Descartes Assembly Instructions

Version 1.1

Contents

For the latest assembly instructions, robot projects and programs visit the download section of our web site at www.divent.com.

These instructions contain images with over 256 colors. For the best image quality, set your display to the highest color depth available. HiColor or TrueColor is preferred (16 or 32-bit). Contact us at support.divent.com concerning comments about your Descartes kit or to join our free newsletter subscription list.

Diversified Enterprises
158 Aero Camino Rd.
Santa Barbara, CA 93117

Descartes Assembly Instructions

Before You Begin

Congratulations on your purchase of a Diversified Enterprises Descartes miniature robot. This robot is intricate well engineered, and a joy to assemble. Before you begin, lay all the parts out on a clean table and check them against the Parts List to make sure you have everything you need. Inspect all of the parts for possible damage during shipment. Collect the necessary tools and supplies listed in the Tools You Will Need section. Be sure your soldering iron has a narrow, fine tip, and keep it well cleaned and tinned with solder. A damp natural sponge works well as a tip cleaner while the iron is hot. The solder you use should not be too large in diameter, or some soldering may be difficult to accomplish due to Descartes' high component density. Read through the entire assembly instructions to the end, and study all of the pictures before beginning.

Note: Due to part source availability and advances in engineering, the components in the images may differ in appearance from the ones in your kit. When comparing the parts you received against the parts list, a magnifying glass or loupe may prove helpful.

Enjoy!

Back to the Table of Contents

Diversified Enterprises
158 Aero Camino Rd.
Santa Barbara, CA 93117

Descartes Assembly Instructions

Parts List

PC Board soldered components

```
Note: some components may already be mounted to the PCB
Components are listed by their PCB locations

Resistors:
R1,2,17                  220-ohm (red, red, brown)
R3,4,9,10,15,16,19       33K ohm (orange, orange, orange)
R5,6,7,8                 2.2K ohm (red, red, red)
R11,12,13,14             150 ohm (brown, green, brown)
R18,P7,P13               10K ohm (brown, black, orange)

Capacitors:
C1,2                     100,000pF (small ceramic)
C3                       10uF 6.3V (electrolytic can)
C4                       100uF 10V (electrolytic can)
C5                       4.7uF 6V (4u7, dipped tantalum)

Sockets:
IC1                      24-pin socket
IC2                      18-pin socket
IC3                      16-pin socket

ICs:
IC1                      BS2-IC computer
IC2                      PIC16C71 computer
IC3                      L293D motor driver
VR1                      7805 +5V voltage regulator

Headers:
EXP1                     2 x 10-pin female SIP strip
GNDs and AD1-4s          Single pins from female SIP strip
J1                       2-pin male SIP
                         2-pin female jumper
Comm.Port                4-pin male SIP

Other:
PI1, PI2                 Four-pin dual-windowed encoder
CR1                      10MHz 3-pin ceramic resonator
SPK1                     Speaker
SW1                      Large DPDT slide switch
                         Battery snap connector
SW2                      Micro tact pushbutton switch
LED1,2,3,4               High-intensity LED
```

Other components

1	Printed circuit board (PCB)
2	Motors
4	Nylon ties
1	6 x AA battery pack
1	4-inch velcro strip
2	Drive wheels
2	Caster gear wheels
2	Encoder stickers
2	Belts
2	Tires
2	Wheel retainer clips
4	6 x 32 Phillips screws
4	6 x 32 nuts
2	Lengths narrow wire
1	Length thick wire
1	Length heat shrink tubing
4	CdS light sensors
1	Thermistor thermal sensor (black with blue top)
1	Length red wire
1	Length black wire
1	Floppy disk
1	Paper wire bending template

Download cable parts:

1	DB9 9-pin serial port connector
1	DB9 hood with hardware
1	5 foot length 4-conductor cable
1	4-pin female plug
4	Female plug crimp pins

Back to the Table of Contents

Diversified Enterprises
158 Aero Camino Rd.
Santa Barbara, CA 93117

Descartes Assembly Instructions

Tools You Will Need

Fine point soldering iron and rosin core (electronic) solder
Diagonal wire cutters
2 Pair needle nose pliers
Philips screwdriver
Scissors or hobby knife
Hair dryer or other heat source
Desoldering braid
Rubbing alcohol and cloth or old toothbrush (preferred)
Sandpaper, medium grit
Black marker pen (Sharpie works well)
Spring-type clothespin or similar clamp
Vaseline or silicone grease (optional)
Paper punch

Back to the Table of Contents

Diversified Enterprises
158 Aero Camino Rd.
Santa Barbara, CA 93117

Descartes Assembly Instructions

Section 1: The PCB Components

The first section in the assembly process is to solder the electrical components to the Printed Circuit Board (PCB.) Remember to read through each step completely before executing it.

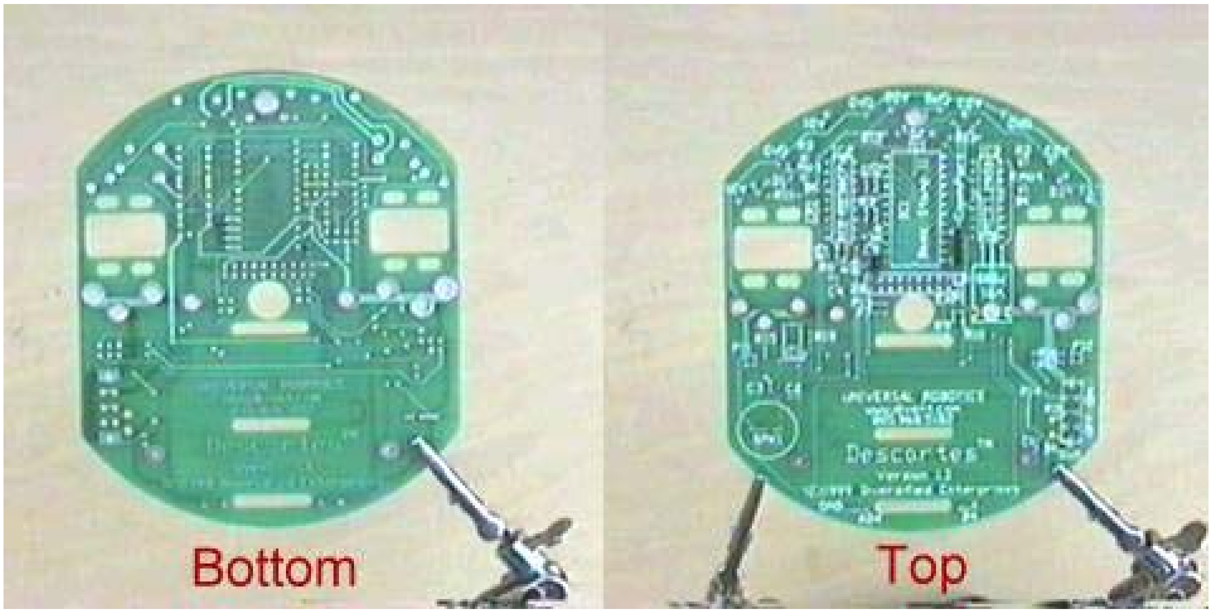

Figure 1. Printed Circuit Board (PCB)

Install the IC sockets, headers, resistors, capacitors, and ceramic resonator. If you are unfamiliar with soldering or a little rusty consult with someone who has soldering experience, or look for soldering tutorials on the web. If you have an old dead piece of electronic equipment with a PC board in it you can practice with that. If you make a mistake you can use desoldering braid to wick the solder away and start over. Remember that a clean tip is important and a damp sponge works well to keep the hot iron tip clean. If you apply heat to the solder for too long it will burn away all of its flux and leave a messy solder joint. Adding a tiny bit of solder will make it attractive again. Where too much solder has been placed, remove it with the desoldering braid and start over. All of the components, with the exceptions noted below, are to be installed from the top of the PCB, the side with the white legend printing. When correctly installed the components will cover up most of the printing. The leads will protrude through the board to the bottom side and will be soldered from the bottom.

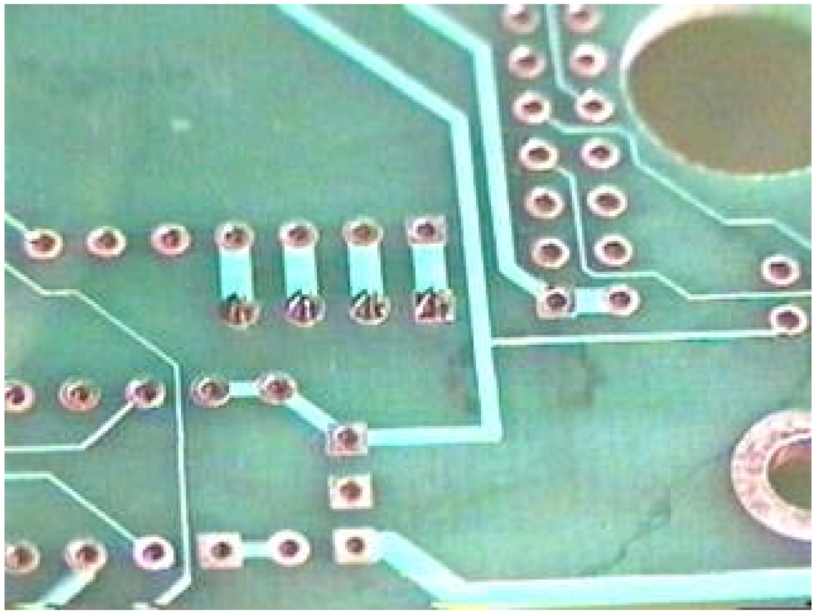

Figure 2. Well Made Solder Joint

Refer to the parts list below for component location reference.

Start with the integrated circuit (IC) sockets. The pin configuration of these sockets and the IC's they accept, is defined as a dual inline package (DIP). Note the orientation; the semicircle marker on one end of the socket will line up with the silk-screened legend on the PCB. By convention, this denotes the end of the DIP that pin 1 is on. When viewed from the top down holding the IC with its semicircle to the left, pin 1 will be on the lower left. Some IC's have an additional mark identifying pin1; you may find such a mark on the PIC16C71 chip. The remaining pins are numbered sequentially in a counterclockwise fashion, ending with the highest numbered pin at the upper left. This pin inserts into the square solder pad on the PCB. The end of the socket with the semicircle (and pin 1) goes towards the middle of the robot on all three sockets. Place the sockets in the PCB, and solder all the pins in place. It helps to start with two opposing corner pins first. Then check to make sure the socket is still fully seated down on the PCB and solder the remaining leads.

Now add the expansion header and programming headers, reset switch select jumper and A/D port plugs. Cut the male SIP (Single Inline Package) header into a four-pin section and a two-pin section. They may already be supplied in two pieces. The four-pin header goes adjacent to pins 1-4 of the BS2-IC. This is the communication port connector, and is used for programming. The two-pin jumper goes in location J1, on the other side of the BS2-IC and adjacent to the expansion header. The 20 pin female SIP header should be cut into two 10-pin sections that get soldered in next to the center hole and makeup the expansion header. The 12 pin female SIP header should be broken into 12 separate female pins from which all the plastic can be removed. These will be placed in the holes labeled AD1 - 4 and the GND positions. These sockets are small and get hot during soldering. Rather than attempt to hold them in place it will be easier to solder them in from the top. Apply the tip of the iron between the board pad and the part on one side and feed the solder in between the part and the PCB on the other side. When the pin gets hot, the solder will wick into the hole and all around the pin.

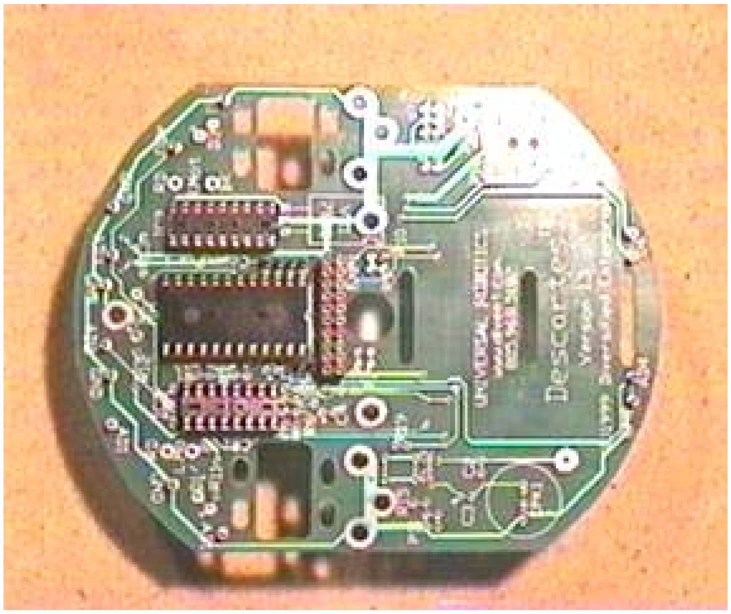

Figure 3. Location of headers and pins

The resistors get soldered into their respective marked locations as noted in the table below. Note that resistor R8 goes next to resistors R5-R7, and does NOT use the through-hole to the left of the 'R' in 'R8' on the board. Consult the large image below for correct parts placement.

Explanation of components J1, P7 and P13: These are optional components that allow Descartes to be configured in a variety of ways to suit the needs of different users. J1, when shorted with the shorting jumper, connects SW2 to the BS2-IC's reset line. When J1 is shorted, pressing the SW2 tact switch will reset the BS2-IC. This is useful when you want to restart a user program many times, as it is faster and easier than cycling the power switch. The resistor in location P13 (stands for Port 13) connects this switch to Port 13 on the BS2-IC. This way, when the shorting jumper is removed, the tact switch can be utilized by the user's program as a function switch, to change operation modes. If you want to use Port 13 for other circuits, simply remove or cut the resistor at this location. Another option is to solder single SIP socket pins such as those used for the A/D ports into this location so that the resistor can be removed and replaced at will. The resistor at P7 connects the IR LED output of the PIC co-processor IC to Port 7 of the BS2-IC. With the resistor in place, this line will go low (ground, logic 0) when the robot is in motion, and will monitor the encoder pulses. The port will go high (5 volts, logic 1) when the robot has completed a motion routine. This allows user programs to monitor when the PIC has completed a motion routine without having to communicate with the processor and may be useful in LOGO-style drawing programs. Again, if you are not interested in sacrificing an I/O line for this purpose you may leave the resistor out or use the SIP socket method.

