Radio Frequency Readout Device (RFRD)

TO DESIGN A RADIO FREQUENCY READOUT DEVICE TO USE IN A BOLT ANCHOR SURVEYING APPLICATION.

Team 11
Client: Dr. Daji Qiao and Dr. Nathan Neihart
Advisors: Same as client
Team Members/Roles: Bailey Akers, Colin Sunderman, Pengyu Qu, Lyle Bishop, Nathan Mulbrook
Team Email: sdmay18-11@iastate.edu
Team Website: http://sdmay18-11.sd.ece.iastate.edu/

Revised: Date: 4/23/2018
Version: 5

Table of Contents

List of Figures/Tables/Symbols/Definitions

1 Introduction

1.1 ACKNOWLEDGEMENT

We would like to acknowledge the support from our advisors Dr. Daji Qiao and Dr. Nathan Neihart. We would also like to thank graduate students Scott Melvin and Chengrui Yang for their assistance throughout this project.

1.2 PROBLEM AND PROJECT STATEMENT

The purpose of this project was to reduce the time it takes to check each nut for tightness on bolts of large lamp post structures and to improve the accuracy of these tightness measurements. Currently each nut must be manually checked for tightness with large tools to ensure that the structure is sound. This method of checking tightness takes time and can give inaccurate tightness measurements.

Inaccurate tightness measurements can occur when the nut cross-threads on the bolt. This results in a torque wrench measuring a tight connection when in fact it is not.

To address this problem, this project uses a radio frequency readout device (RFRD) to read if each nut is secured tightly or if it needs to be properly torqued. We have designed a passively powered RFRD tag that reads the capacitance value of two washers between a bolt and nut. This capacitance measured is proportional to the distance between the washers and can be used to check how tight the nut is.

The goal in the future is to then send a signal back to a reader that tells the user if the nut is tight enough, but for this prototype we will be using a visual indicator to indicate if the nut is tight.

1.3 OPERATIONAL ENVIRONMENT

This prototype will not be subject to any non-ambient weather conditions, but in future iterations it will need to perform in outdoor conditions.

1.4 INTENDED USERS AND USES

This design is intended to be used by civil engineers and construction workers for structures that require inspections to the structure's integrity and tightness of its anchor fasteners.

1.5 ASSUMPTIONS AND LIMITATIONS

Assumptions for prototype:

The circuitry will not be exposed to outdoor conditions. The circuitry will be scalable for future revisions. The project's success will not be contingent on full operation of a reader.

Limitations for prototype:

The circuitry must be powered wirelessly at any distance. The circuitry must provide some sort of feedback to the user to indicate a threshold of anchor nut tightness.

1.6 EXPECTED END PRODUCT AND DELIVERABLES

The expected deliverables are as follows: a printed circuit board that will be complete with a capacitance measuring circuit, a low power microcontroller with firmware, an inverted-F antenna, and a rectifier.

We are expected to be able to demonstrate wireless functionality of our design at any distance.

Our delivery dates will be expected by finals week in May of 2018.

2. Specifications and Analysis

2.1 PROPOSED DESIGN

The goal of this project is to design a radio frequency readout device (RFRD) to use in an anchor bolt surveying application.

We have three major hardware components of our design:

1) Capacitance Measuring circuit

This circuit will measure the capacitance of the two washers on the bolt. This data is used to indicate how tight the nut is.

2) Antenna and Power Harvesting Rectifier

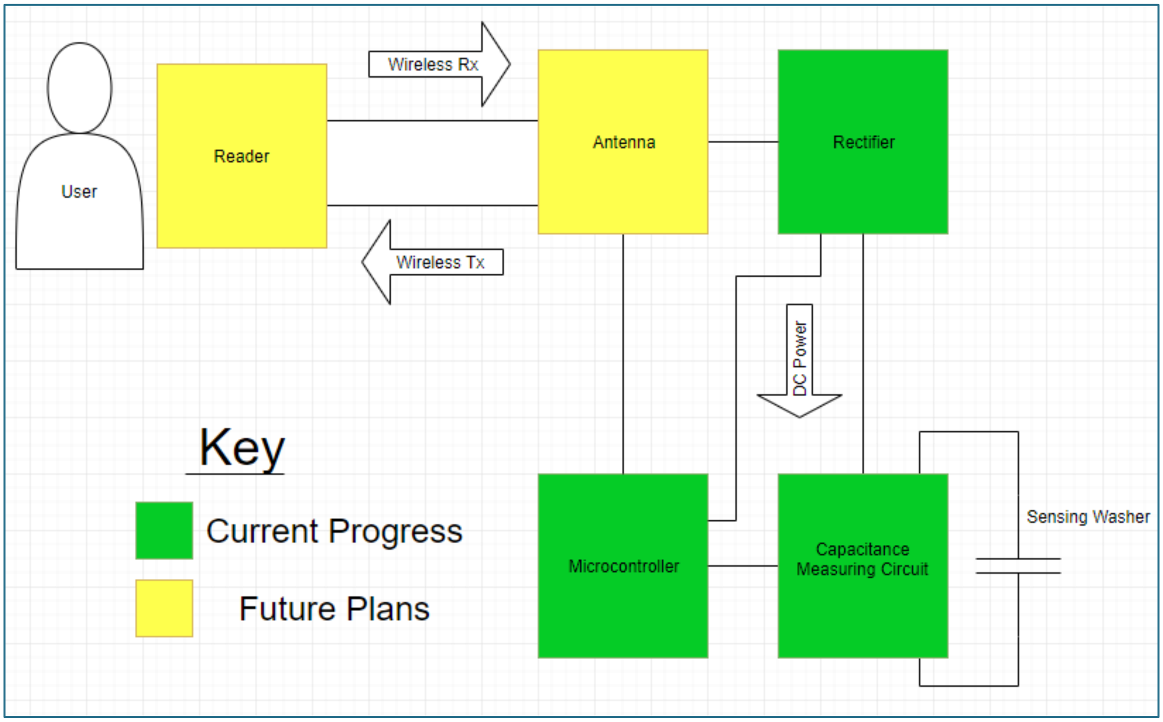
The receiving antenna receives AC power wirelessly from a source. The rectifier then converts this AC power to DC power. This DC power is sent to the capacitance measuring circuit and the microcontroller.

3) Microcontroller

The microcontroller measures the pulse signal output of the capacitance measuring circuit, analyzes the data, and outputs to a visual indicator whether or not the nut is tight.

A simple block diagram for our design is as follows:

Figure 1: RFRD Block Diagram



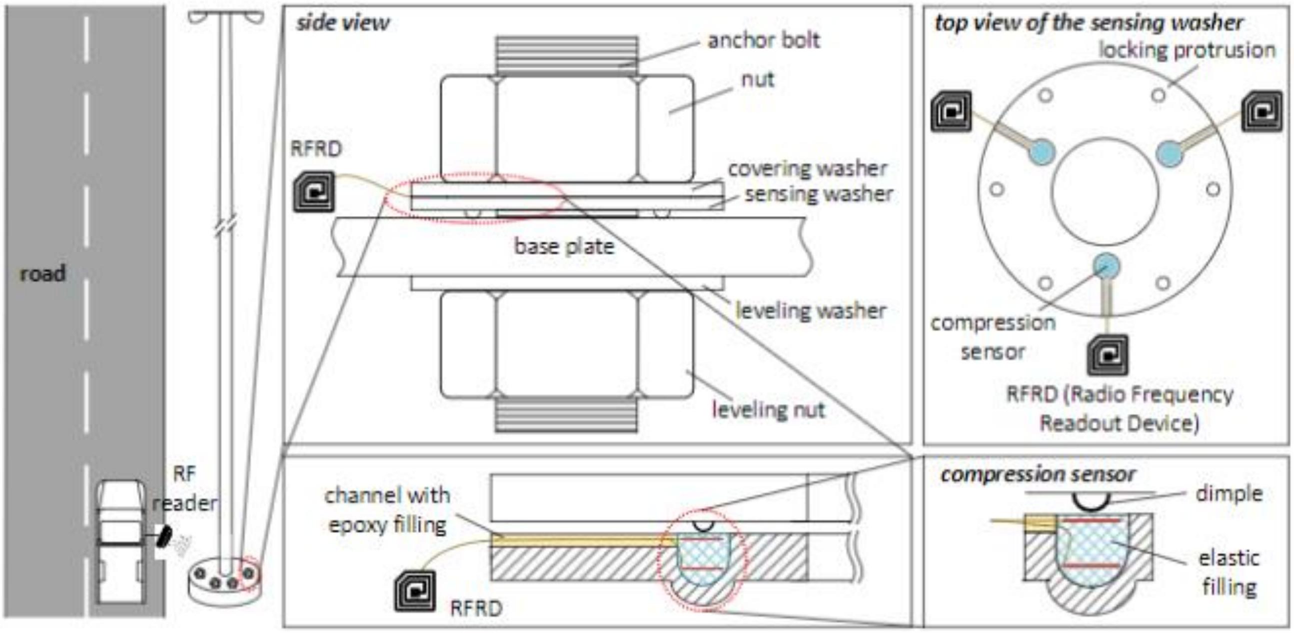
Capacitance Measuring Circuit (Relaxation Oscillator):

The idea is to use a relaxation oscillator to measure the capacitance between the covering and sensing washers. Using this oscillator and the relationship between capacitance and distance, we are able to determine the tightness of the locking nut that's located on top of the washer.

Theoretically, the tighter the locking nut, the more capacitance we see across the covering and sensing washers.

The mechanical design was provided by our advisor Dr. Daji Qiao and his research team specifically for use in our project. Our goal is to design a circuit to fit this mechanical design.

Figure 2: Mechanical Design



The range of capacitances that the circuit must measure is approximately 20 pF to 600 pF. This data was given to us by our client from tests performed on the washer design.

We chose to do some of our own measurements of the washer's capacitances to gather a better understanding of the effect of the pressure on capacitance and to have some measurements for testing purposes. We tested the washer design by soldering two wires, one on the covering and another on the sensing washers. We then measured the capacitance of the system using an LCR meter.

We concluded as follows:

Test 1: Resting washers: 20 pF

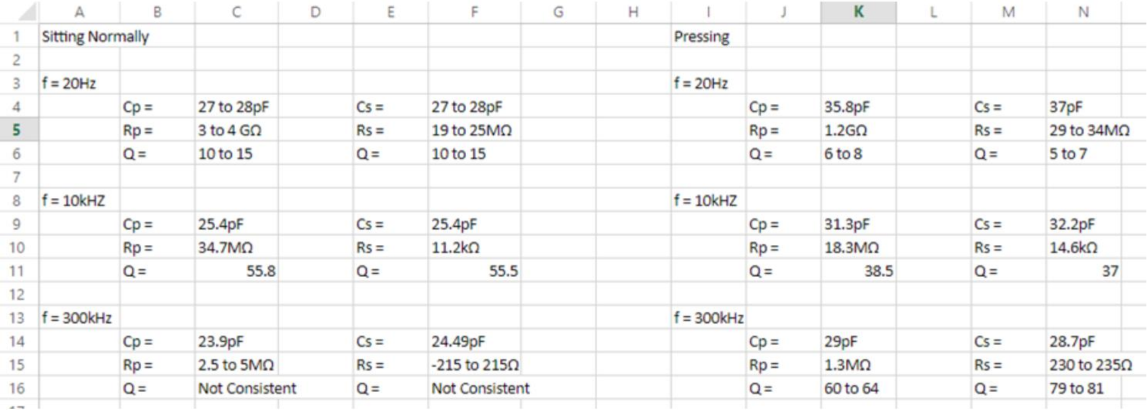
Test 2: No contacting washers: 2 pF

Test 3: Pressured washers: 30 – 40 pF

According to our advisors the actual range of measurement will need to be from ~20pF to ~600pF.

We also performed additional tests using the LCR meter at multiple measuring frequencies. We noted the following measurements:

Figure 3: LCR Measurements



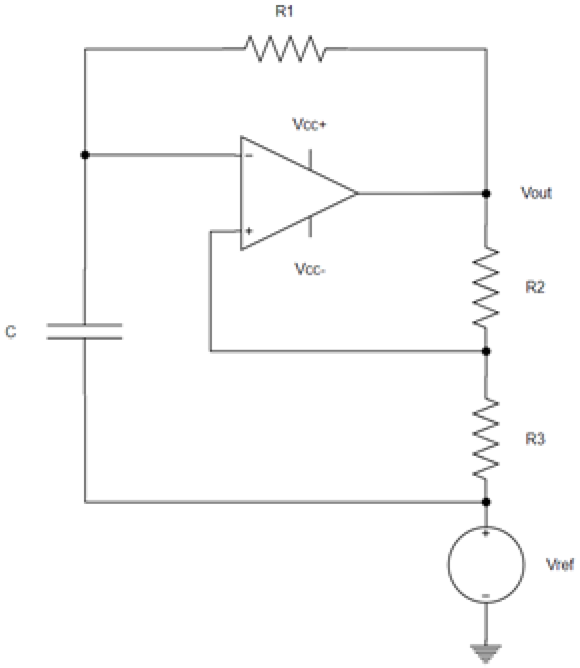
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Sitting Normally								Pressing					
2														
3	f = 20Hz								f = 20Hz					
4		Cp =	27 to 28pF		Cs =	27 to 28pF				Cp =	35.8pF		Cs =	37pF
5		Rp =	3 to 4 GΩ		Rs =	19 to 25MΩ				Rp =	1.2GΩ		Rs =	29 to 34MΩ
6		Q =	10 to 15		Q =	10 to 15				Q =	6 to 8		Q =	5 to 7
7														
8	f = 10kHZ								f = 10kHZ					
9		Cp =	25.4pF		Cs =	25.4pF				Cp =	31.3pF		Cs =	32.2pF
10		Rp =	34.7MΩ		Rs =	11.2kΩ				Rp =	18.3MΩ		Rs =	14.6kΩ
11		Q =	55.8		Q =	55.5				Q =	38.5		Q =	37
12														
13	f = 300kHz								f = 300kHz					
14		Cp =	23.9pF		Cs =	24.49pF				Cp =	29pF		Cs =	28.7pF
15		Rp =	2.5 to 5MΩ		Rs =	-215 to 215Ω				Rp =	1.3MΩ		Rs =	230 to 235Ω
16		Q =	Not Consistent		Q =	Not Consistent				Q =	60 to 64		Q =	79 to 81

This concludes that the quality factor (Q) for our sensing washers are the most consistent while measuring at 10 kHz. We will get best quality of capacitance measurement if we tune our relaxation oscillator to around 10 kHz while measuring the capacitance of the sensing washer.

Relaxation Oscillator:

Our relaxation oscillator design consists of a comparator IC, the capacitor being measured, and several resistors. The comparator compares the voltages at its two inputs. If the voltage at the positive input is greater than the negative input the output is the positive supply voltage of the comparator. If the voltage at the negative input is greater than the voltage at the negative input the output is the negative supply voltage of the comparator. The output of the comparator charges the capacitor. When the voltage across the capacitor is greater than the voltage at the positive input of the comparator the output of the switches from positive to negative output. The capacitor discharges and when its voltage drops below the voltage at the positive input the output flips to positive again. This constant change causes a square wave output that is dependent on the capacitance, resistor values, and the supply values.

Figure 4: Relaxation Oscillator



For our circuit, the theoretical resistor values are as follows:

R1 = 10 Mohms, R2 = 10 kohms, R3 = 13.3 kohms

For our circuit, the voltage values are as follows:

Vref = 1 V, Vcc+ = 2 V, Vcc- = 0 V

This relaxation oscillator generated a pulse wave with a period as follows.

