Build-it Series

By Michael Simpson

Figure 1

Build a Dog Detector

As seen in December 2008 of Servo Magazine
Pick up an issue at
www.servomagazine.com

We have several pets in our home including 2 dogs. One of the problems we have is the dogs like to sneak into the kitchen and eat the cat's food. My wife asked me if there was any way we could create some sort of detector that would chirp each time a dog entered the kitchen. Of course I said yes.

The key to this project is to detect only the dog and not a person or cat entering the kitchen. This can be done with two sensors at various heights and a bit of logic to decode the sensors.

I started doing some research and started looking into some of the various sensors available.

Lasers and Photo Detectors

These would probably be the simplest to interface, but require two separate power sources and mounts one for the lasers, and the other for the photo detectors. This way also makes it difficult to relocate the sensor and adjust the heights of the sensors. There are also alignment issues. The small photo detector surface means you can only have a 1/8" variance. For these reasons, I decided this type of sensor is much too finicky for this application.

Passive IR Sensors

These are also know as PIR sensors. This type of sensor monitors the amount of IR energy hitting the internal sensor and changes the state of the output lead when it detects a sudden change in this energy. The sensor shown in Figure 2 can be purchased from Radio Shack for about $10. I actually built the Dog Detector using these sensors and found that they are not well suited for this type of application. First, they are not very directional and even by placing them in a enclosure they still suffer. Also, they tend to be sensitive to sudden changes in IR like a shadow moving across the sensor. After building and testing the Dog Detector based on these sensors, I knew I would have to look into another sensor.

Figure 2

Active IR Sensors

The concept behind active IR sensors is much like that of the PIR sensor. The main difference is that this sensor actually sends out IR energy and measures the amount of that energy reflected back. Many of the sensors of this type modulate the IR energy to give them a certain amount of immunity from outside IR interference. Some of these sensors can actually return range information. The sensor shown in Figure 3 outputs a voltage based on the amount of IR energy that is reflected back. While these sensors are more immune to outside influences, they do have trouble when incandescent lamps are in use. They also have problems with various surfaces, as some do not reflect the IR energy the same way. For instance, a black dog's hair will reflect very little energy back, while a white dog's hair will reflect much more.

After experimenting with this type of sensor, I again decided to look at other sensors.

Figure 3

Sonar Sensor

I have used several sonar sensors in the past and have had good results with many. The problem is that the sensors have a very wide angle of detection. The folks at MaxBotix have come up with a line of ultrasonic range finders. They all use a single transducer so the size of these sensors are quite small. This single transducer configuration also helps keep the price down. You can review all the performance data for the complete line of sensors on their web site at:

http://www.maxbotix.com/Performance_Data.html

For this application I found the EZ4 shown in Figure 4 is a perfect match. I also used the EZ3 with great success.

Figure 4

These sensors also give you the ability to shut them down using a control lead. This is important when using multiple ultrasonic sensors. The EZ4 ultrasonic sensor has 4 ways to interface, which makes these little gems the most versatile ultrasonic sensor you will ever use.

Pulse Width

With this interface you measure the width of a pulse stream returned by the sensor.

Serial

Asynchronous interface at 9600 baud. This is a TTL level single but it's inverted so you can't use it with most microcontroller UARTS. This one also means you have to decode the data so it's not my first choice.

Ping

With this interface you do all the work. You send a signal to the detector and measure the amount of time it takes to come back.

Analog

The sensor will output a voltage based on the range of the reflected ping. This one is my favorite interfaces and as long as you have a microcontroller that has the ability to measure an analog signal, this is the easiest to use. This is the interface we will be using for this project.

