

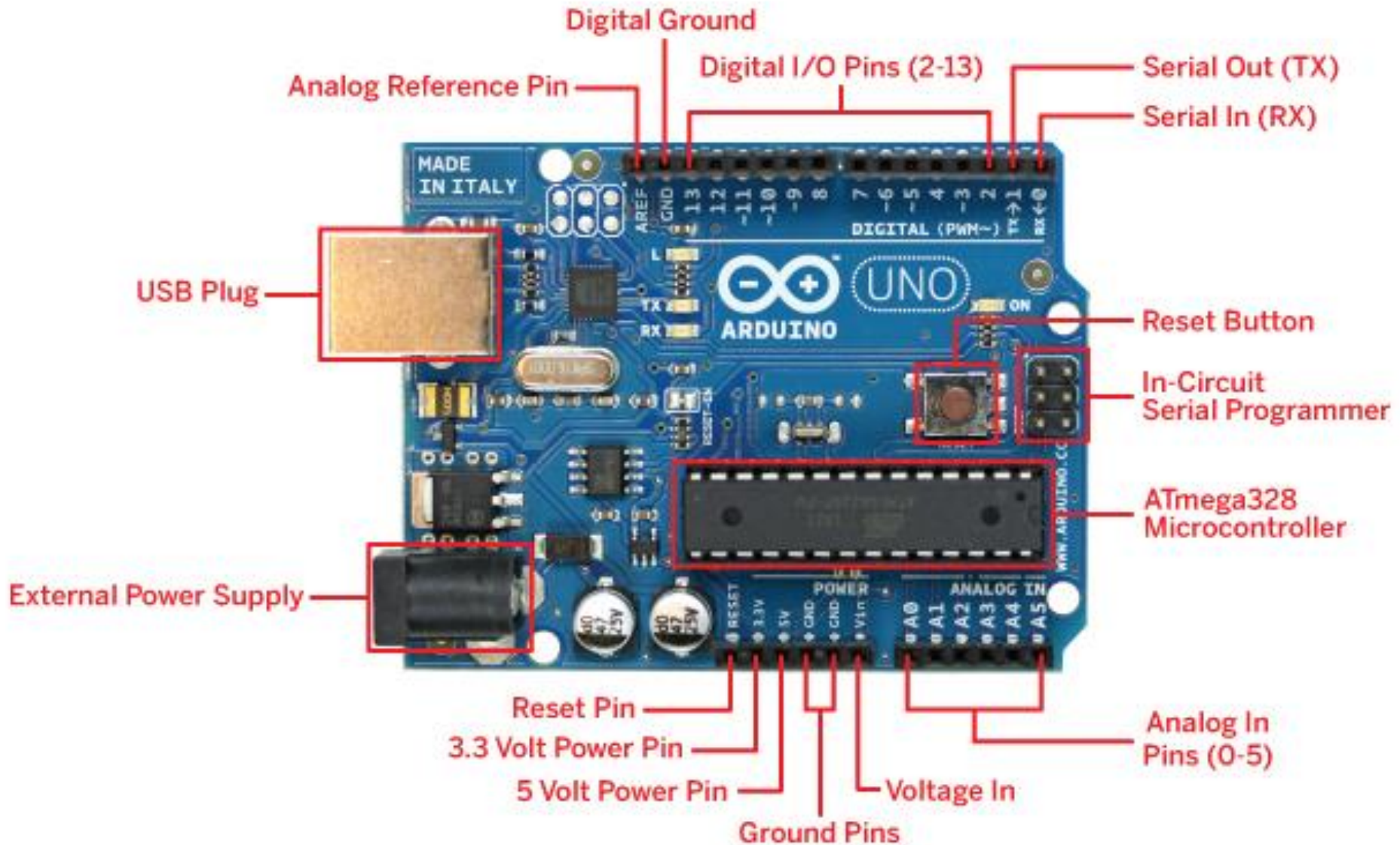
# **Using Single-board Computer in Teaching Programming at Junior Secondary Level**

By Chu K F  
26 June 2018

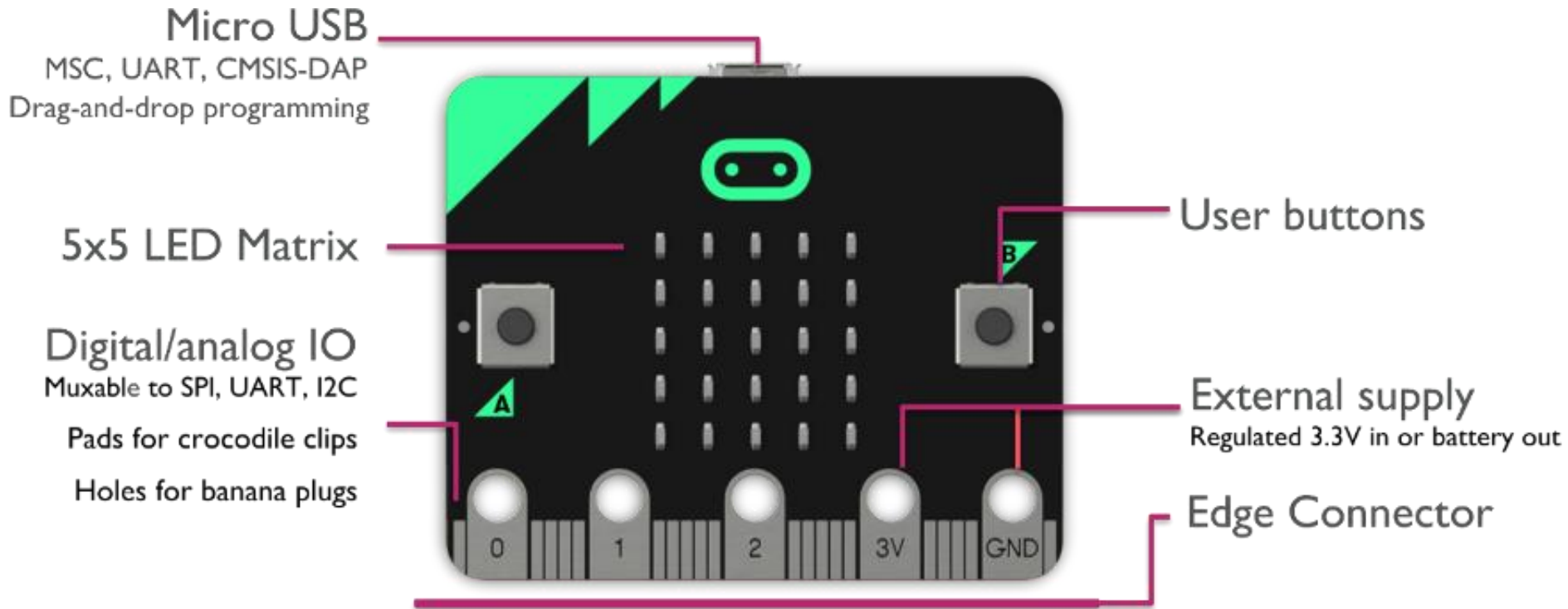
# Some Questions to Consider

- Hardware and Software
  - Balance between Circuit Building and Program Coding
- Choice of microcontroller
  - Arduino or BBC micro:bit
- Development tool:
  - Arduino IDE / Python Editor or Visual Programming
- Duration of each lesson (single / double period)
- Task-based or Project-based
- etc.

# Arduino UNO

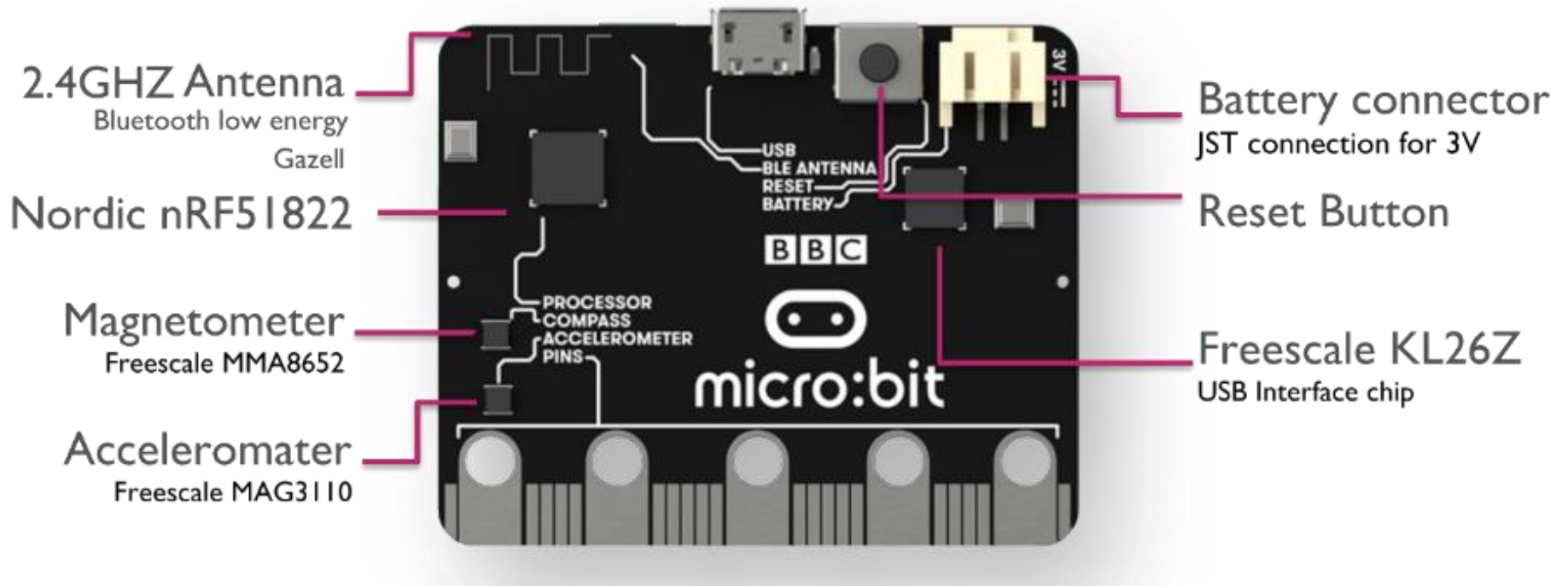


# BBC micro:bit





# BBC micro:bit



# Learning Components

- Basic knowledge
- Guided tasks
- Self-exploration project
  - Students can apply the skills they learnt in the guided tasks to complete simple project

# Sample Projects

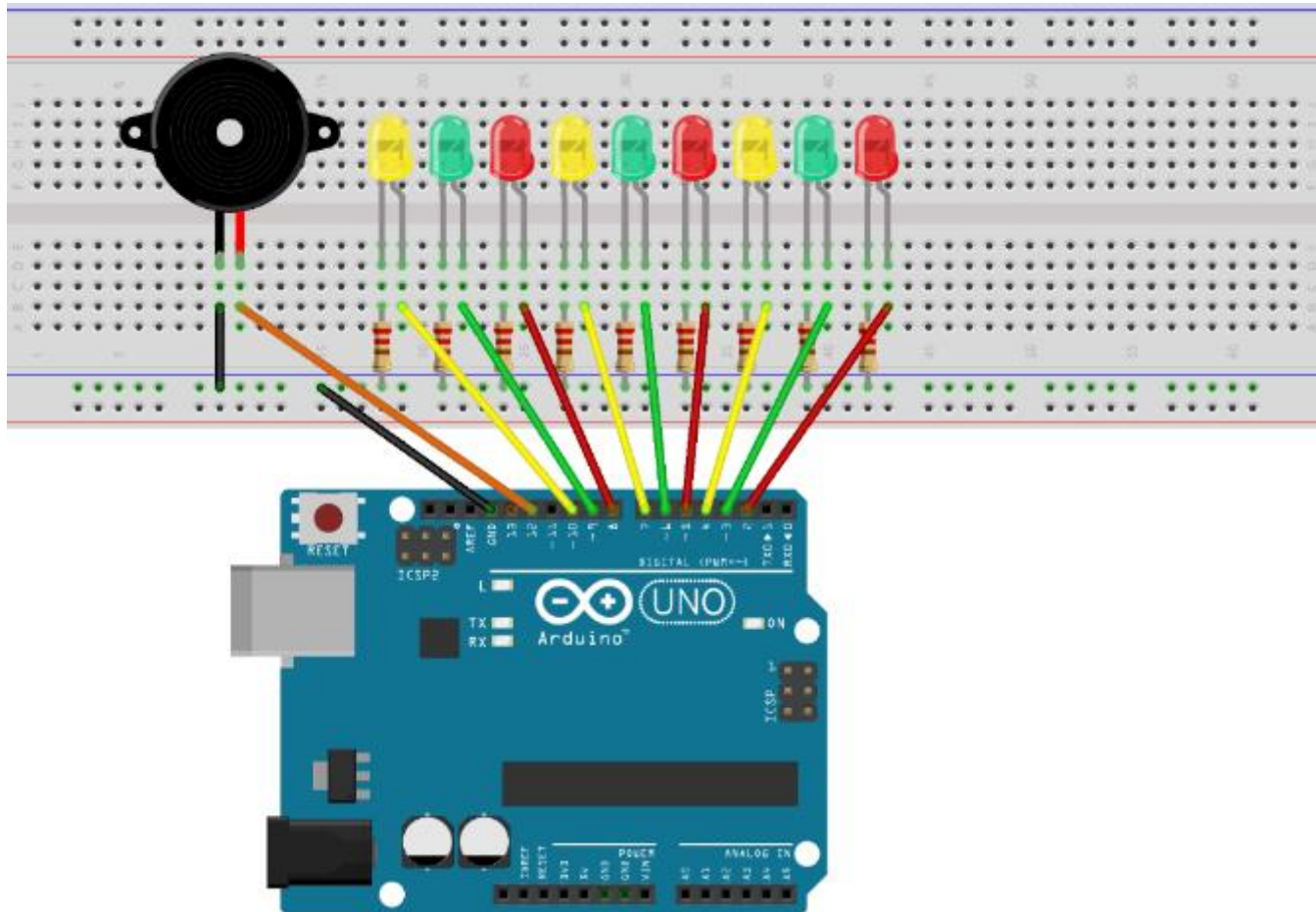
1. A light show with LEDs and Buzzer
2. A simple interactive game using push-buttons, LEDs and buzzer.
3. A light-control music box
4. A smart fan

# Project 1

A Light Show with LEDs and Buzzer

# Project Requirements

- Designing a light show by using a single-board microcontroller, LEDs and buzzer.

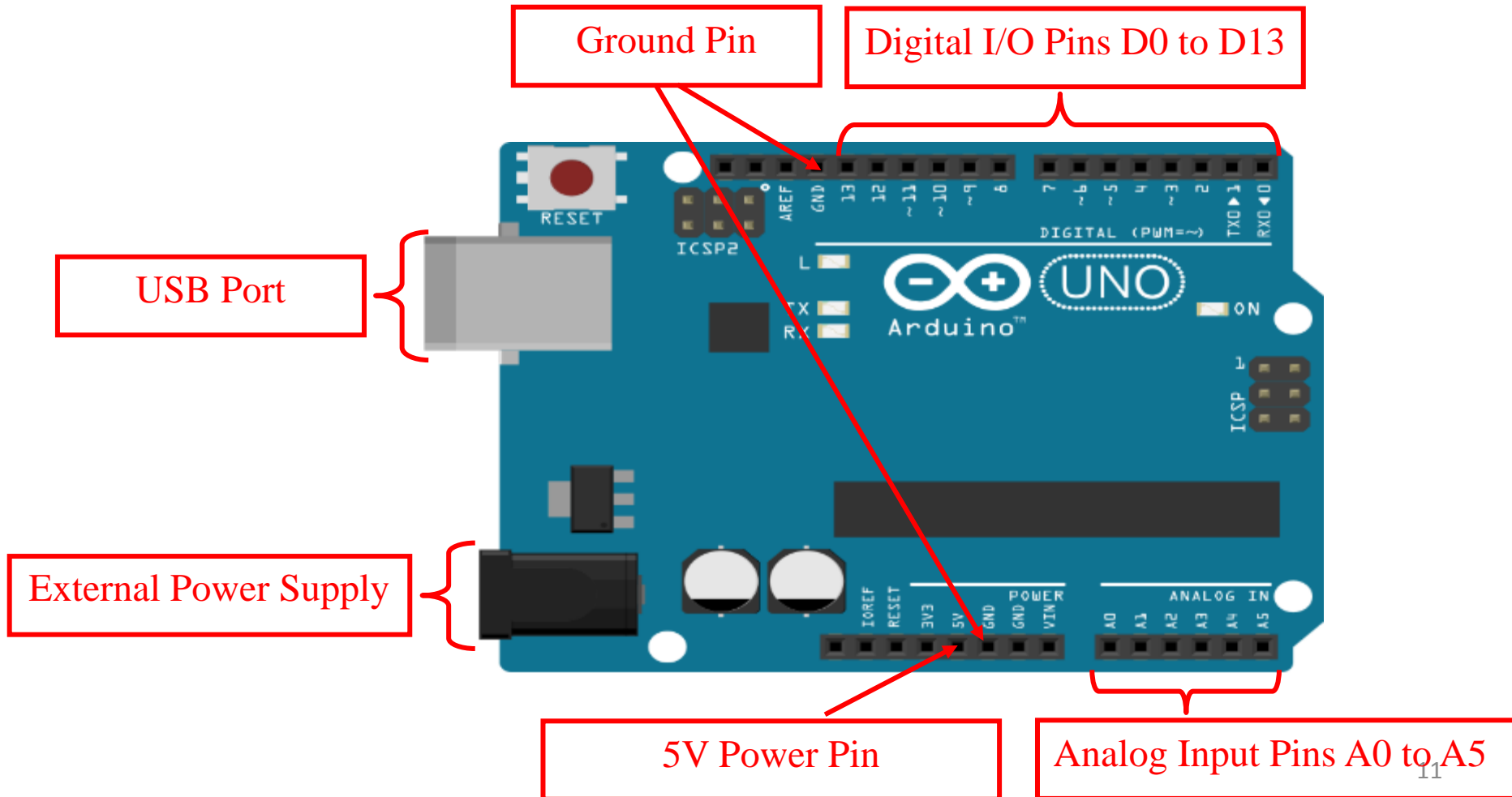


# Knowledge and Skills

- Construct simple circuits by using single-board microcontroller, breadboard, LEDs, buzzers, resistors, etc.
- Create simple programs by using visual programming IDE
  - definite loop
  - indefinite loop
  - infinite loop
  - selection statements
  - etc.

# Basic Knowledge

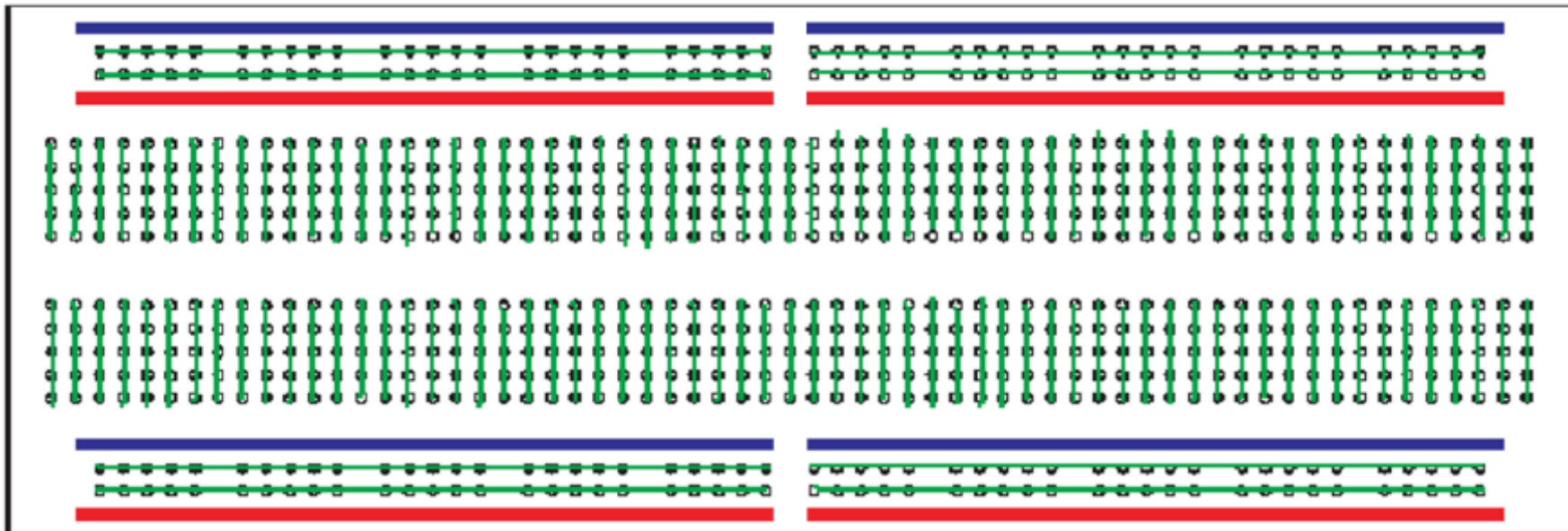
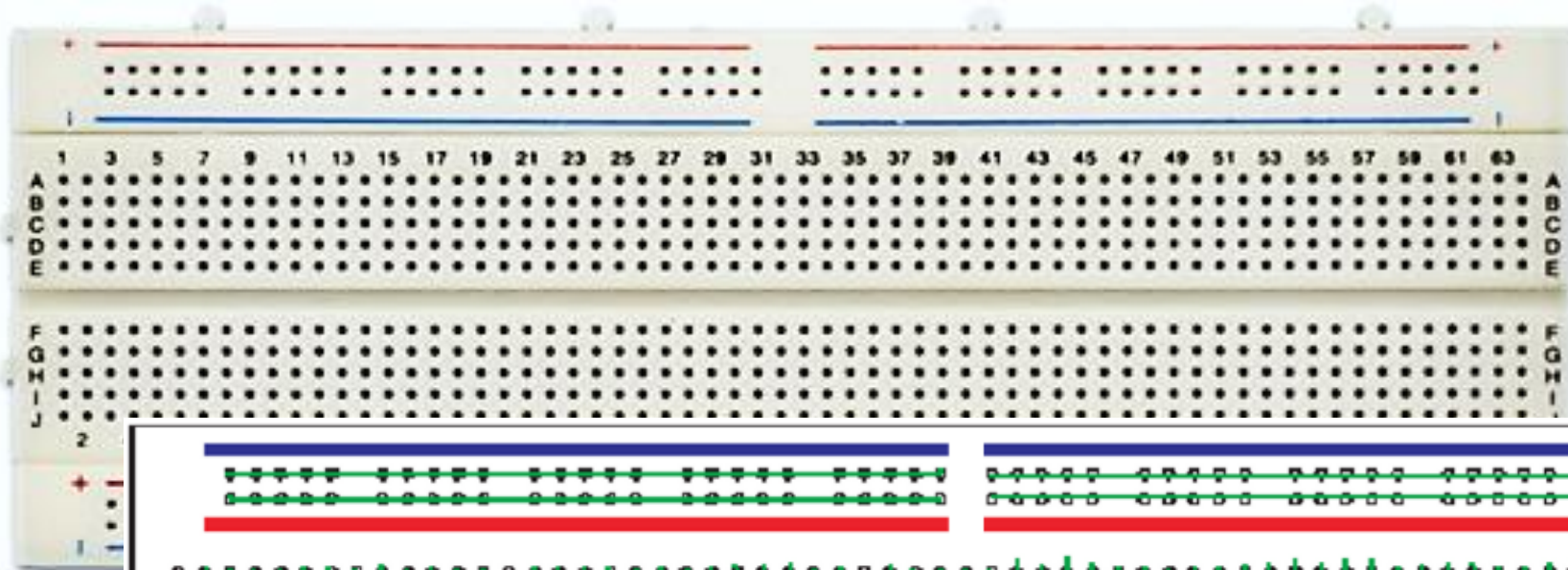
- What is Arduino?





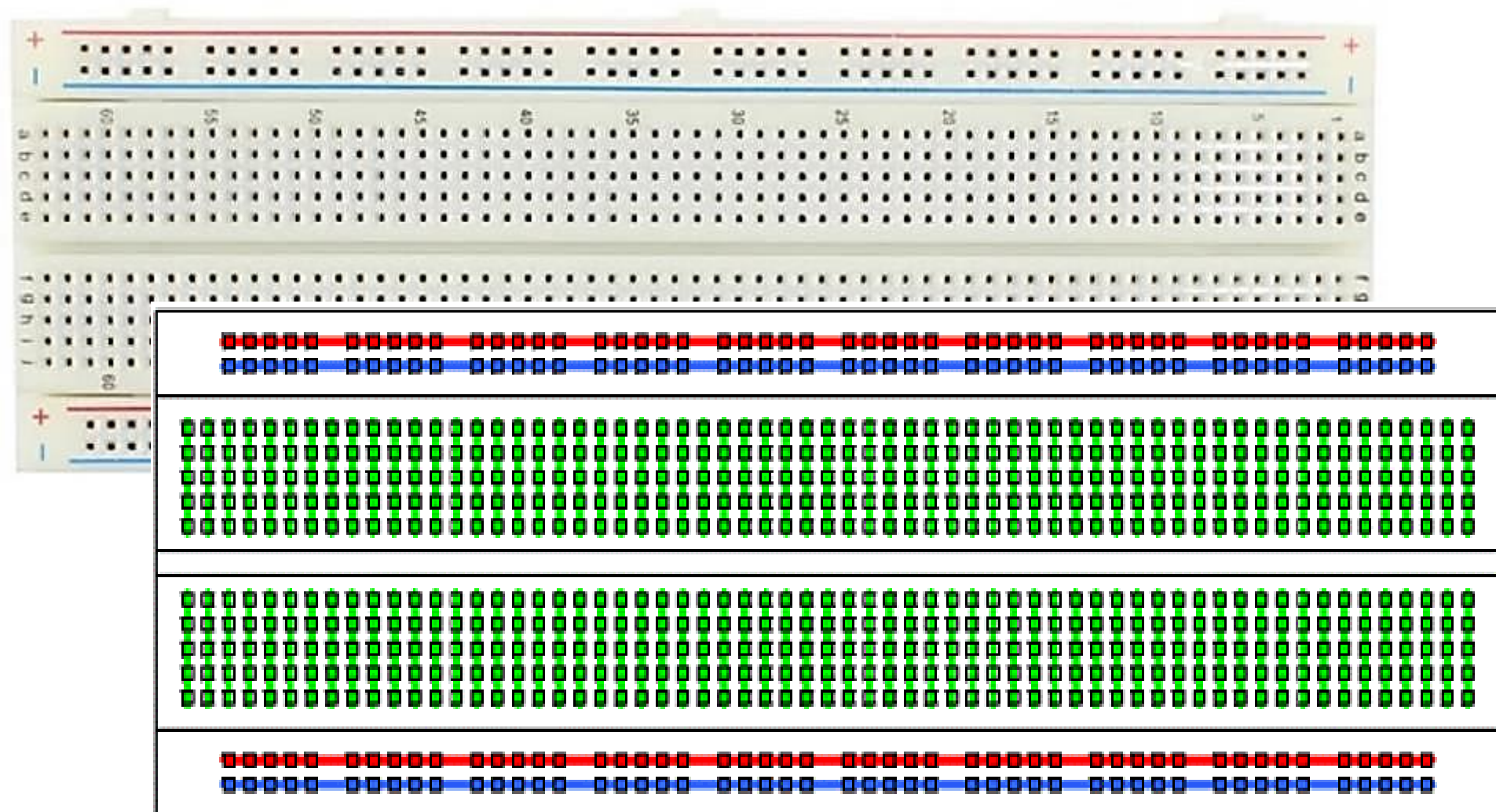
# Basic Knowledge

- Breadboard (different types)



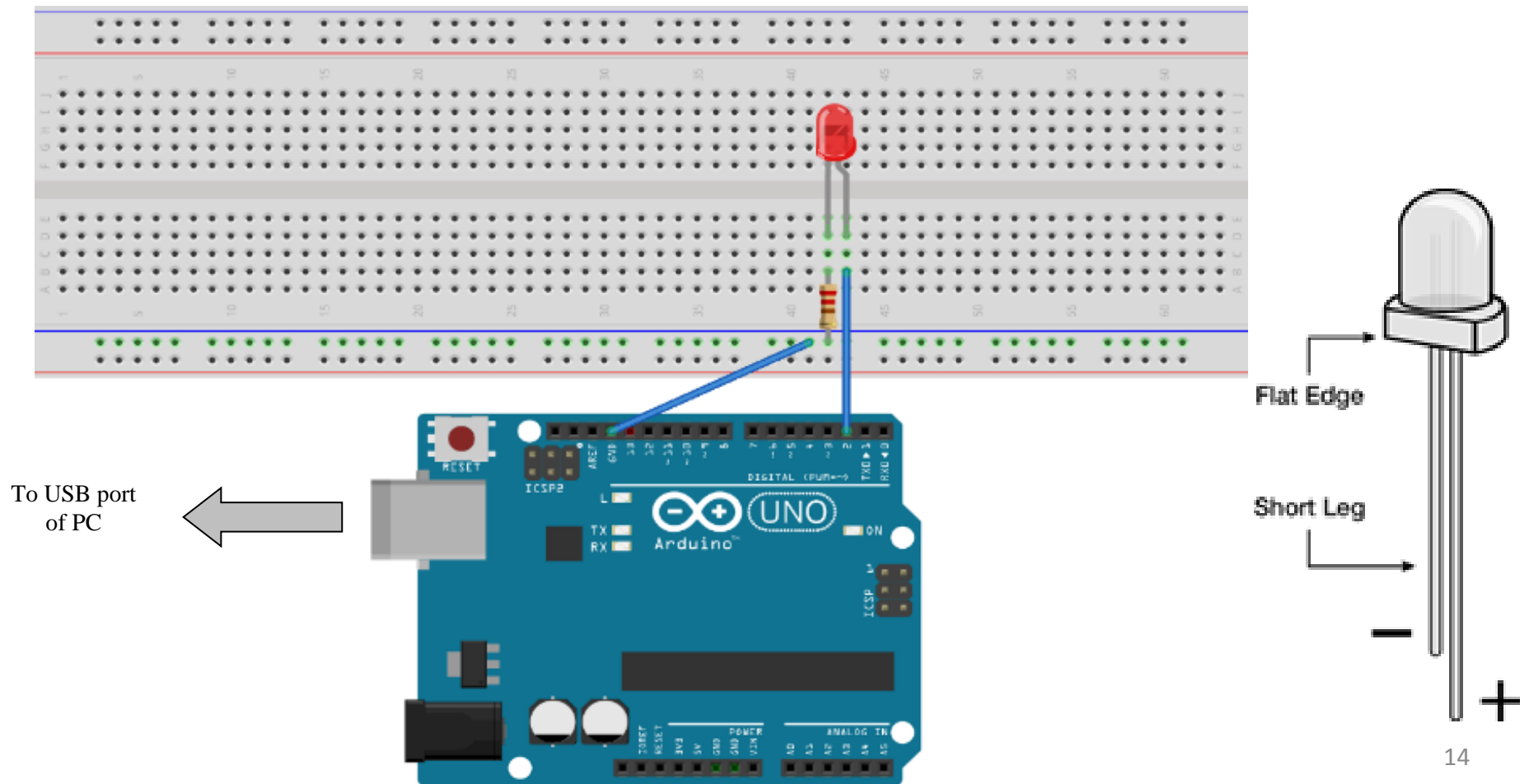
# Basic Knowledge

- Breadboard (different types)



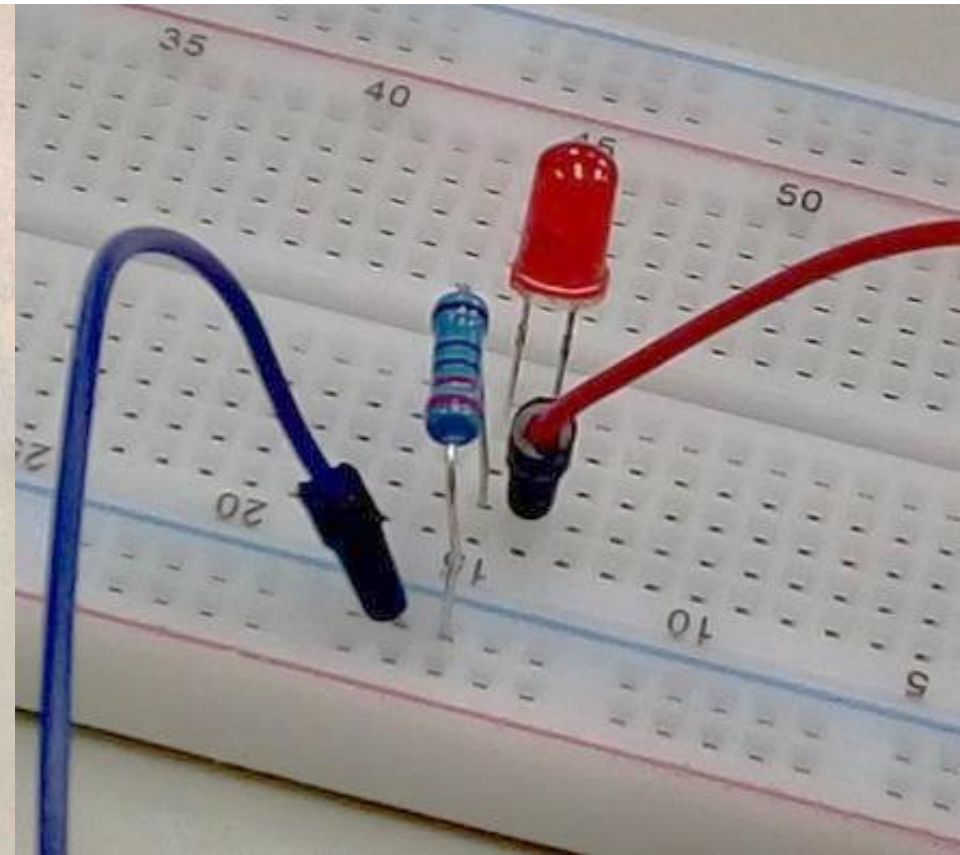
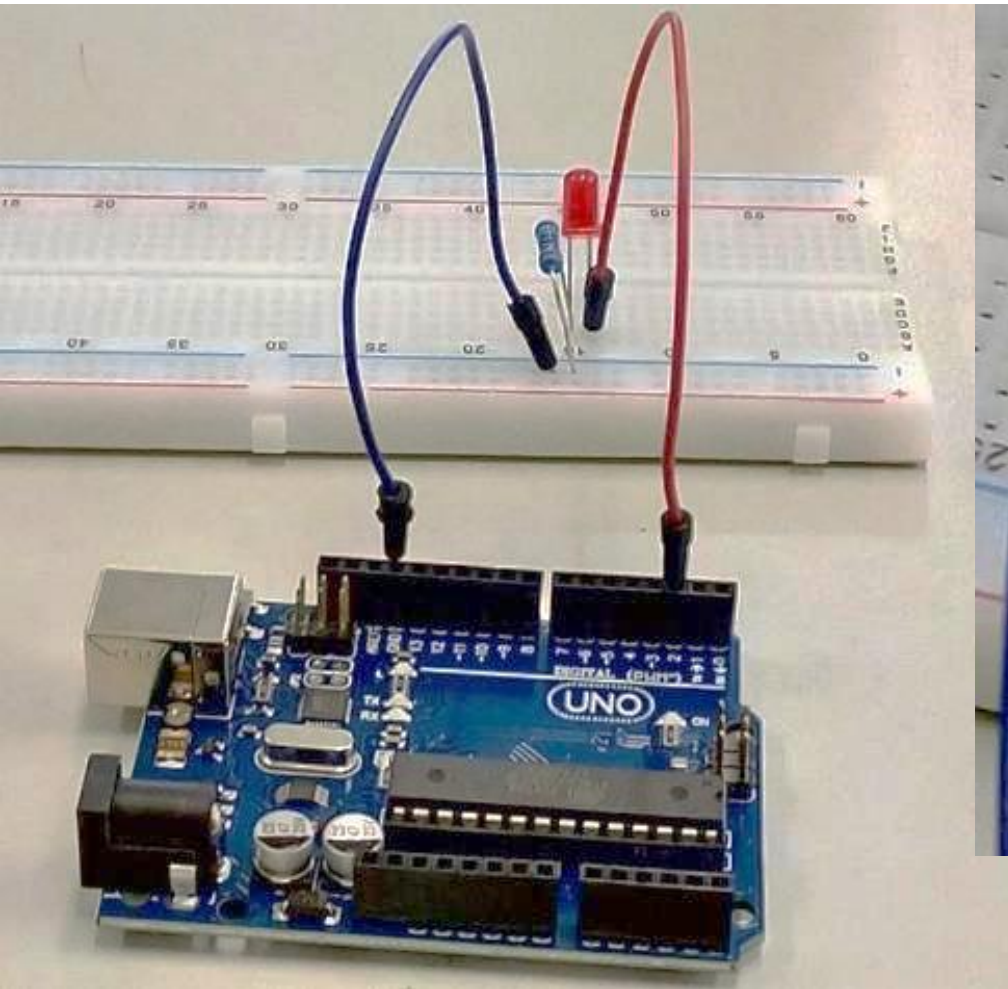
# Basic Skills

- Construction of Circuit (Circuit Diagram)



# Basic Skills

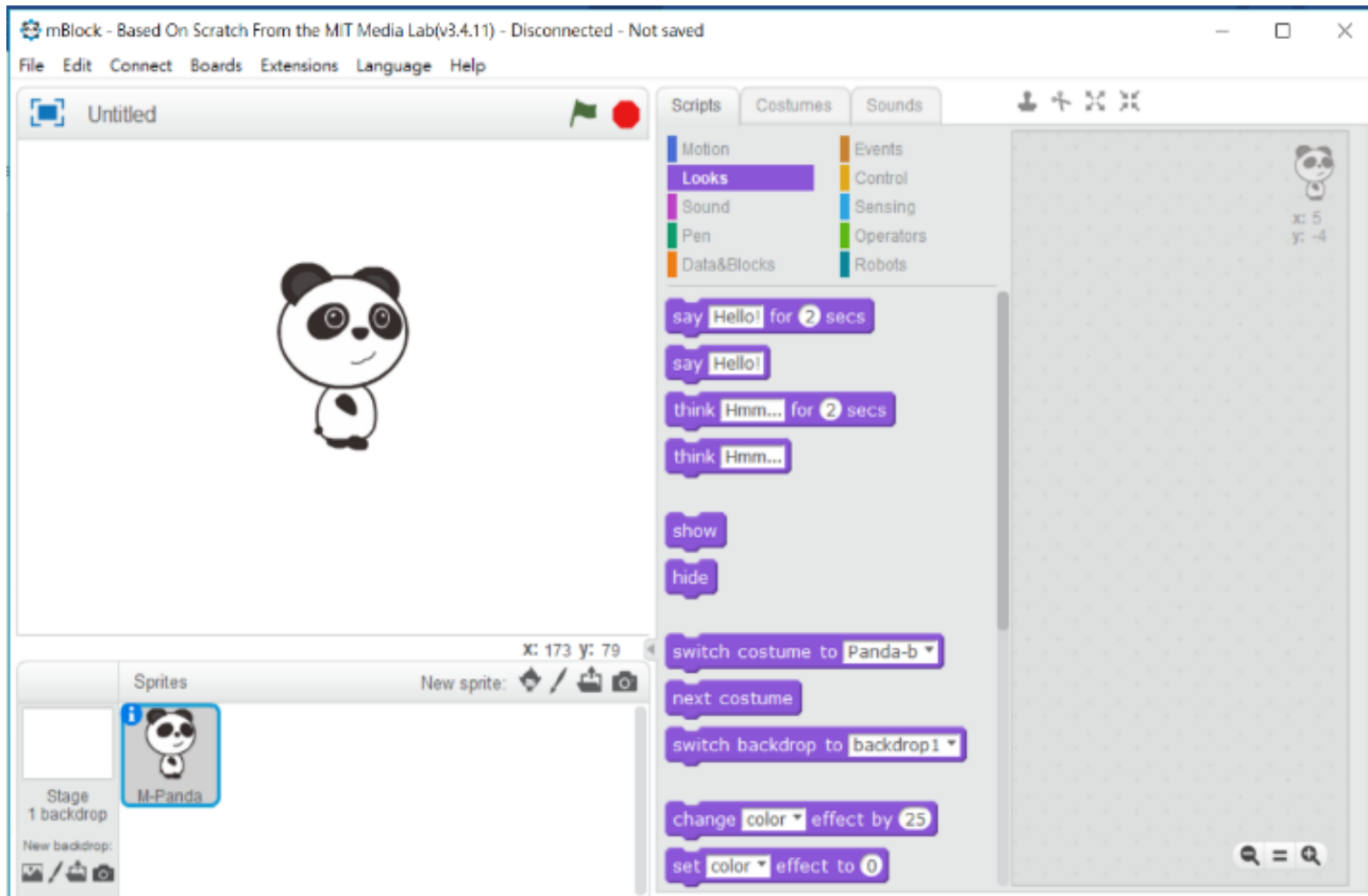
- Construction of Circuit (Photos)





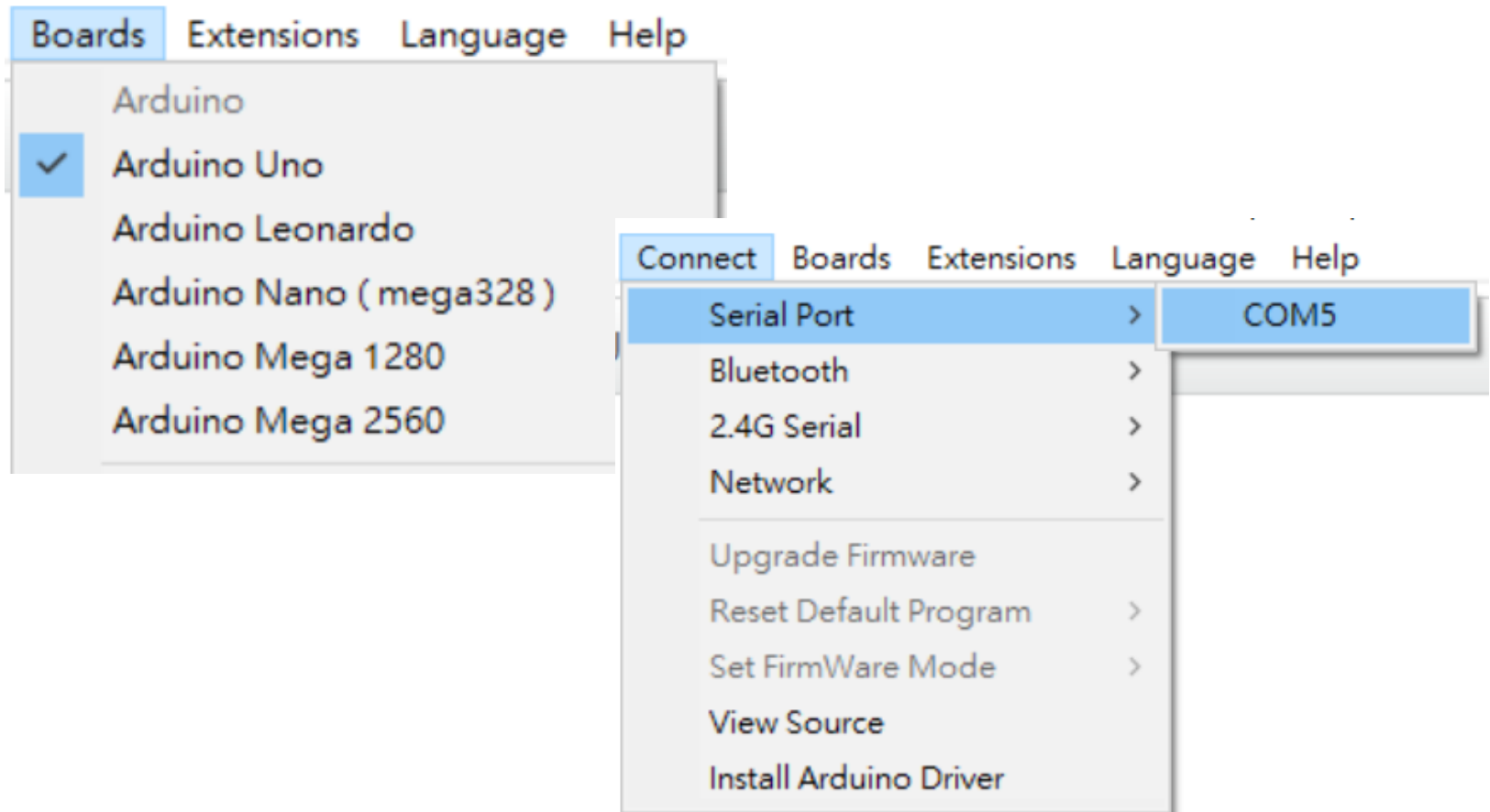
# Knowledge and Skills

- Program Coding



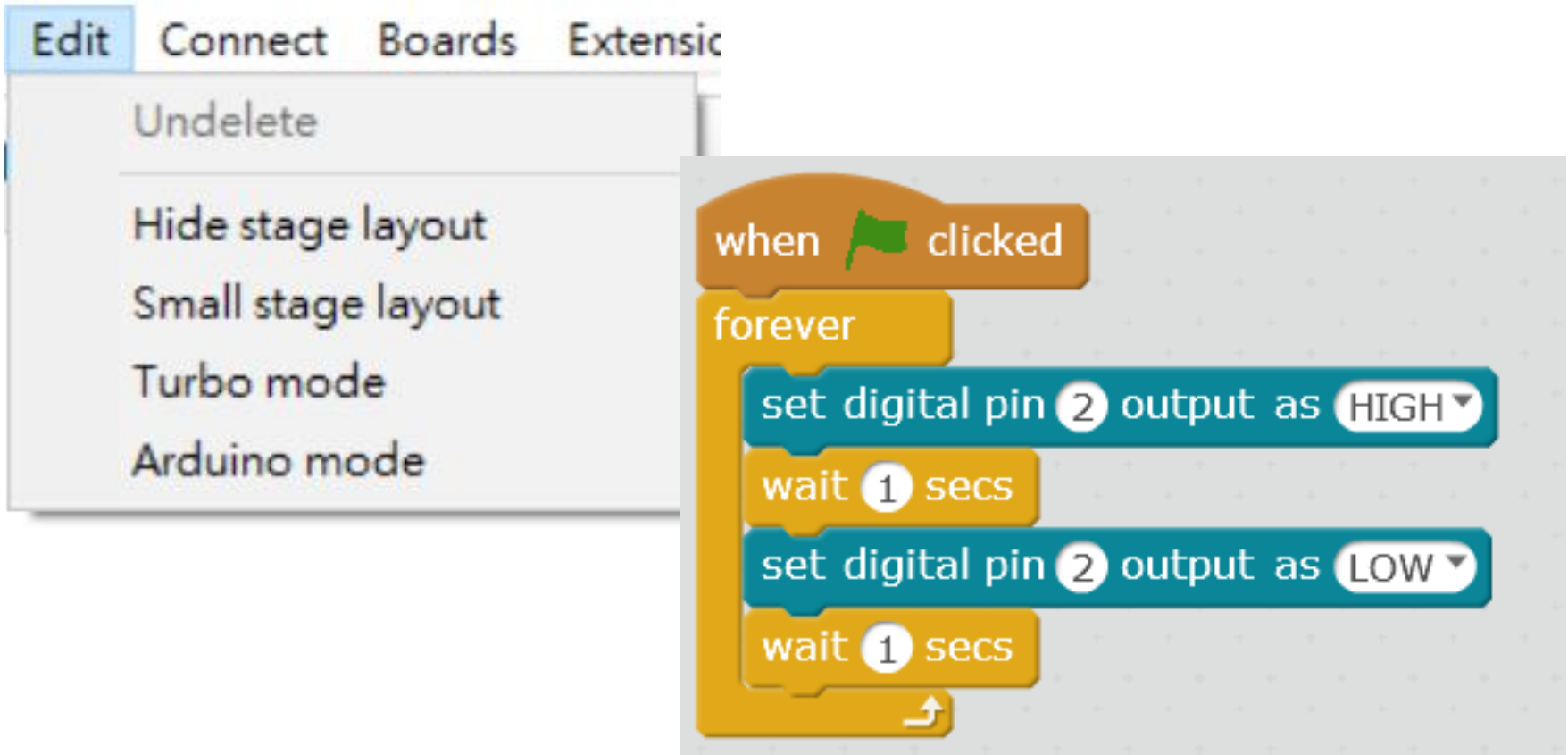
# Knowledge and Skills

- Using mBlock 3 (Basic Settings)



