

이 매뉴얼은 에타일렉트로닉스 RX USER SET 과정을 사용자 입장에서 정리한 문서이다.

USER SET 프로토콜은 ETA_RX_CONTROLLER_CAN_MESSAGE_DEFINITION_V1.4 를 따른다.

1. 세팅 항목

- A. User 는 총 5 개의 항목을 세팅할 수 있다.
- B. FC_VOUT (완충 전압): 완충 상태인지를 판단하는 기준 전압
- C. FC_COMPLETE_IOUT (완충 전류): 완충 전압을 만족한 상태에서 완충임을 판단하는 기준 전류
- D. PC_CUTOFF_VOUT (*pre-charge 전압): pre-charge mode 임을 판단하는 기준 전압
 - i. *pre-charge: 과-방전된 배터리를 대상으로 세류 전류로 충전을 하며 점차 전압을 높여, 최적으로 충전이 될 수 있도록 하는 모드. 배터리의 수명, 안정성을 위해서 권장됨
- E. PC_CUTOFF_RECOVERY_VOUT (pre-charge 해제 전압): Pre-charge 로부터 해제되어 일반 충전 모드로 충전을 시작하는 기준 전압
- F. PC_IOUT (Pre-charge 전류): pre-charge 상태에서 충전하고자 하는 세류 전류의 크기

2. Physical Interface

- A. Specification : CAN V2.0A
- B. Data Ordering : Little Endian
- C. Transmission Speed : 500 kB

3. Command Protocol Structure (User to Eta Rx Module)

A. ID, Data Length, USER_SET_CODE, USER_SET_VALUE

B. VALUE 의 경우 설정하고자 하는 VALUE 에 대해서 3 bytes 로 변환하여 전송

i. Ex. Setting voltage at 28.2V = 28200mV.

1. (VALUE[0] = VALUE&0xFF = 0x28)

2. (VALUE[1] = (VALUE>>8)&0xFF = 0x6E)

3. (VALUE[2] = (VALUE>>16) = 0x00)

ID	Data Length	USER_SET_CODE (1 bytes)	USER_SET_VALUE (3 bytes)	비고
0xDC	0x04	0x01 : USER_SET_START 0x02 : USER_SET_FC_VOUT 0x03 : USER_SET_FC_COMPLETE_IOUT 0x04 : USER_SET_PC_CUTOFF_VOUT 0x05 : USER_SET_PC_CUTOFF_RECOVERY_VOUT 0x06 : USER_SET_PC_IOUT 0x07 : USER_SET_TERMINATION	If USER_SET_CODE is 0x01 : 0x01 0x02 0x03 0x02 : VALUE (3 bytes) 0x03 : VALUE 0x04 : VALUE 0x05 : VALUE 0x06 : VALUE 0x07 : 0x04 0x05 0x06	VALUE[1]=VALUE&0xFF VALUE[2]=(VALUE>>8)&0xFF VALUE[3]=(VALUE>>16)&0xFF

4. Return Protocol Structure (Eta Rx Module to User)

A. Response of USER_SET_CODE

ID	Data Length	RESPONSE_USER_SET_CODE (1 bytes)	RESPONSE_USER_SET_ECHO (3 bytes)	비고
0xCD	0x04	0x01 : RES_USER_SET_START_OK 0x02 : RES_USER_SET_FC_VOUT_OK 0x03 : RES_USER_SET_FC_COMPLETE_IOUT_OK 0x04 : RES_USER_SET_PC_CUTOFF_VOUT_OK 0x05 : RES_USER_SET_PC_CUTOFF_RECOVERY_VOUT_OK 0x06 : RES_USER_SET_PC_IOUT_OK 0x07 : RES_USER_SET_TERMINATION_OK 0xF6 : RES_USER_SET_START_ERROR 0xF7 : RES_USER_SET_FC_VOUT_ERROR 0xF8 : RES_USER_SET_FC_COMPLETE_IOUT_ERROR	If RESPONSE_USER_SET_CODE is 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07 : Value is ECHO 0xF6 : Value = 0x01 0x02 0x03 0xFC : Value = 0x04 0x05 0x06 0xF7, 0xF8, 0xF9, 0xFA, 0xFB : Value returns Default setting value Other : Value is DON'T CARE	Ex) FC_VOUT=28.0V VALUE = 28000 (mV) can.data[1]=VALUE&0xFF can.data[2]=(VALUE>>8)&0xFF can.data[3]=(VALUE>>16)&0xFF

