FERMILAB-CONF-19-390-AD

Data Acquisition System for RF Cavity Hadron Monitor

Kyle Leleux
University of Louisiana at Lafayette
Lee Teng Internship, Summer 2019
Katsuya Yonehara, Fermilab, mentor
August 12, 2019

Abstract

As Fermilab pursues intensity frontier physics, the conventional detectors are not up to the challenge of higher radiation exposure. The detectors in the NuMI beamline rely on knowing the angular distribution of the secondary hadrons in order to infer properties of the resultant neutrino beam. A novel gas-filled RF cavity beam detector is proposed that will be simple and radiation robust in high-radiation environments. Charged particles passing through the cavity produce ionized plasma, which changes the permittivity of the gas and modulate the stored RF field in the cavity. The preliminary beam test of an RF cavity demonstrates the proof of principle of the beam detector. In an effort to acquire a large amount of data for various cavity parameters, an automatic data acquisition (DAQ) system is designed in LabVIEW. The DAQ system utilizes a Real-Time Power meter from Boonton to acquire RF power data from the cavity.

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

1 Introduction

The future of accelerator technology demands innovative and creative ideas while addressing the shortfalls of current technology. The push for more intense beams introduces the problem of higher radiation environments. This causes issues with most beam detector technologies. For example, the current hadron monitor in the NuMI beam line is an array of ionization chambers, whose design necessitates the use of a ceramic insulator [1]. As the detector is exposed to more radiation with each beam spill, the insulator is damaged by radiation and the leakage current becomes larger. As a result, the chamber loses the linearity of a beam signal and cannot reconstruct the passing beam intensity from the signal. In the worst case, the chamber could lose signal entirely. In an effort to develop the next generation of rad-hard hadron monitors, a gas filled RF cavity is proposed. Incident beam proportionally generates a plasma by interacting with a gas in the cavity. The gas modulates the RF signal. The RF cavity is manufactured solely from a metal, but no electrical insulator is used to improve radiation robustness. Aside from a rad-hard beam detector, the RF signal can also be calibrated after each beam spill by observing the quality factor of the cavity. So even if radiation damage does occur, the device may still reproduce the passing beam intensity [2].

A proof-of-principle experiment was carried out using the MI abort line in 2018. Figure 1 shows the modulated RF signal by one beam spill. Because the data is taken using a vendor-made data acquisition (DAQ) system, it is not suited to the R&D experiment for the following reasons:

- The DAQ is self-triggered by the decayed RF signal, causing it to be primarily triggered by RF noise
- The file name and directory are specified after each event
- The beam profile and intensity are taken using a SWIC and a current transformer toroidal coil. However, the RF beam signal is not synchronized to them. Therefore, the timing of events is adjusted by hand, often inducing systematic uncertainty.
- Quality factor measurements are made manually.

Therefore, a DAQ system based in LabVIEW is proposed. This document reports the status of the new DAQ system.

