The CheapBot-14 is the programmable brain of a robot and an inexpensive way to become a roboticist. It’s a flexible robot controller that doesn’t limit you to a single robot design. However, don’t picture your first robot as a giant walking monster. Your first robot will be a small wheeled affair capable of reacting to its environment in exciting ways. After becoming familiar with robot design and behavior, you’ll begin to explore more advance concepts and expand the power and behavior of your robots. Then and only then, you can design your loyal army of robots. So let the CheapBot-14 be your first step to world domination.

Onwards and Upwards,
Your near space guide

**Becoming a Roboticist**
To become a roboticist, you must become familiar with mechanical engineering, electrical engineering, and logical engineering. Robots designed with these three disciplines in mind are usually the most effective robots. Since a roboticist must be three engineers at once, CheapBot-14 focuses on building robots from the ground up. Rather than give you a complete robot kit with a drag and drop programmer, CheapBot-14 saves you money and hones your nascent robotics skills from the ground up.

**The Tools You’ll Need**
While there are literally one hundred tools for soldering, testing, and fixing electronic circuits, you only need a few to make robot. These tools are explained below.

**Safety Glasses**
First and foremost, wear safety glasses. They don’t have to be expensive, but they need to have some wrap-around to protect your eyes from flying wires and hot splashes of solder. While your skin will heal from nicks and burns, your eyes won’t. So get some eye protection.
Figure 2. Wearing cheap safety glasses is far better than trying to make robots blind

Wire Cutters

Figure 3. A small pair of wire cutters lets you clip wires closer to the surface of the robot controller

Wire cutters cut wire to length and to trim the wires of components sticking out of the robot controller. Get a small pair as large pairs are too big to trim wires (or the leads) of components that have been soldered to the CheapBot-14 controller. Don’t use scissors to cut wire and don’t try to trim the plastic insulation from wires with a pair of wire cutters.

When wires are cut, they can often fly away in a snap. Therefore it’s important you wear safety glasses when you clip leads and aim the board away from neighbors.
Figure 4. Be careful when you trim leads. The wires can go flying across the room.

Wire Strippers

Figure 5. Unless they are more expensive automatic strippers, wire strippers have multiple holes for passing wires of different diameters.

Wire strippers are designed to cut the outer plastic insulation of wires without cutting or nicking the wire inside. A very suitable pair of wire strippers has multiple holes and each has a different diameter. The hole diameters are designed to let the copper conductor inside a wire pass unscathed while slicing the plastic outer jacket. The outer plastic jacket is called insulation and it prevents the copper conductor inside the wire from making electrical contact with other wires in the CheapBot-14.

Before using wire strippers, you need to know the diameter of the wire you plan to strip. The wire inside the CheapBot-14 kit is 24 gauge. A larger diameter wire has a smaller gauge number.
Look closely at the holes in the wire strippers and you’ll see each one is labeled with the gauge diameter of the wire it is designed to strip. Open the wire strippers like a pair of pliers and place the wire to be stripped inside the proper opened hole in the strippers. Then close the strippers around the wire and pull the wire through by hand.

Figure 6. After closing the wire strippers on the wire, pull the wire from the back side.

Figure 7. There should be no nicks in the wire after you strip its insulation.
Soldering Iron

Figure 8. The soldering iron in its stand and with its sponge. Never lay a hot soldering iron on the table, always use a stand. And always keep the sponge damp.

The traces on the CheapBot-14 PCB are 10 mils wide. If too much heat is applied to them, they along with the pads will lift off the PCB. The fastest way to overheat a PCB is to use a soldering gun. Therefore, never use a soldering gun to assemble your CheapBot-14. Instead, use a pencil style soldering iron. A fine point or narrow chisel point is perfect. If you use a soldering iron with adjustable temperature, then set the temperature at the mid point of its range.

To transfer heat quickly to the work, the tip of the soldering iron needs to be clean and shiny. You can keep the tip clean and free of oxidation by wiping it frequently across the edge of a damp sponge. If the tip of your soldering iron is dark, then it has a coating of oxidized metal on the surface. The oxidized coating prevents heat from transferring quickly to the work and as a result, the entire work area gets hot, including the neighboring traces. Wiping the tip keeps the oxidation at bay. The flux inside your solder can remove stubborn oxide that just wiping can’t remove. Apply solder to the soldering iron tip and let the flux break up some of the oxide before wiping it clean.

After wiping the tip clean, apply a thin coat of solder to the tip to block oxygen from attacking the iron tip. Keep the solder coat thin, or else a large blob of molten solder will transfer to the PCB when you tap the tip to the PCB. A thin coating of solder will help transfer heat while a thick coating will transfer excess solder.

