Data Acquistion & Quantum Efficiency

Erick Orozco

Enrico Fermi Institute, University of Chicago

August 17, 2018

Abstract

Large-Area Picosecond Photo-Detectors (LAPPDs) are being developed to detect photons with extremely high resolution. The process in developing a LAPPD is complex and nontrivial, but requires some form of a structure to be repeatable. The developmental stages need to be moderated constantly and are required to be testable throughout. A serviceable data acquistion system must be in place to assist in moderating the lab by storing data from a multitude of sources that are within the lab. Alongside this, the instrumentation for testing the LAPPDs Quantum Efficiency(QE) must allow for consistent, repeatable testing methods to the most precise measurement possible.

1 Introduction

Our lab, under Professor Henry Frisch, is currently developing LAPPDs. We split the work in the lab into many different projects due to the complexity of the entire project. Some of these tasks in the lab include developing a cesium detector, finding which materials would be best to utilize, and much more. This paper will touch upon two projects (DAQ and QE) that helped in the process of the main objective of our lab.

2 Data Acquisition (DAQ)

The DAQ project's purpose was to create a more serviceable system compared to the former DAQ system. The former project was a simple system of Raspberry Pis utilizing Python code to send data to a main database system located in PSEC2, our main computer. It was said to be messy, and on occasion the Raspberry Pis would terminate the Python code for unknown reasons, which left many confused. In fact, no one would be able to know if the Pyhthon code was terminated unless someone opened an SSH to that Pi and see what processes were running. This caused issues considering that most Python code ran at all points in time, gathering data all across the lab to help moderate the environment of the lab.

Knowing this, I was assigned to improve this system in a specific manner. The group desired to keep their former Python code that acquired data, but wanted a new method to send data from a Raspberry Pi to PSEC2, our main computer. They advised me to utilize Node-RED, a tool developed by IBM that assists in the wireless interconnectivity of the devices.

2.1 Implementing Node-RED

When first starting on Node-RED, no one in our group knew how to use or implement it, including our Graduate Student and Professor. This left Node-RED to be somewhat of a daunting task at first. I did have programming experience prior to coming to Chicago, but this was different.

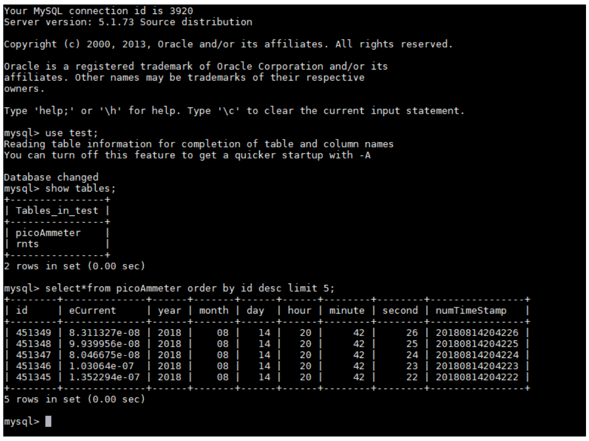
Initially, I was told to use the Message Queuing Telemetry Transport (MQTT) protocal for sending informaton from Pi to PSEC2. This MQTT protocal, also developed by an IBM employee, allows for sending data wirelessly with great security and little memory usage. It requires a host site, some require to be bought or to have a subscription, but shiftr.io was advised to use due to shiftr.io's easy use and it also happened to be free. The inital scheme was for the Pis to send data to other Pis, or to PSEC2, using the MQTT protocal utilizing shiftr.io. This system was operational and simple. However, once I gained the knowledge on how these processes work and once they were put into place, we were steered away from using any MQTT work due to another need for the group.

2.2 MySQL

A major issue of the former DAQ system was that it was quite complicated to retrieve older data. Along with this complication, the database archived all data into text files, separated by commas similar to a csv filetype. This created what would seem like many issues in the forseeable future. Problems were already occuring with the attempts to access old data in a simple manner.

