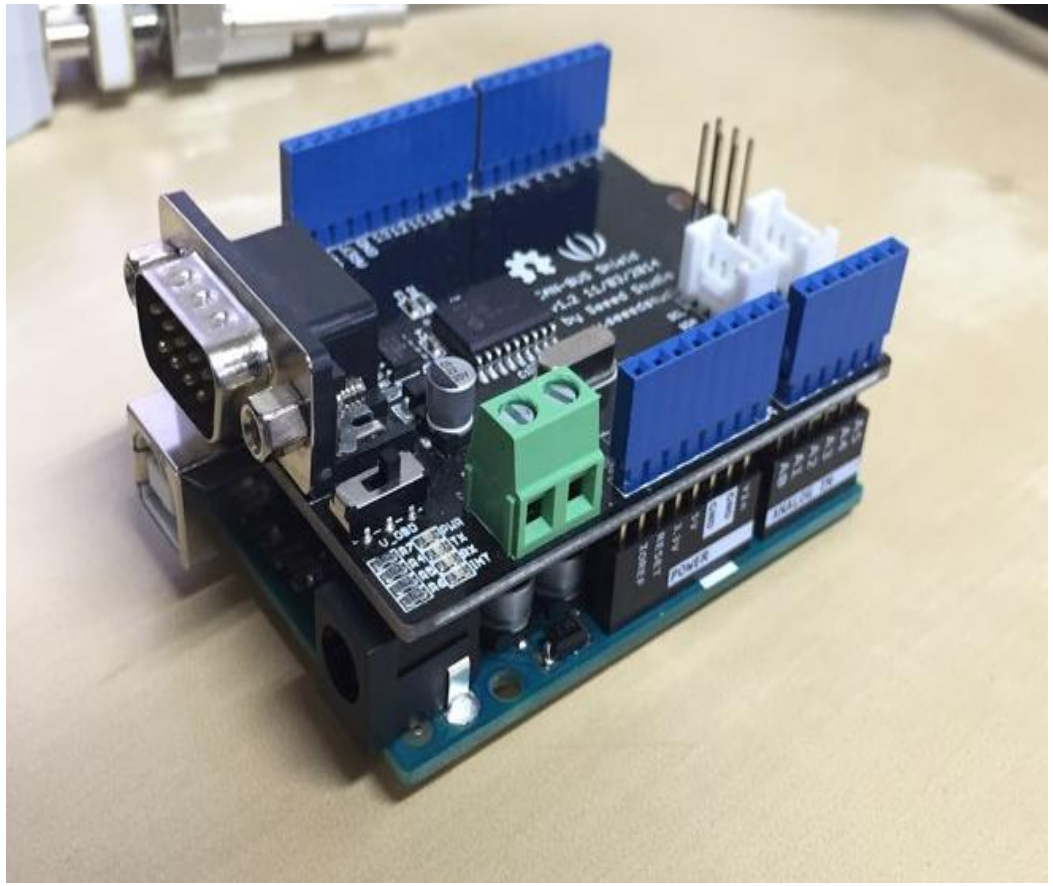
# How to use CAN Bus Shield

**Avees Lab**

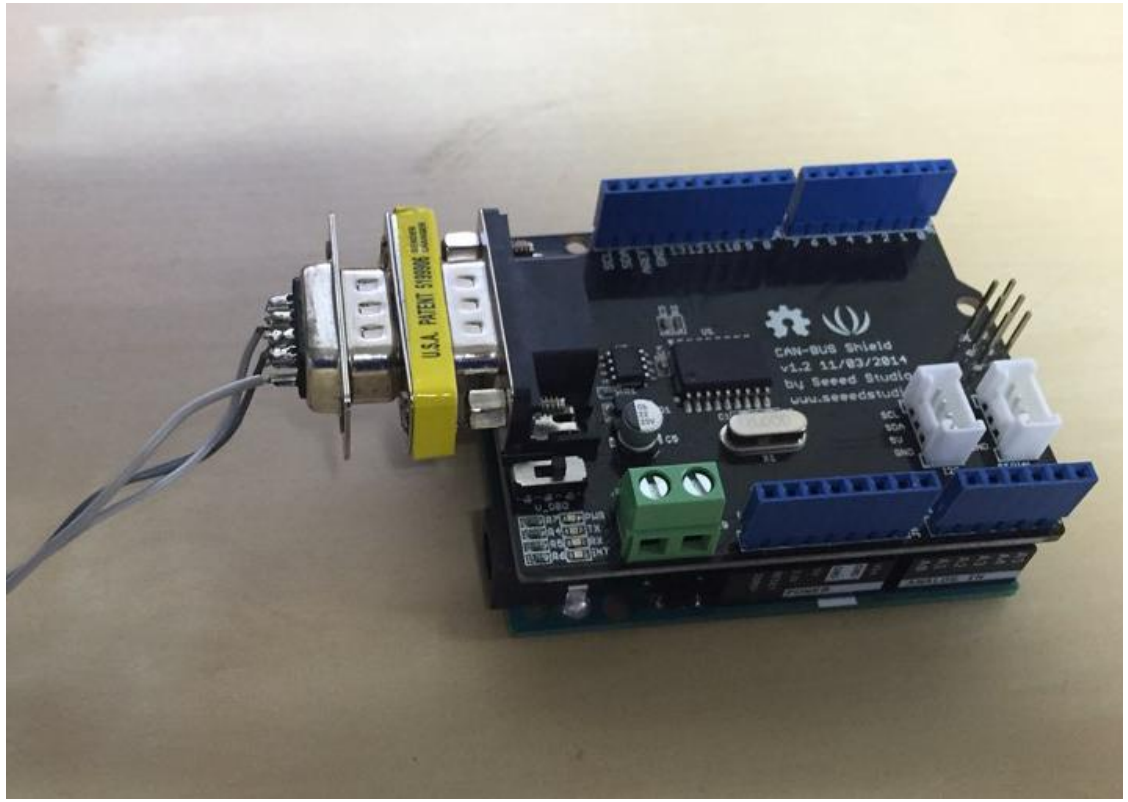
# 장비 사용법

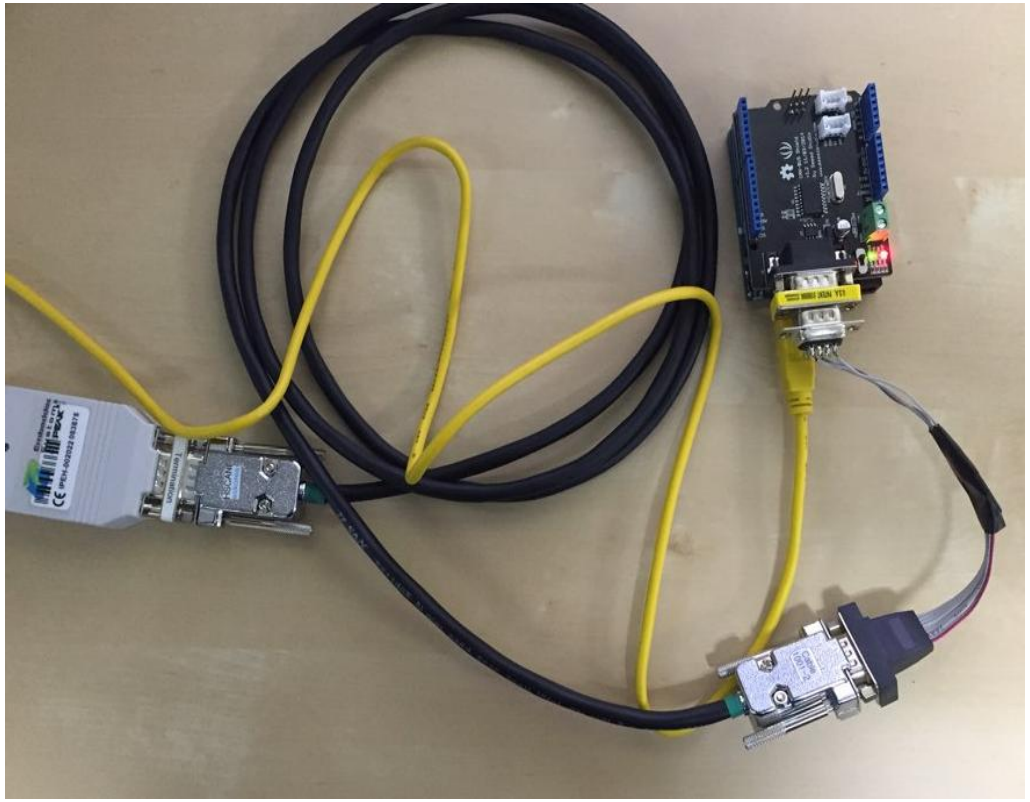