By the way, never jab the hot soldering iron into a damp sponge. The tip should be brushed against the edge of the sponge quickly. Exposing the soldering iron to the cold damp sponge too long creates a thermal shock to the iron. It also needlessly cools the tip and you’ll have to wait for the tip to get hot enough to solder again.
Apply the tip of thin solder to the soldering iron tip, pad, and lead. As the solder begins to melt, run the end of it around the pad, forming a cone of solder that wicks up the lead. Remove the solder before removing the soldering iron or else the solder will be stuck to the soldered pad.

Figure 9. Apply the tip of the soldering iron to both the pad and lead before applying solder to the pad and lead

After the solder cools, clip the lead so it does not stick above the cone of solder around the pad. A well soldered lead will look similar to the image below

Figure 10. A good solder connection is a bright silvery cone

Hot Air Gun

Figure 11. A hot air gun, used to warm heat shrink tubing.
A hot air gun is basically a super hot hair drier. The heating coils inside of one can reach 1,000 degrees, so don’t aim one at yourself or other people. Also be aware of what is down range of the hot air blast coming out of one.

Before soldering wires to a toggle switch, you must slide thin heat shrink tubing over the wire and push it as far away from the bare end of the wire as possible. Heat shrink reduces its diameter by 50% when heated and if the heat shrink is left near the bare end of the wire as it’s soldered, its liable to shrink right on the spot. After the soldered connection is cool, slide the heat shrink over the soldered connection and heat it with the air blast from the hot air gun. You can dangle the work in front of the hot air gun and slowly spin it around to heat all sides. Remember, the heat shrink only shrinks by 50%. Further heating will not shrink it further and is likely to melt the tubing or damage the toggle switch. Ideally the heat shrink is just larger than the connection to be covered so it forms a tight jacket after it is shrunk.

The CheapBot-14 Kit
The CheapBot-14 has four outputs, or ways your robot can affect the world around it. The first two are through the control of its two motors. The other two are through two output ports, Output 0 and Output 1, with three pins each. The three pins are +5 volts, ground, and a connection to the programmable brain. Actuators are plugged into their outputs and code is written to control them.

There are five inputs on the CheapBot-14, or ways the environment can affect your robot. The inputs, called Input 1 through Input 4, also have three pins each. Like the Outputs, the Inputs have +5 volts, ground, and an input to the programmable brain. Sensors are plugged into these ports and code is written to accept input from them.

By adding +5 volts and ground (power) to each input and output, most sensors and affectors can plug right into the CheapBot-14.

Identifying Components
Carefully dump out the bag of parts in front of you onto a table. The largest parts are the flat green printed circuit board or PCB and the two black plastic battery holders. You’ll note that the PCB has white letters and drawings on one side. This layer of writing is called the top silk layer and was silk screened to the PCB when it was manufactured. You’ll also notice that the boxes drawn on the PCB that line up with silver holes (pads) in the PCB. The holes are copper pads with a plating of tin for corrosion control. The wires (leads) of the various components pass through these holes and are soldered into place. A good solder connection makes electrical contact between the leads of each component and creates a strong mechanical connection between the component and the PCB.

The various boxes on the PCB represent components like resistors and capacitors. Therefore, to assemble the robot controller, you’ll have to first identify the proper component, bend its leads to shape (some components won’t need to have their leads bent), align the component with the pattern in the top silk, insert the leads into the proper holes, push the component until its flush with the PCB (some components will stand
above the PCB), bend the leads to hold the component in place, flip the PCB over, solder each lead to its pad, and snip the excess lead off.

Divide up the components you received with this kit according to the pictures below. This will help you properly identify each component.

**Printed Circuit Board**

![Printed Circuit Board](image)

**Figure 12. The CheapBot-14 Printed Circuit Board before adding components**

The printed circuit board (PCB) is a thin fiberglass board with a thin layer of copper on its top and bottom surface. Zones and paths (called traces) are cut into the copper to form paths for current to flow around the circuit. The copper zones and traces are coated in a thin layer of tin for protection from oxidation, so they appear silver in places. A green solder mask protects the area around each pad from the overflow of solder. The mask covers the majority of the PCB, which is why it and the traces and zones appear green. White lettering a boxes printed on top of the green solder mask indicate the location of the components that you will solder to the PCB. Components are soldered to open holes (called pads) in the PCB.

The fiberglass is strong and stable, so components soldered to it will remain in place and electrically connected when the robot operates.

