Summer 7-1997

The Robot Car

Punit Mukhija
University of Tennessee - Knoxville

UNIVERSITY HONORS PROGRAM

SENIOR PROJECT - APPROVAL

Name: PUNIT MUKHIJA

College: ENGINEERING Department: ELECTRICAL ENGINEERING

Faculty Mentor: DR. R. E. BODENHEIMER

PROJECT TITLE: THE ROBOT CAR

I have reviewed this completed senior honors thesis with this student and certify that it is a project commensurate with honors level undergraduate research in this field.

Signed: Robert E. Bodenheimer, Faculty Mentor

Date: 7/22/97

Comments (Optional):

The Robot Car

by

Punit Mukhija

Advisor : Dr. R. E. Bodenheimer

Introduction

Abstract

This paper is based on my experiences as a member of the University of Tennessee's Robot Car team for Southeastcon'97. The object of this paper is to present the technical, as well as the non-technical aspects encountered during the preparation for and at Southeastcon'97. This paper is about the Robot Car competition and about my personal successes and failures associated with the competition. I have attempted to present the paper in a manner so that even a reader not familiar with the field of Electrical Engineering may comprehend the paper.

The IEEE and Southeastcon'97

The Institute of Electrical and Electronics Engineers (IEEE) was founded in 1884 by a few practicing electrical engineers. Today the IEEE is comprised of more than 320,000 members in 147 countries and is the world's largest technical professional society. The IEEE focuses on advancing the theory and practice of electrical, electronics and computer engineering (http://www.ieee.org, 07/97).

IEEE Southeastcon is an annual technical conference aimed at bringing together Electrical Engineering professionals, faculty and students through technical sessions, tutorials and exhibits. Participants of the Southeastcon include universities and colleges from the Southeastern part of the United States. Southeastcon'97 was held in Blacksburg, Virginia on the Virginia Tech Campus (http://www.vt.edu, 07/98).

Hokie Hunt

The robot car hardware competition at Southeastcon'97 was titled "Hokie Hunt." The competition took place on a 12' x 12' plywood playing board. Polar coordinates were painted on the board in white. A pit of 2' diameter and providing a 4" drop was at the center of the table. At two opposing corners there were the "nests" which were square pits with a depth of 4" . The "nests", an area of 1'x1' each, were surrounded by a 1/2" thick and a 2" high yellow boundary wall. At the other two opposing ends were the starting home bases for the cars. Each home base was a 1' square area designated by a 1/8" red line. Starting contacts at the corner of the home at a height of 1/2" and 1.5" extend 6" on both walls. The starting contacts were metallic strips that went from a higher to a lower voltage level to indicate the start. Figure 1 and Figure 2 show isometric and top views of the playing board.

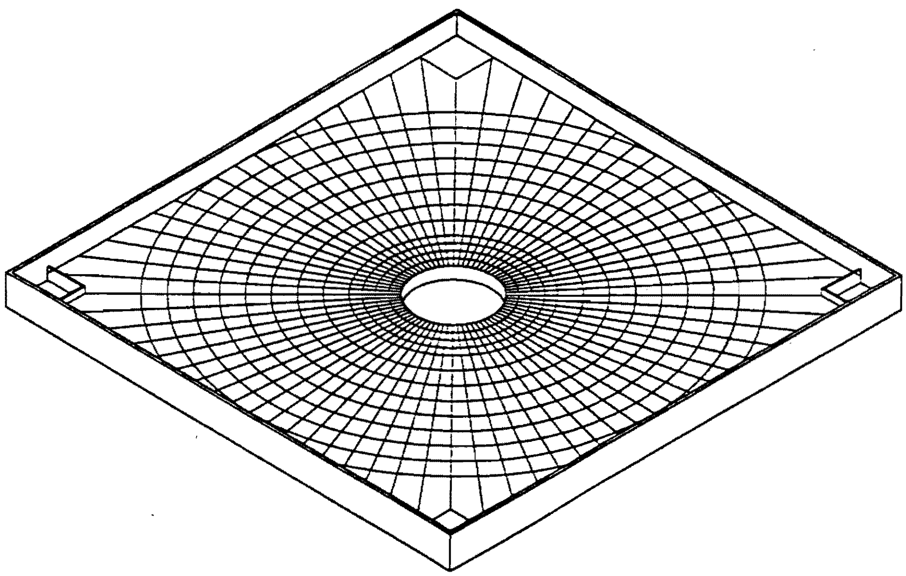


Figure 1 . Isometric view of the playing board

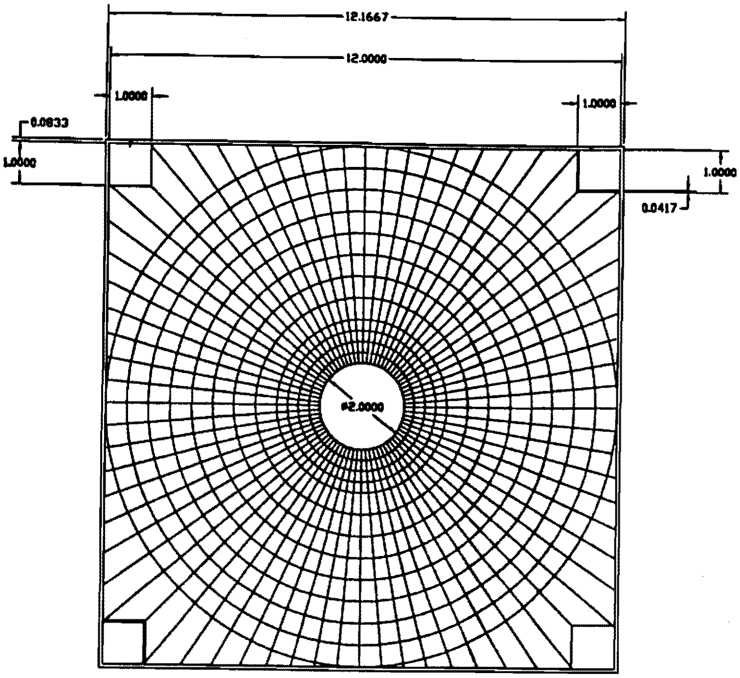
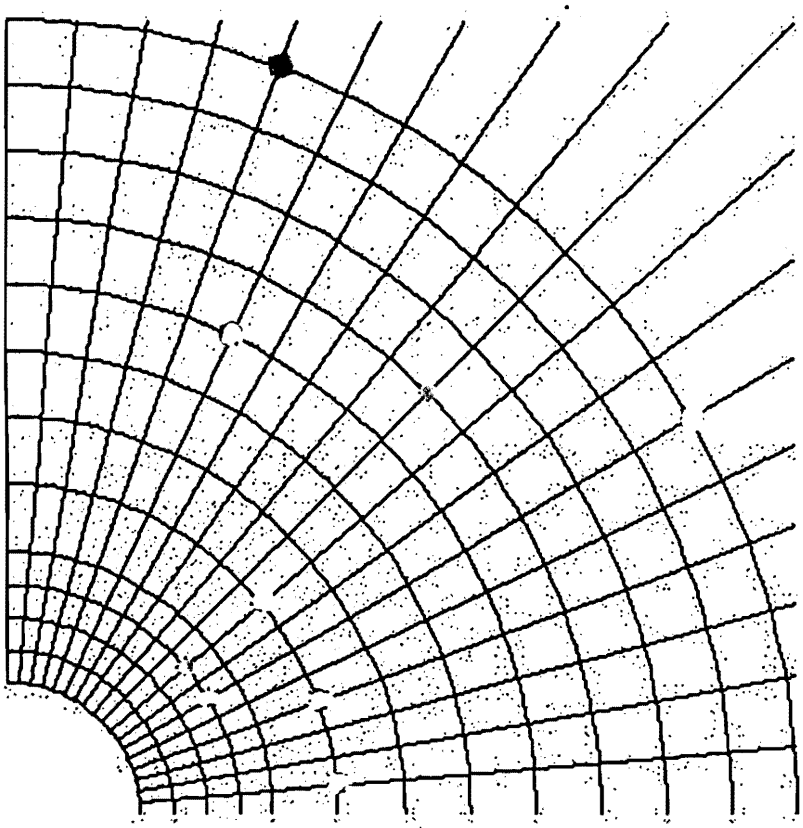


Figure 2. Top view of the playing board

The vehicles were to be less than 1' square and no taller than 1.5'. Vehicles were allowed to extend out on their own after the competition had started. The vehicles would start at home and travel around the table picking up 1/2" diameter balls placed at specified intersections of the polar coordinates. There were three types of balls placed symmetrically in four quadrants on the table . One brass ball worth 8 points each, 4 steel balls worth 3 points each and 4 nylon balls worth -2 points each were placed in each quadrant. A random quadrant map was generated 15 minutes before each round and given to the participants. Figure 3 illustrates a random ball placement map for a single quadrant. Each of the four quadrants had the same ball distribution.



