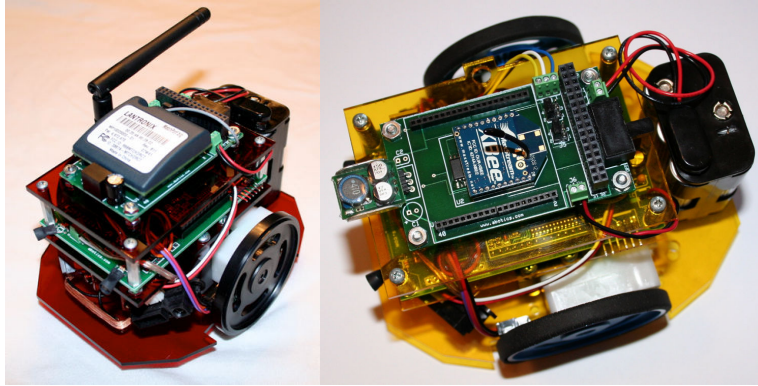


OPEN-ROBOT Owner's Manual



OPEN-ROBOT shown with Matchport b/g (left) and XBee (right)

This owner's manual will guide you through the required assembly steps for your OPEN-ROBOT kit. First you will solder and assemble the controller board. The controller board will be the "Brains" of your robot. Then you will assemble the MatchPort/XBee wireless board and finally configure either you MatchPort b/g or XBee modules. Approximate assembly time is 4 hours, but heavily depends upon your soldering experience. The remainder of this Manual contains information related to programming, add-on sensors, etc...

It is assumed that you have soldering experience and can identify common electronic components, i.e. resistor, capacitor, LED, IC-Socket, header, voltage regulator and common mechanical parts i.e. machine screw, nut, washer. If you are uncomfortable with the above statement then please feel free to return your ***unopened*** OPEN-ROBOT Kit. However, if you are up for the challenge then we hope you enjoy the assembly process. All images are high resolution, so zoom in when you need a closer view.

Contents

| | |
|---|----|
| Chapter 1: Main PIC18F4520 controller board assembly | 5 |
| Review OPEN-ROBOT Controller Board Component List: | 5 |
| Required & Optional Tools:..... | 5 |
| Step#1, Solder 1k Ohm Resistors: | 6 |
| Step#2, Solder 47k Ohm Resistor:..... | 7 |
| Step#3, Solder 4.7k Ohm Resistors: | 8 |
| Step#4, Solder 2.7k Ohm Resistor:..... | 9 |
| Step#5, Solder 0.1µF Tantalum Capacitor: | 10 |
| Step#6, Solder IC-Sockets: | 11 |
| Step#7, Solder GM8 Motor Terminal Blocks:..... | 12 |
| Step#8, Solder Non-Polarized Headers: | 13 |
| Step#9, Solder Remaining Capacitors: | 14 |
| Step#10, Solder Voltage Regulators and Resonator: | 15 |
| Step#11, Solder RFID Jumper Wire:..... | 16 |
| Step#12, Solder Green LED: | 17 |
| Step#13, Solder Cadmium Sulfide Photocells:..... | 18 |
| Step#14, Solder RFID Antennae Header:..... | 19 |
| Step#15, Add Heat Sink Tube To Cadmium Sulfide Photocells:..... | 20 |
| Step#16, Solder Power and PIC18F4520 UART Wires | 21 |
| Step#17, Install Chips: | 22 |
| Chapter 2: Assemble MatchPort/XBee wireless board..... | 23 |
| Step#18, Review OPEN-ROBOT MatchPort/XBee Contents: | 23 |
| Step#19, Prepare the 2mm pitch MatchPort/XBee sockets: | 24 |
| Step#20, Solder 74LVC245 and 74HC245 Octal Buffer: | 25 |
| Step#21, Solder Remaining Components: | 26 |
| Step#22a, Install MatchPort b/g Module (purchase separately):..... | 27 |
| Step#22b, Install XBee ZigBee Module (purchase separately):..... | 27 |
| Chapter 3: How to Test OPEN-ROBOT Circuit Boards | 28 |
| Chapter 4: OPEN-ROBOT body Assembly | 31 |
| Step#23, Assemble OPEN-ROBOT Body:..... | 31 |
| Step#24, Assemble and Attach Rear Caster Wheel:..... | 32 |
| Step#25, Attach GM8 Motor Mounts: | 33 |
| Step#26, Prepare to Attach RFID Antenna (purchase separately): | 34 |
| Step#27, Attach RFID Antenna to Base Plate: | 35 |
| Step#28, Prepare to Solder Noise Filtering Capacitors (Optional):..... | 36 |
| Step#29, Solder Noise Filtering Capacitors:..... | 37 |
| Step#30, Attach GM8 Motors:..... | 38 |
| Step#31, Prepare RFID Antenna Cable (purchase separately):..... | 39 |
| Step#32, Attach RFID Antenna Cable & Prepare to Mount Controller Deck:..... | 40 |
| Step#33, Prepare Controller Deck: | 41 |
| Step#34, Attach Controller Board Standoffs: | 42 |
| Step#35, Mount Controller Deck:..... | 43 |
| Step#36, Mount Controller Board & Connect RFID Cable:..... | 44 |

| | |
|--|-----|
| Step#37, Attach MatchPort/XBee Deck Standoffs: | 45 |
| Step#38, Attach GM8 Wheels: | 46 |
| Step#39, Mount GP2D120 Sensors to Brackets: | 47 |
| Step#40, Mount GP2D120 Sensors to Robot: | 48 |
| Step#41, Connect GP2D120 Sensors to the Controller Board: | 49 |
| Step#42, Attach Dual Lock Tape to Base and Battery Holder: | 50 |
| Step#43, Prepare MatchPort/XBee PCB Deck: | 51 |
| Step#44, Mount the MatchPort/XBee PCB to the Deck: | 52 |
| Step#45, Mount MatchPort/XBee Deck to Robot: | 53 |
| Step#46, Connect Power Wires: | 54 |
| Step#47, Connect PIC18F4520 Serial Port Wires: | 55 |
| Step#48, Attach MatchPort b/g Antenna (purchase separately): | 56 |
| Chapter 5: Get Started with a MatchPort Equipped Robot..... | 57 |
| Getting to Know Your OPEN-ROBOT & MatchPort b/g: | 57 |
| Connecting to OPEN-ROBOT for the First Time: | 59 |
| Connect Using the OPEN-ROBOT TCP-Socket Manual Control Program:..... | 60 |
| (OPTIONAL) - Reconfigure MatchPort Network Settings: | 62 |
| Chapter 6: Get Started with XBee® Equipped Robot | 69 |
| Getting to Know Your OPEN-ROBOT & XBee modules: | 69 |
| Configure XBee Modules: | 70 |
| Connect Using the OPEN-ROBOT XBee/ZigBee Manual Control Program: | 74 |
| Chapter 7: Wirelessly upload PIC18F4520 code..... | 77 |
| Connect a 1k Ohm Resistor Across Pin#1 & Pin#6 on J2 Programming Header: | 78 |
| Connect to Robot Using OPEN-ROBOT TCP-Socket Manual Control Program: | 79 |
| Chapter 8: How to modify PIC18F4520 firmware | 81 |
| Requirements: | 81 |
| Getting Started with the PCWH Compiler: | 81 |
| Architecture of PIC18F4520 Firmware: | 84 |
| Chapter 9: Details of PIC18F4520 controller board..... | 85 |
| Introduction to the OPEN-ROBOT Controller Board | 85 |
| Board Connectors..... | 86 |
| Board Power & Programming Header, J1 & J2..... | 87 |
| Analog to Digital (A/D) Headers, J14 & J15..... | 87 |
| I ² C Header, J7 | 88 |
| GM-8 Motor Headers, J9 & J10 | 88 |
| Extra I/O & WW-02 Wheel Encoders Header, J3 & J4 | 89 |
| PIC18F4520 Serial Headers, J12 | 90 |
| Extended PIC18F4520 Serial Header, J11 (Optional)..... | 90 |
| RFID Module Jumper Header, J13 (Not Required)..... | 90 |
| RFID Antenna Header, J8..... | 91 |
| How to Connect the CCS ICD to the Programming Header | 92 |
| OPEN-ROBOT Serial-Based Firmware Command Set | 93 |
| Serial-Based command set for OPEN-ROBOT..... | 94 |
| Chapter 10: Details of analog to digital (A/D) sensors..... | 100 |
| OPEN-ROBOT's Battery Voltage Measuring Circuit:..... | 100 |
| OPEN-ROBOT's GP2D120 IR Sensors:..... | 101 |

| | |
|---|-----|
| OPEN-ROBOT's Light Sensors: | 103 |
| Chapter 11: Design of PIC18F4520 controller board | 105 |
| List of board requirements | 105 |
| ExpressPCB board services | 106 |
| Chapter 12: Design of MatchPort/XBee wireless board | 107 |
| List of board requirements | 107 |
| ExpressPCB board services | 108 |
| Chapter 13: Mechanical design of OPEN-ROBOT | 109 |
| 3D Modeling software | 109 |
| Chapter 14: Adding the SRV-1 Blackfin Camera | 110 |
| Preparation for the SRV-1 Camera Board | 110 |
| Chapter 15: Upload Modified SRV-1 Firmware | 111 |
| Prerequisites | 111 |
| Uploading Firmware Using ShamCom Terminal and XModem | 111 |
| Chapter 16: Upload srv1.ldr Using Flashload & ICEbear | 116 |
| Prerequisites | 116 |
| Flashload srv1.ldr File Using ICEbear JTAG | 116 |
| Upload srv1.ldr File Using XModem | 118 |
| Appendix | 120 |
| Main PIC18F4520 board schematic | 120 |
| MatchPort/XBee wireless board schematic | 121 |
| PIC18F4520 I/O Map | 122 |
| OPEN-ROBOT Parts List | 123 |
| OPEN-ROBOT Controller Board Parts List | 124 |
| MatchPort/XBee/SRV-1 Board Parts List | 125 |

Chapter 1: Main PIC18F4520 controller board assembly

Review OPEN-ROBOT Controller Board Component List:

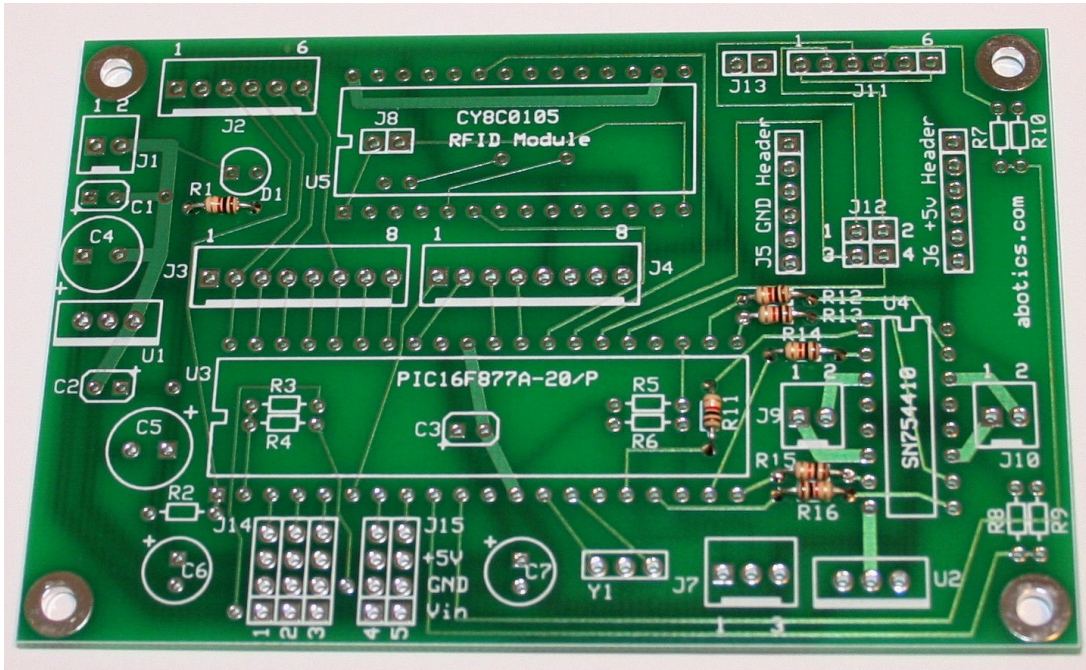
QTY Description

- (7) 1k Ohm Resistor - R1, R11, R12, R13, R14, R15, and R16
- (1) 47k Ohm Resistor – R2
- (5) 4.7k Ohm Resistor – R3, R5, R6, R7 and R8
- (1) 2.7k Ohm Resistor – R4
- (2) Cadmium Sulfide Photocell – R9 and R10
- (3) 0.1 μ F Tantalum Capacitor – C1, C2, and C3
- (2) 33 μ F Capacitor – C6 and C7
- (2) 100 μ F Capacitor – C4 and C5
- (1) 40-Pin IC Socket – U3
- (1) 16-Pin IC Socket – U4
- (1) 28-Pin IC Socket – U5
- (5) 4-Position Non Polarized Header – J14 and J15
- (1) 2-Position Non Polarized Header – J8
- (2) 2-Position 0.1” pitch Terminal Block – J9 and J10
- (1) 6-Position 0.1” right angle non-polarized header – J2
- (1) 20Mhz Resonator – Y1
- (1) Green LED – D1
- (2) +5 Volt 1.5 Amp Voltage Regulator – U1 and U2
- (1) PIC18F4520 Microcontroller – U3
- (1) SN754410 H-Bridge – U4
- (1) 1” Length 3/16” Diameter Heat Shrink Tube

Required & Optional Tools:

- Soldering Iron & Solder (**Required**)
- Wire Snips or Side Cutters and Needle Nose Pliers (**Required**)
- Wire Stripper (**Required**)
- Phillips and flat head screwdriver (**Required**)
- Metal hand file, nail file, or sand paper (**Required when using MatchPort b/g**)
- Multi-Meter (Test assembled board & measure resistors) (**Optional**)

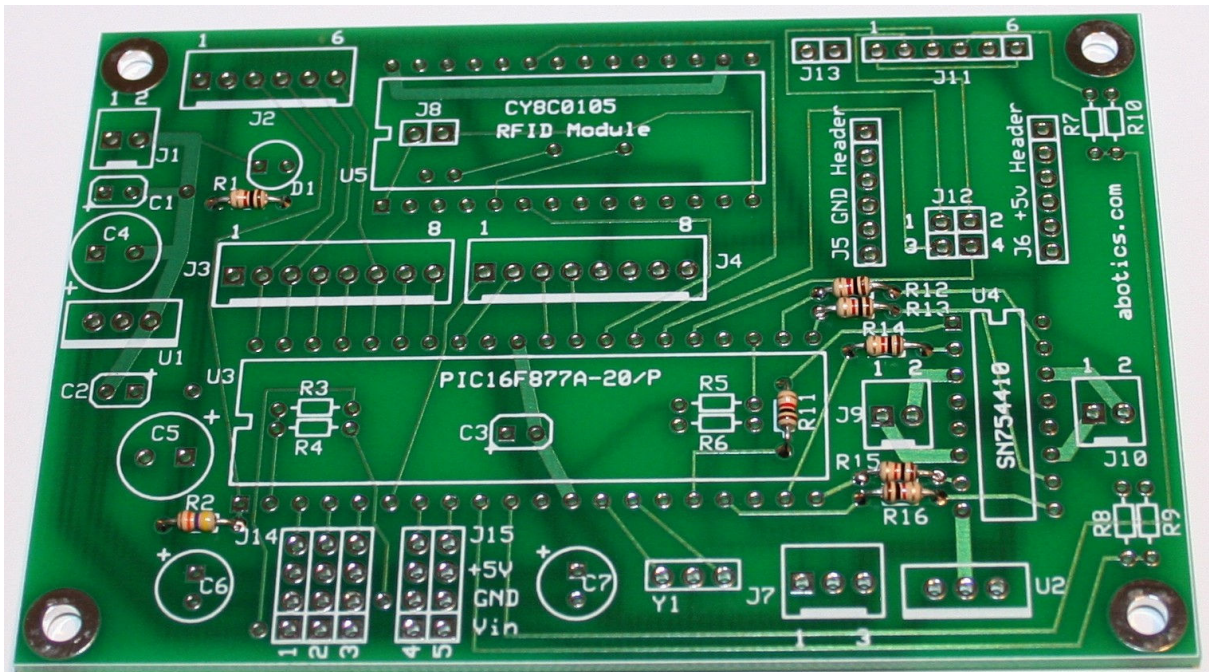
Step#1, Solder 1k Ohm Resistors:



Figure#1, Solder 1k Ohm Resistors.

In step#1 solder (7) 1k Ohm resistors as shown in figure#1. The **brown-black-red** color band identifies a 1k Ohm resistor. Solder the (7) resistors in the following locations on the board: R1, R11, R12, R13, R14, R15, and R16. After soldering the resistors be sure to snip off the excess wire as close to the board as possible without damaging the board or solder joint (save a piece of wire for later use as a jumper). R1 is used as a current limiting resistor for the board's green led. R11, R12, R13, R14, R15, and R16 are used as current limiting resistors for the SN754410 h-bridge chip to prevent damage to the PIC18F4520.

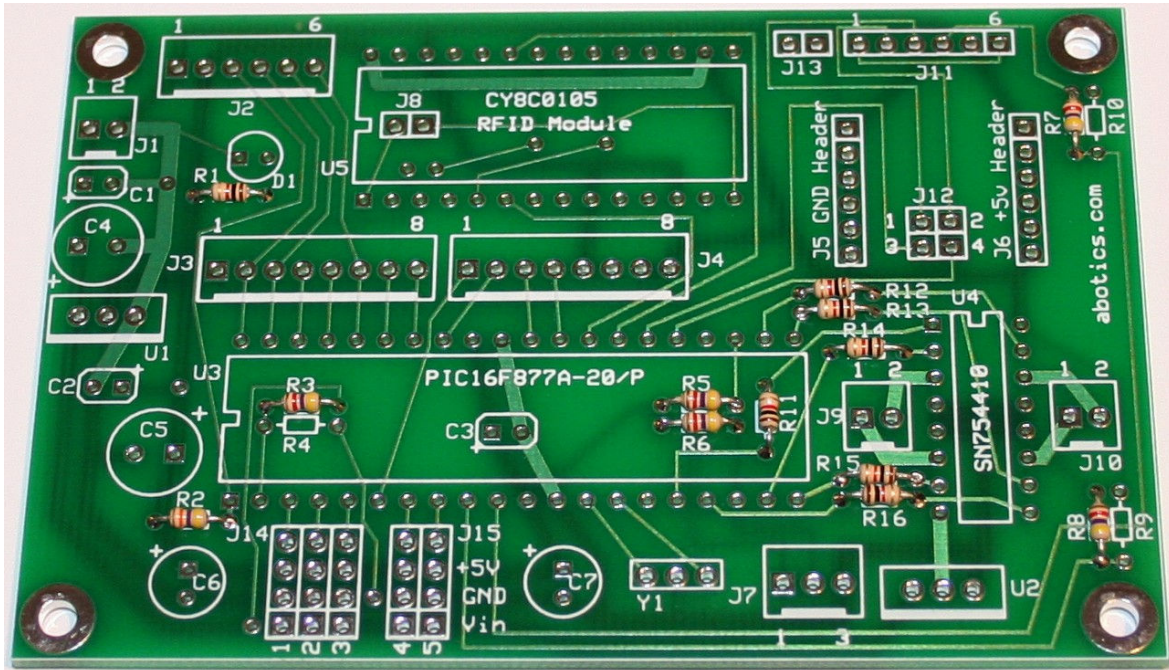
Step#2, Solder 47k Ohm Resistor:



Figure#2, Solder 47k Ohm Resistor.

In step#2 solder (1) 47k Ohm resistor in location R2 on the board. The **yellow-violet-orange** color band identifies a 47k Ohm resistor. R2 is used to pull the PIC18F4520's MCLR pin high, but also allows for in-circuit programming (ICP).

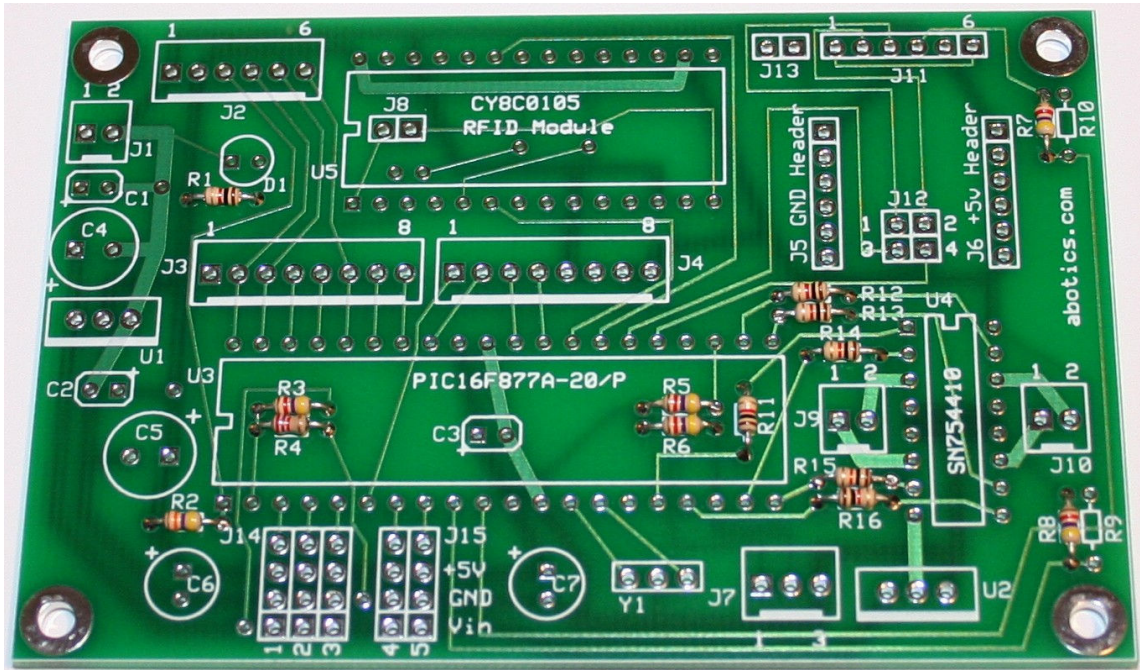
Step#3, Solder 4.7k Ohm Resistors:



Figure#3, Solder 4.7k Ohm Resistors.

In step#3 solder (5) 4.7k Ohm resistors in the following locations: R3, R5, R6, R7, and R8. The **yellow-violet-red** color band identifies a 4.7k Ohm resistor. R3 is used as part of a battery voltage monitoring circuit. R5 and R6 are used as pull-up resistors for the PIC18F4520's I²C pins. R7 and R8 are used in a voltage divider so that the output from the light sensing cadmium sulfide photocells can be measured.

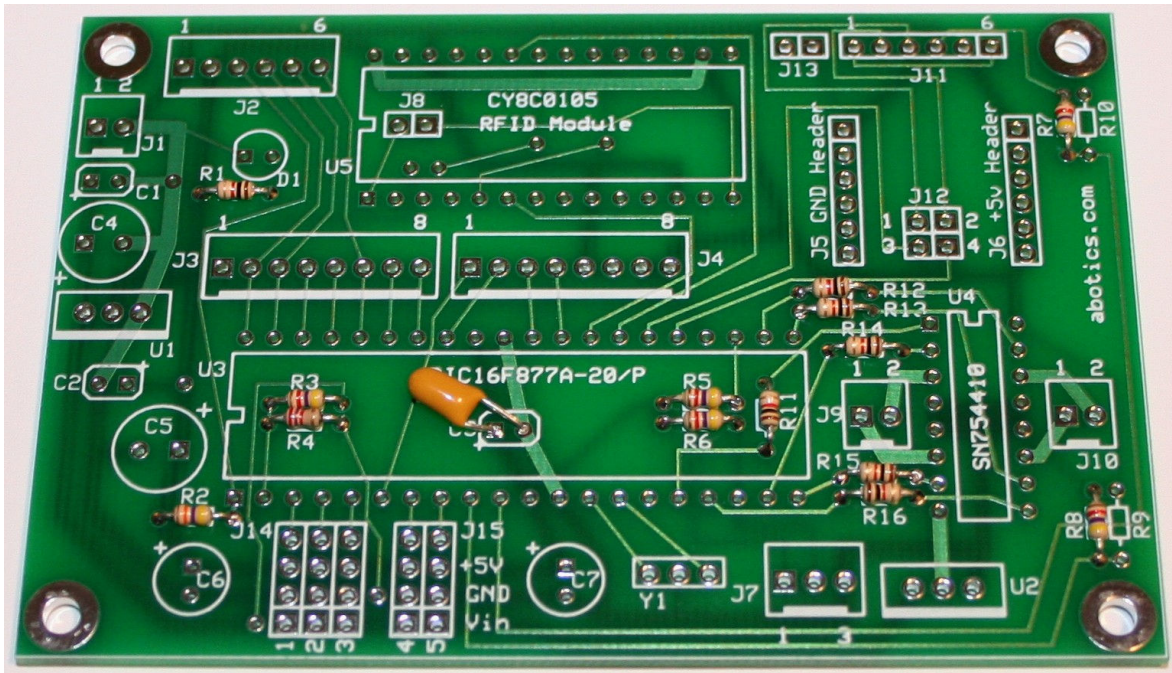
Step#4, Solder 2.7k Ohm Resistor:



Figure#4, Solder 2.7k Ohm Resistor.

In step#4 solder (1) 2.7k Ohm resistor in the location R4. The **red-violet-red** color band identifies a 2.7k Ohm resistor. R4 is used in conjunction with R3 as part of a battery voltage monitoring circuit.

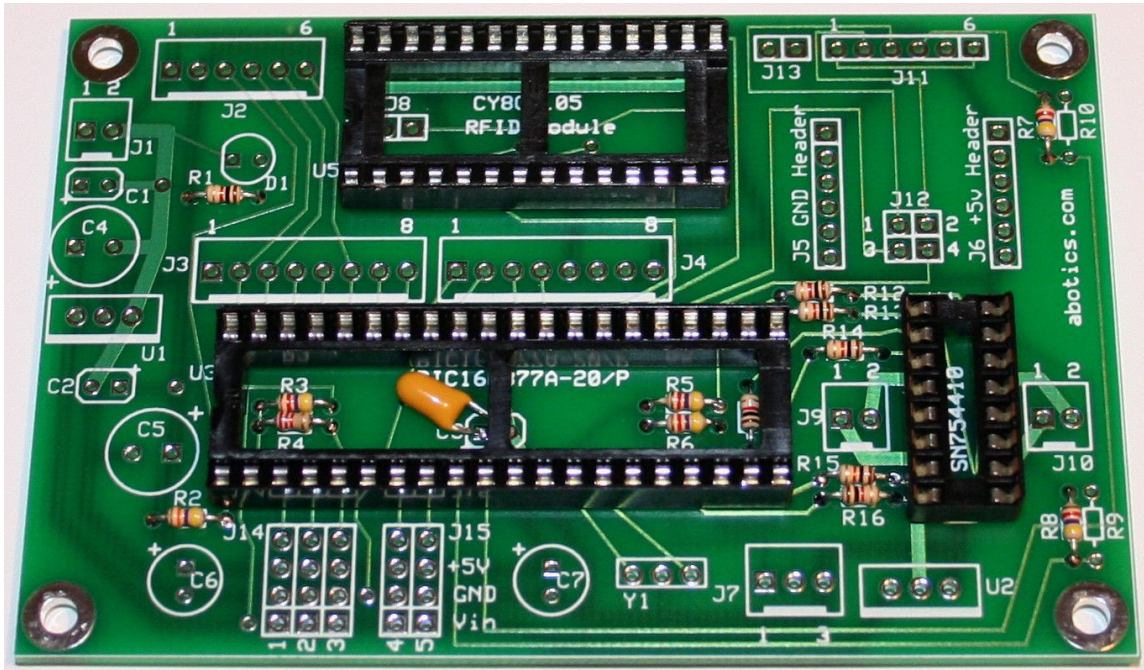
Step#5, Solder 0.1 μ F Tantalum Capacitor:



Figure#5, Solder 0.1 μ F Capacitor.

In step#5 solder (1) 0.1 μ F tantalum capacitor in location C3. Be sure to bend the capacitor as shown in figure#5 otherwise it will prevent you from soldering the PIC18F4520's 40-pin IC-Socket. Before soldering C3 you might want to test fit the 40-pin IC-Socket. C3 is used as a noise filtering capacitor for the PIC18F4520.

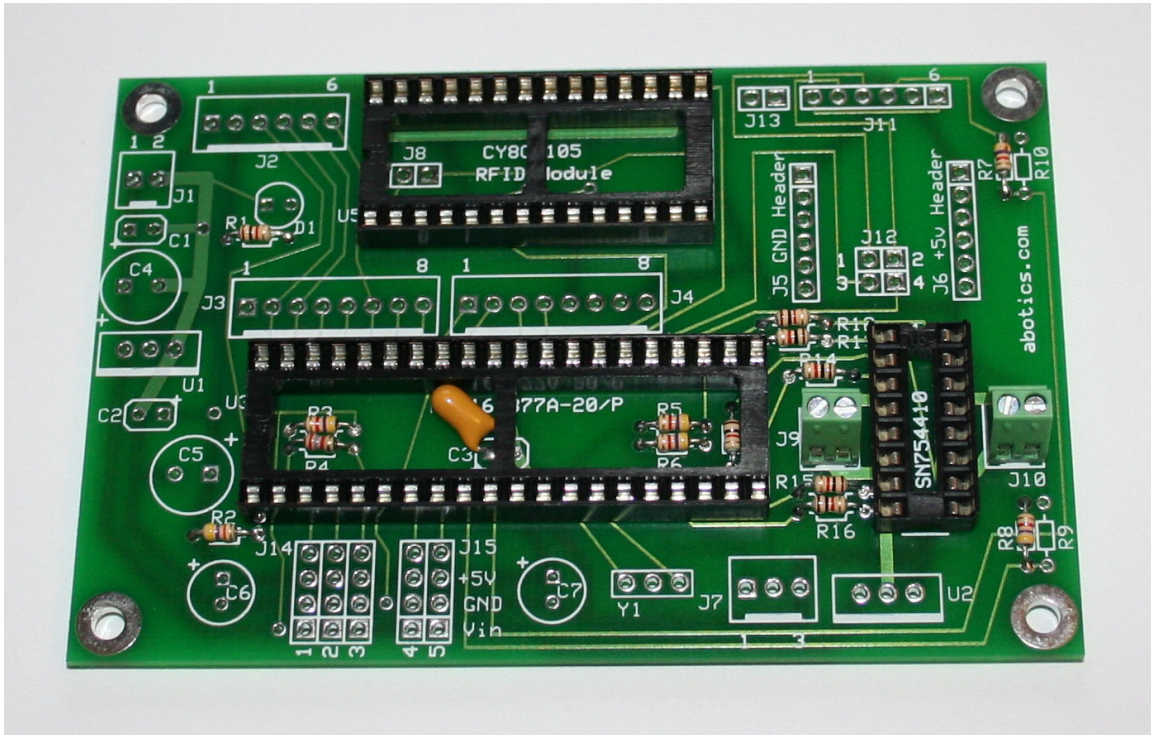
Step#6, Solder IC-Sockets:



Figure#6, Solder IC-Sockets.

In step#6 solder (3) IC-Sockets in the following locations: U3, U4, and U5. U3 receives the 40-pin socket, U4 receives the 16-pin and U5 receives the 28-pin. The 40-pin socket will house the PIC18F4520, the 28-Pin socket will house the CY8C0105 RFID Chip Module and the 16-pin socket will hold the SN754410 H-bridge.

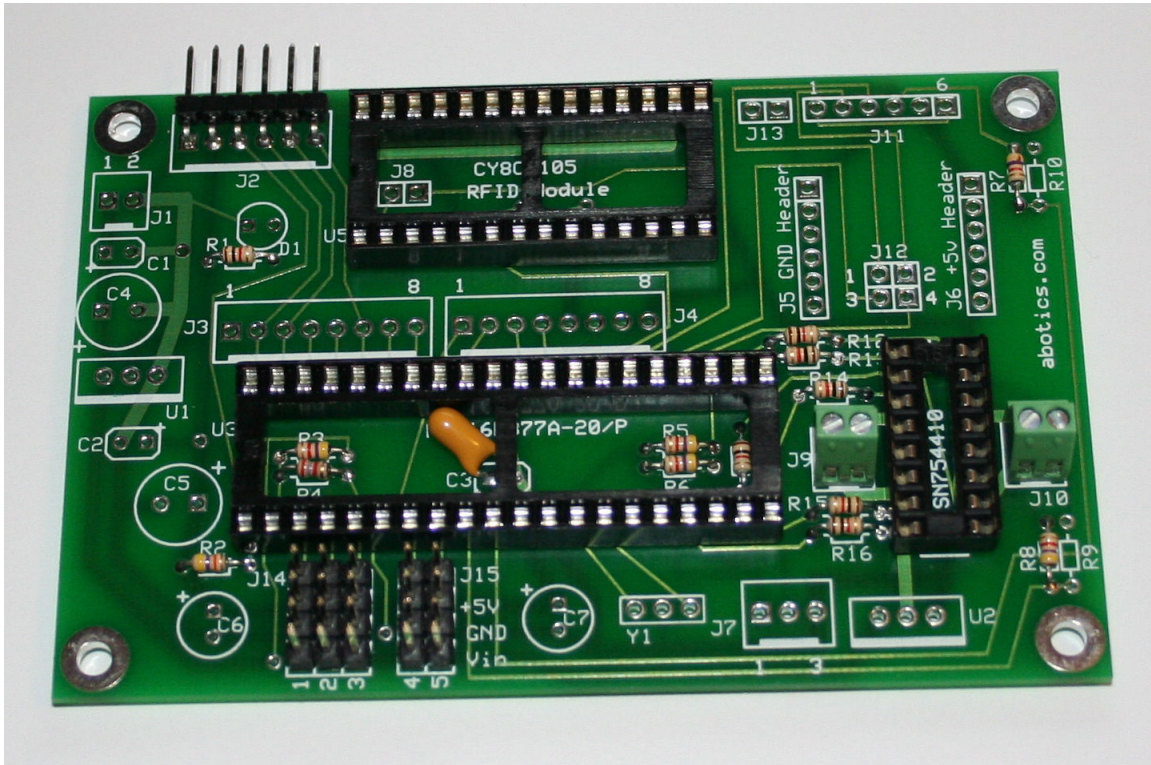
Step#7, Solder GM8 Motor Terminal Blocks:



Figure#7, Solder GM8 Terminal Blocks.

In step#7 solder (2) terminal blocks at locations J9 & J10. These terminal blocks provide a connection point for the GM8 gear motors. There are (2)-small pegs on the bottom of each terminal block. You will need to snip or cut these off so that the terminal block mounts flush with the PCB.

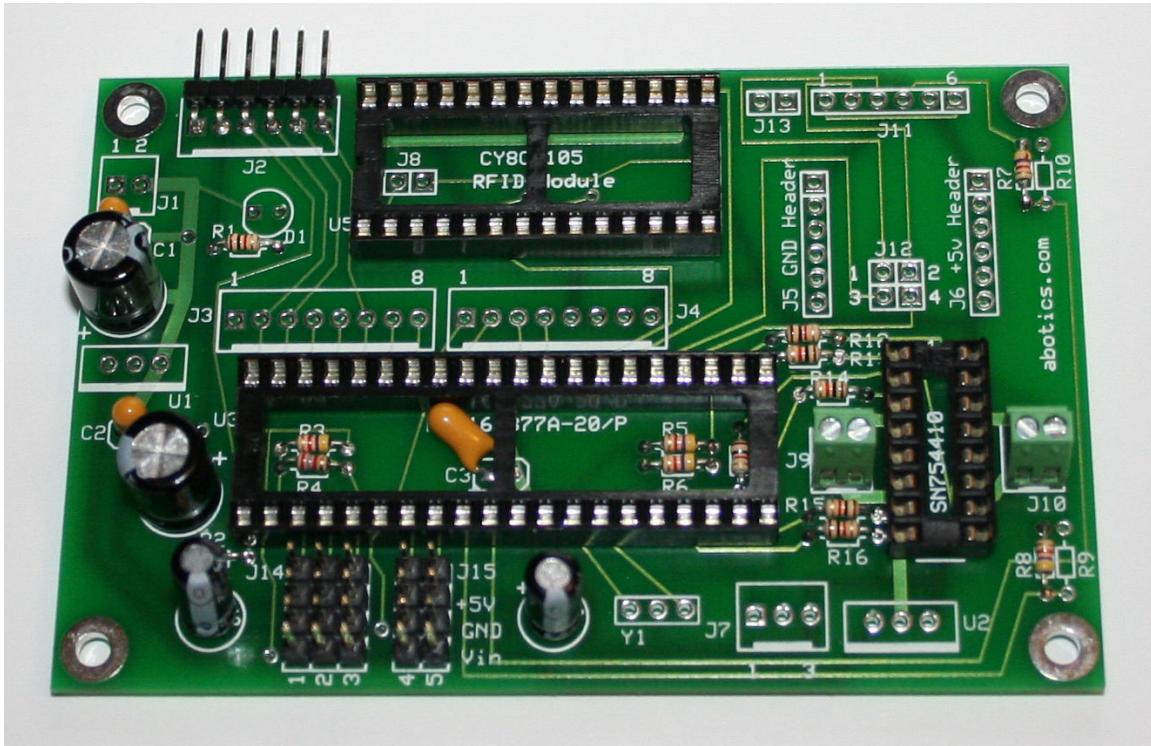
Step#8, Solder Non-Polarized Headers:



Figure#8, Solder Non-Polarized Headers.

In step#8 solder (5) 4-position non-polarized headers in the following locations of J14: 1, 2, 3 and J15: 4 & 5. These headers allow for the connection of devices that require an analog to digital (A/D) conversion, for example, the Sharp GP2D12 or GP2D120 Infrared Range Sensor. Solder the 6-position 90-degree non-polarized header in location J2. The J2 header provides In-Circuit Programming/Debugging access to the PIC18F4520

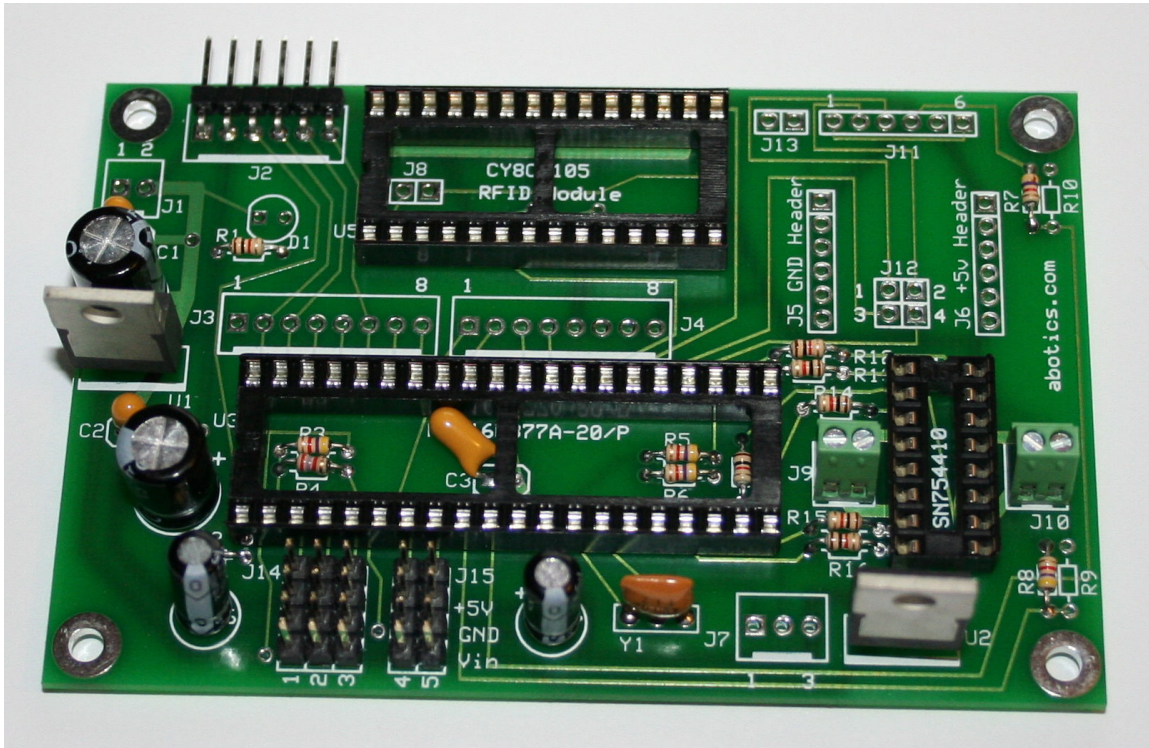
Step#9, Solder Remaining Capacitors:



Figure#9, Solder Remaining Capacitors.

In step#9 solder (2) 0.1 μ F tantalum capacitors in the following locations: C1 and C2. Then solder (2) 100 μ F capacitors in the following locations: C4 and C5 and finally solder (2) 33 μ F capacitors in the following locations: C6 and C7. C6 and C7 help to filter out any noise generated from the connection of Sharp GP2D12 or GP2D120 IR Sensors to the J14 & J15 headers.

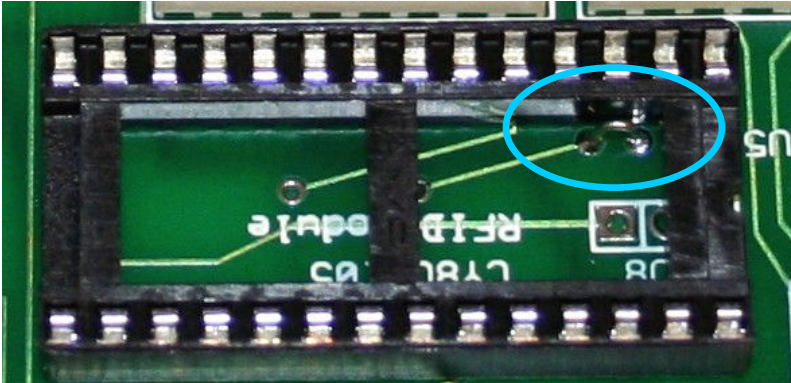
Step#10, Solder Voltage Regulators and Resonator:



Figure#10, Solder Voltage Regulators and 20Mhz Ceramic Resonator.

In step#10 solder (2) +5volt voltage regulators in the following locations: U1 and U2. Be sure to orient the regulators properly. Additionally, solder (1) 20Mhz ceramic resonator in location Y1. The three-pin 20Mhz resonator is not polarized and can therefore be soldered in either of the two possible orientations. The U1 regulator supplies +5 volts to all the logic level devices and the U2 regulator supplies +5 volts to the GM-8 motors. The 20 Mhz resonator clocks the PIC18F4520.

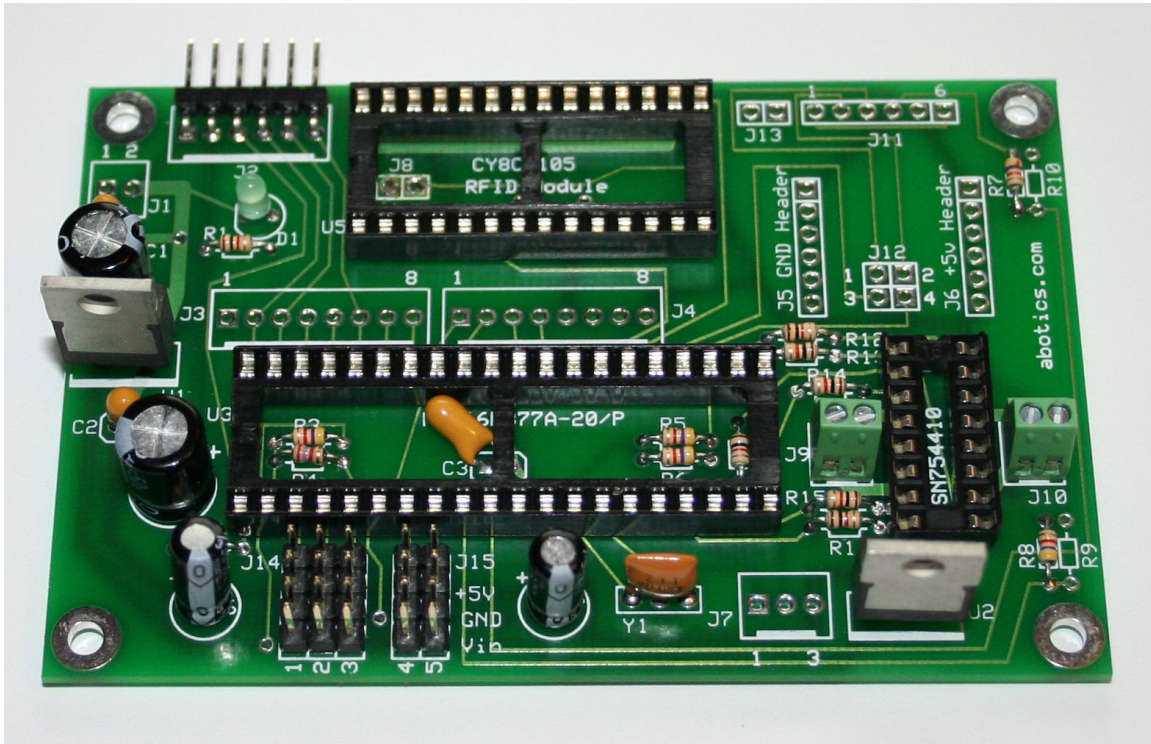
Step#11, Solder RFID Jumper Wire:



Figure#11, Solder RFID Jumper Wire.

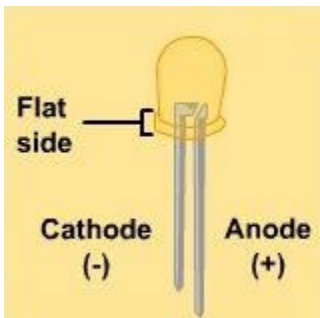
In step#11 solder the RFID jumper wire. From figure#11 you can see that the jumper must be soldered just above J8. You can use a piece of wire that was clipped from one of the resistors soldered earlier, but bend the wire into a small U-shape.

