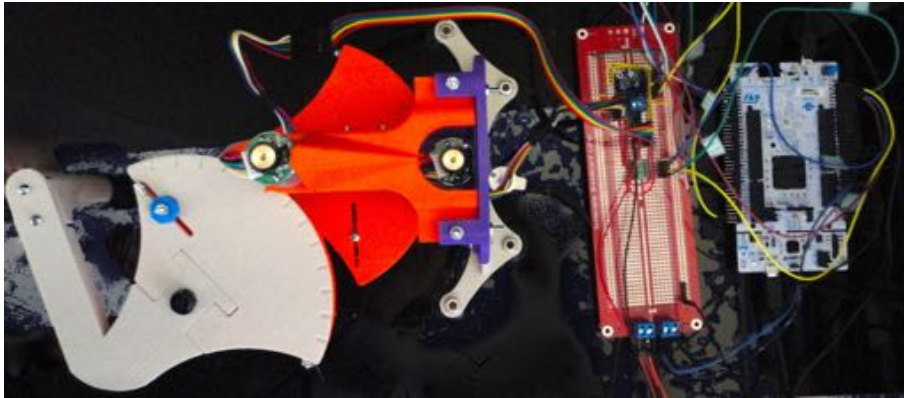


Haplink 2.0 Electronics

Version 10.10.2017



For Haplink 2.0 we chose to use a faster processor than the Arduino Uno which is why we adapted our electronics to be able to use the NUCLEO-F446ZE. We also changed our motors to motors with encoders due to the fact that the coupling of the sensing channels was a limiting factor to render high-stiffness virtual environments for Haplink 1.0. We chose as our motors Pololu's MP 12V Motor with 48 CPR Encoder for 25D mm Metal Gearmotors with No Gearbox. And had to add extra circuitry as the encoder on the motor works with 5V logic but our microcontroller works with 3.3V logic.

In the following sections we describe the main components of the electronic assembly for Haplink 2.0 as well as provide assembly instructions.

1 Electronic Components

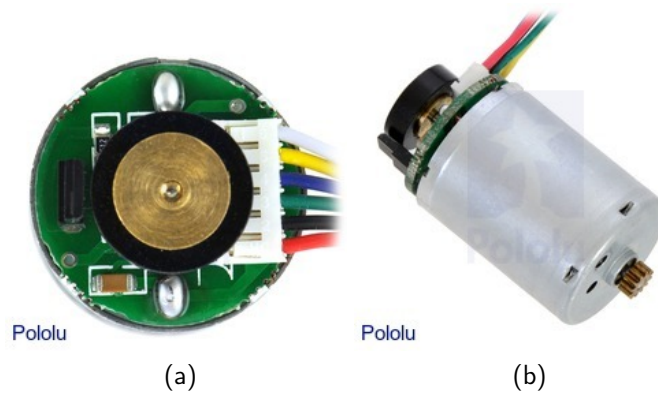
1. **MOTOR: Pololu's MP 12V Motor with 48 CPR Encoder for 25D mm Metal Gearmotors with no gearbox.**

This motor has 7800 RPM, 50 mA free-run, and 2.1 A stall current at 12V. It is sold by Pololu at \$ 21.95.

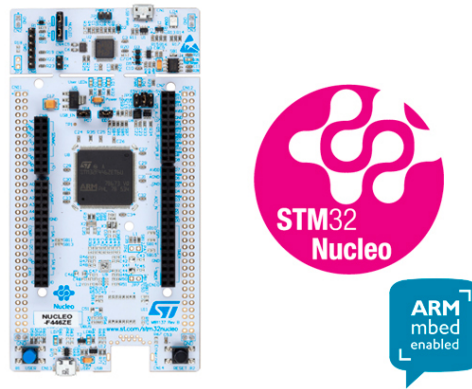
The encoder on the motor is composed of a two-channel Hall effect encoder which is used to sense the rotation of a magnetic disk on a rear protrusion of the motor shaft. The quadrature encoder provides a resolution of 48 counts per revolution of the motor shaft when counting both edges of both channels.

