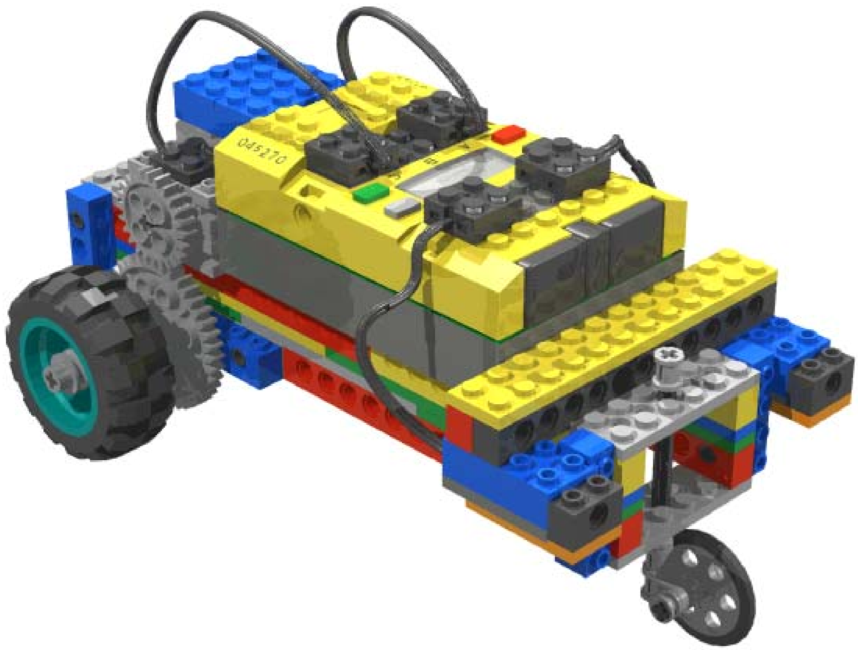
# *Instructions For Constructing A Braitenberg Vehicle 2 Robot From LEGO Mindstorms Components*



**Michael R.W. Dawson,**
**Biological Computation Project,**
**University of Alberta,**
**Edmonton, Alberta**
**September, 2003**

**This is a draft, please do not quote**.

© Michael R.W. Dawson 

## *Introduction*

The purpose of this document is to describe how to construct a particular robot, which we will call Braitenberg Vehicle 2, out of Lego Mindstorms or Lego Dacta components.

This particular robot was inspired by a thought experiment described by neuroscientist Valentino Braitenberg in his classic book *Vehicles* (Braitenberg, 1984). In the original description of this kind of system, the agent was a system that propelled itself under water. It had two separate engines, one on each side, and two separate sensors (e.g., for measuring temperature), again on each side of the agent. The output of one sensor was used to control one motor, and the output of the other sensor was used to control the other motor. In particular, motor speed was directly proportional to the value detected by a sensor, so that when this value increased, the motor sped up, and when this value decreased, the motor slowed down. Braitenberg argued that this simple system would generate complicated behavior if it were embodied and situated in the world. Further more, the kind of behavior that it generated would depend upon whether a sensor was attached to the motor on the same side of the system or to the motor on the other side.