Step#12, Solder Green LED:



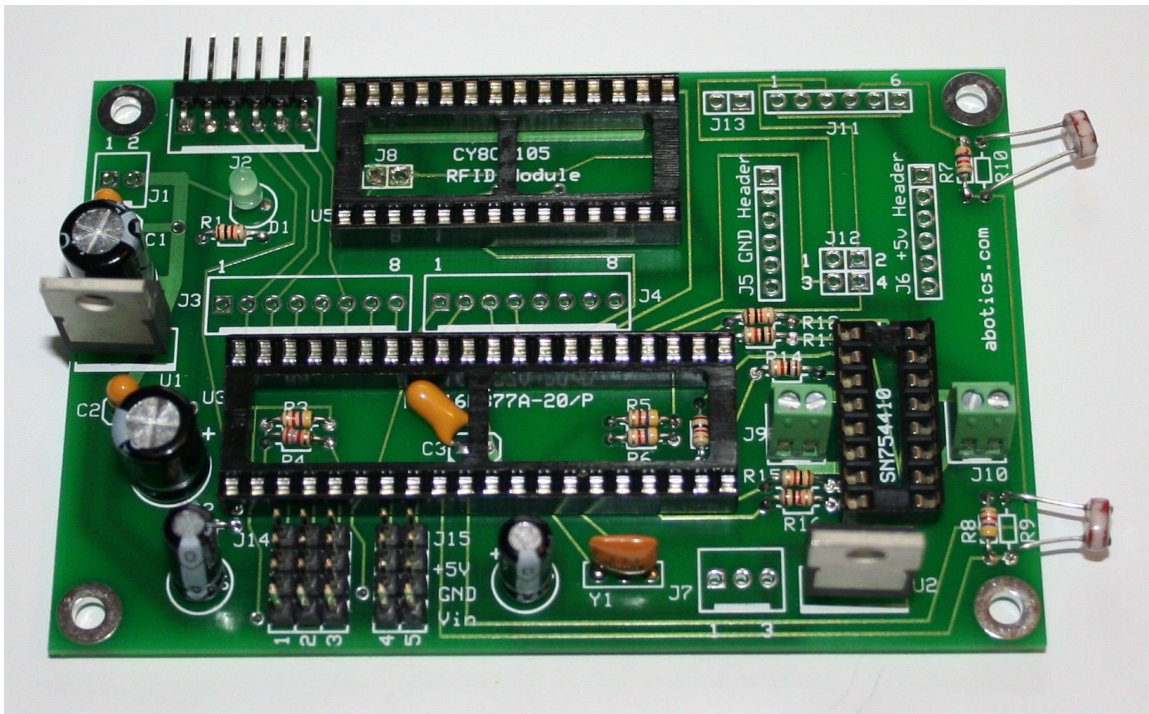
Figure#12a, Solder Green LED.

In step#12 solder the green LED in location D1. You should notice that the D1 outline on the board has one side with a flat spot; this flat spot corresponds to the negative or cathode terminal of the LED. The round side corresponds to the LED's anode or positive terminal. Typically the LED would have a flat spot on the bulb, but since we are using a small LED only the positive terminal can be identified as the longer one and must correspond to the round side of the D1 outline. Please refer to the LED illustration provided below in figure#12b.



Figure#12b, LED Illustration.

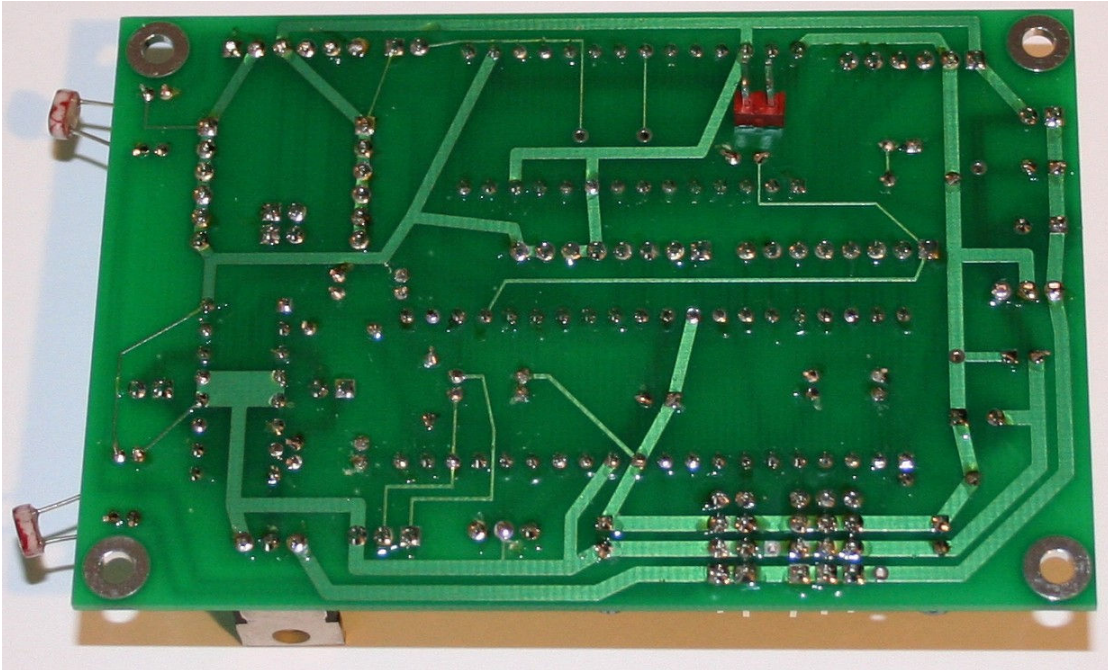
Step#13, Solder Cadmium Sulfide Photocells:



Figure#13, Solder Cadmium Sulfide Photocells.

In step#13 solder the cadmium sulfide photocells in locations: R9 and R10. You can bend these as you see fit, but I have bent mine to look left and right so that the direction of incident light can be determined.

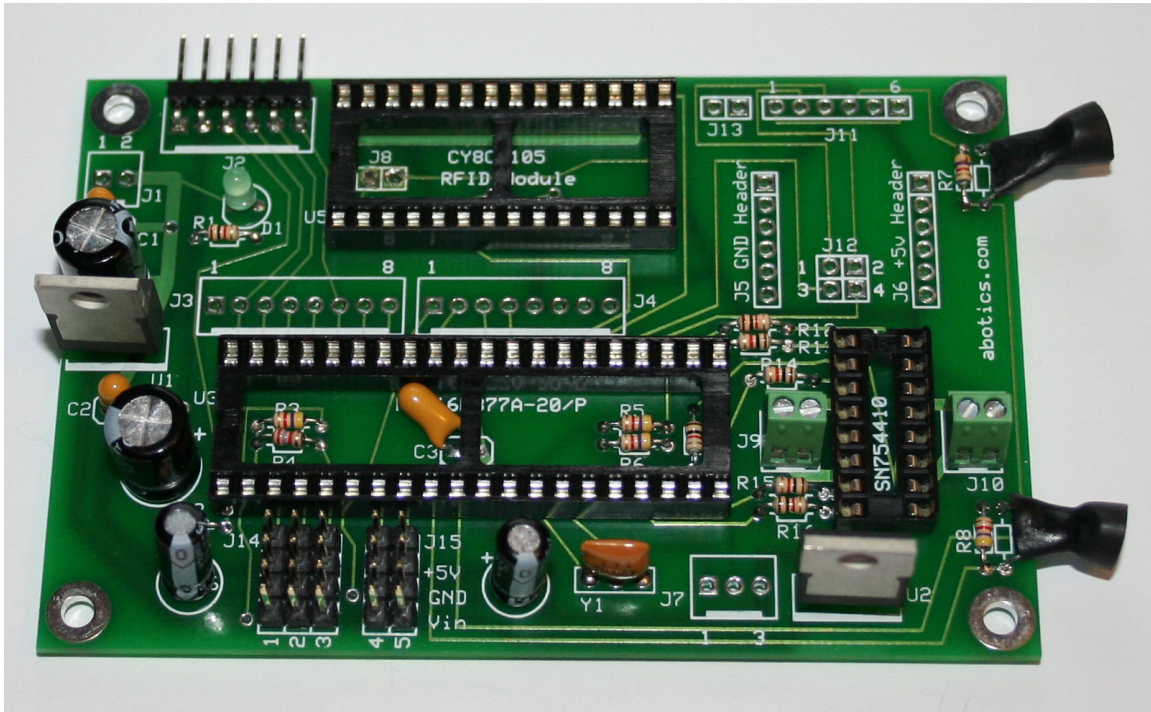
Step#14, Solder RFID Antennae Header:



Figure#14, Solder RFID Antennae Header.

In step#14 solder the 2-position non-polarized RFID antennae header in location J8. Figure#14 shows the bottom side of the board because the J8 RFID antenna header must be soldered on the bottom side, since the RFID chip module is to be installed on the top side of the board and would prevent you from connecting the antenna.

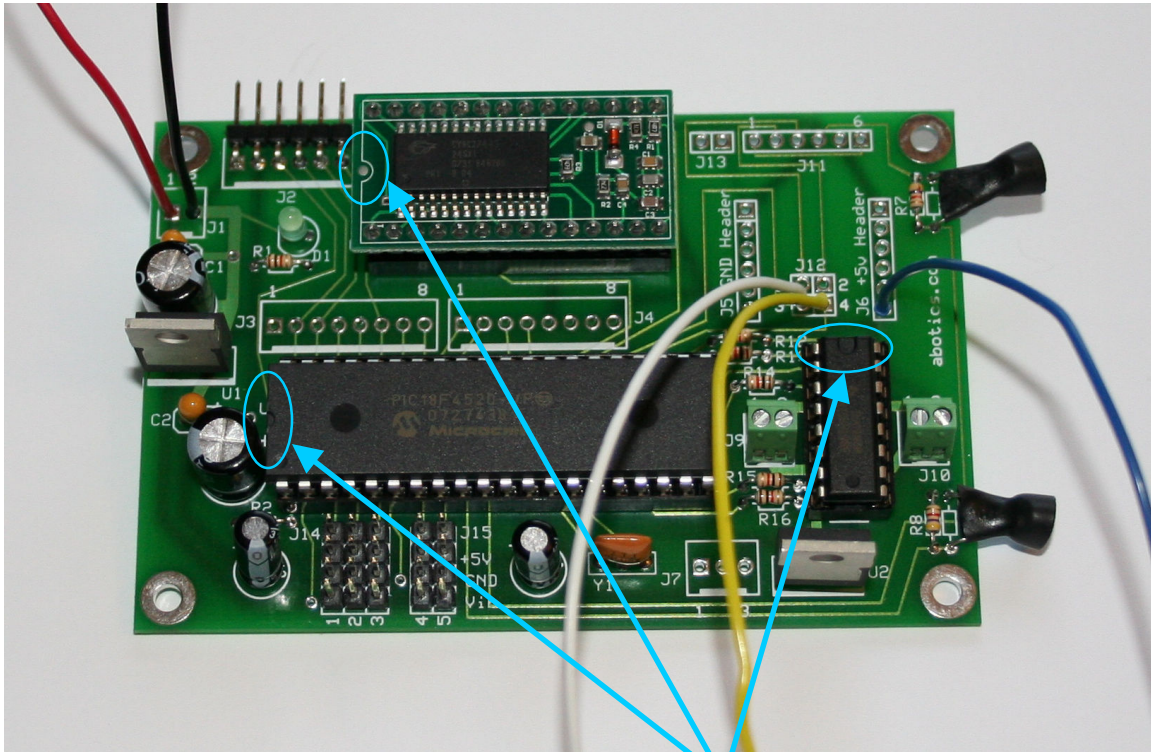
Step#15, Add Heat Sink Tube To Cadmium Sulfide Photocells:



Figure#15, Add Heat Shrink Tube to Cadmium Sulfide Photocells.

In step#15 add heat shrink tube to both of the cadmium sulfide photocells, so that they can be used to detect the direction of incoming light. If you decide to leave the heat shrink tube off the photocells will be less likely to detect the direction of say an incident flashlight beam or other light source.

Step#17, Install Chips:



Figure#17, Install Chips.

*Please note the chip orientations. The front end of a chip is typically denoted by a small U-shaped depression and pin#1 is usually located to the left of this mark.

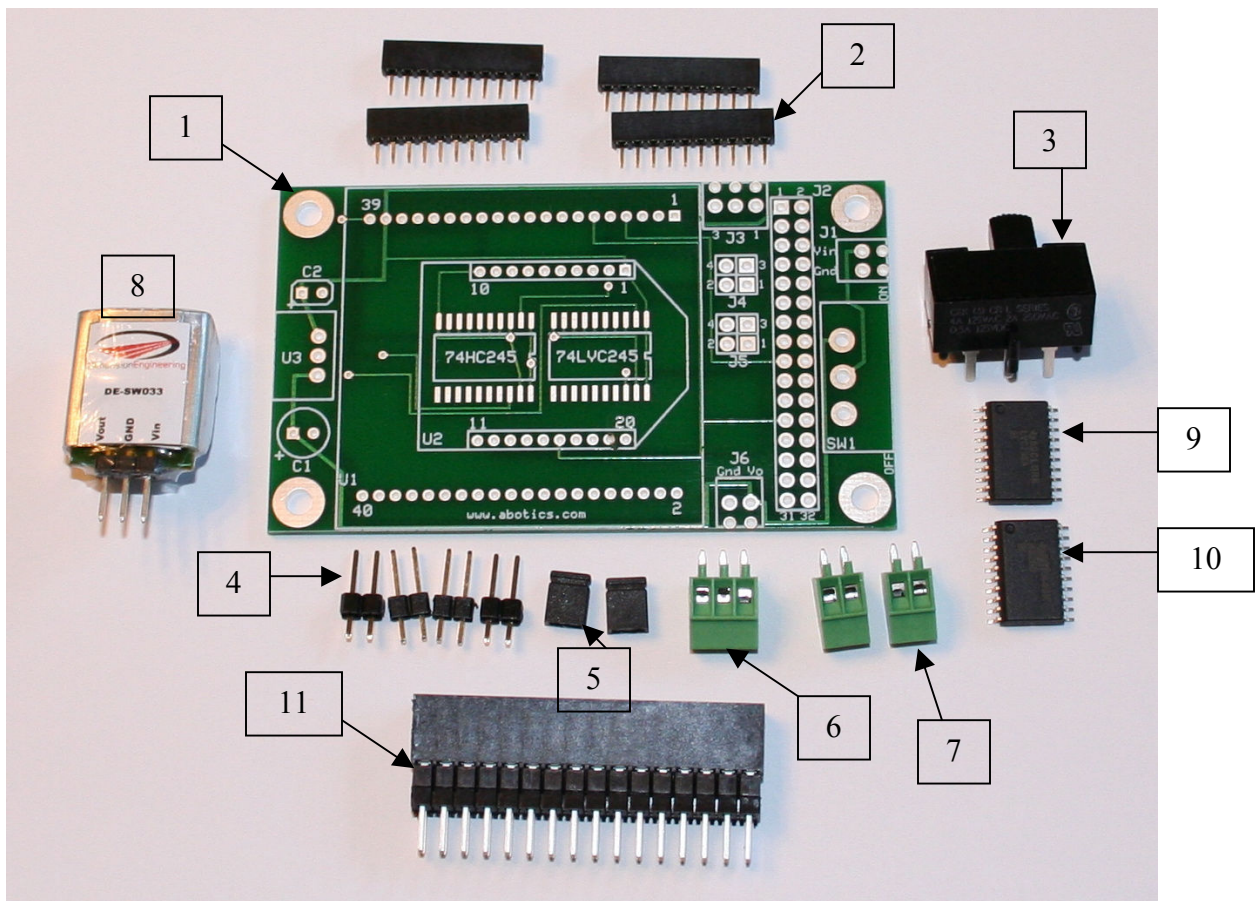
In step#17 install the PIC18F4520, SN754410 and RFID Chip Module and (3) jumpers. Install the PIC18F4520 chip in the 40-pin socket (U3), the SN754410 chip in the 16-pin socket (U4) and the RFID chip in the 28-pin socket (U5). You might have to bend the pins on some of the chips to get them to insert into the respective socket, but be very careful not to over bend pins as they can fracture and break off yielding the chip useless. You may want to take a look at the [How to Test OPEN-ROBOT Circuit Boards](#) chapter before installing the chips.

****NOTE: You should clean the solder flux from your board prior to installing the chips. If you used a water-soluble flux then you can simply clean with water and scrub with an old toothbrush. Otherwise, if you used alcohol-soluble flux you may need to clean the flux with isopropyl alcohol. If you used a non-clean flux then you are already done.***

Chapter 2: Assemble MatchPort/XBee wireless board

Step#18, Review OPEN-ROBOT MatchPort/XBee Contents:

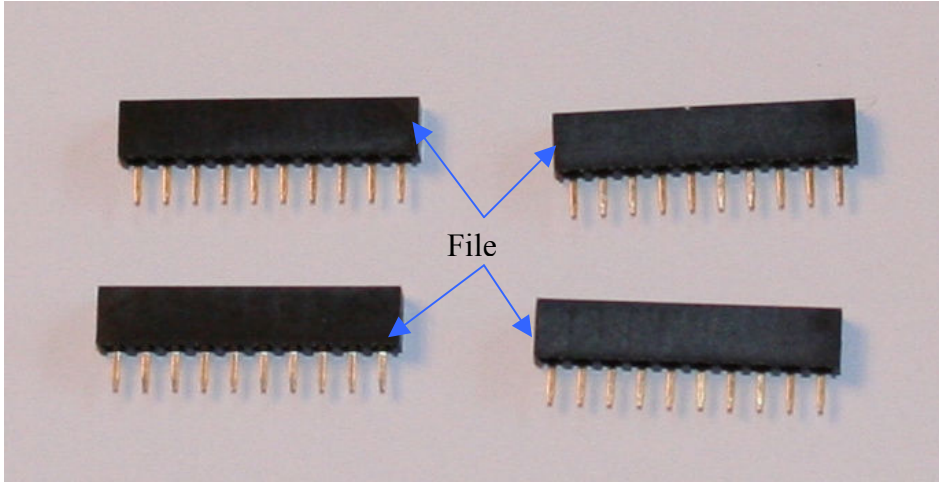
1. (1) – OPEN-ROBOT MatchPort/XBee PCB
2. (4) – 2mm pitch MatchPort/XBee sockets, U1 headers
3. (1) – SPST switch, SW1
4. (4) – 0.1” pitch 2-position non-polarized headers, J4 & J5
5. (2) – 0.1” pitch jumper
6. (1) – 0.1” pitch 3-position terminal, J3
7. (2) – 0.1” pitch 2-position terminal block, J1 & J6
8. (1) – DE-SW033 or RECOM 3.3volt switching regulator
9. (1) – 74LVC245 octal buffer
10. (1) – 74HC245 octal buffer
11. (1) – 32-position socket for connecting SRV1 Camera Board, J2 (*purchase separately*)
12. (1) – 100uF Electrolytic capacitor (*only with RECOM*)
13. (1) – 0.1uF Tantalum capacitor (*only with RECOM*)



Figure#18. Review Kit Contents.

Step#19, Prepare the 2mm pitch MatchPort/XBee sockets:

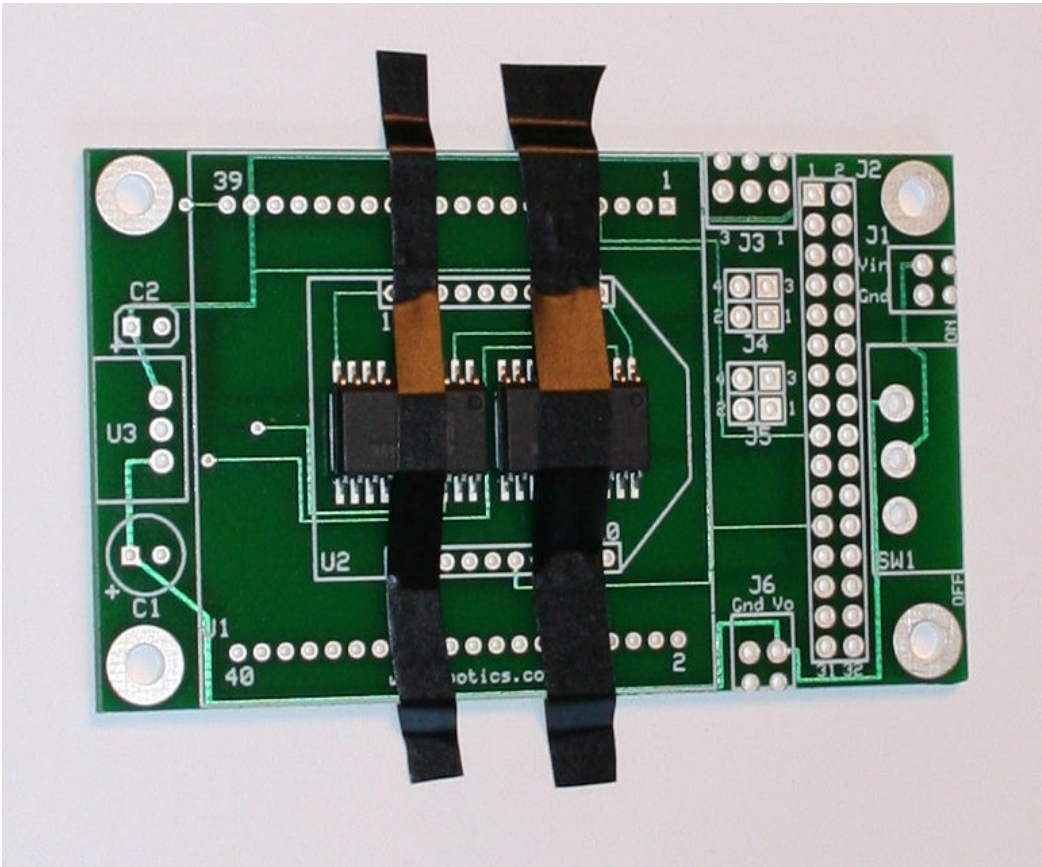
NOTE: Skip this step if you are using the XBee ZigBee module.



Figure#19. File ends of 2mm MatchPort/XBee sockets.

You will need to slightly file or sand one end on each of the 2mm pitch MatchPort/XBee sockets so that they fit on the PCB. Be careful not to file too much of the plastic otherwise the metal pin will be exposed. You only need to perform this step if you are using the MatchPort b/g module.

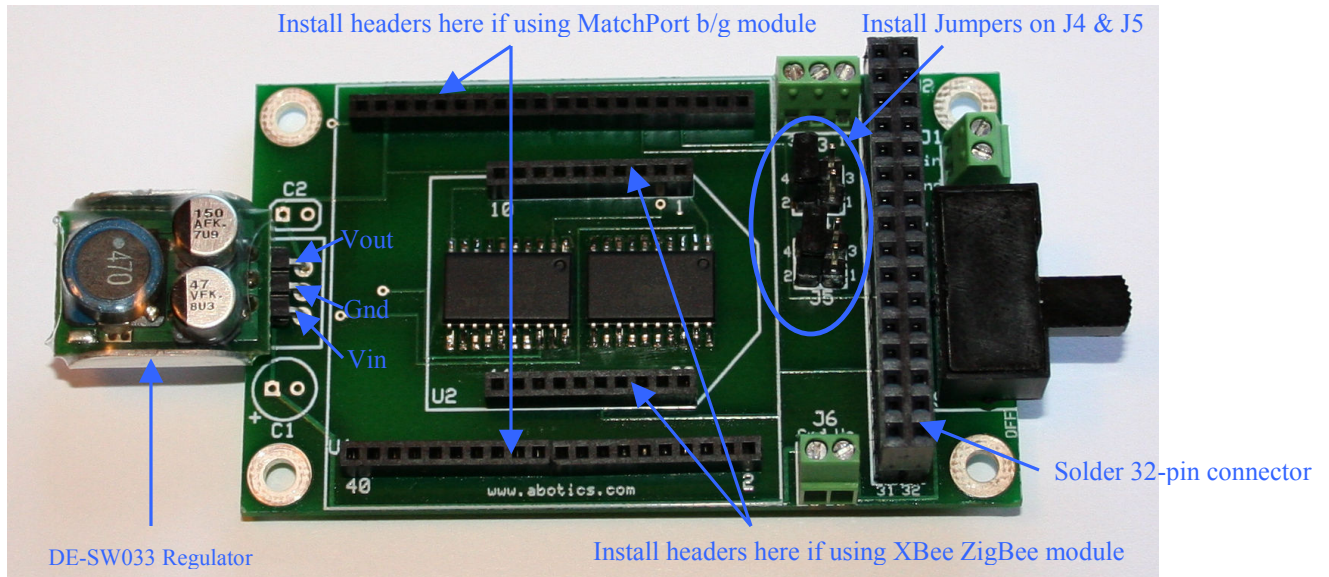
Step#20, Solder 74LVC245 and 74HC245 Octal Buffer:



Figure#20. Solder 74LVC245 and 74HC245 Octal Buffers.

The 74LVC245 and 74HC245 chips are fine pitch SOIC chips and are not as easy to solder as say a 0.1" PDIP component. However, there is something that will make this a lot easier. Simply cut two small pieces of tape and then use these to secure each chip to the PCB and also to your work surface. Be sure to orient the chips as shown and in their proper location on the board. Now you can easily solder using 0.015" diameter solder (RadioShack#64-035) and a fine tip soldering iron. Don't worry if you accidentally solder-bridge two pins together during the solder process because in the end you can use a piece of de-soldering braid (RadioShack#64-2090) to remove the excess solder.

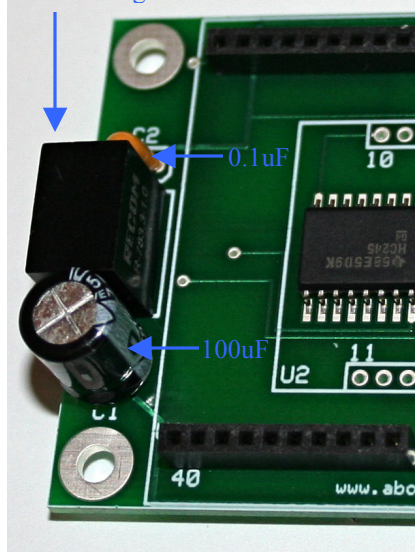
Step#21, Solder Remaining Components:



Figure#21a. Solder remaining components.

Solder the remaining components according figure#21. If you are going to add the SRV1 Camera Board then you will need to bend the DE-SW033 pins 90 degrees so that it won't interfere. Also be sure to add the 0.1" jumpers to J4 and J5. If you are adding the SRV1 Camera Board, then attach the jumpers to pins 1&3 on both J4 and J5. Otherwise attach the jumpers to pins 2&4 on both J4 and J5. Figure#21a shows the jumpers attached to pins 2&4 and is therefore configured for normal non-SRV1 operation.

RECOM Regulator

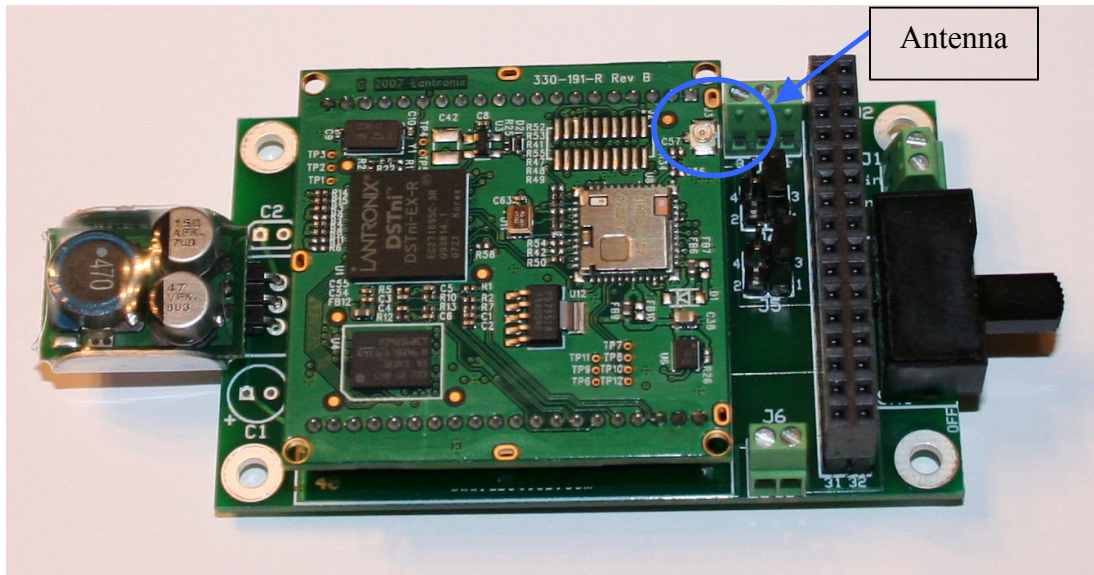


Figure#21b. RECOM 3.3 volt regulator installation.

NOTE: If your kit has a RECOM 3.3 volt regulator instead of the DE-SW033, then install as shown in figure#21b. Be sure to solder the RECOM regulator such that the RECOM text is facing the HC245 octal buffer. You will also need to solder a 100uF electrolytic capacitor in location C1 and a 0.1uF tantalum capacitor in location C2.

***NOTE:** You should clean the solder flux from your board prior to installing the MatchPort b/g module.

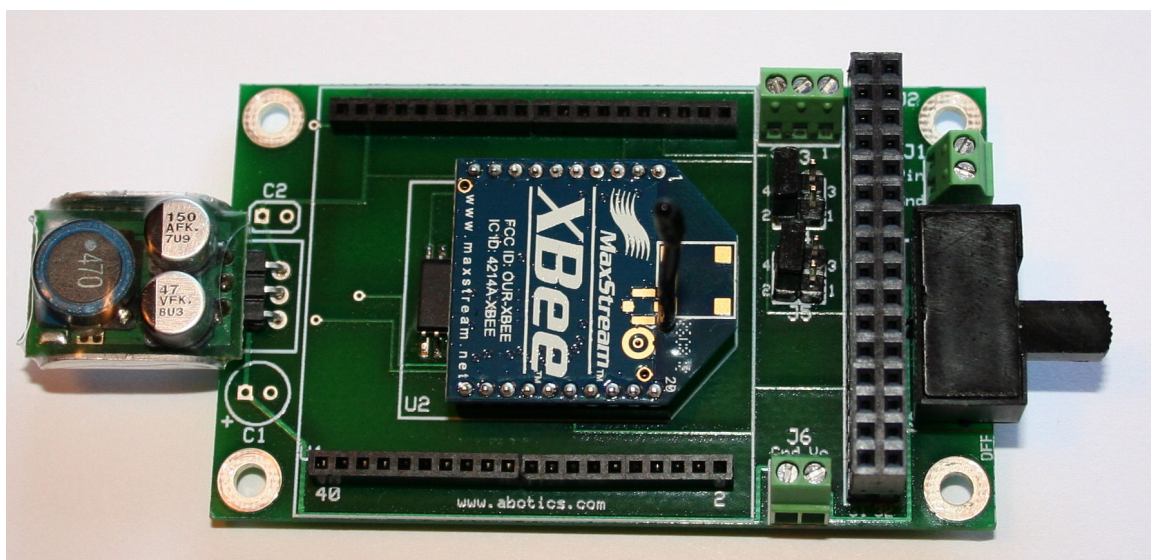
Step#22a, Install MatchPort b/g Module (purchase separately):



Figure#22a. Fully assembled MatchPort/XBee PCB with MatchPort b/g installed.

Install the MatchPort b/g module by orientating the antenna clip towards the 3-position terminal block, J3. Be sure to properly align the pins before firmly pressing into place. Please refer to figure#22a. If you are using the XBee ZigBee module, then refer to figure#22b. The remaining documentation will display the MatchPort b/g. If you are using the XBEE module simply ignore the MatchPort b/g.

Step#22b, Install XBee ZigBee Module (purchase separately):



Figure#22b. Fully assembled MatchPort/XBee PCB XBee module installed.

Chapter 3: How to Test OPEN-ROBOT Circuit Boards

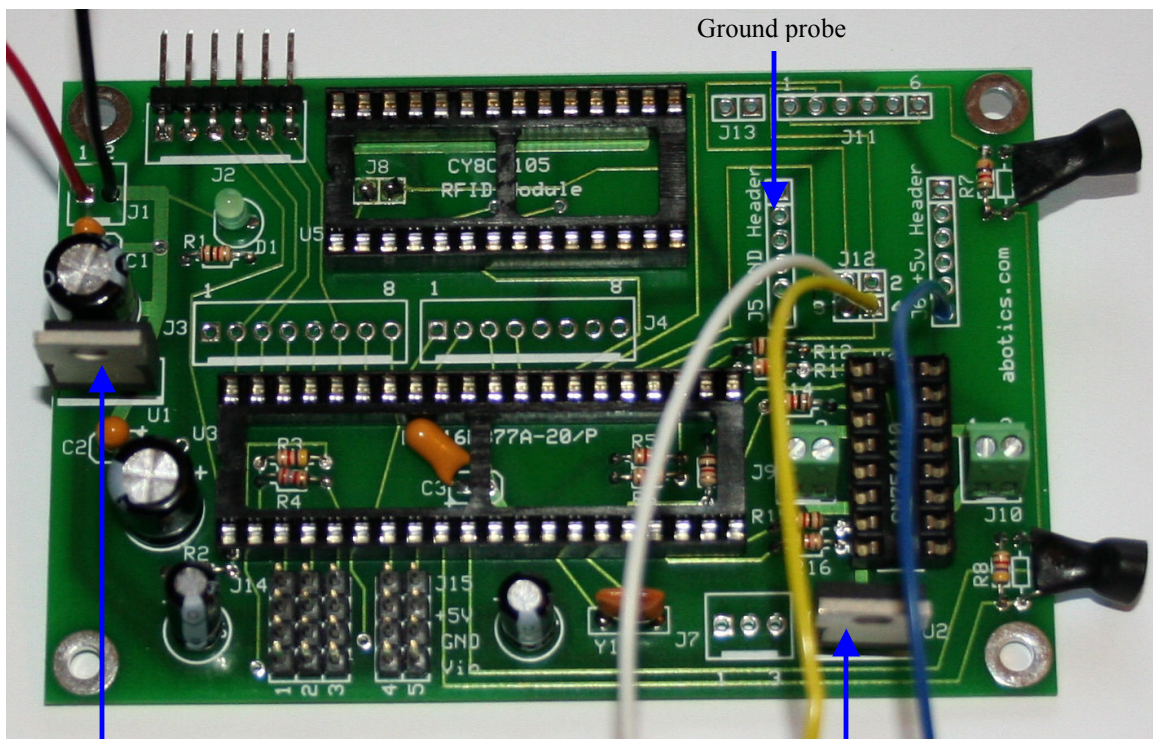
Prerequisites & Setup:

1. Soldered Main PIC18F4520 circuit board & MatchPort/XBEE circuit board preferably *without* chips installed, i.e. PIC18F4520, h-bridge, RFID module, MatchPort b/g or XBee module.
2. Voltmeter capable of measuring up to 10 volts
3. Connect the J1 red and black wires from the PIC18F4520 board to the J6 connector on the MatchPort/XBEE board's J6 connector. The red wire connects to J6 Vo and the black wire to J6 Gnd. This will provide power to the PIC18F4520 board.
4. Connect the (6)-AA battery holder's red and black wires to the J1 connector on the MatchPort/XBee board. The red wire from the battery holder goes to J1 Vin and the black wire goes to J1 Gnd. This setup will provide power to both boards. ***At this point, do not install the AA batteries and leave the power switch off!***
5. Be sure to tape off the blue wire coming from the Main PIC18F4520 board because this wire will have +5volts when the power is turned on and we do not want a short circuit to develop.

Test Procedure:

1. Make sure the MatchPort/XBEE board's power switch is off and then install the (6)-AA batteries. ***Leave the power switch off!***
2. ***Leave the power switch off!*** Check for +7 to +9volts at the MatchPort/XBEE board's J1 connector. Place the ground probe (usually black) from your meter on the J1 Gnd pin and the signal probe (usually red) on the J1 Vin pin.
 - a. If you don't read +7 to +9volts, then the batteries are dead or inserted improperly in the holder (Check battery polarity in the holder. Flat end of battery must be against spring).
3. Now turn the ***power switch on*** and carefully feel both of the +5volt regulators on the PIC18F4520 board, U1 & U2 and the +3.3volt regulator, U3, on the MatchPort/XBEE board, U1. If any of these regulators is hot to the touch then quickly ***turn off power*** because there is a short circuit some place on one of the boards. The green LED should be lit at this point. Turn the power switch off.
 - a. If a short is discovered, then you need to take a look at both boards to determine if it is related to a bad solder joint or a regulator that is soldered in backwards or a mis-wiring of the battery pack or the J1 power wires from the PIC18F4520 board.

4. Turn the power switch on and check for +7 to +9volts at the MatchPort/XBEE board's J6 connector. Place the ground probe from your meter on the J6 Gnd pin and the signal probe on the J6 Vo pin.
 - a. If you do not measure the correct voltage, then the switch might be defective.
5. Check for +5volts on the Main PIC18F4520 board's +5volt logic regulator, U1. Place the ground probe from your meter on the J5 Gnd Header and the signal probe on the metal body of U1. Also check for +5volts at the +5volt motor regulator, U2, by leaving the ground probe on J5 and placing the signal probe on the metal body of U2.
 - a. If you do not read +5volts, then something might be wrong with the corresponding +5volt regulator. Check to be sure the regulator is soldered in properly. If soldered properly, then it may be defective.
6. Check for +3.3volts at the +3.3volt regulator, U3. Place the ground probe from your meter on the J6 Gnd pin and the signal probe on the +3.3volt pin on the U3 +3.3volt regulator.
 - a. If you do not measure +3.3volts then the regulator might be soldered in wrong.



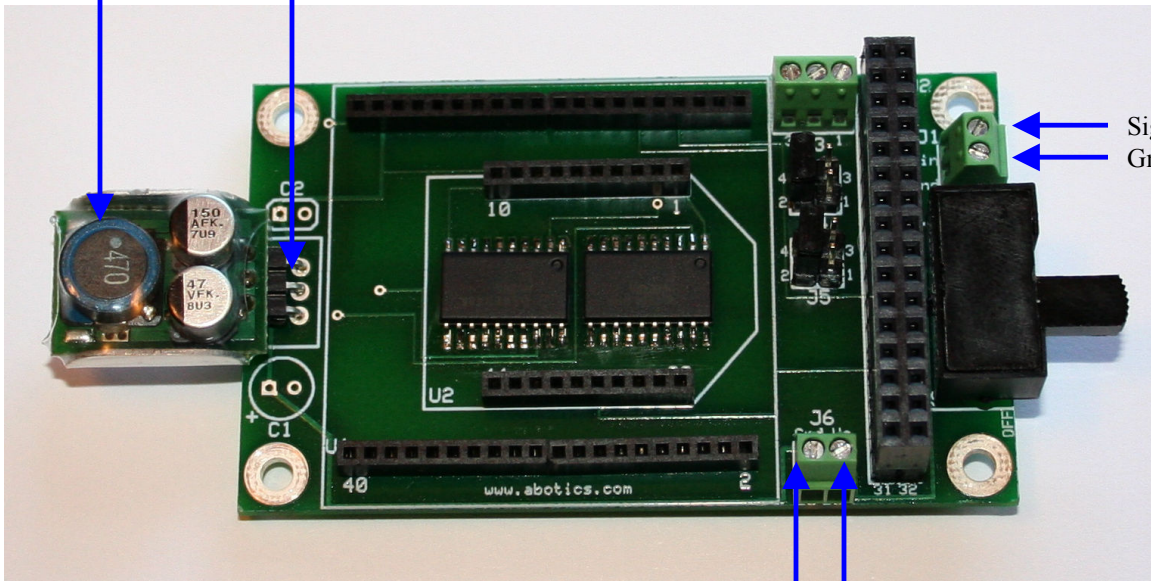
Signal probe for U1

Signal probe for U2

[Main PIC18F4520 Board Test Points.](#)

+3.3volt Regulator, U3

Signal probe, +3.3volts



Signal probe, Vin
Gnd probe, Gnd

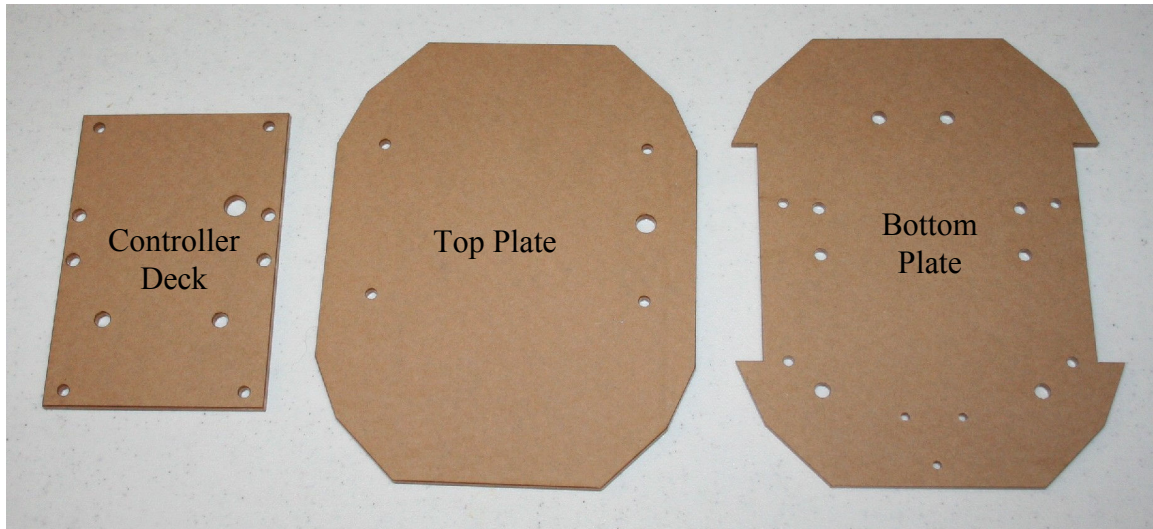
MatchPort/XBEE Board Test Points.

Gnd probe, Gnd

Signal probe, Vo

Chapter 4: OPEN-ROBOT body Assembly

Step#23, Assemble OPEN-ROBOT Body:



Figure#23, Prepare OPEN-ROBOT body pieces.

The first step in assembling the OPEN-ROBOT's body is to prepare the acrylic body pieces. You need to peel off the brown protective paper coating from each piece so that we can begin the assembly process.

***NOTE:** Please note that the color of your body pieces may vary.

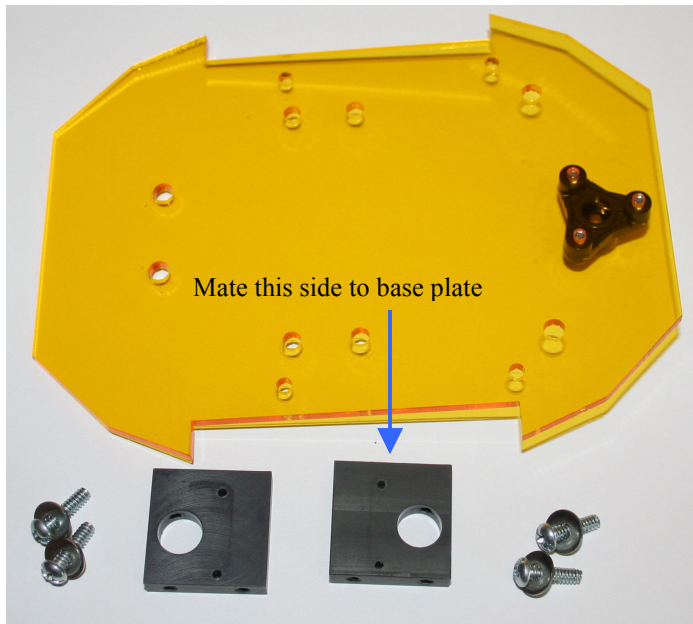
Step#24, Assemble and Attach Rear Caster Wheel:



Figure#24, Assemble and mount rear caster wheel.

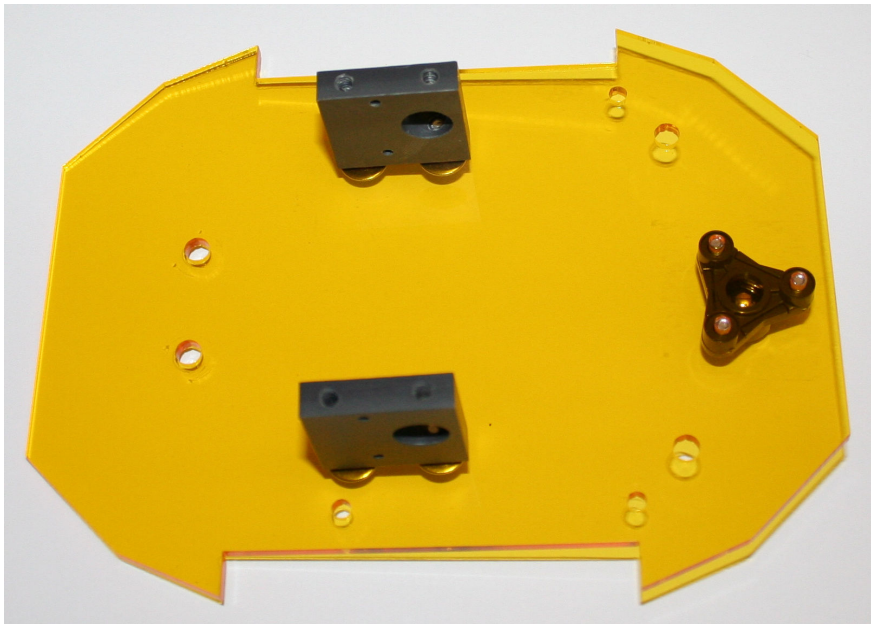
To assemble the caster wheel you need to first insert the metal roller ball then add the (3)-needle rollers and finally attach the (3)-hole cover plate. Use the (3) #4-40x3/8" long machine screws to attach the plastic caster wheel to the base plate.

Step#25, Attach GM8 Motor Mounts:



Figure#25a, Attach GM8 motor mounts to base plate.

In step#25 attach each GM8 motor mount using (4) #6-32x1/2" long machine screws and (4) #6 flat washers. The GM8 mounts must mate on the side of the base plate that is opposite to the caster wheel. If you look closely on each GM8 mount you will notice that one of the #2-56 threaded holes is drilled further away from the edge. This side must be mated to the base plate. Refer to figure#25a if you are unsure.



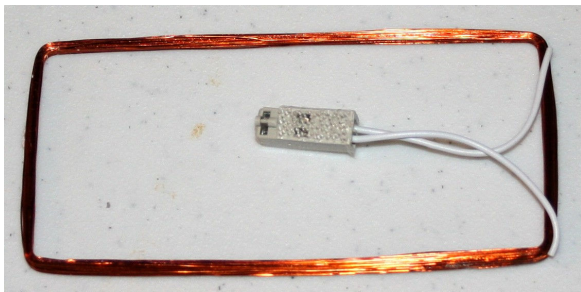
Figure#25b, Attach GM8 motor mounts to base plate.

Step#26, Prepare to Attach RFID Antenna (purchase separately):

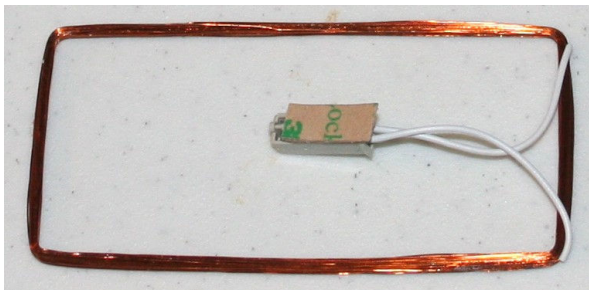


Figure#26a, Prepare to attach RFID antenna.

In step#26 attach a small piece of the 3M Dual Lock® tape to the plastic connector on the RFID antenna. After attaching the first piece, cut a second piece and snap it onto the first piece. Be very careful when handling the RFID antenna because it is very fragile.

