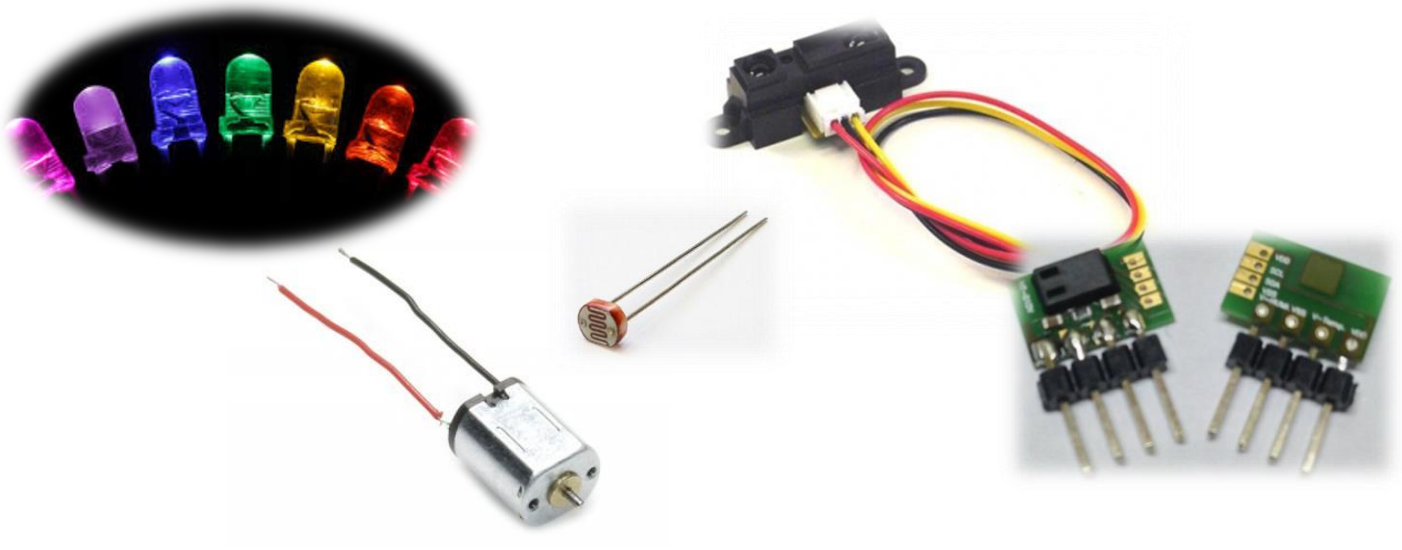
# Ⅱ. ARTIK 기술 교육

## 2. PWM Control



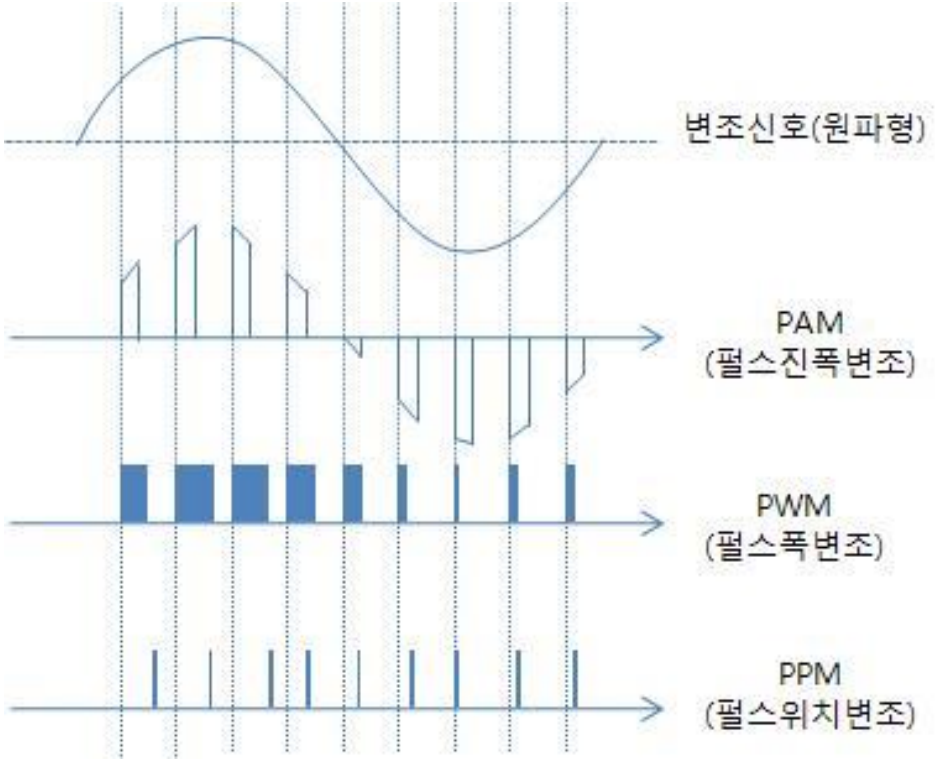
# Pulse Modulation

## ■ PM(Pulse Modulation)

- A method of modulating a periodic pulse by a signal

## ■ Type of Pulse Modulation

- PAM(Pulse Amplitude Modulation)
- PWM(Pulse Width Modulation)
- PPM(Pulse Position Modulation)



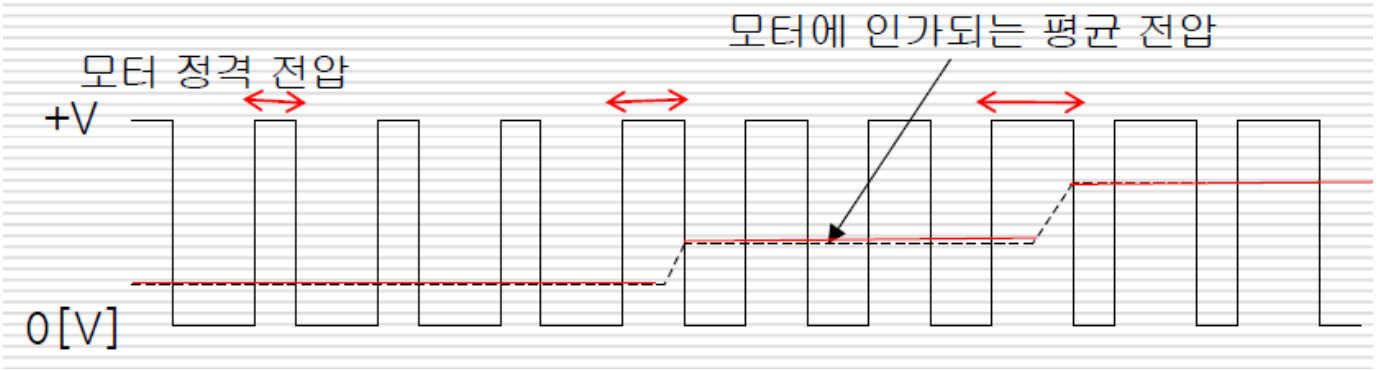
# PWM(Pulse Width Modulation)

