|  |  |
| --- | --- |
|  | Laundry Now |
|  |  |
|  | Frank Monforte, Ryan Russell  Electrical Engineering Senior Design Project  Advisor Dr. Farid Farahmand, Client Tramaine Austin-Dillon RLC |

# Abstract

Laundry Now is a system that is intended to make the process of doing laundry more efficient. It is designed to be used in apartment complexes or dorm housing laundry rooms and it will allow residents to easily check if washers or dryers are available or in use. This system will also report the number of cycles each machine has completed, allowing property management to see which machines are being used and how much. By keeping track of the number of cycles a machine has been used, it can help facility management determine when repairs or maintenance may be needed.

Table of Contents

[Abstract 1](#_Toc415090112)

[List of Figures 4](#_Toc415090113)

[Lists Of Tables 5](#_Toc415090114)

[Introduction 6](#_Toc415090115)

[Body 6](#_Toc415090116)

[Problem Statement 6](#_Toc415090117)

[Problem Solution 6](#_Toc415090118)

[Literature Review/ Existing Patents 6](#_Toc415090119)

[Marketing Requirements 8](#_Toc415090120)

[Product Abilities 8](#_Toc415090121)

[Design Approach 8](#_Toc415090122)

[Project Overview 8](#_Toc415090123)

[Sensor Nodes 9](#_Toc415090124)

[Hardware Communication 16](#_Toc415090125)

[Wi-Fi Communication 16](#_Toc415090126)

[User Interface 16](#_Toc415090127)

[Power 16](#_Toc415090128)

[Demonstration Unit 17](#_Toc415090129)

[Engineering Requirements 18](#_Toc415090130)

[Economic Analysis 19](#_Toc415090131)

[Social and Environmental Impact 19](#_Toc415090132)

[Timeline and Gantt chart 20](#_Toc415090133)

[Summary 21](#_Toc415090134)

[Reference 23](#_Toc415090135)

[Appendices 25](#_Toc415090136)

[Product Comparison 25](#_Toc415090137)

[Detailed Budget 26](#_Toc415090138)

[Test and Measurements 27](#_Toc415090139)

[Initial Proof of Concept 27](#_Toc415090140)

[SPI Communication Test (Controller Node/ Sensor Node) 28](#_Toc415090141)

[Sensor Setup Code (Controller Node / Sensor Node) 31](#_Toc415090142)

[Serial Communication over Distant (Controller Node/ Sensors Node) 34](#_Toc415090143)

[Parallel Communication over Distant (Sever / Controller Node) 42](#_Toc415090144)

[Best Mounting Location 49](#_Toc415090145)

[WI-FI Distant 52](#_Toc415090146)

[Machine Configuration 52](#_Toc415090147)

[Power Usage 52](#_Toc415090148)

[False Triggers 52](#_Toc415090149)

[Long Term Usage 53](#_Toc415090150)

[Flow Charts 54](#_Toc415090151)

[User Interface 63](#_Toc415090152)

# List of Figures

Diagram 1: System Block Diagram

Diagram 2: Controller Node to Sensor Node Connection

Diagram 3: Pin out from Controller Node to Sensors Node

Diagram 4: Microcontroller Diagram

Diagram 5: Pin Setup Server to Controller Node

Diagram 6: Server Wiring Layout

Diagram 7: Practical System Map

Diagram 8: Generating of Test Information

Diagram 9: Testing Graphed

Diagram 10: Device ID Request and Transmit

Diagram 11: Status Register Request

Diagram 12: Status Register Transmission

Diagram 13: Overview Setup Communication

Diagram 14: Initial Register Location in Setup

Diagram 15: Data being placed in Initial Register

Diagram 16: Benchmark, One Sensor, 0 Ft. Distance

Diagram 17: Test, 1 Sensor, 10 Ft. Distance

Diagram 18: Test, One Sensor, 10 Ft. Distance, Error Occurring

Diagram 19: Test, One Sensor, 10 Ft. Distance, Error Effect on Communication

Diagram 20: Test, Eight Sensor, Buffers, 0 Ft. Distance

Diagram 21: Test, Eight Sensor, 0 Ft. Distance, Delay Added Initial Reading

Diagram 22: Test, Eight Sensor, 0 Ft. Distance, Delay Added Second Reading

Diagram 23: Test, Eight Sensor, 10 Ft. in Between Each Node Total 80 ft Distance

Diagram 24: Benchmark, Four Controllers, 0 Ft. Distant

Diagram 25: Benchmark, Four Controllers, 0 Ft. Distant, Overlapped

Diagram 26: Test 1, Four Controllers, 10 Ft. Distant

Diagram 27: Test 1, Four Controllers, 10 Ft. Distant, Overlapped

Diagram 28: Test 2, Four Controllers, 10 Ft. Distant

Diagram 29: Test 2, Four Controllers, 10 ft Distant, Overlapped

Diagram 30: Test 3, Four Controllers, 10 ft Distant

Diagram 31: Sensor Mounting Locations

Diagram 32: Full Wash Cycle

Diagram 33: Zoomed in Look at the Wash Cycle

Diagram 34: Sever Schematic

Diagram 35: Controller Node Schematic

Diagram 36: Sensor Node Schematic

Diagram 37: Webpage Layout

Diagram 38: QR Code

Flow Chart 1: Initial Server

Flow Chart 2: Initial Controller Node to Sensor Node

Flow Chart 3: Higher Level Sever System Block Diagram

Flow Chart 4: UART Initial Request

Flow Chart 5: UART Data Request

Flow Chart 6: Higher Level Controller Node Block Diagram

Flow Chart 7: Three State Diagram

Flow Chart 8: Three State System State Determination

Flow Chart 9: Four State Diagram

# Lists Of Tables

Similar Product Comparisons Similar

Marketing Requirements

Engineering Requirements

Timeline Spring Semester

Project Milestone

Product Comparison

Device Overview

Detailed Budget

Laundry Now Device Suppliers

# Introduction

The purpose of Laundry Now is to create a system that helps the user to efficiently use their time when doing their laundry. This system will allow users to predetermine if there are washing machines and dryers that are available for use before having to leave their home. This is to help users determine the best time to do their laundry.

# Body

## Problem Statement

In many laundry rooms in college dorms and apartment complexes, the only way to tell if a washer or dryer is in use is by going to the laundry room and physically checking if a there is a machine available. It can be frustrating and time consuming having to constantly check the laundry room when waiting for a machine to become available.

## Problem Solution

Laundry Now will allow users to check whether there is a washer or dryer open before leaving their home. The system will monitor accelerometers that will be attached to the washers and dryers, and when motion is detected the system will update itself to show that the machine is in use. When a user checks the website from their computer or smartphone, they will see the most recently updated statuses of the machines in the laundry room. Laundry Now will be non-invasive and simple to install because it only requires a single sensor node to be attached to each unit. Due to the non-invasive design, Laundry Now will be inexpensive to purchase and implement, and can be used with both newer and older washers and dryers.

## Literature Review/ Existing Patents

Currently on the market there are two different products that are similar to Laundry Now. The first is LaundryView, created by Mac Gray, and the second is Remote Laundry Monitoring System, created by LG. Both of these systems are similar to the product that we are designing, but they each have significant limitations for our targeted market. (1,3)

LaundryView is a system that was developed by Mac Gray as another option that works with the washer and dryers that they supply to colleges and apartment complexes. LaundryView allows users to predetermine if there are any washers or dryers that are available for use. They can do this by checking LaundryView’s website that is continuously updated. This system also allows users to set a notification system that will send a message when the machine has finished the wash cycle or is about to finish the wash cycle. The major limitation of this system is that it requires the washers or dryers that Mac Gray supplies, creating the need to purchase entirely new machines in order to use the system. This significantly increases the financial burden of implementing this system. (1,2) .

The other system that is similar to Laundry Now is LG’s Remote Laundry Monitoring Service, which is a device that allows homeowners to view the amount of time left on the current wash cycle. This device requires the installation of modem into a washer and dryer which will connect with the timer, and then will relay the information over the household power lines to the receiver which needs to be plugged into the wall. This system works well for a single household, but it is not scalable for use in a facility with many machines. (3)

Similar Product Comparisons

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Device | Need extra Equipment | Scalable | Internet Capable | Estimated Cost |
| LaundryView | Mac Gray supplied machines | Yes | Yes | Depends on the number of machines needed |
| LG’s Monitoring System | No | No | No | $90.00 |
| Laundry Now | No | Yes | Yes | $525 (Full System Parts); $160 (Demo Unit Parts) |

(1,2,3,4)

## Marketing Requirements

|  |  |
| --- | --- |
| Function | Allows users to predetermine if there are any washers/dryers that are available for use, by checking the website. |
| Number of Machines | 8 sensors per controller, 4 controllers per server, up to 32 different machines |
| Wi-Fi Connection | WPS connection |
| Powered | wall socket |
| Consumer Cost | $525 (Full System Parts); $160 (Demo Unit Parts) |
| Size | Server 4” by 4”, controller 4” by 4”, sensor 2” by 2” |
| Package | Plastic shells |
| PC and Cell Phone capable | Any device that is internet enabled |
| Sensor connection | Non-invasive, attached with double stick tape users might be reluctant as this will damage the finish on the machine if left for a long time. Would suction work? |
| Update Time | The webpage will be updated every 5 minutes to reflect the latest washer/dryer status. |

## Product Abilities

* Keeps track of the number of machine cycles for days, weeks, and months.
* It will be able to tell if the machine is in operation or is not in operation.
* The website will be able to display the time remaining on every machine.
* Onsite internet connection.
* Website updates every 5 minutes.
* Non-invasive and simple to install.

## Design Approach

### Project Overview

This product will allow users to see the number of washers and dryers that are in use or available at their laundry facility without having to leave their home. It will do so through sensor nodes that are placed on every washer and dryer and are connected to the SPI interface of the controller node microcontroller. The microcontroller will keep track of which machines are in use, and monitor for vibrations from the door opening and closing to determine when the previous user removes their clothing, leaving the machine available for use. The status of the machines will be stored on the individual controller nodes and the sensors will be checked every 2 minutes. The server will request the latest machine status and update the website every 5 minutes. When a request is received from a user over Wi-Fi, the server will return the HTML website that displays the latest machine status. See Diagram 1 for the basic system design.

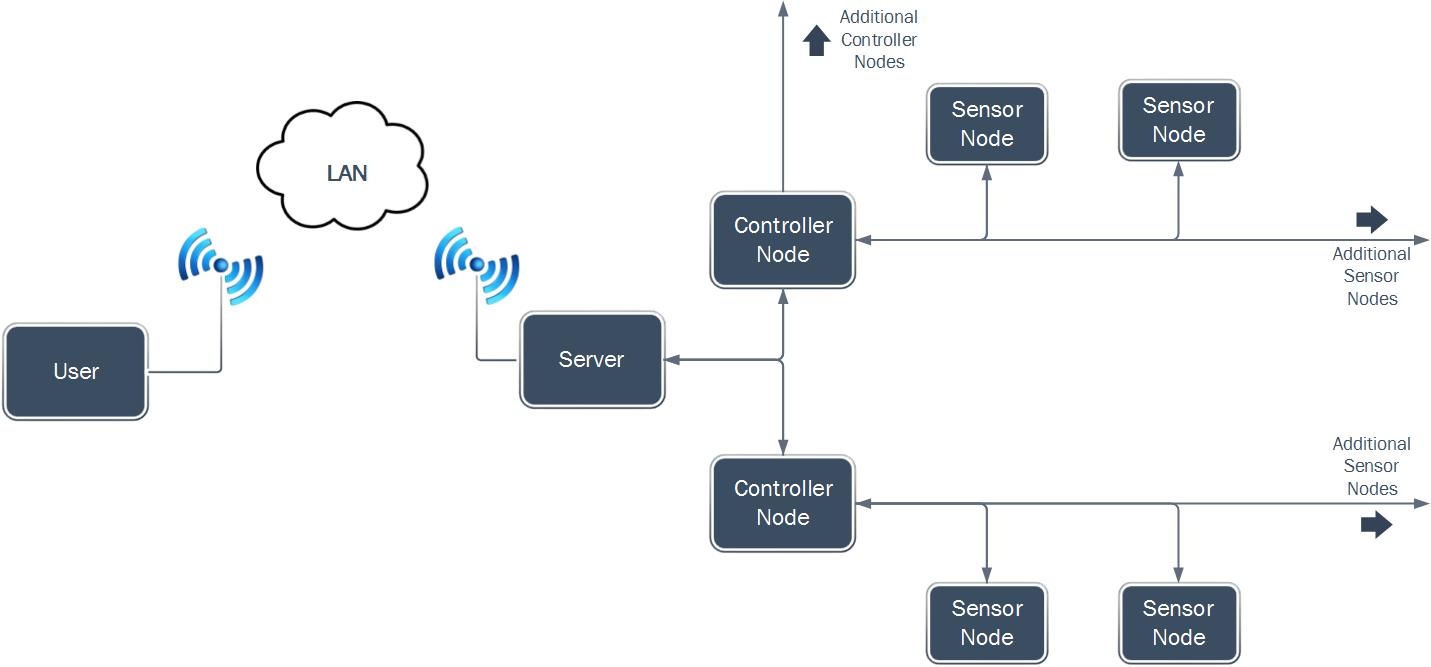


Diagram 1: System Block Diagram

### Sensor Nodes

In order for Laundry Now to detect if a machine is running or has stopped and the door opening and closing, it will utilize accelerometers that will be placed directly on the machines. This will also for the system to be non-invasive, inexpensive, and simple to install, but will still allow it to collect enough necessary information. The sensors will be 3-axis MEMS digital accelerometers with built-in SPI interfaces. These sensors are simple, easy to communicate with, and will allow the system to record enough movement data to track the status of the machine it is monitoring. (6,7)

The Clock, MOSI (Master Out, Slave In) and MISO (Master In, Slave Out) lines of the SPI interface will be connected to the sensors in a standard way, however the CS (Chip Select) line will be controlled through a set of multiplexed sensor select control lines to reduce the number of wires being used. There will be 3 control lines total on each controller, allowing up to 8 sensor nodes to be connected. Diagram 3 shows the pin setup from the controller to the sensors. Diagram 2 shows the high level connection between the sensor node and the controller node.

