

부록 1. 아두이노 기본 함수

I. 디지털 입출력 함수

1. pinMode

```
void pinMode(uint8_t pin, uint8_t mode)
    pin : 설정하고자 하는 핀 번호
    mode : INPUT, OUTPUT, INPUT_PULLUP 중 하나
    반환값 : 없음
```

지정한 번호의 핀이 입력 또는 출력으로 동작하도록 설정한다. mode가 INPUT_PULLUP으로 지정되면 내부 풀업 저항을 사용하는 입력으로 설정한다.

2. digitalWrite

```
void digitalWrite(uint8_t pin, uint8_t value)
    pin : 핀 번호
    value : HIGH 또는 LOW
    반환값 : 없음
```

지정한 번호의 핀으로 HIGH 또는 LOW의 디지털 값을 출력한다. 핀이 입력 상태로 설정된 경우 digitalWrite 함수로 HIGH 값을 출력하면 20 k Ω 내부 풀업 저항이 연결되며 LOW 값을 출력하면 풀업 저항 연결이 해제된다.

3. digitalRead

```
int digitalRead(uint8_t pin)
    pin : 핀 번호
    반환값 : HIGH 또는 LOW
```

지정한 번호의 핀으로부터 HIGH 또는 LOW의 디지털 값을 읽어서 반환한다.

II. 아날로그 입출력 함수

1. analogReference

```
void analogReference(uint8_t type)
    type : DEFAULT, INTERNAL, INTERNAL1V1, INTERNAL2V56, EXTERNAL 중 한
           가지
    반환값 : 없음
```

아날로그 입력을 위한 기준 전압을 설정한다. 아두이노 우노에 사용되는 ATmega328의 경우 5가지 옵션 중 다음 세 가지를 사용할 수 있다.

- ◆ DEFAULT : 아두이노 보드의 동작 전압인 5V를 기준 전압으로 설정한다.
- ◆ INTERNAL : 1.1V의 내부 전압을 기준 전압으로 설정한다.
- ◆ EXTERNAL : AREF 핀에 인가된 0V ~ 5V 사이의 전압을 기준 전압으로 설정한다.

2. analogRead

```
int analogRead(uint8_t pin)
    pin : 핀 번호 (A0 ~ A5)
    반환값 : 0에서 1023 사이의 정수값
```

지정한 번호의 아날로그 입력 핀으로부터 아날로그 값을 읽고 ADC를 통해 0에서 1023까지의 디지털 값으로 변환하여 반환한다.

3. analogWrite

```
void analogWrite(uint8_t pin, int value)
    pin : 핀 번호
    value : 듀티 사이클 (duty cycle). 0(항상 off)에서 255(항상 on) 사이의 값
    반환값 : 없음
```

지정한 번호의 아날로그 핀으로 지정한 듀티 사이클을 가지는 PWM (Pulse Width Modulation, 펄스 폭 변조) 신호를 출력한다.

III. 고급 입출력 함수

1. tone

```
void tone(uint8_t pin, unsigned int frequency, unsigned long duration = 0)
```

pin : 핀 번호

frequency : 출력 주파수 (unsigned int)

duration : 출력 지속시간, 밀리초 단위 (unsigned long)

반환값 : 없음

지정한 번호의 핀으로 50% 듀티 사이클과 지정된 주파수를 가지는 구형파(square wave)를 지정된 시간 동안 출력하여 단음을 재생한다. 지속시간이 지정되지 않으면 noTone 함수가 호출될 때까지 출력이 계속된다.

2. noTone

```
void noTone(uint8_t pin)
```

pin : 핀 번호

반환값 : 없음

지정한 번호의 핀에 tone 함수에 의한 구형파 출력을 정지한다.

