Sensing IR Pulses emitted from a mineral sample

A. Ng

Introduction
One signal that must be analysed from the rock samples is the presence of an infrared pulse, emitted from two of the six samples: netherite and thiotimoline. Pulses are emitted from the minerals at 571 and 353 Hz respectively. A system that will detect the pulses as well as identify the frequency at which the pulses are transmitted will need to be designed and the solution should be minimal, occupying as little space as possible as our rover needs to be lightweight and reliably detect infrared pulses.

What is a phototransistor and how could it be implemented in this project?
Phototransistors are semiconductor devices[1] that comprise the same structure as the bipolar junction transistor (BJT), a three region semiconductor sandwich of two differently doped semiconductors. These regions are called the collector, base, and emitter where the collector and emitter are the same types of a doped semiconductor, while the base is the other type. Combining two differently doped semiconductors forms a PN diode, thus creating the junction between the base-emitter and base-collector. The difference between the BJT and the phototransistor is the lack of a base terminal pin for the phototransistor. Rather than supplying a current into the base, as we would for a BJT, the phototransistor generates this current internally – this is achieved by having the base region exposed and allowing light to reach this region.

Electromagnetic radiation (EM) such as visible and infrared, can fall onto the exposed region of the PN diode and can generate a current in the base region. All EM radiation has energy associated with them; this is defined by the Planck relation or more well-known as $E = hf$, where E denotes the energy of the photon, h is the Planck's constant and f is the frequency of the EM radiation. If the energy incident on the PN diode is greater than the bandgap of silicon, 1.12 eV[2], electron-hole pairs are produced in the PN diode, and the flow of these carriers generates a current in the phototransistor thus allowing conduction. Using the Planck relation, the frequency of the EM radiation needs to be greater than 2.703×10^{14} Hz. Inspection[3] of the EM spectrum shows that infrared and visible light have frequencies above this threshold and allow the phototransistor to conduct.

This property allows us to use a phototransistor to detect an infrared pulse transmitted by the samples, an input to a circuit that could analyse the conduction/current from the phototransistor and identify the frequency of the IR pulse.

Developing a solution
Previously we encountered the phototransistor in the spring term lab skills to allow our EEE Bug to sense light therefore we could use a similar circuit for detecting the infrared pulses. A proposed circuit for analysing the nature of the emitted pulse is shown in fig. 1.

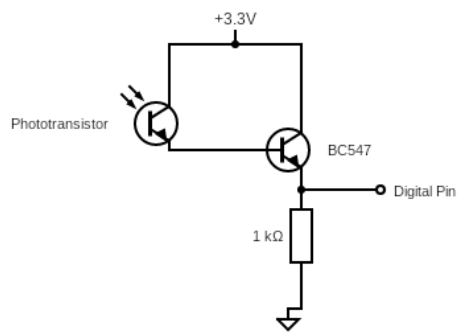

Fig. 1: A circuit to analyse the nature of the infrared pulses emitted by the exorock.

The circuit configuration represented in fig. 1 is of a photodarlington pair[4] where one transistor is a BJT, and the other is a phototransistor. The emitter of the phototransistor is connected to the base of the BJT with the effect of an increased current gain. The current at the emitter of the photodarlington pair is passed through an emitter resistor, which we measure the voltage across it as the digital input for the Adafruit board.

A 3.3V voltage supply was used as opposed to 5V because the logic level of the Adafruit board is 3.3V so using any voltage greater than this would damage the voltage regulator components on the board.

The signal was measured using the Arduino serial plotter software, which reads the digital input voltage from the circuit in fig. 1 and plots the values of the input as either high or low. The same code for the analogue-to-digital conversion

(ADC) task in lab skills was used to plot the digital input readings with both infrared modes of the exorock tested, where results are shown in fig. 2.

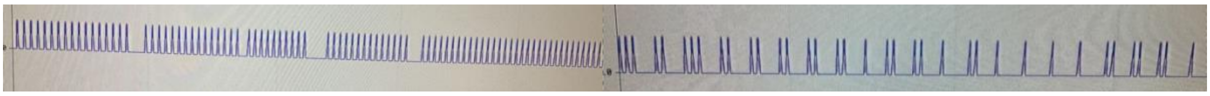

Fig. 2: Arduino serial plotter readings from pulses of infrared of frequency 571 Hz (left) and 353 Hz (right).

