3pi Wireless Robot Set I

Quick Start Guide

Document Revision: 1.02

Date: 23 March 2011

BiPOM Electronics, Inc.

1. Overview

Thank you for purchasing **3pi Wireless Robot Set I** set from BiPOM Electronics.

3pi Wireless Robot Set I is a robotics development set that has everything that you need to get started with wireless robotics. It is a wireless system all in one.

Package Contents

3pi Wireless Robot Set I includes:

- BRD-EXP-PLLU3PI-ZB board
- BRD-USBXB-1 board
- USB Cable A to Mini-B 6feet (1.9meters)
- Two sets of mapped XB24-ACI-001 XBee industry standard Zigbee Mini USB Cable
- 3pi Expansion Set Hardware
    - ü Male connectors are already soldered to BRD-EXP-PLLU3PI-ZB board

Figure 1 shows the package contents (cable not shown):

BRD-USBXB-1 board
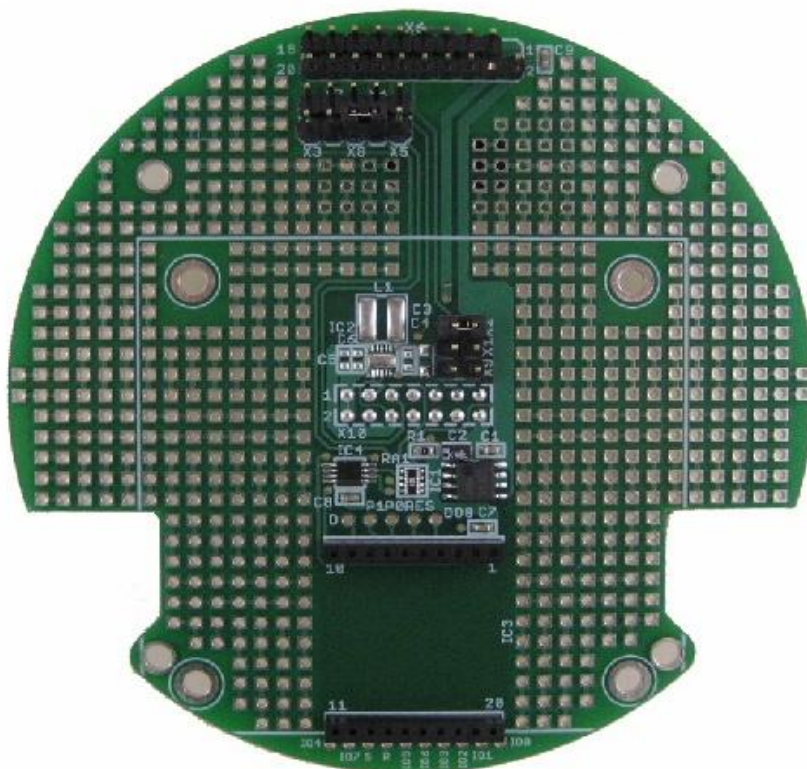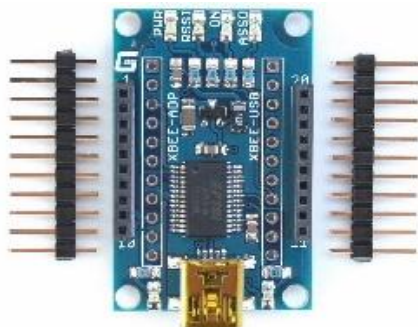
XB24-ACI-001
XBee 802.15.4 OEM RF Modules
Two sets of mapped radio modules for wireless

Figure 1

2. Hardware Setup

Pololu 3pi Robot is not included with **3pi Wireless Robot Set I** set. This set is based on the assumption that you already have your own Pololu 3pi Robot and Pololu USB AVR Programmer.

