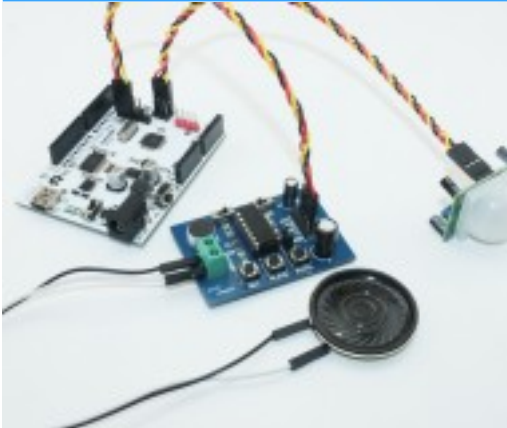


# Voice Record Module – Isd1820

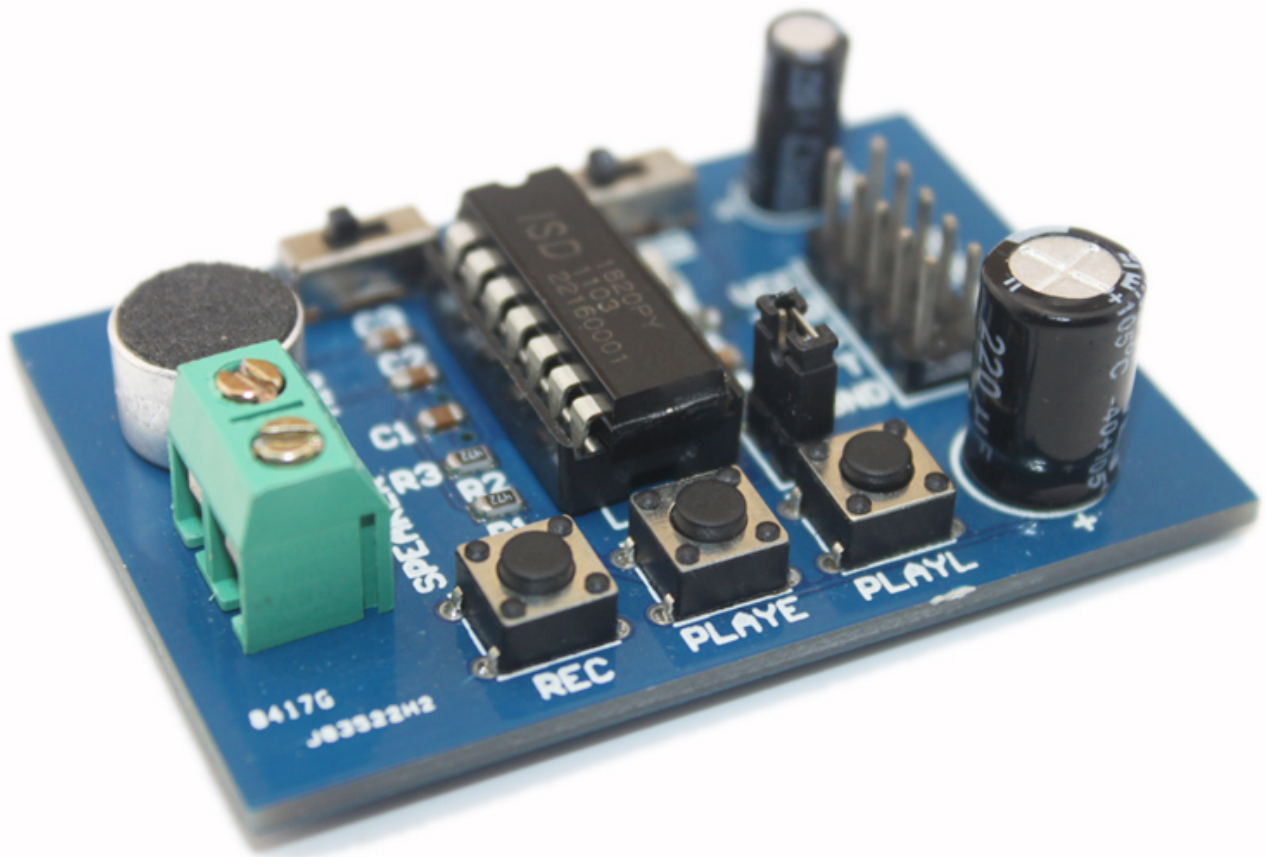
14

Nov



**Voice Record Module** is based on ISD1820, which is a multiple-message record/playback device. It offers true single-chip voice recording, non-volatile storage, and **playback** capability for **8 to 20 seconds**. The sample rate is 3.2kHz and the total time for the Recorder is 20s.

