LANKA NIPPON BIZTECH INSTITUTE

2024/3/4-2024/3/5

Workshop

Kyushu Institute of Technology

Kazuaki Tanaka

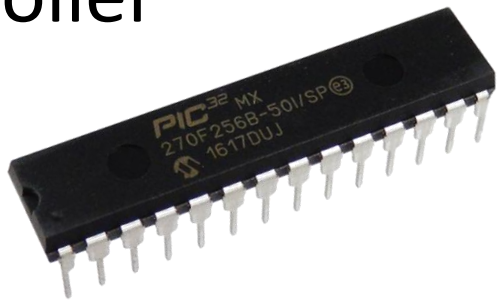
About Me

- Kazuaki TANAKA（たなか かずあき）
- Kyushu Institute of Technology, JAPAN
  Associate Professor
- Ph. D in Information Science

- Research topics: embedded systems, IoT, mruby, wireless communication

Research topic:
mruby and mruby/c

- Ruby language for small devices
- OO Programming Language
- Target: one-chip microcontroller
  PIC, ESP32, STM32, etc.
- Small memory footprint
  minimum 20KB



- Open-Source Software

`https://github.com/mruby/mruby`

`https://github.com/mrubyc/mrubyc`

What is Ruby language?

- Object Oriented Programming Language
  - Scripting language
  - Web application, Ruby on Rails
- Matsumoto has said that Ruby is designed for programmer productivity and fun.

Ruby code example:

```
led1 = GPIO.new(0)

while true do
  led1.write 1
  sleep 0.5
  led1.write 0
  sleep 0.5
end
```

Introduction

This workshop

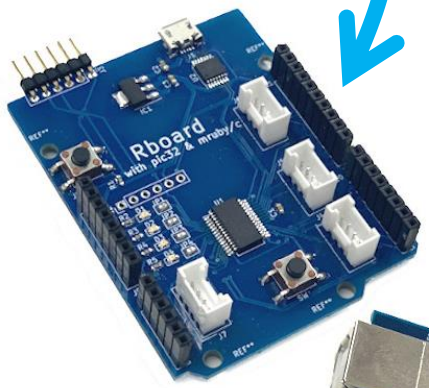
- Experience in embedded software development.

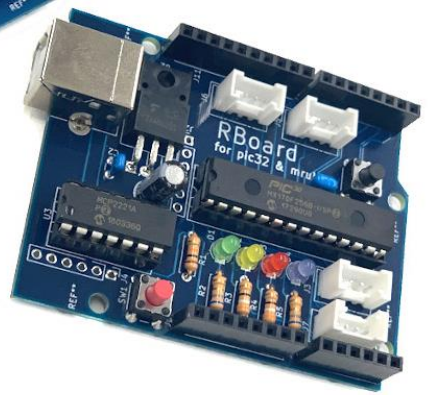
- mruby programming
- Controlling electronic circuits
- Brightness sensor

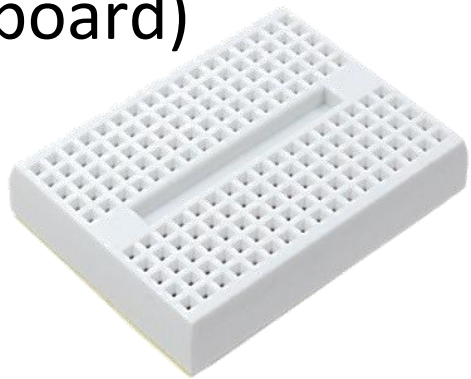
Simple Circuit

Electrical circuit

- Micro controller board
- Electrical parts and circuit
  - LED, sensors
  - Breadboard (prototyping board)







Run the first program

- Coding
- Compile and Download
  Download= transfer compiled program into microcontroller board
- Execute

