物聯網裝置與平台
IoT Devices and Platforms

曾煜棋、吳昆儒

National Yang Ming Chiao Tung University

Ch 4, Arduino Analog - Summary

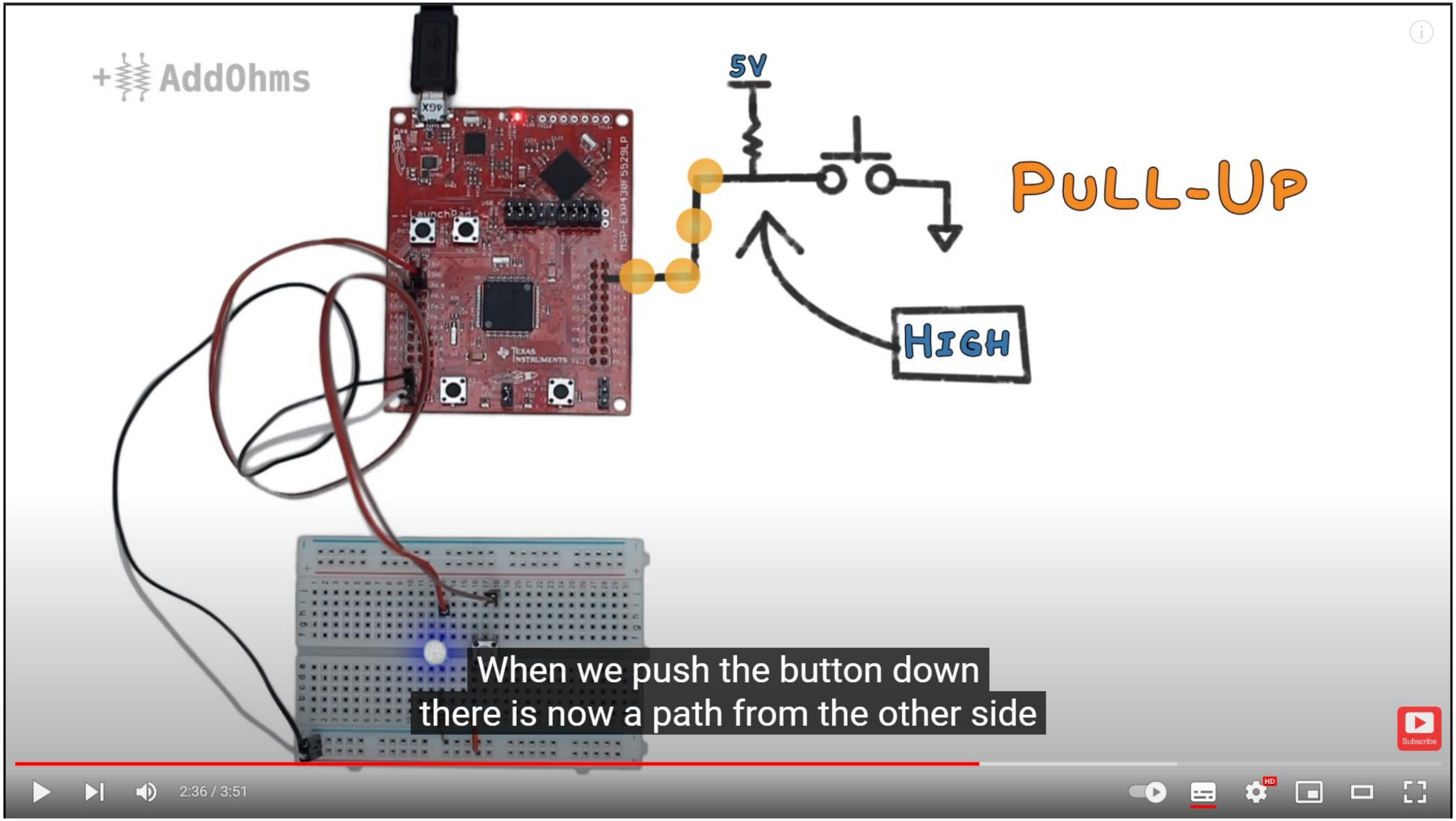
- Arduino IDE and how it interacts with the external world
  - Analog input/output
  - Calibration
  - Smoothing

- Understanding this course:
  - Discussion & quiz
  - Interact with TA
    - For remote-access, use discord to interact with TA

Labs (Last week)

1. **BlinkWithoutDelay**: blinking an LED without using the delay() function.
2. **StateChangeDetection**: counting the number of button pushes.
3. **Debounce**: read a pushbutton, filtering noise.
4. **DigitalInputPullup**: Demonstrates the use of INPUT_PULLUP with pinMode().

Last week



When we push the button down
there is now a path from the other side

Labs (This week)

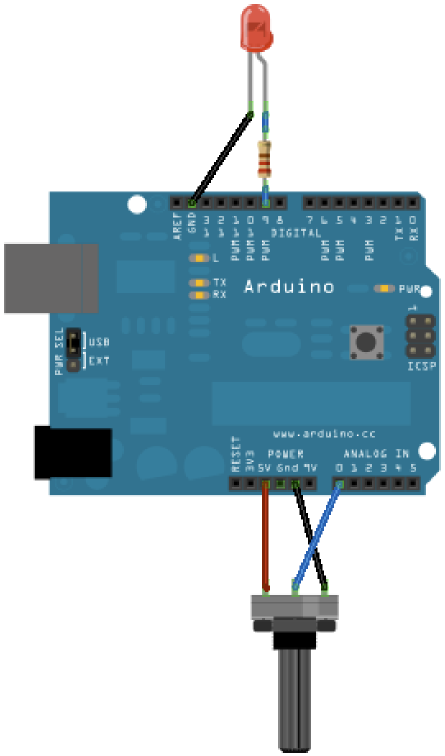
1. **AnalogInOutSerial**: Read an analog input pin, map the result, and then use that data to dim or brighten an LED.
2. **AnalogInput**: Use a potentiometer to control the blinking of an LED.
3. **Calibration**: Define a maximum and a minimum for expected analog sensor values.
4. **Fading**: Use an analog output (PWM pin) to fade an LED.
5. **Smoothing**: Smooth multiple readings of an analog input.

Lab1. AnalogInOutSerial:

Read an analog input pin, map the result, and then use that data to dim or brighten an LED.

Lab 1. AnalogInOutSerial

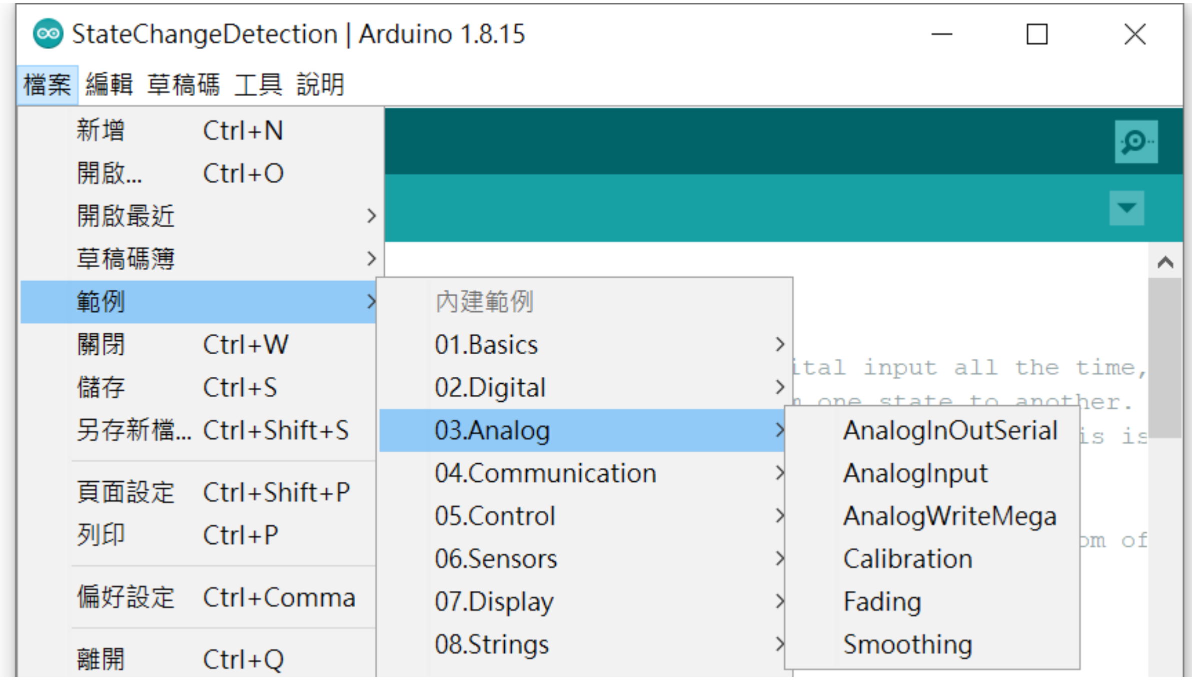
- Goal: read an analog input pin, map the result to a range from 0 to 255, and then use that result to set the PWM of an output pin to dim or brighten an LED.
- Hardware Required
  - Arduino Board
  - Potentiometer
  - LED
  - 220 ohm resistor