	0xF9 : RES_USER_SET_PC_CUTOFF_VOUT_ERROR		
	0xFA : RES_USER_SET_PC_CUTOFF_RECOVERY_VOUT_ERROR		
	0xFB : RES_USER_SET_PC_IOUT_ERROR		
	0xFC : RES_USER_SET_TERMINATION_ERROR		
	0xFD : RES_USER_SET_CODE_ERROR		
	0xFE : RES_USER_SET_TIMEOUT_ERROR		

- i. USER 가 보낸 USER_SET_CODE 에 대한 응답
- ii. ID(0xCD), Data Length(0x04), RESPONSE_USER_SET_CODE(0x01~0x07, 0xF6~0xFE), RESPONSE_USER_SET_ECHO(VALUE;3bytes)

B. Response of USER_SET_RESULT

ID	Data Length	RESPONSE_USER_SET_RESULT	비고
0xCE	0x01	0x01 : RES_USER_SET_RESULT_SUCCESS 0x02 : RES_USER_SET_RESULT_FAIL	0x01 : User setting 이 성공적으로 이루어졌고, 실제 모듈에도 기록이 되었음을 의미 0x02 : User setting 실패

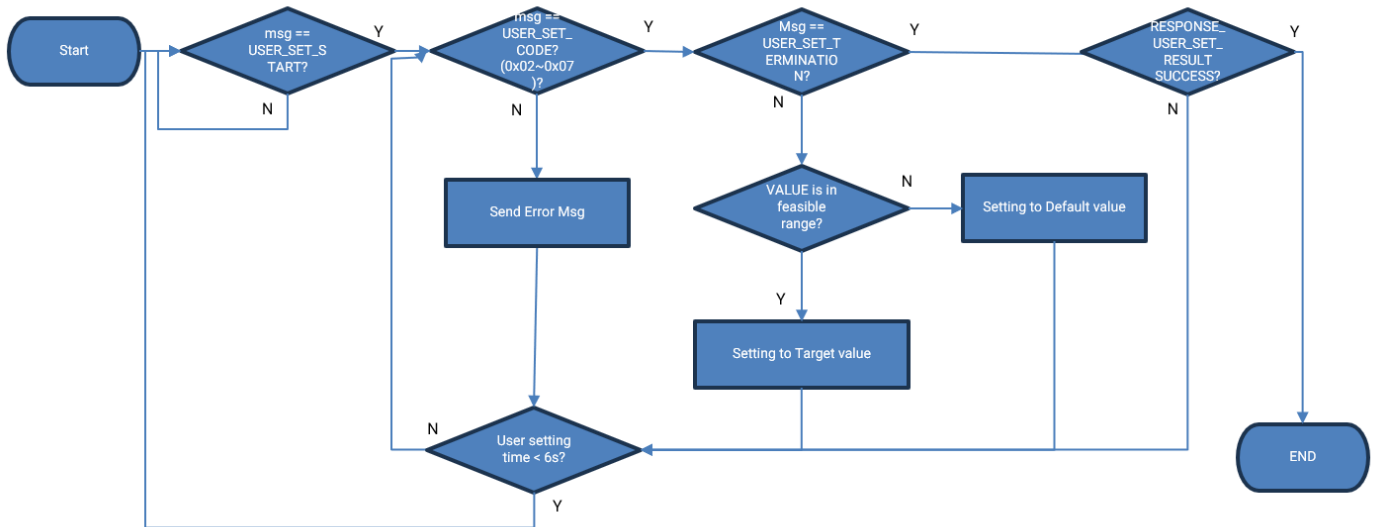
- i. 사용자가 Setting 한 결과 값이 실제로 Setting 되었는지 여부에 대한 응답
- ii. ID(0xCE), Data Length(1), RESPONSE_USER_SET_RESULT(0x01 ~ 0x02)

5. Setting Parameter 허용 가능 범위

- A. 아래 표에 있는 값은 각 셀에 적용 가능한 default value 및 Available setting range 이다.
- B. Rx 의 시리얼 넘버를 기반으로 배터리의 스펙이 정해지게 되며, 배터리 셀의 series value 를 곱해서 parameter 의 default value 및 available setting range 가 정해지게 된다.
 - i. $\text{Default_FC_VOUT} = \text{Default value of FC_VOUT} * (\# \text{ of Battery cells in series})$
 - ii. Ex) default FC_VOUT = 28.98V @ 7S, 53.82 V @ 13S
 - iii. Available FC_VOUT = 28 ~ 29.05 V @ 7S, 52 ~ 53.95 V @ 13S
- C. 해당 제품을 Setting 하지 않았을 시에는 Default value 로 값이 setting 이 되어 있다.
- D. Setting 하고자 하는 parameter 가 허용 가능 범위는 넘어갔을 시에는 RESPONSE_USER_SET_CODE 는 Error 를 반환하며 RESPONSE_USER_SET_ECHO 값은 default setting value 값을 반환한다.