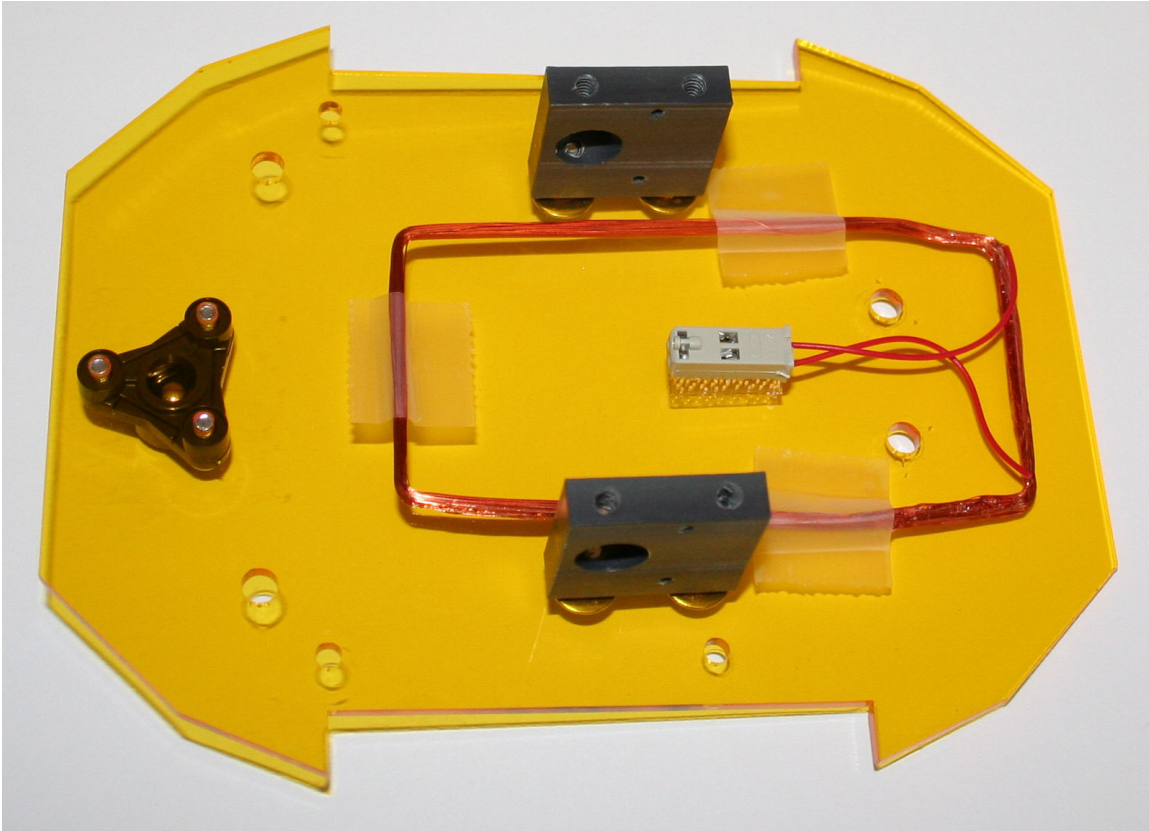


Figure#26b, Attach small piece of tape to connector.



Figure#26c, Cut second piece of tape and snap onto first piece.

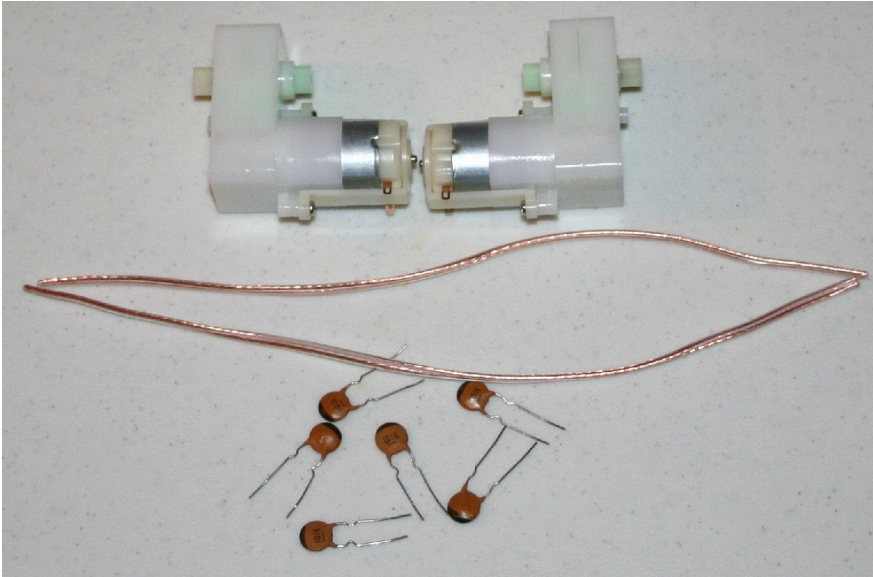
Step#27, Attach RFID Antenna to Base Plate:



Figure#27, Attach RFID antenna to base plate.

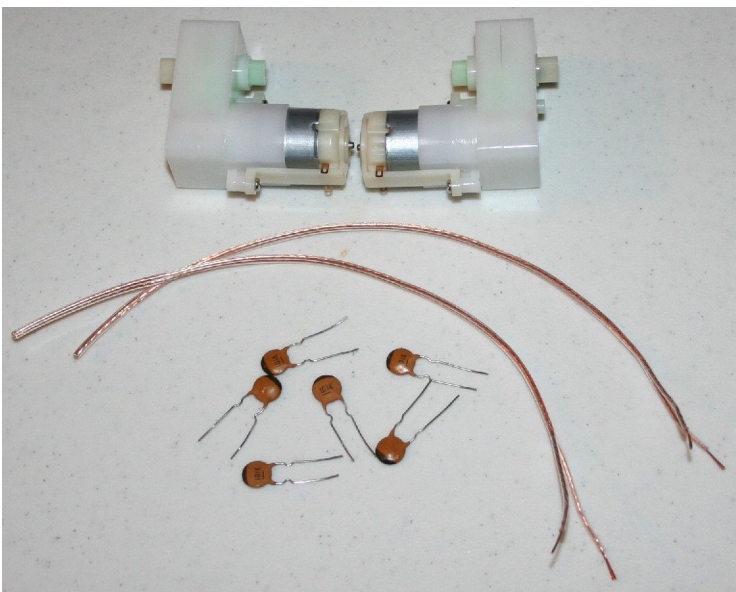
In step#27 peel the paper backing off of the piece of Dual Lock tape on the antenna connector and carefully locate the antenna on the base plate and once you have it in position firmly press the plastic connector onto the base. The antenna should be centered in between the two GM8 mounts and the front edge of the antenna should be approximately 1/8" –1/4" from the front edge of the base plate. Finally secure the antenna to the base plate using three pieces of clear plastic tape.

Step#28, Prepare to Solder Noise Filtering Capacitors (Optional):



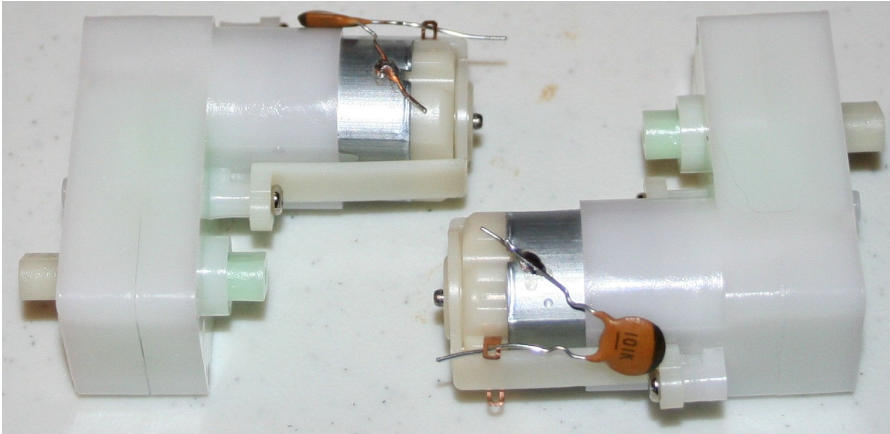
Figure#28a, Prepare to solder noise filtering capacitors to GM8 motors.

First separate and strip back the ends of your 8" long clear motor wires. You will need to separate each conductor by cutting the insulator in between and then peeling the two conductors apart. Strip back approximately 1/8"-3/16" **Note: The noise filtering capacitors are only needed if you plan to install the WW02 Wheel Encoders.**



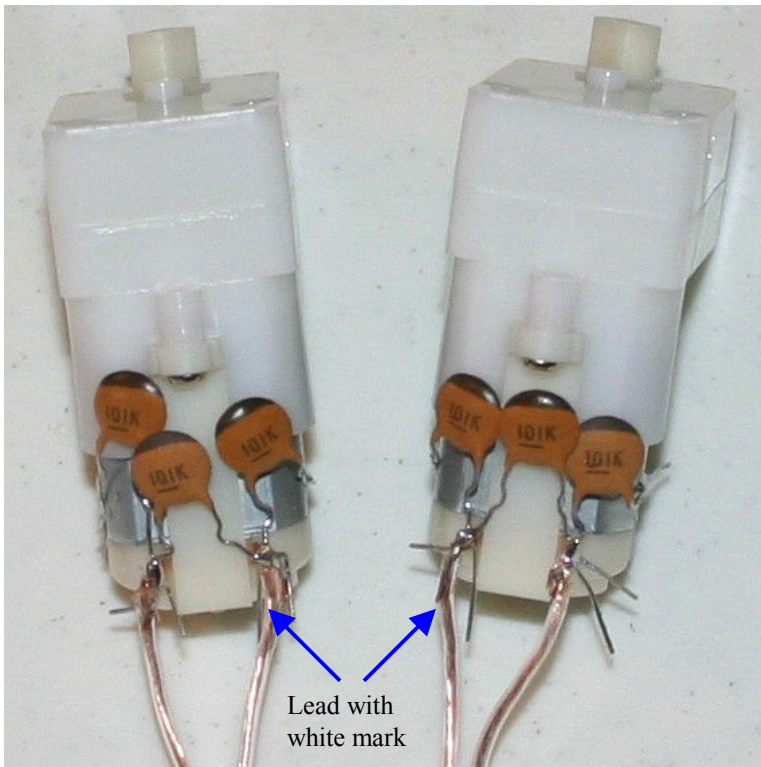
Figure#28b, Strip 8" long GM8 motor leads and peel apart.

Step#29, Solder Noise Filtering Capacitors:



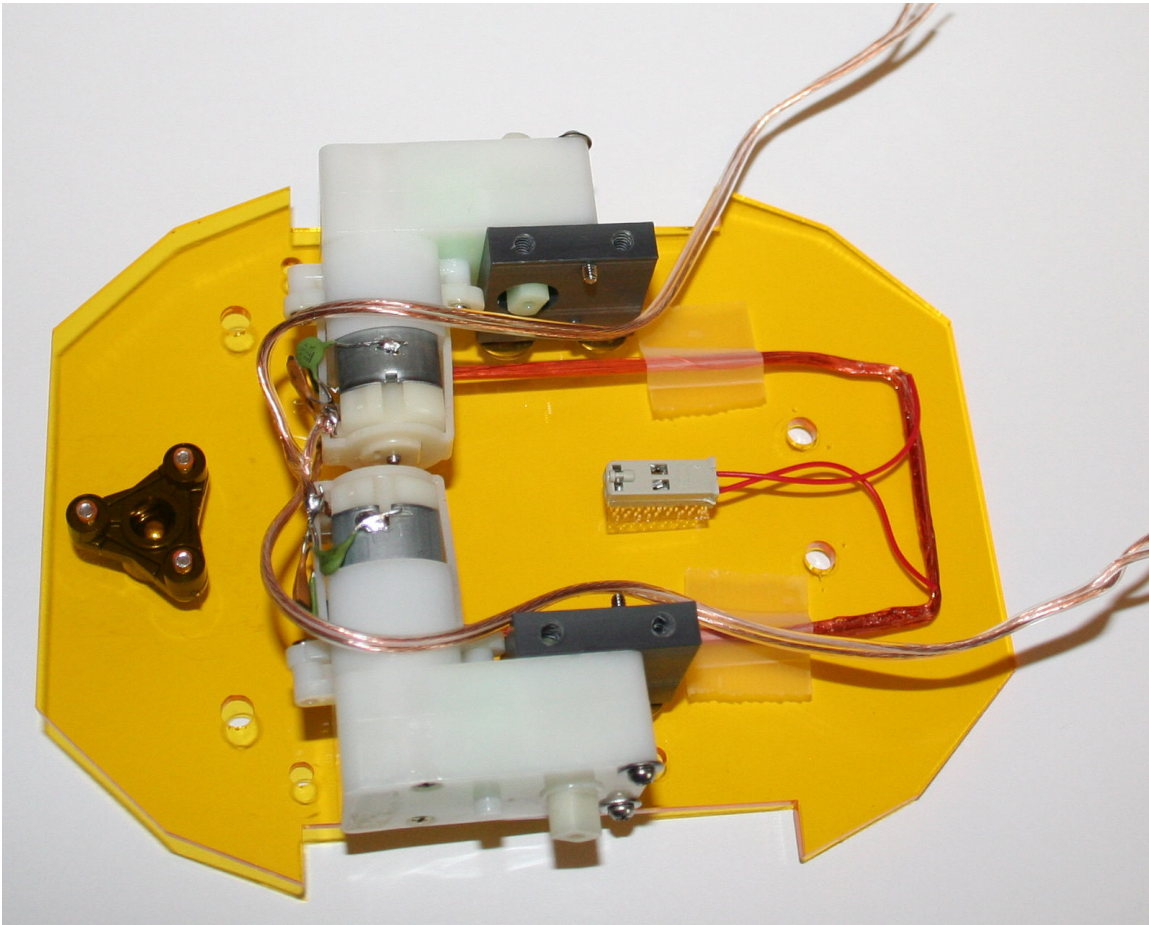
Figure#29a, Solder 100pf ceramic capacitors.

Insert one leg of a 100pf ceramic capacitor into one of the metal GM8 motor tabs and then solder the other capacitor leg to the case of the GM8 motor. You will need to do this twice on each motor. Don't solder the leg that was inserted into the motor tab yet. Take a third capacitor and insert one leg into one of the motor tabs and the other leg into the second motor tab. Finally insert the striped ends of the 8" motor lead into the motor tabs as shown in figure#29b and solder. Be sure to insert the leads with the white marker on the clear insulator into the appropriate motor tab.



Figure#29b, Solder 100pf ceramic capacitors and 8" motor leads.

Step#30, Attach GM8 Motors:

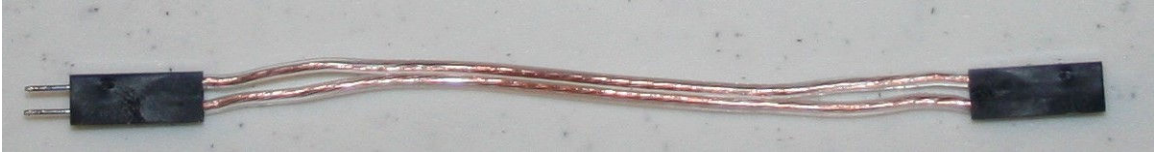


Figure#30, Attach GM8 motors.

Insert and press the GM8 motors into the GM8 mounts. Be sure to mount the motors with the white marked motor lead on the topside away from the base plate. Use the four supplied #2-56x1" machine screws and #2 washers to secure the GM8 motors to the mounts.

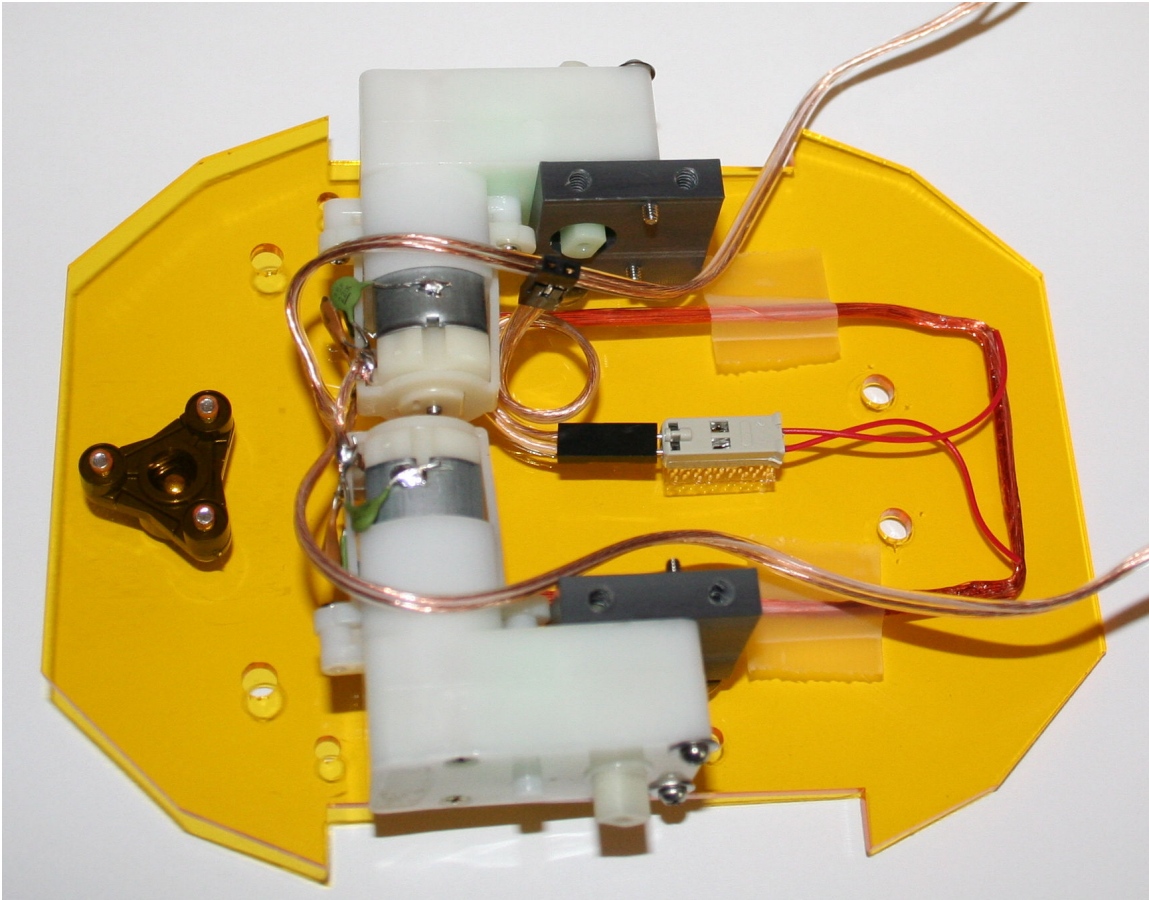
Step#31, Prepare RFID Antenna Cable (*purchase separately*):

In step#31 prepare to attach the RFID Antenna cable. This cable will connect the RFID antenna to the bottom of the controller board, J8. The female end will later attach to the bottom of the controller board, J8. The male end will connect to the RFID antenna.



Figure#31, RFID antenna cable.

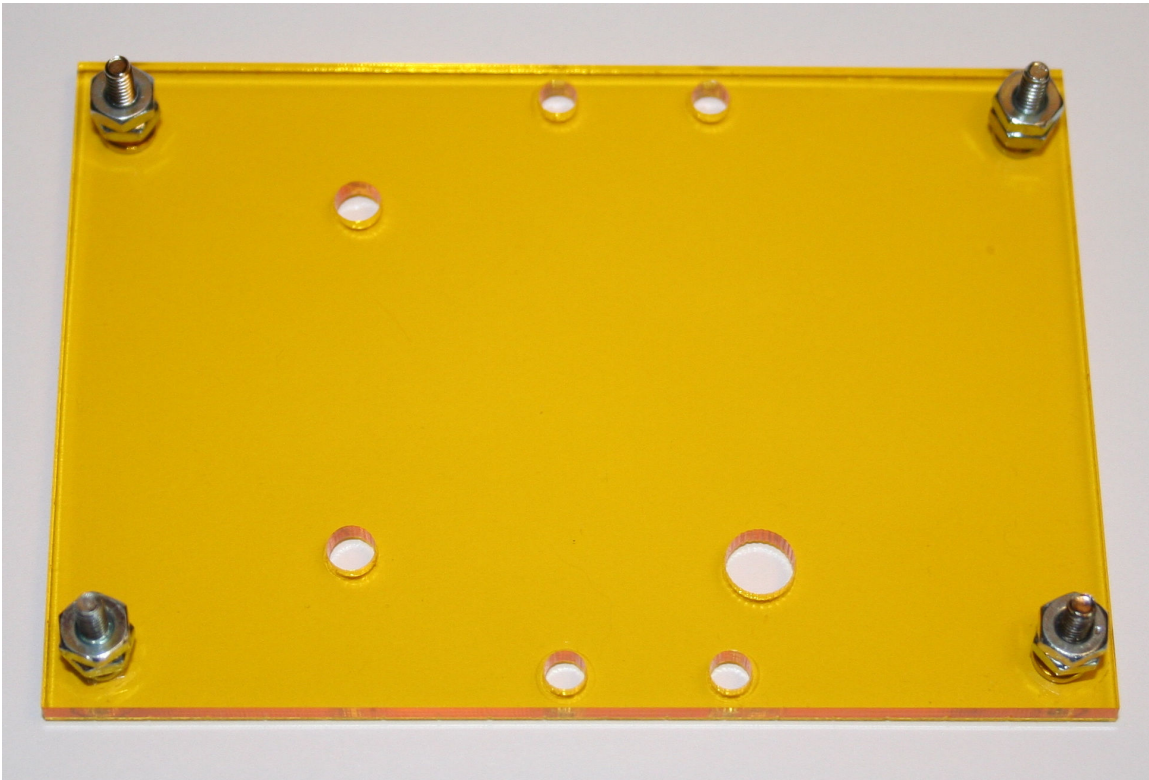
Step#32, Attach RFID Antenna Cable & Prepare to Mount Controller Deck:



Figure#32, Attach RFID antenna cable.

Insert the male end of the RFID antenna cable into the antenna connector and bend the cable slightly upwards as shown in figure#32. Be careful when plugging the RFID cable into the RFID antenna connector because the RFID antenna wires are delicate.

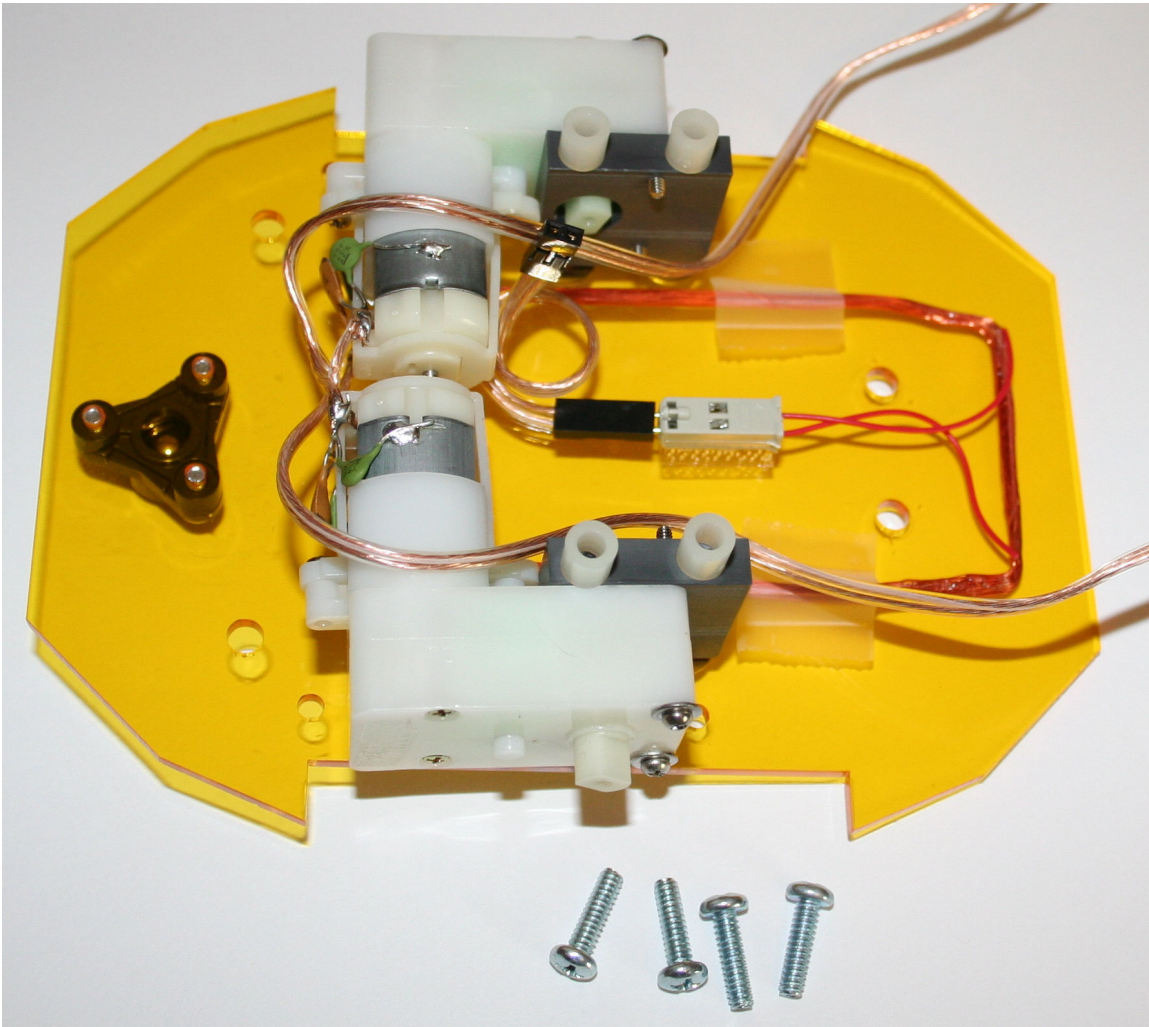
Step#33, Prepare Controller Deck:



Figure#33, Prepare Controller Deck.

Prepare the controller deck by attaching the (4) #4-40 machine screws. Use (2) #4-40 nuts on each screw so that there is enough clearance for the PIC18F4520 Controller Board.

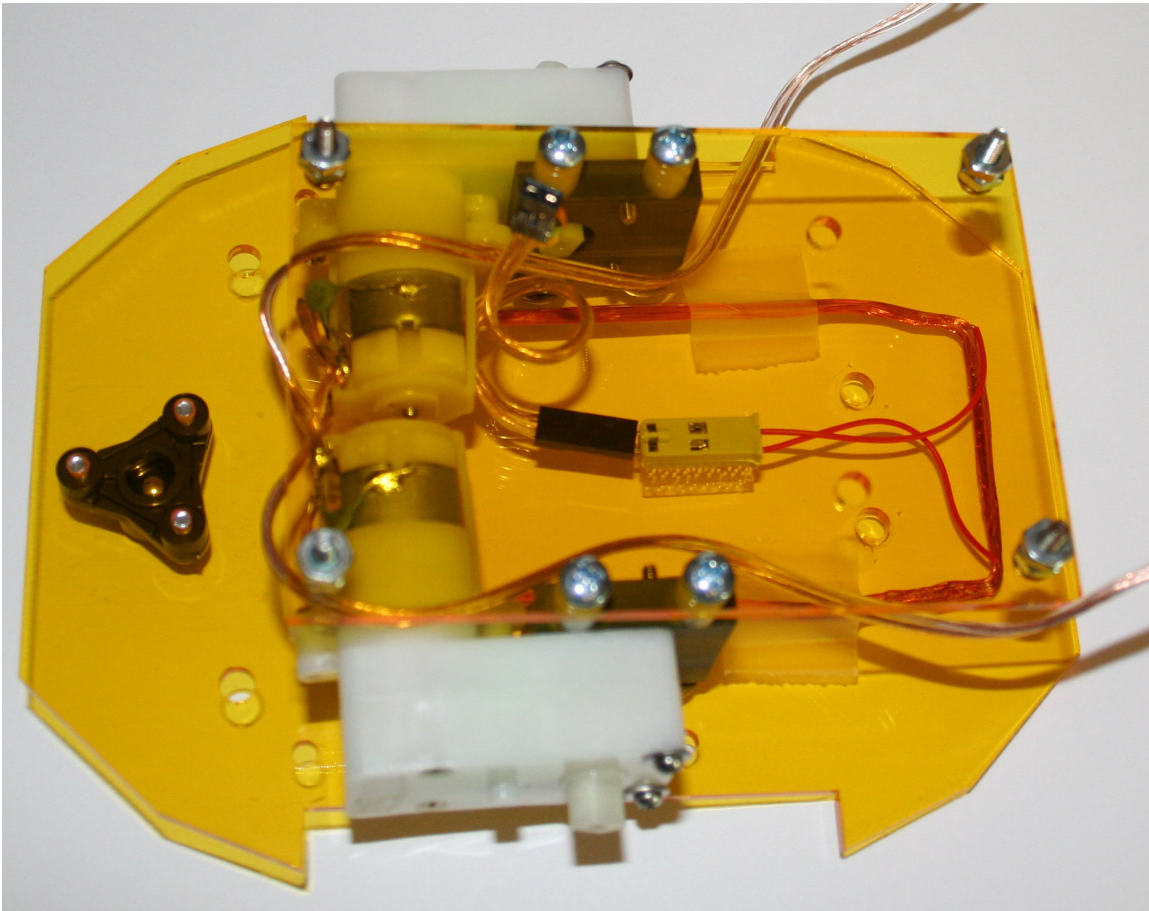
Step#34, Attach Controller Board Standoffs:



Figure#34, Attach controller board standoffs.

In step#34 place the #6 nylon standoff spacers on top of each GM8 motor mount.

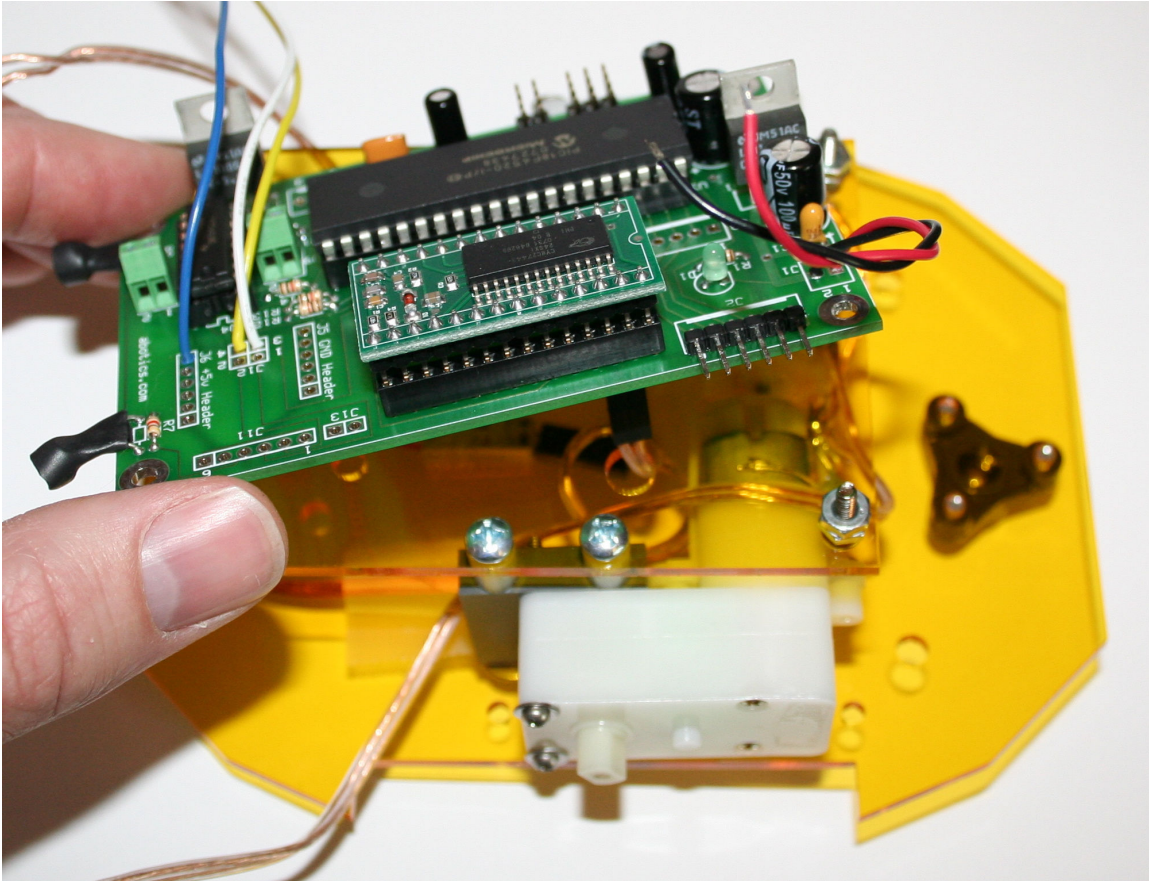
Step#35, Mount Controller Deck:



Figure#35, Mount Controller deck.

Mount the controller deck and secure using (4) #6-32x5/8" machine screws.

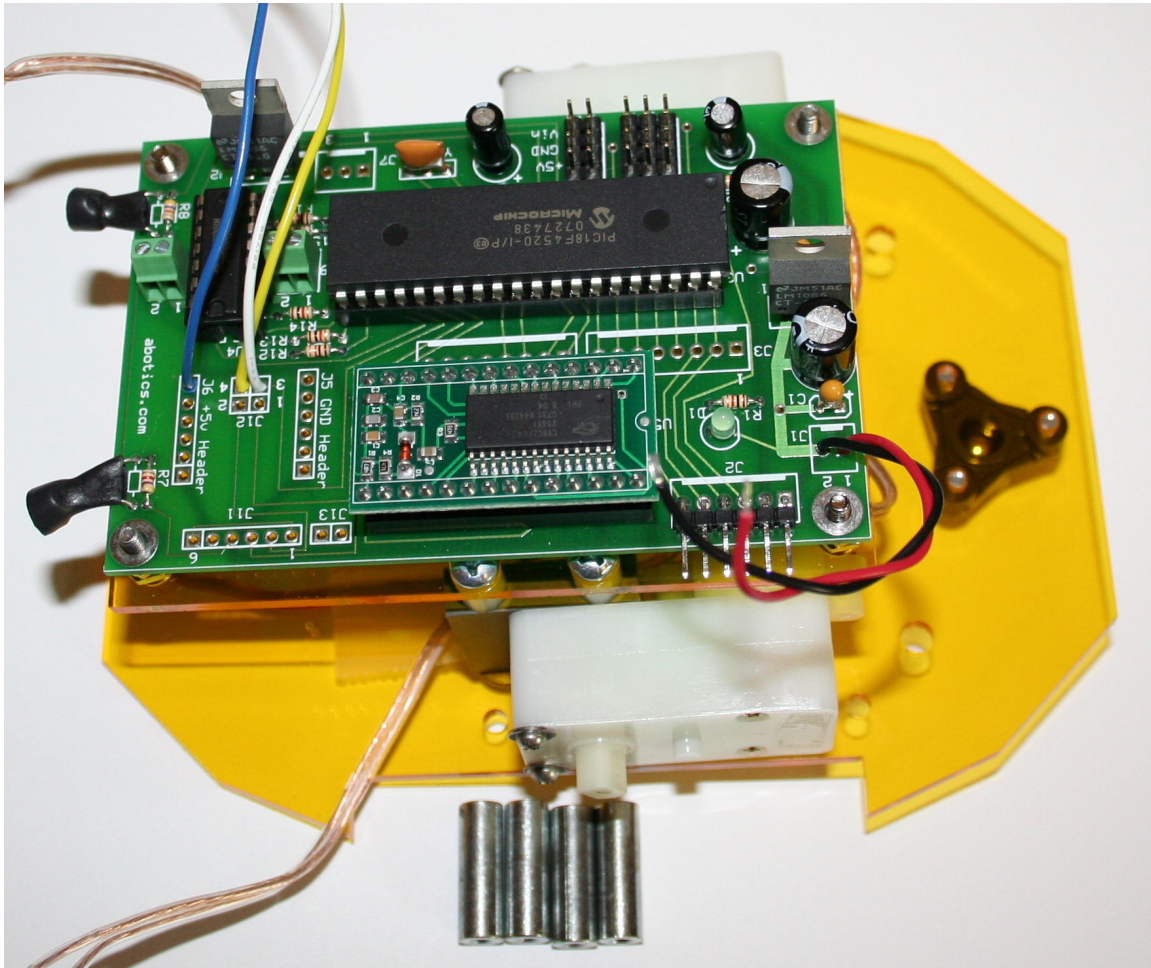
Step#36, Mount Controller Board & Connect RFID Cable:



Figure#36, Connect RFID cable to J8 on Controller Board.

In step#36 connect the RFID cable to J8 on the Controller Board and then align the controller board with the (4) #4-40 machine screws.

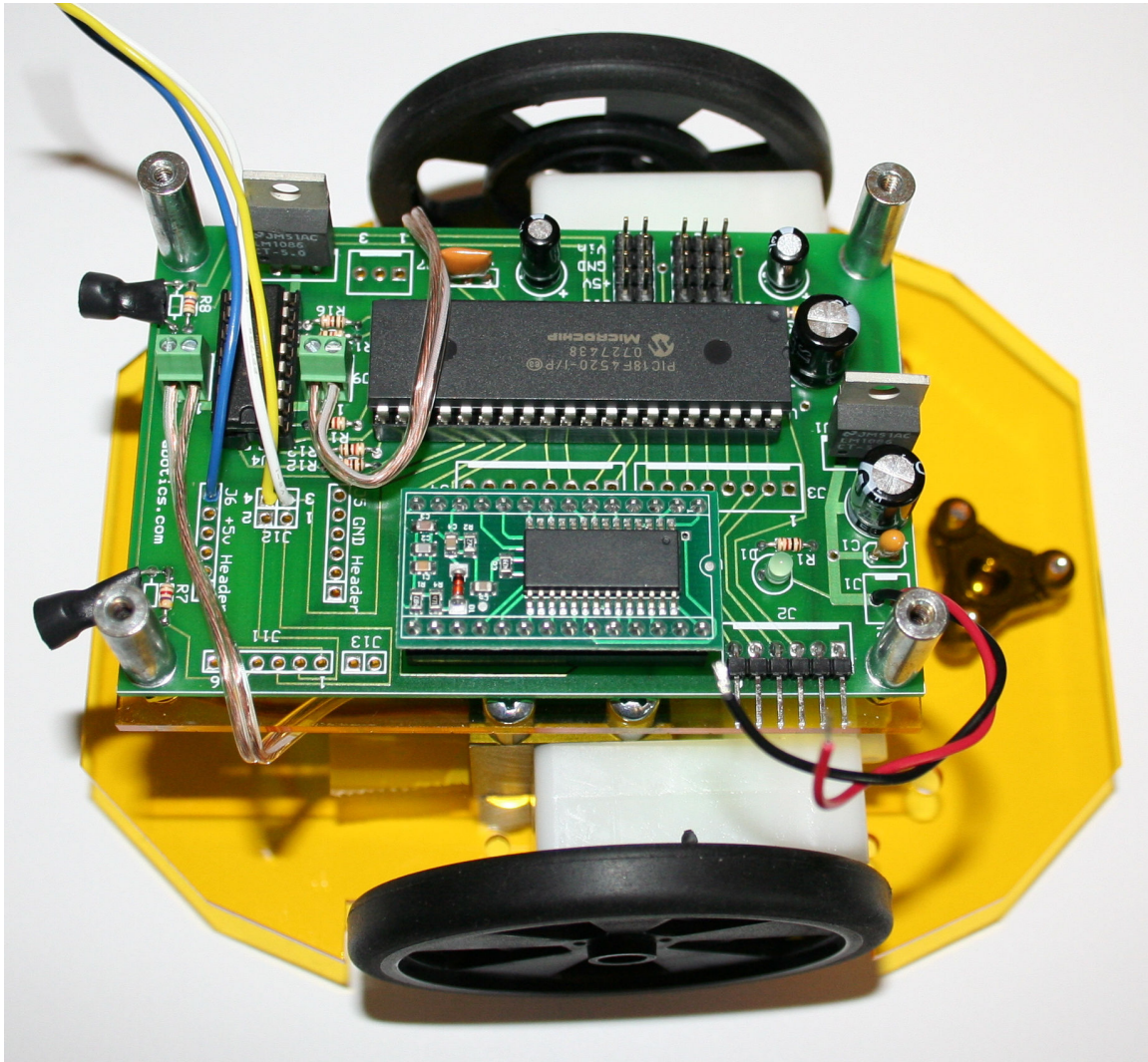
Step#37, Attach MatchPort/XBee Deck Standoffs:



Figure#37, Prepare to attach WW02 Wheel Encoders.

In step#37 thread a 0.75” long standoff onto each of the #4-40 machine screws. This will secure the controller board and allow us to mount the MatchPort/XBee deck later.

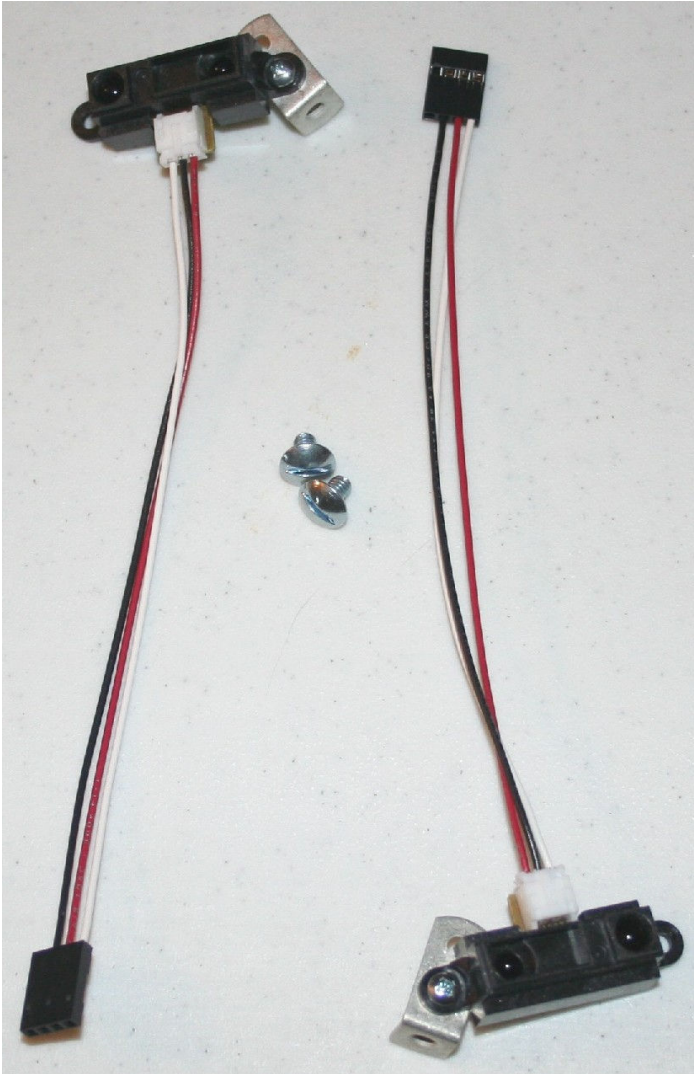
Step#38, Attach GM8 Wheels:



Figure#38, Attach GM8 Wheels.

Attach both the left and right GM8 wheels. Gently press them onto the GM8 motor output shaft

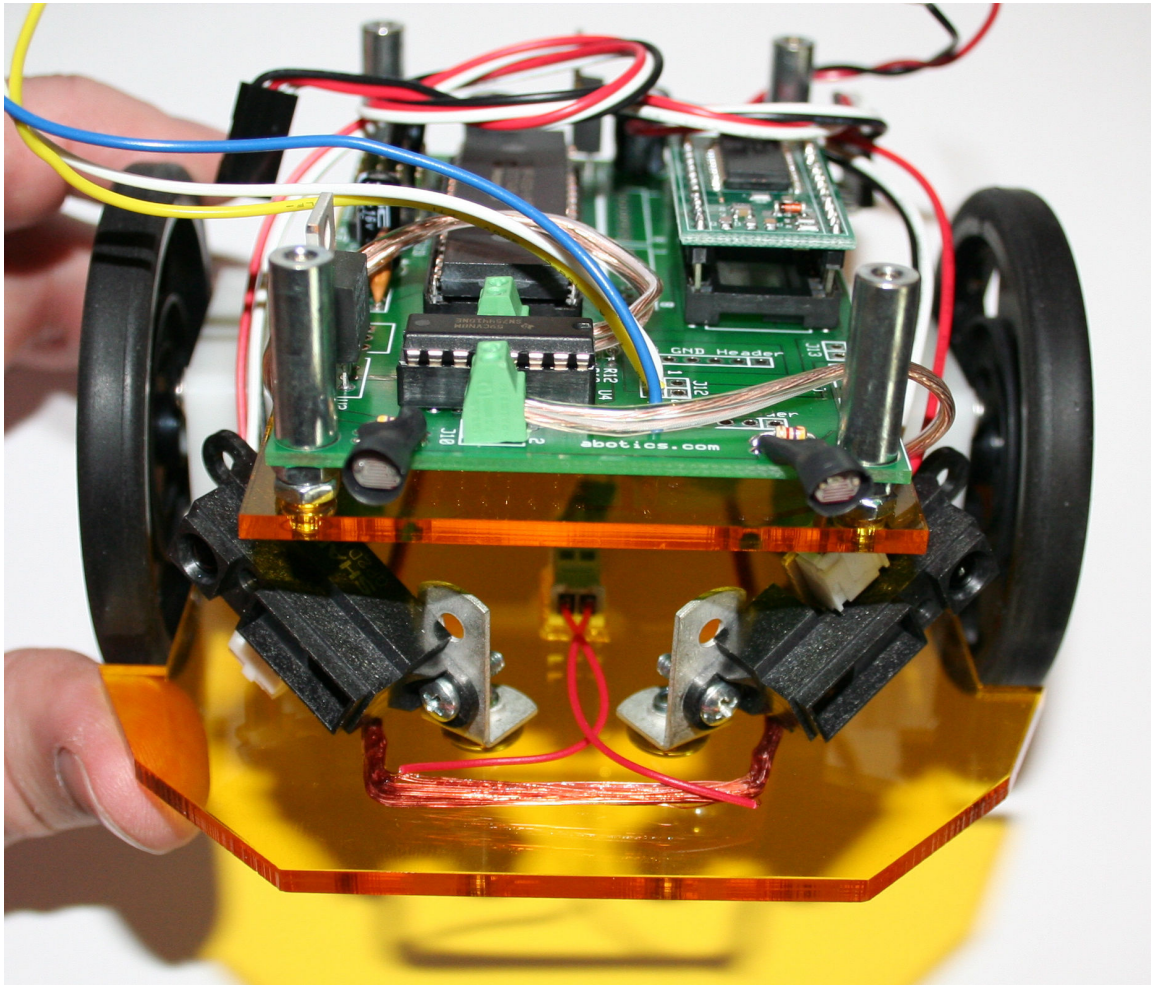
Step#39, Mount GP2D120 Sensors to Brackets:



Figure#39, Mount GP2D120 sensors to brackets.

