

Project Number: FJL-MAGN

PICOSATELLITE MAGNETOMETER

A Major Qualifying Project Report:

submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

By

Danato E. Borelli

And

Justin M. Woodard

Date: March 2, 2006

Approved:

Professor Fred J. Looft, Major Advisor

Abstract

Picosatellites are small, low cost satellites that are designed with the philosophy that sending many small, inexpensive satellites into space is cheaper and more effective than sending one large satellite. The goal of this project was to design a Picosatellite capable of taking magnetic field readings and sending the readings wirelessly to a base station. The team designed, fabricated, and tested a working Picosatellite mock-up to provide a proof-of-concept for the project goal. A base station was also built to receive the satellite data.

For Further Information:

Please contact

Professor Fred Looft (fjlooft@ece.wpi.edu)

Department of Electrical and Computer Engineering

Worcester Polytechnic Institute - 100 Institute Road, Worcester, MA 01609-2280

Index

1) Introduction	1
2) Background	3
2.1) Powering a Satellite	3
2.2) Photovoltaic Cell Operation	3
2.3) Satellite Orbits	4
2.4) Earth's Magnetic Field	5
2.5) Magnetic Field Sensing	6
2.6) Using Magnetic Field Readings to Determine Spacecraft Altitude.....	7
2.6.1) Compensating for Albedo	7
2.7) The Space Environment and its Effect on Spacecraft.....	8
2.7.1) Thermal Effects	8
2.7.2) Plasma	8
2.7.3) Meteoroids and Orbital Debris	9
2.8) Conclusion	9
3) Problem Statement	10
3.1) Problem Statement.....	10
3.2) Specific Tasks.....	10
3.3) Summary.....	12
4) Methods	13
4.1) Power System	13
4.2) Communication System.....	13
4.3) Data Processing	14
4.4) Magnetometer	14
4.5) Physical Constraints	14
4.6) Summary.....	15
5) Satellite System Design	16
5.1) Power System	16
5.2) Magnetometer.....	22
5.3) Transmission and Communication Protocol.....	24
5.4) Processor.....	25
5.5) System Board.....	26
5.6) Physical Construction	27
5.7) Software Design	29
6) Base Station Design	31
6.1) Processor Unit	31
6.2) Power System	32
6.3) Receiver.....	32
6.4) Physical Construction	33
6.5) Software Design	34
6.5.1) Heading Calculation.....	36
7) Testing and Results	38
7.1) Power System Testing	38
7.2) Magnetometer Calibration	39
7.3) Communication System Testing.....	40
8) Summary and Conclusion	41
8.1) Overall Assessment	41
8.2) Future Work.....	41
8.3) Conclusion	42
Bibliography	44
Appendix A: Matlab Solar Cell Simulation Code	45

Appendix B: Satellite Software.....	47
Appendix C: Base Station Software.....	54
Appendix D: LM334 Data Sheet.....	71
Appendix E: LT1300 Data Sheet	72
Appendix F: LM7805 Data Sheet	73
Appendix G: MicroMag3 Data Sheet	74
Appendix H: TLP434/RLP434 Data Sheet	75
Appendix I: TMP35 Data Sheet.....	76
Appendix J: LCDX Data Sheet.....	77
Appendix K: Heading and Calibration Primer.....	78

1) Introduction

A nanosatellite is another name for a very small satellite. Miniature satellites are considered to be those under 200 kg, microsatellites between 10 and 50 kg, and a nanosatellite between 1 and 10 kg (Wertz and Larsen, p.853). The term picosatellite is also often used in referencing satellites of nanosatellite scale or smaller. The small weight means a small size, which also means less area for solar panels and space for internal components. The smaller solar panel area directly affects the satellite's power budget, making designers consider very carefully the power consumption of each component in the satellite.

Nano and picosatellites present a distinctly different method of satellite design than early designers followed. Traditional satellite designers considered each mission separately and then designed every subsystem around the requirements and constraints of the mission. NASA used this method through the late 1980s to produce reliable but expensive satellites. The cost of engineering time, specialized components, and the inability to transfer subsystems between different satellites led to a decline in subsystem-by-subsystem design (Jilla and Miller).

There are two current design methods that have gained popularity with the decline in subsystem-by-subsystem design. The first is Modular Design for Multiple Missions. The modular design approach requires designers to consider the possible range of missions a satellite family could be asked to perform. For example, NASA and Fairchild collaborated on a Multi-Mission Spacecraft Platform. Within the specifications of the platform, each bus contained a nearly identical 1) modular power system, 2) command & data handling module, 3) attitude determination & control system, 4) modular structure, and 5) payload interface (Wilson). The common structure and specifications allow modules to be changed to meet different mission requirements without requiring a redesign of other modules. This reduced the time and money spent on reengineering subsystems to operate together.

The second current design method, and the one the project team was focused on, is the Small Satellite concept. Small satellites are designed to be physically small and lightweight so that they can be launched with much less expensive rockets. The satellites themselves are also designed to be less expensive, meaning lower functionality and redundancy than full-size satellites. The advocates of this methodology assert that launching multiple small, less capable, high risk, low cost satellites to perform a mission will in the long run prove cheaper than launching a few large, highly capable, overly redundant, lower risk, very high cost satellites (Jilla and Miller).

The disadvantage to the small satellite design is that the lack of redundancy can lead to single-point failures that can render a small satellite inoperable where a full-size traditional satellite would be able to survive. For example, the ASUSat1, a nanosatellite size spacecraft designed and built by students at Arizona State University was launched on January 27th, 2000. The satellite only functioned for 14 hours in space due to a catastrophic charging system failure. Though failure analysis was inconclusive, the team agreed that higher levels of redundancy could have saved the satellite (Friedman, 2000)

This document outlines the design and manufacture of a picosatellite. The satellite is equipped with a magnetometer to read the surrounding magnetic fields. These readings are transmitted wirelessly to a battery-powered base station that displays the information from the magnetometer as well as health and safety information from the satellite. The satellite is solar powered and has a power storage system for periods of orbit where no sunlight is present. The design process and component selection are outlined, system diagrams are presented, and test results are discussed to completely encompass all aspects of the picosatellite project.

2) Background

The background section will discuss the concepts of picosatellites and the Earth's magnetic field.

2.1) Powering a Satellite

Satellites are usually powered by a combination of photovoltaic cells (solar panels) and batteries. The solar panels power the satellite and charge the batteries while the satellite is being exposed to sunlight, while the batteries power the satellite systems when the satellite is shielded from the sun. We will use the assumption that the satellite spends 50% of its time in orbit exposed to the sun, and 50% on the dark side of the Earth. This means that the solar cells must be capable of both replacing the energy drawn from the batteries while in the dark phase of the orbit, and also powering the satellite. Simplified, this means that the solar voltaic system must be able to deliver at least twice the current required by the payload while in the sun.

For an example, assume a satellite with a 90 minute orbit and light and dark phases. If the satellite requires 30 mA of current to operate the electrical components, then the solar cells will need to provide 30 mA to operate the satellite and an additional 30 mA to charge the batteries. This is after considering any regulator or converter efficiencies.

2.2) Photovoltaic Cell Operation

Most photovoltaic cells in use today are made of silicon. Each cell is essentially a wide, flat diode, with the negatively doped portion of the diode facing the sun. Solar radiation strikes the cell and transmits energy to electrons in the valence band of the silicon molecules. Once the energy of a photon is greater than the band-gap energy of the silicon, an electron is raised into the conduction band. Once there, the electron is subjected to the electric field that exists across the P-N junction of the diode which forces the electron out of the diode where it can be used as energy (FSEC). Silicon is used for photovoltaic cells because the band-gap energy of silicon is such that most of the wavelengths present in solar radiation produce the desired photovoltaic effect. Energy from the photons present in wavelengths of light with too much or too little energy to be used in the photovoltaic process is converted to heat. Typically, individual silicon cells produce a voltage around 0.5 volts. To make a useful voltage, solar cell designers will wire many cells in series. The resulting module is usually packaged as a self contained solar cell unit that operates at a rated voltage (Driesse p. 23).

2.3) Satellite Orbits

There are two different types of satellite orbits that are commonly used. The first is called geosynchronous orbit. In a geosynchronous orbit, a satellite orbits the earth once per 24-hour day. This means that the satellite will remain over the same line of longitude that it starts over. Geosynchronous satellites always rotate around the equator, which allows ground based stations to aim at the point in space where the satellite position is known to be. In order to maintain a geosynchronous orbit, a satellite must be 22,241 miles above the Earth's surface (NASA). Satellites in this configuration are useful because their location is always known, allowing antennas on earth to be oriented permanently towards their desired satellite with no need for adjustment. The geosynchronous orbit also poses several problems for spacecraft operation. Since the satellite is very high above Earth, its launch cost is high and the rocket payload capability is less than a satellite orbiting at a lower altitude. Also, powerful transmitters must be used to reach Earth, which require a large amount of power from the satellite's on-board power budget. The long distance that communication signals must travel also adds a propagation delay that must be compensated for (Ganz, Gong, Lee p. 1866).

The second common type of orbit is Low Earth Orbit, or LEO. Satellites in LEO orbit the Earth at an altitude between 300 and 1250 miles. The lower altitude reduces receiver-to-transmitter distances when compared to a geosynchronous orbit. The smaller ranges also allow the use of communications equipment that consumes less power and the cost to launch satellites into LEO is less than a geosynchronous launch (Uzunalioglu, p. 89). The biggest difference between a satellite in LEO and a satellite in geosynchronous orbit is the way that they communicate between each other and Earth. Geosynchronous satellites have a stationary position relative to a point on the Earth, so aligning communication systems with them is relatively straight forward. LEO satellites move in relation to the Earth's surface, and are much closer to the Earth, which means that they are not always in the same place and cannot always cover the desired area on their own. To compensate for this, LEO communications satellites form constantly-changing networks used to cover greater distances on Earth. This system is referred to as a multi-hop communications system, because the signal is not bounced directly from the transmitter to the receiver, but rather it goes through any number of intermediate satellites (Ganz, Gong, Lee p. 1867). A representation of a simple multi-hop system is shown below.

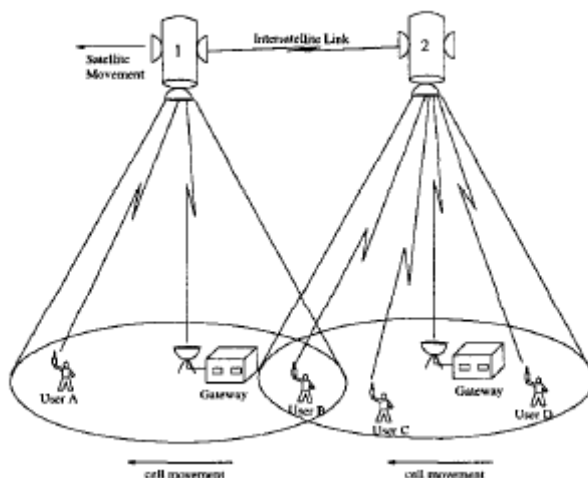


Figure 1: LEO Satellites using Multi-Hop Communication (Uzunalioglu, p. 89)

2.4) Earth Magnetic Field

The magnetic field from the Earth's surface to less than three Earth radii into space can be approximated to the first order by a dipole whose axis is tilted at about 11.5° from the Earth's spin axis and offset slightly from the geographic center. This results in a location for the north geomagnetic pole in Northern Canada at geographic coordinates 73°N , 100°W (Leveson). These coordinates are often referred to as "magnetic north."

Between 50 and 1500 km above the Earth's surface is a region called the Ionosphere. Particles in the ionosphere can be affected by the rotation of earth and the tidal forces of the moon and sun. The ionosphere is broken up into three smaller parts that are distinguished by their different particle actions. These are the D region (50-90 km), the E region (90-120 km), and the F region (120-1500 km). In the D region, neutral particles rotate in unison with the Earth due to the very low degree of particle ionization. In the E region, ionization is higher which causes a partial decoupling between ionized particles and neutral particles. The F region is modeled using an approximation that states that there is complete decoupling between neutral and ionized particles. The motion of particles in this layer are controlled by processes occurring in the magnetosphere and solar winds (Merrill, McElhinny, and Mantle, p.61). The magnetic field is much less consistent in the ionosphere than near Earth, due to the interaction of the geomagnetic field with the neutral and ionized particles, as well as solar activity. The magnetic field in the ionosphere is greatest at the poles, where the field lines converge. The weakest area is off the coast of South America and is called the South Atlantic Anomaly.

Above the ionosphere is the magnetosphere. It is a combination of Earth's dipole type magnetic field and particles being released by the Sun. The Earth is constantly being bombarded by these charged particles. The charged particles cause the phenomena known as Solar Wind, which has a speed range of 270 km/s to 650 km/s, depending on what portion of the sun the earth is currently facing (Merrill, McElhinny, and Mantle, p.58). The charged particles interact with the geomagnetic field of the planet to form a comet-tail type of field. The diagram shown below shows the shape of the magnetosphere and names the different regions.

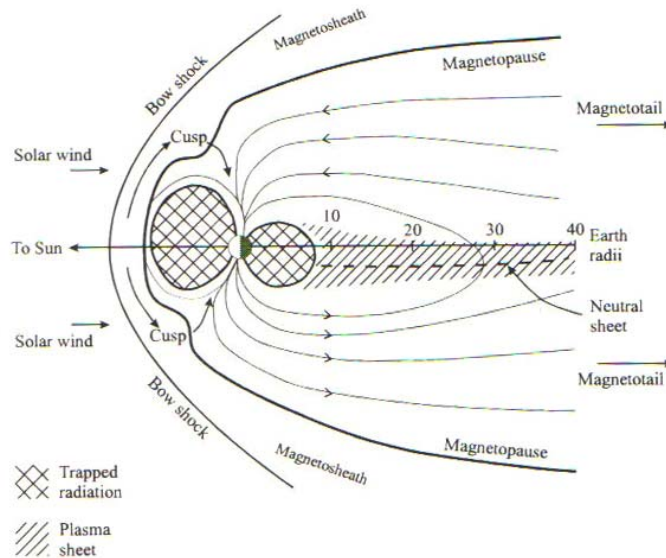


Figure 2: Magnetosphere of Earth (Merrill, McElhinny, and Mantle, p.59)

2.5) Magnetic Field Sensing

A magnetometer is an instrument that is used to detect and measure magnetic fields. The newest magnetometer designs use a magneto-inductive sensor to provide field information to an onboard processor. Magneto-inductive sensors are a type of sensor that changes inductance as the magnetic fields acting parallel to the sensor change. The sensor is a single winding coil on a ferromagnetic core that changes permeability with relation to the Earth's field. The sensor coil is the inductance in an L/R relaxation oscillator. The frequency of the oscillator varies with the field being measured. The frequency can change by as much as 100% as the sensor is rotated 90 degrees from the applied magnetic field, which gives the sensor a wide sensing range (Caruso, et al). These sensors function in only one measurement axis, which necessitates one sensor unit for each axis that needs to be measured. A representative circuit is shown below.

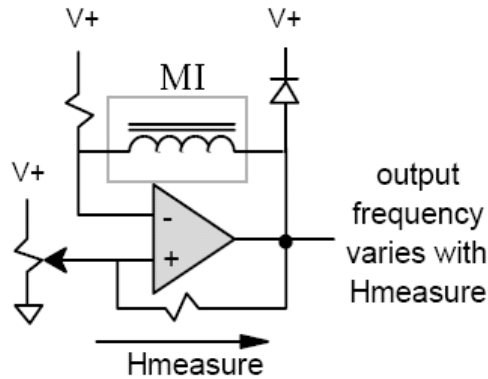


Figure 3: Magneto-Inductive Sensor Circuit (Caruso, et al)

The oscillator frequency can be monitored by a microprocessor's i/o port to determine the value of the sensor reading. Once the magnetic field strength is calculated, the data can be transmitted from the microprocessor onboard the magnetometer to the system processor for further manipulation or transmission. (PNI p. 1-3)

2.6) Using Magnetic Field Readings to Determine Spacecraft Attitude

A spacecraft in low earth orbit (LEO) can use the Earth's magnetic field and a sun sensor to determine its orientation. This is accomplished through the use of a magnetometer, which reads magnetic fields in three dimensions, and a sun sensor placed on each side of the satellite. The sun sensor is simply a small solar panel. A reading is taken from the magnetometer, which defines the direction that the magnetic field is currently pointing. At the same time, the satellite processor is reading the amount of power being produced by each sun sensor. This is compared to a reference value obtained experimentally that shows expected current when the cell is at a certain angle to the sun. The sun angle results are combined with the magnetic field readings to calculate the orientation of the satellite (Theil, Appel, Schleicher p. 1-3).

2.6.1) Compensating for Albedo

Albedo is the fraction of incoming sunlight that Earth reflects back into space. If Earth was covered in ice, its albedo would be about 0.84, meaning it would reflect 84% of the sunlight that hit it. On the other hand, if Earth was completely covered by a dark green forest canopy, its albedo would be about 0.14, meaning only a 14% reflection ratio. Satellite measurements made since the late 1970s estimate Earth's average albedo to be about 0.30 (Allen).

Albedo affects the algorithms used in the process discussed in section 2.3. The reflection of light from Earth can approach 90% over the ice caps. Therefore, the directional algorithm must contain a correction to compensate for the reflected light from the Earth. This was accomplished by experimentally obtaining a three dimensional curve that plots the scaled light intensity against latitude and longitude. This curve was then mathematically modeled to a close fit, and the resulting equations were programmed into the satellite. The satellite now has the ability to calculate in real time its heading and orientation, while taking into account the error that could possibly be induced by albedo (Theil, Appel, Schleicher p. 3-5).

2.7) The Space Environment and its Effects on Spacecraft

The space environment poses unique risks to equipment not present on Earth. Some of these harsh environmental effects will be discussed in greater detail in this section.

2.7.1) Thermal Effects

Spacecraft are affected by three different natural sources:

1. Incoming solar radiation.
2. Reflected solar energy, also known as albedo, which was discussed in detail in section 2.6.1.
3. Outgoing longwave radiation (OLR) emitted by the Earth and atmosphere (Bedingfield, Leach 8).