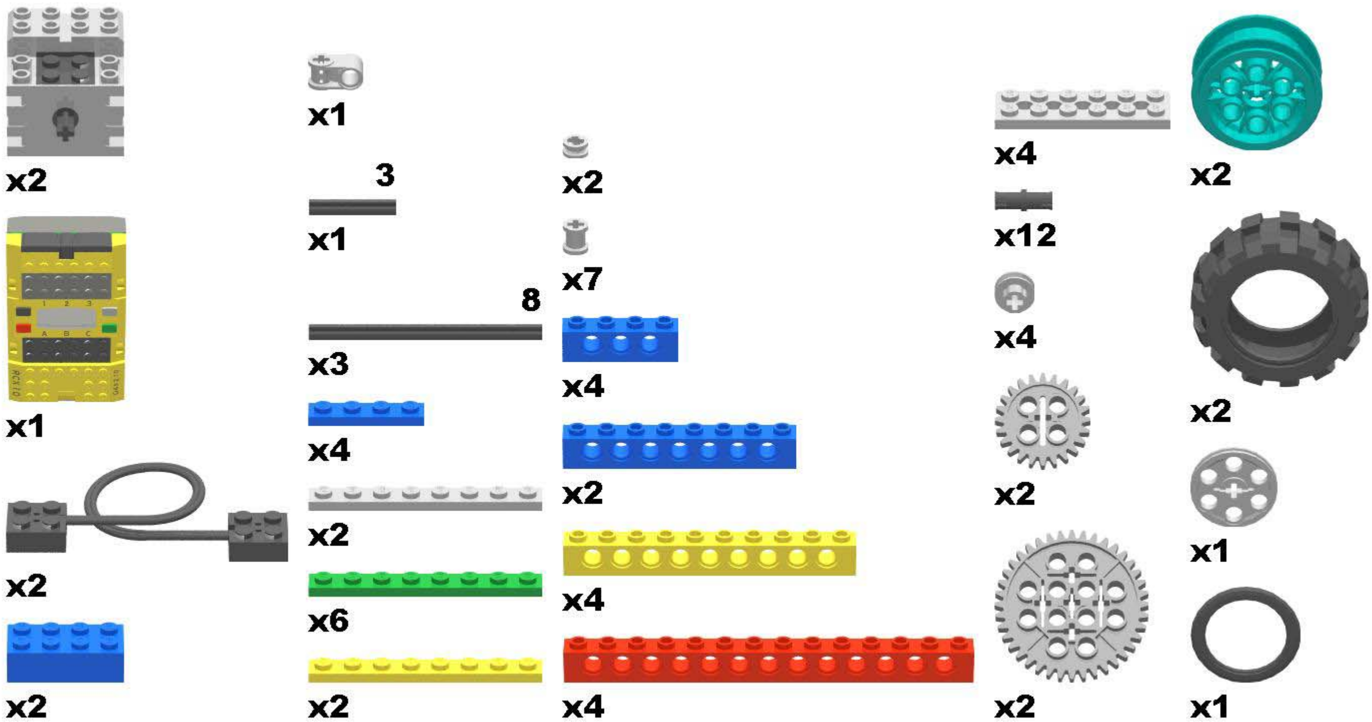
In this particular incarnation of the model, it is not a system that swims, but is instead a “tractor like” robot that is designed to move around fairly flat surfaces. It is constructed from parts in a LEGO Dacta kit that was purchased for the Biological Computation Project at the University of Alberta; these parts are identical to those that are easily found in LEGO Mindstorms kits. The instructions that are provided below for building the robot describe the third generation of this robot. This generation of the robot differs from earlier generations in terms of incorporating a few more of the principles of LEGO design that are intended to create stronger robots (Martin, 1995). The images that are provided below to show how the robot was constructed were created using a suite of LEGO CAD programs (in particular, MLCAD, L3P, POV-RAY, and LPUB) (Clague, Agullo, & Hassing, 2002). The use of plates and beams to restrict the sight line of the light sensors can be found elsewhere (Baum, 2003).

Because the robot is constructed from LEGO components, the relationship between sensor values and motor behaviors is not exactly as Braitenberg intended. The 9 volt LEGO motor that was used has very low torque, and the 5-to-3 gear reduction that we employ does not really help this much. As a result, modifying motor speed does not really affect motor behavior all that much. Instead, the NQC program that is provided below keeps the motors at the same speed, but sets the duration that the motor runs to different lengths of time. The higher the sensor value, the longer the motor runs, which is functionally equivalent to adjusting motor speed in accordance with Braitenberg’s original thought experiment. NQC is available from the following web page: http://www.baumfamily.org/nqc/.

At this time of writing, we have not compiled video footage of the behavior of this generation of Braitenberg Vehicle 2. However, several examples of the behavior of this machine can be viewed at the following web page: http://www.bcp.psych.ualberta.ca/~mike/Book2/Robots/Vehicles/index.html. By the end of the fall of 2003, it is hoped that some new videos of the current generation of robots will be available.

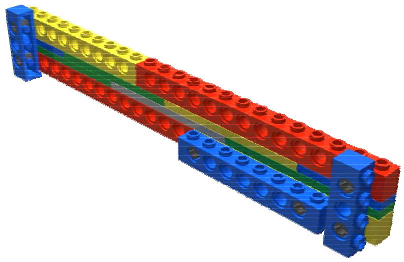
## *LEGO Parts Required*

The image below provides an indication of the parts that are required to construct this robot. When the robot was created, care was taken to ensure that only LEGO parts were used. However, one reason that this document is still in draft form is because I haven't checked to ensure that this parts list is complete.