We were then advised to utilize MySQL, an open-source database management software that is largely used in corporate environments, most commonly for backing up data from websites. In our case, we used it to store information measured in the lab.

MySQL solves all database management complications. This software compresses all files, therefore reducing memory usage significantly. MySQL also allows for simple access to older data stored in databases. With all these positives aside, I believe the best take away from switching over to MySQL is the simplicity of the software's commands. As you can see in Figure 1, the commands are quite simple.



```
Your MySQL connection id is 3920
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use test;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+----------------+
| Tables_in_test |
+----------------+
| picoAmmeter    |
| rnts           |
+----------------+
2 rows in set (0.00 sec)

mysql> select*from picoAmmeter order by id desc limit 5;
+--------+--------------+------+-------+------+------+--------+--------+----------------+
| id     | eCurrent     | year | month | day  | hour | minute | second | numTimeStamp   |
+--------+--------------+------+-------+------+------+--------+--------+----------------+
| 451349 | 8.311327e-08 | 2018 |    08 |   14 |   20 |     42 |     26 | 20180814204226 |
| 451348 | 9.939956e-08 | 2018 |    08 |   14 |   20 |     42 |     25 | 20180814204225 |
| 451347 | 8.046675e-08 | 2018 |    08 |   14 |   20 |     42 |     24 | 20180814204224 |
| 451346 | 1.03064e-07  | 2018 |    08 |   14 |   20 |     42 |     23 | 20180814204223 |
| 451345 | 1.352294e-07 | 2018 |    08 |   14 |   20 |     42 |     22 | 20180814204222 |
+--------+--------------+------+-------+------+------+--------+--------+----------------+
5 rows in set (0.00 sec)

mysql> █
```

Figure 1: Example of simple, easy workflow in MySQL

The process of accessing specific data from a table is as follows. First we ask to change into our specific database titled 'test'. Then we check to see what tables are in this specfic database. Finally, we check for the 5 most recent rows in the table titled 'picoAmmeter'. One can also easily access any range interval of data that has been recorded by simply stating the interval of timestamps one wants to see, allowing for quick access to older data.

2.3 MySQL in Node-RED

MySQL is impressive, but Node-RED's connectivity options with MySQL is the reason we don't have to utilize any MQTT protocal messages. There happens to be a package you can install on Node-RED that allows users to send query commands to a MySQL database. Installing this package titled 'node-red-node-mysql' introduces a new function within Node-RED that will wirelessly commit a query command that a user specifies.

Using this function allowed for easy use with inputting data into a database and also retrieving any data all without the need for MQTT protocal messaging. In fact, I was able to create a few functions such as a live-plotter which updated its plot in intervals of one second. I was also able to plot any interval of older data by specifying the timestamp information needed. Once understanding MySQL query commands anyone can easily plot any data they desire in whichever way they would like to. I believe this to be the true triumph with using Node-RED.

2.4 Limitations

Complications do arrise with our new DAQ system. I cannot blame MySQL whatsoever, the software does its job exceptionally well. All issues stem from Node-RED. It seems that Node-RED is moreso a hobbyist technology than a big data tool. Whenever trying to plot more than 50 data points, it would crash. In fact, there is some noticeable latency with just 50 points being plotted. According to the lab group's goal, this is less than ideal. We would like to plot more than 50 points, in some cases we would like to plot as many as 3000 points. I believe there is a work around within Node-RED for this issue, but I did not have sufficient time to get to an answer.

2.5 DAQ Summary

The DAQ system I have implemented has a few pros and cons when compared to the former system and I still implemented much of the code from the former system. For example, the Picoammetter in the lab is physically connected to a Pi and the former Python code that initialized the PicoAmmeter and sent data to PSEC2 was still used, just modified. The initializing portion of the Python script has little or no change but the data pushing portion of the script was modified to push data to a specific table in PSEC2's MySQL database. After this, Node-RED utilizes the installed MySQL function to pull data from that specific MySQL table and then plotting the data in whichever way is preferred. Figure 2 shows the general workflow of the system.

