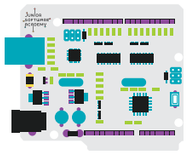
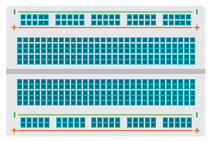
WEEK
04 Have LED My Way

Learning Objective

- Let's turn on the LED with a switch and understand analog output and digital input.

Materials to Prepare


making board


breadboard


USB cable


3 LEDs

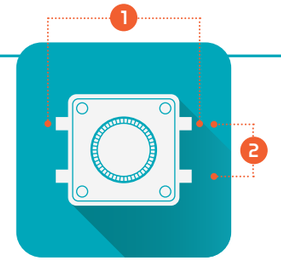

two 330Ω resistors


switch


wire


1KΩ resistor

PRACTICE Turning on LED with a Switch



1 Two pins are inter-connected.
2 Two pins are not inter-connected.

Do you remember the project we worked on in week 3? It was a program that had making board's LED blink every second through a simple program. We intend to make a program that connects LED, resistor, and switch to the breadboard and have the LED light up only when we press the switch this time.

When you press the button on the bottom, the two points in circuits are connected and this is when the current starts flowing. This thing that lets current flow when the circuit is connected and prevents current from flowing when the circuit is disconnected is called the switch. The exact name of the switch that we will be using is tact switch. The current flows only when you are pressing the switch. When you look at the inner structure, pins in groups of two are connected as you can see in the picture. All four legs are connected when you press the button. Therefore, the disconnected circuit becomes connected when you press the button and that is how the current starts flowing.

Closed and open circuits

Closed circuit refers to a circuit where current can flow as everything is connected. Open circuit, on the other hand, refers to a circuit where current cannot flow because some parts are disconnected. Closed circuits are also called close-hauled circuit and open circuit is called open-door circuits. Although 'close' has more negative connotation than the word 'open,' please remember that closed circuit is the one where the current is flowing.

Project Goal

Create a circuit that lights up LED when you press the switch.

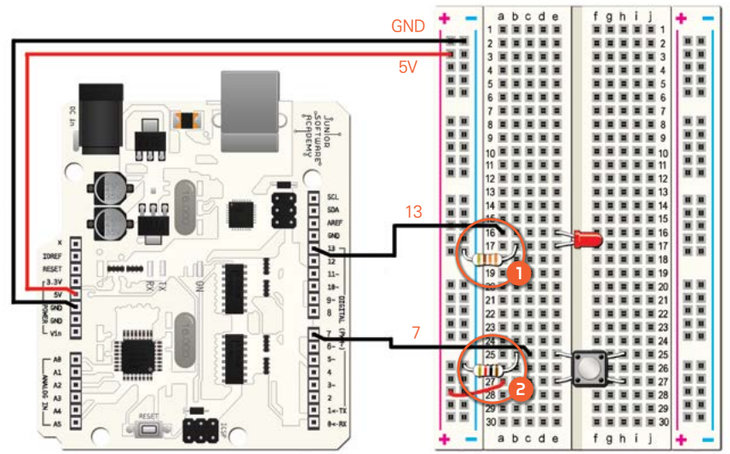
Hardware Expression

Let's look at the hardware diagram and try to understand how it works.

1 One side of the switch is connected to the digital pin7 and resistor (ground connected). The current does not flow to pin 7 because the circuit inside the switch is not connected.

2 When you are pressing the switch, 5V power is supplied to pin 7 so it becomes HIGH, or status 1.

3 When pin 7 is status 1, change the voltage of output pin 13 connected to LED to HIGH to turn on the LED.



1 330Ω resistor

2 1KΩ resistor

Thinking in Code

```
01  int led = 13;
02  int swPin = 7;
03  int swVal = 0;
04
05  void setup() {
06      pinMode(led, OUTPUT);
07      pinMode(swPin, INPUT);
08  }
09
10  void loop() {
11      swVal = digitalRead(swPin);
12      if(swVal == HIGH) {
13          digitalWrite(led, HIGH);
14      } else {
15          digitalWrite(led, LOW);
16      }
17  }
```

Define digital pin used by LED 13.
Define digital pin connected to switch
Apply initial value of O to the variable for saving the value of switch.

Set LED pin as output.
Set switch as input.

Read switch input value and save it in swVal variable.
When input value is HIGH (when the switch is pressed), turn on LED.
When input value is LOW (when the switch is not pressed), turn off LED.