CAN Bus Shield를 그림과 같이 아두이노위에 장착 (핀배열에 유의할 것)
PCAN USB에 Termination을 장착



# 장비 사용법

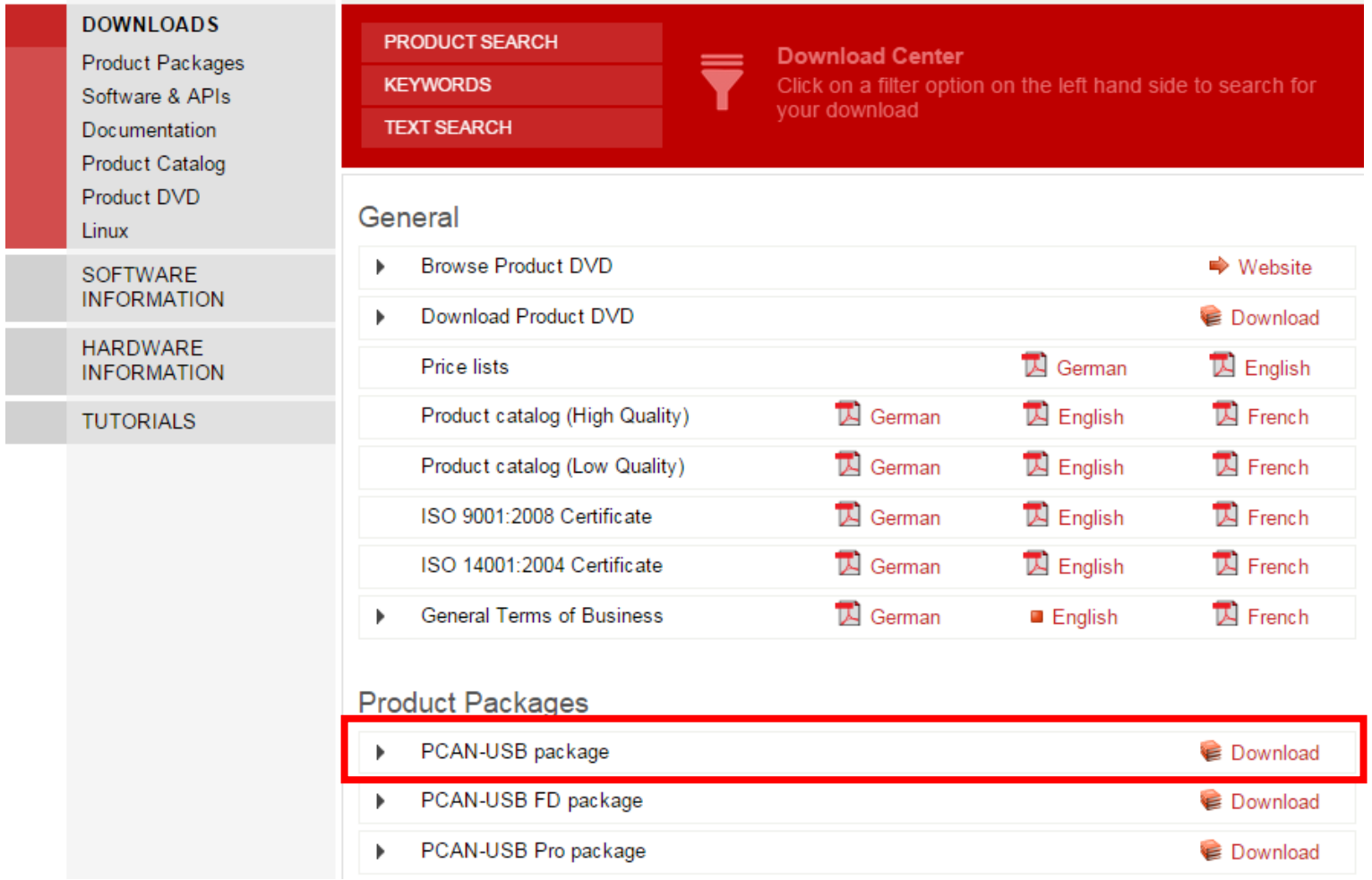
노란색 핀 커넥터가 붙어있는 부분을 CAN Bus Shield에 부착
CAN BUS Shield와 PCAN USB 양쪽에 케이블을 연결 후 사용



