

Grove Speech Recognizer Kit for Arduino



The Grove Speech Recognizer Kit for Arduino is designed for prototype of Smart Home which includes some basic home elements such as Speech Recognizer, Infrared Emitter. You can learn the functions and applications of Speech Recognizer quickly via this kit, the guideline shows some common demos, let's say you

can play music via your speak command 'Play music', or turn on the light according to the corresponding command.



[<https://www.seeedstudio.com/Grove-Speech-Recognizer-kit-for-Arduino-p-2726.html>]

Part List

1. Grove - Speech Recognizer SKU:101020232
2. [Grove - MP3 v2.0](#) [/Grove-MP3_v2.0]
3. [Grove - RTC](#) [/Grove-RTC/]
4. [Base Shield](#) [/Base_Shield_V2/]
5. [Grove - Infrared Receiver](#) [/Grove-Infrared_Receiver/]
6. [Grove - Infrared_Emitter](#) [/Grove-Infrared_Emitter/]

Application

How to turn on the TV

Sometimes, we want to control something by our voice, such as TV, air condition. Those devices are controlled by Infrared Remote controller, so we need to know what the code of each button of infrared remote controller first, and then put the code under our

program. Finally your voice could trigger those button codes and you can control those devices like infrared remote controller.

Before create this demo you should prepare some necessary components:

- Grove - Infrared Receiver
- Grove - Infrared Emitter
- Grove - Speech Recognizer
- Base Shield
- Arduino UNO
- Infrared Remote controller

Get code of a infrared remote controller via Grove - Infrared Receiver

Download necessary libraries from github : [IRSendRev](https://github.com/Seeed-Studio/IRSendRev) [https://github.com/Seeed-Studio/IRSendRev] ,pay attention to the path of your libraries: [.../arduino/libraries](#)

- Now open the example file of "recv" : File --> Sketchbook --> libraries --> IRSendRev --> recv.
- Upload the program to your Arduino UNO.
- Plug Base Shield onto the Arduino UNO, and plug the Grove - Receiver onto the D2 port of Base Shield.

```
const int pinRecv = 2;
```



Also you can change another port while modifying the definition of pin.

- Open the monitor of Arduino UNO.

Press a button of any Infrared Remote Controller, you'll get the detailed information code of the button, see following:

The screenshot shows the Arduino IDE interface. The left pane displays the code for the 'recv' sketch, which defines constants for IR data bits and pins, and implements a loop that checks for IR data and receives it into a buffer. The right pane shows the serial monitor output, which displays the results of an IR reception: 'init over', followed by parameters like LEN, START_H, START_L, DATA_H, DATA_L, DATA_LEN, and the raw data bytes in hexadecimal and decimal.

```

recv | Arduino 1.0.5
#define BIT_DATA_H 3
#define BIT_DATA_L 4
#define BIT_DATA_LEN 5
#define BIT_DATA 6

const int pinRecv = 2; // ir receiver connect to D2

void setup()
{
  Serial.begin(115200);
  IR.Init(pinRecv);
  Serial.println("init over");
}

unsigned char dta[20];

void loop()
{
  if(IR.IsDta()) // get IR data
  {
    IR.Recv(dta); // receive data to dta
  }
}

Done uploading.
Binary sketch size: 4,480 bytes (of a 32,256 byte maximum)
42 Arduino Uno on /dev/tty.usbserial-A6005pkq
init over
+-----+
LEN = 9
START_H: 179  START_L: 89
DATA_H: 11  DATA_L: 33

DATA_LEN = 4
DATA: 0x32  0xA6  0x50  0xAF
DATA: 50  166  80  175
+-----+
Autoscroll No line ending 115200 baud

```

Write down the detailed information of the button you want to press. Following are the information of a button:

The screenshot shows a terminal window with the following output, which is a copy of the data from the serial monitor in the previous image:

```

1 +-----+
2 LEN = 9
3 START_H: 179  START_L: 88
4 DATA_H: 11  DATA_L: 33
5
6 DATA_LEN = 4
7 DATA: 0x80  0x7E  0x10  0xEF
8 DATA: 128  126  16  239
9 +-----+

```

Modify the IRSend program

Now we can use the previous information code of a button.