Mount the GP2D120 sensors to the angle brackets using (2) #4-40x3/8" long machine screws. Try to mount the sensors at a 45-degree angle. Attach the GP2D120 sensor cable by aligning the small white JST 3-pin connector and then firmly press into place.

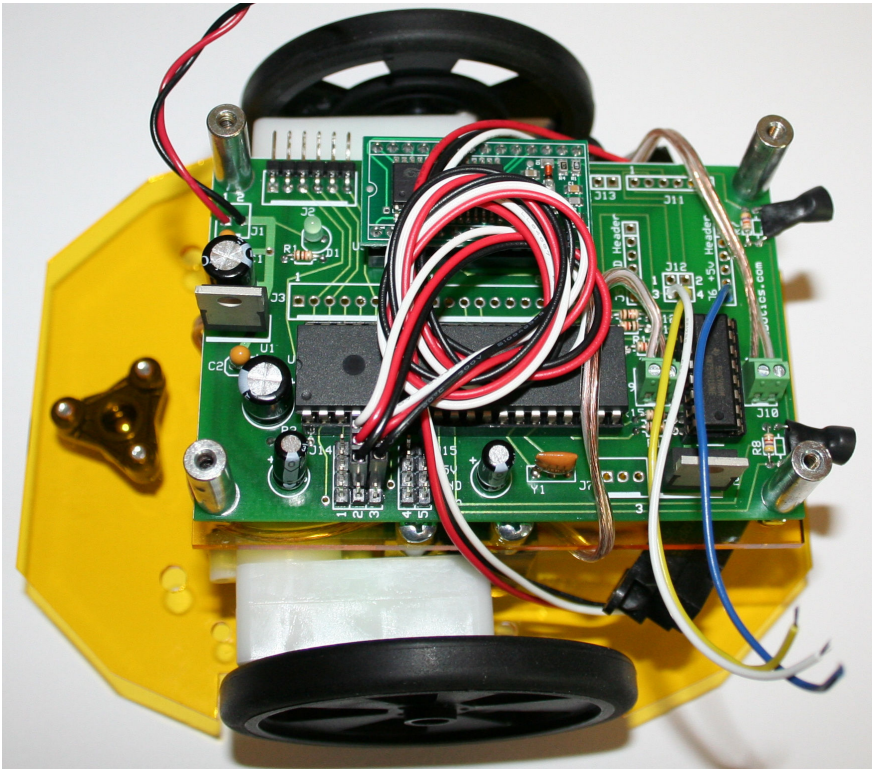
Step#40, Mount GP2D120 Sensors to Robot:



Figure#40, Mount GP2D120 sensor brackets to robot base plate.

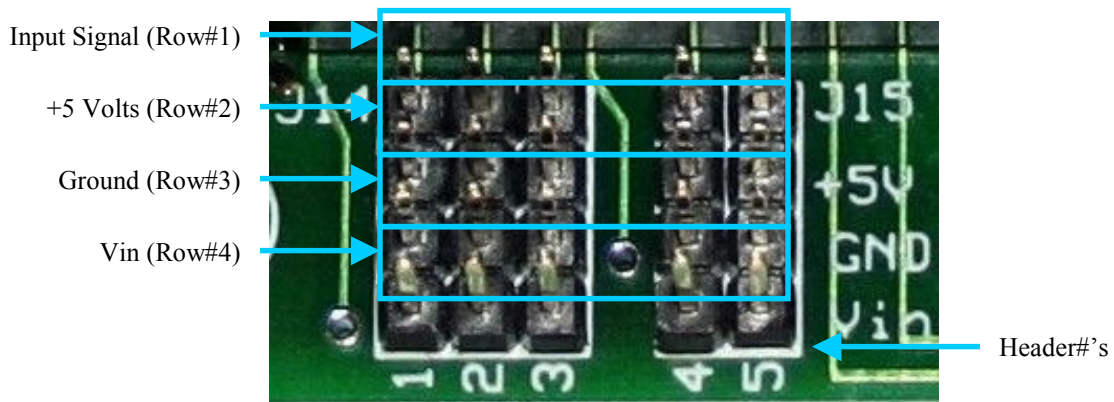
Mount the GP2D120 sensor brackets to the robot base using (2) #8-32x1/4" long truss head screws. You can adjust the sensors to look in whatever direction you want. You will need to bend the sensor wires close to the white JST 3-pin sensor connector so that the brackets can be bolted to the base plate. Please note that you may have to slightly bend the RFID antenna out on each side so that there is enough clearance for the GP2D120 mounts.

Step#41, Connect GP2D120 Sensors to the Controller Board:



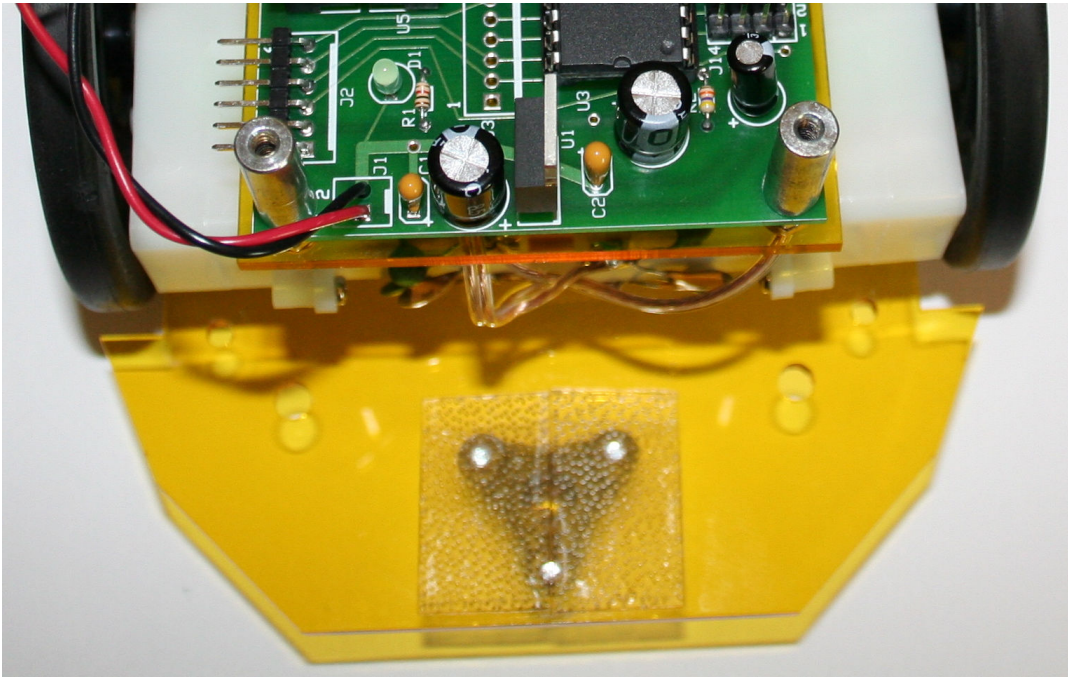
Figure#41a, Connect GP2D120 sensors to the controller board.

In step#41 connect the GP2D120 sensor cables to the controller board. The left cable must connect to J14, header#2, input signal row pin A/D#2 and the right cable connects to J14, header#3, input signal row pin A/D#3. Be sure that the red wire connects to the +5 Volts row and the black wire connects to the ground row. The white wire is the signal from the GP2D120 and connects to the input signal pin, which is closest to the PIC18F4520 microcontroller.



Figure#41b, Close-up of J14 headers on Controller board.

Step#42, Attach Dual Lock Tape to Base and Battery Holder:



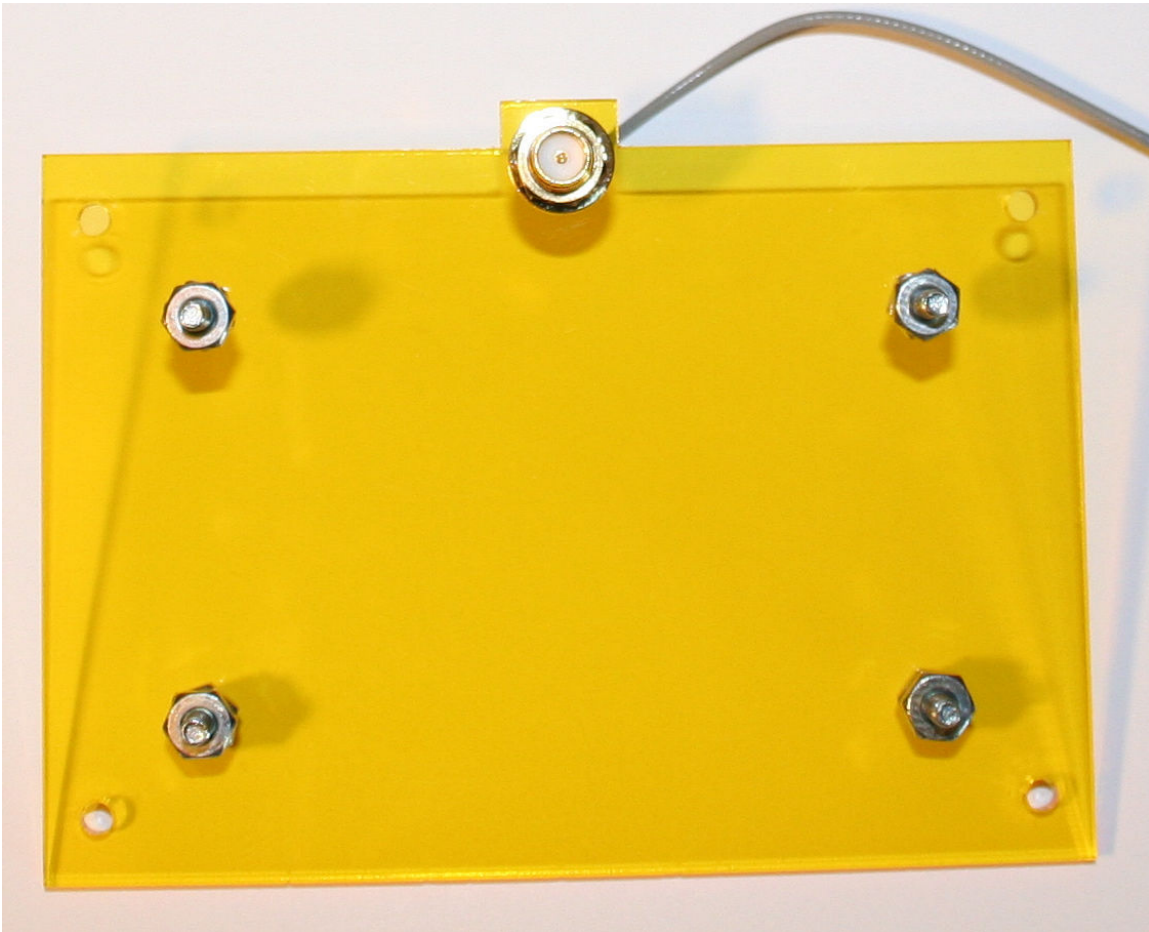
Figure#42a, Attach Dual Lock tape to the base plate.

Attach (2) 1” long pieces of Dual Lock tape to the robot’s base plate. Leave the pieces of tape approximately 1/8”-1/4” in from the back edge of the base plate. Then apply (2) 1” long pieces to the bottom of the battery holder.



Figure#42b, Attach Dual Lock tape to the base of battery pack.

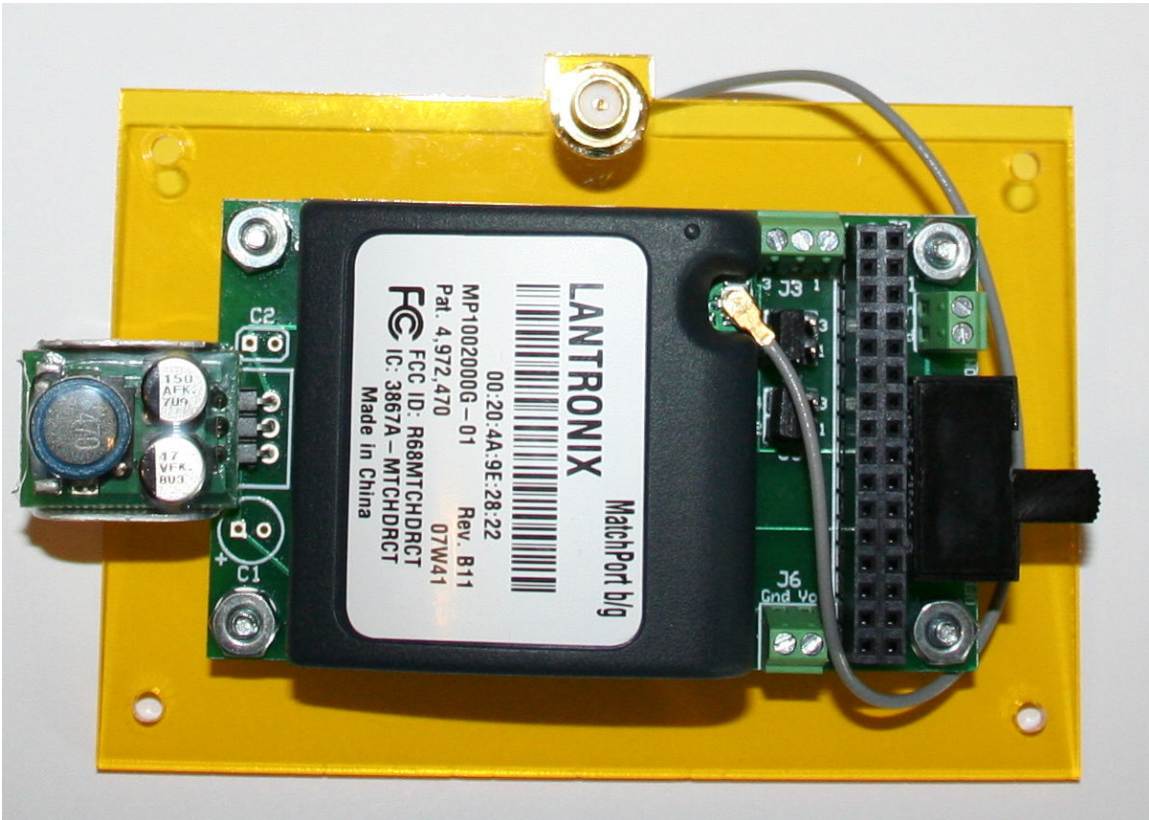
Step#43, Prepare MatchPort/XBee PCB Deck:



Figure#43, Prepare MatchPort/XBee PCB Deck.

In figure#43 prepare the MatchPort/XBee PCB deck by attaching (4) #4-40 machine screws and use (2) #4-40 nuts on each screw so that the MatchPort/XBee PCB has enough clearance. *Also, if you are using the MatchPort b/g module then attach the MatchPort b/g module antenna cable.*

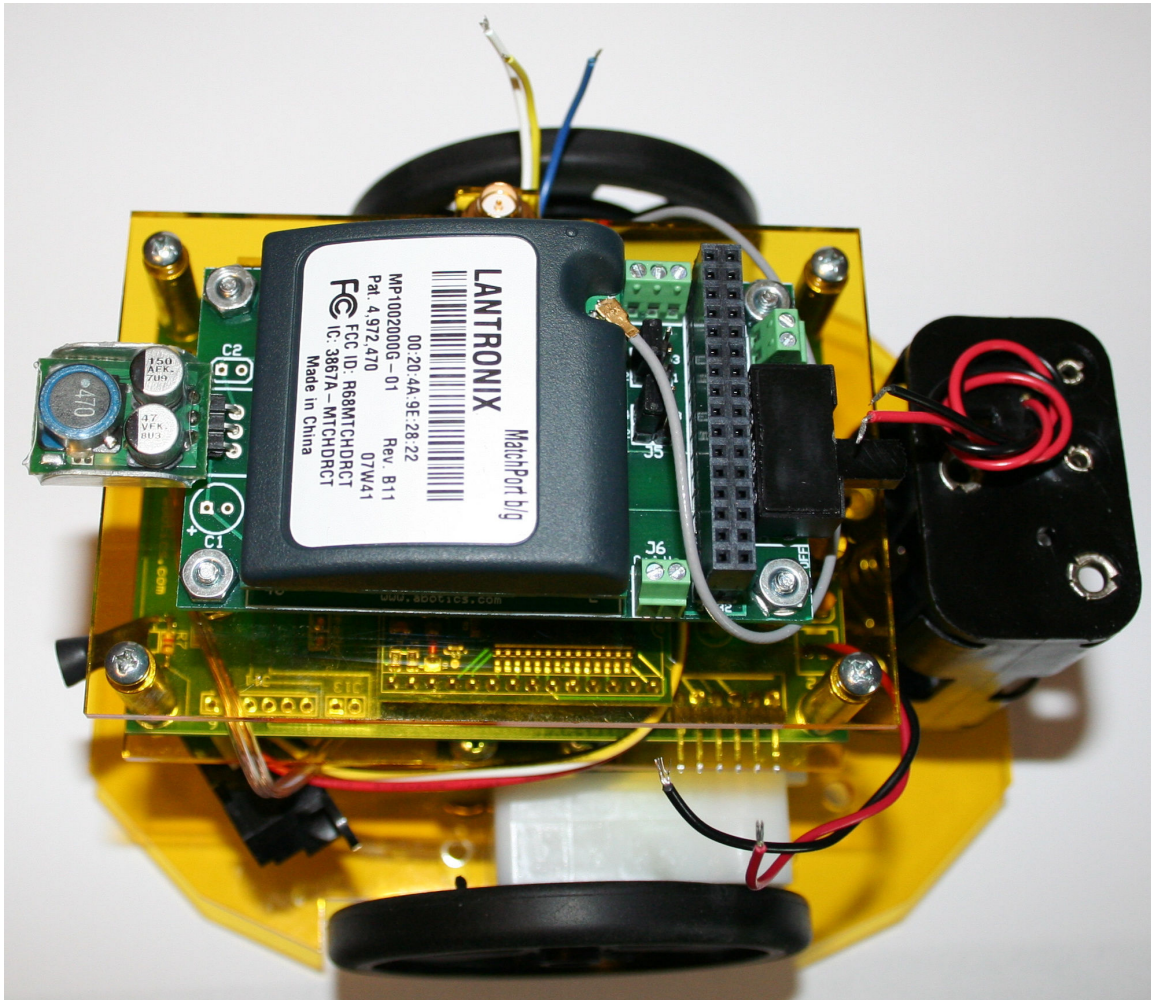
Step#44, Mount the MatchPort/XBee PCB to the Deck:



Figure#44, Mount MatchPort/XBee PCB to Deck.

Mount the MatchPort/XBee PCB to the deck and secure using (4) #4-40 nuts. Connect the antenna cable to the MatchPort b/g module if you are using the MatchPort b/g. We have decided to show you a MatchPort/XBee PCB with a MatchPort b/g module installed to reduce the number images. It should be noted that you might be using the XBee ZigBee module. From this point on we will only show images with the MatchPort b/g module installed. XBEE users can simply disregard the MatchPort b/g specific steps.

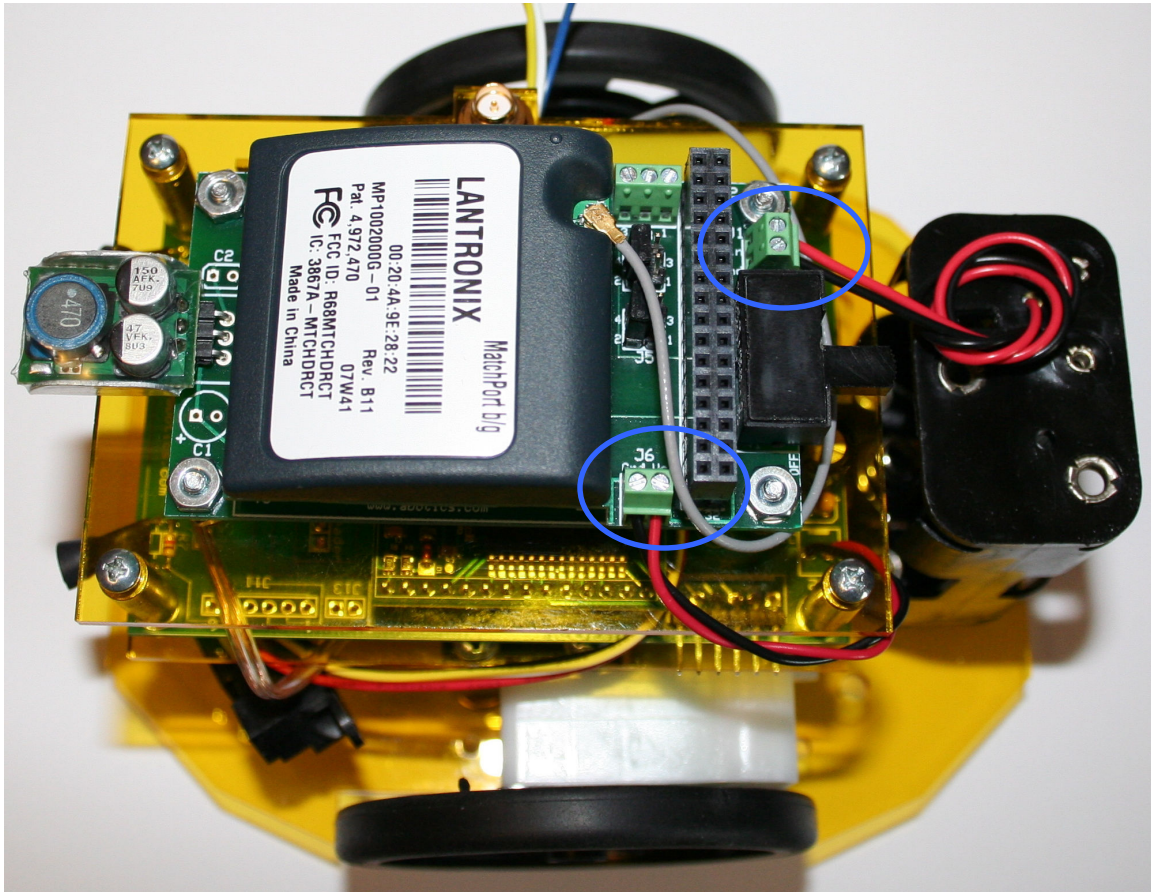
Step#45, Mount MatchPort/XBee Deck to Robot:



Figure#45, Mount MatchPort/XBee Deck to Robot.

Mount the assembled MatchPort/XBee deck to the (4) standoffs and secure with (4) #4-40 machine screws.

Step#46, Connect Power Wires:



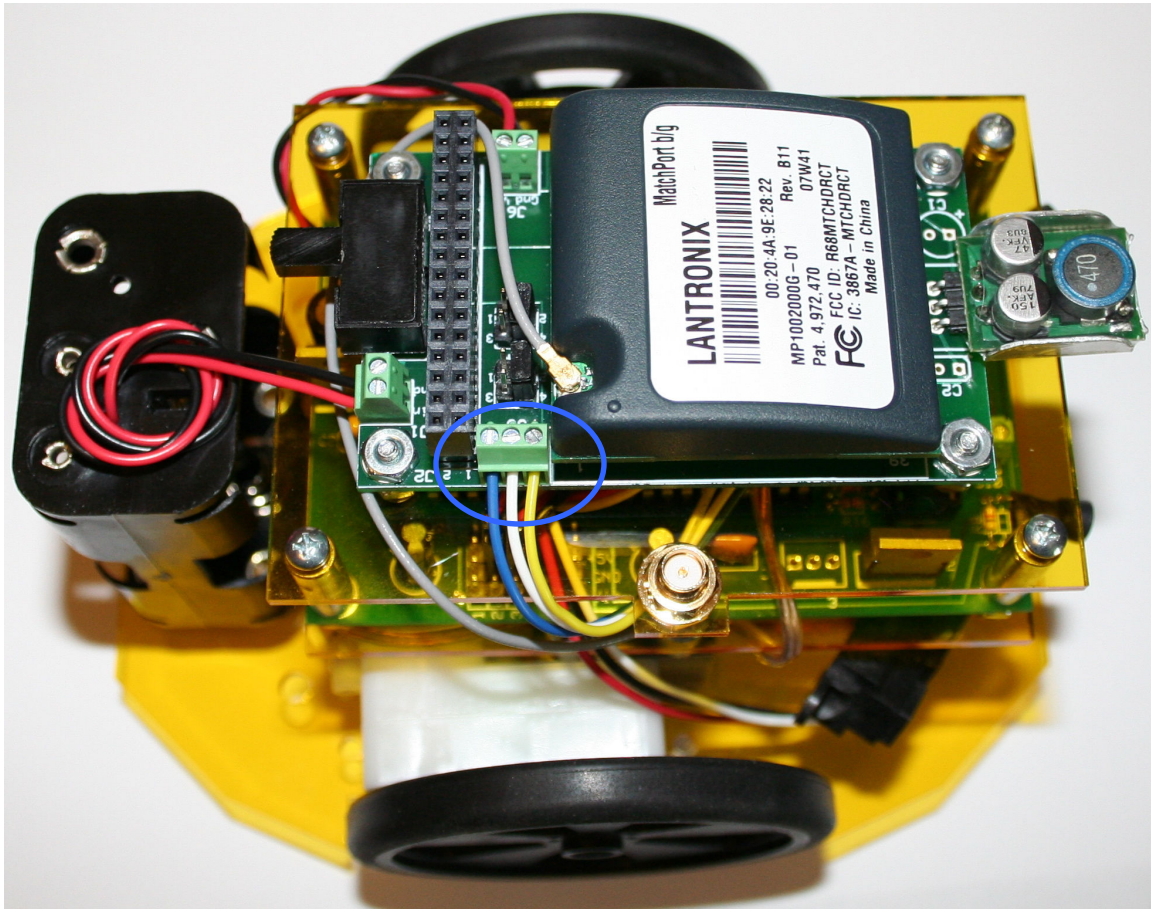
Figure#46, Connect Power Wires.

Connect the (6)-AA battery pack wires to J1 on the MatchPort/XBee PCB. Connect the J1 Controller board power wires to J6 on the MatchPort/XBee PCB. Connect each of the black wires to ground, Gnd. Please look at the connection tables below. Improper connection can result in damage to the Controller Board and/or MatchPort/XBee PCB and/or MatchPort b/g module.

| MatchPort/XBee PCB | Purpose |
|--------------------|--------------------------------|
| J1, Vin pin | Positive Battery Pack Red Wire |
| J1, Gnd pin | Ground Batter Pack Black Wire |

| MatchPort/XBee PCB | OPEN-ROBOT PIC18F4520 Board |
|--------------------|-----------------------------|
| J6, Vo pin | J1, pin#1 Red Wire |
| J6, Gnd pin | J1, pin#2 Black Wire |

Step#47, Connect PIC18F4520 Serial Port Wires:

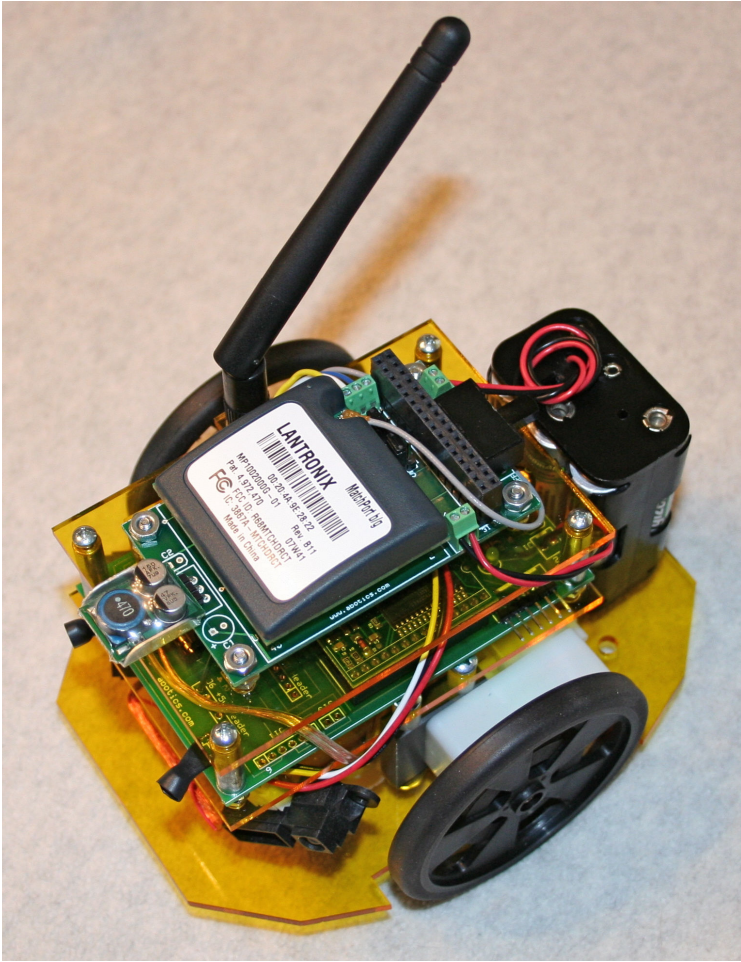


Figure#47, Connect PIC18F4520 Serial Port Wires to MatchPort/XBee PCB.

Connect the controller board PIC18F4520 serial port wires to J3 on the MatchPort/XBee PCB. The blue wire connects to J3 pin#1, the white wire to J3 pin#2, and the yellow wire to J3 pin#3. Please look at the connection table below. Improper connection can result in damage to the Controller Board and/or MatchPort/XBee PCB and/or MatchPort b/g module.

| OPEN-ROBOT PIC18F4520 Board | MatchPort/XBee PCB |
|--------------------------------|--------------------|
| J6, +5 volts header, blue wire | J3, pin#1 |
| J12, pin#3, white wire | J3, pin#2 |
| J12, pin#4, yellow wire | J3, pin#3 |

Step#48, Attach MatchPort b/g Antenna (*purchase separately*):



Figure#48, Attach MatchPort b/g module's rubber antenna.

Finally attach the MatchPort b/g module's rubber antenna. If you are using the XBee module simply disregard this step. Congratulations because you now have a complete OPEN-ROBOT!

If you are using the MatchPort b/g module then move onto Chapter 4 to learn to configure your MatchPort b/g and get started using your robot. However, if you are using XBee modules, then move onto Chapter 5 to get learn how to configure the XBee modules and get started using your robot.

Chapter 5: Get Started with a MatchPort Equipped Robot

This section will help get you started with your MatchPort b/g equipped OPEN-ROBOT. You will be introduced to your robot along with the different sensors and where they are located. *This document assumes that you have already assembled your robot and purchased a MatchPort b/g module, antenna, and UFL to SMA antenna cable.*

Getting to Know Your OPEN-ROBOT & MatchPort b/g:

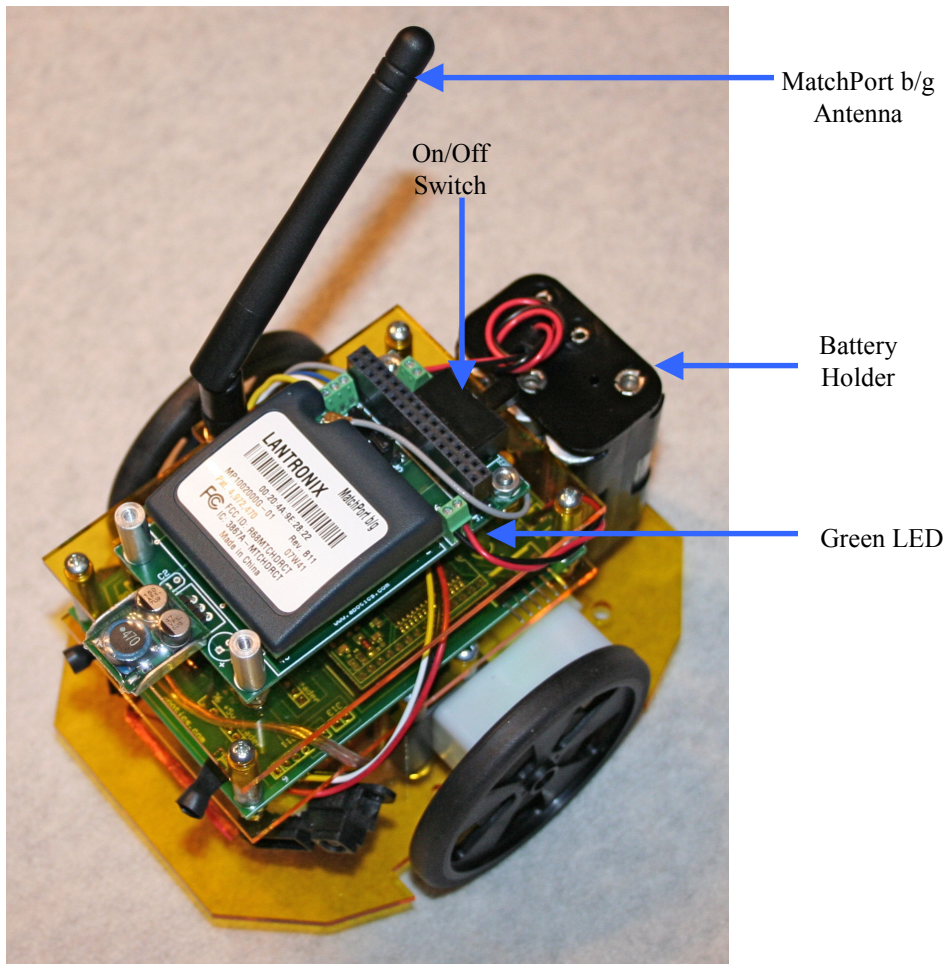


Figure 49. OPEN-ROBOT.

In figure 49 you can see the robot's on/off switch and the (6)-AA battery holder. Before we get started, remove the battery holder by unsnapping it from the robot's base plate. Firmly hold the robot with one hand while pulling the battery holder. Now load (6)-AA batteries into the holder while observing the proper insertion orientation. The negative (-) end of the each battery must be inserted such that it makes contact with the corresponding spring on the holder. ***Failure to do this will damage the robot's controller board.*** We recommend using 15-minute rechargeable AA-batteries, however, you can use regular

alkaline batteries. After inserting the batteries snap the battery holder back onto the robot. You will feel the holder lock into place when you press it onto the robot's base plate. Turn your robot on using the on/off switch. Once on you should observe a green LED on the controller board light up. If not be sure to check the orientation of each of the batteries and also make sure that they are fully charged if using rechargeable batteries.

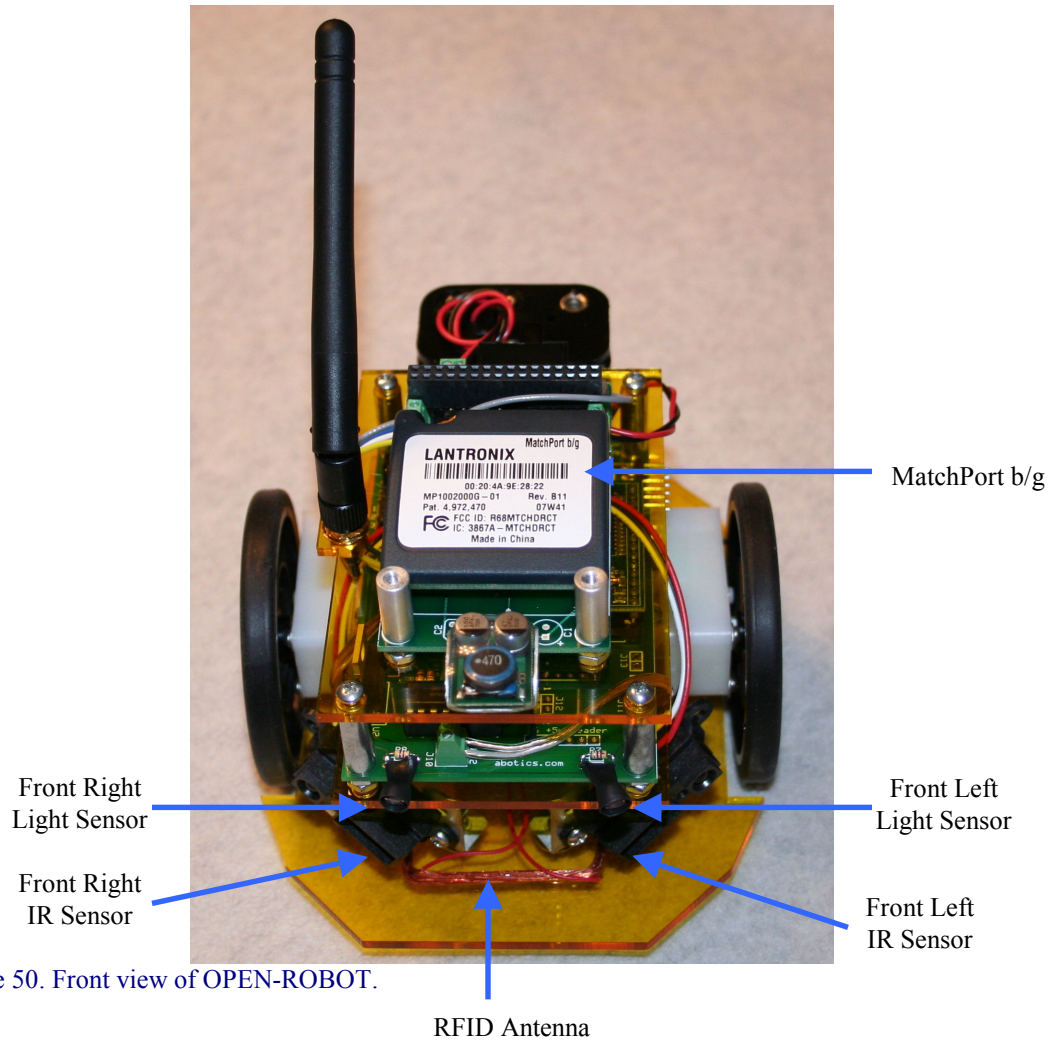


Figure 50. Front view of OPEN-ROBOT.

In figure 50 the front left infrared (IR,) front left light, front right IR, and front right light sensors can be seen. Additionally the RFID antenna is located on the robot's base plate. The IR sensors help the robot to see obstacles, other robots or even to follow walls. Ambient light levels can be sensed using the light sensors, which means the robot can sense whether it is light or dark in a given environment. Passive RFID tags can be read or written to with the RFID antenna. Tags can be used to store x-y coordinate information so that your robot can find its way in a room or small environment. The tags can also store virtual food so that your robot has to search out tags in a game of survival of the fittest. There are numerous other possible uses for the RFID tags and we have only touched on two. The MatchPort b/g allows you to connect and control your robot using wireless 802.11 b/g, WiFi.

Connecting to OPEN-ROBOT for the First Time:

By default the MatchPort b/g module is configured in Ad Hoc mode. This means that you will need to open your Wireless Network Manager and browse for either the [LTRX_IBSS](#) or [OPEN_ROBOT](#) Ad Hoc network. If you purchased your MatchPort from us then it will be configured with the [OPEN_ROBOT](#) Network Name (SSID). Once the [LTRX_IBSS](#) or [OPEN_ROBOT](#) Ad Hoc network is visible, simply connect. Be sure to wait for your computer to acquire an IP address before moving on to the next section.

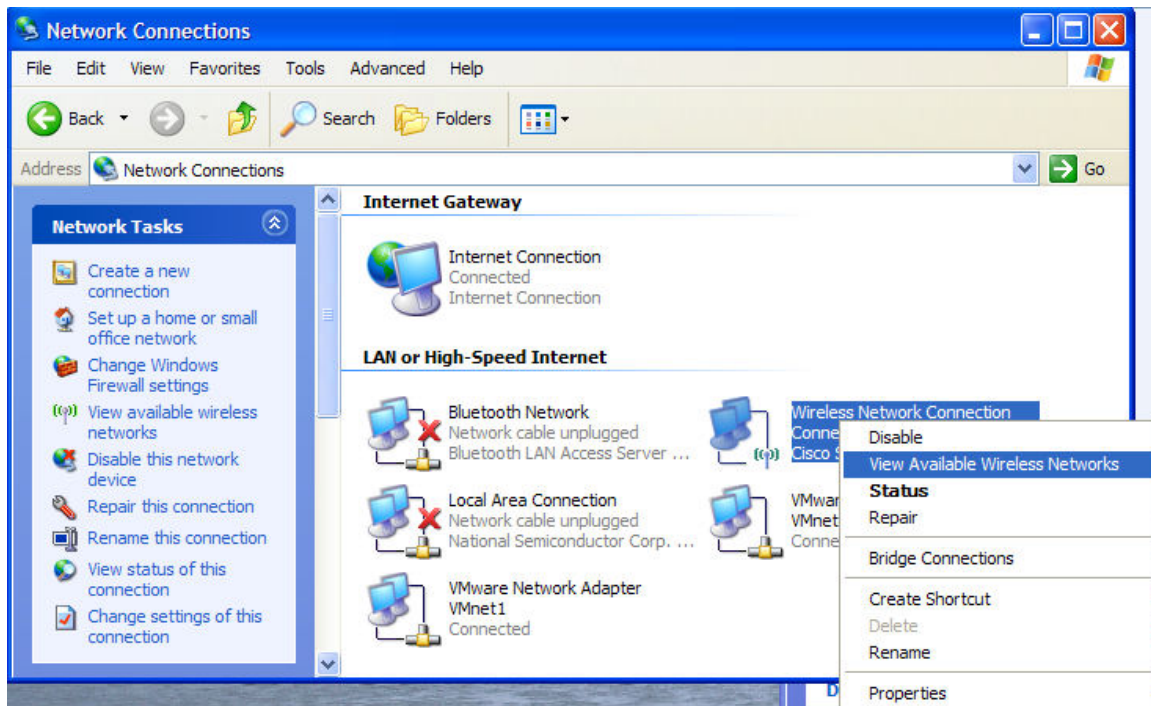


Figure 51. View Available Wireless Networks.

IMPORTANT NOTE - if your host computer is configured to use a static IP address, you need to configure your computer to use DHCP, at least for communication with the Matchport, or you need to assign a static IP address in the 169.254.xxx.xxx subnet. This is because the Matchport acts as a DHCP server for the "LTRX_IBSS" or "OPEN_ROBOT" Ad Hoc network, assigning addresses in the 169.254.xxx.xxx subnet, and if your host computer has a static IP address of 192.168.1.27, it will never be able to communicate with 169.254.xxx.xxx because the network driver won't know how to route packets to a different subnet without explicit routing instructions.

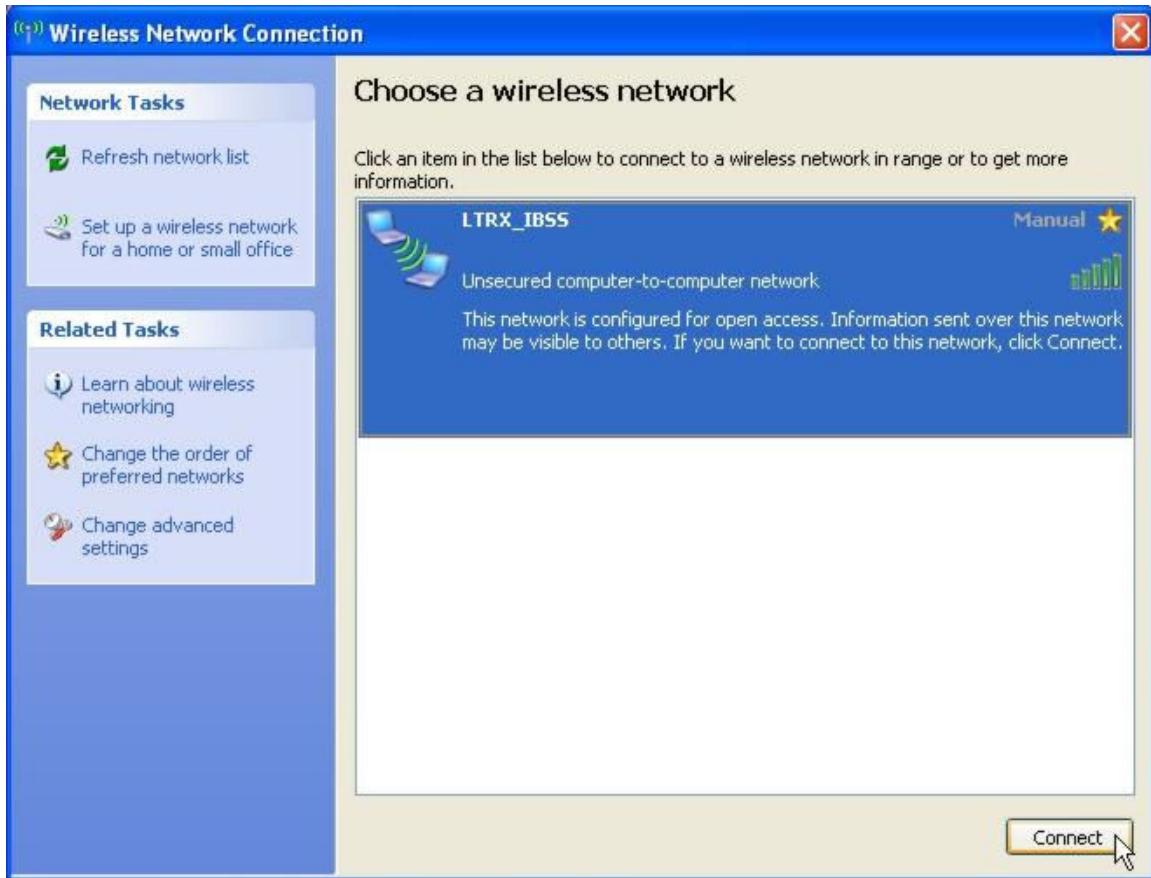


Figure 52. Browse for and connect to LTRX_IBSS Ad Hoc Network.

Click [Refresh network](#) list under Network Tasks. Once the LTRX_IBSS network is available simply click it and then click the [Connect](#) button. It may take a minute or so for your computer to acquire an IP address. Now that we are connected to the LTRX_IBSS network we can connect to the MatchPort b/g and command OPEN-ROBOT.

Connect Using the OPEN-ROBOT TCP-Socket Manual Control Program:

Download and install the [OPEN-ROBOT TCP-Socket Manual Control Program](#). Run the program and enter your MatchPort's IP address and select **Open** from the **Socket** menu. If you purchased your MatchPort b/g module from us it will be configured for **Ad Hoc** mode with an IP Address of [169.254.0.10](#) and the baud rate will be set to **115,200 BPS** so that it matches OPEN-ROBOT's baud rate. If you purchased your MatchPort from some place else, then you will need to scroll down to the next section, [Reconfigure MatchPort Network Settings](#), to set the baud rate to 115200 BPS. After selecting **Open**, the program will attempt to create and open a TCP-Socket connection with the MatchPort. Be sure that the robot is turned on first.

