MIYbot – Make It Yourself Inexpensive Robot for Education

Brian P. Wagner
Georgia Tech OMSCS
8700 Noll Ct
Louisville, KY 40241 USA
1-502-445-2850
bpwagner@gatech.edu
brian@tegrasys.com

ABSTRACT
In this paper, I describe the MIYbot. MIYbot is an educational robot platform for Middle School and High School computer science and engineering classes. MIYbot was designed to be a very low cost educational robot used for teaching soldering and building skills and to be fully programmable in the Arduino programming environment. Because the MIYbot is very low cost, each student makes their own MIYbot that they can take home and continue using at the end of the class. The body of the MIYbot is not a kit and encourages students to design, build, tinker and engineer their MIYbot as they see fit.

CCS Concepts
• **Social and professional topics~Computational science and engineering education** • *Social and professional topics~K-12 education* • *Computer systems organization~Robotic components* • Computer systems organization~Sensors and actuators

Keywords
robot; education; Arduino; sensor; middle school; high school

1. INTRODUCTION
Educational Robots are too expensive! Many schools want to create robot classes to follow Science, Technology, Engineering and Mathematics (STEM) initiatives. Robots are a natural STEM project. Most of these robot classes end up using very expensive reusable kits (often Lego NXT or VEX robots). At the end of the class, however, students are disappointed because they are unable to take their creations home. Students have no ability to continue experimenting, programming or building their robots at home. They have no ability to hone their new skills to create something new. The MIYbot (Figure 1) solves this problem. The only way I have found to combat the high cost of robot classes is to source the robot parts and components directly from China and other suppliers. Sourcing robot parts is not a difficult task and MIYbot documentation contains instructions describing where to get circuit boards, motors, sensors and other electronic components.

Conference'10, Month 1–2, 2010, City, State, Country.

DOI: http://dx.doi.org/10.1145/12345.67890

Another problem with robot kits is they are often put together by snapping plastic parts together or with just nuts and bolts. They come as a box of parts for the students to put together. Students are not actually designing their robot. The MIYbot, on the other hand, can be built with hand tools using things from around the house. The MIYbot can also be built using the tools that are becoming more common in schools like laser cutters and 3D printers. Furthermore, my robot will require soldering, a skill that is both fun and empowering.

For a truly exciting robot class students need to learn all that is required in building a robot, not just put together a kit. Building, testing, breaking, fixing, and programming are all parts of the engineering process. These are the tasks that create the learning environment that schools want in their STEM classes!

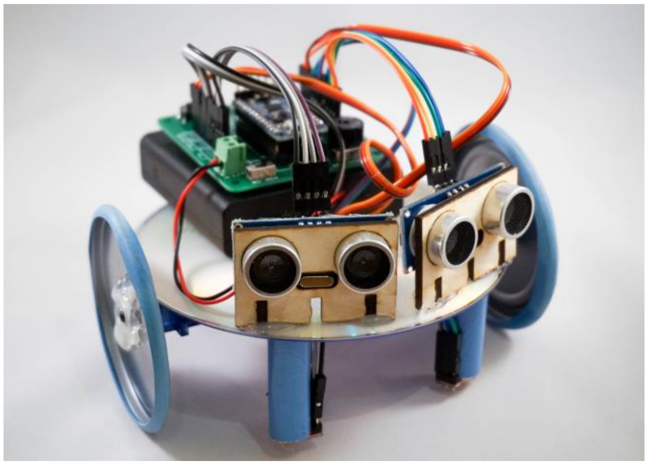

Figure 1. A Sample MIYbot built on a DVD

2. MIYbot components
2.1 Electronics
The MIYbot's electronics is designed around the Adafruit Pro Trinket[1] circuit board. The Pro Trinket is an Arduino compatible circuit board. It is programmed over the built-in USB port and is quite capable at a cost of ~$10. The documentation is extensive and excellent on this circuit board and can be found at https://learn.adafruit.com/introducing-pro-trinket/overview.