There are 6 wires which compose the inputs and outputs of the motor assembly. The following table describes the wire functions:

Red	motor power (connects to one motor terminal)
Black	motor power (connects to the other motor terminal)
Green	encoder Ground (GND)
Blue	encoder Vcc (3.5 – 20 V)
Yellow	encoder A output
White	encoder B output



2. **MICROCONTROLLER: NUCLEO-F446ZE.** The nucleo F446ZE is an stm32 nucleo-144 development board with an STM32F446ZE MCU. This MCU is very powerful. It has 512kB of Flash memoy and 128 kB of SRAM. It uses a 32-bit ARM Cortex-M4 CPU and has the capability to run at up to 180 MHz. The development board also integrates an on-board ST-LINK debugger and programmer which allows us to program it easily using the online mbed application.

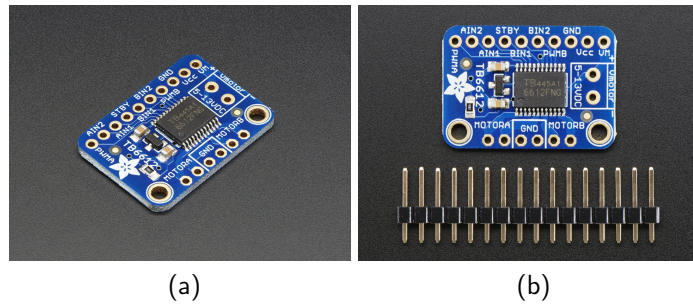


(a)

3. **MOTOR DRIVER: Adafruit TB6612 1.2A DC/Stepper Motor Driver Breakout Board.** This Adafruit motor driver is contains a TB6612 soldered onto a breakout board, with a polarity protection FET on the motor voltage input and a pullup on the "standby" enable pin. Each breakout chip contains two full H-bridges (four half H-bridges). That means you can drive 2-4 solenoids (only two can be active at a time, on opposite bridges), two DC motors bi-directionally, or one stepper motor. Just make sure they're good for 1.2 Amp or less of current, since that's the limit of this chip. They do handle a peak of 3A but that's just for a short amount of time, like 20 milliseconds. This driver comes with built in kick-back diodes internally so you don't have to worry about the inductive kick damaging your project or driver.

There's two digital inputs per H-bridge (one for each half of the bridge) as well as a PWM input per driver so you can control motor speed. Runs at 2.7V-5V logic. The motor voltage is separate from the logic voltage and it is good for motor voltages from 4.5V up to 13.5V. This wont work well for 3V motors.

Comes as one assembled and tested breakout plus a small strip of header. You'll need to do some light soldering to attach the header onto the breakout PCB.



4. **LEVEL SHIFTER: Pololu's 4 channel bidirectional level shifter.**

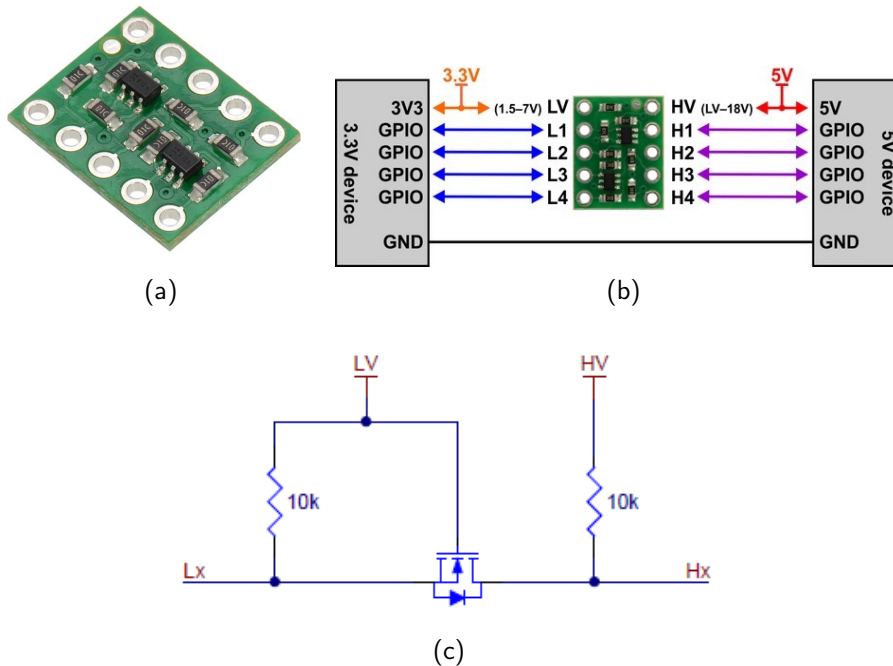
As we mentioned before, the encoders on the motors work with 5V logic, however, our microcontroller works with 3.3V logic. In order to allow the microcontroller to read the output of the encoders then, we added a level shifter in between the two devices.