3. shiftOut

```
void shiftOut(uint8_t dataPin, uint8_t clockPin, uint8_t bitOrder, uint8_t value)
```

dataPin : 비트 출력이 일어날 핀

clockPin : 한 비트 출력 후 데이터 출력을 알려주기 위해 펄스가 출력되는 핀

bitOrder : 비트 출력 순서, MSBFIRST 또는 LSBFIRST

value : 출력될 데이터 (byte)

반환값 : 없음

지정한 번호의 데이터 핀(dataPin)으로 지정한 값(value)을 출력 순서(bitOrder)에 따라 출력한다. 출력 순서는 MSB(Most Significant Bit)나 LSB(Least Significant Bit)부터 출력하도록 지정할 수 있으며, 하나의 비트가 출력된 이후에는 clockPin으로 펄스가 출력되어 데이터 동기화에 사용된다.

4. shiftIn

```
uint8_t shiftIn(uint8_t dataPin, uint8_t clockPin, uint8_t bitOrder)
```

dataPin : 비트 입력을 받아들이는 핀

clockPin : 비트 입력을 위한 클록 핀

bitOrder : 비트 입력 순서, MSBFIRST 또는 LSBFIRST

반환값 : 입력 값 (byte)

지정한 번호의 핀(dataPin)으로부터 데이터를 입력받아 비트 입력 순서(bitOrder)에 따라 정렬한 후 바이트 단위로 반환한다. clockPin은 비트 단위의 데이터 전송을 알려주는 핀으로 데이터 동기화를 위해 사용된다.

5. pulseIn

```
unsigned long pulseIn(uint8_t pin, uint8_t value,  
                      unsigned long timeout = 1000000L)
```

pin : 펄스를 읽어 들일 핀 번호

value : 읽어 들일 펄스의 종류, HIGH 또는 LOW

timeout : 펄스의 시작을 기다리는 시간, 마이크로초 단위이며 디폴트 값은 1초

반환값 : 펄스의 길이를 마이크로초 단위로 반환하며 타임아웃 이전에 펄스가 시작되지 않으면 0을 반환

지정한 번호의 핀으로부터 펄스를 읽어 그 길이를 마이크로초 단위로 반환한다. 읽어 들이는 펄스의 종류는 value에 의해 지정할 수 있으며, 펄스 대기 시간을 타임아웃 값으로 지정할 수 있다.

IV. 시간 함수

1. millis

```
unsigned long millis(void)
```

반환값 : 프로그램이 시작된 이후의 밀리초(millisecond) 단위 경과 시간

현재 실행 중인 프로그램이 시작된 이후의 경과 시간을 밀리초 단위로 반환한다. 반환값은 4바이트 크기이므로 약 50일의 시간 경과 후 오버플로우(overflow)가 발생한다.

2. micros

```
unsigned long micros(void)
```

반환값 : 프로그램이 시작된 이후의 마이크로초(microsecond) 단위 경과 시간

현재 실행 중인 프로그램이 시작된 이후의 경과 시간을 마이크로초 단위로 반환한다. 반환값은 4바이트 크기이므로 약 70분의 시간 경과 후 오버플로우(overflow)가 발생한다.

3. delay

```
void delay(unsigned long ms)
```

ms : 밀리초 단위의 지연 시간

반환값 : 없음

지정한 밀리초 단위의 시간만큼 프로그램 실행을 일시 중지한다.

4. delayMicroseconds

```
void delayMicroseconds(unsigned long us)
```

us : 마이크로초 단위의 지연 시간 (unsigned int)

반환값 : 없음

지정한 마이크로초 단위의 시간만큼 프로그램 실행을 일시 중지한다.

V. 수학 함수

1. min

```
min(x, y)
```

x : 첫 번째 데이터

y : 두 번째 데이터

반환값 : x, y 중 작은 값을 반환

```
#define min(a, b) ((a) < (b) ? (a) : (b))
```

두 숫자의 최솟값을 반환한다.

