EE 582

Final Project Report:

Airborne Sensor Platform

The Aero Group

03-03-00
Dreese Lab 260
10:30 am
The Ohio State University

Table of Contents

TABLE OF CONTENTS .... **I**

ABOUT THE AUTHORS .... **II**

INTRODUCTION **1**

PROJECT OVERVIEW **1**

PAYLOAD **4**

NAVIGATION **8**

COLLISION AVOIDANCE **9**

APPLICATIONS **14**

POWER **15**

MICROCONTROLLER SELECTION **18**

MOTOR CONTROL **21**

COMMUNICATION **22**

VIDEO **29**

USER INTERFACE **33**

CONCLUSION **42**

RESOURCES **46**

INDEX **47**

About the Authors

The Aero Group

Edward Murphy – Project Leader……………………………… murphy.294@osu.edu

Ed is a senior at The Ohio State University, College of Engineering, Department of Electrical Engineering. He is currently working towards a BS in Electrical and Computer Engineering. Ed is also a member of the Information Electronics Group, which is part of the Mixed Signals Electronic Systems Laboratory, where he works under the guidance of Dr. Steve Bibyk.

Ed's work on this project focused primarily on the microcontroller and sensor interfacing. He was also chiefly responsible for group operations.

John Federico – Assistant Project Leader………………………..… federico.8@osu.edu

John is a senior at The Ohio State University, College of Engineering, Department of Electrical Engineering. He currently has a JD in Law and is seeking a BS in Electrical and Computer Engineering. In addition to his studies at OSU, John also works for a local law firm as a practicing attorney.

John has been working on the User Interface and Video integration. He is also the group's Webmaster.

Mark Smertneck – Treasurer……………………………………. smertneck.1@osu.edu

Mark is a senior at The Ohio State University, College of Engineering, Department of Electrical Engineering. He is currently working towards a BS in Electrical and Computer Engineering. Mark also works part-time as an Exhibit Technician at the Center of Science and Industry in Columbus, Ohio.

In addition to seeking financial support for the project, Mark is also responsible for procurement. He also has acted as the group's graphics artist for our presentations. Mark is currently working on the Control and Power subsystems.

Ruth Gebru – Archivist………………………………………………….. gebru.2@osu.edu

Ruth is a senior at The Ohio State University, College of Engineering, Department of Electrical Engineering. She is a full-time student, working towards a BS in Electrical and Computer Engineering.

Ruth has been responsible for recording all group minutes as well as coordinating information via email. Her research for this project has focused on the video subsystem and User Interface. Ruth is also responsible, in part, for the project schedule.

Chi Chiu Cheung ………………………………………………….. cheung.47@osu.edu

Chi Chiu is a senior at The Ohio State University, College of Engineering, Department of Electrical Engineering. He is a full-time student, working towards a BS in Electrical and Computer Engineering.

Chi Chiu has been solely responsible for the Communications development to this point. In addition to investigating possible hardware solutions for the wireless link, he has also implemented two communications protocols.

Ardhi Harianto …………………………………………………... harianto.4@osu.edu

Ardhi is a senior at The Ohio State University, College of Engineering, Department of Electrical Engineering. He is a full-time student, working towards a BS in Electrical and Computer Engineering.

Ardhi is currently working with the Sensor and Control subsystems. His primary focus is on the collision avoidance system. He has been responsible for all research on both the infrared and ultrasonic sensors.

Carissa Sipp ………………………………………………………… sipp.3@osu.edu

Carissa is a senior at The Ohio State University, College of Engineering, Department of Electrical Engineering. She is a full-time student, working towards a BS in Electrical and Computer Engineering.

Carissa's work on the project has included research on possible solutions for our projects power requirements. To this point, she has investigated various battery and photovoltaic technologies. She is primarily responsible for the Power subsystem. Carissa is also responsible, in part, for the project schedule.

Introduction

Our team has been tasked with developing a miniature, airborne, sensor platform. In this paper, we wish to layout our vision for this project and report on our research to this point. Specifically, we would like to give you an overview of our project goal, our prototype for proof of concept, possible sensor configurations, end uses for our product, some design considerations, our project schedule and some information that we have gathered for our subsystems.

Project Overview

Our research has shown that there is considerable interest, in the area of unmanned autonomous vehicles (**UAV**) and remote sensing platforms. So far, most of the research in this area has focused on robotic fixed wing aircraft that utilize a combination of solar and battery sources for powered flight. Unfortunately, in order to provide enough energy for long range missions and extended on station times, they have used large solar arrays and heavy battery storage.

In order to meet the requirement for this platform to be as small as possible, we feel that the ideal configuration of the airframe would be a lighter-than-air vehicle. By generating lift with helium, instead of forward motion, we have considerable energy savings over conventional, powered flight. Additionally, the helium affords our aircraft the ability to hover and travel at low speeds, which gives us even better performance when a mission requires extensive time on station or operation in confined space.

There is, however, one problem in using an LTA. When we scale down the design, the blimps cross-sectional area presents us with stability issues in even the mildest of winds. Therefore we propose a flying wing filled with helium for the project's final form. This design will afford us the required volume we need for the helium, yet reduce the vehicles overall cross section.

After our proof of concept flight, we intend to consult with our Aeronautical Engineering Department regarding design of a custom airframe. With the energy savings provided by the airframe and careful electronics design, we feel that it will be possible to produce a vehicle on a sub-meter scale that could offer an array of sensor configurations, autonomous control, and exceptional endurance, all at a cost much less than current airborne sensor platforms.

In order to focus our attention on the electronics and control systems of this vehicle, we will be using a commercially available micro-blimp for our airborne platform. So far, we have looked at several available models that range from a 58-inch blimp with a conventional shape, to an 84-inch model in the form of a flying saucer. Each of these vehicles can be purchased with motors and servos already included. There are also options, such as two and three-channel radio control and on board video.

After contacting the manufacturers of blimps that met our preliminary requirements, we were able to get a better idea of what our needs would be. A sales representative at **DraganFly Innovations Inc**. confirmed our estimation that we would need a saucer blimp with at least an 8-foot radius to lift the additional payloads we wanted to install. To purchase a vehicle that already contained the motors, video and batteries would drive costs into the range of $3500. If we wished to purchase nothing but the precut Mylar bag would still cost $300 and we would still need to purchase the other components. One other option would be to purchase large sheets of Mylar and construct our own blimp from scratch, but this would subtract from our time available for electronics design and implementation.

For these reasons, we have chosen to change our approach to a three-tier design. Initially, we would like to design and build a platform with on board video and GPS. In addition, we would like to include some proximity sensors, to provide us information regarding the vehicle's surroundings. We intend to demonstrate some forms of environmental sensing and vehicle control abilities. One possible form for this stage may be a simple radio controlled car or perhaps a tracked vehicle. The design considerations would be the same as those that will be needed for later stages, with the exception of a less restrictive weight limit. After demonstrating this design, we could then use this prototype to garner support for the remaining stages.

With a larger budget and preliminary research and development completed, we feel that the next stage of the project should then be to move our system to one of the commercially available blimps that we discussed earlier. The major focus for this design would then be to further reduce the power and weight needed for our electronics and refine the sensor payload capabilities.

The last stage would then be to begin working with the aeronautical engineers on a final form for the vehicle. After the aeronautical design concerns are better understood, then work could begin on further reducing the electronics by designing and fabricating custom VLSI chips.

As for the power source, we will be using a combination of solar power and batteries, to prepare this vehicle for extended operation. Our design will include a high-bandwidth RF communication link for control and sensor communication. In future stages, we will incorporate video transmission into this link as well, but for the initial stages, we would like to use a separate link for the video. This will allow us to purchase off-the-shelf equipment for components we will be refining later and give us the opportunity to develop the control and sensor subsystems in time for our demonstration.

Payload

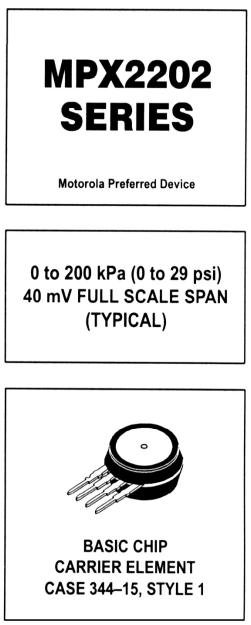
Our goal is to develop a system that will allow us to offer our customers as flexible configurations as possible. We want to design a generic analog front end so that we can use any number of sensor types as long as they meet weight and power consumption restrictions. Some of the sensors that we have discussed include:

- Chemical sensors such as Carbon Monoxide, Natural Gas and Radon
- Weather sensors such as Temperature, Pressure and Humidity
- Radiation sensors such as Radio, Radar and Microwave

Also, the on board video can be replaced or supplemented with infra red and ultra violet detectors. For our first demonstration then we have chosen to include a temperature sensor, a pressure sensor, a radar detector, and a smoke detector.

For the temperature sensor, we will be using the **National Semiconductor LM34**. This sensor is cheap, readily available and used in a number of circuits classes so we feel it is a good choice. The interfacing is straightforward. The chip has only three leads, a range of –50° to 300° Fahrenheit. Accuracy is ± ½° for temperatures near room temperature and ±1½° for the full range. The chip operates with voltage ranges of 5 to 30 volts and only draws 70 µA on average.