# PCAN USB Driver 설치(1/5)

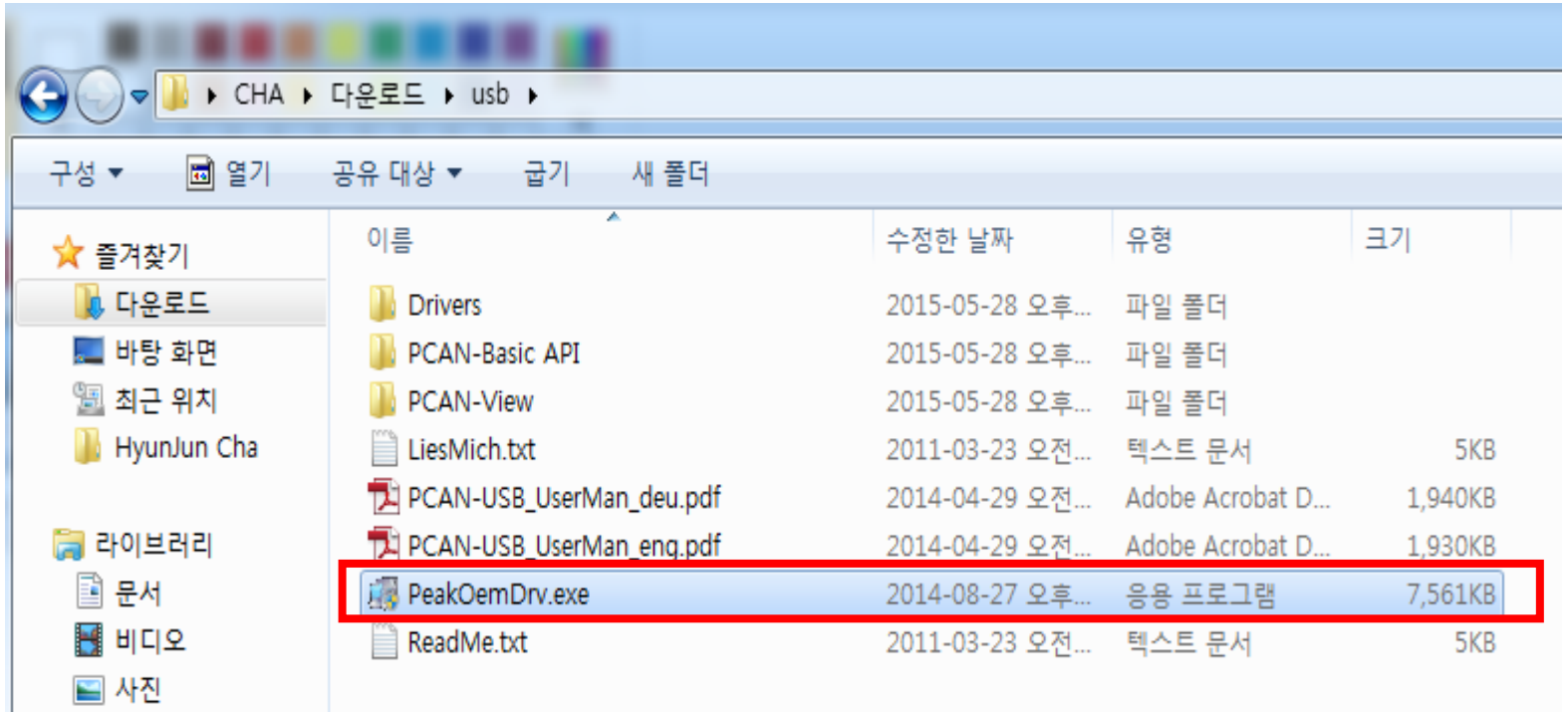
http://www.peak-system.com/Downloads.76.0.html?&L=1
접속 후 PCAN-USB Package 다운로드




**Download 클릭 후 다운로드**

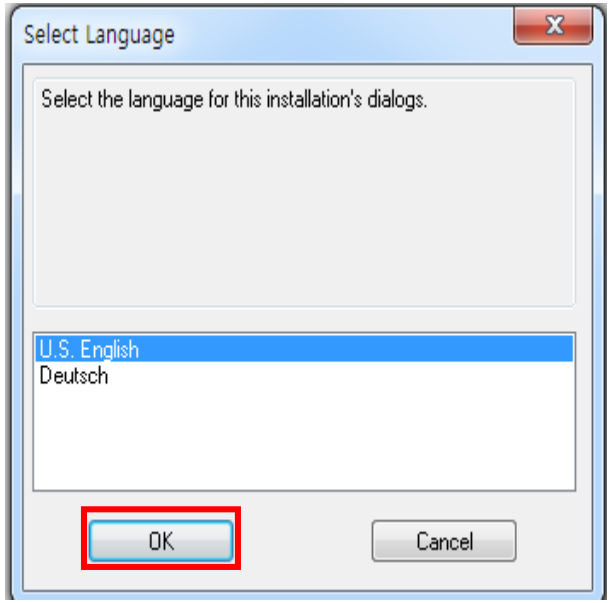
# PCAN USB Driver 설치(2/5)