This module is very easy to use, which you can directly control by a push button on the board or by a microcontroller such as Arduino, STM32, ChipKit, etc. From these, you can easily control record, playback, and repeat, and so on.

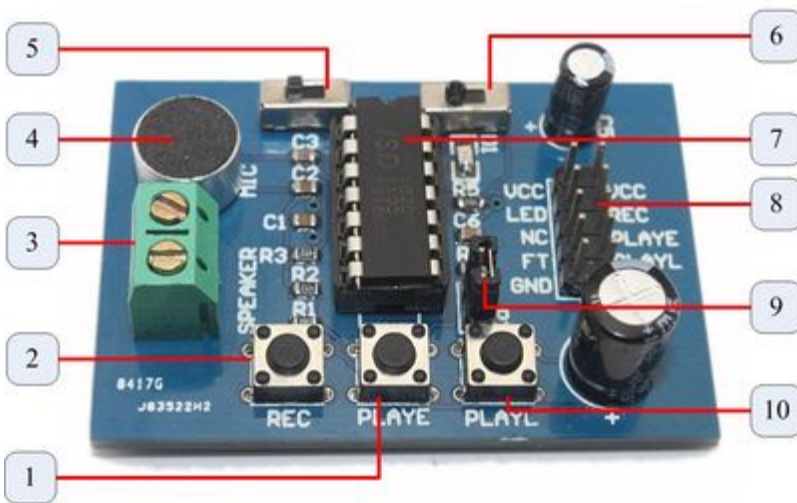


## Feature

- Push-button interface, playback can be edge or level activated
- Automatic power-down mode
- On-chip  $8\Omega$  speaker driver
- Signal 3V Power Supply
- Can be controlled both manually or by MCU
- Sample rate and duration changable by replacing a single resistor
- Record up to 20 seconds of audio
- Dimensions: 37 x 54 mm

If you want change record duration, an external resistor is necessary to select the record duration and sampling frequency, which can range from 8 – 20 seconds (4-12kHz sampling frequency). The Voice Record Module of our provide default connect 100k resistor through P2 by short cap. So the default record duration is 10s.

ROSC	Duration	Sample Rate	Bandwidth
80K $\Omega$	8 secs	8.0KHz	3.4KHz
100K $\Omega$	10 secs	6.4KHz	2.6KHz
120K $\Omega$	12 secs	5.3KHz	2.3KHz
160K $\Omega$	16 secs	4.0KHz	1.7KHz
200K $\Omega$	20 secs	3.2KHz	1.3KHz



1. **PLAYE** – Playback, Edge-activated: When a HIGH-going transition is detected on continues until an End-of-Message (EOM) marker is encountered or the end of the memory space is reached.
2. **REC** – The REC input is an active-HIGH record signal. The device records whenever REC is HIGH. This pin must remain HIGH for the duration of the recording. REC takes precedence over either playback (PLAYL or PLAYE) signal.
3. **Speaker Outputs** – The SP+ and SP- pins provide direct drive for loudspeakers with impedances as low as 8 $\Omega$ .
4. **MIC** – Microphone Input, the microphone input transfers its signals to the on-chip preamplifier.
5. **REPLAY** – loop play the record.
6. **FT** – Feed Through: This mode allows use of the speaker drivers for external signals.
7. **ISD1820** – IC chip
8. **Lead Out IO** – VCC LED NC FT GND / VCC REC PLAYE PLAYL GND
9. **P2** – default short connection ROSC to 100k $\Omega$  resistance, that's means record duration is 10s
10. **PLAYL** – Playback, Level-activated, when this input pin level transits for LOW to HIGH, a playback cycle is initiated.