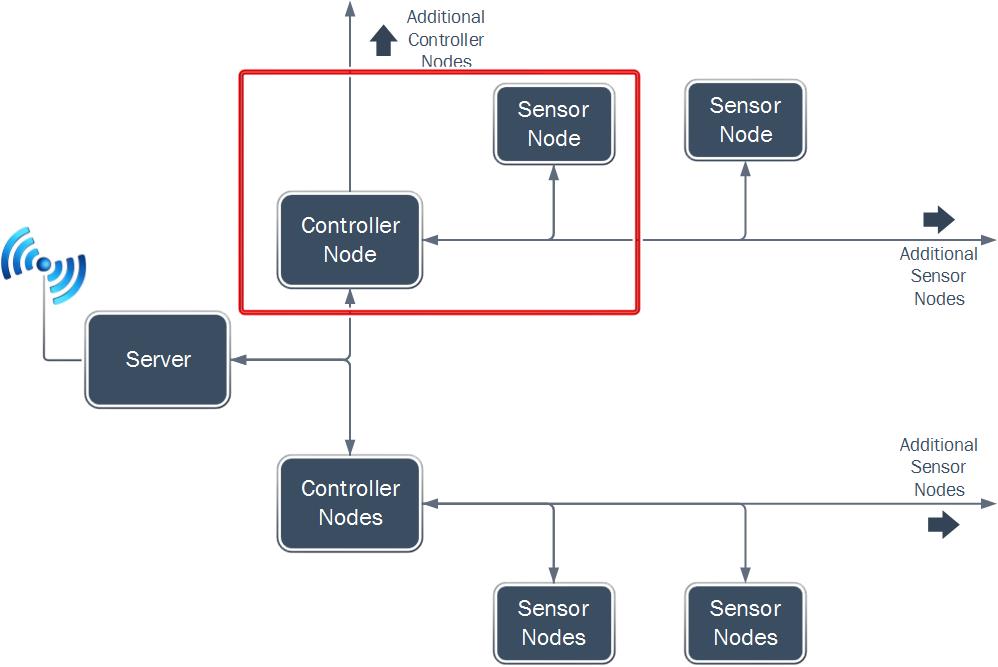


Diagram 2: Controller Node to Sensor Node Connection

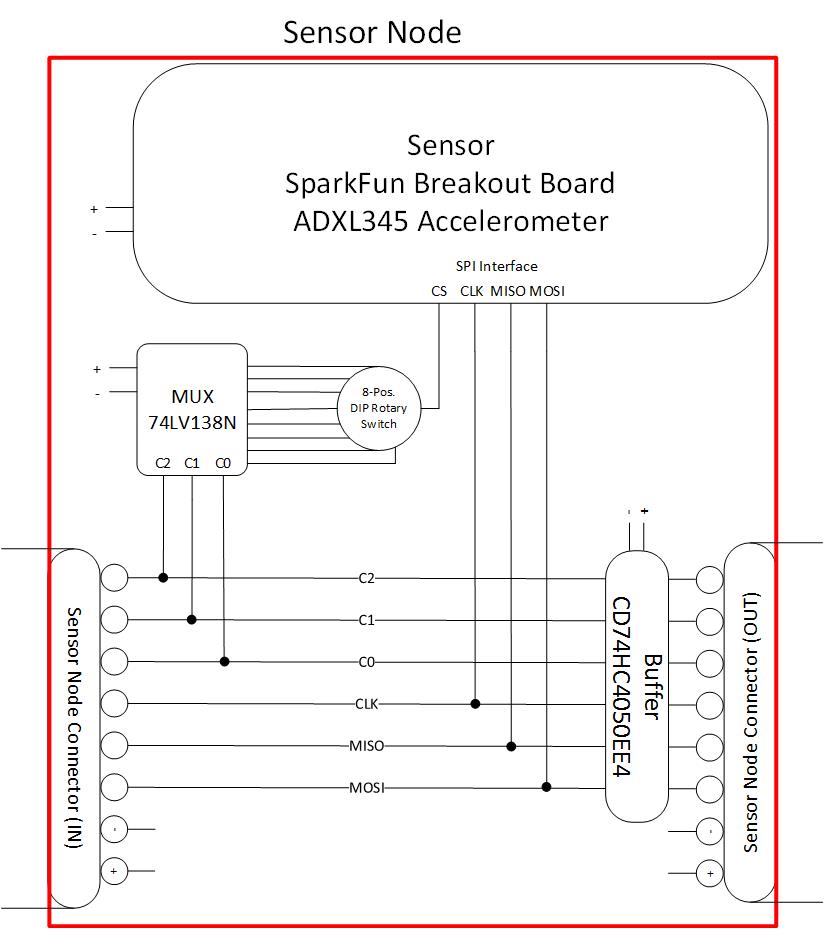


Diagram 3: Pin out from Controller Node to Sensors Node

Microcontrollers

Laundry Now will utilize two types of microcontroller units. The main microcontroller will act as the server, and will store the latest machine status as well as handle data requests from users through a Wi-Fi controller. This microcontroller will be a ChipKIT WF32 produced by Digilent. It features a Microchip PIC32MX69F512L MCU and a Microchip MRF24WG0MA Wi-Fi Controller built on the same chip. It will be programmed with Microchip MPLABX software. Separate microcontrollers will act as control units and will directly interface with the sensor nodes. These will be PIC18F45K20 microcontrollers and will also be programmed using Microchip MPLABX software. The server will be able to handle up to 4 controller nodes at one time, and each controller node will be able to communicate with up to 8 sensor nodes. The high level diagram containing the microcontrollers is shown in Diagram 4 and the pin setup is shown in Diagram 5 and Diagram 6. Diagram 7 shows the practical system application. (8,9,10,11)

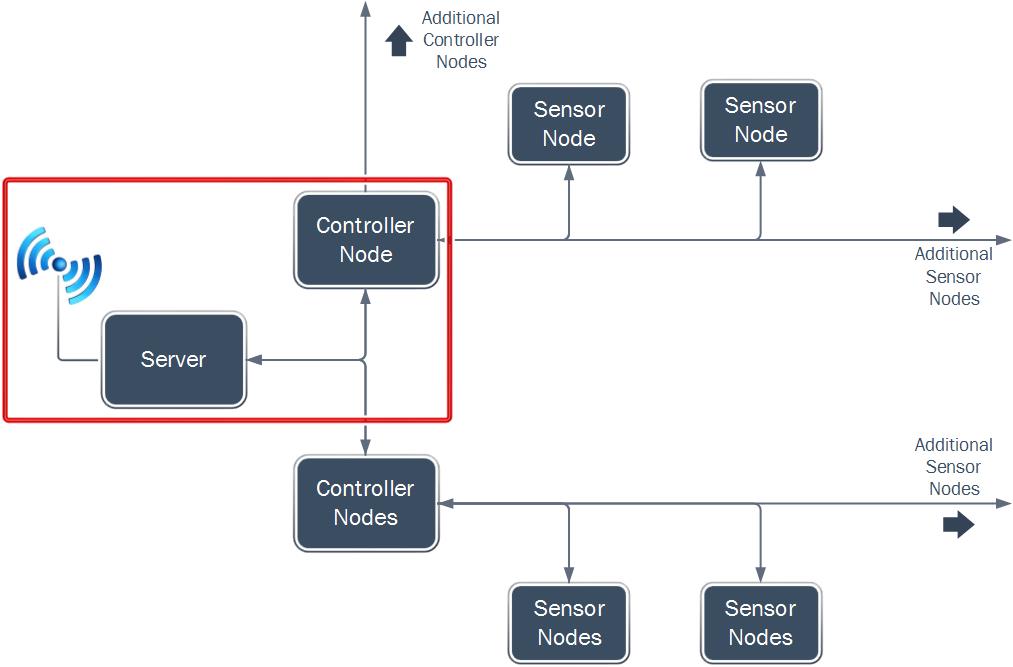


Diagram 4: Microcontroller Diagram

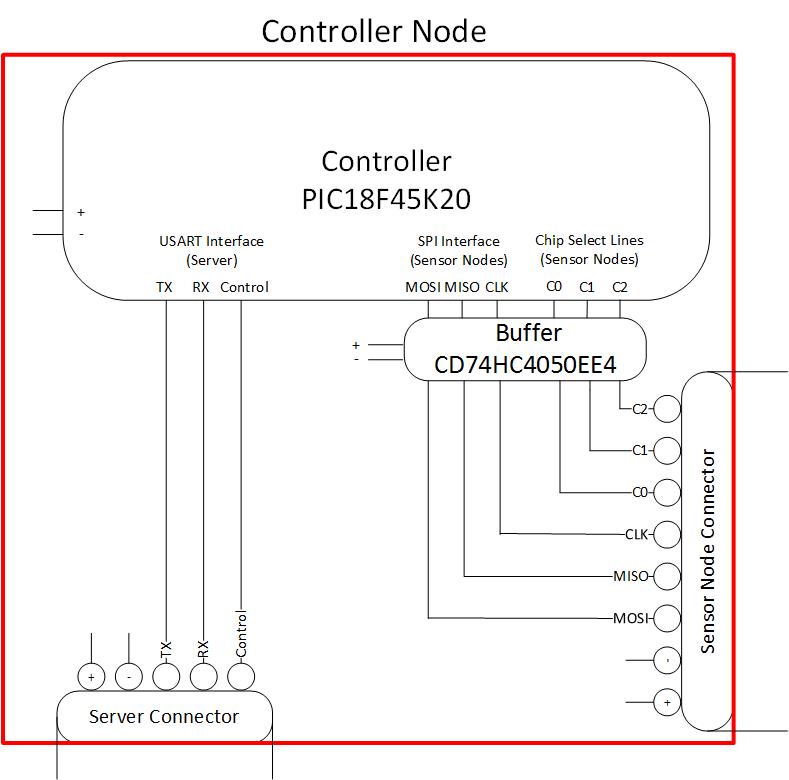


Diagram 5: Pin Setup Server to Controller Node

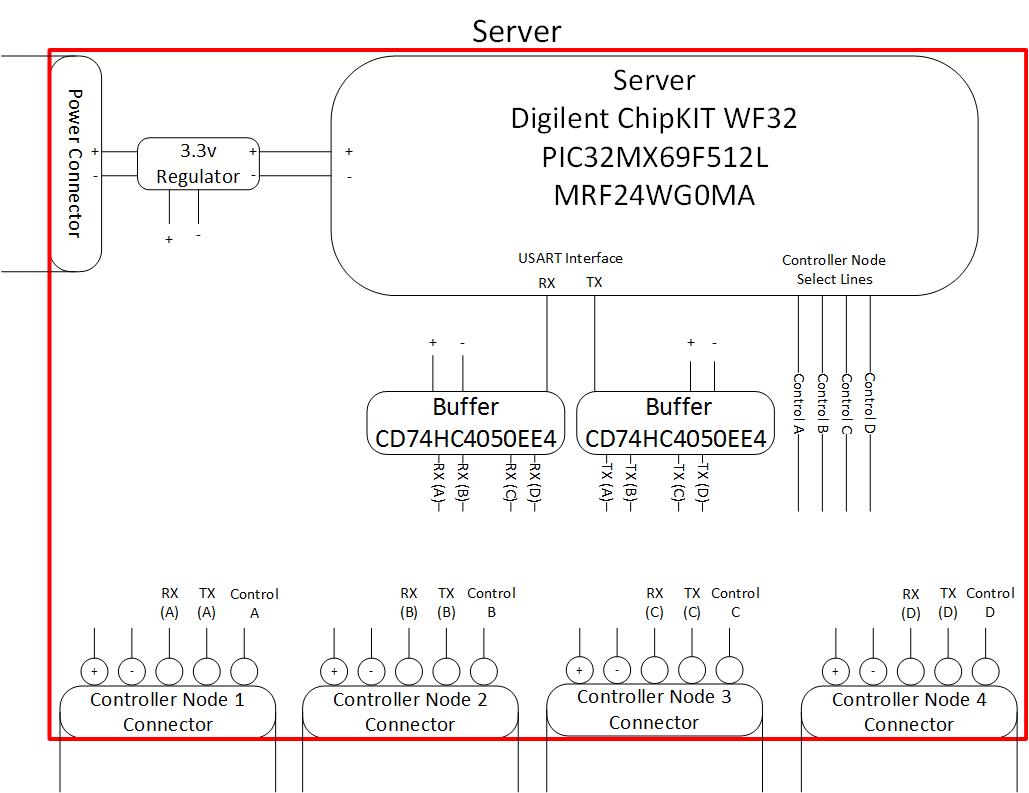


Diagram 6: Server Wiring Layout

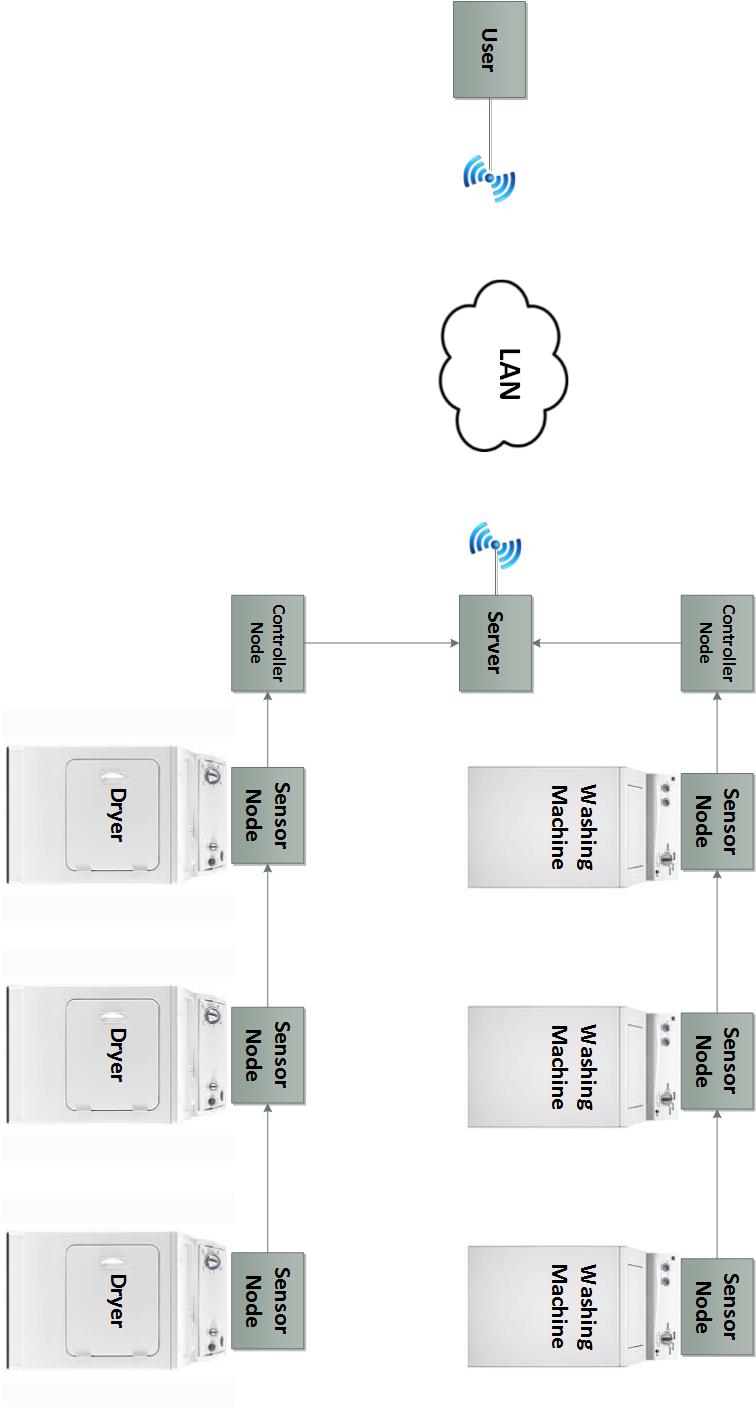


Diagram 7: Practical System Map

### Hardware Communication

The microcontroller that interfaces with the sensor nodes directly will do so using the SPI communication interface available on the PIC18F45K20 and the sensor breakout boards. The microcontroller will cycle through each accelerometer on the SPI bus using three multiplexed control lines over the SPI interface. Each cycle of data will be collected, and then stored on the controller node. The machines will be continuously monitored for changes in motion to determine if a machine is in use. It will check for a spike in movement after a wash or dry cycle is completed to determine if the door has been opened and the machine is available. The machine status is retrieved by the server over a serial UART interface and the website stored on the server is updated accordingly. (8,9)

### Wi-Fi Communication

The Wi-Fi communication will be handled entirely by the main server microcontroller. When a user attempts to access the site from their computer or smartphone, the data stored on the server microcontroller will be sent to the Wi-Fi controller over the SPI communication interface. This is transfer is handled within the Digilent WF32 board. Laundry Now will support push button WPS (Wi-Fi Protected Setup) connections, allowing the user to press the WPS button on the server and the WPS button on the router to setup the Wi-Fi connection for simple network setup. (10)

### User Interface

Users will be able to access the information generated by Laundry Now through a website that is accessible from either a computer, tablet, or smartphone. The website will show which washers and dryers are currently in use. When the user enters the IP of the server microcontroller, the microcontroller will return an HTML webpage that displays the machine status.

### Power

Laundry Now will be powered by a wall outlet. This will allow the system to continuously run without the need to change a battery. The main power source will be connected to the server unit which will then provide power the controller nodes along with the sensor nodes. The server node, controller node and the server all require 3.3 volts which allows for the ease of data communication between the components.

### Demonstration Unit

As proof of concept, we will be building a demonstration unit that utilizes the different parts of this design. The demonstration unit will have one server microcontroller controlling the system. There will be two control nodes (the system will be capable of a maximum of four controller units) with two sensor nodes connected to each controller node unit (capable of a maximum of eight sensors per controller). The demo unit will require a single available wall outlet power source to power the server and connected components. There will also need to be an available Wi-Fi network provided by a router that is push button WPS capable. A video will be created showing the demonstration unit being used.

Laundry Now will be able to detect if the washer or dryer is in operation or is off and will be able to detect the door opening and closing. It will not be able to directly detect if the load has actually been removed from the machine. By analyzing the movement data it will make the best determination of whether the machine is in use, has completed a cycle, or is empty. This information will then be sent to the server which will post if the machine is in use or waiting to be empty (before the door is opened and closed) or currently available (after the door has been opened and closed).

