HALLOWEEN CREEP-O-TRON

Pumpkin with light effects and sound

check it out on: www.elektor.tv

Kit available from STORE!

Halloween is a festival that is now also increasingly celebrated in Europe. A hollowed-out pumpkin with a candle in front of the house, children who will come to 'trick-or-treat' and beg for candy… This amusing circuit hopefully contributes to the goose-bumps atmosphere by making such a hollowed-out pumpkin interactive, with both sound and light effects!

Design: **Niek Laskarzewski**, text: **Harry Baggen** (Elektor.Labs)

Halloween is used to denote the All Saints' Eve (October 31) celebration. In many English-speaking countries this is a public holiday. It is an old custom that children will dress up in costumes on this day and visit the houses in their neighborhood which have been decorated with (hollowed-out) pumpkins and lights. The children will ring the doorbell with the objective of frightening the occupants, after which it is the intention that they will be given something nice to eat (candy or — healthier — fruit). These days, teenagers and adults also organize Halloween festivals (including drinks and fancy dress parties).

In the Netherlands and Belgium there is a similar tradition, the Saint-Martin feast, which is celebrated on 11 November. In this event children will wander about with lanterns. Nevertheless, here too there are now an increasing number of people who have discovered Halloween and will have a go with pumpkins and quasi-scary costumes. It is exactly that 'scary' element which makes it that much more exciting for children.

With an eye on the approaching Halloween date, the Elektor lab had the enlightened idea to develop an amusing circuit for pumpkins, which produces both light and sound effects. The result is impressive, particularly if the circuit is housed in a real, hollowed-out pumpkin. During testing, several of our colleagues at Elektor Castle (mainly the women) appeared to be briefly frightened when they approached the pumpkin and were taken by surprise by the flashing lights and a terrifying cry or scream from the pumpkin.
To realize all this, we have designed a small circuit, which is provided with a proximity sensor, a number of LEDs, a sound module, a small audio amplifier and a loudspeaker. All this is under the control of an AVR microcontroller. And this project is suitable not just for Halloween! With a few Christmas carols on the SD card of the sound module this is also a very nice gadget for the upcoming Christmas festivities. And using the kit that we have

