Model: FES070E7

ARM based Panel PC

User Manual

(Preliminary)

Fanless ultra-compact
Dual Core Cortex A72+Quad core Cortex A53
ARM system for
IIoT/HMI/Vehicle

Applicable Products:

E71Cxx110-xxxS00

Version:	V1.1
Document No:	doc-FES070E7-RK39xxxxxV11
Main Board:	MBE71C-R39xxxSS

Packing List

☑ 1 x FES070E7 device.

☑ 1 x Coated Adhesive Foam Rubber set, 4 x Mounting Clips, 6 x Panel mount Screws.

☑ 1 x Power line 2-pole phoenix Jack to DC Jack-Ø2, 1 x 3-pole phoenix Jack(female).

☑ 2 x Antenna (Only for the product code with "S01", "S02", "S03" or "S0H/L").

Ordering Information

Model Number: FES070E7

- ***070*: Panel Size 7”**
- ***E7*: PCAP + Plastic Bezel with metal rear case**

Product Number: E71Cmn110-abcS00

- **E71C:** Panel Size 7” with PCAP, Plastic Bezel with metal case.
  With Arm Processor Dual Cortex-A72 + Quad Cortex-A53, (0°C ~ 60°C).
- **110:** mainboard version.
- **S00:** Full RS232 x2, DI x3, DO x3, CANbus x2, Grove IOs.

- **m: On Board DDR3 Memory Size**
  1 1GB of LPDDR3.
  2 2GB of LPDDR3. (default)
  4 4GB of LPDDR3.

- **n: On Board eMMC Size**
  8 8GB of eMMC.
  F 16GB of eMMC. (default)
  J 32GB of eMMC.

- **a : Power input Selection**
  K DC12V input, DC Jack-Ø2.
  W DC9-36V input, 3-pole phoenix Jack with ignition control.
  D DC9-36V input, 2-pole phoenix Jack without ignition control.
  P 802.3at Type 2 (PoE+) input, RJ45 with Backup Power 2-pole Phoenix Jack.

- **bc: OS version**
  A8 Android 8.0
  D9 Debian 9
  U0 Ubuntu

Table of Contents

Copyright Notice

Trademarks

Disclaimer

Forenex Technology Ltd. follows the continuous development strategy, reserves the right to make irregular changes to the information, product specifications, hardware and software descriptions in this document at any time without notice. However, the information provided in this document is believed to be accurate and reliable as of the publication date of this document.

The company is not responsible for any patent infringement caused by the use or abuse of the information in this document.

Warranty Description

The starting date of our warranty is recorded on the product serial number. Within one year from the date of delivery, if the customer finds a defect, we will choose to repair or replace the defective product free of charge, provided that the customer has prepaid freight. In the product serial number, the fourth and fifth digits indicate the year of manufacture, the sixth digit indicates the month (1- 9 and “A” for October, “B” for November, “C” for December), and the seventh digit for the week number of the month.

(e.g., the serial number xxx10C3xxxxxxx denotes the third week of December 2010).

RMA Service

You may need the following information ready before RMA procedure.

- Product number & serial number
- Software (OS version, application software, etc.)
- Description of complete problem
- The exact error messages that show up on screen

In addition, for the software compatible issue that we suggest to visit the website of our distributor to find the update information about the product first. Normally, the way might help solve application software problem quickly.

Precautions for use

- More frequent and larger data access on eMMC memory makes the eMMC life span shorter. Therefore, it is highly recommended to use a Micro SD card for large data access.
- When both DC-in and PoE+ are connected to the FES070E7, the connection with higher voltage will be the main power source.
- The Operating Temperature is a result of the test performed in experimental chamber. It is highly suggested to execute a solid testing under actually application environment.

Safety Statement

FCC-A Radio Frequency Interference Statement

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Shielded interface cables and A.C. power cord, if any, must be used in order to comply with the emission limits.
- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Notice:

The product described in this document is designed for general use, Forenex Technologies assumes no responsibility for the conflicts or damages arising from incompatibility of the product. Check compatibility issue with your local sales representatives before placing an order.

1. General Information

The FES070E7 (Panel mount PC) is designed with an ABS plastic bezel, metal rear cover chassis and the PCAP touch panel covers the front of FES070E7 to able to adapt front-IP65 ingress protection. Also due to take full advantage of the lowest power, fanless system design for a wide variety of industrial automation, transportation, door-control, IIoT and HMI applications.

1-1. Product Specifications

Display:

- ✓ 7.0”(Dimension), 1024 x 600 pixels(Resolution)
- ✓ 450 cd/m² (Brightness), 800:1(Contrast Rate), 80°/80°/80°/80°(Viewing Angle)
- ✓ Touch Panel(Multi-fingers Project Capacitive)

Processor:

- ✓ Arm Dual Cortex-A72 @2.0Ghz + Quad Cortex-A53 @ 1.5Ghz Core

System Memory:

- ✓ 1GB/2GB(default)/4GB

Storage(eMMC):

- ✓ 8GB/16GB(default)/32GB

I/O Interfaces:

- ✓ 1 x Controllable surround status light bar (RGB optic fiber)
- ✓ 1 x HDMI
- ✓ 1 x Gigabit Ethernet Port with PoE+(Optional)
- ✓ 1 x USB 3.0 Type A
- ✓ 1 x USB 2.0 Type A
- ✓ 1 x Micro-SD Card Slot
- ✓ 2 x Full modem RS232 port with 3-In/ 3-out DIO
- ✓ 1 x Terminal Block Power Connector(refer to Ordering Information)

Operating System:

- ✓ Android 7.1/ 8.0
- ✓ Debian 9
- ✓ Ubuntu

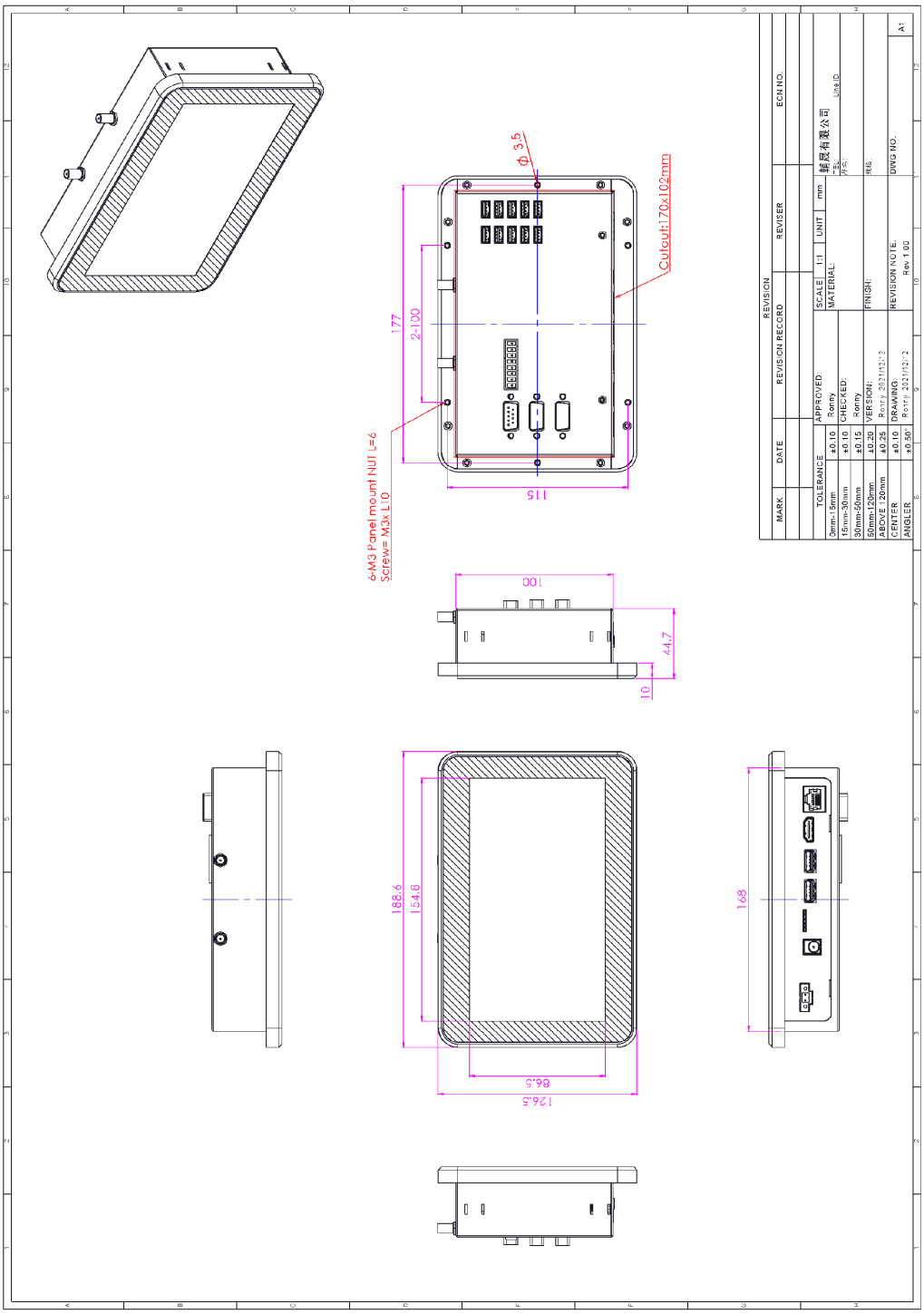
1-2. Optional Functions

- ✓ Front CMOS Camera 5-megapixel module
- ✓ IEEE 802.11a/b/g/n/ac, 2.4G/5Ghz Wi-Fi module, 2x2 MIMO standard + BT 4.1
- ✓ 4G LTE-A Cat6 module with Micro-SIM socket carrier board
- ✓ Secondary 10/100M Ethernet module
- ✓ CANBUS 2.0 module
- ✓ Grove IO port module