Blinking LED

- program1

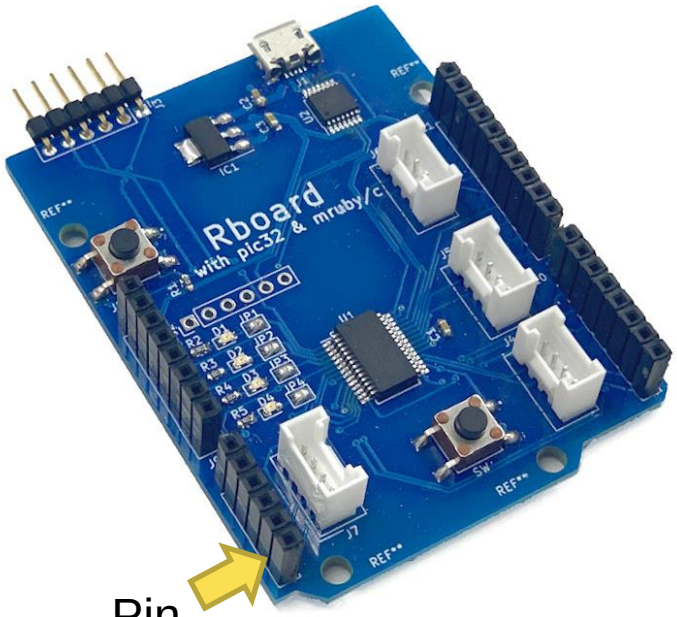
```
led1 = GPIO.new(0)

while true do
  led1.write 1
  sleep 0.5
  led1.write 0
  sleep 0.5
end
```

- First, run this program

Microcontroller board

- RBoard
- Features
  - Execute program
  - Store data
  - I/O of Pins
  - Input= Measure voltage
  - Output= Set voltage
    HIGH(1) or LOW(0)



Onboard LEDs

- 4 LEDs, each LED is connected to Pin
  - LED 1: Pin 0
  - LED 2: Pin 1
  - LED 3: Pin 5
  - LED 4: Pin 6

- Set pin voltage to HIGH, then turn on LED
  Set to LOW, then turn off LED

Program

- GPIO： General Purpose Input and Output

```
led1 = GPIO.new(0)

while true do
  led1.write 1
  sleep 0.5
  led1.write 0
  sleep 0.5
end
```