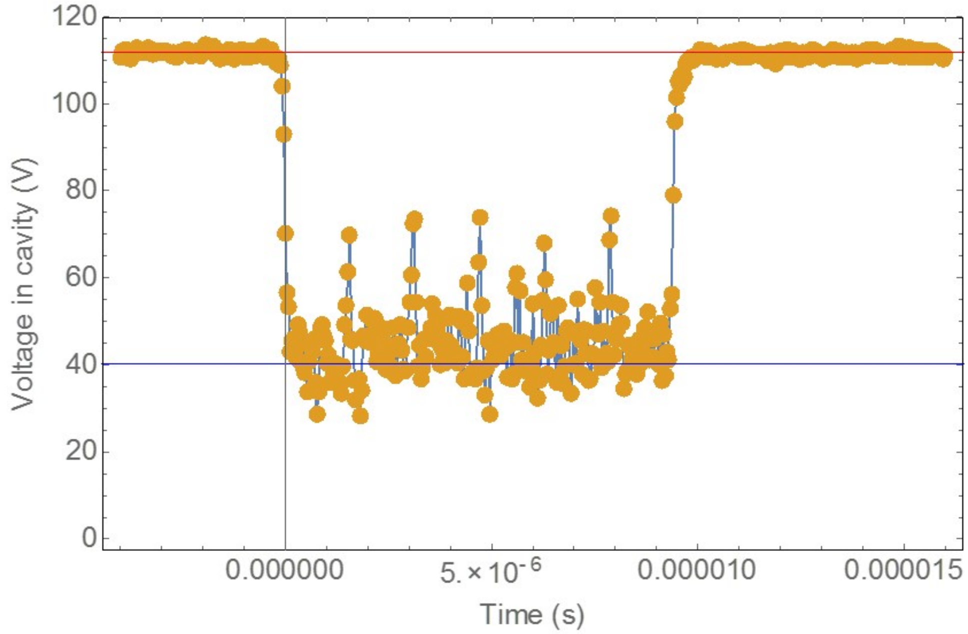

Figure 1. RF Power Waveform with Interaction with Beamline

2 Principles of DAQ System for RF Hadron Monitor

In order to effectively design a reliable data acquisition system, the system parameters must be known. The layout of the NuMI target system is shown in Figure 2. A 120 GeV/c proton beam smashes into the 1.2m long Carbon graphite target and creates subatomic particles that are directed into the decay pipe. Some of these particles decay into neutrinos that are utilized for neutrino experiments. The proposed hardon monitor system is designed to be placed directly after the decay tunnel as a monitor of the beam composition. It monitors the beam centroid to ensure that the primary protons strike the target by measuring the beam profile. The required spatial resolution of the hardon monitor is 0.1mm. The monitor should detect deterioration of the target by measuring the beam flux. The designated flux resolution of the monitor is 5%. The intensity of the primary proton beam is 5.5e13 protons/spill and the spill cycle is 1.33s at 750kW beam operation. About 10% of the changes beam goes into the hadron monitor. This suggests that the single RF cavity of the hadron monitor will receive 10^{10} - 10^{12} changed particles/spill.

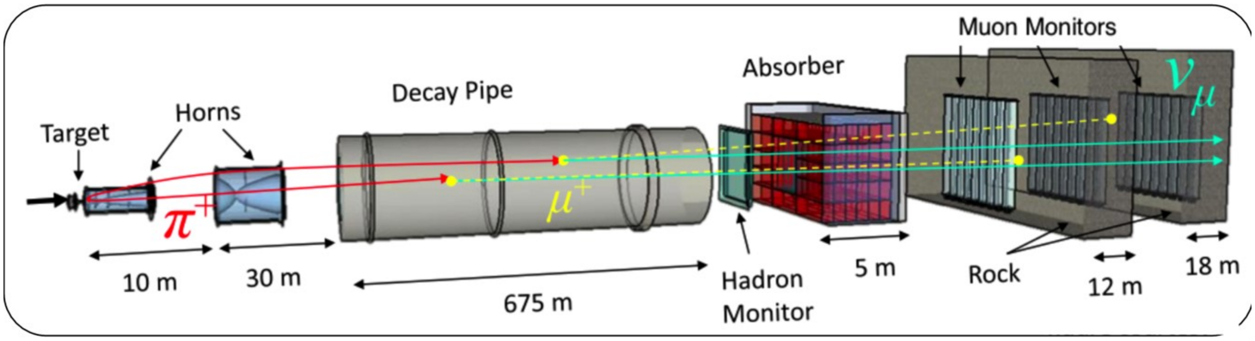

Figure 2. NuMI Beamline Target Chamber

For the aforementioned RF hadron monitor system, the parameter of most concern is the change in the RF power stored in the cavity as it relates to the quality factor. This change occurs because of the interaction of the beam with the gas inside of the cavity. The beam ionizes the gas within the cavity and produces the waveform seen in Figure 1. For the time scale of the RF hadron monitor, on the order of nanoseconds, it is not ideal to continually collect and save data. For this reason, it is important to have trigger events that tell the DAQ system when to collect and save data. A TTL signal can be sent from the main control room as the accelerator clock timing. This signal arrives every 1.33s for the entirety of the beam spill. The beam spill occurs for 10μs. The interaction with the gas-filled RF cavity ionizes the gas and creates a plasma and consumes some of the RF power stored inside of the cavity. In order to collect this data, the Boonton RTP 5006 RF power meter is utilized along with LabVIEW to control the DAQ system. In the next section the front end design methodology is described including instructions on how the user would seemingly interact with the system and the back end principles are described.

2.1 Front End of Data Acquisition System

The front end of the DAQ system is what the user ideally only has to deal with. The front end, shown in Figure 3, is split into three panels: the RF Power Readout Panel, the DAQ Settings Panel, and the Status Panel. The DAQ Settings Panel, left-hand side panel in Figure 3, is where the user specifies input parameters to the sensor, such as the time base, number of samples to collect, and the mode of operation. The time base changes the rate at which measurements are taken while the number of samples determines how long the sensor takes data. The modes of operation are continuous run, batch run, test run, and debug mode. Continuous run will continuously take data based on the parameters initially assigned. Batch run allows the user to specify the number of times to record data and at varying sensor parameters. The idea of this is to automate the testing procedure when varying the properties of the cavity. It also allows the user to specify the cavity parameters. Test run takes in one run of data in order to determine if the DAQ system is correctly tuned to the Main Control Room signal. On top of all of these settings, file name and path are also specified here. The user then selects the Initialize Device button in this panel whenever they are done manipulating the device parameters.