It should be noted that while the Earth and its atmosphere are on average in an incoming solar energy/outgoing radiant energy balance. When designing for thermal considerations, the designer needs to take into account the difference in reflectivity found over different portions of the Earth and plan for worst-case scenario. Thermal control is essential for successful satellite missions. To maximize the spacecraft reliability, changes in temperature need to be minimized, to decrease component fatigue. Cooling for heat-sensitive components need to be considered, as well as proper cycling of the coolant to avoid a freeze-up. Also, lubricants need to be properly selected to allow all moving parts to function appropriately in the hot and cold extremes of space (Bedingfield, Leach p. 9-10).

2.7.2) Plasma

In the upper portions of Earth's atmosphere, radiation from the Sun causes photochemical effects on the present gases. A common effect is that atoms will have negatively charged electrons ejected from their outer shells, leaving a positively charged atom, or ion. This process is called ionization. An ionized region of gas composed of equal numbers of positively and negatively charged particles is called plasma. When spacecraft fly through plasma clouds, it is possible that they will hit an uneven number of electrons

and ions. This can lead to a large accumulated charge on the spacecraft, which can cause serious electrical malfunctions, to the point of arcing and sparking (Bedingfield, Leach p. 13).

2.7.3) Meteoroids and Orbital Debris

Meteoroids and orbital debris both present the possibility of physical damage to the exterior surfaces of a spacecraft. Meteoroids are small pieces of comet that break off and hit the Earth's atmosphere, where usually they burn up. Small pieces of matter from the asteroid belt are also drawn into Earth's orbit by the gravitational pull of the sun. In addition to natural debris, the amount of man-made debris continues to grow. This includes booster stages from rockets, broken off pieces of spacecraft, and decommissioned satellites. At the speeds required to orbit the Earth, the impact of a 90-gram particle will deliver over one MJ of force to a spacecraft. Smaller objects can cause surface erosion, much like the effects of sandblasting. This erosion can greatly reduce the effectiveness of solar panels and other external components. Debris can be planned for, since the spacecraft will only be hit in one direction. Shielding can be applied to the exterior of the spacecraft, or critical components can be placed far from likely impact sites. High strength materials must be used to dissipate the energy of a potential collision (Bedingfield, Leach p. 16-17).

2.8) Conclusion

This section discussed various aspects of satellite operation. It also provided information on the Earth's magnetic field and the space environment.

3) Problem Statement

The purpose of this chapter is to present the problem statement.

3.1) Problem Statement

The goal of our project was to build a small satellite equipped with instrumentation to measure magnetic fields. The data would be transmitted wirelessly to a base station where the data was to be displayed in a human-readable format.

Objectives

The objective of our project work was to design a two-part system:

Part 1) Design and build a self-powered picosatellite that houses a tri-axial magnetometer for the purpose of taking pitch, roll, and heading data, as well as a wireless transmitter to relay the data over short distances.

Part 2) Design a base station to receive the wireless transmissions of the satellite. The base station will also display the data it receives.

3.2) Specific Tasks

To reach the final project objectives listed above, we focused on five specific objectives and the supporting tasks that enabled us to complete each step.

Power Management

Our problem statement specified that the satellite would be self-powered. To meet this specification, we proposed to design, build, and test an efficient power system to take the solar cell output and regulate it to the voltage that the satellite components need. To complete the power system, the following tasks were addressed:

- Test the power production of the available solar panels.
- Create a simulation of the power consumption of the satellite.
- Determine component power requirements.
- Research power storage options.
- Design the power system

Communications System

For the satellite and base station to communicate, a wireless transmission system was designed, built, and tested. The transmission system consists of a transmitter module in the satellite and a receiver module in the base station, as well as supporting circuitry needed to process the incoming and outgoing data. The following tasks were addressed to allow the building of a communication system:

- Research available RF transmitters
- Decide on the communication type and protocol (i.e. AM, FM, etc.)
- Set up packet structure for data stream
- Research antenna type and placement for the satellite and receiver

Data Processing

Our satellite and base station both required a low-spec processor with low power consumption. The processing solution must be able to receive data from the magnetometer, process it, add in health and safety indicators, and then output the data in a format understandable by the transmitter. The tasks addressed to complete this step are as follows:

- Learn to program in the basic programming language
- Develop a program algorithm for satellite to base station communications
- Decide what data will be transmitted from satellite to base station

Magnetometer

The magnetometer senses the magnetic field on three axes, and converts the values to an SPI data string. Our group received a magnetometer to use, so the efforts in this area were focused on making the magnetometer interface correctly with our microprocessor. A system that communicates with the microprocessor was designed, built, and tested. In addition to the communications protocol implementation, we also decided upon the desired data rate and resolution to allow for the best ease of transmission.

Physical Construction

Our satellite's physical design was constrained mainly by the solar panels that the power analysis indicated should be used. Once this decision was made, our objective was to construct a satellite that houses the instrumentation and power regulation internally, and has solar panels on its faces for power. Building the satellite involved the following tasks:

- Decide on the size of solar panel needed.

- Research and choose a frame material.
- Diagram the physical layout of components for PCB manufacture and design the PCBs.
- Arrange the components inside satellite frame for best fit and ease of access.

3.3) Summary

Our group has laid out the objective of the project, as well as the specific tasks necessary to complete the project.

4) Methods

This section will explain the procedures used to move from the tasks presented in section 3.2 onto beginning to design and test the project. It is divided into the five main categories used in the previous section.

4.1) Power System

The first step to defining the power system was learning through experimentation how much power is available to use. This was simplified for the group because we had been presented with only two solar panel options. To calculate the total power availability, each panel was tested in natural sunlight at different angles, with voltage and current measurements being recorded for each angle. Once this data had been accumulated for each panel, a simulation was coded in Matlab that simulated the rotation of a satellite covered in each kind of solar panel to show the worst-case and average power available for each type of solar panel.

Once the total available power from each type of cell was determined, the amount of power needed was calculated. This included the power required by the magnetometer, wireless transmitter, processor, and any other equipment onboard the satellite. Efficiencies of any buck or boost converters were also taken into account. The amount of power needed was used as a guideline when deciding whether to use a battery or capacitor based energy storage system.

Finally, a spreadsheet was constructed that subtracts all of the needed power from the available power. If there was not a margin of safety, different components would have to be chosen. Power cycling was also taken into account, since the satellite will spend 50% of its time operating in darkness. After all parts were ordered and received, the system was assembled on a protoboard to ensure that the power available matches the power requirements. Upon satisfactory testing, PCB design was carried out for the power system.

4.2) Communication System

The communication system was a low power, short range transmission system. Data on readily available commercial systems was collected to find transmitters capable of transmitting from 75-100 yards. Licensing requirements were also collected from the FCC based on frequency bands utilized by the transmitters. A pair that features a simple "pin-in to pin-out" interface was preferable to interface the transmitter and receiver with their respective PICs. Power consumption specifications were then used to select a transmitter/receiver pair. Testing was conducted to ensure that the selected products function as desired, and then the transmitter/receiver were integrated into the circuit board design for both the base station and satellite.

4.3) Data Processing

The data processing system was built around a Basic-X microprocessor. The amount of memory available for program instructions and processor speed was evaluated for processor choices compared to the amount of data transmitted from the magnetometer. The processor was required to have enough memory to buffer readings from the magnetometer for short periods of time. The processor was also required to have an RS-232 serial port for simple communication with the magnetometer and wireless transmitter. Power consumption was a major selection criteria. After the processor was chosen, a software program algorithm was designed. This was accomplished by listing all of the tasks that the processor must do, then ordering them in a high-level block format placing the tasks in the correct orientation with any looping or interrupts. The software was then coded and flashed to the PIC, where it was tested and revised. The final design mounted the processor onto the PCB shared with the other components.

4.4) Magnetometer

The magnetometer is a specified part. Therefore, the system development and communication with other blocks will be the main focus. We will determine experimentally the amount of data that can be gathered from the magnetometer and at what rate. This figure will be used with the processor speed and memory figures, as well as the maximum wireless transmission speed to calculate the maximum data rate that should be requested from the magnetometer without overwhelming the rest of the system. The magnetometer outputs data through an SPI bus, so we will have to implement the SPI communications protocol on the chosen processor. After construction of the satellite is complete, we will calibrate the magnetometer to ensure the highest possible accuracy.

4.5) Physical Construction

The physical construction of the satellite began by defining the solar cells. This made the selection of a frame shape trivial. Frame materials were chosen upon consultation with the ECE shop for the best strength to weight ratio, with cost and availability also playing a role in the decision. A non-destructive way to mount the solar panels was researched. The frame was fabricated, and solar panels mounted. Once component selection and testing was finished, the PCB design began. The PCBs were mounted on a shelf inside the satellite, with separate boards for the power system and the rest of the components. The base station was constructed in an available enclosure, with storage for batteries to be inserted. A board was designed to link the base station components together. A front panel was fabricated to provide a mounting point for the LCD and keypad.

4.6) Summary

This section presented the methodology that will be used to design our project.

5) Satellite System Design

This section will discuss the design choices made for our satellite.

5.1) Power System

The first step in designing the power system was considering the available solar panel choices. The group was given two panels to choose from. The table below shows the experimentally obtained power figures for each of the panels.


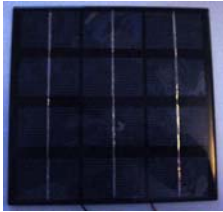
Solar Cells	A	B
Size (W * H * D) in	4.9375 * 2.25 * .0625	6.75 * 6.5 * .125
Max Power @ 5V	430 mW	2W
Current (SC)	.107A	.5A
Voltage (OC)	6.6V	7.10V
Weight (g)	26	136
Photo		

Figure 4: Solar Panel Table

After obtaining this basic information, both panels were taken outside and tested at different angles to the sun. Both of the panels responded in similar ways when presented with off-axis sunlight, and were close enough that with the limits of our equipment that we determined them to have the same general characteristics. A table showing percentage of peak power available versus angle is shown below.

Degrees	Percentage of Power Available
90	100
88	97
78	91
64	86
50	78
39	57
30	50

Figure 5: Angled Solar Panel Power Percentage

Since the group wanted a satellite size of approximately six or seven inches across to allow adequate space for internal components, two possible solar panel configurations were devised using the solar cells shown above. The hexagonal configuration uses Solar Cell A from Figure 3 above. Six of these cells are placed on end to form a tall hexagon with no top or bottom. The square configuration uses four of Solar Cell B to make a cube with no top or bottom. Drawings of the two configurations are shown below.

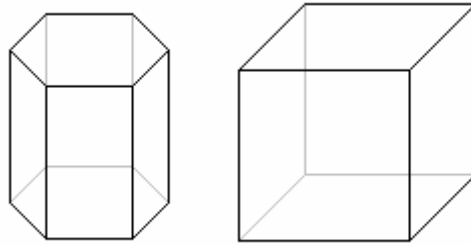


Figure 6: Hexagonal and Square Solar Panel Configurations

From the data taken observing the behavior of the angled solar panels, two Matlab simulations were designed and programmed (See Appendix A). The first simulation assumed the use of Solar Panel A in a hexagonal configuration on the satellite. The program starts with the sun pointed directly at one solar panel, then rotates the satellite along the vertical axis so that best and worst case power numbers can be visually seen. The result of the first simulation is shown below.

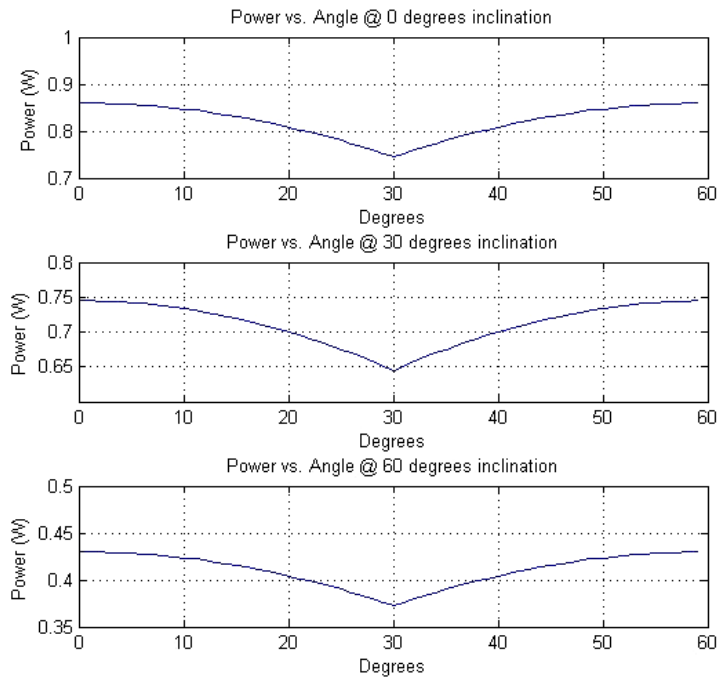


Figure 7: Small Solar Panel Power Simulation

As shown on the above graphs, the best case power is approximately 860 mW when the sun is pointed directly at one panel, and the worst case is when the satellite is at a 60 degree incline relative to the sun, and the sun is shining directly between two panels, which gives a power output of 370 mW.

The second simulation uses Solar Panel B in the square configuration on the satellite. The results of this simulation are shown below.

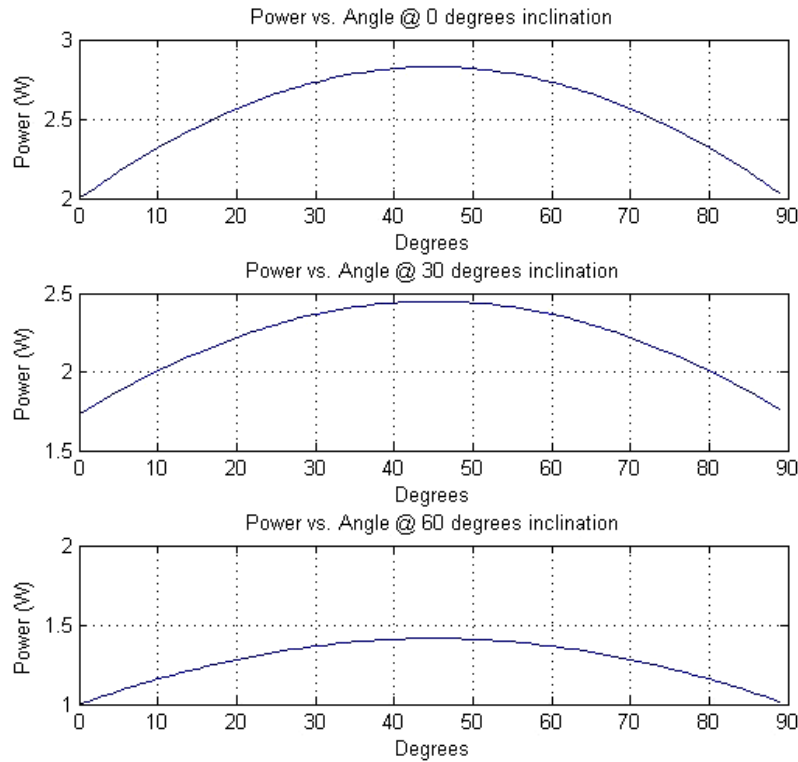


Figure 8: Large Solar Panel Power Simulation

The results show that the large solar panels in a square configuration provide much more power than the small panels in a hexagon configuration. The best case power is approximately 2.8 watts and the worst case power is approximately 1.4 watts. Based on these results, as well as considering that a square frame is much easier to manufacture than a hexagonal one, the group elected to use the large panels in a square configuration.

The next step was to sum up the power needed by the systems we were planning to include in the satellite. The efficiency of the boost converter used must also be considered in calculating the required input power. The table used for this is shown below.

Processor	BasicX BX-24	25 mA
Magnetometer	PNI MicroMag	3 mA
Transmitter	Laipac TLP434A	7 mA
Total System Current		35 mA
Total System Power	@ 5V DC	.175 W
Boost Converter	Linear LT1300	86%
Efficiency		
Total Power Required	@3.6V DC	.203 W
Total Current Required		56 mA

Figure 9: Battery Power Required to Operate Satellite Systems

Once solar panels were chosen and the power requirements were laid out, a power supply was designed. After researching many low-power solar options, the group found a power circuit designed for a wireless weather station, which is shown below.

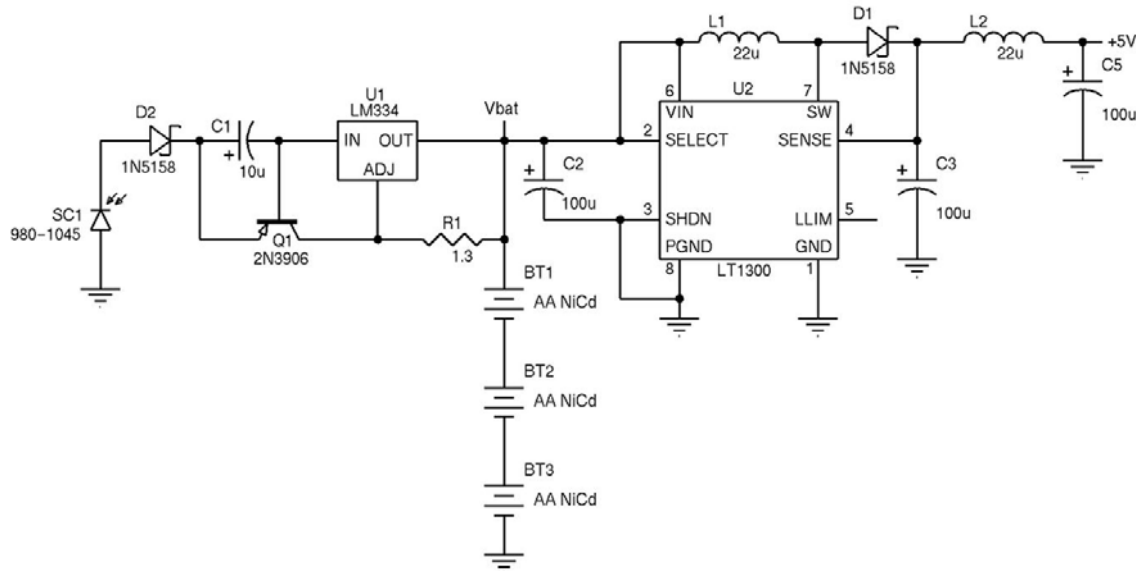


Figure 10: Wireless Weather Station Power System (Wasinger p.36)

This circuit can be thought of as two halves. The left hand side consists of a solar panel, with a diode to prevent a reverse bias condition to the panel. The current from the solar cell flows into the collector of a PNP transistor. The capacitor C1 provides filtering to prevent momentary voltage changes from having a major effect on the transistor loop. The LM334 (See Appendix D) is a constant current source, and it draws approximately half a milliamp of current through the base of the transistor. This turns the transistor into a current amplifier, which causes the output from the solar panel to be multiplied to approximately 50 mA at the emitter. This current acts as a constant trickle charge for the NiCad batteries. The right hand side is a 5V boost converter centered around the Linear LT1300 boost converter IC (See

Appendix E), which is powered by the batteries. The entire system provides a five volt output, but we found it insufficient for our application because it was only able to provide approximately 40 mA of current. With approximately 85% efficiency through the boost converter, and multiple diode drops, we were only left with a maximum of 30 mA of current at 5V from the output of the converter, with no current left over to charge the batteries. This would not work for our system due to our power cycling requirements as discussed in section 2.2.