Pin0 is assigned to variable led1

write: set voltage

sleep: stop execution

Iteration

Exercises

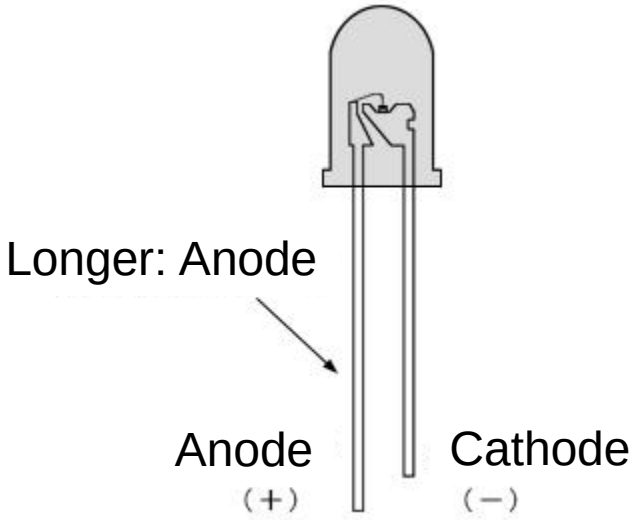
- Blink two LEDs alternately

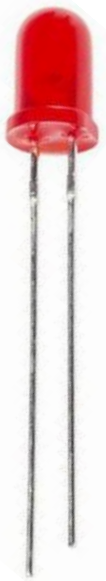
- Blink four LEDs

LED 1: Pin 0
LED 2: Pin 1
LED 3: Pin 5
LED 4: Pin 6

LED

- Light Emitting Diode





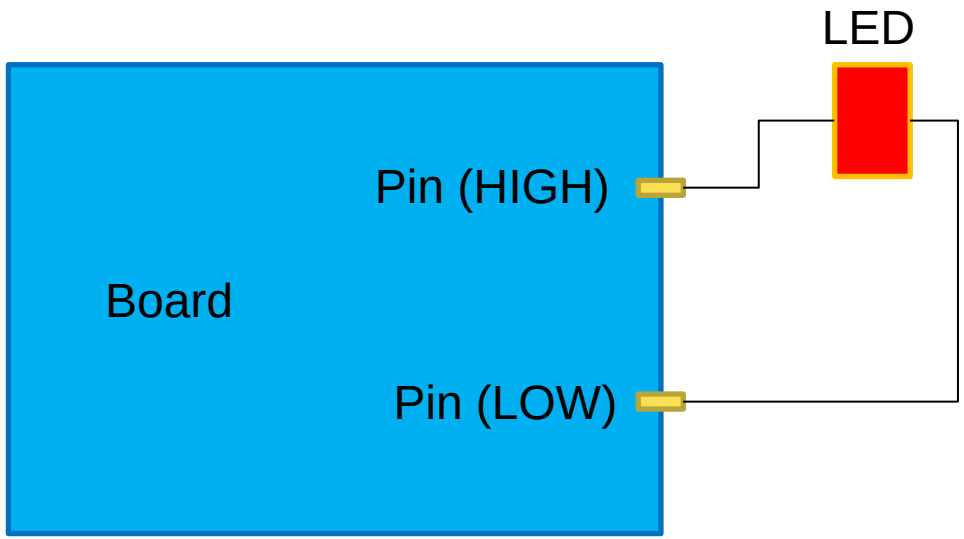
