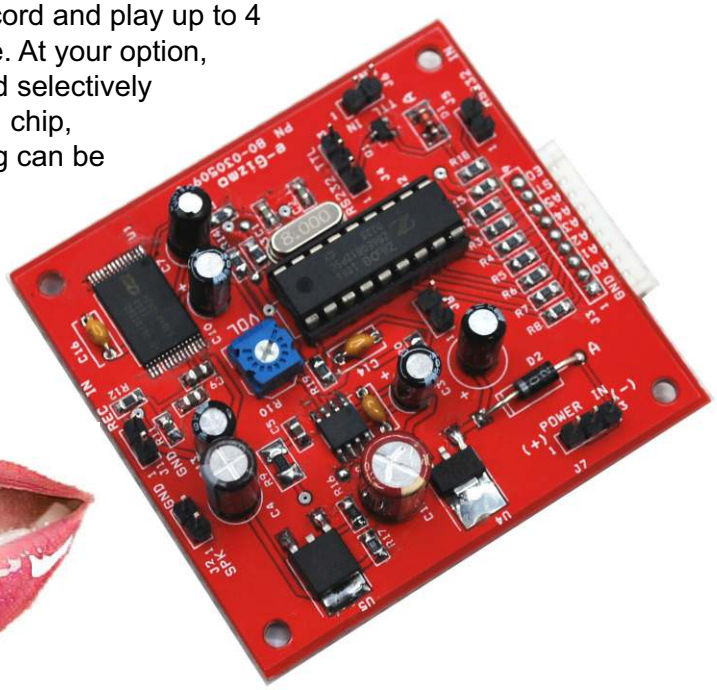
VoiceKit

Sound Recorder kit

Technical Manual Rev 1R0

Talking machines are cool, but even more important, voice makes your machines and circuits more appealing and friendly to its human users. Adding a voice to your machines is made easy with the use of e-Gizmo VoiceKit. The VoiceKit is a tape-less voice and sound recorder module that can record and play up to 4 minutes duration of sounds, any sound you can imagine. At your option, you can record and play sound into short segments, and selectively play them later at any sequence. With a 4 minute sound chip, addressable segments as short 3 seconds per recording can be made, making up to 64 selectable sound segments.



Give your machines and circuits a voice!

You can operate the VoiceKit two ways. One is by sending commands through its serial port, and the other through a 8-9bit wide control port. Two serial ports are provided. One works directly at TTL level, a convenient option when you want to control the module using a microcontroller. The other works at RS-232C level. This port interfaces readily with any RS-232C equipped equipment, from a home PC, to an industrial PLC. The control port allows you to control the VoiceKit using an array of switches, making stand alone operation possible. Of course, you can also control the VoiceKit with a microcontroller using the control port.

General Features

Voice Chip: ISD4002
- Duration: 240sec, 4Ks/S Standard
- Option A: 120sec, 8Ks/S (High sound quality)
- Option B: 180sec, 6Ks/S (Medium sound quality)

Controller:
- Zilog Z86E02

Power Supply:
- 9-12VDC

Audio Output:
- 250mW @ 9V, 16 ohms speaker
- 500mW @ 9V, 8 ohms speaker

Control I/F :
- Serial TTL
- Serial RS-232C
- 8/9 bit wide Control Port (TTL)

Dimensions:
- 59 x 66 x 18mm (LWH)

