

Build the Ultimate Robot

Part 5, Megabot

By Michael Simpson



Figure 1

If you are building the Megabot or a similar robot you should have the base assembled at this point. In this article we are going to add the electronics that will make our robot move. I will show you the techniques I used so that you can apply them to your own design.

Step 1 – Program the Actuator IDs

The first thing we need to do is to connect our RX-64 actuators; before we do this, we need to assign the correct ID's to each actuator. You will use the Dynamixel Configurator software to change the default ID to the ID of your choice. To do this you need to do the following:

- Install the USB2Dynamixel Drivers
- Install the Dynamixel Configurator software
- Connect the USB2Dynamixel to your PC
- Set the selector switch on the USB2Dynamixel to RS485

The USB2Dynamixel does not supply power to the actuator so you will need to apply the correct voltage to the power leads on the cable. You can do this by splitting of some power leads from the connector that connects the USB2Dynamixel to the actuator. You can also insert power by creating a power cable that plugs into the other connector on the actuator. The RX-64 manual goes into detail on making this connector. Since I am only programming the ID into the actuators, I made a temporary setup using a breadboard like the one shown in Figure 2. Notice how I have marked the positive and negative power connections on the connector. For more detail on the power leads you can refer to Figure 4. If you have already mounted the RX-64 actuators, simply plug the connector into the actuator you want to program. Just make sure only a single actuator is connected to the USB2Dynamixel when you program the new ID.



Figure 2

Program the IDs as shown in Figure 13. If you decide to use different IDs you will have to make modifications to the various robot routines in order for them to work properly.

Step 2 – Connect the Actuators

Attach a 4-position barrier strip in front of the center actuator as shown in Figure 3. I used two #4 wood screws. Later we will use this to connect power to the wheel assemblies. Next take two RX-64 connectors and connect the two end actuators to the center actuator as shown in Figure 3. Notice how I used a cable clamp to hold the cables close to the base. This is important as you don't want the wires to snag on something and get ripped out.



Figure 3

To connect the actuators on this wheel assembly to the barrier strip, you need to take one of the female connectors included with the RX-64 actuator and attach four, 12" wires. I recommend using the colors shown in Figure 4. I also recommend 22 gauge or larger wire. In Figure 4 the rounded side of the connector is the one facing you. I used a tool that I purchased at RadioShack to crimp the wires. You can use a pair of needle nose pliers as well. You will need to create two of these connectors, one for each wheel assembly.