For the pressure sensor, we have chosen the **MPX2202 from Motorola's DigitalDNA™** line. It features on-chip temperature compensation and calibration. It has a range of 0 to 29 psi, operates at 10V and draws 6 mA. Once again the interfacing is straightforward. As Figure 1 shows, this chip has only 4 leads. One supply, one ground and a positive and negative output pair. This sensor is available in three different modes. It can operate as a differential, absolute, or gauge sensor. For our purpose we will be using the gauge version. As pressure is asserted on the pressure side, voltage increases linearly in proportion to the difference between the pressure sensed and the vacuum reference.



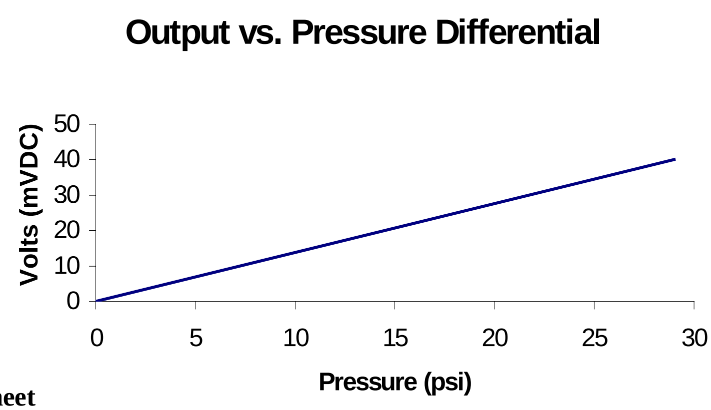


Figure 1. Motorola Datasheet

The sensor can also detect vacuum increases asserted on the opposite side of the sensor. In this case, the sensors negative output will increase linearly with the increased vacuum. Typical output data for this sensor is also represented in the Figure 1.

Our radar detector will be a very simple circuit consisting of a few capacitors, one of which will be our radar "antenna", a dual operational amplifier, and a few resistors. This circuit will be a modified version of one that appeared in ***"Detector Circuits"*** by Rudolf F. Graf.

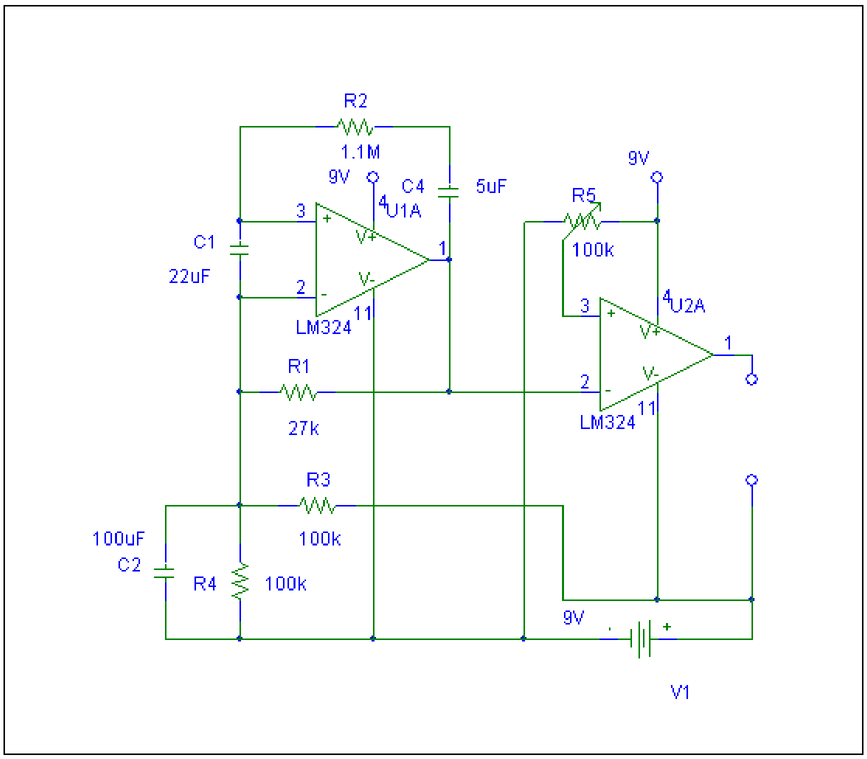


Figure 2. Single IC Radar detector 50Mhz to 500Ghz

The capacitor (C1) acts as the radar antenna. This can be tuned within a range of 50 MHz to 500 GHz by simply changing the length of the capacitor leads. Figure 3. shows the relationship between the lead length and the frequency that can be detected.

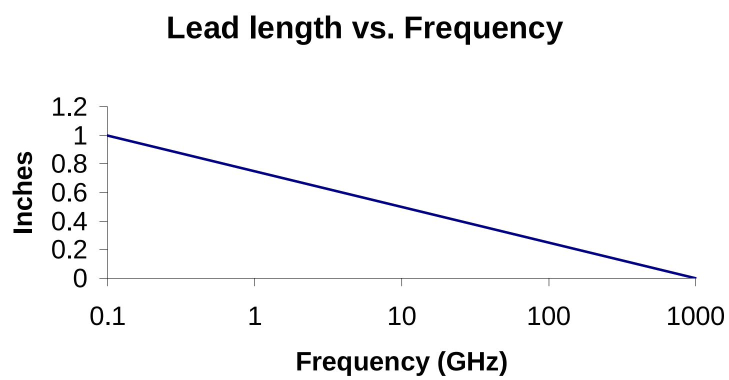
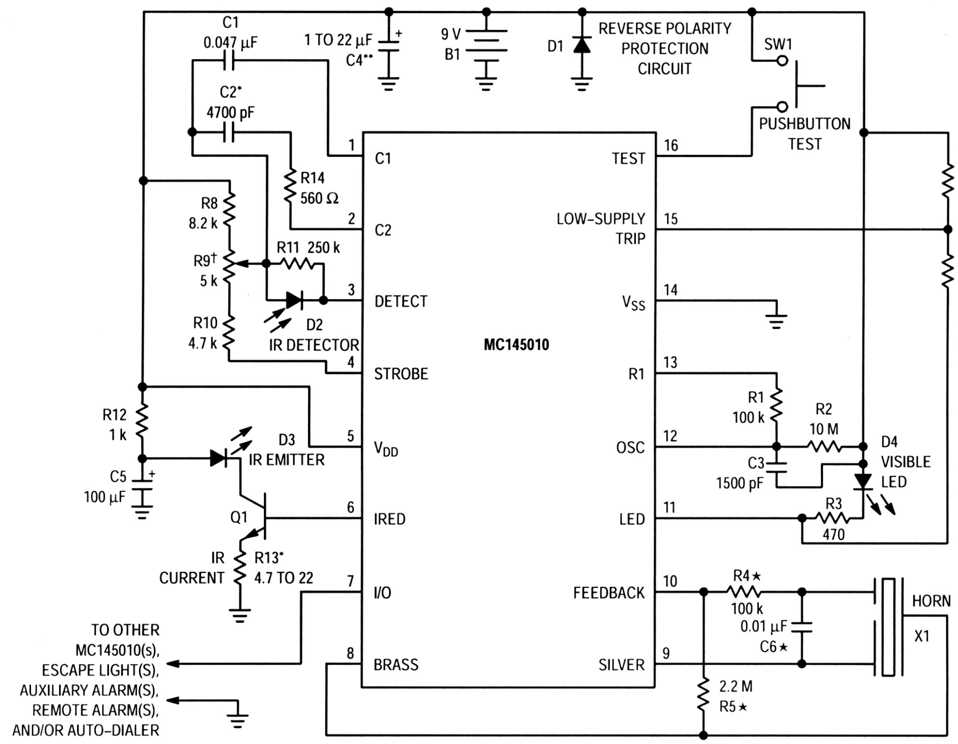


Figure 3. Lead length

In this circuit, the operational amplifier (U1A) acts as a current-to-voltage converter. Once the signal received on the capacitor exceeds the trip point, the operational amplifier will then trigger the second operational amplifier, which acts as an output buffer. This output can then be read by the MCU and appropriate information would then be sent to the operator.

We are also considering a smoke detector, to demonstrate a possible application as a backup to fixed alarm systems. For this, we have found an IC from Motorola that only requires an additional IR smoke chamber to make a complete smoke detector. Shown here, the **Motorola MC 154010** (Fig. 4) is incorporated with some addition components required for adjusting sensitivity of the detector, providing local alarm with the LED and the horn. There are also some protective circuits and testing mechanisms as well.

MC145010



★Values for R4, R5, and C6 may differ depending on type of piezoelectric horn used.
* C2 and R13 are used for coarse sensitivity adjustment. Typical values are shown.
†R9 is for fine sensitivity adjustment (optional). If fixed resistors are used, R8 = 12 k, R10 is 5.6 k to 10 k, and R9 is eliminated.
When R9 is used, noise pickup is increased due to antenna effects. Shielding may be required.
**C4 should be 22 µF if B1 is a carbon battery. C4 could be reduced to 1 µF when an alkaline battery is used.

Figure 4. Motorola Datasheet MC145010

As we see in the schematic from the datasheet, the MC145010 offers simple interfacing and scalability. For our project we could then connect the chip's I/O pin to our MCU which will in turn transmit the alarm to the operator via the communications link.

