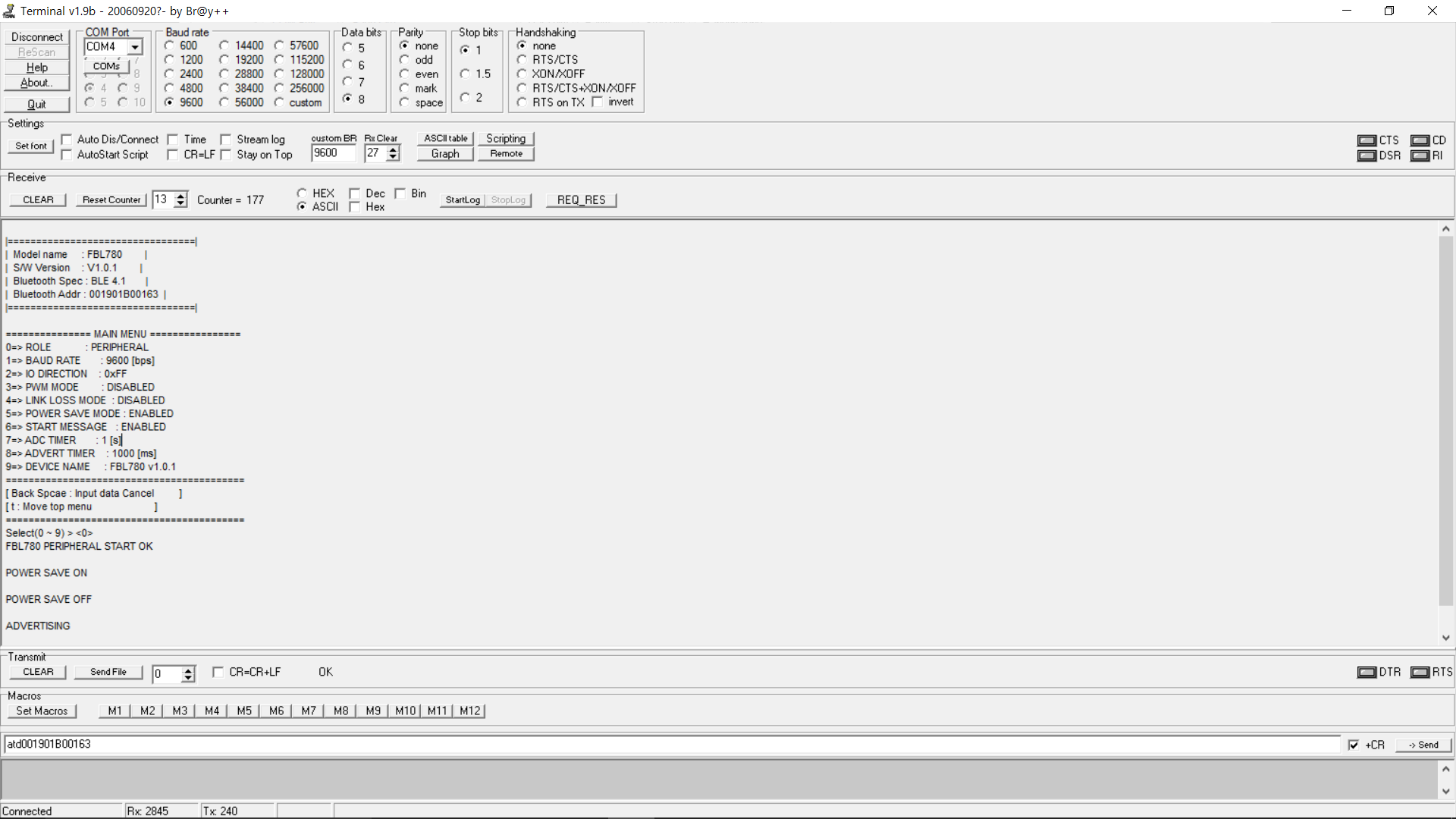
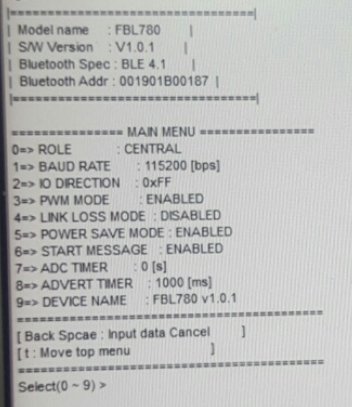
**① Central과 Peripheral 기본 설정**

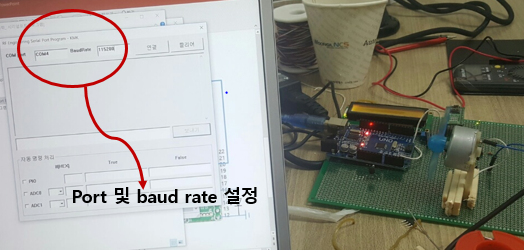


<L : CENTRAL, R : PERIPHERAL>

\* 실시간으로 온도 변화를 체크하기 위해 ADC TIMER 1s 설정했다.

