VISHNU
UNIVERSAL LEARNING

ENGINEERING
SCIENCES

Programming "Arduino"

Harald **Bluetooth** (910 -985)

unifying the various Danish tribes into one Danish kingdom around 970

The name Bluetooth wasn’t originally necessarily meant to be the final name of the wireless standard. When they first named it thus, it was just a code name for the technology. It ultimately ended up sticking though and became the official name of the standard.

Bluetooth

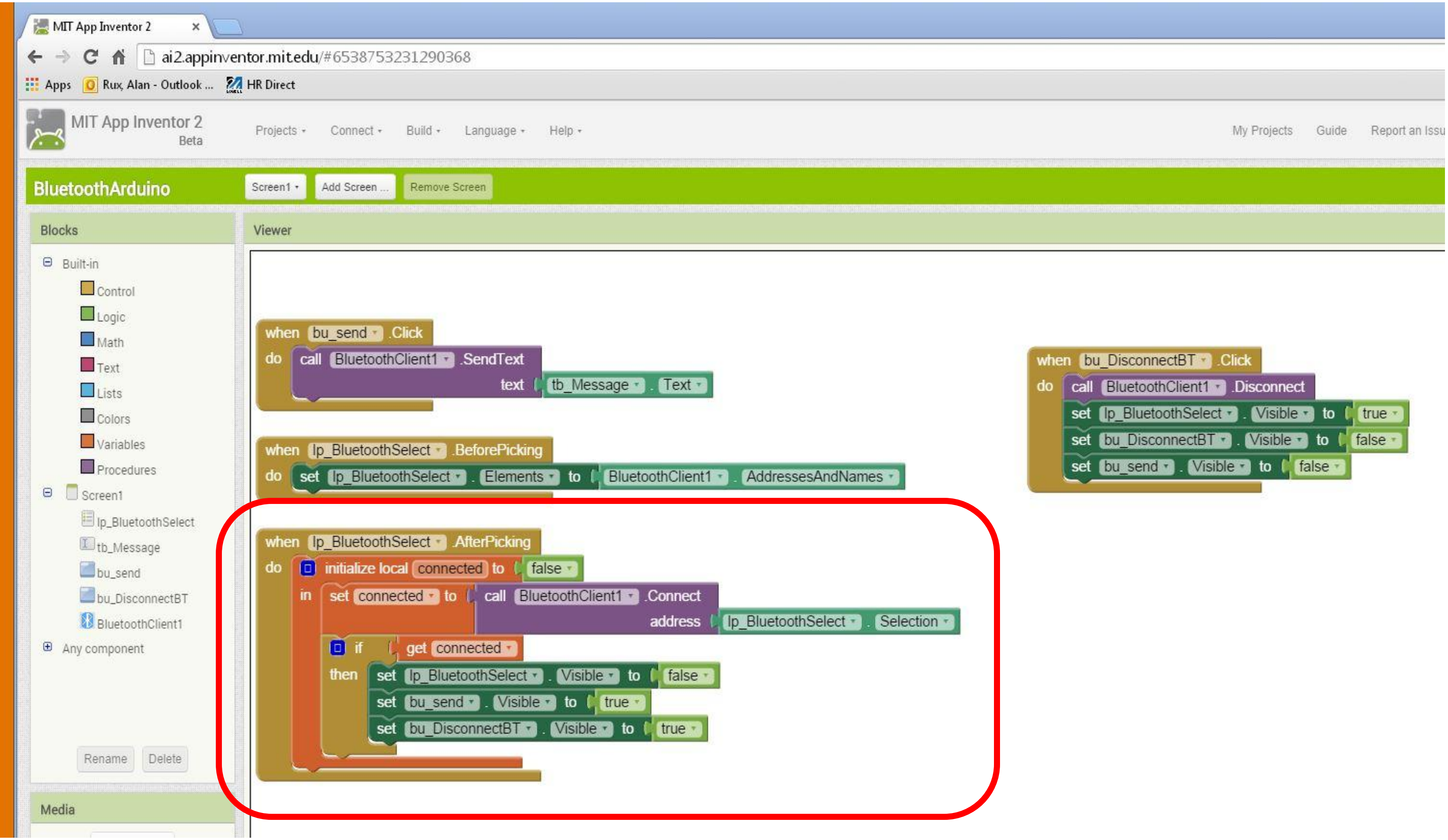
- the wireless Bluetooth standard was developed to be ultra low power and short range
- a maximum range of around 30 feet.
- using short-wavelength in the ISM band
- Nearly 95% of all mobile phones have Bluetooth capabilities.
- The Bluetooth logo is a merging the (Hagall) (ᚼ) and (Bjarkan) (ᛒ), Harald's initials.

Bluetooth

- Bluetooth operates in the range of 2400–2483.5 MHz This is in the globally unlicensed **I**ndustrial, **S**cientific and **M**edical (ISM) 2.4 GHz short-range radio frequency band. Bluetooth uses a radio technology called **frequency-hopping spread spectrum**. The transmitted data are divided into packets and each packet is transmitted on one of the 79 designated Bluetooth channels. Each channel has a bandwidth of 1 MHz. Bluetooth 4.0 uses 2 MHz spacing which allows for 40 channels. The first channel starts at 2402 MHz and continues up to 2480 MHz in 1 MHz steps. It usually performs 1600 hops per second, with **Adaptive Frequency-Hopping** (AFH) enabled.

Google/MIT **App Inventor 2**,

Bluetooth code connecting tablet to Anduino UNO

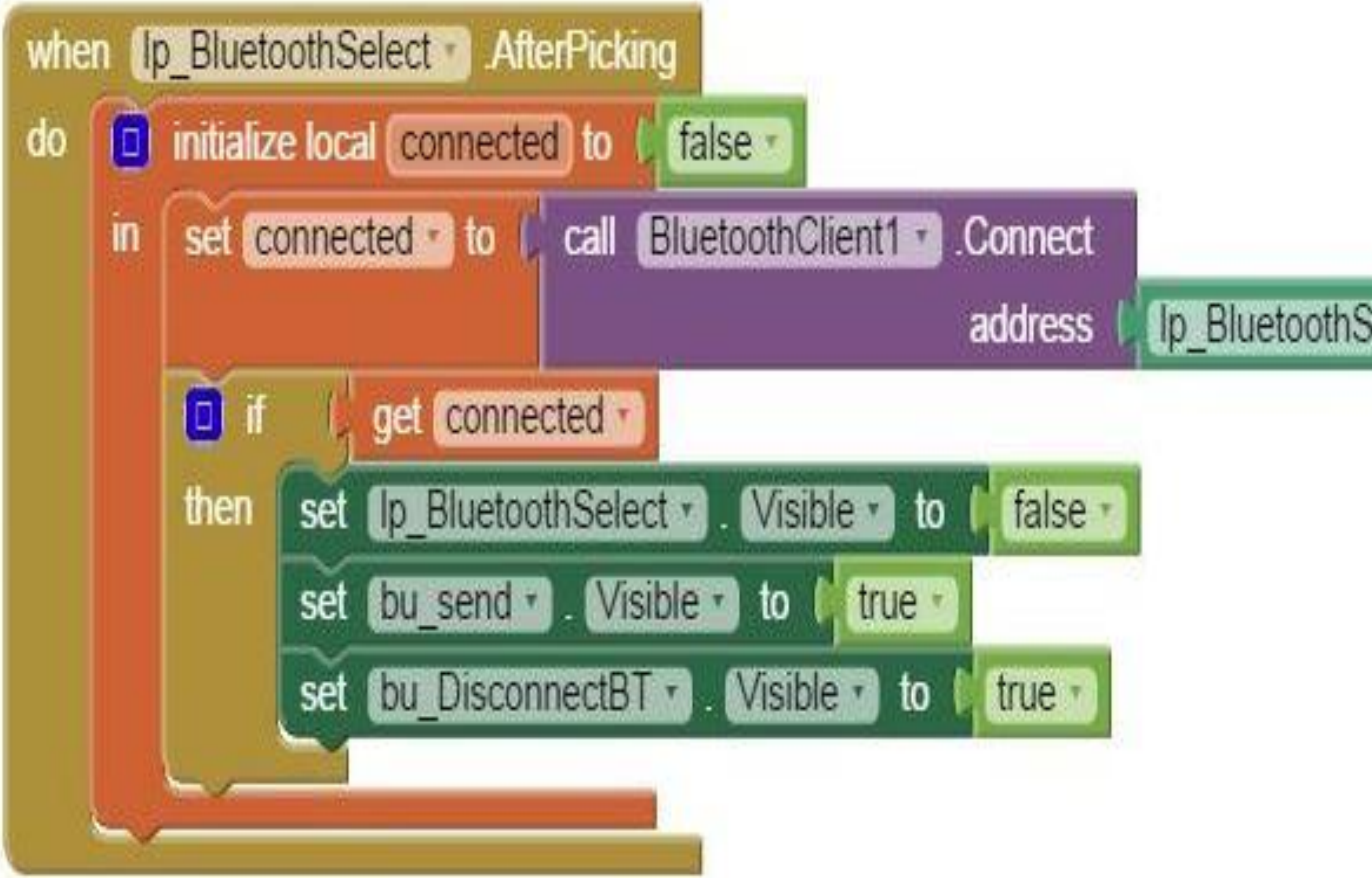


```
when bu_send .Click
do  call BluetoothClient1 .SendText
                          text  tb_Message . Text

when lp_BluetoothSelect .BeforePicking
do  set lp_BluetoothSelect . Elements to  BluetoothClient1 . AddressesAndNames

when lp_BluetoothSelect .AfterPicking
do  initialize local connected to  false
    in  set connected to  call BluetoothClient1 .Connect
                               address  lp_BluetoothSelect . Selection
        if  get connected
        then  set lp_BluetoothSelect . Visible to  false
              set bu_send . Visible to  true
              set bu_DisconnectBT . Visible to  true

when bu_DisconnectBT .Click
do  call BluetoothClient1 .Disconnect
    set lp_BluetoothSelect . Visible to  true
    set bu_DisconnectBT . Visible to  false
    set bu_send . Visible to  false
```



```
when lp_BluetoothSelect .AfterPicking
do  initialize local connected to false
    in  set connected to call BluetoothClient1 .Connect
                                          address lp_BluetoothS
        if   get connected
        then set lp_BluetoothSelect . Visible to false
             set bu_send . Visible to true
             set bu_DisconnectBT . Visible to true
```