Legend:

Material	Color Code
Steel	Blue
Nylon	Green
Brass	Red

Figure 3. Random ball distribution map of one quadrant

Vehicles start at home and gain points by collecting the balls and dropping them off at their respective nests. All teams competed in the first two rounds of competition. The first round consisted of only one vehicle competing on the track against the clock. In this round the vehicle had 8 minutes to collect as many points as possible. In the second round, two vehicles started off in opposite corners and fought each other to get as many points as possible within 8 minutes. The top 4 teams from the first two rounds qualified to the semifinals; the two teams with the most points in the semifinals reached the final.

The Car

Car design

The University of Tennessee Robot Car Team consisted of 7 student members under the supervision of Dr. R. E. Bodenheimer. All of the team members had already had some senior level specialization courses which meant that all the members had something unique to offer to the team. The team was subdivided to work on the different parts of the car by matching the students' strengths and the different aspects of the car. Robert Jones, the team leader, worked on the ball retrieval system. Michael Vann was in charge of the sensors. Monte Cooper and Greg Evans took care of the battery and the motors. Scott Hause and Rahul Bhatt did the majority of the floor plan for the car.

The front of the car formed a V shape as shown in Figure 4. A one way opening door guarded the opening of the V. As the car traveled on the track, balls would roll into the V but would not be allowed to roll out because of the one way gate.

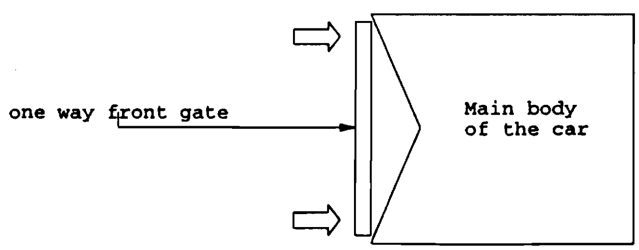


Figure 4. Top view of the car showing the **V** and the front gate

Once the balls were in the V they ended up being towards the narrow part of the V as the car moved forward. At this point a conveyor belt system was used to lift the balls up. The balls were pushed up between the rolling belt and a metallic frame. Figure 5 illustrates the conveyor belt system.

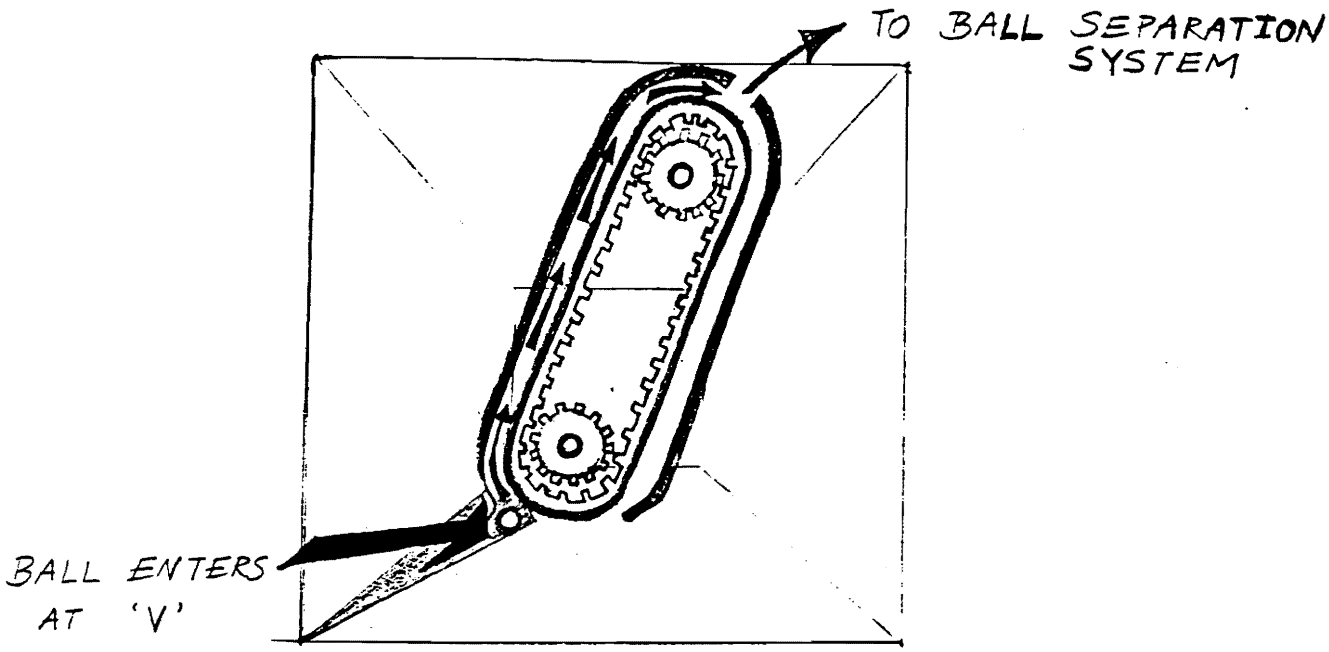


Figure 5. Conveyor belt system

From the conveyor belt the balls moved onto the separation stage. The brass and steel balls had positive points while the nylon balls counted for negative points. Our end goal was to place the brass and steel balls in our nest while storing the nylon balls.The balls from the conveyor belt were fed into a pipe that forked into two different directions. Balls were separated using the fact that nylon is a non-conductor of electricity while steel and brass are conductors. Figure 6 shows the separation system. A nylon ball would move from Point A unhindered to the direct opening ahead.

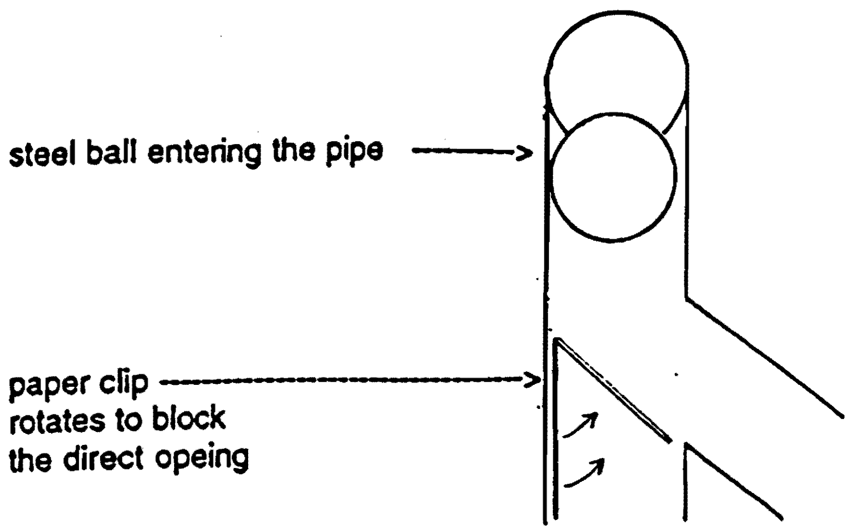


Figure 6. Ball separation system

Whenever a steel or brass ball would approach Point A, a paper clip would rotate to block the direct opening and thus forcing the ball to flow down the opening on the right. The two distinct openings would lead to different storage areas, one for the positive points and the other for negative points. Balls would be stored in their respective storage pipes until the car was at the nest. At this point, a pipe's opening was programmed to open up so that the balls dropped into the nest. Opening one pipe did not mean that the other pipe would be opened as well; this meant that we could drop the positive point balls into our nest without dropping the nylons.