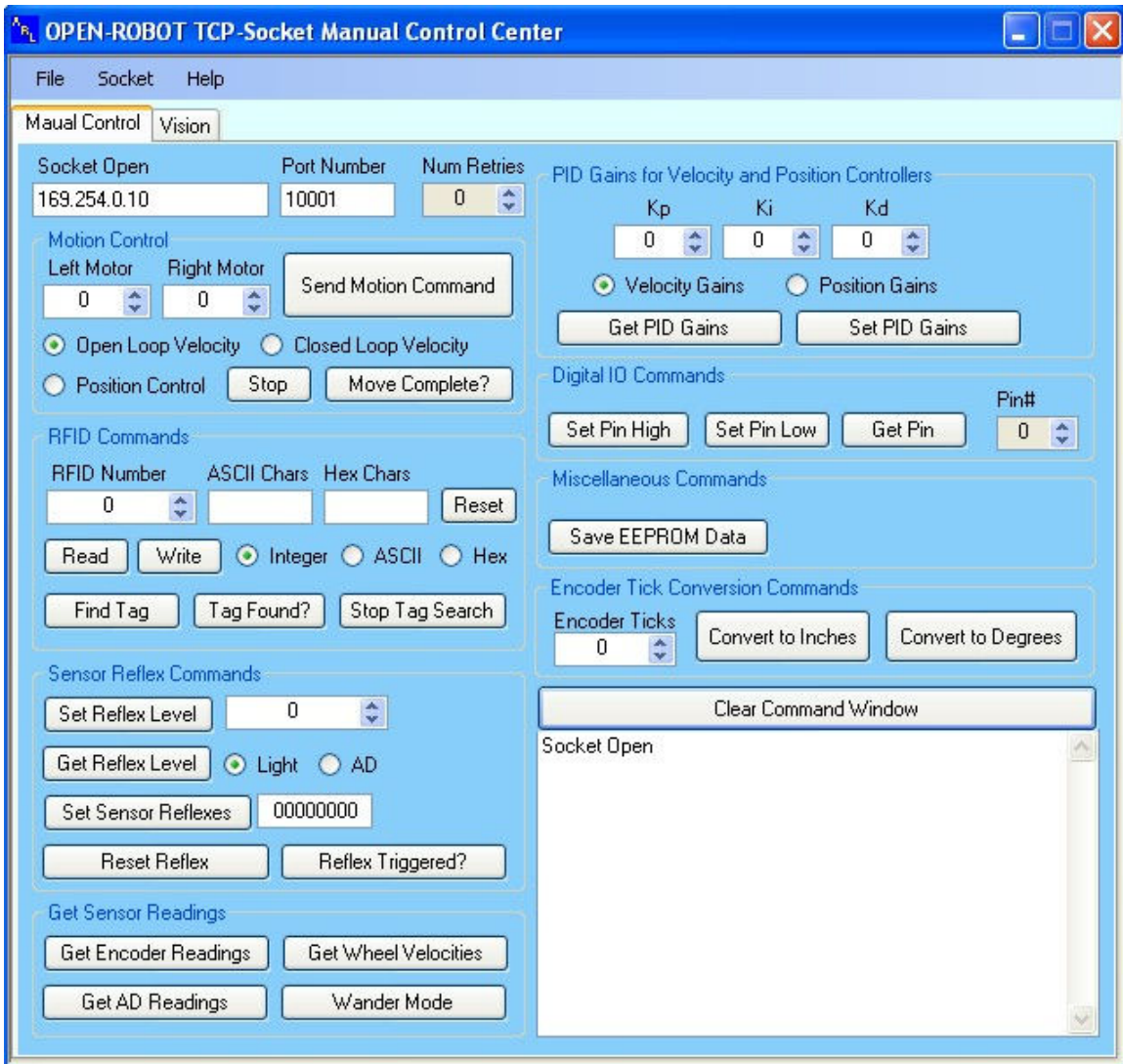


Figure 53. Use Manual Control program to test out OPEN-ROBOT.

Once the socket is open you can command OPEN-ROBOT. Try getting the analog-to-digital sensor readings by clicking on the [Get AD Readings](#) button.

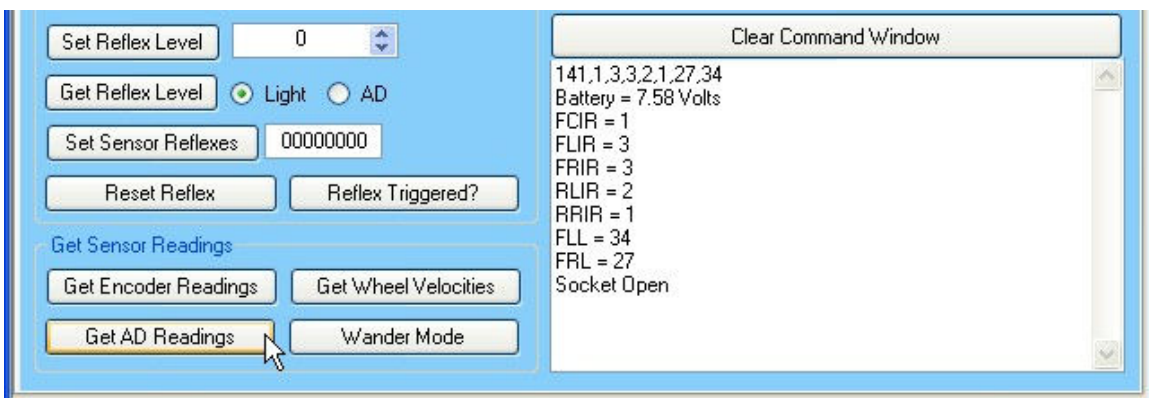


Figure 54. Get AD Sensor Readings.

(OPTIONAL) - Reconfigure MatchPort Network Settings:

If you didn't purchase your MatchPort b/g from Abe Howell's Robotics or simply want to change the Network Settings, then proceed on with this section. By default the MatchPort b/g's ship with a baud rate of 9600 BPS, but OPEN-ROBOT's controller is set to 115,200 BPS, so this needs to be changed. You need to download the [Device Installer](#) utility from Lantronix. Go to the Lantronix downloads web page: <http://www.lantronix.com/support/downloads.html> and click on [Device Installer](#) under the [Configuration Utility](#) column. Download and install the Device Installer.

Connect to the LTRX_IBSS Ad Hoc network using the method described above in "[Connecting to OPEN-ROBOT for the First Time](#)". Then run the Lantronix Device Installer utility. A pop-up window will display and ask if you would like a TCP/IP tutorial. Simply disregard this and click **NO**.

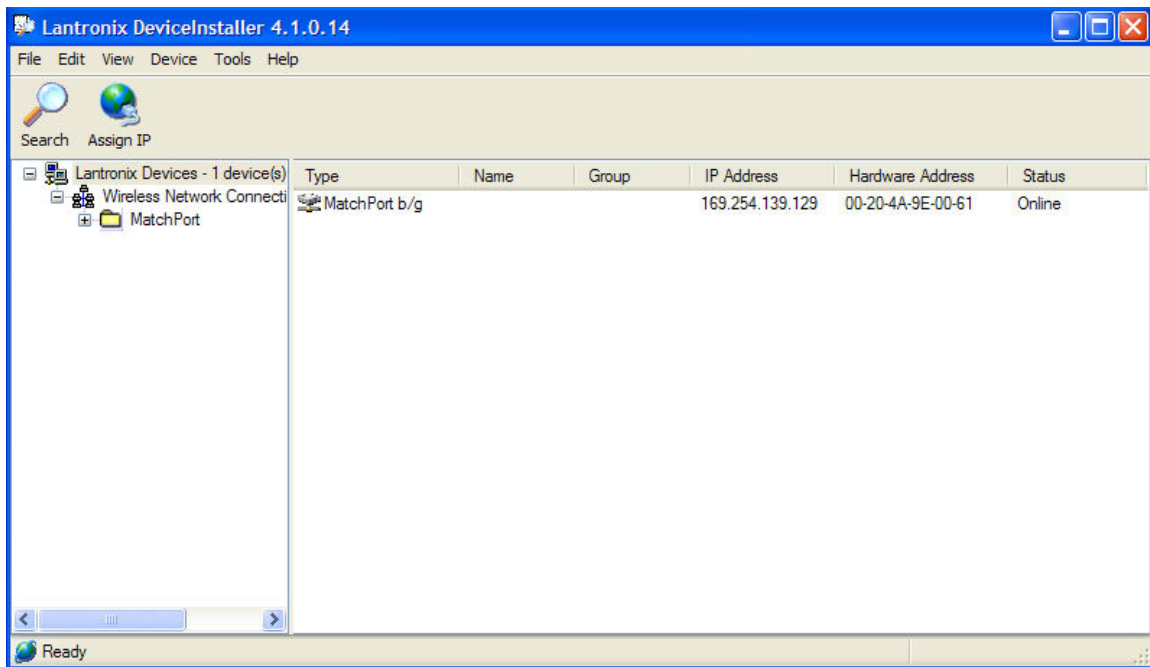


Figure 55. Device Installer Utility.

Device Installer should recognize OPEN-ROBOT's MatchPort b/g module and then populate the list box as shown above in figure#55. Double-click the MatchPort b/g listing in the list box to bring up the device details.

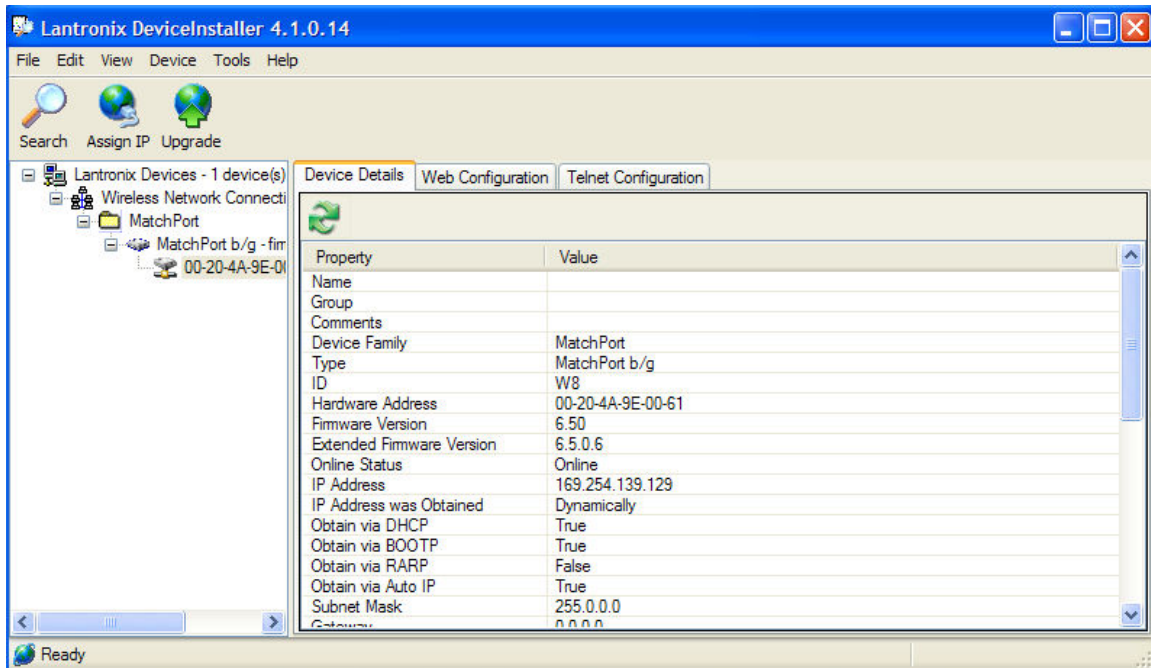


Figure 56. Device Details.

Now click on the Web Configuration tab.

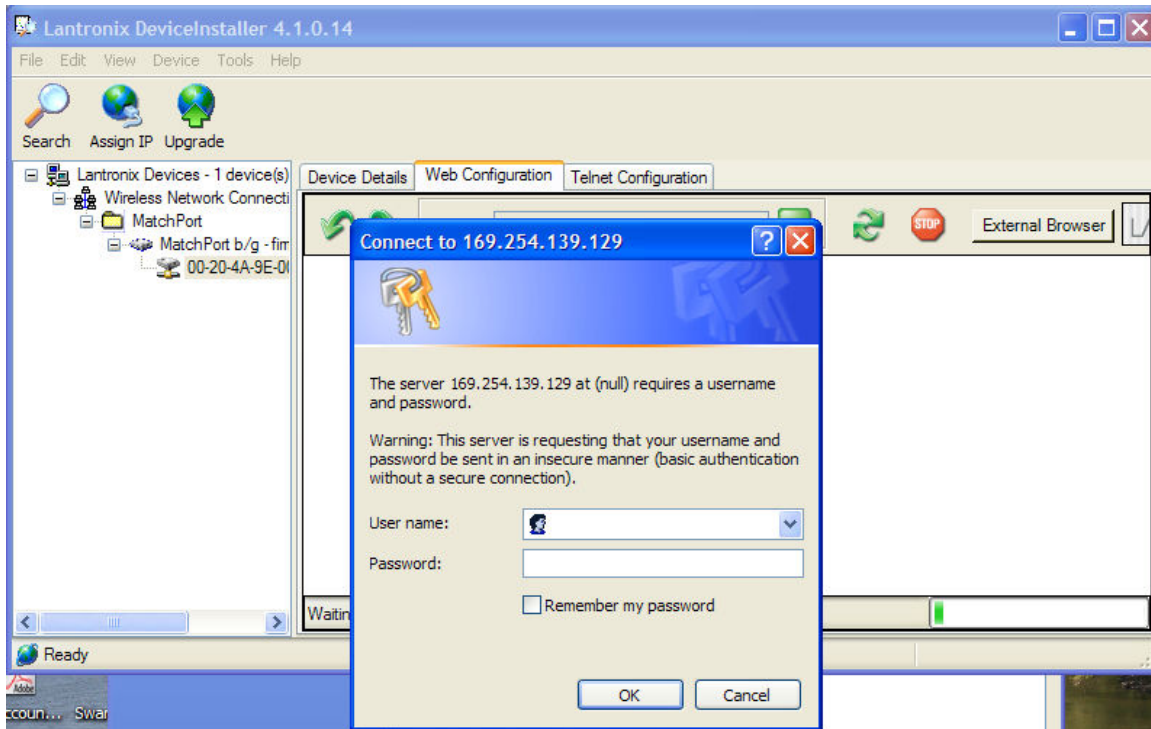


Figure 57. Connect via Web Configuration.

A password message box will appear. Simply click on the **OK** button and don't enter anything in the **User name** and **Password** fields because by default the MatchPort does not utilize a username/password.

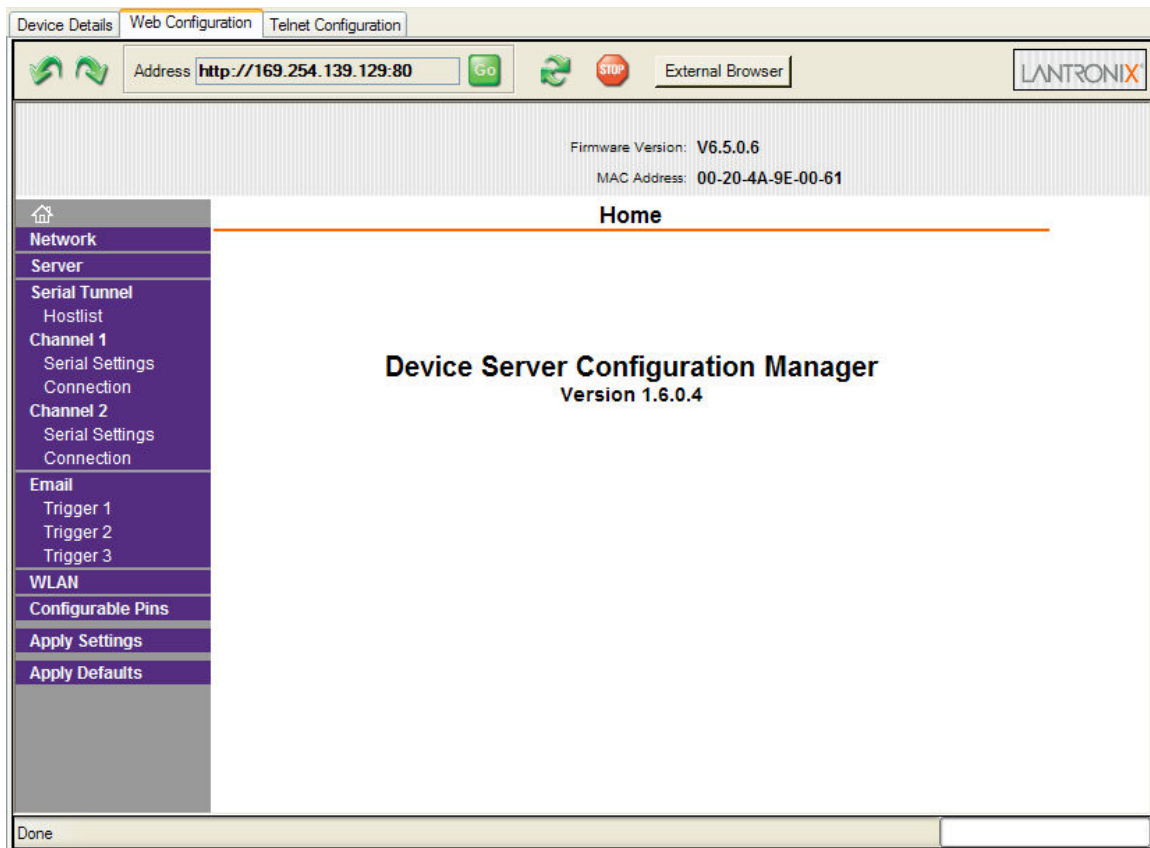


Figure 58. Web Configuration Home.

Click on the [WLAN](#) link so that we can change these settings. By default the MatchPort is configured for Ad Hoc mode and the Network Name (SSID) is set to [LTRX_IBSS](#). You can change from Ad Hoc to [Infrastructure](#) under Network Type and also change the Network Name (SSID) to match your own wireless network name. For this example I will be using [abes_wireless](#) for my Network name as shown below in figure#59. Also be sure to [disable Radio Power Management](#) under the Advanced Settings section. Disabling Radio Power Management will make the MatchPort more responsive during the initial connection upon power-up and will prevent it from going into Radio Power Management during robot operation. Once you have made these changes be sure to click on the [OK](#) button. ***NOTE: Be sure to write down all WLAN Settings or take a screenshot.***

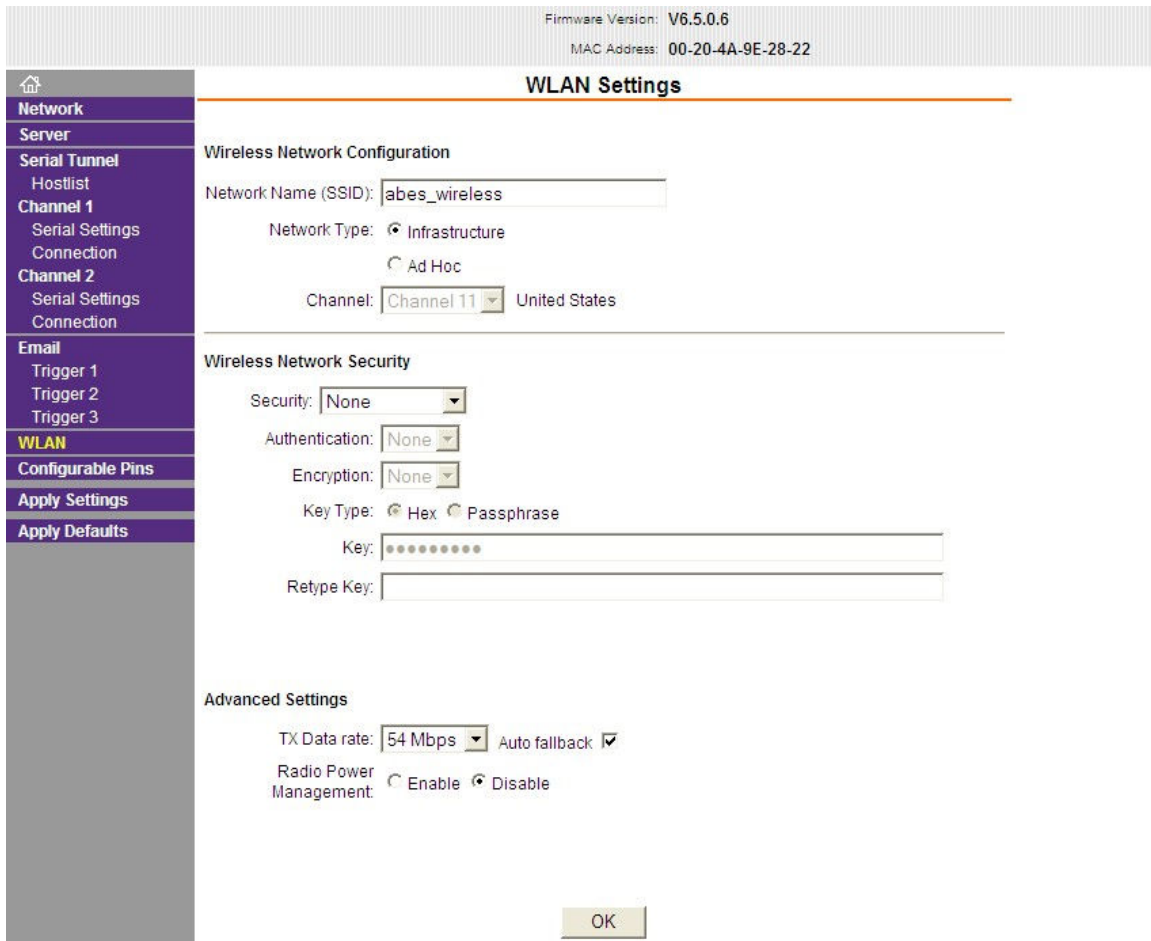


Figure 59. New WLAN Settings.

Now we need to change the [Network settings](#), so click on the [Network](#) link, which is right below the [Home](#) icon. Under the IP Configuration we need to select [Use the Following IP Configuration](#) and then enter a suitable IP Address for the MatchPort, a Subnet Mask, and a Default Gateway. The settings shown in figure#60 should work for most wireless setups, but you must be sure to check your own setup and verify that the IP Address you enter is an available address on your network otherwise you will be unable to connect to the MatchPort. The Subnet Mask and Default Gateway must match your network setup as well. Once the new information is entered be sure to click on the [OK](#) button. If you do enter and apply the wrong settings, you will need to perform a recovery process. Surveyor Corp does have a suitable recovery process documented in the link below. You can also contact Abe Howell's Robotics for assistance if you purchased the MatchPort from us. ***NOTE: Be sure to write down all Network Settings or take a screenshot.***

<http://www.surveyor.com/cgi-bin/yabb2/YaBB.pl?num=1200515001>

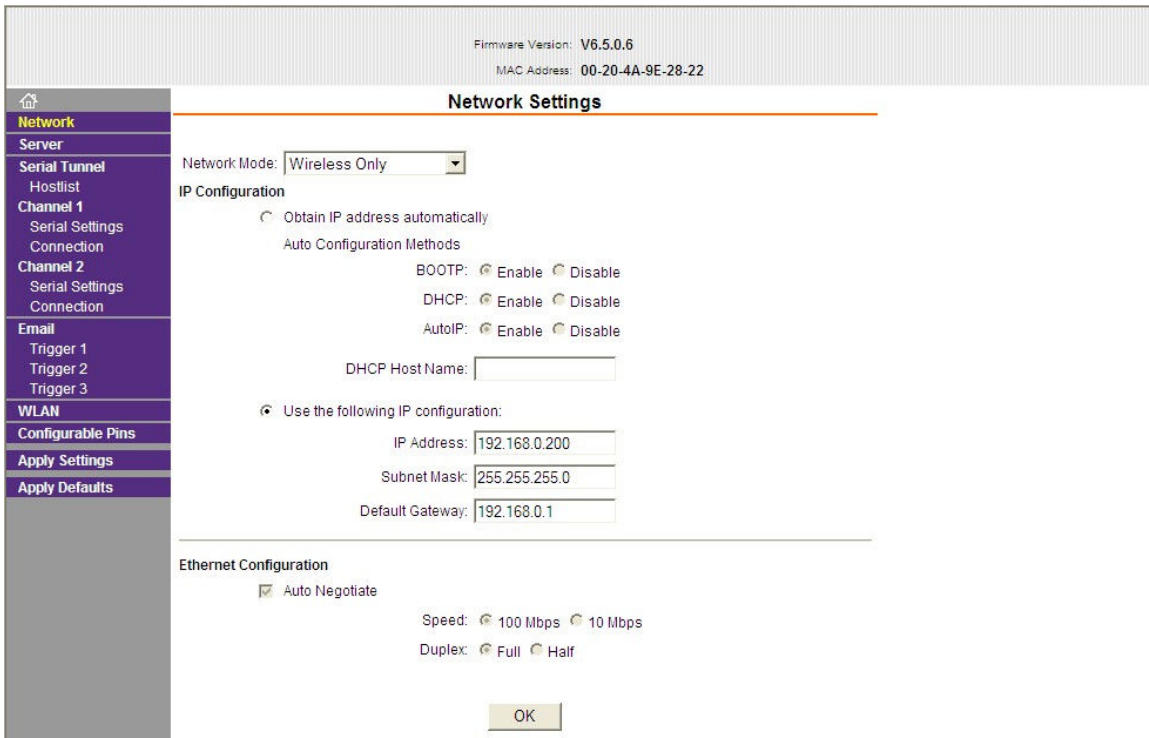


Figure 60. New Network Settings.

You should also change the **CPU Performance Mode** to **High**. Click **OK** when done.

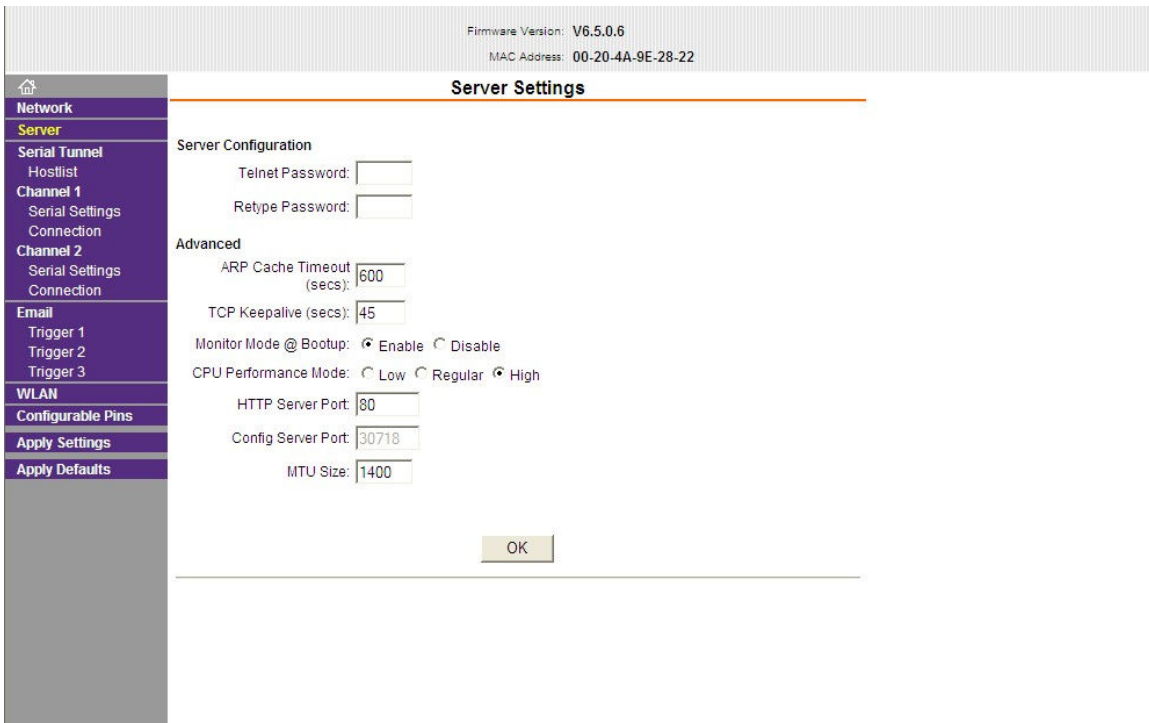


Figure 61. Set CPU Performance Mode to High.

If you purchased your OPEN-ROBOT fully assembled then the serial port settings should already be set. However, if you need to set them please enter as shown in figure#62 below. You should only have to change the Baud Rate to **115200**, Flow Control to **None**, Data bits = **8**, Parity = **None**, and Stop Bits = **1**. Click **OK** when done.

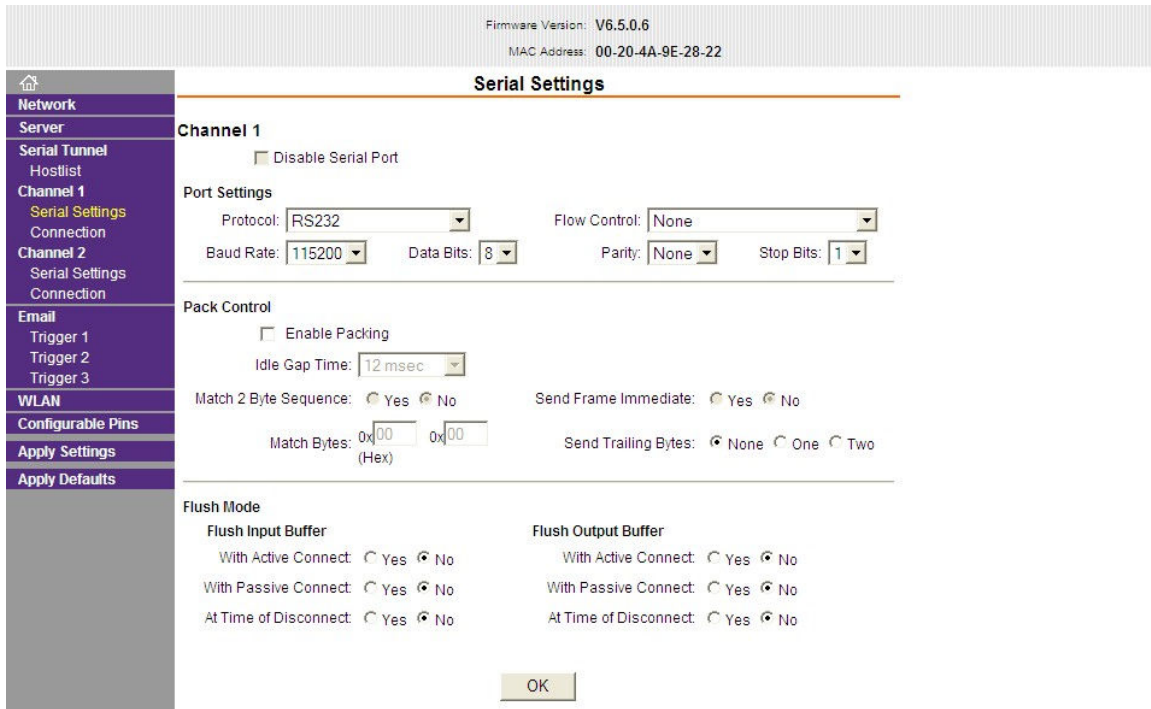


Figure 62. Channel 1 Serial Settings.

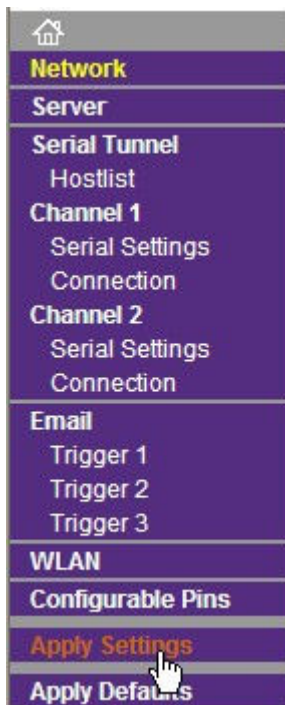


Figure 63. Apply Settings

None of the settings you just changed are permanent until you click on the [Apply Settings](#) link as shown above in figure#63. Once you click on [Apply Settings](#), the MatchPort will save the new values and then reboot so that they take effect. The MatchPort reboot may take a minute or two. ***NOTE: Before applying the new settings, be sure to write them down or take screenshots. This information will be important in case for example you forget the assigned IP Address or Network Name (SSID).***

Chapter 6: Get Started with XBee® Equipped Robot

This chapter will help get you started with your XBee equipped OPEN-ROBOT. You will be introduced to your robot along with the different sensors and where they are located. *This document assumes that you have already assembled your robot and purchased the necessary XBee modules and USB Explorer dongle.*

Getting to Know Your OPEN-ROBOT & XBee modules:

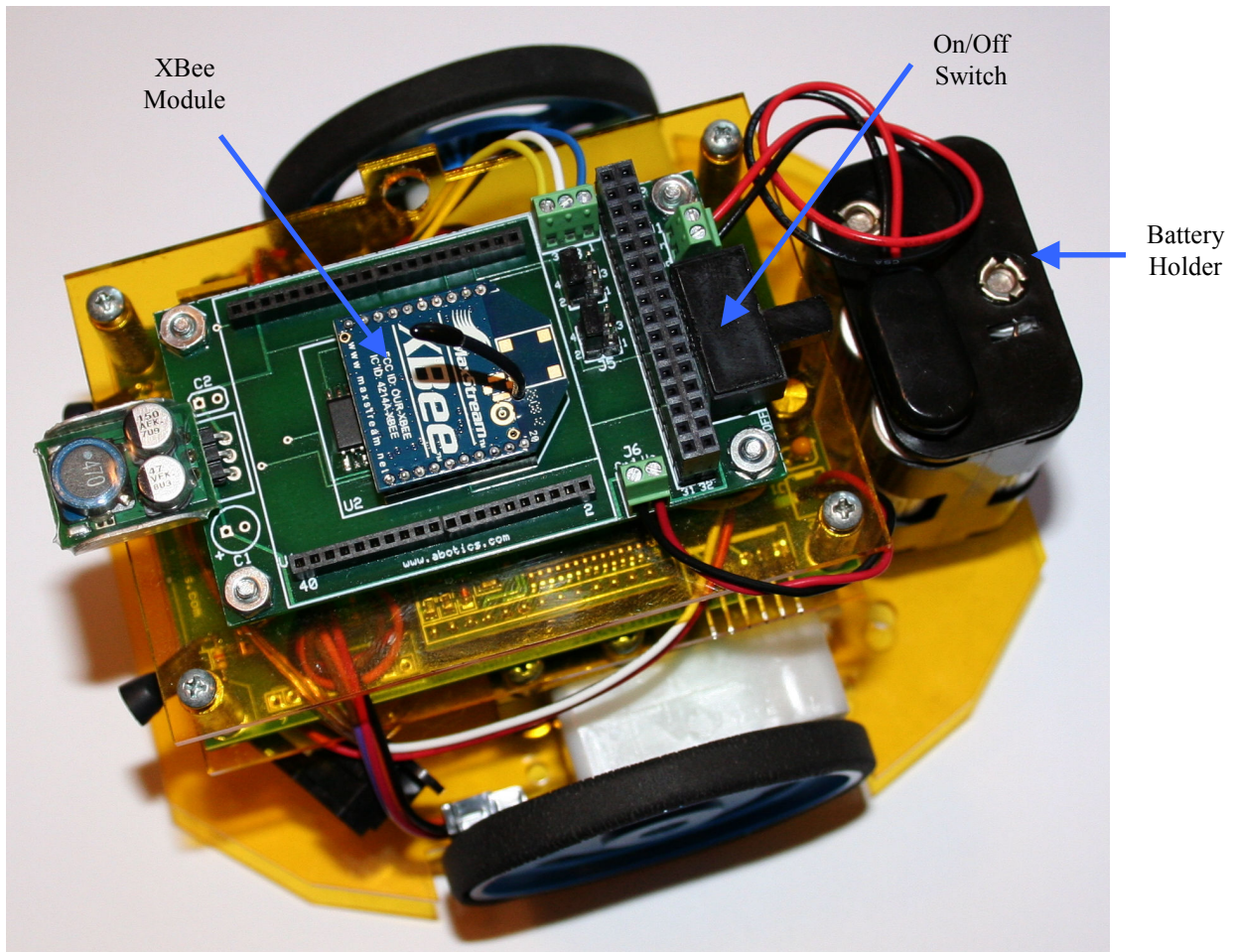


Figure 64. OPEN-ROBOT shown with XBee wireless module.

In figure 64 you can see the robot's on/off switch and the (6)-AA battery holder. Before we get started, remove the battery holder by unsnapping it from the robot's base plate. Firmly hold the robot with one hand while pulling the battery holder. Now load (6)-AA batteries into the holder while observing the proper insertion orientation. The negative (-) end of the each battery must be inserted such that it makes contact with the corresponding spring on the holder. ***Failure to do this will damage the robot's controller board.*** We recommend using 15-minute rechargeable AA-batteries, however, you can use regular

alkaline batteries. After inserting the batteries snap the battery holder back onto the robot. You will feel the holder lock into place when you press it onto the robot's base plate. Turn your robot on using the on/off switch. Once on you should observe a green LED on the controller board light up. If not be sure to check the orientation of each of the batteries and also make sure that they are fully charged if using rechargeable batteries.

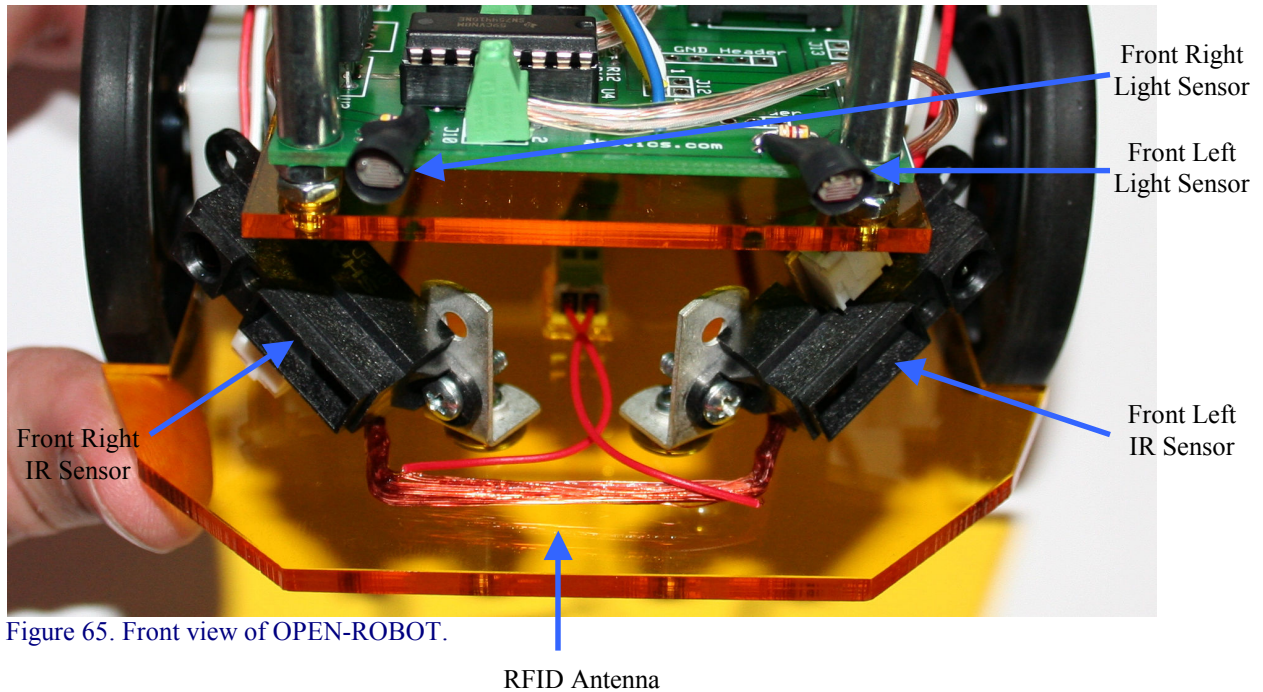


Figure 65. Front view of OPEN-ROBOT.

In figure 2 the front left infrared (IR,) front left light, front right IR, and front right light sensors can be seen. Additionally the RFID antenna is located on the robot's base plate. The IR sensors help the robot to see obstacles, other robots or even to follow walls. Ambient light levels can be sensed using the light sensors, which means the robot can sense whether it is light or dark in a given environment. Passive RFID tags can be read or written to with the RFID antenna. Tags can be used to store x-y coordinate information so that your robot can find its way in a room or small environment. The tags can also store virtual food so that your robot has to search out tags in a game of survival of the fittest. There are numerous other possible uses for the RFID tags and we have only touched on two.

Configure XBee Modules:

OPEN-ROBOT's PIC18F4520 communicates with a baud rate of 115,200 bits per second (BPS). However, the XBee modules are by default configured for 9,600 BPS, so we need to change these to 115,200 BPS so that they will work with OPEN-ROBOT. To do this you will need an XBee Explorer USB from SparkFun Electronics. You also need to download and install the necessary FTDI drivers:

http://www.sparkfun.com/commerce/product_info.php?products_id=8687

<http://www.ftdichip.com/FTDrivers.htm>

Once you have installed the FTDI drivers you then need to download and install the XBee X-CTU program:

<http://www.digi.com/support/kbase/kbasesresultdetl.jsp?kb=125>

The X-CTU program allows us to change the XBee's configuration parameters. Once you have the X-CTU installed, plug an XBee Explorer USB into your computer. Be sure you install one of your XBee modules into the Explorer USB prior to connecting to your computer. Go ahead and run the X-CTU program. You should have a screen similar to the one shown below.

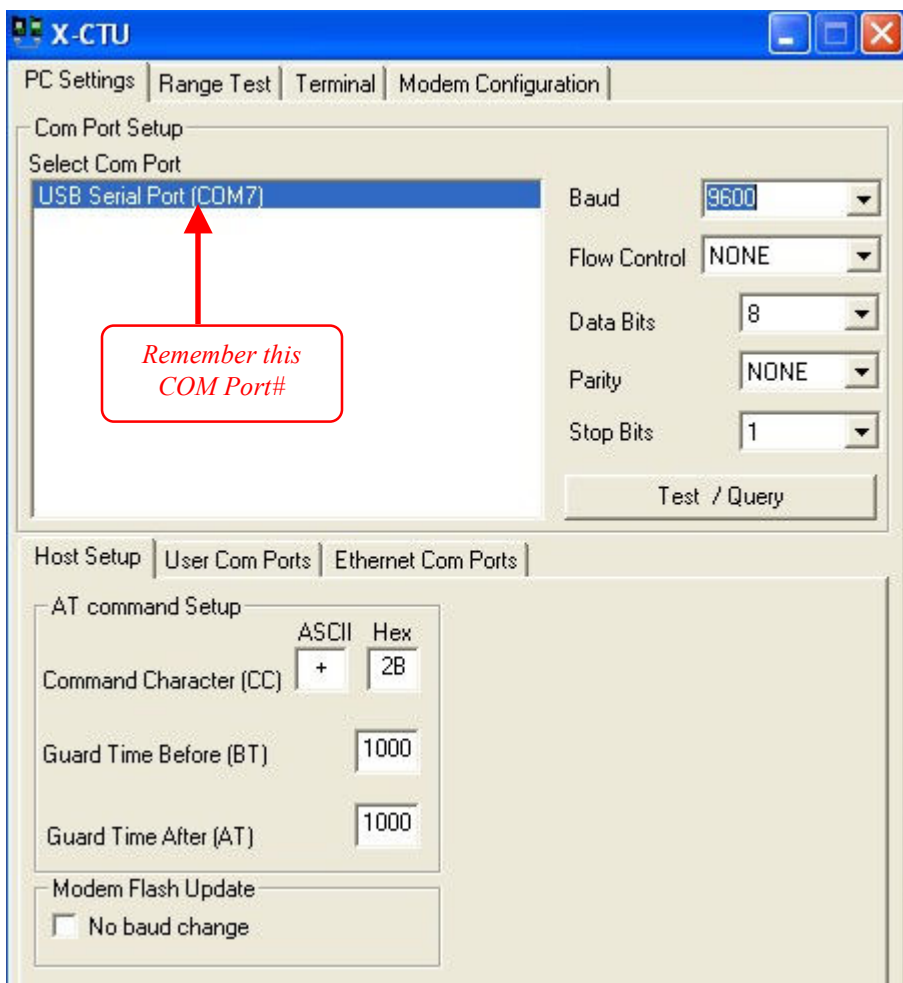


Figure 66. Open X-CTU program.

Be sure to select a Baud Rate if 9600 and then click on the [Modem Configuration](#) tab.

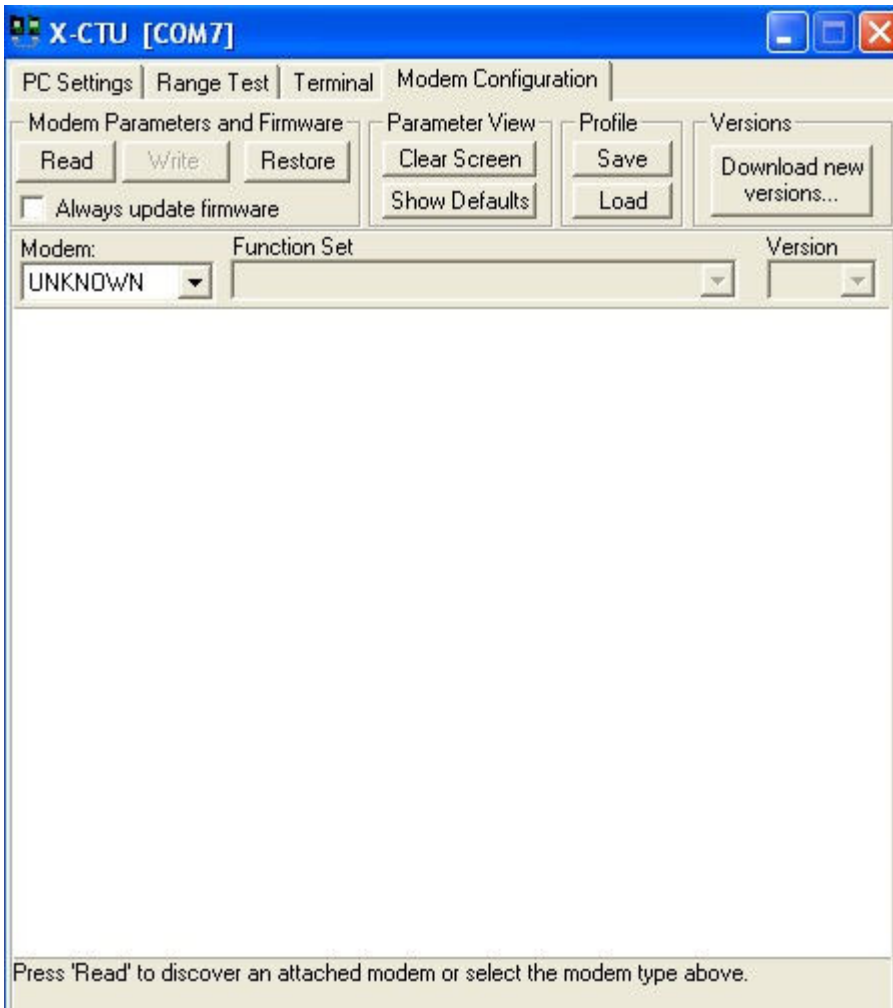


Figure 67. Configuration Tab.

