.BEE Connected

Ryan Wallace

University of Illinois

Table of Contents:

I. Introduction

- Past work summarized

  Before diving into the contents of this study I feel it is necessary to briefly summarize my past work. The goal of my previous study was to obtain audio data from within an active beehive. After a prototype was built the device was placed into an active beehive and audio data was taken from the hive.

  The devices used in the previous study required hands on control from a user that connected to the audio capture device directly with a laptop or pc. This presented a few obvious issues. Every time audio data was to be obtained from a hive a user would need to directly connect an external ADC to the microphone circuit and user's pc. The device used in the previous study to capture the analog signal was the National Instruments MyDAQ, the size limitations made keeping this device within the hive difficult and was eventually ruled out with the idea of using a smaller device in the future.

  My previous study report can be viewed here, and would be beneficial for anyone interested in audio data retrieval and overcoming some of the obstacles faced when placing objects into an active beehive.

- Problem description

  Implement a way in which audio data from the hive can be transferred via wifi, and stored online for later analysis. The need for user interaction in order to obtain the data would be greatly reduced as opposed to the previous study. In addition to sending audio data, the ability to alert a user based off of certain frequencies sensed within the hive to help determine a possible swarm of the colony.[1]

[1]Swarming refers to a colony of bees splitting into two, in which the queen bee leaves the current hive (or area in which the bees have colonized) with approximately half of the colony. There are numerous theories as to why swarming occurs, some speculate a colony that is poorly managed in which food or space become an issue may cause bees to swarm. I encourage you to read into this topic as it is its own field of interesting research. There are though warning signs that take place prior to a swarm that can help prevent or prepare for a swarm. One such warning sign that pertains to this research is bees emitting a certain frequency prior to swarming. (Sanford & Bonney, 89)

- Technical design concept

  The initial challenge was choosing the components that would meet the requirements of the project. This meant choosing or building a device that was capable of WiFi connection, could be interfaced with the microphone circuit and perform the analog-to-digital conversion. The entire setup would need to fit in a housing similar to the existing device box and run off of battery power as well.

  I decided on a microcontroller available from Texas Instruments based on their MSP430 low power line, in conjunction with an attachable WiFi booster pack. The MSP430 microcontroller I chose is equipped with a 12 bit 200kS/s ADC which well exceeds the necessary sample rate. The device is all being powered by two triple A batteries. In addition to this the footprint of the microcontroller was 58mm by 74mm making it feasible to fit within the hive in a constructed housing built very similar to the original design.

- Wireless connection design concept

  The final design choice was to send the audio data to a space in the cloud or something similar where it could be analyzed. I chose to work with Node-RED as my interface between data and storage due to its intuitive GUI and customizable actions. As for storing data, I am using Node-RED along with Google Sheets & Google Forms to write the data into a Google spreadsheet.

The following is a table of items used:

Item	Value (if applicable)	Quantity
Electret Microphone (CMA-4544PF-W)		1
TI WiFi Boosterpack (CC3100BOOST)		1
TI LaunchPad (MSP-EXP430F5529LP)		1
Phenolic Perfboard (38mm x 100mm)		1
Resistor	10 Ω	1
Resistor	70 Ω	1
Resistor	390 Ω	2
Resistor	10k Ω	5
Resistor	100k Ω	1
Capacitor (Ceramic)	0.1 µF	6
Capacitor (Ceramic)	1 µF	1
Capacitor (Electrolytic)	10 µF	2
Capacitor (Electrolytic)	100 µF	2
Red LED		1
Transistor (MPS2222A)		1
Audio Power Amplifier (LM386N-3)		1
Potentiometer (3339P)	10k Ω	1
2.4 GHz PCB Antenna (USR-ANT2G5GS001)		1
Triple-A Battery		2

Table 1: Items List For Build/Circuit

II. Analysis of components

Electret Condenser Microphone (CMA-4544PF-W):
An omnidirectional microphone with a relatively flat frequency response. Compact in size, with a diameter of 9.7mm and a depth of 4.5mm (∓0.2mm)

TI WiFi Boosterpack (CC3100BOOST):
Used to connect the MSP430 microcontroller to the internet via WiFi. Connects to the microcontroller via a set of pins designated for the pairing of the two components. An antenna (USR-ANT2G5GS001) was added to the component to increase signal strength.

TI LaunchPad (MSP-EXP430F5529LP):
Low power microcontroller used in acquiring audio data from the microphone circuit. The audio signal is sampled via the 12 bit ADC of this device and connected with the above mentioned WiFi boosterpack will send the data to a client.

Characterization of MSP430 Microcontroller:
For simplicity I will refer to this microcontroller as the LaunchPad throughout the remainder of this text. The LaunchPad has a total of 42 pins with which can be utilized in a variety of ways. For simplicity on myself and the reader I will cover only the pins utilized in this particular project.

The terminals used from the TI MSP430 are shown in the table below:

Function	**Terminal**
5 Volt Power Supply (for mic. circuit)	5V
3 Volt Power Supply (for mic. circuit)	3V3
Ground	GND
3 Volt Battery Supply	3V3
Analog Input Channel	P6.1
Analog Input Channel	P1.0
Digital Output	P6.2

Table 2: TI MSP430F5529 terminals used in this project.

