



instructables

Getting the Most Out of a WT588D Sound Module

by FlyingDrMike

Electronically activated sounds can add a great deal to many projects. An intruder alert could deliver the sound of dogs barking. An ultrasonic distance detector could speak the distance as numbers. The WT588D module is easy to setup once one knows how. This guide should get you running as painlessly as possible. An example has been given using Arduino that should also be fine for the ESP8266. I derived these from my experience driving WT588D modules from other microcontrollers. I wrote a function for the Arduino code. Hence it is easy to incorporate. Also this should be helpful for writing routines for other devices.

The WT588D can hold up to around 270s of sound/speech at its highest quality or 1000s at 6kHz sampling. However it can string individual sounds together to give a wide capability. Up to 220 'sequences' can be set. For example I wanted an audio notification of distances from 0.01 to 2.20m. I used sound files for 'zero' to 'nine' plus files for 'zero point', 'one point' and 'two point' to deliver the range of 220 numbers from 0.01 to 2.20. So 220 numbers can be delivered with just 13 sound files. Note that the WT588D modules can come with memory from 4Mb to 32Mb. If this is not clear check with the supplier.

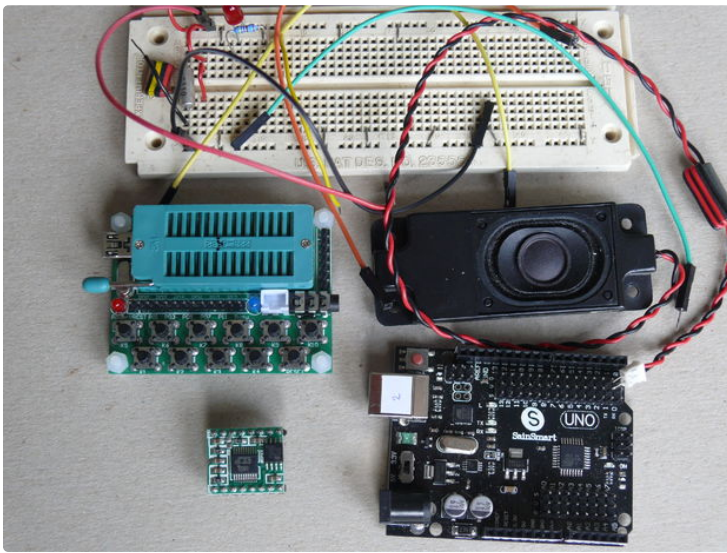
WT588D modules can be driven in several different ways. If only a few sounds or sound collections are being activated it can be set to be driven by triggering input pins. The 16pin module has scope for 4 inputs while the 28pin module has scope for 11. If more than these are needed, as in the case of 220 numbers, one needs to use either the 'one wire' or the 'three wire' interface. I found the 'one wire' method to be unreliable and hence the 'three wire' approach will be used here. There is also a 24 sound 3x8 key matrix option for the 28pin module.

It is worth mentioning another sound module – the WTV020. This has the advantage of taking a 2Gb micro SD card that can be programmed via PC etc. and then inserted. Hence it does not need a programmer. (The WT588D needs a special (but low cost) programmer.) However there are card compatibility problems, including with Samsung cards, and finding cards that work is getting more difficult. This drove me to the WT588D that is now my first choice.

The module can also drive audio amplifiers. Details are given in the WT588D VOICE CHIP MODULE APPLICATION CIRCUIT.pdf in the download referenced later.

Parts needed:

- Arduino (only for the three wire interface)
- WT588D 16pin module (ideally the 32M option)
- WT588D programmer
- 8-16 ohm speaker



Step 1: Sorting the Sound Files

The WT588D requires the sound files in a specific format. They must be mono, and for the best quality - a 22kHz sampling rate, and be 16 bit .wav files. A popular open source sound editor is Audacity, which is pretty straightforward to use. Search on Instructables if you need any help here. Audacity can be downloaded from:

<http://www.audacityteam.org/download/>

You can use your own recordings or search on-line. There are many sound repositories. I got a barking dog sound clip from <http://soundbible.com/> There were several other free sound sites in the same search (Google: dog barking sound file).

A good source for numbers is OpenTX. This RC transmitter firmware needs audio files for numbers for the spoken telemetry values. You can get these from: <http://voices-21.open-tx.org/opentx-taranisplus/en/> . I used english-kilrah.zip. If you want a different

language click the parent directory to see what is available. These files are mono but 32Khz and 32 bit. Hence they need processing in Audacity to enable use in the WT588D.