CONNECTORS AND CONTROLS

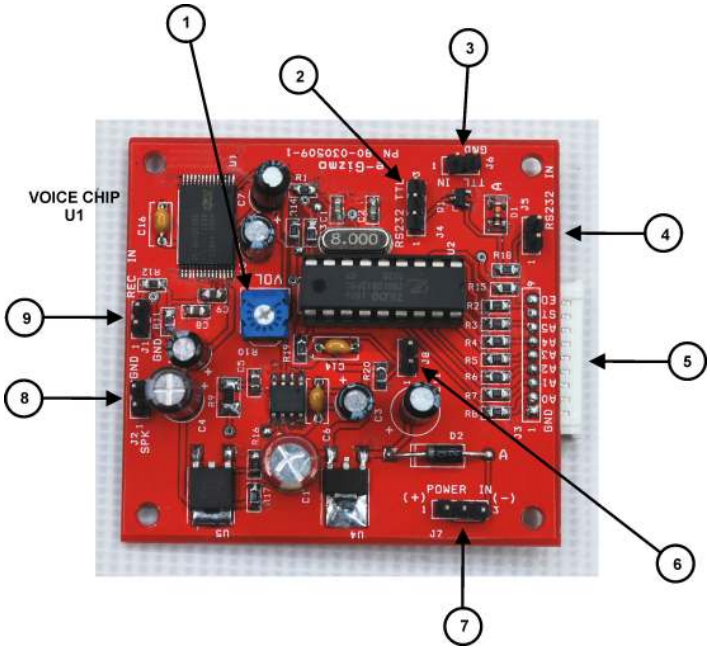


Figure 1. The VoiceKit connectors and controls are identified in this photo and described in the text.

1. Volume Control

2. J4 - Serial Interface Level jumper selector

Jumper Position	Active Input
1-2	J5 - RS-232C Level
2-3	J6 - TTL Level

3. J6 - Serial Input TTL Level.

Pin No.	Signal
1	TTL Serial Input
2	GND

4. J5 - Serial Input RS-232C Level.

Pin No.	Signal
1	GND
2	RS-232C Rx Input

5. J3 - 8 bit control input TTL Level

Pin No:	ID	Description
1	GND	Common GND
2	A0	Address 0 input
3	A1	Address 1 input
4	A2	Address 2 input
5	A3	Address 3 input
6	A4	Address 4 input
7	A5	Address 5 input
8	$\overline{ST}$	Start input
9	$\overline{EOM}$	End Of Message output