1 int swVal = 0;

It is an int type variable for checking variable for saving the value whether the switch is turned on or off. Unlike invariables, variables refer to numbers that can change after it is initialized. Here, we declared initial value of variable swVal as O. When you press the switch, the variable turns into 1. On the contrary, when you detach your hand from the switch, the value of variable becomes 0 again.
The word int in front of the variable declares that it is a variable for saving an integer. You don't have to know this in detail for now but just try to understand the fact that you have to declare the form after considering how big the value is, whether the value is a real number with a decimal point or just a whole number. The int form can save any values between −32,768 and 32,767. It may be cumbersome but the reason why we categorize values is that making board's storage space is not sufficient so we have to use what we have efficiently.

② if statement

This checks whether the value of swVal that shows whether switch is pressed or not is HIGH or LOW. When it meets the condition, turn on or turn off the LED.

What is if statement?

'If statement' is a syntax that gives different results according to the condition. It is often called conditional statement. It is composed of conditional statement and sentence and when the condition of conditional statement is sufficed, the sentence is executed

From	Explanation

if(conditional expression) {
  sentence 1;
} else {
  sentence 2;
}
	If the conditional expression is true (if the condition is sufficed), sentence 1 right below is executed. However, if the conditional expression is false, then the sentence below is skipped moving on to the next code. The next code could be another conditional expression or could be else. When the condition is not sufficed, the sentence 2 below else is executed.

Example	**Explanation**

if(swVal == HIGH) {
  digitalWrite(LED, HIGH);
} else {
  digitalWrite(LED, LOW);
}
	If the input value is HIGH (switch is pressed), turn on LED.
	
	If the input value is LOW (switch is not pressed), turn off LED.

What does '==' mean here? The program becomes too long if we write everything in text, so we use symbols called relational operator to express many situations in the simplest manner. The following is a list of relational operators that are commonly used in conditional expressions.

a == b	Execute if a equals b	**a != b**	Execute if a does not equal b.
a < b	Execute if a is smaller than b	**a > b**	Execute if a is bigger than b
a <= b	Execute if a is smaller or equal to b	**a >= b**	Execute if a is bigger or equal to b

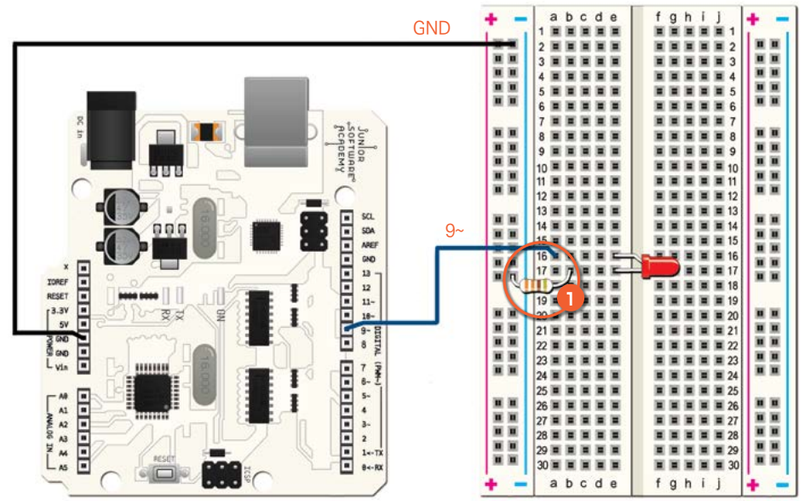
PRACTICE Turning on LED Softly

Project Goal

Let's have the LED turned on naturally soft.

Hardware Expression

The composition is very simple but make sure to connect both terminals with pin 9.



1 330Ω resistor

Thinking in Code

```
01  int ledPin = 9;                          Define digital pin used by LED 9
02
03  void setup() {
04                                           Don't' do anything here.
05  }
06
07  void loop() {
08      for(int fadeValue = 0; fadeValue <= 255; fadeValue += 5) {
09          analogWrite(ledPin, fadeValue);
10          delay(30);
11      }
12
13      for(int fadeValue = 255; fadeValue >= 0; fadeValue -= 5) {
14          analogWrite(ledPin, fadeValue);
15          delay(30);
16      }
17  }
```

What is for statement?

'For statement' is a type of loop and is composed of initial expression, conditional expression, and increment/decrement expressions.

Form	for (1 initial expression;2 conditional expression; 3increment/decrement expression)
	4 sentences
	}

Execution order of for statement is as follows. First, bring the result after executing 1. Initial expression and if it is true when compared to 2 conditional expression, execute 4 sentence and increase or decrease according to 3 increment/decrement expressions. Later, check 2 conditional expression again and execute 4 sentences again if it is true. This process repeats itself until 2 conditional expression becomes false (condition is not sufficed).
In other words, the process repeats itself in this 1243,243,243,243,··· order until 2 becomes false.

