(Mango-M32F4)

Test Guide

http://www.mangoboard.com/
http://cafe.naver.com/embeddedcrazyboys
Crazy Embedded Laboratory

Document History

Revision	Date	Change note
1.0	2016.03.21	Initital 오영환

목 차

1. Booting

1.1. Hardware Connection

1.2. Debug Serial Port

디버그 시리얼 포트의 설정은 아래와 같습니다.

Bud Rate	**115200**
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None

2. Key Test

아래와 같이 출력 됩니다.

```
[CRZ] Work_Event_Process (60): USER Key 1 pressed
[CRZ] Work_Event_Process (64): USER Key 1 released
[CRZ] Work_Event_Process (68): USER Key 2 pressed
[CRZ] Work_Event_Process (72): USER Key 2 released
```

3. LED Test

```
1> LED Test
```

4. SDMMC Test

T-Flash를 장착합니다.

```
2> SDMMC Test
```

2번을 선택합니다.

```
[CRZ] main (148): 2 is selected
[CRZ] SDMMC_Test (82):
[CRZ] SDMMC_Initialization (30):
[CRZ] SDMMC_Initialization (38): FATFS_LinkDriver OK
[CRZ] f_mount (2486):
[CRZ] f_mount (2492): vol: 0
[CRZ] SDMMC_Initialization (46): f_mount OK
[CRZ] SDMMC_Test (106): BSP_SD_Init OK
[CRZ] SDMMC_Format (70): SDMMC Card Format OK
[CRZ] SDMMC_Test (113): Format OK
[CRZ] f_open (2539):
[CRZ] SDMMC_Test (122): f_open STM32.TXT OK
[CRZ] SDMMC_Test (133): f_close STM32.TXT OK
[CRZ] SDMMC_Test (142): STM32.TXT file Write OK
[CRZ] f_open (2539):
[CRZ] SDMMC_Test (151): f_open STM32.TXT OK
[CRZ] SDMMC_Test (162): f_read STM32.TXT OK
[CRZ] Wait_N_Seconds (41): Wait (1/1) second
[CRZ] SDMMC_Test (178): Success demo OK
[CRZ] Wait_N_Seconds (41): Wait (1/1) second
[CRZ] SDMMC_Test (178): Success demo OK
[CRZ] Wait_N_Seconds (41): Wait (1/1) second
[CRZ] SDMMC_Test (178): Success demo OK
[CRZ] Wait_N_Seconds (41): Wait (1/1) second
[CRZ] SDMMC_Test (178): Success demo OK
[CRZ] Wait_N_Seconds (41): Wait (1/1) second
[CRZ] SDMMC_Test (178): Success demo OK
[CRZ] Wait_N_Seconds (41): Wait (1/1) second
[CRZ] SDMMC_Test (178): Success demo OK
```

Success demo OK가 계속 출력이 됩니다.

5. SRAM Test

```
3> SRAM Test
```

3번을 선택합니다.

```
[CRZ] main (148): 3 is selected
[CRZ] SRAM_Test (113):
[CRZ] Buffercmp (102): Count: 1, Data is same.
[CRZ] Buffercmp (102): Count: 2, Data is same.
[CRZ] Buffercmp (102): Count: 3, Data is same.
[CRZ] Buffercmp (102): Count: 4, Data is same.
[CRZ] Buffercmp (102): Count: 5, Data is same.
... ... ... ... ... ... ...
[CRZ] Buffercmp (102): Count: 250, Data is same.
[CRZ] Buffercmp (102): Count: 251, Data is same.
[CRZ] Buffercmp (102): Count: 252, Data is same.
[CRZ] Buffercmp (102): Count: 253, Data is same.
[CRZ] Buffercmp (102): Count: 254, Data is same.
[CRZ] Buffercmp (102): Count: 255, Data is same.
[CRZ] Buffercmp (102): Count: 256, Data is same.
[CRZ] SRAM_Test (140): Test OK.
```

