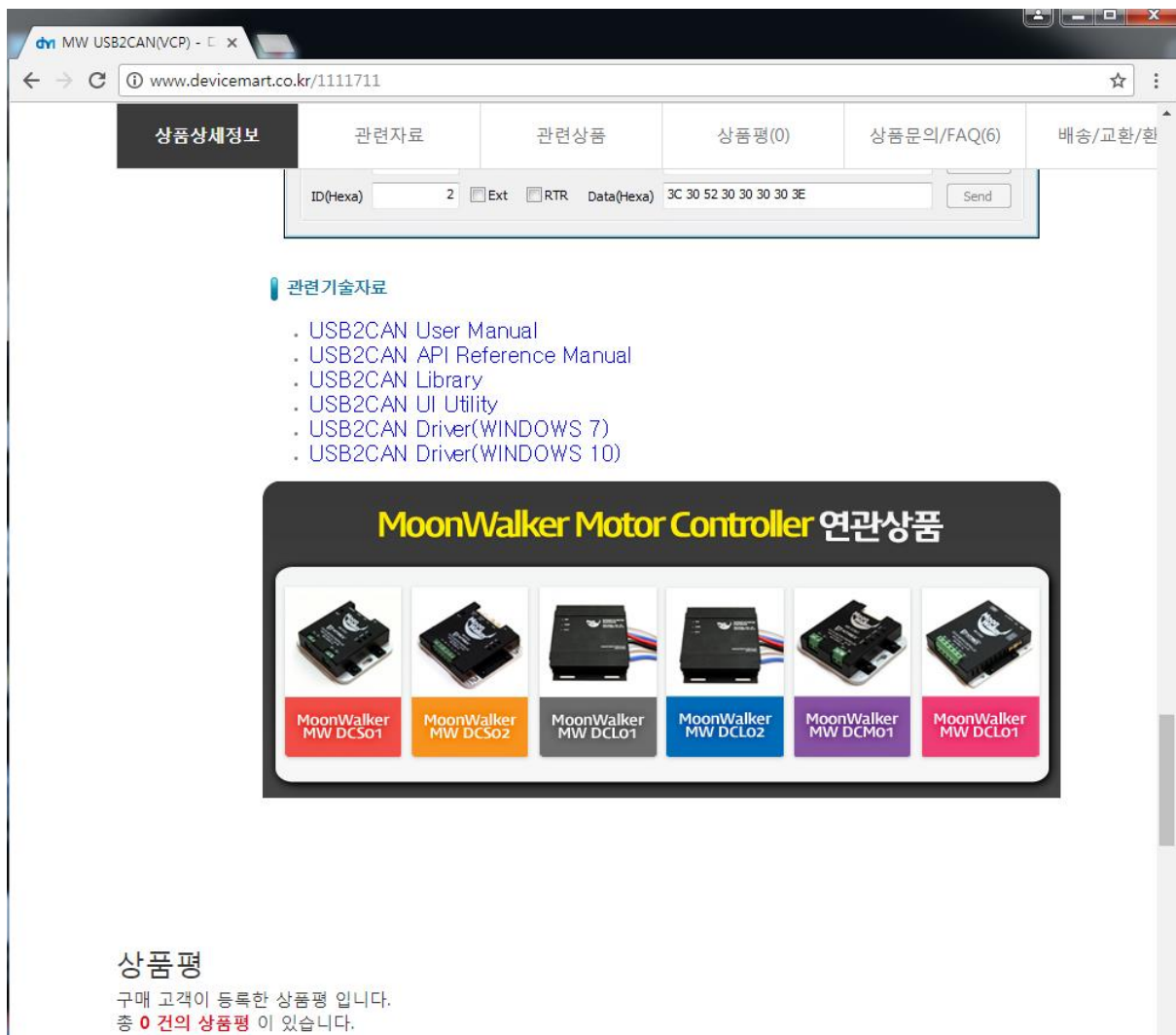


먼저 <http://www.devicemart.co.kr/1111711> 에 들어가서 맨아래쪽에 관련기술자료로 메뉴얼과 드라이버에 관한 자료들이 있다. 드라이버 설치하는 초간단하기때문에 설명을 보고 하면 쉽게 설치 할수있다.



The screenshot shows the website for MoonWalker Motor Controller. The browser address bar displays [www.devicemart.co.kr/1111711](http://www.devicemart.co.kr/1111711). The page has a navigation bar with tabs: '상품상세정보' (selected), '관련자료', '관련상품', '상품평(0)', '상품문의/FAQ(6)', and '배송/교환/환'. Below the navigation bar is a form for sending data via USB2CAN, with fields for ID (Hexa) set to 2, Ext, RTR, and Data (Hexa) set to 3C 30 52 30 30 30 3E. A 'Send' button is next to the data field.

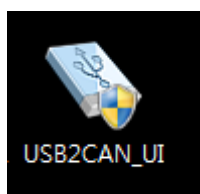
Below the form, under the '관련기술자료' (Related Technical Documents) section, there is a list of links:

- USB2CAN User Manual
- USB2CAN API Reference Manual
- USB2CAN Library
- USB2CAN UI Utility
- USB2CAN Driver(WINDOWS 7)
- USB2CAN Driver(WINDOWS 10)

Below the list, there is a section titled 'MoonWalker Motor Controller 연관상품' (MoonWalker Motor Controller Related Products). It displays six product images with their respective labels:

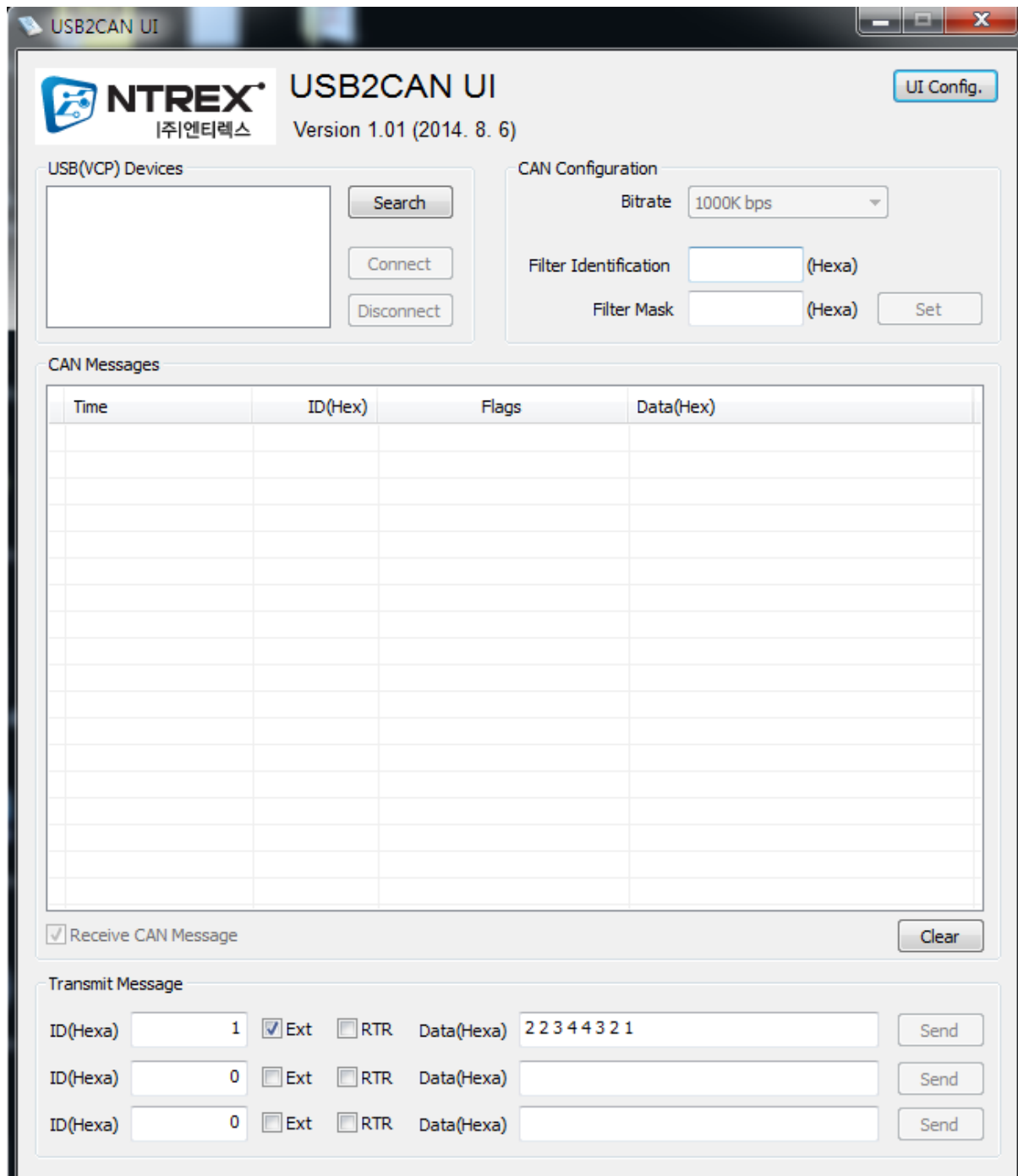
- MoonWalker MW DCS01
- MoonWalker MW DCS02
- MoonWalker MW DCL01
- MoonWalker MW DCL02
- MoonWalker MW DCM01
- MoonWalker MW DCL01

At the bottom left, there is a '상품평' (Product Review) section. It states: '구매 고객이 등록한 상품평 입니다.' (This is a product review registered by a purchasing customer.) and '총 0 건의 상품평 이 있습니다.' (There are 0 product reviews in total.).



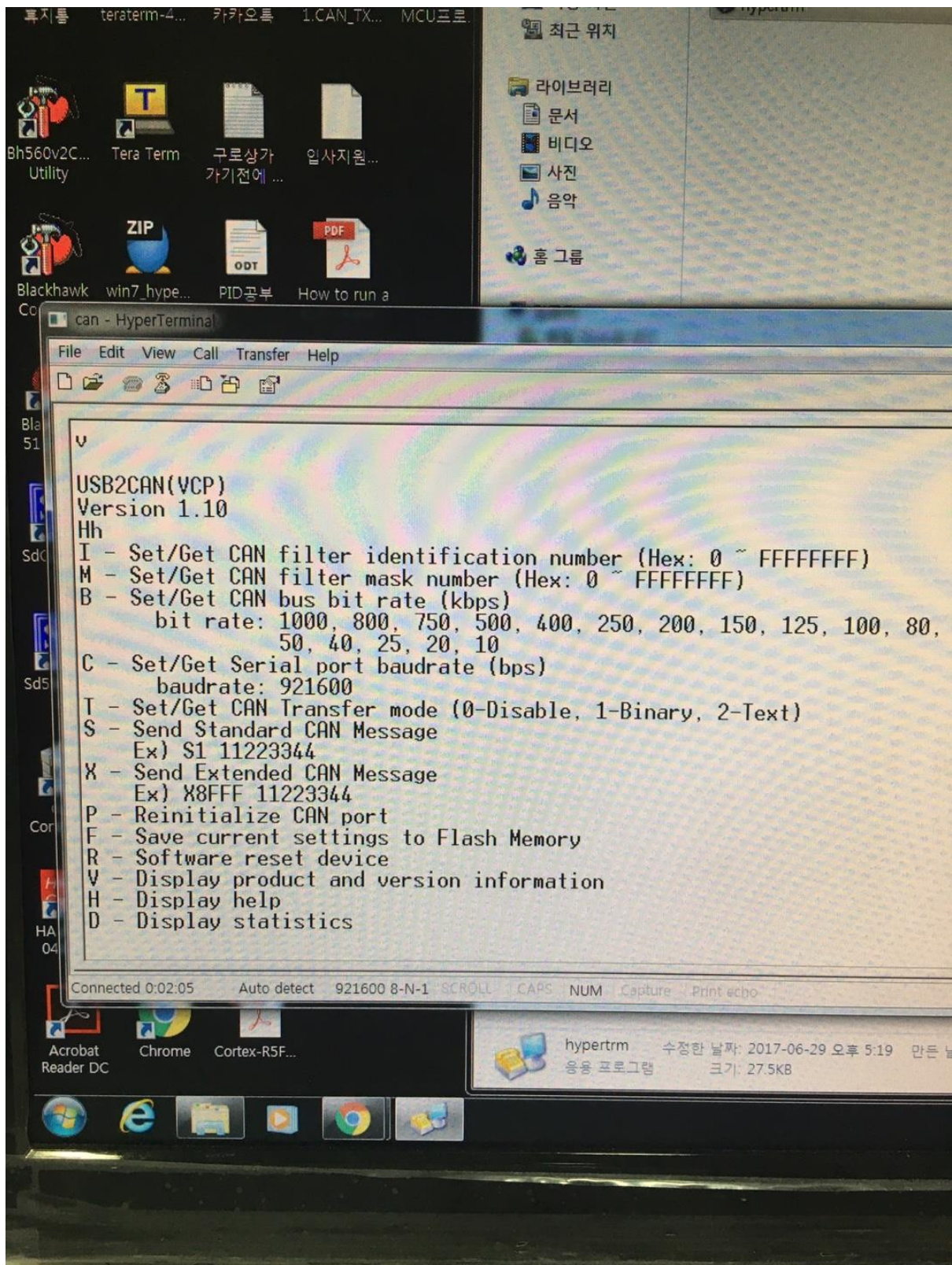
설치 하고 나면 아이콘이 생김. 이것이 우리가 사용할 can ui 유틸리티이다.

켰을때 이렇게 되면 잘 설치가 된것임.



