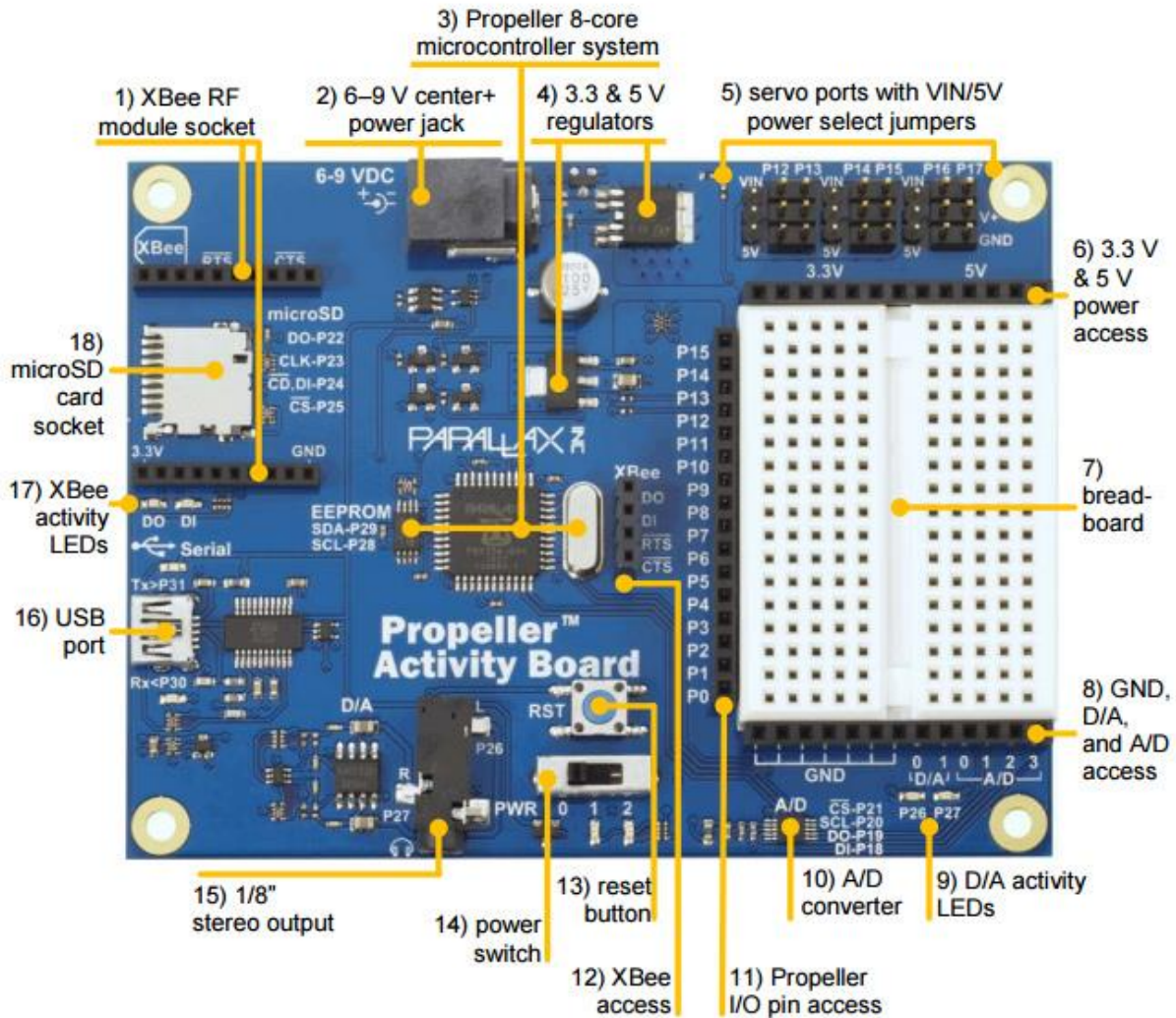


# The ActivityBot board

Jeffrey La Favre

November 9, 2015

Here is some information about the board that is part of the ActivityBot robot.



