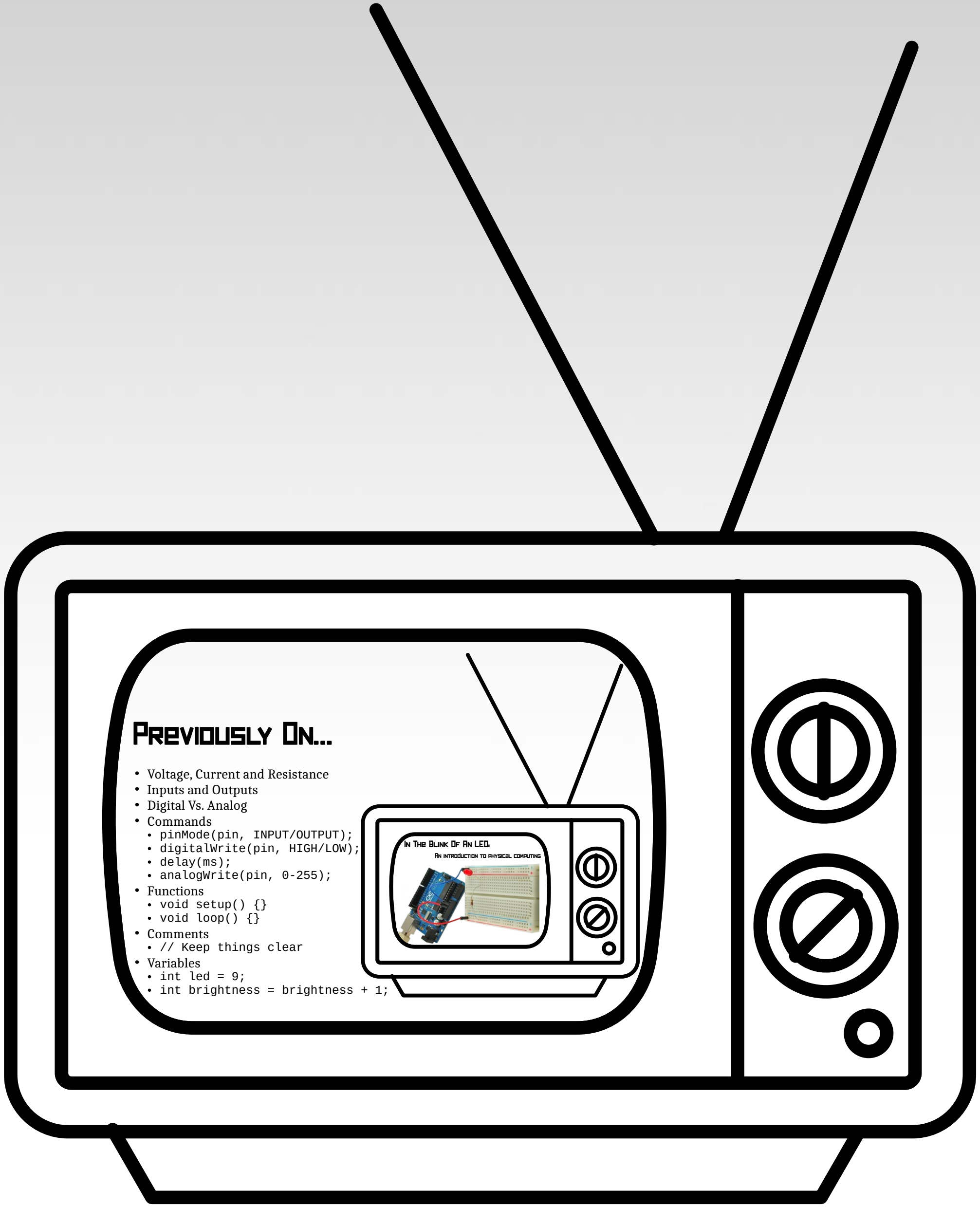
Previously On…

- Digital Inputs



Previously On…

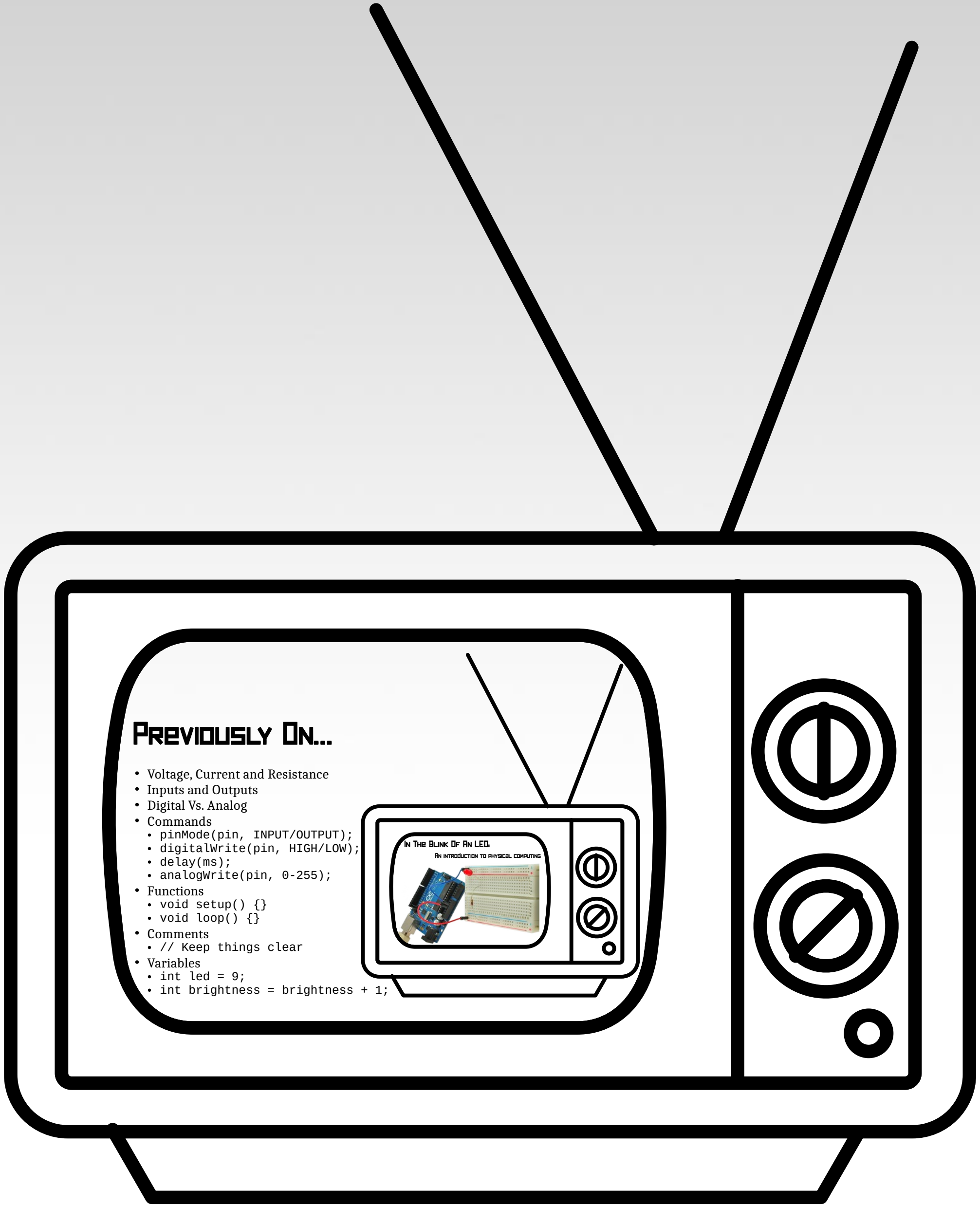
- Voltage, Current and Resistance
- Inputs and Outputs
- Digital Vs. Analog
- Commands
  - `pinMode(pin, INPUT/OUTPUT);`
  - `digitalWrite(pin, HIGH/LOW);`
  - `delay(ms);`
  - `analogWrite(pin, 0-255);`
- Functions
  - `void setup() {}`
  - `void loop() {}`
- Comments
  - `// Keep things clear`
- Variables
  - `int led = 9;`
  - `int brightness = brightness + 1;`

In The Blink Of An LED:

An introduction to physical computing

Previously On...

- Digital Inputs
- if() statements



Previously On...

- Voltage, Current and Resistance
- Inputs and Outputs
- Digital Vs. Analog
- Commands
  - `pinMode(pin, INPUT/OUTPUT);`
  - `digitalWrite(pin, HIGH/LOW);`
  - `delay(ms);`
  - `analogWrite(pin, 0-255);`
- Functions
  - `void setup() {}`
  - `void loop() {}`
- Comments
  - `// Keep things clear`
- Variables
  - `int led = 9;`
  - `int brightness = brightness + 1;`

In The Blink Of An LED:

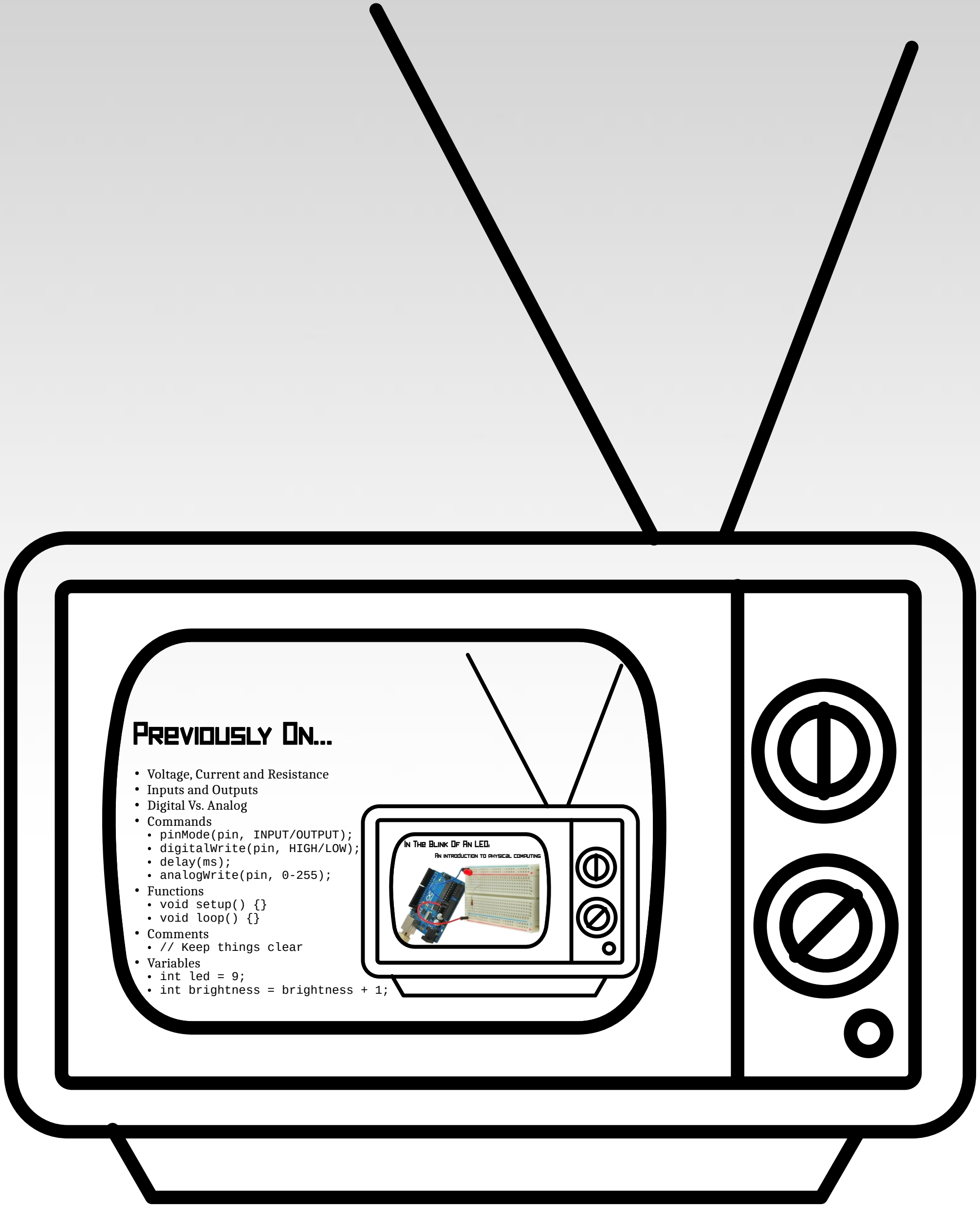
An introduction to physical computing

Boolean Operators

<Boolean>	True if:
(a)	a is true, HIGH or does not equal zero
(!a)	a is false, LOW or equals zero
(a) == (b)	a is equal to b
(a) != (b)	a is not equal to b
(a) > (b)	a is greater than b
(a) >= (b)	a is greater than or equal to b
(a) < (b)	a is less than b
(a) <= (b)	a is less than or equal to b
(a) && (b)	both a is true AND b is true
(a) || (b)	either a is true OR b is true

Previously On…

- Digital Inputs
- if() statements



Previously On…

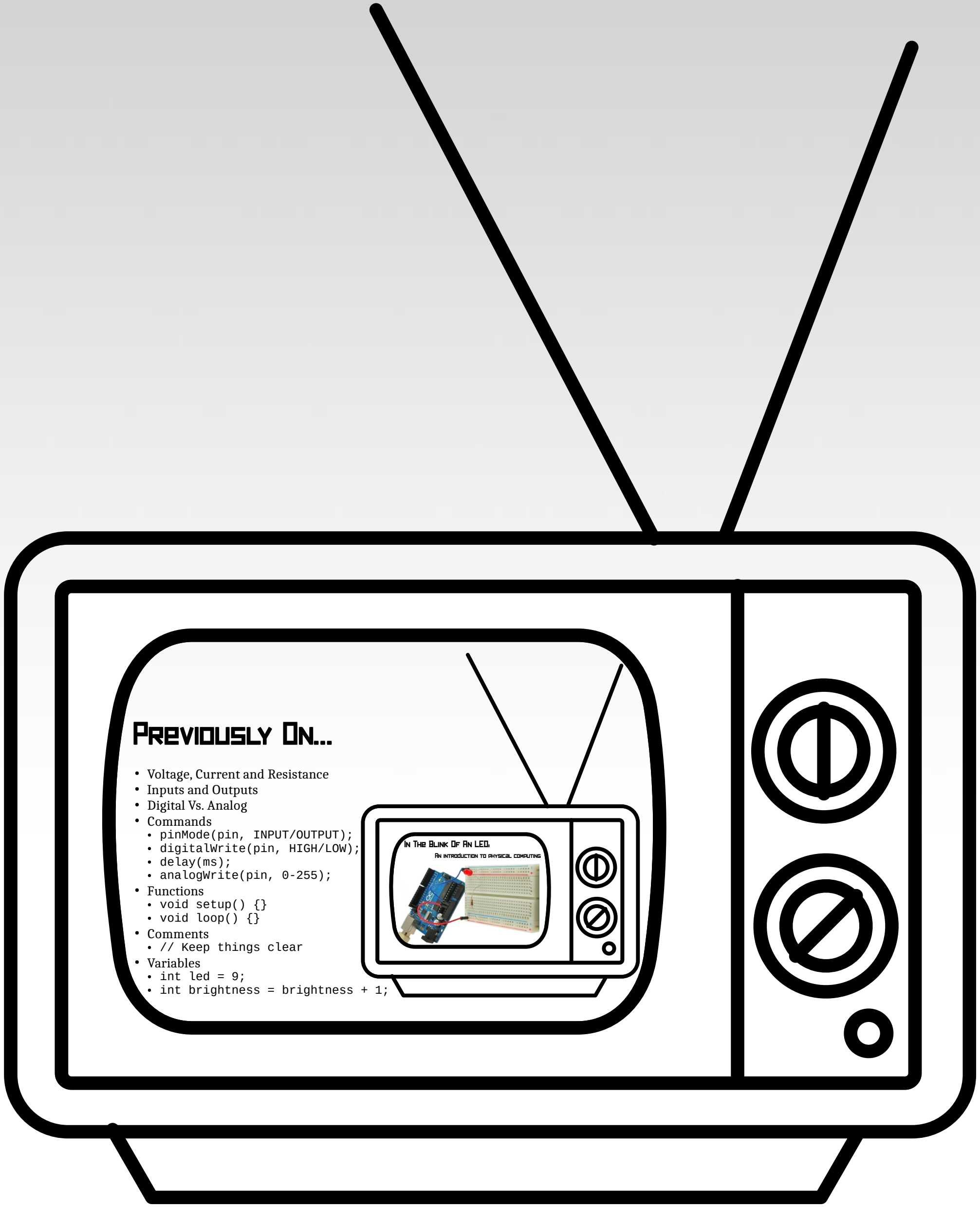
- Voltage, Current and Resistance
- Inputs and Outputs
- Digital Vs. Analog
- Commands
  - pinMode(pin, INPUT/OUTPUT);
  - digitalWrite(pin, HIGH/LOW);
  - delay(ms);
  - analogWrite(pin, 0-255);
- Functions
  - void setup() {}
  - void loop() {}
- Comments
  - // Keep things clear
- Variables
  - int led = 9;
  - int brightness = brightness + 1;