The capacitors are to be added next. Note the polarity on capacitors C3, C4, and C5. C3 and C4 are electrolytic, their NEGATIVE leads denoted by a stripe down the side of the capacitor, with a (-) in it. Also, they usually have a longer positive (+) wire. The positive lead goes in the hole marked (+). C5 is a tantalum type, and will have a (+) marking next to its positive pin. The (+) leads of all polarized capacitors on Descartes go toward the front of the robot. The 3-pin ceramic resonator CR1 goes between the PIC IC and the left motor mount hole and can be installed in either direction. Next install SW1 and SW2. These switches can be installed in either direction. Add VR1 such that the part number faces up, it lays flat against the PCB, and the hole in the tab lines up with the hole in the PCB. You can temporarily bolt it in place with the supplied 6-32 nut and machine screw while soldering. Add SPK1 using whichever PCB holes give the most centered installation. The speaker will work fine installed in either direction, but if it has a (+) marking or a red dot on it this lead can be oriented closest to the middle of the robot. Solder in the battery connector by routing the red and black wires through the marked holes next to the power switch SW1. Solder these wires to the solder pads marked by the white lines on the top of the board. The red wire connects to the center pins of SW1, and the black wire connects to the ground pad at the very end of the switch, the pad that solders the metal case to the board. Twisting the red and black leads together all the way up to the connector plug will enhance the robot's appearance.

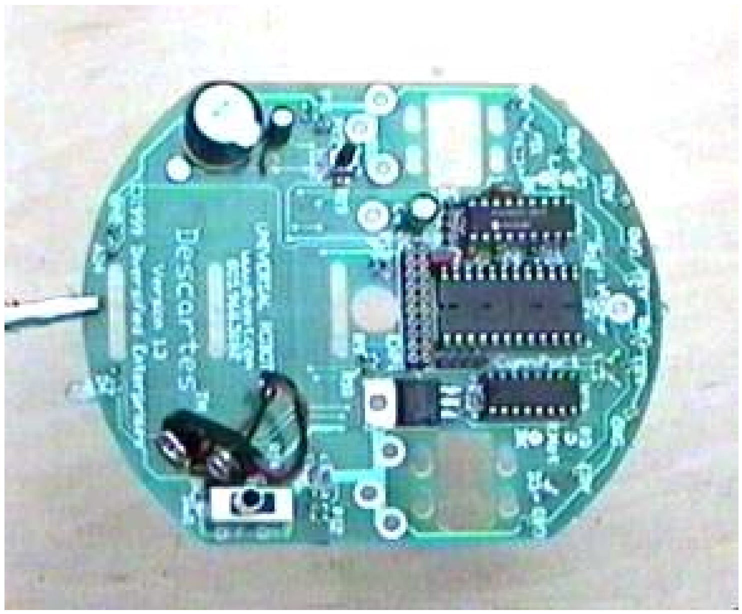

Figure 4. PCB with all components soldered in

PC Board soldered components

Note: some components may already be mounted to the PCB

Resistors:

R1,2,17	220-ohm (red, red, brown)
R3,4,9,10,15,16,19	33K ohm (orange, orange, orange)
R5,6,7,8	2.2K ohm (red, red, red)
R11,12,13,14	150 ohm (brown, green, brown)
R18,P7,P13	10K ohm (brown, black, orange)

Capacitors:

C1,2	100,000pF (small ceramic)
C3	10uF 6.3V (electrolytic can)
C4	100uF 10V (electrolytic can)
C5	4.7uF 6V (4u7, dipped tantalum)

Sockets:

IC1	24-pin socket
IC2	18-pin socket
IC3	16-pin socket

ICs:

```
IC1                      BS2-IC computer
IC2                      PIC16C71 computer
IC3                      L293D motor driver
VR1                      7805 +5V voltage regulator

Headers:
EXP1                     2 x 10-pin female SIP strip
GNDs and AD1-4s          Single pins from female SIP strip
J1                       2-pin male SIP w/2-pin fem jumper
Comm.Port                4-pin male SIP

Other:
PI1, PI2                 Four-pin dual-windowed encoder
CR1                      10MHz 3-pin ceramic resonator
SPK1                     Speaker
SW1                      Large DPDT slide switch
SW2                      Micro tact pushbutton switch
LED1,2,3,4               High-intensity LED
```

Solder the LED's to the PCB. The LED's, as with the photointerruptors, are unique in that they are installed on the BOTTOM of the PCB, the side without the white legend printing. Install them such that the longer lead of the LED goes into the hole marked (+), and they have enough lead length remaining to bend outwards from the robot, with the tip of each LED slightly protruding past the edge of the PCB. They should also be able to point directly parallel to the PCB, even slightly upwards. The solder can be applied from either side. Don't overheat these components, if you have trouble soldering them, let them cool and try again.

Install the photointerrupter encoder sensors. Again, these components are installed on the bottom of the PCB, figures 7 and 8 show them in their final position. Unlike the other components you have installed so far, the physical placement of the photointerrupters are critical for their proper operation. Looking closely at the face of these parts you will notice they are divided into two segments. One is the IR emitter the other is the receiver. These segments can be seen in Figure 8. You will also notice a flat on one side on the component body. Install the photointerrupters so these "flats" face the wheel axle. The dividing line between the emitter and receiver sections should tilted slightly so it points directly at the axle. You may wish to refer to the section describing installation of the axle and bumper wires if you are unclear as to where this is on the PCB. Again figure 8 shows a photointerrupter installed in its correct orientation. In addition, the body of this component should be placed so its face lines up flush with the edge of the PCB. If it is set too far back from the edge of the board a weak reflection and/or excess ambient light will compromise its operation sending faulty readings to the PIC. If the part is set too far out it will drag on the wheel damaging the encoder index marks, or it will be too close to the wheel to provide an adaquate IR signal to the receiver, again resulting in faulty readings being sent to the PIC. In addition, the face of the photointerrupter should be positioned so it lines up with the index marks on the wheel encoder sticker. This is a final adjustment that can be made by carefully bending the part into place after the axles and wheels have been installed. It is best however to install the part as near as possible to its final position before soldering it in to minimize the amount of bending needed later. Figure 7 shows the component and lead positions of the installed part.

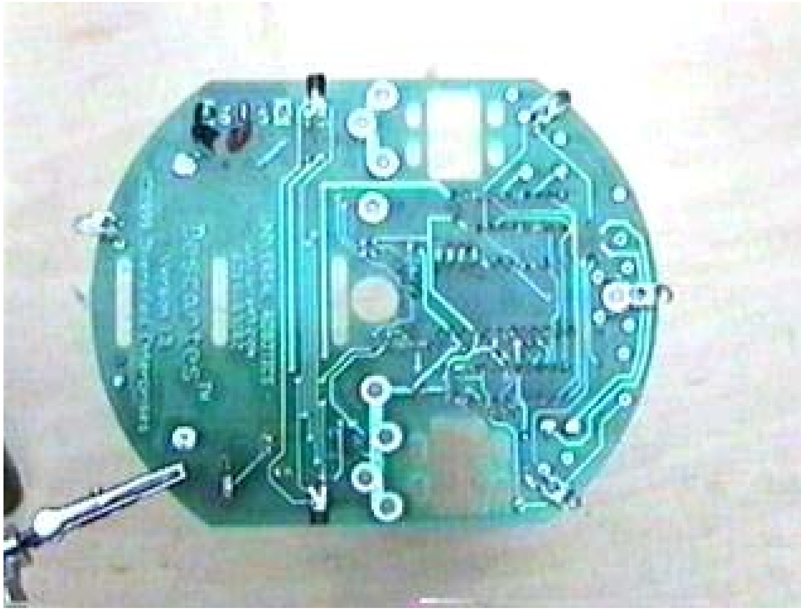

Figure 5. Bottom of board showing LED's and encoders

Add the IC's and sensors. The three IC's can now be plugged into their respective sockets. Often times, IC's come out of the package with their wire leads angled outwards a bit. These leads can be straightened by **CAREFULLY** pressing the whole side of the IC against a flat surface such as your tabletop. These leads don't like to be bent back and forth many times, so don't be in a hurry here. Once the leads are parallel with each other in all directions, plug them into their respective sockets by lining up ALL of the pins, then pushing straight down with your thumb. Since they are all different sizes you shouldn't have difficulty identifying which socket accepts which IC. Remember the end of the IC with the semicircle dimple, is the end that lines up with the semicircle on the PCB and socket; toward the center of the robot. Once all three ICs have been installed, cut the remaining shrink tube into pieces such that you can cover the leads of all four CdS cells and the thermistor, except for at least 1/4 inch at the ends where these parts will plug into the PCB sensor mount sockets. Each sensor needs to plug into an A/D port and a GND port. Some A/D ports are repeated providing a multitude of configuration options for non-contact obstacle avoidance, light/dark seeking, line following of thin or thick lines, etc. If you want to enhance the directionality of the light sensors / CdS cells (recommended!) paint their backs with a black sharpie permanent marker or paint. To further enhance directionality, shrink or glue a small length of black tube, such as heat shrink tubing, around them. Any resistive sensor or device can be plugged into these ports and read by the user program, such as a potentiometer or dew sensor. The device should have a center value matching that of its respective divider resistor. Resistors R5 through R8 are the divider resistors for ports 2, 3, 1, and 4, respectively.

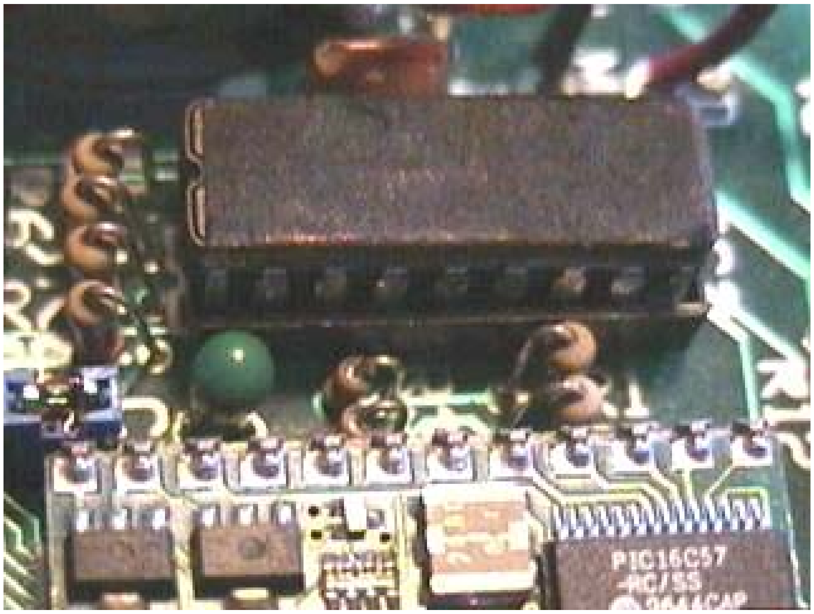

Figure 6. Close-up of PCB parts.

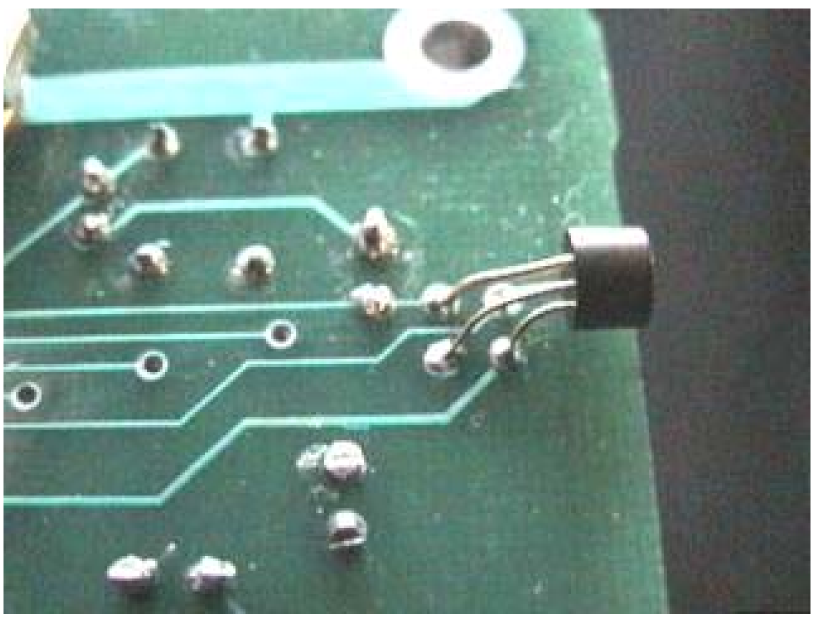

Figure 7. Encoder mounting. Face of encoder should be flush with edge of PCB.

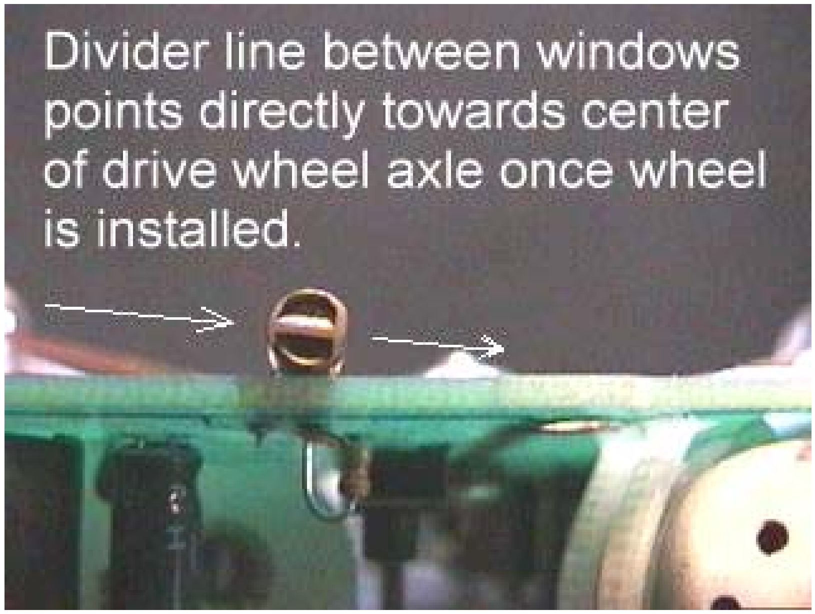

Figure 8. Correct encoder face angle.