Click on the Read button to read the current configuration of the attached XBee module. Then scroll down to the [Serial Interfacing](#) folder and click on the [Interface Data Rate](#).

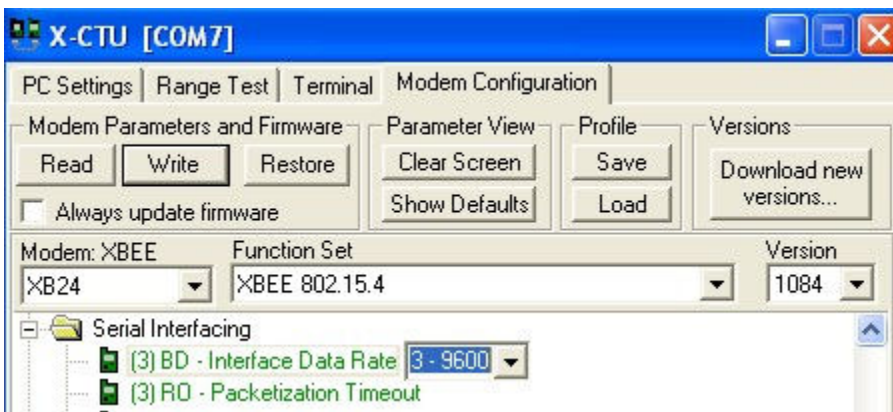


Figure 68. Scroll to Serial Interfacing.

Now select 115200 under the Interface Data Rate parameter.

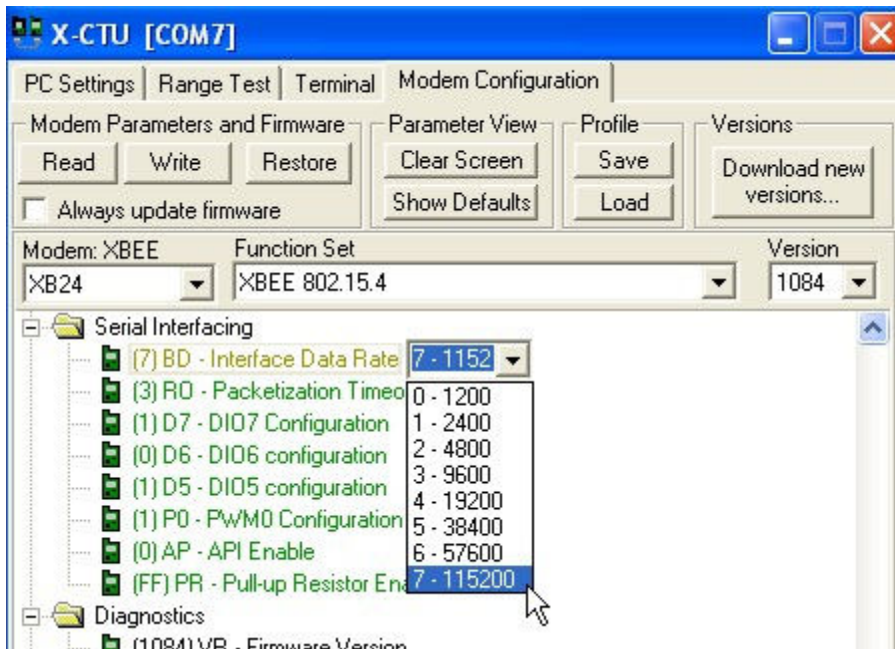


Figure 69. Select 115200 under Interface Data Rate.

Finally click on the Write button to save this parameter change to the XBee module.

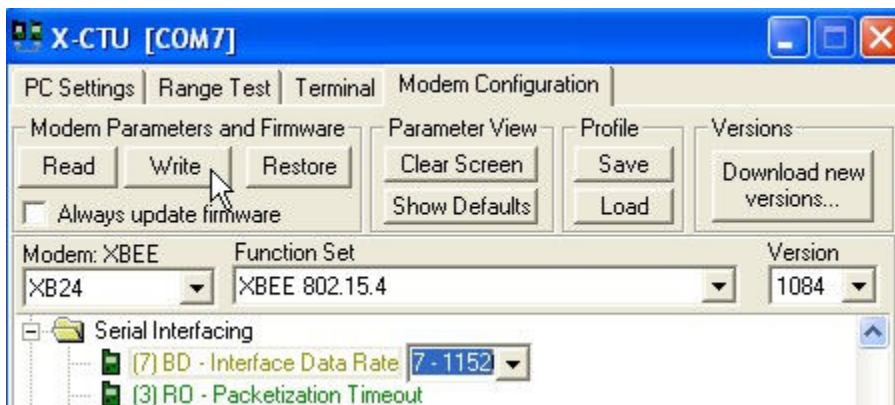


Figure 70. Click on Write button.

After clicking the Write button you should see the following printouts at the bottom of the screen as shown in figure#8 below.

“Setting AT Parameters..OK”
 “Write Parameters...Complete”

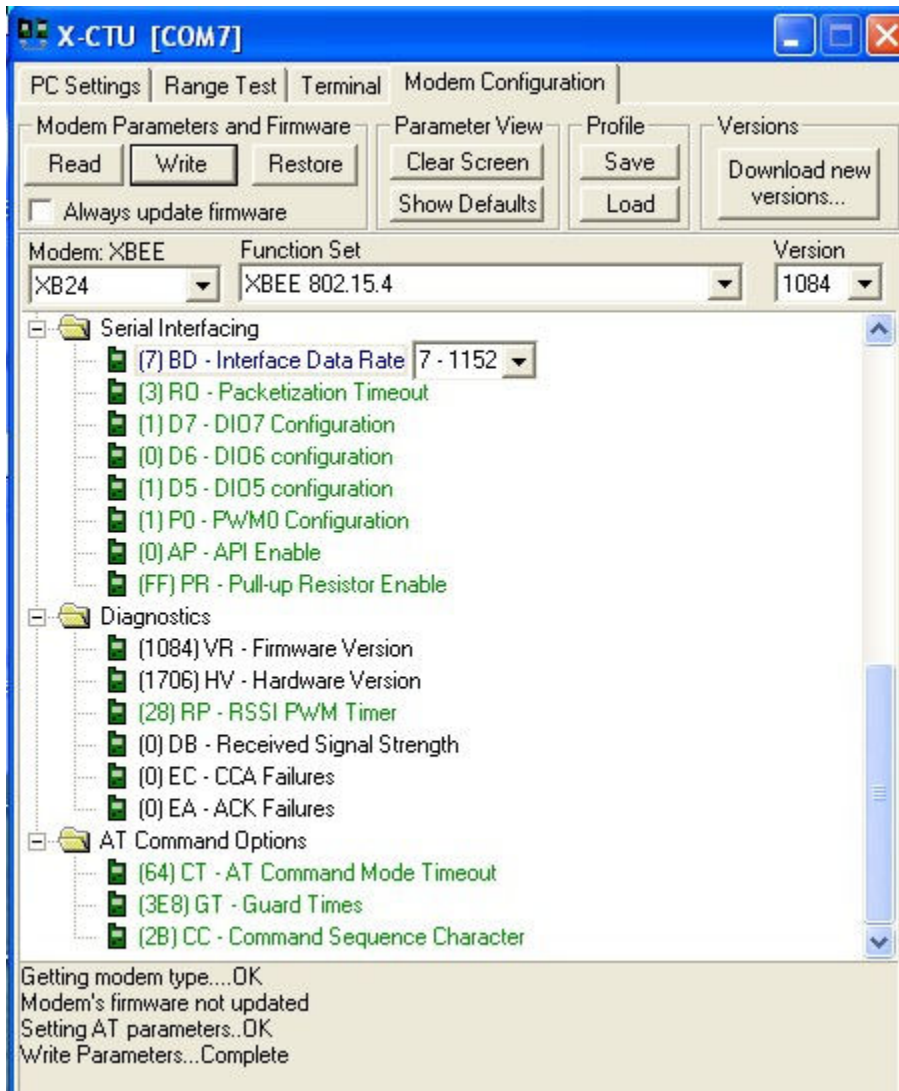


Figure 71. Interface Data Rate parameter change written to XBee module.

Connect Using the OPEN-ROBOT XBee/ZigBee Manual Control Program:

Download and install the [OPEN-ROBOT XBee/ZigBee Manual Control Program](http://www.abotics.com/OPEN_ROBOT_Software/OPEN_ROBOT_ZigBee_Installer.zip).

http://www.abotics.com/OPEN_ROBOT_Software/OPEN_ROBOT_ZigBee_Installer.zip

Run the installer and once the installation is complete go ahead and run the program. You should see a screen similar to the one shown below.

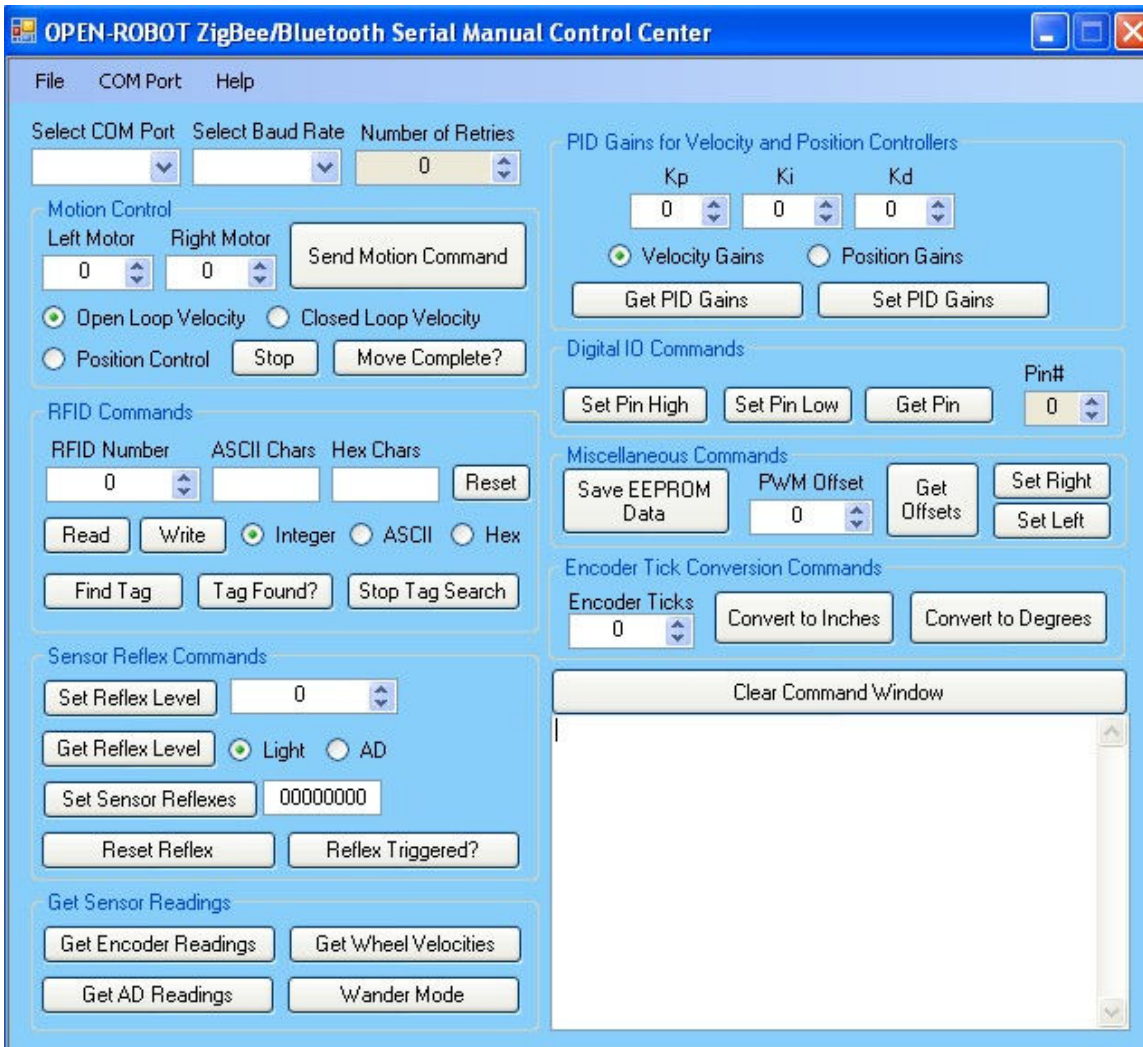


Figure 72. Use Manual Control program to test out OPEN-ROBOT.

Select the appropriate COM Port, set the baud rate to 115200 and set the number of retries to 1. Use the COM Port that was displayed when you ran the X-CTU program. For this example it happens to be COM7, but it may be different for you.

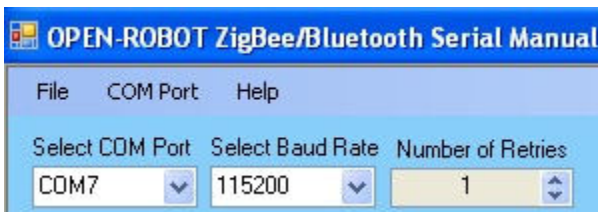


Figure 73. Configure COM Port.

From the **COM Port** menu select **Open**.



Figure 74. Open Serial Port.

Once the serial port is open you can command OPEN-ROBOT. Try getting the analog-to-digital sensor readings by clicking on the [Get AD Readings](#) button.

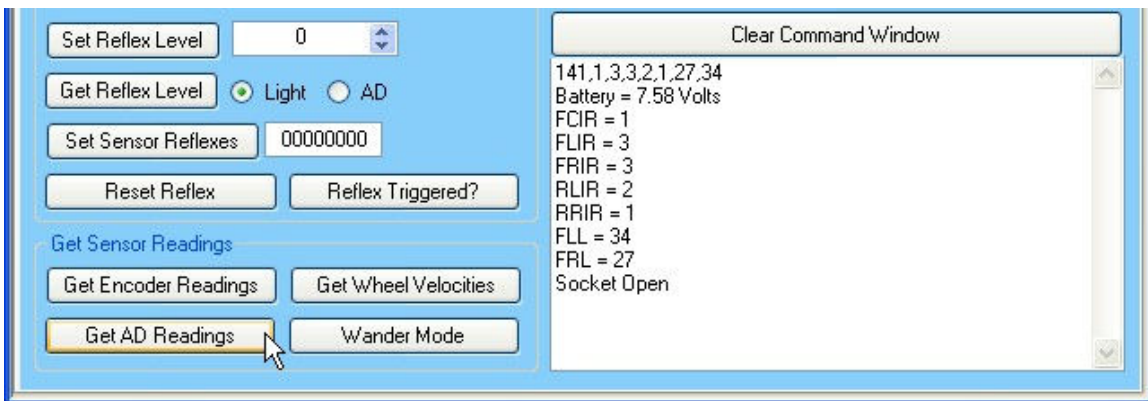


Figure 75. Get AD Sensor Readings.

Go ahead and try the other functions. If you are using your XBee equipped robot in a WiFi environment you may notice that OPEN-ROBOT fails to respond and you see a `READ_ERROR`. This is normal and happens because the XBee wireless packets sometimes get dropped. To counteract this we recommend setting the Number of Retries to at least 1.

Chapter 7: Wirelessly upload PIC18F4520 code

If you are using a MatchPort b/g then run the OPEN-ROBOT TCP Manual Program. Otherwise if you are using the XBee modules then run the OPEN-ROBOT Serial Manual Control Program. For illustration purposes I will be showing the OPEN-ROBOT TCP Manual Program.

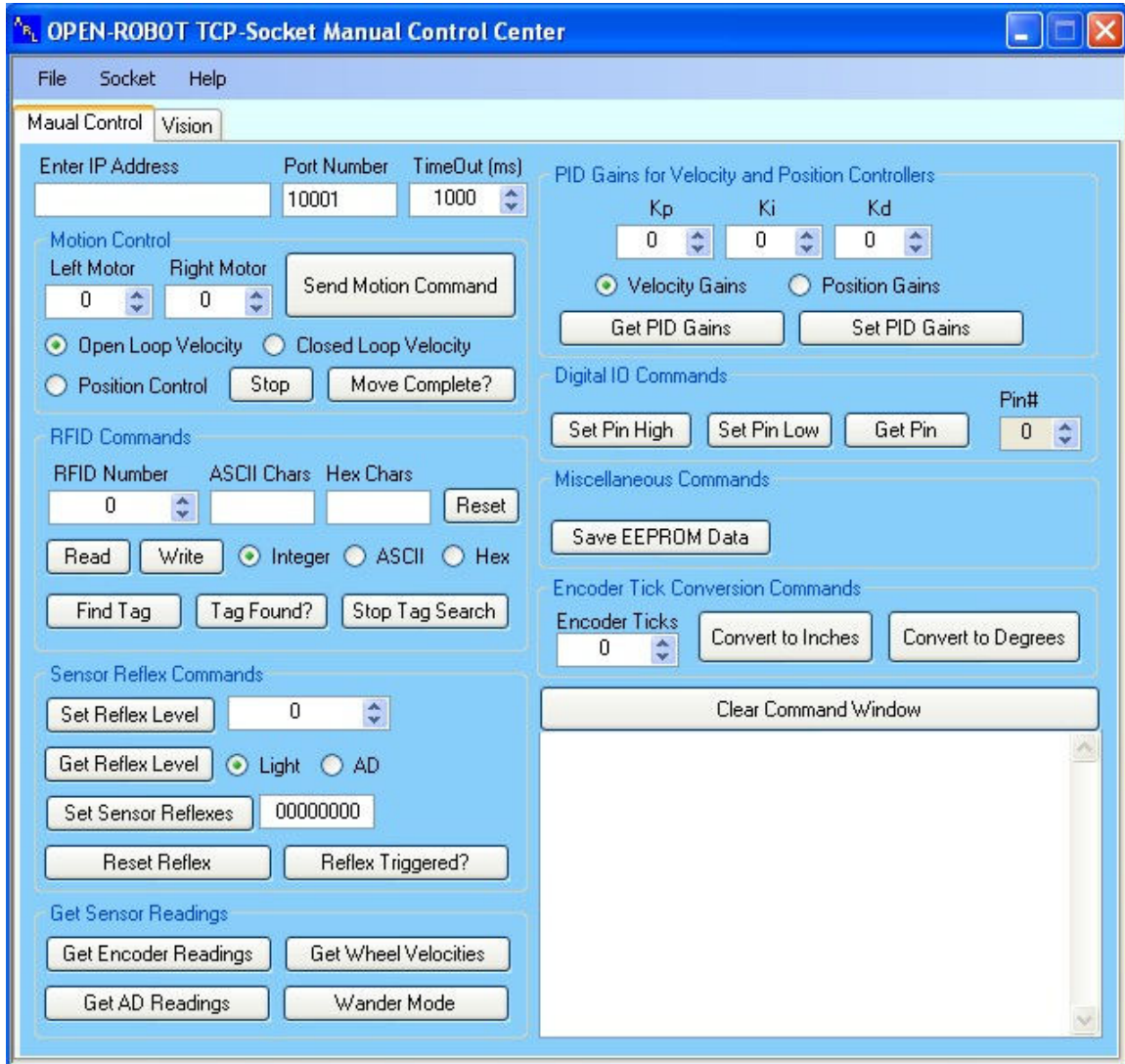


Figure 76. TCP Manual Program.

Connect a 1k Ohm Resistor Across Pin#1 & Pin#6 on J2 Programming Header:

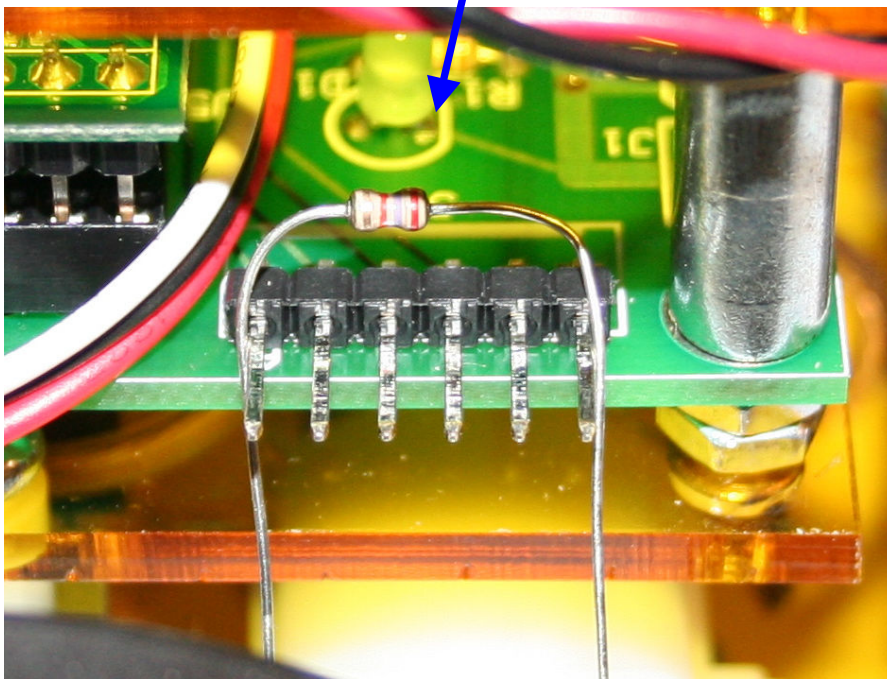
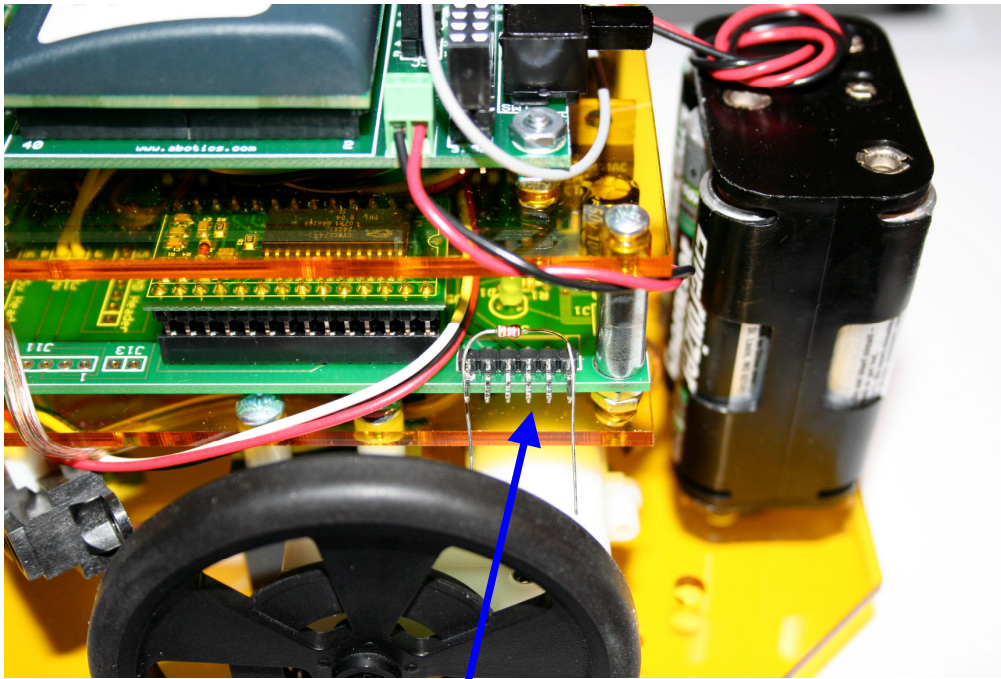


Figure 77. Connect 1k Ohm resistor.

Hold a 1k Ohm resistor across pin#1 and pin#6 on the J2 programming header of OPEN-ROBOT's Controller Board. Now turn on the robot and be sure to hold the resistor firmly in place. After about a second or so of being on you can carefully remove the resistor

because as soon as the PIC18F4520 boots it will check and see the resistor. At this point the PIC18F4520 will enter “boot mode” and wait for a new hex program to be uploaded.

Connect to Robot Using OPEN-ROBOT TCP-Socket Manual Control Program:

Now connect to your robot using the OPEN-ROBOT TCP Socket Manual Control Program. After making the socket connection you can browse for the PIC18F4520 hex code file by selecting “Upload PIC18F4520 Hex File” from the “File” menu.

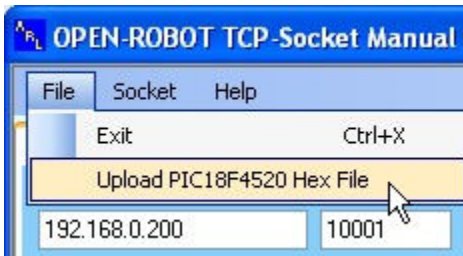


Figure 78. Upload PIC18F4520 Hex File.

Then browse for the new PIC18F4520 hex file to upload.

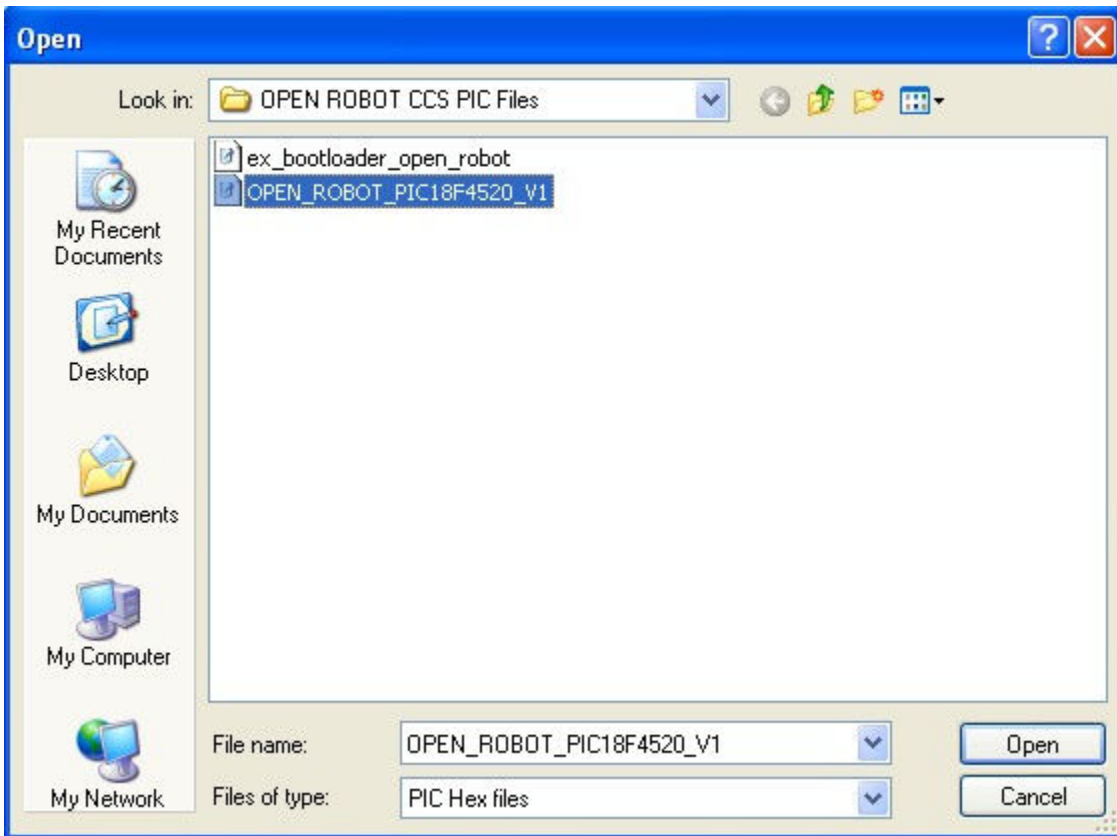


Figure 79. Select hex file.

Select the new hex file and click the “Open” button. If you are creating your own PIC18F4520 programs using the CCS C Compiler then you must follow our instructions for including the “boot loader” files otherwise you will have to reprogram the PIC18F4520 using a In-Circuit Programmer/Debugger pod. ***If you are uploading a new firmware hex code file that was supplied by Abe Howell’s Robotics, then you don’t have to worry about this.***

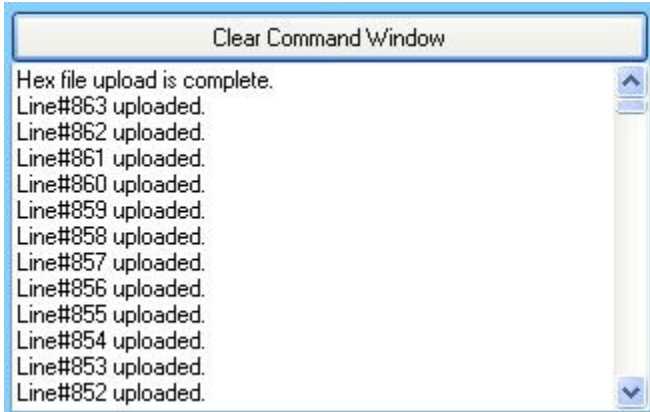


Figure 80. Hex upload complete.

Once the file upload is complete it will print, “Hex file upload is complete” in the message window.

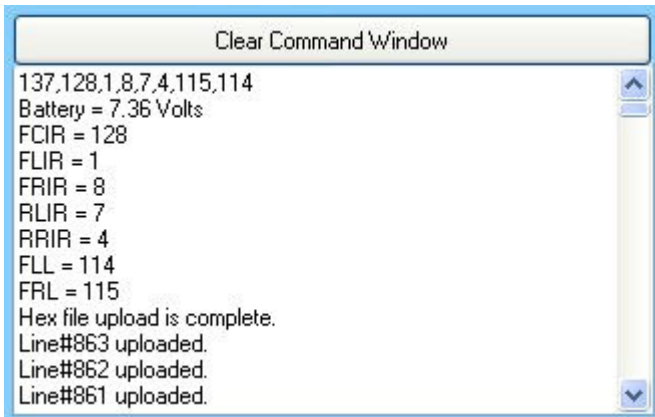


Figure 81. Get A/D sensor readings.

At this point the PIC18F4520 has rebooted and the new hex code should be running, so go ahead and try it out by requesting the A/D Sensor Readings.

Chapter 8: How to modify PIC18F4520 firmware

This chapter will explain the process of modifying OPEN-ROBOT's PIC18F4520 firmware. The PIC18F4520 firmware was created using the CCS C Compiler and also integrates a CCS bootloader. This bootloader allows for firmware upgrades without requiring the use of a PIC programmer unit. This document will focus on using the PCWH IDE version of the CCS C compiler.

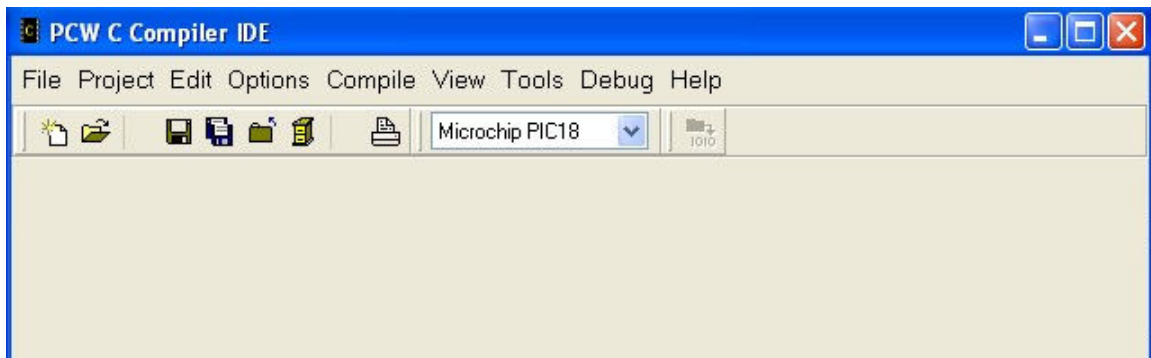
Requirements:

You will need to purchase or have access to a suitable CCS C compiler for the PIC18F4520. I recommend purchasing the PCHW IDE version of the compiler. However, you can use the PCH command line compiler and achieve the same results, but just without the ease of an IDE. You can check out the compilers by following the link below.

<http://www.ccsinfo.com/content.php?page=ideoverview>

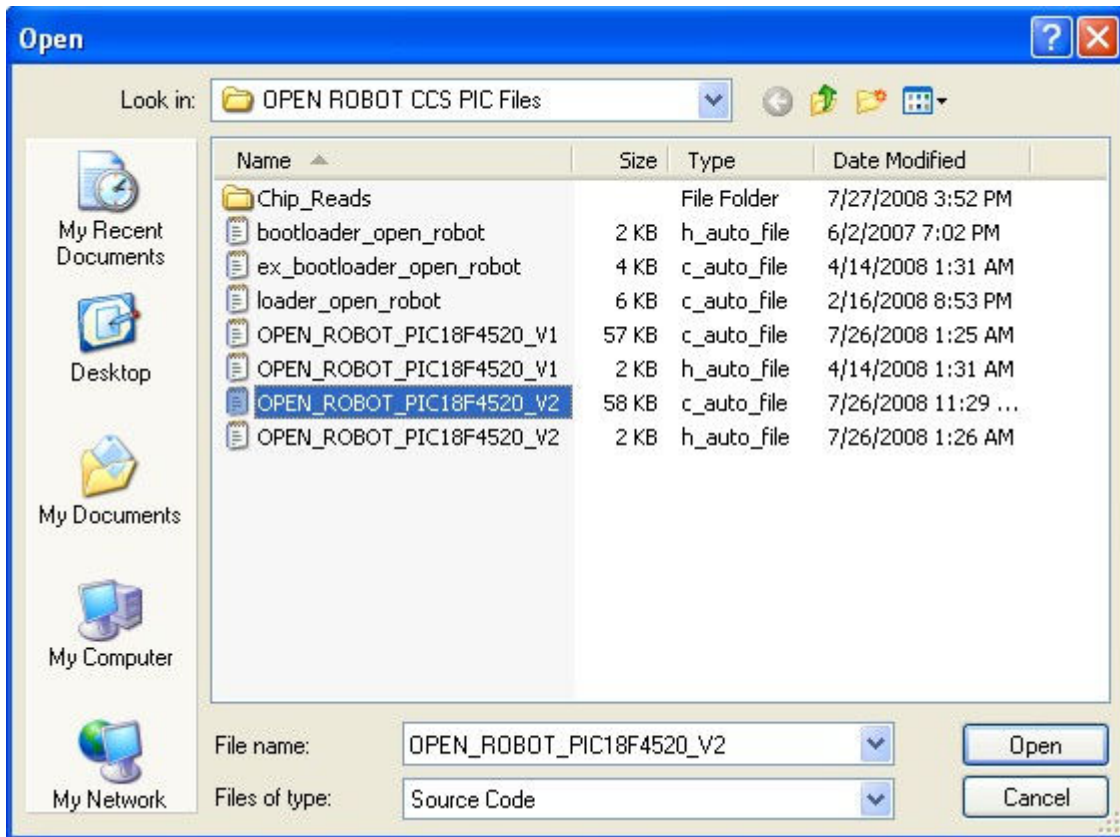
Download the latest version of the PIC18F4520 firmware from the OPEN-ROBOT website. After the download is complete, be sure to extract the zip files to a suitable location.

Getting Started with the PCWH Compiler:



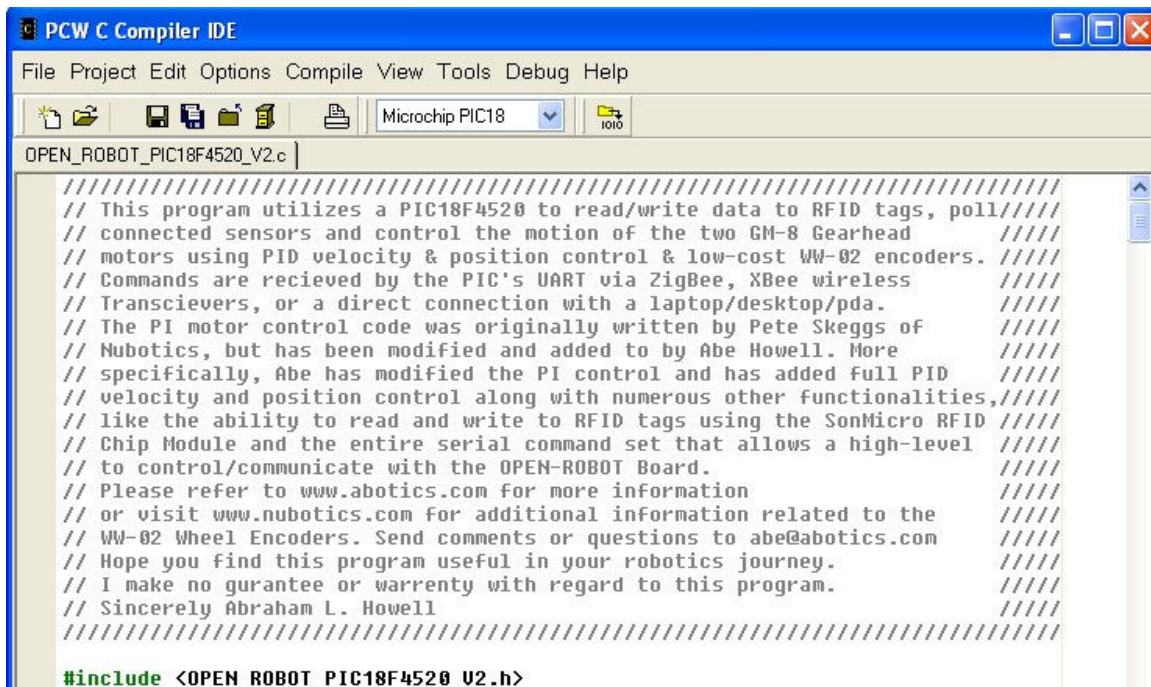
Figure#82. Run PCWH IDE.

Go ahead and run the PCWH IDE. Click Open from the File menu and browse to the location where you unzipped the PIC18F4520 firmware files. For this example the latest version of the firmware happens to be "OPEN_ROBOT_PIC18F4520_V2.c".



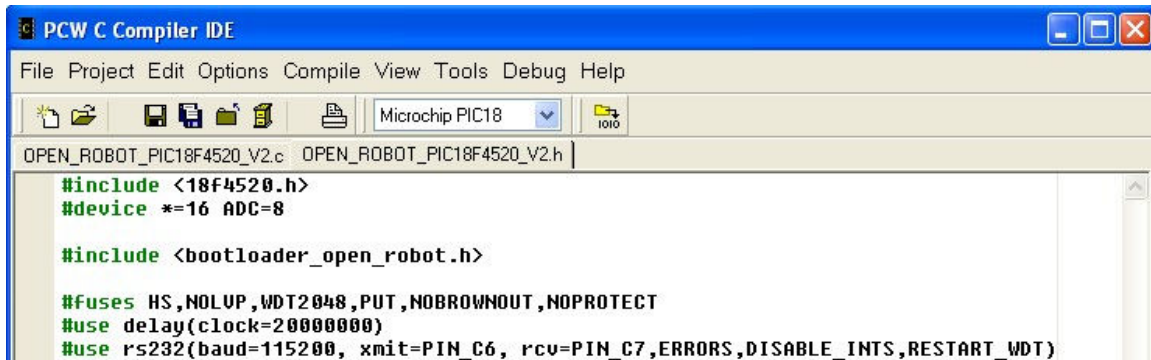
Figure#83. Open Firmware C-File.

Select to view the details of the listed files and select the `c_auto_file`.



Figure#84. Source code displayed in IDE window.

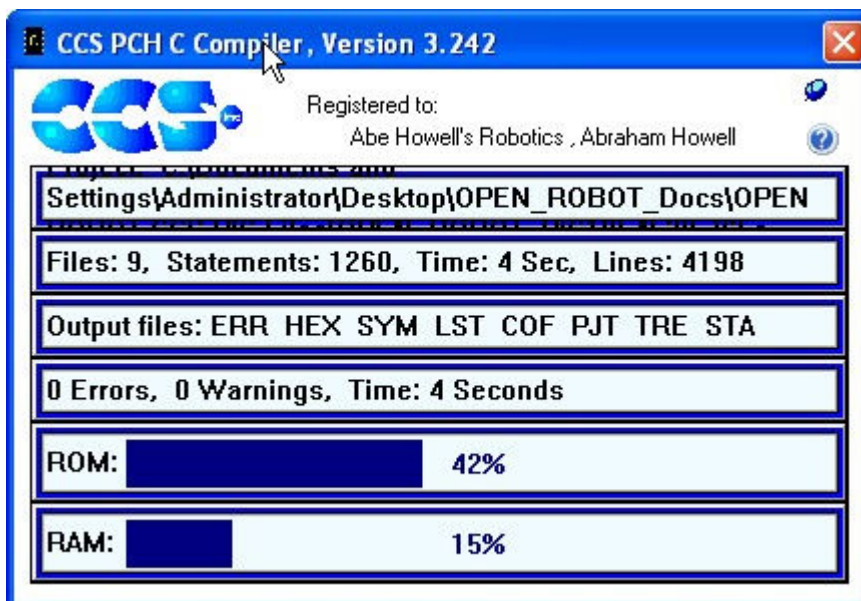
After opening the source c-file you will see the source-code populated in the IDE window as shown above in figure#84.



```
PCW C Compiler IDE
File Project Edit Options Compile View Tools Debug Help
Microchip PIC18
OPEN_ROBOT_PIC18F4520_V2.c OPEN_ROBOT_PIC18F4520_V2.h
#include <18F4520.h>
#define *_16 ADC=8
#include <bootloader_open_robot.h>
#fuses HS,NOLUP,WDT2048,PUT,NOBROWNOUT,NOPROTECT
#use delay(clock=20000000)
#use rs232(baud=115200, xmit=PIN_C6, rcv=PIN_C7,ERRORS,DISABLE_INTS,RESTART_WDT)
```

Figure#85. Open header file.

Now open the corresponding h_auto_file or header file. The source code will be displayed on a second tab. You should notice that the only include file is as follows “[bootloader_open_robot.h](#)”. This include is very important because it lets the compiler know that we are using the bootloader in the firmware.



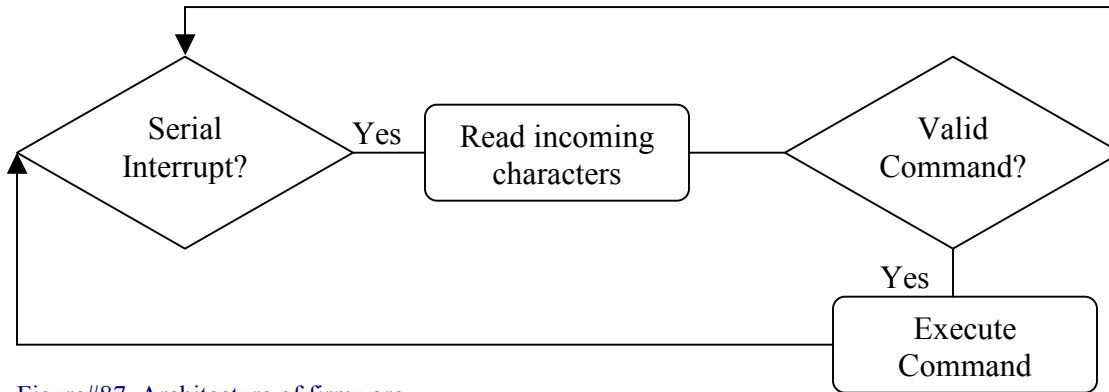
Figure#86. Compile.

Now we can go ahead and test to be sure that our program compiles successfully. Simply select “[Compile](#)” from the Compile menu or press the [F9](#) key. Once the compile is complete you will be notified and the amount of ROM and RAM that has been allocated will be displayed in a pop-up window.

You can now begin modifying the firmware to suit your own needs or simply use this as a starting point for your own custom firmware creation.

Architecture of PIC18F4520 Firmware:

The basic architecture of the firmware is to wait for a serial interrupt to occur, read in the received characters, and once a “\r” char is received decipher the command and execute it.



Figure#87. Architecture of firmware.

The Timer2 interrupt service routine (ISR) is used to update the velocity and position control loops at a fixed frequency. Timer2 is also used to cycle through each of the analog-to-digital pins and retrieve the most current reading for the attached sensor and then store it in the appropriate variable. This ensures that the most up-to-date readings are transmitted when a user requests them.

The Timer0 ISR is used to capture the WW02 Wheel encoder clock signal transitions from the right encoder.

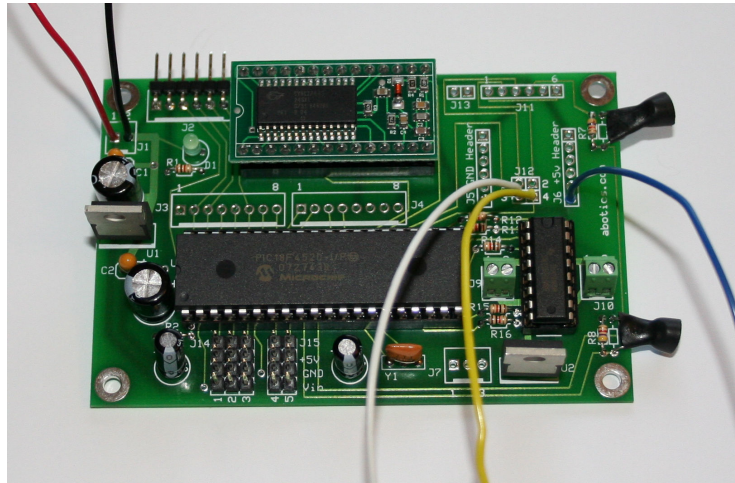
The external ISR for PIN_B0 is utilized to acquire the WW02 Wheel encoder clock signal transitions from the left encoder.