In The Blink Of An LED:

An introduction to physical computing

Previously On…

- Digital Inputs
- if() statements
- Pull-up resistors



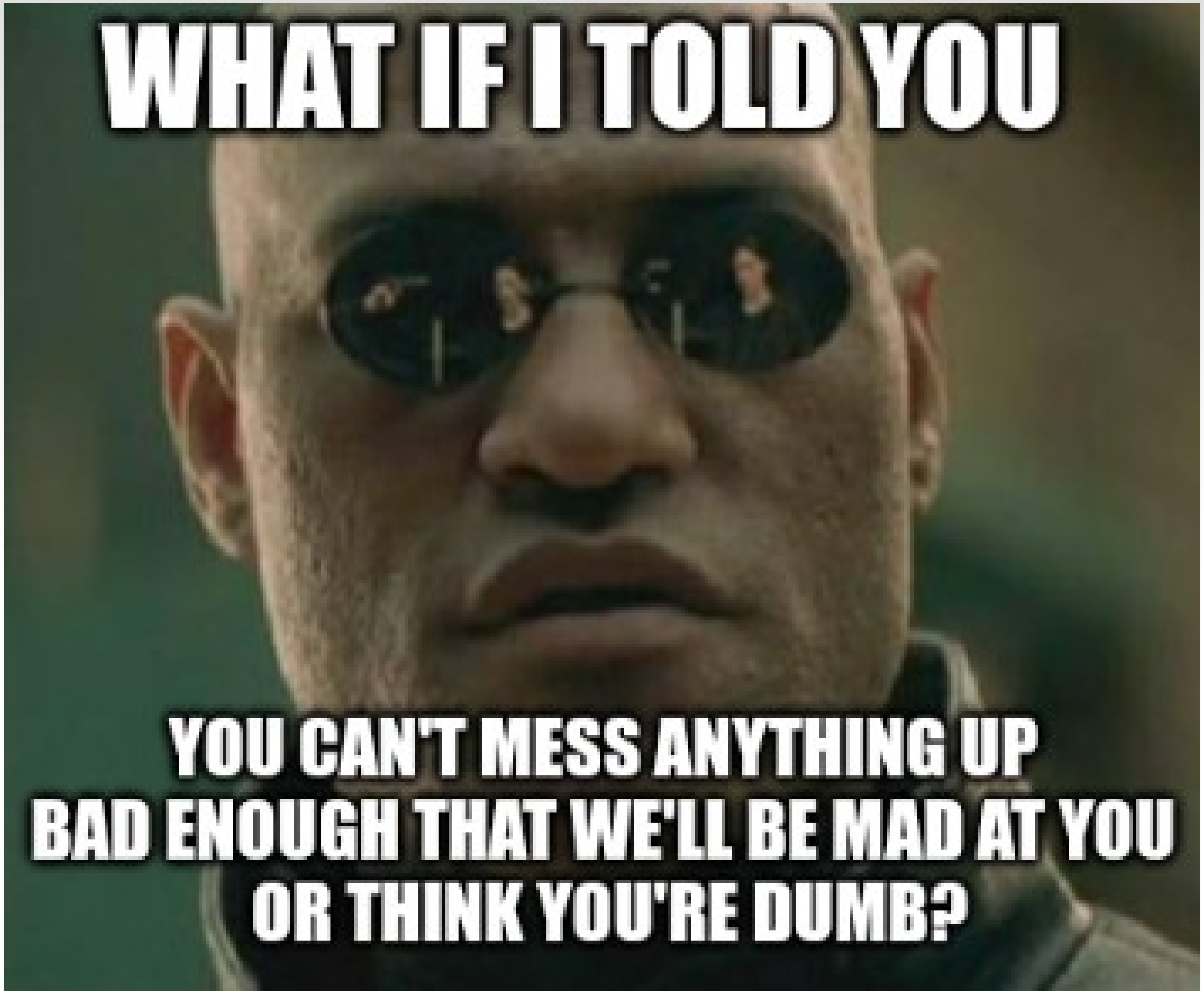
Previously On…

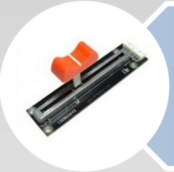
- Voltage, Current and Resistance
- Inputs and Outputs
- Digital Vs. Analog
- Commands
  - `pinMode(pin, INPUT/OUTPUT);`
  - `digitalWrite(pin, HIGH/LOW);`
  - `delay(ms);`
  - `analogWrite(pin, 0-255);`
- Functions
  - `void setup() {}`
  - `void loop() {}`
- Comments
  - `// Keep things clear`
- Variables
  - `int led = 9;`
  - `int brightness = brightness + 1;`

In The Blink Of An LED:

An introduction to physical computing

Seriously, the one thing to take away


WHAT IF I TOLD YOU
YOU CAN'T MESS ANYTHING UP
BAD ENOUGH THAT WE'LL BE MAD AT YOU
OR THINK YOU'RE DUMB?

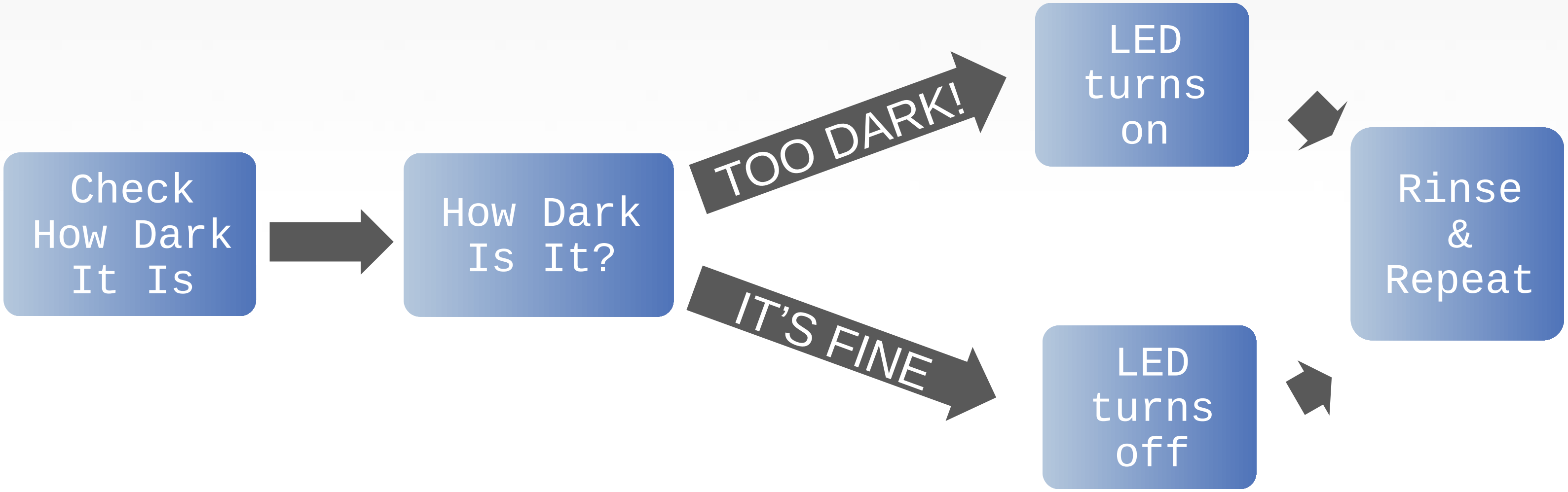


analogRead()

Project # 5 – Night Light

How Dark Is Too Dark?

Psuedo-code – how should this work?



Check How Dark It Is

How Dark Is It?

TOO DARK!