Perseus Microcontroller

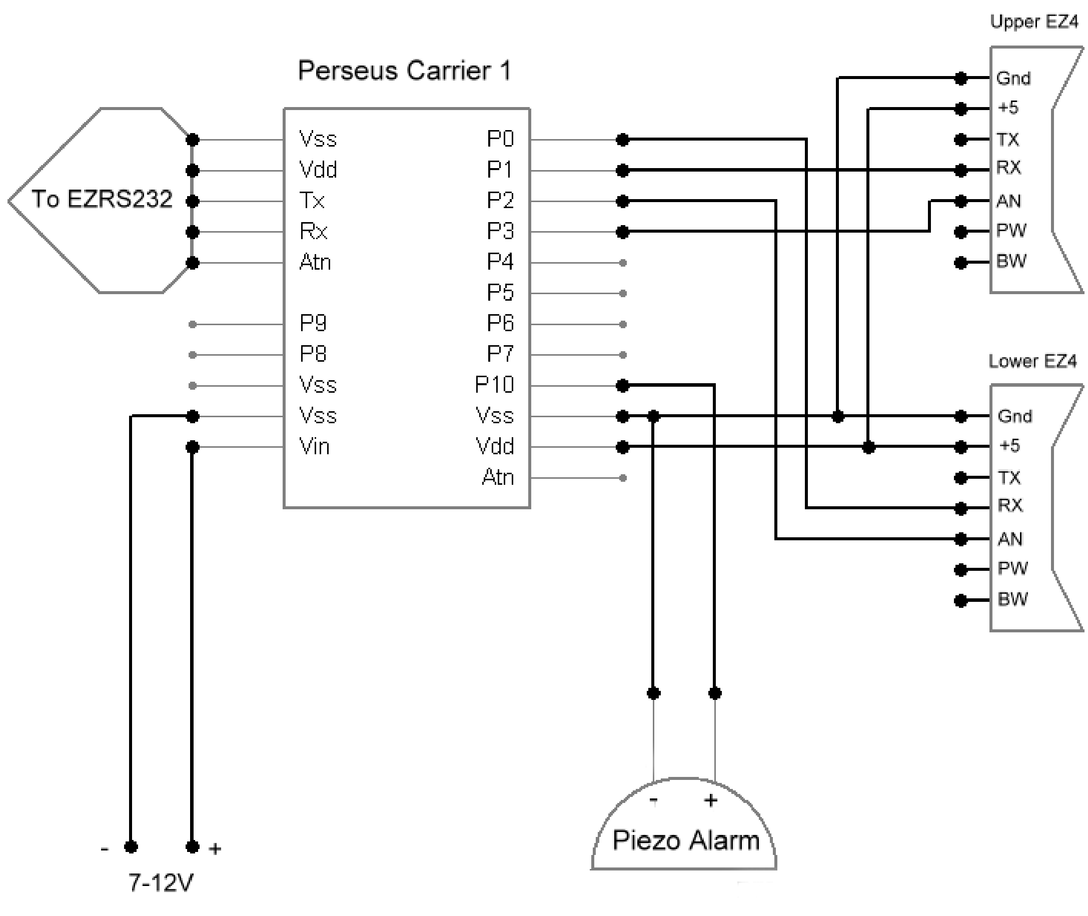
To monitor the two sensors I will be using a Perseus microcontroller and Perseus Carrier 1. Shown in Figure 5, this is a very small and inexpensive controller. To program the controller all that is needed is a $10 EZRs232 driver and free software.

Figure 5

The carrier board, also shown in Figure 5, has a tiny prototyping feature that will allow for easy hookup of our sensors. This carrier also has the ability to add a 7805 voltage regulator to make powering our project very easy.

Perseus Construction

I am going to take you step-by-step through the construction process. I will start with the microcontroller, then show you how to build the main stand for the project. Schematic 1 shows the completed hookup so that you can troubleshoot in case you have problems.



Schematic 1

Step 1

The Perseus Carrier 1 kit comes with complete instructions. You need to build this kit. Make sure you don't install the two headers. When complete, it should look like the carrier shown in Figure 6. The cost of the Perseus Carrier 1 is only $6. If you want to experiment using a breadboard I recommend that you purchase 2 carriers so that you will have one to play with outside the project. You can simply plug the Perseus into whichever carrier you are working on.

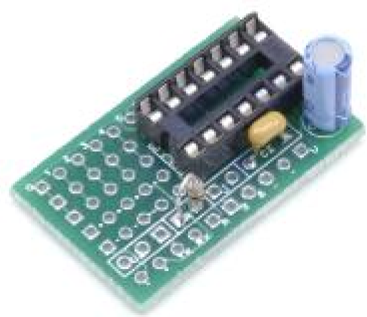


Figure 6

Step 2

You will need to purchase an Option 1 kit for the carrier. This is simply a 7805 regulator and a couple of headers. If you already have a 7805 you won't need this. Attach a 2-pin header to the carrier as shown in Figure 7. This allows you to jumper the regulator. Since we will always be using the regulator on this project, you can simply solder a wire in place of this header. If you do use the header you will need to place a jumper across the two pins.

