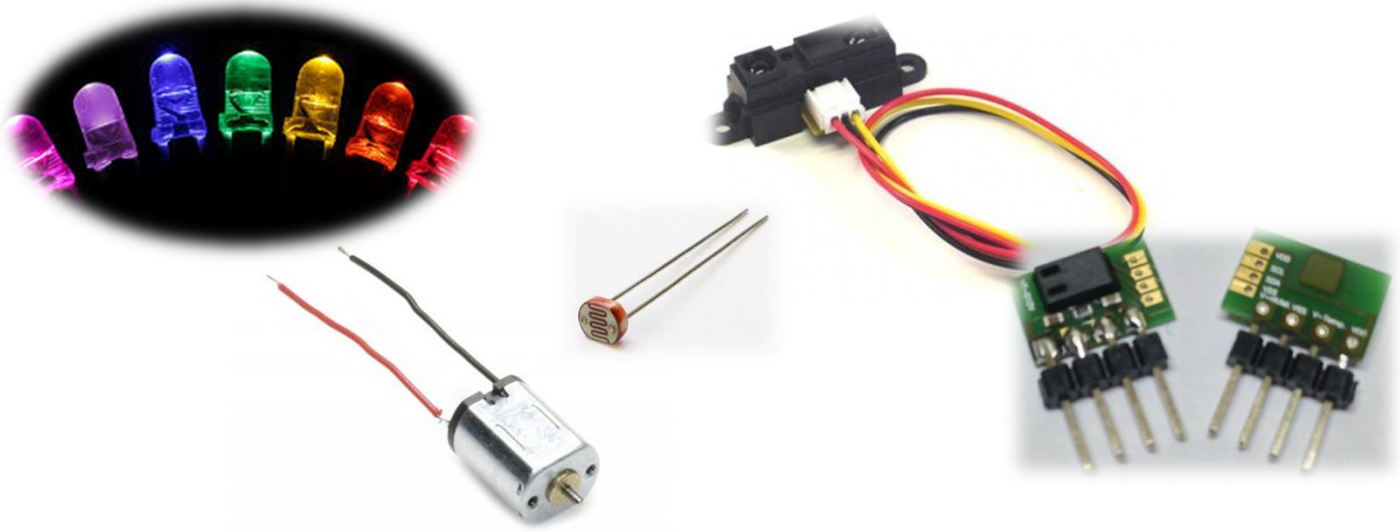
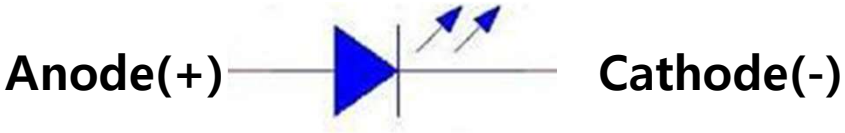
2주차 강의자료

1. GPIO Control



LED(Light-Emitting Diode)

■ Symbol of LED



■ Type of LED

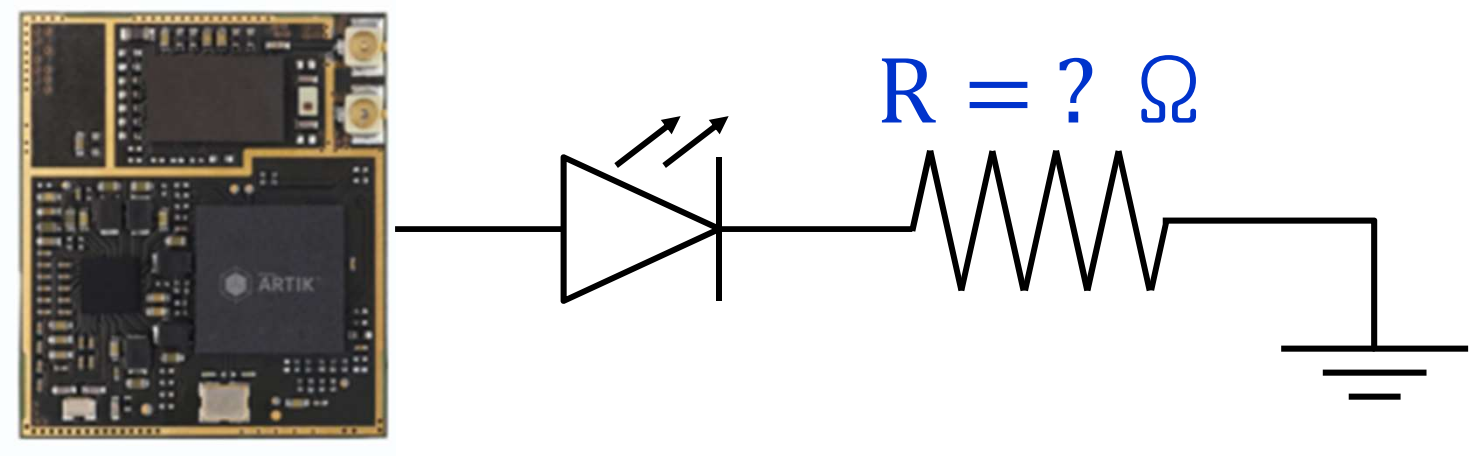
- DIP(Duel In-Line Package) Type
- SMD(Surface Mount Device/Diode) Type