Lab 1. AnalogInOutSerial

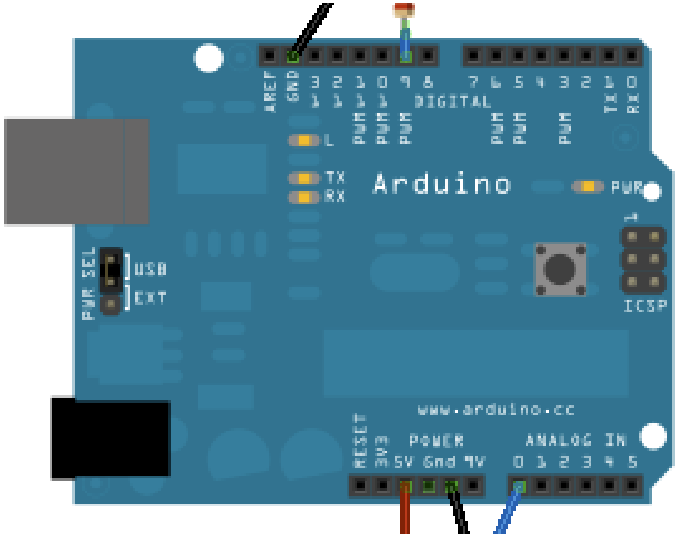
Open--->File--->Examples---> Analog--->AnalogInOutSerial

Arduino IDE



Built-in Sample Code:

```
// These constants won't change. They're used to give names to the pins used:
const int analogInPin = A0;  // Analog input pin that the potentiometer is attached to
const int analogOutPin = 9; // Analog output pin that the LED is attached to

int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}
```



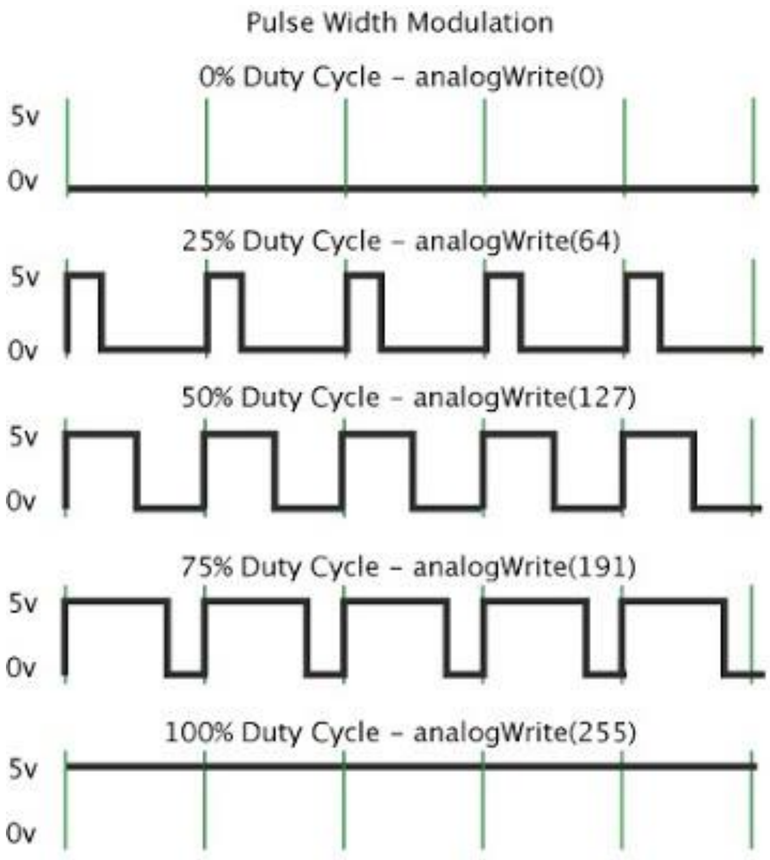
```
void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 0, 1023, 0, 255);

  // change the analog out value:
  analogWrite(analogOutPin, outputValue);

  // print the results to the Serial Monitor:
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);

  // wait 2 milliseconds before the next loop for the analog-to-digital
  // converter to settle after the last reading:
  delay(2);
}
```

Lab 1. Syntax

- Syntax
  - map(value, fromLow, fromHigh, toLow, toHigh)
- Description
  - Re-maps a number from one range to another.
  - Does not constrain values to within the range.
  - May use constrain() function either before or after
- Example
  - /* Map an analog value (10bit) to 8 bits */
  - val = map(val, 0, 1023, 0, 255);

Lab 1. Syntax

- Syntax
  - analogWrite(pin, value)
- Parameters
  - pin: the pin number
  - value: the duty cycle, between 0 (always off) and 255 (always on)
- Example
  - analogWrite(3, 255)



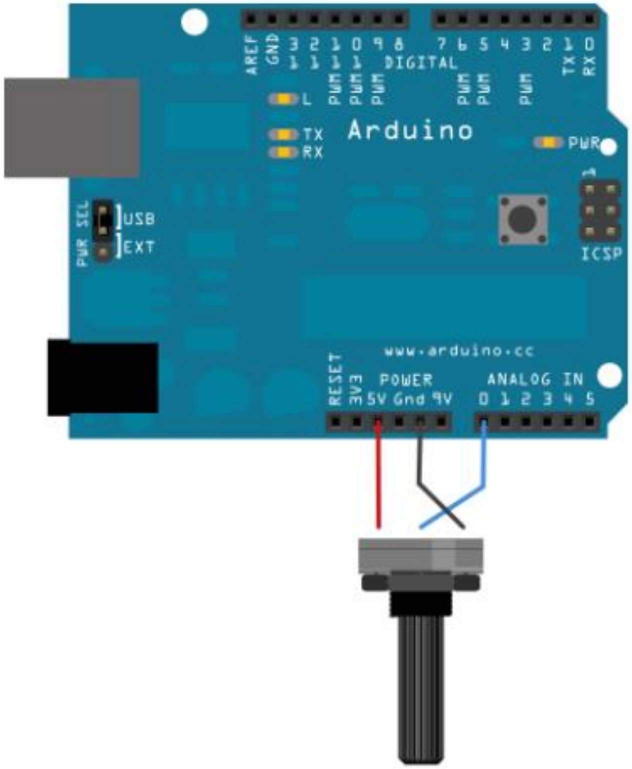
Pulse Width Modulation
0% Duty Cycle - analogWrite(0)
25% Duty Cycle - analogWrite(64)
50% Duty Cycle - analogWrite(127)
75% Duty Cycle - analogWrite(191)
100% Duty Cycle - analogWrite(255)
5v
0v

Lab2. AnalogInput:

Use a potentiometer to control the blinking of an LED.

