1-1-2003

Mechatronics design for automated antenna deployment

Lars Rune Kaldestad
Iowa State University

Mechatronics design for automated antenna deployment

by

Lars Rune Kaldestad

A thesis submitted to the graduate faculty
in partial fulfillment of the requirement for the degree
MASTER OF SCIENCE

Major: Mechanical Engineering

Program of Study Committee:
Greg Luecke, Major Professor
Donald Flugrad
Thomas Rudolphi

Iowa State University
Ames, Iowa
2003

Graduate College
Iowa State University

This is to certify that the master's thesis of
Lars Rune Kaldestad
has met the thesis requirements of Iowa State University

Signatures have been redacted for privacy

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The purpose of this research project has been to develop a mechatronics system for automatic antenna deployment of an existing Winegard Sensar® Recreational Vehicle (RV) TV antenna. Winegard had a set of criteria for how they wanted the antenna system to look and function. The main criterion was that it had to be a retrofit to the existing manual antenna. From these requirements a prototype was developed and built to test the functionality of the system as a whole. Feedback from Winegard resulted in an improved new prototype, but it still did not meet the requirements. Therefore, a third prototype had to be made that met the expectations of the team.

CHAPTER 1 – Introduction

1.1 Mechatronics Systems

By definition a mechatronics system is the "weld" between a mechanical, electrical and a computer system. A good mechatronics system incorporates all these disciplines and therefore challenges people from an interdisciplinary angle. A mechatronics system consists of three parts actuators, sensors, and software or computer/micro controller. The actuators and sensor may consist of mechanical, or electrical or both mechanical and electrical parts together. The micro controller and software provide the operational foundation for the system.

1.1.1 Actuators

Actuators are needed in a mechatronics system to execute the commands from the micro controller. There are several actuators commonly used in a mechatronics system. Examples of pure mechanical actuators are the combustion engine, hydraulic motors, and hydraulic cylinders, while examples of electrical actuators are electrical motors, solenoids, and even loudspeakers. Electrical actuators take advantage of electromagnetic forces to perform the actuation.

1.1.2 Sensors

In a mechatronics system, sensors are used to identify almost anything of physical matter. Examples of sensing matter are temperature, humidity, smoke, radioactivity, infrared light, speed, and magnetism. The sensors are used to identify conditions when the micro controller should respond according to programmed instructions and change the systems behavior. An example is the tachometer in a cruise control system, which indicates the vehicle speed. The micro controller reads the sensor and shuts off or adjusts the actuators to maintain the set speed.

1.1.3 Software and Computer/Micro Controller

Mechatronics systems have a micro controller that controls the operation of the system. It must be programmed to execute the tasks to be performed for a particular application. Micro controllers have designated input and output ports. Input ports can be digital or analog. Digital inputs are signals from sensors, such as photo sensors or hall-effect sensors. Analog input ports can be used to read sensors like the potentiometer or the tachometer. The micro controller will output programmed signals on the output ports that, through switches and relays, can control electrical actuators or indicator lamps/LEDs.

1.2 Advantages and Disadvantages of a Mechatronics System

Mechatronics systems have proven to be greatly beneficial over traditional mechanical systems. Some of the benefits include cost savings, cost efficiency, added durability, reliability and accuracy to a traditional system, and added convenience to every day living. In addition, they can be used in places that pose a great risk to human beings. However, for every advantage there is a disadvantage. These disadvantages may include high initial investment, the need for highly paid and highly skilled workers, and system reliability.

Mechatronics systems are replacing workers in many modern workplaces. Assembly line work is repetitive and causes injuries in workers. These injuries alone cost American manufacturers $3.2 billion each year [http://www.ergonext.com/aa-articles/lib-mutual.html]. Introducing a mechatronics system into the assembly line to take over the work will in most cases save manufacturers at least the equivalent amount of money each year.

One of the first places where a mechatronics system was utilized was in the car industry. Robots were introduced for automated assembly in the 1970's and contributed to cost savings for car manufacturers. In 1985, more than 50 percent of all robots that were sold were sold to the auto industry [8]. These robots were used for welding jobs that were costly when performed by human workers.

Another drawback with using human workers is fatigue. Fatigue causes people to perform at a slower pace and with less accuracy than usual. The National Commission on Sleep Disorders Research estimated in 1993 that over 70 million U.S. workers were suffering from fatigue. When accidents, medical costs, and absenteeism were accounted for the estimated cost ranged from $50 billion to over a $100 billion per year [http://www.trconsultinggroup.com/safety/jan2002.html]. Introducing and maintaining machines that can perform at a consistent pace and repeat the same motion with the same amount of accuracy each time, can save companies millions of dollars.

Moreover, mechatronics systems add convenience to working in hazardous environments. In places where humans have to protect themselves against the environment, mechatronics systems or robots can operate without protection. This is especially true in areas where humans can't breathe or where the dangers are too high for humans to risk working there. Examples include Mars vehicles, bomb robots, and control systems in chemical plants and nuclear power plants.

The mechatronics control systems also add convenience to every day life. A newly developed dish washing machine that calculates how big a load it has and regulates how much water it uses, saves both power and water [Kenmore]. Coffee machines that can be programmed to have the coffee ready when you get up in the morning can give people another 15 minutes of sleep.

Unfortunately, there are not just advantages to mechatronics systems. Machines may not always perform the same every time. They need to be programmed, maintained, and adjusted to perform at the optimum level. In addition there will be some wear and tear that requires attention on a regular schedule, and the initial investment that manufacturers have to put into buying a new system is often times more than what companies are willing to invest. In addition, programming such systems needs highly skilled workers that will have to be paid more than regular labor, adding to the cost of the system as a whole.

There are many reasons to introduce mechatronics systems and each company must decide if it is necessary to introduce such a system. They must way the cost savings, the added efficiency, and convenience against the drawbacks of high initial investment, and added components that will wear, before deciding which is better.

The same considerations are to a certain extent true when buying a “home” product that has been automated. Usually the automated washing machine and the automated coffee maker come at a higher price than the semi automatic ones. They are also more complicated and higher repair costs can be anticipated if and when they break.

1.3 (Sensar®) RV Antenna

The purpose of this thesis and research project is to develop a mechatronics system for Winegard. The system will deploy and adjust an existing recreational vehicle television antenna system for TV reception. This particular antenna, Sensar® has already been produced and sold in great quantities.

The antenna is mounted on the roof of RVs to find the best reception. It can be folded down onto the roof or be deployed for TV reception. The antenna is T-shaped and in order to get optimal reception the antenna needs to be fully deployed and properly oriented. A typical antenna installation on an RV is shown in Figure 1-1a, and a close up of the antenna is shown in Figure 1-1b.

Figure 1-1a. Outside configuration of antenna.

Figure 1-1b. Sensar antenna deployed.

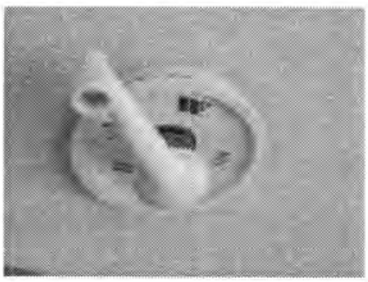


Figure 1-2. Inside configuration

Figure 1-3. Antenna mounting bracket, crank and direction handle

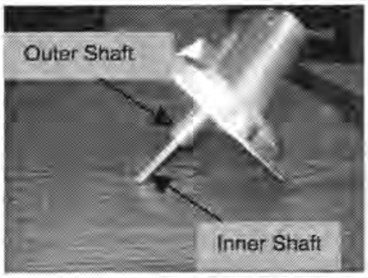


Figure 1-4. Concentric shafts, mounting bracket, and base

Figure 1-3 shows the mounting bracket, crank, and direction handle before installation, but put together as if they were installed on the roof. A hole is put through the RV roof for the two shafts. The mounting bracket is attached to the RV roof with screws.

Deployment of the antenna can be done in less than ten seconds if cranking the antenna by hand. To fully deploy the antenna, the crankshaft needs to be rotated fourteen times from the parked position. On the already mounted antennas the crank is placed under the ceiling, but the exact location under the ceiling varies from one RV to the other.

Location of the antenna may lead to problems reaching the crank. This is especially true in fifth wheel RVs, where the distance from the floor to the roof is greater than in a motor home. The majority of RV owners are older people, and for them reaching high up under the ceiling can be difficult and even dangerous. Even though the crank is mounted in an easily accessible area it is not always mounted near the TV. This makes it difficult to tune the TV and watch it at the same time. The deployment and adjustment of the antenna then becomes a two-person job.

One way to eliminate the need for manually deploying and tuning the antenna is to incorporate a mechatronics design. The operations can then be performed by the push of a button. User feedback also indicates a desire for implementing an automated system.

1.4 Focus of Thesis

The focus of the thesis is the development of three mechatronics prototypes. These were designed and built in order to test if the goals set down by Winegard could be achieved. The thesis will elaborate on some of the problems that were encountered and how these were resolved. It will explain the design process that was used and changes that were made in order to meet the requirements. Different sensors and actuators that were used and why they were preferred before others will also be covered in some detail.

1.5 Preview of Thesis

This thesis is composed of five chapters. Chapter one introduces mechatronics systems in general and explains how they work. A brief background of the project and the focus of this thesis is also explained. Chapter two is a review of mechatronics systems and components for recreational vehicles. The emphasis is on other automated systems available on RVs. Chapter three is the design approach section where the design considerations and approach is explained. In chapter four the development of the prototypes are covered and chapter five provides the conclusion.

CHAPTER 2 – Mechatronics Systems and Components for Recreational Vehicles

2.1 Motion Controlled Systems on RVs

RVs have evolved from being simply vans with beds to becoming modern luxury homes on wheels. The space limitations in an RV call for solutions that will efficiently use the room available. A typical example is a bed that can be transformed into a table and seating. The reorganizing of inventory often involves manual labor, which can be hard for older people. For convenience these inventory changes are often automated. Configuration changes can be made by the push of a button. Examples of such changes made in modern luxury RVs are expandable rooms, automated awnings, automated door openers, and luggage and wheel chair lifts [15].

2.1.1 Expandable Rooms

There are standards regulating the length and width of a vehicle traveling on highways and interstates in the U.S. In an RV this means that all living components must be placed inside a limited area, often resulting in crowded and inconvenient living spaces. One way that the RV industry has dealt with this problem is to use expandable rooms. These rooms are designed to be expanded when the vehicle is parked and retracted when the vehicle is traveling. Usually the bedroom and the kitchen/living room are the rooms that can be enlarged. On the high end of the RV line, these rooms are automated as opposed to cranking them out manually. They are actuated by either a hydraulic cylinder or an electrical motor that is used with a rack and pinion configuration as shown in Figure 2-1b below. A limit switch is used that signals when the room is fully expanded and the actuation can be stopped.

Figure 2-1a. Expandable room

Figure 2-1b. Expandable room actuation, rack and pinion

2.1.2 Automated Awnings

Most RVs have awnings to give protection against sun and light rain. Traditionally these awnings were also cranked out manually, but an increasing number of RVs are equipped with awnings that can be deployed automatically. An electric motor, wires, and gears actuate these systems.

2.1.3 Automated Door Openers

Many RVs have a door opener that works by pneumatic actuation. In addition, they can also have an automatic latter that slides out when the door is open, as shown in Figure 2-2 below. This system has been used on public busses for years, and has more recently been transferred over to the RV industry.

Figure 2-2. Retractable automatic stairs

2.1.4 Automatic Leveling

When an RV is parked on a camping site or "boondocked" on the side of the road there are systems inside the RV that require the RV to be level to operate correctly. One example is the refrigerator. The refrigerator of an RV operates by burning propane and uses ammonia for coolant instead of compressing refrigerant like the home refrigerator. The traditional way of leveling an RV is to use a mechanical crank and level. The high-end luxury motor homes have systems that do this automatically by using hydraulic cylinders and electronic levels. This system can be seen in Figure 2-3a and b below.

Figure 2-3a. Leveling system on a motor home

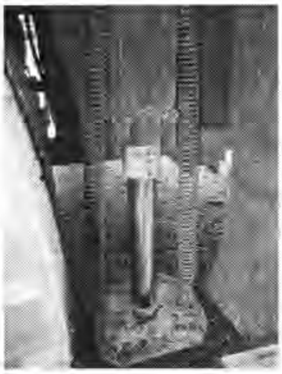


Figure 2-3b. Leveling system hydraulic cylinder

Figure 2-3a shows wheel lifted off the ground and Figure 2-3b shows a close up of the hydraulic cylinder of the leveling system.

2.2 Other Automated Antenna Systems

There are satellite dishes that are either semi automatic or fully automated. The semi automated satellite dishes can be deployed when an RV is parked. The satellite's position is preprogrammed in the dish's control system and by using a compass and GPS it can locate the satellite. An accurate pointing line can be obtained by simple calculations. The fully automated tracking antennas are more complicated because they must account for vehicle movement and make calculations at increments, as the vehicle is moving. These types of antennas also use GPS and compasses, but they also need rate gyros to compensate for the vehicles bouncing, acceleration, and turning. These antennas are available from Winegard Company, SeaTel, KVH, and Datron to mention a few.

There are also automatic RV systems for broadcast TV available. They deploy and search for signals by using a remote control. However, since the antennas for the system discussed in chapter 3, 4, and 5 is intended for has been sold in great quantities, it will be a market for a retrofit that can be mounted onto already existing antennas so people do not have to buy a whole new antenna for automatic deployment.

2.3 Actuation

Every mechatronics system needs actuators that can execute the tasks programmed into the microprocessor. Some examples of actuators that can be used in a mechatronics system are hydraulic and pneumatic systems, ac motors, dc motors, stepper motors, and solenoids. The focus will be on the last three types of actuators, which were used in the three prototypes that were built.

2.3.1 Electric Motors

Electric motors are the most widely used actuators in mechatronics systems. These motors can be classified by either function or electrical configuration. Examples include gear motors, torque motors, servomotors, and stepping motors. The electrical configuration is either alternating current (AC) or direct current (DC). Electrical motors are popular because they are relatively cheap to produce and are compact in their design. Additionally, electric motors are reliable and have a high torque to cost ratio. Moreover, in contrast to a combustion engine, which is unable to run at speeds slower than the idling speed, the electric motor has a high starting torque and can start from zero velocity. This feature makes the electric motor ideal for robotic and mechatronics purposes where frequent on and off cycles are anticipated.

There are two main categories of electric motors, AC and DC. The AC motor is used for heavy machinery applications such as running a hydraulic pump to build pressure in a hydraulic system. Large AC motors require special starting procedures to make them run and are therefore quite common in systems where they are kept running continuously. DC motors can also be used for heavy machinery. They can range in power from a few milliwatts up to a megawatt. Their applications range from small watch motors to big motors that run hydraulic pumps in a hydraulic system. For small mechatronics purposes, the DC motor is most commonly used.

There are several types of motors that can be characterized as DC motors. By definition all motors that run on direct current qualify for this name. DC motors commonly used for mechatronics projects are servomotors, gear motors, and stepper motors. Although they have different names their working principals are somewhat the same in that they all depend on electro magnetic fields to operate. The motors that are being used for this project are the permanent magnet DC motors and the stepper motor.