The car was also equipped with three sensors that differentiated between the black background and the white lines on the table. These sensors were used in the programming to keep the car on track and keep account of where the car was on the playing board. There were two front wheels and a single caster at the back. A circuit designed by Michael Vann was used to keep track of the battery power. Whenever the batteries started becoming weak a small red LED (Light Emitting Diode) would turn on.

Programming

My personal contribution to the robot car was programming the microprocessor, Motorola's 68HC11. The 68HC11 is essentially a computer shrunk down to the size of a postage stamp in the form of a single Integrated Circuit (IC) chip. Motorola's 68HC11 did for the car what the brain does for a living being. If the sensors are to be considered as the car's eyes, then it is the microprocessor which allowed the car to make sense of where the car was. In a similar manner, if the wheels are taken to be the car's limbs, it is the microprocessor that allowed the car to maneuver as programmed.

The programming had four main tasks to perform:

1. To keep the car on a white line
2. To keep track of the car's position on the board
3. To direct the car towards specified coordinates
4. To drop off the balls when the car is at the nest

To perform these tasks, the hardware provided the microprocessor with inputs from:

1. 2 line sensors to keep the car on a white line
2. 1 tracking sensor to help count the intersecting lines
3. 1 push button gate to detect a collision with the nest wall

Figure 7 shows the view of the car from the bottom with the car traveling toward the arrow. M1 and M2 are the two motors that control the speed of the wheels on their respective sides. S1 and S2 are the two line sensors used to keep the car on a white line. S3 is the tracking sensor used to keep count of the number of lines that intersect the line the car is traveling on.

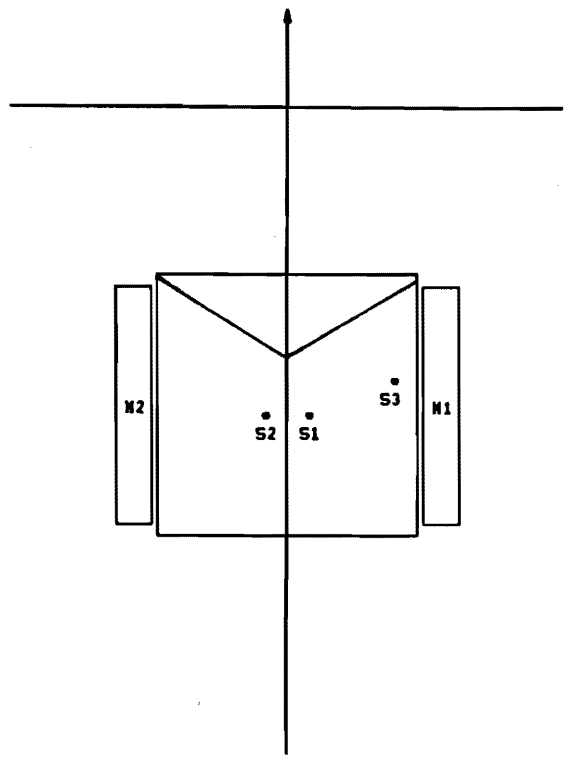


Figure 7. Bottom view of the car indicating the location of the sensors and the motors

The basic algorithm that was used to keep the car on a white line is highlighted in the flowchart illustrated in Figure 8. For instance, if the car started to deviate to the left (in Figure 7), S1 would eventually be over the traveling line. At this point, M1 would be slowed down causing the car to arc towards the right. Once S1 came over the black background, M1 would come back to its original speed. Thus, the car was kept on track by slowing down the motor whose sensor saw the white line.

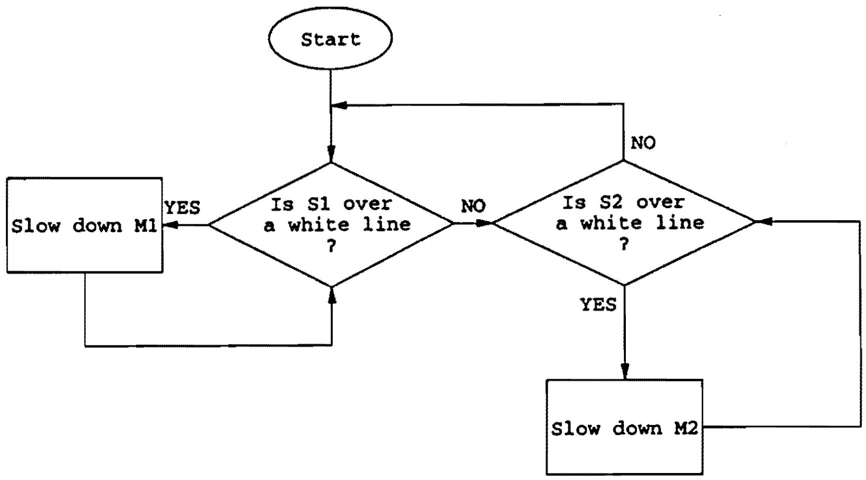


Figure 8. Flowchart to keep the car on a white line

Keeping the car on track was the most difficult aspect of programming but not the only aspect involved. The third sensor (S3 in Figure 7), was used to keep account of the car's position. S3 only saw white whenever the car passed over an intersecting line. A table was set up in the microprocessor's memory that kept a count of the number of lines that the car had passed.

Another table was set up to direct the car and tell it how many lines it should pass. The car would keep traveling on a particular circle until the lines it had passed equaled the lines it was told to pass.

For the ease of programming, the program itself was divided into smaller programs . The function of the major smaller programs or subroutines was as follows:

1. To make the car travel on straight line
2. To make the car travel on the various circles in the polar coordinate system
3. To make the car turn right
4. To make the car turn left
5. To count the intersections that the car passes
6. To drop off the balls

The subroutines were coded separately and tested. Once the subroutines worked individually the were combined to form the complete and final program. Appendix A contains a guide on how to operate the program and a listing of the programming code.

At Blacksburg

Preparation

We arrived in Blacksburg at noon on Friday, April 18th, 1997. The competition was set to take place on Saturday at 9 am. We were allowed to practice on the playing boards used for the competition on Friday afternoon. This gave us just enough time for lunch at a local restaurant known for their hot wings. The "911 Challenge" was a special that put a customer's taste buds in direct competition against the hot spices of the chefs. If you ate 12 of their hottest wings in five minutes you got a free T-shirt. I played the "911 Challenge" and lost. I got a poor start at Blacksburg but I was assured by my team-mates that I could redeem myself at the other competition going on in town; the one that involved robotic cars and some of the brightest young engineering minds in this part of the country.

Practicing on the actual playing boards was quite an educational and uplifting experience. So far we had been working on a makeshift board put together by Robert and Rahul in their spare time. Our board's plywood had cracks and the paint job was not excellent which meant that our caster often got stuck and our sensors often misread the lines. The Virginia Tech boards cost about a $1000 each (compared to our table that cost us about $200). With the help of some last minute touches on the hardware and software we were able to bring our car up to competition level. Our car was performing better than ever and better than any other car we saw that evening. As the time for the competition got closer, the nervous energy also increased. We all knew that a years work would be put on display the next day. Working on the car had already been very rewarding in terms of the educational and social lessons learned. We had not worked in such a large group in any other class and there were quite a few valuable friendships made along the way. However, on that Friday night when I went to bed I didn't know that my most valuable and enlightening experience with the car was yet to come.