To solve the power problem, the group used a modification of the circuit shown above. The key difference is that the solar panel we had selected was capable of providing much more current than one specified in the previous circuit. This allowed us to make a few changes and meet our power budget. The revised circuit is shown below.

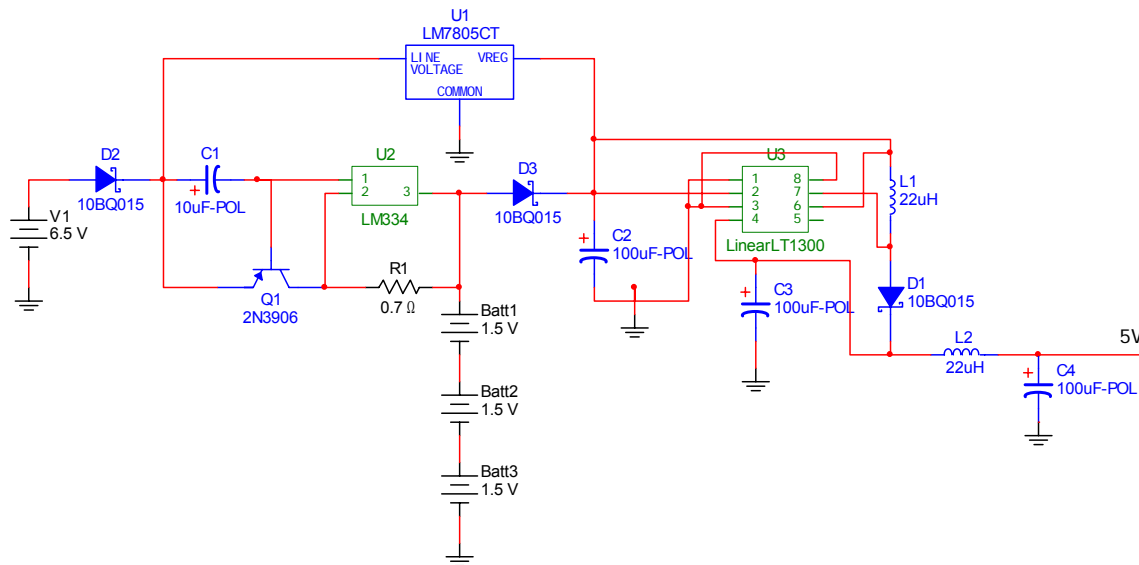


Figure 11: Modified Power Circuit

The first change to this circuit is the addition of a low-dropout voltage regulator, a Fairchild LM7805CT (See Appendix F), which can be seen at the top of the circuit. When the solar panel is producing over 5.5 volts, the regulator turns on and regulates the solar cell power down to 5 volts. Whenever the regulator is on, it provides all the power that the boost circuit needs. D3, which can be seen in the middle of the circuit, prevents current from the regulator from flowing back to the batteries. This allows the batteries to charge while the satellite is in the light without being responsible for providing power to the instrumentation. The value of resistor R1 was the second change. The lower value allows increased current flow through the transistor, which raises the charging current on the batteries to

approximately 80 mA. All of these changes were possible because of the extra power produced by our larger solar panel versus the smaller panel featured in the first circuit.

Following the general power system design, components were chosen to meet the requirements of the circuit. The components were chosen based on price and availability. Surface mount components were chosen when available in reasonable sizes to reduce board space. The final parts list for the power circuit is shown below.

Satellite Power	Model	Type	Digikey part #	Unit Price	Quantity	Total
Schottky Diodes	1N5817	through-hole	1N5817RLGOSCT-ND	0.329	3	\$0.99
Capacitor	100uF	SMD	478-1765-1-ND	1.88	3	\$5.64
	10 uF	SMD	478-1763-1-ND	0.88	1	\$0.88
1/8 Watt Resistors	3 Ohm	through-hole	P3.0BBCT-ND	0.095	1	\$0.10
	1 Ohm	through-hole	P1.0BBCT-ND	0.095	1	\$0.10
Current Source	LM334	TO-92	LM334ZNS-ND	1.21	1	\$1.21
Voltage Regulator	LM7805CT	TO-220	LM7805CT-ND	0.48	1	\$0.48
Boost Converter	LT1300	DIP	LT1300CN8-ND	5	1	\$5.00
Inductor	22 uH	SMD	308-1158-1-ND	0.92	2	\$1.84
Transistor	2N3906	TO-92	2N3906FS-ND	0.18	1	\$0.18
Battery Holder	3 AA		BC3AAW-ND	0.96	1	\$0.96
Batteries	1000 mAh	AA	P252-ND	2.36	3	\$7.08
				Total	\$24.45	

Figure 12: Satellite Power Circuit Parts List

The final step of the design process was the layout of a printed circuit board (PCB). The PCB was designed using software from pcbexpress.com, which allowed PCB layout and also functioned as the ordering mechanism. The board designed was double sided, with through-plated holes on a standard 0.062" FR-4 epoxy glass laminate board, with dimensions of 3.8" x 2.5". Our design allowed for the battery holder to be fastened to the board with bolts, which allows for a completely modular power unit. Four holes, one per corner, are provided for attachment to standoffs for mounting in the satellite. The power board has inputs for six solar panels or power connectors, each of which has an inline diode included on the board for reverse-current protection. The four pin output has a pin for 5V power, ground, battery voltage, and solar cell voltage. The front half of the board design is shown below.

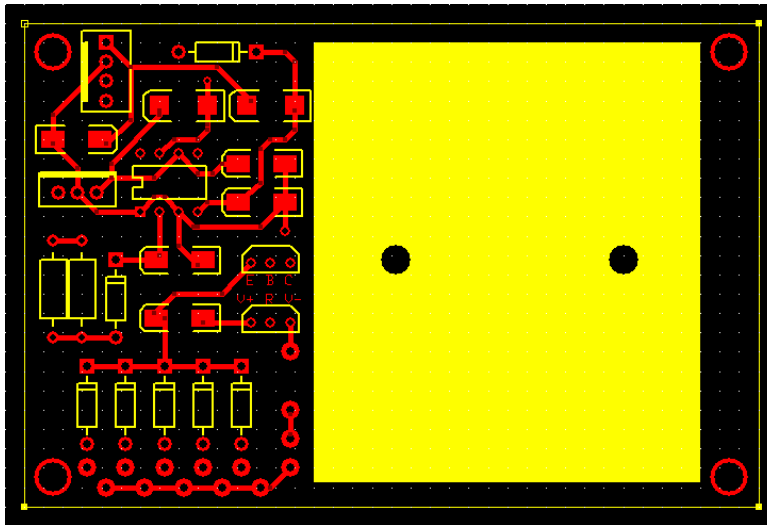


Figure 13: Front Power Board Layout

The large yellow area on the right is taken up by the battery holder. The solar panel connectors attach in the bottom left hand corner at the row of diodes, and then the current flow is generally upwards through the circuit to the four pin header that is seen at the top left corner. All components are mounted directly to the board except the IC in the middle, which has a socket. Final assembly was done by hand in the ECE shop. The completed board is shown below.

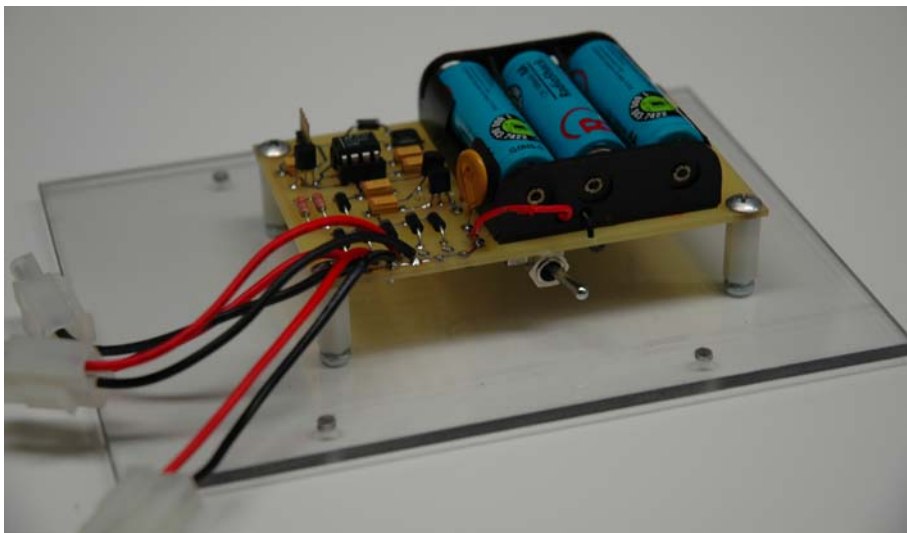


Figure 14: Power Board

5.2) Magnetometer

The magnetometer for this project was a specified device, the PNI Corporation MicroMag3 (See Appendix G). The MicroMag3 is a three axis magnetic sensor module that uses very small and efficient magneto-inductive sensors to provide field strength readings in three directions. The device has the ability to read magnetic fields with values $\pm 1100\mu\text{T}$.

The magnetometer communicates with a processor via an SPI protocol, which is a synchronous serial interface standardized by Motorola. The protocol consists of six signals: SCLK, MOSI, MISO, SSNOT, RESET, and DRDY. RESET resets the magnetometer when toggled high between readings. SCLK is the serial clock, generated by the processor. It is used to keep timing synchronized between the processor and magnetometer. SSNOT is the slave select line, which is always held low by the processor in our application because the magnetometer is always in slave configuration. MOSI is master out slave in, which is the line used to pass data from the processor to the magnetometer. MISO is the opposite, master in slave out, which is used to send data from the magnetometer to the processor. DRDY is the data ready indicator. Once the magnetometer has taken a magnetic field reading, DRDY will be set high to indicate to the processor that data will be output. A diagram of the timing sequence is shown below.

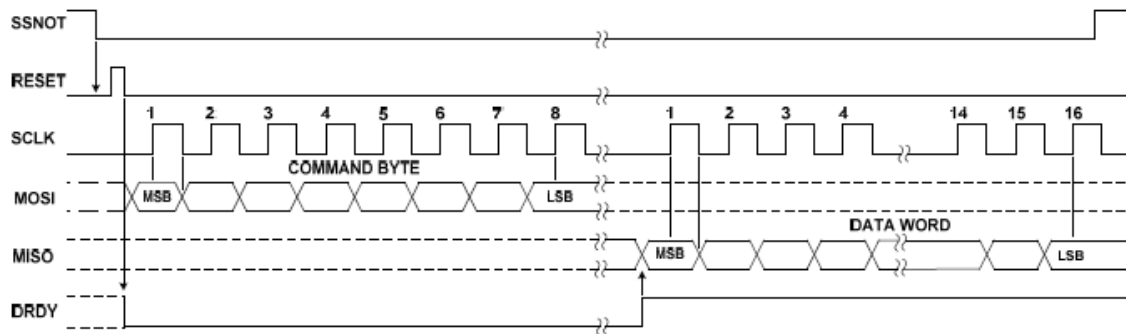


Figure 15: SPI Timing Diagram (MicroMag3 Data Sheet)

The timing diagram above begins by pulling SSNOT low, where it will stay for the duration of the communication process. RESET is pulsed low-high-low to reset the magnetometer. A command byte is then sent to the magnetometer, which includes information about what axis should be read and at what resolution it should be read. Once the command byte has been read by the magnetometer, it will take the indicated reading. Once the reading is complete, the DRDY pin is pulled high by the magnetometer which signals the processor that data is ready and being transmitted.

The MicroMag3 is designed for use with a 3V power source. We considered using another linear regulator to step the 5V power from the power system down to the recommended 3V, but realized that this would introduce complications with logic levels between the processor and magnetometer. Since the processors we were considering used 5V and utilized the basic language, which is an interpreted language, the difference in logic high would make the processor and magnetometer incompatible. To get around this problem, we are powering the magnetometer at 5V, which is below its rated maximum of 5.25V. This is a very small margin of safety, but it makes the logic levels match up for much simpler communication.

5.3) Transmitter and Communications Protocol

At the beginning of the project, the group was given several pairs of low-cost transmitters and receivers. These units used a modulation system called Amplitude Shift Keying (ASK), which is a form of amplitude modulation. The units were inexpensive because they used a simple system where whatever information was received on the input pin of the transmitter was sent out from the antenna, and whatever the antenna received was passed through the output pin of the receiver. The units were only capable of one-way transmission. The decision was made to use the TLP-434 transmitter manufactured by Laipac Technology (See Appendix H). This unit supports transmission speeds of 4.8 KB/s, and the manufacturer recommended the use of a quarter wavelength antenna, which would fit our satellite well.

Initially, the group tried to simply send packets of data to the receiver. This didn't work at all because the receiver considered every bit of random noise that it received to be good data. It was found through research that a communications protocol would have to be written in software. The final protocol consists of a custom header for signal acquisition, and a packet of known data sent at the beginning and end of every transmission. These packets allow the receiver to ignore data sets whose check packets don't match the expected values to reduce the amount of corrupted data that is considered valid. All processing for the receiver is done by the BasicX-35 in the base station. The header used for our protocol as seen by the receiver is shown below.

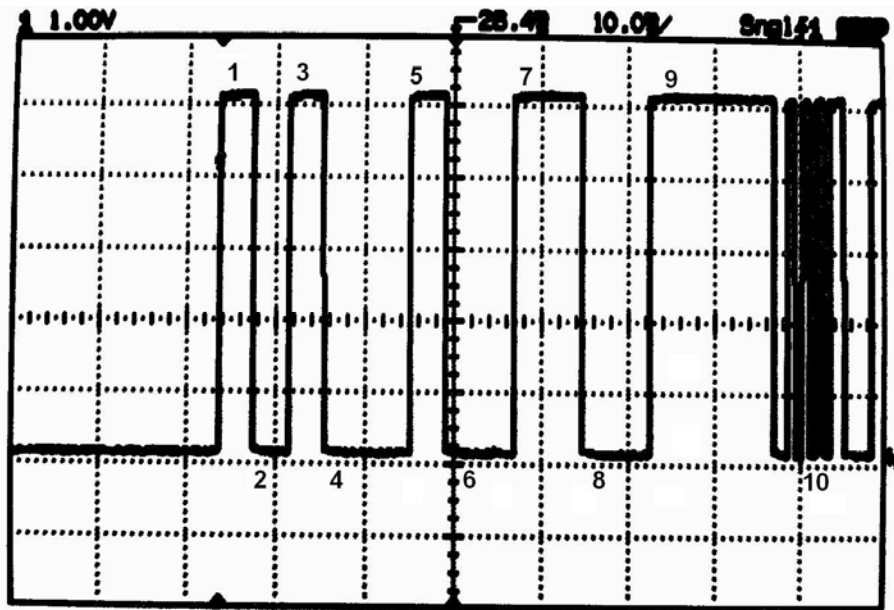


Figure 16: Communications Header

At the beginning of the scope shot, the receiver is placed in a known low state. The receiving processor then begins checking the input pin at rapid intervals. As soon as the processor sees a high, it waits a set amount of time and then checks for a low, which it should see at point two. If the low is found,

the processor waits again then checks for a high which it should find at point three. If this high is found, the processor waits and checks for a low at point four. Once this low is found, an interrupt is set on the processor, and it waits until a rising edge is seen. Once the edge triggers the interrupt, the processor waits a short amount of time and then checks for the high at point five. The processor will then wait and check for the low at six, the high at seven, the low at eight, and the high a nine. If all of these points match with the expected states, the processor clears its buffer in anticipation of the data, which is why there is such a long wait in a high state at point nine.

Point ten shows the first data verification packet, which is 10101010. The processor will throw out the entire data cycle if this packet is not intact. After the verification packet, the X, Y, and Z magnetometer values are sent, followed by the packet number and battery, solar, and temperature voltages. Finally the second verification packet is sent. The entire data stream will be thrown out if the final verification packet doesn't match the expected 10101010. This system assumes that data won't be corrupted in the middle of the verification packets, which is not always true, but the likelihood of a significant percentage of corrupt data not affecting the headers is very low.

5.4) Processor

At the beginning of the project, the group was given several microprocessors and told to choose between them. The processors and their relevant statistics are shown below.

Brand Model	Parallax Basic stamp 2sx	Parallax BS2-IC	NetMedia BasicX-24
Cost	59.99	49.99	49.95
Current (Run/sleep)	60mA/500uA	8mA /100uA	20 mA
Voltage	5-15v	5-15v	3-5V
Speed	20 MHz	20 MHz	35 MHz
Size	24 pin DIP	24 pin DIP	24 pin DIP
Memory (ram)	32 Bytes	32 Bytes	400 bytes
Program + Data	8x2KB	2KB	32KB
SPI Bus	No	No	Yes
On Chip ADC	No	No	Yes

Figure 17: Processor Comparison

After comparing the available processors, the decision was made to use the BasicX-24. The decision was made as a tradeoff between slightly higher power consumption than the BS2-IC for much more memory. The integrated SPI bus for magnetometer communication and the on-chip ADC for temperature and voltage level monitoring also made the BasicX-24 the best choice for our application. The power requirement was low enough to reasonably fit into the power budget.

Once a processor was chosen, the group decided the tasks that the processor would be responsible for. The first task was magnetometer communication. The processor must query the magnetometer for data, and then receive the data when transmitted. The raw magnetometer data must be stored in memory. The processor also reads the battery voltage from the power board through an ADC and stores that value in memory. The solar panel voltage is run through a resistor voltage divider, and then read by the ADC and the value is stored in memory. Finally, the temperature sensor voltage is read through the ADC and stored as well. After all data is acquired, a header packet is sent by the processor through the transmitter, followed by all the data acquired in the last reading cycle. The processor will run in this loop perpetually. The software design to implement this cycle is discussed further in section 5.8.