## ■ Pulse

- Signal level changes over a short period of time
- Pulse Width
- The width of one pulse

## ■ PWM

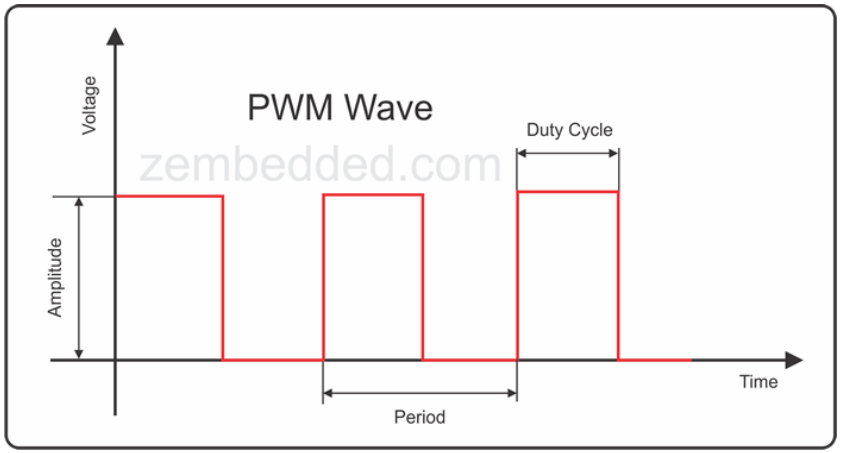
- Methods of adjusting 'ON' or 'OFF' time of pulse
- It is able to adjust pulse width and control motor because the average voltage applied to the motor can be changed and the effect of changing the magnitude of the voltage can be obtained, in the case of a DC motor.



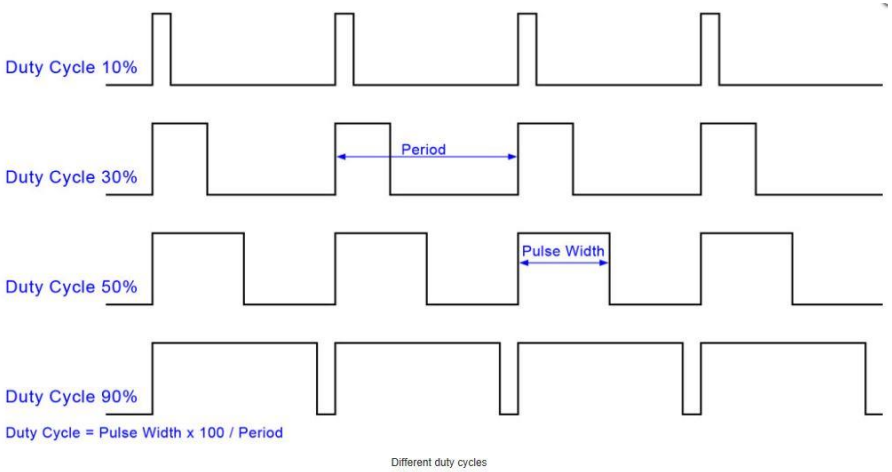
# PWM(Pulse Width Modulation)

## ■ PWM control

- Period : Time at any regular interval
- Duty Cycle : The ratio of 'ON' time in a period




(1) PWM Wave

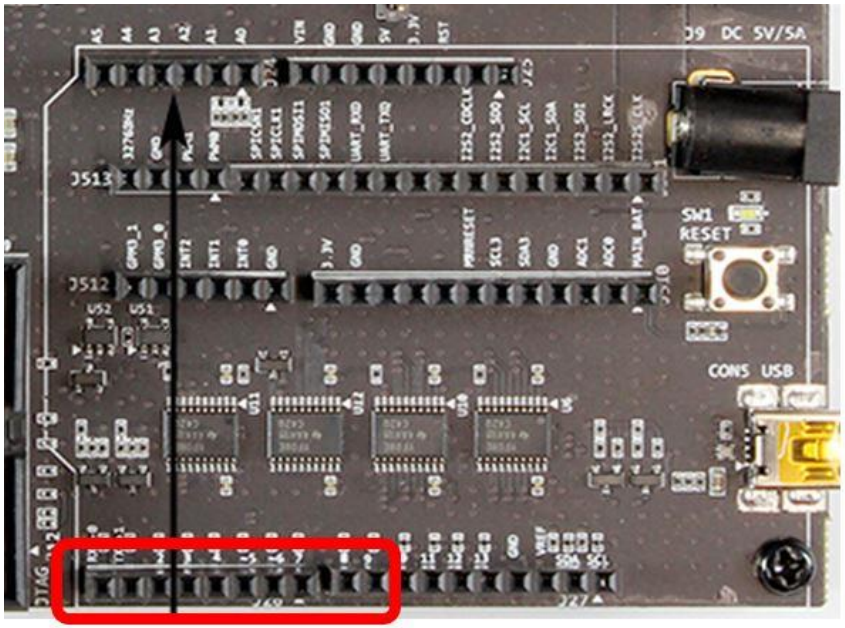



(2) Duty Cycle

# PWMs in ARTIK

## ■ PWM pins

- J26[5] (XPWMO0)
- J26[6] (XPWMO1)




(1) ARTIK5 Board

Header J26

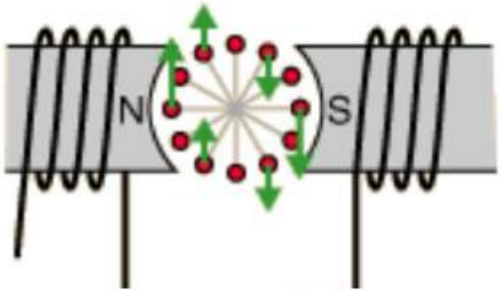
| PIN [SILKSCREEN] | MAPPING |
|---|---|
| J26[Rx-0] | Rx |
| J26[Tx-1] | Tx |
| J26[2] | GPIO 121 (GPX0[0]) |
| J26[3] | GPIO 122 (GPX0[1]) |
| J26[4] | GPIO 123 (GPX0[2]) |
| J26[5] | XPWMO0 |
| J26[6] | XPWMO1 |
| J26[7] | GPIO 124 (GPX0[3]) |

(2) ARTIK5 Pin mapping(J26)