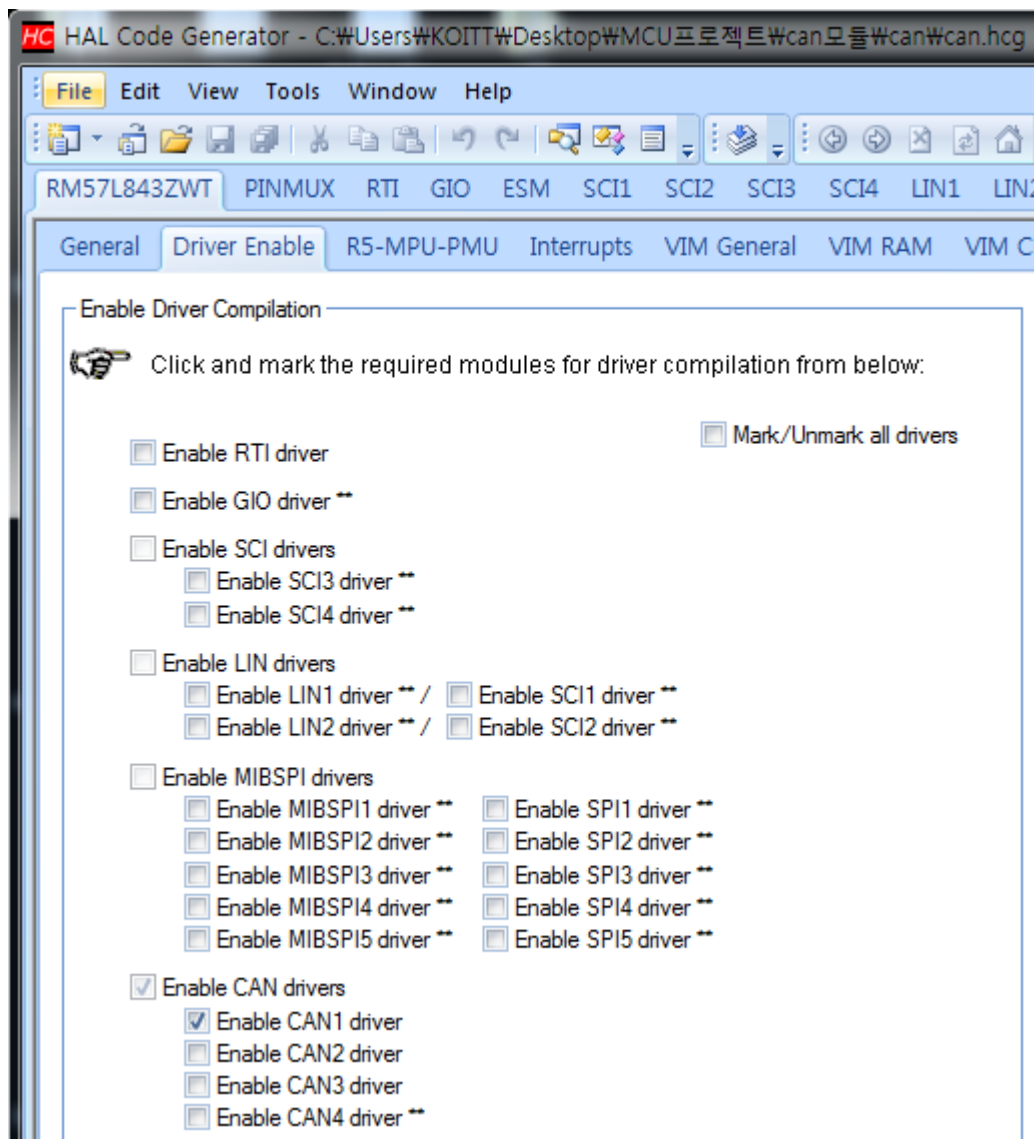
메뉴얼을 읽어보면 하이퍼터미널을 사용하여 연결도 가능하다.

아래의 사진은 하이퍼터미널로 연결했을때 사진임.



halcogen 설정하기.

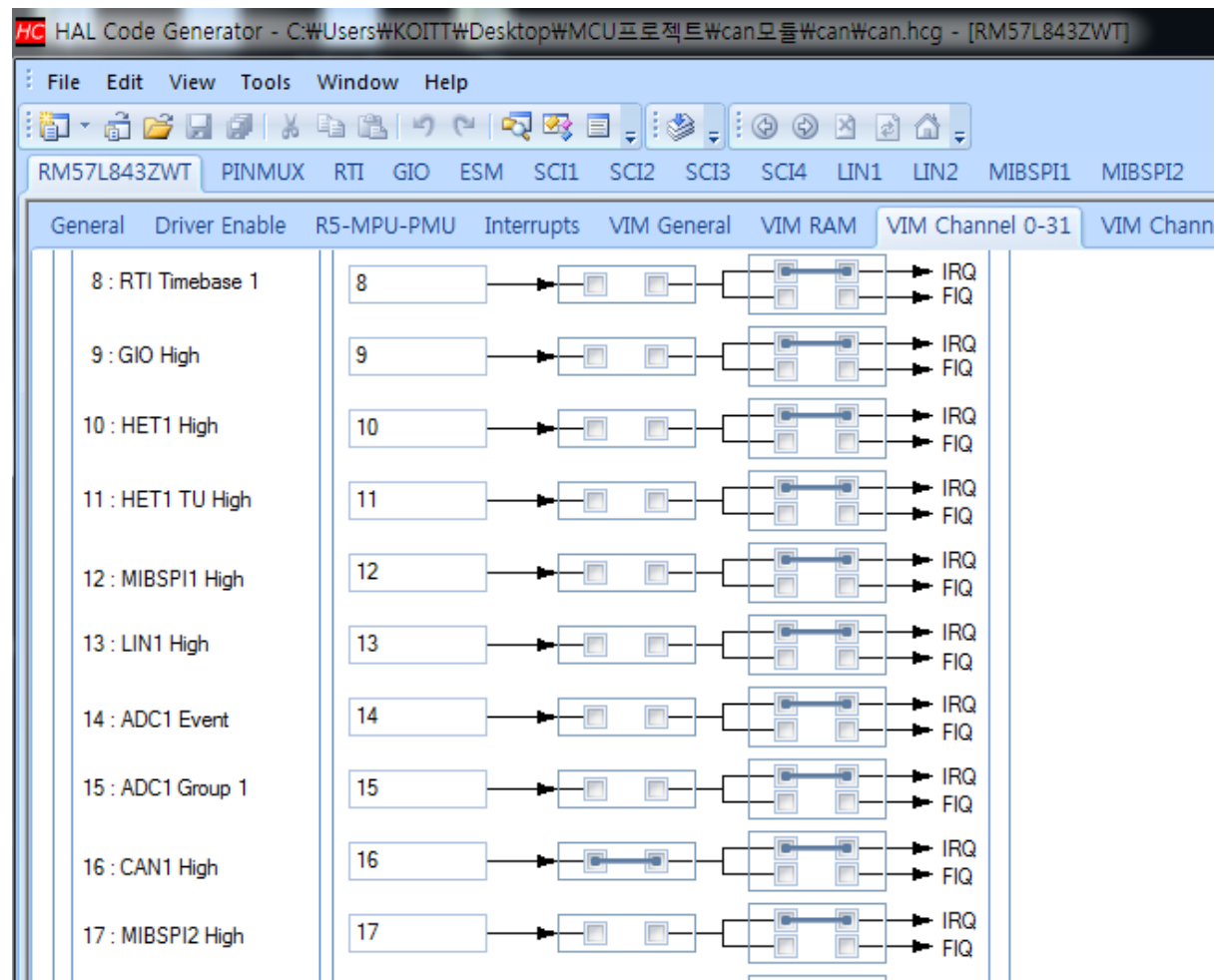
Driver Enable -> Enable CAN1 driver 만 클릭 나머지는 체크해제.



can통신을 할때 receive할때 interrupt를 사용해야함.