2.3.2 Solenoids

Another kind of actuator that also uses electromechanical or electrostatic force is the solenoid. Solenoids, which are used extensively in mechatronics systems throughout the industry, include door latches, relays, and trigger actuators. An example of their application is in a hydraulic circuit where solenoids are used to open valves. The benefit of this actuation in hydraulic systems is that one does not need to use a mechanical handle. By simply pushing a button to open valves, actuation at a higher level by for example a hydraulic motor, a hydraulic pump, or a hydraulic cylinder can be achieved. Solenoids use the same principles of electro magnetism that motors take advantage of when they are rotating.

Solenoids consist of two main parts that make up the functional pair, the permanent magnet and a coil of wires. The permanent magnet is placed inside the coil, and when the coil is energized the magnetic pin will be “sucked” into the coil by the magnetic force. Depending upon the current put through the coil and the strength of the magnet, these solenoids can be very strong. However, they do consume a lot of energy and heat up very quickly. To prevent overheating solenoids should not be active for more than a short period of time. The on/off time in a solenoid is called the duty time and is important in choosing which solenoid best fits the design. When using solenoids in a mechatronics design, one must leave enough time in between each on-cycle to let the solenoid properly cool down. Otherwise, a solenoid will wear out quickly. The consumption of energy leads to the greatest disadvantage of solenoids.

2.4 Sensors

"A sensor is an element in a mechatronic or measurement system that acquires a physical parameter and changes it into a signal that can be processed by the system. Often the active element of the sensor is referred to as a transducer"[12]. Sensors are used to measure physical quantities such as position, distance, force, strain, temperature, vibration, and acceleration. A few examples include infrared photo-emitters and detector pairs, potentiometer, proximity sensors (like the hall effect sensor), strain gages, temperature gages, and accelerometers. There are sensors available to detect almost anything of physical matter, from radioactivity to humidity and pressure. The most important sensors for this mechatronics system are sensors that detect position. They are the photo emitters and detector pairs, potentiometers, and proximity sensors. In addition, a comparator can be used to sense analog voltage potentials and can also be used as a sensor to detect position.

2.4.1 Photo Emitter and Detector Pair

One of the simplest and most basic optical sensor systems is the photo emitter and detector pair. This pair consists, of one Light Emitting Diode (LED) that radiates light in the infrared frequency range and a photo detector that reacts when hit by infrared light. The photo emitter is usually activated at all times, but the detector only lets current through when hit by light from the emitter.

The photo detector or sensor is essentially a photosensitive resistor. The junction is made of a material that conducts only when hit by photons of a certain wavelength [6]. Figure 2-4 below, shows the build up of a photo detector. In order to make it possible for the micro controller to sense the signal from the sensor, a load resistor must be connected in series with the photo detector to ground. The signal line to the micro controller must be connected before the resistor as shown in Figure 2-4.

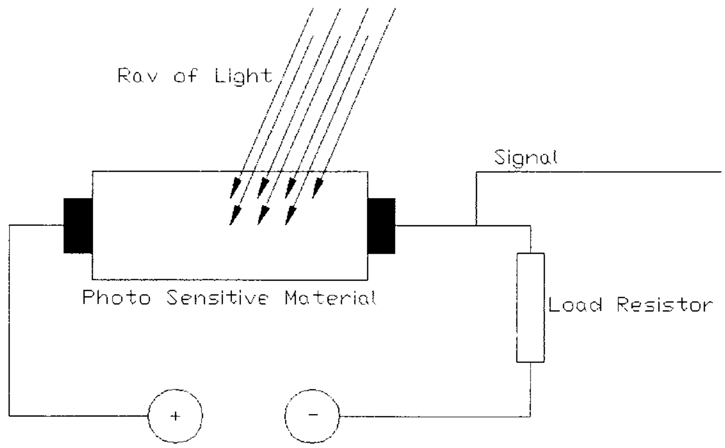


Figure 2-4. Photosensor connected to power source with load resistor connected in series.

The load resistor should be of relatively high resistance so that it can create a voltage divider where the majority of the voltage drop is over this resistor. Since the photo detector sends current through only when hit by infrared light, this is the only time when there is a voltage potential over the load resistor. Some voltage drop is over the photo detector, but if the load resistor is high this drop is very small when current is let through. A high resistor also prevents high current from flowing through the detector and therefore increases the detector's lifetime.

The emitter detector pair needs a disk or a material that can interrupt the beam of light line between the two at a desired location. Such location could be a point on a wheel so that position of the wheel can be determined. While emitter detector pairs are often used for absolute and incremental optical encoders, many applications involve the detection of a single position, like a limit switch. In this application, the emitter-detector pair is used both for incremental encoding and limit detection applications.

Optical sensors can be configured to have a positional accuracy of less than a millimeter. Compared with the hall effect sensor explained next the optical sensors can therefore be more accurate as position detectors.

2.4.2 Hall Effect Sensor or Proximity Sensor

The hall effect sensor is another type of sensor that can be used to identify positions. It is a proximity sensor because it "senses" magnetic fields within its close proximity. The hall effect sensor is based upon the principal that if a magnetic field is introduced on a current carrying conductor, voltage is generated transversely across the conductor (see Figure 2-5).

Figure 2-5. Hall effect sensor working principal
(Taken from: http://www.micronas.com/products/overview/sensors/index.php)

There are several types of hall effect sensors, both analog and digital. Analog sensors output a voltage drop that will increase with increased magnetic flux. Digital sensors detect magnetic flux, but if the strength is not high enough it will not exceed the sensors threshold. Then it will appear as if the sensor did not detect and it will not output any transition of signal. In the digital hall effect sensor class, there are several subgroups: Latched, Bipolar, Unipolar, and Unipolar/Inverted switching sensors. The Latched and the Bipolar-switching sensors will turn to low when a magnetic south pole approaches and turn to high when a magnetic north pole approaches. The Unipolar-switching sensors will turn to low when a magnetic south pole approaches and remain unchanged when a north pole comes within range.

The digital hall effect sensor chip has an onboard Schmitt trigger filtering the signal that the hall effect sensor outputs and converting the sensor output to a digital signal. A Schmitt trigger has an upper and a lower threshold. This means that when the signal from the sensor goes from low to high, the Schmitt trigger will filter out all signals lower than the upper threshold. Furthermore, if the transition of signal from the sensor is from high to low, the Schmitt trigger will cause the output signal from the chip to remain high until the signal from the sensor has dropped below the lower threshold. Therefore, hall effect sensors that have onboard Schmitt triggers have operation characteristics that follows a hysteresis. In practice, this means that if a magnetic object is being moved towards a hall effect sensor chip until it triggers and then immediately being moved the opposite direction, the hall effect sensor will remain triggered for a period after the direction of the object was switched.

The advantage of hall effect sensors over optical sensors is that they can operate in harsh environments where pollution is high and temperatures vary greatly. In contrast to the optical sensors and emitter pairs, which need 4 connectors, the hall effect sensor only needs three connectors, which are all on the same chip. This makes the hall effect sensor easier to mount and use when measuring rotation on objects that rotate relative to the frame. All that is needed is to mount the hall effect sensor on the frame and a magnet on the rotating object. Furthermore, the hall effect sensor can also operate under very high on-off frequencies. The delay or the fact that the on-off status follows a hysteresis causes hall effect sensors to be inaccurate as pure position sensors, but they work well for motion detection.

One example of the hall effect sensor's use is determining the number of revolutions of a rotating object. One such application for the sensor is the speedometer for a bike. A small magnet is attached to the spokes of the bike wheel and the sensor is mounted on the fork of the bike. The sensor will send a pulse to the micro controller each time the magnet is in proximity of the sensor. Speed can be determined by measuring the time passed between each pulse and distance can be measured by counting the total number of pulses corresponding to turns of the wheel.

2.4.3 Potentiometer

A potentiometer can be used to determine position for actuators that have a limited working range. The potentiometer is a variable resistor that consists of a wiper which makes contact with a resistive element. The resistance between the wiper and the connectors is changed as the wiper moves along the element as shown in Figure 2-6 and Figure 2-7.

Figure 2-6. Potentiometer resistor wires and wiper connection

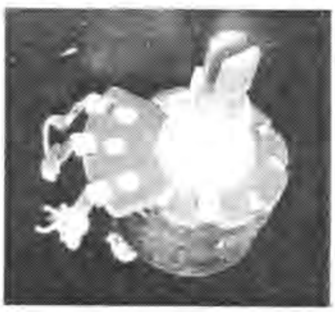


Figure 2-7. Potentiometer

When the potentiometer wiper position is coupled with the output of the actuator, the position of the actuator can be determined by measuring the variation in voltage potential as the actuator moves.

2.4.4 Comparator

In many cases, it is necessary to compare two voltage potentials. As explained above for the potentiometer, different positions on a shaft correspond to different voltage potentials. The same thing is true when using a tachometer, although the voltage potentials generated then refer to the speed of the shaft it is connected to. The comparator, together with the above-mentioned sensors, can be used to signal the micro controller when a certain position has been obtained or when a certain speed has been reached.

This is accomplished by defining a certain reference voltage on the reference pin of the comparator. The potentiometer or the tachometer needs to be hooked up to the input pin of the comparator. When the voltage potential on the input pin has overcome the reference voltage or threshold, a signal is emitted of the out pin of the comparator. This signal can either be a transformation from a logical high signal to a logical low signal or vice versa.

The comparator is in reality an Operational Amplifier (OpAmp) that is connected to sense (Vin) and compare (Vref) voltage potentials. The reference voltage has to be put on the positive + input pin, while the sense voltage is put on the negative – inverting pin of the op-amp. Additionally, the op-amp needs to be connected to power (V+) and ground. The voltage potential that is put on the power pin will be the same potential as the one read on the output pin, see Figure 2-8 for connection details.

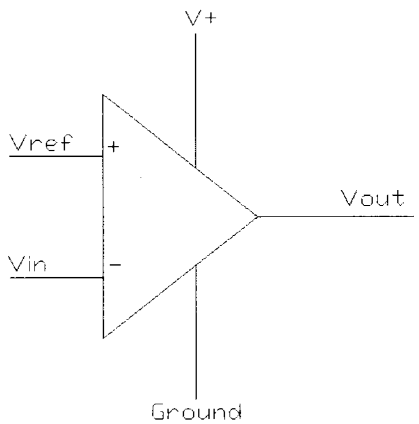


Figure 2-8. Voltage comparator

When the voltage to be sensed (Vin) or compared exceeds the reference voltage (Vref), the output voltage (Vout) falls to 0.

In this project, this sensing or comparing is being used to control current through a DC motor by attaching the sense pin to a load resistor in series with the DC motor. As the current increases through the motor, the voltage over the load resistor also increases and will eventually exceed the reference voltage of the comparator.

It is possible to obtain the same reading of voltage potentials by using the micro controller, but if the controller is relieved from doing the task it will respond faster to other tasks.

2.5 Micro Processors and Micro Controllers

The microprocessor or micro controller is the “brain” of the mechatronics system. It makes it possible to automate tasks by using sensors and actuators. The main difference between a microprocessor and a micro controller is that the micro controller is a stand-alone microcomputer on a single chip, while the microprocessor must cooperate with external components like memory, to work as a computer. Therefore microprocessors are used mainly for PCs or computers. The micro controller is smaller and cheaper and thus more suitable for a mechatronics application where size and price is important.

The low price and the smaller size of the micro controller do come at a functional cost. Micro controllers are limited both in memory and clock speed. While the limited memory of micro controllers may also limit the complexity of the tasks possible, external memory may be available for some controllers that will provide expanded task capabilities. It is also possible to use multiple micro controllers to expand the capabilities of a mechatronics system. Many micro controllers are designed to work in pairs [12].

There are two main methods of programming a micro controller. One uses assembly language, and the other uses higher-level languages such as C, C++, BASIC, and FORTRAN. Different micro controllers have different assembly languages. Language compilers on a PC will convert the assembly language and the higher-level languages into machine code, which is harder to understand and different for each micro controller. The advantage of higher level programming languages is that they are easy to understand and program. The standardized higher-level languages are therefore easier to implement when using different micro controllers. However, the disadvantage with the higher-level languages is that they are less efficient and require more read only memory (ROM) in the micro controller. Due to the memory limitations, it may not always be possible to use higher-level languages [12].

In computer systems and micro controllers there are two main types of memory used, ROM as mentioned above, and random access memory (RAM). The ROM stores the program code of the chip and is a read only type of memory. This means that the memory cannot be changed once the chip is programmed. This memory can be erased only by using ultra violet light through a crystal window on the chip or electrically by putting a voltage onto the erase pin if it is electrically erasable. The electrically erasable ROMs are called EEPROM (Electrically Erasable

Programmable Read Only Memory). When old data is erased, new data can be stored. The RAM memory can, as the name suggests, be accessed while the micro controller is executing its tasks and can also be changed. This memory is used to store dynamic variables, which refer to user input or sensor inputs, and to executing the program itself.

Examples of micro controllers available are Motorola's 68HC11, Intel's 8096, National Semiconducor's HPC 16040, and Microchip's PIC 16C71.

CHAPTER 3 – Mechatronics Design Approach

3.1 Design Considerations

3.1.1 Design Requirements

Winegard Company, who is sponsoring this research project, determined the design requirements. The desired requirements specified the overall size of the system, the deployment speed, and functions.

3.1.1.a Size

The antenna system is to be mounted inside the RV where space is limited. One concern is that if the system's enclosing is too big it could cause possible injuries thru collision. It is also important that the enclosing is aesthetically pleasing. The goal is therefore to build the antenna deployment unit as small as possible, especially focusing on the height below the ceiling. Winegard has specified that the enclosing may be no more than 4 inches thick, while the width dimension of the box is not as important, but should be minimized.

3.1.1.b Deployment Speed/time

The manual version of the antenna can be cranked up in less than 10 seconds by an energetic use. It is therefore important that the automated version deploy quickly enough to prevent the antenna from becoming a point of irritation.

As there is no human work connected with automatic deployment, most people will accept a little longer deployment time. Deployment time requirement is set to a maximum of 20 seconds.

3.1.1.c Functions

The antenna must have operational functions that include collision detection, orientation detection, status indication lights, automatic parking, and remote control. In addition, it is desirable that antenna be equipped with location memory so it will be easy to reposition the antenna when changing channels.

3.1.1.c.a Collision Detection

The movement of the antenna as it is deployed occurs on the outside of the RV and cannot be seen from the inside of the RV. It is possible that the RV may be parked in a location where trees or construction are blocking the antenna workspace. In order to prevent breakage of the antenna mechanism and to avoid damaging the surroundings, the antenna must be able to detect collisions and stop or retract when it hits an obstacle. This is especially important when deploying, but is also important when performing a search for a clear signal.

3.1.1.c.b Position Orientation in Boot-Up Phase

It is important to conserve energy in an RV, especially if parked somewhere where there are no electrical hook ups. Most RV owners wish to turn off the engine when they park, which means they use only battery power. Most RVs have additional generators that can run independently from the engine, but these will run only when necessary. Any unnecessary function should be turned off in order to avoid running the batteries down. It can be expected that people would want to turn off the antenna while watching TV. It is therefore important that the controller be able to determine the position and state of the antenna upon rebooting.

3.1.1.c.c Indication Lights

Since the antenna is operating outside the view of the people inside the RV, it is necessary for the deployment unit to be equipped with indicator lights that show the user what the antenna is doing. There should also be an indicator that shows when the antenna is fully deployed. For factory installation, this could be connected to the dashboard so the driver can see if the antenna is deployed before he starts driving.

