

Arduino Robotics Camp

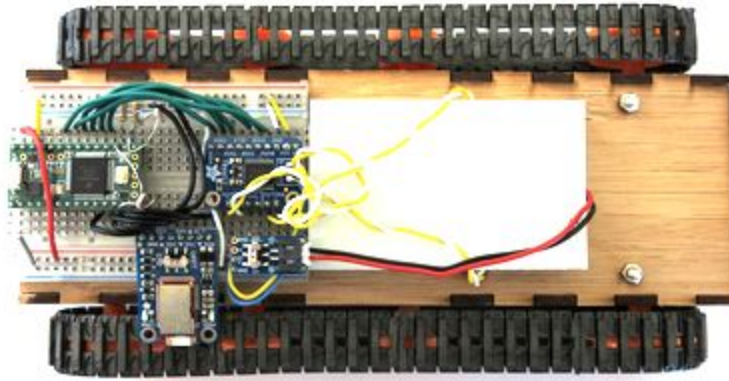
v1.0

16 March 2016

Introduction

Welcome to Arduino Robotics camp, instructor! Over the next five days, your students will be designing, constructing, testing, assembling, and playing with their very own autonomously- and directly-controlled tracked robots, which they'll be controlling from a mobile device. This involves laser cutters, 3D printers, Arduino IDE, and lots of wiring.

Throughout this project, the code base should match the wiring of the schematics - see Appendix A. If you prefer to wire the system differently, just make sure to adjust the code accordingly.



Required hardware (provided by TechShop):

- pjrc's [Teensy 3.2](#)
- Adafruit [BlueFruit LE UART Friend](#)
- Adafruit [TB6612 Dual Motor Driver](#)
- [Lithium polymer battery](#) (500mAh+)
- Tamiya [twin motor gearbox](#)
- Tamiya [track and wheel set](#)
- [Breadboard and jumper kit](#)
- Components from Arduino Wiring (formerly EEE205: Arduino Part 2)
- Laptop or desktop computer
- 24" x 18" x 1/8" acrylic or wood sheet (one sheet per eight students)
- Adhesive (acrylic cement, superglue, etc)
- Mounting screws and nuts

Required hardware (provided by student):

- Mobile device with Bluetooth 4.0 capability

Required software and documents:

- [Arduino IDE](#) (1.6.5+)
- [Teensyduino](#)
- Adafruit Bluefruit LE Connect ([Android](#) or [iOS](#), depending on student's device)
- [Arduino Robotics Camp Github repo](#)

Daily Schedule Summary

This schedule is subject to change based on instructor's analysis and discretion. There is time on both Thursday and Friday for advanced topics if the students are ahead of schedule, or for brevity of topic if the students are behind schedule.

Monday, day 1: A primer on electronics (page 3)

Introduction
Chat about general electronics as well as robotics
Play with littleBits
Install software (computer and mobile device)
Teach Arduino Wiring (formerly EEE205: Arduino Part 2)
For tomorrow: discuss DC motors

Tuesday, day 2: Make some movement! (page 4)

Gearboxes
Solder motor and motor driver connections
Explain and connect motor driver to MC and gearbox
Code the gearbox
"Drive" the breadboard
For tomorrow: discuss mechanical engineering and chassis design

Wednesday, day 3: Industrial design and fabrication (page 5)

Abbreviated laser cutter course
Illustrator tutorial
Design the etched parts of the chassis
Laser cut the chassis
Cement the chassis (with gearbox installed)
Assemble the track and wheel set
Test the tank
For tomorrow: discuss Bluetooth and autonomy

Thursday, day 4: Programming Bluetooth and autonomous motion (page 7)

Discuss Bluetooth connectivity
Connect and begin coding Bluetooth
For tomorrow: finally bringing it all together, and challenges

Friday, day 5: Final tweaks and tank challenges (page 9)

Code cleanup and tweaks, plus Github
Wrap up any final details
Design and build challenge course
Run the challenge course
Show and tell!