Programming “Arduino” Sketches

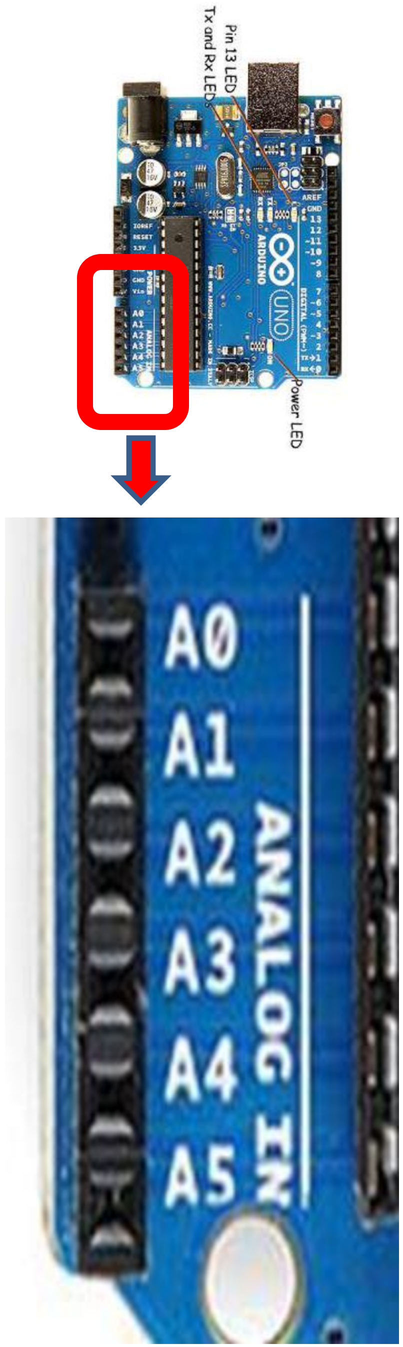
Analog Inputs

Instructor / Facilitator - Alan Rux

"Platform"

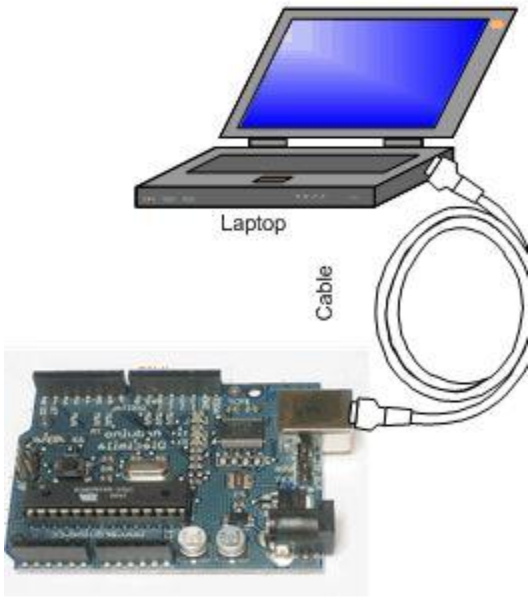
Analog Inputs

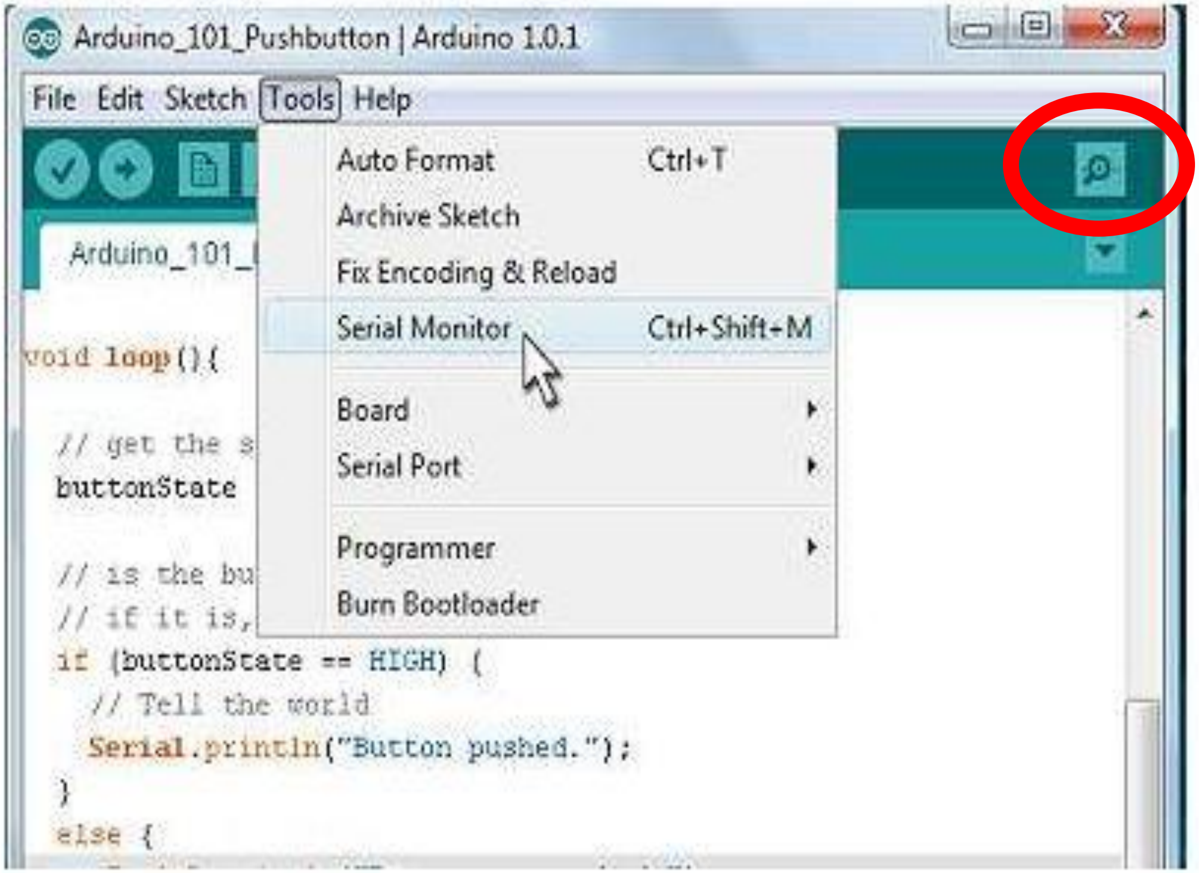
- Six Channels A/D converter
- Pins A0 to A5
- Input voltage = 0v. To + 5v
- 1024 bits conversion (0-1023)
  (10 digital bit converter)
- .004889 volts / step
- Example:
  0 bits = 0 volts
  256 bits = + 1.25 volts
  512 bits = +2.5 volts
  1023 bits = + 5 volts



We will use the **serial monitor** on the Arduino IDE for display of converted data

(same as in Sketches, Digital Inputs/Outputs)