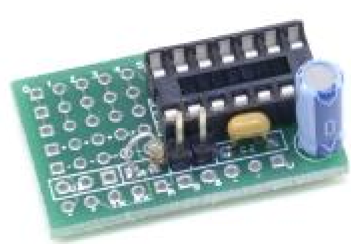


Figure 7

Step 3

Take a 5-pin header and connect it to the carrier as shown in Figure 8. You can break this off from one of the unused headers. This header will be used to program the Perseus at a later time.

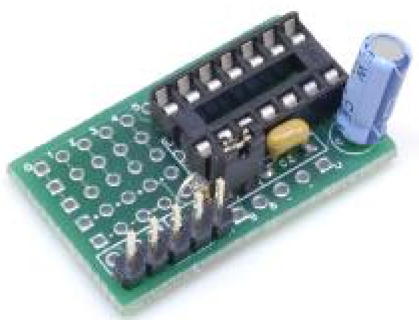


Figure 8

Step 4

Take a coax connector and solder a 2-pin header to the connector as shown in Figure 9. Bend the connector pins closer together to hold the header in place while you solder it. The header does not have to be perfectly straight, but, make sure you have a good solder connection.

Figure 9

Step 5

Now attach the connector/header assembly to the carrier as shown in Figure 10. Pay careful attention to the pins and orientation of the connector.

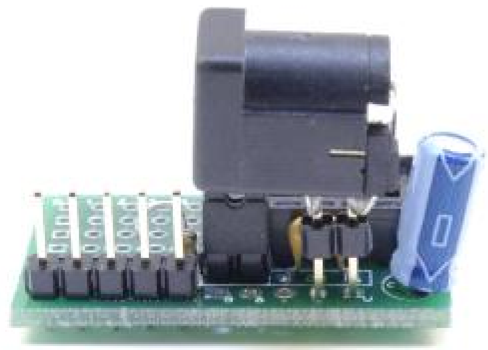


Figure 10

Step 6

Once you solder the header in place, bend it as shown in Figure 11. This bend allows you to plug the AC adapter cord into the board at the same time the EZRS232 adapter is being used.

Figure 11

Step 7

Attach a 3-pin header to the carrier as shown in Figure 12. This step is optional. This header may be used later to add a servo for some sort of active reinforcement.

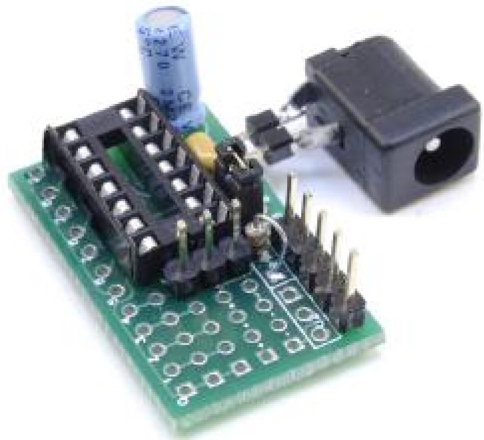


Figure 12

Step 8

Attach a 7805 voltage regulator as shown in Figure 13. I recommend inserting the 7805 just enough to solder the leads in place.

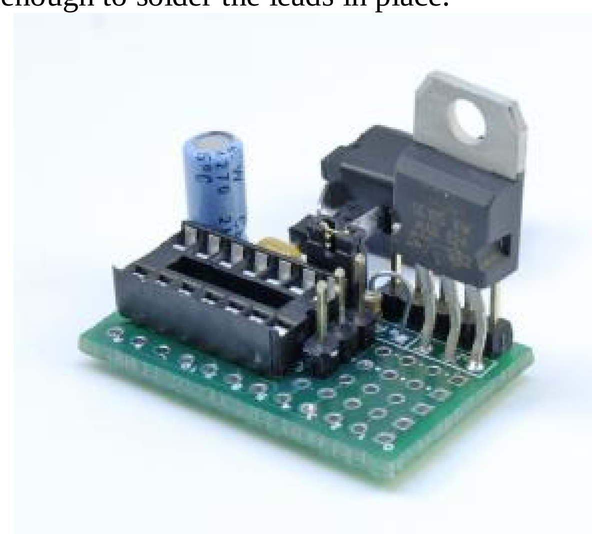


Figure 13

Step 9

Attach a couple of wires to the piezo alarm as shown in Figure 14. I often purchase some multicolored 22 or 24 gauge ribbon cable for this type hookup.