**Resistors**

![Resistor](image)

**Figure 13. A resistor**

The resistor schematic

Resistors are small cylindrical components with a lead sticking out of each end. This shape is called axial. Resistors resist the flow of current in a circuit, and without them, too much current flows through the circuit and damages components. Therefore, we say resistors have resistance. The unit of resistance is the ohm (which is abbreviated with the Greek letter omega, or Ω). Resistors in this kit have four colored bands wrapped around
them. The colors and their order tell the amount of resistance in the specific resistor. The schematic for a resistor is a sharp swiggly shape.

The proper placement of resistors on the PCB are indicated by boxes labeled R1 through R4.

**The Resistor Color Code**
Rather than stamp the value (number of ohms of resistance) of a resistor on its cylindrical body, colored stripes are painted on it. The resistors in the CheapBot-14 have four stripes, but there are resistors with five stripes. Each color signifies a single digit between 0 and 9. The colors and the numbers they are represent are shown below.

<table>
<thead>
<tr>
<th>Color</th>
<th>Digit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black</td>
<td>0</td>
</tr>
<tr>
<td>Brown</td>
<td>1</td>
</tr>
<tr>
<td>Red</td>
<td>2</td>
</tr>
<tr>
<td>Orange</td>
<td>3</td>
</tr>
<tr>
<td>Yellow</td>
<td>4</td>
</tr>
<tr>
<td>Green</td>
<td>5</td>
</tr>
<tr>
<td>Blue</td>
<td>6</td>
</tr>
<tr>
<td>Violet</td>
<td>7</td>
</tr>
<tr>
<td>Gray</td>
<td>8</td>
</tr>
<tr>
<td>White</td>
<td>9</td>
</tr>
</tbody>
</table>

So when you a green stripe, think the digit 5. Now the way the color code on a resistor works is like this. The first two colored bands are read as digits (gold is the fourth band, so read the colored bands for the end opposite the gold band). So a Red followed by a Blue is 26. The third colored band is also a digit, but it represents the number of zeros in the resistance of the resistor. So a Yellow, or 4, in the third band means there are four zeros, or 0000 at the end of the first tow numbers. So take for example resistor R1. It’s a 510 Ω resistor. Therefore, its colored bands are Green, Brown, Brown.

The fourth band of the resistors in your CheapBot-14 kit is always gold. That indicates the resistors have a tolerance of 5%, or that the actual resistance can be as much as 5% different that the indicated value.

**Capacitors**

![Figure 14. Electrolytic capacitors](image1.png)

Schematic of a polarized capacitor
Capacitors come in various shapes and sizes, but in the CheapoBot-14, they appear as yellow lumps or dark gray cylinders. Like the resistor, capacitors have two leads, but in the case of the capacitor, the leads stick out of the same end. This shape is called radial. Capacitors store and release electrons. This ability allows them to act as miniature batteries with small capacities. Capacitors even out the voltage fluctuations that occur when circuits open and close (primarily motors in the robot). By smoothing out voltage fluctuations, the robot behaves more predictably. A capacitor’s ability to store electrons is called capacitance and is measured in units of farads (which is abbreviated with a capital f, or F). However, the farad is a huge unit, so most of the capacitors you’ll see have values in the microfarad range (μF) or the smaller picofarad range (pF). The value of a capacitor is stamped on its body in digits.

The capacitors in the CheapBot-14 are polarized. This means they must be installed with the correct orientation or else they will not provide the proper capacitance to the robot controller. We use polarized capacitors because they are physically smaller than non-polarized capacitors of the same value. The schematic for a polarized capacitor is a flat line “parallel” to a curved line. The flat line is the positive lead of the capacitor and the curved one is the negative lead. Look for the positive sign (+) stamped on the capacitor to identify the positive lead.

The proper placement of capacitors on the PCB is indicated by circles labeled C1 through C3. The plus sign (+) next to one of the pads indicates the position for the capacitor’s positive lead. Look for the white or silver band on the side of the capacitor. It has a negative sign in it to identify the negative of the aluminum capacitor.

**LEDs**

![LED Schematic](image)

*Figure 15. A light-emitting diode (LED)*  
*Schematic of a LED*

Light-emitting diodes are a type of diode the emits light when current flows through them. In this way, they behave like light bulbs, except they will not burn out over the lifetime of the robot controller. To protect LEDs from excessive current, each one is connected to a resistor. LEDs are polarized devices, if they are soldered in backwards they will not operate (but they won’t be damaged, either). The negative lead (called the cathode) is usually the shorter of the two leads. A second and surer way to identify the cathode lead is to look for the flattened side of the LED lens (plastic body). The schematic of an LED is the diode symbol (an arrow touching a bar) with one or two arrows pointing away from the diode (to indicate light is shining out of the diode). The
The arrow of the schematic points to the negative lead or cathode. The other lead of the LED is called the anode.

