Introduction to OPEN-ROBOT

OPEN-ROBOT shown with MatchPort® b/g

Background

My name is Abe Howell and I am a mechanical engineer by degrees. However, I have always had a strong interest in electrical engineering and computer programming. I graduated with a Bachelor of Science in mechanical engineering in 1999, a Master of Science in mechanical engineering in 2003, and currently I am working towards a PhD in mechanical engineering with a focus on engineering education. Throughout my career I have had many great opportunities to work in the following capacities: process engineer, project engineer, automation engineer, mechanical engineer, and systems engineer. While working towards my graduate degree I became involved with several K-12 outreach activities. One of these activities helped to introduce middle school students to robotics and several others focused on integrating and leveraging robots to teach concepts in mathematics at the middle school level. I quickly recognized that there are many benefits to introducing students to the world of science, technology, engineering, and math or STEM as it is commonly referred to in literature. I created the NewCDbot Robot as a result of this work. The NewCDbot is robot whose body is composed of recycled Compact Discs (CDs). I designed it in such a way that middle school students could build this robot in about 3 hours and quickly write programs once it was assembled.

Throughout all of my work I always wanted to create a robot that could be utilized at the K-12 level, college and university level. To be capable of spanning the entire educational spectrum this robot would have to be durable enough to be used in a K-12 environment, but also complex and expandable so that university students could use it to learn about microcontrollers, mechanics, circuits, hardware, sensors and programming. I have worked on evolving the design of OPEN-ROBOT across several years while working full time and pursuing my PhD. One of the other key goals was to make OPEN-ROBOT as low-cost as possible, but not loose the flexibility and learning value. In this short paper I want to provide you with an overview of OPEN-ROBOT, but more importantly I want to convey the great flexibility that has been designed into this robot and how you can leverage this in your classroom, after school activities, or robotics research.

Overview of OPEN-ROBOT

I created the OPEN-ROBOT primarily for use in an educational setting and this is why the robot is completely open-source. What I mean by open-source is that all the documentation, software, printed circuit board (PCB) designs, and mechanical design files can be modified and redistributed so long as you give me credit where credit is due. I feel that this is critical in creating an open community of ideas and a knowledge base across disciplines and end uses. I am hoping that colleges and universities will find OPEN-ROBOT useful in their courses and then they will possibly create additional add-on hardware, sensors, mechanical end-effectors, and software for OPEN-ROBOT. This information and knowledge can then be made available to K-12 educators so that they may use it in their classrooms to benefit students.

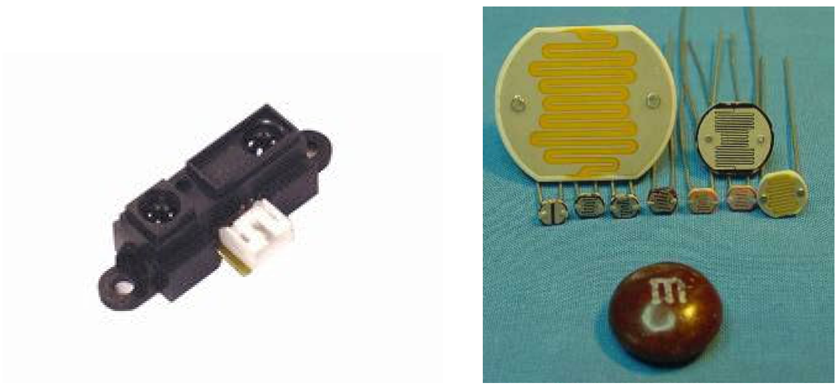

Figure 1. Sharp® GP2D120 Infrared Sensor and various cadmium sulfide photocells.

OPEN-ROBOT has an array of on-board sensors so that it can interact with the environment. The Sharp GP2D120 infrared sensors provide OPEN-ROBOT with the ability to sense obstacles and determine their distance from the sensor face. This means that the distance to an object can be calculated in units of inches, centimeters, or any other unit of length. There are many great learning activities that can be derived from these sensors alone. Students can learn to calibrate their robot's GP2D120s by taking 8-bit obstacle detection readings at incremental distances from the sensor's face and then use this data to create a plot of the sensor's response. From this they can create a regression line and obtain a function that calculates the detection distance in inches or centimeters when a GP2D120 reading is supplied. Please refer to figure#1 above.