XBee (Zigbee) radio modules that come with the set are mapped to each other (paired) and factory configured to work at 115200 baud. XBee module configuration and mapping should not be changed by the user to ensure proper operation with the set.
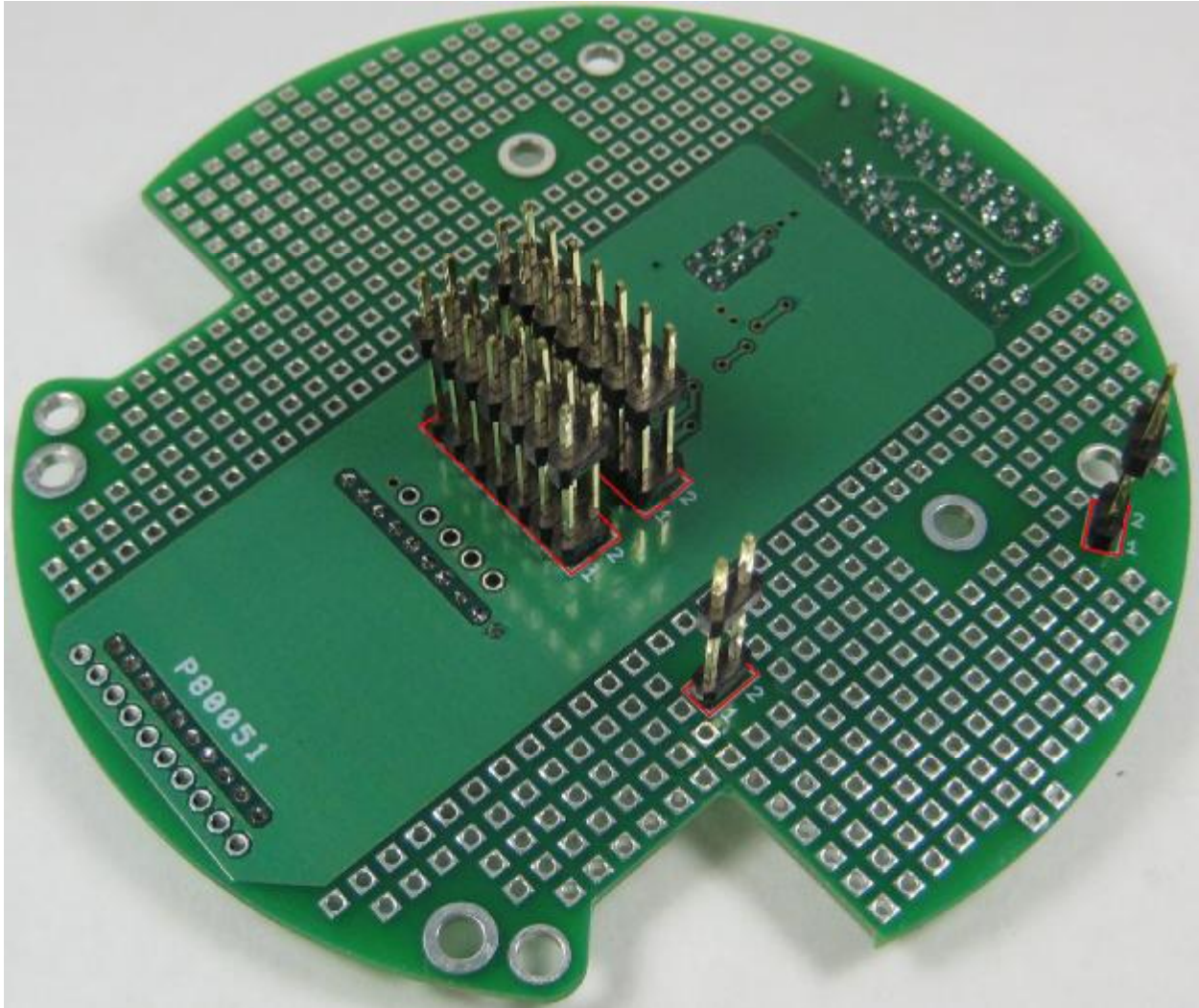
Bottom view of 3pi expansion board rev. 1.05
**(Soldered male connectors shown with red marks).**

Soldering Connectors

The male expansion connectors for the top board are soldered at the factory for the **3pi Wireless Robot Set I** set.

Female connector headers for LCD, CHARGE and PUSH between the robot and the top board are part of the **3pi Wireless Robot Set I** set and can be soldered to the robot by the user if additional functionality is needed on these connectors.

We recommend that you fully assemble the 3pi and its expansion set **before you solder anything**. This will ensure that once everything is soldered in, the expansion platform will align properly with the base. We suggest that you assemble your expansion set in the following order:

1) Place the 2×7 female header and one of the 2×1 female headers into the proper holes in the 3pi base as shown below (see the yellow rectangles).
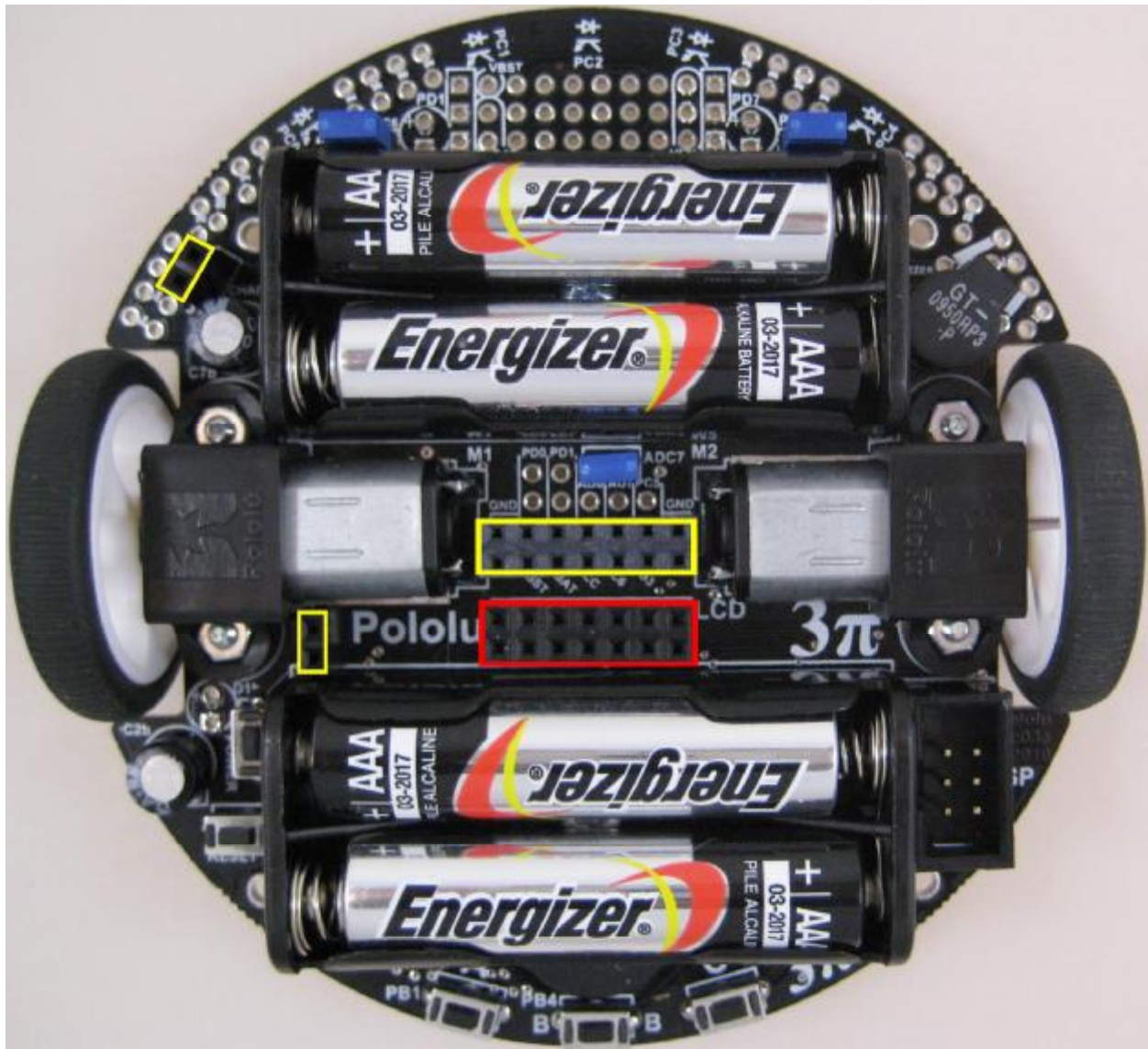
2) Insert the long ends of the 2×7 and one 2×1 extended male header into these female headers, and insert an extended 2×1 male header into the battery charge port. Place the expansion PCB so the tops of these male headers seat in the proper places, as marked by the rectangles in the picture below. Note that the expansion PCB mounts with the silkscreen facing up.
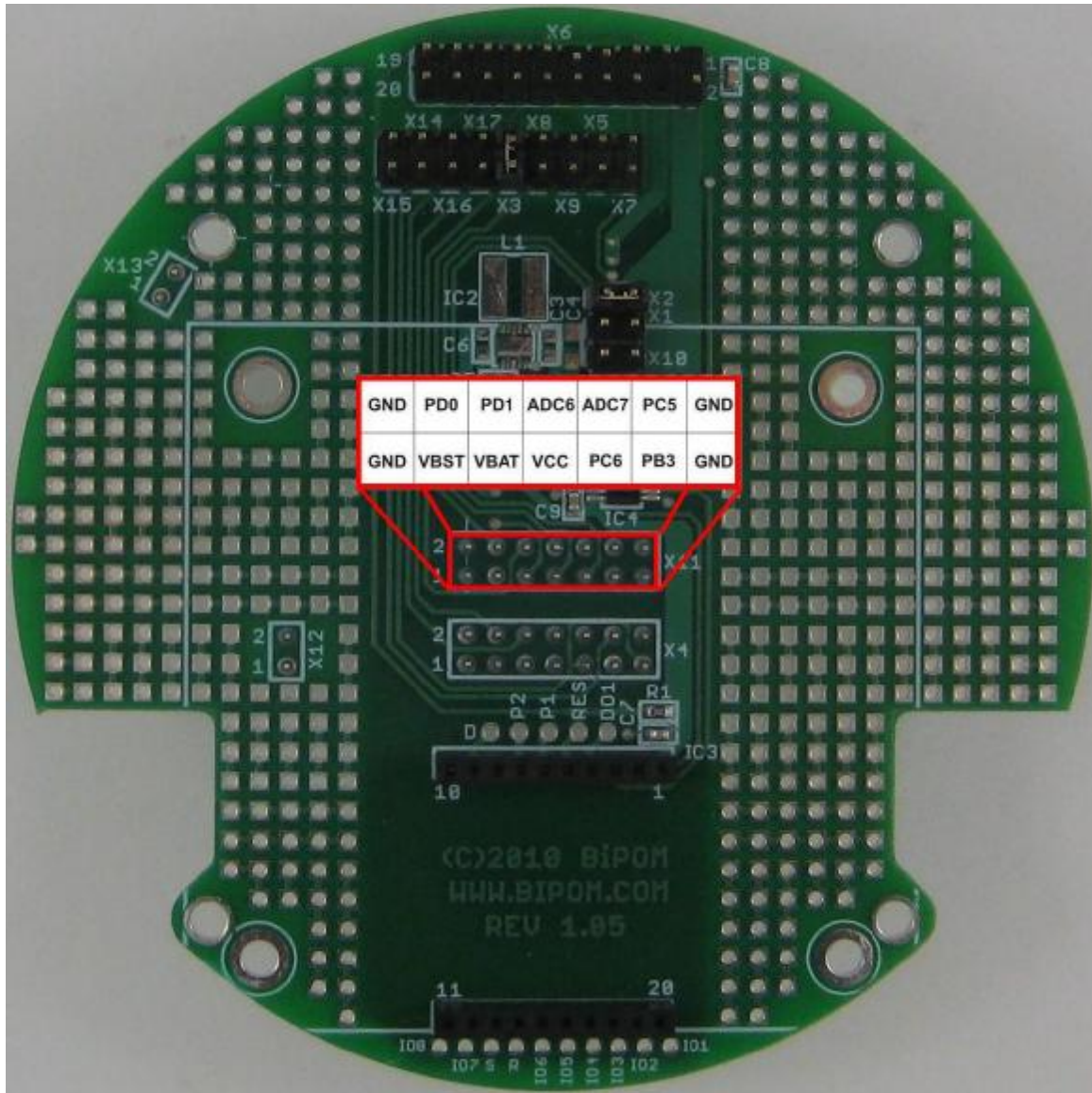
Solder connections for male headers on the 3pi expansion board.