The second window that the user can interact is the readout window. The RF Power Readout Panel plots the waveform collected from the sensor buffer. The RF Power Readout Panel, center panel in Figure 3, takes up most of the screen and provides a readout of the most recent waveform. Since the main measurement of concern is time-domain RF power, the RF Power amplitude is plotted versus the sample number, which refers to a certain time dependent on the timebase specified in the DAQ Settings Panel. The plot is programmatically tuned as to always display the correct number of points as specified in the DAQ Settings Panel. The default values are 1250 samples at a 500ns/div time base, which is a sampling rate of about 100S/us. The plot will update each time it gets a new array to plot, which is generally every 1.33s or every beam spill. This is the largest panel in the front end as this is what the user is mainly concerned with. As the program is running, the Status Panel updates the user on current state of the program in real time. The Status Panel is pictured in the bottom of Figure 3. This includes things like if there was an error encountered, the status of the sensor, the status of the trigger, and others. This is implemented again as a way to limit the interaction of the user with the backend. The Status Panel is also where the user will find the STOP ALL button, which safely closes the connection to the sensor and prompts the user if they want to start data processing, which, if toggled, is done so with the built in Python sessions in LabVIEW.

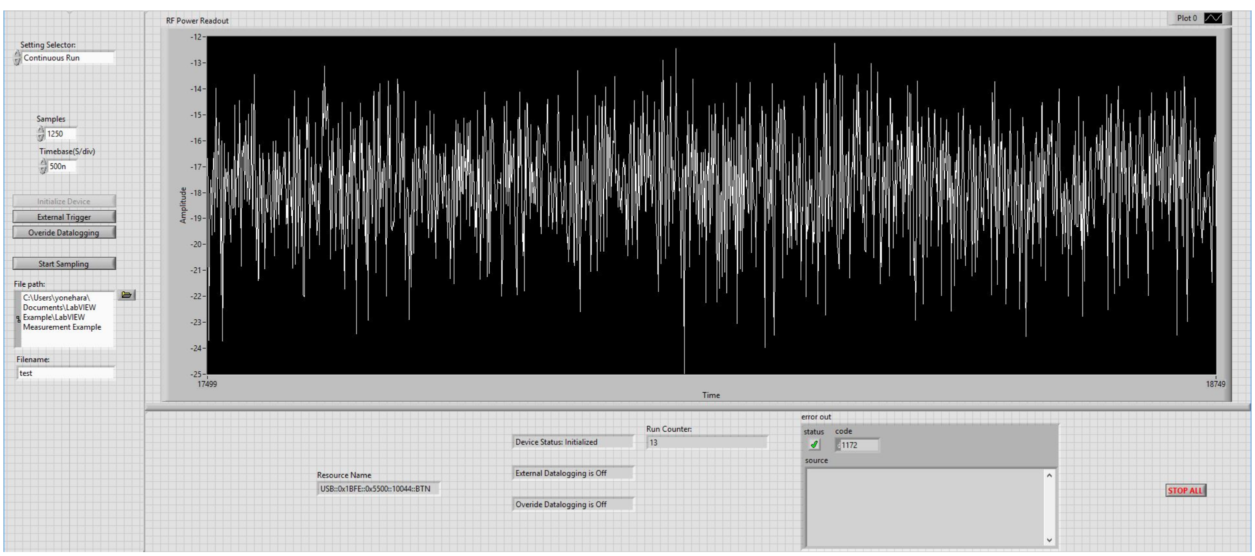

Figure 3. GUI Front End

2.2 Principles of Backend of Data Acquisition System

One of the most important things to understand in the design of a DAQ system is the data flow process. From a hierarchical perspective, there are three main parts to the DAQ system: the Configure Loop, the Initialize Loop, and the DAQ loop. First, the Configure Loop, pictured in Figure 4, takes all of the device setting inputs from the front panel and then feeds them to the Initialize Loop when specified by the user. The Configure Loop also resets all of the program flags to their initial default values such as the STOP ALL button and chart properties important for displaying the measured waveform. Finally, the Configure Loop precomplies some of the DAQ settings such as the file path. It compiles some of the file path that is constant instead of having to recompile every time, increasing computational efficiency ever so slightly. In terms of the system and timescale we are concerned with, it is best to not be wasteful with the computational speeds unless otherwise necessary. The Initialize Device or the STOP ALL buttons are the exit conditions for this loop.