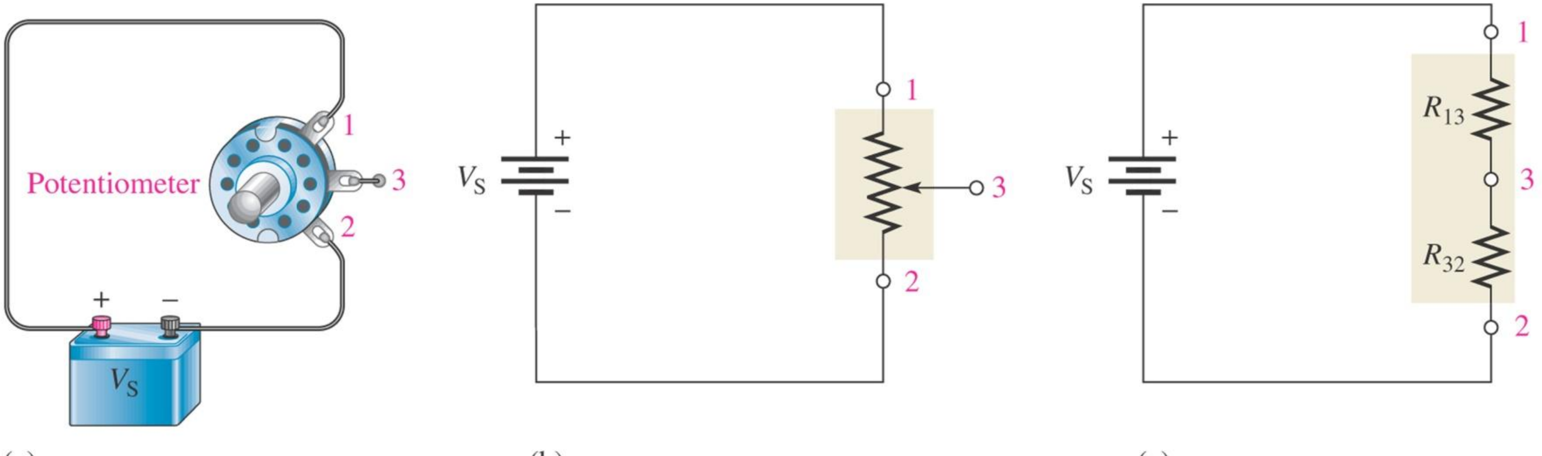
```
void loop(){

  // get the s
  buttonState

  // is the bu
  // if it is,
  if (buttonState == HIGH) {
    // Tell the world
    Serial.println("Button pushed.");
  }
  else {
```

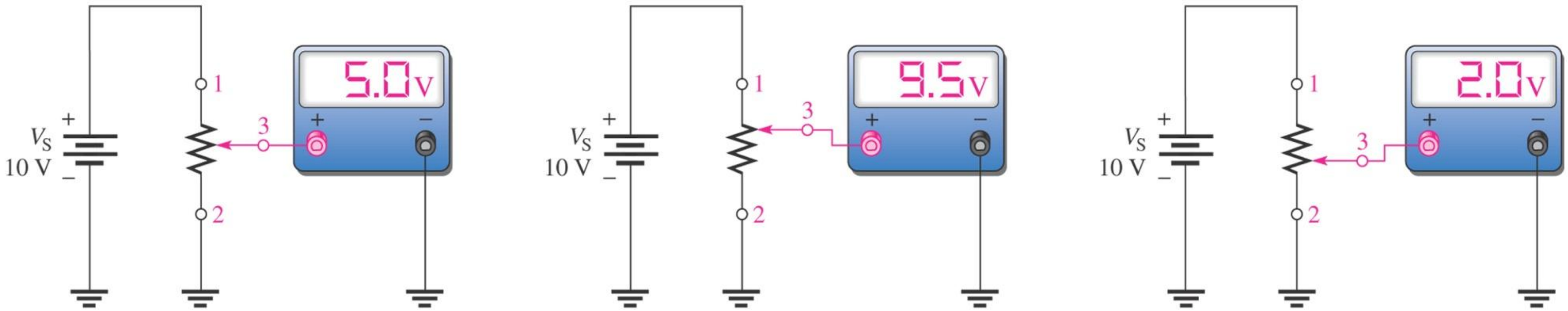
Potentiometer as an Adjustable Voltage Divider

- A **potentiometer** is a variable resistor used to divide voltage
- The potentiometer shown below is equivalent to a two-resistor voltage divider that can be manually adjusted
- The two resistors are between terminals 1 & 3 and 2 & 3



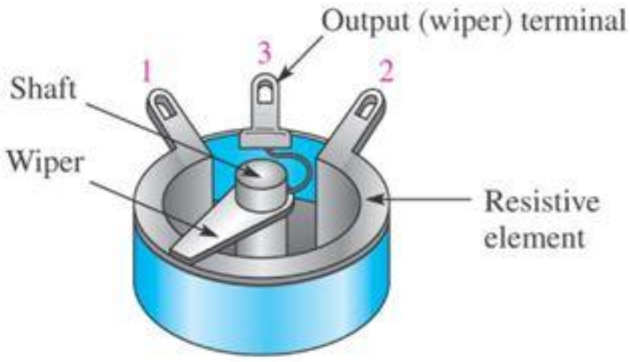
Adjusting the voltage divider

(potentiometer)

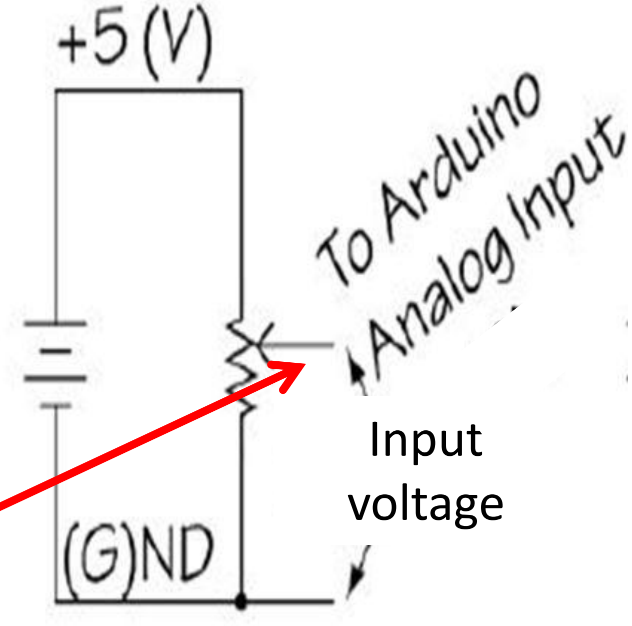


(a)

(b)



(d) Basic construction (simplified)

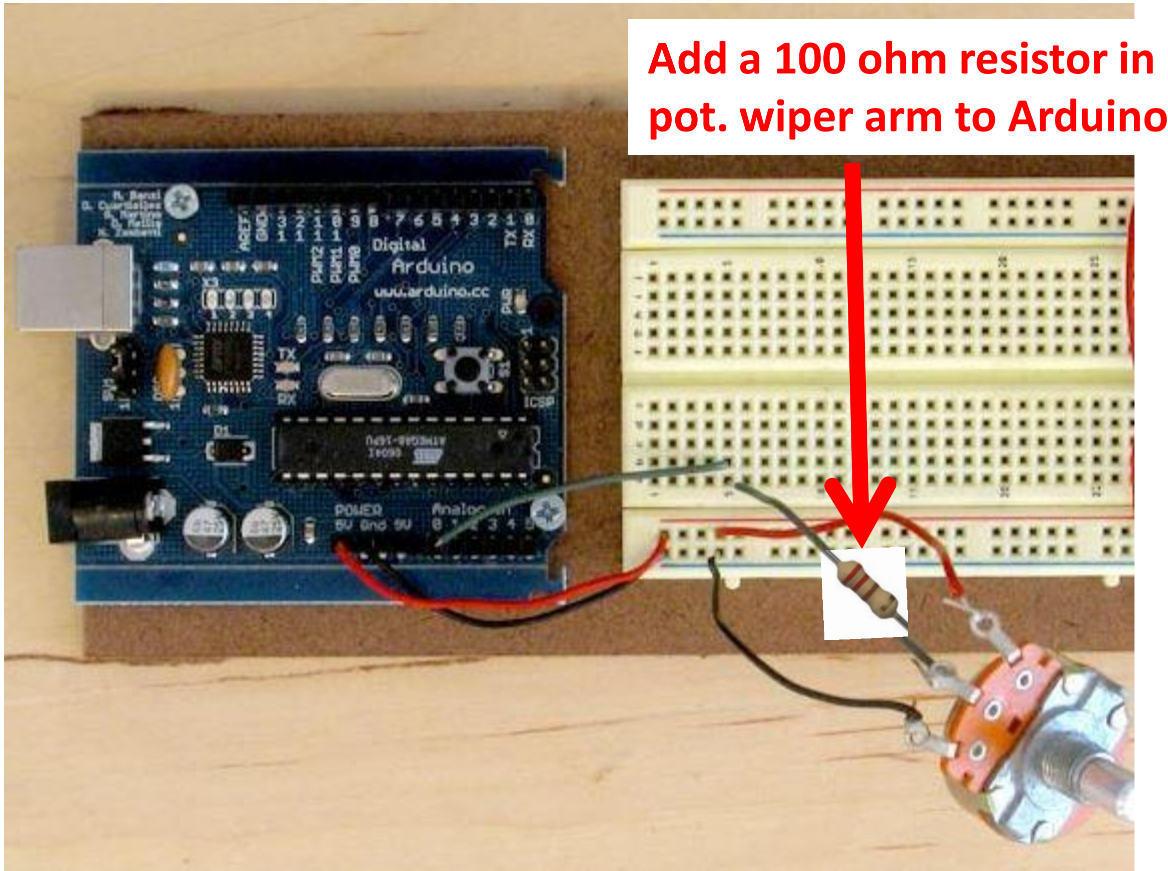


Input voltage

Add a 100 ohm resistor in series from pot. wiper arm to Arduino pin A0

⇦ 0 volts

Wiring the potentiometer to Arduino



Add a 100 ohm resistor in series from pot. wiper arm to Arduino pin A0

You will have to solder wires to potentiometer,
jumper to breadboard is optional

analogRead( )