3) Slip a nylon spacer between the base and the expansion PCB so that it lines up with the mounting hole on the base. Insert a screw from the underside of the base up through the base's mounting hole, the spacer, and the mounting hole on the expansion board. Holding the head of the screw against the base, twist the nut onto the other side, but don't tighten it all the way. Repeat this process for the three remaining screws, and then tighten them together so that the expansion PCB is aligned well with the base.

4) With the screws holding everything in place, you can now solder the female headers to the base. Once everything is soldered in, you can remove the screws and pull the expansion PCB off of the base; it should look like the one in the picture below.

- Insert the Zigbee wireless expansion board BRD-EXP-PLLU3PI-ZB on to the robot's female expansion connectors (in yellow rectangles) in Figure 3 below:
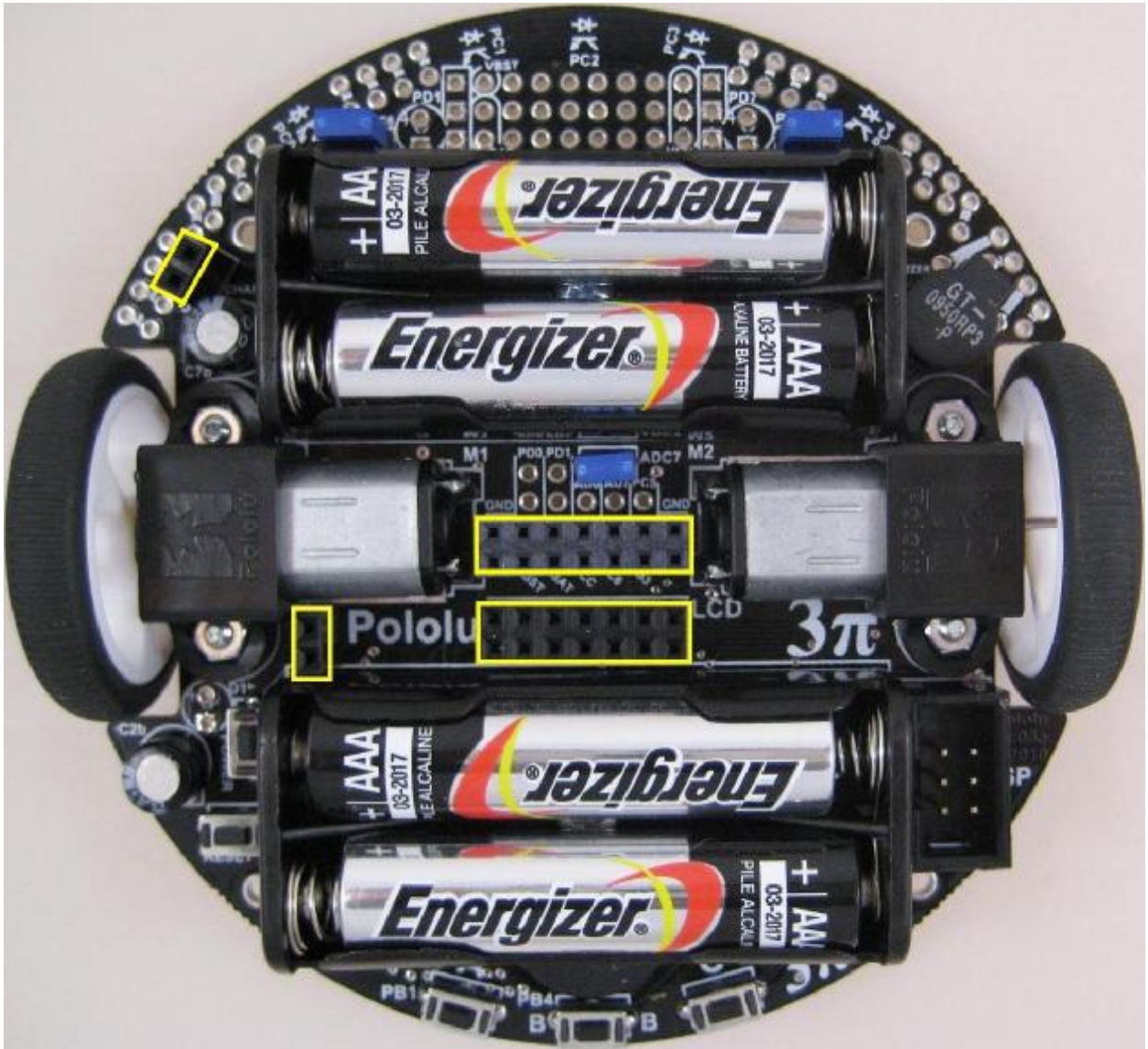