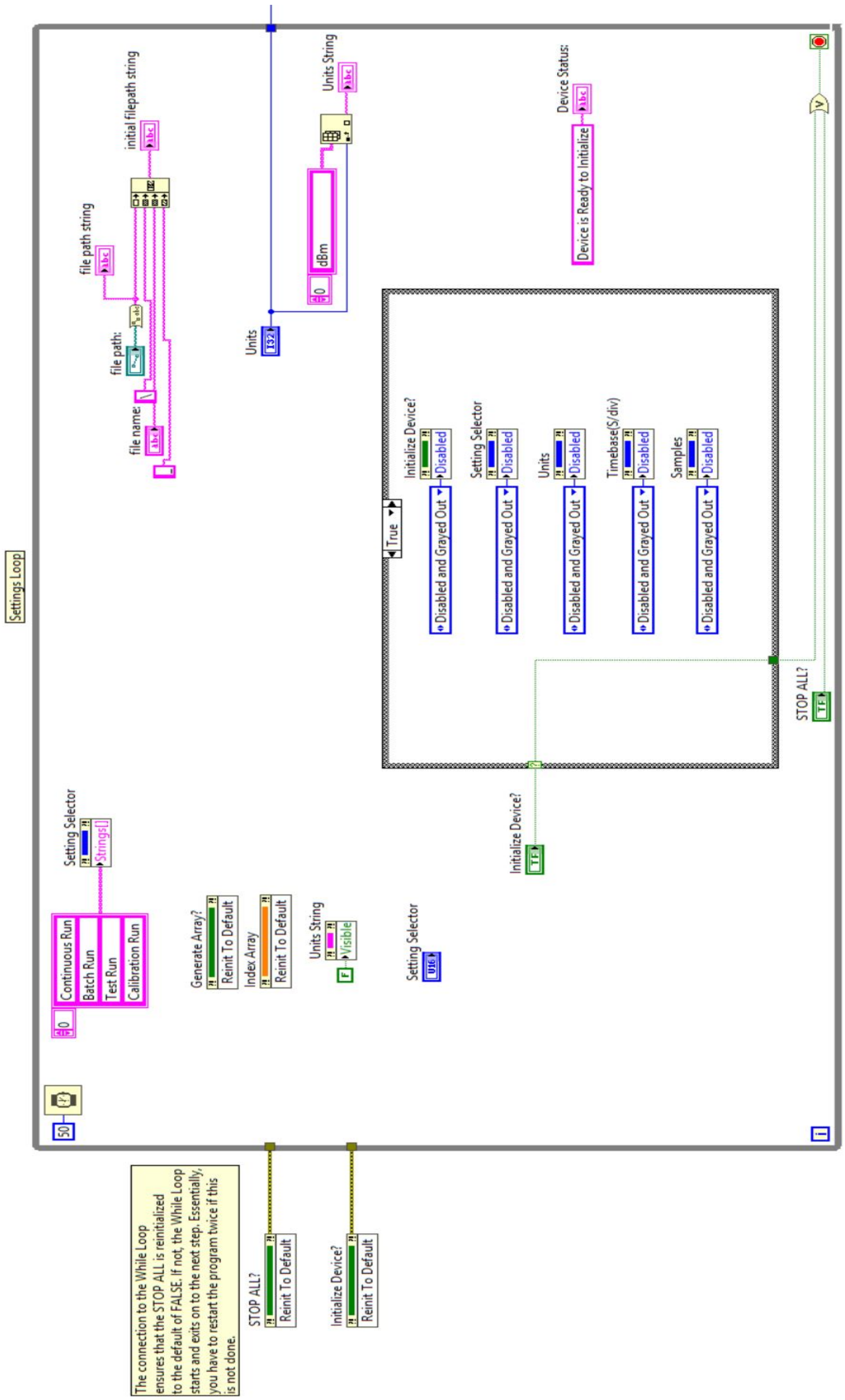

Figure 4. Configure Loop

The next loop is the Initialize Loop, pictured in Figure 5, which first interacts with the sensor through the use of the provides RTP LabVIEW driver. The communication with the sensor is opened up in the first section of the code, denoted by the left top of Figure 5. The Initialize Loop is also where the sensor settings are sent so the device is initialized with these sensing parameters. The second section, denoted by the top center of Figure 5, configures the device with the settings specified from the front end. The Initialize Loop also generates an array based on the users choice of timebase from the front end. The array generation procedure can be seen in the bottom center of Figure 5. The user's choice of timebase greatly affects the time per point. Based on the user's selection of this timebase, a time index array is generated by multiplying by the time per point. This is constant throughout the entirety of the run time. If it is not, then an error occurs, which stops the program. This allows for clear and concise time domain analysis to take place. Otherwise, the program would just be indexing the sample number. It is important that the Initialize Loop runs only once and runs without error, otherwise, an error will be raised and the DAQ Loop is useless. To deal with this problem, the exit condition of this loop is whether the error handler, pictured in the top right of Figure 5, raises an error. If an error is raised, then the program will stop entirely and notify the user that an error has occurred. Part of LabVIEW's functionality is that it can run multiple things pieces of code in parallel. This can cause trouble when you are having to call on values of input parameters that are specified by the user. To mitigate this, the input parameters to the device are fed only after the initialized loop is safely exited. Then, the Initialize Loop takes these inputs and feeds them to the necessary configuration VIs. Because the Initialize Loop is waiting on these values, the while loop does not run until it has received these values. The while loop is utilized instead of a case structure because there are some problems experienced with the device driver VIs and case structures. Although the full problem is not fully understood yet, the problem seemingly stems from the utility of shift registers for while loops in LabVIEW. These allow for the passing of error and configuration data to the RF power meter driver VIs. Case structures do not have the same functionality. For example, if a case structure containing one of the driver VIs is false, then the data is not passed onto the next driver VI, which causes an error.