6. USB Host Test

```
4> USB Host Test
```

4번을 선택합니다.

```
[CRZ] main (148): 4 is selected
[CRZ] USB_Host_Test (211):
[CRZ] USB_Host_Initialization (41):
[CRZ] USB_Host_Initialization (49): FATFS_LinkDriver OK
[CRZ] USBH_LL_Init (214):
[CRZ] HAL_HCD_MspInit (45):
[CRZ] HAL_HCD_MspInit (51):
[CRZ] USB_Host_Initialization (53): USBH_Init done
[CRZ] USB_Host_Initialization (57): USBH_RegisterClass done
[CRZ] USB_Host_Initialization (61): USBH_Start done
USB Device Attached
PID: 6366h
```

```
VID: 58fh
Address (#1) assigned.
Manufacturer : Generic
Product : Mass Storage Device
Serial Number : 058F63666433
Enumeration done.
This device has only 1 configuration.
Default configuration set.
Switching to Interface (#0)
Class      : 8h
SubClass : 6h
Protocol : 50h
MSC class started.
Number of supported LUN: 1
LUN #0:
Inquiry Vendor   : Multiple
Inquiry Product : Card   Reader
Inquiry Version : 1.00
MSC Device ready
MSC Device capacity : 3653238272 Bytes
Block number : 15523839
Block Size     : 512
[CRZ] USB_Host_Test (224): Application state is START
[CRZ] MSC_Application (76):
[CRZ] f_mount (2486):
[CRZ] f_mount (2492): vol: 0
[CRZ] MSC_Application (85): f_mount Ok
[CRZ] f_open (2539):
[CRZ] MSC_Application (128): f_open Ok
[CRZ] MSC_Application (139): f_write Ok
[CRZ] f_open (2539):
[CRZ] MSC_Application (151): f_open Ok
[CRZ] MSC_Application (162): f_read Ok
[CRZ] MSC_Application (174): Compare read data Ok
[CRZ] Wait_N_Seconds (41): Wait (1/1) second
[CRZ] USB_Host_Test (240): Success demo OK
```

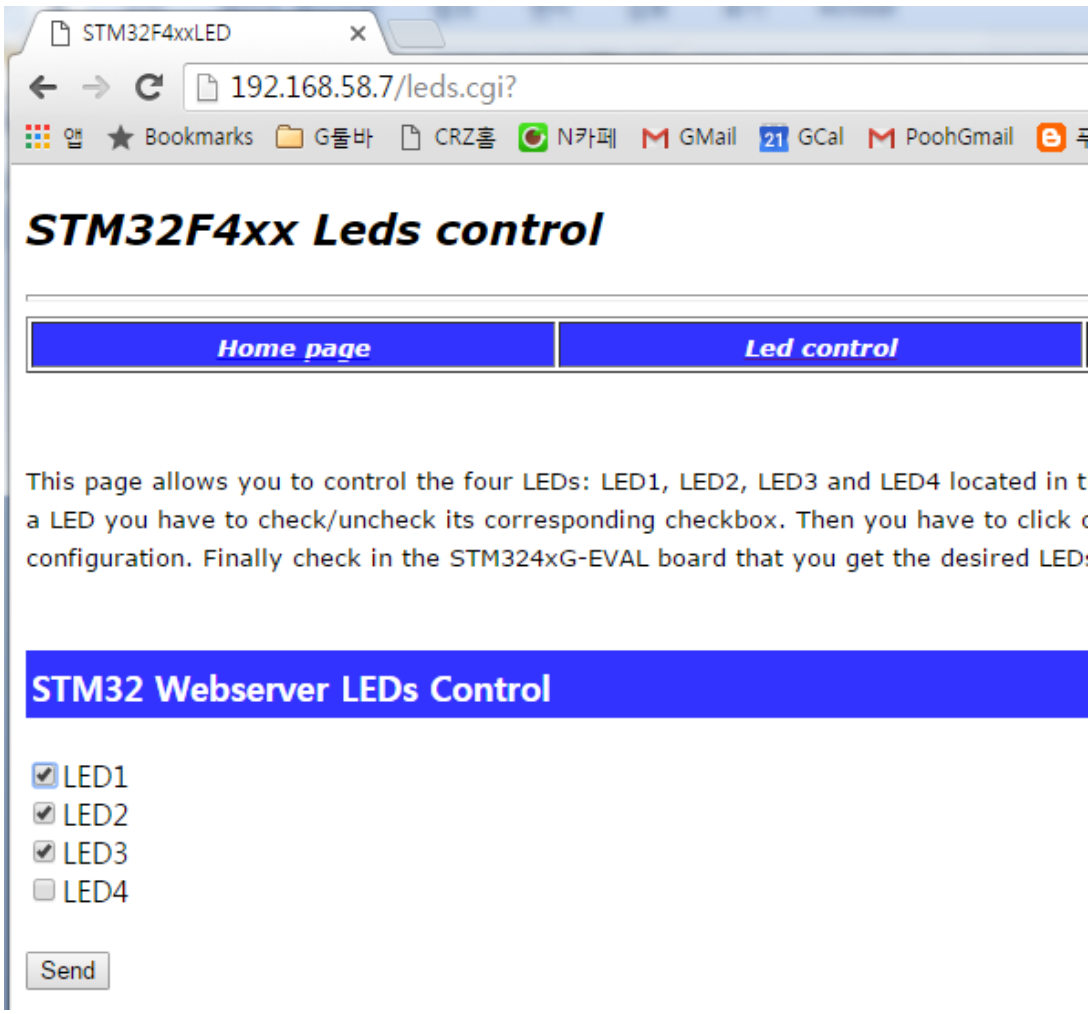
Success demo OK가 계속 출력이 됩니다.

