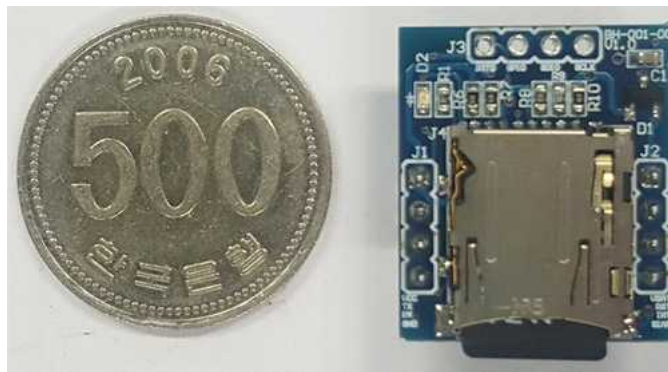
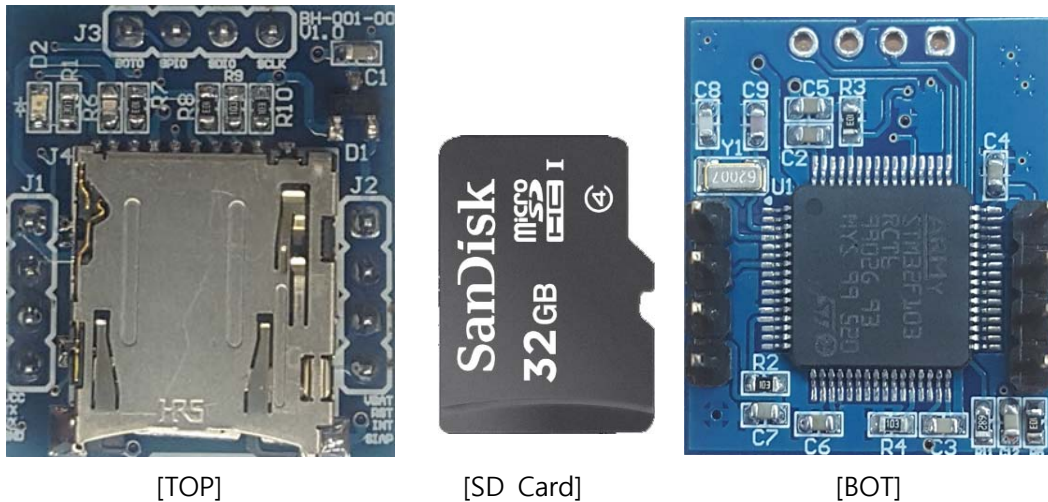


BH-001-00
Micro SD Card Logging
User's Guide



1. 제품개요



[TOP]

[SD Card]

[BOT]

본 제품은 SD 카드를 시리얼 통신으로 간단히 사용할 수 있도록 만든 시리얼 SD 카드 모듈이다. 일반적인 RS232C를 사용하여 다양한 프로세서(아두이노, AVR, Cortex, 8051, PIC, DSP 등)와 쉽게 연결하여 사용할 수 있다.

2. 제품특징

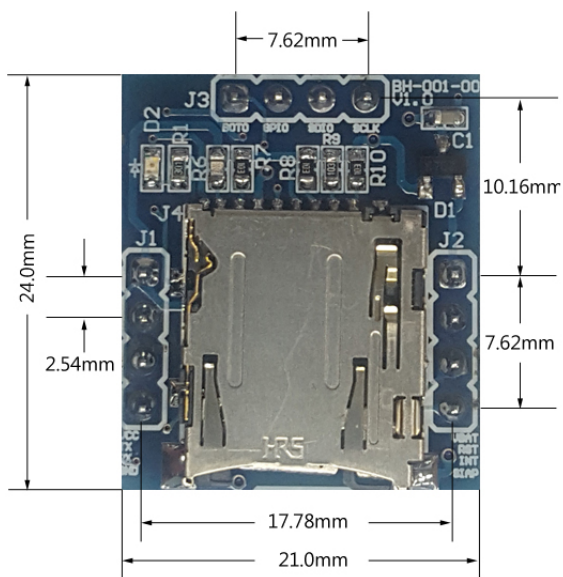
- ▷ 최대 128GB 용량의 Micro SD 카드 사용 가능
- ▷ 간결한 명령에 의해 시리얼통신(TTL, RS232C) 통해 SD 카드 파일 생성, 지우기, 쓰기 읽기 동작수행
- ▷ SD 카드 포맷 가능
- ▷ 카드 장착 자동 검출기능 및 물리적인 핀 출력(EINT_SD : Low(검출))
- ▷ 통신 속도 : 9600 ~ 115200bps, No Parity, 8Bit Data, 1 Stop Bit
Default : 9600bps
- ▷ 통신용 단자(3.3V Level) : RX, TX, GND
- ▷ 펌웨어 업그레이드 기능 설명

※ 주의사항

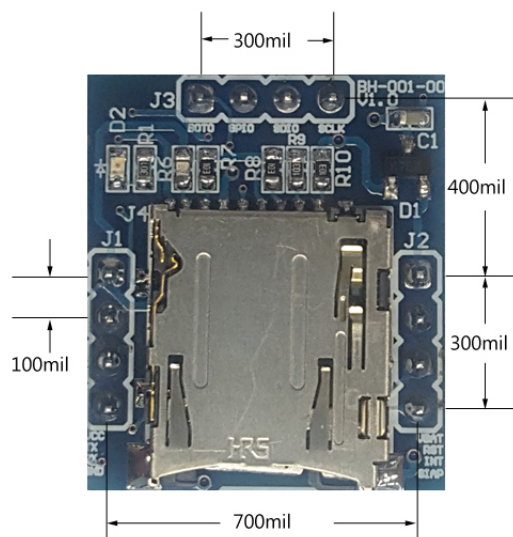
- ▷ SD 카드 최초 장착 후에는 반드시 포맷하여 사용한다.
- ▷ 파일 액세스 중에는 SD 카드를 제거하면 손상될 수 있다.
- ▷ 모든 종류의 SD 카드의 호환성을 보장하지 않는다.
(특히 중국제 저가제품 SD카드 동작은 보장하지 않음)