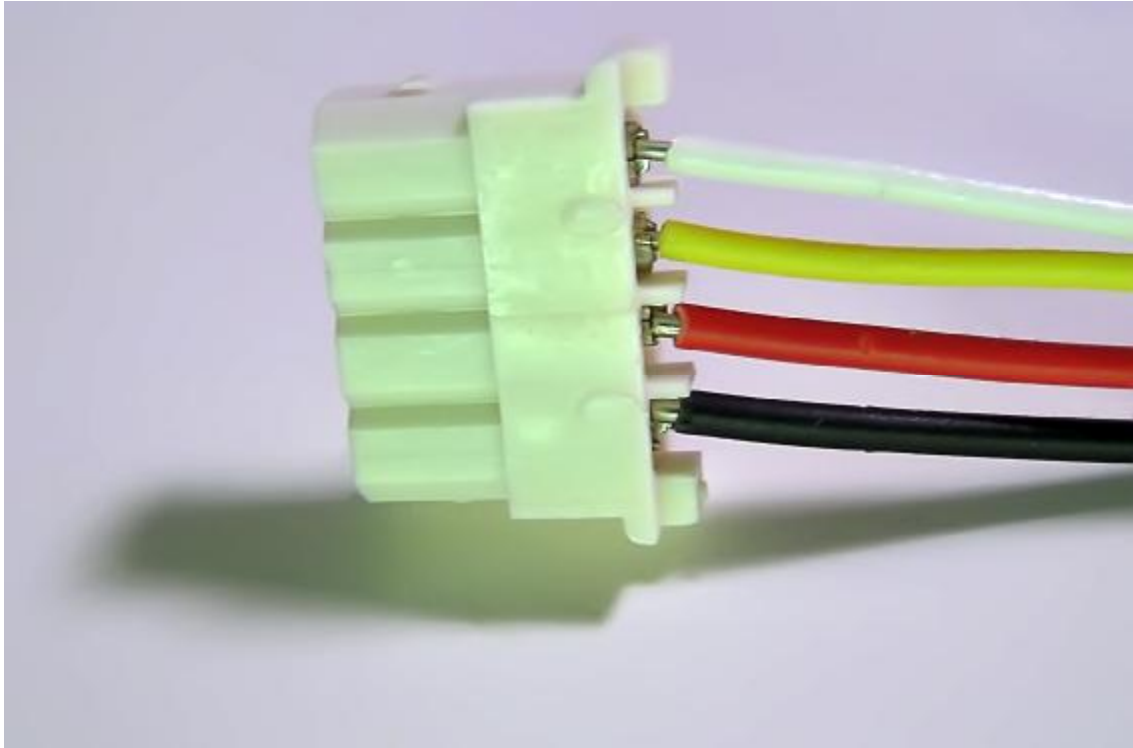


Figure 4

Plug the connector you just created into the wheel assembly. Plug it into actuator 3 on one assembly and 6 on the other. Add a new cable clamp as shown and connect the wires to the barrier strip as shown in Figure 5. Use the order Black, Red, Yellow, and White.



Figure 5

Step 3 – Install the Battery

When I first started this project the power source was the most difficult obstacle I had to overcome. In the course of this project I have come up with two choices. Both involve the use of Li-Ion batteries. Last month I showed you how to use the Ryobi 18V battery shown in Figure 6 to power the Firebot. You can use one for this project as well. You can even use two, one for each wheel assembly.



Figure 6

To show you an alternate configuration I decided to use a battery from a company called atbatt.com. It's the Amstron 133 watt universal battery shown in Figure 7. This battery has a whopping 9000 mAh of power. Its output can be set to 16 or 19 volts. And comes with its own charger. Its low profile and weight makes it perfect for robotic applications. You can even use it to power your laptop for other projects. The battery will cost you about the same as two of the Ryobi batteries.



Figure 7

Place the battery on the underside of the robot between the two wheel assemblies as shown in Figure 8. Place the connectors towards the front of the robot (actuators 1 and 4), make sure you leave enough room to mount a barrier strip and USB2Dynamixel controller as shown in Figure 13. You can use many ways to hold the battery in place; I used some right angle connectors I had on hand, but some hook and loop straps would work just as well. If you do decide to use some hook and loop straps, make sure to use the heavy duty kind available at most home centers. Attach the straps to the base on either side of the battery with wood and finishing washers.



Figure 8

Step 4 – Wire the Battery

The battery comes with a output power connector and you are free to use this one if you wish. Cut off about 3” of cable and attach to a 2-position barrier strip as shown in Figure 9. Since I own several laptops, I wanted to keep the output cable intact, so I cut the plug off of an old AC adapter for my installation. As a last resort you can take the battery down to your local RadioShack and they will help you find a connector.

Attach the barrier strip to the base with #4 wood screws as shown in Figure 9.



Figure 9

Now take two pieces of red 18 gauge wire (12” each) and connect them together with solder. Attach the soldered end to the positive terminal on the barrier strip. Connect each of the other ends to the positive lead on the 4-position barrier strip on each wheel assembly as shown in Figure 10. Repeat this using the two black wires. It’s important that you used at least 18 gauge wires, or you may not supply enough power for your actuators. Notice how I have attached everything close to the base with cable clamps.



Figure 10

Step 5 – Connect the Data Leads

Run a yellow and white wire from barrier strips on the wheel assemblies to each other as shown in Figure 11. Make sure you route the leads through the existing cable clamps and add a couple near the rear of the battery.