Figure 5: Relaxation Oscillator Version 2 Period Formula

$$T = T_H + T_L = R * C * \ln\left(\frac{(V_{cc+} - V_{TL}) * (V_{cc-} - V_{TH})}{(V_{cc+} - V_{TH}) * (V_{s-} - V_{TL})}\right)$$

Our derived theoretical equation for our circuit simplifies to the following:

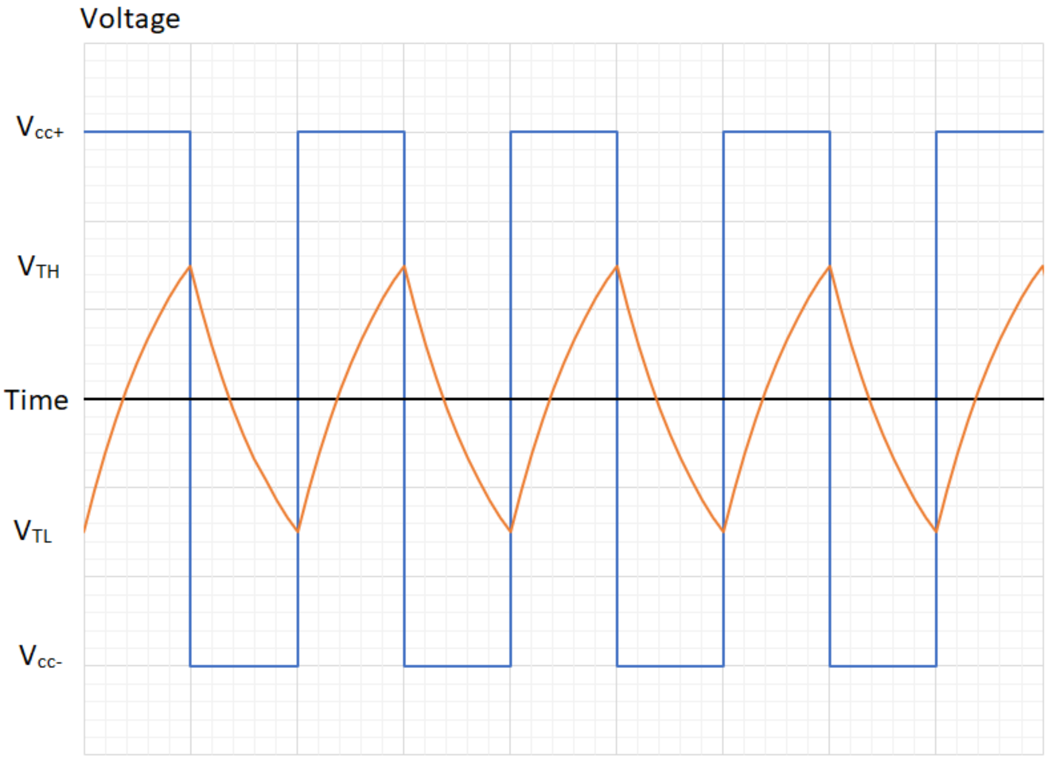
$$T = R * C * 2.19722$$

T = period of pulse wave in seconds.

C = capacitance of the measured capacitor in Farads.

R = resistance value of R in Ohms.

Figure 6: Relaxation Oscillator Version 2 Theoretical Output

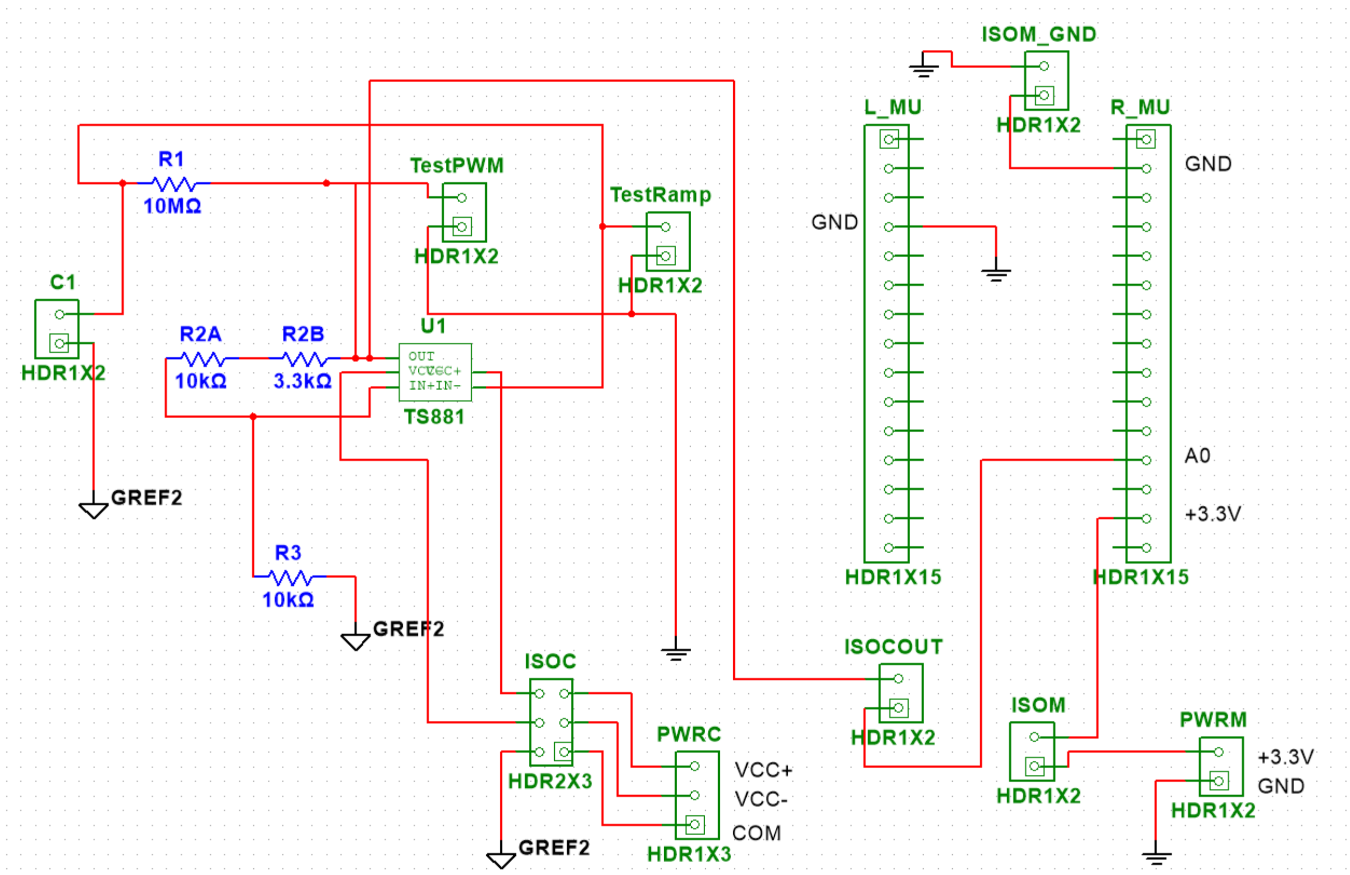


Since this circuit uses a comparator, we decided to use the STMicroelectronics TS88 comparator IC. This chip is very low power and uses ~1.285 uW per our simulations in appendix II.

Once we completed the simulations of the circuit design we built it on breadboard and did testing of the circuit. Information about the breadboard tests can be found in appendix II.

After testing the breadboard circuit, we were ready to create a PCB for our circuit.

Figure 7: Multisim Schematic for PCB Design



We designed our circuit in NI Multisim, then finished the PCB design in NI Ultiboard. Our design for the PCB includes both the relaxation oscillator and the microcontroller development board.

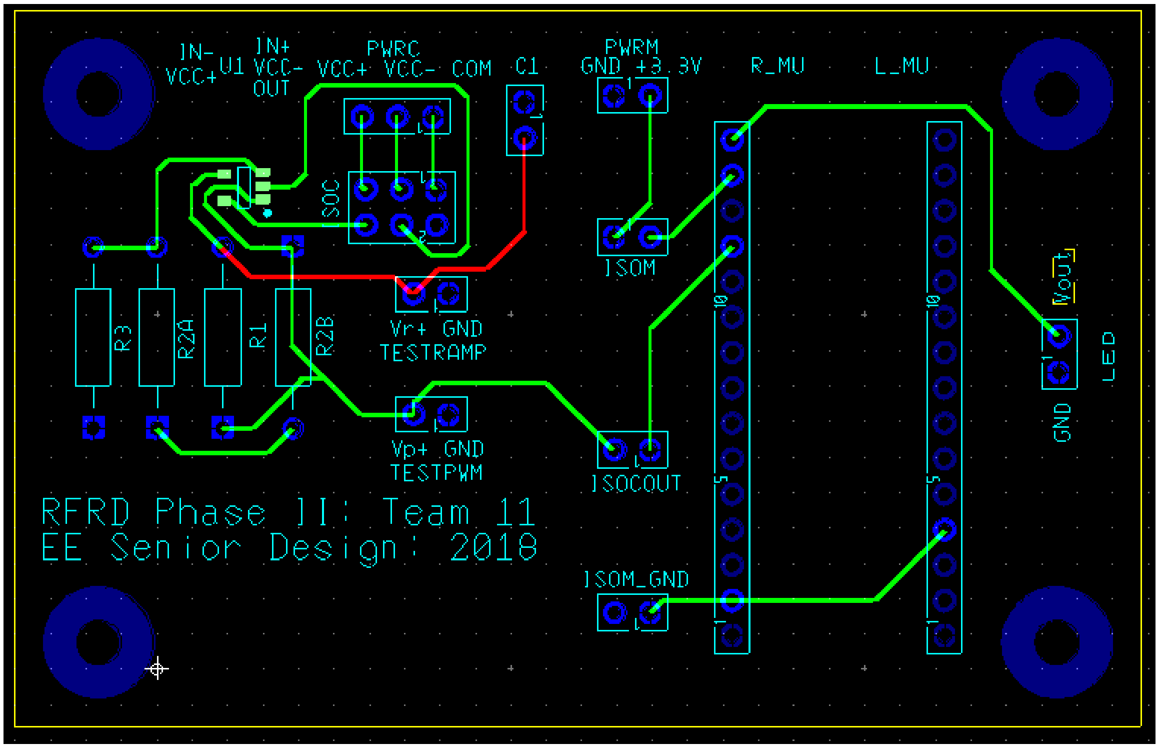
We designed the PCB for multiple isolation points, so we can isolate power from the relaxation oscillator, and microcontroller. We can also isolate the ground from the microcontroller development board and relaxation oscillator output from the microcontroller. This allowed for easier testing of the individual components of the PCB.

We designed two test points for the relaxation oscillator. These test points allow for ease of measuring the ramp and the pulse signal from the relaxation oscillator.

We designed headers for input power for the relaxation oscillator and the microcontroller development board. We also included a shield configuration row of headers for the GPIO pins of the microcontroller development board.

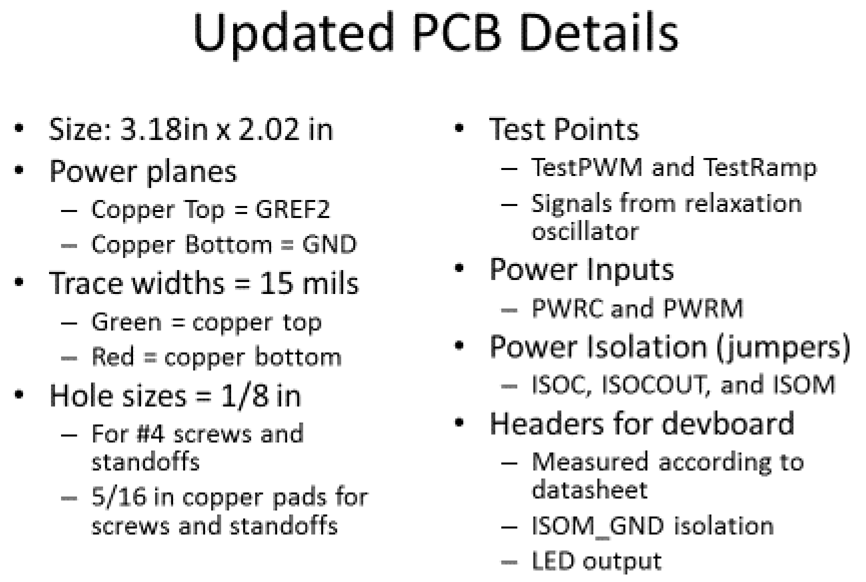
To connect the capacitor to the circuit, insert both leads into the C1 headers. This allows us to connect unbiased capacitors easily. To connect the visual indication LED, insert the anode pin into the Vout and the cathode pin into the GND portions of the LED header.

Figure 8: Ultiboard PCB Layout



We installed mounting holes for standoffs. Using standoffs allows for our circuit to be elevated, which limits exposure to capacitive coupling effects from a lab bench or other exterior components.

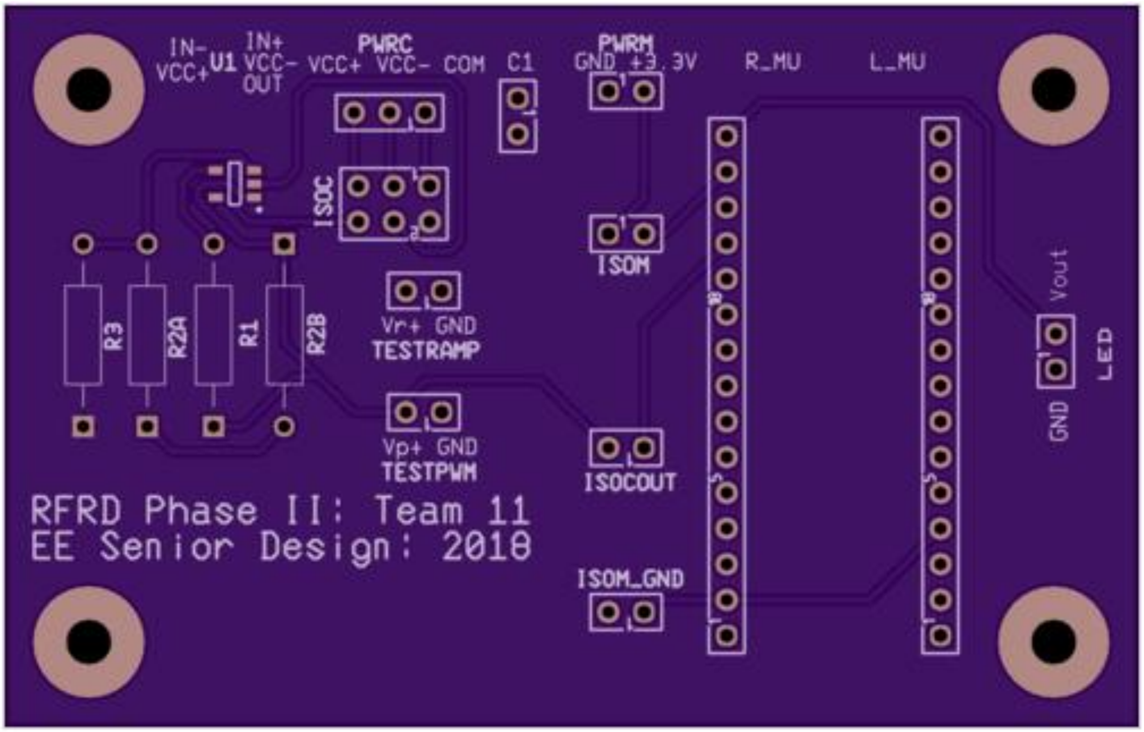
Figure 9: PCB specifications



Updated PCB Details

- Size: 3.18in x 2.02 in
- Power planes
  - Copper Top = GREF2
  - Copper Bottom = GND
- Trace widths = 15 mils
  - Green = copper top
  - Red = copper bottom
- Hole sizes = 1/8 in
  - For #4 screws and standoffs
  - 5/16 in copper pads for screws and standoffs
- Test Points
  - TestPWM and TestRamp
  - Signals from relaxation oscillator
- Power Inputs
  - PWRC and PWRM
- Power Isolation (jumpers)
  - ISOC, ISOCOUT, and ISOM
- Headers for devboard
  - Measured according to datasheet
  - ISOM_GND isolation
  - LED output

Figure 10: PCB Design



Capacitance measurement circuit collaborators: Bailey Akers and Colin Sunderman.

Antenna/Rectifier Design:

Antenna Design:

The design team modeled and simulated an inverted-F antenna PCB to use for our application. We decided on the inverted-F antenna since the power conversion efficiency is the most optimum with our projected size constraints. Our antenna is designed for the operation with a central frequency of 915 MHz.