Navigation

For navigation, we feel that the simplest and most cost-effective way to report the vehicle's location will be through the use of a onboard GPS receiver. When compared to gyroscopic inertial navigation coupled with a compass, the GPS performs much better because "gyro drift" is not an issue. Additionally, with the availability of single chip

GPS receivers, integration and interfacing become straightforward procedures. The GPS also gives us much more information than other navigation systems. For example, we can monitor the vehicle location, speed, acceleration, direction, and rate of climb.

Our research has shown that the **SiGEM SGM5608PM** will give us the information we want at an affordable price. The module and antenna (Fig. 5) make up the entire onboard 12-channel GPS system. The module communicates through a built-in serial port, which we will then interface with our MCU.

Figure 5. SiGEM Datasheet SGM5600KS

Collision Avoidance

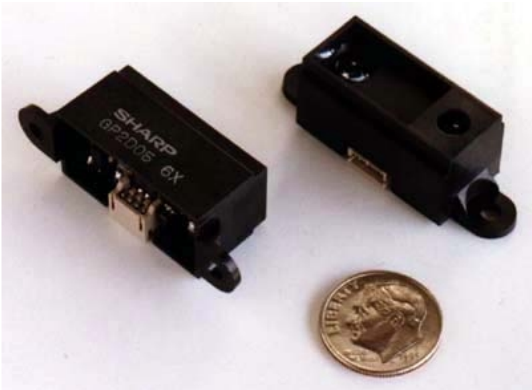
Another type of sensors, that we are planning to install on our platform, are proximity sensors. The purpose for these sensors is collision avoidance. Obviously, we don't want our vehicle to hit anything while it's flying. Therefore, these sensors are crucial to our project.

The basic idea is that we want our microcontroller to send out a signal, and to wait for it to bounce off something and come back. If the signal comes back, and is of ample strength, we know that there is a solid object somewhere around our vehicle (i.e.

wall, person, etc.). If we don't hear back from it, it means that the path is clear, and we can continue going.

These proximity sensors will have the capability to detect any object or barrier in six directions around our vehicle, and measure their distance. We will program the microcontroller in such a way that when the vehicle is too close to any object, the MCU will automatically stop the motor that is thrusting in that direction, disable any command that is controlling that motor, and reverse direction as needed. With the existence of these sensors, a smart and safe vehicle can be realized.

Our team has come up with two methods of dealing with the problem of obstacle avoidance. The first is using infrared (IR), and the second ultrasonic, or called Sonar. Our example for the IR sensor is the **SHARP GP2D05**, which is shown in the figure below:



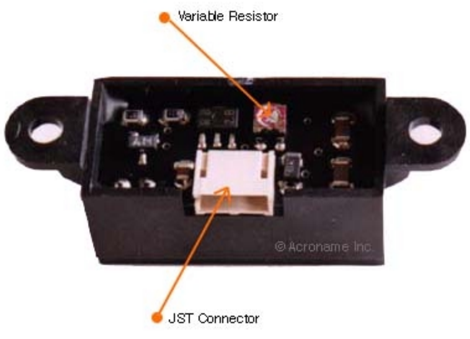


Figure 6. SHARP INFRARED SENSOR GP2D05

Pictures taken from ***ACRONAME, Inc***.(http://www.acroname.com/robotics/parts/R21-IR05.html).

And for the ultrasonic proximity sensor, we are looking at the **POLAROID 6500 SERIES Sonar Ranging Module**, shown below:

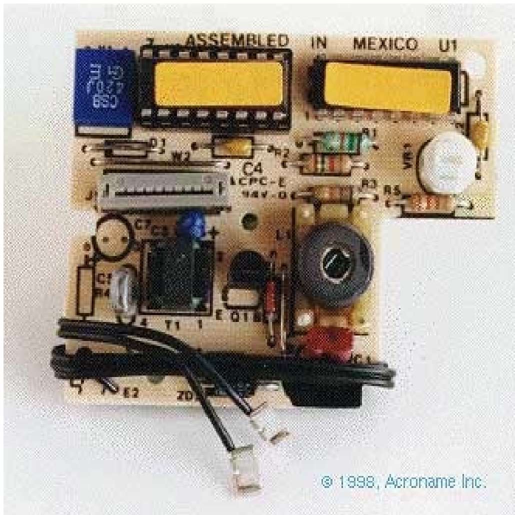


Figure 7. POLAROID 6500 SERIES SONAR RANGING MODULE

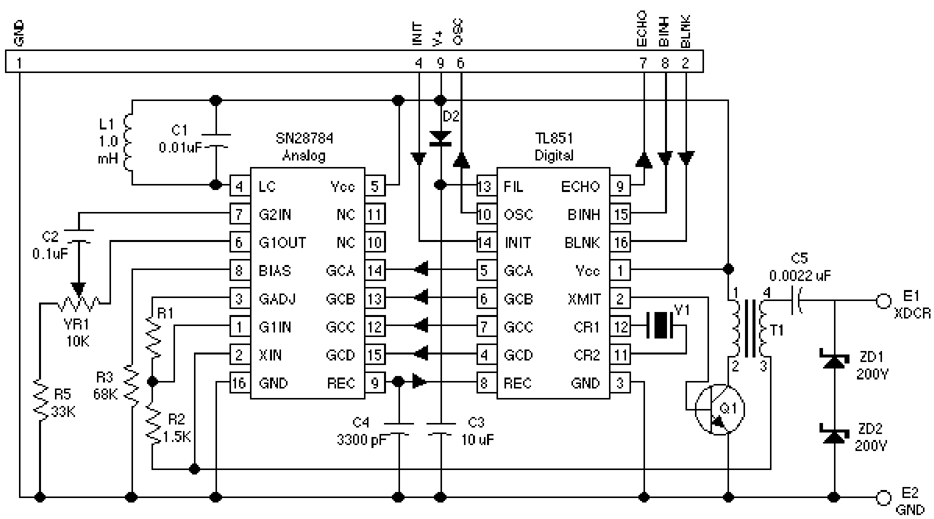


Figure 8. POLAROID 6500 SONAR RANGING MODULE SCHEMATIC

*Pictures taken from **ACRONAME, Inc**.*(http://www.acroname.com/robotics/parts/R11-6500.html)

As for now, these two parts are still very preliminary, but as we continue, we will choose the one that best suits our needs, budget, and vehicle capability.

Some of the advantages and disadvantages of the two sensors above are described in this table below:

Table 1. Proximity Sensor comparison

Infrared (SHARP GP2D05)	Ultrasonic (POLAROID 6500 SERIES)
Advantages : • Small in size. • Light weight (only 0.2 ounce). • Cheap prize (around $20). • High precision in measuring distance. • Easy to interface with the microcontroller. • Low average current dissipation (only 24 mA max., and 0.3 μA at OFF-state)	***Advantages :*** • Have a much longer range (from 6 inches to 35 feet). • Measurement accuracy is within ± 1%. • It is TTL compatible.
Disadvantages : • **Very short range** (only 80 cm max.) • The connectors on the units are non-standard sizes.	***Disadvantages :*** • Bigger size, need a transducer, therefore more weight (around 0.6 ounce) • Draws too much current (2000 mA during transmit and 100 mA after transmit). • Expensive (around $50).

Both sensors have the same voltage requirements for operation. Therefore, we are evaluating different characteristics of their abilities. The SHARP GP2D05 infrared proximity package is small, light-weighted, cheap, accurate, and easy to interface with the microcontroller. Moreover, it has a low average current dissipation. However, its major disadvantage is that this sensor has a very short range. It can only emit the infrared beam to a maximum of 80 cm. Therefore, if our vehicle happens to fly at high speed, the range will not great enough to give the control system enough time to reverse-thrust and avoid the collision with the barrier. Another minor disadvantage is the connector. It is made in Japan, and it does not have a standardized connector, therefore it's going to be rather hard to find one in the market here.

With the POLAROID 6500, we get much better distance measurement, which is adjustable from 6 inches to 35 feet. Therefore, it allows the vehicle to reverse-thrust, to stabilize itself, or anything else it needs to do to avoid the collision. Its accuracy is within ± 1%, and it is TTL compatible. However, it draws so much current during transmit (2000 mA), and even after transmit (100 mA), much more than what the infrared sensor needs. It is 0.4 ounce heavier than the IR, and it is expensive, too (around $50 for a package including module, transducer, and necessary wires).

Once again, just about any sensor that detects at a distance and meets certain physical constraints could be used. We are planning to put a lot of effort in assuring that our design will be as robust as possible so that our vehicle will be attractive to a larger market and have many possible end uses.

Applications

As mentioned earlier, our proposed machine is a UAV that can be suited with multiple sensors as requested by the customer. There is an extremely wide range of uses for such a versatile machine. Here its possible uses are broken down into five categories:

- Military
- Entertainment
- Law enforcement
- Corporate
- Individual

In military use, there are such probable uses as searching for land mines, gathering intelligence information and detecting poisonous gases. In addition, with the appropriate sensors in place, the UAV could be sent to detect radar signatures, radio sources and troop movements

Within the entertainment world uses include news gathering and major news story coverage, sporting events (instant replay), and major glamour or pop culture spectaculars such as the Grammy's, Golden Globes, and MTV awards.

In law enforcement this machine has unlimited possibilities to help with radar and traffic enforcement; surveillance; and search and rescue. When compared to the costs of operating a Police helicopter, which runs in the hundreds of dollars per hour, our vehicle would represent a popular augmentation or replacement for simple airborne reconnaissance.