3. 제품 외형

3.1 기구도면

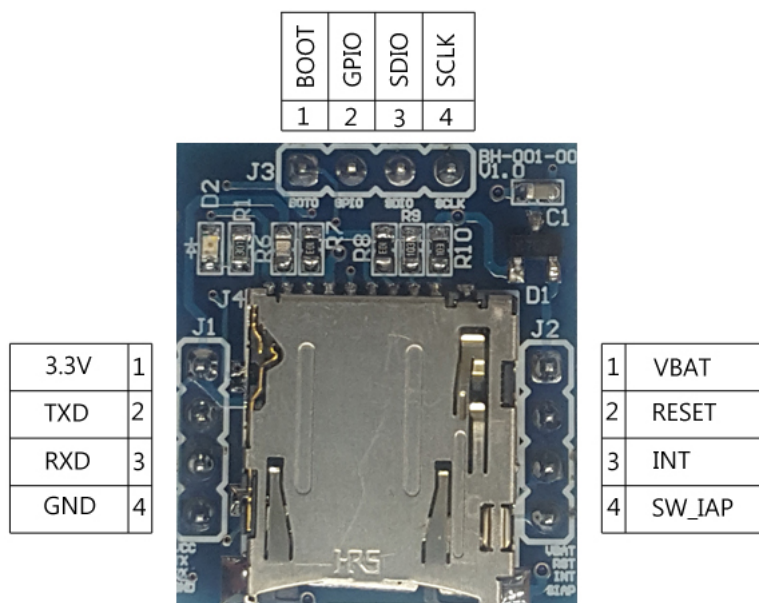


[단위 : mm]



[단위 : mil]

3.2 핀 기능

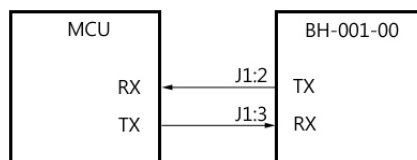


[핀 정의]

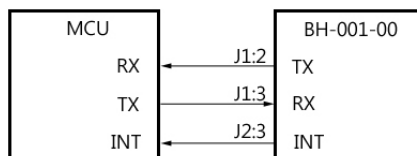
Header	번 호	명 칭	기 능
J1	1	3.3V	전원
	2	TXD	3.3V 레벨의 송신용 출력 핀
	3	RXD	3.3V 레벨의 수신용 입력 핀
	4	GND	Ground
J2	1	VBAT	Battery 입력 핀
	2	RESET	Reset(10K 저항 풀업)
	3	INT	SD 카드 장착 유·무를 나타내는 출력 핀 - SD 카드 삽입 : Low - SD 카드 제거 : High
	4	SW_IAP	펌웨어 업그레이드 설정 핀
J3	1	BOOT	제조사 전용
	2	GPIO	제조사 전용
	3	SDIO	제조사 전용
	4	SCLK	제조사 전용

4. 연결방법

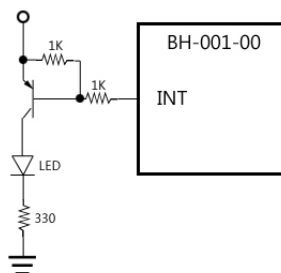
▷ 마이크로 프로세서와 연결 예시



▷ INT 연결 예시 : INT 핀을 마이크로 프로세서의 외부 입력 인터럽트 핀에 연결하여 SD 카드 장착 여부를 확인할 수 있다.



▷ INT 핀 LED 연결 예시

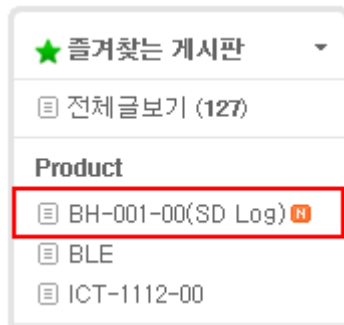


5. 펌웨어 업그레이드 방법

5.1 히스코리아가 운영하는 카페에서 최신 펌웨어 파일제공

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

5.2 Product 항목아래 BH-001-00(SD Log)를 클릭한다.

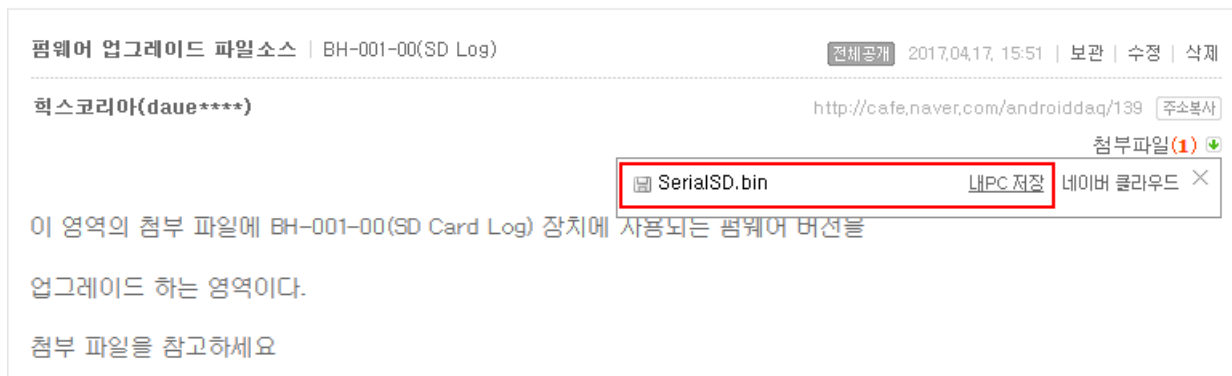


5.3 “펌웨어 업그레이드 파일소스”를 클릭한다.

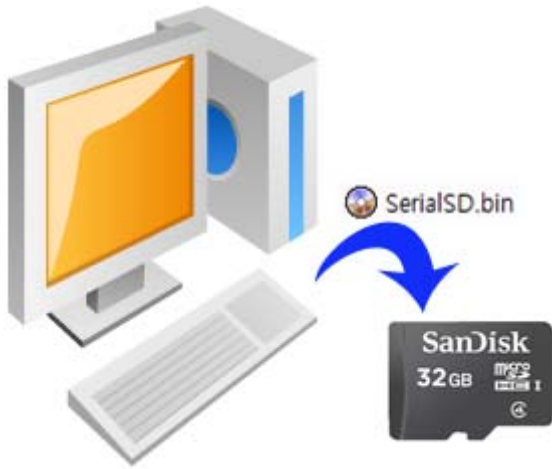
	제목	작성자	작성일	조회	좋아요
<input type="checkbox"/>	공지 BH-001-00(SD Card Log) 제품설명	히스코리아	15:49	0	0
<input type="checkbox"/>	공지 카페를 개설하며.... [1]	seogarae2	2013.02.06	72	0
<input type="checkbox"/>	139 펌웨어 업그레이드 파일소스	히스코리아	15:51	0	0
<input type="checkbox"/>	138 BH-001-00(SD Card Log) 제품설명	히스코리아	15:49	0	0