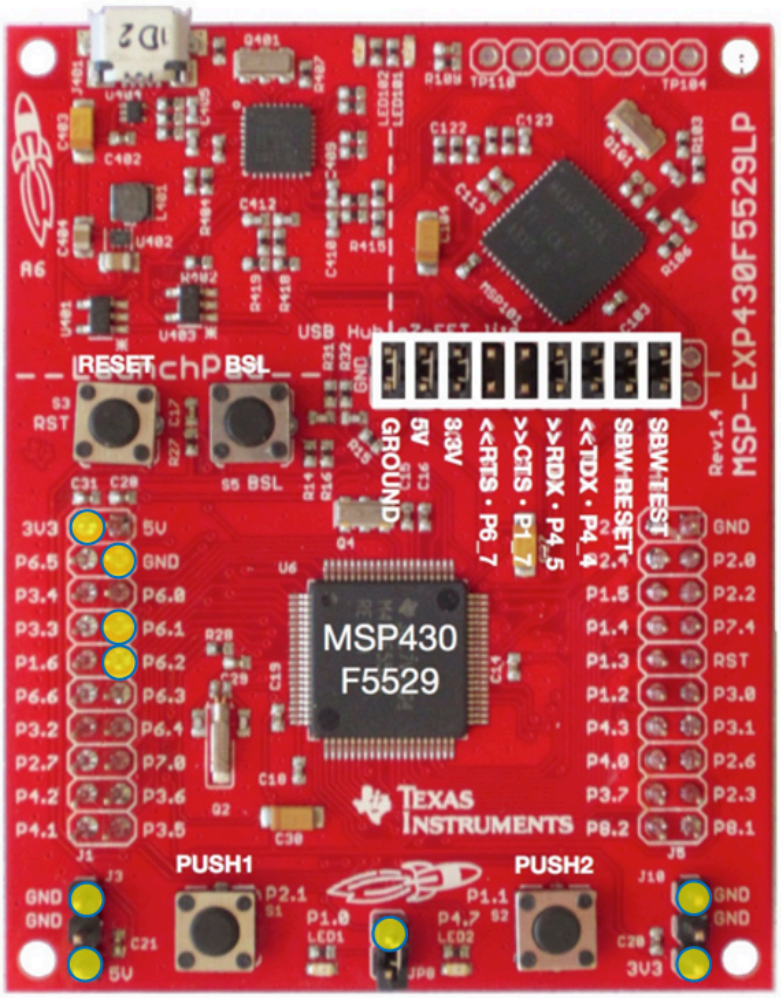

Figure 1: TI MSP430F5529 reference terminal connections.

Please note that the table on the previous page does not list the quantity of ground pins, nor does it list which 3V3 connection to make. These are based solely on design choices. Although the connections shown above reflect my exact pin terminations. Also note in order to gain access to P1.0 the jumper for LED1 must be removed. Please disregard the white text and white boxes.

*Referenced from energia.nu
**TI MSP-EXP430F5529LP Datasheet Link in Appendix

Design Considerations:
The LaunchPad and WiFi booster pack when connected would not fit in the previous device housing which would be located in the hive. The depth of the new housing would have to be increased by approximately 6.4 mm in order to house the LaunchPad with the attached booster pack.

III. Design Description

Block Diagram:

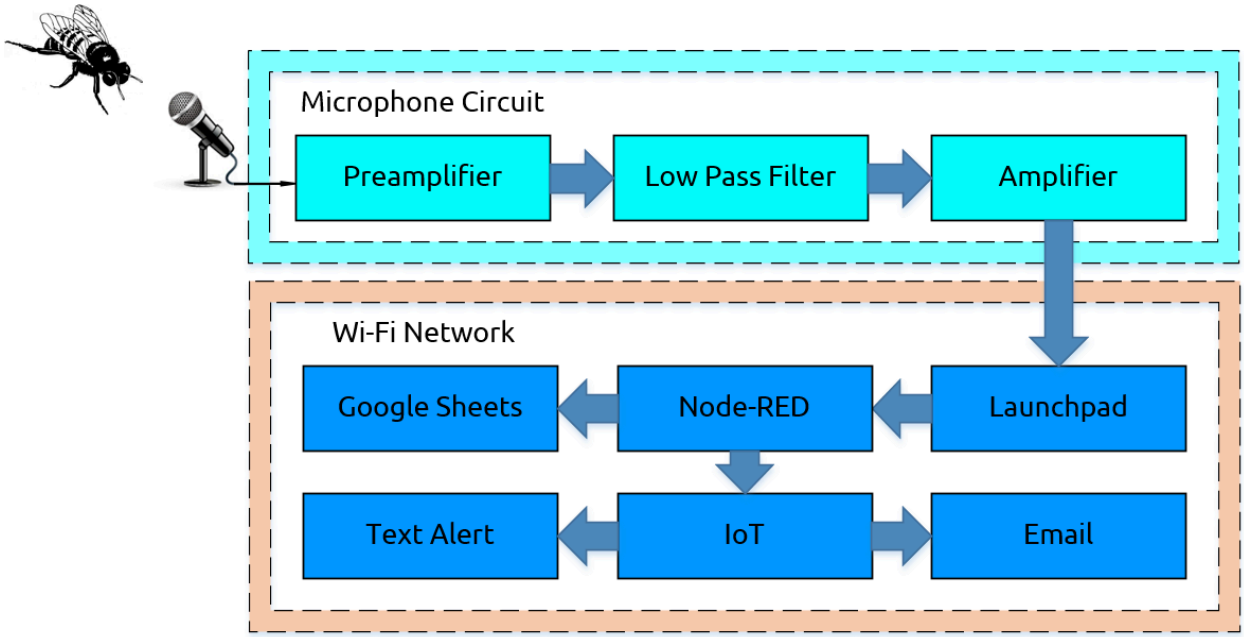

Figure 3: Block Diagram of System Design

Preamplifier:

A Common Emitter Preamplifier has a gain of approximately 20 prior to being filtered. The signal directly from the electret microphone is weak and filtering this initial weak signal may result in the loss of data. The preamplifier boosts the initial signal directly from the electret microphone prior to being filtered and finally amplified. This portion remained unchanged in regards to the previous circuit design but was left in as a reference.

Low Pass Filter:

After the signal is initially staged through the preamplifier it passes through a first order Low Pass Filter. The RC filter within the circuit has a cutoff frequency of approximately 4.1kHz (not including the rate of the roll-off). The filter for this circuit was updated for two reasons, the sampling rate allowed for the larger frequency and as I continued my research on acoustical analysis I found more data of hissing reported around 3kHz.

Amplifier:

This portion was also revamped compared to the previous circuit. The reason behind this was to allow for a flexible output range based off of the adjustment of a 10kΩ potentiometer which receives the low pass filtered signal and outputs the offset adjusted signal to the amplifier's (LM386N-3) input. This amplifier IC was chosen based off of the flexibility of input supply voltage, 4V minimum and low power consumption. The output voltage gains of this particular amplifier range from 20 to 200.

Characterization of Microphone Circuit:

In order to characterize the microphone circuit test tones ranging from 0Hz to 20kHz were played through a speaker which was located approximately 76 mm from the electret microphone. Although you will notice the second graph only reflects up to 7kHz, this is because the output did not exceed nominal ripple beyond 7kHz. It should be noted that the qualities of the speaker would have an impact on the response noted in these experiments.