It works like this:

When Lx, the lower-voltage input, is driven low, the MOSFET turns on and the zero passes through to Hx.

When Hx, the higher-voltage input, is driven low, Lx is also driven low through the MOSFET's body diode, at which point the MOSFET turns on.

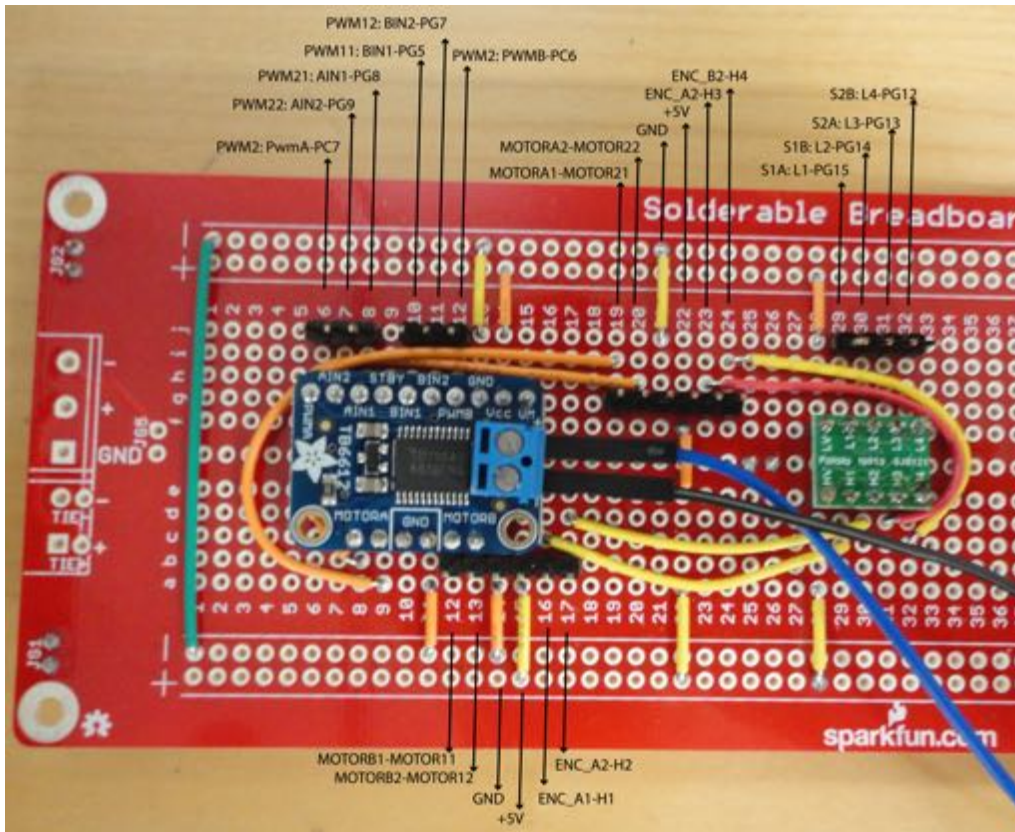
In all other cases, both Lx and Hx are pulled high to their respective logic supply voltages. The circuit works for any pair of voltages (within the limitations of the MOSFET) and can be used with most common bidirectional and unidirectional digital interfaces, including I²C, SPI, and asynchronous TTL serial.