\* led로 peripheral에 전압만 인가되었을 경우와 무선통신 연결되었을 경우 두가지의 상태 변화를 보기 위해 IO DIRECTION을 OxFF로 설정했다.

**② Port 및 Baud rate 설정**



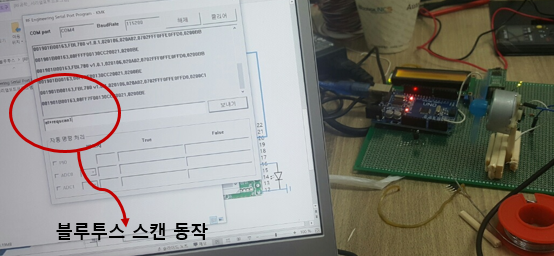
\* Central 인터페이스가 연결된 포트의 넘버를 입력하고 PC와 연결되는 Central의 Baud rate를 115200으로 설정했기 때문에 이와 같이 115200을 Baud rate에 입력한다.

**③Central과 PC 연결**



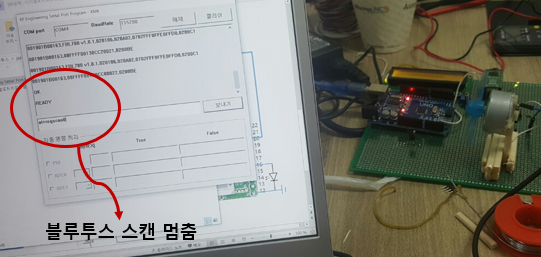
\* Port와 Baud rate를 입력 후 연결 버튼을 누르면 연결 버튼이 해제 버튼으로 변화한다. 이 때 Central 과 PC를 연결시키기 위해 Central 인터페이스 보드의 on/off 스위치를 on에 두면 다음과 같이 Central이 연결 되었다는 창이 출력된다. 이 때 Central의 상태는 POWER SAVE OFF란 것을 볼 수 있다.

**④Peripheral address 스캔**



<Peripheral address 스캐닝 동작 상태>

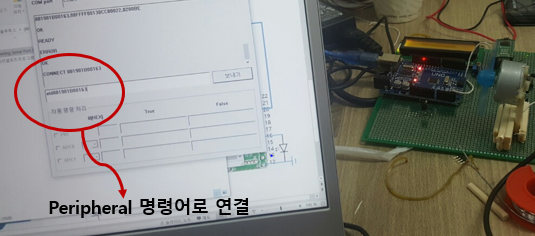
\* Central과 연결하고자 하는 Peripheral의 address를 얻기 위해 스캔을 한다. 이 때 스캔 명령어 AT+REQSCAN1을 입력하면 Peripheral의 정보가 반복적으로 출력된다.



<Peripheral address 스캐닝 정지 상태>

\* 스캐닝 중지 명령어 AT+REQSCAN0을 입력하면 스캐닝이 멈추고 출력 값의 맨 처음에서 Peripheral의 address을 읽을 수 있다.

**⑤ Central과 Peripheral의 연결**



\* Central과 Peripheral을 연결시켜 주기 위해 앞에서 읽어온 Peripheral의 address를 이용하여 ATD(Peripheral address) 명령어를 입력하면 다음과 같이 CONNECT 됨을 알 수 있다.

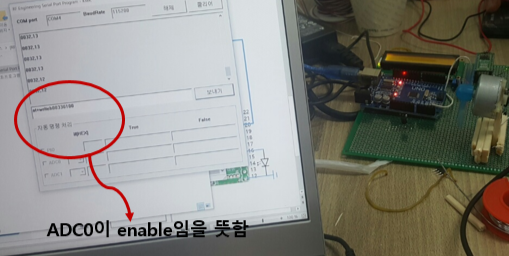
\* 무선 통신 연결이 된 것을 프로그램 말고 회로 자체에서 LED 등으로 확인 가능하다. ④번 과정에서 보면 무선 통신 연결이 되기 전에는 LED등이 켜져 있고 무선 통신 연결을 하면 꺼짐을 볼 수 있다. 이는 LED 단자가 Peripheral의 advertising status와 gnd와 연결되어 있는데 무선 통신 연결 전에는 아두이노로부터 가해지는 3.3V에 의해 불이 들어온다. 하지만 무선 통신 연결을 하게 된다면 모든 단자에 3.3V가 인가되어 아두이노에 의해 가해지는 3.3V와 전위차가 0V가 되어 꺼지게 된다. 만약 이를 반대로 동작시키고 싶다면 gnd대신 Vcc단자에 연결하면 가능하지만 전원만 연결했을 때와 무선 통신 연결을 했을 때의 확인을 위해 전자 방법을 사용했다.

