

# ARPI600

## Introduction

---

Raspberry Pi Expansion Board, Supports Arduino XBee

## How to start up serial debugging function

---

1. Please install a USB to UART driver (cp2102 driver) before applying Raspberry Pi board. After the installation, please open Device Manager to check whether the PC can identify the USB to UART driver.
2. After connecting ARPI600 to Raspberry Pi board, you should power up the Raspberry Pi board and then connect ARPI600 to the USB port on your PC. It is not recommended to connect ARPI600 to the USB port on PC before powering up Raspberry Pi board or for power supply, since the power supply capability of the USB port on PC is not powerful enough to support Raspberry Pi board and ARPI600 expansion board at a same time.
3. The Raspbian system is set to serial debugging output by default, so you should configure the jumpers on the ARPI600 in order to start up the serial debugging function for PC.
  - Connect CP\_RX to P\_TX
  - Connect CP\_TX to P\_RX

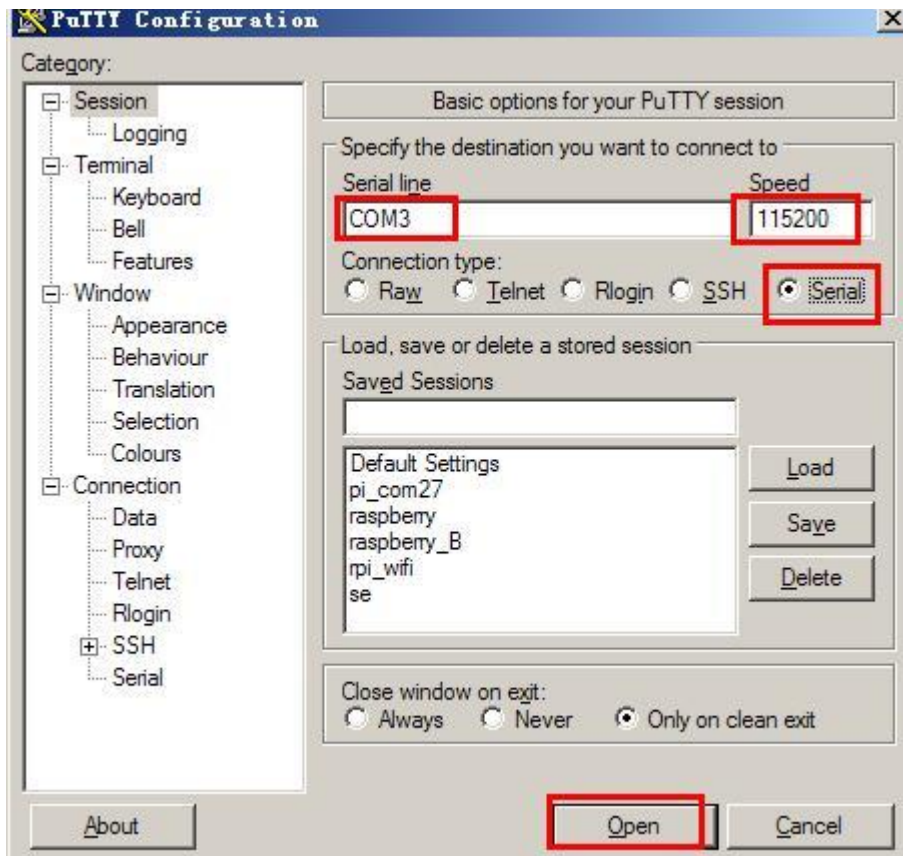
Starting up serial debugging function



4. Start software/putty.exe, and configure the following parameters marked with red boxes shown in the Figure
  - Serial line: it is used to select corresponding serial port. In this example, the serial port is COM3, please configure this option based on the actual situation (The serial port in used can be check by Device Manager).
  - Speed: it is used to set the Baud rate: 115200.
  - Connection type: this option should be set to Serial.

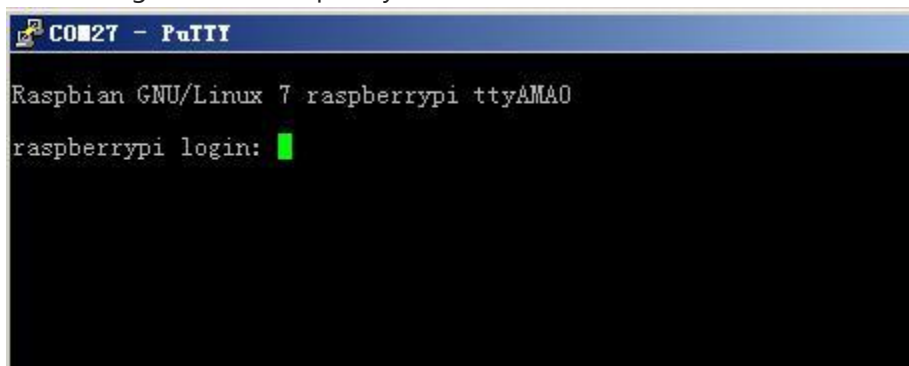
And then, click the button Open.

PuTTY settings



5. Then, you will see a window popped up as the following figure shows (If there is nothing shown in the window, please press the key Enter, then you can see the information displayed as the figure shows).