All you need to do is peel off the number of leads needed. Each lead will have a different color, so it will be easy to keep track of the different connections.

Figure 14

Step 10

Attach the other end of the piezo cable to the carrier as shown in Figure 15. The + side of the alarm is connected to port 10 on the carrier. The other side is connected to the pin labeled Vss.

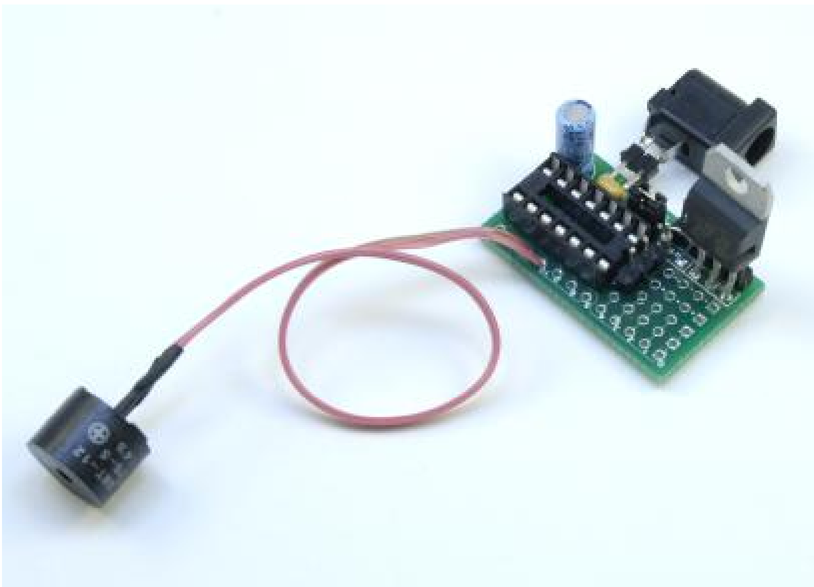


Figure 15

Step 11

Drill 2 holes into a PVC cap as shown in Figure 16a. The center hole should be sized so that the piezo fits tightly into the hole as shown in Figure 16b. I drilled mine just a little smaller than the diameter of the piezo, then enlarged it with a file and knife. The hole near the edge is used to insert the power connector on the AC adapter.

Once fitted, remove the piezo and set aside.

Figure 16a **Figure 16b**

Stand Construction

I built my stand out of 1-1/2" PVC pipe. It's inexpensive and can be painted. For this part of the project you will need the following:

2 pieces of 24" 1-1/2" pipe
1 piece of 2" 1-1/2" pipe
2, 1-1/2" PVC Caps
1, 1-1/2" PVC Cross
1, 1-1/2" Coupling
1, 2-1/2" Machine Screw and nut

Step 12

First, we need to build the base. This will allow the detector to stand free. Cut a piece of 1-1/2" PVC pipe to about 2" as shown in Figure 17a. Take a 1-1/2" PVC cap and drill a 1/4" hole in the center as shown in Figure 17b. Next, drill a 1/4" hole in the center of a 1-1/2' PVC Cross as shown in Figure 17c.

Figure 17a

Figure 17b

Figure 17c

To assemble the base, run a 2-1/2” #6 machine screw down through the inside of the PVC Cap, then the section of pipe, and finally through the hole in the PVC Cross as shown in Figure 18. Place a nut on the end of the screw on the inside of the Cross and tighten. As an option you can drill a 1/2" hole on the other end of the Cross and insert the machine screw through the bottom of the PVC Cross then up through the small pipe section and into the cap. The goal here is to have the cap attached to the Cross so that you can attach a 24” piece of tubing.

Figure 18

Step 13

Peel off 4 leads of ribbon cable about 3’ long as shown in Figure 19a. Strip each lead on both ends. Attach one end to the EZ4 sensor as shown in Figure 19b and Figure 19c. You will use the GND, +5, RX and AN. The RX lead is used to disable the sensor when not in use.

Figure 19a

Figure 19b

Figure 19c

Step 14

Take both of the 24” pieces of pipe and drill a 1/2" hole in each. The hole should be located at 18” from one end and 6” from the other as shown in Figure 20.