Viewing fig. 2 shows the digital pulsed signals received by the Adafruit board. The emitted pulses should maintain a constant distance between them to maintain their frequencies however we do not see this; pulses are inconsistent at 353 Hz but better for 571 Hz. Interference from other infrared and visible light sources could hinder the accuracy of readings therefore filtering the input was attempted. First-order high and low pass filters were implemented at the emitter of the photodarlington pair with corner frequencies at 353 and 571 Hz but provided unexpected results. Despite filtering out signals less than 353 Hz and signals greater than 571 Hz, the serial plotter software picked up pulses without the exorock being in infrared mode – this meant there were still interfering ambient signals that would be removed to determine the pulse frequency accurately.

Fig. 3: Arduino serial plotter readings with no infrared pulses directed at phototransistor. Pulses are a result of ambient signals incident on the phototransistor although a bandpass filter was applied to remove signals outside the pulse frequency.

Also, visible light in the test environment hindered results as this would cause the digital reading in the serial plotter software to be pinned at a “high” and prevented peaks from being displayed. This presented a problem: “how do we remove the effects of visible light on the phototransistor but maintain the effects of infrared?”. Another issue was observed with the circuit in fig. 1 – these readings were measured with the shell of the exorock removed. When placing the exorock shell onto the exorock test board, there was no signal detected and this was checked using the picoscope software. The detected signal was noisy, displaying a pulse with a minimal amplitude; now an amplifier circuit would need to be designed to amplify the signal at the emitter of the photodarlington pair.

Refinements and implementation

Preventing interference from visible light could be achieved in one of two ways: using an infrared phototransistor or 3D printing a custom cover for the phototransistor. Purchasing an infrared phototransistor was what we went for because 3D printing a custom cover would add more weight to the EEE Rover to cover the phototransistor compared to exchanging the existing phototransistor with an infrared phototransistor. The infrared phototransistor purchased was the SFH 309 FA-4/5 ams OSRAM, with each unit costing £0.38. This was an NPN phototransistor, featuring a black plastic cover that blocked visible light signals from reaching the base but didn’t affect the performance of infrared signals.

Improvements introduced by using the infrared phototransistor were analysed using the same circuit as in fig. 1 but with the phototransistor component replaced with the infrared phototransistor. The detected signal is shown in fig. 4.

Fig. 4: Infrared pulse signals read from the exorock board without the shell using the infrared phototransistor.

Viewing fig. 4 shows the nature of the pulsed signal as a square wave. The pulse frequency shown is 567.4 Hz, which would correspond to the 571 Hz infrared pulse mode on the exorock. The readings from the updated configuration were unaffected by visible light sources, making our signal more resistant to interference.

Another issue we encountered was the weak signal perceived by the circuit when the exorock shell was placed on to the exorock board – the pulsed signal was present however at a low amplitude amongst the noise of the signal, so we need to amplify and filter the signal to remove low and high-frequency noise.

Amplifying the signal was accomplished with the use of a non-inverting amplifier with a gain of 20,001. The chosen gain and resistor values for the amplifier were purely experimental as a gain of 11 mildly amplified the already faint pulse signal, resulting in 20 kΩ and 1 Ω resistors being used to achieve this gain in this amplifier configuration. However, depending on the distance between the exorock shell and phototransistor, the signal wouldn't be adequately amplified that it could be read by the Adafruit logic. A second non-inverting amplifier was cascaded with a more conservative gain of 10, achieved with a 10 kΩ and 1 kΩ resistor. The supply rail voltage for the two non-inverting amplifiers was 3.3V for the positive rail and GND for the negative rail. A supply rail voltage of 3.3V was selected over 5V because the amplitude limit of an amplified signal would be equal to the positive rail voltage, 3.3V thus the maximum amplitude of our pulse signal would be 3.3V. This is the logic level of the Adafruit board so there would be no risk of damaging the onboard voltage regulators if the amplified signal was connected to an input pin on the board however a 5V signal would cause damage to the board, which is something we intend to avoid.

Filtering of the signal was achieved with first-order high and low-pass filters, with the corner frequencies of the filters restricted by the available component values in the lab. The high-pass filter comprised of a 470 nF capacitor and a 1 kΩ resistor provides a corner frequency of 339 Hz – this blocks signals and low-frequency noise below our lowest pulse frequency (353 Hz) and the low-pass filter comprised of a 10 nF capacitor and a 27 kΩ resistor provides a corner frequency of 589 Hz – this blocks signals and high-frequency noise above our highest pulse frequency (571 Hz). The positioning of the filters and op-amps in the circuit affected the shape of the output waveform.