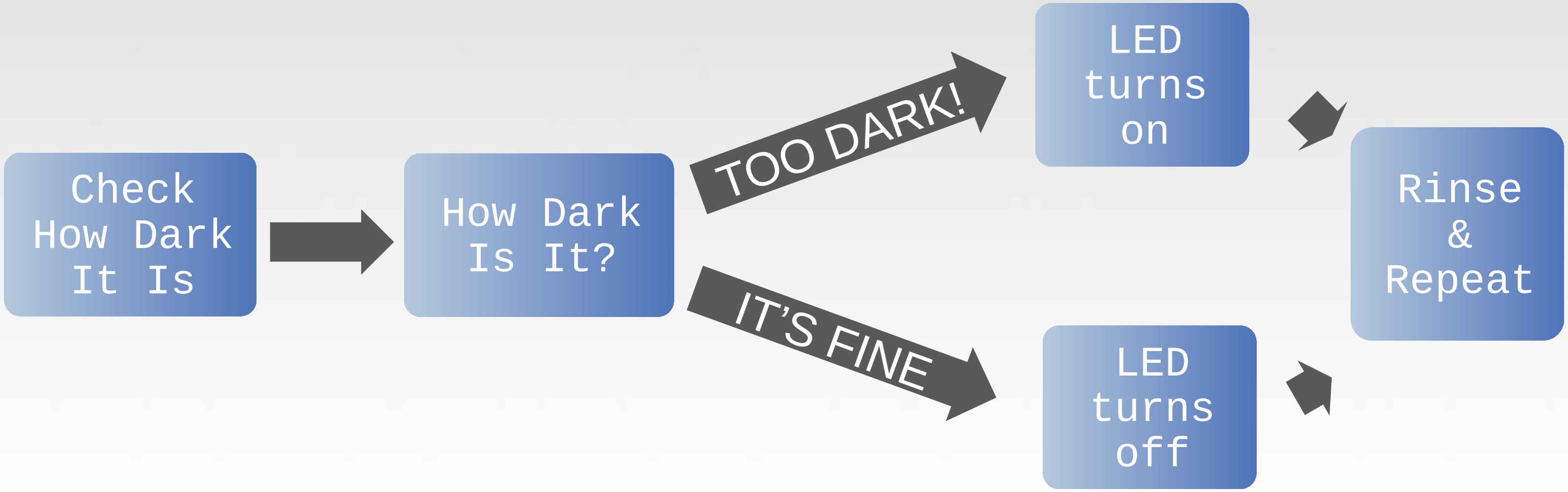
LED turns on

IT'S FINE

LED turns off

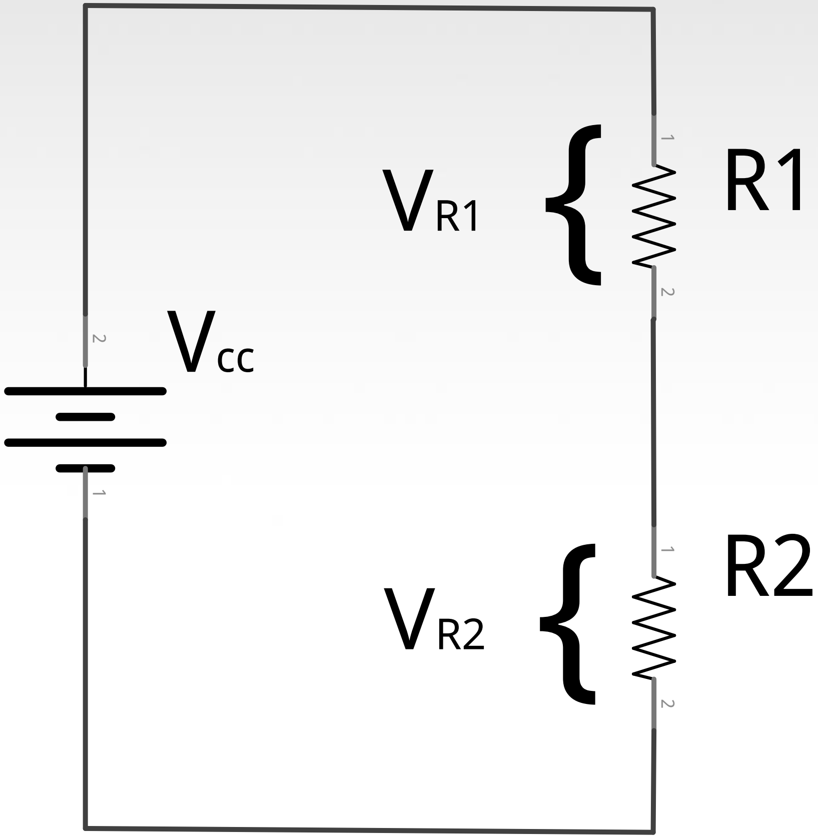
Rinse & Repeat

Project # 4 - Night Light
Inputs and Outputs



Inputs	Outputs
Photoresistor	LED

The Voltage Divider

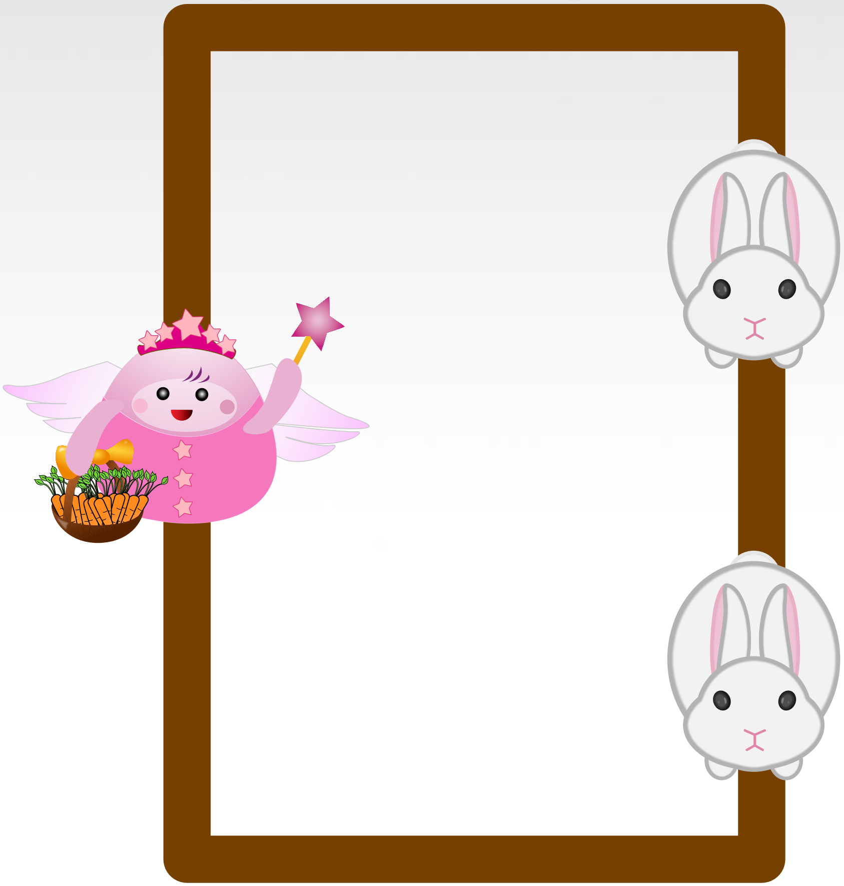


$$V_{R1} = V_{CC} \cdot \left(\frac{R_1}{\mathrm{R}_{Total}} \right)$$

$$V_{R2} = V_{CC} \cdot \left(\frac{R_2}{\mathrm{R}_{Total}} \right)$$

$$R_{Total} = R_1 + R_2$$

The Voltage Divider



$$\xcancel{V_{R1} = V_{CC} \cdot \left(\frac{R_1}{\mathrm{R}_{Total}}\right)}$$

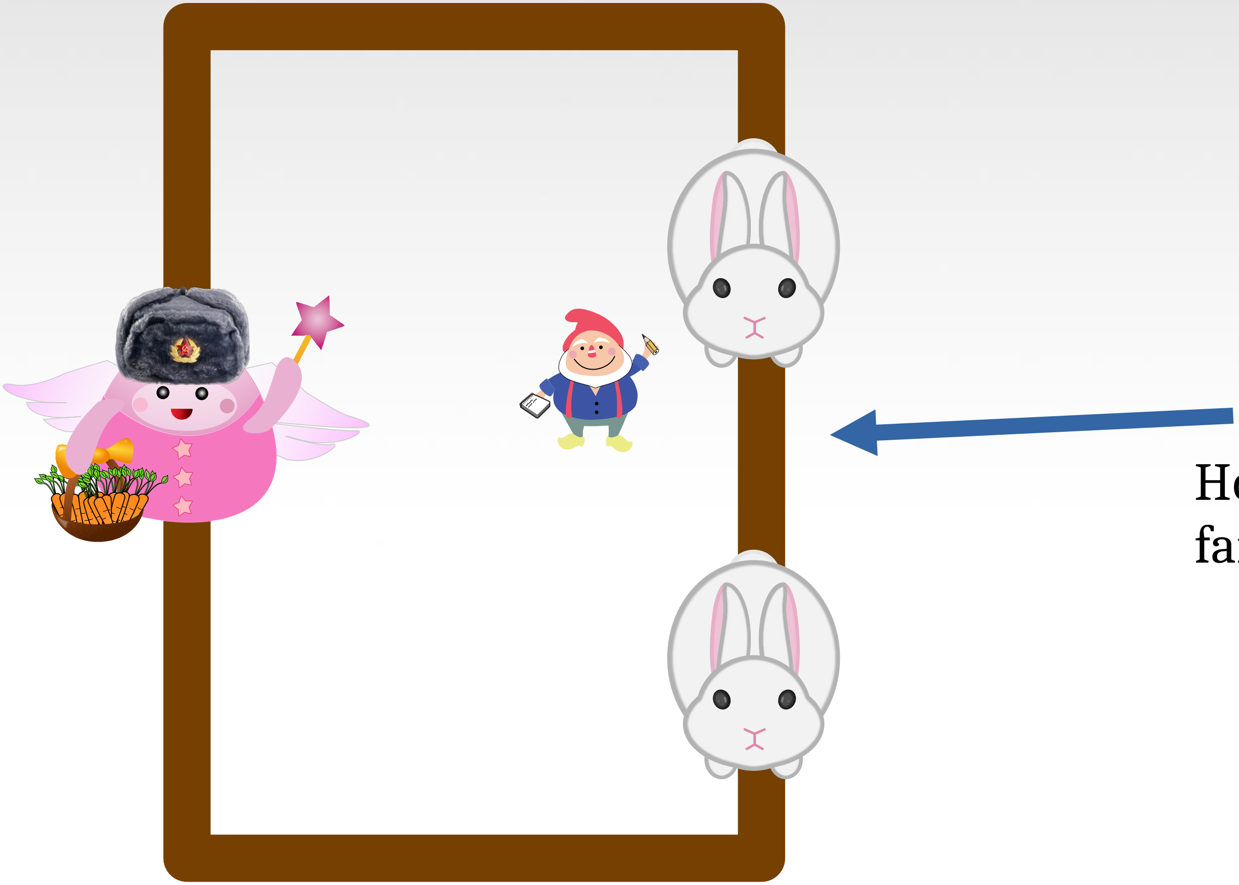
$$\xcancel{V_{R2} = V_{CC} \cdot \left(\frac{R_2}{\mathrm{R}_{Total}}\right)}$$

$$\xcancel{R_{Total} = R_1 + R_2}$$

The Voltage Divider

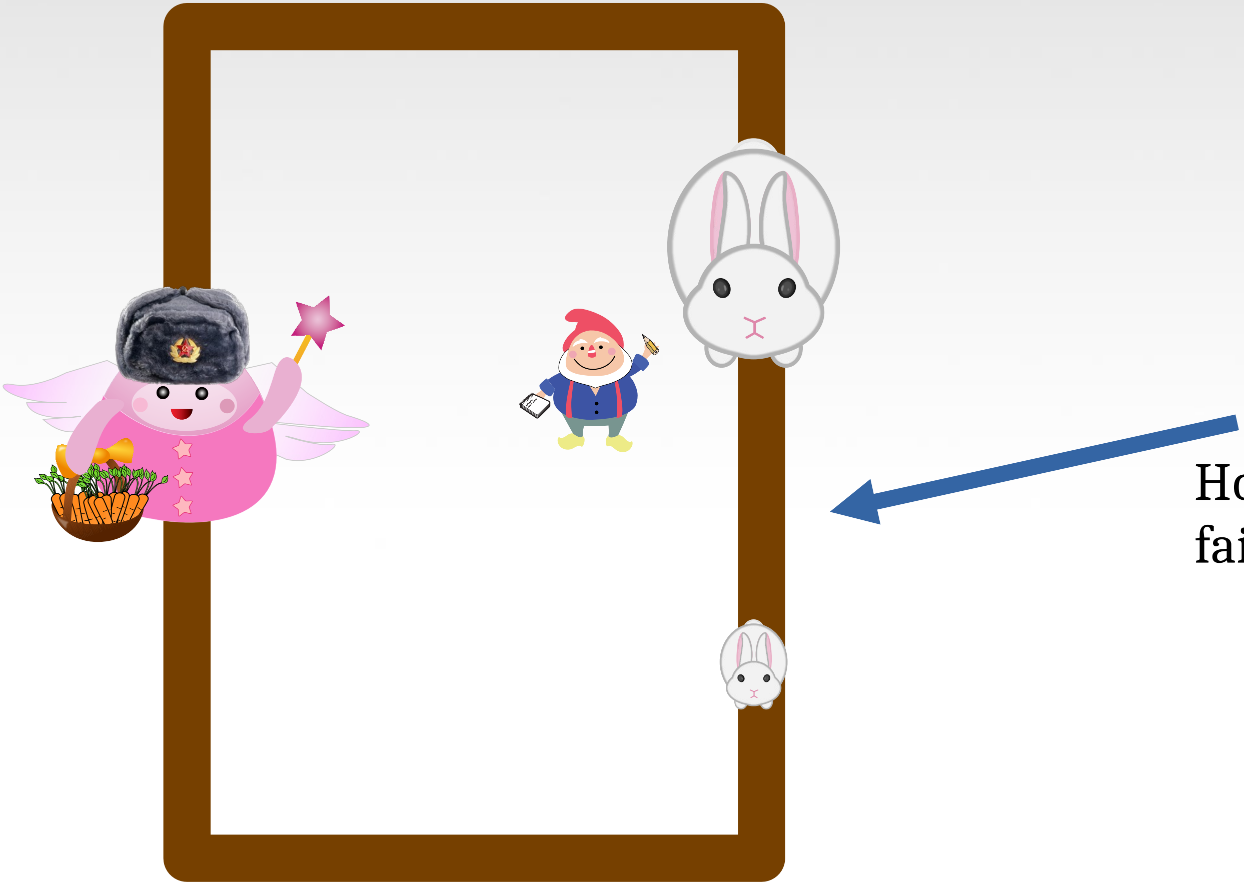
“From each according to their ability, to each according to their need.”

The Voltage Divider



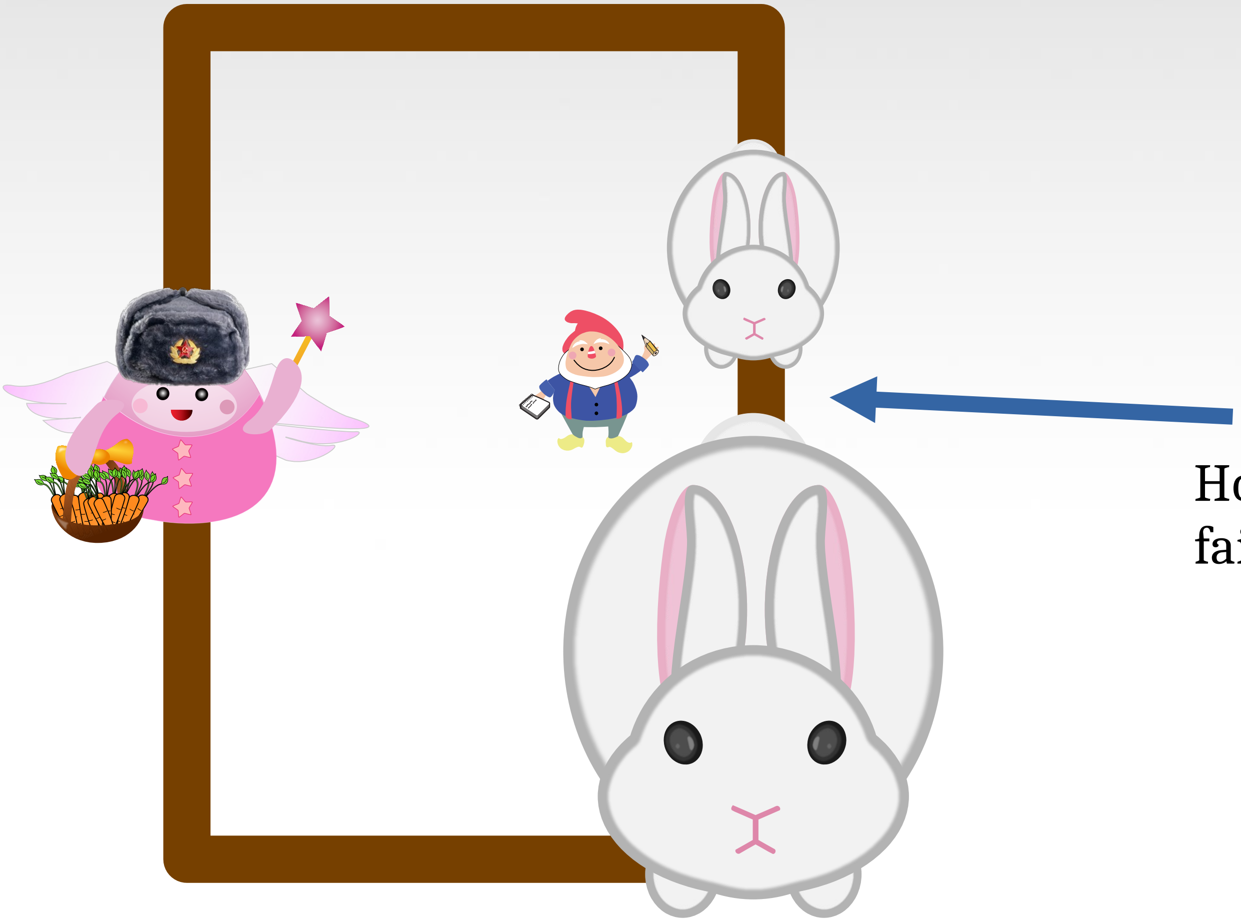
How many carrots does the fairy have left at this point?

The Voltage Divider



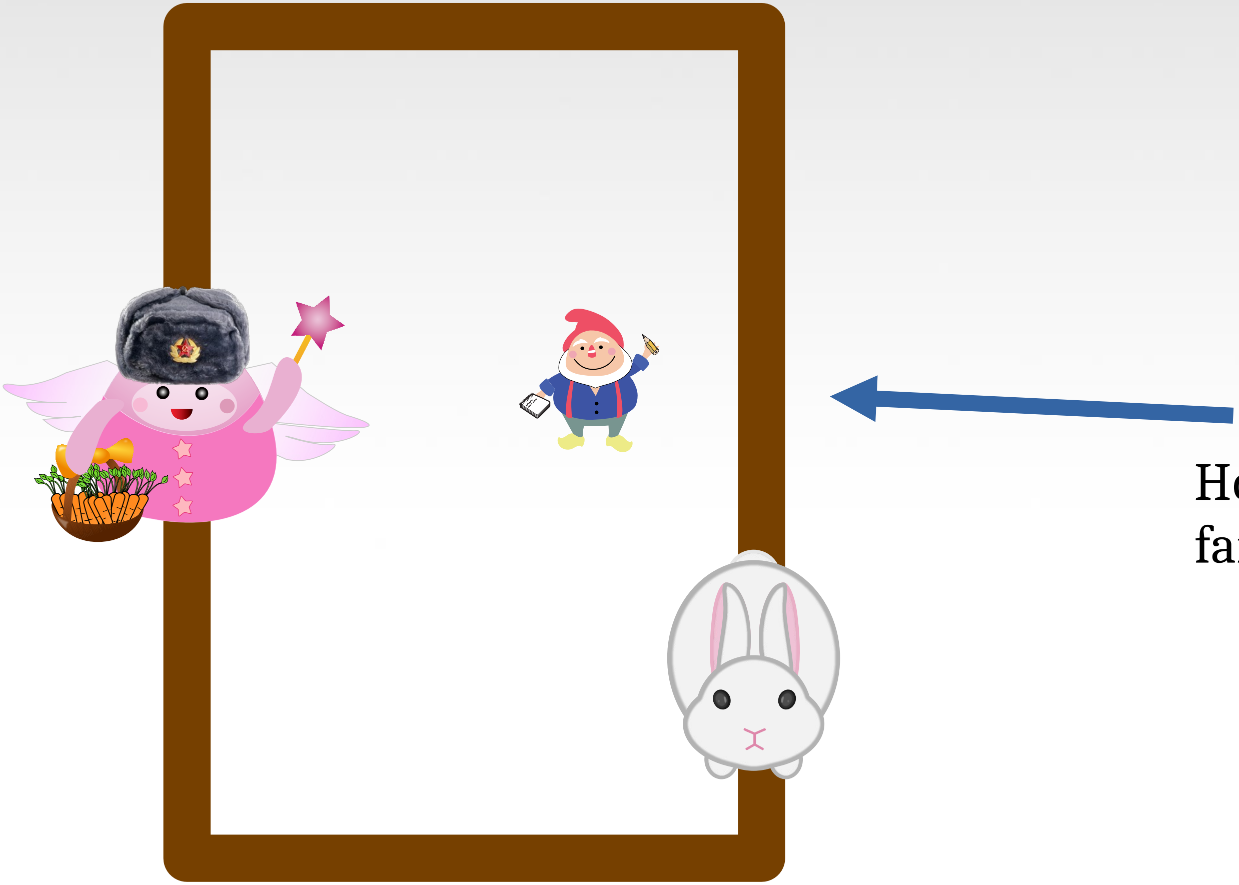
How many carrots does the fairy have left at this point?