Current flows from Anode to Cathode

LED Circuit

- Make LED on
  - Current flowing through the LED

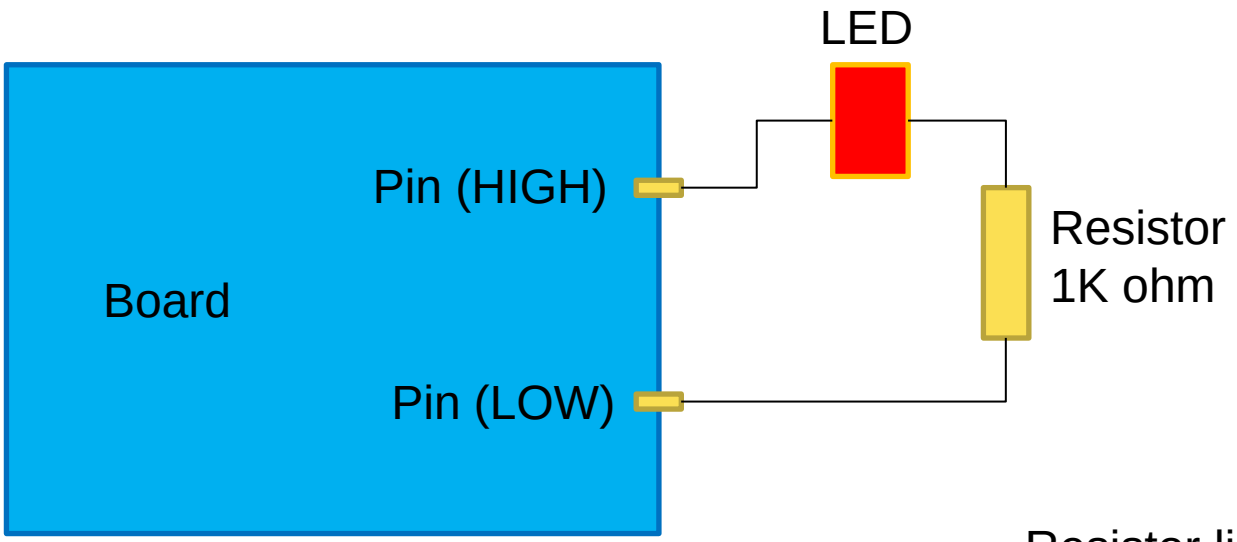
- Rboard feature
  - Set voltage of Pin
  - Set HIGH or LOW

Wrong circuit



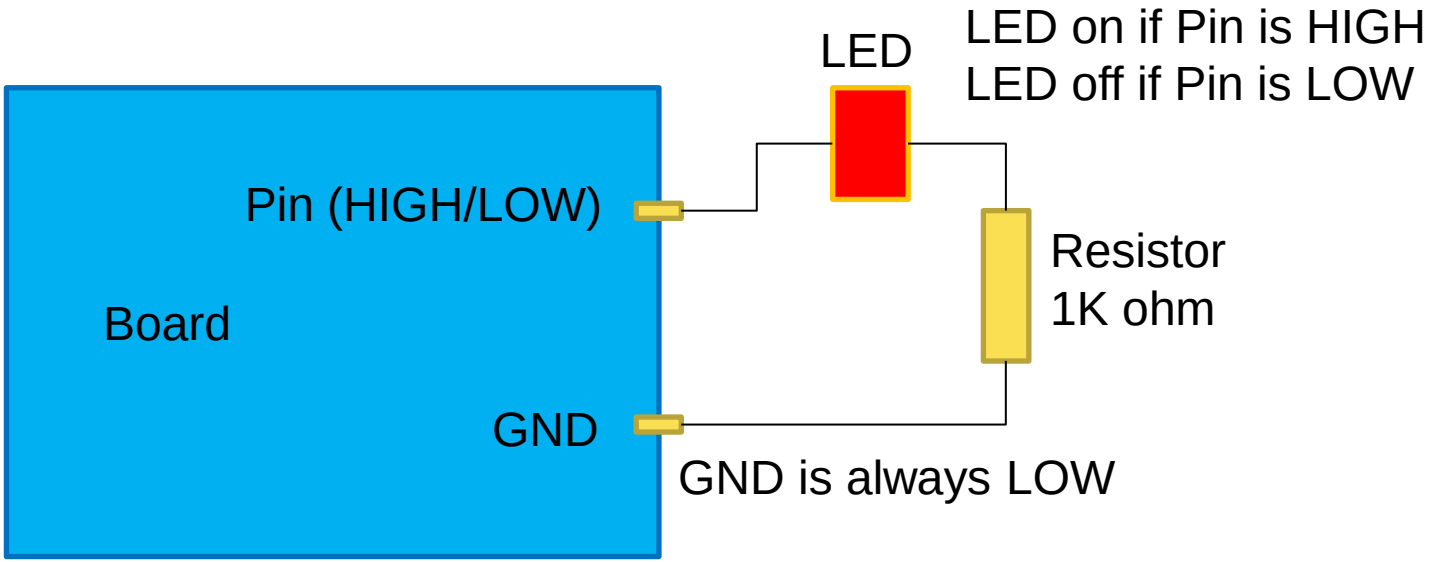
The resistance of LED is almost 0.
Many current flows,
cause damages to LED