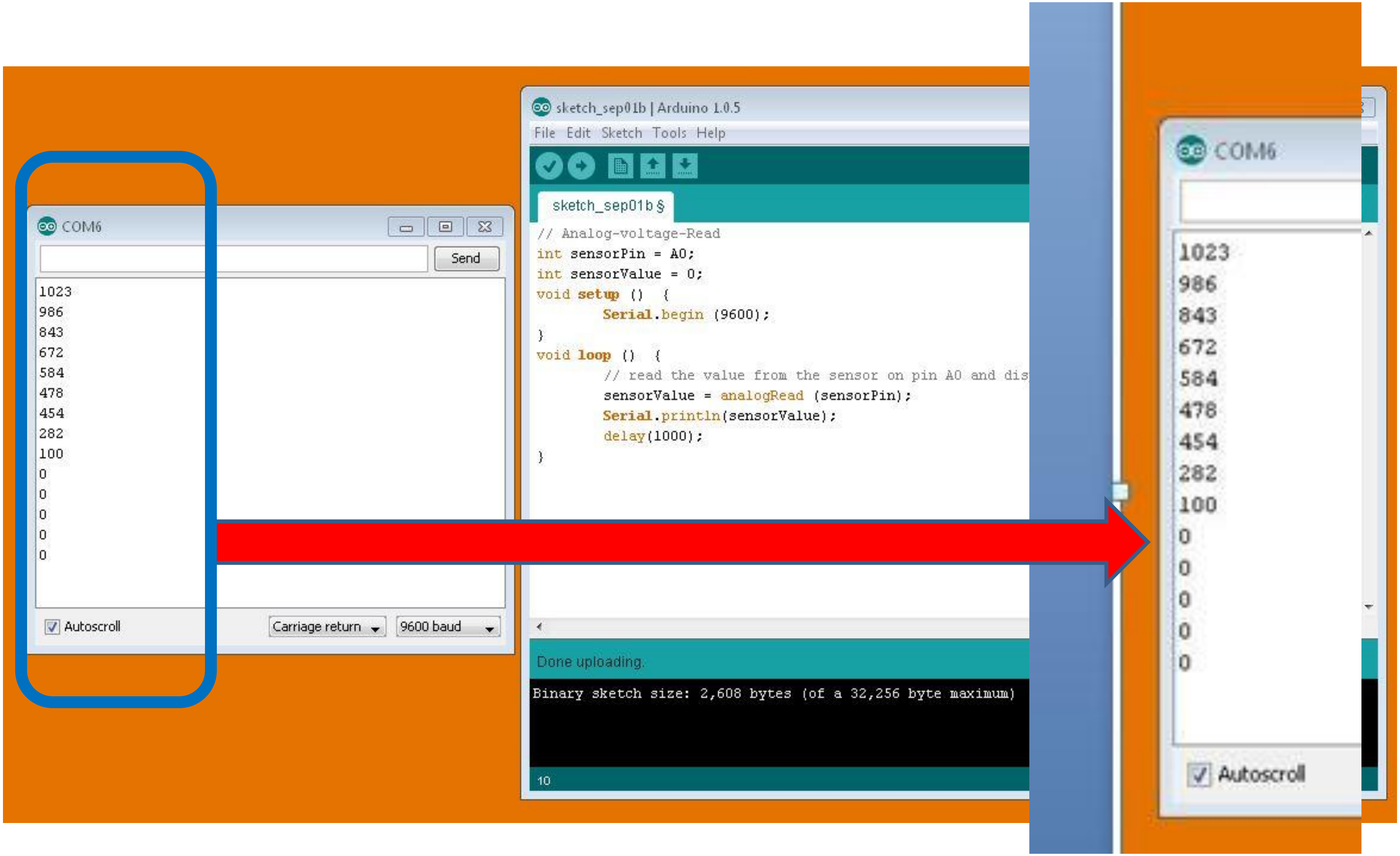
- Reads the value from the specified analog pin. The Arduino board contains a 6 channel 10-bit analog to digital converter. This means that it will map input voltages between 0 and 5 volts into integer values between 0 and 1023. This yields a resolution between readings of: 5 volts / 1024 units or, .0049 volts
- It takes about 100 microseconds (0.0001 s) to read an analog input, so the maximum reading rate is about 10,000 times a second.

Potentiometer Read Sketch

```
// Analog-Input-Read
int sensorPin = A0;
int sensorValue = 0;
void setup ()  {
    Serial.begin (9600);  //setting up baud rate to serial monitor
}
void loop ()  {
    // read the value from the sensor and display it every second
    sensorValue = analogRead (sensorPin);
    Serial.println (sensorValue);   // sending value to serial monitor
delay(1000);  // delay of one second
}   // loop around again
```

This **sketch** should print 0 to 1023 depending on the position of Pot. Wiper contact, watch the serial Leds flash as sending data to monitor

Potentiometer Read Sketch



```
1023
986
843
672
584
478
454
282
100
0
0
0
0
0
```

```
// Analog-voltage-Read
int sensorPin = A0;
int sensorValue = 0;
void setup ()  {
        Serial.begin (9600);
}
void loop ()  {
        // read the value from the sensor on pin A0 and dis
        sensorValue = analogRead (sensorPin);
        Serial.println(sensorValue);
        delay(1000);
}
```

Done uploading.

```
Binary sketch size: 2,608 bytes (of a 32,256 byte maximum)
```

10

COM6

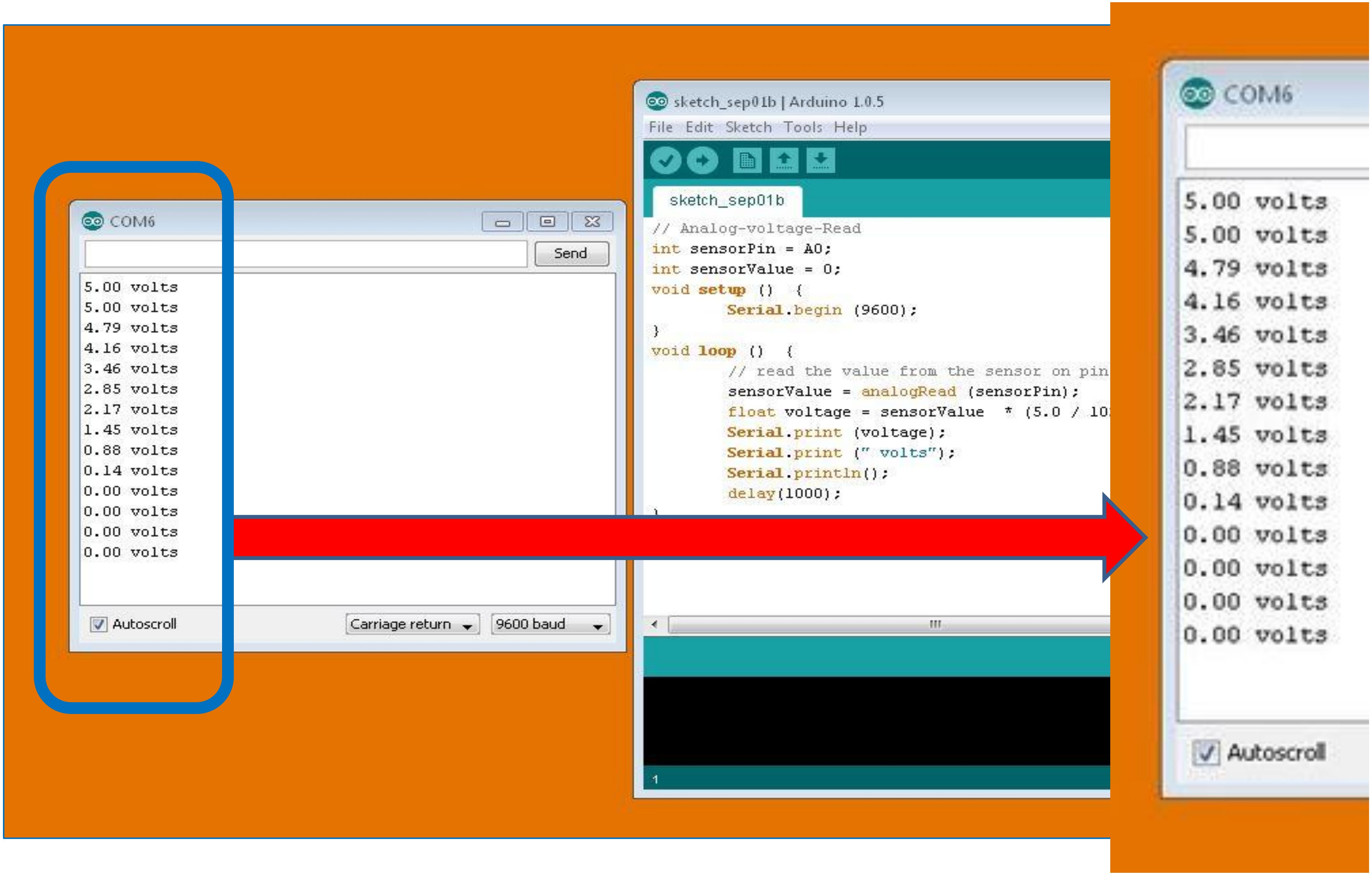
1023
986
843
672
584
478
454
282
100
0
0
0
0
0