The proper placement of the LEDs on the PCB are indicated by a circle with the label D1 and D2. The upper-case A (A) next to the LED circle indicates the pad for the positive lead (anode) of the LED.

**The Integrated Circuits (ICs)**

There are four integrated circuits (ICs) in the CheapBot-14. ICs are specialized components consisting of many individual components. ICs are relatively small because the components on it are built into the silicon wafer inside the black epoxy coating of the IC. A series of pins (leads) stick out of the IC to make electrical contact with the circuit.

**PICAXE-14**

The PICAXE-14 is the programmable brain of the CheapBot-14. The CheapBot-14 uses the 14 pin version of the PICAXE, hence the 14 in its name. The programming language used for the PICAXE is BASIC. BASIC is a simple to learn and use computer language. After a few minutes of looking at the BASIC code, you’ll be programming your robotic creation to drive and turn. The schematic symbol of the PICAXE-14 is a box with 14 lines sticking out of it. The lines represent the 14 leads of the PICAXE-14. Each lead has a specific function as described by the datasheet of the PICAXE-14. The top of an IC is identified by the notch at the center top or a dot in the epoxy body near pin #1. The pins are numbered counter clockwise beginning at the top left as illustrated below.
One the CheapBot-14 controller, the proper placement of the PICAXE-14 is indicated by a rectangle labeled U1. The rectangle has a notch, or rectangle drawn at one end to indicate the top of the IC. The top of the PICAXE-14 is marked with a horseshoe-shaped notch cut into the epoxy body.

IC Socket

Rather than risk damaging the PICAXE-14 by soldering it to the PCB, solder an IC socket instead. The PICAXE-14 then snaps into the socket. The other benefit is that if the PICAXE-14 is ever damaged, you can easily replace it and snap a new IC into place.

There is no schematic for an IC socket, just use the schematic for the IC. Electrically it doesn’t matter if the IC socket is soldered right side up or right side down. However, you’ll notice that the IC socket has a notch cut into it to indicate its top. It’s a good idea to align the notch of the socket with the notch on the PCB top silk. That way you don’t have to consult the CheapBot-14’s assembly directions to plug in the PICAXE-14.
Voltage Regulator

A simplest IC in the CheapBot-14 is the three lead voltage regulator. Battery power is unstable. By that, I mean batteries produce a voltage that decreases over time as the battery is discharged. The PICAXE wants a constant five volts and not the gradual six volts the battery will produce. To drop the six volt battery pack down to a constant five volts, the CheapBot-14 uses a voltage regulator. There are three leads in the voltage regulator, the center one is the ground, or zero volts. The other two are the input voltage from the battery and the other is the constant five volt output. The metal tab on the top of the voltage regulator is a heat sink that keeps the regulator cool (heat is produced by the regulator as it works to keep voltage constant). The schematic symbol of the voltage regulator is a box with three lines sticking out of it.

The proper placement of the voltage regulator is indicated by a rectangle labeled U2. The side of the rectangle with the double line indicates the side of the voltage regulator with the heat sink.

H-Bridge

Figure 20. The H-Bridge used in the CheapBot-14  A schematic for the h-bridge
It takes voltage and current (combined together this is power) to run a motor. The PICAXE can only provide five volts at 20 milliamps of current to the robot motors. This is fine for components like LEDs, but is simply too little power to operate the motors of your robot. The power to operate the motors comes from the second battery pack and controlled by the two H-Bridges. The small amount of current the PICAXE-14 can produce is strong enough to tell the H-Bridges how to control the up to one amp of current robot motors may need. So you can think of the H-bridges as smart switches.

The schematic symbol of the H-Bridges is a rectangle with seven lines coming out of it. Only six of the H-Bridges’ leads are needed so the seventh is left unconnected. Pin #1 is the leftmost pin in the picture above.

The proper placement of the H-Bridges is indicated by the two rectangles labeled U3 and U4. The pad closest to the PICAXE is pad #1 and that’s the rightmost lead in the image above. In the latest PCB design, the #1 pads are marked.

**Headers**

![Figure 21. The two headers used in the CheapBot-14, the right-angle motor header and the three pin programming headers](image)

To make it easy to attach motors to the CheapBot-14, there are two headers. The headers are two pin metal connections bent to a right angle. There’s a third header on the CheapBot-14 PCB, but this one is three pins wide and the pins are straight. This header is where the PICAXE-14 is plugged into the computer for programming. The schematic symbol for the headers are boxes with two or three lines sticking out of them.

