

# 1.0 Executive Summary

This project aims to be a useful tool for the energy conscious consumer. Whether a person knows much about household power consumption or not, our home power management system will provide a way to more efficiently monitor, budget, and control your monthly electric bill.

The system consists of a central unit with a VGA touch screen and radio transceiver, a custom circuit breaker subsystem, and a kilowatt per hour meter attached to outlets. The circuit breaker and power meters are subsystems that will report to the main unit via radio frequency transmitters and transceivers. A series of microcontrollers will facilitate the transmission of necessary data. Via this main unit, the user will be capable of setting up the system based on the desired preferences. The user can provide the monthly rate they are charged per kWh by the power company. The main unit will be receiving information about the rate of consumption from the meter setup on the outlet, and can then be used to begin budgeting the power usage.

The control over the system is one of the most unique and exciting features. One of the objectives is to use the circuit breaker subsystem to be able to command removal of power to a given outlet. The user can determine which outlets the system can choose to cut power to if too much power is being consumed based on the reading from the kWh meter. It can also be used to trip a single breaker if the item is deemed unnecessary. For instance, one useful function would be if the user accidentally left the oven or TV on for an extended period of time, then the system would be able to notify the user that potentially unnecessary power consumption is occurring. However, the user could just be watching the TV for a very long time, or cooking something that requires a long time, for this reason the system would not simply cut the power after a certain amount of time. It would simply notify the user that the main unit is about to command the breaker to cut power to the appropriate outlet.

Thus, the idea to implement a mobile application was born. This is a potential option if we are able to complete it. The application could be used to have remote access to the system so that the user can always be able to respond to notifications that the system would generate as conditions arise. The user would also always be able to command a shut-down of certain devices remotely. At this point, the application is a desired feature, but is not included in the current design.

## 2.0 Project Description

Creating a centralized unit paired along with satellite components capable of tracking power consumption at each individual power outlet and reporting back to the user the kilowatt per hour measurement of each device plugged in is the goal of the design. The motivation behind this design stems from the lack of power monitoring in current day homes. Most homes where the owner has not purchased individual devices to plug into each outlet does not include a way other than going outside to read a meter to determine

how much power is being used. Given the option to read the external meter, a homeowner would not know the breakdown of why the amount of power consumed is where it is. That is where the design of this project tries to take over. Using an array of micro-controllers with a centralized unit can give the user an ability to track individual devices and make the necessary adjustments to control how much power they are using as well as being aware of how much power they are using.

At the center of the system is a Video Graphics Array (VGA) screen, roughly seven to eight inches in size that is driven by a microcontroller that will display to the user different information about the devices that are currently being powered within a home. The software will be written to display each outlet and the power being consumed on each port. From the measurements taken at each outlet a monthly bill can be calculated and displayed to the user to allow them to make adjustments if necessary. To begin with a user can enter how much per kilowatt hour their power company charges them and using that information it will be possible to calculate a monthly bill based off what is being reported to the main unit. The VGA display using the micro-controller and available memory can have the option to keep track of previous months bills and energy usage to give the user an idea of how much power they have been using over an extended period of time and if they have been making adjustments. Other pertinent information that can be displayed can be a manipulation of the measurements already recorded. This can include any various charts and histories that are deemed useful.

The satellite components of the system include the micro-controllers that will be used to measure the kilowatt per hour power usage at the outlet. The microcontroller will be designed to measure individual draw at each port instead of the outlet as a whole so a user can better diagnose which device is drawing which amount of power. It is still uncertain if some implementation can be accomplished which will allow reading across a device such as a surge protector. At the minimum the overall power draw from the surge protector that is plugged into a wall can be reported to the main unit depending on which outlet is drawing the power. Each micro-controller will use radio frequencies to send and receive information and the current method to be implemented is radio-frequency identification (RFID), however other possibilities may be considered depending on the available technology. The use of RFID is necessary because the microcontroller will sit behind an outlet out of sight from the user and transfer the information at request from the RFID reader at the main unit. Each micro-controller will also have a simple liquid crystal display (LCD) soldered into the board that will display the kilowatt hour measurement. This will allow for debugging and verifying that the information being sent to the VGA display is indeed accurate.

Other option being considered includes a remote controlled circuit breaker. The circuit breaker will be installed with the microcontroller at the individual outlets. The main device can be programmed by the user to allow it to turn off devices if certain thresholds are being crossed. As an example, a user can specify what devices are permitted to be

turned off in the event that at current monthly bill calculations, the power usage is surpassing monthly expectations. The device will be intelligent to make the best guess at turning off the appropriate device or devices with as little interference as possible to lower the power consumption. The main unit can be designed to sound an alarm before a device is turned off, alerting the user that the event is about to occur without surprising them. In addition, the tripping of the circuit breaker can be canceled by the user and they can then proceed to turn off appropriate devices if they choose to.

In addition to the circuit breaker, an application is being considered that can run on certain mobile operating systems such as Android. This will allow the user to control the monitoring system from their handset instead of the VGA wall unit. The monitoring from the mobile unit will include the same features and will allow the user to keep track of power consumption as well as review past usage. The application will also be able to receive alerts if any device is going to trip the circuit breaker at a specific outlet, allowing the user to make adjustments or cancel the operation.

## **2.1 Project Motivation**

The motivation behind this project stems from receiving an electric bill every month that is higher than expected despite making a wholehearted effort to reduce power usage. It can be difficult throughout the course of a month to keep track of how much power is being used and what exactly is using the most. That is where the idea of this project takes over and that is to help power conscience users keep track of their power consumption throughout the month by revealing what devices are consuming how much power and reporting it to the user.

By raising consumption awareness, users would show more concern for the energy used and/or wasted. With renewable and efficient energy becoming a strong focus for many groups, having homeowners more aware and alert would help to the cause as it would cut down energy consumption.

While other prototypes of this system have more than likely been attempted and completed several times over they have never been implemented in the home. It may be due to the overall cost of implementing the system but if every home had one it is reasonable to assume that power consumption could be reduced. A system that would report to a user over a network how much power they are using in comparison to their neighbors could help motivate people to lower their usage. There are many things that people do over a month and furthermore the year that wastes energy and if it could be displayed to them in their home and in a precise way rather than seeking out a power meter somewhere outside, it could make them more aware of how much power they really are using, rather than just seeing a bill once a month.

## 2.2 Goals and Objectives

The main goal of this design is to reduce the overall power consumption in homes that implement this type of system. The main unit that displays all of the information will report each device in the home and the amount of power it is using. In addition a history can be kept that helps a user track power consumption from month to month in an effort to reduce overall usage.

### 2.2.1 User Friendly

One key focus every project has is that the unit is simple enough so that any able bodied individual will be able to use it without any difficulty. The project's goal is to provide users a cheap and simple method to monitor, track, and control their power consumption. If a method is unable to become simple, then making the learning methods of utilizing the product easy to follow is an alternative solution.

With the use of a Video Graphics Array screen, the user will be able to view many different features provided by the project. The most basic feature is their current overall power consumption and projected energy bill. Other features will include the ability to monitor consumption coming from individual outlets or segments of the household. All of these features will be set up so that the user is only a few clicks away from having access.

With the possible addition of a smart phone application, the user may be able to have access to all of the features provided by the project through their smart phone. The images and information displayed will be done so in a very simplistic manner. There will be charts and tables to show the user the information they are searching for without overwhelming the user with a screen full of number and characters. The visualization and navigation throughout the menu and features will be as simple as clicking a button.

### 2.2.2 Real World Application

There are always situations where roommates will not agree on their portion of the utilities/electricity bill. With this unit, the users will be able to not only monitor their consumption, but they will also be able to view the consumption based on sections of the house. In other words, college students for example would be able to see who is using the most power in the household.

This feature would allow for a fairer distribution of costs when it comes to the monthly billing cycles. With the consumption tracked and accessibility being convenient, these users would be more aware of their behaviors and cut back on their consumption. With the ability to determine where the most power was being consumed, users could begin to take the necessary steps to reduce consumption and begin to make changes immediately

and continually throughout the cycle instead of taking actions following the end of a cycle.

Alongside with real world application is scalability. With the time and resource constraint, the features of the unit will provide only basic information. Users will be able to view their consumption, estimated bill cost, and even the consumption from different outlets. The goal of having this project being able to be modified in complexity depending on the user would allow for different types of information to be displayed. If an everyday user is merely interested in how much energy is being consumed then that would be a very basic usage/function. Now say a user is very adamant about reducing power consumption. By monitoring their use and making a few changes to the software side, the user would be capable of displaying information such as overall percentage reduction, amount of energy and money saved, as well as taking future steps to reduce this consumption even further.

### **2.2.3 Consumption Awareness**

It can be difficult throughout the course of a month to keep track of how much power is being used and what exactly is using the most. That is where the idea of this project takes over and that is to help power conscience users keep track of

their power consumption throughout the month by revealing what devices are consuming how much power and report it back to the user.

With the ability to proactively stay in check of their power consumption, users will more than likely begin to cut back in areas of consumption they see as unnecessary. Just being aware of their consumption will be enough of a drive and motivation for users to cut back on their usage.

Some individuals see their monthly bills and make a wholehearted attempt to cut back on consumption in hopes of lowering their next bill. But as the month goes on, the user will forget about the promise they made to themselves. This project will be able to remind the user of their goal via smart phone reminders or a message displayed on the VGA screen. With that constant reminder the user will become much more aware and their wholehearted efforts to lower consumption will be successful.

## **2.3 Project Requirements and Specifications**

There are plenty of ideas that we want to implement into this design. Due to the time restraints as well as lack of experience, we will be picking only the ideas that we are comfortable with completing within the allotted timeframe.

### **2.3.1 Basic Requirements**

**Main Unit:** The main unit shall consist of the microcontroller driven VGA screen that shall serve as the central source of interaction between the user and the system

**VGA Screen:** The VGA screen shall be roughly 8 inches in diameter, and be capable of displaying a standard 640 x 480 image.

**Main Microcontroller Subsystem:** The subsystem shall be capable of driving the display on the VGA screen to interface with the user. It shall also incorporate an RFID receiver to read the signals sent from each outlet containing the power consumption data.

**Outlet Microcontroller Subsystems:** Each outlet being monitored shall contain a microcontroller based subsystem within the outlet. The system shall contain a kilowatt hour meter capable of reporting, via radio frequencies, power consumption data to the main unit. The system shall not only report the data to the main unit, but also display the power consumption data in real time to verify the connections between each outlet and the main unit.

**Circuit Breakers:** Each outlet shall contain individual circuit breakers that can cut power to an outlet that is exceeding user specified amounts of power, is not in use but is still consuming power, or is simply deemed to be unnecessary by the user.

**Power Supply:** The power supply shall be capable of providing constant power to the system. The system needs to be able to run in the background, or be capable of being turned on and off remotely for small sample intervals to provide data as appropriate.

**User Interface:** The user shall specify the conditions they would like the power consumption to reflect. The system shall be capable of receiving the user's input and reference the data received from the outlets.

The main unit shall receive data from each individual outlet that is being monitored, via the radio frequency transmitters. The data shall be displayed on the VGA screen, and referenced with the conditions provided by the user. If a threshold is crossed by a certain device, the circuit breaker on that outlet shall be capable of cutting power to that device.

If implicated, there shall be an application capable of allowing the user to trigger the breaker on an outlet subsystem and cut power to a certain device. This may be automatically triggered by the system receiving power consumption totals that are too high, or if the user decides to trigger it themselves. The main unit shall determine if consumption levels are too high and interface with the application to notify the user of potential solutions, and trigger the breaker if no action is taken within a set period of time.

The breakers shall be capable of automatically removing power from any device, due to levels exceeding user conditions, or by the user manually commanding a breaker.

The system shall be capable of running on constant power in the background, or being turned on or off remotely via manual startup or the application.

### **2.3.2 Hardware Requirements**

The main unit shall contain a circuit board with the main microcontroller, a 15 pin female connector, a series of resistors and diodes to create a digital to analog converter that will control the image data sent to the display, and an RFID receiver.

The outlet subsystems shall contain one to two microcontrollers, one to two kilowatt hour meters with LCD displays, one RF transmitter, and one to two custom circuit breakers.

### **2.3.3 Desired Features**

For this project there are many desired features that should be implemented.

**Power Consumption Tracker:** The project will be able to accurately determine how much power has been consumed from the outlets that are connected by this project. By being able to determine the power consumption, the project will then in turn calculate the projected energy cost for that session. There will be a single kilowatt per hour meter at each outlet to read the consumption. Each unit will be labeled its respective use or location, i.e. water heater, master bedroom, etc.

The project would be able to take this data and display them visually. The user will be able to choose how they would desire to view the data, whether through charts, tables, or lists. The user will also have the option to view the numbers as percentages from the previous cycle so that they can further see their usage now compared to then.

With these values, the user would then be able to set a limit for themselves, both mentally and physically. By utilizing this project, the user will be able to input a limit to how much energy they want to consume for the time being. Once the consumption level approaches eighty to ninety percent of their limit, the user will be alerted and this will cue the user to cut back on their consumption.

**Wireless Data Transmission:** All components of the project will use radio frequencies to transmit data between the microcontrollers. This will allow for certain components, such as the kilowatt hour meter, to be placed safely behind walls. Wireless transmission is a must in order for the user to be able to access the data from their smart phone. Having all the information transmitted to the main unit through radio frequencies will also allow for the user to have units set up further away (within range of the signal) without fear of figuring out how and where to run cables.

With the option of using wireless data transmission, the possibilities of expanding the project ideas open up wider. Since all the data is being sent through radio frequencies, the users could potentially upload this information to an online database to further expand their memory base.

**Segmented Household:** A second feature includes the ability to section off desired portions of the home. This feature is primarily for the use in homes where the utilities bills are typically split or for those under individual leases. A lot of college students tend to get into debates when it comes to splitting the utility bills mainly when one tenant is concerned with the bill due to other's power consumption. With the ability to section off areas of the home, these tenants could see what percentage of the power consumption was their own and scale it to their respective payments.

**Automated Breakers:** Another feature will be the option to cut off power via circuit breaker when your consumption has reached an amount set by the user. The user can set a predetermined value, whether it is monetary or power consumption, and once the threshold is broken, the breaker will automatically cut the power off. This method will only work in a forward fashion, in other words, the power can be automatically disconnected, but not reconnected.

### **2.3.4 Bonus Features**

This section will discuss features that while seem ideal for the project, they are out of reach due to the time restraints.

**Smart House:** There are some desired features that may be out of reach at the moment due to time restraints. One of these features includes incorporating a smart house program into the main central hub. From this VGA screen, the user will be able to control the entire home in terms of power.

On top of being accessible from the central unit, the user will be able to control everything through their smart phone. With the completion of the project and the smart home project, an app will be incorporated so that the user may maintain control over their home from anywhere as long as there is a stable internet connection. The user may interact with the unit as if they were right in front of it. With many interactive menus, the user will be able to view, in detail, the power consumption from any portion of the house. From there the user may choose to power off any portion of the home whether its individual outlets or entire sections of the home.

One fun bonus feature would be to give the unit an option to alert the user as they're approaching any preset limits of power consumption. The alert could be as simple as a noise, or to make it a little more exciting, it could be a recording of the user's choosing.



## 3.0 Research Related to Project Definition

Before beginning to design the project, research was done to discover what products already existed and used to guide our group towards what was needed.

### 3.1 Existing Similar Project and Products

The idea of home power monitoring isn't exactly brand new, in fact there are many examples of products that do what this project is trying to accomplish. This project will try to accomplish a cheaper and more effective method for monitoring and controlling appliance power consumption that is easy to use and not be too bulky. There are many variations of these monitoring systems, but this paper will only mention three similar products that each have something that this project would try to emulate.

#### 3.1.1 P3 International P4200 Kill A Watt Wireless Electricity Usage Monitor

The Kill-A-Watt Wireless (Figure 3.1.1), a unit from the P3 International company, that claims to help reduce electricity costs by showing how much is being spent per device being tracked, and it shows how big of a carbon footprint is being left behind by the user's consumption of power.



**Figure 3.1.1: The Kill-A-Watt® Wireless**

The Kill-A-Watt Wireless boasts an ability to monitor up to 8 different devices simultaneously within a 300 foot range. Each sensor sends its data to an LCD screen which displays various measurements. The Kill-A-Watt shows the power consumption in kilowatt hours, the carbon footprint in pounds of used carbon based on the type of electricity the house is being powered by, the price of the power used so far, the actual

current, the actual voltage, and the power factor (the phase value). It also has the ability to forecast the price of using the device being tracked over the course of a day, week, and even year.

P3 International makes installation of the Kill-A-Watt easy to do. The user simply places one of the sensors into a socket and then plugs the device the user wishes to be tracked into the sensor. This can be done for up to eight different devices around the house so long as they are within 300 feet of the display. Once all devices that are desired to be tracked are in place, the user then looks to the hub of the system which is on a 5"x 4" LCD display and chooses the energy source that their energy provider uses such as coal, hydro, atomic, or etc. The system should then be able to make all the necessary calculations from there.

The Kill-A-Watt Wireless system seems like an excellent model to aspire to, but that doesn't mean that it doesn't have its own shortcomings. The price of the set for example is a hefty \$69.99 for the display and only one sensor. In order to track any more than just one device, it would cost \$35.00 for each additional sensor making a grand total of \$314.99 before taxes for the full capability of this product. Another downside to this product is that the sensors are very large and only one may be used per wall outlet because the bulk of it would cover the other plug. This would make it difficult to track multiple devices in the same room. On the positive side however, the system is wireless, giving the user greater freedom and ease, and the system shows many important data feeds. Both qualities would be excellent to achieve in this project, especially if the cost could be driven below the asking price.

### **3.1.2 Belkin Conserve Insight Energy-Use Monitor**

A very similar piece of technology is from the Belkin company called the Conserve Insight (Figure 3.1.2). It claims to show the user relatively the same amount of information as the Kill-A-Watt does, showing the carbon footprint data to alert the user to any devices that are consuming too much energy. There are differences between the two, however, show how varied these devices can be.



**Figure 3.1.2: The Belkin Conserve Insight™**

The Conserve Insight claims to reveal the real energy wasters in the user's home by providing tracking data of the device's wattage, cost of operation, and the amount of carbon dioxide used in pounds over the course of a month or a year. The Insight also shows the data in real time while the device is in use. This data is shown on a small screen that can sit on a table or another surface where it can be easily seen by the user.

The way the Conserve Insight works is very simple. The system's detector plugs into one socket and then the user plugs a device into the detector. The Insight will then automatically begin to send data to the screen about the consumption levels of the device that is being used. The screen itself has a few buttons to switch between what type of data is shown carbon, cost, or power. This project would like to emulate the simplicity of the Insight, and also provide what it lacks.

One of the shortcomings of the Insight is that it is a one-to-one system. The screen connects to the detector by a cord and thus can only show information about one device at a time. The cause of this perhaps, is that it is necessary to keep costs down and the system size small. The tiny size of the system truly promotes how portable home energy management systems can be, a feature that we can hopefully match. The Conserve Insight also has a very simplistic user interface, involving only three buttons on the screen to show different information, and anything that makes life easier for the user is better overall. The Conserve Insight lacks a wireless capability, so it requires a very close proximity to the device being tracked. While a long cord counteracts this, the need to be connected via a wire is unseemly in this era of technology. The good thing about this technology though is the low price of \$30 for the entire unit. Since it is a one-to-one system, no other additional parts are needed making this an ideal unit to track one device in particular like a television, refrigerator, or computer.

### 3.1.3 TED (The Energy Detective) 5000C

The Energy Detective pulls all the stops in its design. It seems to be the dream product of home energy detection because it has so many features. TED (Figure 3.1.3) is the idea that this project would like to achieve, but is probably beyond our capabilities and time constraints. TED is one of the few products out there that actually advertises its accuracy in detecting power. These specifications are what can be taken from this product



**Figure 3.1.3: The Energy Detective 5000C**

The home page of TED shows that they are proud to claim that the average user saves 5-30% on their electric bill, that they have a 99% accuracy, and that their meter is sensitive to within one watt of the true output. TED can also be used to detect solar and wind power, project monthly bills, and not only is it wireless, but the data can be found on a website so any mobile device with a web browser can view it anywhere in the world. TED really does seem to be the end all be all, and it would be this project's biggest mentor and opposition all at once.

TED has a very different approach when it comes to installation. Where the Kill-A-Watt and the Conserve Insight simply plug into an outlet, TED goes the extra mile and asks the user to connect a sensor to the circuit breaker box in the house. The website even recommends that the user call an electrician or another certified professional to handle installation because it involves removing the panel and attaching wires directly into empty circuit breakers. When that is done, two small current transformers are placed around the incoming power lines to the house. The user is then required to find an empty outlet close to their router that is on the same circuit that the sensor is connected to, and connect an Ethernet cable from the gateway to the router. Finally, after registering the

serial number of the product in the website of TED, the user is done with installing and can begin to see data on any internet capable device. TED will then show the power output of every device in the house, as well as CO2 emissions, and the cost of operation in real-time. The software also shows projections of continued usage, and graphs and charts of the history of usage.

The really big downside to TED, aside from the rather technical installation, is the enormous cost to the user. There are many different versions of TED for differing types of businesses, but for the regular residential user the cost is a whopping \$240 for the system. While that price isn't as large as the full Kill-A-Watt system, the Kill-A-Watt can be used at the base level for \$70 and TED cannot which can deter a large amount of potential users. The fact that the entire house can be monitored, however, and that the data can be accessed on the web is very appealing. Implementing a web or app based element would be a great bonus to the project's accessibility, but the actual thing to take from the TED is the confidence to claim 99% accuracy and the ability to save people at least 5% on their electric bills.

## **3.2 Relevant Technology**

There are many different elements of technology involved in home power monitoring. There needs to be a way of measuring voltage and current, a way to send this data wirelessly to other devices, and an interface to interact with this data. These technologies have all been developed separately, and it is this project's task to combine them in a useful manner to create a system that will communicate electrical data between itself and the user seamlessly. A basic understanding of the different elements of technology is required to sufficiently explain what it is that this system does.

### **3.2.1 Amperage/Wattage meters**

Home power monitoring, at its most basic level, is the ability to detect levels of voltage and current that is being used. In order to do this, a device such as a voltmeter or ammeter could be used to detect these measurements. A volt, named after Italian physicist Alessandro Volta, who is said to have invented the first chemical battery, is defined as the difference in electric potential energy between two points. Electrical current is the measure of electrical charge carried by the flow of electrons through a surface, which is usually a wire. These two values are necessary for calculating the electrical power at any given time.

Power can mean a great deal of things in this world. In physics, power is the measure of the amount of energy used per unit time. It is measured in watts, in honor of James Watt, whose advances in steam engines fueled the Industrial Revolution, and one watt is equivalent to one joule of energy per second. The formula for calculating electrical power is different depending on the type of current. For direct current (DC), power is measured

through the multiplication of voltage and current at any point in the circuit. Power can also be found with resistive current by finding either the current squared multiplied by the resistance or voltage squared divided by resistance. Most homes and businesses do not run on DC current however, and they use alternating current (AC) instead. In an AC circuit, the current of electricity is sent through the wire in an alternating sine wave pattern. Since the current is continually changing in this manner, the simple equations to find power need to change. For AC circuits, power is calculated by taking the root-mean-square values of the voltage and current waves and multiplying them together with the cosine of the phase angle between the two wavelengths. Because these calculations can be quite difficult, especially since the waves can vary frequently, there are wattmeters to make these calculations automatically to make life easier.



#### **Digital 60V/100A Balance Voltage Battery Power Analyzer RC Watt Meter Checker**

Wattmeters are designed to measure watts by taking simultaneous voltage and current measurements and calculating the power. The original design involved coils in series that would measure the electromagnetic field generated from the current traveling through the coils. A parallel circuit would then measure the voltage, and the internal software would then multiply the values accordingly. Most wattmeters have an LCD display that shows the value of the data in watts or kilowatts and many other variations depending on the make and model of the meter. For most residential purposes, power is measured in kilowatt hours to determine the cost of the electric bill. This project will need to either use a preexisting wattmeter to implement in the design, or create a small wattmeter ourselves, whichever seems more appropriate. A homemade meter is easily possible, but the accuracy of such a thing would be the deciding factor, and would require much testing before being put to use properly.

### 3.2.2 Wireless Transmitters

Wireless technology, although it has been around since the invention of the radio, wasn't being used to its full advantage aside from broadcasting purposes until recent decades. While products like the Belkin Conserve Insight show that wireless technology isn't completely necessary for a device to be useful or marketable in this era, wireless nonetheless is a feature that is highly desired in most electronic devices today.

In order to have greater ease of access to information, and to be more appealing to users, a wireless component is almost expected. The entirety of wireless technology centers around electromagnetic waves. The most commonly used frequencies belong to the radio waves, as they are used for everything from cellphones and GPS, to garage door openers and toys. They have, overall, a relatively long wavelength. Radio waves can travel through the air and through most mediums at close to the speed of light with very little loss of signal integrity. It is due to this property that we are able to communicate and control a robot on another planet from here on Earth.

Radio transmitters all use the same major components to get their message out. There must be an antenna, a receiver, and software to decode the data. In order for a wireless transmitter to send a message, first an internal software must encode it in machine language. The encoding differs between every device, but in almost all cases it boils down to a stream of binary code. The code itself becomes electrical pulses which travel along the transmitter's antenna. The antenna is, in essence, an electromagnetic emitter because it is nothing more than a strip of metal with a coil of wire around it to force the electric current to induct a field at the frequency of an internal oscillator. On the receiving end, the radio waves hit a sensor which sends the signal to an amplifier within the receiver. The amplifier boosts the signal which, after traveling through a medium such as air, space, water, and walls, may have lost some of its integrity, and recreates the electrical impulses which are then re-interpreted as ones and zeroes. The software then takes the binary and decodes it to do whatever it is that the transmitter was made for. A device which contains both a transmitter and receiver is called a transceiver.

