1. Introduction to the module and the use of port binding

1.1. Introduction to Voice Control Module

1.1.1.CSK4002 chip

The voice control module on ROSMASTER is developed based on the **CSK4002** chip. **CSK4002** is a high-performance, strong arithmetic, low power consumption, resource-rich AISoC developed and designed for the AloT field, which can be widely used in the smart home, smart home appliances, emerging consumer electronics industry.

- CSK4002 adopts Andes D1088 core, and its AI/DSP acceleration module MVA supports a variety of Neural Network operators and vector operations, and is deeply adapted to KDDI AI algorithms, with an arithmetic power of up to 128GOPS.
- Comes with 8M Flash, 8M PSRAM, 1M SRAM.
- Support 8 channels of PDM audio input and 16 channels of I2S Audio Input data processing.
- Integrated rich mainstream peripheral interfaces: GPIO/UART/I2C/SPI/QSPI/SDIO/USB1.1/SDIO, etc..
- Equipped with low-latency embedded operating system Free RTOS, complete BSP driver, and perfect development tool resources.

1.1.2. Module Features

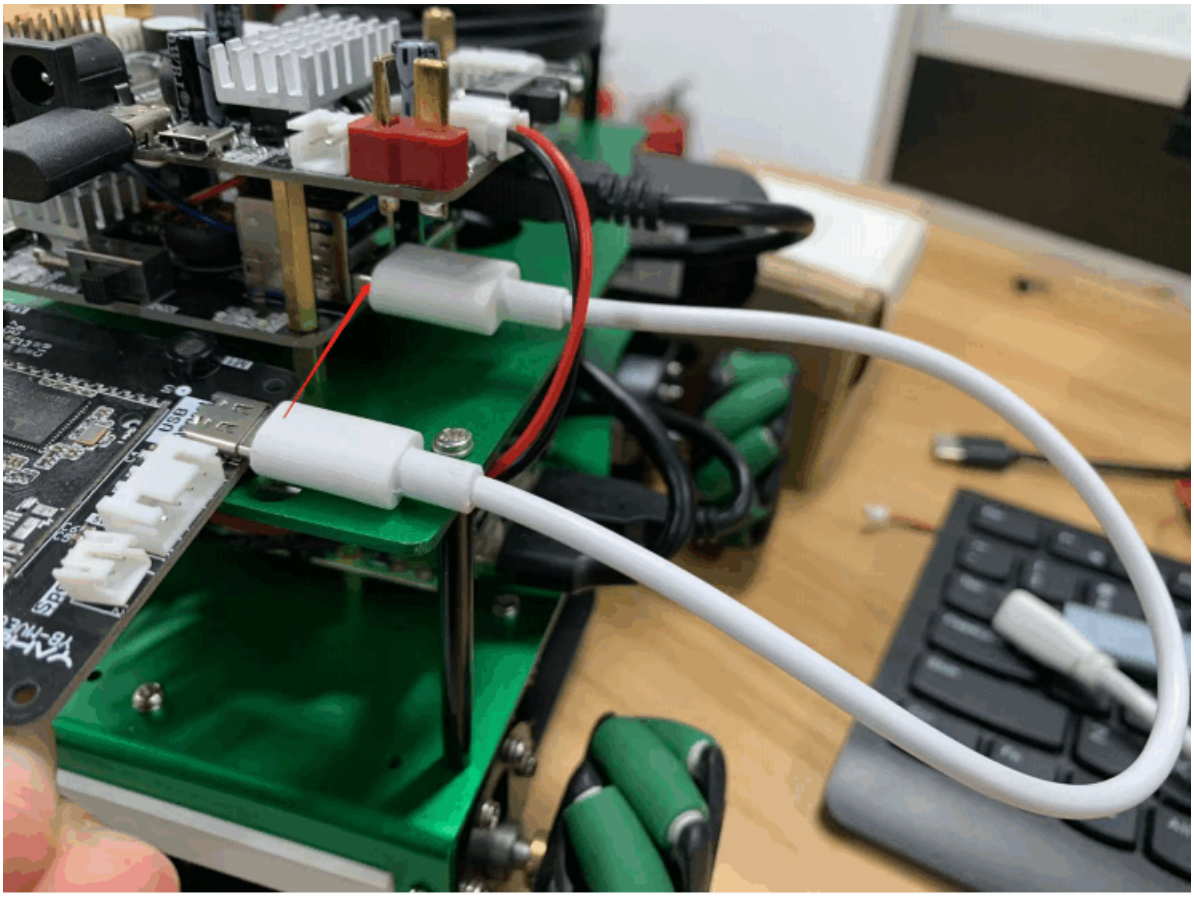
- Far-field sound pickup: The front-end adopts KDDI's dual microphone array algorithm, which can realize 360-degree far-field sound pickup for 5m users. Equipped with human voice automatic gain, according to the user's volume adaptive adjustment, to ensure that the overall audio after noise reduction is consistent.
- Echo cancellation: support in the process of user interaction, the device is broadcasting content or music, the user can wake up to interrupt the broadcast process for the next round of interaction, so that the interaction experience is more natural.
- Voice Announcement: Voice Announcement means that the user wakes up the device and utters a command word, and the device responds with a corresponding reply; or an active prompt. The purpose of voice announcement is to provide feedback to the user when the user sends out voice commands or in appropriate scenarios.
- Offline Command: When the user wakes up the device, he/she utters a command word (instruction) within the specified range, and the voice module receives the information and carries out the relevant processing according to the content of the command word, or transmits the content of the information to the host computer for relevant processing.
- Peripheral communication: The module receives the input from the microphone and processes it, and then communicates with other devices through USB, I2S, SDIO and other interfaces, and there are also general programmable IOs to communicate with other devices.
- Ambient noise reduction: Widely used in home, car, office and other scenes of environmental noise reduction, noise reduction while maximizing the retention of human voice information.

