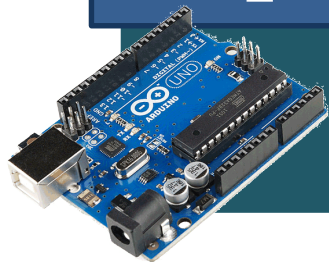


2021년 IoT기반 스마트 솔루션 개발자 양성과정



Firmware [펌웨어]

4-Serial Transmit

담당 교수 : 유근택

010-5486-5376

rgt3340@naver.com

<http://cafe.naver.com/cbdsp>



충북대학교 공동훈련센터

시리얼 통신

❖ 시리얼 통신이란?

시리얼 통신이란 연속적으로 통신 채널을 거쳐 **한 번에 1비트** 단위로 **데이터를 전송**하는 통신을 말한다.

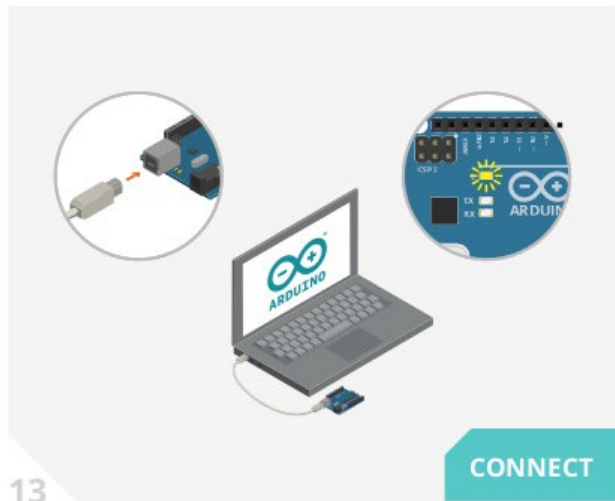
직렬 통신에서 데이터가 계속되어 전송되면, 각 비트를 구별할 방법이 필요하다. 디지털 회로의 입장에서 수신된 데이터의 비트가 시간적으로 어디서 부터 시작이고 끝인지를 알 필요가 있다. 이렇게 데이터 비트를 복구하기 위해 데이터의 시간적 위치를 알리기 위해 동기신호를 보내는 경우와 동기 신호 없이 신호 자체에서 데이터 비트를 복원하는 방식으로 나눌 수 있다.

동기 방식

- 데이터 신호와는 별도로 동기신호를 함께 보낸다.

비동기 방식

- 데이터 신호만을 보내고 각각의 방식에 따라 데이터비트를 찾아낸다.



시리얼 통신을 이용하면 오렌지보드에서 컴퓨터로 신호를 보내서 컴퓨터 화면에 표시할 수 있고, 반대로 컴퓨터에서 오렌지보드로 신호를 보내 오렌지보드에서 확인 할 수 있다.

시리얼 통신

❖ 시리얼 통신 관련 함수

Serial.begin(통신 속도)

입력한 통신 속도로 시리얼 통신을 시작한다.

Serial.available()

시리얼 통신을 통해 보드로 전송된 데이터가 있는지 판단한다.

"PC에서 전송된 데이터가 있니?"

→ "네!(True, 1)" 또는 "아니요!(False, 0)" 반환

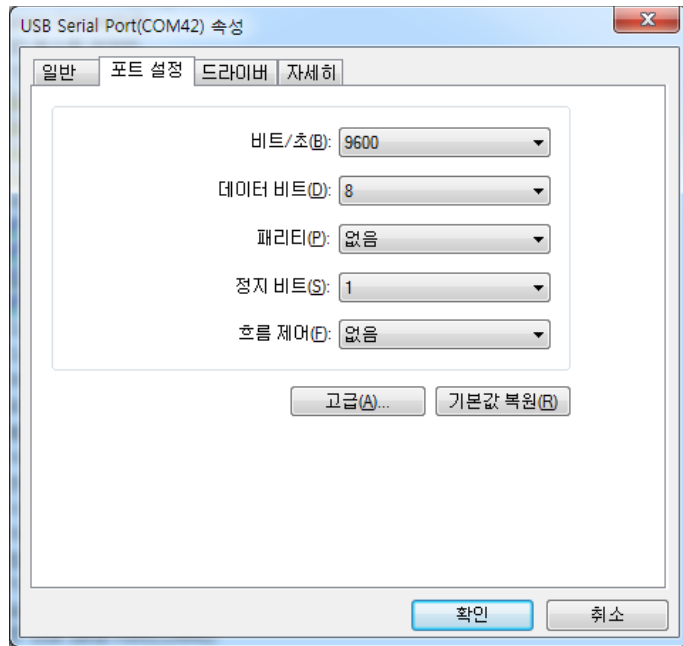
Serial.read()

전송된 데이터를 읽어들인다.

Serial.print(value)

입력한 값을 시리얼 모니터에 출력(print)한다.