6. J8 - Record Control

Pin No.	Signal
1	Record Control input, active low
2	GND

7. J7 - Power Input

Pin No.	Signal
1	+9V to 12VDC Power Input
2	No connection
3	GND

8. J2 - Speaker Connector

Pin No.	Signal
1	Speaker (+), 8-16 ohms
2	Speaker GND

9. J1 - Record Input

Pin No.	Signal
1	GND
2	Sound Record Input

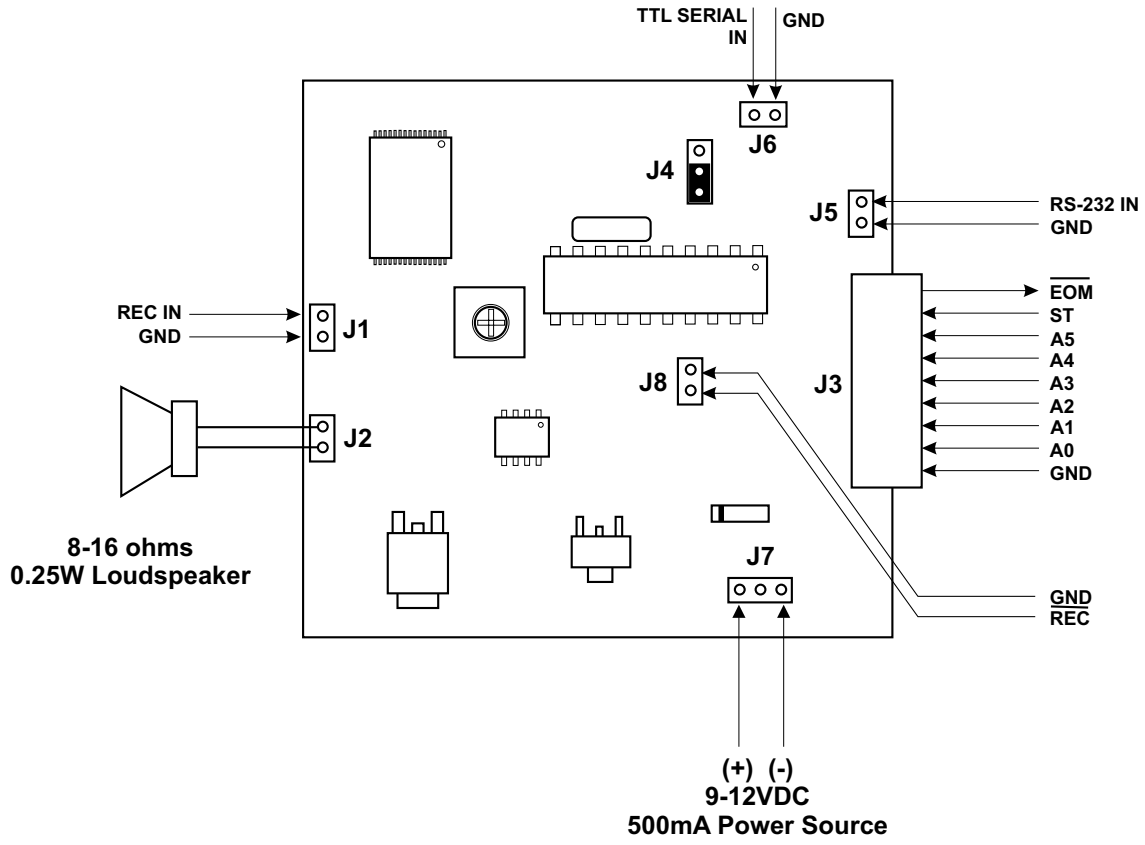


Figure 2. Simplified wiring diagram of the voice kit. Not all connections are used in a typical applications. Read text for more details.

APPLICATION NOTES

As previously mentioned, the VoiceKit can be controlled two ways- using serial port or control port. These will be discussed in greater details in this section.

Figure 2 illustrates the general wiring configuration of the VoiceKit. Not all wirings shown in the diagram will be used in an actual application. If you decide to control the VoiceKit using a serial port (this is the preferred scheme), you only need to wire either J5 or J6, wiring is not needed for control port J3 and J8. If you opt to use the control port instead, then J5 and J6 wirings are not needed.

Needless to say, whatever control scheme you decide to use, you need to have the power source J7 and loudspeaker J2 wired.

VOICEKIT SERIAL CONTROL

VoiceKit control through a serial port is the preferred method because of the following benefits:

1. Only two wires is needed to control the VoiceKit. This minimal I/O requirement is a huge advantage if you are using a low pin count MCU to operate the VoiceKit.

2. Recording and playing is simpler and easier.

3. Program codes is shorter and easier.

Communications Format

The VoiceKit will link with the host controller under the following serial communications settings:

Baud rate: 9600 baud
Data : 8 bit
Parity: None
Stop Bit: 1

Control is implemented by sending a command in three byte format:

[STX]+[Command]+[ETX]

Where:

STX=2 Start of transmission
ETX=3 End of transmission

Command byte bit definition:

D7	D6	D5	D4	D3	D2	D1	D0
V1	V0	A5	A4	A3	A2	A1	A0

where:

A0..A5 : Record/Play start address

V1-V0 : Command Bits

00 - Undefined
01 - Stop Record/Play
10 - Play from indicated address
11 - Record from indicated address

Example:

Objective: Play starting at address 12.