1.2. Utilization Modules

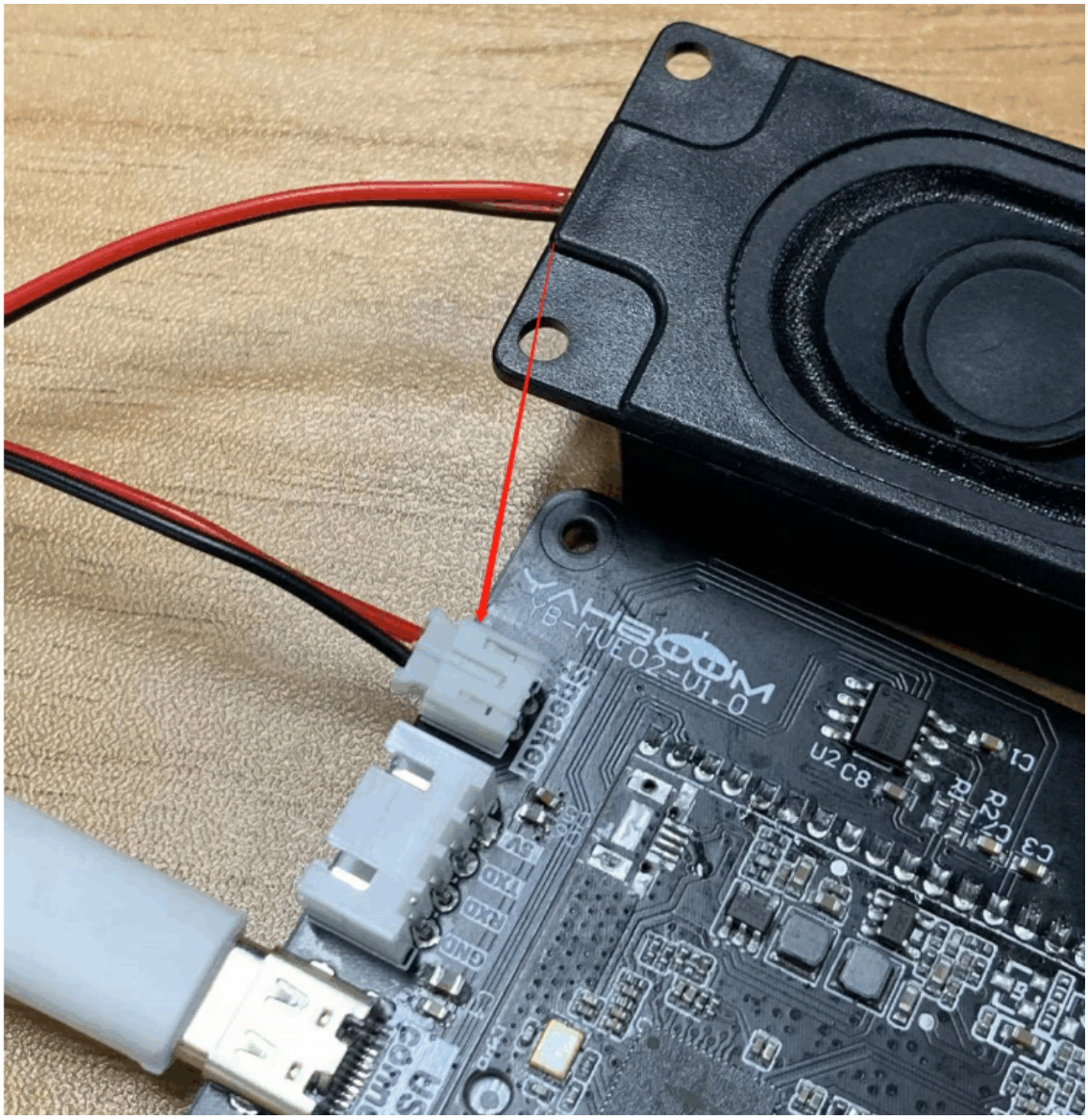
1.2.1. Wiring

The module is connected to the ROSMASTER master (or HUB board) via a universal Type-c cable, and the speakers are connected to the module via a PH2.0 cable.

- As shown in the figure below, one end of the Micro-USB cable is connected to the USB-connet port of the module, and the other end is connected to the port of the ROSMASTER main board (**this plugged in, wait until the back of the port binding can not be modified** , refer to the next section)



- The PH2.0 cable port is connected to the Speaker port on the module as shown below.



1.2.2. Wake-up word

The wake-up word is "Hello, Xiaoya", when waking up, you need to speak slower, if it is too fast, the module will not recognize it. After waking up the module, it can recognize other command words. Within 20 seconds of waking up, there is no need to re-wake up, just say the command word directly.

1.2.3. Command words

1). Voice control of trolley movement

Speech Recognition Content	Voice module sends to host	Host to voice module	Contents of voice announcement
stop	$B002#	$A002#	Okay, it's stopped.
go forward	$B004#	$A004#	Okay, it's moving forward.
draw back	$B005#	$A005#	Okay, it's backing up.
turn left	$B006#	$A006#	Okay, turning left .
turn right	$B007#	$A007#	Okay, it's turning right.
rotate left	$B008#	$A008#	Okay, it's rotating left.
rotate right	$B009#	$A009#	Okay, it's rotating right.

2) Voice control RGB light strip effect

Speech recognition content	Voice module sends to host	Host to voice module	Contents of the voice announcement
turn off the light	$B010#	$A010#	Okay, light's off.
turn on the red light	$B011#	$A011#	Okay, red light is on.
turn on the green light	$B012#	$A012#	Okay, green light is on.
turn on the blue light	$B013#	$A013#	Okay, blue light is on.
turn on the yellow light	$B014#	$A014#	Okay, yellow light is on.
Turn on the flowing light.	$B015#	$A015#	Okay, the flowing light is on.
Turn on the fade light	$B016#	$A016#	Okay, the fade light is on.
Turn on the breathing light.	$B017#	$A017#	Okay. Breathing light is on.
Display power level	$B018#	$A018#	Okay, it's showing the power level.

