TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA
ROMANIA

MICROPROCESSORS

Motion sensor, voice recorder and player using Arduino

Coordinator:

Prof. Mircea Giurgiu, PhD

Student: Ciobu Ioana-Maria

Group 2331, eng

Year III

Introduction

For this project, the requirements were to implement a circuit which would have the purpose of detecting motion, recording the enviroment' sounds, storing the sound, then playing it back using Arduino Uno, ISD1820 Voice Recorder and an IR Sensor

To be able to make this task possible, these are the required components:

- *Arduino UNO board;*
- *ISD1820 sound sensor;*
- *IR sensor;*
- *Speaker*

The Arduino Uno board connects all the components together, while the ISD1820 records - it includes a microphone - the voice for 10 seconds when the IR sesor senses movement. The speaker should playback the sound for the user to hear.

I was able to implement the circuit on the board, thus I had the opportunity to play with real components and to understand better the concepts that I'm using and that I'm studying.

It wasn't so smooth at first because the lack of practice with real components showed, but, at last, I managed to finish the project and learn something from it.

Theoretical background

1.1 Working principle

Figure 1. Arduino Uno

A microcontroller (MCU for microcontroller unit) is a small computer on a single metal-oxide-semiconductor (MOS) integrated circuit (IC) chip. A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals. Program memory in the form of ferroelectric RAM, NOR flash or OTP ROM is also often included on chip, as well as a small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications consisting of various discrete chips.

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Nano and Leonardo.

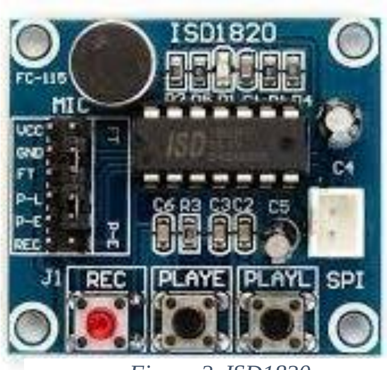


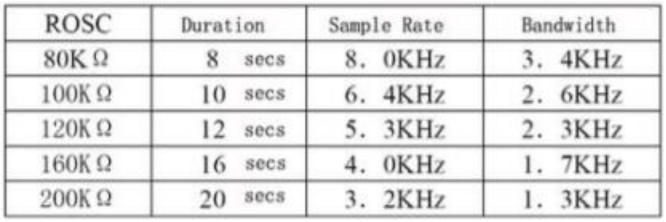
Figure 2. ISD1820

The ISD1820 Voice Recorder Module is based on the ISD1820 IC, which is a single chip Voice recorder IC for single message record and playback. A major feature of the ISD1820 Voice Recorder Module is that is can store the messages in its non-volatile memory and can be configured to store messages of length between 8 Seconds to 20 Seconds.

When the user waves across the IR Sensor, which is a component used to detect movement, the Voice Recorder starts registering the user`s voice.

Another main feature of this module is that it has internal audio amplifier that can drive a 0.5W 8Ω Speaker directly without the need for any external amplifier circuit.

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 by short cap. So the default record duration is 10s.



ROSC	Duration	Sample Rate	Bandwidth
80K Ω	8 secs	8. 0KHz	3. 4KHz
100K Ω	10 secs	6. 4KHz	2. 6KHz
120K Ω	12 secs	5. 3KHz	2. 3KHz
160K Ω	16 secs	4. 0KHz	1. 7KHz
200K Ω	20 secs	3. 2KHz	1. 3KHz

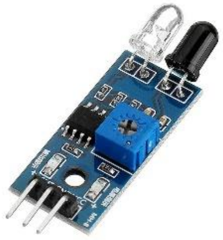


Figure 3. IR Sensor

An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment. Infrared radiation was accidentally discovered by an astronomer named William Herchel in 1800. While measuring the temperature of each color of light (separated by a prism), he noticed that the temperature just beyond the red light was highest. IR is invisible to the human eye, as its wavelength is longer than that of visible light (though it is still on the same electromagnetic spectrum). Anything that emits heat (everything that has a temperature above around five degrees Kelvin) gives off infrared radiation.