The Competition

There were about 15 participants that showed up on Saturday morning. The first round involved the teams playing against the clock. Each team was given a randomly generated map showing the distribution of balls 15 minutes before the start. These 15 minutes were for final programming and for placing the vehicle in home base. After that there was a 30 second "hands off" period and then the cars could take off.

Most of the cars did not perform well. By the time we went on the playing board eleven cars had had their shot at the 1st round. So far University of North Florida (UNF) was the only one that had put up a show. UNF had gained 72 points, and the other ten cars had failed to drop a single ball in their nest. We went on the playing board with the intention of covering 3 of the 12 circles on the board. The way we saw it, if we picked up even half of the balls on these three circles, we would be in good shape. The car was placed in the home base and we sat back counting the 30 seconds of "hands off." Four and a half minutes later we had covered all our intended circles and were dropping off steel and brass balls in the nest. We had secured 42 points and were firmly in second position. The UNF captain came up to me, shook my hand and wished us luck. So far UT and UNF were the teams to watch. And it was obvious that UNF was watching us.

The second round involved head to head competition of two teams. Two teams would start out at opposite corners of the table and try to gain as many points as possible in 8 minutes. We were to play the University of Central Florida (UCF) and UCF in their first round had not scored any points. Our confidence was growing and when we got our ball map it was quite obvious as to which 3 circles we were going to travel. I programmed the car and handed it over to Mike and Rob to place it in the home base. The 30 seconds hands off period started and we waited eagerly. With about 10 seconds left for the start, our little car became a bit too impatient and took off. Our car picked up every single ball on the three circles and deposited all the positive points. All this time the UCF vehicle had not moved out of their home base. UCF did not obtain any points and they did not even get out of their home base. On the other hand, we got disqualified because our vehicle jumped the gun.

The four leaders from the first two rounds qualified to the semifinals. Although, we did not score any points in the second round, we still got into the semifinals comfortably purely on the basis of our first round performance. The two highest scorers in the semifinals were to enter the finals. In the semifinals we were paired against Old Dominion University (ODU) while UNF met University of Alabama, Birmingham (UAB).

ODU had gotten to the semifinals thanks to their strong performance in the second round. ODU had a very large and powerful vehicle. They ran the same pattern each time and traveled in a zigzag between the central pit and the boundary walls. The ODU vehicle covered only their home base's quadrant and their nest's quadrant. Though they covered only half the table, they did so in a very time efficient manner.

We knew that we could give ODU a run for their money (or balls) but we were afraid of getting into a head on collision with their large vehicle. In the case of a collision we knew that our vehicle would be pushed off its track and would not be able to make it to the nest to drop off any balls. ODU did not run on the lines but had a definitive pattern which meant that the had a better chance of getting to their nest after a collision. Another point we had in mind was that ODU ran a short pattern and dropped off the balls in the nest and then came back for more balls which they would later drop off. This meant that if we ran into ODU after they had completed one short pattern, ODU would already have some points secured in their nest.

We considered all this but decided to gamble on our regular routine. Our team picked three loaded circles to travel and the programming was done accordingly. The car traveled the first circle and picked up all the balls in its path. On the second circle our car picked up most of the balls and a rather large vehicle. ODU's vehicle was much bigger and knocked us off our path. Their vehicle got entangled with ours, literally, and got their front wheels thrown into the pit. At this point, the two teams called off the match. ODU had already dropped of some balls into their nest and they won our semifinals.

Next up were UNF and UAB. UAB realized that they needed just a handful of points to beat ODU's score. They did exactly that and secured themselves as the second highest scorers in the semifinals. UNF, of course, was as impressive as ever. And when UNF and UAB met in the finals, UNF gained the first position quite deservedly.

Conclusion

We met ODU for a battle over the 3rd and 4th position. Convinced that our previous routine would lead to another collision we came up with a new strategy. Our plan was to pick up the nearest nylon ball, drop it off at ODU's nest and block off their nest by leaving our car there. This would mean that ODU would get negative points for the nylon ball and we would win.

I had fifteen minutes to program the car to perform the new maneuver. The original program was modified to have the car travel straight toward's the opponent's nest and drop off all the balls picked up on the way. I titled this program "ODU" and tried it out on the practice board. The program worked. "ODU" was modified so that only the nylon balls would be picked up and I labeled this program "RISK." "RISK" made the car zigzag around the steel and brass balls so that these balls would not be collected. We tested this program and it too worked like a charm. This was the first time that I had tried to program the car to run in such a zigzag manner and I was very pleased with the success on the practice board.

I reloaded the program onto the car's microprocessor and handed the car over to be placed in home base. I could not help but think how wonderful this new strategy would look. We were all so confident, and I felt certain that we had the 3rd position secured.

The starting contacts went from a high to a low voltage and the two vehicles took off. Our car headed towards the opponents nest. On the way it picked up all the balls without dodging any balls. I had loaded "ODU" instead of "RISK." At ODU's nest we dropped off one nylon and one steel. We blocked off their nest and their vehicle could not get past our car. At the end, ODU was awarded +3 points for the steel and -2 points for the nylon. We lost by a single point.

I knew then and there, that my most important lesson from this experience was not one about electrical engineering; it was about life. I had let overconfidence get to me and in my lapse of concentration I had loaded the wrong program. I had counted my chickens before they hatched and this cost us the 3rd place.

That night I blamed myself for our 4th place finish. Fortunately, my team-mates were a lot kinder and forgiving. They made me realize that the competition had been a success for us. We had put up a fine exhibition of engineering and all the teams out there knew that we had worked hard to get this far. Nevertheless, even today, I look back at times, and I am amazed at how I could program a robot car but could not differentiate between "RISK" and "ODU."

Appendix A

A few simple changes in the program allow a user to run the car in any specified manner. The points at which these changes are needed are highlighted in the program code by successive lines of “&”.To operate the program follow these instructions:

1) Specify the circles the car should cover; the outer most circle is 1 and the inner most circle the car can cover is circle 9.

2) Specify the number of circles that the car needs to cover;

   Circles = (number of circles) - 1

3) Specify the number of intersections to cover for each circle except the last one

4) Specify the number of intersections to cover on the last circle

```
* Punit Mukhija

* Conections: TTL start indicatior - PA7
* Sensor1 - IC1    Line sensor1 on the side of Motor1
* Sensor2 - IC2    Line sensor2 on the side of Motor2
* Sensor3 - IC3    Tracking sensor
* Motor1  - OC3    Radially faster (outer circle)
* Motor2  - OC2    Radially slower (inner circle)
* +/- M1  - PA3    Direction of motion for motor1
* +/- M2  - PA4    Direction of motion for motor2
REGBAS  EQU     $1000 Starting address for address block
PORTA   EQU     $00   Data register Port A
TCNT    EQU     $0E   Timer counter register
TOC2    EQU     $18   Output compare 2 register
TOC3    EQU     $1A   Output compare 3 register
TCTL1   EQU     $20   Timer Control 1
TCTL2   EQU     $21   Timer Control 2
TMSK1   EQU     $22   Timer Mask 1
TFLG1   EQU     $23   Timer Flag 1
PORTC   EQU     $03   PORT C
DDRC    EQU     $07   DATA DIRECTION FOR PORTC
PORTB   EQU     $04   PORT B
*********** ************ used or not; FIRST = 0 indicates normal tracking
NORM2    RMB     2     Represent HP of forward moiton frequency for M2
NORM3    RMB     2     Represent HP of forward moiton frequency for M1
SLOW2    RMB     2     Represent HP of forward moiton frequency for M2
SLOW3    RMB     2     Represent HP of forward moiton frequency for M1
HP3      RMB     2     Represent HP of forward moiton frequency for M1
HP2      RMB     2     Represent HP of forward moiton frequency for M2
FLAG     RMB     1
DELY     RMB     2     Delay counter
XP       RMB     1     Present X
YP       RMB     1     Present Y
XW       RMB     1     Wanted X
YW       RMB     1     Wanted Y
W        RMB     1
CIRCLES  RMB     1     NUMBER OF CIRCLES COMPLETED
LF       RMB     1
LFLG       RMB     1


*************        1) SET THE CIRCLES TO COVER
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&& TABLE contains the co-ordinates containing the balls in the order that
*&&&&& the car will go to
TABLE    FCB    7, 8, 9, 5
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