It is also necessary to have lights that indicate when the antenna is deploying or searching, but also when the antenna is parked. These indications can be either a constant or blinking light.

3.1.1.c.d Remote Controlled

Depending upon where the antenna is placed on the RV it may be difficult to adjust for TV signals and watch the TV at the same time. The alternative is to either run back and forth between the TV and the deployment unit, or have someone watch the TV while the adjustment is being done. With a remote control the adjustment could be done while sitting on the couch watching TV. It could also make it much easier to switch channels on the TV and adjust the antenna repeatedly. If the deployment unit has a built in last station memory it would be very easy to switch channels by simply pushing one button on the remote control.

3.1.1.d Visually Pleasing

The deployment unit will be mounted inside the RV and will be visible to the occupants. The quality of RVs is improving every day. In fact some of the most expensive ones cost more than a regular home and are often times classier than the average house. This deployment unit is intended for the high-end models, therefore it is important that the unit has a nice appearance.

3.1.2 Actuation Requirements

The physical requirements include the number of turns required for full deployment on the inner shaft and the revolution path of the antenna. Additionally the actuators used must overcome certain torques and frictions necessary to deploy and search.

The number of revolutions needed on the main shaft or inner shaft to fully deploy the antenna is fourteen revolutions. When the antenna has been mounted on the RV roof this will change and depend upon what angle there is between a fully deployed antenna and the roof. Fourteen revolutions will move the antenna approximately 100 degrees from fully functionally parked to fully deployed, but after the antenna has been mounted the actual degree between the roof and the fully deployed antenna may decrease to less than 90 degrees. However, the requirement concerning deployment time applies to the theoretical moving angle and the deployment time may therefore be less than what the deployment unit was designed for.

The torque requirement for deployment of the antenna was measured physically on the antenna to be minimum 200 Oz-in. This measurement was done by attaching a weight to the original crank and measuring the maximum force needed to overcome friction in the shaft. The arm length was measured between the shaft center and the end of the crank. Since friction may increase in the shaft and mechanism over time, a safety factor was built into the design and the requirement for the torque was set to 250 Oz-in.

3.1.3 Constraints

The main constraints for the project were the dual rotation path, and the deployment path, cost, and power consumption.

3.1.3.a Dual Rotation Path

The antenna consists of two rotation paths that need to be controlled thru an outer and inner shaft. The outer shaft or rotation shaft is an extension of the rotational housing of the antenna and is connected to rotation dial inside the RV. The inner shaft or elevation shaft is hexagonal and extends concentric thru the outer shaft and via a worm it engages with a worm gear inside the antenna housing. The antenna bars and the antenna are connected to the housing and the inner shaft and gears are used to elevate the antenna.

The dual rotation path consists of the inner shaft, which is used to elevate or park the antenna, and the housing extension, which is used for rotational adjustment of the antenna for tuning.

The antenna can be actuated three different ways by rotating only the inner shaft, both the inner shaft and the outer shaft, or only the outer shaft. If only the inner shaft is rotated the antenna will move up and down in the elevation. If both shafts are rotated together in the same direction the antenna will rotate. (They can also be rotated in opposite direction, but that does not serve a purpose). And finally if only the outer shaft is rotated the antenna will also rotate, but on a full travel from fully ccw to fully cw the antenna will also move in the elevation a few degrees. This can lead to problems if the antenna is fully deployed and attempted moved clockwise. The antenna is then being forced further up in the elevation even though it is already at its end point of possible travel, which will strain the gears and possible lead to breakage. The antenna mechanism is shown in Figure 1.2, 1.3, and 1.4.

The full range of travel in the elevation is fourteen revolutions on the elevation shaft. For the other rotation path, the adjustment path, the rotation from fully ccw to fully cw is approximately 270 degrees.

3.1.3.b Cost

The deployment unit will be a retrofit to the existing antennas that have already been sold. The deployment unit will therefore add cost to an already existing system. In order to make the consumer willing to buy the product it should be a low cost system to reduce the price on the final product. The product price should not exceed 50 dollars.

3.1.3.c Power Consumption

Since there is a limit to how much power can be put out from an RV's power source, especially when not connected to electrical hook ups, the final product needs to be low in power consumption. This is especially important when the deployment unit is in idle/sleep mode or in other words when none of the actuators are moving. If the unit draws too much power when not in use, it will generate heat and draw all power from the battery so that eventually the RV cannot be started.

Chapter 4 – Development of Prototypes

The two main steps in deploying the antenna is to crank up the antenna and rotate the antenna until the "best" picture is found. A design was developed to use a single motor to actuate both the up/down functions and the rotate-to-tune function. Prototype A was developed to test the functionality of this design. After development several changes were made that resulted in Prototype B. A second design with two motors was also tested and is described as Prototype C.

Winegard provided a commercially available Winegard Sensar® antenna to be used for the prototype development. It was mounted to a flat surface according to the manufacturer's directions and specifications see Figure 4-1.

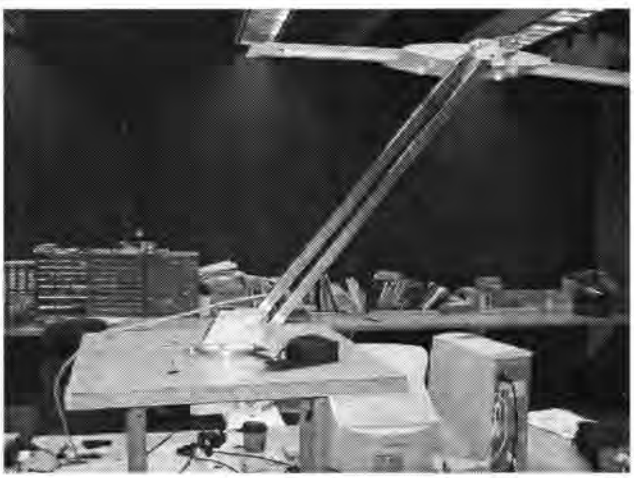


Figure 4-1. Antenna mounted on the test table.

4.1 Prototype A

This prototype was built using off-the-shelf components to verify the operation of the design. The functional parts of the design include drive train, solenoid, engage plate, and sensors. Figure 4-2a and b shows the full assembly of Prototype A.

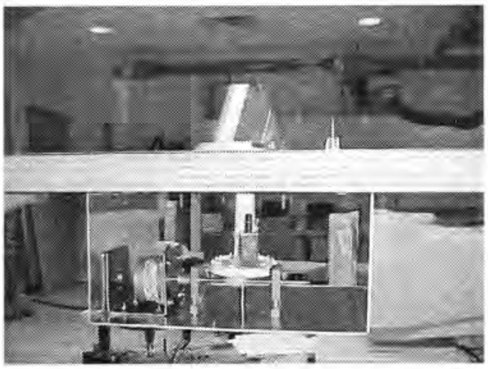


Figure 4-2a. Prototype A

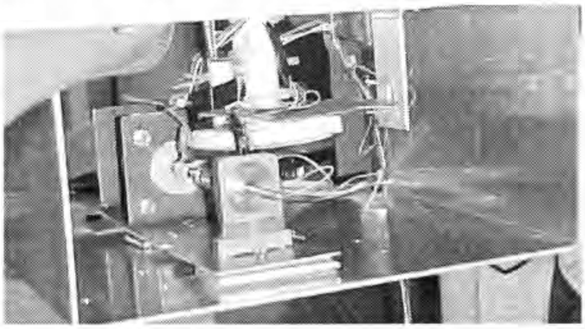


Figure 4-2b. Close up of Prototype A

4.1.1 Housing and Support

The housing consists of an aluminum frame that has been bent into a square channel. The frame can be attached to the ceiling of the RV, or in this case onto the test table. Figure 4-2a shows how the elevation shaft extends through the center of the aluminum plate.

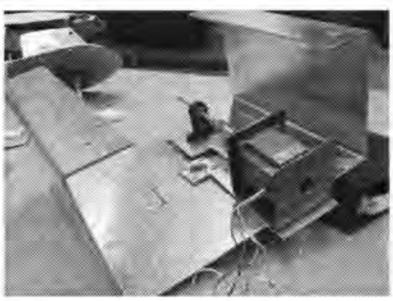


Figure 4-3. Aluminum frame of Prototype A with stepper motor attached

The original plastic adjustment dial was modified to hold a solenoid and a hole was made for the solenoid pin to go through. The plastic dial was then mounted onto the outer shaft like the original version of the manual antenna. Figure 4-4 shows these details. An engagement plate was mounted onto the elevation shaft for the solenoid to engage with.

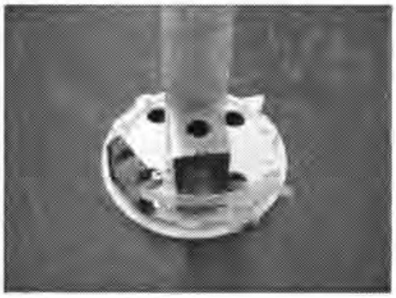


Figure 4-4. Rotation adjustment dial

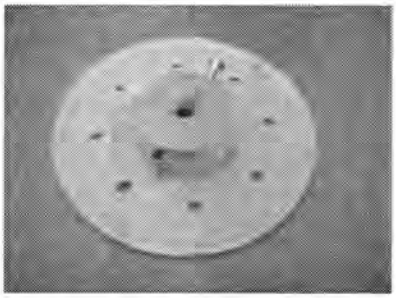


Figure 4-5. Engage plate

Figure 4-6. Solenoid, adjustment dial, and engagement plate

A worm gear was mounted onto the elevation shaft at the correct height for a worm on a motor shaft to engage. The motor mounting was made for a stepper motor and couplings so that the original motor shaft could be extended with a longer shaft. The worm was mounted onto the extension of the motor shaft. Holes were made in the aluminum frame for the motor assembly to be attached. The assembly can be seen in Figure 4-2, 4-3, 4-4, 4-5, and 4-6.

The antenna has enough support between the base and the attachment plate so that no additional bearings were necessary here. A bearing support was put on the lower end of the elevation shaft in order to prevent shaking and to transfer torque from the motor to the shaft without having the shaft bending and the gears slipping.

Supports were also made for the motor shaft where one bearing was put on each of the two shaft supports to hold the motor shaft and provide low friction. Moreover, the motor had additional internal bearings of high quality that added to the motor assembly support. The motor, shaft and shaft supports were put on a plate and slots were made in the frame so that the motor assembly could be adjusted in and out from the center of the frame channel to accommodate for different gear sizes in the prototype phase.

4.1.2 Drive Train

The drive train consists of a stepper motor, a worm, and a worm gear. Figure 4-7 shows the drive train.

Figure 4-7. Drive train

4.1.2.a Stepper Motor

A stepper motor was chosen because it provides accurate open loop position control. Several motors were investigated and tested to find one that would be suitable for the task at hand. Two motors were found to have enough torque to drive the antenna up, 57BYG084 and 57BYG085. 57BYG084 was supposed to have higher torque at same pulse intervals as the 57BYG085. However, this was not the case when the motors were tested. Since the torque was equivalent the 57BYG085 was chosen because of its smaller size.

The 57BYG085 is a bipolar hybrid motor with step angles of 1.8deg/step. The rated holding torque at 12V was 35.28 N-cm with a detent torque of 3.33N-cm. The detent torque is a small residual holding torque that is present even when the motor is not energized. The running torque for this motor can be seen in Figure 4-8 below.

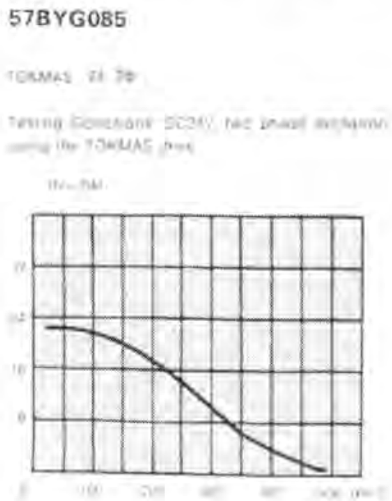


Figure 4-8. Performance curve for the 57BYG085 Stepper motor

The characteristics of this stepper motor as with other stepper motors is that when speed increases the torque falls rapidly, this can be seen in the graph in Figure 4-8. The numbers on the horizontal axis indicates pulses per second while the vertical axis indicates the torque in N-cm. The maximum speed that the motor could be run at under the load applicable to this design was found to be 210 rev/min. At this speed the antenna would deploy in 100 seconds.

4.1.2.b Worm and Worm Gear

The worm and worm gear was chosen with a reduction of 1:25. To reduce cost nylon plastic worm and worm gear were chosen. The worm gear was attached to the elevation shaft and the worm was attached to the motor shaft. Different reductions were tested but lower reductions required more torque from the motor which meant that the motor had to be driven slower. The conclusion was therefore that to optimize the speed and take advantage of the torque produced by the motor the best solution with the existing motor and drive was to use a reduction of 1:25. If lower reduction was to be used a higher torque motor configuration was needed.

4.1.3 Solenoid

As explained the Sensar® antenna has two main shafts to control the antenna movement. The inner/elevation shaft is for up and down movement and the outer/rotation shaft is for the rotate-to-tune function. To move the antenna up or down the inner shaft needs to be rotated, while if the antenna is to be adjusted in the rotational direction the outer and inner shaft must be rotated together. A Solenoid was chosen to lock the two shafts together.

The solenoid was mounted vertically on the adjustment dial that was originally used to turn and tune the antenna. This piece was attached to the antenna base. A hole allowed the solenoid pin to extend and lock the two shafts together. A custom made disk was attached to the elevation shaft . The idea was that when the solenoid was activated it would extend through the adjustment dial and into one of the holes in the disk. The two shafts would then rotate together as the motor ran. Movement of the antenna would go from vertical adjustment to rotational adjustment and only one motor was needed.

It was important to choose a small solenoid because of the size requirement. However, the solenoid had to have enough force to engage and lock the two shafts together. Since the only purpose of the solenoid was to engage and hit a hole in the engagement disk it was thought that the solenoids did not have to be very powerful. However, testing of several solenoids made it clear that it needed to be fairly forceful in order to engage and stay engaged as the two shafts rotated together. The solenoid can be seen in Figure 4-9 below.

Figure 4-9. Solenoids, S-18-100 closest, S-20-150 farthest away

4.1.4 Engage Plate

The engagement plate, shown in Figure 4-5 provides the capability for using single motor for both functions. It is attached to the inner elevation shaft. The solenoid, attached to the outer shaft engages and connects the inner and outer shaft together allowing the motor to drive the turn and tune.

4.1.5 Sensors

For the Prototype A there were two types of sensors used, a photo emitter/detector pair and hall effect sensors. A half circle encoder was made and mounted onto the adjustment dial. The photo sensors were placed at the center of the antenna mounting plate and they would sense to which side right or left the antenna was at all times. The system would determine center location by moving until a change in the sensor status occurred. This only happens when the antenna is in the center position. The optical sensor and encoder plate can be seen in Figure 4-10 below.