Parameter	Default value	Available setting range by Cell
FC_VOUT	4.14 V	(4.0 ~ 4.15 V) (# of Battery cells in series)
FC_COMPLETE_IOUT	3.5 A	2.0 ~ 4.0 A
PC_CUTOFF_VOUT	3.0 V	2.9 ~ 3.1V
PC_CUTOFF_RECOVERY_VOUT	3.15 V	3.12 ~ 3.18
PC_IOUT	1.9 A	1.0 ~ 3.0 A

6. User setting flow (추후 업데이트 예정)



A. User 는 아래의 Protocol 의 형식으로 Command 를 전송할 수 있다.

1. ID(0xDC), Data length(4), USER_SET_CODE, USER_SET_VALUE

B. 사용자가 USER_SET 을 통해서 데이터를 변경하고 싶은 경우, **최초에는** USER_SET_CODE 로 USER_SET_START 를 보내주어야 한다.

i. User send :

1. ID(0xDC), Data length(0x04), USER_SET_CODE(0x01), USER_SET_VALUE(0x01 0x02 0x03)

ii. Message 성공 시 return:

1. ID(0xCD), Data length(0x04), RESPONSE_USER_SET_CODE(0x01), USER_SET_VALUE(0x01 0x02 0x03)

iii. Message 실패 시 return:

1. ID(0xCD), Data length(0x04), RESPONSE_USER_SET_CODE(0xF6), USER_SET_VALUE(0x01 0x02 0x03)

C. USER_SET_START 이후에 Setting 하고자 하는 PARAMETER 에 맞춰서 USER_SET_CODE 를 설정해준다. (USER_SET_CODE = 0x02(USER_SET_FC_VOUT) ~ 0x06 (USER_SET_PC_IOUT))

i. User send :

1. (ID(0xDC), Data length(0x04), USER_SET_CODE(0x02), USER_SET_VALUE (3 bytes Value))

ii. Message 성공 시 return:

1. ID(0xCD), Data length(4), RESPONSE_USER_SET_CODE(0x02 ~ 0x06), USER_SET_VALUE (3 bytes Value)

iii. Message 실패 시 return:

1. ID(0xCD), Data length(4), USER_SET_CODE(0xF7 ~ 0xFB), USER_SET_VALUE(0x01 0x02 0x03)
- D. USER_SET_VALUE 가 정상적으로 Setting 가능한 허용 범위를 넘어 갔을 경우 Error 를 return 한다.
이 경우 USER 가 setting 한 value 가 아닌 default value 가 Setting 되게 된다.
 - E. 사용자는 USER_SET_CODE 로 파라미터 setting 을 마치게 되면, USER_SET_CODE 로 USER_SET_TERMINATION 을 보내야 한다.
 - F. USER_SET_TERMINATION 에 대한 RESPONSE 를 받고 나면 Eta Rx module 은 설정된 파라미터를 Flash memory 에 저장하고 최종적으로 RESPONSE_USER_SET_RESULT 로 0x01(RES_USER_SET_RESULT_SUCCESS)를 보내게 된다.

7. 예제 (example)