```
***** Program begins here******************************************************
START    LDS    #$0047   Load Stack pointer for user stack on EVB
         LDX    #REGBAS  Pointer to register block



******************        2) SET THE NUMBER OF CIRCLES TO COVER
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

         LDAA    #3          NUMBER OF CIRCLES - 1
         STAA    CIRCLES

*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&



         JSR    LINE     Load NORM and SLOW with line speeds


********** MOVE FORWARD SLOWLY

         LDAA   #%00000000      PA3 = 0, PA4 = 0
         STAA   PORTA,X         FORWARD DIRECTION
         STAA   DDRC,X          Configure PortC for input
         STAA   PORTB,X
         JSR    WAIT      Jump to subroutine WAIT
***  Start of the race

*** Forward motion till S2(IC2) sees a rising edge
         JSR    INOC23F   Initialize Output Compare2 and 3 for forward motion


*** Car is now moving forward till IC2 sees a rising edge

         LDAA    #$00        Deactivate input captures
         STAA    TCTL2,X
         LDAA    #0
         STAA    YP
         STAA    XP          Initialize XP and YP
         LDAA    #0
         STAA    XW          INITIALIZE X WANTED


         LDY     #TABLE       Y has TABLE ADDRRES
         LDAA    0,Y
         STAA    XW          ACCA =>  XW
```

```
         JSR      LCL

         DEC      CIRCLES

         BNE      HERE
         JSR      LCK
         JSR      SLINE          SLOW LINE

* stay here till CO goes high

NDROP   LDAA     #$50            OM2,3:OL2,3 = 0:1 (TOGGLE)
        STAA     TCTL1,X

NODROP   BRCLR   PORTC,X %00000001  NODROP

        LDAA     #$00            NO OUTPOUT
        STAA     TCTL1,X
        LDAA     #0
        STAA     PORTA,X
        BRCLR   PORTC,X %00000001  NDROP
*Ready to drop off balls
         LDAA    #1
         STAA    PORTB,X   DROP OFF OUR BALLS

         JSR    RRLONGD
         JSR    RRLONGD
         JSR    RRLONGD
         JSR    RRLONGD
         JSR    RRLONGD
         JSR    RRLONGD
         JSR    RRLONGD
         JSR    RRLONGD
         JSR    RRLONGD
         JSR    RRLONGD
         JSR    RRLONGD
         JSR    RRLONGD
         JSR    RRLONGD
         JSR    RRLONGD
         JSR    RRLONGD
         JSR    RRLONGD

         LDAA    #0
         STAA    PORTB,X

         SEI
         JMP    $E0B2
TRIAL      JMP     TRIAL
***** Program ends here*****************************************************

****************************************************************************
* WAIT  - Subroutine to wait till starting TTL goes low
* Note : PA7 is a default Input port
WAIT
STAY      BRSET  PORTA,X #%10000000 STAY      Stay here till PORTA is less than 12
          JSR    SHORTD
          BRSET  PORTA,X #%10000000 STAY      Stay here till PORTA is less than 12
          JSR    SHORTD
          BRSET  PORTA,X #%10000000 STAY      Stay here till PORTA is less than 12
          RTS              Return from subroutine
**** End of WAIT ***********************************************************

*Initialize OC2 and OC3 for forward motion  *********************************
INOC23F
        LDAA     #$50            OM2,3:OL2,3 = 0:1 (TOGGLE)
```

```
        STAA    TFLG1,X
        STAA    TMSK1,X
        CLI                     ENABLE INTERRUPTS
        LDAA    #$7E            Jump statement
        STAA    $DC             EVB Vector for OC2
        STAA    $D9             EVB Vector for OC3
        LDD     #OCFR2F         Load ACCD with the address of OCFR2F
        STD     $DD             Store at address DD
        LDD     #OCFR3F         Load ACCD with the address of OCFR3F
        STD     $DA             Store at address DA
        RTS                    Return from Subroutine
********************* OC2 and OC3 Initialized *****************************


*Initialize OC2 and OC3 for REVERSE motion  *********************************
INOC23R
        LDAA    #$50            OM2,3:OL2,3 = 0:1 (TOGGLE)
        STAA    TCTL1,X
        LDAA    #$60            Clear OC3F and OC2F
        STAA    TFLG1,X
        STAA    TMSK1,X
        CLI                     ENABLE INTERRUPTS
        LDAA    #$7E            Jump statement
        STAA    $DC             EVB Vector for OC2
        STAA    $D9             EVB Vector for OC3
        LDD     #OCFR2R         Load ACCD with the address of OCFR2R
        STD     $DD             Store at address DD
        LDD     #OCFR3R         Load ACCD with the address of OCFR3R
        STD     $DA             Store at address DA
        RTS                    Return from Subroutine
********************* OC2 and OC3 Initialized *****************************

************************** OUTPUT COMPARE *********************************
*OCFR2 is the service routine for OC2 for forward motion
OCFR2F
         PSHA
         PSHB
         BRCLR PORTA,X %00000010 S2off   if s2 is low then jump to s2off
         LDD    SLOW2    Decrease M2 speed
         STD    HP2      Store Half peroid for forward motion M2
         BRA    S2on
S2off
         LDD    NORM2
         STD    HP2      Store Half peroid for forward motion M2

S2on     LDD      HP2              Update TOC2 by adding HP - STRAIGHT
         ADDD     TOC2,X            by adding half period
         STD      TOC2,X            to the latest TOC2 value for next interrupt
         BCLR     TFLG1,X %10111111 Clear the OC2 flag
         PULB
         PULA
         RTI                    Return from interrupt

*OCFR3 is the service routine for OC3
OCFR3F
        PSHA
        PSHB
        BRCLR  PORTA,X %00000100 S1off   if s1 is low then jump to s1off
        LDD    SLOW3    Decrease M1 speed
        STD    HP3      Store Half peroid for forward motion M1
        BRA    S1on

S1off    LDD    NORM3
         STD    HP3      Store Half peroid for forward motion M1
```

```
        ADDD    TOC3,X           by adding half period
        STD     TOC3,X           to the latest TOC3 value for next interrupt
        BCLR    TFLG1,X %11011111 Clear flag OC3F
        PULB
        PULA
        RTI                      Return from interrupt

*****************************************************************************

*************************** OUTPUT COMPARE ***********************************
*OCFR2 is the service routine for OC2 for forward motion
OCFR2R
         PSHA
         PSHB
         BRCLR PORTA,X %00000010 SS2off    if s2 is low then jump to s2off
         LDD    NORM2     NORMAL M2 speed
         STD    HP2       Store Half peroid for forward motion M2
         BRA    SS2on
SS2off
         LDD    SLOW2
         STD    HP2       Store Half peroid for forward motion M2

SS2on     LDD     HP2             Update TOC2 by adding HP - STRAIGHT
         ADDD    TOC2,X           by adding half period
         STD     TOC2,X           to the latest TOC2 value for next interrupt
         BCLR    TFLG1,X %10111111 Clear the OC2 flag
         PULB
         PULA
         RTI                      Return from interrupt