Appendix A: Full Schematic (page 10)

Daily Schedule Expanded

Monday, day 1: A primer on electronics

Overview

Today is all about getting an understanding of basic electronics and microcontrollers. By the end of the day, students will have the Teensy microcontroller up and running, and play with its inputs and outputs via Arduino's Example sketches - this is effectively an extended Arduino Wiring course, with a different microcontroller.

Hardware introduced today:

- Teensy 3.2
- breadboard and jumper kit
- components for Arduino Wiring course

Introduction

Introduce yourself as well as the camp. Outline what they will be doing during the week. Ask the students if they have experience with electronics or robotics. Hand out hardware.

Play with littleBits

The basic kit will suffice. This will likely be a review for older students, but completely new for younger students. Be sure to distinguish among power, input, and output.

Install software (computer and mobile device)

Each student will need to have a computer with the following installed:

- Arduino IDE version 1.6.5+
- Teensyduino

as well as a Bluetooth 4.0 capable device. The Arduino IDE is for writing code for the Teensy; Teensyduino is for programming the actual hardware. You may substitute microcontroller; the code will work on any Arduino-compatible hardware. Only the pin assignments will need to be adjusted to the particular hardware.

Go through Arduino Wiring (formerly EEE205: Arduino Part 2)

Connect the Teensy to the computer and make sure the IDE will upload the Blink sketch to it. The process is almost identical to a standard Arduino board, except that once the IDE is done compiling, Teensyduino will launch. If Teensyduino is in "Auto" mode, the code should immediately be uploaded, and the software will say "Reboot OK". If Teensyduino is in manual mode, (or if the code does not automatically upload), press the "Reset" button on the Teensy.

From here, Arduino Wiring can be taught, paying particular attention to analog inputs and outputs. If students are advanced, there should be time to let them experiment with the codebase.

For tomorrow: discuss DC motors

Note that Arduino Wiring does not address motor control. Have the students chat about controlling motors directly with the Teensy; is this even possible?

Tuesday, day 2: Make some movement!

Overview

Today is all about building and integrating the gearbox into the MC environment. Students will assemble their gearboxes, and then connect and drive them via the Teensy. There is quite a bit of conceptual and contextual material today.

Hardware introduced today:

- Gearbox
- Motor driver

Gearboxes

Tanks have no steering column, as cars do; so how do they move and turn? Introduce gearing in general, and then the gearbox. Instructions on how to assemble the gearboxes are given with the kit. They can be assembled in three different configurations; the “C type” is the ideal configuration, as it provides the highest torque. This assembly can be frustrating to students, as the parts are small, and requires slow and careful reading.



Solder motor and motor driver connections

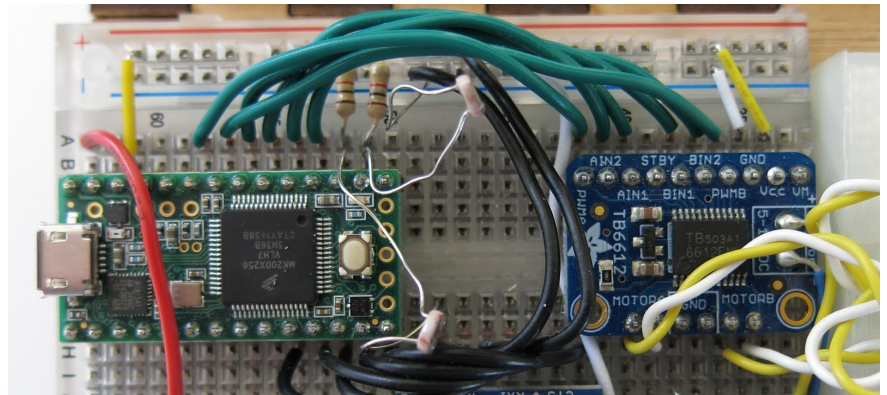
(If students have never soldered before, give a brief safety overview and a tutorial on how to solder.) The leads from the motors should be approximately 15cm long, and should be left unconnected on the opposite end. Be careful, as the motor leads are thin and fragile! Power leads should be soldered directly from the motor driver's power pins; they should be approximately 5cm in length, and also left unconnected on the opposite end. The male header pins can be soldered directly to the motor driver.

