Go Go Golfing Robot

Introduction to Design
EE 382
Dr. Rison and Dr. Wedeward

Jennifer Coleman
Adam Moritz
Jonathan Minor
5/6/2002

Table of Contents

List of Figures

Abstract

The purpose of this exercise described in the following pages was to learn methods of robotic design. The assignment was to build an autonomous robot capable of locating a golf ball, moving it to the hole, and dropping it in the hole. The tasks required to design such a robot were too much for one small group to accomplish in 3.5 months, so the responsibility was divided into 4 parts: Navigation, Ball-Hole Location, Communication, and Chassis.

The purpose of the Chassis Group was to design a "smart" chassis. This chassis was to be capable of receiving direction and distance commands from the Navigation Subsystem and interpret them into a distance and an angle from the current position of the robot. The chassis was to be able to move the allotted distance accurately and then notify Navigation that the command had been implemented. Once the ball had been located, the chassis was to obtain the ball and move it to the hole. The Chassis Group was also responsible for supplying power to the rest of the groups.

On the day the project was due, the H-bridges burnt out, so we weren't able to show off a finished product. While not totally debugged, the code was functional. The ball capture mechanism worked perfectly, doing exactly as we told it to. We are confident that had we been given more time, we would have been able to provide a fully functional chassis.

Introduction

The duties of building the robot were divided between 4 groups. The first group was responsible for the navigational controls of the robot. This subsystem would receive all information from the other subsystems and use it to calculate the next move. The second group was responsible for the external communications, ie, the user friendly part of the robot. The person playing golf through the robot needs an interface so he can tell the robot the GPS coordinates of the ball and hole, and from which the user can observe the robot's movements.

The third group was responsible for the ball and hole locating subsystem. This system needed to be capable of locating the ball and the hole when the limits of the GPS system had been reached. The information gathered by this system should give direction and distance coordinates so the robot can actually obtain the ball in a shorter amount of time.

The final group was responsible for the "smart" chassis. This chassis should be able to receive commands from navigation telling the robot to move a particular distance at a certain angle from it's current position and execute these commands. This chassis should also be able to capture the ball and move it from it's resting position to the hole.

The final goal of the project is to integrate all the subsystems into a moving robot.

Requirements for the Chassis Subsystem

1. Autonomous
2. Capable of receiving and responding to directional and distance commands
3. Code must successfully interact with other groups' code
4. Must provide regulated power for all groups, including chassis
5. Able to obtain the ball
6. Reliable polarized connectors
7. Spend no more than $2000(for the full robot, $500 per group)

Purpose

The purpose of this paper is to educate possible clients to the specifications on this design, allowing an appropriately educated engineer to rebuild the entire robot. The necessary requirements for understanding include an intimate knowledge of the Motorola 68HC12 Microcontroller, Linear, Analog, and Digital Systems, and proficiency in C programming,

Communications Structure

The navigation group was responsible for sending the chassis commands to be implemented. The navigation group did this through the serial communications on the HC12 board. When data is ready to be sent, the navigation group raises the handshaking line on their HC12. In the program, the chassis waits until the handshaking line has been raised. When the handshaking line is raised, the chassis prepares to receive the data. After receiving all of the data, an acknowledgement is sent back to the Navigation group.

The first byte of data received for any data transfer is the header byte. The header byte tells the chassis which group is in control of the robot's movements. A header byte of 0x24 implies that Navigation is in control, whereas a header byte of 0x25 implies Ball-Hole is in control. When the Navigation group is in control, the Chassis group is sent data in four byte packets. The first byte is the header file, while the second and third bytes are the direction bytes. The heading is sent with the most significant byte first. The fourth byte is the distance. The direction is given in degrees to the right. The distance is given in increments of 1/7 of an inch.

When the Ball-Hole group is in control, the Chassis group is sent two bytes. The first byte is the header byte. The second byte is the data byte. The data byte can correspond to a number of instructions. The data byte is structured in such a way as to minimize the amount of data sent. The most significant two bits, the description bits, of the data byte determine the type of command that is being sent. When the description bits are 00, the chassis is being ordered to stop. A heading is being sent when the description bits are 01, a speed to travel is being sent when the description bits are 10, and a command is being sent when the description bits are 11. Whenever a distance or heading is sent the third bit determines whether or not the data is positive or negative. A negative speed corresponds to traveling backward, whereas a negative heading corresponds to turning left. The last five bits of the data byte corresponds to the magnitude of the speed or heading requested. The heading given by ball/hole is in degrees to turn right and the speed is given in increments of 0.4% duty cycle.

