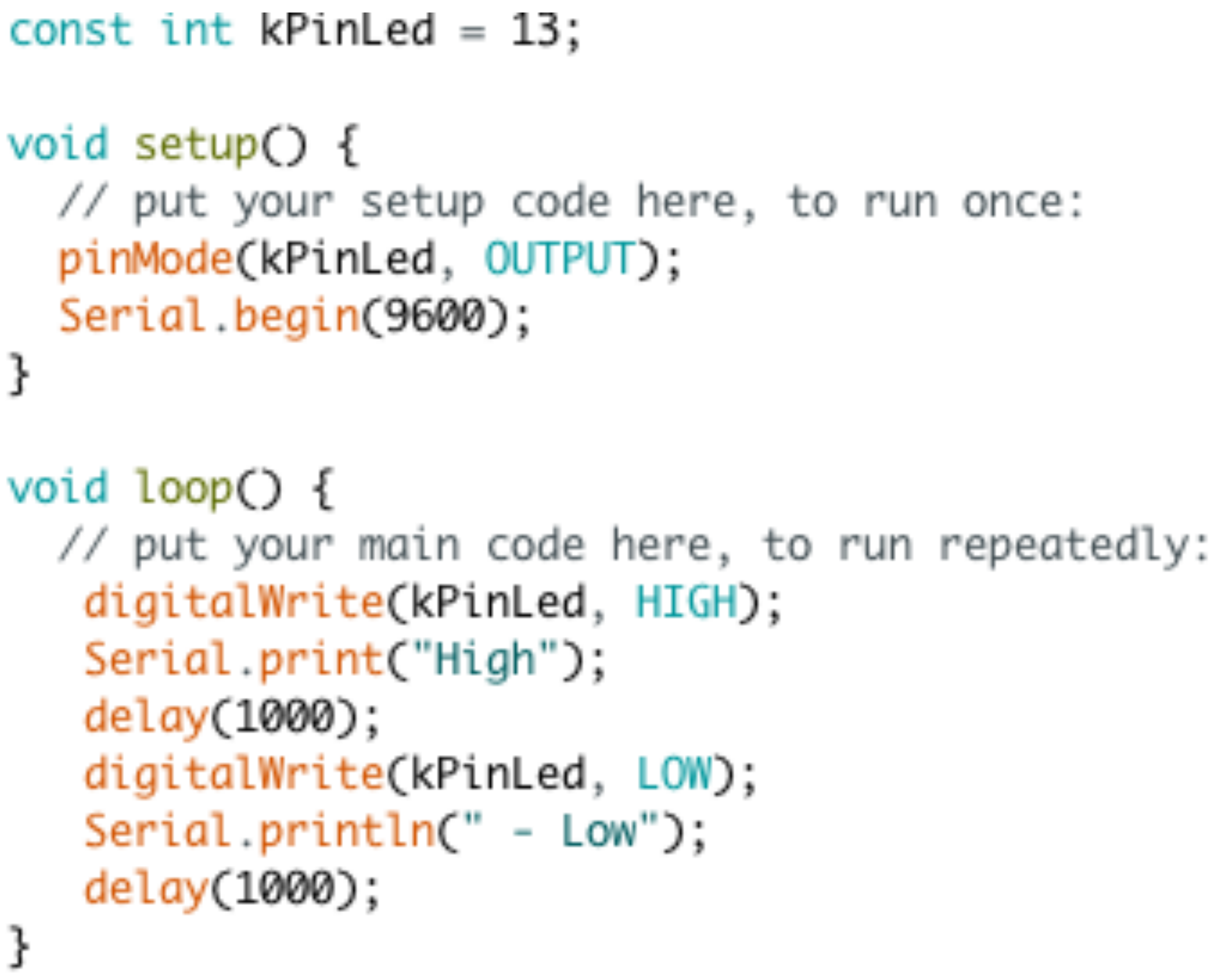
Hello World


RESET
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL - PWM~
L
TM
UNO
ON
TX
RX
Genuino
ICSP
JP2
RESET EN
GENUINO.CC
IOREF
RESET
3.3V
5V
GND
Vin
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5

Hello World part #2

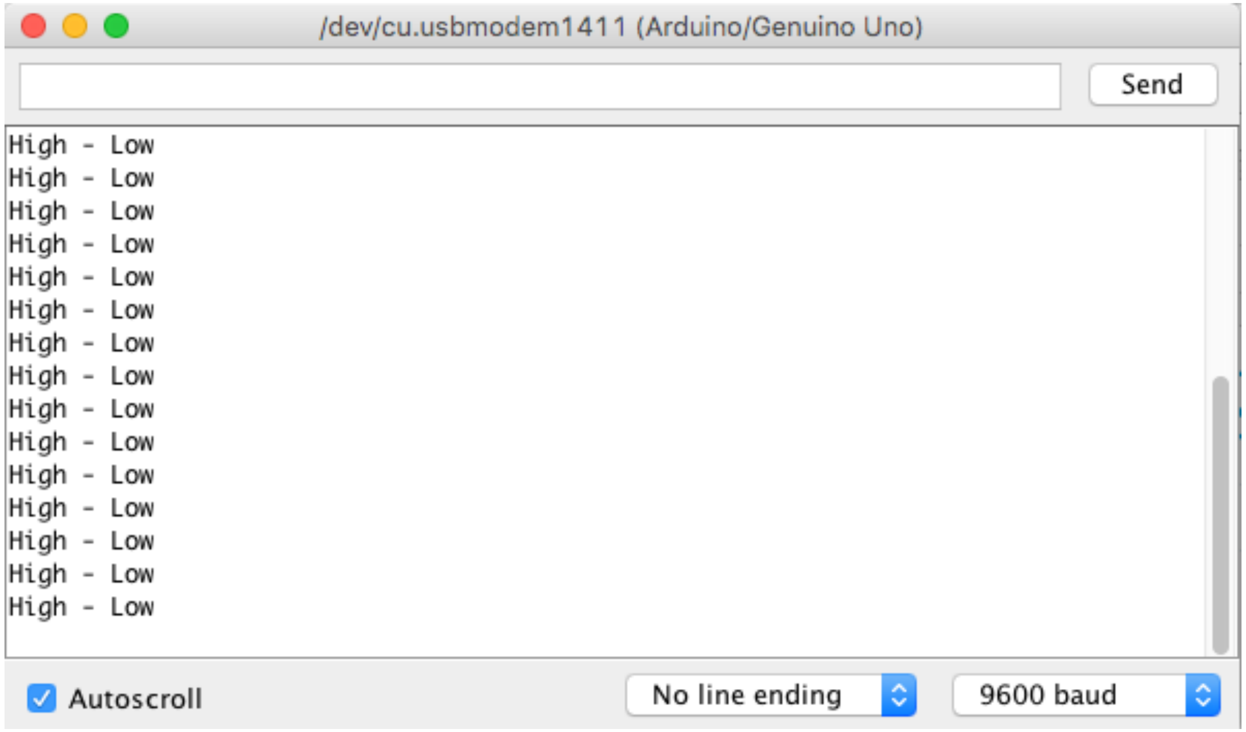


Hello World part #2



```
const int kPinLed = 13;

void setup() {
  // put your setup code here, to run once:
  pinMode(kPinLed, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
   digitalWrite(kPinLed, HIGH);
   Serial.print("High");
   delay(1000);
   digitalWrite(kPinLed, LOW);
   Serial.println(" - Low");
   delay(1000);
}
```

Hello World part #2



```
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
```

☑ Autoscroll

No line ending

9600 baud

Programming is C

for(..;..;..)

while

loop

print

sprintf

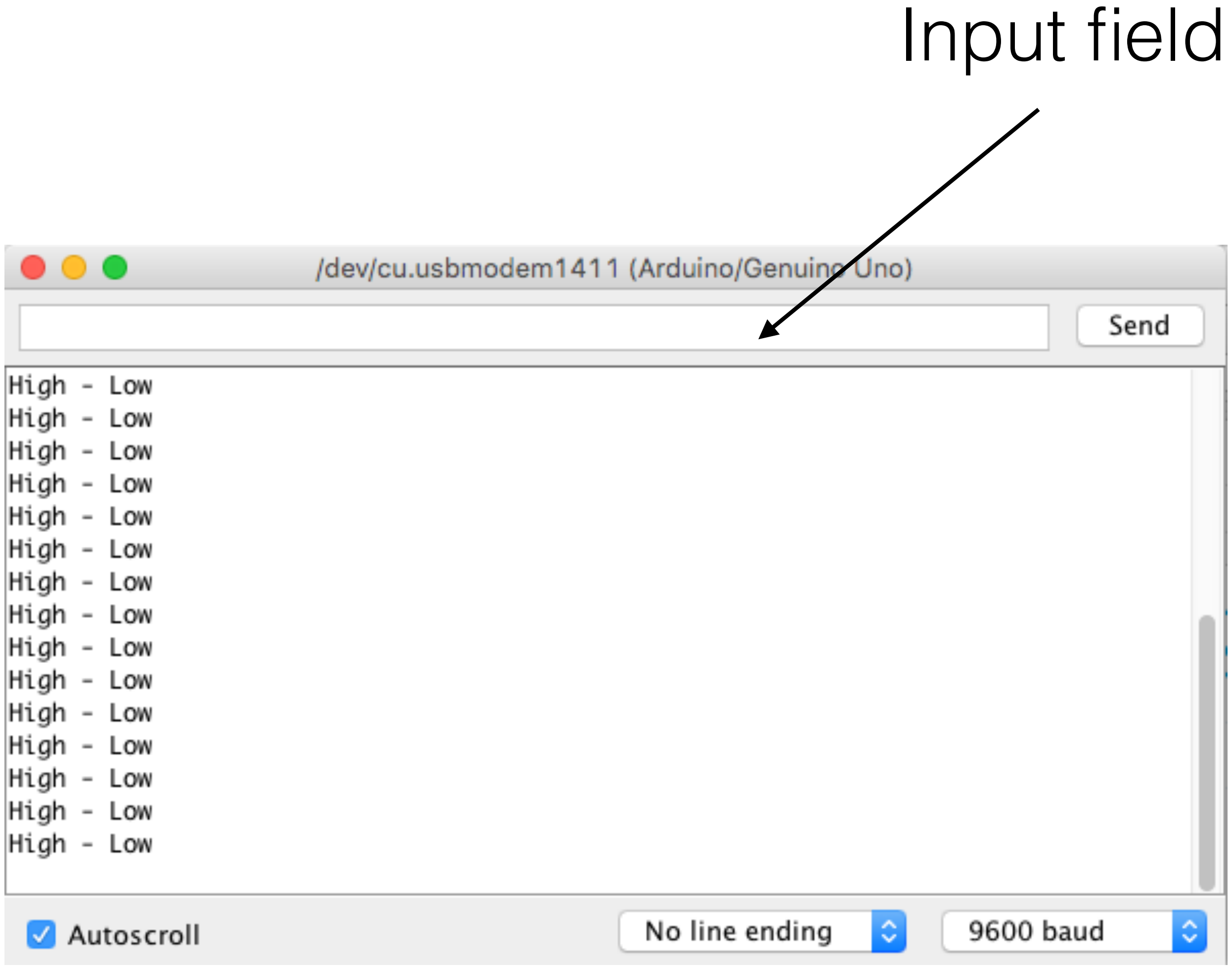
void

const

…

at least to the untrained eye

Reading and Writing

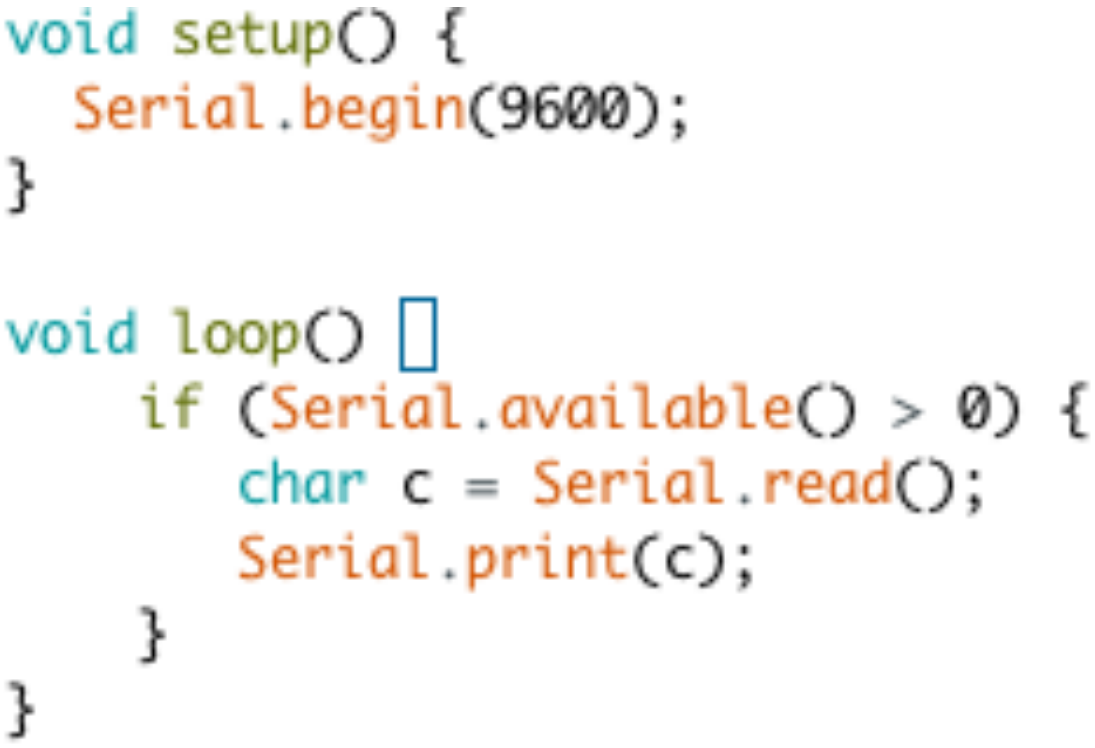


Input field

/dev/cu.usbmodem1411 (Arduino/Genuino Uno)