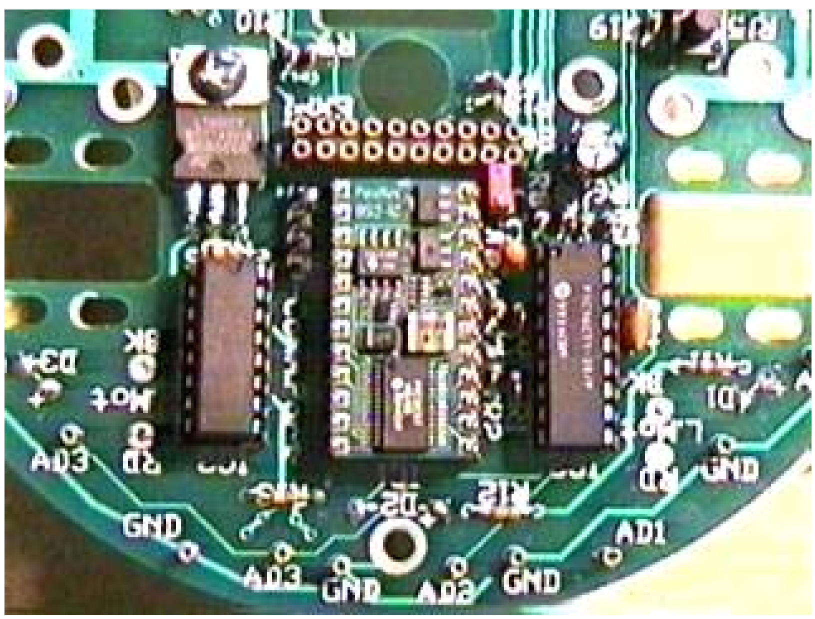

Figure 9. IC's installed.

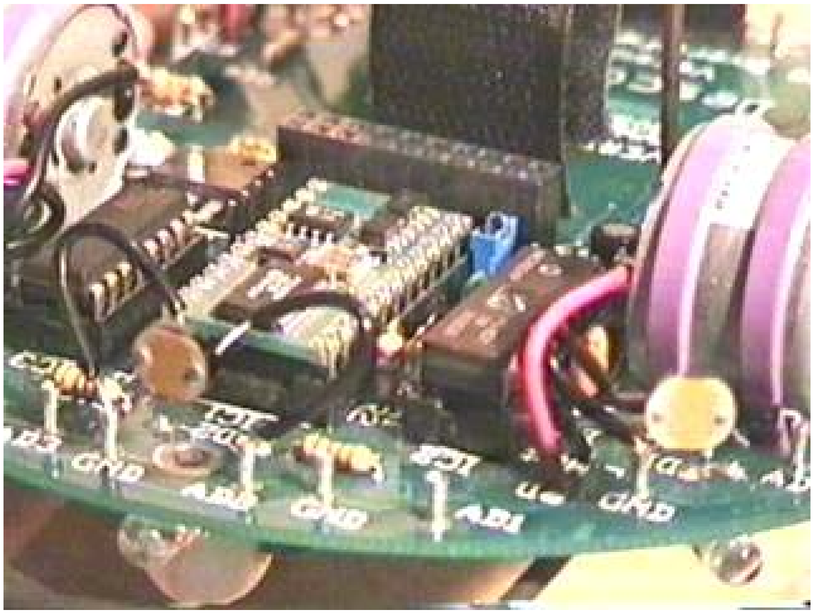

Figure 10. CdS cells plug into port sockets.

4. Clean and inspect your work. Use rubbing alcohol and an old toothbrush or rag to clean the solder flux from all areas of the board. This takes a little time, but is well worth it. Be careful not to get alcohol in the speaker, SW1 or SW2. After cleaning, carefully inspect ALL of your solder connections. Do not underestimate the importance of this step, if you get everything correct the robot will turn on and operate instead of frying difficult-to-obtain and/or expensive parts. Make sure that all components have been installed in the correct holes. Then check to see that they are all in the correct orientation. Compare your work to the pictures. Look for poorly connected solder joints. Look for solder 'bridges' where two adjacent pins have inadvertently been soldered together. Repair any mistakes, and clean again. Remove the sticker from the top of the speaker, if it has one.

You have just completed the longest section of the assembly instructions. You are now ready to move on to Section 2: Shaping the Steel Wires. Before continuing take a moment to clean your bench so you can begin Section 2 with a clean and well-organized workspace.

Back to the Table of Contents

Diversified Enterprises
158 Aero Camino Rd.
Santa Barbara, CA 93117

Descartes Assembly Instructions

Section 2: Shaping the Steel Wires

The second section in the assembly process covers the formation of the bumper sensor/axle assemblies, bumper contacts, caster gear, penholder, and their installation on the robot. Remember to read through each step completely before executing it.

Before we begin it's worth a brief discussion about wirework. As you look over your robot, you'll notice electronics, plastic materials, motors, and so forth. All these items are the result of relatively new technological development. By comparison, the practice of forming shapes from wire is very old. Over 2,000 years ago soft precious metals were worked into thin wire threads and woven into fabrics. Harder metals such as iron were hammered into thin sheets and cut into strips; these were then bent into desired shapes. The practice of making wire by drawing it through a series of progressively smaller holes was developed in the late 13th century and required great physical strength. Machinery for making wire finally appeared with the Industrial Revolution and ushered in an explosion of wirework products such as baskets, trivets, hooks, countless other kitchen items and the ubiquitous barbed wire fencing. Wirework performs practical function and borrows from ancient art. It is rarely exact and its variations impart an individuality that should be seen as distinctions rather than flaws. In addition, the use of various disciplines, new and old, creatively applied in a functional way, is a hallmark of robotics that is in particular expressed in Descartes and its use of wirework

Your Descartes Kit includes a paper template to serve as a guideline by which to make simple yet functional wire parts. This is not intended to restrict you from experimenting with other shapes and functions serving your particular application needs. Nor should you become overly concerned if the parts you make end up looking slightly different from those shown in the following figures. The goal here is to produce a robot that works to your expectations. If your objective is to build a working robot first, and experiment second, then we encourage you to follow the template as closely as possible. We present the following instructions on this premise.

Your kit includes three lengths of copper clad steel wire. Carefully remove the tape from the wound up pieces and **do not allow them to spring open suddenly as you remove the tape.** Some or all of the wires in your kit may be stamped with identification marks that appear as flattened sections describing the gauge and alloy the wire is composed of. These should be cut away at this time but do not dispose of them. In the event that any of the wire parts are to be replaced in the future the same alloy wire should be used. It can be obtained at any welding supply shop and identified with the assistance of these identifying marker tabs. Before bending or cutting the wires, locate the reference marks found on the lower left and upper right of your template. Measure these lengths to make certain that neither horizontal nor vertical dimensional changes have occurred during the template's reproduction. Both American and Metric references are provided for this purpose. Next identify the different wires that will be used for each part. The longest wire will be used for the bumper sensor/axle assemblies and bumper contacts. The short thin gauge wire will be used to form the caster gear. The short heavy gauge wire will be used for the penholder.

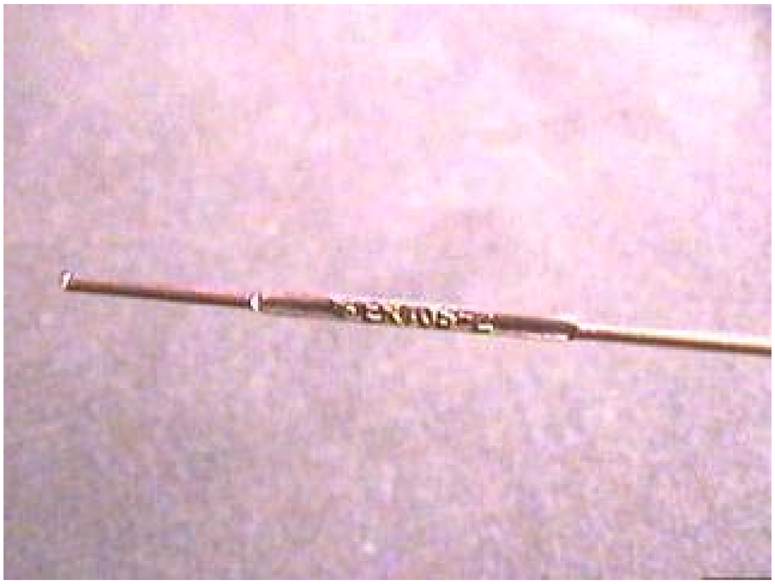

Figure 1. Wires and ID Tag

Form and install the bumper sensor/axle assemblies. Generally straighten out the thin gauge wires trying to not introduce any sharp kinks. Cut the long wire at its midpoint providing two equal lengths sufficient to form both left and right bumper sensor/axle assemblies. Figure A of the template matches the shape needed. The bend labeled #1 must be sufficient to wrap snugly around the 6-32 machine screw used to fasten the part to the robot. This bend should be made first since it is a critical shape that will require some force to form. Wrapping the wire around a nail or a small screwdriver will help to produce the curve needed. The length of the axle is not critical at this point, as it will be cut to fit after its installation on the robot, leaving it a little long will help in the bending process. The axle length should not be made shorter than shown on the template however. Bend #2 is not critical but it should be made to match the template as closely as possible. A pair of needle nose pliers will serve to form it. Once this portion of the bumper sensor/axle assembly has been made, shape the remainder of the part except for the final bend at the end of the wire. The second bumper/axle assembly should be constructed the same way and should duplicate the first part as closely as possible. Since these assemblies will be installed as a symmetrical pair variations between them will be obvious. It is therefore desirable to make sure both parts end up being identical.

Heat shrink tubing is included to cover these wires which protects furniture, quiets the wires from rattling against each other, and provides an attractive look. Since the tubing can be difficult to slide over the wire after it has been shaped, it should be put on prior to making the final bends. The tubing should cover the bumpers from the sharp bend (where the bumper is bolted to the PCB) to the last bend at the other end, leaving the ends of the wires uncovered such that they can maintain a sound electrical connection with the bumper contacts.

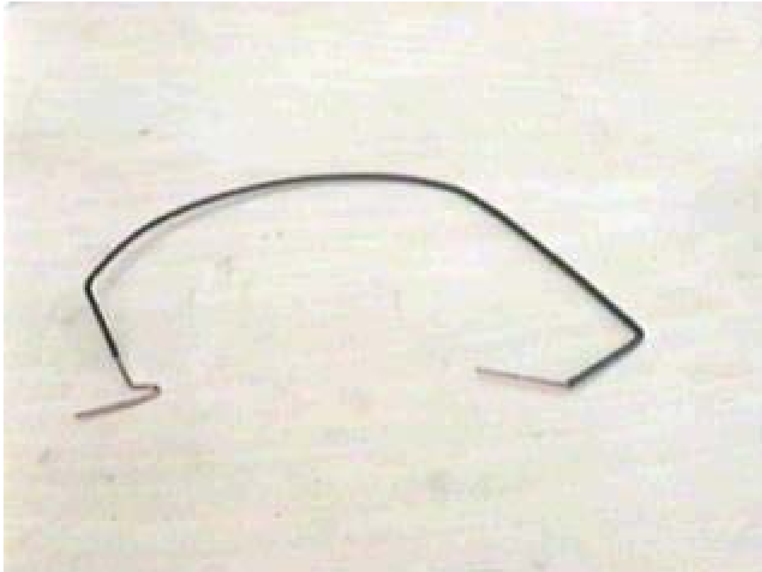

Figure 2. Bumper Sensors with Heat Shrink Tube

Cut the tube with a hobby knife to the middle of where the last bend should be and make the last bend. Using heat to shrink the tubing over the wires is not necessary but will enhance the bumper's appearance. Short lengths of extra wire will be cut away after you have completed making the entire shape. These cutoffs will be used to form the bumper contacts illustrated by Figure B on the template. Again make two of these for the right and left sides.

When installing the bumper sensor/axle assemblies, drop the screw through the robot from the top down. Place the bumper sensor/axle on the bottom of the robot and install the nut finger tight. Position the axle so it forms a 90-degree angle with the flat edge of the main chassis board and position the nut such that one of its flat sides aligns flush with the board edge. Hold the nut and bumper sensor/axle in these positions and tighten the screw using a Phillips head screwdriver. By tightening only the screw, the nut and the bumper sensor/axle will remain in proper position. The bumper sensors will be fine-tuned later in this chapter.

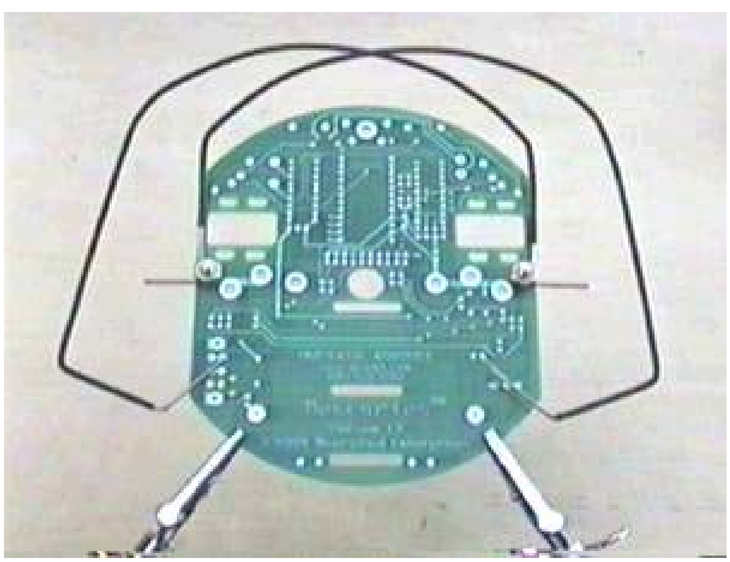

Figure 3. Bumper Sensors Installed

Form the caster gear. The short length of thin gauge wire will be used to shape the caster gear. Start by wrapping one end around the nail or screwdriver used to form the #1 bend on the bumper sensor/axle assemblies. The loop formed must be large enough to accommodate a 6-32 machine screw but do not make it any larger. Bend the wire until it matches the shape shown in Figure C. Now slide the caster bead into place and bend the wire to match Figure D. Enough end play should exist so the bead can move slightly from side to side and spin freely, an excessive amount of end play should be avoided however. If you wish to customize your robot by selecting different beads this is the time to do so. Beads can be purchased in a wide variety of colors and patterns from most hobby and craft stores. Either plastic or glass beads can be used but glass beads tend to be a little noisier when the robot moves along a hard floor. Regardless of the beads you select, try to use those with the hole drilled reasonably on center. We have found that plastic beads are often made a little more accurately in this regard. Also, the diameter of beads can vary greatly which is not of great consequence but small diameter beads should be avoided. In general the larger, rounder, and more accurately drilled the bead is the better it will perform.