Lab 2. AnalogInput

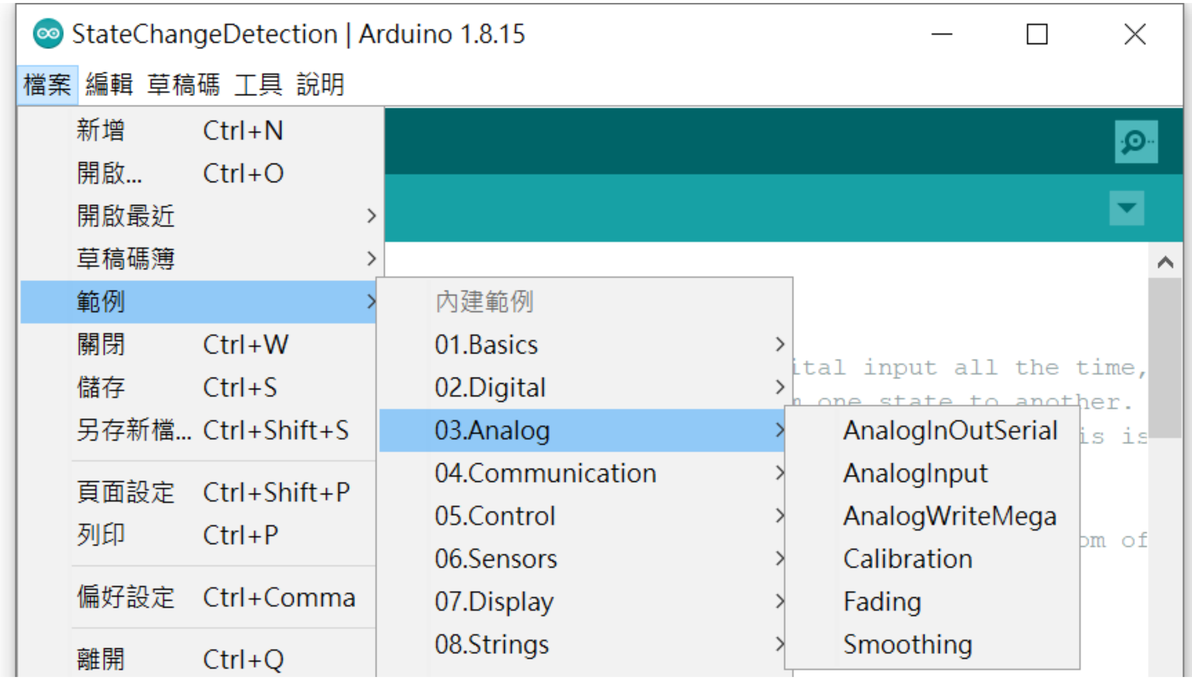
- □ Goal: connect a potentiometer to one of the Arduino's analog inputs to **control the rate** at which the built-in LED on pin 13 **blinks**.
- □ Hardware Required
  - Arduino Board
  - Potentiometer
  - built-in LED on pin 13



Lab 2. AnalogInput

Open--->File--->Examples--->Analog---> AnalogInput

Arduino IDE



Built-in Sample Code:

```
int sensorPin = A0;          // select the input pin for the potentiometer
int ledPin = 13;             // select the pin for the LED
int sensorValue = 0;         // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  sensorValue = analogRead(sensorPin); // read the value from the sensor:

  digitalWrite(ledPin, HIGH);  // turn the ledPin on
  delay(sensorValue);          // stop the program for <sensorValue> milliseconds:

  digitalWrite(ledPin, LOW);   // turn the ledPin off:
  delay(sensorValue);          // stop the program for for <sensorValue> milliseconds:
}
```

Lab 2. Syntax

- Syntax
  - analogRead(pin)
- Description
  - 10-bit analog to digital converter.
  - Map input voltages 0~5 volts ---> 0~1023.
  - Resolution: 5 volts / 1024 or, 0.0049 volts per unit.
- Returns
  - int (0 to 1023)
- Example
  - value = analogRead(3);

Lab 2. Syntax

- Syntax
  - digitalWrite(pin, value)
    - Write a HIGH or a LOW value to a digital pin
- Parameters
  - pin: the pin number
  - value: HIGH or LOW
- Example
  - digitalWrite(13, HIGH);
  - digitalWrite(13, LOW);

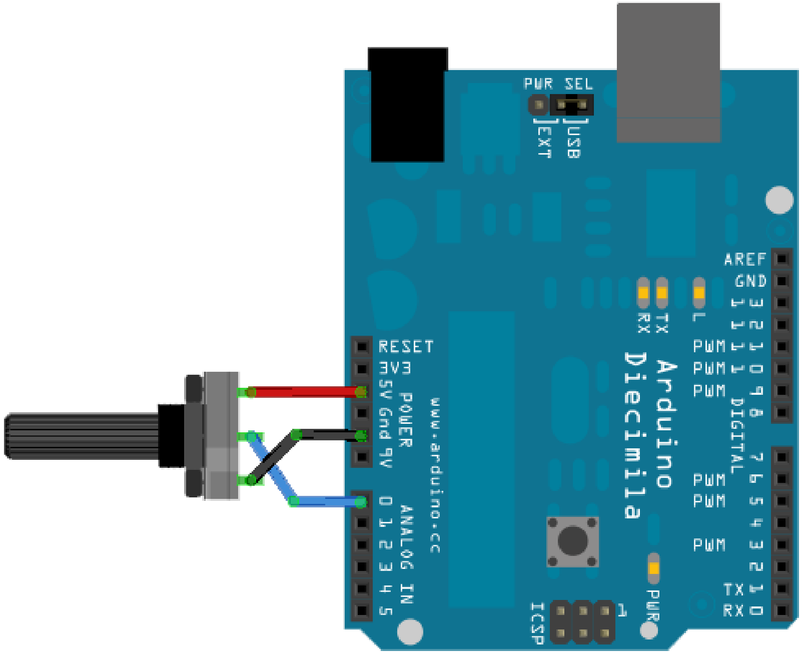
Short Summary

- □ Lab 1 shows how to use **analog sensors** to adjust the **brightness** of LED.

- □ Lab 2 shows how to use **analog sensors** to adjust the **cycle time** of LED

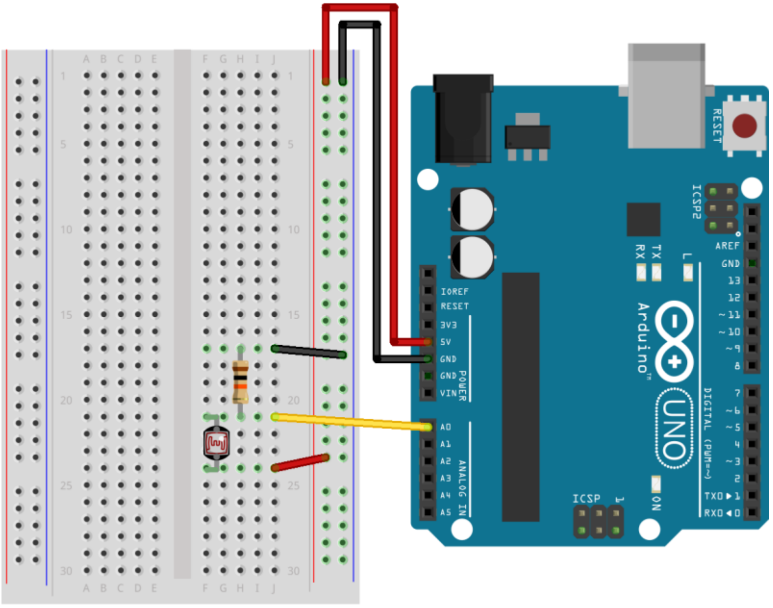
Discussion 1

- □ In this lab, potentiometer is an analog sensor. Are there other sensors that can also be used to control the blinking frequency of LED?
- □ **Please list at least two examples.**

