**SHF Beacon Monitoring System**

Eric Dollinger

Abstract:

The SHF Beacon Monitoring System monitors an amateur radio beacon that is operating at 10 GHz. In order to monitor the beacon, a repurposed Police Microwave Radar Antenna is attached to a Raspberry Pi Pico W that analyzes the incoming morse code, or CW, signal and determines if a 10 GHz beacon is broadcasting. The Raspberry Pi Pico W will notify custodians of the beacon via email if it is fully operational once a day or offline if it is not heard every 15 minutes. The main objectives of the monitoring system are to receive a 10 GHz CW signal, convert it from analog to digital using the Raspberry Pi Pico W’s onboard analog-to-digital converter, analyze the incoming signal on the Raspberry Pi Pico W, determine if the received signal is the desired 10 GHz beacon, and notify a set list of custodians via email about the operational status of the beacon.

Setup:

You will need a Raspberry Pi Pico W, a custom circuit board from John Gibbons, and a repurposed Police Microwave Radar Antenna (as seen in Figure 1). Then you need the MicroPython Code for the SHF Beacon Monitoring System (see Repo). Then you need a way to upload the code to the Raspberry Pi Pico W. I used Visual Studio Code with some Extensions to be able to upload/download and connect to the Raspberry Pi. The extension PyMakr allows for connections with devices running MicroPython, but you must have node.js installed on your computer before installing it on Visual Studio Code. Once you have Visual Studio Code setup with PyMakr, you can connect to the device. This is the link to the guide I used to initially set it up on my system (<https://randomnerdtutorials.com/micropython-esp32-esp8266-vs-code-pymakr/>). If at any point, you are unable to connect to the Raspberry Pi Pico W or get the Raspberry Pi Pico W to connect to the internet, make sure you are running MicroPython on the Pico W and that the firmware is at least from December 2022 as they fixed a bug that would not allow the firmware to connect to an unsecured network. Firmware I used can be found on the SHF Beacon Monitoring System along with the nuke file that clears the previous firmware and files if you accidently run code that crashes. See the Raspberry Pi Pico W documents on their website for how to change/update the firmware.

Approach:

A picture containing bat, mammal, night

Description automatically generatedFirst, the modified Police Microwave Radar Antenna was connected to the Raspberry Pi Pico W using a custom circuit board designed by John Gibbons. Whenever the receiver receives a 10 GHz signal, its voltage output changes from 2.6 V to 3 V. When the signal received is above 2.9 V, it is considered to be in a high state. When it is below 2.9 V, it is in a low state. In the main class on the Raspberry Pi Pico W, a timer interrupt is called every 5 milliseconds. This interrupt checks the voltage of the incoming signal from the Pico W’s ADC. Once the interrupt sees a high state, it starts counting the number of high states. When the value of the signal is below 2.9 V, the number of segments is tallied and determines if the beginning of the incoming signal is one or three units. Then a dot or dash it added to a string. This sequence is then repeated until there is no high signal for one second. The string is then decoded from Morse Code and compared to the known beacon message that is being listened for. If the beacon message is heard, a Boolean is set to true. Another interrupt is called every fifteen minutes that checks if the signal has been heard based on the state of the Boolean. If the signal has not been heard, an email is sent to notify the custodians that the beacon is offline. If the beacon is heard, the Boolean is set to false, and a counter is increased. Once the counter hits 24 hours, a daily update email is sent to the beacon’s custodians.

Figure 1: 10 GHz SHF Monitoring System – Receiver and Raspberry Pi Pico W

In order to send the email, a webhook is created on IFTTT (<https://ifttt.com/>) that receives a JSON payload sent from the Raspberry Pi Pico W using the [W8EDU@case.edu](mailto:W8EDU@case.edu) api token. The [W8EDU@case.edu](mailto:W8EDU@case.edu) account on IFTTT was created by Matthew Canel (ask him for the password). Once the JSON payload is received, it sends an email to [W8EDU@case.edu](mailto:W8EDU@case.edu) with the data from the JSON string formatted so that the club officers can easily understand why the email was sent, either because the beacon has not been heard in the last fifteen minutes or because twenty four hours has passed without the beacon going down.

Repo:

<https://github.com/etd20/SHF-Beacon-Monitoring-System>

<https://github.com/etd20/SHF-Signal-Generator>