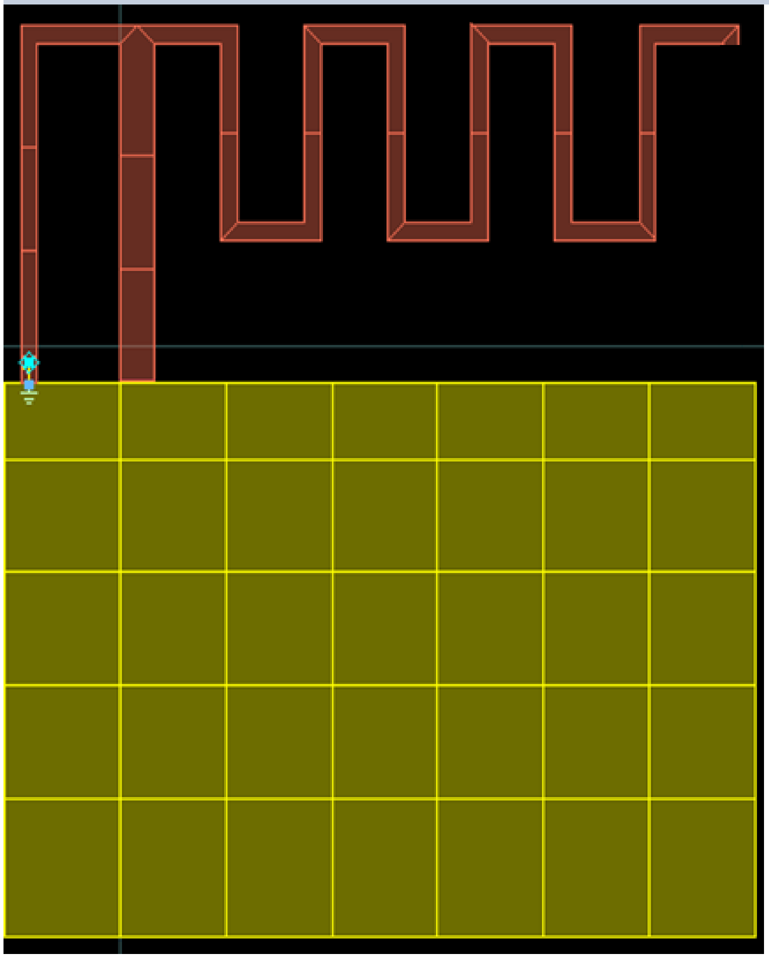
Estimated Antenna Results:

We have estimated an expected received power of ~0.45 mW at 1-meter distance from reader to tag. This estimation was given from the paper, "A Compact Fractal Loop Rectenna for RF Energy Harvesting". In this document they used a fractal loop antenna, while we will be using an inverted-F antenna.

Inverted-F Antenna Design:

We studied the Texas Instruments DN023 antenna and referenced its design while creating our inverted-F antenna layout. To tune the antenna to 915 MHz, we shortened the L6 length to 1 mm.

Figure 11: Antenna Layout in ADS



Power Harvesting Rectifier Design:

We designed a rectifier that will convert a wirelessly received AC input to a DC signal. This DC signal is used to power the microcontroller and capacitance measuring circuitry.

We designed and simulated the rectifier schematic and PCB within Advanced Design System (ADS) software.

Figure 12: Rectifier Schematic in ADS

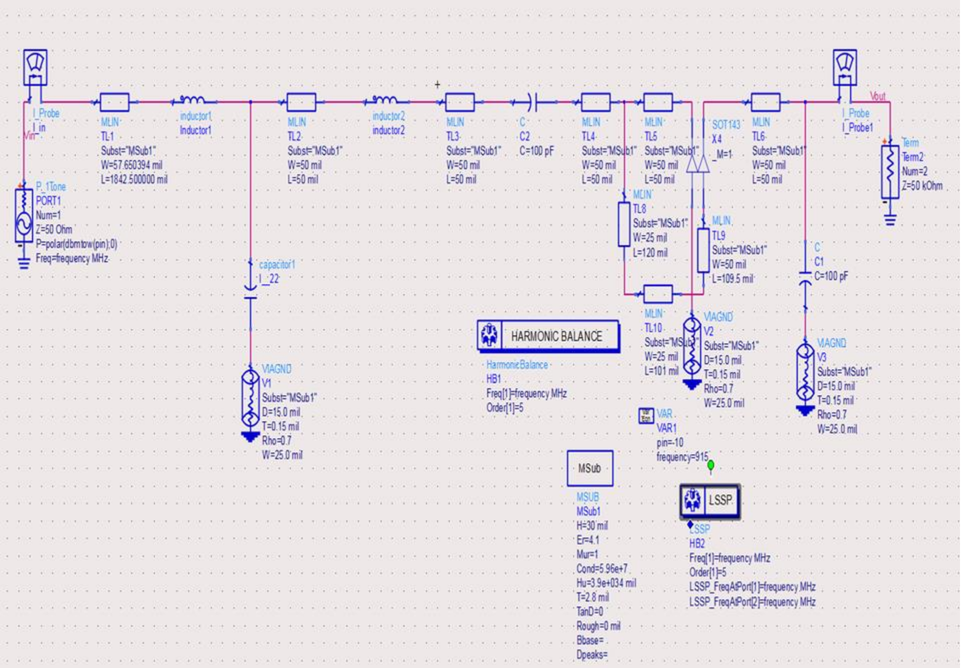


Figure 13: Initial Antenna/Rectifier PCB Layout in ADS

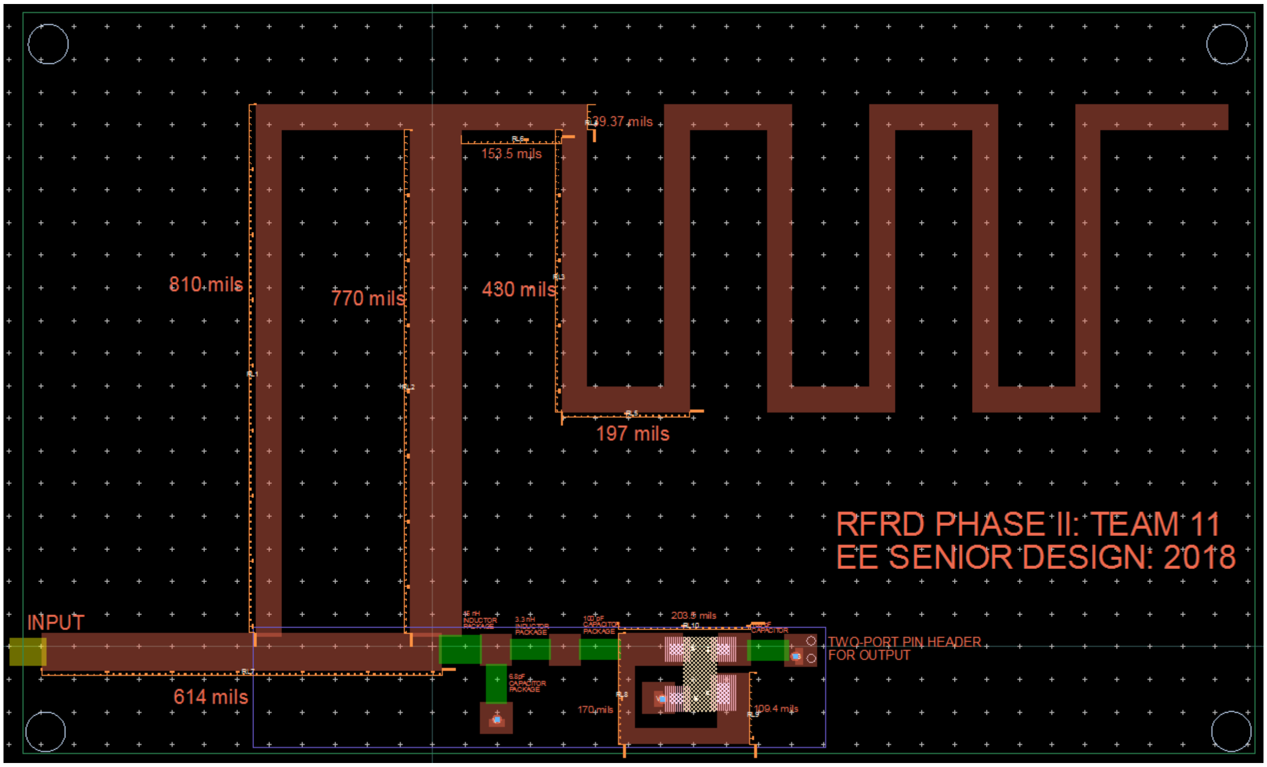
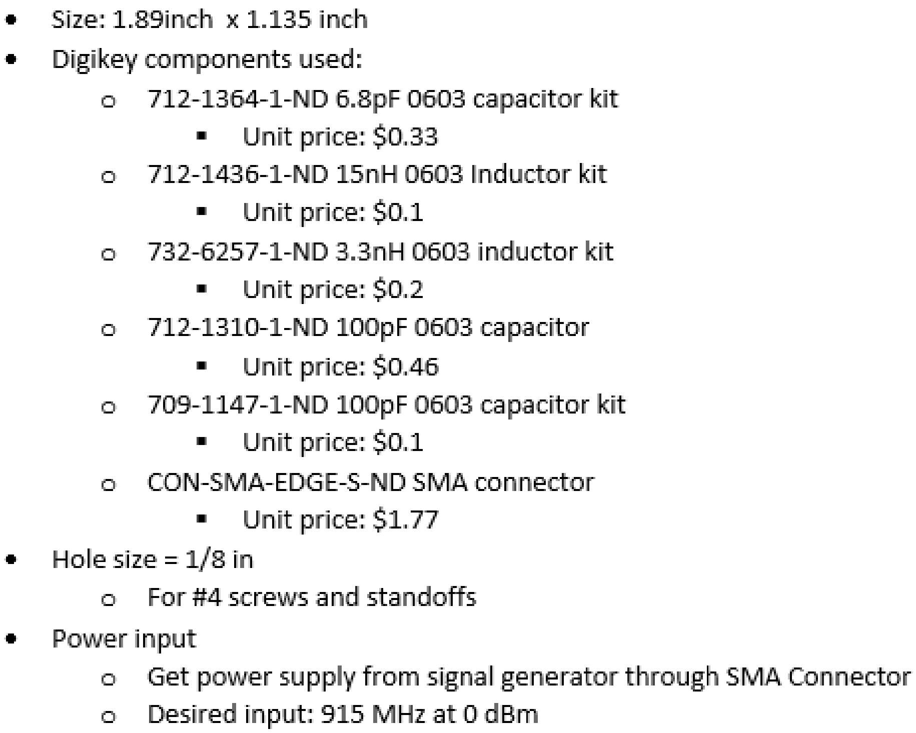


Figure 14: Antenna and Rectifier PCB Details



- Size: 1.89inch x 1.135 inch
- Digikey components used:
  - 712-1364-1-ND 6.8pF 0603 capacitor kit
    - Unit price: $0.33
  - 712-1436-1-ND 15nH 0603 Inductor kit
    - Unit price: $0.1
  - 732-6257-1-ND 3.3nH 0603 inductor kit
    - Unit price: $0.2
  - 712-1310-1-ND 100pF 0603 capacitor
    - Unit price: $0.46
  - 709-1147-1-ND 100pF 0603 capacitor kit
    - Unit price: $0.1
  - CON-SMA-EDGE-S-ND SMA connector
    - Unit price: $1.77
- Hole size = 1/8 in
  - For #4 screws and standoffs
- Power input
  - Get power supply from signal generator through SMA Connector
  - Desired input: 915 MHz at 0 dBm

We didn't fabricate and test this PCB. We do verify the validity of this design with Scott Melvin's antenna and rectifier PCB (noted in the Testing and Implementation section).

Microcontroller:

The microcontroller is utilized for reading and processing the signal received from the capacitance measuring circuit. The microcontroller indicates the state of the capacitance sensor to the user through an LED.

The microcontroller used is the STM32L011K4 from ST's (STMicroelectronics) line of ultra-low power ARM based MCU's. These are ARM Cortex M0+ processors capable of running within the micro-amp range. ST has a simple to use tool chain that allows the software to be easily written or converted to run on other ST microcontrollers.

This microcontroller has many peripheral options that make it easy to work with. The ones that are most important for this project were the low power timer that was used for counting frequency and the hardware UART that was used to display the MCU's clock cycles during testing.

Development Board:

We used ST's Nucleo development board for initial development and the final design. The development board includes voltage regulators to power the MCU and the programer that is needed to load code onto the MCU. After removing two jumpers the microcontroller can be powered from external power source, this was required to operate the microcontroller off of the rectifier DC output.

MCU Power:

The STM32L0 line of MCU's has the ability to operate at very low power levels. In order to operate at the power level that we needed the STM32 was configured in low power run mode at a clock frequency of 65.536 KHz.

Peripheral and Clock Initialization:

The hardware initialization code was generated with STM32CubeMX, a piece of software from ST that allows various hardware parameters to be selected and the initialization code generated based on those parameters. This allows for easy reconfiguration of many parts of the microcontroller which speeds up the prototyping process and removes the need to debug complex initialization sequences.

Figure 15: Configuring Peripherals in CubeMX

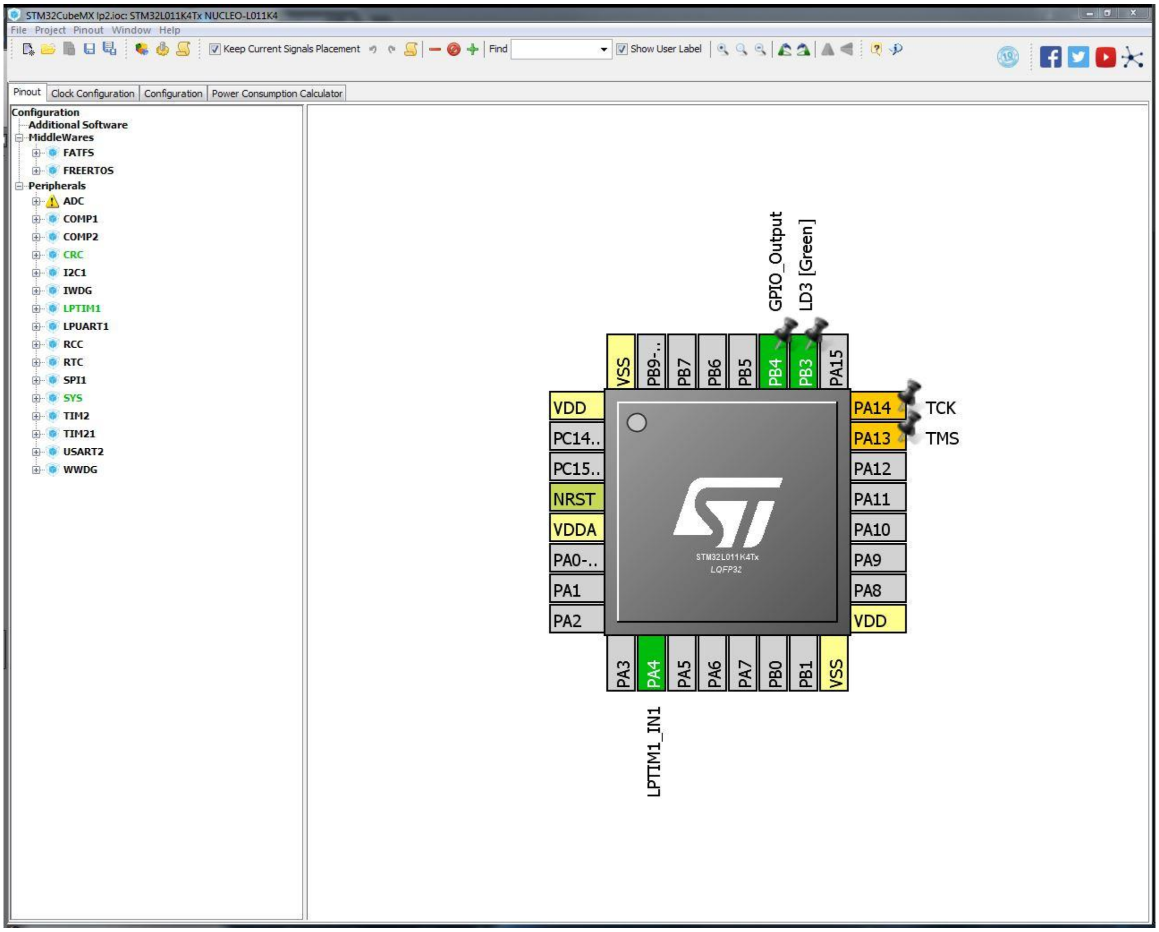
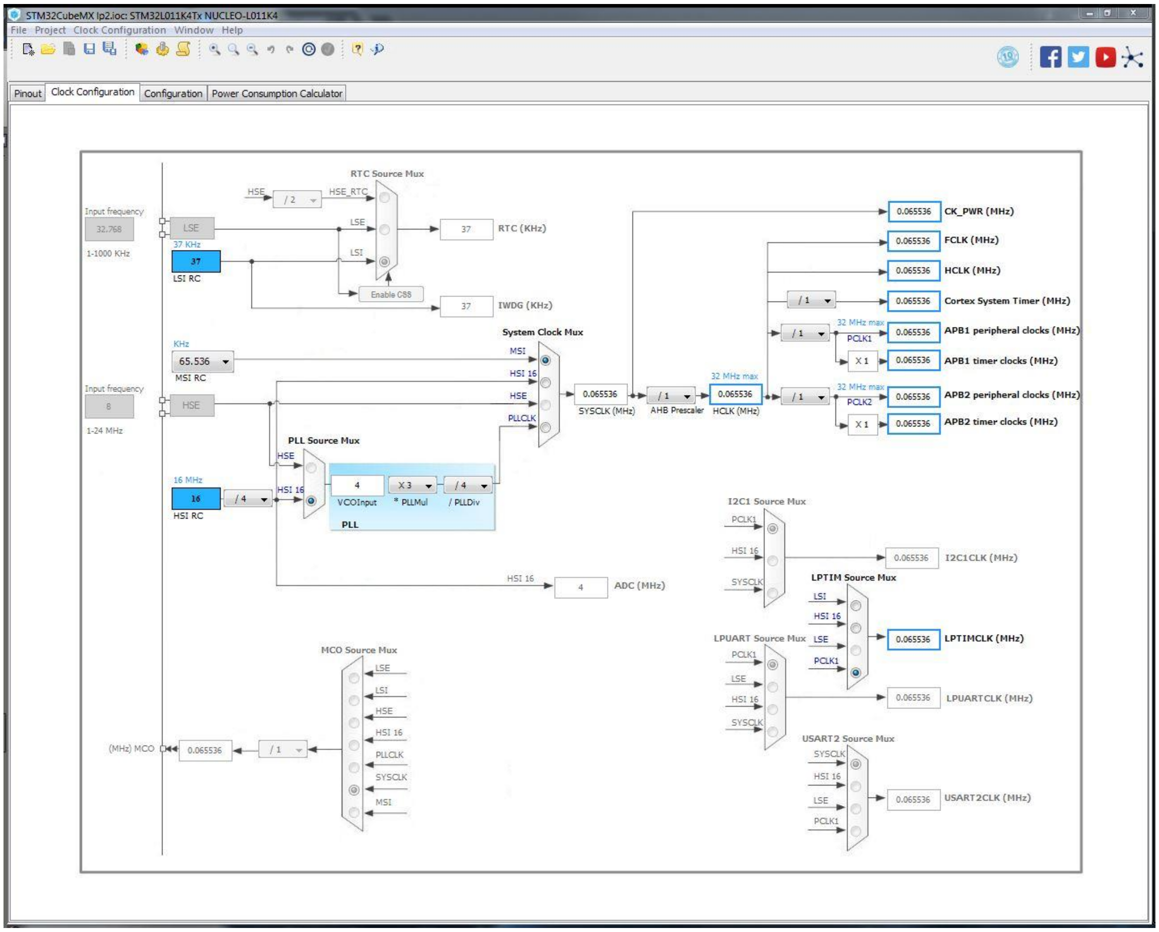