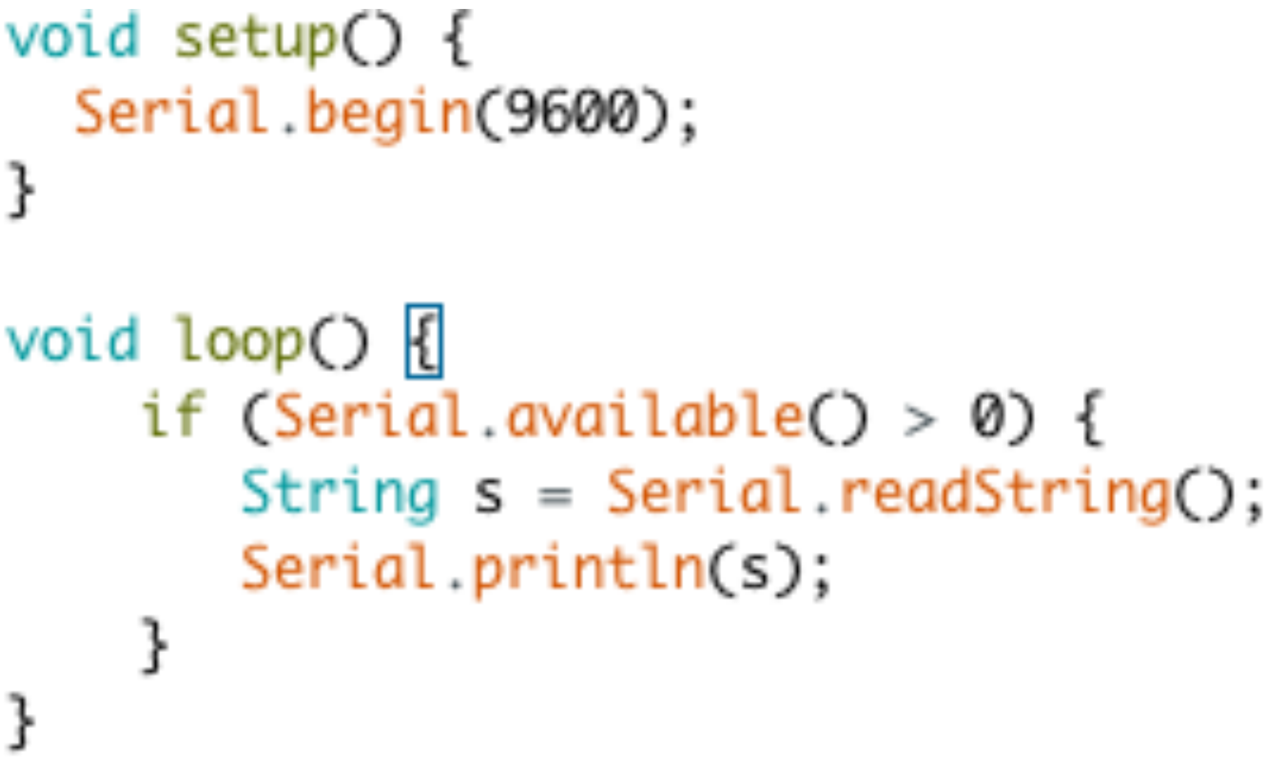
```
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
High - Low
```

Reading and Writing



```
void setup() {
  Serial.begin(9600);
}

void loop() □
    if (Serial.available() > 0) {
        char c = Serial.read();
        Serial.print(c);
    }
}
```

Reading and Writing

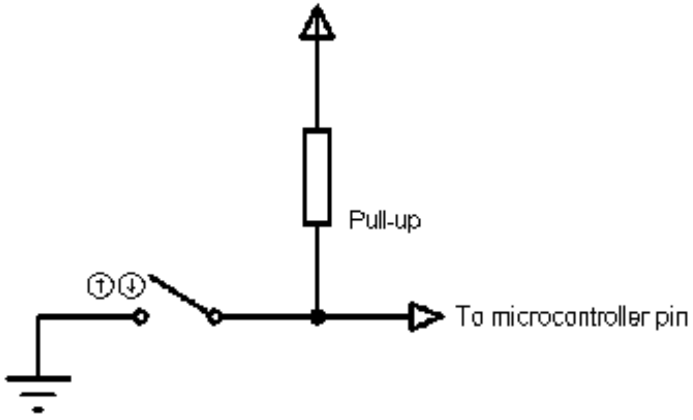


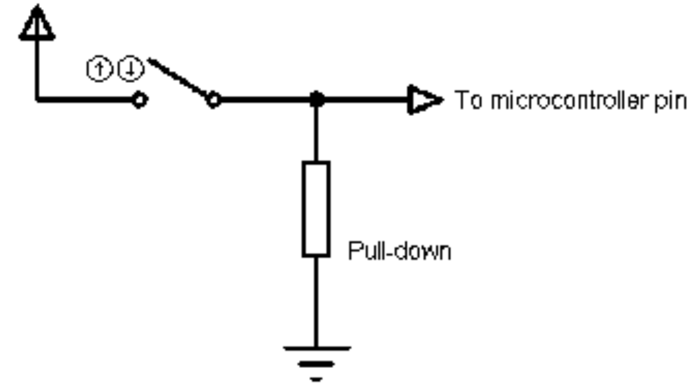
```
void setup() {
  Serial.begin(9600);
}

void loop() {
    if (Serial.available() > 0) {
        String s = Serial.readString();
        Serial.println(s);
    }
}
```

Reading Inputs

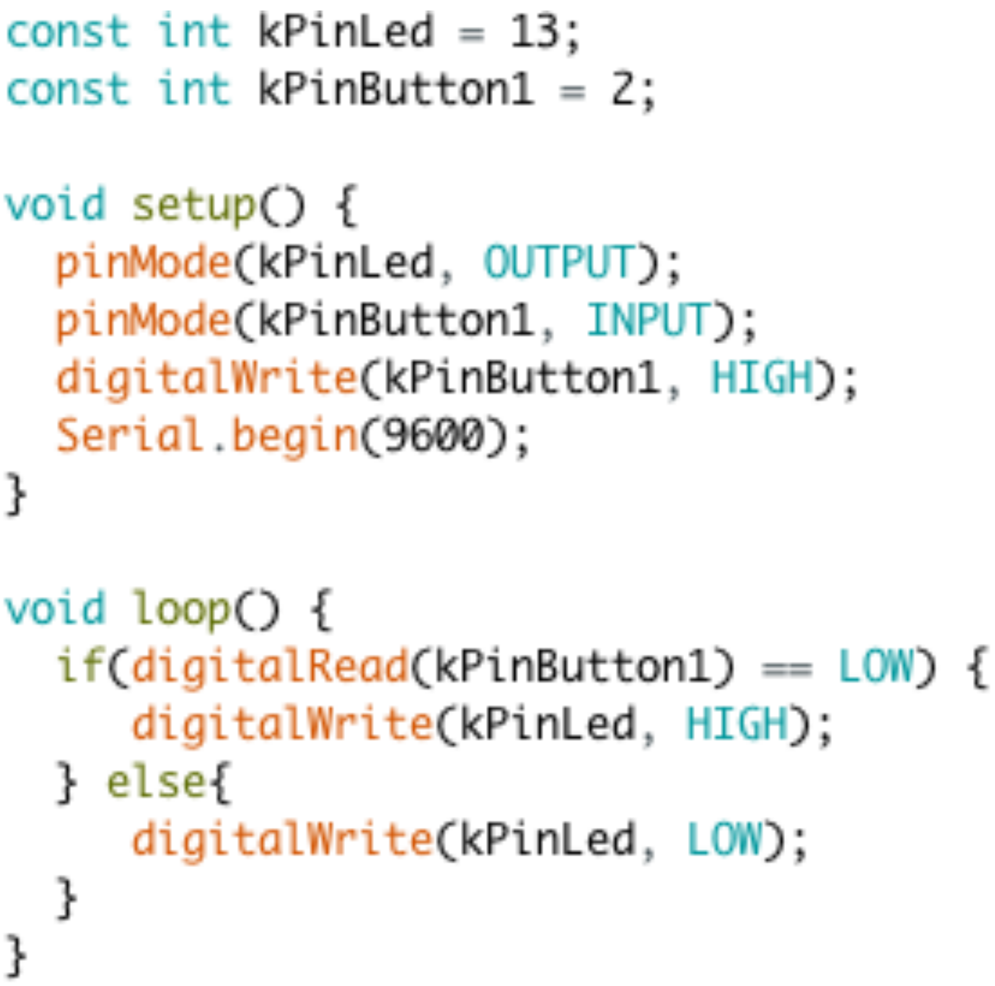
No pull-up/pull-down resistor needed

Reading Inputs



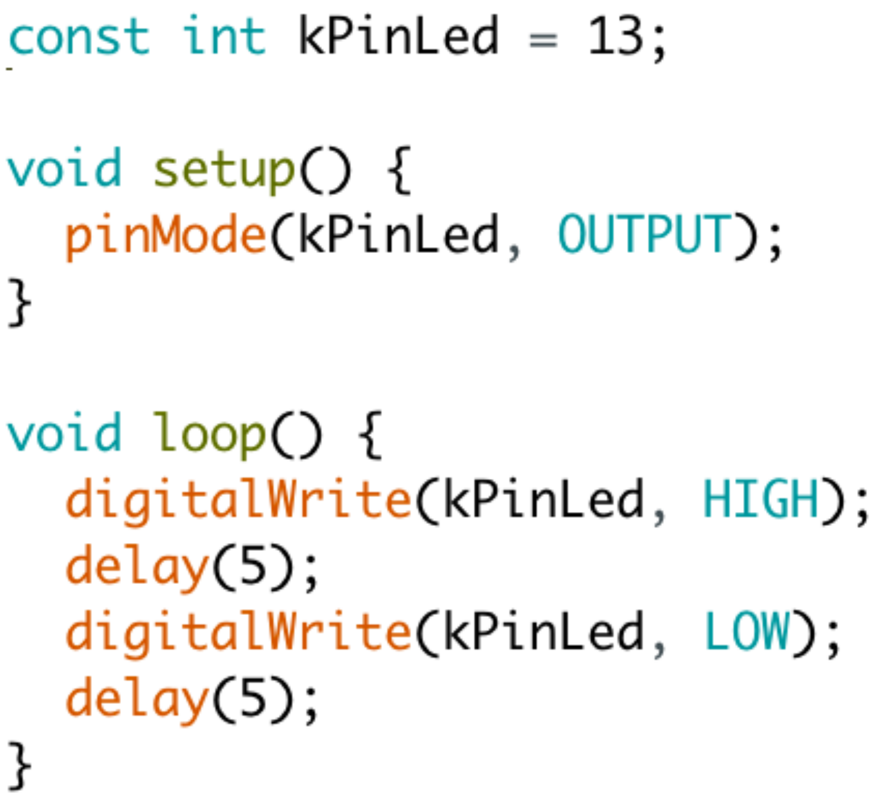


Reading Inputs



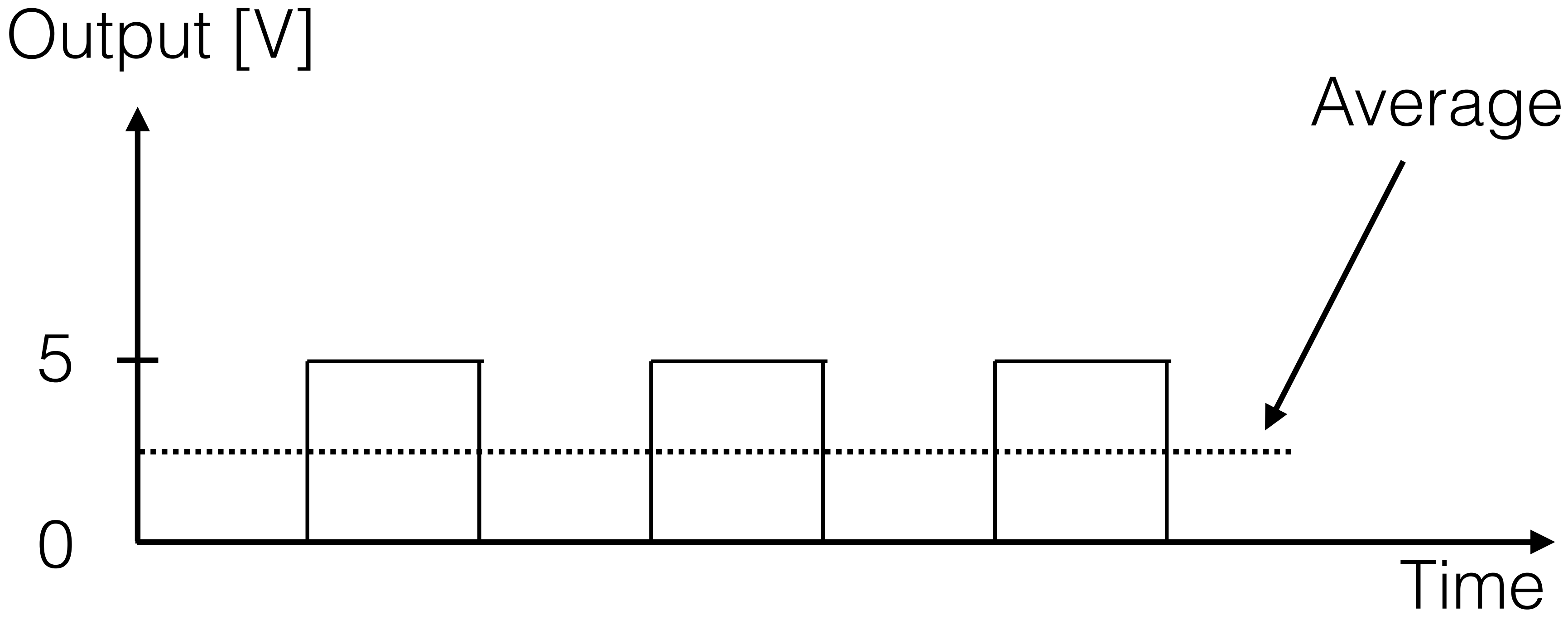
```
const int kPinLed = 13;
const int kPinButton1 = 2;

void setup() {
  pinMode(kPinLed, OUTPUT);
  pinMode(kPinButton1, INPUT);
  digitalWrite(kPinButton1, HIGH);
  Serial.begin(9600);
}

void loop() {
  if(digitalRead(kPinButton1) == LOW) {
     digitalWrite(kPinLed, HIGH);
  } else{
     digitalWrite(kPinLed, LOW);
  }
}
```

“Analog outputs”



```
const int kPinLed = 13;

void setup() {
  pinMode(kPinLed, OUTPUT);
}

void loop() {
  digitalWrite(kPinLed, HIGH);
  delay(5);
  digitalWrite(kPinLed, LOW);
  delay(5);
}
```

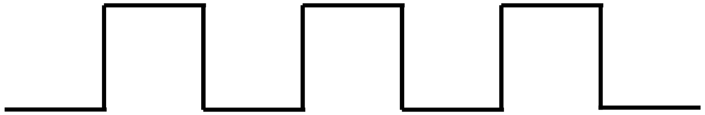
“Analog outputs”



