Voice Slave Mode User Guide

Revision: V1.00 Date: August 14, 2017

www.holtek.com

Table of Contents

1 Quick Start

Voice Slave Mode Introduction

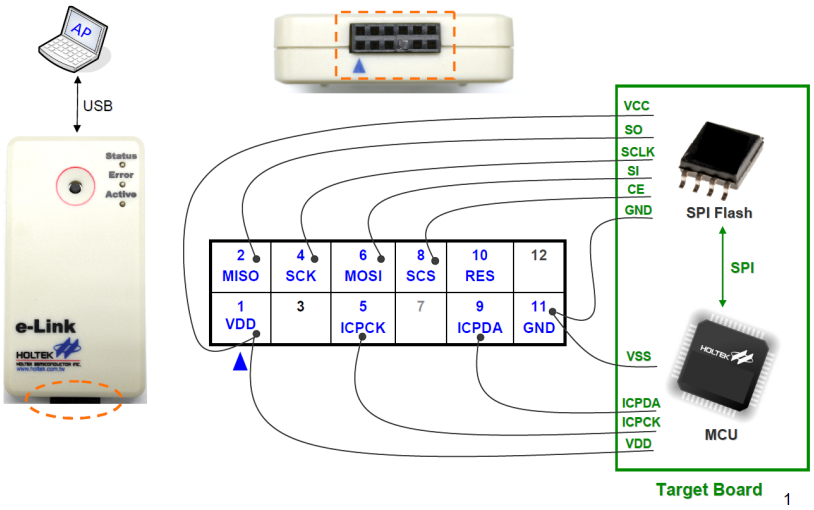
When using the voice slave mode of Holtek Voice MCU Workshop together with the HT66FV130/ HT66FV140 hardware circuits, the following functions can be implemented:

- The user selected voice files can be programmed into the SPI Flash and combine multiple voice files to generate an appropriate voice.
- A complete slave IDE3000 assembler project, which is used to play the voice, will be generated. Users can compile and download the voice playing project to the MCU directly using the IDE3000.
- The master can send control commands to the voice slave according to the HOLTEK I²C/single-wire protocol definition. Refer to "Communication Protocol Description" for details. It supports various features, such as up to 240 voice combinations, 13 volume levels as well as sleep and wakeup commands.