Figure 16: Configuring Clocks in CubeMX



Hardware Interface Library:

ST’s Hardware Abstraction Layer (HAL) was used to control the hardware in the code. This also sped the development time and makes it easy to run the code on a different STM32 microcontroller should the need arise.

Firmware:

The microcontroller measures the frequency of the pulse signal from the capacitance circuit through the use of the low power timer. The timer is used in counter mode and is triggered on the rising edge of each pulse on GPIO PA4. The firmware reads the state of the counter, then waits for 1000 ms and reads the counter value once more. It then subtracts the two to get the total number of pulses in one second. This value is then converted from frequency to period and compared to the threshold that is defined in the code. If the measured value is less than the threshold value then the user indicator LED is activated. This cycle is looped infinitely.

`Firmware with includes and initialization code removed:`

```
#define THRESHOLD_PERIOD 857
#define THRESHOLD_FREQUENCY (1000000/THRESHOLD_PERIOD)
HAL_LPTIM_Counter_Start_IT(&hlptim1, 0xFFFF);
uint16_t first_cap = 0;
uint16_t last_cap = 0;
HAL_PWREx_EnableUltraLowPower();
HAL_PWREx_EnableLowPowerRunMode();

while (1){
	first_cap = HAL_LPTIM_ReadCounter(&hlptim1); //read initial timer count
	HAL_Delay(1000); //wait while number of cycles is counted
	last_cap = HAL_LPTIM_ReadCounter(&hlptim1);  //read second count value
	uint16_t cycles = last_cap - first_cap; //calculate cycles during period
	if (last_cap < first_cap){   //handle overflow of counter
		cycles = (last_cap + 0xFFFF) - first_cap;
	}
	if (cycles > (THRESHOLD_FREQUENCY )){  //if the washer is not pressed
		HAL_GPIO_WritePin(GPIOB, GPIO_PIN_3, GPIO_PIN_RESET); //LED off
	} else {
		HAL_GPIO_WritePin(GPIOB, GPIO_PIN_3, GPIO_PIN_SET); //LED on
	}
}
```

3 Testing and Implementation

3.1 INTERFACE SPECIFICATIONS

We first simulated the relaxation oscillator and inverted-F antenna designs in software. This enabled us to fix errors before manufacturing the PCB's.

We then tested the relaxation oscillator in hardware using the breadboard and PCB design. We then simulated the microcontroller software and interfaced the microcontroller with the relaxation oscillator breadboard and PCB design.

Lastly, we interfaced our capacitance measuring/microcontroller PCB with Scott Melvin's antenna/relaxation oscillator PCB to demonstrate wireless powering of our capacitance measuring/microcontroller PCB.

3.2 HARDWARE AND SOFTWARE

Capacitance Measuring Circuit:

We used OrCad Capture SPICE simulation in order to simulate our relaxation oscillator with STMicroelectronics op amps. We used National Instruments Multisim and Ultiboard to generate our PCB design.

Antenna/Rectifier:

We used ADS software to simulate the inverted-F antenna to simulate the antenna frequency output. We used ADS to create and simulate the rectifier design power output and PCE. We used ADS to create the antenna and rectifier PCB layouts.

Microcontroller:

We used a ST STM32L011K4 microcontroller to measure the state of the capacitance circuit. The firmware was written with the System Workbench for STM32 IDE and STM32CubeMX for initialization code generation.

3.3 PROCESS

Capacitance Measuring Circuit:

We tested our relaxation oscillator design in OrCad Capture by calculating the period of oscillation for our chosen components, then measuring the period of oscillation that was simulated. We then tuned our resistance values to obtain an accurate to theoretical reading of capacitance in our simulation. We updated the reference voltages for the comparator IC in order for the output to be in the positive voltage region of operation and for connectivity with the microcontroller and simulated the output.

We then built the relaxation oscillator on the breadboard. We measured the Period vs Capacitance and the Power vs Voltage for the breadboard.

We then interfaced the breadboard circuit to the microcontroller. At this time the microcontroller was programmed to output the clock cycles measured by the GPIO interrupts.

We then built the relaxation oscillator PCB. We first tested how changing the input capacitor effect the period of the output because this is the main focus of the circuit. We did this by measuring the minimum, maximum, and average period for different capacitances using an oscilloscope. We then tested the effect that different voltages have on the period of the circuit. This test let us see how the circuit would output if the power received by the antenna was varying. We did two versions of this test. In the first version we used a positive Vcc+ value and kept the COM value at half of Vcc+. The second version COM was set at zero, a positive value was used for Vcc+, and a negative value for Vcc-.

Inverted-F Antenna:

We simulated the inverted-F antenna design using ADS in order to obtain the operation frequency of the antenna. We then simulated the rectifier circuitry in ADS and obtained the simulated power output and PCE.

Microcontroller:

There were three stages to testing the microcontroller. After the first draft of the firmware was complete we tested the microcontroller with the capacitance measuring PCB. For this test we left the microcontroller in full power run and used UART to read the period that the microcontroller was detecting. This was an initial test that we used to look for any irregularities or deviations in the measured period from what we expected.

We then set the microcontroller to operate in low power run mode with a clock speed of 65.536 kHz. We then measured the power draw and tested operation at various voltages. For this we used a 220 ohm resistor in line with a variable power supply and measured the voltage drop across the resistor. We then varied the voltage across the operating range of the MCU (1.5-3.6V). We took note of the lowest voltage were the MCU would reliably turn on and run.

We then verified that everything operated properly while in low power run. We were unable to use the UART in low power mode so we used a set threshold that when crossed would trigger a GPIO. We set the threshold to different values within the operating range of the capacitance circuit and verified that the GPIO triggered at the proper point.

System Testing:
We interfaced the capacitance measuring and microcontroller PCB with an inverted-F antenna and rectifier PCB designed by Scott Melvin, ECpE graduate student at ISU. We demonstrated system level functionality of wirelessly powering and functionality of our capacitance measuring circuit and microcontroller PCB.

3.4 RESULTS

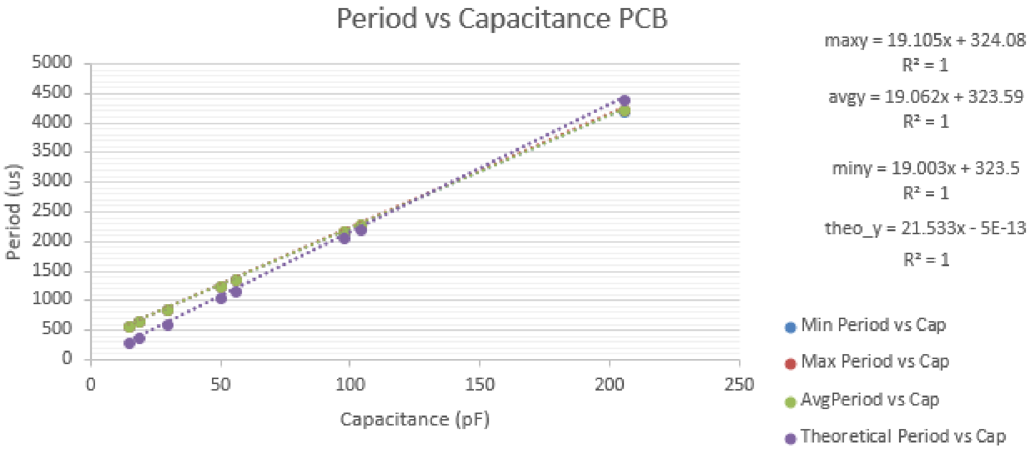
Capacitance Measuring Circuit:

Results for the SPICE simulation of the relaxation oscillator are located within Appendix 2.

Results for the simulation of the breadboard relaxation oscillator are located within Appendix 2.

First, we tested the PCB for the change in period with different capacitors. This test was done using eight capacitor values and maintaining constant voltage of Vcc+ = +2V, Vcc- = 0V, and COM = +1V. The maximum, minimum, and average period were measured using an oscilloscope. These values are plotted on the graph below along with the theoretical value.

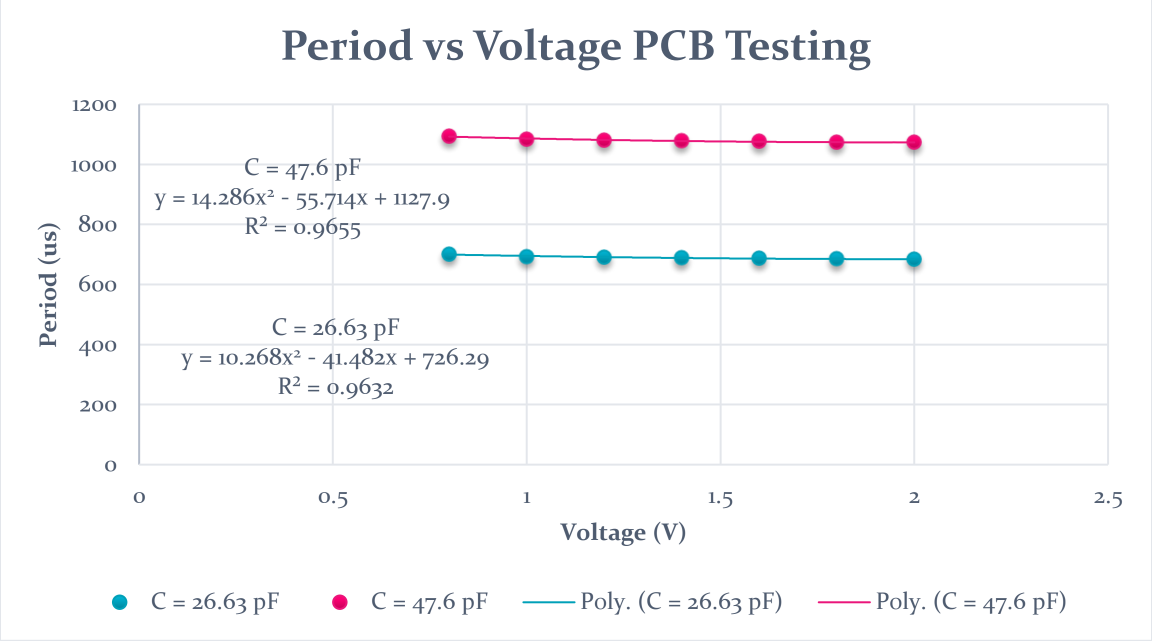
Figure 17: Period vs Capacitance PCB Testing



The results of this test verify that there is a linear relationship between period and capacitance for our PCB circuit. It also shows that this is a minimal difference between the maximum, average, minimum, and theoretical values. These results were used to program the microcontroller to correctly calculate the period and to show what level of variation in period can be expected at a given capacitance.

The second test was to see the effect voltage had on period at a constant capacitance. The input voltage was decreased while keeping a constant ratio between Vcc+ and the COM value. Measurements of the average period were taken at each voltage level and recorded. This tested was performed for both capacitors of 47.6 pF and 26.63 pF.

Figure 18: PSRR Testing of the PCB

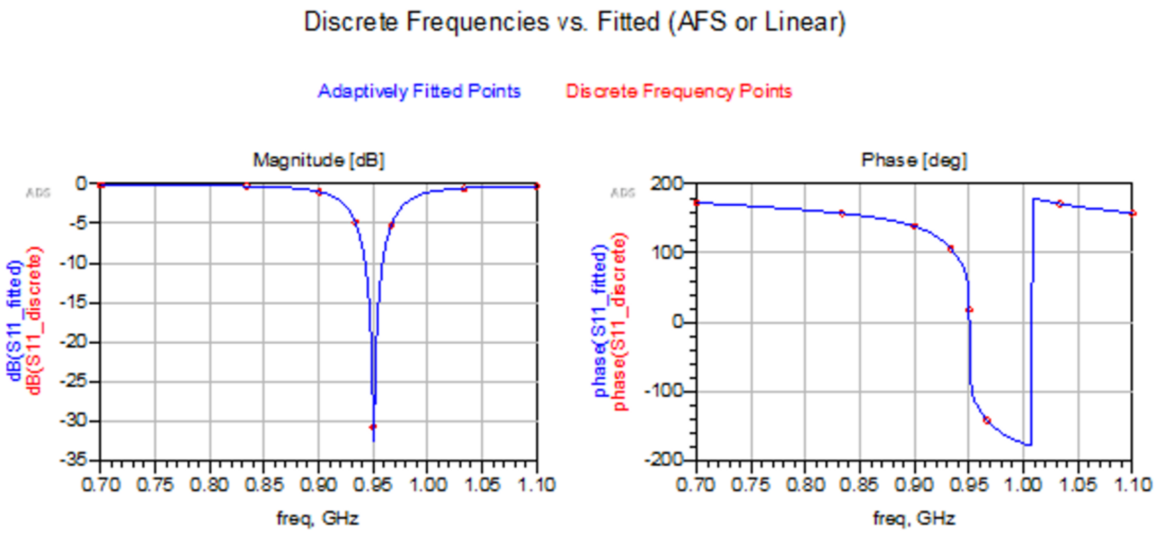


The results of this test show that a change in voltage has a minimal change in period. This is a good sign for the final project because there is a high possibility that power received by this circuit will vary due to the power harvesting. A circuit that's output has minimal period change means that we can expect a more reliable indication of capacitance.