1.2 Applications

- Security Systems
- Accident voice recordings
- Record message during collisions
- Various other applications for personal entertainment

Implementation

1.3 Circuit Diagram

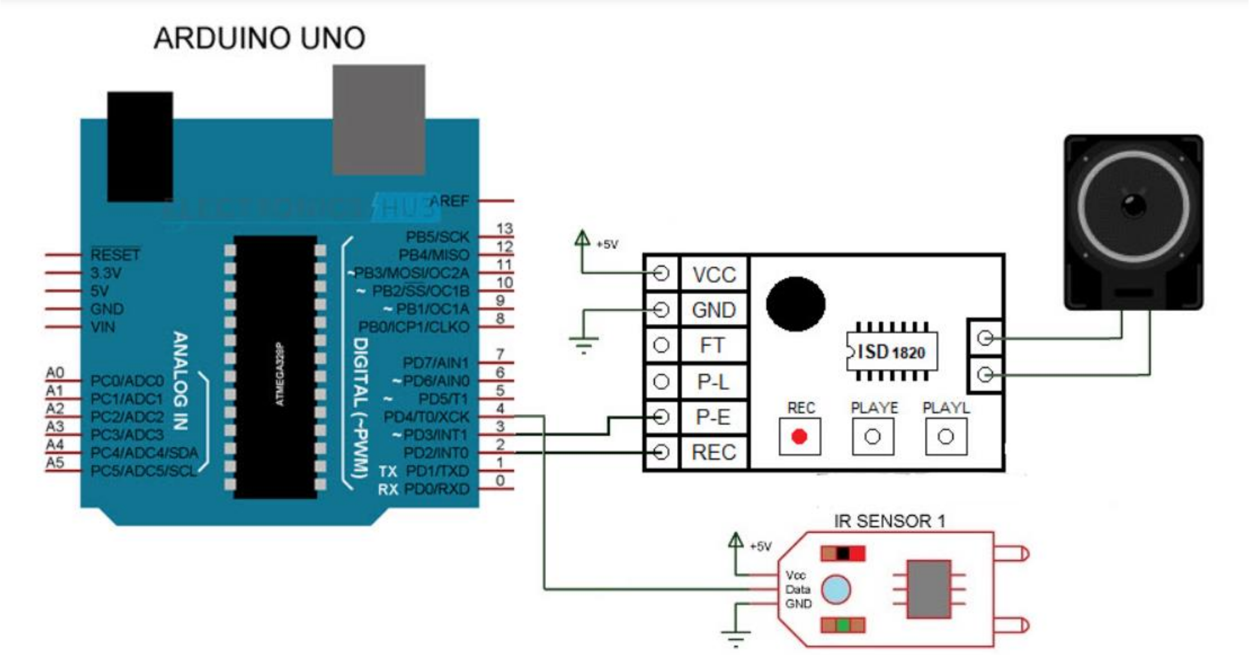


Figure 4. Circuit for voice recorder and player using Arduino UNO

1.1 Used components

Nr. crt	Component name
1.	*Arduino UNO*
2.	*ISD 1820 Voice Recorder*
3.	*IR Sensor*
4.	*Speaker*

The ISD1820 voice recorder and player would be able detect and record sounds on it own, but by using a microcontroller such as the Arduino UNO, it makes the process much easier to control and oversee. The IR Sensor is able to detect motion and transmit it to the ISD1820 voice recorder through the microcontroller.

1.2 Pin connections

- **For ISD1820:**

ISD1820 Voice Recorder	Arduino
VCC	3.3 V
GND	GND
REC	8
P-E	9
P-L	10

- **For IR sensor:**

IR Sensor	Arduino
VCC	5 V
GND	GND
OUT	11

1.3 Pin workings

- ***VCC***– 3.3V power supply
- ***GND***– Power ground
- ***REC*** – The REC input is an active-HIGH record signal. The module starts recording 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.
- ***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.
- ***PLAYL – Playback***: Level-activated, when this input pin level transits for LOW to HIGH, a playback cycle is initiated.

- ***Speaker Outputs – The SP+ and SP-*** : pins provide direct drive for loudspeakers with impedances as low as 8Ω.
- ***MIC – Microphone Input***: the microphone input transfers its signals to the on-chip preamplifier.
- **FT – Feed Through**: This mode enables the Microphone to drive the speaker directly.
- ***P-E*** – Play the records endlessly.