### Engineering Requirements

|  |  |
| --- | --- |
| Controller Node to Sensor Node communication | Data flow is through SPI, Sensor node selection is through a demultiplexer |
| Controller to Server communication | USART |
| Wi-Fi requirements | Needs WPS based router |
| HTML Page | Stored and updated on server |
| Sensor Node dies | If the sensor node does not respond then then the website will be updated to show an error that sensor could not be updated. |
| Storage capability | Onboard |
| Out of service indication | If a controller node does not communicate with the server then the website will be updated to show an error that a controller node could not be updated. |
| Update Time | The server will update the HTML file every 5 minutes.  The controller nodes will check the sensor nodes every 2 minutes. |

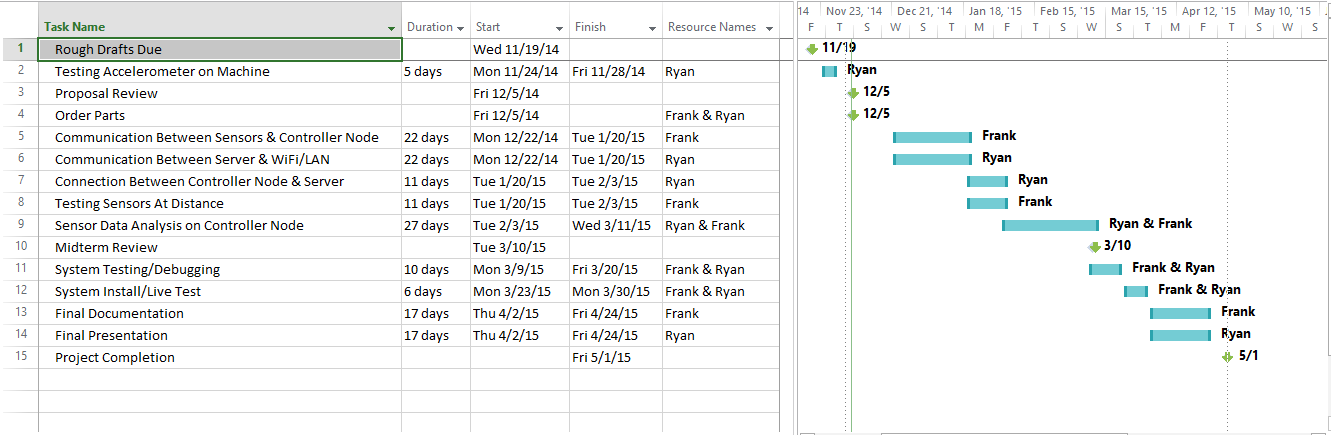
## Economic Analysis

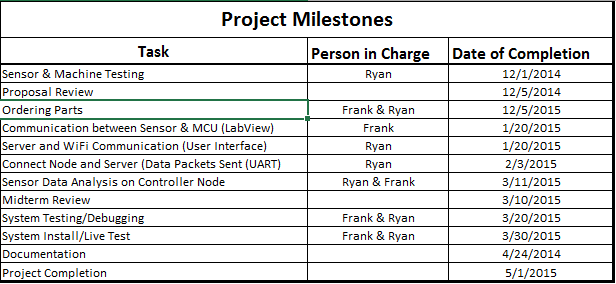
This product is designed to compete with LaundryView, which is a similar product. LaundryView is a system that was create by Mac Gray to work with the washers and dryers that they supply. Mac Gray is a laundry facility service provider which works with some schools and apartment complexes to maintain their laundry faculties. They provide the machines and maintenance on the machines. Their LaundryView system is only designed to work with the machines that they install. They currently do not quote a price on the cost per machine and the price of the LaundryView system, but the machines they install are either Maytag or Speed Queen washers and dryers. The average price of a new Maytag or Speed Queen washer is about $1500.00 per unit, and the average for a Maytag or Speed Queen dryer is about $1500.00 per unit. We wanted Laundry Now to be less expensive than LaundryView. Because of this, our product is designed to work with preexisting washers and dryers in a laundry facility so there is no need to purchase new washers and dryers. The parts cost for a fully expanded Laundry Now system is about $525 at the current parts cost. Using the chips directly (not on breakout boards) would make the cost of Laundry Now be lower. (1,2,4)

## Social and Environmental Impact

Laundry Now is designed to compete with LaundryView, which is a system that have already been installed and is used on college campuses and apartment complexes. Because of this it will have a similar social and environmental impact to LaundryView which has been greatly successful at reducing the time required by users to do their laundry, and increasing the efficiency of the laundry facilities. (2)

## Timeline and Gantt chart





# Summary

Laundry Now will allow users to predetermine if there are washers or dryers that are not in use, or to determine that all of these machines are being used. This product is designed to be placed on any washer and dryer units, including older washers and dryers that are not Wi-Fi enabled. There will be an accelerometer that will be placed on every machine that will then allow the controller node monitor the washer/dryer movement. The controller node will collect this movement data from the sensor every 2 minutes. It will analyze the data and update the status of the washer/dryer to one of three states: In Use, Cycle Completed, or Available. It will store this information until the server requests it. Every 5 minutes the server will request the latest machine status from the controller nodes and update the HTML webpage file. When a person checks the website the server will send the latest HTML file showing the status of the machines. It is important to note that the system will not be able to directly tell if a machine has been emptied. It will monitor the movement of the washer or dryer and make a best determination if it is running, has completed a cycle, or is empty. An example of the webpage is below.



# Reference

1. <http://www.macgray.com/>

Mac Gray is the inventor and supplier of LaundryView and all of the products needed to run LaundryView. 11/15/2014

1. <http://www.campus-solutions.net/index.html>

Campus-solutions is one of the many features that is offered by Mac Gray. This website shows how LaundryView has been implemented in Colleges across the country and how it has improved the efficiency of college laundry facilities. 11/15/2014

1. <http://www.amazon.com/LG-Electronics-RLM20K-Laundry-Monitoring/dp/B0012OMY2K>

Is the main selling source for LG’s Remote Laundry Monitoring System, provides cost along with description of what the product does and some reviews form customers. 11/15/2014

1. <http://www.maytag.com/Laundry-1/Laundry_Laundry_Appliances_Washers-3/102120050+4294966015/>

Provides the estimated cost of the Maytag washers and dryers that Mac Gray offers as part of their laundry facility service, and the machines needed to allow for the use of LaundryView. 11/15/2014

1. <http://www.nmhc.org/Content.aspx?id=4708>

This link provides the number of residents the live in apartment complexes in the state of California and the whole country. 12/4/2014

1. <http://www.analog.com/static/imported-files/data_sheets/ADXL345.pdf>

Analog Devices ADXL345 Data Sheet. 12/1/2014

1. <https://www.sparkfun.com/products/9836>

SparkFun Product Page & Details for ADXL345 Breakout Board. 12/1/2014

1. <http://ww1.microchip.com/downloads/en/DeviceDoc/41303G.pdf>

Microchip PIC18F45K20 Data Sheet. 12/1/2014

1. <http://ww1.microchip.com/downloads/en/DeviceDoc/61156H.pdf>

Microchip PIC32MX69F512L Data Sheet. 12/1/2014

1. <http://ww1.microchip.com/downloads/en/DeviceDoc/70686B.pdf>

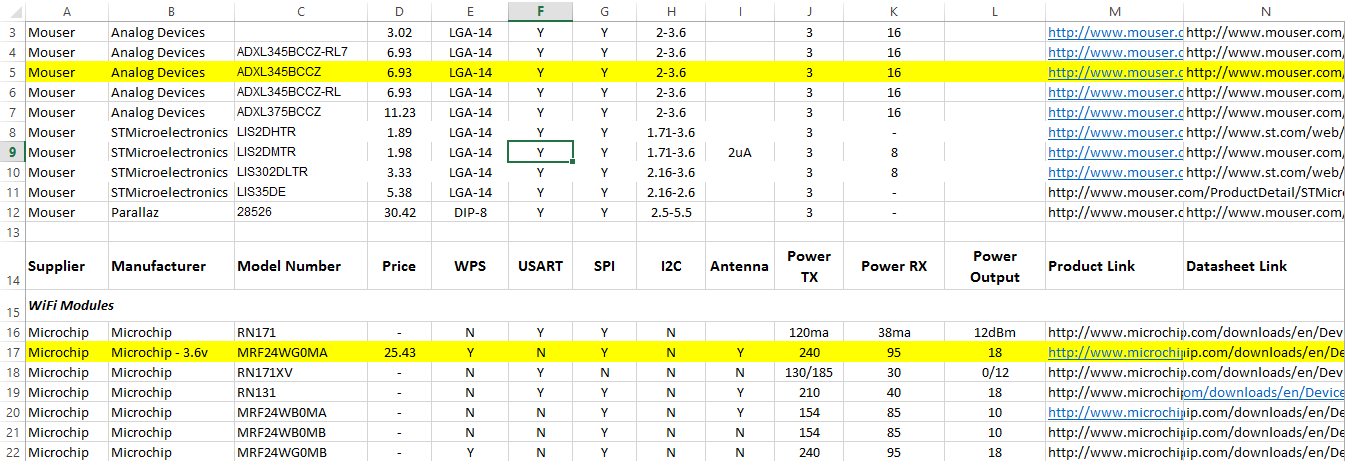
Microchip MRF24WG0MA Wi-Fi Controller. 12/1/2014

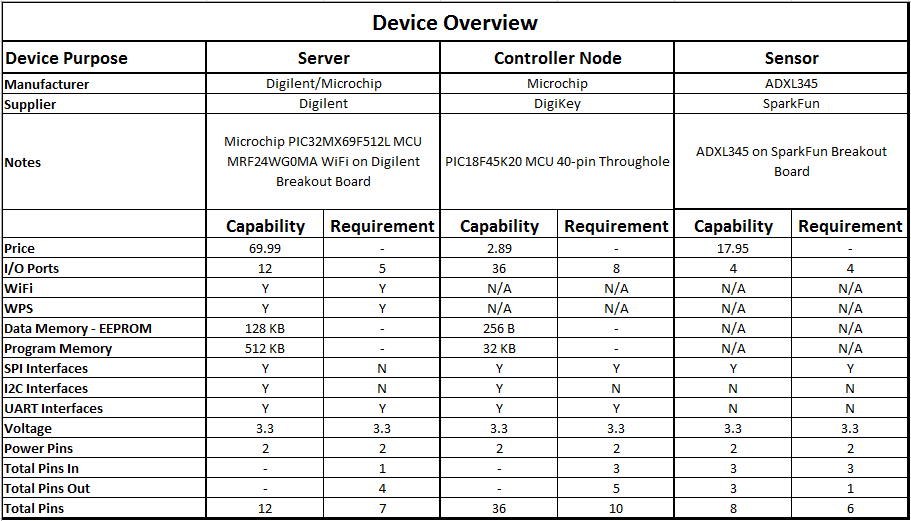
1. <https://www.digilentinc.com/Products/Detail.cfm?NavPath=2,892,1193&Prod=CHIPKIT-WF32>

Digilent ChipKit WF32 Product Page. 12/1/2014

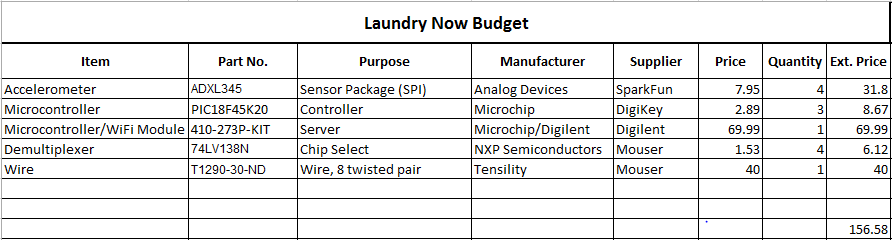
# Appendices

## Product Comparison

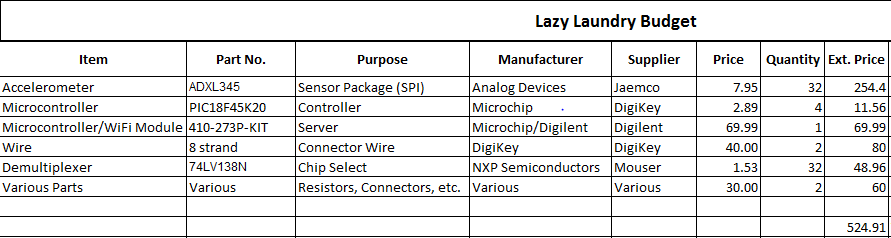


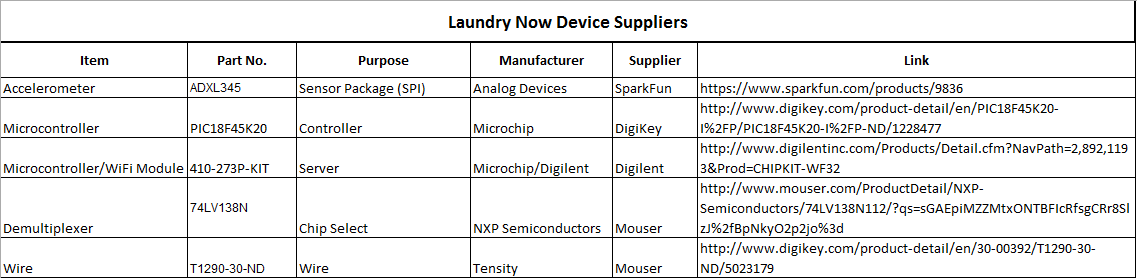


## Detailed Budget



This budget reflects the cost of parts using breakout boards and pre-assembled parts to make the demo unit (1 server, 2 controller nodes, 4 sensor nodes). If Laundry Now was to be implemented in a full scale commercial product breakout boards and pre-assembled boards would not be needed and the cost for the chips would drop, allowing the overall cost for the system to be cheaper. The budget below shows the cost using current parts (breakout boards/pre-assembled boards) for a fully expanded system (1 server, 4 controller nodes, 32 sensor nodes).



****

## Test and Measurements

### Initial Proof of Concept

In order to be sure the microcontrollers could properly collect valid movement data from the machines we tested a similar accelerometer. We used an Arduino Uno microcontroller with a simple script to monitor the machine, collect the data, and transmit it back to a laptop where the data points could be analyzed. Using MATLAB, we plotted the data points across the clock cycles to see if the vibrations created by the machine could be easily analyzed. We found that we could easily see the movement of the machine when it was running and the quick spikes in movement when the door was opened and closed. This data will allow us to do some simple processing on the controller node MCU to determine if a machine is in use or available. The sensor that was used was a Parallax 2-axis MEMS digital accelerometer that provided an output similar to what the Analog Devices 3-axis MEMS accelerometer will output. Diagram 8 is a screenshot take during the data collection process and Diagram 9 is the graph representation of the collected data.