print는 줄바꿈 X, println은 줄바꿈 O



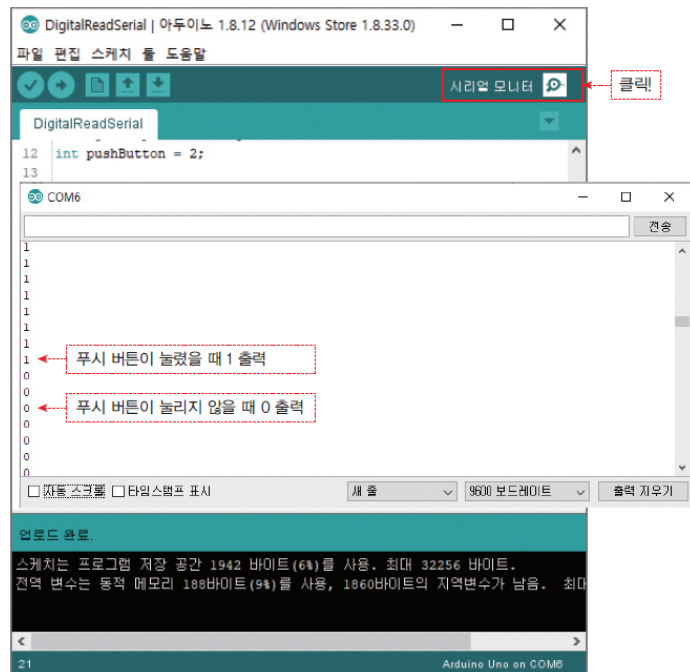
시리얼 통신

❖ 시리얼 모니터 실행과 결과 확인

```
int pushButton = 2;

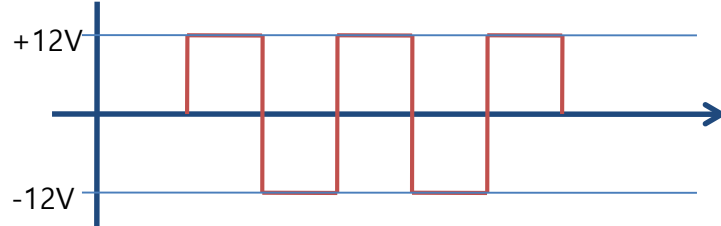
void setup()
{
    Serial.begin (9600);           // 시리얼 통신 속도를 9600 bps로 초기화
    pinMode (pushButton, INPUT);   // 푸시 버튼 핀을 입력으로 설정
}

void loop()
{
    int buttonState = digitalRead (pushButton); // 입력 핀 상태 읽음
    Serial.println (buttonState);              // 버튼 상태를 시리얼 모니터 창에 출력
    delay (1);                                 // 1 ms 지연
}
```

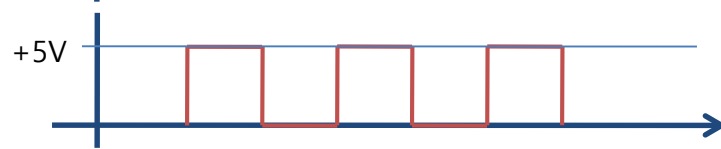


Serial 통신

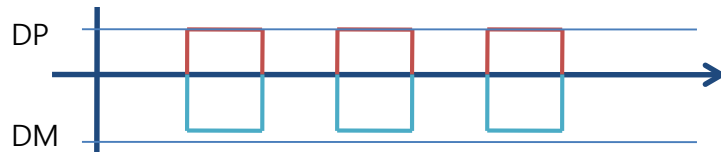
- 직렬통신(USART)
 - 동기식 : 동기 클럭과 데이터를 동시에 전송, (RX, TX, XCK)
 - 비동기식(UART: 데이터를 전송속도에 맞추어 전송 (RX, TX))
- RS232
 - 전기적인 특성, 기계적인 특성(커넥터 사양), 인터페이스 등으로 규정
 - 1 : +12V , 0 : -12V