5.5) System Board

A single board was designed to mount the magnetometer, processor, and transmitter. It was purchased from the same vendor that produced our power board. The board designed was double sided, with through-plated holes on a standard 0.062" FR-4 epoxy glass laminate board, with dimensions of 3.8" x 2.5". Mounting holes were placed in each corner to allow attachment to the polycarbonate cards. The front of the board is shown below.

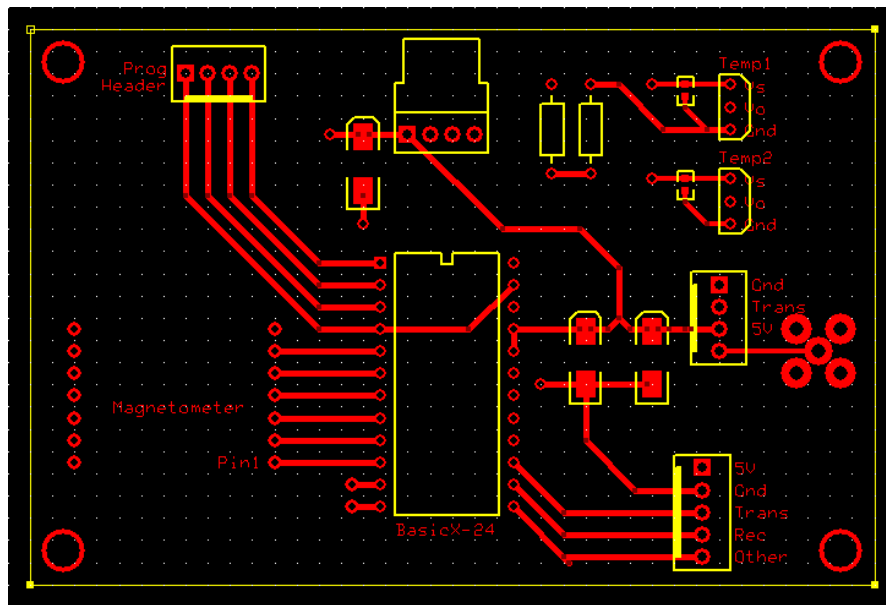


Figure 18: Processor Board Front

The BasicX-24 is placed directly in the center of the board in a socket. Directly to its left is the magnetometer, also in a socket. Above the magnetometer is a programming header, which allows us to plug in a custom-made serial cable to program the microprocessor without removing it from the socket. The pin header at the top center receives the four-pin output from the power board, which contains 5V,

ground, battery voltage, and solar cell voltage. The two resistors are a voltage divider to cut the solar panel voltage in half so that it can be properly read by the 5V ADCs on the processor. Two temperature sensor hookups are provided at the top right corner. When populating the board, we placed an Analog Devices TMP35 temperature sensor (See Appendix I) directly in the board. Another could be attached anywhere else in the satellite and connected to the board with a cable if desired. The four pin socket below the temperature sensors for the transmitter unit. The group of holes to the right of the transmitter is the SMA jack to allow easy cable hookup to an antenna. The five pin header in the bottom right corner is unused. It allows for future expansion of communication capabilities, and has an extra pin for two-way communication. The assembled board with processor and magnetometer in their sockets is shown below.



Figure 19: Processor Board Assembled

5.6) Physical Construction

Upon consultation with the ECE shop, the decision was made to build the satellite frame from rectangular and angle aluminum stock. This decision was made due to the light weight and ease of obtaining the aluminum material. Aluminum is also easily machinable. The frame design consists of four vertical posts with an angle aluminum cap placed at each end. This allows a flat surface on each side for solar panels and end plates to be affixed. The figure below shows the basic frame blueprint.

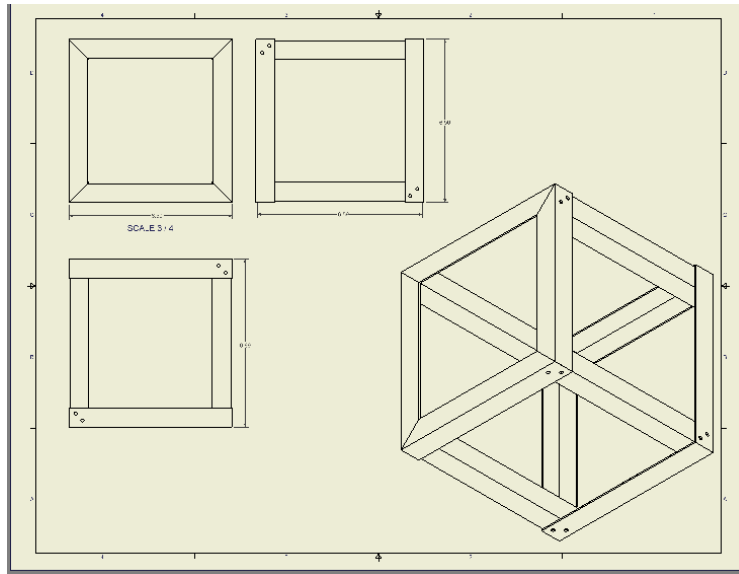


Figure 20: Frame Blueprint

Two sets of rails are also affixed across the inside of the frame to allow for the mounting of polycarbonate cards. The rails can be seen in the photograph of the completed frame below.



Figure 21: Completed Frame

The PCBs are connected to standoffs, which are then bolted onto the cards with machine screws. The cards themselves are then affixed to the frame rails with machine screws. Nylon bolts have been hot-glued to the solar panels, which line up with holes drilled in the angle aluminum sides. The panels are held on by nylon nuts inside the satellite. The top plate is made from polycarbonate which has been tapped for machine screws. An antenna is mounted in the middle of the top plate. An aluminum bottom

plate was cut and drilled to allow the bottom of the satellite to be completely covered. This allowed us to affix hanging hardware to the aluminum plate for flight testing, where the bottom would be suspended so that the antenna on top would be pointing at the ground. The finished product is shown below.

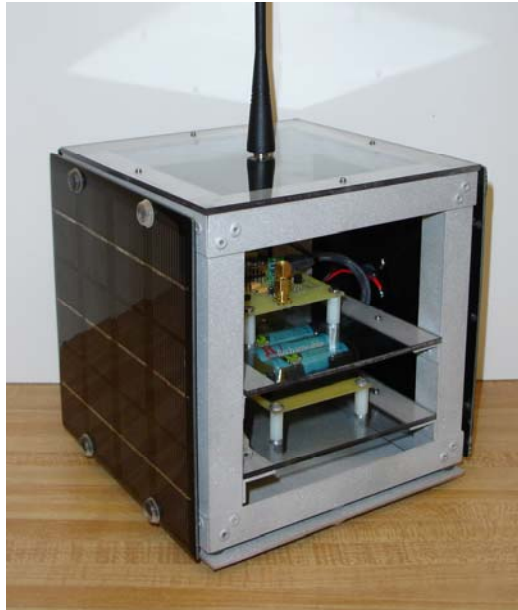


Figure 22: Completed Satellite

5.7) Software Design

The first step in design the satellite software was laying out the operations that the satellite was expected to perform. Since the satellite only utilizes one way communication, the software must function independently of user input. Taking this into account, the group assembled a high-level program structure for the satellite processor. The flow chart is shown below.

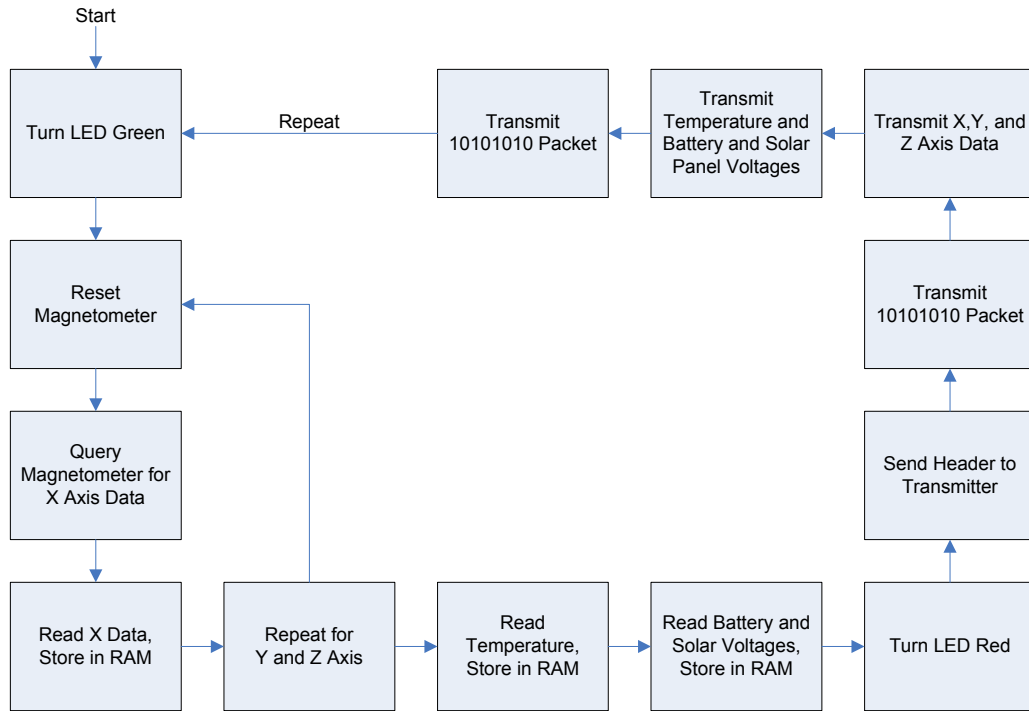


Figure 23: Satellite Software Flow Chart

The software loop begins in the top left hand corner of the flow chart. The BasicX-24 has an onboard LED that is turned green to show that the processor is not transmitting. Next the processor sends a reset pulse to the magnetometer, which must be done before any data requests. A query is sent from the processor to the magnetometer instructing the magnetometer to take an X axis magnetic field reading. When the magnetometer has completed the reading, the DRDY line on the SPI bus will be set high. This alerts the processor that data is incoming. The incoming data is read and stored in RAM. This process is repeated for the Y and Z axis as well. After all magnetometer data is stored, the processor will read the voltage on the temperature sensor, battery voltage, and solar cell voltage ADC ports. These values are also stored in RAM.

Once all data is collected, the LED on the processor turns red, indicating that a transmission cycle is in progress. The communications header outlined in Section 5.4 is sent to the transmitter, followed by the known good data packet of 10101010. Following this, the X, Y, and Z axis values are transmitted. The temperature, battery, and solar cell voltages are then transmitted. Finally, another 10101010 known good data packet is sent. After all data has been transmitted, the loop resets. Any time the satellite is powered, the program loop is operational. See Appendix B for full software code.

6) Base Station Design

This section will describe the design and construction of the base station used to receive broadcasts from the satellite.

6.1) Processor Unit

The entire base station was designed around the LCDX display module (See Appendix J). This is a 4x20 character backlit LCD screen combined with BasicX-35 processor. The module also includes an SPI interface, a serial port for programming and communication, 8 onboard ADCs, and a power jack for wall power. Also included in the kit was a 4x4 keypad for user input. This product was selected first because all of the included components and features covered most of the functionality needed from the base station. This single component choice provides data processing, hardware user interface, and a visual data display. The unit is shown below.



Figure 24: LCDX Module

As shown in the picture, the LCD module is mounted directly on top of the processing hardware, providing a compact package. The DB-9 serial port and power jack can be seen in the front left corner.

With the backlight off, LCDX requires 34 mA of current with a voltage range of 7.2-12 VDC. The current consumption can be as high as 200 mA when the backlight is turned on to maximum intensity. Also, the LCDX has an onboard 5V regulator, with a maximum current output of 40 mA.

6.2) Power System

The power system for the base station uses two power sources: wall power through a wall-plug style converter, or alkaline batteries. The power circuit is much simpler because there is no provision for charging the batteries. This decision was made because we expect the base station to usually be used in a place where wall power is available, such as a lab. However, batteries were included to ensure that we could do testing in a variety of situations with the receiver in different positions, as well as outdoors.

The only components that require power in the base station are the LCDX module and the receiver. For simplicity, it was decided that the 5V courtesy supply on the LCDX would be used to power the receiver. The worst case current requirement of the system was calculated to be 5.5 mA for the receiver and 200 mA for the LCDX, for a total of approximately 205 mA. Since the backlight consumes so much power, the group decided to leave it off unless specifically turned on by user input, so the normal current draw for the entire system would be closer to 40 mA. Panasonic alkaline AA size batteries with a nominal voltage of 1.5V and a current rating of 1550 mAh were chosen as the battery source because of their low cost and ease of availability. Six of these batteries are wired in series to provide the necessary voltage to the LCDX.

The batteries are wired to a power jack for wall power. The power jack is a three pin unit that allows the battery current through to the rest of the system when there is no plug inserted into the jack. When the power plug is inserted into the jack, the batteries are shown an open circuit, and the wall unit powers the system. This ensures that if the system is plugged in to wall power, the batteries are not being used. The power is then routed through a single pole single throw switch used to turn the entire unit on and off, and then into the LCDX unit.

6.3) Receiver

The receiver chosen is the matching pair to the transmitter used in the satellite, the RLP-434 made by Laipac Technology (See Appendix F). Since the transmitter/receiver pair is designed to only send and receive data with no processing or error correction, the receiver system is very simple. The receiver unit is mounted in sockets on the base station circuit board. It receives power from the LCDX 5V regulator, with a 100 μ F filtering capacitor placed on the power supply line. The output from the receiver is directly connected to an SMA jack.

A fabricated cable connects the SMA jack on the circuit board to another SMA jack mounted on the exterior corner of the box. This allows different antennas to be connected and disconnected for testing purposes without opening the box. The selected antennas are $\frac{1}{4}$ wavelength whip antennas, which were selected because of their low cost and simplicity.

6.4) Physical Construction

Prior to the project the group had acquired a large project enclosure, measuring approximately 9.5" wide by 8" deep by 4" tall. The two-part plastic enclosure featured a sloped front with removable aluminum faceplate, and a rear metal backplate. To enhance the aesthetic appeal of the finished product, the aluminum faceplate was discarded, and in its place a piece of clear polycarbonate was substituted. The polycarbonate was machined to utilize the existing mounting hardware. Holes were also drilled to allow the LCD display, keypad, and power switch to be mounted directly into the faceplate. The rear aluminum plate was machined to allow for a power receptacle for wall power, as well as a serial port so that the processor could be reprogrammed without taking apart the entire unit.

The bottom half of the unit had pre-molded standoffs, which we tapped to allow for the mounting of a piece of polycarbonate across half of the bottom of the box. The polycarbonate was drilled and tapped to allow the mounting of two triple AA battery holders and a small circuit board. The circuit board holds the receiver unit, as well as an SMA jack for the antenna cable. It also acts as a connection point for the power jack and battery holders, which are wired to provide wall power when the power jack is in use, and battery power when it is not. The antenna itself is mounted at the top corner of the unit. All cables used to connect the components have been fitted with slide connectors to allow easy disassembly of the unit. The cables are wrapped in heat shrink tubing to provide a uniform look. The finished product is shown below.



Figure 25: Base station Final Construction

The LCDX unit can be seen mounted in the top left corner. The edge of the circuit board and the receiver module are visible at the back right. The batteries are tucked under the keypad, and the backside of the power jack can be seen on the right hand side of the rear plate. The antenna is attached to the SMA jack at the top right corner.

6.5) Software Design

The base station software is much more complicated than the satellite software. The base station performs multiple duties:

- Receive data packets
- Calculate heading from raw magnetometer data
- Refresh display with data
- Accept keypad input

The varied nature of the tasks that the processor must handle required a software design approach that required a main loop to be combined with two subroutines. The software flow diagram is shown on the next page.

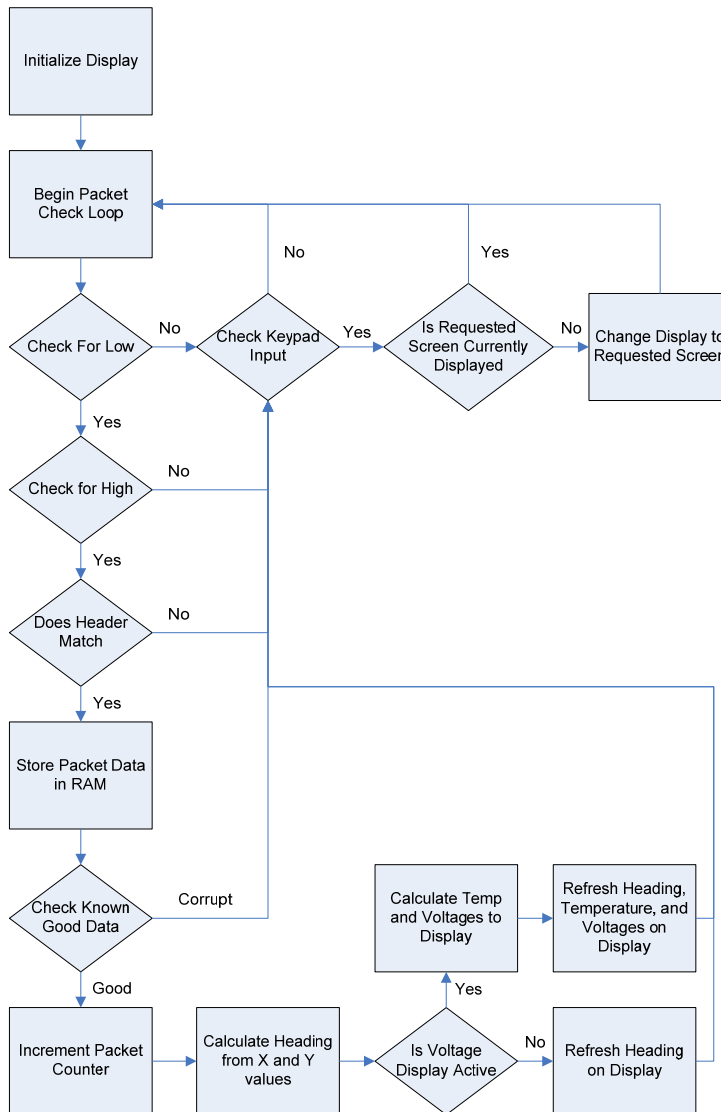


Figure 26: Base Station Software Flow Chart

The software begins by initializing the display and displaying a welcome message. The display remains on the welcome message until a key is pressed to select a screen. The packet check loop begins. The processor checks for a low value from the receiver, which it will usually receive. The processor waits a short amount of time, and checks for a high. If the low and the high are found, the processor will enter the packet checking process explained in Section 5.4. If the initial low and high are not found, or if at any point the packet check process fails, the system will enter the keypad check subroutine.