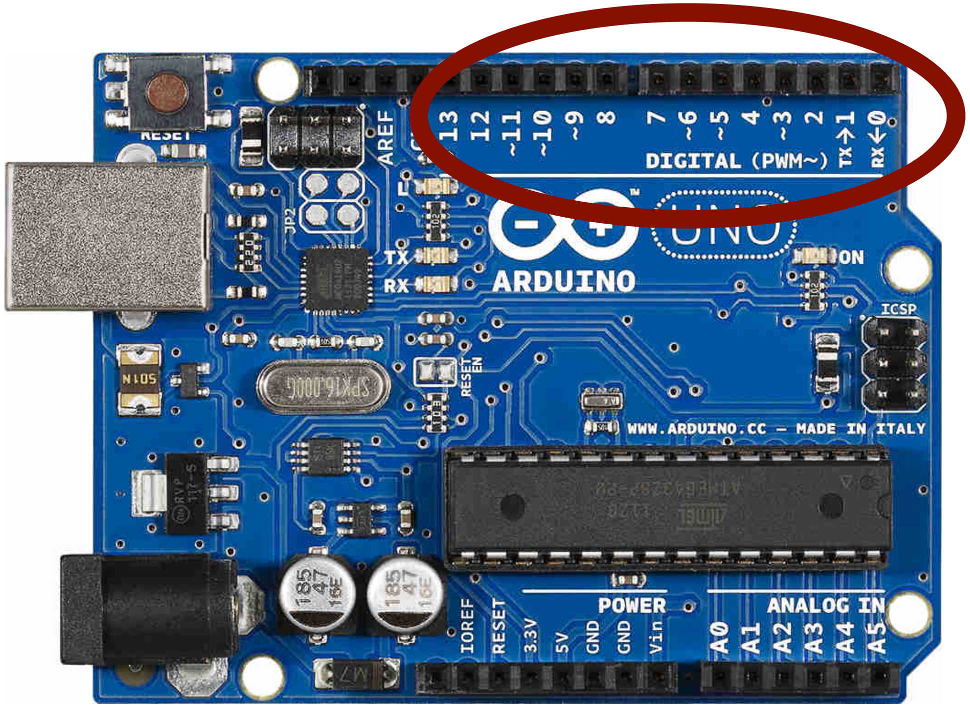
(50% duty cycle)

“Analog outputs”

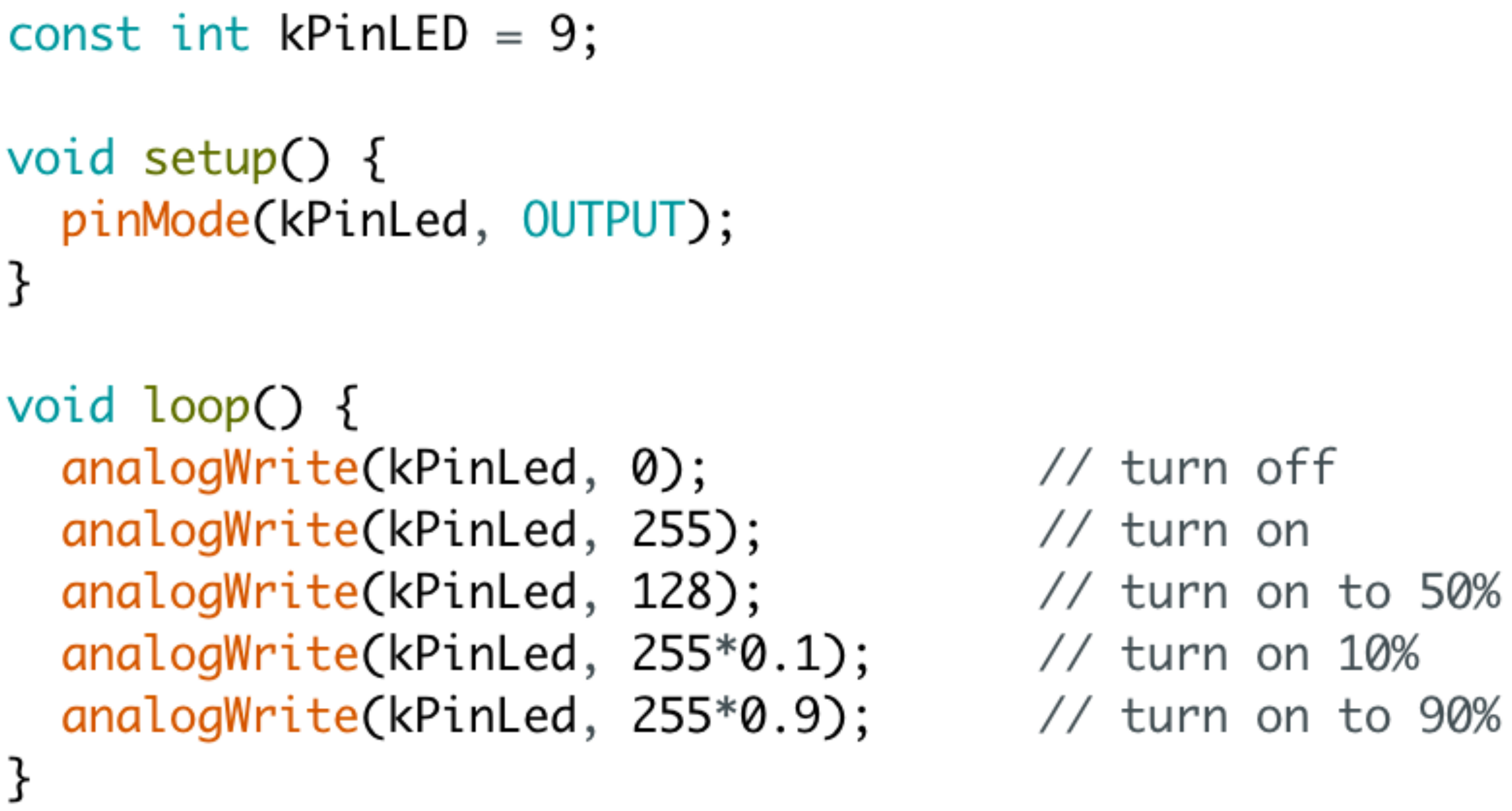
Pulse Width Modulation



Outputs marked with ~ support PWM at 500 Hz

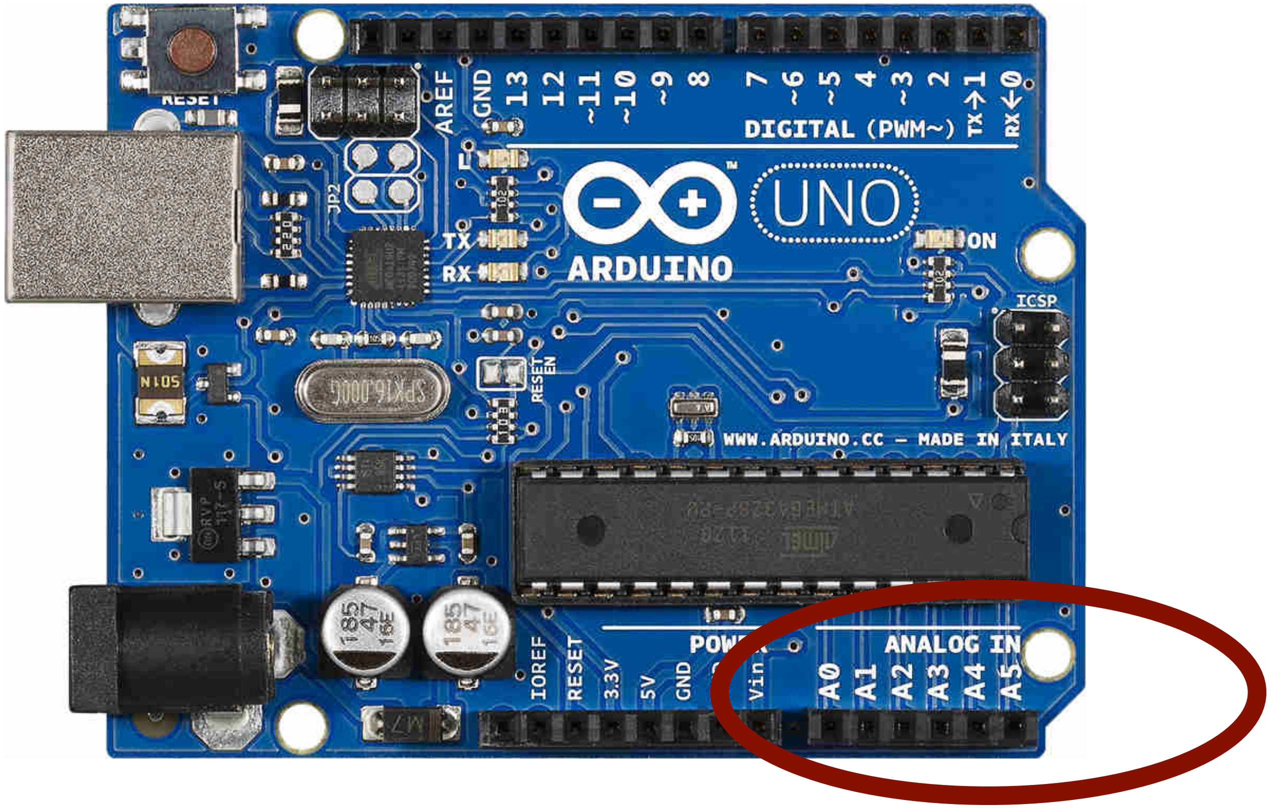


“Analog outputs”

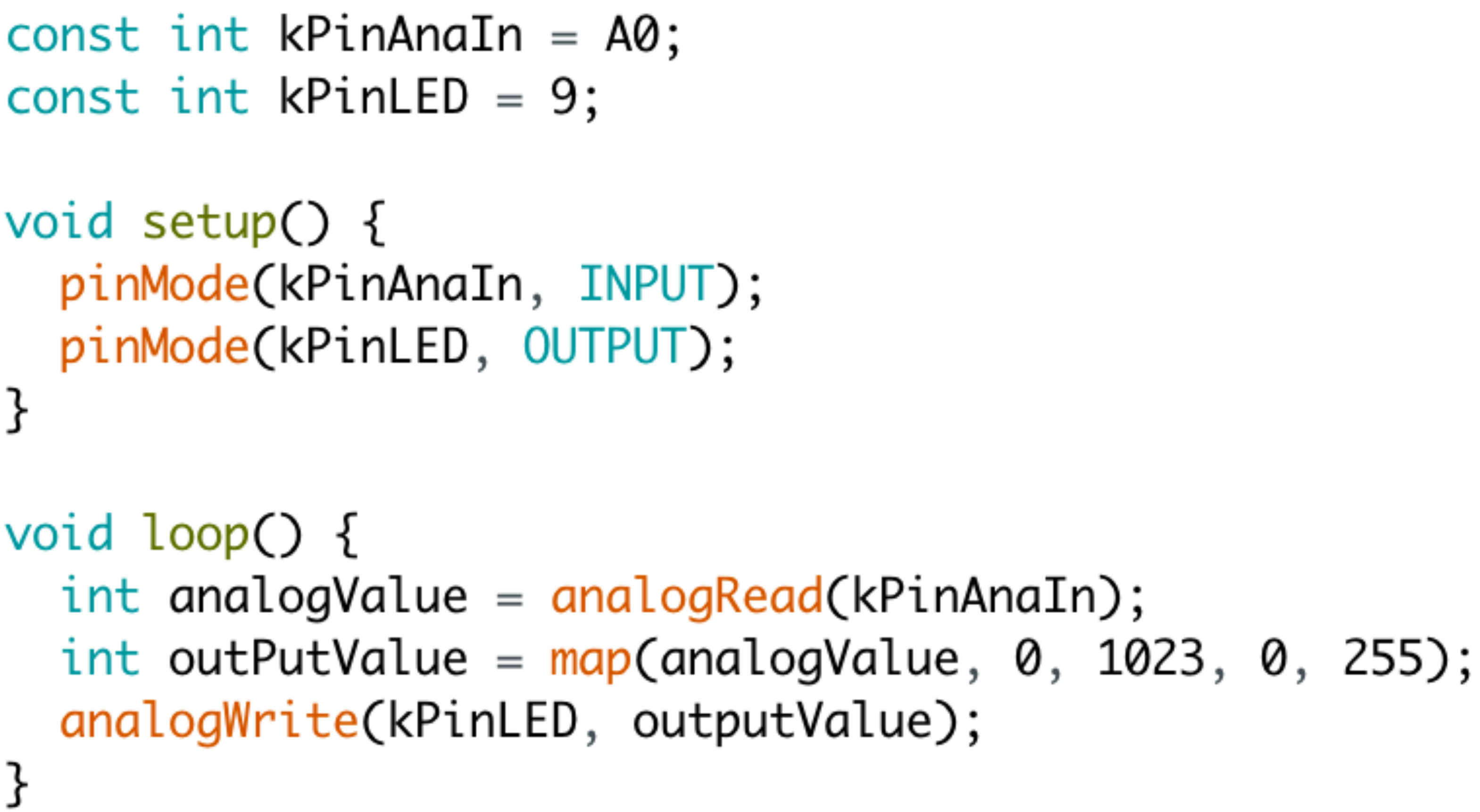


```
const int kPinLED = 9;

void setup() {
  pinMode(kPinLed, OUTPUT);
}

void loop() {
  analogWrite(kPinLed, 0);            // turn off
  analogWrite(kPinLed, 255);          // turn on
  analogWrite(kPinLed, 128);          // turn on to 50%
  analogWrite(kPinLed, 255*0.1);      // turn on 10%
  analogWrite(kPinLed, 255*0.9);      // turn on to 90%
}
```

Analog inputs



Analog inputs



```
const int kPinAnaIn = A0;
const int kPinLED = 9;

void setup() {
  pinMode(kPinAnaIn, INPUT);
  pinMode(kPinLED, OUTPUT);
}

void loop() {
  int analogValue = analogRead(kPinAnaIn);
  int outPutValue = map(analogValue, 0, 1023, 0, 255);
  analogWrite(kPinLED, outputValue);
}
```