Hardware and Software Development Environment

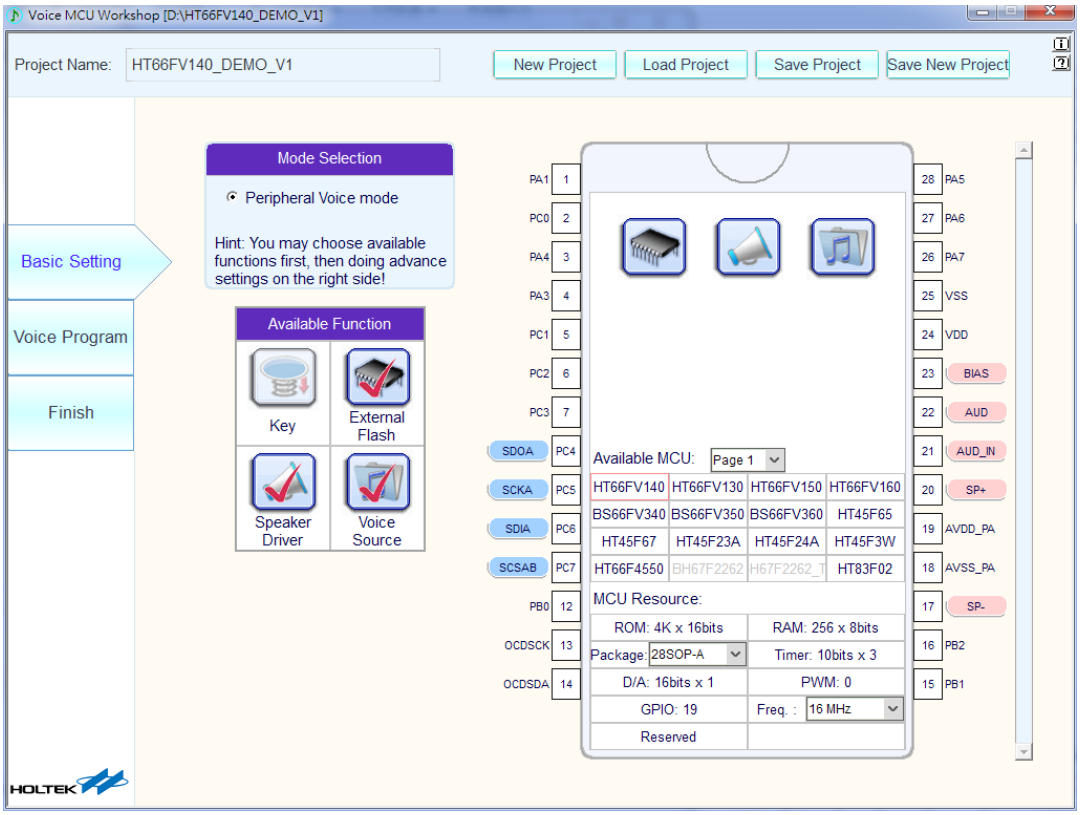
Hardware

- PC with Windows XP or later version
- HOLTEK simulation tool – HT e-Link
- HT66FV130/HT66FV140 hardware circuit, refer to "Application Circuits" for details
- The connection diagram for programming the voice in-circuit into the SPI FLASH is as follows



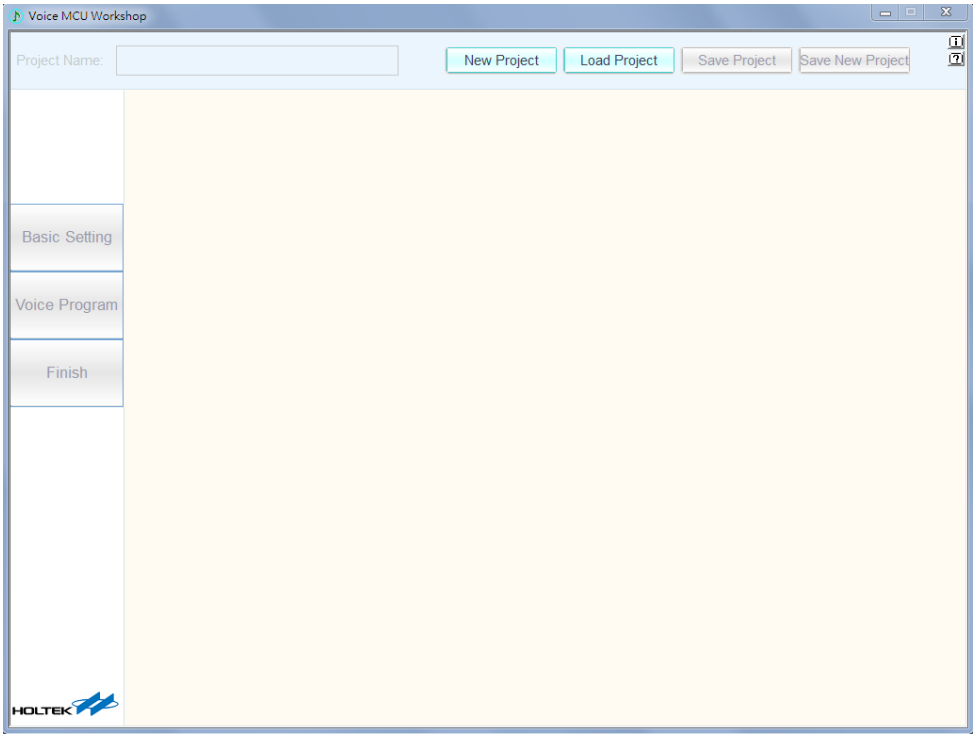
Software

Holtek Voice MCU Workshop V2.30 version or later versions (Click to download).