# Knowledge and Skills

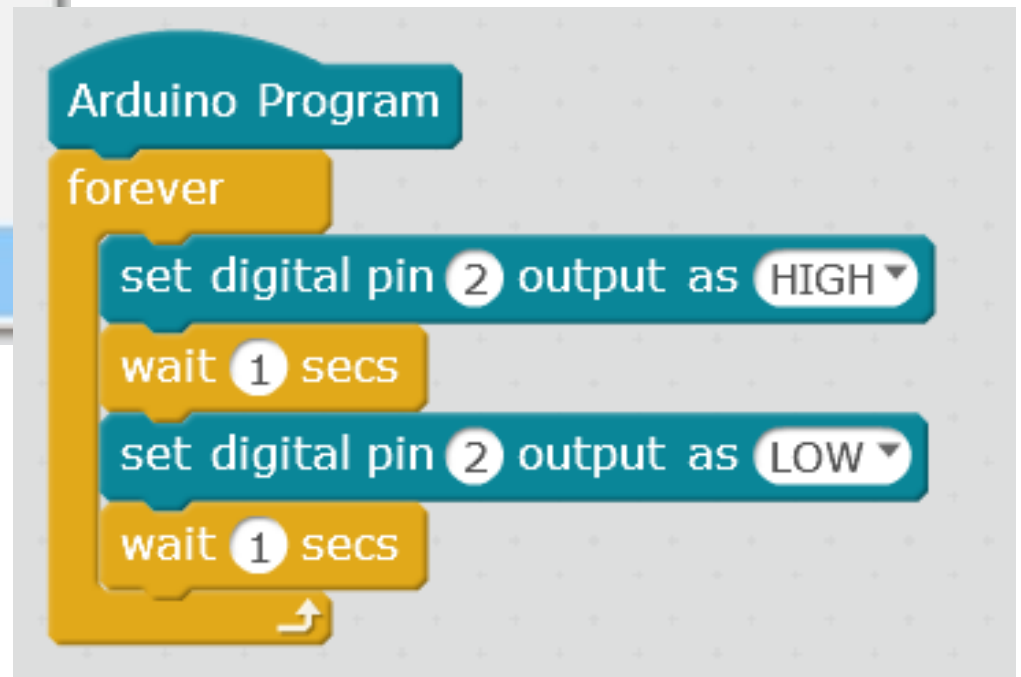
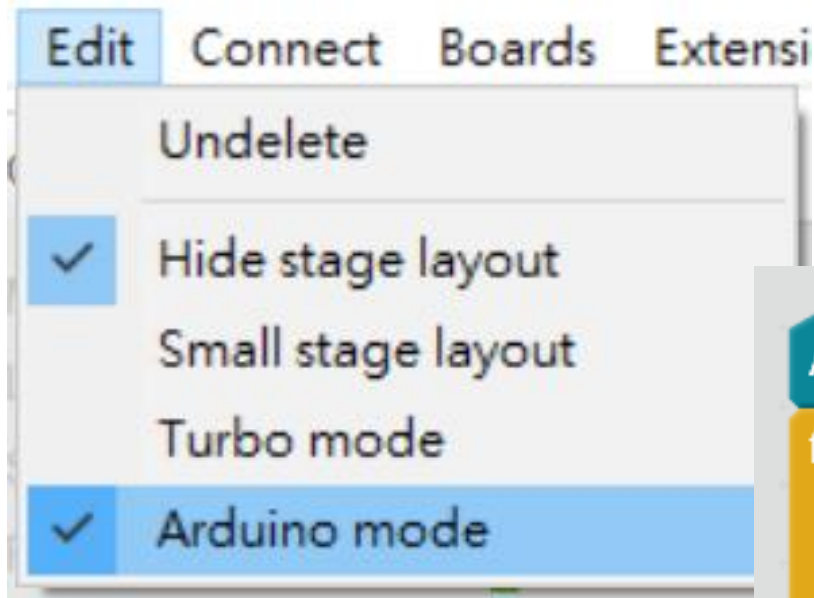
- Scratch mode





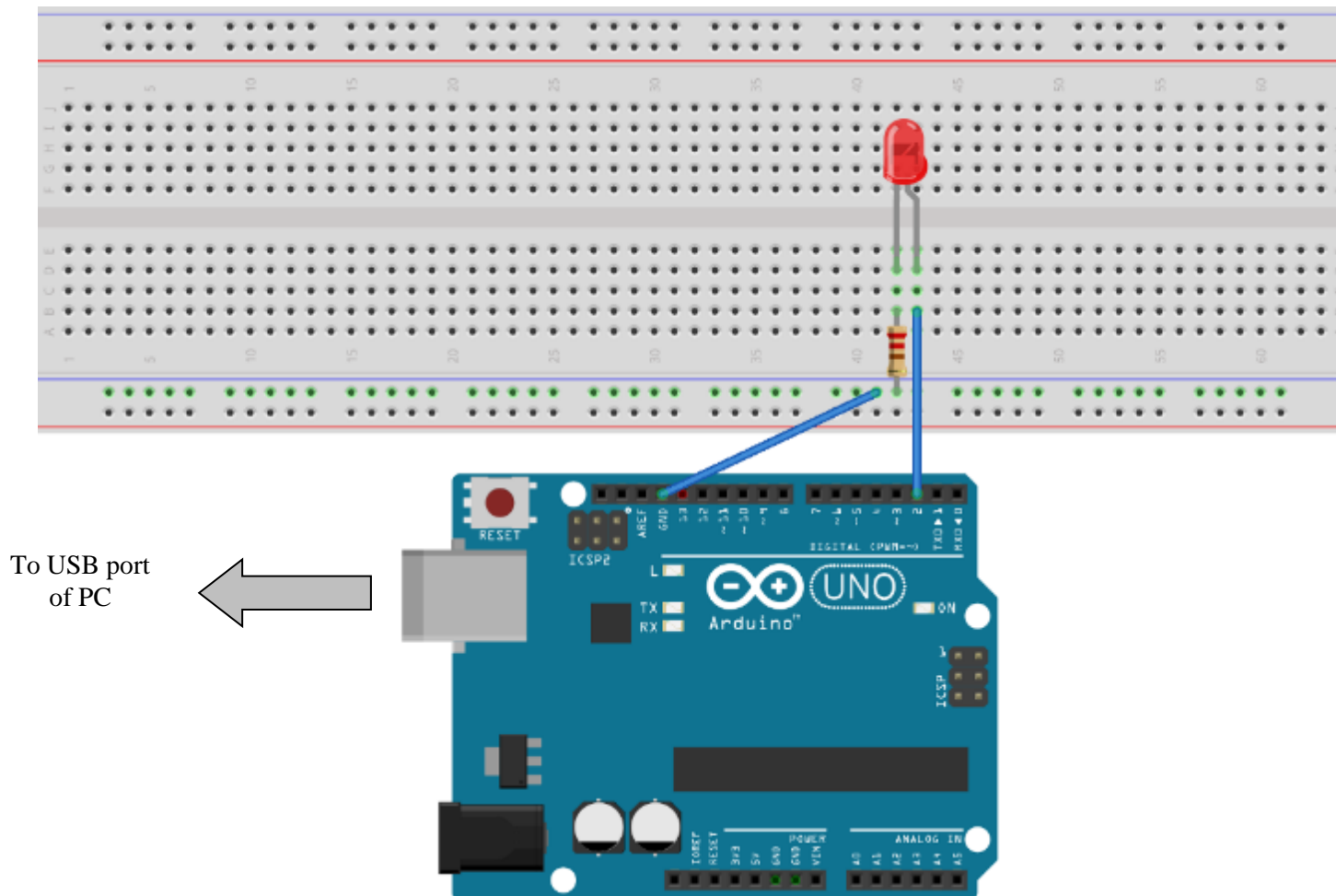
# Knowledge and Skills

- Arduino mode



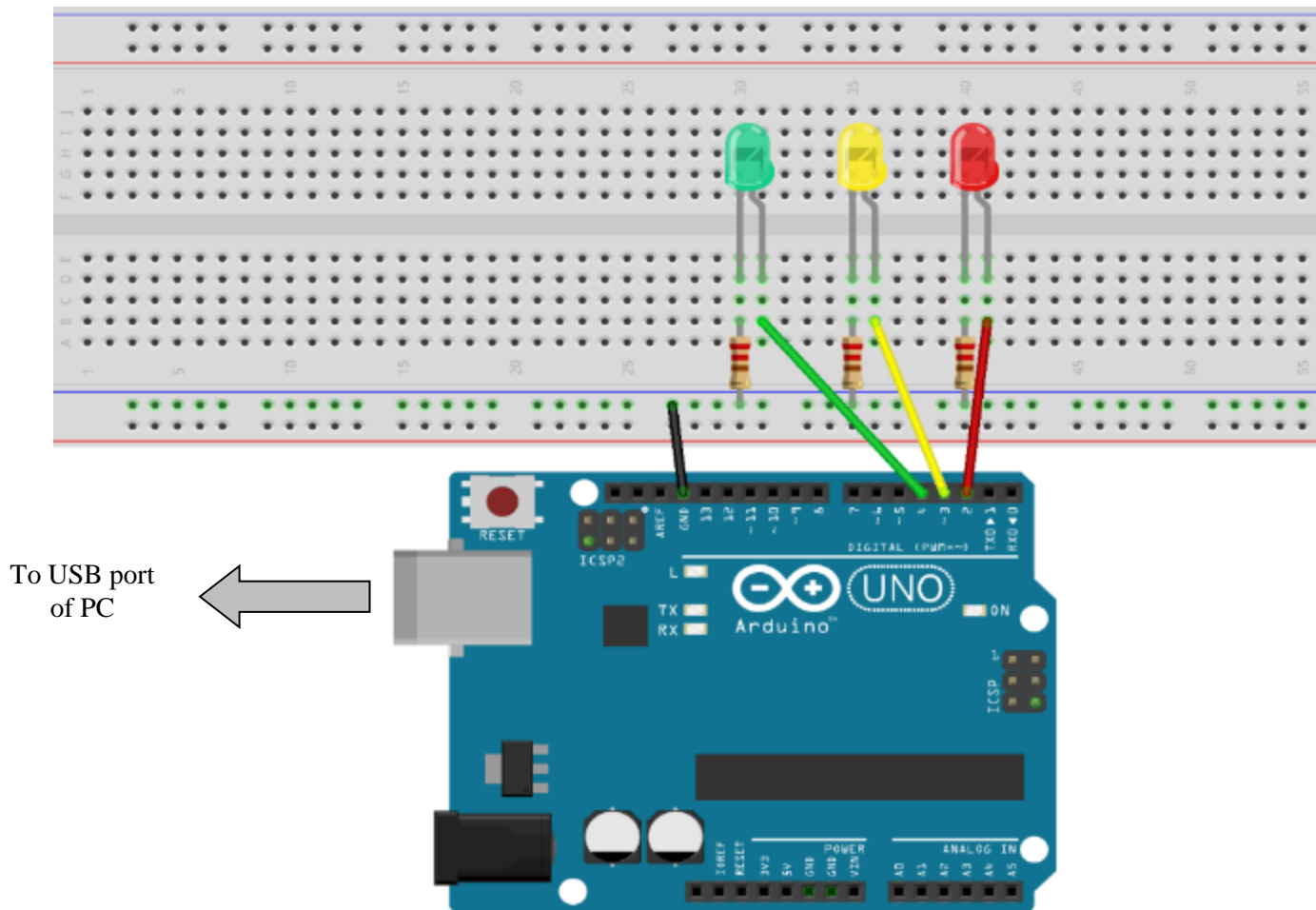
# Guided Tasks

- Blinking 1 LED



# Guided Tasks

- Controlling three LEDs



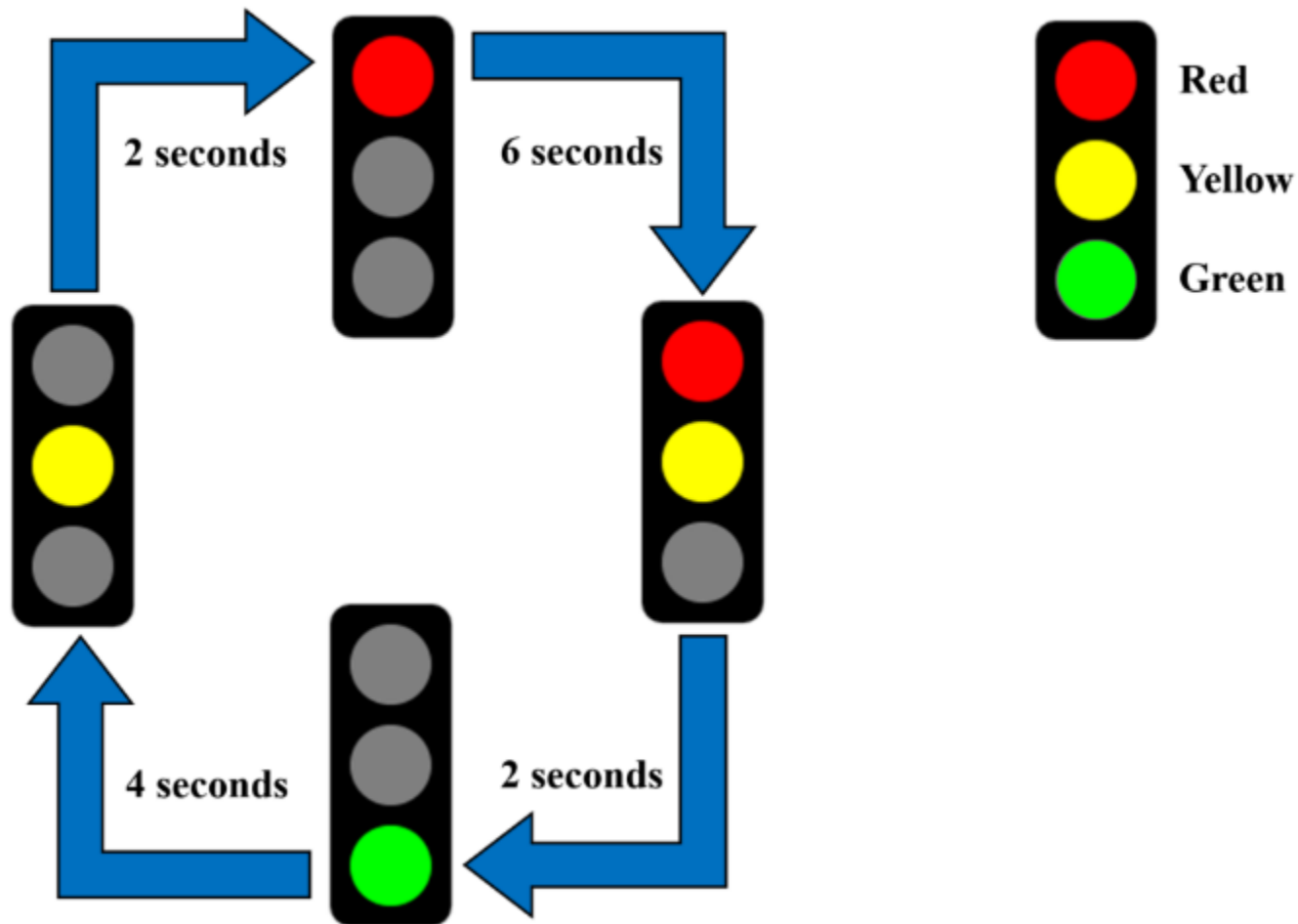
# Guided Tasks

- Controlling three LEDs
  - Other choices

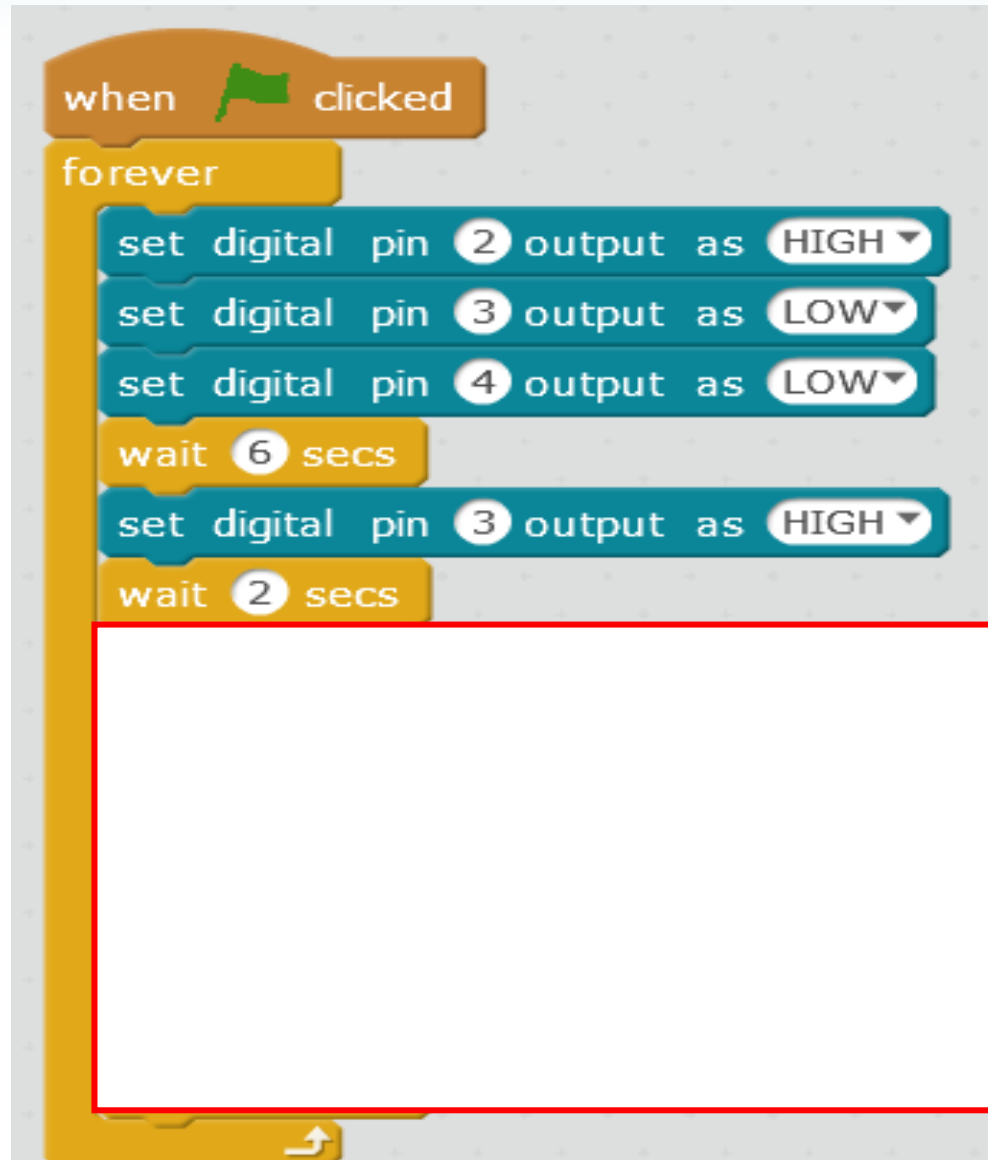


# Guided Tasks

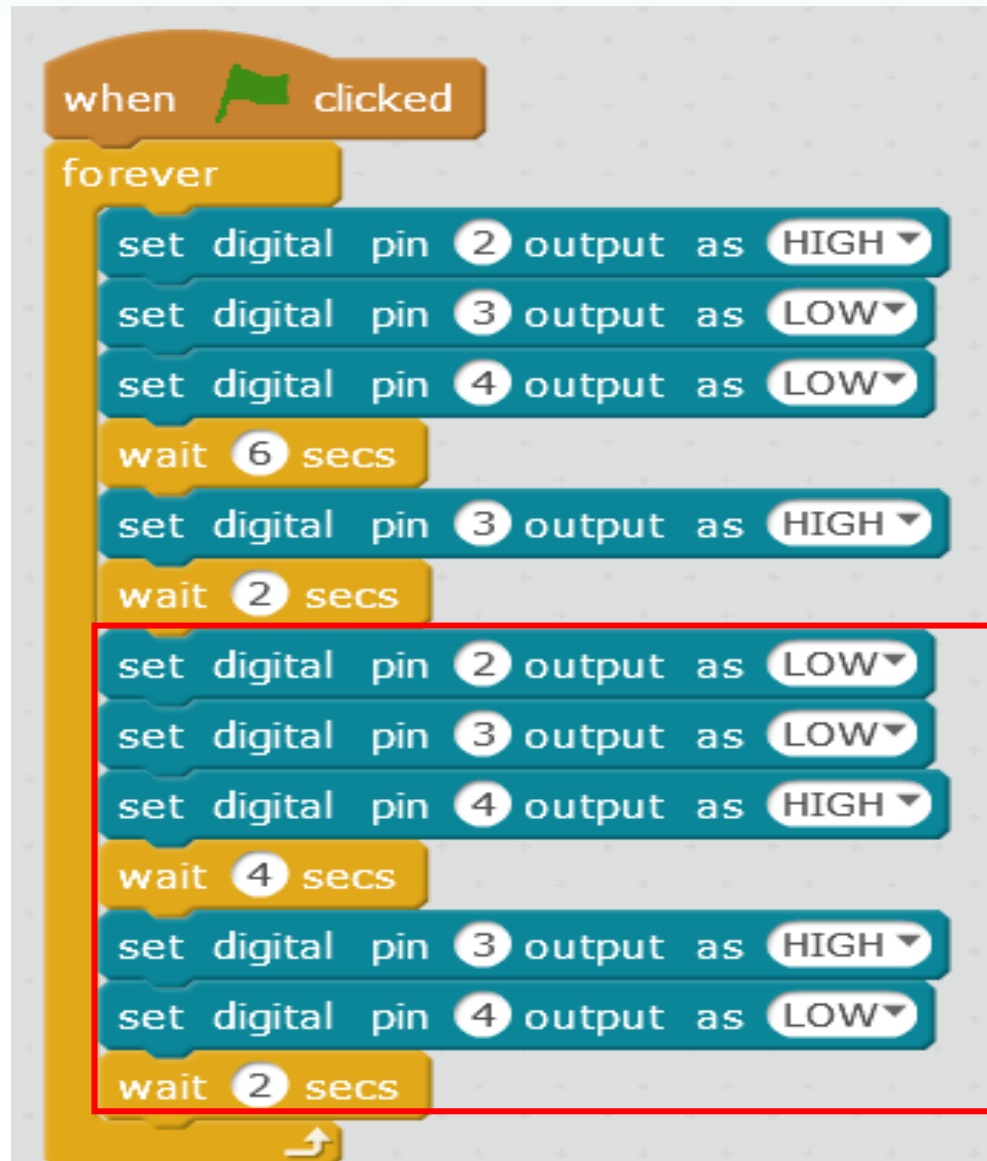
- Traffic Light Simulation



# Guided Tasks



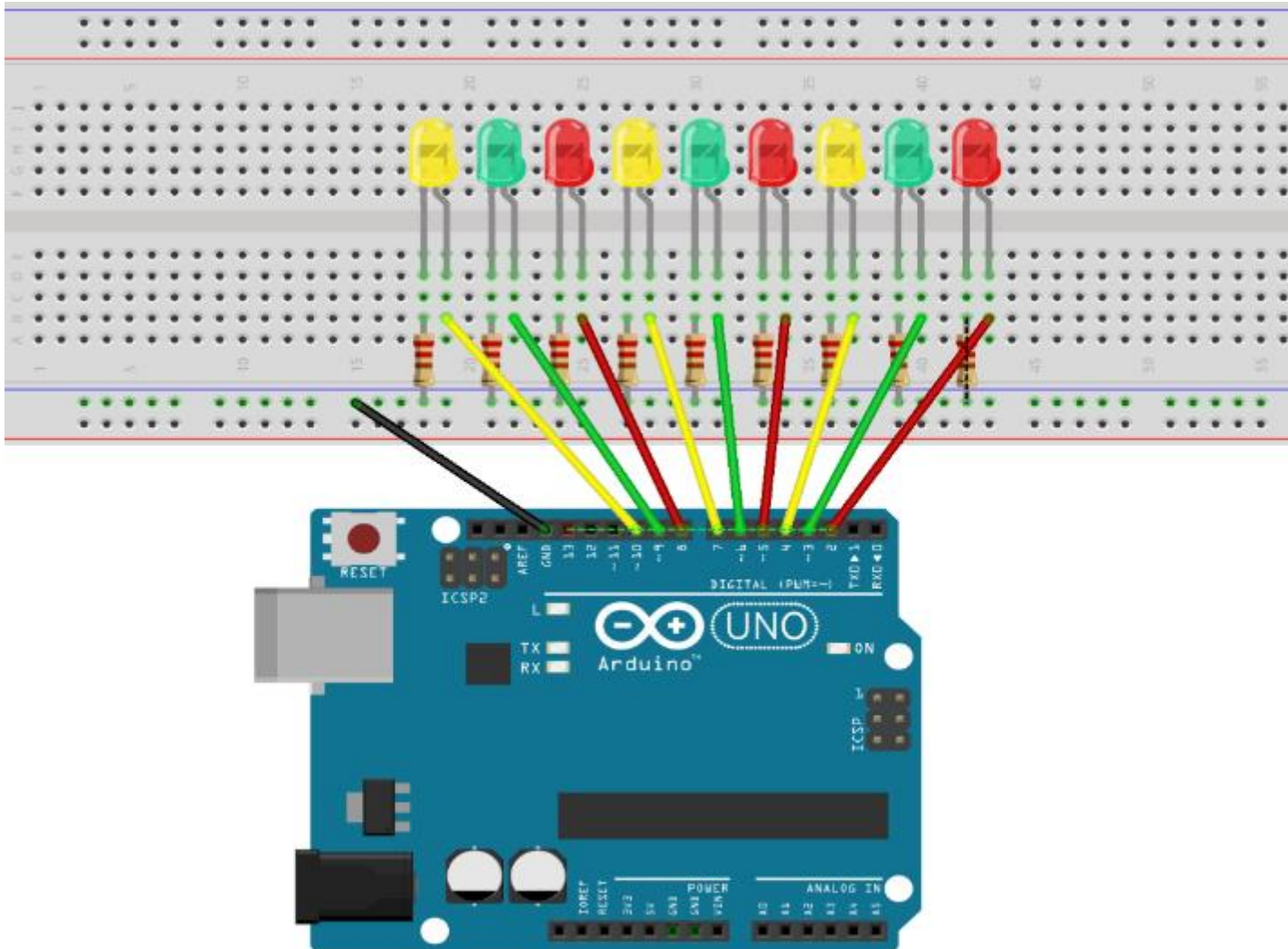
# Guided Tasks





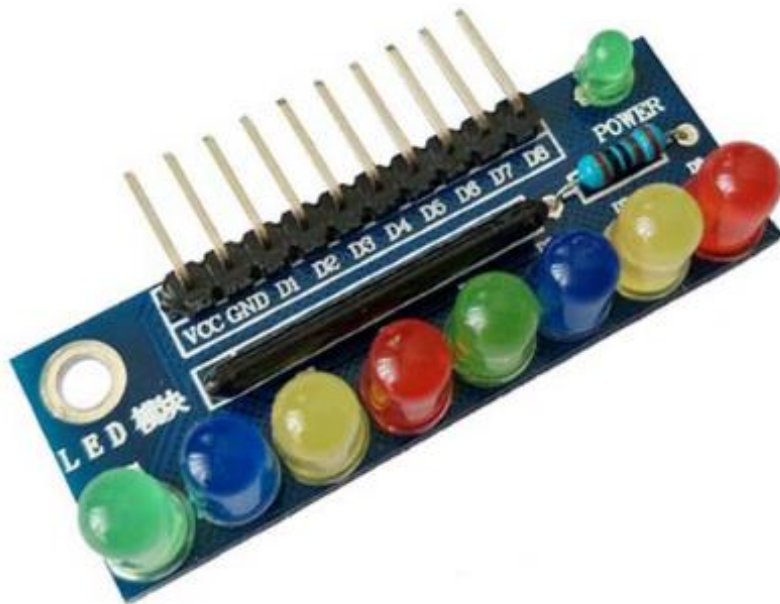
# Guided Tasks

- Controlling a series of LEDs




# Guided Tasks

- Controlling a series of LEDs
  - Other choices



# Guided Tasks

- Turning ON the LEDs One by One
- Programming Skills
  - Using **definite loops** to simplify some repeated patterns.
  - Using **user-defined blocks (subprograms)** to make the program more modular.
  - Using **variable** to adjust the waiting time.

when  clicked

set digital pin 2 output as LOW

set digital pin 3 output as LOW

set digital pin 4 output as LOW

...

set digital pin 9 output as LOW

set digital pin 10 output as LOW

set digital pin 2 output as HIGH

wait 0.2 secs

set digital pin 3 output as HIGH

wait 0.2 secs

set digital pin 4 output as HIGH

wait 0.2 secs

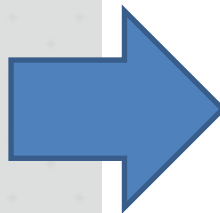
...


set digital pin 9 output as HIGH

wait 0.2 secs

set digital pin 10 output as HIGH

wait 0.2 secs



when  clicked

set waitTime to 0.2

turnOffLeds

turnOnLeds

define turnOffLeds

set pinNo to 1

repeat 9

change pinNo by 1

set digital pin pinNo output as LOW

define turnOnLeds

set pinNo to 1

repeat 9

change pinNo by 1

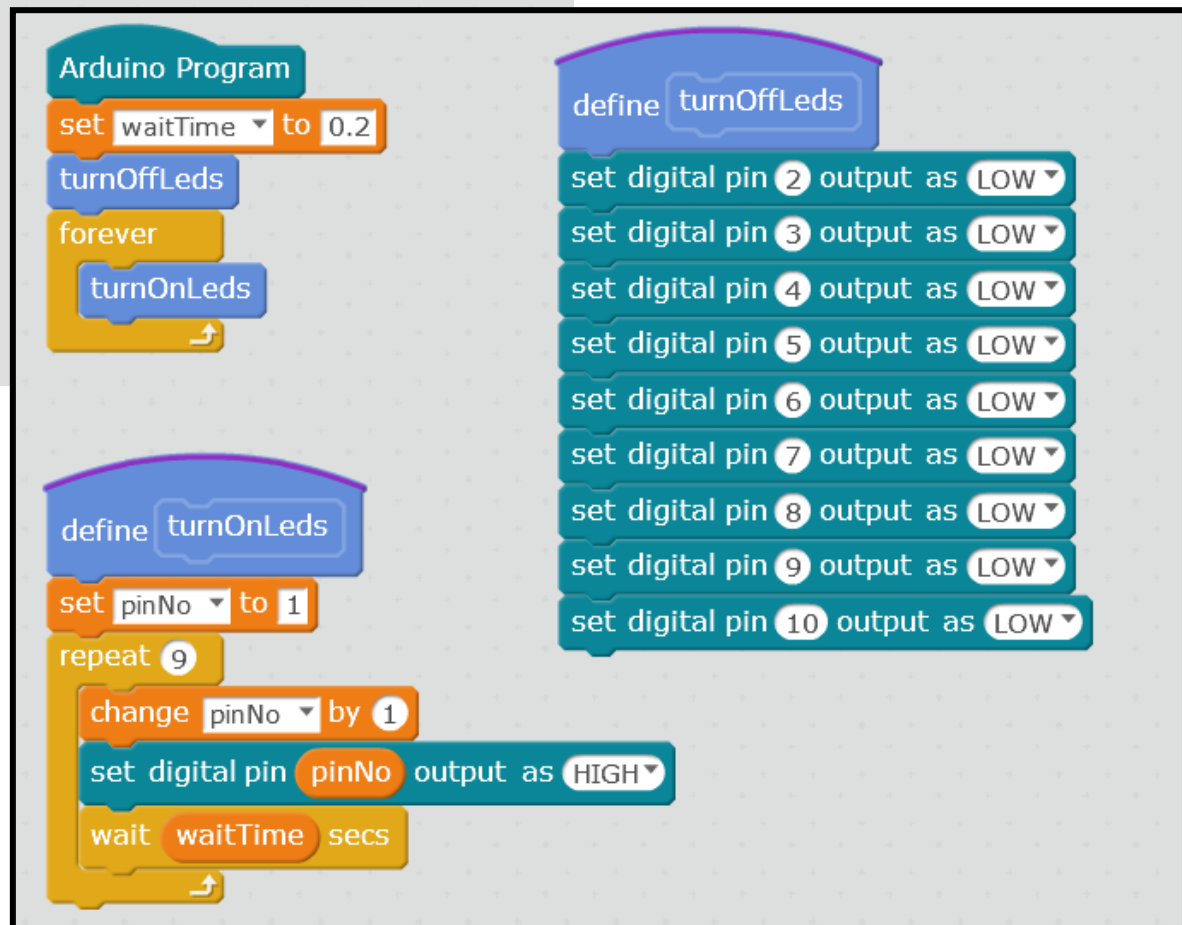
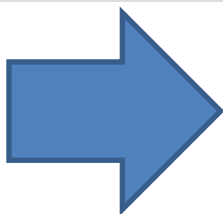
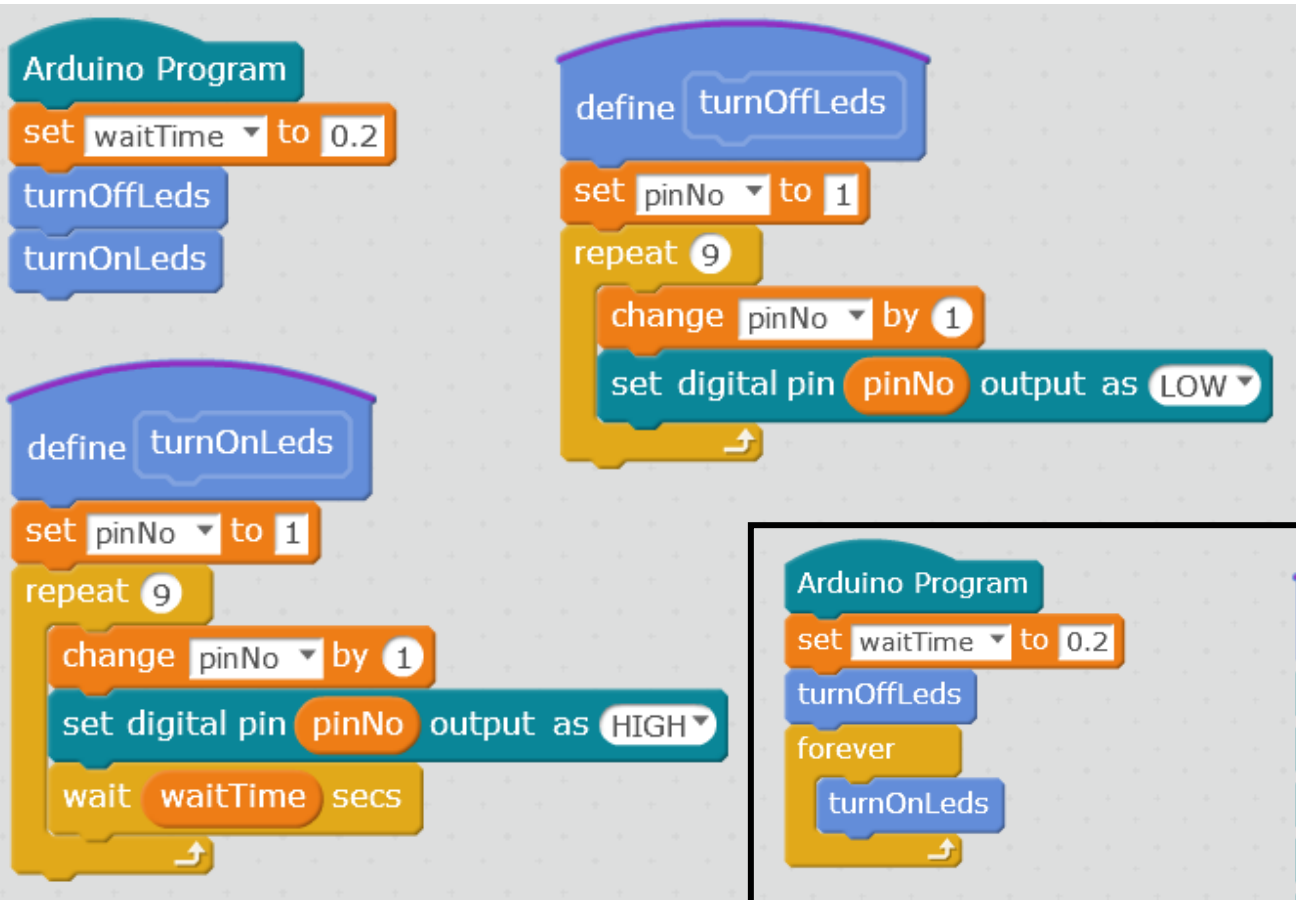
set digital pin pinNo output as HIGH

wait waitTime secs

# Guided Tasks

- Be careful when using **Arduino mode**
  - Inspect the program code if necessary
  - Are the output pins set properly at the start?

```
32
33 void setup() {
34     waitTime = 0.2;
35     turnOffLeds();
36     turnOnLeds();
37     pinMode(pinNo, OUTPUT);
38 }
39
```



# Guided Tasks

- Blinking the LEDs Continuously
- Blinking the LEDs Randomly
- Turning ON and OFF the LEDs in Different Directions
- Blinking the Red, Green and Yellow LEDs Randomly
- Slowing Down the Blinking Rate of LEDs



when  clicked

set waitTime ▾ to 0.2

turnOffLeds

forever

    blinkLedRandomly



define blinkLedRandomly

set pinNo ▾ to pick random 2 to 10

set digital pin pinNo output as HIGH ▾

wait waitTime secs

set digital pin pinNo output as LOW ▾

wait waitTime secs

define turnOffLeds

set digital pin 2 output as LOW ▾

set digital pin 3 output as LOW ▾

set digital pin 4 output as LOW ▾

set digital pin 5 output as LOW ▾

set digital pin 6 output as LOW ▾

set digital pin 7 output as LOW ▾

set digital pin 8 output as LOW ▾

set digital pin 9 output as LOW ▾

set digital pin 10 output as LOW ▾

```
when clicked
  set waitTime to 0.2
  turnOffLeds
  forever
    turnOnLedsRightToLeft
    turnOffLedsRightToLeft
    turnOnLedsLeftToRight
    turnOffLedsLeftToRight
```

```
define turnOnLedsRightToLeft
  set pinNo to 1
  repeat 9
    change pinNo by 1
    set digital pin pinNo output as HIGH
    wait waitTime secs
```

```
define turnOnLedsLeftToRight
```

```
define turnOffLeds
  set digital pin 2 output as LOW
  set digital pin 3 output as LOW
  set digital pin 4 output as LOW
  set digital pin 5 output as LOW
  set digital pin 6 output as LOW
  set digital pin 7 output as LOW
  set digital pin 8 output as LOW
  set digital pin 9 output as LOW
  set digital pin 10 output as LOW
```

```
define turnOffLedsRightToLeft
```

```
define turnOffLedsLeftToRight
  set pinNo to 11
  repeat 9
    change pinNo by -1
    set digital pin pinNo output as LOW
    wait waitTime secs
```

```
when clicked
  set waitTime to 0.2
  turnOffLeds
  forever
    turnOnLedsRightToLeft
    turnOffLedsRightToLeft
    turnOnLedsLeftToRight
    turnOffLedsLeftToRight
```

```
define turnOnLedsRightToLeft
  set pinNo to 1
  repeat 9
    change pinNo by 1
    set digital pin pinNo output as HIGH
    wait waitTime secs
```

```
define turnOnLedsLeftToRight
  set pinNo to 11
  repeat 9
    change pinNo by -1
    set digital pin pinNo output as HIGH
    wait waitTime secs
```

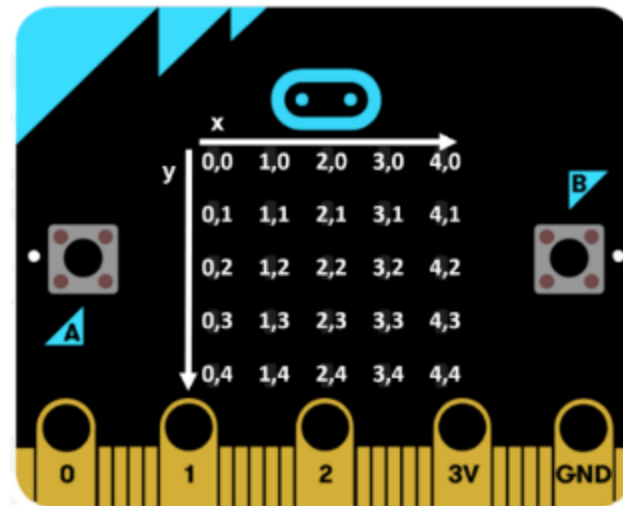
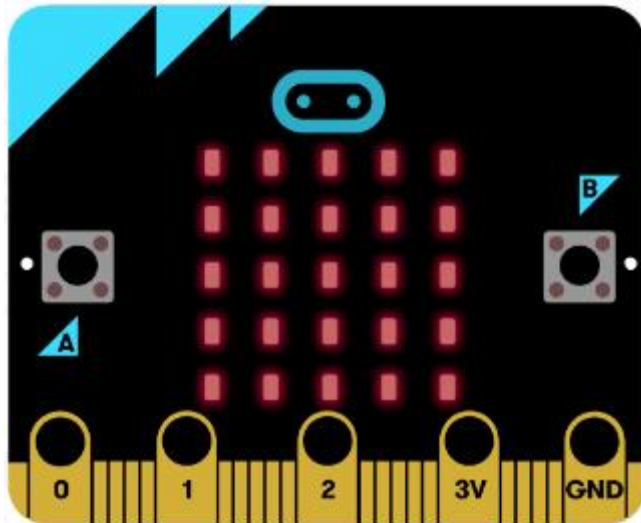
```
define turnOffLeds
  set digital pin 2 output as LOW
  set digital pin 3 output as LOW
  set digital pin 4 output as LOW
  set digital pin 5 output as LOW
  set digital pin 6 output as LOW
  set digital pin 7 output as LOW
  set digital pin 8 output as LOW
  set digital pin 9 output as LOW
  set digital pin 10 output as LOW
```

```
define turnOffLedsRightToLeft
  set pinNo to 1
  repeat 9
    change pinNo by 1
    set digital pin pinNo output as LOW
    wait waitTime secs
```

```
define turnOffLedsLeftToRight
  set pinNo to 11
  repeat 9
    change pinNo by -1
    set digital pin pinNo output as LOW
    wait waitTime secs
```

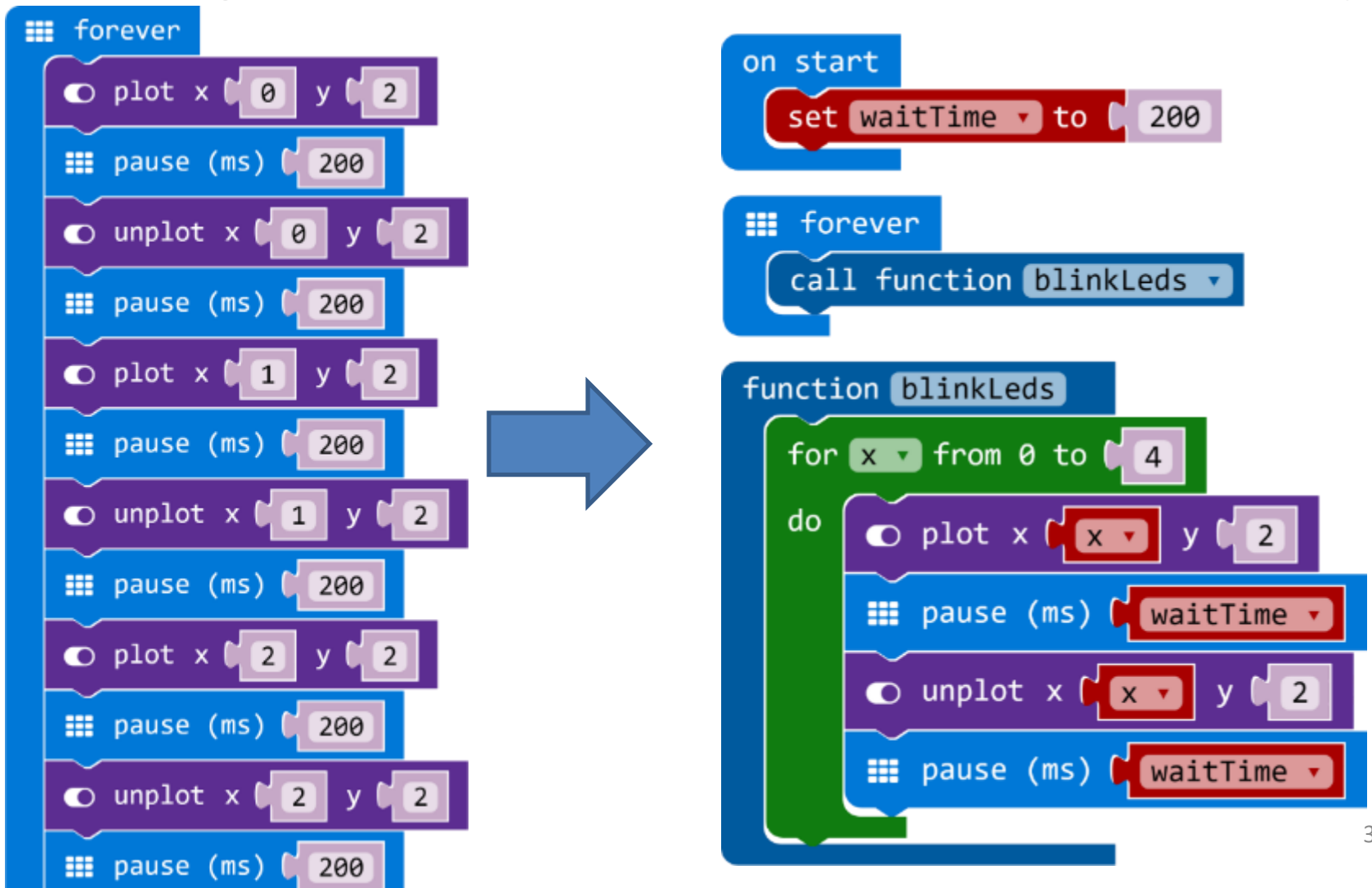
# Guided Tasks

- For micro:bit, using on-board LED Screen



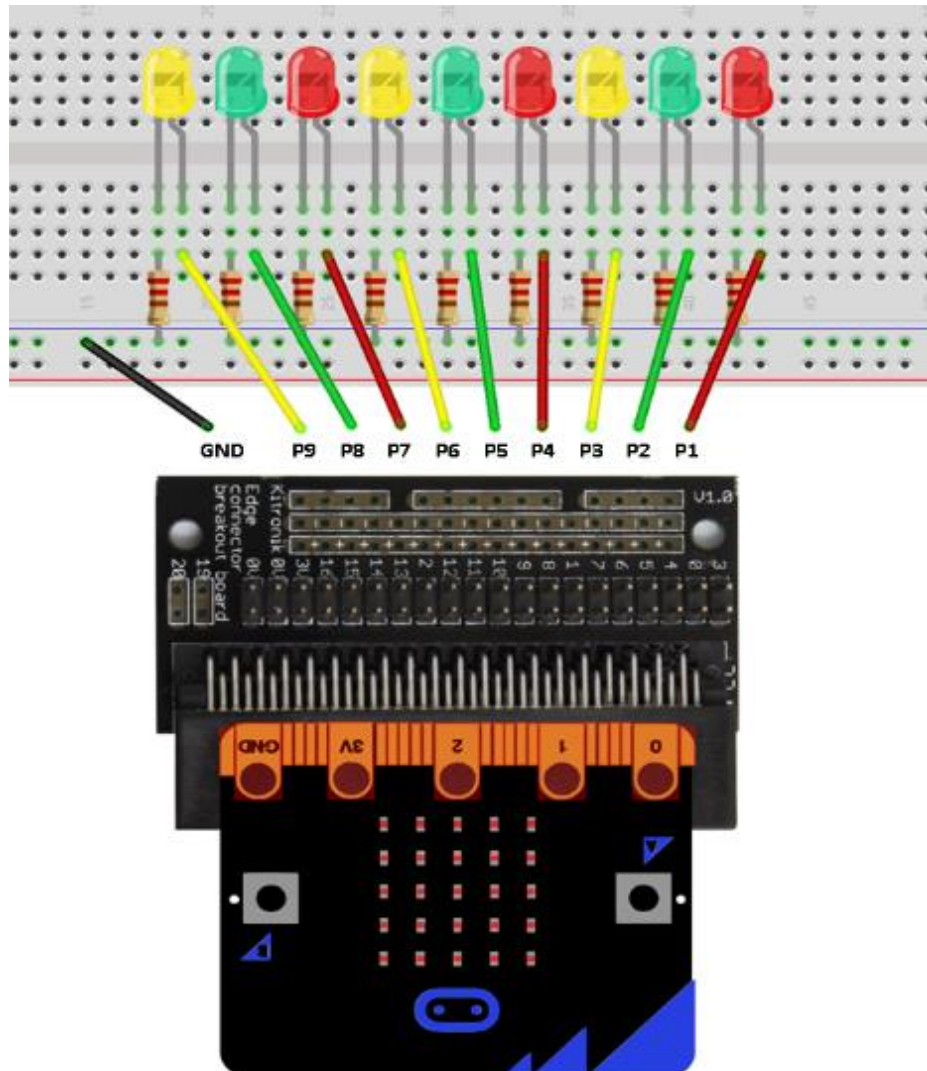
# Guided Tasks

- Blinking the LEDs on the Middle Row Continuously



# Guided Tasks

- micro:bit - Controlling a Series of External LEDs



```

on start
  set waitTime to 300
  led enable false
  call function turnOffAllLeds

forever
  call function blinkRedLeds
  call function blinkGreenLeds
  call function blinkYellowLeds

function turnOffAllLeds
  digital write pin P1 to 0
  digital write pin P2 to 0
  digital write pin P3 to 0
  digital write pin P4 to 0
  digital write pin P5 to 0
  digital write pin P6 to 0
  digital write pin P7 to 0
  digital write pin P8 to 0
  digital write pin P9 to 0

function blinkRedLeds
  digital write pin P1 to 1