1.Build the command byte

a. Address= 12 or 001100 in binary

x	x	0	0	1	1	0	0

b. Play command = 10 (binary) on bit 7 and 6

1	0	0	0	1	1	0	0

2. Send the command to the Voice Kit in the following order

a. Send STX (00000010 in binary, 0x02 in hex)
b. Send Command byte (10001100b in binary, 0x8C in hex)
c.Send ETX (00000011 in binary, 0x03 in hex)

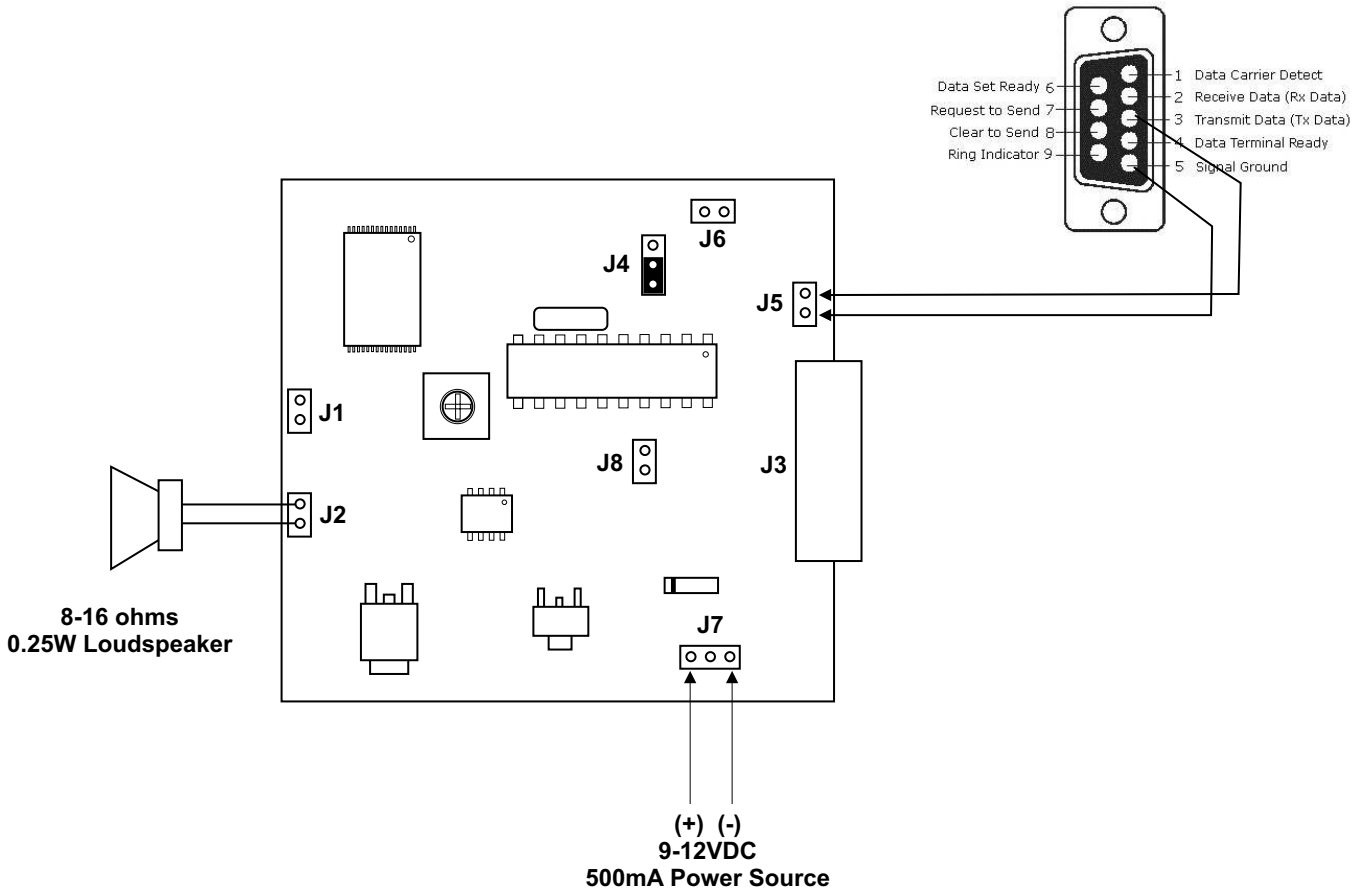


Figure 3. Example conection setup using RS-232 serial control input. A jumper must be installed across pin 1-2 of J4.