The Voltage Divider



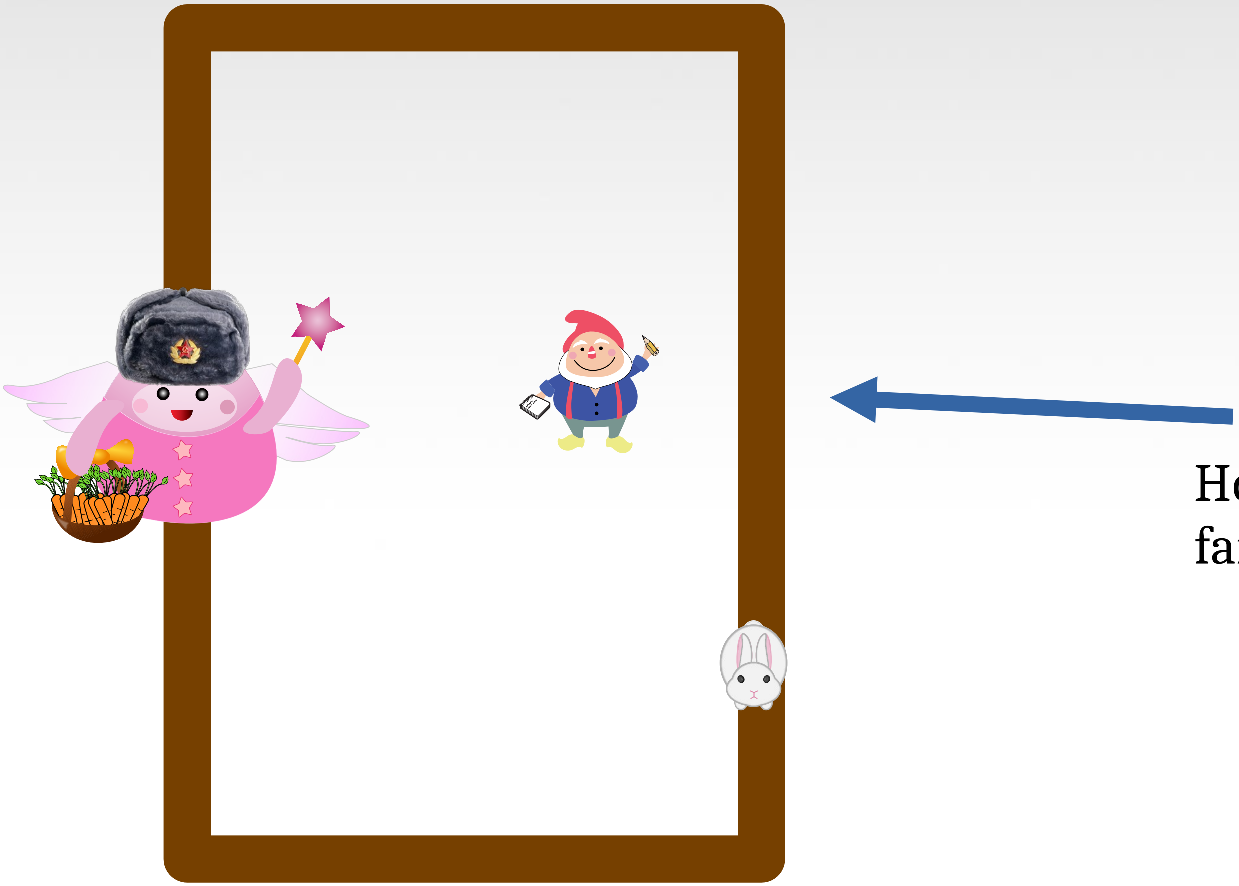
How many carrots does the fairy have left at this point?

The Voltage Divider



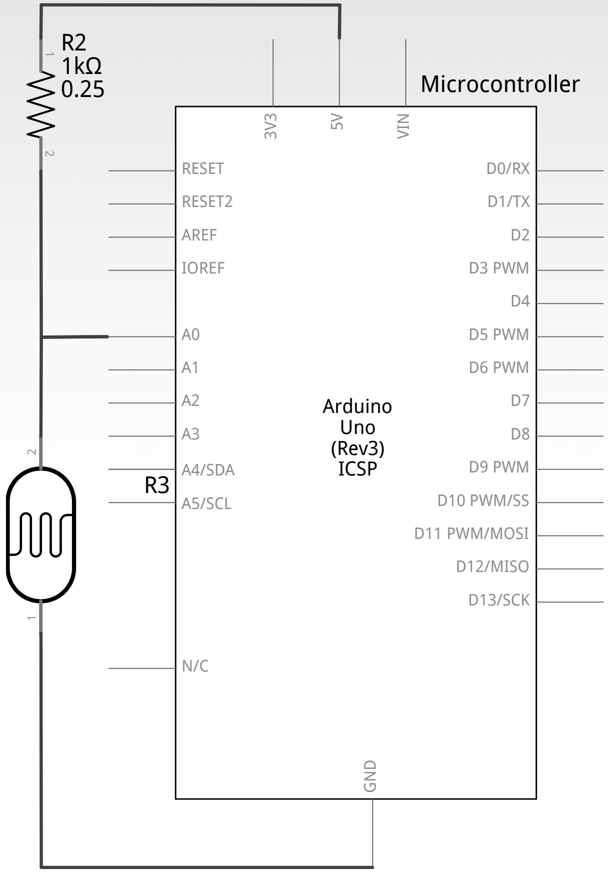
How many carrots does the fairy have left at this point?

The Voltage Divider



How many carrots does the fairy have left at this point?

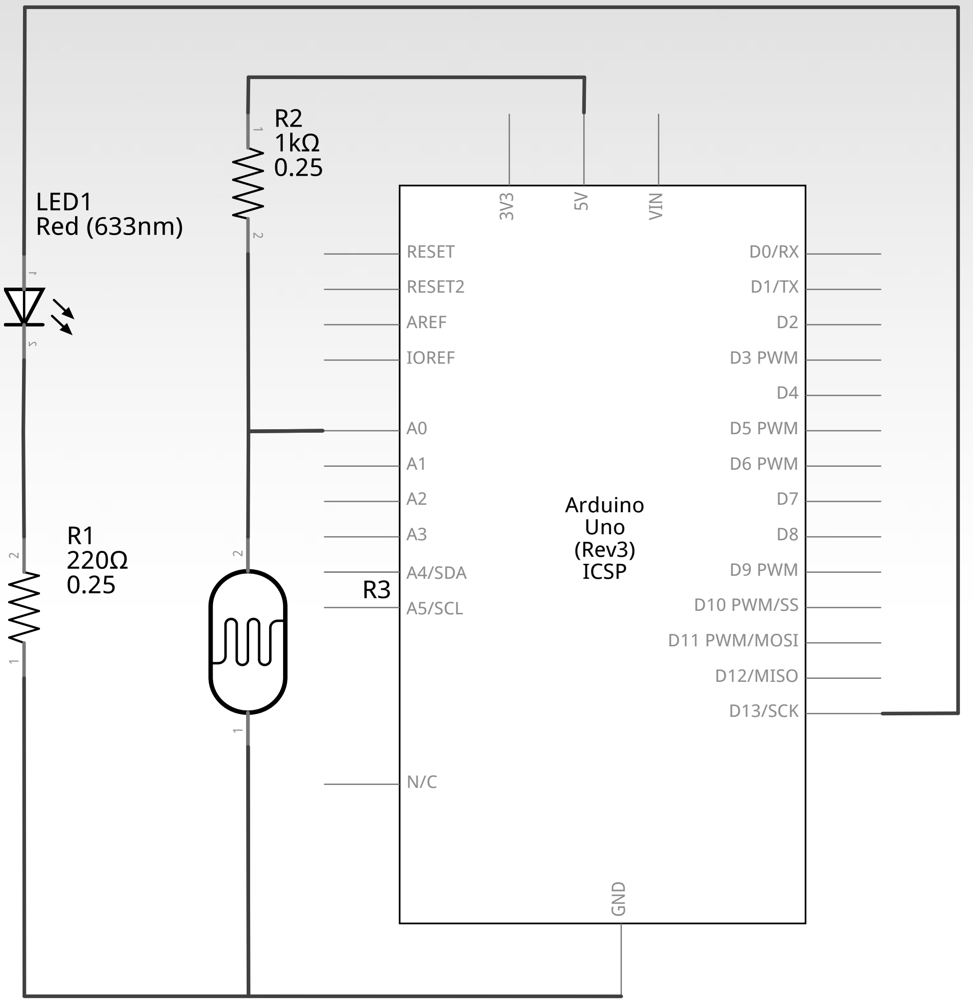
The Switch-Over



fritzing

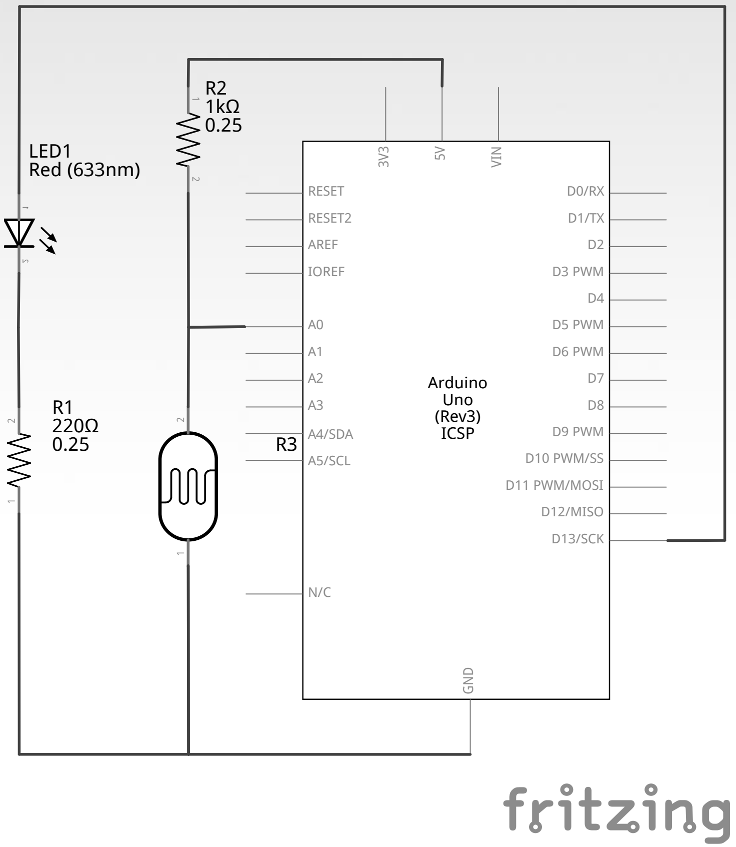
Project # 4 Night Light

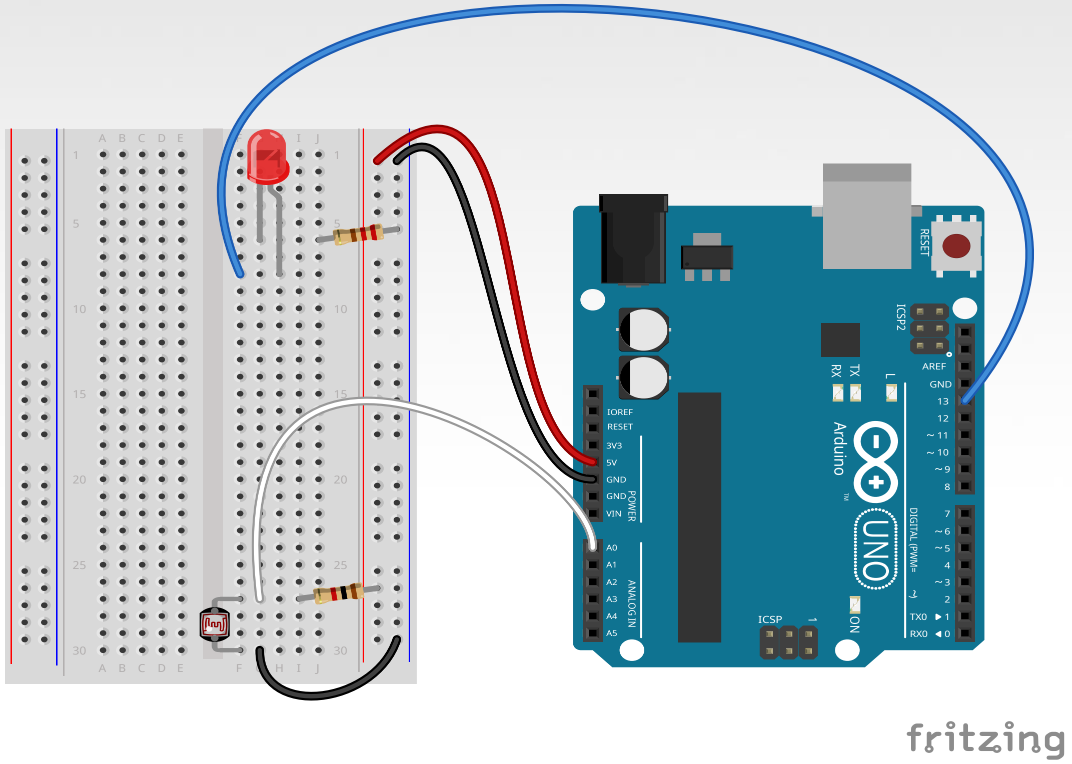
Schematic



fritzing

Project # 4 – Night Light
Wiring Diagram





Project # 4 – Night Light

```
int sensor_val = analogRead(pin);
```

pin – refers to the analog input pin (limited to pins A0, A1, A2, A3, A4 and A5)

Arduino uses a 10-bit analog-to-digital converter (ADC):

- This means that `sensor_val` is always a number between 0 and 1023
  - Minimum Carrots → 0
  - Maximum Carrots → 1023

Determining Analog Values

But the top bunny is always going to get some carrots, so we can’t really get to 1023

And we can’t really make our bottom bunny so small that it gets 0 carrots

So how do we determine the minimum and maximum numbers for our circuit?

- Do math

Determining Analog Values

But the top bunny is always going to get some carrots, so we can't really get to 1023

And we can't really make our bottom bunny so small that it gets 0 carrots

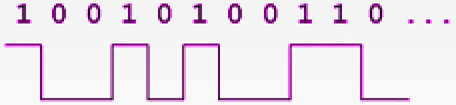
So how do we determine the minimum and maximum numbers for our circuit?

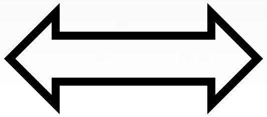
- ~~Do math~~
- Ask the microcontroller

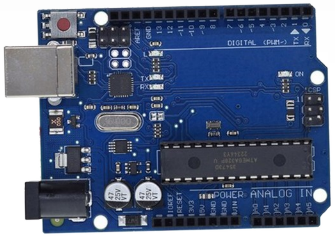
Using Serial Communication

Method used to transfer data between two devices.

Data passes between the computer and Arduino through the Universal **Serial** Bus (USB!) cable. Data is transmitted as zeros (‘0’) and ones (‘1’) sequentially.