7. Ethernet HTTP Test

5> Ethernet HTTP Test

5번을 선택합니다.

```
[CRZ] main (148): 5 is selected
[CRZ] Ethernet_HTTP_Test (72):
[CRZ] Ethernet_HTTP_Test (76): lwip_init done
[CRZ] HAL_ETH_MspInit (85):
[CRZ] Ethernet_HTTP_Test (80): Netif_Config done
[CRZ] Ethernet_HTTP_Test (84): httpd_init done
[CRZ] Ethernet_HTTP_Test (88): User_notification done
[CRZ] Ethernet_HTTP_Test (91): ethernetif_set_link done
[CRZ] DHCP_Process (147):     State: Looking for DHCP sever ...
[CRZ] DHCP_Process (172): IP address assigned by a DHCP server: 192.168.58.7
```



Led control 부분에서 LED1~3 부분을 선택한 후 Send를 누르면 보드의 LED 3개가 켜집니다.

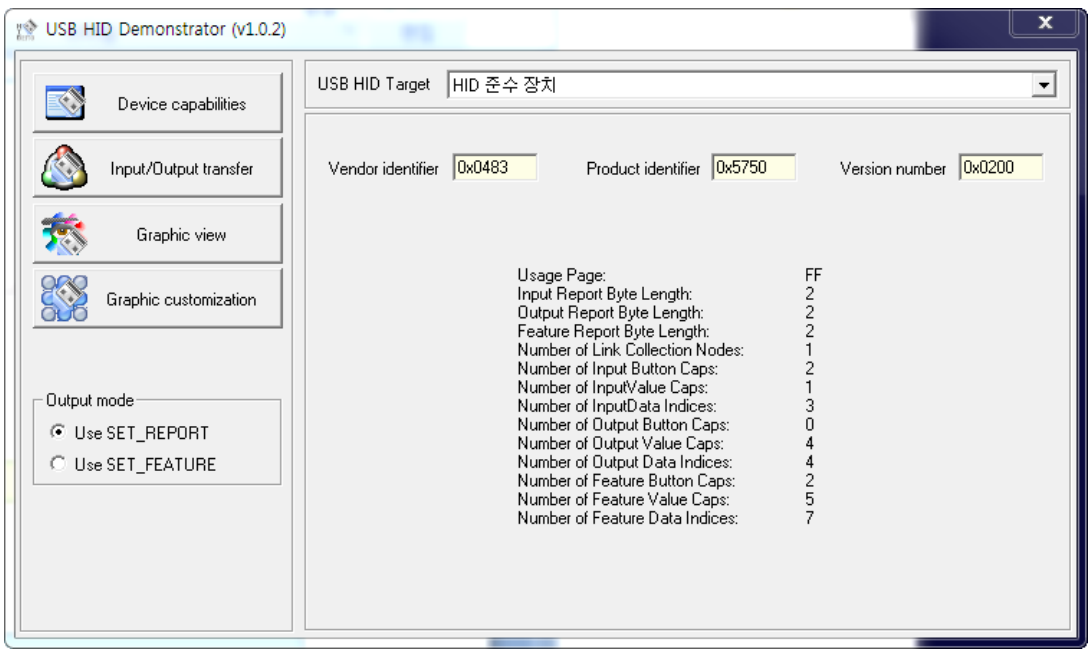
8. USB Device Test

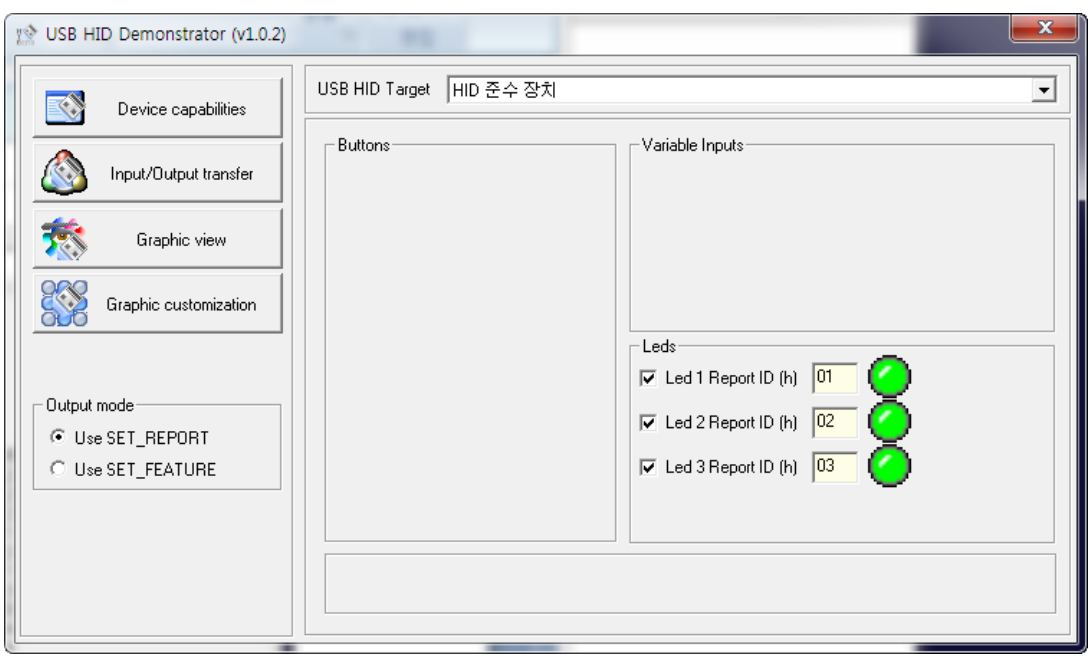
```
6> USB Device Test
```

6번을 선택합니다.

```
[CRZ] main (148): 6 is selected
[CRZ] USB_Device_Test (29):
[CRZ] USB_Device_Test (33): USBD_Init done
```

[CRZ] USB_Device_Test (37): USBD_RegisterClass done
[CRZ] USB_Device_Test (41): USBD_CUSTOM_HID_RegisterInterface done
[CRZ] USB_Device_Test (45): USBD_Start done





Graphic view 부분에서 Led 1~3을 제어하면 실제 보드의 LED도 함께 동작하게 됩니다.