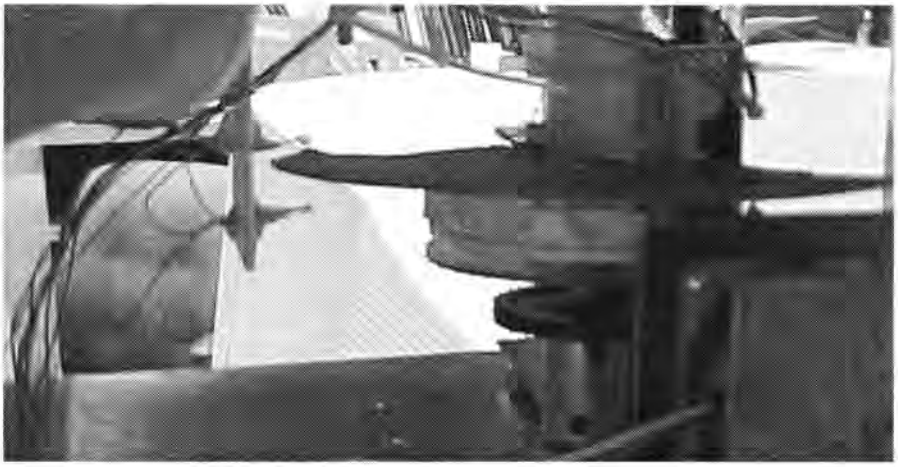


Figure 4-10. Optical sensor and encoder plate

Two Hall effect sensors were mounted onto the white adjustment dial 90 degrees apart while a rare earth magnet was mounted onto the worm gear. Relative motion between the worm gear and the adjustment dial could then be defined. This needed to be defined because the solenoid was mounted on the adjustment dial and the plate with the holes for the solenoid to engage in was mounted on the elevation shaft along with the worm gear. In order to make the solenoid engage at the correct position the hall effect sensors were intended to be used to determine when the holes were lined up with the solenoid and when the antenna was deployed or parked fully. However, since the hall effect sensors were not accurate enough the system was later programmed to engage shortly before the hole and then slide on the engage plate until hitting the hole.

The two hall effect sensors were defined separately as up and down sensor. In the deployment phase of the actuation the system would count revolutions by counting signals from the up hall effect sensor. Fourteen signals equaling fourteen revolutions indicated that the antenna was fully deployed. The system was programmed an designed so that when the antenna was fully deployed the up sensor would always be positioned directly above the rear earth magnet on the worm gear, while if the antenna was parked the down sensor would be positioned directly above the magnet. This way it could be determined whether the antenna was parked or deployed when the system was powered up. If the antenna was not fully deployed or parked before the power was cut it would not be possible for the system to determine where the antenna was positioned. When the antenna was rotating the magnet should always stay directly under the up sensor. If they were moving relative to each other the antenna would not move rotational but in the elevation and this is an indication that the solenoid was not locked.

4.1.4 Micro Controller

The micro controller used for all prototypes was Microchip PIC16C71. This is a controller that is widely used for mechatronics purposes and was found suitable to perform the tasks that were required for this project. The controller can have four analog to digital (A2D) ports and 13 digital in/out (I/O) ports if all analogue ports are set digital. It can use both assembly language and a version of C programming called PIC C. The EPROM memory of the controller is 1Kx14 words while data memory or RAM is 36x8 bytes for the 16c71 version and 68x8 bytes for the 16c711 version which otherwise is identically and was also used for the project. The chip either uses an external oscillator or it can use an internal RC oscillator. However, the RC oscillator requires external capacitors and resistors to work. For this project an external quartz oscillator of 4 MHz was used but the controller can be used with processor speed of up to 20MHz. The quartz oscillator was used because it gives a very steady, reliable oscillation and was therefore preferred over the onboard RC oscillator, which does not produce as steady and reliable pulse.

4.1.5 Functionality of Prototype A

The first prototype was controlled using two buttons. One button will deploy and park and the other will start/stop the tuning. As mentioned in the hall effect section above when the system was booted after having been turned off it would search the Hall effect sensors to detect the position of the antenna. The system had one Hall effect sensor for the deployed position and one Hall effect sensor for the parked position 90 degrees apart.

When the deployment system was turned on and it was determined that the antenna was parked the only possibility would be to deploy it. Deployment was commanded by pushing the deploy/park button. Once this button is pushed, the motor begins to move to deploy the antenna. The micro controller counts each time the “up” Hall effect sensor is triggered and when it has moved 14 revolutions the motor stops.

The antenna is also equipped with a search button. The button is a single pole double throw button and it turns on when pushed once and turns off when pushed one more time. When the button is on the antenna will start to search automatically after it has deployed. The antenna will continue to search until the search button is pushed off. If the search button is pushed off the antenna will stay deployed but will not search until the search button is pushed.

In search mode the micro controller is programmed to first find the center position and then search from side to side. The center is found by reading the photo sensor, when the photo sensor changes its status the center is found. If the photo sensor is low the antenna must be moved ccw to find center and if the photo sensor is high the antenna must be rotated cw to find center. After the center has been located the system will use the stepper motor to move a certain amount of steps to each side. After the stepper motor has moved the commanded amount of steps the system will perform the “find center” routine explained above and move until it reaches the center point. Then the motor will move in steps to the other side again and go back to center and so forth.

The system has been programmed so that the search needs to be stopped before the antenna can be parked. When the deploy/park button is pushed the antenna will do the find center routine explained above, release the solenoid and move down reading the down hall effect sensor until 14 readings or revolutions have been registered and stop. Please see float chart in Figure 4-11 for procedural details.

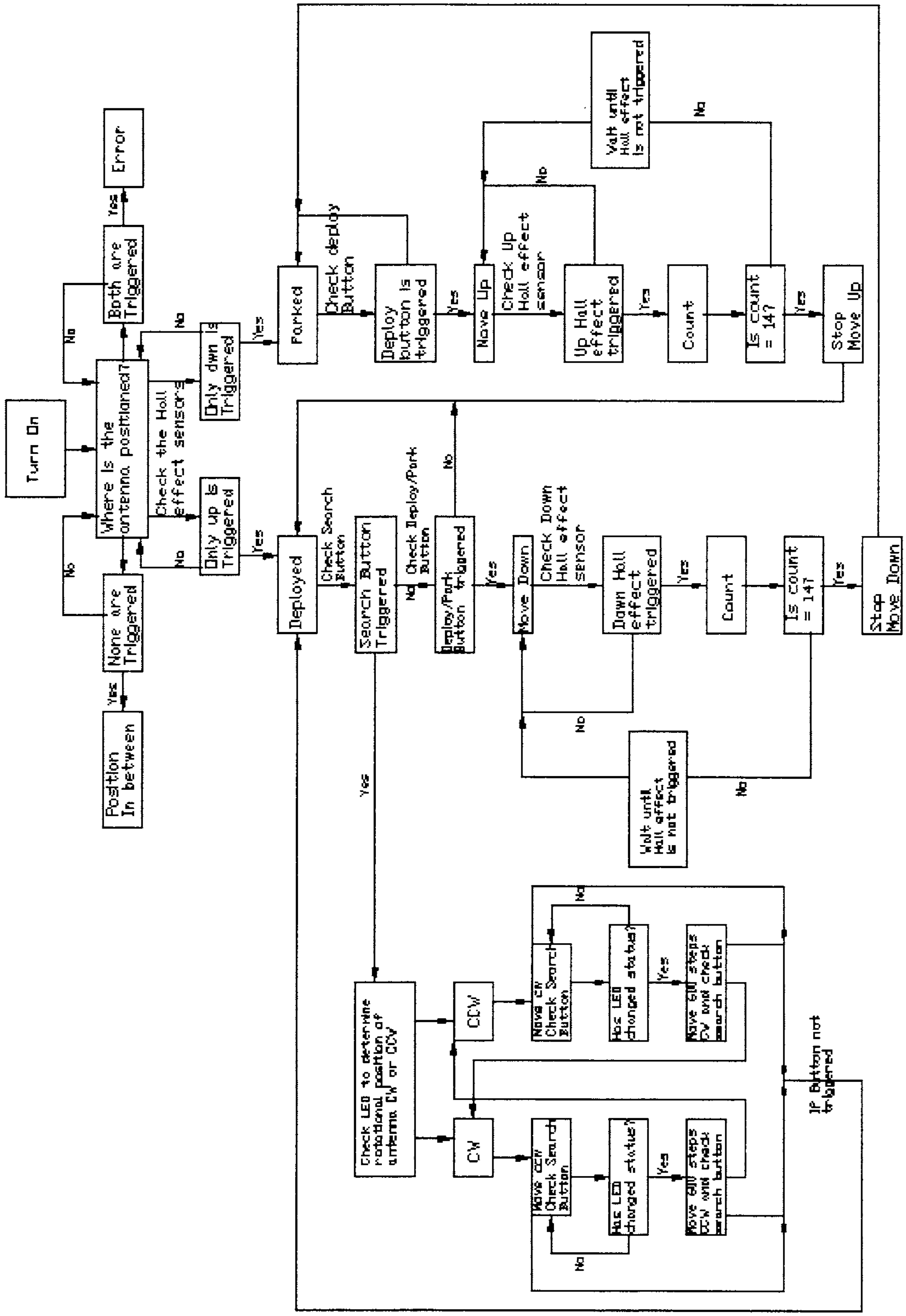


Figure 4-11. Operational flow chart Prototype A and B

There are three predefined speeds or delays between the stepper motor pulses. The maximum speed is set for parking the antenna because the torque needed for parking is less than for deploying. The deploy speed is defined as the medium speed and is a little slower than the park speed because the antenna requires more torque when moving up than moving down. The slowest speed is set for the search mode, which requires that the antenna move so slow that it is possible to watch the TV and find the best signal and stop the antenna.

4.2 Prototype B

Prototype A was presented to a team from Winegard that included representatives from engineering, manufacturing, purchasing, product development, customer service, and marketing. The team critiqued the design from a commercial viewpoint, which resulted in a list of shortcomings and concerns. The main concerns were linked to noise level, speed, and size. Prototype B is in essence the same as prototype A, but includes refinements of the concerns from the Winegard team.

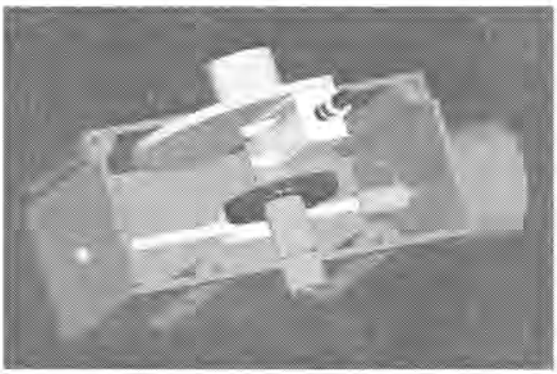


Figure 4-12. Prototype B, ProE model

4.2.1 Housing and Support

The main difference between prototype A and prototype B is the placement of components. Where Prototype A was mainly made by hand with low precision tools Prototype B's main parts were made on a stereo lithography machine provided by CIRAS at Iowa State University. The parts that were made on this machine were the enclosing frame, the encoder plate for the photo detector, and a worm gear retrofit.

The supports were also changed for this prototype, the two bearings that were used to hold the motor shaft in prototype A were replaced with one bearing in prototype B. The one bearing was placed in a slot in the box and the motor shaft would extend through the slot and a hole in the side of the box. The motor bearings for the stepper motor were found to be of high quality and would be efficient at supporting the motor end of the shaft.

The bearing for the elevation shaft used in the first prototype was replaced by a worm gear retrofit made of soft plastic. This provided low friction between the plastic enclosing and the plastic gear retrofit would work as a sliding bearing. The same slide-bearing principal was used to support the encoder disk where the solenoid was now mounted. The encoder disk was also made of soft plastic and the sliding friction between it and the box was low. To prevent wear the slide bearings were to be replaced by a thin layer of frelon or similar low friction bearing material. The drive train and supports can be seen in the proE drawing model in Figure 4-12 above.

4.2.2 Drive train

The actuators for this prototype are the same as for Prototype A. The system uses a worm and worm gear configuration where the elevation axis is 90 degrees relative to the motor drive axis. However, for this prototype the gear ratio cannot be changed. It was found that 1:25 was the optimal ratio by testing prototype A.

4.2.3 Solenoid

The solenoid position has been changed from vertically to horizontal. This change decreases the thickness of the overall design, but it increases the necessary width of the design. In addition, the solenoid engaging plate of the first design was replaced with a slot on the worm gear retrofit. This slot was later reinforced with an aluminum ring with the same diameter as the solenoid pin. The reason for the reinforcement was to prevent the system from twisting when the solenoid was engaged.

4.2.4 Sensors

The sensors for this prototype have essentially the same placement as for the first prototype. A photo sensor emitter/detector pair was placed at the center of the drive shaft side of the box as shown in Figure 4.13b below.

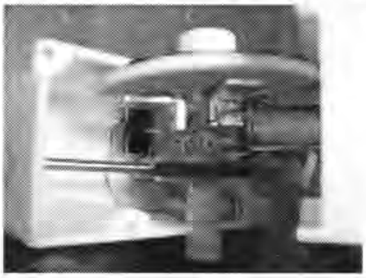


Figure 4-13a. Inside assembly of Prototype B

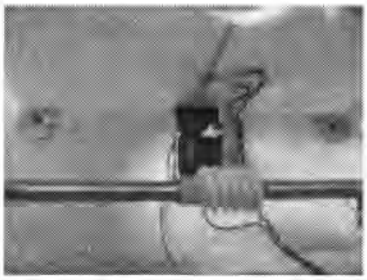


Figure 4-13b.Optical sensor

The decoder plate was shaped essentially the same as in prototype A but since it was made of more solid material it would not deflect and get tangled when rotating. The photo detector reading would show on which side of center the antenna was, but to find center the antenna had to be moved until the sensors detected a change. The encoder plate can be seen in Figure 4.13a above.

4.2.5 Functionality of Prototype B

Prototype B has the same functions as prototype A but an infrared remote control unit has been added in addition to the indicator buttons in Prototype A. The user therefore has the choice between using the buttons on the side of the box and using the remote control. There are four buttons on the box, see Figure 4-14.

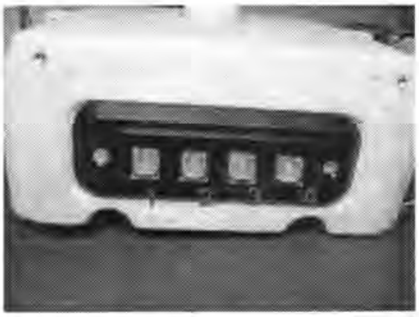


Figure 4-14. Buttons

The left most button, labeled #1 is the deploy/park button and button #2 is the search button; they both have the same functions as in prototype A. An adjustment feature was added in prototype B, which means that if button #3 is pushed the antenna will move 1/14 of the full range down, while if button #4 is pushed it will move 1/14 of the full range up. This feature can be used to adjust the antenna if it has been turned off without being fully parked or if the system for some other reason cannot detect the magnets in parked or deployed position. By pushing one of these two buttons the antenna will move in the corresponding direction until a magnet is found by the down Hall Effect sensor. If the antenna does not move when the deploy button, #1 is pushed this emergency procedure can be used to find the magnet and make the antenna move again.