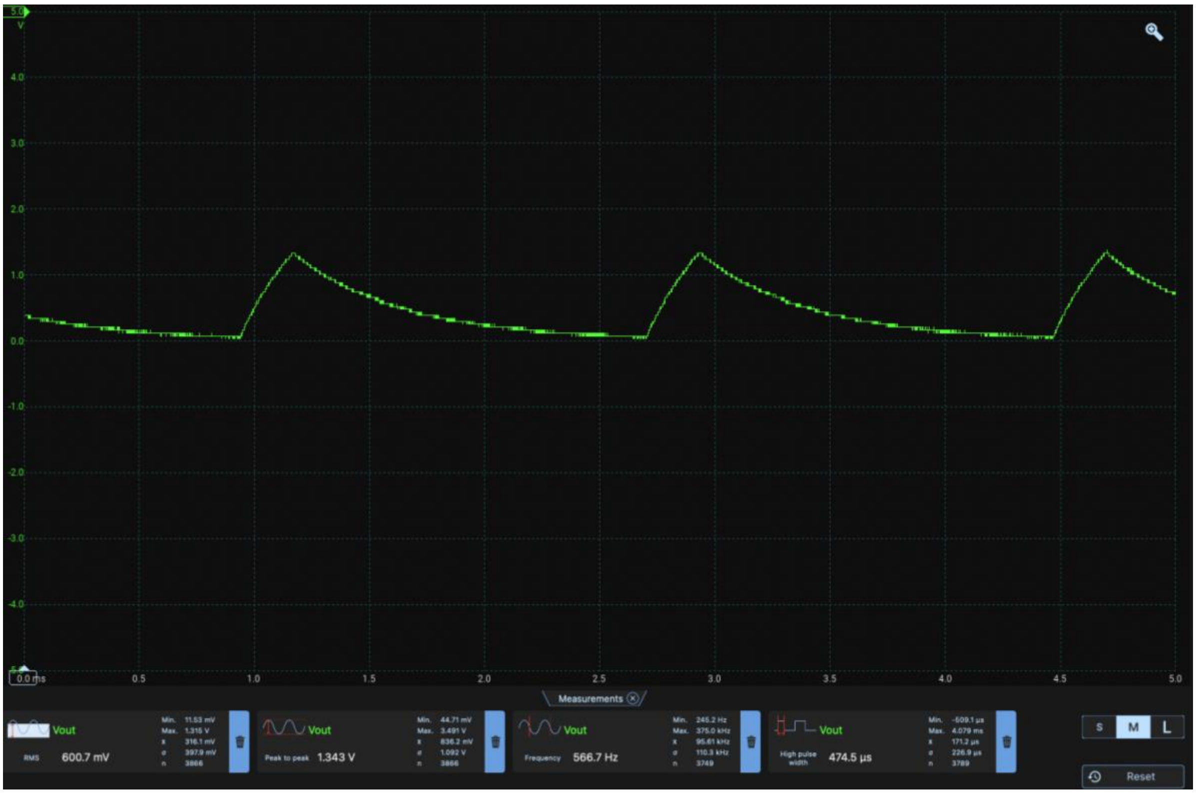

Fig. 5: The output waveform when both amplification stages were conducted before filtering the signal – input signal was 571 Hz IR pulse with the exorock shell on the exorock board.

Fig. 6: The output waveform when both filtering stages were conducted before signal amplification – input signal was 571 Hz IR pulse with the exorock shell on the exorock board.

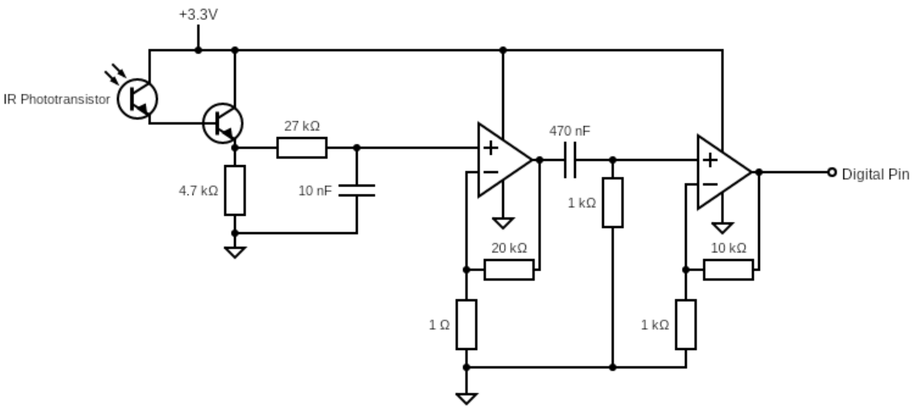

Fig. 7: The final circuit to sense infrared pulses, featuring both amplification and filtering stages of the signal.