The world is inundated with all sorts of electromagnetic waves, so in order to determine which signal is the right one, the receiver is tuned to the matching the frequency of the transmitter's oscillator by a bandpass filter in the circuit. The filter reduces signals from other frequencies and only allows the selected frequency through. Frequency modulation is what allows data to be sent through a carrier wave. The data is combined with the carrier radio wave, altering its phase. This combined signal is what the receiver picks up and decodes it. The transmitter's responsibility in this project would be to take the information received from the wattmeter's sensors, convert it to a radio signal, and have the central hub find the signal and decode it to display the results. Again in this case it is entirely possible to create radio frequency transmitters and receivers using resistors, capacitors, inductors, and diodes, but due to time restrictions and cost, it would be a much better and cheaper decision to simply buy a small RF transmitter, and code the software to our purposes.

### 3.2.3 VGA LCD display

Detecting and sending results would mean nothing if they weren't able to be displayed in some manner. Digital screens have been available for quite some time. Screens have become a part of normal everyday life for humans living in first world economies. There is a screen for every phone, every computer, every television, every tablet, digital clocks, security systems, navigational systems, in cars, in planes, and even in small child's toys. To display the home power data, it goes almost without saying, that a screen of some sort would come into play. The screen would show all relevant data that the user would like to see in a format that is easily understood. Screens can now even take input from the user in the form of touch with today's technology.

One of the most common displays is the liquid crystal display or LCD. The liquid crystal is beneficial to use because unlike the cathode array that was used to display images in the past, the LCD doesn't eventually burn images into the screen, nor does it take much room. In fact, the liquid crystals in between the screen are only microns thick. LCD's make use of the predictability of the liquid crystals' reaction to electric current to display desired images. It works by placing the liquid crystal between two panes of polarized glass. One pane is set perpendicular to the other, and the liquid crystal twists so that it is oriented with both panes of glass, this creates a path that light can pass through. When an electric current is applied to the liquid crystal however, it untwists and the crystal structures obscure the path effectively blocking the light and making that section appear darker. In order to create numbers and letters, tiny electrodes are placed between the glass and the liquid crystal layer and are attached to a circuit. A program then determines when to send current to particular segments of each electrode to create the desired image. In this manner any number of messages and values can be displayed. Showing the data isn't enough however, as there needs to be a way to interact with the data, to change what is being displayed.

Touch screens allow the user to interact with what is seen directly through the use of a finger or stylus. Touch screens were invented in the 1970's, but have only become very prevalent within the last couple decades. Touchscreens can now be seen implemented everywhere. They are now widely used with new smartphones, tablets, bank ATM's, and pretty much any new electronic device that contains a screen interface. Touch is one of the main human senses, so using that to interact with technology makes it feel very natural; thus it makes perfect sense to use touch to invoke objects such as buttons on a screen. There are many different types of touch screen technology as it can be resistive, capacitive, surface acoustic, infrared, and etc. The type chosen for this project will be a resistive touchscreen because it is low cost and requires only pressure to respond as opposed to using bio-electric energy fields or a specific tool.



### 3.3 Possible Architectures

After going over similar ideas and the biggest technologies that this project will be taking advantage of, it becomes necessary to talk about how we plan on actually implementing these technologies to make a home power monitoring system. The similar products section shows three prime examples of the different directions we could take in creating this project. These are the options that are available to us to create our own home monitor. Trying to develop a TED-like device would be incredible, but is probably beyond our means and collective knowledge, so instead we shall look at what other products have and use some ideas from those architectures.

The Belkin Conserve Insight is the perfect example of how a one unit device would be. This option would involve placing a current and volt sensor that can be plugged into an outlet and extends the outlet to allow another device to be plugged int. Once the connection is complete between the outlet and a device, the sensor should begin to detect the electric levels and send them to the hub to be displayed. While this option will obviously work, it is far too simple. To make it more complex, we will need to add a wireless capability. Going wireless would also allow our product to detect more than just one device if we make more than one sensor to send data to the central hub. The small outlet sensor and simple interface however are good features that can be pulled into this project.

The design for the Kill-A-Watt (Figure 3.3.1) wireless is actually very close to what we actually hope to achieve. It has a wireless hub that can detect up to 8 different devices at once, a sensor design that fits into most outlets, and a simple display and interface. The downside with this design is that the sensor itself is too large. It requires two outlets at least, instead of just one which is something we hope we can improve upon. The screen is small and has only three buttons, but we would like to implement a better screen that makes use of touch technology and could possibly display information without any buttons at all. Creating such a display would involve some complicated programming to incorporate the touch screen, but that is not beyond our abilities.



**Figure 3.3.1: The Kill-A-Watt® PS-10**

While the multiple sensor option could work well, there is also this idea. We could have one unit that allows multiple devices to plug into it. It can still transmit the data from each separate device wirelessly, and we can save time and effort by making a multi-use sensor unit instead of multiple sensors. The display doesn't have to be embedded in the object like the PS-10, so the touch screen could still be used to see the data. In this design formation we also have the added opportunity to possibly shut down a block of devices if there is too much power wasted or to prevent ghost power drain from devices that aren't even powered on. This extra feature may not be implemented, but it is a good idea nonetheless, and would be an added bonus if we could get it to work.

## **4.0 Project Hardware and Software Design Details**

### **4.1 Initial Design Architectures and Related Diagrams**

Creating a centralized unit paired along with satellite components capable of tracking power consumption at each individual power outlet and reporting back to the user the kilowatt per hour measurement of each device plugged in is the goal of the design. The motivation behind this design stems from the lack of power monitoring in current day homes. Most homes where the owner has not purchased individual devices to plug into each outlet does not include a way other than going outside to read a meter to determine how much power is being used. Given the option to read the external meter, a home owner would not know the breakdown of why the amount of power consumed is where it is.

That is where the design of this project tries to take over. Using an array of micro-controllers with a centralized unit can give the user an ability to track individual devices and make the necessary adjustments to control how much power they are using as well as being aware of how much power they are using.

At the center of the system is a Video Graphics Array (VGA) screen, roughly seven to eight inches in size that is driven by a micro-controller that will display to the user different information about the devices that are currently being powered within a home. The software will be written to display each outlet and the power being consumed on each port. From the measurements taken at each outlet a monthly bill can be calculated and displayed to the user to allow them to make adjustments if necessary. To begin with a user can enter how much per kilowatt hour their power company charges them and using that information it will be possible to calculate a monthly bill based off what is being reported to the main unit. The VGA display using the micro-controller and available memory can have the option to keep track of previous months bills and energy usage to give the user an idea of how much power they have been using over an extended period of time and if they have been making adjustments. Other pertinent information

that can be displayed can be a manipulation of the measurements already recorded. This can include any various charts and histories that are deemed useful.

The satellite components of the system include the micro-controllers that will be used to measure the kilowatt per hour power usage at the outlet. The micro-controller will be designed to measure individual draw at each port instead of the outlet as a whole so a user can better diagnose which device is drawing which amount of power. It is still uncertain if some implementation can be accomplished

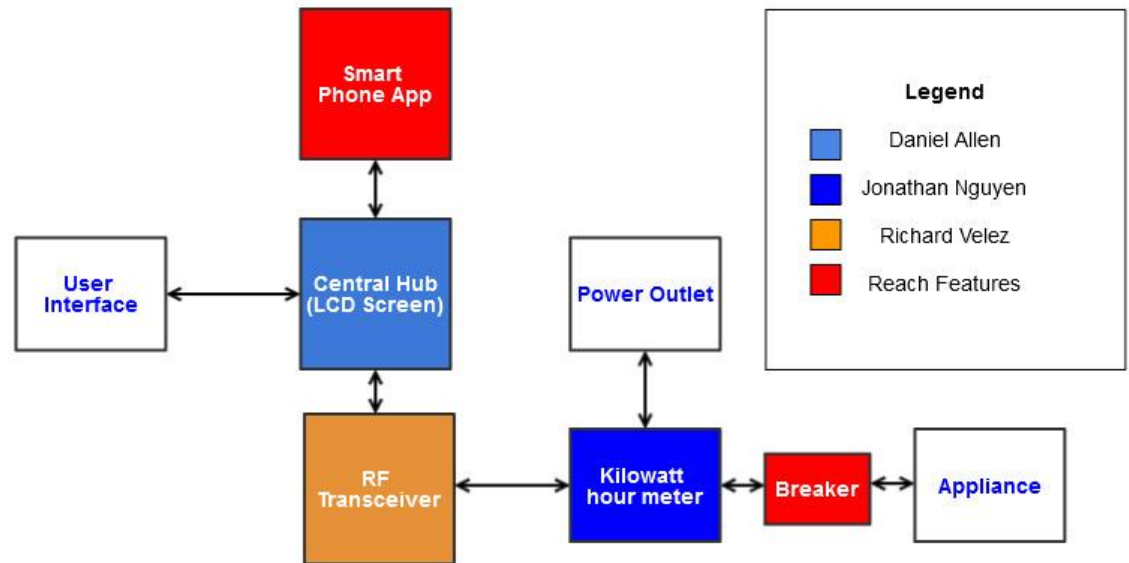
Which will allow reading across a device such as a surge protector? At the minimum the overall power draw from the surge protector that is plugged into a wall can be reported to the main unit depending on which outlet is drawing the power. Each micro-controller will use radio frequencies to send and receive information and the current method to be implemented is through the use of radio transceivers, however other possibilities may be considered depending on the available technology. The use of a transceiver is necessary because the micro-controller will sit behind an outlet out of sight from the user and transfer the information at request from the transceiver at the main unit. Each micro-controller will also have a simple liquid crystal display (LCD) soldered into the board that will display the kilowatt hour measurement. This will allow for debugging and verifying that the information being sent to the VGA display is indeed accurate.

Other options being considered include a remote controlled circuit breaker. The circuit breaker will be installed with the micro-controller at the individual outlets. The main device can be programmed by the user to allow it to turn off devices if certain thresholds are being crossed. As an example, a user can specify what devices are permitted to be turned off in the event that at current monthly bill calculations, the power usage is surpassing monthly expectations. The device will be intelligent to make the best guess at turning off the appropriate device or devices with as little interference as possible to lower the power consumption. The main unit can be designed to sound an alarm before a device is turned off, alerting the user that the event is about to occur without surprising them. In addition, the tripping of the circuit breaker can be canceled by the user and they can then proceed to turn off appropriate devices if they choose to.

In addition to the circuit breaker, an application is being considered that can run on certain mobile operating systems such as Android. This will allow the user to control the monitoring system from their handset instead of the VGA wall unit. The monitoring from the mobile unit will include the same features and will allow the user to keep track of power consumption as well as review past usage. The application will also be able to receive alerts if any device is going to trip the circuit breaker at a specific outlet, allowing the user to make adjustments or cancel the operation.

Referring to Figure 4.1.1, the block diagram shows a total of four subsystems that will make up this project. The first subsystem is the Video Graphics Array (VGA), the second

subsystem is the kilowatt hour meter, the third subsystem is the radio transceiver, and the reach goals are the circuit breaker and smart phone application. Each of these subsystems will be discussed in their respective sections.



**Figure 4.1.1: Block Diagram**

## 4.2 First Subsystem: Video Graphics Array

### 4.2.1 Video Graphics Array Controller Hardware Design

**Overview:** In order to interface with the user within this system a display needs to be used. In several instances a Video Graphics Array (VGA) display would be enough to meet these needs, however more has to be taken into consideration. Having evaluated the multitude of ways of receiving input from the user, a touch screen was determined to be the best option. A touch screen will allow the system to collect input from the user in addition to displaying a relevant image. Incorporating these two features into the same unit will eliminate other equipment used for input such as push buttons, keyboards, and mice. All of the aforementioned equipment can be rather bulky and add additional size to the embedded system design.

**Resistive Properties:** A resistive screen was selected over the option to use the more popular and responsive capacitive display because it is easier to integrate with a microcontroller in addition to being cheaper and easy to find in a barebones format. What is meant by barebones is that there is no surrounding housing or casing and contains only the necessary components to make the screen run. In its most basic form a touch screen is

a transducer which allows it to take the energy from the user touching the screen and convert it to electrical energy to be used by a microcontroller. As displayed in figure 4.2.2.1, the screen has resistive layers in both the X and Y directions. There is a polyester top sheet that has a conductive coating as well as the glass that protects the components. When the user touches the screen the resistive change will allow the position to be measured using the analog data sent over the pins (Mehta).

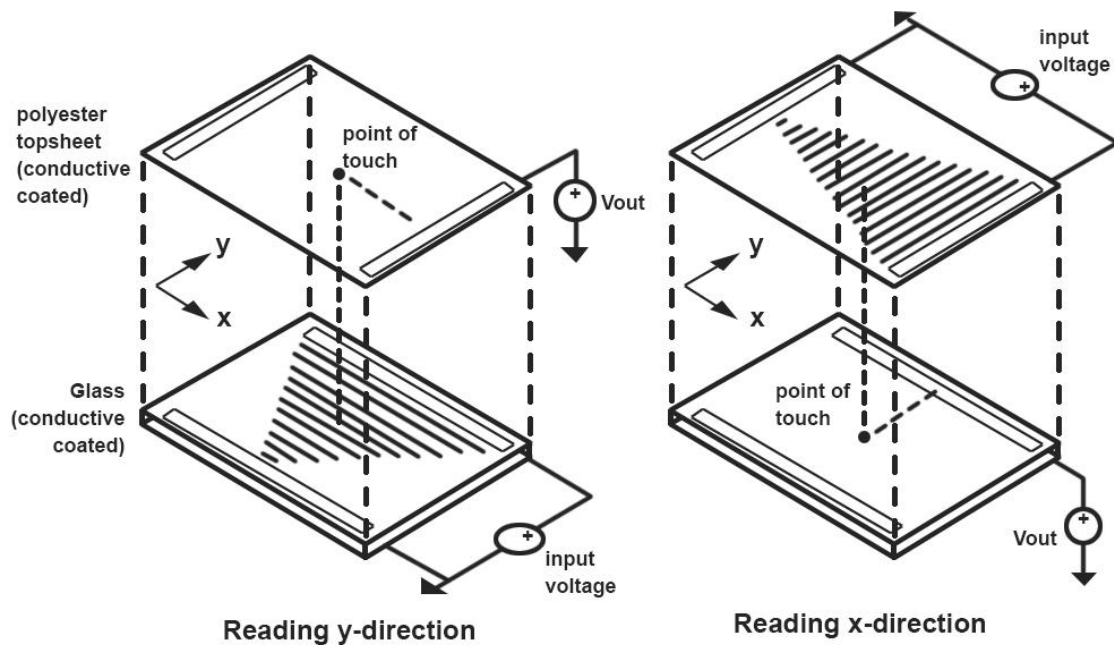


Figure 4.2.1.1 Courtesy Dhaval

The resistive touch screen can interface with a microcontroller directly if the microcontroller has a built in analog to digital converter or in combination with an external analog to digital converter that is connected to the appropriate pins on a separate microcontroller. The analog to digital converter needs to have at least two channels because of the way the output from the screen is configured. The later of the two options will be used even though the 40 pin Parallax P8X32A-D40 Propeller microcontroller that was selected does include a built in analog to digital converter on its pins, it would use two more pins than the TI one for feedback.

**Flat Flex Cable (FFC) Connection:** Shown in the bottom right of Figure 4.2.1.2 on either y or x axis diagrams is a connection known as a flat flex cable. In this instance the cable has four pins that are used for sending touch information to a microcontroller to be processed. Moving from left to right the pins are labeled 4, 3, 2, and 1. Pin 4 carries the X1 information shown in Figure 4.2.1.2, pin 3 carries the Y2 information, pin 2 carries the X2 information, and finally pin 1 carries the Y1 information.

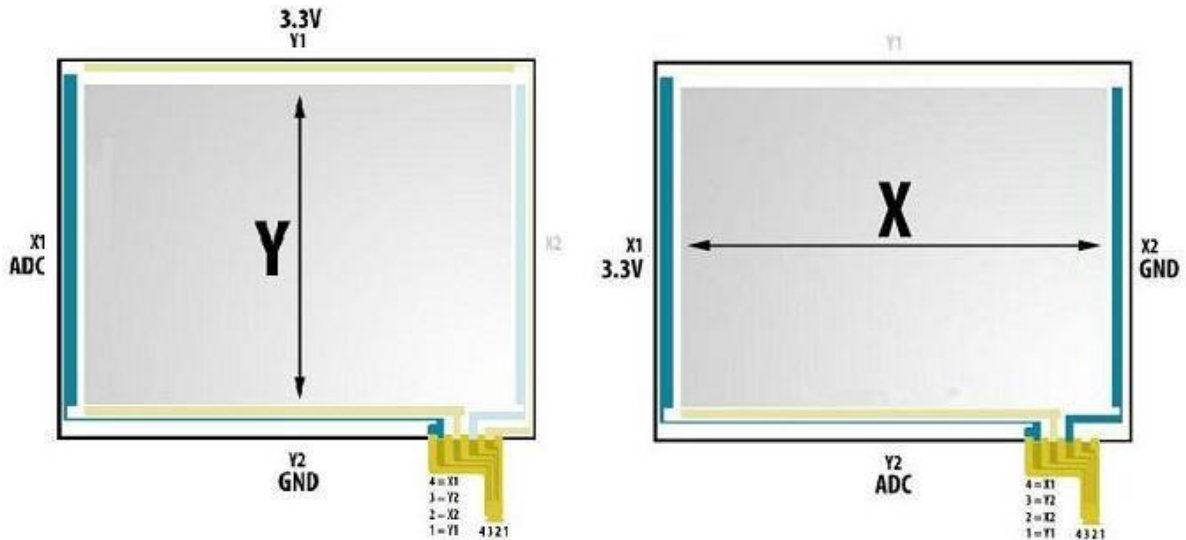


Figure 4.2.1.2 Resistive touch screen wiring and voltage Courtesy Dhaval

The X1 and Y2 pins will send analog information and will be required to be wired to an analog to digital converter (ADC). As was mentioned previously the microcontroller that was selected for use in the design does not have one built in so an additional Texas Instruments ADS1015IDGST will be used for converting the necessary analog signals. As can be seen with Figure 4.2.1.2 Y1 is wired to 3.3V as well as X1 and Y2 and X2 will be wired to ground. The wired connections are shown in Table 4.2.1 and show that logic high can be either +5V or +3.3V. The ADC being considered from Texas Instruments can operate between -0.3V to +5.5V so either configuration can be used depending on the screen.

Configurations				
	X1	X2	Y1	Y2
Measure Position X	Logic High (+5V or +3.3V)	Logic Low (GND)		ADC
Measure Position Y	ADC		Logic Low (GND)	Logic High (+5V or +3.3V)

Table 4.2.1 Configuration for the pin out of the flat flex cable to necessary components

The input/output pins that will receive the input from the screen on the microcontroller will need to be configured as general purpose I/O. The order of the data being read from the screen will matter as the X position needs to be read first followed by the Y position. As stated previous pins 4 and 3 (X1 and Y2) will be connected to the ADC and pins 2

and 1 (X2 and Y1) are to be connected to general purpose I/O pins on the microcontroller. The diagrams and selected pin connections to the microcontroller will be detailed later when presented (Mehta).

To read the X position from the touch screen the X1 pin needs to be programmed as logic high (+5V or +3.3V) and pin X2 needs to be set as logic low (GND). Given that the touch screen contains the resistive layer in both directions if a voltage of +5V and GND is applied to its pins it will create a voltage gradient in the X direction. Voltage on the X channel will vary according to where the screen was touched and the same happens with the Y channel. In order to measure the X position, Y2 needs to be configured with the ADC input and Y1 needs to be set to a high impedance state. Y1 on the microcontroller can be set to input and this will meet the requirement of high impedance and then the value from the ADC can be read. Given that the ADC has a 12 bit resolution the values on the scale will range from 0 to 4095. This value will vary depending on where the screen is touched. The next step is to read the Y position and for this to happen Y2 needs to be configured as logic high and Y2 as logic low in addition to X1 as ADC input and X2 as high impedance state.

**Analog to Digital Converter (ADC) Configuration (TI):** The ADS1015 shown on the right in Figure 4.2.1.3 is capable of four channels of analog input but only two will be needed to interface with the touch screens four pin FFC. As was stated previously, the resolution of this particular ADC is 12 bits which will allow for high precision measurements from 0 to 4095 which will be more than sufficient given that the 7" touch screen has a maximum resolution of 1920 x 1080 pixels. Analog In 0 (AIN0) and Analog In 1 (AIN1) will have pins X1 and Y2 respectively wired to the pins. To select AIN0 and AIN1 versus the bottom two inputs there is a three bit configuration register that can be used to select the appropriate input. Given that the voltage from the output of the touch screen is currently unknown and will need to be measured, resistors in series or an external Schottky clamp diode will need to be used to limit the input current to be within range of the specifications (Ultra-Small, Low-Power).

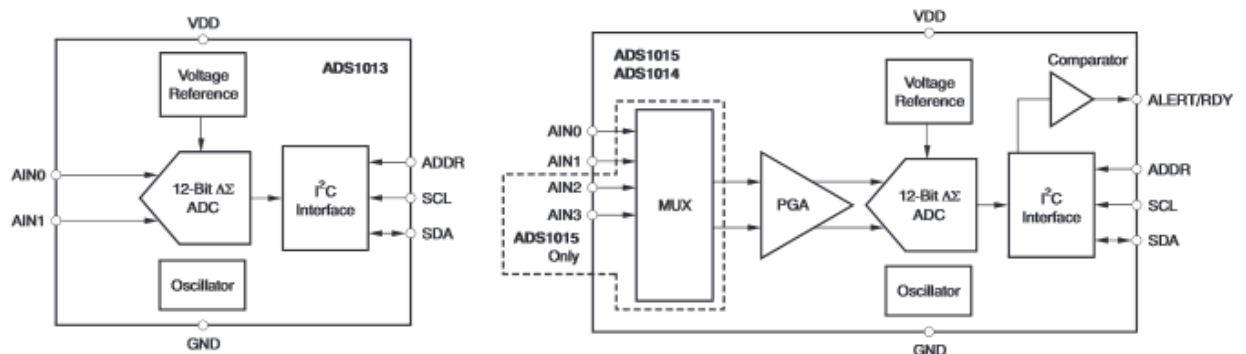


Figure 4.2.1.3 Texas Instruments ADS1015IDGST block diagram Courtesy Texas Instruments

One of the more important parts of the ADC is the reference voltage. These values will help determine the minimum and maximum values that will be sent to the output when a voltage is given on one of the channels. Depending on the maximum and minimum voltage measured from the screen with a voltmeter will determine what these reference voltages will be  $V_{ref-}$  shown in Figure 4.2.1.3 is the ground pin and  $V_{DD}$  will be tied to  $V_{ref+}$ . Based on the reference voltages, if an input voltage equals  $V_{ref-}$  then a 0 will be sent to SDA and if  $V_{ref+}$  is measured on the input then 4095 will be sent to SDA (Baker).

**Sigma Delta Architecture:** The Sigma Delta architecture is used to convert analog to digital signals. The sigma delta architecture uses a stream of ones and zeros from the interpreted voltage to represent it in a discrete manner. The more ones that are sent in the stream the more positive the signal is and the more zeros that are sent the more negative the value is. The closer the number of ones and zeros get in terms of quantity the closer to the value zero the signal is. A counter is used within the DAC during a specified clock time that will count the number of ones that pass through. As was discussed earlier, the greater number of ones that pass through in the period of time the greater the value is to the max  $v_{ref+}$  value it is (Baker).

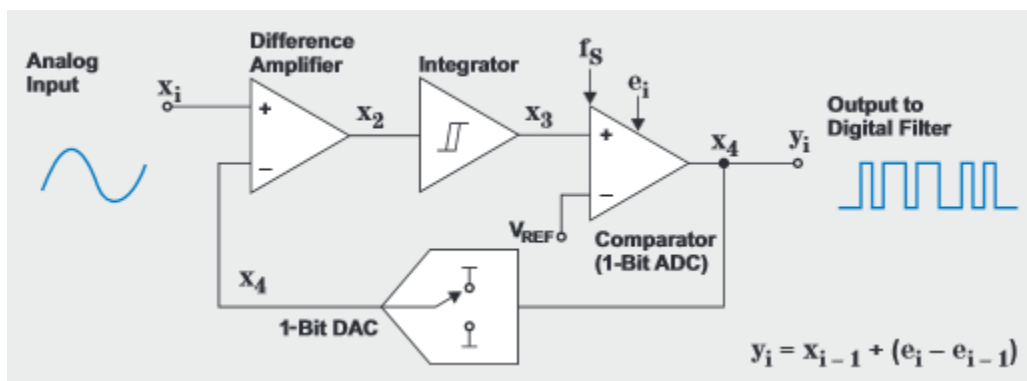


Figure 4.2.1.4: Sigma Delta ADC circuit Courtesy Texas Instruments

Figure 4.2.1.4 shows a typical sigma delta circuit that would be used within a ADC. The difference amplifier is the point in the circuit where a one will be added to it if there is a difference at  $x_2$  and will continuously do that as long as there is a difference. If there is not a difference then zeros will be used so that the counter will not be incremented.