For corporate use, it can be used in amusement parks and malls to search for lost children and help in loss prevention. Also it can be used at business offices for security purposes.

Finally, use at home is a long-range goal. Once in mass production, we think that the price will decrease enough that individuals can buy our machine for uses such as: personal security, detection of poisonous gas such as CO, and to keep an eye on their kids when they can't.

Power

When you consider weight, one of the heavier items onboard will be the battery, which ties in to the next challenge – Power. A three-ounce battery will put out 4 mAh at 12 volts. This only gives us about 1 mAh for all of our circuitry for a 4-hour flight. This is not a lot when you consider all the electronics that we are putting on this vehicle. Ideally we would like this power to come from both solar cells and a rechargeable battery. With this arrangement in mind, we need to incorporate onboard battery charging capabilities. Also we need to supply the ship with enough output power to run such items as the sensors and motor.

When choosing our battery we had to consider cost, recharging capacity, and weight. There are three rechargeable battery types on the market: Ni-Cd, Ni-MH, and

Li-ion. Among these three, we focused our search on the two most durable: Ni-Cd and Li-ion. Li-ion is on average half the weight of Ni-Cd and the best rechargeable batteries available. They offer long life and high capacity in slim, lightweight batteries that are memory-free. The biggest drawback to this type of battery is the higher cost. Despite the budget constraints we have chosen to go for a Li-ion battery with its many benefits outweighing the cost.

To power our UAV we have calculated we will presently require two 1100mAh batteries. One battery we are considering is used in the **Qualcomm QCP820**. It is a Li-ion 1100mAh battery that costs $69-95 per battery and weighs in at 37g. Another battery we have considered is the **Ultralife U9VL9V** lithium battery. It weighs in at 34.4g and delivers a punch of 1200mAh at 900 ohms to 5.4V. At this time, we have still been unable to find exact data on several aspects of this battery such as its rechargeability and its cost. Due to these preliminary findings we are leaning toward the Qualcomm battery, with further research needed.

For our dropout power management we are considering using a low dropout 200mA Linear Regulator from **Unitrode**. We have specifically focused on the **UCC387** of the Unitrode positive linear pass regulator series. These regulators are specifically designed for Li-ion batteries and contains a chip resistor network for preset to regulate at a fixed 5V output. Other features that focused our attention toward this regulator were its logic shutdown capability; guaranteed reverse input/output voltage isolation with low leakage; short circuit protection and duty cycle limiting; and remote load voltage sensor for accurate load regulation.

To help control battery operation we have a Li-ion linear charge management IC. For this component, we have chosen the **Texas Instrument Benchmarq bq2057** series of advance Li-ion linear charge management IC. These IC's are designed specifically for weight-sensitive compact portable electronics. Several of the attractive features that drew us to this IC were its specific design for single-and dual-cell Li-ion packs, dropout voltage as low as .3V, dynamic compensation of the battery pack's internal impedance, and its capability to measure battery temperature before and during charge. Along with these helpful attributes, this powerful chip has integrated cell conditioning for reviving deeply discharged cells and minimizing heat dissipation during the initial stage of charge.

To actually charge out battery onboard our ship we have chosen a Unitrode switch mode Li-ion battery charger controller. This chip will hopefully allow us to recharge our Li-ion battery to maximum capacity while the ship is powered by our chosen solar cells. Our charger is from the **UCC3956** family of battery charger controllers. This charger is especially designed for single cell or multiple Li-ion battery packs. Several desirable features on this charger are its resistor programmable charge currents, fully differential switch mode current sensing, and fully differential switch mode current sensing.

Finally, for our solar cells we have chosen the **Pegasus Bifacial High-Efficiency One-Sun Silicon Solar Cell from Emcore**. These cells produce 369.6 mV/g and each cell, including its by-pass diode, weigh around 2.37g and are 21.96 cm^2 in area and

40mg/cm^2 in specific weight. The Pegasus cell is designed using interdigitated back contact with no front side grids. They are fabricated using thin monocrystalline silicon wafers and have a double-layer of anti-reflective coating. These cells were used on NASA's solar plane the Pathfinder which achieved an altitude record for a propeller-driven aircraft on August of 1998 when it flew over 80,000 feet. The drawback for such an advanced solar cell is the cost at $250/watt. Other cheaper solar cells that are the size of those you see on a solar calculator produce a current of 62.5mA @ 5V. At only $.50 each though their use is not out of consideration with budget constraints hovering over our heads.

Microcontroller Selection

There are a few reasons that we selected the **PICmicro™** microcontroller from **Microchip Technology Inc.**, over the **MIT Handy Board**. The most important two reasons were weight and cost. The Handy Board has a lot of features included and all of that comes at a cost in terms of price and weight. However these were not the only reasons that we decided that the PICmicro™ was the appropriate solution.

The PICmicro™ family of microcontrollers has been growing in popularity among the microcontroller marketplace and with good reason. They offer a number of features that allow the programmer far more options than the microprocessors of not so long ago. For instance, the **PIC16F874** has onboard A/D conversion, built in USART, five I/O ports (made up from a total of 33 dedicated I/O pins), 12 available interrupts, and more. The chart below, while informal in nature, gives a strong argument for the mighty PICmicro™.

Table 2. Microcontroller comparison

PIC 16F874	Handy Board
Advantages -Cost (around $8) -Weight – a few grams - Wide operating voltage (approx. 2 to 6 volts) - Multiple I/O ports - A/D conversion (8 to 10 bit ports) - Flash memory is easy to program (no EEPROM eraser required) - Fast! 20 MHz clock, with 5 MHz per instruction execution	Advantages -C compiler based programming language - Built in motor controller, A/D hardware - Plenty of built in I/O, little external hardware required
Disadvantages - PIC assembly language can be difficult - C compiler not free so it will be necessary to use assembly	Disadvantages - Cost ($190 to $300) - **Weight** (more than our project allows for total cargo - Requires 9 volts to operate - Slower system clock (2 MHz)

We have two uses for our microcontroller. We need it to control the motors and to handle communications. With one microcontroller handling the encoding and decoding of communications and another microcontroller controlling the motors for moving the UAV we will have plenty of processing power in small, lightweight and inexpensive packages. The PICmicro™ is a RISC (reduced instruction set computer) with only 35 commands to master. This is helpful because it reduces the difficulty normally associated with a CISC (complex instruction set computer). This helps us because we will not have to purchase the C compiler for programming and can use the free editor/compiler/simulator that Microchip provides. The architecture of the PICmicro™ is more complex than the average microcontroller as far as the register set (over 50 registers in the PIC16F874) but that is less complex than say the **Motorola MC68HC09** with six different addressing modes for memory which apply differently to different commands.

So the PIC16F874 is a solid choice for our project and will provide us with the functionality that our group requires within the specification our project needs.

When building our first iteration we realized that in the limited time frame true autonomous control may not be possible and that we may end up with object avoidance as the sum of our vehicle's autonomous control until later revisions. Once again the PIC16F874 offers us the opportunity to upgrade the system when it becomes necessary without sacrificing equipment purchased for the first iteration. For example if a standard three-channel radio were purchased, as is used on many remote control hobby vehicles, we would have to maintain an implementation that will work within the boundaries of that system. By using a PICmicro™ with an **AeroComm, Inc.** or equivalent transceiver we have the option of software upgrades changing the functionality with little or no hardware changes required. Also complex solutions for maneuvering the vehicle through a combination of user input and current sensor data or solutions for things like turning the vehicle without changing it's position are possible where a simple radio would create the need for a very skilled user to perform the same maneuvers. Additionally, if we can use the same transceiver for the video and sensors, we remove the need for a three-channel receiver onboard the UAV. This allots us significant weight/power savings. In conclusion the microcontroller will solve challenges that we face now as well as challenges which we must address in the future.

Motor Control

Deciding to use the PIC16F874 to control our motors does not limit us as to how the PICmicro™ will control our motors. There are a number of different solutions ranging from ON/OFF through the application of DC to more complex and elegant solutions such as Pulse Width Modulation.

The solution chosen must take into consideration that the UAV needs to be able to maintain position with no other input from the user other than a simple instruction to do so. This means that even such a simple operation will require that the UAV must have on board processing power with the ability to provide complex solutions to the motor driver problem. Instances such as low power will require decreased power consumption and limiting the available speeds will be a possible solution to such a problem.

Considering the need for expanding the capability of the motors from a single speed to multiple speeds and even reverse speeds also inspires the need that a PICmicro™ can provide. Two possible solutions are the before mentioned pulse width modulation and another where an eight bit digital output is turned into an analog signal with 256 distinct operational speeds. Consider first pulse width modulation. This is a method where the duty cycle of a DC signal is manipulated such that the effective power applied is between 0 and 100%. Of course the signal out is still incremented in discrete measures and will therefore not be perfectly analog. The eight bit analog solution provides yet another benefit. A zero voltage point may be defined giving positive and negative voltages or forward and reverse speeds. Consequently the UAV can be set up using the motors operating in both directions instead of limiting operation in a single direction. This expands the range of solutions for motion control significantly through the use of software and limited hardware, which fits the stringent needs of the project.