3) Voice control color recognition

Speech recognition content	Voice module sent to host	Host to voice module	Contents of voice announcement
What colour is this	$B060#	$A061#	It's red.
What colour is this	$B060#	$A062#	It's blue.
What colour is this	$B060#	$A063#	It's green.
What colour is this	$B060#	$A064#	It's yellow.

4) Voice control color tracking

Speech recognition content	Voice module sent to host	Host to voice module	Content of the voice announcement
Start tracking yellow	$B072#	$A072#	Okay, start tracing the yellow.
Start tracking red	$B073#	$A073#	Okay, start tracking red.
Start tracking the green	$B074#	$A074#	Okay, start tracking green.
start tracking blue	$B075#	$A075#	Okay, start tracking blue.
Cancel tracking	$B076#	$A076#	Okay. Cancel tracking.

5) Voice-controlled autopilot (patrol line)

Speech recognition content	Voice module sent to host	Host to voice module	Content of the voice announcement
Turn off line patrols	$B022#	$A022#	Okay, line patrol has been turned off.
red line patrol	$B023#	$A023#	Okay, red line patrol has been turned on
green line patrol	$B024#	$A024#	Okay, green line patrol has been turned on.
blue line patrol	$B025#	$A025#	Okay, blue line patrol is enabled.
yellow line patrol	$B026#	$A026#	Okay, the yellow line patrol has been activated.

6) Voice-controlled multi-point navigation

Speech recognition content	Voice module sent to host	Host to voice module	Content of the voice announcement
Navigate to position one.	$B019#	$A019#	Okay, heading to position one.
Navigate to site two.	$B020#	$A020#	Okay, heading to two.
Navigate to site three.	$B021#	$A021#	Okay, heading to three.
Navigate to position four.	$B022#	$A022#	Okay. Going to four.
Go back to the original point.	$B023#	$A023#	Okay, it's going back to the original point.

1.3. Voice Control Module Port Binding

Preface: Because the ID device numbers of the ROS expansion board and the voice control module are the same, it is not possible to bind the ID device numbers to them in the same way as in the previous tutorials.

The ROS expansion board and voice control module have the same ID device number. **Not binding the port may lead to port conflict or incorrect device identification, and the bound port can not be arbitrarily changed change position**, otherwise, the port will not be able to be used.

The bound ports cannot be changed at will*, or the binding will be invalid. This section of the course is completed in the host, do not need to bind in the docke, this section to the Raspberry Pi as the host to demonstrate, before binding, you need to comment out the original host rules file, involving myserial part, otherwise there will be a voice board and the ROS expansion board is recognized as the same device, the terminal input, the

```
sudo gedit /etc/udev/rules.d/usb.rules
```

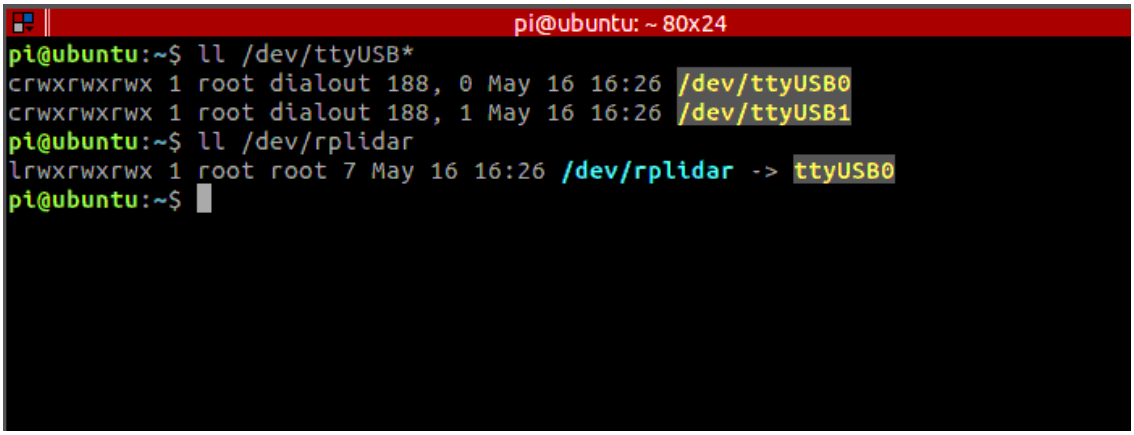
As shown above, comment out the myserial-related bindings, save and exit, and reload the rule by entering the following command in the terminal.

```
sudo udevadm trigger
sudo service udev reload
sudo service udev restart
```