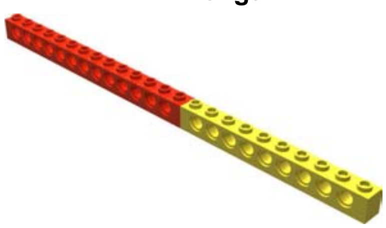


## *Stage 1 – Build The Left Side*

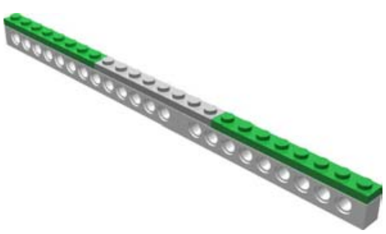
The first stage in constructing this robot is to build the left side of fits chassis, which is illustrated below. The constructed side can be considered to be a beam that is 24 LEGO studs long. The beam is reinforced by two 4-stud beams attached to either end, in accordance with Martin's paper The art of LEGO design. An 8-stud beam is attached to the side to provide support for an axle that is attached later.



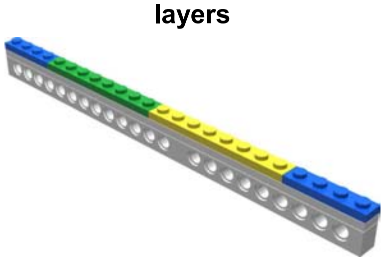
**Step 1: Lay two beams end to end to create a 1 X 24 length**



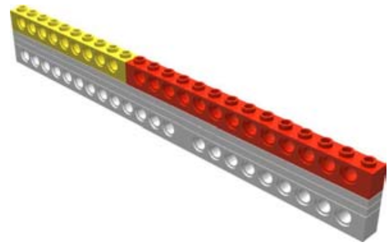
**Step 2: Attach a layer of plates to the beams to hold them together. Don't overlap a seam between plates with the seam between the two beams.**



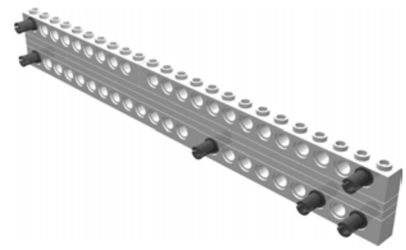
**Step 3: Attach another layer of plates to the previous layer, once again avoiding the overlapping of seams between adjacent layers**



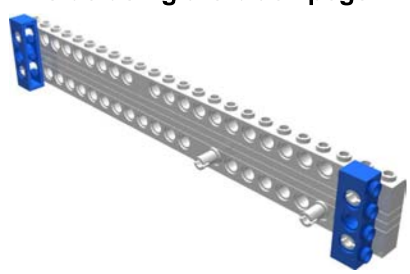
**Step 4: Attach a final layer of beams to the top layer of plates.**



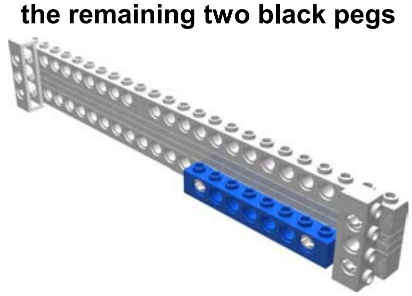
**Step 5: Insert 6 black pegs to be used to attach shorter beams to the side**



**Step 6: Attach a 1X4 beam to each end of the side using the black pegs**



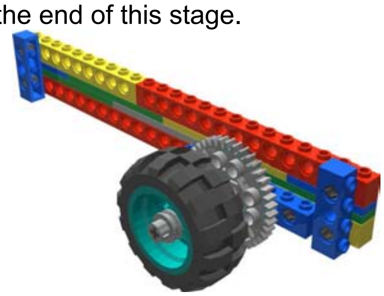
**Step 7: Attach a 1X8 beam to the side using the remaining two black pegs**



When this last step is carried out, Stage 1 is complete. The next stage involves building a wheel and axle assembly that will be attached to this side.