For a command, there are two possible commands that can be requested. When the data byte is a 0xFF, Ball-Hole requests that the chassis relocate. This command is used in the case that Ball-Hole has lost the ball and needs to rescan for it. When the data byte is a 0xF0, Ball-Hole is informing the chassis that the robot is in position. When this command is received, the chassis checks the status of the switch in the ball capture mechanism and acts accordingly.

Original Design

Our original design contained many of the ideas that remained throughout the project. Many modifications have been made since that time, but the illustration below shows the basic concept of our robot design.

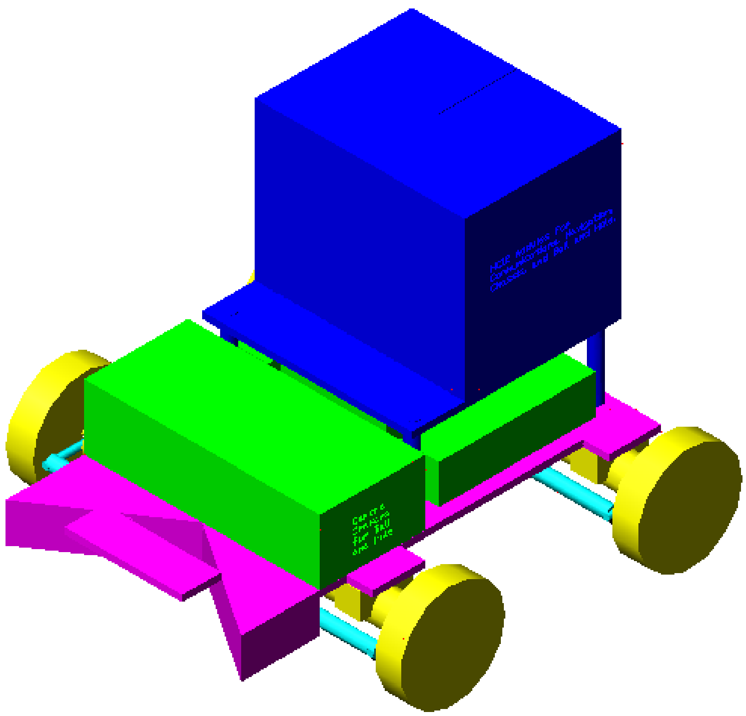


Figure 1: The original chassis design. There is a new structure for the wheel base and the ball capture mechanism, but other than that, the chassis is the same as originally planned.

Final Design

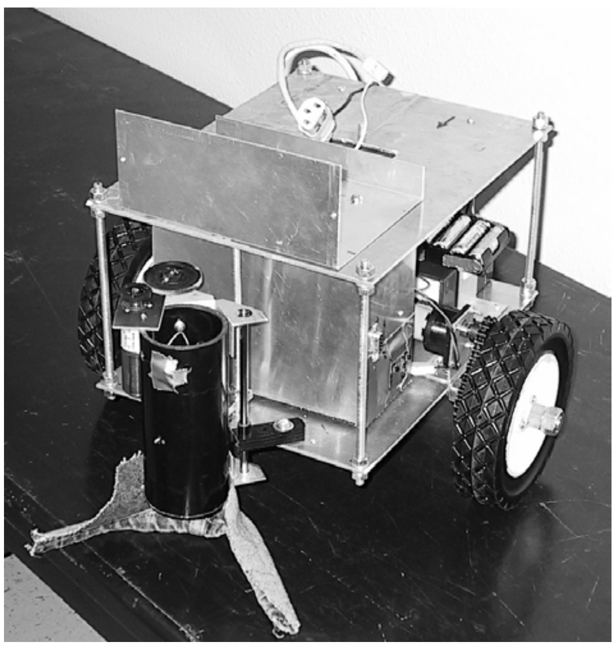


Figure 2: The final chassis design. Note the new ball-capture mechanism and the placement of the wheels. Instead of dual steering and a 4 wheel chassis, there are 2 drive wheels and a caster.