I will use the program that has three LEDs blink in interval of a second as an example.

Coding that does not use for statement

```
int ledPin1 = 5;
int ledPin2 = 6;
int ledPin3 = 7;

void setup() {
    pinMode(ledPin1, OUTPUT);
    pinMode(ledPin2, OUTPUT);
    pinMode(ledPin3, OUTPUT);
}

void loop() {
    digitalWrite(ledPin1, HIGH);
    delay(1000);
    digitalWrite(ledPin1, LOW);
    delay(1000);
    digitalWrite(ledPin2, HIGH);
    delay(1000);
    digitalWrite(ledPin2, LOW);
    delay(1000);
    digitalWrite(ledPin3, HIGH);
    delay(1000);
    digitalWrite(ledPin3, LOW);
    delay(1000);
}
```

Coding that uses for statement

```
int del=100;

void setup() {
    for(int i = 5; i <= 7; i++) {
        pinMode(i, OUTPUT);
    }
}

void loop() {
    for(int i = 5; i <= 7; i++) {
        digitalWrite(i, HIGH);
        delay(del);
        digitalWrite(i, LOW);
        delay(del);
    }
}
```

Digital Output and Analog Output

We only used executions to turn on or turn off up till now. We call it digital, where there are only two conditions of HIGH(1) or LOW(0). You have probably heard about how only 0 and 1 exist in the computer world. The function we use here is called digitalWrite. Analog, on the contrary, allows gradual change to the highest level like volume control. This project used analogWrite function, which allows analog output, to adjust the brightness in stages instead of turning on and off LED. The number O, as the minimum value, represents turned off state and 255, as the maximum value, represents the brightest state. For now, please just remember that LED's brightness changes slightly in proportion to the values in between. Let's look deeper into this material in week 11. You use digitalWrite function to output digital signal values. Please also note that you use analogWrite function in digital output pin number (~3,~5,~6, ~9, ~10, ~11) in order to output analog signal values.

You will be creating your own project next time. Think of an object such as a doll, a piggy bank, a pillow, etc. you want to attach LEDs on and bring the materials to class next time.

REVIEW Review

Let's think about your favorite part among today's activities and share with friends.

Let's think about the part you found the hardest and share with friends.

WEEK
05

CREATIVE DESIGN 01

Let's Talk in Lights

Learning Objective

Let's make my own project based on the things we learned until week 4.

Materials to Prepare

making kit, various objects (used objects such as doll, piggy bank, pillow, paper cup, .or ping pong ball), handicraft tools, scotch tape

PROJECT

Yogurt Illuminator

Project Goal

Let's make an illuminator by placing LED inside yogurt bottles.

1. Prepare multiple yogurt bottles that are empty.

2. Color or decorate or stick origami paper on to empty yogurt Bottles.

3. Connect wire to 3 to 5 LEDs. (you can connect more LEDs if you wish)

4. Punch holes in the bottom of the bottles and put wire connected LED inside the bottles and fix them.

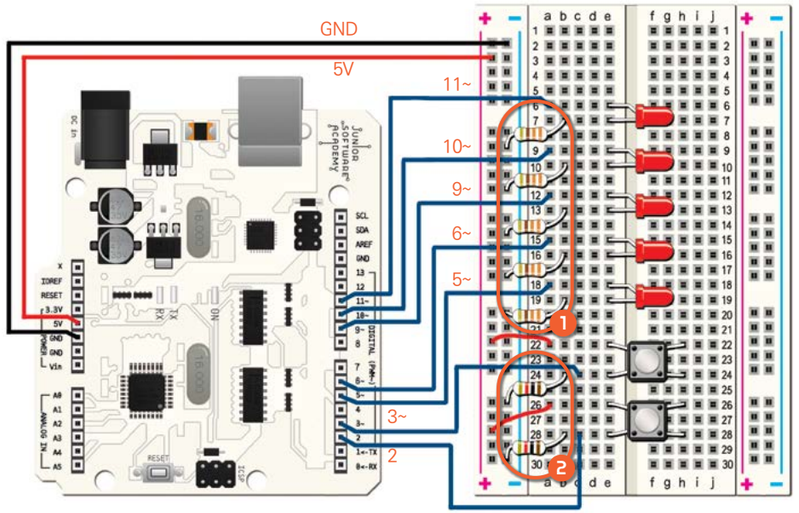
5. Place paper cups upside down and fix a switch on the bottom. Poke a hole on the side of the paper cup and thrust the wire and connects it with switch pins. The switch is used to turn on and turn off LEDs.