As an overview, the MIYbot circuit board (Figure 2) connects the

Pro Trinket to the various robot parts needed to make a functioning robot. Specifically, the board connects the Pro Trinket to the battery, via a power switch, an RGB LED indicator light, two general purpose pushbutton switches, a buzzer, headers to connect 3 servos, headers to connect 2 line sensors, headers to connect two HC-SR04 ultrasonic sensors and headers to connect to the Pro Trinket's I2C bus.

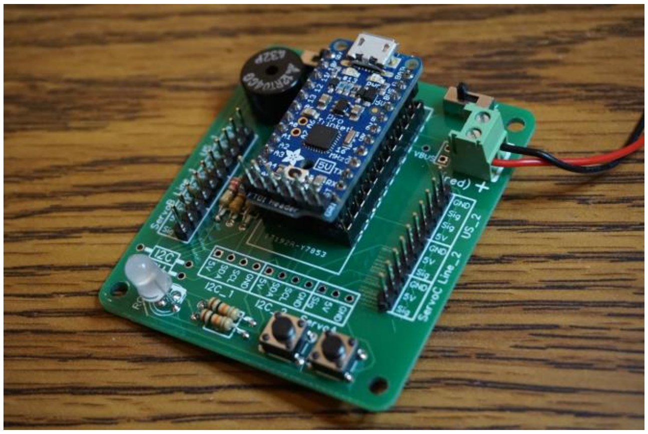

Figure 2. MIYbot Circuit board

2.1.2 Battery System

The battery system for the MIYbot is a battery pack consisting of 4 AA batteries. Alternatively, the MIYbot can be powered through USB port on the Pro Trinket.

When the MIYbot is being powered by the AA battery pack, there is a switch that will turn on the circuit. It is labeled POW_SW. The other switch on the circuit board is labeled SERVO_PWR. This switch chooses where the power to the servos comes from, the USB or the AA battery pack.

One of the nice features of the MIYbot circuit board is that it can be powered using a battery powered, rechargeable USB cell phone charger and a micro USB cable. If this is the case, be sure to set the SERVO_PWR switch to power the servos using USB power.

2.1.3 RGB Indicator Light

The MIYbot circuit board features a common cathode RGB LED light that can be programmed to create any color. This light actually has 3 different LEDs in one package, red, green and blue. Pins D3, D5 and D6 of the Pro Trinket connect these colors via a current limiting 220 ohm resistor. The common cathode to all the LED's is connected to ground. The pulse width modulating (PWM) command analogWrite() command is used to make the diodes light.

2.1.4 Pushbutton switches

The two pushbutton switches are connected as inputs to the Pro Trinket on pins A0 (D14) and A1 (D15). The circuit for the switches uses 10K ohm pull-up resistors for a stable circuit.

2.1.5 Buzzer

The buzzer is connected to pin D4 and can be controlled using the TimerFreeTone library as described later.

2.1.6 SERVO Headers

The servo headers are connected to pins D9, D10, and D11. Using these pins for the servos allows analogWrite() to be used for the RGB LED. The power for the servos is controlled from the SERVO_PWR switch and it feeds the center pin. The outside pins are GND and Sig (D9, D10, D11). Often the Sig wire for a servo is yellow and should be connected accordingly. If it does not work the header can be reversed without damaging the servo. The servos used for the MIYbot are the Feetech FS90R continuous rotation micro servos, but any continuous rotation servo should work.

2.1.7 Line Sensor Headers

The line sensor used by MIYbot are the QTR-1RC line sensors from pololu.com (https://www.pololu.com/product/2459). They are attached to pins D8 and D12. The Pololu QTR line sensor library is used to operate these sensors. Make sure they are attached with GND and Sig connected to GND and OUT on the board. 5V is connected to the center pin on the QTR Sensor.

2.1.8 Ultrasonic Sensor Headers

The ultrasonic sensor headers used by MIYbot are the HC-SR04 easily obtained from ebay.com and other sources. They are connected to pins A2 (D16) and A3 (D17). They are controlled using the NewPing library by Tim Eckel (http://playground.arduino.cc/Code/NewPing). Since these headers connect to analog pins, they could also be used to monitor analog sensor signals such as thermistors, resistive light sensors, or other analog distance sensors.