LED circuit



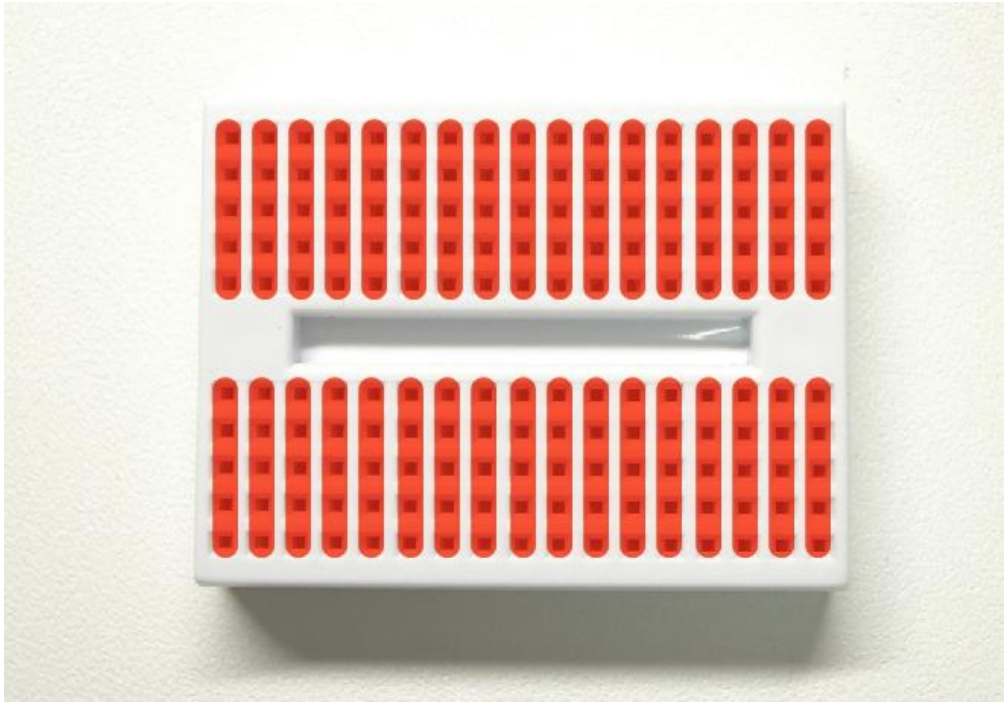
Resistor limits current flowing.

LED circuit


LED
LED on if Pin is HIGH
LED off if Pin is LOW
Pin (HIGH/LOW)
Board
Resistor
1K ohm
GND
GND is always LOW

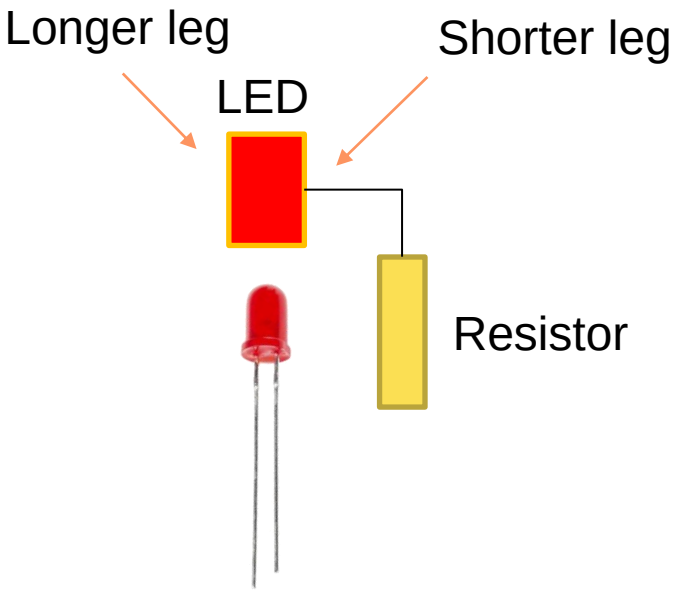
Breadboard
(Prototyping board)