1.4 How to use it

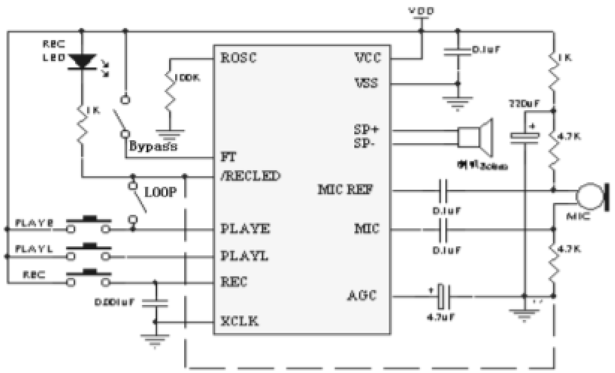


Figure 5. Schematic Diagram

When the user presses the ***REC*** button then the ***RECLED*** will light. In order to keep recording, the REC button must be pressed during the entire duration of the recording. When the user sees fit to end it, the REC button can be released.

In order to enter *PLABACK* mode, which means the sound that was recorded will be played back, the user will need to perform the following actions:

- The button ***PLAYE:*** will need to be pressed by the user only one time, and all of the records will be played until the pre-recorded sound comes to an end.
- The button ***PLAYL***: a playback cycle is initiated. It needs to be pressed continuously, until the user wants to stop the playback record or end it.
- ***P-E mode***: when short P-E jumper the record will playback repeatedly until jumper off or power down.
- ***FT mode***, when short FT jumper, that means that everything that is spoken into the *MIC* will be directly played back to *Speaker*.

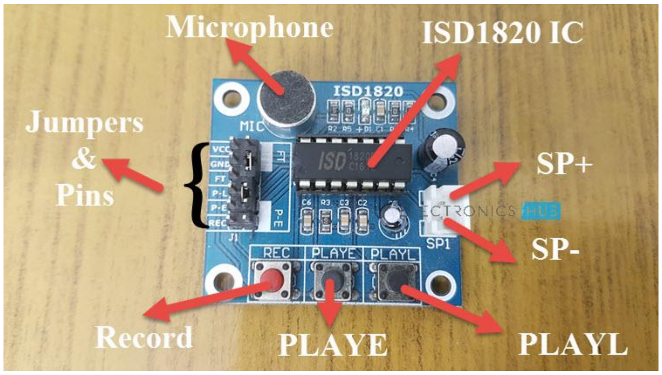


Figure 6. ISD1820-explained

When there is no object in front of the ***IR Sensor***, its output is LOW and Arduino does nothing.

When there is an object in front of the ***IR Sensor***, its output becomes HIGH and Arduino then starts recording a message by making the REC Pin HIGH for about 10 Seconds.