2.1.9 I2C Bus Headers

The I2C headers and I2C pull-up resistors on the board are connected to A4 (SDA) and A5 (SCL). These were designed on the board for advanced users to connect I2C devices (such as OLED displays) to the Pro Trinket and are an optional feature of the MIYbot. The square pads on the header are +5V.

2.1.10 Pro Trinket Pin D13

Pin D13 on the Pro Trinket is connected to a red LED on the Pro Trinket itself. It can be programmed and used for the MIYbot. The example program in the Arduino programming environment "Blink" will blink this red LED.

2.2 Chassis

The chassis of MIYbot is the body of the robot. Looking at this simple robot's design, the minimum needed to make this robot functional is a body 'plate' to mount the servos, a place to hold the battery pack and the electronics and finally a castor wheel to keep the robot from tilting over.

Discussing each of these components with examples is important because it will give the teacher and student ideas how to build his or her robot. One of the greatest assets to this robot is that it is not a kit and thus each robot will be unique and creative. The components I show in this document are ideas that should be mixed and matched by the student's imagination.Figure 1 is a sample chassis built to test the circuit board. It is just one of an infinite ways the MIYbot can be built.

2.2.1 Chassis Plate

In its simplest form, the chassis of this robot is simply a small plate the size of an index card made out of cardboard, wood or another non-conductive material. On this small plate, the servos can be mounted on the bottom using double-stick tape, hot-melt glue, Velcro or some other adhesive. On the top of the plate, the batteries and electronics can be mounted using the same techniques.

Getting a little more creative, the plate can really be any shape and size (within reason). The plate could be a DVD (Figure 1), a small box, an electrical outlet plate, a flat LEGO plate, Popsicle sticks glued together or a coaster. It is important to realize that as long as the electronics, battery pack, and servos can be mounted to the plate it will work! Students need to look around their house, school, stores and think about what might make a good chassis plate. If one design does not work, they should be encouraged to try and try again.

It is also possible to use the battery holder as the plate for a very minimalist robot, but it might be difficult to change the batteries or reconfigure the robot.

Another option is to use a 3D printed design like the ones found on thingiverse.com. 3D designs can be created using tinkercad.com or other 3D design software.

2.2.2 Servo Mounting

The servos need to be mounted on the bottom of the robot. The easiest way to do this is to simply attach the servo on the bottom of the plate with double-stick mounting tape. If this method is used it can be difficult to remove the servo if that is needed. Another option is to use hot melt glue. With hot melt glue you can remove the servo if needed and pick off the remaining glue. You can also use stronger glues like 5 minute epoxy, but those may never be removable.

Another way to mount a servo on the robot is to screw the servo onto a servo mount or directly on the chassis. Servos come with the screws needed for this type of mount. There are 3D printed servo motor mount designs on thingiverse.com that could be adapted if needed.

There is a small adjustment screw on the bottom of the servo that needs to be set prior to installing the servos. To adjust this screw, load the MIYbot software and run the pink() program. This will tell the servo to go to its center position. Adjust the screw on the bottom of the servo right or left using a jeweler's screwdriver until the servo stops moving. Now you can attach the servo to the body plate.

2.2.3 Battery Pack Installation

The battery pack needs to be mounted on the top or bottom of the chassis plate. It is helpful to have the battery pack located directly over the servos because the weight of the batteries will help the servos and wheels gain traction. Because of the weight of the battery pack, often the pack is the main component that governs the center of gravity for the robot. This is the main reason some robots are built with the batteries under the chassis plate. If the battery pack is mounted too high on the robot, then the robot might tip easier and fall over, especially when the robot is on an incline.

Generally double stick tape, Velcro, or hot melt glue is used to attach the battery pack to the plate.

2.2.4 Electronics Installation

The electronics is a small circuit board that needs to be installed on the plate. Again, the easiest way to install this is to attach with double stick tape, Velcro, or hot melt glue to the plate or the battery holder. Make sure the mounting of the electronics will make it easy to replace the batteries of your robot. There are also screw holes on the circuit board that can be used with standoffs and small machine screws. Pay careful attention that the bottom of the board is not touching metal. The soldering points could short out the circuit board if this is the case.

