The Organic Robot

By: Ken Boone

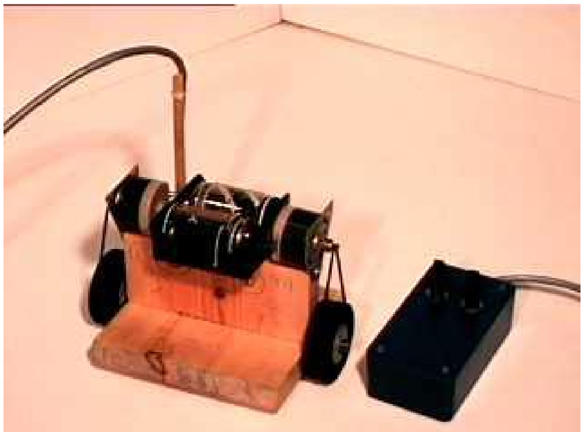
The Organic Robot is a simple hands-on classroom presentation that lets students design, program and build a simple robot in the classroom. The presentation can be performed without tools or expensive robot kits. The Organic Robot was originally developed for amateur robotics hobbyist to use to teach robotics to gifted elementary school students. Since the presentation was developed it has also been used with middle and high school students, Boy Scouts and adults. Practically any technically oriented teacher/volunteer can learn to give the Organic Robot presentation. The presentation can be performed in two to three hours or used as the introduction of an entire unit on robotics.

Only a few simple props and a black board are needed. Instead of using an expensive robot (that the whole class has to share) the presentation actually lets the students brainstorm to design the robot and brainstorm to develop a robotic programming language. After the robot is designed and the programming language is developed the students flow chart a program for the robot using their programming language. All of the design, development and programming are performed on a black board. The robot is constructed out of organic parts (the students). Then the robot executes the flow charted program.

History of the Organic Robot

Since 1980 members of the Triangle Amateur Robotics Club (TAR) have been giving robotic presentations for the Parents for the Advancement of Gifted Education (PAGE) Saturday morning program. The PAGE program was developed by a group of Raleigh NC parents that wanted their gifted children doing something better than watching TV on Saturday mornings. In the early days we demonstrated our robots to the class and talked about robotics. The students enjoyed the demonstrations, but it was hard to keep the student's attention when we talked. We thought the class would be better if each student could actually built a robot they could take home when the class was over.

We developed a wire control device called the TAR Rat (Rat). The Rat basically was a wooden base with two wheels, a couple of motors and a battery pack that was attached with wires to a box with two switches. The students enjoyed making the Rats and over 100 of them where built in the classes. After the Rats where built the students put their Rat's through several exercises where they simulated robotic tasks. The Rat went through several design iterations and plans for the last design can be found on my web page: http://www.kensrobots.com.



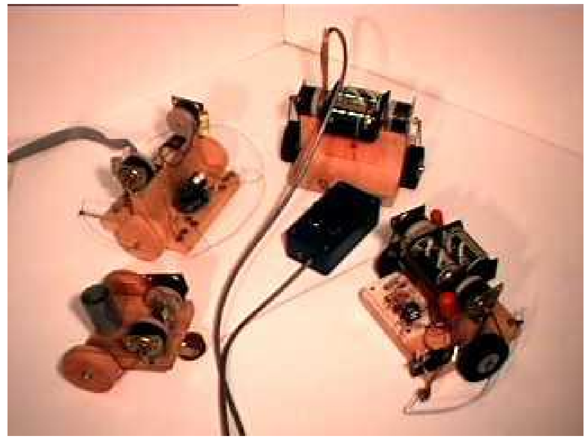


Photo 1- Last version of the wire controlled TAR Rat that was used to teach gifted children.

Photo 2- Several versions of the TAR Rat including the one on the bottom right that solves mazes.

The Rat was good but it really wasn't a robot, we had to charge extra for the class to pay for the supplies and it took too much out of class time to prepare the parts. This is where the Organic Robot came in. We started discussing how we could design a small modular real robot the students could help assemble and program during the class. We finally decided that a real robot was impractical. It would be so expensive the students would not be able to take one home, it would require more preparation time then the Rats and it would be difficult to involve all of the students. We just couldn't figure out how to build a robot the students could assemble and program in just 6 one-hour classes.

Then I got the ideal for the Organic Robot. Why not brainstorm with the students to let them design the robot; brainstorm to develop the programming language; and let the students write the program all on the black board, (with some guidance from the presenter) then build the robot out of the students. Since the students are the organic part of the Organic Robot they even get to take part of the robot home with them. Telling younger students they will be able to take part of the robot home with them helps keep their attention. The best part is that the Organic Robot presentation takes little out of class preparation time and you can do the presentation in just three one hour classes. Table 1 lists the materials needed to do the Organic Robot presentation.

Materials required:

For each group of five to eight students:

Large paper grocery bag
Three blindfolds
Compass - If you already have one.
Ultrasonic tape measure - If you already have one.
Notepad and pencil

For the class:

Pair of two-way hand held radios - If you already have them.
Very bright light
Drop cord

These items are not required but greatly enhance the presentation:

True robotic toy - One that can be programmed to do different things. There is usually an inexpensive robotic toy in a large toy store that can be programmed to move forward, reverse, turn and make sounds.