Communication

Table 3. AeroComm PKLR2400 Datasheet

PKLR2400 Specifications

2. PKLR2400 Specifications

GENERAL	
Bus Interface	Serial (TTL Level Asynchronous) or Parallel through 40 pin mini connector. Amp P/N 177986-1 or Berg
Interface Data Rate	
Serial	Prog to 882 kbps. Supports PC rates to 57.6 Kbps
Parallel	4 Mbps
Compliance	Certifiable under: US - FCC15.247 Canada - DOC Europe - ETSI Japan - MKK
Power Consumption	
TX/RX Active (All Modes)	115mA typical
Interface ON/RF OFF (API Mode Only)	45mA typical
Sleepwalk (All Modes)	35mA typical
Standby (API Mode Only)	30mA typical
Channels	Supports 77 non-interfering channels
Security	User assigned System ID. Unique IEEE address on each radio.
RADIO	
Frequency Band	2.402 – 2.478 GHz
Radio Type	Spread Spectrum Frequency Hopping
Output Power	10mW
Voltage	5V nominal ±5%, ± 50mV ripple
Sensitivity	-90dBm
Data Rate	1Mbps
Range	Indoors up to 300ft, Outdoors up to 3000ft Can be extended with directional antenna
ENVIRONMENTAL	
Temperature (Operating)	0 to +60 C
Temperature (Storage)	-50 to +85 C
Humidity (non-condensing)	10% to 90%
PHYSICAL	
Dimensions	1.65" x 2.65" x 0.20"
Antenna Connector	Standard MMCX jack
Weight	Less than 0.5 ounce
SOFTWARE	
User Configurable Options	
Host Interface Data Rate	Up to 800Kbps
Variable Packet Length	Up to 2Kbyte
Serial Interface Modes	(3) Transparent and (1) API
Diagnostic Error Counters	
User Programmable Attempts	Up to 255

One of the transceivers we have been considering is **AeroComm PKLR2400**. This transceiver provides parameters that are very useful in our project such as low power consumption, small physical dimension, light weight, high RF transmission rate and long reception range. The above (Table 3.) is the specifications for PKLR2400 taken from the manufacturer's datasheet.

Despite the nice properties of this system, the major problem the AeroComm transceiver has is the fact that it can only operate in half-duplex mode, which it means the transceiver cannot transmit and receive simultaneously.

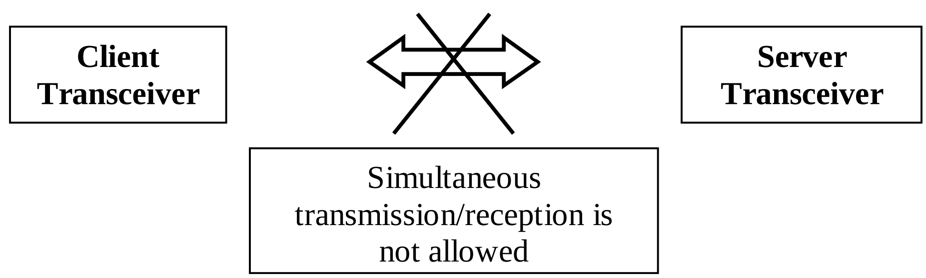


Figure 9. Half-duplex

There are three types of information traveling in the wireless link – sensor and video signals from the blimp; and control signals from the base computer. So far, we are only considering using one pair of transceivers. We would like the sensor information and video signals to be continuously sent back to the base, so that the live video and updated sensor information can be seen on the computer. The half-duplex mode could be troublesome because, on one hand, we would like to have smooth non-stop video and sensor signals coming from the blimp; and on the other hand we want to send control signals to the blimp. What we really need is a two-way communication system that allows simultaneous signal transmitting and receiving. It does not seem to be a good idea to stop the incoming video and sensor signals every time we want to send a control signal to the blimp.

Two possible solutions have been proposed in order to address the half-duplex limitation. One obvious solution is to use two pairs of transceivers, each of which provides one point-to-point channel – one for video and sensor signals; the other for control signals. Technically speaking, this solution seems to work flawlessly. However, such two-way communication has been established at the cost of using an extra pair of transceivers. In other words, the cost of building a prototype is increased. Given the fact of limited resources, the extra cost directly affects the success of our final prototype realization. Also, the extra channel reserved for the control signals will work less efficiently as the one for video and sensors due to less demand for the channel occupation. The control signals only require a mere fraction of the channel's bandwidth.

Another solution we have considered is the use network protocols. A protocol is a set of rules that govern the information traffic in the network. Its goal is to optimize the use of a communication channel shared by many users. In our case, we believe that 2-way communication can be achieved by applying a network protocol so that only one communication channel will be needed. Towards that aspect, the AeroComm transceiver provides some unique features that make the idea of using a network protocol possible. These features include:

- Different modes of serial interface.
- The use of a command set.
- Fast switching time.

Before discussing the protocol, a brief discussion about the above features may be helpful. The PKLR2400 provides four serial interface modes that include three transparent modes and one API mode. Three conditions are supported by a transparent mode: end character, fixed packet length with timeout and fix packet length without timeout. This means that data transfer through the RF interface will occur if those parameters are defined. As for API mode, the PKLR2400 uses AeroComm's command set to control the data transfer. In short, the API mode allows the user to control data transfer by using radio control commands and the three transparent modes allow the user to send data without the use of commands. These different modes of interface offer the user maximum system flexibility with programmability within each mode.

The following is the summary for the radio command set taken from the manufacturer's datasheet:

Table 4. AeroComm Radio Command Set

Name	Command	Length	Data	Checksum
RF enable*	80H	0H	0H	80H
Send Data*	81H	1 to 7f0h (includes 802.3 Header)	As required	As Required
Send Data Complete*	82H	1H	0 or 1 (see 5.1.4)	As Required
Received Data*	83H	1 to 7f0h (includes 802.3 Header)	As Required	As Required
In range*	84H	06H	IEEE address of Server Radio	As Required
Out of Range*	85H	0H	0H	85H

Switching time is defined by how long it takes for the transceiver to switch from transmitting mode to receiving mode or vice versa. It is defined as a minimum delay of 40 µs between (1) and (2) shown in Figure 10.

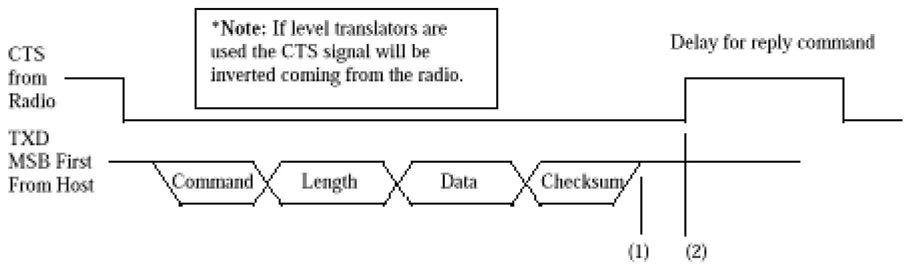


Figure 10. Mode Switching Timing

In a protocol, the packet is the basic unit that the protocol works on and also contains the information that actually travels in the network. The generic packet format is usually defined as follows:

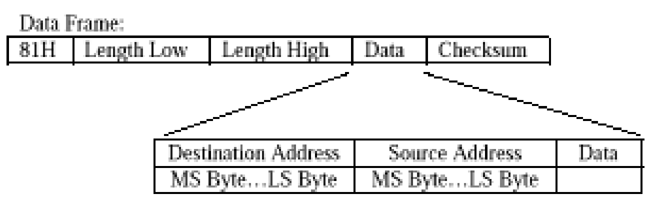


Figure 11. Generic Packet Format

In general, only one packet format needs to be defined for both the transmitting and receiving ends of the network for simple protocol implementation. Our case is a little different however, because of different data traffic demands and information which have different priorities. For instance, video and sensor signals have more traffic demand but less priority; control signals have less traffic demand but more priority. To deal with these concerns, we propose two protocols and two packet formats – one packet format for each protocol. They are defined as the following:

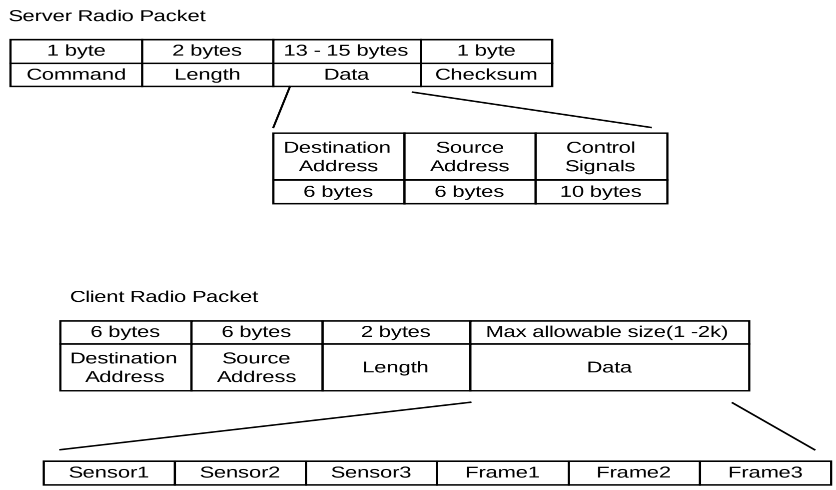


Figure 12. Frame Layout

The goal of using two protocols and two packet formats is to maximize the efficiency of channel occupation. The timeout parameter that is key for the base radio to get control of the RF link is defined as the time the RF interface waits before it sends the next packet. Coupled with fast switching time, the protocols we propose fool the user in such a way that live and smooth reception will be achieved despite the fact that there is actually an imperceptible time delay in between the transmission and reception. On the other hand, if our alternative design is used, which incorporates an analog camera transmitting a video signal over its own separate transceiver, then only one packet format is needed. In either way, our goal is to make use of the channel as efficient as possible while assuring correct reception of the information.