The two frontward facing cadmium sulfide cells provide OPEN-ROBOT with ability to measure the amount of ambient light that is incident upon the cell's face. These sensors are sheathed in heat-shrink tubing so that they are more sensitive to the direction of light. This allows OPEN-ROBOT to follow or avoid light. As the light becomes brighter OPEN-ROBOT will return a larger 8-bit reading.

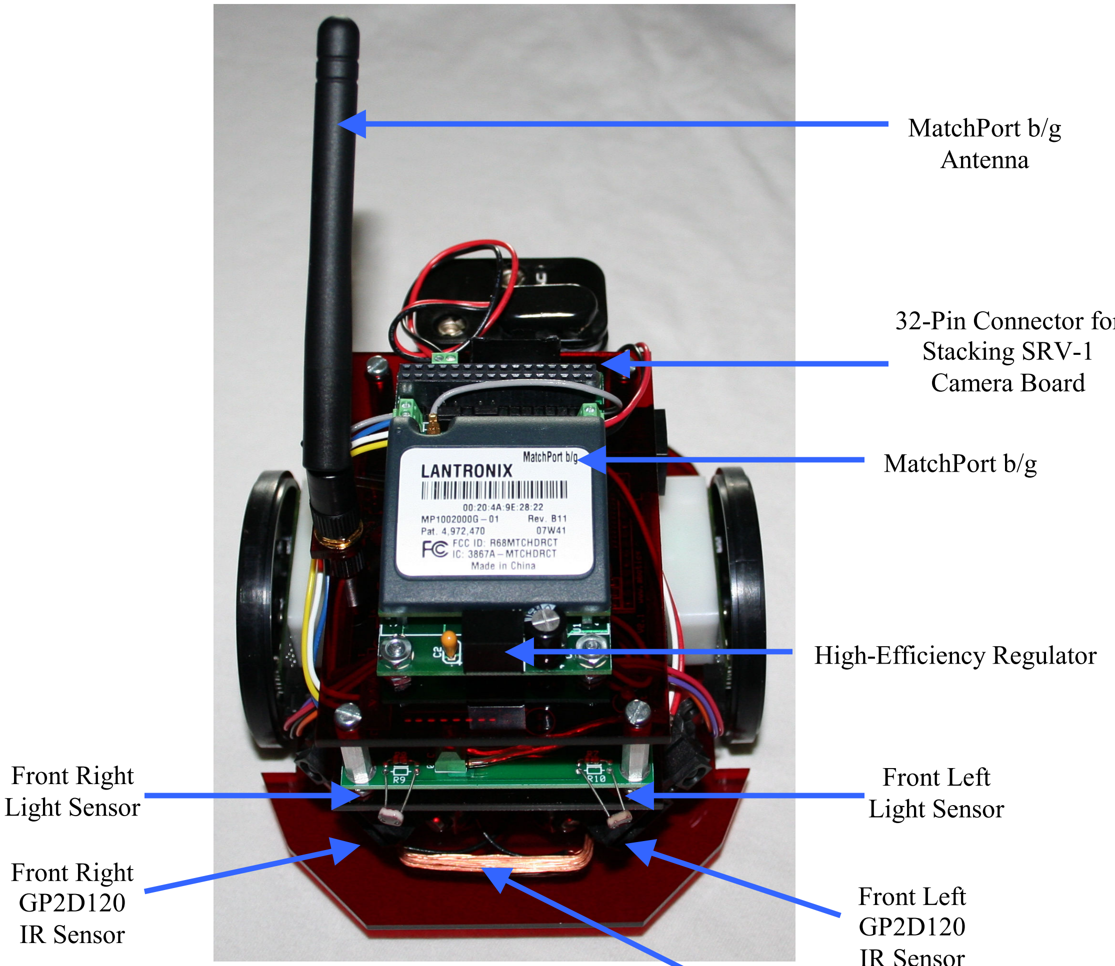

Figure 2. Front view of OPEN-ROBOT shown with MatchPort b/g.