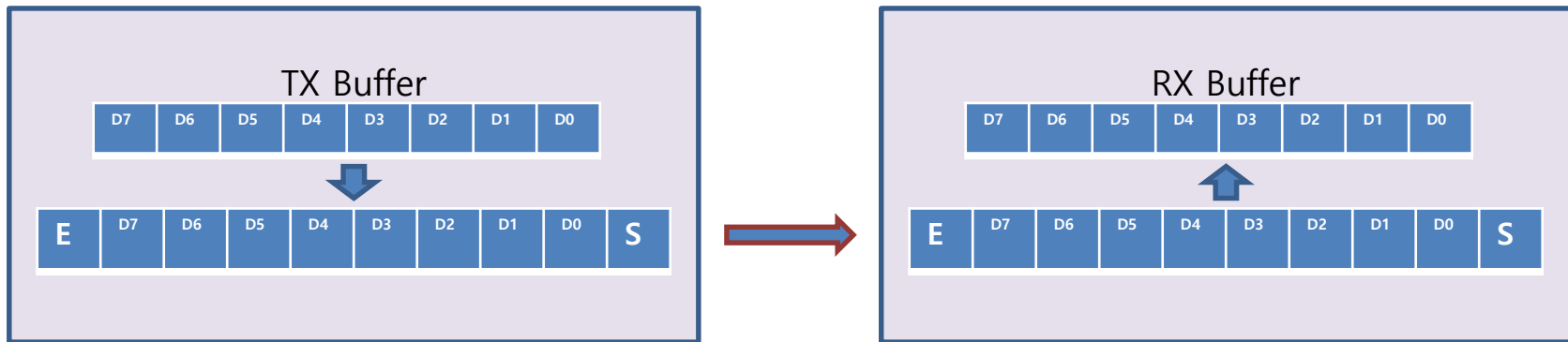
- TTL
 - 1: +5V , 0: 0V



- USB
 - DP, DM 차동

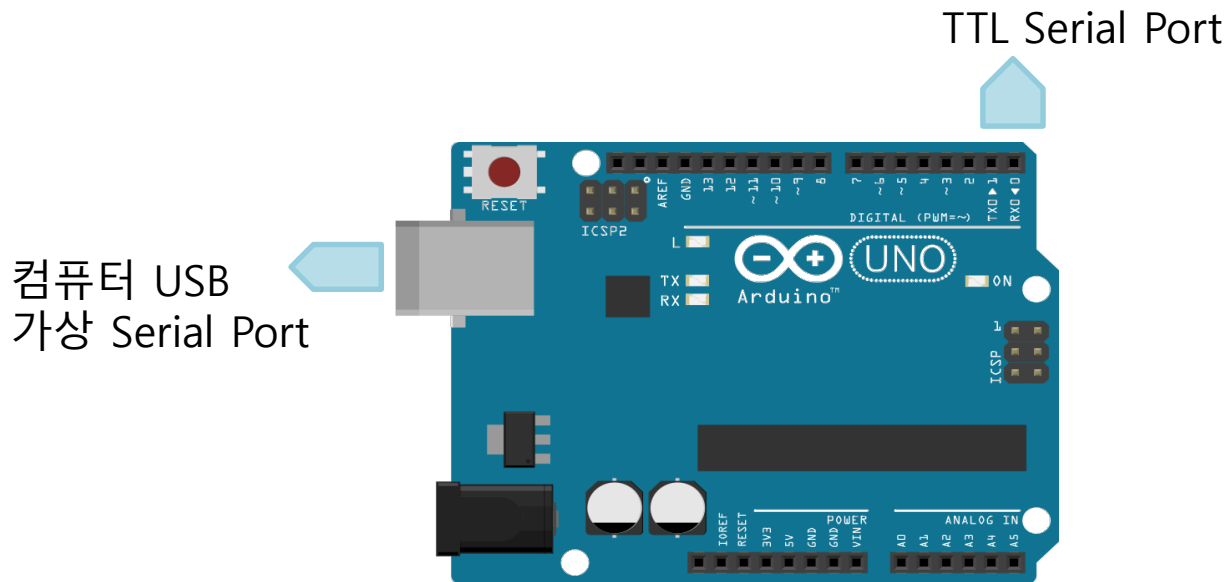


Data Flow



Arduino Serial

- Arduino Library
 - Software Serial 가능



Serial.begin/available

- Serial.begin(baud rate) //전송 속도를 설정
 - Baud rate : 초당 전송 비트수
 - 9600, 14400, 19200, 28800, 38400, 57600, 115200
 - EX) Serial.begin(115200);
- Serial.available() //수신 버퍼의 바이트 수
 - EX) int k=Serial.available();
 - EX) if ((Serial.available()>0) { -----}
 - EX) if ((Serial.available()) { -----}



Serial.print/printIn

- Serial.print(value) // value값을 문자로 변환하여 전송
 - EX) Serial.print(123); // "123" 전송
 - EX) Serial.print("Batt="); //"Batt=" 전송
- Serial.printIn(value) //value값을 문자로 변환 후 개행 문자와 함께 전송
 - 개행 문자 : "\r\n" , 0x0d 0x0a
 - \r : ASCII(13) Return, \n : ASCII(10) Line Feed



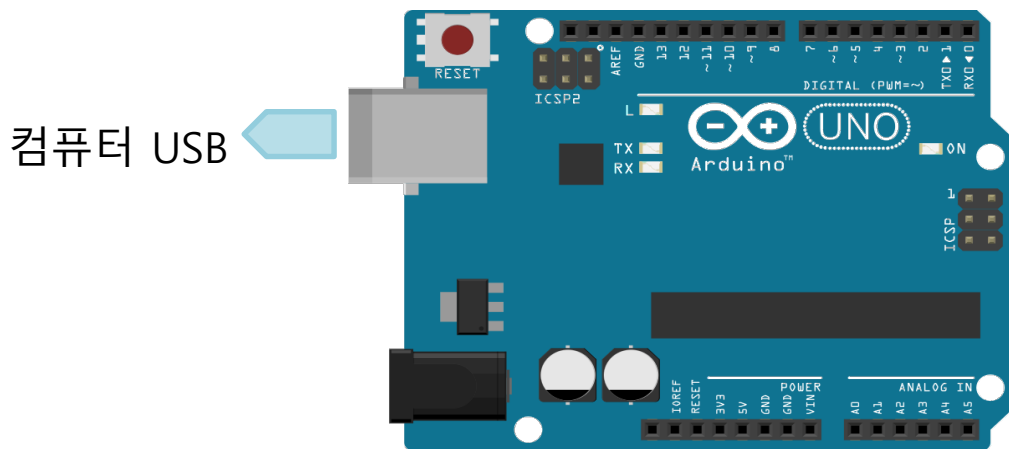
ASCII Code

10	HEX	문자	10	HEX	문자	10	HEX	문자	10	HEX	문자	10	HEX	문자	10	HEX	문자
0	0x00	NULL	22	0x16	STN	44	0x2C	.	66	0x42	B	88	0x58	X	110	0x6E	n
1	0x01	SOH	23	0x17	ETB	45	0x2D	-	67	0x43	C	89	0x59	Y	111	0x6F	o
2	0x02	STX	24	0x18	CAN	46	0x2E	.	68	0x44	D	90	0x5A	Z	112	0x70	p
3	0x03	ETX	25	0x19	EM	47	0x2F	/	69	0x45	E	91	0x5B	[113	0x71	q
4	0x04	EOT	26	0x1A	SUB	48	0x30	0	70	0x46	F	92	0x5C	\	114	0x72	r
5	0x05	ENQ	27	0x1B	ESC	49	0x31	1	71	0x47	G	93	0x5D]	115	0x73	s
6	0x06	ACK	28	0x1C	FS	50	0x32	2	72	0x48	H	94	0x5E	^	116	0x74	t
7	0x07	BEL	29	0x1D	GS	51	0x33	3	73	0x49	I	95	0x5F	_	117	0x75	u
8	0x08	BS	30	0x1E	RS	52	0x34	4	74	0x4A	J	96	0x60	.	118	0x76	v
9	0x09	HT	31	0x1F	US	53	0x35	5	75	0x4B	K	97	0x61	a	119	0x77	w
10	0x0A	LF	32	0x20	SP	54	0x36	6	76	0x4C	L	98	0x62	b	120	0x78	x
11	0x0B	VT	33	0x21	!	55	0x37	7	77	0x4D	M	99	0x63	c	121	0x79	y
12	0x0C	FF	34	0x22	"	56	0x38	8	78	0x4E	N	100	0x64	d	122	0x7A	z
13	0x0D	CR	35	0x23	#	57	0x39	9	79	0x4F	O	101	0x65	e	123	0x7B	{
14	0x0E	SO	36	0x24	\$	58	0x3A	:	80	0x50	P	102	0x66	f	124	0x7C	
15	0x0F	SI	37	0x25	%	59	0x3B	;	81	0x51	Q	103	0x67	g	125	0x7D	}
16	0x10	DEL	38	0x26	&	60	0x3C	<	82	0x52	R	104	0x68	h	126	0x7E	~
17	0x11	DC1	39	0x27	'	61	0x3D	=	83	0x53	S	105	0x69	i	127	0x7F	DEL
18	0x12	DC2	40	0x28	(62	0x3E	>	84	0x54	T	106	0x6A	j			
19	0x13	DC3	41	0x29)	63	0x3F	?	85	0x55	U	107	0x6B	k			
20	0x14	DC4	42	0x2A	*	64	0x40	@	86	0x56	V	108	0x6C	l			
21	0x15	NAK	43	0x2B	+	65	0x41	A	87	0x57	W	109	0x6D	m			