On top of the electronics board there are headers (connectors) to connect the servos, sensors and the batteries to the board. There is also a USB connector that should not be blocked or it will be difficult to program your robot.

Line and ultrasonic sensors are connected to the circuit board using dupont wires connected to the headers. The sensors are then routed to mounting locations and glued in place. The line sensors need to be mounted in such a way that the sensor is 3-5 mm above the ground. Dowel rods work well for line sensor mounting. Use your creativity to mount these sensors. There is an ultrasonic sensor mount design is included in the MIYbot files.

2.2.5 Wheels

Wheels can be made from just about anything that rolls. Anything from toy wheels, to can and drink lids to laser cut wheels can be used. The center of the wheel needs to be found (a good exercise for a student to discover) and a hole the size of the inner connector for the servo horn needs to be drilled. The servo horn needs to then be press fitted into the hole and glued into place with hot glue or 5 minute epoxy. The servo horn can also be screwed into the wheel but that can prove more difficult. The small black screw that connects the servo horn to the servo should be used to keep the wheel from falling off the servo.

Traction for wheels can be made by installing a wide rubber-band over the wheel's diameter. The rubber band that is usually found on a bunch of broccoli works well for this purpose! Silicone caulk or RTV sealant could also be applied around the outer diameter of the wheel to create a traction surface.

Finally silicon chain-style bracelets could be used to create a tank tread on a robot. This has been documented on the Red Rover instructions on the Adafruit website [2] (https://learn.adafruit.com/trinket-powered-rover/red-rover). Adafruit is the manufacturer of the Pro Trinket and has served as an inspiration for this project.

2.2.6 Castors

A castor acts as a third (or fourth) wheel for this robot. The castor keeps the robot from falling over as it is driving around. Two types of castors exist, those that roll and those that do not. The simplest castors can be made from half of a ball mounted on the bottom of the robot chassis. Then the robot will slide around on this ball. Anything can really be used to do this as long as it slides easily. A half of a ping pong ball might work well. So might a small furniture leg slider or perhaps a dowel rod end that has been sanded into a half dome.

Rolling castors are a little more involved. The simplest castors are rolling wheels that can be purchased at a hardware store. Other rolling castors can be made from marbles or steel ball bearings housed in 3d printed enclosures or other fashioned housings that allow the ball to roll freely. There are several 3D printed castor designs at thingiverse.com. I have included a 3D castor design in the MIYbot files.

Once a castor design has been decided upon, the castor is then attached to the bottom of the robot using hot glue or similar. Tank type robots will not need a castor because they are already supported.

2.2.7 Useful materials

When running a MIYbot class, it is a good idea to have these materials on hand: ⅛ inch Baltic Birch plywood, Foamboard, Acrylic, Cardboard, Chipboard, Recycled materials, Cutting boards, Circuit board standoffs and machine screws, dupont wires, Silicon bracelets, 100% silicone caulk, RTV automotive sealant, Marbles (large and small), Steel ball bearings (½ inch or greater), Large broccoli rubber bands, Hot glue or hot melt glue and a glue gun, Double Stick mounting tape, Superglue, Velcro, 5 minute epoxy, and of course, Duct Tape.

2.3 MIYbot software

The MIYbot is easily programmed thanks to the fact that the main "brains" is the Adafruit Pro Trinket microprocessor circuit board. The Pro Trinket is part of the popular open-source Arduino family of embedded processor boards based upon the Atmel 328 processors. The Pro Trinket is heavily documented here: https://learn.adafruit.com/introducing-pro-trinket/overview and should be reviewed carefully.

As mentioned previously, the Pro Trinket is an Arduino; this means it can be programmed using the Arduino integrated development environment (Arduino IDE). Simply stated, the Arduino IDE is software that runs on Windows, Mac and Linux and allows the user to write, edit, compile and upload programs to Arduino (and thus the Pro Trinket) circuit boards.

2.3.1 Getting Started

First, since the MIYbot is based upon the Pro Trinket [1], the instructions for programming a Pro Trinket should be followed. Begin by visiting web page https://learn.adafruit.com/introducing-pro-trinket/starting-the-bootloader and downloading the latest version of the Arduino specific to the Pro Trinket. Follow the instructions for uploading code via the USB bootloader and not the FTDI cables. You will also have to download and install the USB drivers using the directions on the same web page.