Several of the tiny wind up walking, hopping, flipping toys *

Several battery powered walking plush toys * - The barking dog that sits down is great. If you have two identical barking dogs take the fur off of one of them.

Radio controlled (RC) or wire controlled toy * - TAR Rat

Furby - If you have one. Mine has removable fur.

Overheads or pictures of actual robots - Find pictures of robots in science magazines, amateur robot books and magazines and on the web.

* It is easy to find these items at yard sales.

Table 1 - Lists the materials required in the classroom to give the Organic Robot presentation.

The First Hour

During the first hour of class develop definitions for a robot and a machine and illustrate the definitions with mechanized toys and robots.

What is a robot?

Start out with the class materials in closed boxes. After introducing yourself ask the students to define a robot. Do not say any definition is wrong but keep agreeing positively when someone mentions part of the definition you are looking for. Once someone or consecutive students come up with definitions that include; a computer, programming and do different things. Then give a simplified definition of a true robot. I like to use: "A robot is a machine that can be programmed to do a task. Once programmed it can repeat the task without further programming and it can be programmed to do another task without redesigning the robot."

What is a machine?

Next start taking items out of the boxes. Start out with tiny wind up walking and hopping toys. Take one out, wind it up and let it start across your desk. Ask; "Is this a robot?" You will usually get both yes and no answers from the students. Then define a machine. "A machine is a device that is built to only do one thing. It can not do something different unless the machine is taken apart and put back together in a different way."

Continue taking wind up toys out and asking the question; "Is this a robot or a machine"? Then take out a battery powered toy that looks like a robot and ask; "Is this a robot?" A lot of younger students get this question wrong. Explain again that this machine can not change what it does. Take out the dog with the fur and plastic body removed, turn it on and start it across the desk. Ask the question again. What is this? While they are guessing get the second dog out that still has fur and start it running. Show them that the dog is just a very ingenuously designed machine that can bark, sit down and walk. But you can not change what it does.

Tele-Operated Machine?

Next bring out a TAR Rat or any wire or RC control device and drive it around. Ask the question; "Is this a robot or a machine?" After getting the replies from the students discuss the difference between a tele-operated or remote control machines and robots. Lots of times machines that can be operated from a remote location are inaccurately called a robot because most of these devices can do nothing on their own. I call them tele-operated machines. However if the tele-operated device can do things on it's own and the operator just gives it directions once in a while it may be a robot.

Robot Overheads

At this point start showing the students overheads, pictures or videos of interesting or famous robots and mention something about each of them. Since I give the Organic Robot presentation often I am always looking for pictures of robots in science magazines, amateur robot books, amateur robot magazines and on the web. The reference section of my web page lists several good sites to start looking for robot information. One of my favorite robots to talk about is the "Raccoon Buster" described below.

Useful Amateur Robot

One of my overheads is of one of the first amateur robots I herd of that actually did something useful. The robot is called the "Raccoon Buster" and I found it in the October 1985 issue of "Robotics Age" magazine. The robot built by Mike Rigsby is made out of the first true robot toy made by Milton Bradley named the "Big Trak". The Big Trak can be programmed to move forward, reverse, turn right/left, and fire its blaster. Once programmed it will run the program entered every time the run button is pressed. It is a true robot by my definition. The overhead has the picture of the Big Trek with an automatic camera attached to it. . When focusing on something the camera turns it's lens. The author attached a lever to the lens and a switch to the camera. The switch is connected across the "Run" button on the Big Trek. When some thing comes in front of the camera, the camera focuses on the object by turning the lens, the lever attached to the lens closes the switch and the switch makes the connections across the run button starting the Big Trak's program. The robot is programmed to go forward two feet, fire it's blaster then back up two feet. Once programmed the robot is placed under the author's beach house in front of his garbage cans. When a raccoon heads for the trash cans the camera focuses on the raccoon, pressing the switch starting the robot's program. Certainly no self respecting raccoon would hang around a charging robot firing it's blasters.

If you have one, program a robot/robotic toy to run the same program and demonstrate running the program a couple of times reenacting the raccoon situation. Then change the program to demonstrate how you can make a robot do something different without rebuilding it. I was lucky a few years ago and found a Big Trak at a yard sale. I use it for the demonstration but any true robotic toy or amateur robot will work. Of course you can just act out this scenario if you do not have access to a robot.

If the presentation is divided up in three one-hour classes I use the rest of the first hour to demonstrate additional robots or show robotic video clips.

The Second Hour

During the second hour the Organic Robot is designed, the operating language is developed and the program is written.

Brainstorming to design robot.

Give the students the rules of brainstorming.

1. In brainstorming everyone says the first thing that comes to mind.
2. No one is allowed to comment that an ideal was bad or would not work.
3. You are allowed to add on to previous ideals.

