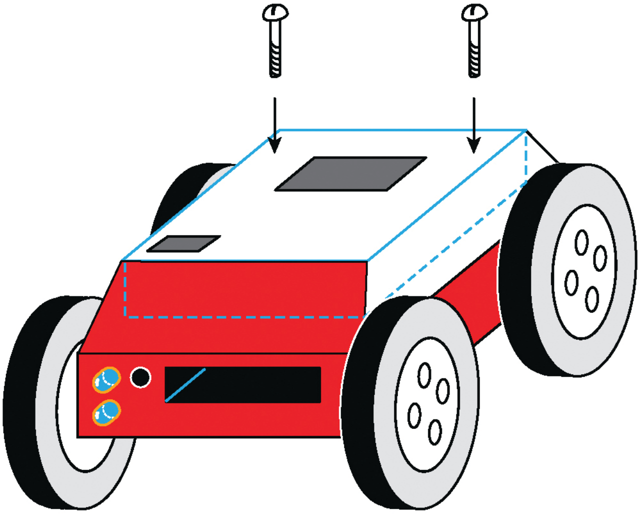


BUILD A BOT

Learn everything you need to know to build and program a 2 wheeled battle-worthy sumobot! We'll start by learning about the components that make up a robot, and you'll assemble your custom bot from an assortment of parts on hand. Next we'll look into the software side of things and figure out how to make them move. Not a great programmer? No problem! We have options including manual joystick control, a simple drag and drop lego-like programming environment and of course traditional code too. Finally, it's time to put your bot to the test by battling it out in the sumo arena with other bots or navigating the timed obstacle course!

AUTHORS

Pawel Szymczykowski is a software engineer at Zappos Labs (but his business card says "King of Goblins") and a proud member of the awesome #VegasTech community. He is one of the board members of SYN Shop, a local hackerspace, and likes to tinker with a little bit of everything and make all sorts of things. He loves to organize, volunteer and participate in tech and maker community events. Recently, he was bitten by the robotics bug and designed an open source wooden robot kit called Sumobot Jr to share his passion with others via workshops. Would you like to build robots together?

BUILD A BOT

OVERVIEW

In this workshop you will learn how to assemble and program an arduino based robot that you can control with your laptop, teach to 'see' by using sensors, or use to battle other bots in the sumo ring.

PREPARATIONS

We will be using a piece of software called **Robotnik** that lets you program and control the Arduino robot. In order to run Robotnik on your laptop, you will first need to install Node.js. Please go to *http://nodejs.org/*, click *Install* and then follow the instructions.

Once you have installed Node.js, you can install Robotnik using npm. If you are on a Mac or Linux, just open a command propmt. If you are a Windows user, open 'Node.js command prompt' that was installed along with Node.js. Then you can installed Robotnik by typing:

```
npm install -g robotnik
```

Once the install is complete, connect the Arduino to your laptop using the USB cable:

Then you can start the program by typing:

```
robotnik
```

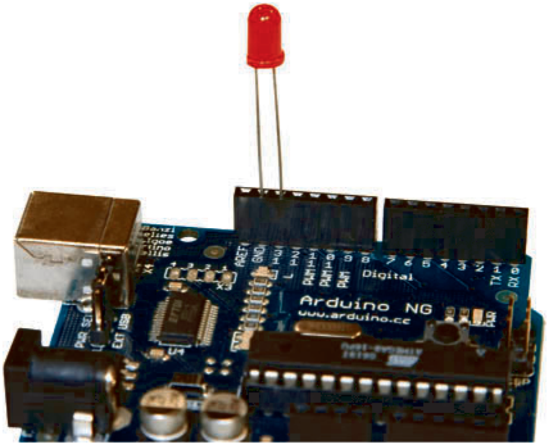
That's it! The program may instruct you to open a web browser to *http://localhost:8057/* if it couldn't do so for you automatically. You are now ready to begin the tutorial. Remember, if you run into any trouble, just ask one of the workshop assistants and we'll help you as much as you require!

MAIN FLOW

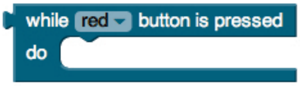
QUEST 1: MAKE A LED BLINK

Robotnik is a programming system that works like Lego™ or puzzle pieces. On the left are available blocks and on the right is where you drag them to make your program. We're going to make a program that turns a LED on when you press a virtual button in Robotnik.

First, you'll need to plug a LED (any color) into the Arduino. The short leg goes into GND and the long leg goes into 13.



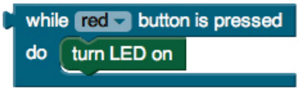
Next, find a block like this:



and drag it on to the right side of the screen. As you can see, the block describes what it will do in English. It's just up to us to finish the sentence now. If you wish, you can change 'red' to 'blue' to change the sentence to talk about a different button. Let's leave it at 'red' for now.

Next, look for the 'turn LED on' block and drag it inside of the first block next to 'do', so that it looks like this:

Make sure the two pieces are connected - you'll briefly see a yellow outline and hear a clicking noise if you did it right.



```
while red button is pressed
do  turn LED on
```

Now click the *Run Program* button. If you did everything correctly, a virtual joystick will appear on your screen and clicking on the red button will make the LED blink!