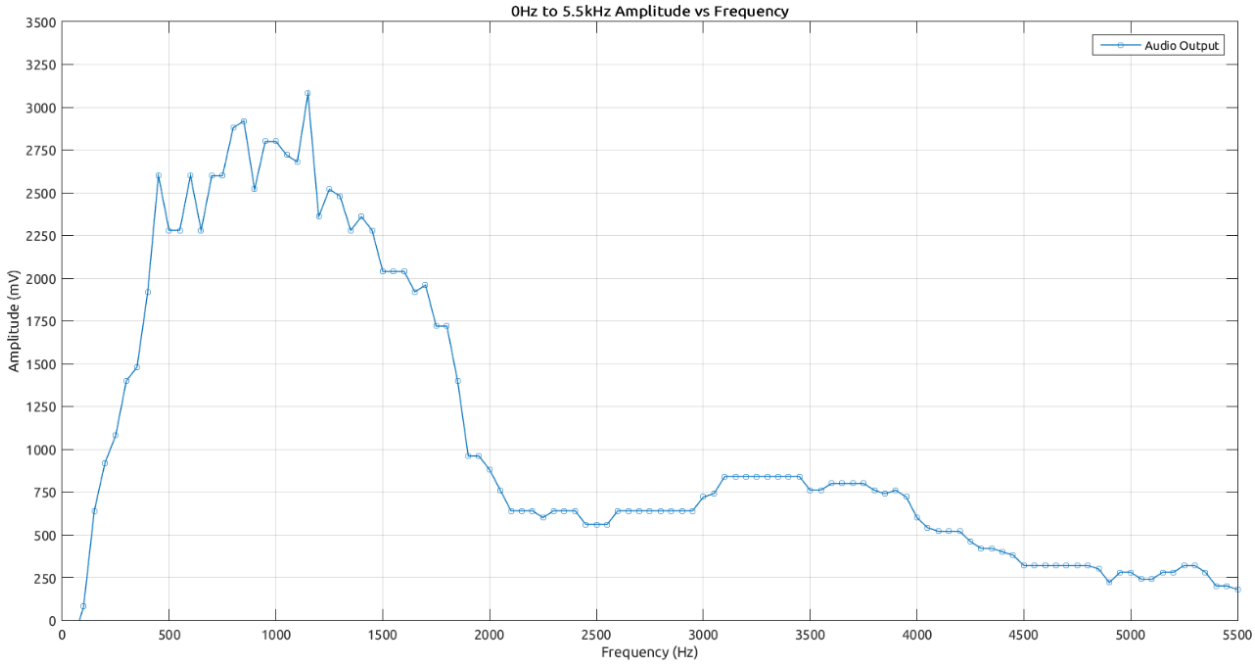

Figure 2: Circuit response to 0 Hz to 5.5kHz tones in 50Hz increments.

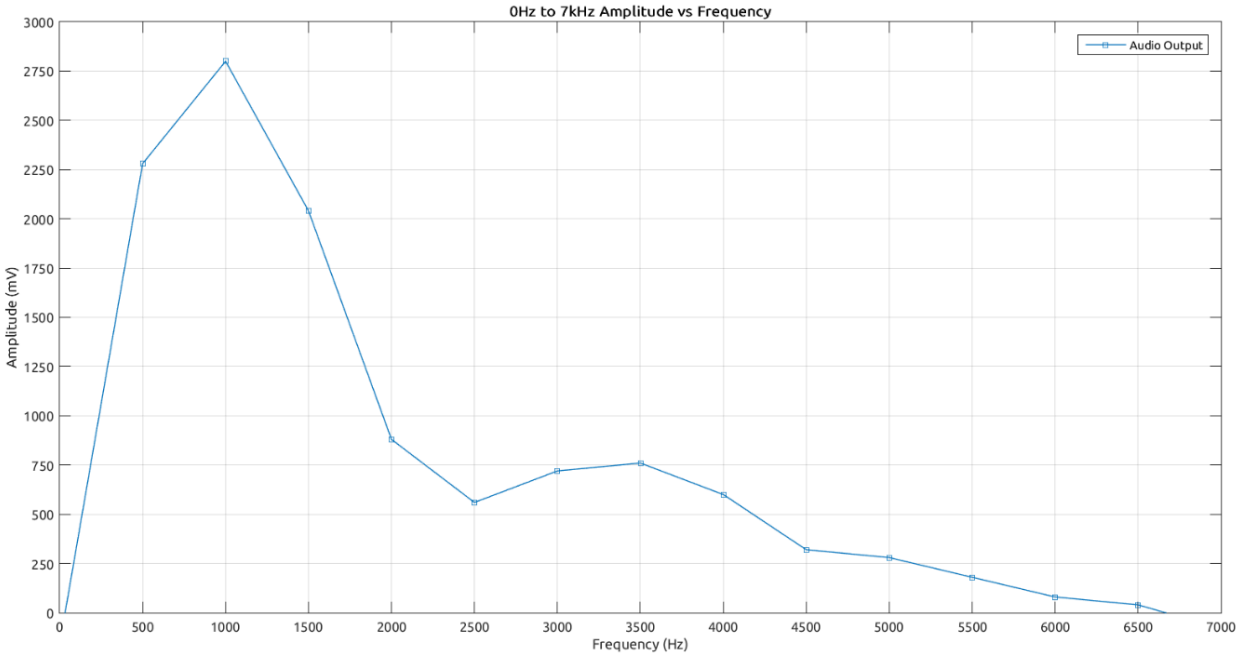

Figure 3: Circuit response to 0 Hz to 7kHz tones in 500Hz increments.

Circuit Schematic:

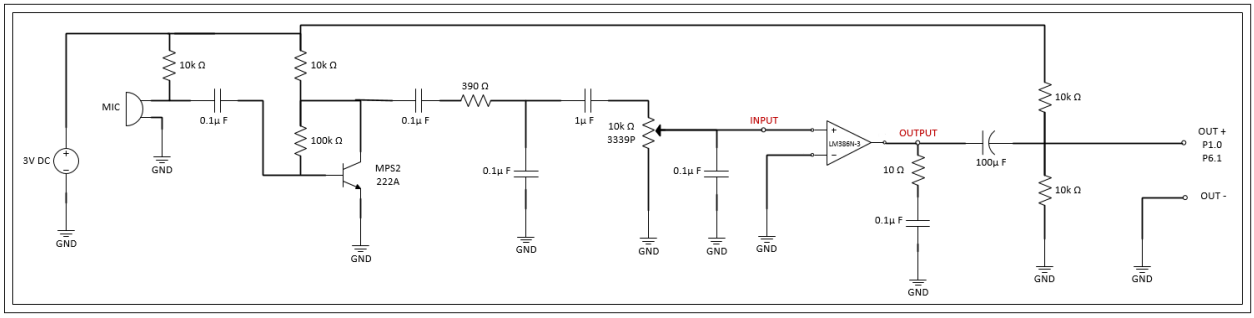

Figure 4: Microphone Circuit Diagram with Pin Terminations.