2. max

<pre>max(x, y)</pre> <p> x : 첫 번째 데이터 y : 두 번째 데이터 반환값 : x, y 중 큰 값을 반환 </p>
<pre>#define max(a, b) ((a) > (b) ? (a) : (b))</pre>

두 숫자의 최댓값을 반환한다.

3. abs

<pre>abs(x)</pre> <p> x : 데이터 반환값 : x의 절대값을 반환 </p>
<pre>#define abs(x) ((x) > 0 ? (x) : -(x))</pre>

주어진 숫자의 절대값을 반환한다.

4. constrain

<pre>constrain(x, a, b)</pre> <p> x : 데이터 a : x가 가질 수 있는 최솟값 b : x가 가질 수 있는 최댓값 반환값 : x가 범위 내의 값이면 x를, a보다 작은 경우 a를, b보다 큰 경우 b를 반환 </p>
<pre>#define constrain(amt, low, high) ((amt) < (low) ? (low) : ((amt) > (high) ? (high) : (amt)))</pre>

주어진 값이 지정한 범위 내의 값인 경우 그대로 반환하고 범위 밖의 값인 경우 최솟값이나 최댓값을 반환한다.

5. map

```
long map(long value, long fromLow, long fromHigh, long toLow, long toHigh)
```

value : 데이터

fromLow : 현재 데이터가 가질 수 있는 최소값

fromHigh : 현재 데이터가 가질 수 있는 최대값

toLow : 변환하고자 하는 범위의 최소값

toHigh : 변환하고자 하는 범위의 최대값

반환값 : 지정한 범위로 사상된 값

주어진 데이터 값을 지정된 범위의 값으로 선형 사상(linear mapping)하여 반환한다. 이 때 주어지는 값은 [fromLow fromHigh] 범위의 값이 아니어도 무방하다.

6. pow

```
double pow(double base, double exponent)
```

base : 밑

exponent : 지수

반환값 : 거듭제곱 값

주어진 밑과 지수로 거듭제곱 값을 계산하여 반환한다.

7. sqrt

```
double sqrt(double x)
```

x : 데이터

반환값 : x의 제곱근

주어진 값의 제곱근을 계산하여 반환한다.

VI. 삼각 함수

- sin

```
double sin(double rad)
    rad : 라디안 형식의 각도
    반환값 : 사인 값
```

라디안(radian) 형식으로 주어진 각도에 대한 사인 값을 반환한다.

2. cos

```
double cos(double rad)
    rad : 라디안 형식의 각도
    반환값 : 코사인 값
```

라디안(radian) 형식으로 주어진 각도에 대한 코사인 값을 반환한다.

3. tan

```
double tan(double rad)
    rad : 라디안 형식의 각도
    반환값 : 탄젠트 값
```

라디안(radian) 형식으로 주어진 각도에 대한 탄젠트 값을 반환한다.

VII. 난수 함수

1. randomSeed

```
void randomSeed(unsigned int seed)
    seed : 의사 난수의 시작 위치 결정을 위한 값
    반환값 : 없음
```

의사 난수 생성기를 초기화한다. 의사 난수 생성에서 중요한 값인 시드(seed)는 외부 회로가 연결되지 않은 아날로그 핀으로부터 무작위 값을 입력받아 사용하는 것이 일반적이다.

2. random


```
long random(long max)
long random(long min, long max)
    min : 생성될 난수의 최소값
    max : max - 1이 생성될 난수의 최대값
    반환값 : [min, max - 1] 범위의 난수값
```

의사 난수를 생성하여 반환한다. 최솟값을 지정하지 않은 경우 최소값은 0으로 설정된다.

VIII. 비트 조작 함수

1. lowByte

```
lowByte(x)
    x : 데이터
    반환값 : x의 최하위 바이트를 반환 (byte)

#define lowByte(w) ((uint8_t) ((w) & 0xff))
```

주어진 값의 최하위 바이트를 반환한다.