```

```

function blinkGreenLeds
  digital write pin P2 to 1
  digital write pin P5 to 1
  digital write pin P8 to 1
  pause (ms) waitTime
  digital write pin P2 to 0
  digital write pin P5 to 0
  digital write pin P8 to 0
  pause (ms) waitTime

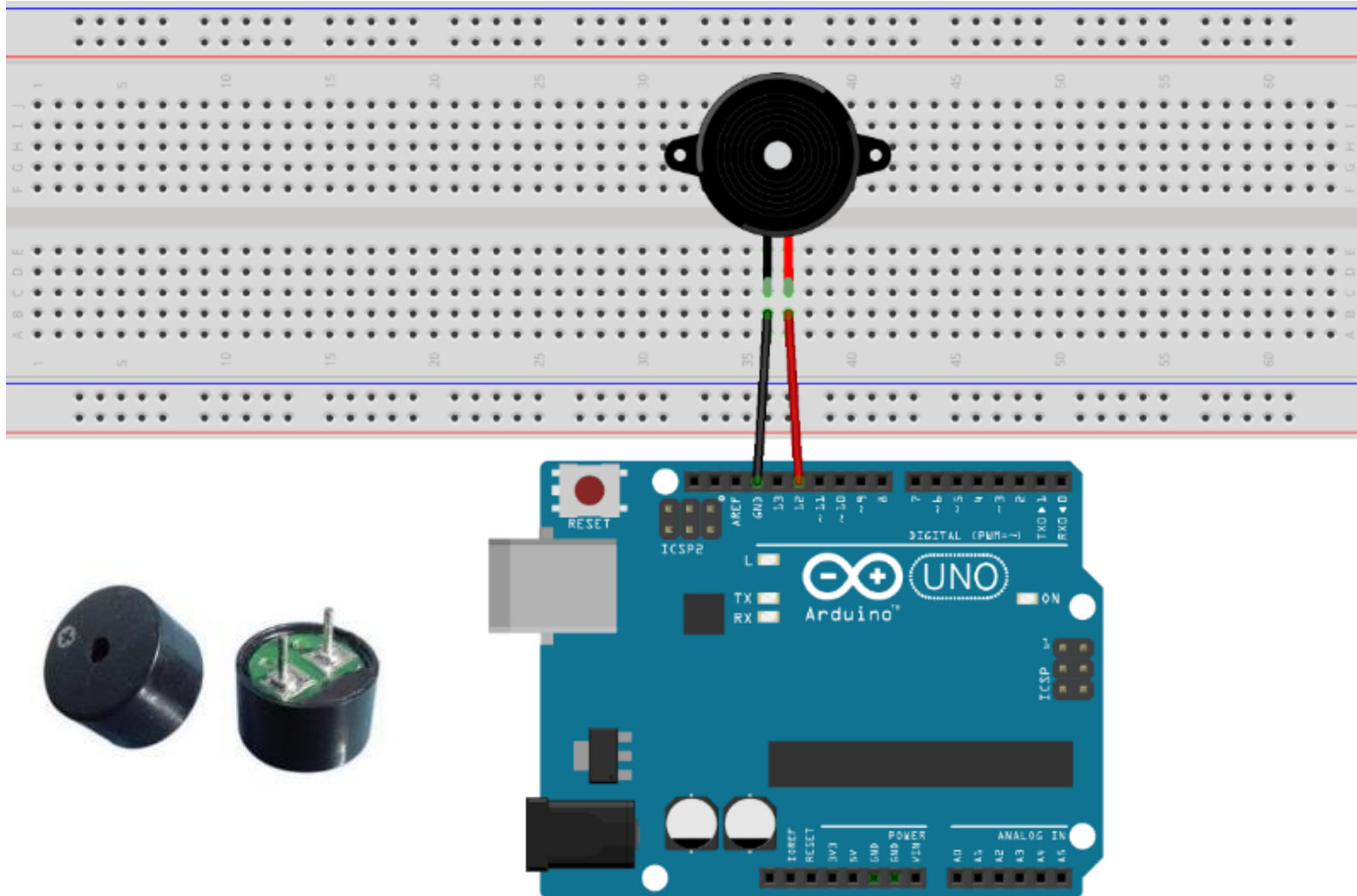
function blinkYellowLeds
  digital write pin P3 to 1
  digital write pin P6 to 1
  digital write pin P9 to 1
  pause (ms) waitTime
  digital write pin P3 to 0
  digital write pin P6 to 0
  digital write pin P9 to 0
  pause (ms) waitTime

```



# Guided Tasks

- Composing and Playing Music





when  clicked

forever

playMusic1

define playMusic1

play tone pin 12 on note C4 beat Quarter

play tone pin 12 on note C4 beat Quarter

play tone pin 12 on note G4 beat Quarter

play tone pin 12 on note G4 beat Quarter

play tone pin 12 on note A4 beat Quarter

play tone pin 12 on note A4 beat Quarter

play tone pin 12 on note G4 beat Quarter

wait 0.2 secs

play tone pin 12 on note F4 beat Quarter

play tone pin 12 on note F4 beat Quarter

play tone pin 12 on note E4 beat Quarter

play tone pin 12 on note E4 beat Quarter

play tone pin 12 on note D4 beat Quarter

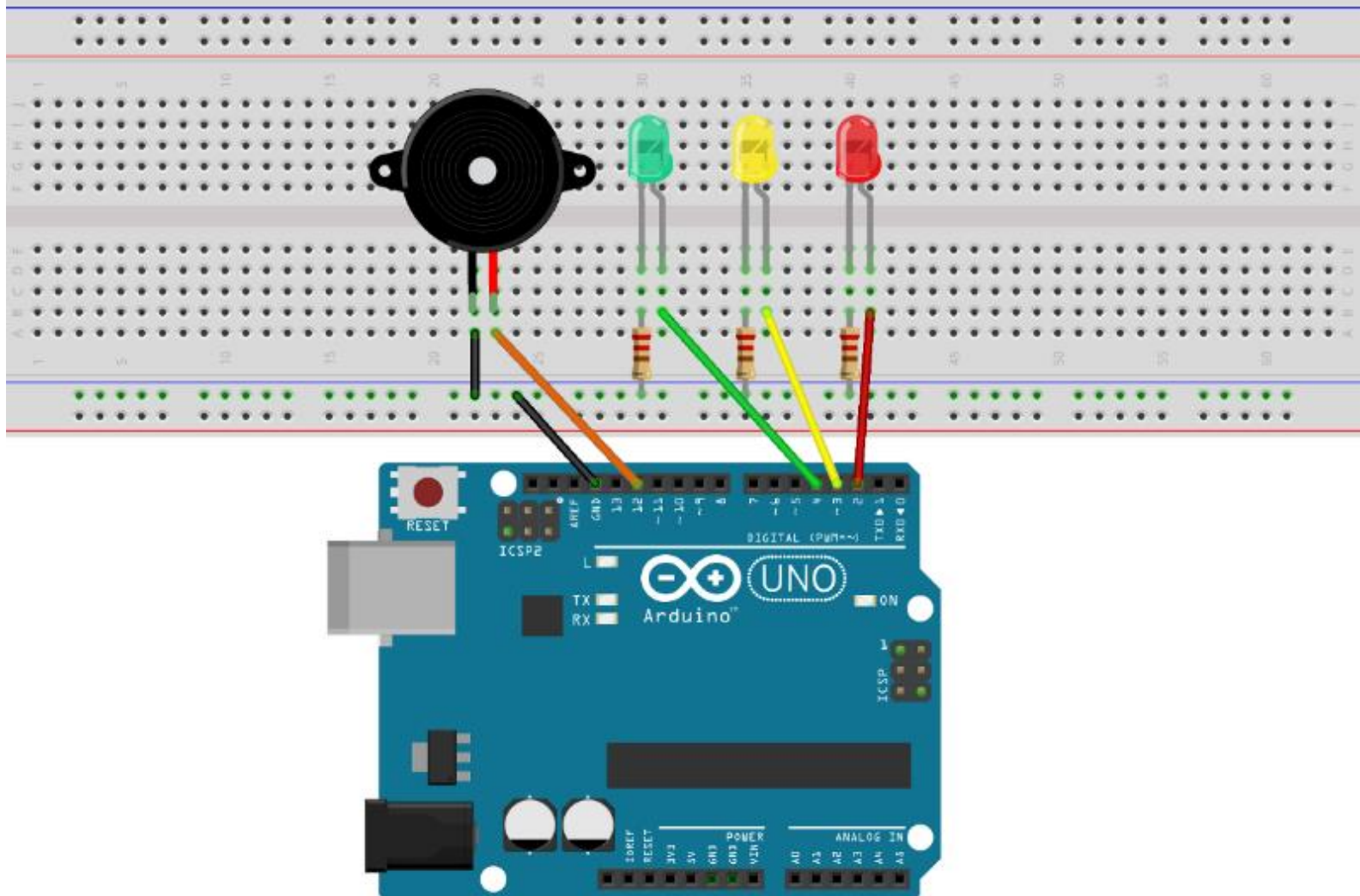
play tone pin 12 on note D4 beat Quarter

play tone pin 12 on note C4 beat Quarter

wait 0.2 secs

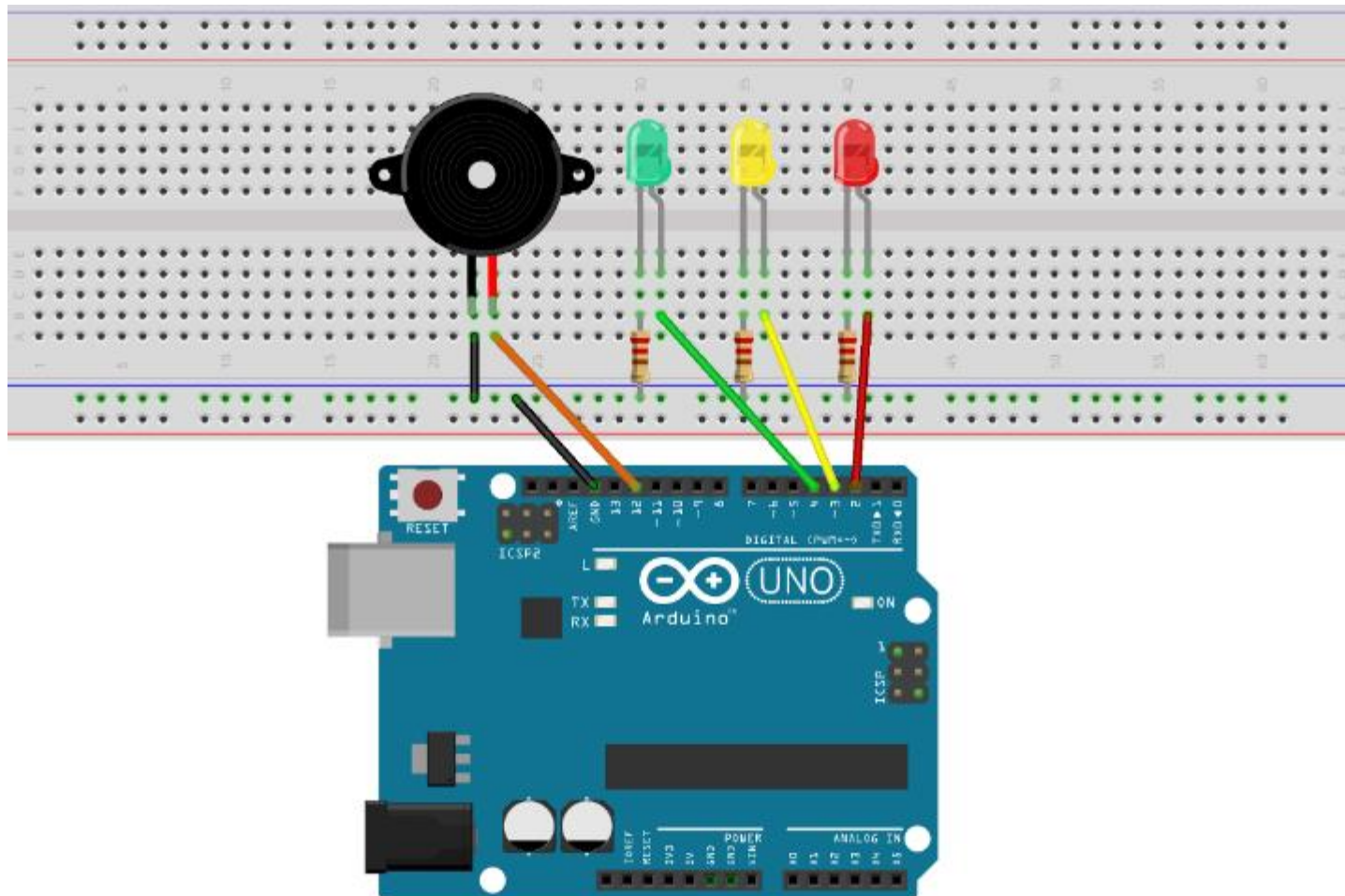
# Guided Tasks

- Blinking LEDs and Playing Music **Concurrently**



# Guided Tasks

- Blinking LEDs and Playing Music **Concurrently**
  - Scratch mode vs Arduino mode



when  clicked

set digital pin 2 output as LOW

set digital pin 3 output as LOW

set digital pin 4 output as LOW

forever

set digital pin 2 output as HIGH

wait 0.2 secs

set digital pin 2 output as LOW

wait 0.2 secs

set digital pin 3 output as HIGH

wait 0.2 secs

set digital pin 3 output as LOW

wait 0.2 secs

set digital pin 4 output as HIGH

wait 0.2 secs

set digital pin 4 output as LOW

wait 0.2 secs



when  clicked

forever

play tone pin 12 on note C4 beat Quarter

play tone pin 12 on note C4 beat Quarter

play tone pin 12 on note G4 beat Quarter

play tone pin 12 on note G4 beat Quarter

play tone pin 12 on note A4 beat Quarter

play tone pin 12 on note A4 beat Quarter

play tone pin 12 on note G4 beat Quarter

wait 0.2 secs

play tone pin 12 on note F4 beat Quarter

play tone pin 12 on note F4 beat Quarter

play tone pin 12 on note E4 beat Quarter

play tone pin 12 on note E4 beat Quarter

play tone pin 12 on note D4 beat Quarter

play tone pin 12 on note D4 beat Quarter

play tone pin 12 on note C4 beat Quarter

wait 0.2 secs





## Arduino Program

set digital pin 2 output as LOW

set digital pin 3 output as LOW

set digital pin 4 output as LOW

forever

set digital pin 2 output as HIGH

wait 0.2 secs

set digital pin 2 output as LOW

wait 0.2 secs

set digital pin 3 output as HIGH

wait 0.2 secs

set digital pin 3 output as LOW

wait 0.2 secs

set digital pin 4 output as HIGH

wait 0.2 secs

set digital pin 4 output as LOW

wait 0.2 secs



## Arduino Program

forever

play tone pin 12 on note C4 beat Quarter

play tone pin 12 on note C4 beat Quarter

play tone pin 12 on note G4 beat Quarter

play tone pin 12 on note G4 beat Quarter

play tone pin 12 on note A4 beat Quarter

play tone pin 12 on note A4 beat Quarter

play tone pin 12 on note G4 beat Quarter

wait 0.2 secs

play tone pin 12 on note F4 beat Quarter

play tone pin 12 on note F4 beat Quarter

play tone pin 12 on note E4 beat Quarter

play tone pin 12 on note E4 beat Quarter

play tone pin 12 on note D4 beat Quarter

play tone pin 12 on note D4 beat Quarter

play tone pin 12 on note C4 beat Quarter

wait 0.2 secs

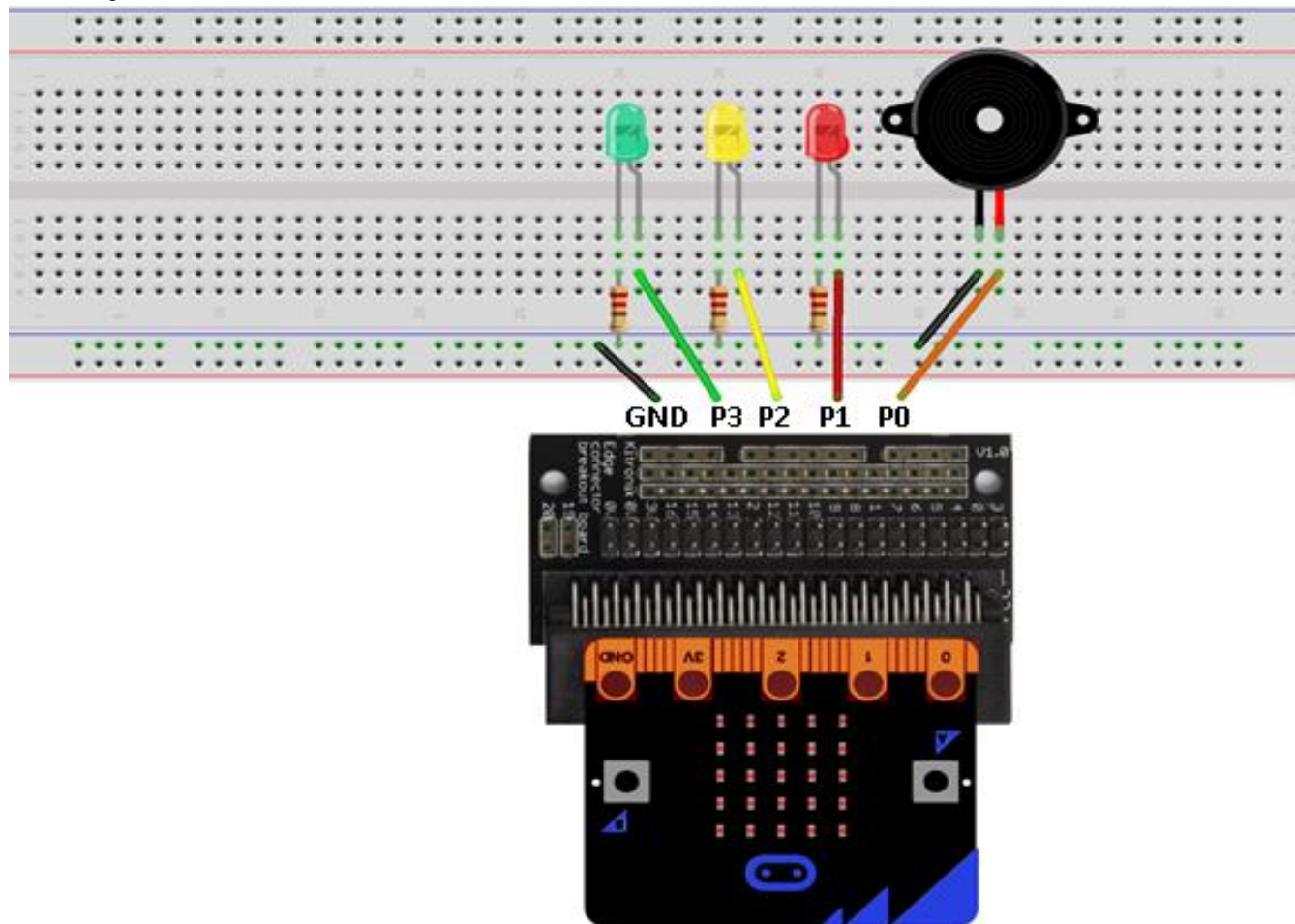


# Guided Tasks

- Blinking LEDs and Playing Music **Concurrently**
- Reference:
  - <https://learn.adafruit.com/multi-tasking-the-arduino-part-1/overview>

# Guided Tasks

- Blinking LEDs and Playing Music **Concurrently**
  - No problem with micro:bit



on start

led enable false

forever

digital write pin P1 to 1

pause (ms) 200

digital write pin P1 to 0

pause (ms) 200

digital write pin P2 to 1

pause (ms) 200

digital write pin P2 to 0

pause (ms) 200

digital write pin P3 to 1

pause (ms) 200

digital write pin P3 to 0

pause (ms) 200



forever

play tone Middle C for 1/2 beat

play tone Middle C for 1/2 beat

play tone Middle G for 1/2 beat

play tone Middle G for 1/2 beat

play tone Middle A for 1/2 beat

play tone Middle A for 1/2 beat

play tone Middle G for 1/2 beat

pause (ms) 200

play tone Middle F for 1/2 beat

play tone Middle F for 1/2 beat

play tone Middle E for 1/2 beat

play tone Middle E for 1/2 beat

play tone Middle D for 1/2 beat

play tone Middle D for 1/2 beat

play tone Middle C for 1/2 beat

pause (ms) 200



# Self-Exploration Project

- Students are required to develop a Light Show by using Arduino/micro:bit and LEDs.
  - with no more than 10 LEDs
  - design a program that display at least four lighting patterns
  - If possible, add music by using a buzzer

# Self-Exploration Project

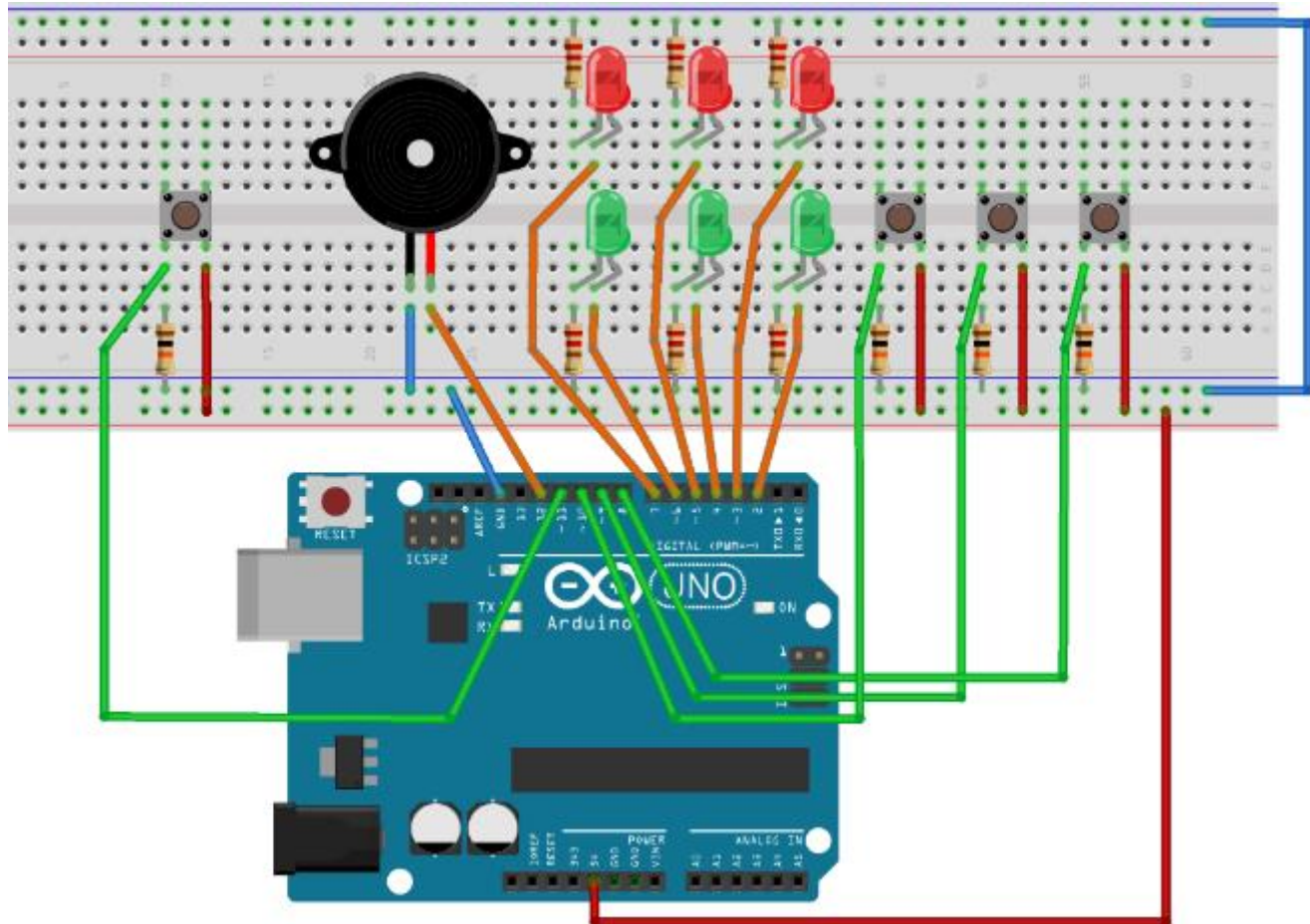
- System Development Life Cycle
  1. Problem definition
  2. Problem analysis
  3. Algorithm design and Program coding
  4. Program debugging/testing
  5. Program documentation

# Project 2

A Simple Interactive Game

# Project Requirements

- Designing a simple interactive game using push-buttons, LEDs and buzzer.

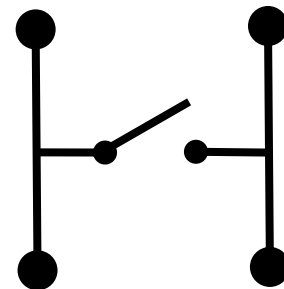
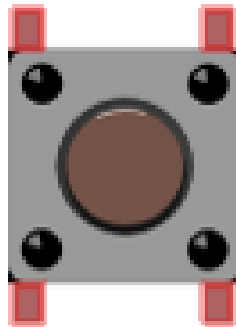


# Knowledge and Skills

- Construct simple circuits by using single-board microcontroller, breadboard, push-buttons, LEDs, buzzers, resistors, etc.
- Create simple programs by using visual programming IDE
  - definite loop
  - indefinite loop
  - infinite loop
  - selection statements
  - etc.