9. RTC Calendar Test

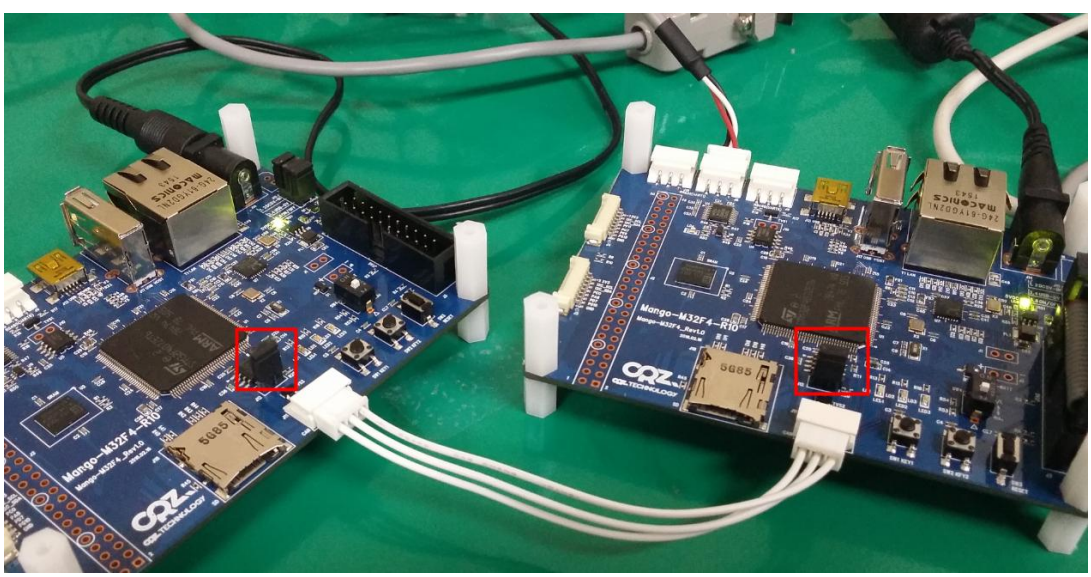
```
7> RTC Calendar Test
```

7번을 선택합니다.

```
[CRZ] main (148): 7 is selected
[CRZ] RTC_Test (97):
[CRZ] RTC_Test (132): Date: 02-18-2014, Time: 02:00:00
[CRZ] RTC_Test (132): Date: 02-18-2014, Time: 02:00:01
[CRZ] RTC_Test (132): Date: 02-18-2014, Time: 02:00:02
[CRZ] RTC_Test (132): Date: 02-18-2014, Time: 02:00:03
[CRZ] RTC_Test (132): Date: 02-18-2014, Time: 02:00:04
```

시간이 1초에 한번씩 출력되면 정상입니다.

10. CAN Test



두 보드를 위 사진과 같이 연결합니다.
붉은색 박스와 같이 점퍼가 연결되어야 합니다.

8> CAN Test

8번을 선택합니다.

[CRZ] main (148): 8 is selected	[CRZ] main (148): 8 is selected
[CRZ] CAN_Test (109):	[CRZ] CAN_Test (109):
[CRZ] Wait_N_Seconds (41): Wait (1/5) second	[CRZ] Wait_N_Seconds (41): Wait (1/5) second
[CRZ] Wait_N_Seconds (41): Wait (2/5) second	[CRZ] Wait_N_Seconds (41): Wait (2/5) second
[CRZ] Wait_N_Seconds (41): Wait (3/5) second	[CRZ] HAL_CAN_RxCpltCallback (95):
[CRZ] Wait_N_Seconds (41): Wait (4/5) second	CanHandle->pRxMsg->Data[0]: 1
[CRZ] Wait_N_Seconds (41): Wait (5/5) second	[CRZ] Wait_N_Seconds (41): Wait (3/5) second
[CRZ] CAN_Test (132): ++ubKeyNumber: 1	[CRZ] Wait_N_Seconds (41): Wait (4/5) second
[CRZ] Wait_N_Seconds (41): Wait (1/5) second	[CRZ] Wait_N_Seconds (41): Wait (5/5) second
[CRZ] Wait_N_Seconds (41): Wait (2/5) second	[CRZ] CAN_Test (132): ++ubKeyNumber: 2
[CRZ] HAL_CAN_RxCpltCallback (95):	[CRZ] Wait_N_Seconds (41): Wait (1/5) second
CanHandle->pRxMsg->Data[0]: 2	[CRZ] Wait_N_Seconds (41): Wait (2/5) second
[CRZ] Wait_N_Seconds (41): Wait (3/5) second	**[CRZ] HAL_CAN_RxCpltCallback (95):**