Inverted-F Antenna Layout Simulation Results:

Figure 19: Antenna Layout Simulation from ADS

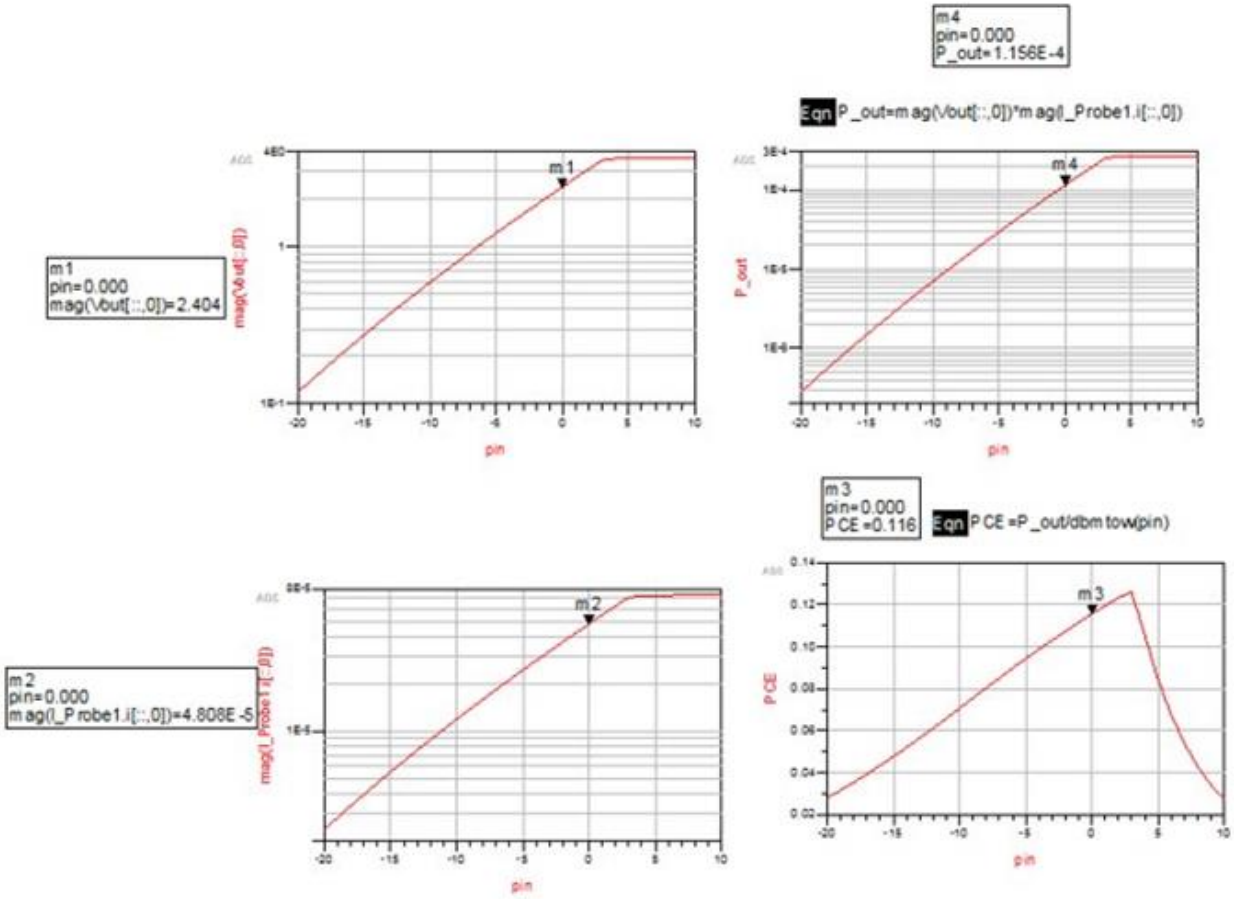
Tuning changing length of L6 of Antenna



Antenna Layout Simulation Results:
These results show that the central frequency of our antenna is 950 MHz. Though we can operate this antenna at 915 MHz as proposed.

Power Harvesting Rectifier Simulation Results:

Figure 20: Rectifier Schematic Simulations from ADS



Rectifier Simulation Results:

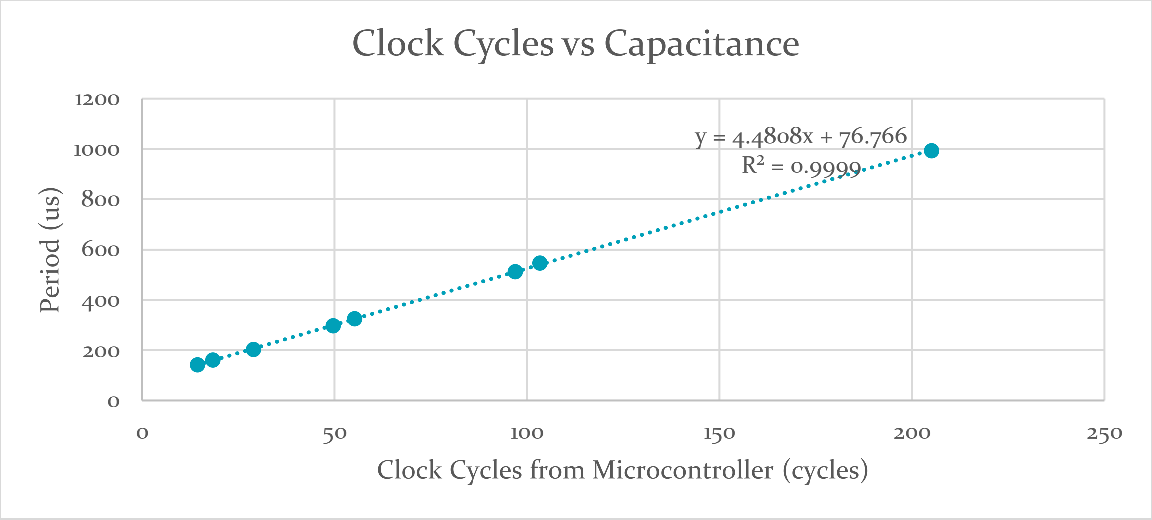
Our rectifier design simulated an output voltage of 2.404 volts and output power of 115.6 uW at 0 dBm received power. This is shown in the top right graph.

Our simulation also showed a power conversion efficiency ratio (PCE) of 0.116 for the rectifier design. This is shown in the bottom right graph.

Microcontroller:

After the initial revision of the firmware was complete we tested the microcontroller's operation with the capacitance circuit. We received the measured number of clock cycles that elapsed per period from the MCU through UART. We then entered them into a spreadsheet to ensure it operated properly over the entire operating range of the capacitance circuit. Below is our results:

Figure 21: Microcontroller Measured Clock Cycles vs Capacitance Measured



We found no issues with the integration of the capacitance measuring circuit and microcontroller.

We measure the voltage drop across the 220 ohm resistor to be 4.8 mV while running the final version of the code. This equates to an amperage draw of just under 22 uA, this value was constant at different voltages. In terms of power consumption, these measurements prove that the microcontroller in run mode draws less than 50 uW.

The MCU is rated to run at 1.65 to 3.6 V but we found that stability could not be guaranteed when operating below 1.8 V. At voltages below 1.8V the MCU did not reliably turn on, resulting in the minimum safe operating voltage for the circuit to be 1.8V.

System Testing Results:

For the following system level testing we interfaced our capacitance measuring and microcontroller PCB with an inverted-F antenna and rectifier PCB designed by Scott Melvin, ECpE graduate student at ISU. The goal is to demonstrate system level functionality of wirelessly powering our capacitance measuring circuit and microcontroller.

Scott Melvin's design used an SMA connection to get a signal from a signal generator. This AC signal is generated and send to the transmitting inverted-F antenna. It is then received with an identical inverted-F antenna at a fixed distance of 3.25 inches.

The received AC signal is then sent to the rectifier circuit to be converted from an AC to a DC voltage. The DC voltage is then fed directly to our capacitance measuring and microcontroller PCB.

This rectifier circuit was designed with a center frequency of 848 MHz. This is important to note because in our tests we transmit a signal at 887 MHz to simulate ISM band frequencies. This results in an inefficient transfer of power and a higher dBm to be sent by the signal generator.

Testing Procedure:

We first powered the capacitance measuring circuit with the rectifier. We then powered the capacitance measuring circuit with the transmit/receive antennas and rectifier. We then powered the system (capacitance measuring circuit and microcontroller) with the transmit/receive antennas and rectifier.

Figure 22: Coupling the Antenna/Rectifier PCB to the Capacitance Measuring/Microcontroller PCB

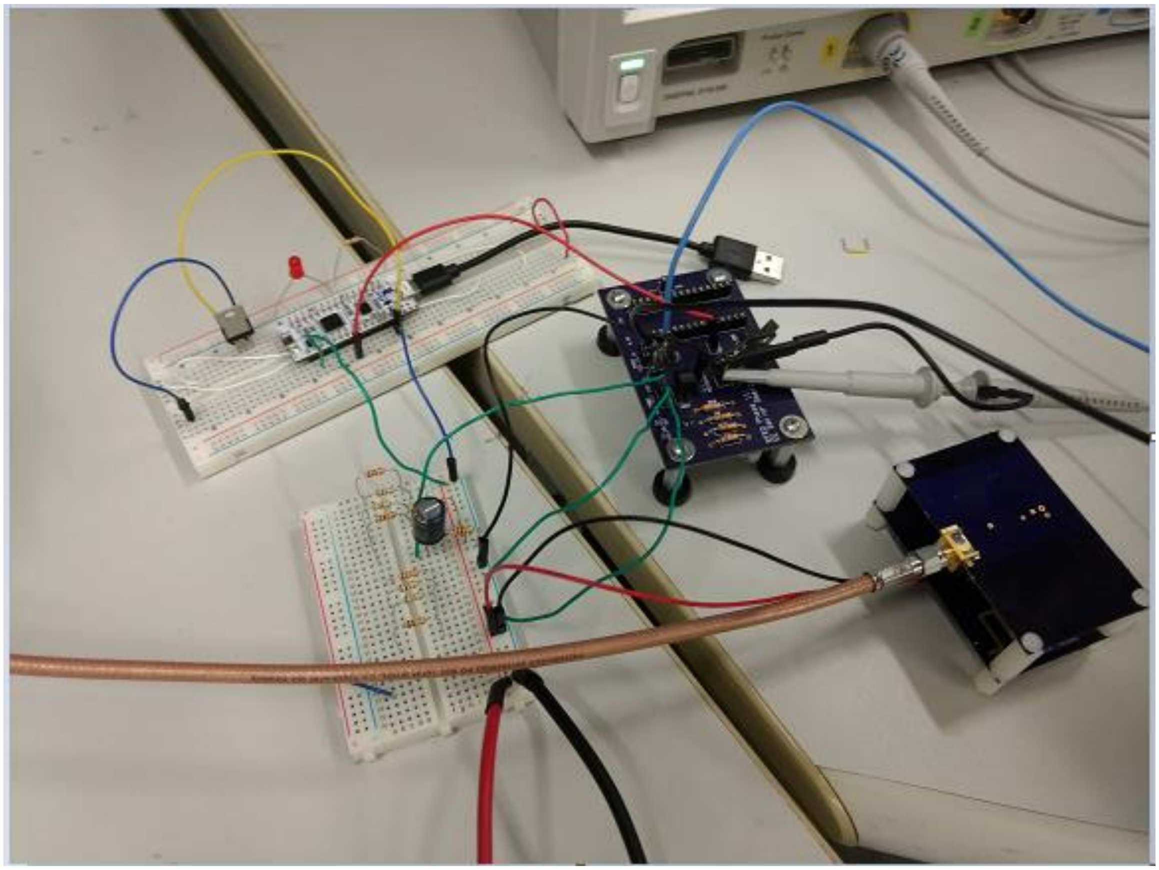
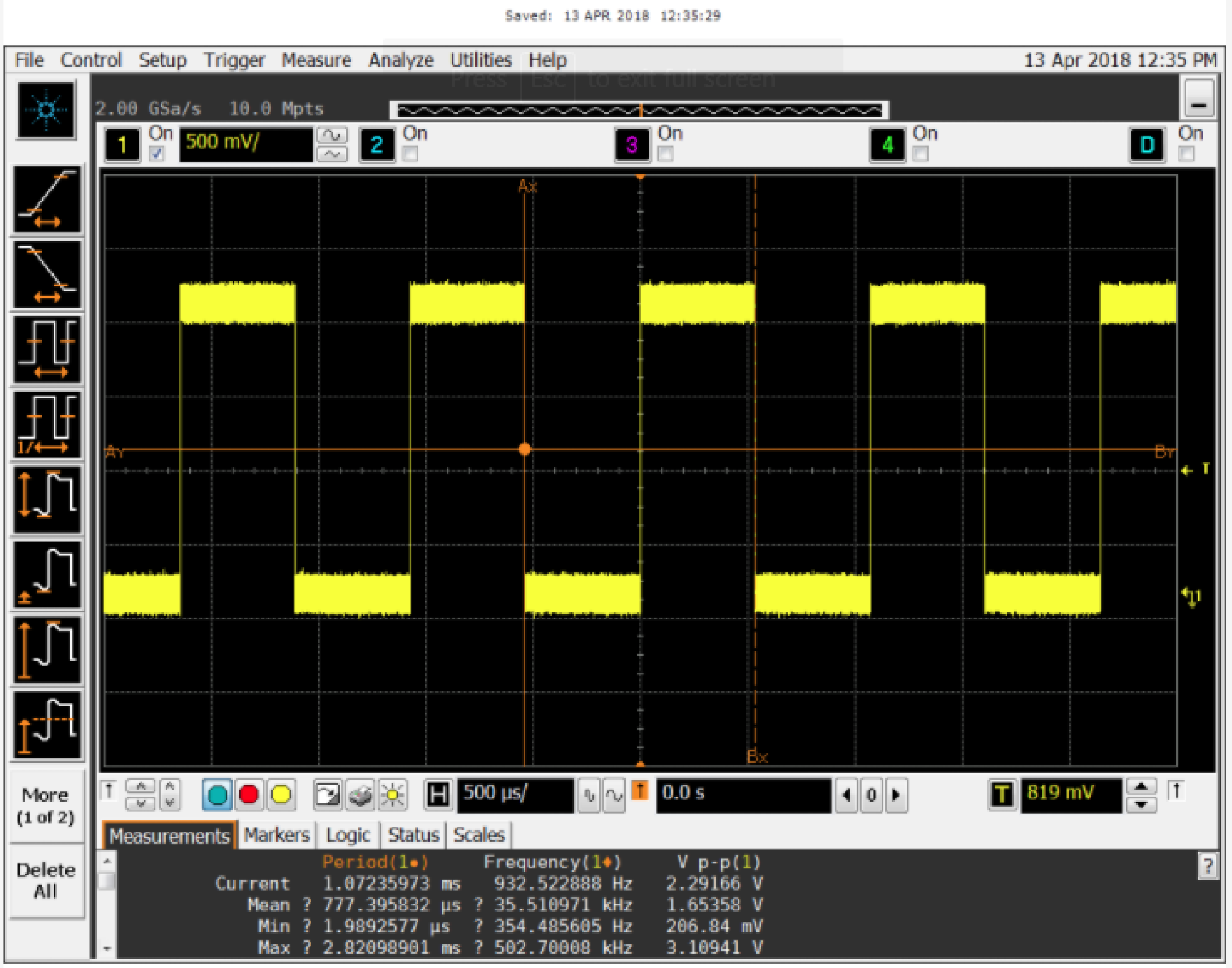


Figure 23: System Test Capacitance Measuring Circuit Output (C = 47pF)