☐ 전체선택 이동 삭제 글쓰기

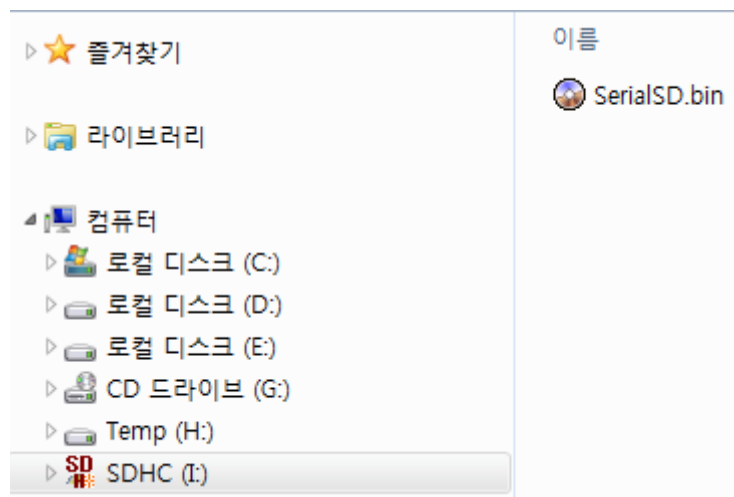
5.4 첨부파일을 선택하여 “SerialSD.bin” 파일을 사용자 PC에 저장한다.



5.5 PC에 SD Card를 삽입하고 "SerialSD.bin" 파일을 SD Card에 복사한다.

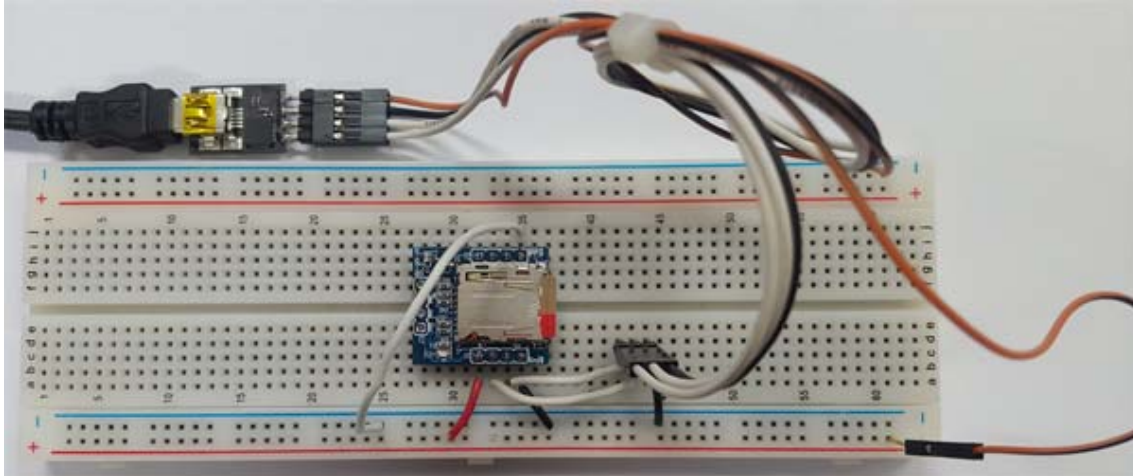


5.6 SD Card에 "SerialSD.bin"파일을 복사한다.



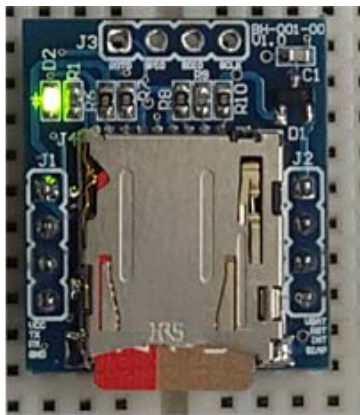
5.7 아래 그림과 같이 하드웨어를 구성한다.

- ▷ PC에 USB to Serial를 연결하지 않고 아래 그림과 같이 구성한다.
- ▷ USB to Serial(**TTL 3.3V Level**) 장치와 BH-001-00을 연결하고 J2(4번) : SW_IAP 단자를 Low에 연결한다.
- ▷ 펌웨어 로그데이터를 확인하고 싶지 않을 때는 RX, TX 단자 연결하지 않아도 된다. 아래 에서는 로그 데이터를 확인하는 방법을 설명하기 위해 RX, TX 단자를 연결하였다.



5.8 BH-001-00 장치에 SD Card를 삽입하고 USB to Serial 장치를 PC에 연결한다.

- ▷ 전원을 인가하고 펌웨어 업그레이드 동작 수행 중에는 녹색 LED가 점등된다.
- ▷ 펌웨어 업그레이드 작업이 완료되면 녹색 LED가 소등된다.
- ▷ 응용 프로그램의 정상적인 수행을 확인하기 위해 녹색 LED가 3회 점멸 후 소등된다.



[업그레이드 중]



[업그레이드 완료]

6. 통신 프로토콜

6.1 프로토콜 기본 규칙

- ▷ Host와 BH-001-00 상호간의 데이터 송·수신은 UART 접속을 기반으로 한다.
- ▷ UART 기본 설정 값
 - Baud Rate : 9600bps
 - Data Bit : 8
 - Parity Bit : None
 - Stop Bit : 1
 - Flow Control : None
 - 기본 설정 값의 변경을 원할 경우 baud 명령에 의해 수정할 수 있다.
- ▷ 통신 규칙 : 모든 프로토콜은 ASCII 값의 조합으로 구성되며 "Carriage Return"을 통해 명령의 끝을 알린다.
- ▷ 명령형태
 - AT명령어 형태를 기본으로 "명령어 [옵션:데이터][CR]" 형식을 갖는다.
 - 명령어에 따라서 옵션 또는 옵션에 데이터를 포함하며, 명령 끝에는 CR을 반드시 붙여야 한다.
 - 명령어 : 명령어는 소문자를 사용한다.
 - 옵션 : 옵션 구분자는 "-"이며, 옵션에 데이터가 필요할 경우 옵션 뒤에 ":"붙이고 데이터를 넣는다.
- ▷ 응답형태 : AT명령에 대한 응답 형태는 3가지 형식을 갖는다.
 - 형태1 : 명령 정상 수행 후 응답 데이터가 없을 경우
OK[CR]
 - 형태 2 : 명령 정상 수행 후 응답 데이터가 있을 경우
데이터[CR]OK[CR]
 - 형태 3 : 명령 정상 수행을 하지 못한 경우
ERR_xx(xx : 에러 코드)

▷ 명령 리스트

명 령	옵 션	응 답
at		OK
baud	-s:1	
time	-s:hmmss	
date	-s:yymmdd	
mkfs	-f	
open	-f:filename	
close		
lseek	-n:file offset	
put	-w:string	
	-a:CR/LF 추가	
read	-n:number of bytes	
reada	-f:filename	
readl	-f:filename -s:start line -n: line count	