Hardware

Sensors

Sensors used in the chassis portion of this robot include a microswitch mounted in the ball capture mechanism and encoders on the drive motors.

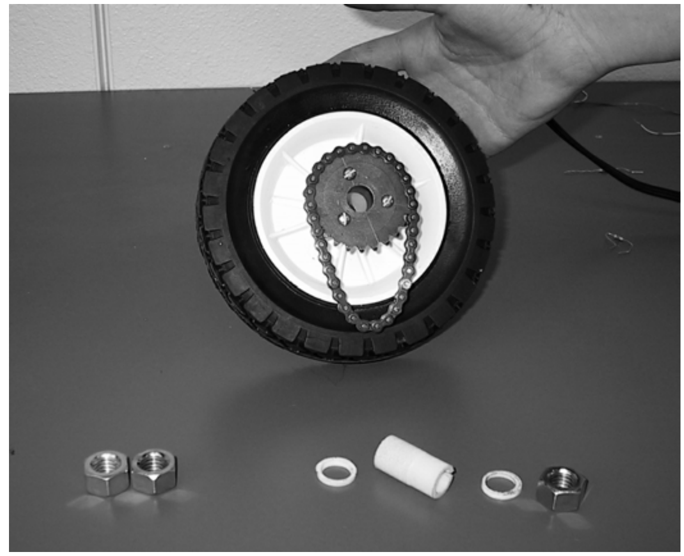
The microswitch is mounted at the height that a standard golf ball will reach after it pushes through the mouth of the PVC pipe. The switch is normally closed and opens upon contact with the ball. When the ball is sucessfully captured, a MOSFET circuit connected to the microswitch produces a five volt output. This tells the HC12 that the ball has been obtained and that it is now time to go find the hole. This information is delivered to the Navigation subsystem, which then delivers it to the Ball-Hole Location system.

The encoders consist of a thin disk with 500 slits near the border. These slits allow light from 2 diodes through to 2 photodetectors spaced the same distance apart as the slits. The square wave output by the photodetectors is output through the encoder to the HC12, where the data is used to calculate distance travelled and velocity.

Wiring

Figure 3 (next page) shows a general layout of the chassis wiring. The wiring in the chassis system consists of various gauges of stranded insulated cables. This allows for movement of the wires without breaking and prevents shorting. Wiring requirements called for reliable polarized connectors throughout the chassis. We used square polarized 4 conductor connectors to connect the NiMH battery packs to the regulators. Slot (female) and tab (male) connectors came with the lead acid batteries. The main H-Bridge connector was a 14 pin dual row rectangular connector with a polarizing bump on top. The chassis module used standard RJ45 (ethernet type) connectors for communications. All other I/O interfacing to the chassis module was accomplished via a single standard DB25 connector. Power was supplied to the navigation and ball/hole location subsystems using standard IEC power cables such as those used to plug appliances into wall outlets. Male IEC receptacles were mounted on the navigation and ball/hole location modules and power was supplied through the female sockets.

Drive System



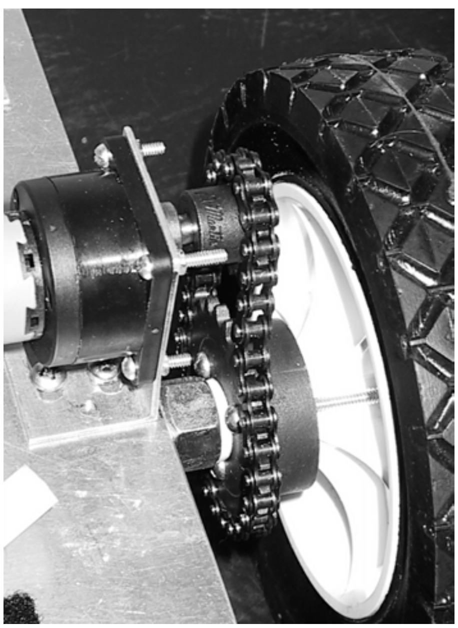


Figure 4: a)The axle with all machining as well as the bushing.
b)The bushing, washers, sprockets, and nuts used to hold the wheel on the axel and have it run smootly
c)The sprockets connected to the wheel and motor, run by a chain.

