Voice Recognition Module Introduction

1. Module Introduction

CI1302 is a new generation high-performance neural network intelligent voice chip developed by Chipintelli. It integrates Chipintelli's self-developed Brain Neural Network Processor BNPU V3 and CPU core, with system main frequency up to 220MHz, built-in up to 640KByte SRAM, integrated PMU power management unit and RC oscillator, integrated dual-channel high-performance low-power Audio Codec and multiple peripheral control interfaces such as UART, IIC, IIS, PWM, GPIO, PDM, etc. The chip only requires a few external components such as resistors and capacitors to implement various intelligent voice product hardware solutions with extremely high cost performance.

The CI1302 chip has a Brain Neural Network Processor core (BNPU), supports offline NN acceleration operations and voice signal processing hardware acceleration, CPU main frequency up to 220MHz, capable of offline far-field voice recognition, built-in 2MB FLASH storage, can support 300 command words.

2. Working Principle

This module adopts command mode wake-up. Users need to say the set wake-up word to activate the voice interaction module first. After activation, voice recognition can be performed. The default wake-up keyword in the factory firmware is "Hello, yahboom". If no voice is recognized after 20 seconds, the module will enter sleep mode and needs to be re-awakened for next use. The module supports wake-up word modification, command word modification and custom vocabulary.

3. Precautions

- Use 5V voltage power supply, exceeding 5V will damage the module
- The usage environment should be quiet, noisy environments will affect recognition performance
- When speaking vocabulary, the voice should be loud and the speech rate should not be too fast. It is recommended to stay within 5 meters of the module.

4. Install Driver

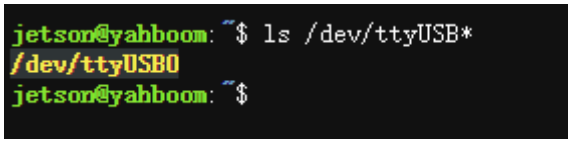
Before leaving the factory, the driver has been installed on the main board and can be used directly without installing the driver library. This explains the steps needed to install the driver if installing on a new main board. In the provided program source code folder, find the CH341SER folder, then copy it to the main board directory, open the terminal and enter the following commands to install,

```
cd CH341SER
make -j6
sudo make install
sudo modprobe ch34x
```

After successful installation, connect the voice module and the main board through the data cable, then enter the following command in the terminal,

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

If the following content is displayed, it means the installation is complete,

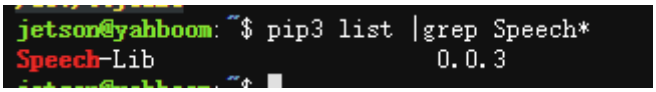


5. Install Driver Library

Before leaving the factory, the Chinese driver library has been installed on the main board and can be used directly without installing the driver library. For installing the English driver library, please see the next lesson "Flash the firmware (Must Read)". This explains the steps needed to install the driver library if installing on a new main board. In the provided program source code folder, find the py_install_v0.0.3 folder, then enter the following commands to install the driver library,

```
cd py_install_v0.0.3
sudo python3 setup.py install
```

After successful installation, you can use `pip3 list |grep Speech*` and if the following content appears, it means the installation was successful,



In the driver library, the default port for recognizing the voice module is /dev/ttyUSB0. If the port is incorrect, you need to modify the content in the library before installation. The modification location is,

```
/py_install_v0.0.3/Speech_Lib/Speech_Lib.py
```

```
import time
import threading
import sys
import serial
import os


# V0.0.3
class Speech(object):
    def __init__(self,path = '~/speech_music/', com="/dev/ttyUSB0"):
        # com="/dev/ttyUSB0"
        self.ser = serial.Serial(com, 115200)
        #self.ser = serial.Serial("/dev/ttyUSB0", 115200)
        self.language = 'cn' #默认是中文
        self.path = path

        if self.ser.isOpen():
            print("Speech Serial Opened! Baudrate=115200")
        else:
            print("Speech Serial Open Failed!")

    def __del__(self):
        self.ser.close()
        print("speech serial Close!")
```

The value of com should be modified to the actual port value recognized by the voice module, generally starting with /dev/ttyUSB.