☑ Autoscroll

Analog Voltmeter Sketch 0 to 5 volts

```
// Analog-voltage-Read
int sensorPin = A0;
int sensorValue = 0;
void setup ()  {
      Serial.begin (9600);
}
void loop ()  {
      // read the value from the sensor on pin A0 and display it every second
      sensorValue = analogRead (sensorPin);
      float  voltage = sensorValue  * (5.0 / 1023.0); // converts from digital to voltage
      Serial.print(voltage);  // prints the converter voltage reading
      Serial.print (“  volts”);  // adds the word “volts”
      Serial.println();  // carriage return, next line
      delay(1000);
}
```

Analog Voltmeter Sketch 0 to 5 volts


COM6
Send
sketch_sep01b | Arduino 1.0.5
File Edit Sketch Tools Help
sketch_sep01b
1
COM6

```
5.00 volts
5.00 volts
4.79 volts
4.16 volts
3.46 volts
2.85 volts
2.17 volts
1.45 volts
0.88 volts
0.14 volts
0.00 volts
0.00 volts
0.00 volts
0.00 volts
```

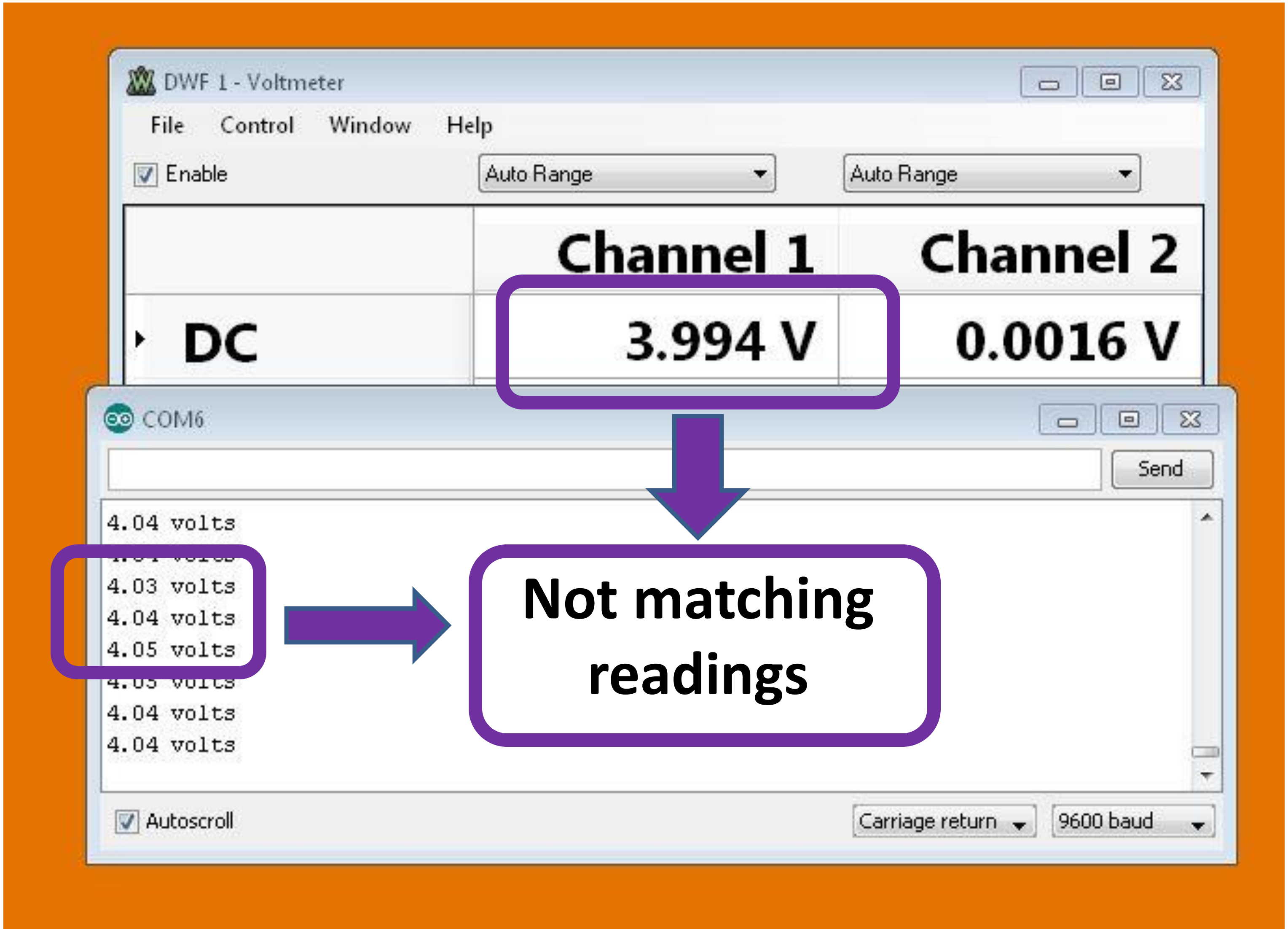
☑ Autoscroll   Carriage return   9600 baud

```
// Analog-voltage-Read
int sensorPin = A0;
int sensorValue = 0;
void setup ()  {
        Serial.begin (9600);
}
void loop ()  {
        // read the value from the sensor on pin
        sensorValue = analogRead (sensorPin);
        float voltage = sensorValue  * (5.0 / 10
        Serial.print (voltage);
        Serial.print (" volts");
        Serial.println();
        delay(1000);
}
```

```
5.00 volts
5.00 volts
4.79 volts
4.16 volts
3.46 volts
2.85 volts
2.17 volts
1.45 volts
0.88 volts
0.14 volts
0.00 volts
0.00 volts
0.00 volts
0.00 volts
```

☑ Autoscroll

Using your **Analog Discovery Kit Voltmeter** measure the voltage on wiper arm as compared to value displayed on the monitor



	Channel 1	Channel 2
DC	3.994 V	0.0016 V

```
4.04 volts
4.04 volts
4.03 volts
4.04 volts
4.05 volts
4.05 volts
4.04 volts
4.04 volts
```