▷ 응답 에러코드

번호	설 명
00	펌웨어 업데이트 에러
01	SD Card 디스크 I/O 레이어 에러
02	프로그램 에러
03	물리적인 드라이브 작동하지 않음
04	파일을 발견할 수 없음
05	경로를 발견할 수 없음
06	경로이름 형식이 잘못되었음
07	금지 된 액세스 또는 디렉토리가 꽉 차서 액세스가 거부되었음
08	금지 된 액세스로 인해 액세스가 거부되었음
09	파일 / 디렉토리가 유효하지 않음
10	드라이브가 보호되어 있음.
11	논리 드라이브 번호가 잘못되었음
12	볼륨에 작업 영역이 없음
13	유효한 FAT 볼륨이 없음
14	매개 변수 오류로 인해 f_mkfs()가 중단되었음
15	정의 된 기간 내에 볼륨에 액세스 할 수 있는 권한을 얻을 수 없음
16	파일 공유 규칙에 따라 작업이 거부됨
17	LFN 작업 버퍼를 할당 할 수 없음
18	열린 파일 수 > _FS_SHARE
19	주어진 매개 변수가 유효하지 않음
50	아규먼트 에러
51	옵션 에러
52	옵션 데이터 에러
53	옵션 데이터 형식 에러
54	메모리 마운트 에러
60	읽을 데이터 없음
97	핸들러 에러
98	명령 없음
99	펌웨어 오류

▷ 통신규약 설명

① at

동 작	Host에서 BH-0001-00에 요청
응 답	OK
설 명	정상적인 명령을 수행하면 "OK"응답
예 시	(HOST→BH-001-00) : at (BH-001-00→HOST) : OK

② baud

동 작	Host에서 BH-0001-00와의 통신 속도를 확인 및 변경한다.
응 답	OK 또는 ERRxx
설 명	정상적인 명령을 수행하면 "OK"응답 그렇지 않다면 "ERRxx"를 응답 baud -s:통신 속도 통신 속도는 아래 종류가 가능 함 9600 14400 19200 38400 57600 115200
예 시	통신 속도 확인 (HOST→BH-001-00) : baud (BH-001-00→HOST) : 9600OK 통신 속도 설정 (HOST→BH-001-00) : baud -s:115200 (BH-001-00→HOST) : OK

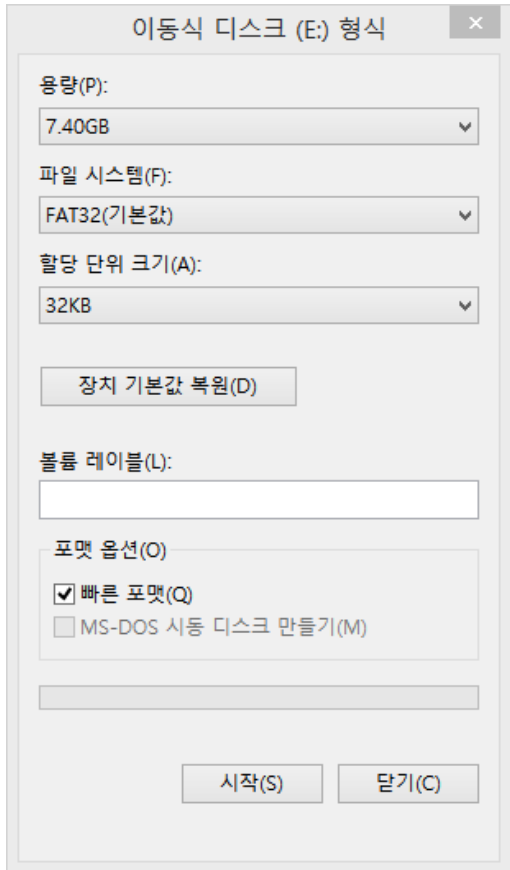
③ time

동 작	Host에서 BH-0001-00의 시간 정보를 확인 및 변경한다.
응 답	OK 또는 ERRxx
설 명	정상적인 명령을 수행하면 "OK" 응답 time -s:hhmmss BH-0001-00 전원 공급이 차단되면 초기화 된다.
예 시	시간 확인 (HOST→BH-001-00) : time (BH-001-00→HOST) : 23:59:00 OK (HOST→BH-001-00) : time -s:120000 (BH-001-00→HOST) : OK

④ date

동 작	Host에서 BH-0001-00의 날짜 정보를 확인 및 변경한다.
응 답	OK 또는 ERRxx
설 명	정상적인 명령을 수행하면 "OK" 응답 date -s:yymmdd 요일은 날짜를 계산을 통해서 자동 설정된다. 요일 코드 0 : SUNDAY 1 : MONDAY 2 : TUESDAY 3 : WEDNESDAY 4 : THURSDAY 5 : FRIDAY 6 : SATURDAY BH-0001-00 전원 공급이 차단되면 초기화 된다.
예 시	날짜는 2017년 1월 1일 이후 설정해야 한다. 날짜 확인 (요일-월-일-년도) 형식으로 반환 (HOST→BH-001-00) : date (BH-001-00→HOST) : 0-01-02-2017 OK 날짜 설정 (HOST→BH-001-00) : date -s:170421 (BH-001-00→HOST) : OK

⑤ mkfs

동 작	uSD(Micro 메모리를 포맷한다.
응 답	OK 또는 ERRxx
설 명	<p>정상적인 명령을 수행하면 "OK"응답 mkfs -f</p> <p>FAT32 형식으로 포맷된다. 포맷 결과를 PC에서 확인한 결과는 아래와 같다.</p> 
예 시	<p>(HOST→BH-001-00) : mkfs -f</p> <p>(BH-001-00→HOST) : OK</p>

⑥ open

동 작	기록할 파일을 엽니다.
응 답	OK 또는 ERRxx
설 명	정상적인 명령을 수행하면 "OK"응답 open -f:filename 파일이 없으면 파일을 새롭게 생성한다. 기존 파일이 존재하면 파일에 데이터를 추가하게 된다. 읽기/추가 속성으로 파일을 엽니다. 파일명은 최대 128바이트까지 지원한다.
예 시	(HOST→BH-001-00) : open -f:log.txt (BH-001-00→HOST) : OK

⑦ close

동 작	open 명령으로 엽힌 파일을 닫는다.
응 답	OK 또는 ERRxx
설 명	정상적인 명령을 수행하면 "OK"응답 close
예 시	(HOST→BH-001-00) : close (BH-001-00→HOST) : OK

⑧ lseek

동 작	open 명령으로 엽혀져 있는 파일의 포인터를 이동한다.
응 답	OK 또는 ERRxx
설 명	정상적인 명령을 수행하면 "OK"응답 lseek -n:offset open 명령으로 파일이 엽혀진 상태여야 한다.
예 시	엽혀진 파일의 처음 위치로 이동 (HOST→BH-001-00) : lseek -n:0 (BH-001-00→HOST) : OK 엽혀진 파일의 마지막 위치로 이동 (HOST→BH-001-00) : lseek -n: (BH-001-00→HOST) : OK