If a key is being pressed while the program is checking for keypad input, the processor will store the value of the key being pressed in memory. The value of the key is then compared against the key value of the screen that is currently being displayed. If they are different, the new screen is displayed. If they are

the same, the processor throws out the keypress to avoid an unnecessary screen refresh. After the keypad check subroutine is executed, the processor will return to the initial packet check loop.

If the processor receives a good header, it will store the following data in memory. The known good data packets of 10101010 are checked. If either packet is corrupt, the processor will ignore the data. If both the lead and trailing packets are valid, the processor will begin to process the stored data. First the received packet counter is incremented. Then the processor uses the X and Y axis values to calculate the heading in degrees. A compass direction value is also calculated. This operation is discussed in more detail in section 6.5.1. If the system is currently in the display mode where the temperature and system voltages are shown, the processor will extract the relevant data from the stored satellite transmission. The temperature sensor voltage is converted to Celsius using the simple ratio of 10mV/°C. The solar cell voltage is doubled because a half value is sent due to limits of the ADC on the satellite. All the changed information is refreshed in one display pass. The system then checks for any key presses, and then proceeds back into the packet check loop.

In addition to the program code, different display modes were also programmed. These allow the user to choose between different screen outputs to see different types of data. Each display is listed below by its corresponding keypress.

- 1) Title Screen: Displays “Magnetometer MQP”
- 2) Heading Screen: Displays packet count, heading in degrees, and cardinal direction.
- 3) Satellite Health: Display battery voltage, solar cell voltage, and satellite temperature.
- 4) Calibration Data Max Screen: Displays maximum magnetometer reading received for all three axes.
- 5) Calibration Data Min Screen: Displays minimum magnetometer reading received for all three axes.
- 8) Packet Data Screen: Displays number of packets sent and received, with a corresponding percentage of successful packets received.
- 9) Power Test Screen: Displays “Power testing in progress, do not disturb”. The base station also outputs serial data to the rear panel serial port with time and voltage information for acquisition by a laptop computer to allow the rundown test to proceed without human intervention.
- Help) Help Screen: Shows user the different available displays
- 0) Brightness Screen: Allows used to adjust the backlight intensity

See Appendix C for full software code.

6.5.1) Heading Calculation

To compute the heading the group utilized the multipoint calibration primer made available by the manufacturer of the magnetometer (See Appendix I). There are three steps to computing the heading. The

first step was to adjust the data using the calibration previously discussed. The second step was to compute the heading angle through a simple trigonometry operation on the X and Y variables. The following formula is used to calculate the current heading.

$$\text{Heading} = \text{Arctan} \left(\frac{Y}{X} \right)$$

This makes sense when the following diagram is considered, which shows the physical basis for the mathematical calculation.

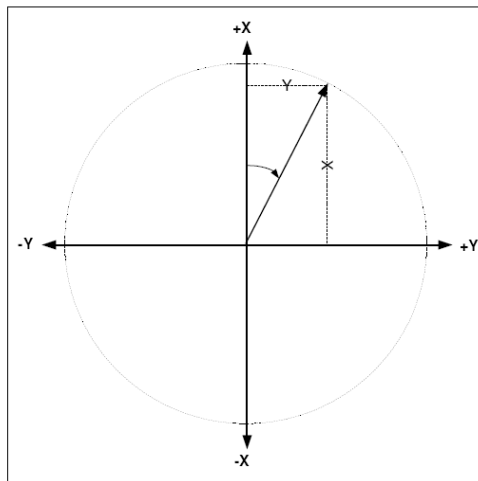


Figure 27: Heading Angle Calculation Diagram

The formula returns the heading from 0 – 90 degrees, so one more calculation is necessary to determine the correct quadrant for the angle to be placed in. The following table shows the logic table used to identify the correct quadrant.

X Value	Y Value	X	Y	Calculation	Final Heading
≥ 0	≥ 0	+	+	Final Heading = Heading + 0	0 - 90
< 0	≥ 0	-	+	Final Heading = Heading + 180	91 - 180
< 0	< 0	-	-	Final Heading = Heading - 180	181 - 270
≤ 0	≤ 0	+	-	Final Heading = Heading + 360	271 - 359

Figure 28: Quadrant Determination Table

7) Testing and Results

This section will detail the tests conducted on the satellite and base station to verify proper operation.

7.1) Power System Testing

To ensure that our satellite met the design goal of functioning in a 50/50 light/dark environment with periods of 45 minutes, the power system of the satellite was tested. In normal operation in the dark, the batteries provide 63 mA of current when the transmitter is idle, and 71 mA at approximately 3.5V when the satellite is transmitting data. After passing through the boost converter, the output of the power circuit is 33 mA normally and 39 mA at 5V when transmitting. This is an efficiency of 79%. The Linear LT1300 boost converter that we used has an efficiency rating at our operating voltages of approximately 86%. Our experimentally found efficiency number makes sense because we used an additional LR filter at the boost converter output for less noise in the power supply. There is also an additional diode drop where a diode was placed to prevent the voltage regulator from charging the batteries.

When charging, the batteries receive an input current between 71 and 73 mA. Since the charging current is higher than the average current output, the satellite successfully meets the goal of being able to function in a 50/50 light/dark environment.

The ability of the batteries to power the satellite in a no-light situation for long periods of time was also tested. The batteries were fully charged, and then the satellite was left running until failure. Battery voltage was tracked by a laptop computer plugged into the base station. The satellite functioned for just over 17 hours. The discharge curve of the batteries can be seen below.

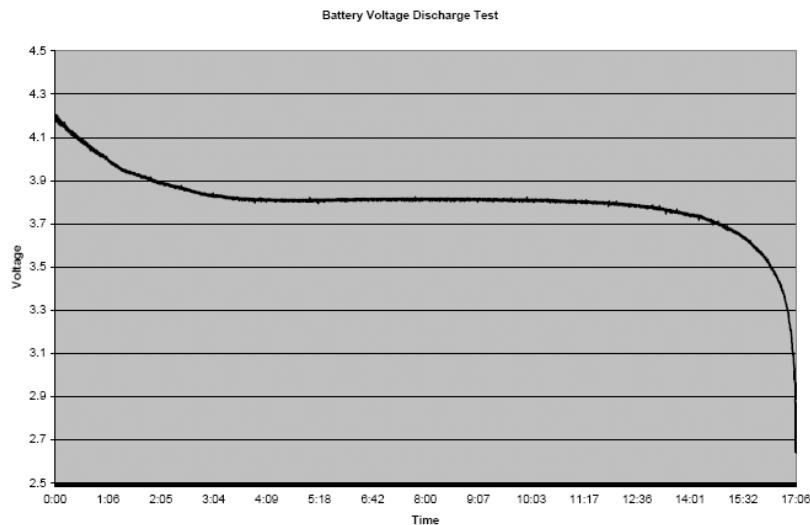


Figure 29: Battery Voltage Discharge Curve

7.2) Magnetometer Calibration

To ensure that the magnetometer was functioning correctly in our project, a calibration test was performed. This test showed the raw data gathered from the magnetometer at different headings. To perform this test the receiver was reprogrammed with code that would output only the raw data sent from the satellite. Next a compass was used to reference the sensor to a specific heading. The final step was to increment the satellite in ten degree increments and record the new X and Y values. Ideally, two sine wave functions would be present with a 60-70 degree offset. The resulting graph is shown below, which shows close to ideal behavior with some inconsistencies resulting from magnetic fields in the satellite itself.

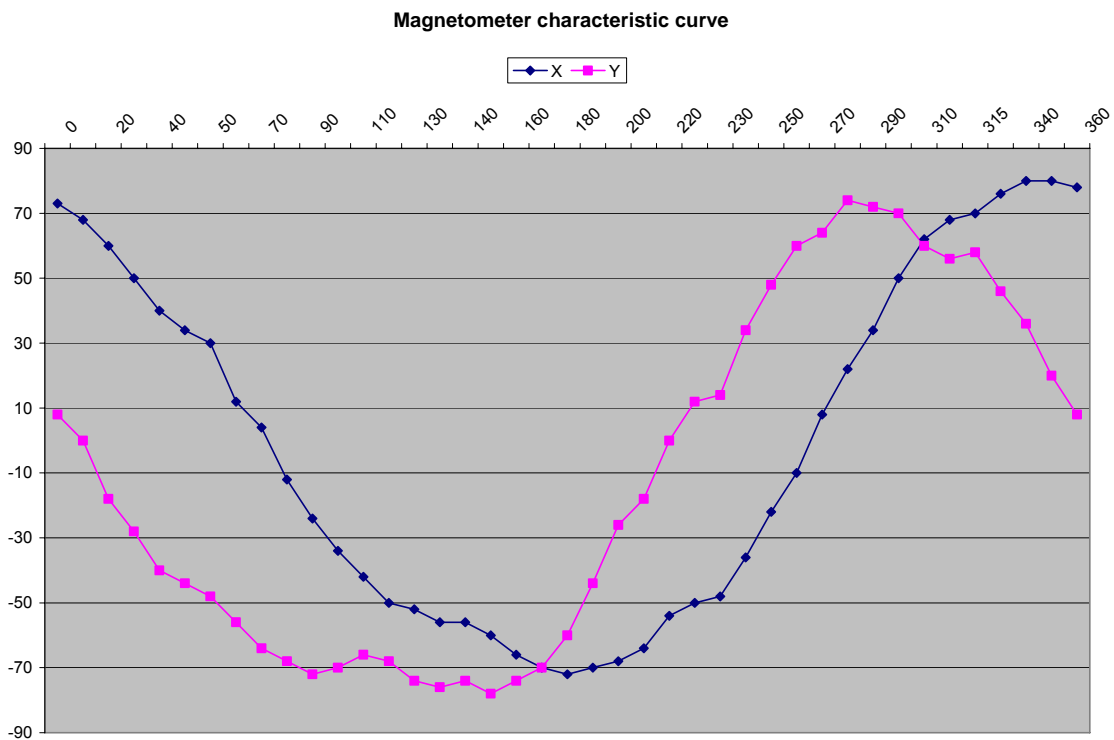


Figure 30: Magnetometer Calibration Curve

The data gathered was used to perform the data calibration in the receiver. The maximum and minimum were used to calculate offset and range of the sensors readings. The offset and range are essential in calculating the correct heading. The software checks each packet looking for a value that is outside its current range. If it finds one it will update the maximum value used in calculating a heading, so calibration is constantly taking place.

7.3) Communications System Testing

To test the range of the satellite communications system, the group conducted a simple range test. Initially the group expected good data to drop off with distance in a somewhat linear fashion. Preliminary tests showed that this was not the case. With the first communications protocol used by the group, which relied heavily on timing instead of an interrupt to pick up the main part of the header, reception was almost perfect out to approximately 25 feet, but then the receive rate dropped to almost zero.

This range was not acceptable for our project, as the transmitter/receiver pair claimed a range of up to 500 feet. This range, however, assumes that the transmitter is being powered with the maximum input voltage of 12 volts. Our satellite power system only operates at 5 volts. At less than half the maximum voltage, a lower range should be expected. The group re-wrote the receiver code that interpreted the header and slightly changed the shape of the header to the one that was discussed in section 5.4. Shorter pulses and tighter timing tolerances gave the receiver less of a chance to pick up a false positive. The use of an interrupt instead of a set time for the large low in the middle of the timing sequence also helped more packets be received successfully.

Finally, in a chance observation, it was noted that the packet receive rate was much greater when the cover of the base station was removed or propped open. With the cover closed the receive rate dropped off markedly. It was discovered that the serial line from the LCDX to the external serial port on the back of the enclosure was coupling noise into the antenna line. This line was being used to look at the packet values during our algorithm re-writing process. The noise was corrupting the data stream to the point of uselessness. A simple re-routing of the serial line eliminated the noise problem.

The range testing was redone, and the results were much better. With the simple $\frac{1}{4}$ wavelength dipole antennas, the satellite and base station transmitted successfully with less than 5% packet loss over distances up to 125 feet. At approximately 130 feet, the packet rate dropped to almost zero. These results satisfied the group, since as discussed above our transmitter was underpowered compared to the manufacturer's test units.

8) Summary and Conclusions

This section discusses whether our project met the design goals and possible future improvements to the project.

8.1) Overall Assessment

Overall the group considers the project to be successful. The original goal was the production of a nanosatellite with the ability to read magnetic fields and transmit the readings back to a base station for processing. To accomplish the overall goal, the group successfully accomplished many smaller objectives:

1. The design of a power system for the satellite. The group designed a power supply that allows the satellite to operate in a 50/50 light/dark cycle environment. The supply has an efficiency of 79%, which is reasonable for this application. The power supply is a self-contained module, and can be used in a variety of other solar power projects if desired.
2. Implementation of an SPI bus in BasicX software. The built-in SPI port had timing issues that were causing failed communication between the magnetometer and processor. This forced the group to implement a self-coded SPI protocol.
3. Design and implementation of a wireless communications protocol. The simple nature of our transmitter/receiver pair necessitated a software communications protocol to allow for the transfer of data. The protocol was designed and coded by the group.
4. The physical design and construction of the satellite frame and base station enclosure. A high-quality aluminum frame was designed and built by the group for the satellite, and an enclosure with a custom-made polycarbonate panel was fabricated for the base station.
5. The design of two printed circuit boards. The group designed two printed circuit boards for the satellite to give the final product a neat, professional look.

The team feels that we succeeded in all of these intermediate tasks, as well as the main task of completing the entire satellite system.

8.2) Future Work

There are many places where the group feels that the magnetometer nanosatellite project could be expanded.

- Space Worthiness – The satellite mockup is obviously not space ready. The frame could never withstand the physical strain of a launch and the harsh climate of space. It would be possible with much more time and resources to create a frame that could utilize most of the same internals. The components themselves would also have to be replaced with space-certified components to ensure

their functionality in space. This would present a difficulty with single-source components, like the BasicX-24 processor and the MicroMag magnetometer.

- Transmitter Range – Improved transmitter range would be a desirable improvement to our project. The satellite is hindered by a low voltage power supply for the transmitter. Also, a transmitter with a higher current draw would put the satellite into a negative power situation, where the solar panels could not adequately charge the batteries during each light cycle. A combination of a different transmitter and a more directional antenna could help the situation if the power problem could be worked around.
- Power System – The power budget for the satellite is extremely tight. There are only a few milliamps to spare before the budget would become negative resulting in a situation where the satellite would never fully recharge and would eventually turn off from lack of power.
- Batteries – The batteries used were simple AA Ni-Cad rechargables. These were selected because they were easily available at a local retailer. Ideally, a smaller sized Lithium Ion battery could be used, for lighter weight and smaller size.
- Expansion – The satellite frame has more space in the top third for the addition of an additional set of frame rails and another polycarbonate card. This card could possibly hold more instrumentation, a different transmitter, or anything else deemed interesting and worthwhile. However, due to the tight power budget, it may need to be powered directly from the solar cells and only operate while the satellite is in the light.
- True tri-axial heading calculation - Late in our project, the manufacturers of the MicroMag posted an application note that stated that heading could not be determined with their product alone. A tilt sensor would also have to be employed to allow a processor to calculate a three dimensional heading. Since this note was posted over 2/3 of the way through our product, the group elected to remain with the current two dimensional system and note the opportunity for future improvement in this document.

All of the points listed above provide a large opportunity for expansion on the group's original nanosat design.

8.3) Conclusion

The group successfully designed, built, and tested a picosatellite mockup with a built in magnetometer and wireless transmission module. A base station to receive and display the collected data was also constructed. This document has outlined and described all of the planning and design work, as well as providing test results on the finished product. Further ideas for expansion of the project were also presented.