4.2.6 Chopping Circuitry

It was thought that the reason why the stepper motors could not run faster than they did was because the coils in the stepper motor needed time before they let the full current through. In order to make the shifting of polarity of the magnetic fields go faster chopping circuitry can be introduced. The problem with electromagnetic coils is that it takes time from a coil is energized until maximum current through it is obtained. This reaction/response time is governed by the differential equation where V is the voltage applied, L is the inductance and I is the current:

$$V(t) = L\frac{dI}{dt}$$

In order to change the direction of current faster and obtain full current in shorter amount of time the chopper circuitry introduces a higher voltage than what was first used. For the L297 and L298 stepper motor driver and H bridge pair a voltage of typically 40 V is being used and put over the coils. The L298 H Bridge used for driving the motor is rated for potentials up to 46 volts. The increased voltage makes the maximum current reach its maximum faster, but in order not to burn the coils with the high voltage, sense lines are drawn from the H Bridge to the L297 stepper motor driver. A load resistor is connected between the H Bridge and ground and the current through the motor coils also passes through this resistor. When the current increases the voltage drop over the resistor also increases and finally the voltage reaches a predefined reference voltage set on the L297 driver. When this reference voltage has been reached the L297 sends a signal to the L298 bridge to shut the transistors to cut the current through the motor coils. The schematics can be seen in Figure 4-15 below.

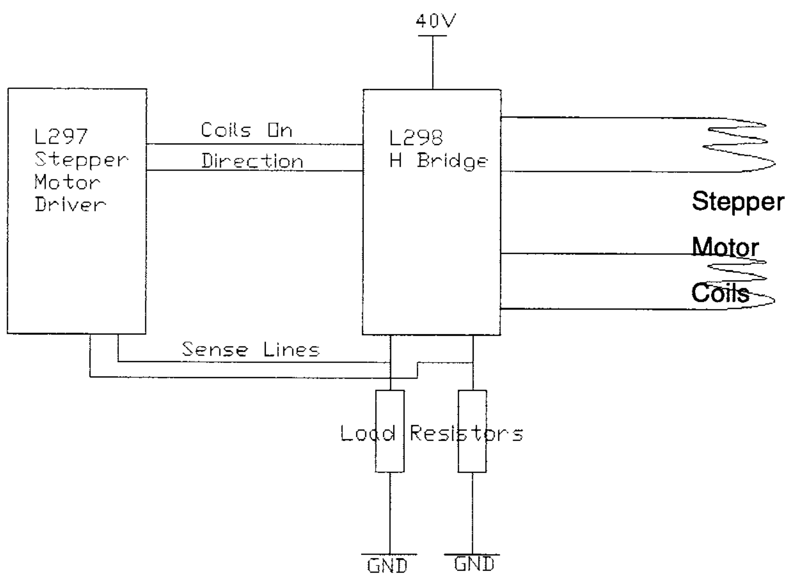


Figure 4-15. L297 and L298 stepper motor driver and H bridge pair used for chopping circuitry.

4.3 Prototype C

After presenting Prototype B to the Winegard team the feedback was mainly that the prototype was still performing too slow. Some of the other concerns mentioned were wire wrap for solenoid and solenoid engagement. At this point a new idea was developed; replace the solenoid with a second motor. The cost of the solenoids versus the cost of a second motor was compared. It was found that the cost of a second motor was approximately the same as the solenoid. Figure 4-16a and b shows the new design.

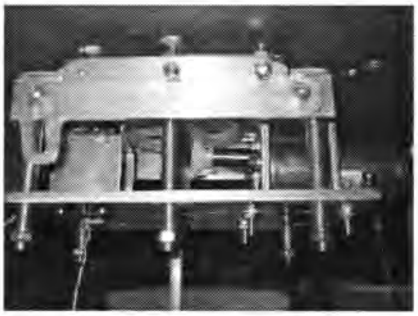


Figure 4-16a. Prototype C

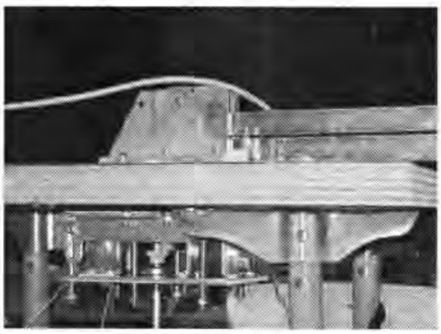


Figure 4-16b. Prototype C and antenna

4.3.1 Drive Trains

The drive trains are divided into two parts. One part is the elevation drive train, which consists of two miter gears with a reduction of 1:1. The other part is the rotation gear set which has a reduction of 12:84. These can both be seen in Figure 4.17 below.

Figure 4-17. Drive trains

4.3.1.a Elevation Drive Train

The elevation motor used for this design is a motor that has been custom made for Winegard. The motor is currently used for several other Winegard antenna systems. By using this motor for the design one can maintain a standard inventory, which can save the company cost. The motor is produced by Globe Motors and has the part number 455A679. The motor has a reduction of 1:130 and a no load speed of 39 rev/min at 12V with the torque rating of 300 oz-in. At 12 volt this configuration deploys the antenna in approximately 20 seconds.

The elevation drive train is supported by a ball bearing and a thrust bearing in the lower part of the framework. The ball bearing reduces friction for the rotation. Because a miter gear is being used there are forces pushing the shaft downwards and it was necessary to put a thrust bearing around the shaft to support this. A round shaft with hexagonal inner shape was acquired and shaped so that it would slide onto the elevation shaft and into the thrust bearing and the ball bearing in the lower end of the antenna. The ball bearing is locked into the lower aluminum plate of the framework while the thrust bearing is placed above the ball bearing resting on the top of the lower aluminum frame. One of the miter gears were mounted on this piece and used to transfer forces from the DC motor to the elevation shaft as can be seen in Figure 4-17. The elevation motor and corresponding miter gear is attached to the lower part of the framework via a motor mounting and screws. The motor is mounted so that the miter gear engages with the miter gear on the elevation shaft.

4.3.1.b Rotation Drive Train

The rotation motor used for the design is a Hsiang Neng motor with part number GH12-1641T-L. The motor has a reduction of 1:270 and a no load speed of 20 revolutions per minute at 12V. The torque rating at 12 V is 139 oz-in.

The rotation drive train consists of two spur gears with a 12:86 reduction. The large gear is attached to the antenna housing and supported by a ball bearing with the elevation shaft. The outer shaft of the antenna extends from the housing of the antenna. It has two slots, which are used to attach the big gear with setscrews. The smaller gear is attached and supported by the small DC motor and a ball bearing mounted in a hole in the upper part of the framework.

An important note is that there is no support between the upper framework and the big spur gear. The unit must therefore be mounted so that there is no stress between the deployment unit and the original shafts on the antenna. It must merely be pushed onto the shafts without forcing it sideways.

4.3.2 Sensors

Only two "real" sensors are needed in this prototype, the other sensors have been replaced by using the micro controller to sense the current through the motors. One Photo emitter/detector pair is placed on the motor mounting of the rotation motor. It detects the gear teeth as they rotate and make it possible to count teeth. Another sensor have been put on the elevation motor shaft to detect shaft revolutions.

A small resistor is hooked up to each of the motors. When the motors are running the voltage drop over the resistor is small because the bulk of the total resistance in the system is over the motor. When the load increases on the motor the current will increase through both the motor and the load resistor and the voltage over the load resistor will be at its maximum. This will occur when the motor for example drives the antenna into a fixed barrier like a tree or its end points. By using an op-amp voltage comparator it can be sensed when the motor is subjected to too much load or when the motor is driving the system into a physical barrier. The comparator reads the voltage drop over the load resistor to be equal or higher than its reference voltage and it sends a logical high out on its output pin. The controller reads this and makes the motors stop. The sensor on the motor shaft makes the micro controller able to read revolutions on the elevation shaft. The micro controller can then distinguish between a collision with an object and the end point of travel.

In the prototype this sensing is being used to define position of the antenna. When the antenna is being booted up after being powered down the micro controller will drive the antenna down and up and read the overload sensor to find out at which position the antenna is. This will be explained in more detail in the Functionality of Prototype C section.

The antenna also uses the overload sensors to find its end points when the antenna is rotating. The micro controller is programmed to change direction of the search when these end points are detected.

In order to find the center location for parking the antenna the micro controller will first home the antenna fully ccw until the end point is sensed by the overload sensor. When the endpoint has been located the micro controller will count teeth by reading the optical sensor until the number of teeth counted corresponds to the center location value.

4.3.3 Microprocessor

This system requires only simple programming to control the dual H Bridge used to control direction and actuation of the two DC motors. There are two options to sense the current through the motors, either the controller can be used directly to sense the voltage over the two load resistors, which are placed in series with the motors, or an op-amp comparator can be used. The comparator reduces the workload on the controller because it does not have to go through the analog to digital conversion process. It is therefore the preferable solution to the current detection. If the controller is relieved from this task it will perform the other tasks quicker and the response to inputs are more reliable.

This prototype also utilizes the infrared remote control and the deploy/park buttons on the side of the box. The programming therefore includes an infrared detection code to read signals from a TV remote control. The program code and explanation comments for this prototype can be seen in Appendix C.3.

4.3.4 Functionality of Prototype C

The available functions for prototype C is deploy, park, and search. When the antenna first is booted a search routine is programmed into the controller that will determine where the antenna is positioned. It follows the flow chart in Figure 4-18 below.

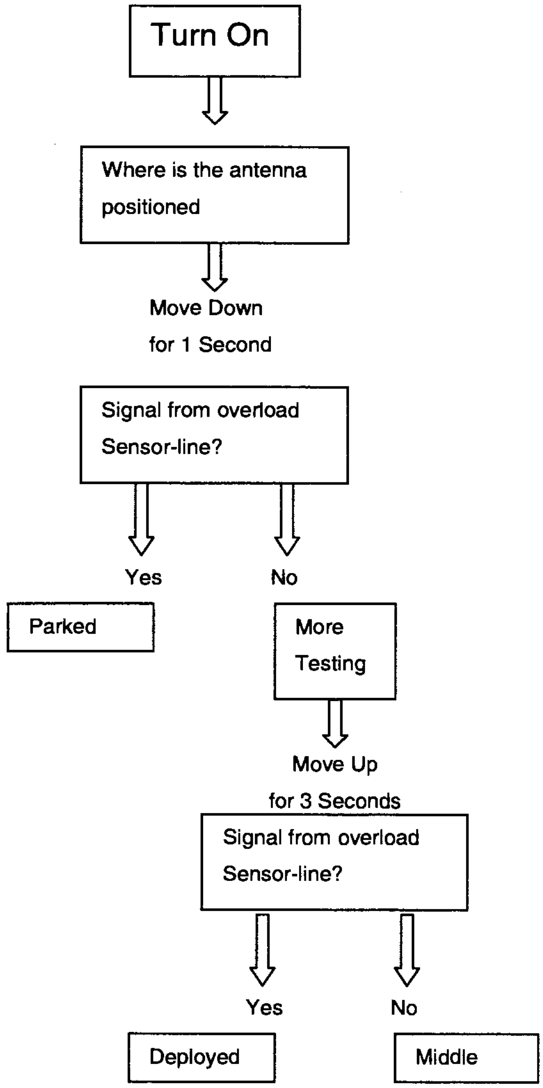


Figure 4-18. Start up procedure float chart for Sensar deployment drive

When the antenna system boots it will first move down for a short interval and if the controller senses that current increases through the elevation motor it indicates that the antenna is parked. However, if there is no increase in current the antenna will move up for an interval that is twice the time of the down interval and if the current increases it is deployed. The last option is that the current does not increase in either up or down movement for the intervals tested. If that happens the antenna is somewhere in the middle and it is the users choice whether he or she wants to deploy the antenna or park it.

When deploying the antenna it will move up until it hits the physical end barrier of the travel. It will then move back for a short interval to “loose” up the strain on the antenna and make it possible to move cw. If the antenna is moved up as far as it physically can be moved and is attempted to be moved cw without also moving the elevation motor at the same time it will not move. This is explained above in the section on the dual rotation paths. After the antenna has been moved up it will automatically go into search mode where it will move back and forth until the user hits the stop button. The antenna moves back and forth at a speed of approximately 3 revolutions per minute, which is slow enough for the user to stop it when good reception has been achieved.

If the user wants to park the antenna it will move to one of the end revolution barriers and count the necessary teeth to hit center and stop. After it has hit the center the elevation motor will drive the antenna down. The users also have the option of connecting the system to the ignition of the RV and make the antenna auto park if the RV is started. This can prevent damage to the antenna if the user forgets to park the antenna before driving.

CHAPTER 5 – Comparison of Prototypes and Conclusion

The three different design prototypes can be seen in Figure 5-1a, b, and c below.

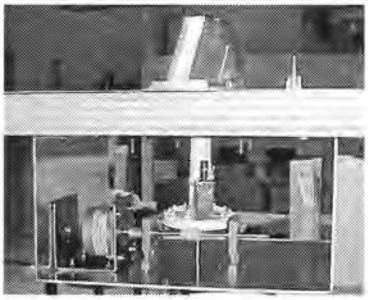


Figure 5-1a. Prototype A

Figure 5-1b. Prototype B

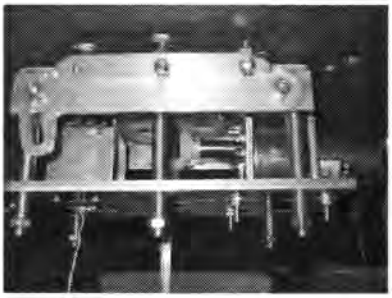


Figure 5-1c. Prototype C

5.1 Conclusions from Prototype A

Prototype A, see Figure 5-1a above, was the first prototype that was made for the project and it was anticipated that after testing this model many changes would have to be done. A complete parts list is available in Attachment A.1.1. Prototype A's main purpose was to test the functionality of the idea and how well it would work. The main problems with the first prototype were it was not made to precision, it made too much noise, the solenoid did not engage reliably, it had wire wrap from the solenoid and hall effect sensors, it was too big, it was not a retrofit, and it deployed too slow.

The lack in precision and the use of aluminum for the framework channel made the system very noisy. The aluminum channel would even resonate and amplify the already loud noise from the stepper motor running at high speed.

The lack in precision would also cause problems with the solenoid that was supposed to hit the engaging hole reliably each time. Since the encoder/solenoid plate was not supported against the shaft it would wiggle back and forth and it was therefore more luck than accuracy if it hit the hole. Since the solenoid and the hall effect sensors were mounted on a rotational frame the wires was wrapped around the adjustment dial and connected to the circuit board attached on the framework. The loose wires connected to the solenoid and the hall effect sensors would get tangled into other components and cause the system to work with lower reliability. The twisting also caused the wires to break and the solenoid and the hall effect sensors to stop working.

Although at this stage the design functionality was more important than the overall size requirements set down by Winegard, this prototype was too big, exceeding the requirements. Moreover, this prototype was not a retrofit to the already existing and mounted antennas. The unit was supposed to slide onto existing shafts and be ready to use. Prototype A would have to be mounted with setscrews, etc. in order to work.

However, the main problem with Prototype A was that the stepper motor would not run fast enough and with high enough torque. As a result the deployment time for the antenna was over 100 seconds, which exceeded the design requirements by over 80 seconds. In order to fix the problem several other stepper motor drivers were tested but none would give any significant change in speed.

5.2 Conclusions from Prototype B

Prototype B, see Figure 5-1b was built with the idea that the speed of the stepper motors could be increased. A complete parts list can be found in Attachment A.1.2. After building Prototype A it was found that in order to test the functions properly and to get down the noise level a better precision prototype had to be made. In addition, it was also important to minimize the overall size of the design. The second design's functionality is very similar to the first.