Connecting PuTTY to Raspberry Pi board



6. In this window, you should input following information:

User name: pi

Password: raspberry

Then, you can enter the serial terminal.

How to control peripherals by Raspberry Pi

## System serial port configuration

1. Enter the terminal of Raspberry Pi, and run:

```
sudo nano /boot/cmdline.txt
```

Then, you should modify the following lines:

```
wc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2  
rootfstype=ext4 elevator=deadline rootwait
```

into:

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4  
elevator=deadline rootwait
```

Press the keys Ctrl+X, and select the option Y to save the modification.

## 2. Run the code:

```
sudo nano /etc/inittab
```

And modify the following lines:

```
#Spawn a getty on Raspberry Pi serial line  
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

into:

```
#Spawn a getty on Raspberry Pi serial line  
#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

Press the keys Ctrl+X, and select the option Y to save the modification.

## 3. Run the code:

```
sudo reboot
```

After completing the modifications described above and restarting the Raspberry Pi board, the serial debugging function is started up. In this mode, you cannot enter the terminal of the Pi via the serial port any more, but can control the serial output by the software (If you want to reset the serial port to the serial debugging output, please restore the factory settings and restart the Raspberry Pi. Therefore, it is recommended to backup the Raspbian before making any modification).

## Install relative libraries

1. Copy the library file software/wiringPi.tar.gz into Raspberry Pi (you can perform it by a U disk), and enter the terminal of Raspberry Pi to input the following code:

```
sudo tar xvf wiringPi.tar.gz
cd wiringPi/
chmod 777 build
./build
```

2. After installing the library, input the following code:

```
gpio -v
```

3. Now, you can check whether the installation is successful.

## Serial data display

1. Set the jumpers on the ARPI600: connect P\_RX to P\_TX, and CP\_TX to CP\_RX. And connect the ARPI600 to the USB port on your PC.
2. Start the PuTTY serial debugging software to configure following parameters:
  - Serial line: it is used to select corresponding serial port.
  - Speed: it is used to set the Baud rate: 9600 (Notices: the parameter Speed in here is set to 9600, differently from the configuration shown in the figure).
  - Connection type: this option should be set to Serial.
3. Copy the file program/Xbee/send into the Raspbian, and enter the folder send, then, execute the following code:

```
sudo make
sudo ./serialTest
```

The terminal will display the data as



## Building a wireless network with two XBee modules

### Preparations

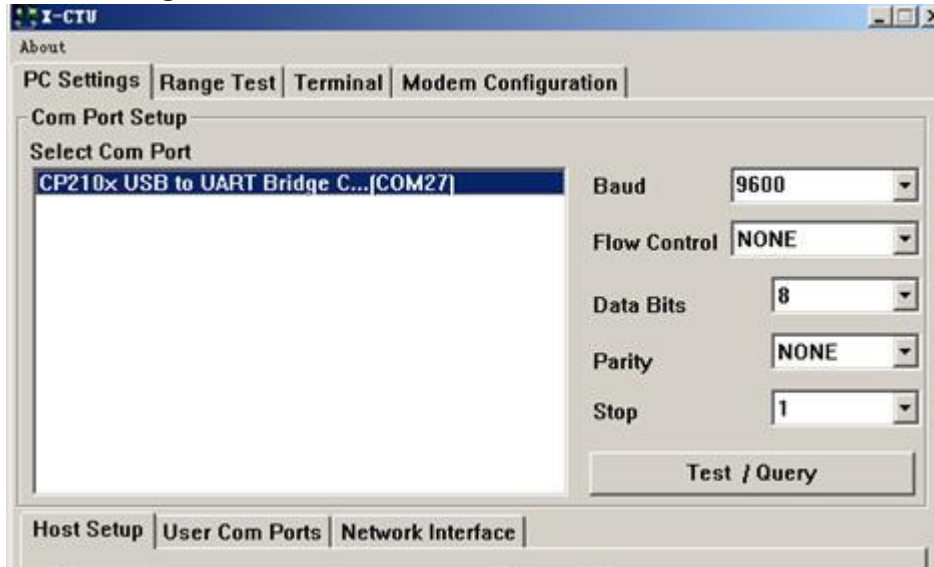
1. Two XBee modules
2. Two ARPI600 modules
3. Two Raspberry Pi boards

- In this document, we will divided the device above into two groups: Group A and Group B, of which Group A contains XBee-A, Raspberry Pi board-A and ARPI600-A, and Group B contains XBee-B Raspberry Pi board-B and ARPI600-B.

## Installing X-CTU tool

1. Double click the file software/X-CTU V5.2.8.6.exe on your PC to start installing X-CTU tool. After a successful installation, you can open the X-CTU tool as the figure shows.

X-CTU setting



2. Configure the XBee module. The default setting of XBee is as followed:
  - Baud: 9600
  - Data Bite: 8
  - Parity: NONE
  - Stop: 1

## Testing the connection between PC and XBee

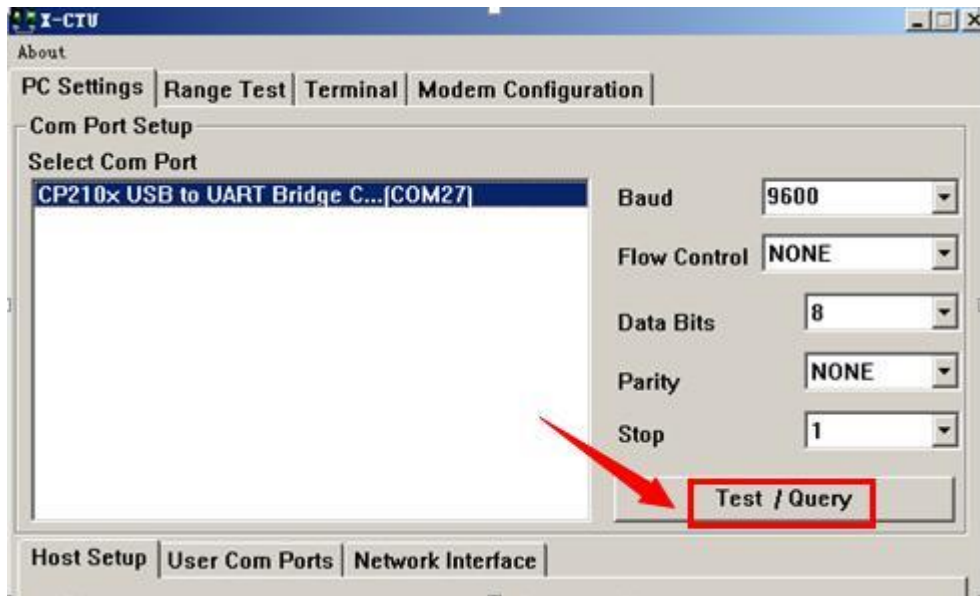
1. Connect XBee-A to ARPI600-A, and XBee-B to ARPI600-B, respectively.
2. Set the jumpers on ARPI600 to start up the serial debugging function for the XBee, as the figure shows.
  - Connect XB\_RX to CP\_RX
  - Connect CP\_TX to XB\_TX

Jumpers setting for serial debugging function between the Pi and the XBee



3. Power up Raspberry Pi board (For more detailed information.

- Click the button Test/Query to check whether the connection between the ARPI600 and the XBee is established successfully.



- For a successful connection, you can see the dialog box as the figure shows.

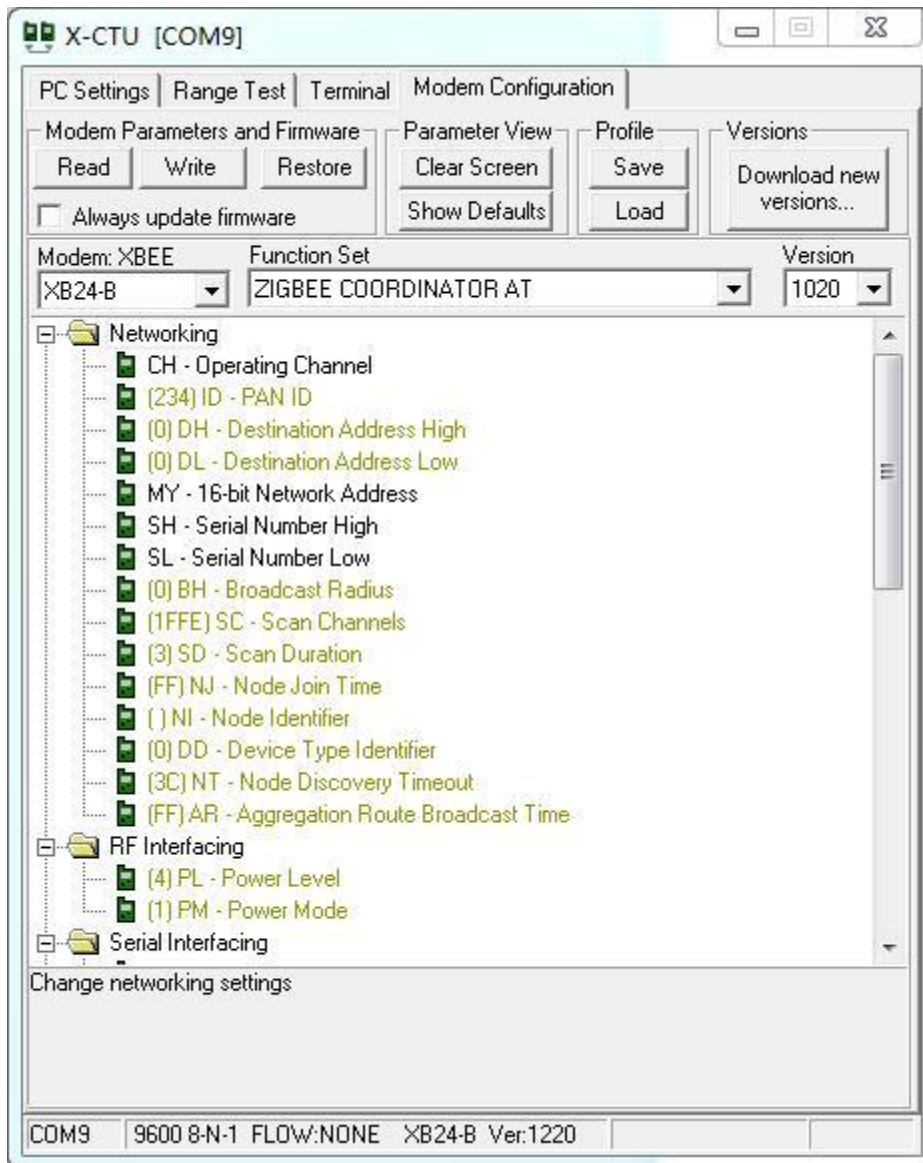
Successful connection



## Configuring XBee-A module

- Select the option Modem Configuration, and click the button Read to read out the current parameters of XBee.

Reading out current parameters



2. Select the option ZIGBEE ROUTER/END DEVICE AT under the pull-down menu Function Set:

Selecting the option ZIGBEE ROUTER/END DEVICE AT under Function Set



3. Set the read Networking parameters:
  - ID: 234
  - DH: 0
  - DL: 0