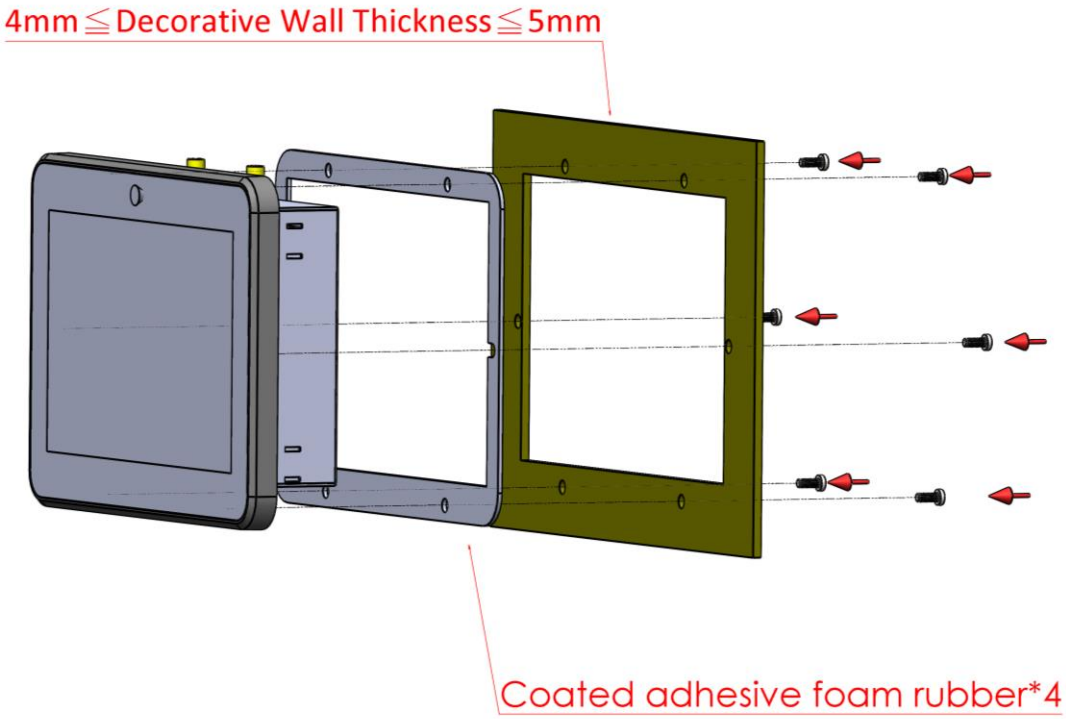
1-3. Mechanical Specification

1-3-1. The Outline Dimension

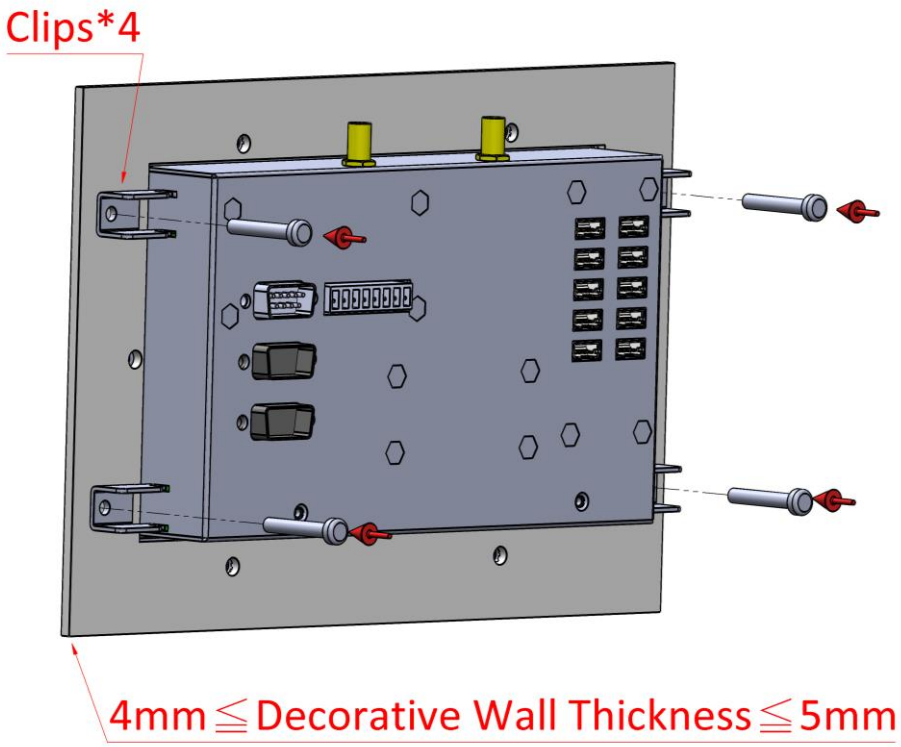
- The dimension adapts to all of FES070E7 model.



1-3-2. Wall Mounting: Panel-Mount Illustration



1-3-3. Wall Mounting: Clips Mounting Illustration

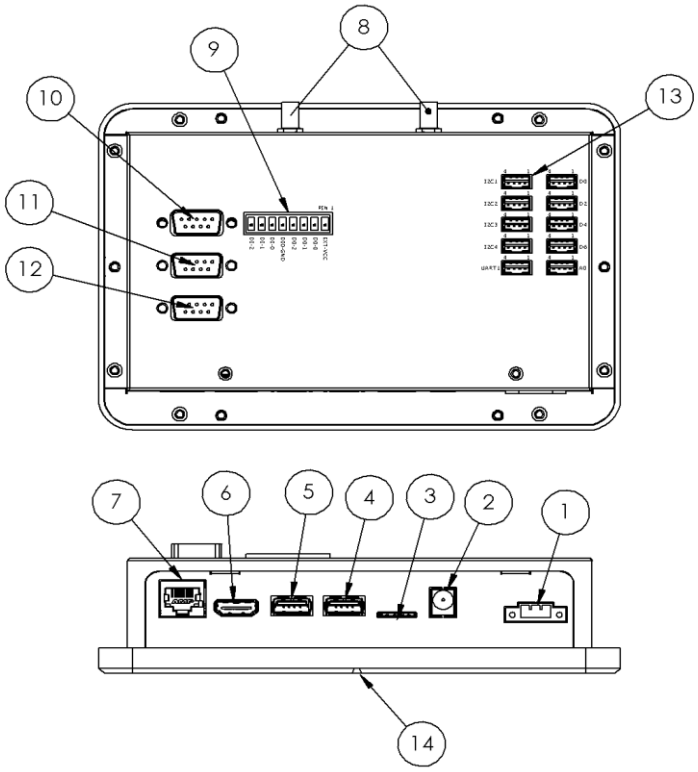


2. Peripherals Port Description

2-1. External IO port placement of FES070E7

- The various peripherals refer to individual model specification.

Product_number	E71CZF110-WxxS00	E71CZF110-PxxS00	E71CZF110-KxxS00
Item	Fuction		
1	DC9-36V Power input	DC-12V Backup Power input	NC
2	NC	NC	DC Jack/Ø2.0
3	Micro SD Slot	Micro SD Slot	Micro SD Slot
4	USB 2.0 Port	USB 2.0 Port	USB 2.0 Port
5	USB 3.0 Port	USB 3.0 Port	USB 3.0 Port
6	HDMI 2.0 Port	HDMI 2.0 Port	HDMI 2.0 Port
7	GigaE Port	GigaE Port with PoE+	GigaE Port
8	Wi-Fi Antenna(Option)	Wi-Fi Antenna(Option)	Wi-Fi Antenna(Option)
9	DIO Port(3xDI, 3xDO)	DIO Port(3xDI, 3xDO)	DIO Port(3xDI, 3xDO)
10	COM1 Port	COM1 Port	COM1 Port
11	COM2 Port	COM2 Port	COM2 Port
12	CANbus Port(Option)	CANbus Port(Option)	CANbus Port(Option)
13	Grove IO Port(Option)	Grove IO Port(Option)	Grove IO Port(Option)
14	Camera(Option)	Camera(Option)	Camera(Option)

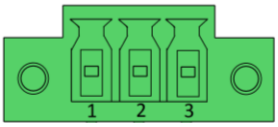


2-2. DC Power Connector

E71Cxx110-WxxS00 (The model number with wide range power input)

- On front panel, main power DC9-36V input.

- Pin Assignment:



Connector: (Terminal Block-3P/3.81mm/Male)	
Pin number	**Description**
1	GND
2	DCIN
3	Ignition control, high state 9V-36V acceptable

E71Cxx110-PxxS00 (The model number GigaE with PoE+)

- On front panel, external backup power input.

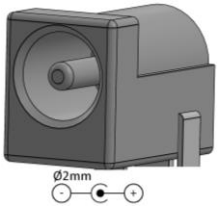
- Pin Assignment:

Connector: (Terminal Block-2P/3.81mm/Male)	
Pin number	**Description**
1	GND
2	DC12V

E71Cxx110-KxxS00

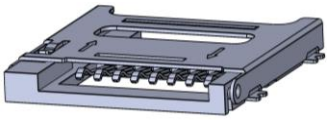
- On front panel, external DC Adaptor input.