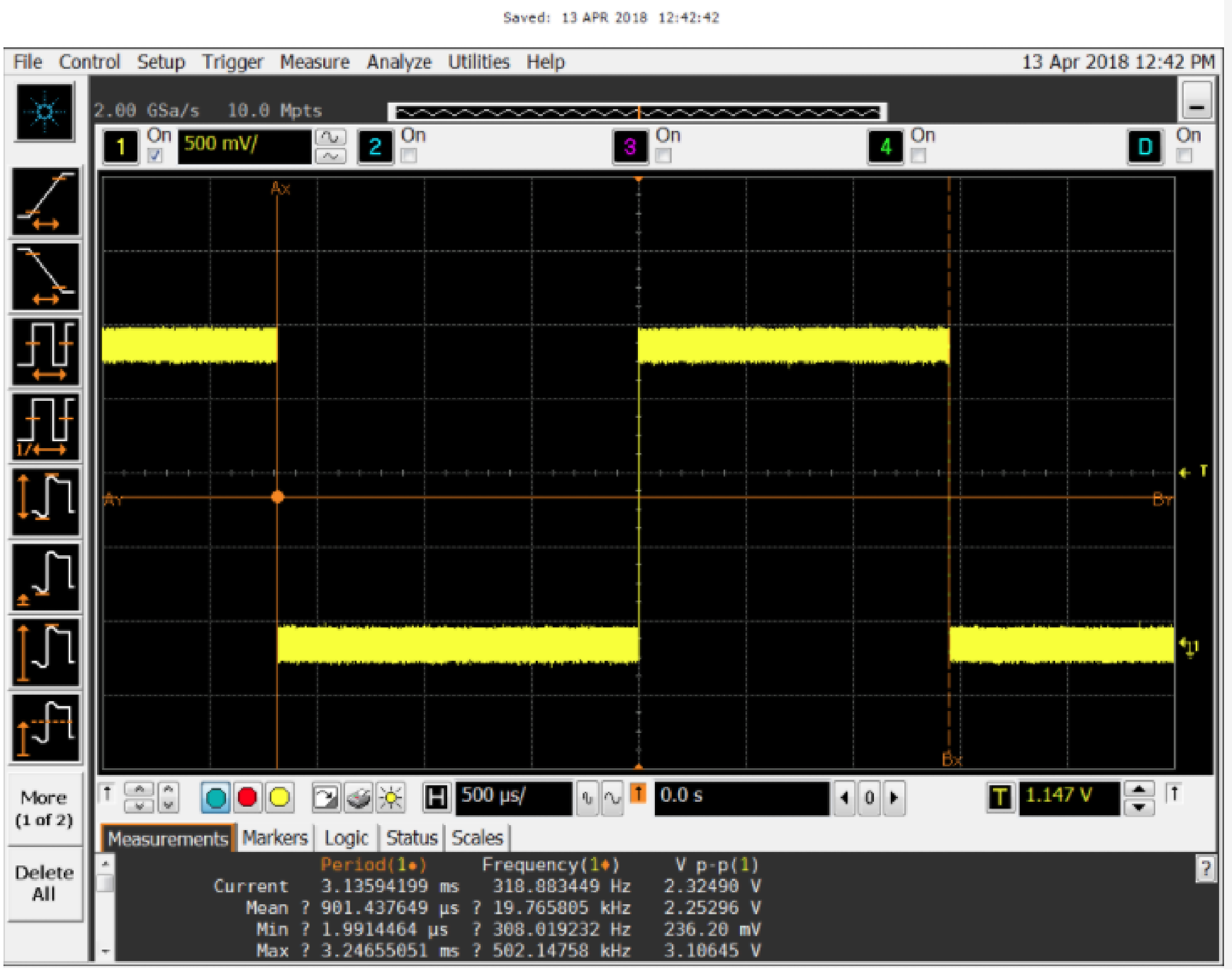


```
           Period(1•)       Frequency(1•)    V p-p(1)
Current    1.07235973 ms    932.522888 Hz    2.29166 V
   Mean  ? 777.395832 µs  ? 35.510971 kHz    1.65358 V
    Min  ? 1.9892577 µs   ? 354.485605 Hz    206.84 mV
    Max  ? 2.82098901 ms  ? 502.70008 kHz    3.10941 V
```

The above figure shows that we were receiving accurate period output for a 47 pF capacitor. In PCB testing we measured a period of 107.3 us with a different oscilloscope.

The above tests were performed with the signal generator at 848 MHz and 15.5 dBm power transmitted to the transmitting antenna.

Figure 24: System Test Capacitance Measuring Circuit Output (Resting Washer)

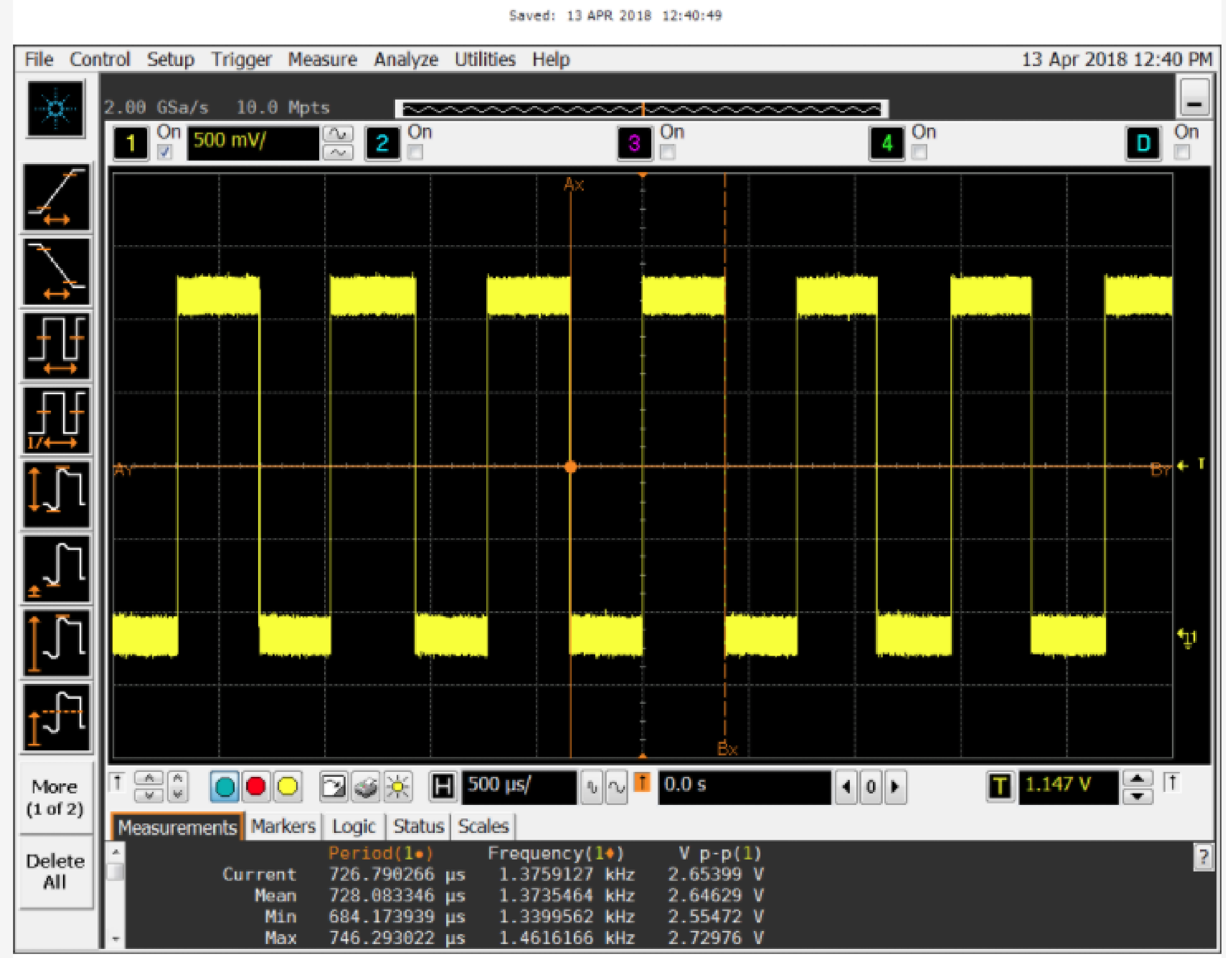


```
           Period(1•)        Frequency(1•)     V p-p(1)
Current   3.13594199 ms    318.883449 Hz    2.32490 V
   Mean ? 901.437649 µs ? 19.765805 kHz    2.25296 V
    Min ? 1.9914464 µs  ? 308.019232 Hz    236.20 mV
    Max ? 3.24655051 ms ? 502.14758 kHz    3.10645 V
```

With a resting washer the period isn't guaranteed as we see above. This is due to unpredictable effects with the mechanical design. While testing, the washer design and its coating seems to couple with the lab benches and other materials.

In order to receive an accurate capacitance reading of the washer design it needs to be resting on a non-conductive material.

Figure 25: System Test Capacitance Measuring Circuit Output (Pressured Washer)



```
         Period(1•)        Frequency(1•)     V p-p(1)
Current  726.790266 µs     1.3759127 kHz     2.65399 V
   Mean  728.083346 µs     1.3735464 kHz     2.64629 V
    Min  684.173939 µs     1.3399562 kHz     2.55472 V
    Max  746.293022 µs     1.4616166 kHz     2.72976 V
```

When we pressure the washer, we expect a capacitance of 25 to 30 pF. According to our results, a period of 726 us is expected since our PCB testing results showed a period of 680 us for a 26.43 pF capacitor. Which means we're slightly above that capacitance level.

This error may be due to the washer coupling with other materials or other undesirable effects.

System Test Capacitance Measuring Circuit Output with Washer Comments:

The goal of this test is to show the full functionality of our design with the washers. For this test we increased the frequency of the signal generator to 887 MHz to approach ISM band. In order to guarantee a 2-volt input to our design we outputted +15.5 dBm to the transmitting inverted-F antenna.

System Test Capacitance Measuring Circuit Output with Washer Results:

We were able to demonstrate the full functionality of our design with the washers. We were also able to see visual indication from the microcontroller onboard LED that our set point was being detected. We had it programmed to activate the LED when the washer is being pressured, and to not activate the LED when the washer is resting.

4 Closing Material

4.1 CONCLUSION

The goal of this project was to make a RFRD prototype that wirelessly checks the structural integrity of anchor bolts. The prototype consists of a low power capacitance measuring and microcontroller PCB, rectifier, inverted-F antenna, and visual indication of tightness for the user.

Capacitance Measuring PCB:

We successfully designed, fabricated, tested, and demonstrated wireless powering of our low power (44.5 uW) relaxation oscillator PCB. We demonstrated the precision of capacitance measurement and it's functionality with the provided mechanical design while being wirelessly powered by Scott Melvin's antenna/rectifier PCB.

In the future this design is scalable for a much smaller PCB footprint. Further scaling can be made by using all surface mount parts and using the base microcontroller in a smaller package rather than the development board.

Antenna Design:

We simulated both the inverted-F antenna and the rectifier circuitry. The antenna simulations simulated wireless transfer of power at ISM frequencies. The rectifier simulations simulated wireless powering of 115.6 uW at 0 dBm. This simulated power output is sufficient to power the low power capacitance measuring and microcontroller PCB.

We designed a rectifier and antenna PCB. We did not get to fabricate and test the PCB hardware.

We were able to demonstrate the functionality of these designs through Scott Melvin's antenna/rectifier PCB. Even though we didn't design this PCB, we were able to prove that with further development we would be able to fabricate a functional wireless power source for the capacitance measuring/microcontroller PCB.

Microcontroller Design:

We successfully programmed and tested the low power (below 50 uW) microcontroller functionality in interfacing with the capacitance measuring circuit. We successfully integrated the capacitance measuring circuit and STM32 microcontroller development board within an all-encompassing PCB design.

We were able to successfully slow the clock rate and enter low power run to ensure wireless powering of the MCU. We successfully demonstrated the functionality of the MCU's timers/counter, firmware, and LED indication output while being wirelessly powered by Scott Melvin's antenna/rectifier PCB.

In future revisions other functionality could be added to the microcontroller. A digital communications method could return data to the user over RF and persistent tag IDs could allow tracking of the washer over time and make management of bolt testing easier.

4.2 REFERENCES

Capacitance Measuring Resources:

Liu, Yili, et al. “Limitations of a relaxation oscillator in capacitance measurements.” IEEE Transactions on Instrumentation and Measurement, vol. 49, no. 5, 2000, pp. 980–983., doi:10.1109/19.872917.

Tuttle, Dr. Gary. “Non Linear Oscillators.” EE 230 Website, Dr. Gary Tuttle, tuttle.merc.iastate.edu/ee230/topics/op_amps/non_linear_oscillators.pdf.

“TS881 Rail-To-Rail 0.9 V nanopower comparator.” www.st.com/content/ccc/resource/technical/document/datasheet/a2/60/3e/5d/b2/c1/4a/e9/DM00057901.pdf/files/DM00057901.pdf/jcr:content/translations/en.DM00057901.pdf.

Antenna Resources:

Zeng, Miaowang, et al. “A Compact Fractal Loop Rectenna for RF Energy Harvesting.” IEEE Antennas and Wireless Propagation Letters, vol. 16, 2017, pp. 2424–2427., doi:10.1109/lawp.2017.2722460.

Kervel, Fredirk. “868 MHz, 915 MHz and 955 MHz Inverted F Antenna.” www.ti.com/lit/an/swra228c/swra228c.pdf.

Technologies, Avago. “Linear Models for Diode Surface Mount Packages.”www.avagotech.com, 21 July 2010, ddd.uab.cat/pub/trerecpro/2010/hdl.../PFC_SergiCarreraColet_annex.pdf.

Microcontroller Sources:

STMicroelectronics. “Access Line Ultra-Low-Power 32-Bit MCU Arm®-Based Cortex®-M0 , up to 16KB Flash, 2KB SRAM, 512B EEPROM, ADC.” www.st.com/resource/en/datasheet/stm32l011d4.pdf.

STMicroelectronics. “STM32 Nucleo-32 board.” http://www.st.com/resource/en/data_brief/nucleo-f031k6.pdf.

4.3 APPENDICES

Appendix I – Operation Manual

This section outlines how to use the entire system, including the relaxation oscillator and the microcontroller. First ensure that all the isolation points have jumpers connected across the pins. This should include pins labeled ISOC, ISOM, ISCOUT, and ISOM_GND. Then connect the washers or the capacitor that is being testing into the female pins labeled C1. Only use unbiased capacitors with this system.

Next plug in the power supplies for the relaxation oscillator into the proper pins. They are next to the PWRC label. VCC+ is the pin for the positive supply for the comparator. VCC- is the pin for the negative supply for the comparator. COM is the reference voltage that is used for comparisons. For most testing of this circuit VCC+ = +2V, VCC- = 0V, and COM = +1V.

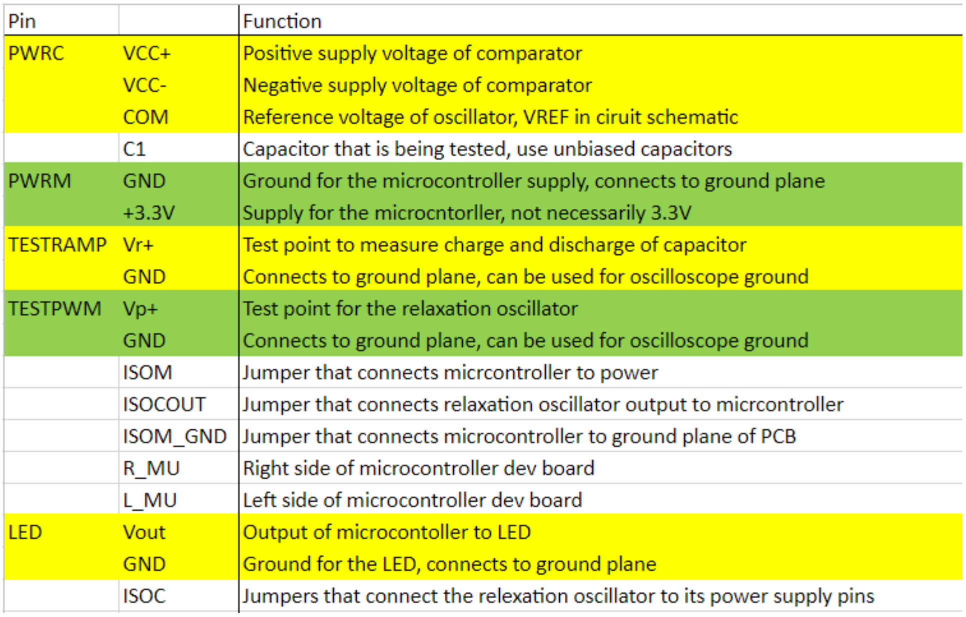
There is not a ground pin by the PWRC pins but it is also important to connect the ground of the power supply to one of the GND pins so that the ground plane is sharing a common ground with the power source. The ground could be connected to the GND pin by TESTRAMP, TESTPWM, PWRM, or LED. It is also important to note that while VCC- is generally used as 0V and needs to be grounded, this pin is not connected to the ground plane. This allows for it to be possible to use other values at this pin. So, to properly have the grounds connected when using VCC- = 0, VCC-, the power supply ground, and a GND pin connected to the ground plane should all be connected together.

Next the power needs to be connected for the microcontroller. The pins for the microcontroller are next to the label PWRM. GND is the ground for the microcontroller power supply and is connected to the ground plane. This is not connected to the microcontroller's ground pin. The pin labeled +3.3V is the input power. Although it is labeled 3.3V it is possible to use other voltages including the 2V used to supply the relaxation oscillator. To share power with the relaxation oscillator, connect the VCC+ pin to the +3.3V pin. To use a separate power supply for the microcontroller, connect the input to the +3.3V pin and the ground to the GND pin.