# Basic Knowledge

- Momentary Switch
- Maintained Switch (Toggle Switch)
- Push-button



# Basic Knowledge

- Push-button module
- No need to connect the 10 k $\Omega$  pull-down resistor



# Basic Knowledge

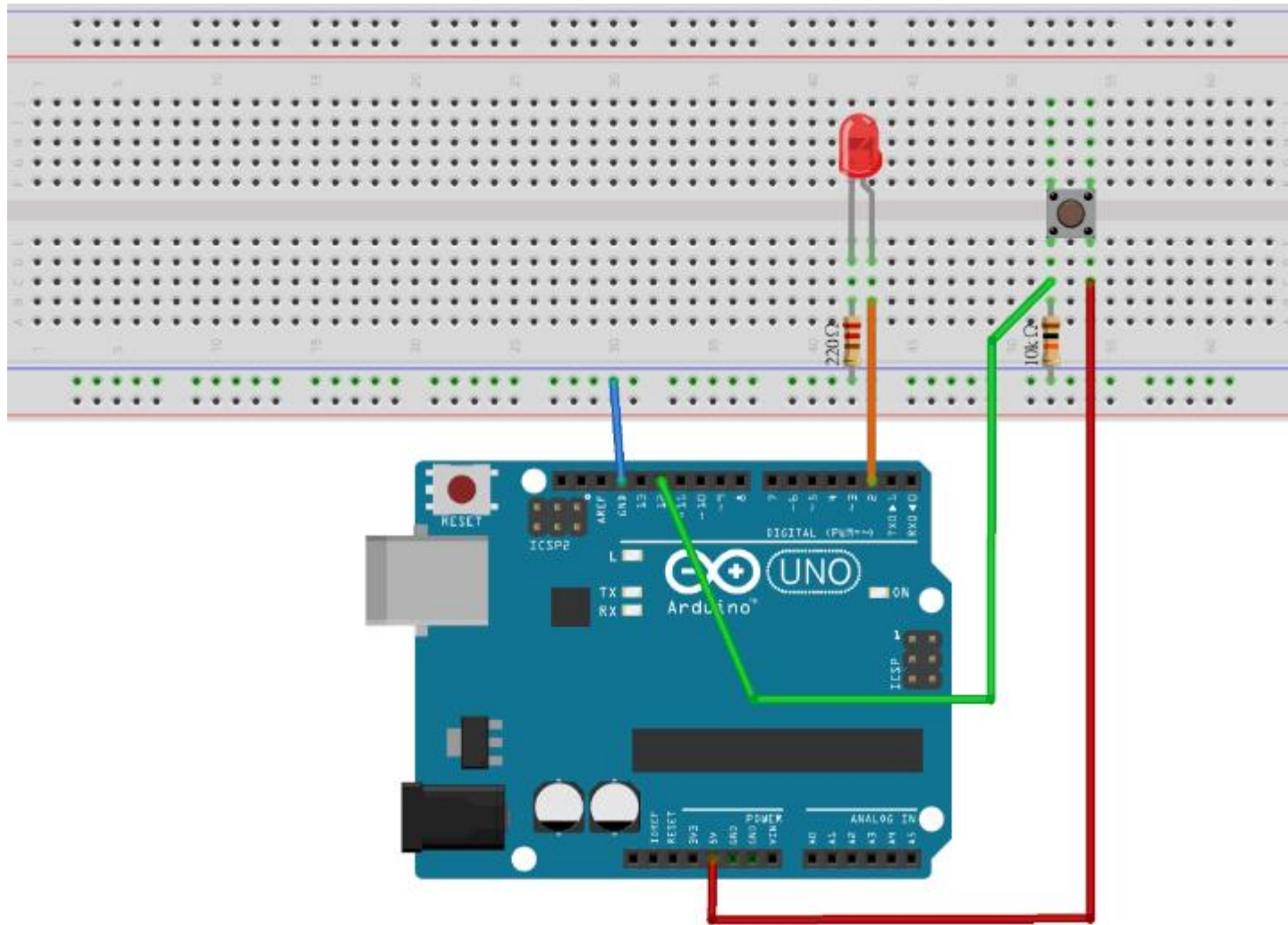
- Other choice
  - Touch Sensor



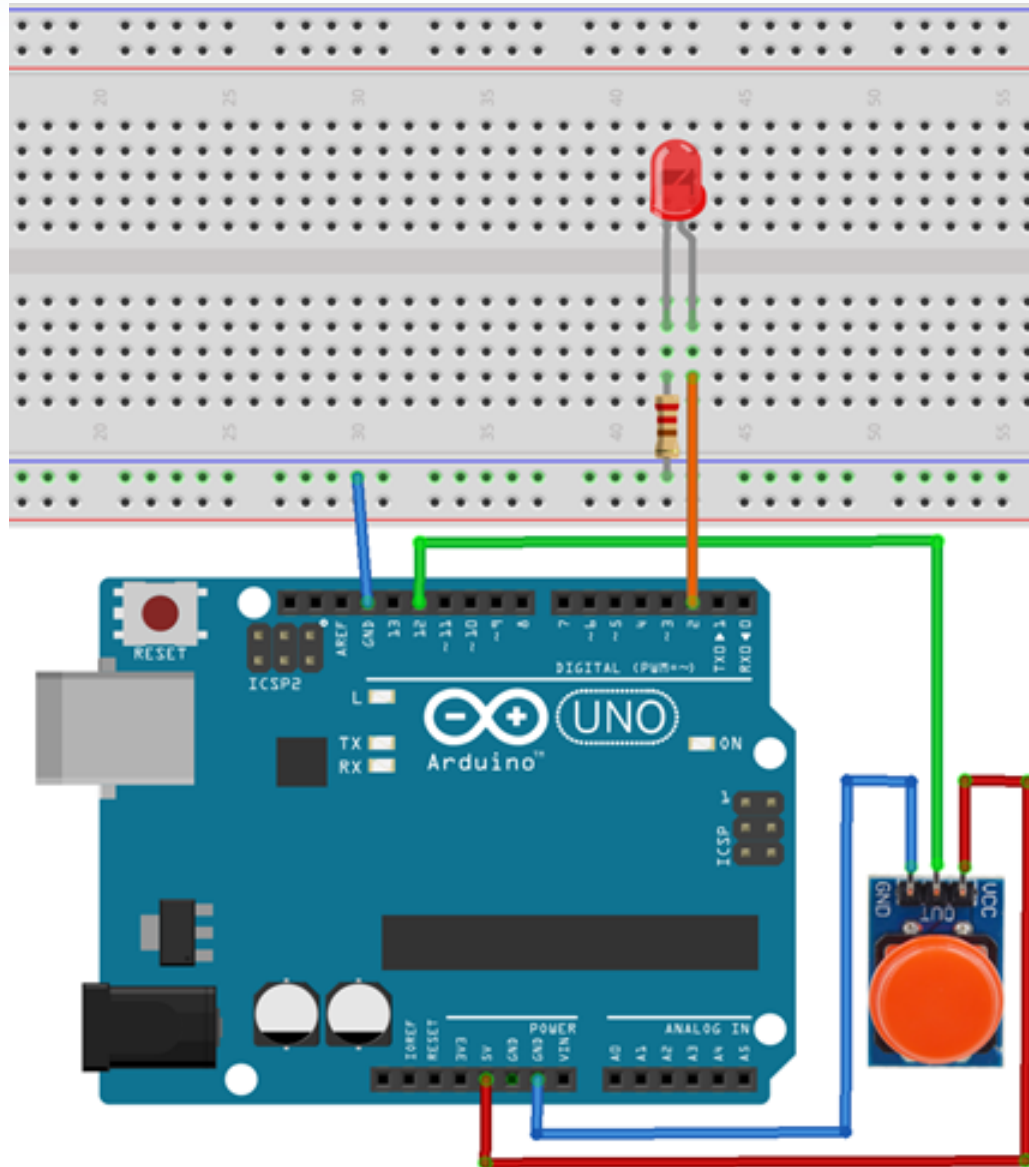


# Guided Tasks

- 1 Push-button & 1 LED



# Guided Tasks

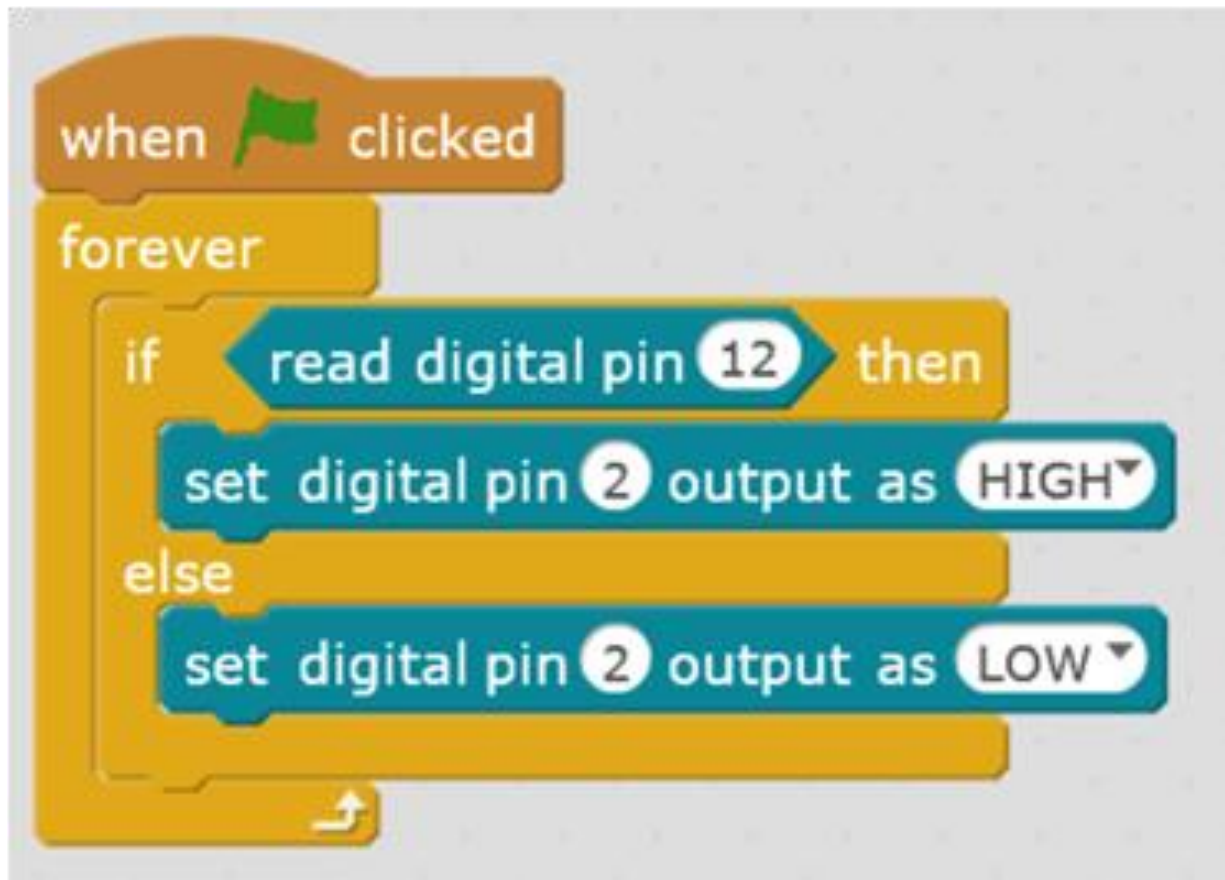


LED	Arduino
Long Leg (+)	D2 Pin
Short Leg (-)	GND Pin

Push-button	Arduino
GND Pin	GND Pin
VCC Pin	5V Pin
OUT Pin	D12 Pin

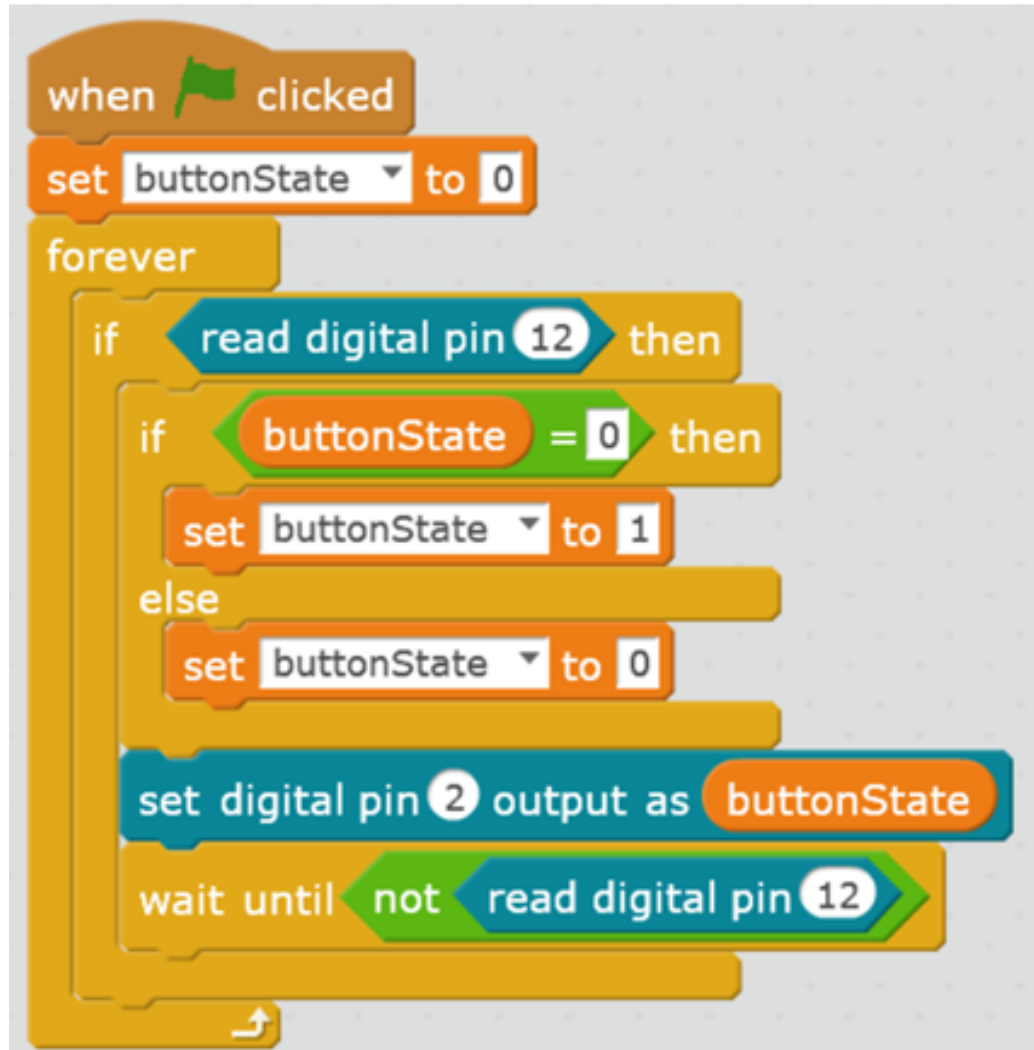
# Guided Tasks

- Using a Push-button as a Momentary Switch



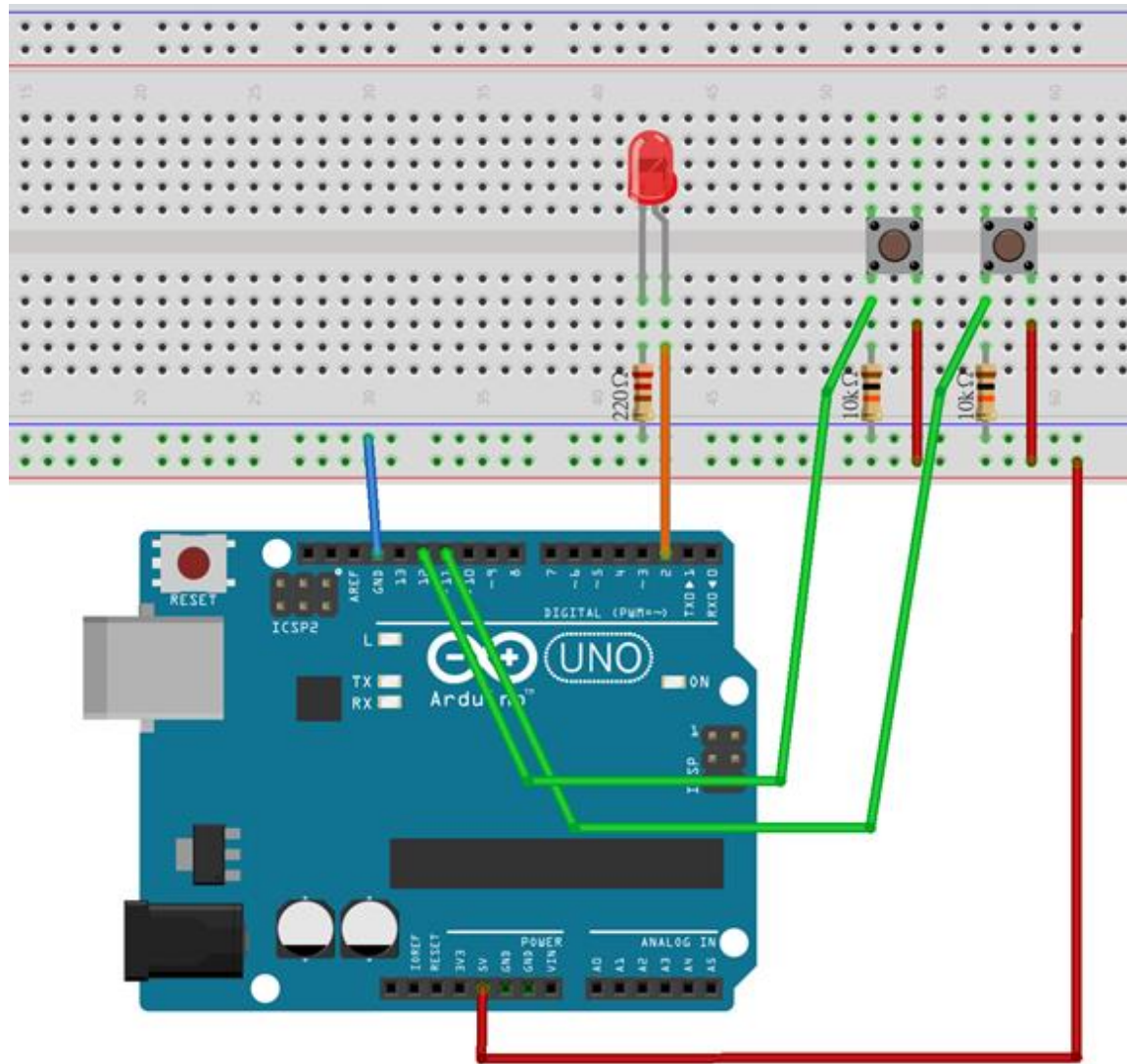
# Guided Tasks

- Using a Push-button as Toggle Switch



# Guided Tasks

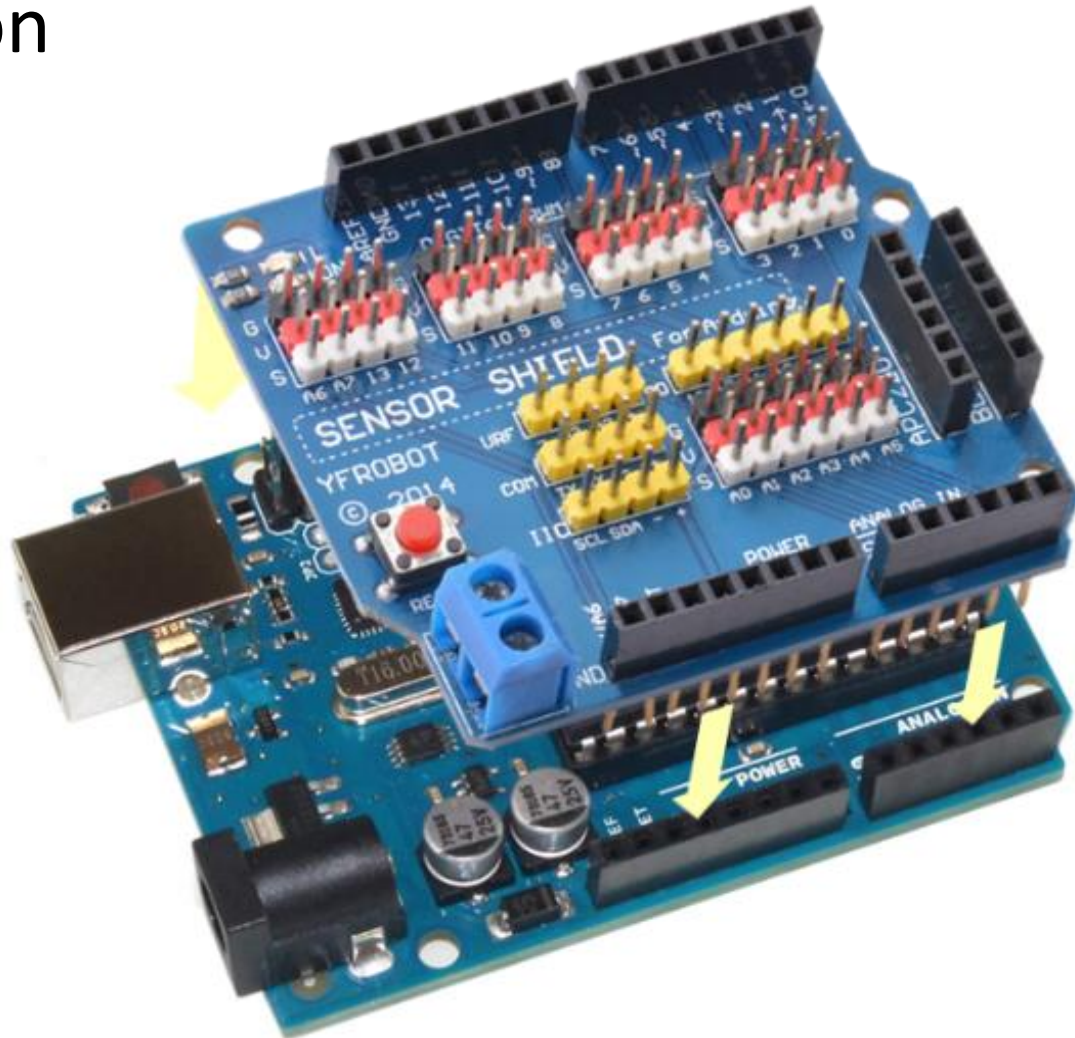
- 2 Push-buttons & 1 LED

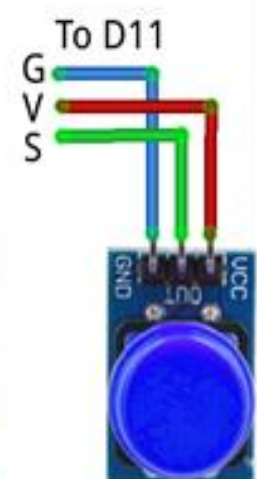
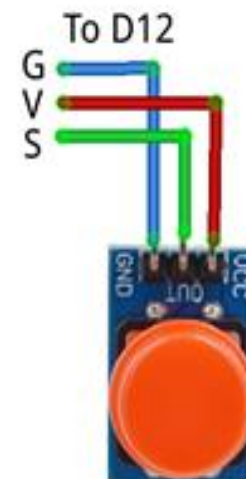
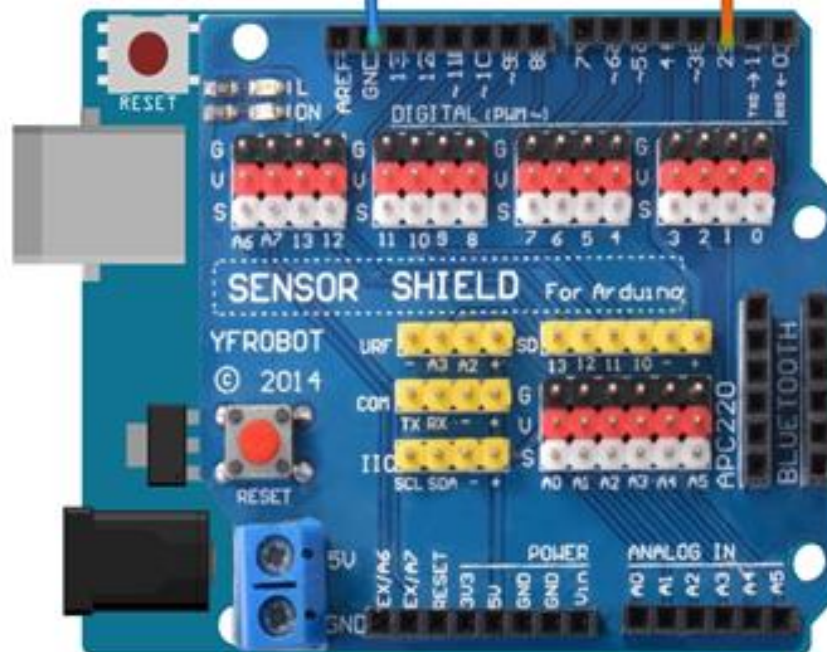
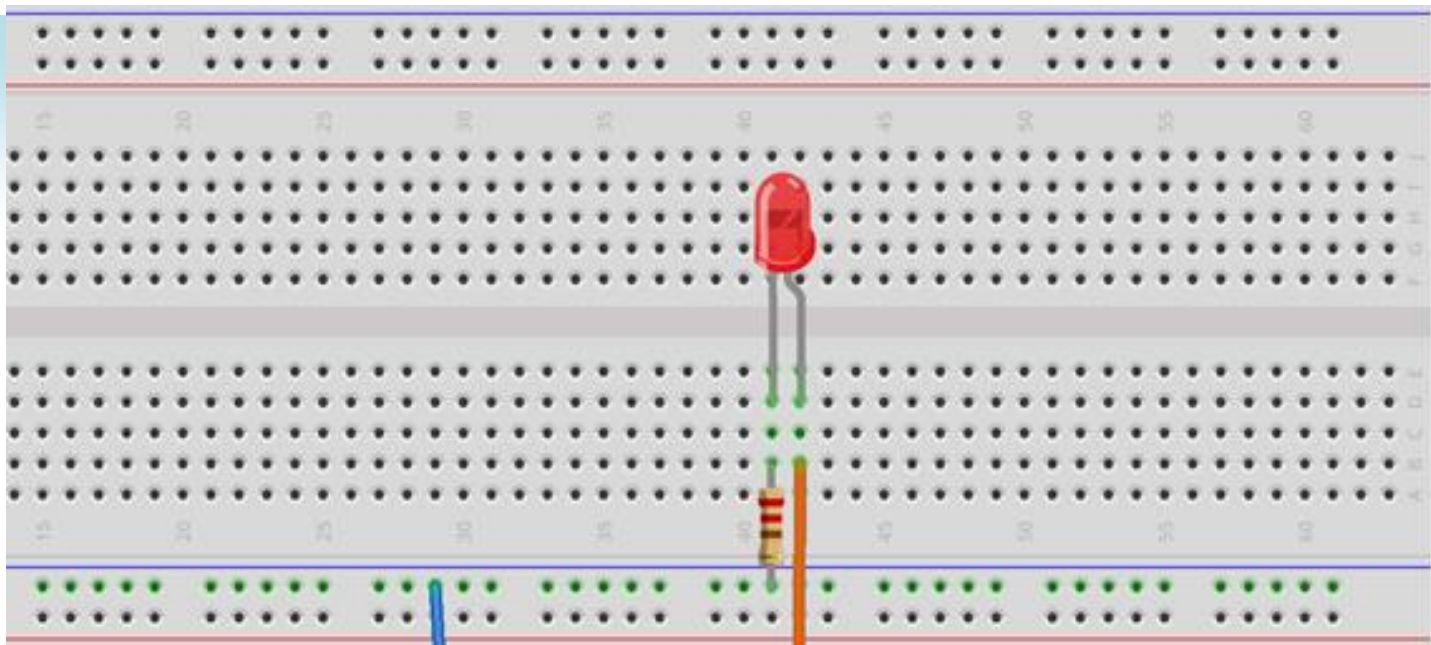




# Guided Tasks

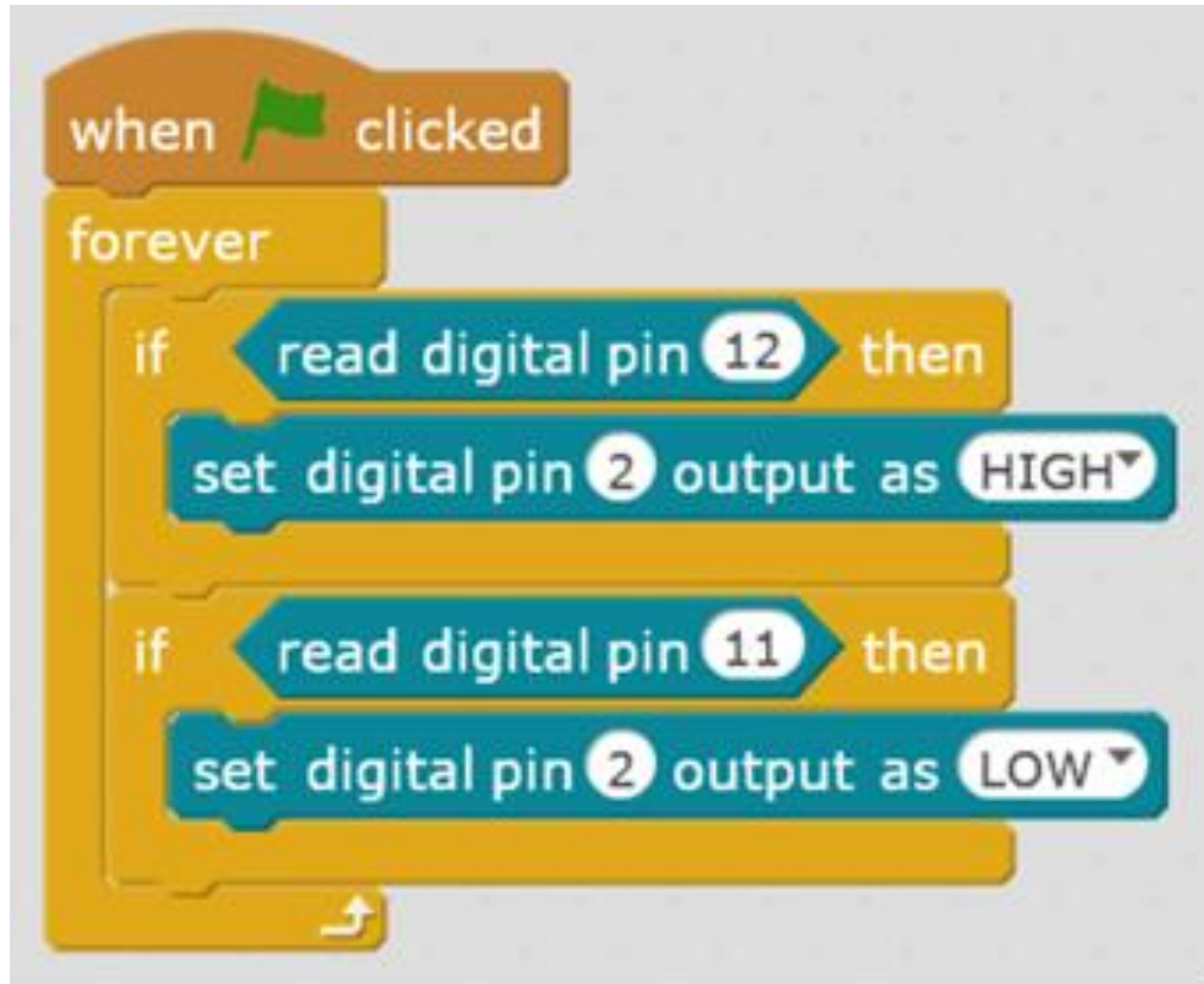
- Using Sensor Shield with GVS pins to simplify connection





# Guided Tasks

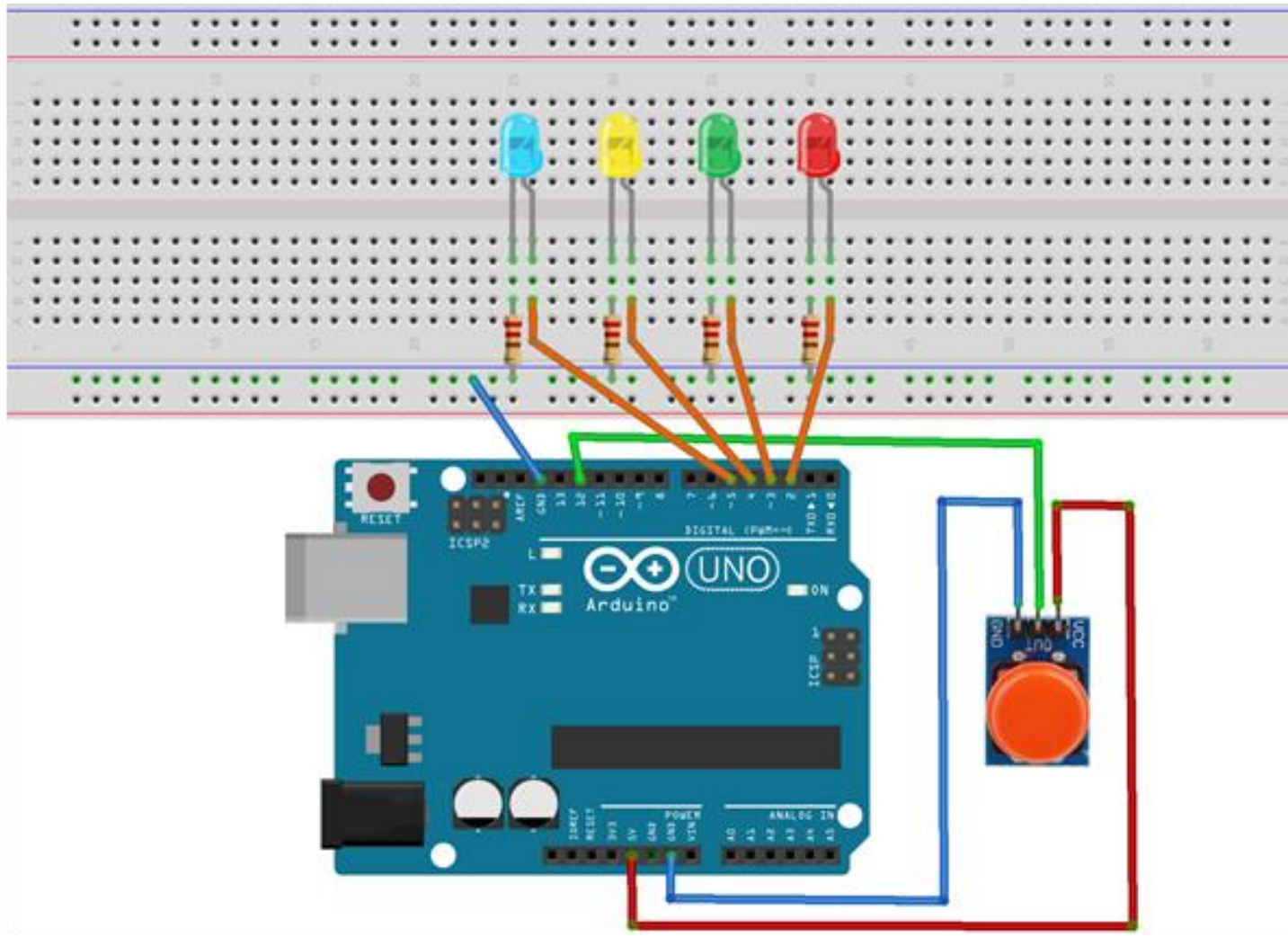
- Using two Push-buttons as On and Off Switches

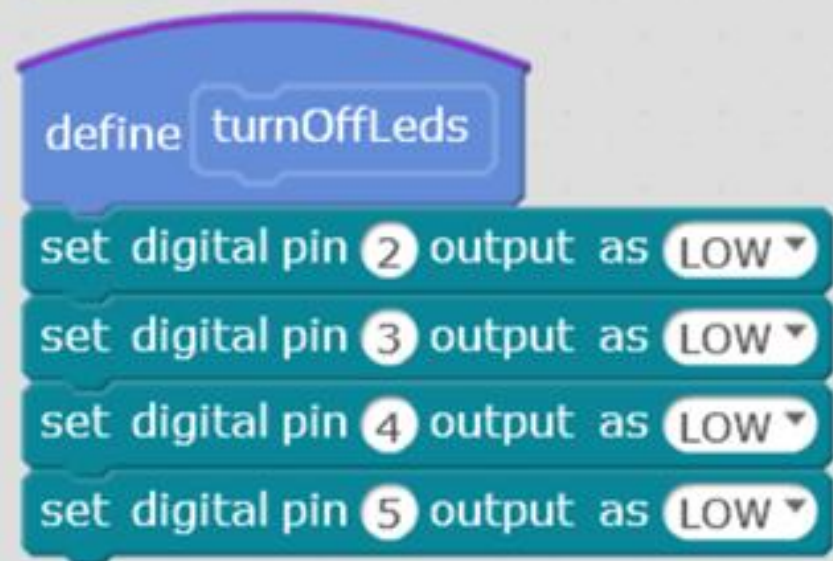
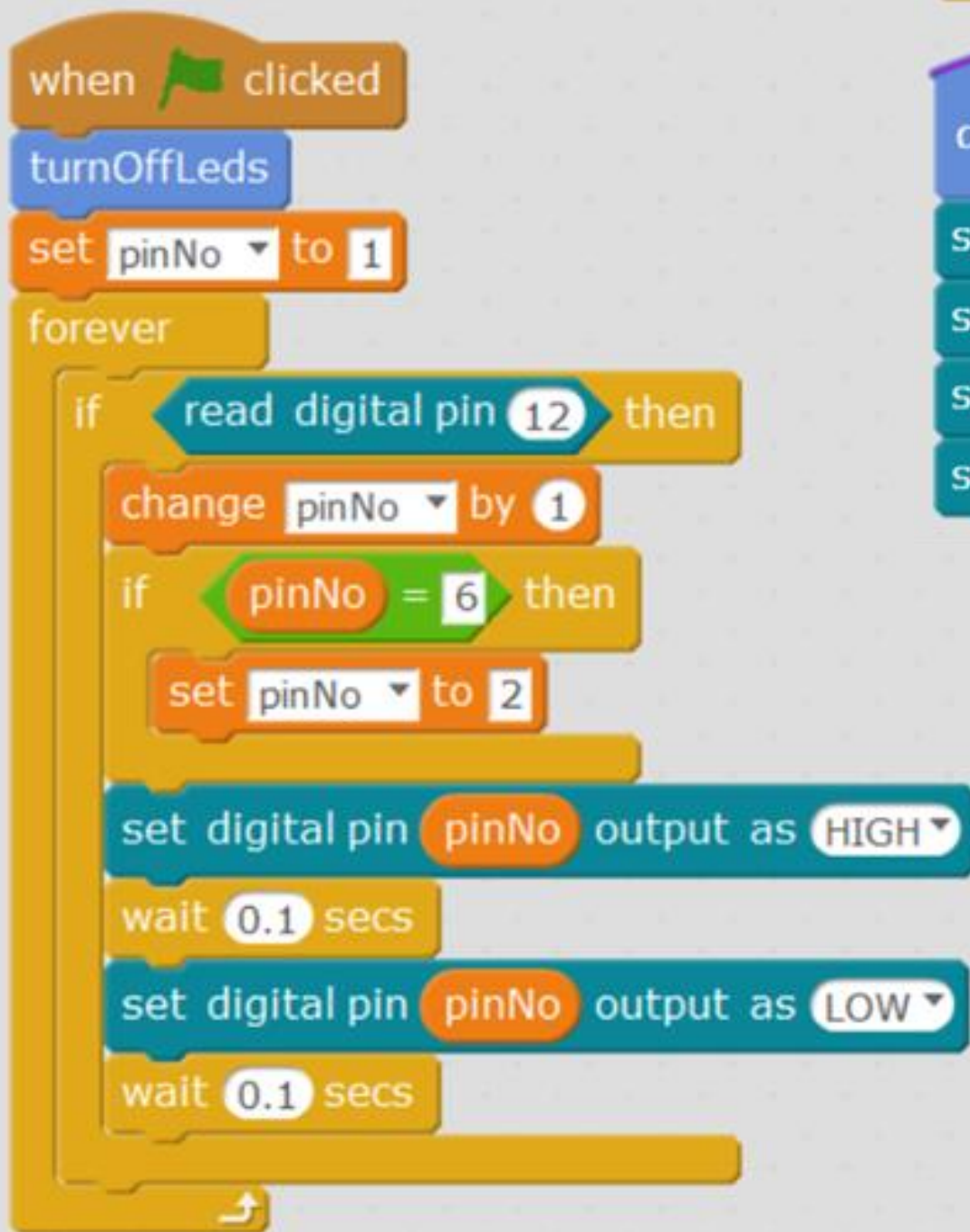




# Guided Tasks

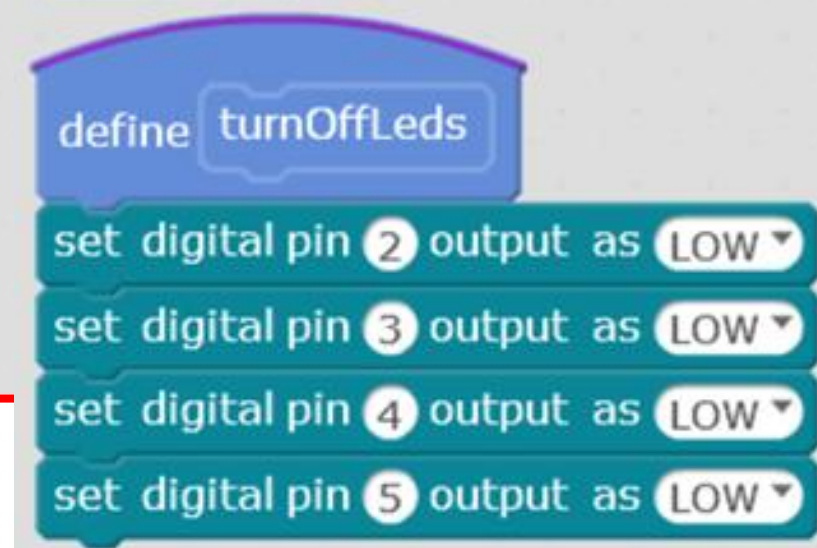
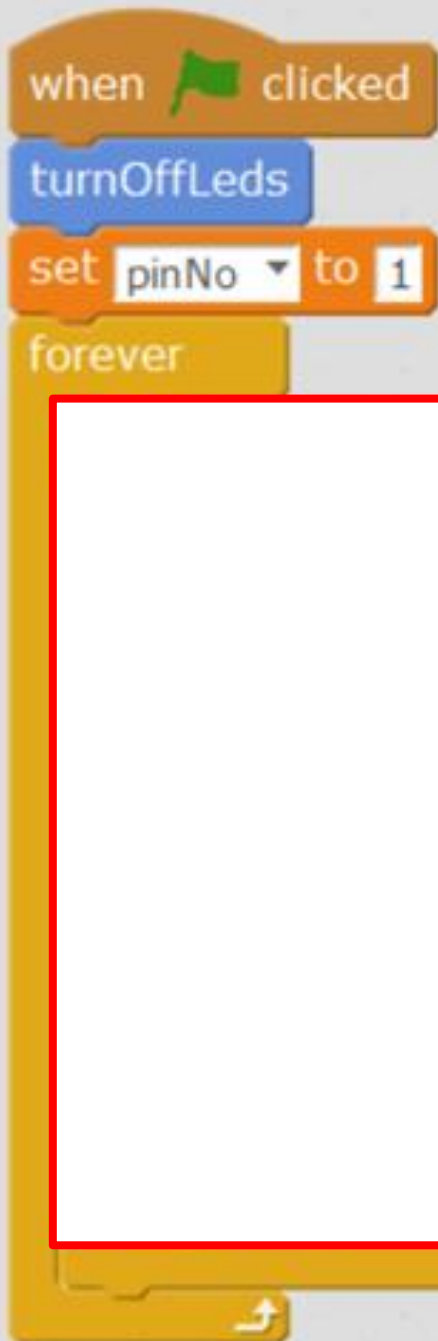
- Controlling several LEDs by a Push-button

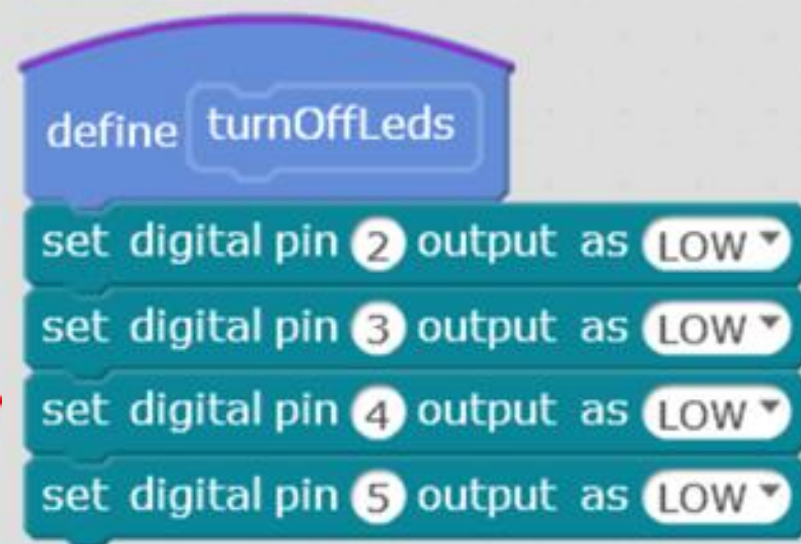
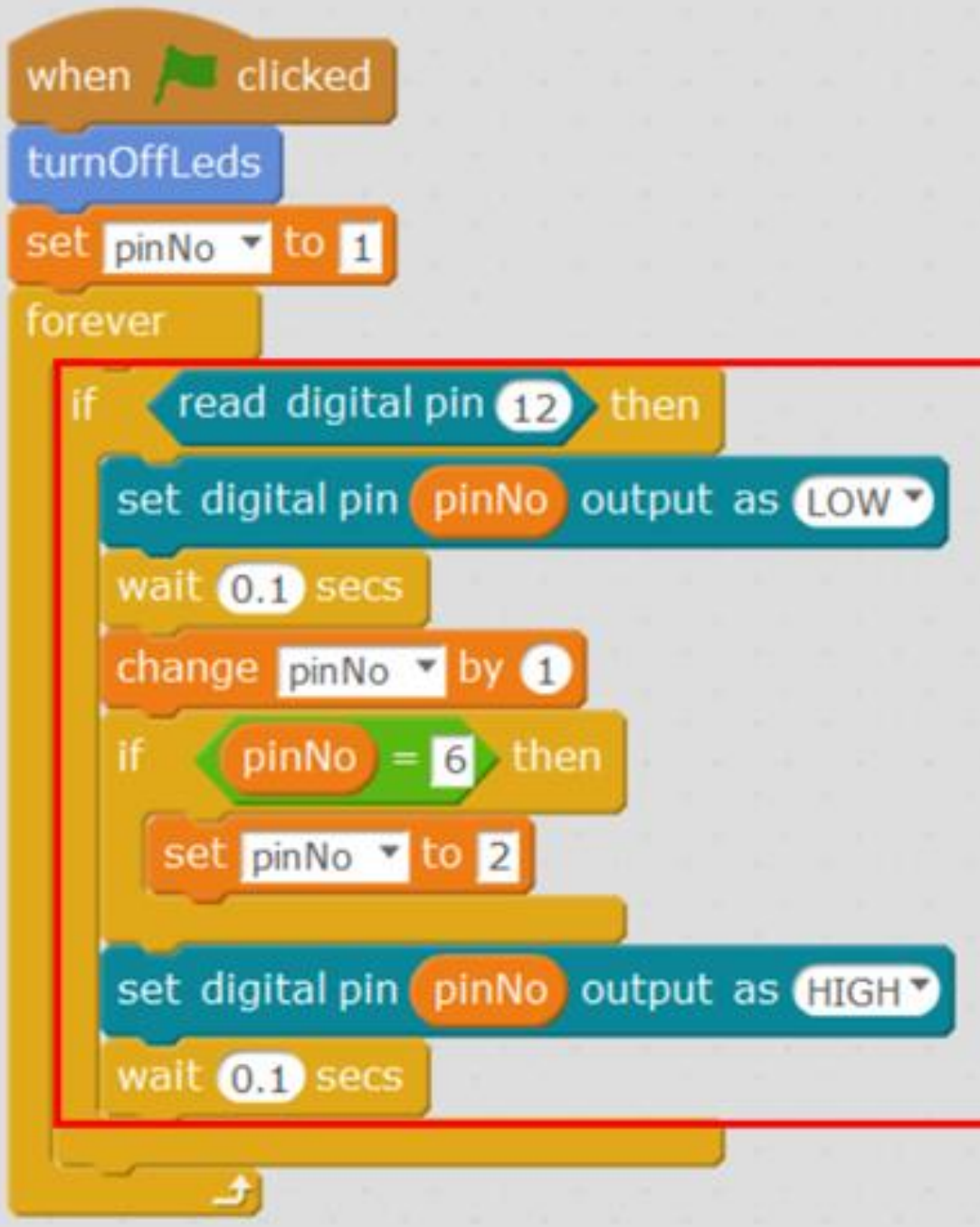




# Guided Tasks

- Referring to the previous task, when the push button is pressed, the LEDs will blink. When the button is released, the LEDs will stop blinking and all the LEDs will be turned off.
- **Question:**  
Modify the program so that when the button is released, the LEDs will stop blinking and only one LED will remain on.

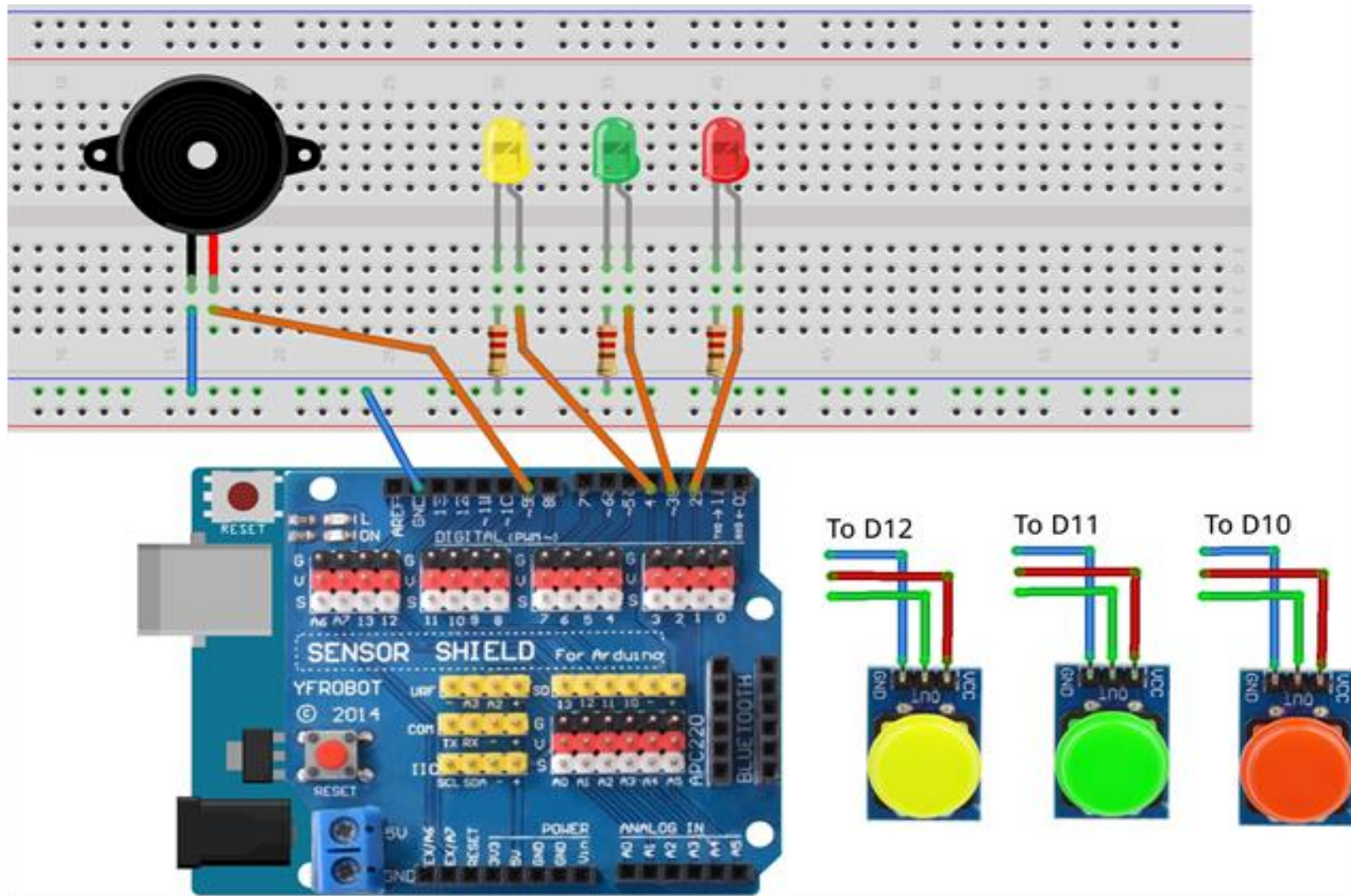


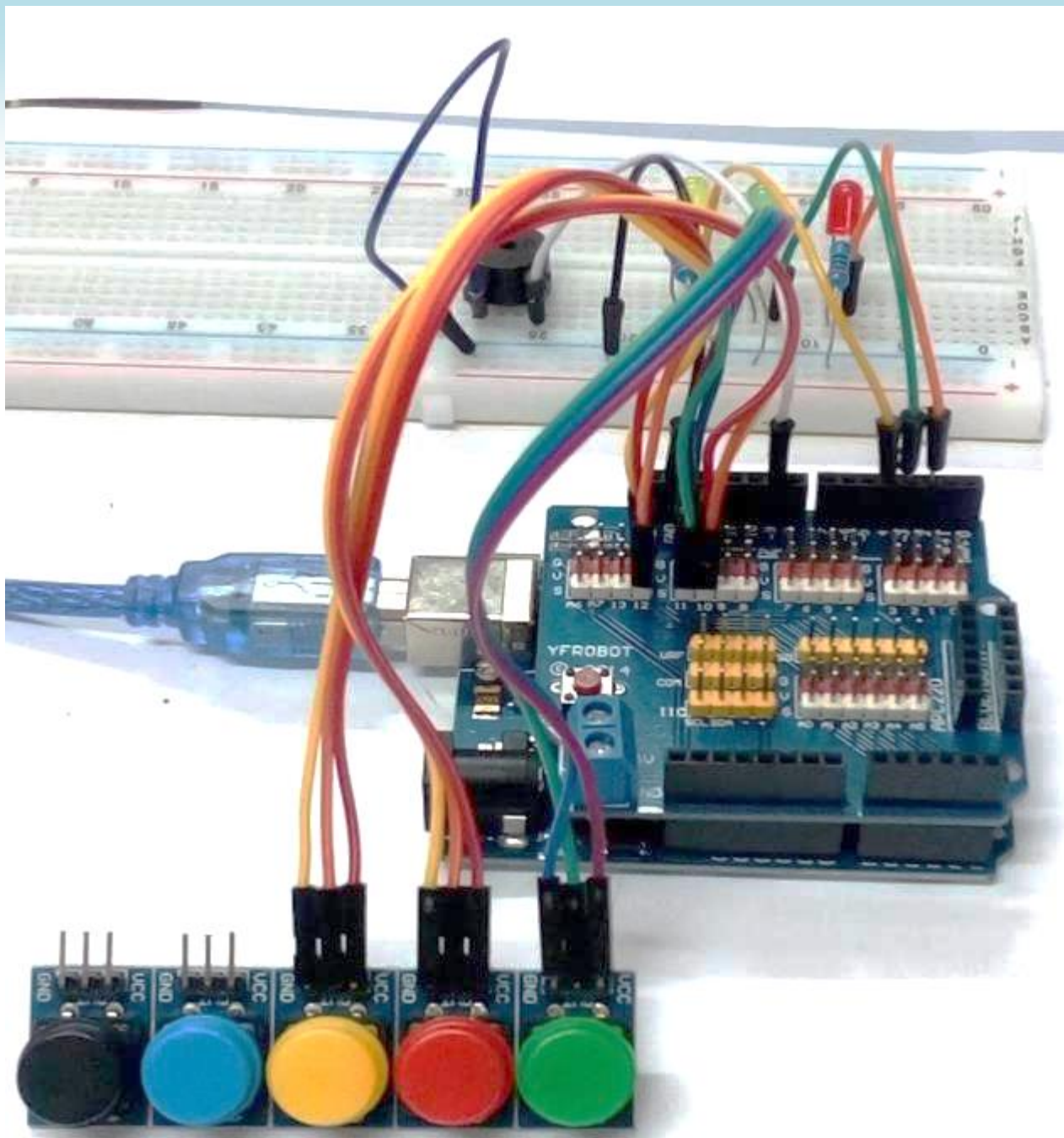




# Guided Tasks

- Designing a Fast Answer Game System (搶答遊戲)





# Guided Tasks

- Step-by-Step
  1. Turn an LED on when pressing the corresponding push-button.
  2. Ensure that only one LED is turned on at a time.
  3. Produce suitable sound effect when pressing each button.
  4. Make the program more modular