*OCFR3 is the service routine for OC3
OCFR3R
        PSHA
        PSHB
        BRCLR  PORTA,X %00000100 SS1off    if s1 is low then jump to s1off
        LDD    NORM3     Decrease M1 speed
        STD    HP3       Store Half peroid for forward motion M1
        BRA    SS1on

SS1off    LDD    SLOW3
         STD    HP3       Store Half peroid for forward motion M1
SS1on
        LDD     HP3             Update TOC3
        ADDD    TOC3,X           by adding half period
        STD     TOC3,X           to the latest TOC3 value for next interrupt
        BCLR    TFLG1,X %11011111 Clear flag OC3F
        PULB
        PULA
        RTI                      Return from interrupt

*****************************************************************************

*********************** START TURNING LEFT ***********************************
LEFT
         JSR    RLONGD
         JSR    RLONGD
         LDAA   #%00010000        PA3 = 0, PA4 = 1
         STAA   PORTA,X           Start turning LEFT

         LDD     #10500
         STD    NORM3
         STD    SLOW3             FREQUECY FOR RIGHT MOTOR

         LDD     #12000
```

```
KL1     BRCLR   PORTA,X %00000100 KL1  WAIT HERE TILL S1 GOES HIGH
        JSR     SHORTD
        BRCLR   PORTA,X %00000100 KL1
        JSR     SHORTD
        BRCLR   PORTA,X %00000100 KL1
        JSR     SHORTERD
        BRCLR   PORTA,X %00000100 KL1
        JSR     SHORTERD
        BRCLR   PORTA,X %00000100 KL1

KL2     BRSET   PORTA,X %00000100 KL2    WAIT HERE TILL S1 GOES LOW
        JSR     SHORTD
        BRSET   PORTA,X %00000100 KL2
        JSR     SHORTD
        BRSET   PORTA,X %00000100 KL2
        JSR     SHORTERD
        BRSET   PORTA,X %00000100 KL2
        JSR     SHORTERD
        BRSET   PORTA,X %00000100 KL2
        JSR     SHORTERD
        BRSET   PORTA,X %00000100 KL2


KL3     BRCLR   PORTA,X %00000100 KL3  WAIT HERE TILL S1 GOES HIGH
        JSR     SHORTD
        BRCLR   PORTA,X %00000100 KL3
        JSR     SHORTD
        BRCLR   PORTA,X %00000100 KL3
        JSR     SHORTERD
        BRCLR   PORTA,X %00000100 KL3
        JSR     SHORTERD
        BRCLR   PORTA,X %00000100 KL3

KL4     BRSET   PORTA,X %00000010 KL4  WAIT HERE TILL S2 GOES LOW
        JSR     SHORTD
        BRSET   PORTA,X %00000010 KL4
        JSR     SHORTD
        BRSET   PORTA,X %00000010 KL4
        JSR     RSHORTD
        BRSET   PORTA,X %00000010 KL4
        JSR     RSHORTD
        BRSET   PORTA,X %00000010 KL4

        LDAA   #%00000000          PA3 = 0, PA4 = 0
        STAA   PORTA,X             Forward direction
        RTS                        TURN COMPLETE
********************** STOP TURNING LEFT **********************************


********************** START TURNING RIGHT ********************************
RIGHT

        JSR    RLONGD
        JSR    RLONGD
        LDAA   #%00001000        PA3 = 1, PA4 = 0
        STAA   PORTA,X            Start turning RIGHT

        LDD     #11000
SIR1    STD    NORM2
        STD    SLOW2              FREQUECY FOR LEFT MOTOR


        LDD  #12500
```

```
KR1       BRCLR  PORTA,X %00000010 KR1  WAIT HERE TILL S2 GOES HIGH
          JSR    SHORTD
          BRCLR  PORTA,X %00000010 KR1
          JSR    SHORTD
          BRCLR  PORTA,X %00000010 KR1
          JSR    SHORTERD
          BRCLR  PORTA,X %00000010 KR1
          JSR    SHORTERD
          BRCLR  PORTA,X %00000010 KR1

KR2       BRSET  PORTA,X %00000010 KR2    WAIT HERE TILL S2 GOES LOW
          JSR    SHORTD
          BRSET  PORTA,X %00000010 KR2
          JSR    SHORTD
          BRSET  PORTA,X %00000010 KR2
          JSR    SHORTERD
          BRSET  PORTA,X %00000010 KR2
          JSR    SHORTERD
          BRSET  PORTA,X %00000010 KR2

KR3       BRCLR  PORTA,X %00000010 KR3  WAIT HERE TILL S2 GOES HIGH
          JSR    SHORTD
          BRCLR  PORTA,X %00000010 KR3
          JSR    SHORTD
          BRCLR  PORTA,X %00000010 KR3
          JSR    SHORTERD
          BRCLR  PORTA,X %00000010 KR3
          JSR    SHORTERD
          BRCLR  PORTA,X %00000010 KR3

          LDAA   XP
          CMPA   #8
          BGT    SLCIR

KR4       BRSET  PORTA,X %00000100 KR4  WAIT HERE TILL S1 GOES LOW
          JSR    SHORTD
          BRSET  PORTA,X %00000100 KR4
          JSR    SHORTD
          BRSET  PORTA,X %00000100 KR4
          JSR    SHORTERD
          BRSET  PORTA,X %00000100 KR4
          JSR    SHORTERD
          BRSET  PORTA,X %00000100 KR4

          BRA     YO
SLCIR
          LDAA    #0
          CMPA    FLAG
          BNE     NODELAY
          JSR     RLONGD
          JSR     RLONGD
          JSR     LONGD
NODELAY   JSR     RLONGD
YO        LDAA   #%00000000          PA3 = 0, PA4 = 0
          STAA   PORTA,X             Forward direction
          RTS                        TURN COMPLETE
*********************** STOP TURNING RIGHT ***********************************

*********************** SPEED FOR LINE ***************************************
LINE      LDD    #10000
          STD    NORM3      Store Half peroid for forward motion M1
          STD    NORM2      Store Half peroid for forward motion M2
          LDD    #20000
```

```
        RTS                 Return
************ end line

*********************** SPEED FOR LINE ************************************
SLINE   LDD   #12000
        STD   NORM3          Store Half peroid for forward motion M1
        STD   NORM2          Store Half peroid for forward motion M2
        LDD   #22000
        STD   SLOW3         Store HP for forward motion M1 - SLOWING DOWN
        STD   SLOW2         Store HP for forward motion M2 - SLOWING DOWN
        RTS                  Return
************ end line

*********************** SPEED FOR VARIOUS CIRCLES *************************
*********************** SLOWER ONES ***************************************
SPEED1  LDAA    XP
        CMPA    #1
        BNE     gt1


        LDD   #8000
        STD   NORM3          Store Half peroid for forward motion M1
        LDD   #9200
        STD   NORM2          Store Half peroid for forward motion M2
        LDD   #16000
        STD   SLOW3         Store HP for forward motion M1 - SLOWING DOWN
        LDD   #18400
        STD   SLOW2         Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return

gt1     LDAA    XP
        CMPA    #2
        BNE     gt2

        LDD   #9000
        STD   NORM3          Store Half peroid for forward motion M1
        LDD   #10000
        STD   NORM2          Store Half peroid for forward motion M2
        LDD   #12000
        STD   SLOW3         Store HP for forward motion M1 - SLOWING DOWN
        LDD   #14000
        STD   SLOW2         Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return

gt2     LDAA    XP
        CMPA    #3
        BNE     gt3

        LDD   #8000
        STD   NORM3          Store Half peroid for forward motion M1
        LDD   #9400
        STD   NORM2          Store Half peroid for forward motion M2
        LDD   #13000
        STD   SLOW3         Store HP for forward motion M1 - SLOWING DOWN
        LDD   #14000
        STD   SLOW2         Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return

gt3     LDAA    XP
        CMPA    #4
        BNE     gt4

        LDD   #8000
        STD   NORM3          Store Half peroid for forward motion M1
```