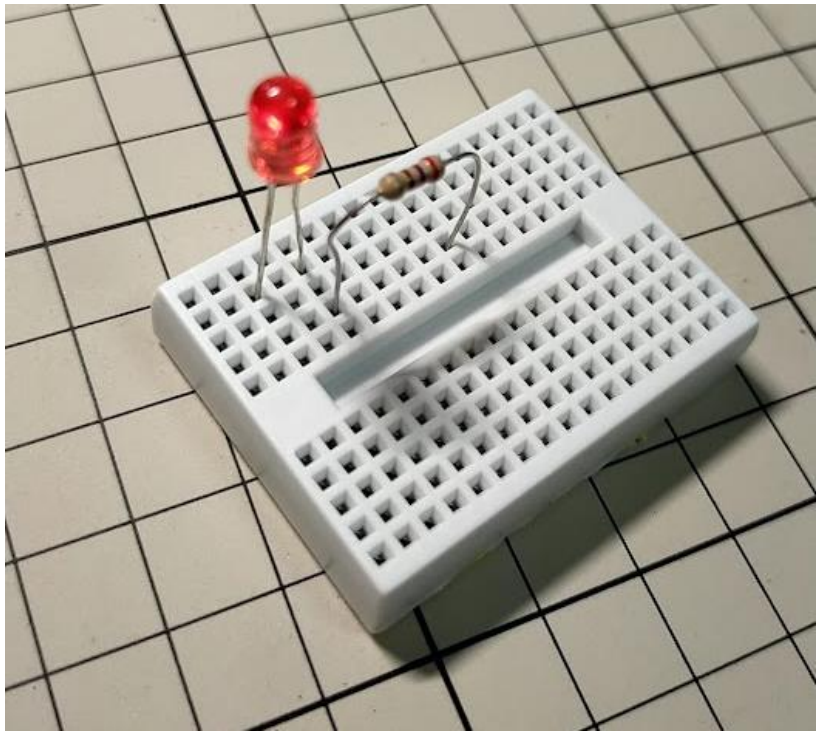
- Red holes are connected



Using breadboard

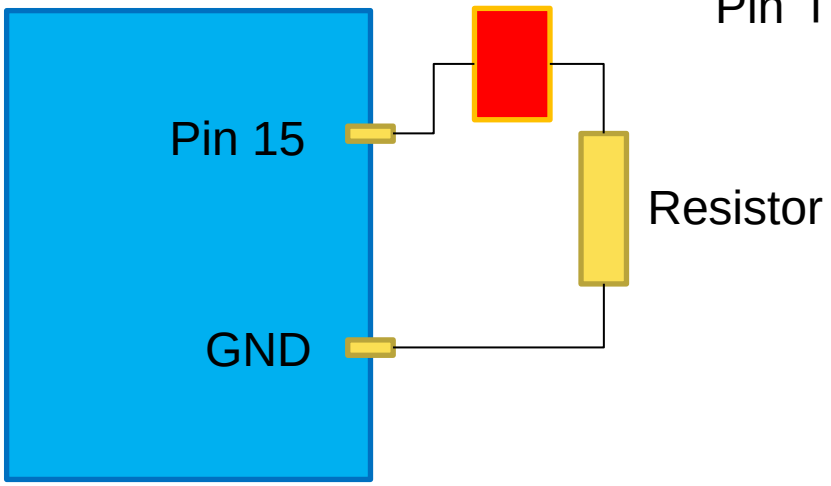
- Connecting a LED and a resistor

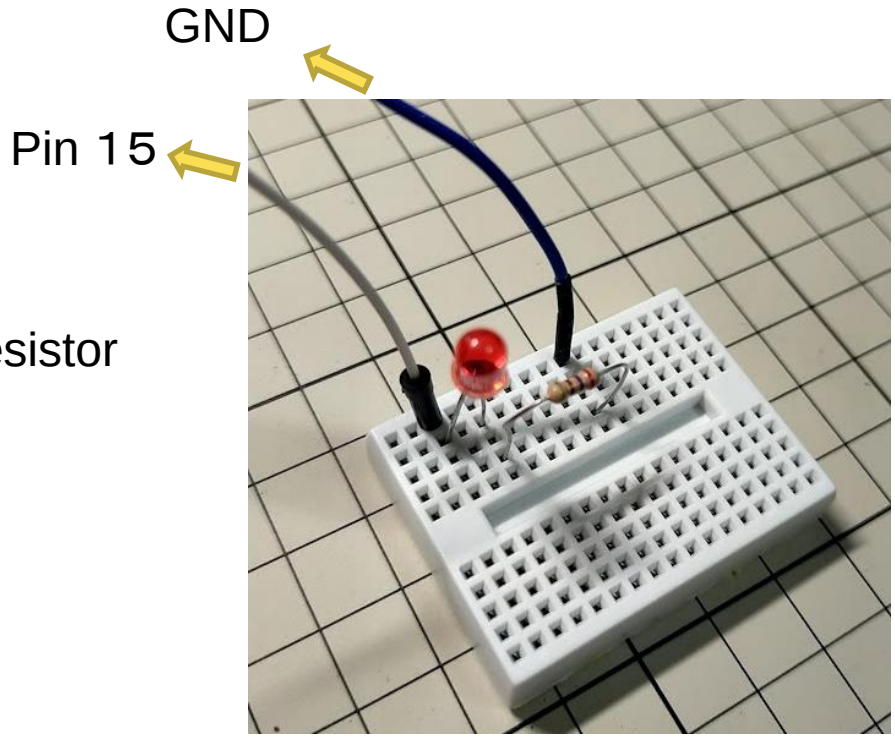




Connecting

- Use jumper wires





Implement your code

```
led1 = GPIO.new(15)
led1.setmode(0)

while true do
  led1.write 1
  sleep 0.5
  led1.write 0
  sleep 0.5
end
```

Pin 15

Set pin mode
0: Output voltage
1: Input voltage

PWM

PWM
Analog output

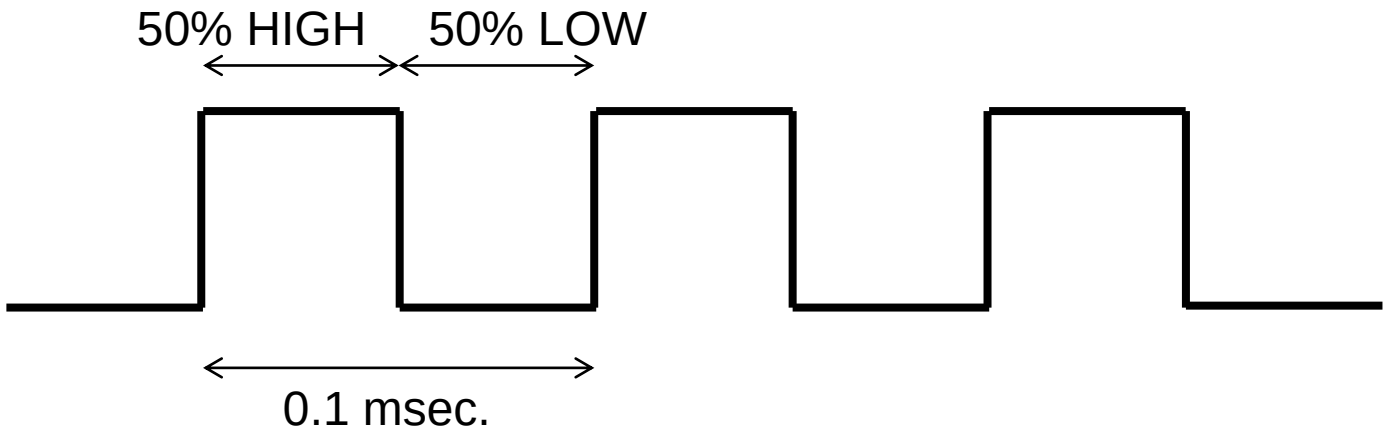
- Output of microcontrollers: Digital
  - HIGH or LOW voltage
  - In LED, ON or OFF