2. highByte

```
highByte(x)
    x : 데이터
    반환값 : x의 두 번째 최하위 바이트를 반환 (byte)

#define highByte(w) ((uint8_t) ((w) >> 8))
```

주어진 값의 두 번째 최하위 바이트를 반환한다.

3. bitRead

<pre>bitRead(x, n) x : 데이터 n : 읽어낼 비트 위치로 LSB부터 0에서 시작 반환값 : 0또는 1의 비트값</pre>
<pre>#define bitRead(value, bit) (((value) >> (bit)) & 0x01)</pre>

주어진 값(x)의 n번째 비트값을 읽어 반환한다.

4. bitWrite

<pre>bitWrite(x, n, b) x : 데이터 n : 기록할 비트 위치로 LSB부터 0에서 시작 b : 0또는 1의 기록할 비트값 반환값 : 없음</pre>
<pre>#define bitWrite(value, bit, bitvalue) (bitvalue ? bitSet(value, bit) : bitClear(value, bit))</pre>

주어진 값(x)의 n번째 비트에 b값(0 또는 1)을 쓴다.

5. bitSet

<pre>bitSet(x, n) x : 데이터 n : 기록할 비트 위치로 LSB부터 0에서 시작 반환값 : 없음</pre>
<pre>#define bitSet(value, bit) ((value) = (1UL << (bit)))</pre>

주어진 값(x)의 n번째 비트를 1로 설정한다.

6. bitClear

bitClear(x, n) x : 데이터 n : 기록할 비트 위치로 LSB부터 0에서 시작 반환값 : 없음
<pre>#define bitClear(value, bit) ((value) &= ~(1UL << (bit)))</pre>

주어진 값(x)의 n번째 비트를 0으로 설정한다.

7. bit

bit(n) n : 계산하고자 하는 비트 위치로 LSB부터 0에서 시작 반환값 : n에 의해 지정된 비트의 비트값
<pre>#define bit(b) (1UL << (b))</pre>

지정된 위치(n)에 해당하는 비트의 비트값인 2^n 값을 반환한다.

IX. 외부 인터럽트 함수

- attachInterrupt

<pre>void attachInterrupt(uint8_t interrupt, void (*function)(void), int mode)</pre>
interrupt : 인터럽트 번호 function : 인터럽트를 처리할 인터럽트 서비스 루틴(interrupt service routine, ISR). ISR은 파라미터를 가지지 않으며 반환값이 없다. mode : 인터럽트가 발생하는 시점 반환값 : 없음

외부 인터럽트가 발생하였을 때 인터럽트를 처리할 인터럽트 서비스 루틴(interrupt service routine, ISR)을 지정한다. 아두이노 우노의 경우 디지털 2번과 3번 핀을 통해 인터럽트 0과 인터럽트 1의 두 개 외부 인터럽트를 사용할 수 있으며 보드 종류에 따라 차이가 있다. mode는 인터럽트가 발생하는 시점을 나타내며 LOW (입력값이 LOW일 때 인터럽트 발생), CHANGE (입력값의 상태가 변할 때 인터럽트 발생, RISING & FALLING), RISING (입력값이

LOW에서 HIGH로 변할 때 인터럽트 발생), FALLING (입력값이 HIGH에서 LOW로 변할 때 인터럽트 발생) 등을 사용할 수 있다.

2. detachInterrupt

<pre>void detachInterrupt(uint8_t interrupt) interrupt : 인터럽트 번호 반환값 : 없음</pre>

해당 인터럽트에 대한 인터럽트 처리 루틴을 제거한다.

X. 인터럽트 함수

1. interrupts

<pre>interrupts() 반환값 : 없음</pre>

<pre>#define interrupts() sei()</pre>

noInterrupts 함수에 의해 인터럽트 금지된 인터럽트 발생을 허용한다.

2. noInterrupts

<pre>noInterrupts() 반환값 : 없음</pre>
--

<pre>#define noInterrupts() cli()</pre>

interrupts 함수에 의해 허용된 인터럽트 발생을 금지한다.