Explain and connect motor driver to MC and gearbox

How do you get a motor to spin? How do you get it to then spin in the opposite direction? Get students to brainstorm why a motor driver is required. Hook up the motor drivers as shown below. (There is a full schematic of the robot in Appendix A of this packet.)

Code the gearbox

Ask students to conceptualize how to spin the motors in an arbitrary direction. The goal is to have them code four functions: `forward()`, `backward()`, `left()`, and `right()`, and use those functions to write a “directionDemo” sketch, which loops a fixed command set. See [github repo](#) for example code. Instructor is free to provide as little or as much of the sample code to students, dependent upon their coding skill level.



“Drive” the breadboard

Gently tape the gearbox to the bottom of the breadboard. Take out a pair of drive wheels (they have teeth) from the track and wheel set; the diameter does not matter. Attach them to the gearbox axles, and have the students upload their “directionDemo” code. Experiment.

For tomorrow: discuss mechanical engineering and chassis design

As students how they will turn this setup into an actual driving system. Introduce mechanical engineering as a concept, and talk about fabrication. Have students begin to think about how they want their chassis styled.

Wednesday, day 3: Industrial design and fabrication

Overview

Today students will learn the basics of CAD and design the rastered portion of the chassis, then actually go through the fabrication process, and assemble the chassis with the gearbox mounted.

Hardware introduced today:

- Laser cutter
- Track and wheel set

Abbreviated laser cutter course

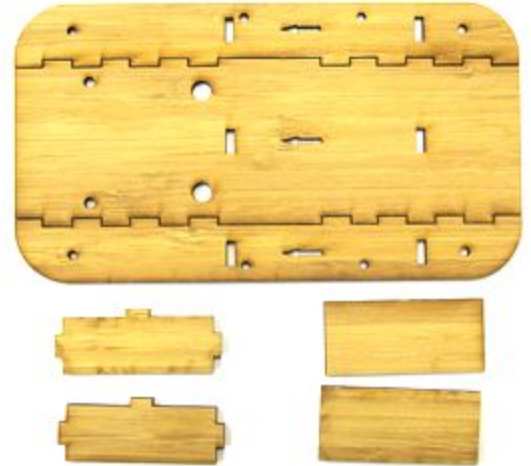
Teach an abbreviated laser cutter class, so students can contextualize what they'll be doing today.

Illustrator tutorial

Give a simple vectoring and rastering lesson.

Design the etched parts of the chassis

Students can then begin to design their chassis based on the template called "Chassis Design" - this is found in the Github repo under "Support Files". Make sure students do not add any stray vectors which would cut through the chassis.

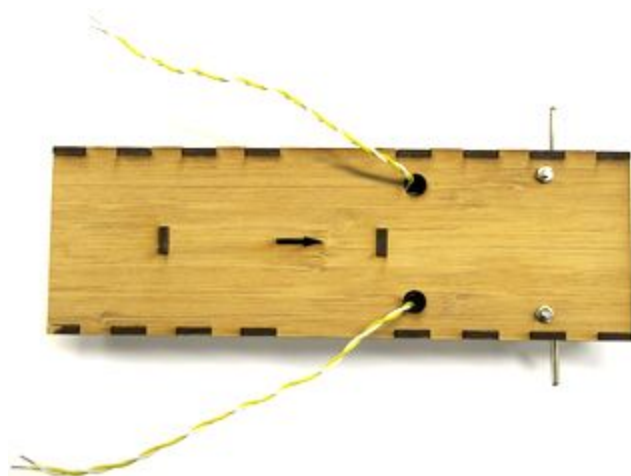


Laser cut the chassis

You may prefer to combine students' designs into one document to save time. Up to 8 chassis can fit on one 24" by 18" sheet.