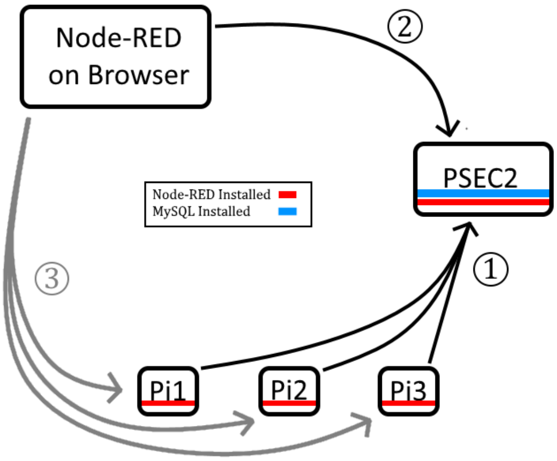
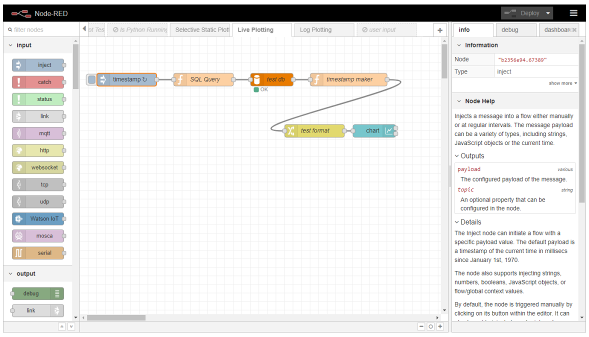


Figure 2: DAQ Workflow

With ①, the data collected by the Pis directly get sent to a MySQL database in PSEC2 via a Python program. In ② an internet browser can be opened on a personal computer and then point said computer to PSEC2 via Node-RED. ③ is a separate option to replace ①, which is why it's gray colored, not black. In ③ a personal computer points to any of the Pis. Each Pi will have its own Node-RED on separate tabs if you wish to run more than one. This way is a bit slower but may be easier to fix due to Node-RED's interface. However, it requires more steps to set up and contains more moving pieces, which makes it slower and possibly a bit more frustrating to work with. I recommend ① which does not require the Pis to have Node-RED but you can use whichever method you find more appropriate. Whether you use ① or ③, the ② step is the same process.



Information

Node	"b2356e94.67389"
Type	inject

show more

Node Help

Injects a message into a flow either manually or at regular intervals. The message payload can be a variety of types, including strings, JavaScript objects or the current time.

Outputs

payload various
The configured payload of the message.

topic string
An optional property that can be configured in the node.

Details

The Inject node can initiate a flow with a specific payload value. The default payload is a timestamp of the current time in millisecs since January 1st, 1970.

The node also supports injecting strings, numbers, booleans, JavaScript objects, or flow/global context values.

By default, the node is triggered manually by clicking on its button within the editor. It can

Figure 3: Node-RED flow for liveplotting

In Figure 3 we see Node-RED's interface. We see a program that shows a flow of 6 functions. It starts with a blue 'inject' function that is titles "timestamp" that initiates the program every second. The second function titled "SQL Query" sets up a query command to retrieve a specific amount of the most recent rows of data to send to the MySQL function, which is next. Once the MySQL function receives the command, it will send out those specific rows that were requested. The following two functions exist to set up those rows to the correct format for plotting on Node-RED's "chart" function, which is the final function in this program. Given a certain website, which is just '[Pi IP Address]:1880/ui/#/0/'.

In Figure 4 we see how the plot looks like. This plot updates itself every second, as the program is launched every second and the PicoAmmeter send data to a MySQL table every second.