Figure 11

Step 6 – Mount the USB2Dynamixel

Add a couple pieces of double stick foam tape to the bottom (Flat Side) of the USB2Dynamixel. Make sure the USB2Dynamixel is close enough to connect to actuator 4 as shown in Figure 12. Also make sure you have room to connect a USB extension cable to the controller. Try dry fitting the controller before attaching it.

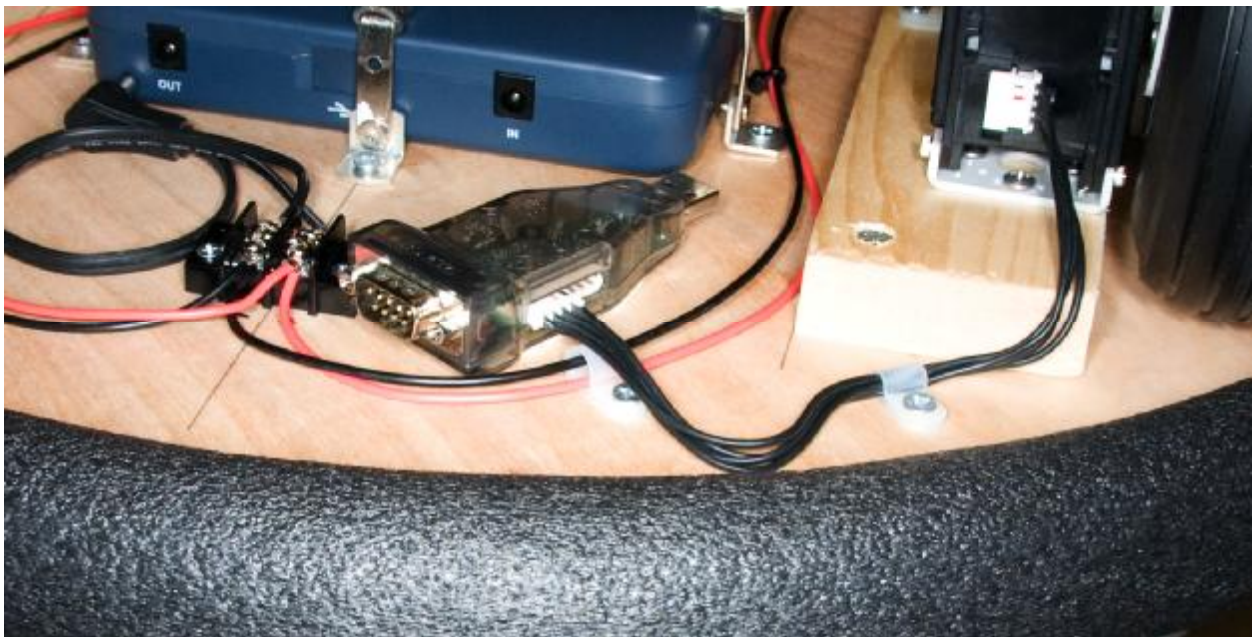


Figure 12

The bottom of the base is now complete. It should be laid out like Figure 13. I have included the IDs for each of the actuators.