The above-listed hardware interrupts are the only four used in the firmware. Communication with the onboard RFID module is achieved by using a software UART. The remainder of the program is basically a collection of functions that support the Serial-Based firmware interaction with the outside world and provide an easy to use interface for controlling OPEN-ROBOT. For additional information, I recommend taking a look at the OPEN-ROBOT Control Board schematic and the PIC18F4520 I/O Allocation.

http://www.abotics.com/ExpressPCBs/OPEN_ROBOT_BOARD_SCHEMATIC.pdf

http://www.abotics.com/OPEN_ROBOT_DOCS/OPEN_ROBOT_PIC18F4520_IO.xls

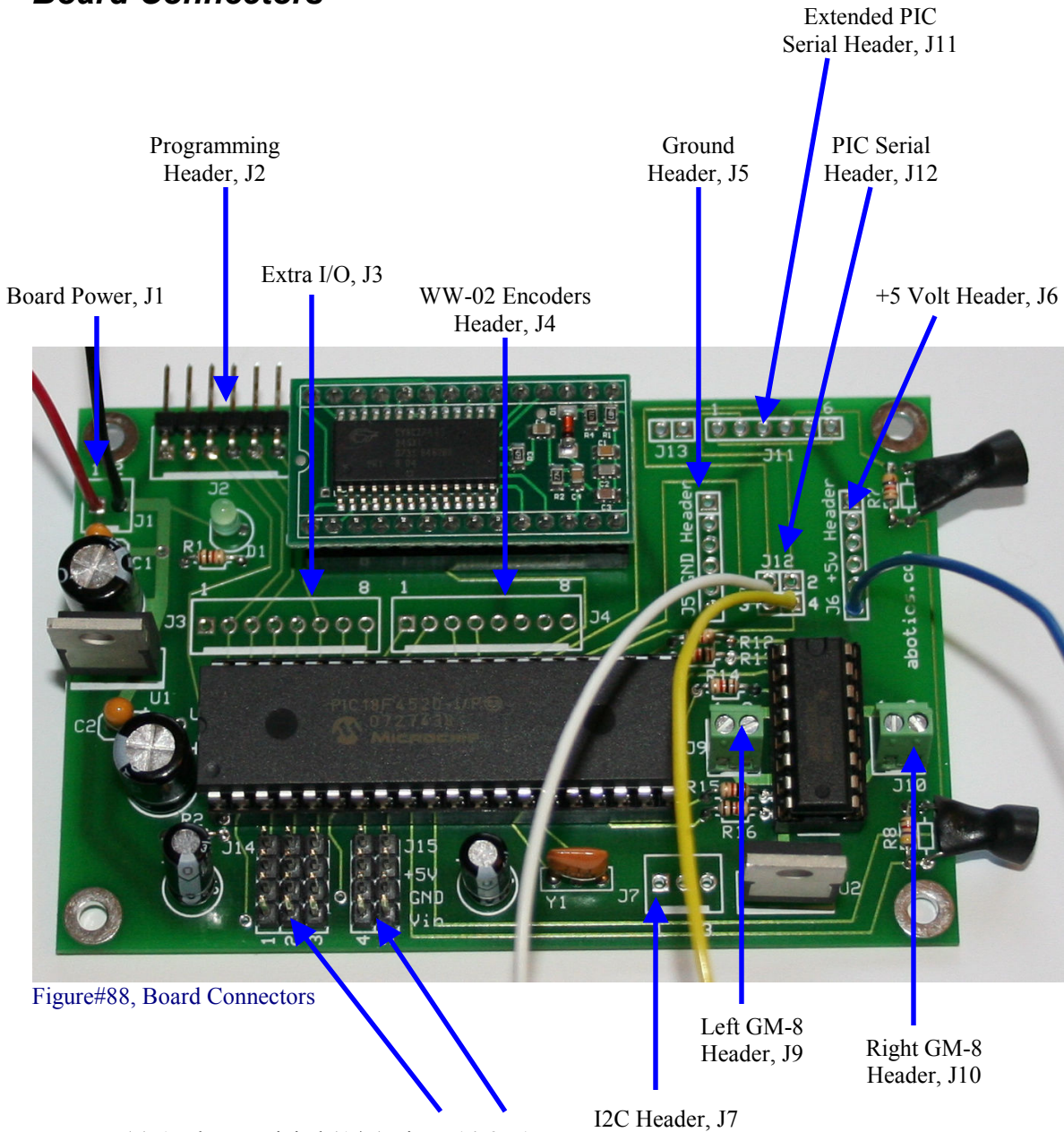
Chapter 9: Details of PIC18F4520 controller board



Introduction to the OPEN-ROBOT Controller Board

This board was specifically designed for the OPEN-ROBOT and is a very unique robot controller in that it not only incorporates a suitable H-Bridge chip for controlling two GM-8 gearhead motors, but it also has dedicated pins and a connector for utilizing two WW-02 Wheel Encoders from for Proportional Integral Derivative (PID) velocity and position control of the GM-8 motors. While this in and of itself is a tremendous benefit, there are other significant features such as direct integration of a low-cost Radio Frequency Identification (RFID) chip module, which gives OPEN-ROBOT the ability to read and write to in-range passive RFID Tags. There are a total of eight (8) analog to digital capable pins, two (2) are dedicated to light sensing photocells, one (1) is used to monitor the board's supply voltage and the remaining five (5) are available to be used as needed by the user. A user-friendly firmware and boot-loader is programmed into the PIC18F4520. The boot-loader allows for wireless uploading of new firmware versions, but also of user created code that is developed using a suitable CCS C compiler. This board also supports development, programming, and debugging of user created programs via a dedicated programming header. This programming header can be used to load and debug new programs or new firmware upgrades and has been tested with an In-Circuit Programmer/Debugger (ICD) from CCS Inc. The OPEN-ROBOT controller board has been developed and evolved over several years and we hope that you find this to be a useful tool in your quest to build a great robot. Please send all questions and comments to support@abotics.com

Board Connectors



Figure#88, Board Connectors

(5) Analog to Digital (A/D) Pins, J14 & J15

Figure#88 points out all the board connectors. The upcoming sections will explain each of these connections in more detail.

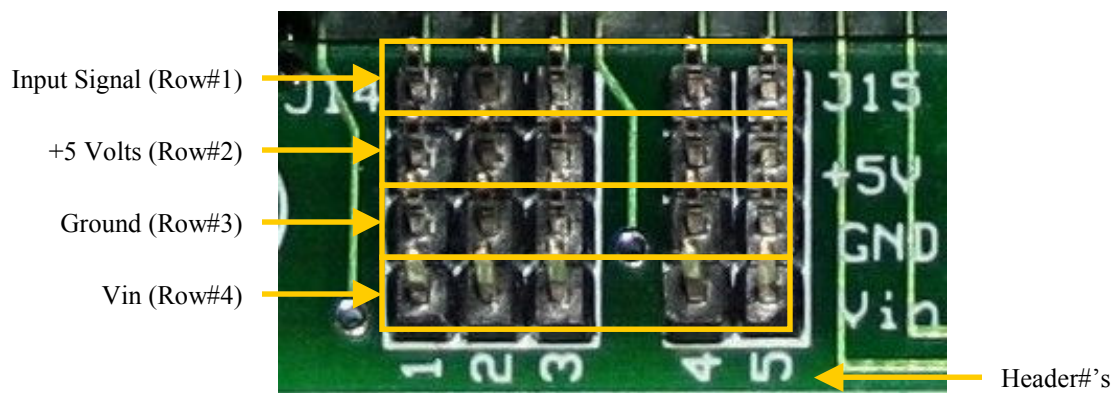
Board Power & Programming Header, J1 & J2

| Header Pin# | Description |
|-------------|--------------------------|
| J1 Pin#1 | V_{in} (6 to 10 volts) |
| J1 Pin#2 | Ground |
| J2 Pin#1 | Ground |
| J2 Pin#2 | +5 volts |
| J2 Pin#3 | PIC18F4520 MCLR |
| J2 Pin#4 | PIC18F4520 Pin B7 |
| J2 Pin#5 | PIC18F4520 Pin B6 |
| J2 Pin#6 | PIC18F4520 Pin B3 |

Figure#89, Power & Programming Headers (J14 & J2)

The board can be powered in the range of 6-10 volts. We recommend using (6)-AA batteries. Power is applied at the J1 header and pin#1 is $+V_{in}$ and pin#2 is ground. A polarized header is used to help prevent plugging the board power connector in backwards. Reversing the polarity of the J1 connector will more than likely result in damage to the one or both of the +5volt regulators at U1 and U2 and possibly other components as well, so please be careful when applying power to the board. The programming header allows the PIC18F4520 to be programmed and debugged with an In-Circuit Programmer/Debugger (ICD) from CCS Inc. J2 pin#1 connects to ground on the board, pin#2 to +5 volts on the board, pin#3 to MCLR on the PIC18F4520, pin#4 to B7 on the PIC18F4520, pin#5 to B6 on the PIC18F4520, and pin#6 to B3 on the PIC18F4520. You will need to fabricate a cable to connect to the appropriate pins on the ICD (*explained later on in detail*). If you have a PIC Programmer you can also fabricate a cable to connect to J2 as well.

Analog to Digital (A/D) Headers, J14 & J15



Figure#90, A/D Headers (J14 & J15)

The J14 & J15 headers provide connection points to the PIC18F4520's analog to digital (A/D) capable pins. Header#1 is for A/D Sensor#1, header#2 is for A/D Sensor#2,

header#3 is for A/D Sensor#3, header#4 is for A/D Sensor#4, and header#5 is for A/D Sensor#5. The headers run vertically so that a GP2D120 or GP2D12 IR Sensor can be plugged in. Each header provides an input signal, +5 volts, and ground pin for the connected IR Sensor. Be sure to plug your sensors into the correct pins, since an incorrect connection can damage either the sensor and/or the board. Each input signal pin connects to a corresponding A/D channel on the PIC18F4520.

I²C Header, J7



| Header Pin# | Description |
|-------------|----------------------|
| J7 Pin#1 | I ² C SDA |
| J7 Pin#2 | I ² C SDL |
| J7 Pin#3 | Ground |

Figure#91, I²C Header, J7

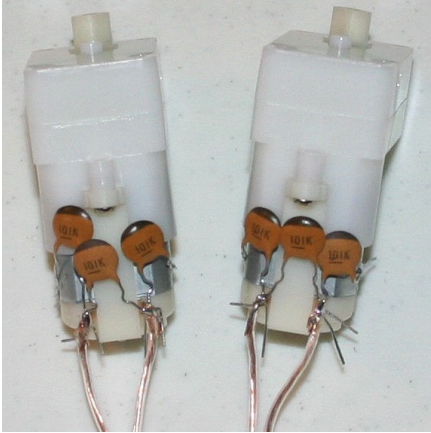
J7 provides access to the board's I²C pins and ground, so that I²C devices can be connected here. Pin#1 on J7 connects to SDA on the PIC18F4520, pin#2 connects to SCL on the PIC18F4520, and pin#3 connects to ground on the PIC18F4520. You will need to solder a suitable 3-position header to use the I²C and make the appropriate PIC18F4520 firmware changes since I²C is not supported in the supplied serial firmware.

GM-8 Motor Headers, J9 & J10

| Header Pin# | Description |
|-------------|-------------------------|
| J9 Pin#1 | Left Motor Top Lead |
| J9 Pin#2 | Left Motor Bottom Lead |
| J10 Pin#1 | Right Motor Bottom Lead |
| J10 Pin#2 | Right Motor Top Lead |

Figure#92a, GM-8 Headers, J9 & J10

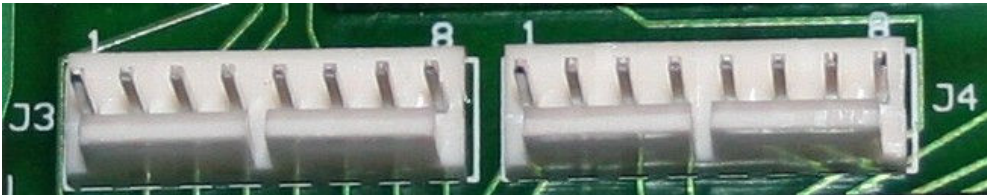
The left GM-8 gear motor connects to the J9 header and the right GM-8 to the J10 header. Connecting the GM-8's incorrectly will only result in improper wheel rotation and encoder readings and can be corrected by simply switching the motor leads.



Figure#92b, GM-8 Noise Suppression Capacitors

Three 100pf ceramic capacitors must be soldered to each GM8 motor when WW02 Wheel Encoders are added otherwise motor noise will negatively affect the WW02 Wheel Encoder readings.

Extra I/O & WW-02 Wheel Encoders Header, J3 & J4



Figure#93, Extra I/O & Wheel Encoders Header, J3 & J4

| Header Pin# | Description |
|-------------|--|
| J3 Pin#1 | PIC18F4520 Pin C5 |
| J3 Pin#2 | PIC18F4520 Pin B7 |
| J3 Pin#3 | PIC18F4520 Pin B6 |
| J3 Pin#4 | PIC18F4520 Pin B5 |
| J3 Pin#5 | PIC18F4520 Pin B4 |
| J3 Pin#6 | PIC18F4520 Pin B3 |
| J3 Pin#7 | PIC18F4520 Pin B2 |
| J3 Pin#8 | PIC18F4520 Pin B1 |
| J4 Pin#1 | Left WW-02 Clock Line (Violet wire) |
| J4 Pin#2 | Right WW-02 Clock Line (Violet wire) |
| J4 Pin#3 | Left WW-02 Direction Line (Orange wire) |
| J4 Pin#4 | Right WW-02 Direction Line (Orange wire) |
| J4 Pin#5 | Left WW-02 Ground (Black wire) |
| J4 Pin#6 | Right WW-02 Ground (Black wire) |
| J4 Pin#7 | Left WW-02 V _{cc} (Red wire) |
| J4 Pin#8 | Right WW-02 V _{cc} (Red wire) |

Please refer to the WW-02 Wheel Encoder User Manual for additional connection information. You will need to either solder the WW02 cables directly to the appropriate locations on J4 or purchase a suitable header, crimp pins, and connector to house the crimp pins. ***WW02 Wheel Encoders must be purchased separately.***

PIC18F4520 Serial Headers, J12

| Header Pin# | Description |
|-------------|-----------------------|
| J12 Pin#1 | Connects to J11, pin2 |
| J12 Pin#2 | Connects to J11, pin3 |
| J12 Pin#3 | PIC18F4520 Serial RX |
| J12 Pin#4 | PIC18F4520 Serial TX |

Figure#94, Serial Headers, J12

The J12 headers provide a connection point to the PIC18F4520's Serial Port. Optionally, you can use a set of jumpers to connect the TX and RX pins to the J11 header. To use this feature simply attach a jumper across J12 pin#3 & pin#1 and a jumper across J12 pin#4 & pin#2. This is not necessary when using OPEN-ROBOT in its default configuration. However, it might be useful if you decide to interface your own wireless hardware.

Extended PIC18F4520 Serial Header, J11 (Optional)

| Header Pin# | Description |
|-------------|---------------|
| J11 Pin#1 | Pin Not Used |
| J11 Pin#2 | PIC18F4520 TX |
| J11 Pin#3 | PIC18F4520 RX |
| J11 Pin#4 | Ground |
| J11 Pin#5 | +5 volts |
| J11 Pin#6 | Pin Not Used |

Note: This Board is a TTL device and can only accept serial connections that are in the 0 to 5 volt range. You must use a level shifter cable to connect to non-TTL serial devices, which have levels in the -15 to 15 volt range.

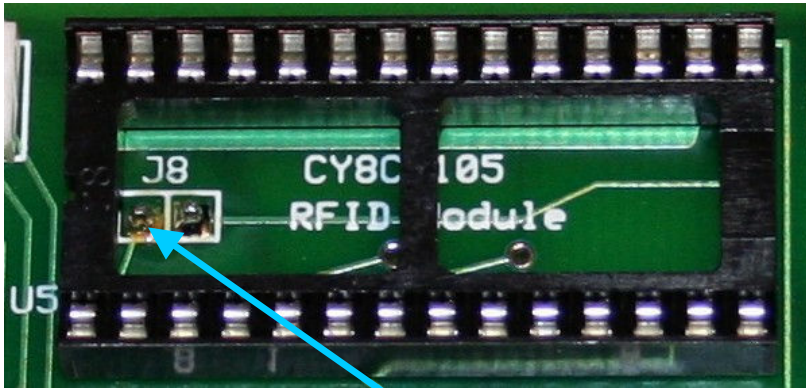
Figure#95, Serial Header, J11

To use this feature, attach a jumper across J12 pin#3 & pin#1 and a jumper across J12 pin#4 & pin#2. This is not necessary when using OPEN-ROBOT in its default configuration. However, it might be useful if you decide to interface your own wireless hardware.

RFID Module Jumper Header, J13 (Not Required)

The J13 Jumper is no longer required. Please refer to the documentation available on SonMicro's web site: www.sonmicrom.com for more information.

RFID Antenna Header, J8



Top Side of Board

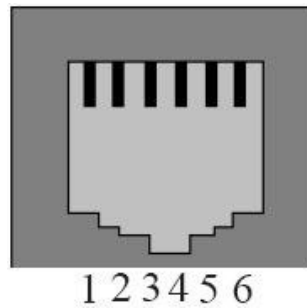
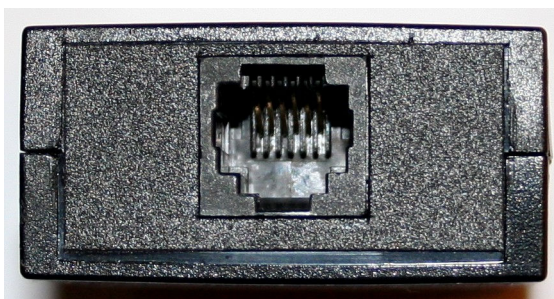
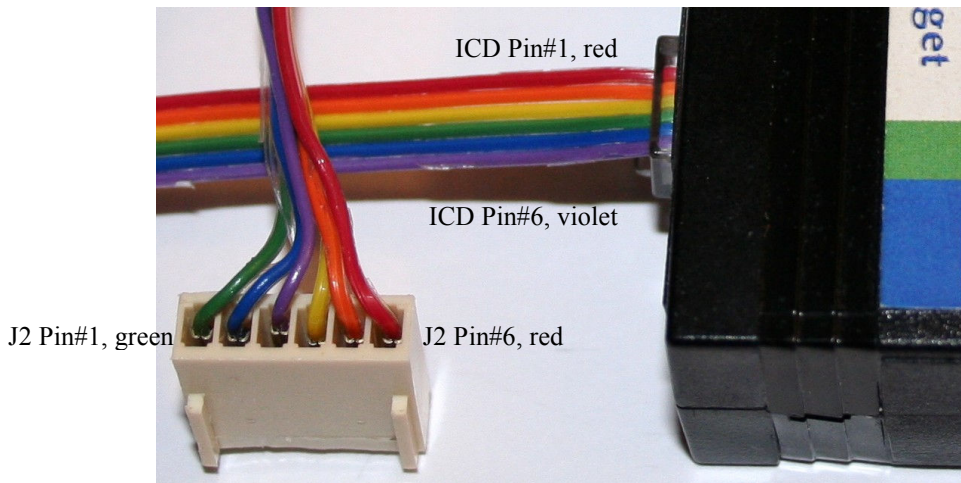


Bottom Side of Board

Figure#96, RFID Antennae Header, J8

The RFID antennae header, J8, is located on the bottom side of the board. This is where the RFID antennae must be plugged in. The RFID antenna that SonMicro sells for use with the RFID Chip Module has short leads. These need to be extended by a short cable so that the antenna can be mounted properly on the base of OPEN-ROBOT.

How to Connect the CCS ICD to the Programming Header



Figure#97, CCS ICD Cable & Connectors

| CCS ICD Connector to Program Connector | |
|---|---------------------|
| ICD Pin# | J2 Pin# |
| Pin#1 (Red Wire) | Pin#6 (Red Wire) |
| Pin#2 (Orange Wire) | Pin#5 (Orange Wire) |
| Pin#3 (Yellow Wire) | Pin#4 (Yellow Wire) |
| Pin#4 (Green Wire) | Pin#1 (Green Wire) |
| Pin5 (Blue Wire) | Pin#2 (Blue Wire) |
| Pin#6 (Violet Wire) | Pin#3 (Violet Wire) |

Figure#98, ICD to Program Header, J2, Table

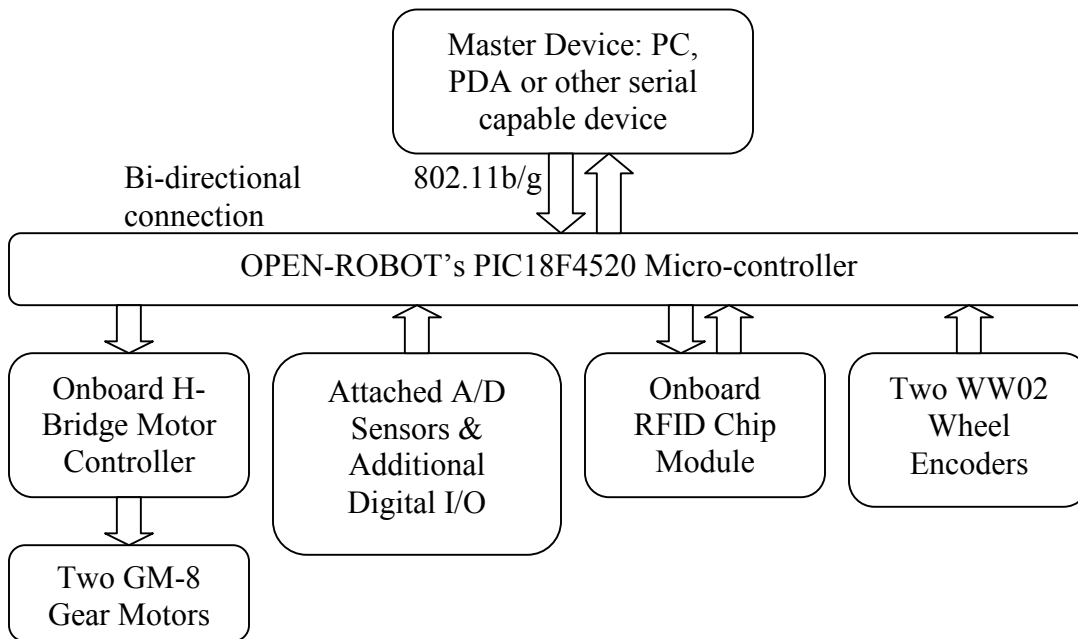
To build the above ICD programming cable you will need approximately 6-8” of 6-conductor multicolor 28-gauge flat ribbon cable and a 6-pin RJ25 modular connector. The 6-pin connector can be purchased from Radio Shack, **part#279-421**. The flat ribbon cable (**part#112547**) can be purchased from Jameco Electronics. You will also need one 6-position 0.1” non-polarized connector (**part#103211**) and (6) crimp pins (**part#100756**) from Jameco Electronics. It is recommended that you use the appropriate crimp tools, but you can sometimes get away with using a pair of regular or needle-nose pliers. Updating firmware or loading your own custom program can be performed using the MatchPort b/g module and this does not require the use of an In-Circuit Programmer since the PIC18F4520 has been pre-programmed with a boot-loader. Using the In-Circuit Programmer/Debugger allows for true hardware debugging and can be an invaluable tool when developing new code.

OPEN-ROBOT Serial-Based Firmware Command Set

The 802.11b/g protocol is used to maintain a wireless communication link between OPEN-ROBOT and a corresponding 802.11b/g wireless equipped laptop, desktop pc, Pocket PC (PPC) or micro-controller. However, as an option you can wire OPEN-ROBOT’s serial port directly to a suitable device, but this will require that you fabricate any necessary cables. The default serial port settings for OPEN-ROBOT are as follows: 115,200BPS, 8-data bits, 1-stop bit and no parity. Once properly installed & setup the user can communicate with OPEN-ROBOT and create their own high-level programs using Visual Studio.Net® or any software development tool of choice.

Simple serial command packets are used to communicate with OPEN-ROBOT and these packets will inevitably vary in size, since different commands require either more or less information. An entire command packet must be constructed and sent by the high-level controlling program. OPEN-ROBOT will receive this packet and determine the packet’s nature. Once OPEN-ROBOT deciphers the enclosed packet command, it will begin execution or return the requested data. All incoming move commands take priority over a previously issued command. Finally, all packets must be terminated with a line return (“\r”) character or else they will become part of the next incoming command upon receipt of the next line return along with any other packets that were sent prior to the line return. The line return character allows OPEN-ROBOT to determine the end of a command packet and then parse out the appropriate data. A built-in watchdog timer will restart the PIC18F4520 if it is not reset in time, which in turn eliminates OPEN-ROBOT from

locking up. A diagram below illustrates the overall communication structure for OPEN-ROBOT.



Figure#99, OPEN-ROBOT's Communication Structure

Serial-Based command set for OPEN-ROBOT.

Get All A/D Sensor Readings: "A\r"

- Will retrieve all the 8-bit A/D sensor readings.
- Returns "aX₀,X₁,X₂,X₃,X₄,X₅,X₆,X₇\r"
- X₀ = Channel 0 (battery level).
- X₁ = Channel 1 (Front Center IR A/D Sensor J14 Pin#1). *Purchase Separately*
- X₂ = Channel 2 (Front Left IR A/D Sensor J14 Pin#2).
- X₃ = Channel 3 (Front Right IR A/D Sensor J14 Pin#3).
- X₄ = Channel 4 (Rear Left IR A/D Sensor J15 Pin#4). *Purchase Separately*
- X₅ = Channel 5 (Rear Right IR A/D Sensor J15 Pin#5). *Purchase Separately*
- X₆ = Channel 6 (Right Front Light Sensor).
- X₇ = Channel 7 (Left Front Light Sensor).
- Example return: "a140,120,110,108,0,1,145,151\r"

Get Digital I/O Pin Reading: "LX₀\r"

- Gets the appropriate I/O pin reading.
- X₀ = Pin number (0 to 7).
- Returns "I" + "**pin reading**" + "\r" upon receiving the command.
- The pin reading value will be either a **1** or **0**. A **1** corresponds to +5 volts and **0** is 0 volts.
- Example Return: "I0\r"

Get Wheel Encoder Readings: “E\r”

- Retrieves the left and right encoder readings.
- Returns “e X_0 , X_1 \r”
- X_0 = Left encoder reading.
- X_1 = Right encoder reading.
- Example Return: “E100,99\r”

Get A/D Sensor and Light Reflex Levels: “F\r”

- Retrieves 8-bit A/D Sensor and Light Reflex Level.
- Returns “f X_0 , X_1 \r”
- X_0 = A/D Reflex Level.
- X_1 = Light Reflex Level.
- Example Return: “f60,200\r”

Get PID Gains: “GX₀\r”

- Gets the current set of 8-bit PID gains for the velocity or position controller.
- X_0 = For position send “P” and for velocity send “V”
- Returns “gpKp,Ki,Kd\r”
- Returns “gvKp,Ki,Kd\r”
- Example Return: “gp5,2,1\r”
- Default Position PID gains are **Kp = 5, Ki = 2, and Kd = 1.**
- Default Velocity PID gains are **Kp = 5, Ki = 1, and Kd = 1.**

Get PWM Motor Offsets: “@G\r”

- Gets the current left and right 8-bit PWM Motor Offsets. These values are used to offset the motor PWMs. The PIC18F4520’s hardware PWM can be set from 0 to 255, but the GM-8 motors may not begin to spin until say 195. By setting the PWM Offsets to 195 we ensure that the motors start moving when we send an open-loop speed greater than zero. This also helps the robot to drive straighter during open loop velocity control, but is equally important for proper PID velocity and position control.
- Returns “@ X_0 , X_1 \r”, where X_0 is the left PWM Motor Offset and X_1 is the right PWM Motor Offset.
- Example Return: “@195,198\r”

Is Robot Sensor Reflex Triggered: “B\r”

- Checks to see if robot is blocked by a sensor reflex. Used in conjunction with “Set sensory reflex”, and “Reset Agent Blockage” commands.
- Returns “bc\r” if clear and “bb\r” if blocked.

Is Move Complete: “D\r”

- Checks to see if the current move is complete.
- Returns “da\r” if move complete and “dn\r” if not.

- When using Position control you do not know how long a move will take so simply check this periodically until the move is complete and then issue the next command. This way the high-level controlling program can perform other calculations while the robot moves to the requested position

Reset Robot Sensor Reflex: “R\r”

- Resets the robot’s sensor reflex. Used in conjunction with “**Set sensory reflex**” and “**Is Robot Sensor Reflex Triggered**” commands.
- Returns “r\r” upon receipt of the command.

Reset RFID Module: “X\r”

- Resets the RFID Module.
- Returns “x\r” upon receipt of the command.

RFID Tag Found: “z\r”

- Checks to see if a valid RFID tag has been found by “**RFID Search for Tag**”. Used in conjunction with “**RFID Search for Tag**” command, “Z\r”.

RFID Search for Tag: “Z\r”

- Continuously searches for an in-range tag.
- Halts robot when tag is located or Stop Robot, “H\r”, is issued.
- Used in conjunction with RFID Tag Found command, “z\r”.

RFID Tag Read: “T\r”

- Attempts to read an in range RFID tag. If no tag in range or error, returns “tn\r”.
- Returns “t” + “**four hex data bytes in ASCII**” + “r” upon receiving the command.
- Example Return: “**tabcd\r**”, when the four bytes of tag data is converted from ASCII characters to hexadecimal we get the following: “**t61626364\r**”.

RFID Write Tag: “WX₀,X₁,X₂,X₃\r”

- Writes four ASCII bytes to an in range tag. Only way to guarantee a successful write is to read the tag afterwards and check the data.
- X₀ = Byte#0.
- X₁ = Byte#1
- X₂ = Byte#2
- X₃ = Byte#3
- Returns “wa\r” upon receiving and acknowledging the command.
- Suppose we want to write “**abcd**” to the tag. We would send the following: “**Wabcd\r**”. If we then read the tag we should get: “**tabcd\r**”.

Save All Parameters To PIC18F4520 Data EEPROM: “Y\r”

- Saves all parameters to the PIC18F4520 data EEPROM.
- Saves velocity gains (KP, KI, KD), position gains (KP, KI, KD), A/D Reflex Level, Light Reflex Level.

- Returns “y\r” upon receipt of the command.

Set A/D Sensor Reflex Level: “CX₀\r”

- Sets A/D Sensor Reflex level for Sensory Reflexes.
- X₀ = 8-bit A/D sensor level to use with Sensory Reflexes.
- Use “**Save All Parameters To PIC18F4520 Data EEPROM**” so that setting persists.
- Returns “c\r” upon receipt of the command.

Set Light Sensor Reflex Level: “NX₀\r”

- Sets Light Sensor Reflex level for Sensory Reflexes.
- X₀ = 8-bit Light sensor level to use with Sensory Reflexes.
- Use “**Save All Parameters To PIC18F4520 Data EEPROM**” so that setting persists.
- Returns “n\r” upon receipt of the command.

Set Sensory Reflexes: “IX₀\r”

- Sets sensory reflexes to be active or inactive for each of the (8) A/D Channels. In active mode the agent will immediately stop when any of the active A/D Channels exceeds the value set by **Set A/D Reflex Level or Set Light Reflex Level**.
- X₀ = An 8-bit value that sets the specified A/D channels to be active or inactive.
- Each of the (8)-A/D channels is represented by one of the 8-bits. The least significant bit, bit 0, represents A/D channel#0 (battery level), bit 1 corresponds to A/D channel#1 (Front Center IR, J14 Pin#1), ... , bit 7 corresponds to the front left light sensor. To turn off all the A/D channel reflexes we would need to send a zero (0), which would correspond to the following 8-bit string: “00000000”. If we only wanted to turn on the reflex for the Rear Right IR (A/D channel#5) we would send a hexadecimal **20** for X₀, which corresponds to the following binary string: “00100000”. If we only wanted to turn on the reflex for the front right light sensor we would send a hexadecimal **40** for X₀, which corresponds to the following binary string: “01000000”. Next you would have to convert the hexadecimal value to a corresponding character to send out as serial data. All Sensory Reflexes are turned off upon startup.
- Returns “i\r” upon receipt of the command.

Set Digital I/O Pin Low: “JX₀\r”

- Sets the appropriate I/O pin low.
- X₀ = Pin number (0 to 7).
- Returns “j\r” upon receiving the command.

Set Digital I/O Pin High: “KX₀\r”

- Sets the appropriate I/O pin high.
- X₀ = Pin number (0 to 7).
- Returns “k\r” upon receiving the command.

Set Position PID Controller Gains: “PX₀X₁\r”

- Sets the specified 8-bit Position PID gain.
- **X₀** = For KP send “P”, KI send “I” and KD send “D”.
- **X₁** = 8-bit Gain Value.
- Use “**Save All Parameters To PIC18F4520 Data EEPROM**” so that setting persists.
- Returns “**p\r**” upon receiving command.
- Default Position PID gains are K_p = 5, K_i = 2, K_d = 1.
- Changing the default values can degrade OPEN-ROBOT’s performance.

Set PWM Motor Offsets: “@X₀X₁\r”

- Sets the specified 8-bit PWM Motor Offset.
- **X₀** = “L” to set the left PWM Motor Offset or “R” to set the right PWM Motor Offset.
- **X₁** = 8-bit PWM Motor Offset value to set.
- Use “**Save All Parameters To PIC18F4520 Data EEPROM**” so that this setting change persists.
- Returns “**@\r**” upon receiving command.
- Default PWM Motor Offset value is **195**.
- Example to set left PWM Offset to 199. “**@L199\r**”.
- Changing the default value can degrade OPEN-ROBOT’s performance.
- Read PWM Motor Offsets using the Get PWM Motor Offsets command, “**@G\r**”.

Set Velocity Controller PID Gains: “Q X₀X₁\r”

- Sets the specified 8-bit Velocity PID gain.
- **X₀** = For KP send “P”, KI send “I” and KD send “D”.
- **X₁** = 8-bit Gain Value.
- Use “**Save All Parameters To PIC18F4520 Data EEPROM**” so that setting persists.
- Returns “**q\r**” upon receiving command.
- Default Velocity PID gains are K_p = 5, K_i = 1, K_d = 1.
- Changing the default values can degrade OPEN-ROBOT’s performance.

Set Position Closed Loop: “MP X₀, X₁\r”

- Sets the desired position in terms of wheel encoder ticks and uses PID position & velocity control to complete the move. Use **Is Move Complete** to check for when the move is done.
- **X₀** = Number of Left encoder ticks.
- **X₁** = Number of Right encoder ticks.
- Left & Right encoder ticks must be an integer in the range of –250 to 250.
- All incoming move commands will cancel a previously issued command.
- Returns “**ma\r**” upon receiving and acknowledging the command.

Set Velocity Closed Loop: “MV X₀, X₁\r”

- Sets the wheel velocities and performs PID velocity control.

- X_0 = Left motor velocity.
- X_1 = Right motor velocity.
- Left/Right motor velocity will be an integer in the range of -55 to 55 and have units of tenths of an inch per second.
- All incoming move commands will cancel a previously issued command.
- Returns “**ma\r**” upon receiving and acknowledging the command.

Set Velocity Open Loop: “MO X_0 , X_1 \r”

- Sets the open loop velocity of each wheel.
- X_0 = Left PWM motor speed.
- X_1 = Right PWM motor speed.
- Left/Right PWM motor speed will be an integer in the range of -55 to 55 .
- All incoming move commands will cancel a previously issued command.
- Returns “**ma\r**” upon receiving and acknowledging the command.

Stop Robot: “H\r”

- Immediately stops the GM-8 Motors, however, robot may coast before stopping.
- Returns “**h\r**” upon receiving the command.

Wander Mode: “U\r”

- Robot will wander with an open loop velocity of 25 and avoid obstacles based upon currently set **A/D Sensor Reflex Level** and using the front left and front right IR sensor.
- Behavior halts when **Stop Robot** command, “**H\r**”, is issued.

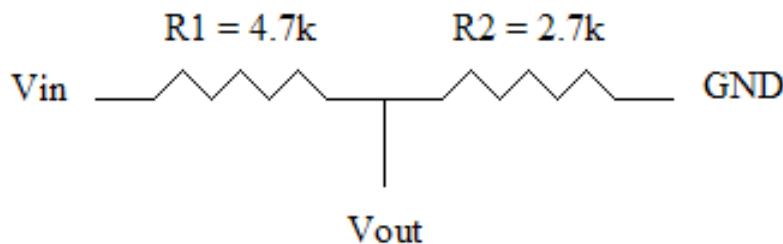
If you intend to use Visual Studio® to create high level control programs then you might want to use our freely available C# class library, which encapsulate all the above-listed commands for OPEN-ROBOT and allows you to quickly get started with writing some useful code.

Chapter 10: Details of analog to digital (A/D) sensors

This chapter will explain how to interpret OPEN-ROBOT's analog-to-digital sensor readings. The battery level, GP2D120 infrared sensors and cadmium sulfide light sensors comprise this sensor array.

OPEN-ROBOT's Battery Voltage Measuring Circuit:

Battery voltage cannot be measured directly by one of the PIC18F4520's analog-to-digital capable pins since we are using 6-AA batteries and the output voltage will vary from about 7 to 9 volts. The PIC18F4520 is setup to measure input voltage by using a +5volt reference, so we need to map the 7 to 9volts over to 5 volts. To accomplish this we have created a simple voltage divider circuit by placing (2) appropriately sized resistors in series as shown below. The output voltage from the 6-AA batteries is V_{in} and the V_{out} and is measured by the PIC18F4520 and then compared against the +5 volt reference to generate an 8-bit digital value.



Figure#100. Voltage Divider for Battery Voltage Measurement.

To solve for the battery voltage, V_{in} , we can use the following formula.

$$V_{in} = [(8\text{-bit digital value}) * (R1 + R2) * 5\text{volts}] / (R2 * 255)$$

If the OPEN-ROBOT returns a 128 for the battery voltage, we can convert this 8-bit number to volts.

$$V_{in} = [(128) * (7.4\text{kohms}) * 5\text{volts}] / (2.7\text{kohms} * 255) = 6.87 \text{ volts}$$

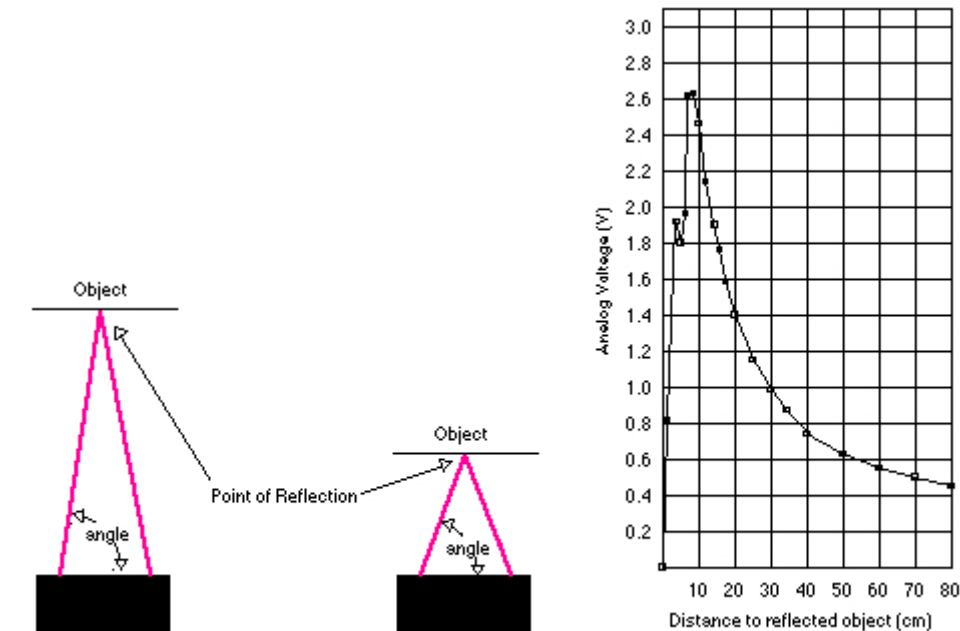
When using rechargeable batteries you should recharge at around 6.87 volts or 128.

OPEN-ROBOT's GP2D120 IR Sensors:



Figure#101. GP2D120 IR Sensor.

The GP2D120 IR sensor is a small and powerful sensor. Triangulation is used to measure the distance to an incident object. The infrared output is modulated at 40KHz to prevent erroneous measurements due to ambient infrared. Measurements are taken approximately every 30 to 40ms.



Figure#102. GP2D120 Triangulation and output.

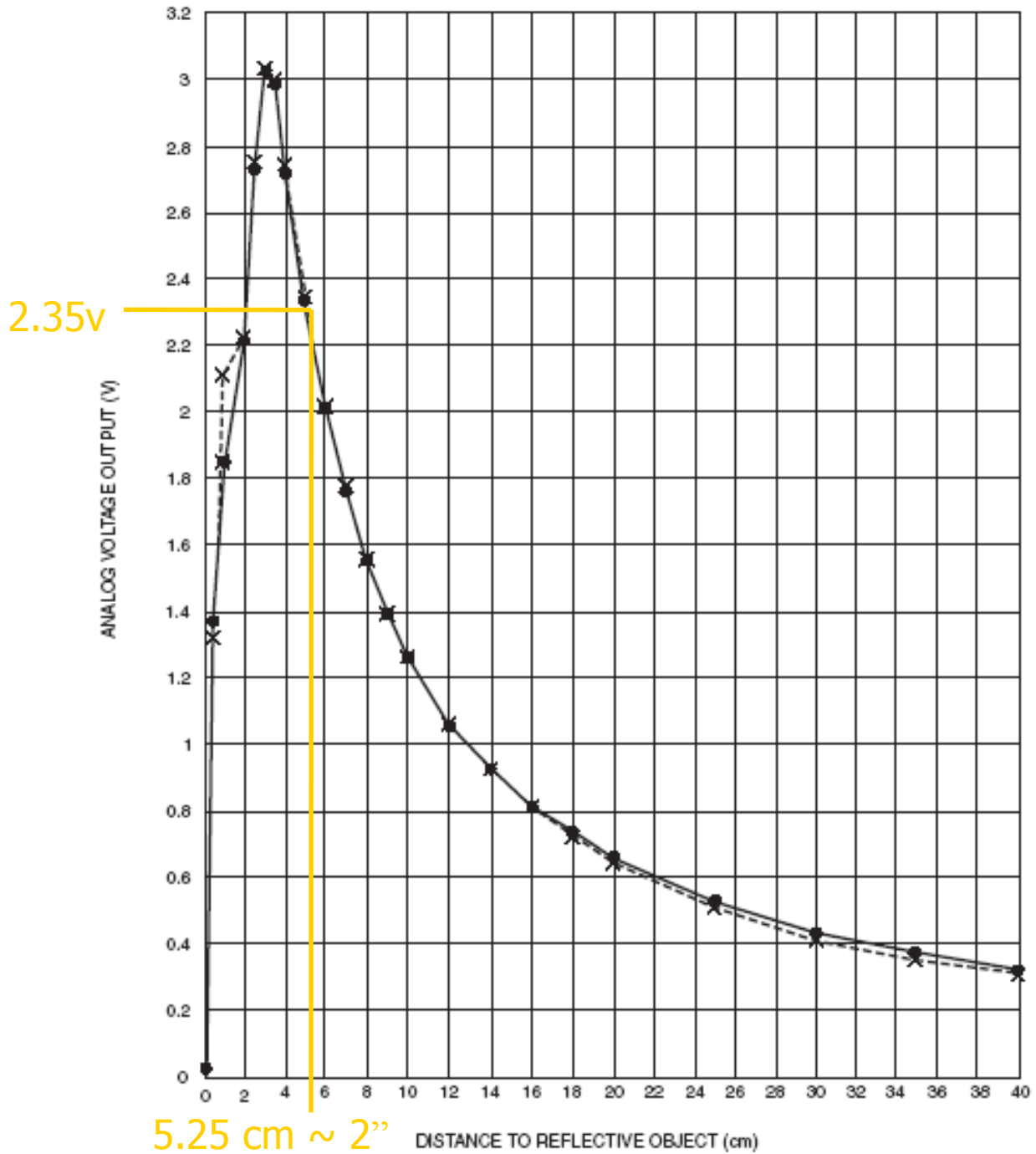
The output from a GP2D120 is a varying voltage level. The PIC18F4520 can directly measure the output of a GP2D120 since it is within the acceptable 0 to 5 volt range. The PIC18F4520 converts the voltage output to an 8-bit value. This 8-bit value can be converted to a distance reading in units of inches, centimeters, or any other unit of length. The output from a GP2D120 is non-linear and can be seen above in figure#102. To convert the 8-bit value to a distance we must first convert the 8-bit digital value back to a voltage reading. We can do this using the formula below.

$$V_{out} = (8\text{-bit digital value}) * 5\text{volts} / 255$$

If the returned 8-bit value is 120 we can calculate the voltage as follows.

$$V_{out} = (120) * 5\text{volts} / 255 = 2.35 \text{ volts}$$

To convert this to a distance reading we need to look at the GP2D120 data sheet.



Figure#103. GP2D120 Voltage Output versus Object Distance Data Sheet.

Looking at the data sheet in figure#103 it can be seen that 2.35 volts corresponds to an object distance of approximately 5.25cm or 2.07 inches. This method works fine if you are only interested in converting a few distance readings, but what if you want to be able to convert these on the fly? There are several different ways to accomplish this. You could take all the data points from figure#103 and build a spreadsheet of output voltage versus object distance. Then use the spreadsheet to create a 2-D plot. Now we simply have a plot similar to the one in figure#103, but what good is this? Now we can perform a regression fit of the plot and attain an equation for the line. You can then use this equation to calculate object distance in inches or centimeters on the fly. You must of course be careful with regard to how you fit the regression line i.e. would it be best to use a n-degree polynomial, logarithmic, power, or linear fit? Using the wrong fit will simply calculate erroneous distance measurements.

You could also use the same spreadsheet information as a look-up table in your program. After converting the 8-bit reading to a voltage measurement simply look up that value in your table. What if the value falls in-between the values in your spreadsheet? For a coarse calculation you could approximate by using the closest value. For example, suppose you had an 8-bit value of 120, which is equal to 2.35 volts. You also have the following spreadsheet data.