Based on the above criteria, using a network protocol is preferable as far as cost and possibility of implementation are concerned. In addition, instead of writing our own protocol, there are already many well-known protocols that are ready for us to use.

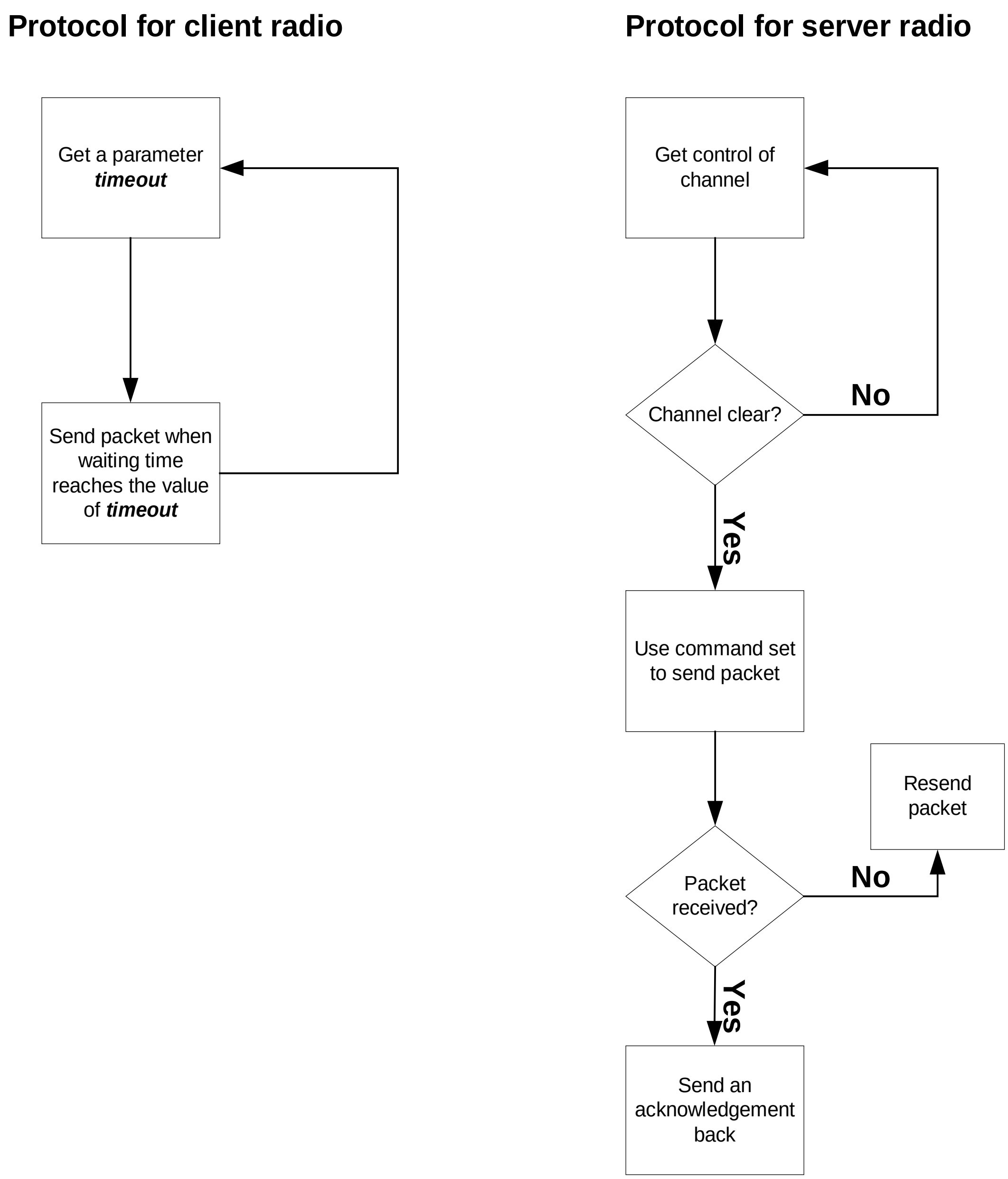


Figure 13. Protocol Flowchart

The microcontroller that resides in the transceiver also may have its own protocol. The AeroComm transceiver itself transfers packets using the IEEE 802.3 protocol. Overall, we prefer the AeroComm transceiver because of its high performance and flexible configuration.

Video

We have considered several types of cameras to use for our project, some are analog and some are digital. For the time being, we have decided to use a digital camera. Although a digital image is not inherently better than an analog image, there are many reasons to choose a digital camera. We chose a digital camera because digital images are capable of being compressed to fit our transceivers' bandwidth, and because a digital camera is more consistent when transmitted over long distances.

First, for the same amount of bandwidth we can transmit a lot more information with a digital signal than we could with analog because analog signals cannot be compressed. For example, to transmit an image in analog, every pixel is included in the signal. A standard NTSC screen includes 378,000 pixels per frame, which fits into the 6MHz of bandwidth of a TV channel. On the other hand, a standard ATSC (Advanced Television Systems Committee) screen can have up to 5 times as much per frame which has to squeeze into the same bandwidth of 6Mhz. This is done using a compression algorithm called MPEG-2, which takes advantage of how the eye perceives color variations and motion. Inside each frame, an MPEG-2 encoder records just enough detail to make it look like nothing is missing.

Second, we can also transmit data that stays more consistent over distance. Both analog and digital signals get weaker with distance, but while the picture on an analog TV slowly gets worse with more distance, a picture on digital TV will stay near perfect until the signal becomes too weak for the receiver to pick up.

There are also disadvantages to using digital, however. A drawback to compression schemes such as MPEG-2 is that at higher compression rates, it gives poorer pictures. Still, the quality of the picture is a lot better than a regular NTSC image and using this scheme, MPEG-2 in some instances can reduce the amount of bits by 55 to 1! Another problem is that we need a separate DSP chip to compress video before transmitting, which means additional power consumption and extra weight, both of which are great concerns on this project.

Once the decision is reached to use a digital camera, the question of what kind of camera would be ideal for our project arises. Historically, there have been two major classes of image capture devises. They are CCDs (charge coupled devices) and CMOS (complementary metal oxide semiconductor) sensors.

Even though currently CCD is the most used we chose to use a CMOS video camera for several reasons. Below is a table of how these technologies compare.

Table 5. CCD vs. CMOS

Sensor size	1/2", 1/3", etc	1/2", 1/3", etc
Integration	Amplifier present at each photosite, A/D converter and additional processing easily added	Requires separate components for amplification, processing, storage, etc.
Power Consumption	20-50 Milliwatts	2-5 Watts
Manufacturing	Can be produced by any standard CMOS fab in great quantities	Requires a dedicated facility
Resolution	Up to one megapixel	Up to six megapixels
Signal Quality	Not as good as CCD	Historically the best
Output	No signal degradation	Data subject to fading
Speed	Up to thousands of frames per second	Usually up to 100 frames per second
Cost	Higher yields and less susceptibility to defects makes it a low cost technology	Expensive to produce

Our decision to use a digital CMOS camera over a CCD was based on the above criteria. Below is a description of how these factors influenced our decision.

- **Sensor size**: They are basically the same.
- **Integration**: With CCD imagers, only the image capture and formation functions can be included on the CCD image-sensing device. Therefore, supporting semiconductors are required to operate the device, condition the image signal, perform post-processing, and generate standard video output. In contrast, with CMOS, Digital Signal Processing (DSP) functions can be integrated directly on the

chip. Our need for a compact camera system that can easily integrated with the rest of the system makes CMOS imagers very convenient.

- **Power Consumption**: Power consumption directly affects battery life and, even more importantly for this project, battery size and weight. CMOS sensors consume less power, usually up to 10 times less, than CCDs. CCD systems are generally very power hungry. They are capacitive devices that need external control signals and large clock swings (5 to 15V) to achieve acceptable charge transfer efficiencies. The off chip circuitry used with CCDs also dissipates significant power, and CCD systems require numerous power supplies, clock drivers and voltage regulators. CMOS imagers, on the other hand, require only a single standard clock input and a single 3.3V or 5V power supply, reducing both power consumption and system cost.
- **Manufacturing**: CMOS is used to manufacture approximately 90 percent of all semiconductors today, therefore, manufacturing imagers using this process is less costly and does not need a separate facility, whereas CCDs are made using specialized fabrication processes, making them more expensive and therefore less desirable for our project.
- **Resolution**: up to six megapixels for CCDs compared to up to one for CMOS. This is a minor consideration because our pixel resolution is already limited by our transceivers' bandwidth.
- **Signal quality**: Although CCDs are historically the best, superior signal quality is not a great enough concern to affect our decision.
- **Output**: CMOS data is accessed like RAM, therefore, there is no signal degradation. In a CCD, data is accessed by shifting rows and reading serially, so output fades.

- **Speed**: CMOS sensors are very fast, some as fast as thousands of frames per second.
- **Cost**: Because CMOS is so versatile, it's more cost effective.