☑ Autoscroll

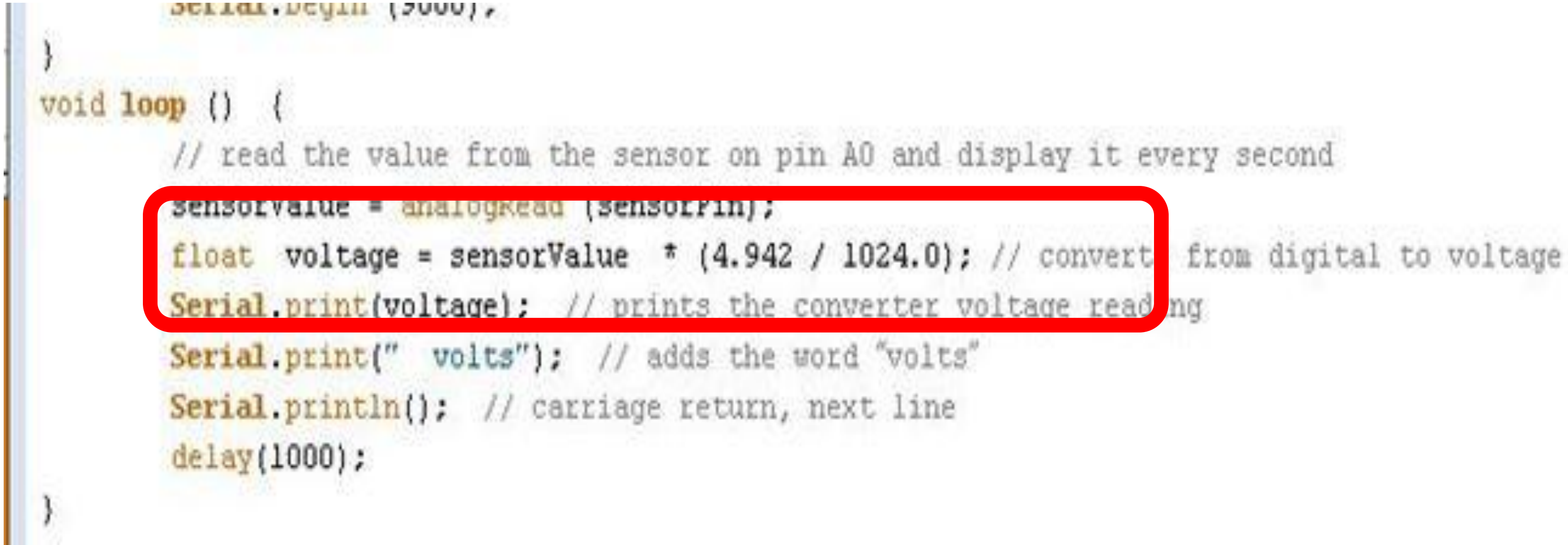
Carriage return

9600 baud

Not matching readings

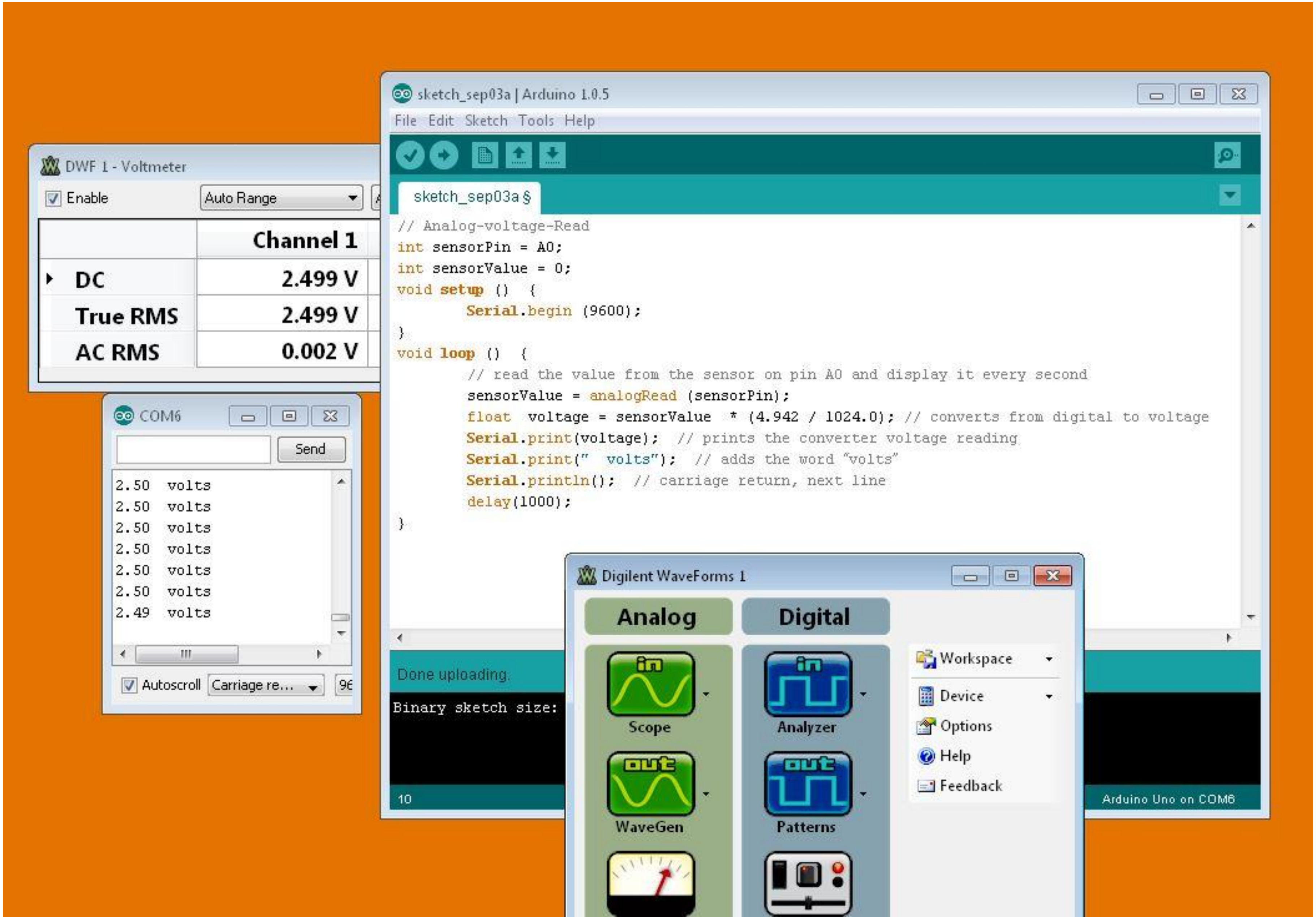
error correction

- With ADK meter measured +5 volts from Arduino , it was found to be + 4.942 volts DC, not +5.0 volts DC
- Changed the input max voltage in math function to 4.942
- The Arduino has a 10 bit A/D converter = 1024 steps
- In math function changed divisor function to 1024
- Retested voltage readings, much improved



```
        Serial.begin (9600);
}
void loop ()  {
        // read the value from the sensor on pin A0 and display it every second
        sensorValue = analogRead (sensorPin);
        float  voltage = sensorValue  * (4.942 / 1024.0); // convert  from digital to voltage
        Serial.print(voltage);  // prints the converter voltage reading
        Serial.print("  volts");  // adds the word "volts"
        Serial.println();  // carriage return, next line
        delay(1000);
}
```

reading error corrected



☑ Enable Auto Range

	Channel 1
DC	2.499 V
True RMS	2.499 V
AC RMS	0.002 V

```
// Analog-voltage-Read
int sensorPin = A0;
int sensorValue = 0;
void setup ()  {
        Serial.begin (9600);
}
void loop ()  {
        // read the value from the sensor on pin A0 and display it every second
        sensorValue = analogRead (sensorPin);
        float  voltage = sensorValue  * (4.942 / 1024.0); // converts from digital to voltage
        Serial.print(voltage);  // prints the converter voltage reading
        Serial.print("  volts");  // adds the word "volts"
        Serial.println();  // carriage return, next line
        delay(1000);
}
```

```
2.50  volts
2.50  volts
2.50  volts
2.50  volts
2.50  volts
2.50  volts
2.49  volts
```

☑ Autoscroll Carriage re...

Done uploading.

```
Binary sketch size:
```

10

Arduino Uno on COM6

Analog Input Examples (page 1 of 2)

MAKING DECISIONS & CREATING WORKING SYSTEMS

Voltage Controlled Light (on/off)

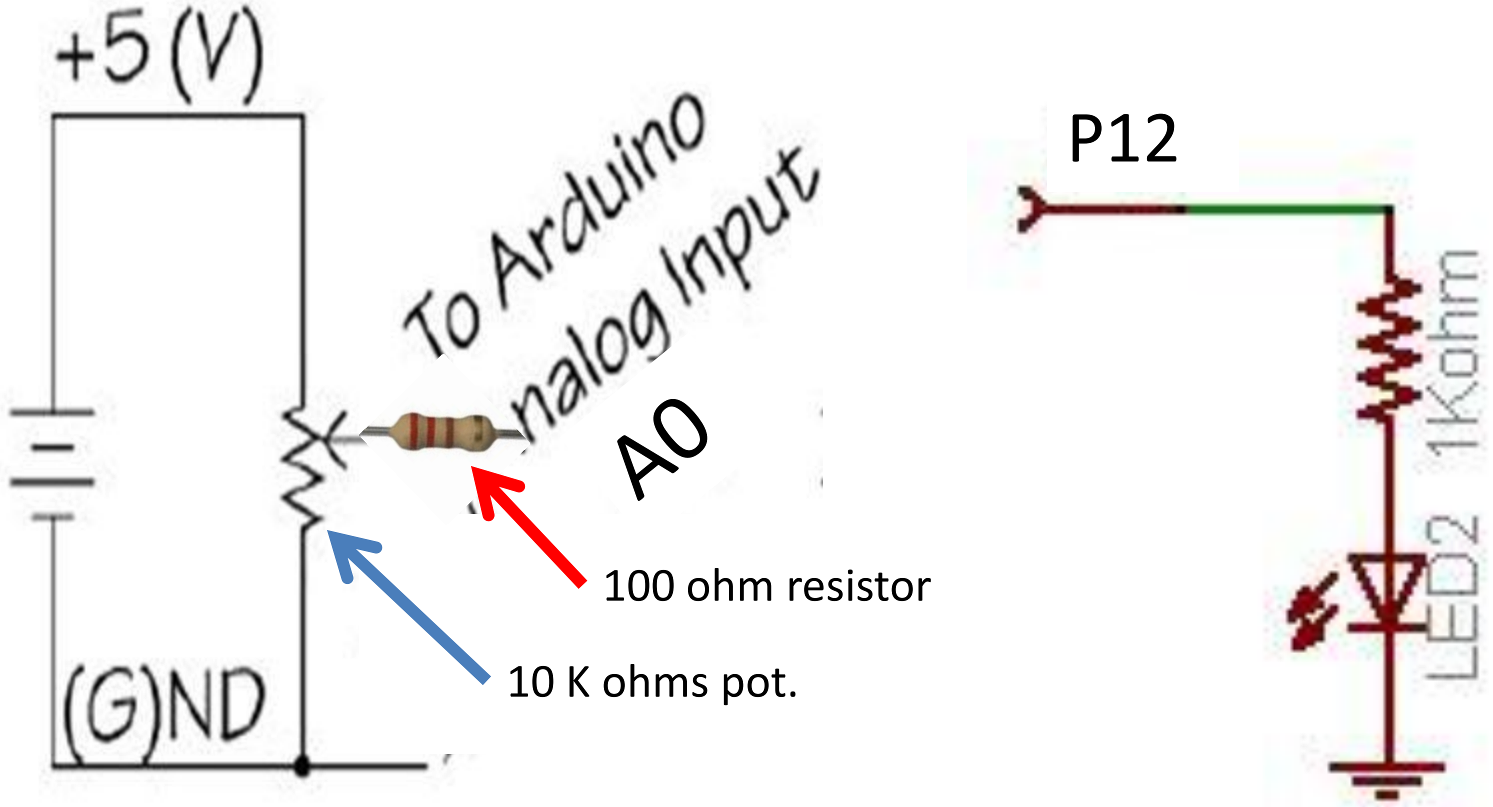
```
/* Analog Read to LED  turns on and off a light emitting diode(LED) connected
    to digital  pin 13. the LED will be on or off depending on the value obtained
    by analogRead(). */
 int potPin = A0; // select the input pin for the potentiometer
int potValue=0; // variable to store the value coming from the pot.
 int ledPin = 12;   // select the pin for the LED
int ledValue=LOW
void setup() {
    pinMode(ledPin,OUTPUT);   // declare the ledPin as an OUTPUT
    Serial.begin(9600);   // serial communication
}
```