Figure 3

- Complete the hardware set up by adding 4 plastic stand offs and metal bolts as shown in Figure 4 (red circled).
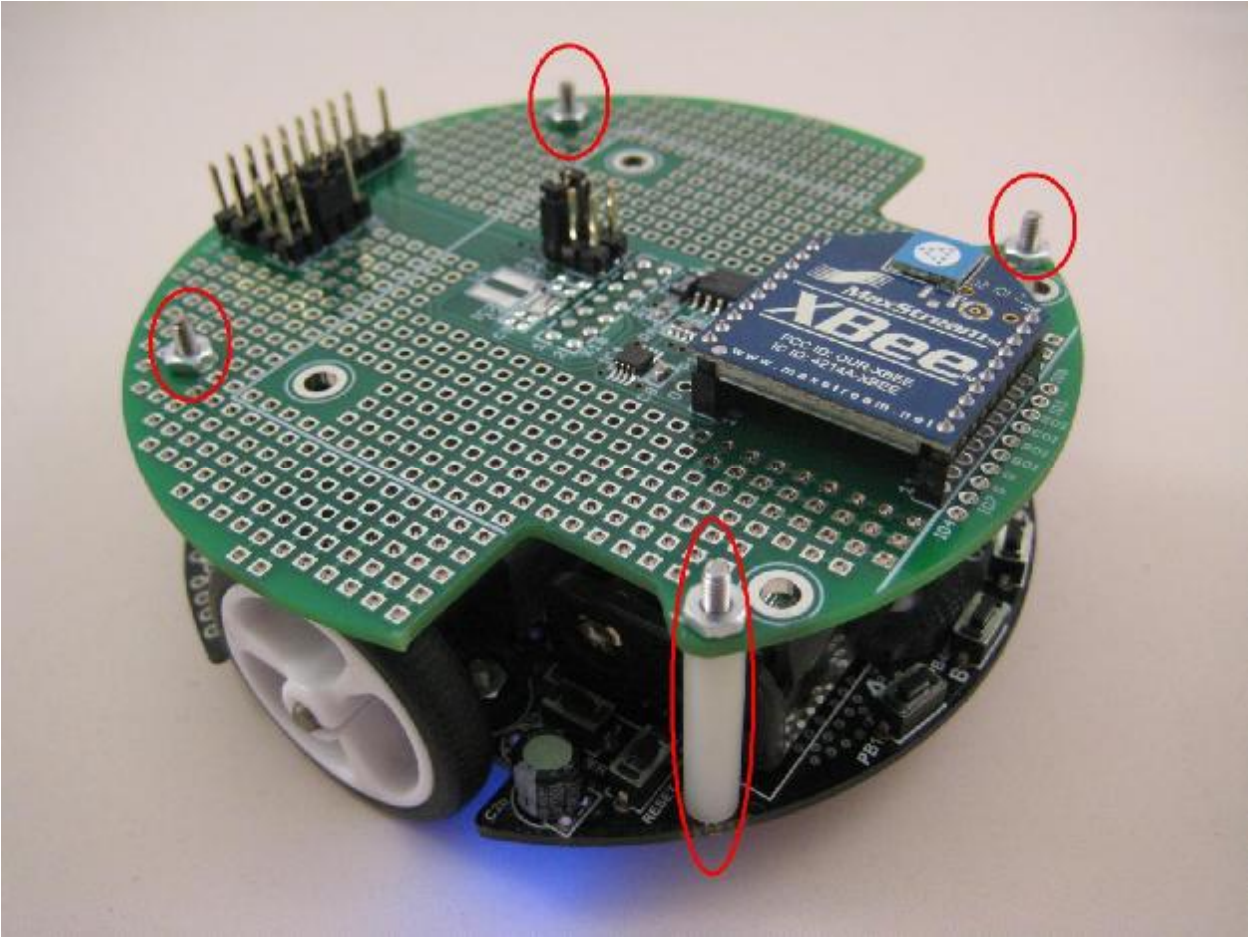

Figure 4

3. Software Setup

3.1 Robot Side

3pi Robot runs a **SLAVE** firmware to respond to requests coming over the wireless link. The 3pi Robot which is not included with **3pi Wireless Robot Set I** set needs to have this firmware downloaded. If you change this firmware and want to restore it at any time, please refer to Appendix A for instructions on how to download the **SLAVE** firmware to the 3pi Robot.

SLAVE firmware supports the following features:

- Send the version of the slave firmware
- Read the line sensors and send their values
- Send the raw (uncalibrated) sensor values
- Send the calibrated sensor values
- Compute the position of a black line using the read_line() function and send the value
- Send the trimmer value
- Send the battery voltage in millivolts
- Drive robot's motors
- Play a musical sequence
- Display messages on robot's LCD display
- Run through an automatic calibration sequence

3.2 PC Side

On the PC side, software setup involves 2 steps:

1. Installation of FTDI driver software for USB dongle for **BRD-USBXB-1** board

2. Installation of 3pi Robot Control Panel

3.2.1 Installation of FTDI Driver

Please download the FTDI chip driver file "CDM20600.exe" from the link below.

http://www.ftdichip.com/Drivers/CDM/CDM20600.exe

Start the installation of this driver by double clicking on "CDM20600.exe". And then follow the on screen instructions to complete the installation.