A4-1 : USB Serial Tx

- USB Port를 이용하여 매 1초마다 Data 송신
- Data는 0~ 9 까지의 1 Byte 숫자



A4-1 : Program

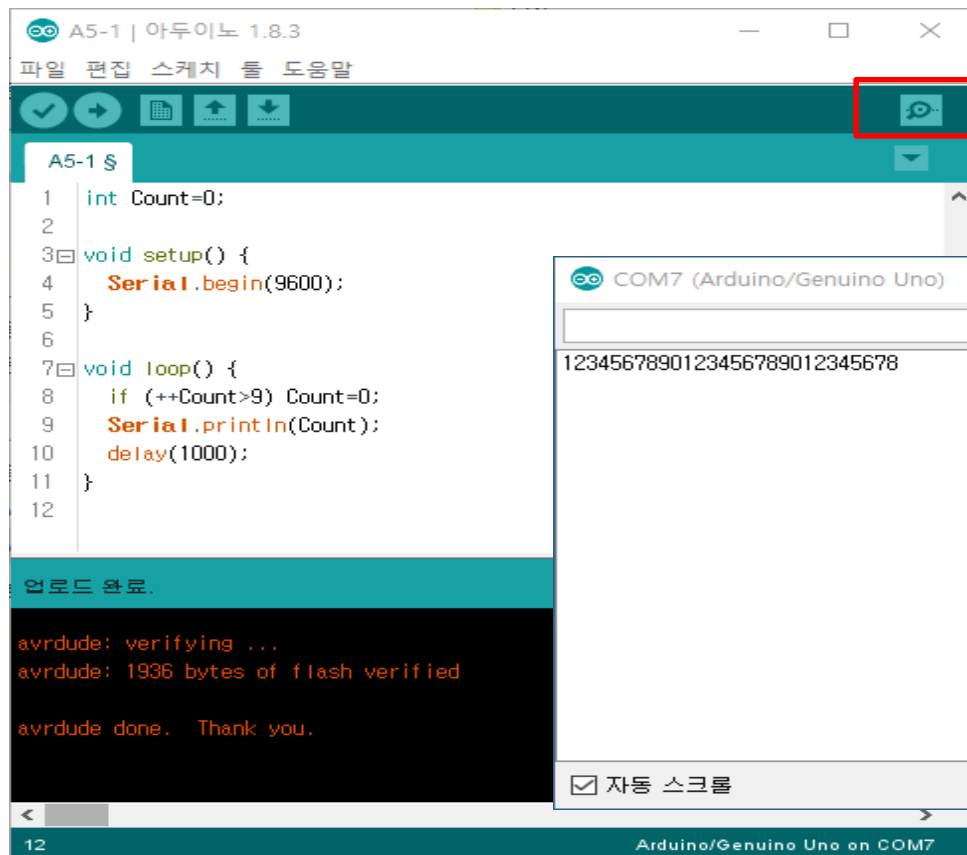
```
int Count=0;

void setup( ) {
  Serial.begin(9600);
}

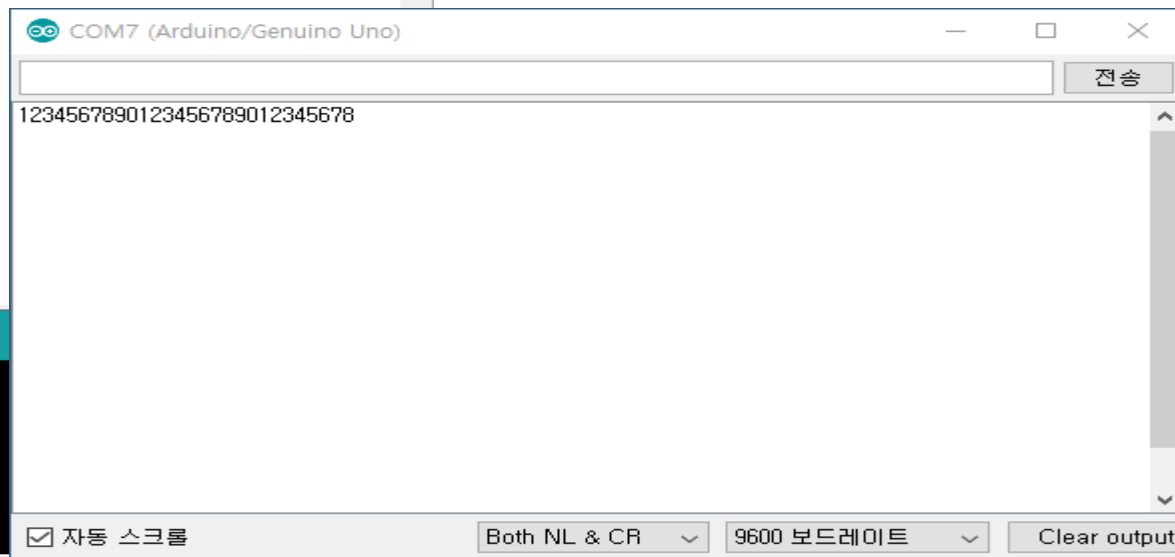
void loop( ) {
  if (++Count>9) Count=0;
  Serial.print(Count);
  delay(1000);
}
```