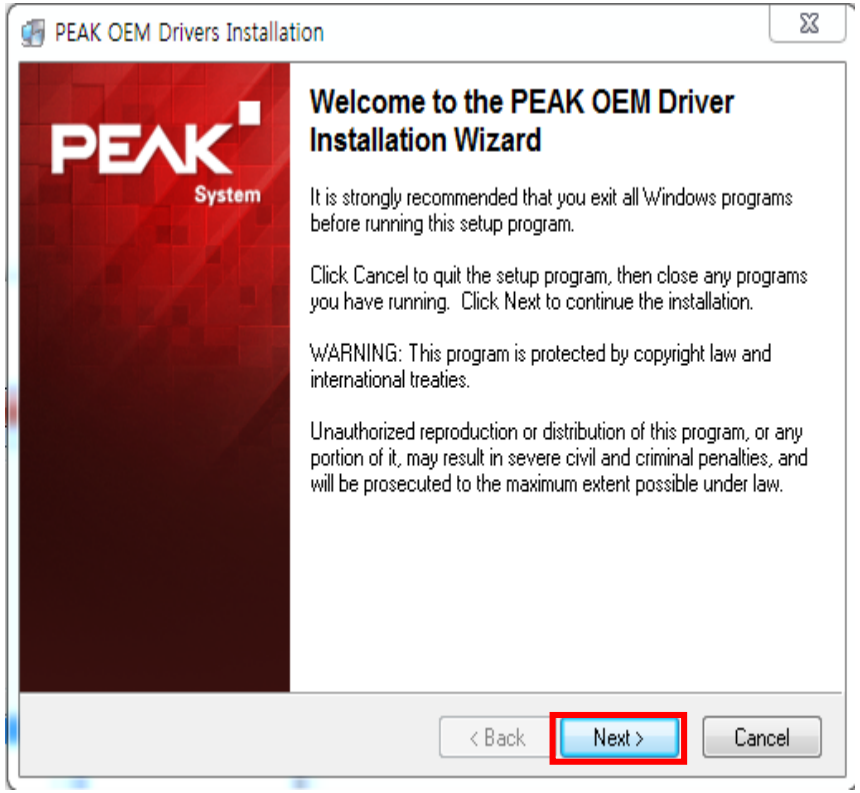
압축 해제 후 PeakOemDrv.exe 실행

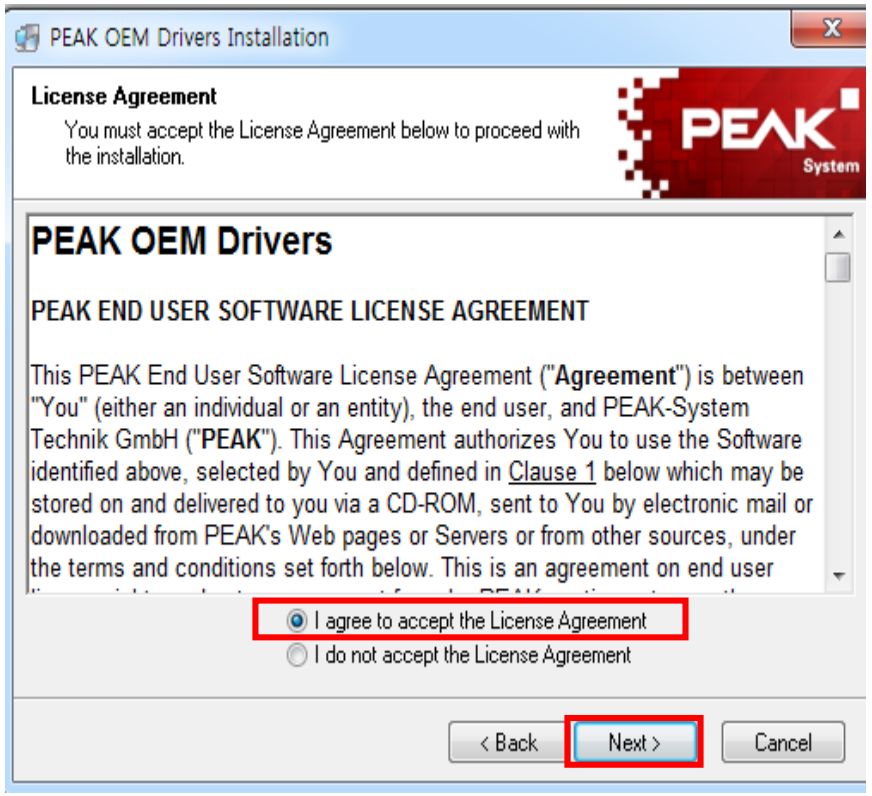


# PCAN USB Driver 설치(3/5)

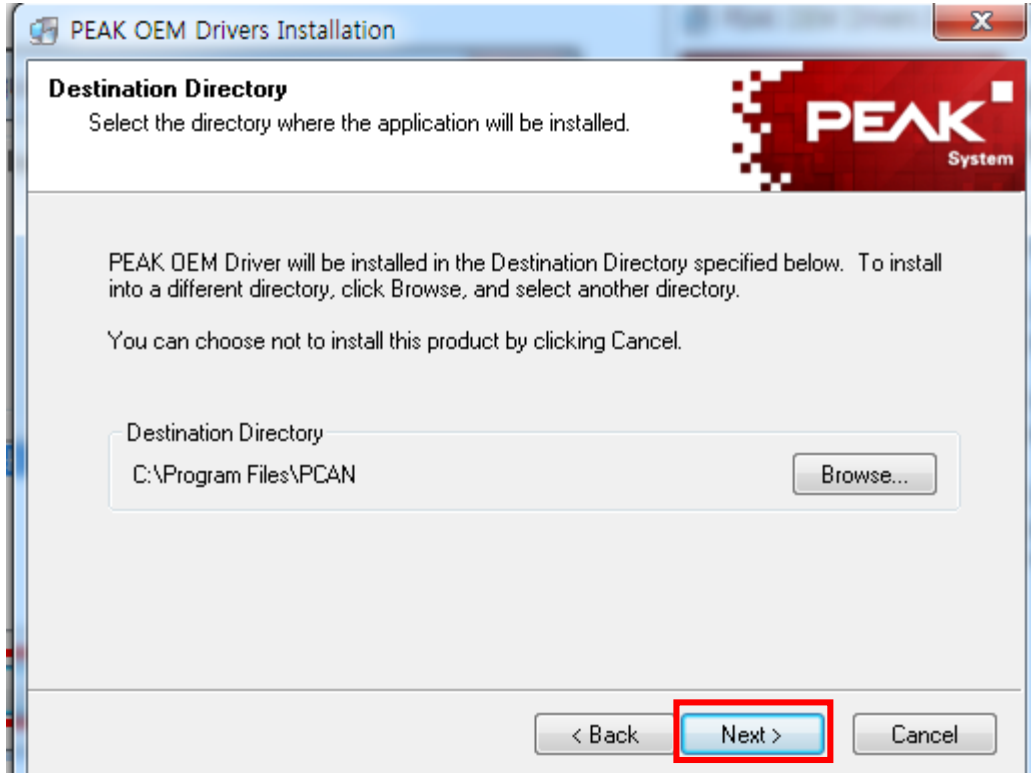
U.S English 설정 후 OK
그림 순서대로 Next를 클릭하여 설치 진행



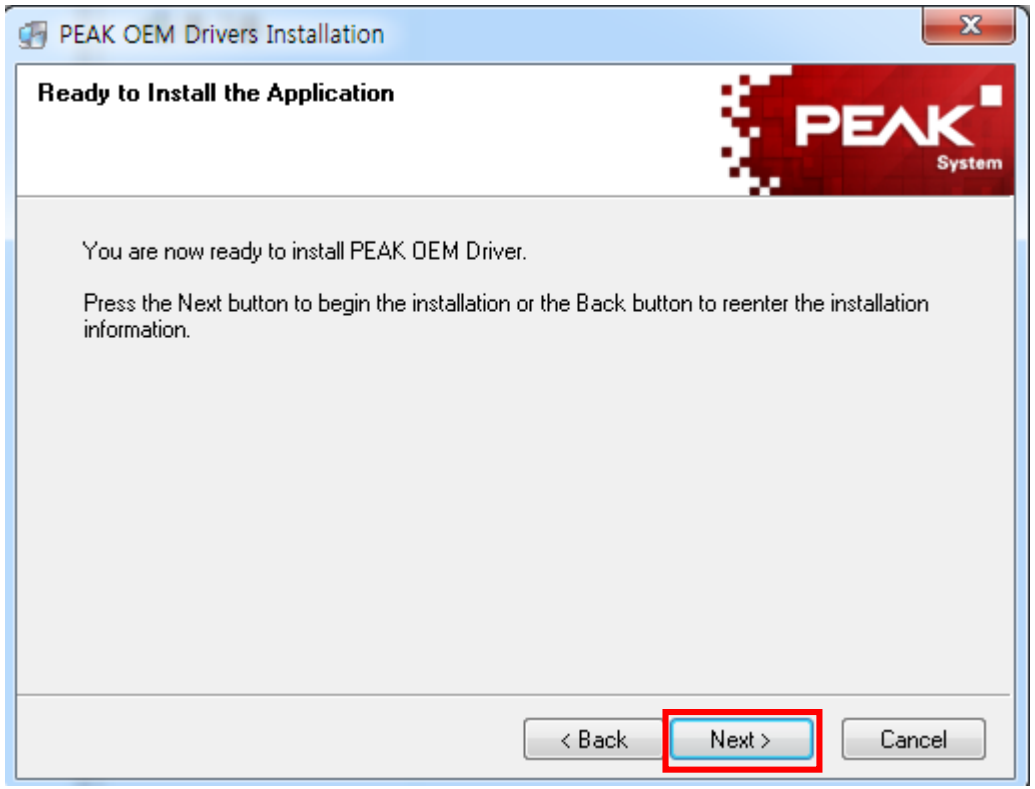



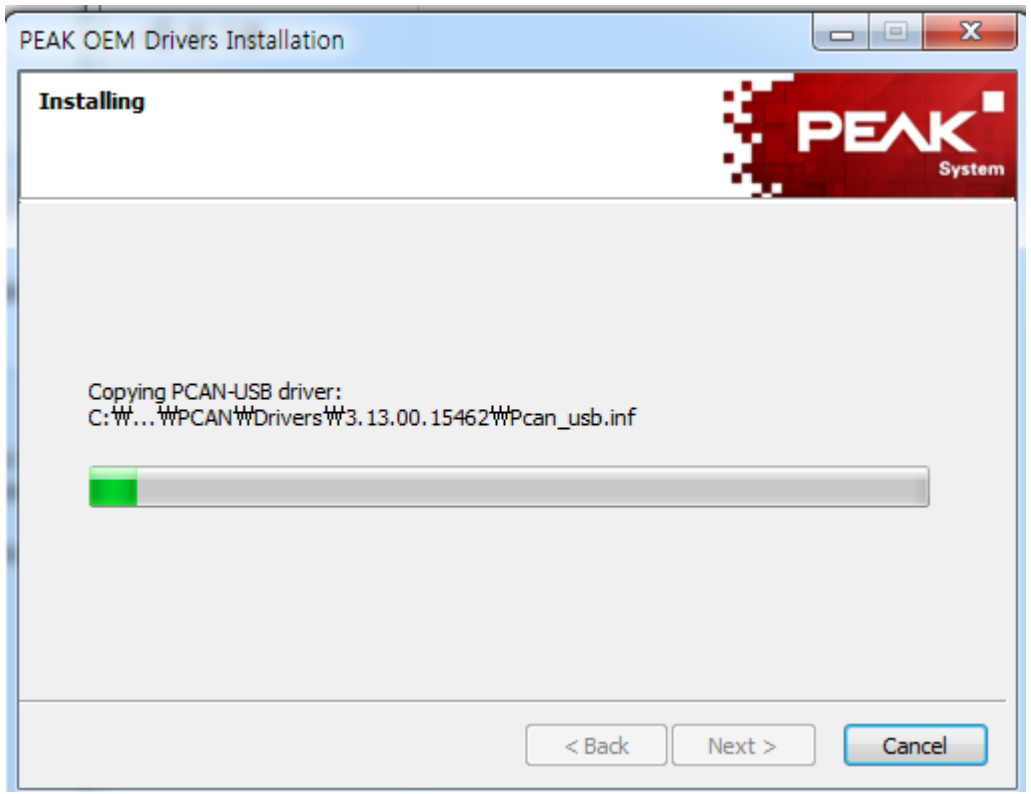
# PCAN USB Driver 설치(4/5)