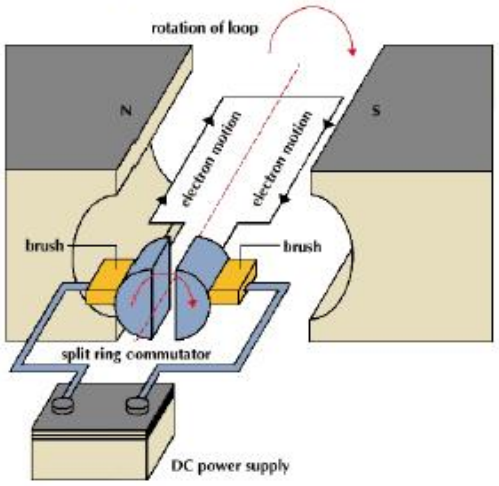
# DC Motor

## ■ DC Motor

- A motor, which uses DC power, rotates with a repulsive force between the magnetic field of the permanent magnet and the electromagnetic force generated by the current flowing in the conductor lying in the magnetic field.
- Electrical flame, rotation noise at contact point because of brush and shortening of life.
- Large driving torque
- Rotation characteristics linearly proportional to the applied voltage
- Output torque linearly proportional to input current
- Good output efficiency and low cost



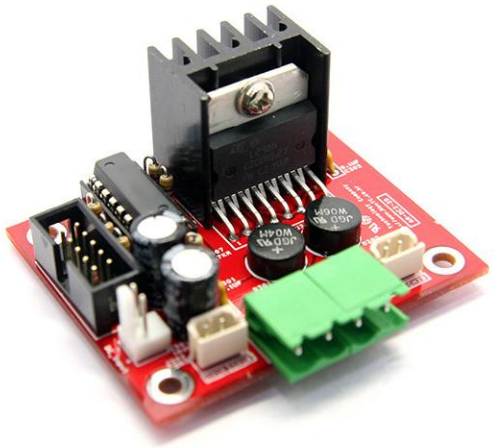




# DC Motor

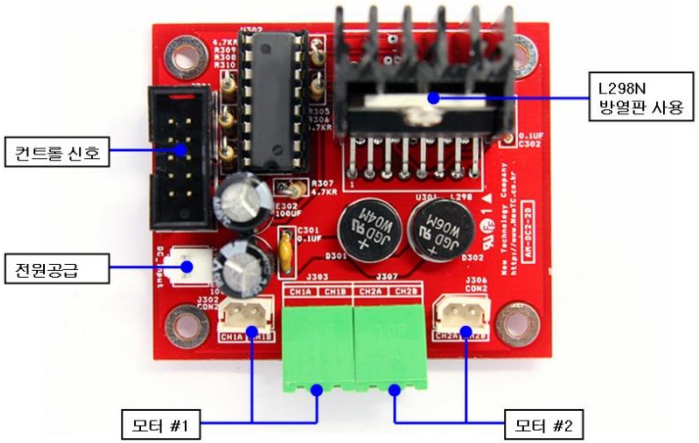
## ■ Motor Driver(AM-DC2-2D)

- Use to control motor rotation direction and driving speed
- Control by 3 signals (Enable, Dir, PWM)

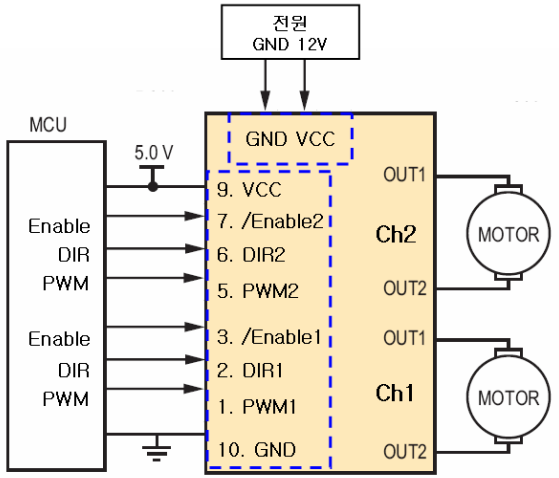


| Model | AM-DC2-2D |
|---|---|
| Motor control | 2EA |
| Maximum current | 2A |
| Driving voltage | 5V~45V |
| Controller | L298N(1EA) |
| Control signal | /Enable<br>DIR, CLK |
| Board size | 58·50(mm) |