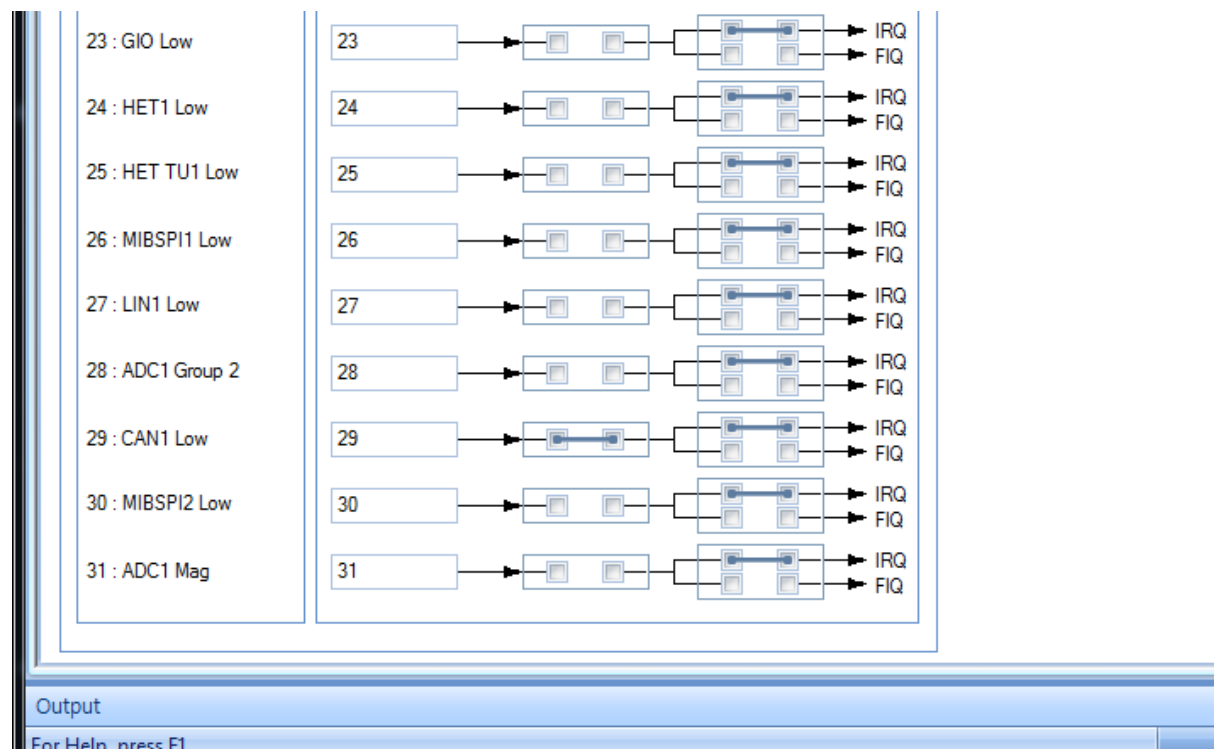
따라서 VIM channel에서 can1과 관련된것들 체크를 해야함.

VIM Channel 0-31-> CAN1 High 체크

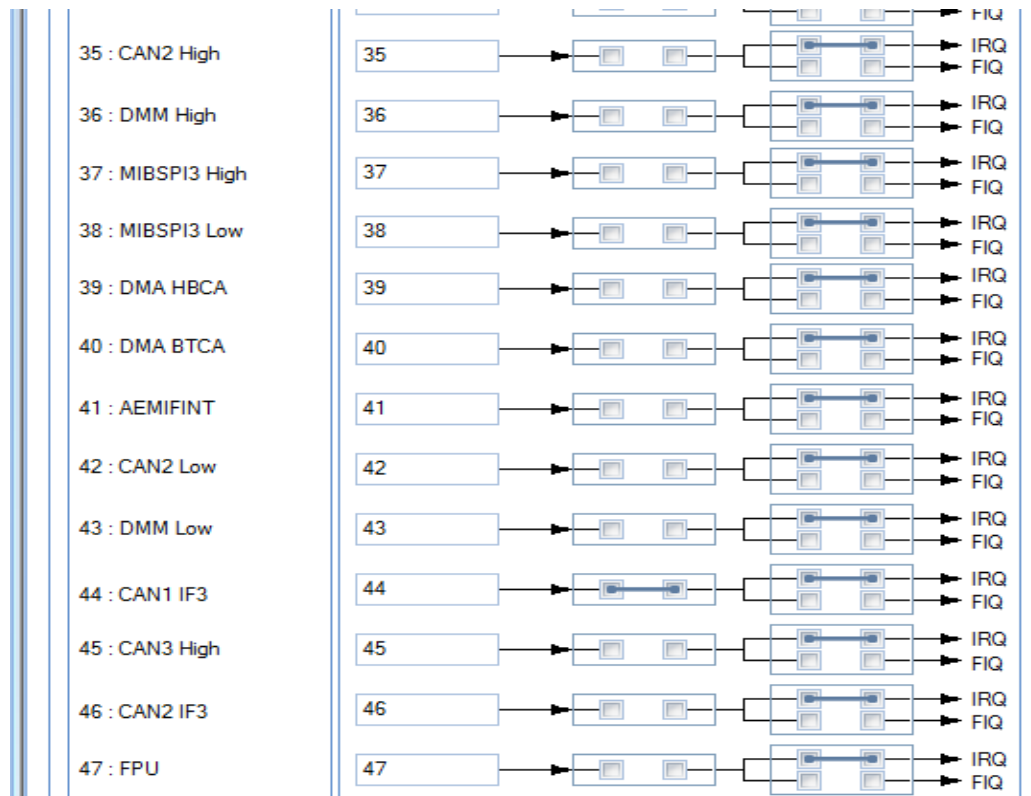




VIM Channel 0-31-> CAN1 Low 체크



VIM Channel 32-63-> CAN1 IF3 체크



RTFM!

메뉴얼을 꼭!!!!!! 꼼꼼하게 읽어야함.(대충읽었다가 바보짓을많이하게됨ㅏㅏ)

메뉴얼을 읽어보면 우리가 사용하는 can module은 vcp형태로 데이터 전송률이 최대 300K byte/second라고 나와있음.

CAN1->bit Rate를 250으로 설정. (최대 300이지만 250으로 하는이유는 UI에서bit rate 구성을 250가 최대라서..)



Calculated CAN1 Timing

VCLKA1: 75.000 → 24 (BRP) → 3.000 (fBp) → 0 (BRPE) → 3.000 (fCan) → 333.333 (tQ)

Nominal Bit Time: 12

Nominal Bit Rate: 250.000

Sample Point: 66.667

Synchronization Jump Width: 4

CAN1 General Configuration

☐ Disable Automatic Retransmission    ☒ Enable Identifier Extension    ☐ Enable Ram ECC

먼저 TX 설정하는 부분!!!

