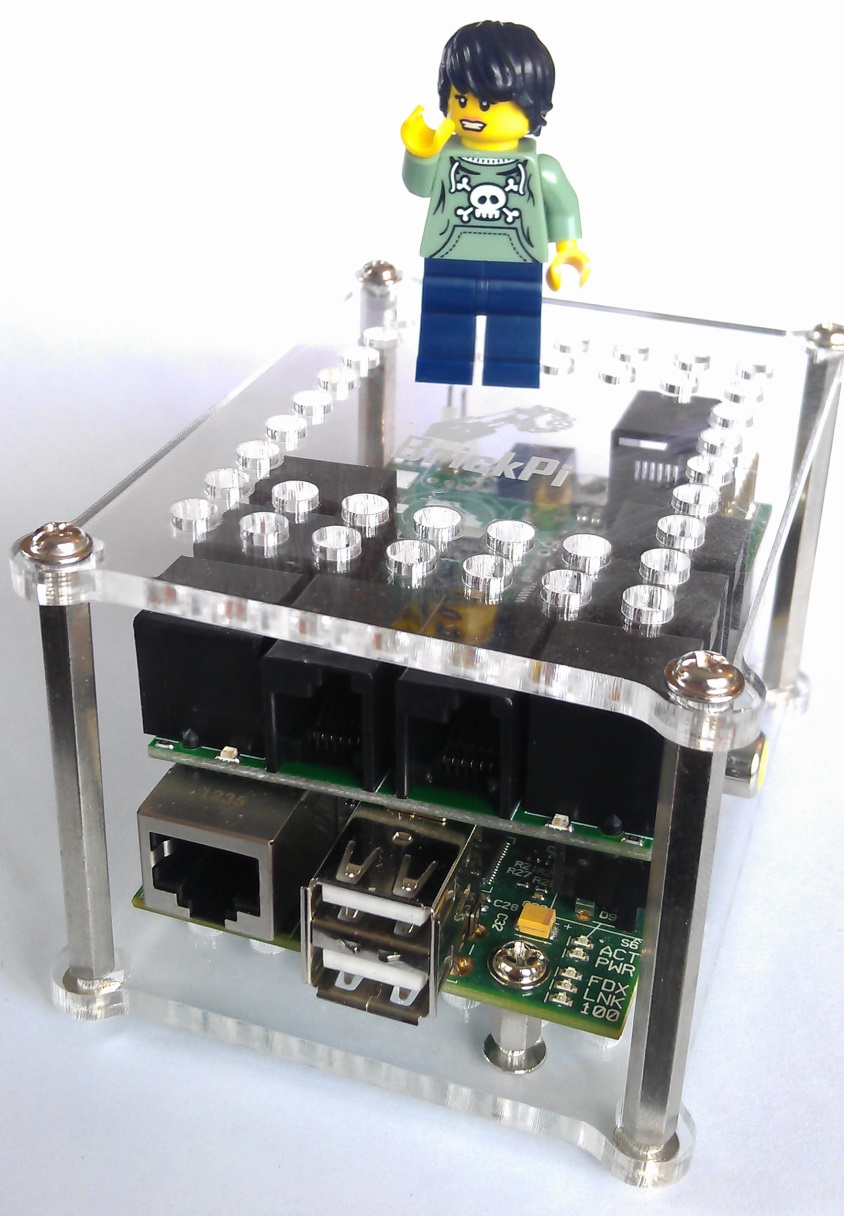
The BrickPi: LEGO® Robots with the Raspberry Pi

The BrickPi is a system for making and controlling LEGO robots with the brains of a Raspberry Pi. It brings the Raspberry Pi into the physical world, allowing you to quickly construct robots. Want to add a mouse, keyboard, Bluetooth to a robot? No problem, you can easily add just about anything through USB. Want to control your robot with tablets, phones, or the web? No problem, the Raspberry Pi can connect to these devices and more (and we have some examples that make it even easier!). Want to disconnect your robot from the wall? Again, no problem: the BrickPi comes with a 9V battery adapter. In short, the BrickPi makes it easy to make super-smart robots built with the ease of LEGO.