An on-board voltage measurement circuit continuously samples OPEN-ROBOT's battery voltage level. This 8-bit reading can be converted to a voltage reading and used to determine when the batteries need to be replaced or recharged when using rechargeable batteries.


Figure 3. RFID Chip Module, Antennae, and Tags from SonMicro Electronics.

A unique Radio Frequency Identification (RFID) chip-module and antenna have been integrated and can be used to read or write to passive RFID tags (*purchase separately*). A passive RFID tag does not contain a power source and therefore receives its power when a reader's antenna interrogates the tag. These tags have about the same footprint as a credit card and are almost as thin. You can place the RFID tags on the floor of OPEN-ROBOT's environment and when the antenna is over top of a tag you can command OPEN-ROBOT to read the tag contents or write new data to the tag. OPEN-ROBOT also has a built-in function that will continuously sample for an in-range tag and then stop if a tag is discovered. There are a plethora of experiments and learning exercises that can be based upon OPEN-ROBOT's RFID capability.

Figure 4. MatchPort® b/g by LANTRONIX® and XBee® ZigBee™ module by Digi Corp.

Wireless control can be achieved through the use of a MatchPort b/g or XBee module (*purchase separately*). The MatchPort module allows you to control OPEN-ROBOT across a WiFi 802.11b/g connection using TCP or UDP Sockets. This means that your laptop or desktop must have an 802.11b/g wireless card to be able to communicate with OPEN-ROBOT. The XBee module allows you to control your robot wirelessly, but using a simple serial port protocol. You can use our freely available Manual Control program to connect to and drive OPEN-ROBOT or test out its sensors and other features. However, to write your own programs for a MatchPort equipped robot you will need to be proficient at writing TCP or UDP Socket code or you can use our freely available OPEN-ROBOT TCP Functions Class library. This library is designed to be used with either Microsoft's® Visual Basic or C# Express (*freely available*). Using this library does not require knowledge of TCP Socket programming. To write code for an XBee quipped robot you will need to know how to write serial port code or again simply use our free OPEN-ROBOT ZigBee Functions Class library.

If you plan to add-on the SRV-1 Camera board then we suggest using the MatchPort b/g so that you will have the added bandwidth that is needed to transmit captured images back to your controlling desktop or laptop computer. The XBee module is great for large robot swarms since it is low power and much cheaper than the MatchPort b/g. The XBee module is good choice for those who want to keep their robot as low cost as possible or may have legacy code that is using a serial port protocol.

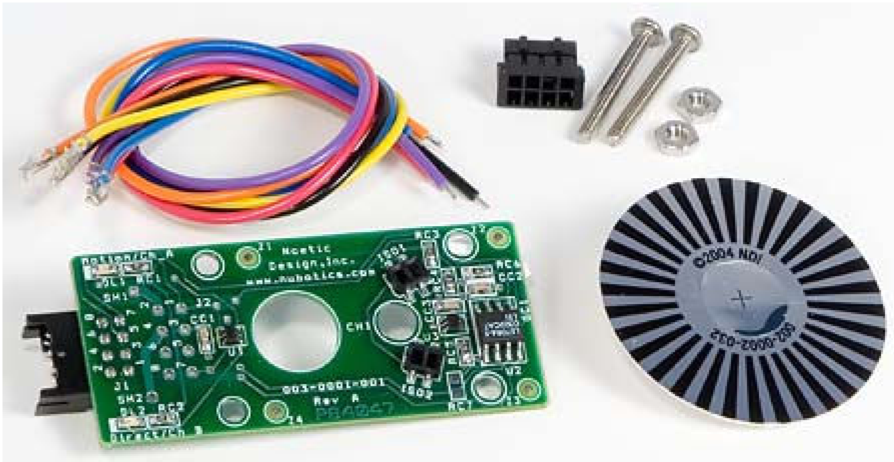

Figure 5. WW02 Wheel Encoders by Nubotics™.