2.38 volts = 5.0cm
2.00 volts = 6.0cm

We would pick the closer spreadsheet value and use 2.35 volts, which corresponds to 5.0cm. To attain a more precise value we can perform a linear interpolation using the spreadsheet data and the formula for linear interpolation, which can be found below.

$$Y2 = [(X2 - X1)(Y3 - Y1) / (X3 - X1)] + Y1$$

Where X1 = 2.38volts, Y1 = 5.0cm, X2 = 2.35volts, X3 = 2.00volts, and Y3 = 6.0cm

Solving for Y2 yields.

$$Y2 = [(2.35\text{volts} - 2.38\text{volts})(6.0\text{cm} - 5.0\text{cm}) / (2.00\text{volts} - 2.38\text{volts})] + 5.0\text{cm} = 5.08\text{cm}$$

The GP2D120 data sheet can be downloaded from the following location.

http://www.abotics.com/OPEN_ROBOT_DOCS/GP2D120_SS.pdf

OPEN-ROBOT's Light Sensors:

OPEN-ROBOT has (2)-frontward facing cadmium-sulfide cells that are capable of measuring the amount of ambient light. A cadmium-sulfide (cds) cell is simply a photo-resistor and changes resistance based upon the amount of incident light. As the amount of

light increases the cds cell's resistance drops and as the amount of light decreases the resistance increases. To measure this change in resistance with the PIC18F4520 we need to place the cds cell into a voltage divider with a 4.7kohm resistor. We can calculate the cds cell's resistance value for the current voltage reading by using the voltage divider formula from before.

$$V_{in} = [(8\text{-bit digital value}) * (R1 + R2) * 5\text{volts}] / (R2 * 255)$$

Where $V_{in} = 5$ volts, $R1 =$ cds cell resistance, $R2 = 4.7\text{kohms}$

$$V_{in} = [(8\text{-bit digital value}) * (R1 + R2) * 5\text{volts}] / (R2 * 255)$$

$$R1 + R2 = (R2 * 255 * V_{in}) / (8\text{-bit digital value}) * 5\text{volts}$$

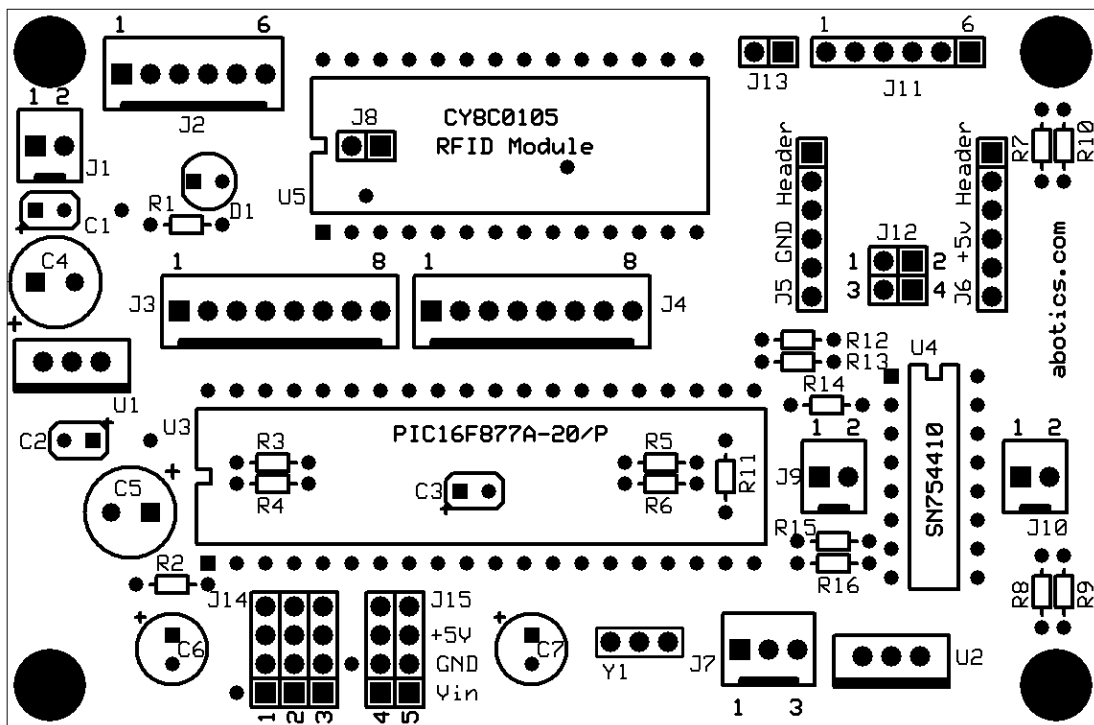
$$R1 = [(R2 * 255 * V_{in}) / (8\text{-bit digital value}) * 5\text{volts}] - R2$$

$$\text{cds cell resistance} = [(4.7\text{kohms} * 255) / (8\text{-bit digital value})] - 4.7\text{kohms}$$

It is possible to convert this resistance value back to a reading in units of illuminance (lux), but I will leave that up to you. The cds cell data sheet can be downloaded from the following location.

http://www.abotics.com/OPEN_ROBOT_DOCS/202403-PhotoCell.pdf

Chapter 11: Design of PIC18F4520 controller board



Figure#104. Main PIC18F4520 controller board.

Originally the board was designed around a PIC16F877A. However, later on it was discovered that the PIC18F4520 is a direct drop-in replacement and is more powerful. From my years of experience designing and building robots I had compiled a list of requirements that a flexible autonomous mobile robot's controller board should satisfy.

List of board requirements

1. Contain a powerful, low-cost, microprocessor that is capable of handling A/D, I2C, serial, and digital sensors.
2. Contain enough object detection sensors such that the robot is able to sense objects around its body perimeter.
3. Contain sensors that can detect and measure the amount of ambient light.
4. Contain a low-cost RFID module and antenna so that the robot can interact with and collect specific information that is embedded within the environment.
5. Contain the necessary motor driver circuitry such as an h-bridge.

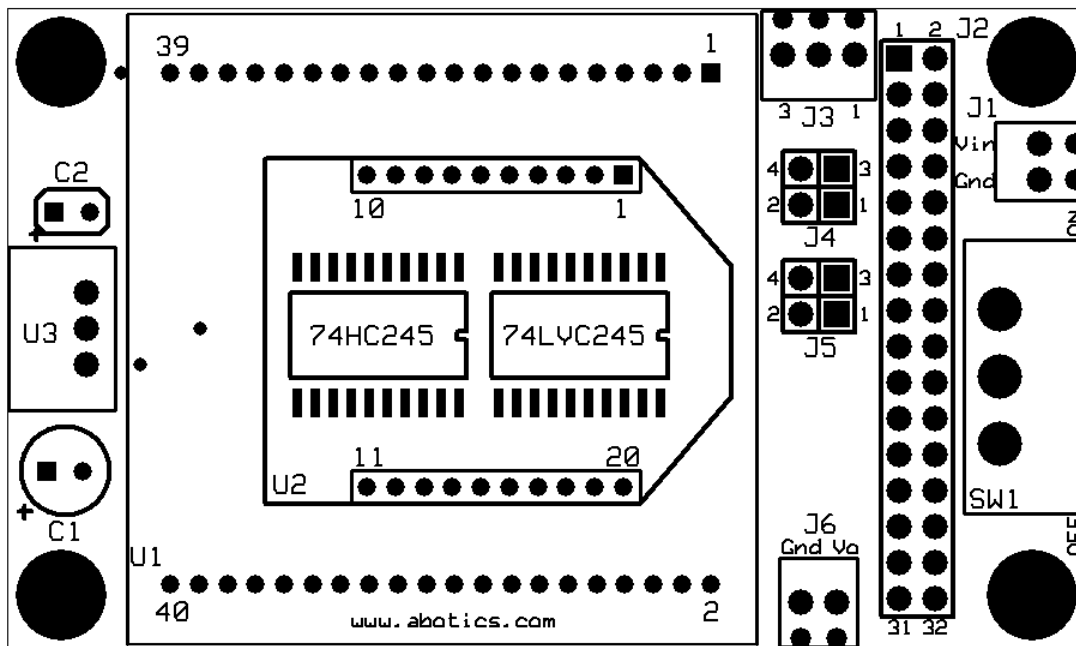
6. Provide a programming and debugging connection point to the chosen microprocessor.
7. Provide the ability to upgrade and add new sensors.
8. Provide connector access to all pins on the chosen microcontroller.
9. Contain the required circuitry to measure its own battery voltage level.
10. Board should be designed using freely available software so that end-users can modify the design to suit their needs in the future.

The above-list of (10)-requirements is what drove the design of OPEN-ROBOT's main controller board.

ExpressPCB board services

The main PIC18F4520 controller board was designed using freely available software from ExpressPCB. You can download this software from <http://www.expresspcb.com/> and make design changes to the board as you see fit. To have the boards manufactured I recommend using ExpressPCB's MiniBoard service. This service will cost you around \$60 (as of 1/02/09) and includes (3)-boards along with shipping. However, you must leave the board size at 3.8"x2.5". You can quickly and economically test out your changes using this service and when you're ready to get a production quality board you can use their ProtoPro service, which includes (4)-boards with silkscreen and solder mask. The schematic file for this board is located on our website, but can also be viewed at the end of this document in the Appendix section.

Chapter 12: Design of MatchPort/XBee wireless board



Figure#105. MatchPort/XBee wireless board.

In the beginning, OPEN-ROBOT made use of a low-cost wireless Bluetooth® module from SparkFun Electronics. This module provided an easy to use wireless connection that leveraged the Bluetooth Serial Port protocol. This worked well for some time; however, later on it was decided that OPEN-ROBOT needed to be more forward compatible with the developing technologies. From this evolved the current MatchPort/XBee wireless board. After reviewing the current state of the art in wireless technologies and low-cost vision systems the following requirements list for the new wireless board was formulated.

List of board requirements

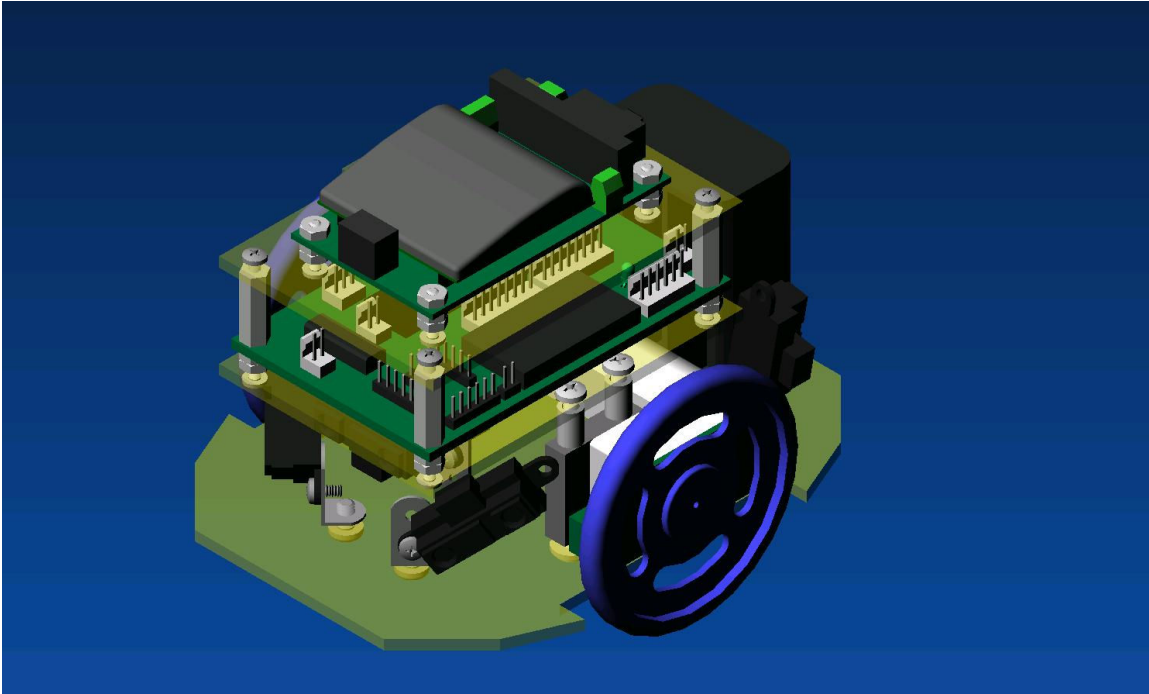
1. Must accommodate the WiFi MatchPort b/g and XBee module.
 - a. Users can choose whether to use MatchPort b/g or XBee module.
2. Must provide an on/off switch to control battery power for the entire robot.
3. Must provide any necessary level shifting circuitry to interface the wireless modules with the PIC18F4520's UART.
4. Must provide an efficient +3.3volt voltage regulator that will supply power to the MatchPort and XBee module.

5. Must provide the necessary hardware to properly interface the SRV-1 Blackfin Camera board from Surveyor Corporation.
 - a. The board must function for an SRV-1 and non-SRV-1 equipped robot.

ExpressPCB board services

From the above (5)-requirements the MatchPort/XBee wireless board was designed using ExpressPCB's free software. You can download this software from <http://www.expresspcb.com/> and make design changes to the board as you see fit. To have the boards manufactured I recommend using ExpressPCB's MiniBoard service. This service will cost you around \$60 (as of 1/02/09) and includes (3)-boards along with shipping. However, you must leave the board size at 3.8"x2.5". You can quickly and economically test out your changes using this service and when you're ready to get a production quality board you can use their ProtoPro service, which includes (4)-boards with silkscreen and solder mask. The schematic file for this board is located on our website, but can also be viewed at the end of this document in the Appendix section.

Chapter 13: Mechanical design of OPEN-ROBOT



Figure#106. OPEN-ROBOT Alibre module shown with MatchPort b/g.

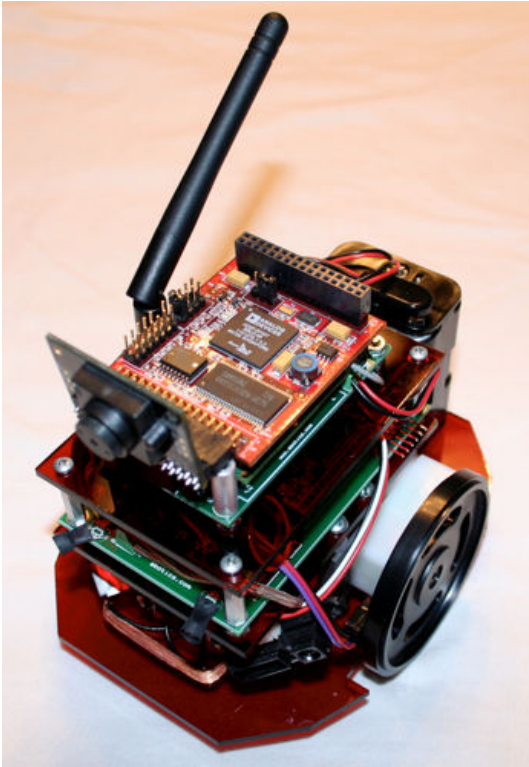
3D Modeling software

OPEN-ROBOT's mechanical design was created using a low-cost 3D modeling program named Alibre. If you are only interested in modifying a few of OPEN-ROBOT's parts, then you might be able to use Alibre Design Xpress, which is a completely free version of their Alibre Standard design software. However, you can only build assemblies that contain up to (6)-different parts. You will not be able to open the entire OPEN-ROBOT model using Xpress, but you can open any of the part files.

https://www.alibre.com/products/xpress/xpress_for_all.asp

There are two OPEN-ROBOT models: one with the MatchPort b/g and one with the XBee module. Every component has been modeled as a part and then constrained to the final assembly. Feel free to redesign and/or add components to OPEN-ROBOT because this is why we are providing the entire 3D model. A zip file containing the 3D model files can be downloaded from http://www.abotics.com/open_robot.htm

Chapter 14: Adding the SRV-1 Blackfin Camera



Figure#107. OPEN-ROBOT shown with SRV-1 Camera.

Preparation for the SRV-1 Camera Board

The SRV-1 camera from Surveyor Corporation can be added to OPEN-ROBOT when the 32-pin connector is soldered to the MatchPort/XBee wireless board as shown in Chapter#2. You will also need to install the jumpers on pins 1&3 on both J4 and J5 on the MatchPort/XBee wireless board (refer to Chapter#2). After completing the above steps, you can simply plug the SRV-1 camera board into the 32-pin connector. To prevent the SRV-1 camera board from moving around you may want to purchase a pair of #4-40 x 0.625" long threaded standoffs from Jameco Electronics (*Jameco Part# 77519*), www.jameco.com. To install the standoffs simply remove the two #4-40 nuts on the front of the MatchPort/XBee wireless board and then thread on the standoffs.

Chapter 15: Upload Modified SRV-1 Firmware

To be able to use the SRV-1 camera with OPEN-ROBOT you either need to know how to modify the SRV-1 firmware yourself or download suitable firmware that has already been modified to work with the SRV-1 and OPEN-ROBOT. To start out we suggest downloading already modified firmware.

Prerequisites

Download the appropriate OPEN-ROBOT modified SRV1 Firmware from Abe Howell's Robotics. You must match the modified firmware baud rate with the current firmware that is installed on your SRV1. For example, if your SRV1 was purchased with a firmware running 115,200 bps then you need to use the `srv1.ldr.115k` firmware. These files were built from a 10-19-08 SRV1 firmware and will not work with earlier versions.

http://www.abotics.com/OPEN_ROBOT_Software/srv1_openrobot_firmware.zip

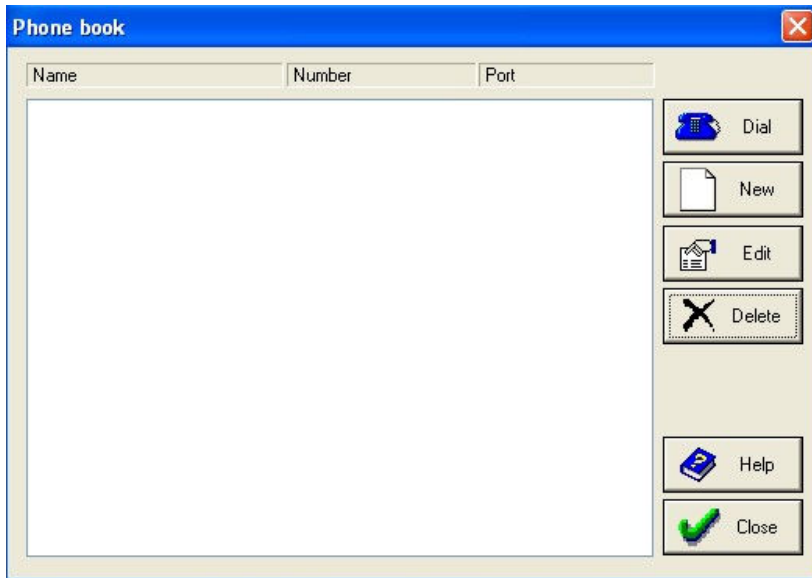
After downloading the firmware zip file, go ahead and unzip the contents to a known location. If you are running Windows® XP I would suggest that you download a suitable Terminal program like Tera Term Pro or ShamCom. I will be using ShamCom in this tutorial.

<http://www.shamrock.de/tools.htm>

<http://hp.vector.co.jp/authors/VA002416/teraterm.html>

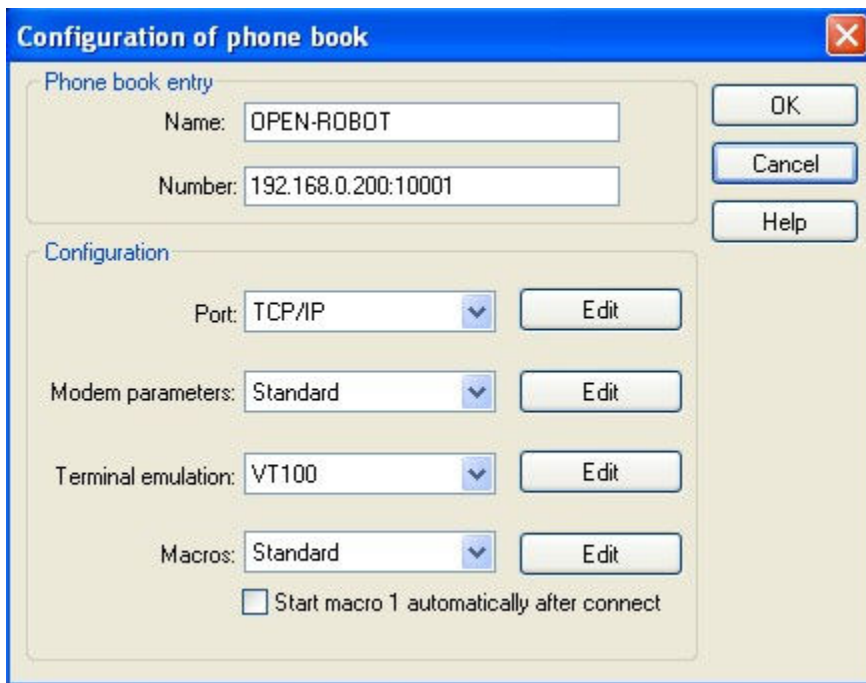
Uploading Firmware Using ShamCom Terminal and XModem

The SRV1 firmware has to be uploaded using XModem and the '**X**' SRV1 command. Once the firmware has been uploaded to the flash buffer then it needs to be written to the boot sectors of flash memory using the '**ZZ**' SRV1 command. First you need to go ahead and connect to your SRV1. Run the ShamCom program and select Dial from the Connection menu.



Figure#108. ShamCom New entry.

Then press the **New** button to create a new entry.



Figure#109. Add new entry for OPEN-ROBOT.

I will be naming my entry with OPEN-ROBOT, but you can name yours anything. You need to enter the IP Address of your robot's MatchPort b/g in the **Number:** field. Don't forget to add **:10001**, which is the MatchPort b/g port number. There is no need to change the other settings, so simply click the **OK** button.



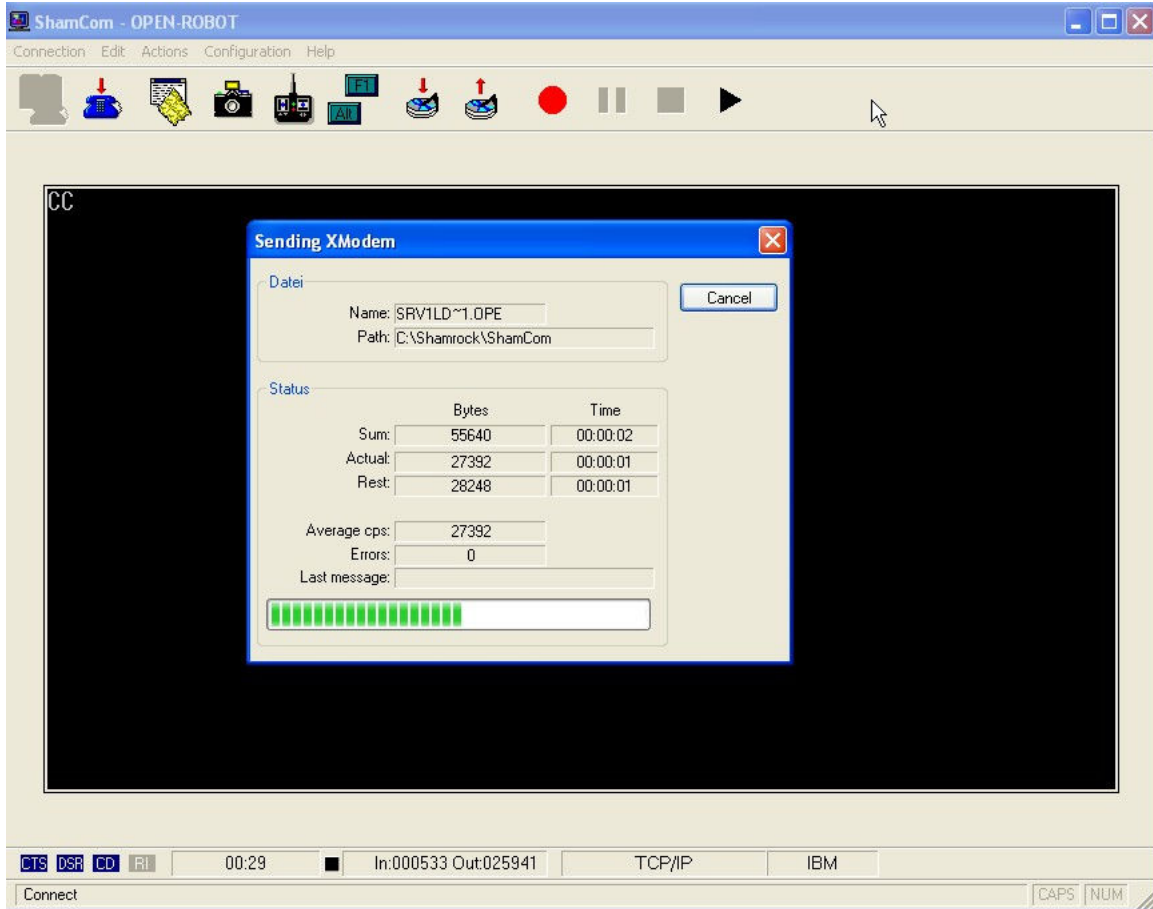
Figure#110. Save configuration

Be sure to press **Yes** when you are asked to save the entry.



Figure#111. Connect to OPEN-ROBOT.

Now you can go ahead and power up your robot and click on the **Dial** button. The ShamCom program will now attempt to connect to your robot. At this point I would suggest copying the appropriate `srv1.ldr` file to the ShamCom folder, which should be located at **C:\Shamrock\ShamCom**. This way the file will be readily accessible for when you upload using XModem. Go ahead and type the SRV1 XModem receive command, **X**. You should start to see CCCC printed in the ShamCom window. This means that the SRV1 is waiting to receive a file upload via XModem. Click on the Action menu and select **Upload** then **XModem...** Quickly browse for the correct SRV1 firmware file, which should be any of the following: **srv1.ldr.115k**, **srv1.ldr.921k**, or **srv1.ldr.2500k**. The file upload should only take a minute or so.



Figure#112. New firmware upload in process.

Once the upload is complete, you need to write the flash buffer to boot sectors of flash memory using the **zZ** SRV1 command. Type the **zZ** command in the ShamCom window and you should receive a message similar to the one shown below. Your new firmware has been loaded.



Figure#113. New firmware upload complete.

You will now be able to command the SRV1 camera board, but also send and receive commands from your OPEN-ROBOT. Now all your OPEN-ROBOT commands have to be preceded by the tilde (~) character. This lets the SRV1 firmware know that the following command is for the OPEN-ROBOT. For example, to retrieve the analog-to-digital sensor readings we would type the following: **~A** and then press the Return or

Enter button. To command OPEN-ROBOT to drive forward using open-loop velocity we would type the following: **~MO25,25** and then press the Return or Enter key.

Chapter 16: Upload srv1.ldr Using Flashload & ICEbear

If you decided to purchase the ICEbear USB JTAG, then this section will help to provide a brief explanation with regard to uploading new loader files to the SRV-1 camera board.

Prerequisites

To be able to complete this process you must have ICEbear and Cygwin installed on your Microsoft® Windows machine. I have only verified this running Windows XP. You can obtain ICEbear from the Section5 website. Be sure to download the Win32/Cygwin package. At the time of writing this I was using Version 1.2. This installer will provide you with the flashload.exe and other necessary files. You must have the ICEbear JTAG.

<http://www.section5.ch/software>

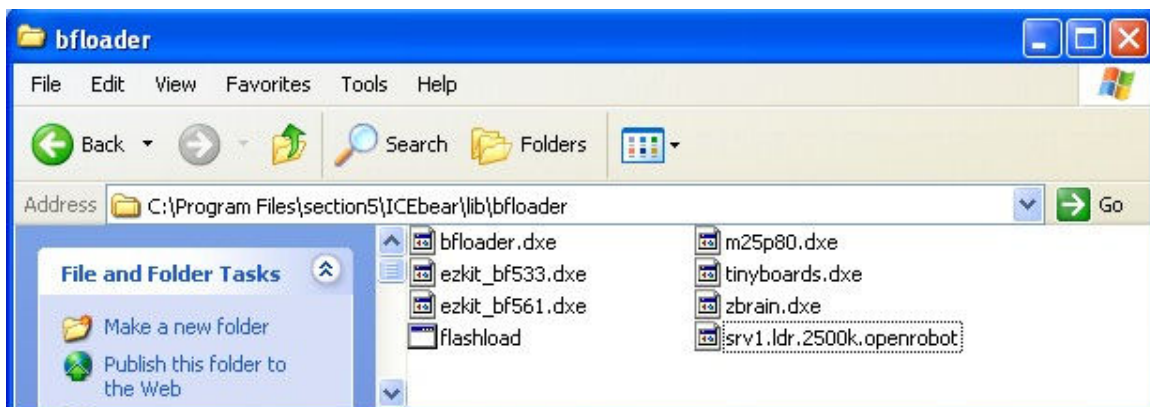
The Cygwin environment can be downloaded from the following website.

<http://www.cygwin.com/>

Flashload srv1.ldr File Using ICEbear JTAG

I like to copy my srv1.ldr file into the bfloder directory, which can be found under the ICEbear install directory. This folder also has the necessary backend file, [m25p80.dxe](#), for uploading the srv1.ldr file using flashload.exe. For this example, the name of my srv1.ldr file is [srv1.ldr.2500k.openrobot](#).

<C:\Program Files\section5\ICEbear\lib\bfloder>



Figure#114. ICEbear bfloder directory.

Now plug in your ICEbear to the SRV-1 board on the 14-pin connector, J2, and then attach the USB cable. You should notice that one of the pins on J2 has been cut off. You should also notice that one of the pins on your ICEbear has been plugged. This is a way

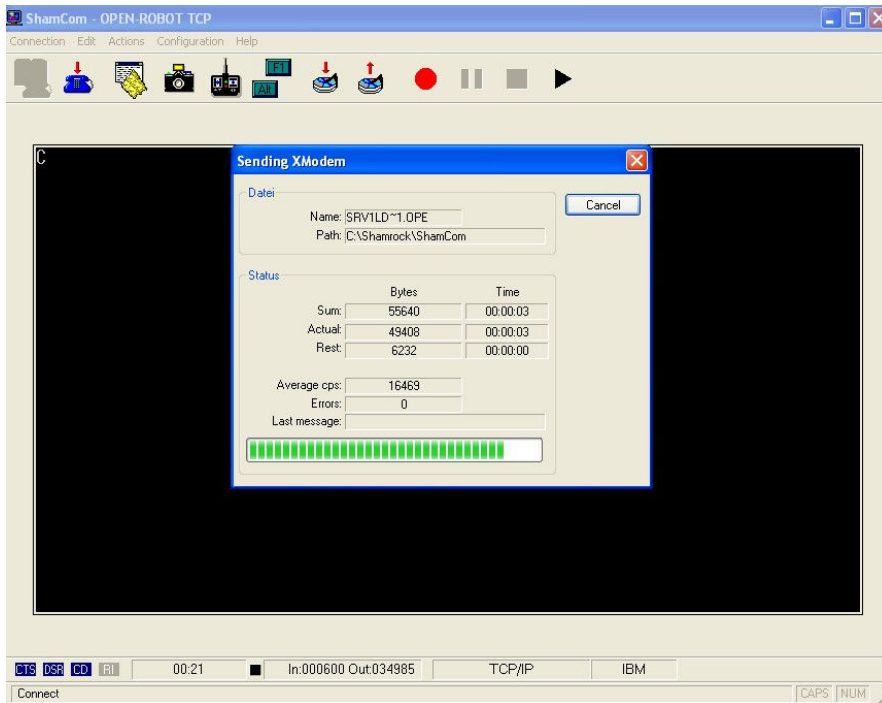
to polarize the ICEbear and J2 so that you don't improperly connect them. Make sure the robot is powered off when plugging in the ICEbear. Open a command console and navigate to the above-listed Cygwin directory. Power up your robot and type the following command into the command console.

`Flashload - -eraseall - -backend=m25p80.dxe - -program=srv1.ldr`

```
C:\Program Files\section5\ICEbear\lib\bfloder>flashload --eraseall --backend=m25p80.dxe --program=srv1.ldr.2500k.openrobot
Number of devices in JTAG chain: 1
JTAG wait cycles: 1
Detecting device...
Detected Device: BF-537(536) rev:2
emulation_enter(): CPU was not in normal state [3]
-----
Verify is on (default).
-----
Erasing entire Flash chip, please wait. This can take minutes to complete.
Do not interrupt.
Done.
-----
Writing Flash image from file 'srv1.ldr.2500k.openrobot'
Writing 5376 bytes to flash at offset $00000000
Writing 5376 bytes to flash at offset $00001500
Writing 5376 bytes to flash at offset $00002a00
Writing 5376 bytes to flash at offset $00003f00
Writing 5376 bytes to flash at offset $00005400
Writing 5376 bytes to flash at offset $00006900
Writing 5376 bytes to flash at offset $00007e00
Writing 5376 bytes to flash at offset $00009300
Writing 5376 bytes to flash at offset $0000a800
Writing 5376 bytes to flash at offset $0000bd00
Writing 1880 bytes to flash at offset $0000d200
C:\Program Files\section5\ICEbear\lib\bfloder>
```

Figure#115. ICEbear flashload.exe erasing flash and then programming chip.

The ICEbear will begin to erase the flash chip and then start the process of programming. I have had this process freeze on me several times and the only way I know how to resolve it is to simply power down the robot and try again.



Figure#116. ShamCom uploading srv1.ldr file using XModem.

Upload srv1.ldr File Using XModem

Once the flashload process is complete, you must then connect to the SRV1 and upload the same srv1.ldr file using XModem as shown above in figure#116. I use ShamCom to accomplish this task because it is a free program and it works well. You can download ShamCom from the following website.

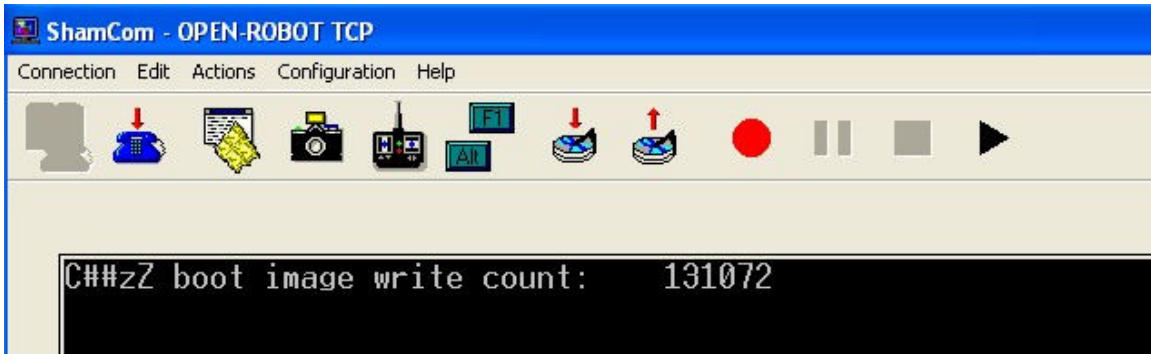
<http://www.shamrock.de/tools.htm>

If you are going from a srv1.ldr with a different baud rate (i.e 921,600 bps to 2500k bps) then you will need to adjust the baud rate of your MatchPort b/g before proceeding. **DO NOT** turn off your robot, otherwise you will have to start over.

Once connected to the SRV1 using ShamCom you need to enter the **X** command to set the SRV1 to accept an XModem file transfer. From the ShamCom menu select Actions->Upload->XModem and then browse to find the same srv1.ldr file that you loaded using flashload. I prefer to copy the srv1.ldr file to the ShamCom folder.

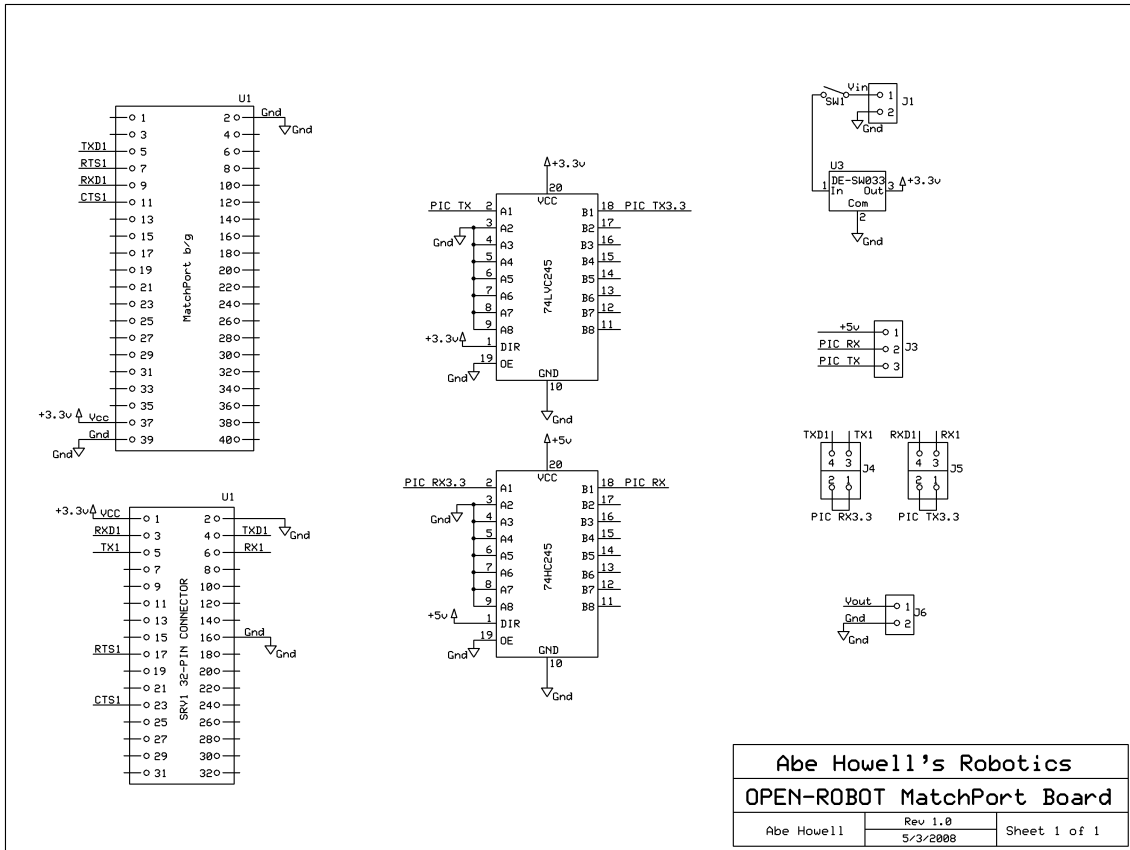
C:\Shamrock\ShamCom

After the file has been transferred to the SRV-1 go ahead and save using the **zZ** command. If this is successful you should see the same as is shown in figure#117 below. You are now safe to reboot your robot and your new srv1.ldr should be in use.



Figure#117. Successful zZ command.

MatchPort/XBee wireless board schematic



PIC18F4520 I/O Map

| OPEN-ROBOT PIC18F4520 Board | | | |
|-----------------------------|------|-----|--------------|
| Master Clear/Reset | MCLR | RB7 | SPARE_IO/PGD |
| Battery Level | RA0 | RB6 | SPARE_IO/PGC |
| Front Center IR | RA1 | RB5 | SPARE_IO |
| Front Left IR | RA2 | RB4 | SPARE_IO |
| Front Right IR | RA3 | RB3 | SPARE_IO/PGM |
| ENC_R_CLK | RA4 | RB2 | SPARE_IO |
| Rear Left IR | RA5 | RB1 | SPARE_IO |
| Rear Right IR | RE0 | RB0 | ENC_L_CLK |
| Front Right Light | RE1 | Vdd | 5 volts |
| Front Left Light | RE2 | Vss | GND |
| 5 volts | Vdd | RD7 | ENC_R_DIR |
| GND | Vss | RD6 | ENC_L_DIR |
| 20MHz Resonator | OSC1 | RD5 | RFID_RX |
| 20MHz Resonator | OSC2 | RD4 | RFID_TX |
| SPARE_IO | RC0 | RC7 | UART_RX |
| L_GM8_PWM (1,2 EN) | RC1 | RC6 | UART_TX |
| R_GM8_PWM (3,4 EN) | RC2 | RC5 | RESET_RFID |
| SCK_I2C | RC3 | RC4 | SDI_I2C |
| L_GM8 (1A) | RD0 | RD3 | R_GM8 (4A) |
| L_GM8 (2A) | RD1 | RD2 | R_GM8 (3A) |

| SN754410 H-Bridge Chip | | | |
|------------------------|----|-----|------------|
| 1,2 EN | P1 | P16 | Vcc1 (+5v) |
| 1A | P2 | P15 | 4A |
| 1Y (L_GM8) | P3 | P14 | 4Y (R_GM8) |
| GND | P4 | P13 | GND |
| GND | P5 | P12 | GND |
| 2Y (L_GM8) | P6 | P11 | 3Y (R_GM8) |
| 2A | P7 | P10 | 3A |
| Vcc2 (+5v) | P8 | P9 | 3,4 EN |

OPEN-ROBOT Parts List

| OPEN-ROBOT w/ GM8s and MatchPort b/g | | | | | |
|---|--------------------------|---|---------|---------|--|
| QTY | Part# | Description | Price | Total | Where? |
| 1 | GC-MATCHPORT-G-01 | MatchPort b/g Module | \$75.00 | \$75.00 | www.gridconnect.com |
| 1 | GC-WIIPORT-ANT | MatchPort Rubber Antenna | \$7.00 | \$7.00 | www.gridconnect.com |
| 1 | GC-MATCHPORT-CAB-UFL-SMA | Matchport cable U.FL to Reverse SMA | \$3.90 | \$3.90 | www.gridconnect.com |
| 1 | MatchPort/SRV1-PCB | MatchPort/XBee Wireless Board | \$34.35 | \$34.35 | www.abotics.com |
| 1 | 66 | Tamiya Ball Caster Pair (\$5.95) | \$2.98 | \$2.98 | www.pololu.com |
| 2 | R246-GM8 | GM8 Gearmotor | \$6.00 | \$12.00 | www.acroname.com |
| 2 | R249-BLACK-GM-WHEEL | GM8 Wheel | \$3.50 | \$7.00 | www.acroname.com |
| 2 | | GP2D120 Mounts | \$0.50 | \$1.00 | www.abotics.com |
| 2 | IR_Cable | Sharp GP2D120 JST Cable | \$2.99 | \$5.98 | www.roboticsconnection.com |
| 2 | GP2D120 | Sharp GP2D120 | \$8.99 | \$17.98 | www.roboticsconnection.com |
| 4 | 133605 | 3/4"x3/16" Female-Female Threaded 4-40 Hex Standoff | \$0.39 | \$1.56 | www.jameco.com |
| 2 | | GM8 Motor mount (100+) | \$6.25 | \$12.50 | www.emachineshop.com |
| 1 | | Robot Base, 1/8" Acrylic | \$5.50 | \$5.50 | www.pololu.com |
| 1 | | PCB Deck, 1/8" Acrylic | \$3.50 | \$3.50 | www.pololu.com |
| 1 | | MatchPort Deck | \$3.50 | \$3.50 | www.mcmastercar.com |
| 2 | | 8-32 x 0.25" Truss Screw | \$0.05 | \$0.10 | www.jameco.com |
| 8 | 106810 | 4-40 x 1/2" Pan Head Machine Screw | \$0.04 | \$0.34 | www.jameco.com |
| 9 | 40970 | 4-40 x 3/8" Pan Head Machine Screw | \$0.02 | \$0.16 | www.jameco.com |
| 22 | 40943 | 4-40 Nut (100 for \$2.50) | \$0.04 | \$0.77 | www.jameco.com |
| 1 | | OPEN-ROBOT Controller Board | \$28.20 | \$28.20 | www.abotics.com |
| 4 | | 3M Dual-Lock Recloseable Tape (units of inches) | \$0.25 | \$1.00 | www.fastenation.com |

Complete OPEN-ROBOT + MatchPort + RFID \$242.32
Estimated Shipping Charges from: Jameco, Mouser, GridConnect, etc... \$50.00
Approximate Total Cost \$292.32

****NOTE: Some Items Require you to purchase more than is required
This price increase is not reflected in the Approximate Total!
Prices may have changed!**

OPEN-ROBOT Controller Board Parts List

| Custom PCB Pricing from ExpressPCB | |
|------------------------------------|---------|
| Number of Boards Purchased | 100 |
| Production Service | 473.575 |
| Price Per Board | 4.73575 |

| OPEN-ROBOT Board (JAMECO) | | | | | |
|---------------------------|--------------------|--|--------|--------|--|
| QTY | Part# | Description | | | Where? |
| 1 | 579-PIC18F4520-I/P | PIC18F4520 40-Pin Chip | 10.010 | 10.010 | www.mouser.com |
| 1 | 324671 | 20 MHz Ceramic Resonator w/capacitors(100) | 0.279 | 0.279 | www.jameco.com |
| 1 | 1054684 | SN754410NE 16-Dip H-Bridge (100) | 1.070 | 1.070 | www.jameco.com |
| 0 | CY8C0105-B5 | RFID Module (10+) | 10.600 | 0.000 | www.sonmicro.com |
| 0 | Q5(T5555) | Q5 ISO T5555 Tag (10+) | 3.000 | 0.000 | www.sonmicro.com |
| 0 | IND-125 | Antennae(11+) | 2.900 | 0.000 | www.sonmicro.com |
| 0 | PRT-08614 | RFID Antenna Cable | 1.500 | 0.000 | www.sparkfun.com |
| 1 | PCB | Custom PCB From Express PCB (100) | 4.736 | 4.736 | www.expresspcb.com |
| 2 | 651-1725656 | 2-Pos Terminal Block | 0.970 | 1.940 | www.mouser.com |
| 1 | 108338 | 2 Pos Non-Polarized Header | 0.200 | 0.200 | www.jameco.com |
| 5 | 117560 | 4-Pos Non-Polarized Header | 0.210 | 1.050 | www.jameco.com |
| 1 | 153699 | 6-Pos Non-Polarized Right Angle Header | 0.25 | 0.25 | www.jameco.com |
| 1 | 253681 | T-100 Green LED | 0.260 | 0.260 | www.jameco.com |
| 3 | 33486 | 0.1 uF Tantalum Capacitor, 35volt | 0.230 | 0.690 | www.jameco.com |
| 1 | 41111 | 40 Pin IC Socket (10+) | 0.250 | 0.250 | www.jameco.com |
| 1 | 40301 | 28 Pin IC Socket (0.6" w/0.1" pin to pin) (10) | 0.171 | 0.171 | www.jameco.com |
| 1 | 37372 | 16 Pin IC Socket (10) | 0.090 | 0.090 | www.jameco.com |
| 2 | 192313 | LM1086CT-5.0 +5volt 1.5 Amp Regulator | 1.490 | 2.980 | www.jameco.com |
| 1 | 216215 | (6)- AA Holder | 0.890 | 0.890 | www.jameco.com |
| 1 | 108206 | 47k Ohm resistor (100) | 0.007 | 0.007 | www.jameco.com |
| 5 | 108089 | 4.7k Ohm resistor (100) | 0.007 | 0.035 | www.jameco.com |
| 1 | 108062 | 2.7k Ohm resistor (100) | 0.007 | 0.007 | www.jameco.com |
| 7 | 108011 | 1k Ohm resistor (100) | 0.007 | 0.048 | www.jameco.com |
| 2 | 202403 | 3k - 200k Ohm Photocell | 1.490 | 2.980 | www.jameco.com |
| 2 | 29964 | 100 uf radial capacitor (10) | 0.078 | 0.156 | www.jameco.com |
| 2 | 93745 | 33uf radial capacitor (10) | 0.050 | 0.100 | www.jameco.com |

| | |
|--------------------------|----------------|
| Total without PCB | \$23.46 |
| Total with PCB | \$28.20 |

MatchPort/XBee/SRV-1 Board Parts List

MatchPort/XBee Wireless Board

| QTY | Part# | Description | Price | Total | Where? |
|-----|--------------------|---|---------|---------|-----------------------|
| 4 | PRT-08272 | 2mm 10-Pin Sockets | \$1.25 | \$5.00 | SparkFun |
| 1 | | PCB from Express PCB (100+) | \$3.78 | \$3.78 | PCB Express |
| 1 | 204142 | SWITCH,MINI SLIDE,SPDT (10) | \$0.19 | \$0.19 | Jameco |
| 4 | 538-22-03-2021 | 0.1" 2-position non-polarized header | \$0.12 | \$0.48 | Mouser |
| 2 | 151-8010-E | 0.1" jumper | \$0.12 | \$0.24 | Mouser |
| 2 | 651-1725656 | 2-Pos Terminal Block | \$1.03 | \$2.06 | Mouser |
| 1 | 651-1725669 | 3-Pos Terminal Block | \$1.57 | \$1.57 | Mouser |
| 1 | 595-SN74HC245DWR | SN74HC245 | \$0.44 | \$0.44 | Mouser |
| 1 | 595-SN74LVC245ADWR | SN74LVC245 | \$0.44 | \$0.44 | Mouser |
| 1 | DE-SW033 | SWITCHING REGULATOR, 1.0 AMP, SIP 3, 3.3V (20+) | \$15.00 | \$15.00 | Dimension Engineering |
| 1 | ESQ-116-23-T-D | Samtec 32-Pin Connector | \$5.00 | \$5.00 | abotics |
| 5 | | 6" of 24 Gauge Wire | \$0.03 | \$0.15 | Jameco |

Total \$19.35
Total w/DE-SW033 \$34.35