- Open the example file of "send" : File --> Sketchbook --> libraries --> IRSendRev --> send.
- Upload the program to your Arduino UNO.
- Plug Base Shield onto the Arduino UNO, and plug the Grove - Receiver onto the D3 port of Base Shield.

First we need modify the default information of the button:

```

1  unsigned char dtaSend[20];
2
3  void dtaInit()
4  {
5      .....
6  }
```

Modify those information according to the previous one you wrote:

```

1  unsigned char dtaSend[20];
2
3  void dtaInit()
4  {
5      dtaSend[BIT_LEN]      = 9;           // all data
6      dtaSend[BIT_START_H]  = 179;        // the logi
7      dtaSend[BIT_START_L]  = 88;        // the logi
8      dtaSend[BIT_DATA_H]   = 11;        // the logi
9      dtaSend[BIT_DATA_L]   = 33;        // the logi
10
11     dtaSend[BIT_DATA_LEN]  = 4;         // Number of
12
13     dtaSend[BIT_DATA+0]    = 128;       // data th
14     dtaSend[BIT_DATA+1]    = 126;
15     dtaSend[BIT_DATA+2]    = 16;
16     dtaSend[BIT_DATA+3]    = 239;
17     //dtaSend[BIT_DATA+4]   = 192;
18     //dtaSend[BIT_DATA+5]   = 63;
19 }
```

You can observe that:

```
DATA_LEN = 4
```



So you can note or delete those two lines.

```
1 //dtaSend[BIT_DATA+4] = 192;
2 //dtaSend[BIT_DATA+5] = 63;
```



Of course you can also define many buttons:

```
1 #include <IRSendRev.h>
2
3 #define BIT_LEN      0
4 #define BIT_START_H  1
5 #define BIT_START_L  2
6 #define BIT_DATA_H   3
7 #define BIT_DATA_L   4
8 #define BIT_DATA_LEN 5
9 #define BIT_DATA     6
10
11 const int ir_freq = 38;
12
13 unsigned char OpenTV[20];
14 unsigned char CloseTV[20];
15 unsigned char IncreaseTemp[20];
16 unsigned char DecreaseTemp[20];
17
18 void OpenTVInit()
19 {
20     OpenTV[BIT_LEN] = 11;
21     OpenTV[BIT_START_H] = 179;
22     /*..... omit .....*/
23 }
24
```



```
25 void CloseTVInit()
26 {
27     CloseTV[BIT_LEN]      = 11;
28     CloseTV[BIT_START_H] = 179;
29     /*..... omit .....*/
30 }
31
32 void IncreaseTempInit()
33 {
34     IncreaseTemp[BIT_LEN]      = 11;
35     IncreaseTemp[BIT_START_H] = 179;
36     /*..... omit .....*/
37 }
38
39 void DecreaseTempInit()
40 {
41     DecreaseTemp[BIT_LEN]      = 11;
42     DecreaseTemp[BIT_START_H] = 179;
43     /*..... omit .....*/
44 }
45
46 void setup()
47 {
48     OpenTVInit();
49     CloseTVInit();
50     IncreaseTempInit();
51     DecreaseTempInit();
52 }
53
54 void loop()
55 {
56     IR.Send(OpenTV, 38);
57     delay(200);
58     IR.Send(CloseTV, 38);
59     delay(200);
60     IR.Send(IncreaseTemp, 38);
61     delay(200);
62     IR.Send(DecreaseTemp, 38);
63
64     delay(2000);
65 }
```

Add Speech Recognizer to your IRSend program

- It's very simple to use Speech Recognizer alone, please see the wiki of it first.