- Pin Assignment:



Connector: (Power Barrel Jack6.3mm/ Tip2.0mm)	
Pin number	**Description**
Sleeve	GND
Tip	DC12V

2-3. Micro SD/SDHC card Slot

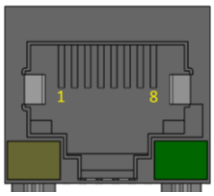


- On Front panel.
- Adapted card size(11 x 15 x 1.0 mm)
- Micro SD/SDHC card slot without spring and enable the SD storage up to 32GB size.

2-4. Gigabit Ethernet Port

For E71Cxx110-WxxS00, E71Cxx110-KxxS00

- On front panel.



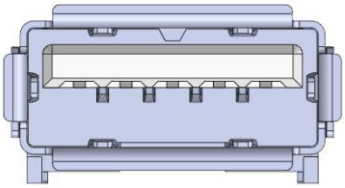
- The RJ45 port that supports high-speed Gigabit Ethernet and conform to the 802.3at specification.
- The integrated 8-pin Gigabit Ethernet port is using an 8 Position 8 Contact (8P8C) receptacle connector (commonly referred to as RJ-45).
- The Gigabit Ethernet port (RJ-45 port) has two individual LED indicators located on the front side to show:
  - Active LED is blinking in green color means activity of data flow IN or OUT of the device.
  - Link LED is in Red color means devices is operating in speeds 10/100Mbps. Link LED is in Green color means devices is operating in speeds 1000Mbps.

For E71Cxx110-PxxS00 (GigaE with PoE+ version)

- The RJ45 port that supports high-speed Gigabit Ethernet and conform to the 802.3at specification. The Inner 802.3at PoE+ board is Powered Device (PD) controller and switching regulator for high power IEEE 802.3at(25W) and 802.3af(12W) applications.
- The integrated 8-pin Gigabit Ethernet port is using an 8 Position 8 Contact (8P8C) receptacle connector (commonly referred to as RJ-45).
- The Gigabit Ethernet port (RJ-45 port) has two individual LED indicators located on the front side to show:
  - Active LED is blinking in green color means activity of data flow IN or OUT of the device.
  - Link LED is in Red color means devices is operating in speeds 10/100Mbps. Link LED is in Green color means devices is operating in speeds 1000Mbps.

2-5. USB 2.0 Port

- On Front panel.

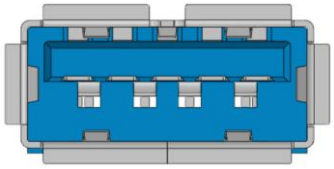


- Provides a USB 2.0 port gives complete hot plug capability and complies with USB UHCI, Rev. 2.0.
- Pin Assignment:

Connector: (USB Type A)	
Pin number	**Description**
1	+5V/500mA
2	USB Data -
3	USB Data +
4	GND

2-6. USB 3.0 Port

- On Front panel.

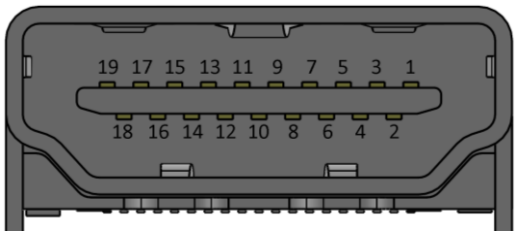


- The dedicated USB port to in charge of updating the Image File while the FES070E7 board being enter update mode.
- Under Linux OS environment, the USB3.0 default as host and gives complete hot plug capability and complies with USB xHCI, Rev. 3.x(USB 3.1 Gen 1).
- Under Android OS environment, the USB3.0 can be defined as host or device.
- Pin Assignment:

Connector: (USB 3.0 Type A)			
Pin number	**Description**	**Pin number**	**Description**
1	VBUS	6	SSRX+
2	D-	7	GND
3	D+	8	SSTX-
4	ID	9	SSTX+
5	GND		

2-7. HDMI®-2.0 port

- On front panel.
- The HDMI port uses an HDMI Type-A receptacle connector. It allows connecting the digital video devices which utilize a high definition video signal without a HDCP.
- Support HDMI® V2.0, Res. up to 4Kx2K @60fps, HDCP1.4 /2.2

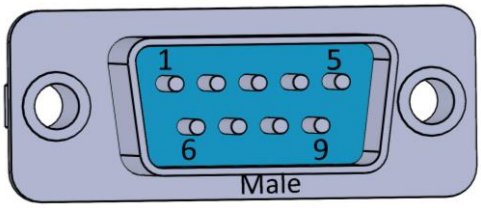


- Pin Assignment:

Connector: (19-pin HDMI Type A)			
Pin number	**Signal**	**Pin number**	**Signal**
1	TMDA_Data2+	2	Data2_GND
3	TMDA_Data 2-	4	TMDA_Data1+
5	Data1_GND	6	TMDA_Data1-
7	TMDA_Data0+	8	Data0_GND
9	TMDA_Data0-	10	TMDA_CLK+
11	CLK_GND	12	TMDA_CLK-
13	NC	14	NC
15	DDC-SCL	16	DDC-SDA
17	CEC GND	18	Power 5V supply
19	Hot Plug Detect		

2-8. Dual full modem RS232 Port

- On rear side.

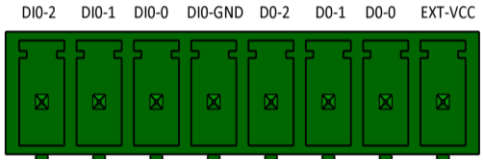


- Pin Assignment.

Connector: (DTE-DB9/Male /Blue Color)	
Pin number	**Description**
1	DCD
2	RXD
3	TXD
4	DTR
5	COM-GND
6	DSR
7	RTS
8	CTS
9	RI

2-9. DIO Port(3-IN/3-OUT)

- On rear side.

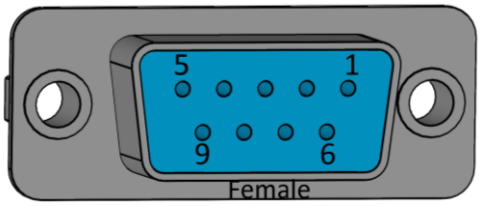


- Pin Assignment.