**I2C and Bit-Banging:** Two wires make up the I2C line and they are called the clock line (SCL) and the data line (SDA). The clock line is used to synchronize data transfers over the bus while the data line send the data. These wires are connected to all devices on the I2C bus therefore eliminating the need of having two additional wires for every device added to the overall design. A third wire is used for grounding the circuit. In addition to the third wire another may be used to power all the devices on the bus, but in this design devices will receive power independently of existing on the bus. The SCL and SDA



wires are referred to as open drain drivers which means that any chip that is on the bus may drive the output low but cannot force it to go high. In order to go high a pair of pull up resistors would be used and again this would be two for the whole bus and not a pair for each individual component on the bus (Using the I2C Bus).

Looking at the right side of the diagram in Figure 4.2.1.3 it can be seen that data will be sent from the ADC by using an I2C interface. The I2C interface does not exist on the Parallax Propeller microcontroller to communicate directly. In order for the Parallax Propeller to receive data from the ADC it will be necessary to simulate the I2C protocol by bit-banging the data in software.

An example of this type of configuration is shown in Figure 4.2.1.5 in which the SCL line is connected to GPIO\_1 on the microcontroller and the SDA line is connected to GPIO\_0. The SDA line is the data line where the bits are sent over. The SCL line is the clock line that is used to synchronize all data transfers over the I2C bus. As can be seen in the same figure, there is a pull up resistor that is used to make sure the states are correct for data transfer. Bit-banging I2C with GPIO pins begins by setting the GPIO pin to 0 and toggling between input and output.

In order to drive the line low the pin is set to output and the opposite is true in that to make the line go high it is set to input. When the device is on input the data can be read and if another device is pulling the line low then the data is read as a 0 in the port input register. The microcontroller is allowed to leave the SCL line on output and set the 1 and 0 as appropriate without the use of a pull up resistor. This is possible because there is nothing that will drive the clock line low. A pull up resistor would be necessary if there are any connections made that can force the line to go low and thus the SCL line should be high-X or 0.

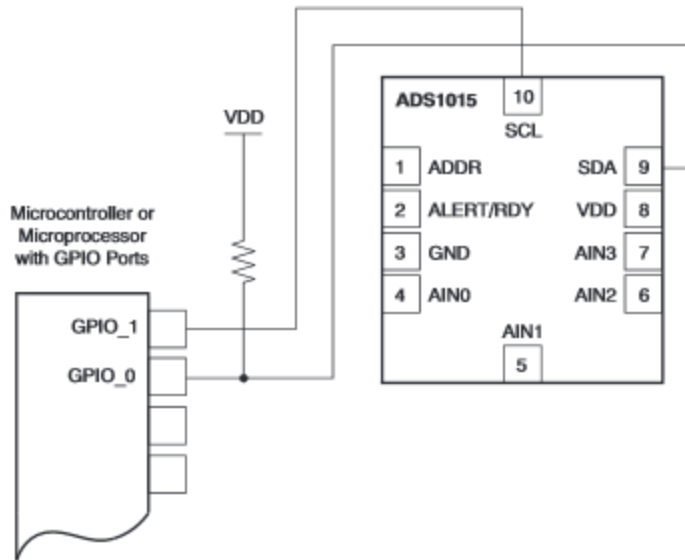


Figure 4.2.1.5: Bit-bang configuration for receiving data on GPIO pins on microcontroller from I2C interface  
Courtesy Texas Instruments

**Analog to Digital Converter (ADC) Configuration (Onboard):** It is possible for delta sigma analog to digital conversion on the Propeller Parallax using the counters available in each cog. The difference in this option is that it is easier to implement, but it does take an extra pin for each channel as the option involves positive edge detector with feedback. The feedback in the loop was introduced in Figure 4.2.1.4 and discussed during sigma delta conversion. Depending on the number of pins remaining after essential external components have been connected will determine the kind of analog to digital converter method used. Unlike the TI ADC, using the GPIO pins on the Propeller microcontroller allows for a resolution between 5 and 14 bits giving it two more bits of resolution.

The setup in terms of what is needed is different, but the overall effect is the same for what is required for the loop to work. In Figure 4.2.1.6 there is a 100 k $\Omega$  resistor between two general purpose I/O pins. This is what is used for feedback. Like what was shown in Figure 4.2.1.4, there has to be feedback into the difference amplifier for it to work. In addition to the resistor connecting to GPIO pins is the need for two 1 nF capacitors connected to VCC (+3.3V) and VSS (ground) as with the previous case. This difference will allow the conversion between the analog signal coming and the digital one being sent to the GPIO pins. A stipulation for using the capacitors is that they must be within 2.5 cm of the Propeller chip in order to work sufficiently. If they are any further than the noise that gets introduced into the circuit will become a problem. The asterisk component shaped like a hexagon is meant to distinguish between an analog to digital circuit or a digital to analog circuit. Of course in this case an analog signal is being read so a 0.1  $\mu$ F capacitor will be needed.

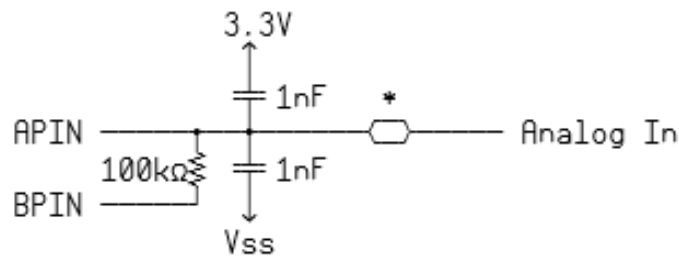


Figure 4.2.1.6: Sigma Delta ADC using GPIO on microcontroller with feedback Courtesy Parallax

**Visual Graphics Array Signal Specifications:** In order to better understand the hardware design and programming behind producing Visual Graphics Array (VGA) signals it is necessary to understand the specifications that have been put in place. As Figure 4.2.1.7 shows, the DE15 connector has 15 pins and three of them are no connect, meaning they are not used. Usually there is a fourth no connect on pin 15, but in this example it was used as a display data channel clock. Beginning with pin 13, Horizontal Sync is responsible for starting a new line on the screen. When the horizontal sync signal is given an electron beam will restart at the next screens scan line. The pulses on horizontal sync line will make sure the monitor stays in bound on the left and right of the screen. Vertical sync will make an electron beam restart at the first screens scan line which starts a new frame. Vertical sync is responsible for ensuring that the monitor displays lines between the top and bottom of the screen. These two synchronous pulses combined is what determines the screens resolution. Pins 1, 2, and 3 are R, G, B which represent red, green, and blue respectively. These three colors when combined in different combinations of intensity will comprise the colors seen on the screen. Horizontal and vertical sync comprise square wave pulses of +5V where RGB signals are analog and range from +0V to +0.7V. It should be apparent that the smaller the magnitude of voltage that is delivered on the pixel elements the darker the pixel will be (Sanchez).

Figure 4.2.1.7 is a schematic detailing the connection from a VGA DB15 connector being wired to the Parallax Propeller. The schematic shows a mouse and keyboard being wired from pins 3-6, but this will not be used in the design as there is no expectation to get input from those types of peripherals. This is also the reason the adapter board for VGA that Parallax has manufactured will not be purchased because it includes two PS/2 ports that will have no use. If for any reason they are included in the future it will be easy to adapt the design to include them.

As can be seen in the diagram five 240  $\Omega$  resistors will be needed as well as three 470  $\Omega$  resistors. There are two orientations on the Parallax Propeller that the VGA connector can be wired to. The first one uses pins P16 to P23 or as shown in Figure 4.2.1.7 pins P7

to P14. In either case it is a valid connection, but the important part is that they are aligned in order starting with the vertical sync pin and ending with the red pin or vice versa. The general purpose input/output pins on the Parallax controller can be accessed by any processor (cog) at any time which means that the configuration is not as important for where it is started as long as the order is kept.

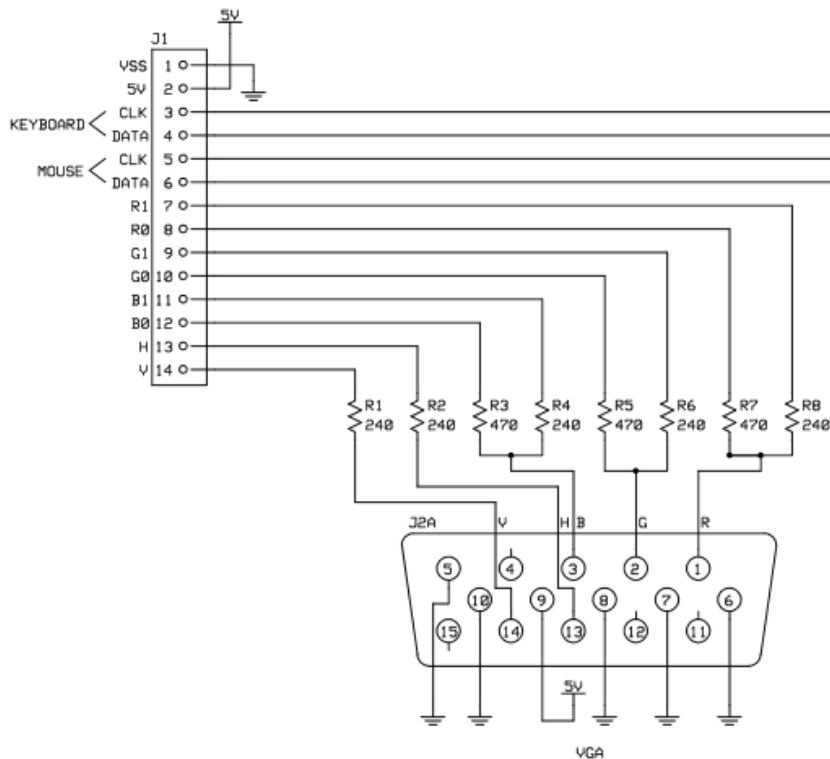


Figure 4.2.1.7: VGA connector as will be wired with micro controller Courtesy Parallax

Each color pin (RGB) on the VGA connector splits off into two different wires with different resistors that connect to the Parallax Propeller. As shown in Figure 4.2.1.7 as pin 1, which is red, the left connection has a resistor with a value of  $470\ \Omega$  and the right has a resistor with a value of  $240\ \Omega$ . These resistors are put in place to help with getting the colors correct on the VGA connection. The  $470\ \Omega$  resistor and wire is connected to pin 8 on the microcontroller and the  $240\ \Omega$  resistor and wire connected to pin 8. The same will be done with the green and blue connections, pin 2 and 3 respectively. The order will be R1, R0, G1, G0, B1, B0 and the final two are horizontal and vertical sync being connected to pins 13 and 14 respectively.

To take advantage of the libraries that Parallax has created for outputting images on the VGA connection it is necessary to connect the VGA connector in the manner shown in Figure 4.2.1.8. The pin numbers have change but the order of R1 to vertical sync has remained the same. The adapter makes use of the a PS/2 mouse and keyboard and therefore pins P24 - P27 would remain free for other connections because they will not be

used. It is more likely that Figure 4.2.1.8 will be used over the previous because the pins line up flush with the bottom of the controller and out of the way as well as giving the option to use included libraries created by Parallax.

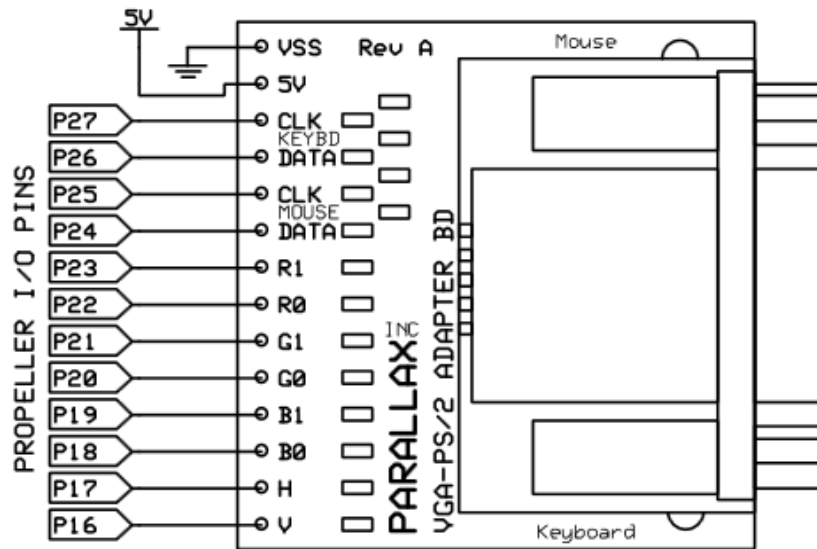


Figure 4.2.1.8: Alternate pin configuration to make use of Parallax libraries Courtesy Parallax

## Parallax Propeller P8X32A

**Overview:** The Propeller P8X32A is designed to provide high-speed processing power for embedded systems while providing low current consumption. The eight core design is simple enough to make learning and utilizing the chip easy. The two languages available for this processor include Spin which is a high level object oriented language and Propeller Assembly which gives users more intricate control over the processors and pins. In Figure 4.2.1.9 in the bottom right corner is a diagram that demonstrates that each cog (which is the name given to each processor) has access to each I/O pin on the chip and each has access to the counter so each processor can work in unity with the other seven. Given the orientation of the diagram it may be apparent that this system uses round robin scheduling from cog 0 to cog 7.

**32-bit Multi-core Architecture:** This chip has true parallel processing with eight symmetric processors. The built in run-time management allows for an event driven system rather than an interrupt driven system which simplifies the programming required to handle both asynchronous and synchronous events. It is because of this design that the chip allows for a more responsive and easily maintained application. The event driven aspect is especially useful with the VGA touch screen that will be used within the design of the system. It can be assumed that user input from the screen can happen at any time the system is running and with the event driven design with the eight processors, there is

no need to stop the system to check on it and then come back to the running program, it can just be handled separately (Propeller™ P8X32A Datasheet).

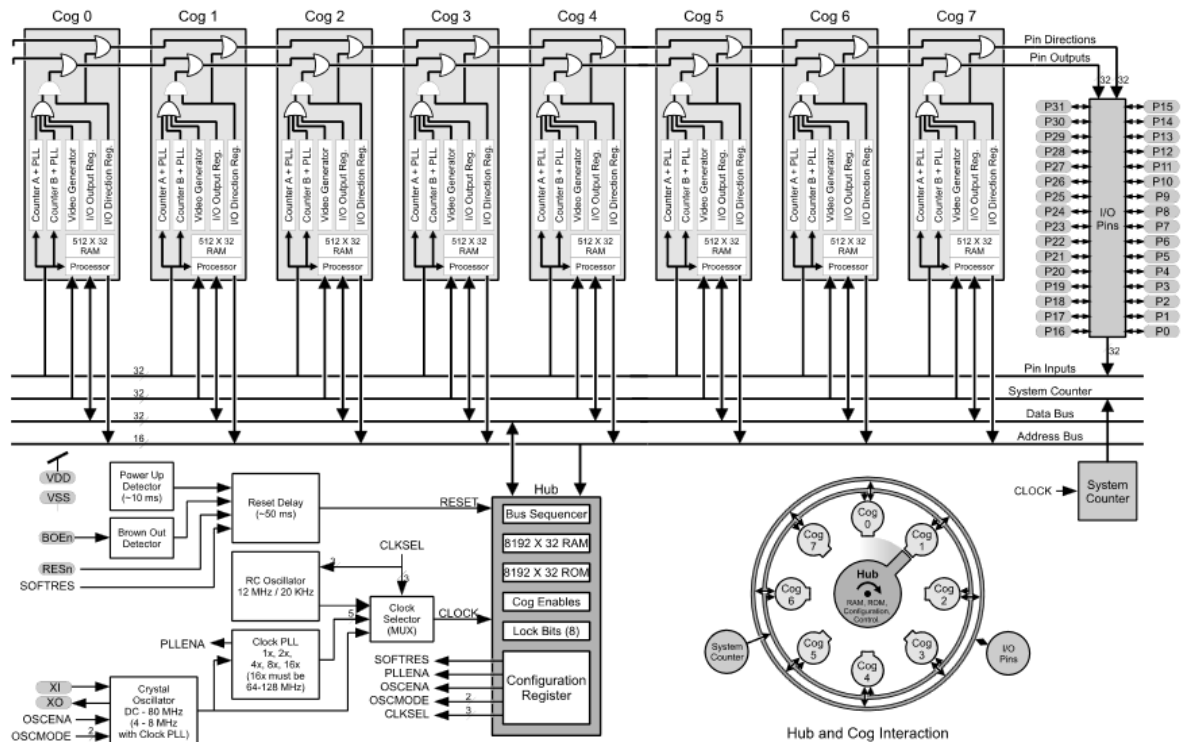


Figure 4.2.1.9: Parallax Propeller Block Diagram Courtesy Parallax

Figure 4.2.1.10 represents one of eight cogs within the Propeller microcontroller. Within each cog is a processor block located at the bottom left of the diagram and 2 kilobytes of random access memory (RAM) located just to the right (above) in a 512 x 32 bit configuration. Among one of the more unique and valuable features is the video generator capable of producing NTSC, PAL, and VGA signals. There are two advanced counter modules with Phase-Locked Loops (PLL) that is capable of maintaining a constant phase angle. Each cog also includes an I/O Output and Direction Register.

The same System Clock controls each of the eight cogs so that they all maintain the same time reference. In addition to sharing the same system clock each cog executes instructions simultaneously with the others as well as share the same resources. Each cog has the ability to be stopped and started at run time. All eight processors have general purpose RAM that excludes 16 special purpose registers.

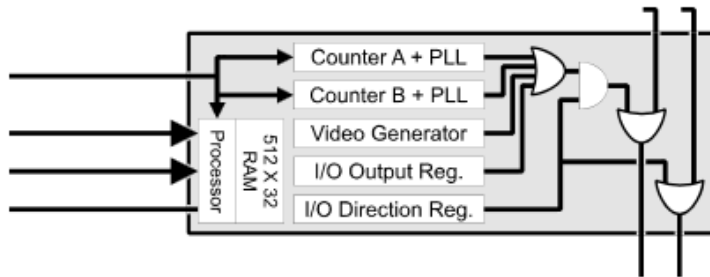


Figure 4.2.1.10: Individual Processor (Cog) within the Parallax Propeller Courtesy Parallax

### Pin Descriptions:

Pin	Direction	Description
P0 - P31	I/O	General Purpose  Special upon power-up/reset  P28 - I2C SCL connection to EEPROM  P29 - I2C SDA connection to EEPROM  P30 - Serial Tx to host  P31 Serial Rx from host
VDD		3.3 volt power
VSS		ground
BOEn		brown out enable (active low)
RESn	I/O	reset (active low)
XI	I	crystal input
XO	O	crystal output

Table 4.2.1.2: Pin descriptions of the Parallax P8X32A

**I/O and Peripheral Interface:** Aside from the special pins listed in Table 4.2.2 during power up and reset, all of the I/O pins including those special ones after the two mentioned events are general purpose I/O pins. The Propeller chip was designed so that it would take only a single instruction to access any individual or contiguous group of pins

on the board. An example of needing to access several pins at the same time would be for events like outputting a VGA signal to a monitor. There are 32 pins total that allow for great flexibility in adding a lot of devices to the chip without resorting to input expanders (Propeller™ P8X32A Datasheet).

The counters built in with the cogs and by extension are used with the pins have many advanced functions that can be very useful. Some especially useful features include its ability to convert analog inputs to digital. Other features include a configurable state machine that can sense repetitive signals, measure frequency, detect edges, count cycles and operate autonomously with optional run-time monitoring. There are two counters available in each cog and this can be seen in Figure 4.2.1.10 with counter A and counter B (Propeller™ P8X32A Datasheet).

Given that this 8 core microprocessor is very flexible and every cog can access any I/O pin on the board there are three rules that need to be followed when working with the I/O pins. The first one is that a pin is an input only if no active cog sets it to an output. In other words if a pin is an input any other cog (that is not disabled) cannot be relying on it to be output. Second, a pin outputs low only if all active cogs that set it to output also set it to low. Again, in this scenario all the processors need to be on the same page with themselves. Finally a pin outputs high if any active cog sets it to an output and also sets it high. This is an extension of the second one as for it to be low, all of the cogs had to be in agreement so by extension if one went to output and set it high all the others have to do the same thing ((Propeller™ P8X32A Datasheet).

**Video Generator:** Looking back at Figure 4.2.1.10 and Figure 4.2.1.9 it is possible to view a video generator available to each cog. This video generator is used to transmit video image data at a constant rate with the help of Counter A, also seen in Figure 4.2.1.10. Counter A must be running in PLL mode and is used to generate the necessary timing for the video signal. In addition to the counter used with video generation, there are two registers and one instruction which give control over the video generator on the cogs to the user. The first of the two registers is responsible for specifying the number of Counter A PLL clock cycles for each pixel as well as how many clock cycles to wait in-between frames and is known as the Video Scale Register. The second of the two registers is the Video Configuration Register and it is responsible for establishing the type of signal that will generated as either VGA or composite video which can run in either NTSC or PAL standards. The WAITVID instruction is the one used to interface with these two registers and counter A. To initialize the video generator the user should first start Counter A, then set the Video Scale Register, followed by setting the Video Configuration Register, and then finally providing the correct information to the WAITVID instruction. The cog will hang indefinitely if this procedure is not followed (Propeller™ P8X32A Datasheet).



**Available Packages:** There are three packages that are available with the Parallax P8X32A microprocessor. The Parallax Propeller microcontroller comes with two pin layouts that can be used in testing and production scenarios. Of the three types the first is the 44-Pin QFP chip which is useful for circuits that require a processor with a small footprint. The Q44 (QFP) chip uses a SMT design for placement on a board. The second type of chip has the same layout as the QFP chip and is of the QFN type which is the smallest chip that they make and also is used for production-level SMT-based circuits. Finally the 40 Pin DIP chip is useful for prototyping circuits with a breadboard where soldering would not be necessary. This is of course the most bulky chip, but is also the easiest to manipulate on a research level. All three versions of the chip have the same monetary cost and therefore selecting one really only requires knowing what needs to be done with it. In the most ideal situation the DIP would be purchased in conjunction with the QFP chip. The DIP could be used with prototyping as the manufacturer suggests and QFP while not the smallest would still save room on the board that will be fabricated while still providing an easier means of soldering it to the board.

**Serial Connection:** As has been discussed previously pins P28-P31 on the Propeller microcontroller have a special purpose when starting up or being reset. Two of those pins are necessary to interface with a serial port and those are P30 which transmits data and P31 which receives data from the serial port. There are two easy ways to communicate with the Propeller microcontroller from a pc and that is using a serial port shown in Figure 4.2.1.11 or the Propeller Clip designed by Parallax to interface with a Universal Serial Bus (USB) port on a pc. The serial port option will be used over the clip because the clip is a rather big piece of hardware to add to a board. Granted the DB9 connector is not small, but it can be mounted to the edge of the board and spans more in size in width than in depth unlike the clip which is the opposite.

Looking at figure 4.2.1.11 the RS-232 has nine pins that make up the DB9 connector. Pin 1 is the data carrier detect which will not be necessary in this instance as communication between the microcontroller and a pc does not require a data carrier. However the next four pins on the DB9 connector are required for the pc and the microcontroller to communicate with each other. The first of those four is pin 2 which is responsible for receiving data. In addition to receiving data, sending data will also be necessary and therefore pin 3 will be used as well. Next is pin 4 which determines if the data terminal is ready and this is connected with pin 2 the receive line with a 4.7 k $\Omega$  resistor to indicate that it is always ready. It is because of this implementation that this RS-232 implementation cannot be used for any other reason than programming the Propeller microcontroller. The transmit line on pin 3 will always be positive and never go to a negative state which indicates that it is idle. The DTR pin (pin 4) is held negative to provide a negative pull-down for the transmit transistor. Finally pin 5 which is ground is grounded to VSS (USB Serial, RS232).

On the opposite side at the microcontroller pin P30 which is the special purpose transmit data pin for RS232 serial communications is connected to pin 2 (receive data) on the DB9 connector as well as the Data Terminal Ready line. In addition to pin P30, pin P31 is the receive data line and is connected to pin 3 on the DB9 connector responsible for transmitting data. Finally the reset pin is connected to DTR on the DB9 pin. In addition to the pin connections between the microcontroller and the DB9 connector there are several other hardware components required to complete the connection. Four resistors are necessary as seen in Figure 4.2.1.11 and three of them have a 4.7 k $\Omega$  value and the other is 1 k $\Omega$ . Two more circuit elements will be used between the transmit and receive lines that require the implementation of a 2N3906 diode and a 2N3904 diode.

Finally on the reset line a 0.1  $\mu$ F capacitor will be used.

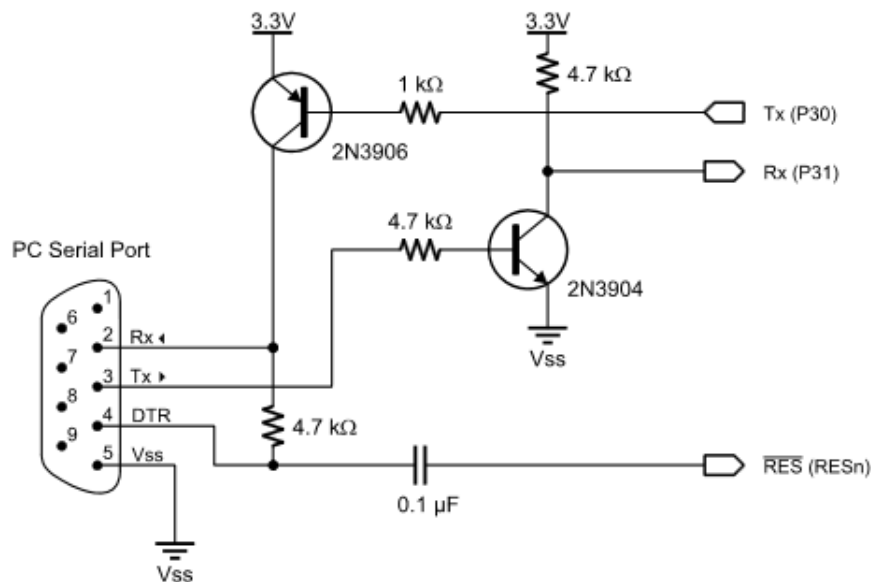


Figure 4.2.1.11: Serial connection for programming Parallax Courtesy Parallax

**Boot-Up Procedure:** The serial circuit and connection designed in the preceding section is to be used during the boot-up procedure. This procedure is run any time the Propeller is reset or powered on. The first step in the process starts the internal RC oscillator at 20 kHz and quickly switches to 12 MHz after a 50 ms reset delay. The first cog is tasked with loading and running the built in boot loader program.