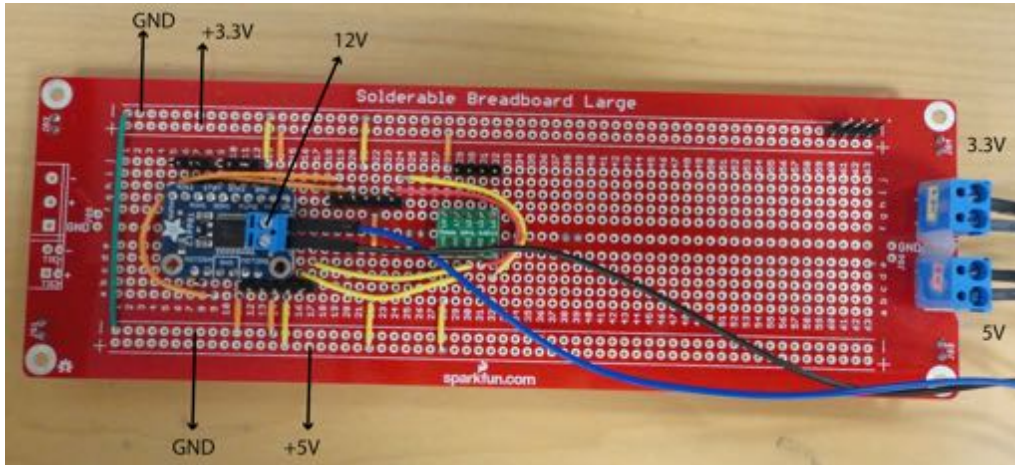
5. **ELECTRONICS BOARD:** This board contains the motor driver and level shifter.

Figure (a) outlines the connections that need to be made between the electronics board and the microcontroller board and the motors.

Figure (b) shows the power rails of the electronics board.



(a)

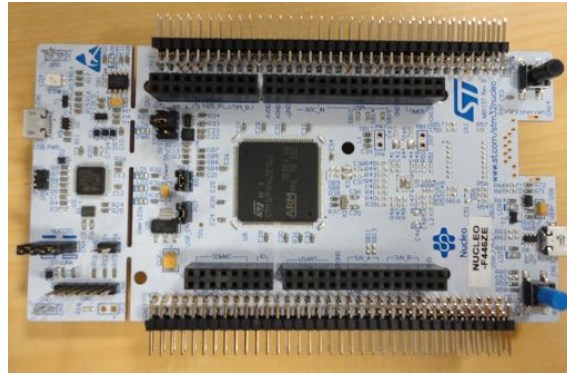


(b)

2 Assembly Instructions

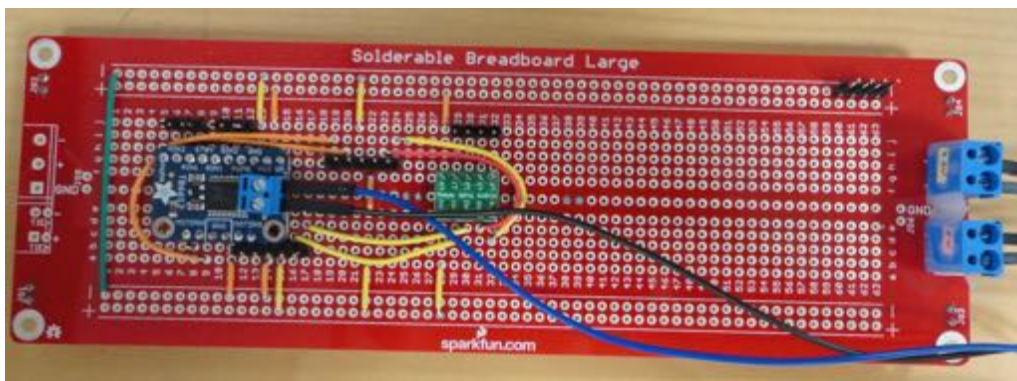
As with the mechanical assembly instructions, we will first connect Hapkit, and then we will connect Haplink.