Finally, our chosen camera is a low cost CMOS digital camera from Spectronix called RoboCam RC-2-C.

Figure 14. Taken from www.spectronix.net

This CMOS camera is ideal for our project because it is low cost ($159), has a built in A/D (analog to digital) converter with digital outputs, low power requirement and small size.

User Interface

The Aero Group's blimp system will consist of the blimp and a base station. In order to operate the blimp, the user will set up the blimp, plug the base station into the computer's USB port, run the software, and start flying. The user will control the blimp through a joystick or gamepad connected to the computer. The computer will receive sensor and position data from the base station and display it to the user. The goal of the user interface is to present enough information so that the user can fly the blimp remotely from the computer without having to actually keep an eye on the blimp itself.

Setup: Using USB to connect base station to user's computer.

People don't normally associate setting up a system with the user interface. However, set up is still an important user experience. A system that is easy to set up and use is more likely to get used, and subsequently recommended to other people. For this reason, the system will interface to the user's computer using a USB (universal serial bus) interface.

The base station of the blimp system will connect to the user's computer through the USB port. USB was chosen as the interface between the base station and the computer because it simplifies operation and has the required bandwidth for the data. USB devices are plug and play and hot swappable. This means that the user will be able to plug in (or remove) the base station from the computer without having to shut the computer off or restart the operating system. With USB, the user will be able to plug the base station into the computer, run the user interface program, and start flying.

USB also provides the necessary bandwidth for data and video transmission. The Aerocomm transceivers have a bandwidth of approximately 800 kbps. As stated earlier, much of this bandwidth will be consumed by the video transmission. USB has a maximum bandwidth of 12 Mbps; more than enough to handle the maximum bandwidth of the Aerocomm unit. In comparison, a standard serial port has a maximum data speed of approximately 115 kbps, or about 8 times slower than the Aerocomm transceiver.

There are several other possibilities for transferring data from the base station to the computer. One is building a dedicated PCI card. However, a PCI card would require the user to open the computer, install it, find a free IRQ, etc…. In addition, a PCI card is not compatible with notebook computers. USB offers a far superior user experience than a dedicated PCI card. Instead of having to turn off the computer, open the computer, and install a PCI card, our users will be able to merely plug our base station into their computer. In addition, a USB base station will work with any notebook computer with a USB port.

A second possibility is using a parallel port to interface the computer with the base station. While older parallel ports had a maximum speed of approximately 115 kbps (too slow for our application), most newer ECC/ECP parallel ports have a maximum speed of greater than 1 Mbyte/s. (See IEEE 1284). However, most users already have a printer or some other device already connected to their computer's parallel port. Therefore, in order to use the blimp system, the user might have to disconnect some other device in order to use blimp. The user would also be precluded from attaching another device to the parallel port when using the blimp. In comparison, USB allows up to 128 devices to be connected to the USB port. The user is far more likely to have an available USB position in the USB chain than having a free parallel port.

A third possibility is an infrared IRDA interface. IRDA has the necessary speed and supports a connection as fast as 4 Mbps. IRDA also provides a superior user experience; all the user has to do is place the base station in range of a computer's IRDA port and run the software to use the blimp. However, most desktop computers do not have an IRDA port. USB is a far more common interface than IRDA.

The final possibility is connecting the base station to the computer over an ethernet network. Ethernet supports speeds of 100 Mbps and higher. However, to put an ethernet connection in the base station would essentially require placing a computer and an operating system within the base station itself. In addition, this would require that the base station's network protocols and settings be configurable by the user. This adds a great deal of complexity both for the user and for the designer. Therefore, an ethernet connection will not be used.

Control: Controlling the blimp with a gamepad or keyboard.

Set-up is only one aspect of the user experience. The user must also be able to control the blimp. The control for the blimp will be provided via a keyboard, game controller, or autopilot.

Manual control of the blimp will be performed either through the use of the keyboard or game controller. Users will be able to choose their preferred method of control via a preference setting in the user interface software. Either set of controls will provide eight possible "moving" directions for the blimp: left, right, up, down, forward, backwards, rotate left, and rotate right. Actual layout of the controls has not yet been determined. However, one constraint is that any game controller used must have at least 8 buttons in order to provide full control over the blimp.

The main idea is to be able to use whatever input method the user's computer already has available. Virtually every computer has a keyboard, and most Intel based personal computers have a common interface for a game controller. Using the input devices available on the user's computer eliminates the need to design any special input hardware to control the blimp, and leave the problem of control strictly as a matter of software.

A long-term goal is to add an autopilot function to the user interface software. The first prototype of the blimp system will likely not have this function available. However, eventually this will be added so that a user can tell the blimp to travel to a certain location, or fly in a certain pattern, and the computer will automatically direct the blimp accordingly.

Regardless of the input method, the moving direction commands will be transmitted from the computer to the base station over the USB interface, and then from the base station to the blimp over the Aerocomm transceivers.

Feedback: Getting information to the user through the display.

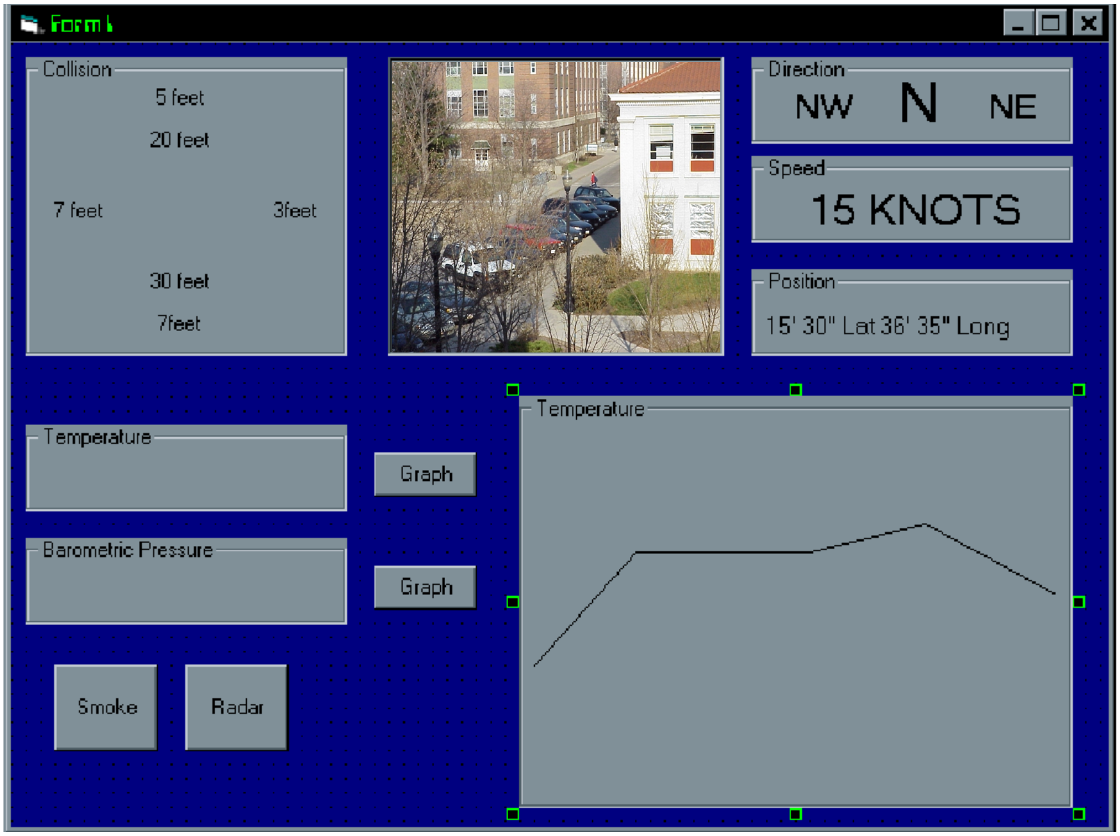


Figure 15. User Interface

The main part of the user interface is the display. Figure 15. shows a rough draft of what the user interface might look like. The goal is to provide enough information to be able to fly the blimp remotely and provide the sensor data to the user on one screen.

The display is divided into two sections: control and sensor information. The lower half of the display contains the data from the blimp's sensors. As was stated earlier, the blimp will initially have sensors for temperature, pressure, smoke detector and radar detector. The display will be able to show both current and historical data. Current data will be constantly displayed, and historical data will be shown on the graph. In addition, the graph could be replaced with a map to graphically show the current location of the blimp.

The top half of the display contains all the information necessary to remotely control the blimp. This includes information from the GPS receiver including direction, speed, and position. The data from the blimp's proximity sensors is displayed to help avoid collisions. Finally, video is displayed to show what is going on around the blimp.

Developing the software with Visual Basic.

Providing real-time video to the user is the most difficult part of the user interface. In addition to the bandwidth problems discussed earlier, displaying video on a custom computer interface is not a trivial matter to program in software. Fortunately, there are several operating systems that support video and provide a multimedia or video API that greatly simplifies this problem. Therefore, video support has become the main constraint in choosing a development environment.

There are at least three operating systems which run on standard Intel X86 hardware which provide video APIs. These are Windows, Linux, and BeOS. Windows provides a video API through either DirectX or VFW (video for windows). Both of these API's are accessible through Visual Basic and Visual C++. Linux supports video through the Video4Linux project. This API is accessible through a C/C++ interface. Finally, BeOS has video support built in the operating system in the form of its Media class. BeOS is an object-oriented operating system, and this class is accessible through a C++ interface. Sample code is available for all the different video API's, either through the organizations that developed them or through third parties.

