Alerts for the hard of hearing

Designed by Alex Hunt

What is this project trying to achieve?

It is estimated that around 20% of the world's population suffers from some extent of hearing loss. Even relatively low levels of hearing loss can cause difficulties in day to day life. This project is intended to address one of those difficulties.

Shakelet is a simple device consisting of a sensor and a receiver. The sensor detects sound (or more accurately vibrations) and transmits this information wirelessly to the receiver which in turn will flash and vibrate to alert the user. It is intended that the sensor can be stuck to anything that makes noise – doorbells, home phones, baby monitors etc.

Doesn't something like this exist already?

There are commercial alternatives already on the market. However, these typically have high price tags and require you to purchase special phones / doorbells etc. Shakelet allows you to buy standard consumer items and convert them for the needs of the hard of hearing. It also aims to do this at a very low price point.

What is this document?

This document sets out exactly what you need to do in order to make one yourself. It also sets out plans for extending the platform and the technical difficulties that have been encountered.

Bill of materials

Sensor (per sensor)

ATMEGA168p QFNP
Piezo sensor
NRF24L01 mini SMD
5050 RGB LED
1MΩ potentiometer [PACKAGE]
4 x 100nf 0805 capacitors
10kΩ 0805 resistor
15kΩ 0805 resistor
2 x 5.6MΩ resistors
3 x 100Ω resistors
CR2032 battery and SMD holder
PCB

Bracelet

ATMEGA168p QFNP
NRF24L01 mini SMD
5050 RGB LED
SMD vibrating pager motor
4 x 100nf 0805 capacitors
10kΩ 0805 resistor
2N7002 MOSFET
3 x 100Ω resistors
PCB
CR2032 battery and SMD holder

The total BOM cost of one sensor and one bracelet is around $15.

The Sensor

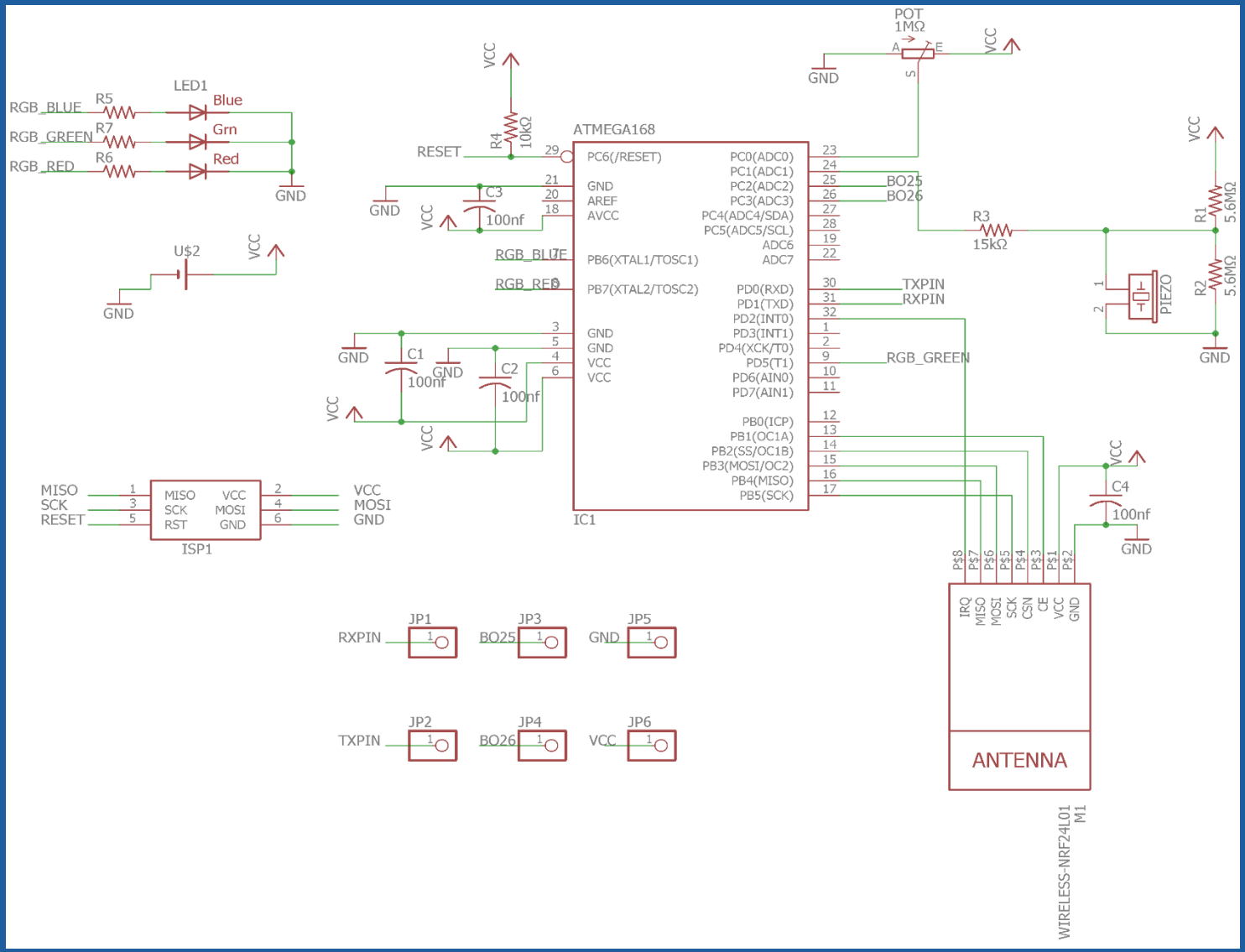
The sensor picks up noise / vibrations through a piezo sensor attached the to the MCU's ADC pin. Two resistors are used to give the piezo a resting voltage of 50% of VCC. A potentiometer is used to set a threshold level in the MCU (effectively to allow the user to set the sensor's sensitivity). Once the piezo's voltage breaches the threshold it send a brief signal through the NRF24L01 to the bracelet.