[DIP type]

[SMD type]

LED(Light-Emitting Diode)

■ How to calculate LED connection resistance



LED(1.7V, 10mA 기준)

$$R = \frac{V_{supply} - V_{LED}}{I_{LED}} = \frac{3.3V - 1.7V}{10mA} = \mathbf{160\Omega}$$

List of examples

Using Linux command line

- Control LED on/off

Using C compiler

- Control LED on/off
- Control LED with switch

Using the Arduino IDE

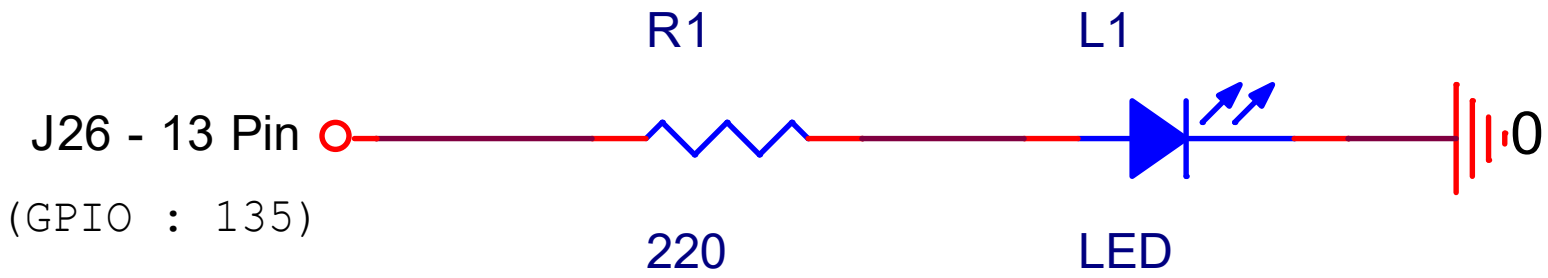
- Control LED on/off

Using Linux Command Line

Required Hardware

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

Circuit Configuration



Using Linux Command Line

Command

- Request control of the desired GPIO pin.
  - *# echo 135 > /sys/class/gpio/export*
  - This command makes a GPIO-specific directory created.
- Configure the GPIO pin to be an output.
  - *# echo out > /sys/class/gpio/gpio135/direction*
- Set the output level by writing the value file contents to "0" or "1".
  - *# echo 1 > /sys/class/gpio/gpio135/value*
  - *# echo 0 > /sys/class/gpio/gpio135/value*
- Unexport GPIO pin
  - *# echo 135 > /sys/class/gpio/unexport*

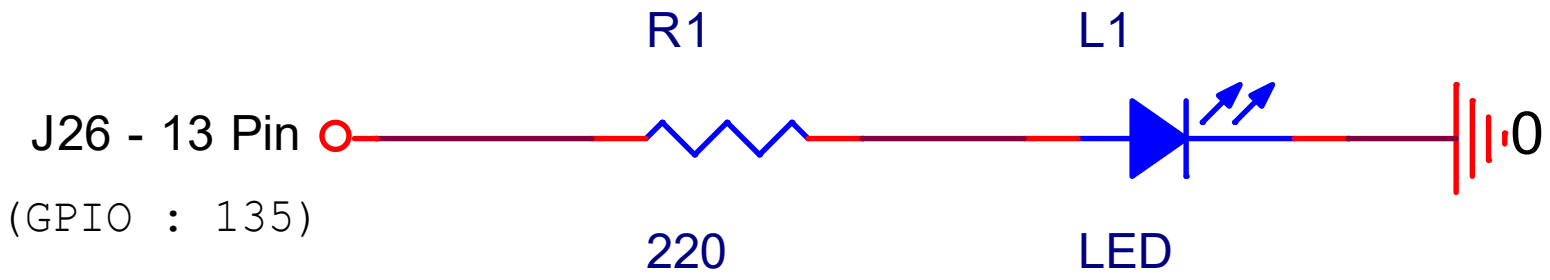
```
[root@localhost ~]# echo 135 > /sys/class/gpio/export
[root@localhost ~]# echo out > /sys/class/gpio/gpio135/direction
[root@localhost ~]# echo 1 > /sys/class/gpio/gpio135/value
[root@localhost ~]# echo 0 > /sys/class/gpio/gpio135/value
[root@localhost ~]# echo 135 > /sys/class/gpio/unexport
[root@localhost ~]#
```