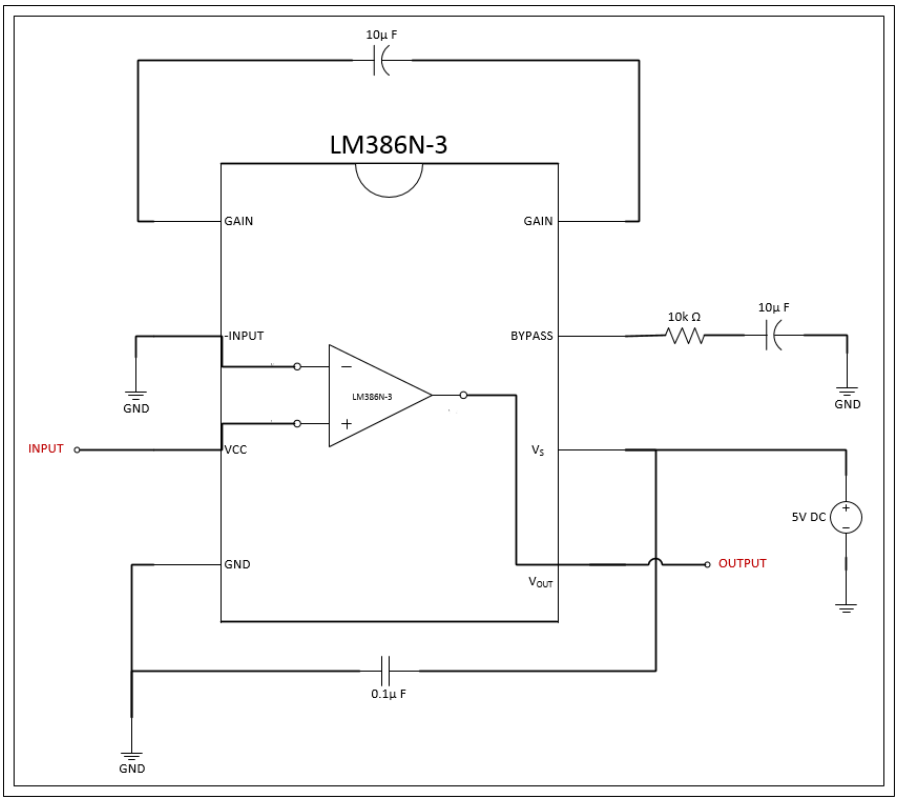

Figure 5: Microphone Circuit Amplifier Diagram.

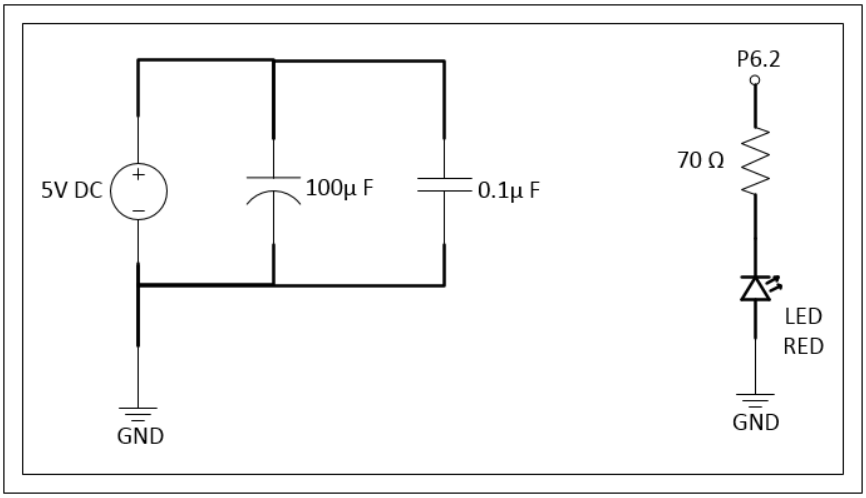

Figure 6: Decoupling Capacitors & Clipping Indicator LED Diagram.

Physical Construction:

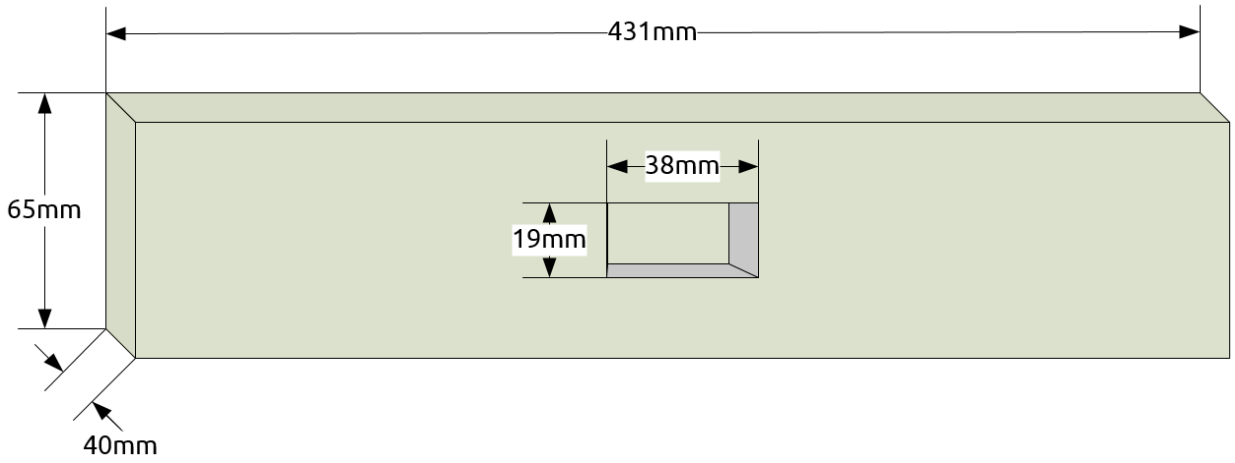

Figure 7: External Measurements of the Device Box.

The new external measurements can be seen above in Figure 7 with changes to reflect the addition of the LaunchPad. Please refer to the previous research study for full build log. The new device box will be identical aside from the increased depth.

IV. Code and Collected Data

LaunchPad Data Collection:

Prior to delving into the code necessary for the launchPad I want to cover one more portion of the microphone schematic. Due to the analog input on the LaunchPad which ranges from 0V to ~4.1V the amplified output which centered at 0V needed to be offset. You will notice above the use of a voltage divider which offsets the output by approximately +1.8V to compensate for proper input.

The LaunchPad code was all written in Energia, Energia allows the user to write code in "sketches" and upload them to the LaunchPad to alter the LaunchPad's functionality. The language used in Energia is based on C/C++, I will list a few reference websites in the appendix if you are either learning these languages for the first time or need to brush up on your coding.

In order to make the code a bit more readable all comments are highlighted in red text, keep in mind this code will throw errors unless the comments are commented out using proper syntax.