Circuit design

The full circuit schematic and PCB design is on the following pages. Please note the following points:

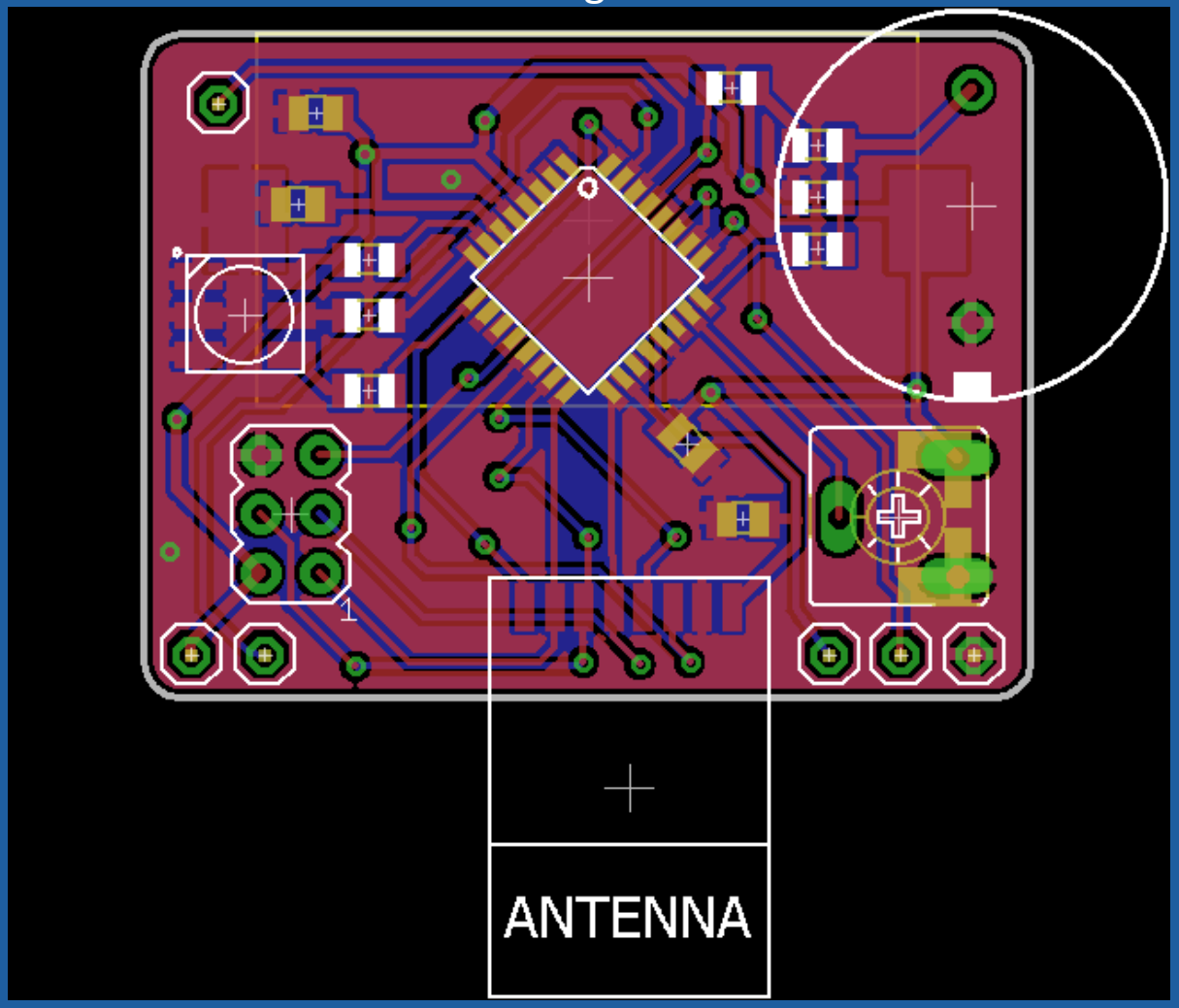
1. Additional pins have been broken out to allow for future additions to the board;
2. An RGB LED has been added. This is not strictly necessary but again may be useful in future development.
3. If more sensitivity is required, the 5.6MΩ resistors can be swapped for higher value resistors.
4. ISP1 is included to allow updating of the firmware at a later date. Again, this is not strictly necessary.