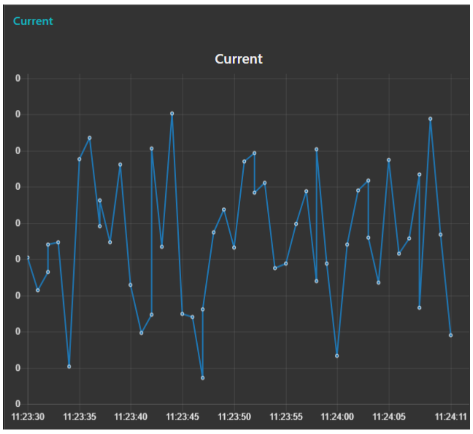


Figure 4: Node-RED plot

We are able to get a plot that looks nice but it comes at a cetain cost. Below in Table 1 I will state what I believe are the benefits and disadvantages of this newer DAQ system.

Pros	Cons
memory usage reduction	no axis-labels
quicker and simpler access to any stored data	cannot handle as much data at once for plotting as desired
easier to navigate through program for newcomers	usage of many programming languages rather than one centralized language
easier to track mistakes	

Table 1: Pros and cons of new DAQ system

Many of the positives are due to MySQL, not Node-RED. In fact, the limitations of Node-RED are not too much of a complication. It's only one issue that outweighs everything, which is that it cannot handle large amounts of data at once to plot. Therefore, I believe Node-RED will be scarcely used in the lab moving forward, but MySQL should be heavily relied since it has no weaknesses when compared to the older DAQ system.

3 Quantum Efficiency (QE)

In this LAPPD project, being able to test the device in a repeatable, reliable fashion is of utmost importance. Quantum Efficiency (QE) is the main quantity that is tested once development is done for a LAPPD. QE is the ratio of the number of electrons out of a system per number of photons into a system, as shown in equation 1.

$$QE = \frac{N_e}{N_\gamma} \tag{1}$$

To test our QE system, we will be using a Photomultiplier Tube (PMT) in place of a LAPPD. The understanding of the PMT will assist in the understanding of how the LAPPD works.

3.1 Instrumentation

The environment of how we test QE is of great importance. Luckily, it was not too difficult to set up. More than anything, the limitation of outside light is of great importance due to the PMT's extremely high sensitivty to all light. To be able to know the fixed amount of light hitting the lens is necessary. The lab already had a dark box, a black large structure that has plenty of space for a PMT inside. The dark box also has SHV and BNC jacks so the PMT can be powered and send out its current without needing to leave the box open, as seen in Figure 5.

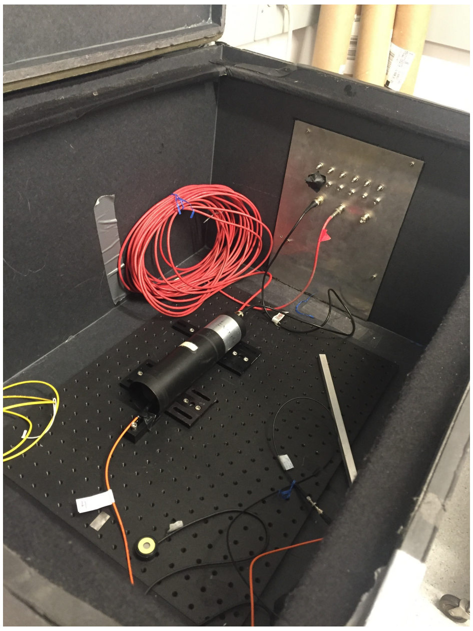


Figure 5: Inside the Dark Box

As seen, there are also little bases screwed to the floor of the dark box with many possible cofirgurations. This is to help the PMT not roll around and be firmly in place for all measurments.