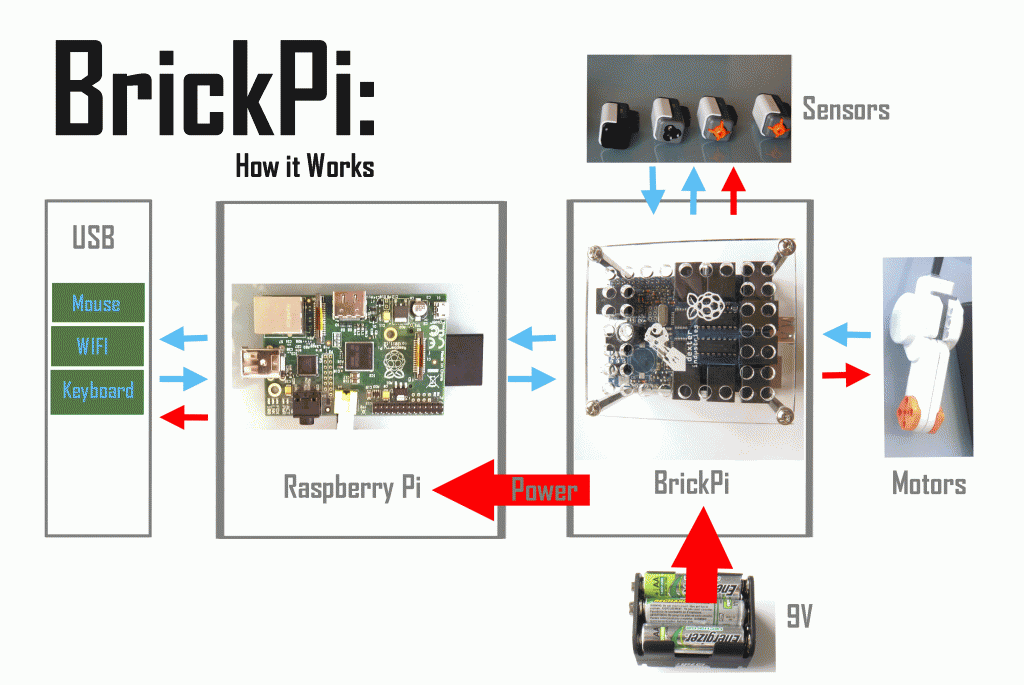


## What is the BrickPi?

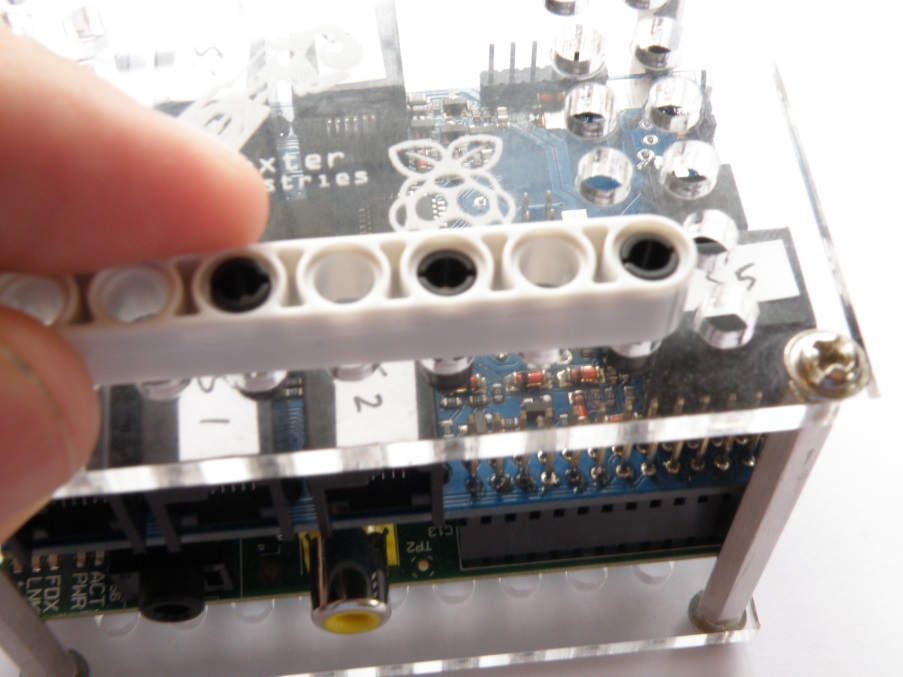
The BrickPi is an interface between the Raspberry Pi and LEGO Mindstorms sensors and motors, a portable power source for the Raspberry Pi, and a case that allows you to attach LEGO motors, beams, sensors, and other parts. The BrickPi controls four LEGO NXT motors and read the motor encoder values (encoder values tell you how far the motor has turned). The BrickPi also has four sensor ports. These ports can read digital NXT sensors (like the Dexter Industries dGPS sensor, or the HiTechnic Infrared Sensor), and they can also read analog sensors (like the Dexter Industries thermometer). We have an extra sensor port (Port 5) that can read just digital sensors.