Since Prototype B is made out of mainly plastic, which absorbs and damps any shaking the noise was reduced to a comfortable level. The new prototype was also a retrofit to the existing antennas in that the shaft gear was made hexagonal on the inner faces and the solenoid plate was made to slide onto the base of the antenna. The overall size of the design was within the design requirements. The new prototype was 3 ¼" thick and the requirements indicated a thickness of 4" or less.

Although this new design is of higher precision the solenoid plate still could be better supported. The solenoid would at times still miss the engaging hole. Furthermore, the solenoid required a lot of power and would heat up quickly. The duty time for the solenoid was 90 seconds at 12V, but after that it needed to be properly cooled down before it should be engaged again. The higher power consumption for the solenoid and the unpredictable engaging called for a different design that would exclude the solenoid.

In addition, the problem with the speed was not solved. In order to fix this problem a chopper circuitry was introduced to increase the current going through the coils of the stepper motors. The chopper circuitry consisted of the stepper motor driver L297 and the dual H bridge L298H. However, chopper circuitry did not add any substantial difference in speed to the system.

The cost of making just one prototype B was relatively high. However, it would not cost more than $20.00 to $30.00 to produce the unit in high volumes. The difference in prices obtained by buying one unit of each component compared with buying several thousand units is quite substantial and it is therefore difficult to state the accurate price of the final product.

5.3 Conclusions from Prototype C

For prototype C a whole new concept was utilized. Prototype C is shown on page 51 in Figure 5-1c and in Attachment A.1.3 a complete parts list is available. The idea of using one motor was thrown out and this prototype was developed using two DC motors to replace the solenoid and the stepper motor. The use of DC motors solves the two main problems with the two previous designs, namely speed and reliability.

By using a DC gear motor for elevation the antenna is deployed in 20 seconds, which was what the design requirements were. When the antenna is deployed there is no pin that is supposed to hit a hole. The smaller DC gear motor that is used for the elevation predictably performs the rotation. In addition to solving the reliability problem, it conserves the energy since it does not draw nearly as much current as the solenoid. The heating problems of the two previous designs are therefore also eliminated.

5.4 Future Considerations

Prototype C is not designed to be visually pleasing but rather to be a functional prototype. In order to make the product ready for the market designers will have to streamline it and enclose it in a molded plastic box that the consumer will like. There are other features that can be added. For example, programmed memory so that the antenna adjusts itself to a memorized TV channel at the push of the channel button on the TV. Another example includes automatic TV reception adjustment where the antenna searches the horizon for TV signals and stops when one is detected.

5.5 Conclusion

Prototype C meets all the functional requirements set forth by Winegard. The problems with getting stepper motors to run fast with high enough torque made it necessary to change the design. The two first prototypes had problems both in terms of deployment time and reliability. Introducing prototype C eliminates both those problems. In addition, prototype C will require fewer electronic components; because in order to make the stepper motors run faster additional chopper circuitry and power supplies must be added. The required components prototype C are also all relatively cheap. The conclusion is therefore that in order to have a functional, cheap and reliable product the last prototype is the best choice and it should be further developed.

BIBLIOGRAPHY

1.) Kamm, L. J (1995). "Understanding Electro Mechanical Engineering." New York: The Institute of Electrical and Electronics Engineers, Inc.

2.) Fuduka & Menz (1998). "Handbook of Sensors and Actuators." Amsterdam, Netherlands: Elsevier Science B.V.

3.) Hughes, A. (1994). "Electric Motors and Drives." Manchester, United Kingdom: Newnes

4.) Acarnley, P.P. (1992). "Stepping Motors: A guide to modern theory and practice." 3rd edition, London: Peter Peregrinus Ltd

5.) Kenjo, T. (1991). "Electric Motors and their Controls." Oxford: Oxford University Press.

6.) Rogalski, A. (2000). "Infrared Detectors." Warsaw, Poland: Military University of Technology, Gordon and Breach Science Publishers.

7.) Jha, A.R. (2000). "Infrared Technology: applications to electro-optics, photonic devices, and sensors." New York: John Wiley & Sons, Inc.

8.) Craig, J. J. (1989). "Robotics." New York: Addison-Wesley Publishing Company, Inc.

9.) Moczala, Draeger, Krauβ, Schock, Tillner (1998). "Small Electric Motors." London: The Institution of Electrical Engineers.

10.) Hambley, A. R. (1997). "Electrical Engineering: Principals & Applications." Upper Saddle River, New Jersey: Prentice Hall

11.) Schlessinger, M. (1995). "Infrared Tecnology Fundamentals." 2nd edition. New York: Marcel Dekker, Inc.

12.) Histand, Alciatore (1998). "Introduction to Mechatronics and Measurement Systems." New York: WCB McGraw-Hill

13.) Jones, D. W. (2003). "Jones on Stepper Motors." Iowa City, Iowa: University of Iowa. Available on http://www.cs.uiowa.edu/~jones/step/

14.) Floyd, T. L. (2002). "Electronic Devices." 6th edition. Upper Saddle River, New Jersey: Prentice Hall

15.) Gardner, Siegesmund (1998). "PIC C, An introduction to programming the Microchip PIC in CCS C." United Kingdom: Character Press Limited

ACKNOWLEDGEMENTS

I wish to express thanks to my major professor, Dr. Greg Luecke for accepting me as a research assistant and allowing me to create this research project. He has been of great help and inspiration. I also thank Dr. Rudolphi and Dr. Flugrad for serving on my Program of Study committee. Mark Wong and I worked together on the early stages of the project; he contributed to some of the ideas and solutions, and I thank him. I thank my wife, Line for her help and support and my son Christian for being a great inspiration.

APPENDIX A – Mechanical Parts List

A.1 Mechanical Part List Prototype A

Item	Part No.	Quantity	Name	Material	Source
1	Custom	1	Frame	Aluminum	Cent. Store
2	A7Y55PSS5025	1	Ball Bearing	St. Steel	SDP/SI
3	57BYG085	1	Stepper Motor	N/A	Jameco
4	A1M6Y24050	1	Worm Gear	Nylon	SDP/SI
5	A1M5Y24	1	Worm	Nylon	SDP/SI
6	Custom	1	Mtr. Shft Cpl	Aluminum	Cent. Store
7	Custom	1	Motor Shaft	Steel	Lab
8	Custom	2	Shaft Support	Cast Iron	Lab
9	S9912Y-UBM-1F	2	Bearings	St. Steel	SDP/SI
10	Antenna Kit	1	Antenna Kit	N/A	Winegard
11	Custom	1	Motor Bracket	Steel	Cent. Store
12	Assorted bolts	14	Bolts	Steel	Lowes
13	Assorted Nuts	4	Nuts	Steel	Lowes

A.2 Mechanical Part List Prototype B

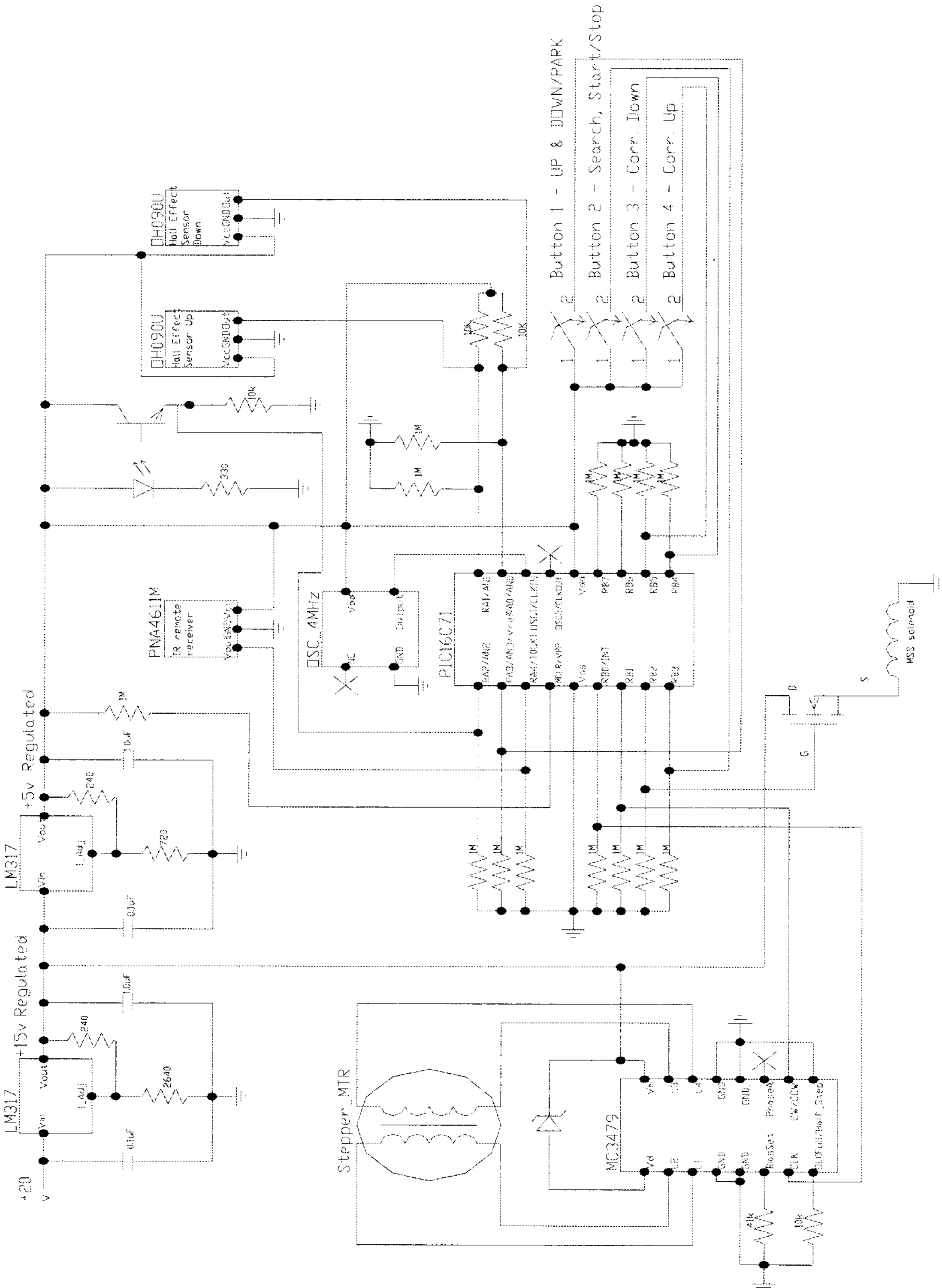
Item	Part No.	Quantity	Name	Material	Source
1	Custom	1	Left Enclosing	Plastic	CIRAS/ISU
2	Custom	1	Right Enclosing	Plastic	CIRAS/ISU
3	Custom	1	Solenoid Encoder Plate	Plastic	CIRAS/ISU
4	Custom	1	Worm gear retrofit	Plastic	CIRAS/ISU
5	57BYG085	1	Stepper Motor	N/A	Jameco
6	A7X108050	1	Motor Shaft	St. Steel	SDP/SI
7	A7Y55PSS5025	1	Ball Bearing	St. Steel	SDP/SI
8	Custom	1	Mtr.Shft Coupl.	Aluminum	Cent. Store
9	A1M6Y24050	1	Worm Gear	Nylon	SDP/SI
10	A1M5Y24	1	Worm	Nylon	SDP/SI
11	GSRG040203	1	Keypad	Plastic	Storm
12	S-18-100	1	Solenoid	N/A	MSS
13	Assorted Hardware		Blts, Nts, Wsh	St. Steel	Lowes

A.3 Mechanical Part List Prototype C

Item	Part No.	Quantity	Name	Material	Source
1	Custom	1	Top Frame	Aluminum	Cust. Steel
2	Custom	1	Bottom Frame	Aluminum	Cust. Steel
3		6	Adj. Spacer	Steel	ACE
4		6	Bolts	Steel	ACE
5		6	Nuts	Steel	ACE
6	Custom	1	Sml Mtr. Brack.	Aluminum	ISU_IE
7	Custom	1	Big Mtr. Brack.	Aluminum	Cust. Steel
8		8	Mtr. Brckt Bolts	Steel	ACE
9		8	Mtr. Brckt Nuts	Steel	ACE
10	455A679	1	Gear Motor		Globe/Wineg
11		1	Gear Motor		Jameco
12	A1M4Y24024	2	Miter Gear	Nylon	SDP/SI
13	A1N2N24084	1	Spur Gear	Nylon	SDP/SI
14	A1N2N24012	1	Spur Gear	Nylon	SDP/SI
15	Custom	2	Al Bracket	Aluminum	ISU_IE
16	Custom	2	Angle Bracket	Aluminum	ISU_ME
17	Custom	1	Spur connector	Aluminum	ISU_IE

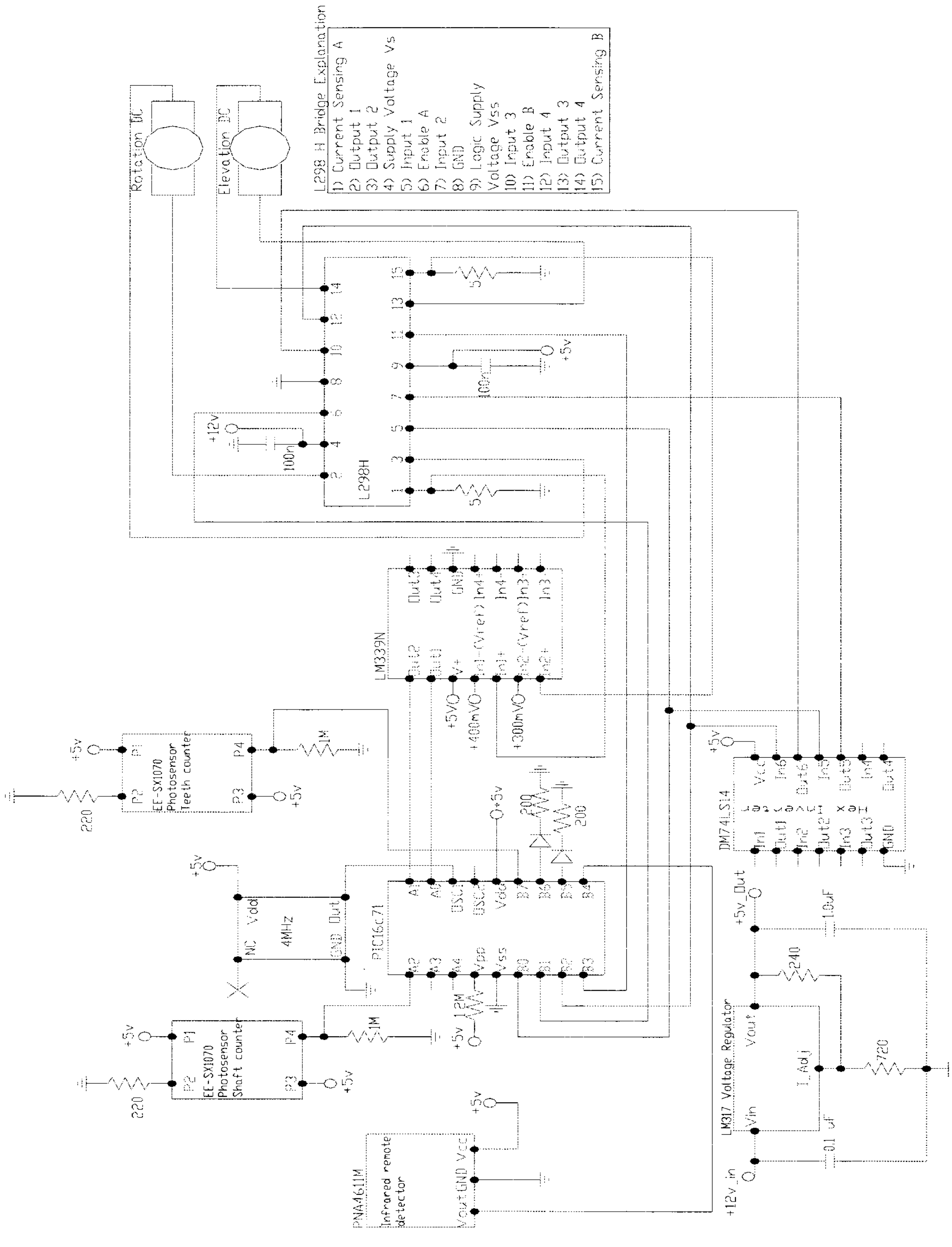
Appendix B – Electrical Schematics and Parts List

B.1 Electrical schematic and part list for Prototype A and B



Item	Part No.	Quantity	Name	Source
1	MC3479	1	Stpr mtr driver	Motorola
2	PIC16C71	1	Micro Controller	Microchip
3	X107-ND	1	4MHz Crystal	ECS inc.
4	OHB090U	2	Hall Effect	Optek
5	PN150L	1	Photo diode	Panasonic
6	LN54PA-ND	1	Photo Detector	Panasonic
7	LM317	1	Var. volt regula.	Digikey
8	Resistrs., wires etc.			