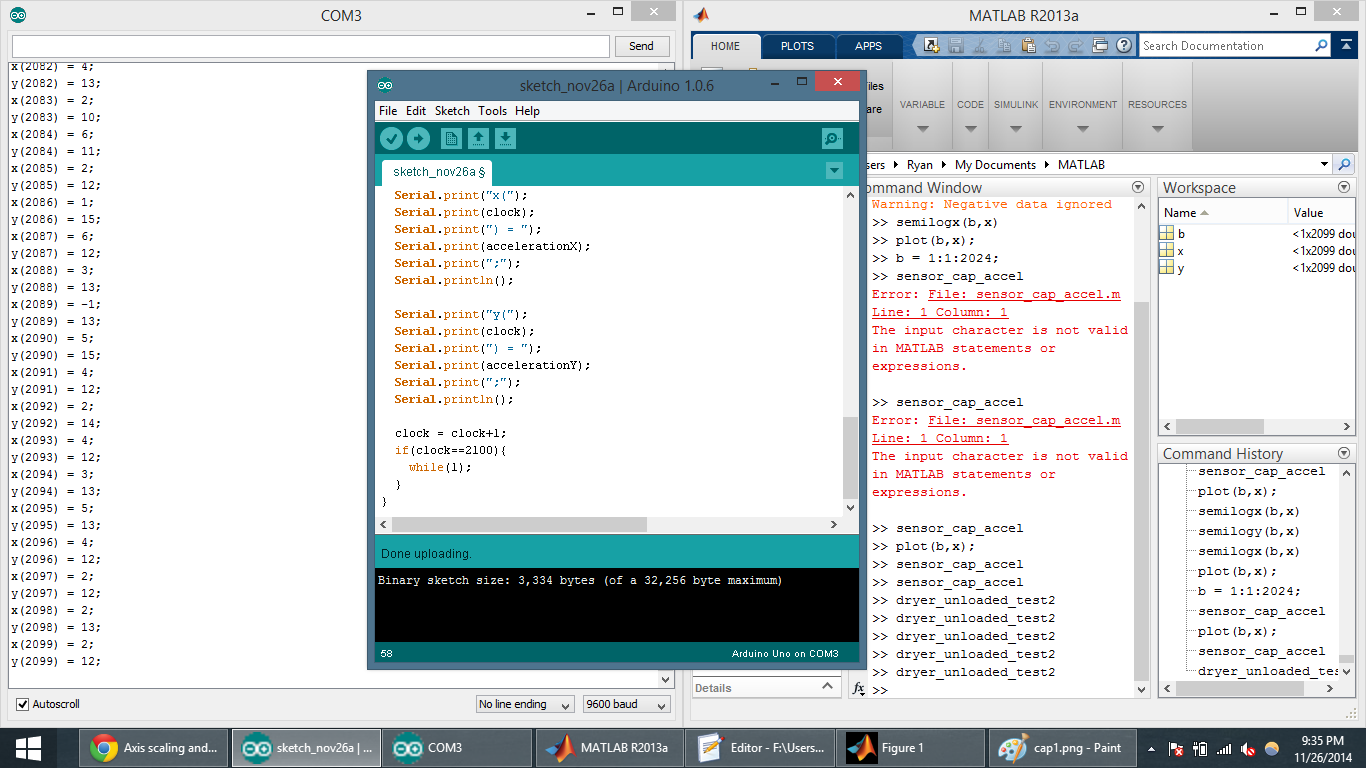


Diagram 8: Generating of Test Information

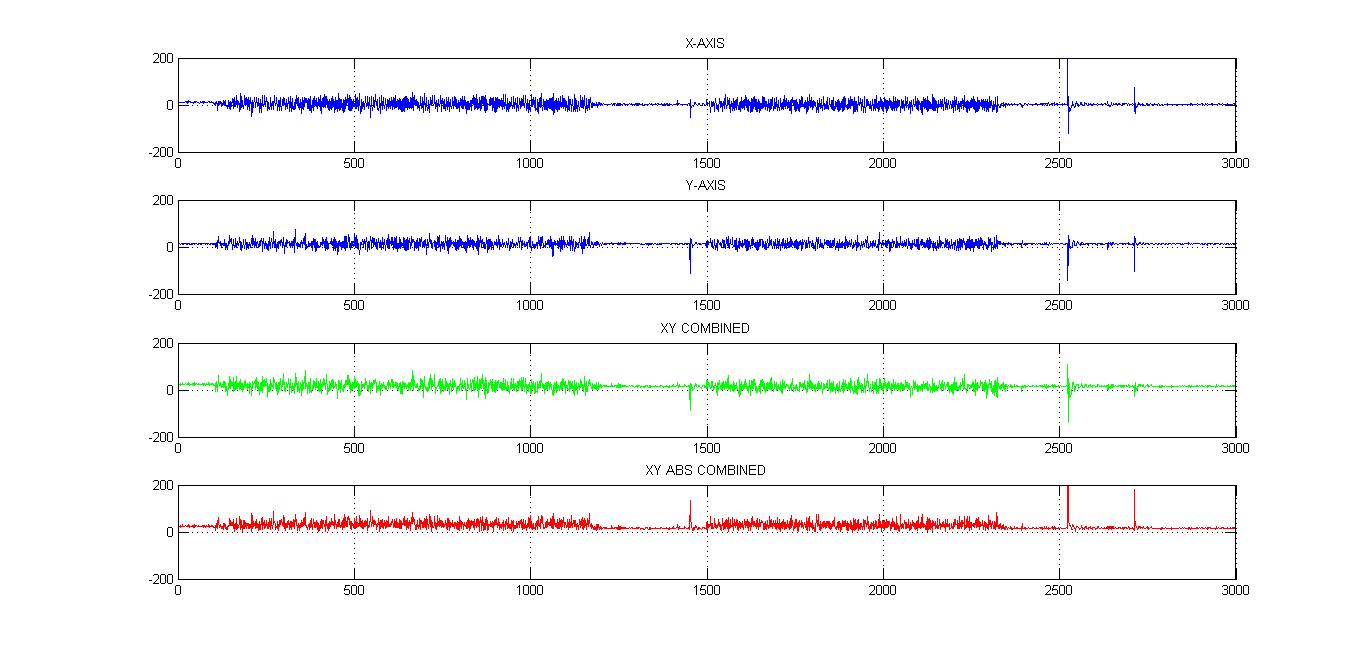


Diagram 9: Testing Graphed

The first two graphs are the raw data output of the X and Y axis of the accelerometer. The third graph is one method of analysis to emphasize the spikes by summing the axes. We found that while this helped amplify certain parts of the data, it cancelled out in other cases. In order to get around this, we added the absolute value of the two axes. This produces a much better output that can easily be analyzed by the controller node.

### SPI Communication Test (Controller Node/ Sensor Node)

How to send and receive communication between controller node and the sensor node via SPI. First I will test this by sending a device ID request and the response should be 0xE5. The second test will request the Status register which will indicate if there is any movement detected on the sensor. If there is it will send an indicating flag.

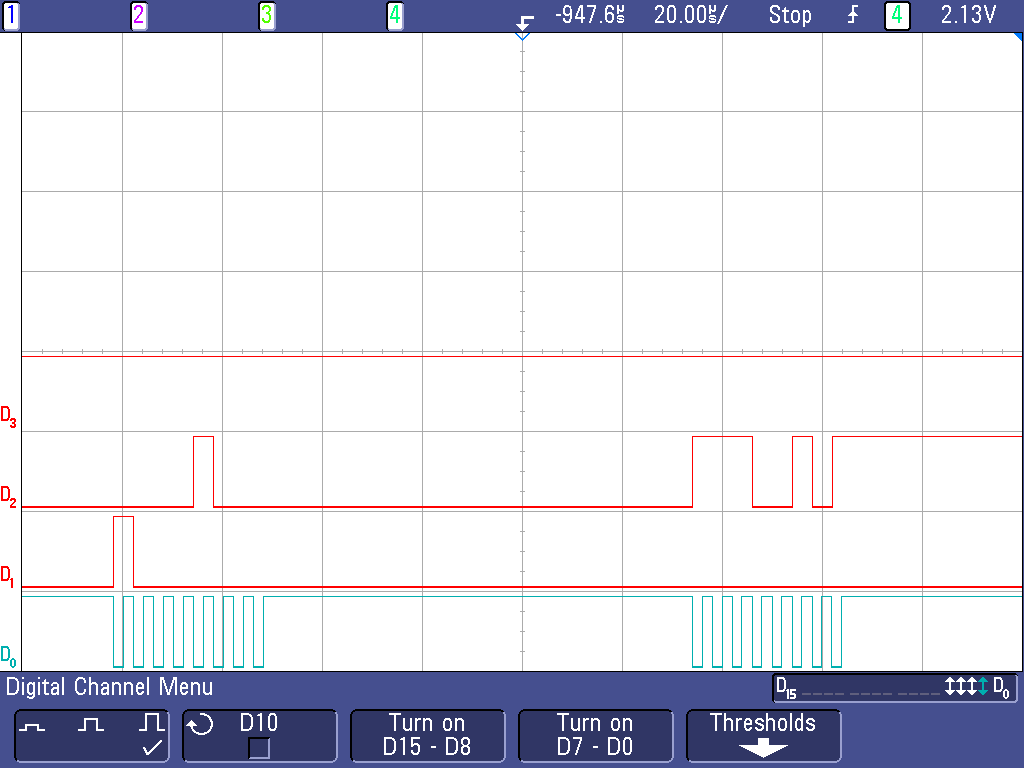


Diagram 10: Device ID Request and Transmit

This shows communication back and forth with the sensor and with the controller. D0 shows the SPI clock that is sent by the controller to the sensor, D1 shows the controller output and the sensor input, D2 shows the sensor output and the controller input. D1 shows the request register to the sensor for register 0x00 with the most significant bit indicating it is a read request. Then on D2 shows the sensor response the first series of information is not read by the controller but the second set of information is the response to the request which is 0xE5 or 11100101.



Diagram 11: Status Register Request

This scope image shows the request code being sent to the sensor asking for the information stored in the status register. D0 and 1 show the SPI clock, D1 and 2 show the output request from the controller and input of the sensor, D3 and 3 show the output from the sensor and the input of the controller. The first spike on the sensor output is the data ready response and the last two spikes are set when the data ready bit is set.

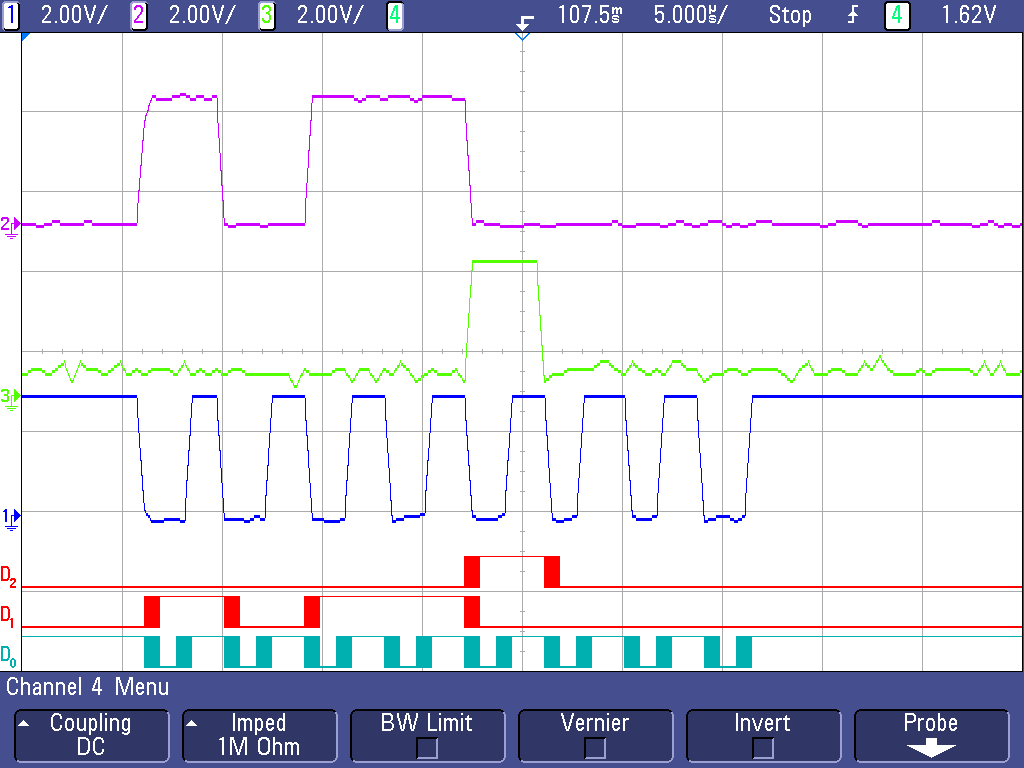


Diagram 12: Status Register Transmission

This shows the response sent from the sensor to the controller. D0 and 1 show the SPI clock, D1 and 2 show the output request from the controller and input of the sensor, D3 and 3 show the output from the sensor and the input of the controller. The spike shows that the activity flag has been set.

### Sensor Setup Code (Controller Node / Sensor Node)

For the sensors to work they first have to be set up in order to do this several different register have to be set up. The registers that need to be set up is the thresh tap, duration, latency, window, thresh act, fifo control, power control, act inact, tap axes, data format, interrupt map registers.

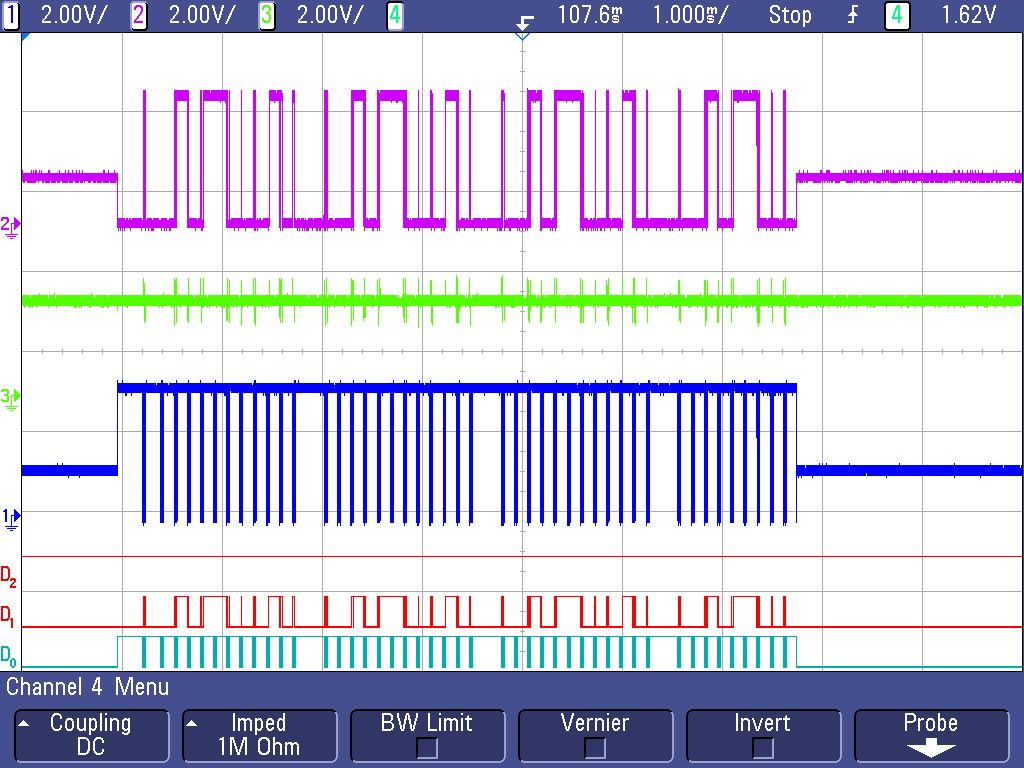


Diagram 13: Overview Setup Communication

The image above shows all of the initialization of the sensors being sent from the controller. D0 and 1 show the SPI clock, D1 and 2 show the output request from the controller and input of the sensor, D3 and 3 show the output from the sensor and the input of the controller. On D0 and 1 shows the SPI clock being sent from the controller each spike is a total of 8 bits being sent.

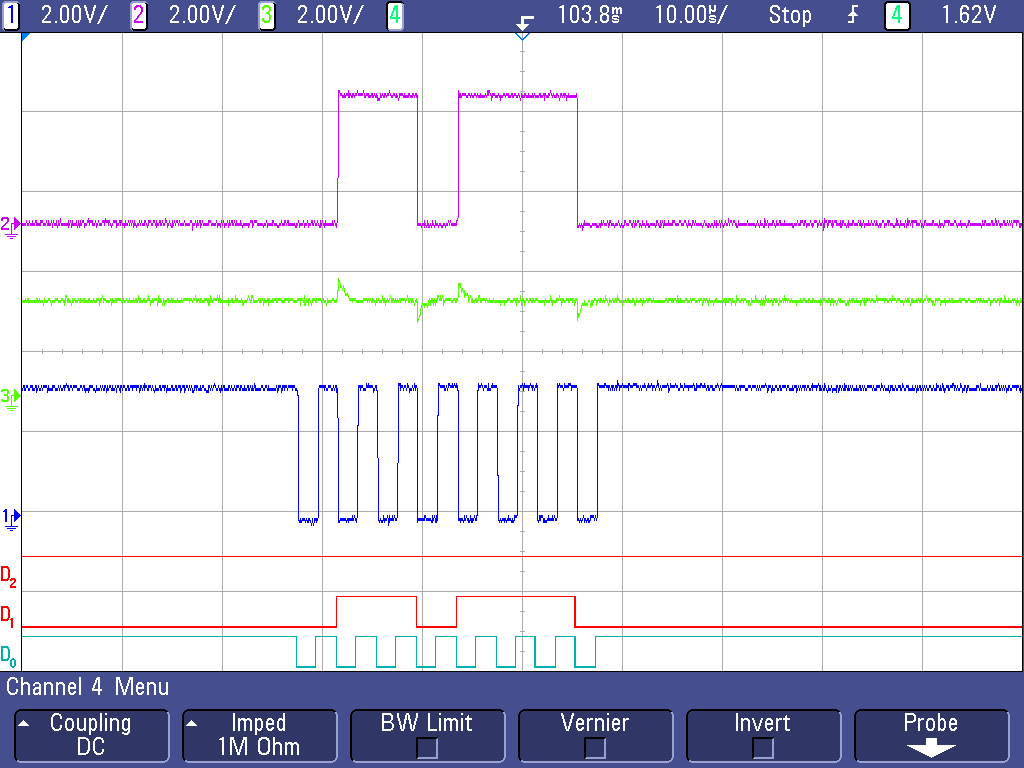


Diagram 14: Initial Register Location in Setup

This shows the register location on the senor (01101110), the most significant bit is for multiple enters being entered, the send most significant bit is indicating a write request, then the rest of the bits is the register location for int\_enabled register. D0 and 1 show the SPI clock, D1 and 2 show the output request from the controller and input of the sensor, D3 and 3 show the output from the sensor and the input of the controller.