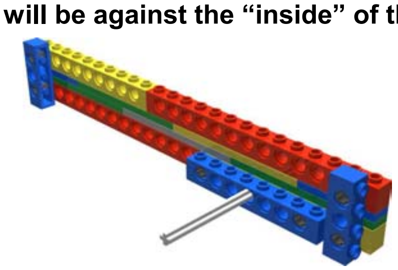
## *Stage 2: Build The Left Rear Axle*

An axle assembly needs to be added to the left side that has just been constructed. The axle will be able to rotate freely in the side. A wheel will be attached to the axle (to propel the robot), and a 40-tooth gear will be attached as well (to be rotated by a motor that will later be mounted on top of the robot). Other attachments to the axle will keep everything fixed in place so that there will be no trouble meshing the axle's gear with the gear that will be attached to the motor. The figure below illustrates how things should look at the end of this stage.

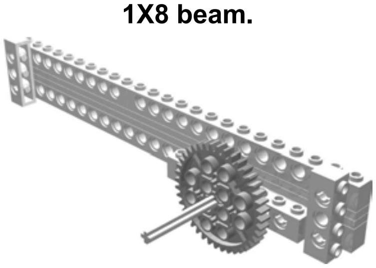


**Step1: Take an 8-stud axle, and attach an offset pulley to one end.**

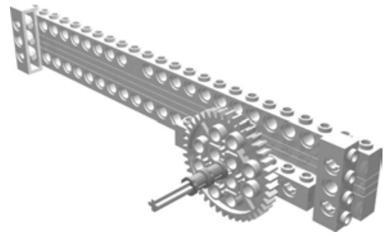
**Step 2: Insert this axle through the middle hole of the 1X8 beam attached to the left side that was built. The free end of the axle will come through the beam, and the offset pulley will be against the "inside" of the side.**



**Step 3: Place a 40-tooth gear onto the free end of the axle, and push it up against the 1X8 beam.**



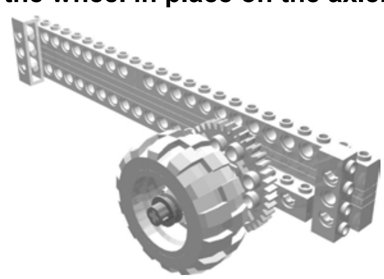
**Step 4: Place two bushes on the axle, adjacent to one another, and against the gear. They will hold the gear in place.**



**Step 5: Place a 20X30 wheel hub on the axle, pushing it against the bush.**

**Step 6: Add a 20X30 balloon tire to the hub on the axle.**

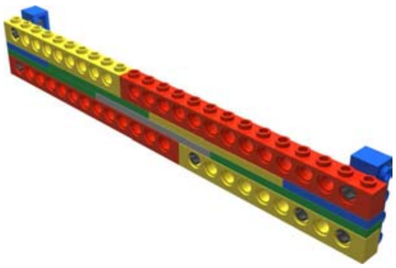
**Step 7: Attach another offset pulley to keep the wheel in place on the axle.**



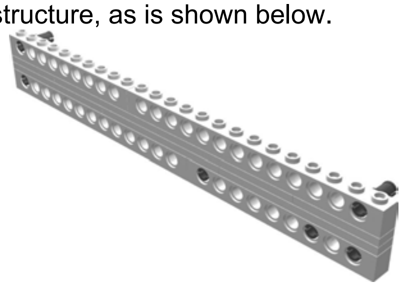
At this point, the left side of the robot chassis is now complete. The next stage is to build a mirror image of this assembly that will serve as the right side of the robot chassis.

## *Stage 3: Build The Right Side*

The right side of the robot chassis is constructed in exactly the same manner as was the left side during Stage 1. The only difference is to ensure that this side is a mirror image of the left side, reflected across the length of the 24-stud structure. The completed right side is illustrated below.

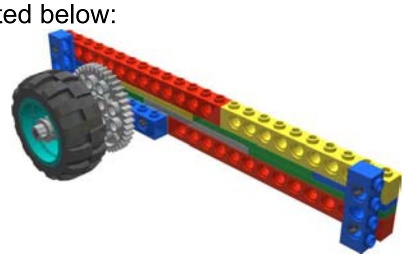


The mirror image nature of this side, in comparison with the left, is determined by how the black pegs are placed into the long beam that has been constructed. In particular, the pegs go in the same holes as were used previously, but the pegs poke out the other side of the structure, as is shown below.