The orange cable at the bottom of Figure 5 is an optical fiber, which is extremely sensitive to where and how it is placed in an environment. In this instrumentation, it is more or less firm in place. Overall, it will be rare for it to move inside the dark box due to it being fastened to a base plate in the system. Once placed inside, the use of a Thorlabs PM100A is to measure the intensity of light from our diode. Both our diode source and intensity reader can be seen in Figure 6 as the grey box with black knobs and the red outlined device on top of the diode source, respectively. This is all located under the dark box.

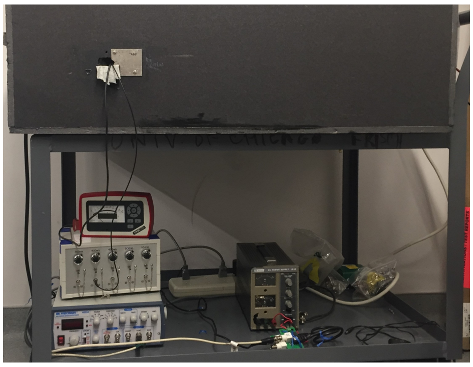


Figure 6: Under the Dark Box

With this, measuring the intensity of the diode prior to placing the PMT is important since the measurment of the PMT in action and measuring the intensity of light cannot be done simultaneously. The distance between the sensor from the optical fiber end should also match the distance between the PMT lens and the optical fiber end to ensure accuracy.

In Figure 7 we see more of the instrumentation. This is located right next to the dark box, to the right and out of the picture in Figure 6. To the left in Figure 7 is our PicoAmmeter, which has our Raspberry Pi attached on top. To the right we see three devices, with only one we need to discuss. The bottom device is our high voltaege supply. We have since replaced this high voltage supply, but I only worked with the one pictured.

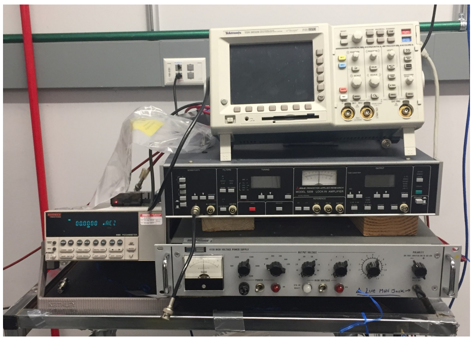


Figure 7: PicoAmmeter and High Voltage Supply

3.2 Modifying the PMT

PMTs are quite difficult to work with for QE. The number of electrons out of the PMT is multiplied inside the device by the use of dynodes and the process of secondary emission. Electrons do not have to follow the same path as previous electrons, so the multiplication factor is not consistent. Therefore, an analysis of the circuit for the PMT was made.

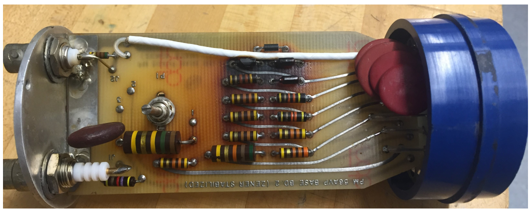


Figure 8: Circuit for Photomultiplier Tube

In Figure 8 we see that the circuit utilizes voltage dividers to supply varying voltages to each dynode. Each flow of current is connected to a pin, so analyzing the pins was the next logical step. After some tests with an ohmmeter the pins connected to the dynodes or photcathode were revealed.

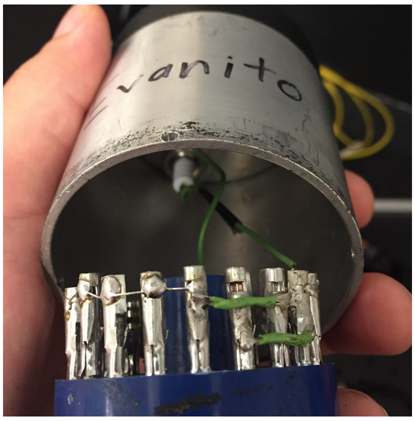


Figure 9: Modified PMT pins

Figure 9 shows the modifcations made. Here we have two pins in series connected to the SHV on one end and both pins are internally connected to the photocathode on the other end. All other pins are connected in series to the BNC jack on one end and are connected internally to all dynodes on the other end. With this, now all the dynodes collect electrons and push them out to the BNC jack rather than multiplying electrons by secondary emission. Now a crude number of electrons leaving the PMT is made and a QE calculation can be made.