CAN1-> CAN1 Msg1-8 -> Message 1 Configuration

Activate체크, TX체크, id는 1로설정, id 아래 000007F0로 한다.

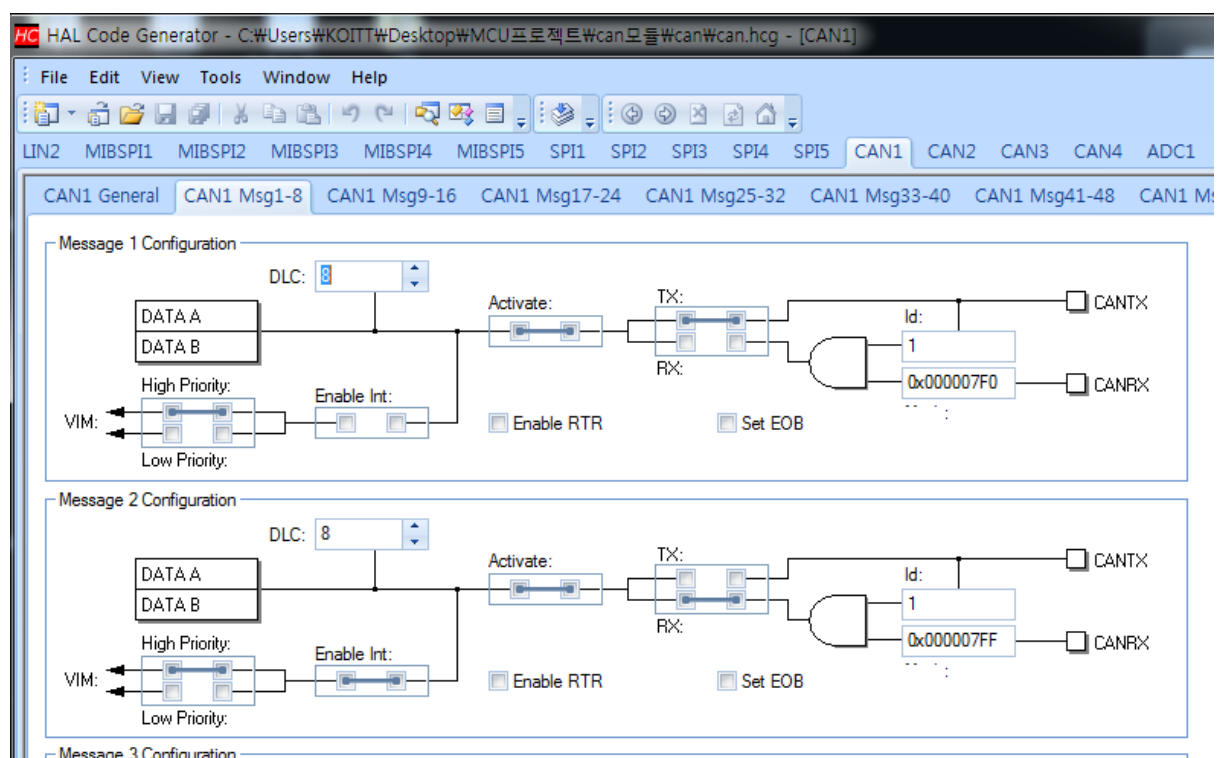
(id 아래부분은 mask 설정하는부분임.

TX와 RX의 id는통일해야하고 mask 설정값은 달라야 함.)

RX 설정하는 부분!!!

CAN1-> CAN1 Msg1-8 -> Message 2 Configuration

Enable Int 체크, Active체크, RX 체크, id=1로 설정, mask값은 000007FF로 한다.



generated code 를 함.

[CCS main code]

```
#include "HL_sys_common.h"  
#include "HL_system.h"  
#include "HL_can.h"  
#include "HL_esm.h"  
#include "HL_sys_core.h"
```

```
#include "stdio.h"  
#define D_COUNT 8  
#define D_SIZE 8
```

```
uint32 cnt = 0;  
uint32 error = 0;  
uint32 tx_done = 0;
```

```
uint8_t tx_data[D_COUNT] = {1,2,3,4,4,3,2,1};  
uint8_t rx_data[D_COUNT] = {0};
```

```
uint32_t checkPackets(uint8_t *src_packet, uint8_t *dst_packet, uint32_t psize);
```

```
void main(void)
```

```

{
    /* enable irq interrupt in */
    _enable_IRQ_interrupt_();

    /** - configuring CAN1 MB1,Msg ID-1 to transmit and CAN2 MB1 to receive */
    canInit();

    printf("start\n");
    /** - enabling error interrupts */
    canEnableErrorNotification(canREG1);

    while(1)
        canTransmit(canREG1, canMESSAGE_BOX1, (const uint8 *) &tx_data[0]);

void canMessageNotification(canBASE_t *node, uint32_t messageBox)
{
    if(node==canREG1)
    {
        while(!canIsRxMessageArrived(canREG1, canMESSAGE_BOX2));
        canGetData(canREG1, canMESSAGE_BOX2, (uint8 *)&rx_data[0]); /* copy to RAM */
//        rx_data[0] += 8;
        printf("rx_data : %x\n",*rx_data);
    }
}
}

```

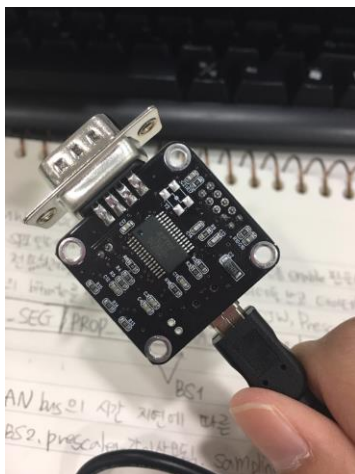
```
HL_sys_main.c HL_sys_main.c *HL_sys_main.c
14
15 uint32 cnt = 0;
16 uint32 error = 0;
17 uint32 tx_done = 0;
18
19 uint8_t tx_data[D_COUNT] = {1,2,3,4,4,3,2,1};
20 uint8_t rx_data[D_COUNT] = {0};
21
22 uint32_t checkPackets(uint8_t *src_packet, uint8_t *dst_packet, uint32_t psize);
23
24 void main(void)
25 {
26     /* enable irq interrupt in */
27     _enable_IRQ_interrupt_();
28
29     /** - configuring CAN1 MB1,Msg ID-1 to transmit and CAN2 MB1 to receive */
30     canInit();
31
32     printf("start\n");
33     /** - enabling error interrupts */
34     canEnableErrorNotification(canREG1);
35
36     //canGetData(canREG1, canMESSAGE_BOX2, rx_data);
37
38
39     // error = checkPackets(&rx_data[0],&rx_data[0],D_SIZE);
40
41     while(1)
42     {
43         canTransmit(canREG1, canMESSAGE_BOX1, (const uint8 *) &tx_data[0]);
44         // printf("rx_data : %s\n",rx_data);
45     }
46 }
47
48 /* can interrupt notification */
49 void canMessageNotification(canBASE_t *node, uint32_t messageBox)
50 {
51     if(node==canREG1)
52     {
53         while(!canIsRxMessageArrived(canREG1, canMESSAGE_BOX2));
```

코드를 입력하고 망치를 때립니다.