[CRZ] Wait_N_Seconds (41): Wait (4/5) second	**CanHandle->pRxMsg->Data[0]: 3**
[CRZ] Wait_N_Seconds (41): Wait (5/5) second	[CRZ] Wait_N_Seconds (41): Wait (3/5) second
[CRZ] CAN_Test (132): ++ubKeyNumber: 3	[CRZ] Wait_N_Seconds (41): Wait (4/5) second
[CRZ] Wait_N_Seconds (41): Wait (1/5) second	[CRZ] Wait_N_Seconds (41): Wait (5/5) second
[CRZ] Wait_N_Seconds (41): Wait (2/5) second	[CRZ] CAN_Test (132): ++ubKeyNumber: 4
[CRZ] HAL_CAN_RxCpltCallback (95):	[CRZ] Wait_N_Seconds (41): Wait (1/5) second
CanHandle->pRxMsg->Data[0]: 4	[CRZ] Wait_N_Seconds (41): Wait (2/5) second
[CRZ] Wait_N_Seconds (41): Wait (3/5) second	[CRZ] Wait_N_Seconds (41): Wait (3/5) second
[CRZ] Wait_N_Seconds (41): Wait (4/5) second	[CRZ] Wait_N_Seconds (41): Wait (4/5) second
[CRZ] Wait_N_Seconds (41): Wait (5/5) second	[CRZ] Wait_N_Seconds (41): Wait (5/5) second
[CRZ] Wait_N_Seconds (41): Wait (1/5) second	[CRZ] Wait_N_Seconds (41): Wait (1/5) second
[CRZ] Wait_N_Seconds (41): Wait (2/5) second	[CRZ] Wait_N_Seconds (41): Wait (2/5) second
[CRZ] Wait_N_Seconds (41): Wait (3/5) second	**[CRZ] HAL_CAN_RxCpltCallback (95):**
[CRZ] Wait_N_Seconds (41): Wait (4/5) second	**CanHandle->pRxMsg->Data[0]: 1**
[CRZ] Wait_N_Seconds (41): Wait (5/5) second	[CRZ] Wait_N_Seconds (41): Wait (3/5) second
[CRZ] CAN_Test (132): ++ubKeyNumber: 1	[CRZ] Wait_N_Seconds (41): Wait (4/5) second
[CRZ] Wait_N_Seconds (41): Wait (1/5) second	[CRZ] Wait_N_Seconds (41): Wait (5/5) second
[CRZ] Wait_N_Seconds (41): Wait (2/5) second	[CRZ] CAN_Test (132): ++ubKeyNumber: 2
[CRZ] HAL_CAN_RxCpltCallback (95):	[CRZ] Wait_N_Seconds (41): Wait (1/5) second

