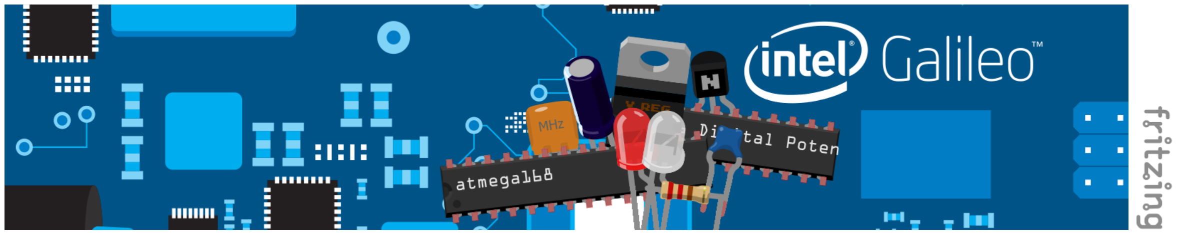
Module 4.1: Digital/Analog & IO

hadrihl // hadrihilmi@gmail.com

Monday September 16th 2014

Lab 4 School of Computer Science USM



Parallel & Distributed Computing Center (PDCC), 412-1 Level 4 School of Computer Sciences, Universiti Sains Malaysia, 11800 USM Pulau Pinang, MALAYSIA. +604 – 653 3888 (Ext: 2319)

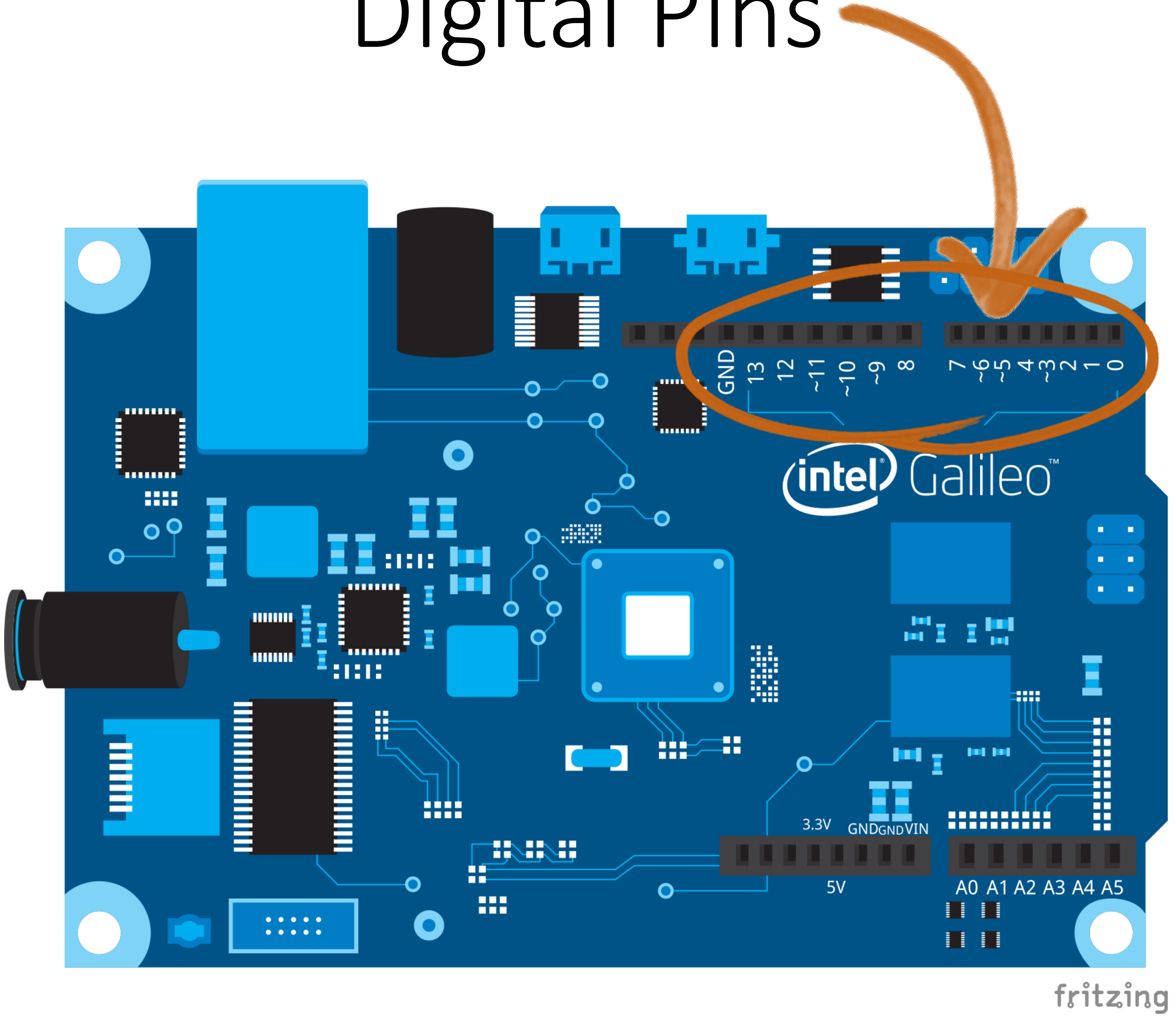
Agenda

- Understand Input and Output
- Understand Digital and Analog pins

Activity 1

- Name the examples of:
- Digital input
- Digital output
- Analog input
- Analog output

Digital Pins



Digital Pins APIs (cont.)

pinMode()

Description: configures specified pin to behave either input or output

Syntax:

Pinmode(pin, mode);

Parameter:

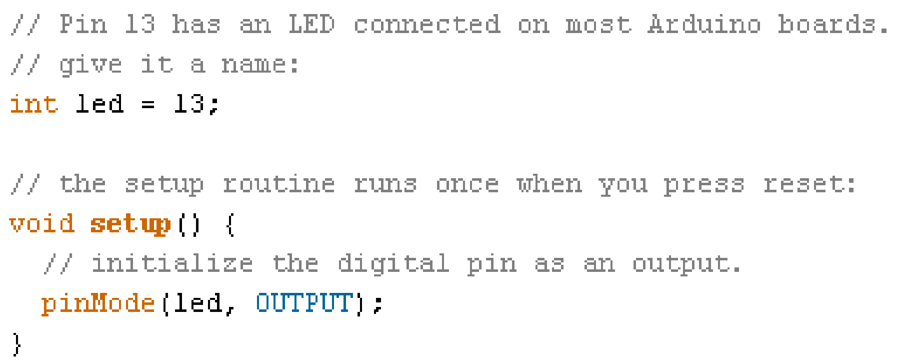
pin: number of pin (int)

mode: INPUT, OUTPUT, or INPUT_PULLUP

Returns:

HIGH or LOW

sample code



```
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}
```

reference: http://arduino.cc/en/Reference/PinMode

Digital Pins APIs (cont.)

digitalRead()

Description:

Read the value from specified digital pin (HIGH or LOW)

Syntax:

digitalRead(pin);

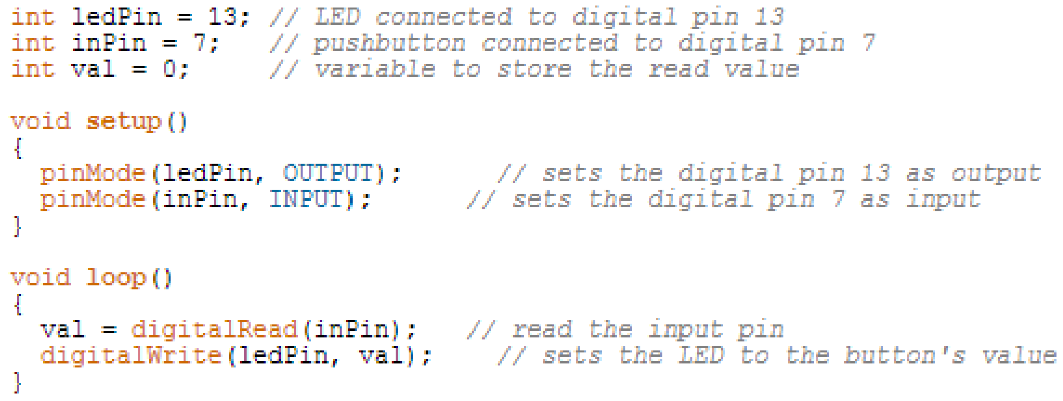
Parameter:

pin: the pin of digital input (int)

Returns:

HIGH or LOW

sample code



```
int ledPin = 13; // LED connected to digital pin 13
int inPin = 7;   // pushbutton connected to digital pin 7
int val = 0;     // variable to store the read value

void setup()
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin 13 as output
  pinMode(inPin, INPUT);      // sets the digital pin 7 as input
}

void loop()
{
  val = digitalRead(inPin);   // read the input pin
  digitalWrite(ledPin, val);    // sets the LED to the button's value
}
```

http://arduino.cc/en/Reference/digitalRead

Digital Pins APIs (cont.)

digitalWrite()

Description:

Write a HIGH or LOW value to a digital pin

Syntax: digitalWrite(pin, value);

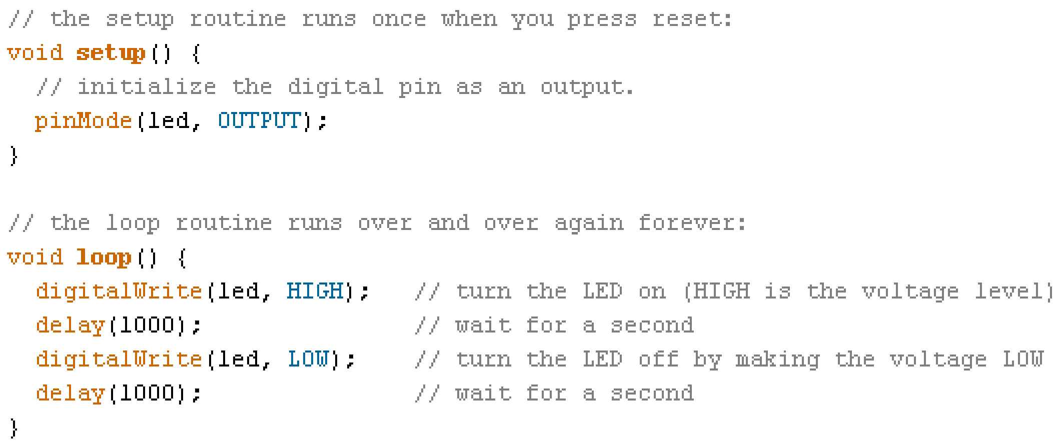
Parameter:

pin: the pin of digital output (int)