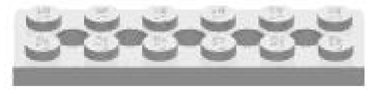
## *Stage 4: Build The Right Rear Axle*

Once the right side has been constructed, a rear axle assembly can be added to it by following the same instructions that were presented in Stage 2. The desired "mirror image" structure is illustrated below:

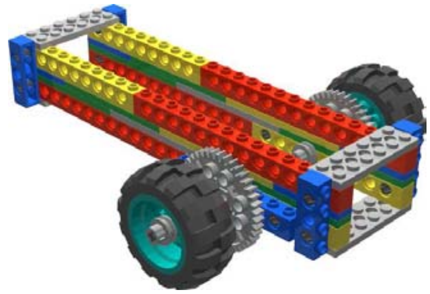


## *Stage 5: Complete The Chassis*

The chassis can now be completed by joining together the two sides using 2X6 plates with holes, such as the piece that is illustrated below. The wider plates are used (instead of 1X6 plates) to keep the chassis from bending, and as a result to keep the axles from binding. Plates with holes (instead of solid plates) are used to permit vertically oriented axles to be added to the chassis in later stages.



When constructed, the chassis will appear as illustrated below. Note that four plates are used, one on the top and one on the bottom of the front of the chassis, as well as one on the top and one on the bottom of the back of the chassis.



## *Stage 6: Build The Robot's Motor Assembly*

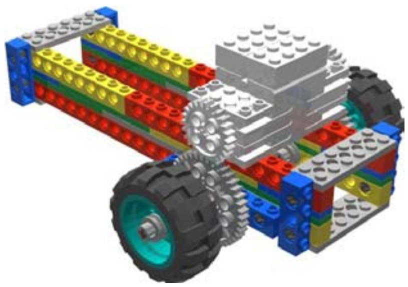
The chassis at this point has two rear wheels attached to it, with each wheel attached to its own axle. In this stage of assembly, two motors are added to the chassis so that each will power its own wheel. The RCX brick is added to control the motors. Both the addition of the motors and of the RCX will provide further reinforcement to the robot chassis.

**Step 1: Take two 9V motors, and place them end to end. Place a 24 tooth gear on each of the motors.**

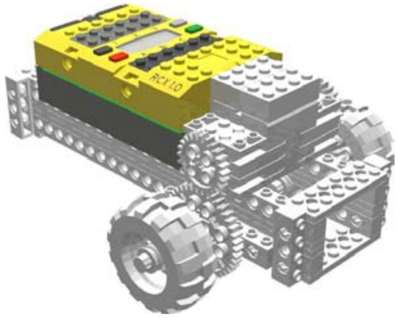
**Step 2: Take two 2X8 bricks, and place them across the seam of the two motors to connect the two motors together into a single assembly.**

**Step 3: Attach the motor assembly to the robot chassis, pressing it down to connect to the two long beams that make up the sides of the robot. Place the motors in such a way that the gears on the motors mesh with the larger gears on the wheel axles. There**

**should be two free studs on the robot chassis behind the motors when this step is completed.**



**Step 4: Attach the RCX brick directly to the chassis directly in front of the two motors. There should be two free studs on the sides of the robot in front of the RCX brick.**



## *Stage 7: Build The Front Axle*

The general design of the robot chassis is like a tricycle. The two wheels at the back of the robot will be used to drive it forwards. A third wheel is required at the front of the robot. This third wheel is passive, in the sense that it is not driven by a motor. It is designed to move freely to the left and to the right, providing the ability for the robot to turn. However, this front wheel is not used to steer. It simply responds to forces produced by the rear motors. The robot will turn when the two motors are not running at exactly the same speed.

In this stage of constructing the robot, the front wheel assembly is built. It will be connected to the robot chassis in Stage 7.

**Step 1: Take an 8-stud length axle, and attach to one end of it a perpendicular axle joiner.**

**Step 2: Take two bushes, and place them side by side on the axle, so that the lower of the two is in direct contact with the perpendicular axle joiner.**

**Step 3: Insert a 3-stud length axle horizontally through the perpendicular axle joiner.**

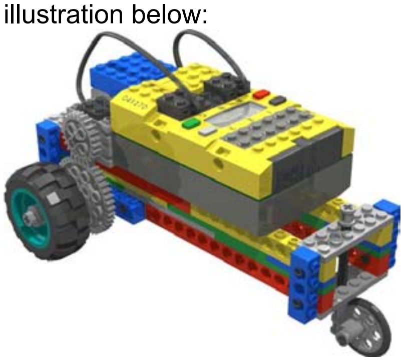
**Step 4: Attach a half bush to one end of the shorter axle.**