OPEN-ROBOT is designed to work with low-cost WW02 wheel encoders from Nubotics™ (*purchase separately*). Adding the WW02 wheel encoders will provide OPEN-ROBOT with the ability to perform closed loop Proportional-Integral-Derivative (PID) position and velocity control. This means that you can command OPEN-ROBOT to drive with a specified wheel velocity and it will attempt to maintain that velocity by using feedback from the WW02 encoders. You can also command each wheel to drive to a specific position in encoder ticks. For example, you can command the robot to drive forward or backward 8 inches or rotate 90 degrees clockwise or counter-clockwise.

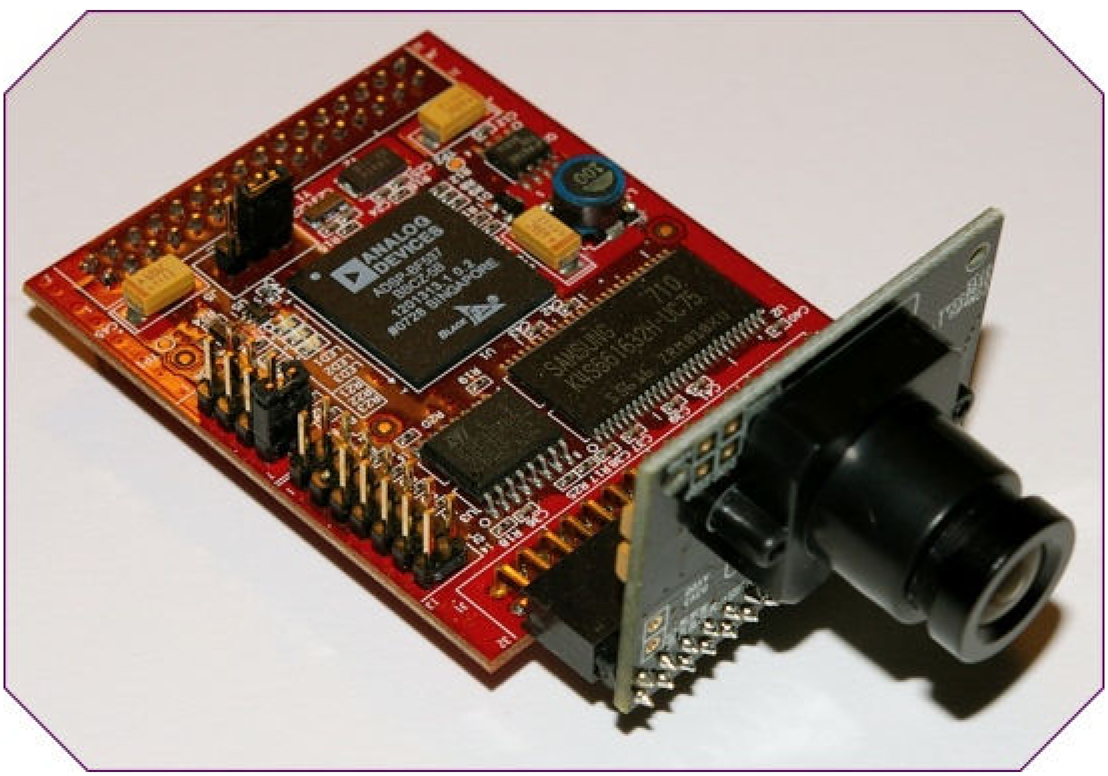

Figure 6. Isometric view of SRV-1 Blackfin Camera Board from Surveyor Corp.

OPEN-ROBOT was designed with forward thinking so that it is compatible with the SRV-1 Blackfin Camera board (*purchase separately*) from Surveyor Corporation. The SRV-1 contains a 1.3 mega-pixel camera attached to a 500MHz Blackfin Digital Signal Processor (DSP). The OPEN-ROBOT's MatchPort/XBee PCB integrates a 32-pin connector that allows for the stacking of the SRV-1 Camera board on top of OPEN-ROBOT. Adding the SRV-1 Camera board opens up a whole new world of possibilities since your robot will be able to perform complex vision processing or simply drive around and while capturing images of its environment.

OPEN-ROBOT's PIC18F4520 Based Controller Board

The "brain" of OPEN-ROBOT is based upon its PIC18F4520 controller board. This board contains all the required hardware to make OPEN-ROBOT move, read attached sensors, and interface with the MatchPort b/g so that you can control it wirelessly.