(1) Motor Driver Spec




(2) Motor Driver H/W

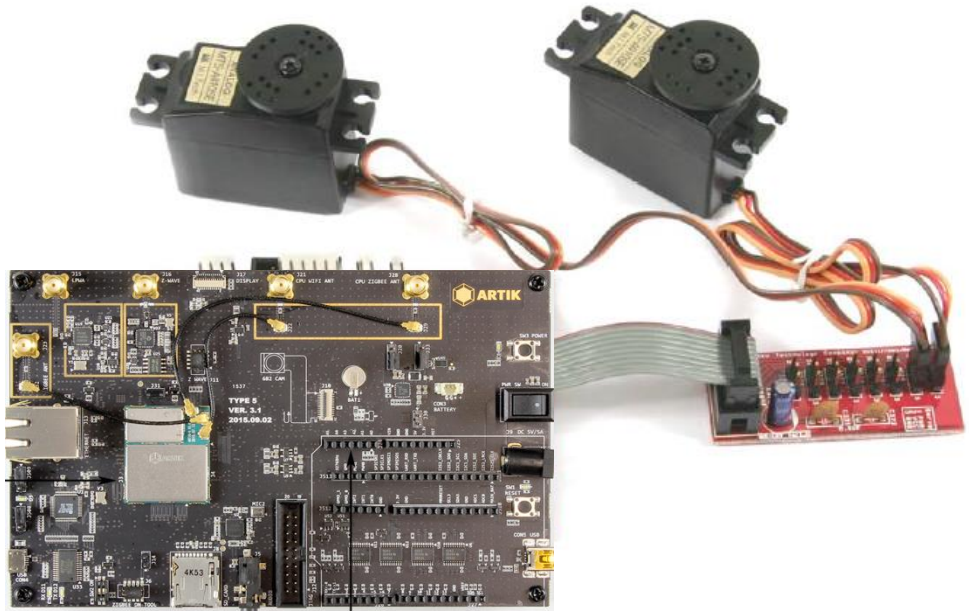



(3) Motor Driver blockdiagram

# Servo Motor

## ■ RC Servo Motor

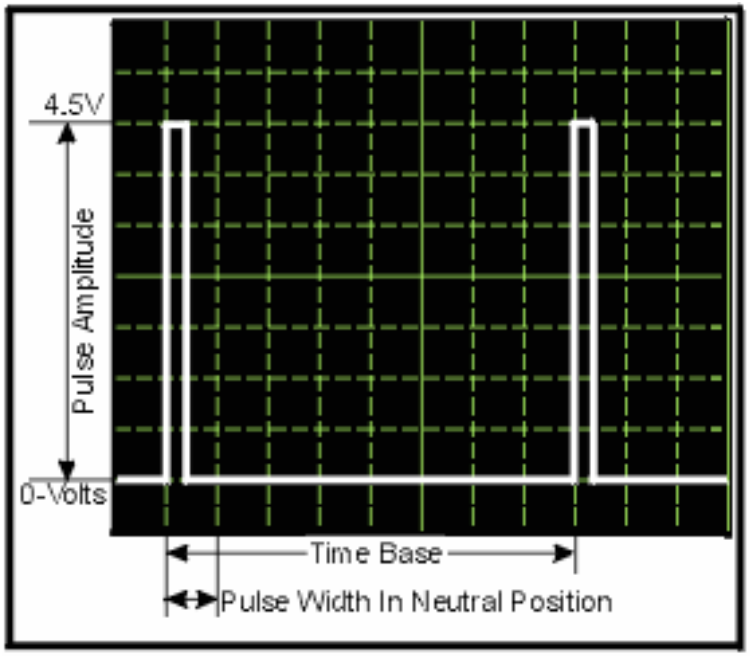
- Used for Radio Control's airplane, automobile, rear-wheel wireless models
- Open-loop control
- Able to produce large force with small size by using large gear ratio
- Mainly use in small size robots
- The operating range of the servo is -60° ~ +60°, -90° ~ +90°, torque and response speed are different, depending on servo characteristics and price.
- There are three signal lines(VCC, GND, control line)

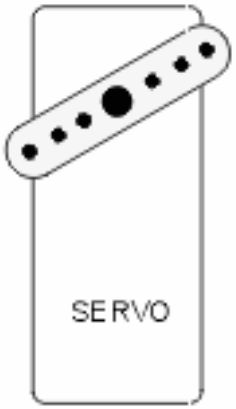


# Servo Motor

## ■ RC Servo Motor control

- Passing target angle through control line(specified by pulse width) is called by PCM (Pulse Code Modulation)
- Pulse period : 20msec
- Generally, 1.5 msec is a neutral point. If it is shorter than 1.5 msec, the motor moves toward 0 degree. If it is longer than 1.5 msec, the motor moves toward180 degree.

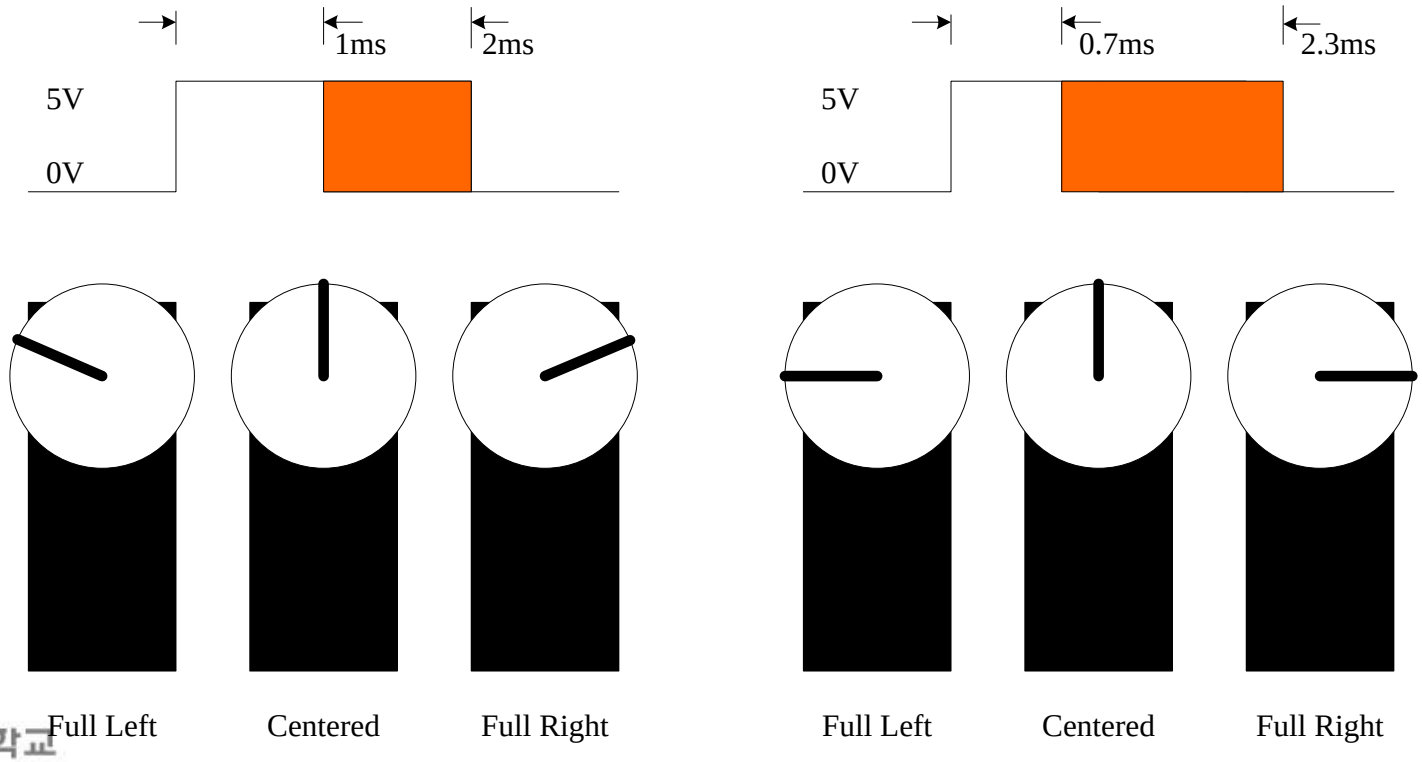


# Servo Motor

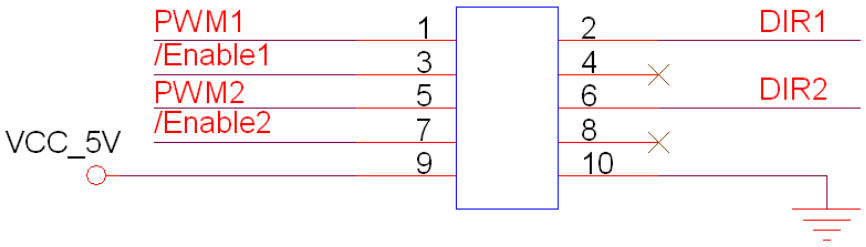
## ■ RC Servo Motor control

- less than 0.7ms → 0° (90° to the left)
- 1.5ms→90° (neutral point)
- More than 2.3m →180°(90° to the right)



# DC Motor

## ■ Motor Driver(AM-DC2-2D) connection methods




Number of Pin

| 1 PIN | 3 PIN | 5 PIN | 7 PIN | 9 PIN |
|---|---|---|---|---|
| PWM1 | /Enable1 | PWM2 | Enable2 | 5V |
| 2 PIN | 4 PIN | 6 PIN | 8 PIN | 10 PIN |
| DIR1 | | DIR2 | | GND |