Next the students are given the task their robot must perform. "Without assistance the robot has to be able to get out of a room with furniture in it. To help the robot navigate there will be a bright light in the hall outside the open door of the room."

Now the brainstorming session starts. As the students call out the parts of the robot, you draw the items on the board. Figure 1 shows the way I typically draw the robot parts on the board. Table 2 Lists the typical parts the students suggest during the robot design brainstorming session. Sometimes the students will need hints from you to get the brainstorming sessions started.

Wheels - drive
Wheels - caster
Drive motors
Drive trains - gears and axels
Batteries - Draw a box and write in "Power Source"
Computational unit/controller/computer
Communications interface
Wires
Light sensors - Suggest two so the robot can turn toward the light
Camera
Compass
Sound sensors
Distance sensors
Bumpers - suggest four: front, back, right and left
Speaker
Frame, Chassis or Body - Holds everything together
Blasters, Flame-Throwers and Guns! Just say, this is a peaceful robot.

Table 2 - List of the typical robot parts the students suggest during the robot design brainstorming session.

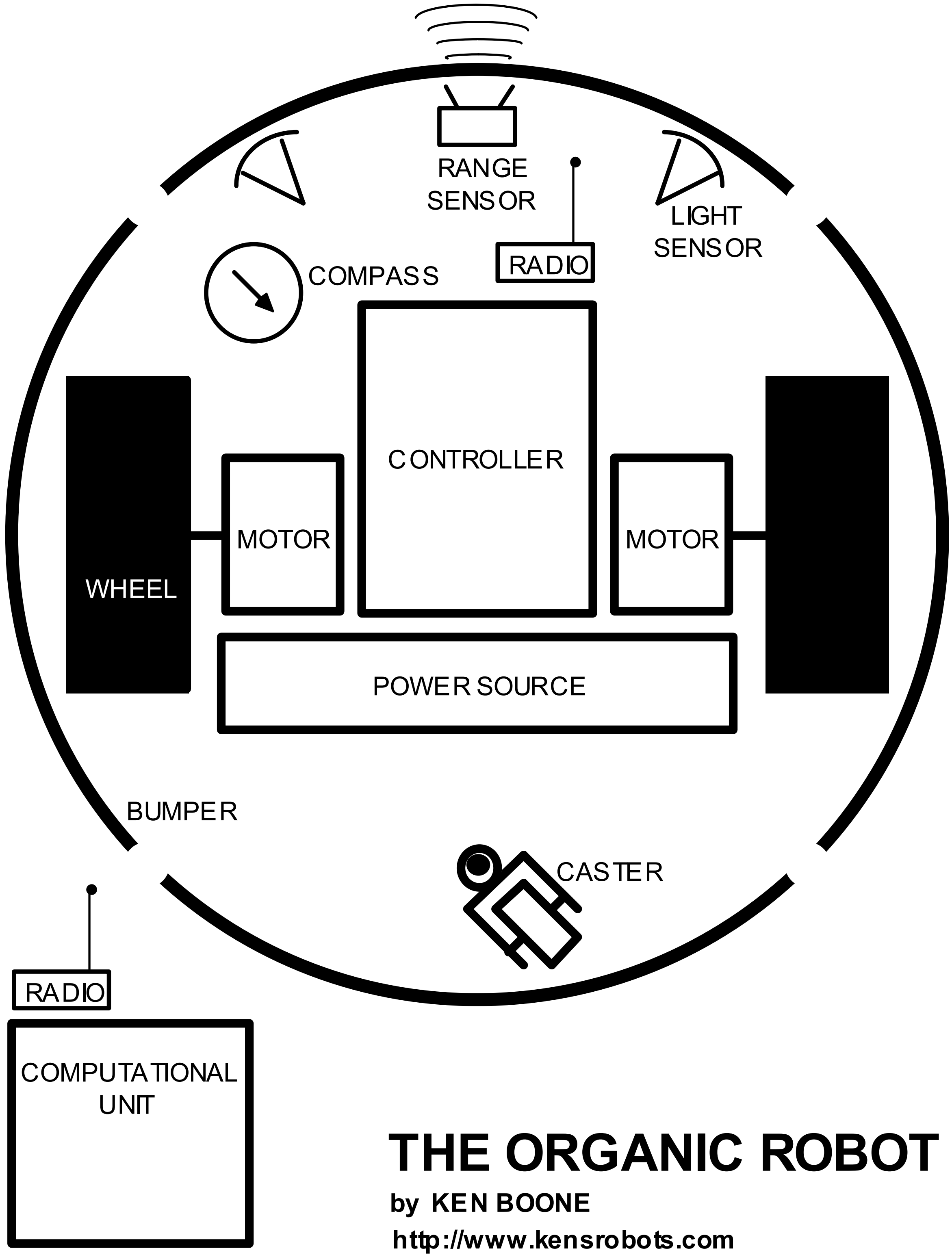


Figure 1 - As the students call out the robot parts draw them on the board as shown above. Remember to place the computational unit away from the robot. Tell them computational unit is too heavy for the robot to carry.