Cement the chassis (with gearbox installed)

The chassis is designed with the gearbox permanently installed. To do this, first disconnect the motor leads from the breadboard. Start assembling the chassis by cementing one of the long runners to the rectangular base, then cement the lower upright supports; run the motor leads through the chassis base holes, and insert the gearbox into the chassis; then mount the second long runner. For best adhesion, the cement must set for several hours.



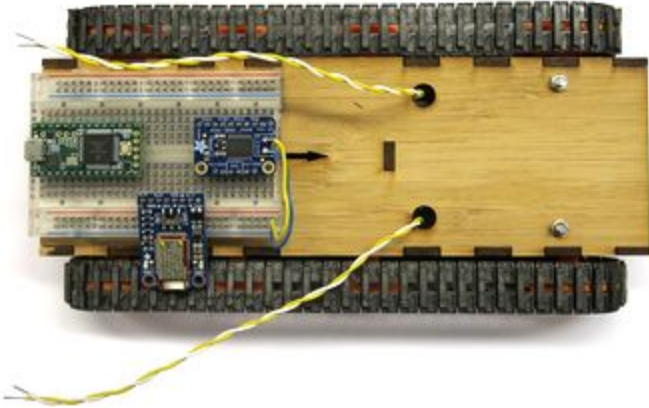
Assemble the track and wheel set

The tank uses the following from the track and wheel set:

- 1 long, 2 medium, and 2 short track sections (per track)
- 2 large drive wheels (or "sprockets")
- 2 large idler wheels
- 4 medium idler wheels
- 3 round idler axles
- 8 end caps



Additionally, the caps on the end of the drive wheels will need to be clipped, and the plastic on each idler wheel which covers the axle will need to be snipped ever so slightly, so that the track will align with the wheels.



Test the tank

Students can wire the gearbox back into the breadboard and see their tank move for the first time. Just be mindful of adhesive set time. Additionally, make sure to wire the battery connector board into the system; as always, there is a full schematic in Appendix A.

For tomorrow: discuss Bluetooth and autonomy

Now that the chassis is fully built and the basic electronics are in place, discuss how to improve upon the code base. How could they communicate with it remotely? What other modes might they want their tank to operate in?

Thursday, day 4: Programming Bluetooth and autonomous motion

Overview

Students' tanks currently only operate in a "direction" demo mode, which shows simple, repetitive movement. Today they'll be overhauling that code to allow two modes of functionality: manual control (via Bluetooth) and autonomous light-following. Ask students how they would go about getting the robot to follow light. *This is possibly the most important exercise in understanding how to fundamentally program robots!*

Hardware introduced today:

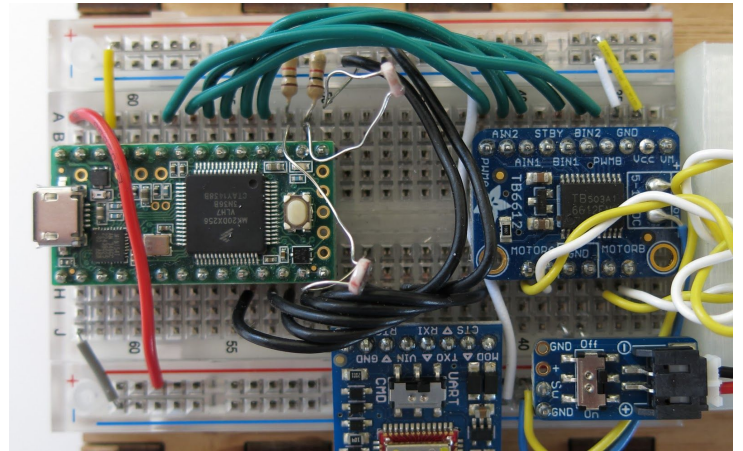
- adafruit Bluefruit LE UART friend
- photoresistors