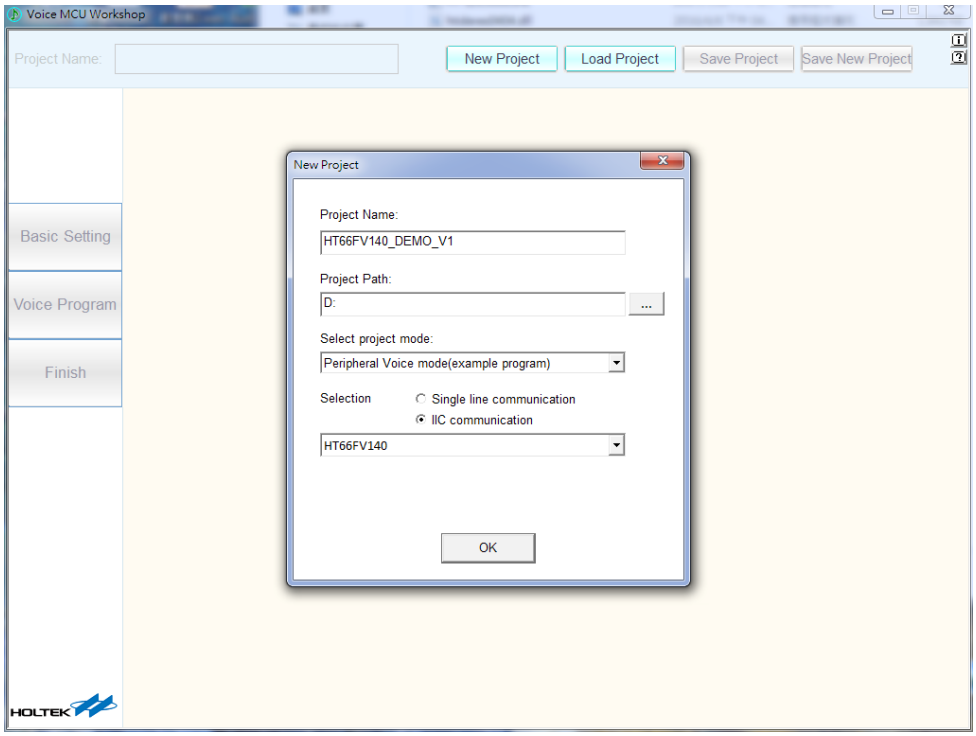


Detailed Operating Steps

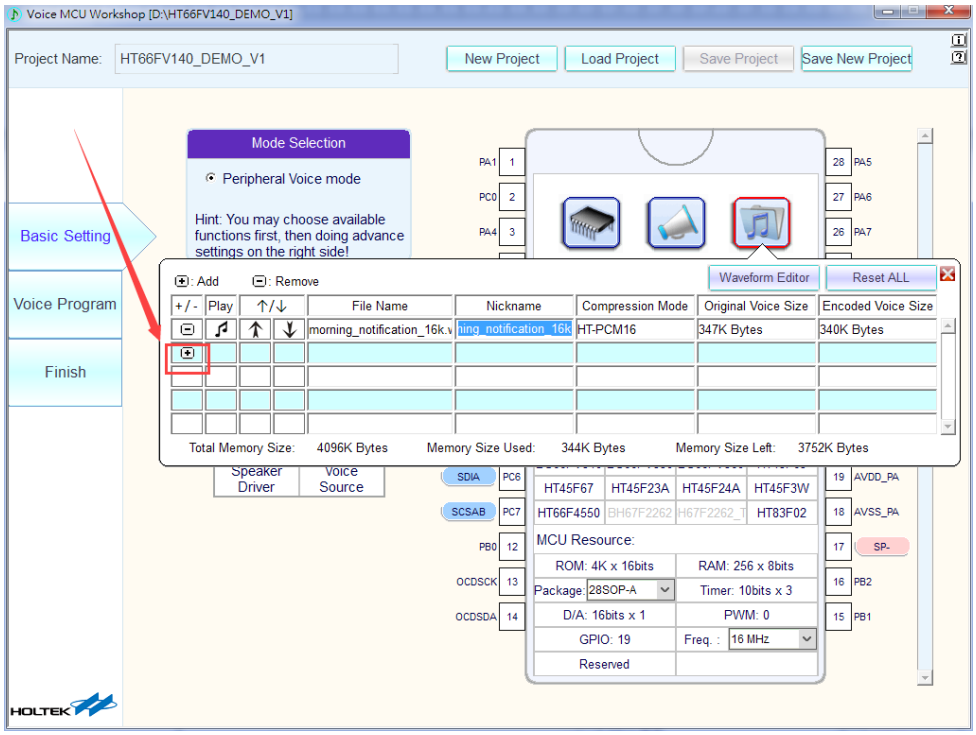
Step 1. After download and installation, double-click the icon  to open the desktop.