We can choose what buttons we need according to the table<ref>Table of return value, [Grove - Speech Recognizer](#) [/Seeed_BLE_Shield]</ref> of return value of Speech Recognizer. Following is the table of return value:

Command	Return Value
Turn on the light	1
Turn off the light	2
Play music	3
Pause	4
Next	5
Previous	6
Up	7
Down	8
Turn on the TV	9
Turn off the TV	10
Increase temperature	11

Decrease temperature	12
What's the time	13
Open the door	14
Close the door	15
Left	16
Right	17
Stop	18
Start	19
Mode 1	20
Mode 2	21
Go	22

The following example uses two commands: "Turn of the TV" and "Turn off the TV"

- After reading, we can embed the program of Speech Recognizer in the IRSend program. See the following completed program:

```
1 #include <IRSendRev.h>
2 #include <SoftwareSerial.h>
3
4 /*===== IR type =====*/
```



```

5
6  #define BIT_LEN          0
7  #define BIT_START_H      1
8  #define BIT_START_L      2
9  #define BIT_DATA_H       3
10 #define BIT_DATA_L       4
11 #define BIT_DATA_LEN     5
12 #define BIT_DATA         6
13 const int ir_freq = 38;           // 38k
14
15 /* ===== How many IR buttons you wanna send =====
16
17 unsigned char OpenTV[20];
18 unsigned char CloseTV[20];
19
20
21 /*===== Choose the pins of Speech Recognizer =====
22
23 #define SOFTSERIAL_RX_PIN  5
24 #define SOFTSERIAL_TX_PIN  6
25
26 SoftwareSerial speech(SOFTSERIAL_RX_PIN,SOFTSERIAL_T
27
28
29 /* ===== How to write the IR data =====*/
30 /* ===== You can get those data via IR Recevier =====
31
32 void OpenTVInit()
33 {
34     OpenTV[BIT_LEN]          = 9;           // all da
35     OpenTV[BIT_START_H]      = 180;        // the
36     OpenTV[BIT_START_L]      = 88;        // the L
37     OpenTV[BIT_DATA_H]       = 11;        // the L
38     OpenTV[BIT_DATA_L]       = 33;        // the L
39
40     OpenTV[BIT_DATA_LEN]     = 4;         // Number
41
42     OpenTV[BIT_DATA+0]       = 50;        // data
43     OpenTV[BIT_DATA+1]       = 166;
44     OpenTV[BIT_DATA+2]       = 80;
45     OpenTV[BIT_DATA+3]       = 175;

```

```
46 }
47
48 void CloseTVInit()
49 {
50     CloseTV[BIT_LEN] = 9; // all d
51     CloseTV[BIT_START_H] = 178; // the
52     CloseTV[BIT_START_L] = 89; // the
53     CloseTV[BIT_DATA_H] = 10; // the
54     CloseTV[BIT_DATA_L] = 33; // the
55
56     CloseTV[BIT_DATA_LEN] = 4; // Numbe
57
58     CloseTV[BIT_DATA+0] = 50; // data
59     CloseTV[BIT_DATA+1] = 166;
60     CloseTV[BIT_DATA+2] = 168;
61     CloseTV[BIT_DATA+3] = 87;
62 }
63
64
65 void setup()
66 {
67     OpenTVInit()
68     CloseTVInit()
69     Serial.begin(9600);
70     speech.begin(9600);
71     speech.listen();
72 }
73
74 void loop()
75 {
76     int a=0;
77
78     if(speech.available())
79     {
80         a = speech.read(); // Read the return val
81         switch (a)
82         {
83             case 9: // if (retu
84                 IR.Send(OpenTV, 38);
85                 delay(1000);
86                 break;
```

```
87         case 10:
88             IR.Send(CloseTV, 38);
89             delay(1000);
90             break;
91         default:
92             break;
93     }
94 }
95 }
```

How to control music

Watch the commands of Speech Recognizer, there're many commands which are related to music, such as "Play music", "Pause", "Stop", "Previous", "Next". So let's do this Speech Music Box!

Download necessary libraries from github:

[Grove_Serial_MP3_Player_V2.0](https://github.com/Seeed-Studio/Grove_Serial_MP3_Player_V2.0) [https://github.com/Seeed-Studio/Grove_Serial_MP3_Player_V2.0]<ref>library, [Grove - MP3 v2.0](#) [/Grove-MP3_v2.0]</ref>, **pay attention to the path of your libraries: `.../arduino/libraries`** .