Serial Monitor



Serial Monitor로 확인

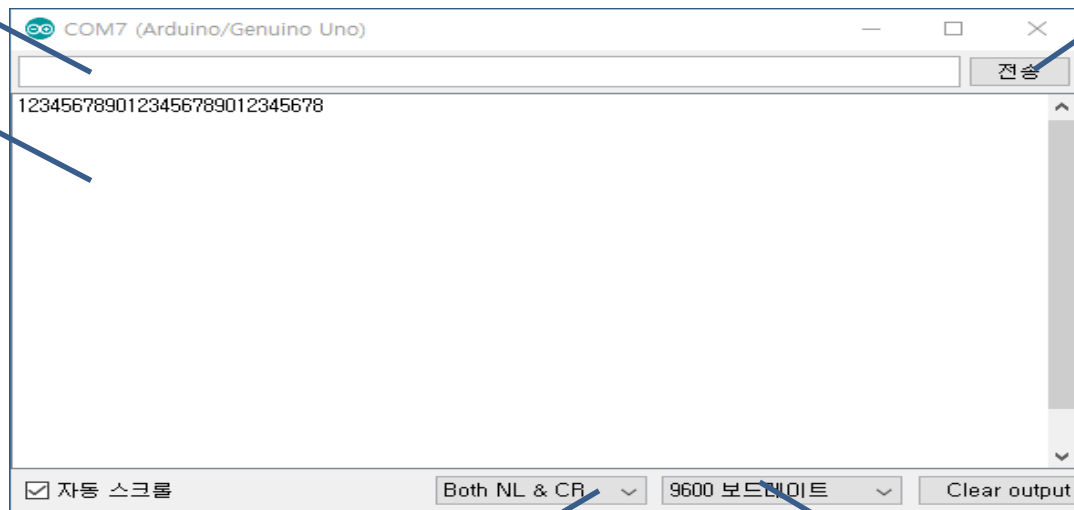


Serial Monitor Window

송신 문자열

수신 문자열

전송 버튼



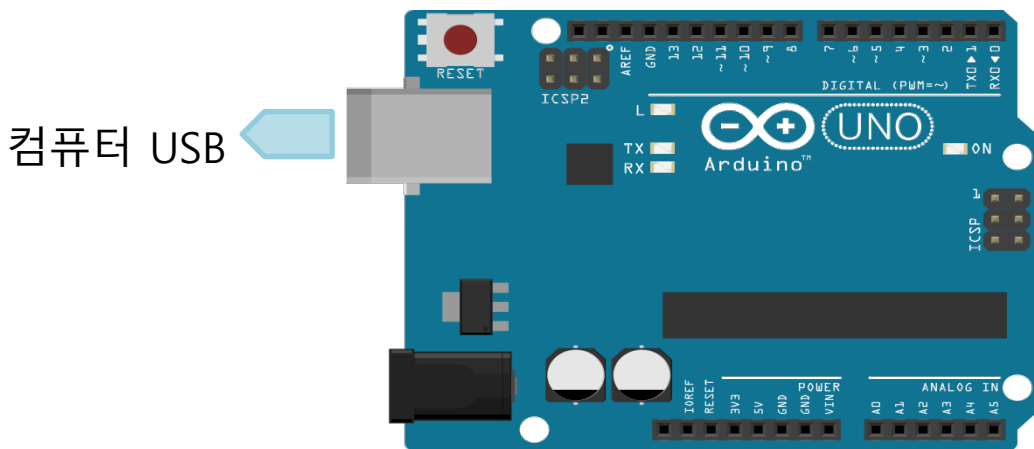
Line Ending 없음
새줄
케리지 리턴
Both NL & CR

전송 보드레이트



A4-1 : USB Serial Tx

- USB Port를 이용하여 매 1초마다 Data 송신
- Data는 0~ 9 까지의 1 Byte 숫자
- **Count = 0, 1, 2 , 3**

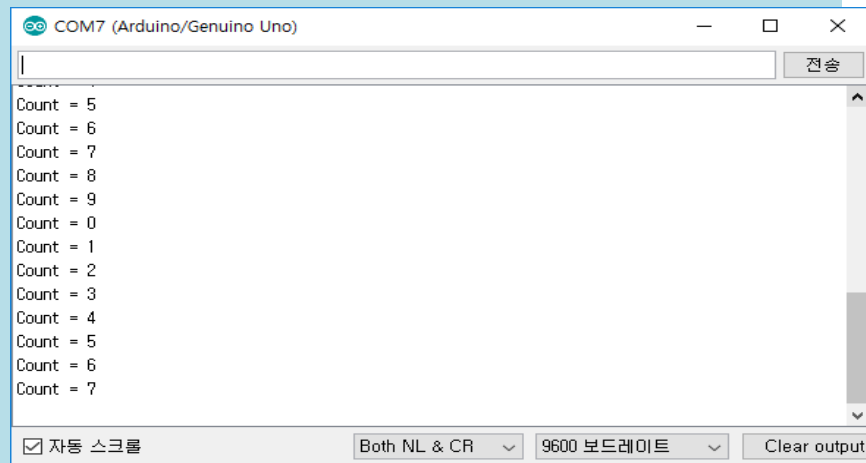


A4-2 : Program

```
int Count=0;
```

```
void setup( ) {  
    Serial.begin(9600);  
}
```

```
void loop( ) {  
    if (++Count>9) Count=0;  
    Serial.print("Count = ");  
    Serial.println(Count);  
    delay(1000);  
}
```



Serial.write()

- **Syntax**

`Serial.write(val)`

`Serial.write(str)`

`Serial.write(buf, len)`

- **Parameters**

val : byte형 Data

str : String형 문자열

buf : byte형 문자배열

len : 배열의 길이

- **Exam**

`Serial.write(45);` *// send a byte with the value 45 'A'*

`Serial.write("hello");` *//send the string "hello" and return the length of the string.*



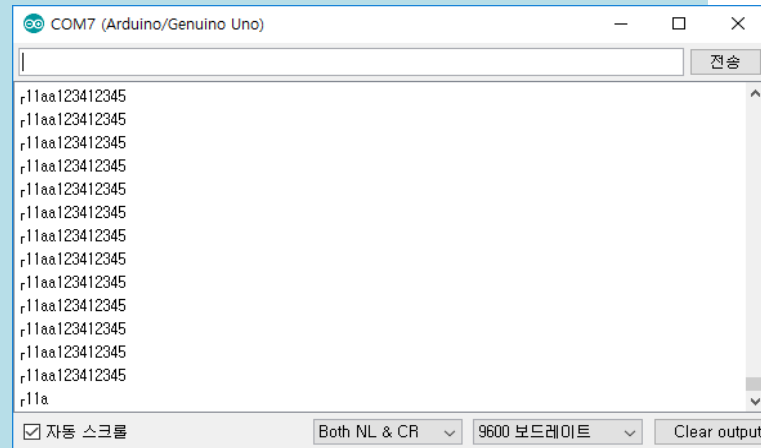
A4-3 : Serial.write()

```
const unsigned char temp[5] = {'1', '2', '3', '4', '5'};
void setup( ){
  Serial.begin(9600);
}
void loop( ){
  Serial.write(1);      // write( )함수로 1 전송
  Serial.print(1);      // print( )함수로 1 전송
  Serial.write(49);     // write( )함수로 49 전송

  Serial.write(0x61);    // write( )함수로 'a' 전송
  Serial.write('a');     // write( )함수로 'a' 전송

  Serial.write(temp, 4); // write( )함수로 temp배열을 4만큼 전송
  Serial.write("12345"); // write( )함수로 string값 전송

  Serial.write('\n');    // 줄바꿈
  delay(500);
}
```



randomSeed()