1. **Obtain all the Electronic Components:** Consult the parts list provided on the website.
2. **Solder headers to your Nucleo Board:** Your Nucleo Board only comes with a reduced amount of pins with headers. (a) You will need to solder headers to bring out the other pins. We recommend you use right-angle breakaway headers.



(a)

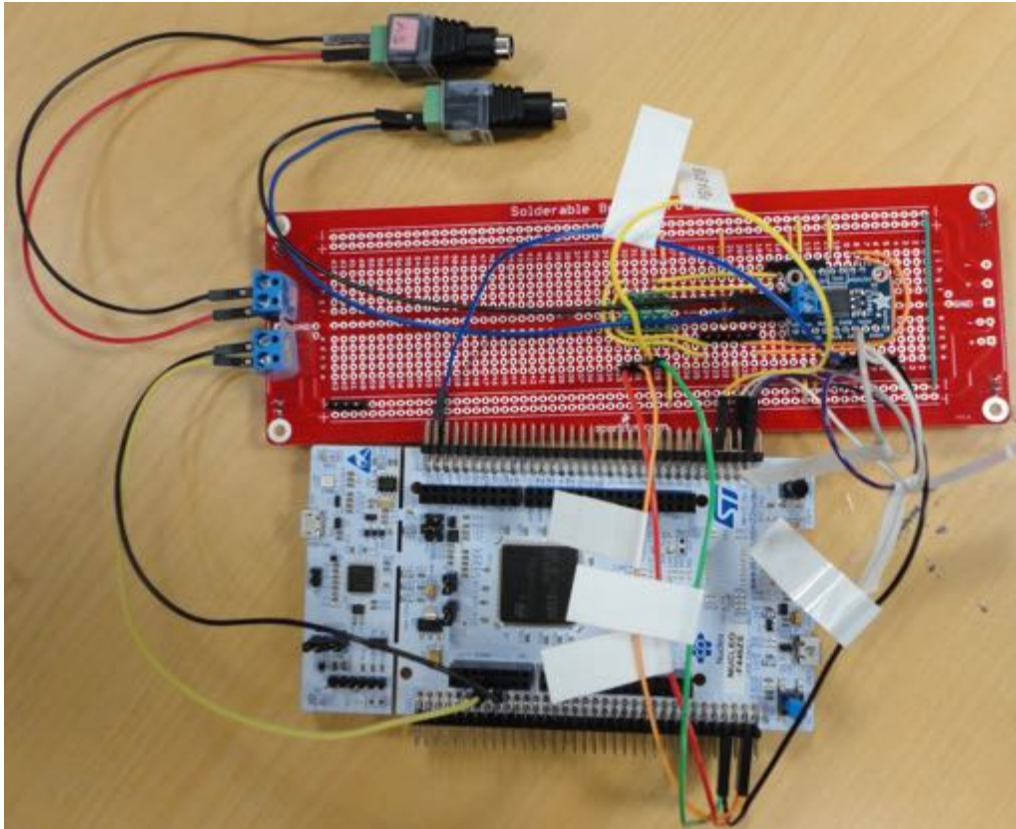
3. **Assemble your electronics board.** There is really no right or wrong way to do this step. What you need is an organized way to connect your Nucleo and motor driver and level shifter together, as well as a way to power all your circuitry. We provide the template bellow which you are welcome to follow. However, if you are not a fan of soldering, using a solderless breadboard should also work. Remember to also solder headers to connect the pins from the nucleo as well as motors.