LaunchPad Code:

```
Counter library used in determining frequency
#include <CounterLib_t.h>

The following libraries are used in connecting to the internet via
WiFi as well as connecting with the MQTT Client
#include <WiFi.h>
#include <PubSubClient.h>
#include <SPI.h>
WiFiClient wclient;

----------Definitions----------
numReads: This value determines the number of readings
directly before the frequency is detected

timePast: This value determines the number of readings
past the time frequency is detected
```

```
publishDelay: This value determines the delay between
client calls and the rate at which data is published
in milliseconds

minAnalog: Minimum analog value used to determine
clipping

maxAnalog: Maximum analog value used to determine
clipping

ledPin: The digital output to control the clipping
LED indicator

WIFI_SSID: The SSID of the WiFi you are connecting to

WIFI_PWD: The password of the WiFi you are connecting to
#define numReads 1050
#define timePast 1000
#define publishDelay 800
#define minAnalog 0
#define maxAnalog 4095
#define ledPin 25
#define WIFI_SSID "HOME_WIFI"
#define WIFI_PWD "HOME_WIFI_PASSWORD"

Counter set to capture frequency on pin P1.0
Counter<CL_TimerA0> MyCounter;

Used in loop data output for frequency catch
int first = 1;

Wait time before next frequency catch setting this
value equal to 600 will delay the next frequency
by one minute. n*600 = n minute delay after readout
int nextRead = 3000;

Array to hold values which is to be dumped
int senseVals[(numReads+timePast)];
```

```
Array used in publishing the values read in
char sensorRead[4];

The MQTT server information
byte server[] = { 198, 41, 30, 241 };
byte ip[]     = { 172, 16, 0, 100 };

Setup main, connects to wifi and starts the counter to
determine frequency
void setup()
{
 Serial.begin(9600);

 Serial.println("Start WiFi");

 WiFi.begin(WIFI_SSID, WIFI_PWD);
 //uncomment line below and comment line above for open WiFi
 //WiFi.begin(WIFI_SSID);

 while(WiFi.localIP() == INADDR_NONE)
 {
  Serial.print(".");
  delay(300);
 }

 start counter
 MyCounter.start();
}

Main loop read each section for specific details
void loop()
{
 reset counter to zero
 MyCounter.reset();

 currFreq: Current frequency, initialized to zero

 trgrFreq: Trigger frequency, initialized to zero
 int currFreq, trgrFreq;
```

```
Loop used for initial sample
for(int i = 0; i < numReads; i++)
{
  senseVals[i] = (analogRead(24));
  delay(.01);
}

Current frequency variable set and output to serial
 monitor for debugging.
currFreq = (MyCounter.read() * 10);
Serial.print(currFreq);
Serial.println(" Hz");

Loop used to determine clipping and turn "ON" LED
if((analogRead(24) <= minAnalog) || (analogRead(24) >= maxAnalog))
{
  digitalWrite(ledPin,HIGH);
}

Loops used to test current frequency and output array
of data stored if certain frequency detected.
if((currFreq > 180) && (currFreq < 200) && ((first) || (nextRead <= 0)))
{
  trgrFreq = currFreq;
  first = 0;

  for(int i = numReads; i < (numReads+timePast); i++)
  {
    senseVals[i] = (analogRead(24));
    delay(.1);
  }

  for(int i = 0; i < (numReads + timePast); i++)
  {
    values output to serial monitor for debugging
    Serial.println(senseVals[i]);

    convert into char array
```

```
   String str = (String)senseVals[i];

   Length (with one extra character for the null terminator)
   int str_len = str.length() + 1;

   Prepare the character array (the buffer)
   char char_array[str_len];
   str.toCharArray(char_array, str_len);

   publish data to MQTT broker
   if (client.connect("LaunchPadClient"))
   {
    client.publish("ECE397", char_array);
    Serial.println("Publishing successful!");
    client.disconnect();
    delay(publishDelay);
   }
  }

  output of trigger frequency used in debugging
  Serial.print("The trigger frequency detected: ");
  Serial.println(trgrFreq);

  used to set a delay before next frequency catch
  nextRead = 3000;
}
else
{
  else the frequency is not read in and next read is decremented
  if(!first)
  {
   nextRead--;
  }
}

Loop used to determine clipping and turn "OFF" LED
if((analogRead(24) > minAnalog) || (analogRead(24) < maxAnalog))
{
  digitalWrite(ledPin,LOW);
```

```
  }
}

Function to print the status of the WiFi connection
void printWifiStatus()
{
  print the SSID of the network you're attached to:
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());

  print your WiFi shield's IP address:
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);
}
```

*Credit for any libraries used in this code are located in References

Node-RED Collection:

Node-RED contains the client which receives the data from the LaunchPad, as well as the decision maker for what service gets what piece of the received information. I will briefly explain each node and list the flow that is generated when backing up the node information and layout in Javascript in the appendix.

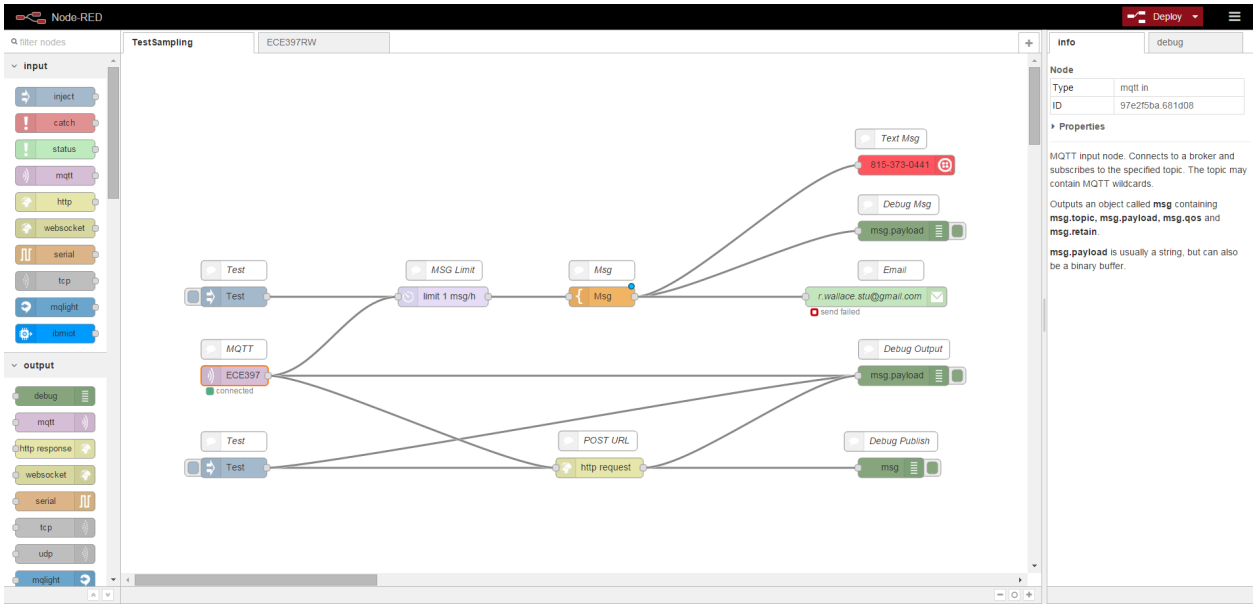

Figure 8: Node-RED GUI.