⑨ put

동 작	open 명령으로 오픈되어 있는 파일에 데이터를 기록한다.
응 답	OK 또는 ERRxx
설 명	정상적인 명령을 수행하면 "OK"응답 put -w:데이터
예 시	1. 데이터 마지막에 CR을 포함하지 않는다. (HOST→BH-001-00) : put -w:1234567890 (BH-001-00→HOST) : OK 2. 데이터 마지막에 CR을 포함한다. (HOST→BH-001-00) : put -w:1234567890 -a (BH-001-00→HOST) : OK 3. CR(줄바꿈)만 기록한다. (HOST→BH-001-00) : put -a (BH-001-00→HOST) : OK

⑩ read

동 작	open 명령으로 오픈되어 있는 파일의 데이터를 읽는다.
응 답	OK 또는 ERRxx
설 명	정상적인 명령을 수행하면 "OK"응답 read -n:number 현재 파일 포인터에서 number 개수만큼 읽는다. number값이 클 경우 파일끝까지만 읽는다. lseek 명령으로 파일 포인터를 이동할 수 있다.
예 시	(HOST→BH-001-00) : read -n:10 (BH-001-00→HOST) : 1234567890OK

⑪ reada

동 작	주어진 파일명 파일의 모든 데이터를 읽는다.
응 답	OK 또는 ERRxx
설 명	정상적인 명령을 수행하면 "OK"응답 reada -f:파일명
예 시	(HOST→BH-001-00) : reada -f:log.txt (BH-001-00→HOST) : OK

⑫ readl

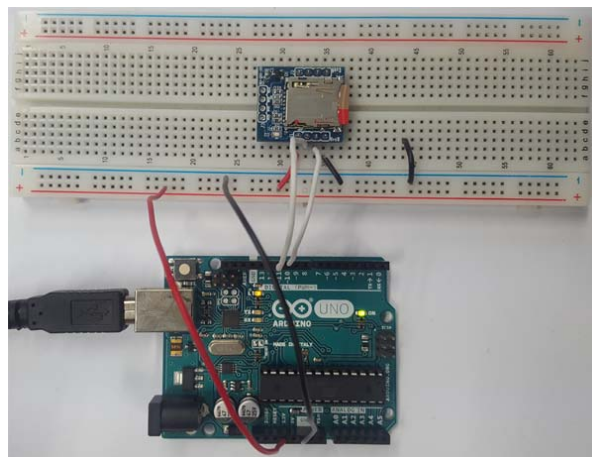
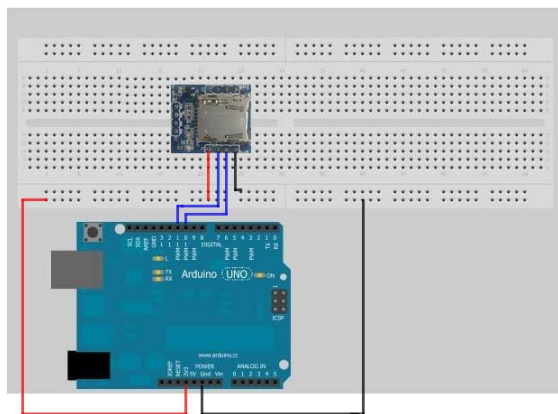
동 작	주어진 파일명 파일의 데이터를 행 단위로 읽는다.
응 답	OK 또는 ERRxx
설 명	정상적인 명령을 수행하면 "OK"응답
예 시	log.txt 파일의 1행부터 2행까지 읽는다. (HOST→BH-001-00) : readl -f:log.txt -s:1 -n:2 (BH-001-00→HOST) : 데이터 OK

7. 샘플 프로그램

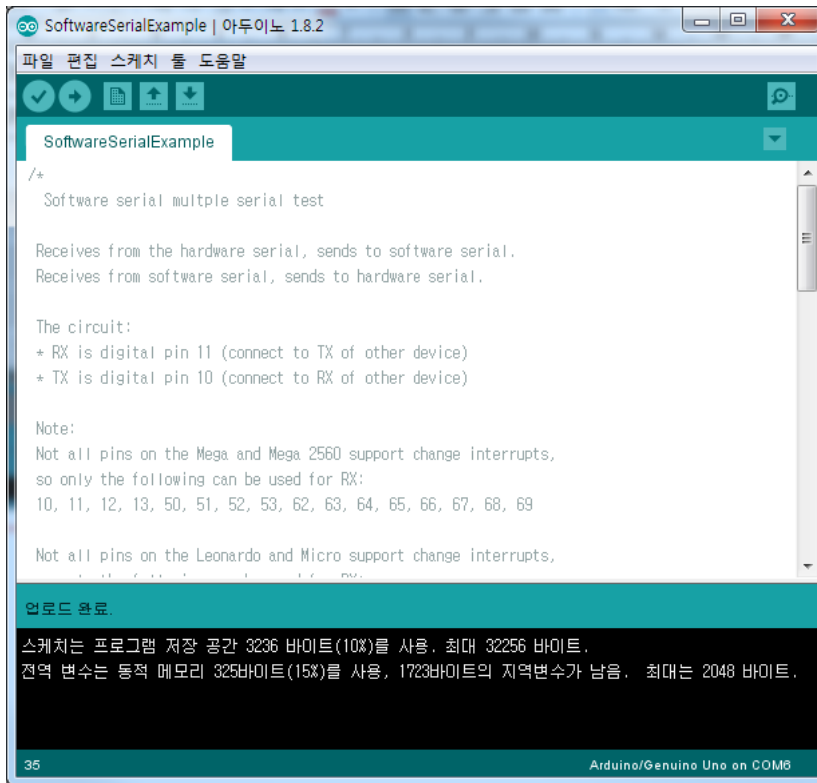
7.1 아두이노 바이패스

▷ 아래 그림과 같이 아두이노 유노 보드와 BH-001-00 장치를 연결한다.