(1) Motor Driver Pin Connection

| Signal | Motion |
|---|---|
| /Enable | The motor of the corresponding channel can be operated or stopped.<br>0: Enable<br>1: Disable |
| Dir | It is able to determine the orientation of the motor.<br>0: Rotation in negative direction<br>1: Rotation in positive direction<br>(It can be different because of different motor connection) |
| PWM | It is used to adjust the speed of the motor.<br>0: Motor stop<br>1: Motor rotation start<br>(The speed of the motor can be controlled depending on the ratio of 1 to 0.) |

(2) Motor Driver Control Signals

# List of examples

## ■ Using Linux command line

- Control duty cycle and period

## ■ Using C compiler

- Control DC motor with kbhit() function
- Control Servo motor

## ■ Using the Arduino IDE

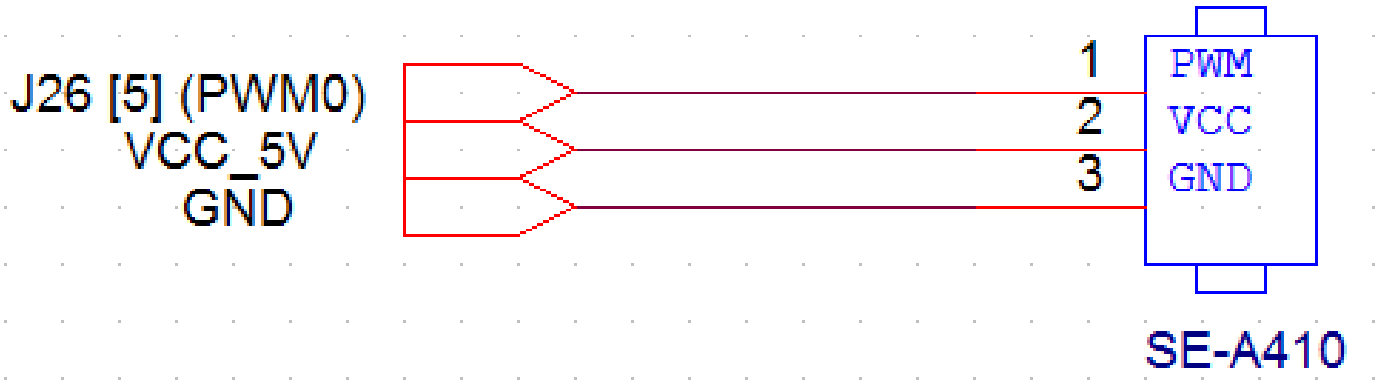
- Control PWM pin, making the LED pulse slowly on and off

# Using Linux Command Line

## ■ Required Hardware

- ARTIK 5 beta developer kit
- Servo Motor (SG90 or SE – A410)
- Breadboard
- Connector wires

## ■ Circuit Configuration



# Using Linux Command Line

## ■ Command

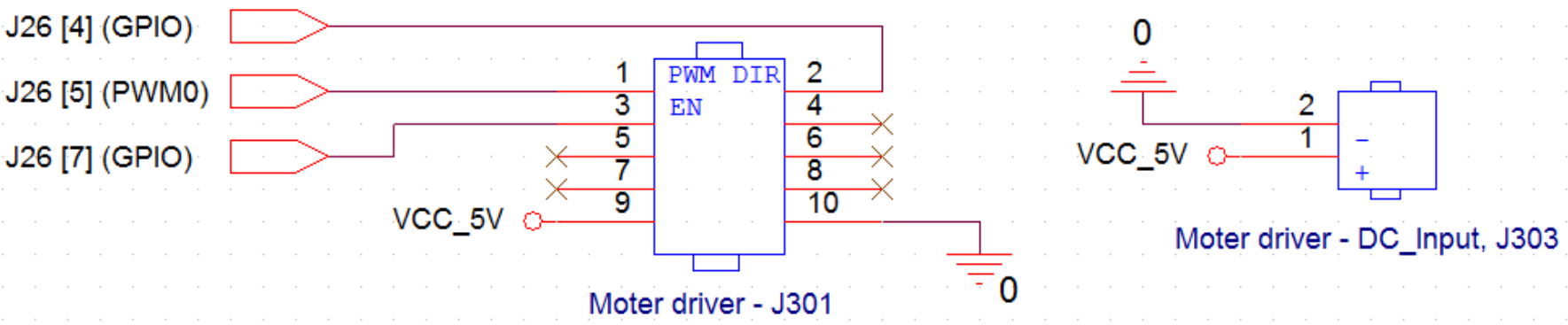
- Export PWM0
  - *# echo 0 > /sys/class/pwm/pwmchip0/export*
  - It will make a pwm0 subdirectory created.
- Set period to 20ms (Unit : ns)
  - *# echo 20000000 > /sys/class/pwm/pwmchip0/pwm0/period*
- Set duty cycle to 1.5ms
  - *# echo 1500000 > /sys/class/pwm/pwmchip0/pwm0/duty_cycle*
  - It will set the angle of rotation of servo motor to 0˚.
  - For SE-A410, duty_cycle = 1500000 + angle*10000
  - For SG90, duty_cycle = 1500000 + angle*8889
- Enable PWM0
  - *# echo 1 > /sys/class/pwm/pwmchip0/pwm0/enable*
- Disable PWM0
  - *# echo 0 > /sys/class/pwm/pwmchip0/pwm0/disable*
- Unexport PWM0
  - *# echo 0 > /sys/class/pwm/pwmchip0/unexport*

# C Example (1) - DC Motor

## ■ Required Hardware

- ARTIK 5 beta developer kit
- DC Motor (KR - 250)
- Motor driver(AM-DC2-2D)
- Breadboard
- Connector wires

## ■ Circuit Configuration



# C Example (1) - DC Motor

## ■ PWM function

- bool pwmPin(int pin, int period, int duty_cycle, int val)
  - int pin : PWM Pin setting(XPWMO0, XPWMO1)
  - int period : PWM period setting
  - int duty_cycle : PWM duty_cycle setting
  - int val : PWM enable, disable setting

