List of Tests to be Completed by December

Test to Perform for 5 Functionalities:

Functionalities	Tests
Receive Signal	• Verify distance to receive signal • Simulate signal and verify it can be picked up by device, compare to signal analyzer • Measure level of signal received from source at know distances • Ensure bandwidth of signal received is wide enough to get accurate reading
FFT	• Test what size ‘buckets’ work best for producing a consistent output • Test how long the FFT and processing takes before returning data • Test and compare different SDRs and processors as needed to determine which pair/set works and what are key features to focus on • What type of resolution of our signal can be calculated
Transfer data	• Verify data input as fault is processed and results in known output • Test how storage works, make sure to update or clear data after uploaded/transferred to PC/server for mapping • Do not overwrite valuable data
GPS	• Test to make sure we can read the coordinates of GPS device • Verify I&Q data are accurate by comparing with other devices or know coordinate points that are known • Test the processing time and make sure we can output or attach GPS data onto our other data • Calibrate GPS movement while driving to ensure more accurate GPS data is calculated
Map Data	• Test precision of heat map compared simulated location of fault • Test intensity levels plotted vs data input • Test multiple uploads and over various spacing of time

Kenneth Kleinsmith
Ian Furniss
10/24/18

Risk Areas, Consequences, and Contingencies:

Area of Risk	Consequences	Contingencies
Antenna does not receive signal at desired ranges	The reception and analysis of the arcing power line signal is the backbone to our project. Without detecting the signal effectively, our project is kaput.	We will be testing a variety of antennas as well as testing various frequencies to figure out what is the best frequency (or frequencies) to look for our signal.
SDR does not process data as fast as needed	If we cannot keep our calculations to a timely manner, our data once finished may be unusable due to wait time, data loss, or add larger error to our final data results.	By testing the processing time, by purchasing a decent SDR, we can reduce the calculation time. Additionally, we are considering adding a MPU or MCU to perform the 'heavy lifting' calculations to speed up the calculation time.
SDR does not interface and output data as desired	If we cannot read in the data from the SDR, there is no final-result data. We need to be able to obtain the data from the SDR and so that we can perform further calculations on the results.	We have considered buying an adaptor if needed or could consequently build one if we cannot find an adequate solution. Additional programming could also be done to modify the output data so that we can obtain the formatting we desire.
GPS does not output data in format or at speed required for accuracy	Lots of GPS units output data in various ways, not having the data output as we need would throw off the location that the fault is detected or not provide valid data.	We can purchase alternate GPS devices to obtain the desired output. Also, we can purchase different levels of speed for which the output is pushed from the device.
GPS does not connect to or interface with SDR	If our GPS does not work with our SDR, we would lose functionality in outputting the heat-map of the results thus removing of the most important qualities of our device.	By purchasing an alternate GPS or building an adaptor so that we can make sure the signal is received correctly. We may consider writing a small script or change the programming on how the device expects to receive the data.
Signal is hard to distinguish from noise or other dedicated signals (radio)	If we cannot extract our signal from ambient noise, we will have not have a clear signal to analyze. If we cannot determine the difference between our signal and other broadcasted (on purpose) signals, we will obtain false positive for faults.	To reduce the noise floor, we can take more sampling buckets (averaging over narrower bands). We can add filters to pre- and post processing to help distinguish between a broadband arcing signal and other broadcasted signals. Layering a more complicated averaging algorithm can help filter out unwanted data as well.
Output data is in difficult format to deal with (plot)	Another key feature is to plot the detection levels of arcing powerlines on an online map. Without the user being able to view faults, there is not alternative way to view the results from our device.	By choosing different ways to output and store the data, we can modify how the initial data is output. We may also post process the data before it is stored if it still isn't in a usable format. Though this may add some delay in the time between computation and storage, this would not affect the results of our final data.

Ensuring data is stored and uploaded correctly	If we do not store or display the uploaded correctly, our results of our fault level (heat map) may contain errors or lose valuable data of previously detected faults.	If our data storage device does not work, we can try to supplement the memory storage capability (flash drive or larger SD cards). We can write a program that simply adds the data to the map in addition to what is there so that we do not lose data. By using a cloud system, we do not have much worry about our online map data filling up. Otherwise we can buy more space or use an alternate mapping service.