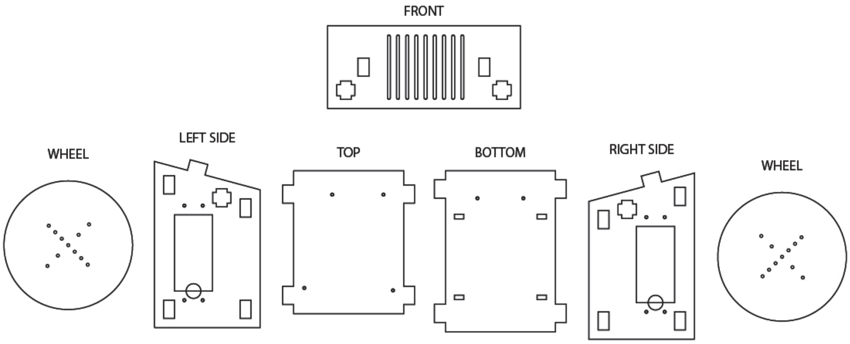
Didn't work? Uh oh. Make sure:

- your USB cable is plugged into your computer and Arduino
- the short leg of the LED is in GND and the long leg is in 13 and it is plugged in
- that Robotnik is still running - look at your terminal window and run `robotnik` again if needed

Don't forget to ask for help if you need it!

QUEST 2: CONTROLLING A ROBOT

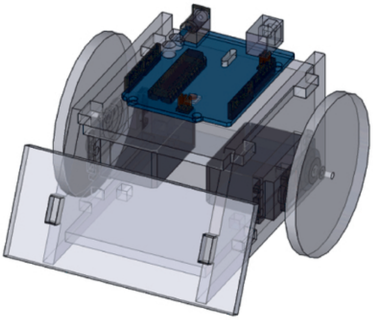
You now know the how to create programs by using Robotnik blocks and how to run programs. We are ready to move on to something more advanced. The first step is to assemble the robot. All of the wooden pieces are labeled and slot together - look at the example robot and ask questions if you need help.



Put all the pieces together, then pop the motors in through the sides and screw them in. Attach the Arduino to the top, plug the shield on to the top of the Arduino and plug the motors and batteries in to the labeled ports of the shield. Attach the wheels to the sides of the motors. Many Arduino shields add additional hardware, but ours is just used to connect everything together in the right way. Under the shield and behind the scenes, the batteries are connected to the motors, and one pin from each motor called the 'control pin' goes to the appropriate numbered pin on the Arduino.

When you're done, your robot should look something like this:

Good job! Our goal is now to make the robot move forward when we press the joystick forward, backward when we move backward and to turn left and right when we move that way. You can do



this by assembling the blocks correctly. Remember that each motor moves independently, and when they move in the same direction, the robot will move in that direction. When they move in opposite directions, the robot will twist in place. When only one motor turns, the robot will pivot around the wheel that isn't moving. Remember to think about the way the motors are facing.

Once you have this down, you can move pilot your robot around the obstacle course, or battle someone in the sumobot ring! The first robot to leave the circle loses the match!

Problems?

- Make all the cables are plugged in correctly
- Is Robotnik is still running? Look at your terminal window and run `robotnik` again if needed
- Are there batteries in the battery case and is it turned on?
- Remember that the motors are facing opposite directions
- Are there any 'extra' sticking out of the Arduino shield? The shield and Arduino should align perfectly and there should be no 'extra' pins.

Quest 3: Teaching the robot to think for itself

You've made a robot that you can control with your computer. This is one kind of robot, but to really be useful, a robot can be made autonomous so that it senses the world around it with 'sensor modules' and makes decisions automatically based on the data that it gets from them.

We have provided a 'proximity sensor' that tells the robot how far it is from an object. Attach it to the front of the robot, and plug it into the labeled port on the shield. Now the robot should be able to 'see'. Our goal is to tell the robot to:

1. Move forward
2. If there is something 5cm in front of the robot, turn right slightly
3. Repeat

Again, you can do this by dragging the spaces Once you have this program working, you should be able to set your robot down in the obstacle course and let it go without bumping into anything.

Did it work? Congratulations on creating a robot that can move and think for itself! Do your part to prevent the robot apocalypse by always staying one step ahead of the machines.

ADVANCED FLOW

You program? Great! I want you to do the same three quests as in the main flow, but I'm going to give you extra information so you can go on side quests. :)

Robotnik is a Google project that actually generates source code when you place the blocks. You can see the source code by clicking on the 'Code' tab. Behind the scenes, we are generating Javascript code and using Rick Waldron's excellent Johnny-Five library to talk to the Arduino over protocol called Firmata.

You can modify the code in place in the editor and run directly from there instead of using the block interface. This gives you much more flexibility and you can do all sorts of things! The Johnny-Five documentation is linked in the 'Code' tab. Between looking at the code generated by the blocks, and the documentation, you should be able to complete all the quests and add some of your own special twists.

NOT AT MAKERLAND?

Are you home from the conference and want to do this again? No problem! You missed Makerland but want to follow this tutorial anyway? No problem! Everything we use is open source and readily available. First, you'll need an Arduino Uno R3 with the firmata program uploaded to it. This is a requirement to use Johnny-Five. To do that:

- Download Arduino IDE (*http://arduino.cc/en/main/software*)
- Plug in your Arduino or Arduino compatible microcontroller via USB
- Open the Arduino IDE, select: *File* > *Examples* > *Firmata* > *StandardFirmata*
- Click the *Upload* button.

Then you'll need the parts for the robot and schematics for the shield. This is all available at *http://sumobotkit.com* - if you have access to a nice Hackerspace or a laser cutter and 3D printer, you can make everything yourself, otherwise you can find places to order the parts through a service like *http://ponoko.com*

Finally, here is the shopping list of parts from Adafruit:

- Arduino Uno R3: *http://www.adafruit.com/products/50*
- Proximity Sensor: *http://www.adafruit.com/products/164*
- Battery Case: *http://www.adafruit.com/products/830*
- Continuous Rotation Servo Motor (2): *http://www.adafruit.com/products/154*

Any color of LED will do, and you'll also need a 'type B' USB cable to connect the Arduino. You can find them at your local electronics store or at Adafruit if you search.