- Syntax

`randomSeed(seed)`

- Parameter

seed : 난수 발생 지점

- Exam

`randomSeed(analogRead(5));` *//analogRead(5)으로 부터 seed형성*



random()

- Syntax

`random(max)`

`ranodm(min, max)`

- Parameter

`max` : 최대값

`min` : 최소값

- Exam

`int randomX=random(300); //0~299`

`int randomX=random(10,20); //10~19`



A4-4 : random()

```
int randNumber;

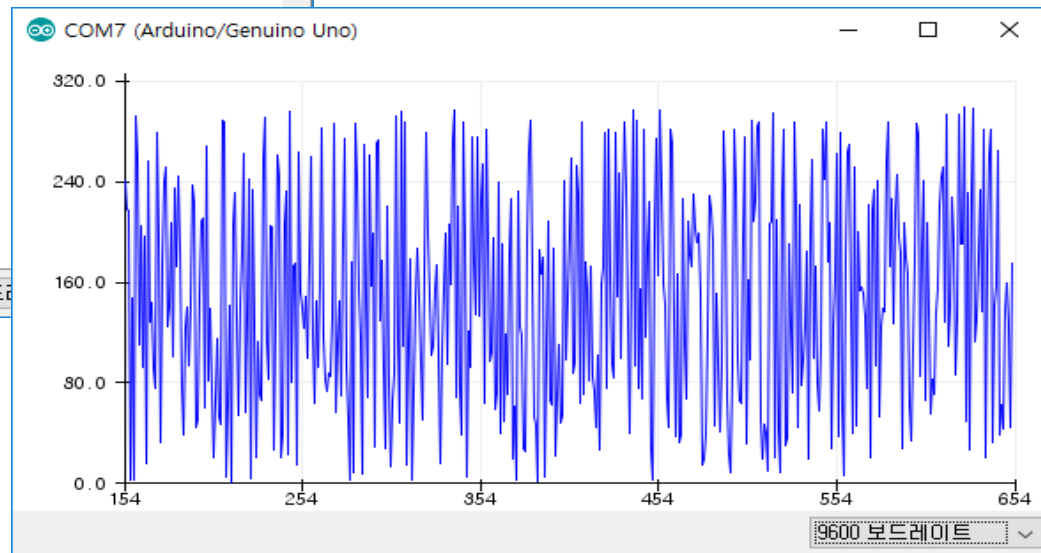
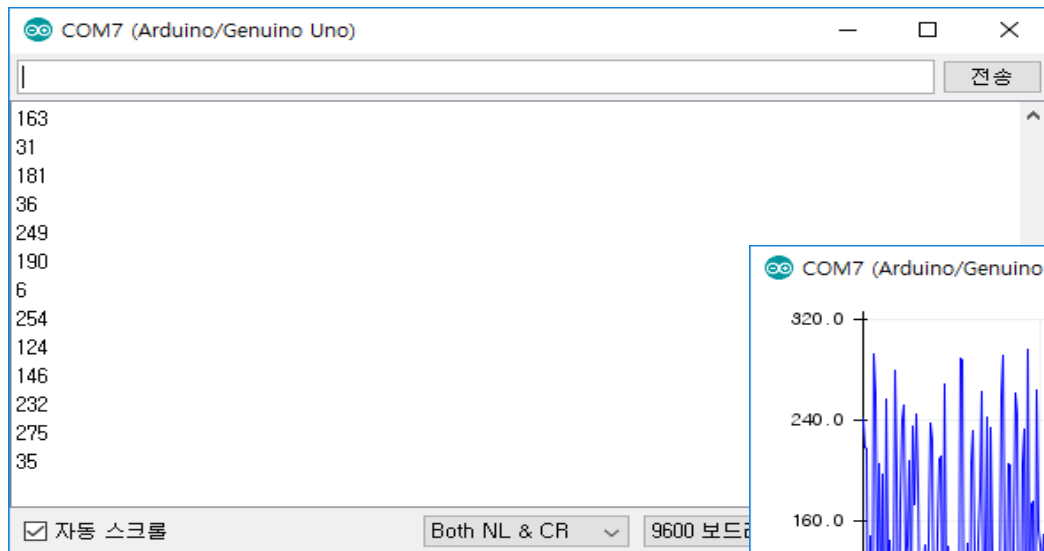
void setup( ){
  Serial.begin(9600);
  randomSeed(analogRead(5));
}

void loop( ){
  randNumber = random(300);
  Serial.println(randNumber);

  delay(50);
}
```

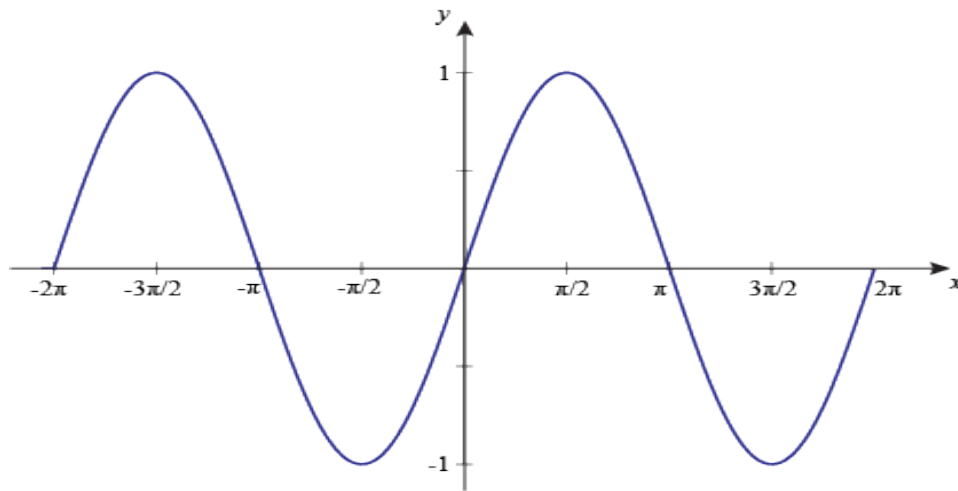


A4-4 : Serial Plotter



A4-5 : Sin Graph

- $Y = \sin\theta$
- X : 위상 $0 \sim 360^\circ$
- rad을 degree로 변환 , $\text{degree} = 180^\circ / \Phi \text{ radian}$
 $\Rightarrow Y = \sin \theta * (180^\circ / \Phi)$