Figure 4. Caster Gear

Form the penholder. Figures E through H illustrate the procedure for forming the penholder. The heavy gauge wire used for this part is difficult to form and requires several bends. Sufficient length is provided to make the job a little easier. Using a bolt or rod slightly larger than the pen you select for your robot, wrap the wire into a loop shown at the top of Figure E. Wrap the other end around the tool used to form the #1 bend on the bumper/axle assemblies. Cut off the excess from each end. The small loop must be large enough to accommodate a 6-32 machine screw but do not make it any larger. The shape you have completed should match Figure E as closely as possible. Next twist the wire along its length until the upper and lower loops are at 90 degrees to each other. This is shown in Figure F in which the upper loop is seen on its edge standing out from the plane of the paper. Now bend a 90-degree angle in the wire as shown in Figure G. Once this bend is complete bend the small loop up 90 degrees. Figure H shows the top and side view of the finished penholder. This part will be installed later in this chapter.

Figure 5. Penholder

Bumper contacts. If you haven't already done so form the small 'L'-shaped bumper contacts illustrated in Figure B of the template. Lightly sand the shorter of the ends to aid in solder adhesion. Do not touch these shorter, freshly sanded ends or the oils from your fingers will interfere with soldering. Install them from the bottom, as shown, with the long ends pointing directly towards the very rear center edge of the PCB. There must be enough clearance between the long ends running parallel to the PCB and the PCB itself for the ends of the bumper sensors to slide freely. There will also be a piece of heat shrink tubing covering the long exposed end, so leave about two wire diameters of clearance. You may want to shim them up with some folded paper and clamp them in place with a clothespin to make soldering easier. Solder the bumper contacts into place getting them hot enough for the solder to flow and adhere along the sanded part of the wire. Leave the extra length of wire, or at least ¼ inch of it, protruding through the topside of the PCB. This small stub will help to center the 6xAA-battery pack.

Once installed, cut two small lengths of heat-shrink tubing to cover the sections of the bumper contacts parallel to the PCB. Do not push the tubing past the bend point because the bumper sensor must contact bare metal at the base of the bumper contact in order to conduct electricity. The tubing ensures that when the bumper sensors are pressed, they break the electrical circuit and the BS2-IC can detect it. Cut the tubing to match the length of that section of wire and heat it with a blow dryer or other source of heat so it shrinks around the wire and holds tight. Overheating the tubing such as with a lighter or match may melt it. A tiny amount of Vaseline or switch lubricant coating the bare exposed part of the contact wire will help to prevent oxidation and maintain a sensitive bumper sensor without false triggerings.

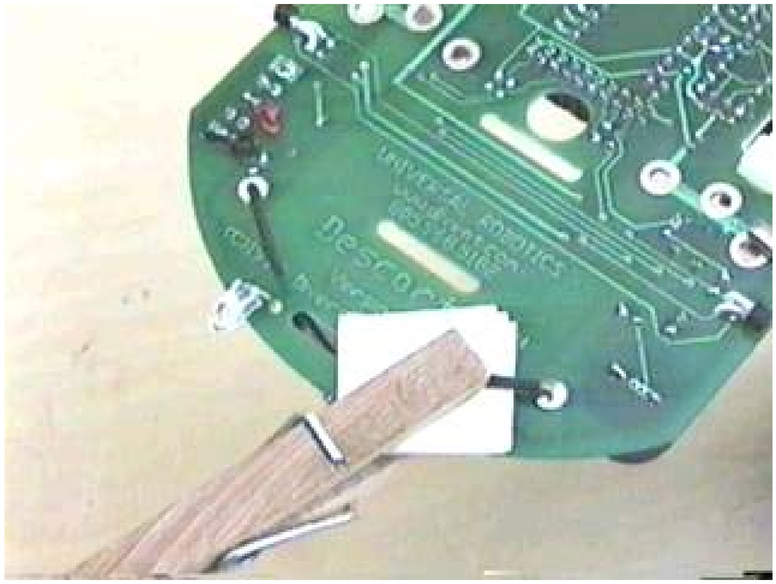

Figure 6. Bumper Contacts

Fine-tune the bumper sensors. First, be sure that the axles protruding from the sides of the robot are perpendicular to the flat sides of the PCB. The corner of a piece of paper can be used as a square. Check them from both the front and rear sides. Make certain the nuts holding the bumpers to the PCB are not turned such that they overhang and will interfere with the wheel rotation. A flat side of the nut should be flush with the flat side of the PCB. The forward part of the bumper perpendicular to the axle should also be flush with the flat side of the PCB and again should not overhang such that it can interfere with the wheel.

Once attached, the bumper sensors should be adjusted for proper operation. When properly adjusted and unhooked, the ends of the bumper sensors should rest freely about ¼ inch (6mm) from the bumper contacts. This will establish the correct tension once they are hooked together. This is not the position the bumper sensors will be left in for actual robot operation.

The ¼ inch distance is measured from the bumper contact soldered into the PCB, in a direct line drawn from the center rear of the robot through the solder pad where the bumper contact has been installed. In other words, in the exact opposite direction that the bumper contacts are pointing. The ends of the bumper sensors should be perpendicular to the bumper contacts. Refer to the figure below. The ends of the bumper sensors should rest parallel to the PCB and just barely touching it when the PCB is held right-side up (wires underneath.) Once you have fine-tuned the bumper sensors, hook them around the bumper contacts. Applying light pressure against each bumper sensor from the front of the robot should move them away from the bumper contacts without becoming unhooked. This method creates an extremely reliable, positive-acting and sensitive tactile sensor system. As with the bumper contacts, the ends of the bumper sensors should be coated with a very thin layer of Vaseline or switch lubricant. Silicone grease works very well, and will not attract much dust. Don't use the hardening kind!

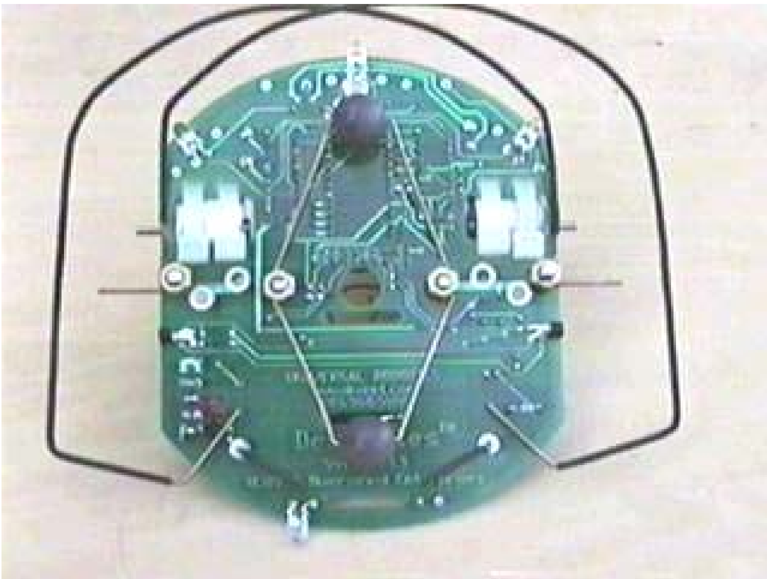

Figure 7. Adjusting the Bumper Sensors

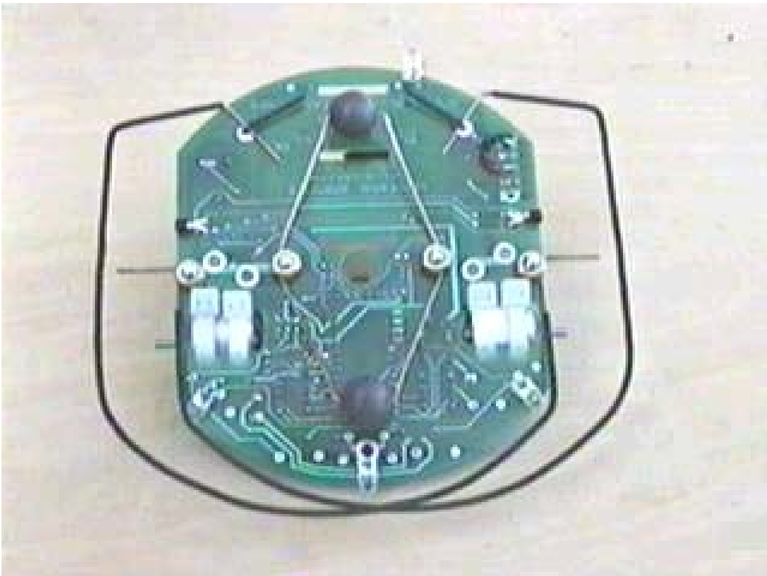

Figure 8. Bumper Sensors in Place

At this point the bumper sensors should adjusted for proper elevation. Bend them down in front such that they neither block light from the LED's nor drag on the ground. After making this adjustment again check their alignment with the bumper contacts. Make sure the bumper sensors continue to move freely when depressed and that they otherwise remain securely engaged with the bumper contacts.

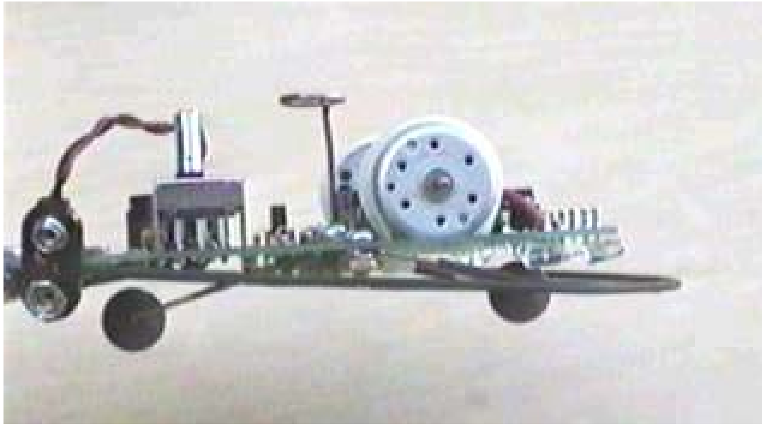

Figure 9. Side view of Bumper Sensors

Install the gear and penholder. Descartes' two most inner mounting holes are used to bolt on the penholder, caster gear, and to clamp down the 7805 voltage regulator IC. Position one caster gear-mounting loop under the other's. Align all of the parts and snug them down with the 6-32 machine screws placed in from the top down. When you tighten these fasteners, the caster wheels will exert pressure against the bottom of the PCB. Pull the gear ends away from the PCB bending them until the casters can rotate freely. This is an adjustment based on the type of terrain you plan to use the robot on. The more uneven the terrain, the closer to the PCB the casters should be. This will allow the drive wheels to have the best chance of maintaining traction with the ground. On smooth surfaces you may want to adjust them further away in order to minimize front-to-rear rocking. Never adjust the casters such that both of them are always in contact with the ground or the robot's drive wheels will loose contact with the surface it is being used on. Doing so will create drive and turning problems.

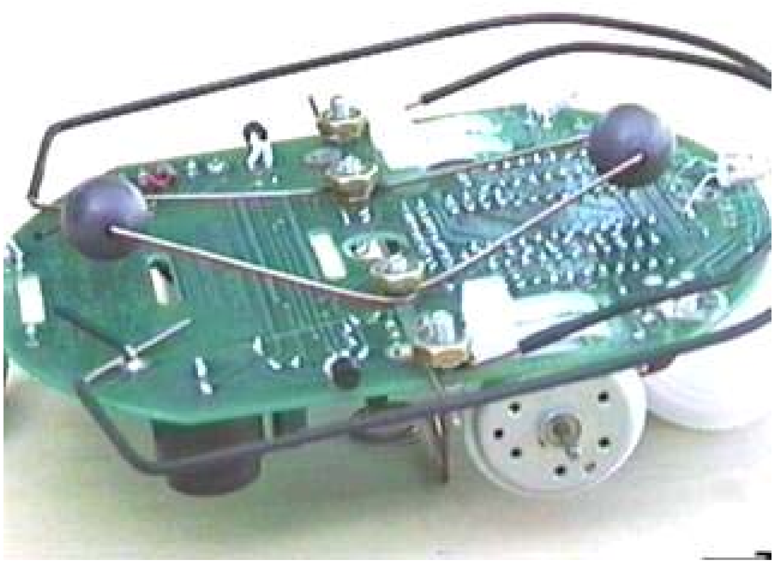

Figure 10. Attached Caster Gear

After the penholder has been attached, make any adjustments necessary such that its large loop will be positioned directly over the access hole through the center of Descartes. This will allow your writing tool to remain upright. Take care to not allow the penholder to overlap the circuit traces on the board adjacent to the mounting hole.