## ■ PWM Code(1)

```
//period, duty_cycle -> nsec 기준
bool pwmPin(int pin, int period, int duty_cycle, int val){
    FILE *fd;
    char fName[128];

    if ((fd = fopen("/sys/class/pwam/pwmchip0/export", "w")) == NULL) {
        printf("Error: unable to export pin₩n");
        return false;
    }
    fprintf(fd, "%d₩n", pin);
    fclose(fd);
```

pin

```
    sprintf(fName, "/sys/class/pwm/pwmchip0/pwm%d/period", pin);
    if ((fd = fopen(fName, "w")) == NULL) {
        printf("Error: can't open pin direction₩n");
        return false;
    }
    fprintf(fd, "%d₩n", period);
    fclose(fd);
```

period

# C Example (1) - DC Motor

## ■ PWM Code(2)

```
sprintf(fName, "/sys/class/pwm/pwmchip0/pwm%d/duty_cycle", pin);
if((fd = fopen(fName, "w")) == NULL){
    printf("Error: can't open pin direction₩n");
    return false;
}
fprintf(fd, "%d₩n", duty_cycle);
fclose(fd);
```

duty_cycle

```
if(val == HIGH){
    sprintf(fName, "/sys/class/pwm/pwmchip0/pwm%d/enable", prn);

    if((fd = fopen(fName, "w")) == NULL){
        printf("Error: can't open pin value₩n");
        return false;
    }
    fprintf(fd, "1₩n");

else{
    sprintf(fName, "/sys/class/pwm/pwmchip0/pwm%d/enable", pin);

    if((fd = fopen(fName, "w")) == NULL){
        printf("Error: can't open pin value₩n");
        return false;
    }
    fprintf(fd, "0₩n");

}

fclose(fd);

return ture;
```

val

# C Example (1) - DC Motor

## ■ Keyboard hit function

- int kbhit()
  - Method of inputing by keyboard
  - Visual studio : O, Linux : X
  - Mainly used by the user to create a header file

```
#include <stdio.h>
#include <termios.h>
#include <unistd.h>
#include <fcntl.h>

int kbhit(void) {
    struct termios oldt, newt;
    int ch;
    int oldf;

    tcgetattr(STDIN_FILENO, &oldt);
    newt = oldt;
    newt.c_lflag &= ~(ICANON | ECHO);
    tcsetattr(STDIN_FILENO, TCSANOW, &newt);
    oldf = fcntl(STDIN_FILENO, F_GETFL, 0);
    fcntl(STDIN_FILENO, F_SETFL, oldf | O_NONBLOCK);

    ch = getchar();

    tcsetattr(STDIN_FILENO, TCSANOW, &oldt);
    fcntl(STDIN_FILENO, F_SETFL, oldf);

    if(ch != EOF) {
        ungetc(ch, stdin);
        return 1;
    }

    return 0;
}
```

(1) kbhit function

# C Example (1) - DC Motor

## Source Code(1)

```
#include "wiringARTIK.h"
#include "kbhit.h"

#define DIRECTION_PIN   123
#define ENABLE_PIN      124

//===== pin number ======
#define PWM_PIN         0
//=======================

#define PWM_ENABLE      1
#define PWM_DISABLE     0

#define INC_DEC         100000  // increase or decrease
#define PERIOD          1000000

static int index=0;

int setup() {
    digitalPinMode(DIRECTION_PIN, OUTPUT);
    digitalPinMode(ENABLE_PIN, OUTPUT);

    digitalWrite(ENABLE_PIN, LOW);
}
```

```
void print_monitor(int duty_value) {
    int duty_ratio = 0;

    duty_ratio = (duty_value * 100) / PERIOD;

    //printf("duty_value : %d₩n", duty_value);
    //printf("duty_ratio : %d₩n", duty_ratio);

    printf("====== PWM Test =======₩n");
    printf("Please keyboard hit!!!!₩n₩n");

    if((index == 0) || (index == 1)) {
        printf("F : Forward Direction₩n");
        printf("R : Reverse Direction₩n");
        printf("Q : Exit₩n");
    }

    else {
        printf("E : PWM Enable  & init₩n");
        printf("Q : PWM Disable & exit₩n");
        printf("I : PWM Increase₩n");
        printf("D : PWM Decrease₩n₩n");
        printf("Duty Ratio (now) : %d%₩n", duty_ratio);
    }

    printf("=======================₩n");
}
```

# C Example (1) - DC Motor

## Source Code(2)

```
int main() {
    int loop=1;
    int ch=0;
    int pin_dir=0;
    int duty_cycle = 500000;
    int direction = 2;

    setup();
    print_monitor(0);

while(loop) {
        if(index == 0) {
          if(kbhit()) {
              pin_dir = getchar();
              index = 1;
           }
        }
        if(index == 1) {
          switch (pin_dir) {
              case 'f' :  // f : forward direction
                  direction = 1;
                  index = 2;
                  print_monitor(0);
                  break;

               case 'r' :  // r : reverse direction
                  direction = 0;
                  index = 2;
                  print_monitor(0);
                  break;
```

```
        case 'q' :  // q : exit
        loop = 0;
        break;
    }
    if(!pin_dir)    print_monitor(0)
}

if(index == 2) {
    if(kbhit())     ch = getchar();

    switch (ch) {
        case 'e' :  // e : enable & init
              duty_cycle = 500000;

              digitalWrite(DIRECTION_PIN, direction);
              pwmPin(PWM_PIN, PERIOD, duty_cycle,
          PWM_ENABLE);
              printf("Pressed '%c'!₩n", ch);
              printf("PWM Enable₩n");

            index = 2;
            print_monitor(duty_cycle);
            break;

       case 'i' :  // i : inc
            duty_cycle += INC_DEC;  // Max = 1000000

            if(duty_cycle > PERIOD) duty_cycle = PERIOD;

            digitalWrite(DIRECTION_PIN, direction);
            pwmPin(PWM_PIN, PERIOD, duty_cycle, PWM_ENABLE);
            printf("Pressed '%c'!₩n", ch);
            printf("PWM Increase₩n");

            index = 2;
            print_monitor(duty_cycle);
            break;
```