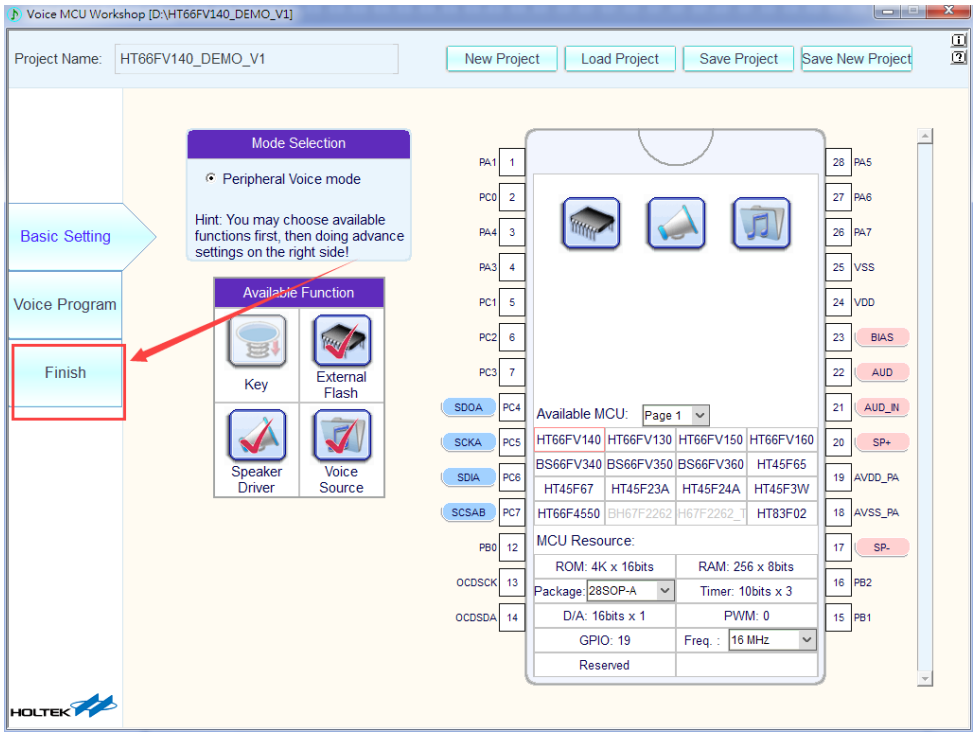
Step 2. Click the "New Project" button to create a new project, then enter a project name and project path. Then select the fourth mode, the voice slave mode, and select the communication mode and IC type. Here the HT66FV140 I^2C slave mode is selected as an example.

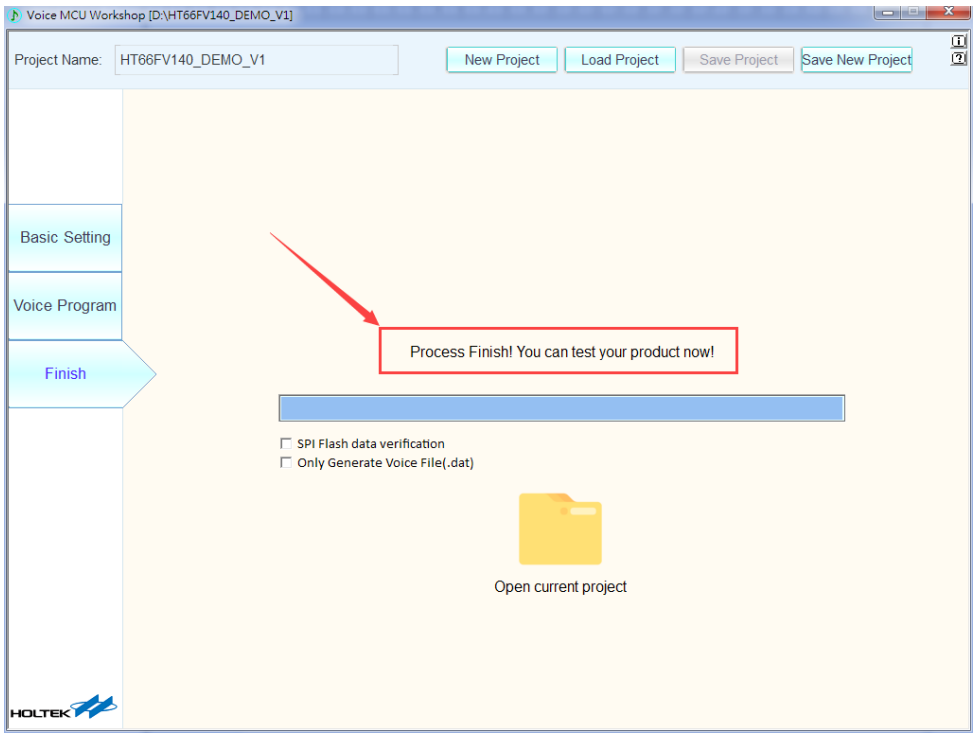


Step 3. Click the "OK" button to create the voice slave mode. All settings have a default setting. Clicking the upper right corner of the red block diagram can change the voice source. Users can also change other default voice sources.