# PCAN USB Driver 설치(5/5)

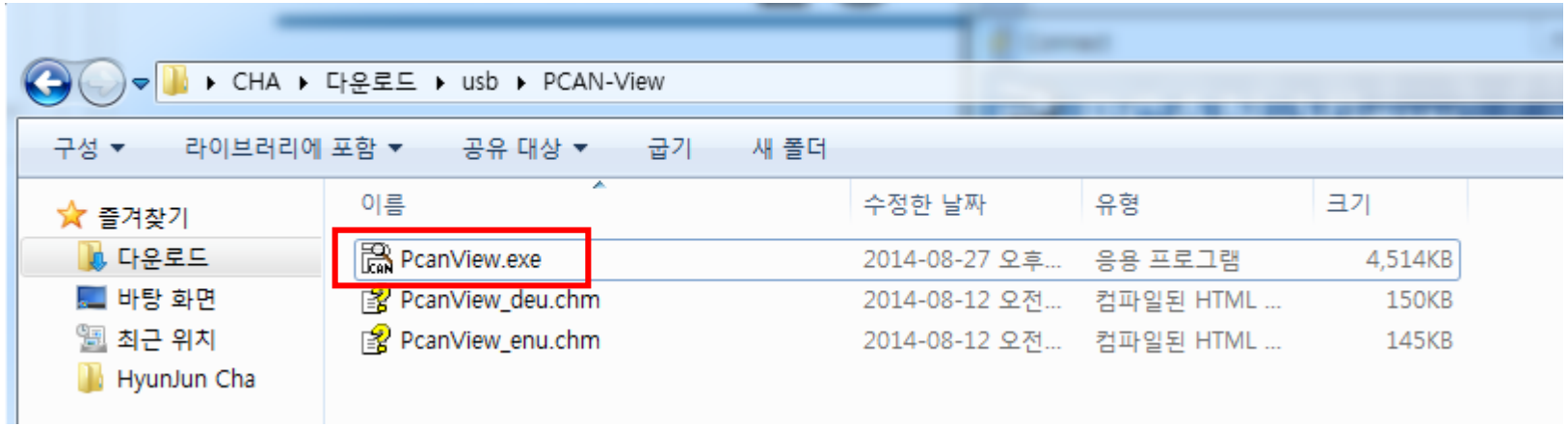
Next를 누른 후 설치 진행
설치가 마무리 된 후 컴퓨터 재부팅



# PCAN View 실행

PCAN-View -> PcanView.exe 실행



# libcfg.mk 파일 수정

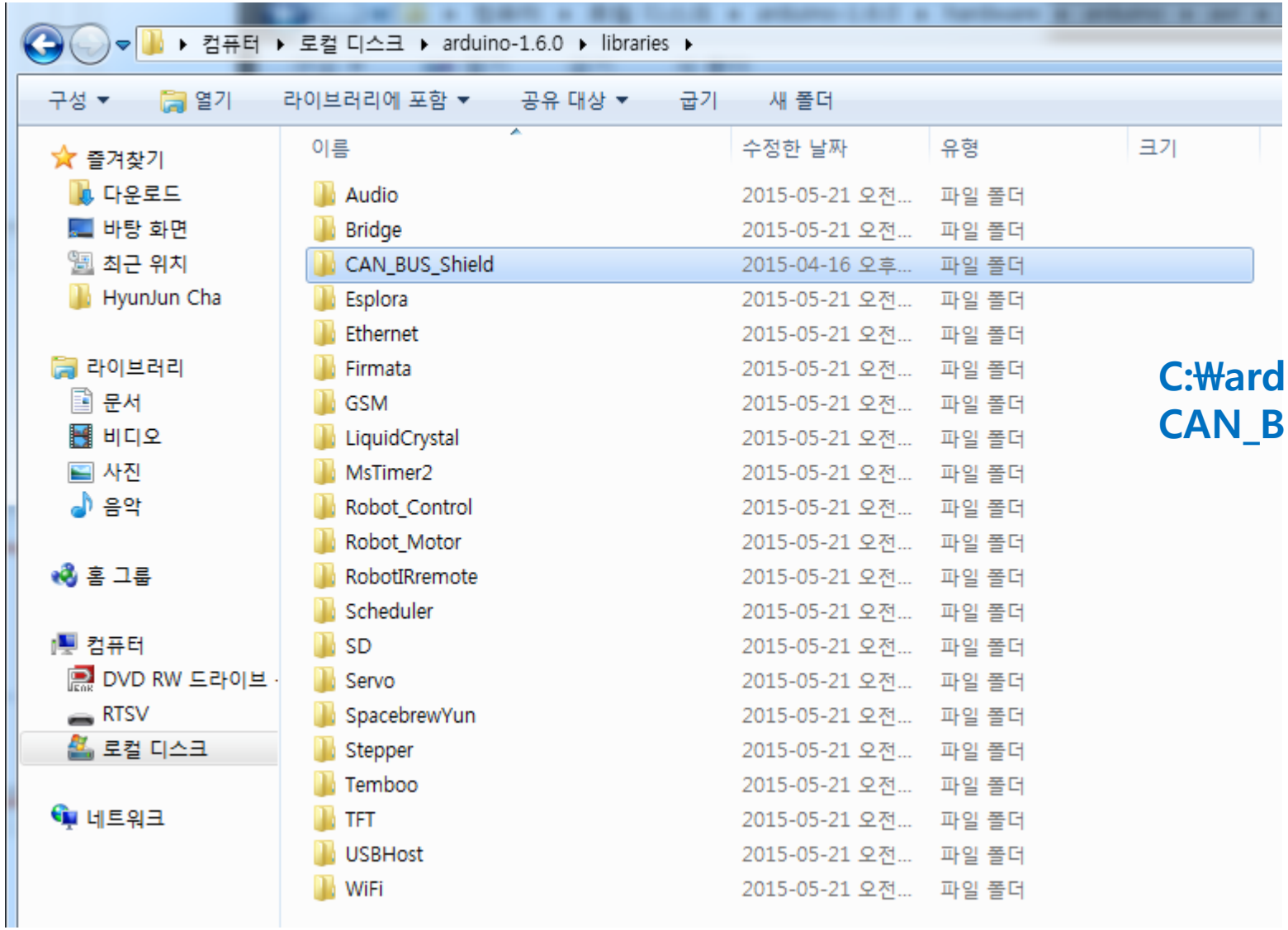
C:₩EE_RT-Druid-2.4.0-juno-win32-x86_64₩
eclipse₩plugins₩com.eu.evidence.ee_2.4.0.20141105_0021₩
ee_files₩contrib₩arduino₩cfg 밑에 있는 **libcfg.mk** 파일 수정

```
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/wiring.c
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/wiring_digital.c
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/wiring_analog.c
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/wiring_pulse.c
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/wiring_shift.c
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/CDC.cpp
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/HardwareSerial.cpp
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/HID.c
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/IPAddress.cpp
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/new.c
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/Print.cpp
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/Stream.cpp
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/Tone.cpp
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/USBCore.cpp
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/WMath.cpp
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/WString.cpp
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/hooks.c
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/WInterrupts.c
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/abi.cpp
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/HardwareSerial0.cpp
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/HardwareSerial1.cpp
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/HardwareSerial2.cpp
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/cores/arduino/HardwareSerial3.cpp
```

박스안에 있는 내용을 추가 (오타주의!)