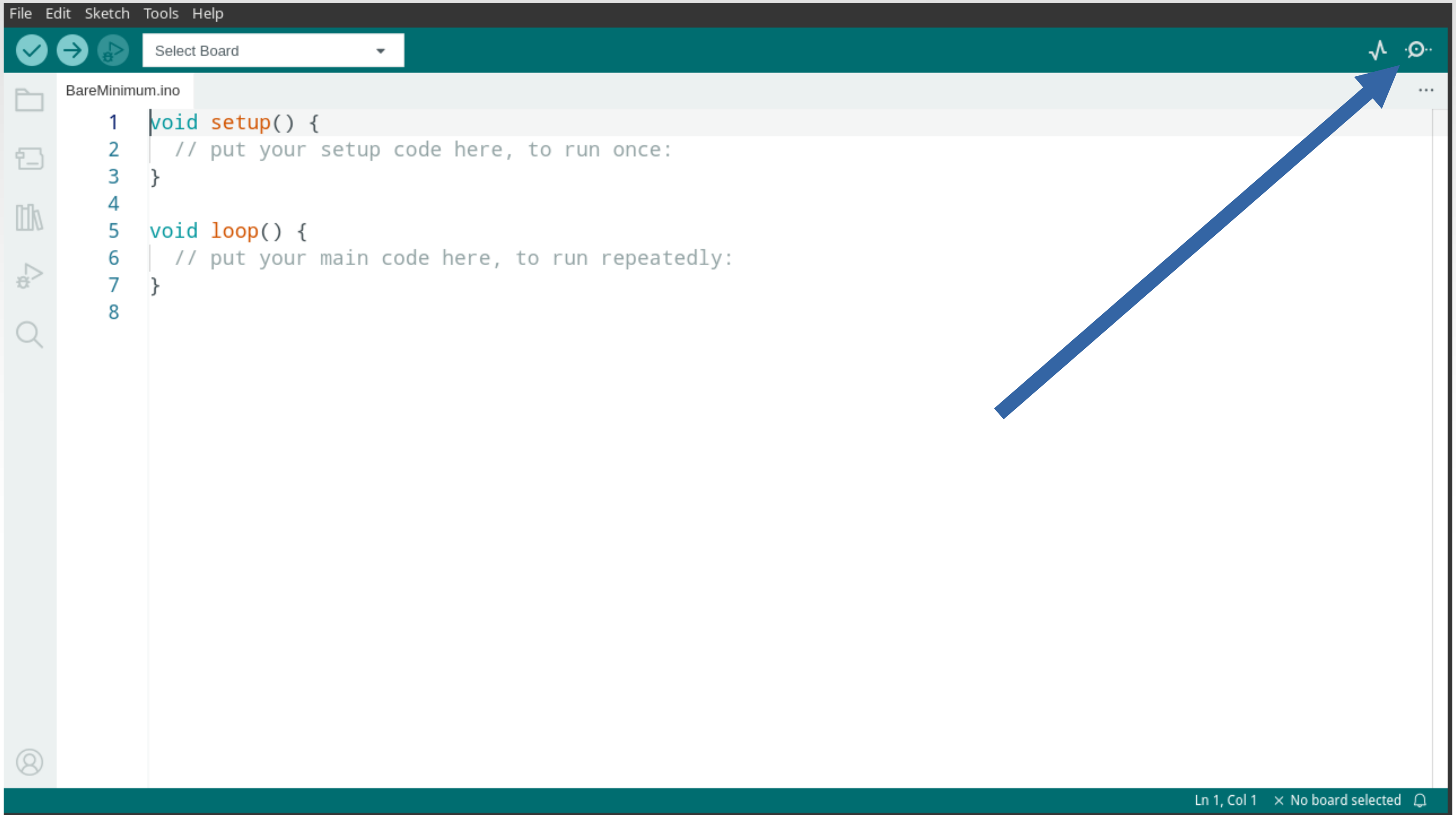






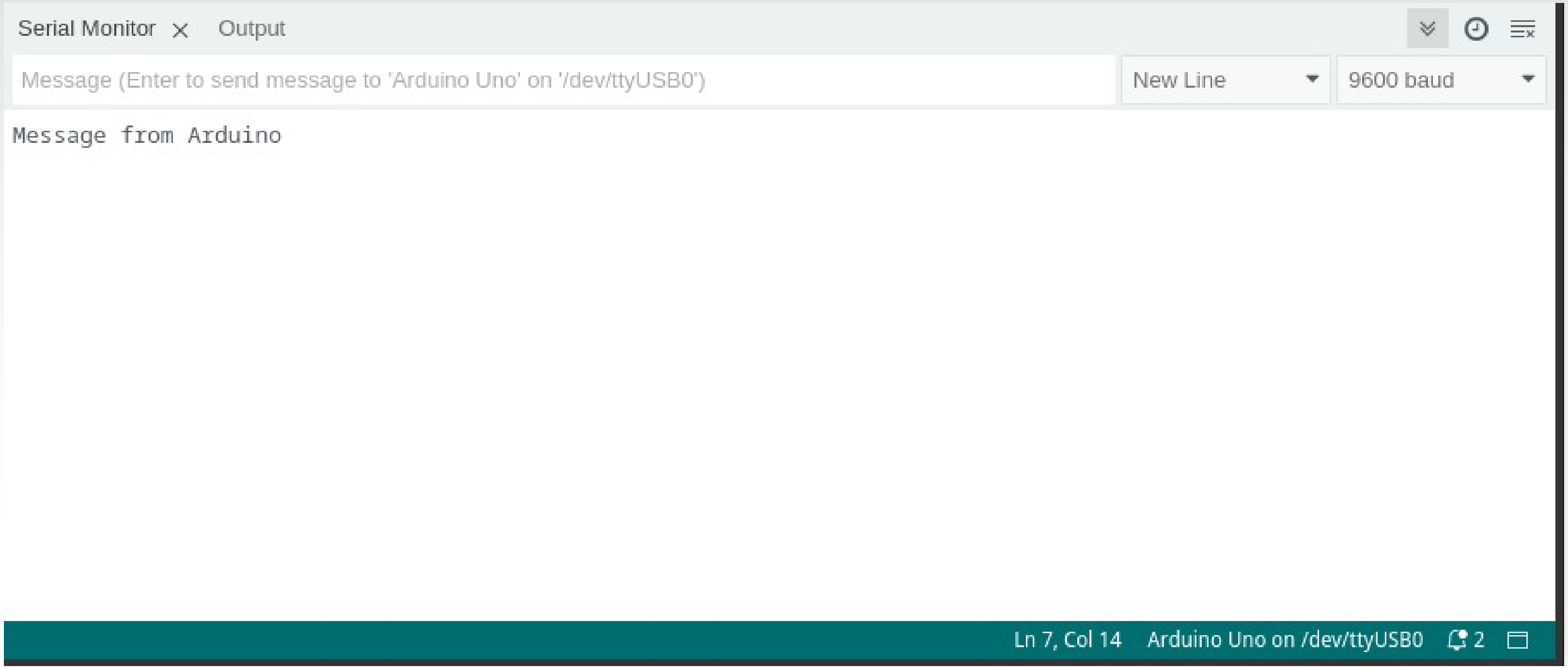
Arduino dedicates Digital I/O pin # 0 to receiving and Digital I/O pin #1 to transmit.

Displaying the Serial Monitor



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

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

```

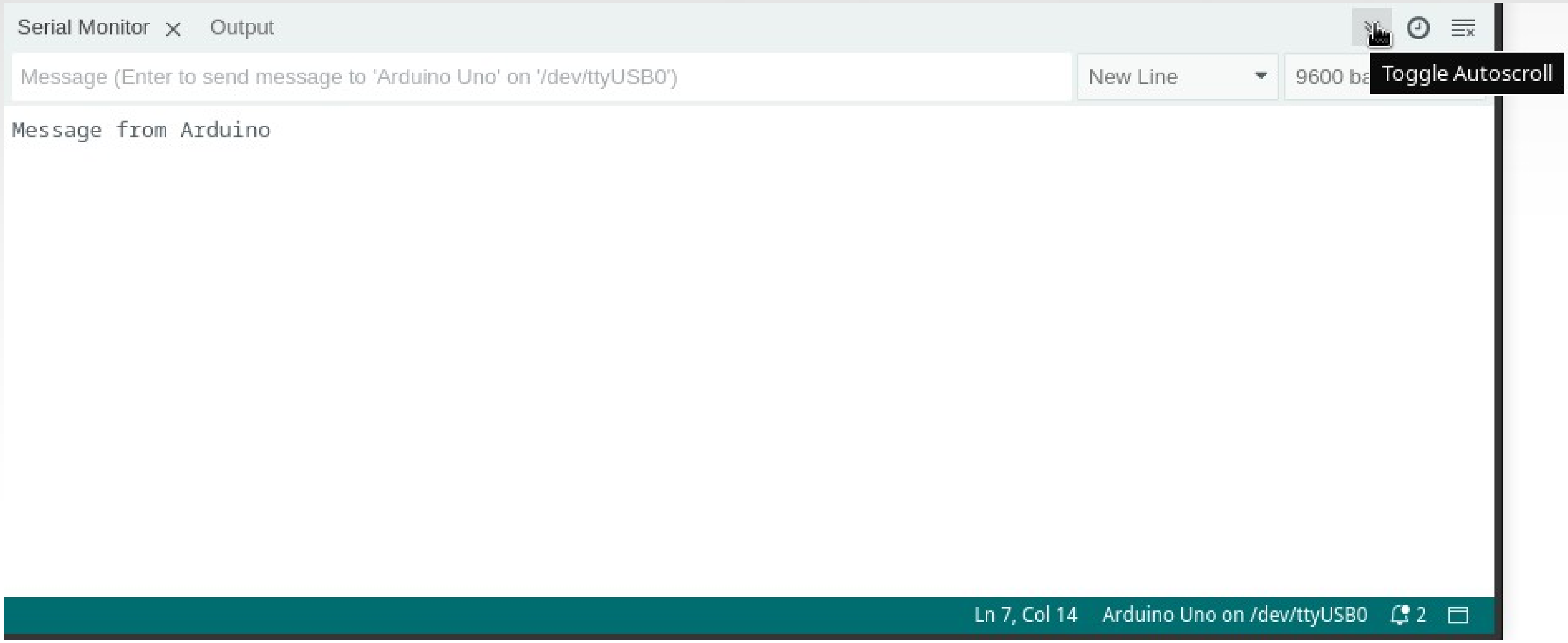
A Tour of the Serial Monitor



Serial Monitor ×
Output
Message (Enter to send message to 'Arduino Uno' on '/dev/ttyUSB0')
New Line
9600 baud
Ln 7, Col 14
Arduino Uno on /dev/ttyUSB0
2

```
Message from Arduino
```

A Tour of the Serial Monitor



Message (Enter to send message to 'Arduino Uno' on '/dev/ttyUSB0')

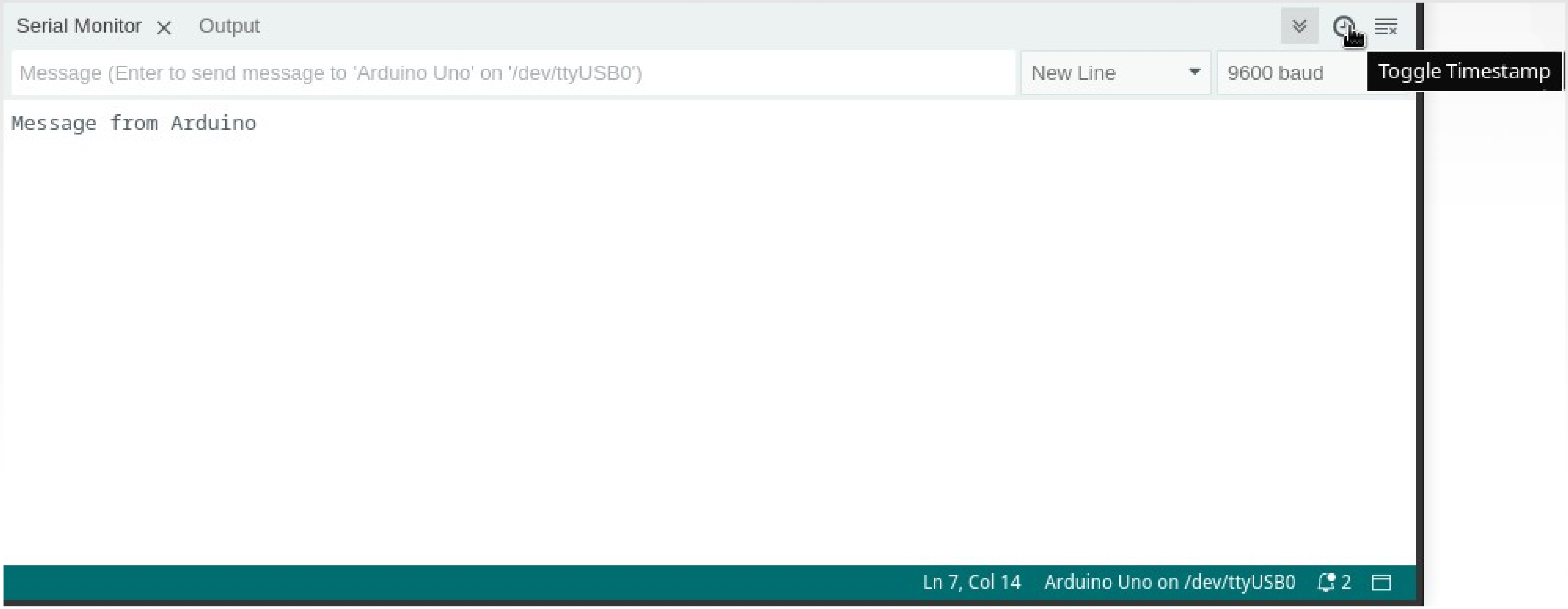
New Line

9600 ba

```
Message from Arduino
```

Ln 7, Col 14 Arduino Uno on /dev/ttyUSB0 2

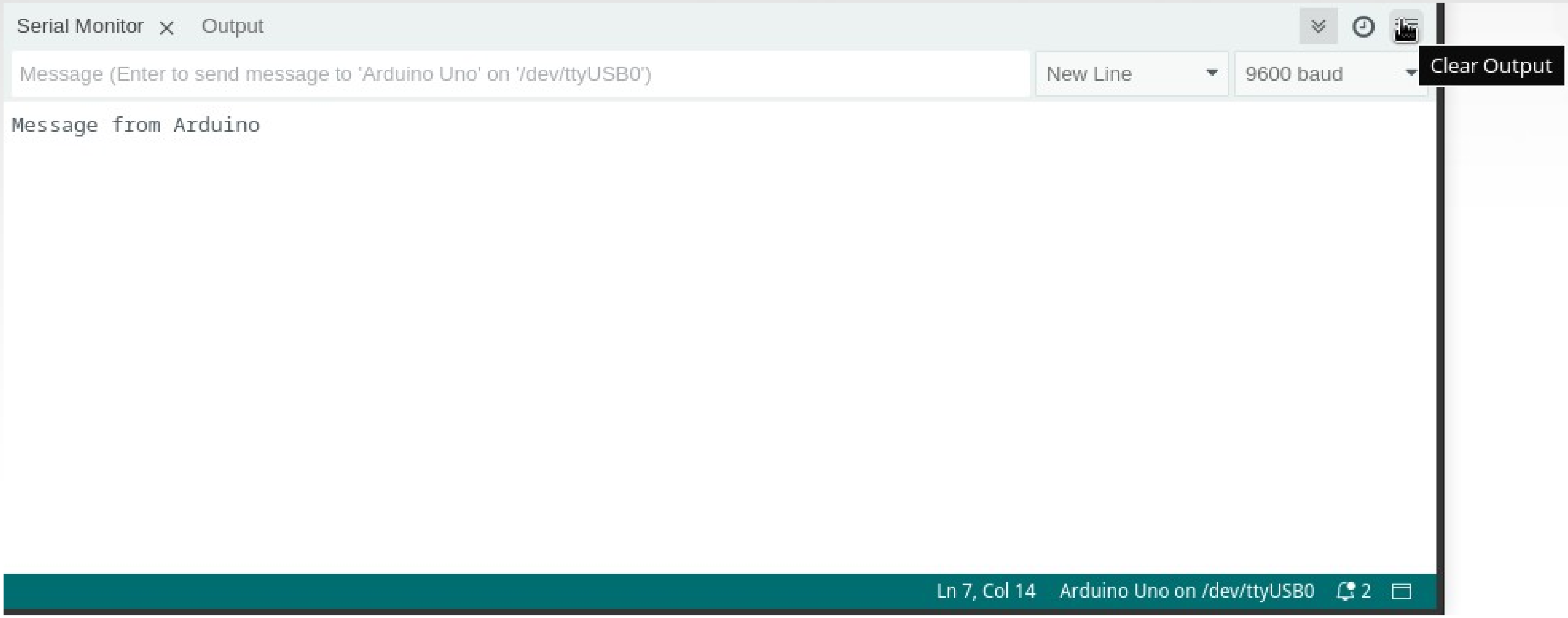
A Tour of the Serial Monitor


Serial Monitor
Output
Message (Enter to send message to 'Arduino Uno' on '/dev/ttyUSB0')
New Line
9600 baud
Toggle Timestamp