Advanced

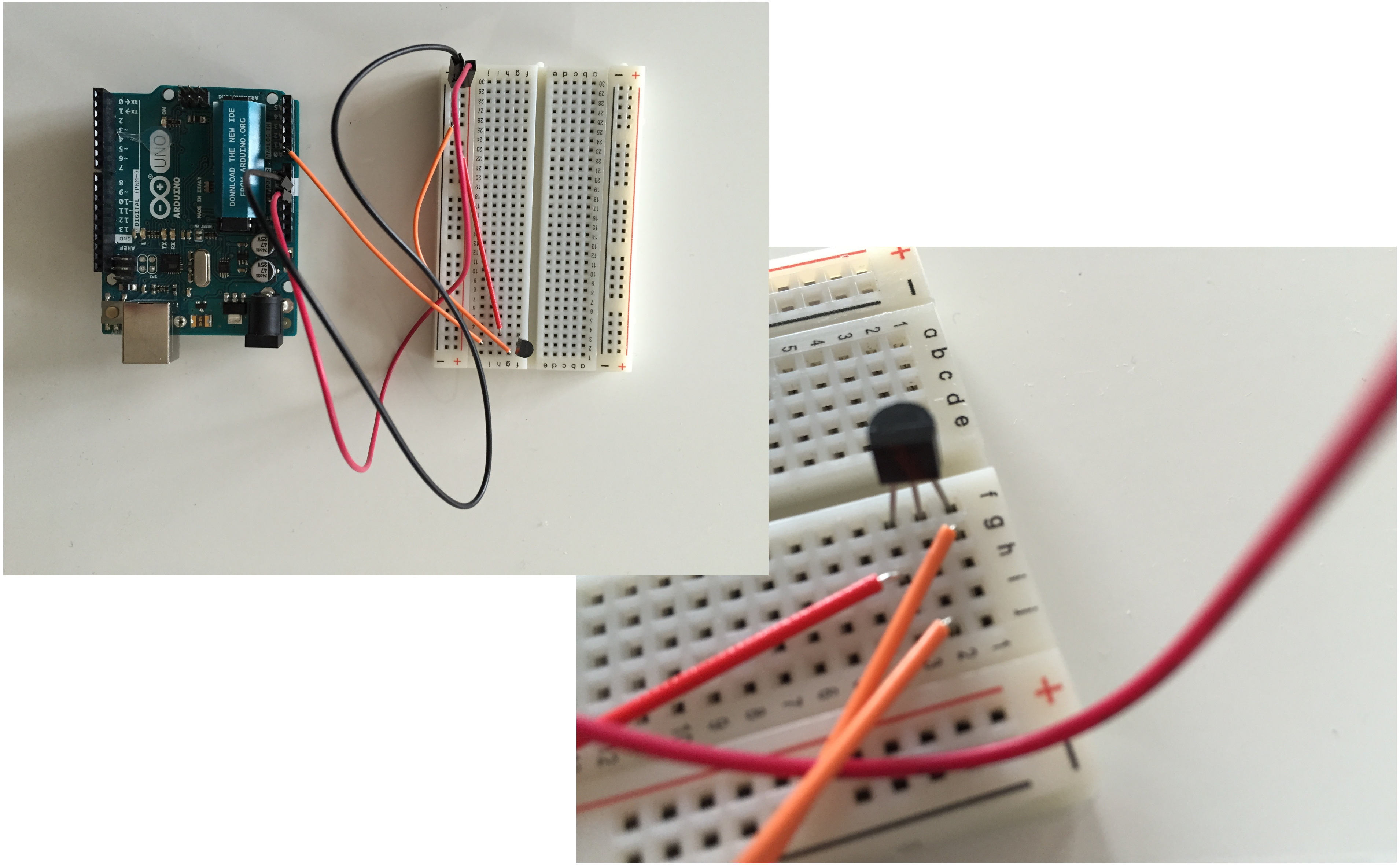
So far we have seen

Digital I/O

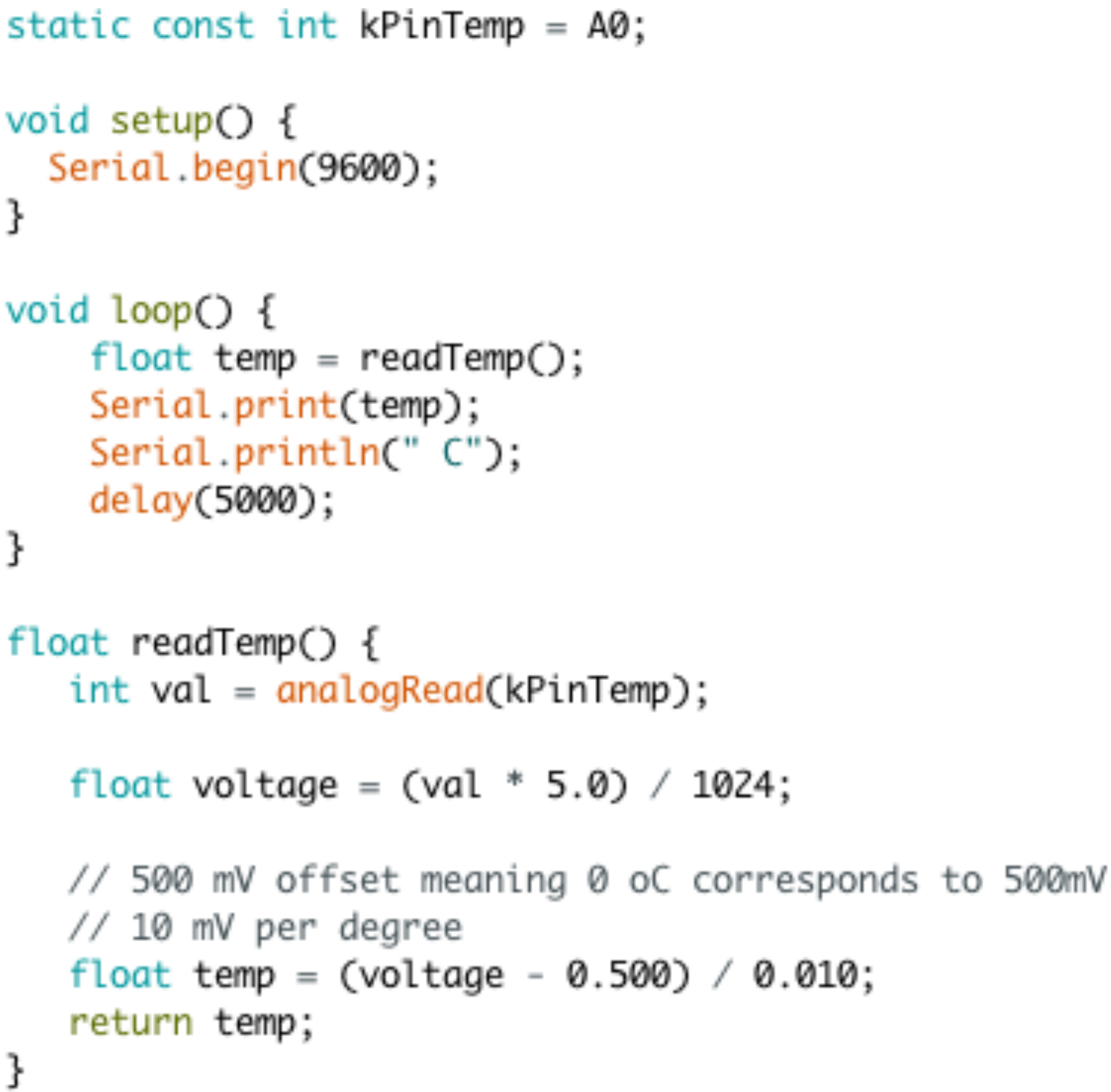
Analog I/O

- we could have done that with wires and soldering too!

Advanced

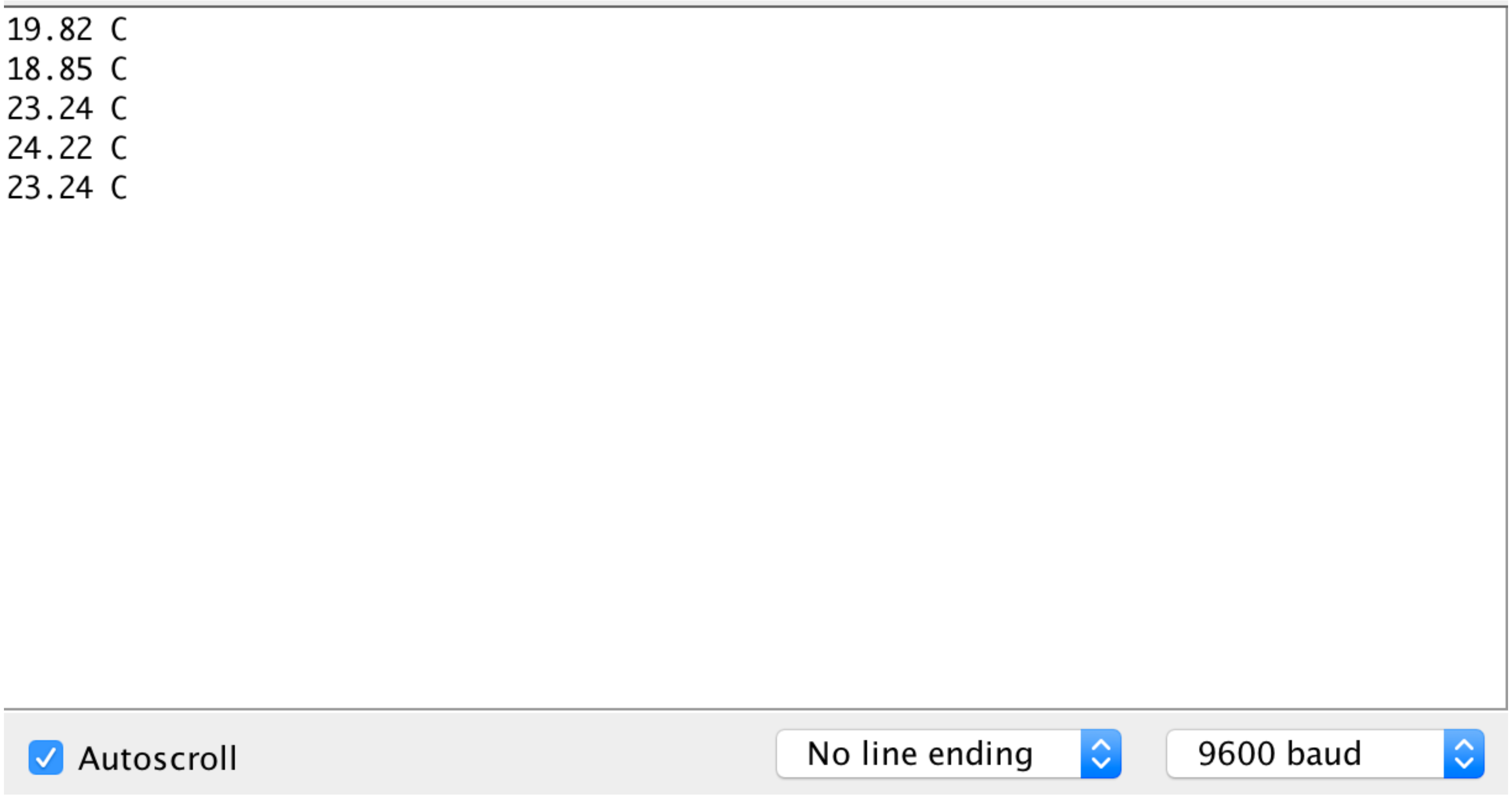

UNO
ARDUINO
DOWNLOAD THE NEW IDE FROM ARDUINO.ORG
MADE IN ITALY
DIGITAL
a b c d e
f g h i j

Advanced



```
static const int kPinTemp = A0;

void setup() {
  Serial.begin(9600);
}

void loop() {
    float temp = readTemp();
    Serial.print(temp);
    Serial.println(" C");
    delay(5000);
}

float readTemp() {
   int val = analogRead(kPinTemp);

   float voltage = (val * 5.0) / 1024;

   // 500 mV offset meaning 0 oC corresponds to 500mV
   // 10 mV per degree
   float temp = (voltage - 0.500) / 0.010;
   return temp;
}
```

Advanced



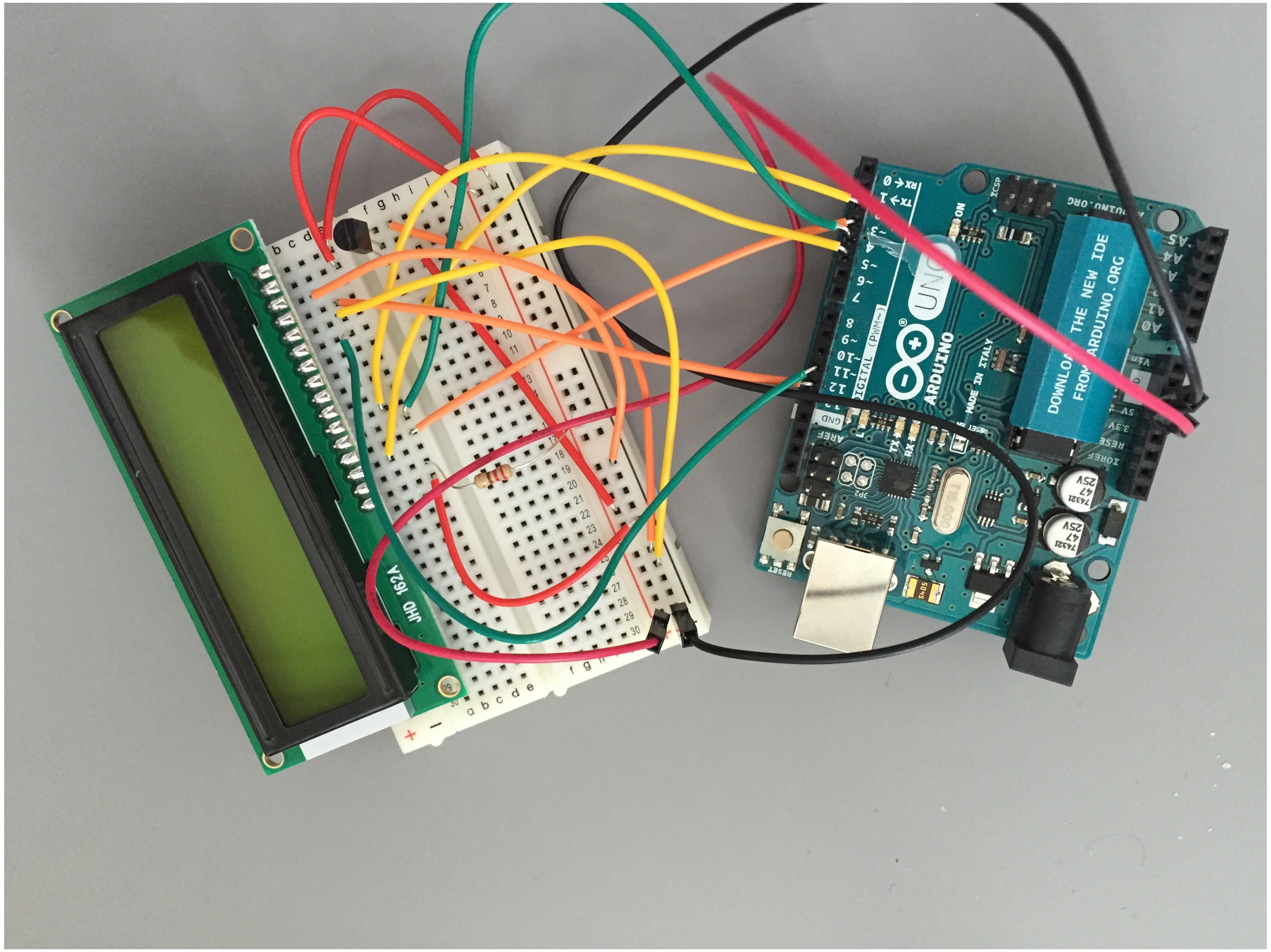
```
19.82 C
18.85 C
23.24 C
24.22 C
23.24 C
```