value: HIGH or LOW

Returns: none

sample code

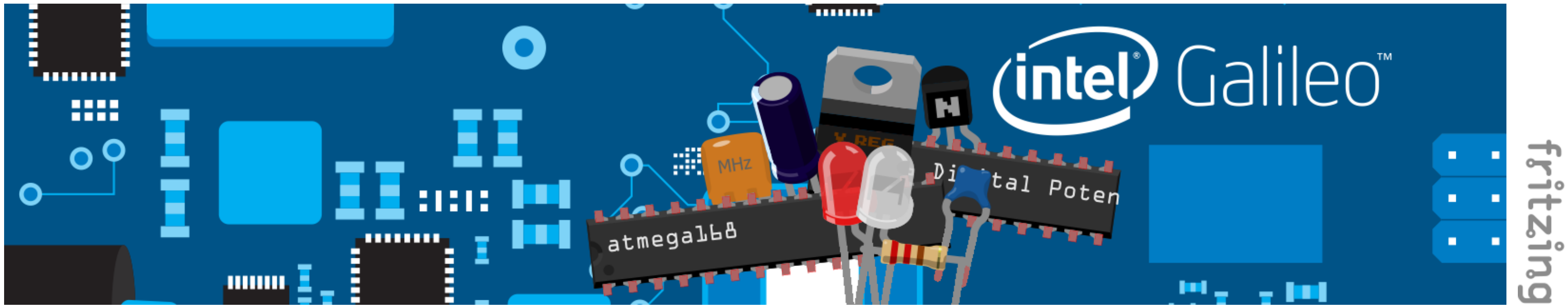


```
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

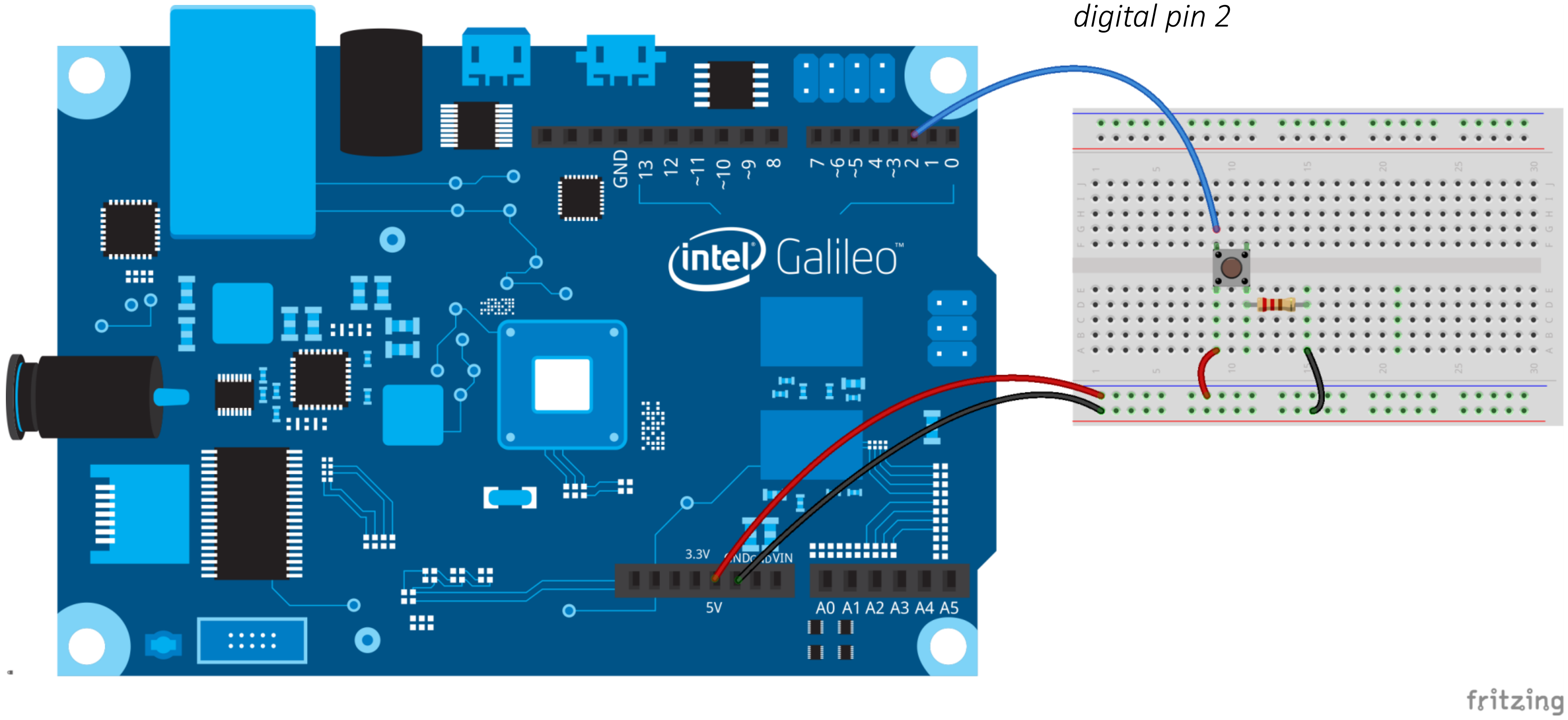
http://arduino.cc/en/Reference/DigitalWrite

Activity 2

- Use *Pushbutton*, turns ON the LED on pin 9 when you press the button on pin 2



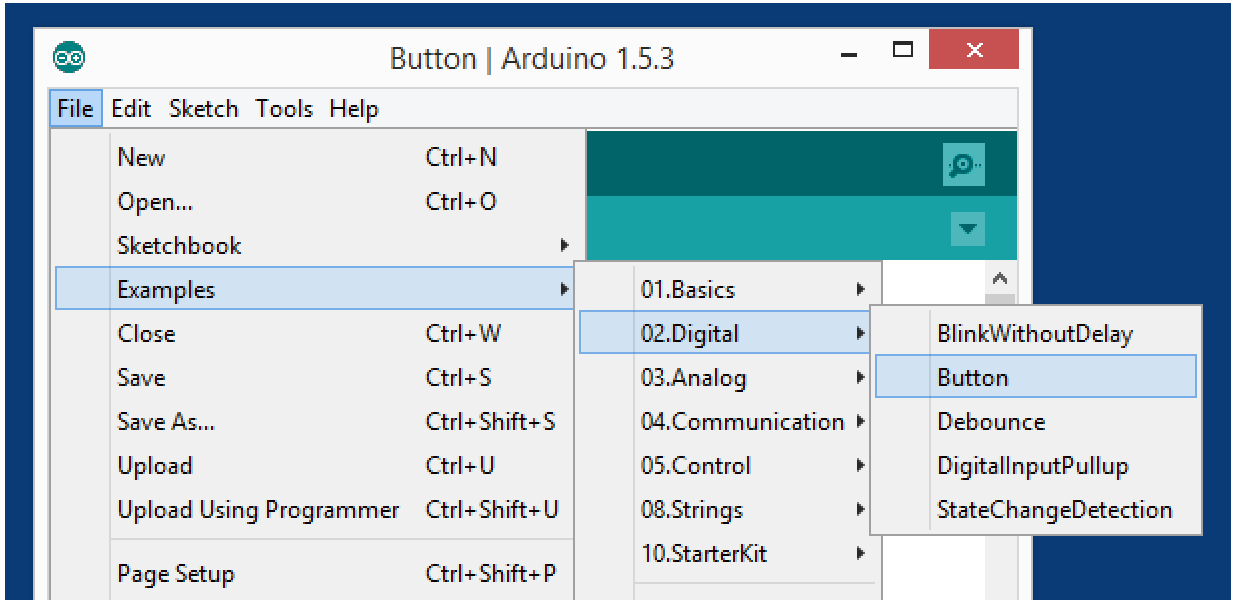
Activity 2 (cont.)



- *Pushbutton attached to digital pin 2*

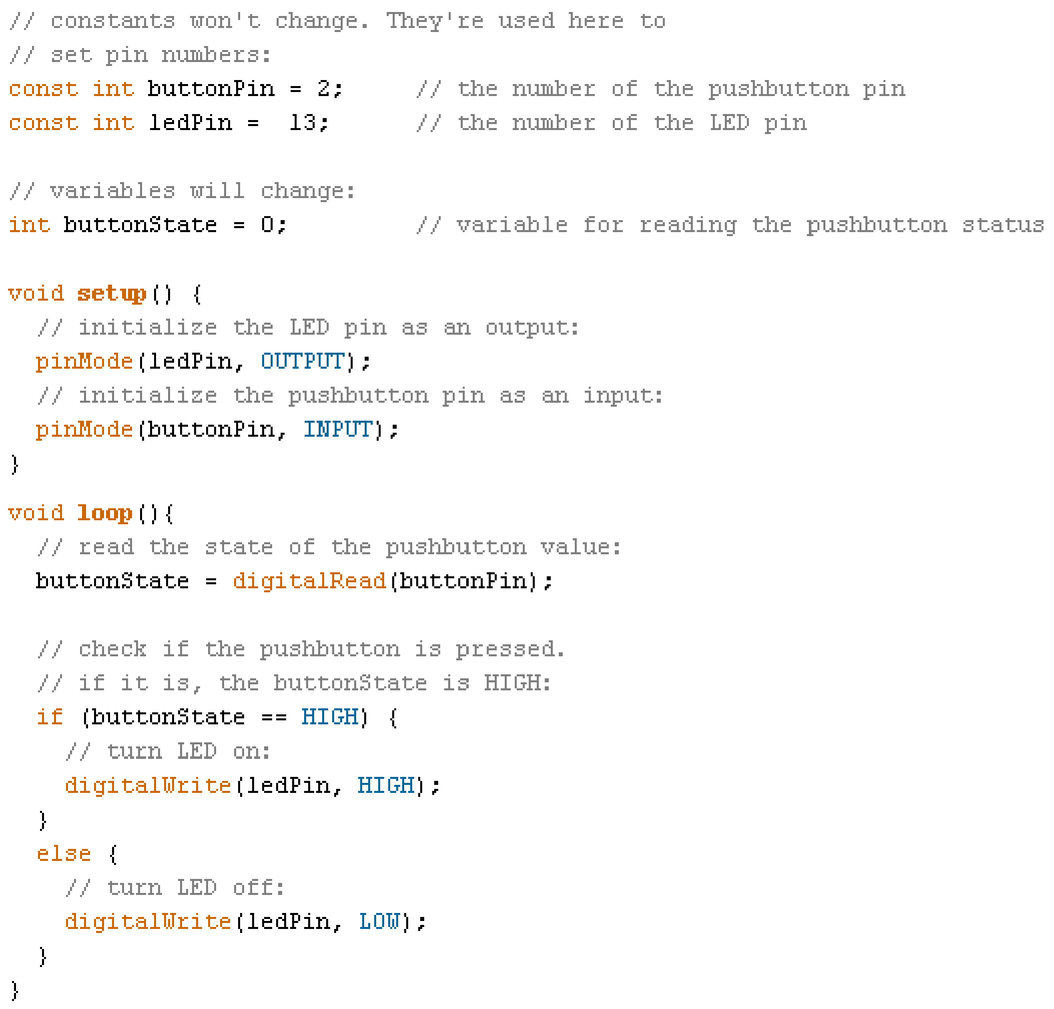
Activity 2 (cont.)

- Open “Button” sketch example, compile and upload



WARNING: verify COM port before uploading the sketch into Galileo

Activity 2 (cont.)

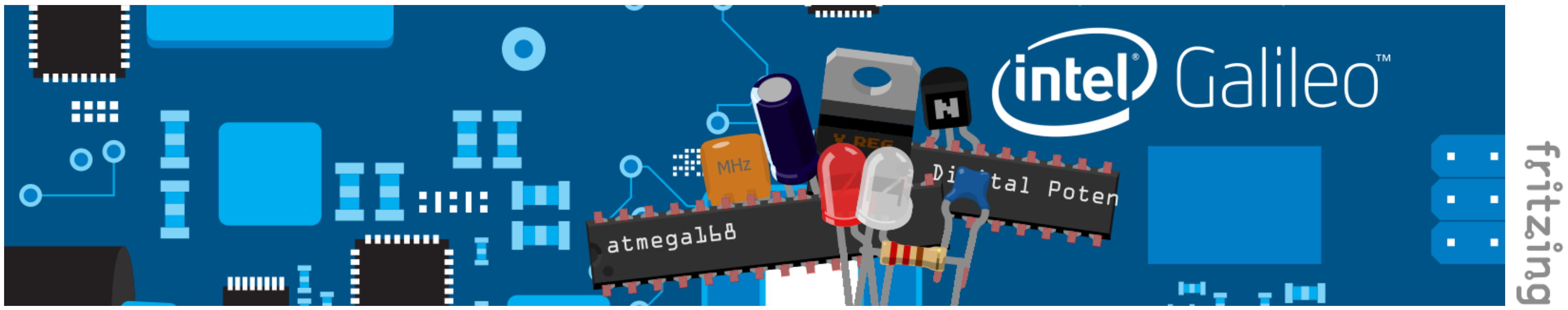


```
// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

Code Example: http://www.arduino.cc/en/Tutorial/Button

Activity 2 (cont.)

- Print out the value of pushbutton and observe through Serial Monitor
  - What is the value of buttonState when pushbutton is not pressed?
  - What is the value of buttonState when pushbutton is pressed?



Digital Pins APIs (cont.)

pinMode()

Description: configures specified pin to behave either input or output

Syntax:

Pinmode(pin, mode);

Parameter:

pin: number of pin (int)

mode: INPUT, OUTPUT or **INPUT_PULLUP**

Returns:

HIGH (1) or LOW (0)

Digital Pins APIs (cont.)

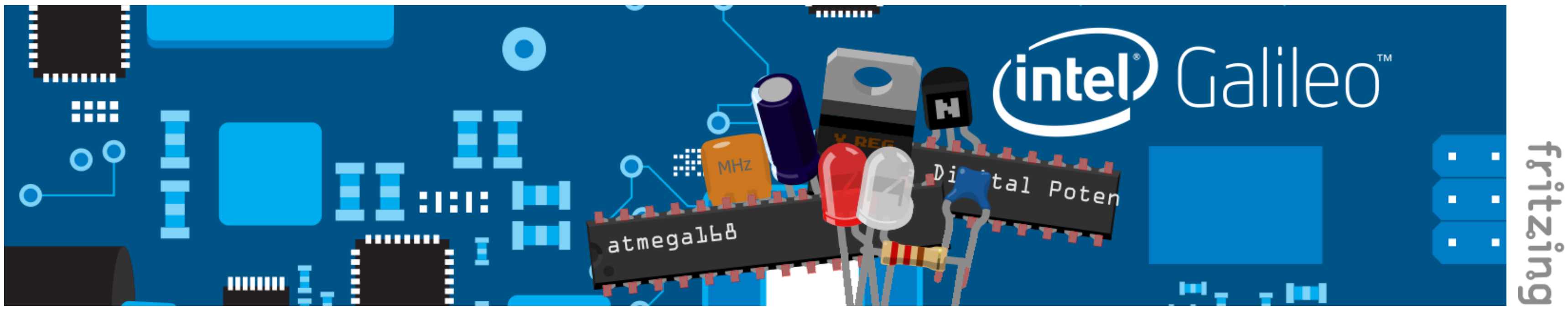
- Pin default as INPUT. When pin disconnect, It randomly read HIGH or LOW
- Examples, when wiring buttons or switches or anything "normally open", we have to tie them to the ground. A 10k Ohm resistor can act as pull down resistor for digital input pin