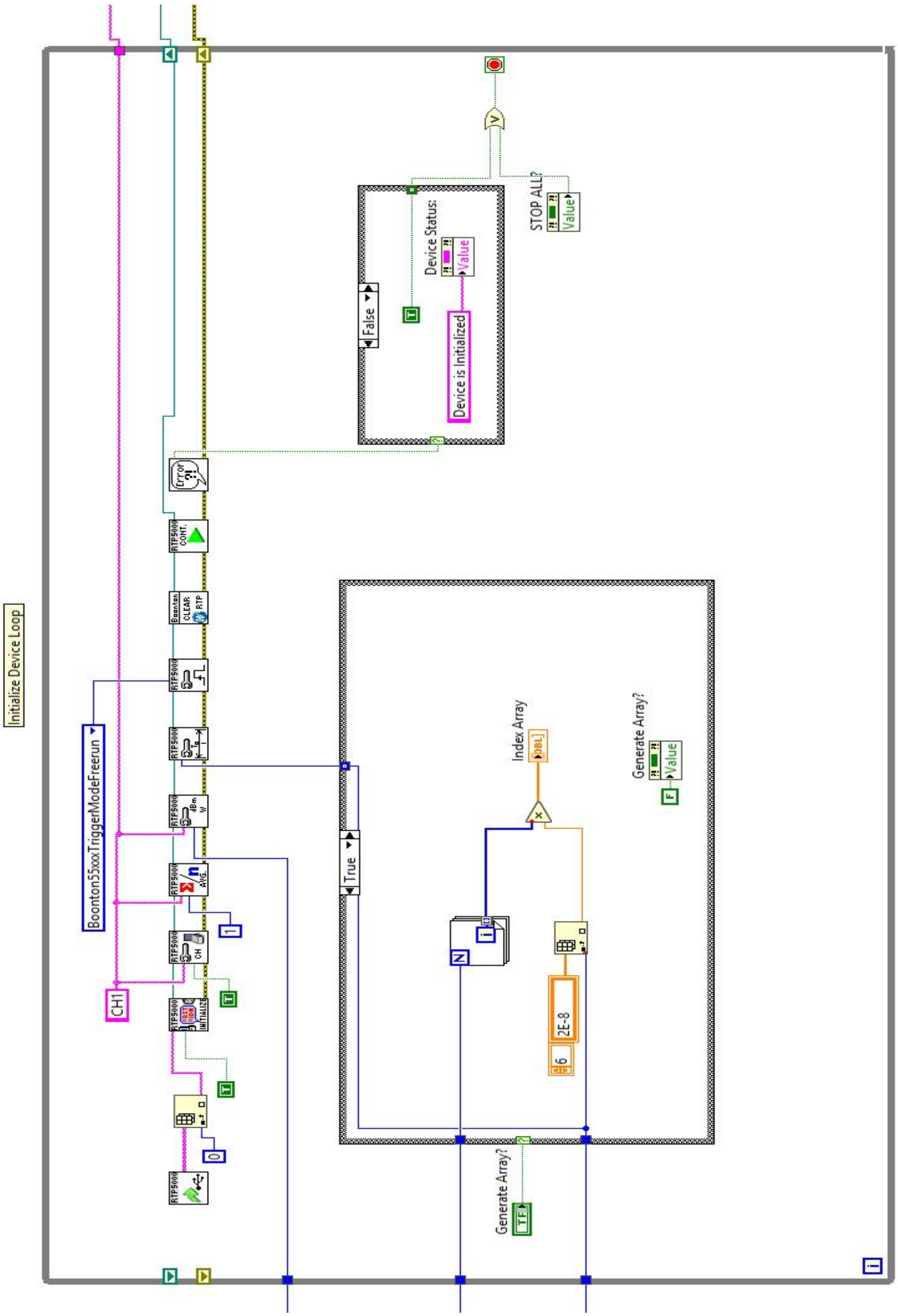

Figure 5. Initialize Loop

The final loop is the DAQ Loop, shown in Figure 6. This is the most important loop of the program as this is what does the most work. When the device is done initializing, the DAQ Loop first checks for the external trigger in the trigger loop. When the device receives the signal from the main control room, it exits the trigger loop and enters the fetch loop. The fetch loop reads the waveform from the buffer and displays it on the chart. The waveform returned is determined by the initial parameters specified in the Configure Loop. Once it has taken this measurement, a check is done to ensure that all of the elements in the waveform array are valid. An example of an invalid measurement is an element with a not a number (NaN) value. If the array passes this test, the time index array is appended to the sample array and then saved to a timestamped csv file. The timestamp is generated when the trigger starts. The time index array are measured relative to this time stamp. The timestamp is generated and concatenated to the constant part of the file path string. The data array is transposed when saved in order to have to columns; the first being the time index array and the second being the RF power array.