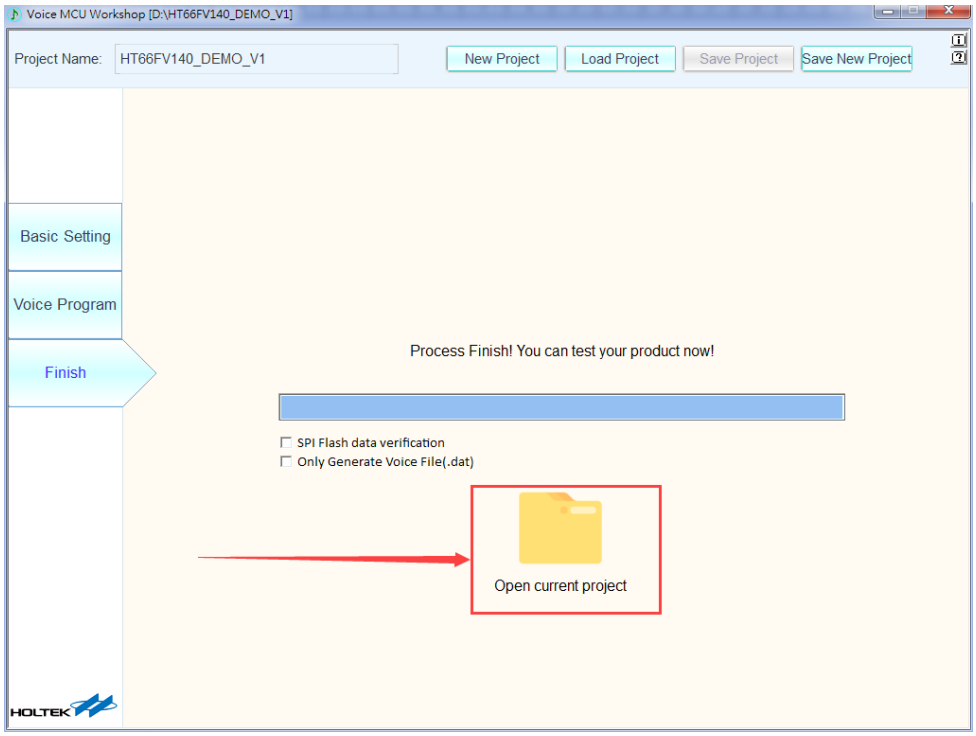


Step 4. Click the left side “Finish” button to program the voice files into the SPI Flash. If users want to program the voice files into the SPI Flash using other means, then this can be implemented by using the voice “.dat” file which is generated by the Voice MCU Workshop.



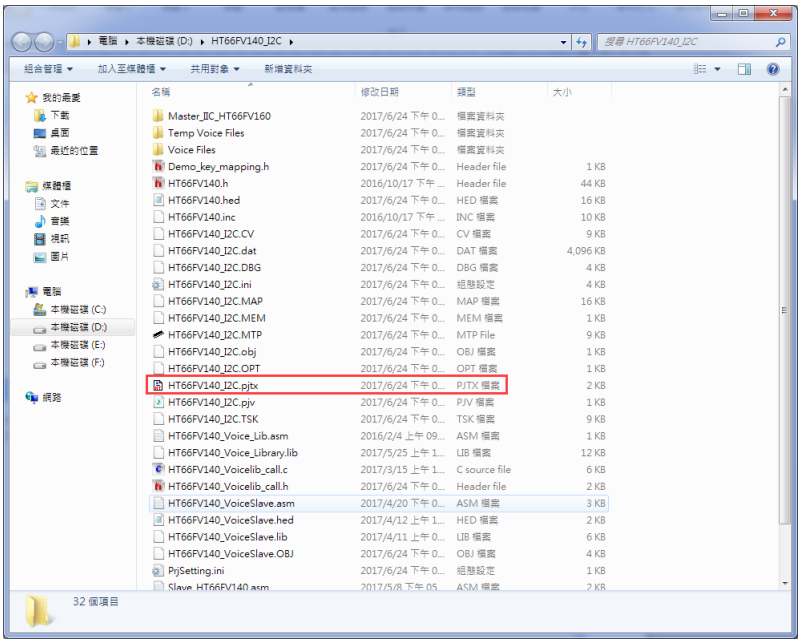


Step 5. After the voice MCU Workshop has indicated that the voice has been programmed, click on the "open current project" icon to open the integrated HT66FV140 I²C slave HT-IDE3000 project folders. The project program can be implemented directly without any user modification.