The boot loader is an important program that will do most of the heavy lifting with getting a program loaded into a processors RAM. The first step among three performed in the order described begins with an attempt to detect communication from a host on pins P30 and P31. As described previously these are the pins responsible for sending and receiving data from a PC over the serial connection. If communication from a host is

detected an attempt to download a program into global RAM is made or optionally into an external 32 KB EEPROM.

It is not always necessary to download a program into the Propellers global RAM if an EEPROM is detected. The second step of the boot loader program (if the first fails) is to look for an external 32 KB EEPROM on pins P28 and P29 and if it is detected the entire 32 KB image is loaded into global RAM. Finally if neither of the two previous steps from the boot loader is successful the microcontroller has no choice but to terminate cog 0, move into shutdown mode and set all of the I/O pins to inputs which would put them at a state of high impedance. Alternatively one of the previous two steps was successful and the termination command is not given and cog 0 is loaded with the spin interpreter and the user code runs from global RAM.

**Electrically Erasable Programmable Read Only Memory (EEPROM):** An EEPROM will be necessary with the design of this embedded system as it will not be feasible to have a computer connected to the serial port of the Propeller microcontroller at all time to load a program on every start up. With the EEPROM the ability to store the program in nonvolatile memory will allow the program to be stored and ready to load on startup if for any reason the system is shutdown or restarted. In addition to storing the program needed to run the system, user data can be stored in the EEPROM allowing for a history feature in which the user can go back and look at previous months power usage.

As can be seen from Figure 4.2.1.12 the EEPROM is an optional component that can be added to the Propeller microcontroller and in this design will be necessary. The SDA and SCL lines show that it uses the I2C protocol to send data to and from the microcontroller from the EEPROM. Unlike the previously mentioned bit banging technique used with the analog to digital signal converter, this protocol can be directly followed in this instance, however the user will not be controlling anything at this point as it is all automatic. There will be no need to bit bang because for the short instance that the microcontroller is booting from a powered down state or a reset the pins take on a special purpose to transfer a program to one of the microcontrollers cogs and makes this transfer automatic though the boot loader.

There are two pins that need to be wired as shown in Figure 4.2.1.12 and those are pins P28 and P29 for the SCL and SDA lines respectively. In addition to wiring those pins, pins VSS, A0, A1, A2, and WP are wired to ground. VCC will of course receive the 3.3V that it needs from a voltage regulator that will be used to control the input voltage. Finally a 10 k $\Omega$  resistor is wired into the circuit between the input voltage and the SDA line.

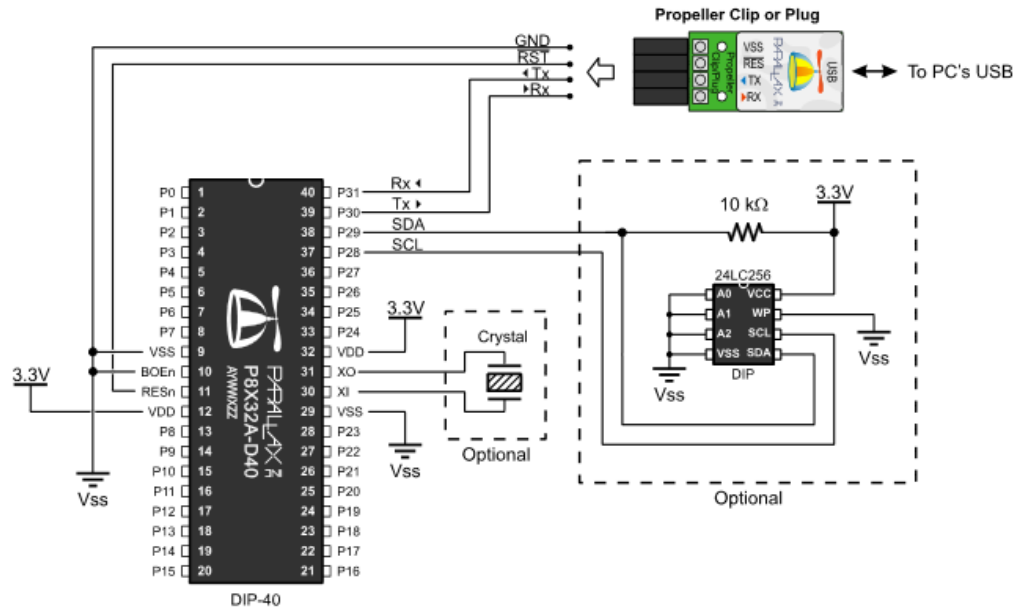


Figure 4.2.1.12: EEPROM connected to Parallax Propeller Courtesy Parallax

**Voltage Regulator:** The Parallax Propeller microcontroller can withstand a maximum supply voltage of 3.6V, but to be safe as seen in most of the figures throughout this section a voltage of 3.3V will be maintained. In order to accomplish this task, a voltage regulator that outputs a maximum voltage of 5V will be used and then a voltage divider to bring it to 3.3V will be used. The Texas Instruments TPS54231D is a switching regulator shown in Figure 4.2.1.13 that is capable of performing this task.

A switching regulator is used in the design because liner regulators while cheap are not near as efficient as a switching regulator. The reason for this is that a linear regulator measures the difference between the input and output voltage and releasing as heat what is not needed at the output pin. It is because of the way this works that in many circumstances that a linear regulator will use more power stepping down the voltage that it does delivering it to the microcontroller in this case. Considering this design is observe the careless use of electricity it would seem counterproductive to include such a design in the system. The design of this system will include a standard 12V AC to DC adapter that plugs into the board so the difference between 12V and 5V will be substantial to warrant the switching regulator versus a linear one (A beginner's guide to switching regulators).

Unlike a linear regulator a switching regulator works by taking only a small amount of energy from the input and moving it to the output. This is accomplished by using a switch (hence the name) to regulate the rate at which power is delivered to the circuit. By delivering power to the system in this fashion versus other methods, a relatively low amount of energy is wasted. In some instances the design of a switching regulator can be

as much as 85% more efficient than a linear regulator (A beginner's guide to switching regulators). While the design tends to be more complicated (referencing Figure 4.2.1.13) with more parts, the benefits of using one for designs that are energy conscious outweighs the complications.

An input voltage in the range of 7- 28V can be accepted on the input pin and given the correct configuration, output 5V on the PH pin. As can be seen in Figure 4.2.1.13, it is not enough to simply wire a voltage regulator to the circuit and expect it to do what needs to be done. Care must be taken to design a circuit that will perform in a way fitting with the design.

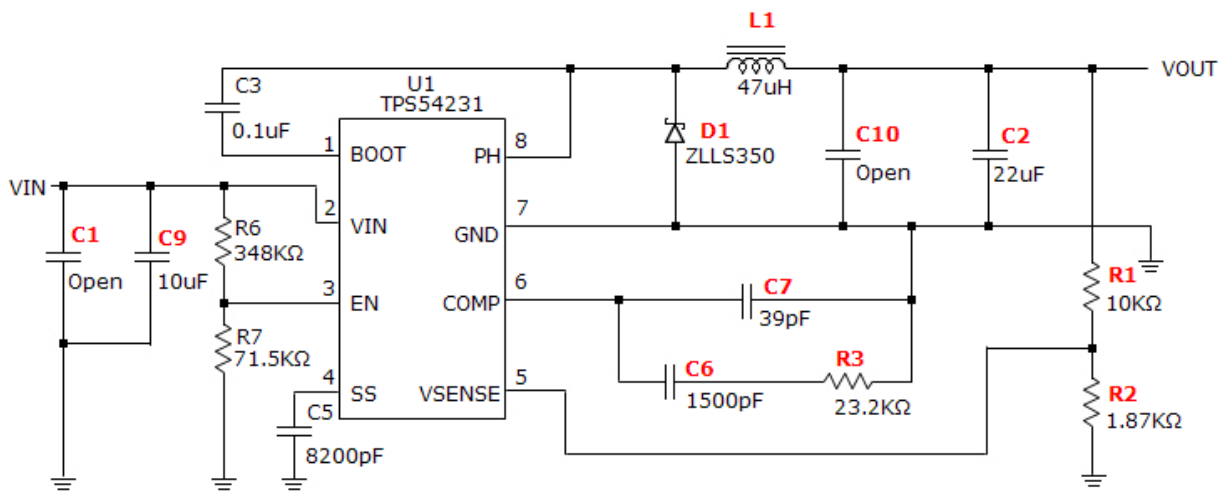


Figure 4.2.1.13: Voltage regulator with auxiliary connections Courtesy Texas Instruments

There are a total of eight pins with the package that are used for controlling the chip. The pin and descriptions are supplied in Table 4.2.3.

Pin	Description
boot	A 0.1 $\mu$ F capacitor is required between this pin and ph. Should the voltage fall below the minimum on this capacitor the unit will shut off until refreshed
vin	Input voltage from 3.5V to 28V
en	Enable pin. A voltage below 1.25V disables the device while floating enables it. The under voltage lockout is set using resistors Ren1 and Ren2

ss	Sets the output rise time using an external capacitor
vsense	Inverting node of the gm error amplifier
comp	Error amplifier output
gnd	Ground
ph	High-side power MOSFET source

**Table 4.2.1.3: Pin assignments of the TI switching regulator**

While the output voltage in Figure 4.2.1.13 is adjustable depending on the configuration and amount of resistors used, the output voltage will be controlled using resistor R1 and R2 at the right most part of the circuit. It is worth noting that the from the Parallax Propeller documentation the maximum current that can be sustained without damaging the microprocessor is 300mA and that is all being considered in the design of the switching voltage regulator. The following two equations are used to determine the values of the resistors given that an output voltage of 5V is desired:

$$1. R2 = \frac{R1 * V_{ref}}{V_{out} - V_{ref}}$$

$$2. V_{out} = V_{ref} * \left[ \frac{R1}{R2} + 1 \right]$$

With R1 selected to be 10 kΩ and a desired output voltage of 5V is known the value of the R2 resistor can be solved and that value ends up being around 1.87 kΩ. In addition to the two resistor divider network, the voltage regulator needs an input decoupling capacitors to work. In this case the typical recommended value for the decoupling capacitor is 10 μF and this has been shown to be successful in a wide variety of circuits. This capacitor should be of the type X7R and rated for a voltage that meets or exceeds the input voltage. The use of this type of capacitor is especially favorable if the voltage regulator is not located within 2 inches from the voltage source. It is fine to use smaller value capacitors at the input if the requirements are still meet. A second example would put two 4.7 μF capacitors in parallel with a 0.1 μF capacitor that is used for filtering out high frequencies. Under no circumstances is the maximum input voltage or current to ever be exceeded.

Two components are used for filtering the output and that is the inductor located at the top of the circuit in Figure 4.2.1.13 and the capacitor located just before  $V_{out}$  at C2. There are four equations that are used to determine the necessary inductor that should be used and those are included in Table 4.2.4.

1. $L_{min} = \frac{V_{OUT(max)} * (V_{IN(max)} - V_{OUT})}{V_{IN(MAX)} * K_{IND} * I_{OUT} * F_{SW}}$	2. $I_{LPP} = \frac{V_{OUT} * (V_{IN(MAX)} - V_{OUT})}{V_{IN(MAX)} * L_{OUT} * F_{SW} * 0.8}$
3. $I_{L(RMS)} = \sqrt{I_{OUT(MAX)}^2 + \frac{1}{12} * I_{LPP}^2}$	4. $I_{L(PK)} = I_{OUT(MAX)} + \frac{I_{LPP}}{2}$

**Table 4.2.1.4: equations used to calculate inductor and capacitor values at output**

After doing the necessary calculations the value that is required for the maximum 0.3 A and 5V output is 47  $\mu$ H and a CoilCraft PCH-27-473 will meet these requirements. In addition to the inductor as previously was stated is the need for a capacitor at C2. While a 22  $\mu$ F capacitor has been chosen to complete this job any value capacitor would work here, but due to the limitations of the voltage regulator, the maximum crossover frequency should be less than 1/5 the loop crossover frequency. The frequency the voltage regulator works at is 570 kHz so the desired maximum crossover frequency should be 25 kHz. With the necessary calculations completed a ceramic capacitor GRM32ER61C226KE20L from Murata Manufacturing will be used. As with the input the output capacitor needs to be rated for more than the output voltage plus half of the ripple voltage.

A mandatory capacitor is located at C3. This component is known as the bootstrap capacitor and resides between the PH and BOOT pins. A value of 0.1  $\mu$ F is used in addition to it being a higher quality ceramic capacitor of the X7R dielectric type like the ones used for the input. This type of capacitor is also used for temperature stability. In addition to the bootstrap capacitor there is a catch diode that can be seen between the PH and GND pins. The diode is required to meet the maximum ratings for the design. The reverse voltage must be  $V_{INMAX} + 0.5V$  at the PH pin in addition to peak current being  $I_{OUTMAX}$  plus half the peak to peak inductor current. A ZLLS350 manufactured by Diodes Inc will be used in this application which has break down voltage of 40V a forward current of 0.38A, and a forward voltage of 790mV.

The TPS54231 allows for a multitude of output configurations but the Type II compensation scheme is what is recommended by Texas Instruments. The Type II compensation scheme uses ceramic X5R dielectric capacitors. The characteristics found in this scheme include a dc gain component, a low frequency pole, and a mid-frequency zero / pole pair. In order to accomplish this a resistor must be put in series with a capacitor from COMP to ground as well as a capacitor in parallel with the resistor and capacitor in series.

There are a few steps that need to be followed to calculate the values of the compensation components that will be needed. The first step is to choose the closed loop crossover frequency which as discussed previously should be no greater than 25 kHz. In the next step the required gain and phase boost of the crossover network is calculated. When the equations have been manipulated and all values solved for, it is determined the value for the R3 resistor is 23.2 k $\Omega$  in series with a 1500 pF capacitor. In addition to these two elements is the C7 capacitor in parallel with both of the previous elements and the value of that is 39 pF.

**PCB Layout:** Referencing Figure 4.2.1.14, the  $V_{IN}$  pin should be bypassed to ground with a low equivalent series resistance (ESR) ceramic bypass capacitor. Given that it is an ESR and is ceramic the typical value for this type of capacitor is less than 0.015 $\Omega$ . Texas Instruments recommends that a ceramic bypass capacitor that measures 10  $\mu$ F ceramic with a X5R or X7R dielectric is used. The optimum placement in the design is near the  $V_{IN}$  pin and the source of the anode of the catch diode.

The ground pin GND should be tied to the PCB ground pin as can be seen with the two resistors and capacitor. Look at the catch diode the source of the low-side MOSFET should be connected to the topside ground area seen in the top left. This topside ground area is used to connect the input and output capacitors as well as the anode of the catch diode. The PH pin is connected to the cathode of the catch diode as well as the output inductor. Due to the switching nature of PH these pins should be located as close together as possible in addition to the area of the PCB conductor being minimized. To help dissipate heat the GND pin acts as a conductive path away from the die. Many vias can be used to route top side ground area to those layers internal or on the back side to aid in removing heat. The additional components should be placed as seen in Figure 4.2.1.14 in order to obtain acceptable performance.



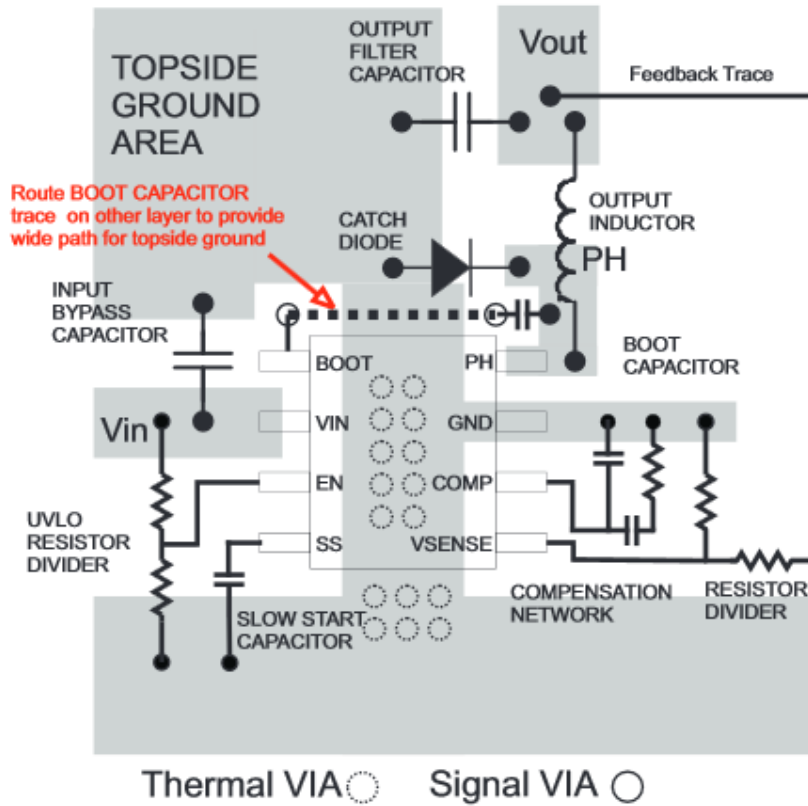


Figure 4.2.1.14: PCB Layout of Voltage Regulator Courtesy Texas Instruments

**3.3 V Voltage Divider:** The use of the switching regulator will not be enough for the design being implemented. The 5 volts that will be used to power pin 9 on the VGA DE 15 connector cannot be used to power the remaining components on the board without destroying them. Figure 4.2.1.15 shows an extension of Figure 4.2.1.13 in which two resistors will be added on  $V_{OUT}$  to get the desired 3.3 volts. The voltage divider equation of  $V_{OUT} = \frac{V_{IN} \cdot R_2}{R_1 + R_2}$  with the choice of using a 3 k $\Omega$  resistor will yield a result in which a 1.5454 k $\Omega$  resistor will be needed.

With the voltage divider in place on the output, two different voltages can be used on the board with the need to only one switching regulator. The 3.3 volts will be maintained indirectly by the switching regulator maintaining the 5 volts properly. While the resistor values have been chosen to get the desired value of 3.3 volts, it may not be the ultimate values that are used. The availability of resistors that can maintain a tolerance of  $\pm 0.01\%$  or less is minimal. Many companies have shown that bulk orders of these desired values and tolerances are required, which will not fall within budget of this project. With this revelation, alternative values can be used that will still suffice on a slightly higher resistance. With a higher resistance and a worse tolerance, the circuit will receive at maximum a value of 3.3 volts but on average will receive something less. This is fine

because the voltage does not have to be exactly 3.3 volts but it cannot go much beyond that value without damage. Much consideration will be made when designing these values and voltmeters will be used to make sure these values are not being exceeded before the rest of the components are soldered to the board and wired to the circuit. Failure to be cautious will result in irreversible damage to the system that cannot be repaired.

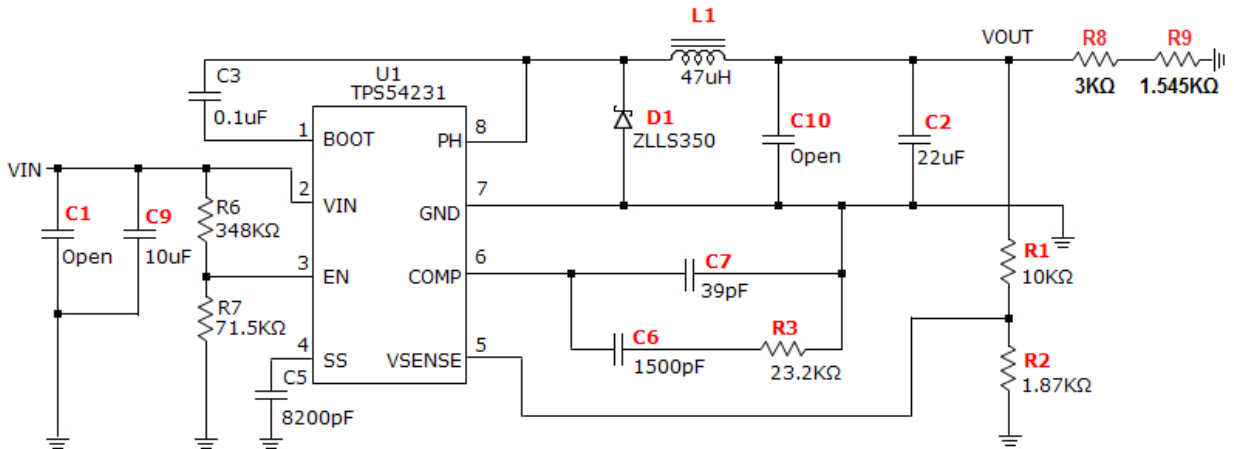


Figure 4.2.1.15: Texas Instruments

**DC Connector:** In the previous section the circuit used to control the incoming input voltage and current was designed. In order to actually deliver power to the circuit a 2.5mm barrel power connector will be used. This design uses a through hole connector versus any SMT/SMD option. In addition to connecting the power connector through the printed circuit board, it will be oriented at a right angle near the edge of the board. The right angle design is fairly common and will work well here as well. The connector is rated to handle 8A and 16VDC which is well above what will be expected of it. Under extreme circumstances the connector can withstand 500VDC for roughly one minute (*SwitchCraft*).

Observing Figure 4.2.1.16 it can be seen how the DC connector will be affixed to the printed circuit board. There are four contacts total that will be mounted to the board but terminal 1 and terminal two are the conductors. Terminal 1 which is a brass material with silver plating is 1 by 3 mm in dimension and terminal 2 is slightly smaller at 1 by 2.3mm in dimension and is made of a copper alloy and silver plated. The other two contacts that are not involved with conducting electricity have the same dimensions of 1 by 2.5 mm. There is a round plastic mount just above terminal 2 that helps stabilize the connection to the board and that measures 1.6mm in diameter. All of these measurements including where they reside in relation to each other will have to be accounted for when designing the printed circuit board for the connector to fit properly.

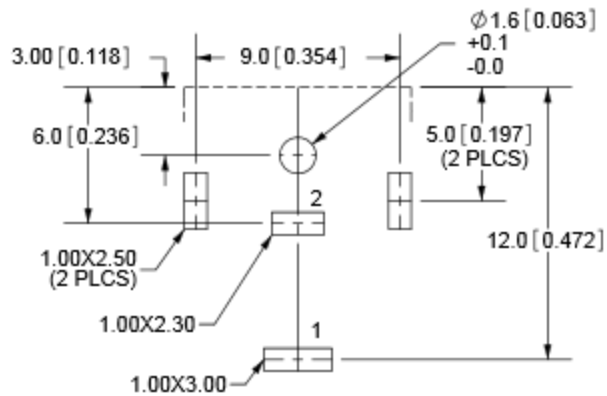


Figure 4.2.1.16: Top view of DC connector layout on PCB

In addition to the external contacts and conductors that are on the outside of the power connector there is a 2.5mm copper pin that an AC adapter will be plugged into. The selected AC adapter is an Emerson / Astec Power DA12-120US-M plug that conforms to specifications. The adapter is capable of taking an AC current and delivering 12V and 1A to the power connector. The most important dimension of this plug is that it is 2.5mm in diameter and will fit with the dc power connector. This power connector can receive an input voltage of 90VAC to 264VAC which is within range of any US system.

**Wireless Transceiver:** The final element to be added to the design of this board is the wireless transceiver. This module will be used for receiving data from the outlets about the amount of power that is being consumed. In addition to receiving data about power consumption will be the ability for the unit to send a signal to a specific outlet to trip a remote circuit breaker that will prevent it from using any more power. This is an attempt to prevent a user from exceeding their set power consumption for any given month.

The type of transceiver in consideration is the Linx TRM-900-NT. It is a long range, low cost, low power consumption, transparent data transfer module that will make sending and receiving data seem effortless. The power specifications of this transceiver do leave it in good standing with what will be done at the switching regulator. The operating voltage of this device is between a minimum of 2.5 VDC and a maximum of 5.5 VDC. The notes within the documentation measure the device's current ratings at a voltage of 3.3 VDC which works in favor of the design because the Parallax Propeller design requires an input voltage of 3.3 VDC. The typical supply current at 12.5 dBm is 36 mA with a max of 40 mA. At 0 dBm the unit uses a typical consumption of 16 mA and a maximum of 21 mA. Exceeding the voltage ratings can possibly lead to permanent damage to the device and even extended operation at the maximums will lead to premature failure. With the device operating at 3.3VDC it can be expected that the device will last close to the mean time before failure measurement from the manufacturer (NT Series Transceiver Module).

The pins of concern to the Parallax Propeller microcontroller are READY and RSSI that are monitored by the microcontroller and T/R\_SEL, POWER\_DOWN and STANDBY are controlled by it. In addition to those pins are DATA\_IN and DATA\_OUT that are used by the GPIO pins on the microcontroller that generates and decodes over-the-air data. There are other pins to consider when setting up the wireless transceiver but these will be detailed more depth in a later section. In the figure below (Figure 4.2.1.17) it can easily be seen which pins from the transceiver will interface with the microcontroller.