Near the center of the board is item 3), the Propeller 8-core microcontroller system, consisting of three parts: an EEPROM chip on the left, the 8 core microprocessor, part number P8X32A, which has 44 pins, 11 on each side, and to the right a 5 MHz crystal (the silver colored object). A pin of the crystal is connected to a pin of the microprocessor and the microprocessor multiplies the 5 MHz signal by 16 to get a clock speed of 80 MHz, which is the speed at which the microprocessor runs.

EEPROM is the acronym for Electrically Erasable Programmable Read-Only Memory. It is used to store the robot programs you write, just as a computer hard drive or flash drive stores programs. The EEPROM contains 64 KB of memory. The lower 32 KB is available to store your programs, but only one program at a time. Each time you load a new program into the EEPROM, the former program is erased.

The upper 32 KB of the EEPROM is mostly reserved to store information useful in running various programs, such as calibration data. While it is possible to use some of the upper half of EEPROM memory for your programs, usually you should keep your programs small enough to fit in the lower 32 KB of memory. The primary reason is that your program must load into RAM before it runs, and RAM memory is only 32 KB. The RAM (random access memory) is located inside the microprocessor chip. RAM memory is volatile, and anything stored in it is lost when the power is turned off.

It is also possible to store some data used in programs on a microSD card, inserted into the card socket (number 18 in photo).

When you want to transfer a program you have written, from the SimpleIDE software on the computer, to the ActivityBot board, you have two choices: **1)** load into EEPROM or **2)** load into RAM. If you want the program to remain stored on the robot after it is turned off, then you need to load it into the EEPROM. If you want to just do a quick test of a program, while the robot remains connected to the computer, then you can load it into RAM. The advantage in loading into RAM is that a previous program stored in EEPROM is not erased.

When the power switch (number 14 on photo) is set to the ON position the microprocessor first checks for a USB connection to a computer through a USB cord connected to the USB port (number 16 on photo). If a connection is detected, then the robot waits for a download from the computer. That download may be directed to load into RAM or into EEPROM depending on the menu selection used in SimpleIDE. If a connection is not detected to a computer, then the program stored in the EEPROM is copied to the RAM memory and the program runs. There is a reset button (number 13) on the board just above the power switch. Use the reset button to restart a program, either while it is running or after the program has finished running.

The power switch (number 14) has three positions: 0, 1, 2, as marked on the board. Position 0 is off. Position 1 is ON, but no power will be connected to the servo power pins associated with ports P12 to P17. Use position 1 when the robot is connected to the computer to prevent the wheels of the robot from spinning. If your program includes drive code for the robot and you want to test the movement of the robot, then you must place the power switch in the 2 position so that power will be delivered to the wheel motors.

Power from the battery pack is connected to the board at the power jack (number 2 on photo).

XBee sockets (numbers 1 and 12 on photo) are for XBee accessories, that provide wireless communications with other devices.

The audio out jack (number 15 on photo) is used to connect a small speaker. You can store audio files on a microSD card and have the playback sent to the speaker. Using this feature your robot can talk!

The board includes a breadboard (number 7 on photo) where you can build custom circuits for the robot. Surrounding three sides of the breadboard there are a number of sockets that allow connections to various ports of the board. On the left side of the breadboard there are 16 sockets (number 11 on photo) to connect to input/output ports numbers P0 to P15. On the top side of the breadboard are sockets to connect to regulated positive power. The sockets to the left provide 3.3 volts and those to the right provide 5.0 volts (number 6 on photo). At the bottom of the breadboard, to the left, there are sockets to the negative power supply (also known as ground or GND - number 8 on photo). Also at the

bottom are sockets for connection to an analog to digital converter (A/D) and a digital to analog converter (D/A) – (number 8 on photo). The A/D converter chip is number 10 in photo. The D/A chip does not have a number label on the photo but is located just to the left of the audio output jack (number 15).