when  clicked

forever

if  read digital pin 11 and  read digital pin 12 then

if read digital pin 10 then

set digital pin 2 output as HIGH

else

set digital pin 2 output as LOW

if  read digital pin 10 and  read digital pin 12 then

if read digital pin 11 then

set digital pin 3 output as HIGH

else

set digital pin 3 output as LOW

if  read digital pin 11 and  read digital pin 10 then

if read digital pin 12 then

set digital pin 4 output as HIGH

else

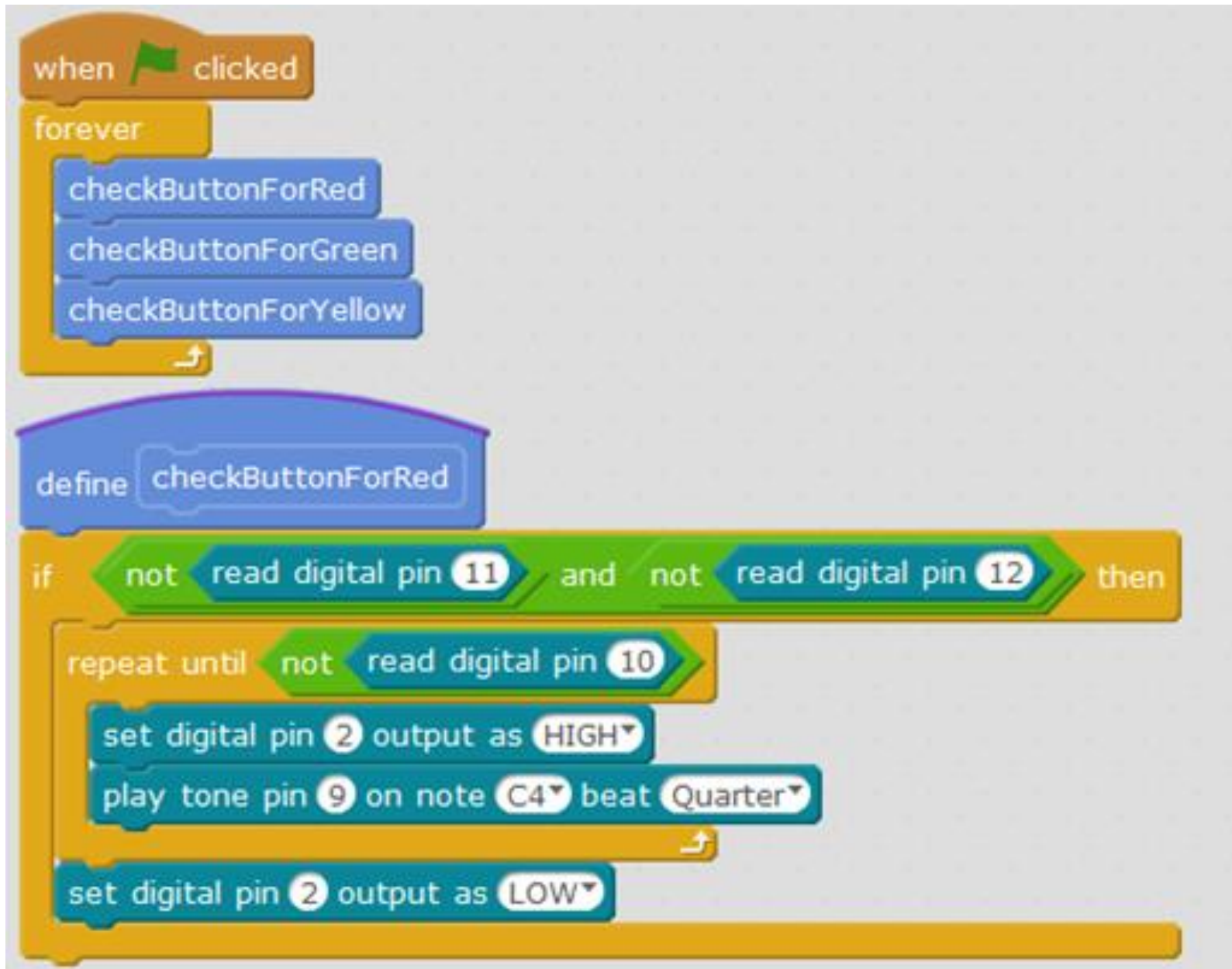
set digital pin 4 output as LOW











define checkButtonForGreen

if not read digital pin 10 and not read digital pin 12 then

repeat until not read digital pin 11

set digital pin 3 output as HIGH

play tone pin 9 on note D4 beat Quarter

set digital pin 3 output as LOW

define checkButtonForYellow

if not read digital pin 11 and not read digital pin 10 then

repeat until not read digital pin 12

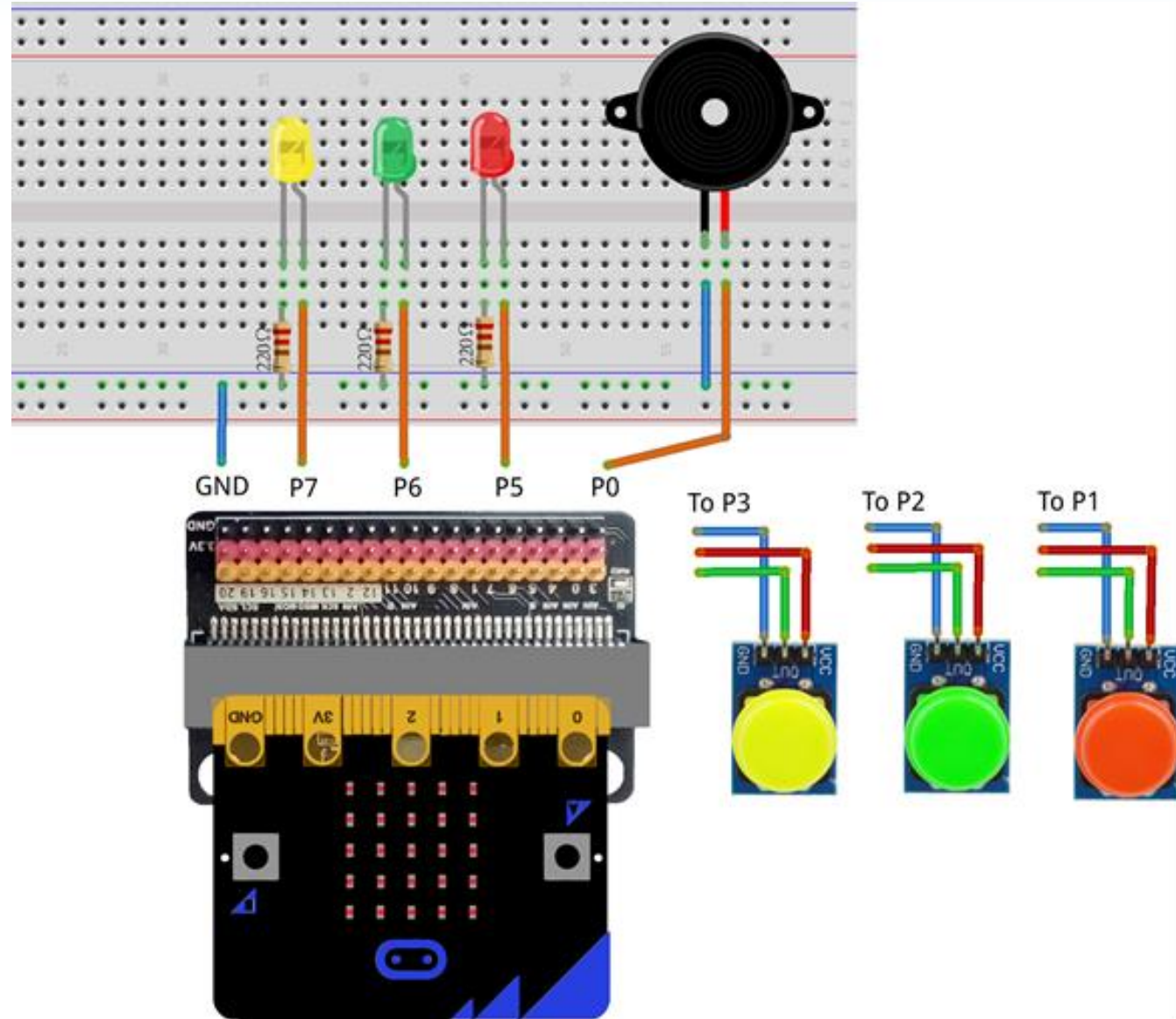
set digital pin 4 output as HIGH

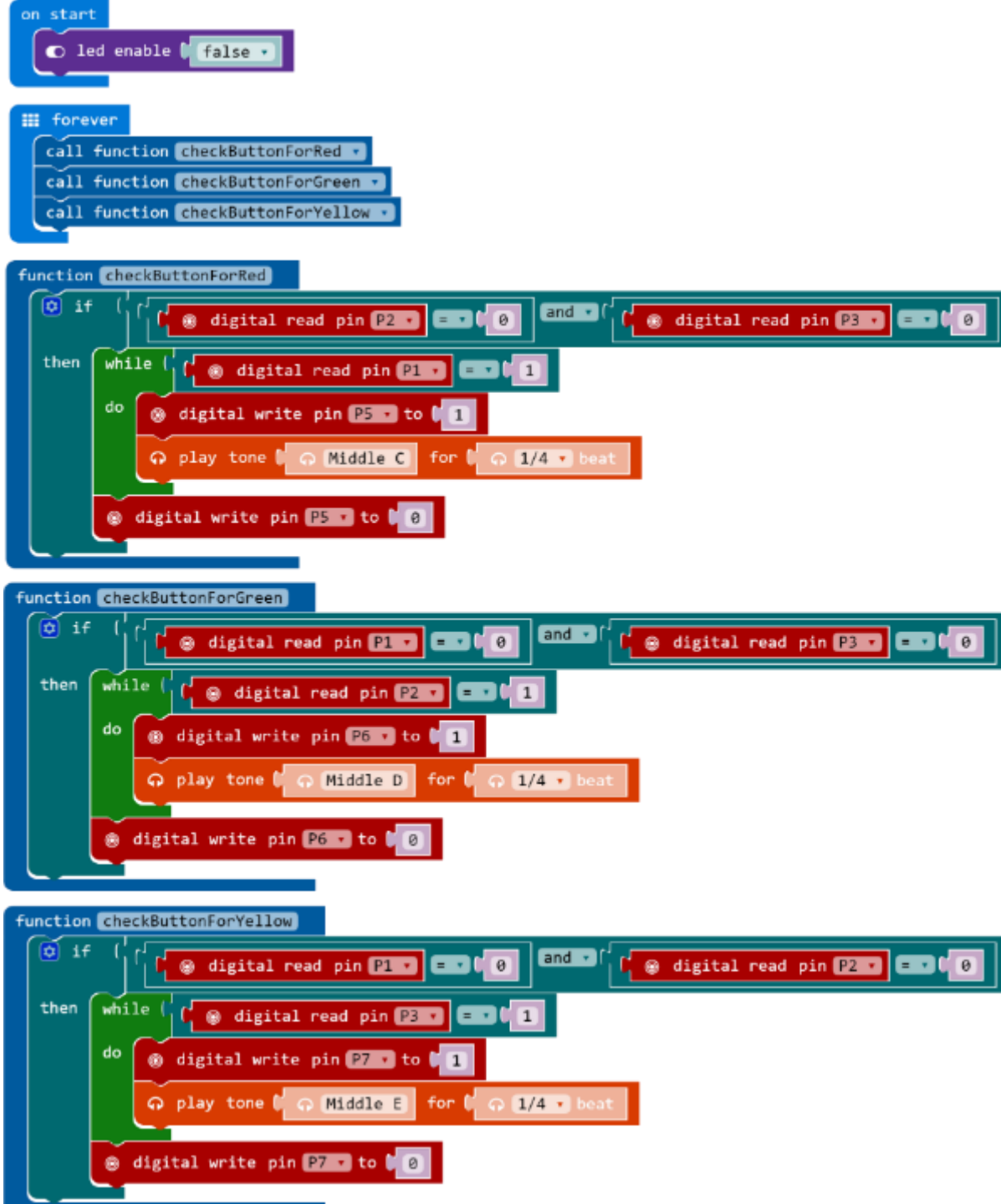
play tone pin 9 on note E4 beat Quarter

set digital pin 4 output as LOW

# Guided Tasks

- For micro:bit

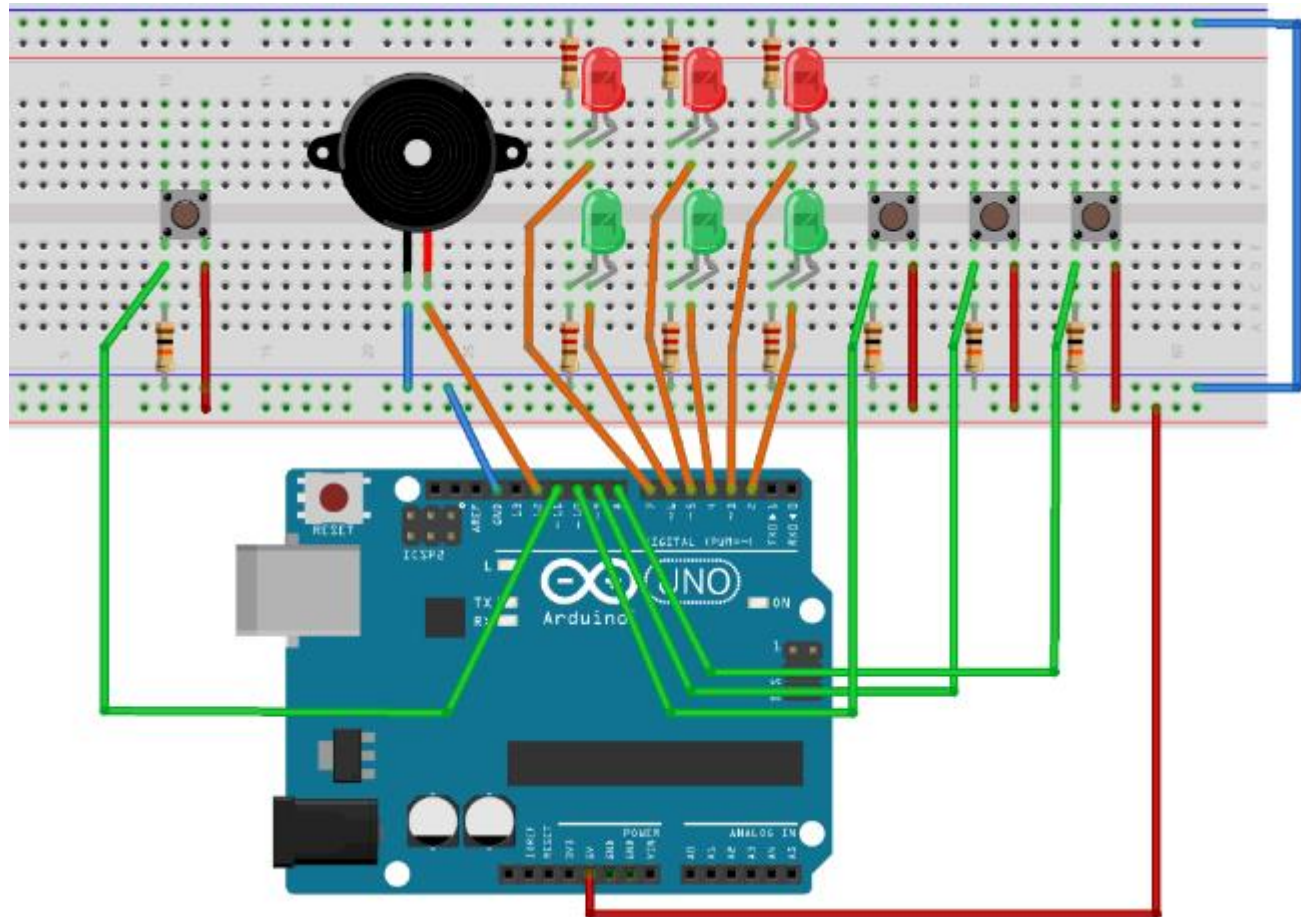




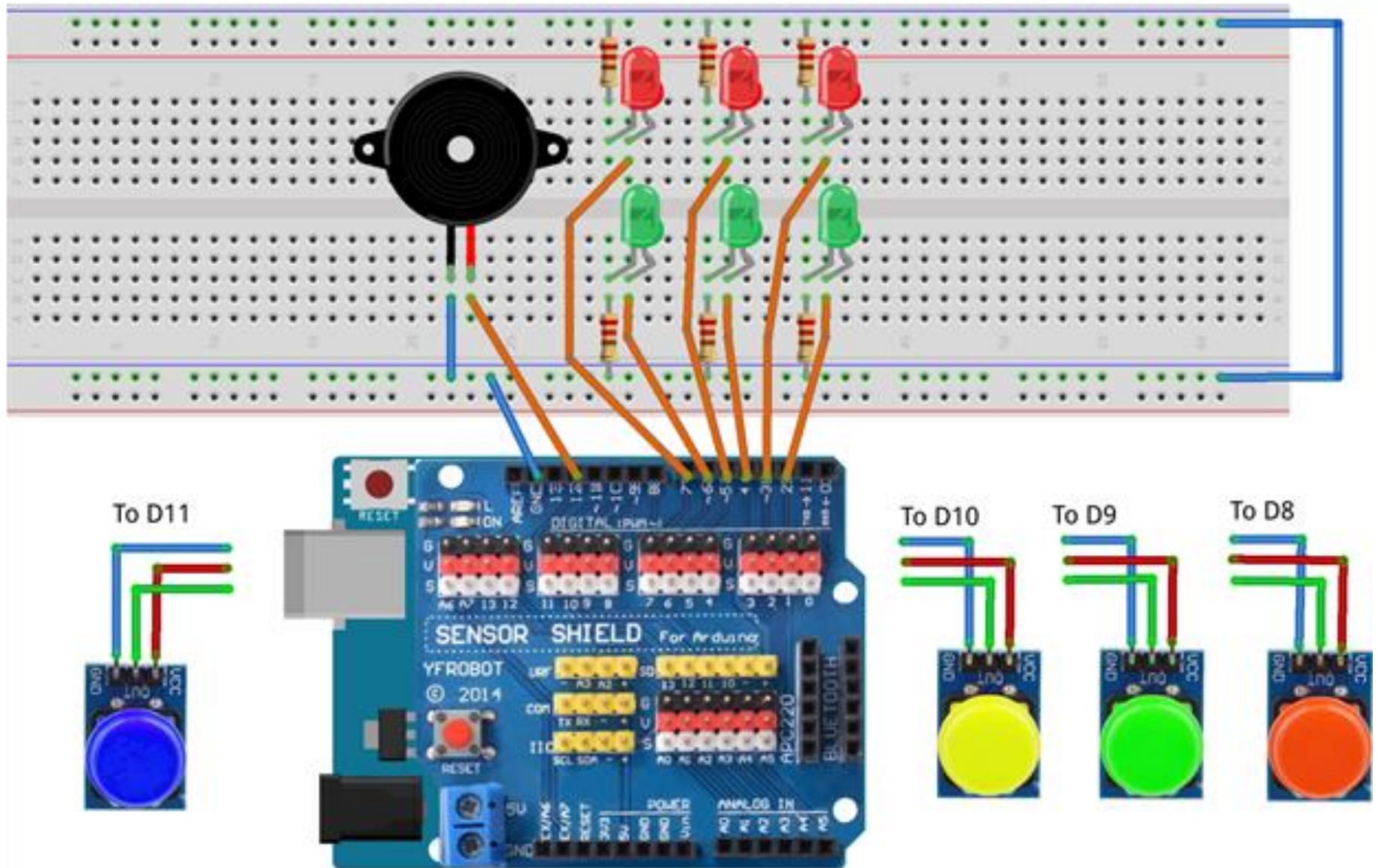


# Self-Exploration Project

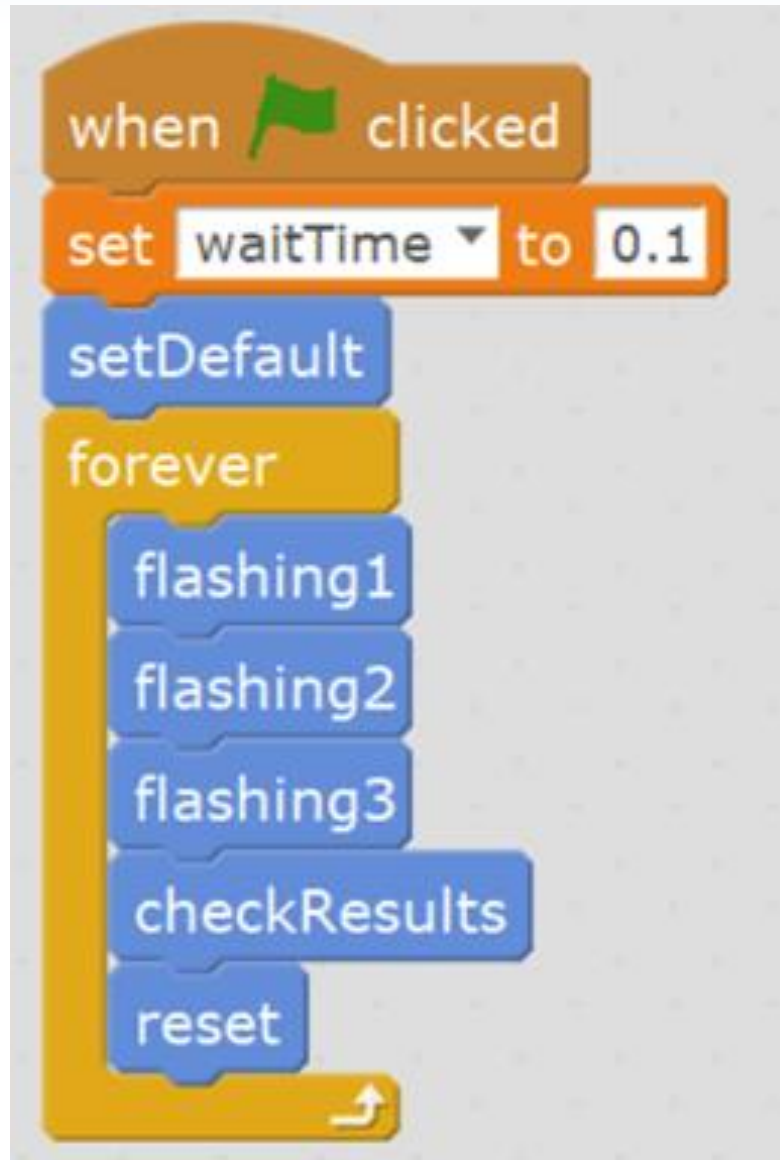
- Designing a simple interactive game using push-buttons, LEDs and buzzer.



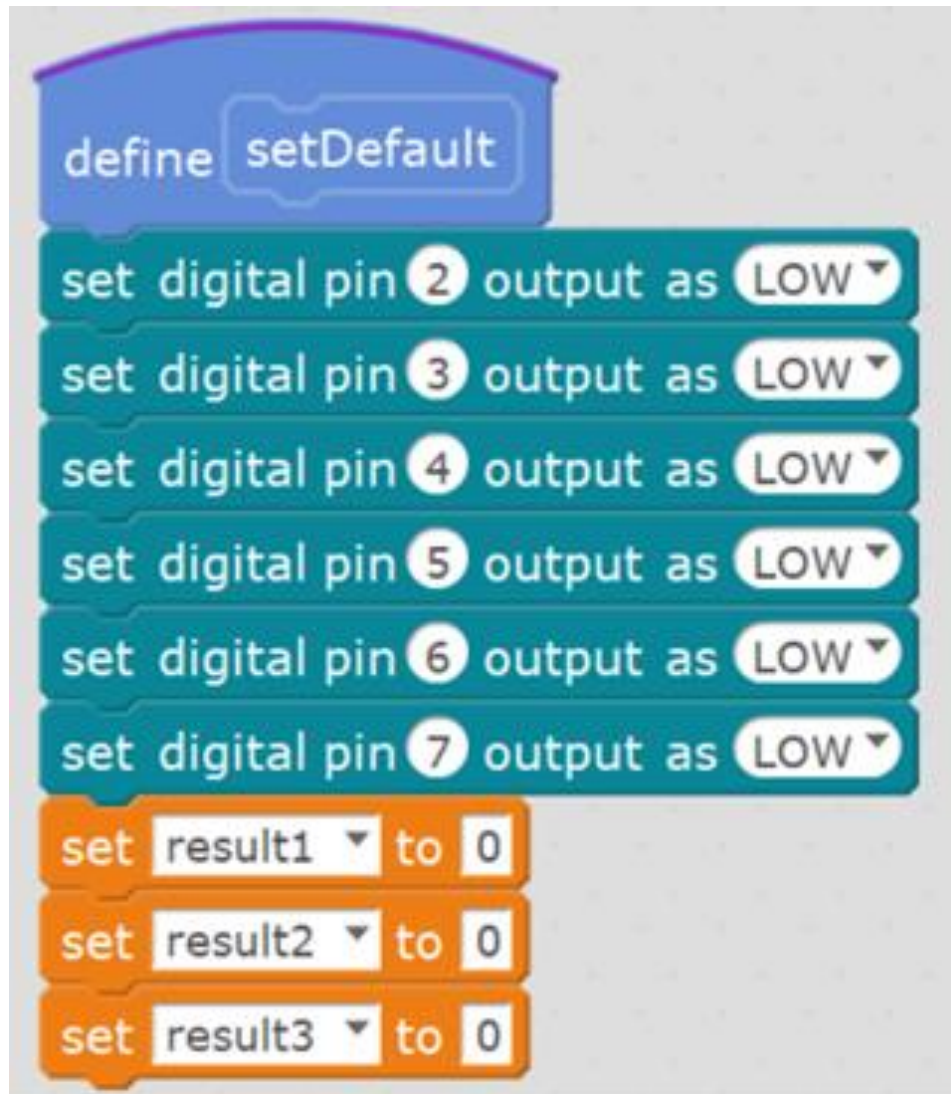
# Self-Exploration Project



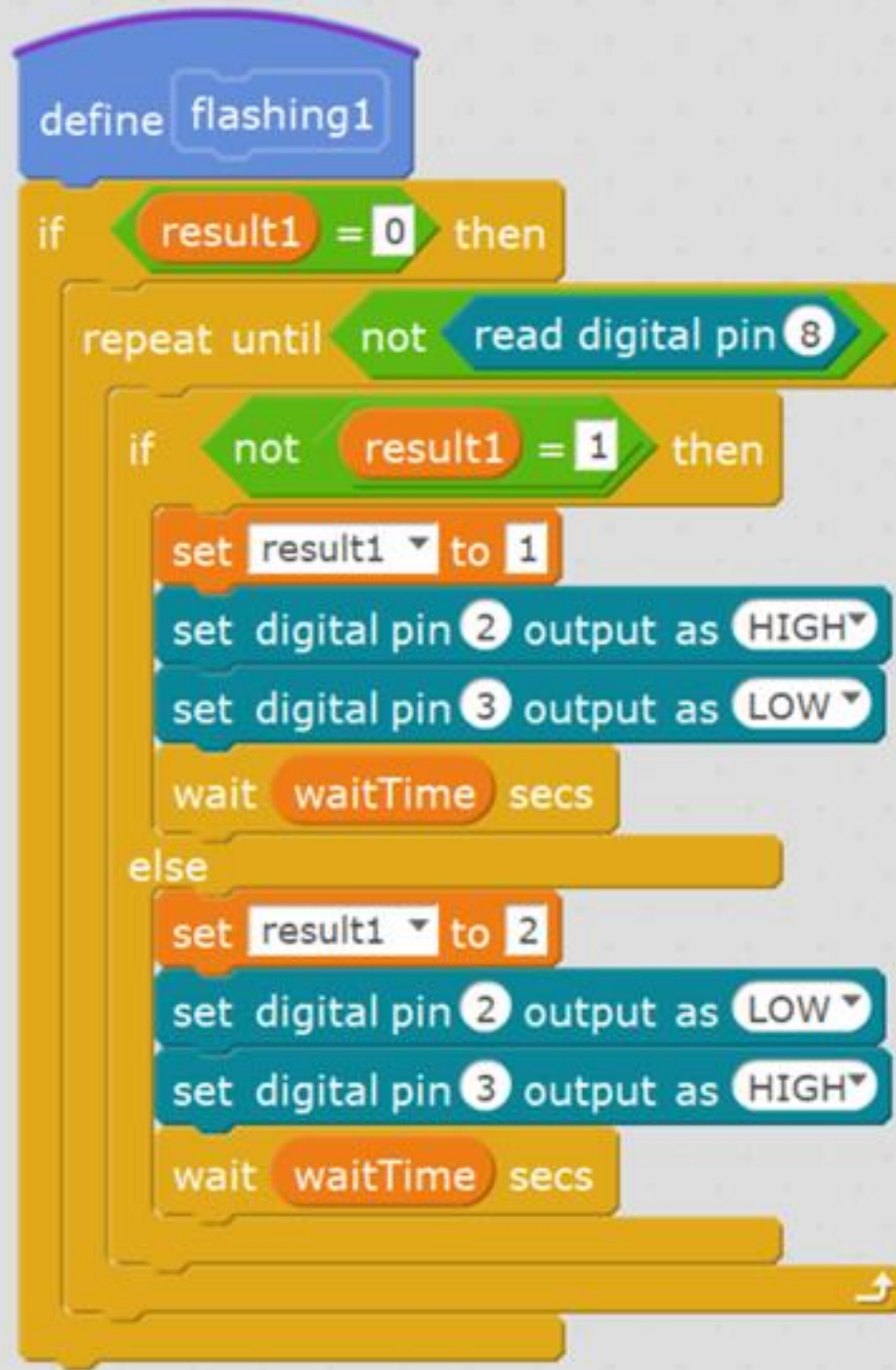
# Self-Exploration Project



# Self-Exploration Project







```

when clicked
set waitTime to 0.1
setDefault
forever
  Flashing1
  Flashing2
  Flashing3
  checkResults
  reset

```

```

define setDefault
set pinNo to 1
repeat 6
  change pinNo by 1
  set digital pin pinNo output as LOW

```

```

set result1 to 0
set result2 to 0
set result3 to 0

```

```

define reset
if read digital pin 11 then
  setDefault

```

```

define Flashing1
if result1 = 0 then
  repeat until not read digital pin 8
    if not result1 = 1 then
      set result1 to 1
      set digital pin 2 output as HIGH
      set digital pin 3 output as LOW
      wait waitTime secs
    else
      set result1 to 2
      set digital pin 2 output as LOW
      set digital pin 3 output as HIGH
      wait waitTime secs

```

```

define Flashing2
if result2 = 0 then
  repeat until not read digital pin 9
    if not result2 = 1 then
      set result2 to 1
      set digital pin 4 output as HIGH
      set digital pin 5 output as LOW
      wait waitTime secs
    else
      set result2 to 2
      set digital pin 4 output as LOW
      set digital pin 5 output as HIGH
      wait waitTime secs

```

```

define Flashing3
if result3 = 0 then
  repeat until not read digital pin 10
    if not result3 = 1 then
      set result3 to 1
      set digital pin 6 output as HIGH
      set digital pin 7 output as LOW
      wait waitTime secs
    else
      set result3 to 2
      set digital pin 6 output as LOW
      set digital pin 7 output as HIGH
      wait waitTime secs

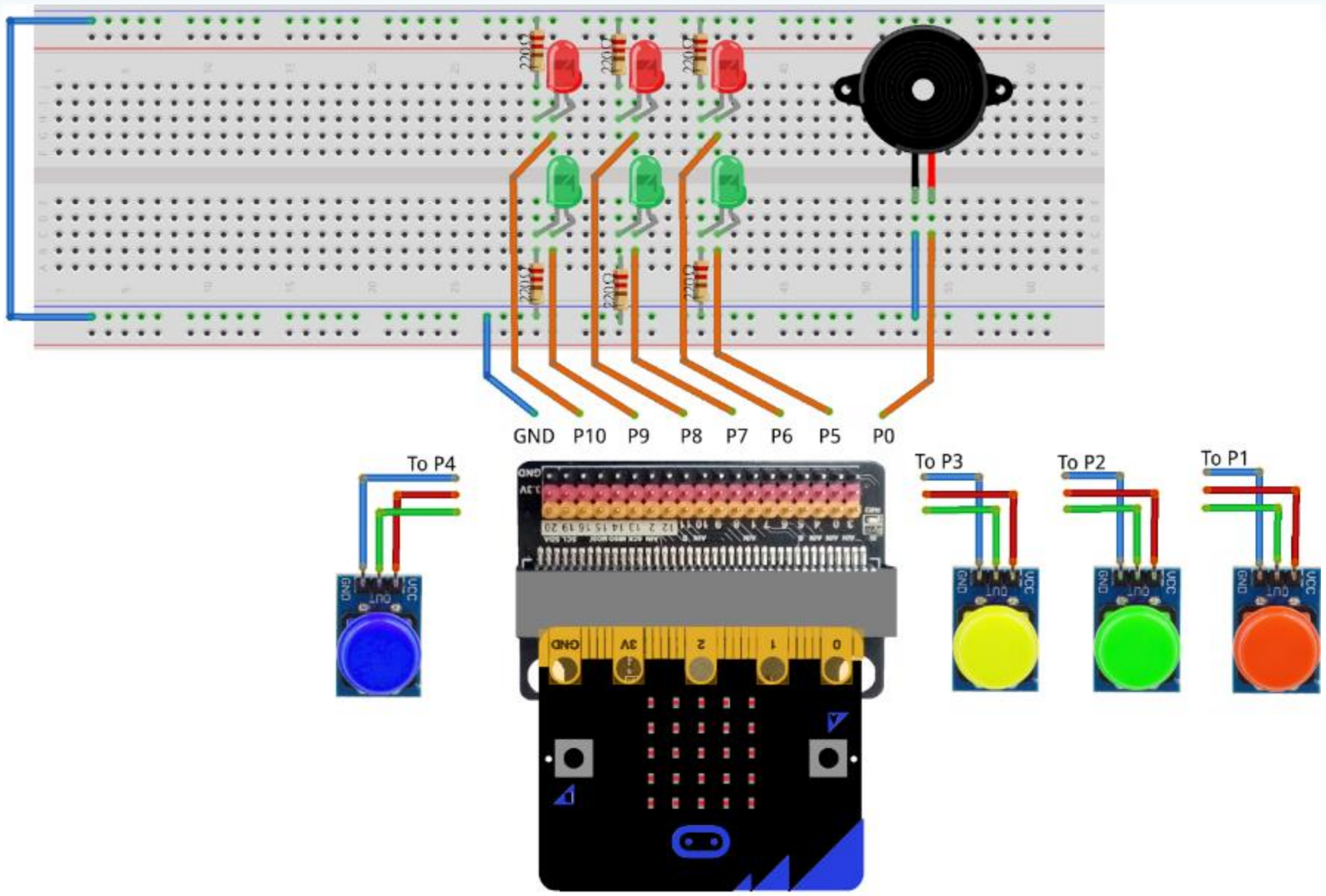
```

```

define checkResults
if not result1 = 0 and not result2 = 0 and not result3 = 0 then
  if result1 = result2 and result2 = result3 then
    play tone pin 12 on note C7 beat Quarter
    play tone pin 12 on note D7 beat Quarter
    play tone pin 12 on note E7 beat Quarter
  else
    play tone pin 12 on note C2 beat Quarter
    play tone pin 12 on note D2 beat Quarter
    play tone pin 12 on note E2 beat Quarter

```

# micro:bit version



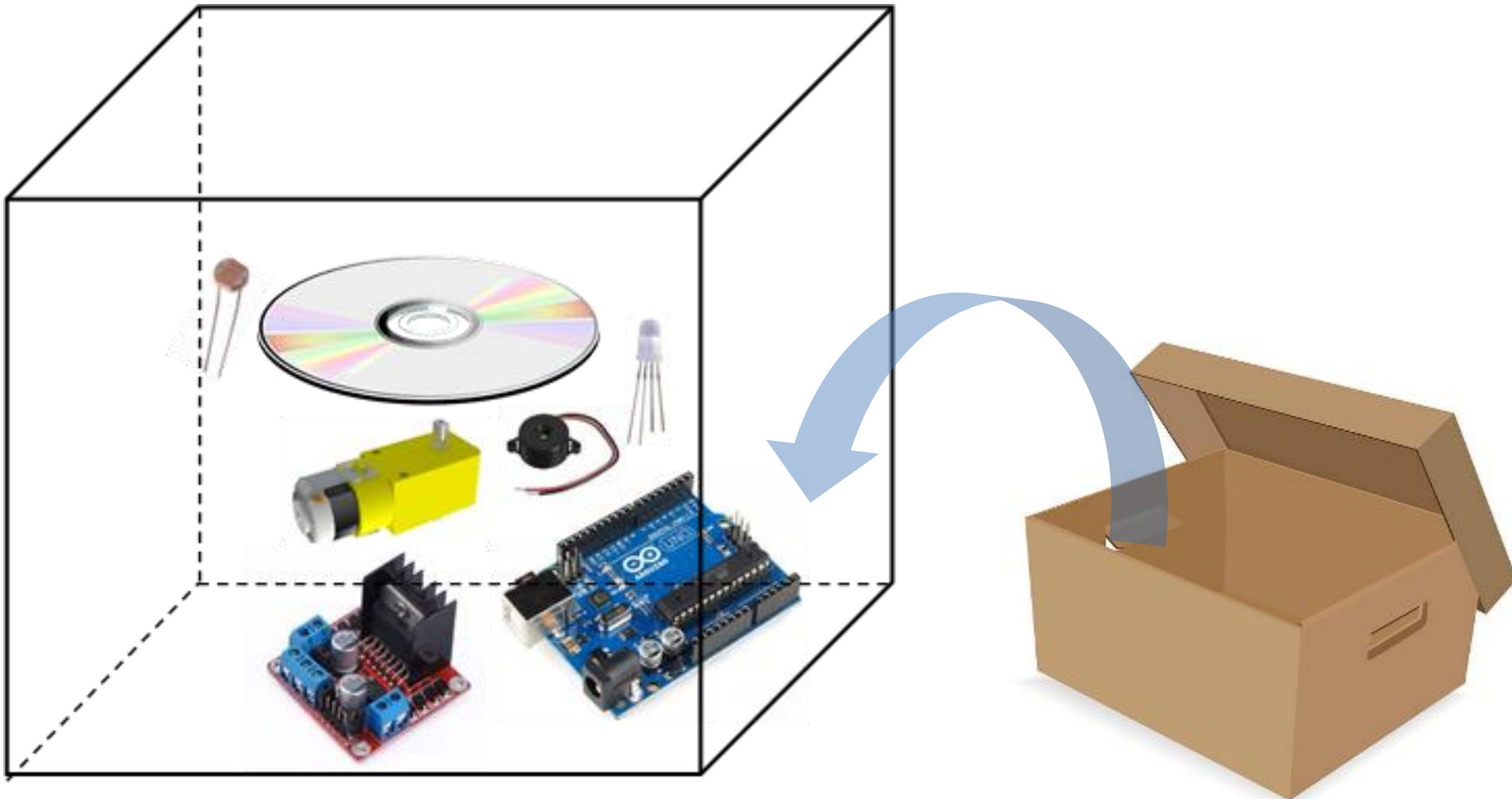
# Project 3

A Light-Control Music Box



# Project Requirements

- Design a light-control music box.

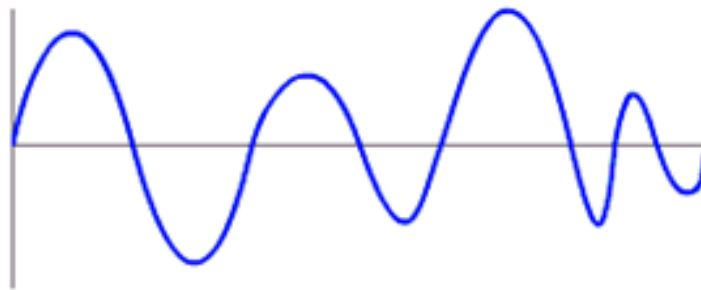


# Knowledge and Skills

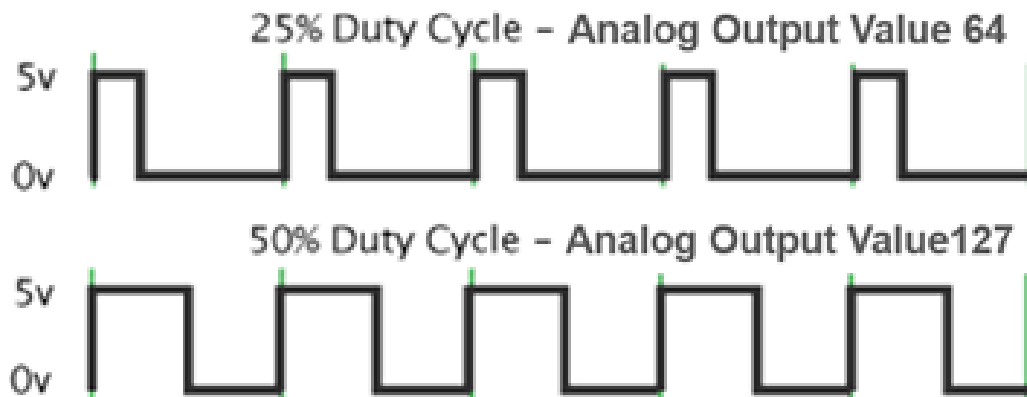
- Construct simple circuits with single-board microcontroller, potentiometer, RGB LED, light sensor (LDR) and DC motor, etc.
- Know how to handle **analog input** in microcontroller
- Know how to handle **analog output (PWM)** in microcontroller
- Create simple light sensor control systems

# Basic Knowledge

- Analog Input



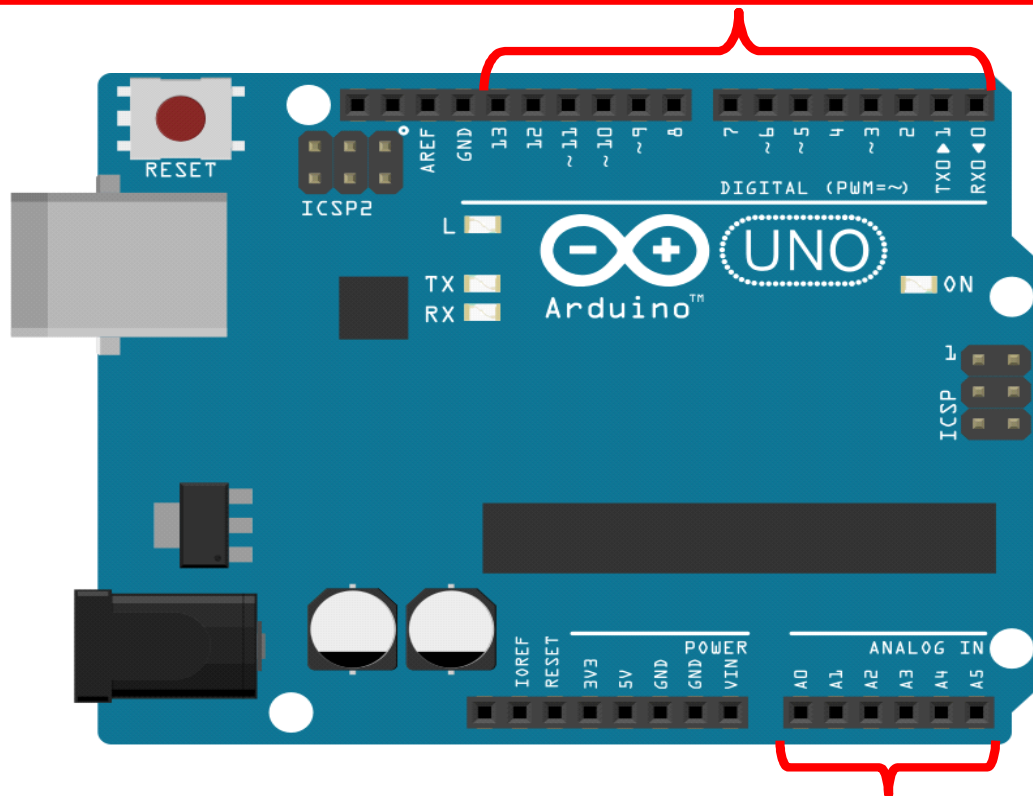
- Analog (PWM) Output



# Basic Knowledge

- Analog Input in Arduino

**Digital Input / Output pins D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13**

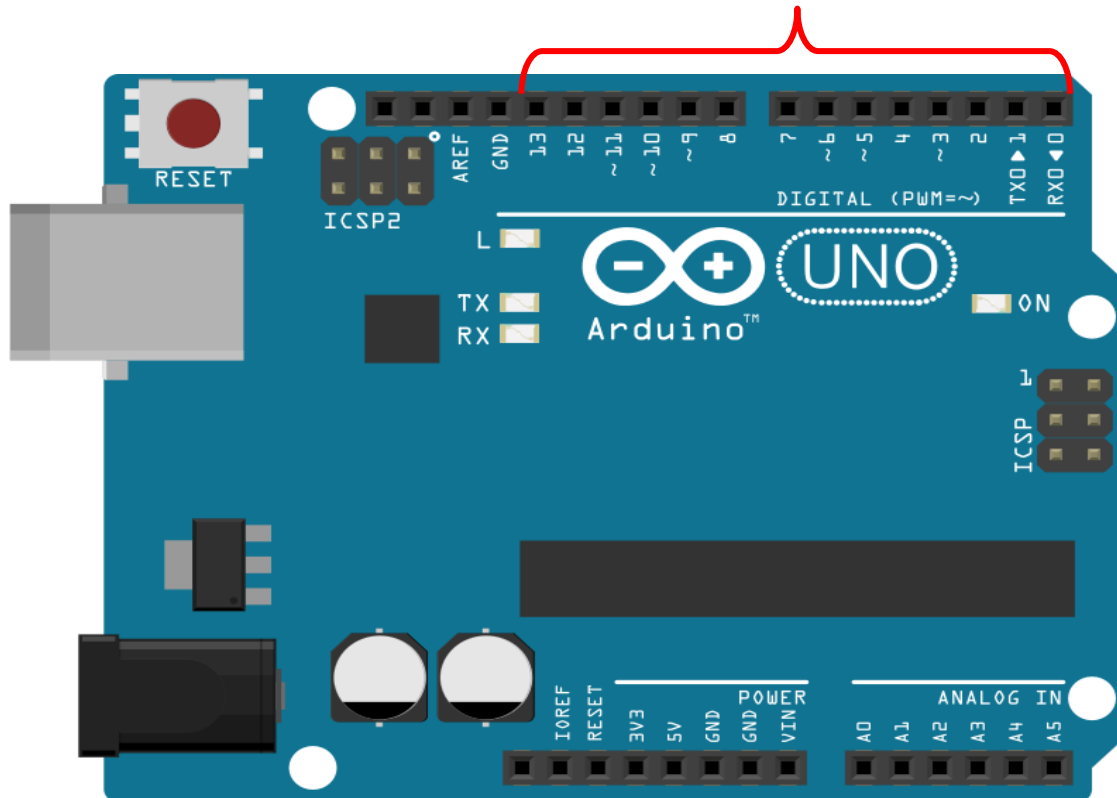


**Analog Input pins A0, A1, A2, A3, A4, A5**

# Basic Knowledge

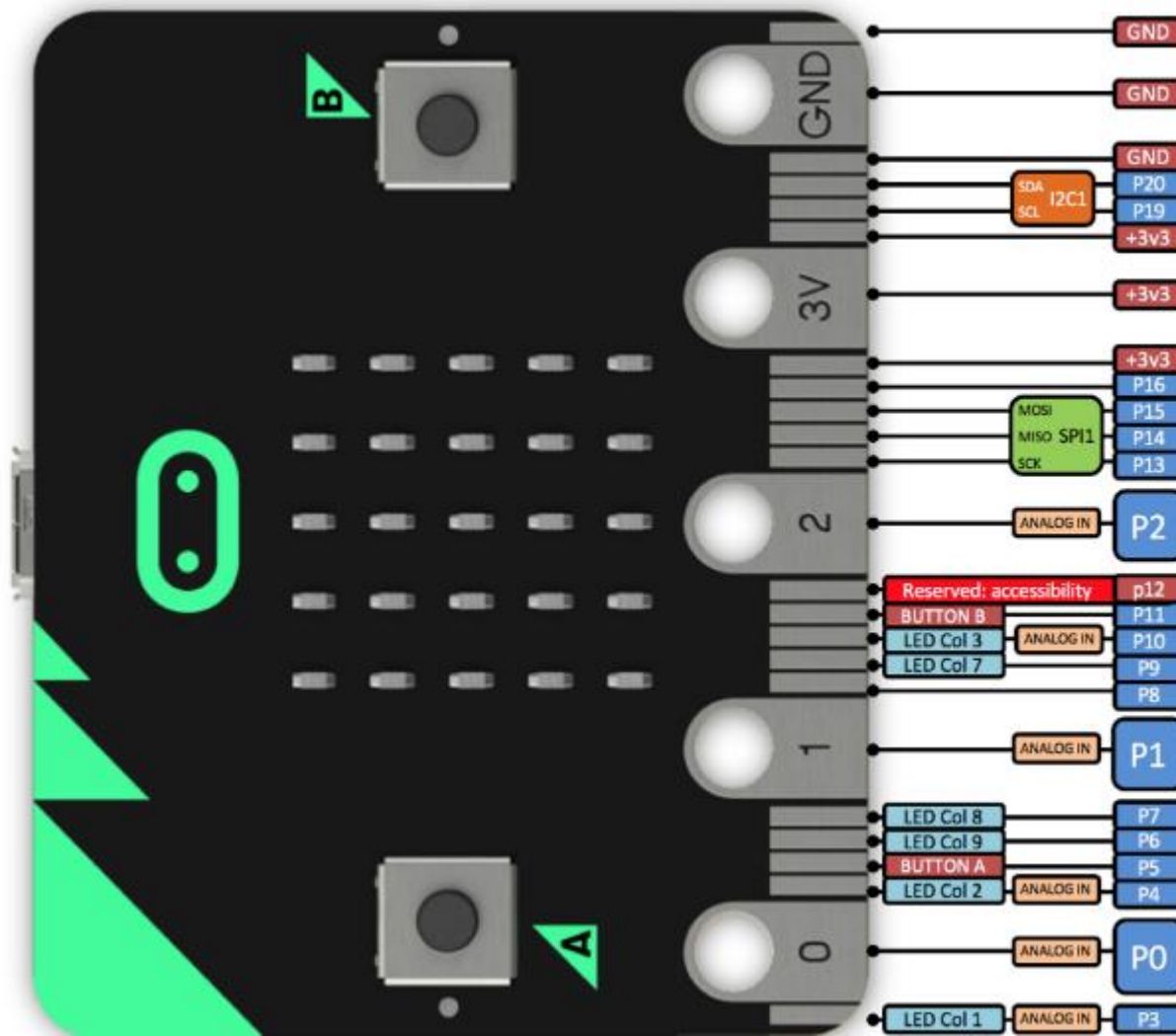
- Analog (PWM) Output in Arduino

**PWM Output pins D3, D5, D6, D9, D10, D11**



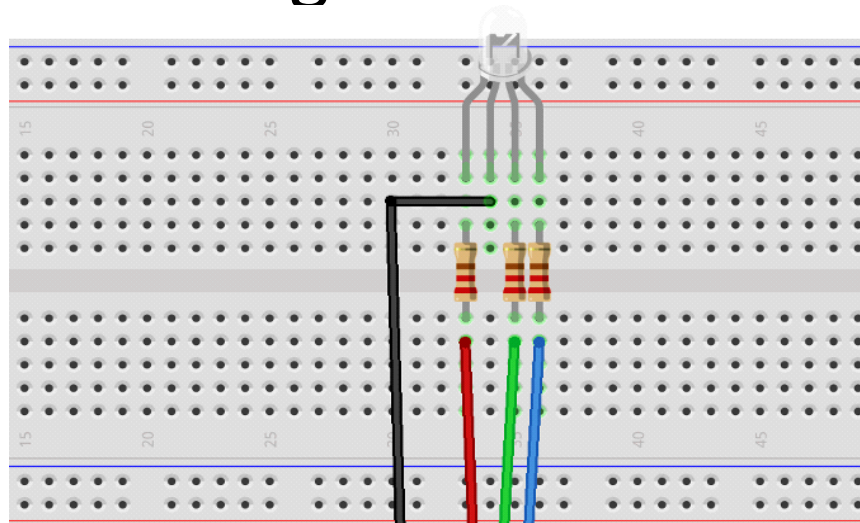
# Basic Knowledge

- Analog Input and Output in micro:bit



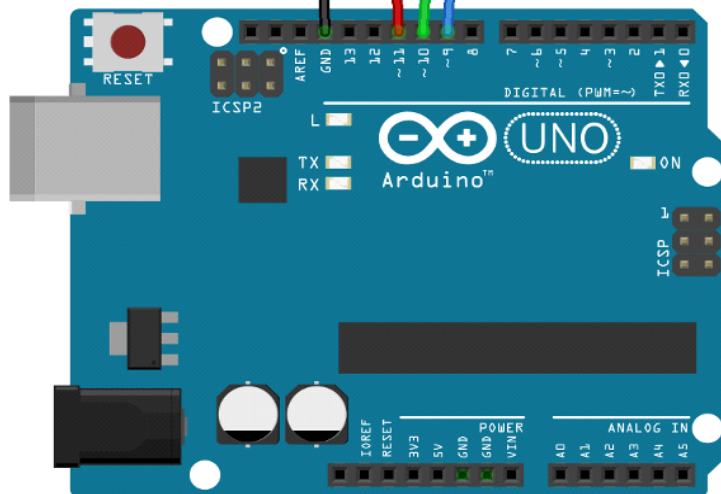
# Guided Tasks

- Controlling the Colour of an RGB LED



- 1 - RED
- 2 - GROUND
- 3 - GREEN
- 4 - BLUE

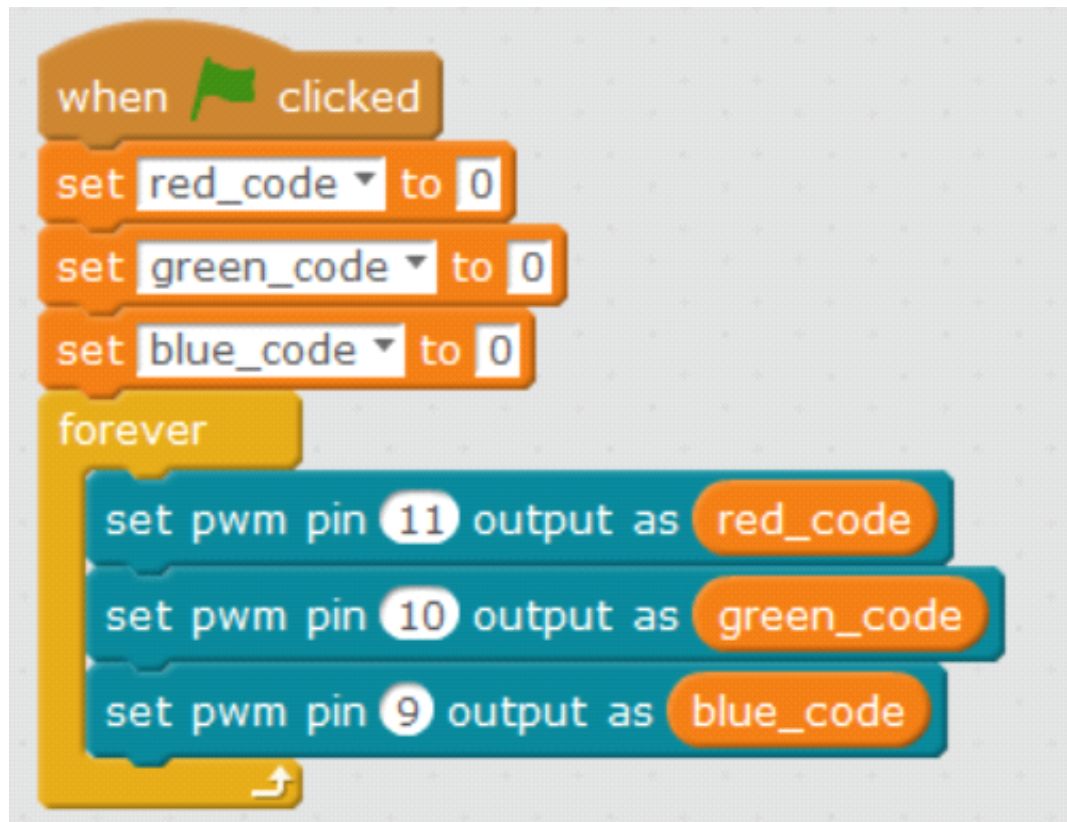
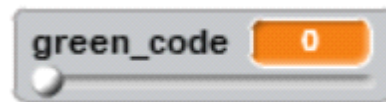
Common cathode