Programming a Windows PC with Visual Basic, this sequence is send to the VoiceKit through the MSComm object as follows:

```
MSComm1.Output=chr(&H02) + chr(&H8C) +
chr(&H03)
```

VoiceKit Control Port

The control port J3 provides another convenient way of operating the VoiceKit. Playing a pre-recorded sound is straightforward:

1. Setup address lines A0-A5.
2. Momentarily pull the ST control input to logic zero.

The selected message will play immediately (actually, there is about 100ms delay) and the end of message EOM output will turn high (logic 1).

Playing will stop when the end of message EOM is reached. The VoiceKit EOM output will switch to logic 0 to indicate and end of message. You can stop play and play a new recordings at any time by repeating 1 and 2.

Manual recording is done in a slightly different way. With the sound source ready at REC IN input J1:

1. Setup address lines A0-A5.
2. Play the sound source and pull REC control input low. Recording continues as long as REC control is held low.
3. Release the REC control input to logic high at anytime to stop recording.

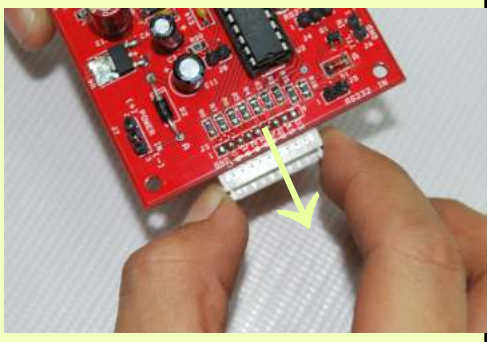
As you would have probably notice, ST control input is not used during recordings. REC control is level sensitive, driving it low will start recording and will stop when the REC control is released high or the recording memory is full.

Recordings exceeding a segment time will automatically record over the next segment address. Let me elaborate further for you to get a clearer view. If your VoiceKit is using the 240sec recorder chip, each address segment can store up to 240/64 = 3.75 seconds worth of recording. If you start recording at address 5 and the recorded sound took less than 3.5 sec (allowing 0.1s start and stop delay), it will fill just one record segment, with address equal to 5. If the recording lasts, say 15 seconds, then it will fill 3 succeeding segments, with addresses 5,6, and 7. Next available free segment will then be address 8. To play the 15 seconds recordings, send play command with address set at 5. If you set the address equal to 6 and then initiate a play, play will skip the first 3.75 seconds and start playing midway the stored recordings.

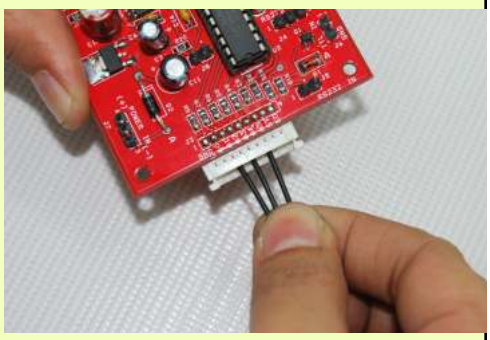
J3 SNAP ON CONNECTOR

J3 is a snap ON connector that will accept AWG 25-26 wires without the need for a mating connector.

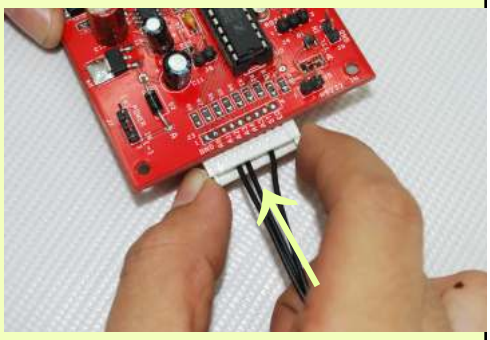
1. Pull cover in the direction of the arrow.