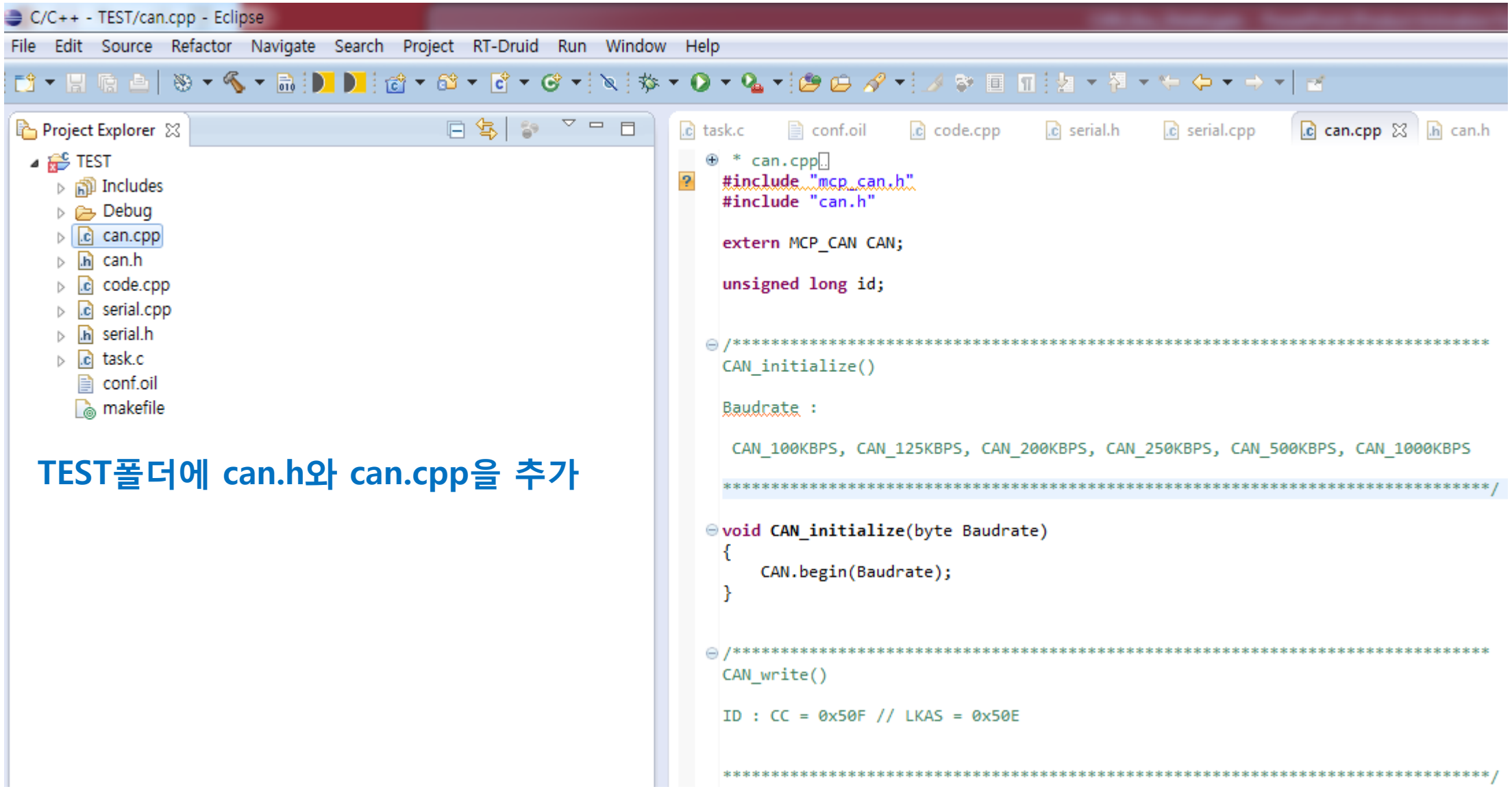
```
INCLUDE_PATH := $(ARDUINO_SDK_ROOT)/libraries/CAN_BUS_Shield        $(INCLUDE_PATH)
EE_SRCS_ARDUINO_SDK += libraries/CAN_BUS_Shield/mcp_can.cpp

INCLUDE_PATH := $(ARDUINO_SDK_ROOT)/hardware/arduino/avr/libraries/SPI      $(INCLUDE_PATH)
EE_SRCS_ARDUINO_SDK += hardware/arduino/avr/libraries/SPI/SPI.cpp
```