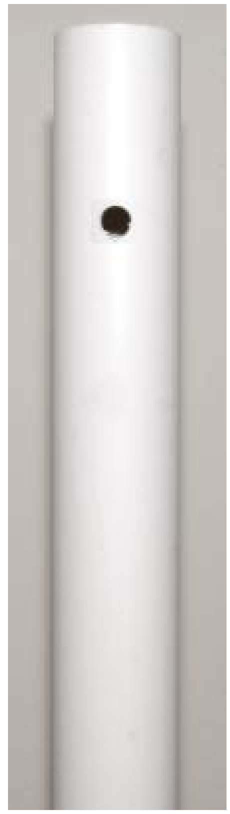


Figure 20

Step 15

Using some double stick foam tape, attach the EZ4 sensor module to the pipe and run the cable through the hole as shown in Figure 21. As an option you can use some small wood screws to attach the module to the pipe. Just mark and drill some small pilot holes into the pipe first.

Figure 21

Step 16

Insert one of the pipes into the PVC Cap on the Base. Insert the pipe so that the sensor is near the top. Pull the cable through the top of the pipe and attach it to the Perseus Carrier as shown in Figure 22.

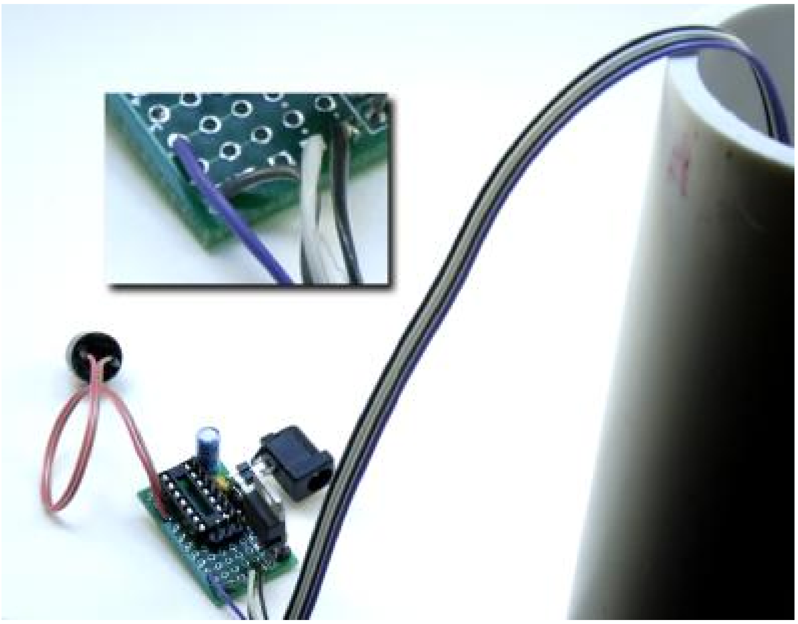


Figure 22

Make the following connections:

- Sensor GND – Negative on port 0 bus of carrier. Shown here with the black wire.
- Sensor 5v – Positive on port 0 bus of carrier. Shown here with the white wire.
- Sensor RX – Port 0 on carrier shown here in gray.
- Sensor AN – Port 2 on carrier shown here in blue.

Please note that your colors will most likely be different than the ones I have shown here. I suggest you create a small chart on paper once you connect the ribbon cable to the sensor.

Step 17

Attach a PVC coupling to the bottom of the other 24" pipe. This is the end farthest away from the sensor. Next run the complete Perseus assembly up through the bottom of this pipe and out through the top.

At this point you should be able to attach the top pipe to the bottom pipe. Take the sensor wires from the upper sensor and attach them to the carrier as shown in Figure 23.

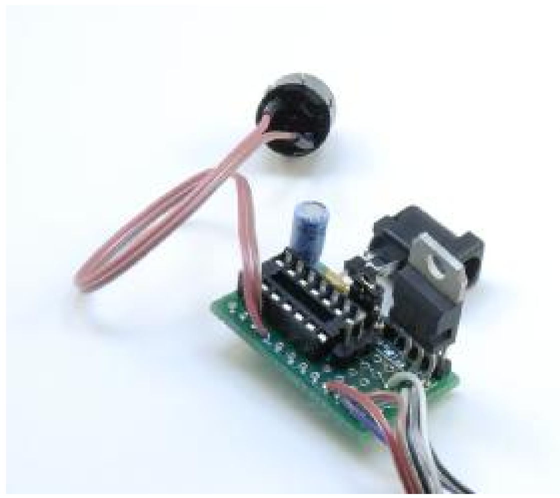


Figure 23

Make the following connections:

- Sensor GND – Negative on port 1 bus of carrier. Shown here with the black wire.
- Sensor 5v – Positive on port 1 bus of carrier. Shown here with the white wire.
- Sensor RX – Port 1 on carrier shown here in brown.
- Sensor AN – Port 3 on carrier shown here in red.