Discuss Bluetooth connectivity

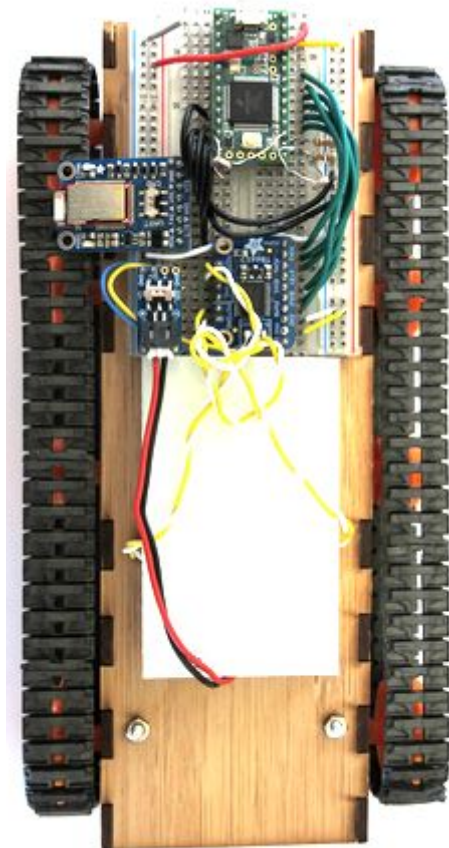
The Bluetooth boards communicate through UART (aka the same old type of serial connection that Arduino uses to communicate to the computer through the Serial Monitor).

Connect and begin coding Bluetooth

For the Bluetooth board, In addition to the obvious RX, TX, and power pin connections, you'll need to connect up the MODE, RTS, and CTS pins. Connect up two photoresistors as standard voltage divider sensors as well. The final wiring should look something like the picture to the right.



Coding for Bluetooth is already in skeleton form in the repo's sample code. Students will need to once again fill in sections that allow the robot to actually move, except that this time the movement functions have been abstracted into only two places: `motorSpeeds()` and `motorDirections()`. Discuss how abstraction helps to make their code more efficient. The sensor code is already completed; give as much or as little to the students as you deem appropriate.



The students should install Adafruit's Bluefruit app on their Bluetooth device; from there, they should be able to pair with their robot and use the "Controller" mode to issue it meaningful commands.

Ultimately, their robots will look something like the picture to the left. It may take a bit of fiddling to get the entire system to work properly; with so many wires going to so many places, it's very easy to misconnect things. This tends to be the place where students have the most trouble.

For tomorrow: finally bringing it all together, and challenges

The tracked robot is fully operational! Tomorrow's time is for students to tidy up any loose ends, experiment, and show off their new devices.

Friday, day 5: Final tweaks and tank challenges

Overview

The last day of camp sees students doing final tweaks on their tanks, and then putting them to a test! If you have extra time today, you can cover 3D printing, Github repos, and/or other modes for the robotic tank.

Hardware introduced today:

- none

Code cleanup and tweaks, plus Github

Make sure the students' codebases are as bug-free as possible. Have them comment generously so that they understand what they've done, and can reference it in the future. If you have extra time, have them create a Github account (if they don't have one already) and push their very own new repo.

Wrap up any final details

Time has been allotted for catch up, in case students are behind schedule. If they are on schedule and you have more sensors, they can design and add a third mode to their tanks (eg. sonar, pathfollowing, etc). Make sure they push to the repo again afterward. If they are ahead of schedule, an abbreviated course on the 3D printers can be taught, so that students can design and print a battery case for their robot.

Design and build challenge course

Have students design and build an obstacle course for their tanks to manually navigate through. Assuming the gearboxes have been build in the "C-type" configuration, the tanks can climb over surprisingly large obstacles, and traverse inclines at upwards of 45° (if the incline has a sufficient coefficient of friction).

Run the challenge course

If students want to compete, have them each run the obstacle course individually for time.

Show and tell!

Show off the tracked robots to the rest of TechShop, and to the students' parents! Have them explain to their parents how the system is connected together, and what they learned.

Appendix A: Full Schematic