The chassis is mounted on a single axle and a caster. The axle is made out of 5/8" all-thread. 2" of each side is lathed down to 3/8" and is the actual axle for each wheel as shown in Figure 4a.

The last 1” of the axle is still threaded in order to allow the wheel to be locked onto the axle with nuts. The wheels rotate freely around the central axle on teflon bushings. Teflon washers were used to separate the lock nuts and the wheels, allowing for freer spin. (Figure 4b.) Attached to the wheels are sprockets, 2” in diameter. These are connected to the motors by chains, and allow the motors to rotate the wheels without putting the weight of the chassis on the motor shaft itself. (Figure 4c) The wheels are common utility tires used most often on small push-type lawn mowers. They are 6” in diameter and 1.25” thick. The caster mounted with a ¾” spacer keeps the back part of the chassis 3” off the ground.

The axle is mounted just in front of the motors. This keeps all the heavy items in the back such as the batteries, keeping the chassis from tipping over. The weight of the batteries was enough to counterbalance the camera from the Ball-Hole Location group.

Motor Choice

The motors we used were donated from previous Junior Design classes. They are Maxon 6-Watt motors with a 30:1 spur gearhead. This makes them incredibly powerful for their small size. The are 2.6 cm in diameter and about 6 cm long. They have included encoders, making it easier to control the movements of the robot. They are specified to run at 9V, but can be run at 12V, the unregulated voltage we are supplying.

Motorola MC68912B32 (HC12)

The Motorola HC12 is the microcontroller for the robot. The microcontroller is used to receive information sent by the Navigation group, interpret the information and then act on the information. Code listed in Appendix B and C is loaded into the memory of this board and then run. The HC12 is basically the brain of the robot.

Batteries

The high voltage power is supplied by lead acid batteries. They can deliver 6V for 4.5 Ah. We chose these batteries because of their long running time and high current capability relative to their small size. Two of these were connected in series to provide the 12V source.

The batteries supplying power for microcontrollers and other electronics are 1.2V, 1.6Ah NiMH cells chosen for their high energy density and sufficient current delivery. Twelve AA size cells were used, configured as two sets of six. Each set of six was connected in series. These two sets of six batteries were then connected in parallel to provide the 7.2V source that would be regulated to 5 V.

Ball Capture Mechanism

Structure

The ball capture mechanism was made of a 7" piece of 2" PVC pipe. The bottom of the pipe is laced with fishing line in order to fashion a simple mechanical grabber. (Figure 5) The ball stretches the fishing line apart just enough to get through. Once the ball is in the device, the line snaps back into its former shape. There are no moving parts to malfunction. The fishing line is quite sturdy. However, if the string broke, it could be relaced easily.