1.3.1 Binding ROS expansion board device number

```
ll /dev/ttyUSB*
ll /dev/rplidar
```

- Without accessing the voice control board first, you can get ttyUSB0 and ttyUSB1

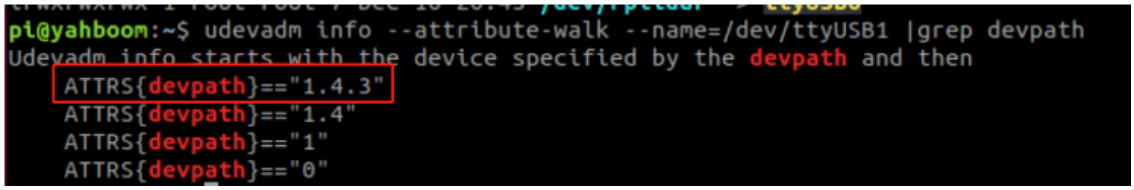


  Here ttyUSB0 is the device number recognized by the radar, then ttyUSB1 is the device number of the ROS expansion board.

- Then, we start by checking the port information of the ROS expansion board, mainly the device path information, terminal input, the

  ```
  udevadm info --attribute-walk --name=/dev/ttyUSB1 |grep devpath
  ```

  to get the following information (the actual situation prevails here), and the red box illustrates the path information of the device.

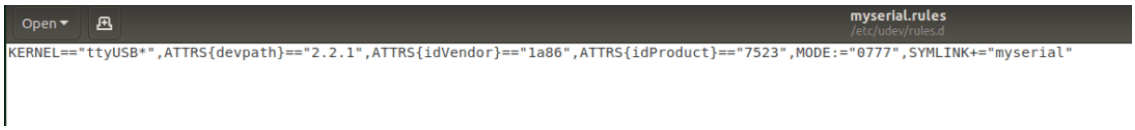


- Then, we modify the /etc/udev/rules.d/rplidar.rules file to bind the port number of the ROS expansion board first, and the terminal enters.

  ```
  sudo gedit /etc/udev/rules.d/myserial.rules
  ```

Add content, as shown below.

```
KERNEL=="ttyUSB*",ATTRS{devpath}=="1.4.3",ATTRS{idVendor}=="1a86",ATTRS{idPr
oduct}=="7523",MODE:="0777",SYMLINK+="myserial"
```



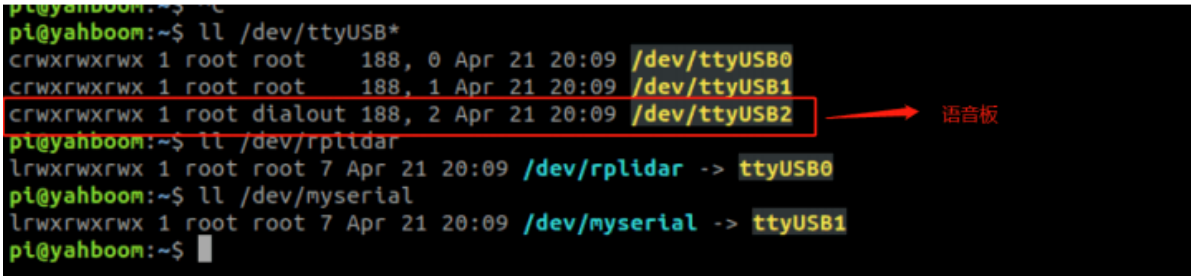
- After saving and exiting, the terminal reloads the device by entering the following three commands, the

```
sudo udevadm trigger
sudo service udev reload
sudo service udev restart
```