Connector: (Terminal Block-8P/Male)	
Pin	**Description**
1	VCC (External voltage input provides for the pull-high level of pin DO-0,1,2 use)
2	DO-0 (Open Drain output)
3	DO-1 (Open Drain output)
4	DO-2 (Open Drain output)
5	DIO-GND
6	DI-0 (the input range of 5V-36V)
7	DI-1 (the input range of 5V-36V)
8	DI-2 (the input range of 5V-36V)

2-10.CAN bus Port(option)

- On rear side.

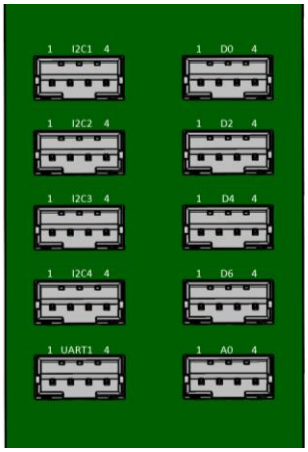


- Pin Assignment.

Connector: (DTE-DB9/Female /Blue Color)			
Pin number	**Description**	**Pin number**	**Description**
1	CANH1	6	CAN2-GND
2	CANL1	7	NC
3	CAN1-GNG	8	NC
4	CANH2	9	NC
5	CANL2		

2-11.Grove IO Port(Option)

- On rear side.



- Pin Assignment.

Pin	Connector: (JST-4P-2.0mm/ 180°)			
	Port-I2C1/ I2C2/ I2C3/ I2C4	Port-D0/D2/D4/D6	Port-A0	Port-UART
1	GND	GND	GND	GND
2	3.3V	3.3V	3.3V	3.3V
3	SDA	GPIO1/GPIO3/ GPIO5/ GPIO7	ADC1, 10bit, full scale 1.8V	TXD
4	SCL	GPIO0/GPIO2/ GPIO4/GPIO6	ADC0, 10bit, full scale 1.8V	RXD

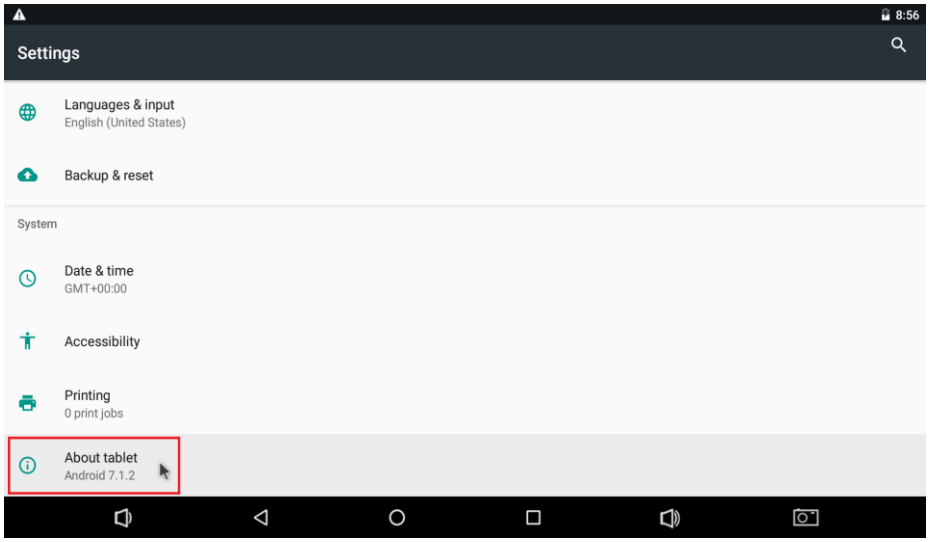
3. Software and Technical Supports

3-1. Android Programming Guide

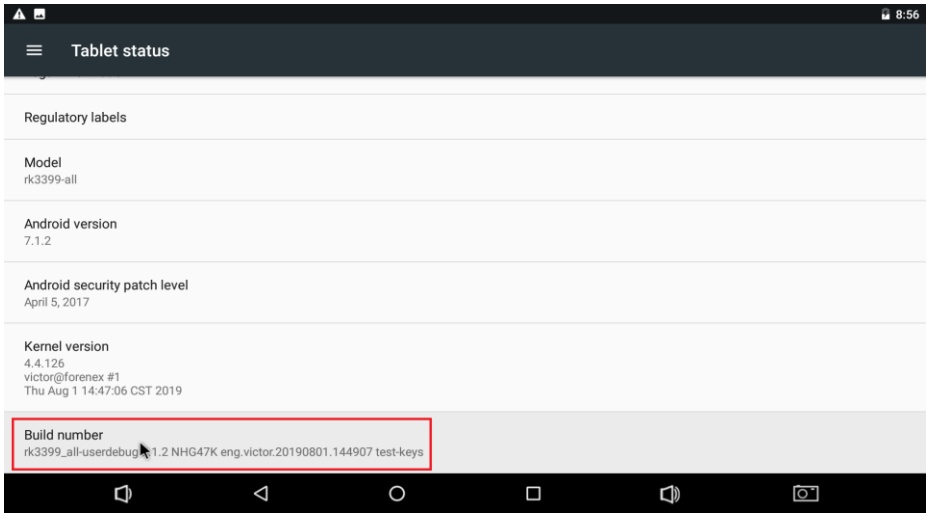
3-1-1. ADB installation

Originally, the USB3.0 was defaulted as host mode. Before use function ADB, the USB 3.0 Port have to be set to device mode by following steps.

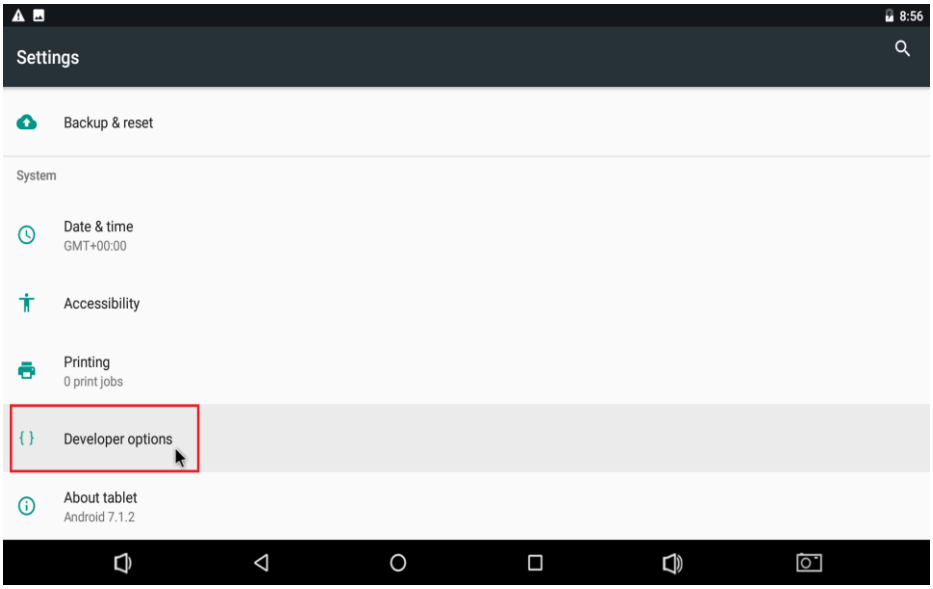
Step1. Scroll to "Settings > About Tablet"