# CAN_BUS_Shield 라이브러리 복사



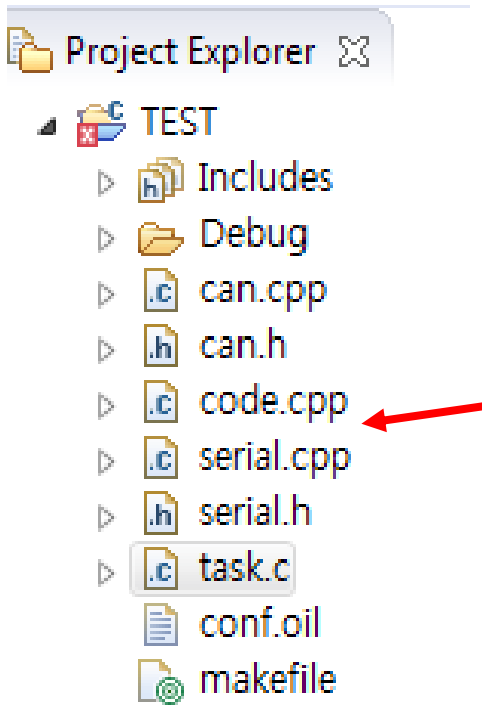

C:₩arduino-1.6.0₩libraries 경로에
CAN_BUS_Shield 폴더 복사

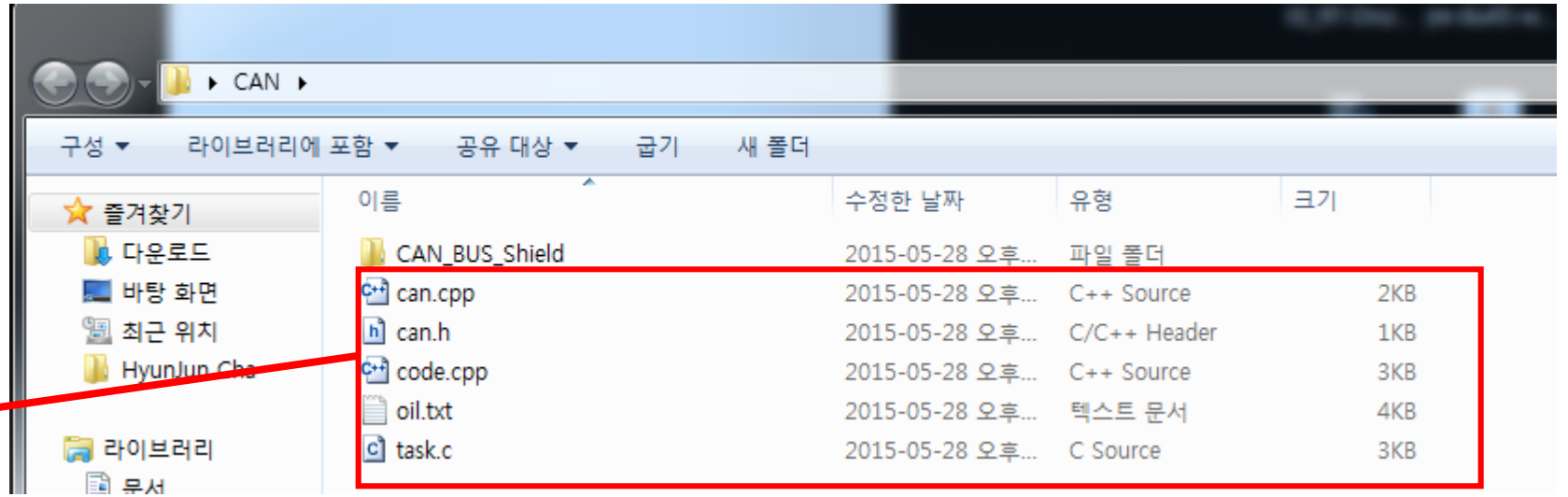
# Eclipse 실행 후 파일 추가




TEST폴더에 can.h와 can.cpp을 추가

# code.cpp, conf,oil, task.c 파일 수정




**기존에 있는 code.cpp / conf.oil / task.c / can.h / can.cpp 을 가상대학에서 다운 받은 코드 내용으로 수정 or Copy & Paste**

# CAN Message 전송

can.cpp에 정의되어 있는 CAN_write 함수를 사용
CAN_write(ID, Data Length, Buffer)

*task.c | conf.oil | code.cpp | serial.h | serial.cpp | can.cpp | can.h

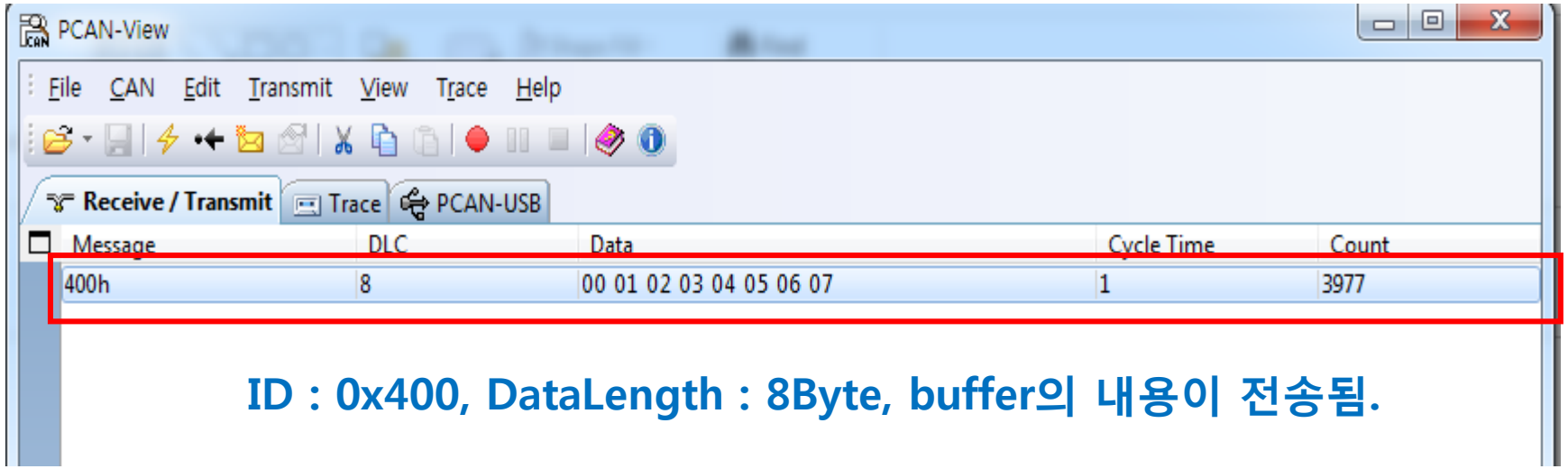
```
/* ###*B*###

#include "ee.h"
#include "Arduino.h"
#include "can.h"


unsigned char len;
unsigned char buf[8];
unsigned char buffer[8] = {0, 1, 2, 3, 4, 5, 6, 7};
unsigned char temp[8] = {10, 11, 12, 13, 14, 15, 0 , 1};
extern unsigned long id;


ISR(isr) {
    static int count = 1;
    if (count == 4) {    // Increment Counter every 1 second.
        IncrementCounter(SysTimerCnt);
        count = 0;

    }
    count++;


}

TASK(TaskL1) {

    CAN_write(0x400, 8, buffer);  // send CAN Message


    TerminateTask();

};
```