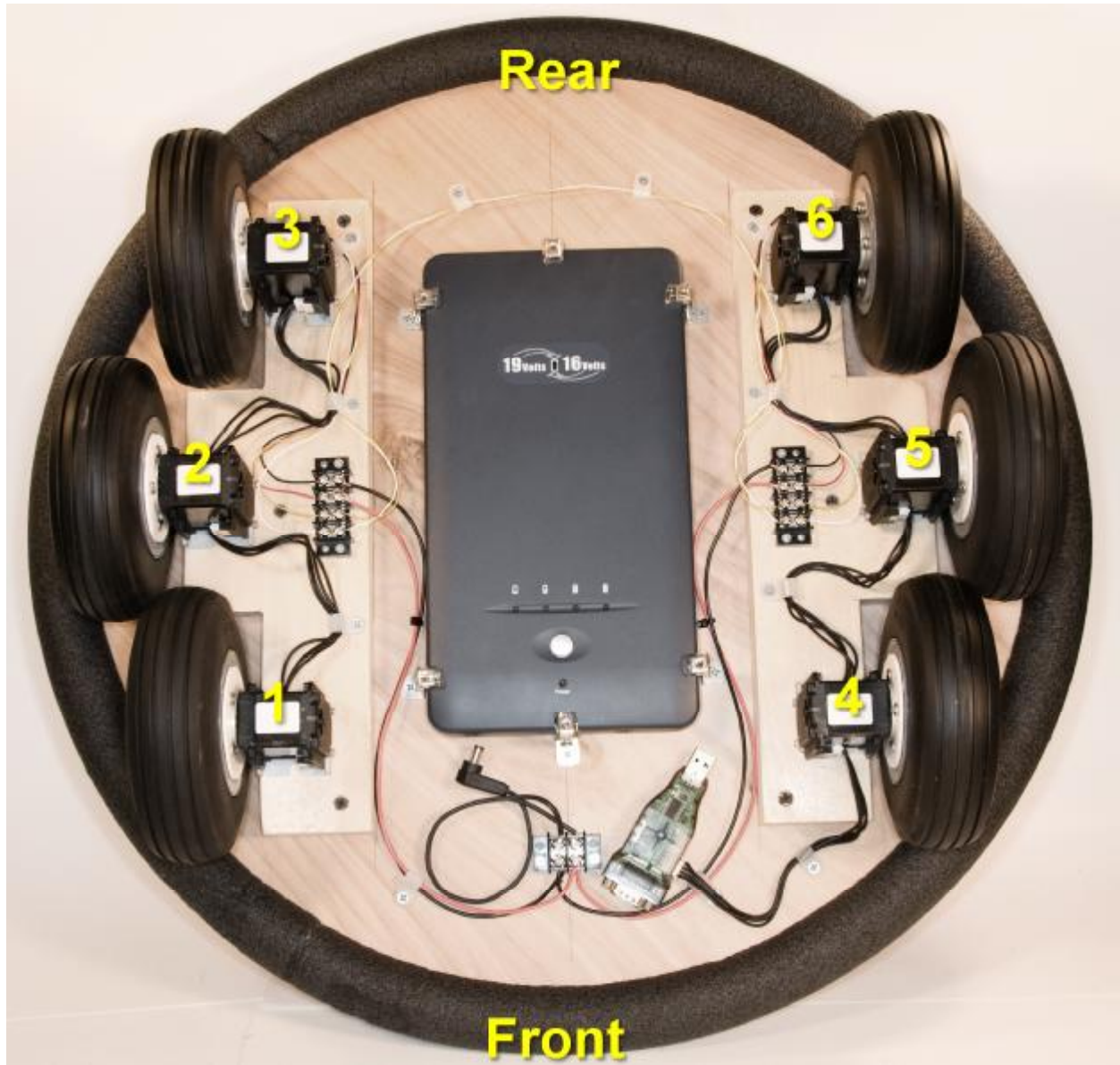


Figure 13

Before we move to the top of the base, I want to say something about using two Ryobi batteries. First I recommend mounting the batteries on the underside of the base. However, this can be problematic if the Megabot will be used on rough and rocky terrain. Connect each battery to the power leads on each wheel assembly. When connecting the data leads from wheel assembly to wheel assembly, you will also need to connect the negative terminal as well.

Step 7 – Wire Switch (Optional)

While removing the plug from the battery works pretty well, you may want to add a switch. Just wire the switch in series with the positive lead on your plug as shown in Figure 14.



Figure 14

The shaft on the switch I used was not long enough to protrude through the 1/2" stock I used for the base. To solve this problem I used a larger drill bit and created a counter sink on the top of the base as shown in Figure 15. There are other ways you can mount the switch. If you decide to do something different I recommend mounting the switch first. Once mounted you can run your wires for the best fit.