C Compiler Example (1)

Required Hardware

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

Circuit Configuration



C Compiler Example (1)

Source Code

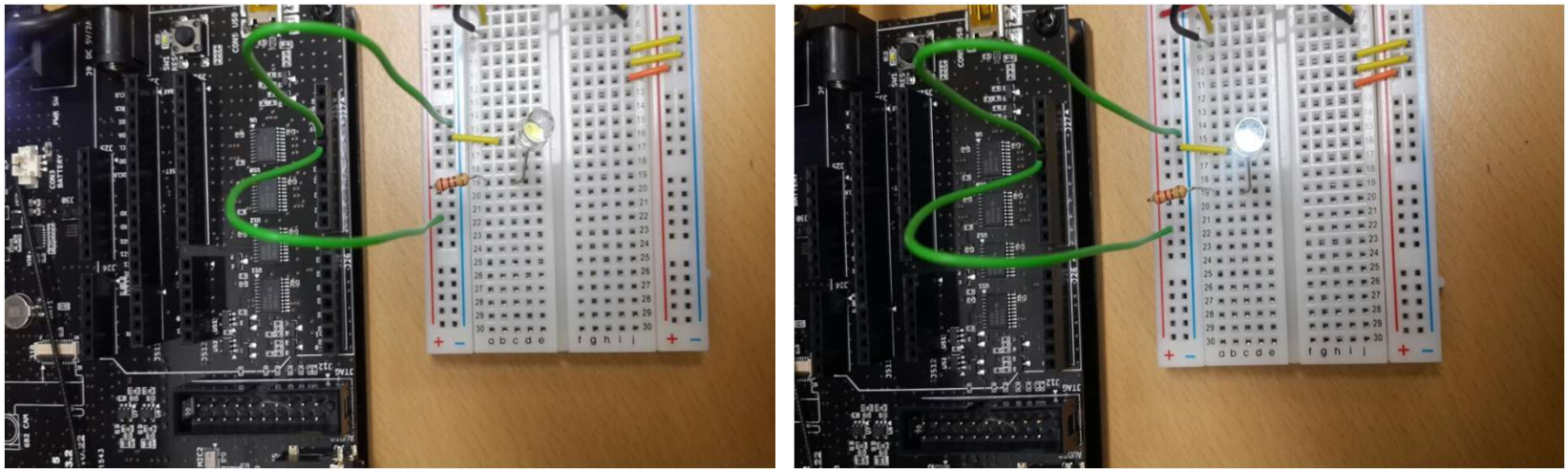
```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

#define HIGH 1
#define LOW 0
#define INPUT 1
#define OUTPUT 0

int outputPin = 121;

bool digitalPinMode(int pin, int dir)
{
   FILE * fd;
   char fName[128];

   // Exporting the pin to be used
   if(( fd = fopen("/sys/class/gpio/export", "w")) == NULL){
      printf("Error: unable to export pin₩n");
      return false;
   }
   fprintf(fd, "%d₩n", pin);
   fclose(fd);

   // Setting direction of the pin
   sprintf(fName, "/sys/class/gpio/gpio%d/direction", pin);
   if((fd = fopen(fName, "w")) == NULL){
      printf("Error: can't open pin direction₩n");
      return false;
   }
   if(dir == OUTPUT) { fprintf(fd, "out₩n"); }
   else {fprintf(fd, "in₩n"); }

   fclose(fd);
   return true;
}
```

```
bool digitalWrite(int pin, int val)
{
   FILE * fd;
   char fName[128];

   // Open pin value file
   sprintf(fName, "/sys/class/gpio/gpio%d/value", pin);
   if((fd = fopen(fName, "w")) == NULL)
   {
      printf("Error: can't open pin value₩n");
      return false;
   }
   if(val == HIGH) {fprintf(fd, "1₩n"); }
   else {fprintf(fd, "0₩n"); }

   fclose(fd);
   return true;
}

int setup()
{
   if (!digitalPinMode(outputPin, OUTPUT)) return -1;

   return 0;
}

int main(void)
{
   if (setup() == -1) {exit(1);}

   while(1)
   {
      digitalWrite(outputPin, HIGH);
      sleep(1);
      digitalWrite(outputPin, LOW);
      sleep(1);
   }

   return 0;
}
```