4. Click the button Write to download the configured parameters into the XBee-A module.

## Configuring XBee-B module

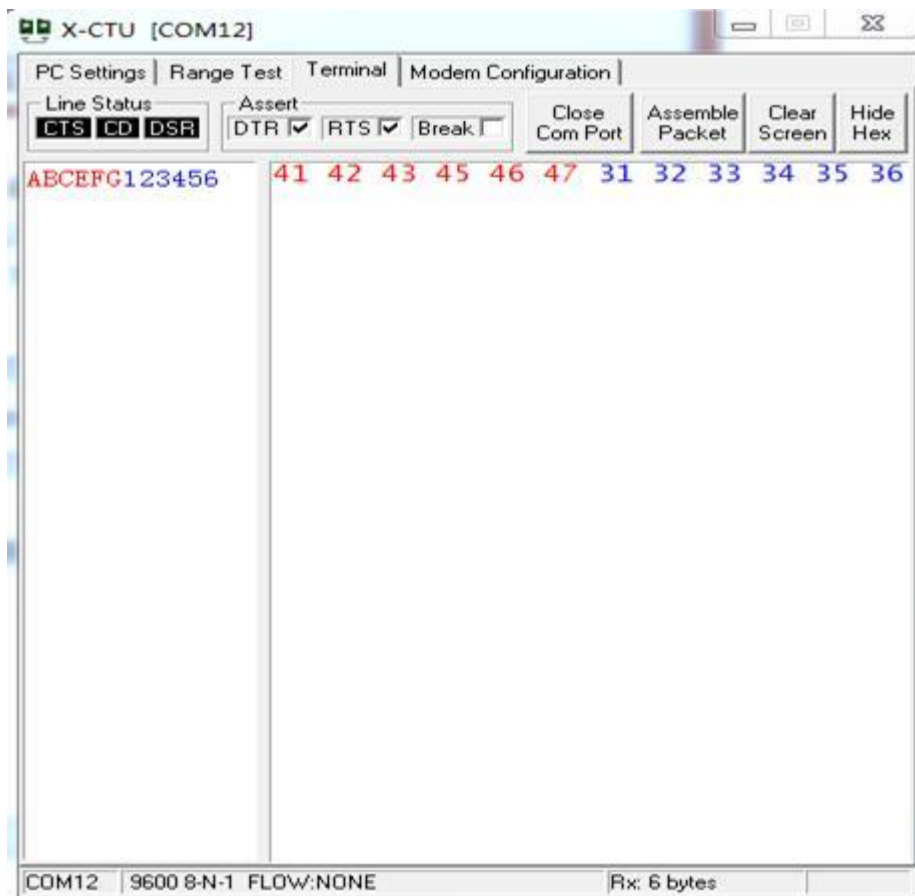
1. Configure XBee-B module according to the processes described in the Section 3.1 and the Section 0. However, there is something different. In configuring XBee-B, you should select the option ZIGBEE COORDINATOR AT under the pull-down menu Function Set:

Selecting the option ZIGBEE COORDINATOR AT under Function Set



2. Set the read Networking parameters:
  - ID: 234
  - DH: 0
  - DL: ffff
3. Click the button Write to download the configured parameters into the XBee-B module.
4. In order to implement a simple P2P network, please configure XBee-A and XBee-B according to the processes described above. Start two X-CTU tools, and select different COM interfaces in the option PC Settings to control Group A and Group B respectively.
5. Input the data to be transmitted in the X-CTU Terminal of XBee-A, then, you can find that the inputted data will be sent to XBee-B automatically, and displayed in the X-CTU Terminal of XBee-B. In the X-CTU, data in blue is the data to be sent, and data in red is the received data.

Data transmission and receive



6. This Figure shows the normal operating state of XBee module.



## How to perform wireless transmission with XBee

---

Before performing the wireless transmission with XBee, please make sure the wireless network built by two XBee modules is work properly.

### 1. Set the jumpers on ARPI600:

Here, we should use two Raspberry Pi boards: Raspberry Pi-A and Raspberry Pi-B, of which Raspberry Pi-A is used for transmitting data and Raspberry Pi-B is used for receiving data.

Connect Raspberry Pi-A to the serial port of XBee-A and Raspberry Pi-B toXBee-B, respectively. And then, set the jumpers on ARPI600, as the shows.

- Connect XB\_RX to P\_TX
- Connect XB\_TX to P\_RX

Setting jumpers on ARPI600



### 2. Testing the serial port:

Copy the file program/Xbee/getdata to the Raspberry Pi-B, and enter the folder getdata.

Then, execute the code:

```
sudo make
sudo ./serialTest
```

The relative data will be displayed as followed.

Displaying the message getdata

```
pi@raspberrypi /getdata $ sudo ./serialTest
getdata
```

### 3. Run the code for transmitting data on Raspberry Pi-A. Then, copy program/Xbee/send into the Raspbian, and enter the folder send to execute the following code:

```
sudo make
sudo ./serialTest
```

You will see the following message displayed in the PuTTY of Raspberry Pi-B.

## RTC clock

---

1. Set the jumpers on RTC JMP of the ARPI600.
2. Open the LXTerminal on the desktop of Raspbian, and input the code:

```
i2cdetect -y 1
```

- Then, you will see the device address of PCF8563 connected to Raspberry Pi. Here, the device address of PCF8563 is 51, which means the PCF8563 is identified by Raspberry Pi.

The device address of PCF8563 connected to Raspberry Pi



```
pi@raspberrypi ~$ i2cdetect -y 1
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- UU -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- 51 -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

- Enter LXTerminal, and run:

```
modprobe i2c-dev
echo pcf8563 0x51 > /sys/class/i2c-adapter/i2c-1/new_device
hwclock -r
```

(Read out the time of the connected RTC based on I2C)

LXTerminal will display the time clocked by PCF8563 which may be different from the Raspbian)

- Enter LXTerminal, and run:

```
hwclock -w (Write the time of the Raspbian into PCF8563)
hwclock -r
```

(Synchronize the time of Raspbian to PCF8563)

```
hwclock -s
```

(Synchronize the time of Raspbian with hardware RCT)

## AD conversion (ARPI600 on-board chip TLC1543)

### Configuring Pin A0 to Pin AD

- Please make sure you have installed relative libraries (refer to the Section 2.2: Installing relative libraries).
- Set the jumpers to select reference voltage:
  - Connect REF to 5V, which means the AD conversion reference voltage is 5V (connecting to 5V is a default setting).
  - Connect REF to 3V3, which means the AD conversion reference voltage is 3.3V.

**Notices: REF can only be connected to one reference voltage at a time.**

Setting AD reference voltage

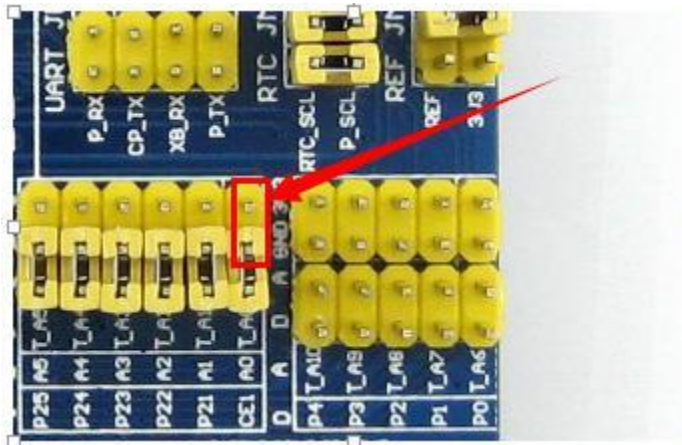


- Copy the file program/AD\_TLC1543 in to Raspbian. Then, enter the folder AD\_TLC1543, and execute the following code under the terminal:

```
sudo make
sudo ./t1c1543
```

- Terminal will display relative AD conversion value. By default, the displayed conversion value comes from Pin AD0 on TLC1543 (Pin T\_A0 on ARPI600).
- Connect the jumpers T\_A0 to A0, then the Pin A0 on Arduino interface can sever as an AD conversion pin, as shows.

connecting Pin T\_A0 to Pin A0



## Configuring as other AD pins

- If you want to display the conversion value from other AD pins on TLC1543, please enter the terminal and edit the file t1c1543.c:

```
sudo nano t1c1543.c
```

Find out the following line:

```
re=ADCSe1Channel(0);
```

Modify the "0" in the line into the number corresponding to other AD pin (For example, modify to "1" for testing the conversion value from Pin AD1 (Pin T\_A1), and modify to "2" for testing the conversion value from Pin AD2 (Pin T\_A2), and so on, until to "10" for testing the conversion value from Pin AD10 (T\_A10)). After completing the operation described above, press the keys Ctrl+X, and select the option Y to save the modification.

2. Execute the following code under the terminal:

```
sudo make  
sudo ./t1c1543
```

Now, the modification is in effect.

## Interface description

---

### Interface over view

The default relationship between Arduino digital control pins and Raspberry Pi IOs is shown in the table. The relationship between Arduino digital control pins and Raspberry Pi IOs

<b>APRI600</b>	<b>IO of Raspberry Pi B+</b>
D0	P_RX
D1	P_TX
D2	P0
D3	P1
D4	P2
D5	P3
D6	P4
D7	P5

D8	P6
D9	P7
D10	CE0
D11	MOSI
D12	MISO
D13	SCK

The jumper pins D11, D12 and D13 on the module are used for configuring the ARPI600. And these pins should be shorted by the 0Ω resistances.

