**Designing a wireless, non-invasive heart monitor**

**Mark Harris**

# **Introduction**

I am Mark Harris, a 4th year mechanical engineering student. I set out to create a wireless heart rate sensor, that would be used for doctors to diagnose heart conditions without the patient having to go to a hospital and be fitted with electrodes to determine what the heart is doing. I got in contact with a cardiologist at the Sheffield Teaching hospital (Tracy Cook) and a computer science student (Harvey Cash) working at Passion for Life in Chester.

The project went well, despite a lack of equipment available. This forced me to step outside of my comfort zone, and delve into the computer science side of things, as I feel a lot more at home with circuits and electronics than I do with coding and programming. As a result, I have developed several new skills, including how to use python and node-red.

# **Sensor Design**

When creating a device as sensitive as a heartrate sensor, it is important to mitigate against knocks and vibrations. As such, an accelerometer can be fitted to a heartrate sensor to reduce the risk of skewed data from the heart sensor.

Using the pulsesensor (1) is straightforward, and works best when connected to the earlobe. There is a stronger pulse when using it here, in comparison to the fingertip. There is also a handy clip to allow for this, however the terminals on the sensor itself need to be insulated, to stop it being shorted out. The library provided by the pulsesensor company is also very useful if a simple live heartrate feed is desired, but it is difficult to workout exactly which lines of code do what, due to a lack of commenting. As such an alternative (and much simpler) sketch was found online (2). The sample rate is low (10 Hz) however this is easy to change by adjusting the delay in the loop section. Recommended sample rates are 100 Hz and up. (3)

# **Security**

TTN (The Things Network) states that using LoRaWAN is secure, provided that it is done correctly. Unhelpfully, TTN gives no guidance on how to go about creating a secure network.

# **Wireless application**

TTN recommends using Node-Red and Mosquitto to see incoming and outgoing traffic on the TTN console, however Mosquitto is unnecessary. It’s a useful program to have as it has the sub/pub command lines, and provides a broker, but it’s possible to just subscribe to the TTN broker and monitor the traffic coming through and just save it to a CSV file.

# **Potential for AI implementation**

The heartrate sensor is really only the beginning. There is potential to analyse the data from the sensor using AI and machine learning algorithms, however this is an area that is currently being studied by PhD students, and is beyond the scope of this 10-week internship.

# **A circuit board Description automatically generatedA close up of a device Description automatically generatedSensor**

# **Bibliography**

(1) <https://pulsesensor.com/pages/installing-our-playground-for-pulsesensor-arduino>

(2) <https://create.arduino.cc/projecthub/24Ishan/simple-heart-rate-sensor-using-arduino-7de39d>

(3) <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6085204/>

Sources for python modules

<https://www.scipy.org/install.html#package-manager>

<https://matplotlib.org/users/installing.html#installing-an-official-release>

https://pypi.org/project/pyserial/#files