B.2 Electrical schematic and part list for Prototype C



Item	Part No.	Quantity	Name	Source
1	PIC16C71	1	Micro controller	Microchip
2	LM339A	1	Comparator	Digikey
3	DM74LS14	1	Hex inverter	Digikey
4	EE-SX1070	2	Photo Sensor	Omron
5	L298H	1	H bridge	ST electro.
6	PNA4612M	1	Infrrd rmte snsr	Panasonic
7	P302-ND	1	Green LED	Digikey
8	P301-ND	1	Red LED	Digikey
9	Resistrs, wires, etc.			

Appendix C – Controller Programming

C.1 Controller program code Prototype A

```
//*************************************
//*****revision: 03/20/2002************
//********Lars Kaldestad***************
//*************************************

//Use of Pins:
// A0 = Hall Effect Sensor Up
// A1 = Hall Effect Sensor Down
// A2 = LED photosensor center
// A3 = UP/DOWN Button
// A4 = Nothing
// B0 = Puls to stepper motor chip
// B1 = Direction of Stepper Motor
// B2 = Solenoid
// B3 = Search Button
// B4 = LED Debug
// B5 = LED Debug
// B6 = LED Debug
// B7 = LED Debug

#include <16C71.H>
#fuses   XT,NOWDT,NOPROTECT
#use delay(clock=1000000)

int position=0;
int count_rev = 0;
int count_rev_max = 12;
int count_turn_max = 5;
int count_turn = 0;
short int cw = 0;
long int i = 0;
long int countmax = 32000;
long int count_step = 0;

//**************************************MOVE UP*************************************
void move_up (void)
{
   output_low(PIN_B2);      //Solenoid retracts
   output_high(PIN_B1);     //Set direction of stepper motor set to up
   output_high(PIN_B0);     //Start Pulsing on B0, sends to stepper chip
   delay_us(1350);          //Speed of Pulsing
   output_low(PIN_B0);
   delay_us(1350);
   output_high(PIN_B0);
   delay_us(1350);
   output_low(PIN_B0);
   delay_us(1350);
}
//**************************************MOVE DOWN***********************************
void move_down (void)
{
   output_low(PIN_B2);      //Solenoid retracts
   output_low(PIN_B1);      //Set direction of stepper motor set to up
   output_high(PIN_B0);     //Start Pulsing on B0, sends to stepper chip
   delay_us(800);           //Speed of Pulsing
   output_low(PIN_B0);
   delay_us(800);
   output_high(PIN_B0);
   delay_us(800);
    output_low(PIN_B0);
   delay_us(800);
}
//**************************************TURN CCW************************************
void turn_ccw (void)
{
```

```
   output_high(PIN_B2);     //Solenoid extends to lock the two shafts together
   output_high(PIN_B1);     //Set direction of stepper motor set to move antenna ccw
   output_high(PIN_B0);     //Start Pulsing on B0, sends to stepper chip
   delay_us(4000);          //Speed of Pulsing
   output_low(PIN_B0);
   delay_us(4000);
}
//***********************************TURN CW************************************
void turn_cw (void)
{
   output_high(PIN_B2);     //Solenoid extends to lock the two shafts together
   output_low(PIN_B1);      //Set direction of stepper motor set to move antenna clockwise
   output_high(PIN_B0);     //Start Pulsing on B0, sends to stepper chip
   delay_us(4000);
   output_low(PIN_B0);
   delay_us(4000);
}
//************************************STATES*************************************
void UPUP(void)
{
   output_high(PIN_B7);
   position = 4;
}
void UPRIGHT(void)
{
   output_high(PIN_B6);
   position = 3;
}
void DOWN(void)
{
   output_high(PIN_B5);
   position = 2;
}
void NEITHER(void)
{
   output_high(PIN_B4);
   position = 1;
}
void state(void)
{
   output_low(PIN_B4);
   output_low(PIN_B5);
   output_low(PIN_B6);
   output_low(PIN_B7);
   if (!input(PIN_A0))           //Hall Effect UP
   {
      if (input(PIN_A2))         //LED photosensor
      {
         UPUP();
      }
      else if (!input(PIN_A2))   //LED photsensor
      {
         UPRIGHT();
      }
   }
   else if (input(PIN_A0))       //Hall Effect UP
   {
      if (!input(PIN_A1))        //Hall Effect Down
      {
         DOWN();
      }
      else if (input(PIN_A1))    //Hall Effect Down
      {
         NEITHER();
      }
   }

}
//************************************FINDCENTER********************************
```

```
void findCenter(void)
{
   if (input(PIN_A2))
   {
      do
      {
         turn_ccw();
         cw=0;
      }while(input(PIN_A2) && input(PIN_B3));
   }
   else if (!input(PIN_A2))
   {
      do
      {
         turn_cw();
         cw=1;
      }while(!input(PIN_A2) && input(PIN_B3));
   }
}

void findCenterDown(void)
{
   if (input(PIN_A2))
   {
      do
      {
         turn_ccw();
         cw=0;
      }while(input(PIN_A2));
   }
   else if (!input(PIN_A2))
   {
      do
      {
         turn_cw();
         cw=1;
      }while(!input(PIN_A2));
   }
}
//********************************RELEASESOLENOID**********************************
void releaseSolenoid(void)
{  i=0;
   if (cw==0)
   {
      do
      {
         turn_cw();
         i=i+1;
      }while(i<10);
   }
   else if (cw==1)
   {
      do
      {
         turn_cw();
         i=i+1;
      }while(i<10);
   }
}
//*************************************SEARCH*************************************
void search()
{  i=0;
   findCenter();
   do
   {
      if (cw==0)
      {
         do
         {
```

```
            turn_ccw();
               if (i==0)
                  {delay_ms(1000);}       //Delay to give the solenoid time to move down
            i=i+1;
         }while(i<1500 && input(PIN_B3));
         findCenter();
         i=0;
      }
      else if (cw == 1)
      {
         do
         {
            turn_cw();
               if (i==0)
                  {delay_ms(1000);}       //Delay to give the solenoid time to move down
            i=i+1;
         }while(i<1500 && input(PIN_B3));
         findCenter();
         i=0;
      }
   }while(input(PIN_B3));
}
//*******************************FROM STATE DOWN TO UP*******************************
void fromStateDownToUp (void)
{
         count_rev = 0;
         count_step = 0;
               do
               {
                  move_up();
                  count_step=count_step+1;
                  if (!input(PIN_A0))//Up Hall Effect Sensor
                  {
                     count_rev=count_rev+1;
                     do
                     {
                        move_up();
                        count_step=count_step+1;
                     }while(!input(PIN_A0) && count_rev<count_rev_max);
                  }
               }while (count_rev<count_rev_max && count_step<countmax);
         count_rev = 0;
         count_step = 0;
}
//*******************************FROM STATE UP TO DOWN*******************************
void fromStateUpToDown (void)
{
   findCenterDown();
   releaseSolenoid();
   delay_ms(1000);
   count_rev=0;
   count_step=0;
   do
   {
      move_down();
      count_step=count_step+1;
      if (!input(PIN_A1))
         {
            count_rev=count_rev+1;
            do
            {
               move_down();
               count_step=count_step+1;
            }while(!input(PIN_A1) && count_rev<count_rev_max);
         }
    }while (count_rev<count_rev_max && count_step<countmax);
   count_rev=0;
   count_step=0;
}
```

```
//******************************FROM STATE LOST to UP******************************
void fromStateLostToFindThenDown(void)
{
        count_step = 0;
        do
        {
              move_up();
              count_step=count_step+1;
        }while (count_step<countmax && input(PIN_A2));
        output_low(PIN_B2);
        output_high(PIN_B2);
        i=0;
        for(i=0;i<100;i++)
              move_down();
              findCenterDown();
              releaseSolenoid();
              fromStateUpToDown();
              count_rev=9;
        do
        {
           move_down();
           count_step=count_step+1;
           if (!input(PIN_A1))
           {
              count_rev=count_rev+1;
              do
              {
                 move_down();
                 count_step=count_step+1;
              }while(!input(PIN_A1) && count_rev<count_rev_max);
           }
        }while (count_rev<count_rev_max && count_step<countmax);
        count_rev=0;
        count_step=0;
}
//*************************************MAIN*****************************************
main()
{
long int i;
   for(;;)
   {
      i = 0;
      count_rev = 0;
      count_rev_max = 12;
      count_step = 0;
      countmax = 35000;
      count_turn = 0;
      state();
      output_low(PIN_B2);  //Set solenoid to low to save energy consumption when at rest
      output_low(PIN_B4);
      output_low(PIN_B5);
      output_low(PIN_B6);
      output_low(PIN_B7);
//Check to see if user wants to move up
      if (input(PIN_A3) && position == 2)      // checking move up / move down button && 2 is
down
      {
         fromStateDownToUp();
         releaseSolenoid();
      }
//Check to see if user wants to search
      else if(input(PIN_B3) && position > 2)    //Check search button && greater than 2 (up)
      {
         search();
         releaseSolenoid();
      }
//Check to see if user wants to park the antenna
      else if(input(PIN_A3) && position > 2)    //checking move up / move down button
      {
```

```
         fromStateUpToDown();
      }
//Lost!! Check to see where we are
          else if(input(PIN_A3) && position == 1)
          {
                fromStateLostToFindThenDown();
          }
    } //End of for(;;)
} //End of Main
```