☑ Autoscroll

No line ending

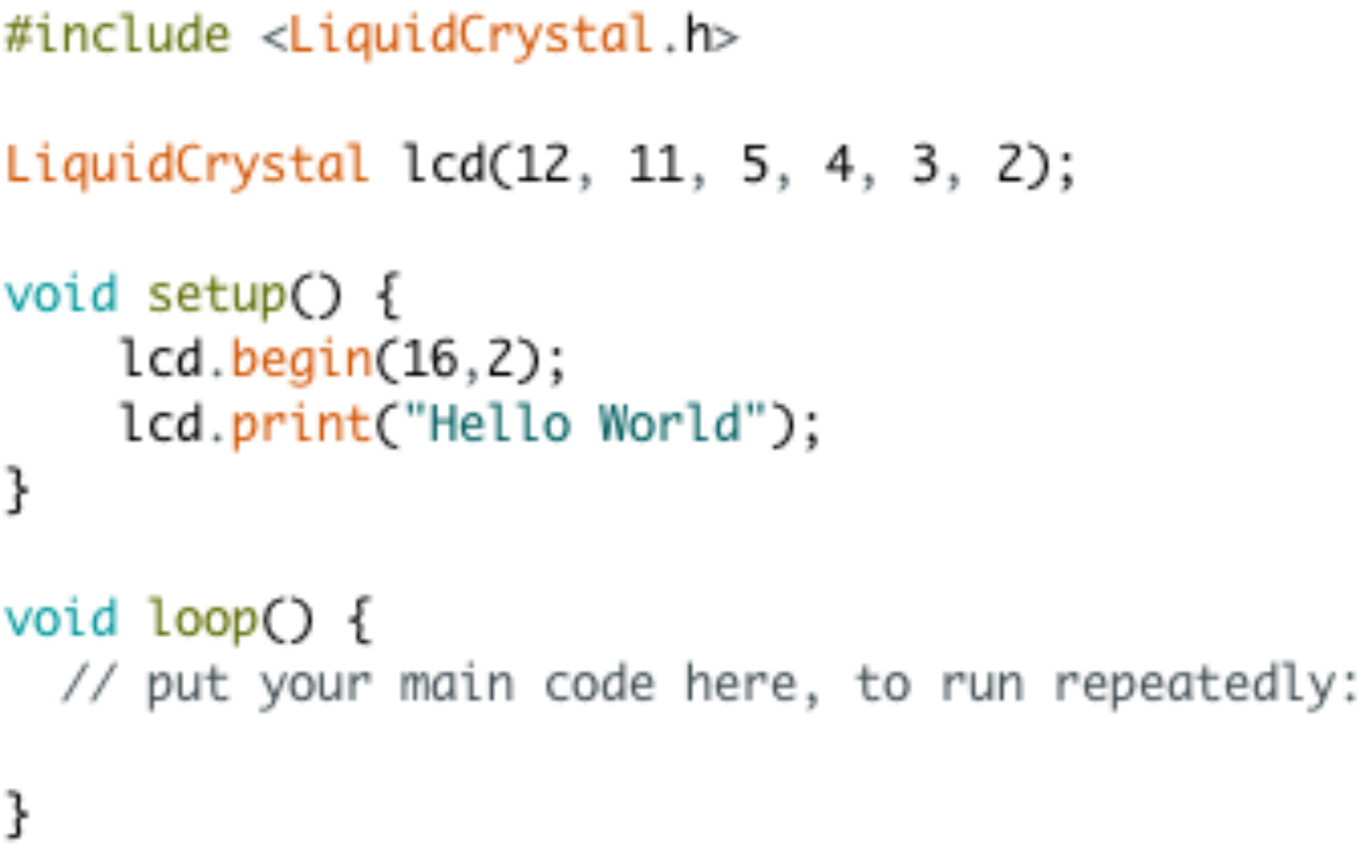
9600 baud

Advanced



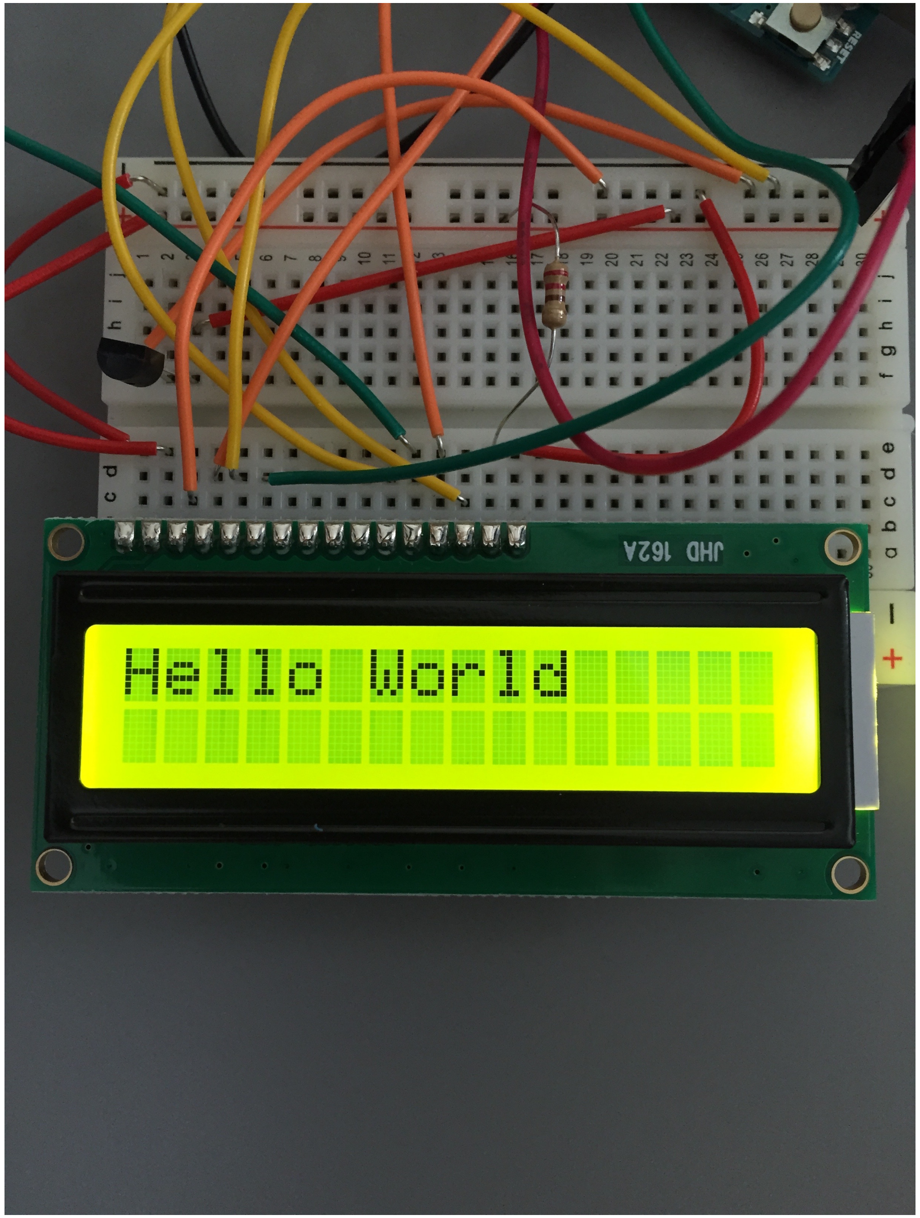
Display: HD44780

Advanced

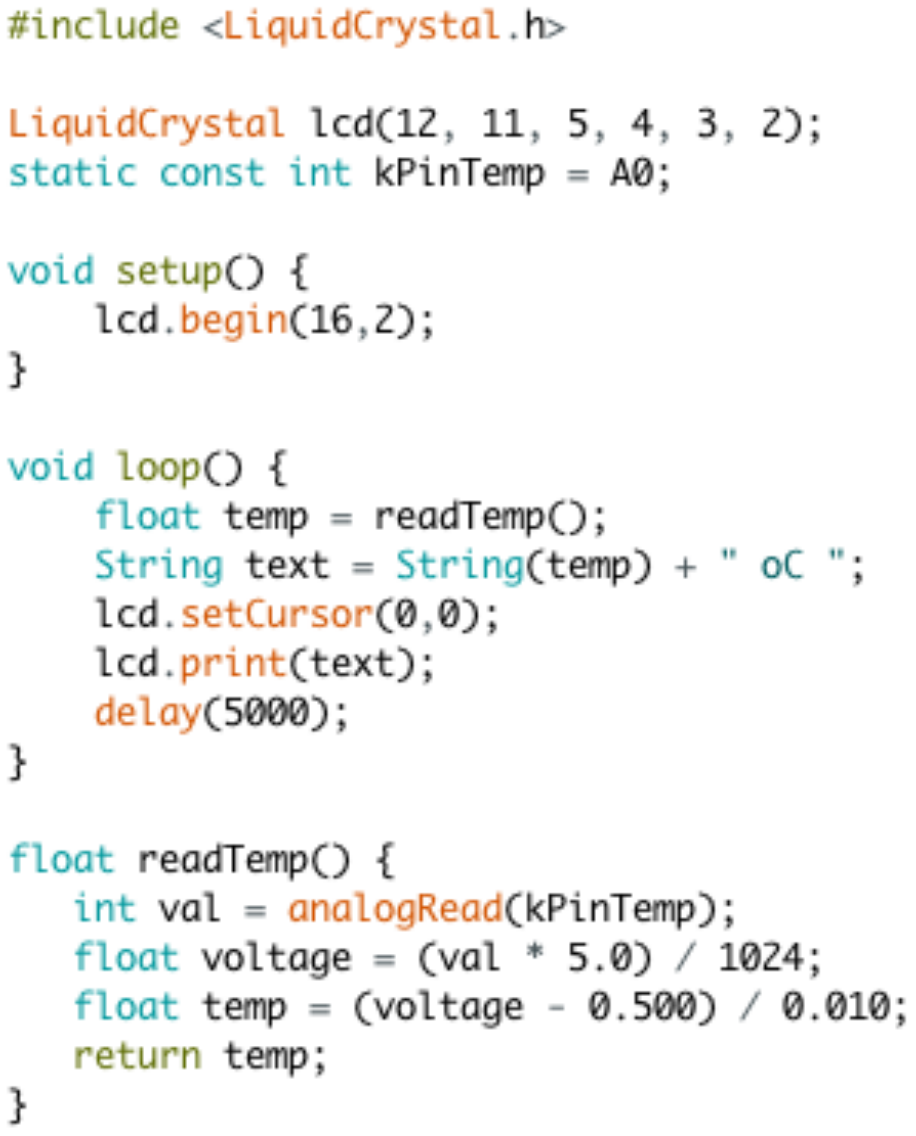


```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
    lcd.begin(16,2);
    lcd.print("Hello World");
}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Advanced

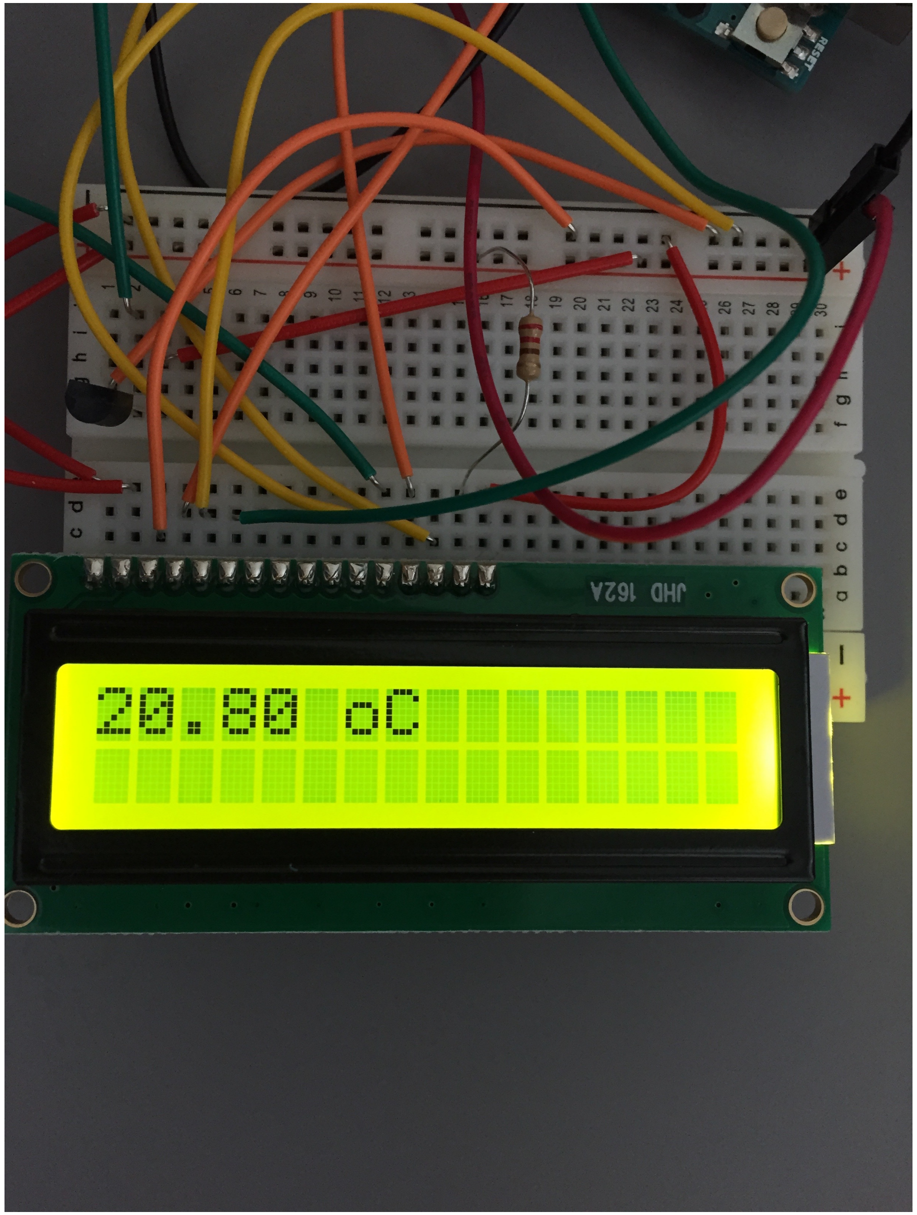


Advanced



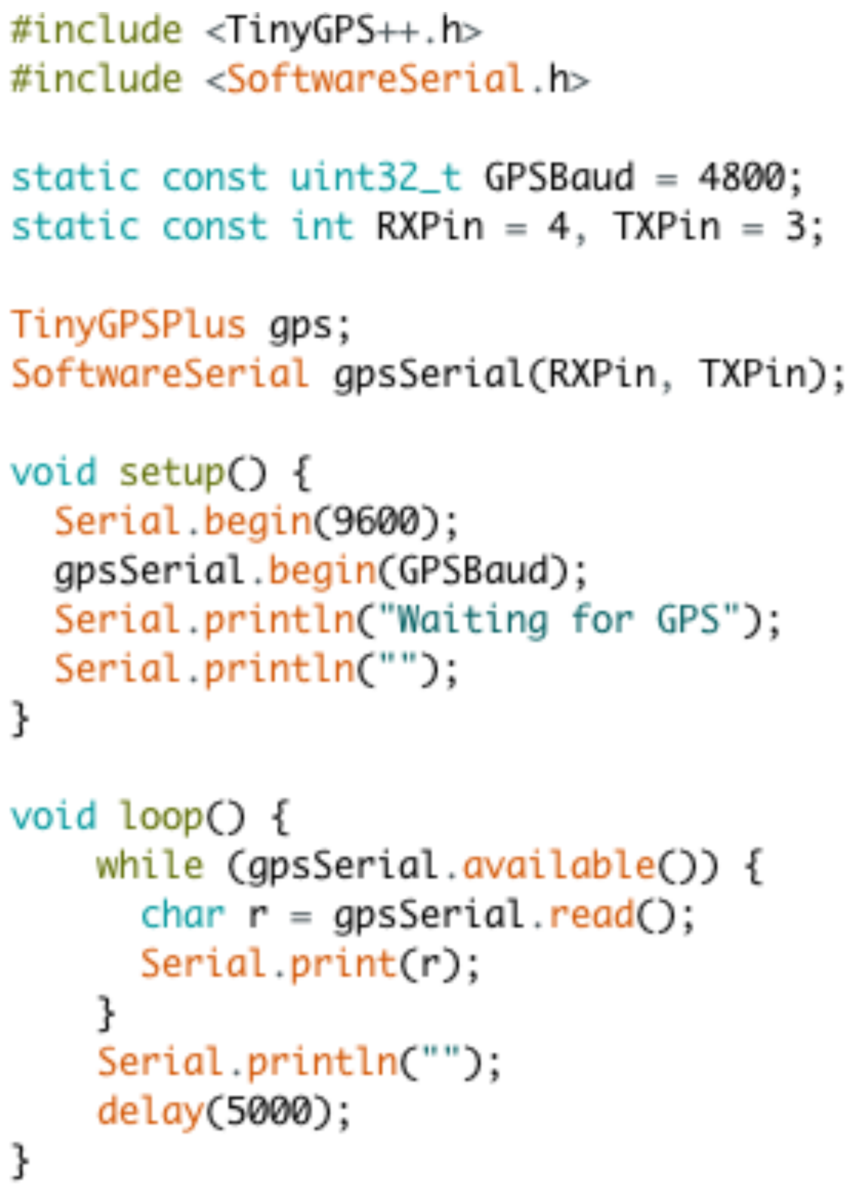
```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
static const int kPinTemp = A0;