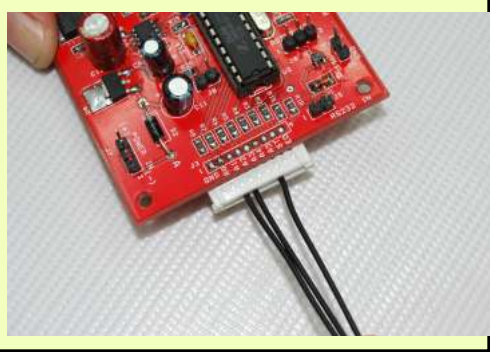
2. Insert pre-stripped wires inside desired slots.



3. Push the cover all the way back.



4. Check to ensure wires are securely clamped in place.



Record source and Using your PC to record sounds

The VoiceKit was designed to accept recording from audio sources with large signal output. Examples are headset player (MP3, Ipod, etc) and PCs. VoiceKit can handle input signal swings as high as without overloading distortion. Each audio devices will have different output levels with according to their volume control settings. As such, you may have to experiment first with each devices at different volume settings, choosing the settings that gives the best play and loudness quality.

A PC would make and excellent recording source (through the soundcard speaker output, see Figure 4) for the following reasons:

- Using standard Windows recorder utility program, sounds can be enhanced, edited, showered with effects, before transferring it to the VoiceKit.
- Multitudes of free and ready to play sound clips and sound effects are available online.
- Recording process can be automated.

In fact, included in this kit is a ready to use Visual Basic program source code listing that automates the recording process using a PC. Since the source code itself is provided, you can modify the program as you please to add more features or make it more adapted to your needs. Discussion and usage of this program is covered in greater detail in a separate manual included with the VoiceKit.

SCHEMATIC DIAGRAM

The complete schematic diagram of the VoiceKit is shown in figure 5. Central to the operation of this circuit is U1 ISD4002; the sound recorder chip. This chip comes in three variants separated by their recording duration capacities: 120,180, and 240 secs capacity. Actually, these variants have the same storage capacity. Differences in the recording time are due to their different sampling speed. It's a choice between recording time and sound recordings quality. The 120sec chip uses a sampling rate twice that of the