Step 18

If you have not done so already, insert the Perseus chip into the socket on the carrier. The notch faces the center of the board. In order to program the Perseus we need to connect the EZ232 driver to the carrier. The EZ232 and Perseus Carrier 1 are meant to be used on a breadboard. In order to program the Perseus in this application, you need to create a simple adapter by connecting two 5-pin female headers together as shown in Figure 24.

Figure 24

To program the chip in our project, attach the EZ232 to the program header on the carrier as shown in Figure 25. Then connect a 7-12v AC adapter (Center Positive) to the coax connector. At this point you can connect the EZ232 to your PC with a 9-pin cable. Please note that there are a couple more options you can use to program the Perseus. You can build a second carrier and include the standard headers per the included documentation. With the second carrier you can use a breadboard. You can also use an Athena Workboard Deluxe to program the Perseus.

Figure 25

I have included a program called **Doggy1.txt**. The program will monitor the sensors and chirp the Piezo when a dog is detected. Load this program into the Athena compiler and program the Perseus.

Step 19 (Final Step)

Remove the EZ232 and power connector from the carrier and attach the Piezo to the PVC cap as you did before. Next, run the AC adapter cable down through the other hole and then into the coax connector. You can now place the cap on the top of the "Dog Detector".

The Program

The Doggy1.txt program is straightforward. The first thing the program does after initializing the IO ports is to take a set of calibration readings with the **docal** subroutine. This allows you to place the detector in a great many locations without having to change the code. The **docal** subroutine makes several calls to the sensor read routines then sets a couple of variables based on the last reading of each sensor. The value 10 is subtracted from this reading and will be used as the detection point for each sensor.

The Main Loop is where all the work gets done. An AtoD reading is taken every 10 milliseconds and compared against the calibration reading taken earlier. If the current reading is less than the calibration reading the sensor is considered tripped.

Once tripped a slight 50 millisecond pause is observed then 30 readings are taken from the upper sensor at 10 millisecond intervals. These readings again are compared against the calibration values and if triggered, it is assumed a human has passed the detector. If all 30 readings are taken without tripping the upper sensor it is assumed that a dog has passed the detector and the chirp routine is called. After a pause of 2 seconds the whole process starts again.

Normally the RX lead of each sensor is held low to disable them from taking readings. When a reading from a sensor is needed, the RX lead is brought high then after a 10 millisecond delay an AtoD reading is taken from the sensors AN lead. After the reading, the RX lead is again taken low.

How does it work?

Unlike the early experiments using the IR sensors, the EZ4 sensors have proven a real winner. The small chirp from the Piezo is just quiet enough not to annoy or wake you. It is enough to let both you and the dog know that he/she has entered where they should not have.

Going Further

I added a 3-pin header that would make it possible to add some sort of negative feed back to the detector.

The height of the lower sensor should be high enough to let small cats pass if you wish, and low enough to catch the dog. You could use space the sensors to detect small children roaming where they should not.

All the example programs as well as the source are available for download at:
http://www.kronosrobotics.com/Projects/dogdetect.shtml

Parts

Kronos Robotics - www.kronosrobotics.com

Perseus Chip
    http://www.kronosrobotics.com/xcart/product.php?productid=16382
Perseus Carrier 1
    http://www.kronosrobotics.com/xcart/product.php?productid=16390
Perseus Carrier 1 Option 1
    http://www.kronosrobotics.com/xcart/product.php?productid=16391
Piezo Alarm
    http://www.kronosrobotics.com/xcart/product.php?productid=16252
EZ232
    http://www.kronosrobotics.com/xcart/product.php?productid=16167

DiosCompiler
    Free Download from www.kronosrobotics.com

SparkFun – www.sparkfun.com

EZ4 Sonar Sensor
  http://www.sparkfun.com/commerce/product_info.php?products_id=8504

Maxbotix – www.maxbotix.com

EZ4 Sonar Sensor
  http://www.maxbotix.com/Maxbotix__Buy_Now.html#