Three Rules of Robotics

When the students suggest adding weapons to the robot mention the "Three Rules of Robots". Writer Isaac Asimov proposed the three fundamental "Rules of Robotics" in his robot series "I, ROBOT" (You have to read it). The three rules are programmed into robots in his novels to keep the robots from hurting humans.

The rules are:

"1. A robot must not injure a human being, or, through inaction on its part, allow a human being to come to harm.
2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First and Second Laws."

Drawing the Robot and Suggestions

When the students first mention wheels quickly draw the two large drive wheels. Then mention something like: "Is this all it takes to make it move?" Usually you get motors at this point. Draw two motors and connect them to the drive wheels. Then mention how it is easier to move around if you have two drive motors and can turn in place. Grab the controls of the TAR Rat to demonstrate the moves a two wheeled robot can make; spinning in place with the wheels going in opposite directions, forward, reverse and sweeping turns by just running one motor. (If you do not have something like the TAR Rat them demonstrate the motions with an eraser on the black board.) If no more wheels are suggested, ask if the robot will fall over if it just has two wheels? If you get more wheels call them casters and mention they are like the wheels on the bottom of chairs that you can move in any direction.

When the students suggest either a camera or light sensor/detector I draw two light detectors and mention that this is a simple robot. It can only tell which light detector is detecting more light.

If bumpers are not mentioned tell the students you are a robot and walk against a desk or chair. Keep pretending you are walking and being held back by the chair. If no one suggests what the robot needs a touch or bumper sensor suggest it and draw four bumpers on the robot (front, back, right and left).

When someone suggests something to make it work, controller or computer draw a box on the robot and label it the "controller" and draw a box away from the robot and label it the "computational unit." Just explain that the computational unit is too big for the robot to carry. If it is not mentioned by the students draw wireless communications units on the robot and the computational unit so the controller and computational unit can communicate. The controller is the part of the robot that controls the motors and reads the sensors. The computational unit interprets the program and sends commands to the controller.

Of course the only sensors a simple robot needs to get out of the room are bumpers and the two light sensors. If any other sensors are suggested just draw them on the robot.

Brainstorming to design control language

Now that we have a robot and a computational unit we need a programming language. We can start out by asking the students if they have done any programming. Even with my youngest classes someone has programming experience. Discuss programming and mention the robot's programming language already has all of the usual BASIC programming language commands. We just need to add additional high level robot commands.

Start the robot control language brainstorming session. Ask the students to call out high-level control commands for their robot. Give an example of a high level command like, "Turn toward light." We are just coming up with high level commands right now. Write the robot commands on the board as the students call them out. Table 3 lists the high level commands the students usually suggest.

Start	**Turn CCW X degrees**
Stop Motion	**Turn toward light**
Forward till interrupt	**Turn north**
Reverse till interrupt	**Check Bumpers**
Forward X feet	**Check Range**
Reverse X feet	**Boogie**
Turn CW X degrees	**End**

Table 3 - This is the typical high level robotic commands the students suggest during the brainstorming session.

High Level Commands

The operation of each of each high level command is described below along with some suggestions on how to present them. If you are working with elementary students keep the definitions simple and do not mention the interrupt variable.

The "Start" command is placed in the computer program to indicate where the computer should start executing high level commands.

When forward is suggested ask if anything else is needed with the forward command. If they do not come up with "Forward Until Interrupt" then you simulate a robot running the forward command and walk forward running into things in the room without stopping. Say that computers have things called interrupts. When the computer gets an interrupt is stops what it is doing and determines what it should do next. When the computer is executing the "Forward till Interrupt" or "Reverse till Interrupt" commands the computer will stop the robot when a bumper is pressed, set the interrupt variable to 1 then return to the program to execute the next high level command. You do not have to mention the interrupt variable to younger students. The interrupt variable is useful in the more advanced programs written by older students.

If the "Forward X feet" and "Reverse X feet" commands are not suggested mention how useful it would be if the robot didn't have to go all the way across the room until runs into something. Mention that the computer stops the robot and returns to the program to execute the next high level command after moving the commanded distance. If a bumper is pressed before the commanded distance is traveled the computer stops the robot, sets the interrupt variable to 1and returns to the program to execute the next high level command.

The turn command also needs an amount to turn or the robot will not be able to stop turning. If the students just suggest "Turn" as a command, tell the students you are a robot and have them give you the turn command. Start turning and don't stop until they tell you to stop. Suggest "Turn CW X degrees" and "Turn CCW X degrees" commands. You may have to teach the students what clock wise and counter clockwise mean. Also mention that the robot is weak and can not turn if one of its bumpers is touching something. When the robot stops, after turning the commanded degrees, the computer returns to the program to execute the next high level command. If the bumper is pressed before the robot turns or is pressed while the robot is turning the computer stops the robot, sets the interrupt variable to 1and returns to the program to execute the next high level command.