C Compiler Example (1)

Execution result

- Before executing code : LED off
- After executing code : LED repeats on / off at 1 second intervals



ARTIK 520

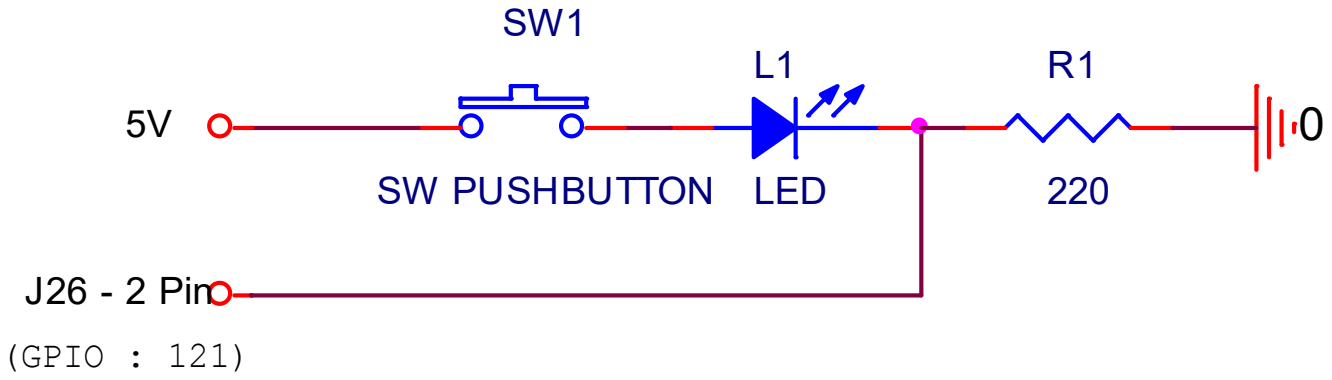
PS-ED500

C Compiler Example (2)

Required Hardware

- ARTIK 5 beta developer kit
- LED
- 220 ohm resistor
- Momentary button or Switch
- Breadboard
- Connector wires

Circuit Configuration



C Compiler Example (2)

Source Code

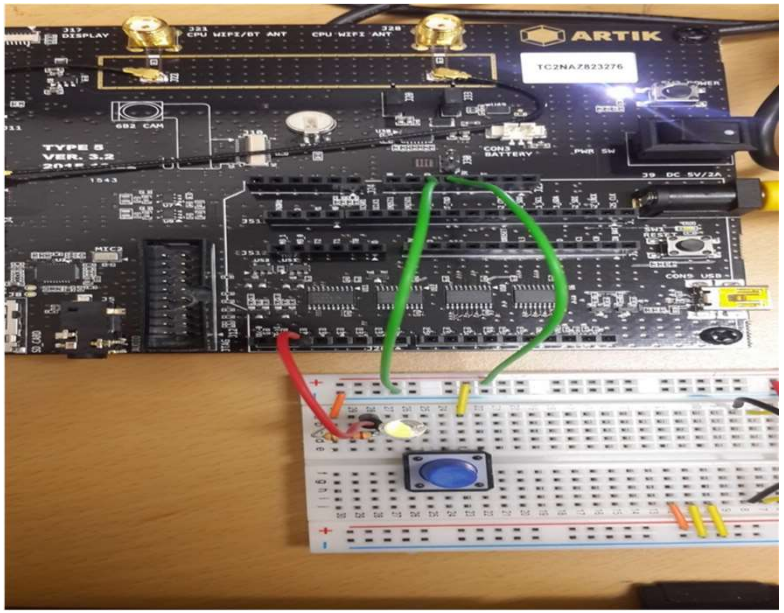
```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

#define HIGH 1
#define LOW 0
#define INPUT 1
#define OUTPUT 0

int inputPin = 121;

bool digitalPinMode(int pin, int dir)
{
   FILE * fd;
   char fName[128];

   // Exporting the pin to be used
   if(( fd = fopen("/sys/class/gpio/export", "w")) == NULL){
      printf("Error: unable to export pin₩n");
      return false;
   }
   fprintf(fd, "%d₩n", pin);
   fclose(fd);

   // Setting direction of the pin
   sprintf(fName, "/sys/class/gpio/gpio%d/direction", pin);
   if((fd = fopen(fName, "w")) == NULL){
      printf("Error: can't open pin direction₩n");
      return false;
   }
   if(dir == OUTPUT) {fprintf(fd, "out₩n");}
   else {fprintf(fd, "in₩n"); }

   fclose(fd);
   return true;
}
```