The process I use in Audacity is:

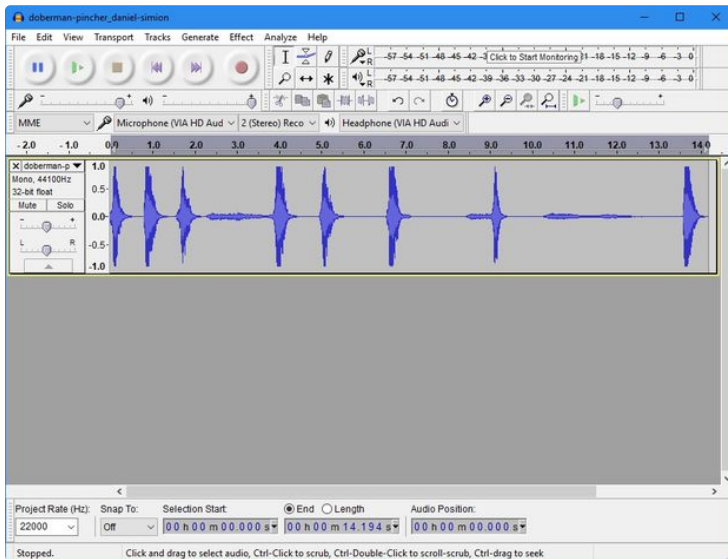
If the source file is stereo then combine to mono via Tracks>Stereo Track to Mono

Enter 22000 in Project Rate (bottom LHS)

Effect>Amplify>New Peak Amplitude (dB): -1, OK

File>Export Audio, enter file name and for 'Save as type' select WAV (Microsoft) signed 16-bit PCM.

Leave the 'Edit MetaData' fields blank.



Step 2: Programming the WT588D and Key Trigger Example

The programming software and data sheets can be downloaded from:

<https://drive.google.com/file/d/0B4IMeS-6RiDJWGNhTHV1aF9pSDg/view>

Extract WT588D.rar and then extract WT588D Software.rar in the extracted files.

Then click WT588D VoiceChip Beta 1.6 Setup.exe to install.

Start the program. Click the 6th Chinese pair of characters (2nd from RHS) to change to English.

The left pane is where the sound files are loaded. The middle 'Equation No.:' is the selection of the 220 sound combinations. The right hand pane is where the sound combinations for each 'Equation No.' is entered, referring to the list of sounds in the left hand pane.

One sound, Key trigger example

This will give us a standalone sound module where a Barking sound is triggered by an input pin.

Get a sound file e.g. from <http://soundbible.com/> and process it on Audacity.

Create a new folder called e.g. Bark. Start the WT588D VoiceChip V1.6 program. Change the language to English as above.

Start with a new project: File>New Project and enter a project name and select the folder. Note some keyboard characters are not recognised.

□

Add files: Right click in the left pane, select Load and browse to the sound file and click Open.

Set the sound selection: right click the right pane (with Equation No: 00H highlighted), select Insert.

□

We now need to set the input and output modes. Click the hammer (or Operation>Options). Set Control Mode to Key Mode. Set Sound Output to PWM if connection a speaker direct to the module or to DAC if using an amplifier. Busy Mode can be either. I default to setting this LOW and connecting a LED from the busy output (Pin15) via a resistor to 5v. Leave the Pull High Resistor ticked. Click OK.

Next we need to compile the file to download. Set the Flashsize (e.g. 32M for 32Mb module). Click the deck of cards (or Operation>Compile(save BIN)). This always gives the error: Fail! Click OK, and OK to the file reference, and OK to BIN File saved!

Now connect the WT558D programmer with module inserted. Note the position and orientation in the photo above.

□

Now start the Download tool via Tool>Download Data. Click connect, the status red circle should change to green. Click Erase to clear the module. Then click DownLoad. Alternatively click OnekeyDownLoad to do these in one operation.

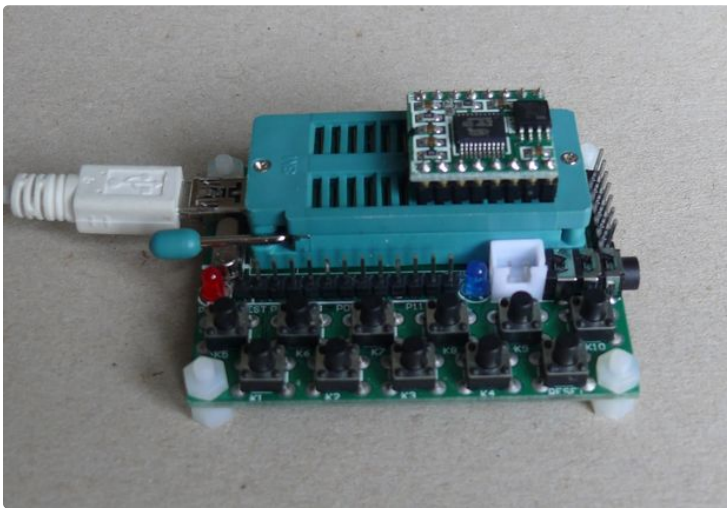
Move the module to a breadboard. Connections needed are:

- Pins 3 and 4 to 8ohm speaker
- Pin 8 ground
- Pin 14 to 3.3v
- Pin 15 optionally to a LED with 470ohm resisitor to 5v
- Pin 16 to 5v

I used an Arduino to provide the 3.3v and 5v supplies. The application note suggests an alternate to use two diodes to drop 5v to 3.3v. I found this is not enough and three would probably be better. Best is a 3.3v regulator. Pin 16 can also be 3.3v when connecting to systems running at this voltage.