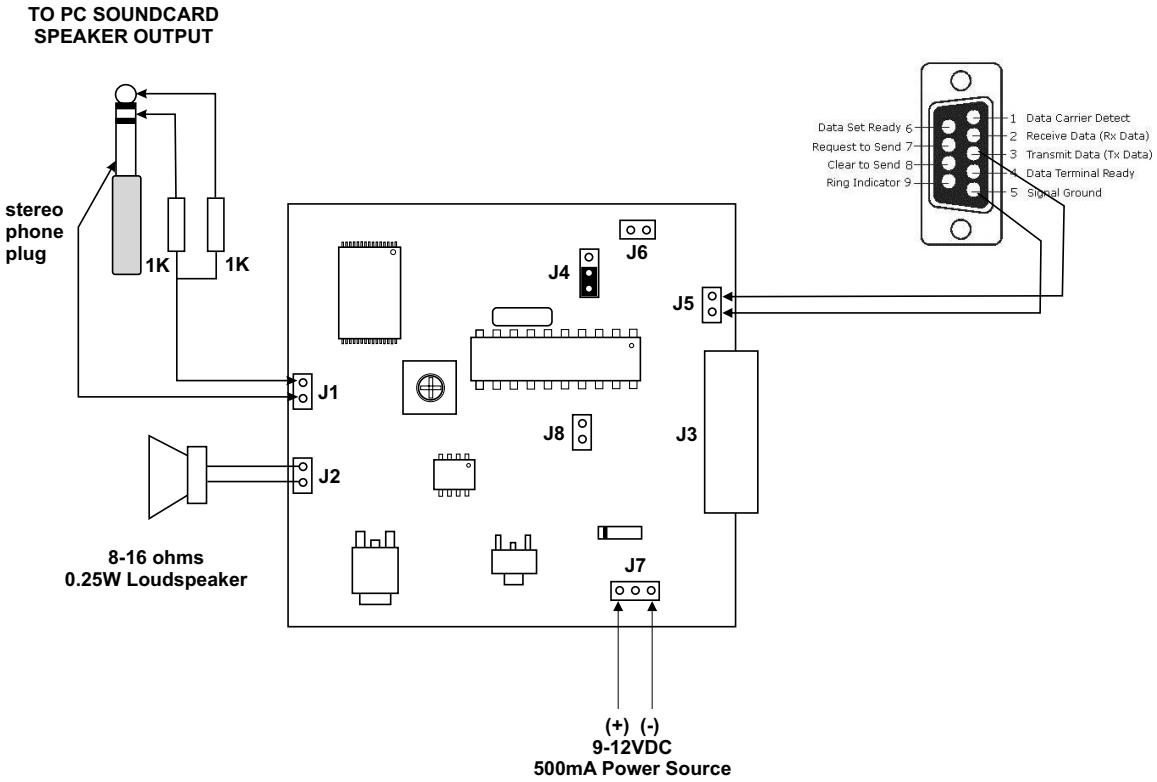


Figure 4. Automatic recording connection setup using a PC. See text for more details.

240sec chip, hence produces better sound quality and a wider bandwidth.

U2 Z86E02 is a programmed microcontroller taking in some complex task required by U1 to operate. With U2, using the voice chip becomes as simple as already described in the preceding sections of this manual. In fact, stand alone operation even becomes possible with the use of the control port J3.

Q1 translate RS-232 input levels to TTL level prior feeding the signal to U2 for processing. U3 is an audio amplifier that boosts the audio output from U1 to an audible level. U4 provides the 5V regulated supply needed by U2, whereas U5 scale this voltage down further to 3.3V to supply U1.

BILL OF MATERIALS

Electronics Components Only

ITEM	QTY	COMPONENT ID	DESCRIPTION
1	2	C2,C1	18pF ceramic SMD 0805
2	2	C10,C3	10u/16V electrolytic
3	1	C4	220u/10V electrolytic
4	1	C5	39nF ceramic SMD 0805
5	3	C6,C14,C16	0u1/16V Multilayer
6	1	C7	1uF electrolytic
7	2	C8,C9	0.47uF ceramic SMD 0805
8	1	C11	100uF/16V electrolytic
9	1	C12	100uF/10V electrolytic
10	1	C13	220uF/10V electrolytic
11	1	D1	1N4148 SMD
12	1	D2	1N4001
13	1	J1	1x2 Header REC IN
14	1	J2	1x2 Header Speaker
15	1	J3	1x2 Header CONTROL IN
16	1	J4	2x2 Header RS232 TTL
17	1	J5	1x2 Header RS-232C RX In
18	1	J6	1x2 Header TTL Serial
19	1	J7	1x2 Header REC IN
20	1	J8	1x2 Header REC Control
21	1	Q1	2SC3928A NPN Transistor SOT-23
22	13	R1,R2,R3,R4,R5,R6,R7,R8,	3k chip resistor 0805
		R13,R14,R18,R19,R20	
23	1	R9	10 chip resistor 1206
24	2	R11,R10	1K chip resistor 0805
25	2	R15,R12	10K chip resistor 0805
26	1	R16	150 chip resistor 0805
27	1	R17	330 chip resistor 0805
28	1	U1	isd4002 - 240/180/120 Voice Recorder IC TSSOP-28
29	1	U2	Z86E02 Programmed MCU DIP-18
30	1	U3	LM386 Audio Amplifier SOIC-8
31	1	U4	LM2937/TO263 5V LDO regulator
32	1	U5	LM1117DT-1.8 LDO VR