A. 아래의 코드와 콘솔창은 사용자의 입장에서 예제로 작성하였다.

- i. CAN 을 통해서 파라미터를 setting 하는 메시지는 5 초에 한번씩 보내게 된다.
- ii. 보내는 명령어는 USER_SET_CODE (0x01 ~ 0x07) 까지 순차적으로 보내게 된다.

```
if (canTestCnt >= 5000) {
    canTestCnt = 0;
    canTestStep++;

    if (canTestStep == CAN_USER_SET_START) {
        printf("===== START USER SET ===== \n\r");
        eta_rx_can_send_user_set_test(CAN_USER_SET_START);
    }
    else if (canTestStep == CAN_USER_SET_FC_VOUT) {
        eta_rx_can_send_user_set_test(CAN_USER_SET_FC_VOUT);
    }
    else if (canTestStep == CAN_USER_SET_FC_COMPLETE_IOUT) {
        eta_rx_can_send_user_set_test(CAN_USER_SET_FC_COMPLETE_IOUT);
    }
    else if (canTestStep == CAN_USER_SET_PC_CUTOFF_VOUT) {
        eta_rx_can_send_user_set_test(CAN_USER_SET_PC_CUTOFF_VOUT);
    }
    else if (canTestStep == CAN_USER_SET_PC_CUTOFF_RECOVERY_VOUT) {
        eta_rx_can_send_user_set_test(CAN_USER_SET_PC_CUTOFF_RECOVERY_VOUT);
    }
    else if (canTestStep == CAN_USER_SET_PC_IOUT) {
        eta_rx_can_send_user_set_test(CAN_USER_SET_PC_IOUT);
    }
    else if (canTestStep == CAN_USER_SET_TERMINATION) {
        eta_rx_can_send_user_set_test(CAN_USER_SET_TERMINATION); canTestStep = 0;
    }
}
```

- iii. 사용자 입장에서 CAN 을 통해 받은 메시지는 다음과 같이 콘솔로 표시를 하며

```
printf("  READ ID = 0x%X, CODE : 0x%X\n\r", msg->can_id, msg->data[0]);
printf("  READ DATA : ");
for (int i=1; i<msg->can_dlc; i++) {
    printf("0x%X ", msg->data[i]);
}
printf("\n\r");

if (msg->can_id == 0xCE) {
    printf("===== END USER SET ===== \n\r\n\r");
}
```

- iv. 사용자 입장에서 CAN 을 통해 SETTING 을 한 메시지는 다음과 같이 콘솔로 표시한다

```
printf("  send id = 0x%x, code : 0x%x\n\r", canMsg1.can_id, canMsg1.data[0]);
printf("  send data : ");
for (int i=1; i<canMsg1.can_dlc; i++) {
    printf("0x%x ", canMsg1.data[i]);
}
printf("\n\r\n\r");
```

B. 결과

===== START USER SET =====

```
set start
send id = 0xdc, code : 0x1
send data : 0x1 0x2 0x3
READ ID = 0xCD, CODE : 0x1
READ DATA : 0x1 0x2 0x3
```

```
set fc_vout
send id = 0xdc, code : 0x2
send data : 0x48 0x71 0x0
READ ID = 0xCD, CODE : 0x2
READ DATA : 0x48 0x71 0x0
```

```
set fc_complete_iout
send id = 0xdc, code : 0x3
send data : 0xac 0xd 0x0
READ ID = 0xCD, CODE : 0x3
READ DATA : 0xAC 0xD 0x0
```

```
set pc_cutoff_vout
send id = 0xdc, code : 0x4
send data : 0x8 0x52 0x0
READ ID = 0xCD, CODE : 0x4
READ DATA : 0x8 0x52 0x0
```

```
set pc_cutoff_recovery_vout
send id = 0xdc, code : 0x5
send data : 0x22 0x56 0x0
READ ID = 0xCD, CODE : 0x5
READ DATA : 0x22 0x56 0x0
```

```
set pc_iout
send id = 0xdc, code : 0x6
send data : 0x6c 0x7 0x0
READ ID = 0xCD, CODE : 0x6
READ DATA : 0x6C 0x7 0x0
```

```
set termination
send id = 0xdc, code : 0x7
send data : 0x4 0x5 0x6
READ ID = 0xCD, CODE : 0x7
READ DATA : 0x4 0x5 0x6
```

```
READ ID = 0xCE, CODE : 0x1
READ DATA :
```

===== END USER SET =====

```
READ ID = 0xCB, CODE : 0x0
READ DATA : 0x0 0x0 0x0 0x0 0x0 0x0 0x0
```

```
READ ID = 0xCC, CODE : 0x0
READ DATA : 0x0 0x0
```


i. (SET START; 파라미터 셋팅 시작)

1. 사용자 송신: ID(0xDC), USER_SET_CODE(0x01; USER_SET_START), USER_SET_VALUE(0x01 0x02 0x03)
2. Eta Rx 모듈 회신: ID(0xCD), RESPONSE_USER_SET_CODE(0x01; RES_USER_SET_START_OK), RESPONSE_USER_SET_ECHO(0x01,0x02,0x03)
3. 파라미터 셋팅을 시작하기 위한 준비 완료

ii. (SET FC_VOUT; 완충 전압 셋팅)