The BrickPi has a power unit that allows you to power the BrickPi and Raspberry Pi with a battery (rather than the usual wall plug), so you can make robots that roam around the house or classroom. The power unit on the BrickPi can take 9-12 Volts, and provides power to the

Raspberry Pi, motors, and sensors.



The BrickPi system also includes an enclosure that LEGO Technic parts, sensors, and motors can be attached to. The enclosure houses the Raspberry Pi and the BrickPi. It’s made of acrylic and comes with evenly-spaced holes along the top and bottom that are big enough for Technic pegs.



The Raspberry Pi USB ports are open and available. This is where things can get really interesting! You can easily attach WIFI and Bluetooth to make your robot wireless. But you can also attach keyboards, joysticks, mice, printers, memory sticks, and Arduinos . . . the possibilities become pretty endless.

# Tips for Getting Started

**Our Guide:**

We wanted to get a definitive quick setup guide that gets you started and running as quickly as possible. On our website, we have a section for the BrickPi called “Getting Started” that has numbered steps (there are 6!) to get the BrickPi running. The guide covers everything from assembled the case and powering up, to how to choose or setup your SD card, and how to attach LEGO parts.

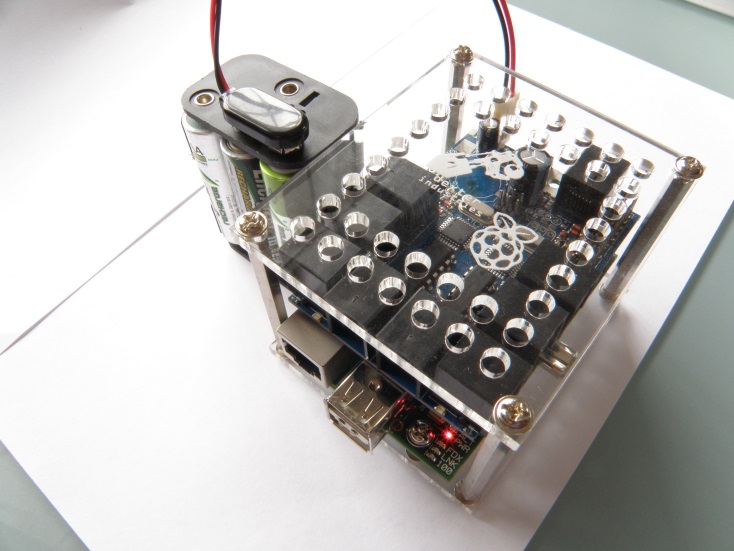
<http://www.dexterindustries.com/BrickPi/getting-started/>

**The Image:**

Setting up the Raspberry Pi to communicate with the BrickPi can be confusing. There are a few software packages that need to be installed to communicate with the BrickPi. These are listed in our guide. However, loading software packages can be especially daunting for new users, so we put together a custom version of the Raspbian distribution, and that too can be downloaded from our website. We are also offering SD cards with this distribution already loaded.

**Batteries:**

The Raspberry Pi can be a real energy hog (especially compared to the Arduino) and can draw quite a bit of power. To increase the life-span of the battery, we recommend using a battery pack that carries 6 AA batteries. We also recommend keeping the Raspberry Pi plugged into a computer or wall while programming to save on batteries.