Fig. 7 shows the final circuit configuration used to sense infrared pulses emitted from the exorock. In this configuration, the output shows a distinct pulsed signal as seen in fig. 8. Either configuration – amplification followed with filtering, filtering followed with amplification or the one in fig. 7 could be used to process the IR pulses since they report a similar frequency to that of the pulse when viewed in the picoscope software. As aforementioned, the circuit in fig. 7 gives a more distinct pulse than other configurations, which makes it less likely for the Adafruit to pick up false signals and report frequency readings more accurately.

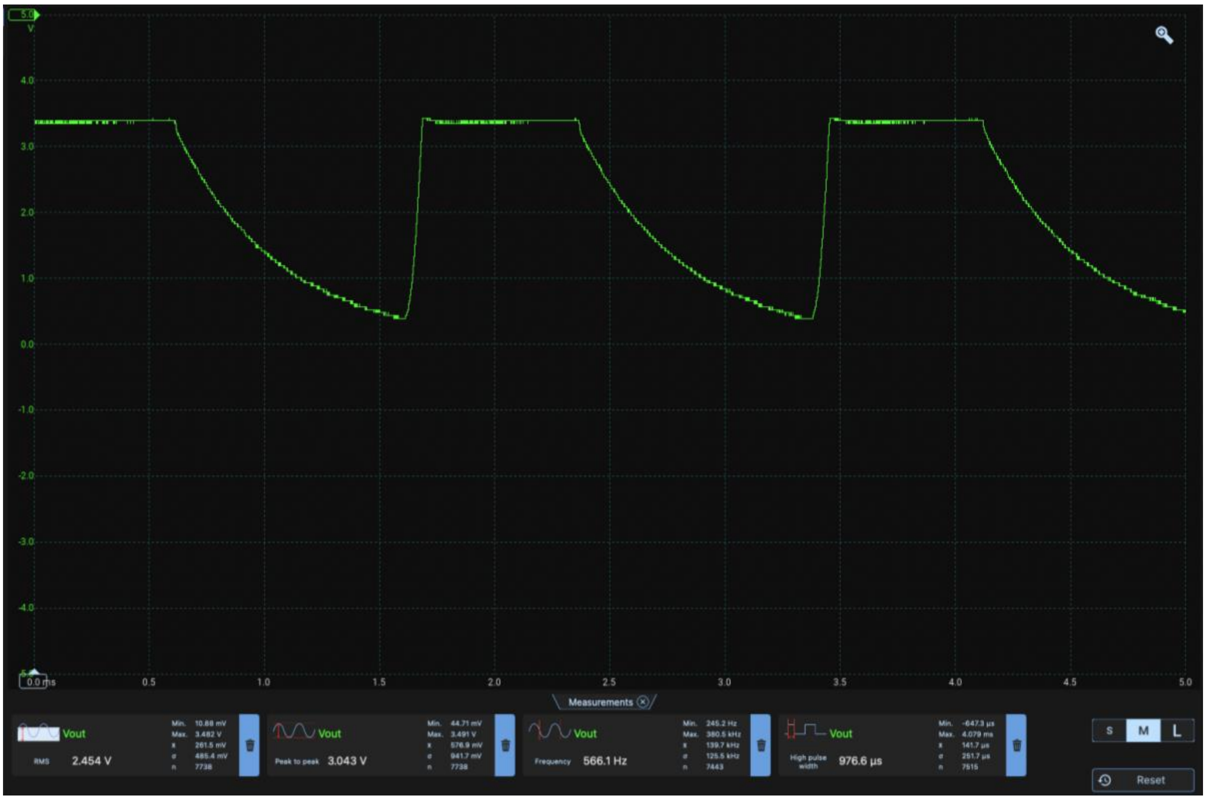

Fig. 8: The output waveform when using the circuit in fig. 7 – the input signal was 571 Hz IR pulse with exorock shell on the exorock board.

Data processing and assessing performance

The objective of this circuit is to detect infrared pulses and report the pulse frequencies. We intend to detect the frequency of the pulse by performing some computation on the Adafruit board by sending the digital signal into a digital input pin on the board. Initially, we were unaware of the process required to determine the frequency of a digital signal and obtained a solution by using some exemplar code on an Arduino forum[5].

Fig. 9: Test code to determine if the frequency was being reported correctly.