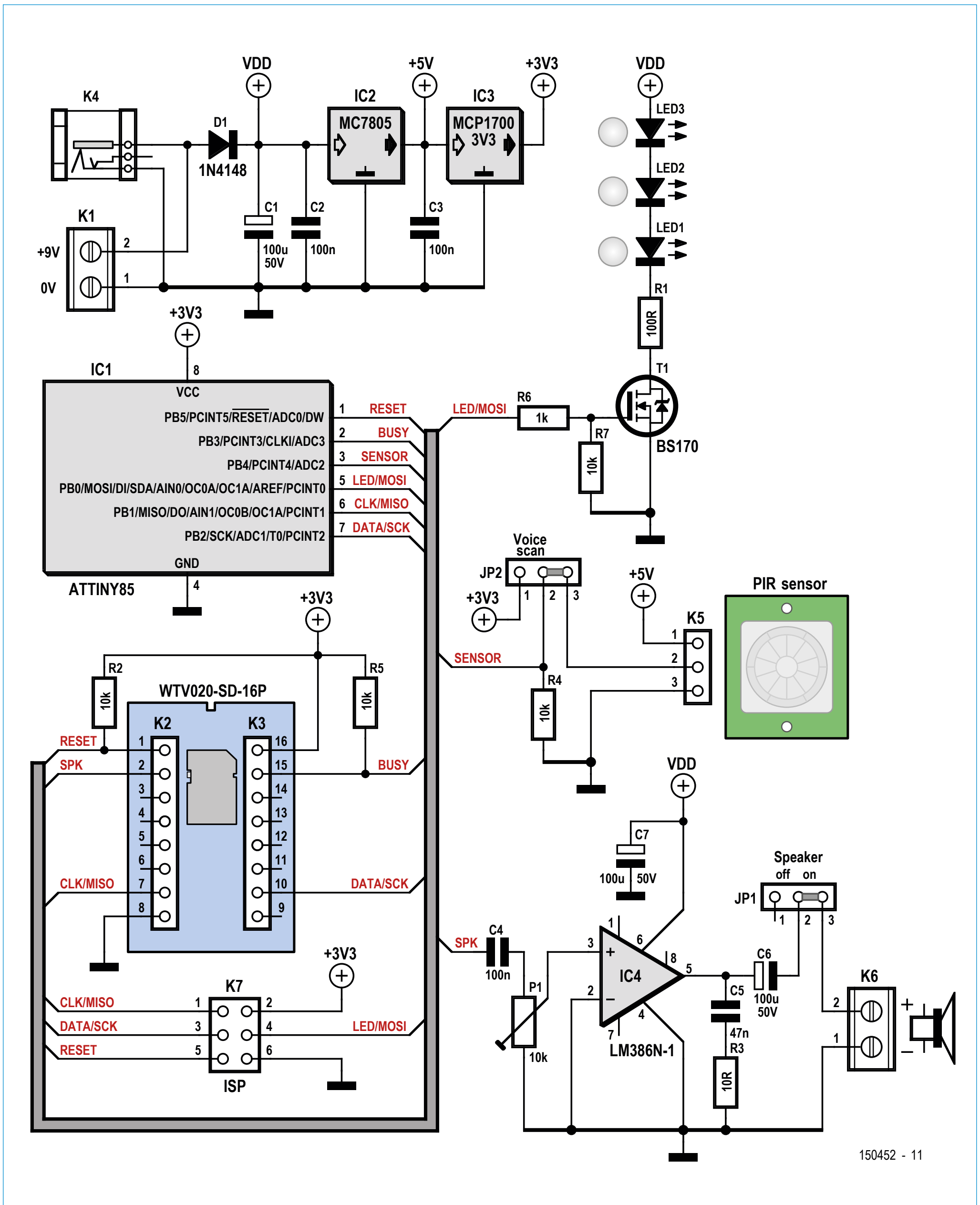


Figure 1.
This circuit produces sound and light effects when someone approaches.

put together for this project, it can all be built very quickly.

Circuit

The entire circuit can be seen in **Figure 1**. There are only a few components, because of the use of a ready-made sound module and a module for the complete proximity detector (PIR sensor). To control everything, a small ATtiny85 was selected, since in this application only a few I/O lines are required. The controller can be programmed via ISP connector K7. Three large LEDs with a diameter of 10 mm, LED1 through LED3, provide the light effects. These are driven from the controller (pin 5) via FET T1.

The small loudspeaker is driven by an LM386 (IC4), which is connected in the typical way as indicated in its datasheet. This amplifies the input signal 20 times. The volume can be adjusted with P1. The loudspeaker can be connected or disconnected using header JP1. The circuit requires two power supply voltages: 3.3 V for the controller and the sound module, and 5 V for the PIR sensor. The LEDs and the small audio amplifier are powered directly from the line adapter or batteries (8 to 12 V). Because of the requirement for two power supply voltages, there are two voltage regulators connected in series, IC2 and IC3. D1 serves as reverse-polarity protection.

This concludes the bird's-eye view of the circuit. We will now look a little closer at the PIR sensor and the sound module (**Figure 2**).

Sound module

The sound module that we chose for this project is the type WTV020-SD-16P, which is provided with a micro-SD card slot for the storage of sound files. The module has 16 pins and fits on two 8-way headers on the circuit board. On one side there is a small notch in the circuit board, this indicates the location of pin 1.

The maximum power supply voltage for this module amounts to only 3.5 V, that is why we chose the 3.3 V. The module can play a maximum of 512 different sound files, which are stored on a micro-SD-card that can have a maximum capacity of 1 GB (according to the manufacturer; in practice 2 GB worked just as well).

The sound module can only operate with so-called ad4 files, a relatively exotic sound format that is used very little. You will therefore have to convert your sound files before you can use them. Stereo sounds have to be converted to mono first (for example using the program Audacity) and compress the sound levels. You will then have to export this as a 16-bit WAV-file and subsequently convert it to the ad4-format with the help of a conversion program that was made for this purpose by an Australian company [1]. This program can handle MP3- and WAV-files.

When the files are copied to the SD card, you will have to pay attention to the correct numbering. The files have to be named XXXX.ad4 and have to be numbered consecutively, starting with 0000.ad4. A maximum of 512 files can be used.