6. Connect the wires with LEDs inside the yogurt bottles. While you connect them with two wires of battery holders, have one side connected to the paper cup button. When you connect the wire, fix them sturdy with scotch tapes. Please note that two terminals of LED have to be connected to the correct polar terminal of batteries.

7. When you are done connecting, turn on the switch to check whether the LED gets turned on.

Hardware Expression

This is a hardware composition and source you can use as a reference when making something that not just turns on or off but can carry out various motions. When you become creative with your projects, create something fabulous using



1 330Ω resistor 

2 1KΩ resistor 

Thinking in Code

```
01  const int led1 = 5;                          Define digital pin used by the first LED as 5.
02  const int led2 = 6;                          Define digital pin used by the second LED as 6.
03  const int led3 = 9;                          Define digital pin used by the third LED as 9.
04  const int led4 = 10;                         Define digital pin used by the fourth LED as 10.
05  const int led5 = 11;                         Define digital pin used by the fifth LED as 11.
06
07  const int swPin1 = 2;                        Define digital pin used by the first switch as 2.
08  const int swPin2 = 3;                        Define digital pin used by the second switch as 3.
09
10  int swValue1 = 0;                            Variable to tell on/off status of swPin1
11  int swValue2 = 0;                            Variable to tell on/off status of btPin2
12
13  int count = 0;                               Variable to tell what number button each button is.
14
15  void setup() {
16    pinMode(swPin1, INPUT);                    Set swPin1 as input mode
17    pinMode(swPin2, INPUT);                    Set swPin2 as input mode
18
19    pinMode(led1, OUTPUT);                     Set led1 as output mode
20    pinMode(led2, OUTPUT);                     Set led2 as output mode
21    pinMode(led3, OUTPUT);                     Set led3 as output mode
22    pinMode(led4, OUTPUT);                     Set led4 as output mode
23    pinMode(led5, OUTPUT);                     Set led5 as output mode
24  }
25
26  void loop() {
27    swValue1 = digitalRead(swPin1);            Read on/off status value of swPin1 and save it to swV-
                                                 laue1.
28    swValue2 = digitalRead(swPin2);            Read on/off status value of swPin2 and save it to swV-
                                                 laue2.
29
30    if(swValue1 == HIGH) {                     If sw/Value1 is on (if swPin 1 is pressed), save 1 as
31      count = 1;                               count variable
32    }
33    if(swValue2 == HIGH) {                     If sw/Value2 is on (if swPin 2 is pressed), save 2 as
34      count = 2;                               count variable
35    }
36    if(swValue1 == HIGH &&  swValue2 == HIGH) {  If both sw/Value1 and sw/Value2 are on (if 2switches
37      count = 3;                               are pressed), save 3 as count variable
38    }
39
```

```
40    if(count == 1) {
41      digitalWrite(led1, HIGH);
42      digitalWrite(led2, HIGH);
43      digitalWrite(led3, HIGH);
44      digitalWrite(led4, HIGH);
45      digitalWrite(led5, HIGH);
46      delay(300);
47      digitalWrite(led1, LOW);
48      digitalWrite(led2, LOW);
49      digitalWrite(led3, LOW);
50      digitalWrite(led4, LOW);
51      digitalWrite(led5, LOW);
52      delay(300);
53    }
54    else if(count == 2) {
55      digitalWrite(led1,HIGH);
56      delay(300);
57      digitalWrite(led1, LOW);
58      delay(300);
59      digitalWrite(led2,HIGH);
60      delay(300);
61      digitalWrite(led2, LOW);
62      delay(300);
63      digitalWrite(led3,HIGH);
64      delay(300);
65      digitalWrite(led3, LOW);
66      delay(300);
67      digitalWrite(led4,HIGH);
68      delay(300);
69      digitalWrite(led4, LOW);
70      delay(300);
71      digitalWrite(led5,HIGH);
72      delay(300);
73      digitalWrite(led5, LOW);
74      delay(300);
75    }
76    else if(count == 3) {
77      for(int i = 0; i < 255; i += 5) {
78        analogWrite(led1, i);
79        delay(20);
80      }
```