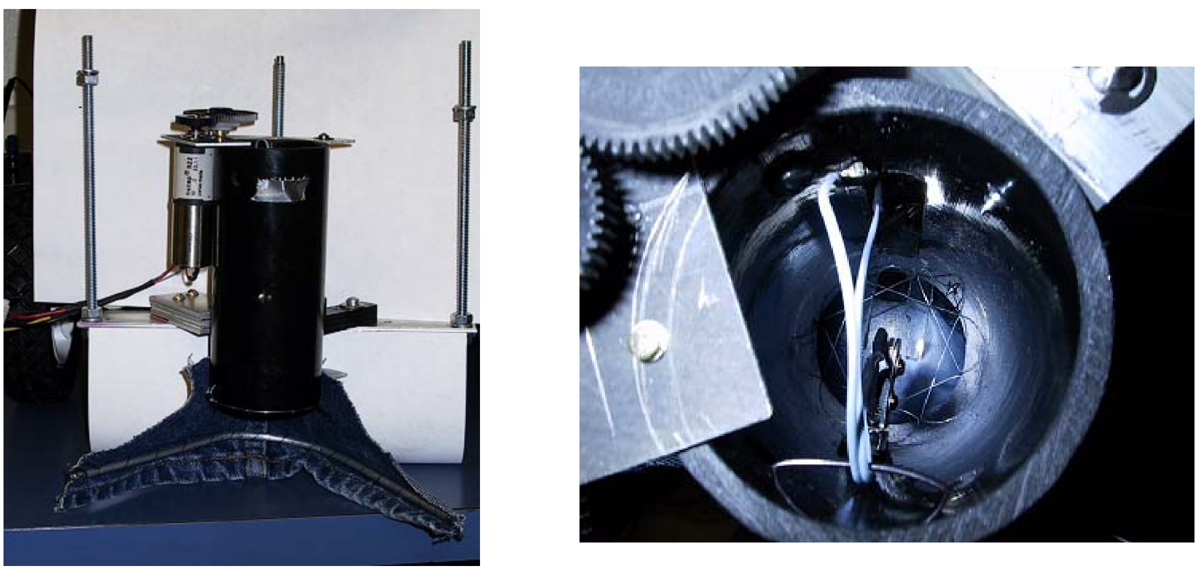


Figure 5: An image of the whole ball capture mechanism including the nut holder(on the left side just under the motor) and the stabilizer on the right side. The second figure is a view down the pipe showing the "fishing net" and the microswitch.

The PVC has 2 brackets mounted on it which hold a piece of all ¼" all-thread and a piece of ¼" bar stock. The all-thread is turned to raise and lower the mechanism. This process will be described later. The bar stock keeps the mechanism stable and in a consistent position on the chassis.

Drive Mechanism

On top of the pipe is mounted a small motor which is connected to the all-thread through a set of gears. This motor is a 24 V, 400 rev/minute motor that is run at 12V. There is a nut captured in a "wrench" bolted to the bottom layer of the chassis. This nut stays put while the all-thread is turned by the motor. This causes the whole device to be lowered. When the HC12 receives the command that the ball is under the device, the motor starts to turn, lowering the device. Once a microswitch mounted above the lacing triggers, signaling that the ball has been obtained, the motor reverses and brings the mechanism back up into the riding position. Once the robot has maneuvered to the hole, the device raises further, pushing a fork mounted on the chassis against the ball. This pushes the ball out of the device and into the hole.

We tried two different motors to run this device. The one that was finally used was a motor purchased from American Science and Surplus. We received no information on it, but it ran at 400 revolutions per minute at 12V. This combined with a 3:1 gear ratio to the drive shaft and 32 threads per inch on the drive shaft made for a nicely moving ball capture mechanism.

On the bottom of the mechanism is mounted a “skirt” used to direct the ball through the ball-hole location camera’s blind spot. It is made out of a piece of denim stitched and glued to an aluminum rod. The aluminum rod prevents the ball from rolling underneath the denim and weighs down the denim so it straightens out after being crushed. The denim allows the rod to fold up when the mechanism is lowered and won’t wear out easily, but will allow for rocks and other hard objects to move underneath without destroying anything.

H-Bridges

The speed of a DC motor can be effectively controlled by sending it a pulse-width modulated (PWM) signal. An H-Bridge is a circuit designed to deliver a high power pulse-width modulated signal to a motor using a low power PWM input, say from a microcontroller. Modern commercially-available H-Bridges come in integrated circuit chips and have inputs for PWM, direction and brake. We chose the National Semiconductor LMD 18200 H-Bridges to control the drive motors and ball capture mechanism motor. Some heat is dissipated in the process, so a large heatsink was added to avoid burnout. (Figure 6)

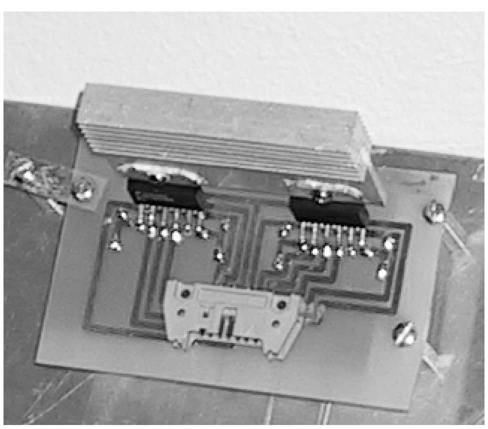


Figure 6: The Heat Sink on the H-bridges to avoid burnout.