When discussing the light sensors mention that vision systems are extremely expensive and require powerful computers to interpret the video signals. Since this is a simple robot it just uses two simple light brightness sensors. Each sensor is attached to the robot's computer and the computer determines which sensor is receiving the most light. When the "Turn toward light" command is executed the computer turns the robot toward the sensor receiving the most light. When both sensors receive the same light or a bumper is pressed the computer stops the robot and returns to the program to execute the next high level command. If the bumper is pressed before the robot turns to the light or a bumper is pressed while the robot is turning toward the light the computer stops the robot, sets the interrupt variable to 1and returns to the program to execute the next high level command.

If the students suggest a compass as a sensor tell the students the robot is very simple and it's compass can only determine if the robot is facing north. When the "Turn north" command is executed the computer turns the robot until the robot is facing north, then the computer returns to the program to execute the next high level command. If the bumper is pressed before the robot turns north or a bumper is pressed while the robot is turning north, the computer stops the robot, sets the interrupt variable to 1and returns to the program to execute the next high level command.

Explain to the students how the bumpers are going to operate. When the computer is executing any command that requires turning or moving the computer will stop the robot when a bumper is pressed. After the robot stops the computer sets the interrupt variable to 1 and returns to the program to execute the next high level command. When a new program command requiring motion is executed the computer will check to see if a bumper is touching before moving. The computer will only move the robot with a bumper touching in the following situations:

Move forward if the back bumper is touching.
Move reverse if the front, right or left bumpers are touching.

When the "Check Bumpers" command is executed the computer sets the "All Bumpers", "Front Bumper", "Rear Bumper", "Right Bumper", and "Left Bumper" variables values to "0" if none of the bumpers are touching. If a bumper is touching it's corresponding bumper variable is set to a value of "1". The "All Bumpers" variable will have a value greater than "0" if any bumper is touching. Once the bumper variables are set the computer will execute the next high level command in the program.

The computer stores the distance sensors reading in the "Range" variable when the "Check Range" command is executed. The distance sensor measures the distance from the robot to an object directly in front of the robot. After storing the range reading the computer returns to the program to execute the next high level command.

The "Boogie" command could be used to celebrate the robot's completion of its task. When the computer executes this command the robot will start making noises and turning back and forth for 4 seconds before returning to execute the next high level command in the program.

When the "End" command is executed the computer will stop executing commands and start beeping briefly every 10 seconds to indicate that the robot power is still on.

Flow Chart Program

When you finish the high level commands it is time to start flow charting the program. Explain how program flow charts are used to visually represent the paths taken in a program as the program executes. The flow chart is made up of process boxes, decision diamonds and lines with arrows to indicate the direction of program flow. Use the flow charts in figure 2 as an example of how to draw the charts on the board.

The process boxes represent actions being taken by the program. In our flow chart process boxes will contain high level commands from our robot programming language. To flow chart the robot turning north, moving forward two feet, then turning toward the light you would start out with a process box with "Turn North" in it. Next draw a short line with an arrow down from the north box to a box with "Forward 2'" in it. Then draw an arrow down from the forward box to a box with "Turn to Light" in it. The program starts at the top box and continues down following the lines and arrows.

The decision diamonds are used when program flow needs to take different paths based on variable or sensor data. A simple program for a robot to move forward until it is within two feet of a wall could be flow-charted in the following way. A "Start" process box is at the top. Below it is a "Check Range" box. Below it is a "Range < 2" decision diamond. Below the diamond is an 'End" box. Arrows go down from one symbol to the next. The line below the diamond box would have a "yes" written beside the line. The "yes" indicates the program path taken if the result of the test in the decision box is true. If the Range variable contains the value of 1 the program would flow straight down because the expression "Range < 2" is true. If the range variable contains a 10 the program would need to do something different at the decision diamond. Draw a short arrow line out the side of the diamond with a "no" above the line. Draw