```
Message from Arduino
```

Ln 7, Col 14
Arduino Uno on /dev/ttyUSB0
2

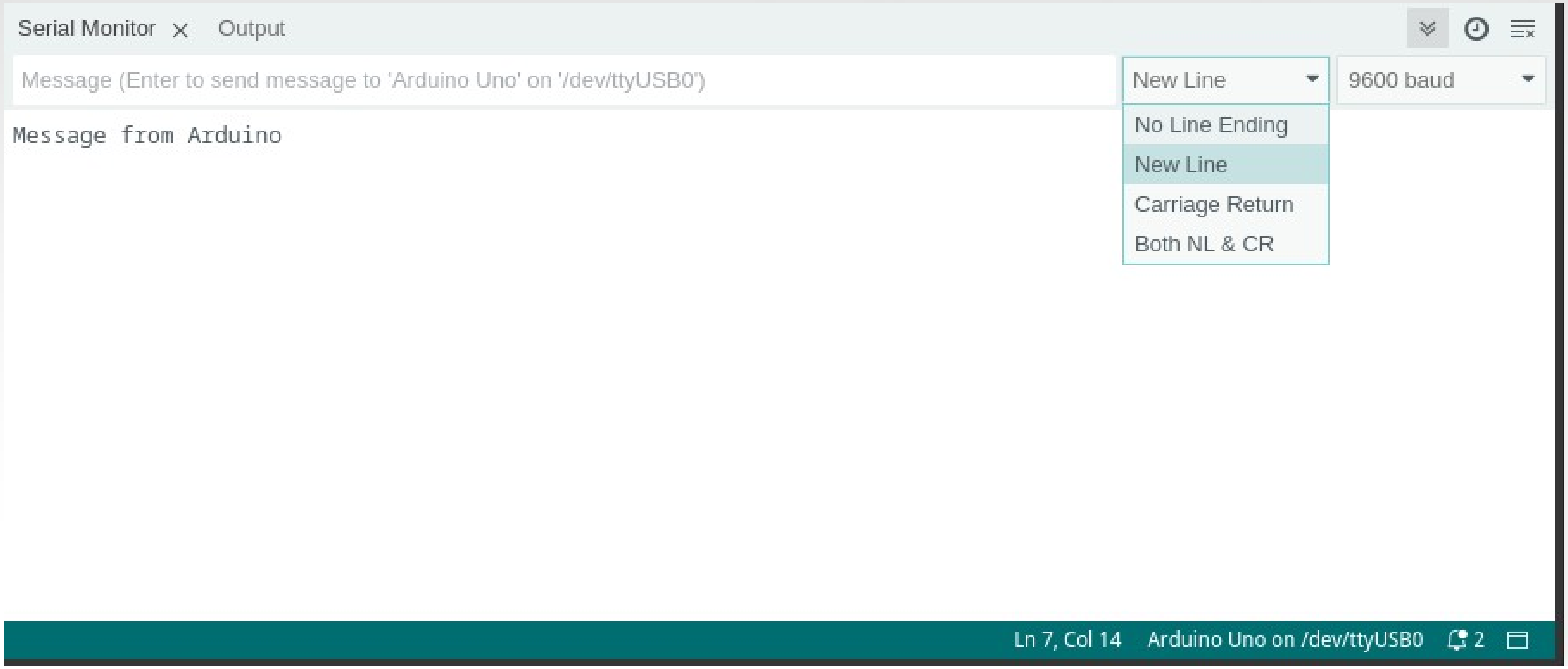
A Tour of the Serial Monitor



Message (Enter to send message to 'Arduino Uno' on '/dev/ttyUSB0')

```
Message from Arduino
```

A Tour of the Serial Monitor



Message (Enter to send message to 'Arduino Uno' on '/dev/ttyUSB0')

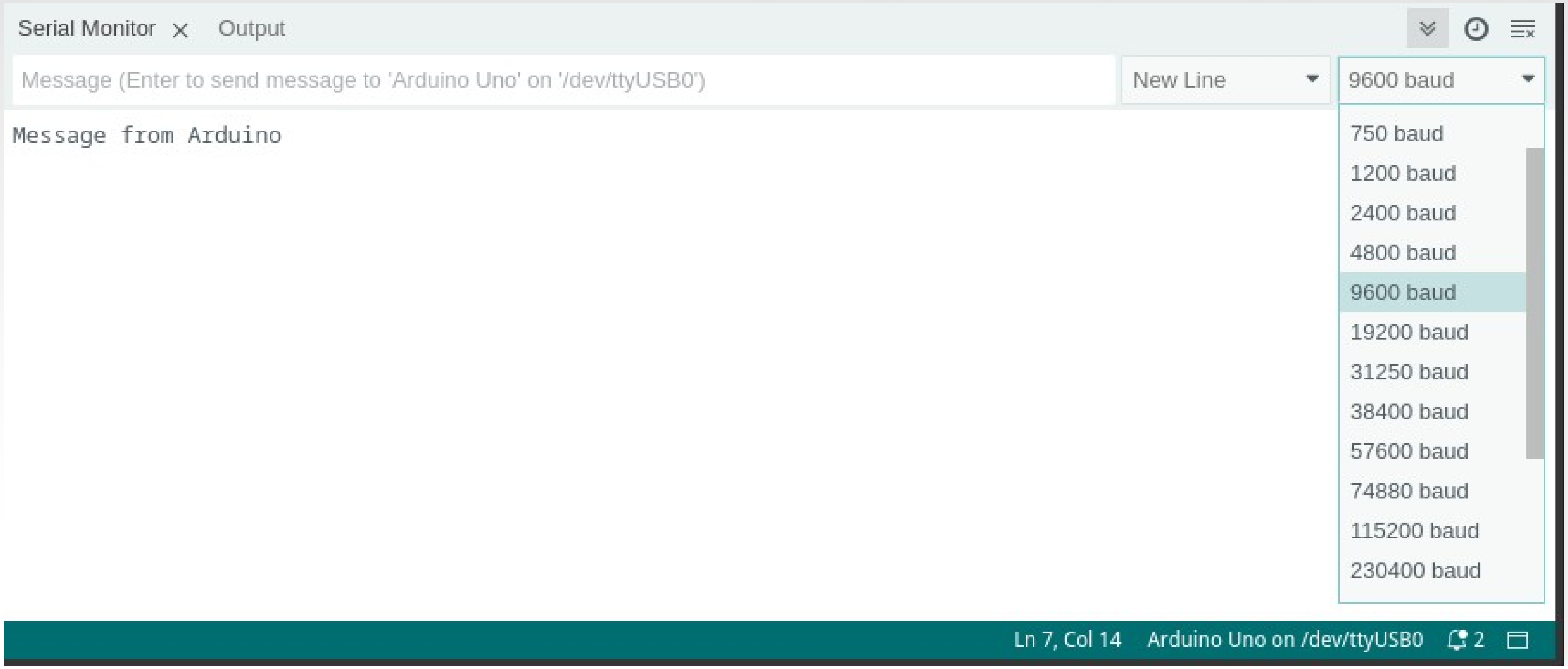
New Line

9600 baud

```
Message from Arduino
```

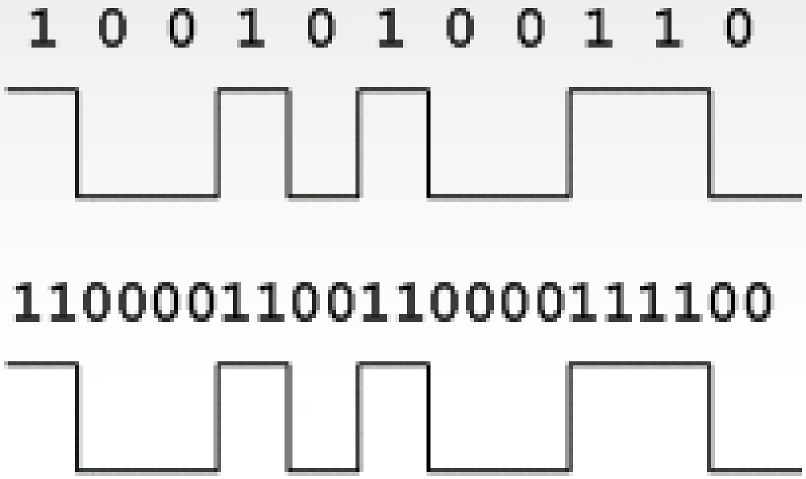
Ln 7, Col 14 Arduino Uno on /dev/ttyUSB0 2

A Tour of the Serial Monitor



```
Message from Arduino
```

Baud Rate



How do we know how many 0’s and 1’s there are?

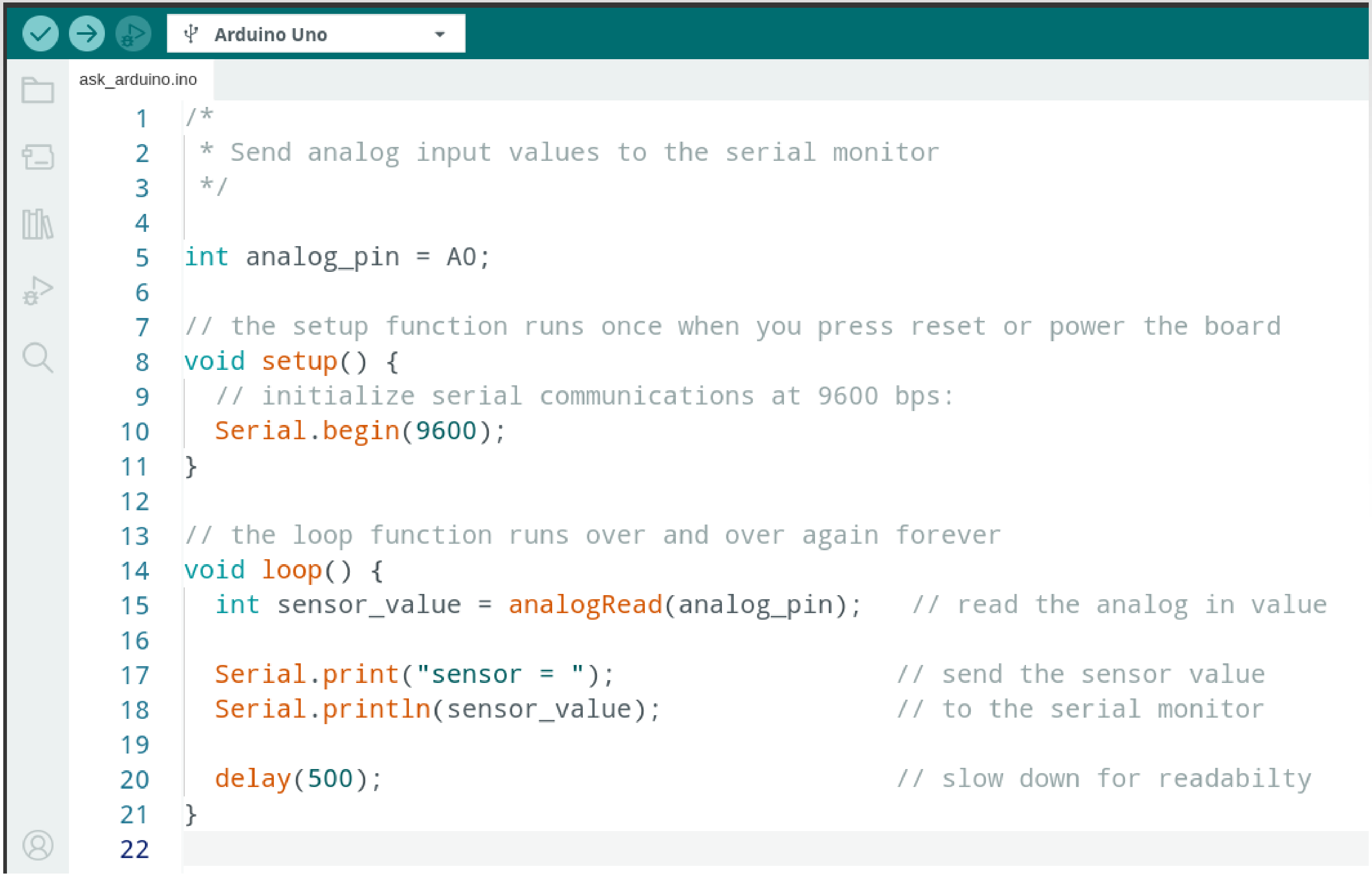
We tell how the computer and microcontroller how long a signal has to be high or low to count as a 1 or a 0 by setting the **baud rate**

For our purposes it doesn’t really matter *what* the baud rate is, it’s just important that the microcontroller and computer agree!

Serial Commands

```
Serial.begin(baud_rate);
  ex: Serial.begin(9600);

Serial.print(“Text”);
Serial.print(variable);

Serial.println(“Text”);
Serial.println(variable);
```

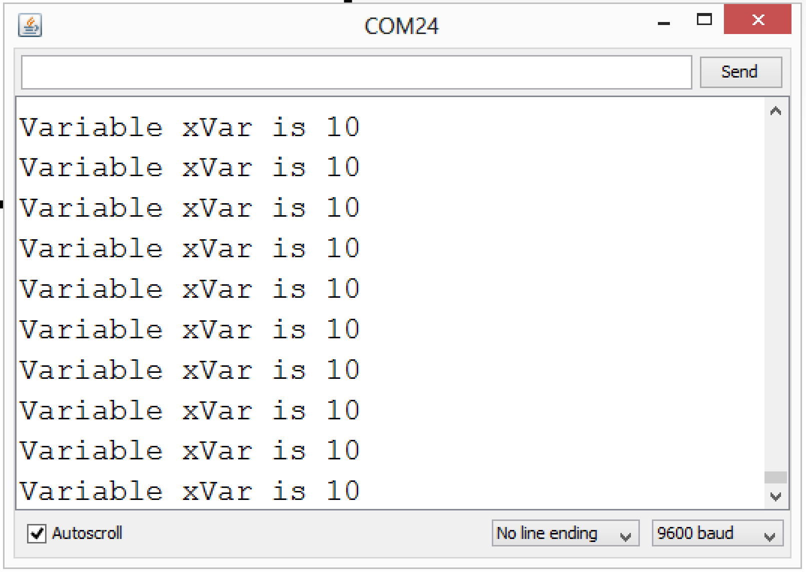
Ask Arduino - Code Review



```
/*
 * Send analog input values to the serial monitor
 */

int analog_pin = A0;

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}

// the loop function runs over and over again forever
void loop() {
  int sensor_value = analogRead(analog_pin);   // read the analog in value