Syntax: pinMode(pin, INPUT_PULLUP)

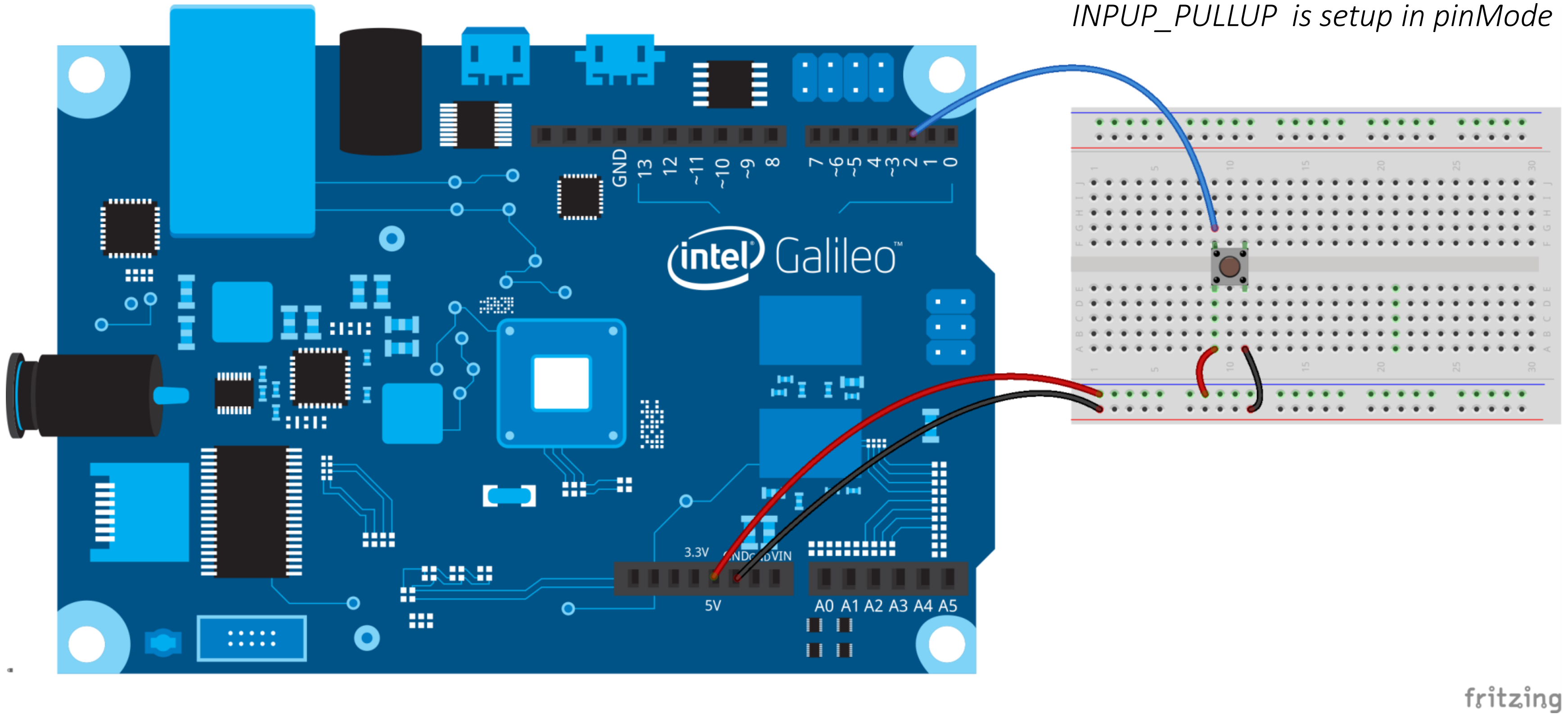
Description: configures specified pin to behave either input or output

Activity 3

- Turns OFF the LED on pin 9 when you press the button on pin 2 using INPUT_PULLUP in pinMode()



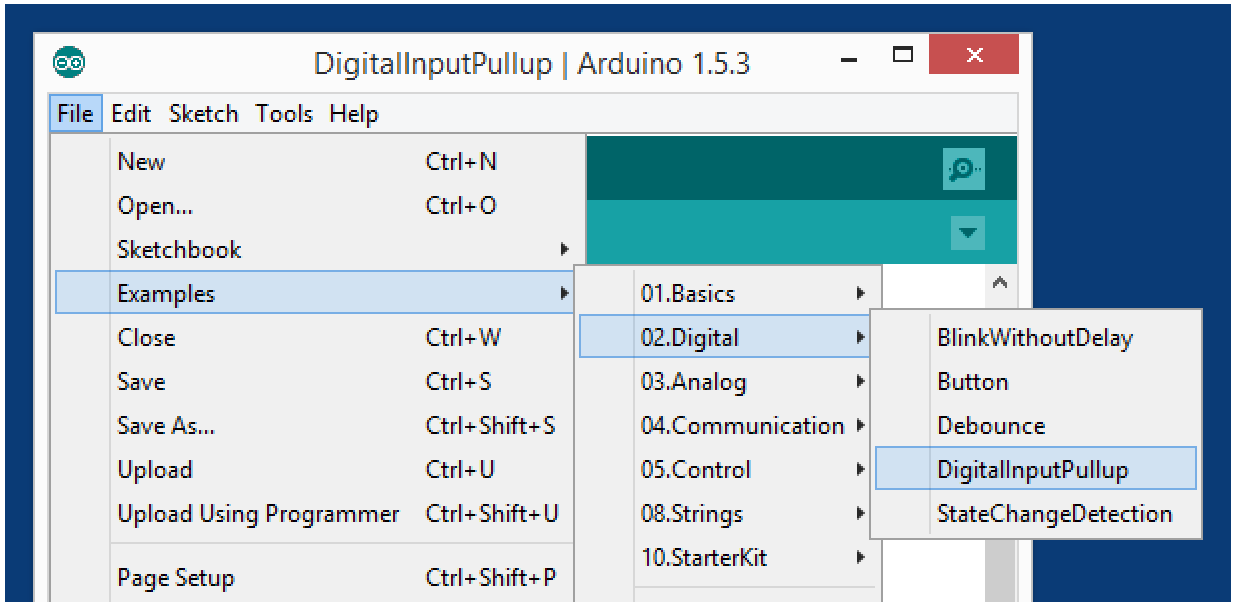
Activity 3 (cont.)



- NO resistor this time since INPUP_PULLUP is setup in pinMode

Activity 3 (cont.)

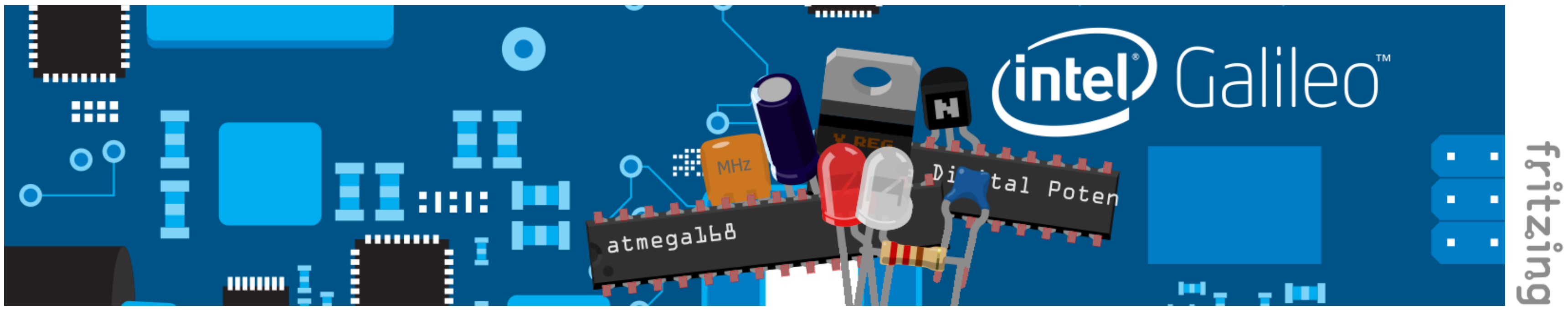
- Open "DigitalInputPullup" sketch example, compile and upload