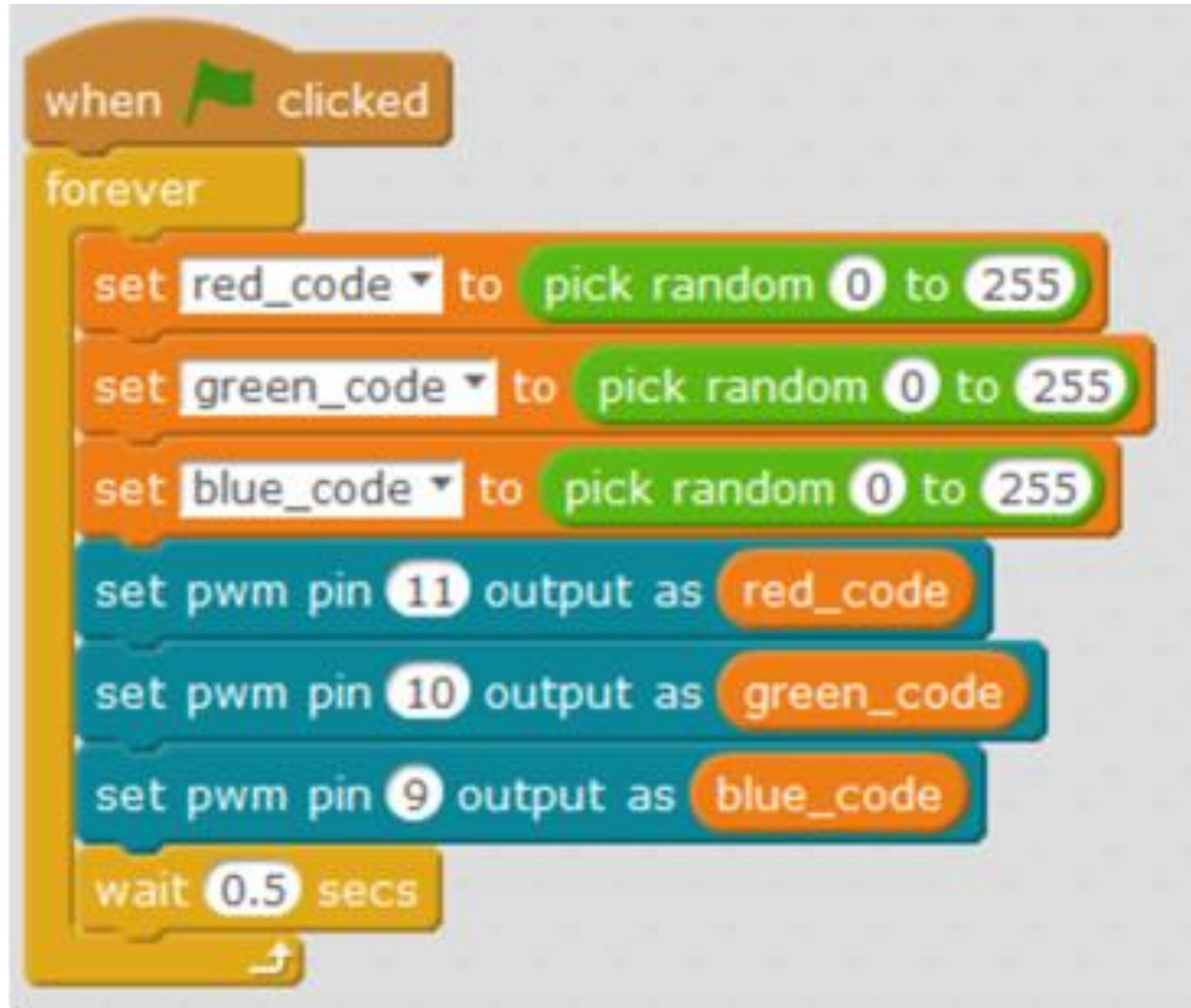
# Guided Tasks

- Controlling the Colour of an RGB LED



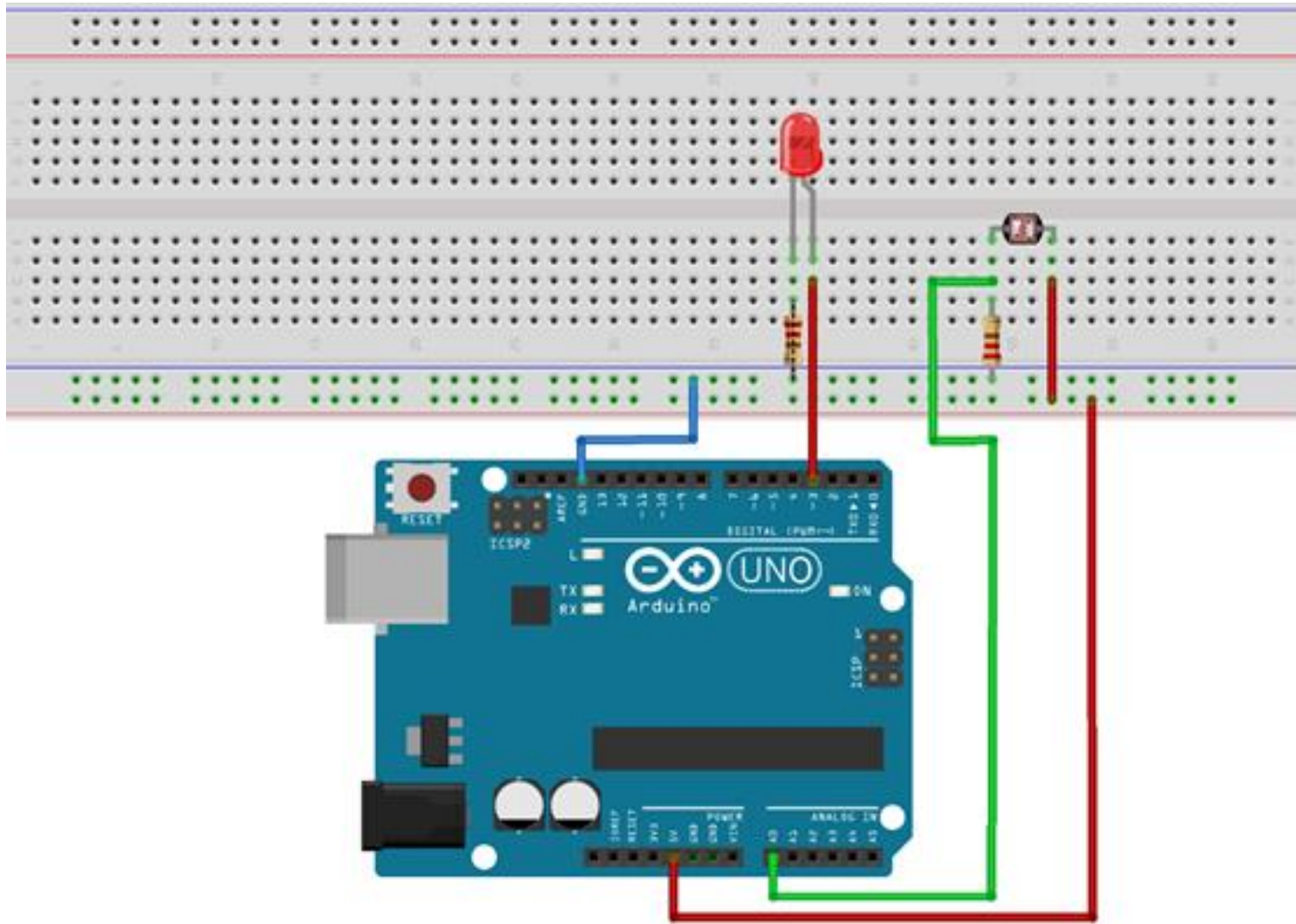
# Guided Tasks

- Controlling the Colour of an RGB LED



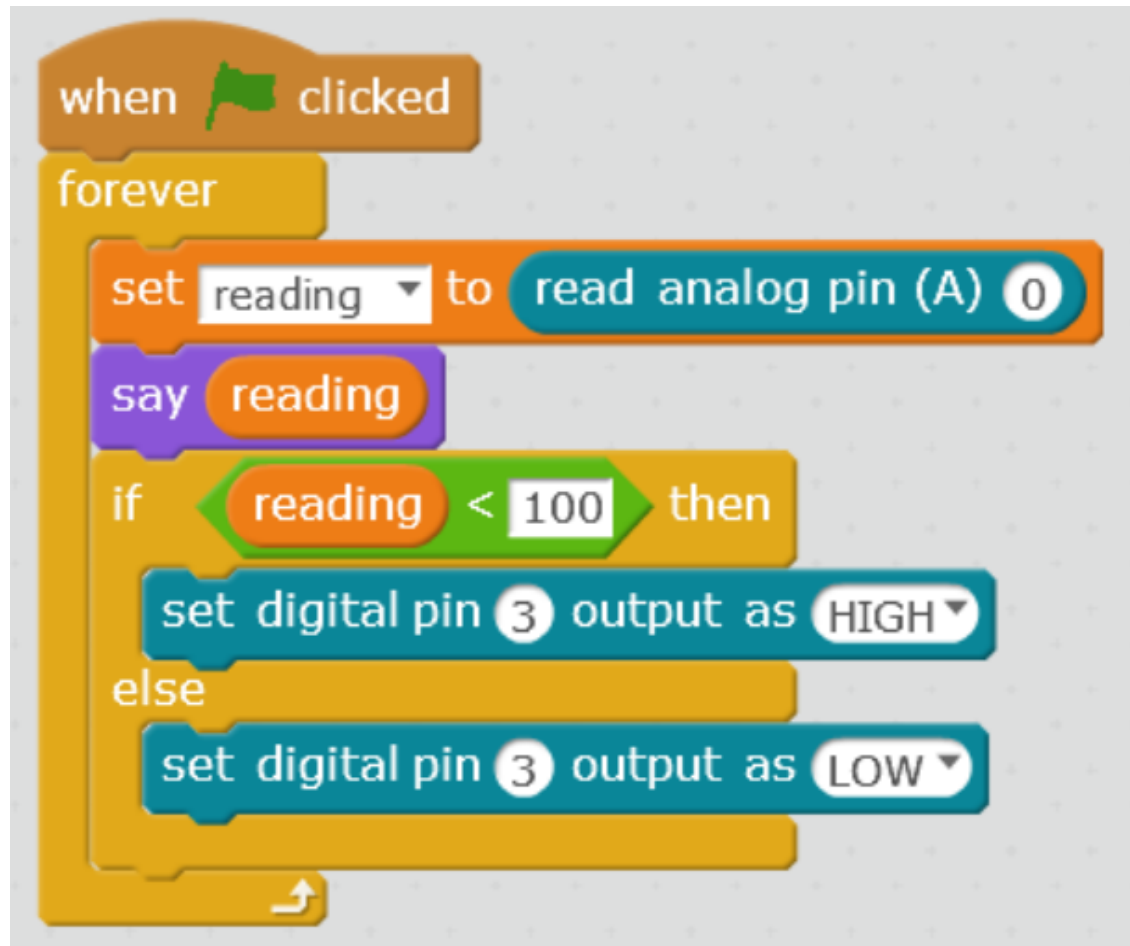
# Guided Tasks

- Automatic LED Control by Light Sensor (LDR)



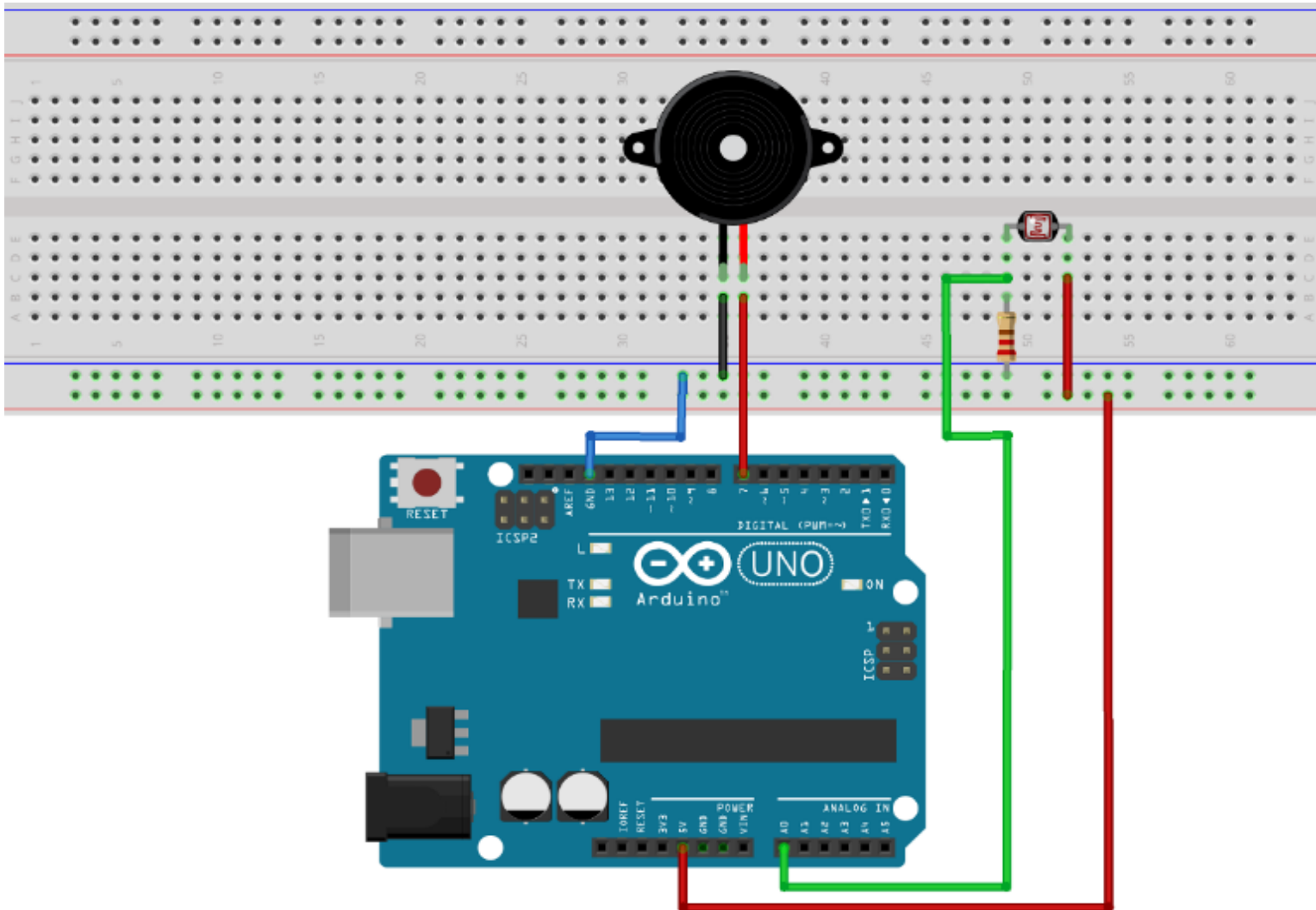
# Guided Tasks

- Automatic LED Control by Light Sensor (LDR)



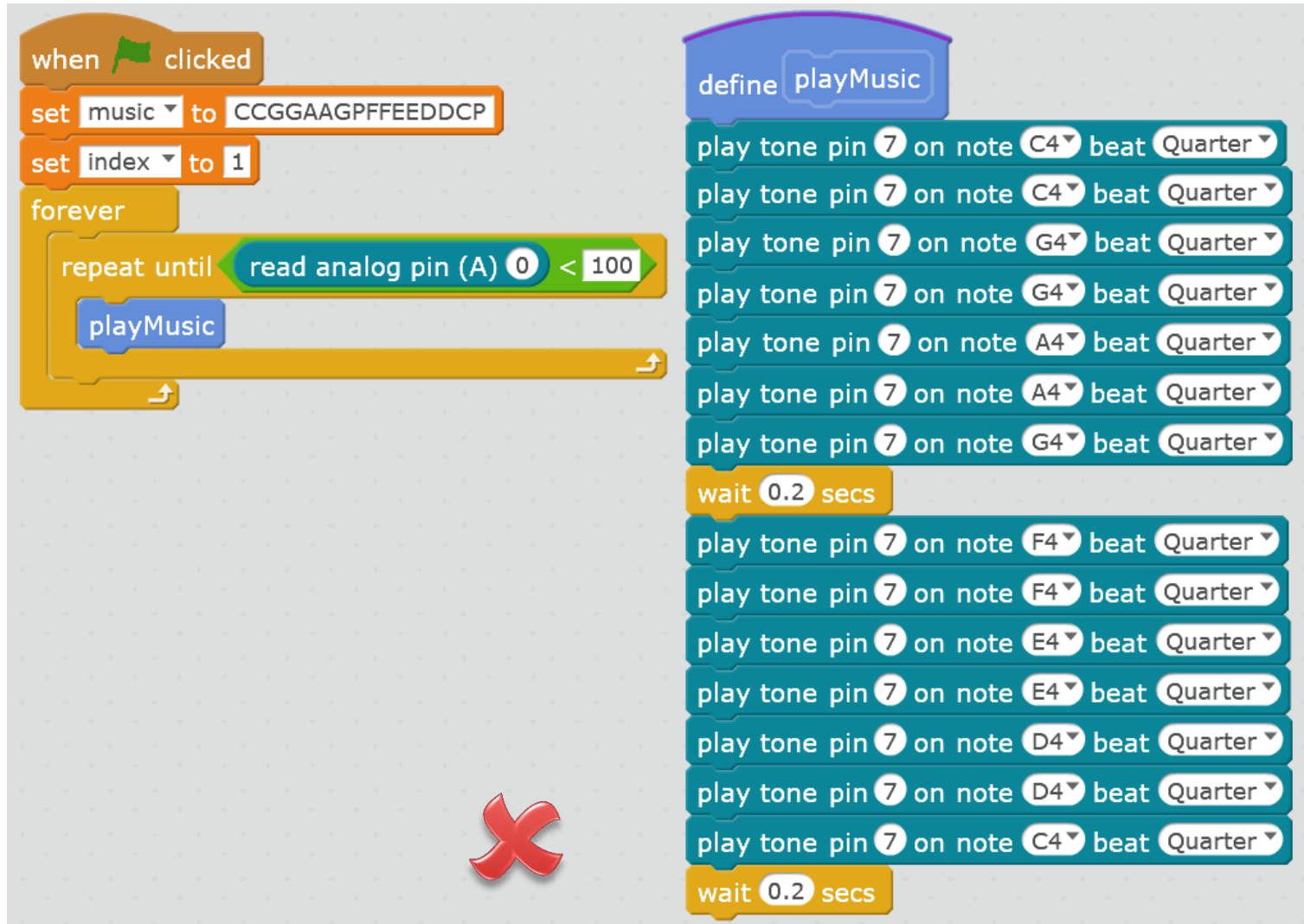
# Guided Tasks

- Control of Music Playing by Light Sensor (LDR)

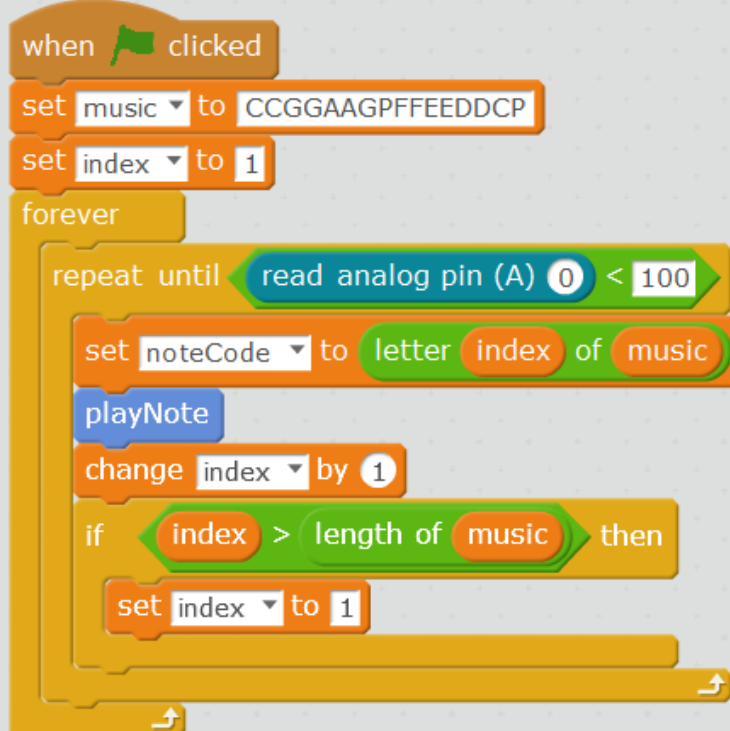


# Guided Tasks

- Control of Music Playing by Light Sensor (LDR)









## Arduino Program

set index to 1

forever

repeat until read analog pin (A) 0 < 100

set noteCode to letter index of CCGGAAGPFFEEDDCP

playNote

change index by 1

if index > length of CCGGAAGPFFEEDDCP then

set index to 1

define playNote

if noteCode = 'A' then

play tone pin 7 on note A4 beat Quarter

if noteCode = 'B' then

play tone pin 7 on note B4 beat Quarter

if noteCode = 'C' then

play tone pin 7 on note C4 beat Quarter

if noteCode = 'D' then

play tone pin 7 on note D4 beat Quarter

if noteCode = 'E' then

play tone pin 7 on note E4 beat Quarter

if noteCode = 'F' then

play tone pin 7 on note F4 beat Quarter

if noteCode = 'G' then

play tone pin 7 on note G4 beat Quarter

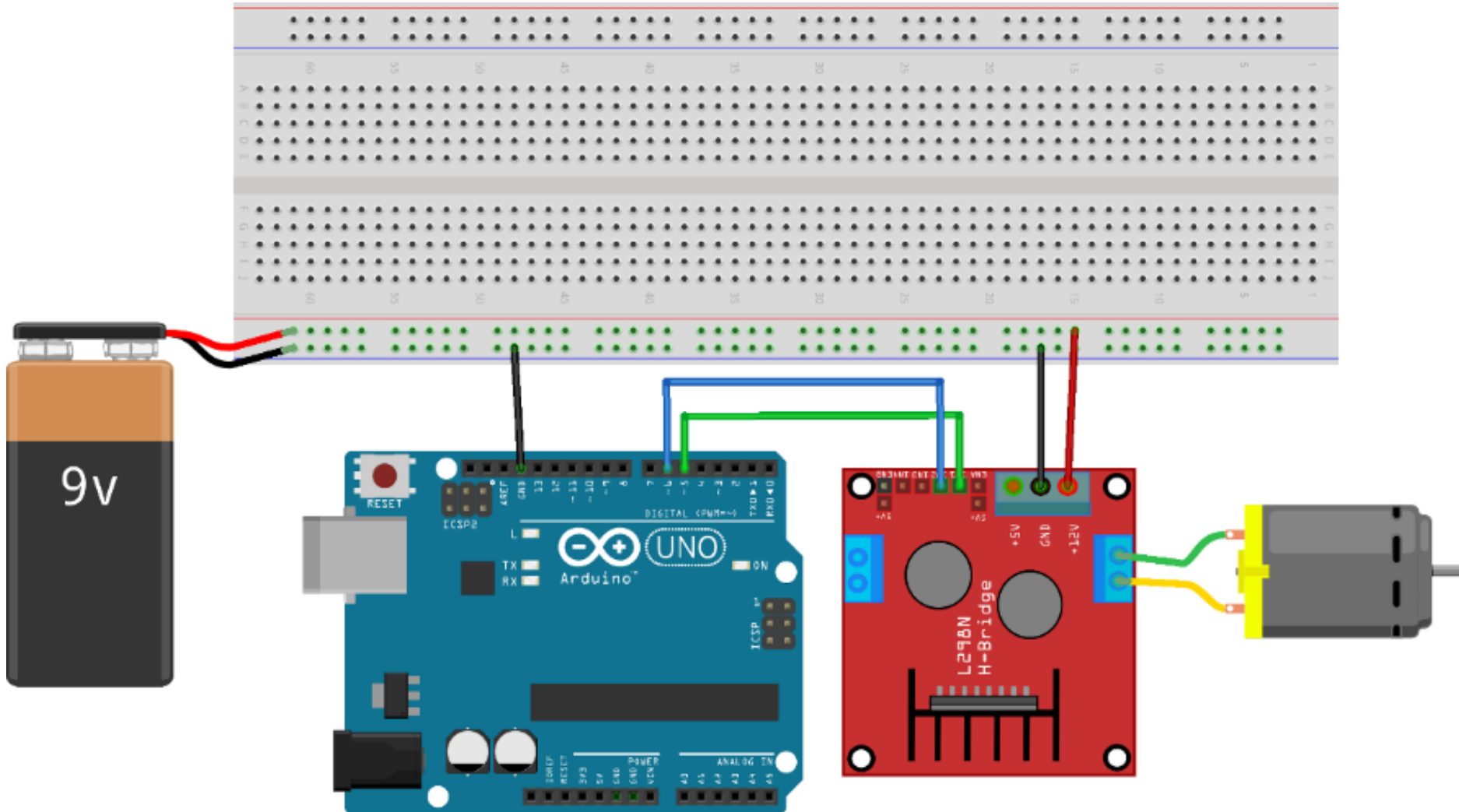
if noteCode = 'P' then

wait 0.2 secs



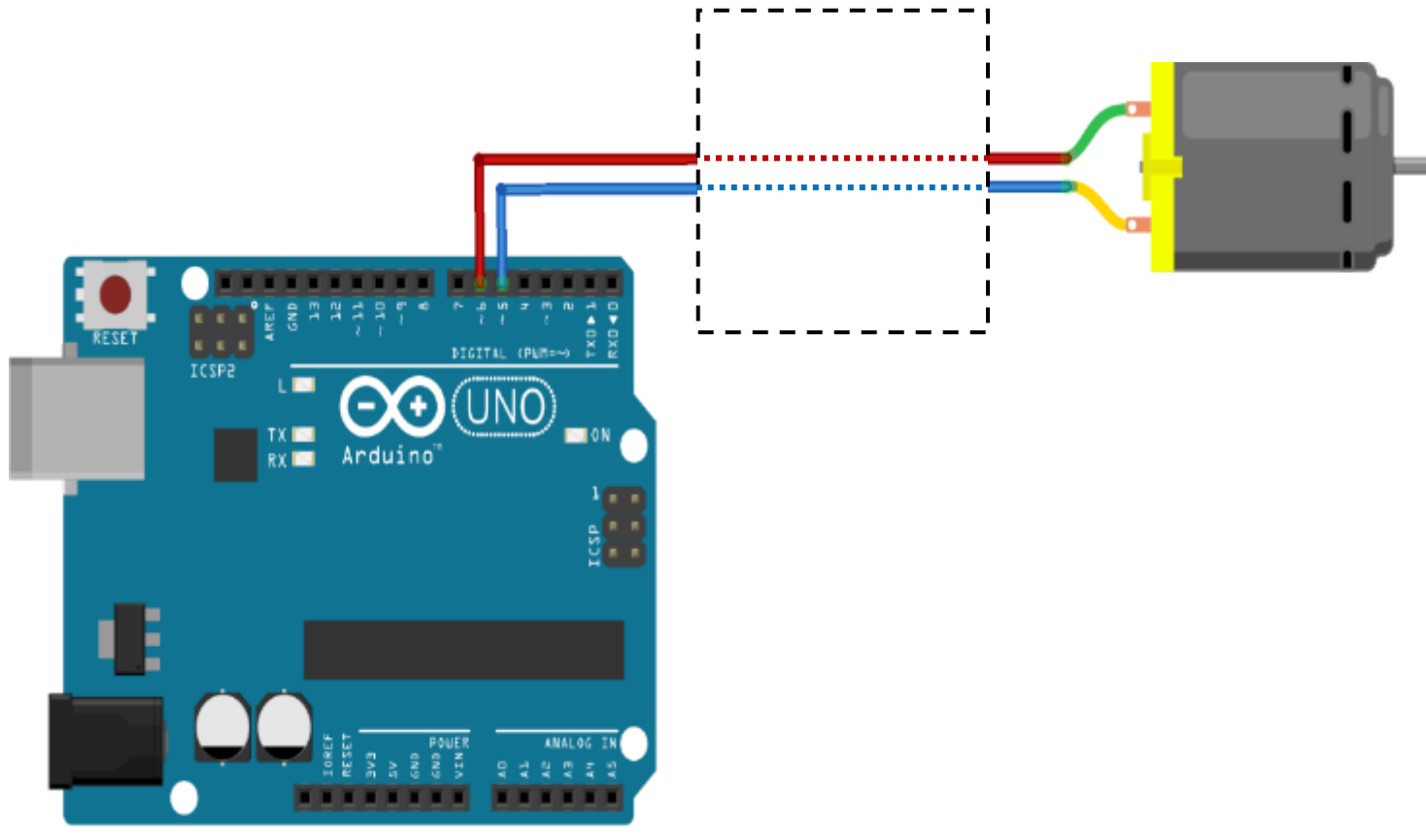
# Guided Tasks

- Controlling DC Motor



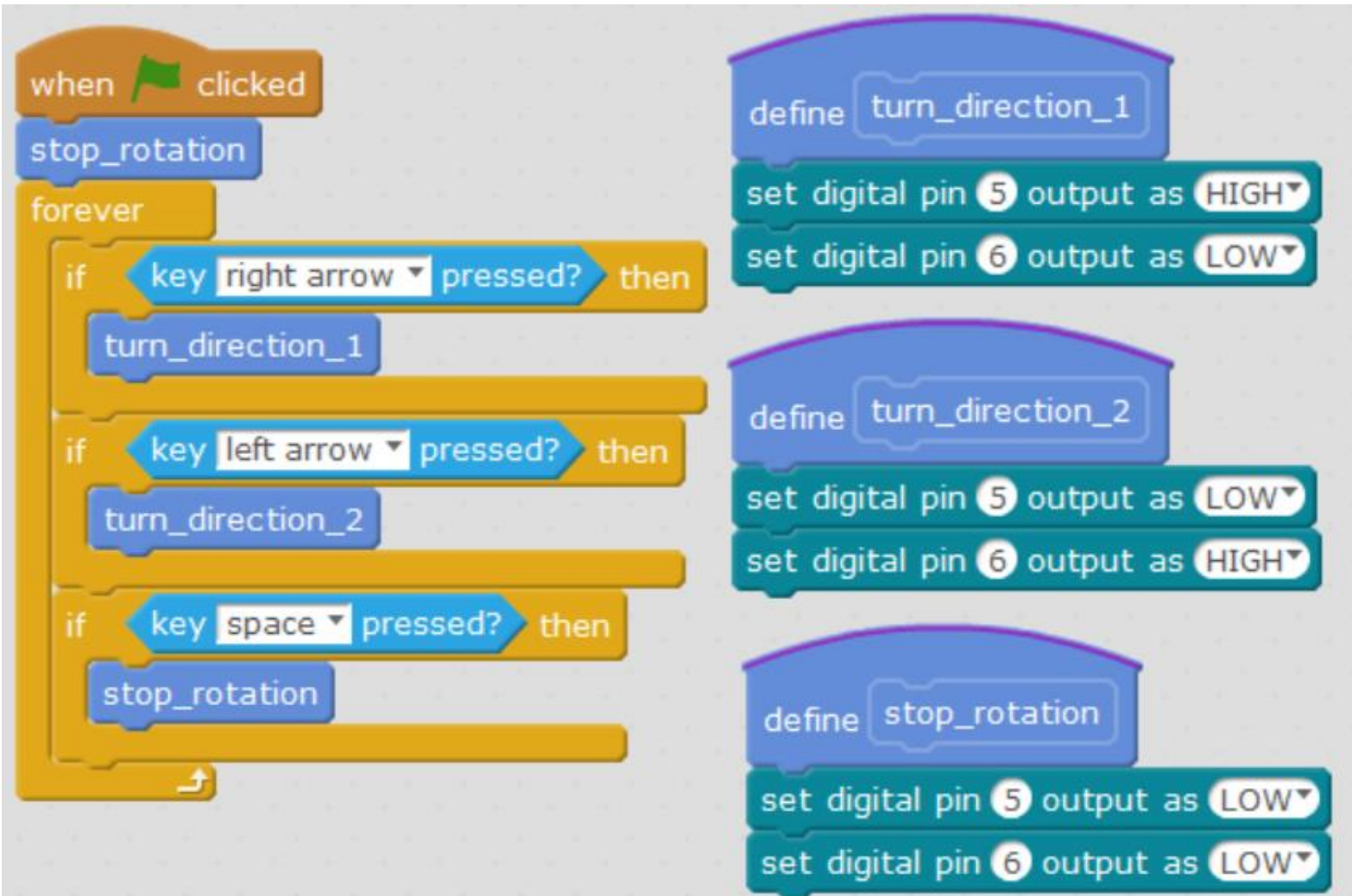
# Guided Tasks

- Controlling DC Motor



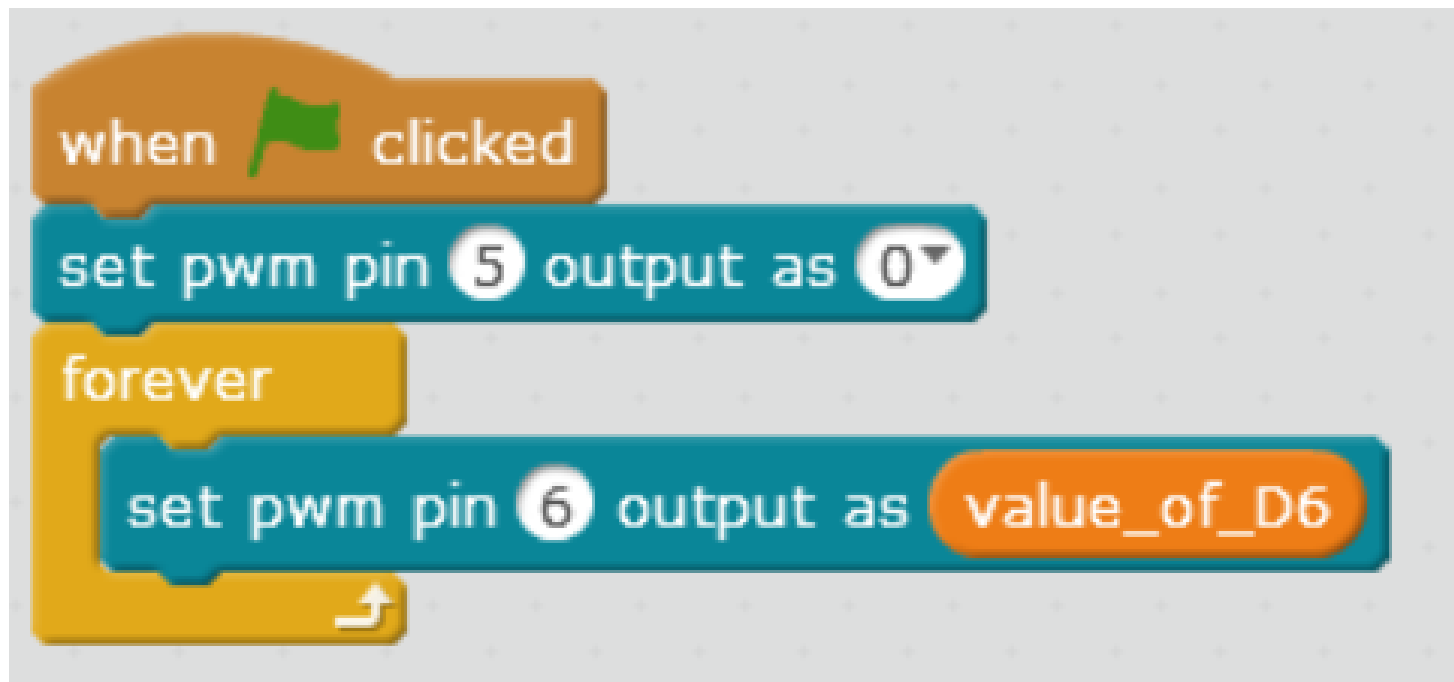
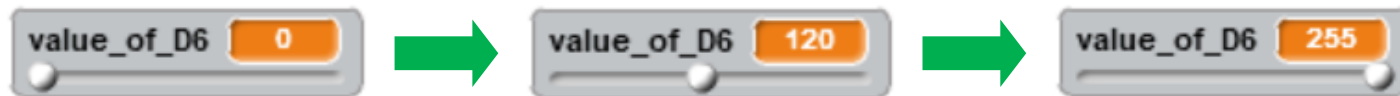
# Guided Tasks

- Controlling Direction of Rotation



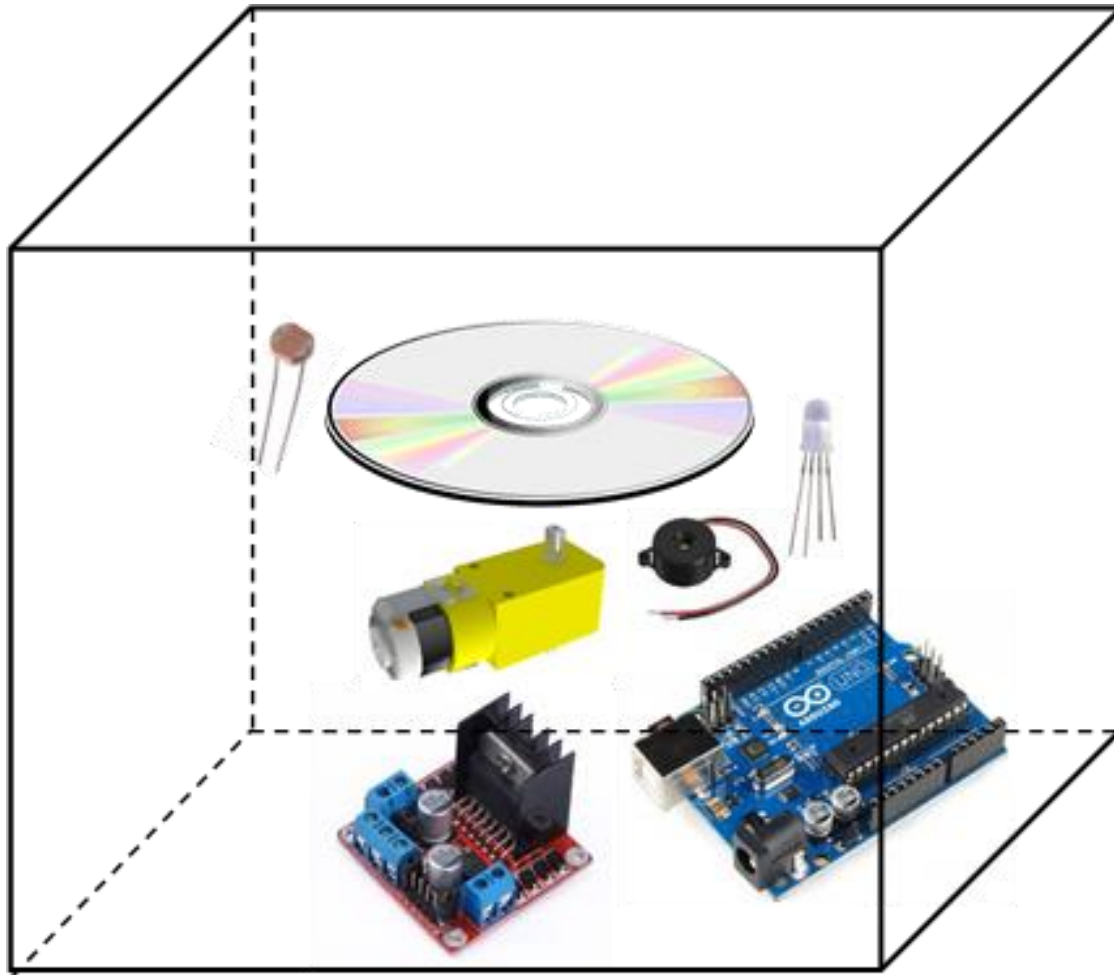
# Guided Tasks

- Controlling Speed of Rotation



# Self-Exploration Project

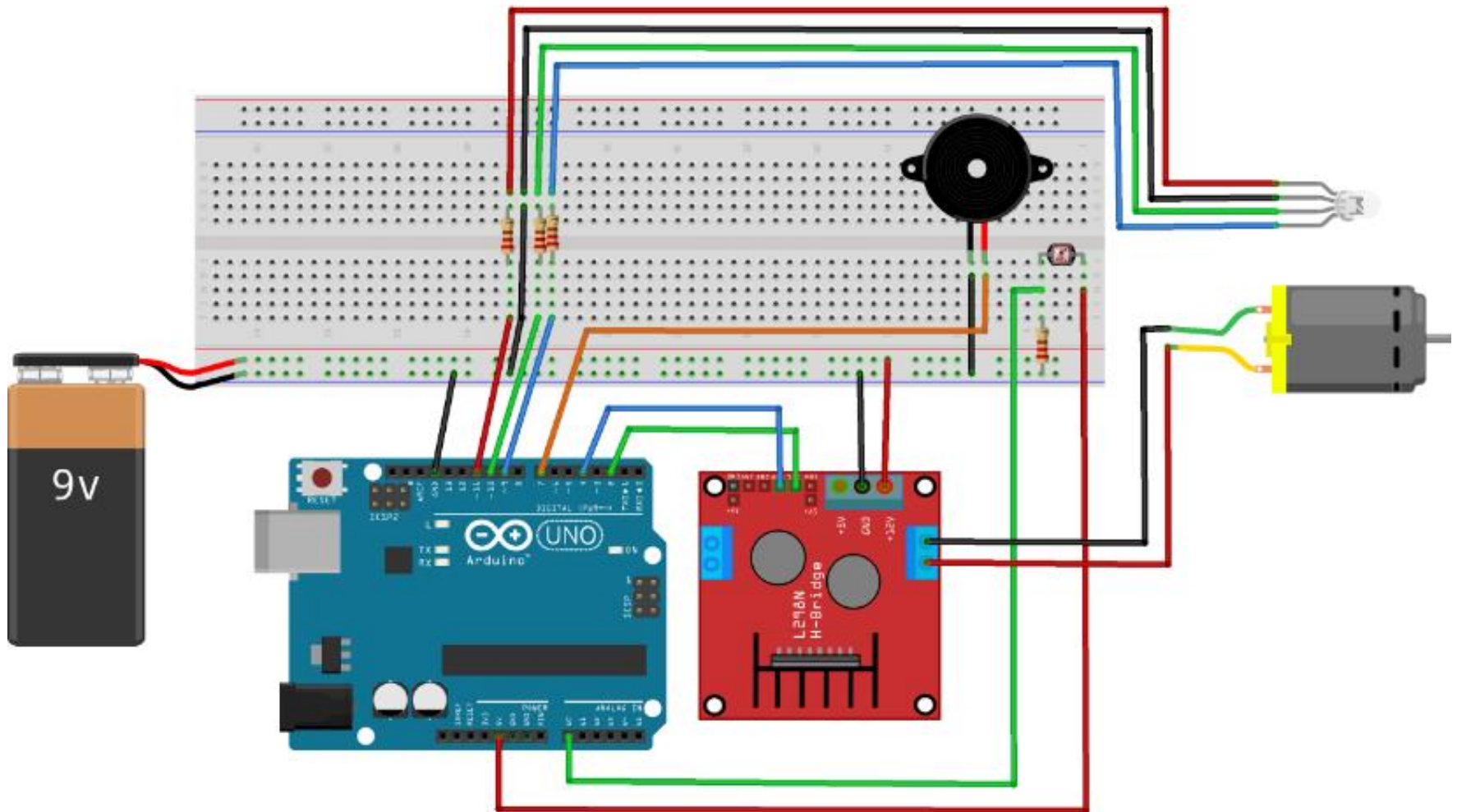
- Design a light-control music box.



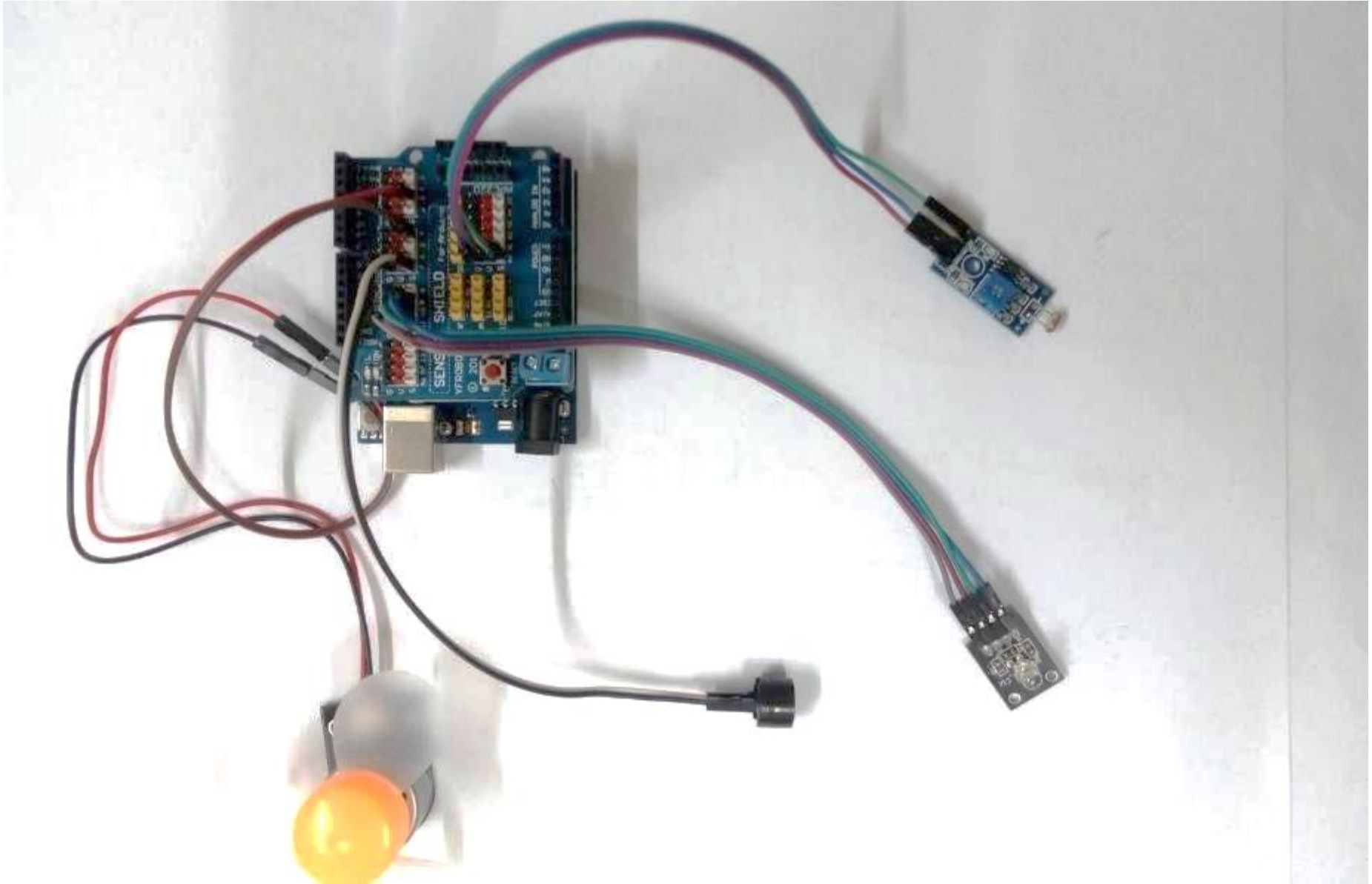


# Self-Exploration Project

- Design a light-control music box.