Figure 15

Step 8 – Create Stand

The actuators used on the Megabot are extremely powerful. If the Megabot decides to make a move unexpectedly while you are working on the program, something is going to get damaged. A stand is a must. I made the stand shown in Figure 15 out of some scrap wood. It is simply two pieces of 3" x 12" wood connected together with wood screws. It is designed so that the four outside actuators sit on top of these two pieces. I made some small notches so that the actuators sit firmly in place as shown in Figure 16. You can make your stand out of any material you have on hand. You can even use foam board or heavy cardboard. The goal of the stand is to keep the wheels off the ground. If you decide to proceed without the use of a stand, don't say I didn't warn you.



Figure 16



Figure 17

Step 9 – The Cradle

I have seen many robots, both commercial and do-it-yourself, that lack one main component: when using a PC of any kind that has a hard drive you have to create some sort of cradle like the one shown in Figure 18.

The cradle keeps every single bump and bounce from transferring to the laptops internal hard drive. If you don't build a cradle be prepared to replace the hard drive the first time you run the bot into a wall or go over a bump.

The cradle is just 4 pieces of wood connected together with woodscrews. The cradle is large enough to give you about 1" clearance to the lap top on all sides. To actually cradle the lap top, we connect some elastic straps with some wood screws and washers.

The elastic can be purchased at most department stores that have a craft or sewing center. For the laptop I used, only four straps were used as shown in Figure 19. The key is to support the laptop without making it too firm. If you find you need more straps, you can use thinner elastic. If the laptop wants to bottom out when its bounced, you should add a couple more straps.

Notice that I have cut out a portion of the cradle with a jig saw. This is done to reduce weight and provide ventilation for the laptop.



Figure 18



Figure 19

The Laptop

Let's talk a little bit about the brain for Megabot. For most of my medium and large robots I tend to use CE devices. The reason is that CE devices are small and are fairly rugged. They are also reasonably priced. In many cases they can be purchased for less than \$400 new, and less than \$200 used. Well the times are changing my friends.

There is a new influx of laptops hitting the scene. Sometimes referred to as NetBooks or ultra-portable notebooks, they are small, inexpensive and very powerful. The device I have chosen for the Megabot brain is the **Acer Aspire One** shown in Figure 20.

While the Aspire One comes in many flavors, the one I chose has the following features.

- 1.63 Intel Atom Processor
- 1 GB of RAM
- 120 GB Hard Drive
- 8.9" Display
- 3 USB Ports
- External Monitor Hookup
- Built-in Wi-Fi

- Only 2.4 Lbs with battery
- Windows XP Home installed
- Built-in Webcam

Now for the best part, it only cost me \$349. I friend just told me you can pick one up at Wall Mart for \$339. There are other manufactures that have similar offerings.



Figure 20

Using a PC over a CE device is a no-brainer for this application. It's four times faster than most CE devices and offers USB ports capable of utilizing the USB2Dynamixel interface. The USB2Dynamixel interface allows us to communicate with both RX64 and AX12 actuators.

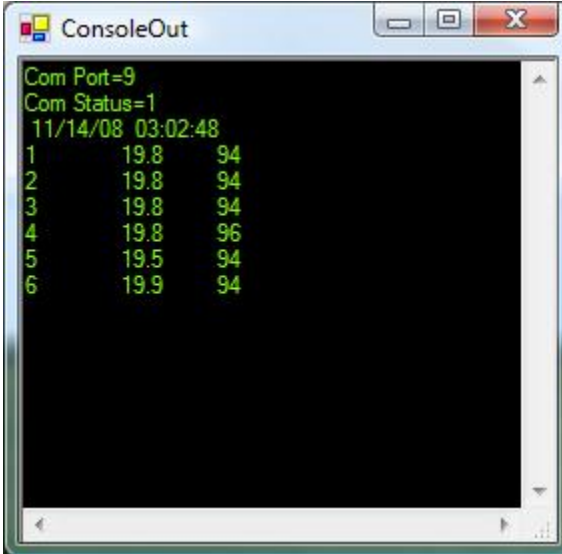
There are versions of the Acer One laptop that come with a solid state drive that make it more suitable for robotics, but most of these come with the Linux operating system which will not work with this project.