컴퓨터 2대를 사용해서 하나는 mcu, 하나는 can모듈을 연결해서 사용하였다.

(컴퓨터 한대로 하면 송수신 충돌일어날까봐 혹시 몰라서 2대를 사용함.)

can 모듈에서 LED 불빛을 확인해보면 초록-RUN, 파란-RX, 빨강-TX



10번먼저 읽어보세요!!

1. MCU보드에 코드를 망치를 때리고
2. 벌레를 잡고
3. 포트연결을 한다.



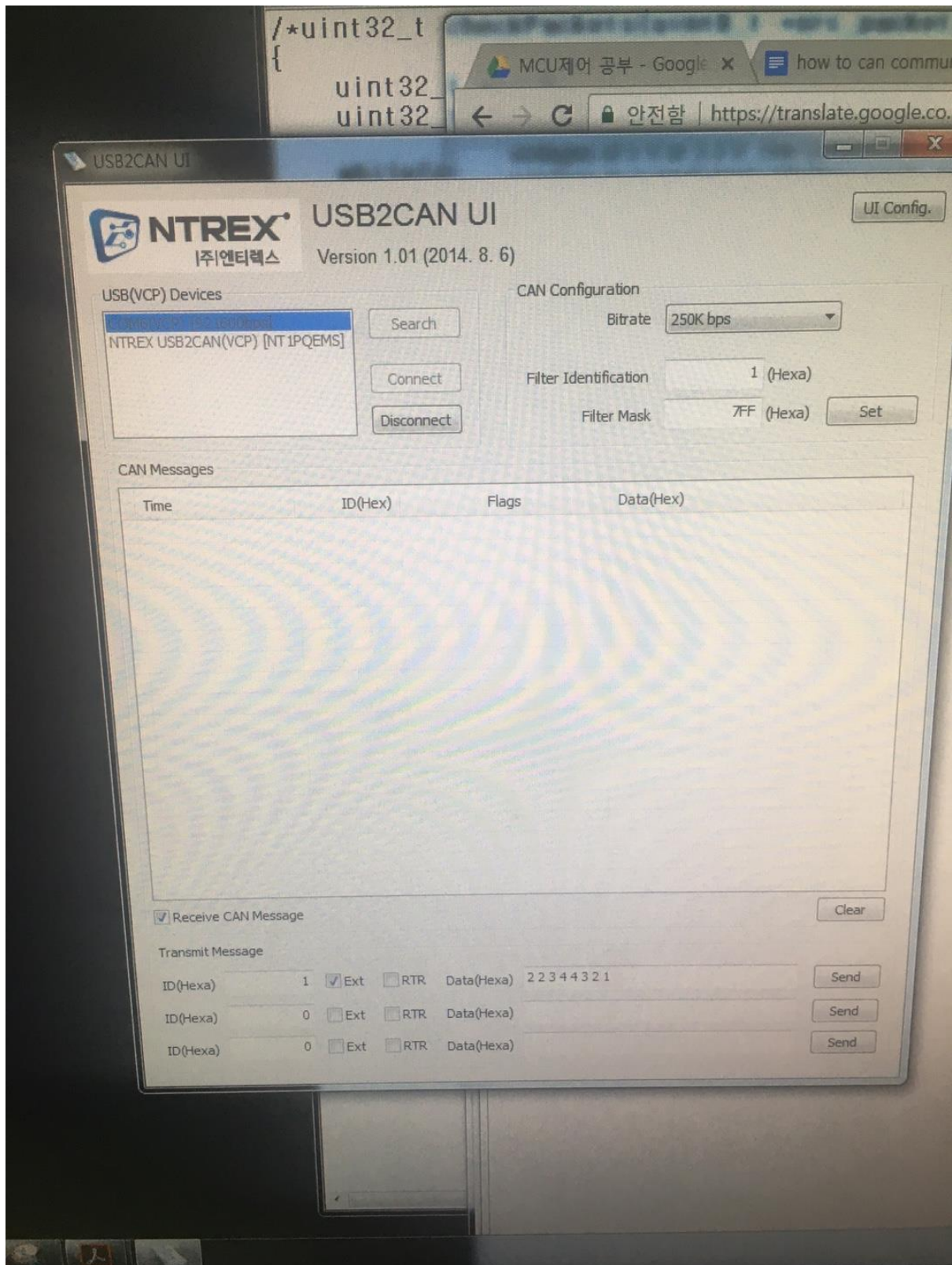
그림 1-5 D-Sub 9pin male 커넥터와 핀 위치

Pin	Function
1	-
2	CAN Low (CAN_L)
3	Ground (GND)
4	-
5	-
6	-
7	CAN High (CAN_H)
8	-
9	-



4. 재생버튼 누르기전에 UI 유틸리티를 켜고 filter identification은 1로, filter mask는 7FF로 하고 set을 누르고 connect 해놓는다.