3.3 PMT Tests

Given that all instrumentation is now set up, I conducted a few tests. First, I checked the voltage dependence of the PMT by varying the voltage while keeping the intensity of light fixed. Figure 10 the varying voltage can be seen. At first it acts somewhat linear, but as the voltage increases it begins to plateu at about 100 volts. I chose to work with 200 volts for all other tests due to me feeling that it was the most stable, lowest value of voltage based on these data points.

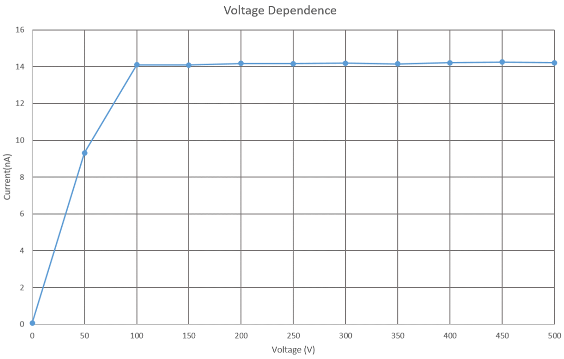


Figure 10: Voltage vs Current

Continuing on, I then worked with a QE test. Given the equations 2 & 3 I was able to figure out the number of photons and electrons with some mathematical manipulations.

$$I_v = \frac{N_\gamma h\nu}{A\lambda t} \tag{2}$$

$$QE = \frac{Q}{t} \tag{3}$$

I then conducted a QE test with varying intensities of light as shown in Figure 11. As can be seen, QE is more or less stable for decent stretch of light intensity, then it falls somewhat linearly. This can be attributed to the number of photons being too large when hitting the photocathode causing the electrons of the photocathode to not get back into the stream of photons quick enough, therefore less electrons get excited.

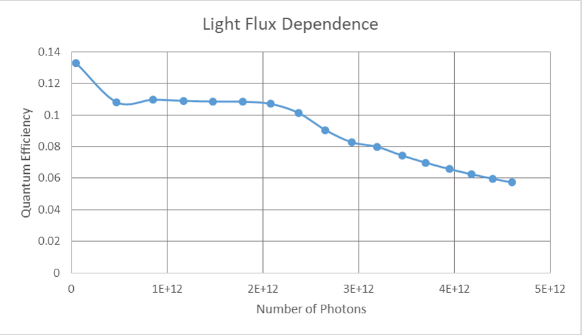


Figure 11: QE vs Number of Photons

3.4 QE Summary

In general, all these tests for the PMT served only one purpose: for me to learn how QE works. The information the PMT offers us is nice but not very useful. The truly useful portion of this entire project is the instrumentation. I have assisted in commissioning a QE measurement in which a LAPPD can now replace the PMTs place and measure QE for our then developed product. Hopefully, when I leave and they soon put a LAPPD into this sytem, it works smoothly.

4 Acknowledgements

I would like to thank the Professor I worked under, Henry Frisch, for presenting me with the opportunity to work at the University of Chicago's Enrico Fermi Institute. I would like to thank the entire PSEC group, without their assistance and support this work would not be possible. Thank you to all REU organizers, participants, and the NSF for all funding. And a final thank you to my parents, if it was not for their sacrifices I would be

nowhere near as successful in my academic career as I am today.

References

[1] Eliza Gazda. *Summer 2017 REU Report.* Chicago, IL, 2017. `http://lappddocs.uchicago.edu/documents/317/sendit`