  Serial.print("sensor = ");                  // send the sensor value
  Serial.println(sensor_value);               // to the serial monitor

  delay(500);                                 // slow down for readabilty
}
```

Serial Communication: Serial Debugging

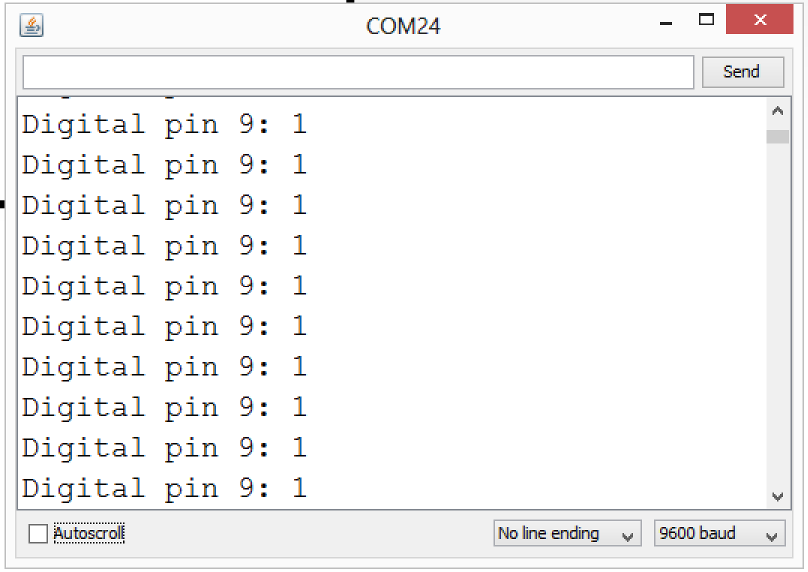
```
void loop()
{
   int xVar = 10;
   Serial.print ( “Variable xVar is “ ) ;
   Serial.println ( xVar ) ;
}
```



```
Variable xVar is 10
Variable xVar is 10
Variable xVar is 10
Variable xVar is 10
Variable xVar is 10
Variable xVar is 10
Variable xVar is 10
Variable xVar is 10
Variable xVar is 10
Variable xVar is 10
```

Serial Communication:
Serial Troubleshooting

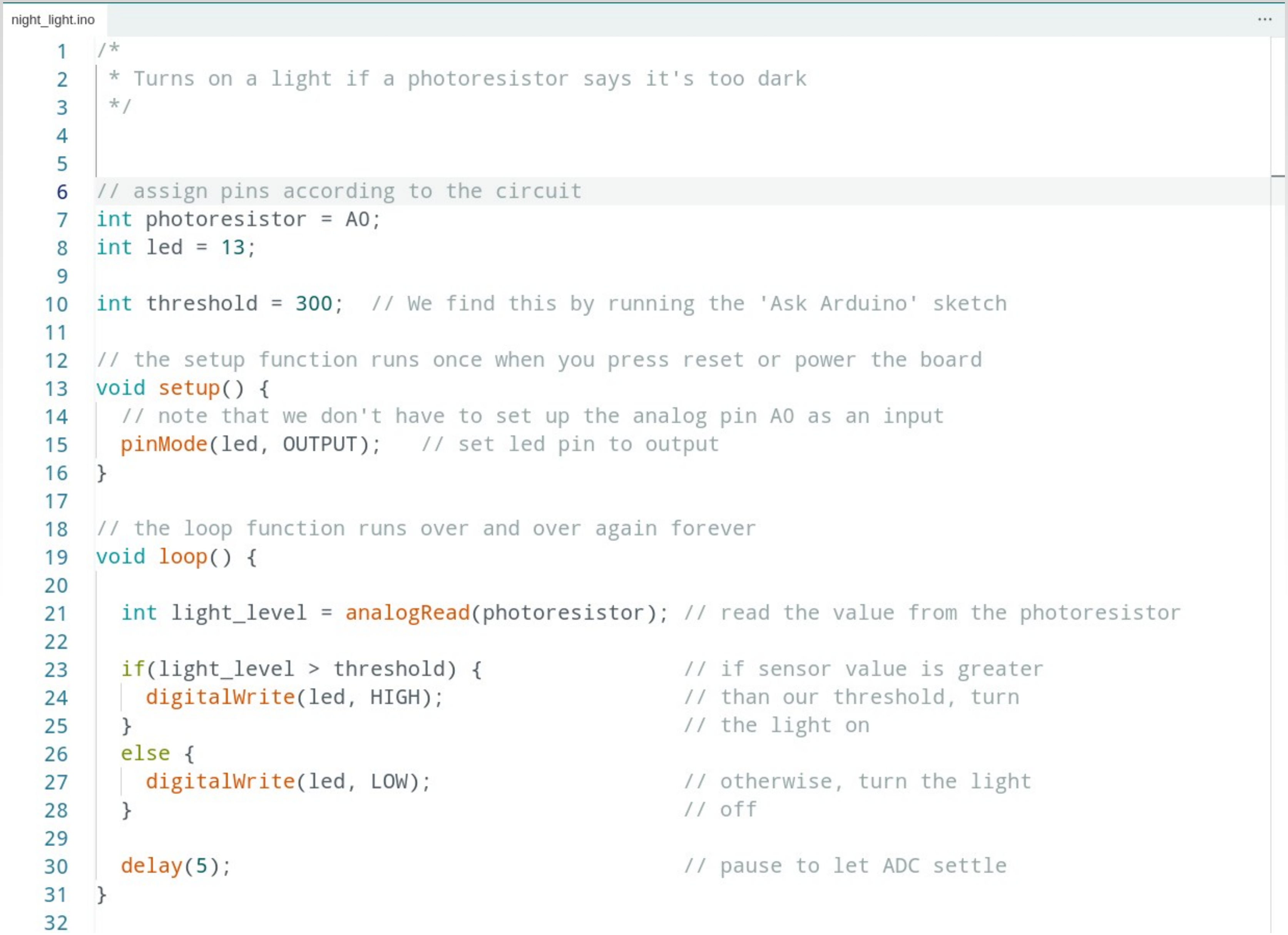
```
void loop ( )
{
  Serial.print ("Digital pin 9: ");
  Serial.println (digitalRead(9));
}
```



```
Digital pin 9: 1
Digital pin 9: 1
Digital pin 9: 1
Digital pin 9: 1
Digital pin 9: 1
Digital pin 9: 1
Digital pin 9: 1
Digital pin 9: 1
Digital pin 9: 1
Digital pin 9: 1
```

No line ending   9600 baud

Night Light - Code Review



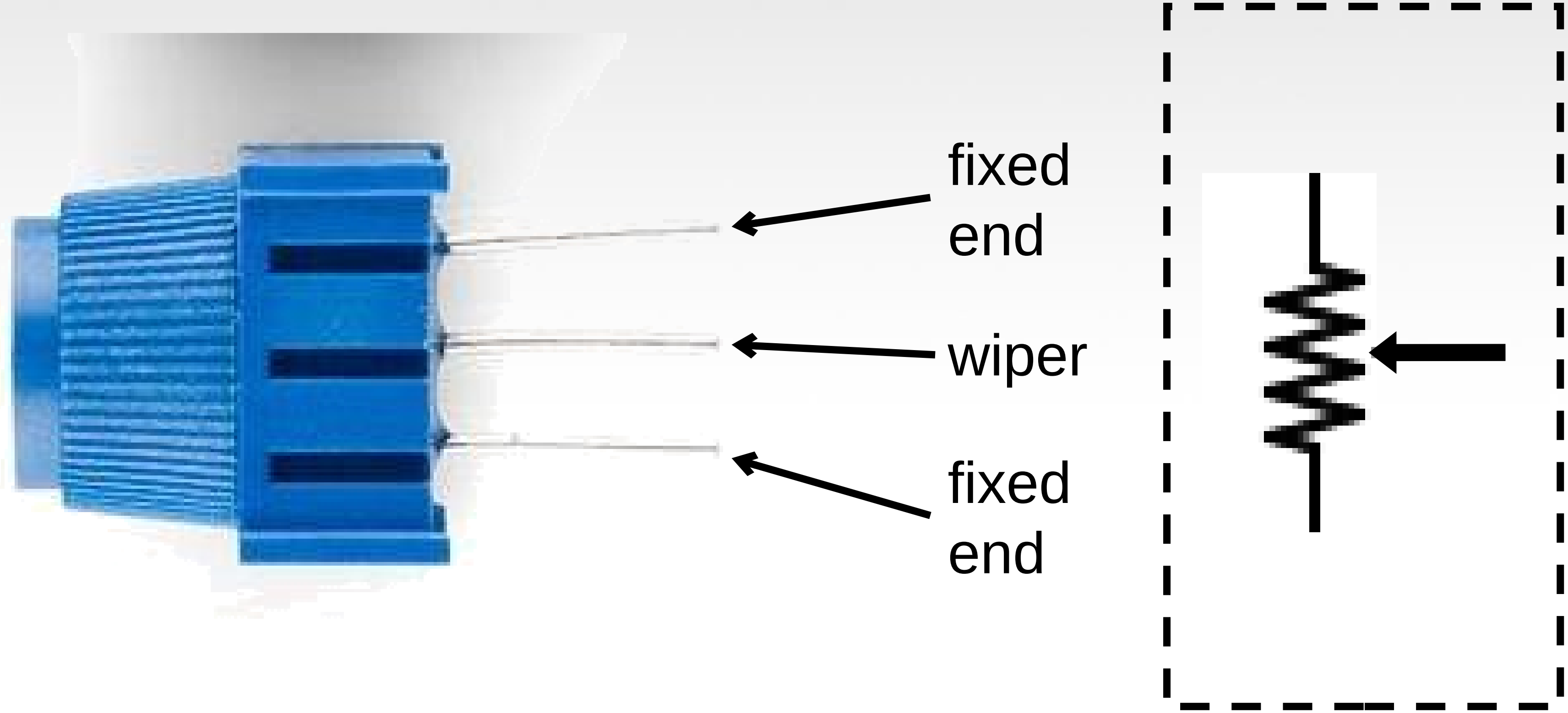
```
/*
 * Turns on a light if a photoresistor says it's too dark
 */


// assign pins according to the circuit
int photoresistor = A0;
int led = 13;

int threshold = 300;  // We find this by running the 'Ask Arduino' sketch

// the setup function runs once when you press reset or power the board
void setup() {
  // note that we don't have to set up the analog pin A0 as an input
  pinMode(led, OUTPUT);   // set led pin to output
}

// the loop function runs over and over again forever
void loop() {

  int light_level = analogRead(photoresistor); // read the value from the photoresistor

  if(light_level > threshold) {                // if sensor value is greater
    digitalWrite(led, HIGH);                   // than our threshold, turn
  }                                            // the light on
  else {
    digitalWrite(led, LOW);                    // otherwise, turn the light
  }                                            // off

  delay(5);                                    // pause to let ADC settle
}

```

Analog Sensors

Examples:

Sensors	Variables
Photoresistor	light_level
Microphone	sound_volume
Temp Sensor	temp
Flex Sensor	bend
Accelerometer	tilt, acceleration

Trimpot (Potentiometer)
Variable Resistor

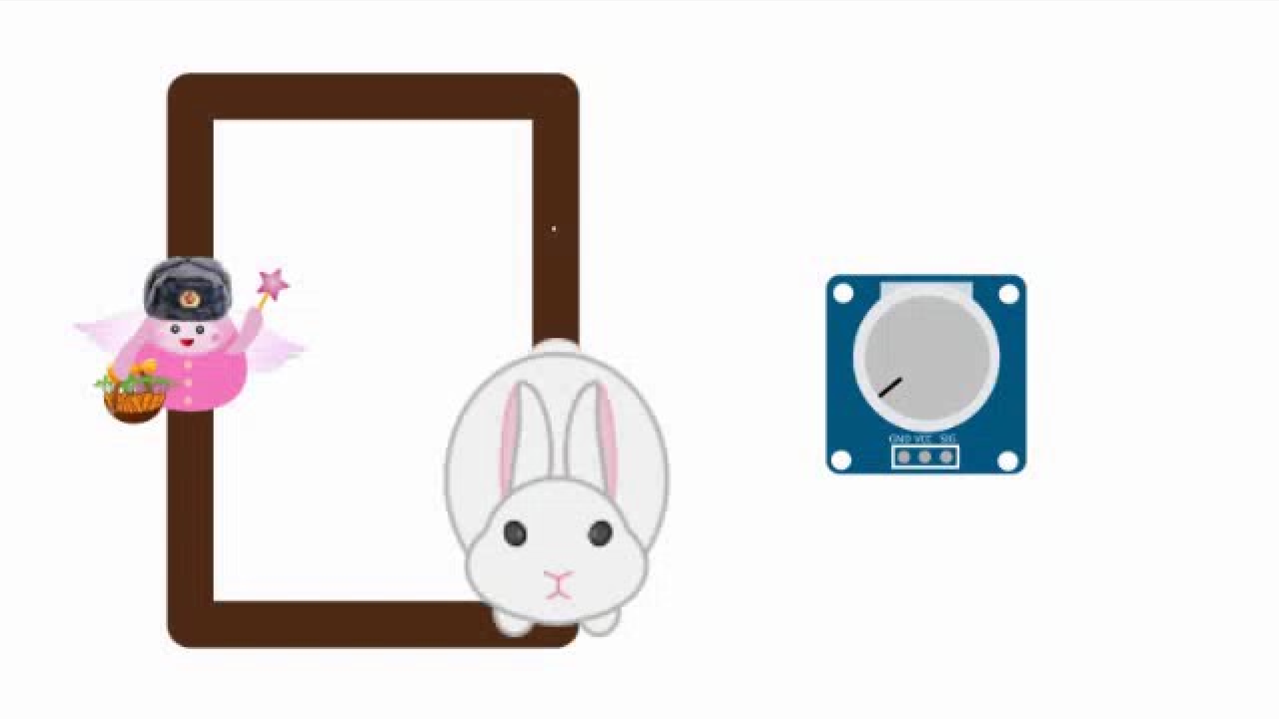


fixed
end

wiper

fixed
end

Trimpot (Potentiometer)
Variable Resistor



Trimpot (Potentiometer)
What does it mean?

Since we’re controlling the size of *both* bunnies:

- We can make either bunny so small it gets zero carrots
- We get the full range of 0-1023!
  - This makes potentiometers excellent controllers for physical computing “settings”
  - Can we think of any examples?

Trimpot (Potentiometer)
What does it mean?

Since we’re controlling the size of *both* bunnies:

- We can make either bunny so small it gets zero carrots
- We get the full range of 0-1023!
  - This makes potentiometers excellent controllers for physical computing “settings”
  - Can we think of any examples?
  - But what if the setting we want to control has less (or more) than 1023 values? Or if we want it not to start at 0?

Trimpot (Potentiometer) What does it mean?

Since we're controlling the size of *both* bunnies:

- We can make either bunny so small it gets zero carrots
- We get the full range of 0-1023!
  - This makes potentiometers excellent controllers for physical computing "settings"
  - Can we think of any examples?
  - But what if the setting we want to control has less (or more) than 1023 values? Or if we want it not to start at 0?
    - Do math

Trimpot (Potentiometer)
What does it mean?

Since we're controlling the size of *both* bunnies:

- We can make either bunny so small it gets zero carrots
- We get the full range of 0-1023!
  - This makes potentiometers excellent controllers for physical computing "settings"
  - Can we think of any examples?
  - But what if the setting we want to control has less (or more) than 1023 values? Or if we want it not to start at 0?
    - ~~Do math~~
    - Get the microcontroller to do it for us

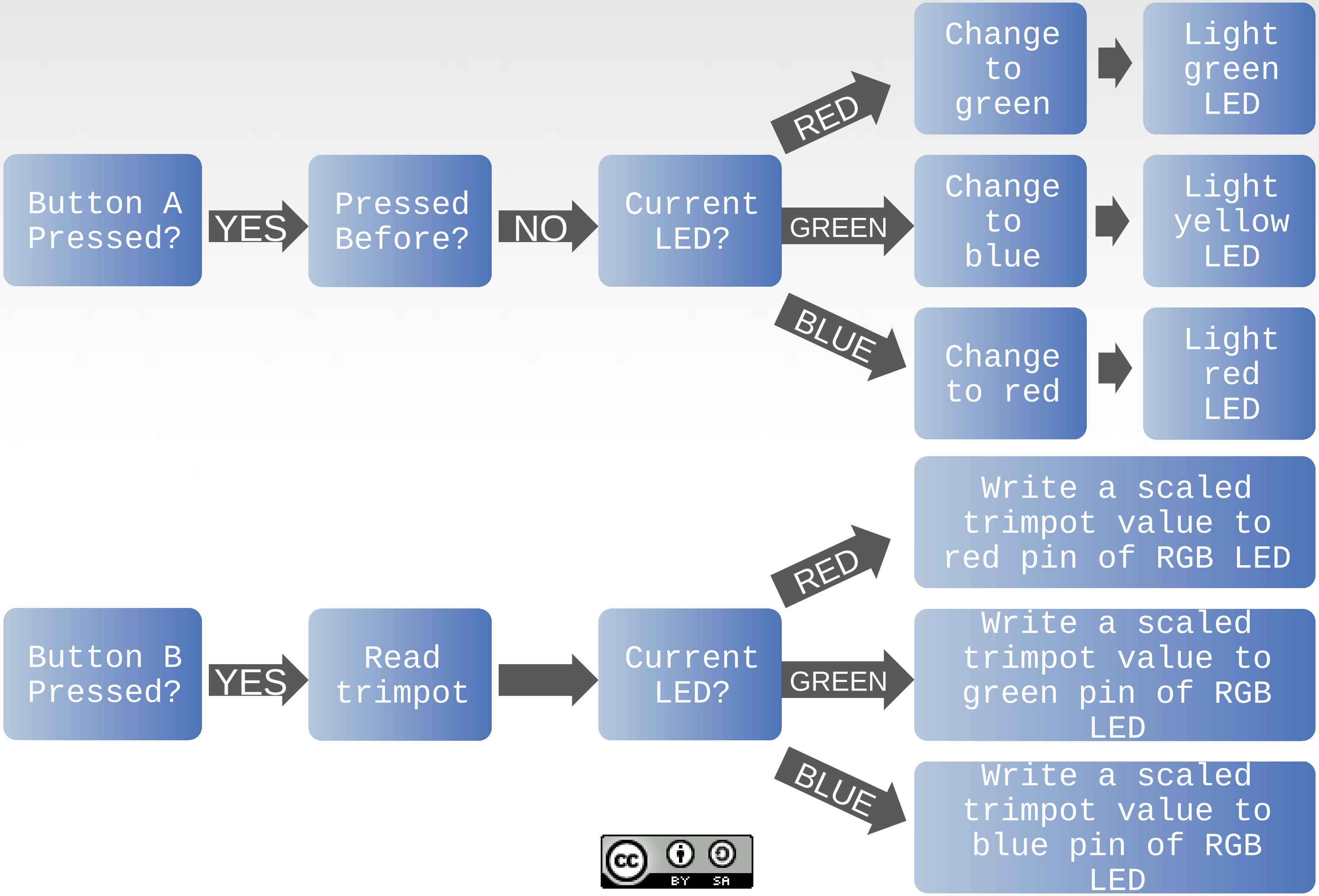
Map Command