- We want: Analog
  - Change voltage
  - In LED, ON…bright…dark…OFF

Pseudo analog output

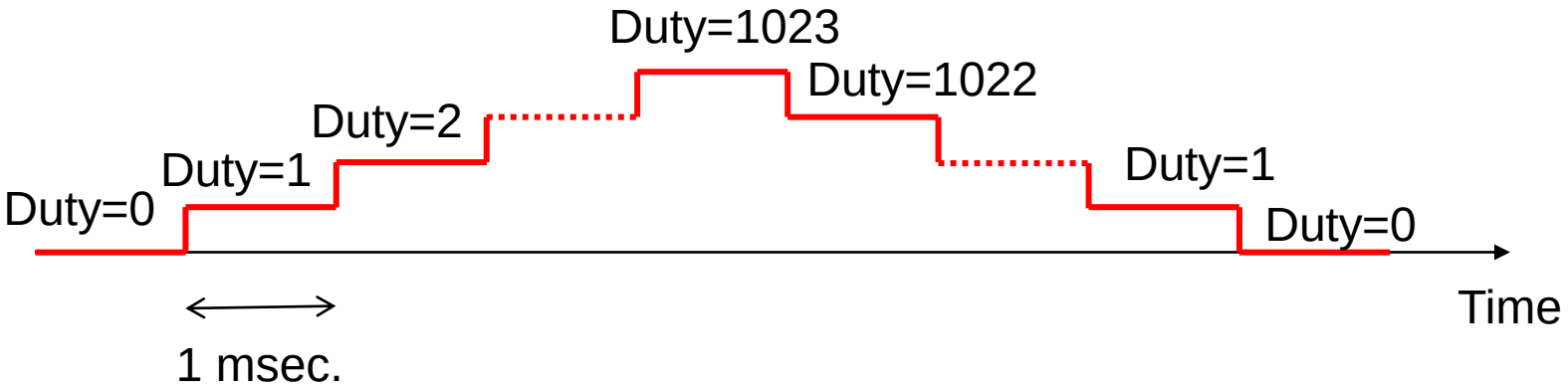
- PWM, pulse width modulation

- Control HIGH and LOW rapidly
  - 1000 times a second



Change brightness

- Change brightness => Change duty ratio



PWM example

```
led1 = PWM.new(15)
led1.frequency 10000

while true do
  for i in 0..1023 do
    led1.duty i
    sleep 0.001
  end
  for i in 0..1023 do
    led1.duty 1023-i
    sleep 0.001
  end
end
```

duty: 0 to 1023
0: 0% HIGH
1023: 100% HIGH

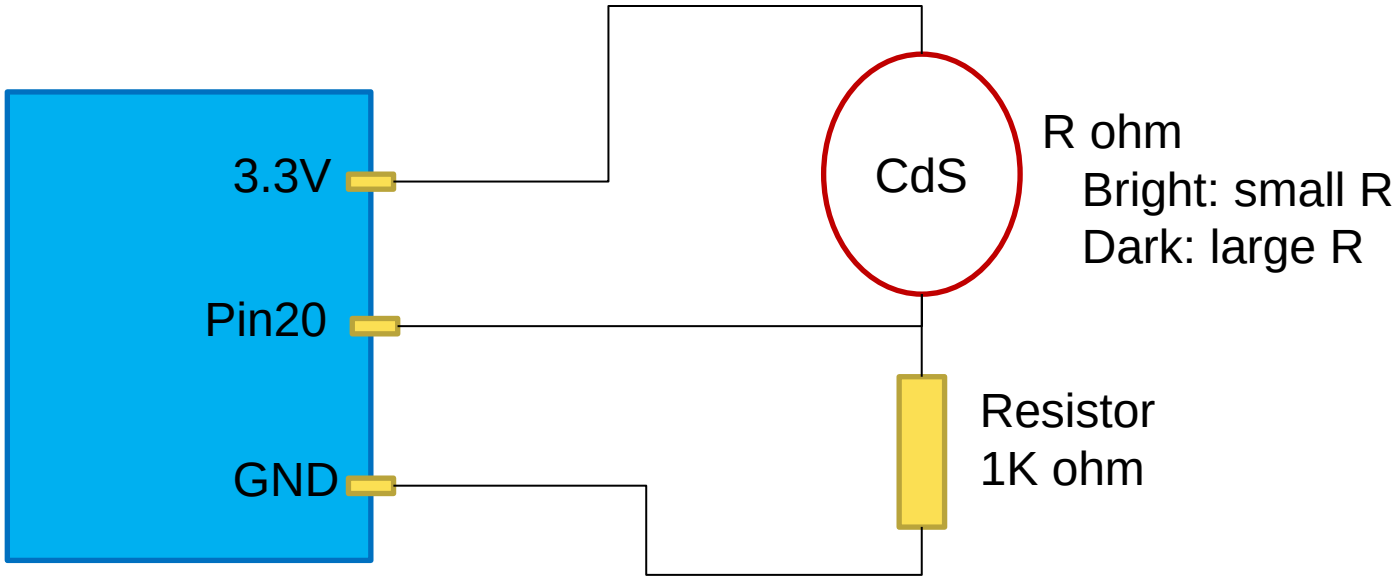
Brightness Sensor

CdS

- CdS: Resistance changes with varying light levels.