Programming the Brain

As with the Firebot in the last article, I will be using Zeus as the primary programming language for Megabot. Zeus is a very inexpensive windows development environment for creating both Windows PC and CE applications.

Let's start with a simple program that will let you test your RX-64 actuators. The program is called **Megabot_Actuator1_DT.exe**

When you run this program on the PC that is connected to the USB2Dynamixel, it will display the output screen shown in Figure 21.



```
Com Port=9
Com Status=1
11/14/08 03:02:48
1      19.8    94
2      19.8    94
3      19.8    94
4      19.8    96
5      19.5    94
6      19.9    94
```

Figure 21

If the program is unable to communicate with the com port connected to the USB2Dynamixel interface, the Status will return a value of 0. This can happen if the software is using the wrong com port. To correct this, create a file called Port and place the number of the com port in that file.

If the individual actuators return a value of -1 and 30, this indicates that the USB2Dynamixel interface is not communicating with the actuator. This can happen if the actuators are not powered, improperly wired, or have the wrong ID.

Once you get good readings, load up the program called **Megbot_Actuator2_DT.exe**. This program will cause the wheels to move in one direction, then reverse. If you don't see the wheels moving, go back and check your power connections.

Now it's time for some fun stuff. Normally I like to create radio interface to my bots as one of the first projects. However, since I did this with the Firebot, I decided to do something a little different this time. Since the Aspire One has built-in Wi-Fi, I decided to write a little client/server program that lets me control the Megbot remotely from my desktop PC.

Load the program called **Megabot_Wifibot_DT.exe** on the laptop used in the Megabot. This is the server program shown in Figure 22. Its job is to listen on the network and wait for commands from the client program. When a command is received, it takes an action. Notice how the server program displays the IP address at the top of the form. In this case it's 192.168.1.201. Keep this in mind, as we will need it later.

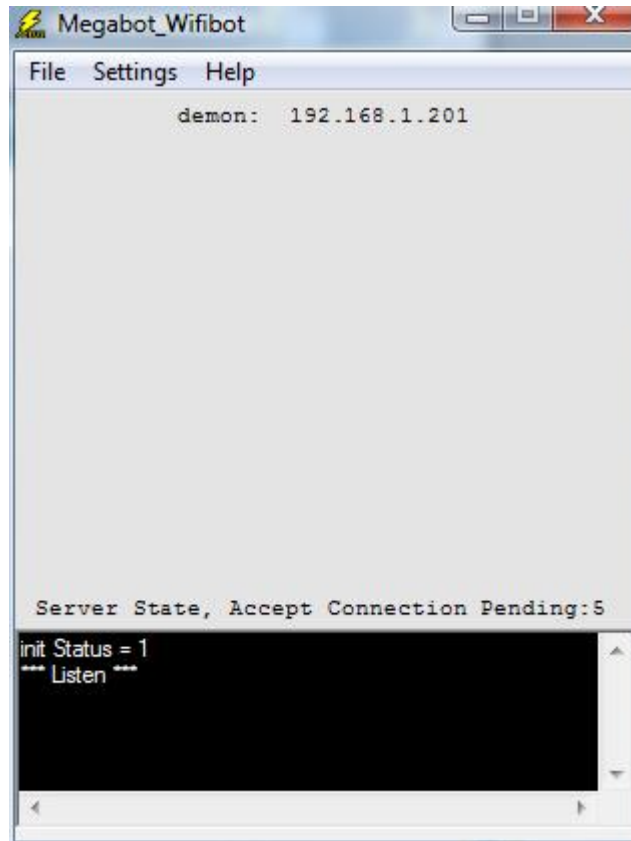


Figure 22

The Aspire One has a webcam. It is possible to utilize this webcam when controlling your robot from a remote location. There are a number of ways to do this. The cheapest and simplest way is to use one of the peer to peer programs like Skype or Yahoo Messenger. All you need to do is to create an account for your Desktop machine and one for your laptop. Before proceeding, you may want to set up these accounts and get the display working on your laptop. It's easy and best of all, it's free.