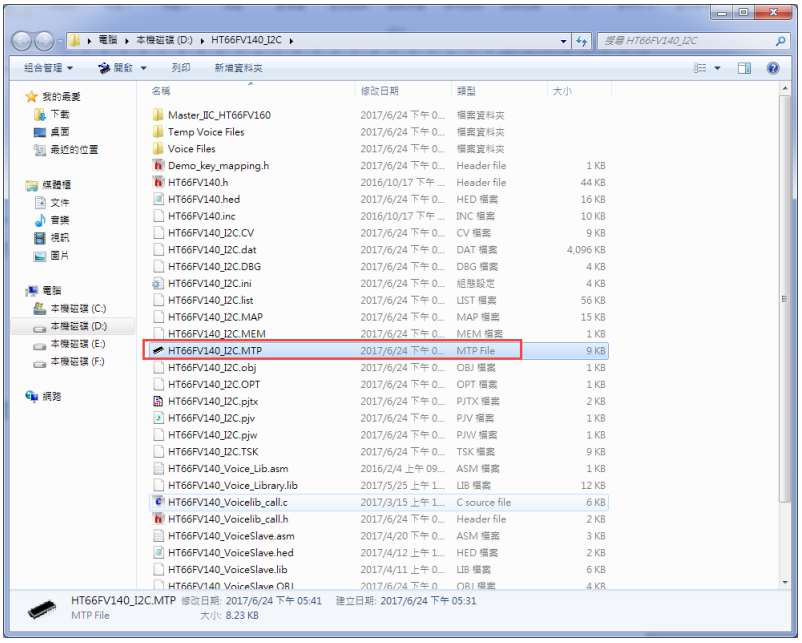


Step 6.

A: If the HT66VV130/HT66VV140 emulation device is used, users can click the "HT66FV140_I2C.PJTX" to open the HT-IDE3000 project and then compile and download the program using the e-Link.

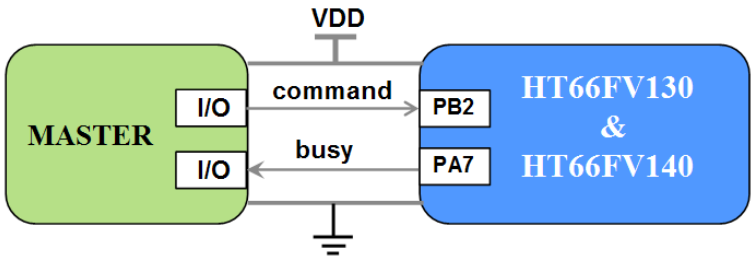


B: If the HT66FV130/HT66FV140 device is used, it needs to download the HT66FV140_I2C.MTP file for programming using the HOPE-3000.