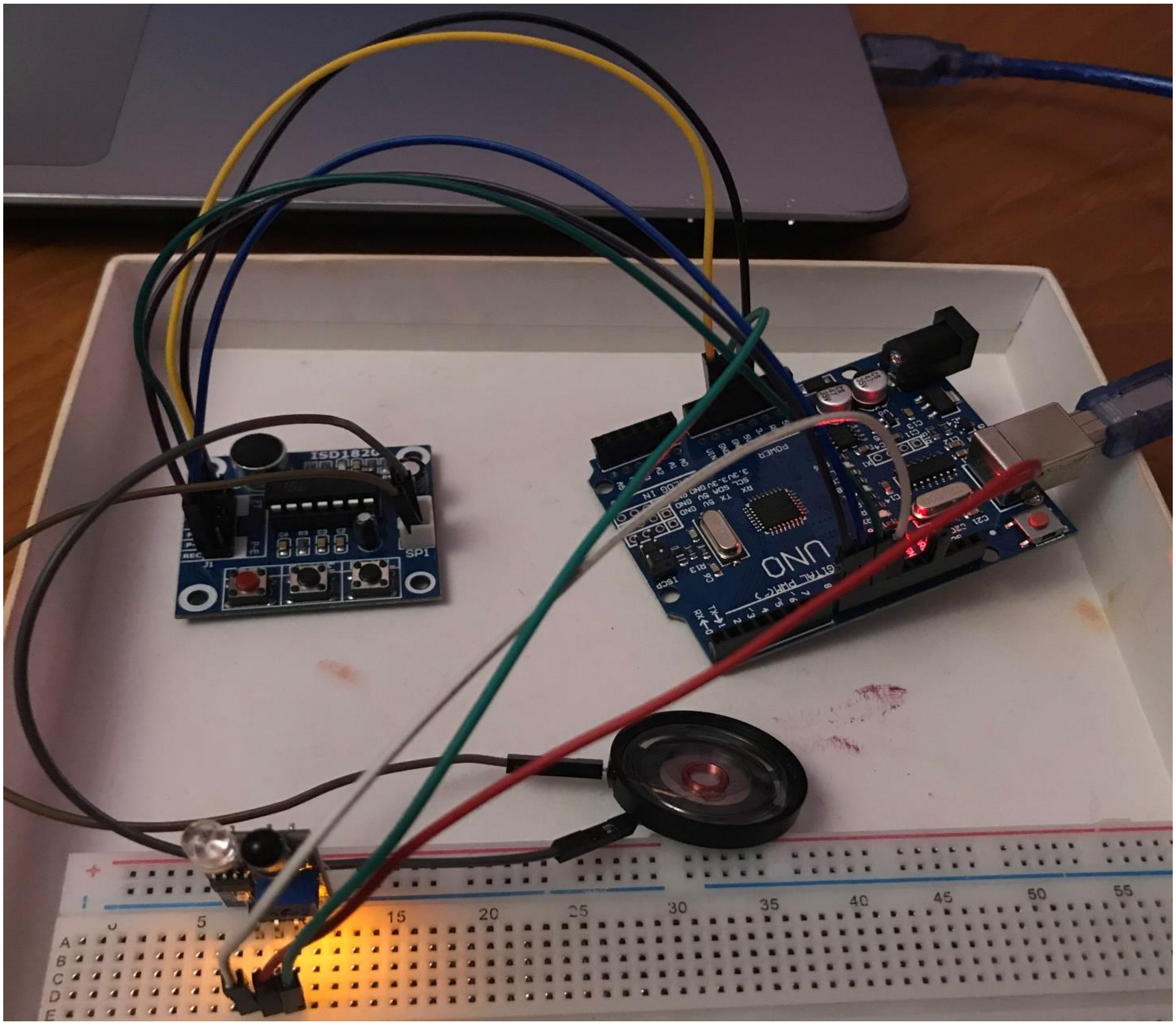


Figure 7. Circuit implementation

3.5. Circuit implementation

The SP+ and SP- pins of the ISD1820 Module are connected to the terminals of the speaker. I managed to connect them using tin. For the rest of the components I made the connections as per the circuit diagram and due to providing power to the circuit we can observe that when there is no object in front of the IR Sensor, its output is 0 and Arduino does nothing.

When there is an object in front of the IR Sensor, its output becomes 1 and Arduino then starts recording a message by making the REC Pin 1 for about 10 Seconds.

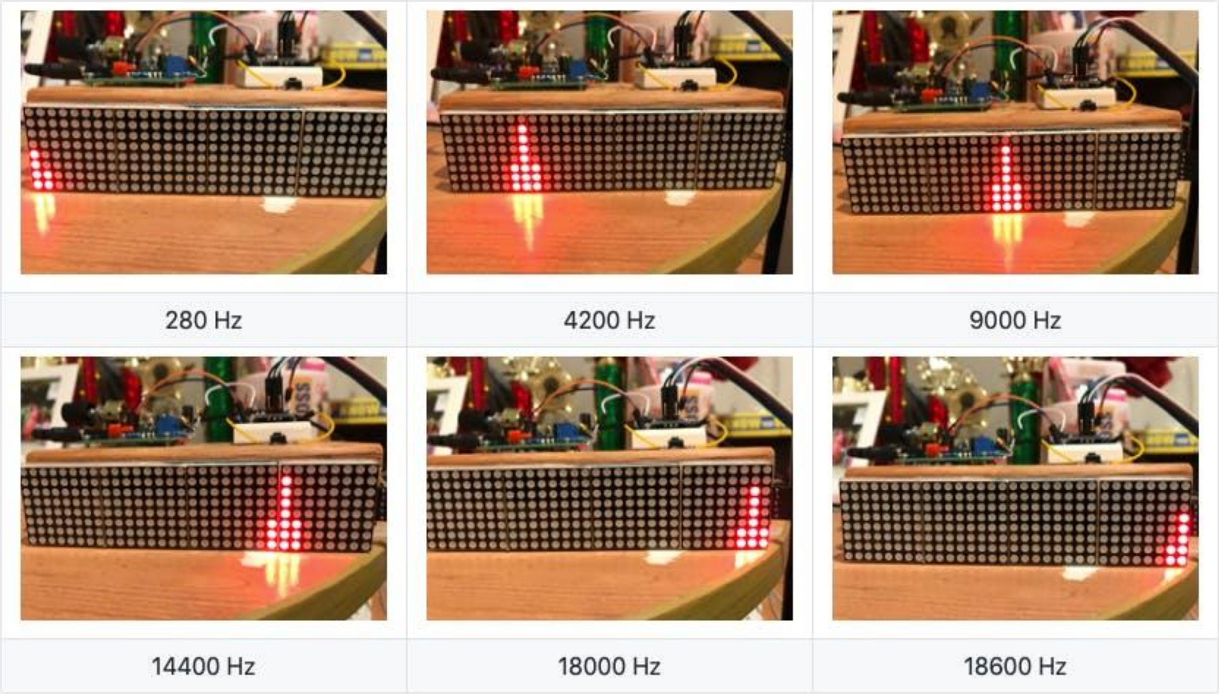
After recording the message, the message is played back by making the PLAYE pin 1 for about 10 Seconds.