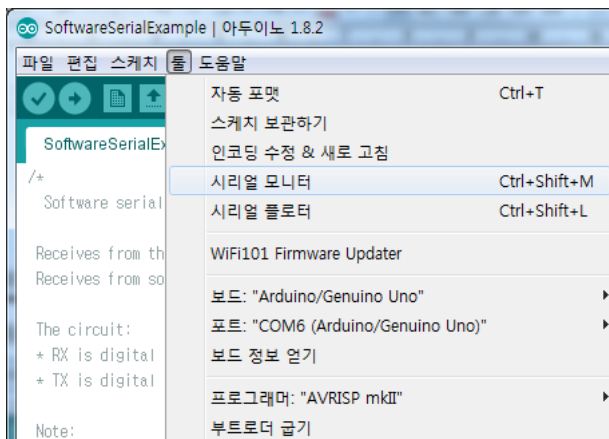
아두이노 유노		BH-001-00	
설 명	번 호	번 호	설 명
3.3V	PWR : 4	J1 : 1	3.3V
RXD	IOH : 10	J1 : 2	TXD
TXD	IOH : 11	J1 : 3	RXD
GND	PWR : 6.7	J1 : 4	GND



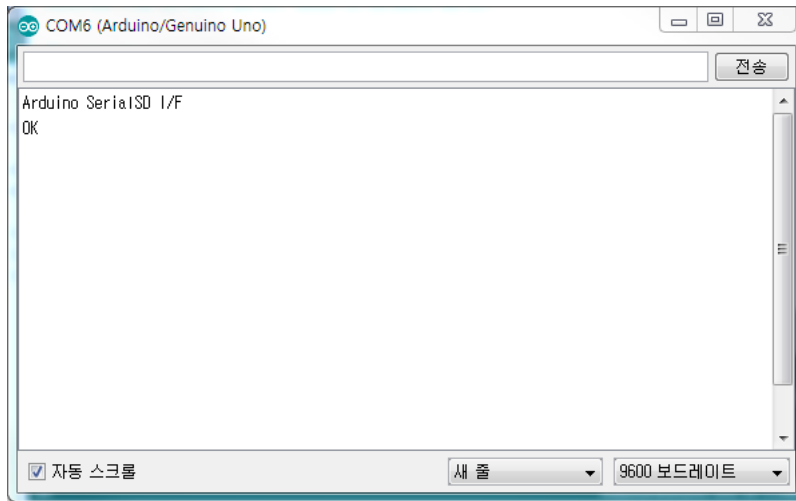
- ▷ 카페에서 제공되는 "softwareSerialExample" 파일을 다운로드하여 아래 그림과 같이 업로드 한다.



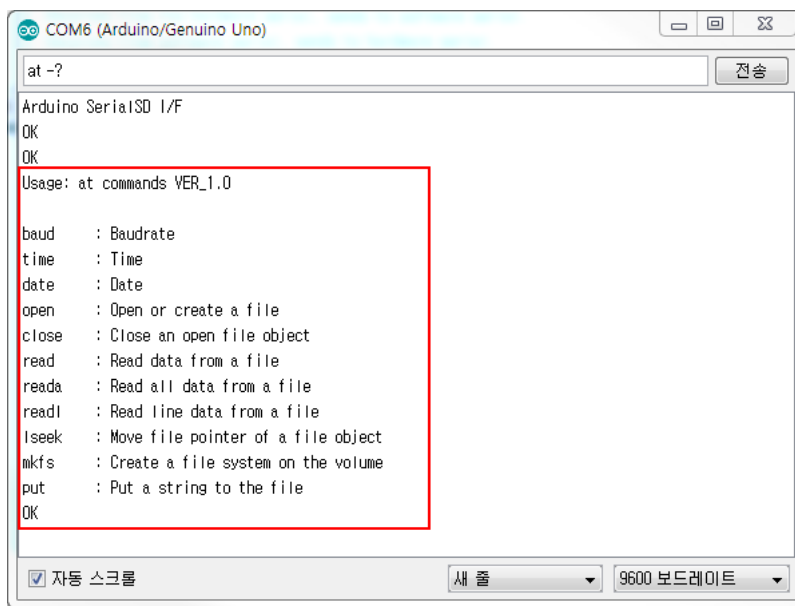
- ▷ 툴 → 시리얼 모니터를 클릭한다.



- ▷ 새 줄, 9600 보레이트 항목을 선택한다.



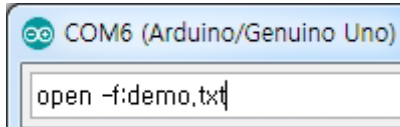
- ▷ “at -?” 명령을 기록하고 전송 버튼을 누르면 아래 그림의 적색 사각형과 같이 명령어에 대한 정보가 표출된다.



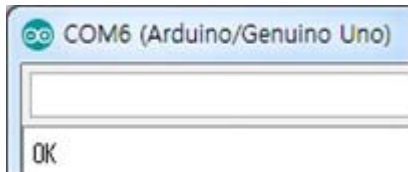
▣ 아두이노 바이패스 기능을 이용한 BH-000-01 기능검증 - 1

- SD Card에 텍스트 파일을 만들기 위해 open 명령어를 사용한다.

명령어 : open -f:demo.txt

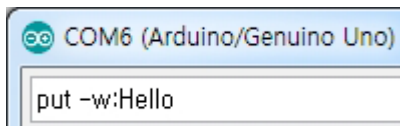


- demo.txt 파일이 성공적으로 만들어지면 명령에 대한 응답으로 OK가 수신된다.



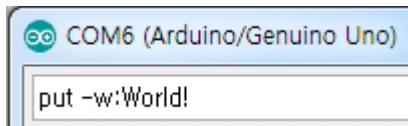
- demo.txt 파일에 내용을 기록하기 위해 put 명령어를 사용한다.

명령어 : put -w:Hollo



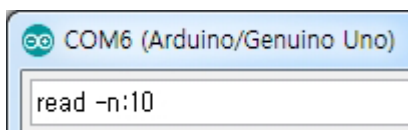
- demo.txt 파일에 연속적으로 내용을 기록하기 위해 put 명령어를 사용한다.

명령어 : put -w:World!



- demo.txt 파일에 기록된 내용을 읽어오기 위해 read 명령어를 사용한다.

명령어 : read -n:10



- 아래 그림과 같은 메시지가 출력된다.

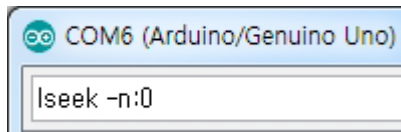
ERR60

아래 표와 같이 현재 포인터의 위치는 12에 위치하고 있다. 12번 이후에 문자가 저장되어 있지 않으므로 에러 메시지를 출력한다.

위치	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
문자	H	e	l	l	o		W	o	r	l	d	!			

- lseek 명령어는 파일 읽기 또는 쓰기 동작 중에 파일 포인터의 위치를 이동합니다.
파일 포인터의 위치를 0번으로 보내기 위해 아래와 같은 명령어를 전송한다.

명령어 : lseek -n:0

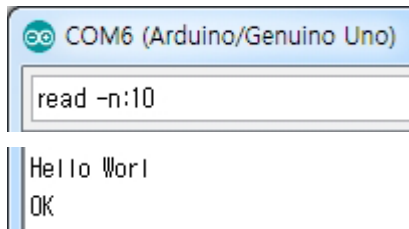


- 문자 포인터 위치 0번으로 이동

위치	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
문자	H	e	l	l	o		W	o	r	l	d	!			