Regulators

Power requirements were requested from the other robot subsystems prior to the selection and implementation of the voltage regulators. The microcontrollers require a regulated 5V input to operate correctly. Two 5V regulators with different current ratings were used. Regulating 6V down to 5V required regulators with low dropout voltages. The dropout voltage indicates how low the regulator's input voltage can drop while maintaining the correct output voltage. Modern regulators are available with dropout voltages down to tens of millivolts, so this was not a problem.

One regulator was needed to satisfy the communication group's request for an isolated power supply. It was estimated in the preliminary design that the communication electronics would draw a maximum of 300mA. Therefore a regulator capable of delivering a maximum of half an amp (500mA) of current was chosen. This is the Texas Instruments TPS7350Q with a dropout voltage of 75mV @ 300mA. This means that theoretically, the input voltage can drop down to 5.075V before the output voltage begins to decrease.

The second regulator supplied power to the chassis, navigation and ball/hole location subsystems, the motor encoders and the ball capture mechanism's MOSFET switch circuit. It was estimated that all these loads together would draw no more than one amp of current, so we chose a regulator that could deliver 1.5A. This is the National Semiconductor LP3962EMP-5.0

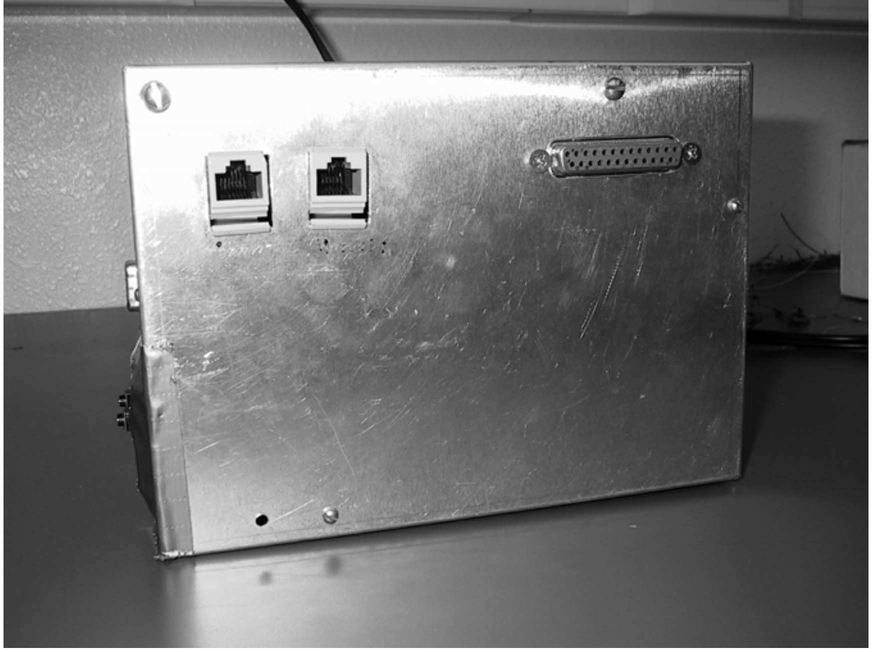
with a dropout voltage of 200mV @ 1A. This means that the input voltage should be able to drop down to 5.2V while maintaining an output of 5V.

There was a mistake made when ordering the main (1.5A) regulator. We decided on the 8 pin dual-inline package, but received a surface-mount chip. Due to time constraints, the surface-mount chip was implemented. The regulator for the communications subsystem came in the 8 pin dual-inline package as desired.

In hindsight, a better package choice for both regulators would have been the TO-220, which has a convenient metal tab with a hole drilled for easy heatsink mounting. Our heatsinking was achieved by mounting the regulators on the bottom of the circuit boards and attaching the boards inside the module so that the chips made contact with the metal surface of the box.

<u>Circuit Boards</u>

The circuit boards used by our group include 2 regulator boards, 2 H-bridge boards, and the HC12. All but the HC12 were fabricated by us. The PCB layout was designed using Protel®, a computer program that was readily available in the labs. The designs were printed onto special transfer paper using a normal printer. That paper was placed on copper circuit boards and ironed to transfer the circuit design to the copper surface. The boards were then placed in a heated PCB etching solution available from Radio Shack. Holes were drilled in the boards using drill



presses. Finally, the components and pins were soldered on. Below are the schematics for each board.