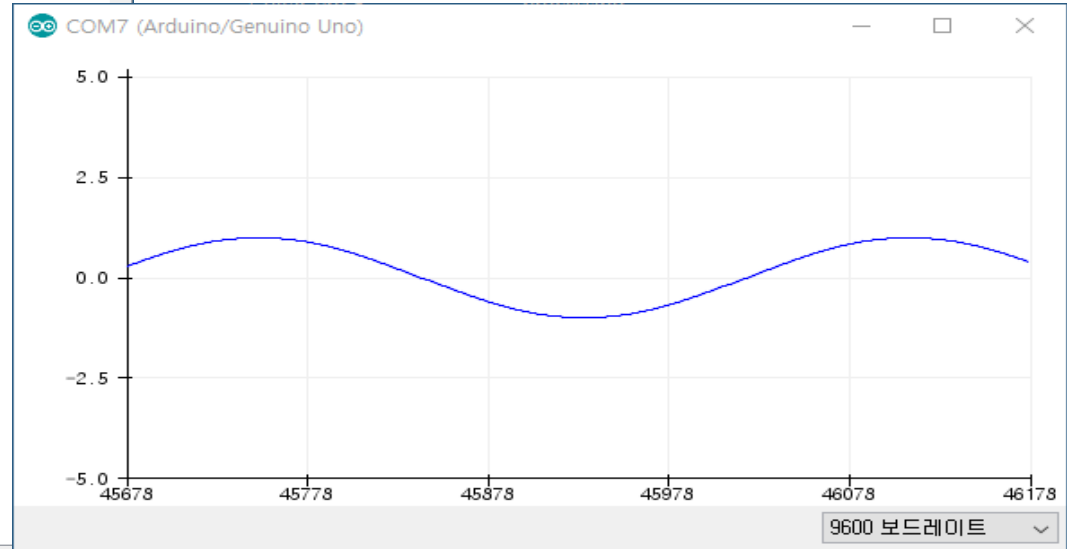
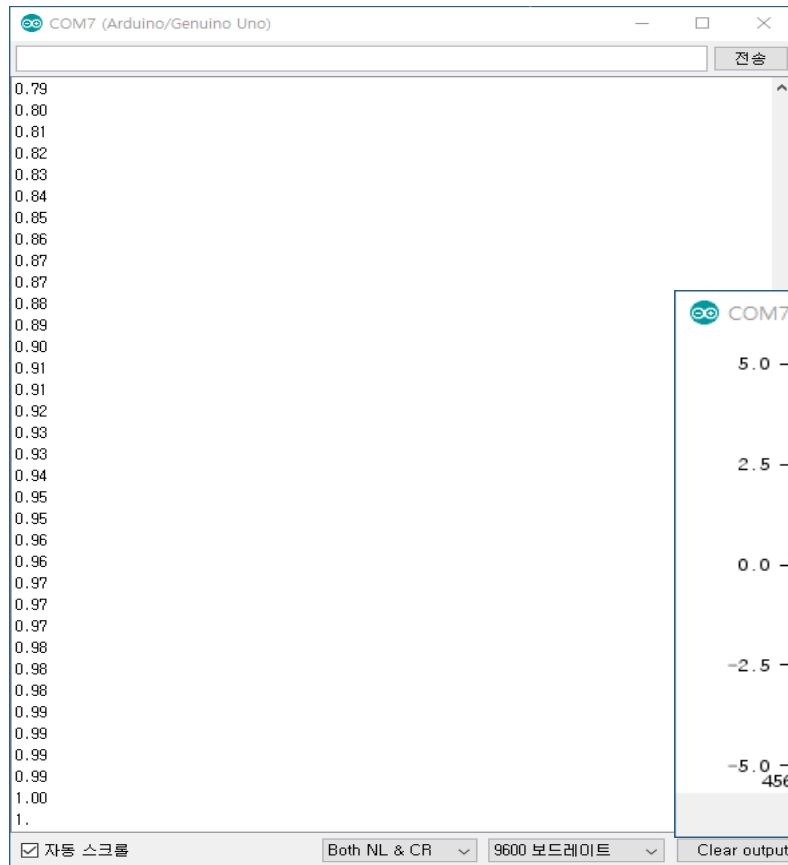


A4-5 : Program

```
void setup( ) {  
    Serial.begin(9600);  
}  
  
void loop( ) {  
    for (int k = 0; k < 360; k++) {  
        Serial.println(sin(k * (PI / 180)));  
    }  
}
```