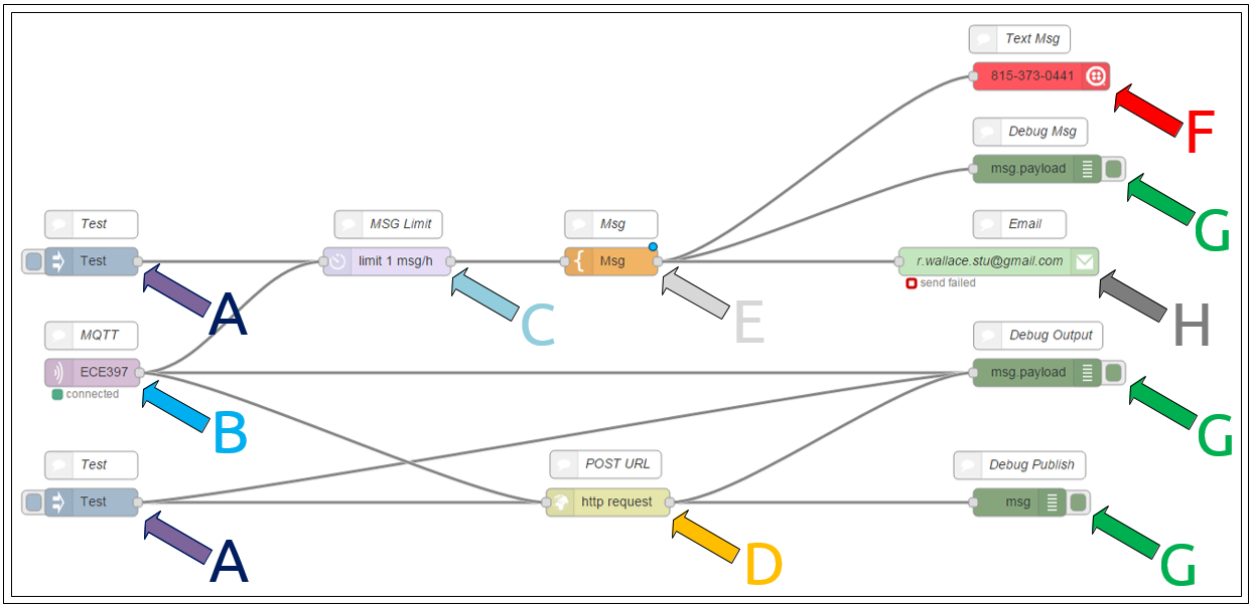

Figure 9: Node-RED Flow with Nodes Labelled.

Node-RED Nodes:

First you may notice a white box above all nodes listed, this is a comment that I have placed near boxes with some information in regards to the node.

A. These are test nodes, used to test the results of receiving data prior to having the LaunchPad connected by implementing strings into the receiving nodes.
B. This is the MQTT node that receives the data from the LaunchPad. This is the node that receives all client publishings written in the above the LaunchPad code.
C. In order to keep text messages or emails from receiving all data sent this node allows one signal which is the trigger to be passed through and disregards all messages for a specified period of time.
D. This node publishes the data to a specified Google Sheet. Note that this node is not affected by the limiting node since we want all data published into the spreadsheet.
E. With this node I am altering the message to be sent to downstream connected nodes. This message includes information as well on what number caused the data send to trigger.
F. Via Twilio this node sends a text message with the form filled out in node E to the phone number specified. This node is downstream from the limiting node therefore the text message is only sent if enough time has passed or if there was no previous message sent.

G. These nodes are used in the debugging process, I can use these nodes to see how things will be output prior to connecting to sending nodes.
H. This node will email the message with the form filled out in node E to the email address specified. Similarly to node F, the emails sent are limited based off of time of past messages sent.

Data Retrieved:

For testing purposes a 400Hz test tone was played into the LaunchPad which was set to trigger on input and began sampling the input signal, the data was immediately stored into an array on the LaunchPad and published to a Google Sheet at a much slower rate. It should be noted that the reason for the much slower rate is to keep data transfer accuracy as high as possible.

	A	B
1	Timestamp	Audio Samples
2	12/19/2015 21:28:14	1414
3	12/19/2015 21:28:16	1045
4	12/19/2015 21:28:18	759
5	12/19/2015 21:28:20	557
6	12/19/2015 21:28:22	436
7	12/19/2015 21:28:25	411
8	12/19/2015 21:28:27	505
9	12/19/2015 21:28:29	665
10	12/19/2015 21:28:31	902
11	12/19/2015 21:28:33	1212
12	12/19/2015 21:28:35	1566
13	12/19/2015 21:28:37	1943
14	12/19/2015 21:28:39	2320
15	12/19/2015 21:28:42	2674
16	12/19/2015 21:28:44	3007
17	12/19/2015 21:28:46	3280
18	12/19/2015 21:28:48	3492
19	12/19/2015 21:29:13	2329
●	●	●
●	●	●
●	●	●
2039	12/19/2015 22:40:46	864
2040	12/19/2015 22:40:48	1132
2041	12/19/2015 22:40:50	1446
2042	12/19/2015 22:40:52	1824
2043	12/19/2015 22:40:55	2188
2044	12/19/2015 22:40:57	2553

Figure 10: Google Sheet with Published Data.

The audio samples from the Google spreadsheet were then analyzed in MATLAB and the results are shown below. Here we can see that the frequency of the data obtained is listed as ~395.6Hz. The percentage error of the data sent was approximately 1% compared to the 400Hz input signal.