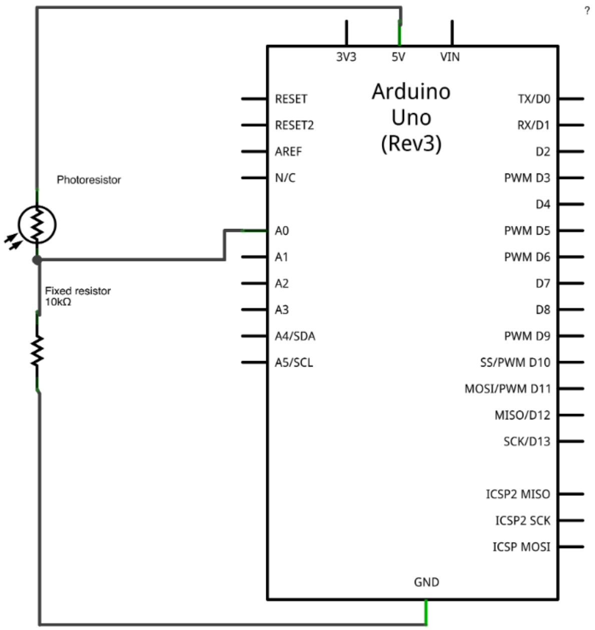


Discussion 2

- Replace the potentiometer by a **photocell** as follows.
  - What is the maximum cycle time (hint: sensor value) that you can get? Is it the same as the value given in the spec (0~1023)?



Made with Fritzing.org



Made with Fritzing.org

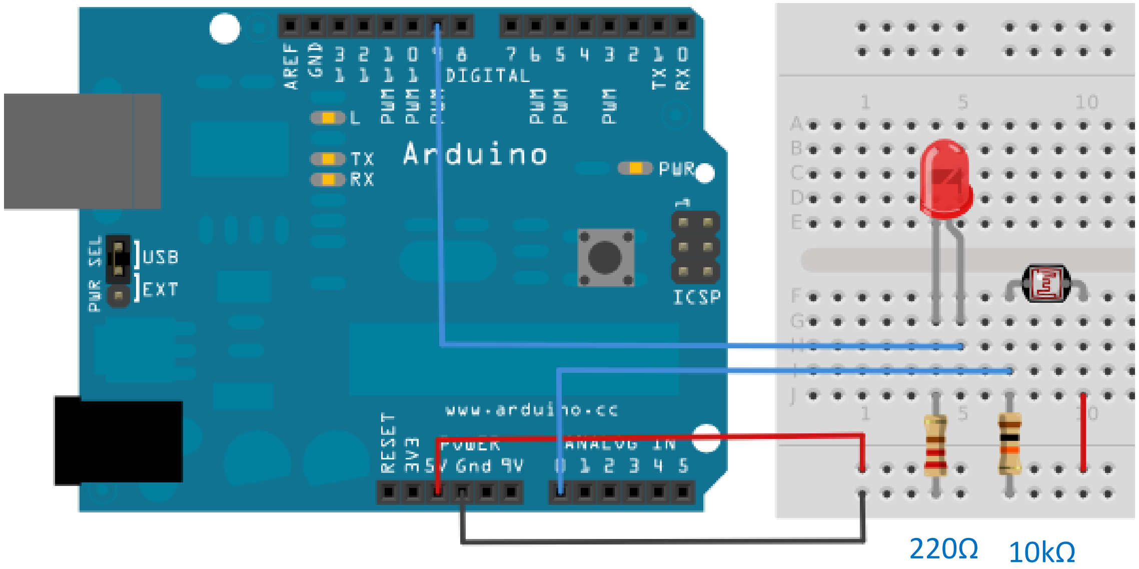
https://www.arduino.cc/en/tutorial/AnalogInput

Lab3. Calibration:

Define a maximum and a minimum for expected analog sensor values.

Lab 3. Calibration

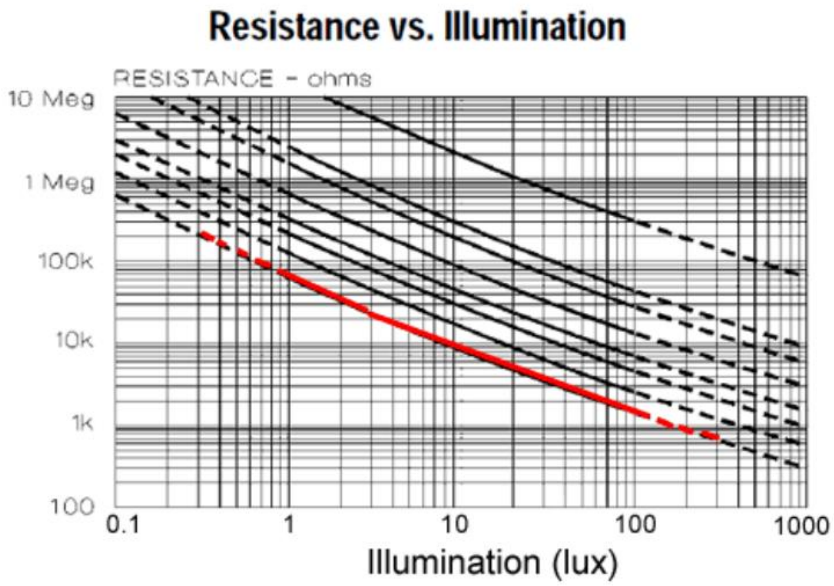
- Goal: demonstrates one technique for calibrating sensor input.
- Hardware Required
  - Arduino board
  - LED
  - analog sensor
    - Ex: photocell
  - 10K ohm resistor
  - 220 ohm resistor
  - breadboard
  - hook-up wire



Lab 3. Calibration

- Why calibration?
  - Each photocell might have different measurements because of standard error

Use calibration to obtain the same measurement!!

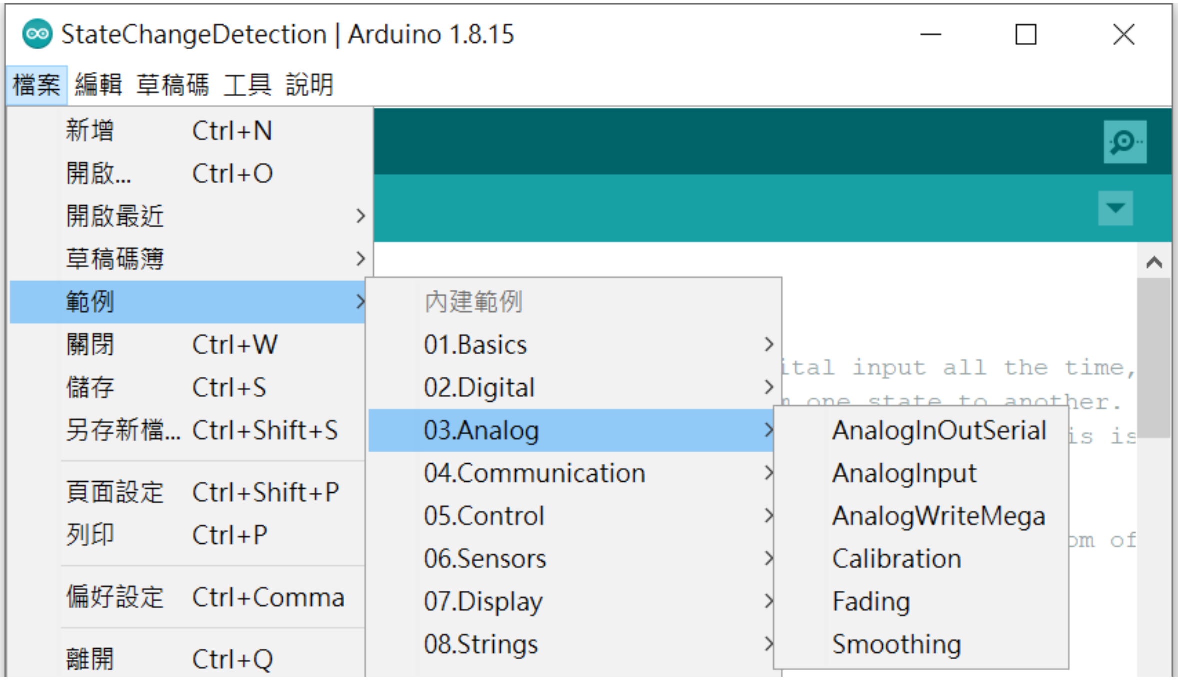


https://physics.berea.edu/wordpress/2017/10/05/calibrating-the-cds-photoresistor/