```
        LDD     #13000
        STD     SLOW3       Store HP for forward motion M1 - SLOWING DOWN
        LDD     #15000
        STD     SLOW2       Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return

gt4     LDAA     XP
        CMPA     #5
        BNE      gt5

        LDD     #8000
        STD     NORM3        Store Half peroid for forward motion M1
        LDD     #9600
        STD     NORM2        Store Half peroid for forward motion M2
        LDD     #13000
        STD     SLOW3       Store HP for forward motion M1 - SLOWING DOWN
        LDD     #16200
        STD     SLOW2       Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return


gt5     LDAA     XP
        CMPA     #6
        BNE      gt6

        LDD     #9000
        STD     NORM3        Store Half peroid for forward motion M1
        LDD     #10800
        STD     NORM2        Store Half peroid for forward motion M2
        LDD     #14000
        STD     SLOW3       Store HP for forward motion M1 - SLOWING DOWN
        LDD     #16900
        STD     SLOW2       Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return


gt6     LDAA     XP
        CMPA     #7
        BNE      gt7

        LDD     #8000
        STD     NORM3        Store Half peroid for forward motion M1
        LDD     #9900
        STD     NORM2        Store Half peroid for forward motion M2
        LDD     #13000
        STD     SLOW3       Store HP for forward motion M1 - SLOWING DOWN
        LDD     #16800
        STD     SLOW2       Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return

gt7     LDAA     XP
        CMPA     #8
        BNE      gt8

        LDD     #8000
        STD     NORM3        Store Half peroid for forward motion M1
        LDD     #11000
        STD     NORM2        Store Half peroid for forward motion M2
        LDD     #13000
        STD     SLOW3       Store HP for forward motion M1 - SLOWING DOWN
        LDD     #17000
        STD     SLOW2       Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return
```

```
        BNE     gt9

        LDD     #17000
        STD     NORM3      Store Half peroid for forward motion M1
        LDD     #21000
        STD     NORM2      Store Half peroid for forward motion M2
        LDD     #26500
        STD     SLOW3     Store HP for forward motion M1 - SLOWING DOWN
        LDD     #29000
        STD     SLOW2     Store HP for forward motion M2 - SLOWING DOWN
        RTS               Return

gt9
        LDD     #11000
        STD     NORM3      Store Half peroid for forward motion M1
        LDD     #17000
        STD     NORM2      Store Half peroid for forward motion M2
        LDD     #28000
        STD     SLOW3     Store HP for forward motion M1 - SLOWING DOWN
        LDD     #32000
        STD     SLOW2     Store HP for forward motion M2 - SLOWING DOWN
        RTS               Return

****************************************************************** END SPEED

***********************LINE CIRCLE LINE ************************************
* ON STRAIGHT
*** CHECK SENSOR3 TO SEE IF AN INTERSECTION HAS BEEN REACHED
LCL

S3offL    BRSET   PORTA,X %00000001 S3offL  STAY HERE TILL S3 IS OFF
          JSR    SHORTD
          BRSET  PORTA,X %00000001 S3offL
          JSR    SHORTD
          BRSET  PORTA,X %00000001 S3offL


S3off   BRCLR  PORTA,X %00000001 S3off       STAY HERE TILL S3 IS SET
        JSR    RSHORTD
        BRCLR  PORTA,X %00000001 S3off
        JSR    RRSHORTD
        BRCLR  PORTA,X %00000001 S3off

****** INTERSECTION REACHED - CIRCLE
        LDAA   XP
        CMPA   XW
        BGT    D
        INC    XP               Update X

        LDAA   #1
        STAA   LF           NEED TO TURN RIGHT
        BRA    Y000

D       DEC     XP
        LDAA    #0
        STAA    LF             GOING BACK, NEED TO TURN LEFT
Y000    LDAA   XW               Compare XW and XP
        CMPA   XP               If not yet equal keep going
        BNE    S3offL           straight till next intersection

        LDAA   LF
        CMPA   #0
        BNE    RT
        JSR    LEFT
```

```
RT1
         JSR     SPEED1             Reload NORMs and SLOWs


*** CHECK SENSOR3 TO SEE IF AN INTERSECTION HAS BEEN REACHED      - STRAIGHT
S3offL1    BRSET  PORTA,X %00000001 S3offL1  STAY HERE TILL S3 IS OFF
           JSR    SHORTD
           BRSET  PORTA,X %00000001 S3offL1
           JSR    SHORTD
           BRSET  PORTA,X %00000001 S3offL1


S3off1     BRCLR  PORTA,X %00000001 S3off1
            JSR RRSHORTD
           BRCLR  PORTA,X %00000001 S3off1
           JSR RRSHORTD
           BRCLR  PORTA,X %00000001 S3off1
           JSR    SPEED



*******************                3) SPECIFY INTERSECTIONS
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

         LDAA #72              18 = 1 QUADRANT, 72 = ENTIRE PLAYING BOARD

*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&



         STAA   YW

****** INTERSECTION REACHED
         INC     YP                 Update Y
         LDAA    YP
         INCA
         CMPA    YW
         BEQ     SP
         LDAA    YW                 Compare YW and YP
         CMPA    YP                 If not yet equal keep going
         BNE     S3offL1             Keep going till next intersection
         BRA     L2
SP       JSR     SPEED1               SLOW DOWN
         BRA     S3offL1
******** Turn left if the CAR needs to go towards the center, otherwise,
******** turn to the right
L2       INY                   Point to next X
         LDAA    0,Y
```

```
         LDAA    XP
         CMPA    XW
         BGT     RT9
         JSR     LEFT     Turn left
         BRA     YOO
RT9      JSR     RIGHT
YOO      JSR     LINE     Load speeds for a straight line travel
* BACK TO LINE
        LDAA #0
        STAA  YP


         RTS
*****************************************************************************



***********************LINE CIRCLE BACK ***********************************
* ON STRAIGHT
*** CHECK SENSOR3 TO SEE IF AN INTERSECTION HAS BEEN REACHED
LCK

S3offK    BRSET   PORTA,X %00000001 S3offK  STAY HERE TILL S3 IS OFF
          JSR    SHORTD
          BRSET  PORTA,X %00000001 S3offK
          JSR    SHORTD
          BRSET  PORTA,X %00000001 S3offK


S3ofK   BRCLR  PORTA,X %00000001 S3ofK
         JSR     RSHORTD
         BRCLR  PORTA,X %00000001 S3ofK
         JSR RRSHORTD
         BRCLR  PORTA,X %00000001 S3ofK
*S3 ON
****** INTERSECTION REACHED - CIRCLE
         LDAA    XW
         CMPA    XP
         BGT     OLD
         DEC     XP
         LDAA     #1
         STAA     LFLG
         BRA     COMPARE

OLD      INC     XP              Update X
         LDAA     #0
         STAA     LFLG

COMPARE  LDAA    XW              Compare XW and XP
         CMPA    XP              If not yet equal keep going
         BNE     S3offK          straight till next intersection

         LDAA    #0
         STAA    FLAG
         LDAA    #1
         CMPA     LFLG
         BNE      OLD2
         JSR     LEFT
         BRA     WOW
OLD2     JSR     RIGHT           TURN RIGHT HERE
*** Right turn completed, motors are still slow

WOW      JSR     SPEED1           Reload NORMs and SLOWs
```

```
*** CHECK SENSOR3 TO SEE IF AN INTERSECTION HAS BEEN REACHED      - STRAIGHT
S3offK1    BRSET  PORTA,X %00000001 S3offK1  STAY HERE TILL S3 IS OFF
           JSR    SHORTD
           BRSET  PORTA,X %00000001 S3offK1
           JSR    SHORTD
           BRSET  PORTA,X %00000001 S3offK1