Once the Arduino IDE and drivers are installed, test the programming of the Pro Trinket by following the "Blink" instructions on page https://learn.adafruit.com/introducing-pro-trinket/setting-up-arduino-ide. If all goes well you will be program the #13 LED on the Pro Trinket to blink on and off. One common problem when programming the Pro Trinket is failing to press the reset button prior to compiling and uploading your code. Make sure you can upload and run the Blink code successfully prior to continuing. If you have trouble getting this working, re-read the Adafruit Pro Trinket web pages and also query the Adafruit forums.

2.3.2 Installing the MIYbot libraries

The MIYbot robot relies on several libraries to run the ultrasonic sensors, the line sensors and the buzzer. You will need to install these libraries into the Arduino IDE.

For the Ultrasonic Sensor, download the latest NewPing library zip file. The download can be found here http://playground.arduino.cc/Code/NewPing Once the zip file is downloaded, the library is installed by Arduino IDE menu item Sketch->Import Library->Add Library and choosing the NewPing.zip file you downloaded.

For the Line Sensor, download the latest QTR-RC Arduino library zip file. The download can be found here https://github.com/pololu/qtr-sensors-arduino Once the zip file is downloaded, the library is installed by Arduino IDE menu item Sketch->Import Library->Add Library and choosing the QTRSensors.zip file you downloaded. More information on these sensors can be found here: https://www.pololu.com/product/2459

The buzzer needs to use the library TimerFreeTone because of conflicts with the servos and the PWM routines needed for the tri-colored LED to work. Download the latest TimerFreeTone library here: https://bitbucket.org/teckel12/arduino-timer-free-tone/downloads More information about the TimerFreeTone library can be found here: https://forum.arduino.cc/index.php?topic=235774.0 Once the zip file is downloaded, the library is installed by Arduino IDE menu item Sketch->Import Library->Add Library and choosing the TimerFreeTone.zip file you downloaded.

2.3.3 Installing the MIYbot software

The latest MIYbot software and documentation can be downloaded on github page: https://github.com/bpwagner/MIYbot. You will find three files, MIYbot.cpp, MIYbot.h and MIYbot1.ino in a folder called MIYbot1. Put these three files in the same folder named MIYbot1 anywhere on your computer. You can load the MIYbot software into the Arduino IDE by double clicking the MIYbot1.ino file or using File-Open and choosing MIYbot1.ino. All three files need to be in the same folder!

Try compiling the software by clicking the check-mark icon. If it does not compile, make sure the libraries are all installed properly. You can also upload the software to your Pro Trinket if you want.

2.3.4 MIYbot code

The MIYbot is software that has evolved over the past three years. It is still a work in progress so you may see new features added and bugs fixed periodically. The software is contained in three files, MIYbot.cpp, MIYbot.h and MIYbot1.ino. MIYbot.cpp and MIYbot.h are library files used to abstract the entire robot so the programming of the MIYbot for novice programmers is easy. The MIYbot1.ino file is the Arduino "main" program that controls the robot.

If you look at the MIYbot.h file you will find the compiler #defines that describe the Arduino pins that the robot uses as well as many constants that are useful for programming (like the frequencies for musical notes) the MIYbot. You will also see the function definitions for the robot functions like motorRun(), setRGB() and beep(). The actual code for these functions is contained in MIYbot.cpp.

The MIYbot1.ino file contains the functions needed to run the MIYbot. The two functions void setup() and void loop() are common to all Arduino programs. Setup() is used to initialize the variables as needed and start the Arduino program. In the MIYbot's case, the code initializes a few variables, the serial monitor and the servo configuration. Then it flashes the blue, green and red LEDs and beeps twice.

The loop() function works like a simple menu for the robot. Basically, the robot will cycle through many colors of the rainbow using the RGB LED by pressing one button and then runs a function when the other button is pushed. This way many programs can be written and run on the MIYbot. The remaining functions are colors: red(), orange(), yellow(), green(), blue(), purple(), white(), pink() and aqua(). As the menu cycles through the colors and one is chosen, the corresponding function is called.