Lab 3. Calibration

Open--->File--->Examples---> Analog---> Calibration

Arduino IDE



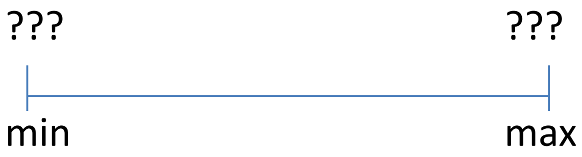
StateChangeDetection | Arduino 1.8.15

```
ital input all the time,
n one state to another.
is is

om of
```

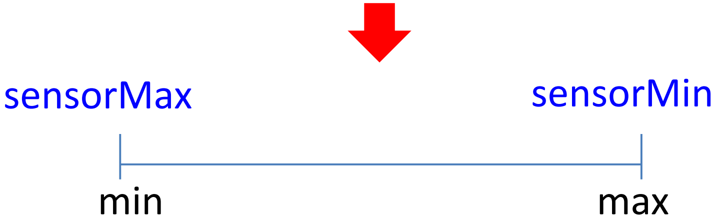
Built-in Sample Code:

```
// These constants won't change:
const int sensorPin = A0;         // pin that the sensor is attached to
const int ledPin = 9;             // pin that the LED is attached to

// variables:
int sensorValue = 0;              // the sensor value
int sensorMin = 1023;             // minimum sensor value
int sensorMax = 0;                // maximum sensor value

void setup() {
  // turn on LED to signal the start of the calibration period:
  pinMode(13, OUTPUT);
  digitalWrite(13, HIGH);

  // calibrate during the first five seconds
  while (millis() < 5000) {
    sensorValue = analogRead(sensorPin);
```

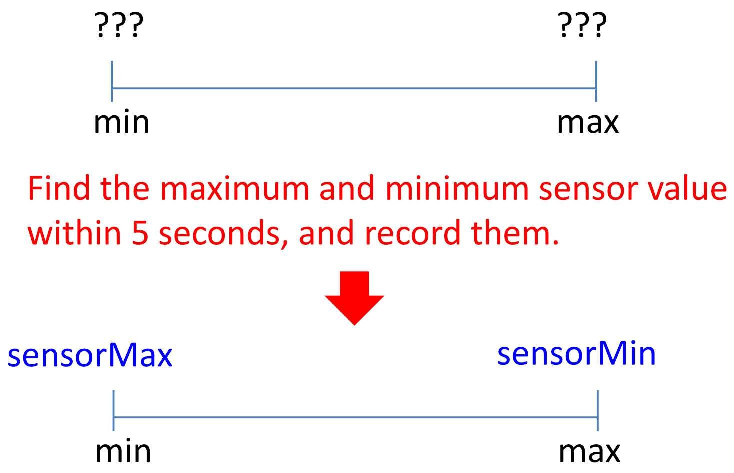
使用手電筒來照射 & 遮蔽光敏電阻
來獲得max與min值



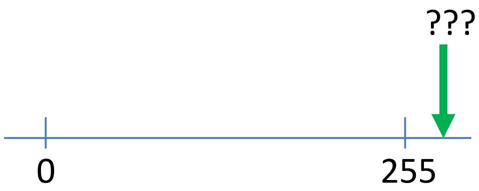
Find the maximum and minimum sensor value within 5 seconds, and record them.



```
    // record the maximum sensor value
    if (sensorValue > sensorMax) {
      sensorMax = sensorValue;
    }

    // record the minimum sensor value
    if (sensorValue < sensorMin) {
      sensorMin = sensorValue;
    }
  }

  digitalWrite(13, LOW); // signal the end of the calibration period
}

void loop() {
  sensorValue = analogRead(sensorPin); // read the sensor:

  // in case the sensor value is outside the range seen during calibration
  sensorValue = constrain(sensorValue, sensorMin, sensorMax);

  // apply the calibration to the sensor reading
  sensorValue = map(sensorValue, sensorMin, sensorMax, 0, 255);

  // fade the LED using the calibrated value:
  analogWrite(ledPin, sensorValue);
}
```



Find the maximum and minimum sensor value within 5 seconds, and record them.

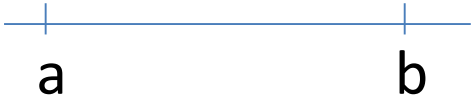


Lab 3. Syntax

- Syntax
  - map(value, fromLow, fromHigh, toLow, toHigh)
- Description
  - Re-maps a number from one range to another.
  - That is, a value of fromLow would get mapped to toLow, a value of fromHigh to toHigh, values in-between to values in-between, etc.
- Returns
  - The mapped value.
- Example
  - map(sensorValue, sensorMin, sensorMax, 0, 255);

Lab 3. Syntax

- Syntax
  - constrain(**x**, a, b)
- Description
  - Constrains a number to be within a range.
- Returns
  - **x**: if x is between a and b
  - a: if x is less than a
  - b: if x is greater than b



- Example
  - constrain(**sensorValue**, 0, 255);

Discussion 3

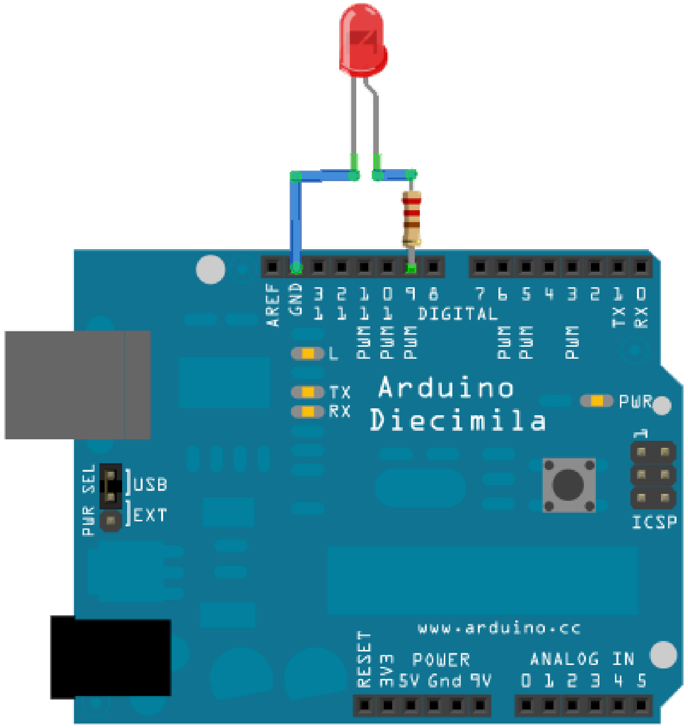
- **What may happen** if we don’t use the constrain() to constrain the sensor value.
- Try to identify the input range of photocell sensor, i.e., the max value and min value of the photocell in Lab. 3.

Lab4. Fading:

Use an analog output (PWM pin) to fade an LED.