**Build 후 업로드!!**

**ID : 0x400, DataLength : 8Byte, buffer의 내용을 Send**

# CAN Message 전송 확인

PCAN View 실행 후 화면 확인



# CAN Message 수신

CAN_read(&len, buf) len: 수신된 Data의 길이가 저장, buf에는 수신된 Data가 저장

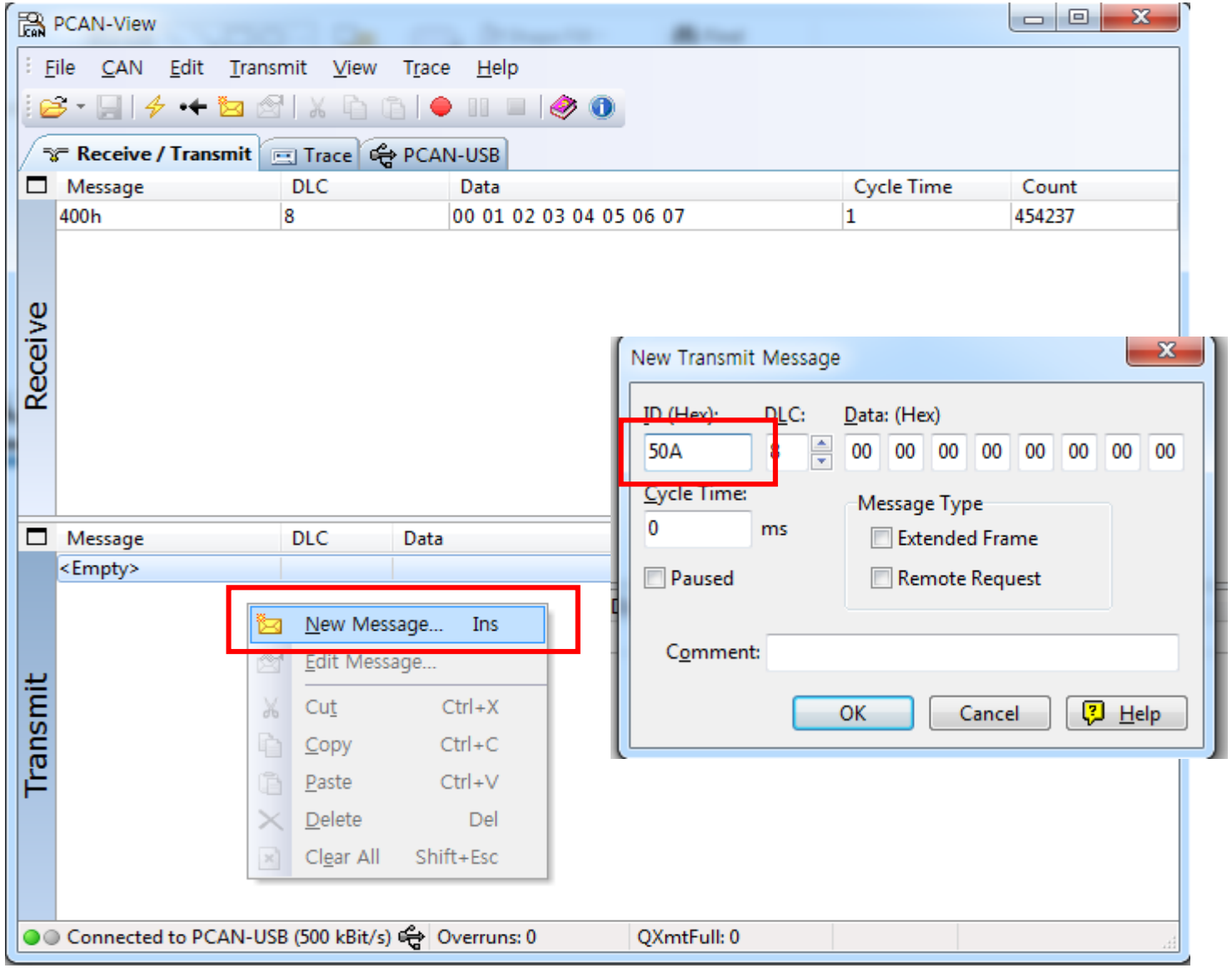
*task.c | conf.oil | code.cpp | serial.h | serial.cpp | can.cpp | can.h

```
unsigned char len;
unsigned char buf[8];
unsigned char buffer[8] = {0, 1, 2, 3, 4, 5, 6, 7};
unsigned char temp[8] = {10, 11, 12, 13, 14, 15, 0 , 1};
extern unsigned long id;


ISR(isr) {
    static int count = 1;
    if (count == 4) {    // Increment Counter every 1 second.
        IncrementCounter(SysTimerCnt);
        count = 0;

    }
    count++;


}

TASK(TaskL1) {


    CAN_write(0x400, 8, buffer); // send CAN Message


    TerminateTask();

};


TASK(TaskL2) {

    CAN_read(&len, buf);
    if (id == 0x50A)
    {
        CAN_write(0x300, 8, temp);
    }

    TerminateTask();
};
```

**Build 후 업로드!!**

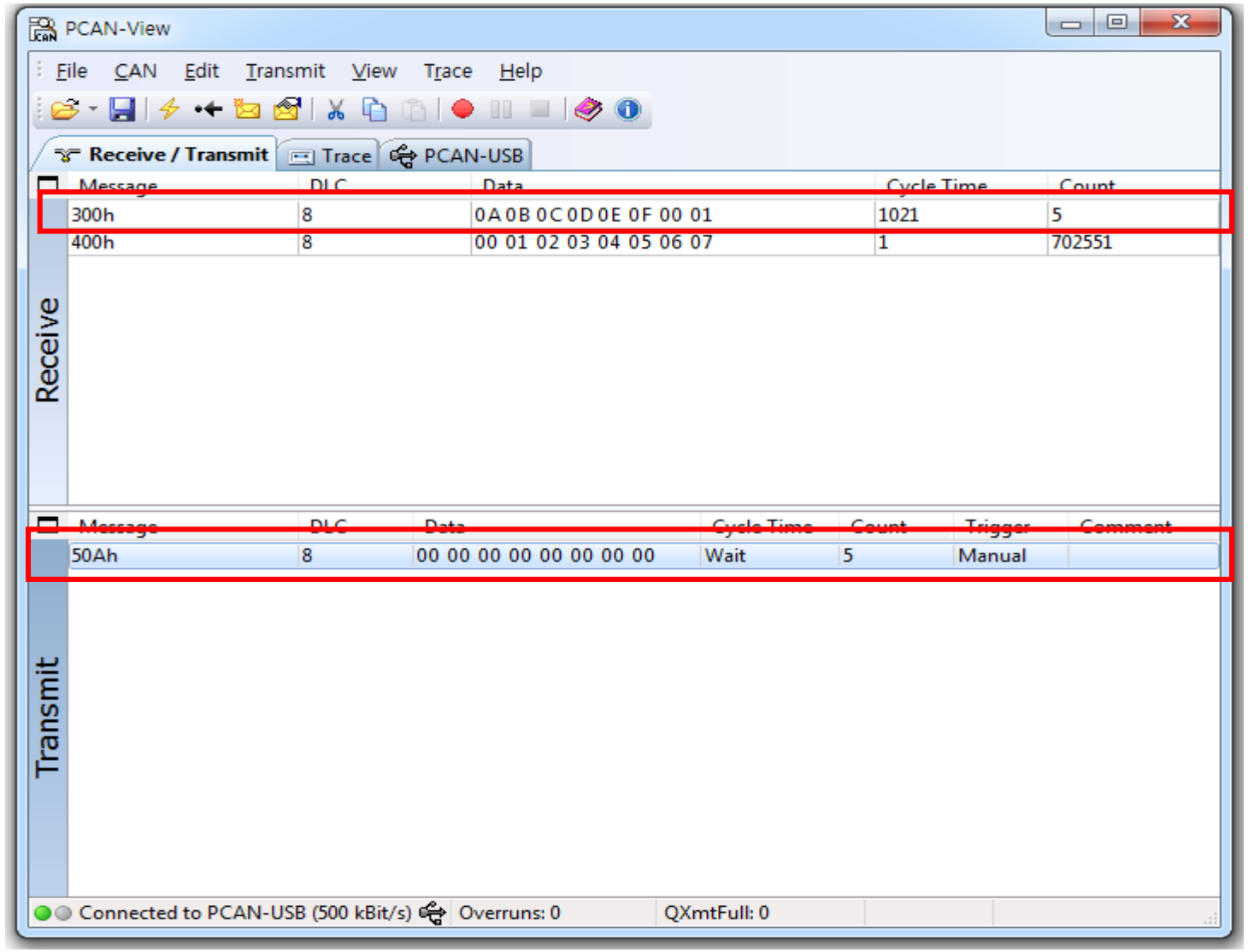
**수신된 메시지의 ID가 0x50A일 경우**
**ID : 0x300, DataLength : 8, temp에 내용을 전송**

# CAN Message 수신 확인(1/2)




Transmit 부분에서 우클릭 후
New Message클릭
ID를 50A로 설정 후 OK 클릭

# CAN Message 수신 확인(2/2)




50A Message 클릭 후
스페이스를 누르면 0x300의 아이디가 도착

# THANK YOU