**Step 5: Attach a Technic wedge belt wheel to the other side of the short axle, and add a wedge belt tire to this assembly.**

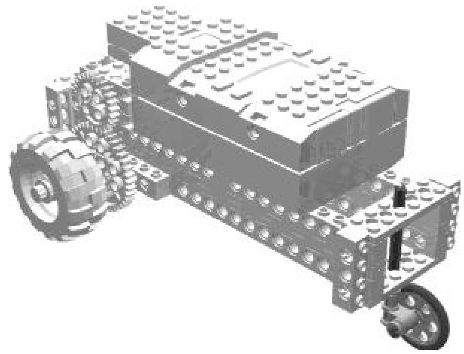
**Step 6: Attach a half bush to the other end of the short axle to prevent the wedge belt wheel from falling off. Make sure that the two half bushes on the short axle are not pressed in too tight to prevent the wheel from spinning fairly easily.**

## *Stage 8: Attach The Front Axle To The Robot Chassis*

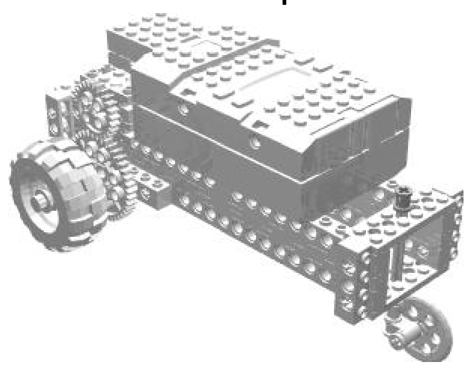
In this stage, the front wheel is added to the chassis. As well, wires are used to connect RCX output ports to the two motors. At the end of this stage, the robot should appear as in the illustration below:



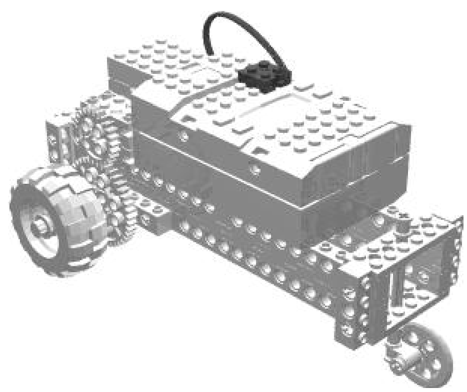
**Step 1: Take the complete chassis that was constructed in Stage 6. Take the wheel assembly that was constructed in Stage 7. Pass the long, vertical axle of the wheel assembly through the middle hole of both of the 2X6 plates that are at the front of the chassis.**



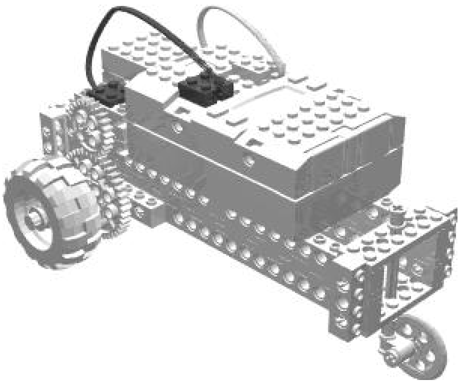
**Step 2: Attach a bush to the end of the vertical axle that is sticking through the top plate. Make sure that it is not on so tight that the vertical axle cannot freely rotate in the two plates.**



**Step 3: Attach the electric plate of a short electrical cable to Output Port A on the RCX. Attach the plate at the other end of the cable to the left motor of the robot.**

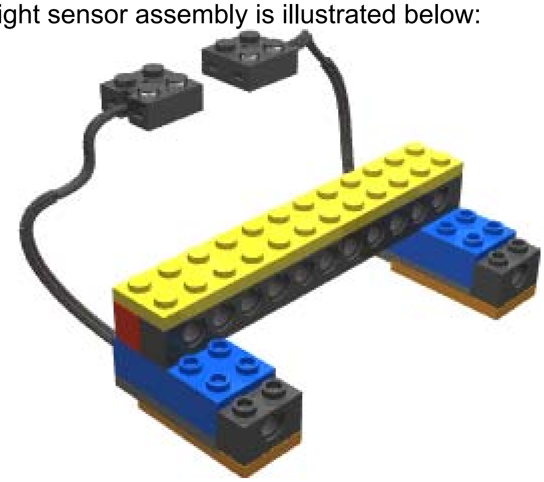


**Step 4: Use another short electric cable to connect RCX Output Port C to the motor on the robot's right. The orientation of this connection will have to be adjusted to ensure that the two motors spin the wheels in the same direction. Conversely, the direction that the motors spin will have to be set in the software that runs the robot.**