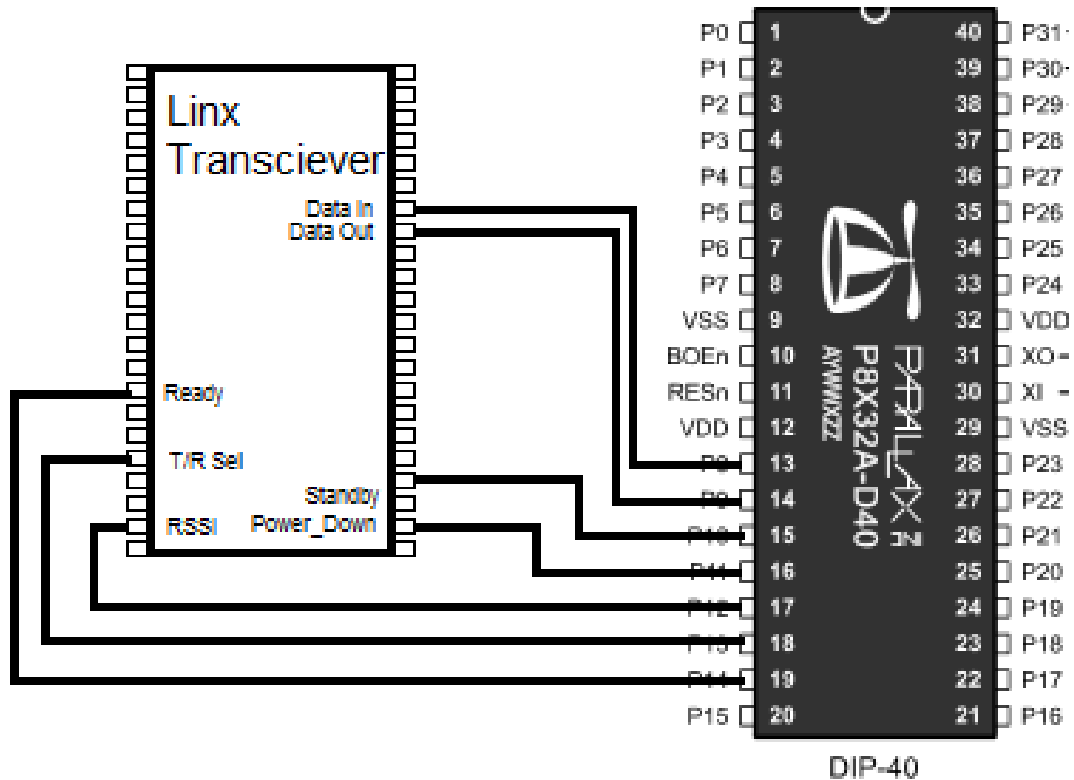


Figure 4.2.1.17: Link Transceiver connected to microcontroller

Along with setting up the pins is the necessity to meet power supply requirements and carefully plan where the unit will sit on the board. The manufacturer (Linx) recommends using a 10  $\Omega$  resistor in series with the power supply as well as a 10  $\mu$ F capacitor with ground in cases where the quality of supply from the power is poor. If these components are necessary they will need to be added close to the modules supply lines. Given the design of the power supply on this board, this probably will not be a problem but it will need to be tested. A part of the problem is that this device should be isolated from as many components on the board as possible. This does include crystal oscillators that can

be found on the Parallax Propeller microcontroller and switching power supplies. In addition to these guidelines, PCB traces should not be routed under the module that is on the same layer. If there are traces or vias under the module they could come in contact with the device and short the circuit board (NT Series Transceiver Module).

**Overall Design:** The schematic shown in Figure 4.2.1.18 has been rotated to the right by 90 degrees to show more detail. All references to the figure will be made in a fashion in which the figure has been rotated back to the left by 90 degrees. Starting from the top right is the DC connector that will be used to deliver power to the system. The wiring leads to two different voltage regulators for the system. Moving to the right is the Texas Instruments TPS54231D which will drop the DC current down to 5V. This is a switching regulator versus a linear one and will be used for delivering power to the VGA DE-15 connector on pin 9 and the voltage divider after  $V_{OUT}$  will deliver power to the serial port, the Parallax Propeller, the Linx Transceiver and the EEPROM.

From the DC connector again and past the switching regulator on the far right is the DB-9 connector that will be used for programming the microcontroller. As has been stated this is an automatic process that begins when the device is powered on and connected to a computer. The other reason for this port involves debugging. While programming the device and needing to observe the program flow the output will be sent to the Parallax Serial Terminal for observation. It is also the only way to send input to the program from the computer without using a VGA display.

Below the DC connector is the VGA connector that will be used displaying information to the screen. This port has been wired to the left side of the Parallax microcontroller. The eight resistors seen in between the two connections are used for controlling the red, blue, and green intensities for controlling colors. In addition to the VGA port is the flat flex cable connector seen beneath it. This port has four pins as the flat flex cable from the monitor has only four pins. As can be seen two ports are directly wired to the microcontroller. The other two ports contain two capacitors and a resistor and each take up two ports on the microcontroller instead of the one to one connection on the previous ports. This is used for the analog to digital conversion that will be needed on these pins for the proper interpretation of a user pressing the touch screen.

Between the flat flex cable connector and the microcontroller is the wireless transceiver. Wired to the microcontroller is the data in and data out lines that will use a serial like connection for sending data. Other connections made include the T/R line which tells the transceiver whether to send or receive, the RSSI, and the Ready line. Finally all of the ports that needed to be grounded have been and the ports that needed a 3.3V source have been given the power from the TPS54231D regulator with the additional voltage divider circuit.

The microcontroller has been placed just to the right of the wireless transceiver the appropriate ground and voltage connections have been made. All the connections from the VGA port, the flat flex cable, the serial port, and wireless transceiver terminate here. Everything that will be controlled from the board will be done through this device. While the devices are not shown, consideration is being made with the use of a few remaining ports on the right side of the microcontroller being available of adding an led light that will help with debugging when the microcontroller is not connected to a computer. In addition to the LED would be a reset button that when pressed will reset the device.

Finally the EEPROM is connected to the right of the Parallax microcontroller. This device also required the 3.3V from the TPS54231D switching voltage regulator with the additional voltage divider wired onto the end. All of the right pins have been grounded to VSS and the left pins connected to the SDA and the SCL lines on the microcontroller. This device is what will store any long term information such as the program itself as the memory is non-volatile.



Figure 4.2.1.18: Schematic showing all parts and connections

**Parts List:** Table 4.2.1.4 shows the complete parts list of everything that will be needed to build the board and control the main system.

Part	Quantity
TE Connectivity FFC 84981-4	1
100k $\Omega$	2
1nF	4
240 $\Omega$	5
470 $\Omega$	3
DE-15 connector	1
P8X32A-Q44 microcontroller	1
DB-9 connector	1
1k $\Omega$	1
4.7k $\Omega$	3
0.1 $\mu$ F	2
2N3906 diode	1
2N3904 diode	1
24LC256 EEPROM	1
TI TPS54231D	1
10 $\mu$ F	1
348k $\Omega$	1
71.5k $\Omega$	1
8200pF	1
1500pF	1
23.2 k $\Omega$	1
39pF	1
ZLL5350 diode	1
47 $\mu$ H	1
22 $\mu$ F	1
10k $\Omega$	1
1.87k $\Omega$	1
3k $\Omega$	1
1.545k $\Omega$	1
RASM732X DC connector	1
Linx TRM-900 NT transceiver	1
DIP switch	1
10 $\Omega$	1
10 $\mu$ F	1

Table 4.2.1.4: Complete parts list for the VGA controller / main system board

**Fabrication:** After the board schematic has been designed, the wires have been routed properly, and everything has been verified by the appropriate programs, in particular



DipTrace, the designs will need to be sent to be fabricated. The company that has been selected to accomplish this task is PCB Fab Express. This company has been around since 2003 and is located in Silicon Valley with over 7000 registered PCB designers and engineers that use their service which provides a lot of confidence when using their service.

This company provides many options when designing a board. To begin with their printed circuit board capabilities they offer 2 to 6 layers for boards that are not custom quoted. This option provides users with a cheap way to prototype boards with quantities needing to be under 50. This is fine in this design's case as only one will be needed. There are different turn times depending on what will be needed and these can range from three days, five days or ten days. Boards with these promotional offers are 10 square inches in area and the price depends on the amount of days. Three days will be the most expensive and ten days will be the cheapest.

Without custom ordering a board and using the promotional offers a few requirements must be met within the design. First the thickness must be 0.062 inches. The minimum trace distance must be at least 0.006 inches but 0.005 inches may be used at a small additional charge. While ten square inches is allowed a maximum X or Y dimensional cannot be longer than 15.9 inches. A green LPI mask will be given with the option to select a white mask on the component side only. This company uses hot air solder leveling to finish the board and individual routing is expected.

Considering this board will have surface mount devices there is a requirement that they be a minimum of 0.008" between SMT pads to create proper soldermask dams. An annular ring must be a minimum of 0.005 inches meaning that the pad diameter has to be 0.010 inches more than the hole diameter. No gold, cutouts or slots are allowed on this board. The distance from the edge of the board to a copper feature must be 0.01 inches. Fortunately there is no minimum quantity that must be ordered and they will handle just one board if necessary.

#### **4.2.2 VGA Software Overview**

There are many components that will be considered when designing the VGA controller. The first of these components is the software that will be used to design the software that will be loaded onto the board. The Parallax Propeller Tool as it is known will be the integrated development environment that is used. This software is provided by Parallax and is capable of producing the SPIN or PASM files needed to run on the microcontroller that has been used.

In addition to the main tool that will be used is the serial terminal software that has been provided by the same company. This tool will allow communication between a personal computer and the microcontroller to aid in development with debugging capabilities. Messages can be sent between both devices under the discretion of the programmer to get a better understanding of what is going on during software execution to potentially find problems.

Other systems that need to be considered in the software design is the execution on first startup and the execution of the program while it sits idle at the main screen waiting for input to process. During first time startup which will be checked every time the device starts up after being completely powered off the user will be prompted to add outlets to the system that can be checked for by the VGA controller. If any problems occur with communicating with the outlets then these can be alerted to the user so that they may take action to resolve the issue.

The second system that is constantly running when the program has made it to the main screen is the one that checks for user input or information being received from the Linx transceiver module. If user input is detected the appropriate procedures will be run to determine if it is valid and then move to the correct segment of the program as the user would expect and display the correct information to the VGA display. The other is meant to check the input buffer from the transceiver module and interpret any data that may be stored there. Examples of data include a response from one of the outlets should the main program make a request to communicate with it, but more regularly will be the measurement taken that will report back how much power is being consumed on that outlet. These measurements will constantly be taken and reported very frequently.

One of the benefits of the microprocessor is that these processes will never have to contend for the processor as it contains eight cores. This means that there is relatively no lag time between a user or perhaps the transceiver requesting processor time and getting it. Parallax has made sure that the Propeller will always allot a free processor to whatever task is requesting it and given that there are eight of them working together, having the resources to do so should never be an issue.

### **4.3 Second Subsystem: Kilowatt per Hour Meter**

The second subsystem consists of the Kilowatt per Hour Meter. This device is used to measure the amount of power consumed per hour in Kilowatts. Although each house is already equipped with this meter, its only function is to read in the overall power consumption of the house and is generally only accessible by the power company. The kilowatt per hour meter, or KWH meter for short, that will be designed will be catered

more towards the user. Its main function will be reading individual outlets and then transmitting this information to a central hub where the data will be accumulated so that the user may view the many different outlet consumptions as well as the overall consumption. By allowing the user to have access to this information, steps can actively be taken to reduce the overall power consumption in the household.

#### **4.3.1 Kilowatt per Hour Meter Hardware Design**

There are many different designs that already exist for the hardware component of the KWH meter. As mentioned earlier in section 3.1.1, the P3 International P4400 Kill A Watt is a portable KWH meter but it lacks the functionality to transmit this data to any source other than itself.

For any basic electro mechanical KWH meter, the power consumption is read through an electric induction motor that powers a series of gears connected to the meter's face. There consists of two electromagnets and a metal disk that rotates based on the amount of current coming from the house. The rotation is then controlled by the current feeding in from the power lines. The meter records how many full rotation the disk makes, which in turn determines a household's power consumption.

The unit being designed will have the basic functions required to read power consumption as well as being able to wirelessly transmit that data to the VGA Controller. This unit's design will be based on the Electronic KWH meters. These meters are more accurate and reliable than their electromechanical counterparts. "The electronic KWH meters utilize solid-state circuits that produce electrical signals whose frequency or strength relies on the voltage and current being used." (Electric Meters)

The schematic for the design can be viewed at the end of the hardware section. Figure 4.3.3 shows all the connections of the components being discussed below.

**Microcontroller:** The purpose of microcontrollers is to function as a mini computer and their general use is for situations when automatic control is desired. For this unit, the microcontroller will be responsible for taking in the voltage and current draw and using this information to calculate the power consumed in kilowatts per hour. From there it will transmit the data wirelessly to the main unit.

One of the microcontrollers considered consist of some from the MSP430xxxx family primarily due to its relatively cheap cost as well as the vastness of option for functions. Also, the MSP430 families required power source are relatively inexpensive to create. After researching through this family of MSP430, the microcontrollers selected will be the MSP430F6638 and the MSP430AFE. An example of how these two microcontrollers will interact with each other can be seen in Figure 4.3.1. This block diagram (courtesy of

Texas Instrument) shows the layout of an electronic kilowatt hour meter. The main medium of communication for any microcontroller involved will be via Universal Asynchronous Receiver/Transmitter or UART for short. The design used will be very similar to this block diagram. Although the idea of using two microcontrollers raises some flags.

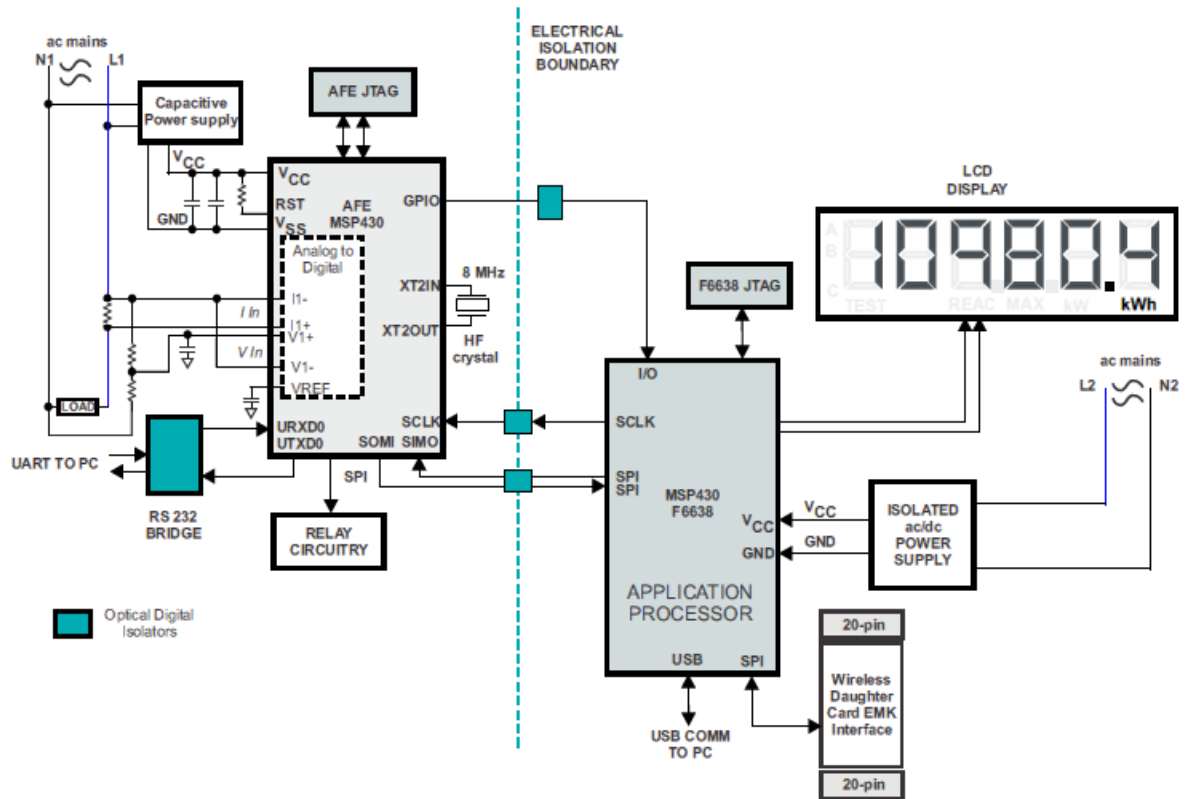


Figure 4.3.1: Kilowatt Hour Meter Block Diagram (Courtesy of Texas Instruments)

Some other mentionable microcontrollers were the ATMEGA32 and the ATMEGA328. There were plenty of application notes that went along with the project idea as well. But due to simplicity, working knowledge, and for consistency; the microcontroller chosen was the Parallax Propeller P8X32A, which is the same microcontroller that will be used for the VGA display.

As mentioned, the reason for using the Parallax for this power board is for the continuity. With research already done on this particular microcontroller, less time will be used in researching and more time will be used for implementing and troubleshooting a single design for both subsystems.

**Power Supply:** In order to power the kilowatt hour meter board, we need to take into account that there needs to be a separate source of power aside from the power coming through the outlet. The design was split into two halves: one half where the high voltage components will be contained and then a second part where the second power supply will

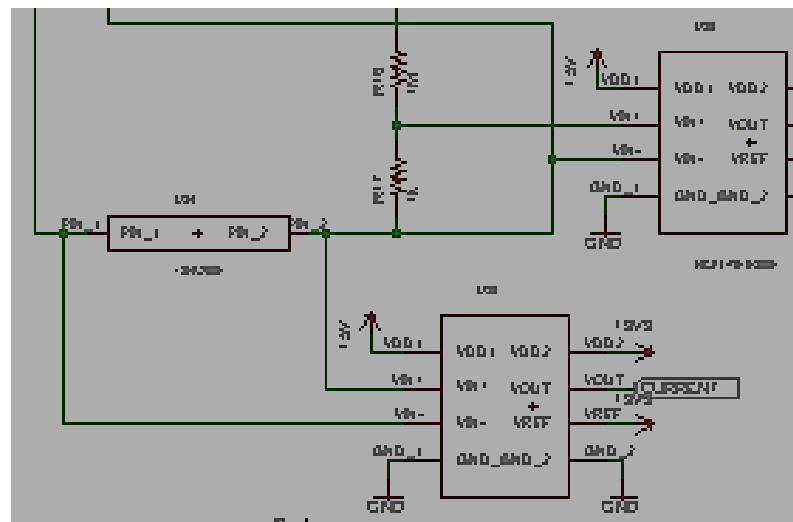
be used to power it. The dangerous side will have the  $170V_{rms}$  that feeds from the wall outlet through the circuit to the voltage and current sensors while the safe side will be in charge of other pins such as  $V_{cc}$ . After careful research, it was discovered that all the components needed either 3.3V or 5V to power them, therefore the power supply chosen was a 12V 2A AC/DC adapter. The use of voltage regulators was implanted in order to step down this 12V power supply to the necessary 3.3V or 5V depending on the component.

**Current and Voltage Inputs:** In order to measure the current and voltage inputs, the use of two isolation amplifiers will be implemented. The specific component chosen for this design was the HCPL-7800. These isolation amplifiers ignore high Common Mode Slew Rates of at least  $10kV/ms$  and they contain their own Analog to Digital convertors.

The voltage from the main lines read in at roughly  $170V_{rms}$ . This voltage is too high to directly feed into any of the components, therefore a voltage divider of ratio 1001:1 will be used before the voltage reaches our isolation amplifiers. The resistor values chosen were 1Megaohm and 1kiloohm, thus reducing the AC voltage from 170 to roughly  $0.17V_{pp}$ .

The first isolation amplifier will be in charge of reading in the voltage value from the AC line. This will be placed right on the voltage divider line. The second isolation amplifier will be in charge of reading in the current value. In order to read in the current value, a current-sensing resistor must be used.

The current sensing resistor chosen was a 0.2ohm shunt resistor. In order to connect this resistor, we must break the neutral line of the outlet and place this resistor within the break. The second isolation amplifier will then be connected to this shunt resistor. These connections can be viewed below in Figure 4.3.2.



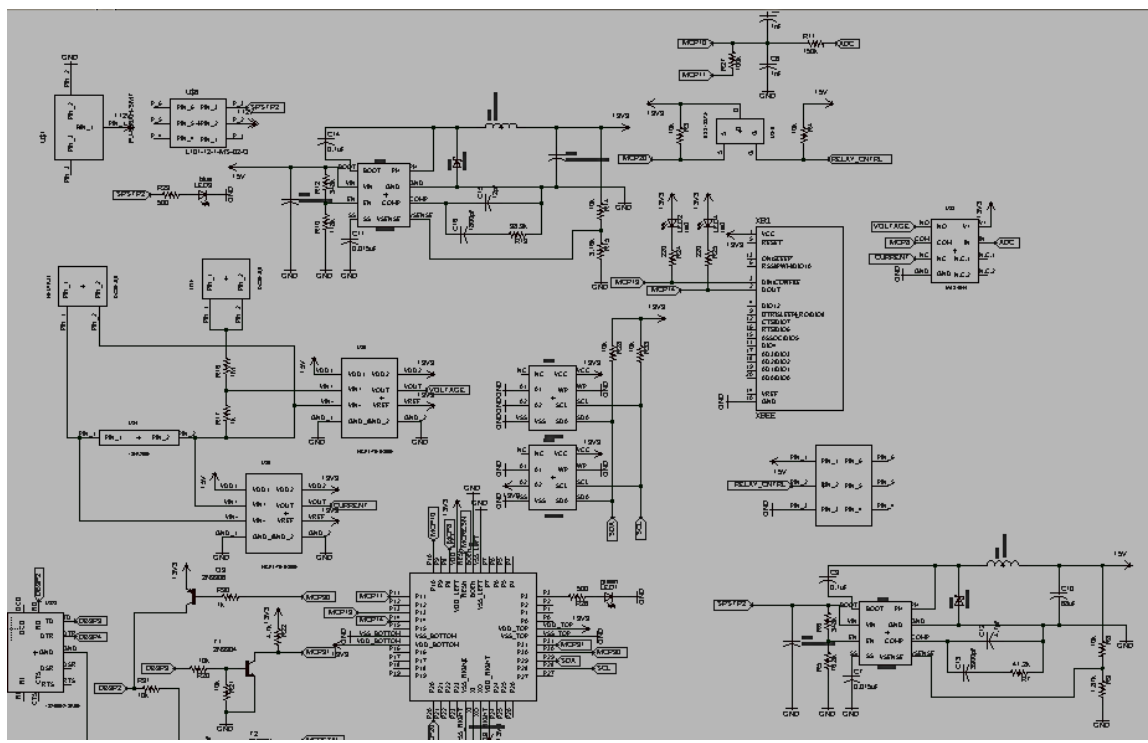
**Figure 4.3.2:** Voltage/Current Isolators

**Outlets:** One other hardware to consider for this design is which outlet to use for the setup. Due to the nature of this project and working with live AC lines, we had to take into consideration some safety measures when selecting this specific hardware. With this in mind, we decided to take a step away from any traditional home outlet. What we did find were GFCI Outlets, or Ground Fault Circuit Interrupters. These are commonly found in kitchens or bathrooms where there may be a risk of shock for the users.

The way these outlets function is that if it detects a slight change in the amount of current going into a device and when it leaves it, it will automatically cut power to the outlet. The reaction time is as little as 1/40 of a second which will greatly reduce the risk and injury involved with handling live AC lines.

**Relay:** A reach goal for this design is the implantation of a method to either automatically or manually turn on or off an outlet. For instance if the user see's that one outlet is pulling too much power and wants to shut it off, they may do so through our device.

Many different methods were research to implement this idea. The one that was chosen was a relay. The COM-10924 Relay Single Pole Double Throw will be implemented to control the power between a device and the outlet. It is rated up to 2000VAC at 20A, which is a lot higher than many of the other relays considered. In order to trigger this relay, 5VDC signal must be sent to the pin labeled “5V.”



*Figure 4.3.3: KWH Schematic*

### 4.3.2 Kilowatt Hour Meter Software Design

**24-Bit Sigma Delta Analog-Digital-Converter:** 24-Bit Sigma Deltas, or SD24, are required in order to measure the voltage or current inputs. The modulation frequency for the SD24 is chosen to be set at 1 MHz, with an OSR at 256, and an 8 MHz ACLK. With these values, the sampling frequency can be calculated by dividing the modulation frequency (1 MHz) by the OSR (256) which would equate to a sampling frequency of roughly 3906.25 Hz. The Sigma Delta converter should periodically interrupt after each sample.

**Voltage, Current, Power, Energy Calculations:** Once the samples have been collected using the sampling rate from above, utilizing the following the equations will result in the  $V_{RMS}$ ,  $I_{RMS}$ , and Power(both active and reactive):

$$V_{RMS} = K_v * \sqrt{\frac{\sum_{x=1}^y v^2(x)}{y}} \quad I_{RMS} = K_i * \sqrt{\frac{\sum_{x=1}^y i^2(x)}{y}}$$

$$P_A = K_p * \sqrt{\frac{\sum_{x=1}^y v(x) * i(x)}{y}} \quad P_R = K_p * \sqrt{\frac{\sum_{x=1}^y v_{90}(x) * i(x)}{y}}$$

Where ‘v(x)’ and ‘i(x)’ is the sample reading of their respective kind at the instant ‘x’, ‘P’ is the active and reactive power respectively, ‘y’ is the sample count taken in one second,  $v_{90}$  is the voltage shifted by 90° and ‘K’ is the scaling factor of their respective sources.