Useful functions about Grove - MP3 v2

There're some useful basic functions of Grove - MP3:

```
1 PlayPause(); // pause music
2 PlayResume(); // restart a music
3 PlayNext(); // next song
4 PlayPrevious(); // previous song
5 PlayLoop(); // loop all songs
6 SetVolume(uint8_t volume); // set volume. default value
7 IncreaseVolume(); // increase volume
8 DecreaseVolume(); // decrease volume
```

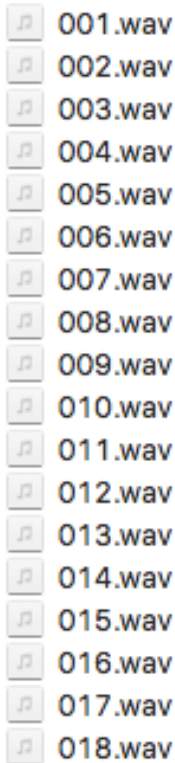
But there're some special functions you need to pay attention:

- `SelectPlayerDevice(uint8_t device)`. *The default device value is 0x02, Select SD card as the player device.*

```
SelectPlayerDevice(0x02);
```

- `SpecifyMusicPlay(uint16_t index)`. *play a song by name.*

Pay attention to the name of songs, you can set a name like following type:



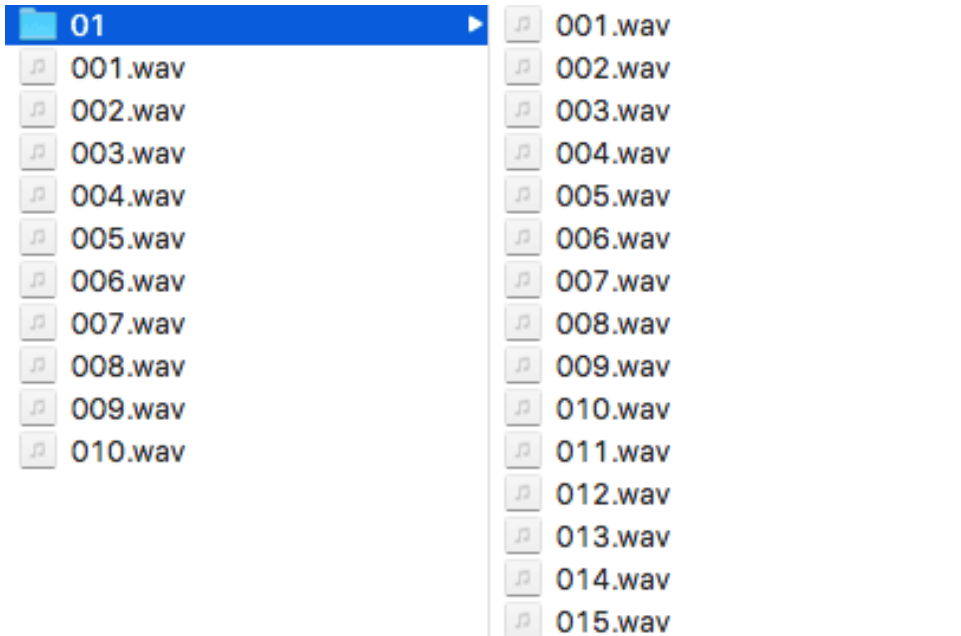
- 001.wav
- 002.wav
- 003.wav
- 004.wav
- 005.wav
- 006.wav
- 007.wav
- 008.wav
- 009.wav
- 010.wav
- 011.wav
- 012.wav
- 013.wav
- 014.wav
- 015.wav
- 016.wav
- 017.wav
- 018.wav

So we can play the song "005" by this function:

```
SpecifyMusicPlay(5);
```

* Specify `folderPlay(uint8_t folder, uint8_t index);` // play a song by name in a appointed folder

Furthermore, sometimes we may play music in a new folder, the previous function comes in handy.