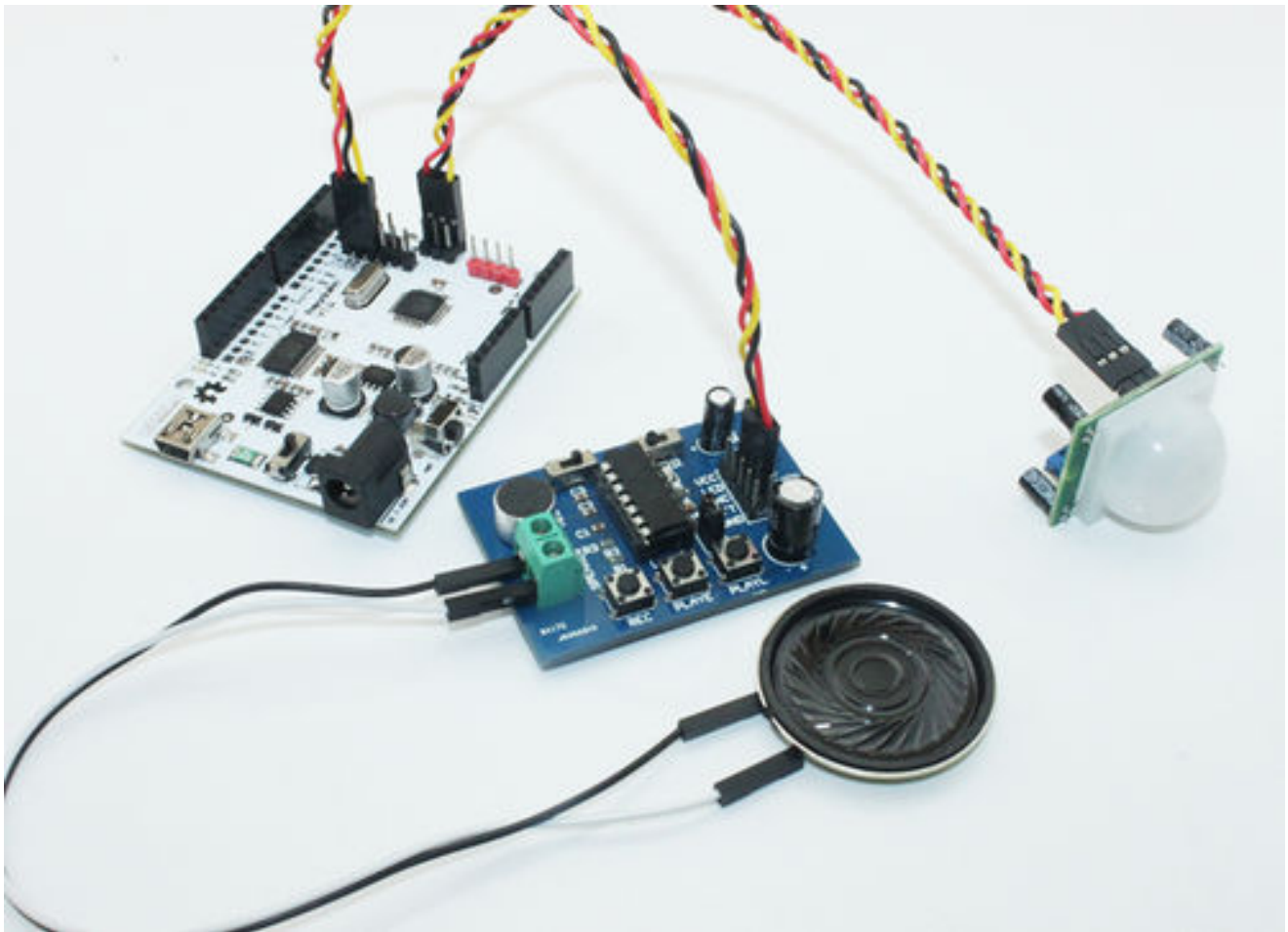
More detail of IO define please reference our [WIKI](#).

## Record Operate Guide

1. Push REC button then the RECLEL(D1) will light and keep push until record end.
2. Release the REC button
3. Select Playback mode: **PLAYE**, just need push one time, and will playback all of the record or power down ; **PLAYL**, you need always push this button until you want to stop playback record or end ; **REPEAT**, switch 5 to right side, and the record will playback time a time until switch to light or power down
4. FT mode, when you switch 6 to right side, that means all of you speak to MIC will direct playback from Speaker.

## Example

We can build a Auto PIR Alarm small project. We have record a short alert tone in Voice record module. Then we control **PIR Sensor** and Voice record module by **Freaduino ATMage328**. Of course you can also use PIR and record function. Use your imagination.



Programming Demo code like :

```
/
*****
*****
** Device: ISD1820
**
** File: EF_Alert.c
**
**
**
** Created by ElecFreaks Robi.W /14 Nov 2011
**
**
**
** Description:
**
** This file is alert demo code base on PIR sensor and ISD1820
**
** Voice record module, just a sample code for your reference.
```

```
    **
**
    **
** Copyright (C) 2011 ElecFreaks Corp.
**
** http://www.electfreaks.com
**
*****
*****/

#define SOUT 12
#define REC 8
#define PLAYE 9
#define PLAYL 10

void setup(){
  pinMode(SOUT, INPUT);
  Serial.begin(9600);
}

void loop(){
  int ret = digitalRead(SOUT);
  if(ret == 1)
  {
    Serial.println("-----> PIR");
    digitalWrite(PLAYE, 1);
    delay(5000);
    digitalWrite(PLAYE, 0);
  }
}
```