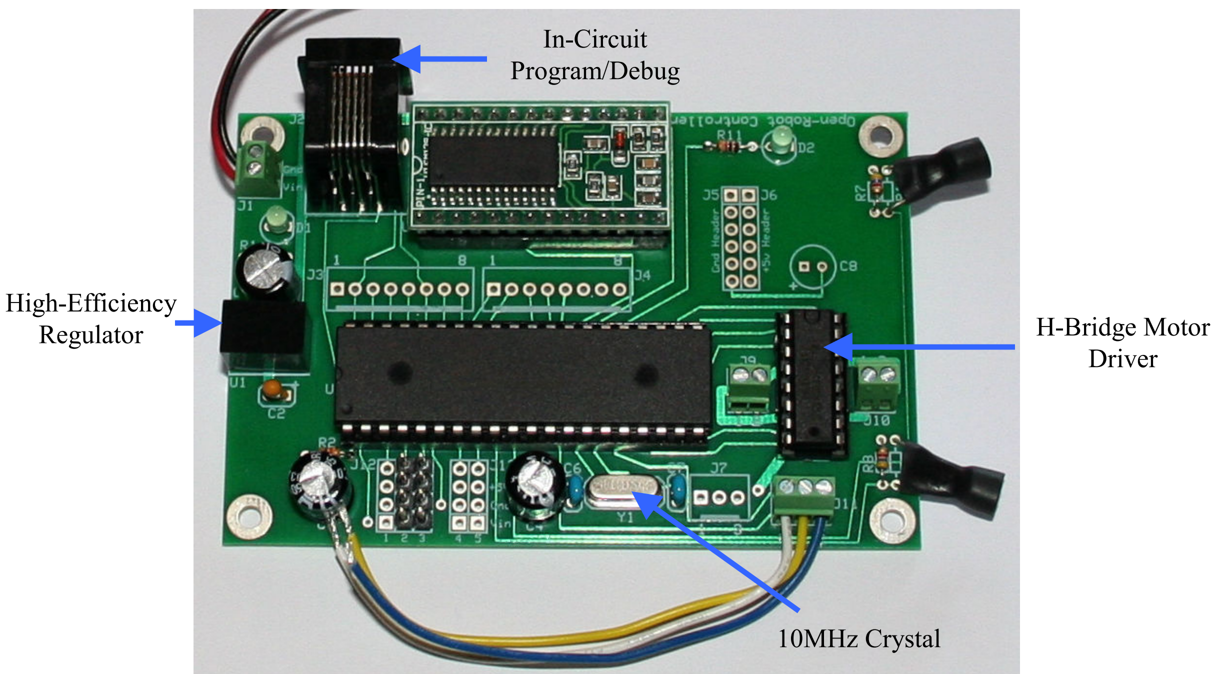

Figure 7. OPEN-ROBOT's PIC18F4520 Based Controller Board.

The PIC18F4520 is clocked at 40 MHz and runs a specially designed firmware that continuously reads the attached GP2D120 infrared sensors, battery voltage, and two frontward facing light sensors and updates special variables so that the latest reading is always available to the end-user. This firmware contains a serial interrupt service routine that is responsible for receiving all incoming serial data from the MatchPort b/g. Once a command has been read in and is understood the PIC18F4520 performs the requested command, sends back data or an acknowledgement and the interrupt routine is exited.

The PIC18F4520 is programmed with a special **boot loader** program. This boot loader program is what allows you to wirelessly upload new or custom firmware. Once a new firmware program is uploaded the PIC18F4520 resets and begins running the new firmware. It must be noted that user created firmware programs must include the appropriate CCS C boot loader files otherwise you will over write the boot loader. If this happens you will need to reprogram the boot loader using a PIC programmer or send it back to us for reprogramming (*you must pay all shipping charges*).

You can choose to either use the original firmware or create your own using a suitable CCS C PIC Compiler. I recommend using the PCWH compiler (~$500) because it has a Windows® IDE along with many other nice features. However, you can just as easily use the much cheaper PCH command-line compiler (~$200). If you are an educator or student you are eligible to receive a significant discount on the price of the PCWH compiler software (~$50). The educational version of the PCWH compiler is only good for the PIC18F4520, so you will not be able to compile code for any other PIC chip. If you are looking to outfit an entire lab, CCS offers additional discounts depending upon the number of licenses you need. If you want to perform in-circuit programming and debugging then I recommend purchasing an in-circuit programmer/debugger (~$75) from CCS.