Configuring the jumpers D11, D12 and D13



In factory settings, the jumpers are set as followed:

- Connect SCK to D13
- Connect MISO to D12
- Connect MOSI to D11

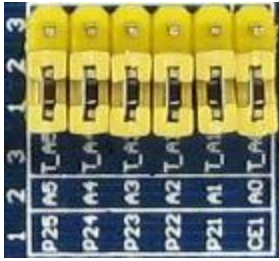
The following settings are connecting pins D11, D12 and D13 to the general IO control pins of Raspberry Pi board.

- Connect D13 to P26
- Connect D12 to IO\_SD
- Connect D11 to IO\_SC

**Notices: Users can modify the settings of these jumpers as required. In this operation, welding is required. Any changes under no guidance from Waveshare will be considered as a waiver of warranty.**

The pins A0-A5 of ARPI600 can also be configured as IO pins or ADC pins.

configuring Pins A0-A5



a) When the pins A0-A5 are connected to 1, they will sever as IO control pins. The relationship between the pins A0-A5 and the pins of Raspberry Pi board is as the table shows.

The relationship between the pins A0-A5 and the pins of Raspberry Pi board

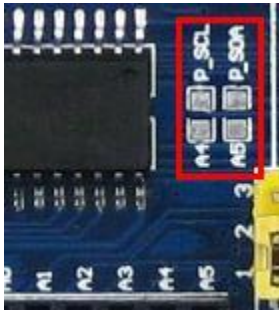
	APRI600 IO of Raspberry Pi B+
A0	CE1
A1	P21
A2	P22
A3	P23
A4	P24
A5	P25

b) When the pins A0-A5 are connected to 3, they will sever as ADC pins.

You can also connect the pin A4 to P\_SCL, and the pin A5 to P\_SDA (as the figure shows), to making them sever as I2C control pins of Raspberry Pi board. However, in default settings, the pins A4 and P\_SCL are disconnected, and so do the pins A5 and P\_SDA.

**Notices: Users can modify the settings of these jumpers as required. In this operation, welding is required. Any changes under no guidance from Waveshare will be considered as a waiver of warranty.**

Setting the pins A4 and A5 as I2C control pins



ARPI600 provides sensor interfaces 4P. The figure shows the jumper settings for sensor interface 4P.

Sensor interface 4P



In which:

- The pins on A are connected to the ADC pins A6-A10 of TLC1543;
- The pins on D are connected to the IO control pins P0-P4 of Raspberry Pi board.

Users can apply different sensors via the sensor interface.

### **How to use ARPI600 with external sensor kits (purchased separately)**

- In the following operations, the ARPI600 should be installed to the RPi board firstly, or else the ARPI600 cannot work with the sensor kits.

- [Sensors Pack](#)

#### *Color Sensor*

- Install the ARPI600 to the RPi board;
- Connect the sensor to the corresponding pins on the ARPI600, according to the following table:

Color Sensor Pins	ARPI600 Pins
LED	3.3V

OUT	P0
S3	P4
S2	P3
S1	P2
S0	P1
GND	GND
VCC	3.3V

- Copy the folder Color\_Sensor into the Raspberry Pi system, and then power up the device. Enter the terminal and run the following commands:

```
cd Color_Sensor
```

```
chmod +x Color_Sensor // Grant execution permission for the first run
```

```
sudo ./Color_Sensor
```

- The program performs the white balance adjustment to the sensor and it may take 2 seconds. When finished, you can see relative data of RGB are outputted on the terminal. A color check list is helpful for finding out what the measured color it is.
- Press the keys **Ctrl+C** to end the program.

#### *Flame Sensor*

- Install the ARPI600 to the RPi board;
- Connect the sensor to the corresponding pins on the ARPI600, according to the following table:

<b>Flame Sensor Pins</b>	<b>ARPI600 Pins</b>
DOUT	P0



AOUT	T_A6
GND	GND
VCC	3.3V

- Copy the folder Flame\_Sensor into the Raspberry Pi system, and then power up the device. Enter the terminal and run the following commands:

```
cd Flame_Sensor
```

```
chmod +x General_Sensor //Grant execution permission for the first run
```

```
sudo ./General_Sensor
```

- The signal indicator will turn on, when the sensor is close to a fire. And it will turn off, when the sensor is away from the fire.
- The terminal output changes along with the distance from the sensor to the fire.
- Press the keys **Ctrl+C** to end the program.

**Notice: The flame sensor is designed to detect fire only, but it is not fireproof itself. When using it, please keep a safe distance from the fire to avoid burning out.**

#### *Hall Sensor*

- Install the ARPI600 to the RPi board;
- Connect the sensor to the corresponding pins on the ARPI600, according to the following table:

Hall Sensor Pins	ARPI600 Pins
DOUT	P0
AOUT	T_A6
GND	GND
VCC	3.3V

- Copy the folder Hall Sensor into the Raspberry Pi system, and then power up the device. Enter the terminal and run the following commands:

```
cd Hall_Sensor
```

```
chmod +x General_Sensor //Grant execution permission for the first run
```

```
sudo ./General_Sensor
```

- The signal indicator will turn on, when the sensor is close to a magnet. And it will turn off, when the sensor is away from the magnet.
- The terminal output changes along with the distance from the sensor to the magnet.
- Press the keys **Ctrl+C** to end the program.