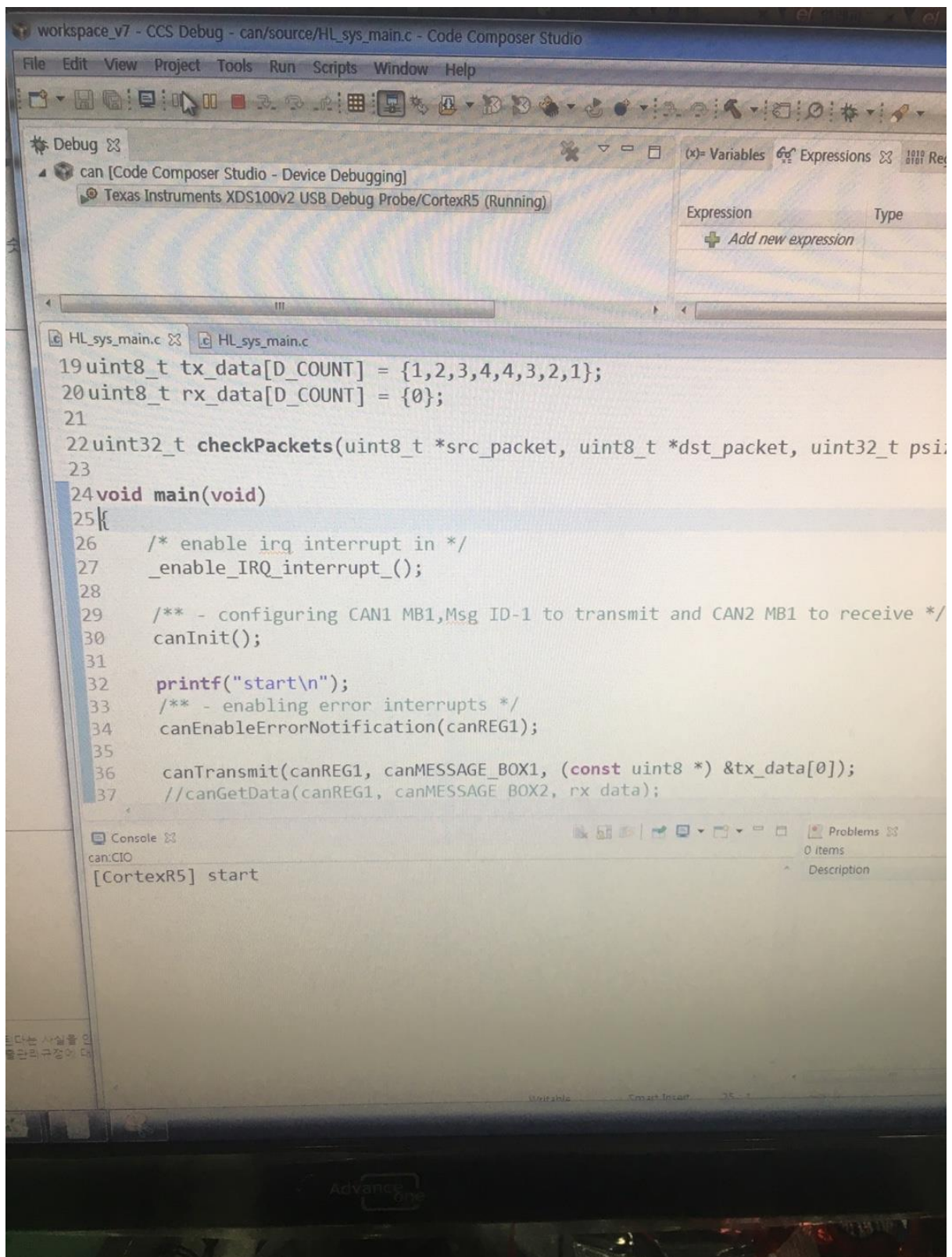


5. 캔을 연결하면 모듈에 초록불이 들어온다. 초록불이 들어오면 현재 run상태..