Analog Input Examples (page 2 of 2)

MAKING DECISIONS & CREATING WORKING SYSTEMS

```
void loop() {
    potValue = analogRead(potPin);   // read the value from the pot
    serial.println(potValue);  // send value to monitor
    if  (potValue >=  512.)
    {
    digitalWrite(ledPin,High); // led on if value is > 512
    delay(300);  // delay 300 ms.
    }
    if  (potValue < 512.){
    digitalWrite(ledPin.LOW); // led off if value is < 512
    delay(300); // delay 300 ms.
    } }  // loop
```

schematic



P12

A0

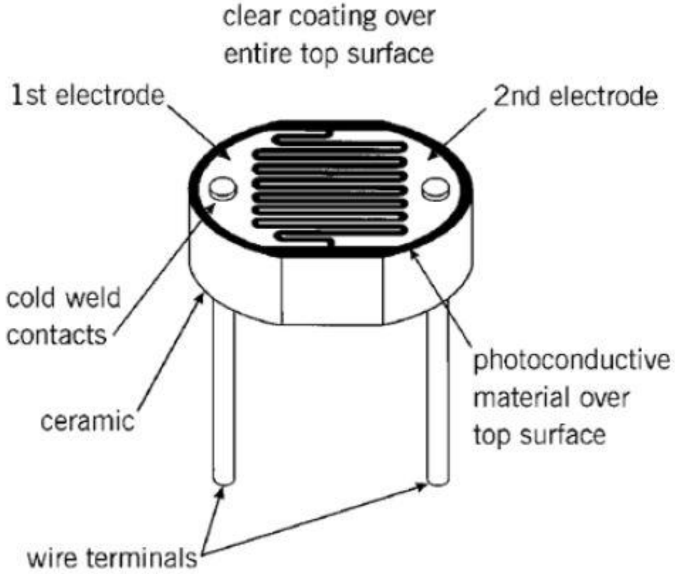
100 ohm resistor

10 K ohms pot.

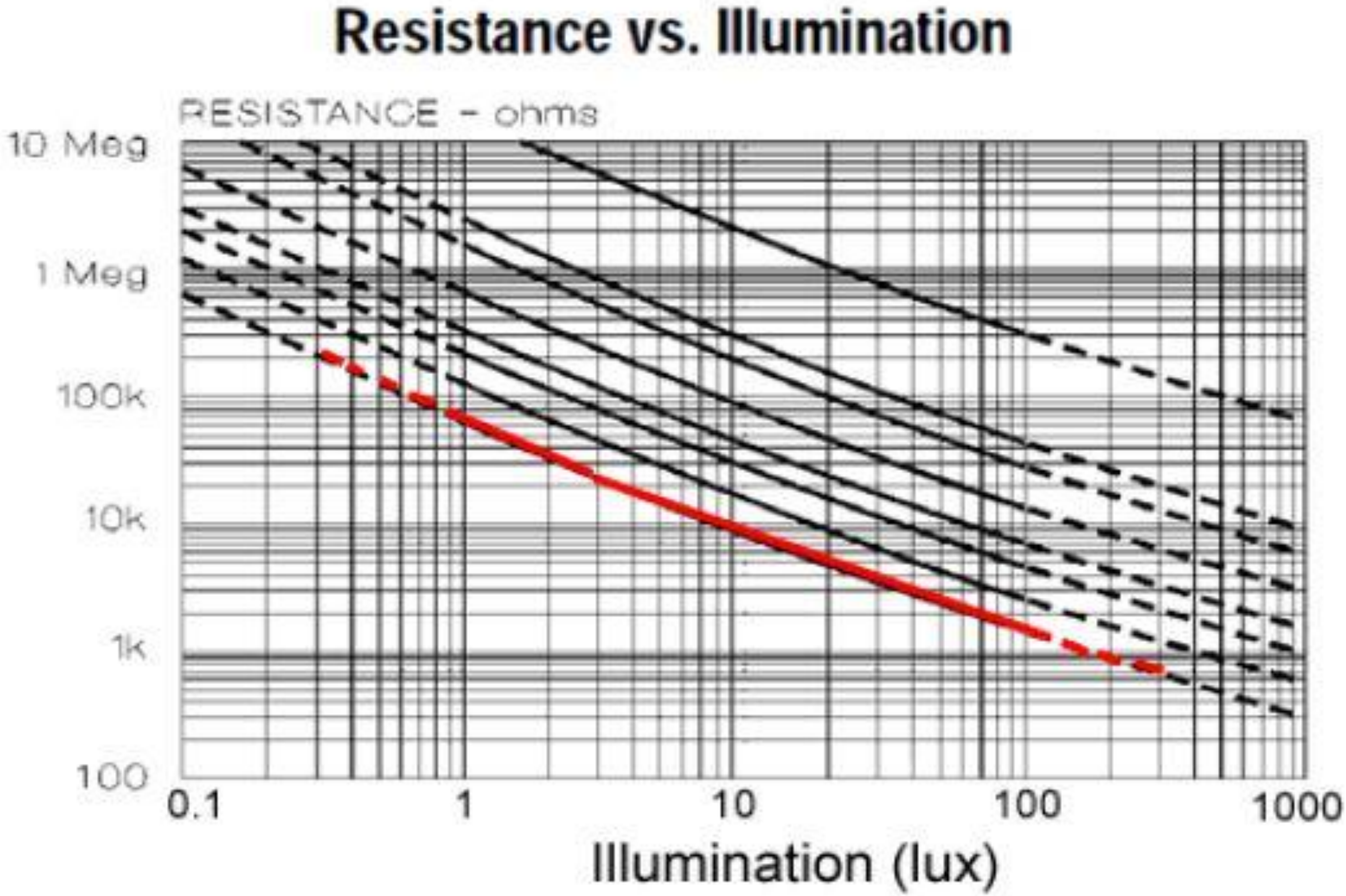
Analog Input Examples
MAKING DECISIONS & CREATING WORKING SYSTEMS
PhotoCells

- Photocells are sensors that allow you to detect light. They are small, inexpensive, low-power,

  easy to use They are often referred to as CdS cells **l**ight-**d**ependent **r**esistors (LDR), and photoresistors.