#### *Infrared Reflective Sensor*

- Install the ARPI600 to the RPi board;
- Connect the sensor to the corresponding pins on the ARPI600, according to the following table:

<b>Infrared Reflective Sensor Pins</b>	<b>ARPI600 Pins</b>
DOUT	P0
AOUT	T_A6
GND	GND
VCC	3.3V

- Copy the folder Infrared\_Reflective\_Sensor into the Raspberry Pi system, and then power up the device. Enter the terminal and run the following commands:

```
cd Infrared_Reflective_Sensor
```

```
chmod +x General_Sensor //Grant execution permission for the first run
```

```
sudo ./General_Sensor
```

- The signal indicator will turn on, when the sensor is close to a barrier. And it will turn off, when the sensor is away from the barrier.

- The terminal output changes along with the distance from the sensor to the barrier.
- Press the keys **Ctrl+C** to end the program.

#### *Laser Sensor*

- Install the ARPI600 to the RPi board;
- Connect the sensor to the corresponding pins on the ARPI600, according to the following table:

<b>Laser Sensor Pins</b>	<b>ARPI600 Pins</b>
DOUT	P0
GND	GND
VCC	3.3V

- Copy the folder Laser\_Sensor into the Raspberry Pi system, and then power up the device. Enter the terminal and run the following commands:

```
cd Laser_Sensor
```

```
chmod +x General_Sensor //Grant execution permission for the first run
```

```
sudo ./General_Sensor
```

- The signal indicator will turn on, when a barrier is placed above the sensor. And the signal indicator will turn off, when the barrier is away from the sensor. By reading the status of the indicator, you can get to know whether the sensor has detected the barrier.
- Press the keys **Ctrl+C** to end the program.

#### *Moisture Sensor*

- Install the ARPI600 to the RPi board;
- Connect the sensor to the corresponding pins on the ARPI600, according to the following table:

<b>Moisture Sensor Pins</b>	<b>ARPI600 Pins</b>
DOUT	P0

AOUT	T_A6
GND	GND
VCC	3.3V

- Copy the folder Moisture\_Sensor into the Raspberry Pi system, and then power up the device. Enter the terminal and run the following commands:

```
cd Moisture_Sensor
```

```
chmod +x General_Sensor //Grant execution permission for the first run
```

```
sudo ./General_Sensor
```

- Insert the sensor into the soil and water the soil little by little. And then you may find the terminal output changes.
- Press the keys **Ctrl+C** to end the program.

#### *Rotation Sensor*

- Install the ARPI600 to the RPi board;
- Connect the sensor to the corresponding pins on the ARPI600, according to the following table:

<b>Rotation Sensor Pins</b>	<b>ARPI600 Pins</b>
SIA	P0
SIB	P1
SW	P2
GND	GND
VCC	3.3V

- Copy the folder Rotation\_Sensor into the Raspberry Pi system, and then power up the device. Enter the terminal and run the following commands:

```
cd Rotation_Sensor
```

```
chmod +x Rotation_Sensor //Grant execution permission for the first run
```

```
sudo ./Rotation_Sensor
```

- There are three actions of the rotary encoder: clockwise rotation, anticlockwise rotation, and reset button press. The terminal output of the module varies with different actions.

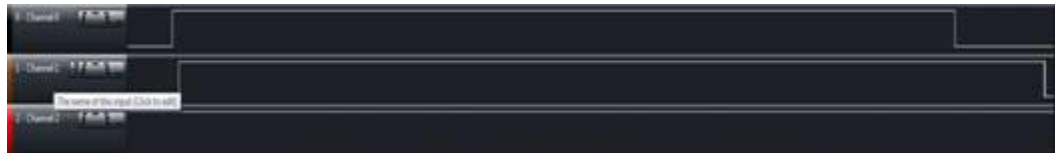
Turn right!

Turn left!

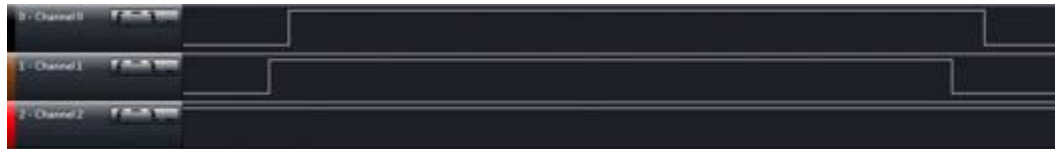
Turn down!

- Connect S1A, S1B, SW of the module to CH0, CH1, CH2 of a logic analyzer (purchased separately), respectively.

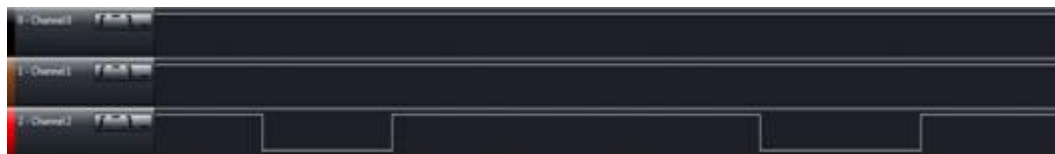
For clockwise rotation, the waveform outputs are shown as below:



- For anticlockwise rotation, the waveform outputs are shown as below:



- When the reset button is pressed, the waveform outputs are shown as below:



- Press the keys **Ctrl+C** to end the program.