Flexibility of OPEN-ROBOT

As I mentioned earlier, one of the most important aspects of OPEN-ROBOT is its inherent flexibility across the entire educational spectrum.

Level of Education	Basic Robot	Robot with SRV-1 Camera	Manual Control Using Free Software	Sensor Based Activities	Program at High Level Using C#, VB, Java, etc…	Program and Debug PIC18F4520 Using CCS C Compiler	Program High Level Vision or at the embedded Blackfin DSP level
Elementary	X	X	X	X			
Middle	X	X	X	X	X		
High	X	X	X	X	X	X	
College or University	X	X	X	X	X	X	X
Graduate or Research	X	X			X	X	X

Table 1. OPEN-ROBOT's Flexibility Across Various Levels of Education.

From table#1 above it can be seen that OPEN-ROBOT is flexible across the entire range of the educational spectrum. This table is only a basic guideline since there are always students who are capable of performing at a higher level and can benefit from working with the robot at a higher level.

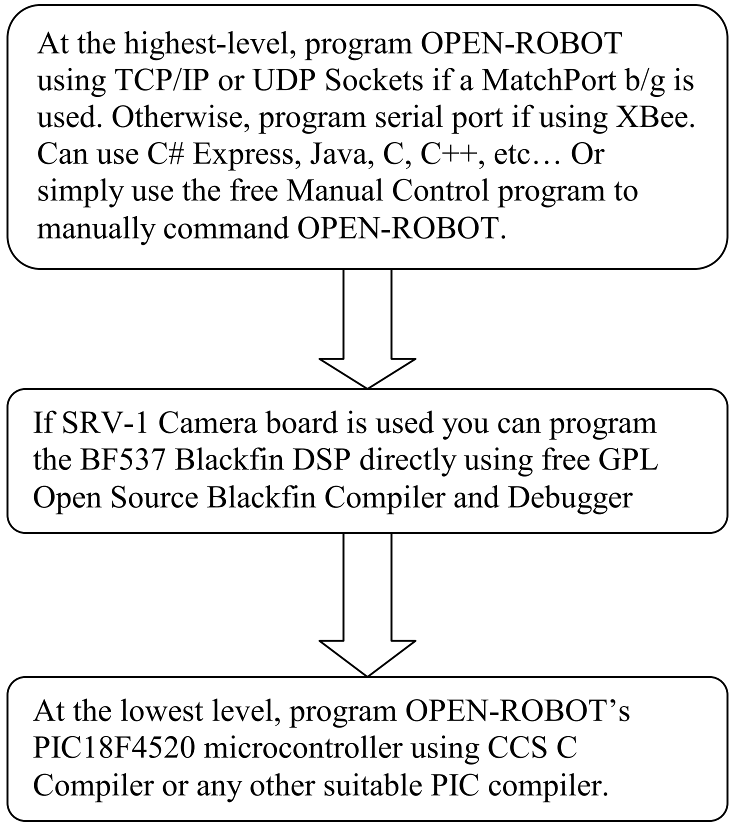

Figure 8. Hierarchical view of OPEN-ROBOT's programmability and control.

OPEN-ROBOT can be used as tool for teaching concepts in mechanics, physics, circuits, programming, or any other relevant area. When concepts in programming are the topics of interest, you can choose to program at the highest level on a desktop, laptop, or handheld computer. However, if you need to cover the area of embedded computing, then you will be able to leverage the PIC18F4520 to investigate assembly or PIC C. When vision concepts are the focus, you will be covered since the OPEN-ROBOT is compatible with the SRV-1 Camera board. With the SRV-1 topics related to Digital Signal Processors (DSPs) can also be explored.

I hope that this has been both informative and inspiring. Whether you decide to purchase an OPEN-ROBOT from my company or build your own using the freely available documentation I wish you the best experience and as always feel free to send your comments or questions to abe@abotics.com. Happy Robot Building!