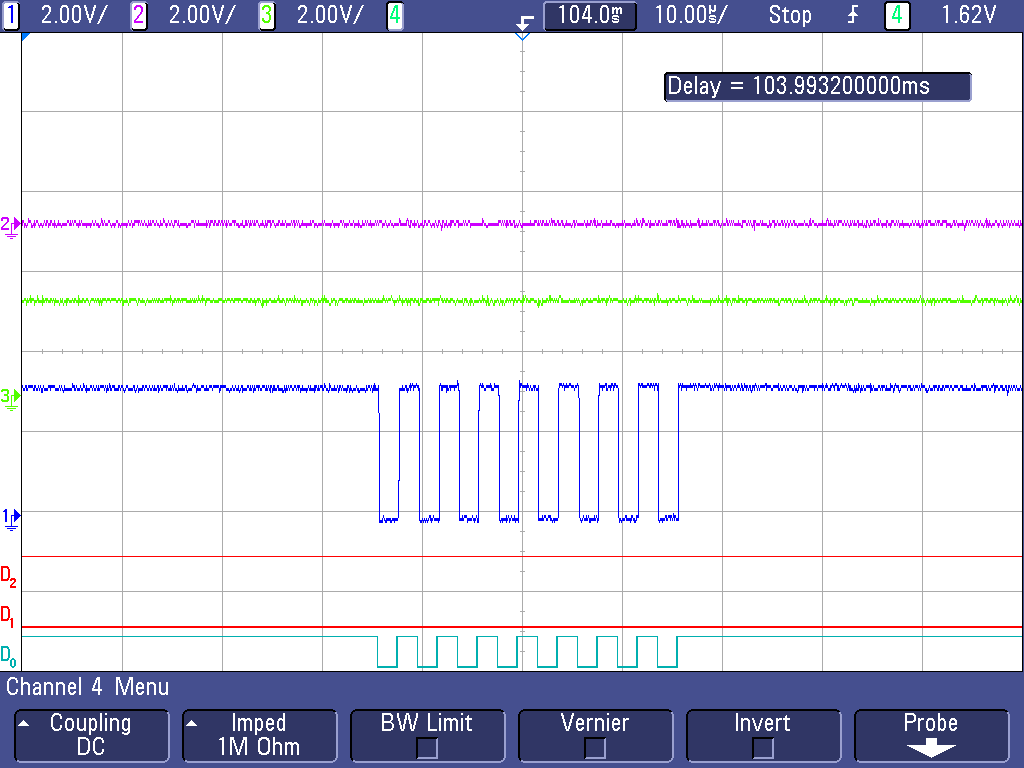


Diagram 15: Data being placed in Initial Register

This shows the information being loaded into the int\_enabled register or lack of information being loaded (00000000) so that in the initialization process interrupts cannot be triggered. D0 and 1 show the SPI clock, D1 and 2 show the output request from the controller and input of the sensor, D3 and 3 show the output from the sensor and the input of the controller.

### Serial Communication over Distant (Controller Node/ Sensors Node)

In order for our product to have serial communication form the controller node to the sensor node we will need to determine if there is any data loss cause by both distant and by the load generated by the sensor node. In order to do this we ill test communication with the addition of 10 feet of wire between the controller node and the sensor node and then in between every sensor node.

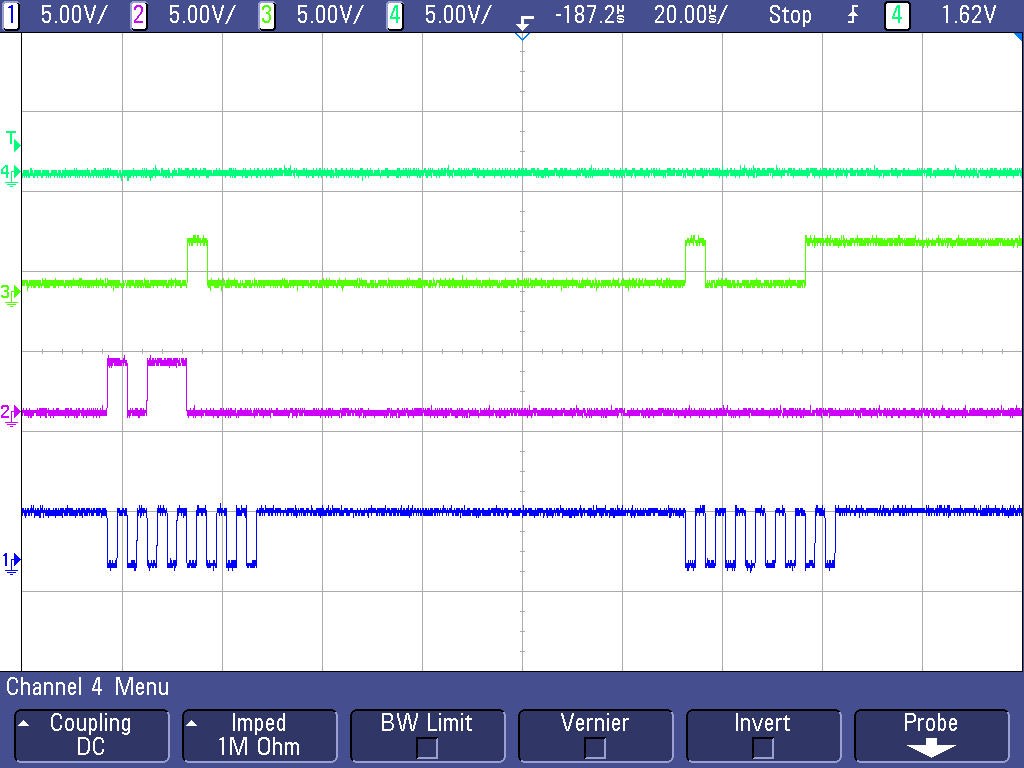


Diagram 16: Benchmark, One Sensor, 0 ft Distance

This is a view of the SPI communication without 10 feet of wires and no buffering of the signal. The signal on 1 is the SPI clock that is sent from the controller to the sensor for communication, 2 is the signal sent from the controller to the sensor to request the information stored in the Int\_enabled register, 3 is the signal sent from the sensor to the controller with the values stored in the int\_enabled register, and 4 is the control line that is pulled down on the sensor to indicate with sensor is in communication with the controller.

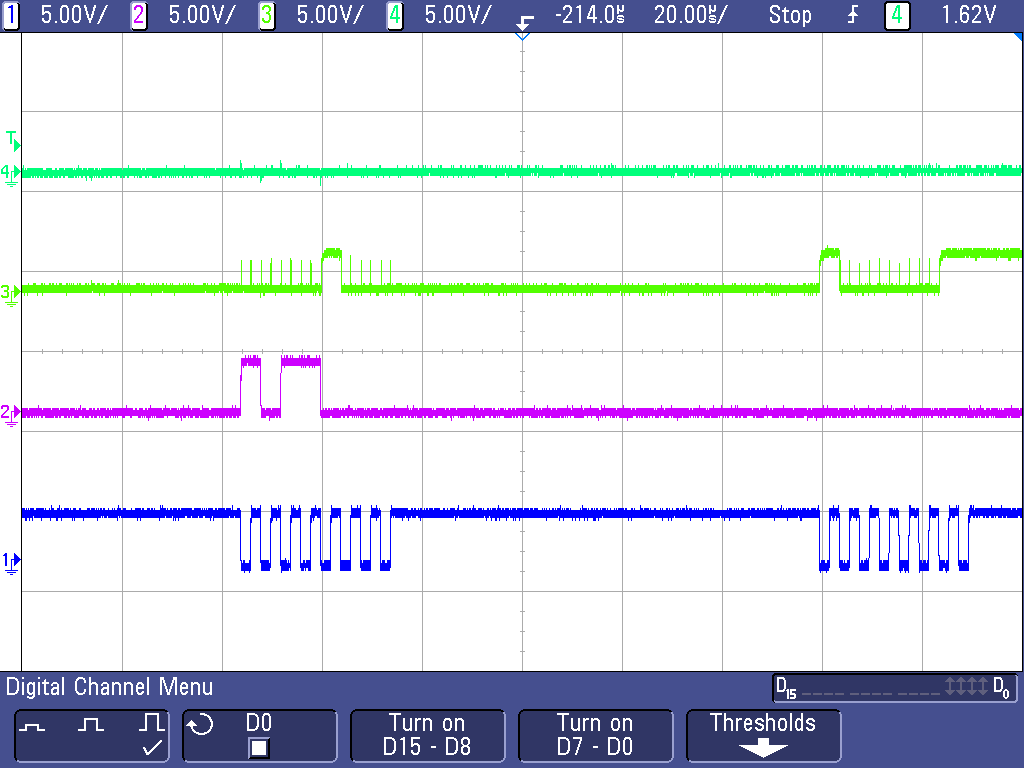


Diagram 17: Test, 1 Sensor, 10 ft Distance

This is with 10 feet of wires and with a buffer added. On signal 3 there is some peaks that are shone but the controller is still able to read the correct value. The signal on 1 is the SPI clock that is sent from the controller to the sensor for communication, 2 is the signal sent from the controller to the sensor to request the information stored in the Int\_enabled register, 3 is the signal sent from the sensor to the controller with the values stored in the int\_enabled register, and 4 is the control line that is pulled down on the sensor to indicate with sensor is in communication with the controller.

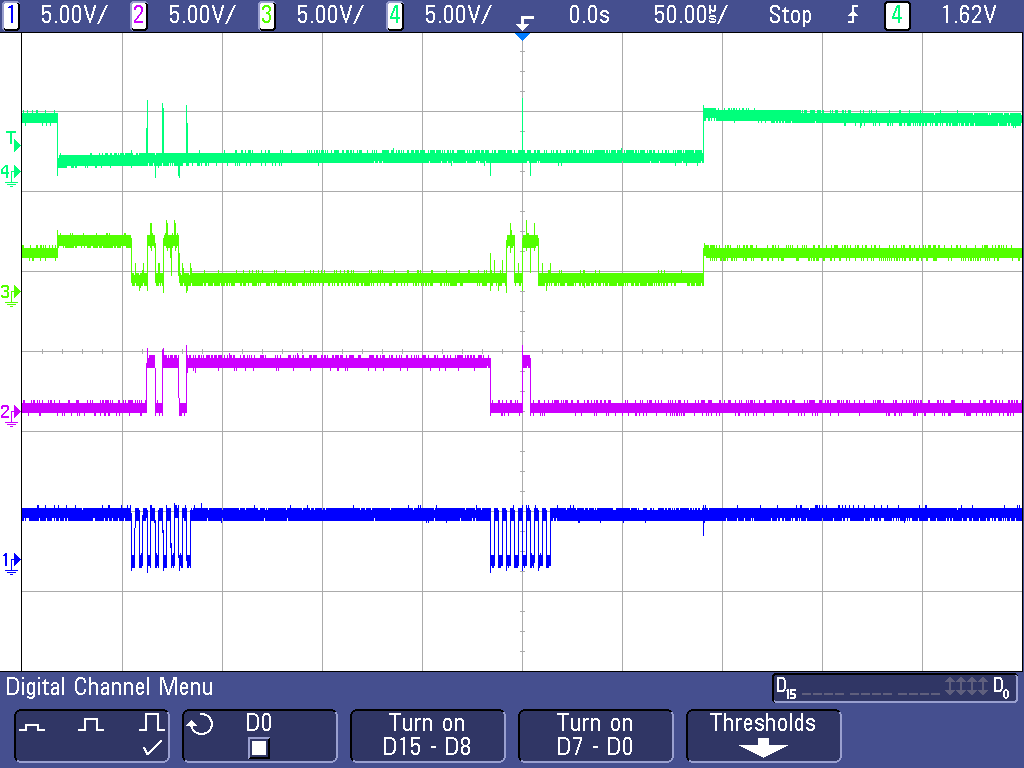


Diagram 18: Test, One Sensor, 10 ft Distance, Error Occurring

This is showing the error that occurs on the 10 feet of line without a buffer. The problem that occurs with communicating over distance with no buffer is that the control line will turn off and one randomly which will cause the sensor to not fully receive the information and will not sent the correct information. The signal on 1 is the SPI clock that is sent from the controller to the sensor for communication, 2 is the signal sent from the controller to the sensor to request the information stored in the Int\_enabled register, 3 is the signal sent from the sensor to the controller with the values stored in the int\_enabled register, and 4 is the control line that is pulled down on the sensor to indicate with sensor is in communication with the controller.

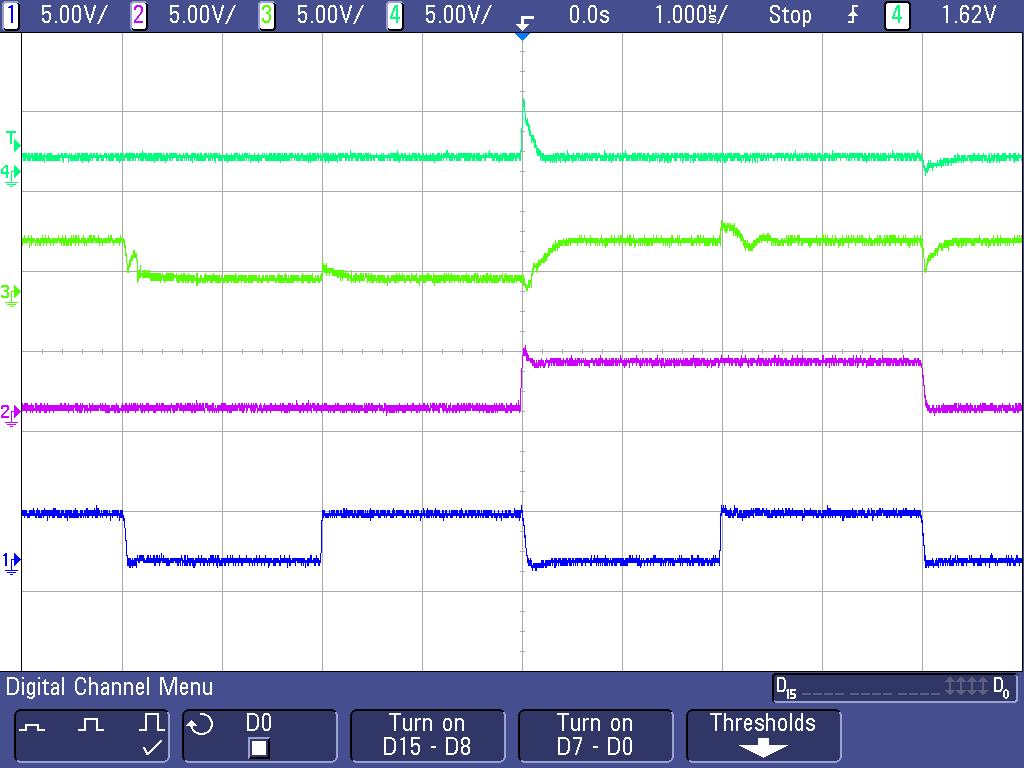


Diagram 19: Test, One Sensor, 10 ft Distantce, Error Effect on Communication

This is a more zoomed in look at the same error that is occurring in the image above but this is showing on line 4 the error that occurs. When the control line switches to high it cause an error to occur in the rest of the information and the rest of the communication wires will then sink to the control line error. The signal on 1 is the SPI clock that is sent from the controller to the sensor for communication, 2 is the signal sent from the controller to the sensor to request the information stored in the Int\_enabled register, 3 is the signal sent from the sensor to the controller with the values stored in the int\_enabled register, and 4 is the control line that is pulled down on the sensor to indicate with sensor is in communication with the controller.

This problem was fixed by adding a non-inverting smit trigger to all of the data communication lines. These allows for the signal to be cleaner when entering the controller node and the sensor node without these buffers communication would not be possible over distant.