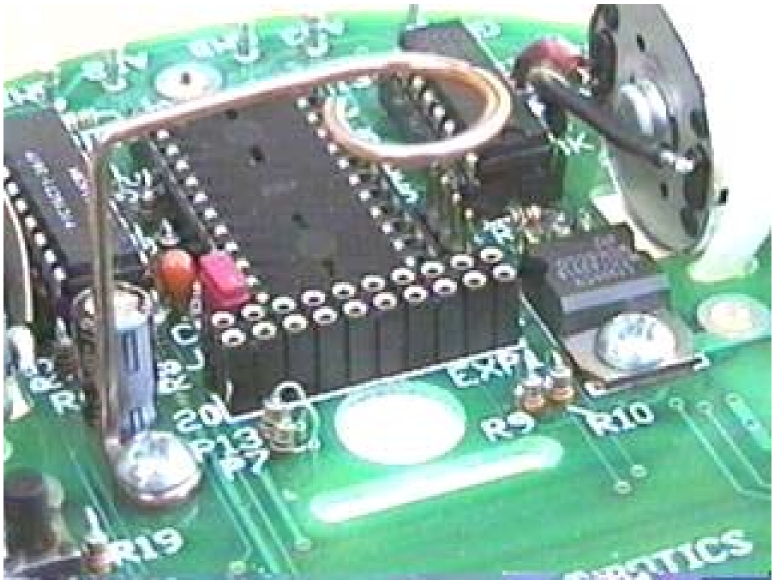

Figure11. Attached Penholder

You are now ready for final assembly, Section 3: The Drivetrain. Before continuing take a moment to clean your bench so you can begin Section 3 with a clean and well-organized workspace.

Back to the Table of Contents

Diversified Enterprises
158 Aero Camino Rd.
Santa Barbara, CA 93117

Descartes Assembly Instructions

Section 3: The Drivetrain

The third assembly section covers the installation of the motors, wheels, encoder discs, and batteries. Remember to read through each step completely before executing it.

Fasten the drive motors and connect to the PCB. The two drive motors are cradled in the square cutouts near the sides of the robot with the output shafts facing outward. Each should be positioned such that the outside edge of the motor rests against the outer edge of the cutout; in other words so the motors are as far away from each other as possible. Forward and aft of these cutouts are two pairs of rectangular holes to accommodate the nylon ties needed to secure the motors in place. From the bottom of the robot insert a nylon tie into the forward outer hole such that its locking mechanism faces to the front of the robot. Wrap the tie around the motor and feed its end back down through the rearward hole. Make sure you have it facing the correct direction before you connect the ends because they can't be unlocked once they are engaged. Refer to the images below. Tighten the nylon tie loosely around the motor holding it in place. DO NOT cut the extra length off yet. Install the inner nylon tie in the same fashion. Rotate the motor such that the (+) or red marked wire terminal is toward the front of the robot, and the two terminals are roughly parallel with the PCB. Make sure the motor is flush with the outer edge of its slot. Tighten the nylon ties with the locking end pressed snugly against the bottom of the PCB over the rearward tie holes. You can use one pair of needle nose pliers to press the locking end against the PCB and another pair to grasp the exposed length pulling them nice and tight. Usually the motors can be shifted around a little after tightening. Now cut off the excess length flush with the locking end. If you like, the motors can be glued into place with a small amount of super glue applied from the bottom. Be careful to use only a tiny drop, as too much glue will not bond well and will look messy. Do not glue the locking ties to the PCB, you may need to remove the motors for some reason in the future. Replacement ties can be obtained at any electronics store or most hardware or auto parts stores.

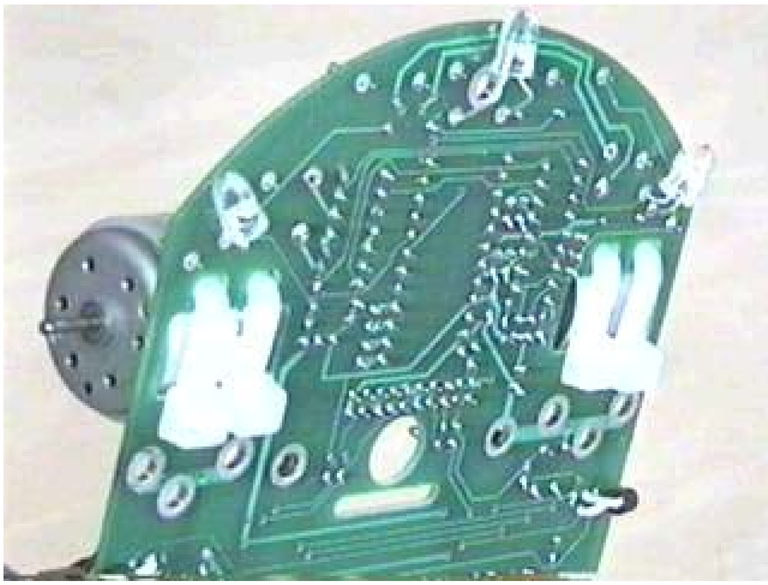

Figure 1. Motor Mechanical Installation

Once the motors are firmly held in place, cut the red and black wires to length to serve as jumpers between the motors and PCB. Strip some insulation from the ends of each wire so it can be soldered in place. Measure for the correct length of each wire before cutting, and leave a little excess in case you make a mistake while stripping the outer insulation. One red and one black wire should be made for each motor. The red wire connects the red marked solder terminal of each motor to the RD motor PCB holes on each side of the robot, directly in front of the motors. The black wires connect the other terminals to the BK holes. Clean the flux from the PCB after soldering.

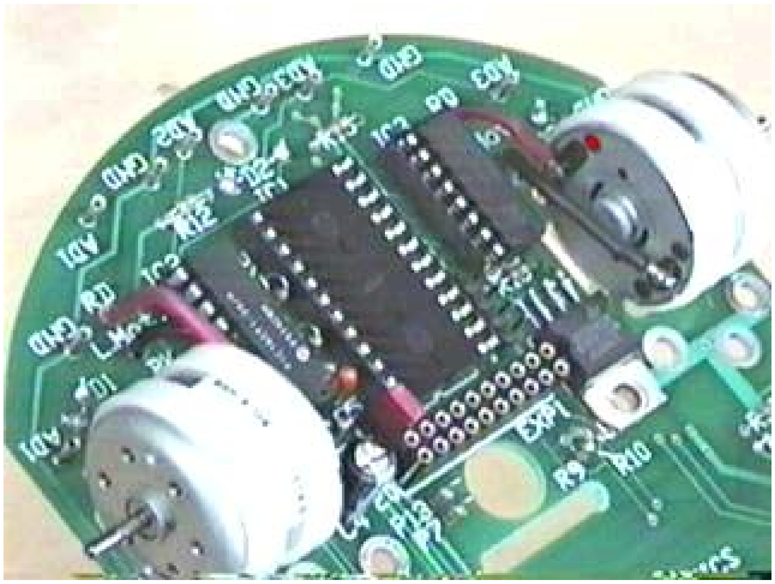

Figure 2. Motor Electrical Installation

The homestretch. By now you should be running out of parts! Stretch the rubber tires over the drive wheels into the deepest groove. Using a paper punch, remove the center of the encoder disks creating a hole approximately ¼ inch in diameter. Cut the encoder discs from the sticker sheet and trim the discs along the outer printed ring. You may want to wash your hands before handling the stickers to prevent getting the white surfaces dirty. Lay one of the wheels flat on the table such that the tire is against the bottom edge and clean the exposed surface of the wheel with rubbing alcohol. Let it dry thoroughly. Peel the protective paper away from the back of the encoder disk and line up the hole in its center with the hub machined into the wheel. Press the sticker firmly onto the wheel surface. Repeat with the other wheel.

Now unhook the bumper sensors and slide each wheel onto the axles such that the encoder disks are facing inward. Push the wheel retainer clips over the axles and measure the distance left between the inside of the clip and the outside of each wheel. Remove the excess axle length with a pair of wire cutters in small steps until the wheel retainer clip holds the wheel snug against the robot without any play.

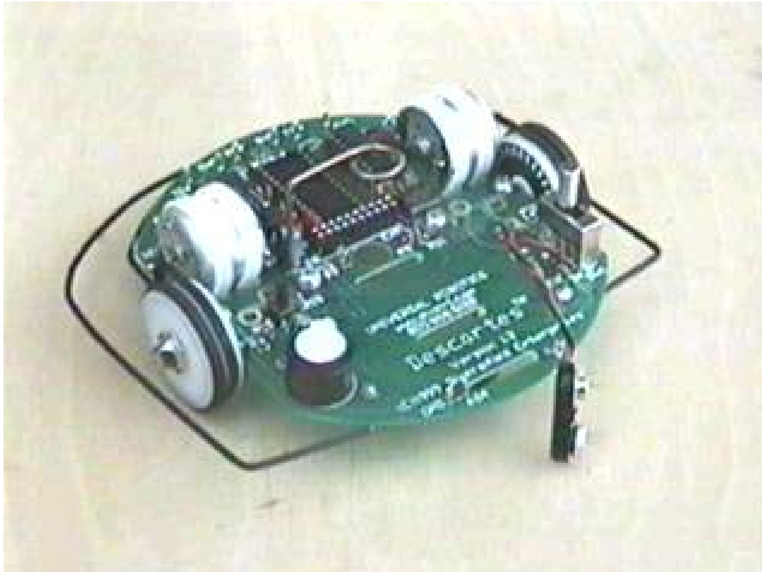

Figure 3. Wheels Installed

Figure 4. Correct belt alignment.
Drive belt rides centered on top of ridge.

Clip your 9V or 6 x AA battery pack to the battery clip and fasten it to the robot with the Velcro straps.

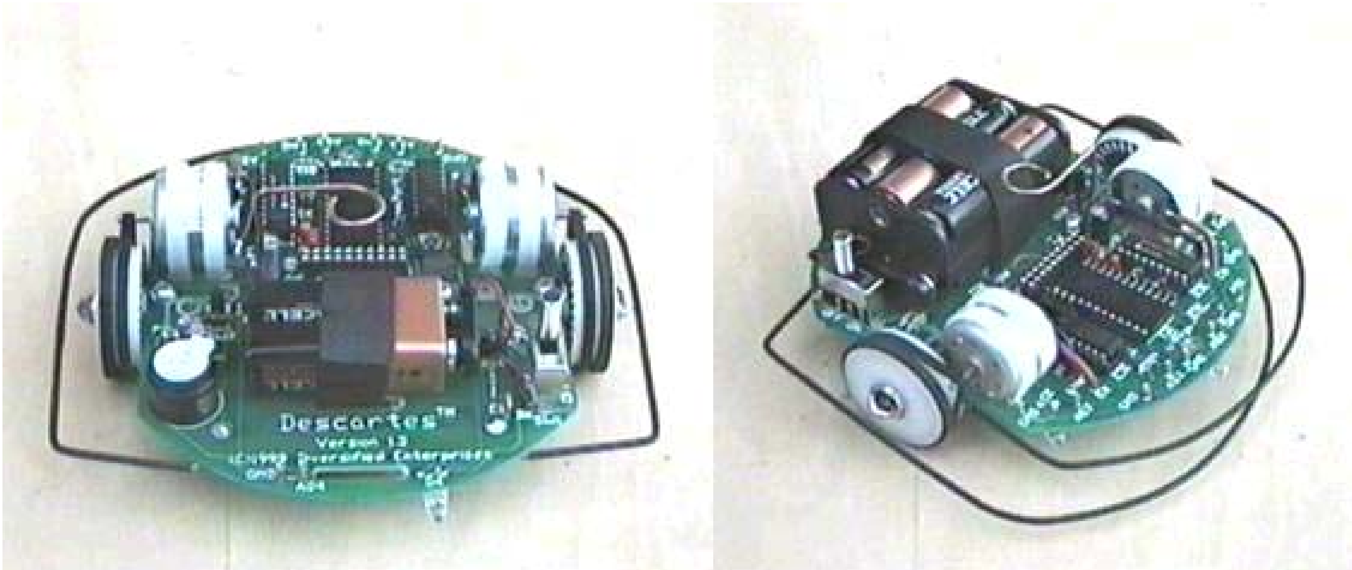

Figure 5. 9 volt and 6xAA Batteries Installed

Congratulations, you have completed the assembly of your robot and are now ready to move on to Section 4: The Programming Cable. Before continuing take a moment to clean your bench so you can begin this section with a clean well organized workspace.

Back to the Table of Contents

Diversified Enterprises
158 Aero Camino Rd.
Santa Barbara, CA 93117

Descartes Assembly Instructions

Section 4: The Programming Cable

In this section, you will create the data cable which connects Descartes to the serial port on your PC.

1. Attach the female header plug. Strip 3/4 inch from the outer insulation from both ends of the 5-foot, 4-conductor cable. Strip a small amount of insulation from each of the wires. Cut the female header crimp connectors from their keeper strip and attach one to each of the four wires on one end of the cable. Bend the small tabs over the wire to crimp them in place (Figure 1). Solder them together after crimping with a TINY bit of solder, the less the better. Any solder build up on the crimp connectors will prevent their insertion into the header plug. Be careful not to let the solder flow into the mating end of the connector! Plug each of these connectors into the 4-position female header plug in the order they exit the data cable. The connectors must be inserted into the plastic header plug such that the crimp tabs lock under the small plastic retaining fingers or they will pull right back out. Line them all up correctly, and snap them into place. When the cable is plugged into the 4-pin male header next to the BS2-IC on Descartes, the black wire should be towards the front of the robot (Figure 2). The connector will be tight until it is used a few times.

