Me Audio Player

1. Overview:

The audio playback module is compatible with the entire series of MakeBlock control boards, with a built-in voice decoding chips, able to play music and recordings. This module's connector is marked in white, meaning that it is controlled by I2C signals, and must be connected to main board's white-marked port. Insert the TF memory card to feel the joy of music and it is very convenient to use.

1. Technological Specifications:

Working voltage: 5V DC

Microphone sensitivity (1Khz): 50-54dB

Microphone impedance: 2.2 kΩ

Microphone signal to noise ratio: 58 db

Speaker rated power: 1W

Speaker rated impedance: 8±15%Ω

Communication method: I2C

Largest electric current: 500mA

Module size: 56 x 41 x 28 mm (LxWxH)

1. Characteristics:

Onboard blue LED solid on means music playback mode, flashing means recording mode.

High sensitivity to sound;

The module's metal hole area is the reference area in contact with the metal beam;

With reverse polarity protection, reverse power supply will not damage the IC;

Supports mBlock graphical programming, suitable for all ages users;

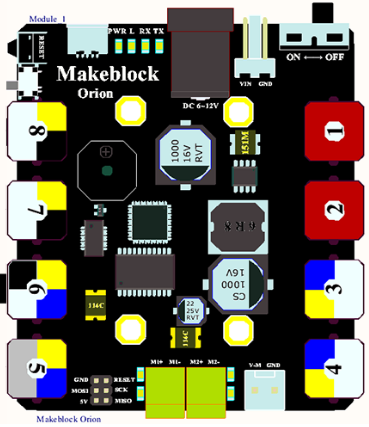
Convenient connection by using the RJ25 port;

Modular installation, compatible with LEGO series;

Module supports Micro USB to copy audio files directly without the need for a card reader.

Module directly supports MP3, WMA, and WAV files

1. Connection Method:

RJ25 connection: Since the audio playback module's connector port is marked in white, when using the RJ25 port, it needs to be connected to the port marked in white on the main control board as well. Taking MakeBlock Orion as an example, you can connect it to ports 3, 4, 6, 7 and 8, as shown.

1. Programming Guide:

Arduino programming

If you are using Arduino programming, you must call the "Makeblock-Library-master" to control the audio playback module.

This program controls the audio playback module through buttons through Arduino programming, achieving the playback, pausing, start recording, and stop recording.

1. #include "MeOrion.h"
2. #include <Wire.h>
4. MeAudioPlayer AudioPlayer(PORT\_3);
5. Me4Button btn(PORT\_7);
7. uint8\_t keyPressed = KEY\_NULL;
8. uint8\_t keyPressedPrevious = KEY\_NULL;
10. **void** setup()
11. {
12. Serial.begin(115200);
13. AudioPlayer.PlayerInit();
14. AudioPlayer.setMusicVolume(100);
15. delay(100);
16. AudioPlayer.stopMusic();
17. delay(100);
18. }
20. **void** loop()
21. {
22. keyPressedPrevious = keyPressed;
23. keyPressed = btn.pressed();
24. **if** (keyPressedPrevious != keyPressed)
25. {
26. **if** (keyPressed == KEY\_1)
27. {
28. Serial.println("KEY1 pressed:playNext");
29. AudioPlayer.playNextMusic();
30. }
31. **if** (keyPressed == KEY\_2)
32. {
33. Serial.println("KEY2 pressed:stopMusic");
34. AudioPlayer.stopMusic();
35. }
36. **if** (keyPressed == KEY\_3)
37. {
38. Serial.println("KEY3 pressed:startRecording");
39. AudioPlayer.startRecordingFileName("R001");
40. }
41. **if** (keyPressed == KEY\_4)
42. {
43. Serial.println("KEY4 pressed:stopRecording");
44. AudioPlayer.stopRecording();
45. }
46. }
47. }

Audio playback module-function list

|  |  |  |
| --- | --- | --- |
| MeAudioPlayer(uint8\_t port) | Choose a port |  |
| playMusicFileIndex(uint16\_t music\_index) | Specify audio file index playback | Numbers: 1.2.3… |
| playMusicFileName(char \*str) | Specify the audio file name playback |  |
| pauseMusic() | Pause playback |  |
| stopMusic() | Stop playback |  |
| playNextMusic() | Next |  |
| playPrevMusic() | Previous |  |
| setMusicVolume(uint8\_t vol) | Set the volume | Range 0-100 |
| setMusicPlayMode(uint8\_t mode) | Set playback mode | 0: Single play  1: Single cycle  2: Playlist loop  3: Random play |
| startRecordingFileName(char \*str) | Select a file name to begin recording |  |
| stopRecording() | Stop recording |  |