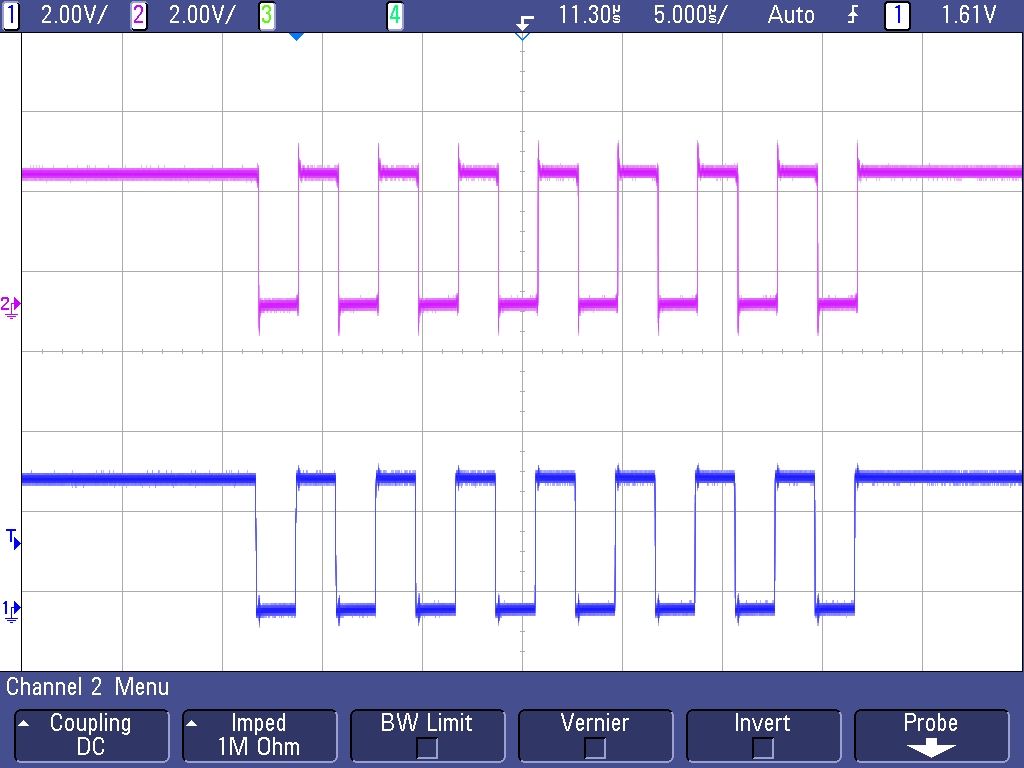


Diagram 20: Test, Eight Sensor, Buffers, 0 ft Distance

This shows the SPI clock being transmitted over the buffers without extra wire added. The bottom signal is the signal is the SPI clock being sent from the controller node. And the top signal is the signal after it has passed through 8 buffers.

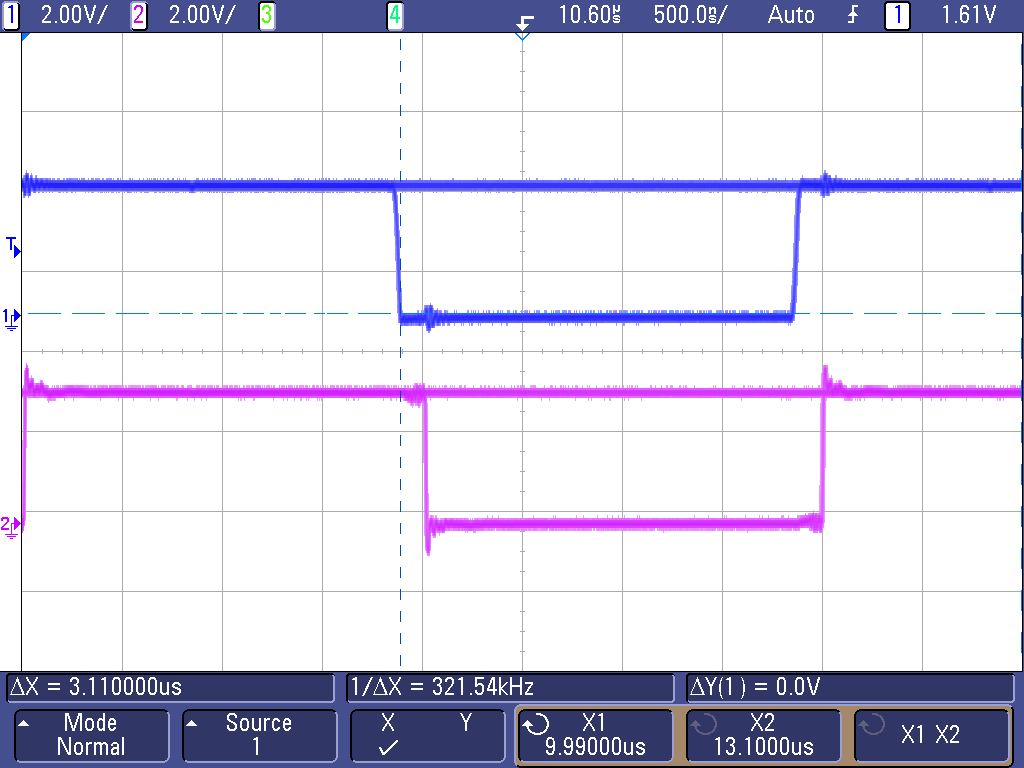


Diagram 21: Test, Eight Sensor, 0 ft Distance, Delay Added Initial Reading

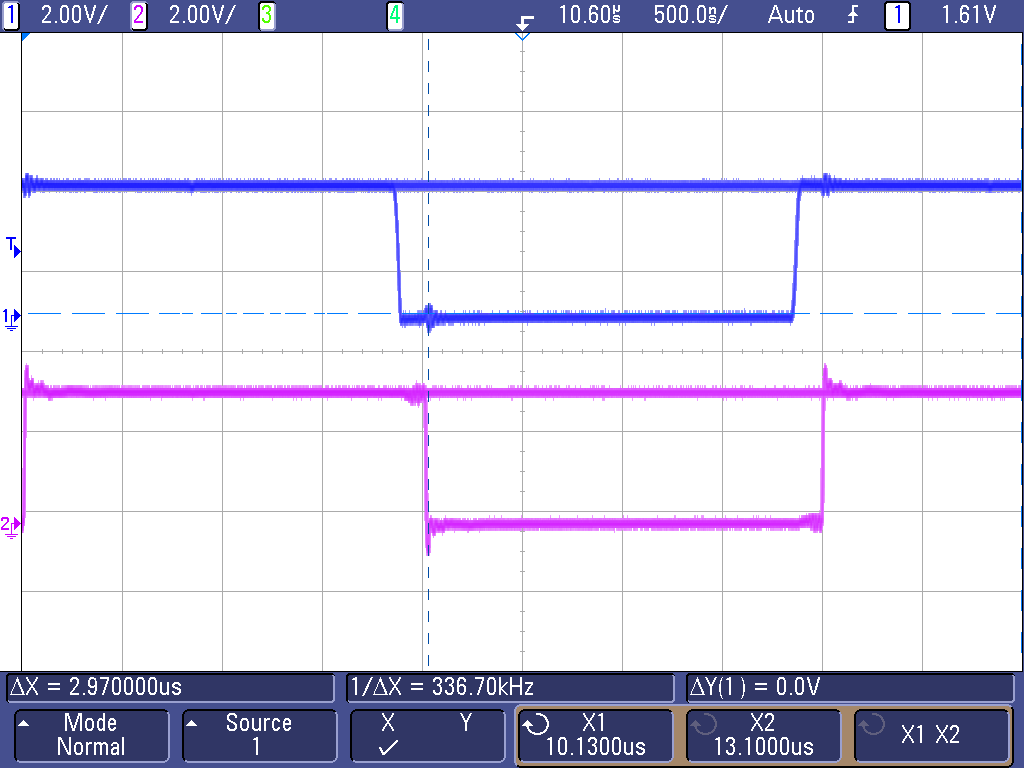


Diagram 22: Test, Eight Sensor, 0 ft Distance, Delay Added Second Reading

This is showing that there is a delay caused by the eight buffers but it is only .14us which is not enough of a delay to cause issues with communication to the sensor nodes. Also when all of the signals required are sent through the buffer they will have the same delay also negating any problem caused by a delay introduced by the buffers. The bottom signal is the signal is the SPI clock being sent from the controller node. And the top signal is the signal after it has passed through 8 buffers.

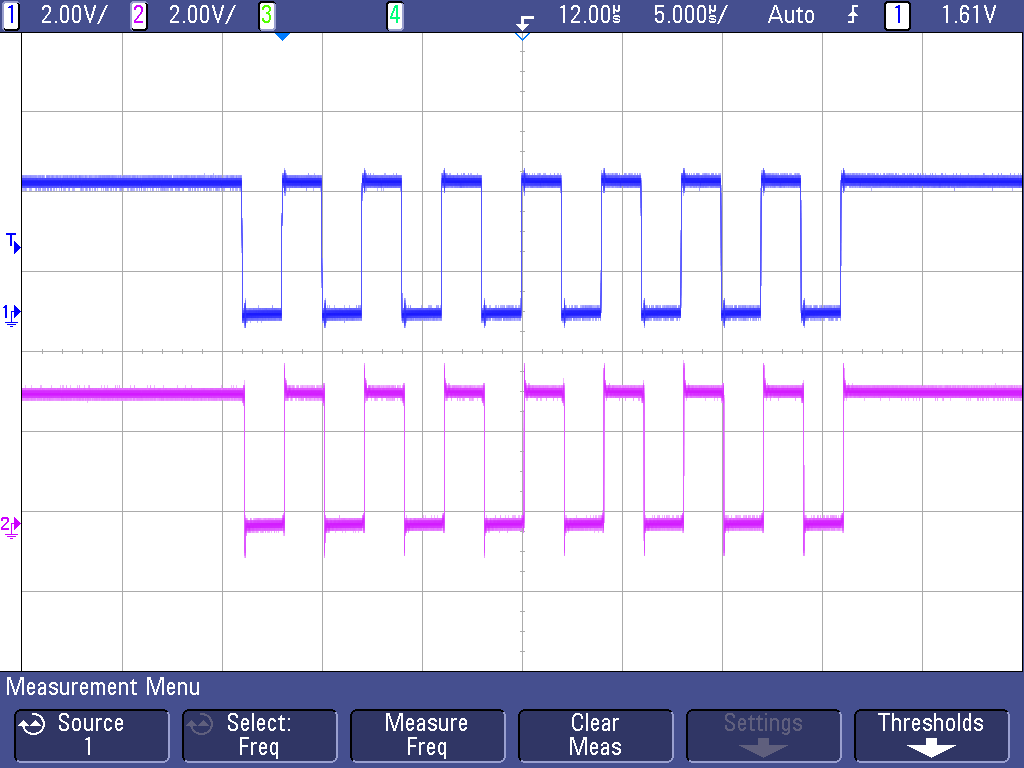


Diagram 23: Test, Eight Sensor, 10 ft in Between Each Node Total 80 ft Distance

This is showing the effect on the signal after the signal has passed through 8 buffers and also through 80 feet of wire. There is no loss of signal with the added wire and the buffers. With the wires added it adds a delay of only 0.05us to the 0.14us of delay that was added with the buffers. The bottom signal is the signal is the SPI clock being sent from the controller node. And the top signal is the signal after it has passed through 8 buffers.

### Parallel Communication over Distant (Sever / Controller Node)

Since all four controllers share the same TX/RX UART lines from the server, testing needed to be done to ensure the load of all four controllers being connected would not interfere with the communication even though only one would be transmitting at a time.

We also determined the longest distance our wires may be could be ten feet. We needed to be sure that with all controllers at a max distance combined with the load of all the controllers connected that the signal would not be lost.

Our test consisted of connecting 4 controllers to the TX/RX UART wires at a length of ten feet each.

We found that the system still worked with the distance and load attached. We connected several points to the scope to see how the signal was affected from the added load and distances. There was a noticeable difference, however the signal was still clear and acceptable to not need added buffers to improve it.