Note

The folder index range is 01-99, so the folders' name will only be any number 01 between 99. Pay attention to the number between 1-9, the name of folder should be added to 01-09.

So if we want to play the "005" in the folder "01", we can do this:

```
SpecifyfolderPlay(1,5);
```



- Other attention:

Each song has its length of time, so if you want to appoint songs one by one, there've two ways you can choose:



```

1  delay(length of time); // delay the length of time until
2
3  while(QueryPlayStatus() != 0); // Return: 0: played out

```

Normally, we can use the latter. For example:

```

1  SpecifyMusicPlay(1);
2  while(QueryPlayStatus() != 0);
3  SpecifyMusicPlay(2);
4  while(QueryPlayStatus() != 0);
5  SpecifyMusicPlay(3);
6  while(QueryPlayStatus() != 0);

```

Integrated Program

OK, let's embed Speech Recognizer and Grove - MP3 together. The following demo can control some MP3 functions: play music, pause music, continue music, next song, previous song.

- Copy the program and upload it to your Arduino UNO.
- Plug Base Shield onto the Arduino UNO, and plug the Grove - MP3 v2 onto the D2 port of Base Shield.

```

1  #include <SoftwareSerial.h>
2  #include <MP3Player_KT403A.h>
3
4  /***** Define the pins of MP3 *****/
5  SoftwareSerial mp3(2, 3);
6
7  /***** Define the pins of Speech Recognizer *****/
8  #define SOFTSERIAL_RX_PIN 5
9  #define SOFTSERIAL_TX_PIN 6
10
11  SoftwareSerial speech(SOFTSERIAL_RX_PIN, SOFTSERIAL_T
12

```

```
13
14 void setup()
15 {
16     mp3.begin(9600);
17     speech.begin(9600);
18     Serial.begin(9600);
19     delay(100);
20
21     SelectPlayerDevice(0x02); // Select SD card
22     SetVolume(0x15); // Set the volume
23 }
24
25 void loop()
26 {
27     int a=0;
28     if(speech.available())
29     {
30         a = speech.read(); // Read the return value
31         switch (a)
32         {
33             case 3: // speech command : Play music
34                 SpecifyMusicPlay(1); // MP3: play the music
35                 break;
36             case 4: // speech command : Pause
37                 PlayPause(); // MP3: pause music
38                 break;
39             case 19: // speech command : Start
40                 PlayResume(); // MP3: continue music
41                 break;
42             case 5: // speech command : Next
43                 PlayNext(); // MP3: play next song
44                 break;
45             case 6: // speech command : Previous
46                 PlayPrevious(); // MP3: play previous song
47                 break;
48             default:
49                 break;
50         }
51
52         delay(1000);
53     }
```



```
54 }
```

How to broadcast real-time

Did you use Grove - MP3 to broadcast real-time? Let's have a try via Speech Recognizer, Grove - MP3, and Grove - RTC.

Adjust the real-time

Download necessary libraries from github: [RTC_DS1307](https://github.com/Seeed-Studio/RTC_DS1307) [https://github.com/Seeed-Studio/RTC_DS1307]<ref>library, Grove - RTC [/Grove-RTC]</ref>, pay attention to the path of your libraries: [.../arduino/libraries](#) .

- Open the example file of "SetTimeAndDisplay" : File --> Sketchbook --> libraries --> RTC_DS1307 --> SetTimeAndDisplay.
- Plug Base Shield onto the Arduino UNO, and plug the Grove - RTC onto the I2C of Base Shield.
- Set the right time of RTC.

```
1 clock.fillByYMD(2016,1,19);//May 23,2016
2 clock.fillByHMS(15,28,30);//15:28 30"
3 clock.fillDayOfWeek(Mon);//Saturday
```

- Upload the modified program to your Arduino UNO.

Text-to-speech

As all we known, it has 60 numbers while broadcasting time (0 ~ 59), and we can mouth some words before the MP3 broadcast the time (It's). So we need to add 61 sound files in SD Card.