Currently the colors are programmed as follows:
red() - drive around with the ultrasonic sensors, avoiding obstacles
orange() - test the ultrasonics. LEDs change colors when sensors 'see something'
yellow() - test the line sensors. Cover the line sensor with a finger to see LED colors change
green() - SumoBot fight - not programmed yet
blue() - To be programmed by student
purple() - To be programmed by student
white() - To be programmed by student
pink() - turn on servos so they can be centered using the tiny screw on the bottom of the servo.
aqua() - play the theme to Mario on the buzzer!

There is a small adjustment screw on the bottom of the servo that needs to be set prior to installing the servos. To adjust this screw, load the MIYbot software and run the pink() program. This will tell the servo to go to its center position. Adjust the screw on the bottom of the servo right or left using a jeweler's screwdriver until the servo stops moving. Now you can attach the servo to the body plate.

One final thing: The configuration of the servos needs to be set so the robot drives forward when it is programmed to drive forward. Depending on how the servos are physically mounted to the robot, this configuration can be one of 4 ways. The ways are both servos forward, left servo forward and right servo reversed, left servo reversed and right servo forward and both servos reversed. The configuration is changed near the top of the MIYbot1.ino file in the lines below. If your robot spins around instead of driving forward, you probably need to fix this!

```
//********************************************
// Please set this value to 0, 1, 2, or 3
// Controls what initial direction of the servos
// 0 = both servos fwd
// 1 = left fwd, right reverse
// 2 = left reverse, right fwd
// 3 = both servos rev
//
int Servo_Config = 2;
//
//
//********************************************
```

3.0 Teachers Guide

Running a class where your build and program a MIYbot (Make It Yourself robot) is an ambitious project indeed. As schools push for more and more STEM (Science, Technology, Engineering and Mathematics) a robotics project is often referenced. Most of the time robotics classes utilize Lego NXT robots or VEX robots. These are good robot systems but they are expensive and cannot be taken home at the end of the year. MIYbot solves these problems because a MIYbot can be built for about $35 and students are encouraged to truly make the robot their own. They design and build their own robot chassis and program their robot to have its own personality. Plus (and perhaps most importantly) the MIYbot should be taken home at the end of the class to encourage the student to continue tinkering, playing with and building their robot.

3.1 Documentation and Files

All of these files, and perhaps more, are stored on https://github.com/bpwagner/MIYbot. The following documentation was written to help run a successful MIYbot class:

- Chassis Designs - this document discusses the design of the robot body. It gives ideas on how to build the body plate, the wheels, and the castor and how to mount the sensors.
- Electronics Build Document - this document gives step by step instructions on how to solder the circuit board that controls the MIYbot
- MIYbot Electronics - this document describes in detail how the electronics work for the MIYbot.
- MIYbot software - this document describes how to install and program the Arduino IDE and the MIYbot software on the robot. It also describes the functions of the robot.
- MIYbot Circuit Board BOM - this spreadsheet includes all the parts and pieces needed to build a MIYbot in a classroom setting. It also includes several other useful tools needed, like soldering irons.

The following files are also included

- MIYbot2.dip, MIYbot2.dch - these files are Diptrace design files for the circuit board. If you need to make changes to the circuit board you can use Diptrace to make the changes. Diptrace also generates the Gerber and Drill files needed to manufacture the circuit board. Diptrace can be downloaded and purchased at www.diptrace.com.
- MIYbot2Gerbers.zip - these are the Gerber files needed to have the circuit board manufactured. You can view them with the free program GERBV if you want to take a look. http://gerbv.geda-project.org/
- MIYbot.cpp, MIYbot.h, and MIYbot1.ino - the actual code to run the MIYbot. See the documentation on MIYbot software for more information
- HC-SR04-bracket.svg - Nice laser cut bracket that can be useful for mounting the HC-SR04 ultrasonic sensors.
- castor.scad and castor.stl files - these are 3D designs for making a castor that uses a large marble. The stl files can be printed on a 3D printer.