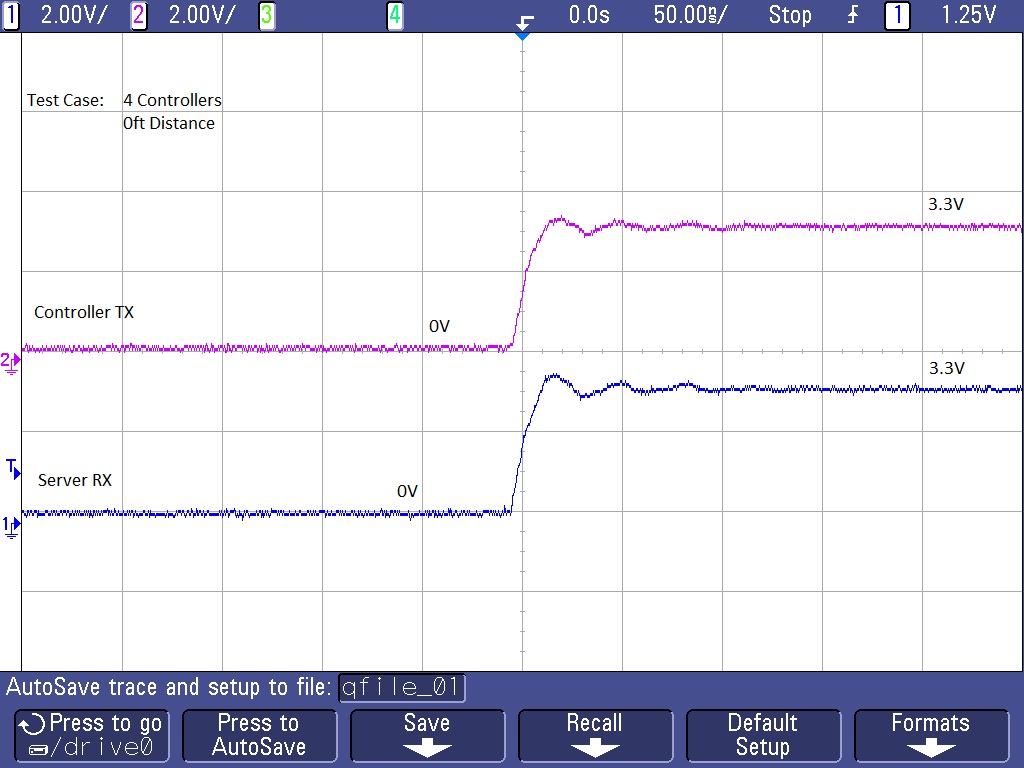


Diagram 24: Benchmark, Four Controllers, 0 ft Distant

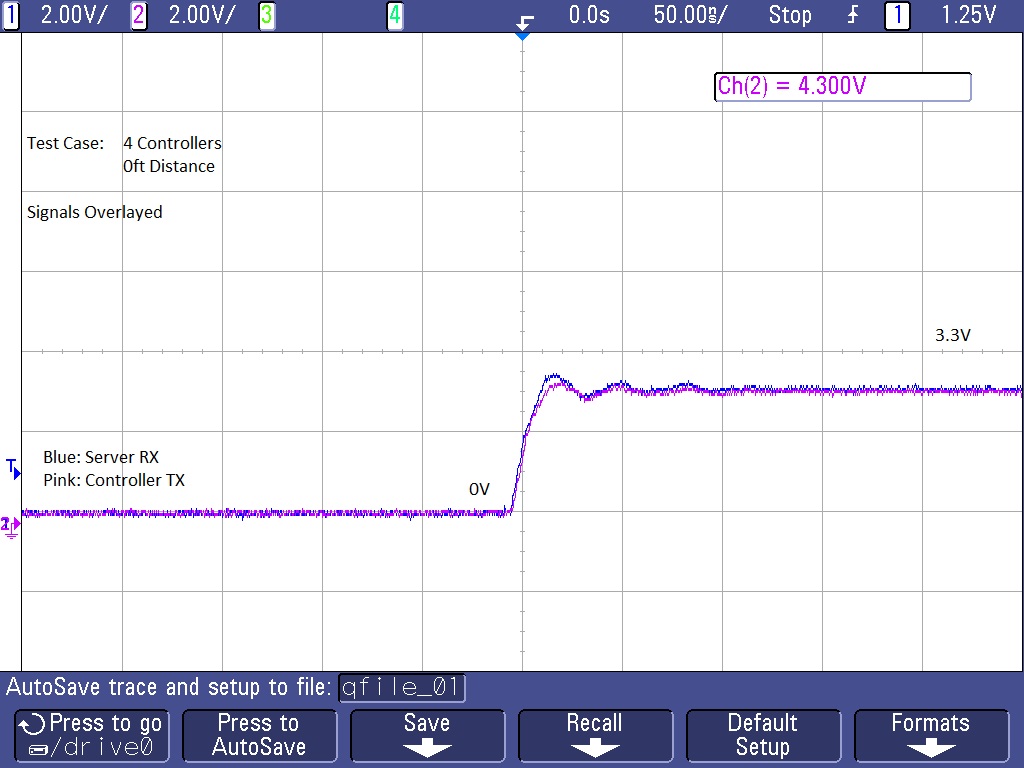


Diagram 25: Benchmark, Four Controllers, 0 ft Distant, Overlapped

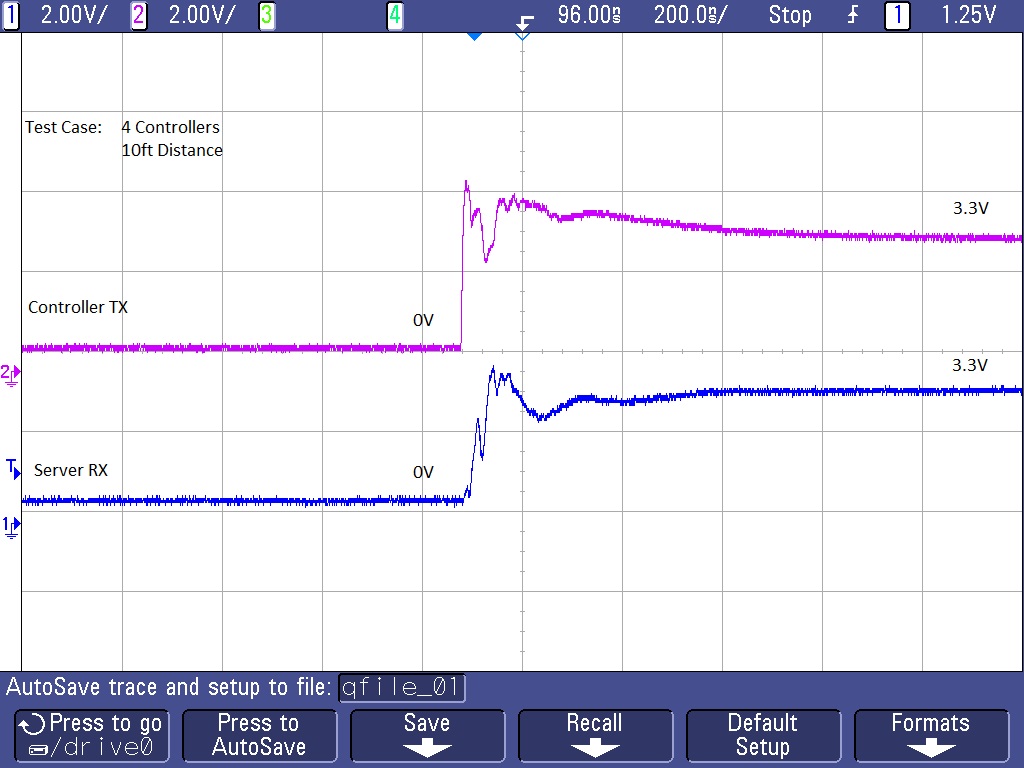


Diagram 26: Test 1, Four Controllers, 10 ft Distant

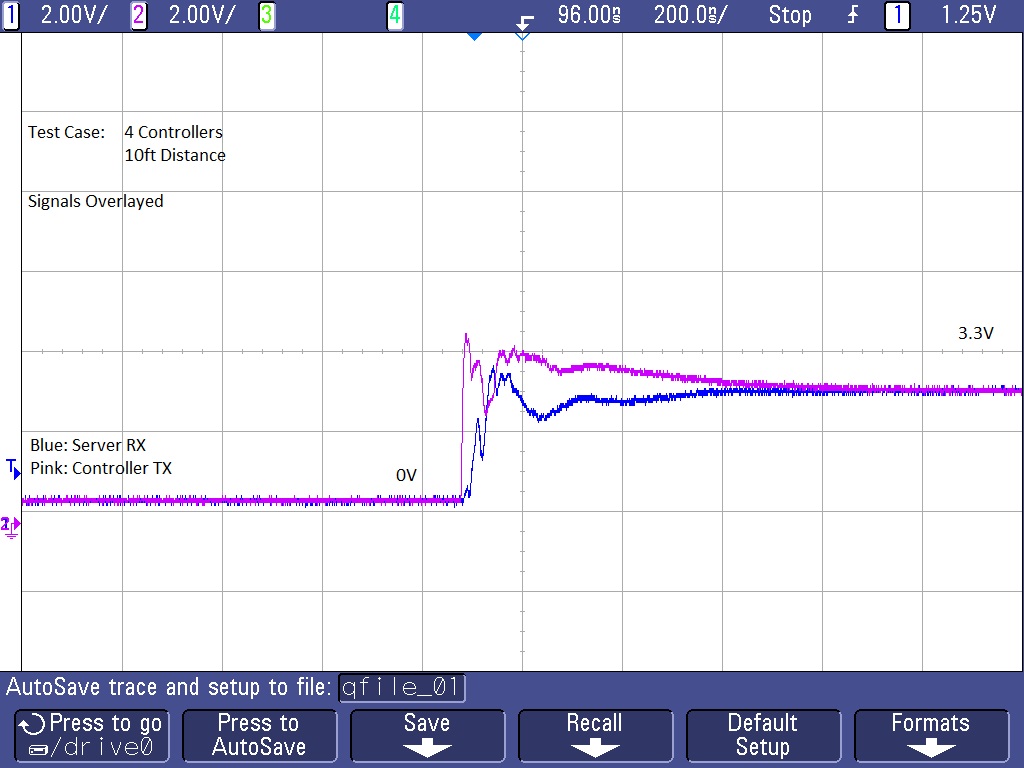


Diagram 27: Test 1, Four Controllers, 10 ft Distant, Overlapped

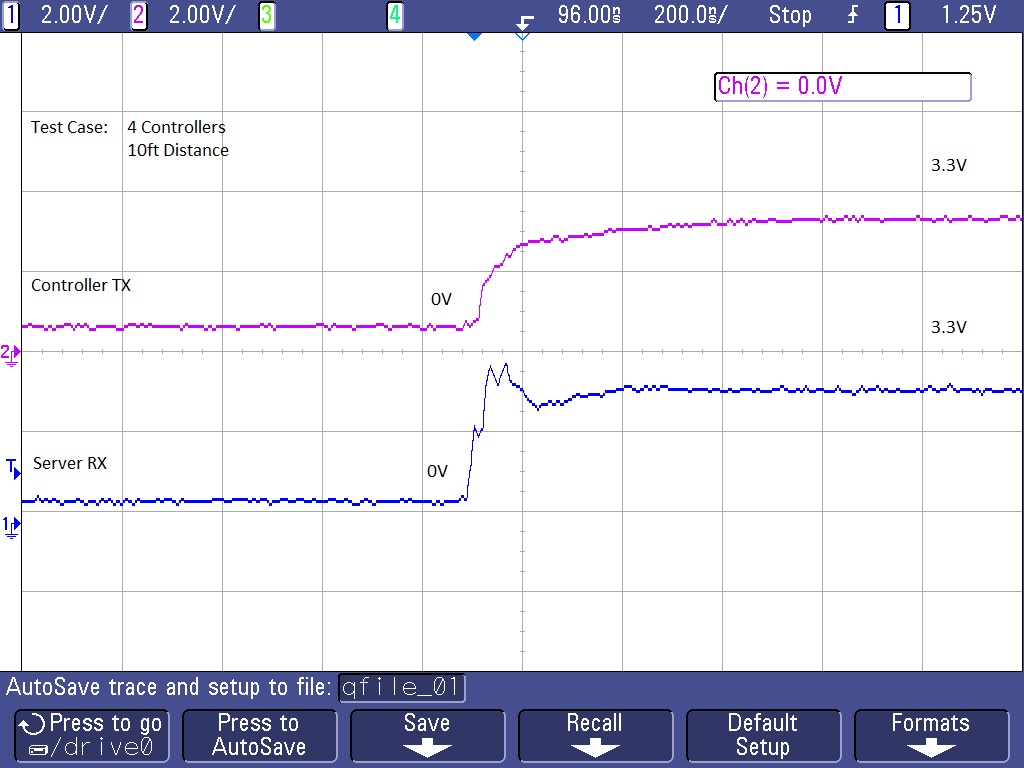


Diagram 28: Test 2, Four Controllers, 10 ft Distant

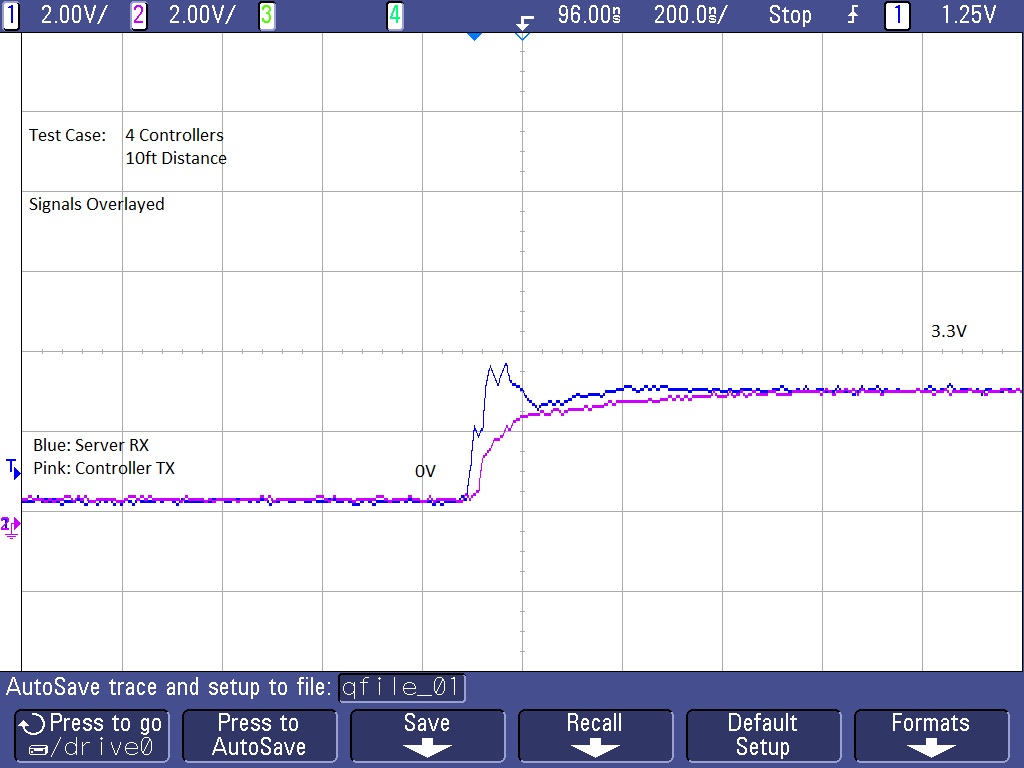


Diagram 29: Test 2, Four Controllers, 10 ft Distant, Overlapped

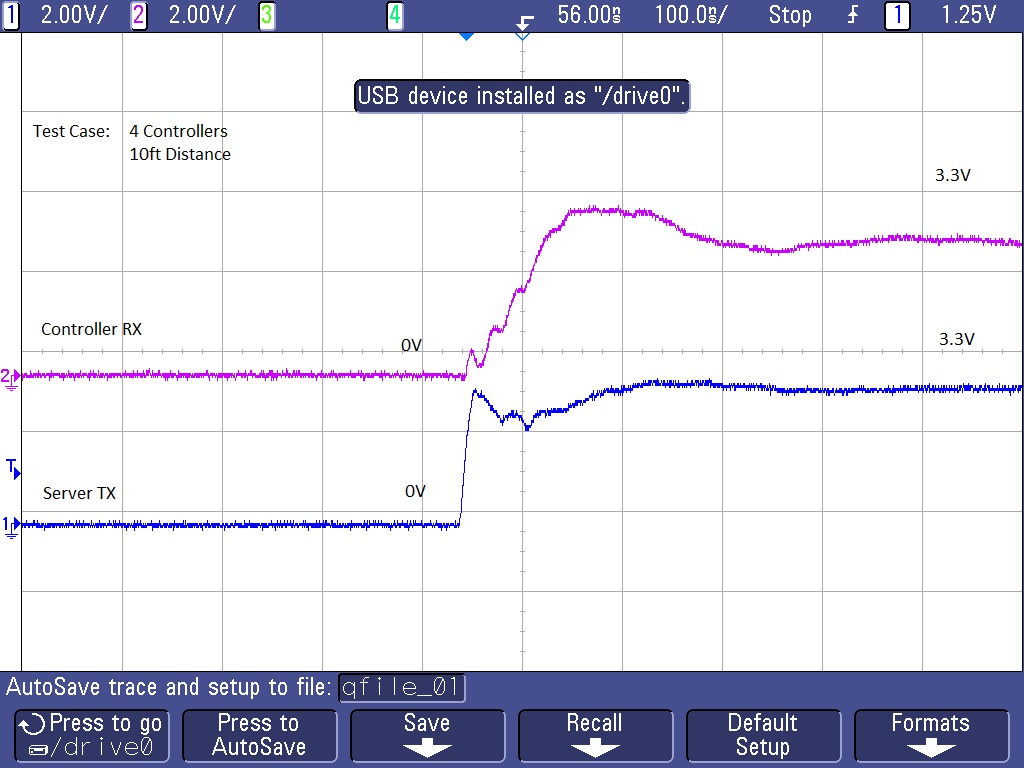


Diagram 30: Test 3, Four Controllers, 10 ft Distant

### Best Mounting Location

We tested three different locations of the sensor on the dryer. Ideally, the best location would have a large range of values in order to best determine the status of the machine. The first location was on the back of the dryer (side that faces the wall), along the top center. This location provided the best movement data with the highest overall average of values. This location was also the least susceptible to outside interference such as the sensor being bumped.

The second location we tested was the side of the machine, in the center. This location provided the smallest range of values. The panel along the sides acted as a buffer from the internal movement. This may have been intentional from the manufacturer to reduce vibrations and noise resulting from the movement within the machine. This location also could possibly interfere with machines on either side.

The third location tested was the top of the machine in the center. The range of values in this location was better than the side, but not as high as the movement range on the back of the machine. This location was also the least practical, and most susceptible to outside interference.

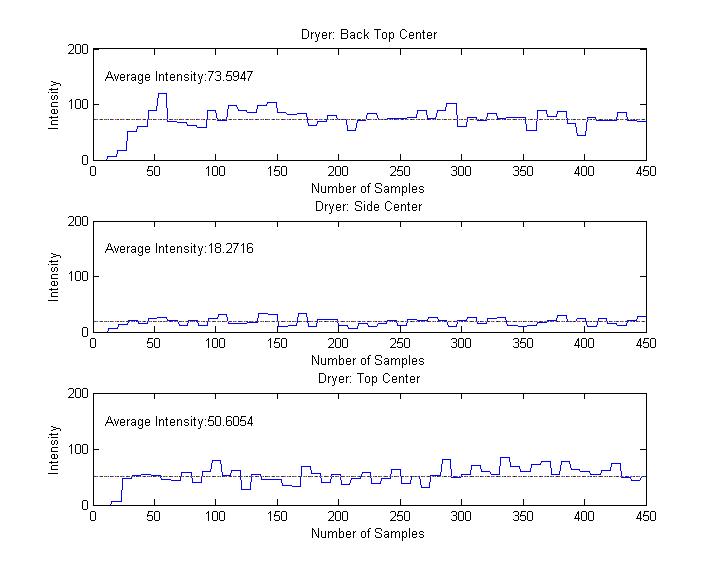


Diagram 31: Sensor Mounting Locations

In the Process of determining the best mounting location for the sensor we also found that the best measurements that could be achieved was with the sensor attached to the machine and not on a breadboard. In order for our system to work properly and reliably we will have to mount the sensor directly to the machine. In order to do this the best means that we found was by attaching a magnet to the sensor and then having the magnet be the connection form the machine to the sensor.