1.3.2. Binding the Voice Board Port Number

- Access the voice board and enter the following command at the terminal to view the device number.

```
ll /dev/ttyUSB*
```

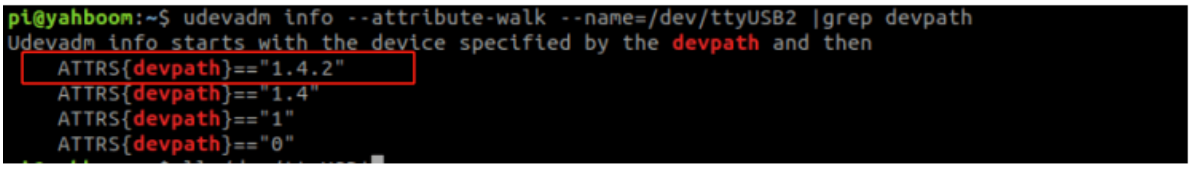


Here we find that the system recognizes the voice board as /dev/ttyUSB2 by the system, so let's enter the following command to view the device path

information.

```
udevadm info --attribute-walk --name=/dev/ttyUSB2 |grep devpath
```

Get the following image.



- Then, we modify the /etc/udev/rules.d/myspeech.rules file to bind the port number of the voice board, and the terminal enters.

```
sudo gedit /etc/udev/rules.d/myspeech.rules
```

Add content as shown in the image below.

```
KERNEL=="ttyUSB*",ATTRS{devpath}=="1.4.2",ATTRS{idVendor}=="1a86",ATTRS{idPr
oduct}=="7523",MODE:="0777",SYMLINK+="myspeech"
```

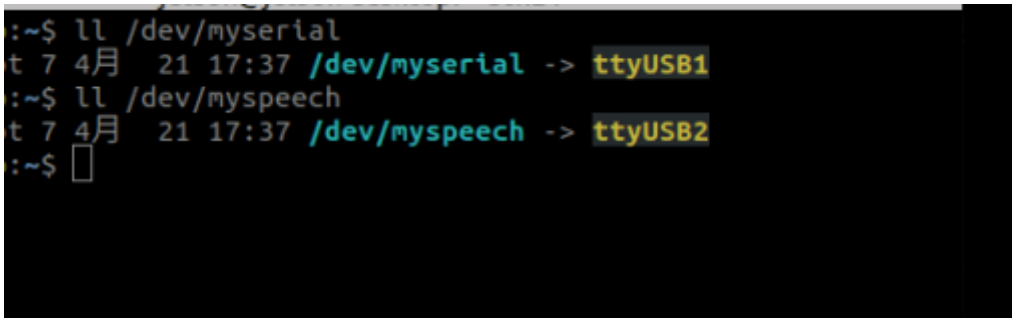
- After saving and exiting, the terminal reloads the system device by entering the following statement

```
sudo udevadm trigger
sudo service udev reload
sudo service udev restart
```

1.3.3. Testing

Note their wiring position. This is fixed after binding, subsequent not to change the position at will or the system will not recognize the device , terminal input.

```
ll /dev/myserial
ll /dev/myspeech
```



As shown in the picture above, if you can recognize /dev/myserial and /dev/myspeech and the port number of the ttyUSB* after it is not the same, it means that the binding is successful.

Note: After binding, the ROS expansion board and voice board cannot be plugged into other ports, otherwise the device number will not be recognized, and if both are connected to the HUB expansion board, the HUB board cannot be plugged into other motherboard ports.