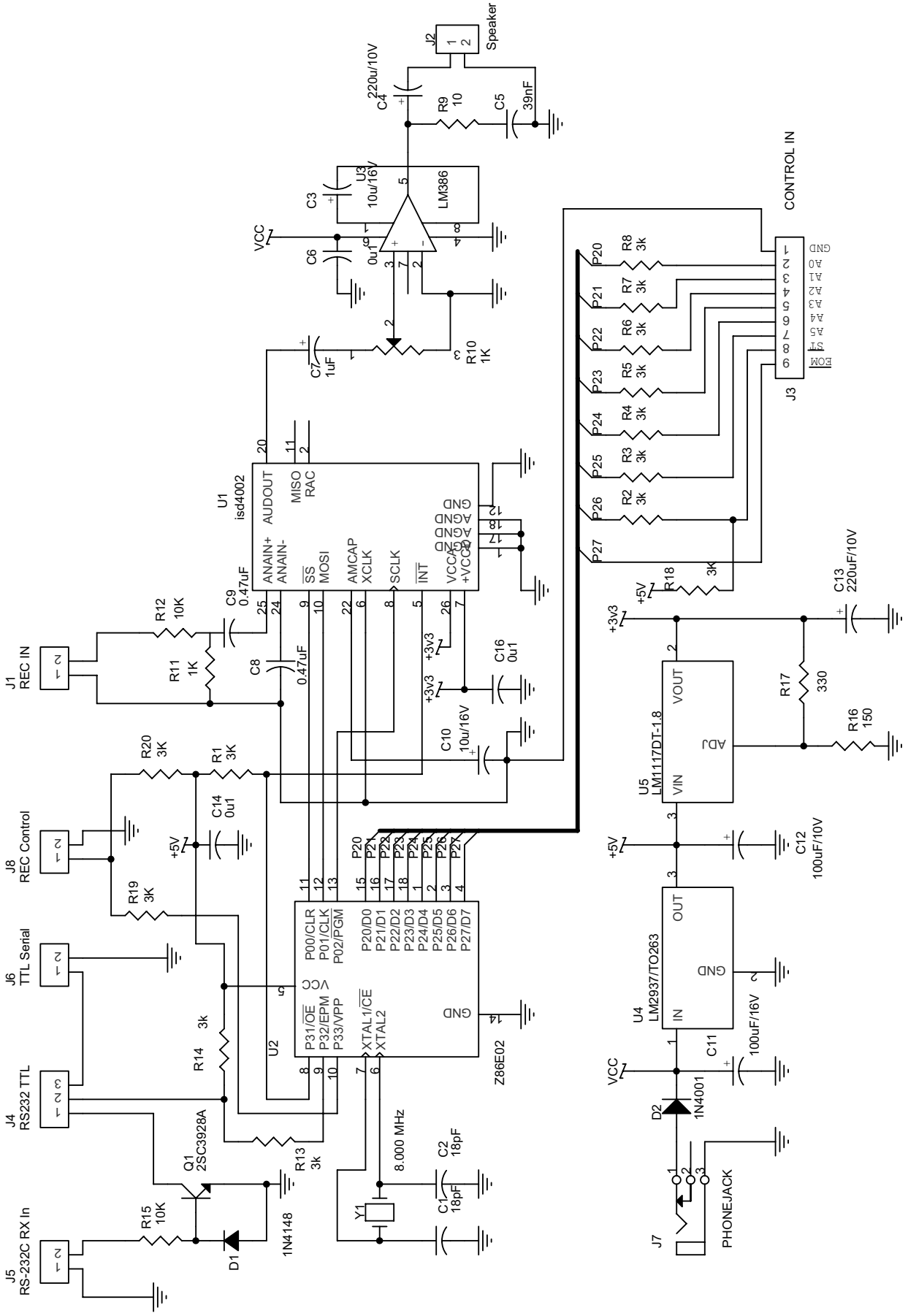


Figure 5. Complete schematic diagram of the VoiceKit.