3.2.2 Installation of 3pi Robot Control Panel

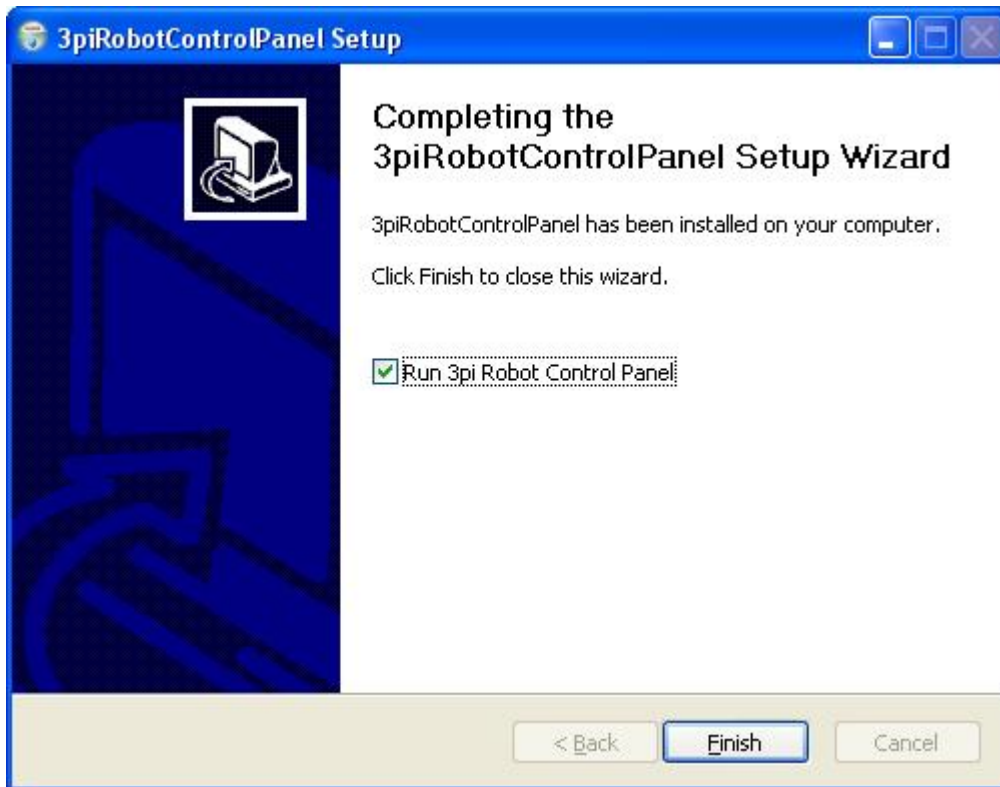"3piRobotControlPanel.exe"can be downloaded from the link below.

http://www.bipom.com/web_softwares/3415761.html

At first, you may see a message window appearing as shown below.
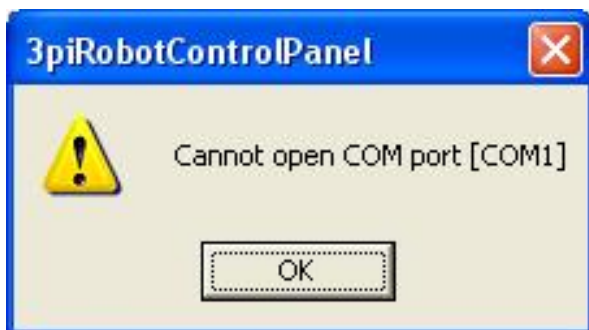

Please click "Run" to start installing the program.

Follow the on screen instructions of setup wizard to complete installation of the program.

Please check the "Run 3pi Robot Control Panel" option and click "Finish" to start the program.

You may see a warning window similar to the one above. This is due to configuration of PC's communication port. Please click "OK" to proceed with configuration of the software.
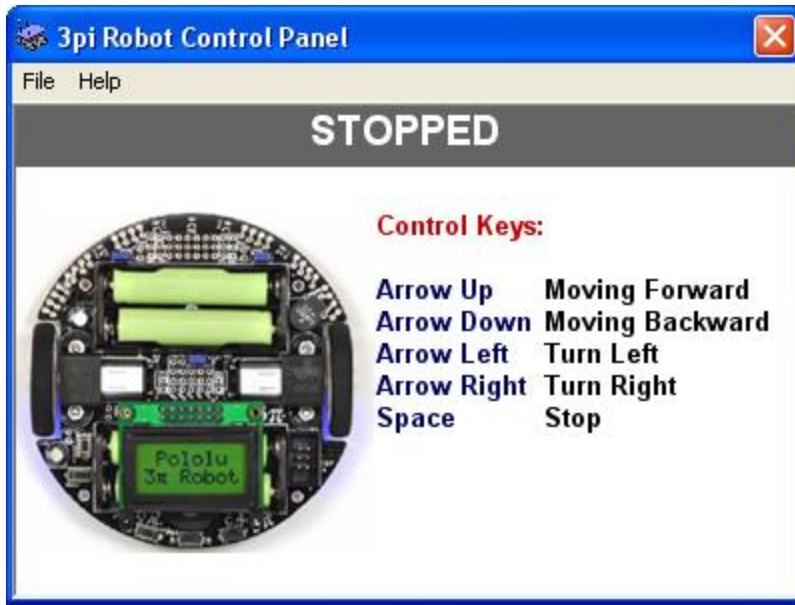
The main window of the software is shown above.