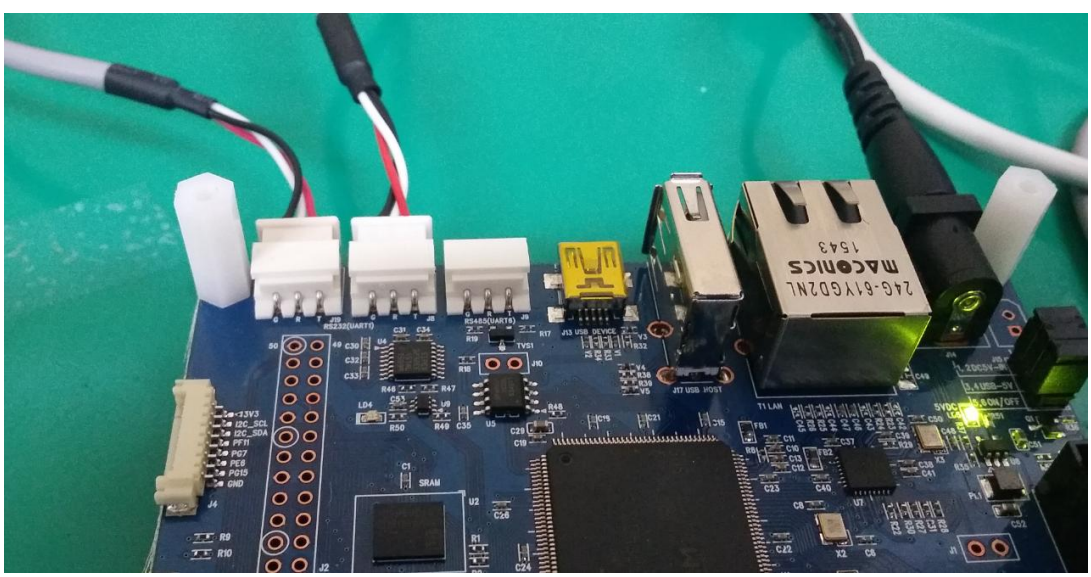
CanHandle->pRxMsg->Data[0]: 2 [CRZ] Wait_N_Seconds (41): Wait (3/5) second [CRZ] Wait_N_Seconds (41): Wait (4/5) second [CRZ] Wait_N_Seconds (41): Wait (5/5) second [CRZ] CAN_Test (132): ++ubKeyNumber: 3 [CRZ] Wait_N_Seconds (41): Wait (1/5) second [CRZ] Wait_N_Seconds (41): Wait (2/5) second **[CRZ] HAL_CAN_RxCpltCallback (95):** **CanHandle->pRxMsg->Data[0]: 4** [CRZ] Wait_N_Seconds (41): Wait (3/5) second [CRZ] Wait_N_Seconds (41): Wait (4/5) second [CRZ] Wait_N_Seconds (41): Wait (5/5) second [CRZ] Wait_N_Seconds (41): Wait (1/5) second [CRZ] Wait_N_Seconds (41): Wait (2/5) second	[CRZ] Wait_N_Seconds (41): Wait (2/5) second **[CRZ] HAL_CAN_RxCpltCallback (95):** **CanHandle->pRxMsg->Data[0]: 3** [CRZ] Wait_N_Seconds (41): Wait (3/5) second [CRZ] Wait_N_Seconds (41): Wait (4/5) second [CRZ] Wait_N_Seconds (41): Wait (5/5) second [CRZ] CAN_Test (132): ++ubKeyNumber: 4 [CRZ] Wait_N_Seconds (41): Wait (1/5) second [CRZ] Wait_N_Seconds (41): Wait (2/5) second [CRZ] Wait_N_Seconds (41): Wait (3/5) second [CRZ] Wait_N_Seconds (41): Wait (4/5) second [CRZ] Wait_N_Seconds (41): Wait (5/5) second [CRZ] Wait_N_Seconds (41): Wait (1/5) second