- Microcontroller can detect VOLTAGE

CdS circuit

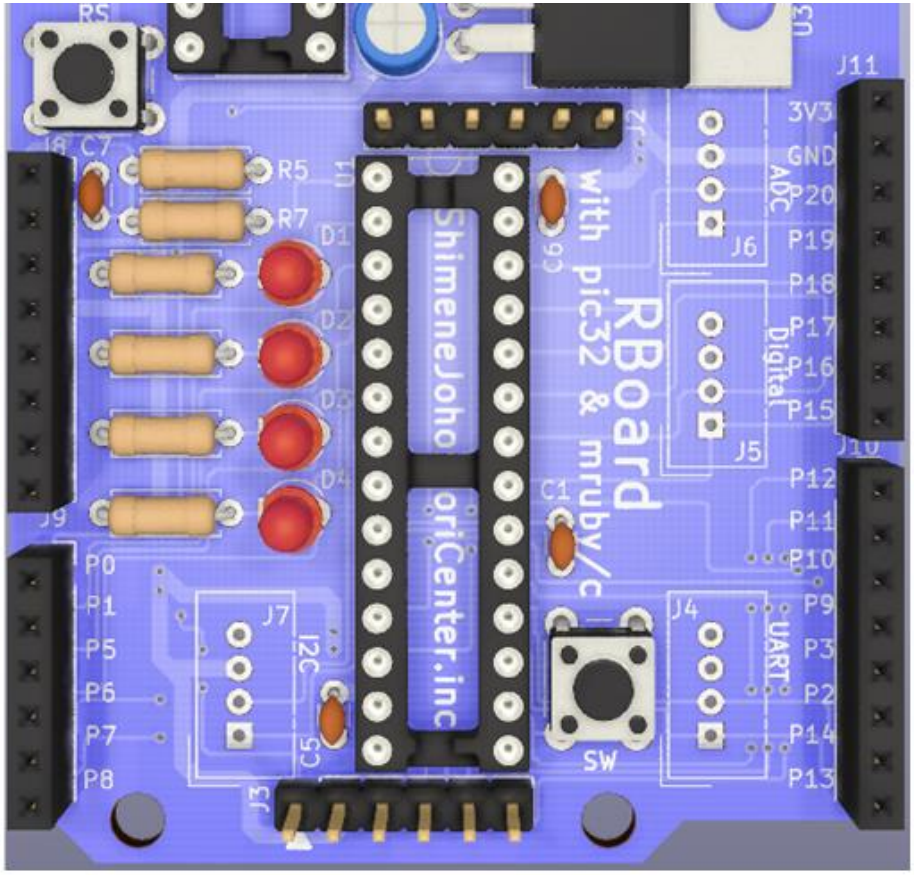


$$Pin20\ voltage = 3.3 \times \frac{1K}{1K + R}$$

3.3V

Use this Pin

		NC
		3.3V
		RST
		3.3V
		5V
		GND
		GND
		5V
LED1		P0
LED2		P1
LED3		P5
LED4		P6
I2C	SDA	P7
I2C	SCL	P8



3.3V		
GND		
P20		ADC
P19		ADC
P18		
P17		
P16		Digital
P15		Digital
P12		SW
P11		
P10		
P9	U1:TX	
P3		
P2		
P14	U2:TX	UART
P13	U2RX	UART

ADC

- ADC: Analog Digital Converter
  - Read voltage at Pin20
  - Voltage v
    0 < v < 1.5

```
adc = ADC.new(20)

v = adc.read
```

CdS example

```
led1 = GPIO.new(0)
adc = ADC.new(20)

while true do
  v = adc.read
  if v<0.5 then
    led1.write 1
  else
    led1.write 0
  end
  sleep 0.1
end
```

Pin20 for ADC

Read voltage from Pin20

Voltage: 0 < v < 1.5

Exercises

- Brightness sensor
- LED changes by brightness
  - Green: bright
  - Yellow: a little dark
  - Red: dark
- adc.read returns around 0.2 in dark, and returns around 0.9 in bright

Conclusion

Embedded systems

- Software and Hardware integration

- Software controls hardware
  - Requires both software and hardware knowledge