Please select your COM (Communication Port) by choosing File > Select COM Port…

The selection COM port is the same port in which your BRD-USBXB-1 board connected to PC.

Please note that COM ports numbered above COM9 may not work with the program. If this is the case, try choosing a COM port numbered below COM9. This can be done by modification of that COM port within "Device Manager" or physically choosing a different port in your PC.
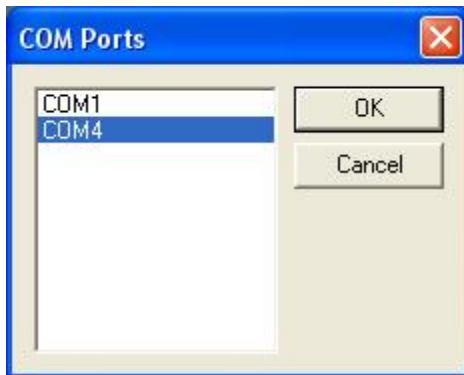
After choosing the correct COM port in the program, 3pi robot can now be turned on by pushing the power button.

3pi Robot can now be remotely controlled via this program by using arrow and space bar buttons of your keyboard.
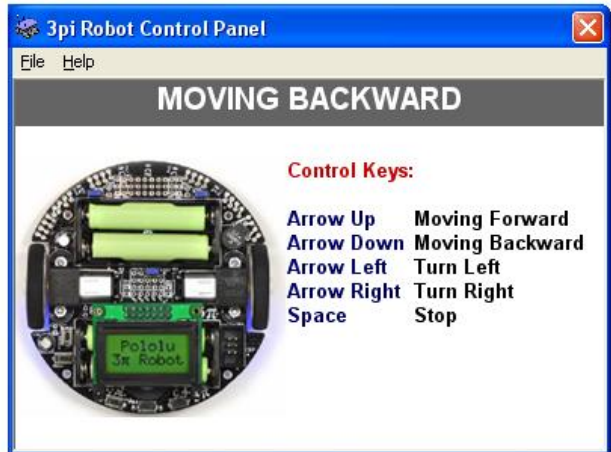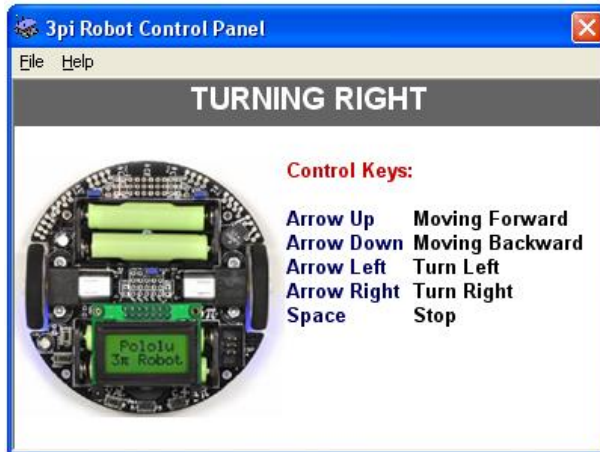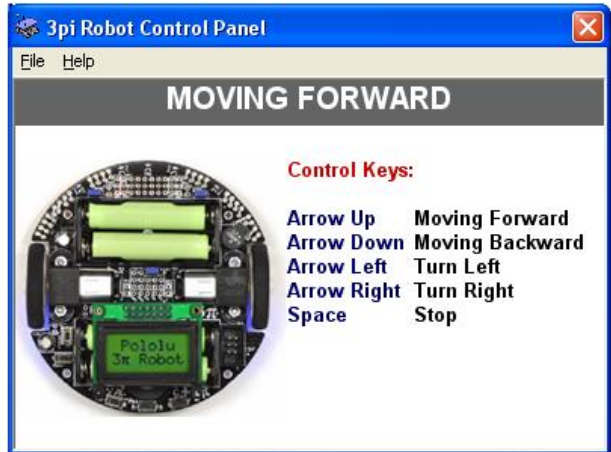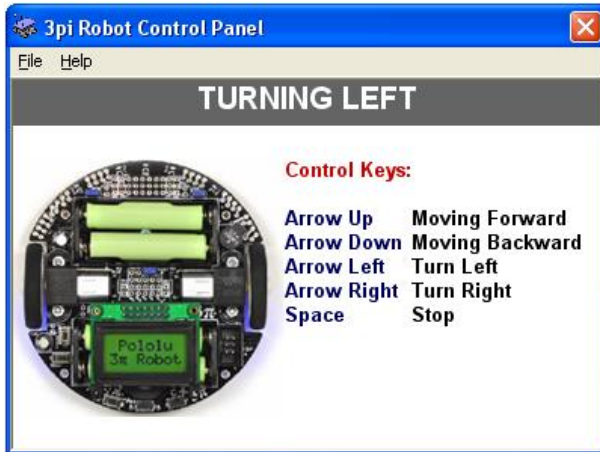
Above are some screenshots of moves of the "3pi Robot Control Panel".
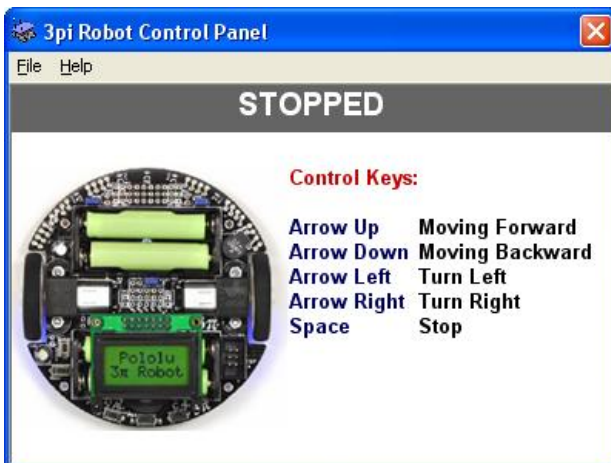
Above is the screenshots of "3pi Robot Control Panel". Pressing the "Space" key only stops the robot from moving, but it does not turn the robot off, the robot is still on.