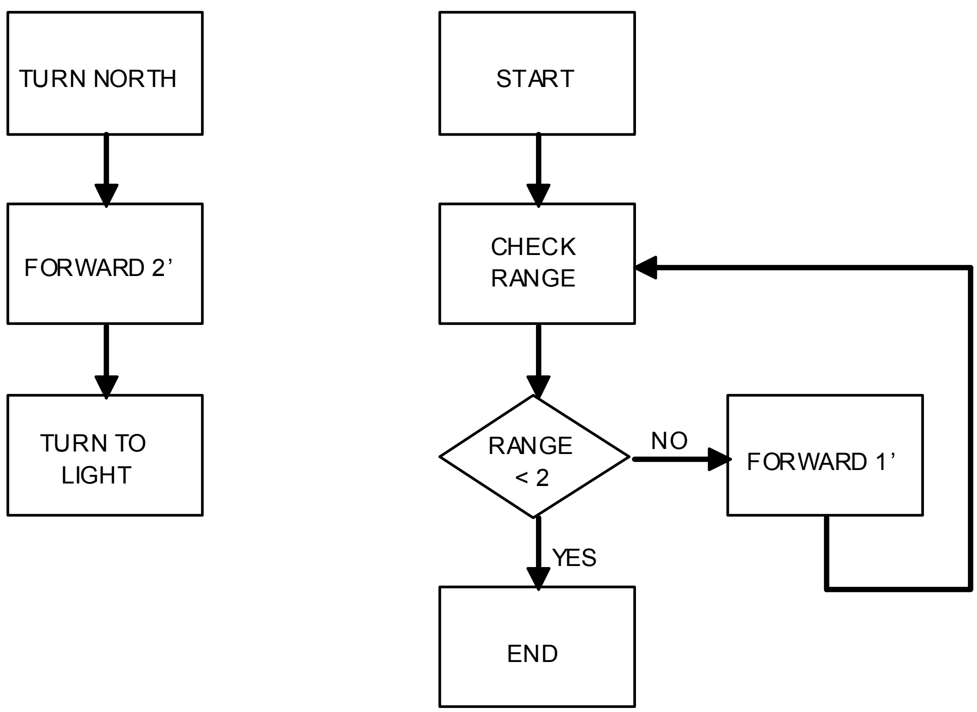


Figure 2 - Examples of how to draw flow charts on the board. Each example demonstrates a simple program that you can demonstrate to the students by pretending you are a robot and executing them.

a "Forward 1" box at the end of the no line beside the decision diamond. Then draw a line from the bottom of the Forward box up and around to the side of the Check Range box. Now the robot will move forward one foot at a time until it gets within two feet of the wall.

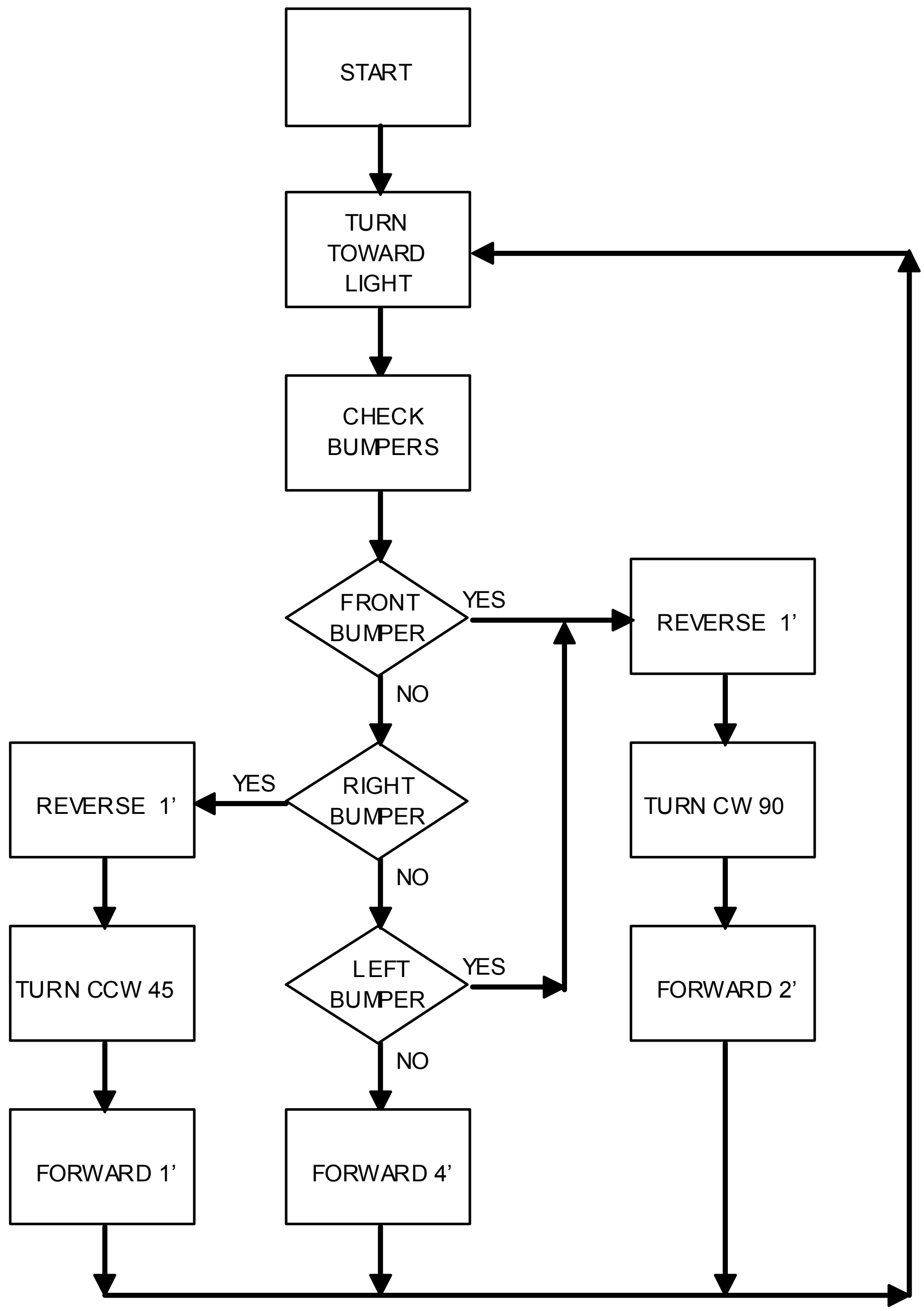


Figure 3 - Simple "Get out of the Room" Organic Robot flow chart program. Elementary grade students can write a program like this with some guidance from the presenter.