Many folks have asked us how long a battery usually lasts. It’s a really tough question to answer (and we don’t even try to answer it) because, depending on how you write your program, and how many motors and sensors and USB devices you use, the results can vary widely. A battery can last hours, or it can go dead in a matter of 20 minutes.

**Connecting to the Raspberry Pi:**

We wanted to make it as easy as possible to start programming the BrickPi. Not everyone has an extra monitor, mouse, and keyboard lying around. And if you’re building a robot, chances are that at some point, you’ll want to disconnect it from the wall and monitor and let it roam around. To try to make it easier to connect to the Raspberry Pi, the image above comes loaded with VNC server (and automatically sets one up), SSH, and a few other goodies. We put together a short tutorial for connecting, running, and programming your Raspberry Pi from your laptop or desktop for the first time with just an Ethernet cable.

# Programming

We made this project open source: the best way to help people learn is to let them take things apart. This includes the physical design (how we wired things together) and the different layers of programming we put together on the firmware and software. Both the hardware and the software are open-source and available on Github. We hope that by leaving the design open we can get help on improvements from the community.

We’ve put a lot of work into a software package for the Raspbery Pi. This includes easy-to-follow examples for robotic cars, turrets, tanks, and arms. We have drivers for C, Python, and MIT’s graphic language Scratch. The beauty of the hardware architecture and the Raspberry Pi is that any language that can run on the Raspberry Pi can be used to control a student’s robot; we’ve been contacted by a number of folks that are already working to port it to their favorite language.

# Two Examples in Python

Below we’ll give a few examples of how to setup and run a few sensors and motors. Again, we’ve got examples for the full range of LEGO and Dexter Industries sensors for the LEGO Mindstorms NXT on Github. We also have the same examples in C, and most in Scratch.

**Running a Motor:** The python example below runs one motor on Port A forward for 3 seconds, and then backwards for 3 seconds. At its simplest, running a motor requires two lines of code: one to setup the motor, and one to set the motor power.

from BrickPi import \* #import BrickPi.py file to use BrickPi operations

BrickPiSetup() # setup the serial port for communication

BrickPi.MotorEnable[PORT\_A] = 1 #Enable the Motor A

BrickPiSetupSensors() #Send the properties of sensors to BrickPi

while True:

print "Running Forward"

BrickPi.MotorSpeed[PORT\_A] = 200 #Set the speed of MotorA (-255 to 255)

ot = time.time()

while(time.time() - ot < 3): #running while loop for 3 seconds

BrickPiUpdateValues() # Ask BrickPi to update values for sensors/motors

time.sleep(.1) # sleep for 100 ms

print "Running Reverse"

BrickPi.MotorSpeed[PORT\_A] = -200 #Set the speed of MotorA (-255 to 255)

ot = time.time()

while(time.time() - ot < 3): #running while loop for 3 seconds

BrickPiUpdateValues() # Ask BrickPi to update values for sensors/motors

time.sleep(.1) # sleep for 100 ms

**Reading an Ultrasonic Sensor:** The LEGO Ultrasonic sensor can be attached to any of the four ports and tells you the distance an object is from the robot. The program below sets up the Ultrasonic Sensor on Port 1, and then reads the value from the sensor in a loop, printing the measured distance on the screen. Again, in its simplest form, running the ultrasonic sensor requires all of two lines of code: one initializes the sensor, and one refreshes the value the sensor is reading.

from BrickPi import \* #import BrickPi.py file to use BrickPi operations

BrickPiSetup() # setup the serial port for communication

BrickPi.SensorType[PORT\_1] = TYPE\_SENSOR\_ULTRASONIC\_CONT #Set the type of sensor at PORT\_1

BrickPiSetupSensors() #Send the properties of sensors to BrickPi

while True:

result = BrickPiUpdateValues() # Ask BrickPi to update values for sensors/motors

if not result :