It was also determined in this process that our system will need two separted system state detection programs. One that can be used for driers and one that can be used by washers. With the current system state detection our product will detect the correct operating state of the system for driers but has problems with washing machines. This problem is cause by the wash cycle which has larger portions of the run time with the machine not moving.

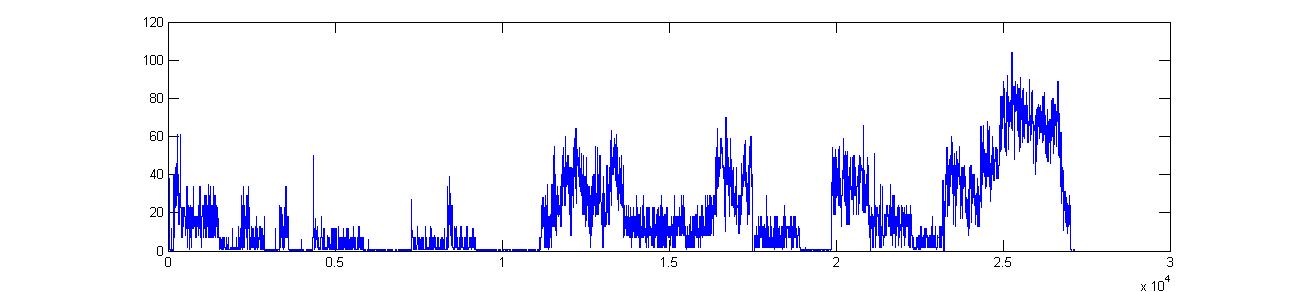


Diagram 32: Full Wash Cycle

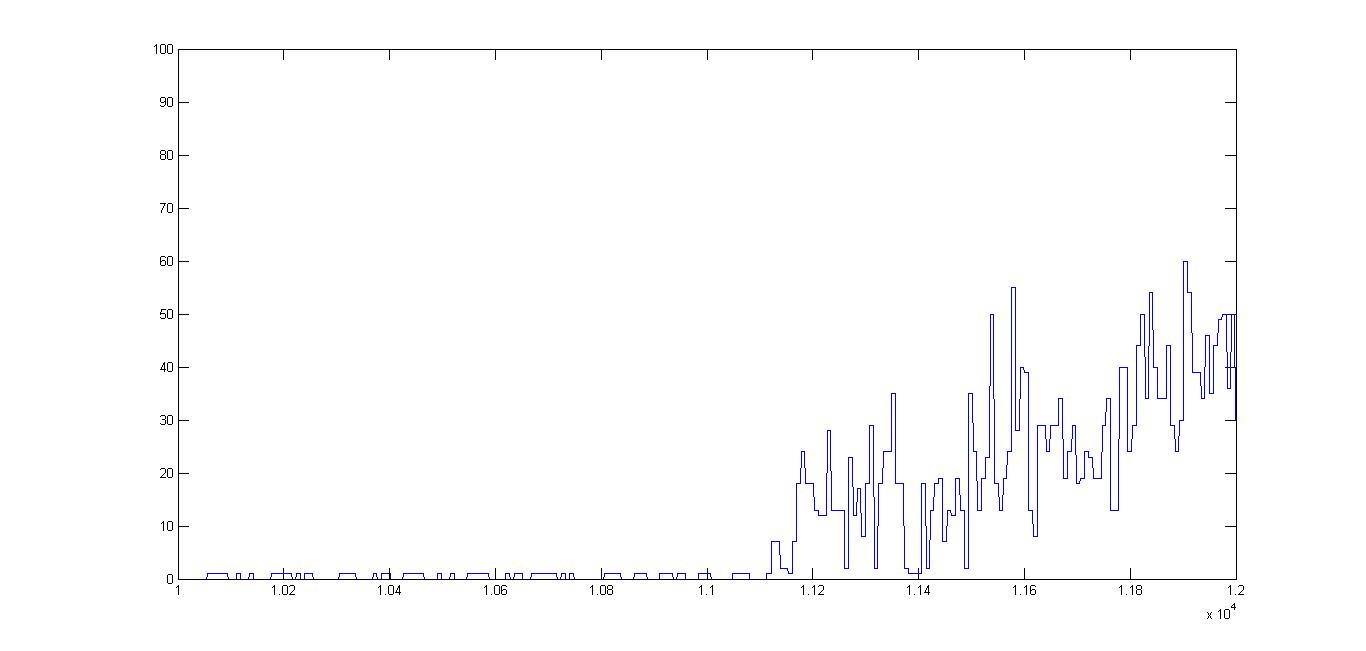


Diagram 33: Zoomed in Look at the Wash Cycle

### WI-FI Distant

We tested the distance the WF32 could be from the router in two different scenarios. The first test was indoors and the approximate distance that was achieved was 50ft. This test was through an interior wall and one floor below the router. The second test that we conducted was outside with no obstructions. The router was placed just inside a window, and the WF32 was able to be approximately 180 feet away.

### Machine Configuration

What machine configurations are necessary for Laundry Now to properly work, there are several different configurations that might be set up in different laundry rooms. From machines standing alone, machines touching each-other, machine connected to each-other (stacked, side by side), and finally stacked and side by side.

### Power Usage

What is the power usage of the system based on the number of controller nodes, number of sensors and total distance of the wire connecting all of the different nodes to the sever.

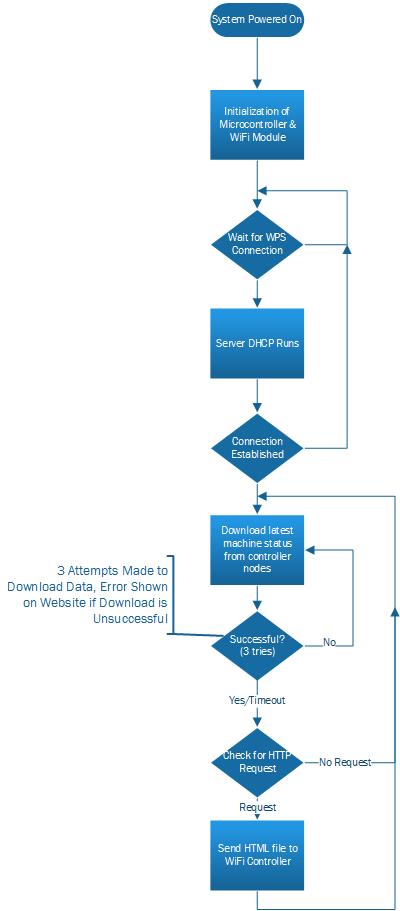
### False Triggers

Is it possible for a machine to be placed into cycling mode without the machine running. Some false triggers that might cause this to occur is if another machine is running right next to the machine in question, earthquakes, a kick, the process of loading the machine or un loading the machine, someone sitting on the machine.

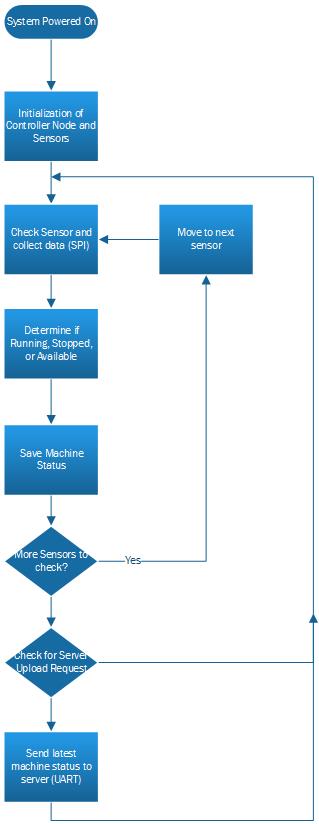
### Long Term Usage

Determination of potential system problems that might occur with the system running long term. Possibilities are sensor node malfunctioning, loss of connect due to the machine movement, slaughter breaks cause by movement, potential calendar problems with storing the number of cycles per machine. The best means to test this problem is by placing the system in a laundry room and checking the connections daily or weekly to see if errors are occurring.

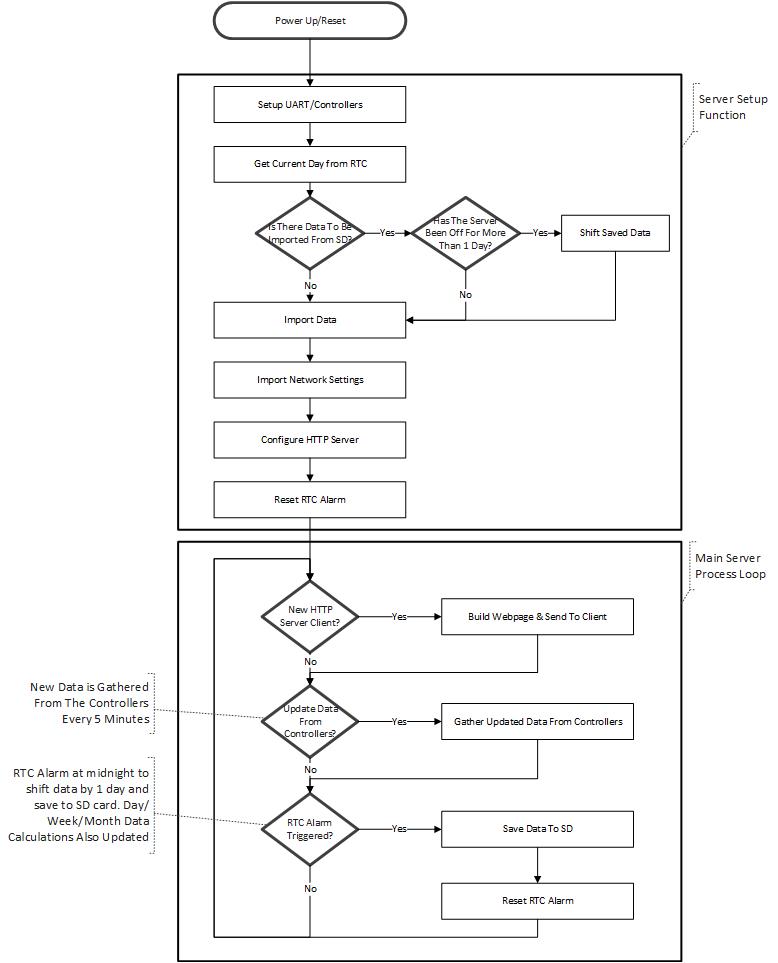
## Flow Charts



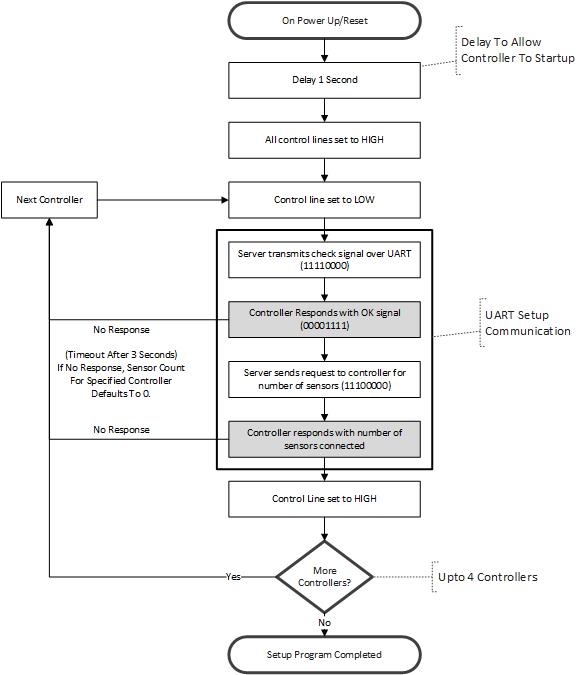
Flow Chart 1: Initial Server



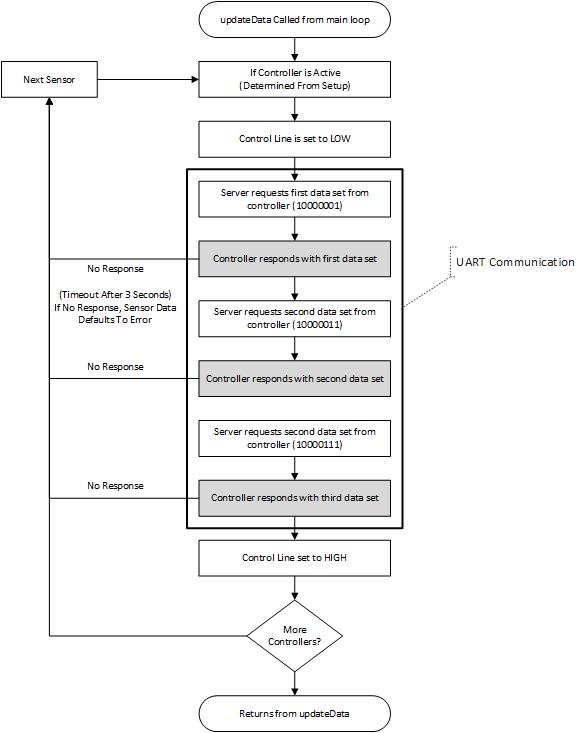
Flow Chart 2: Initial Controller Node to Sensor Node



Flow Chart 3: Higher Level Sever System Block Diagram



Flow Chart 4: UART Initial Request



Flow Chart 5: UART Data Request



Flow Chart 6: Higher Level Controller Node Block Diagram



Flow Chart 7: Three State Diagram



Flow Chart 8: Three State System State Determination



Flow Chart 9: Four State Diagram

## Circuit Diagrams

## C:\Users\Daniel Monforte\Senior Design Project\relevent documents\server_sch.JPG

Diagram 34: Sever Schematic

## 

Diagram 35: Controller Node Schematic

## 

Diagram 34: Sensor Node Schematic

## User Interface

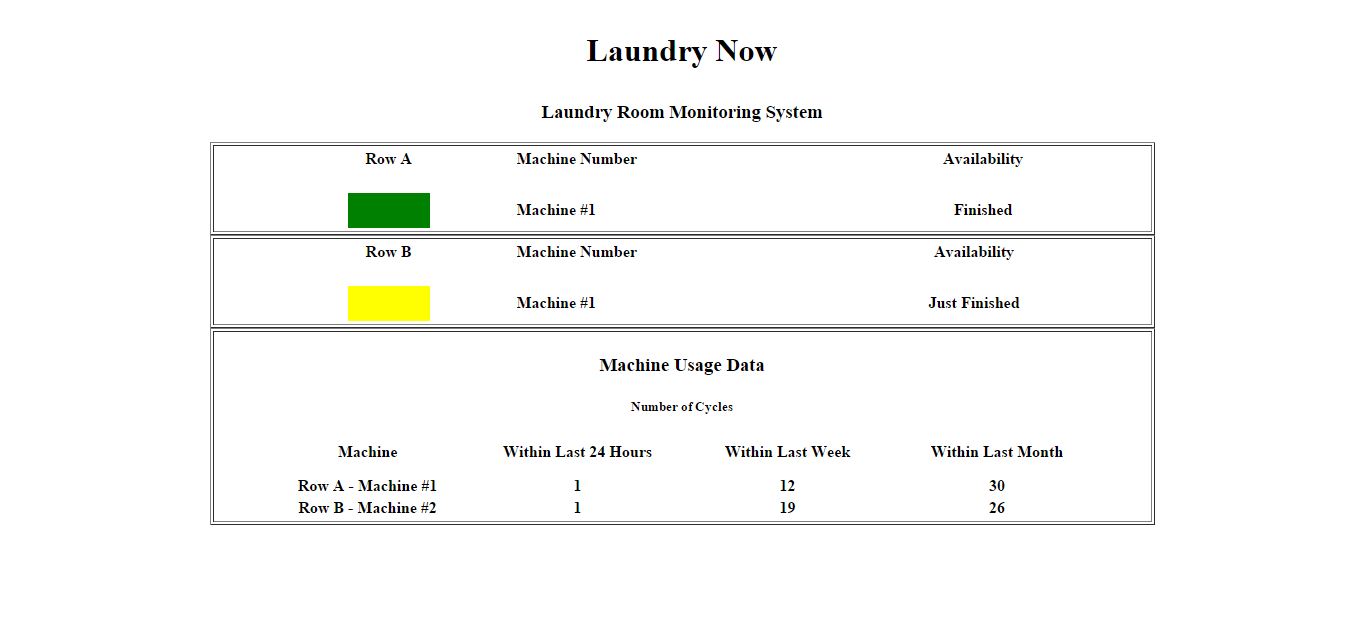


Diagram 37: Webpage Layout



Diagram 38: QR Code