Get out of the room

It is difficult to keep a large or young class focused on making the "get out of the room" program flow chart. I recommend that the class be divided into equal groups of 5 to 8 students with an advisor helping each group that understands the robot commands and flow charting. Start another brainstorming session to flow chart the "get out of the room" program on the black board. If the students want to use a decision box, remind them that you have to read a sensor just before the decision box can use the sensor data. Flow chart part of the program with the whole class. Then have each group finish the program on their own. The advisors can suggest situations that the programs may not be able to handle and check to make sure the programs are flow charted correctly.

When programming the flow chart with older students or adults divide them into groups of four with each advisor covering two groups. Brainstorm the first five or so boxes then have each group finish their own program. With the older classes the advisors just answer question and check the flow charts for programming errors. Figure 3 is a simple "Get out of the Room" programming example.

The Third Hour:

During the third hour the robots are constructed and the students use the flow chart programs to control the robots

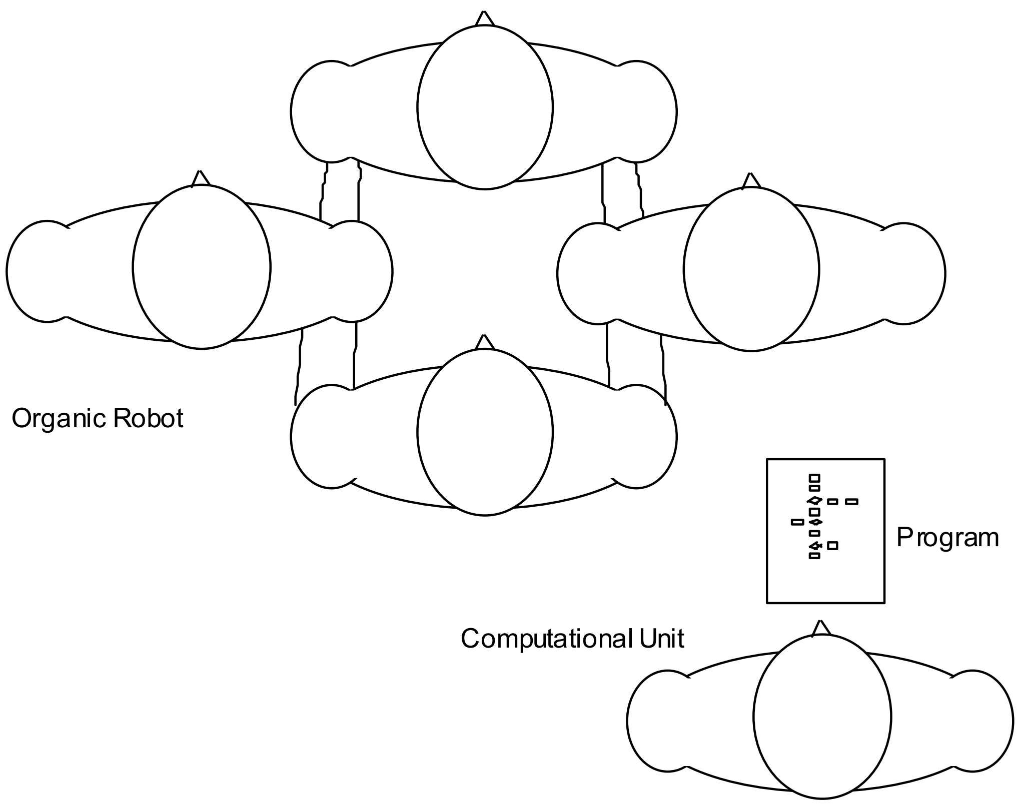


Figure 4: Assembled Organic Robot showing the positioning of the four students forming the moving part of the robot and the separate Computational Unit.

Assemble the Robot

The robot is constructed in two parts. Four students work together to form the mobile part of the robot and 1 to 4 students act as the computational unit to interpret the groups flow charted program. Ask four volunteers to come to the front of the class to demonstrate the robot assembly. Assemble the volunteers into the Organic Robot configuration shown in figure 4. Place one (front) student facing the class. It helps if this front student is the tallest of the group. Instruct the second (rear) student to stand behind the front student and place both hands on the front student's waist. The rear student's arms must remain straight. The third (right) student stands on the right side of the robot facing the class. The left arm of the right student goes over and locks around the extended right arm of the back student. The fourth student (left student) stands on the left side of the robot and locks on to the left arm of the back student.

Explain to the class the function of each student. Each of the students is the power source, the wheels, and the corresponding bumper. The front student's head is covered with a paper bag and can only look straight down to read the sensors or determine which direction to turn to face the light. The front student is also the local controller of the robot and directs the robot as it executes each high level command sent to the robot from the computational unit. The right, left and rear students wear blindfolds. Either the right or left student can operate the communications device. Have the class give the volunteer commands to demonstrate how it operates.