Once the power supplies are connected and are turned on the circuit should be operating. The microcontroller's onboard LED should be on for capacitances less than the capacitance designated in the microcontroller code. For most testing this value was 30 pF.

Below is a table describing the functions of the pins on the PCB.

Figure 26: Pins and Functions on PCB Design



Pin		Function
PWRC	VCC+	Positive supply voltage of comparator
	VCC-	Negative supply voltage of comparator
	COM	Reference voltage of oscillator, VREF in ciruit schematic
	C1	Capacitor that is being tested, use unbiased capacitors
PWRM	GND	Ground for the microcontroller supply, connects to ground plane
	+3.3V	Supply for the microcntorller, not necessarily 3.3V
TESTRAMP	Vr+	Test point to measure charge and discharge of capacitor
	GND	Connects to ground plane, can be used for oscilloscope ground
TESTPWM	Vp+	Test point for the relaxation oscillator
	GND	Connects to ground plane, can be used for oscilloscope ground
	ISOM	Jumper that connects mircontroller to power
	ISOCOUT	Jumper that connects relaxation oscillator output to mircontroller
	ISOM_GND	Jumper that connects microcontroller to ground plane of PCB
	R_MU	Right side of microcontroller dev board
	L_MU	Left side of microcontroller dev board
LED	Vout	Output of microcontoller to LED
	GND	Ground for the LED, connects to ground plane
	ISOC	Jumpers that connect the relexation oscillator to its power supply pins

Appendix II – Alternative Designs:

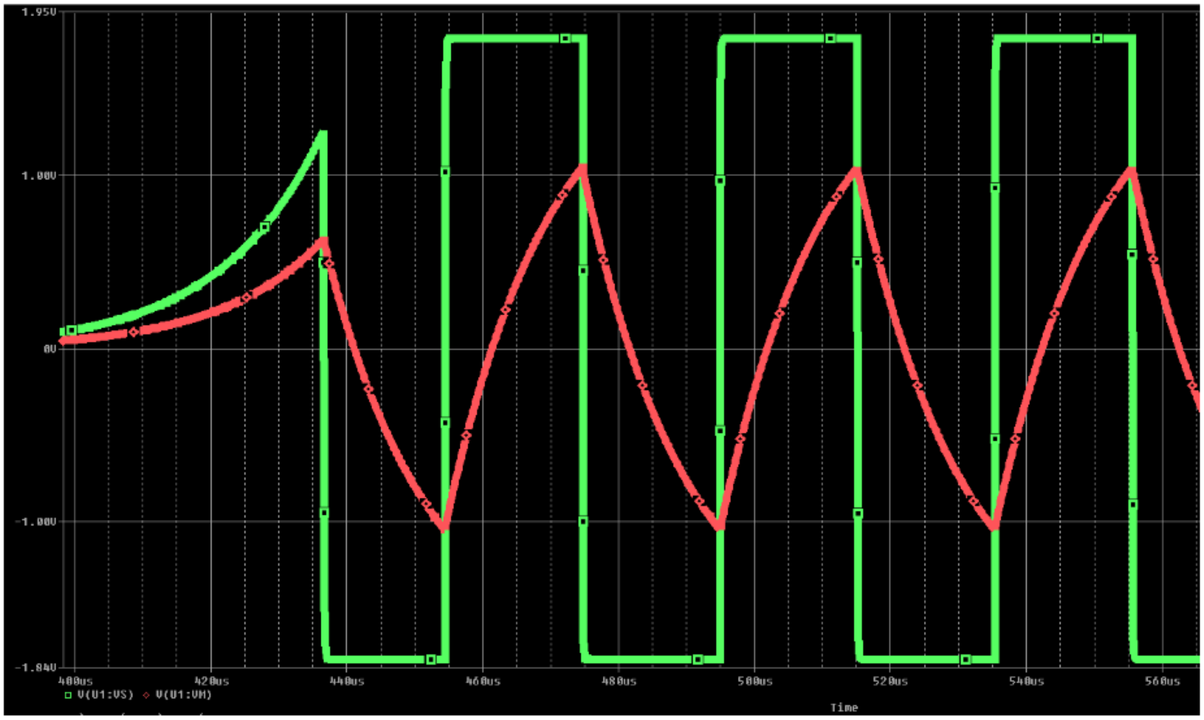
We first referenced a relaxation oscillator design in the published IEEE article, "Limitations of a Relaxation Oscillator in Capacitance Measurements" by Yili Liu, Song Chen, Masakatsu Nakayama, and Kenzo Watanabe.

We decided this design wasn't sufficient for our application. This decision was made after SPICE simulations of this design weren't able to reciprocate theoretical behavior proposed in the paper.

Appendix II – Additional Simulations:

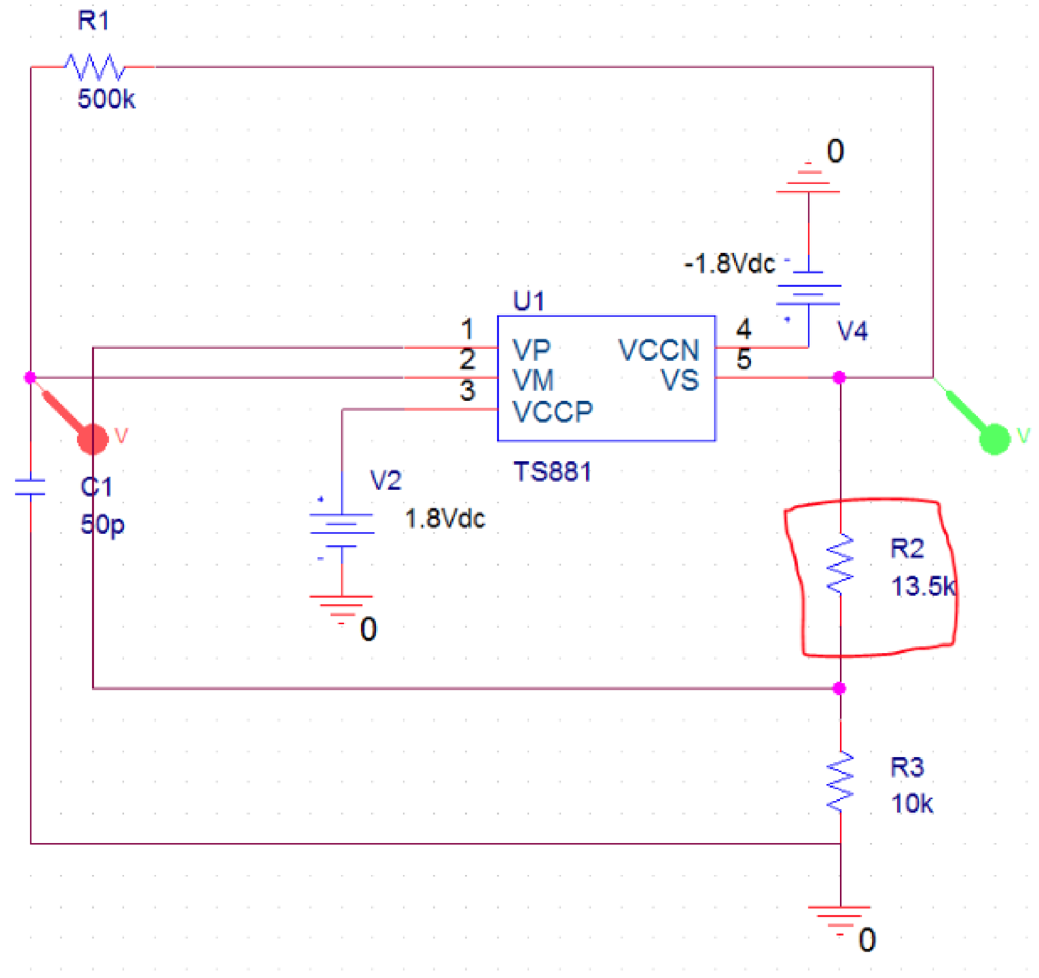
Capacitance Measuring Relaxation Oscillator Version 2 Simulations:

Figure 27: Relaxation Oscillator Version 2 (C = 30 pF)



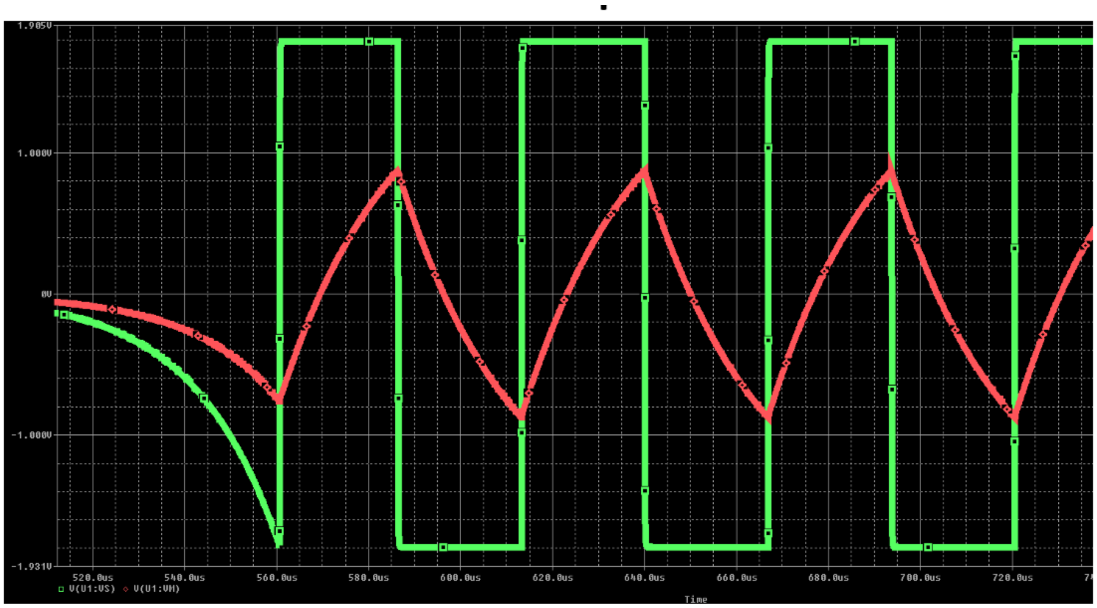
Simulated: T = 40us
Calculated: T = 33us

Figure 28: Relaxation Oscillator Version 2 Schematic Tuned



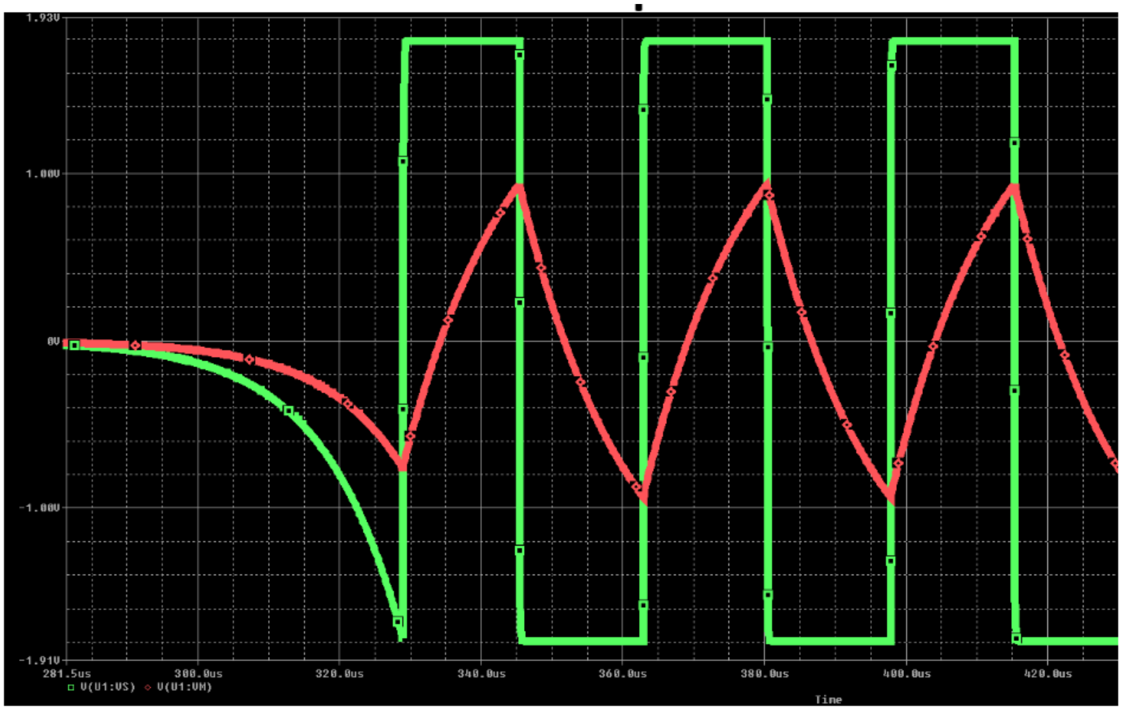
Comments: We tuned the resistor value in order to measure the capacitance accurately.

Figure 29: Relaxation Oscillator Version 2 Tuned (C = 50 pF)



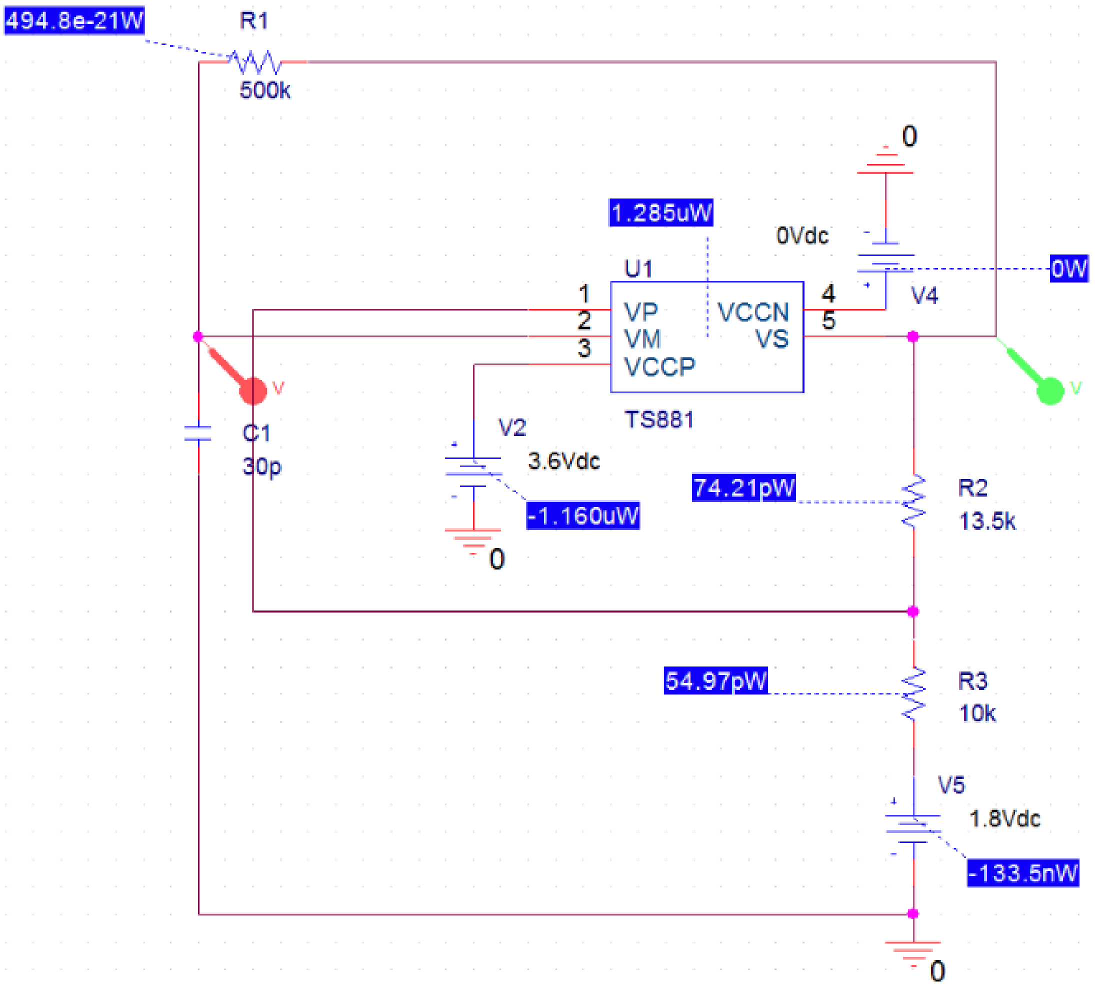
Simulated: T = 53us
Calculated: T = 55us