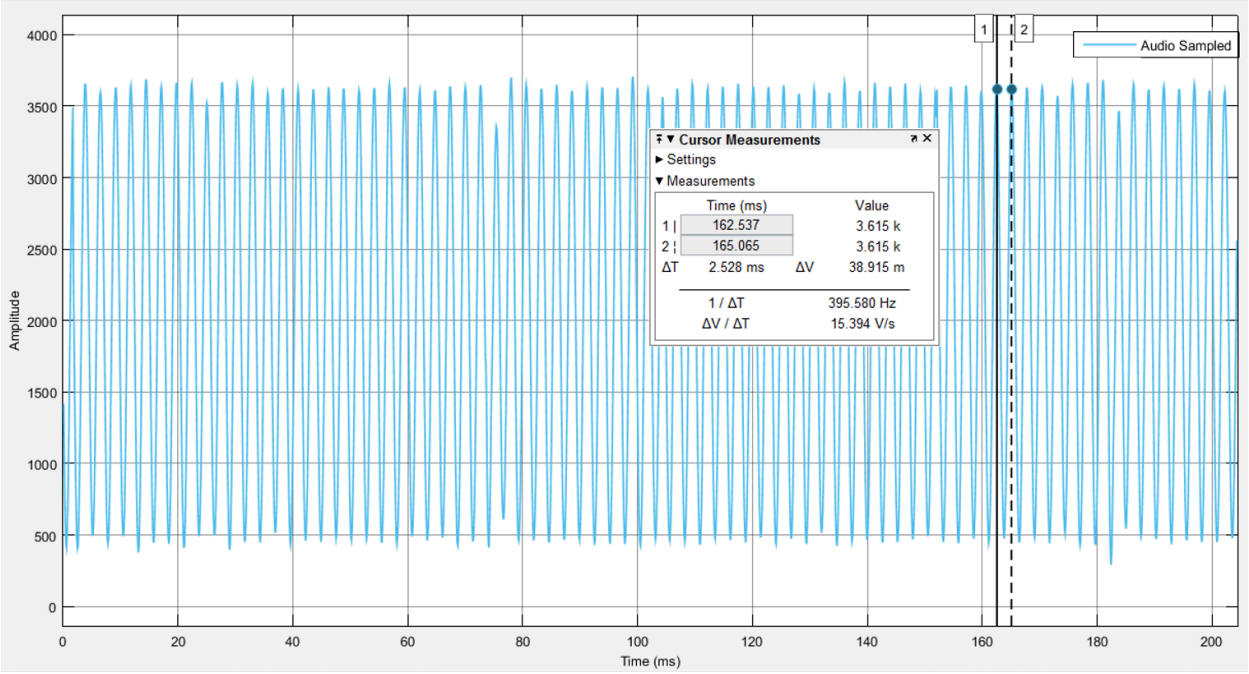

Figure 11: Google Sheet Data Analyzed in MATLAB.

V. Conclusion

- Lessons learned

  Beginning where things left off last semester, I initially began by troubleshooting issues with the previous microphone circuit. The voltage was offset and would float to lower values when processing audio. This was determined to be an op amp whose negative supply voltage was wired directly to ground. This would prove to be a non existent issue with the latest circuit since I was now using a different amplifier stage for my output.

  The push to make LED was removed from the original circuit as well, to keep the circuit at compact as possible and the need for this LED was now unnecessary. The LaunchPad is equipped with onboard LEDs which signal power.

  The clipping LED built into the new circuit makes tuning the potentiometer around max input volume level possible on the fly without having to connect the LaunchPad to serial monitor. The values which trigger the LED are the absolute max values before the signal begins to clip, these will likely be adjusted to allow for volume past the max setting.

- Self-assessment

  Although I am at the point where data can be gathered and sent over a number of mediums through WiFi, there still exists a portion of the code that causes unwanted errors when triggering off of a certain frequency. After some debugging I have determined the cause is multiple header files in my code using the same timer which is causing a conflict.

  This project wrapped up at the very end of the semester, I was unable to test the device in the hive as I would have hoped. This delays testing of actual longevity that the device can continue to operate properly within an active beehive.

  The transfer of data proved most difficult, especially when it came to choosing which service to go with for the client and what service to provide an “internet of things” or way of connecting to various mediums such as text, email, Google

Sheets. Having now worked with a few clients and determined pros and cons of each, I feel this trial and error process was the major setback. That is not to say it wasn't a great learning experience.

- Future work

  The arrays that I am filling with data as of now are relatively small due to the limitation on actual storage size allowed by the LaunchPad. I have considered workarounds for this and will attempt to implement them but I ultimately feel the fix for this will be the addition of an SD card storage device that the LaunchPad will write to.

  A few of the header files used could likely be slimmed down a bit more considering some were written for multiple microcontrollers. This goes back to workarounds for current limitations in the amount of sampled data that can be stored on the LaunchPad.

  Placing this device into the hive is one of the next main steps, it will be interesting to see how long the device will last in the hive and continue operating. In addition to the operation time, I currently do not have a battery readout of any sort. Adding a possible warning to alert a user when batteries are nearing the end of their life would also likely be a necessary addition. As opposed to placing a device in a hive, I would also like to work on a device that could be placed on the exterior of the hive with only a microphone placed internally. Furthermore it would be interesting to learn the effect of having a device such as this in a hive.

VI. References

Books:

Kudeki, E., & Munson, D. (2009). Analog Signals and Systems. Upper Saddle River, N.J.: Pearson Prentice Hall.

Sanford, M., & Bonney, R. (2010). Storey's Guide to Keeping Honey Bees. North Adams, Mass.: Storey Pub.

Websites:

http://energia.nu/

- Libraries used in my code supplied at: http://energia.nu/reference/libraries/

http://www.honeybeesonline.com/

http://www.mathworks.com/

http://www.ni.com/

http://nodered.org/

http://www.ti.com/

VII. Appendix

User Guides & Datasheets:

Link to TI LaunchPad (MSP-EXP430F5529LP) Users Guide:
http://www.ti.com/general/docs/lit/getliterature.tsp?baseLiteratureNumber=slau533&fileType=pdf

Link to TI WiFi Boosterpack (CC3100BOOST) Users Guide:
http://www.ti.com/lit/ug/swru371b/swru371b.pdf

Link to Electret Microphone (CMA-4544PF-W) Datasheet:
http://www.cui.com/product/resource/cma-4544pf-w.pdf

Link to Transistor (MPS2222A) Datasheet:
http://www.farnell.com/datasheets/115091.pdf

Link to Audio Power Amplifier (LM386N-3) Datasheet:
http://www.mouser.com/ds/2/405/lm386-453073.pdf

Link to Potentiometer (3339P) Datasheet:
https://www.bourns.com/pdfs/3339.pdf

Useful Links:

Link to my previous study report:
https://docs.google.com/document/d/11WVsHn-d3FlYHKH-5euWJz4IlRZ5RHq8AnGWWGuaaTI/edit?usp=sharing

Link to my Node-RED to Google Spreadsheet tutorial:
https://docs.google.com/document/d/1JyYio7BOsgcO82UCMofVR-SoUVMyYP8Q44nAin8Gntc/edit?usp=sharing

Link to Energia's guide that starts from the ground up:
http://energia.nu/guide/

TI's wiki setup to help beginning users learn their LaunchPad:
http://processors.wiki.ti.com/index.php/Getting_Started_with_the_MSP430_LaunchPad_Workshop

Link to the Energia pin mapping of the MSP430F5529:
http://energia.nu/pin-maps/guide_msp430f5529launchpad/

Link to Google's spreadsheets information:
https://www.google.com/sheets/about/

Recommended Links:

Link to reviews of various IoTs:
http://embeddedcomputing.weebly.com/iot-services-which-solution.html

Beginners tutorial to Node-RED:
http://nodered.org/docs/getting-started/first-flow.html

C Language tutorials:
http://www.cprogramming.com/

C++ Language tutorials:
http://www.cplusplus.com/doc/tutorial/

Link to honey bee tracking research:
http://news.illinois.edu/news/14/0722RFID_GeneRobinson.html

Link to site devoted to hive monitoring:
http://colonymonitoring.com/cmwp/

Link to another site devoted to hive monitoring:
http://hivetool.org/w/index.php?title=HiveTool.org

Link to a site devoted to combining technology and beekeeping.
http://www.beehacker.com/

Full Node-RED Code:

*My exact Node-Red flow can be copied into a new Node-RED page by copying the text and pressing CTRL+i in the design area in order to import this flow.