Now apply power and trigger the sound by connecting pin13 (P00) to ground:

https://www.youtube.com/embed/IV_y9xRkNtl?rel=0



Step 3: Multiple Sounds 3 Wire Control

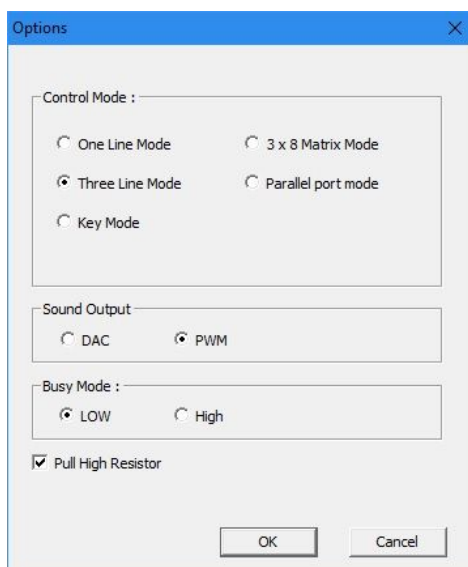
Download the WT588D programming software as in the previous section. Create a new folder. Start the WT588D VoiceChip V1.6 program. Change the language to English as above. Start with a new project: File>New Project and enter a project name and select the folder.

First we have to add the sound files. These will be copied into a directory with the project name. Simply right click within the left pane and select Load. Multiple sound files can be added at a time.

Now select the 'Equation No.' to edit. Then right click the right hand pane, Insert, and add either a .wav file or a silence period. Selecting .WAV brings up the list of sounds loaded earlier into the left pane.

□

For the system to speak numbers from zero to 100 I loaded the processed sound files for zero to 20 plus 30, 40, 50.....100. So the sound for '23' is created by the sound for 20 plus the sound for 3 and so on.



With the sounds loaded and collated we need to set the control and output parameters (Operation>Options). For multiple sounds we want the three wire interface. Directly connected speaker needs PWM. Use of an amplifier would need ADC. Leave the Pull High Resistor ticked. Click OK.

Next we need to compile the file to download. Set the Flashsize (e.g. 32M for 32Mb module). Click the deck of cards (or Operation>Compile(save BIN)). This always gives the error: Fail! Click OK, and OK to the file reference, and OK to BIN File saved!

Now connect the WT558D programmer with module inserted. Note the position and orientation in the photo in the previous step.

□

Now start the Download tool via Tool>Download Data. Click connect, the status red circle should change to green. Click Erase to clear the module. Then click DownLoad. Alternatively click OnekeyDownLoad to do these in one operation.

Now to Arduino. Download the WT588Dnos.ino file attached and put it into a folder named WT588Dnos in your sketches folder. Now start the Arduino IDE and open the sketch.

The key function to copy to your sketches is: void WT588D(byte data)

```
// 3 wire send data to WT588D
```

```
void WT588D(byte data){  
  
digitalWrite(CSpin,LOW);  
  
delay(2);  
  
for(byte i=0;i<8;i++) {  
  
digitalWrite(CLKpin,LOW);  
  
if(data & 1 == 1){  
  
digitalWrite(DATpin, HIGH);  
  
} else {  
  
digitalWrite(DATpin, LOW);  
  
}  
  
delayMicroseconds(25);  
  
digitalWrite(CLKpin,HIGH);  
  
delayMicroseconds(25);  
  
}
```

```
data = data>>1;

}

digitalWrite(CSpin,HIGH);

}
```

The pins are defined at the beginning and can be set to others if required. You could add a reset but I did not find this necessary.

The function call is:

```
WT588D(Num);
```

Where Num is a byte and set to the Equation number collection to be implemented.

Connect your Arduino and compile and load the sketch.

Now connect everything up.

Move the module to a breadboard. The WT588D connections needed are:

- Pins 3 and 4 to 8ohm speaker
- Pin 8 ground
- Pin 10 to Arduino D3 (CLK)
- Pin 11 to Arduino D4 (DATA)
- Pin 12 to Arduino D2 (CHIP-SELECT)
- Pin 14 to 3.3v
- Pin 15 optionally to a LED with 470ohm resisitor to 5v
- Pin 16 to 5v

To test just power up the Arduino.

<https://www.youtube.com/embed/l4lHK29siuw?rel=0>

I hope this helps you add sound to your projects.

Mike



<http://www.instructable...>

Download (<https://cdn.instructables.com/ORIG/FHX/VVZ8/J7UR0O12/FHXVVZ8J7UR0O12.ino>)

(<https://cdn.instructables.com/ORIG/FHX/VVZ8/J7UR0O12/FHXVVZ8J7UR0O12.ino>)