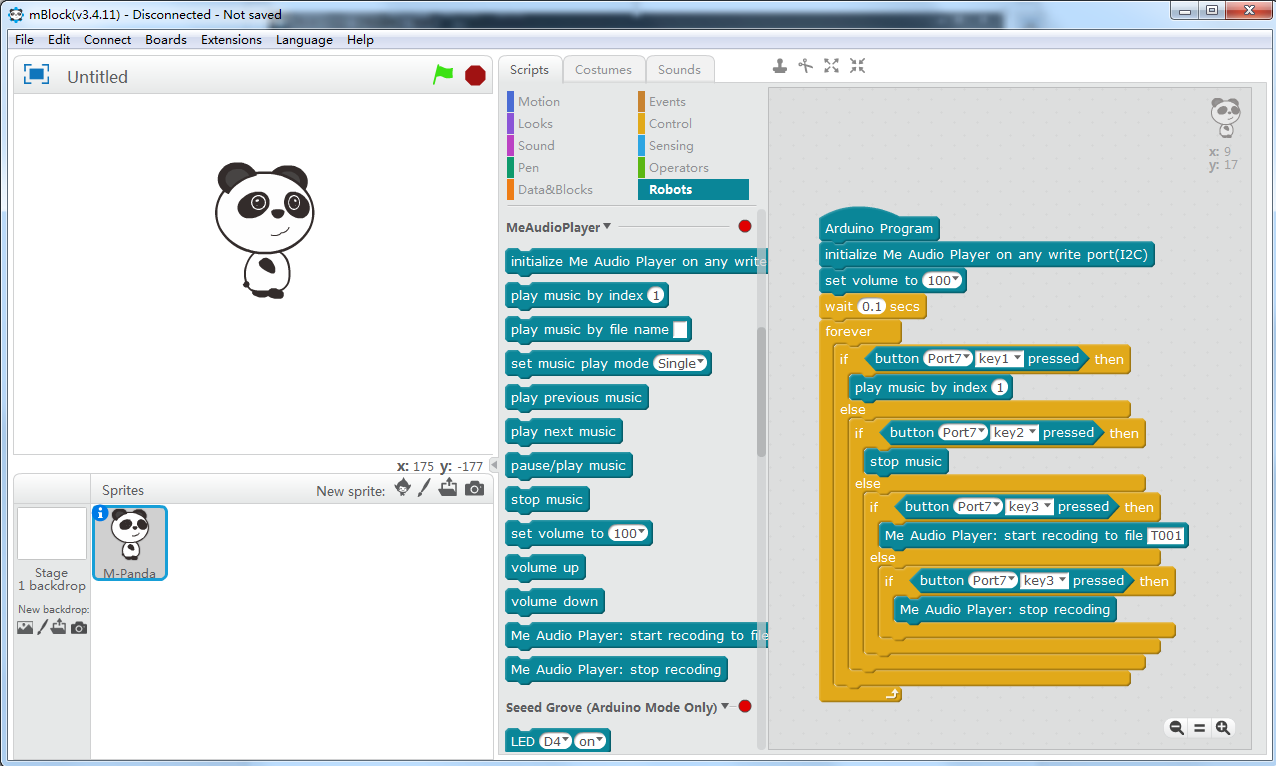
mBlock programming

Audio playback module supports the mBlock programming, The following is an introduction to the module instructions.

|  |  |
| --- | --- |
|  | Choose a port |
|  | Specify audio file index playback |
|  | Specify the audio file name playback |
|  | Set playback mode: Single play, single cycle, playlist loop, random play |
|  | Play previous audio |
|  | Play next audio |
|  | Pause/resume playback |
|  | Stop playback |
|  | Volume setting |
|  | Increase volume |
|  | Decrease volume |
|  | Begin recording with the name "T001" |
|  | Stop recording |

The following is an example of how to use mBlock to control the audio playback module：

This program can control audio playback through the button module. To achieve playback, pause, start recording, and stopping recording audio files, the following are the results:



1. Audio file format instructions:

* Please add an appropriate delay after After the volume setting blocks, and wait for it to take effect.
* Using external memory TF card to store audio files, support MP3, WAV, WMA high quality audio format files.
* Using FAT and FAT32 file systems.
* The audio file naming format supports English naming (case-insensitive). English and numbers are mixed and the length of the naming is recommended to be less than 8 characters. For example: Hello.MP3, T002.MP3, and R000001.MP3. (It is not recommended to use pure number naming.)
* Sorting of audio files in TF card: It is recommended to sort by file name
* This module does not support Chinese naming of audio files
* Do not use special character names such as: v1.0", o\_o0, … ( not support)