WARNING: verify COM port before uploading the sketch into Galileo

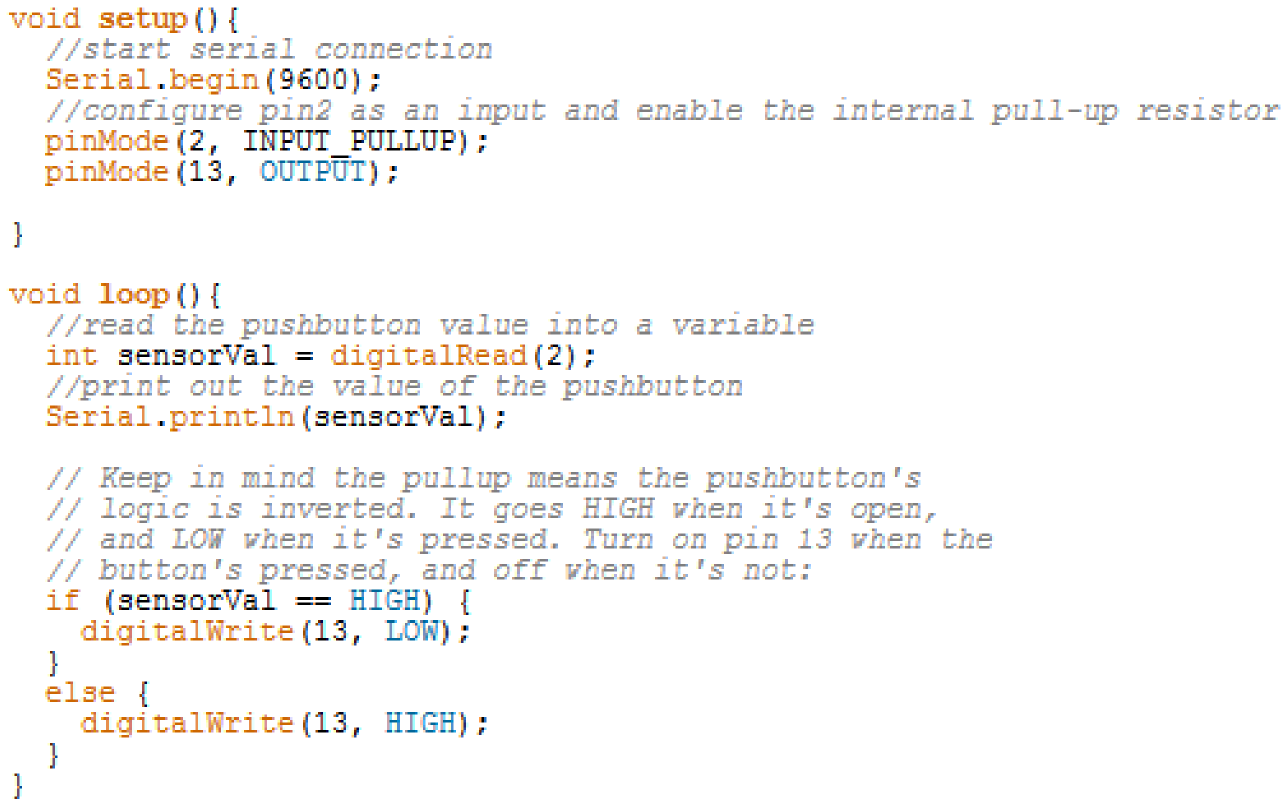
Activity 3 (cont.)

- Print out the value of pushbutton and observe through Serial Monitor
  - What is the value of buttonState when pushbutton is not pressed?
  - What is the value of buttonState when pushbutton is pressed?



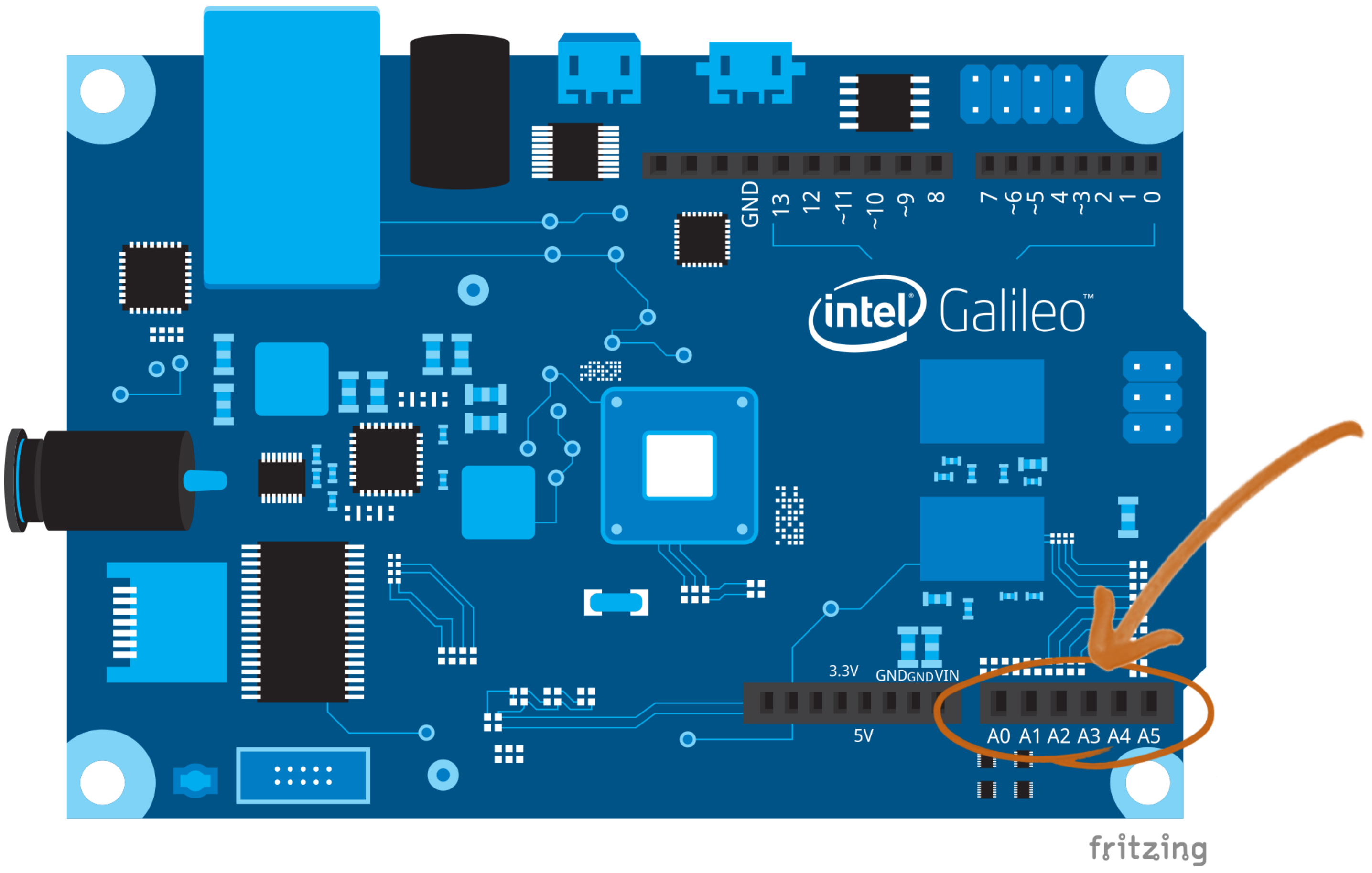
Activity 3 (cont.)