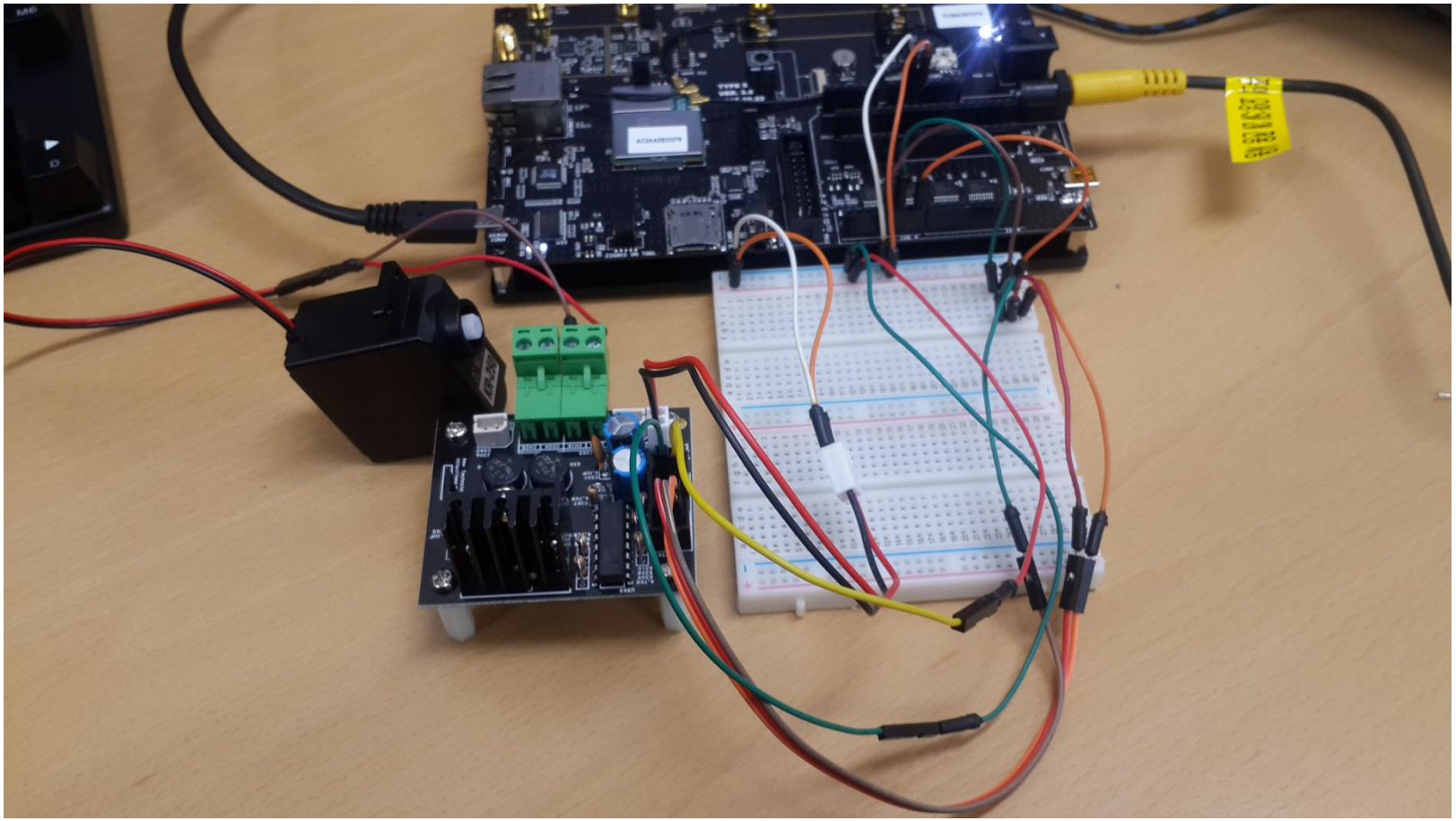
# C Example (1) - DC Motor

## ■Source Code(3)

```
            case 'd' :  // d : dec
                duty_cycle -= INC_DEC;  // Min = 200000

                if(duty_cycle < (PERIOD / 5))   duty_cycle = (PERIOD / 5);

                digitalWrite(DIRECTION_PIN, direction);
                pwmPin(PWM_PIN, PERIOD, duty_cycle, PWM_ENABLE);
                printf("Pressed '%c'!₩n", ch);
                printf("PWM Decrease₩n");

                index = 2;
                print_monitor(duty_cycle);
                break;

                case 'q' :  // q : disable & exit
                    pwmPin(PWM_PIN, PERIOD, 0, PWM_DISABLE);
                    printf("Pressed '%c'!₩n", ch);
                    printf("PWM Disable & Exit₩n");

                    index = 0;
                    print_monitor(0);
                    break;
            }
            ch = 0;
        }
    }

    return 0;
}
```

# C Example (1) - DC Motor

## ■ Execution result

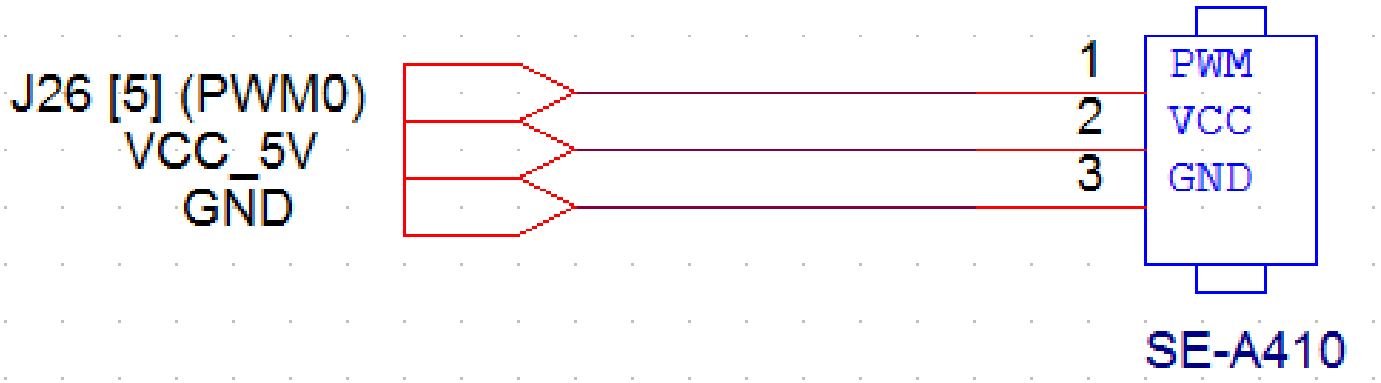


# C Example (2) - Servo Motor

## ■ Required Hardware

- ARTIK 5 beta developer kit
- Servo Motor (SG90 or SE – A410)
- Breadboard
- Connector wires

## ■ Circuit Configuration



# C Example (2) - Servo Motor

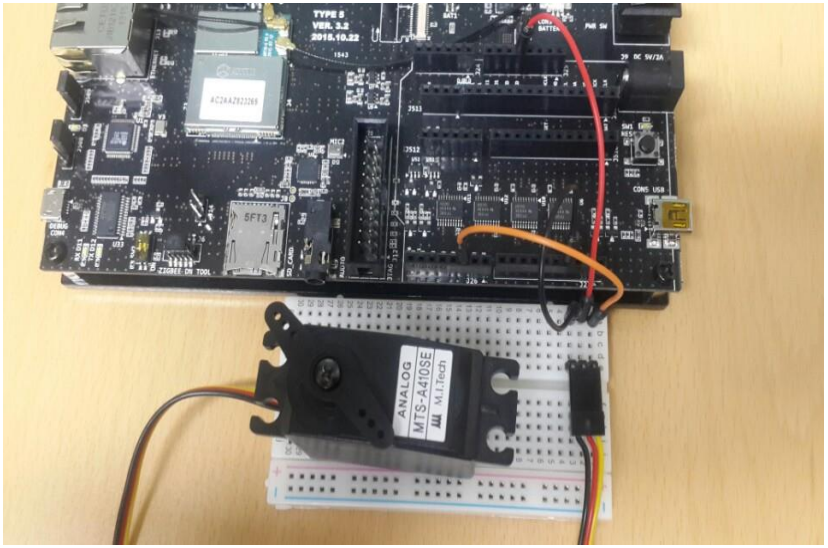
## Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <unistd.h>
#include "wiringARTIK.h"