**⑥ Central 수신**



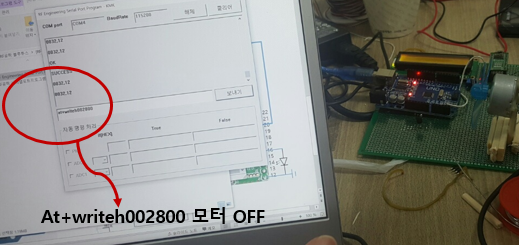
\* 센서의 값을 읽어 들이기 위해 AT+READ0033의 명령어를 사용했다. 이는 센서를 아날로그 신호를 출력하는 LM35 온도센서를 사용했기 때문에 ADC포트를 사용했고 또한 이 센서를 ADC 0를 INPUT으로 사용했기 때문에 다음의 명령어를 사용했다. 이 의미는 0033은 ADC 포트의 핸들 값이다. 이 때 0000이라는 숫자가 출력되는데 이는 ADC 0 포트가 Disable 상태라는 것을 알 수 있다.

**⑦ Central 송신**



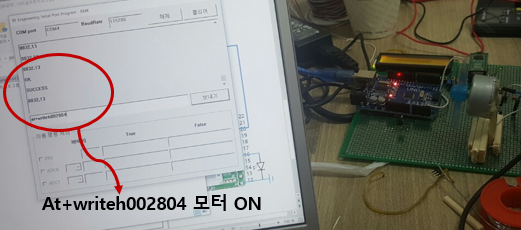
\* 이제 받은 센서 값을 PC로 보내기 위한 동작이다. 이 때 AT+WRITEH00330100 명령어를 사용했는데 이 의미는 0033은 앞에서와 같이 ADC 포트의 핸들 값이고, 0100은 ADC 0 포트를 Enable로 하겠다 라는 뜻이다. 이를 종합적으로 해석해보면 핸들 0033에 0100이라는 데이터를 HEX값으로 송신한다는 뜻이다. 출력 창을 확인하면 온도센서가 감지한 온도를 디지털 HEX타입의 값으로 쉼표 뒤에 두 자리 수로 나타남을 볼 수 있다.

**⑧ AT+WRITEH002800 모터 OFF**



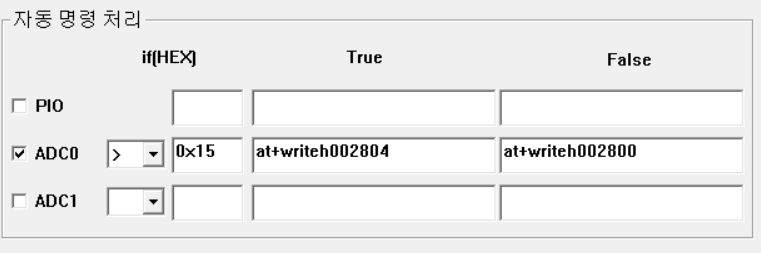
\* 이번 과정은 PIO 3 포트에 모터를 설계하여 이를 제어하는 것임으로 PIO 3 포트를 제어하는 과정이다. Central과 Peripheral의 기본 설정 중 IO DIRECTION을 0Xff로 설정하였다. 이는 기본 설정이 OUTPUT으로 설정되어 있다는 것임으로 5V가 PIO단에 출력되어 아무 명령어 주지 않은 상태, 즉, 기본 상태에는 선풍기가 작동되고 있는다. 이 때 AT+WRITEH002800이란 명령어를 사용하면 PIO 3 포트를 제어해서 선풍기의 작동을 제어 할 수 있다.

**⑨ AT+WRITEH002804 모터 ON**

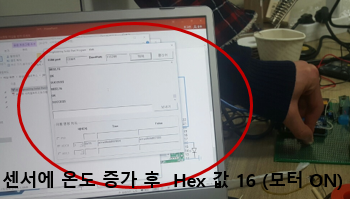
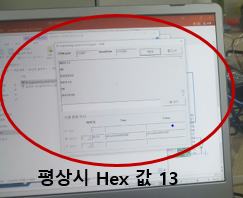


\* 반대로 다시 선풍기를 키기 위해 AT+WRITEH002804 명령어를 사용하여 PIO 3 포트를 ON 시킨다. 이때 AT+WRITEH002804의 맨 뒤의 2자리 숫자는 PIO 3 포트를 의미한다.

**⑩ 매크로 설정**



<조건부 설정>



<L : 실내온도, R: 손으로 온도 증가>

\* 모터를 ON/OFF 시킨 명령어를 사용해 매크로를 설정하여 기준 온도를 유지시키도록 한다. 실용도는 정부에서 권하는 권장 온도로 자동 유지되는 것이지만 구현하여 작동이 되는 것을 보여주기 위해 실내온도를 기준으로 작동한다. 실내온도가 Hex값으로 13으로 나오기 때문에 이보다 큰 15에서 보다 크면 선풍기가 작동하고 15보다 작으면 선풍기가 작동되지 않도록 설계하였다.

**⑪ 작동 영상**



\* 선풍기가 돌아가고 꺼짐이 사진으로는 잘 보이지 않아 동영상으로 촬영한 것이다.