Sensor schematic

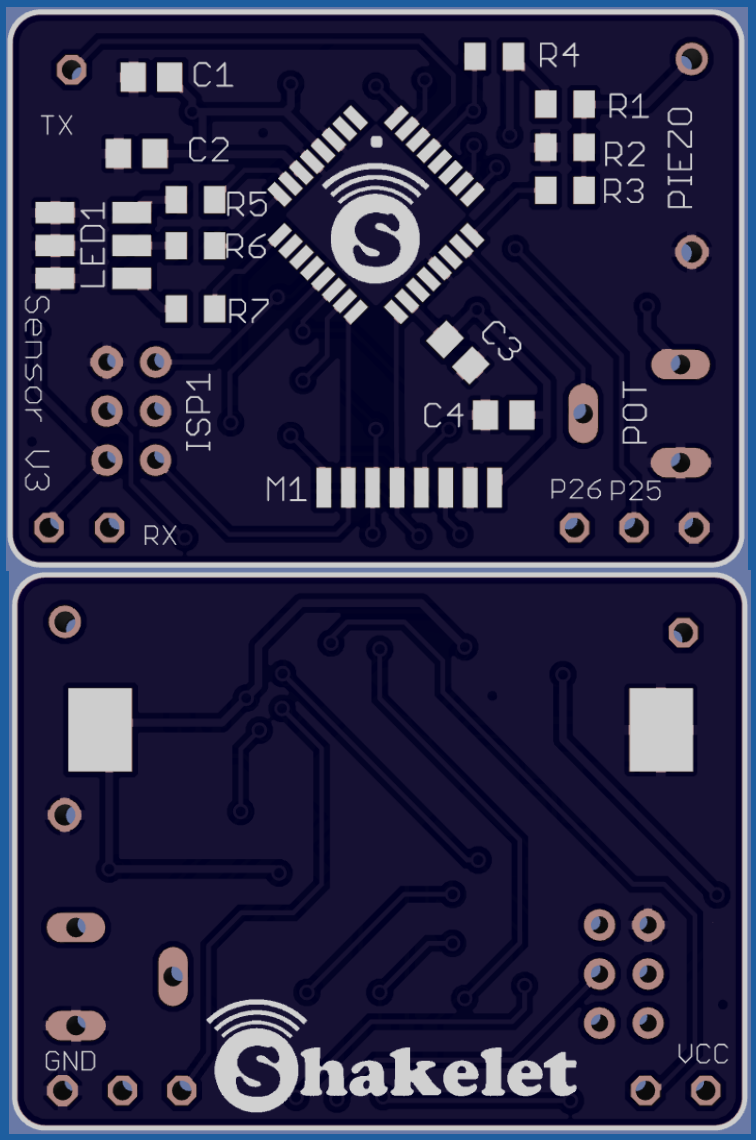


Sensor PCB design

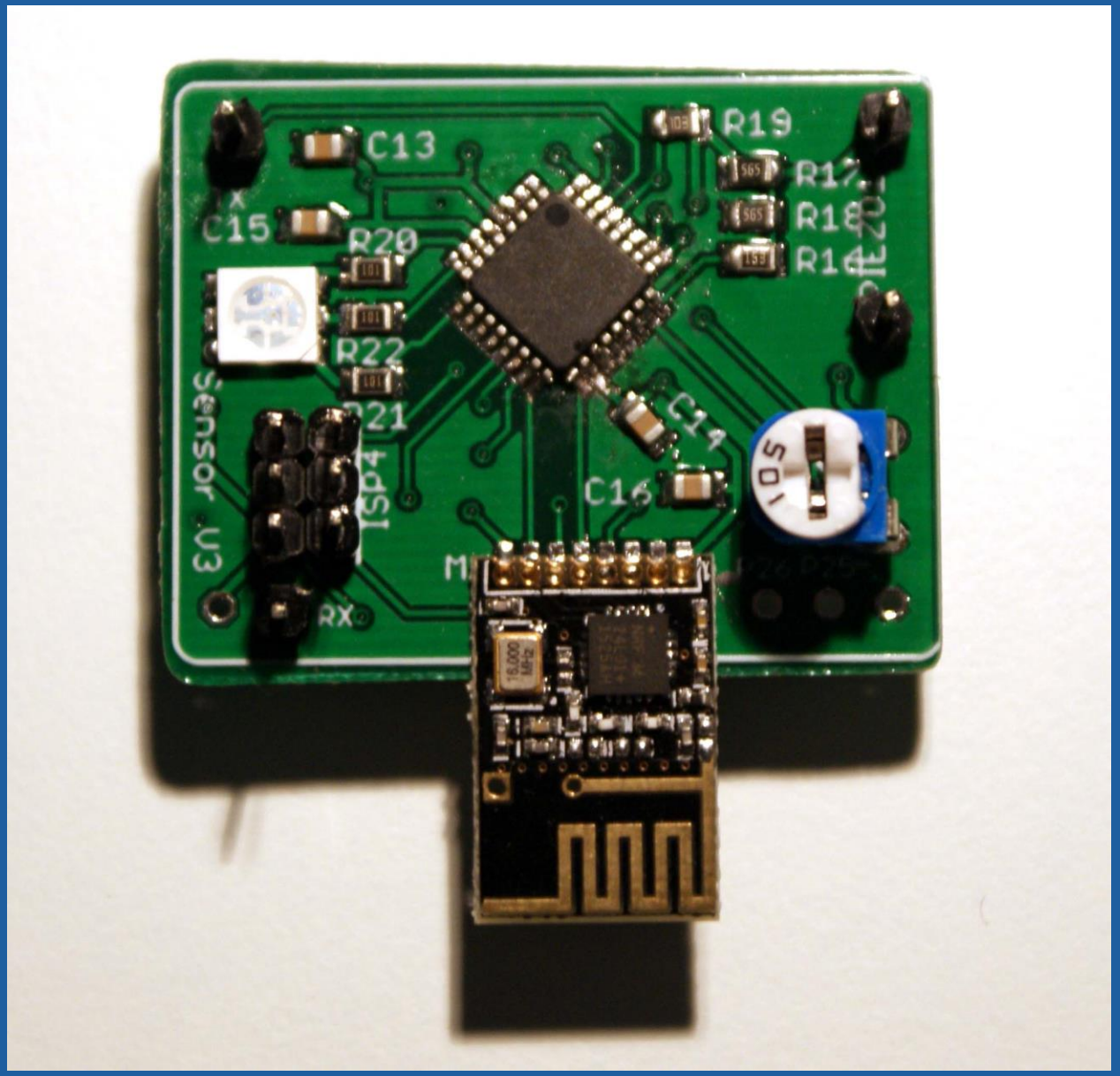
Eagle



3D render



Soldered PCB



The Bracelet

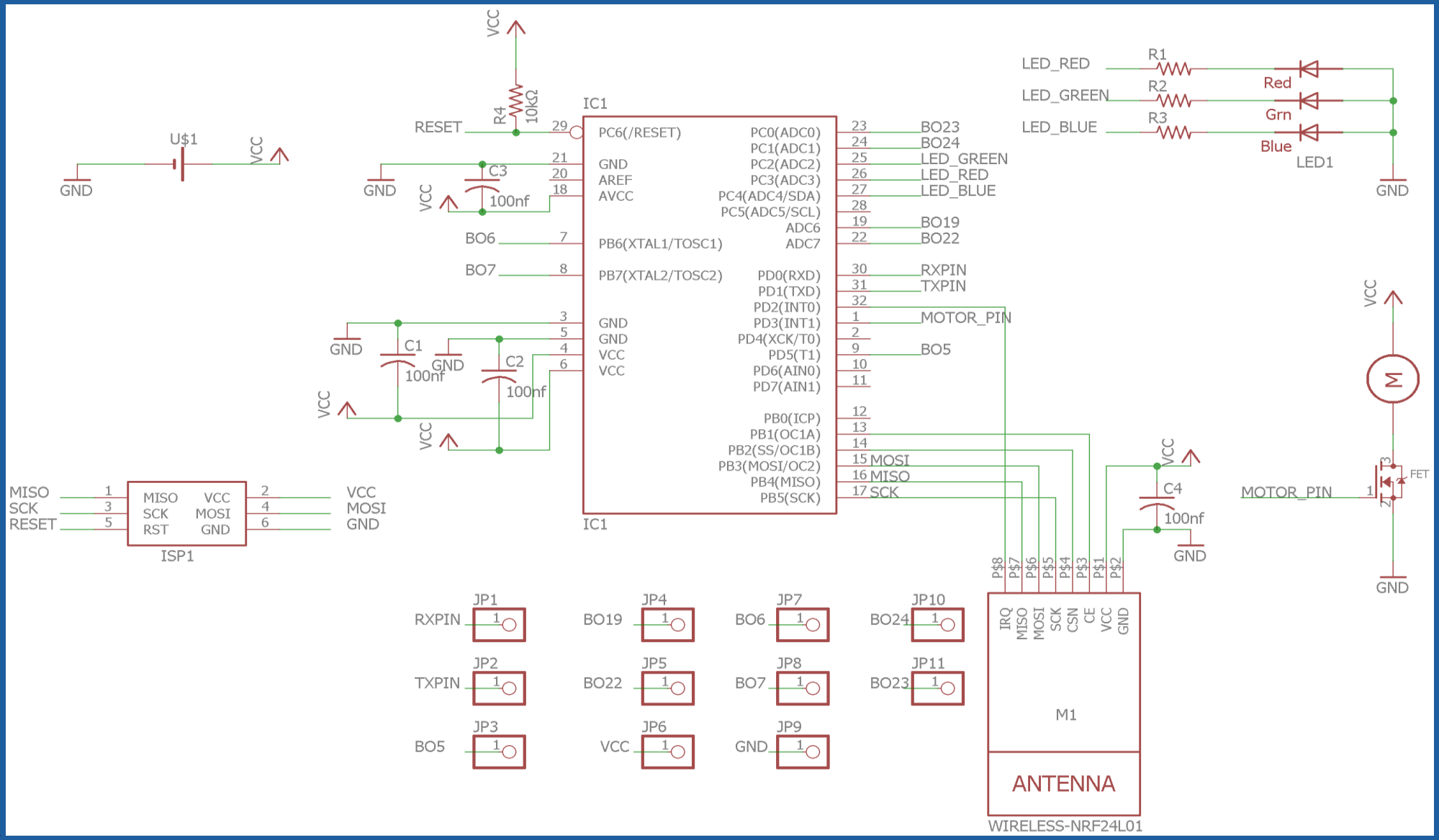
The bracelet picks up the signal from the sensor through another NRF24L01 chip. The signal data indicates which sensor has been triggered and the bracelet then flashes the corresponding colour and vibrates the pager motor with a unique pattern for each sensor.

Circuit design

The full circuit schematic and PCB design is on the following pages. Once again, please note the following points:

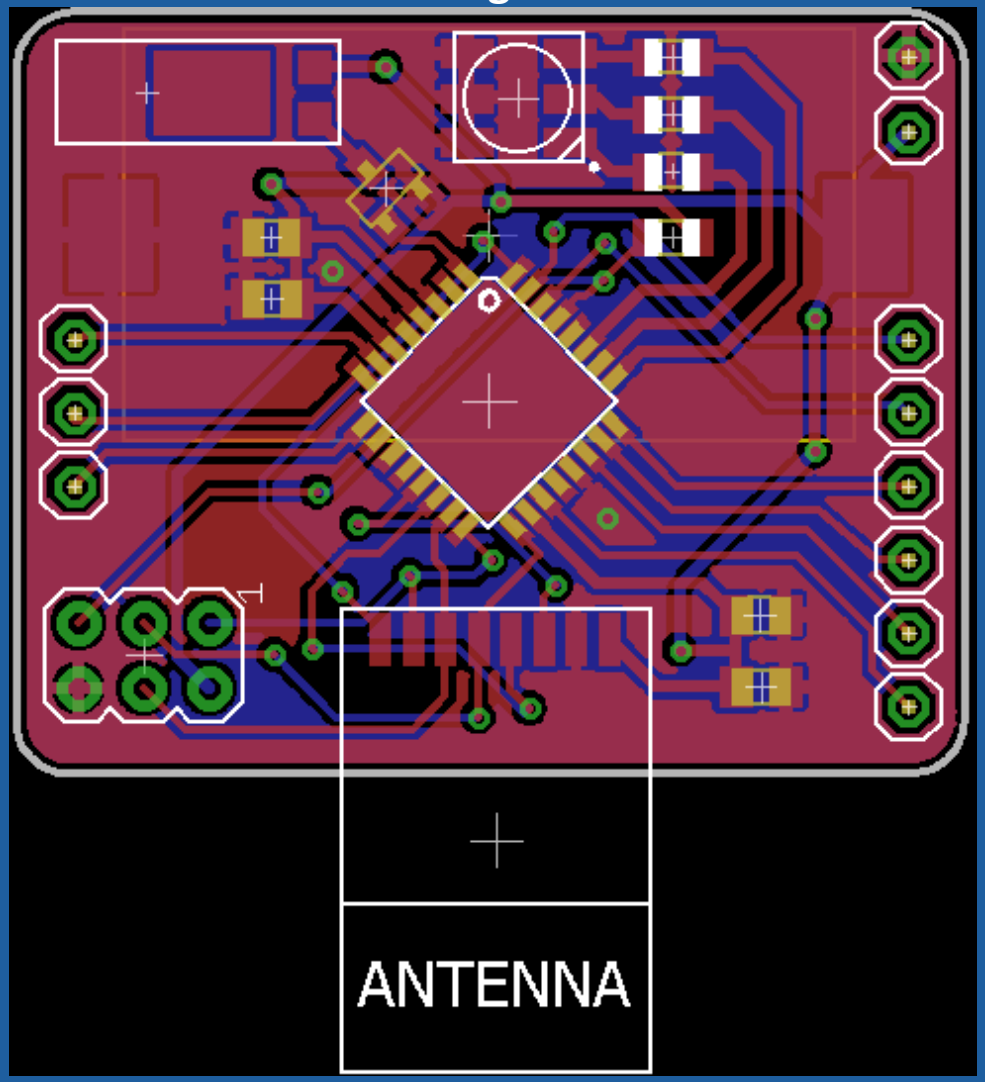
1. Additional pins have been broken out to allow for future additions to the board;
2. ISP1 is included to allow updating of the firmware at a later date. Again, this is not strictly necessary.