void setup() {
    lcd.begin(16,2);
}

void loop() {
    float temp = readTemp();
    String text = String(temp) + " oC ";
    lcd.setCursor(0,0);
    lcd.print(text);
    delay(5000);
}

float readTemp() {
   int val = analogRead(kPinTemp);
   float voltage = (val * 5.0) / 1024;
   float temp = (voltage - 0.500) / 0.010;
   return temp;
}
```

Advanced



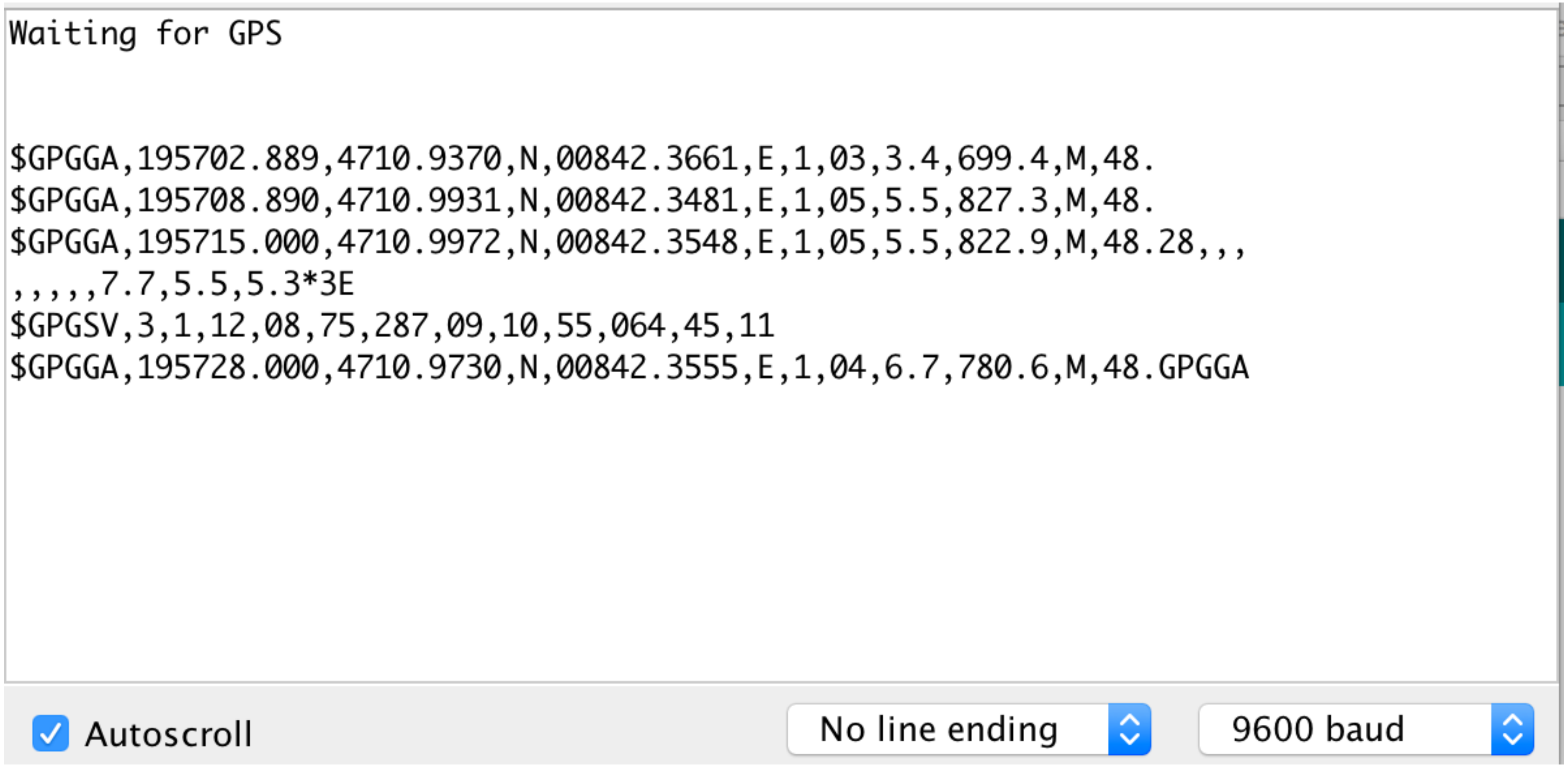
Advanced

Will become your friend!



```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

static const uint32_t GPSBaud = 4800;
static const int RXPin = 4, TXPin = 3;

TinyGPSPlus gps;
SoftwareSerial gpsSerial(RXPin, TXPin);

void setup() {
  Serial.begin(9600);
  gpsSerial.begin(GPSBaud);
  Serial.println("Waiting for GPS");
  Serial.println("");
}

void loop() {
    while (gpsSerial.available()) {
      char r = gpsSerial.read();
      Serial.print(r);
    }
    Serial.println("");
    delay(5000);
}
```

Advanced



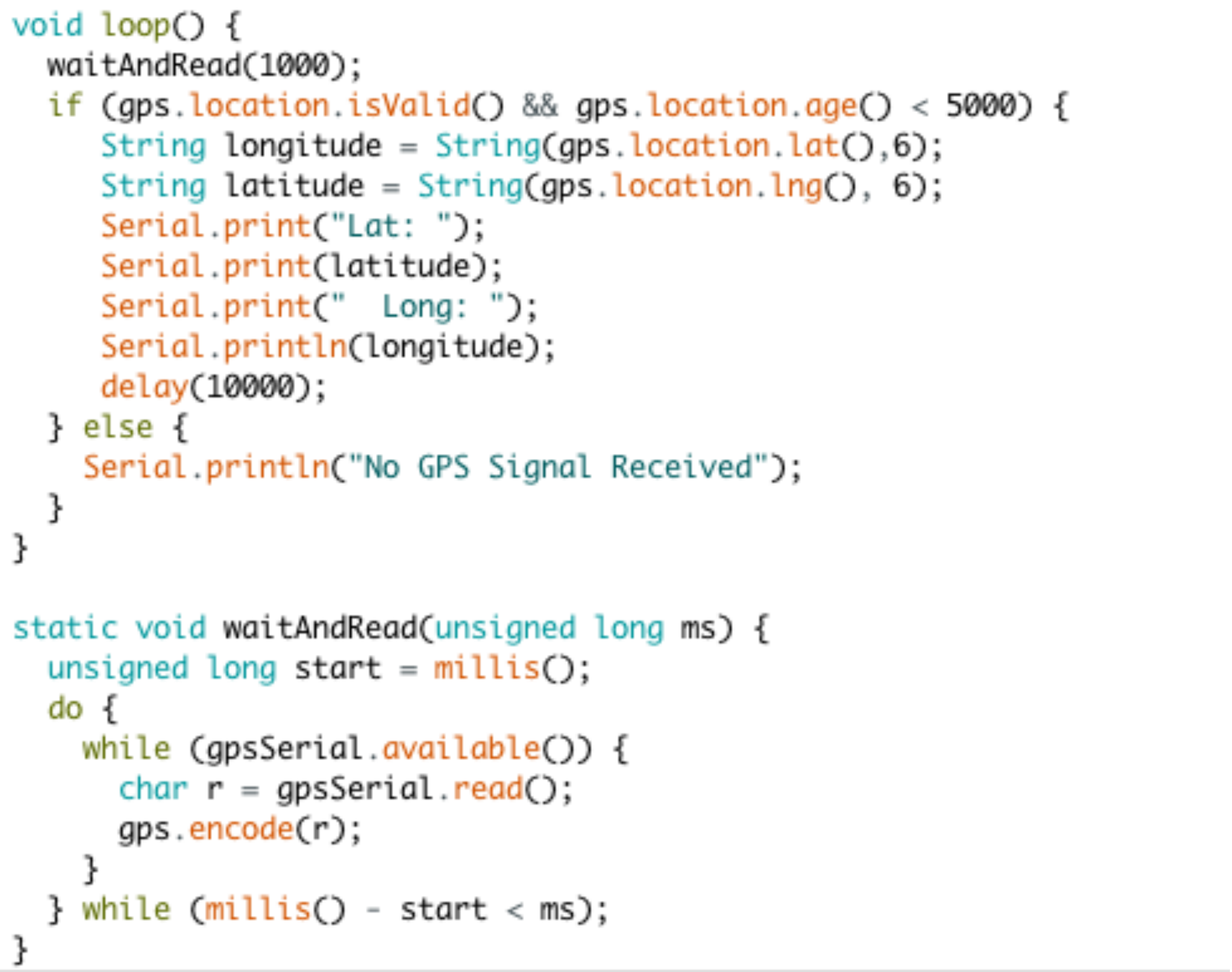
```
Waiting for GPS


$GPGGA,195702.889,4710.9370,N,00842.3661,E,1,03,3.4,699.4,M,48.
$GPGGA,195708.890,4710.9931,N,00842.3481,E,1,05,5.5,827.3,M,48.
$GPGGA,195715.000,4710.9972,N,00842.3548,E,1,05,5.5,822.9,M,48.28,,,
,,,,,7.7,5.5,5.3*3E
$GPGSV,3,1,12,08,75,287,09,10,55,064,45,11
$GPGGA,195728.000,4710.9730,N,00842.3555,E,1,04,6.7,780.6,M,48.GPGGA
```

☑ Autoscroll

No line ending

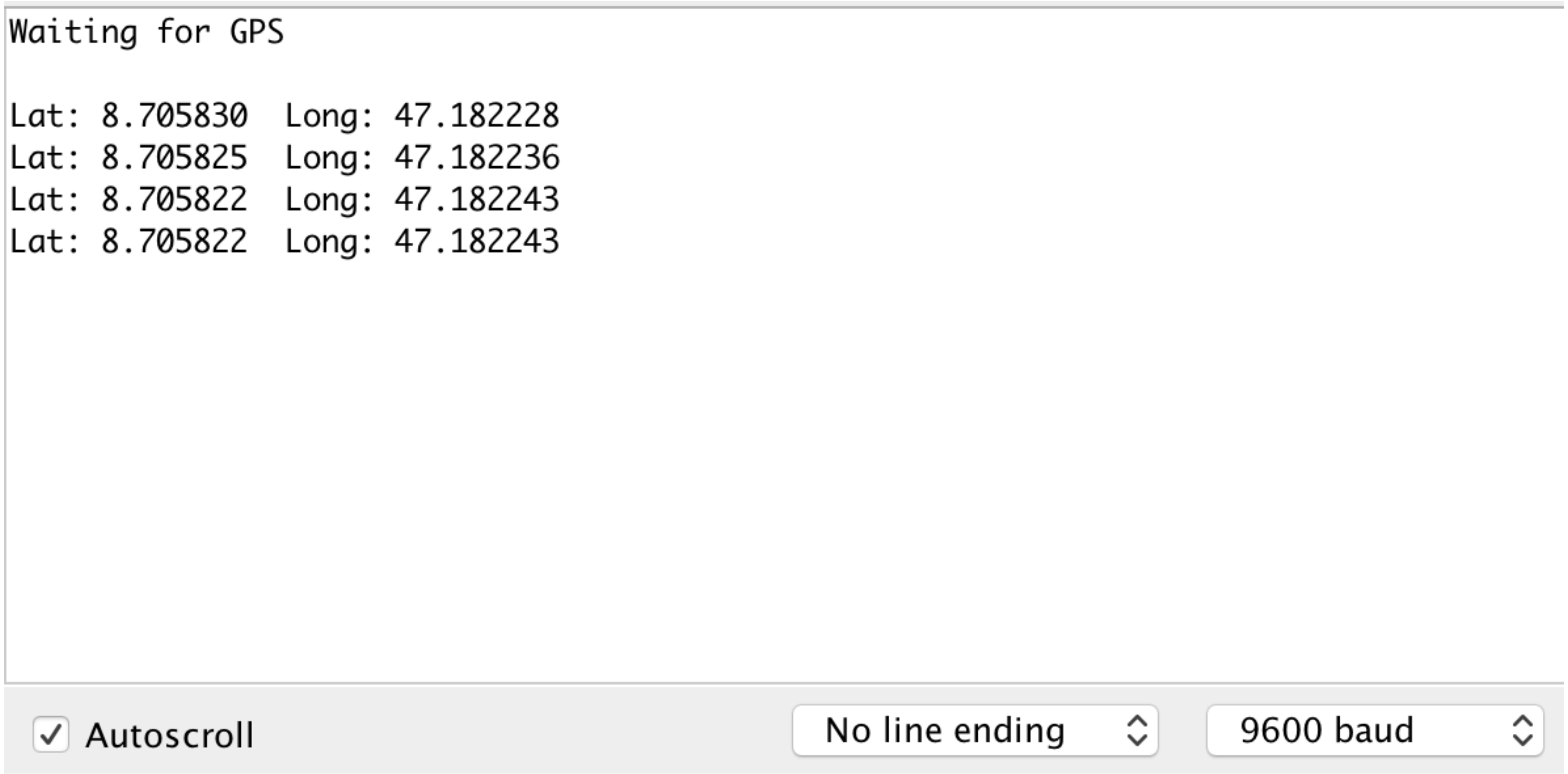
9600 baud

Advanced