The proper placement of the headers on the PCB is indicated by three rectangles labeled J1 through J3 (J1 is the three straight pin programming header and J2 and J3 are the right-angle motor headers).
Receptacles

Figure 22. Two views of receptacles  
Schematic for a three pin receptacle

Receptacles are female versions of headers. They are places to plug the cables from inputs (like sensors) and outputs into the CheapBot-14 robot controller. The design or receptacles makes it easy to switch cables around. The schematic symbol for the receptacles is a rectangular box with three lines sticking out of it. The lines represent where +5 volts, ground (0 volts), and input/output I/O pins from the PICAXE-14 are connected. Each row in the receptacles has three pins and this allows an input or output device to interface to the PICAXE-14 while also receiving the power it needs to operate.

The proper placement of the receptacles on the PCB is indicated by the text, Input and Output. Each receptacle’s connection to a PICAXE-14 is labeled with the I/O pin it connects to. Therefore, you’ll notice the input receptacles connect to I/O pins 0 through 4 and the output receptacles connect to PICAXE I/O pins 0 and 1.

Switch

Figure 23. A single-pole, single-throw toggle switch  
The schematic of a single-pole, single throw switch

A switch is a mechanical device that controls the flow of current. When opened, current doesn’t flow into the circuit and the circuit doesn’t operate. There is no switch to solder into the PCB. Instead, there are two wires soldered into the PCB and they run up to the switch. This allows you to mount the switch to the robot in a convenient location. There are many kinds of switches, but the ones used on the CheapBot-14 robot controller are
single pole, single throw (SPST) switches. Their schematic is a line being broken open by the switch. The name for the physical switch is a toggle switch and its bat (lever) is used to turn on and off the robot. Indicator LEDs on the robot controller verify when the switch is applying power.

**Heat Shrink Tubing**

![Heat shrink tubing](image)

**Figure 24. Heat shrink tubing before it is shrunk**

Heat shrink tubing is not an electronic component; it’s a plastic covering for bare wires. After soldering wires to the leads of the switch, you’ll slide a short length of heat shrink tubing over the soldered connection and then shrink it tight with a hot air gun. This replaces the insulation that was removed from the wires to make the connections. A hair dryer probably does get hot enough to shrink the tubing, but in a pinch, a soldering iron can shrink the tubing if the heat is applied carefully as not to burn the tubing.

**Battery Holder**

![Battery holder](image)

**Figure 25. A four cell battery holder**

There are two battery holders attached to the CheapBot-14 PCB (the battery packs attach to the pair of pads on both sides of the logic power LED. The black wire attaches to the pad marked with a G). The first battery pack provides power for the PICAXE-14 and the sensors (and is called logic power) and the second one provides power to operate the
motors (and is called motor power). A separate battery is used for the motors because their operation can create fluctuations in battery voltage that the PICAXE-14 doesn’t like to see (they can make the PICAXE behave erratically). There is no schematic for a battery holder, but there is for a battery. The symbol is historically based on the fact that the first batteries were piles of two kinds of metal plates. The wider plate represents the positive terminal of the battery and the narrower plate represents the negative terminal of the battery.

**Wire**

![Figure 26. Wire.](image)

Wire is a metal conductor (usually flexible) covered (usually) in an insulating jacket. The jacket is normally a soft plastic and easier to cut than the wire inside. The conductor inside is usually copper coated in a thin layer of tin for protection from oxygen (corroded copper does not make a good electrical connection). The color of the insulation is irrelevant to the functioning of the wire (electrons don’t have color and could careless the color of the wire they travel through). However, the color of the wire is often used to indicate the function of the particular wire. A red wire is often used to indicate a wire connected to the positive voltage of a battery and a black or green wire often indicates the wire is connected to ground or the negative terminal of the battery.

The conduct inside a wire consists of a single strand or multiple stands twisted together. A single conductor is stiffer than multiple stands and is called a solid wire. Multiple twisted wires are referred to as stranded wire and they flex and bend more easily and will less likelihood of breaking from use.

The diameter of wire is called its gauge and given a number. The smaller the diameter of the wire the larger its gauge number. The wire used in the CheapBot-14 kit is a stranded 24 gauge wire.
CheapBot-14 Parts List
Now verify you have the following parts in your CheapBot-14 solder kit.

