**Lab5. USART 통신**

# **예제1. 터미널 입력을 그대로 출력 Serial.write() -> serial.print()**

|  |  |  |
| --- | --- | --- |
| **구분** | **설명** | **코드** |
| Serial.begin(rate)  Serial.print(val)  Serial.write(val)  Serial.available()  Serial.read() | 컴퓨터의 시리얼과 통신할때 통신 속도를 결정해주는 코드이다.  일반적으로 9600을 많이 사용한다.  시리얼 통신화면에 print내에 있는 값을 보여주는 역할을 한다. 숫자는 ascii코드로 바꿔서 전송하기 때문에 숫자를 print에 넣으면 자연스럽게 숫자로 변환되어서 나온다.  Print함수와 마찬가지로 시리얼 통신화면에 write 내에 있는 값을 보여준다. 그러나 byte단위로 보내기때문에, 숫자를 입력할 경우 ascii코드로 읽게 되어 잘못된 결과가 출력되기 때문에 주의가 필요하다.  시리얼 통신을 할때 데이터를 수신할 때 사용된다. 데이터를 수신받으면 버퍼에 저장되고, Serial.available 함수에 1이상의 값이 저장된다.  버퍼에 저장되어있는 값을 바이트단위로 읽어온다. | Void setup(){  Serial.begin(9600);  }  Serial.print(“Hello”);  Serial.print(100);  Serial.write(“Hello”);  Serial.write(45);  If(Serail.available()){  Serial.write(Serial.read());  } |

=========================================================================================

**1. 터미널 입력을 그대로 출력 Serial.write() -> serial.print()**

**소스 코드**

=========================================================================================

Void setup(){

Serial.begin(9600);

}

Void loop(){

If(Serial.available()){

char a = Serial.read();

Serial.print(a);

}

}

=========================================================================================

**예제2.터미널 입력을 받아 LED와 모터가 동작하는 프로그램 구성**

**소스 코드**

=========================================================================================

#include<Servo.h>

int ledPin = 13;

int servoPin = 5;

int Level = 0;

Servo servo;

void setup() {

Serial.begin(9600);

pinMode(ledPin, OUTPUT);

servo.attach(servoPin);

}

// the loop function runs over and over again forever

void loop() {

if (Serial.available())

{

char a = Serial.read();

Serial.print(a);

if(a=='1'){

Serial.print("LED on/off");

digitalWrite(ledPin, HIGH);

delay(1000);

digitalWrite(ledPin, LOW);

}

if(a=='2'){

Serial.print("Servo turn once");

for(Level = 0; Level<180 ;Level++){

servo.write(Level);

delay(10);

}

}

}

}



# **예제3. USART 채팅 프로그램 만들기**

|  |  |  |
| --- | --- | --- |
| **구분** | **설명** | **코드** |
| #include<SoftwareSerial.h>  SoftwareSerial mySerial(rxPin,txPin) | SoftwareSerial.h의 라이브러리를 추가해줍니다.  시리얼 포트를 추가적으로 설정할 수 있습니다.  Rxpin은 시리얼 데이터를 수신하는 핀  Txpin 은 시리얼 데이터를 전송하는 핀 | #include<SoftwareSerial.h>  SoftwareSerial mySerial(10,11);  Void setup(){  mySerial.begin(9600);  }  Void loop(){  mySerial.print(“Hello”);  } |

=========================================================================================

**3.USART 채팅 프로그램 만들기**

**소스 코드**

=========================================================================================

#include<SoftwareSerial.h>

#define LEDPIN 13

SoftwareSerial talk(10,11);

void setup(){

Serial.begin(9600);

talk.begin(9600);

pinMode(LEDPIN,OUTPUT);

}

void loop(){

while(talk.available(){ //신호 받음

Serial.write(talk.read());

digitalWrite(LEDPIN,HIGH);

}

delay(1000);

digitalWrite(LEDPIN,LOW);

while(Serial.available(){ //신호 보냄

talk.write(Serial.read());

digitalWrite(LEDPIN,HIGH);

}

delay(1000);

}

=========================================================================================



|  |
| --- |
| **고찰 및 실험 평가**  **이번 실험은 터미널 입력을 출력하는 코드 serial.write를 serial.print로 동일한 동작을 수행하는 실험과, 터미널 입력을 받아서 led와 모터가 동작하는 프로그램을 만드는 실험, softwareserail을 사용아여 usart채팅 프로그램을 만드는 실험, 총 3가지 실험을 하였다.**  **먼저, 첫번째 실험에서는 serial.write 함수와 serial.print 함수의 특성을 파악해야지 코드를 동일하게 만들수 있었다. write함수는 바이트 단위로 읽으며, 숫자가 들어올 경우 아스키 코드로 인식을 하게 되지만, print함수는 아스키 코드로 변환해서 전송하기 때문에 숫자를 따로 변환하지 않아도 시리얼 창에서 숫자를 읽을수 있는 특징을 가지고 있다. 이때문에 read함수는 바이트 단위로 들어오기 때문에 이들을 CHAR형태로 받은 후 print함수에 입력을 시켜 출력하면 WRITE함수와 동일한 결과를 가져올수 있었다. 두번째 실험에서는 터미널에서 입력을 받아 led와 모터를 동작하는 프로그램을 만들었다. serialavailble을 이용해 입력이 들어오면, read를 이용해서 그 값을 가져오고, if문을 이용해 1과 2를 입력할때 다른 동작을 수행하는 코드를 만들었고, 무사히 동작할수 있었다. 세번째 실험에서는 두조가 함께 아두이노를 이용해서 채팅프로그램을 만드는 실험이었는데, softwareserial을 이용하여 시리얼 데이터를 전송하는 핀과 수신하는 핀을 설정하여 컴퓨터가 아닌 두개의 아두이노를 통신할 수있었다. 이때는 상대방의 아두이노의 통신속도도 결정해야하기 때문에 9600으로 추가로 설정하였고, 신호를 주고 받는 것을 마찬가지로, serialavailable을 이용하여 코드를 작성하였고 무사히 동작하는 것을 확인 할수 있었다. 이번 실험에도 몇개의 동작하지 않는 핀이 있었기 때문에 실험전에 핀이 제대로 동작하는 지 확인하고 실험했어야 했다. 항상 실험에서 주의하면서 임해야겠다.** |
|  |