11. UART6 Test

9> UART6 Test

9번을 선택합니다.

11.1. RS232 Test



디버그 왼쪽 편에 하나의 포트를 더 PC에 연결합니다.

1> RS232 Test

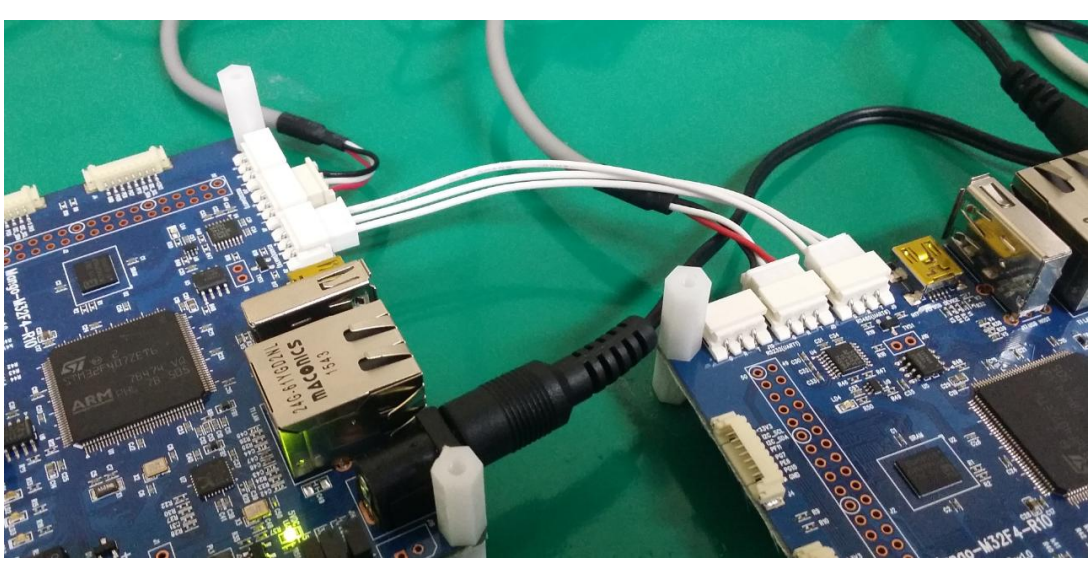
1번을 선택합니다.

아래의 결과에서 왼쪽은 디버그 창이고 오른쪽은 하나 더 연결한 UART RS232 포트 입니다.

[CRZ] UART6_Test (174): 1 is selected [CRZ] UART6_Test (189): Rx: a (0x61) [CRZ] UART6_Test (189): Rx: b (0x62) [CRZ] UART6_Test (189): Rx: c (0x63)	Test chars 1 Test chars 2 Test chars 3 Test chars 4 Test chars 5 Test chars 6 Test chars 7

1초에 한번씩 Test chars를 출력합니다. 오른쪽 창에서 입력한 글자는 왼쪽 디버그 창에 출력이 됩니다.

11.2. RS485 Test



두 개의 보드에서 RS485 포트 부분을 1:1로 연결합니다.

2> RS485 Test

2번을 선택합니다.

[CRZ] UART6_Test (174): 2 is selected abcdfsdfsdfsadfd	[CRZ] UART6_Test (174): 2 is selected abcdwefwdfsdf

왼쪽 디버그 창에서 입력한 글자는 오른쪽 창에, 오른쪽 창에서 입력한 글자는 왼쪽 창에 나타나게 됩니다.