Code Example: http://arduino.cc/en/Tutorial/InputPullupSerial



```
void setup(){
  //start serial connection
  Serial.begin(9600);
  //configure pin2 as an input and enable the internal pull-up resistor
  pinMode(2, INPUT_PULLUP);
  pinMode(13, OUTPUT);

}

void loop(){
  //read the pushbutton value into a variable
  int sensorVal = digitalRead(2);
  //print out the value of the pushbutton
  Serial.println(sensorVal);

  // Keep in mind the pullup means the pushbutton's
  // logic is inverted. It goes HIGH when it's open,
  // and LOW when it's pressed. Turn on pin 13 when the
  // button's pressed, and off when it's not:
  if (sensorVal == HIGH) {
    digitalWrite(13, LOW);
  }
  else {
    digitalWrite(13, HIGH);
  }
}
```

Analog Pins


GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
1
0
intel
Galileo
3.3V
GND
GND
VIN
5V
A0 A1 A2 A3 A4 A5
fritzing

Analog Pins API (cont.)

analogRead()

Description: read Read the value from specified analog pin

Syntax:

analogRead(pin);

Parameter:

pin: the pin of analog input (0 to 5)

Returns:

int (0 to 1023)

Scale:

0 to 1023 = 0V to 5V

http://arduino.cc/en/Reference/analogRead

Analog Pins API (cont.)

analogWrite() (PWM) ~

Description: Write the value to specified analog input

Syntax:

analogWrite(pin, value);

Parameter:

pin: the pin to write to

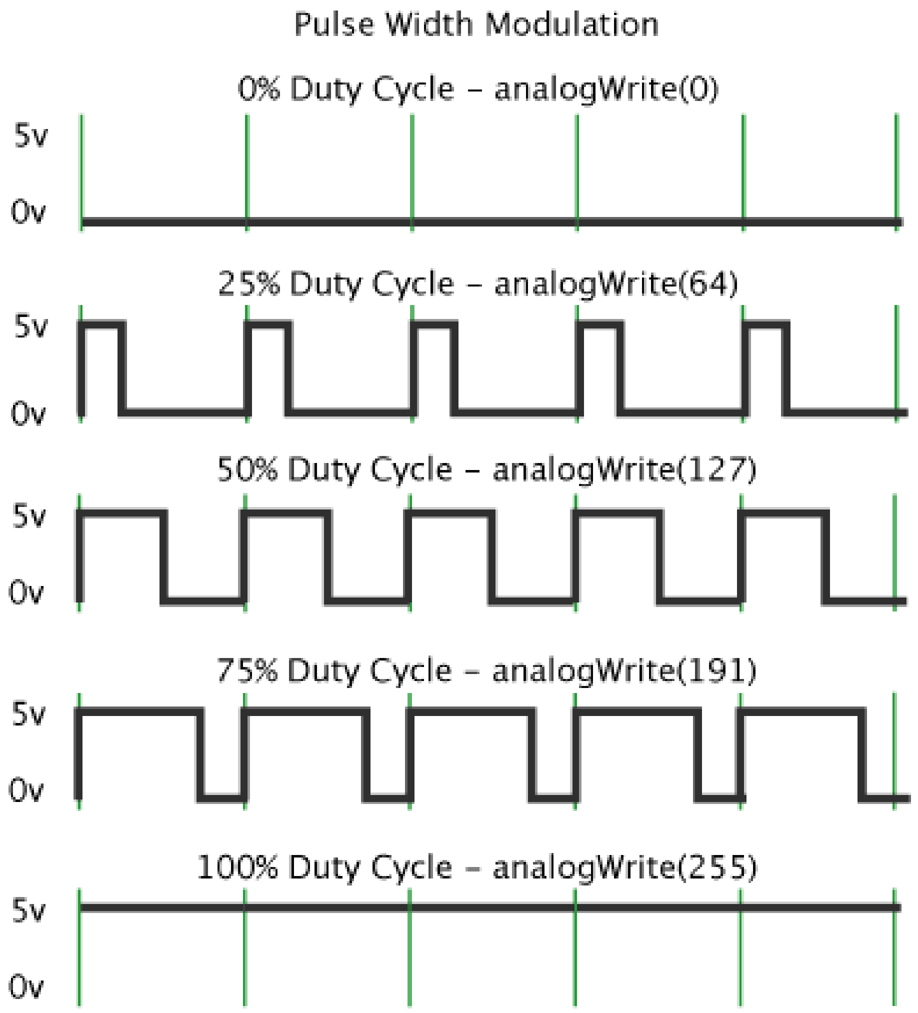
Value: int (0 to 255)

http://arduino.cc/en/Reference/analogWrite

PWM

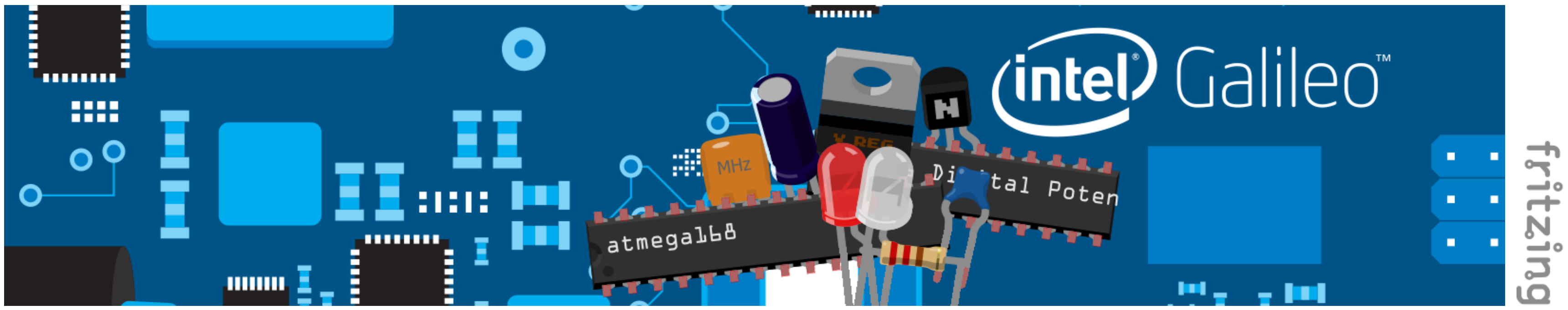
- Pulse Width Modulation: output analog results with digital means
- It is not a “pure” analog output

PWM (cont.)

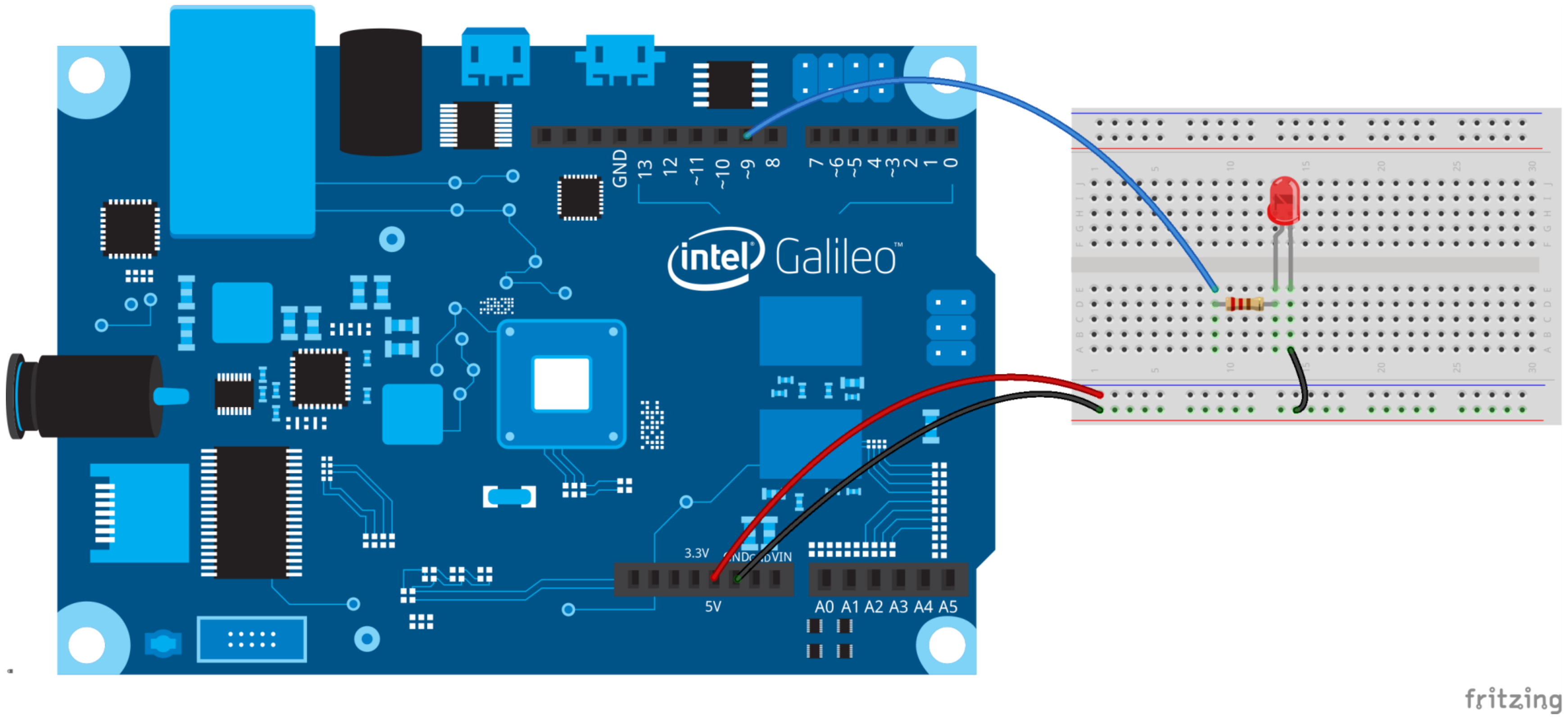

Pulse Width Modulation
0% Duty Cycle – analogWrite(0)
5v
0v
25% Duty Cycle – analogWrite(64)
5v
0v
50% Duty Cycle – analogWrite(127)
5v
0v
75% Duty Cycle – analogWrite(191)
5v
0v
100% Duty Cycle – analogWrite(255)
5v
0v