```
int new_var = map(old_var, old_low, old_high, new_low, new_high);

    ex: offset_val = map(val, 0, 100, -50, 50);

    ex: reverse_val = map(val, 1, 100, 100, 1);

    ex: write_val = map(read_val, 0, 1023, 0, 255);
```

Project # 5 – Real Time Color Mixer

Tying it all together...

Psuedo-code – how should this work?



Button A Pressed? → YES → Pressed Before? → NO → Current LED?

RED → Change to green → Light green LED

GREEN → Change to blue → Light yellow LED

BLUE → Change to red → Light red LED

Button B Pressed? → YES → Read trimpot → Current LED?

RED → Write a scaled trimpot value to red pin of RGB LED

GREEN → Write a scaled trimpot value to green pin of RGB LED

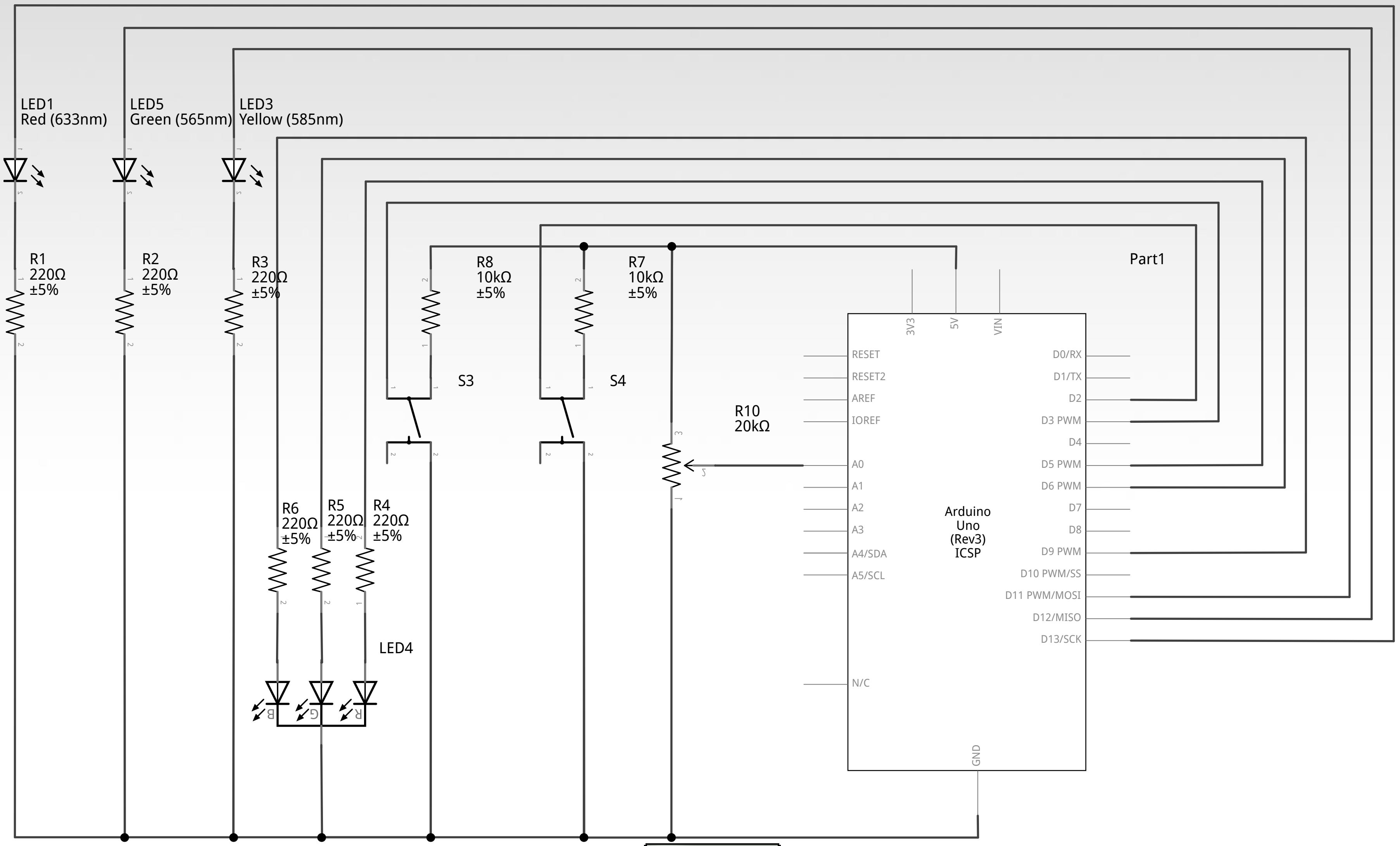
BLUE → Write a scaled trimpot value to blue pin of RGB LED

Project # 5 – Inputs and Outputs

Inputs	Outputs
a “Select” Button	Red Indicator
an “Adjust” Button	Green Indicator
Trimpot	Blue Indicator
	RGB LED

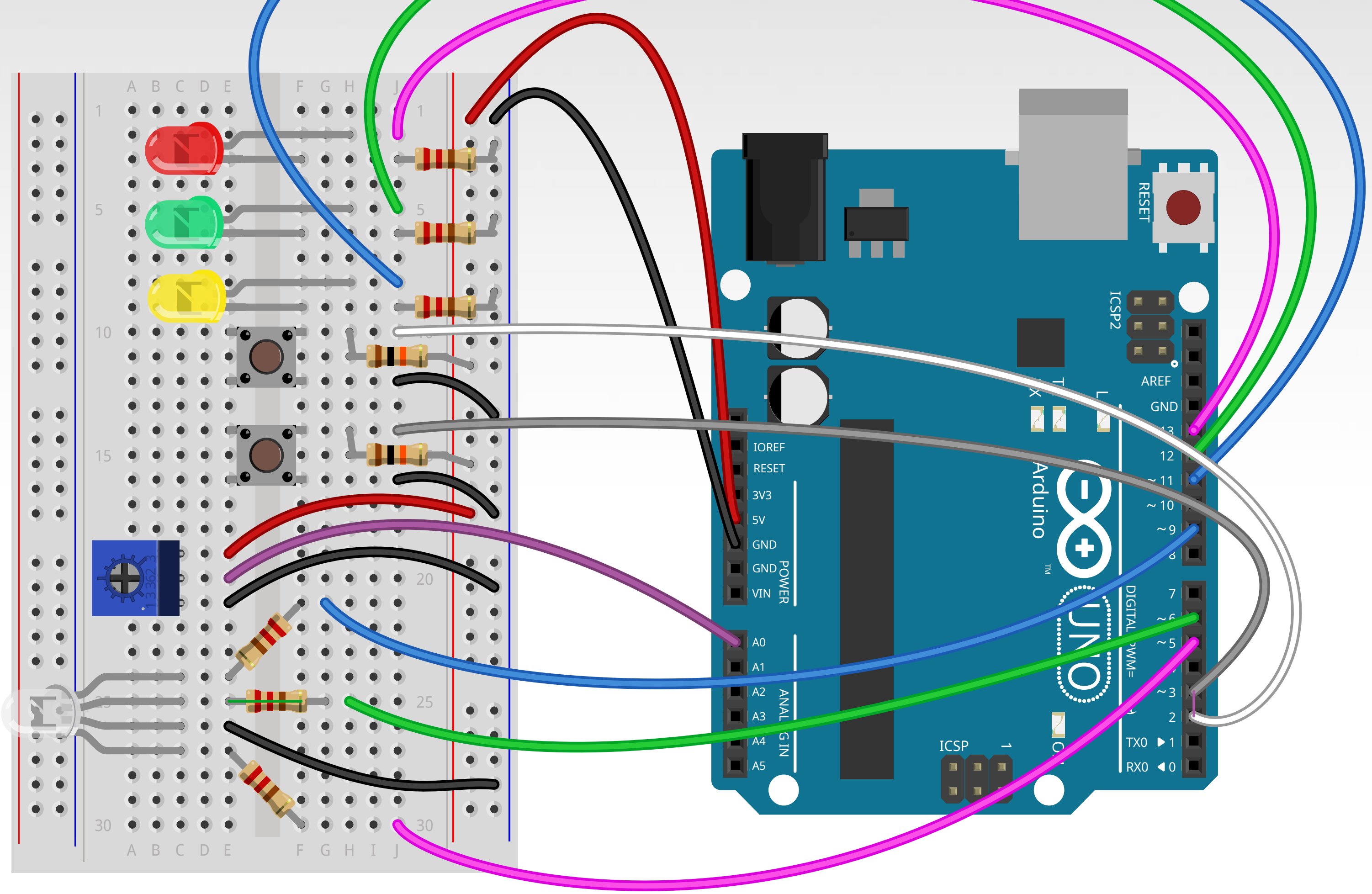
Project # 5 – Real Time Color Mixer

Schematic



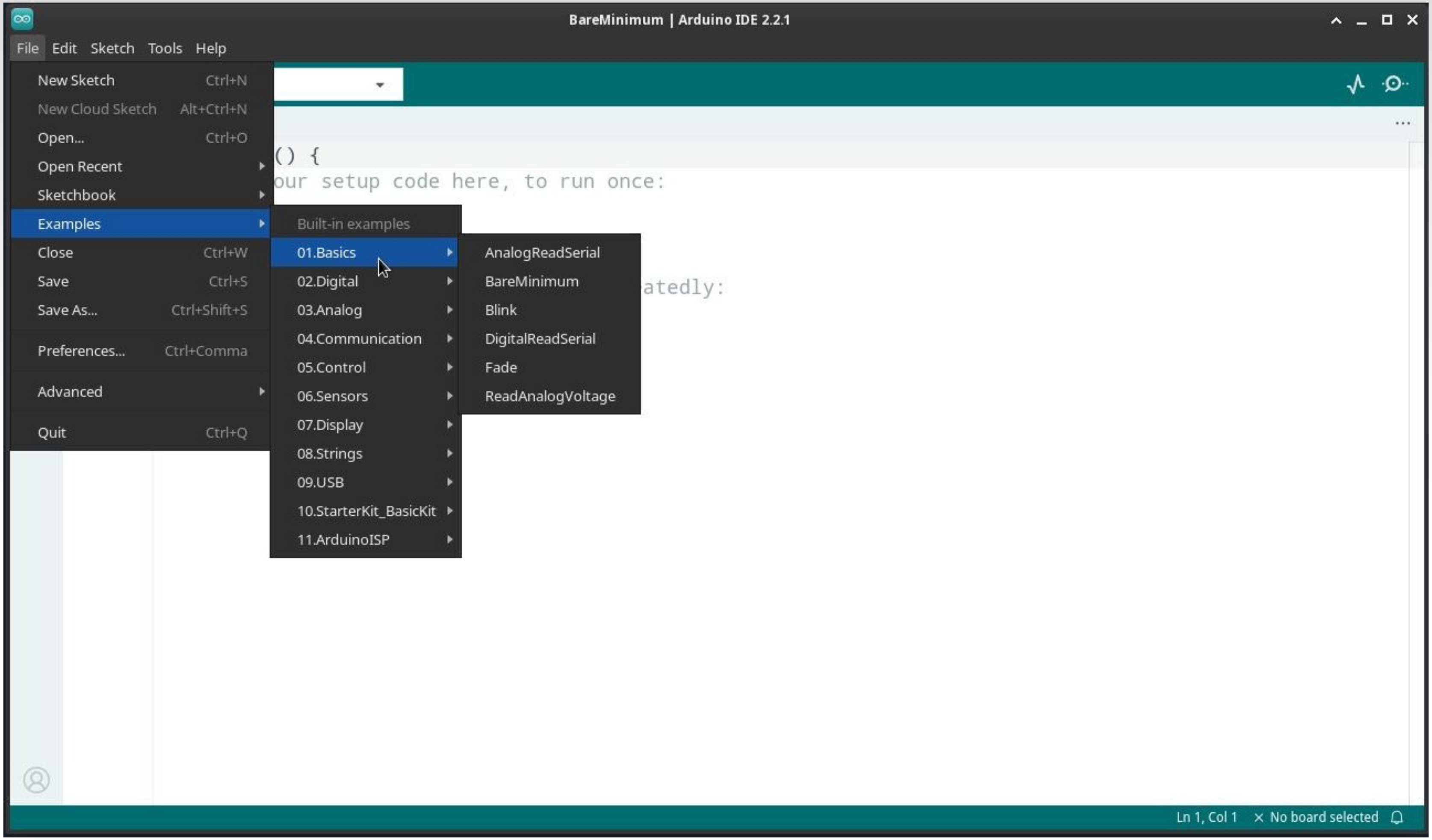
Project # 5 – Real Time Color Mixer

Wiring Diagram



fritzing

Where To Go From Here:



```
() {
our setup code here, to run once:

atedly:
```

Special Thanks:

sparkfun
ELECTRONICS

www.sparkfun.com
6175 Longbow Drive, Suite 200
Boulder, Colorado 80301

roto

SenseICs

SMARTY
PANTS
CONSULTING