To play back a file, a 16-bit data word containing the number of the file has to be sent to the data pin (pin 10), with the corresponding clock pulses on pin 7 of the module. While the file is being played back the module sets

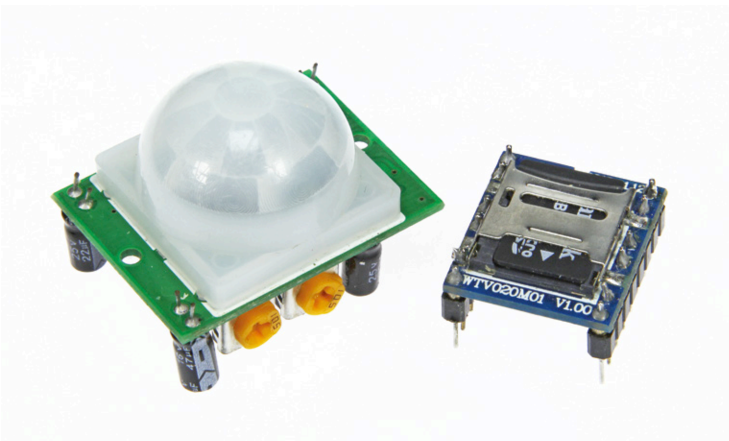


Figure 2.
The sound module and IR sensor module used in the project.

the Busy pin (15) high.
Other useful commands that can be sent to the sound module are (in hex format):

- 0xFFF0...0xFFF7 = volume adjustment (0xFFF0 = minimum)
- 0xFFFE = pause/play
- 0xFFFF = stop/play back

Unfortunately there is no command to query the number of files on the SD card.
When a file is playing, the audio signal appears on connections SPK+, SPK– (PWM signal) and audio-I (DAC output). A loudspeaker can be connected directly to SPK+/SPK–, but for a bit more power it is better if you connect a separate amplifier stage, as we have done here. The latter will then obtain its signal from the connection audio-I (pin 2).
Since there is no command to establish how many files there are stored on the micro-SD-card, the firmware contains a routine which counts them after initialization. For this purpose the jumper on JP2 has to be set to 'Voice scan'. After the power supply voltage is turned on, the firmware will enter the scan mode and, using a time measurement, detect the presence of each file. Starting from 0000, and increasing consecutively, all data values are sent to the module. The busy-pin of the module goes high while the playback of a file is in progress. If there is no file for a particular number then the busy-pin will go low again after about 1.8 s.
By testing the state of the busy-pin 2 s after sending the file number you can therefore check whether the corresponding file exits on the memory card. For this procedure to work properly it is necessary that the files on the card have a minimal length of 2.5 s. During the scan it is recommended to temporarily disconnect the loudspeaker using jumper LP1.

PIR-sensor

The PIR-sensor module HC-SR501 is a circuit board which contains a complete IR-detection circuit, with a PIR-sensor and corresponding plastic lens on the front. This module is connected to connector K5. The module is powered from a power supply voltage of 5 V by voltage regulator IC2. After powering-on, the PIR-module requires about 1 minute to initialize itself. During this time the output generates 4 pulses, one at the beginning and three at the end. There are two trimpots on the circuit board, which are used to set the sensitivity and the duration of the output pulse. In this application only the sensitivity (xS, in the middle of the board) is of interest.

Firmware

The firmware has been written using the Atmel Studio design suite. The program begins by configuring the outputs: clockPin, dataPin and resetPin. The inputs are busyPin and sensorPin (see source code [2] and the flow diagram at the end of this article, **Figure 4**).
After the I/O-ports have been initialized, the firmware resets the SD-module by simultaneously setting both clockPin and resetPin high and then applying a low pulse of about 5 ms to resetPin. This is then followed by a delay time of 300 ms before the program continues. Subsequently, the microcontroller reads the stored value for the number of sound files on the SD-card from its internal EERPOM and places it in the variable 'voiceAmount'. If this value is equal to 0 (no files) or greater than 512 (maximum) then voiceAmount is set to the default value (21).
One all this is done, the LEDs flash once and then the firmware waits until the PIR-sensor is finished with its initialization. This can take up to a minute. During this time the sensor module generates 4 pulses, which causes the LEDs to light up, one at the beginning and 3 at the end.
If during this time it has been detected that jumper JP2 is set to 'Voice scan', then there is a 15-second delay after which the subroutine 'scanVoice' will count the number of files on the SD-card, in the manner as explained in the sound module description. The LEDs flash a few times to indicate the beginning and end of the scan.
In the scanVoice routine the variable voiceAmount is first set to 0, to reset the number of files. Then inside a loop each file is started and the level of busyPin is checked. If this is still high after 2 s then the file is present. In this case the playback is interrupted and voiceAmount is incremented by 1. When busyPin is low after 2 s then the file with that number is not present, the loop is then exited and the present value of voiceAmount is stored in the EEPROM. Once the scan is complete the LEDs continue to flash until JP2 is changed back to its normal operating position.
The scan can only be initiated during power