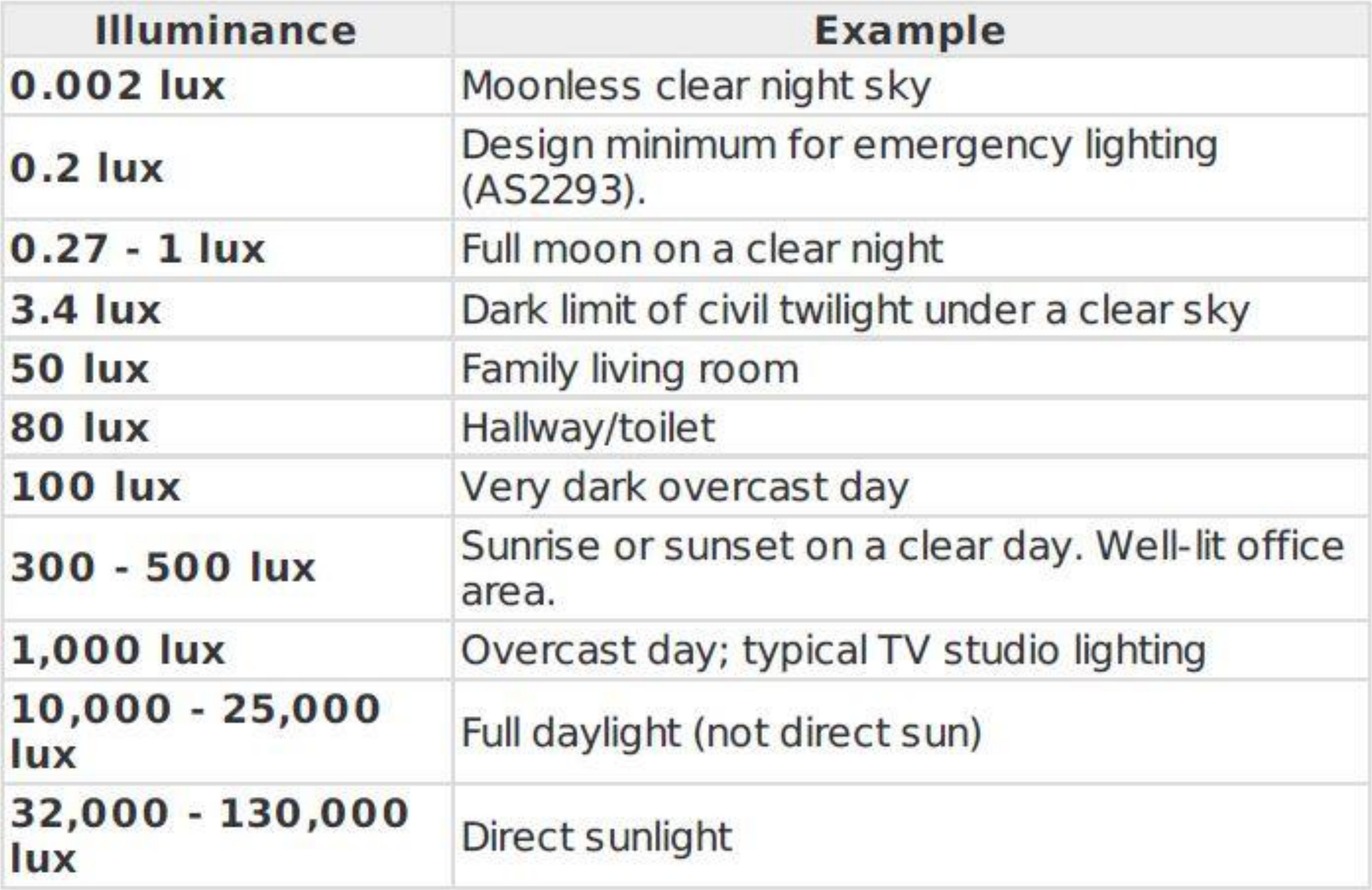

clear coating over entire top surface
1st electrode
2nd electrode
cold weld contacts
ceramic
photoconductive material over top surface
wire terminals

Photocell


Resistance vs. Illumination
RESISTANCE - ohms
10 Meg
1 Meg
100k
10k
1k
100
0.1
1
10
100
1000
Illumination (lux)

Photocell

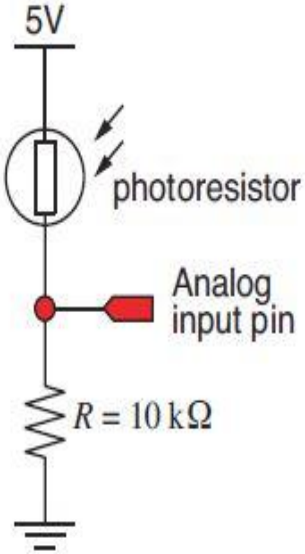
Illuminance typical reference levels

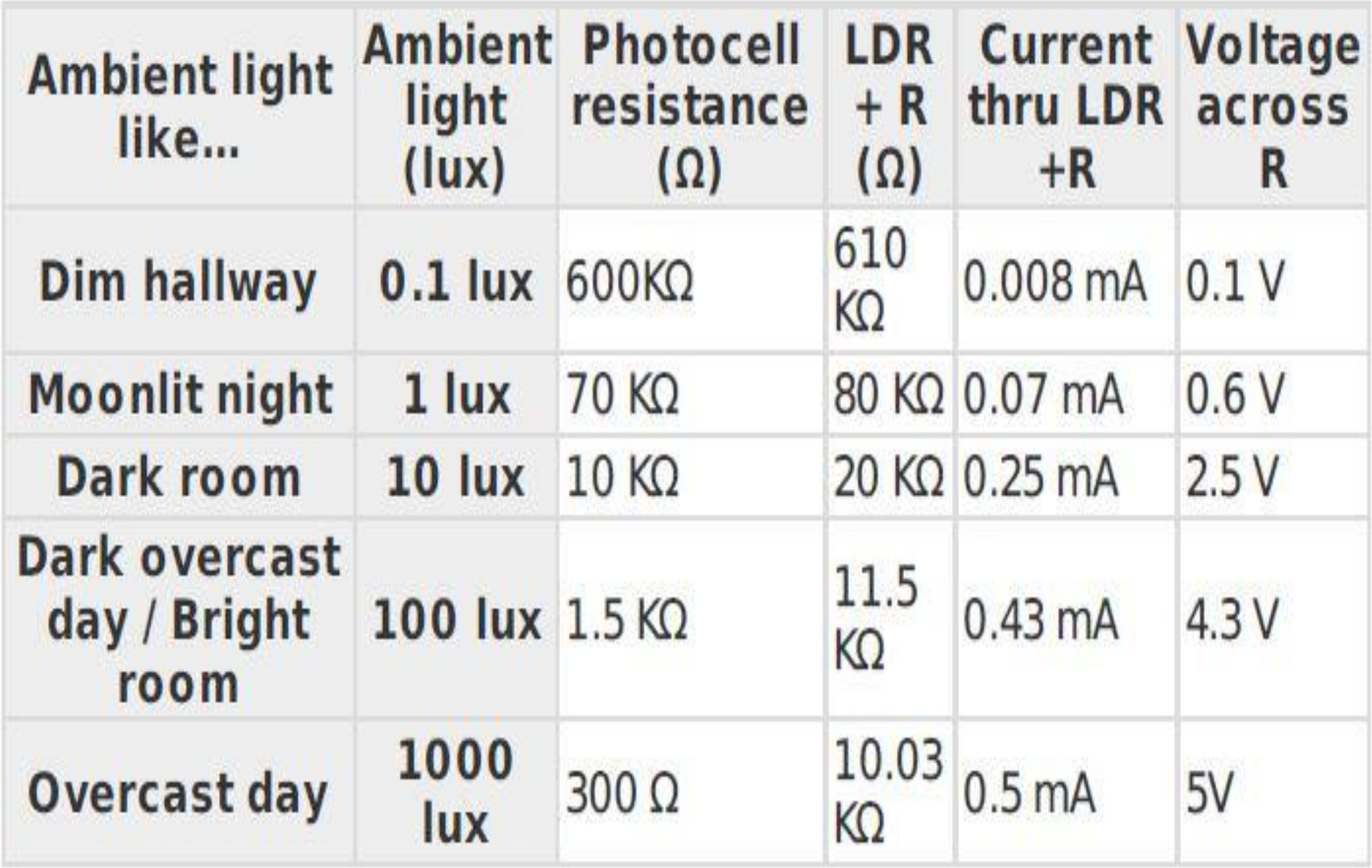


Illuminance	Example
0.002 lux	Moonless clear night sky
0.2 lux	Design minimum for emergency lighting (AS2293).
0.27 - 1 lux	Full moon on a clear night
3.4 lux	Dark limit of civil twilight under a clear sky
50 lux	Family living room
80 lux	Hallway/toilet
100 lux	Very dark overcast day
300 - 500 lux	Sunrise or sunset on a clear day. Well-lit office area.
1,000 lux	Overcast day; typical TV studio lighting
10,000 - 25,000 lux	Full daylight (not direct sun)
32,000 - 130,000 lux	Direct sunlight

Photocell

Lux typical levels related to sensor output in ohms

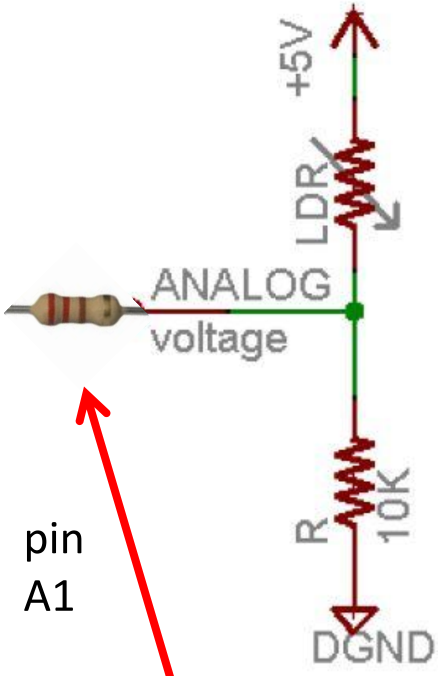




Ambient light like...	Ambient light (lux)	Photocell resistance (Ω)	LDR + R (Ω)	Current thru LDR +R	Voltage across R
Dim hallway	**0.1 lux**	600KΩ	610 KΩ	0.008 mA	0.1 V
Moonlit night	**1 lux**	70 KΩ	80 KΩ	0.07 mA	0.6 V
Dark room	**10 lux**	10 KΩ	20 KΩ	0.25 mA	2.5 V
Dark overcast day / Bright room	**100 lux**	1.5 KΩ	11.5 KΩ	0.43 mA	4.3 V
Overcast day	**1000 lux**	300 Ω	10.03 KΩ	0.5 mA	5V

Light Sensor Sketch (DIY ASDE Project)

- **Replace the potentiometer with a Photocell sensor**
- Use analog input pin A1, keep the pot on pin A0.
- Change the code in the sketch to work with pin A1.
- Use the ADK to measure voltages across the LDR, use something to shield light from LDR.
- Change the code to turn on the led when it becomes dark.
- Use the pot on A0 to change the fixed value of 512 to a settable threshold to trip the Led. on/off, send that value to monitor also.
- Find other things you can do with this type of circuit and code. Explain in class meeting with a demo. Google help is acceptable.



pin A1

Add a 100 ohm resistor in series to Arduino pin A1

Questions