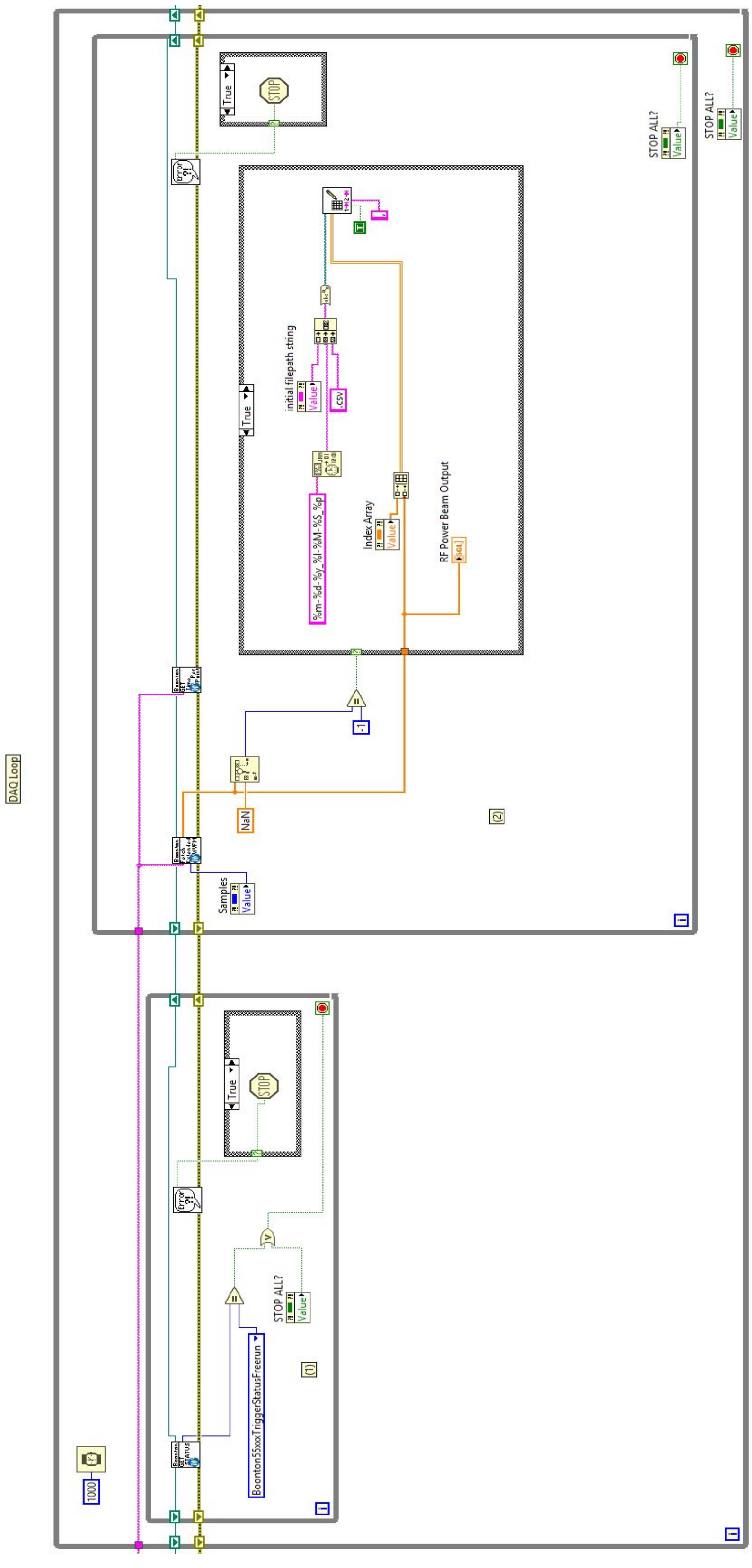

Figure 6. DAQ Loop

After the DAQ system has finished running and collecting samples, some final processes need to be run. First, the connection to the sensor needs to be closed. If this is not done then the next time the program is run, it will raise an error. If there is no error in closing the connection to the sensor, then the user is queried if they would like to start data processing. If yes is selected, then some Python functions can be called straight from LabVIEW in order to process the data collected. An example of how this looks is presented in Figure 7. The lone Python function implemented currently can go in and read all of the csv files saved and generate plots of each of them. The inputs to this function are fed from LabVIEW. The first is a Boolean input of true to start data processing. The second input is the path string and the final two are the units used. A chart is then generated and saved as a PDF although this can be customized. This automates an otherwise manual task. Since it can also be run straight from LabVIEW, it limits the amount of interaction the user has with the backend. The user also does not need to keep track of a Python script that does this since this is called directly from LabVIEW. This analysis is currently limited, but provides the opportunity for expansion.