Bibliography

- Wertz, J.R. and Larsen, W.J.; Space Mission Analysis and Design, 3rd Edition, 1999.
- Jilla, Cyrus and Miller, David; "Satellite Design: Past, Present, and Future", International Journal of Small Satellite Engineering, February 1997
- Wilson, Andrew; Jane's Space Directory: 1995-96. 11th Edition. Jane's Information Group Ltd. Surrey, UK. 1995.
- Friedman, Assi; ASUSat1: "Low-Cost, Student-Designed Nanosatellite", 14th Annual USU Conference on Small Satellites, U of Utah, 2000
- Merrill, Ronald, McElhinny, Michael, and McFadden, Phillip; The Magnetic Field of Earth, San Diego: Academic Press, 1996.
- Purvis, C.K.; Magnetic and Electric Fields, NASA Glenn Research Center. 3 Aug. 1994
<http://powerweb.grc.nasa.gov/pvsee/publications/seeov/magelfield.html>
- Allen, Jesse; Earth's Albedo in Decline, NASA Earth Observatory. March 2005
http://earthobservatory.nasa.gov/Newsroom/NewImages/images.php3?img_id=16905
- Theil, Stephan, Appel, Pontus, and Schleicher, Alexander; "Low Cost, Good Accuracy - Attitude Determination Using Magnetometer and Simple Sun Sensor", 17th Annual AIAA/USU Conference on Small Satellites, U of Utah, 2003.
- PNI Corporation; Application Note: Discrete Circuit for Magneto-Inductive Sensors, Dec. 2003
<https://pnincorp.com/downloadResource/c40c/manuals/67/Ap+Note+Discrete+Circuit+for+Magneto-Inductive+Sensors.pdf>
- Driesse, Anton; "Powering the Future with Photovoltaics", IEEE Canadian Review, Fall 2004: 22-26.
- National Aeronautics and Space Administration; Geosynchronous Orbit, Dec. 1995
http://liftoff.msfc.nasa.gov/academy/rocket_sci/satellites/geo-high.html
- Ganz, Aura, Gong, Yebin, and Li, Bo; "Performance Study of Low Earth-Orbit Satellite Systems." IEEE Transactions on Communications 42 (1994): 1866-1867
- Uzunalioglu, Huseyin; Probabilistic Routing Protocol for Low Earth Orbit Satellite Networks. IEEE International Conference on Communications, 7-11 June 1998, Atlanta, GA
- Bedingfield, K.L., and Leach, R.D.; Spacecraft System Failures and Anomalies
Attributed to the Natural Space Environment, NASA Reference Publication 1390, Aug. 1996
- Wasinger, Jerry; "Build a Wireless Weather Station" Circuit Cellar, Nov. 2001 p.34-41
- Florida Solar Energy Center (FSEC) ; Photovoltaic Fundamentals.
<http://www.fsec.ucf.edu/pvt/pvbasics/>
- Leveson, David; Magnetic Declination. City College of New York, 2000
<http://academic.brooklyn.cuny.edu/geology/leveson/core/linksa/magnetic.html>

Caruso, Michael, Bratland, Tamara, Smith, Carl, and Schneider, Robert; A New Perspective on Magnetic Field Sensing, Honeywell, Inc., May 1998

Appendix A - Matlab Solar Cell Simulation Code

Hexagonal Solar Panel Power Simulation

```
powernum = .43;
yangle = 0;
count = 1;
while (count <4)
yangle = (pi/6)*(count-1);
x = 0;
y=1;
angle = 0;
totalpower = 0;
while (x<(60*(pi/180)))
anglecorrect = cos(yangle);
anglecorrect;
panel1 = abs(cos((pi/3)+x));
panel2 = cos(x);
panel3 = cos((pi/3)-x);
power1(y)= panel1 .* (powernum * anglecorrect);
power2(y)= panel2 .* (powernum * anglecorrect);
power3(y)= panel3 .* (powernum * anglecorrect);
totalpower(y) = power1(y)+power2(y)+power3(y);
angleshort(y) = (180/pi)*x;
angle(y) = angleshort(y);
y = y+1;
x = x+(pi/180);
end
subplot (3,1,count)
plot (angle, totalpower)
xlabel('Degrees')
ylabel('Power (W)')
title('Power vs. Angle @ 0 degrees inclination')
grid on
count = count+1;
end
subplot (3,1,2)
title('Power vs. Angle @ 30 degrees inclination')
subplot (3,1,3)
title('Power vs. Angle @ 60 degrees inclination')
```

Square Solar Panel Power Simulation

```
powernum = 2.25;
yangle = 0;
count = 1;
while (count <4)
yangle = (pi/6)*(count-1);
x = 0;
y=1;
angle = 0;
```

```

totalpower = 0;
while (x<(90*(pi/180)))
anglecorrect = cos(yangle);
anglecorrect;
panel1 = abs(cos((pi/2)+x));
panel2 = cos(x);
power1(y)= panel1 .* (powernum * anglecorrect);
power2(y)= panel2 .* (powernum * anglecorrect);
totalpower(y) = power1(y)+power2(y)
angleshort(y) = (180/pi)*x;
angle(y) = angleshort(y);
y = y+1;
x = x+(pi/180);
end
subplot (3,1,count)
plot (angle, totalpower)
xlabel('Degrees')
ylabel('Power (W)')
title('Power vs. Angle @ 0 degrees inclination')
grid on
count = count+1;
end
subplot (3,1,2)
title('Power vs. Angle @ 30 degrees inclination')
subplot (3,1,3)
title('Power vs. Angle @ 60 degrees inclination')

```

Appendix B – Satellite Software

'The following code is for the Magnetometer MQP Picosat unit.
'The Code transmits raw data to the receiver

'Department of Electrical and Computer Engineering
'Worcester Polytechnic Institute.
'Danato Borelli & Justin Woodard

```
'----- MAIN
Public Sub Main()
'----- Set up

        Call PutPin(7, bxOutputHigh)      ' Pin 7 SSNOT
        Call PutPin(5, bxOutputLow)       ' Pin 5 Reset
        Call PutPin(10, bxOutputLow)      ' Pin 10 CLK
        Call PutPin(8, bxInputTristate)   ' Pin 8 MOSI
        Call PutPin(9, bxInputTristate)   ' Pin 9 MISO
        Call PutPin(6, bxInputTristate)   ' Pin 6 DRDY

'----- Define variables
        Dim preheader as byte             'used for communications
        Dim Header as byte                'used for communications

        Dim SendX As Byte                 'Get X Value
        Dim SendY As Byte                 'Get Y Value
        Dim SendZ As Byte                 'Get Z Value

        Dim BattVoltage as Integer        'Battery Voltage
        Dim SolarVoltage as Integer        'Solar Cell Voltage
        Dim SatTemp as Integer             'Satellite temperature

        Dim MX As Integer                  'X Axis Integer
        Dim MY As Integer                  'Y Axis Integer
        Dim MZ As Integer                  'Z Axis Integer

        Dim Xmax as integer                'Used for calibration
        Dim Xmin as integer                'Used for calibration
        Dim Ymax as integer                'Used for calibration
        Dim Ymin as integer                'Used for calibration

        Dim Temp As Integer                 'Temporary variable
        Dim Number As Integer
        Dim column As Byte                  'Temporary variable
        Dim count1 as byte

'----- Setup communications
Dim InputQueue3(1 to 16) as Byte
Dim OutputQueue3(1 to 50) as Byte
Call Definecom3(0,15,bx0000_1000) 'was pin 16
Call OpenQueue(InputQueue3, 16)
```

```
call Openqueue(OutputQueue3, 50)
Call OpenCom(3, 1200,InputQueue3,OutputQueue3)
```

```
'----- Set constants
```

```
    SendX = bx00100001      'Value sent over SPI bus
    SendY = bx00100010      'Value sent over SPI bus
    SendZ = bx00100011      'Value sent over SPI bus

    Preheader = bx10101010  'Preheader value 170
    Header = bx10101010    'Header value 170
    Temp = 0
```

```
Used to set calibration
```

```
    Xmax = 45 '44 '192
    Ymax = 100 '64 '250
'    Zmax = 0 ' 342
    Xmin = -70 '-82 ' 204
    Ymin = -80 '-98 '224
'    Zmin = 0 '230
```

```
'----- Start Main loop
```

```
'Debug.Print " Info , X  , Y  ,  Z "
call delay(1.0)
Do
```

```
'Call PutPin(26, bxOutputLow) 'red led off
'Call PutPin(25, bxOutputHigh) 'Green led on
'Call PutPin(26, bxOutputlow) 'red led on
'Call PutPin(25, bxOutputHigh) 'Green led on
'dontsend:
```

```
'----- Get H&S information
```

```
    BattVoltage = GetADC(16)
    SolarVoltage = GetADC(17)
    SatTemp = GetADC(19)
```

```
'----- Start SPI
```

```
Call SPIReset()
```

```
'----- Get data X axis
```

```
Call SPIsend(SendX)
Call SPIWait()
MX = SPIRead
```

```
'----- Get data Y axis
```

```
Call SPIReset()
Call SPIsend(SendY)
Call SPIWait()
MY = SPIRead
```

```

'----- Get data Z axis
Call SPIReset()
Call SPIsend(SendZ)
Call SPIWait()
MZ = SPIRead

'----- SPI Done
Call SPIDone()

'-----min & max data for cal
goto skipcal
If MX > Xmax then
  Xmax = MX
End if
if MX < Xmin then
  Xmin = MX
End if

If My > ymax then
  ymax = My
End if
if My < ymin then
  ymin = My
End if

'If Mz > zmax then
'  Zmax = Mz
'End if
'if Mz < zmin then
'  zmin = Mz
'End if
skipcal:

goto jumpdebug
Debug.Print " "
Debug.Print "---New Data---"
Debug.Print "X = "; CStr(mx)
Debug.Print "Y = "; CStr(my)
Debug.Print "Z = "; CStr(mz)
Debug.Print "--Max/Min Data--"
Debug.Print "Xmax = "; CStr(Xmax)
Debug.Print "Ymax = "; CStr(ymax)
Debug.Print "Zmax = "; CStr(zmax)
Debug.Print "Xmin = "; CStr(Xmin)
Debug.Print "Ymin = "; CStr(ymin)
Debug.Print "Zmin = "; CStr(zmin)
Debug.Print "-----"

jumpdebug:

```

```
Call PutPin(26, bxOutputLow) 'red led off
Call PutPin(25, bxOutputHigh) 'Green led on
```

```
'Call PutPin(25, bxOutputLow) 'red led on
'Call PutPin(26, bxOutputHigh) 'Green led off
'goto dontsend
```

```
'----- Setup Data and send custom header
call clearQueue (Outputqueue3)
Call delay(0.001)
```

```
Call PutPin(15, bxOutputhigh)      ' pre state1
    Call delay(0.001)
Call PutPin(15, bxOutputlow)      ' pre state1
    call delay(0.025)
```

```
Call PutPin(15, bxOutputlow)      ' State1
    Call delay(0.001)
```

```
Call PutPin(15, bxOutputHigh)     ' State1
    call delay(0.001)
```

```
Call PutPin(15, bxOutputlow)     ' state2
    Call delay(0.001)
```

```
Call PutPin(15, bxOutputHigh)    ' state3
    call delay(0.001)
```

```
Call PutPin(15, bxOutputlow)     ' state4
    Call delay(0.006)
```

```
Call PutPin(15, bxOutputHigh)    ' state5
    call delay(0.0001)
```

```
Call PutPin(15, bxOutputlow)     ' state6
    Call delay(0.005)
```

```
Call PutPin(15, bxOutputHigh)    ' State7
    call delay(0.005)
```

```
Call PutPin(15, bxOutputlow)     ' State8
    Call delay(0.005)
```

```
Call PutPin(15, bxOutputHigh)    ' State9
    call delay(0.010)
```

```
'----- Send Data
```

```
call putqueue(Outputqueue3, Preheader , 1)
call putqueue(Outputqueue3, Mx , 2)
call putqueue(Outputqueue3, My , 2)
call putqueue(Outputqueue3, Mz , 2)
call putqueue(Outputqueue3, Temp , 2)
call putqueue(Outputqueue3, BattVoltage, 2)
call putqueue(Outputqueue3, SolarVoltage, 2)
```

```

call putqueue(Outputqueue3, SatTemp , 2)
call putqueue(Outputqueue3, Header , 1)

'call putqueue(Outputqueue3, Xmax , 2)
'call putqueue(Outputqueue3, Xmin , 2)
'call putqueue(Outputqueue3, Ymax , 2)
'call putqueue(Outputqueue3, Ymin, 2)
'call putqueue(Outputqueue3, Zmax , 2)
'call putqueue(Outputqueue3, Zmin , 2)

'Debug.Print "Info"; ","; cstr(mx); ","; cstr(my); ","; cstr(mz)

Call PutPin(25, bxOutputLow) 'red led on
Call PutPin(26, bxOutputHigh) 'Green led off

'If temp < 244 then
Temp = temp +1
Loop
End Sub

'----- Reset SPI
Private Sub SPIReset()
' Set the SSNOT pin high.
    Call PutPin(7, bxOutputLow)
    Call Delay(0.0001)

' Set the RST pin high
    Call PutPin(5, bxOutputHigh)
    Call Delay(0.0001)
    Call PutPin(5, bxOutputLow)
    Call Delay(0.00001)

End Sub

'----- End SPI
Private Sub SPIDone()
    Call PutPin(7, bxOutputHigh) 'end SPI communication
End Sub

'----- Output command
Private Sub SPIsend(Value as Byte)
    Dim number as Byte
    Dim column As Byte
    Number = bx10000111 ' Number = bx10000000

    For column = 0 To 7 ' transmits one byte at a time

        If (Value AND Number) > 0 then
            Call PutPin(8, bxOutputHigh)

```

```

                Else
                    Call PutPin(8, bxOutputLow)
                End If

                Call PutPin(10, bxOutputHigh)
                Call Delay(0.0001)
                Call PutPin(10, bxOutputLow)
                Number = Number \ 2
                Call Delay(0.00001)
            Next
        End sub

'----- Wait for Data
Private Sub SPIWait()
    Dim PinLogicLevel As Byte
    PinLogicLevel = 0

    ' Read the value of Drdy.
    Do While (PinLogicLevel = 0)
        PinLogicLevel = GetPin(6)
        call CPUSleep()
        'Call Delay(0.00001)
    Loop
End Sub

'----- read SPI bus
Private Function SPIRead() As Integer
    Dim Number As Integer
    Dim column As Byte
    Number = 0

    For column = 0 To 15
        Call PutPin(10, bxOutputHigh)
        Number = Number * 2
        if GetPin(9) = 1 then ' was pin 6
            Number = Number + 1
        end if
        Call PutPin(10, bxOutputLow)
        Call Delay(0.00001)
    Next

    SPIRead = Number

End Function

'----- Send Data
Private Function senddata() As Integer
    Dim number as Byte
    Dim column As Byte
    Dim Value As Byte
    Number = 0

```



```
'----- Send header
  For column = 0 To 7
    If (Value AND Number) > 0 then
      Call PutPin(8, bxOutputHigh)
    Else
      Call PutPin(8, bxOutputLow)
    End If

    Call Delay(0.00001)
    Number = Number \ 2
  Next

End Function
```


'----- Setup timer Variables

dim Hour as byte 'Time variable for testing
dim Minute as byte 'Time variable for testing
dim Second as single 'Time variable for testing

'----- setup calibration variables

Dim Xmax as integer 'X calibration variable
Dim Ymax as integer 'Y calibration variable
dim Zmax as integer 'Z calibration variable
Dim Xmin as integer 'X calibration variable
Dim Ymin as integer 'Y calibration variable
Dim Zmin as integer 'Z calibration variable

dim Xoffset as integer 'X value for Offset
dim Yoffset as integer 'Y value for Offset
dim Xrange as integer 'X value for Range
dim Yrange as integer 'Y value for Range
dim heading as integer ' the current heading

Dim Xvalue as integer 'Current X Value used to calculate heading
Dim Yvalue as integer 'Current X Value used to calculate heading

Dim BattVoltage as Integer 'Battery voltage value
Dim SolarVoltage as Integer 'solar cell voltage value
Dim SatTemp as Integer 'Satellite temperature value

Dim BattVoltage1 as single 'display value for Battery voltage
Dim BattVoltage2 as integer 'display value for Battery voltage
Dim SolarVoltage1 as single 'display value for Solar Cell voltage
Dim SatTemp1 as single 'display value for Satellite temperature

'----- Set Variables to constant values or initial values

comtest = 0 ' Test variable for communications test state = 0
PreHeader = 0 ' PreHeader value to check data that is sent value is = 0
Header = 0 ' Header value to check data that is sent value is = 0
PacketCount = 0 ' Number of packets received starts at zero
pagenumstart = 1 ' used to display first page
pagenum = 1 ' used to display current page first page is 1
TimedOut = true ' System will timeout if data is not received in time

Xmax = 45 '44 '192 'X Maximum calibration variable
Ymax = 100 '64 '250 'Y Maximum calibration variable
'Zmax = 0 ' 342 'Z Maximum calibration variable not used
Xmin = -70 '-82 ' 204 'X Minimum calibration variable
Ymin = -80 '-98 '224 'Y Minimum calibration variable
'Zmin = 0 '230 'Y Minimum calibration variable not used

'----- Other Setup

Call PutPin(17, bxInputTristate) 'set pin 17 communications pin to high impedance
Call SetBrightness (0) 'Set LCD backlight brightness level to min

'--- definable characters

Dim Happy (1 to 8) As Byte 'Setup variable for charecter 0
Dim D_Arrow (1 to 8) As Byte 'Setup variable for charecter 2
Dim U_Arrow (1 to 8) As Byte 'Setup variable for charecter 1
Dim line (1 to 8) As Byte 'Setup variable for charecter 3

' Make a sad face

Happy(1) = 0
Happy(2) = 0
Happy(3) = 10
Happy(4) = 0
Happy(5) = 14
Happy(6) = 17
Happy(7) = 0
Happy(8) = 0

' Make a down arrow

D_Arrow(1) = 0
D_Arrow(2) = 4
D_Arrow(3) = 4
D_Arrow(4) = 4
D_Arrow(5) = 4
D_Arrow(6) = 21
D_Arrow(7) = 14
D_Arrow(8) = 4

' Make a up arrow

U_Arrow(1) = 4
U_Arrow(2) = 14
U_Arrow(3) = 21
U_Arrow(4) = 4
U_Arrow(5) = 4
U_Arrow(6) = 4
U_Arrow(7) = 4
U_Arrow(8) = 0

' Make a line

line(1) = 0
line(2) = 4
line(3) = 4
line(4) = 4
line(5) = 4
line(6) = 4
line(7) = 4

```

line(8) = 0

' Store the new characters in the LCD
Call SetCustomChar(0, Happy) 'store charecter 0
Call SetCustomChar(1, U_Arrow) 'store charecter 1
Call SetCustomChar(2, D_Arrow) 'store charecter 2
Call SetCustomChar(3, line) 'store charecter 3

' Note:
' Use DisplayAtCell to add the down arrow to the text
'Call DisplayAtCell(Chr(7), 1, 11)

'----- set display variables to sad face
Heading = Chr(0)
Comdir = Chr(0)

'----- Lightup display for startup

for count1 = 3 to 20
Call setbrightness (count1*9)
call delay(0.05)
next

'----- Setup communications

Dim InputQueue3(1 to 148) as Byte ' Create buffers
Dim OutputQueue3(1 to 16) as Byte ' Create buffers
Call OpenQueue(InputQueue3, 141)
call Openqueue(OutputQueue3, 16)
Call Definecom3(17,0,bx0000_1000) ' Setup the com port
Call OpenCom(3, 1200,InputQueue3,OutputQueue3)

'----- Lightup display

call delay(0.5)
for count1 = 20 to 1 step -1
Call setbrightness (count1*9)
call delay(0.05)
next

'---- Main
Do

'----- Wait for data
checkfordata:
'Call PutPin(16, bxoutputlow) 'Used for com test
state1: ' firat state of the com
If (getpin(17) = 0 ) Then
    Goto state2