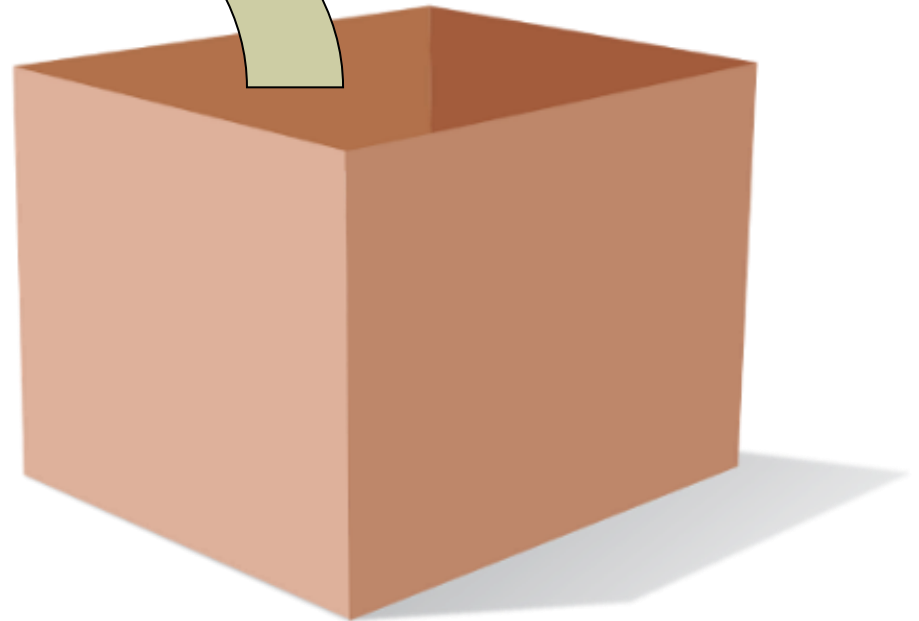
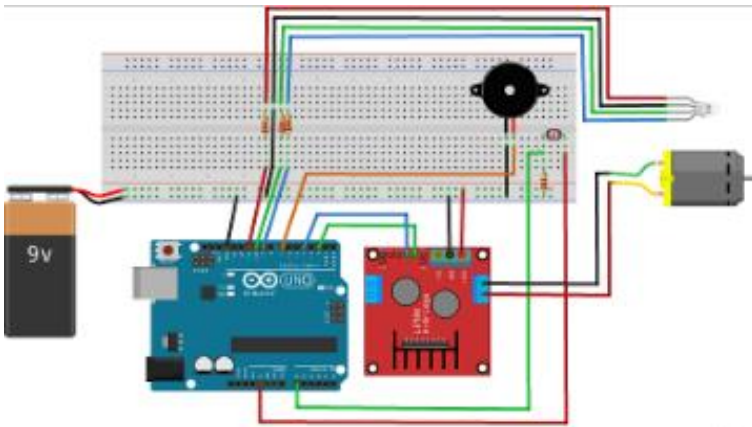
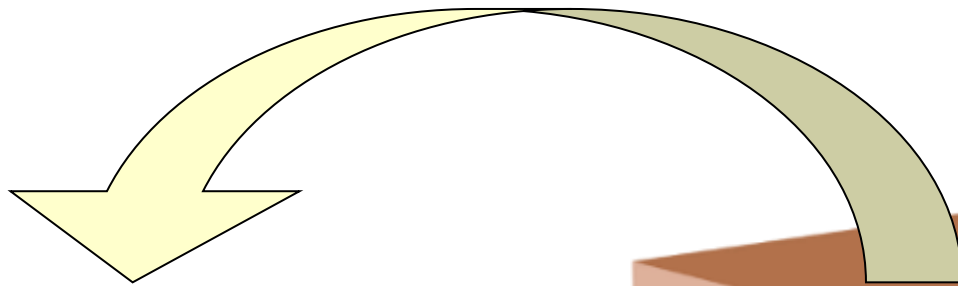


# Self-Exploration Project



# Self-Exploration Project

- Design a light-control music box.



# Self-Exploration Project

- Design a light-control music box.

Start

.

Forever

Repeat until light is dim

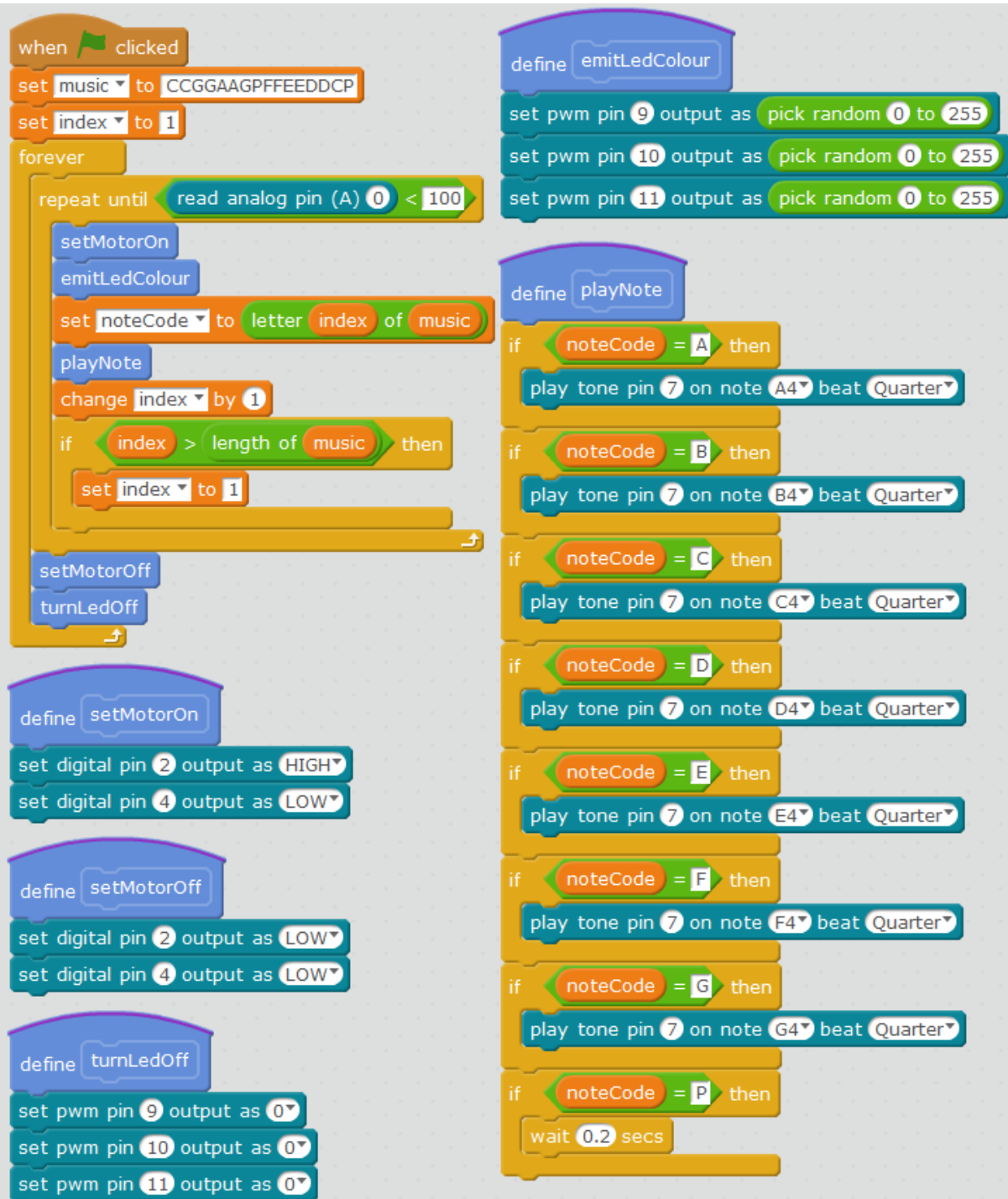
Set motor on

LED emits random colour

Play a musical note of the music

Set motor off

Turn off LED



```

Arduino Program
set index to 1
forever
  repeat until read analog pin (A) 0 < 100
    setMotorOn
    emitLedColour
    set noteCode to letter index of CCGGAAGPFFEEDDCP
    playNote
    change index by 1
    if index > length of CCGGAAGPFFEEDDCP then
      set index to 1
  setMotorOff
  turnLedOff

```

```

define setMotorOn
  set digital pin 2 output as HIGH
  set digital pin 4 output as LOW

```

```

define setMotorOff
  set digital pin 2 output as LOW
  set digital pin 4 output as LOW

```

```

define turnLedOff
  set pwm pin 9 output as 0
  set pwm pin 10 output as 0
  set pwm pin 11 output as 0

```

```

define emitLedColour
  set pwm pin 9 output as pick random 0 to 255
  set pwm pin 10 output as pick random 0 to 255
  set pwm pin 11 output as pick random 0 to 255

```

```

define playNote
  if noteCode = 'A' then
    play tone pin 7 on note A4 beat Quarter
  if noteCode = 'B' then
    play tone pin 7 on note B4 beat Quarter
  if noteCode = 'C' then
    play tone pin 7 on note C4 beat Quarter
  if noteCode = 'D' then
    play tone pin 7 on note D4 beat Quarter
  if noteCode = 'E' then
    play tone pin 7 on note E4 beat Quarter
  if noteCode = 'F' then
    play tone pin 7 on note F4 beat Quarter
  if noteCode = 'G' then
    play tone pin 7 on note G4 beat Quarter
  if noteCode = 'P' then
    wait 0.2 secs

```



# Project 4

A Smart Fan

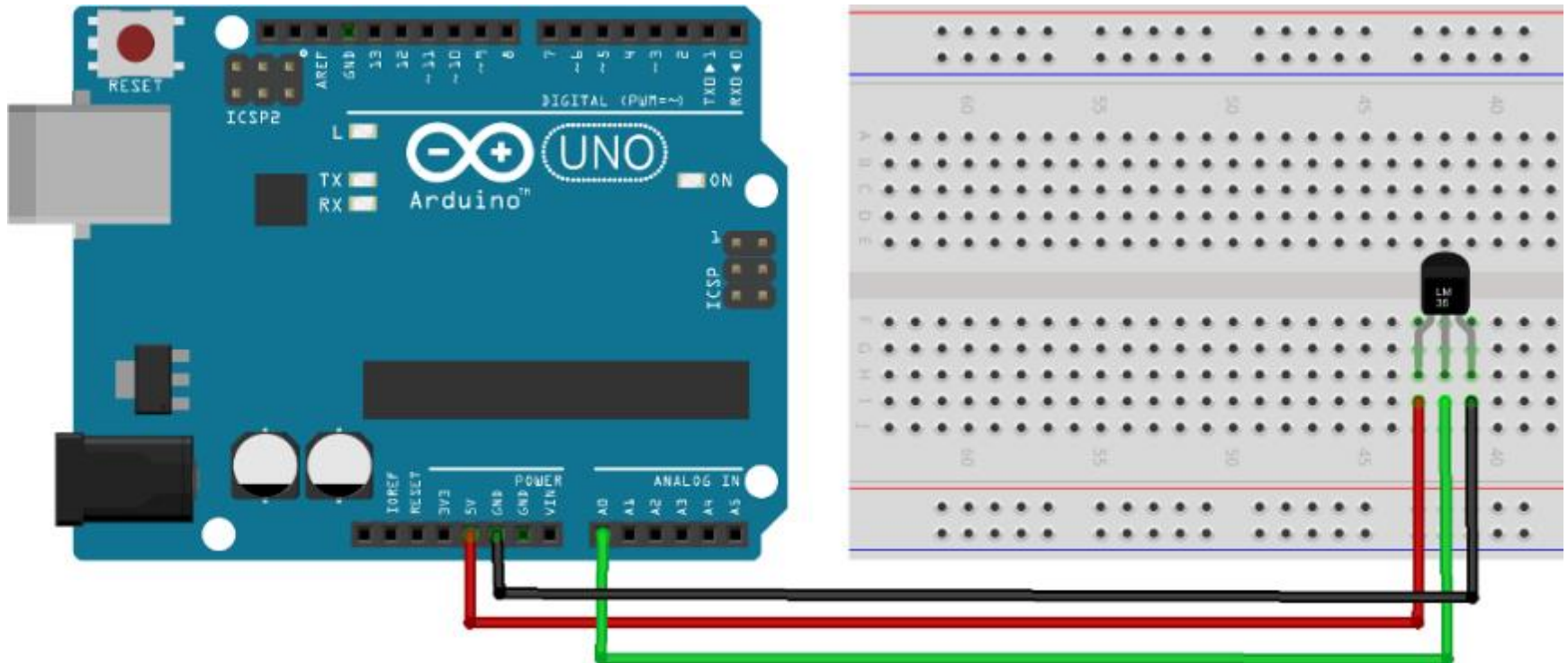
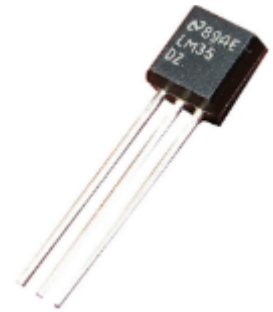
# Project Requirements

- Design a smart fan



# Guided Tasks

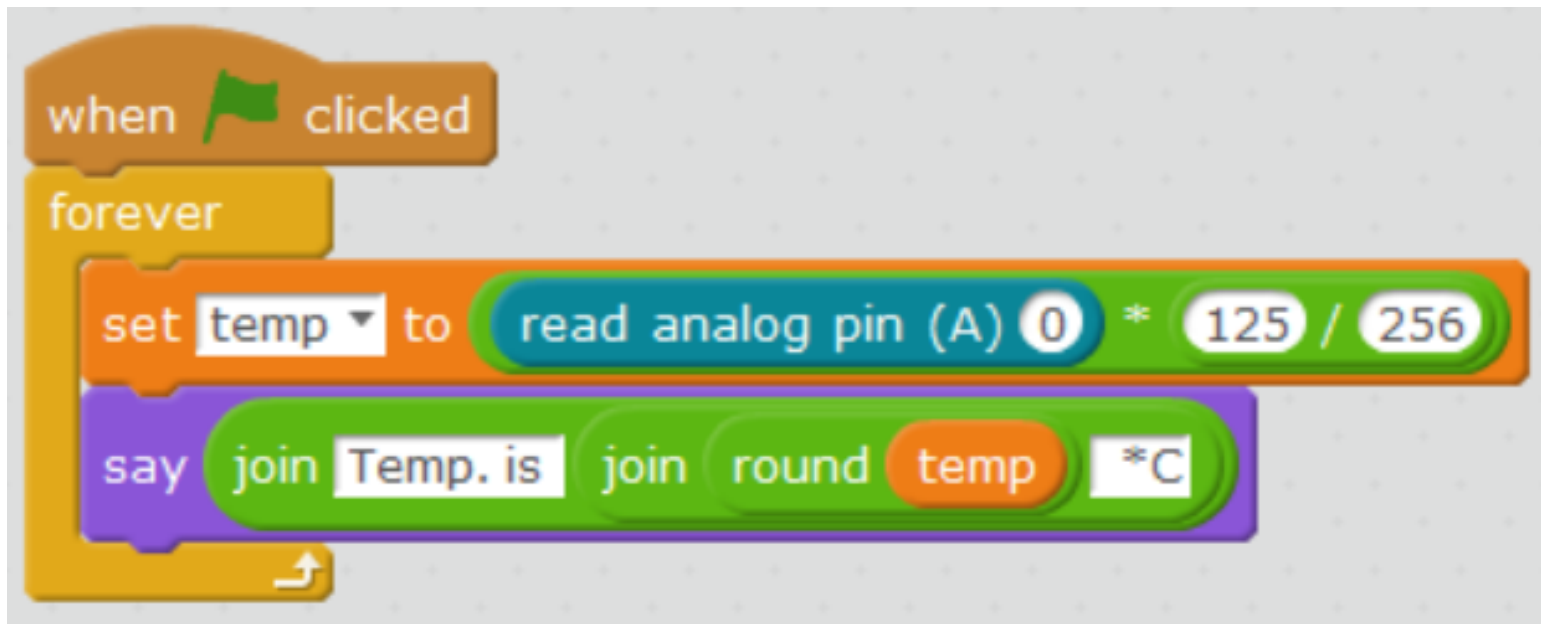
- Measuring Temperature by LM35



# Guided Tasks

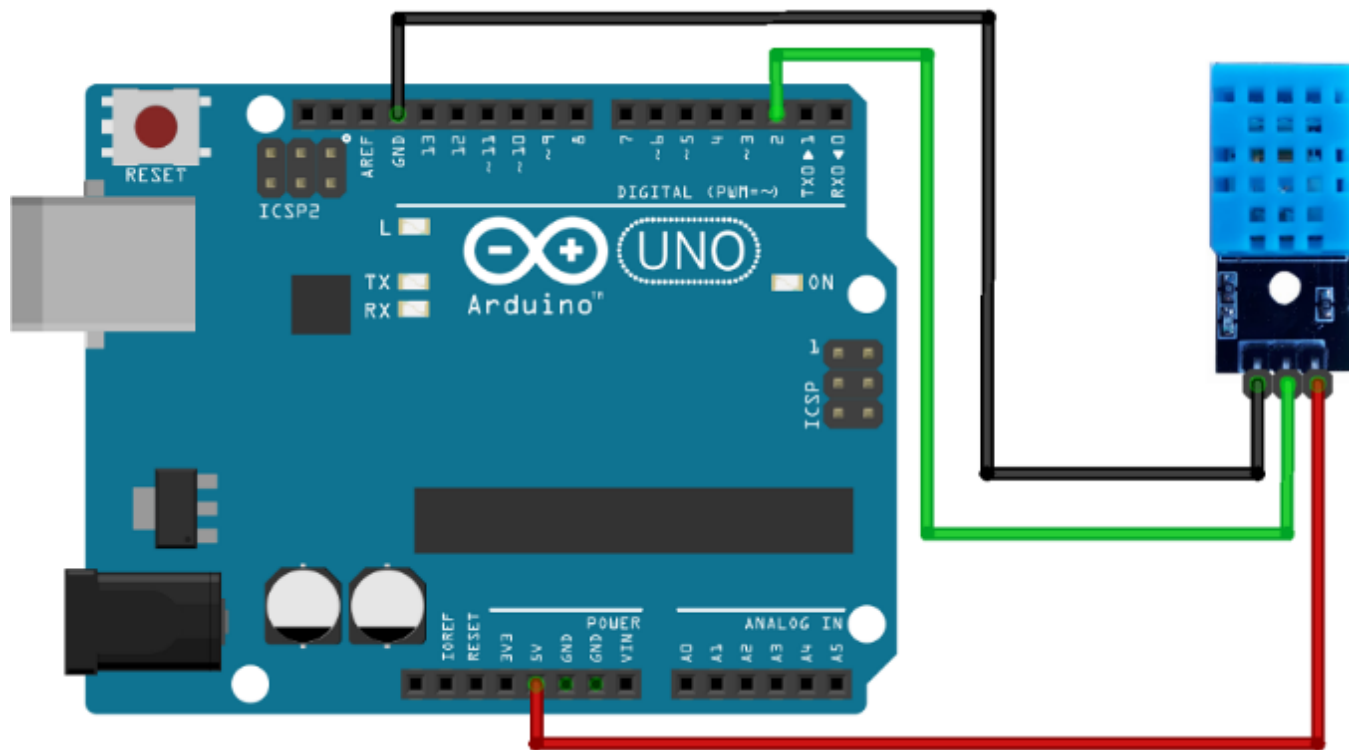
- Measuring Temperature by LM35

$$T = \frac{\text{A0 value} \times 5000}{1024 \times 10} = \text{A0 value} \times \frac{125}{256}$$



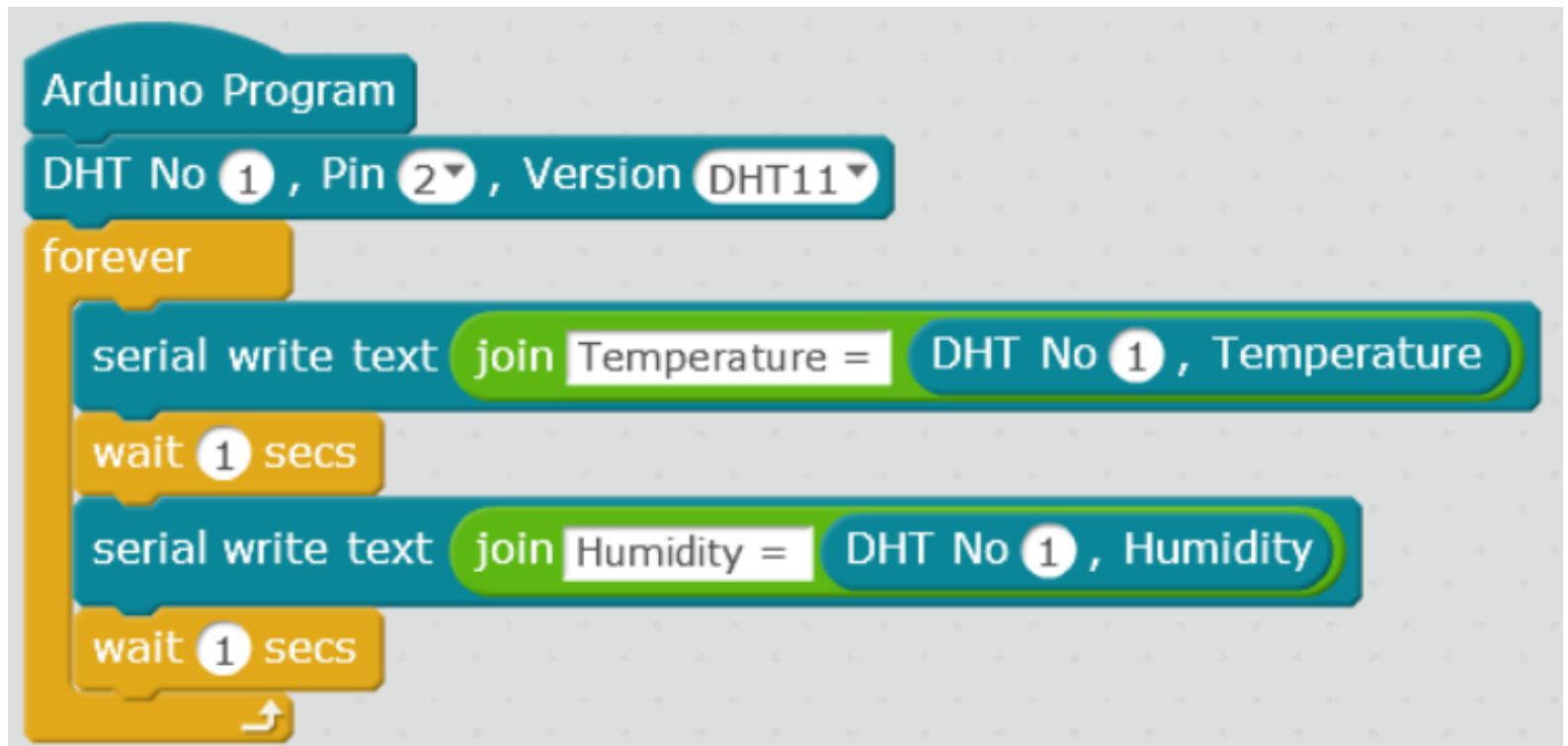
# Guided Tasks

- Measuring Temperature and Humidity by DHT11



# Guided Tasks

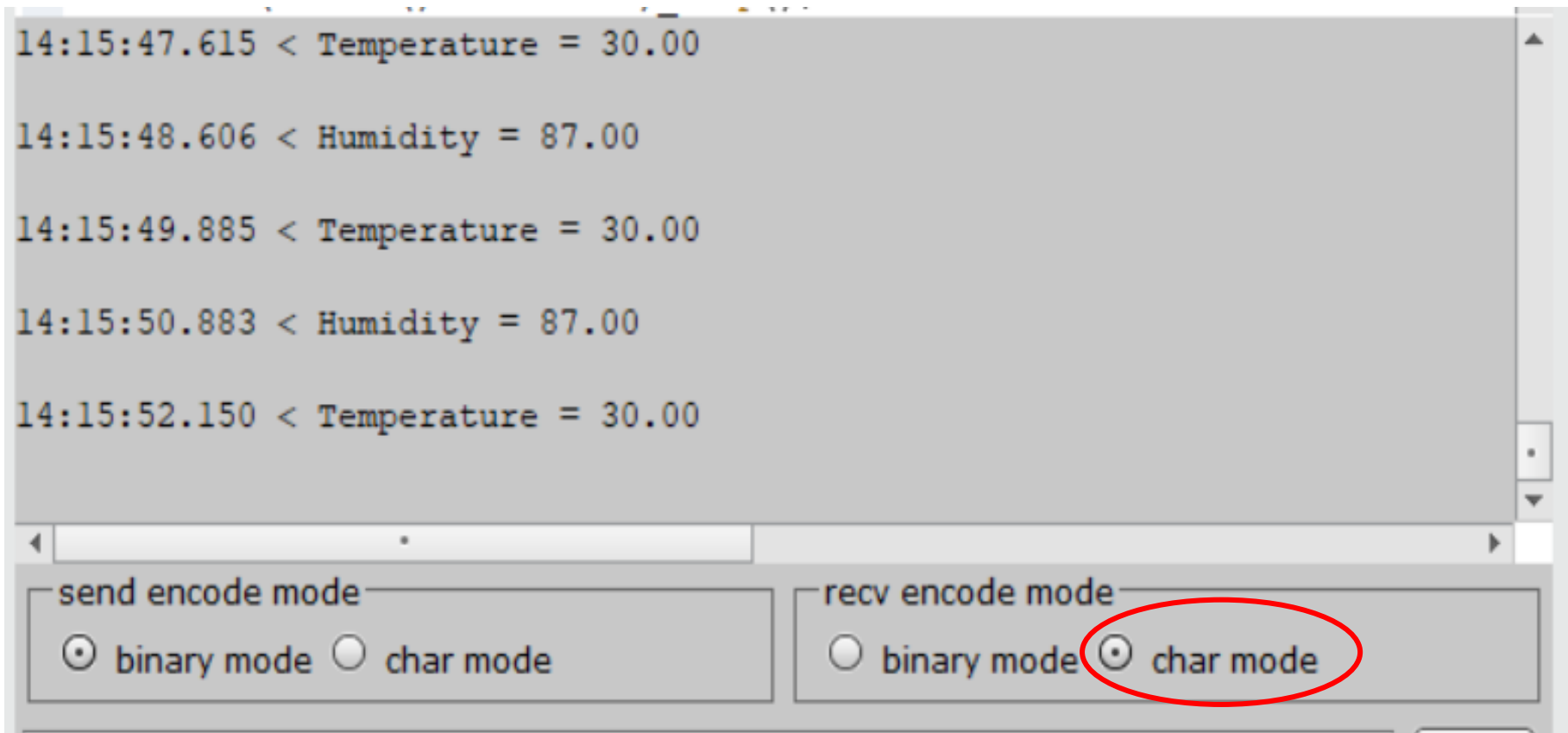
- Measuring Temperature and Humidity by DHT11





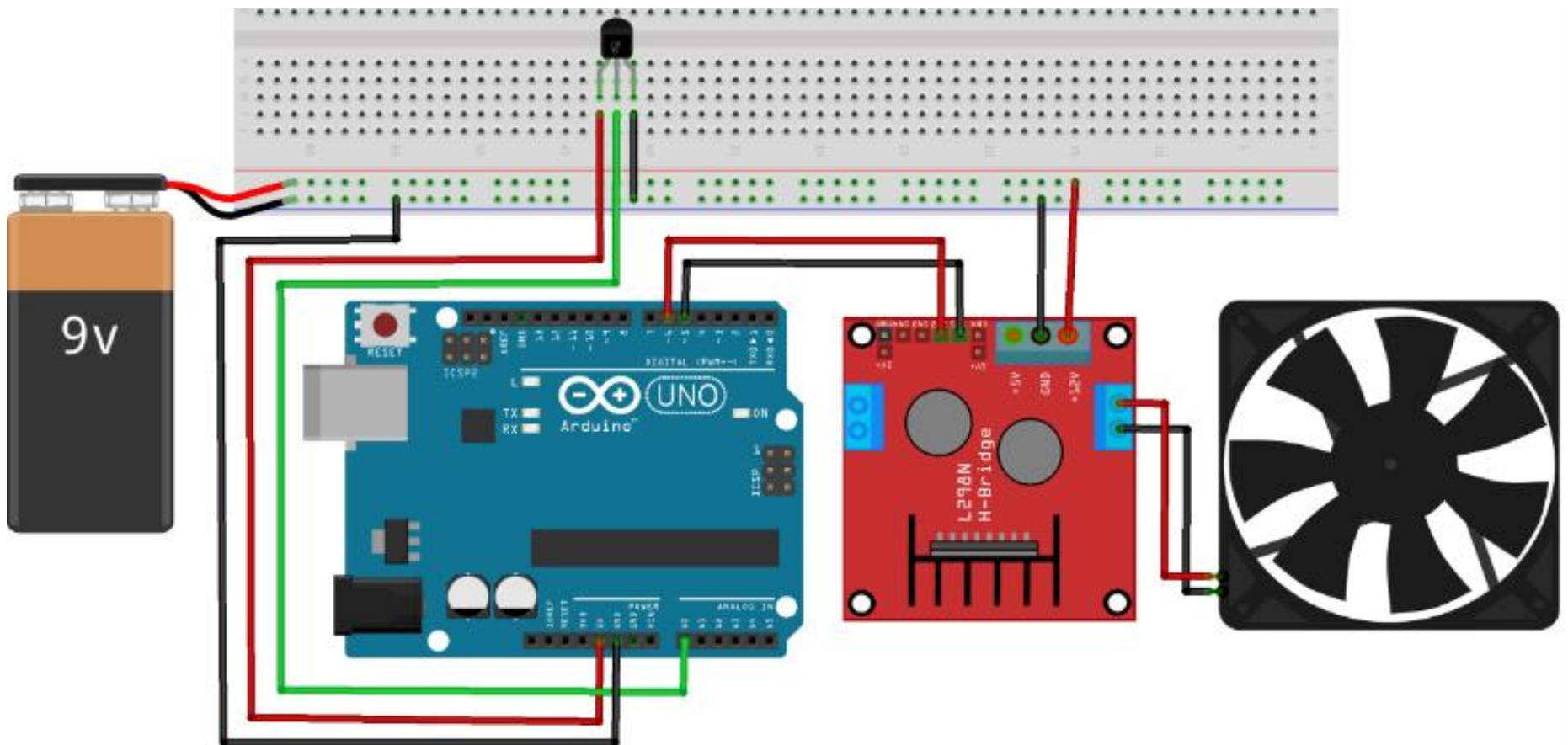
# Guided Tasks

- Measuring Temperature and Humidity by DHT11



# Guided Tasks

- Controlling a Fan by using Temperature Sensor



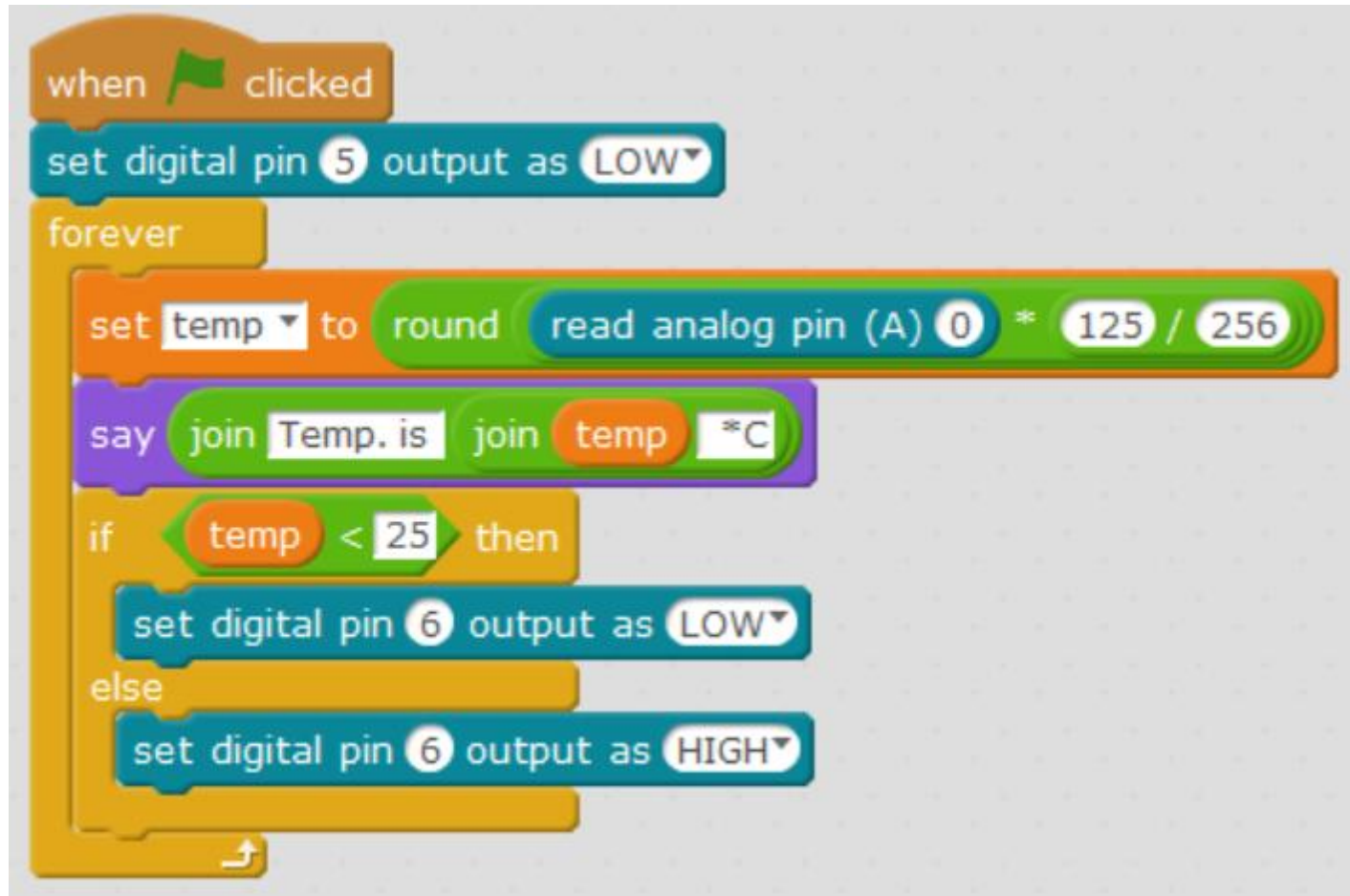
# Guided Tasks

- Controlling a Fan by using Temperature Sensor
  - Other choices



# Guided Tasks

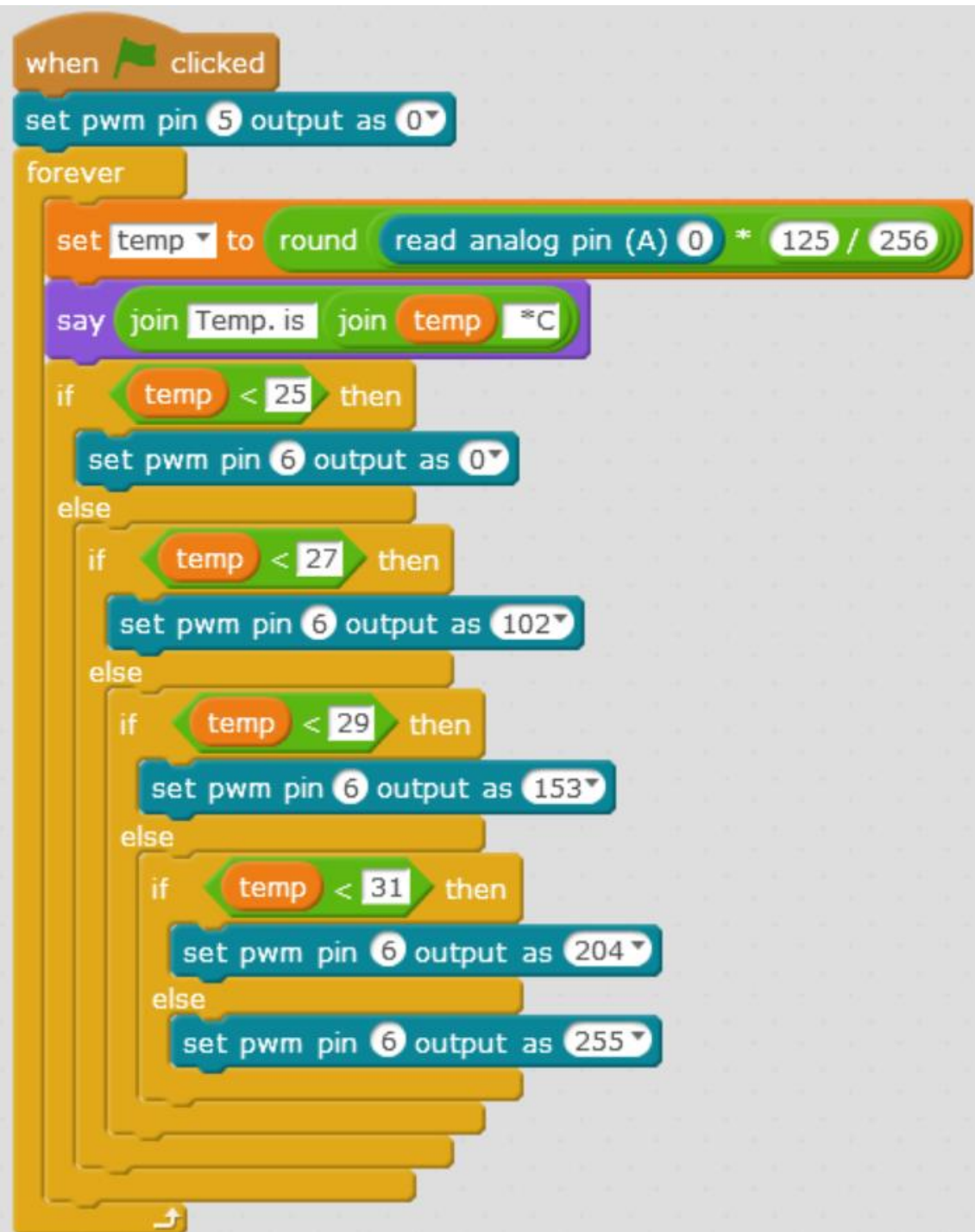
- Controlling the On / Off of a Fan



# Guided Tasks

- Controlling the Speed of a Fan

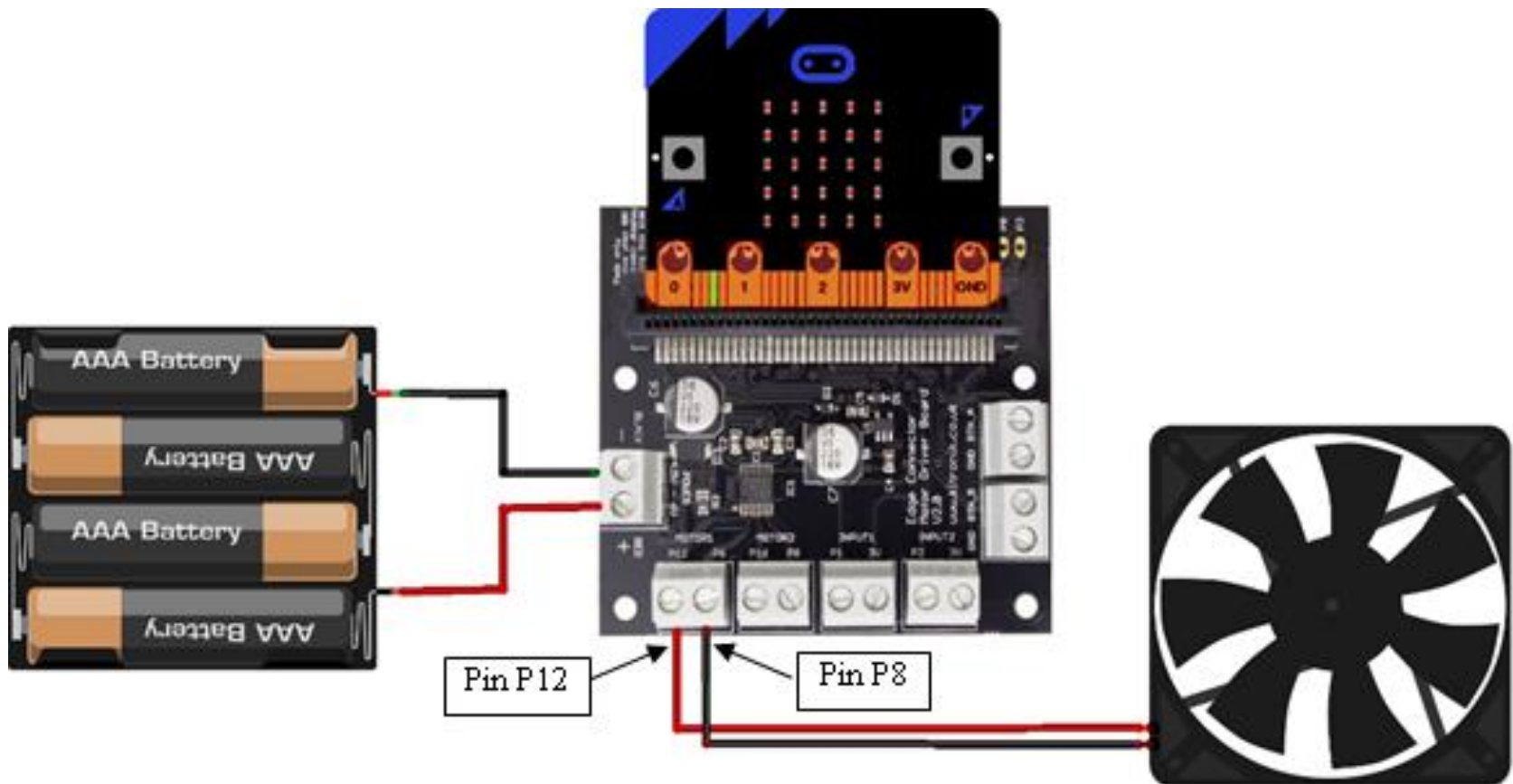
Temperature (°C)	Speed (%)	PWM (D6)
< 25	0	0
25 - 26	40%	$255 \times 0.4 = 102$
27 - 28	60%	$255 \times 0.6 = 153$
29 - 30	80%	$255 \times 0.8 = 204$
> 30	100%	255