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Part Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCB</td>
<td>Printed circuit board</td>
<td></td>
</tr>
<tr>
<td>U1</td>
<td>PICAXE-14</td>
<td>Programmable microcontroller, the brains</td>
</tr>
<tr>
<td>U2</td>
<td>LM2940</td>
<td>Voltage regulator, converts battery power to +5 volts</td>
</tr>
<tr>
<td>U3</td>
<td>TA8080k H-bridge</td>
<td>Voltage control for first motor</td>
</tr>
<tr>
<td>U4</td>
<td>TA8080k H-bridge</td>
<td>Voltage control for second motor</td>
</tr>
<tr>
<td>R1</td>
<td>680 ohm resistor</td>
<td>Limits current to motor power LED indicator (blue, gray, brown, gold)</td>
</tr>
<tr>
<td>R2</td>
<td>680 ohm resistor</td>
<td>Limits current to logic power LED indicator (blue, gray, brown, gold)</td>
</tr>
<tr>
<td>R3</td>
<td>10,000 ohm resistor</td>
<td>Pull down resistor for programmer (brown, black, orange, gold)</td>
</tr>
<tr>
<td>R4</td>
<td>22,000 ohm resistor</td>
<td>Current limiting resistor for programmer (red, red, orange, gold)</td>
</tr>
<tr>
<td>D1</td>
<td>Red LED</td>
<td>Motor power indicator</td>
</tr>
<tr>
<td>D2</td>
<td>Green LED</td>
<td>Logic power indicator</td>
</tr>
<tr>
<td>C1</td>
<td>10 microfarad capacitor</td>
<td>Power filtering for H-bridge</td>
</tr>
<tr>
<td>C2</td>
<td>10 microfarad capacitor</td>
<td>Power filtering for H-bridge</td>
</tr>
<tr>
<td>C3</td>
<td>22 or 100 microfarad capacitor</td>
<td>Power filtering for voltage regulator</td>
</tr>
<tr>
<td>J1</td>
<td>Three pin straight header</td>
<td>Programming connection to robot controller (solder the short side of the leads to the PCB and leave the long side of the leads sticking up)</td>
</tr>
<tr>
<td>J2</td>
<td>Two pin right angle header</td>
<td>Connection for first motor to robot controller</td>
</tr>
<tr>
<td>J3</td>
<td>Two pin right angle header</td>
<td>Connection for second motor to robot controller</td>
</tr>
<tr>
<td></td>
<td>Two toggle switches (2)</td>
<td>Power control for logic and motor battery packs</td>
</tr>
<tr>
<td></td>
<td>AAA battery packs (2)</td>
<td>Battery holders for logic and motor power packs</td>
</tr>
<tr>
<td></td>
<td>3 by 2 receptacle</td>
<td>Output connectors to PICAXE-14</td>
</tr>
<tr>
<td></td>
<td>3 by 5 receptacle</td>
<td>Input connectors to PICAXE-14</td>
</tr>
<tr>
<td></td>
<td>#24 wire (2 mini rolls)</td>
<td>Hook up wire for switches</td>
</tr>
<tr>
<td></td>
<td>Heat shrink tubing</td>
<td>Covering for soldered connections</td>
</tr>
</tbody>
</table>

Note: R1 and R2 were recently 510 or 1,000 resistors. Your kit could possibly contain one of these old resistors.
Theory of Operation

A programmable microcontroller is an electronic device that reacts to the voltages on its inputs in meaningful ways. The inputs, typically sensors, indicate the current state of the environment to the microcontroller, usually through their presence or absence of voltage. Occasionally, the amount of voltage from a sensor is more important than its presence or absence, but we’ll ignore this for now.

Even if a microcontroller can sense the world, it can’t control the robot unless it has a way to affect the behavior of the robot. The primary way to affect the behavior of a robot is by controlling the way it moves. The PICAXE-14 is capable of causing a robot to travel in a straight line (forwards and backwards), turning (left or right), and stopping by sending two signals to each of the two h-bridges (each motor has its own h-bridge). The signals are voltages, either +5V or ground (0V). The two voltages are treated as logical signals, the +5V is a one (1) and the 0V is a zero (0). When both signals are 0,0 the h-bridge stops the motor from turning. When the signals are 0,1 the motor rotates one way and when the signals are reversed to 1,0, the motor rotates in the opposite direction. If both motors are set to spinning in the same direction, the robot travels forward or backward. If the motors are set to spinning in opposite directions, the robot turns clockwise or counter clockwise in one place.