3.2 Timeline

To run a successful class building MIYbots, the following schedule is suggested. Times are based upon experience but your mileage might vary. Each class is 45 minutes meeting once per week.

3.2.1 Summertime

Before school familiarize yourself with soldering by taking a class, being taught, practicing, watching youtube.com etc. Be sure you know how to clear holes in a circuit board and know how to remove components. There are many soldering 'experts' in the community, be sure to find one and have them help you.

Purchase soldering supplies for the class. Don't forget safety glasses.

Purchase the components for the robots. Many of the components can be cheaper if purchased through ebay.com. If the component is coming from China, be sure to leave plenty of time for shipping. You can also choose eBay suppliers from USA in the search options on eBay. A list of all of the components needed (and suggested suppliers) is found in the MIYbot Circuit Board BOM document.

Order the circuit boards. smart-prototyping.com is a good supplier out of Hong Kong. Just choose to make a circuit board and then upload the MIYbot2Gerbers.zip. When you place the order, choose 10cm X 10cm, e-test 100% and leave the other options at default. You can choose gold surface finish and a different solder mask (board) color if you want. Ship using DHL from China. It will take about 3 weeks.

When the parts come in, it is highly recommended that you build and program a MIYbot robot yourself so you can see what is required!

3.2.2 First month (3-4 45 minute classes)

Teach the students to solder by making a simple soldering project. There are many simple soldering kits that would be appropriate. Velleman makes some inexpensive kits that are easy to solder: http://www.vellemanusa.com/products/list/?id=342509 It is important to have the kids practice before building their MIYbot Board. Also, have students collect materials that might be useful for building MIYbot bodies.

3.2.3 Second month (3-4 45 minute classes)

Solder the MIYbot circuit board. Install the headers on the Pro Trinket and plug into the MIYbot board. Follow the instructions in the Electronics Build Document. Continue to have students collect materials that might be useful for building MIYbots.

3.2.4 Third and Fourth month (5-8 45 minute classes)

Use the Chassis designs and MIYbot software document directions to build a chassis for your robot and program it to run. These months are where the real engineering happens. Encourage your students to design, test, build and redesign their robots to continually make them better. Encourage them to program their robot to play songs, dance, follow lines and fight SumoBot (Google it!) style. For the chassis design, it is useful to have many recycled materials from which to choose. Have students bring in useful robot parts during the first months of the class. If you have 3D printers and/or a laser cutter, it is useful to have castors and sensor mounts made so the students can use them. You can also use a laser cutter to make wheels for the robots.

Have your students name their robot. Robots are sad without names.

3.2.5 Fifth month

Have a robot show to celebrate student accomplishments! Awards could be given to students with Best Soldering Job, Most Creative MIYbot, Most Unique New Software Feature, Best Overall, Best SumoBot, etc. Create contests and celebrate!

4. About the Author

I have been teaching robotics classes at Kentucky Country Day school in Louisville Kentucky for the past 5 years. My robot classes began as a trimester elective using Lego NXT robots. Now it is a full year class for both the 6th and 7th graders. My 6th graders still use Lego NXT robots and my 7th graders, for the past two years, have been building robots of my own design. The MIYbot is the third iteration of this "BaxterBot". Much of the curriculum I am using in my 7th grade classes has never been written down (until now) and could not be reproduced by another teacher. I hope to add to the Educational Tech community by sharing the robot knowledge that I have gained from designing my 7th grade robot, and in the past 5 years as a robotics teacher.

I am also working on my Masters in Computer Science in the OMSCS program at Georgia Institute of Technology. Much of the writing for this paper has be in conjunction with CS 6460 Educational Technology by Professor Dr. David Joiner

3. ACKNOWLEDGMENTS

My thanks to Dr. David Joiner for leading such an interesting class on Educational Technology. Also thanks to Lulu Cheng, my mentor for the class.

4. REFERENCES

[1] Fried, L. (n.d.). Introducing Pro Trinket. Retrieved December 2, 2015, from https://learn.adafruit.com/introducing-pro-trinket/overvie

[2] Winscot, R. (n.d.). Trinket Powered Rover. Retrieved December 2, 2015, from https://learn.adafruit.com/trinket-powered-rover/red-rover