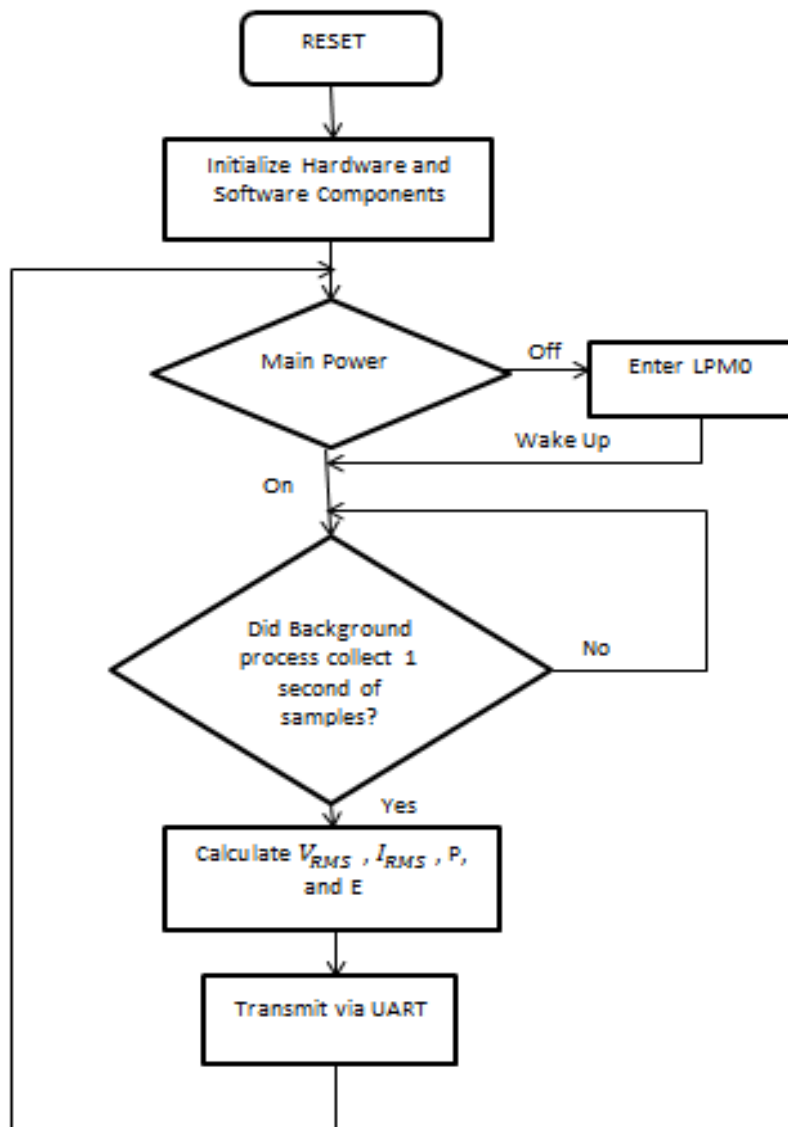
Energy is calculated by multiplying the respective power ‘P’ (active or reactive) by the sample count ‘y’.

$$E = P * y \quad \text{*Use active P to calculate active Energy and vice versa}$$

**Software Processes:** The process consists of both a Foreground and Background Process. The background process interrupts and handles all the data collection while the foreground handles the calculations. Figures 4.3.5 and 4.3.6 demonstrate these processes in a flowchart manner.

**Foreground Process:** Refer to Figure 4.3.5 for the flowchart of this process. This process will initialize all the hardware and software components (USART, ADC, Clock, I/O, and Timer) following the device reset. Once initialized, it will check for the main

power and if it is found to be off, then the device will enter LPM0. From here, the unit will wait for confirmation from the background process that one second worth of data has been accumulated and collected. When confirmed, the foreground process will begin calculation of the RMS voltages, currents, power and energy which will then be stored in the SD24 registers and then transmitted via UART. After transmission, the process will restart after the initialization of the components.

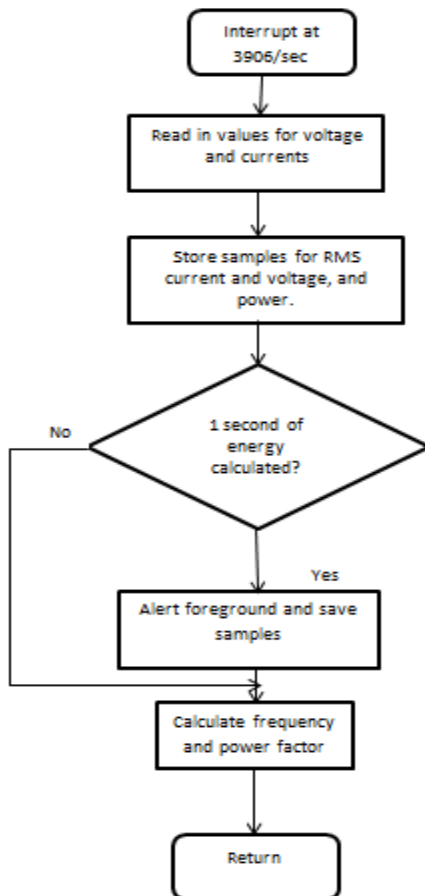


**Figure 4.3.5: Foreground Process**

**Background Process:** This section will interrupt the foreground process at a rate of 3.906 kHz. The background process is responsible for the accumulation of data for every



one second cycle as well as storing them in resistors for the foreground process to recall and use for calculations. But the foreground isn't responsible for all the calculations, the background will also calculate the frequency as well as power factors needed. Refer to flowchart 4.3.6 for the flowchart of the background process.



**Figure 4.3.6: Background Process Flowchart**

**Sample Tracking:** The instantaneous values of the voltage and current signals for each sample set are saved in the 48-bit registers of the SD24. A counter for both the sample and the sample sets is used to determine the number of samples accumulated from each set. Once there is one second of samples collected, these samples are then stored in the registers via background process and then a signal is sent to the foreground to calculate the RMS voltage, current, and power.

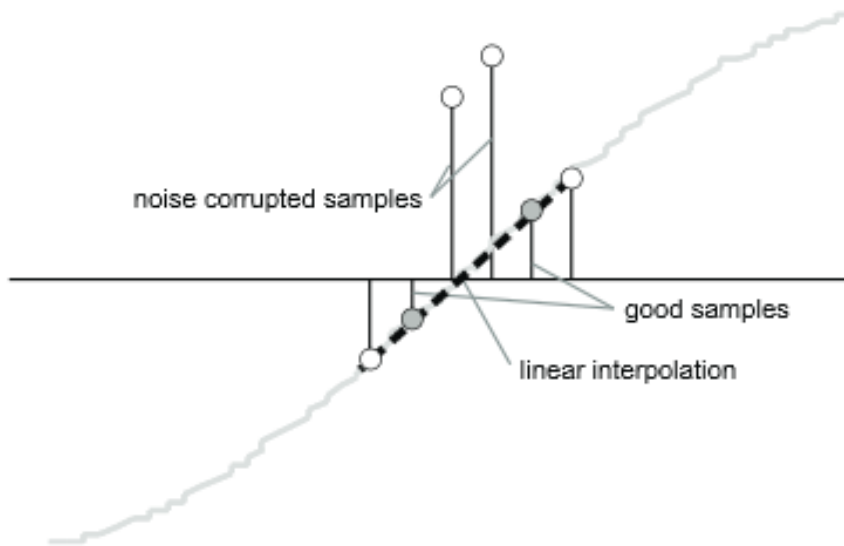
**Phase Compensation:** With the combined use of the Current Transformer and passive components, a phase shift has occurred which needs to be compensated. This phase shift exists between the voltage and current signals but the Sigma Delta 24-Bit ADC has an

innate delay hardware. With this delay, the existing phase shift can be compensated. From Texas Instrument's Application Report, this delay is calculated using:

$$\text{Delay resolution}_{Deg} = \frac{360^\circ * f_{in}}{OSR * f_s} = \frac{360^\circ * f_{in}}{f_m}$$

(Courtesy of Texas Instruments)

**Frequency Measurement:** Figure 4.3.7 shows an example of an interpolation for frequency measurements (courtesy of Texas Instruments). The figure shows how some samples might be corrupted due to noise from the unit. Using the interpolation, these samples can be brought to attention and removed resulting in a much more smooth and accurate representation of the frequency.

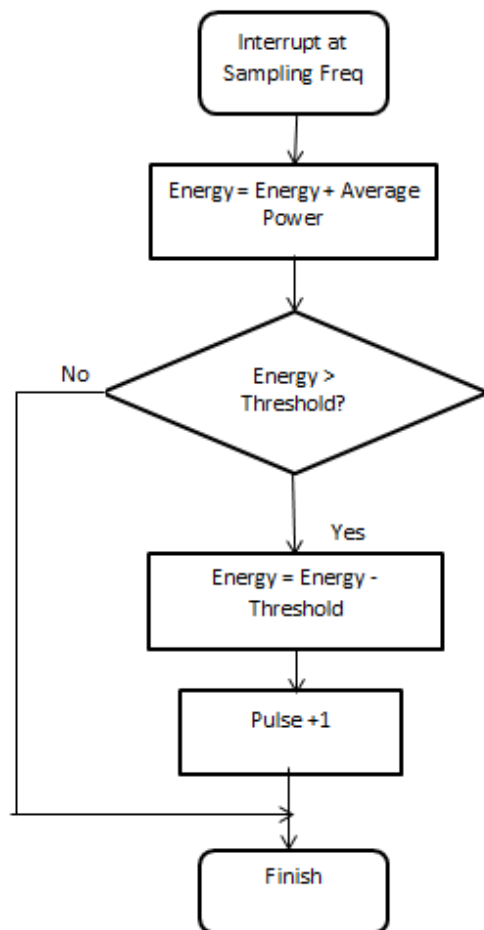


**Figure 4.3.7: Example of Frequency Interpolation**

**Pulse Energy Calculation:** Typically, energy consumption for electronic kilowatt hour meters are measured using LED pulses. Therefore the kilowatt hour meter's unit must be capable of recording and storing the number of pulses generated in a predetermined cycle set. Once the energy consumption crosses a set threshold, the unit will emit a single pulse. Any amount of energy above the threshold will be stored and be counted towards the next pulse cycle. When a pulse is triggered, an LED will flicker to notify the system. This method is used to notify the system that a pulse has been created and that the count

must be incremented. A constant defined for the number of pulses per kilowatt hour is set by the user. This constant will determine how much energy is required in order for a pulse to be generated.

Figure 4.3.8 shows a flow chart that depicts the process in which a pulse is generated. Beginning with the interrupt each sampling frequency, the unit will check to see if the overall energy consumption has broken the threshold. If so it will then subtract the threshold from the energy to store the excess energy for the next cycle and then generate one pulse. If the threshold has not be reached, the unit will continue to perform as usual and end the pulse check.



**Figure 4.3.8: Flowchart for Pulse Generation**

## 4.4 Third Subsystem: Radio Transceiver Implementation

After discovering the current, voltage and the power values, the data needs to be sent to the central hub, which in this case will be the display itself. While it is possible for us to create a radio transceiver, it would be much more cost and time effective to simply purchase a transceiver and integrate it into the project. For our purposes, we will choose to use the LINX TRM 900NT, as it fits all of our needs and is fairly inexpensive.

### 4.4.1 RF Technology

We had briefly spoken about how wireless technologies worked in section 3.2.2, but that was a very cursory look at what is actually going on. RF technology is everywhere and used all the time, but it is much more complex than most people realize. It encompasses electromagnetic waves, complex electric circuitry, and a bit of computer programming to function properly.

RF transmitters are very small pieces of technology, but there are quite a few components that it is comprised of. There needs to be an oscillator to create the electromagnetic wave, amplifiers to boost the signal, a modulator to encode the data, and an antenna to emit the radio waves themselves, and that is only the transmitter circuitry. On the receiving end a filter blocks incorrect frequencies, then a demodulator decodes the signal to a usable level, and finally the data goes into the processor where it can be used to send the proper values to be displayed on the screen.

Oscillators work much like a pendulum, bouncing energy back and forth between a capacitor and an inductor connected in parallel. Where a pendulum goes back and forth between potential and kinetic energy however, the capacitor and inductor pass electrical and magnetic energy between each other. When the capacitor becomes fully charged, it discharges into the inductor where the current creates a magnetic field that then recharges the capacitor's opposite plate and starts the process all over again. Eventually this energy would fade because resistance within the wire would dissipate the energy in the form of thermal energy, so to counteract this the circuit requires a power source to maintain the frequency.

After the oscillator creates the signal frequency, and the signal is modulated with the data stream built into it the signal is sent to the antenna to be emitted at the same frequency that was set by the oscillator. The receiver, which is taking in all radio waves, is tuned to that particular frequency, and thus, only takes in the signal from the transmitter. The signal then travels to the demodulator to decode the signal, taking apart the carrier wave from the data signal and sending the data to the processor.

Radio waves can vary in frequency from 3 kHz to 300 GHz. The higher the frequency, the shorter the range of the wavelength. Lower frequencies have a much larger range and can penetrate through and around obstacles better. Radio waves are a form of radiation, so the FCC put Maximum Permissible Exposure Limits for field strength and power density on radio frequencies between 300 kHz to 100 GHz to limit the effect of the radio waves on the human condition.

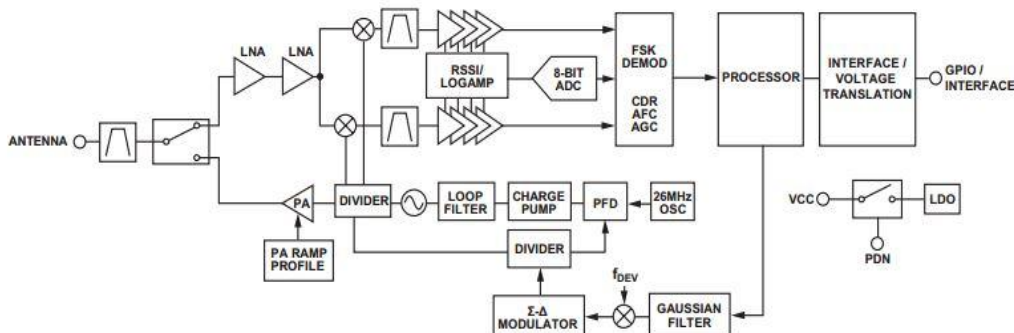
#### 4.4.2 Hardware Design Components

**LINX 900 TRM NT:** One possible RF transceivers this project will be using is the LINX 900 TRM NT (Figure 4.4.2.1). The LINX is an extremely small piece of hardware, nonetheless it has many different important parts. The LINX boasts a range of up to 3000 feet with line of sight, a low cost, low power consumption, and a data rate up to 300 kbps.



*Figure 4.4.2.1: The LINX 900 TRM NT*

As can be seen from the figure below, the LINX 900 is a transceiver, capable of sending and receiving data. The path of transmission can be followed if the figure is followed from the processor clockwise down through the Gaussian filter where the data from the processor is modulated into the carrier signal generated by the oscillator, which oscillates at 26 MHz. The signals are then combined and the frequency is boosted to 902-928 MHz as it transmitted out the antenna. Receiving signals begin instead at the antenna, are amplified, and then go through the demodulator before entering the processor where it is translated to machine coding.



Absolute Maximum Ratings				
Supply Voltage $V_{CC}$	-0.3	to	+5.5	VDC
Any Input or Output Pin	-0.3	to	$V_{CC} + 0.3$	VDC
RF Input		0		dBm
Operating Temperature	-40	to	+85	°C
Storage Temperature	-55	to	+125	°C

Exceeding any of the limits of this section may lead to permanent damage to the device. Furthermore, extended operation at these maximum ratings may reduce the life of this device.

*Figure 4.4.2.3: Chart of Absolute Maximum Ratings*

As with any piece of technology, there are limits as to how it can operate. For the LINX 900 the voltage cannot exceed 0.3 volts on any single pin or 5.5 volts on the supply voltage or else the device runs the risk of overheating and destroying very delicate parts. In most cases, temperature would not be an issue, but in cases of extreme weather, or

misplacing the chip in a freezer, or near a heat source could break or melt connections within the chip. These specifications shouldn't be much of an issue in the overall project, although they are to be kept in mind in order to maintain the integrity of the chip and make sure that we don't waste any hardware due to carelessness.

What is most important about the LINX is not how it works, but how we will make use of its capabilities. The LINX does not encode or packetize any data given to it, it only modulates the data. This allows great freedom for us to implement coding for the chip to send the data that we need to the display. There will be at least two chips, one in the sensor array and another within the central hub. The chip on the sensor will be connected to the sensor taking the data in. From there, the transmitter chip will send the power data to the receiver chip within the display. The receiver will then take the data and output it to the display to be shown on screen.

1	GND	GND	44
2	NC	ANTENNA	43
3	GND	GND	42
4	NC	NC	41
5	NC	NC	40
6	GND	GND	39
7	NC	DATA_IN	38
8	NC	DATA_OUT	37
9	TRPT/PKT	NC	36
10	CHN_SEL0	NC	35
11	GND	GND	34
12	CHN_SEL1	NC	33
13	CHN_SEL2	NC	32
14	LVL_ADJ	CMD_DATA_BAUD	31
15	READY	CMD_DATA_TYPE	30
16	NC	CMD_DATA_OUT	29
17	GND	GND	28
18	T/R_SEL	CMD_DATA_IN	27
19	BAUD0	STANDBY	26
20	BAUD1	VCC	25
21	RSSI	POWER_DOWN	24
22	GND	GND	23



*Figure 4.4.2.4: The pin set of the LINX Transceiver Processor*

**Pin Descriptions:** This figure shows the connections available for use within the LINX 900. All the NC notations depict no connection. GND are all ground lines and VCC is the voltage supply. The LVL\_ADJ line sets the transceiver output power level. In some legal cases a lower power level is required so the manual recommends connecting this pin to the ground along with a resistor to lower the power level this line has an internal 100 k $\Omega$  pull-up resistor so we have the option to pull high or leave open for full power. Unless doing this causes an overheat this will likely not be an issue. The READY line is low voltage when the chip is ready to receive data and high when it is busy with data. This can be of use during the software stage in order to determine whether or not there is any communication occurring at any given moment. TRPT/PKT stands for transparent/packet data select this may prove necessary to select each data packet. T/R\_SEL is the setting to change whether or not the chip is in transmit mode and receive mode. The BAUD lines set the data rates and filter bandwidths. RSSI represents received signal strength indicator and it outputs an analog voltage that is proportional to the received signal. POWER\_DOWN sets the module into a very low power state which disables the module until reset. STANDBY is similar to powering it down by placing it in a low power state the module will remain functioning, but will perform no operations until standby is released. All the CMD lines are special operations that require contact with the LINX company itself for their function. To keep normal operations, we must set CMD\_DATA\_IN to high, CMD\_DATA\_TYPE and CMD\_DATA\_BAUD to low, and CMD\_DATA\_OUT disconnected.

The DATA\_IN and DATA\_OUT lines will be the most commonly used lines, as these two are essentially the input at output lines for the data stream. The in line accepts data to be transmitted, and the outline outputs the demodulated data. The CHN\_SEL lines are for choosing which frequency the transceiver is tuned to. There are three lines which means there can be 8 different channels we can use. The figure below displays the channel configurations and their corresponding frequencies.

Channel Selection				
CHN_SEL2	CHN_SEL1	CHN_SEL0	CHANNEL	FREQUENCY
0	0	0	3	903.37
0	0	1	15	906.37
0	1	0	21	907.87
0	1	1	27	909.37
1	0	0	39	912.37
1	0	1	51	915.37
1	1	0	69	919.87
1	1	1	75	921.37



*Figure 4.4.2.5: Channel settings and their frequencies*

The LINX was the first and most desired choice for the RF Transceiver component. But as research continued for other subsystems, another option was discovered: the XBee Series One.

**XBee 802.15.4 Series 1:** The XBee family consists of many different components that need to be taken into consideration before a decision is made. To begin, there are two categories for the XBee; Xbee Series 1 and XBee Series 2. Series 1 devices are generally a lot easier to use then the Series 2. Due to the nature of this design and course, the Series 1 will be selected.

Now within the series categories, there is another set of classification for the devices. Consumers may choose to use either a XBee Series 1 or an XBee Series 1 Pro. The difference between this two pretty much summarizes into how much power each one consumes, the range of which is can broadcast, and the cost. The biggest factor between deciding which one to use is the range. The Xbee Series 1 has a functioning range of about 300 feet which the Xbee Series 1 Pro has a much larger range of roughly 1 mile. The Xbee Series 1 Pro is a lot stronger than the standard and is capable of a lot more feats, but the Regular Series 1 will be more than enough for the nature of this project since it will used within a household. In the table below (Figure 4.4.2.6) you will see all of the specifications of the XBee Series 1 Standard chip.

RF Data Rate	250 kps
Range	300 ft
Transmit Power	1 mW
Serial Data Interface	3.3 CMOS UART
Frequency Band	2.4 GHz
ADC Inputs	(6) 10 Bit ADC
Supply Voltage	2.8-3.4VDC
Transmit/Receive Current	45mA – 50mA @ 3.3VDC

*Figure 4.4.2.6: XBee Series 1 Specifications*

An important note to make while researching is the pin layout and definitions of the XBee RF Transceiver. In Figure 4.4.2.7 you will find the exact pin numbers and definition for each pin. The main ones that will be utilized are pins 2 and 3 since the Parallax Propeller Microcontroller will be handling all the data collection and calculations. The XBee chip's input and output pins could've been utilized but due to the potential speed, the decision was made to spread out the load and just use the XBee for data transmission.

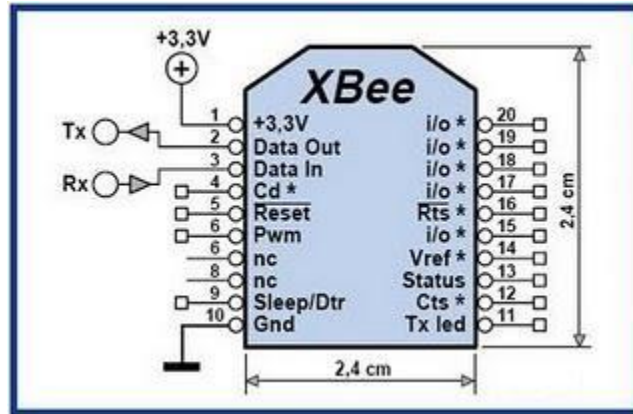


Figure 4.4.2.7: XBee Pin Layout

### 4.4.3 Software Design Components

The signals sent and received would mean nothing if we couldn't process them into meaningful information. The LINX chip has no modulation restrictions, so as long as the voltage limitations are not surpassed it should be possible to program the chip with any development kit. The LINX has its own development kit, but if we could make use of the Parallax system we could save money and space components.

The software's programming would heavily depend on the amount of information gathered from the sensors, the tasks that we would have accomplished, and how intricate we would need the information displayed. The transmitting chip would send from the sensor the voltage, current, power, and the time values to the receiver. The current and voltage values can be taken and sent off right away to display them in close to real time. The power and time factor data however, first need to have some calculations done to them.

In the processor the data would meet with functions. There would be a function that will calculate the projected cost of operating the current device by referring to the rate of the electric company in the user's area. Every company has a different rate depending on the state and what sort of power is being supplied. We could have a database for the state, for the rates of the different electricity suppliers, so that the user would only need to enter their zip code or city to determine the rate of cost. With the rate in consideration, the cost of operation can be estimated by calculating how much power is being used in the current count of time multiplied by the number of days in a month or a year. There would also be a function to then determine the amount of carbon used by the device by dividing the amount of power consumed by the rate of the company, then taking that number and multiplying it by the carbon emission factor of the state for the sector the user is in. These new values would then have to be sent to the display processor to be configured on the screen.

There would be other functions as well that are more for the function of sending and receiving data. There would be a function that detects what state the chip is in, whether it is busy or ready to receive more data, a function to determine how much data has come in, and function to determine which value is being sent to the receiver. On the transmitting end there would be functions to create the input buffer and take in the data from the sensors, and to attempt to send the data whenever the receiver is ready to take them.

## 5.0 Design Summary of Hardware and Software

**VGA Summary** The visual graphics array (VGA) controller will have multiple functions. The first of these functions will be to output a picture to the user. This display will contain important information about what is going on in the system as well as give the user a way to interact with it. The second function will be controlling everything in the overall system. Each of the individual outlets that measure power will be sending the data to this unit and it is up to this microcontroller to process the data and display the correct information back to the user.

There are several components that comprise the system that will do all of these functions. The first component which does most of the heavily lifting in terms of processing data into information will be the Parallax Propeller. This microcontroller is capable of generating VGA signals that can be sent to a display for the user to interpret. In addition to having this unique ability over many options on the market it will also be responsible for interpreting analog signals from the monitor. The VGA display will be activated by touch and these analog signals will need to be interpreted by the Parallax Propeller microcontroller to know what data needs to be processed to generate the next correct screen.

With the microcontroller selected an EEPROM will be used to store the data that will need to be kept long term. This component will store both the program and its data to ROM where it can be accessed in the event that the main unit is shut down for whatever reason. In choosing to implement a long term memory storage solution it is possible to keep a history of the users data that can be reviewed at a later time in order to be interpreted for meeting goals that have been set. This information can also be used to be displayed in graphs that gives a better representation of the power consumption in a home over the course of using the system.