Please, select "Help > Open Web Page" to learn more about 3pi Robot.

Appendix A

Restoring SLAVE firmware

The **SLAVE** firmware for 3pi Robot as part of **3pi Wireless Robot Set I set** is the standard slave software that comes with Pololu software package.

Please visit http://www.pololu.com/docs/0J21/10.a to obtain more information. In order to simplify this step BiPOM provides a complete package to run/build the slave firmware. Please download http://www.bipom.com/support/pololu/pololu_package.zip and unzip to any folder. Download the pre-compiled firmware (slave.hex**)** to Pololu 3pi Robot board using Pololu USB AVR Programmer and Atmel AVR Studio.

Appendix B

Installing drivers for Pololu USB AVR Programmer

Please first download the drivers "pgm03a_windows_091222.zip" for this hardware from the link below.

http://www.pololu.com/file/0J198/pgm03a_windows_091222.zip

Save this file and unzip the file to a folder first. And then connect the hardware to your PC's USB port with the provided cable. When Windows detects the new hardware, choose to install the driver for the hardware manually. Choose the path where you already unzipped the driver files for driver installation. There will be three different driver installations, so please follow up and choose the same unzipped driver files folder for each driver.

Appendix C

3pi Robot Control Commands

This section is for 3pi users that wish to control the robot from their PC through the serial port. This can be a wired or wireless serial port. Commands are the same for either case. Serial port settings are 115200 baud and 8 data bits, No Parity, 1 Stop bit. Each command has its own identifier and can contain an optional parameter. The software "3pi Robot Control Panel" that comes with the kit uses the following commands:

STOP: Turns off motors on the robot

FORWARD: Sets forward speed of left or right motor

BACKWARD: Sets backward speed of left or right motor

Robot command identifiers:

    CMD_STOP   = 0xBC
    CMD_M1_FORWARD   = 0xC1
    CMD_M2_FORWARD   = 0xC5
    CMD_M1_BACKWARD   = 0xC2
    CMD_M2_BACKWARD   = 0xC6

Speed parameters:

    SPEED_MAX   = 127
    SPEED_MIN   = 0

Stop command does not have any parameters.

To send commands, simply send 1 byte command identifier to the serial port. For speed commands, send 1 byte speed value directly after command identifier.

Sample C code below shows usage of these commands:

```c
...

char CmdBuffer[8];
int idx = 0;

switch(Key)
{
case KEY_FORWARD:
        // set half of full speed forward
        CmdBuffer[idx++] = CMD_M1_FORWARD;
        CmdBuffer[idx++] = SPEED_MAX / 2;
        CmdBuffer[idx++] = CMD_M2_FORWARD;
        CmdBuffer[idx++] = SPEED_MAX / 2;
        break;

case KEY_BACKWARD:
        // set half of full speed backward
        CmdBuffer[idx++] = CMD_M1_BACKWARD;
        CmdBuffer[idx++] = SPEED_MAX / 2;
        CmdBuffer[idx++] = CMD_M2_BACKWARD;
        CmdBuffer[idx++] = SPEED_MAX / 2;
        break;

case KEY_LEFT:
        // set right motor to 1/4 speed backward
        // stop left motor to turn robot around left wheel
        CmdBuffer[idx++] = CMD_M2_FORWARD;
        CmdBuffer[idx++] = SPEED_MAX / 4;
        CmdBuffer[idx++] = CMD_M1_FORWARD;
        CmdBuffer[idx++] = 0;
        break;

case KEY_RIGHT:
        // set left motor to 1/4 speed backward
        // stop right motor to turn robot around right wheel
        CmdBuffer[idx++] = CMD_M1_FORWARD;
        CmdBuffer[idx++] = SPEED_MAX / 4;
        CmdBuffer[idx++] = CMD_M2_FORWARD;
        CmdBuffer[idx++] = 0;
        break;

case KEY_STOP:
        // stop the robot
        CmdBuffer[idx] = CMD_STOP;
        break;

default:
        break;
}

// send command to port
if(idx > 0)
        ComPort.Send(CmdBuffer, idx);
...
```