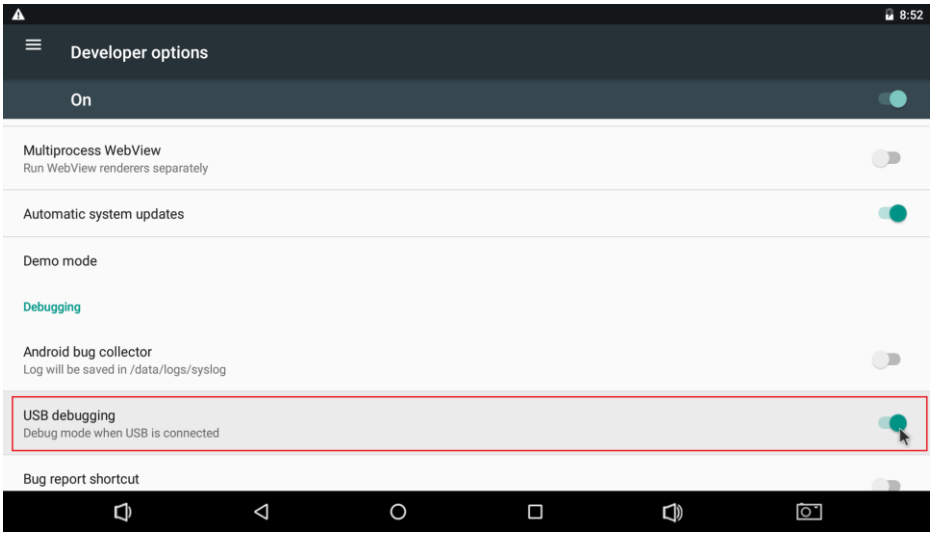
Step2. From “Settings” select "About tablet" to enter the dialog, and then click "Build number" in the dialog as shown. Android will pop up a countdown message. Keep clicking it until zero for Android to authorize the user to be a Developer.



Step3. After finished above action, a new item "Developer Options" will appear in the system block.

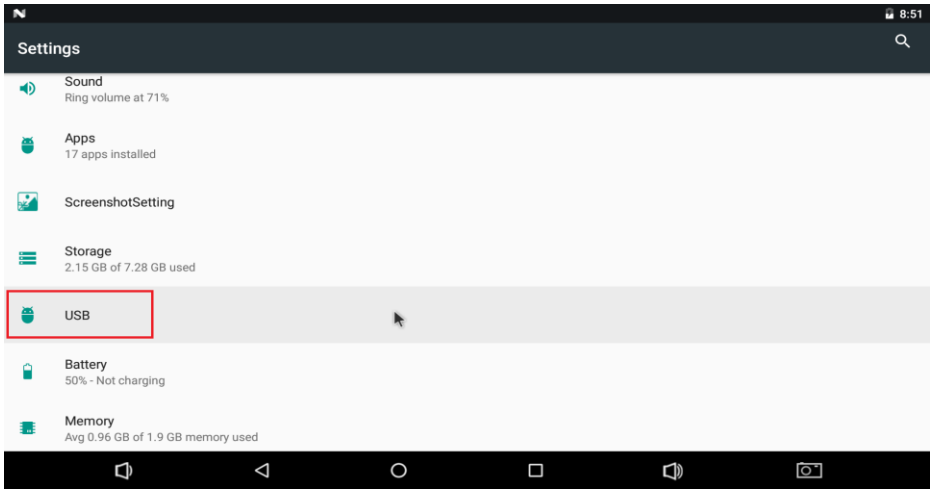


Step4. Get into the new item "Developer Options" and turn on the USB debugging function.

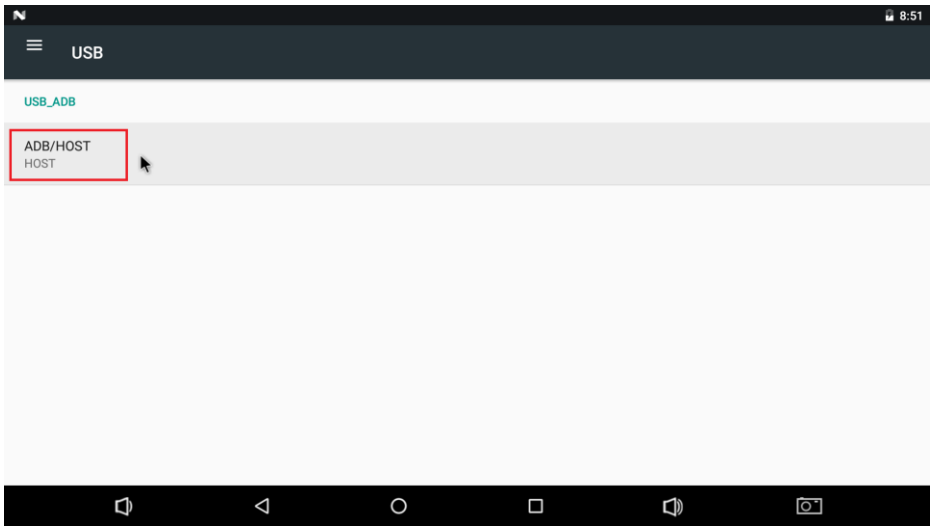


Note: Please do not change the others that you do not understand what it does.