Another vital component is the wireless card that will be integrated onto the board. This card will be the medium for which the main unit and the outlet microcontrollers communicate with each other. This form of communication will take place using radio frequencies and interpreting the data in a serial port type of fashion. Without this

component on the board it would not be easy to set up the system in a home without wiring every microcontroller to each other which is not practical to do. In using this device, fast and efficient communication can be accomplished between the devices.

Finally the last component for consideration is the voltage regulator that will be needed to control the incoming voltage to the main unit. There are stringent and specific requirements that manufacturers expect to be made while using their components and these will have to be addressed. If too much voltage or current is consumed by a device due to the failure to regulate the power will result in catastrophic failure that will not be possible to repair.

On the software side there will be a few functions that will determine the flow of the program. The first of these is receiving input from the user from the touch screen. This will dictate where in the program the user will move to. This will change depending on different options being selected in the menus. In addition to receiving input from the user the other function that will control what is going on is receiving data from other microcontrollers. The data will always be recorded from the outlet microcontrollers for historical purposes, but the display will also need to be updated if the user is on a screen where real time data is required. These two independent functions will fortunately have the ability of running on separate processors within the same microcontroller.

**Kilowatt Hour Meter:** The hardware components consist of the following main parts: Microcontroller, Power supply, voltage and current input signals. The microcontroller was selected from the MSP430xxxx family due to its low energy requirement, functionality, and cost efficient design. Having a cheap design was an overall goal of this project and choosing cost efficient components is definitely a must.

The power supplies chosen to drive the microcontrollers were based upon their ability to maintain the units as well as their cost and simplicity. A simple and cost efficient design is crucial to a project with limited time and resources. The power supplies used are based off of the resistor capacitor power supply design as well as the switched base power supply design.

The final hardware component of the meter was the voltage and current input signals. These signals had to be manipulated so that their values stayed within the allowed ranges of the 24-bit sigma delta analog digital converters. In order to manipulate the signals, two filters were placed at the analog front end of the current and voltage inputs. These filters were the anti-aliasing filter and an RC low-pass filter. By utilizing the voltage divider and filters and calculations based on the maximum main of 230 V, a hardware design was created that allowed for a voltage who met the criteria to be passed into the positive input terminals.

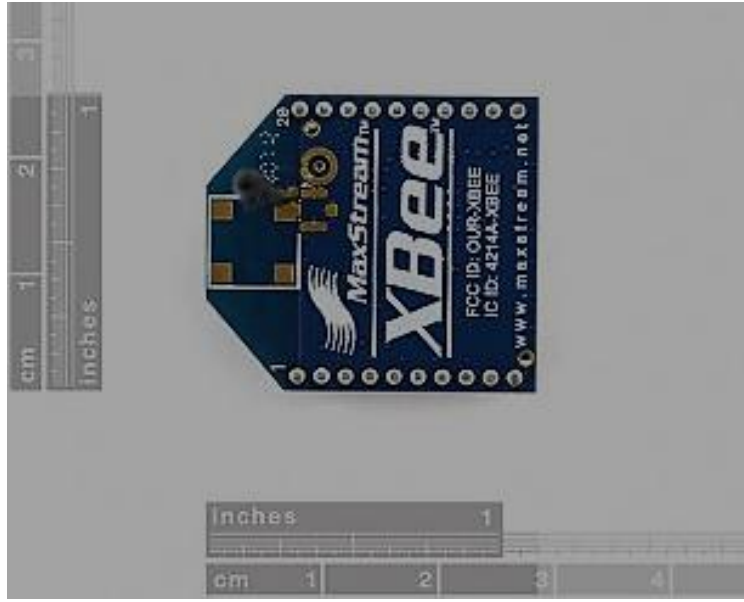
The software design for the kilowatt hour meter is responsible for all the data collection, storing, and calculations. Within this section, the unit will run through several phases of data collection and calculation.

The foreground process is what occurs on the outside. During this process the software will receive data from the unit. It will then be passed to the background process which in turn stores the data. The foreground process will check to ensure that sufficient data has been stored before completing the computation process. Once calculations have been completed, the foreground process will relay this information via UART to the desired receiver.

The energy calculation done in the software components utilize a threshold breaching system. The user will set an energy threshold and once that limit has been reached, the system will raise a flag to increment the multiplication constant. This constant is used to keep track of how many times the energy consumption passes a set value and in turn determines the cost of the monthly energy bill.

Amongst the units measured are the frequencies and phase compensations. A phase shift does occur between the inputs, therefore this need to be compensated for. Luckily, the hardware chosen has a built in delay to compensate for the phase which needs to be set using the formula provided in section 4.3.2. The frequency measurements are tracked in order to account for any peaks and erroneous samples. By utilizing an interpolation graph, these samples can easily be spotted and ignored.

**Radio Transceiver:** For the radio transceivers, the design is fairly simple. The XBee chips will be set to a frequency between 903.37 MHz to 921.37 MHz, depending on the channel chosen for that particular transceiver. The XBee transceiver is a very tiny piece of equipment. The dimensions of it are shown below, just barely over an inch long and an inch wide as displayed in Figure 5.0.1, this little transceiver should be able to fit into the sensor and the central hub with no problems.



*Figure 5.0.1: XBee RF Dimensions*

The project will need at the very least two transceivers, one connected to the sensor array and the other connected to the central hub. The transceiver in the sensor will need to be receiving information from the sensor to the DATA\_IN pin. This will take in data to be modulated and transmitted to the receiver. The sensor will need to collect the voltage, current, and if possible the power or at least the phase of the current, so that the power can be calculated.

The receiver in the central hub would take these transmitted electrical statistics and then perform calculations on them to find out desired values such as cost of operation, the amount of carbon used to keep the device running, the close to real-time values of the current and voltage, and the power used in kWh over the course of a day, or a month, or a year.

In order to find these values the processor of the receiver must be programmed to not only find the transmission and decode it, but to use the data in a meaningful way. First, there would be a loop that constantly checks to see if there is a transmission ready. When one is detected, the program would then move to an input function. This function will take in the demodulated signal and read the data and then save the values into the registers of the processor. These registers would hold each value from the sensor separately until the program uses the calculation functions to find the calculated data. When the cost, carbon, and power values have been calculated, the data is then ready to be sent to the central hub for display processing. The display would be connected to the DATA\_OUT line to receive this data.

**Relay:** The design for the circuit breaker will be an electromagnetic circuit breaker with a radio transceiver and microcontroller implemented for automatic triggering of the switch. The electromagnetic design is most suitable for household electronics, and it is very cheap to implement. The radio transceiver can receive the signal from the main unit

commanding the power off of a device, and command the microcontroller to send a trip signal that will create a short circuit in the appropriate circuit. After the breaker is tripped, the circuit remains open until the switch is reset, and normal operation commences. This process can be repeated whenever necessary.

## 6.1 Printed Circuit Board Assembly

### 6.1.1 PCB Design Software:

One of the more popular free printed circuit board design software programs out there is CadSoft's EAGLE PCB Design. The version used will be its free version, EAGLE Light Edition. There are certain restrictions/limitations that come along with the EAGLE Light Edition.

The PCB area is limited to a design of dimension 100 x 80 mm which is equivalent to 4 x 3.2 inches. This edition only allows for single layers to be used and the schematic editor is able capable of creating one sheet. However, the Light Edition is still capable of loading, viewing, or printing designs outside of the set limitations.

### 6.1.2 PCB Fabrication:

The board fabrication will be handled through a third party vendor. The budget for this cost has been accounted for and included in the overall budget. There are many companies that will offer student discounts for fabricating PCBs. Such companies that do so are "4PCB" which offers a 2 layer for a mere \$33 and "PCBFABEPRESS" offers 2 layer PCBs for a base of \$40 + \$.60/in<sup>2</sup>. With the current rates and deals, "4PCB" will be the source for this project's PCB fabrication.

## 6.2 Embedded Software Detailed Design

### 6.2.1 Video Graphics Array Controller Software Design

**System Startup:** The flowchart seen in Figure 6.2.1.1 details the boot processes before the system makes it to the main program. The software will first check to see if the system was previously setup. This will be possible with the EEPROM included in the design as a permanent variable can be set in ROM that will keep track of this setup information. If the system determines that the device has not been booted previously it will then ask the user to input a serial for one of the outlet receivers. It is imperative that the user know all the serials to each device as this is how they will be identified. It is a

bad design to ask how many outlets there are or for it to randomly scan to find outlets in the area as some may not be working and the user would not know. With each outlet having its own unique serial any failure within the system can be immediately determined if a response is never received.

From that point the user can enter an alias for an outlet on the top socket and on the bottom socket. An alias can be along the lines of naming a television, gaming console, personal computer, etc. If the user decides they do not want to give a specific outlet in the list a name as it has no real meaning to them then the field can be left blank and a default name of "Outlet 1A" for example can be automatically assigned. The system will continue asking if there are any more outlets that are a part of the system until the user selects no. At this point it sets the flag of first time booting to false so that the left side can initiate. As previously stated the flag within the ROM will be set so this path cannot be taken again without resetting the system to factory (granted it did not come from a factory) defaults.

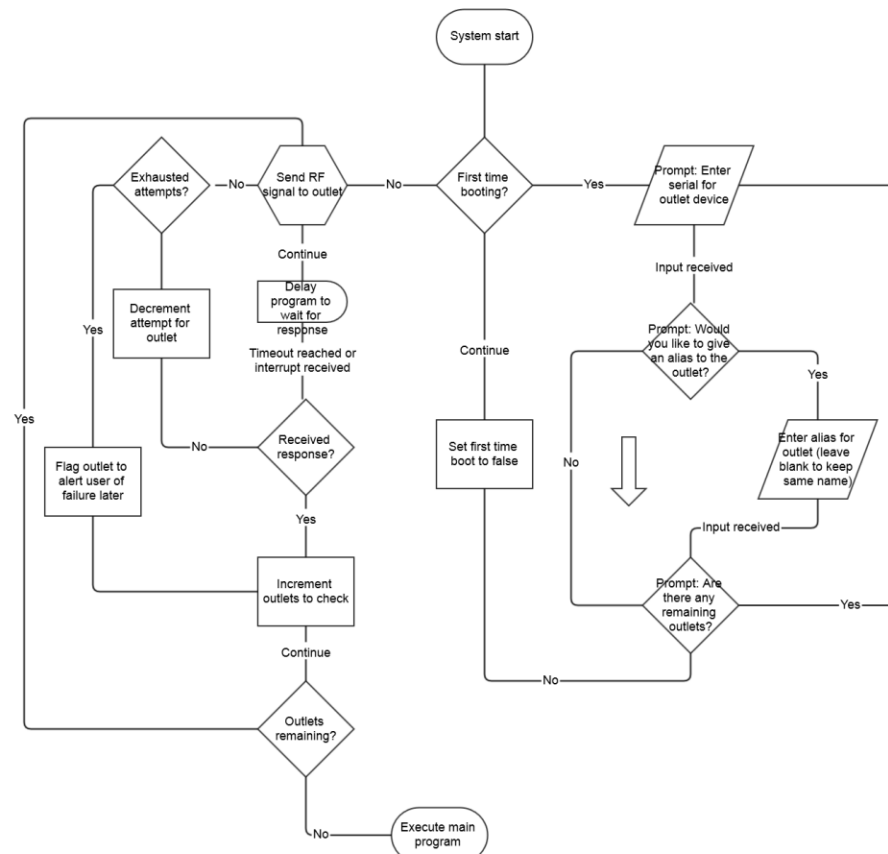


Figure 6.2.1.1: Flow chart of the boot up processes



The left side of the flow chart is important in that communication with the outlets is initiated in this phase. A predetermined timeout duration will be set in that if no response is received in this time frame it will be considered a failed attempt at contact. If this failure happens another predetermined amount of attempts will be made and should they all fail the outlet will be flagged. With each serial entered the system begins asking each outlet to respond to the main system if its serial matches the request. It is important to note that all of the outlets will receive the same signal asking the same question, but only one outlet will reply if a match is found. In the case of a failure, nothing will reply in the set timeout duration. Should this be the case a flag will be set for that outlet and at a later point the user will be notified of the problem.

After all the outlets are scanned for and responses or lack thereof are received the main program will start. The scanning of the ports will happen each time the device is turned on, granted it will probably not be powered off after the initial setup. The serial system with the outlets will allow for newer outlets to be added to the system at a later point. This flexibility is in place so the user does not have to setup every outlet at once if only a few are to be tested at first. It also prevents the main unit from ever having to be set up again or reprogrammed. Once the system has been given the serial it will be stored in ROM until for whatever reason the user decides to remove the device from the system.

**Main Program:** Figure 6.2.1.2 shows the basic flow of the main program after it is loaded. As has been stated before, the Propeller Parallax is a multi-core processor with the capability of running different tasks on different cores known as Cogs. It should be known that despite what names have been given for the Cogs in the figure, it does not reflect which processor the task will take place on. The microcontroller is designed to look for a free processor and assign it a task.

Looking at the same figure, Cog 0 as it is arbitrarily named will be waiting for input from the user. It could have been possible to have the system be interrupt driven and not waiting on input through polling, but aside from checking on the RF buffer filling up from incoming radio signals, there is not much else to do without user interaction. After an attempt to check if user input was received it will then question if where the screen was touched is valid versus what is on the screen. As this is just a basic flow chart of the flow of the program, it does not go into detail about which positions on the screen will be valid depending on where it is in the program. All that really needs to be known at this point is that if the position is valid at the time of the press it will move on to parse what will be required for the next display on the screen. After the code runs for parsing the next screen the LCD will be updated accordingly and the process will start.

On the right side of Figure 6.2.1.2 is the process that defines checking the buffer for incoming communication via radio frequency. This process like the other is always running independent of the system and therefore always being checked without waiting

on another part of the program to complete. Upon checking the if there is data in the buffer, which will usually be the case it will parse what is in it and update the screen if necessary. There are many readouts throughout the program that indicate power consumption which would need to be updated regularly and displayed to the screen. However there are screens that the user can reside in where real time data would not be displayed, for instance in the options menu and in that case the appropriate registers will be updated but the screen will remain unchanged.

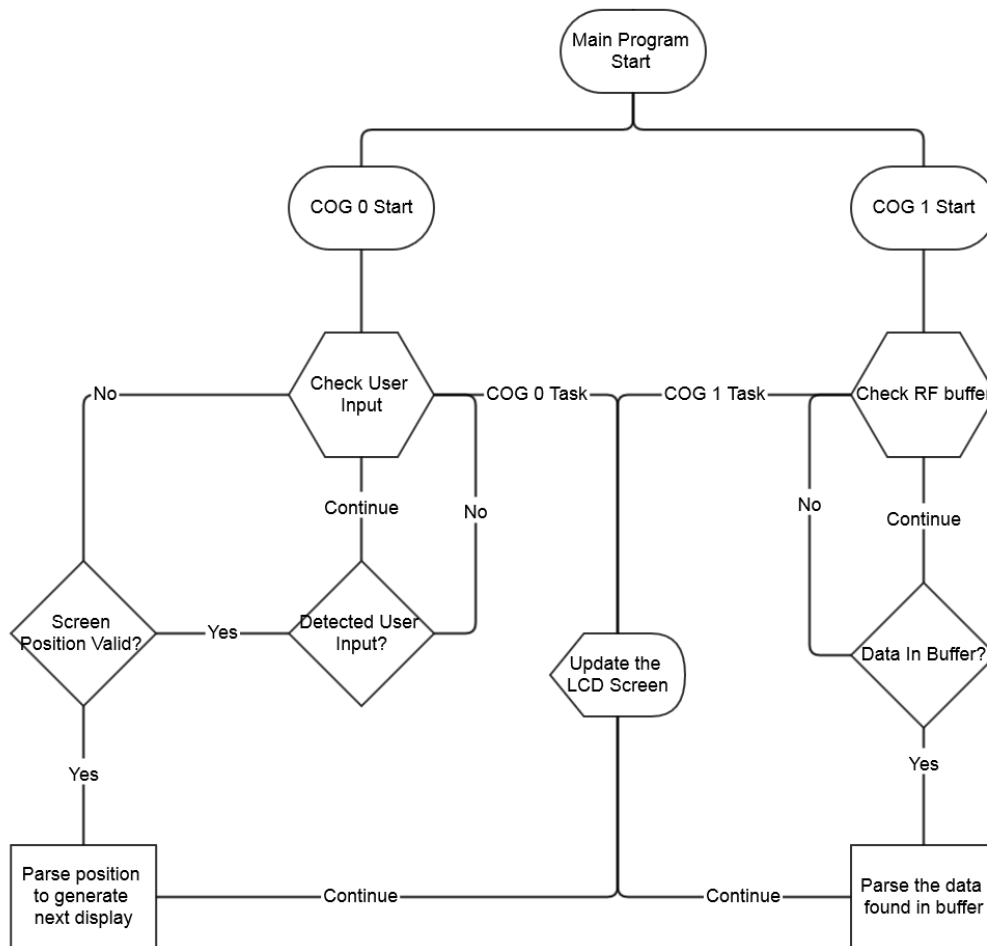


Figure 6.2.1.2: Basic flow of main program running on device

**Development Environment:** The Parallax Propeller supports many different programming languages within the IDE. Spin is a high level object based language has been developed by Parallax that can be compiled and run on the microcontroller. In addition to programming Spin files is Propeller Assembly which has been high optimized to run at a low level like all assembly languages.

Many hardware commands that exist for Propeller Assembly also exist in the Spin language. The microcontroller has a built in Spin interpreter for reading the compiled tokens that was created by the compiler. The assembly version language is assembled into machine code as to be expected and then is run directly. An interesting feature with this chip is that both the high level and low level languages can be used in the same program. Nothing has to be written entirely in one language, but rather a combination of the two types can be used. While the manufacturers languages can be used if the user so chooses, third party languages are still supported that include C, Basic, and more.

Figure 6.2.1.3 shows the development environment for the Parallax Propeller chip. It is simple but includes enough features to provide an environment that is neatly organized with examples to help users get started. On the right side is the directory that is automatically expanded to the Propeller directory that includes examples and other libraries. Upon selecting one of the files the code will display in the right side pane. There are four views that the user can select to get different outputs that is located above the right pane. The first is full source and includes the code and commented lines of code that reside on their own line. The second option is condensed which keeps comments that are on the same line as an instruction, but removes comments that reside on their own line. These comments are related to the program but usually only give an overall description of it and are not as important to always see. The summary section overviews the labels that exist within the program. A nice feature is that the program color codes each section so it is easy to see where one label ends and another begins. Finally is the documentation section which includes any extra documentation a user may have included.

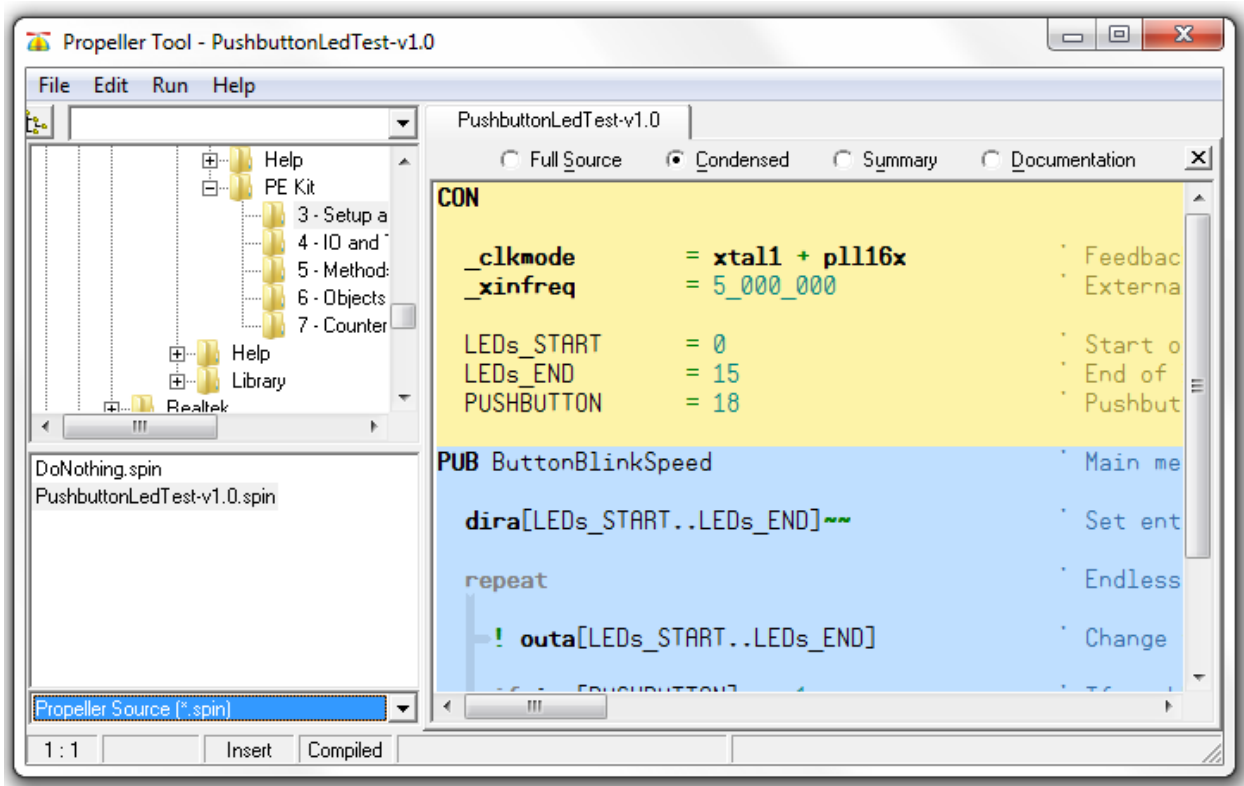


Figure 6.2.1.3: Parallax Propeller IDE

In addition to the previous features previously described are those in the Run menu. The first option in this menu within the Compile Current is the View Info option that bring up the windows shown in Figure 6.2.1.4. This window displays how much RAM a program will use and at what locations it will be stored. In addition to giving RAM usage it displays how much memory the program will consume versus the variables and the stack. These are all measured in Longs.

The second section in this menu shows the clock mode that the microcontroller is operating in as well as the frequency and the XIN frequency. Clicking on the Show Hex option reveals where and what will be stored in memory. The options shown at the bottom of Figure 6.2.1.4 to Load into RAM or the EEPROM are also located separately in the Run menu



to the device involves typing it into the white text box and anything received from the computer will be displayed in the blue box.

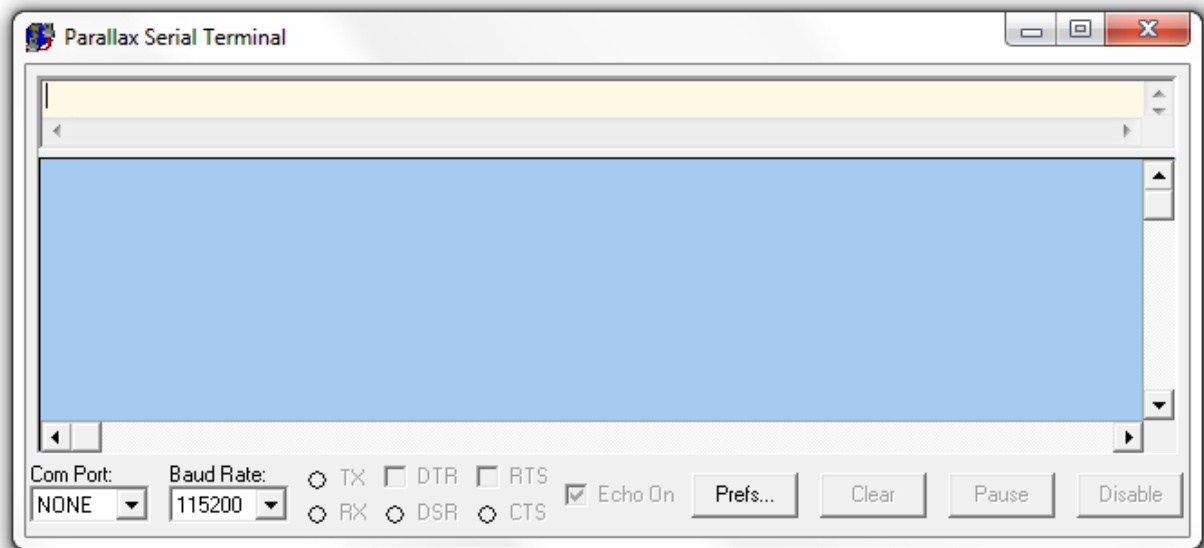


Figure 6.2.1.5: Parallax Serial Terminal Interface

**Interface Design:** The picture in Figure 6.2.1.6 shows an example of the main screen of the program. As can be seen the remaining bill period is displayed so the user has an idea of how much time is left in the billing cycle. The current consumption shows how much power is being consumed overall. This will be further broken down by outlet. Further down the list is how much money it would cost to sustain the current power consumption for the next hour, the next day, and for the rest of the billing period. From this it is able to calculate the current bill, the overall bill based on current consumption and finally if the user will pass what was set for the goal.

Next to the date in the right hand corner is what the option menu button is. This design is found with many mobile websites and the user should be somewhat familiar with it. Within the options menu will be the ability to set the cost per kilowatt hour as well as set what day of the month the bill falls on and setting the current date. This menu will also be used to for accessing the menu to set the goal and to switch between each screen.