A A/D converter takes an analog signal (*i.e.*, a certain voltage) and converts that to a digital number that the microprocessor can work with (microprocessors cannot manipulate analog values directly). An example use would be certain IR distance sensors that output a voltage proportional to the distance to an object. The sensor output must be converted to a digital number before it is sent to the microprocessor. By connecting the output of the sensor to the A/D converter, the A/D converter can convert the analog signal received from the sensor and then send the digital value to the microprocessor for processing by the stored program.

There are 6 servo ports above the breadboard (number 5 in photo). These are numbered P12 to P17 and each one contains three pins. The top pin is the actual input/output port. The middle pin connects to positive power and the bottom pin connects to ground (GND). Just to the left of each pair of ports there is an additional set of three pins to be used with a jumper. The jumpers are not in place in the photo. If a jumper is used to connect the center pin to the lower pin, then the positive power pins of the pair of ports receive 5 volts regulated. If a jumper is used to connect the center pin to the upper pin, then the positive power pins of the pair of ports receive a direct connection to the positive side of the power pack (which will be 7.5 volts for the battery pack containing 5 1.5 volt cells).

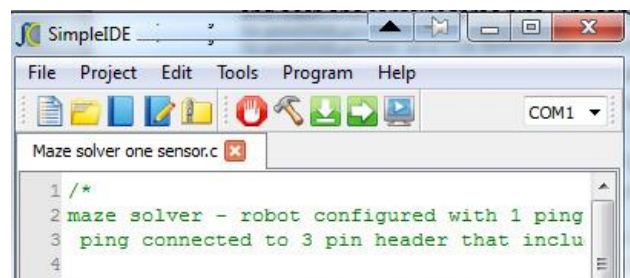
It is possible by programming to select the purpose of each servo port, but usually you will just use the default configuration, which is as follows: P12 is for the left wheel motor (continuous rotation servo), P13 is for the right wheel motor, P14 if for the left wheel encoder and P15 is for the right wheel encoder. Ports P16 and P17 are free for other uses. I have used P17 to connect to a Ping sensor. This is a convenient way to connect this sensor as it requires a 5 volt supply.

The jumper for ports P12 and P13 is set to deliver 7.5 volts for the motors. The jumper for ports P14 and P15 is set to deliver 5 volts regulated for the wheel encoders as they require a 5 volt supply. You must be very careful to set the jumpers properly. Otherwise you may damage sensors by supplying excessive voltage (7.5 volts).


### Loading a Program into EEPROM of the robot

Suppose you have just written a program to make your robot follow a maze. You want to test it out with the robot by loading the program into the robot EEPROM memory. Here is the procedure to follow:

1. Start the SimpleIDE software.
2. Open your program.
3. Take note of the COM port setting before you connect your robot to the computer (in image to right it is set to COM1).
4. Connect the robot USB cord to your computer and the robot.
5. A few seconds after connecting your robot, you should find a new COM port



available in the drop-down list (on my computers it is COM3). Select the new COM port as this is the one the computer is using to connect to the robot. If you don't have the correct COM port selected, your attempt to load the program will fail.

6. Set the power switch on the robot to position 1 (if you don't do this the program won't load).
7. Click the EEPROM button on the toolbar  to load your program into the robot EEPROM. Alternatively, you can open the **Program** menu and select **Load EEPROM & Run**.
8. The progress of the load will be shown at the bottom of the SimpleIDE window. If there is an error in your program, a warning will appear and the program will not "build." If the software does not detect any problems with your program, after it is finished "building" the program (compiling it), it will begin to load it into the EEPROM of the robot. When it is done loading it will let you know at the bottom of the software window. **You must wait until the program has finished building AND loading before you continue.**
9. Now set the power switch to position 0 (off) and disconnect the USB cord from the robot.
10. Place the robot on the floor, or in the maze if you have one, then slide the power switch to position 2. After a second or two, the robot will start running.