Bracelet schematic

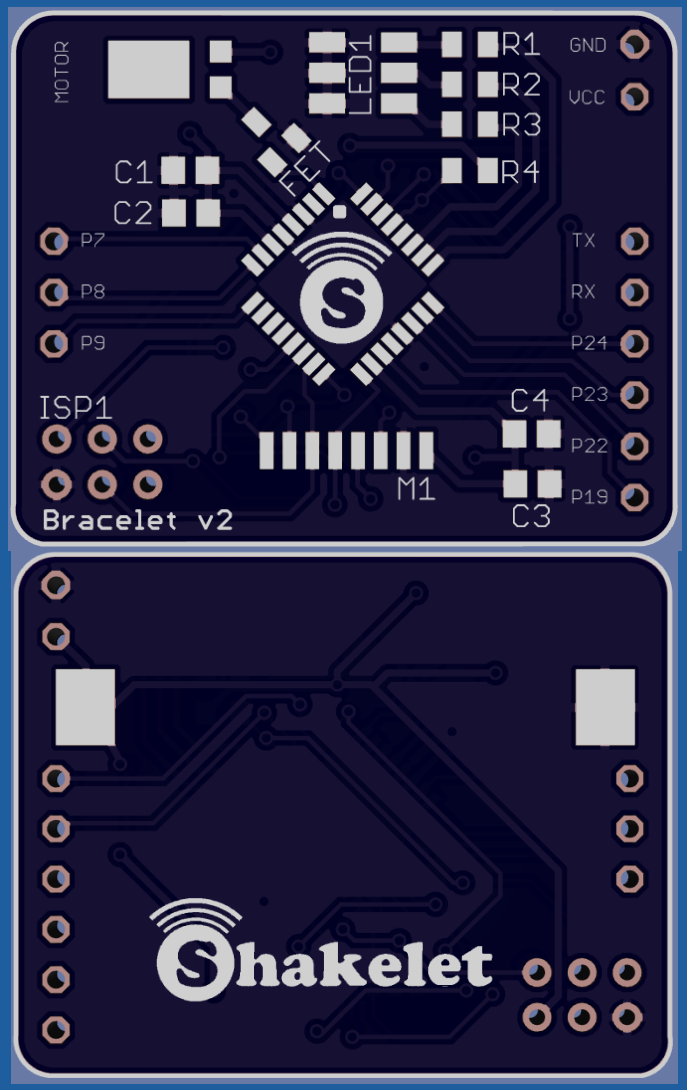


Bracelet PCB design

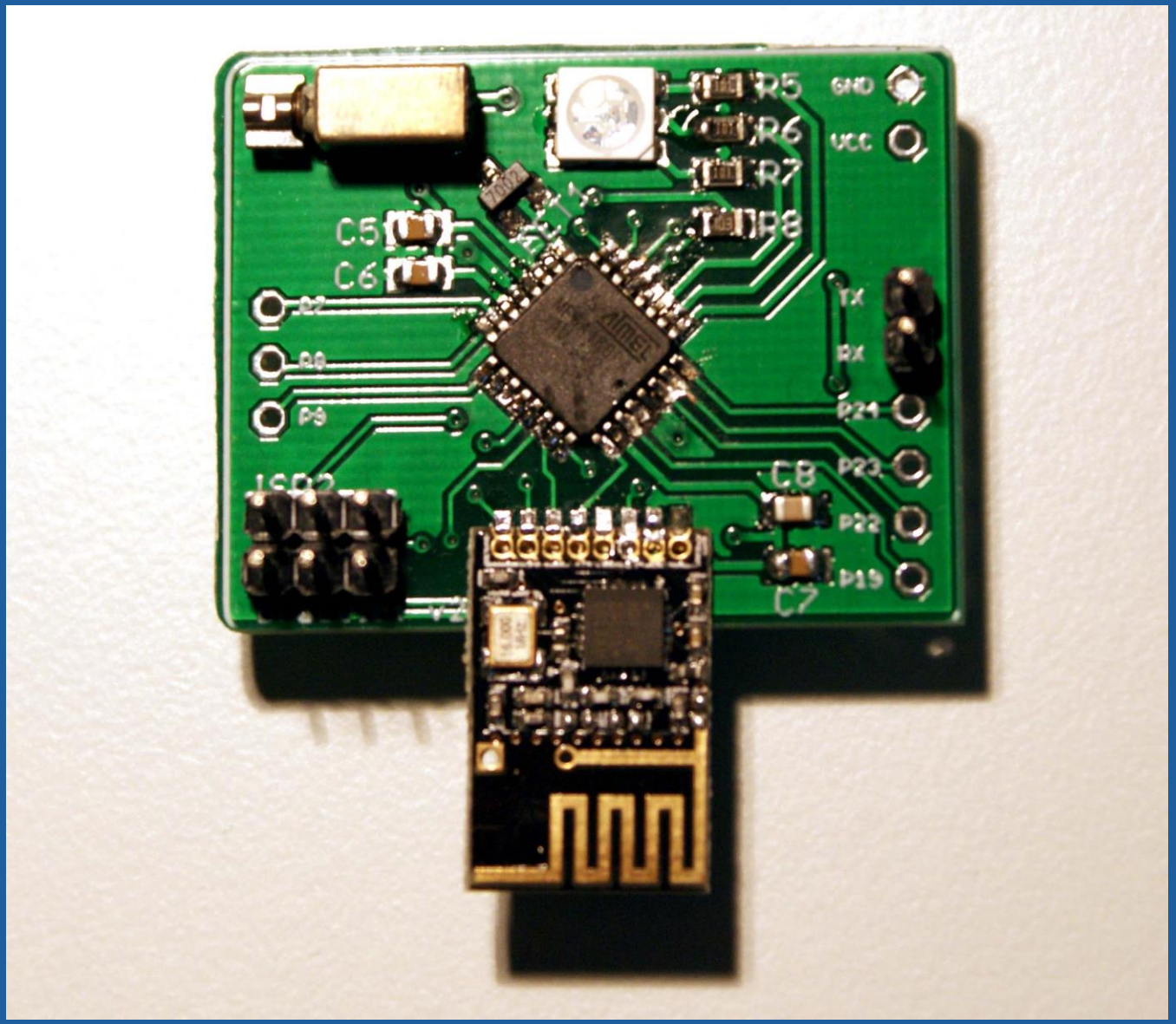
Eagle



3d Render



Soldered PCB



Coding overview

The code for both the sensor and the bracelet can be found on GitHub here: https://github.com/arhunt35/shakelet.