The h-bridges each have their own capacitor to help maintain a constant voltage. When the motors begin to spin, they draw extra current and this can cause the voltage to drop, or sag. This only lasts a moment, but it can effect the operation of the motors. To prevent this, the capacitors act as tiny batteries and help push up the battery voltage for that moment it wants to sag. While the motors are running at a constant speed, the capacitors recharge in preparation for the next time they will be needed.
In the CheapBot-14, the LED labeled D1 receives current from the resistor labeled R1 and LED D2 receives current through resistor R2. These two LEDs light up when the power switches for the motor and logic power are turned on.

Resistor R4 limits the current flowing into the PICAXE-14 from the PC while it’s being programmed. Resistor R3 is a pull-down resistor that ensures the PICAXE sees 0 volts when the PC is not sending a signal to it.

U2 is the voltage regulator. It takes the six volts from the logic battery pack and drops it to a constant five volts. The regulator is a low drop-out regulator, so it can function until the battery voltage drops to around 5.3 volts. The capacitor C3 helps the regulator maintain a constant voltage. It discharges when the robot controller requires an increase in power faster than the voltage regulator can adapt. When the robot controller is functioning with a relatively constant load, the capacitor recharges.

Headers J1 and J2 are the connection points for the robot’s motors. There is no positive or negative on the J1 and J2. The direction the motors rotate is dependent on the code you write and if you decide to reverse the motor connector, you’ll just change the program to adapt to the change. J3 is the programming jack for the PICAXE-14. One of J3’s pins is marked with the letter G. This is the ground pin and tells you which way to plug the programming cable into the CheapBot-14.

Close to one end of the PCB are two columns of input and output receptacles. These are where you plug sensors into the robot controller. The receptacles are laid out in three columns. The column closest to the PCB’s edge is the ground column. Any pin plugged into this column is connected to ground, or 0 volts. Any two pins plugged into this column are also connected to each other. The middle column is +5 volts, so any pin plugged into this column is connected to +5 volts. And like the ground column, any two pins plugged into this column are also connected to each other. Unlike the outer two columns, pins plugged into the inner column are not connected to each other; each is individually connected to either an input or output of the PICAXE-14. This arrangement of columns allows any sensor to receive power, ground, and connection to the microcontroller through a three pin connector.

**Assembly Directions**
Components are inserted on the printed side (top) of the PCB and their leads are soldered on the unprinted side (bottom). Bend the leads of each component to their proper spacing before inserting the leads in the PCB. Most likely, only resistors will need their leads bent, all other components will probably already have the proper lead spacing. Bend both leads at the resistor body to a 90 degree angle as illustrated below.

![Figure 28. Bend resistor leads very close to their bodies](image-url)
Insert each component into the PCB at the location reserved for that specific component. The reserved location for each one is printed on the PCB in white letters (top silk) and identified by their number, e.g. R1 for the first resistor and C2 for the second capacitor.

It’s generally easier to assemble a printed circuit board if you solder the lowest lying components first. This means you’ll begin by adding the resistors to the PCB. As you add and solder a component, check off its step below.

Note that the following components are polarized and therefore must be inserted in the proper orientation.

Battery pack – Red leads solder to the +V pads and black leads solder to the G pads.
C1, C2, and C3 – align the + mark on the capacitor with the pad marked + on the PCB
U1 socket – align the socket’s notch with notch in top silk
U2 and U3 – Align pin 1 with 1 printed on top silk
D1 and D2 – Align the long lead with the A in the top silk
Note: D1 and D2 can also be aligned with flat of LED lens on pad opposite the A in the top silk

Set components flush with the PCB surface (some components like the capacitors won’t fit flush) and bend their leads apart just slightly to hold the component in place while it is being soldered. Flip the PCB over and apply a tinned soldering iron tip to both the pad and lead simultaneously.

Step by Step Assembly List

Figure 29. The placement of components.
□ R1 - 680 ohm resistor (blue, gray, brown, gold)  
□ R2 - 680 ohm resistor (blue, gray, brown, gold)  
□ R3 - 10,000 ohm resistor (brown, black, red, gold)  
□ R4 - 22,000 ohm resistor (red, red, orange, gold)  
□ J2 - Two pin right angle header  
□ J3 - Two pin right angle header  
□ U1 socket – 14 pin DIP socket  
□ C1 - 10 microfarad capacitor  
□ C2 - 10 microfarad capacitor  
□ C3 - 22 microfarad capacitor  
□ J1 - Three pin straight header  
□ D1 - Red LED  
□ D2 - Green LED  
□ U2 - LM2940  
□ U3 - TA8080k H-bridge  
□ U4 - TA8080k H-bridge  
□ Input (5 by 3 receptacle)  
□ Output (2 by 3 receptacle)

Sanding the Receptacles
The receptacles are rough cut with a saw. Electrically, this is fine and it won’t effect the operation of the CheapBot-14 robot controller. However, it looks ugly. So smooth the sides of both receptacles by running their rough edges over a sheet of sand paper. For the best edges, place the sand paper face up on the top of a table or counter. For even smoother edges, use a finer grade of sand paper for a second sanding after the first.