Now that the class knows how the robots will be built and operated, have each group assign a student to each part of the robot. If there is enough room build two robots at a time. Multiple robots operating together make the experiment more interesting.

Set up the simulation

Now that the robots are assembled in the classroom it is time to set up the simulation. Make a drawing of the classroom so the computational units and the robots will be able to see how they moved around in the room. Locate the computational unit students and their programs in the hall where they can not see into the classroom. Move the robots around in the room so they do not know how they are oriented and move some of the classroom furniture around. If you have two way hand held radios place one in the hall with the computational units and one with the robots. If you do not have radios then just stand in the doorway and pass the computational units commands from the hall to the robots and the replies from the robots to the computational units.

Operate the robots

Each computational unit takes turns sending one command and waiting for a reply from their robot. To start the first computational unit sends the first command of their program to their robot. Their robot executes the command and replies with "ready" if the robot completed the command successfully, the value of the sensor variable requested or "interrupt" if a bumper stopped the robots motion. The path of the robot is plotted on the drawing of the classroom then the second computational unit sends a command. Encourage the computational units to follow their programs as written as they take turns.

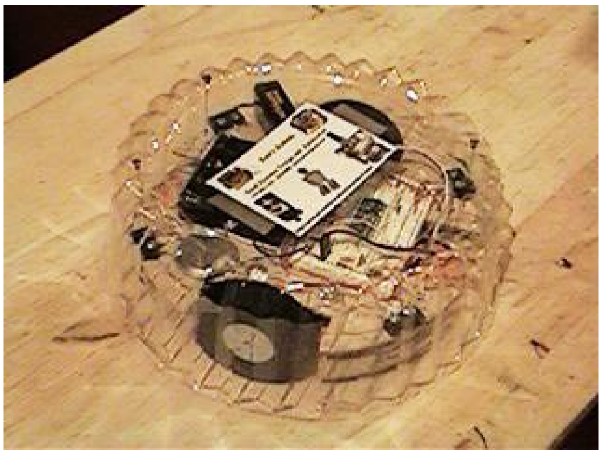
When one of the robots makes it through the door or both robots become trapped the simulation is over. Go over the paths the robots took with the groups and discuss what changes would improve the operation of the robots and programs.

Conclusion

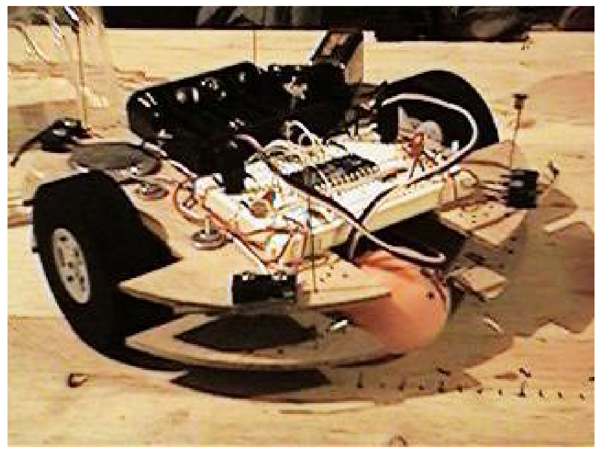
The Organic Robot presentation gives students a chance to experience the limitations of today's robots. They see how robots sense their surroundings and how programs control robots. Most importantly the students have a great time. If you have any suggestions to improve the Organic Robot presentation or some good stories please email them to me.

Organic Robot Simulator

After all of the Organic Robots have completed their runs I transfer the most successful flow chart program to my Organic Robot Simulator (ORS). This electronic version of the Organic Robot as shown in photos 3 and 4 has the same high level commands as the Organic Robot. I designed the ORS for a robotics class I developed and teach to gifted high school students. The students are in the class 7 hours a day for 18 days. The first 7 days each student learns how build robots from raw materials by making their own copy of the ORC. Once the robots are complete the students are divided into teams and each team spends the rest of the class designing an International Fire Fighting Robot. The fire fighting robots compete with each other at the end of the class. Hopefully I will have a construction article for the ORS in next year's robot issue.



Picture 3 - The Organic Robot Simulator with its cover on.



Picture 4 - internals of the ORS

REFERENCES

Trinity Collage International Fire Fighting Contest; http://www.trincoll.edu/~robot

Isaac Asimov, "I, ROBOT", Bantam Books reprinted 12, 1991; ISBN 0-553-29438-5

Mike Rigsby, article "Raccoon Buster"; "Robotics Age" magazine, October 1985

Ken Boone is Information Systems Specialist II with the telecommunications group at Lord Corporation's World Headquarters campus. He has been building amateur robots since 1963 and his fire fighting robots placed 8th and 13th respectively the last two years at the Trinity contest. He has been teaching robotics to gifted students for 20 years. Information about his robots can be found at http://www.kensrobots.com. *He earned a BS in Industrial Arts Education at North Carolina State University. He holds a control theory patent. You can reach him at kensrobots@aol.com.*

Last revised 3/15/2001