Activity 4

- Use PWM to fade the LED automatically

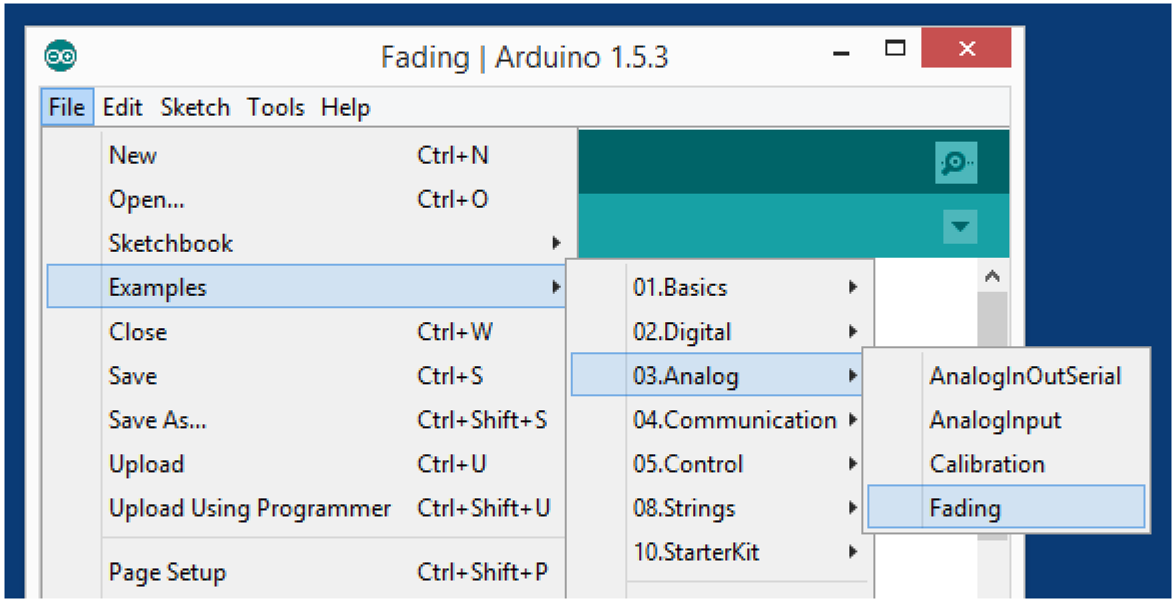


Activity 4 (cont.)



Activity 4 (cont.)

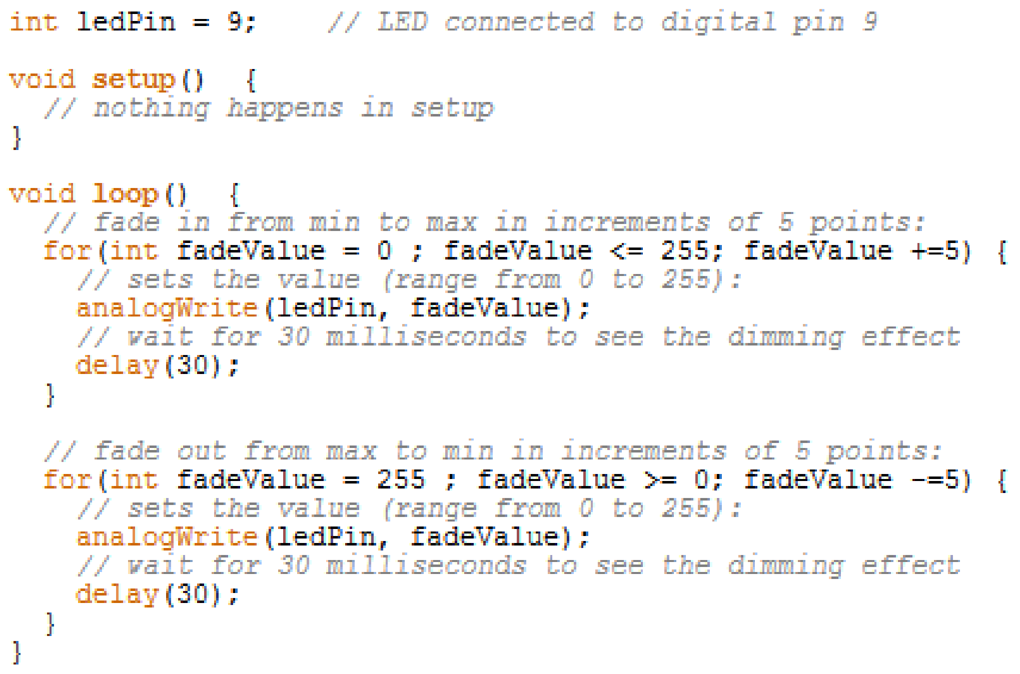
- Open "Fading" sketch example, compile and upload



WARNING: verify COM port before uploading the sketch into Galileo

Activity 4 (cont.)

Code Example: http://arduino.cc/en/Tutorial/Fading



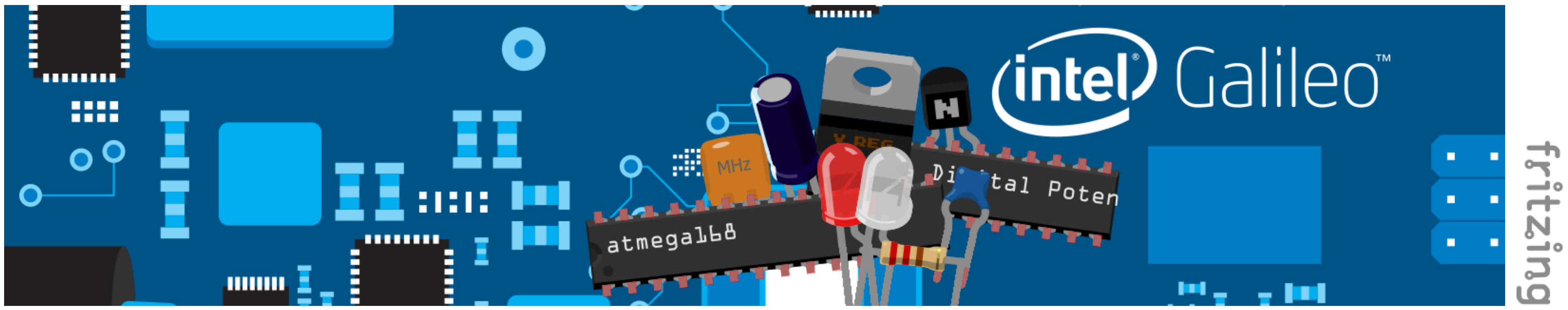
```
int ledPin = 9;    // LED connected to digital pin 9

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);
  }
}
```