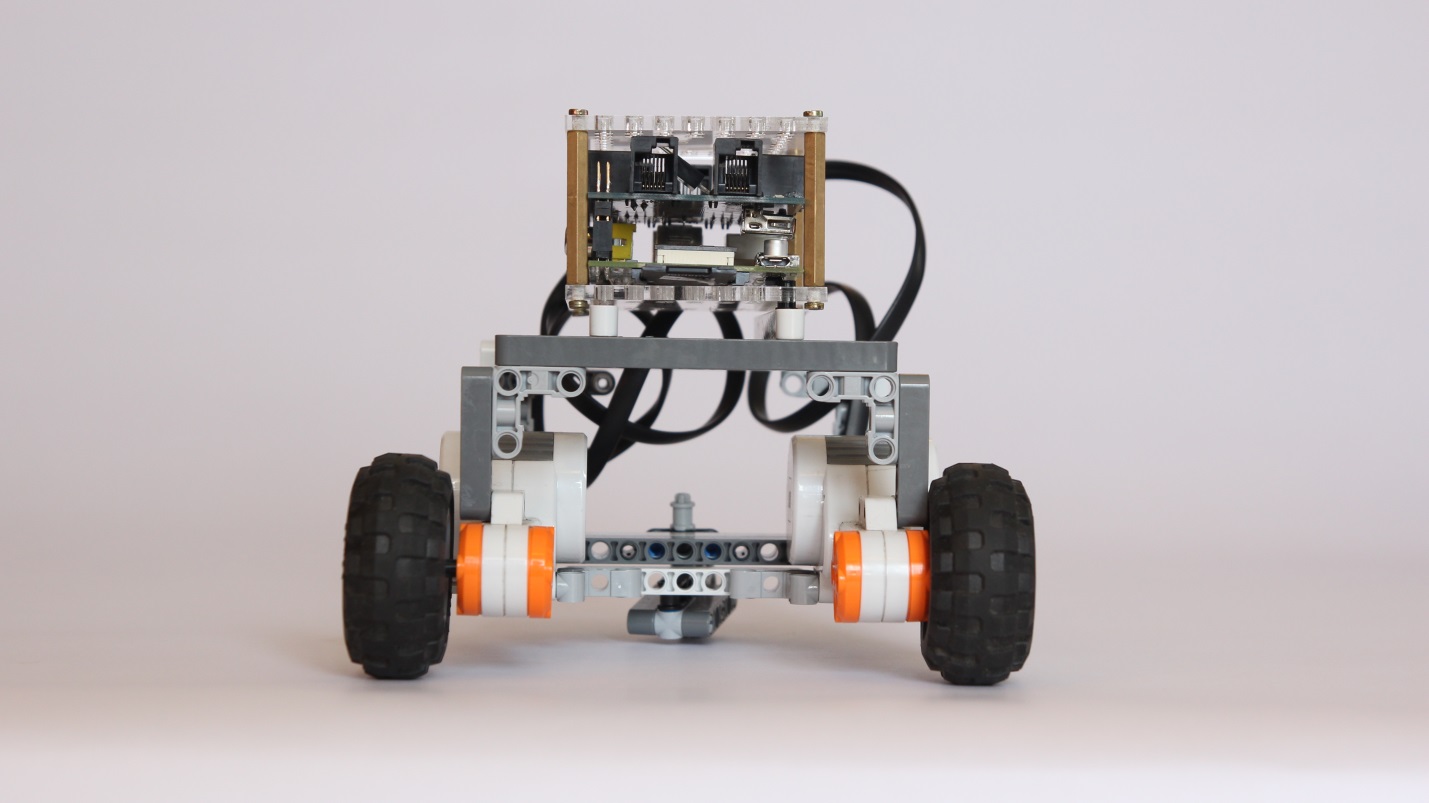
print BrickPi.Sensor[PORT\_1] #BrickPi.Sensor[PORT] stores the value obtained from sensor