Lab 4. Fading

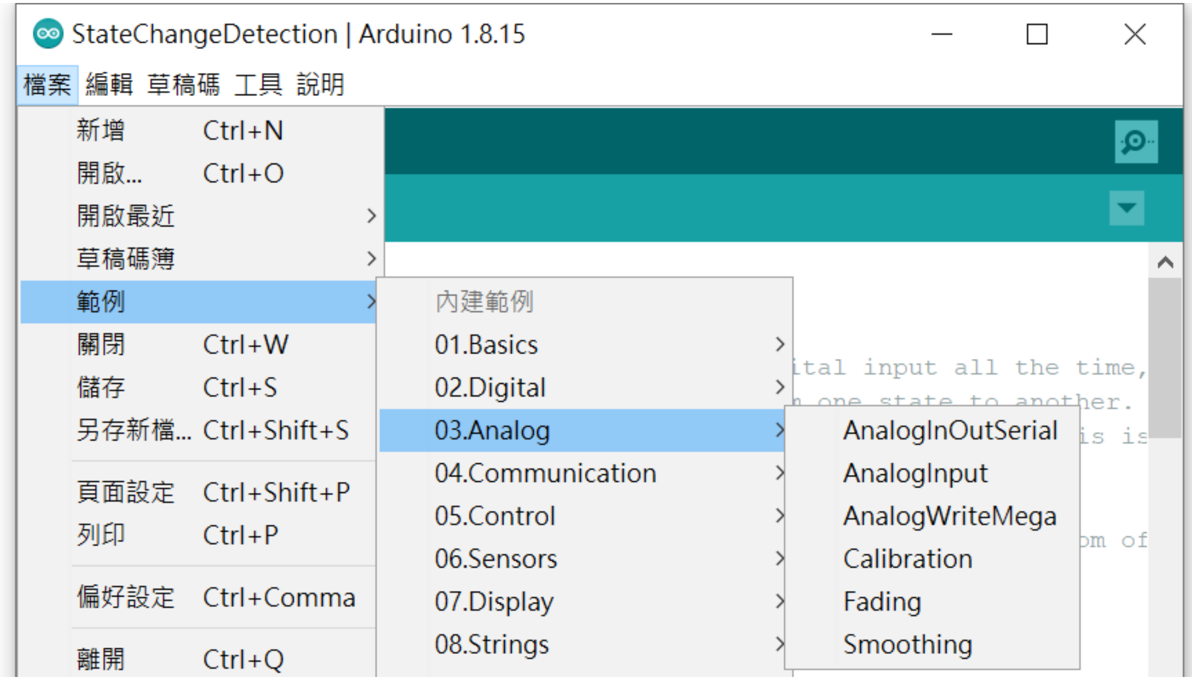
- □ Goal: demonstrates the use of analog output PWM to fade an LED.
- □ Hardware Required
  - Arduino board
  - LED
  - 220 ohm resistor



Lab 4. Fading

Open--->File--->Examples--->Analog---> Fading

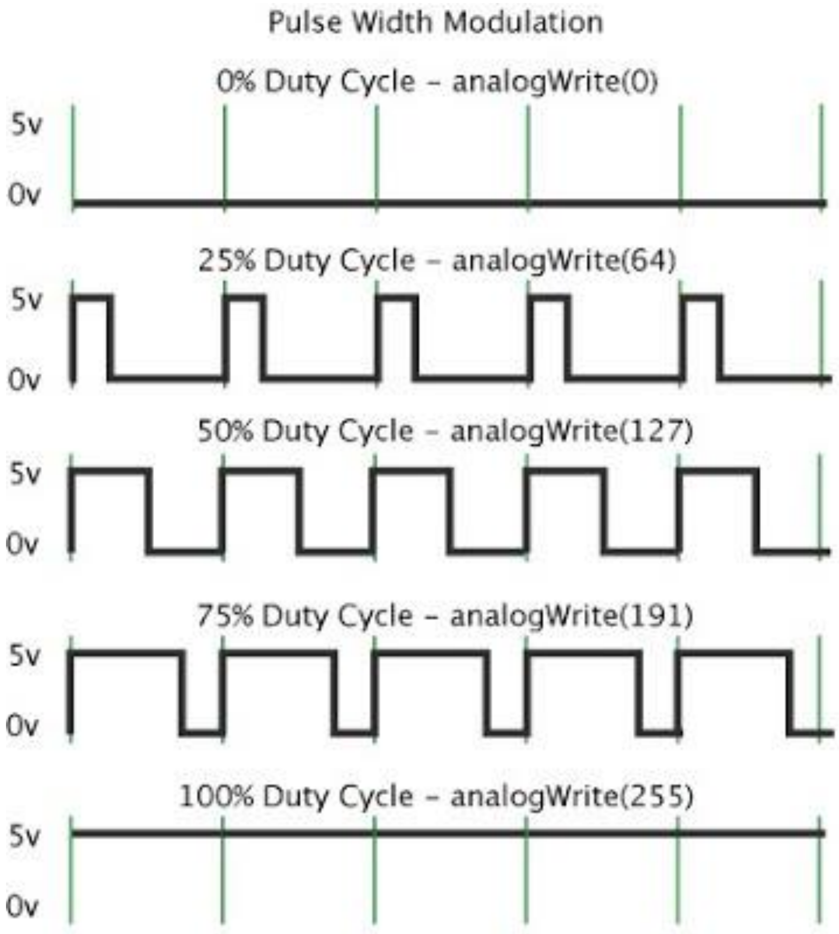
Arduino IDE



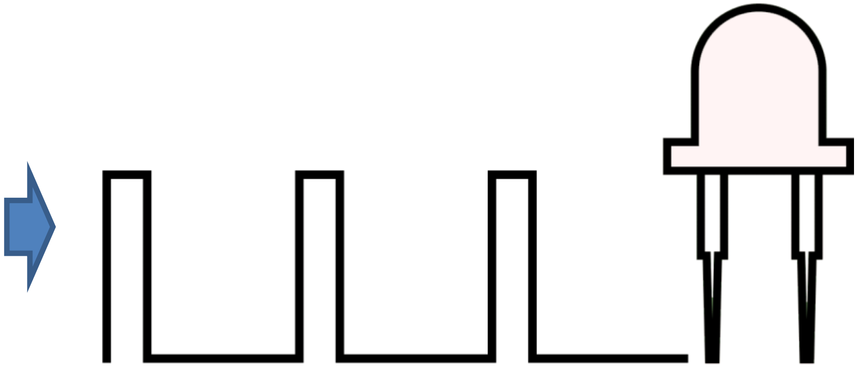
Built-in Sample Code:

```
int ledPin = 9;    // LED connected to digital pin 9

void setup() {
  // nothing happens in setup
}

void loop() {
  // fade in from min to max in increments of 5 points:
  for (int fadeValue = 0 ; fadeValue <= 255; fadeValue += 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }

  // fade out from max to min in increments of 5 points:
  for (int fadeValue = 255 ; fadeValue >= 0; fadeValue -= 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }
}
```

Lab 4. Fading



Darkness

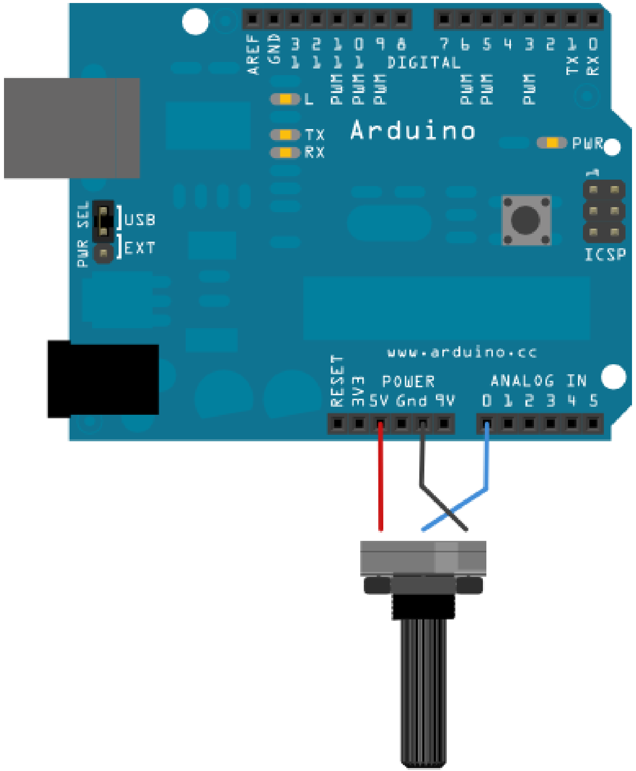


Brightness

http://yehnan.blogspot.tw/2012/02/arduino_16.html
http://www.electronicwings.com/pic/pic18f4550-pwm