But there're some tips about SD card you need to pay attention to:

- Form of SD Card: FAT32.
- Format SD Card before deleting any songs; Nothing should to do while adding any songs.
- The sequence of playing is depended on the sequence of song addition in SD Card. *so if we want to play some songs in order, we need to add those songs in SD Card in order.*

By the way, we have ranked 61 sound files in a folder, you can [download it](#)

[https://files.seeedstudio.com/wiki/Grove_Speech_Recognizer_Kit_for_Arduino/res/Sound_file.zip] and copy it to your SD Card. Of course you may need to format SD Card first.

Name of Sound File	NO. of file in SD Card	Voice Text
000	1th	0
001	2th	1
...
059	60th	59
060	61th	It's

Integrated Program and broadcast real-time

- Plug Base Shield onto the Arduino UNO; plug Grove - MP3 v2 onto the D2 port of Base shield; plug Grove - Speech Recognizer

onto the D5 port of Base Shield; plug Grove - RTC onto the I2C port of Base Shield.

- Copy the following codes on a new sketch of Arduino IDE and upload the program to Arduino UNO.
- Say "HiCell, What's the time" , the MP3 will broadcast the real-time.

```

1  #include <Wire.h>
2  #include "DS1307.h"
3  #include <SoftwareSerial.h>
4  #include <MP3Player_KT403A.h>
5
6  ***** Define the pins of MP3 *****/
7  SoftwareSerial mp3(2, 3);
8
9  ***** Define the pins of Speech Recognizer *****/
10 #define SOFTSERIAL_RX_PIN  5
11 #define SOFTSERIAL_TX_PIN  6
12
13 SoftwareSerial speech(SOFTSERIAL_RX_PIN,SOFTSERIAL_T
14
15 ***** Define a object of DS1307 class *****/
16 DS1307 clock;//define a object of DS1307 class
17
18 void setup ()
19 {
20   mp3.begin(9600);
21   speech.begin(9600);
22   clock.begin();
23   Serial.begin(9600);
24   delay(100);
25
26   SelectPlayerDevice(0x02);    // Select SD card
27   SetVolume(0x15);    // Set the volume, the range is
28 }
29
30 void loop ()

```

```
31 {
32   int a=0;
33   speech.listen(); // start to receiver data from
34   if(speech.available())
35   {
36     a = speech.read(); // Read the return value fr
37     if(a==13)
38     {
39       clock.getTime(); // get the real-time from G
40       int b=1+clock.hour; // get hour data; because
41       int c=1+clock.minute; // get hour data; becau
42
43       mp3.listen(); // start to receiver data from
44       SpecifyMusicPlay(61); // The voice "It's" is
45       while(QueryPlayStatus() != 0); // play next s
46       SpecifyMusicPlay(b); // play the name of "b"
47       while(QueryPlayStatus() != 0);
48       SpecifyMusicPlay(c); // play the name of "c"
49       while(QueryPlayStatus() != 0);
50     }
51   }
52   delay(1000);
53 }
```

Resource

- [Github: IRSendRev](https://github.com/Seeed-Studio/IRSendRev) [https://github.com/Seeed-Studio/IRSendRev]
- [Github: MP3](https://github.com/Seeed-Studio/Grove_Serial_MP3_Player_V2.0) [https://github.com/Seeed-Studio/Grove_Serial_MP3_Player_V2.0]
- [Github: RTC](https://github.com/Seeed-Studio/RTC_DS1307) [https://github.com/Seeed-Studio/RTC_DS1307]
- [Sound files of broadcast](https://files.seeedstudio.com/wiki/Grove_Speech_Recognizer_) [https://files.seeedstudio.com/wiki/Grove_Speech_Recognizer_

Kit_for_Arduino/res/Sound_file.zip]

Tech Support

Please submit any technical issue into our [forum](https://forum.seeedstudio.com/)
[<https://forum.seeedstudio.com/>].



[https://www.seeedstudio.com/act-4.html?utm_source=wiki&utm_medium=wikibanner&utm_campaign=newproducts]