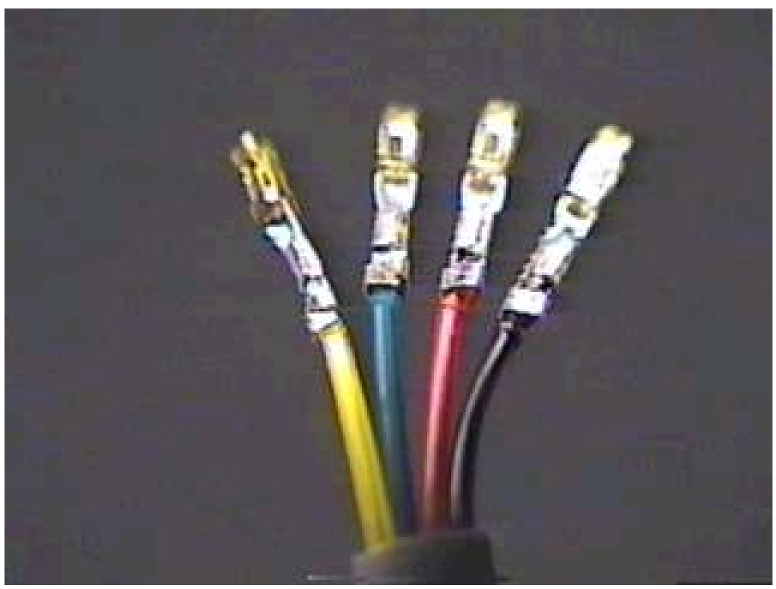

Figure 1. Header plug crimp connectors

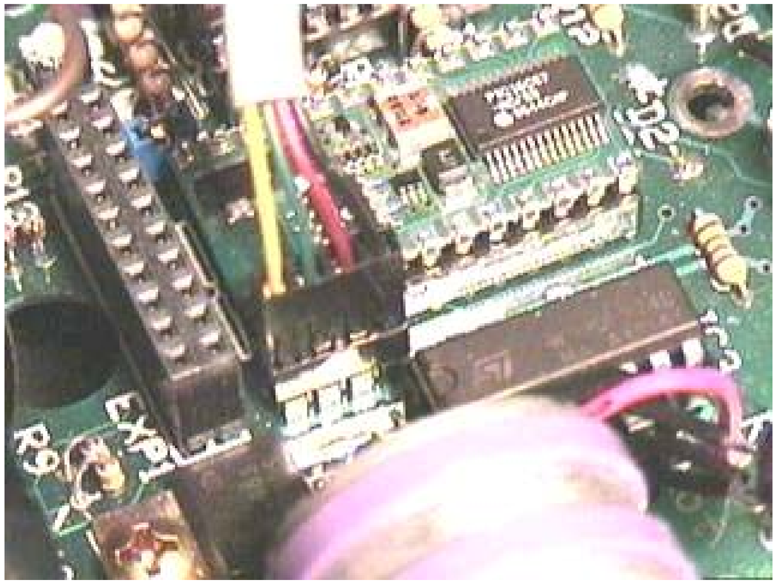

Figure 2. Black wire towards front of robot with cable plugged in

2. Attach the DB-9 serial port connector. Solder the four wires of the other end of the cable to pins 2, 3, 4 and 5 on the DB-9. Use pin 5 for the black wire, and pins 4, 3 and 2 for the remaining wires in the order they exit the cable (Figure 3). The wires should be in the same order on both cable connectors when finished (Figure 4). Also, for the BS2-IC's STAMP2.EXE downloading software to automatically find your com port, you should create a solder bridge or solder in a jumper wire between pins 6 and 7 on the DB-9 serial connector. See the end of this document for an excerpt from the Parallax documentation regarding this.

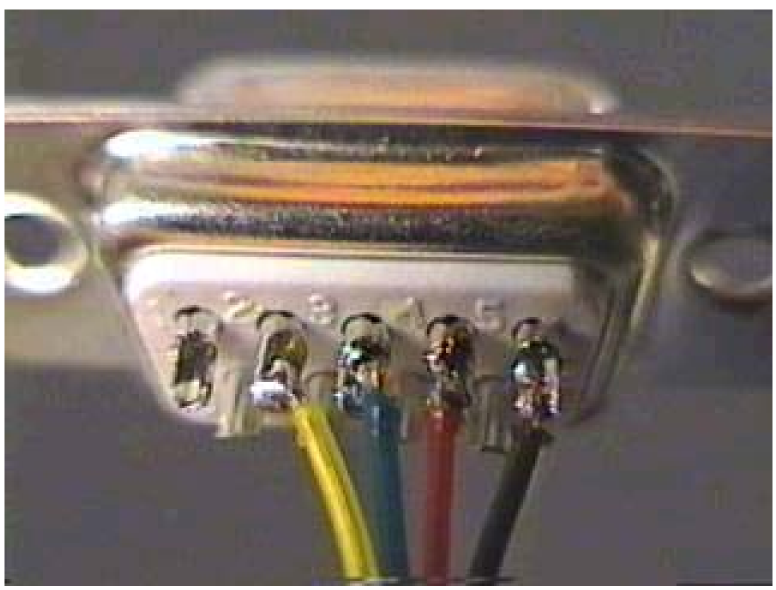

Figure 3. DB-9 wire connections

Figure 4. Wire colors should line up

3. Cover the DB-9 with the plastic hood. Screw the strain relief clamp halves together such that when the DB-9 is inside the plastic hood, the strain relief keeps the cable from pulling on the solder connections if it is tugged (Figures 5 and 6). Do not overtighten or you may damage the wires inside. Put the two halves of the hood together with the cable and DB-9 inside and add the connector securing screws if you want to use them. Since you may be plugging the cable into your computer often, you will likely not use them. Once the parts all fit nicely inside the hood halves, screw the hood together.

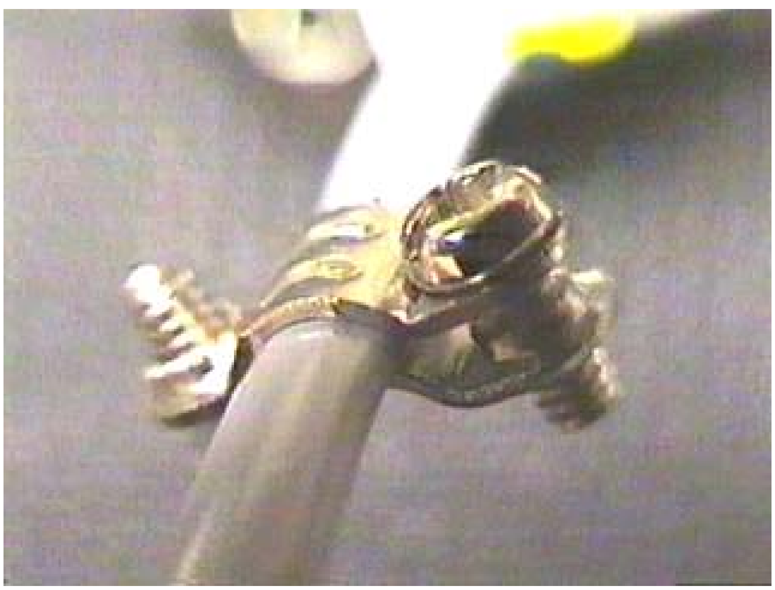

Figure 5. Strain relief configuration

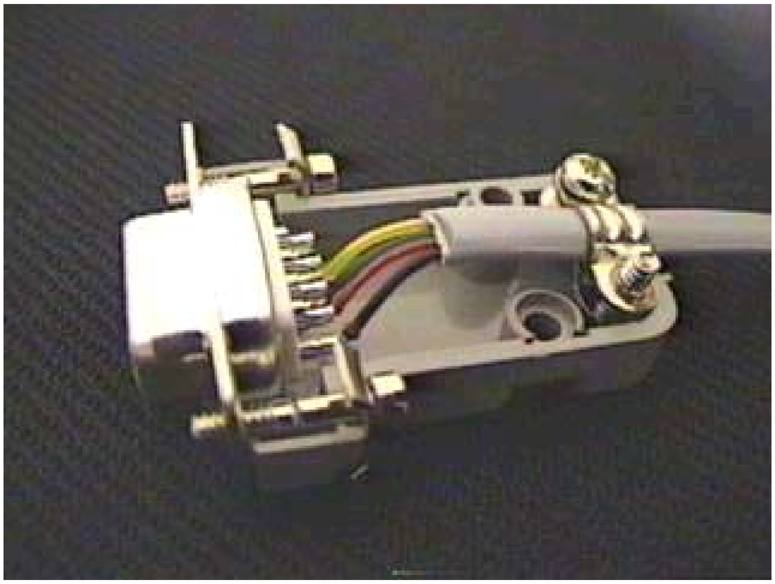

Figure 6. The wires should be loose with the strain relief in place in the hood

Following is an excerpt, included with permission, from the Parallax BS2-IC documentation:

Stamp II note: for the Stamp II software to automatically locate the Stamp II on any serial port, you must connect DSR (DB9 pin 6) to RTS (DB9 pin 7) on your PC's serial port. Otherwise, you will receive a "Hardware not Found" error. If this is a problem, you can tell the software which port to try by using the command-line option as follows:

STAMP2 /x

Where "x" is the serial port number (1-3)...

Assembly of your robot kit is complete! Let's make sure everything is functioning properly in the Testing and Troubleshooting section. This would be a good opportunity to return all your tools to their proper place and remove anything from your work area used during assembly.

Back to the Table of Contents

Diversified Enterprises
158 Aero Camino Rd.
Santa Barbara, CA 93117

Descartes Assembly Instructions

Testing and Troubleshooting

To insure that the Descartes is functioning properly a test program (TEST.BS2) is provided and should be the first program you run upon completing the assembly process. Refer to the included Parallax BS2-IC documentation for information on how to download programs.

When the test program has been loaded, place the robot in a well lit area and turn the power switch on. The LEDs should flash on in sequence, and sounds should be heard from the speaker. If they do not, then try the bumper test procedure below to see if the motors are functioning. If they are not, then the BS2-IC computer is likely not operating. Make sure that it is installed in the correct orientation. Also check your batteries for proper voltage. Look for a faulty solder joint or solder bridge short around the power switch. If one or more of the LEDs do not light up or the speaker does not sound, then look for bad or bridged solder connections around the LEDs, LED current limiting resistors, speaker, speaker capacitor, and BS2-IC.

If the wheels run either forward or backward in short jumps, one or both of the bumper sensors is stuck open. You can tell which one by the direction that the wheels go: left is forward, right is backward. If there is any solder flux left on the bumper switch contact wires from soldering it will act like an insulator and block the electrical contact. Scrape the bumper contact wire with a small knife where the bumper switch touches it, and clean again with alcohol. Then re-coat with Vaseline or silicone grease. Only a tiny coat is necessary.

Place your finger over each CdS cell one at a time, the corresponding LED should turn on while your finger is against the cell and turn off when your finger is removed. Make sure that each CdS cell responds correctly. If not, check that the corresponding CdS cell and BS2-IC I/O pins are soldered correctly. If you leave the download cable plugged into the robot after running the program from the editor program, the values from the A/D ports will be shown on the screen. The values should change with different amounts of light hitting the surface of the CdS cells. If a port value is 0, you have a short somewhere (likely a solder bridge.) If a port value is 255, there is either an open circuit, there is no sensor plugged into that port or the sensor's resistance is too high to detect. You can also try removing a CdS cell and plugging in the thermistor. The robot need not be turned off to change A/D port sensors.

Press the left bumper. Both motors should energize, rotating forward for 200 pulses and then stop. Pressing the right bumper will make the motors run in reverse. This tests the bumpers, motors and motor control IC, and encoders. If a motor does not turn then check for faulty soldering or detached wiring in the motor area. If the motor runs backwards when the left bumper is pressed, the motor wires are reversed. If the motor rotates and does not stop then the encoder is not functioning correctly. First make sure that the robot is not in direct sunlight or near a large window as stray light leaking into the wheel encoder may cause faulty readings. Next check the photointerruptor (encoder) leads for proper solder and lead placement, and proper alignment and spacing of the encoder face to the encoder sticker. There should be a small gap of about two matchbook thicknesses between the encoder face and the wheel encoder sticker. The 'bump' on the inside center of the wheel which the encoder sticker must fit around is designed to provide the correct spacing when the axle is perfectly straight. The encoder face should also line directly up with the alternating black and white patches on the wheel encoder sticker, and should face directly towards it.

Once the test program has been successfully executed, you may start writing your own code for your new robot. Program examples have been provided for you to learn about your new robot. Refer to the Parallax BS2-IC documentation as well as these example programs to familiarize yourself with the BS2-IC computer. A great way to learn is to understand the example programs, starting with the TEST.BS2 program used here, and make modifications to them to see how the robot's behavior changes. Then you may cut and paste sections of code into your own program, or simply add functionality to the ones provided. Be sure to visit our web site at www.divent.com frequently for new example programs and projects for Descartes, and request to be added to our mailing list.

Back to the Table of Contents

Diversified Enterprises
158 Aero Camino Rd.
Santa Barbara, CA 93117

Descartes Operating Manual

Version 1.1

We are constantly maintaining and improving our user documentation. It is recommended that you visit the download section of our web site at www.divent.com to check for the latest version.

If your Descartes was purchased in kit form, start with the Assembly Instructions

Contents

For the latest manuals, robot projects and programs visit the download section of our web site at www.divent.com.

Diversified Enterprises
158 Aero Camino Rd.
Santa Barbara, CA 93117

Descartes Assembly Instructions

Section I - General Information

Congratulations on your new purchase of Descartes, the most advanced low-priced programmable robot available. This document provides important information about inspecting, using, and caring for your Descartes, please read it carefully.

About Diversified Enterprises - Our company has provided test and calibration services on advanced electromechanical systems since 1991. Our Robotics Division is a natural outcome of our history with these technologies. Our mission is to further the growth and acceptance of robotics through alliances with education partners and by supplying robots of learning, research and entertainment value.

About Descartes (1596-1650) – A French mathematician and philosopher, Rene Descartes was known to his contemporaries as the father of modern philosophy. To him we credit the famous statement, "I think, therefore I am." Descartes introduced concepts on optics and formulated the law of refraction. He investigated the weather and offered an explanation of the rainbow. He laid the foundation for modern analytic geometry using algebraic notation to deal with geometric problems and through this work introduced Cartesian Coordinates. Descartes' progressive thinking advanced astronomy, engineering, mechanics, physics, medicine, mathematics, and although unimagined at the time, robotics. Descartes (the robot) honors this great scholar and the contributions he made to human thought.

How to contact us - Our factory is located approximately 125 miles north of Los Angeles, USA. Surface mail can be directed to us at: Diversified Enterprises, 158 B Aero Camino, Santa Barbara, CA, 93117. We can also be reached during normal business hours Pacific Standard Time at 805-968-5182, or by e-mail http://www.divent.com .