### *Sound Sensor*

- Install the ARPI600 to the RPi board;
- Connect the sensor to the corresponding pins on the ARPI600, according to the following table:

Sound Sensor Pins	ARPI600 Pins
DOUT	P0

AOUT	T_A6
GND	GND
VCC	3.3V

- Copy the folder Sound\_Sensor into the Raspberry Pi system, and then power up the device. Enter the terminal and run the following commands:

```
cd Sound_Sensor
```

```
chmod +x General_Sensor //Grant execution permission for the first run
```

```
sudo ./General_Sensor
```

- The signal indicator will turn on, when the microphone of the module is close to a sound source. And it will turn off, when the microphone is away from the magnet.
- The terminal output changes along with the distance from the sensor to the sound source.
- Press the keys **Ctrl+C** to end the program.

#### *Temperature-Humidity Sensor*

- Install the ARPI600 to the RPi board;
- Connect the sensor to the corresponding pins on the ARPI600, according to the following table:

<b>Temperature-Humidity Sensor Pins</b>	<b>ARPI600 Pins</b>
DOUT	P0
GND	GND
VCC	3.3V

- Copy the folder Temperature-Humidity\_Sensor into the Raspberry Pi system, and then power up the device. Enter the terminal and run the following commands:

```
cd Temperature-Humidity_Sensor
```

```
chmod +x DHT11 //Grant execution permission for the first run
```

```
sudo ./DHT11
```

- The data of temperature and humidity detected are shown on the terminal.  
For example:

```
Humidity=33
```

```
Temperature=28
```

- Press the keys **Ctrl+C** to end the program.

#### *MQ-5 Gas Sensor*

- Install the ARPI600 to the RPi board;
- Connect the sensor to the corresponding pins on the ARPI600, according to the following table:

<b>MQ-5 Gas Sensor Pins</b>	<b>ARPI600 Pins</b>
DOUT	P0
AOUT	T_A6
GND	GND
VCC	3.3V

- Copy the folder MQ-5\_Gas\_Sensor into the Raspberry Pi system, and then power up the device. Enter the terminal and run the following commands:

```
cd MQ-5_Gas_Sensor
```

```
chmod +x General_Sensor //Grant execution permission for the first run
```

```
sudo ./General_Sensor
```

- Warn-up the sensor for a minute.
- Put the sensor into a container filled with sensitive gas, you will find the indicator turns on. While take the sensor out of the container, you can see the indicator turns off. By reading the indicator, you can get to know whether the concentration of the sensitive gas exceeds standard.
- Press the keys **Ctrl+C** to end the program.

### *Tilt Sensor*

- Install the ARPI600 to the RPi board;
- Connect the sensor to the corresponding pins on the ARPI600, according to the following table:

<b>Tilt Sensor Pins</b>	<b>ARPI600 Pins</b>
DOUT	T_A6
GND	GND
VCC	3.3V

- Copy the folder Tilt\_Sensor into the Raspberry Pi system, and then power up the device. Enter the terminal and run the following commands:

```
cd Tilt_Sensor
```

```
chmod +x General_Sensor //Grant execution permission for the first run
```

```
sudo ./General_Sensor
```

- The signal indicator will turn on, when the sensor is being vibrated or in a tilted state. And it will turn off, when the sensor is laid flat. By reading the indicator, you can get to know whether the module is in a condition of vibrating or tilting.
- Press the keys **Ctrl+C** to end the program.

### *UV Sensor*

- Install the ARPI600 to the RPi board;
- Connect the sensor to the corresponding pins on the ARPI600, according to the following table: :

<b>UV Sensor Pins</b>	<b>ARPI600 Pins</b>
AOUT	T_A6
GND	GND



VCC	3.3V
-----	------

- Copy the folder UV\_Sensor into the Raspberry Pi system, and then power up the device. Enter the terminal and run the following commands:

```
cd UV_Sensor
```

```
chmod +x General_Sensor //Grant execution permission for the first run
```

```
sudo ./General_Sensor
```

- Place the sensor close to the sun light. The terminal output changes along with the distance from the sensor to the sun light source.
- Press the keys **Ctrl+C** to end the program.

### *Liquid Level Sensor*

- Install the ARPI600 to the RPi board;
- Connect the sensor to the corresponding pins on the ARPI600, according to the following table:

<b>Liquid Level Sensor Pins</b>	<b>ARPI600 Pins</b>
AOUT	T_A6
GND	GND
VCC	3.3V

- Copy the folder Liquid\_Level\_Sensor into the Raspberry Pi system, and then power up the device. Enter the terminal and run the following commands:

```
cd Liquid_Level_Sensor
```

```
chmod +x General_Sensor //Grant execution permission for the first run
```

```
sudo ./General_Sensor
```

- Immerse the sensor into the water deeply. The terminal output changes along with the water depth.
- Press the keys **Ctrl+C** to end the program.

### Resources

---

- [Schematic](#)

- [Demo Code](#)

## **Application Notes**

- [CP2102](#)
- [PCF8563](#)
- [Writing the Image to the SD card](#)
- [Libraries Installation for RPi](#)

## **Datasheets**

- [TLC1543 Datasheet](#)
- [CP2102](#)
- [PCF8563](#)
- [Panasonic SDFormatter](#)
- [Win32DiskImager](#)
- [putty](#)
- [wiringPi & bcm2835 C LIB](#)