- demo.txt 파일에 기록된 내용을 읽어오기 위해 read 명령어를 사용한다.

명령어 : read -n:10



- 현재 포인터 위치는 아래 표와 같이 9번에 위치하고 있다.

위치	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
문자	H	e	l	l	o		W	o	r	l	d	!			

- 연속적으로 read 명령어를 사용한다.

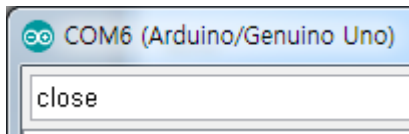
명령어 : read -n:10



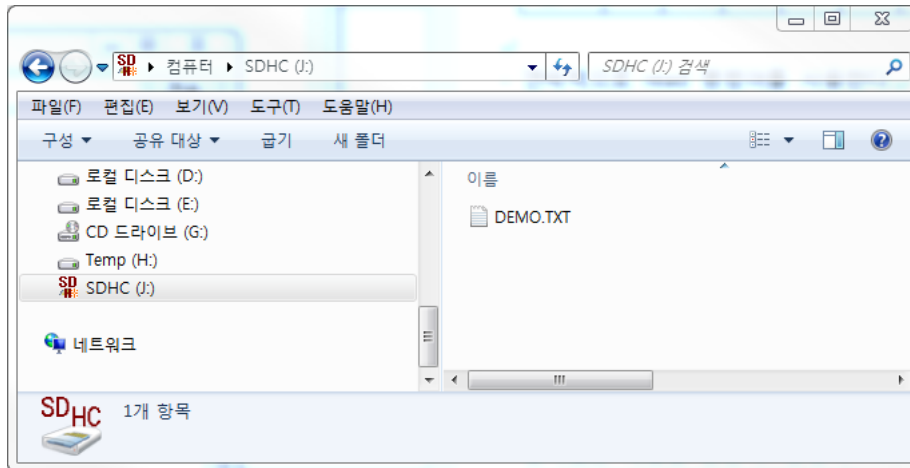
- read 명령의 읽어오는 문자 개수 보다 저장된 문자 개수가 작으므로 9번부터 전체 문자를 읽어오고 포인터는 12번에 위치한다.

위치	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
문자	H	e	l	l	o		W	o	r	l	d	!			

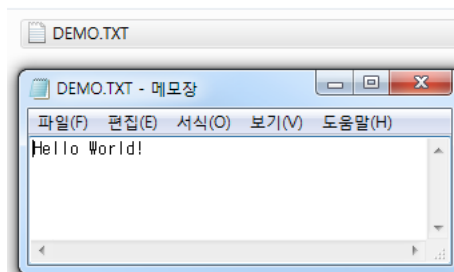
- 열려있는 파일을 닫는다. 명령어 : close



- SD Card를 분리하여 PC와 연결하면 아래 그림과 같이 DEMO.TXT 파일이 생성된 것을 확인할 수 있다.



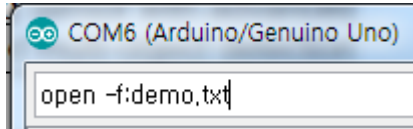
- DEMO.TXT 파일을 내용을 확인하면 아래 그림과 같이 "Hello World!" 내용이 기록되어 있는 것을 확인할 수 있다.



▣ 아두이노 바이패스 기능을 이용한 BH-000-01 기능검증 - 2

- open 명령을 이용하여 기존의 demo.txt 파일을 읽어온다.

명령어 : open -f:demo.txt

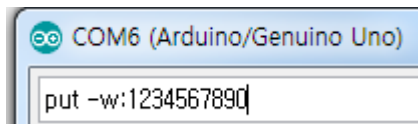


- demo.txt 파일을 읽어왔을 때 12번에 포인터가 위치한다.

위치	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
문자	H	e	l	l	o		W	o	r	l	d	!			

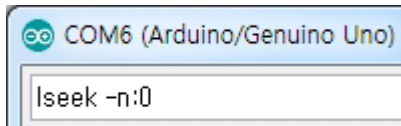
- demo.txt 파일에 내용을 기록하기 위해 put 명령어를 사용한다.

명령어 : put -w:1234567890



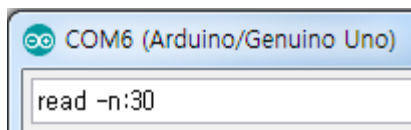
- 파일 포인터의 위치를 0번으로 보내기 위해 아래와 같은 명령어를 전송한다.

명령어 : lseek -n:0



- demo.txt 파일에 기록된 내용을 읽어오기 위해 read 명령어를 사용한다.

명령어 : read -n:30

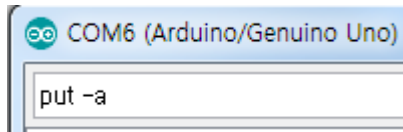


- 아래 그림과 같이 demo.txt 내용에 기록된 내용을 확인할 수 있다.

```
HelloWorld1234567890
OK
```

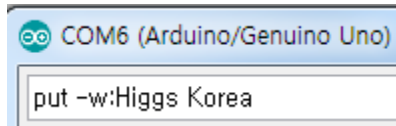
- 라인을 변경한다.

명령어 : `put -a`



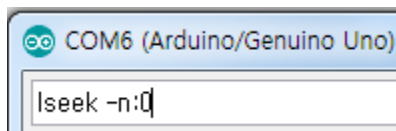
- 변경된 라인에 문자를 기록한다.

명령어 : `put -w:Higgs Korea`



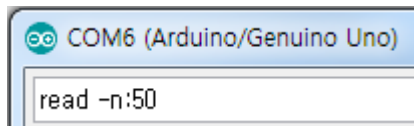
- 파일 포인터의 위치를 0번으로 보내기 위해 아래와 같은 명령어를 전송한다.

명령어 : `lseek -n:0`

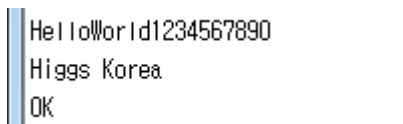


- demo.txt 파일에 기록된 내용을 읽어오기 위해 `read` 명령어를 사용한다.

명령어 : `read -n:35`

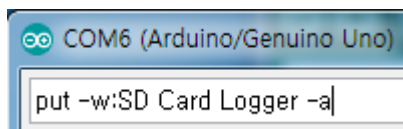


- 아래 그림과 같이 demo.txt 내용에 기록된 내용을 확인할 수 있다.



- 기록하는 내용을 줄을 바꾸기 위해서는 아래와 같은 명령어를 전송한다.

명령어 : `put -w:SD Logger -a`



- 파일 포인터의 위치를 0번으로 이동하고 내용을 읽어오면 아래 그림과 같은 결과를 확인할 수 있다.