C.2 Controller program code Prototype B

```
//*******************************************************
//**rev:07/10/2002 Winegard visit removed speeds for up**
//************** rev:06/07/2002 added IR *****************
//************** revision: 05/30/2002 ********************
//************** revision: 05/21/2002 ********************
//************** revision: 05/14/2002 ********************
//**************  Lars R. Kaldestad   ********************
//*******************************************************

// Use of Pins:
// A0 = Hall Effect Sensor Up
// A1 = Hall Effect Sensor Down
// A2 = LED photosensor center
// A3 = UP/DOWN Button
// A4 = IR input pin
// B0 = Puls to stepper motor chip
// B1 = Direction of Stepper Motor
// B2 = Solenoid
// B3 = Search Button         //B3 and B4 combined can later be manual search
// B4 = Correction Button Down
// B5 = Correction Button Up
// B6 = Nothing //debug output
// B7 = Nothing //debug output

#include <16c71.h>
#list
#fuses XT,NOWDT,NOPROTECT
#use delay(clock=4000000)
#define START_MIN 120    /* 16us X 120 = 2.0ms min start bit */
#define START_MAX 260    /* 16us X 260 = 4.2ms max start bit */
#define BITTIME 60       /* 16us X 63 = 1ms */

int position = 0;
//int count_rev_max = 12;
int cw=0;
long int i = 0;

//<<<<<<<<<<<<< IR STATEMENTS START >>>>>>>>>>>>>>>>>>>>>>>>>>>>>
byte time, n;
byte value;
void wait_for_high_to_low()
{
   while(!input(PIN_A4));
   delay_us(3);
   while(input(PIN_A4));
}
void wait_for_high()
{
   delay_us(3);
   while(!input(PIN_A4));
}
void infrared()
{
    value = 0;
//    if(!input(PIN_A4)&&(value!=128||value!=129||value!=130||value!=131))
```

```
    if(!input(PIN_A4))
      {
      do {                                    /* start bit  */
         wait_for_high_to_low();                      /* must be between 2-3ms */
         set_rtcc(0);
         wait_for_high();
         time = get_rtcc();
      } while ((time < START_MIN) || (time > START_MAX));
      for (n = 0 ; n <= 7 ; n++){               /*get data bits */
         wait_for_high_to_low();
         set_rtcc(0);
         wait_for_high();
         time = get_rtcc();
         if (time > BITTIME)
            bit_set(value,n);
      }                          /*Get data bit 8        *
     }
}
//<<<<<<<<<< IR STATEMENT/FUNCTIONS END >>>>>>>>>>>>>>>>>>>>>>>>>>>
void releaseSolenoid(void);
void findCenter(void);
void findCenterDown(void);
void moveOneRevUp(void);
void moveOneRevDown(void);
//******************************* MOVE UP ********************************
void move_up (void)
{
   output_low(PIN_B2);
      output_high(PIN_B1);    // Set direction of stepper motor to move up
      output_high(PIN_B0);    // Start pulsing on B0, sends to stepper chip
      delay_us(775);         // speed of pulsing
      output_low(PIN_B0);
      delay_us(775);
      output_high(PIN_B0);
      delay_us(775);
      output_low(PIN_B0);
      delay_us(775);
}
void move_down (void)
{
   output_low(PIN_B2);
   output_low(PIN_B1);     // Set direction of stepper motor to move up
   output_high(PIN_B0);     // Start pulsing on B0, sends to stepper chip
   delay_us(675);          // speed of pulsing
   output_low(PIN_B0);
   delay_us(675);
   output_high(PIN_B0);
   delay_us(675);
   output_low(PIN_B0);
   delay_us(675);
}
void turn_ccw (void)
{
   output_high(PIN_B2); //hold output on solenoid pin to high to keep the solenoid engaged
   output_high(PIN_B1); //set direction of stepper motor to move ccw
   output_high(PIN_B0);
   delay_us(3200);
   output_low(PIN_B0);
   delay_us(3200);
   output_high(PIN_B0);
   delay_us(3200);
   output_low(PIN_B0);
   delay_us(3200);
}
void turn_cw (void)
{
   output_high(PIN_B2); //engage solenoid
   output_low(PIN_B1);  //set direction of stepper motor to move cw
   output_high(PIN_B0);
```

```
   delay_us(3200);
   output_low(PIN_B0);
   delay_us(3200);
   output_high(PIN_B0);
   delay_us(3200);
   output_low(PIN_B0);
   delay_us(3200);
}
void release_ccw(void)
{
   output_low(PIN_B2); //hold output on solenoid pin to high to keep the solenoid engaged
   output_high(PIN_B1); //set direction of stepper motor to move ccw
   output_high(PIN_B0);
   delay_us(1500);
   output_low(PIN_B0);
   delay_us(1500);
   output_high(PIN_B0);
   delay_us(1500);
   output_low(PIN_B0);
   delay_us(1500);
}
void release_cw(void)
{
   output_low(PIN_B2); //engage solenoid
   output_low(PIN_B1);  //set direction of stepper motor to move cw
   output_high(PIN_B0);
   delay_us(1500);
   output_low(PIN_B0);
   delay_us(1500);
   output_high(PIN_B0);
   delay_us(1500);
   output_low(PIN_B0);
   delay_us(1500);
}
//********************************* STATES *********************************
void UPUP(void)
{
   position = 4;
}
void DOWN (void)
{
   position = 2;
}

/*void NEITHER (void)
{
   position=1;
}*/
void state (void)
{
   if (!input(PIN_A0))
   {
//       if (input(PIN_A2))
    //   {
          UPUP();
  //     }
      //else if (!input(PIN_A2))
      //{
         //UPRIGHT();
        // UPUP();
      //}
   }
   else if(input(PIN_A0))
   {
     // if (!input(PIN_A1))
      //{
         DOWN();
      //}
/*       else if(input(PIN_A1))
```

```
        {
           NEITHER();
        }
*/
   }
}
//********************** FROM STATE DOWN TO UP ***************************
void fromStateDownToUp(void)
{
   i = 0;
   //count_step = 0;
   do
   {
      move_up();
      //count_step=count_step+1;
      if(!input(PIN_A0))    //Up Hall Effect Sensor
      {
         i=i+1;
         do
         {
            move_up();
            //count_step=count_step+1;
         }while(!input(PIN_A0) && i<13);
      }

    }while(i<13);
 output_high(PIN_B2);
 delay_ms(1000);
 i=0;
 do
 {
   turn_ccw();
   i=i+1;
 }while(i<285);
i=0;
 //  count_step=0;
}
//******************************* SEARCH *********************************
void search (void)
{
   i=0;
   value=100;
   delay_ms(500);
   findCenter();
   do
   {
      if (cw == 0)
      {
         do
         {
            turn_ccw();
             // if (i==0){delay_ms(1000);}
            i=i+1;
            if (value!=130)
               infrared();
         //}while((!input(PIN_B3)||(value!=130))&&(i<600));
         }while((value!=130)&&(i<650));
         findCenter();
         i=0;
      }
      else if (cw == 1)
      {
         do
         {
            turn_cw();
            //if (i==0) {delay_ms(1000);}
            i=i+1;
            if (value!=130)
               infrared();
```

```
          //}while((!input(PIN_B3)||(value!=130))&&(i<600));
          }while((value!=130)&&(i<650));
          findCenter();
          i=0;
       }
//    }while(!input(PIN_B3)||(value!=130));
   }while(value!=130);
  // do//This is put here to prevent the search from quitting while pin b3 is still high
which will send it into a
   //{// new search right away.
   // i=0;
   //}while(input(PIN_B3));
//   value = 100;
}
//************************ FROM STATE UP TO DOWN ************************
void fromStateUpToDown (void)
{
   findCenterDown();
   releaseSolenoid();
   i=0;
  // count_step=0;
   do
   {
      move_down();
//       count_step = count_step+1;
      if (!input(PIN_A1))
      {
         i=i+1;
         do
         {
            move_down();
          //  count_step=count_step+1;
         }while(!input(PIN_A1) && i<13);
      }
   }while(i<13);
   i=0;
//   count_step=0;
}
//************************** RELEASE SOLENOID ***************************
void releaseSolenoid(void)
{
   i=0;

output_low(PIN_B2);
   if (cw==0)
   {
      do
      {
         release_cw();//same direction as cw
         i=i+1;
      }while(i<80);
   delay_ms(500);
   }
   else if (cw == 1)
   {
      do
      {
         release_ccw();//same direction as ccw
         i=i+1;
      }while(i<80);
   delay_ms(1000);
   }
   i=0;
}
//**************************** FIND CENTER *****************************
void findCenter(void)
{
 if (input(PIN_A2))
 {
```

```
   do
   {
      turn_cw();
      cw=1;
      if (value!=130)
         infrared();
//   }while((!input(PIN_B3)||(value!=130))&&(input(PIN_A2)));
   }while((value!=130)&&(input(PIN_A2)));
 }
 else if (!input(PIN_A2))
 {
   do
   {
      turn_ccw();
      cw=0;
      if (value!=130)
         infrared();
//   }while((!input(PIN_B3)||(value!=130))&&(!input(PIN_A2)));
   }while((value!=130)&&(!input(PIN_A2)));
 }
 //value=100;
}
void findCenterDown(void)
{
   if (input(PIN_A2))
   {
      do
      {
         turn_cw();
         cw=1;
      }while(input(PIN_A2));
   }
   else if (!input(PIN_A2))
   {
      do
      {
         turn_ccw();
         cw=0;
      }while(!input(PIN_A2));
   }
}
void moveOneRevDown(void)
{
   if (!input(PIN_A1))
   {
      do
      {
          move_down();
      } while(!input(PIN_A1));
   }
   do
   {
      move_down();
   }while(input(PIN_A1));
}
void moveOneRevUp(void)
{
   if(!input(PIN_A1))
   {
      do
      {
         move_up();
      }while(!input(PIN_A1));
   }
   do
   {
      move_up();
   }while(input(PIN_A1));
}
```

```
//******************************* MAIN ********************************
main()
{
//long int i;
setup_counters(RTCC_INTERNAL, RTCC_DIV_16);
   for(;;)
   {
      i=0;
      output_low(PIN_B2);
      state();
      infrared();
      if ((input(PIN_A3)||(value == 131)) && position == 2)
      {
        fromStateDownToUp();
         search();
         delay_ms(500);
         //releaseSolenoid();
      }
      else if ((input(PIN_B3)||(value == 130)) && position > 2)
      {
         search();
         delay_ms(500);
         //releaseSolenoid();
      }
      else if ((input(PIN_A3)||(value == 131)) && position > 2)
      {
         fromStateUpToDown();
      }
      else if (input(PIN_B5)||(value == 129))
      {//This is also needed to adjust the parking position
         moveOneRevUp();
      }
      else if (input(PIN_B4)||(value == 128))
      {//This is needed to adjust the parking position
         moveOneRevDown();
      }
     value = 100;
   }//End of for(;;)
} //End of Main
```

```
//*******************************************************
//************* revision: 11/22/2002 *******************
//*************  Lars R. Kaldestad   *******************
//*******************************************************

// Use of Pins:
// A0 = Current Sense A      A2 |^^^*^^^| A1
// A1 = Current Sense B      A3 |   P   | A0
// A2 = Optical Shaft Ctr    A4 |   I   | OSC1
// A3 = x                    Vpp|   C   | OSC2
// A4 = x                    Vss|   1   | Vdd
// B0 = Enable A             B0 |   6   | B7
// B1 = Enable B             B1 |   c   | B6
// B2 = Direction A          B2 |   7   | B5
// B3 = Direction B          B3 |___1___| B4
// B4 = Infrared
// B5 = diode green
// B6 = diode red
// B7 = Optical Teeth Counter
//
//
//*******************************************************
//******** Infrared Remote Control Button Values ********
//******** 1 = 128 , 2 = 129, 3 = 130, 4 = 131, 5 = 132


#include <16C71.H>
#list
//#fuses   PAR,NOPROTECT,NOWDT,INTRC,MCLR   /*for internal clock */
//#fuses   PAR,NOPROTECT,NOWDT,XT,MCLR    /* for external crystal/resonat*/
#fuses XT,NOWDT,NOPROTECT
#use delay(clock=4000000)
//#use rs232(baud=2400, xmit=PIN_B0)
//#rom 0x03ff={0x0c40}           /*w factor for EPROM devices*/
#define START_MIN 100 /* 16us X 120 = 2.0ms min start bit */
#define START_MAX 260 /* 16us X 260 = 4.2ms max start bit */
#define BITTIME 60    /* 16us X 63  = 1ms   700us = 0  1200us = 1  */

//byte const valid_code[22] =
// {128,129,130,131,132,133,134,135,136,137,
// 146,147,144,145,88,89,92,91,90,106,148,187};
byte time, n;
byte prev_value, value;
int dir = 0;
int counter=100;
int prev_hi_lo=0;
int state = 0;
int prev_state=0;

void wait_for_high_to_low() {
   while(!input(PIN_B4)) ;      /* if it's high, wait for a low */
   delay_us(3);                 /* account for fall time */
   while(input(PIN_B4));        /* wait for signal to go high */
}

void wait_for_high() {
   delay_us(3);                 /* account for rise time */
   while(!input(PIN_B4));       /* wait for signal to go high */
}

void infrared()
{
     value = 0;

  if (!input(PIN_B4)){
     do {                                 /* start bit  */
        wait_for_high_to_low();                  /* must be between 2-3ms */
        set_rtcc(0);
        wait_for_high();
```

```
        time = get_rtcc();
      } while ((time < START_MIN) || (time > START_MAX));

      for (n = 0 ; n <= 7 ; n++){              /*get data bits */
        wait_for_high_to_low();
        set_rtcc(0);
        wait_for_high();
        time = get_rtcc();
        if (time > BITTIME)
           bit_set(value,n);
      }                            /*Get data bit 8        */
      bit_clear(value,6);          /* This code removes the value for bit 6*/
      wait_for_high_to_low();       /* and uses the value of bit 8 instead */
      set_rtcc(0);                  /* bits 6,9,10, and 11 are ignored     */
      wait_for_high();
      time = get_rtcc();
      if (time > BITTIME)
           bit_set(value,6);
  }
}

void collision()
{int deploy=0;
         do
         {
            if (!input(PIN_A2))
            {
               deploy=deploy+1;
               do
               {
                  output_high(PIN_B5);
                  output_high(PIN_B6);
                  delay_ms(200);
               }while(!input(PIN_A2));

                  output_low(PIN_B5);
                  output_low(PIN_B6);
            }
         }while(input(PIN_A0));

         if (deploy<12)
         {
            output_high(PIN_B5);
            output_high(PIN_B6);
            output_low(PIN_B2);
            dir=2;
            delay_ms(1000);
         }
         else
         {
            dir=1;
         }

         state=3;
}

void parked()
{
   output_high(PIN_B5); //Green indicator light on
   output_low(PIN_B6);
   do
   {
      infrared();
      if(value == 128)
      {
         output_high(PIN_B0);
         output_high(PIN_B2);
         delay_ms(2000);
         collision();
```

```
      }

   }while (state==1);
   prev_state=1;
}
void home()
{   output_high(PIN_B6);
   do
   {
      output_high(PIN_B1);     //homing
      output_high(PIN_B3);
      delay_ms(1000);
   }while(input(PIN_A1));
   output_low(PIN_B1);
   counter=0;
   dir=0;
}
void find_center()
{
//signed int ctr;
   home();
   output_low(PIN_B3);
   output_high(PIN_B1);

   delay_ms(400);
//   ctr=40;

   for (n = 0 ; n <= 73 ; n++)
   {

      if(!input(PIN_B7))
      {
            do
            {
               output_low(PIN_B5);
            }while (!input(PIN_B7));
      delay_ms(50);
         output_high(PIN_B5);
      }
      else if(input(PIN_B7))
      {
         do
         {

         }while(input(PIN_B7));
      delay_ms(50);
      }


   }

   state=3;
   output_high(PIN_B3);
   delay_ms(100);
   output_low(PIN_B1);

   output_high(PIN_B0);
   output_low(PIN_B2);
   dir=2;
   delay_ms(400);
}

void deployed()
{
int passes = 0;
output_high(PIN_B6);
if (prev_state==1)
{
   output_high(PIN_B1);
```

```
      output_low(PIN_B3);
      delay_ms(400);
      dir=4;
      prev_state=0;
   }


      do{
         infrared();
         if(value==130)
         {
            if (dir==3)
            {
               output_low(PIN_B3);
               delay_ms(100);
            }
            else if (dir==4)
            {
               output_high(PIN_B3);
               delay_ms(100);
            }
          output_low(PIN_B1);
          dir=0;
         }
         else if(value==131)
         {  passes=0;
            output_high(PIN_B1);
            output_high(PIN_B3);
            delay_ms(400);
            dir=3;
         }
         else if(value==132)
         {  passes=0;
            output_high(PIN_B1);
            output_low(PIN_B3);
            delay_ms(400);
            dir=4;
         }
         else if (value==133)
         {
            state=4;
         }
         else if(!input(PIN_A1))
         {
            passes=passes+1;
            if (passes==5)
            {
               output_low(PIN_B1);
            }
            else
            {
               if(dir==3)
               {
                  output_low(PIN_B3);
                  dir=4;
               }
               else if(dir==4)
               {
                  output_high(PIN_B3);
                  dir=3;
               }
               delay_ms(400);
            }
         }
         if(!input(PIN_B7))
         {
            output_low(PIN_B5);
         }
         else
```

```
        output_high(PIN_B5);

    }while(state==2);

}

void middle()
{
    output_low(PIN_B5);
    output_high(PIN_B6);
    do
    {
        infrared();
        if(value==130)
        {
            output_low(PIN_B0);//stop movement
            dir=0;
        }
        else if (value==128)
        {
            output_high(PIN_B0);
            output_high(PIN_B2);
            delay_ms(1000);
//            dir=1;
            collision();

        }
        else if (value==129)
        {
            output_high(PIN_B0);
            output_low(PIN_B2);
            delay_ms(2000);
            dir=2;
        }
        else if (!input(PIN_A0))
        {

            if(dir==1)
            {
                output_low(PIN_B2);
                delay_ms(1200);
                output_low(PIN_B0);
                state=2;
            }
            else if(dir==2)
            {
                output_low(PIN_B0);
                state=1;
            }
        }
        if (!input(PIN_A2))
        {
            output_high(PIN_B5);
            output_high(PIN_B6);
        }
        else
        {
            output_low(PIN_B5);
            output_low(PIN_B6);
        }
    }while(state==3);
}

void find_location()
{
    output_high(PIN_B0);
    output_low(PIN_B2);      //move down first
    delay_ms(1000);
    if(!input(PIN_A0))
```

```
    {
       state=1;
    }
    else
    {
       output_high(PIN_B2);     //move up 3 sec. second
       delay_ms(3000);
       if(!input(PIN_A0))
       {
          state=2;
          output_low(PIN_B2);  //relieve the strain by moving down
          delay_ms(1000);
       }
       else
       {
          state=3;
       }
    }
    output_low(PIN_B0);          //Cut Power to elevation motor
}

main()
{

    setup_counters( RTCC_INTERNAL, RTCC_DIV_16);
    for(;;)

    {
       if (state==0)
       {
          find_location();
       }
       else if(state==1)
       {
          parked();
       }
       else if(state==2)
       {
          deployed();
       }
       else if(state==3)
       {
          middle();
       }
       else if (state==4)
       {
          find_center();
       }

    }
}
```