on, that is, during the initialization of the PIR-sensor. It is recommend to wait one minute before you return the jumper to its normal operation position if the scan took less than 1 minute, otherwise the firmware will play back a file at the end of the PIR-sensor initialization.

After the initialization of the PIR-sensor and/or the end of the file scan the firmware enters a wait loop. The instant that the PIR-sensor detects any movement, sensorPin will be set high. A random file number is then selected and that file is then played using the playVoice function.

In this playVoice subroutine, clockPin is first set low for about 2 ms to start the data transfer. Subsequently a loop is used to send a data string of 2 bytes to the sound module. At the end dataPin is set high and the subroutine returns to the main program.

The program will then wait 200 ms to give the sound module the opportunity to set busyPin high as an indication that the sound file is being played. The firmware will remain in a loop while the file is being played, that is, until busyPin is low gain. In the while-loop – and therefore during the playback – the LEDs flash with a frequency of 5 Hz.

Construction

Naturally we have designed a printed circuit board for this circuit (**Figure 3**) in order to simplify the construction, also for those who are not that experienced with electronics. Only through-hole components have been used, which are easy to insert and solder. In addition, we have also put together a complete kit of components, which contains everything you will need for a successful realization of this amusing project: circuit board, all components, programmed microcontroller, loudspeaker, PIR-module and sound module with a micro-SD-card of 1 GB. You will, however, have to supply your own hollowed-out pumpkin or some other nice enclosure which is appropriate for Halloween and is (partially)

Component List

Resistors

R1 = 100Ω
R2,R4,R5,R7 = 10kΩ
R3 = 10Ω
R6 = 1kΩ
P1 = 10kΩ preset

Capacitors

C1,C6,C7 = 100µF 50V radial, 3.5mm pitch
C2,C3,C4 = 100nF, 5mm pitch
C5 = 47nF, 0.1’’ pitch

Semiconductors

D1 = 1N4148
LED1, LED2, LED3 = LED, 10 mm, white
T1 = BS170
IC1 = ATtiny85-20PU, DIP-8, programmed, # 150452-41
IC2 = MC7805
IC3 = MCP1700-3302E/TO
IC4 = LM386N-1, DIP-8

Miscellaneous

K1,K6 = 2-way PCB screw terminal block, 0.1” pitch*
K2,K3 =8-pin pinheader receptacle, 0.1’’ pitch
K4 = DC power adaptor socket, with 2.1mm plug*
K5 = 3-pin pinheader, right angled, 0.1’’ pitch
K7 = 6-pin (2x3) pinheader, 0.1’’ pitch
JP1,JP2 = 3-pin pinheader, 0.1’’ pitch, with jumper
Loudspeaker 8 Ω/0.3W, diam. 20 mm (e.g. Kingstate KDMG20008)
HC-SR501 PIR sensor module
WTV020M01-SD-16P sound module with SD card holder
1 GB SD card
8-way IC socket for IC1 and IC4
PCB # 150452-1
or
Kit of parts including PCB, all parts, PIR sensor module, sound module and programmed controller: order code 150452-71

* Depending on the desired supply connection, fit only K1 or K4.