Now load up the program called **Megabot_Remote1_DT.exe**. This program is the client and it will connect to the server. It's best to overlap the Megabot_Remote1 form on the remote webcam display like I did in Figure 23.

Before doing anything fancy, I really recommend that you test everything out using a stand. Also be advised that the lid on the laptop needs to be up when using the built-in webcam. This means you may need to add a couple more elastic straps to balance things.

Step 1:

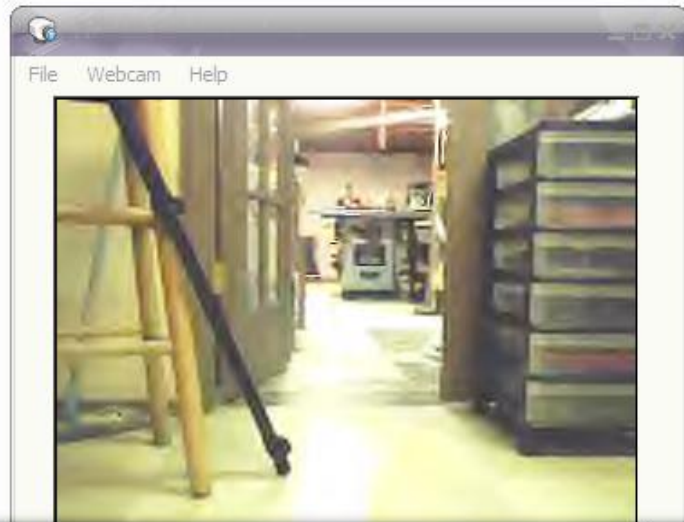
Take the IP address shown on the top of the server (bot) form and place it in the Remote IP: field on the client (remote1) form.

Step 2:

Click the **Start** button at the bottom of the form. This will connect the client to the server. Once connected, both forms will enter state 4, This is the Receive State Pending state. In this state you can issue commands.

Step 3:

You can now hit the buttons to issue the various commands. Try each to make sure they are all working. Once you are satisfied, you can place the Megabot on the floor without the stand. Give yourself plenty of room and start simple.



Megabot_Remote1

File Help

Fast

Slow

Left Stop Right

Slow

Fast

Volts: 19.8

Temp: 90

Stop Connection Made, Receive State Pending:4

Remote IP: 192.168.1.203 Shutdown Bot

This computers name demon
This computers address 192.168.1.201
Connect to 192.168.1.203
Connect to 192.168.1.203

Figure 23

What's Next

Originally I was going to add an interface to the GPS Data Logger I did in a previous article. I wanted to try and experiment to determine if a GPS with a WAAS enabled receiver is accurate enough to keep a robot on a 5' driveway that twists and turns down the front of my property. However, since I have been experimenting with the Wi-Fi remote I want to take it one step further and add a robot arm and a couple of sensors to help automate the remote control system. Next month we are going to add an upper base and attach a robotic arm. I will also show you how to connect some AX-S1 sensors to the system.

Be sure to check out the KronosRobotics website for updates to this project at:

<http://www.kronosrobotics.com/Projects/megabot.shtml>

Parts

Jameco

www.jameco.com

- Toggle Switch – Jameco #76241
- 4-Pos Barrier Strip – Jameco #230990
- 1/4" Cable Clamps – Jameco #173737

Crustcrawler

RX-64

<http://www.crustcrawler.com/motors/RX64/index.php?prod=67>

Treaded Wheels

<http://www.crustcrawler.com/products/rover/wheels.php?prod=28>

Dynamixle Confiurator

http://www.crustcrawler.com/electronics/USB2Dynamixel/software/Dynamixel_Configurator/DXCONFINST1.2.1.0.exe

KronosRobotics

ZeusPro Development Environment

<http://www.krmicros.com/Development/ZeusPro/ZeusPro.htm>

RadioShack

4-Position Barrier Strip
#274-658

2-Position Barrier Strip
#274-656

AtBatt.COM

P133 External Laptop Battery.

<http://www.atbatt.com/product/7901.asp>

Messenger Software

<http://www.skype.com>

<http://messenger.yahoo.com/>