Modules

Each group was given a box in which to mount all their boards and accessories. They were allowed to mount on top of the module as well. These modules were aluminum boxes 3"x5"x7", large enough to contain 2 HC12s and a few other small boards. One external wall to the box is set aside for plugs. These plugs are "dummy-proof", preventing mistakes by tired engineers during testing. In this way, all the debugging of each module could be done by each separate group, and the final robot would have a finished appearance. The robot could then be shipped and a new person could hook everything up, knowing that it would all work because the plugs are labeled, making assembly simple.(Figure 8)

Figure 8: This picture shows one side of our module and the polarized plugs used.

Software

All of the programming for the robot was done in the C programming language. One of the main reasons that C was chosen was due to the fact that the programmer had most of his past experience using C. Using C allowed the chassis group to be able to separate the different aspects of the control into appropriately named subroutines. Breaking the program down into smaller parts makes programming easier and increases readability for the end user. There were separate subroutines for initializing the HC12, acquiring data, turning, driving forward, driving backward, stopping, controlling the ball capture mechanism, and repositioning the robot for Ball-Hole Location scans.

Data is acquired from the Navigation subsystem of the robot. After the data is acquired, the chassis acts based upon the subsystem controlling the robot at the time. When navigation is in control, the chassis program is designed to turn the robot the requested number of degrees and then travel the specified distance. Two functions are set up to implement these requests. The canned_turn function is used to turn the robot a certain number of degrees. In canned_turn, the heading is checked to determine whether or not the heading is less than or greater than 180 degrees. If the heading is greater than 180 degrees, the heading is subtracted from 360 degrees and the robot is setup to turn left the modified value of degrees. If the heading is less than 180 degrees, the robot is setup to turn right the requested number of degrees. In the program, turning is implemented by setting the wheels to go in opposite directions, i.e. the right wheel moves backward and the left wheel moves forward when turning right.

The move function is used to move the robot based upon the distance and direction given. In the case of the navigation group being in control, the direction is always set-up for forward movement. The pulse width is initialized at 40% duty cycle for distances greater than one foot, whereas it is setup at 20% for distances less than one foot. When the robot has moved within one foot of the requested distance, it slows down to half the speed it was at originally. This part of the function was added to help keep the chassis from tipping when it comes to a stop. After the robot has executed these commands, an acknowledgement is sent to navigation informing them that the chassis is ready to receive another command.

When the Ball-Hole group is in control, the canned_turn function works the same as when the navigation group is in control. However, since there is only a range of 0 to 16 degrees to the left or the right, when the value to turn is to the left, the degrees turned to the left are subtracted from 360 degrees before being sent to the canned_turn function. The canned_turn function then subtracts this value from 360 degrees to return back the number of degrees to the left to turn. The move function works when Ball-Hole is in command by setting the duty cycle for the wheel motors to the speed requested plus an arbitrary constant. There are also separate subroutines for other possible commands the Ball-Hole group can send.

When the relocate command for Ball-Hole is called, the chassis executes a certain command depending upon the number of repetitive times the function has been called. Upon receiving this command for the first time, the robot is ordered to turn 120 degrees to the right. When this command is received for the second time, the robot is ordered to turn another 120 degrees. On the third consecutive time the command is called, the robot is then ordered to backup 3 feet. If another type of command is requested following a relocate command, the loop is broken. Thus, the robot starts at the beginning of the loop the next time the relocate command is called.

In the stop command for the chassis, the duty cycles are set to zero for each of the wheels. In the original phases of programming, the brakes on the H-bridges for each motor were used to stop the robot. However, upon further testing, it was determined that the brakes were unnecessary. The weight of the robot is sufficent enough to stop the robot quickly once the duty cycle has been pulled. This enabled the chassis group to free up the two outputs on the HC12 that were being used to control the brake.