Warranty - Diversified Enterprises warrants this product to the owner for a period of 90 days from the date of shipment. During this period the company will repair or, at its option, replace any item found to be defective in material or workmanship, without charge to the owner for parts, service labor, or return shipping costs. Replacement or repaired parts will be warranted for only the unexpired portion of the original warranty. This warranty will extend to any subsequent owner during the warranty period. It does not apply to damage caused by accident, misuse, fire, flood, acts of God, or from failure to use, operate, or maintain the product in accordance with the documentation provided.

This warranty is in lieu of any other warranties, expressed or implied, including merchantability or fitness for a particular purpose. The owner agrees that Diversified Enterprises' liability with respect to this product shall be as set forth in this warranty, and incidental or consequential damages are expressly excluded.

Back to the Table of Contents

Diversified Enterprises
158 Aero Camino Rd.
Santa Barbara, CA 93117

Descartes Assembly Instructions

Section II - Descartes Operation and Service

Packing List

Your Descartes shipment includes one floppy disk and software, one programming cable, and one Descartes assembly. Please verify that all these components have been received in good order.

System Requirements

To reprogram Descartes, you will need an IBM PC or compatible computer, 3.5" floppy disk drive, serial port, 128K of RAM, and MS-DOS 2.0 or higher. A color video display is recommended.

Overview

At the 1998 Computational Neuroscience Meeting held in Santa Barbara, CA (CNS '98) our company organized a workshop to discuss the integration of Robotics and Neuroscience with top researchers from around the world. While on the topic of sensors and computer interfacing, we learned that if our brain were to receive, and attempt to process, all input sensory information all the time, it would simply overload. Therefore, the sorting of sensory data as well as the control of many fundamental tasks is the job of the peripheral nervous system. For example, you probably aren't aware of the temperature of the chair you are sitting in right now, but you can be sure if it was a hundred and fifty degrees Fahrenheit, you'd know it!

The same 'sensor overload' problem exists in the science of Robotics. If the main processor had to continually monitor all of the robot's input sensors, especially those requiring frequent checks and high-speed reactions as well as pulse the motors to maintain correct speed and heading, it would have little processing time left over for its main task. Programming would be much more difficult as well requiring the user's programs to thread and jump through various subroutines trying to keep track of everything. Not to mention the extra use of precious memory.

When we returned to the next year's CNS '99 meeting in Pittsburgh, PA we were proud to present the results of our interdisciplinary communication, Descartes. Descartes incorporates a Microchip(TM) PIC16C71 RISC secondary processor running at 10MHz. This processor is preprogrammed at our factory, and contains assembly code routines inspired by the biology of the peripheral nervous system. It receives its commands from the main processor at high speed and executes those commands on its own, requiring no intervention. Motor PWM pulsing for speed control, encoder monitoring for wheel position and velocity, A/D conversions and navigation control and monitoring tasks are automatically performed. If the user wishes to have full control of individual motor speed and direction, they can. Or, if the user simply wants Descartes to move forward 100 centimeters in a straight line at a given speed, one command is sent and the main processor is fully available for more important tasks such as robot 'behavior.'

An additional benefit of this architecture is ease of programming. Descartes' PIC processor uses motion commands not unlike those of LOGO, only shorter. Commands are available, however, to also take direct control of the hardware. And all of the commands follow a simple, one-line format.

Technical Specifications

The Engineering Team at Diversified Enterprises may incorporate design changes without notice as improved component technologies become available. Some such changes may effect the product's technical specifications. As of this writing, accurate technical information describing Descartes is as follows:

```
Size
    Length                          127mm
```

```
     Width                          153mm
     Height                         53mm

Power
     Drive and Computer             6 x AA or 1 x 9-volt
     Typical Current Draw           200mA while motors are on

Computers
     Main processor                 Parallax Inc. BS2-IC
     Sub processor                  Preprogrammed PIC16C71 at 10 MHz
     Available I/O                  5-7 Non-Dedicated I/O lines
     Documentation                  Supplied with BS2-IC
     Example Programs               Supplied with BS2-IC

Drive System
     Speed Control                  Independent, software variable
     Differential Steering          Yes
     Independent L-R                Yes
     Drive Wheels                   2
     Encoder Resolution             5.0 Counts/cm

Special Characteristics
     Assembly                       Full, Partial, or Kit Form
     Sensors
          L-R tactile bumper sensors
          Light Level Detection
          Thermal Detection
          Wheel encoders
          Programmable function / reset switch
     Output devices
          Four High-intensity LED's independently controlled
          Speaker
```

Using Descartes

Although Descartes is rugged and designed to move freely, it contains intricate mechanisms that may become damaged with misuse. Three common problem sources have been shown to cause the most serious damage to Descartes. These are pets, tables, and pedestrians.

Pets can become very frightened of Descartes if not given an adequate chance to become acquainted. Some pets see Descartes as an unfamiliar creature infringing upon their territory. This can provoke aggression and defensive actions even in pets that normally do not behave aggressively. If your pet and Descartes will be sharing the same area, we advise you to slowly introduce them. Give yourself time to study your pet's reaction and take steps to protect Descartes from attack or total destruction.

Tables, desks, and other raised surfaces are a natural playground for Descartes. Their flat wide open space and convenient height make them ideal to observe Descartes moving about. If contained by walls, books, or other structures Descartes will provide many hours of enjoyment. If not so guarded Descartes will find its first opportunity to run off the table and crash to the floor. Often the damage resulting from such incidents is limited to a bent or loosened component, which is easily repaired. Bouncing down a flight of stairs or sailing off a second story landing can inflict far more fatal damage however.

Although Descartes may be programmed for awareness of objects and motion, its detectors are no match for inattentive human footsteps. Complicating this danger is Descartes' tendency to dart and scurry across walkways when allowed the opportunity to do so. Pedestrian traffic should always be given due consideration and Descartes should only be used with gentle caution and careful supervision.

Operating Descartes

Battery - Descartes uses easily obtained AA or 9 volt batteries. At a fairly consistent activity level the AA pack should last all day. If Descartes is programmed to maintain a largely dormant state with little to no movement the battery can last several days. Depending on your intended usage you may want to consider using a rechargeable battery. Since these batteries typically supply under 9 volts even when fully charged, and have a much lower capacity than alkaline batteries, using only a high quality rechargeable is recommended.

Connecting the Programming Cable - The Programming Cable connects a standard 9 pin serial port (also called a COM PORT) on your computer, to the four pin programming plug located adjacent to the BS2-IC. The black wire of the 4-pin connector should be oriented towards the front of Descartes. If the Programming Cable is installed in the reverse direction, no damage will occur, but a " hardware not found" message will be displayed when attempting to download programs.

Use in a Clean Area - An added pet related hazard involves fur. Pet hair seems to have a natural attraction for Descartes' wheels and belts. Needless to say human hair can cause the same problem as can carpet fibers, threads, and similar materials. Descartes' drive components are easily cleaned by pulling the wheel retainer clip and removing the wheel. Some versions of Descartes include a spacing washer between the wheel and PCB, don't lose this! Put the retainer clip back on the axle while cleaning the wheels and belts. A cotton swab and rubbing alcohol are fine for cleaning the drive components. Do not soak them, though! The axles do not require any lubrication.

Detensioning (Storing) the Drive Belts - When Descartes is not in use, the drive belts should be removed from the motor shafts and left around the drive wheels only. This will greatly prolong their life. Should a drive belt or tire be worn or lost, new ones can be obtained at an electronics shop that carries rubber drive belts and idler tires for VCRs and tape decks. Replacement parts are, of course, always available from us as well.

The Penholder - Descartes has been designed to capture a writing utensil through a loop in the penholder and the hole in the center of the PCB. Depending on the type of pen used, just dropping it through the two holes will generally not create enough pressure on the writing tip to leave an ink mark when the robot moves. If the pen chosen only requires a small amount of force to draw, a rubber band from the top of the pen to the penholder with light tension should provide enough downward force without adversely affecting the motions of the robot. If the pen requires more force than can be applied with the rubber band, and without sacrificing robot performance, a weight can be used. A stack of steel nuts or washers taped to the pen will often suffice for this. Tape can also be used around the pen to enlarge its diameter so that it does not wobble in the guides. The pen must be able to travel smoothly in the vertical direction, so do not make it too snug.

The penholder feature on Descartes allows 'Spirograph' type programs to be written resulting in some very cool designs. The pen can also be used to keep a hardcopy record of the robot's movements that can be valuable for some behavior studies. When using multiple robots configured for line tracking, the pen can be used to lay down an analogy to a scent trail such as that followed by ants. In the classroom, interesting challenges can be put forth in which the robot must be programmed to draw some specified shape of a particular size. The application potential of this powerful feature is nearly limitless. For advanced experimenters, the addition of an R/C type hobby servo to lift and lower the pen will provide a complete LOGO type drawing robot. There is plenty of extra 5-volt power available from the expansion header and the BS2-IC's PULSOUT command to make controlling the servo a breeze. See the Parallax, Inc. documentation for more information.

Software

Descartes' onboard Basic Stamp II computer is manufactured by Parallax Inc. The files COMMANDS.TXT and SYNTAX.TXT are included and provide information on programming the BS2-IC. Complete software, manuals, and application notes covering the BS2-IC computer may be obtained at no charge from either www.parallaxinc.com or www.radioshack.com/sw/swb/projects/bstampidx.htm .

Installing the Software - Insert the floppy disk provided with Descartes into your computer's floppy drive. From this drive run the install program specifying the target drive where the software is to be placed. This is done by typing <INSTALL C: > assuming the software is to be placed on your computer's "C" drive.

With the installation procedure complete, a new directory called DESCARTE will be found on your specified target drive. From within the DESCARTE Directory run the STAMP2.EXE program by typing "STAMP2 /x". The 'x' in this command should be the number of the COM port you intend to plug the download cable into. This command will load the BASIC Stamp II editor/downloader software, a programming environment from which routines for Descartes can be developed. Refer to Section III of this manual for further information on developing programs with the BASIC Stamp II editor/downloader software.

Downloading Programs - Once a Descartes routine has been written and is ready to download, attach the Programming Cable as described under "Connecting the Programming Cable". With a fully charged battery installed, turn the On-Off switch to the "on" position. Within the Editor Program type "ALT R" to download the program into Descartes' BS2-IC. Descartes will immediately begin executing your program.

Example Programs - The sample programs are designed to help you get started programming your Descartes. They are to be loaded into the editor program, 'STAMP2.EXE.' It is assumed that you have already familiarized yourself with the Programming Cable and editor/downloader software.

Modify the programs by changing variables and motor commands as you please, and develop a feel for using them. It is a good idea to re-save the files under a different filename before you modify them. They must have a ".BS2" filename extension. If you make an error, the editor will likely let you know when you attempt to download your program to the robot. Don't forget to turn Descartes' power switch on or you'll get a 'hardware not found' error!

Pin Assignments - When writing your own programs the following BS2-IC pin out assignments will be needed for pin number referencing. Pins 7-13 can be used for your own circuits.

```
BS2-IC Pinouts:

P0 Speaker                P6 Right Bumper           P12 Available
P1 LED #1                 P7 'In Motion' line       P13 Available or Button
P2 LED #2                 P8 Available              P14 Flow control
P3 LED #3                 P9 Available              P15 Communication
P4 LED #4                 P10 Available
P5 Left Bumper            P11 Available


Expansion header pinouts:

1   Ground         11   P8
2   Ground         12   P9
3   P0             13   P10
4   P1             14   P11
5   P2             15   P12
6   P3             16   P13
7   P4             17   P14
8   P5             18   P15
9   P6             19   5 Volts regulated
10  P7             20   9 Volts unregulated
```

Diagnosing Faults and Troubleshooting Hints

Fails to Track Straight - If Descartes turns when it should be on a straight heading, one or both photointerrupters are not reflecting from the encoder stickers. Make sure that the solder connections are good, and that the face of the photointerrupter is flush with the edge of the PCB. Correct sensor-to-wheel spacing is important. Another possibility is that the encoder stickers are smudged or scratched, or the encoders are confused by bright lights and are missing pulses.

Should Descartes be stepped on or sustain an accidental impact, it is possible for an axle to bend. This will cause the wheel on that axle to slant which can cause belt derailment. The problem can usually be remedied by simply bending the axle back into position until the wheel is upright.

Hardware Not Found - This error message appears when your PC cannot communicate with Descartes' computer. This can occur when attempting to download a program with Descartes turned off or with a weak or dead battery. This can also occur with the Programming Cable installed backward. To correct this fault simply reverse the cable connection on the four pin Programming Plug.

Having the Programming Cable plugged into an undefined COM PORT on your computer will provoke this same error message. There may be up to four COM PORTS on your computer. If you are using COM PORT one, type "STAMP2 /1" at the DOS prompt when loading software. This command tells your computer to load Descartes' program though COM PORT 1.

Erratic or Improper Movement - Make sure that the batteries have sufficient charge in them. Most high-end alkaline batteries include a tester. Dead battery problems will cause the BS2-IC computer to reset when large surges of electricity are required, such as when reversing direction at top speed. Adding a line at the beginning of every program you write which beeps the speaker will alert you to this condition.

Occasionally one or both bumper sensors may malfunction causing the onboard computer to believe Descartes is constantly in contact with an object. This is most likely due to the bumper sensor having become bent or dirty. The bumper sensors are essentially a pair of normally closed switches that open when encountering an obstacle. Looking from the bottom up at Descartes' bumper contacts it should be possible to see if a sensor is in permanently open contact. Unhook the bumper sensor and straighten it until there is about 1/4 inch of gap between it and the bumper contacts. Then re-hook it. This will provide the correct tension for reliable operation and good sensitivity. If Descartes is used at high speed on very bumpy surfaces, the bumper wires may bounce off of their contacts. This problem can be remedied using the BS2-IC's BUTTON command, which includes a 'debounce' feature. Consult the Parallax documentation for more information.

No Movement - If one of the encoder sensors is not getting the correct reflection from the drive wheel, Descartes will run both wheels for a brief period and then run only one wheel continuously. The reason for this is that the secondary processor is trying to align the robot with its correct heading and it thinks that the other wheel is not moving so it runs only one until the other catches up. Check the solder joints and encoder alignment. Also be sure that your program is using the correct commands to communicate with the secondary processor. There must be a 100ms pause at the beginning of your programs to give the secondary processor a chance to 'boot up.' Use the command PAUSE 100.