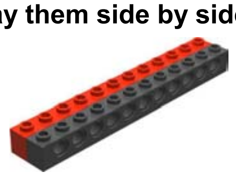


## *Stage 9: Construct The Light Sensor Array*

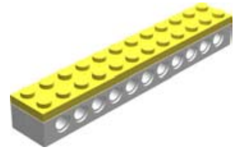
The next stage of construction involves creating an assembly that will hold two light sensors whose output will be used to determine how the two robot motors will be run. The completed light sensor assembly is illustrated below:



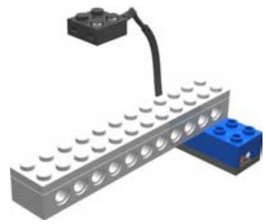
**Step 1: Take two 2X12 Technic beams and lay them side by side.**



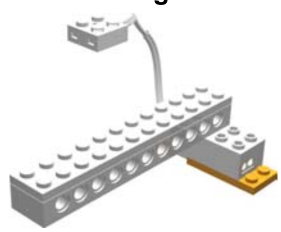
**Step 2: Use a 2X12 plate to attach the two beams together.**



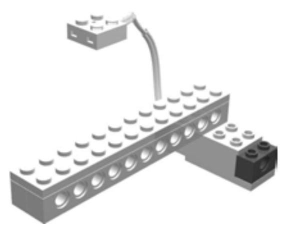
**Step 3: Take a light sensor, and attach it to the underside of the two beams, connecting it to the far end of the beam assembly. There should be four free studs at the front of the light sensor.**



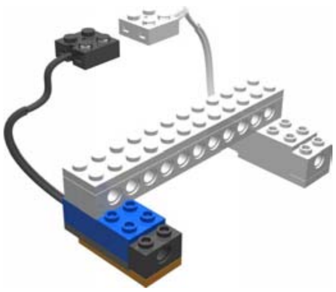
**Step 4: Take a 2X8 plate and attach it to the bottom of the light sensor. Place it in such a way that it has two free studs available in front of the light sensor.**



**Step 5: Take a 1X2 Technic beam (with a hole through the middle) and attach it to the two free studs in front of the light sensor. This will help limit the field of view of the sensor.**



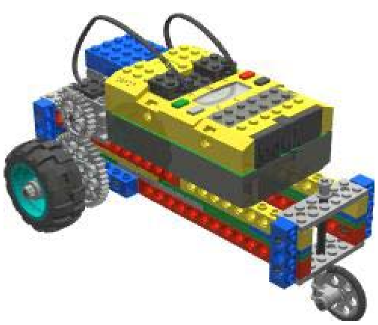
**Step 6: Repeat Steps 3, 4, and 5 with a second light sensor, 2X8 plate, and 1X2 beam to create a second light sensor assembly. Attach it to the other end of the two long beams.**



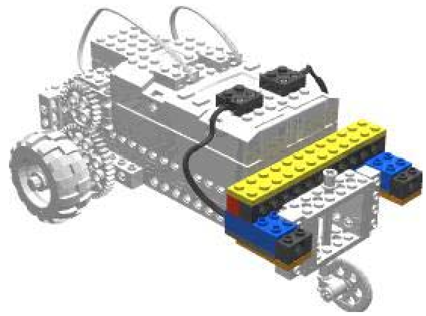
## *Stage 10: Attach the light sensors to the robot chassis*

The final stage of robot construction involves attaching the light sensor assembly to the front of the robot chassis, and connecting the light sensors to two of the input ports on the RCX. When this stage has been completed, the robot construction is finished. The next stage of exploring this robot would be to program it in such a way that each of the motors responds in a way that depends on the amount of light that is being detected by one of the light sensors.