Activity 5

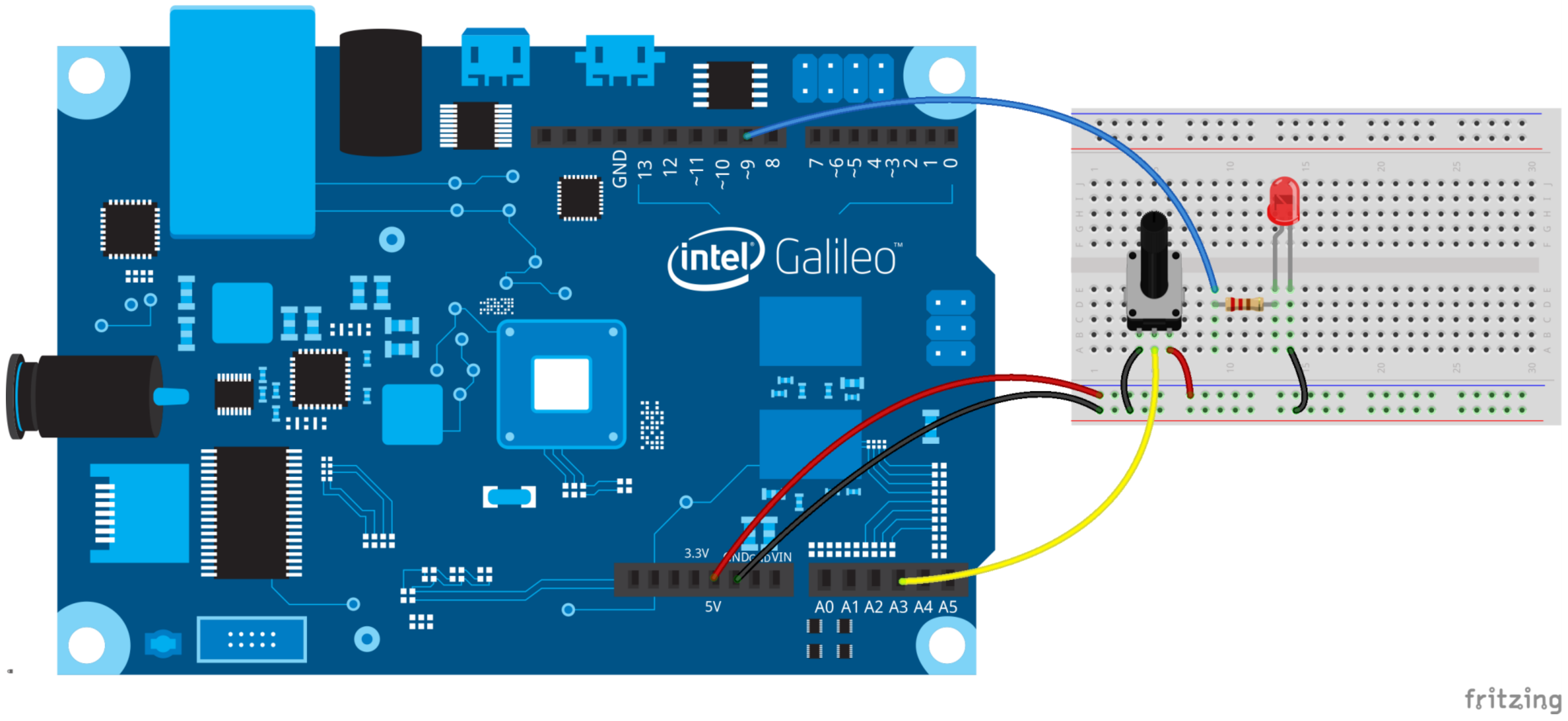
- Use potentiometer to fade the LED via PWM

Potentiometer has 3 pins

- Pin 1: connect to Galileo GND
- Pin 2: connect to Galileo Analog Pin 3
- Pin 3: connect to Galileo 5V

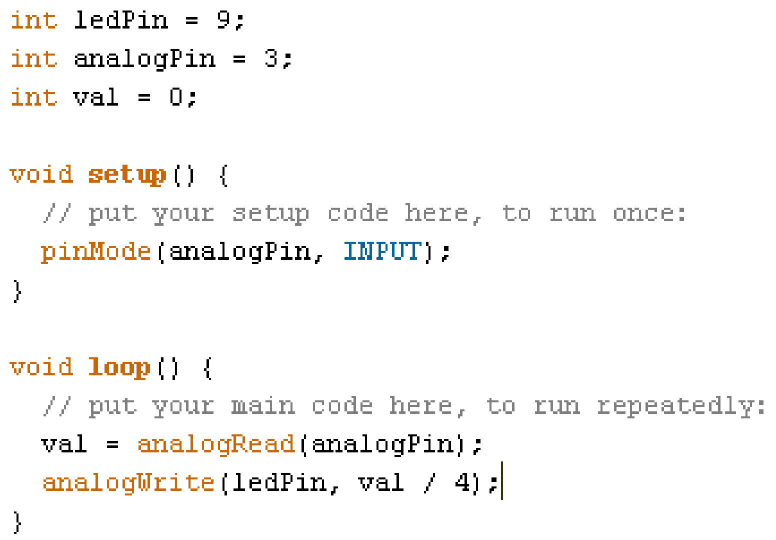


Activity 5 (cont.)



Activity 5 (cont.)

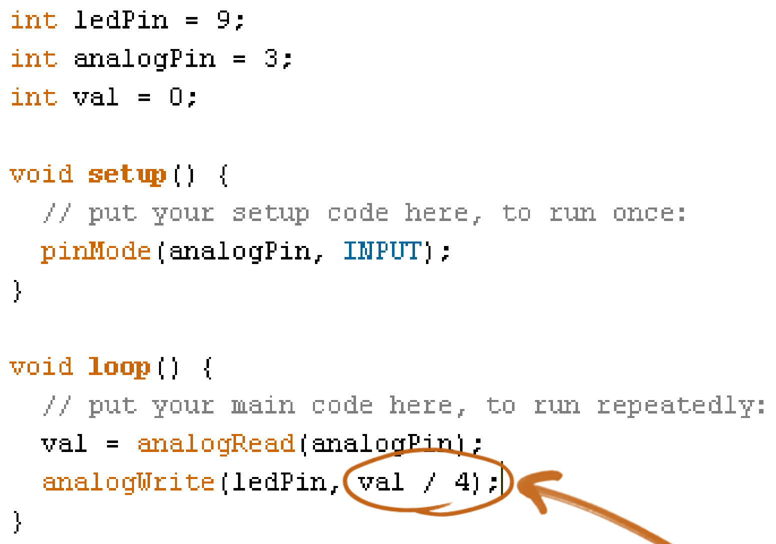
Code Example:



```
int ledPin = 9;
int analogPin = 3;
int val = 0;

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

void loop() {
  // put your main code here, to run repeatedly:
  val = analogRead(analogPin);
  analogWrite(ledPin, val / 4);
}
```

Activity 5 (cont.)

Code Example:



```
int ledPin = 9;
int analogPin = 3;
int val = 0;

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

void loop() {
  // put your main code here, to run repeatedly:
  val = analogRead(analogPin);
  analogWrite(ledPin, val / 4);
}
```

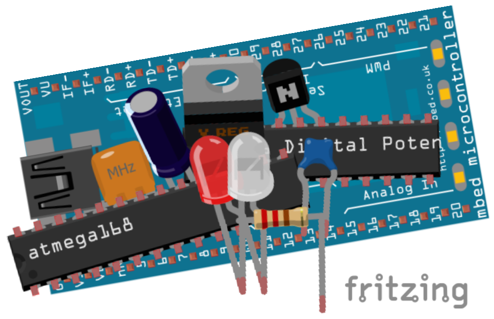
Why divide by 4?

Small Project

- Blink an LED with pushbutton. While pressing, the interval of the blinking should be adjustable by using potentiometer. Once you release (the pushbutton), the LED should turn off.

Small Project (cont.)

- Things to be submitted:
  - Source code (.ino)
  - Fritzing circuit design
- Send it to:
  - hadrihilmi@gmail.com
  - khairolnadzrinsaufi@gmail.com



Small Project (cont)

⚠ WARNING:

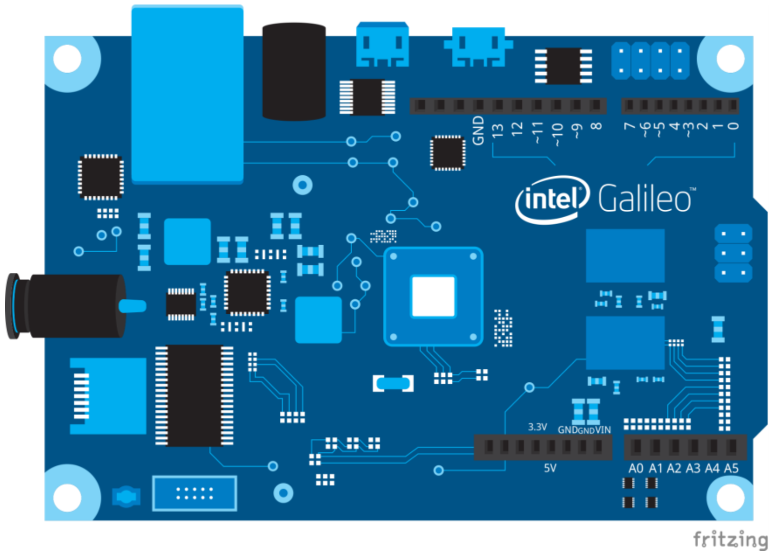
- Source code – CODING STANDARD!
- Put all documents together and compress it!
- Name your compressed file as follow:

  <matrix-no>_smallproject.<zip/rar/tar.gz/7z>

  e.g: 95960_smal
- Send with email's subject

  "[CPT211] Small Project - <matrix-no>"
- Submission due: Sept 17th 2014 before 2300

END



Intel Galileo® : What will you make?

What is coding standard?

- Proper indent
- Commenting
  - single line comment vs block comment
- Describe your code
  - File, author, code description
  - Versioning
- Git your code (if applicable, it’s a good practice tho)