3.6. Sound spectral analysis

The circuit can be modified in order to display the spectrum of the sound signal.

In order to do this, a *32x8 LED matrix* will need to be attached to the Arduino UNO board, which has and ADC pin, which is use for converting input audio signal into digital samples, so that when sound is detected the LEDs on each column of the matrix will light up according to the frequency of the sound.

By feeding audio signal to the LED matrix, it will filter out frequency bands centered around certain frequencies. The frequencies are peak detected and multiplexed to the output to provide a DC representation of the amplitude of each band. These DC values will be read by the microcontroller (in our case the Arduino board) analog input and output the spectrum to the LED Matrix so that it can be displayed on the screens



280 Hz

4200 Hz

9000 Hz

14400 Hz

18000 Hz

18600 Hz

Figure 8.Example of how the audio spectrum analysis would look like at different frequencies

4. Appendix

Code and its explanation

```
#define IR  11     // connects the OUT of the IR sensor to pin 11 of the Arduino board
#define REC 8      // connects the REC pin of the ISD1820 to pin 8 of the Arduino board
#define PLAYE 9    // connects the PLAYE pin of the ISD1820 to pin 9 of the Arduino board
#define PLAYL 10    // connects the PLAYL pin of the ISD1820 to pin 10 of the Arduino board
void setup(){
 pinMode(IR, INPUT);   //defines the IR pin of the IR Sensor as input
 pinMode(REC, OUTPUT);    // defines the REC pin of the ISD1820 as output

 Serial.begin(9600);
}
void loop(){
 int i = digitalRead(IR);
 if(i == 0)
 {
   Serial.println("The user waved!");        //the IR sensor detected motion
   digitalWrite(REC, 1);                // a recording of the voice has started
   delay(10000);                        //it will record for 10 seconds
   digitalWrite(REC, 0);               //it stops recording
    delay(1000);                       //it waits 1 second
    digitalWrite(PLAYE, 1);              // the recording is played back
   delay(10000);                         //playing for 10 seconds
   digitalWrite(PLAYE, 0);              //the recording stops
 }
}
```

5. Conclusion

Taking everything into account, what can be remembered from everything that has been mentioned in this project is that, it would be quite easy to record and playback a voice or a sound using Arduino Uno and a ISD1820 voice recorder. I am really happy that I was able to work with real components, not just using simulations. It feels more real and useful when you can touch and combine the components and concepts that we were and are talking about at this faculty so much.

This has been a fun and interesting topic to gather information on, and I wish to implement more circuits like this in the future, and maybe more complex projects for personal development.

Bibliography

[1] https://create.arduino.cc/projecthub/gadgetprogrammers/diy-auto-voice-record-and-playback-7a47d7
[2] https://duino4projects.com/arduino-based-bi-color-led-matrix-audio-spectrum-visualizer/
[3] https://www.electronicshub.org/interfacing-isd1820-voice-recorder-module-with-arduino/
[4] ISD1820 datasheet https://datasheetspdf.com/pdf/786127/ETC/ISD1820/1
[5] https://www.electroniclinic.com/arduino-isd1820-voice-recorder-playback-module/