```

```

End If

call checkkey(pagenum,pagenumb4)
    If (getpin(17) = 0 ) Then
        Goto state2
    End If
If pagenumb4 = 0 then
goto displaynew
end if
Goto State1

State2:
delay(0.0002) ' delay(0.0044)
If (getpin(17) = 1 ) Then
Goto State2a
End if
Goto State1

State2a:
comtest = 2
delay(0.0002)
If (getpin(17) = 0 ) Then
Goto state2b
End If
Goto State1

State2b:
delay(0.0002) ' delay(0.0044)
comtest =3
If (getpin(17) = 1 ) Then
Goto State3
End if
'Debug.Print "State 2b = "; CStr(getpin(17))
Goto State1

State3:
delay(0.002)
If (getpin(17) = 0 ) Then
Goto State4
End If
'Debug.Print "State 3 = "; CStr(getpin(17))
Goto State1

State4:
WaitForInterrupt(bxPinRisingEdge)

State5:
delay(0.004)
If (getpin(17) = 0 ) Then
Goto State6

```

```

End If
'Debug.Print "State 5 = "; CStr(getpin(17))
Goto State1

```

State6:

```

delay(0.004)
If (getpin(17) = 1 ) Then
End if
Goto State7
'Debug.Print "State 6 = "; CStr(getpin(17))
Goto State1

```

State7:

```

delay(0.006)
If (getpin(17) = 0 ) Then
Goto State8
End If
'Debug.Print "State 7 = "; CStr(getpin(17))
Goto State1

```

State8:

```

delay(0.004)
If (getpin(17) = 1 ) Then
Goto State9
End if
'Debug.Print "State 8 = "; CStr(getpin(17))
Goto State1

```

State9:

```

call ClearQueue(Inputqueue3)

```

'----- Get data

```

call Getqueue(Inputqueue3, PreHeader ,1 , 0.1, Timedout)      'First packet sent 170
call Getqueue(Inputqueue3, MagValue_X      ,2 , 0.1, Timedout)
call Getqueue(Inputqueue3, MagValue_Y      ,2 , 0.1, Timedout)
call Getqueue(Inputqueue3, MagValue_Z      ,2 , 0.1, Timedout)
call Getqueue(Inputqueue3, PacketsSent      ,2 , 0.1, Timedout)
'call Getqueue(Inputqueue3, Xmax            ,2 , 0.5, Timedout)
'call Getqueue(Inputqueue3, Xmin            ,2 , 0.5, Timedout)
'call Getqueue(Inputqueue3, Ymax            ,2 , 0.5, Timedout)
'call Getqueue(Inputqueue3, Ymin            ,2 , 0.5, Timedout)
call Getqueue(Inputqueue3, BattVoltage      ,2 , 0.5, Timedout)
call Getqueue(Inputqueue3, SolarVoltage     ,2 , 0.5, Timedout)
call Getqueue(Inputqueue3, SatTemp          ,2 , 0.5, Timedout)
call Getqueue(Inputqueue3, Header           ,1 , 0.1, Timedout) ' Last packer sent 170

```

```

If (PreHeader = 170) then
    If (header = 170) then

```

```

        PacketCount = PacketCount + 1
        Goto Printvalue
    End if
end if
comtest = 11
Goto skipprint

Printvalue:

'----- min max

If MagValue_X > Xmax then
    Xmax = MagValue_X
End if
if MagValue_X < Xmin then
    Xmin = MagValue_X
End if

If MagValue_Y > ymax then
    ymax = MagValue_Y
End if
if MagValue_Y < ymin then
    ymin = MagValue_Y
End if

'----- computing the heading

Xoffset = Xmax + Xmin ' calculate offsets
Yoffset = Ymax + Ymin
Xrange = Xmax - Xmin ' calculate the range
Yrange = Ymax - Ymin

Xvalue = MagValue_X - Xoffset ' subtract out the offsets
Yvalue = MagValue_Y - Yoffset

if (Xrange > Yrange) then
    Yvalue = ( Yvalue * Xrange ) \ Yrange ' perform gain matching
else
    Xvalue = ( Xvalue * Yrange ) \ Xrange ' perform gain matching
end if

angle = ((atn(csng(Yvalue)/csng(Xvalue)))/csng(3.14159)) * csng(180) ' ' compute the angle

heading = cint(angle)

'Debug.Print ""
'Debug.Print "yvalue="; CStr(yvalue)
'Debug.Print "xvalue="; CStr(xvalue)
'Debug.Print "angle="; CStr(angle)

```



```
'Debug.Print "angle= atn "; 'CStr(atn( csng(Yvalue) / csng(Xvalue) ))
```

'----- the following code computes the quadrant

```
if (Xvalue >= 0) then
    if (Yvalue >= 0) then
        'heading = cint(angle) 'quadrant +X, +Y (0 to 90)
        'Debug.Print "1 ----heading = "; cstr(heading)

    end if
end if
if ( Xvalue < 0 ) then
    if (Yvalue >= 0) then
        'heading = cint(angle) 'quadrant -X, +y (91 to 180)
        heading = 180 + heading
        'Debug.Print "2 ----heading = "; cstr(heading)

    end if
End if
if ( Xvalue < 0 ) then
    if ( Yvalue < 0 ) then
        'heading = cint(angle) 'quadrant -X, -Y (181 to 270)
        heading = heading + 180
        'Debug.Print "3 ----heading = "; cstr(heading)

    end if
End if
if ( Xvalue >= 0 ) then
    if ( Yvalue < 0 ) then
        'heading = cint(angle) 'quadrant +X, -Y (271 to 359)
        heading = 360 + heading
        'Debug.Print "4 ----heading = "; cstr(heading)
    end if
end if
heading = (heading * (-1) +180 )

if heading < 0 then
heading = heading + 360
End if
```

'---- the following code displays the direction

```
if heading > 360 then
heading = 360
end if
if heading < 0 then
heading = 0
end if
        if heading < 360 then
            compdir = "N"
```

```

        end if
        if heading < 315 then
            compdir = "w"
        end if
        if heading < 225 then
            compdir = "S"
        end if
        if heading < 135 then
            compdir = "E"
        end if
        if heading < 45 then
            compdir = "N"
        end if
'----- stuff
battvoltage2 = cint(csng(battVoltage) * 0.489)
BattVoltage1 = csng(BattVoltage) * 0.00489 ' csng((csng(0.0489) * ))
SolarVoltage1 = csng(SolarVoltage) * 0.00978 '* csng(2)))
SatTemp1 = (((csng(SatTemp)* 0.00489)-0.5) / csng(0.01))

```

'----- The following debugs help with testing

```

'Debug.Print " PreHeader = ";
'Debug.Print cstr(PreHeader)
'Debug.Print " heder = ";
'Debug.Print cstr(header)
'Debug.Print " angle = ";
'Debug.Print cstr(angle)
'Debug.Print " X = ";
'Debug.Print cstr(MagValue_X)
'Debug.Print " Y = ";
'Debug.Print cstr(MagValue_Y)
'Debug.Print " Z = ";
'Debug.Print cstr(MagValue_Z)
'Call Delay(0.1)
'Debug.Print "END " 'cstr(heading)
'Debug.Print " new heading = "; cstr(heading)

```

'----- the following is the section that displays all information
each section corresponds to the number pressed on the display

```

call checkkey(pagenum,pagenumb4)
displaynew:

```

```

' Screen one home page
if pagenum = 1 then
Display1:
call clearscreen() ' clears the screen
Call Displayatcell("Magnetometer MQP", 1,2)
Call Displayatcell("Press", 2,0)
Call Displayatcell("2 for Heading", 3,0)

```

```
Call Displayatcell("3 for Health&Safety", 4,0)
end if
```

```
if Pagenumstart = 1 then
  If Pagenum = 10 then
    if Pagenum = 11 Then
      Goto display1
    End if
  End if
end if
```

```
' page two the heading dispaly
if pagenum = 2 then
call clearscreen()
Call Displayatcell("Packet Number ", 1, 2)
Call Displayatcell("heading", 3, 11)
Call Displayatcell("Current heading ", 2, 0)
Call Displayatcell("X = ", 3, 0)
Call Displayatcell("Y = ", 4, 0)
```

```
Call Displayatcell(cstr(xvalue), 3, 4)
Call Displayatcell(cstr(yvalue), 4, 4)
Call Displayatcell(cstr(heading), 4, 13)
Call Displayatcell(cstr(PacketCount), 1, 16)
Call Displayatcell(compdir, 2, 17)
```

```
end if
```

```
' page three the Health & Safety
if pagenum = 3 then
call clearscreen()
Call Displayatcell("Packet Number ", 1, 2)
Call Displayatcell(" Battery V = ", 2, 0)
Call Displayatcell("Solar Cell V = ", 3, 0)
Call Displayatcell(" Temp = ", 4, 0)
```

```
Call Displayatcell(cstr(PacketCount), 1, 16)
Call Displayatcell(Fmt(BattVoltage1, 2), 2, 15)
Call Displayatcell(Fmt(SolarVoltage1, 2), 3, 15)
Call Displayatcell(Fmt(sattemp1, 2), 4, 15)
end if
```

```
' calibration information
if pagenum = 4 then
call clearscreen()
Call Displayatcell("calibration mode max", 1, 0)
Call Displayatcell("Xmax =", 2, 0)
Call Displayatcell("Ymax =", 3, 0)
Call Displayatcell("Zmax =", 4, 0)
```

```
Call Displayatcell(cstr(Xmax), 2, 8)
Call Displayatcell(cstr(ymax), 3, 8)
Call Displayatcell(cstr(zmax), 4, 8)
end if
```

```
' calibration information
if pagenum = 5 then
call clearscreen()
Call Displayatcell("calibration mode min", 1, 0)
Call Displayatcell("Xmin =", 2, 0)
Call Displayatcell("Ymin =", 3, 0)
Call Displayatcell("Zmin =", 4, 0)
```

```
Call Displayatcell(cstr(Xmin), 2, 8)
Call Displayatcell(cstr(ymin), 3, 8)
Call Displayatcell(cstr(zmin), 4, 8)
end if
```

```
' this screen will sent information out the rs-232 port
if pagenum = 6 then
call clearscreen()
Call Displayatcell("Sending info", 1, 4)
'the debug statement sends current information out the RS-232 port
Debug.Print cstr(header);","; cstr(heading);","; cstr(MagValue_X);","; cstr(MagValue_Y);",";
CStr(PacketCount); ","; cStr(PacketsSent);","; Cstr(BattVoltage2)
```

```
end if
```

```
' Packet information
if pagenum = 8 then
call clearscreen()
Call Displayatcell("Info on radio ", 1, 0)
Call Displayatcell("send count =", 2, 0)
Call Displayatcell("receive count =", 3, 0)
```

```
Call Displayatcell(cstr(PacketCount), 3, 15)
Call Displayatcell(cstr(PacketsSent), 2, 16)
Call Displayatcell(cstr((packetcount * 100) \ PacketsSent ), 4, 8)
end if
```

```
' setup for the power calibration mode
if pagenum = 9 then
call clearscreen()
Call Displayatcell("Power Calibration ", 1, 0)
Call Displayatcell("don't touch ", 2, 0)
Debug.Print " Info , Hour , minute , BattVoltage1 , PacketCount , PacketsSent "
Hour = 0
Minute = 0
```

```
Second = 0.001
Call PutTime (Hour, Minute, Second)
pagenum = 100
end if
```

```
if pagenum = 12 then
call clearscreen()
end if
```

```
' run the power calibration mode
if pagenum = 100 then
call clearscreen()
Call Displayatcell("Power Calibration ", 1, 0)
Call Displayatcell("don't touch ", 2, 0)
Call GetTime(Hour, Minute, Second)
```

```
Debug.Print "Info"; ";"; cstr(Hour); ";"; cstr(minute); ";"; cstr(second); ";"; CStr(BattVoltage1); ";";
CStr(PacketCount); ";"; cStr(PacketsSent)
end if
```

```
skipprint:
'Call PutPin(17, bxoutputlow)
```

```
Loop
End Sub
```

```
' sub that checks the keypad
Public Sub checkkey(byref pagenum As byte, byref pagenumb4 As byte)
If getkeypad = 255 then 'if the key pad was not pressed return
pagenumb4 = 1
goto endcheckkey
end if
pagenumb4 = 0 ' key was presed
dim Keyvalue As Byte
Keyvalue = getkeypad ' used to see if new key was pressed
```

```
Select case keyvalue ' used to store key number
    Case 0
        pagenumb4 = 1
        call chgbirtLev() ' go to set Brightness sub
    Case 1
        pagenum = 1
    Case 2
        pagenum = 2
```

```

Case 3
    pagenum = 3

Case 4
    pagenum = 4

Case 5
    pagenum = 5

Case 6
    pagenum = 6

Case 7
    pagenum = 7

Case 8
    pagenum = 8

Case 9
    pagenum = 9

Case 10
    Pagenum = 10

Case 11
    pagenum = 11

Case 12
    Pagenum = 12

Case 13
    pagenum = 13
    call ClearScreen()

Case 14 ' goto get HELP sub
    pagenumb4 = 1
    call gethelp()

Case Else

```

```
End Select
```

```
endcheckkey: ' return back to the program
End sub
```

```

'----- this is a sub for ppl that need help
Public sub gethelp()
Dim Page As Byte
Dim PageMAX As Byte
dim Keyvalue As Byte
PageMAX = 3 ' number of pages

```

page = 0 ' number of pages

```
call clearscreen()  
Call Displayatcell("Help Page ", 1,6)  
Call Displayatcell("To view help ", 2, 0)  
Call Displayatcell("scroll down ", 3, 0)  
Call Displayatcell("To end press Enter", 4, 0)  
Call DisplayAtCell(Chr(3), 3,19)  
Call DisplayAtCell(Chr(2), 4, 19)
```

Do

' update the display

checkforkey:

keyvalue = getkeypad

if keyvalue = 255 then ' if no key is pressed then check again

call delay(0.01)

goto checkforkey

end if

Select case keyvalue ' update if key is pressed

Case 10 'up

if Page = 0 then

goto endcase

end if

Page = Page - 1

Case 11

if Page = Pagemax then

goto endcase

end if

Page = Page + 1

Case 15

goto endgethelp

Case 13

goto endcase

Case 14

goto checkforkey

Case Else

goto endcase

endcase:

End Select

'---- display pages

'display help information

Select case page

Case 0

call clearscreen()

Call Displayatcell("Help Page ", 1,6)

```
Call Displayatcell("To view help ", 2, 0)
Call Displayatcell("scroll down ", 3, 0)
Call Displayatcell("To end press Enter", 4, 0)
    'Call DisplayAtCell(Chr(3), 2, 19)
    Call DisplayAtCell(Chr(3), 3,19)
    Call DisplayAtCell(Chr(2), 4, 19)
```

Case 1

```
call clearscreen()
Call Displayatcell("Help Page ", 1, 4)
Call Displayatcell(cstr(Page), 1, 16)
    'Fill in help
    Call Displayatcell("Screen 1 is the ", 2, 0)
    Call Displayatcell("home page and will", 3, 0)
    Call Displayatcell("general information", 4, 0)

    Call DisplayAtCell(Chr(1), 2, 19)
    Call DisplayAtCell(Chr(3), 3,19)
    Call DisplayAtCell(Chr(2), 4, 19)
```

Case 2

```
call clearscreen()
Call Displayatcell("Help Page ", 1, 4)
Call Displayatcell(cstr(Page), 1, 16)
    ' Fill in help
    Call Displayatcell("about the different", 2, 0)
    Call Displayatcell("screens", 3, 0)
    Call Displayatcell("", 4, 0)

    Call DisplayAtCell(Chr(1), 2, 19)
    Call DisplayAtCell(Chr(3), 3,19)
    Call DisplayAtCell(Chr(2), 4, 19)
```

Case 3

```
call clearscreen()
Call Displayatcell("Help Page ", 1, 4)
Call Displayatcell(cstr(Page), 1, 16)
Call Displayatcell("The End", 4, 6)

    ' Fill in help
    Call Displayatcell("", 2, 0)
    Call Displayatcell("", 3, 0)
    Call Displayatcell("", 4, 0)

    Call DisplayAtCell(Chr(1), 2, 19)
    Call DisplayAtCell(Chr(3), 3,19)
    'Call DisplayAtCell(Chr(3), 4, 19)
    goto endpage
```

Case Else

```
goto endpage
```



```

endpage:
End Select
checkforkey1:
'wait for new key value
if keyvalue = getkeypad then
    call delay(0.1)
    goto checkforkey1
    end if
Loop

```

```

endgethelp: ' end help
call clearscreen()
End Sub

```

```

'----- This is a sub to change the brightness of the display
Public Sub chgbrtLev()
Dim Temp1 As Byte
Do
'----- Display information about current settings
if Register.OCR1AL < 10 then
    Call SetBrightness (0)
    end if
call clearscreen()
Call Displayatcell("Set Brightness Level", 1, 0)
Call Displayatcell("Level = %", 3, 4)
    Call DisplayAtCell(Chr(1), 2, 19)
    Call DisplayAtCell(Chr(3), 3, 19)
    Call DisplayAtCell(Chr(2), 4, 19)
Call Displayatcell(cstr((Register.OCR1AL * 100 \ 255)), 3, 13)

```

```

'----- Get updated information
checkforkey:
temp1 = getkeypad
    if temp1 = 255 then
        call delay(0.00001)
        goto checkforkey
    end if

```

```

Select case temp1
    Case 10 ' increases brightness
        if Register.OCR1AL > 240 then
            Call SetBrightness (255)
            goto endcase
        end if
        Call SetBrightness (Register.OCR1AL + 10)
    Case 11 'decrees brightness
        if Register.OCR1AL < 10 then

```

```

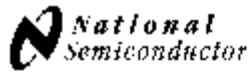
                Call SetBrightness (0)
                goto endcase
            end if
            Call SetBrightness (Register.OCR1AL - 10)
        Case 15 ' end
            goto endchgbrt

        Case 13 ' clear brightness
            Call SetBrightness (0)

        Case Else
            goto endcase
    endcase:
End Select
'----- wait for selection to end
checkforkey1:
temp1 = getkeypad
    if temp1 > 255 then
        call delay(0.00001)
        goto checkforkey1
    end if

loop
endchgbrt:
call ClearScreen() ' clear the screen befor going back
end Sub

```



March 2005

LM134/LM234/LM334 3-Terminal Adjustable Current Sources

General Description

The LM134/LM234/LM334 are 3-terminal adjustable current sources featuring 10,000:1 range in operating current, excellent current regulation and a wide dynamic voltage range of 1V to 40V. Current is established with one external resistor and no other parts are required. Initial current accuracy is $\pm 3\%$. The LM134/LM234/LM334 are true floating current sources with no separate power supply connections. In addition, reverse applied voltages of up to 20V will draw only a few dozen microamperes of current, allowing the devices to act as both a rectifier and current source in AC applications.