Given that the overview screen is displayed, upon clicking the option menu and then clicking the detailed option will allow the user to view the power consumption on each outlet as well as the ability to shut down any particular outlet through the use of a remote circuit breaker. Another detailed section will show different graphs of the various data that has been accumulated over the history of using the device so the user can make adjustments in certain months of the year or to see if they are making any progress toward their goal.

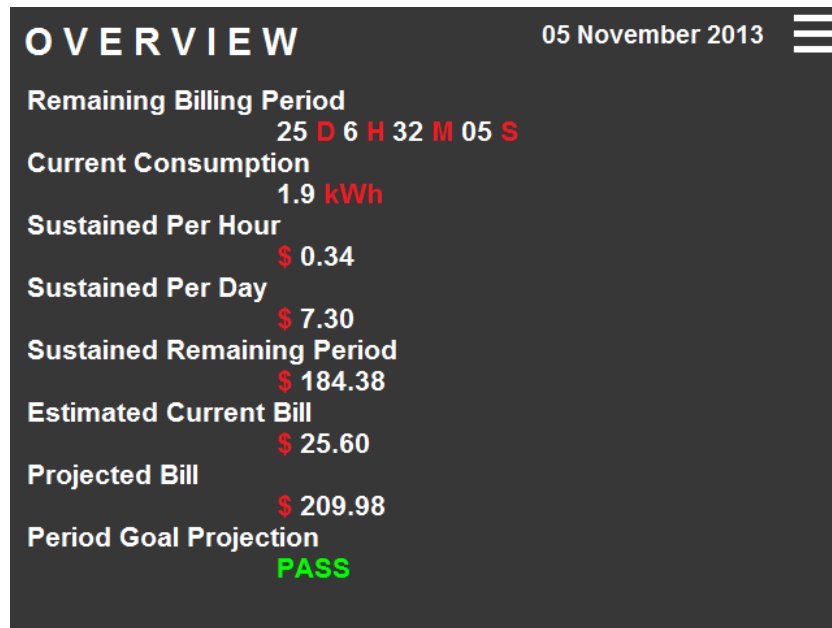
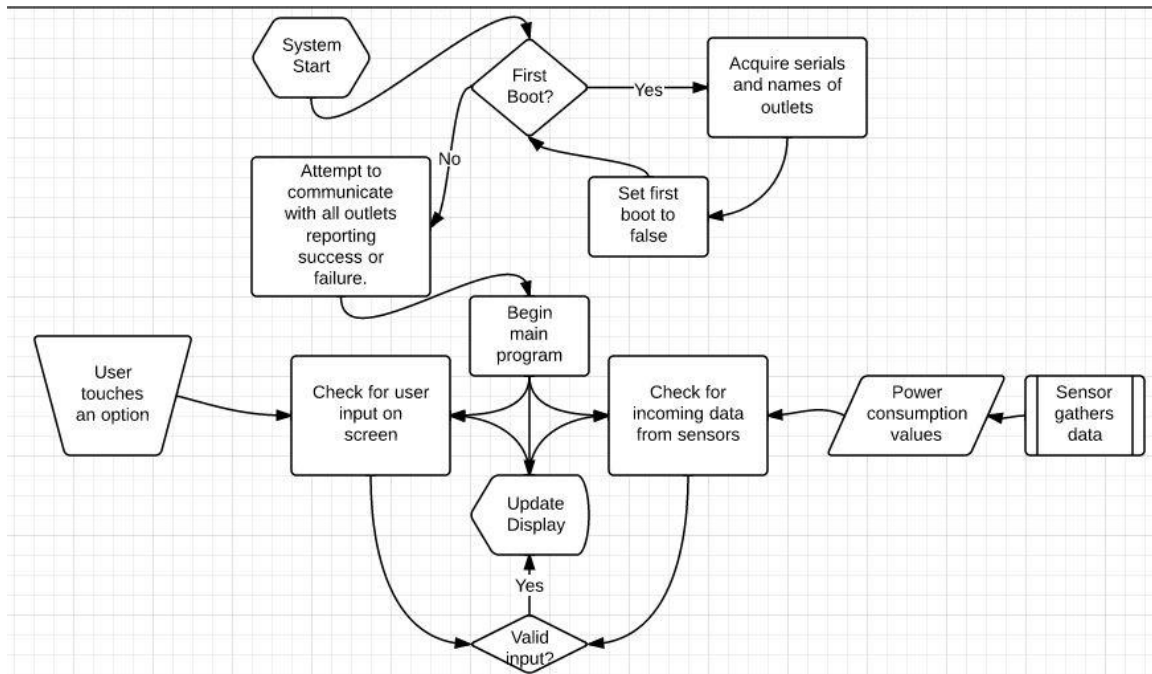


Figure 6.2.1.6: Main screen the program comes to after startup

## 6.3 Final Coding Plan

The main portion of the coding will be within the central hub of the project's system. Upon start up, the program will first attempt to gather information about the sensors and grant them names, then it will attempt to communicate with each outlet. When communication is confirmed, the program can then move on from start up to the principle part of the program which is to update the display with information gathered from the transceivers, and to constantly check to see if the user is providing any input to the screen. The transceivers have their own coding that determine the proper values to be shown on screen. The coding plan process can be viewed below in Figure 6.3.1.



**Figure 6.3.1: Final Code Flow Chart**

The Parallax Propeller program will be able to handle most, if not all, of this functionality. The LINX transceivers use a UART program to handle data being sent so it will be necessary to create a program that detects the input buffer for data at all times within a loop. When the input buffer is detected to be full, the READY pin will get a high voltage and block all other incoming data until the buffer is cleared. The UART method usually means that only one character can be transmitted at a time, so the entire message, which could be anything from volt, current, power, price, or carbon emission values, must be saved one number at a time in a register, careful to maintain the value of the message and not add to it or take from it in any way.

These values must then be sent to a display function. The display function, configures the data to values that show the equivalent value on the screen. It appears inefficient to convert values that were already converted and calculated back into a coded version, but it truly is the only way to ensure that the data being displayed is the same data that was sent. The reason that the display values are different from the actual value is because there is a different number, usually in hexadecimal, required to activate each pixel of the screen. The value of a 1, for example, becomes the value of changing all of the pixels required to represent a 1 onscreen. In this manner, all of the sensor's data must be reconfigured to the screen's visual requirements which are the entire purpose of the display function.

Checking for, receiving, and configuring data from the sensors isn't the only part that the hub handles. Simultaneously, the program is also actively waiting for a user to touch the screen. Upon touching, the program will detect where exactly the screen has been



touched and determine if that location is available for interaction. There will likely be buttons on the screen to change what sort of data is being shown, to change which device the screen should focus on, and to possibly other options could be available like shutting off the power of objects that are consuming too much power. For each button, a function would be present to alter certain values that are reflected onscreen. To change the data being shown, possibly the data that gets displayed lays in an array and the user simply advances the array to the next index whenever the button is touched. The array could then loop back around to the beginning whenever the end is reached in this way. Changing the device that is being shown is slightly trickier, as it entails sending a message to the transceiver to change the channel of incoming information using the CHN\_SEL lines available to the LINX. Once the channel has been changed the code should still operate as normal, only with another sensor as the sender whose inputs are being accepted.

The hub doesn't do much in the way of calculating things aside from calculating display values. The proper power consumption values are sent to the receiver first and it is there that the values are converted, that is where the second program comes into play. The transceiver program mainly verifies what is being sent between them and then calculates the power, carbon footprint, and cost of operating the device being tracked. Once the values are calculated the data is fed into the hub via the DATA\_OUT pin from the LINX transceiver where it can be processed for display purposes.

Optional functions could be made if the feature is available. For instance, the ability to shut off power to one of the sensors, and therefore, shut down the device itself, is a feature we would like to implement. It would add an option on the screen, which would of course also mean that there would be another function to meet this need. The option to shut down may be always present, or it may only be made available when certain requirements are met such as if the user creates a budget, and the cost of operating a device exceeds that limit. Creating a profile itself is another option and another function which could add the ability to save. Creating a save could be done if the program has the ability to create files. It would be a simple task to take all of the values currently gathered and drop them into a text file where it can be read the next time the system starts up. This may require attaching the hub to a computer or perhaps the hub will have a small hard disk of its own in order to keep data while it is turned off.

The coding of this project will not be done by just one person, so there will be some coding etiquette that will need to be observed in order to smoothly integrate all of the different functions of the program. Function names for example, will be completely spelled out as opposed to abbreviated in order to be clear as to the usage of the function. Every word within the function name should have a capital letter as the first letter. This is different from variable names that start lowercase and have subsequent words have a capital first letter. This is called camel-ing the names, and it helps other programmers understand another programmer's work by being very unambiguous. Another piece of etiquette to follow is proper indenting and managing of white space. When for loops, while loops, and conditional if statements start to be used, it can be very confusing to keep track of what line is within what loop without proper indenting. The highest level loop has the same indentation as the rest of the main code. Every level of loop or nested statement must maintain their own indentation. The further indented a line is, the more

nested it is. This makes finding the central section, the point in which all of the loops circle around a much easier task.

No matter what language the chips may use, the format of functions must be the same. If the function makes use of any registers or variables used elsewhere in the code, the first step of the function would be to place those values within the stack and the last step must be returning those values to where they were. Functions only need to do their job and nothing else. If a function alters a variable without warning, it could lead to serious problems from simple errors to crashing the entire program. The function list for the central hub is as follows: a data read in from the transceiver, an identifier for each sensor and assignment of names, a display converter for data to go onscreen, a communication tester that creates a success or failure message for the user, a function that finds the location of the finger of the user and sends the program to the proper function if an option is pressed, a function for each option and feature, a function that changes the channel frequency, and a selector for which data to be displayed. The transceiver also has a few functions that include: checking for data to be received, transmitting data, calculating power consumption values, and sending acknowledgment to the hub during start up.

## **7.0 Project Prototype Testing**

Utilizing the designs discussed a prototype is constructed and tested. This section will discuss the testing cases for each subsystem in the unit, which includes the VGA Display, Kilowatt Hour Meter, Radio Transceivers, and the Automated Circuit Breakers. The tests will be broken down into hardware and software to ensure that all angles and directions are accounted for. Each component within their respectful setting will be tested to ensure prevent faulty pieces.

### **7.1 Hardware Specific Testing**

#### **7.1.1 VGA Controller:**

The hardware will be tested in units during the installation process and further progress will not be made until each step in the process is complete. There is no reason to solder every part to the board at one time before testing the more essential parts along the way. This will also greatly reduce the time to complete the task as troubleshooting can be quickly narrowed in on the part most recently installed.

Making sure that the hardware works properly before software testing is essential. As previous diagrams have shown the VGA board design (Figure 4.2.1.18) the power supply needs to be tested first. The reason this should be tested first is that the hardware will be damaged if this is not correct. The first component to be soldered onto the board will be the DC (RASM732X) connector and voltage and current will be measured from this connector using a multimeter. A AC to DC adapter will be used to plug into the DC

connector to convert the electricity coming in from the wall. A 12V adapter should show 12 volts on the other side of the DC connector. If this does not occur, a new adapter will need to be purchased/exchanged as it is probably damaged.

The next step is the integration of the Texas Instruments TPS5423D switching regulator. This regulator will take the 12V from the DC connector and step the voltage down to 5 volts and the current to 0.3 amps. This will take some time to do as there are many components to be soldered into the board including seven capacitors, five resistors, a diode, and an inductor. This is in addition to needing to mount the switching regulator into the board as well. After all of these components have been mounted a multimeter will be used on  $V_{OUT}$  to make sure the voltage and the current are where they were expected to be. When this information is verified the voltage divider on the end of circuit will be added to  $V_{OUT}$  to drop the voltage down to 3.3 volts to supply power to the many components that will need it. This voltage like before will be verified with a multimeter before any new connections are made.

Following the voltage switching regulator assembly will be the microcontroller. Figure 4.2.1.12 shows a few connections to ground and a few connections to the 3.3 volts. These connections will be made, but before any power is supplied to the board again the serial connection will be assembled because this will be the only way to test if the microcontroller is working. The serial port will require four resistors being added to the board, a capacitor, two diodes (2N3906 and 2N3904), and the appropriate 3.3 volt and ground connections made. When the serial port is wired it can be connected to a personal computer for testing. A program can be written to send data back to the serial terminal to verify that it is working. If no data is sent back to the terminal when the program is loaded, it is a good indication that a connection was not made properly.

Provided the previous steps have successfully been completed the EEPROM should be wired to the board and to the microcontroller. A resistor will be needed in addition to wiring the memory to 3.3 volts and ground. Testing this device will be similar to before, but the software will be loaded into the EEPROM space from the computer and a message be sent back. Again, if the message is not received, either a connection has not been made properly or the ROM in conjunction with the microcontroller has not been configured properly.

Next to be wired to the board is the VGA DE-15 connector with the required eight resistors. These will be wired with the microcontroller pins and then tested. A standard VGA screen will suffice and a demonstration program provided by Parallax should be ample enough to see if it is working. It will be as simple as seeing a picture on the screen or not seeing anything. If a picture is not seen then the resistors should be checked as well as the connections to the microcontroller.

The TE Connectivity FFC 84981-4 is a flat flex cable connector that will take the touch data from the VGA screen and send it to the microcontroller. There are 4 pins from this cable but a total of 6 on the microcontroller will be used because of the need to convert analog to digital data. The analog to digital converters will be soldered in one pin at a time. There will be two converters total but one will be tested before moving onto the other. This will require soldering in two capacitors and a resistor for each converter. When one has been verified the other will be soldered and the connection made to the microcontroller. The conversion will be tested using software and sending that data to the serial terminal.

Finally the last component to add to the design is the Linx RF transceiver. Figure 4.2.1.17 shows that it will require many connections to 3.3 volts and many connections to ground. In addition to these connections, there are a few to make to the microprocessor as well as soldering in a DIP switch to select the channel and a resistor as well. This will be the most difficult unit to test as it will require another function board that has been set up properly to know that this component has been set up. A way to get around this would be to purchase a development board that has already been set up to send to the transceiver on the VGA board as well as receive data from it as well. This should make it easier to test.

### **7.1.2: Kilowatt per Hour Meter**

Testing each component of the kilowatt per hour meter one step at a time is essential so that none of the other hardware components become damaged. If it is seen that one component fails to function properly, replacing the faulty components as testing progresses will cause prevent any future damages.

The first task in testing the Kilowatt per Hour Meter for proper functionality is to ensure the meter's power supply is working. Each kilowatt hour meter requires two power supplies to power the two microcontrollers. The MSP430AFE will utilize the Resistor-Capacitor power supply while the MSP430F6638 will be utilizing the Switch-Based power supply due to its requirement of powering a radio frequency transmitter. The first step is to ensure the RC power supply functions properly using a multimeter. Once the RC power supply is tested, the switch-based supply needs to be checked so that the power switches when the load is applied to the circuit. In other words, the power supply will switch to power the radio frequency transmitter when needed.

Since the Analog Front End of the 24-bit Sigma Delta Analog Digital Convertors is only capable of withstanding a voltage input range of  $\pm 500$  mV, this is the next step required in testing. To ensure the hardware isn't fried and that the components stay undamaged, it is essential to make sure that the 230 V main is diminished to under 500mV. This is done by connecting a series of resistors to act as the voltage divider which is then fed into an RC low-pass filter. The values of the resistors and capacitor for this method can be seen

back in Figure 4.3.3 in section 4.3.1. With the selected schematic/layout, the peak voltage should be less than 495 mV. This is tested by applying an input of 230 V and measuring the output using a multimeter. If while testing the peak voltage is higher, then recalculate the necessary changes in resistor and capacitor values to ensure this threshold is not breached.

Next is to test the current inputs for the analog front end of the SD24 ADC. Following the same constraints as the analog front end for voltage inputs, the current input resulting voltage reading must not breach the  $\pm 500$  mV threshold. Referring to Figure 4.3.4, a setup similar to the voltage input is set up utilizing anti-aliasing filters and low pass filters.. Once these components have been soldered onto the board, a test similar to the voltage test should be run.

Now the next step is to ensure that all the components on board are functionally properly. First will be the LCD screen which can be tested by powering on the unit. If the screen blinks with any sort of data, the unit is functioning properly. Next is to send data wirelessly using the radio frequency transmitter. This will be a little more challenging to test since it requires functioning software components as well. But by utilizing an already existing program and receiver, this task can become a little simpler. If the receiver detects the transmitter in any way then the RF transmitter is all set.

The final step is to connect the kilowatt hour meter to a main line and ensure that it is properly reading the power consumption. This process might be a little difficult as well without an already function unit. By using a preexisting unit like one mentioned in section 3.1, this task will merely become a simple plug and check. If the reading is off then the issue is software based, but if no reading is being made then a check on the previously discussed components will need to be made once more.

### **7.1.3 Radio transceiver testing**

Testing every component will be important to determine the success of this project. The radio transceivers are no different in this case. In order to properly test them, a series of test cases could be made. The first test would be a simple communication between the two. Perhaps one transceiver could be hard coded to send a message, and the other simply raises a flag to make us aware that it got the message. If there is a light or switch on the development board, the message could simply change the value from off to on. Once we can confirm that the two chips can communicate, we can move on to further testing.

The next test would be to find the actual range of the transceivers. The manual claims to have a 3000 foot range with line of sight, so we should test that. It would require finding a large open field, and if we can get a signal 3000 feet out we can verify that claim. If

not, we will need to find the actual limitations by bringing the transceivers closer and closer together until a signal is acquired. This would only get the line of sight range however, and because this project is mostly for indoors operations, we would have to find how far the range is while the transceivers are obscured by walls and furniture. These ranges would likely vary depending on the thickness of the walls, or the materials that the signal would have to penetrate through.

After discovering the range for indoors and outdoors, we will need to experiment with the baud rates in order to determine which rate would be the best choice for sending information. This will largely be determined by how fast the data can be transmitted and interpreted. If the data is sent too fast, the receiver may miss data being sent and some values can be skewed or incorrect entirely. To test this we could create a repeating pattern to be sent from one end to the other and change the rates each time to see the results. If the pattern isn't exactly the same, as in some characters may be missing or doubled up, then we would know that the rate of data exchange isn't quite right. This is important to get right because the data needs to be sent accurately and consistently.

The last test would be to make sure that the transceivers can send and receive information on each available channel. This could be done simultaneously with the other tests, since we would need to confirm the same values for each channel. After completing the communications test and range test for one channel, we would simply switch to the next channel and reconfirm the results until all 8 channels have been tested.

## **7.2 Software Specific Testing**

Software testing of the transceivers would of course be necessary, as learning a new chip set always has its problems at the start with bugs simply due to unfamiliarity with the language. This could lead to memory hiccups and mistaken values saved in registers. To prevent this we need to do regular debugging of the code, checking and double checking that the values that are being shown are actually correct.

The first test would involve checking that the processor detects when the transmitter is ready to send and the receiver is ready to receive by determining the value of the READY pin. When that is confirmed, data can actually be sent. The next test would be to test the accuracy of the data being sent. The LINX has its own demodulator, but we would need to be sure that the signal being decoded is the actual value being transmitted. This would require the use of a multimeter to verify the numbers and values being sent to the receiver. The values would be held within the registers of the receiver. If the registers' values match those that the multimeter reports, then we can confirm that the radio transceivers are operating normally.

The next test would then be completely programming based. We must be sure that the flow of the code makes sense and that proper coding techniques are used. While debugging the code we have to go step by step into it and follow the logic of all the function calls to ensure that the right information is being sent to the right functions, and

the correct values are being returned from the functions themselves. To this end, we will have to place lines of code during the debugging phase that show the current values of the registers during each of the functions to ensure that there are no errors taking place due to human error concerning code syntax.

The internal functions calculating the values could then be tested with dummy values whose answers we calculated ourselves to verify the functions work properly. The final test of the transceivers would be if the central hub receives the calculated data and if those values match the same values within the registers of the receiver. This test involves the processor of the display, and if the screen processor has the right values sent to it than the chip is outputting correctly and all of the functions are working properly.

## **8.0 Administrative Content**

### ***8.1 Project Milestone***

Project milestones are essential for any successful project. A milestone was created to monitor the rate or work being accomplished throughout the semesters. This milestone was used as a tool to ensure efficient productivity from the group and its members and to prevent any overwhelming amounts of work.

Each section of this milestone was created so that tasks could simultaneously be completed or that one task would begin and build off of the previous milestones. While planning out the semester, time was allotted so that if any unexpected problems or conflicts came about, the timeline would still be achievable without too much of a hassle.

Figures 8.1.1a, 8.1.1b, 8.1.2a, and 8.1.2.b attached at the end of this section show the milestone drawn up for Senior Design I and Senior Design II as well as the task breakdown by week. The data within those charts are discussed in the following sections to come.

#### **8.1.1 Research**

The researching phase of the milestones makes up a majority of Senior Design I. During this phase, group members were assigned to research their respective fields with respect to the project.

Group members Daniel Allen and Richard Velez were charged with the tasks of researching the implementation of the radio transceiver as well as the hardware design/overview of the Video Graphics Array Display. Allen also researched the existing

products already created as well as any relevant technology to assist the group in designing their project. Group member Jonathan Nguyen were charged with the tasks of researching the design of the kilowatt per hour meter and the remote controlled circuit breaker.

In the midst of the hardware research, the group began their research on the software component of their designs. This portion of the research was essential since the primary source of communication for the project was through radio frequency. The first software research task was to discover how to go about wireless communication between the hardware components. This point of the research marked the beginning of the design portion of the milestone.

After all the researching is complete, a budget needs to be drawn up based on the research. This budget is not necessarily set in stone, but it will give an idea of what to look for when the designing phase begins.

### **8.1.2 Design**

In this phase of the project the research should be roughly ninety percent completed. While the research continues, designs should be thought up and drawn out for the already completed research. Group members were responsible for designing their respective assigned research.

Designs are first drawn up for the hardware portion of the project. This is necessary to accomplish prior to the software design since the hardware will determine the software components. At this point of the milestone, a clearer budget should be in the midst of being finalized as the idea of what components will be used becomes more concise.

### **8.1.3 Integration and Implementation:**

Once all the researching and designing has been completed, Senior Design I should be coming to an end, which will mark the beginning of Senior Design II with the integration and implementation phases.

These phases consist of the actual hands on portion of the milestone. Here is when the project construction will begin and all the research and design becomes implemented. The first task is to purchase any and all necessary components discussed in the design. From here the members will follow their designs and being composing both the hardware and software components.



### 8.1.4 Testing

Testing is a phase that should be done progressively throughout the milestone. As each component's design is implemented and constructed, the unit should be tested to ensure functionality is as expected. If testing is only implemented once the project is completed then when an error occurs there will be too many potential directions to choose from to efficiently troubleshoot the issue. The testing methods can be viewed in section 7.0.

Date	Task	
9/17/2013	Submit Initial Project and Group Identification Document	
9/24/2013	Identify different components of project	
10/1/2013	Discuss the addition/removal of components	
10/8/2013	Research hardware components	
10/15/2013	Complete designated Research and Compile Information	
10/22/2013	Research Software Components	
10/29/2013	Complete designated Research and Compile Information	
11/5/2013	Begin designing phase & Final Document	
11/12/2013	Determine which Hardware/Software parts to purchase	
11/19/2013	Finalize Budget/Design	
11/26/2013	Continue Final Document	
12/3/2013	Finish and Review Final Document	
12/10/2013	Review Project design to prepare for Senior Design II	

***Figure 8.1.1a: Milestone for Senior Design 1***

[illegible]

**Figure 8.1.1b: Milestone Task Breakdown for Senior Design**

<i>Date</i>	<i>Task</i>	
<i>1/7/2013</i>	<i>Review documentation from Senior Design I</i>	
<i>1/14/2013</i>	<i>Start finding hardware Components</i>	
<i>1/21/2013</i>	<i>Purchase Hardware Parts/Find Software Components</i>	
<i>1/28/2013</i>	<i>Purchase Software components</i>	
<i>2/4/2013</i>	<i>Begin Construction of Hardware Portion</i>	
<i>2/11/2013</i>	<i>Begin Completion of Software Portion</i>	
<i>2/18/2013</i>	<i>Continue with Hardware/Software</i>	
<i>2/25/2013</i>	<i>Continue with Hardware/Software</i>	
<i>3/4/2013</i>	<i>Begin Initial Testing of Project</i>	
<i>3/11/2013</i>	<i>Review Testing/Make necessary changes/adjustments</i>	
<i>3/18/2013</i>	<i>Test Project under situational stress</i>	
<i>3/25/2013</i>	<i>Review Testing/Make final changes</i>	
<i>4/1/2013</i>	<i>Test Project</i>	
<i>4/8/2013</i>	<i>Test Project</i>	
<i>4/15/2013</i>	<i>Present Project</i>	

**Figure 8.1.2a: Milestone for Senior Design II**

[illegible]

**Figure 8.1.2b: Milestone Task Breakdown for Senior Design II**

## 8.2 Budget and Finance Discussion

Attached at the end of this section is Figure 8.2.1, which is a table listing the components and estimated prices. The project has been approved and was estimated to cost roughly \$1400, and this amount was fully sponsored by Duke Energy.

Items	Price	Total
7" VGA Screen	\$80	\$815
Soldering Tools and Accessories	\$52.94	
PCB & Fabrication	\$450	
Breadboard and accessories	\$5	
Xbee RF Transceivers	\$51	
Parallax Propeller	\$16	
Resistors, Capacitors, etc.	\$120	
Power Outlet and Cable	\$20	
VGA Accessories	\$20	

**Figure 8.2.1: Projected Budget Breakdown**

Please note that the projected budget is susceptible to change for any reason, but the expected total should never go beyond \$1600. This buffer zone is in place for any damaged parts, additional parts, or unforeseen circumstances that may arise during the construction of the project.