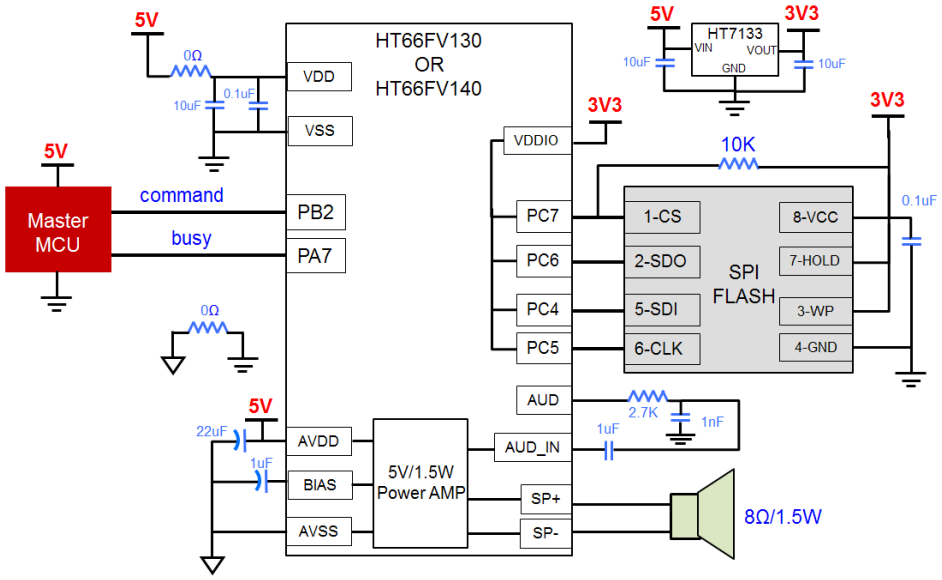
2 Application Circuit Description

Single-wire Communication Application Circuit Description

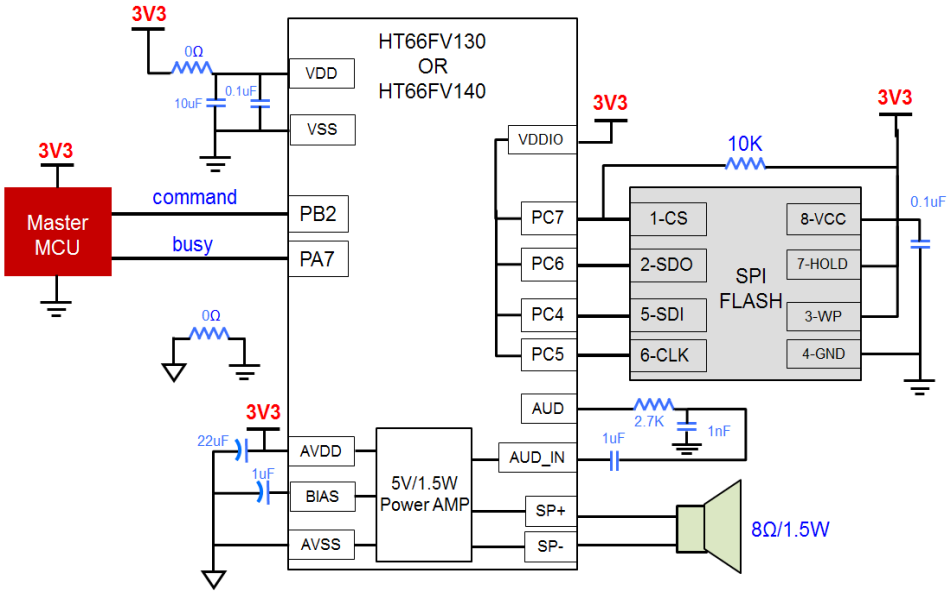


- The HT66FV130/HT66FV140 device command receive pins are both located on PB2. This pin should be connected to the master command transmitter pin.
- The play busy status output on PA7 should be connected to the master play status input detection pin.
- The power and ground pins should be connected to ensure that the communication voltage levels are matched.

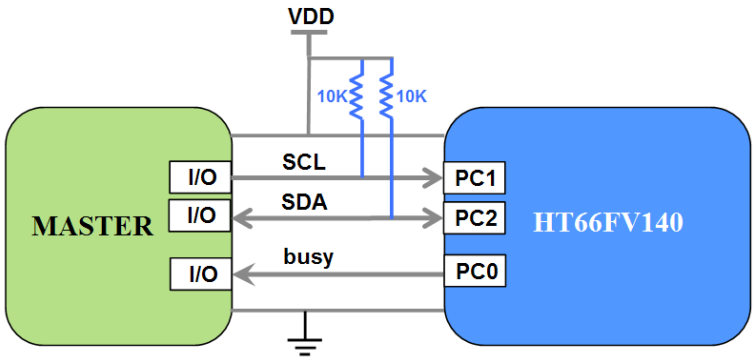
Single-wire Communication Application Circuit – 5V