time.sleep(.01) # sleep for 10 ms

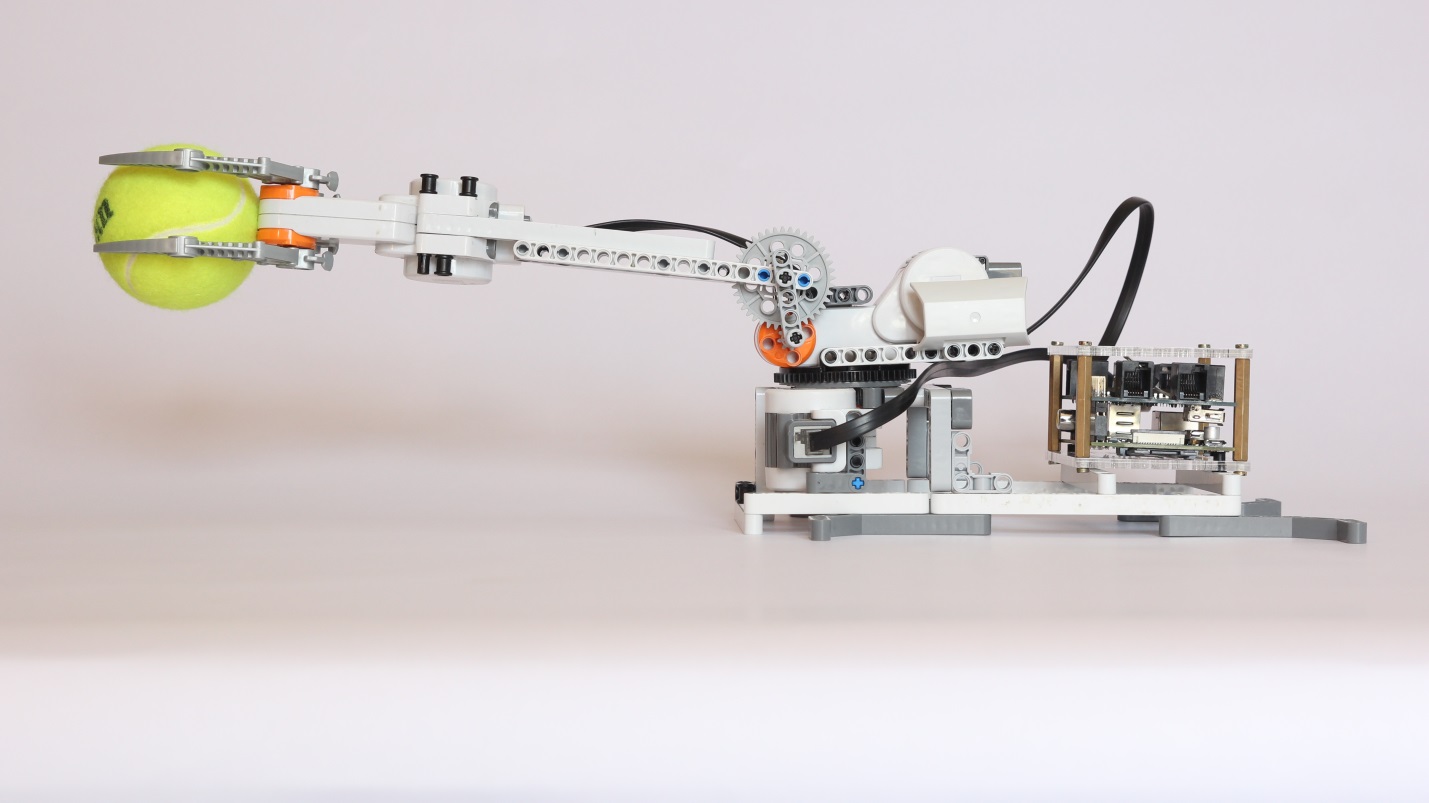
# Cool projects

After testing out some of the examples of LEGO sensors, you can start to think about some really creative projects that use the horsepower of the Raspberry Pi. We have a few examples up already on Github, and we’ll be adding a few more. We have 4 core projects, that we included building instructions for, as well as full python code:

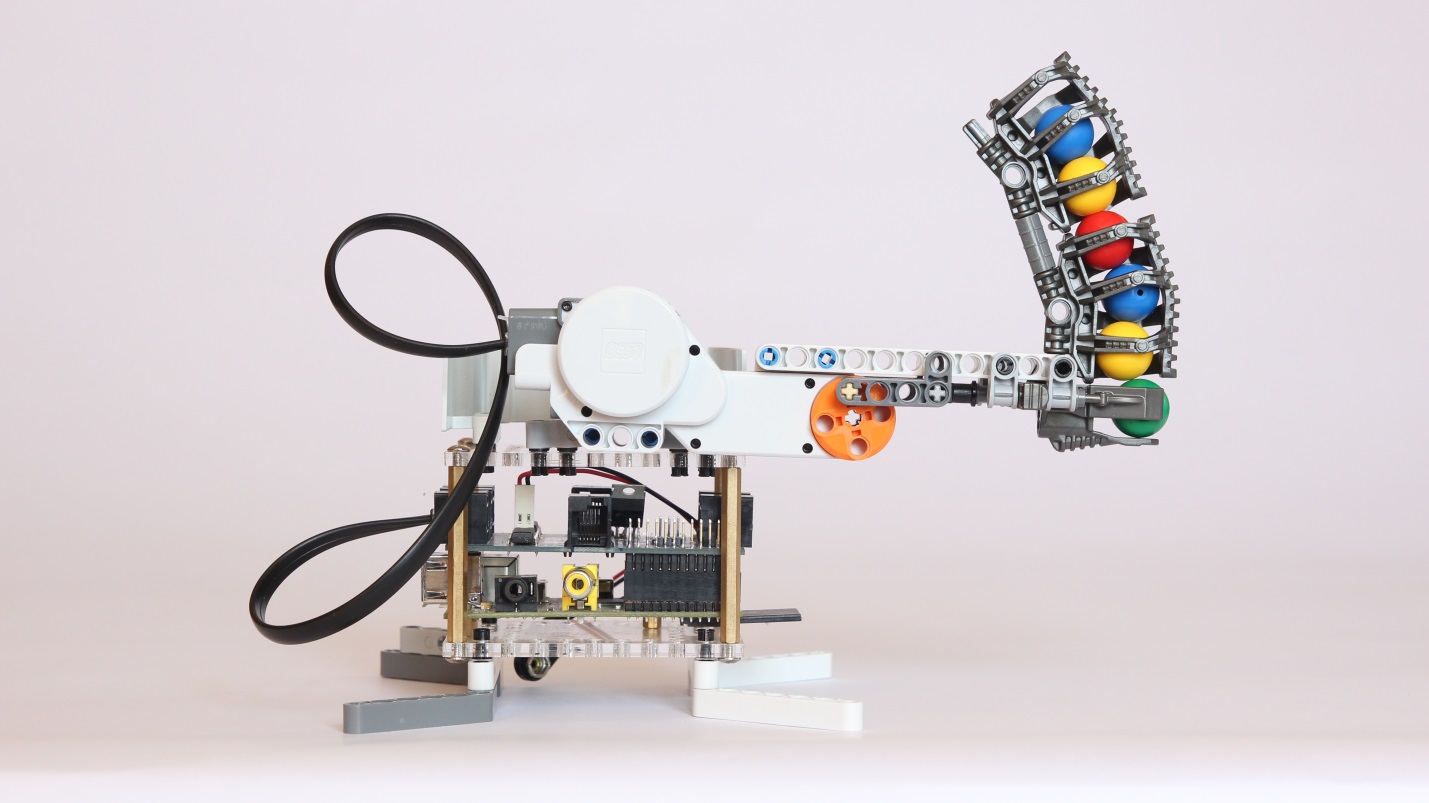
**WIFI Controlled Car:** This example is a car controlled with WIFI. In the example, you log into the Raspberry Pi through SSH, and then control movements over the internet via keystrokes (8 for forward, 4 for left, etc). In the example, the Ultrasonic Sensor is used to detect objects and obstructions, and stop before crashing into them.