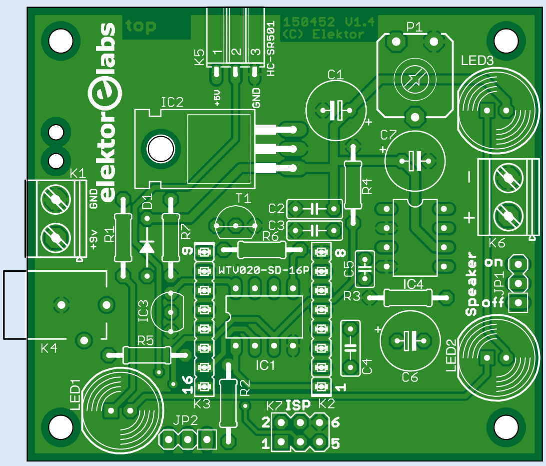


Figure 3.
Using this circuit board the project is easily assembled.

transparent.
We start with mounting all the standard components on the circuit board. K1 is not required if you fit K4, and this is therefore not included in the kit of components. The controller is fitted in a socket. You can fit the LEDs to the circuit board vertically, bending their legs at a right angle or connect them with some extension wires, depending on where you want to position them and how you want to spread the light around. The ATtiny supplied with the kit is already programmed, but if you would like to program it yourself then you can download the source- and hex-code free from the Elektor website [2].
Once you have done all this, you can connect the sound module and the PIR-sensor to connectors K2/K3 and K5 respectively. Pay attention to the correct orientation, the notch on the sound module goes on the side where pin 1 of K2 is. The PIR-module is mounted such that is will protrude from the component side of the circuit board.
The SD-card will now need to be programmed with a few sounds. Connect it with a card reader to your computer and copy the 21 sound files 0000.ad4 through 0020.ad4, which can be found in the zip-file 150452-12 [2], to the card. This can now be plugged into the sound module. You can, of course, use your own sound files, as we have already described earlier. There now remains the connection of the loudspeaker to K6 and fitting the jumpers on JP1 (on) and JP2 (right next to LED1). Set trimpot P1 to the center position and connect a line power adapter to K4 (8 to 12 V, min. 500 mA, center positive). If you copied the sound files from our website then everything is ready to go right away. After the initialization of 1 minute the circuit is ready for action. When you move in front of the sensor the LEDs will flash and a random sound file will be played. If you have copied your own sound files to the card then you will first have to carry out the file-scan as described earlier. Now you can fit the entire assembly in a pumpkin are some other kind of enclosure.

You can leave the LEDs on the circuit board, so that they will illuminate the inside of the housing, or, for example, mount the LEDs as eyes and nose of the pumpkin. Enjoy yourself!

Halloween over & done, the circuit is still usable for other occasions and parties. You can, for example, put Christmas melodies on the SD card and place the circuit in a Christmas tree or transparent ball. Also guaranteed to be a success during the Christmas holidays!

(150452)

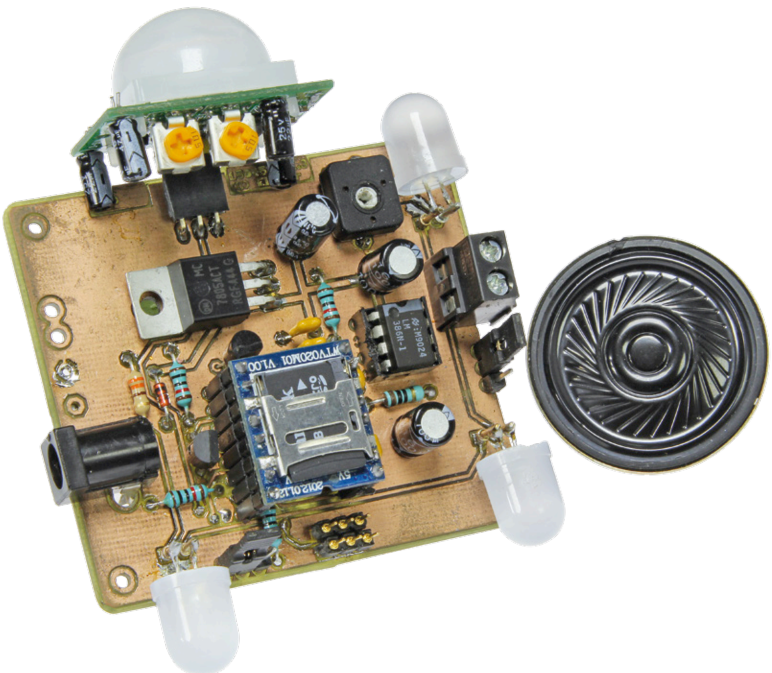
Web Links

[1] www.4dsystems.com.au/product/SOMO_14D/

[2] www.elektormagazine.com/articles

Buy the kit:

www.elektor.com/kit-scary-halloween-150452-71



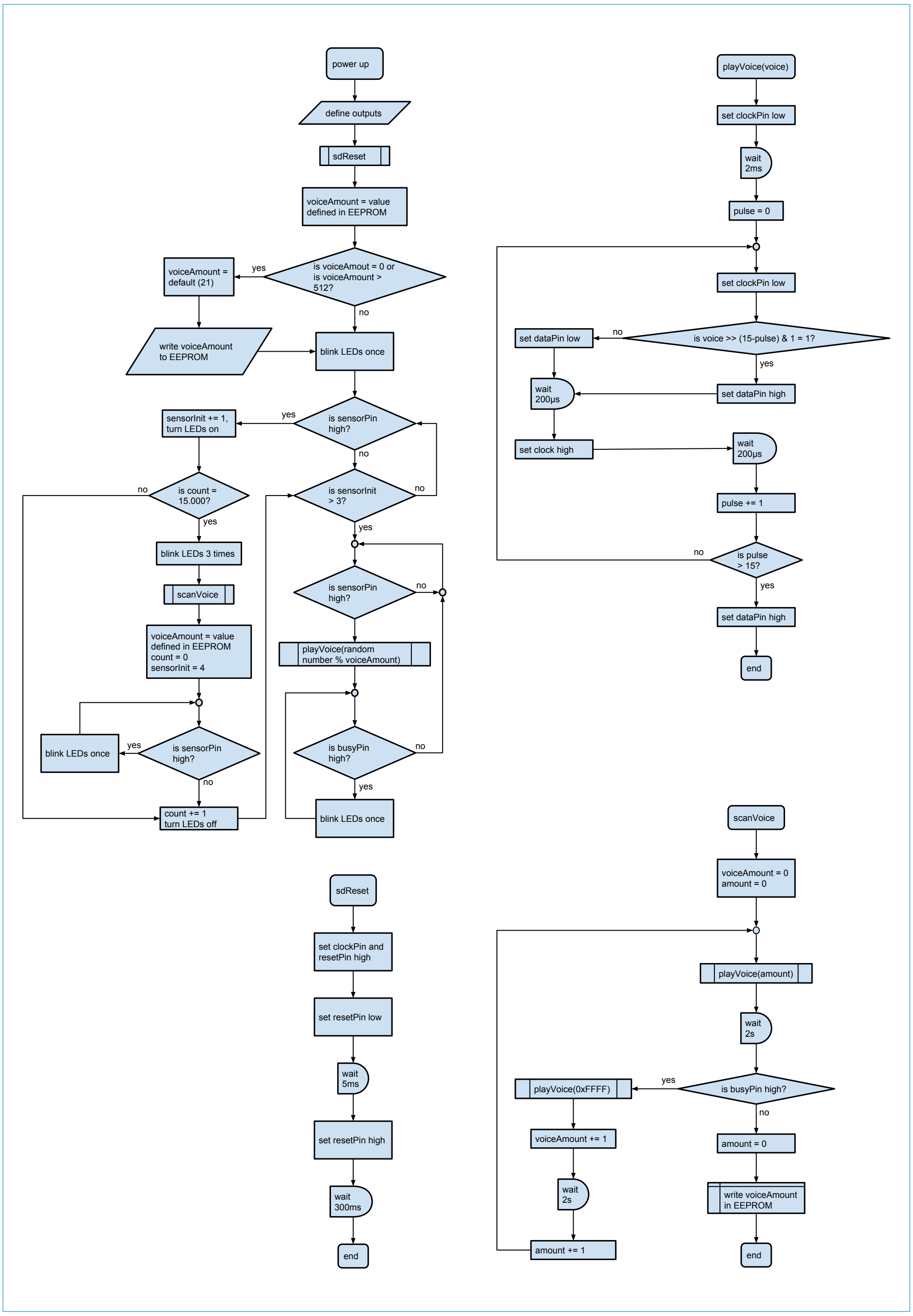


Figure 4.
Flow diagram for the firmware.