# (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

Booting	
1.1. Hardware Conne	ction4
1.2. Debug Serial Po	<b>t</b> 4
Key Test	
LED Test	
SDMMC Test	
SRAM Test	
USB Host Test	
Ethernet HTTP Test	
USB Device Test	
RTC Calendar Test	
. CAN Test	
. UART6 Test	
11.1. RS232 Test	
11.2. RS485 Test	
	Booting.1.1.Hardware Conne1.2.Debug Serial PorKey Test.LED Test.SDMMC Test.SRAM TestUSB Host TestEthernet HTTP Test.USB Device Test.RTC Calendar Test.CAN TestUART6 Test11.1.RS232 Test.11.2.RS485 Test.

## 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



← → C 🗋 192.168.58.7/leds.cgi?	?
🏥 앱 ★ Bookmarks 🗀 G둘바 🗋 CRZ홈	💽 N카페 M GMail 🔁 GCal M PoohGmail 🕒 폭
STM32F4xx Leds con	trol
Home page	Led control
a LED you have to check/uncheck its co configuration. Finally check in the STM3	rresponding checkbox. Then you have to click a 24xG-EVAL board that you get the desired LED:
STM32 Webserver LEDs Contr	rol
STM32 Webserver LEDs Control  LED1 LED2 LED3 LED4	rol

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

12 USB HID Demonstrator (v1.0.2)	×
Device capabilities	USB HID Target HID 준수 장치 💌
Input/Output transfer	Vendor identifier 0x0483 Product identifier 0x5750 Version number 0x0200
Graphic view	
Graphic customization	Usage Page: FF Input Report Byte Length: 2 Output Report Byte Length: 2 Feature Report Byte Length: 2 Number of Link Collection Nodes: 1
Output mode	Number of Input/Button Caps:2Number of Input/Data Indices:3Number of Output Button Caps:0Number of Output Button Caps:4Number of Output Value Caps:4Number of Output Data Indices:4Number of Feature Button Caps:2Number of Feature Value Caps:5Number of Feature Data Indices:7

USB HID Demonstrator (v1.0.2)	1 ( MA )	×
USB HID Demonstrator (v1.0.2)	USB HID Target HID 준수 장치           Buttons         Variable Inputs           Leds         Image: I	

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: 02:00:02
[CRZ] RTC\_Test (132): Date: 02-18-2014, Time: 02:00:03
[CRZ] RTC\_Test (132): Date: 02-18-2014, Time: 02:00:03

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 (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	

## 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	Test chars 1
[CRZ] UART6_Test (189): Rx: a (0x61)	Test chars 2
[CRZ] UART6_Test (189): Rx: b (0x62)	Test chars 3
[CRZ] UART6_Test (189): Rx: c (0x63)	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	[CRZ] UART6_Test (174): 2 is selected
abcdfsdfsdfsadfd	abcdwefwdfsdf

왼쪽 디버그 창에서 입력한 글자는 오른쪽 창에, 오른쪽 창에서 입력한 글자는 왼쪽 창에 나타나 게 됩니다.