**Step 1: Take the robot chassis in the form that was achieved at the end of Stage 8. Note the two free studs on each beam in front of the RCX, and behind the top plate at the front of the chassis.**



**Step 2: Take the light sensor assembly that was constructed in Stage 9. Press it down to connect it to the front of the robot chassis, using the free studs in front of the RCX. Then connect one of the light sensors to Input Port 1 on the RCX, and the other light sensor to Input Port 3.**



The next step is to program the robot that has been constructed. An example program is provided on the next page of these instructions.-

# Software

Now that the robot has been constructed, a program must be written and downloaded to the RCX brick to control the motors on the basis of the light values detected by the two sensors. Here is an example program written in NQC:

```
/* Michael R.W. Dawson -- vehicle2 code -- freeware -- July 9, 2003 */

/* global variable definitions here */
int TIME1,TIME3;
#define PAUSETIME 2 /* determines how long system waits */
#define RUNTIME 50  /* determines how long system runs */

/* This code is NQC stuff for creating a robot that is similar to
   Braitenberg vehicle 2.  It uses the amount of light measured by
   a sensor to determine how to run a motor.  Because the Lego motors
   are low in torque, instead of setting motor speed on the basis of a
   light sensor, the light sensor determines how long a time that a
   motor will be turned on. */

/* This routine runs one motor for a time determined by the difference between sensors*/
task motor1() {
while (true)
 {
        /* Phase 1 -- forward motion, run the motor for a period of time */

        Float(OUT_A);  /* Float motor A (stop it, but don't put on the brake) */
        OnFwd(OUT_A);  /* Turn motor A on in the forward direction */
        Wait(RUNTIME); /* Wait for the runtime amount of time */
        Float(OUT_A);  /* Float the motor at the end of running it */

        /* Phase 2 use sensors to stop or run motor for a variable length of time */

        TIME1 = SENSOR_3 - SENSOR_1;  /* compute a time using the difference between sensors */
        if (TIME1 < 0)
                {
                        TIME1 = TIME1 * -1; /* make the time positive */
                        Float(OUT_A);           /* float the motor */
                        Wait(TIME1 * PAUSETIME); /* float it for this amount of time */
                }
        else
                {
                        OnFwd(OUT_A);     /* run the motor forward */
                        Wait(TIME1 * PAUSETIME);  /* do this for the computed time */
                }
 }
}

/* This routine runs the other motor for a time determined by the other light sensor.  It's code
is nearly identical to the code above, so I haven't bothered to provide detailed
comments on it.*/

task motor2() {
while (true)
 {
        Float(OUT_C);  /* note that this works with motor C */
        OnFwd(OUT_C);
        Wait(RUNTIME);
        Float(OUT_C);
        TIME3 = SENSOR_1 - SENSOR_3;  /* compare to task motor 1 */
        if (TIME3 < 0)
                {
                        TIME3 = TIME3 * -1;    /* TIME3 will be different than TIME1 -- vehicle will therfore turn */
                        Float(OUT_C);
                        Wait(TIME3 * PAUSETIME);
                }
        else
                {
                        OnFwd(OUT_C);
                        Wait(TIME3 * PAUSETIME);
                }
```

```
  }
}


task main(){

  SetSensorType(SENSOR_1, SENSOR_TYPE_LIGHT); /* define as light sensor */
  SetSensorType(SENSOR_3, SENSOR_TYPE_LIGHT); /* ditto */
  SetSensorMode(SENSOR_1, SENSOR_MODE_RAW); /* raw mode works best! */
  SetSensorMode(SENSOR_3, SENSOR_MODE_RAW); /* ditto */
  SetPower(OUT_A + OUT_C, 4);        /* motors run at speed = 4 */
  SelectDisplay (DISPLAY_SENSOR_1);        /* display sensor 1 for debugging behaviour */

start motor1;                /* start motor 1 task */
start motor2;                /* start motor 2 task */

/* the two above tasks will drive the two motors (at different "speeds") indefinitely
   until the robot is stopped or turned off */

}
```

## *References*


Baum, D. (2003). *Definitive Guide To LEGO Mindstorms*. New York, NY: Apress.

Braitenberg, V. (1984). *Vehicles: Explorations In Synthetic Psychology.* Cambridge, MA: MIT Press.

Clague, K., Agullo, M., & Hassing, L. C. (2002). *LEGO Software Power Tools*. Rockland, MA: Syngress Publishing.

Martin, F. G. (1995). The art of LEGO design. *The Robotics Practitioner: The Journal For Robot Builders*, *1(2).*