

John E Bertsch
8677 Vickie Lynn Ln
Apt 9
Brighton MI 48116

Race Car for the SR-52

Race Car is a game of skill written for the Texas Instruments SR-52 programmable calculator, in which the player maneuvers a car around a race track drawn on graph paper. The object of the game is to drive the car from the start to the finish line in the best possible time without leaving the track. The program calculates the coordinates of the car based on steering, throttle and brake information entered by the driver. Players can choose to go left, straight or right (relative to the driver's seat), and to accelerate, brake or cruise. Each time the player makes a move, 1 second of time elapses.

For each move, the program calculates the X and Y coordinates of the car's new position and the elapsed time. The player simply plots the car's position and makes the new move based on the present position. Using different colored pens, two or more players can race on the same track and compete for the best finishing time.

The program moves the coordinates of the car during the 1 second time interval based on a velocity vector constructed by the player's decisions. Left and right steering increments are fixed at $+15^\circ$ and -15° , respectively. For example, if you go left, the velocity vector is rotated $+15^\circ$ from its present direction. The acceleration and deceleration constants are $+1$ division per second² and -1 division per second². If you wish to accelerate, the magnitude of the velocity vector is increased by one and the distance traveled during that move will be one division farther than the previous move. If you decide to cruise, the magnitude of the velocity vector remains constant and the distance traveled during that move will be the same as the previous move.

Text continued on page 30

Listing 1: The Race Car program, written for the SR-52.

Loc	Keys						Commentary
000	*LBL	*C	STO	0	1		Store XX.YY.
005	INV	*D.MS	INV	*D.MS	*fix	0	Mask off YY and store XX.
011		*D.MS	INV	*fix			
014		STO	0	3			
017	+/-	+	RCL	0	1	=	Get XX.YY, mask off XX
023		x	1	0	0	=	and store YY.
028		STO	0	4	0	HLT	
033	*LBL	*D'	STO	0	5		Store the starting angle.
038		0	INV	*fix	HLT		
042	*LBL	*E'	RCL	0	3		Initialize X position.
047		STO	0	6			
050	RCL	0	4	STO	0	7	Initialize Y position.
056	RCL	0	5	STO	0	8	Initialize velocity angle.
062	0	STO	0	9			Set velocity magnitude
066		STO	1	0			and elapsed time = 0.
069	GTO	*A'					
071	*LBL	A	1	5			Increment velocity angle

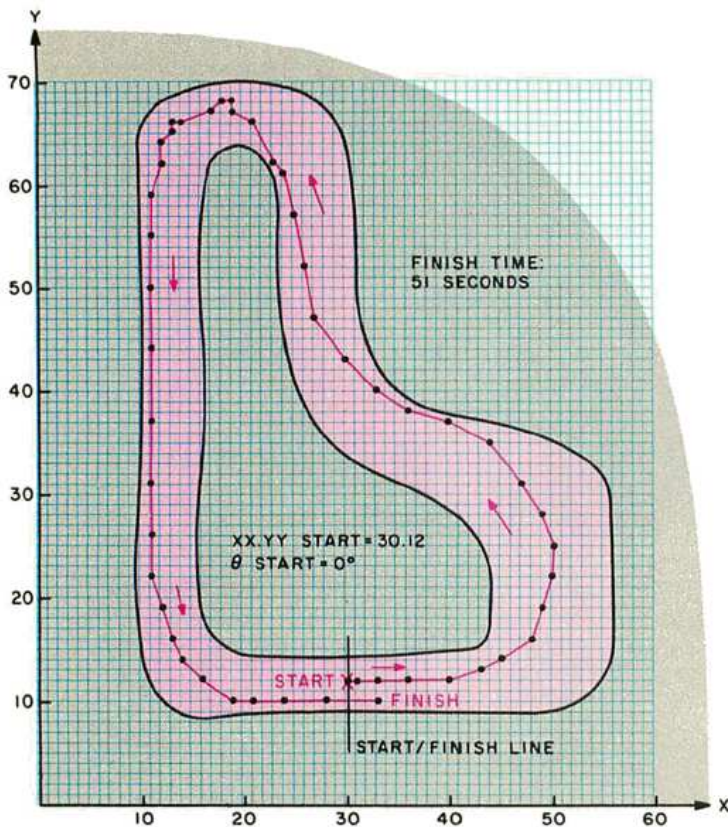


Figure 1: A typical route for one player in the game of Race Car, written for the Texas Instruments SR-52 programmable calculator. Players must stay within the bounds of the track (which is arbitrarily drawn on graph paper) or risk disqualification. Increments of change to acceleration and steering are deliberately restricted in order to make the game more challenging, but players have free choices within these limitations.

Remember that you are steering the race car as if you were sitting in the driver's seat, so don't confuse your left and right directions. The degree of difficulty can also be set in the layout of the race track. Hair-pin turns and straightaways will test your skill and make the game exciting. I've had hours of fun racing with my friends and family; I hope you will, too. ■

User Accessible Labels:

A = Left	A' = X.YT
B = Right	B' = Vel
C = Cruise	C' = X.Y Start
D = Accel	D' = θ Start
E = Brake	E' = Init

Register Utilization:

00 = P/R conversion
01 = Work area
02 = Work area
03 = Initial X coordinate
04 = Initial Y coordinate
05 = Initial velocity angle
06 = X coordinate
07 = Y coordinate
08 = Angle of velocity vector
09 = Magnitude of velocity vector
10 = Elapsed time
11 = Work area
12 = Work area

Race Car Operating Instructions

1. Draw the race track on a suitable piece of graph paper, preferably ten divisions to the inch. Figure 1 shows an example of a typical race track. The race track must be located in the area bound by the lines $X=0$, $X=100$, $Y=0$ and $Y=100$. It is a good idea to leave space between the edge of the race track and the boundaries, since the car might leave the track. Select the combination start and finish line on the track and the initial direction of the car.
2. Enter sides A and B of the program.
3. Set the D/R switch to D.
4. Enter the starting coordinates of the car in the form XX.YY, where XX is the initial X coordinate and YY is the initial Y coordinate. Both numbers must be positive integers between 0 and 100. Press *C.
5. Enter the initial direction of the car. This is a positive angle in degrees measured from the positive X axis to the initial direction of the car. This angle should equal $n \times 15^\circ$, where $n = 0, 1, 2, 3, \dots, 23$. Press *D.
6. Press *E' to initialize the game. The initial X and Y coordinates will be displayed in the form XX.YY000.
7. Select the direction in which you will steer the car.

Press A to steer left. Zero will be displayed.
Press B to steer right. Zero will be displayed.

If you want to go straight, skip this step and proceed to step 8.

8. Select the throttle and brake conditions.

Press C to cruise.
Press D to accelerate.
Press E to brake.

The position of the car and the elapsed time will be displayed in the form XX.YYTTT, where:

XX = X coordinate of the car.
YY = Y coordinate of the car.
TTT = Elapsed time in seconds.

9. Repeat steps 7 and 8 for each move. If the car should leave the track, the player is disqualified. Once the car passes the finish line, the winning time is read from the display.
10. To display the magnitude of the velocity vector, press *B'. To return the display back to XX.YYTTT, press *A.
11. To play a new game on the same race track, press *E'.