```
void loop() {
  waitAndRead(1000);
  if (gps.location.isValid() && gps.location.age() < 5000) {
     String longitude = String(gps.location.lat(),6);
     String latitude = String(gps.location.lng(), 6);
     Serial.print("Lat: ");
     Serial.print(latitude);
     Serial.print("  Long: ");
     Serial.println(longitude);
     delay(10000);
  } else {
    Serial.println("No GPS Signal Received");
  }
}

static void waitAndRead(unsigned long ms) {
  unsigned long start = millis();
  do {
    while (gpsSerial.available()) {
      char r = gpsSerial.read();
      gps.encode(r);
    }
  } while (millis() - start < ms);
}
```

Advanced



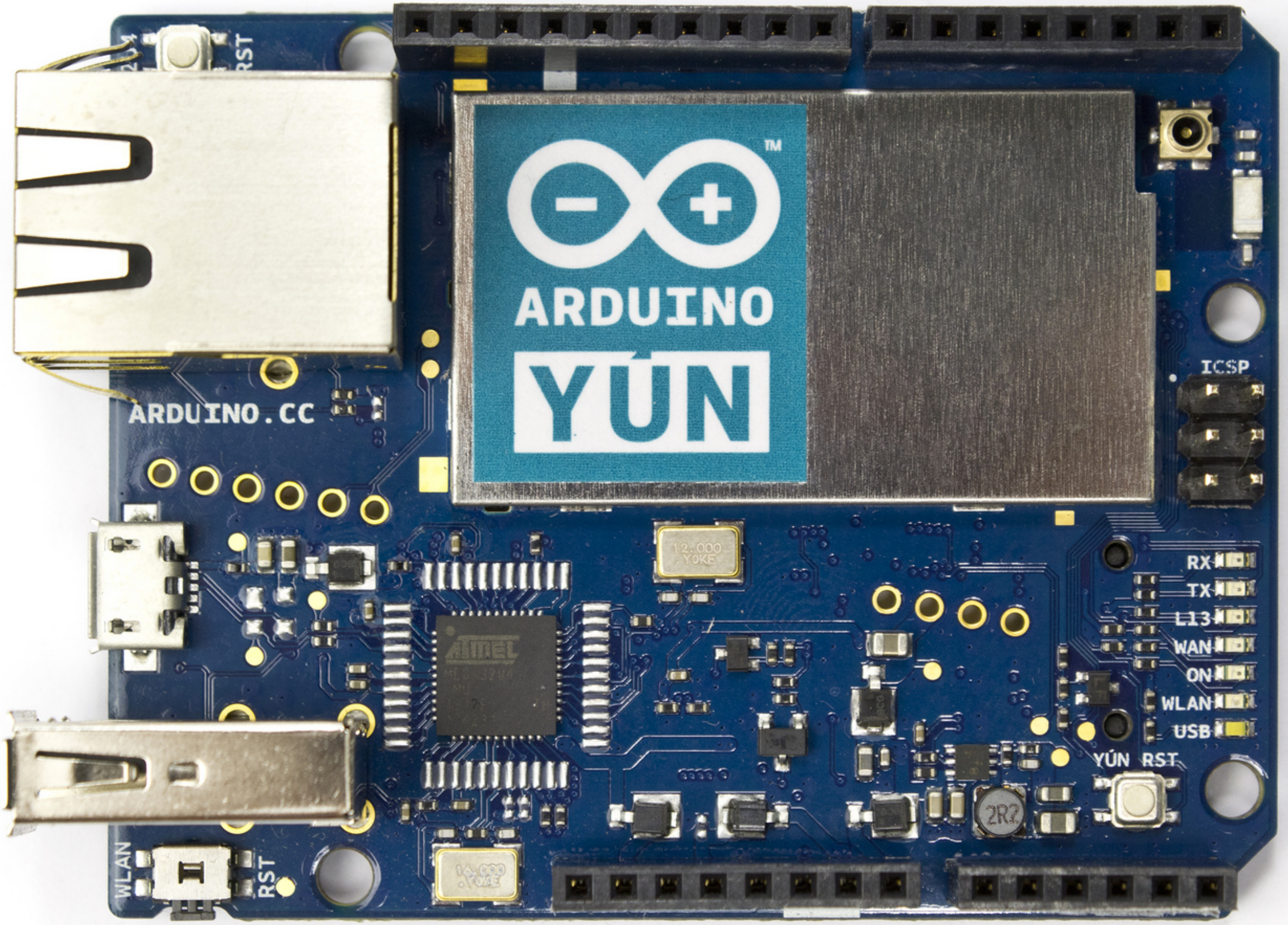
```
Waiting for GPS

Lat: 8.705830  Long: 47.182228
Lat: 8.705825  Long: 47.182236
Lat: 8.705822  Long: 47.182243
Lat: 8.705822  Long: 47.182243
```

☑ Autoscroll

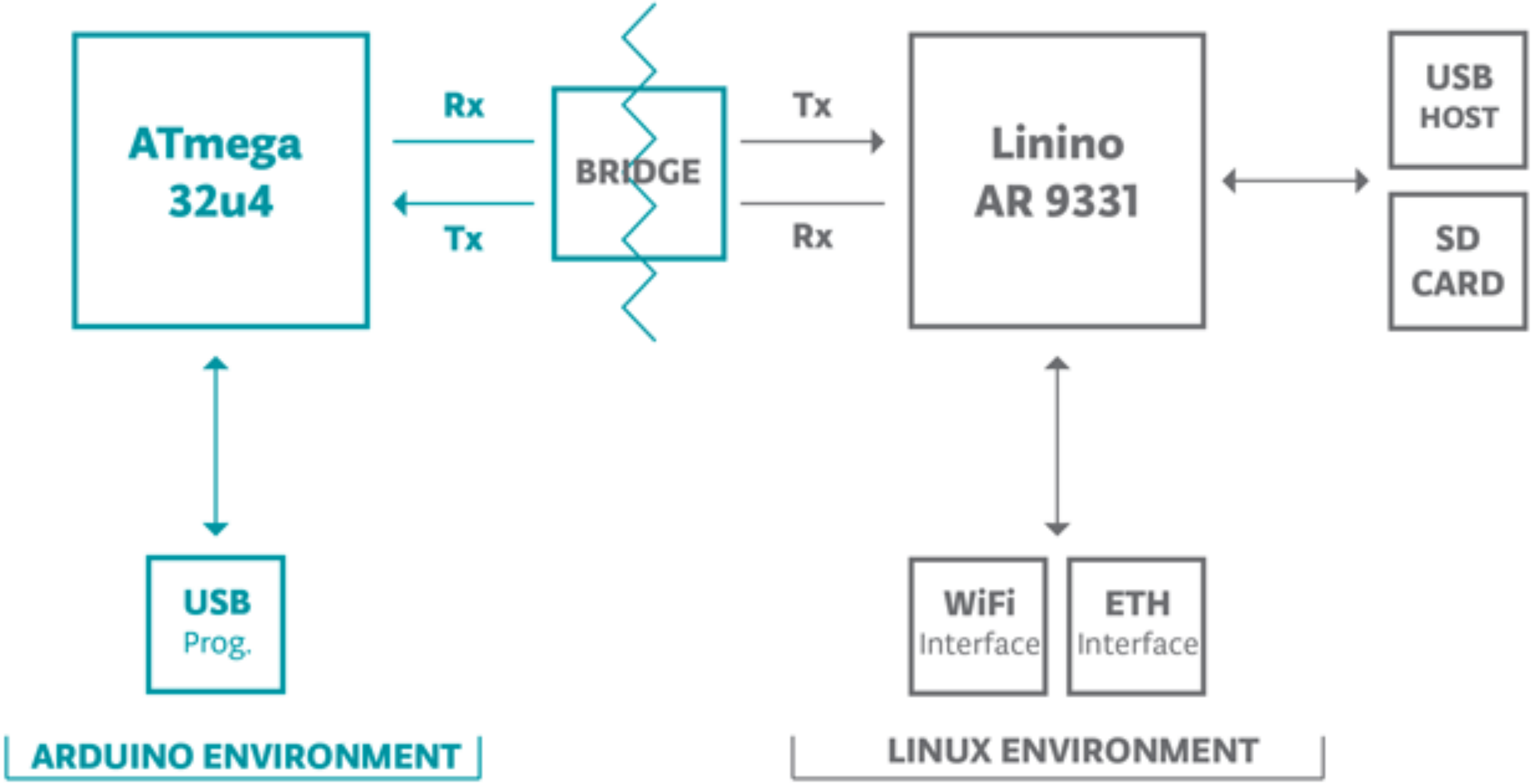
No line ending

9600 baud

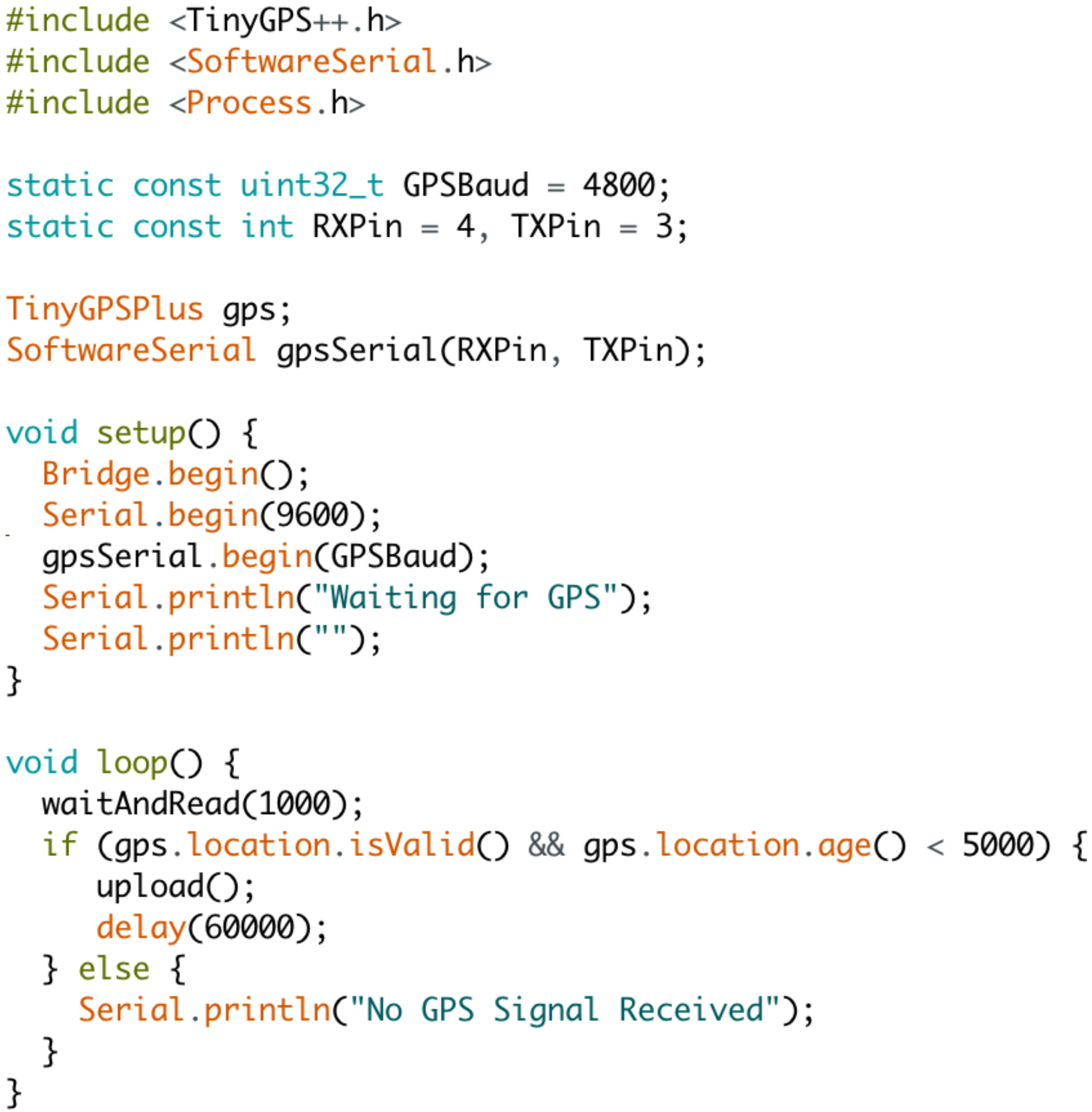
Advanced


ARDUINO
YÚN
ARDUINO.CC
RST
ICSP
RX
TX
L13
WAN
ON
WLAN
USB
YÚN RST
WLAN
RST

Advanced

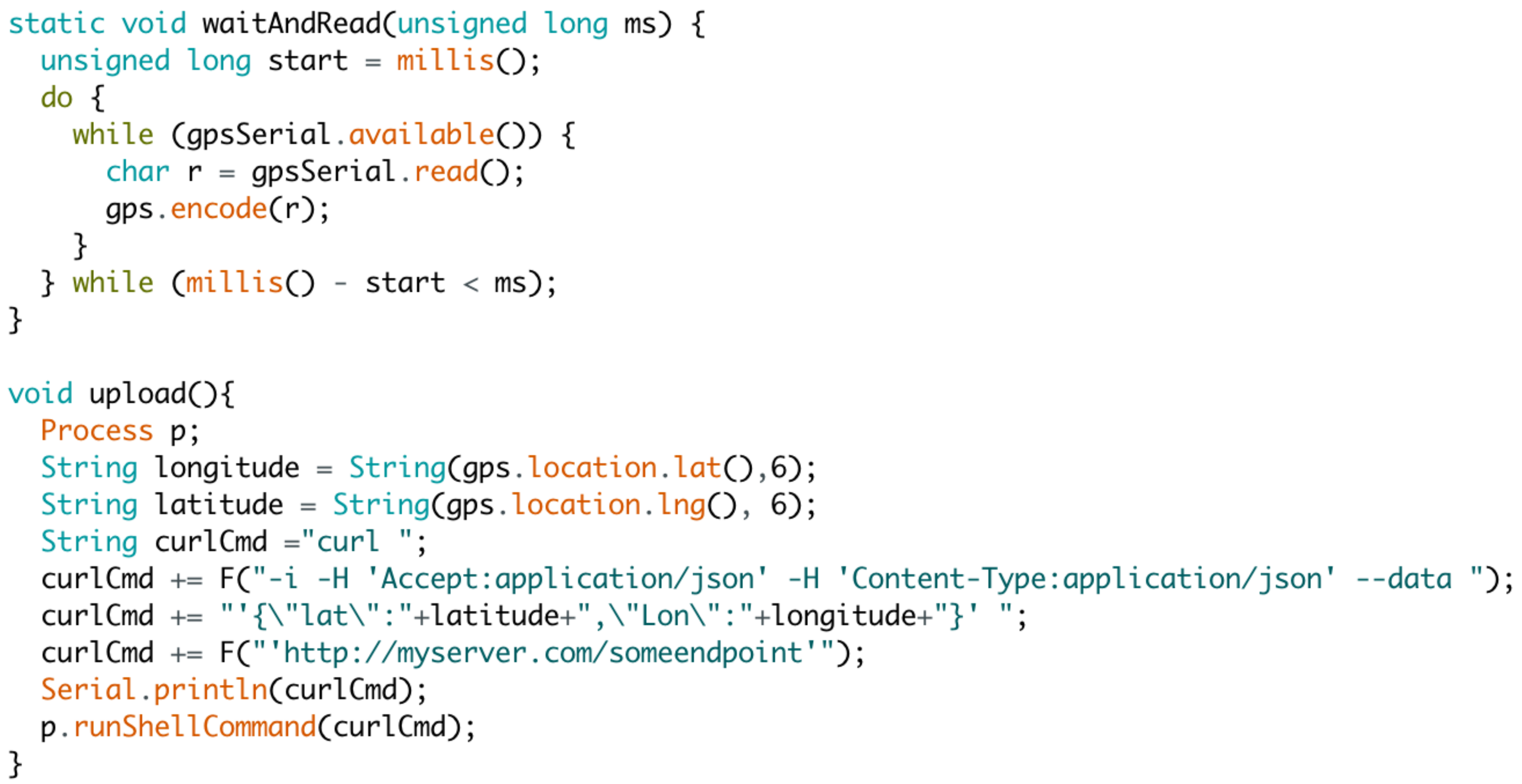

ATmega 32u4
Rx
Tx
BRIDGE
Tx
Rx
Linino AR 9331
USB HOST
SD CARD
USB Prog.
WiFi Interface
ETH Interface
ARDUINO ENVIRONMENT
LINUX ENVIRONMENT

Advanced



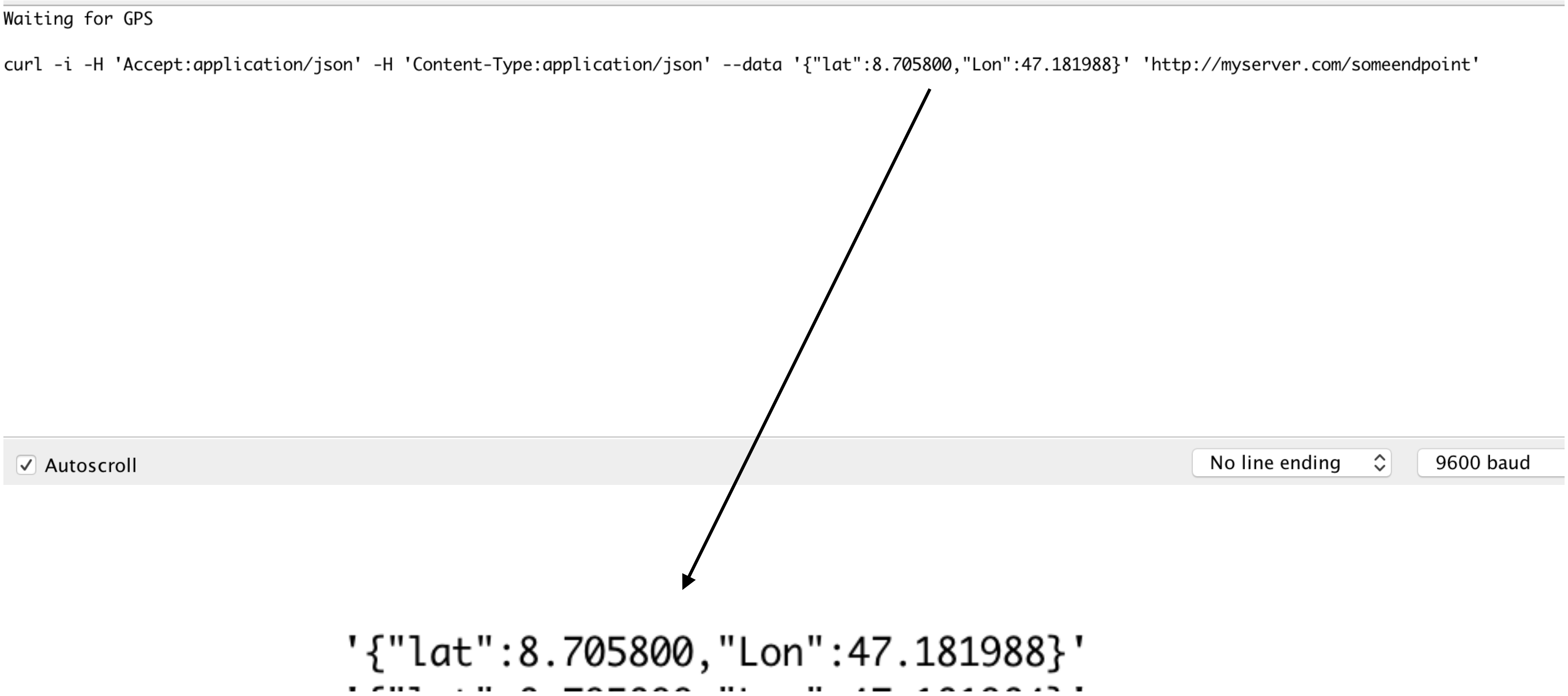
```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <Process.h>