Lab 5. Smoothing

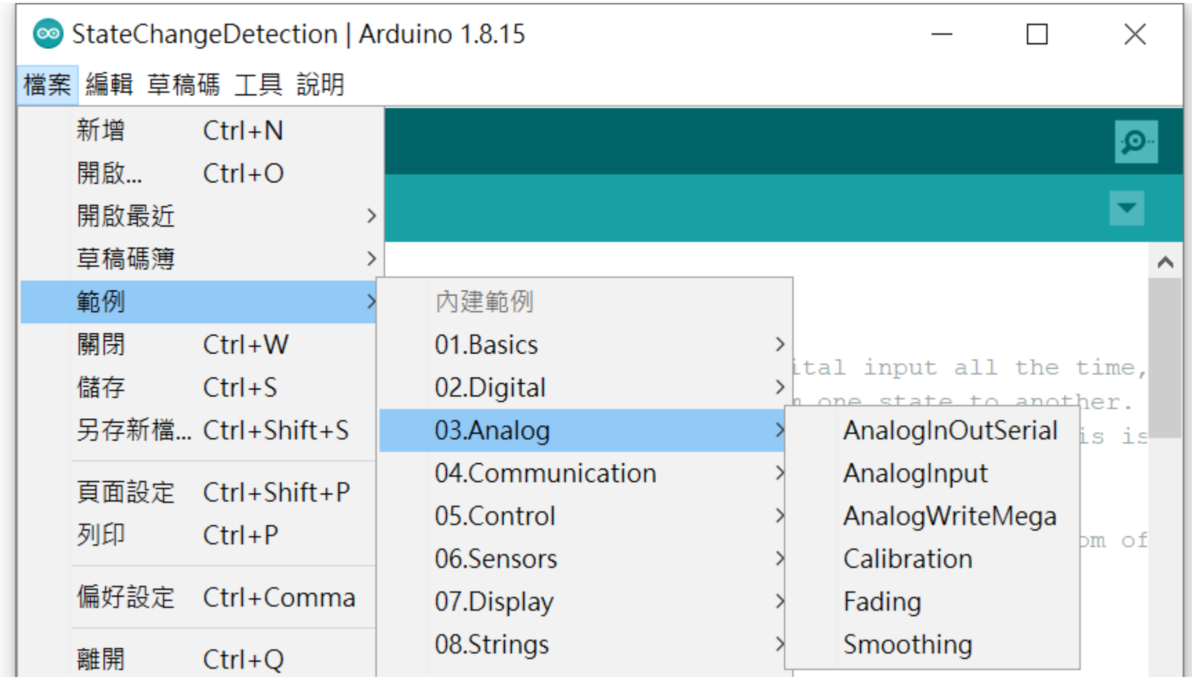
- □ Goal: learn how to smooth out the values from jumpy or erratic sensors.
- □ Hardware Required
  - □ Arduino Board
  - □ Potentiometer



Lab 5. Smoothing

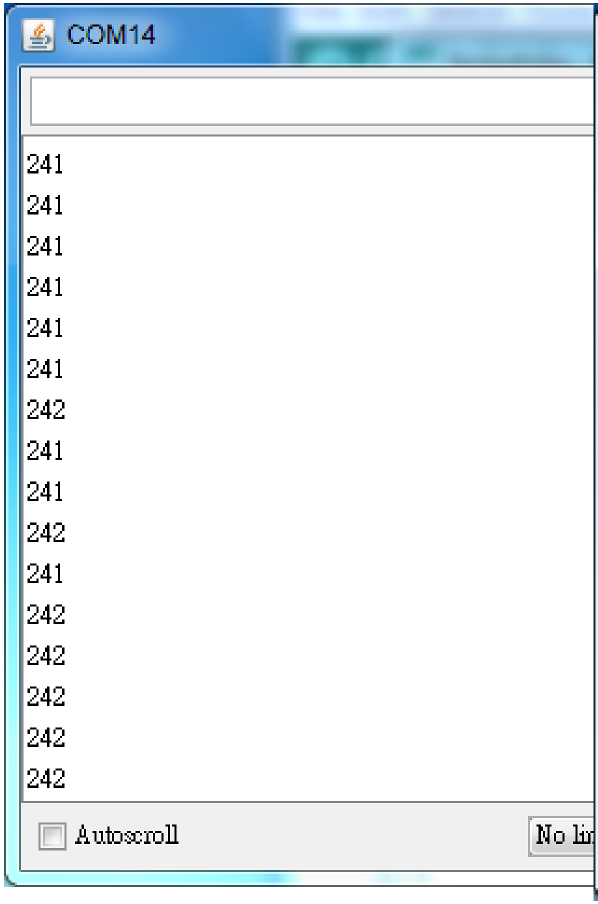
Open--->File--->Examples---> Analog---> Smoothing

Arduino IDE



Lab 5. Smoothing

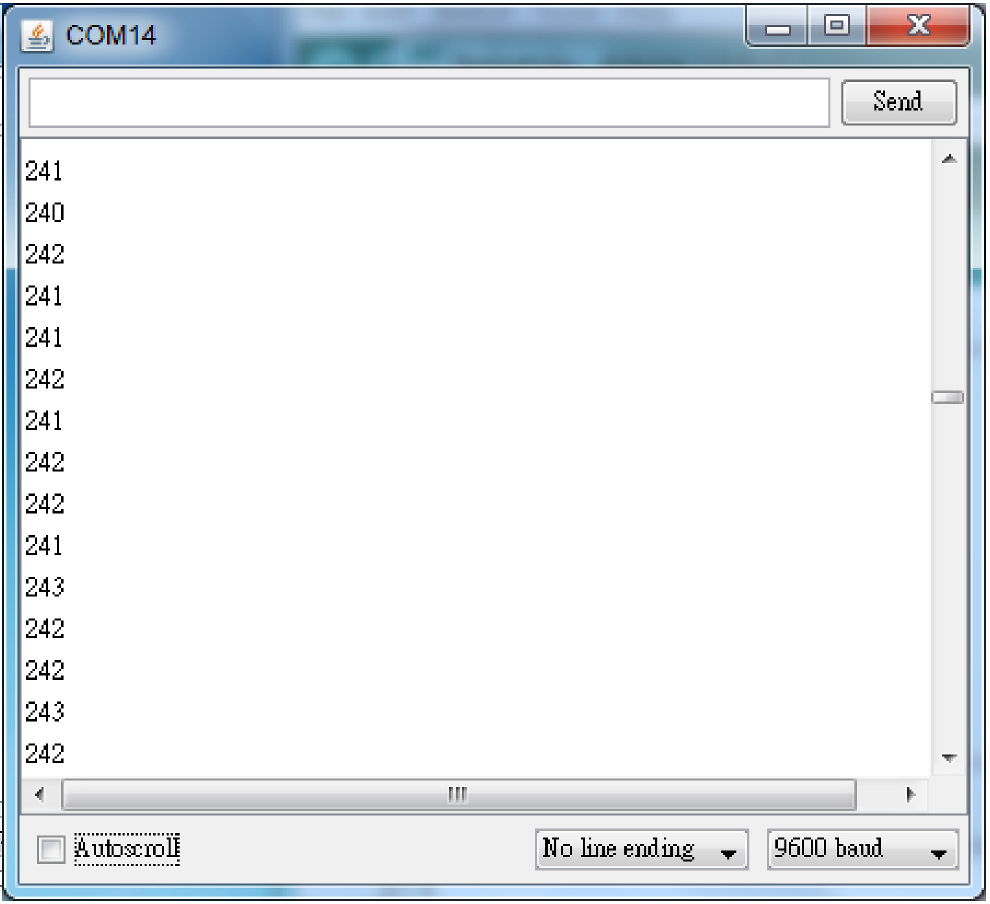
with smoothing



```
241
241
241
241
241
241
242
241
241
242
241
242
242
242
242
242
```