A separate subroutine was written to deal with capturing and dropping the ball when the Ball-Hole group tells the chassis that the robot is in position for pick-up or drop-off. A microswitch is set up inside the ball capture mechanism to determine whether or not the ball is currently captured. When the microswitch is closed, the chassis group can assume that the ball is within the mechanism. When the Ball-Hole group tells the chassis that the ball or hole is in position, the ball_hole subroutine determines if the microswitch is open or closed. If the switch is open, the ball capture mechanism's motor is set in the downward direction and the duty cycle is set to 60%. The ball capture mechanism continues down until the switch has been closed and a 1.2 second delay has occurred. Once the delay has been implemented, the direction is switched, and the ball capture mechanism raises for 12.8 seconds and then stops. The first delay is used to ensure that the ball has had time to lock into place. The second delay is used to ensure that the ball cpature mechanism is not dragging along the ground after picking the ball up. The time for the delays were used based upon tests done on the mechanism. If the microswitch is closed, we know that we have the ball captured already. Thus, we assume that the hole has been located. The ball capture mechanism is set-up to rise up and the duty cycle is set to 60%. When the switch loses contact with the ball, the ball capture mechanism continues up for an additional 6.4 seconds. This delay is used to ensure that the ball pops off, and is set up so that as soon as the motors stop, the ball will pop out.

The wheel speed is controlled via closed loop control, so that the robot goes relatively straight when moving forward, and there is limited slippage in turning. The distance traveled as well as the angle turned is determined by checking both motor encoders and executing a program using the Altera chip on the memory expansion board of the HC12. The number of ticks read from the encoders per revolution of the motors are stepped down in the Altera code. It was found that for one inch of movement, there were 98 ticks and for one degree of turn, there were 11.5 ticks. The Altera code used for the closed loop control was graciously shared with the chassis group by Team 7 from Spring 2001 Junior Design.

Conclusion

Overall, the robot was a success. As a team, we learned to work together. Although we were not successful at integrating with the other subsystems, we were able to complete many aspects of our design. We sharpened our time management skills immensely by learning the hard way. Had we devoted the time and effort at the start of the project that we contributed in the end, we may have developed a well oiled machine. However, the effort itself was a valuable lesson in teamwork, engineering skills and career preparation.

Appendix A:

Specs

Weight: 35 lbs

Height: 28”

Width: 15”

Depth: 16”

References

HC12 Advance Information, Motorola INC, 2000

Full Power Budget

Main Nickel Metal Hydride Power Budget

Subsystem	Maximum Current Draw	Maximum Power Needed
Ball/Hole Location	400mA	2W
Navigation	300mA	1.5W
Chassis	300mA	1.5W
Total	1.3A	6.5W

Communication System Nickel Metal Hydride Budget

Subsystem	Maximum Current Draw	Maximum Power Needed
Communication	300mA	1.5W

Lead Acid Power Budget

Motors	Maximum Current Draw	Maximum Power Needed
Ball Capture Mechanism Motor	.75A	9W
Main Drive Motors (together)	2A	24W
Total	2.75A	33W

Battery Life

Using the maximum expected current delivery required and (Amp)x(Hour) rating of batteries, run time can be calculated using the following equation:

Hours of run time = (Ah rating) / (Current Drawn)

The NiMH batteries are rated at 1.6Ah each. When connected in series, the total Ah rating is roughly equivelant to that of one battery. Thus, the run time of the main batteries in the worst case scenario (maximum current draw for the entire run time) was calculated as follows:

Main NiMH Run Time = (1.6Ah)/(1.3A) = 1 hour and 14 minutes

The same NiMH batteries were used for the communication subsystem and the run time is significantly longer:

Communication Subsystem Run Time = (1.6Ah)/(.3A) = 5 hours and 20 minutes

The lead acid batteries are rated at 4.5Ah each and were connected in series to provide a 12V supply for the motors. The calculation for their run time is shown below:

Lead Acid Run Time = (4.5Ah)/(2.75A) = 1 hour and 38 minutes

Budget

ITEM	COST
parts from the instrument room	**$60.82**
batteries	**$86.62**
h-bridges and voltage regulators	**$55.44**
maxon drive motor	**$54.15**
sprockets, gears, chains	**$52.80**
wheels	**$6.04**
connectors	**$40.37**
ball capture motors	**$34.50**
NiMH battery charger	**$30.00**
fuses, fuse holders, switches	**$14.65**
TOTAL	**$435.38**

Appendix B: Altera code

Appendix C: HC12 code