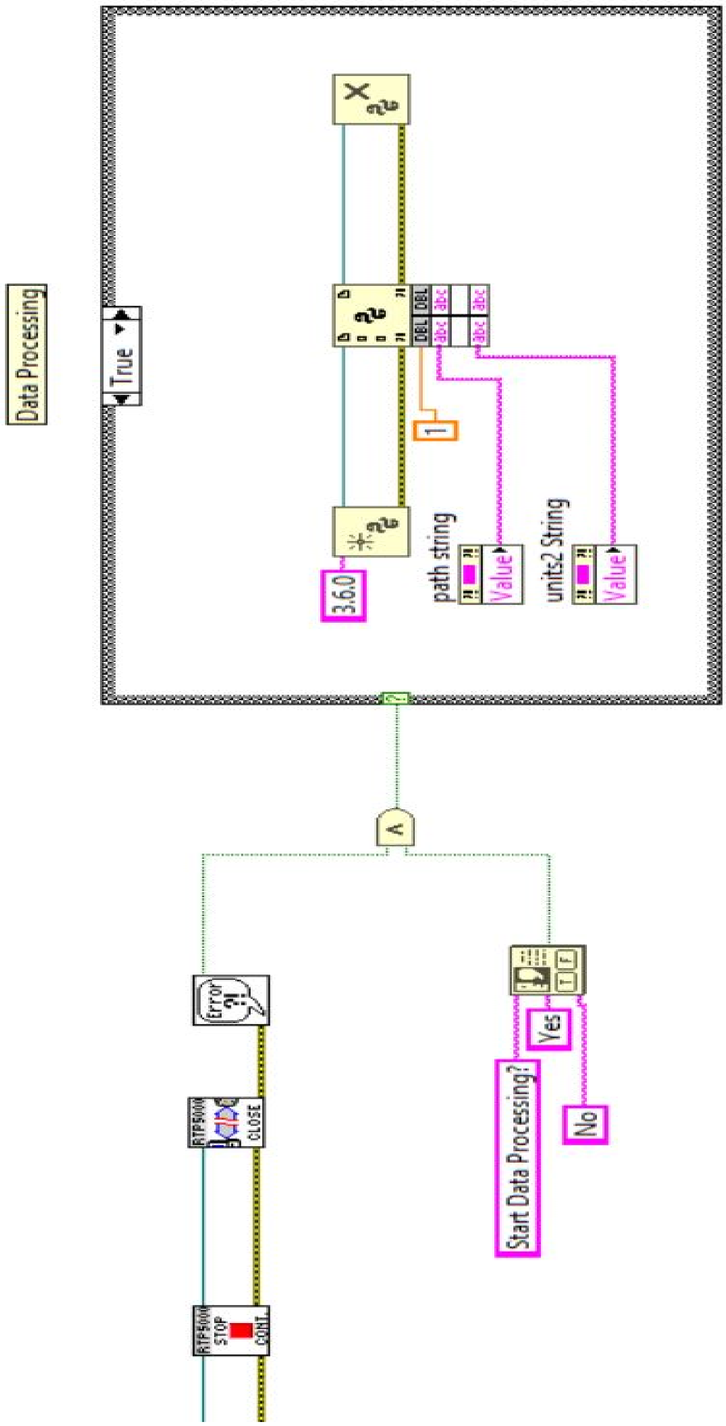

Figure 7. Data Processing

3 Benchtop Setup and Testing Procedure

An important aspect of developing a DAQ system is running the system on a benchtop setup. This benchtop setup, shown in Figure 8, is comprised of four tools used to generate an RF noise and supply it to the computer. The first is a power supply which powers the RF noise generator. The power supply is set to 15.1V at 1.71A in order to deliver a noise signal with a power of about 15-16dBm. The second tool is the RF noise generator itself, which supply a noise signal to the third tool, the Boonton RTP 5006 RF power meter. The RF power meter stores data points to its buffer and then sends those measured points over USB when requested by the DAQ program. The fourth and final tool in the benchtop setup is the DAQ system itself, which is just a laptop computer. As mentioned previously, the DAQ reads a trigger and then reads the waveform from the buffer. The buffer is used in order to bypass the lower data transfer rate of the USB port. A simple flow diagram of the entire setup can be seen in Figure 9.