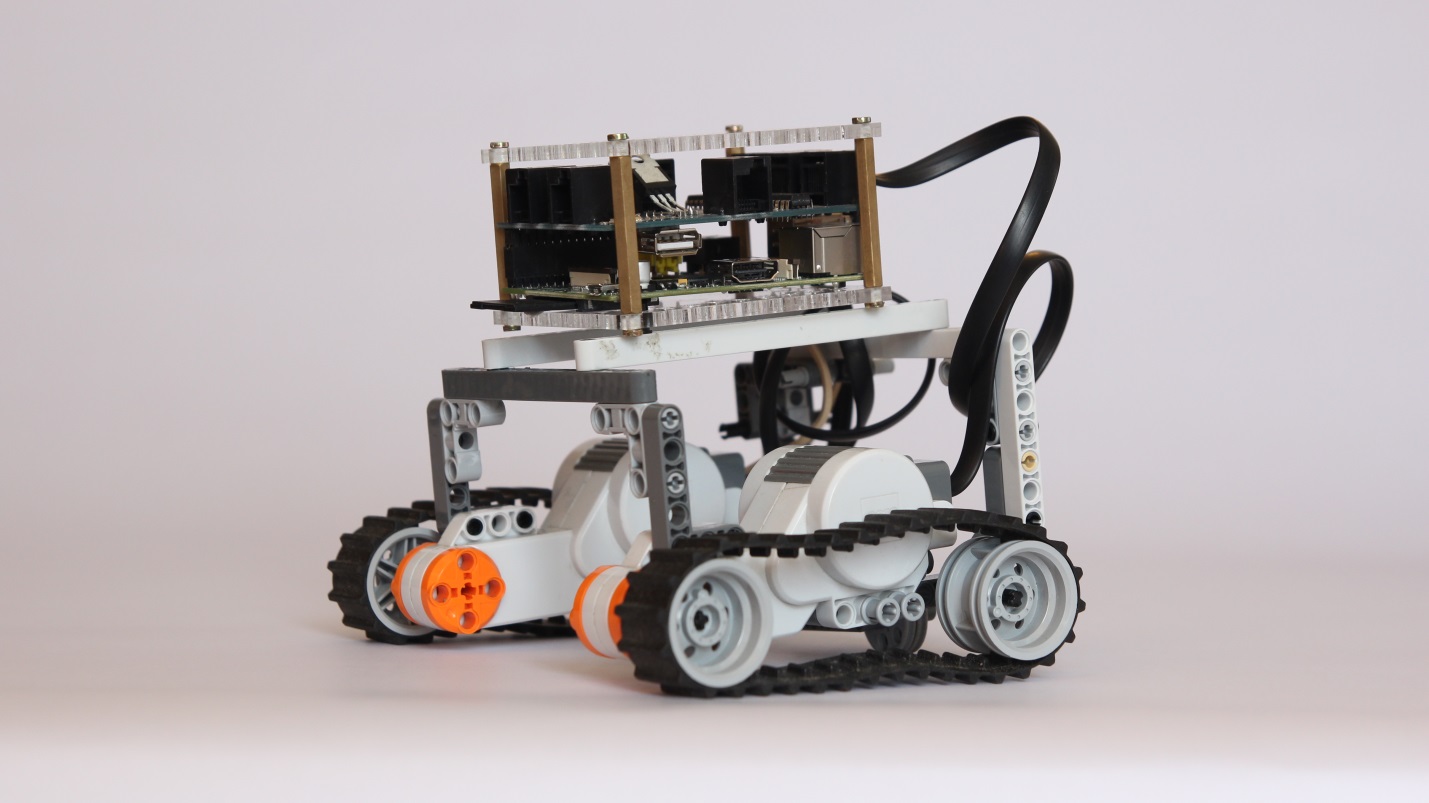
**Robotic Arm:** This example uses the BrickPi as a robotic arm for grabbing and moving materials. In this example, a keyboard in the Raspberry Pi is used to control the movement of the arm.



**Shooter Turret:** This example uses the BrickPi as a turret ball shooter. The turret can rotate and is controlled by the mouse. Moving the mouse left and right changes the angle of the shooter, and the firing mechanism is controlled with a press of the mouse button.



**Browser Controlled Tank:** This example uses the BrickPi as a tank. The example sets up a webserver over wifi on the Raspberry Pi. When you login to your Raspberry Pi, the Pi serves up a web page that has controls for moving the tank around and controlling the tank motion.



We hope you enjoy playing with the BrickPi! Even more, we hope you can learn a lot about robotics and coding by playing with the BrickPi. We encourage everyone to contribute with their feedback (complaints and compliments are welcome through Github or our forums) and if you build something cool, we would love to share it with the community!