No lin

without smoothing

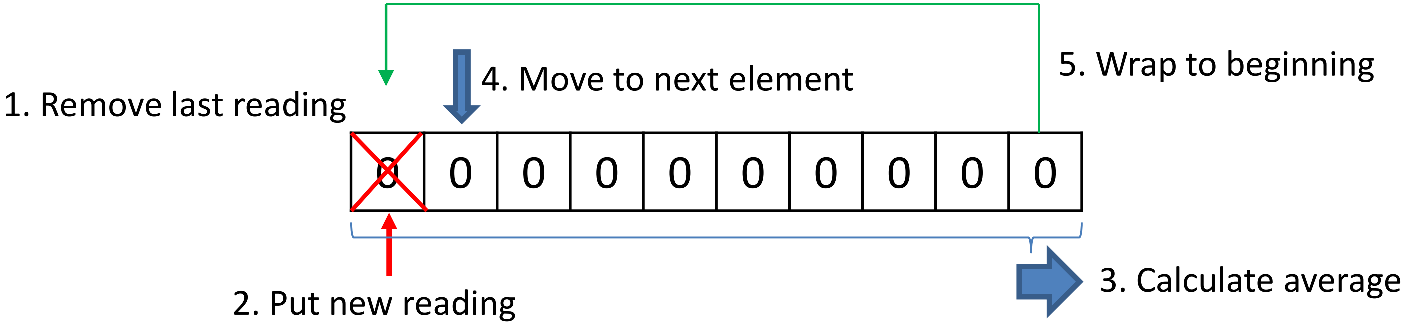


```
241
240
242
241
241
242
241
242
242
241
243
242
242
243
242
```

No line ending
9600 baud

Built-in Sample Code:

```
// Define the number of samples to keep track of. The higher the number, the
// more the readings will be smoothed, but the slower the output will respond to
// the input. Using a constant rather than a normal variable lets us use this
// value to determine the size of the readings array.
const int numReadings = 10;

int readings[numReadings];      // the readings from the analog input
int readIndex = 0;              // the index of the current reading
int total = 0;                  // the running total
int average = 0;                // the average

int inputPin = A0;

void setup() {
  // initialize serial communication with computer:
  Serial.begin(9600);
  // initialize all the readings to 0:
  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = 0;
  }
}
```



```
void loop() {
  total = total - readings[readIndex];                          // subtract the last reading:
  readings[readIndex] = analogRead(inputPin);                   // read from the sensor:
  total = total + readings[readIndex];                          // add the reading to the total:
  readIndex = readIndex + 1;                                    // advance to the next position in the array:

  // if we're at the end of the array...
  if (readIndex >= numReadings) {
    // ...wrap around to the beginning:
    readIndex = 0;
  }

  average = total / numReadings;                   // calculate the average:
  Serial.println(average);                         // send it to the computer as ASCII digits
  delay(1);                                        // delay in between reads for stability
}
```

Lab 5. Smoothing

{5, 6, 8, 10, 6, 7, 7, 7, 9, 6, 7, 8, 10, 8, 9}

5	6	8	10	6	7	7	7	9	6	Total: 71	
7	6	8	10	6	7	7	7	9	6	Total: 73	Print: 73/10 = 7
7	8	8	10	6	7	7	7	9	6	Total: 75	Print: 75/10 = 7
7	8	10	10	6	7	7	7	9	6	Total: 77	Print: 77/10 = 7
7	8	10	8	6	7	7	7	9	6	Total: 75	Print: 75/10 = 7
7	8	10	8	9	7	7	7	9	6	Total: 78	Print: 78/10 = 7

Quiz 1

- □ Try to use FunctionDeclaration for Calibration and Smoothing in Lab 3.
  - □ Step 1: Write Smoothing as a function and add the Smoothing() into Lab 3.
  - □ Step 2: Write Calibration as a function Calibration() as well.
  - □ **PS: Calibration () and Smoothing () will be used in the future labs.**
- □ Submit your code

Example:

```
void loop() {
  int i = 2;
  int j = 3;
  int k;

  k = myMultiplyFunction(i, j); // k now contains 6
  Serial.println(k);
  delay(500);
}
```

FunctionDeclaration

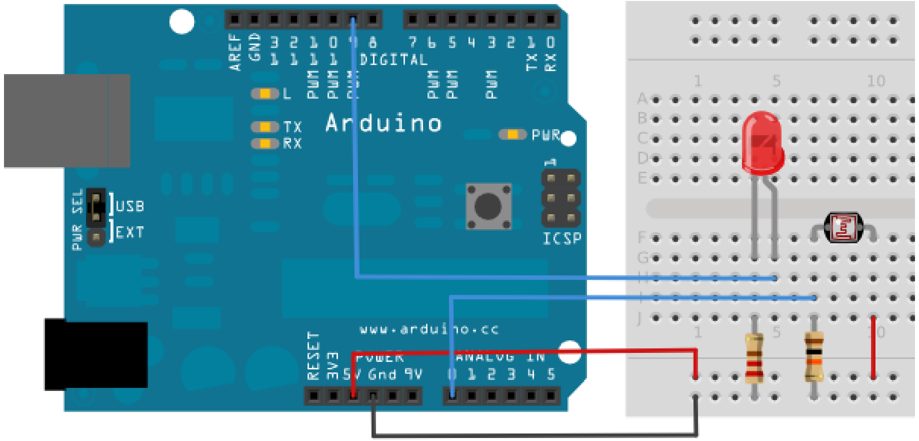
```
int myMultiplyFunction(int x, int y){
  int result;
  result = x * y;
  return result;
}
```

https://www.arduino.cc/en/Reference/FunctionDeclaration

Quiz 2

- Scenario: Automatically turn on/off the lights in the building depending on the room light level.
- Use photocell sensor to design a LED switch.
  - Turn on the LED, if it is dark
  - Turn off the LED, otherwise

PS: Calibration() and Smoothing() should be used in this quiz.



you may refer to the circuit in Lab 3.

Quiz 3

- In smoothing, it uses simple moving average
  - When calculating, a new value comes into the sum and an old value drops out
- try to **use "weighted moving average"** to smooth sensor data
  - It gives different weights to data at different positions in the sample window

$$\mathrm{WMA_M} = \frac{n * p_M + (n-1) * p_{M-1} + \cdots + 2 * p_{M-n+2} + 1 * p_{M-n+1}}{n + (n-1) + \cdots + 2 + 1}$$

$$= \frac{10 * p_{10} + 9 * p_9 + \cdots + 2 * p_2 + p_1}{10 + 9 + \cdots + 2 + 1}$$

https://en.wikipedia.org/wiki/Moving_average

Summary

Summary

- □ Practice Labs by yourself

- □ **Write Answers for Discussion 1 to 3**
  - □ Upload to e3 before next class

- □ **Quiz: Write code for quiz, then demonstrate to TAs**
  - □ 1. Use “FunctionDeclaration” for Calibration and Smoothing
  - □ 2. Use photocell sensor to design a LED switch.
  - □ 3. Use “weighted moving average” to smooth sensor data.