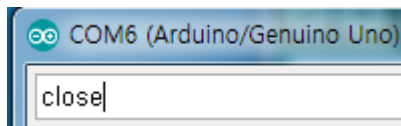
명령어 : lseek -n:0

명령어 : read -n:100

```
HelloWorld!1234567890
Higgs Korea
SD Card Logger
```

- demo.txt 작업하고 있는 상태를 종료하기 위해 close 명령어를 사용한다.

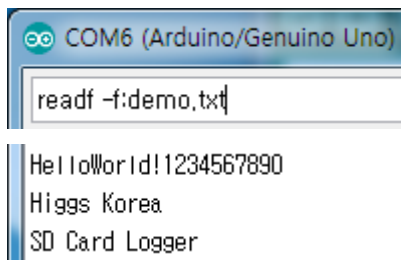
명령어 : close



- demo.txt 전체 내용을 읽어올 때 readf 명령어를 사용한다.

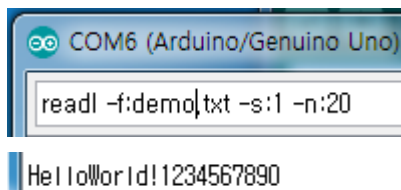
readf, readl 명령어는 반드시 close한 상태에서만 명령어가 동작한다.

명령어 : readf -f:demo.txt



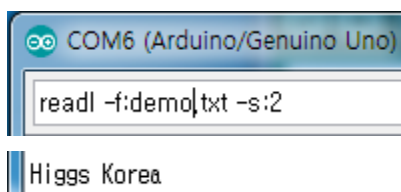
- demo.txt 행별 내용을 읽어올 때 readl 명령어를 사용한다.

명령어 : readl -f:demo.txt s:1 -n:20



- demo.txt 2행 전체를 읽어올 때 readl 명령어를 사용한다.

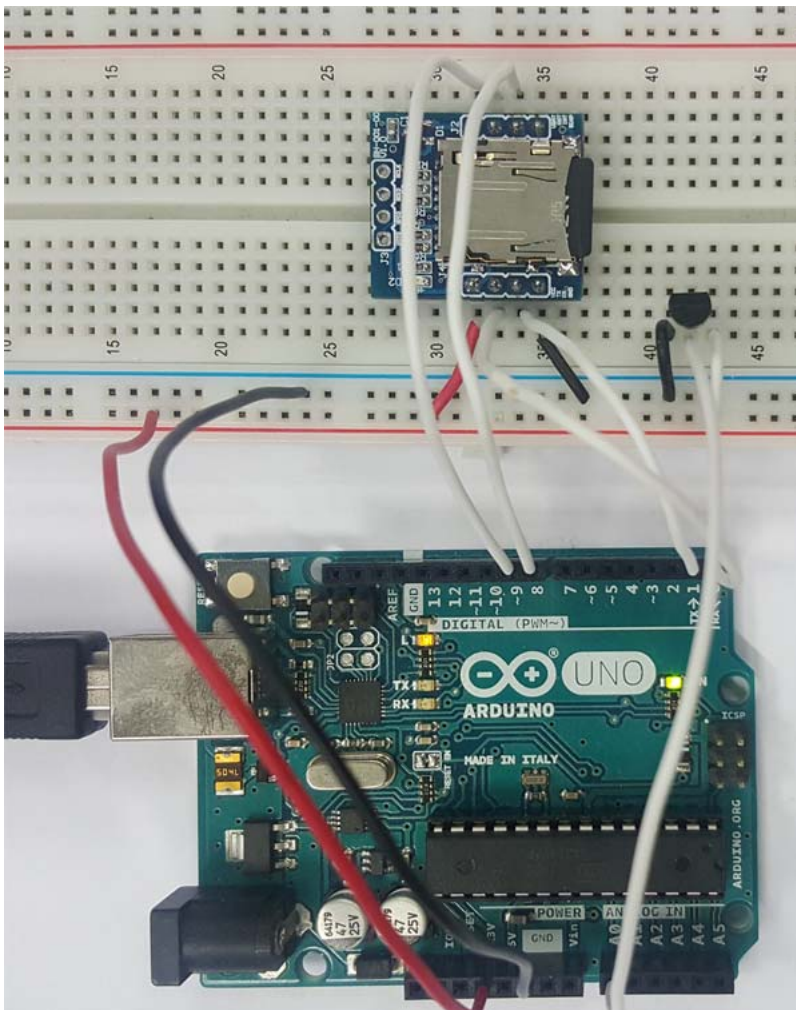
명령어 : readl -f:demo.txt -s:2



7.2 아두이노 온도측정 및 데이터 저장

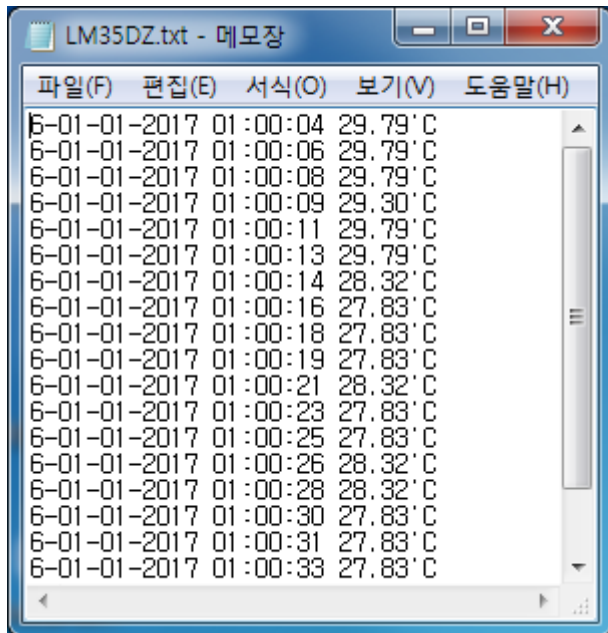
▷ 아래 그림과 같이 아두이노 유노 보드와 BH-001-00 장치를 연결한다.

아두이노 유노		BH-001-00	
설 명	번 호	번 호	설 명
3.3V	PWR : 4	J1 : 1	3.3V
RXD	IOH : 0	J1 : 2	TXD
TXD	IOH : 1	J1 : 3	RXD
GND	PWR : 6.7	J1 : 4	GND
	IOH : 8	J2 : 2	RESET
	IOH : 9	J2 : 3	INT
LM35DZ	AD0		



▷ 카페에서 제공되는 "LD35DZ_UART" 파일을 다운로드하여 실행한다.

▷ SD 카드에 저장된 결과를 확인하면 아래 그림과 같다.



요일	월	일	년	시	분	초	온도
6	01	01	2017	01	00	04	29.79