#define EN 1
#define DIS 0
#define PERIOD 20000000 //nsec

bool ServoAngle(int pwm_pin, int angle)
{
	int duty_cycle;
	duty_cycle = 1500000 + angle * 10000;
	// SG90의 경우 1500000 + angle * 8889;
	pwmPin(pwm_pin, PERIOD, duty_cycle, EN);
}
```

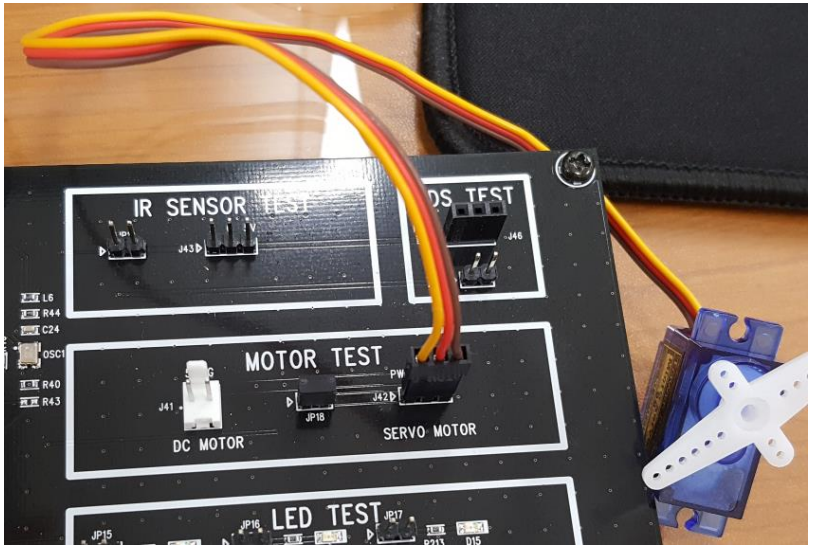
```
int main(void)
{
	while(1)
	{
		ServoAngle(0,90);
		sleep(1);

		ServoAngle(0,45);
		sleep(1);

		ServoAngle(0,0);
		sleep(1);

		ServoAngle(0,-45);
		sleep(1);

		ServoAngle(0,-90);
		sleep(1);
	}
	return 0;
}
```

# C Example (2) - Servo Motor

## ■ Execution result




ARTIK 520



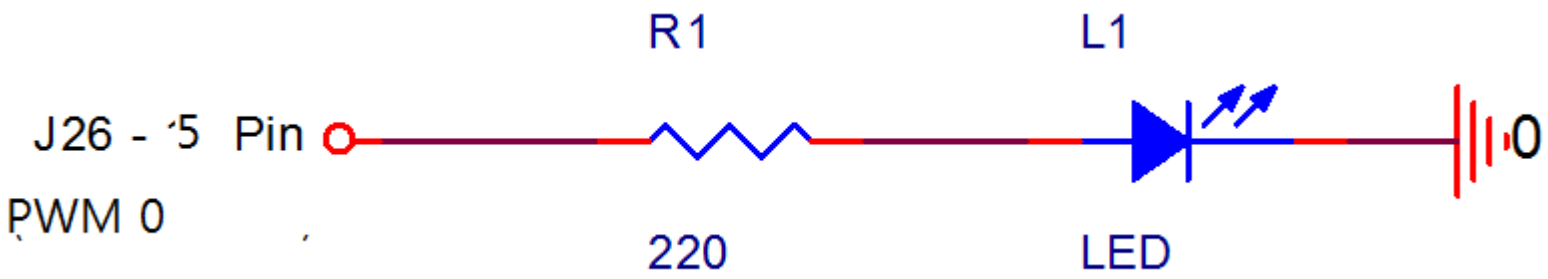

PS-ED500

# Using the Arduino IDE

## ■ Required Hardware

- ARTIK 5 beta developer kit
- DIP type LED
- 220 ohm resistor
- Breadboard
- Connector wires

## ■ Circuit Configuration



# Using the Arduino IDE

## ■ PWM Pin

| Header Pin | Native Mapping ARTIK 520 v0.5 | Arduino Pin |
|---|---|---|
| J26[5] | PWM1 | 5 |
| J26[6] | PWM0 | 6 |

## ■ A

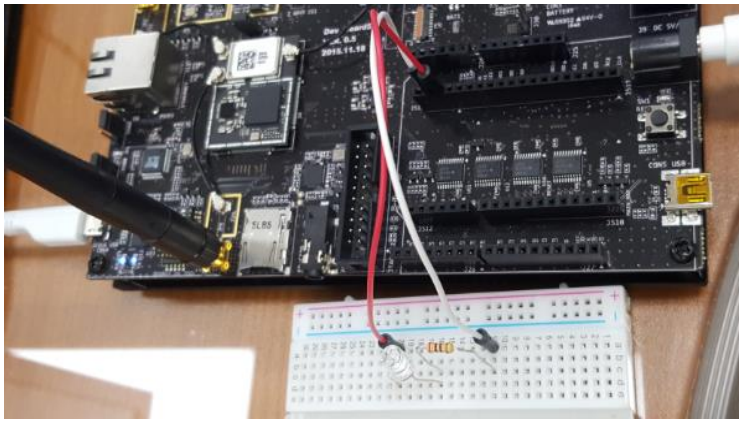
LED_PWM

```
const int thisPin = 5;

void setup() {
}

void loop() {
  for (int brightness = 0; brightness < 255; brightness++) {
    analogWrite(thisPin, brightness);
    delay(2);
  }
  for (int brightness = 255; brightness >=0; brightness--) {
    analogWrite(thisPin, brightness);
    delay(2);
  }

  delay(100);
}
```

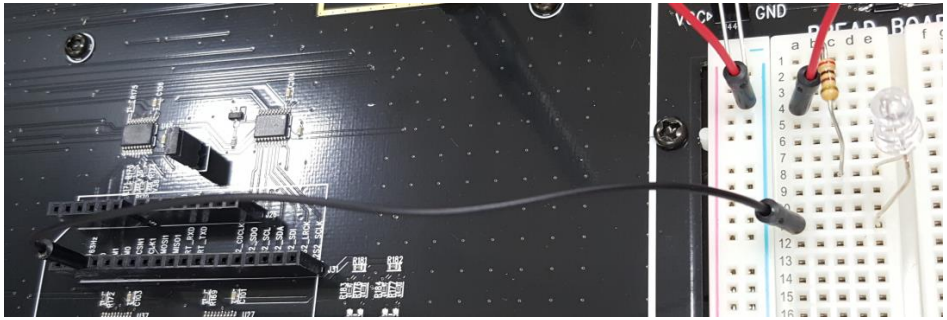
# Using the Arduino IDE

## ■ Execution result

- Enter [Ctrl]+[u] or use menu to upload the compiled program to ARTIK.
- After uploading, you should see LED slowly on and off.



ARTIK 520




PS-ED500