static const uint32_t GPSBaud = 4800;
static const int RXPin = 4, TXPin = 3;

TinyGPSPlus gps;
SoftwareSerial gpsSerial(RXPin, TXPin);

void setup() {
  Bridge.begin();
  Serial.begin(9600);
  gpsSerial.begin(GPSBaud);
  Serial.println("Waiting for GPS");
  Serial.println("");
}

void loop() {
  waitAndRead(1000);
  if (gps.location.isValid() && gps.location.age() < 5000) {
     upload();
     delay(60000);
  } else {
    Serial.println("No GPS Signal Received");
  }
}
```

Advanced



```
static void waitAndRead(unsigned long ms) {
  unsigned long start = millis();
  do {
    while (gpsSerial.available()) {
      char r = gpsSerial.read();
      gps.encode(r);
    }
  } while (millis() - start < ms);
}

void upload(){
  Process p;
  String longitude = String(gps.location.lat(),6);
  String latitude = String(gps.location.lng(), 6);
  String curlCmd ="curl ";
  curlCmd += F("-i -H 'Accept:application/json' -H 'Content-Type:application/json' --data ");
  curlCmd += "'{\"lat\":"+latitude+",\"Lon\":"+longitude+"}' ";
  curlCmd += F("'http://myserver.com/someendpoint'");
  Serial.println(curlCmd);
  p.runShellCommand(curlCmd);
}
```

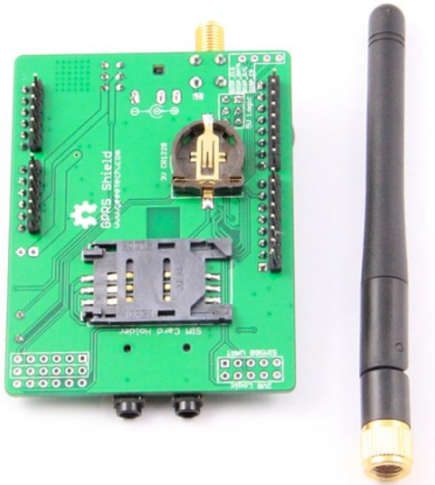
Advanced



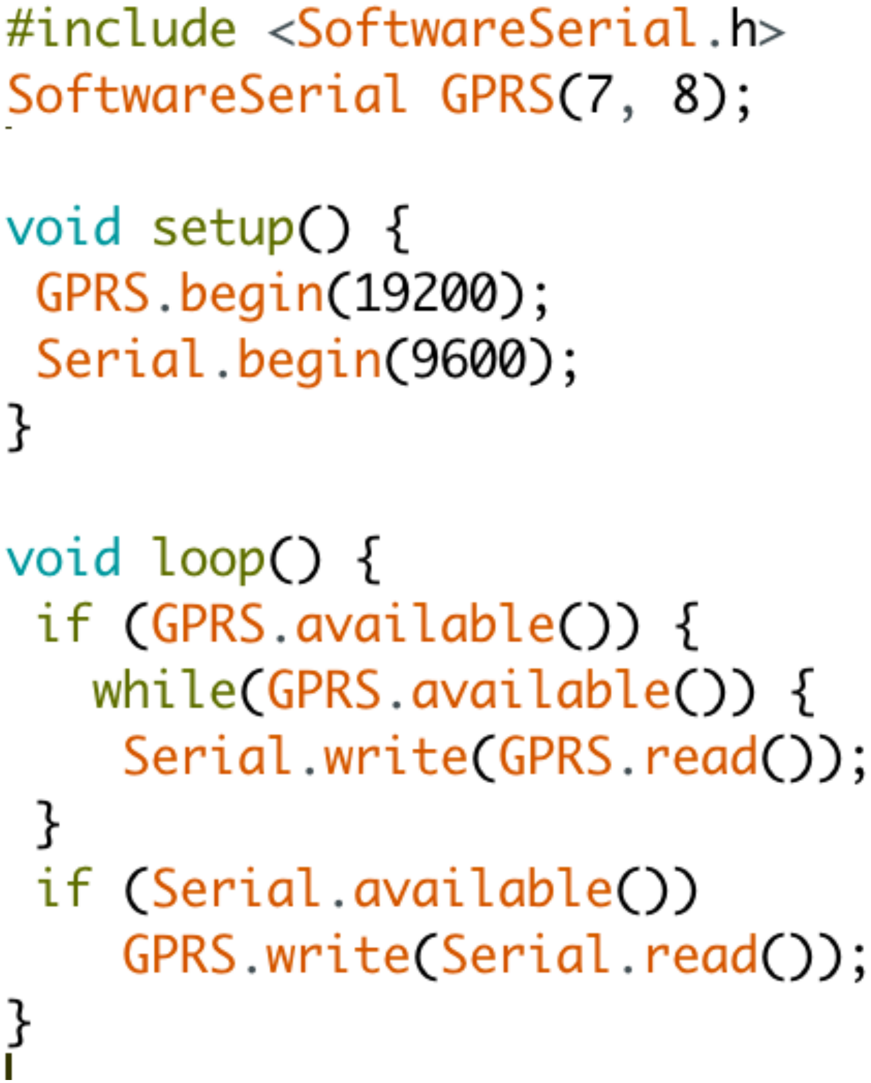
```
Waiting for GPS

curl -i -H 'Accept:application/json' -H 'Content-Type:application/json' --data '{"lat":8.705800,"Lon":47.181988}' 'http://myserver.com/someendpoint'
```

```
'{"lat":8.705800,"Lon":47.181988}'
```

Advanced



Advanced



```
#include <SoftwareSerial.h>
SoftwareSerial GPRS(7, 8);

void setup() {
 GPRS.begin(19200);
 Serial.begin(9600);
}

void loop() {
 if (GPRS.available()) {
   while(GPRS.available()) {
    Serial.write(GPRS.read());
 }
 if (Serial.available())
    GPRS.write(Serial.read());
}
```

Advanced

Much better examples here:

http://www.geeetech.com/wiki/index.php/Arduino_GPRS_Shield#A_Simple_Source_Code_Examples

(Simple dial command: ATD+ 12345678)

HTTP requests:
https://www.arduino.cc/en/Guide/ArduinoGSMShield

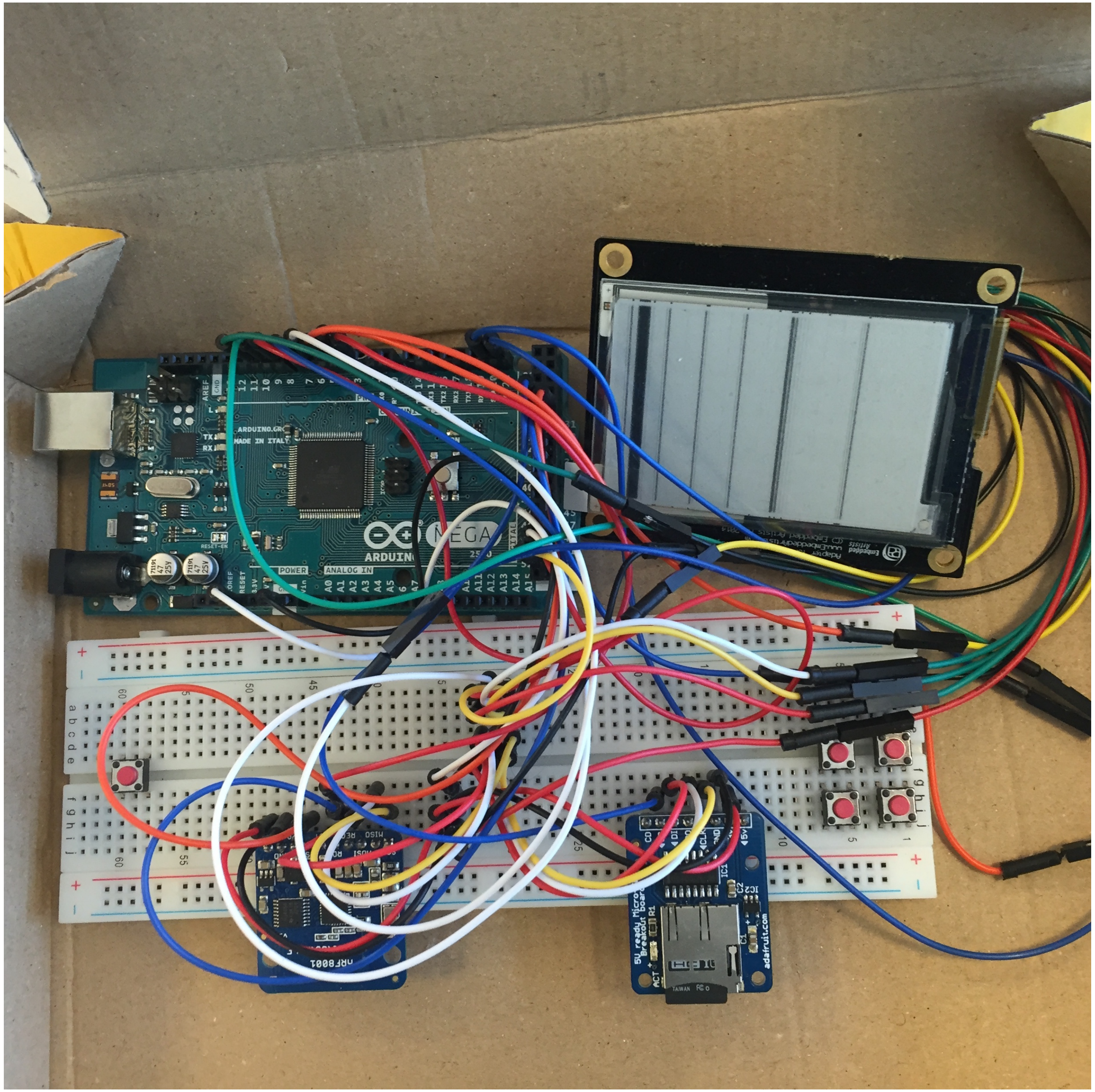
Summing Up

We have seen

Input, Output
Digital, Analog
Serial I/O
Display
Network communication
GPS (which is serial I/O)

Everything is a combination of the above

Summing Up


ARDUINO
MEGA
nRF8001
5V ready Micro-SD Breakout board
adafruit.com