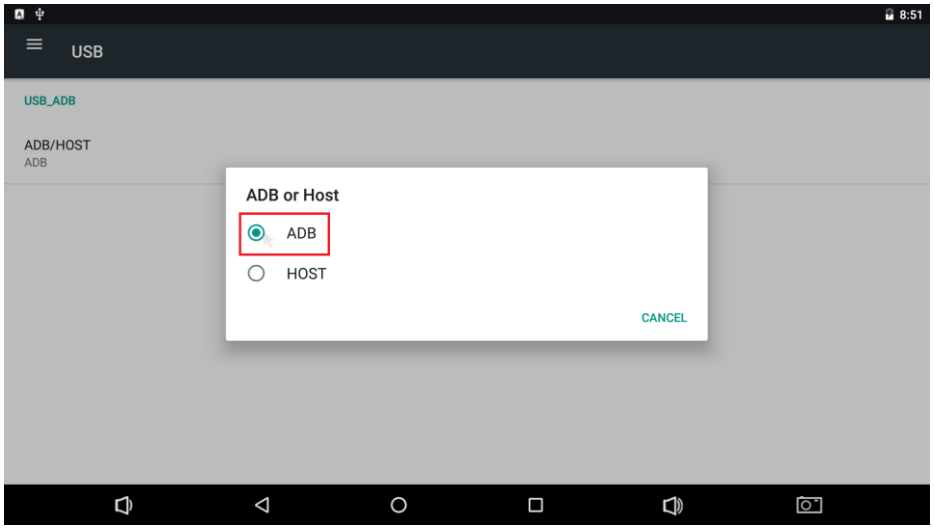
Step5. Scroll to "Settings > USB"



- Then Scroll to "USB > ADB/HOST"



- Set USB3.0 as ADB



3-1-2. To install APK software over the ADB function of PC :

1. Complete the connection between USB3.0 (Type A) port of FES070E7 and USB port of PC.
2. Enter the command string "adb install xxxxx.apk" from pc that will begin user's APK software installation.

3-1-3. To control DIO Ports:

Refer to the API files.

FR_u2r2_a12.cpp

FR_u2r2_a12.h

```
// First initial gpio
int FR_u2r2_gpio_init(eGPIO_value);

// To set each PINs DO-0, DO-1, DO-2
void set_FR_u2r2_gpio_out0 (eGPIO_value); // set DO-0
void set_FR_u2r2_gpio_out1 (eGPIO_value); // set DO-1
void set_FR_u2r2_gpio_out2 (eGPIO_value); //set DO-2



// Retrieve the status of PINs DO-0, DO-1 and DO-2 for confirmation
int get_FR_u2r2_gpio_out0 ();
int get_FR_u2r2_gpio_out1 ();
int get_FR_u2r2_gpio_out2 ();

// To get value of PINs DI-0, DI-1, DI-2
int get_FR_u2r2_gpio_in0 (); // read DI-0
int get_FR_u2r2_gpio_in1 (); // read DI-1
int get_FR_u2r2_gpio_in2 (); // read DI-2
```

3-1-4. To control COM port Mode:

FR_u2r2 generate two devices node, the path are "/dev/ttyUSB0" and "/dev/ttyUSB1".

Refer to the API files.


FRu2r2_comport_a.cpp


FRu2r2_comport_a.h

```
// To set ttyUSB0 comport mode
void set_FRu2r2_ttyUSB0_mode(comport_mode);

// To set ttyUSB1 comport mode
void set_FRu2r2_ttyUSB0_mode(comport_mode);
```

MODE_2	MODE_1	MODE_0	Mode	Status
0	0	0	RS-422 Full Duplex	1T/1R RS-422
0	0	1	Pure RS-232	3T/5R RS-232.
0	1	0	RS-485 Half Duplex	1T/1R RS-485, TX ENABLE Low Active
0	1	1	RS-485 Half Duplex	1T/1R RS-485, TX ENABLE High Active
1	0	0	RS-422 Full Duplex	1T/1R RS-422 with termination resistor and bias resistor.
1	0	1	Pure RS-232	1T/1R RS-232 co-exists with RS485 application without the need for the bus switch IC (for special usage).
1	1	0	RS-485 Half Duplex	1T/1R RS-485 with termination resistor and bias resistor. TX ENABLE Low Active
1	1	1	Low Power Shutdown	All I/O pins are High Impedance

(Table 1)

3-1-5. To control CAN bus Port:

FR_u2c2 generate two devices node, the path are "/dev/can0" and "/dev/can1".

Refer to the API files.


FR_u2c2_a.cpp


FR_u2c2_a.h

```
// First initial CAN bus
void init_FR_u2c2_can();

// To send data from CAN0 and CAN1
void FR_u2c2_can0_tx(int,int,char*);
void FR_u2c2_can1_tx(int,int,char*);

// Receive data from CAN0 and CAN1
QStringList get_FR_u2c2_can0_rx();
QStringList get_FR_u2c2_can1_rx();
```

3-1-6. To control Grove I/O Port:

FR_gro1 generate devices node, the path are "/dev/ttyS4".

Refer to the API files.


fr_gro1_a.cpp


fr_gro1_a.h

➢ The UART part of Grove I/O port

```
// To init UART_ttyS4
bool init_FR_gro1_uart();

// To send data from UART_ttyS4
void tx_FR_gro1_uart(QString data);

// To receive data from UART_ttyS4
QString rx_FR_gro1_uart();
```

➢ The I2C part of Grove I/O port

```
// To write data from I2C1 & I2C2 & I2C3 & I2C4
void set_FR_gro1_i2c1(int address, int value);
void set_FR_gro1_i2c2(int address, int value);
void set_FR_gro1_i2c3(int address, int value);
void set_FR_gro1_i2c4(int address, int value);

//To read data from I2C1 & I2C2 & I2C3 & I2C4
QString get_FR_gro1_i2c1(int address);
QString get_FR_gro1_i2c2(int address);
QString get_FR_gro1_i2c3(int address);
QString get_FR_gro1_i2c4(int address);

//To get value from ADC0 & ADC1
QString get_FR_gro1_adc0();
QString get_FR_gro1_adc1();
```