Cabling Steps
□ Strip one end of all four wires back ¼ inches
□ From the bottom of the PCB, insert the bare ends of wires through the large pads (strain relief pads) marked Switch, bend them over, and insert them into the neighboring smaller pads. See the illustration below.

![Figure 30. A strain relieved wire ready for soldering](image-url)
□ Solder the wire ends and clip excess length
□ Strip ½ inch of insulation from the other ends of the four wires
□ Slide a ½ inch long length of heat shrink tubing over each wire and push the heat shrink down to the PCB so it won’t get hot during soldering
□ Insert each bared wire midway through the hole in the pins of the switches
□ Fold the wire over and twist on itself as illustrated below

![Figure 31. A wire twisted around a switch lead](image)

□ Slide the heat shrink over the soldered switch pin and shrink, covering up all exposed wire as illustrated below

![Figure 32. Heat shrink over soldered switch leads](image)

□ Slide the battery pack wires through their larger strain relief pads and into their solder pads (watch polarity, black leads solder to the pads marked G and the red leads solder to the pads marked +V)
□ Solder the wire ends and clip excess length

**Test Procedures**
Before inserting the PICAXE-14 and flipping the switch, we want to make some measurements to verify the PCB is assembled properly.

□ Inspect the bottom of the PCB and verify there are no soldered connections that overflow their pads to neighboring pads
□ Turn on the switch and make sure there is no continuity between the positive and negative terminals in the battery packs
□ Insert AAA cells into the battery packs and flip the switches, the LEDs should light
Measure voltage across the IC sockets pins 1 (+5V) and 14 (Ground)

- Shut off power and insert the PICAXE-14 (align the notch of the chip with the notch in the socket)
- Connect a programming cable to the programming header and start the PICAXE program Editor
- Set the Editor to the PICAXE-14 and proper Com Port
- Type and download the following one line program

```
DEBUG
```

- The Debug terminal will pop up and you should get a single reply with all variables being zero

**Using the CheapBot-14**
The next step is to mount your CheapBot-14 to a robot body. The cables from the robot’s two motors terminate in receptacles and these plug into the motor output headers (labeled J2 and J3). I/O pins 2 and 3 control one motor and 4 and 5 control the other motor. For the example in these directions, we’re going to plug the right motor into J3 and control it with I/O pins #2 and #3 and plug the left motor into J2 and control it with I/O pins #4 and #5.

The code to drive the motors is simple, but the direction the motors will rotate depends on the orientation of the motor cable receptacles. An example of the code looks like this.

**Right:**
```
HIGH 2
LOW 3
LOW 4
HIGH 5
RETURN
```

**Left:**
```
LOW 2
HIGH 3
HIGH 4
LOW 5
RETURN
```

**Backwards:**
```
HIGH 2
LOW 3
HIGH 4
LOW 5
RETURN
```

**Forwards:**
```
LOW 2
HIGH 3
LOW 4
HIGH 5
RETURN
```
Freeze:
LOW 2
LOW 3
LOW 4
LOW 5
RETURN

These are subroutines and they’re called with the following command.

GOSUB “name of subroutine”

An example would be

GOSUB Forwards

**GOSUB Forwards** jumps the program to the subroutine with the label *Forwards* at the beginning (labels have a colon after them and cannot be the same name as a valid command name – so no label by the name HIGH is allowed). The commands following the label *Forwards* are executed in sequence until the command **RETURN** appears in the subroutine. At that point, the program jumps back to the original GOSUB command and continues executing the rest of the program.

The call to the subroutine named Forwards drives the right motor forward by setting I/O pin #2 low (zero volts) and I/O pin #3 high (+5 volts). It drives the left motor forward by setting I/O pin #4 low (0 volts) and I/O pin #5 high (+5 volts).

If both I/O pins controlling a motor are reversed (a low becomes and a high and a high becomes a low), the motor spins in the opposite direction. If both I/O pins are set low, then the motor stops rotating. There’s no need to set both pins high (which acts as a brake).

Your robot will turn in place if you command one motor to rotate forward and one motor to rotate backwards.

22 November 2009