The sense voltage used to establish operating current in the LM134 is 64mV at 25°C and is directly proportional to absolute temperature (°K). The simplest one external resistor connection, then, generates a current with $+0.33\%/^{\circ}\text{C}$ temperature dependence. Zero drift operation can be obtained by adding one extra resistor and a diode.

Applications for the current sources include bias networks, surge protection, low power reference, ramp generation,

LED driver, and temperature sensing. The LM234-3 and LM234-6 are specified as true temperature sensors with guaranteed initial accuracy of $\pm 3^{\circ}\text{C}$ and $\pm 6^{\circ}\text{C}$, respectively. These devices are ideal in remote sense applications because series resistance in long wire runs does not affect accuracy. In addition, only 2 wires are required.

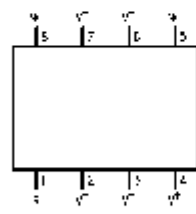
The LM134 is guaranteed over a temperature range of -55°C to $+125^{\circ}\text{C}$, the LM234 from -25°C to $+100^{\circ}\text{C}$ and the LM334 from 0°C to $+70^{\circ}\text{C}$. These devices are available in TO-46 hermetic, TO-92 and SO-8 plastic packages.

Features

- Operates from 1V to 40V
- 0.02%/V current regulation
- Programmable from 1 μA to 10mA
- True 2-terminal operation
- Available as fully specified temperature sensor
- $\pm 3\%$ initial accuracy

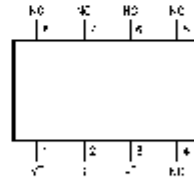
Connection Diagrams

SO-8
Surface Mount Package



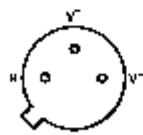
Order Number LM334M or LM334MX
See NS Package Number M08A

SO-8 Alternative Pinout
Surface Mount Package



Order Number LM334SM or LM334SMX
See NS Package Number M08A

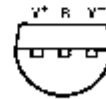
TO-46
Metal Can Package



V⁻ Pin is electrically connected to case.

Bottom View
Order Number LM134H,
LM234H or LM334H
See NS Package
Number H03H

TO-92 Plastic Package



Order Number LM334Z, LM234Z-3 or LM234Z-6
See NS Package Number Z03A



LT1300
Micropower High Efficiency
3.3/5V Step-Up DC/DC Converter

FEATURES

- Up to 220mA Output Current at 5V from 2V Supply
- Supply Voltage as Low as 1.8V
- Up to 88% Efficiency
- Small Inductor –10 μ H
- 120 μ A Quiescent Current
- Shutdown to 10 μ A
- Programmable 3.3V or 5V Output
- I_{LIM} Pin Programs Peak Switch Current
- Low V_{CESAT} Switch: 170mV at 1A Typical
- Uses Inexpensive Surface Mount Inductors
- 8-Lead DIP or SOIC Package

APPLICATIONS

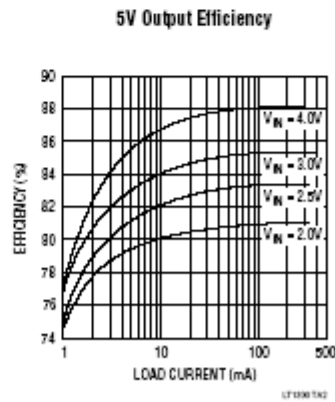
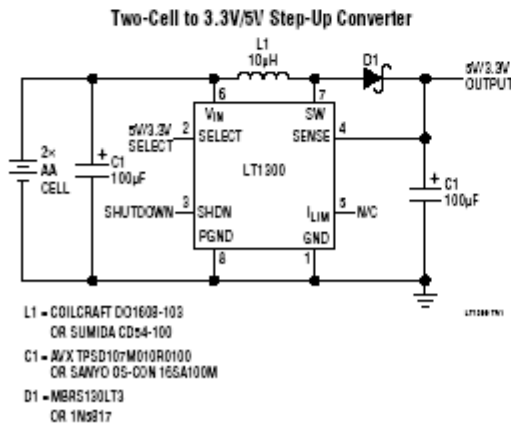
- Palmtop Computers
- Portable Instruments
- Bar-Code Scanners
- DC/DC Converter Module Replacements
- Battery Backup Supplies
- Personal Digital Assistants
- PCMCIA Cards

DESCRIPTION

The LT1300 is a micropower step-up DC/DC converter that utilizes Burst Mode™ operation. The device can deliver 5V or 3.3V from a two-cell battery input. It features programmable 5V or 3.3V output via a logic-controlled input, no-load quiescent current of 120 μ A and a shutdown pin which reduces supply current to 10 μ A. The on-chip power switch has a low 170mV saturation voltage at a switch current of 1A, a four-fold reduction over prior designs. A 155kHz internal oscillator allows the use of extremely small surface mount inductors and capacitors. Operation is guaranteed at 1.8V input. This allows more energy to be extracted from the battery increasing operating life. The I_{LIM} pin can be used to program peak switch current with a single resistor allowing the use of less expensive and smaller inductors and capacitors in lighter load applications. The LT1300 is available in an 8-lead SOIC package, minimizing board space requirements. For a 5V/12V Selectable Output Converter see the LT1301. For increased output current see the LT1302.

Burst Mode is a trademark of Linear Technology Corporation.

TYPICAL APPLICATIONS



Appendix F - LM7805 Data Sheet



April 1999
Revised December 2005

**LM7805 • LM7806 • LM7808 • LM7809 •
LM7810 • LM7812 • LM7815 • LM7818 • LM7824 •
LM7805A • LM7806A • LM7808A • LM7809A •
LM7810A • LM7812A • LM7815A • LM7818A • LM7824A**

3-Terminal 1A Positive Voltage Regulator

General Description

The LM78XX series of three terminal positive regulators are available in the TO-220 package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

Features

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 12, 15, 18, 24
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

Ordering Code:

Product Number	Output Voltage Tolerance	Package	Operating Temperature
LM7805CT	4%	TO-220	40°C - 125°C
LM7806CT			
LM7808CT			
LM7809CT			
LM7810CT			
LM7812CT			
LM7815CT			
LM7818CT			
LM7824CT	2%		0°C - 125°C
LM7805ACT			
LM7806ACT			
LM7808ACT			
LM7809ACT			
LM7810ACT			
LM7812ACT			
LM7815ACT			
LM7818ACT			
LM7824ACT			

LM7805 • LM7806 • LM7808 • LM7809 • LM7810 • LM7812 • LM7815 • LM7818 • LM7824 • LM7805A • LM7806A • LM7808A
 • LM7809A • LM7810A • LM7812A • LM7815A • LM7818A • LM7824A 3-Terminal 1A Positive Voltage Regulator

Appendix G - MicroMag3 Data Sheet

1008121 R01 – April 2005



MicroMag3

3-Axis Magnetic Sensor Module

General Description

The MicroMag3 is an integrated 3-axis magnetic field sensing module designed to aid in evaluation and prototyping of PNI Corporation's technology. The MicroMag3 combines PNI Corporation's patented Magneto-Inductive (MI) sensors and measurement circuit technology for unparalleled cost effectiveness and performance. The MI sensors change inductance by 100% over its field measurement range. This variable inductance property is used in a cost and space efficient ASIC (PNI 11096) which incorporates a patented temperature and noise stabilized oscillator/counter circuit with an SPI interface. The microprocessor compatible SPI interface allows easy access to the MicroMag3's measurement parameters and resulting field measurement data.

Advantages include 3V operation for compatibility with new systems, low power consumption, large signal noise immunity under all conditions, and a large dynamic range. Resolution and field measurement range are software configurable for a variety of applications. The measurement is very stable over temperature and inherently free from offset drift.

These advantages make PNI Corporation's MicroMag3 the choice for prototyping, education and low volume applications. A reference design is also available for use in high volume applications.



Features

- Low power: draws < 500 μ A at 3 VDC
- Small size: 25.4 x 25.4 x 19 mm
- Large field measurement range: $\pm 1100 \mu$ T (± 11 Gauss)
- High resolution field measurement: 0.015 μ T (0.00015 Gauss)
- Fast sample rate: up to 2000 samples/second
- Operation: 3.0 VDC
- Fully digital interface: SPI protocol at 3 V

Applications

- High-performance magnetic field sensing
- High-performance solid state navigation equipment for automotive, marine, and aeronautic applications
- Fast prototyping tool for PNI Corporation's Magneto-Inductive (MI) sensors

Ordering Information

Name	Part Number	Package
MicroMag3	12349	Each



PNI Corporation 133 Aviation Blvd., Ste. 101, Santa Rosa CA 95403-1084 USA

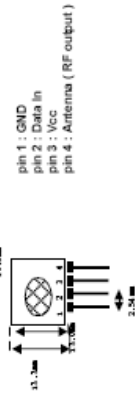
1

For more information, please call PNI Corporation direct at (707) 566-2260, email: sales@pnicorp.com, or visit PNI's website at <http://www.pnicorp.com>

Appendix H - TLP434/RLP434 Data Sheet

TLP434A & RLP434A RF ASK Hybrid Modules for Radio Control (New Version)

TLP434A Ultra Small Transmitter



**Easy-Link
Wireless**

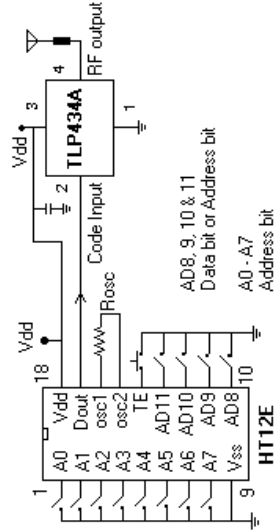
Frequency 315, 418 and 433.92 Mhz

Modulation : ASK
Operation Voltage : 2 - 12 VDC

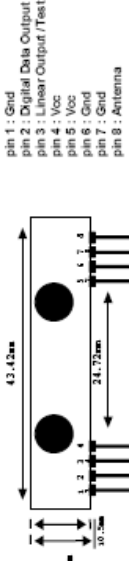
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Vcc	Operating supply voltage		2.0	-	12.0	V
Icc 1	Peak Current (2V)		-	-	1.64	mA
Icc 2	Peak Current (12V)		-	-	19.4	mA
Vh	Input High Voltage	Idata = 100uA (High)	Vcc-0.5	Vcc	Vcc+0.5	V
VI	Input Low Voltage	Idata = 0 uA (Low)	-	-	0.3	V
FO	Absolute Frequency	315Mhz module	314.8	315	315.2	MHz
PO	RF Output Power-50ohm	Vcc = 9V-12V	-	16	-	dBm
DR	Data Rate	External Encoding	512	4.8K	200K	bps

Notes : (Case Temperature = 25°C ± 2°C , Test Load Impedance = 50 ohm)

Application Circuit :
Typical Key-chain Transmitter using HT12E-18DIP, a Binary 12 bit Encoder from Holtek Semiconductor Inc.



RLP434A SAW Based Receiver



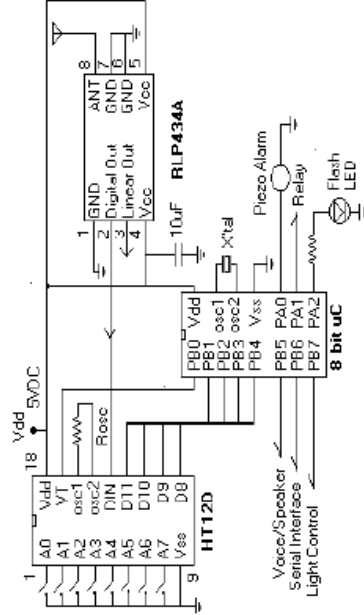
Modulation : ASK
Supply Voltage : 3.3 - 6.0 VDC
Output : Digital & Linear

Frequency 315, 418 and 433.92 Mhz

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Vcc	Operating supply voltage		3.3	5.0V	6.0	V
Icc	Operating Current		-	4.5	-	mA
Vdata	Data Out	Idata = +200 uA (High) Idata = -10 uA (Low)	Vcc-0.5	-	Vcc	V

Electrical Characteristics					
Characteristics	SYM	Min	Typ	Max	Unit
Operation Radio Frequency	FC	315, 418 and 433.92	-	-	MHz
Sensitivity	Pref	-110	-	-	dBm
Channel Width		+/-500	-	-	KHz
Noise Equivalent BW		4	-	-	KHz
Receiver Turn-On Time		5	-	-	ms
Operation Temperature	Top	-20	-	80	C
Baseboard Data Rate		4.8	-	-	KHz

Application Circuit :
Typical RF Receiver using HT12D-18DIP, a Binary 12 bit Decoder with 8 bit uC HT48RXX from Holtek Semiconductor Inc.



Laipac Technology, Inc.
105 West Beaver Creek Rd. Unit 207 Richmond Hill Ontario L4B 1G6 Canada
Tel: (905)762-1228 Fax: (905)763-1737 e-mail: info@laipac.com



Low Voltage Temperature Sensors

TMP35/TMP36/TMP37

FEATURES

- Low voltage operation (2.7 V to 5.5 V)
- Calibrated directly in °C
- 10 mV/°C scale factor (20 mV/°C on TMP37)
- ±2°C accuracy over temperature (typ)
- ±0.5°C linearity (typ)
- Stable with large capacitive loads
- Specified -40°C to +125°C, operation to +150°C
- Less than 50 µA quiescent current
- Shutdown current 0.5 µA max
- Low self-heating

APPLICATIONS

- Environmental control systems
- Thermal protection
- Industrial process control
- Fire alarms
- Power system monitors
- CPU thermal management

GENERAL DESCRIPTION

The TMP35, TMP36, and TMP37 are low voltage, precision, centigrade temperature sensors. They provide a voltage output that is linearly proportional to the Celsius (centigrade) temperature. The TMP35/TMP36/TMP37 do not require any external calibration to provide typical accuracies of ±1°C at +25°C and ±2°C over the -40°C to +125°C temperature range.

The low output impedance of the TMP35/TMP36/TMP37 and its linear output and precise calibration simplify interfacing to temperature control circuitry and A/D converters. All three devices are intended for single-supply operation from 2.7 V to 5.5 V maximum. The supply current runs well below 50 µA, providing very low self-heating—less than 0.1°C in still air. In addition, a shutdown function is provided to cut the supply current to less than 0.5 µA.

The TMP35 is functionally compatible with the LM35/LM45 and provides a 250 mV output at 25°C. The TMP35 reads temperatures from 10°C to 125°C. The TMP36 is specified from -40°C to +125°C, provides a 750 mV output at 25°C, and operates to 125°C from a single 2.7 V supply. The TMP36 is functionally compatible with the LM50. Both the TMP35 and TMP36 have an output scale factor of 10 mV/°C. The TMP37 is intended for applications over the range 5°C to 100°C and

FUNCTIONAL BLOCK DIAGRAM

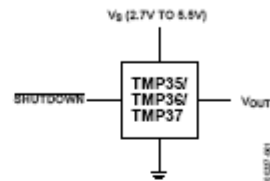


Figure 1.

PIN CONFIGURATIONS

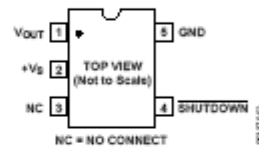


Figure 2. RP-5 (SOT-23)

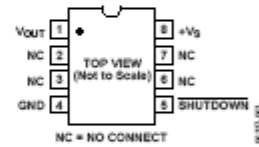


Figure 3. RN-8 (SOIC)



Figure 4. TO-92

provides an output scale factor of 20 mV/°C. The TMP37 provides a 500 mV output at 25°C. Operation extends to 150°C with reduced accuracy for all devices when operating from a 5 V supply.

The TMP35/TMP36/TMP37 are available in low cost 3-lead TO-92, SOIC-8, and 5-lead SOT-23 surface-mount packages.

Rev. D

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.461.3113 © 2005 Analog Devices, Inc. All rights reserved.

LCDX Hardware Guide

Ver 1.6



Appendix K - Heading and Calibration Primer

1001766 R01- Revised 13Jan04



Application Note: Multipoint Calibration Primer

Introduction

PNI Corporation's Magneto-Inductive (MI) sensors are based on patented technology that delivers breakthrough, cost-effective magnetic field sensing performance. These sensors change inductance with different applied magnetic fields. This variable inductance property is used in a patented temperature and noise stabilized oscillator/counter circuit to detect field variations. The PNI 11096 ASIC is the recommended implementation of this patented circuit, and can be used with the MI sensors to construct a magnetometer with up to 3-axes.

PNI Corporation's MI sensors employ a single solenoid winding for each axis and consume roughly an order of magnitude less power than conventional fluxgate or magneto-resistive technologies. The sensor coil serves as the inductive element in a very simple, low-power LR relaxation oscillator, with its effective inductance being influenced by the magnetic field component parallel to the coil axis. The frequency of the oscillation changes with the magnetic field.

The 11096 ASIC outputs a digital signal that can be fed directly into a microprocessor. This eliminates the need for any signal conditioning or analog/digital interface between the sensor and a microprocessor. The simplicity of the sensor circuit combined with the lack of signal conditioning makes the magneto-inductive sensor easier and less expensive to use in your product design than fluxgate or magneto-resistive sensors.

Magneto-inductive sensors and the 11096 ASIC are available from PNI Corporation through licensing agreements. Please contact PNI Corporation to discuss the possibility of a license agreement before designing these sensors into your products.

This primer will cover the basics of how to determine heading, calibrate out hard iron distortions, and perform sensor gain matching for a compass built with PNI Corporation's sensors and ASIC. More detailed information regarding the use of the ASIC and sensors can be found in their respective data sheets.

- PNI 11096 ASIC – *"PNI-11096 3-Axis Magneto-Inductive Sensor Driver and Controller with SPI Serial Interface"* (Document Number 1000005)
- SEN-S65 – *"SEN-S65 Magneto-Inductive Sensor"* (Document Number 1000619)
- SEN-L – *"SEN-L Magneto-Inductive Sensor"* (Document Number 1000620)



PNI Corporation 5464 Skyline Blvd., Ste. A, Santa Rosa CA 95405-1064 USA

1

For more information, please call PNI Corporation direct at (707) 566-2260, email: sales@pni.com, or visit PNI's website at <http://www.pnicorp.com>