Factory Remedial Support - If all efforts to identify and remedy a fault do not prove successful please contact the factory for assistance. You can reach us at 805-968-5182 or our Support Department can be found at support@divent.com .

Back to the Table of Contents

Diversified Enterprises
158 Aero Camino Rd.
Santa Barbara, CA 93117

Descartes Assembly Instructions

Section III - Programming the Parallax BASIC Stamp II Computer

Communicating with the PIC secondary processor

Command structure

The secondary PIC processor communicates with the on-board Parallax BS2-IC main computer in serial data format at 50kbps. Port 15 is used for communication, and port 14 is used for flow control. The PIC requires about 80ms to start-up, so the command PAUSE 100 should be included at the beginning of each program. Also it is a good idea to include the command SEROUT 15\14, 0, ["I"] at the beginning of each program to reset the PIC. The PIC is not necessarily reset every time the BS2-IC is, so this ensures they are both starting fresh at the start of the user program.

To send a command to the PIC, use the format:

```
SEROUT 15\14, 0, ["C", VALUE.LOWBYTE, VALUE.HIGHBYTE]
```

where **C** is the command character, **VALUE.LOWBYTE** is the low byte of the value, and **VALUE.HIGHBYTE** is the high byte of the value. The command character is not case-sensitive, so upper and lower case will work fine. Some commands do not require any value to be sent, some commands require a single 8-bit value (byte,) and a few commands allow for but do not require a 16-bit value (two 8-bit values, or bytes. Two bytes equal a word.) For example, to make Descartes go forward 100 encoder counts:

```
SEROUT 15\14, 0, ["F", 100]
```

To make Descartes go backward 300 counts:

```
SEROUT 15\14, 0, ["B", 44, 1]
```

Look strange? The maximum value you can have with an 8-bit number is 255. Therefore, to go forward 300 counts we need a 16-bit number. The low byte of this number is added to the highbyte of the number times 256. So, 44 + (1 x 256) = 300. Within your code, it will be easier to put the value 300 in a 16-bit (word) variable, and let the BS2-IC figure out the values of the high and low bytes. The following program segment shows this:

```
DISTANCE VAR WORD

DISTANCE = 300
SEROUT 15\14, 0, ["F", DISTANCE.LOWBYTE, DISTANCE.HIGHBYTE]
```

Another method of doing the same thing:

```
DISTANCE VAR WORD
DL VAR DISTANCE.LOWBYTE
DH VAR DISTANCE.HIGHBYTE

DISTANCE = 300
SEROUT 15\14, 0, ["F", DL, DH]
```

This second method helps to reduce excessive typing in your code.

To receive serial data from the PIC, you must first send a SEROUT request for data command, and then follow it IMMEDIATELY with a SERIN command. Example:

```
DISTANCE VAR WORD
DL VAR DISTANCE.LOWBYTE
DH VAR DISTANCE.HIGHBYTE

SEROUT 15\14, 0, ["["]
SERIN 15\14, 0, [DL, DH]
```

This will request for and receive the distance traveled in encoder counts of the left wheel. Another example:

```
HEADING VAR BYTE

SEROUT 15\14, 0, ["?"]
SERIN 15\14, 0, [HEADING]
```

This will get the current robot heading and put it in the variable HEADING. You may be wondering, angles can be from 0-359 degrees, how does this fit into a single byte variable? The answer is that the PIC in Descartes works with angle numbers between 0-179. There are a few reasons for this. One is that the resolution of the encoder disks on the drive wheels gives a turning resolution of 2 degrees. This means that 180 counts is 360 degrees of turning. Why waste an extra byte of precious BS2-IC RAM when it doesn't provide any greater accuracy? If you prefer to work with 360 degrees as a full turn in your programs, simply divide the angles by 2 when sending, and multiply them by 2 after receiving. Examples:

```
HEADING VAR WORD

HEADING = 320
SEROUT 15\14, 0, ["H", HEADING/2]
SEROUT 15\14, 0, ["?"]
SERIN 15\14, 0, [HEADING]
HEADING = HEADING * 2
```

Load the various example programs included on your diskette into the editor program to get a better feel for the BS2-IC to PIC communication.

Command list

F distance(word) - Go forward
Returns: nothing

This command tells Descartes to move forward, monitoring its wheel encoders and throttling its motors to keep the robot on track. Descartes will automatically keep itself on its assigned target heading. This target heading, as well as the robot's own heading, are reset to 0 upon power-up. The target heading is automatically updated after a turn command. The target heading can also be changed directly by sending a 'set target heading' command, described below. If this target heading is changed while Descartes is in motion, Descartes will automatically choose the shortest direction to the new heading and turn 'on the fly.'

Descartes will accept one or two byte variables for distance to travel. When Descartes nears its destination, it will slow down and stop. If a destination high byte of 255 is sent, Descartes will travel forward indefinitely. Each distance number is equivalent to 1/5 (0.2) centimeters. As with all commands, Descartes can accept and send serial data while it is in motion. If another motion command is sent before Descartes reaches its destination, the new command will override the old one. Because of this feature, your programs must wait until the previous motion command has finished before sending new commands if the commands are to be performed sequentially. There are at least three ways to do this, see the **# - send stop bit** command below.

```
SEROUT 15\14, 0, ["F", 200]
```

B distance(word) - Go backward
Returns: nothing

This command is similar in function to **F distance(word)** - Go forward

L angle(byte) - Turn left
Returns: nothing

If Descartes is stopped, this command will cause it to turn left to the angle sent. If the angle sent is 20, Descartes will turn left 20 counts, or 40 degrees, from its current heading. This turn will be performed on its own central axis, by running the right wheel forward and reversing the left. If Descartes is in motion when this command is sent, only the target heading will be adjusted. This means that Descartes will track to the new heading 'on the fly' if it is currently executing an "F" or "B" command. If Descartes is operating in direct motor control mode, this command will be ignored. Note that Descartes will come to a stop when either wheel has reached the destination distance, so turning 'on the fly' will shorten the overall run. Also note that Descartes will take a bit of time to track to the new heading, so the ending run location will not be a perfect vector from the point at which the command was sent.

```
SEROUT 15\14, 0, ["L", 20]
```

R angle(byte) - Turn Right
Returns: nothing

This command is similar in function to **L angle(byte)** - Turn left

H angle(byte) - Set heading
Returns: nothing

Descartes will set its own internal heading variable to the angle provided. Descartes will also set the target heading to the same angle. If you wish to maintain a different target heading, it will have to be resent with the command below:

```
SEROUT 15\14, 0, ["H", 90]
```

T angle(byte) - Set target heading
Returns: nothing

Descartes will set its own internal target-heading variable to the angle provided. If this command is sent during a motion command, Descartes will turn to this angle while running, or 'on the fly.'

```
SEROUT 15\14, 0, ["T", 45]
```

M code(byte or four bits) - Start direct motor control.
Returns: nothing

This command will cause Descartes to enter direct motor control mode. The direction of motion of the wheels is determined by the four least significant bits of the byte sent. This coding is identical to the coding used in the Pocket-Bot miniature robot. With the BS2-IC's SEROUT command, the four motor control bits can be represented by a % sign followed by the four bits in binary format. Example:

```
SEROUT 15\14, 0, ["M", %1010]
```

The above command will cause both motors to go forward. The first two digits represent the left motor, and the second two digits represent the right. A '00' for either motor will turn it off. A '10' for either motor will set it to forward motion. A '01' will set it to reverse. '11' will cause the motor to go forward, unlike Pocket-Bot where '11' causes the motor to turn off. The speed at which the motor turns is determined by the left and right motor speed settings, described below. Individual motor speeds and directions can be directly controlled with this command, allowing very smooth, analog turns even at slow speeds. If you wish to create your own encoder monitoring or arc drawing routines, this command coupled with the wheel distance request and wheel velocity request commands is for you. This command also allows for easy porting of Pocket-Bot code for use with Descartes. Instead of using the OUTA command, use the SEROUT command with the same bit coding.

S speed(byte) - Set speed
Returns: nothing

Descartes will set its operating speed for both left and right wheels to the number provided. Valid speeds are 0-32. If a speed greater than 32 is sent, Descartes will default to a speed of 32. Note that if a very low speed is sent, Descartes' wheels may not have enough power to turn. Under these circumstances, one of two things will happen: In direct motor control mode, the wheel will simply stop moving. This is useful for line following and light/dark seeking behaviors. In a programmed motion command mode, Descartes will try to compensate for what it thinks is a stalled wheel, and will give that wheel full power until it moves again making every effort to reach its destination. This will cause very erratic motion as well as inaccurate encoder counts. For this reason, it is recommended to use a value of 10 or higher for programmed motions.

```
SEROUT 15\14, 0, ["S", 20]
```

< speed(byte) - Set left motor speed
Returns: nothing

Descartes will set the left motor speed to the number provided. This command is only useful in direct motor control mode.

```
SEROUT 15\14, 0, ["<", 12]
```

> speed(byte) - Set right motor speed
Returns: nothing

Similar in function to **< speed(byte) - Set left motor speed**

X - Stop robot
Returns: nothing

Descartes will stop its motion, clearing any current motion command in progress. The target-heading variable will not be changed.

```
SEROUT 15\14, 0, ["X"]
```

C - Clear distances traveled.
Returns: nothing

Descartes' internal distances traveled for each wheel will be cleared to 0. If Descartes is in forward or backward motion during this command, the motion command will in effect be restarted.

```
SEROUT 15\14, 0, ["C"]
```

I - Initialize (reset) PIC
Returns: nothing

Descartes' PIC variables will be reset to power-on conditions. This stops the robot, clears the heading, target-heading, distances traveled, and direct motor control bits, and sets the speed to the default of 15.

```
SEROUT 15\14, 0, ["I"]
```

? - Request heading
Returns: angle(byte)

Descartes' PIC will return its current heading. Use the SERIN command immediately after the SEROUT command to receive the angle. Remember that a returned value of 0 is 0 degrees, 90 is 180 degrees.

```
HEADING VAR BYTE

SEROUT 15\14, 0, ["?"]
SERIN 15\14, 0, [HEADING]
```

1 - Request A/D port #1 current value
Returns: value(byte)

Descartes' PIC will perform an analog to digital conversion of the voltage level at port 1 and return the value. 0 volts is a value of 0, 5 volts is a value of 255. When using the CdS cells, more light hitting the cells will cause a lower resistance in the cell, and will return a lower value. Darkness will return a higher value.

```
P1 VAR BYTE

SEROUT 15\14, 0, ["1"]
SERIN 15\14, 0, [P1]
```

2, 3 and 4 - Request A/D port #2, 3 or 4 current value
Returns: value(byte)

Similar in function to command **1** above.

[- Request left wheel distance traveled
Returns: distance(word)

Descartes will return the distance traveled of the left wheel since the last time the distance was reset. Motion commands such as **F, B, L, and R** reset this distance at the start of the motion.

```
L_DISTANCE VAR BYTE

SEROUT 15\14, 0, ["["]
SERIN 15\14, 0, [L_DISTANCE.LOWBYTE, L_DISTANCE.HIGHBYTE]
```

] - Request right wheel distance traveled
Returns: distance(word)

Similar in function to command **[** above.

(- Request left wheel instantaneous velocity
Returns: velocity(byte)

Descartes will return the instantaneous velocity of the left wheel. This number will range from 1 to approximately 25. It will also be very erratic because of the pulsing of the motor voltage, bouncing of the wheel over bumps, and other factors. If you incorporate a routine to average this number over time, you may end up with something useful. It is an internal variable used by Descartes' assembly code PIC program and is provided so that your user programs can be written to detect a stalled wheel (a value of 1.)

```
L_VELOCITY VAR BYTE

SEROUT 15\14, 0, ["("]
SERIN 15\14, 0, [L_VELOCITY]
```

) - Request right wheel instantaneous velocity
Returns: velocity(byte)

Similar in function to command **(** above.

- Request stop state
Returns: stopped(byte)

Descartes will return a byte value of 0 if the robot is currently in motion, or a 1 if it is in the stop state. This command is not useful to detect stalled wheels, as the robot still thinks it is in motion. This command is useful to detect when a motion command has completed execution, so that the next command in sequence can be sent. The example programs provide examples of two methods for doing this. One is to send this command in a loop, continually looping until the returned value is a 1. The other is to install the optional P7 resistor that connects the PIC's encoder IR LED drive pin to the BS2-IC's port 7. Then port 7 can be checked instead of communicating with the PIC. The PIC will turn off its IR LED's by allowing this line to rise to 5 volts when not in use, monitoring the encoders. This saves battery power. The PIC will ground this line, causing the IR LED's to light when it wants to count encoder pulses. A third way of monitoring whether or not the robot has completed a motion command is to request the distances traveled on the wheels and compare to the distance value sent originally. If one or both distances traveled equals or exceeds the distances sent, the robot has stopped. You may have a special circumstance in which this method is better for you.

```
STOPPED VAR BIT

SEROUT 15\14, 0, ["#"]
SERIN 15\14, 0, [STOPPED]
```

If STOPPED = 1, the robot has stopped. Only a bit value needs to be allocated for this variable, the other bits sent will be ignored.

V - Request current installed PIC version
Returns: version(byte)

Descartes will return its internal PIC version number, multiplied by 10. For example, a returned byte value of 13 is PIC version 1.3. The example program VERSION.BS2 will perform this query and DEBUG the version value to the screen for you.

```
VERSION VAR BYTE

SEROUT 15\14, 0, ["V"]
SERIN 15\14, 0, [VERSION]
DEBUG "Version number: ", DEC VERSION
```

BS2-IC Programming Resources

Please refer to the Parallax, Inc. BS2-IC documentation for more information regarding programming the Parallax BS2-IC computer. Parallax, Inc. documentation is provided on your diskette with permission, as well as the Parallax downloader program, "STAMP2.EXE." Also visit the Parallax web site at www.parallaxinc.com for the most recent BS2-IC support software, including Microsoft Windows 95 programming files.

Back to the Table of Contents

Diversified Enterprises
158 Aero Camino Rd.
Santa Barbara, CA 93117