A second constraint is USB support. An operating system must be able to support USB. Currently, Windows has the best USB support. USB support under Linux or BeOS is still under development. Therefore, Windows will be used for our first implementation of the user interface software.

The DirectX and VFW API's are accessible both in Visual Basic and Visual C++. The choice of language amounts to a trade off between portability and speed of development. C++ code would be easier to port to another operating system. Even if the initial interface is written for windows, user demand may justify porting the interface to another operating system in the future. While our code for the "visual" part of the interface will contain OS specific API calls, the data collection and processing routines probably could be written in ANSI C++. Writing the user interface in C++ could ease any porting effort, as these ANSI based routines could be recompiled on any operating system that supports ANSI C++.

In comparison, Visual Basic is only available on the Windows platform. Any non-operating specific routines could not be ported to any other operating system because no other operating system supports Visual Basic. However, Visual Basic is far easier to program than C++. In addition, Visual Basic is a higher level language with many built in function. These advantages translate directly into a faster development cycle. Time is a critical resource for this project, and outweighs the future concern over portability. Therefore, the initial user interface will be written in Visual Basic.

Proposed user interface proof of concept.

Currently the Aero Group is in possession of a wireless black and white video camera and a TV-Video capture card. Windows, Linux, and BeOS video API's support the video capture card. The Group will attempt to construct a sample user interface using these components in order to evaluate the practicality of using Visual Basic to program the user interface.

Because the video camera is wireless, this opens up the possibility of attaching it to a toy remote controlled car. By interfacing the car with the computer, the group will be able to test the practicality of the control and video systems. The goal for this proof of concept is to be able to control the car with a game-controller attached to the computer, while displaying the video from the video camera on the screen.

The advantage of such a proof of concept is that the group will be able to gain experience in programming the user interface and be able to produce a working demonstration, all for the cost of a cheap remote controlled car.

Conclusion

The Aero Group's blimp system will consist of the blimp, base station, and software. All a user will have to do is (1) provide a computer; (2) plug the base station in that computer's USB port; (3) run the software; and (4) start flying. The user will control the blimp either through the keyboard, game-controller, or the software's autopilot function. Navigation data will be provided on the computer's display. This will include live video and speed, direction, and proximity data. The sensor data will also be provided on the lower half of the display in an easy to read format. The final system will provide all the information necessary to remotely pilot the blimp, the necessary sensor data, and a user-friendly experience.

The next challenge is cost. We want to keep the bare bones model, sufficient to fly, transmit it's current position, send video and a data from a couple of sensors, under $2000. The client can specify sensors to configure their UAV, which will raise the price accordingly. In this manner our product will be accessible by small and large business as well as individuals

The final challenge we must confront is autonomous operation. The big challenge is that true autonomous operation will require a lot of information processing which translates out to hardware and that requires power and weight. This implementation will have guided operation but the principles we use in this UAV will be the basis for the final implementation, which will use VLSI design to integrate multiple chips into one or two lower power chips.

In addition to the challenges mentioned, another issue we will be concerned with is efficient use of time. Efficient time management and the ability to follow a set schedule are very crucial in any project because they are indications of your commitment and credibility.

Keeping this in mind, we have devised a realistic and practical schedule to which the whole group can adhere. We have twenty weeks in which to implement and demonstrate our autonomous flying vehicle. These twenty weeks have been divided into four blocks of time, five weeks each.

In the first phase of our project we were mainly interested in research, collection of data, allocation of tasks, and promotion of outside interest in our project. To that end, at the end of phase one we compiled the information we had collected and incorporated it into a short presentation.

In phase two, we started our subsystem design. We also start actively soliciting further support for the project. Here at the end of phase two, we now have a list of materials and resources needed for our project. We only await financial support before we can begin our procurement and construction steps.

Phase three will be dedicated to subsystem testing, design recursion, and design integration.

In the final five weeks of our project, we will conduct exhaustive tests of our model in different environments and compose our final project proposal and demonstrate our completed design.

Our current planned implementation of our sensor platform will use a computer controlled, electrically powered, model car or rover. Our initial hopes were that we would have the available funding to take our design to some airborne platform. However, our primary goal is not the flying vehicle itself. Our main concern is the electronics needed to make the concept of a flying autonomous sensor platform a reality. Our goal is to miniaturize these systems and make them efficient enough to be used in the final vehicle. Sensors, control systems, power supplies, communications, and user interface are the focus of our project. If we can make these systems small enough and keep power requirements to a minimum, it should then be a rather straight forward exercise to later move our design to a micro-blimp, when the budget becomes available.

Our solution will be capable of supporting many different kinds of sensors. So, in addition to being used in different vehicles, the systems we develop will be customizable for different applications. We will be able to use the same base platform, consisting of the GPS, control systems, power, communications and user interface, and customize that platform with whatever sensors the customer requires. If the customer needs chemical sensors, we will be able to provide that. If the customer needs video and audio, we can provide that. Infrared, temperature, ultraviolet, and radar sensors; whatever the customer requires we will be able to provide; all over a remote link on a mobile sensing platform.

Customizability permits our platform to used in many different applications. Everything from entertainment and sporting events to military, security or search and rescue. We will be developing a flexible product with multiple uses.

Our current development cycle is scheduled for 20 weeks. At the end of the 20 weeks, we will have a working prototype. Our prototype will develop directly into the final product.

Once again, our focus is on the electronics. The electronics we develop for the rover, control, sensors, communications and user interface, will transfer directly to the final goal of a flying autonomous sensor platform in a flying wing.

Resources

DraganFly Innovations Inc.	www.draganfly.com
National Semiconductor	www.national.com
Motorola	mot-sps.com
SiGEM	www.sigem.ca
Acroname Inc.	www.acroname.com
Unitrode/Benchmarq	www.benchmarq.com
Emcore	www.emcore.com
Microchip Technology Inc	www.microchip.com
Aerocomm	www.aerocomm.com
Spectronix	www.spectronix.com

Index

A

Acroname *45*
ACRONAME *11, 12*
AeroComm *21, 23, 24, 25, 26, 29*
ANSI C++ *40*
API *25, 39, 40, 41*
API mode *25*
Appendix *46*
Applications *14*
ATSC *30*
autopilot *37, 42*

B

base station *34, 35, 36, 38, 42*
Benchmarq *17, 45*
BeOS *40, 41*
bq2057 *17*

C

CCD *31, 32, 33*
CISC *20*
CMOS *31, 32, 33*
Collision Avoidance *10*
Command Set *26*
Communication *23*
Conclusion *42*
Control *37*
Corporate *15*

D

Detector Circuits *7*
DigitalDNA *6*
DirectX *40*
display *34, 38, 39, 42*
DraganFly Innovations Inc *3, 45*

E

ECC/ECP *35*
Emcore *18, 46*
Entertainment *15*
Ethernet *36*

G

GP2D05 *11, 13*
GPS *4, 9, 10, 39, 44*
Graf *7*

H

half-duplex *24*
Handy Board *19, 20*

I

IEEE 1284 *35*
IEEE 802.3 *29*
Individual *15*
infrared *11, 13, 14, 36*
Introduction *2*
IR *8, 11, 14*
IRDA *36*

L

Law enforcement *15*
lighter-than-air vehicle *2*
Li-ion *16, 17, 18*
Linear Regulator *17*
Linux *40, 41*
LM34 *6*

M

MC 154010 *8*
Microchip Technology Inc *19, 46*
Microcontroller Selection *19*
Military *15*
MIT *19*
Motor Control *22*
Motorola *6, 8, 9, 20, 45*
MPEG-2 *30*
MPX2202 *6*

N

NASA *18*
National Semiconductor *6, 45*
Navigation *9, 42*
network protocols *25, 36*
Ni-Cd *16*
Ni-MH *16*
NTSC *30*

O

object-oriented *40*

P

packet *25, 27, 28*
Payload *5*
PCI *35*
Pegasus *18*
PIC16F874 *19, 20, 21, 22*
PICmicro *19, 20, 21, 22*
PKLR2400 *23, 25*
POLAROID 6500 SERIES *12, 13*
Power *16, 31, 32*
pressure sensor *5, 6*
Project Overview *2*

protocol *25, 27, 29*
proximity sensors *4, 10, 11, 39*
Pulse Width Modulation *22*

Q

Qualcomm *17*

R

radar detector *5, 7, 39*
Resources *45*
RISC *20*
RoboCam *33*
rover *44, 45*

S

serial port *10, 35*
SGM5608PM *10*
SHARP *11, 13*
SiGEM *10, 45*
smoke detector *5, 8, 39*
Solar Cell *18*
Sonar *11, 12*
Sony *17*
Spectronix *33, 46*

T

Texas Instrument *17*

U

U9VL9V *17*
UAV *2, 14, 15, 17, 20, 21, 22, 42, 43*
UCC387 *17*
UCC3956 *18*
Ultralife *17*
ultrasonic *11, 12*
Unitrode *17, 18, 45*
unmanned autonomous vehicles *2*
USB *34, 35, 36, 38, 40, 42*
User Interface *34*

V

VFW *40*
Video *29*
Visual Basic *39, 40, 41*
Visual C++ *40*
VLSI *5, 43*

W

Windows *40, 41*