Figure 30: Relaxation Oscillator Version 2 Tuned (C = 30 pF)



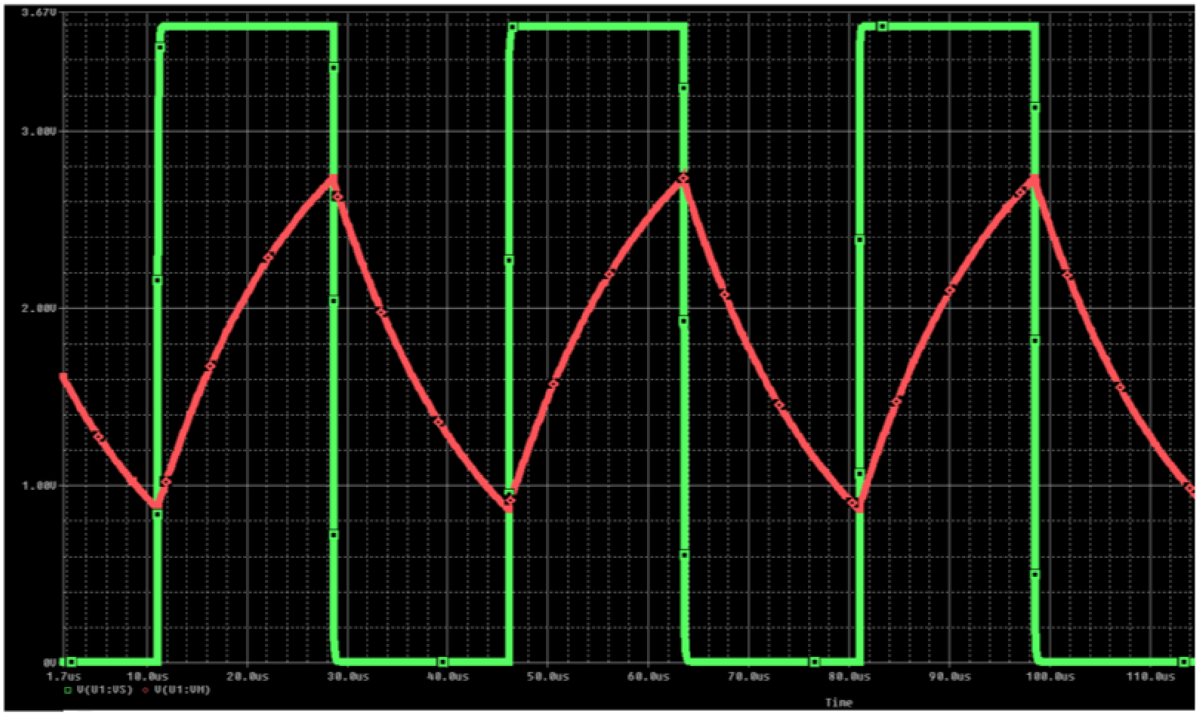
Simulated: T = 35us
Calculated: T = 33us

Figure 31: Relaxation Oscillator Version 2 Schematic Tuned and Shifted



Comments: We shifted the reference voltage on VCCN from -1.8V to 0V. This will allow the output to be in the positive region.

Figure 32: Relaxation Oscillator Version 2 Tuned and Shifted (C = 30 pF)

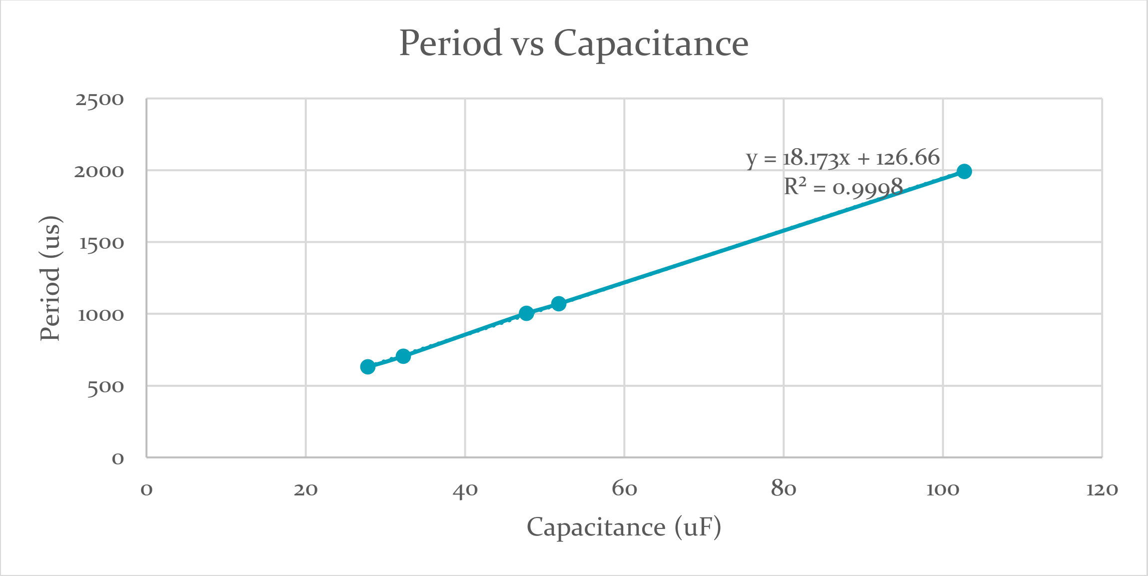


Simulated: T = 35us
Calculated: T = 33us

Relaxation Oscillator Breadboard Tests:

We then built the relaxation oscillator on the breadboard. We measured the Period vs Capacitance and the Power vs Voltage for the breadboard.

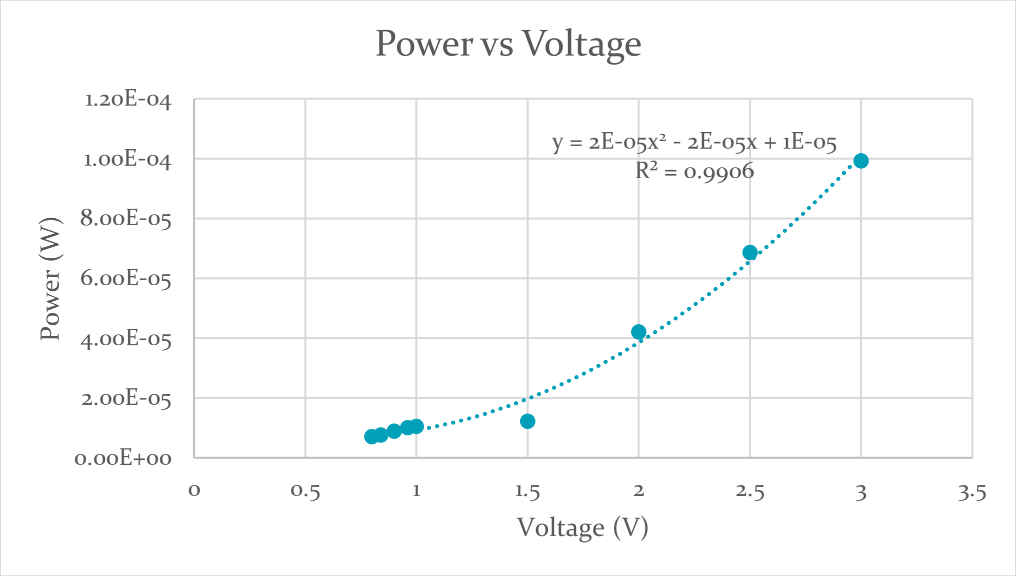
Figure 33: Breadboard Period vs Capacitance Testing Results



This plot shows that the output period is close to linear. It shows that the non-ideal components and the added capacitance from the breadboard slightly affect the performance of the circuit.

We decided to wait until testing of the PCB before using the Period vs Capacitance equation for our microcontroller readings. This will allow the microcontroller to be as precise as possible since the PCB will have less capacitive effects compared to the breadboard circuit.

Figure 34: Breadboard Power vs Voltage Testing Results



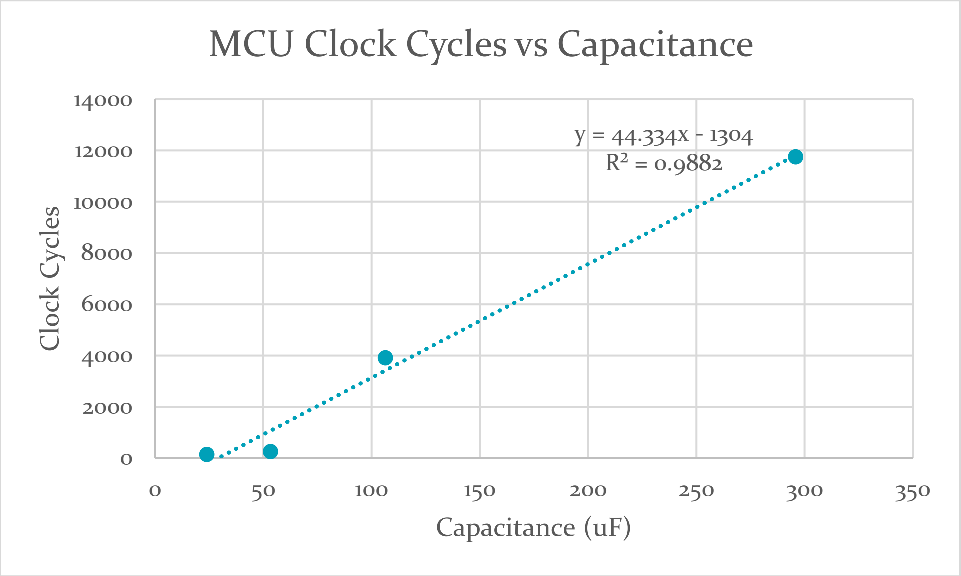
This plot shows that the power consumption is directly proportional to the voltage input to the relaxation oscillator. To maintain low power with decent performance, we need to supply the relaxation oscillator with ~1 volt or above.

We decided to operate at 2 volts since the microcontroller GPIO interrupts won't trigger reliably with a voltage less than 1.8 volts. Using 2 volts give us some room for error.

This plot shows that the measured power output for the breadboard circuit with 2 volts supplied is ~42.1 uW. Our circuit is well below the 115.6 uW at 0 dBm input power expected from the antenna and relaxation oscillator circuitry.

We interfaced the breadboard circuit with the microcontroller and demonstrated its functionality. We plotted the microcontroller's measured clock cycles vs capacitance.

Figure 35: Breadboard Microcontroller Clock Cycles vs Measuring Capacitance Testing Results



This shows that the microcontroller was able to measure the output from the breadboard circuit.

Appendix IV: Code

```
/**

******************************************************************************
  * @file           : main.c
  * @brief          : Main program body

******************************************************************************
  ** This notice applies to any and all portions of this file
  * that are not between comment pairs USER CODE BEGIN and
  * USER CODE END. Other portions of this file, whether
  * inserted by the user or by software development tools
  * are owned by their respective copyright owners.
  *
  * COPYRIGHT(c) 2018 STMicroelectronics
  *
  * Redistribution and use in source and binary forms, with or without
modification,
```

```
  * are permitted provided that the following conditions are met:
  *   1. Redistributions of source code must retain the above copyright
notice,
  *      this list of conditions and the following disclaimer.
  *   2. Redistributions in binary form must reproduce the above copyright
notice,
  *      this list of conditions and the following disclaimer in the
documentation
  *      and/or other materials provided with the distribution.
  *   3. Neither the name of STMicroelectronics nor the names of its
contributors
  *      may be used to endorse or promote products derived from this software
  *      without specific prior written permission.
  *
  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS"
  * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
  * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE
  * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
LIABLE
  * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
  * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
  * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
  * CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY,
  * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
USE
  * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
  *

******************************************************************************
  */
/* Includes ------------------------------------------------------------------
*/
#include "main.h"
#include "stm32l0xx_hal.h"
#include "crc.h"
#include "lptim.h"
#include "gpio.h"

/* USER CODE BEGIN Includes */

/* USER CODE END Includes */

/* Private variables ---------------------------------------------------------
*/

/* USER CODE BEGIN PV */
/* Private variables ---------------------------------------------------------
*/
volatile uint16_t first_cap = 0;

#define THRESHOLD_PERIOD 857
```

```
#define THRESHOLD_FREQUENCY (1000000/THRESHOLD_PERIOD)
/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------
*/
void SystemClock_Config(void);

/* USER CODE BEGIN PFP */
/* Private function prototypes -----------------------------------------------
*/

/* USER CODE END PFP */

/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  *
  * @retval None
  */
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------
--*/

  /* Reset of all peripherals, Initializes the Flash interface and the
Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */
  HAL_PWREx_EnableUltraLowPower();
  HAL_PWREx_EnableLowPowerRunMode();

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_LPTIM1_Init();
  MX_CRC_Init();
  /* USER CODE BEGIN 2 */
   HAL_LPTIM_Counter_Start_IT(&hlptim1, 0xFFFF);
```

```
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
   uint16_t last_cap = 0;
   HAL_PWREx_EnableUltraLowPower();
   HAL_PWREx_EnableLowPowerRunMode();

  while (1){
      first_cap = HAL_LPTIM_ReadCounter(&hlptim1); //read initial timer count
      HAL_Delay(1000); //wait while number of cycles is counted
      last_cap = HAL_LPTIM_ReadCounter(&hlptim1);  //read second count value
      uint16_t cycles = last_cap - first_cap; //calculate cycles during period
      if (last_cap < first_cap){   //handle overflow of counter
            cycles = (last_cap + 0xFFFF) - first_cap;
      }
      if (cycles > (THRESHOLD_FREQUENCY )){  //if the washer is not pressed
            HAL_GPIO_WritePin(GPIOB, GPIO_PIN_3, GPIO_PIN_RESET); //LED off
      } else {
            HAL_GPIO_WritePin(GPIOB, GPIO_PIN_3, GPIO_PIN_SET); //LED on
      }
  }

  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */

  }
  /* USER CODE END 3 */

}

/**
  * @brief System Clock Configuration
  * @retval None
  */
void SystemClock_Config(void)
{

  RCC_OscInitTypeDef RCC_OscInitStruct;
  RCC_ClkInitTypeDef RCC_ClkInitStruct;
  RCC_PeriphCLKInitTypeDef PeriphClkInit;

    /**Configure the main internal regulator output voltage
    */
  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE3);

    /**Initializes the CPU, AHB and APB busses clocks
    */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_MSI;
  RCC_OscInitStruct.MSIState = RCC_MSI_ON;
  RCC_OscInitStruct.MSICalibrationValue = 0;
  RCC_OscInitStruct.MSIClockRange = RCC_MSIRANGE_0;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_NONE;
```

```
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

    /**Initializes the CPU, AHB and APB busses clocks
    */
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                              |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_MSI;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_0) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

  PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_LPTIM1;
  PeriphClkInit.LptimClockSelection = RCC_LPTIM1CLKSOURCE_PCLK;

  if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

    /**Configure the Systick interrupt time
    */
  HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);

    /**Configure the Systick
    */
  HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);

  /* SysTick_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
}

/* USER CODE BEGIN 4 */

/* USER CODE END 4 */

/**
  * @brief  This function is executed in case of error occurrence.
  * @param  file: The file name as string.
  * @param  line: The line in file as a number.
  * @retval None
  */
void _Error_Handler(char *file, int line)
{
  /* USER CODE BEGIN Error_Handler_Debug */
  /* User can add his own implementation to report the HAL error return state
*/
```

```
  while(1)
  {
  }
  /* USER CODE END Error_Handler_Debug */
}

#ifdef  USE_FULL_ASSERT
/**
  * @brief  Reports the name of the source file and the source line number
  *         where the assert_param error has occurred.
  * @param  file: pointer to the source file name
  * @param  line: assert_param error line source number
  * @retval None
  */
void assert_failed(uint8_t* file, uint32_t line)
{
  /* USER CODE BEGIN 6 */
  /* User can add his own implementation to report the file name and line
number,
     tex: printf("Wrong parameters value: file %s on line %d\r\n", file, line)
*/
  /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

/**
  * @}
  */

/**
  * @}
  */

/************************ (C) COPYRIGHT STMicroelectronics *****END OF
FILE****/
```