The Sensor

The code for the sensor consists of a free-running ADC which polls the piezo sensor and the potentiometer. As the piezo is split across two resistors, it has a voltage bias of 50% of VCC (and ADC value of 514). This allows both the positive and negative parts of any voltage change in the piezo disk to be monitored. An exponentially weighted moving average is used to calculate the resting voltage of the piezo and to smooth out any noise.

The potentiometer allows the user to set a sensitivity level. Once the value of the piezo exceeds the EWMA value plus the value of the potentiometer (which is amended to give a max figure of 50) then the sensor will send a signal to the bracelet. The signal data consists of a unique ID for each sensor plus the values of the piezo and the potentiometer (in case any future data analysis is required on the bracelet).

The Bracelet

The bracelet idles until it receives a signal from the sensor. Once received, it analyses the signal to pick up the sensor ID. Based on the sensor ID, a simple switch statement will determine which vibrating and flashing pattern should be displayed.

Step by step instructions

1. Download the PCB gerber files from GitHub.
2. Send the gerber files to a PCB house for printing.
3. Purchase items listed on Bill of Materials.
4. Solder components to printed boards (see links at the end of this document for a tutorial).
5. Solder pin headers to the board for the various breakout pins and the programming pins.
6. Download and install AVRDude.
7. Connect to USBASP to computer / programming pins. The solid line on the board has been included to make it easier to use the correct orientation.
8. Flash the appropriate HEX files to the sensor / bracelet.
9. Connect piezo disk to item that you wish to monitor.
10. Adjust potentiometer to set background noise level. Increase the resistance until the sensor is not triggered by background vibrations.
11. Check that bracelet component is picking up transmissions from sensor.
12. For additional sensors, change the SensorID variable in the sensor code (so that the bracelet can distinguish between different triggers).

Outstanding issues and future development

The project is still ongoing and there are a few issues that need to be addressed.

Power

The NRF24L01 modules use a lot of current. The sensor currently uses an average of 2ma whilst idling and 9ma whilst transmitting. The received uses a constant 17ma. I have experimented with keeping both the sensor and receiver in power down mode until needed but have had difficulties with the timings between the modules. I am hoping to reduce average power consumption to 0.5ma so that acceptable battery life can be had from a CR2032 battery.

Range

The NRF24L01+ has a low power / long range mode which is used in the code. The range is acceptable for an average sized family home. To cater for mansion dwellers the range would need to be increased. I have not considered shielding etc. to improve performance of the wireless module.

I considered using Bluetooth of Wi-Fi instead but was attracted by the low price and ease of use of the NRF24.

Casing

At the moment, the project consists of two bare circuit boards. One of the aims of the project was to make something that looked more modern and aesthetically pleasing that the existing offerings on the market. Beautiful cases will therefore be an important part of the finished project.

Future development

There are many possible extensions for the platform (some of which were suggested by the enthusiastic Hackaday community). Here are a few of my favourites which I would love to implement:

1. Include an ESP8266 receiver in the network so that alerts can be broadcast over Wi-Fi. This would remove any range issues and allow for remote notifications and integration with smart watches etc.

2. Transpose the sensed audio signals to feelable frequencies. I liked this idea but wondered how easy it would be to differentiate between different sensors.

3. Create a sensor which only sends a signal when a sound / vibration stops. I like this idea as it would be useful for alerting the user of when something like a washing machine has finished running.

References and thanks

At the start of this project all I had was an idea and no knowledge. In the course of producing my prototypes I have learnt how to design a PCB, how to make custom parts for Eagle and how to solder SMD components. Below are the references I used to help create this project. A huge thank you must also go to Gabe Buckmaster for being so generous with his time in reviewing my PCB designs.

- AVR adventures in low power land – link
- Gizmosnack NRF24L01 tutorial – link
- Sparkfun Eagle PCB design tutorial – link
- Sparkfun creating components in Eagle tutorial – link
- SMD soldering YouTube video by Mulletsrokkify – link
- AVRDude – link