

Part 1

The first part is just about sending a message in morse code using the Raspberry Pi 4. This can be done by just connecting the LED to a GPIO pin and periodically turning it high or low. I chose GPIO pin 4 for this purpose.

I then created a python program called "send.py" that does this using a library called RPi4.GPIO. I just turn it high and then sleep for a certain duration to make the LED light up for that long.

I used my own timings for the dots and dashes in the program.

Part 2

The second part involves connecting the ESP32C3 board to the Photoelectric Sensor. An example file under esp/esp-idf/peripherals/adc/oneshot_read can read the voltage produced by this light if you place it in the right channel. Which turned out to be GPIO3 on the ESP32C3 for me.

I used the example file to print to console a bunch of times. When the LED lit up, the voltage jumped to 110 mV, so that's what I used as my threshold.

It took me 3 mins 6.52 seconds to transmit a 35 letter sentence 5 times, which gives me about 0.93 characters per second

Part 3

I realized that instead of using a for loop to search for the correct translation of the morse code, I could just use a hashmap which uses $O(1)$ time instead of $O(n)$ that I had before. Now I could improve the timings from 0.1 seconds to 0.06 seconds, because the overhead from the translation function was gone. So there was a 40% improvement.

It took me 1 minute and 52 seconds to transmit the same 35 letter sentence 5 times, giving me a speed of 1.56 characters per second which actually gives me a speed increase of about 38.8% which falls in line with my assumption.