Single-wire Communication Application Circuit – 3.3V

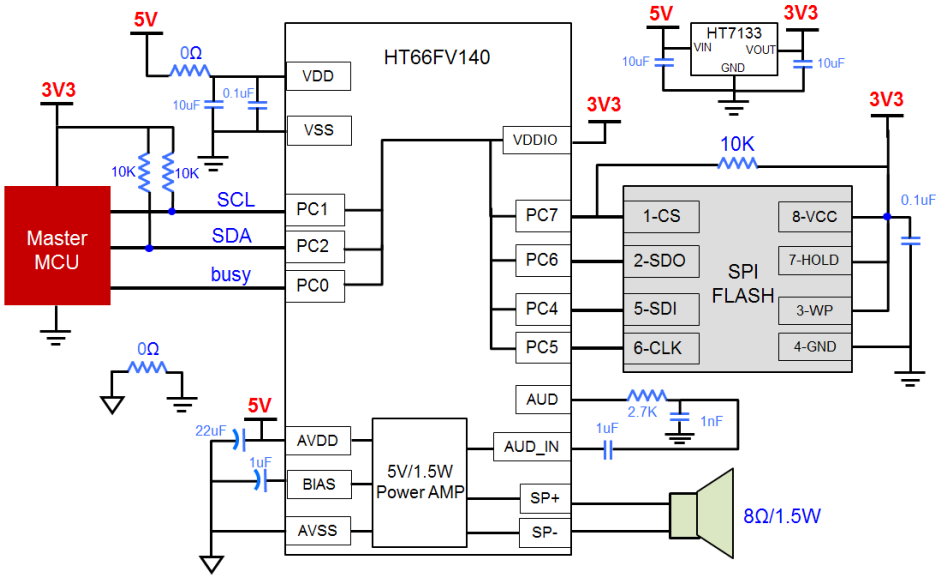


I²C Communication Application Circuit Description

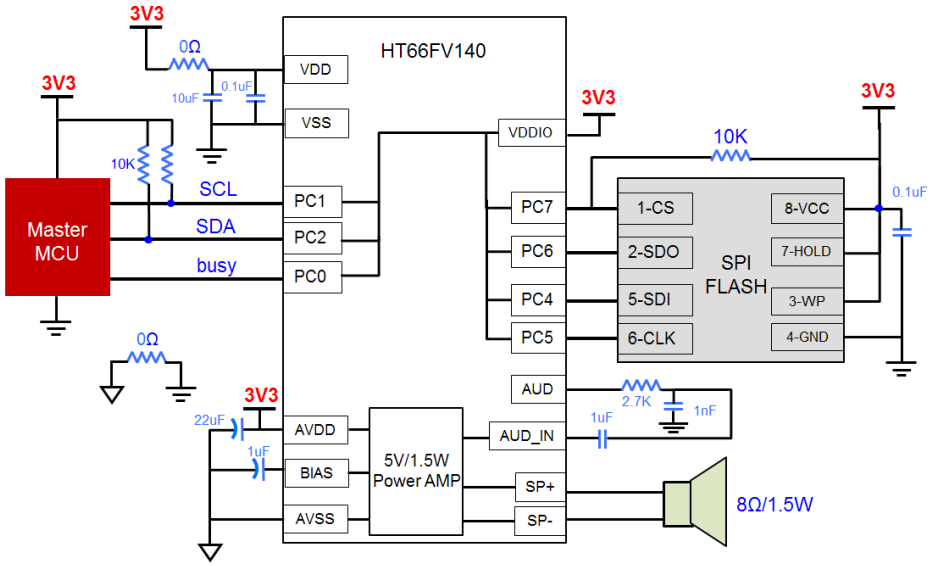


- At the present time the I²C communication only supports the HT66FV140 when used as a slave. It does not support the HT66FV130.
- The I²C clock pin, SCL, is on PC1 and the I²C data pin, SDA, is on PC2 and the play status output pin is on PC0. These pins should be connected to the master correctly. It is recommended that the I²C communication pins should be connected to a 10KΩ pull-high resistor. It should be noted that the communication level can only be 3.3V.
- The power and ground pins should be connected to ensure that the communication voltage levels are matched.

I²C Communication Application Circuit – 5V

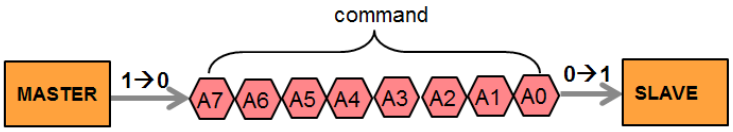


I²C Communication Application Circuit – 3.3V



3 Communication Protocol Description

Single-wire Communication Protocol Description



As shown in the accompanying figure, the master only needs to send a 1-byte command to the slave. The command definitions are as follows:

0X00~0XDB: Set the play address	0x01: Play the first voice source original file
	0x02: Play the second voice source original file
	0x03: Play the third voice source original file
	……
	0xdb: Play the 219th voice source original file
	For example, sending 0x03 means to play the third combination voice in the Voice MCU Workshop "voice program "
0xe0~0xec: Set the Volume Level	0xe0: Set the volume to 0
	0xe1: Set the volume to 1
	0xe2: Set the volume to 2
	……
	0xec: Set the volume to 12
0xf2: Loop playing	Loop playing the current voice
0xf4: Enter the SLEEP Mode	The voice slave enters the SLEEP mode
0xfe: Stop playing	Stop playing the current voice
Single-wire communication the master program example	Addressed in the current project folder

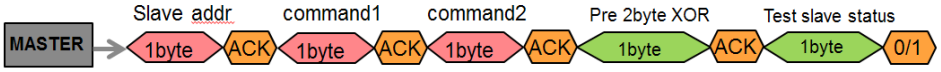
Note: Waking up the voice slave does not require any special operations, each communication can wake up the voice slave, other data definitions are reserved.

Single-wire communication data definition:

Data 0: High level: Low level=1:3, the high level is held for 200μs and the low level is held for 600μs, as shown in the following diagram.

Data 1: High level: low level=3:1, the high level is held for 600μs and the low level is held for 200μs, as shown in the following diagram.

I²C Communication Protocol Description



On the basis of the standard I²C communication format, the master transmits the slave address (1-byte) + play command 1 (1-byte) + play command 2 (1-byte) + XOR check value (1-byte) + test data (1-byte) to the slave.

Slave address (7-bit address +1-bit read/write control)	Currently defined as 0xf6
Play command 1 (8-bit)	1~0xf0: 1~240 voice play addresses
	0xf2: Loop playing
	0xf4: Enter the SLEEP Mode
	0xfe: Stop playing
Play command 2 (8-bit)	0x00~0x0c: Play volume level Others reserved
XOR check value (8-bit)	The XOR check value between play command 1 and play command 2 is sent to the slave to check whether the previous two data blocks have been received correctly
Test data (8-bit)	Used to test the data response from the slave. If the slave responds with a 0, this means that the play command, play command 2 and XOR check values have been received correctly. If the slave responds with a 1, this means that a reception error has occurred. The master will need to execute a new complete I²C communication with the slave to support the error retransmission mechanism
Simulate I²C communication, the master program example In this example, the HT66FV160 device has been chosen as an example. Users can use the HOLTEK HT66FV1X0 development board for testing.	Addressed in the current project folder