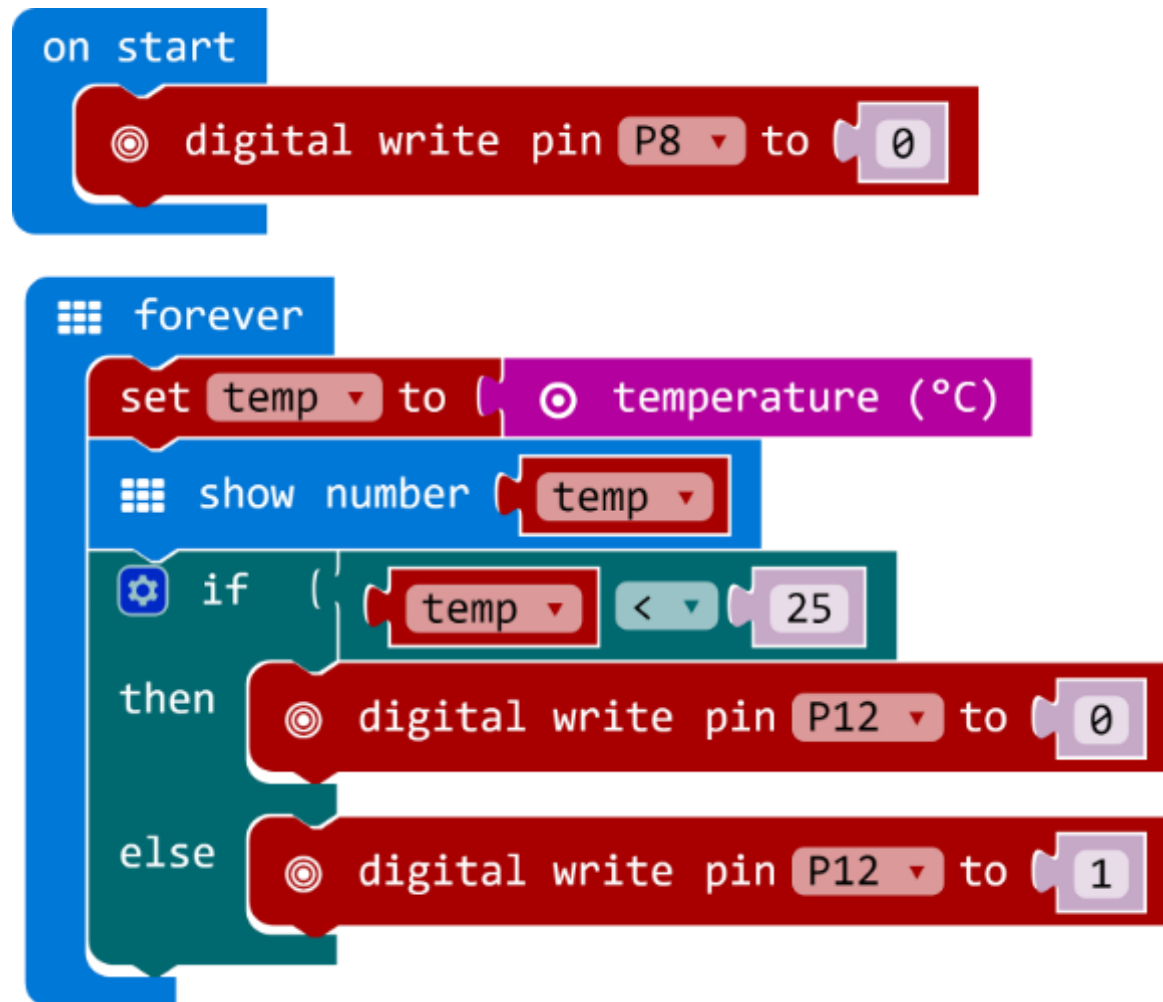
# Guided Tasks

- Controlling a Fan by using Temperature Sensor



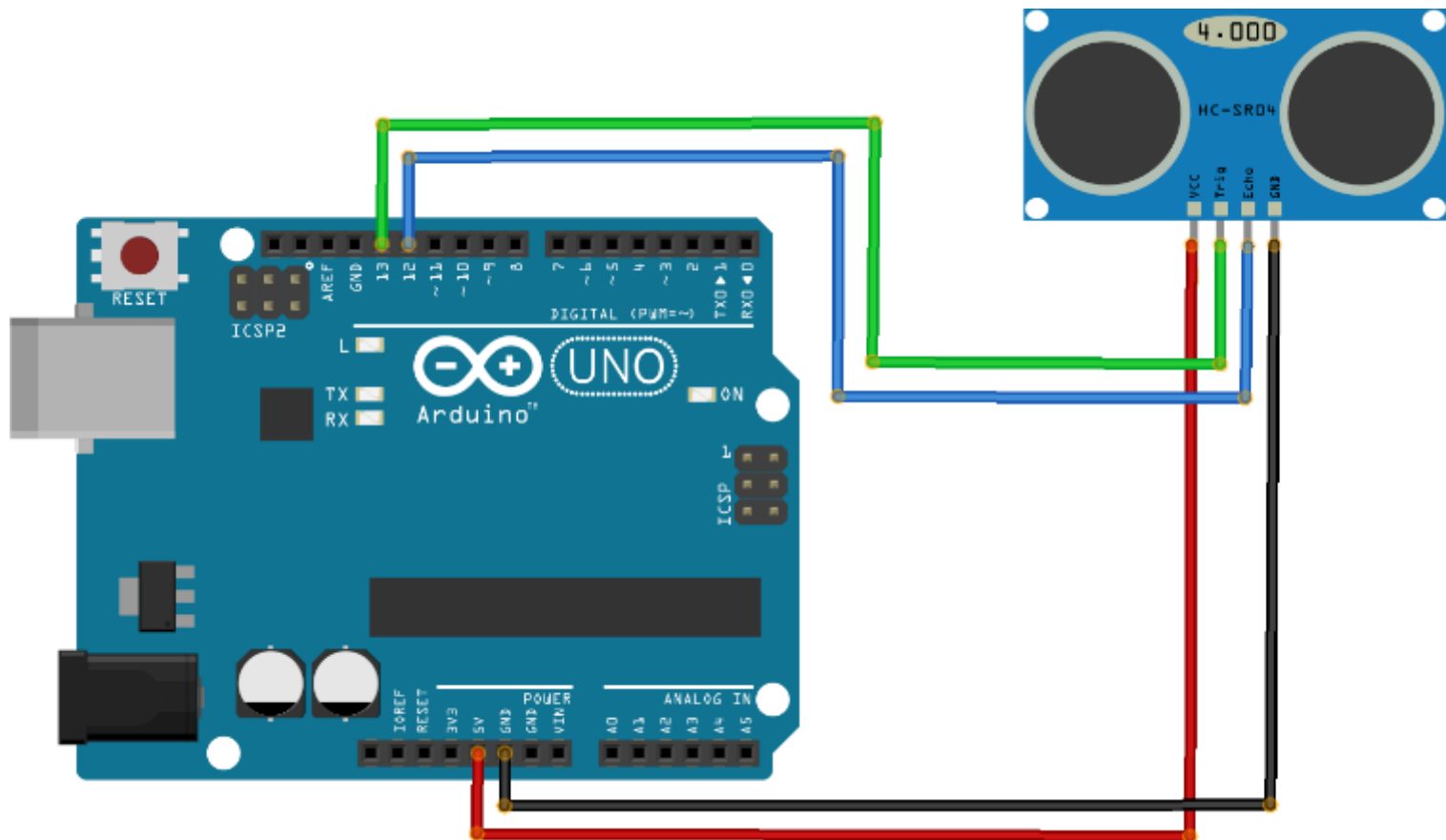
# Guided Tasks

- Controlling a Fan by using Temperature Sensor



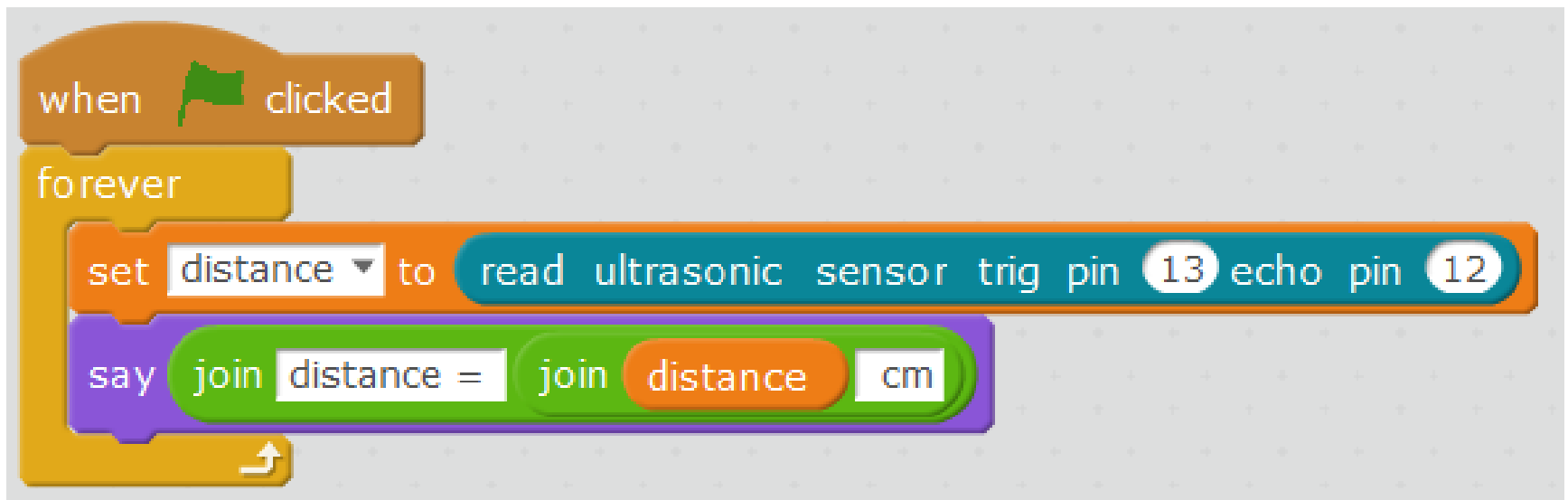
# Guided Tasks

- Measuring Distance by using Ultrasonic Sensor



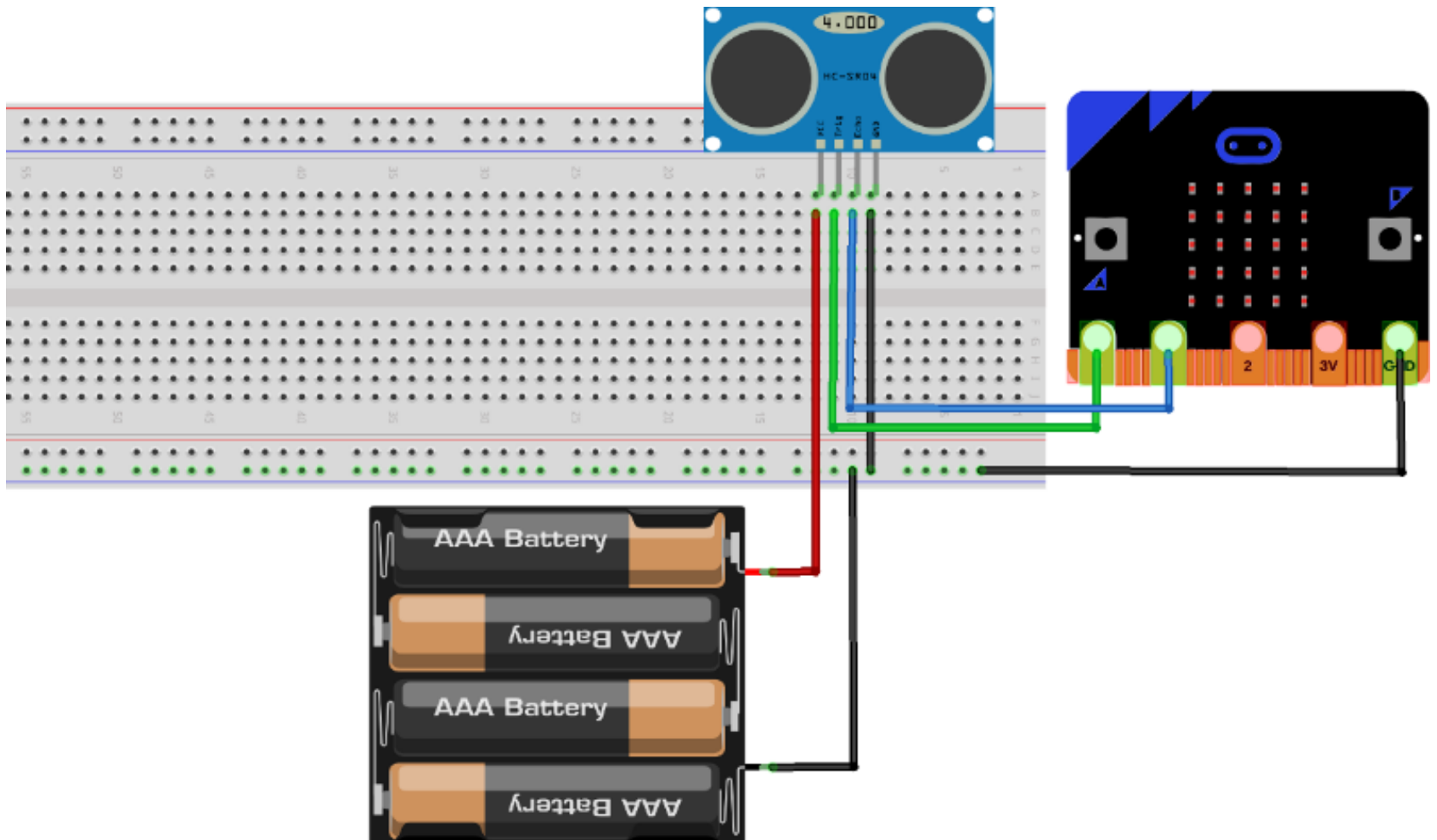
# Guided Tasks

- Measuring Distance by using Ultrasonic Sensor



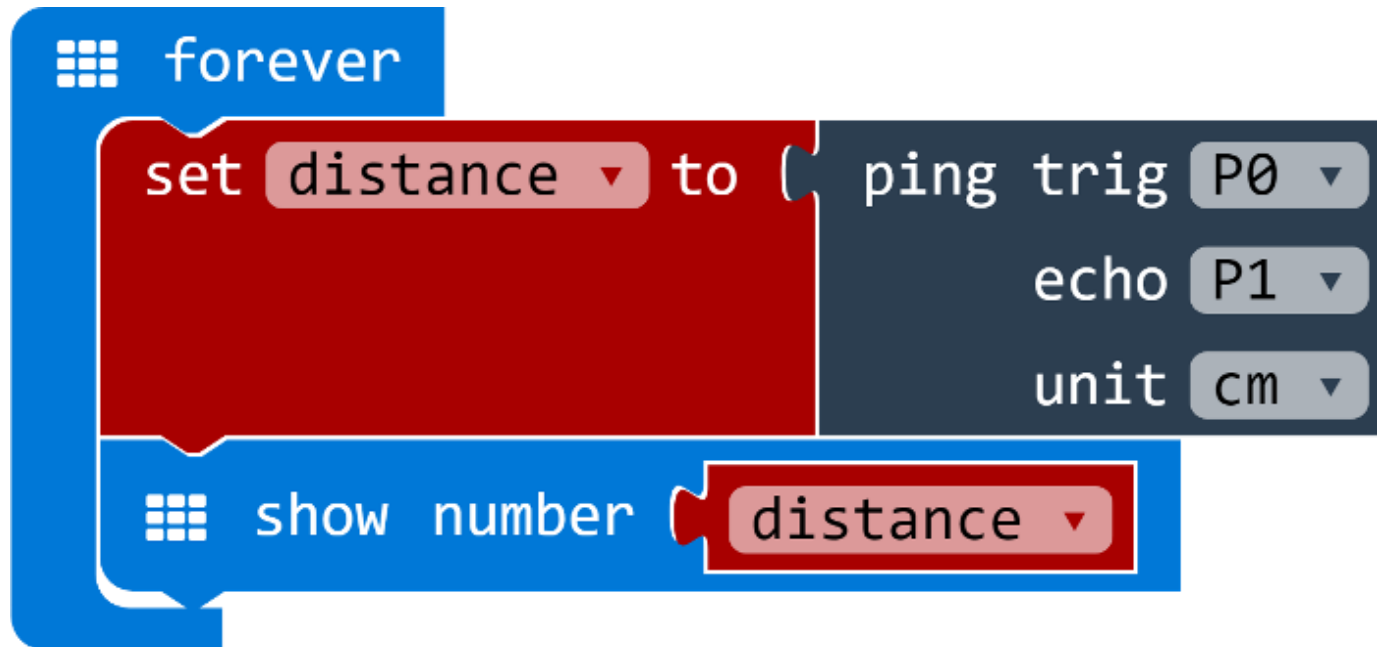
# Guided Tasks

- Measuring Distance by using Ultrasonic Sensor



# Guided Tasks

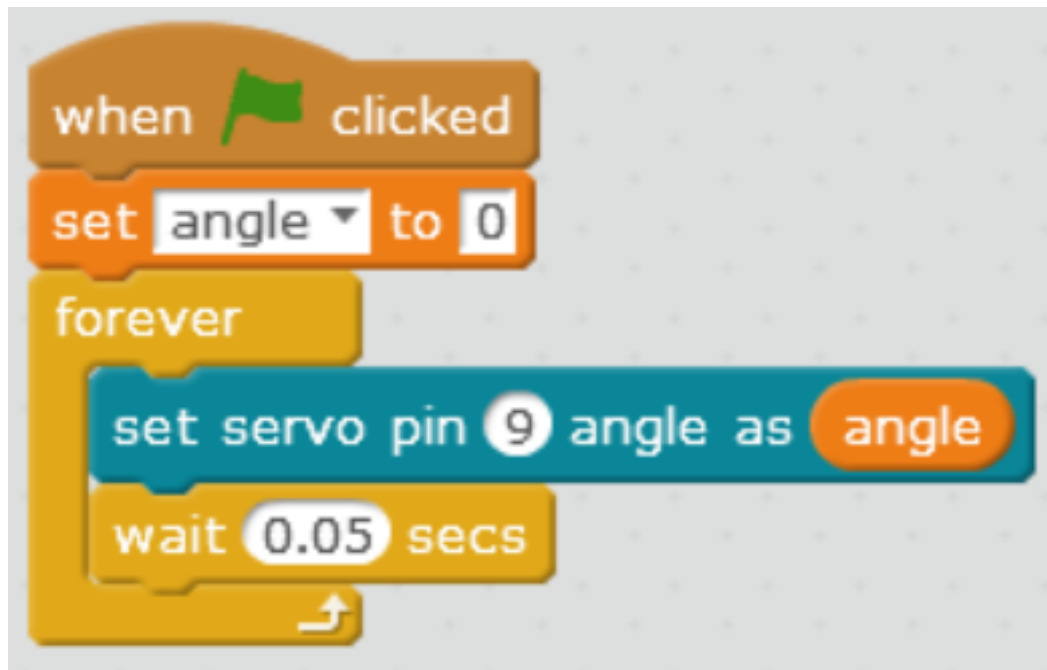
- Measuring Distance by using Ultrasonic Sensor





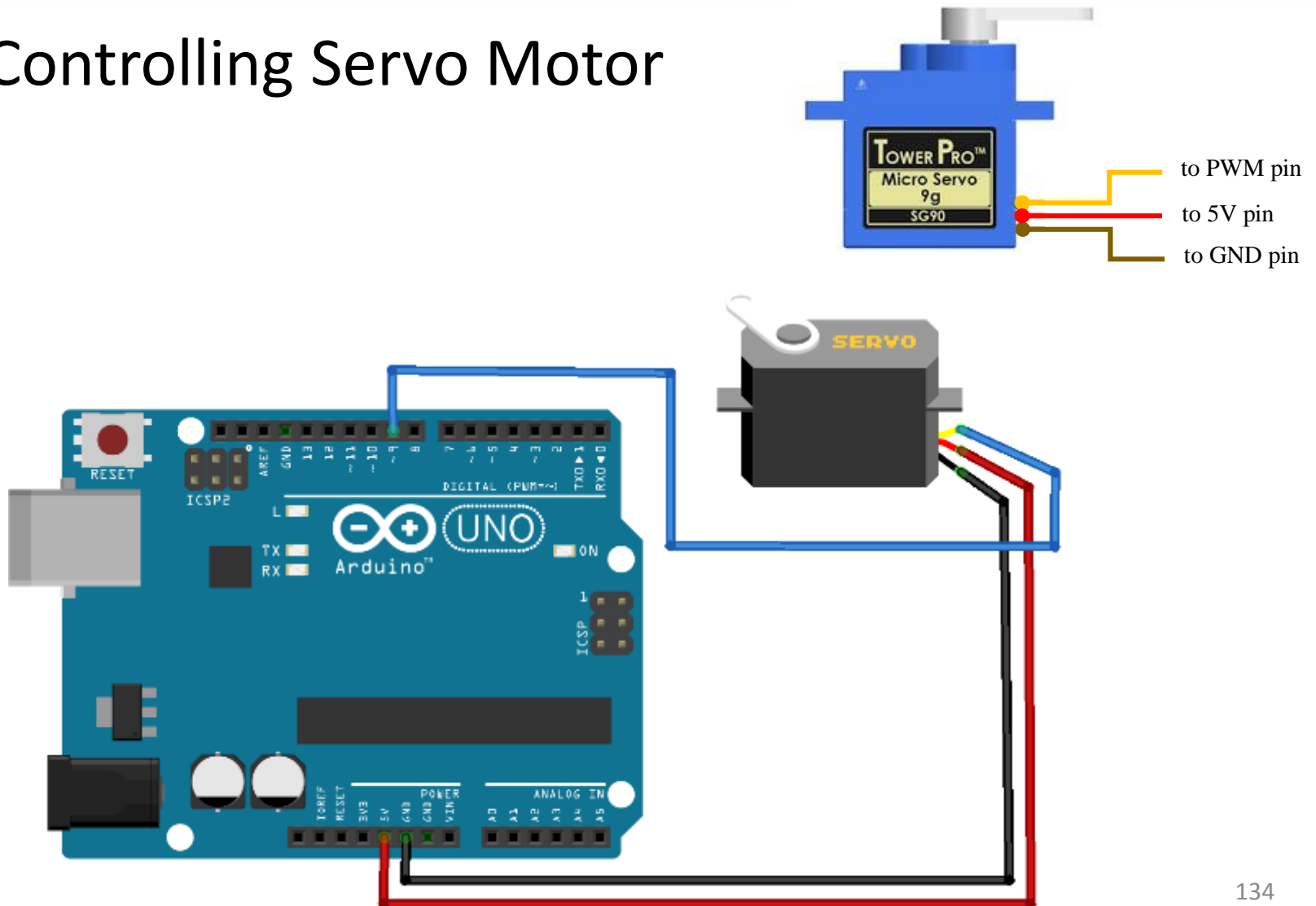
# Guided Tasks

- Controlling Servo Motor



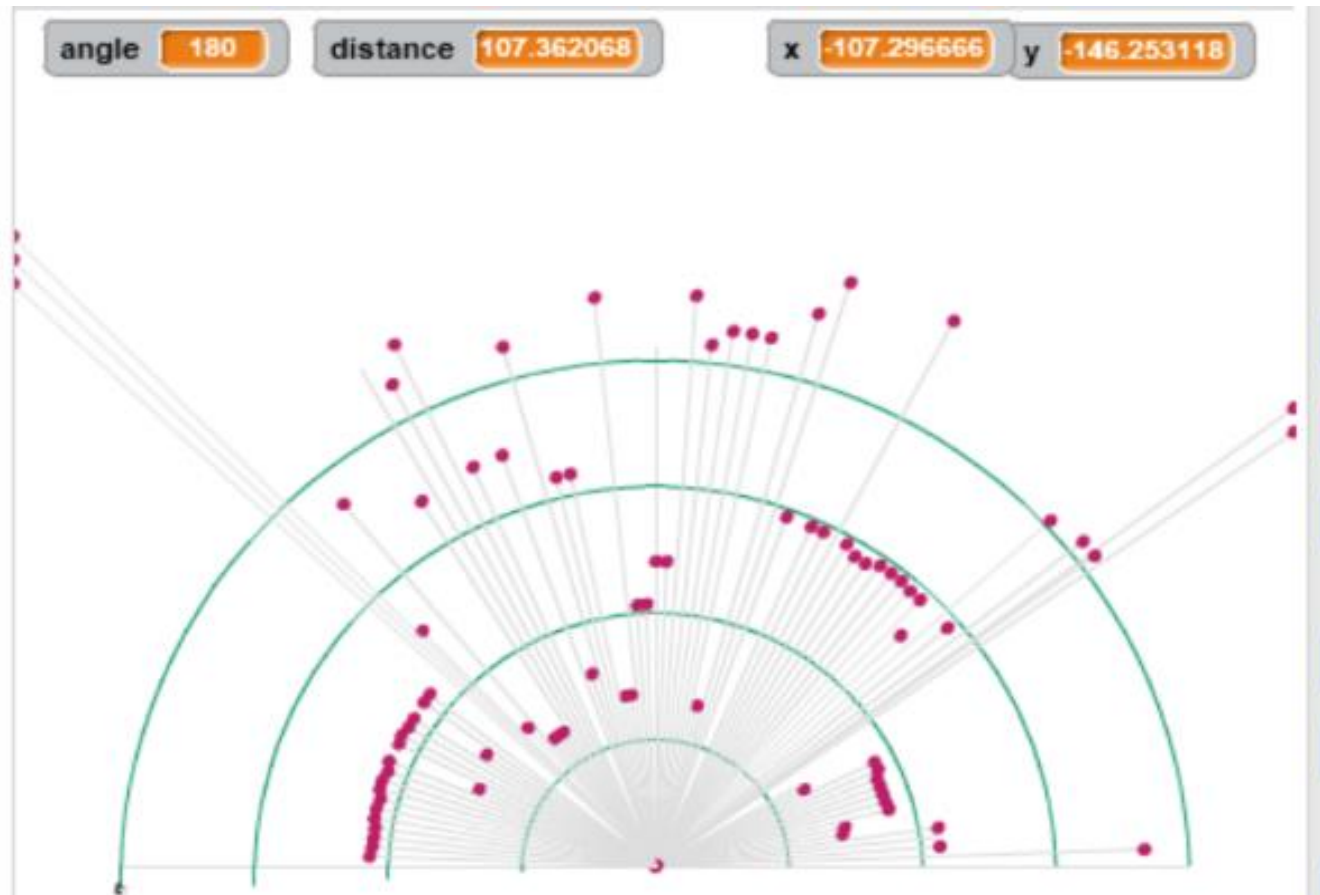
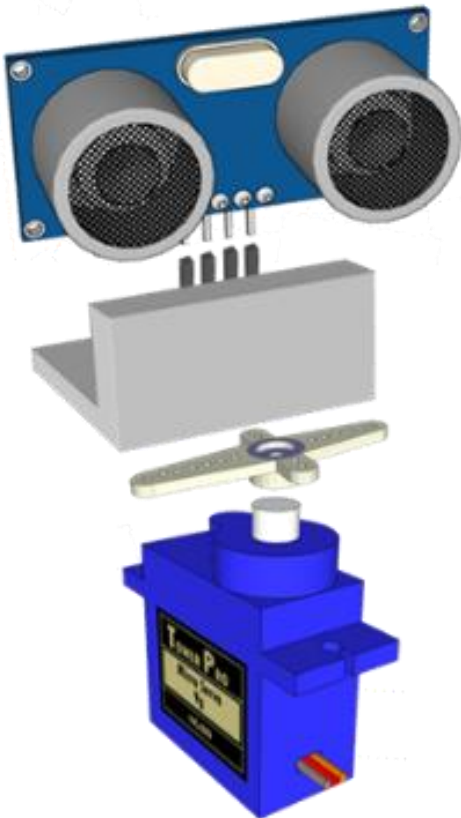
# Guided Tasks

- Controlling Servo Motor



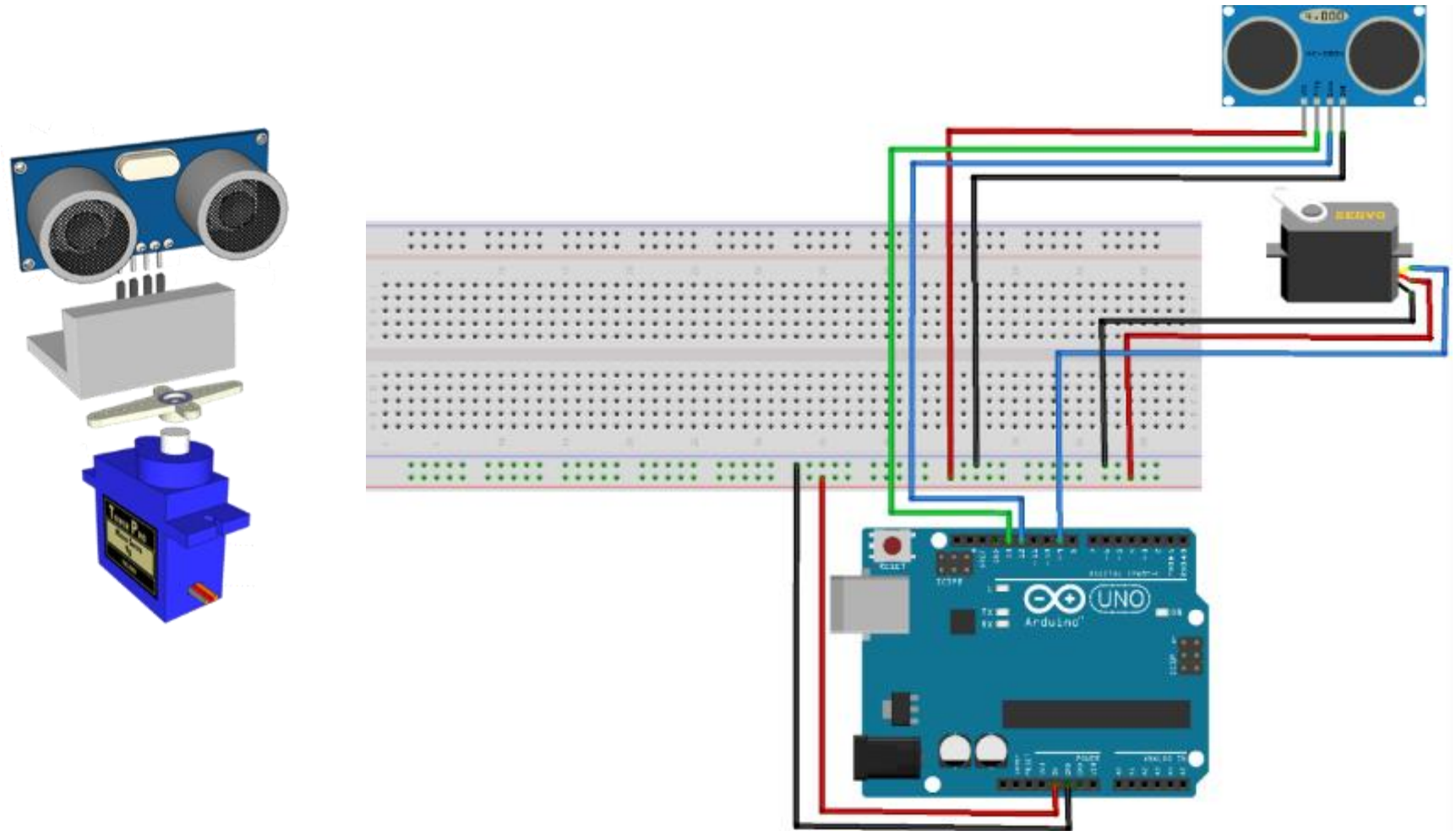
# Guided Tasks

- Servo Motor with Ultrasonic Sensor



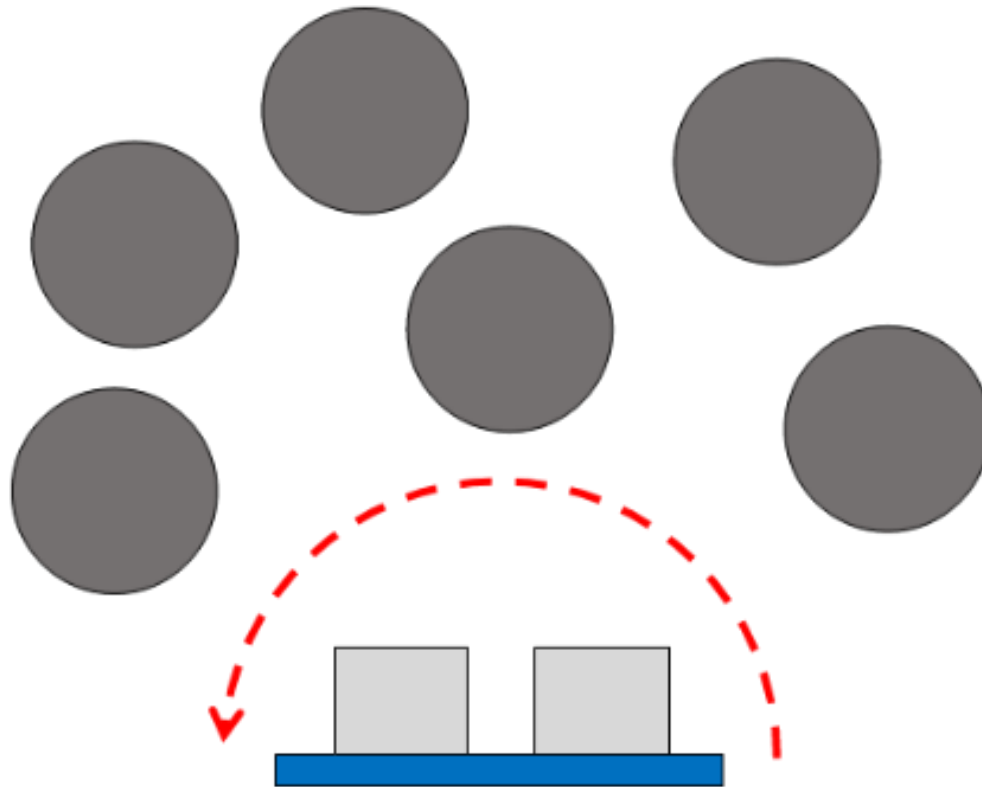
# Guided Tasks

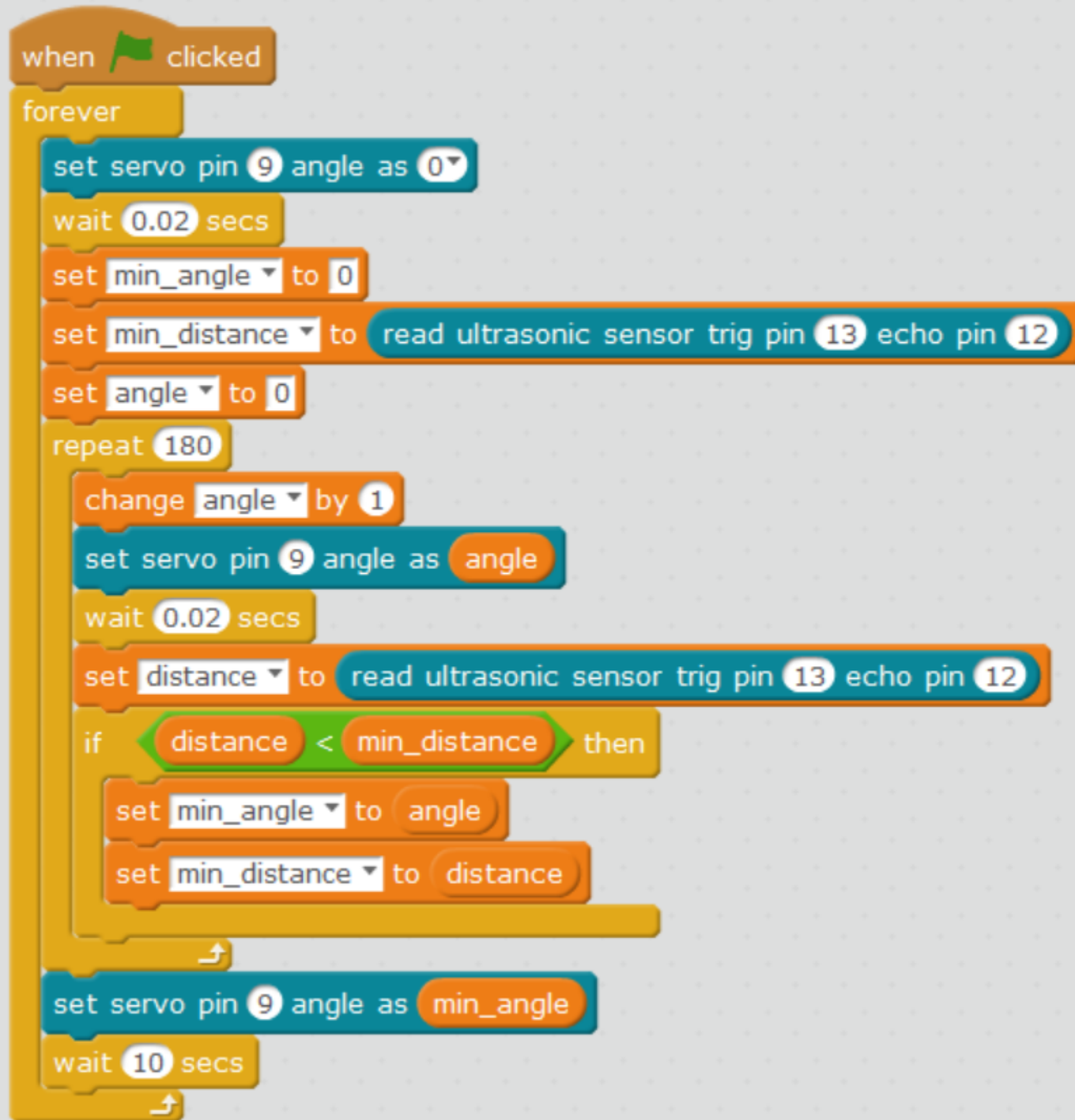
- Detecting the Nearest Obstacle to the Sensor



# Guided Tasks

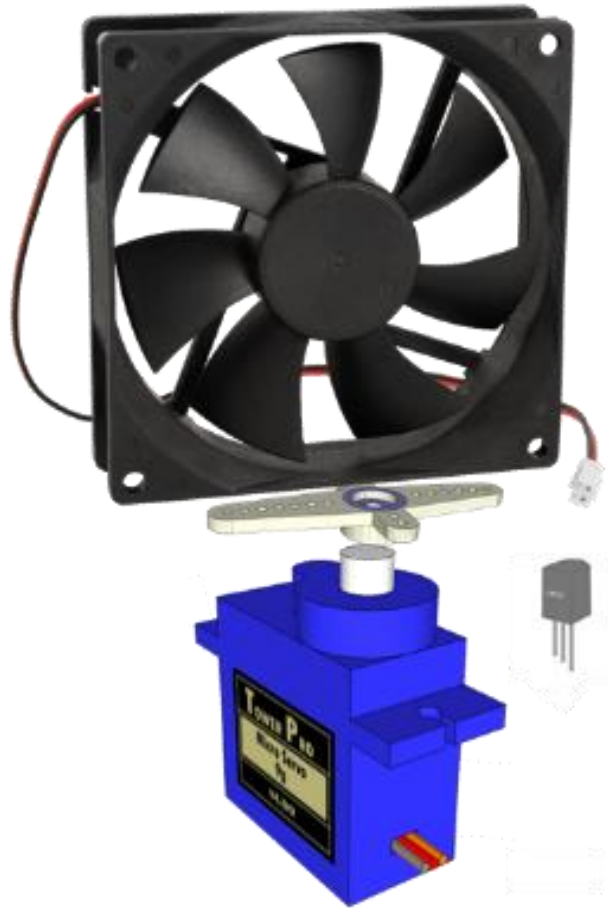
- Detecting the Nearest Obstacle to the Sensor





# Self-Exploration Project

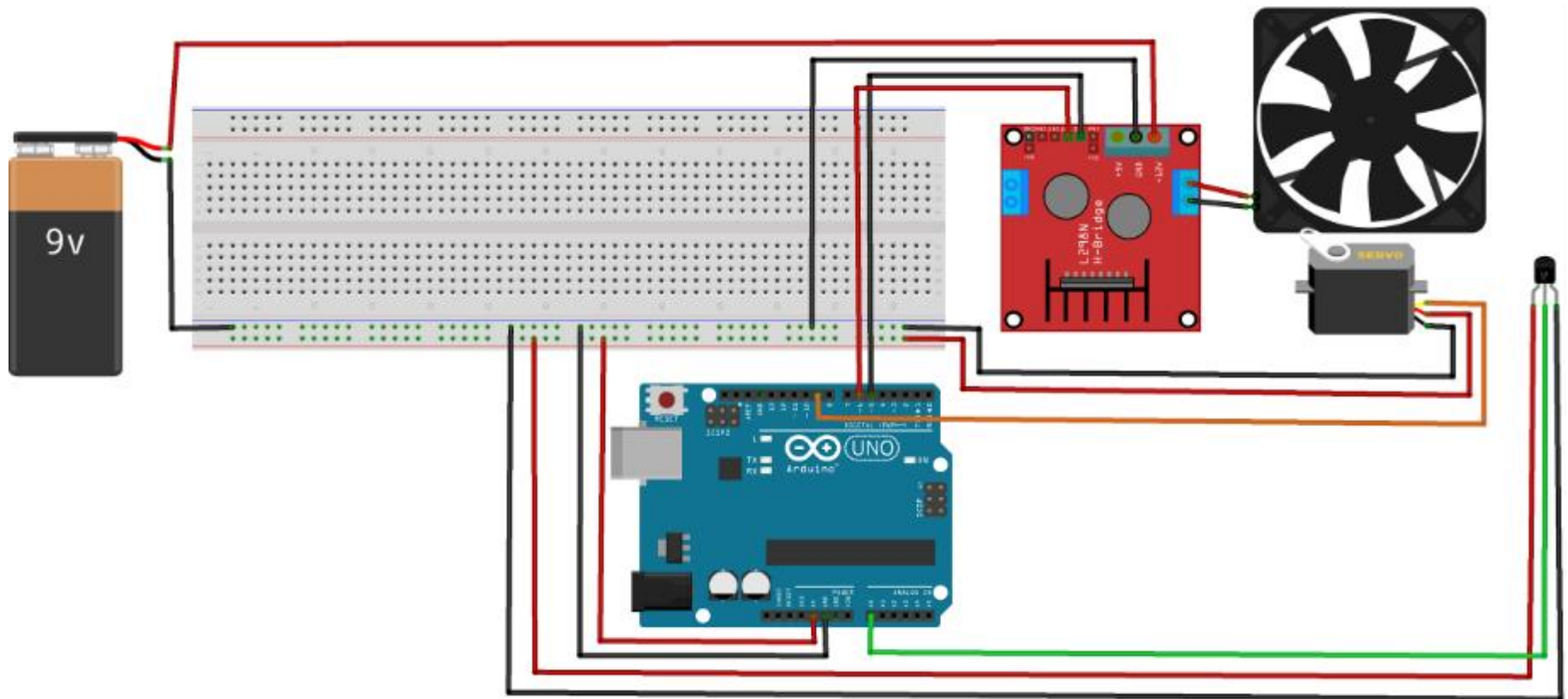
- Design a smart fan



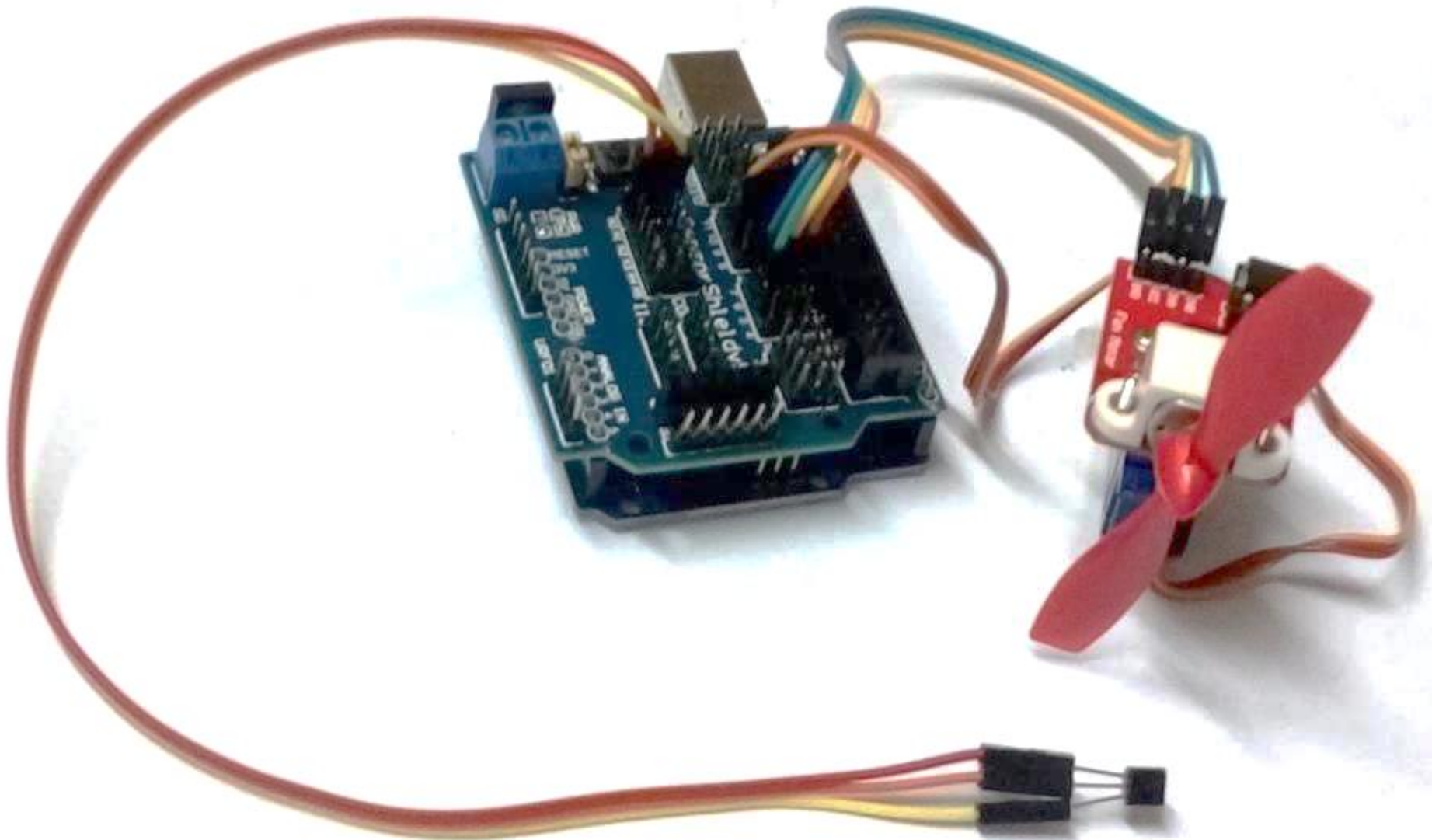


# Self-Exploration Project

- Design a smart fan



# Self-Exploration Project



when  clicked

set digital pin 5 output as LOW

set digital pin 6 output as LOW

set servo pin 9 angle as 0

set angle to 0

set step to 1

forever

set temp to round read analog pin (A) 0 \* 125 / 256

if temp < 25 then

set pwm pin 6 output as 0

else

setSpeed

setAngle

define setAngle

if angle = 180 then

set step to -1

if angle = 0 then

set step to 1

change angle by step

set servo pin 9 angle as angle

wait 0.02 secs

define setSpeed

if temp < 27 then

set pwm pin 6 output as 102

else

if temp < 29 then

set pwm pin 6 output as 153

else

if temp < 31 then

set pwm pin 6 output as 204

else

set pwm pin 6 output as 255

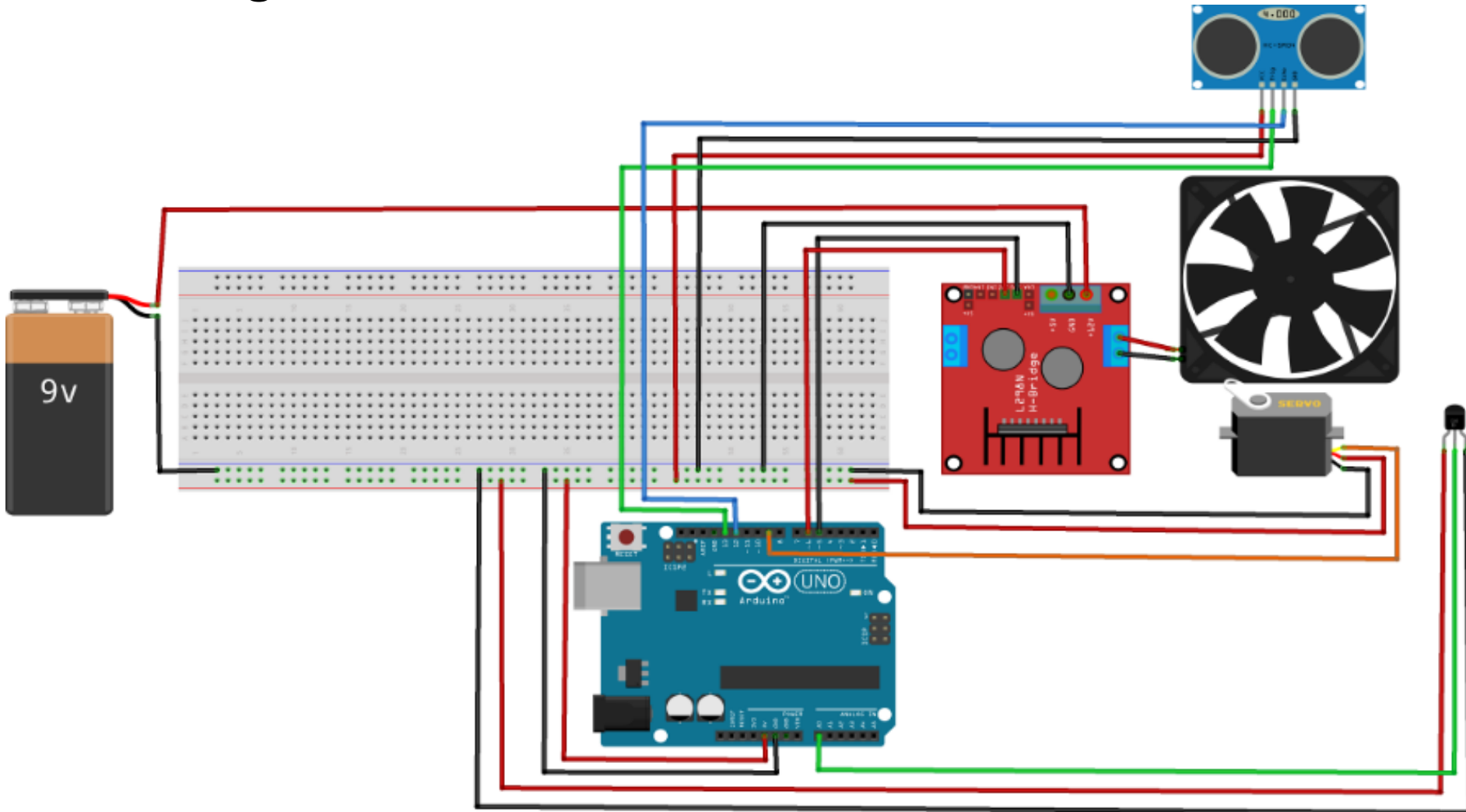
# Self-Exploration Project

- Design a smart fan



# Self-Exploration Project

- Design a smart fan



# Self-Exploration Project

- Further example
  - [https://www.youtube.com/watch?time\\_continue=79&v=nhLk\\_kOy6w8](https://www.youtube.com/watch?time_continue=79&v=nhLk_kOy6w8)



# Reference

- *Arduino輕鬆入門：範例分析與實作設計 / 葉難，博碩文化股份有限公司*
- *Beginning Arduino by Michael McRoberts*
- *用mBlock玩Arduino - Starting from Scratch：林信良*
- *etc.*



**~ END ~**

**Thank You!**