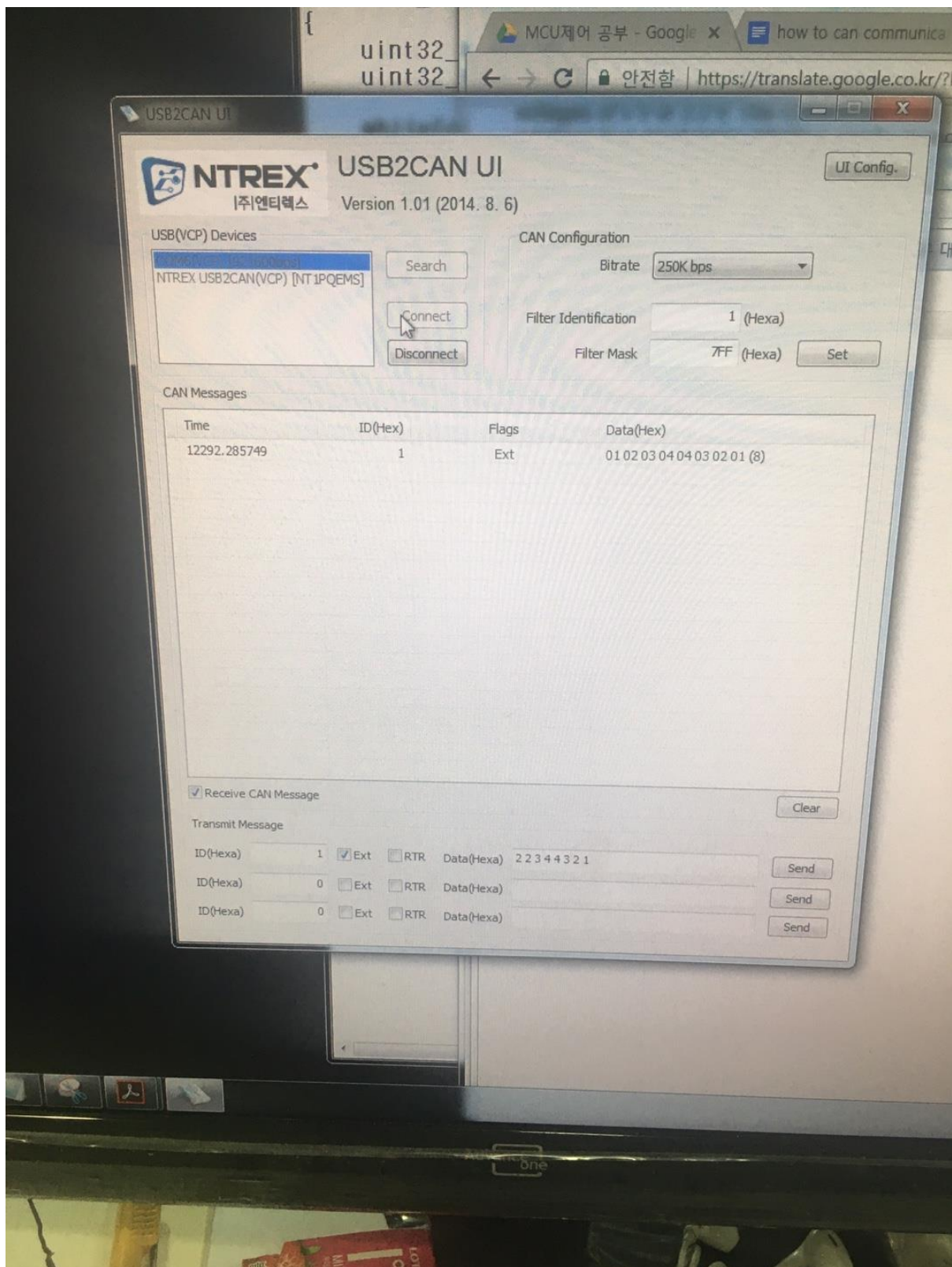


6.MCU에서 can으로 통신을 보냄



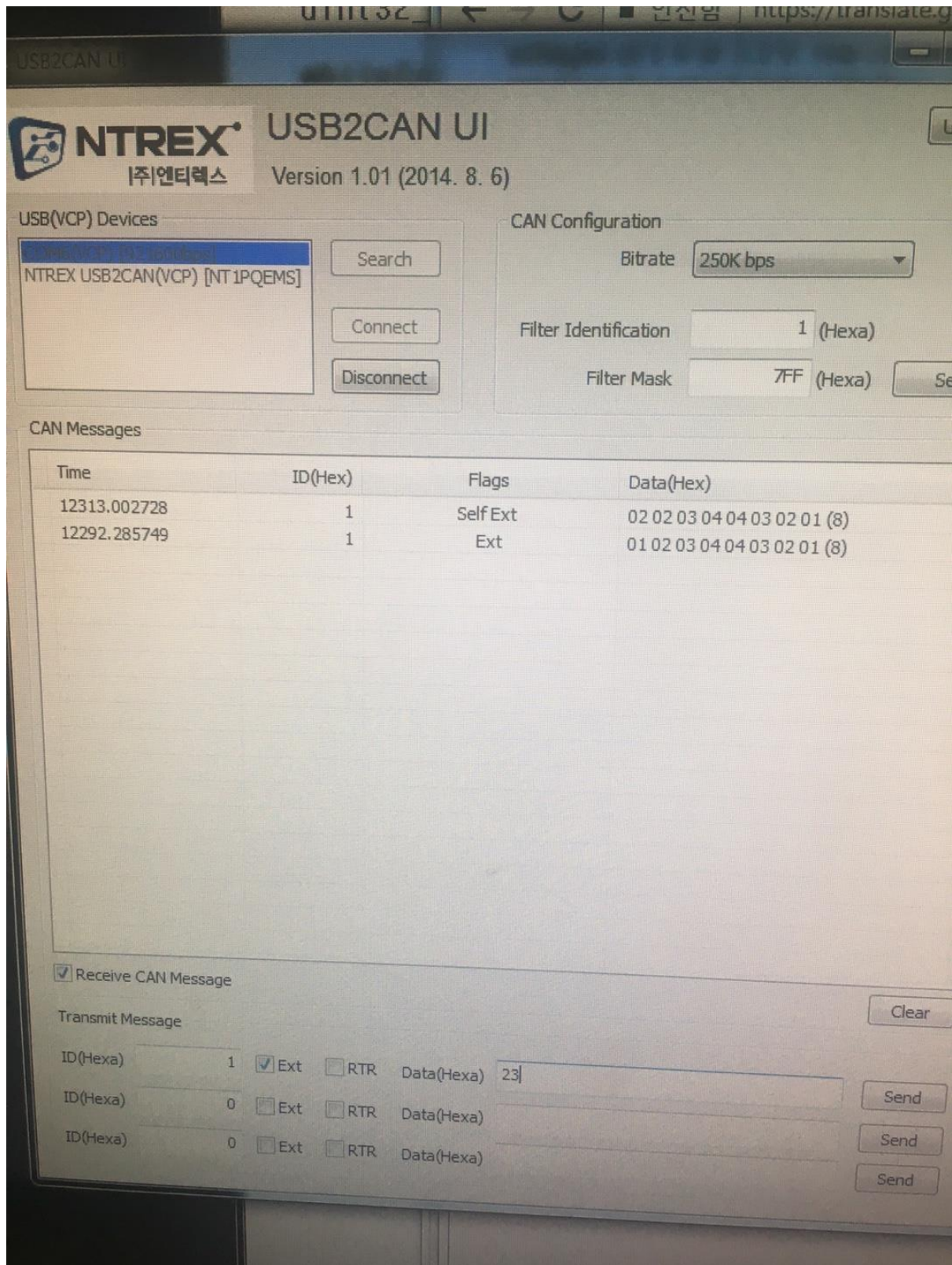


7. MCU에서 보낸 신호가 들어옴.

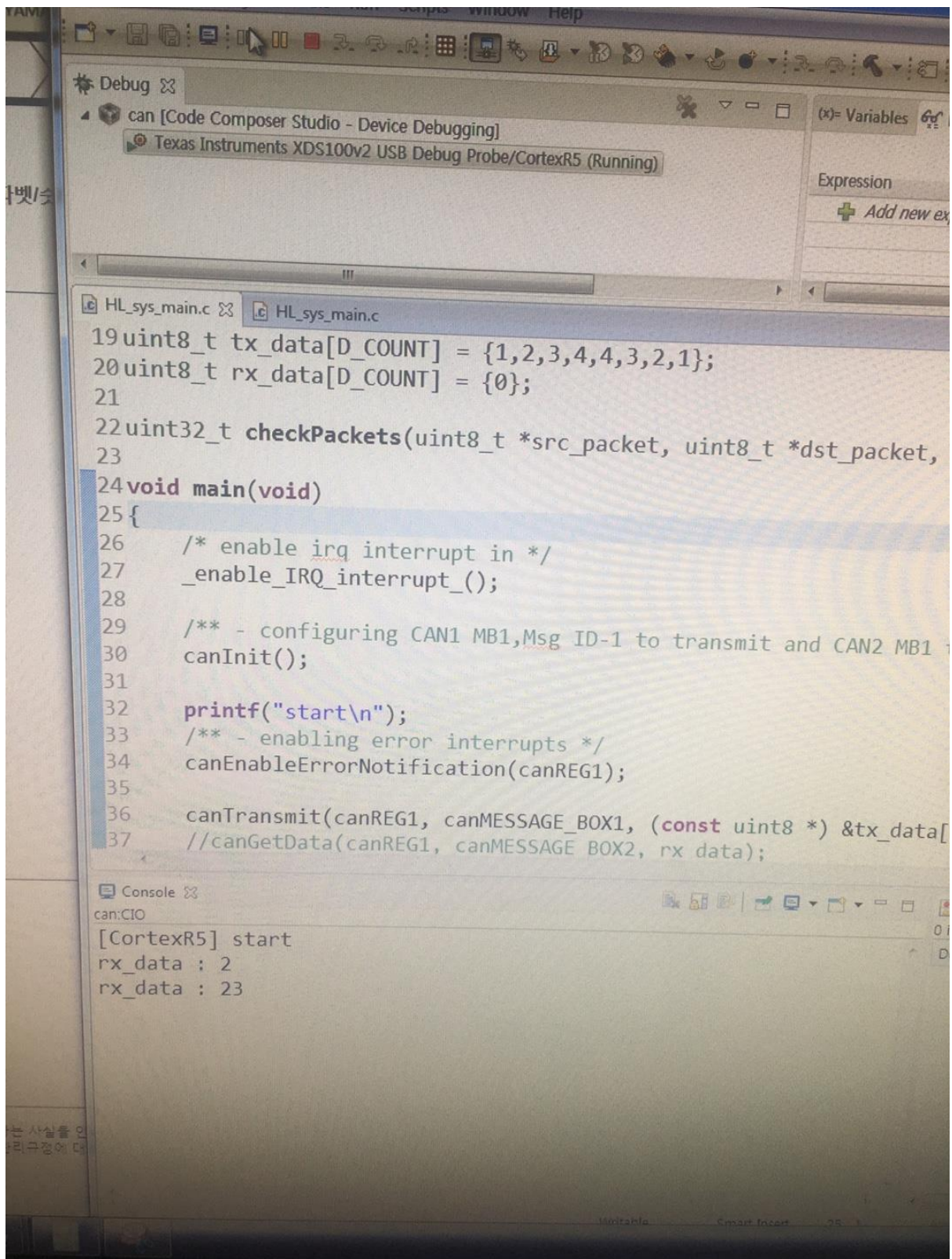




8. 반대로 can에서 mcu로 메시지를 보낸다. 2글자만 보내짐 (16bit)



9. can에서 보낸 메시지를 MCU에서 받음.







10. can모듈 기준으로 메시지를 받으면 파란불(RX)가 켜지고, can에서 메시지를 보내면 빨간불(TX)로 켜짐.

#### 11. !!!주의!!!

현재 올린 코드는 while문안에 transmit 를 넣었기 때문에 mcu에서 계속 메시지를 보낼것이다.

학원 컴퓨터가 사양이 안좋아서 멈춤ㅠㅠ 그러니 transmit를 while문 위로 올려서 하나씩 통신 하면 될것이다. (멈춤 조심하세요!!!)