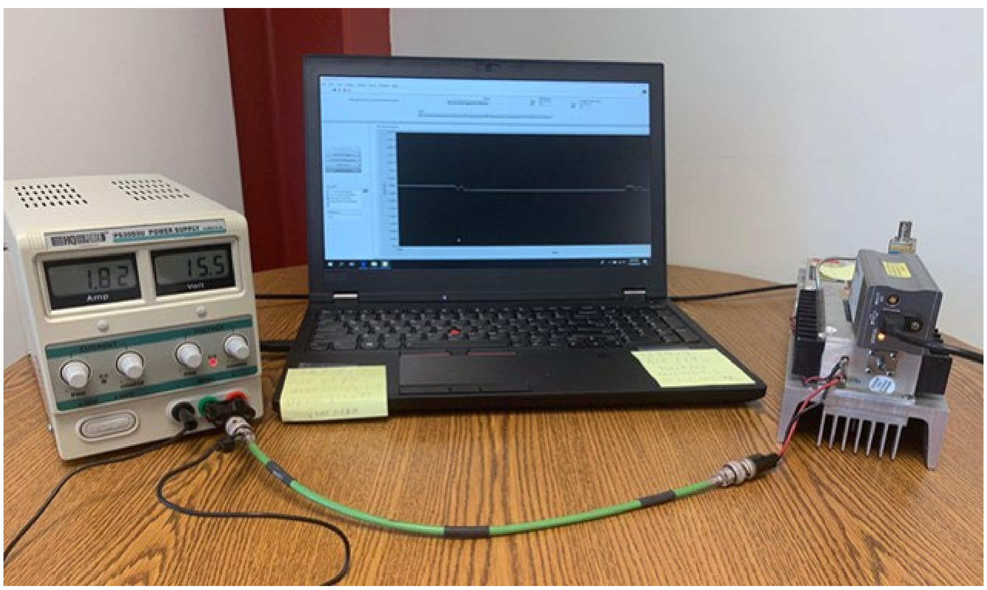

Figure 8. Benchtop DAQ Test Setup

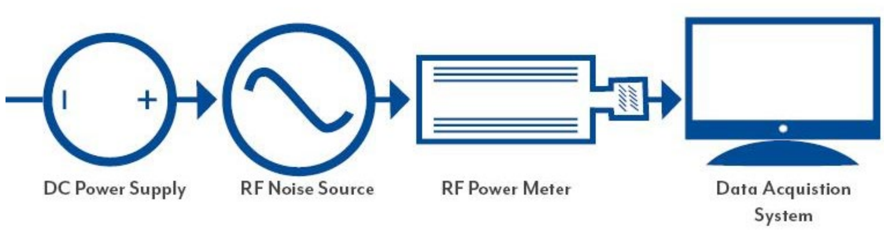

Figure 9. Benchtop Test Setup Process Diagram

4 Future Works

Time constraints limited the ability to test the full functionality of the current DAQ program. For example, a TTL signal has yet to be generated and sent to the device. Once this has been tested, then the next step is to actually tune to device to the accelerator clock. On top of this, the future of the DAQ system sees many different upgrades to functionality, primarily a focus on control. One of the strengths of an RF cavity hadron monitor is its ability to calibrate itself after each run. As the device encounters more and more radiation damage, the strength of the signal decreases. The recalibration of the device during each beam spill helps maintain a linear system for a longer time. This is something that the current hadron monitors cannot provide. In order to implement a system that does this, the program must adapt to not only collect and save data, but to control some parameters that affect this data. For example, to calibrate the system during the beam spill, it needs to control the RF source. After the beam spill occurs, the RF cavity must discharge and then have its quality factor recalibrated. This is done by measuring the rise and fall times of the cavity without the interaction of the beam. Another aspect of the system that needs to be controlled is the frequency at which the cavity is powered. As the cavity interacts more and more with the beam, the plasma created from the ionization of the gas could raise the temperature, morphing the geometry and causing a change in resonance. It is important to have the device operating at resonance so the cavity absorbs the most amount of energy possible.

The actual hadron monitor system also sees the expansion from one test monitor to an array of monitors. Some off-center studies of the cavity are needed in order to simulate what the full array of cavities might see. This is in place of introducing multiple cavities to the DAQ system, meaning more variables to juggle. The array of cavities is utilized to help determine properties of the beam such as the particle distribution and centroid. A certain level of understanding of the cavity properties must be attained in order to build up to an array of multiple cavities.

5 Conclusion

The development of the next generation of high intensity accelerator technology calls from more robust sensors in terms of radiation. The current generation of hadron monitor technology is sensitive to radiation damage and will inevitably get worse has more intense beams are developed. The technology monitoring the beam must keep up with the intensity of beams. To this end, a rad-hard gas-filled RF cavity hadron monitor is currently being developed. In an effort to automate the data acquisition process, a DAQ system is developed in LabVIEW using the Boonton RTP 5006 RF power meter. The current status of the DAQ system reads the waveform from the buffer when triggered by the main control room. This expedites the development process by increasing the amount of data to study. The DAQ system is developed modulary in order to allow for adaptability when it comes to the future needs of control along with data acquisition.

7 References

[1] R. Zwaska, “Accelerator Systems and Instrumentation for the NuMI Neutrino Beam”, Ph.D. Thesis, 2005, University of Texas at Austin, FERMILAB-THESIS-2005-73.

[2] K. Yonehara et al., “Radiation Robust RF Gas Beam Detector R&D for Intensity Frontier Experiments”, Proc. 10th Int. Particle Accelerator Conf. (IPAC2019), Melbourne, Australia, June 24, 2019, pp. 2770-2772, doi: 10.8429/JACoW-IPAC2019-WEPGW115.