1. 사용자 송신: ID(0xDC), USER_SET_CODE(0x02; USER_SET_FC_VOUT), USER_SET_VALUE(0x48 0x71 0x00)
A. 셋팅 완충 전압 value: 29000mv = (0x48 || (0x71 < <8))
2. Eta Rx 모듈 회신: ID(0xCD), RESPONSE_USER_SET_CODE (0x02; RES_USER_SET_FC_VOUT_OK), RESPONSE_USER_SET_ECHO(0x48,0x71,0x00)
3. 완충 전압 29.0V 셋팅 완료

iii. (SET FC_COMPLETE_IOUT; 완충 전류 셋팅)

1. 사용자 송신: ID(0xDC), USER_SET_CODE(0x03; USER_SET_FC_COMPLETE_IOUT), USER_SET_VALUE(0xAC 0x0D 0x00)
A. 셋팅 완충 전류 value: 3500 mA = (0xAC || (0x0D < <8))
2. Eta Rx 모듈 회신: ID(0xCD), RESPONSE_USER_SET_CODE (0x03; RES_USER_SET_FC_COMPLETE_IOUT_OK), RESPONSE_USER_SET_ECHO(0xAC,0x0D,0x00)
3. 완충 전류 3.5 A 셋팅 완료

iv. (SET PC_CUTOFF_VOUT; pre-charge 전압 셋팅)

1. 사용자 송신: ID(0xDC), USER_SET_CODE(0x04; USER_SET_PC_CUTOFF_VOUT), USER_SET_VALUE(0x08 0x52 0x00)
A. Pre-charge 전압 value: 21000mv = (0x08 || (0x52 < <8))
2. Eta Rx 모듈 회신: ID(0xCD), RESPONSE_USER_SET_CODE (0x04; RES_USER_SET_PC_CUTOFF_VOUT_OK), RESPONSE_USER_SET_ECHO(0x08,0x52,0x00)
3. Pre-charge 전압 21.0V 셋팅 완료

v. (SET PC_CUTOFF_RECOVERY_VOUT; pre-charge 해제 전압 셋팅)

1. 사용자 송신: ID(0xDC), USER_SET_CODE(0x05; USER_SET_PC_CUTOFF_RECOVERY_VOUT), USER_SET_VALUE(0x22 0x56 0x00)
A. 셋팅 pre-charge 해제 전압 value: 22050mv = (0x22 || (0x56 < <8))

2. Eta Rx 모듈 회신: ID(0xCD), RESPONSE_USER_SET_CODE (0x05;
RES_USER_SET_PC_CUTOFF_RECOVERY_VOUT_OK),
RESPONSE_USER_SET_ECHO(0x22,0x56,0x00)
 3. Pre-charge 해제 전압 22.05 V 셋팅 완료
- vi. (SET PC_IOUT; pre-charge 전류 셋팅)
1. 사용자 송신: ID(0xDC), USER_SET_CODE(0x06; USER_SET_PC_IOUT), USER_SET_VALUE(0x6C
0x07 0x00)

A. 셋팅 pre-charge 전류 value: 1900 mA = (0x6C || (0x07 < < 8))
 2. Eta Rx 모듈 회신: ID(0xCD), RESPONSE_USER_SET_CODE (0x06;
RES_USER_SET_PC_IOUT_OK), RESPONSE_USER_SET_ECHO(0x48,0x71,0x00)
 3. Pre-charge 전류 1.9 A 셋팅 완료
- vii. (SET TERMINATION; 셋팅 파라미터 기록 및 완료 신호 전송)
1. 사용자 송신: ID(0xDC), USER_SET_CODE(0x07; USER_SET_TERMINATION),
USER_SET_VALUE(0x04 0x05 0x06)
 2. Eta Rx 모듈 회신: ID(0xCD), RESPONSE_USER_SET_CODE (0x07;
RES_USER_SET_TERMINATION_OK), RESPONSE_USER_SET_ECHO(0x04,0x05,0x06)
 3. TERMINATION COMMAND 전송 완료
 4. Eta Rx 모듈 회신: ID(0xCE), RESPONSE_USER_SET_RESULT (0x01;
RES_USER_SET_RESULT_SUCCESS)