S3ofK1     BRCLR  PORTA,X %00000001 S3ofK1
           JSR RRSHORTD
           BRCLR  PORTA,X %00000001 S3ofK1
           JSR    RRSHORTD
           BRCLR  PORTA,X %00000001 S3ofK1
            JSR   SPEED




**********          4) SPECIFY NUMBER OF INTERSECTIONS FOR THE LAST CIRCLE
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

          LDAA #90

*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&



         STAA  YW

****** INTERSECTION REACHED
          INC    YP               Update Y
          LDAA   YP
          INCA                     ACCA = YP + 1
          CMPA   YW
          BEQ    SPK
          LDAA   YW               Compare YW and YP
          CMPA   YP               If not yet equal keep going
          BNE    S3offK1           Keep going till next intersection
          BRA    K2
SPK       JSR    SPEED1             SLOW DOWN
          BRA    S3offK1

          LDAA   #1
          STAA   XP
          LDAA   #1
          STAA   FLAG

K2        JSR    RIGHT      Turn RIGHT - TO FACE BACKWARDS
```

```
***************** LONG DELAY
LONGD
      JSR DELAY
      JSR DELAY
      JSR DELAY
      JSR DELAY
      RTS      DELAY OF 0.8SECONDS
******************

****************** REALY LONG DELAY
RLONGD
      JSR LONGD
      JSR LONGD
      JSR LONGD
      JSR LONGD
      JSR LONGD
      JSR LONGD

      RTS      DELAY OF 0.8*4SECONDS
******************

****************** 0.2 SECOND DELAY
DELAY
       LDD  #$FFFF
       STD  DELY
NOTYET DEC  DELY
       BNE  NOTYET
       RTS  0.2 SECOND DELAY
******************

SHORTD
       LDD  #30000
       STD  DELY
NTYET  DEC  DELY
       BNE  NTYET
       RTS  0.1 SECOND DELAY
******************

RSHORTD
       LDD  #2   REALLY SHORT DELAY
       STD  DELY
RNTYET  DEC  DELY
       BNE  RNTYET
       RTS
******************


RRSHORTD
       LDD  #1   REALLY, REALLY SHORT DELAY
       STD  DELY
RNTT  DEC  DELY
       BNE  RNTT
       RTS
******************


SHORTERD
       LDD  #100    SHORTER THAN SHORT DELAY
       STD  DELY
RTT    DEC  DELY
       BNE  RTT
       RTS
******************
```

```
RRLONGD
        JSR    RLONGD
        JSR    RLONGD
        JSR    RLONGD
        JSR    RLONGD
        JSR    RLONGD
        JSR    RLONGD
        JSR    RLONGD
        JSR    RLONGD
        JSR    RLONGD
        JSR    RLONGD
        JSR    RLONGD
        JSR    RLONGD
        JSR    RLONGD
        JSR    RLONGD
        JSR    RLONGD
        JSR    RLONGD
        RTS             END OF REALLY, REALLY LONG DELAY
*******************************************************************************

***  Get the  wantedX
***  TABLE contains the starting address of the TABLE
*        LDY  #TABLE    Load Y with the TABLE address
*        LDAA  XW,Y     Load XW
*        INY            For next XW
****************************** ***********************************************

*********************** Faster one *******************************************
********************** SPEED FOR VARIOUS CIRCLES ******************************
SPEED   LDAA     XP
        CMPA     #1
        BNE      gt11


        LDD    #8000
        STD    NORM3      Store Half peroid for forward motion M1
        LDD    #9200
        STD    NORM2      Store Half peroid for forward motion M2
        LDD    #16000
        STD    SLOW3     Store HP for forward motion M1 - SLOWING DOWN
        LDD    #18400
        STD    SLOW2     Store HP for forward motion M2 - SLOWING DOWN
        RTS              Return

gt11    LDAA     XP
        CMPA     #2
        BNE      gt21

        LDD    #5300
        STD    NORM3      Store Half peroid for forward motion M1
        LDD    #5790
        STD    NORM2      Store Half peroid for forward motion M2
        LDD    #8000
        STD    SLOW3     Store HP for forward motion M1 - SLOWING DOWN
        LDD    #9225
        STD    SLOW2     Store HP for forward motion M2 - SLOWING DOWN
        RTS              Return

gt21     LDAA     XP
        CMPA     #3
        BNE      gt31

        LDD    #5300
        STD    NORM3      Store Half peroid for forward motion M1
```

```
        LDD     #8000
        STD     SLOW3       Store HP for forward motion M1 - SLOWING DOWN
        LDD     #9350
        STD     SLOW2       Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return

gt31     LDAA     XP
        CMPA     #4
        BNE      gt41

        LDD     #5200
        STD     NORM3        Store Half peroid for forward motion M1
        LDD     #5800
        STD     NORM2        Store Half peroid for forward motion M2
        LDD     #7825
        STD     SLOW3       Store HP for forward motion M1 - SLOWING DOWN
        LDD     #9525
        STD     SLOW2       Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return

gt41     LDAA     XP
        CMPA     #5
        BNE      gt51

        LDD     #5200
        STD     NORM3        Store Half peroid for forward motion M1
        LDD     #5900
        STD     NORM2        Store Half peroid for forward motion M2
        LDD     #7850
        STD     SLOW3       Store HP for forward motion M1 - SLOWING DOWN
        LDD     #9525
        STD     SLOW2       Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return


gt51     LDAA     XP
        CMPA     #6
        BNE      gt61

        LDD     #4000
        STD     NORM3        Store Half peroid for forward motion M1
        LDD     #5025
        STD     NORM2        Store Half peroid for forward motion M2
        LDD     #7750
        STD     SLOW3       Store HP for forward motion M1 - SLOWING DOWN
        LDD     #9700
        STD     SLOW2       Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return


gt61     LDAA     XP
        CMPA     #7
        BNE      gt71

        LDD     #4000
        STD     NORM3        Store Half peroid for forward motion M1
        LDD     #4160
        STD     NORM2        Store Half peroid for forward motion M2
        LDD     #7725
        STD     SLOW3       Store HP for forward motion M1 - SLOWING DOWN
        LDD     #10025
        STD     SLOW2       Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return
```

```
        BNE     gt81

        LDD     #4000
        STD     NORM3        Store Half peroid for forward motion M1
        LDD     #5400
        STD     NORM2        Store Half peroid for forward motion M2
        LDD     #7850
        STD     SLOW3       Store HP for forward motion M1 - SLOWING DOWN
        LDD     #10750
        STD     SLOW2       Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return

gt81     LDAA    XP
        CMPA    #9
        BNE     gt91

        LDD     #6000
        STD     NORM3        Store Half peroid for forward motion M1
        LDD     #9200
        STD     NORM2        Store Half peroid for forward motion M2
        LDD     #11500
        STD     SLOW3       Store HP for forward motion M1 - SLOWING DOWN
        LDD     #17000
        STD     SLOW2       Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return

gt91
        LDD     #7000
        STD     NORM3        Store Half peroid for forward motion M1
        LDD     #11000
        STD     NORM2        Store Half peroid for forward motion M2
        LDD     #18500
        STD     SLOW3       Store HP for forward motion M1 - SLOWING DOWN
        LDD     #22000
        STD     SLOW2       Store HP for forward motion M2 - SLOWING DOWN
        RTS                 Return

****************************************************************** END SPEED
```

Bibliography

"IEEE." N.page. ONLINE. Available; http://www.ieee.org.

Greenfield, Joseph D. The 68HC11 Microcontroller. Orlando; Harcourt Brace & Company, 1991.

"Southeastcon'97." N.page. ONLINE. Available; http://www.vt.edu

Spasov, Peter. Microcontroller Technology: The 68HC11. Englewood Cliffs; Prentice Hall, 1996.