[{"id":"3d37b2b.fc2c84e","type":"twilio-api","z":"a6255960.59daa8","sid":"AC7931ad9b85563fa1549522bdd6a5a834","from":"815-373-0441","name":""},
{"id":"61a6b3c9.9e594c","type":"mqtt-broker","z":"a6255960.59daa8","broker":"198.41.30.241","port":"1883","clientid":"","usetls":false,"verifyservercert":true,"compatmode":false,"keepalive":"10","cleansession":true,"willTopic":"","willQos":"0","willRetain":"false","willPayload":"","birthTopic":"","birthQos":"0","birthRetain":"false","birthPayload":""},
{"id":"97e2f5ba.681d08","type":"mqttin","z":"a6255960.59daa8","name":"","topic":"ECE397","broker":"61a6b3c9.9e594c","x":173,"y":490,"wires":[["7197a771.8e6858","17ede230.e8121e","6d8f13ac.9270ec"]]},
{"id":"3cd19bb4.c32e64","type":"inject","z":"a6255960.59daa8","name":"","topic":"","payload":"Test","payloadType":"string","repeat":"","crontab":"","once":false,"x":172,"y":630,"wires":[["6d8f13ac.9270ec","7197a771.8e6858"]]},
{"id":"7197a771.8e6858","type":"debug","z":"a6255960.59daa8","name":"","active":true,"console":"false","complete":"payload","x":1191,"y":490,"wires":[]},
{"id":"74402a1d.8bbfd4","type":"comment","z":"a6255960.59daa8","name":"POSTURL","info":"https://docs.google.com/forms/d/**FormID**/formResponse?entry.**Entry Number**=AudioSamples\n\nAreas requiring changing are marked with dual asterisks before andafter","x":726,"y":589,"wires":[]},
{"id":"88869f33.77796","type":"debug","z":"a6255960.59daa8","name":"","active":true,"console":"false","complete":"true","x":1172,"y":630,"wires":[]},
{"id":"6d8f13ac.9270ec","type":"httprequest","z":"a6255960.59daa8","name":"","method":"POST","ret":"obj","url":"https://docs.google.com/forms/d/1ILSopseehOrvIaf6zWGyWfUhiJoOm7j113tiepqw5vc/formResponse?entry.1514992666={{payload}}","x":728.5,"y":630,"wires":[["88869f33.77796","7197a771.8e6858"]]},
{"id":"a1955d44.5e6aa","type":"twilioout","z":"a6255960.59daa8","service":"_ext_","twilio":"3d37b2b.fc2c84e","from":"","number":"<Desired_Cell_Phone_Number>","name":"","x":1195.5,"y":170,"wires":[]},
{"id":"261ca125.d9e35e","type":"debug","z":"a6255960.59daa8","name":"","active":true,"console":"false","complete":"false","x":1191,"y":270,"wires":[]},
{"id":"e764c9b1.189b38","type":"inject","z":"a6255960.59daa8","name":"","topic":"","payload":"Test","payloadType":"string","repeat":"","crontab":"","once":false,"x":172,"y":370,"wires":[["17ede230.e8121e"]]},
{"id":"da4cdb44.25b328","type":"comment","z":"a6255960.59daa8","name":"MQTT","info":"You server # will depend on the MQTT server you are using\n\nThe topic will be the same as the topic with which you publish your code to and connect your client to in energia.","x":172,"y":450,"wires":[]},

{"id":"c0367019.3fc99","type":"comment","z":"a6255960.59daa8","name":"Test",
"info":"This node was set up to test the publishing to a Google
Sheet.","x":172,"y":590,"wires":[]},
{"id":"974285f1.68bd78","type":"comment","z":"a6255960.59daa8","name":"De
bug Output","info":"Debug node to confirm that the test node and MQTT node
are outputting properly.","x":1192,"y":450,"wires":[]},
{"id":"3f3dbf15.c0c24","type":"comment","z":"a6255960.59daa8","name":"Debu
g Publish","info":"Debug node for http request response, if deployed you can
debug your code in the debug area using the output from this
node","x":1172,"y":590,"wires":[]},
{"id":"6810206d.97efe","type":"template","z":"a6255960.59daa8","name":"Msg",
"field":"payload","format":"handlebars","template":"The frequency was:
{{payload}}!","x":732,"y":370,"wires":[["261ca125.d9e35e","a1955d44.5e6aa","7a
9c8c2d.856374"]]},
{"id":"17ede230.e8121e","type":"delay","z":"a6255960.59daa8","name":"","paus
eType":"rate","timeout":"5","timeoutUnits":"seconds","rate":"1","rateUnits":"ho
ur","randomFirst":"1","randomLast":"5","randomUnits":"seconds","drop":true,"x
":490.5,"y":370,"wires":[["6810206d.97efe"]]},
{"id":"3277d7fa.cd8828","type":"comment","z":"a6255960.59daa8","name":"MS
G Limit","info":"Limits the amount of messages that go through to time
set\n\nAlso disregards messages sent until the limit time has
passed","x":492,"y":330,"wires":[]},
{"id":"7a9c8c2d.856374","type":"e-mail","z":"a6255960.59daa8","server":"smtp.
gmail.com","port":"465","name":"<Desired_Email_Address>","dname":"","x":114
9.5,"y":370,"wires":[]},
{"id":"3fb7b521.c0484a","type":"comment","z":"a6255960.59daa8","name":"Test
","info":"This node was set up to test the publishing to a Google
Sheet.","x":172,"y":330,"wires":[]},
{"id":"7884545e.877bac","type":"comment","z":"a6255960.59daa8","name":"Ms
g","info":"Alters the message to include given text","x":732,"y":330,"wires":[]},
{"id":"f23fe0bb.0dc02","type":"comment","z":"a6255960.59daa8","name":"Email
","info":"","x":1172,"y":330,"wires":[]},
{"id":"72398c4a.8dc674","type":"comment","z":"a6255960.59daa8","name":"Deb
ug Msg","info":"Debug node","x":1183.5,"y":230,"wires":[]},
{"id":"ed23a91f.12dc58","type":"comment","z":"a6255960.59daa8","name":"Text
Msg","info":"Sends a text message","x":1172,"y":130,"wires":[]}]