The exemplar code was modified to suit our needs and is shown in fig. 9 (left). The period of the pulse is determined by measuring the times when the signal is high and low and adding the values together. The frequency of the waveform is determined by taking the reciprocal of the period measurement. A conversion between microseconds and seconds is required hence why 1000000 appears in the numerator of the frequency calculation as opposed to 1. Lines 16 – 20 write the frequency of the pulse in the serial monitor window – results can be viewed in fig. 10.

Although our code can report the frequency of the pulse, we need to establish a region where frequencies within this region will report the pulsed wave frequency accurately and not report false frequencies despite filtering the signal. We continue to measure frequency readings from the 353 Hz pulsed signal, obtaining a large sample size of around 1000 values. Then the contents from the serial monitor window were imported to excel, where the values were sorted and obtained the boundary values for our region. Using the percentile function to obtain the 10th and 90th percentile gives us the region where 80% of the data will fall into. For the 353 Hz pulse, our 10th and 90th percentiles were 340.91 and 366.78 Hz respectively. The whole process was repeated for the 571 Hz with percentiles of 564.97 and 585.28 Hz respectively.

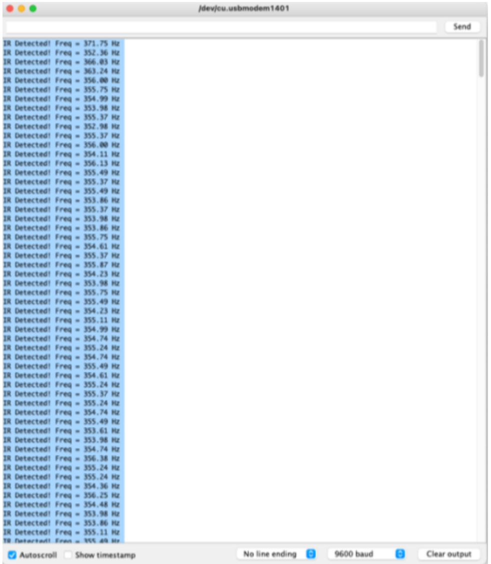

Fig. 10: The serial monitor window when the code in fig. 9 is run, using the circuit in fig. 7 and pulse frequency of 353 Hz.

Fig. 11: Updated Adafruit code to determine the frequency of the pulse and distinguish between the two types.

We implement these percentile values into the region of acceptable values for our code accurately report the pulse frequency, which can be viewed in fig. 11. Now we can test if our circuit can amplify the signal and filter out unwanted signals and test if our code can correctly detect the different pulse frequencies.

Firstly, shining visible light at the circuit will not cause any signal to appear at the output since visible light is not accepted by the IR phototransistor – this solves the issue of interference of the signal from ambient lighting. Next, we assess the effectiveness of the circuit in fig. 7 by placing a picoscope probe out of the output node, which the detected signal is the same as in fig. 8 for both pulse frequencies. Then attaching the output node to digital pin 4 (the pin chosen for testing the code), we obtain the output in the serial monitor window in fig. 12.

Our code successfully distinguishes between the two pulse frequencies with the given parameters in fig. 11. This is as far as testing for this section will go since we just need to test the if statements used to detect the two pulses. The IR sensing code in the final code uploaded to the Adafruit will feature different lines of code due to integration with the web interface.

Fig. 12: Serial monitor window showing both pulses (353 and 571 Hz) being detected.

References:

1. Circuit Globe. Phototransistor [Internet]. Circuit Globe; n.d. [cited 2022 May 30]. Available from: https://circuitglobe.com/phototransistor.html
2. Toshiba Electronic Devices & Storage Corporation. What is a wide-band-gap semiconductor? [Internet]. Toshiba Electronic Devices & Storage Corporation; n.d. [cited 2022 May 17]. Available from: https://toshiba.semicon-storage.com/eu/semiconductor/knowledge/faq/diode_sic-sbd/sic-sbd001.html
3. Phet University of Colorado. The EM Spectrum [Internet]. Phet University of Colorado; n.d. [cited 2022 May 17]. Available from: http://labman.phys.utk.edu/phys222core/modules/m6/The%20EM%20spectrum.html
4. Electronics Notes. Photodarlington [Internet]. Electronics Notes; n.d. [cited 2022 May 17]. Available from: https://www.electronics-notes.com/articles/electronic_components/transistor/what-is-a-photodarlington.php
5. Ambhatt. Frequency and Duty Cycle Measurement Using Arduino © GPL3+ [Internet]. Arduino Project Hub, Ambhatt; 2019 August 31 [cited 2022 May 29]. Available from: https://create.arduino.cc/projecthub/ambhatt/frequency-and-duty-cycle-measurement-using-arduino-1e4896
6.