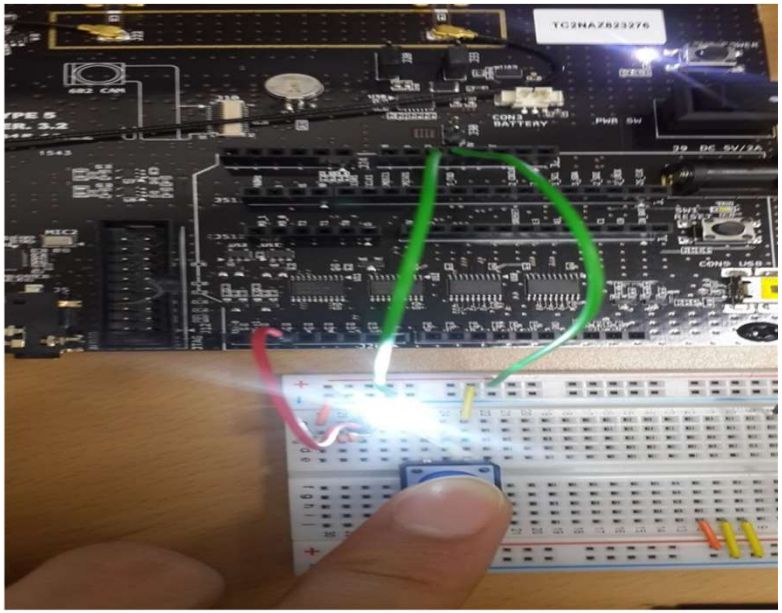
```
int digitalRead(int pin)
{
   FILE * fd;
   char fName[128];
   char val[2];

   // Open pin value file
   sprintf(fName, "/sys/class/gpio/gpio%d/value", pin);
   if((fd = fopen(fName, "r")) == NULL)
   {
      printf("Error: can't open pin value₩n");
      return false;
   }
   fgets(val, 2, fd);
   fclose(fd);

   return atoi(val);
}

int setup()
{
   if (!digitalPinMode(inputPin, INPUT)) return -1;

   return 0;
}

int main(void)
{
   if (setup() == -1) { exit(1); }

   while(1)
   {
      int sensorVal;
      sensorVal = digitalRead(inputPin);
      printf("sensorVal is %d₩n", sensorVal);
      sleep(1);
   }

   return 0;
}
```

C Compiler Example (2)

Execution result

- Before executing code : LED off
- After executing code
  - Switch is pressed – LED on, sensorVal = 1
  - Switch is released– LED off, sensorVal = 0





ARTIK 520

Using the Arduino IDE

GPIO Pin Number

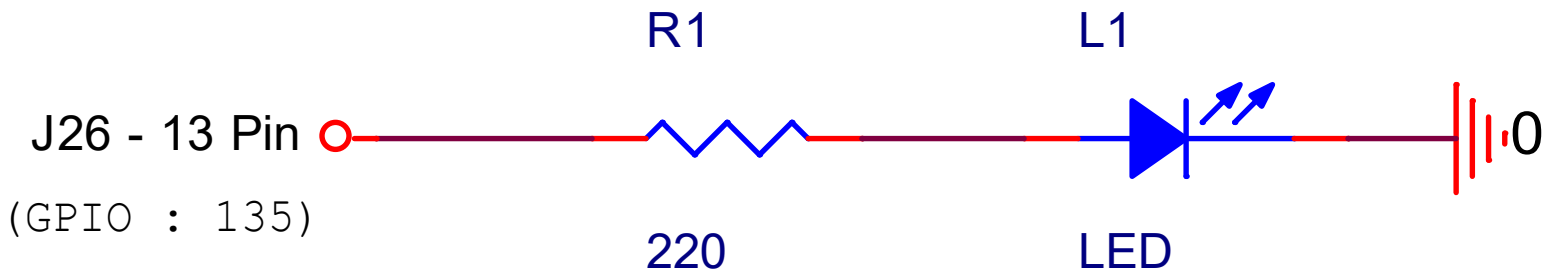
External Pin	A520-GPIO	Arduino Pin
J26[2]	121	2
J26[3]	122	3
J26[4]	123	4
J26[7]	124	7
J27[8]	125	8
J27[9]	126	9
J27[10]	127	10
J27[11]	129	11
J27[12]	134	12
J27[13]	135	13

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

■ Arduino Source Code

LED_GPIO

```
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

■ Execute result

- Enter [Ctrl]+[u] or using menu, compile and upload your code to ARTIK.
- Then, you should see LED blinking.