When the count variable is 1

When the count variable is 2

When the count variable is 3

```
81      for(int i = 0; i < 255; i += 5) {
82        analogWrite(led2, i);
83        delay(20);
84      }
85      for(int i = 0; i < 255; i += 5) {
86        analogWrite(led3, i);
87        delay(20);
88      }
89      for(int i = 0; i < 255; i += 5) {
90        analogWrite(led4, i);
91        delay(20);
92      }
93      for(int i = 0; i < 255; i += 5) {
94        analogWrite(led5, i);
95        delay(20);
96      }
97      for(int i = 255; i >= 0; i -= 5) {
98        analogWrite(led5, i);
99        delay(20);
100     }
101     for(int i = 255; i >= 0; i -= 5) {
102       analogWrite(led4, i);
103       delay(20);
104     }
105     for(int i = 255; i >= 0; i -= 5) {
106       analogWrite(led3, i);
107       delay(20);
108     }
109     for(int i = 255; i >= 0; i -= 5) {
110       analogWrite(led2, i);
111       delay(20);
112     }
113     for(int i = 255; i >= 0; i -= 5) {
114       analogWrite(led1, i);
115       delay(20);
116     }
117   }
118 }
119
```

Just like this, you can control 5 LEDs with two switches in three different modes. It works in the following three modes: when you press switch 1; when you press switch 2; when you press both 1 and 2. Did you make an exciting project?

- When the count is 1, all LEDs turn on off together with an interval period of 0.3 seconds. We used digitalWrite function here.
- When the count is 2, LEDs turn on for 0.3 second and turn off for 0.3 second in order. This process is repeated from LED 1 to LED5 one after another.
- When the count is 3, LED1~5 turns on one after another in an interval period of 5.12(20*256ms) seconds and turns off in the reverse order in an interval period of 5.12 seconds. We used analogWrite in turning on and off LEDs and this function can express the brightness of light in 256 stages. The level of brightness starts as O when you turn on the light and gets brighter until stage 256. It works the other way round when you turn it off.

If statement on row 36 uses two conditions. Likewise, if statements can distinguish multiple conditions and if you place '&&(and)' between conditions, the machine operates when both conditions are met and if you place 'II(or),' then the machine operates when either conditions are met. We used '&&(and)' to have the LED operate when both switches are pressed.

This code operates in consecutive order so when the third function (count == 3) led is operating, it does not change to another function even if you press the button. Therefore, in order to switch to another function, you have to press the button when it reaches the end of code or you have to have the switch pressed until another function is operated. As making board uses delay, there is nothing we can do to solve this with the present code.

CREATIVE PROJECT

Let's Talk in Lights

Shall we make a project of our own?

You can create fabulous works by applying what you learned until week 4 and what you learned during the yogurt illuminator project.

1 Sitting in groups

Sit in groups of 4 to 6. Think about what to make as a group for five minutes. Write down some sketches while you are brain storming. Each one of you then talks about your idea and what to make to your friends for a minute or two. Group friends should not be judging whether the project is good or bad. Instead, provide constructive advices that can help your friend's idea grow.

2 Preparing

Now start preparing for your project in earnest. Prepare a making kit, other materials for handicraft that go well with LED lights. You brought the materials with you to class, right?

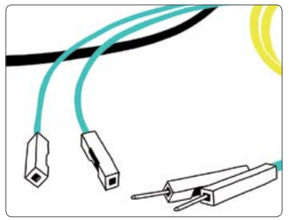
3 Summarizing ideas

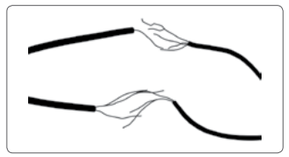
Make brief summary of your idea in pictures and writings in a notebook. Then come up with a sketch, composition of hardware, a circuit and describe what you are going to do and what motions you are going to make in detail.

4 Connecting Circuit

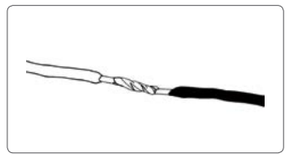
Building upon what you learned between week2 and week 4, make a project using a button and LED. You might need a long wire and you might have to connect two wires. It would be most sturdy if you soldered the wires together but this time, let's work without soldering. Fix two wires between components together by twisting them. You can also fix them with a glue gun after twisting the wires between the components.

There are wires that look like the picture inside the kit. When the wire is short, you can combine socket wire and jumper wire. If the connection part is loose, wrap a tape around.

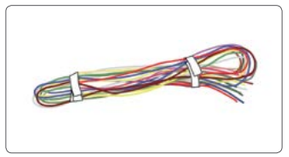




Twisting the wires



Soldering



Bundle of wires

5 Presentation

As the last part of this class, name your project and make a presentation about your creation.
During presentation, you should go over project name and descriptions (why it was made; how it was made; how it motions) and listen to what your friends think about your project.

REVIEW Review

Let's think about your favorite part among today's activities and share with friends.

Let's think about the part you found the hardest and share with friends.