➢ The GPIO part of Grove I/O port

```
//Set each GPIO(0 - 7) as input port
void set_FR_gro1_gpio0_in();
void set_FR_gro1_gpio1_in();
void set_FR_gro1_gpio2_in();
void set_FR_gro1_gpio3_in();
void set_FR_gro1_gpio4_in();
void set_FR_gro1_gpio5_in();
void set_FR_gro1_gpio6_in();
void set_FR_gro1_gpio7_in();

//Set each GPIO(0 - 7) as output port
void set_FR_gro1_gpio0_out();
void set_FR_gro1_gpio1_out();
void set_FR_gro1_gpio2_out();
void set_FR_gro1_gpio3_out();
void set_FR_gro1_gpio4_out();
void set_FR_gro1_gpio5_out();
```

```
void set_FR_gro1_gpio6_out();
void set_FR_gro1_gpio7_out();

// Output to each GPIO(0 - 7)
void set_FR_gro1_gpio0(eGPIO_value);
void set_FR_gro1_gpio1(eGPIO_value);
void set_FR_gro1_gpio2(eGPIO_value);
void set_FR_gro1_gpio3(eGPIO_value);
void set_FR_gro1_gpio4(eGPIO_value);
void set_FR_gro1_gpio5(eGPIO_value);
void set_FR_gro1_gpio6(eGPIO_value);
void set_FR_gro1_gpio7(eGPIO_value);

//Get from each GPIO(0 - 7)
QString get_FR_gro1_gpio0();
QString get_FR_gro1_gpio1();
QString get_FR_gro1_gpio2();
QString get_FR_gro1_gpio3();
QString get_FR_gro1_gpio4();
QString get_FR_gro1_gpio5();
QString get_FR_gro1_gpio6();
QString get_FR_gro1_gpio7();
```

3-2. Debian Programming Guide

3-2-1. To control DIO Ports:

Refer to the API files.


FR_u2r2_12.c


FR_u2r2_12.h


libfintek_api.a

```
// First initial gpio
  int FR_u2r2_gpio_init (eGPIO_value);
```


fintek_api.h

```
// To set each PINs DO-0, DO-1, DO-2
void set_FR_u2r2_gpio_out0 (eGPIO_value); // set DO-0
void set_FR_u2r2_gpio_out1 (eGPIO_value); // set DO-1
void set_FR_u2r2_gpio_out2 (eGPIO_value); // set DO-2

// Retrieve the status of PINs DO-0, DO-1 and DO-2 for confirmation
int get_FR_u2r2_gpio_out0 ();
int get_FR_u2r2_gpio_out1 ();
int get_FR_u2r2_gpio_out2 ();

// To get value of PINs DI-0, DI-1, DI-2
int get_FR_u2r2_gpio_in0 (); // read DI-0
int get_FR_u2r2_gpio_in1 (); // read DI-1
int get_FR_u2r2_gpio_in2 (); // read DI-2
```

3-2-2. To control COM port Mode:

FR_u2r2 generate two devices node, the path are "/dev/ttyUSB0" and "/dev/ttyUSB1".

Refer to the API files.


FR2ur2_comport.c


FR2ur2_comport.h

```
 // To set ttyUSB0 comport mode
Int set_FR2ur2_ttyUSB0_mode(comport_mode);
 // To set ttyUSB1 comport mode
Int set_FR2ur2_ttyUSB1_mode(comport_mode);
```

Note: The comport_mode setting, refer to Table 1 of section 3-1-4

3-2-3. To control CAN bus Port:

FR_u2c2 generate two devices node, the path are "/dev/can0" and "/dev/can1".

Refer to the API files.


FR_u2c2.cpp


FR_u2c2.h

```
// First initial CAN bus
void init_FR_u2c2_can();

// To send data from CAN0 and CAN1
void FR_u2c2_can0_tx(int,int,char*);
void FR_u2c2_can1_tx(int,int,char*);

// Receive data from CAN0 and CAN1
QStringList get_FR_u2c2_can0_rx();
QStringList get_FR_u2c2_can1_rx();
```

3-2-4. To control Grove I/O Port:

FR_gro1 generate devices node, the path are "/dev/ttyS4".

Refer to the API files.


fr_gro1.h


fr_gro1.cpp

➢ The UART part of Grove I/O port

```
// To init UART_ttyS4
bool init_FR_gro1_uart();

// To send data from UART_ttyS4
void tx_FR_gro1_uart(QString data);

// To receive data from UART_ttyS4
QString rx_FR_gro1_uart();
```

➢ The I2C part of Grove I/O port

```
// To write data from I2C1 & I2C2 & I2C3 & I2C4
void set_FR_gro1_i2c1(int address, int value);
void set_FR_gro1_i2c2(int address, int value);
void set_FR_gro1_i2c3(int address, int value);
void set_FR_gro1_i2c4(int address, int value);

//To read data from I2C1 & I2C2 & I2C3 & I2C4
QString get_FR_gro1_i2c1(int address);
QString get_FR_gro1_i2c2(int address);
QString get_FR_gro1_i2c3(int address);
QString get_FR_gro1_i2c4(int address);

//To get value from ADC0 & ADC1
QString get_FR_gro1_adc0();
QString get_FR_gro1_adc1();
```

➢ The GPIO part of Grove I/O port

```
//Set each GPIO(0 - 7) as input port
void set_FR_gro1_gpio0_in();
void set_FR_gro1_gpio1_in();
void set_FR_gro1_gpio2_in();
void set_FR_gro1_gpio3_in();
void set_FR_gro1_gpio4_in();
void set_FR_gro1_gpio5_in();
void set_FR_gro1_gpio6_in();
void set_FR_gro1_gpio7_in();

//Set each GPIO(0 - 7) as output port
void set_FR_gro1_gpio0_out();
void set_FR_gro1_gpio1_out();
void set_FR_gro1_gpio2_out();
void set_FR_gro1_gpio3_out();
void set_FR_gro1_gpio4_out();
```

```
void set_FR_gro1_gpio5_out();
void set_FR_gro1_gpio6_out();
void set_FR_gro1_gpio7_out();

// Output to each GPIO(0 - 7)
void set_FR_gro1_gpio0(eGPIO_value);
void set_FR_gro1_gpio1(eGPIO_value);
void set_FR_gro1_gpio2(eGPIO_value);
void set_FR_gro1_gpio3(eGPIO_value);
void set_FR_gro1_gpio4(eGPIO_value);
void set_FR_gro1_gpio5(eGPIO_value);
void set_FR_gro1_gpio6(eGPIO_value);
void set_FR_gro1_gpio7(eGPIO_value);

//Get from each GPIO(0 - 7)
QString get_FR_gro1_gpio0();
QString get_FR_gro1_gpio1();
QString get_FR_gro1_gpio2();
QString get_FR_gro1_gpio3();
QString get_FR_gro1_gpio4();
QString get_FR_gro1_gpio5();
QString get_FR_gro1_gpio6();
QString get_FR_gro1_gpio7();
```