A4-5 : Serial Monitor/Plotter



능력 향상



충북대학교 공동훈련센터

Arduino에서 컴퓨터로 데이터 전송하기

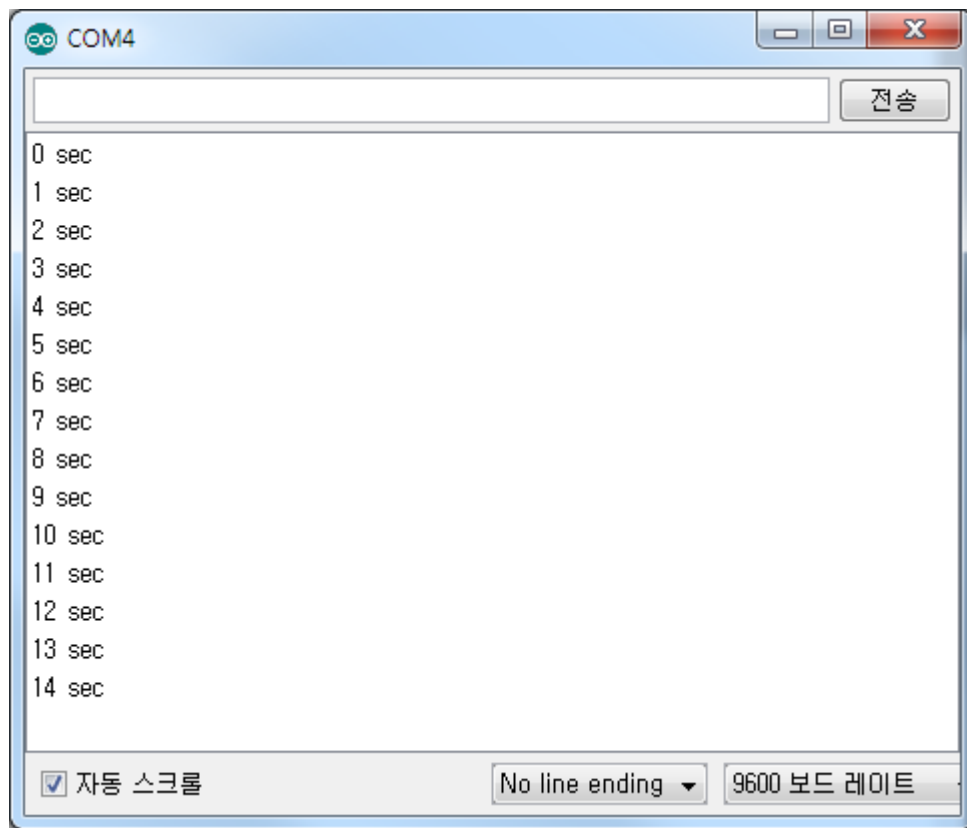
- **Arduino**에서 컴퓨터로 변수와 문자열 전송하기
 1. Arduino에서 문자열과 데이터를 시리얼 통신을 이용하여 컴퓨터로 전송한다.
 2. 전송할 데이터는 0부터 1초 간격으로 1씩 증가하는 숫자와 'sec'라는 문자열이다.
 3. Arduino IDE의 시리얼 모니터에서 이를 확인해 본다.



Commands

- **Serial.begin(전송속도)**
 - 시리얼 통신 포트를 컴퓨터와 연결한다. 전송속도는 bps(bits per sec)로 일반적으로 9600으로 설정한다. 19200, 57600, 115200 등의 값을 설정할 수 있다.
- **Serial.print(전송내용)**
 - 괄호 안의 내용을 시리얼 통신으로 전송한다. 따옴표로 구분된 부분은 텍스트를 직접 전송하고 따옴표 없이 변수를 써주면 변수의 값이 전송된다.
- **Serial.println(전송내용)**
 - 'Serial.print'와 같으나 전송 뒤 줄 바꿈을 한다.
- **delay(지연시간)**
 - 지연시간에는 잠시 동작을 지연시키기 위한 값을 넣는다. 1/1000초 단위로 넣는다. 즉 1초를 지연시키기 위해선 1000의 값을 입력시킨다.





- 변수 유형별 **Arduino**에서 컴퓨터로 전송하기
 1. Arduino에서 컴퓨터로 데이터를 전송할 때 변수 유형별로 출력한다.
 2. char로 선언된 변수, int로 선언된 변수, float로 선언된 변수를
`Serial.print` 명령어를 이용하여 PC로 전송하자.
 3. `Serial.print` 명령어의 출력 옵션을 변경하여 전송해 보자.
 4. 문자열 변수를 사용해 보자.
 5. 각 변수 유형별 출력되는 차이를 비교해 보자.

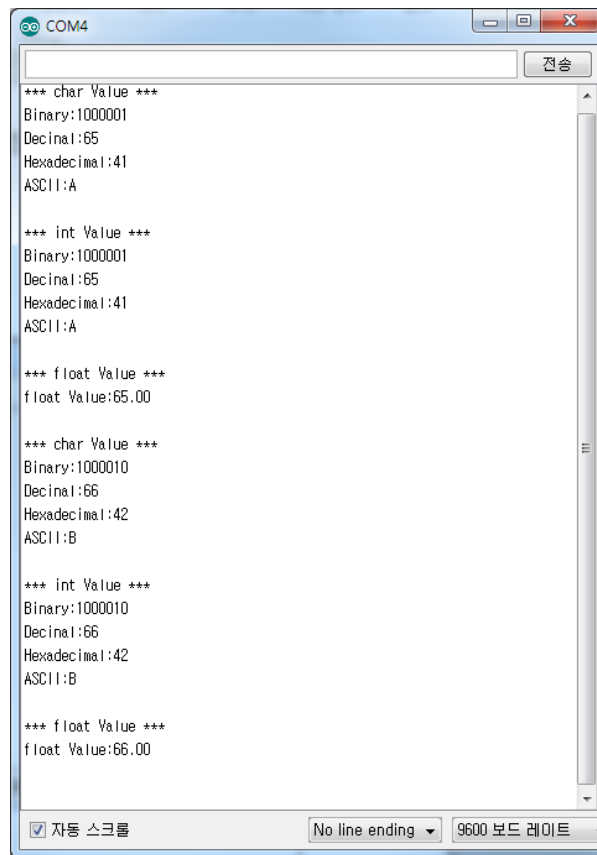


Commands

- `Serial.write(char 변수);`
 - char변수에 해당하는 ASCII 코드값의 문자를 출력한다.
- `Serial.print(변수,BIN);`
 - 변수를 2진수(Binary)로 표시한다.
- `Serial.print(변수,DEC);`
 - 변수를 10진수(Binary)로 표시한다.
- `Serial.print(변수,HEX);`
 - 정해진 변수를 16진수(Hexadecimal)로 표시한다.



Result



A screenshot of a Windows terminal window titled "COM4". The window contains the following text output:

```
*** char Value ***  
Binary:1000001  
Decimal:65  
Hexadecimal:41  
ASCII:A  
  
*** int Value ***  
Binary:1000001  
Decimal:65  
Hexadecimal:41  
ASCII:A  
  
*** float Value ***  
float Value:65.00  
  
*** char Value ***  
Binary:1000010  
Decimal:66  
Hexadecimal:42  
ASCII:B  
  
*** int Value ***  
Binary:1000010  
Decimal:66  
Hexadecimal:42  
ASCII:B  
  
*** float Value ***  
float Value:66.00
```

At the bottom of the window, there are three checkboxes: "자동 스크롤" (checked), "No line ending", and "9600 보드 레이트".