(a)

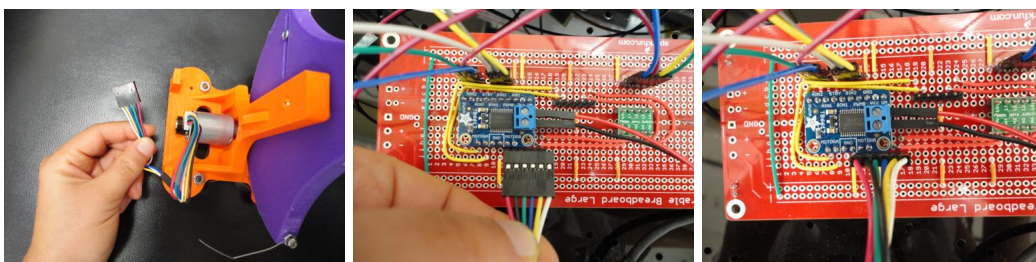
4. **Connect your electronics board to your Nucleo board.** (a) Since we used male headers both in our Nucleo as well as in our electronics board, we recommend that you use female to female jumper wires to connect the pins together. The following table outlines the connections:

Port Name	Nucleo Pin Name	Electronic Component Pin Name
S2B	PG12	L4
S2A	PG13	L3
S1B	PG15	L2
S1A	PG16	L1
PWM1	PC6	PWMB
PWM12	PG7	BIN2
PWM11	PG5	BIN1
PWM2	PC7	PWMA
PWM22	PG9	AIN2
PWM21	PG8	AIN1
GND	GND	GND
3.3V	3.3V	3.3V



(a)

5. **Connect Hapkit to your electronics Board:** Now that you have your electronics board assembled, you are ready to connect your Hapkit to it. (a) As explained in the motor section, the pololu motor has a 6 wire interface to our electronics. (b)-(c) The first wire (the red one) is one of the motor terminals. Make sure you align that with the header for Motor 1 on the board so that the red wire is connected all the way to the left (to Motor B1). Once you are aligned, plug in the header.

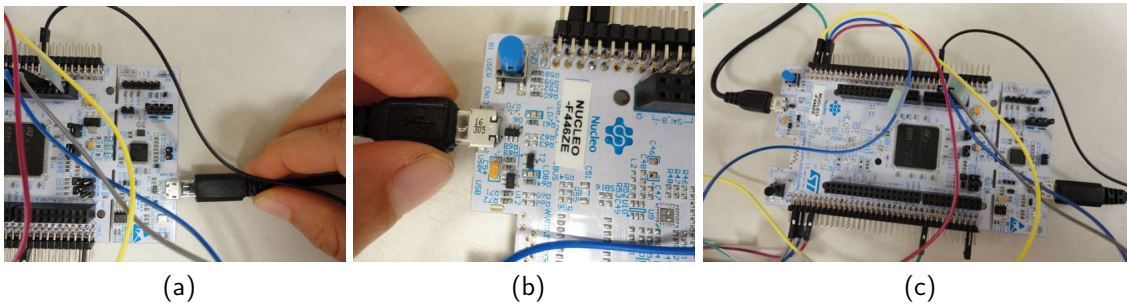


(a)

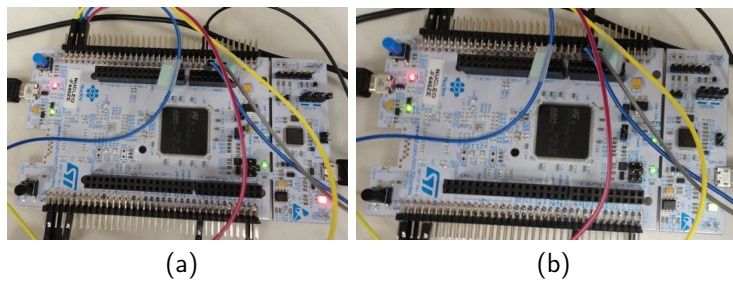
(b)

(c)

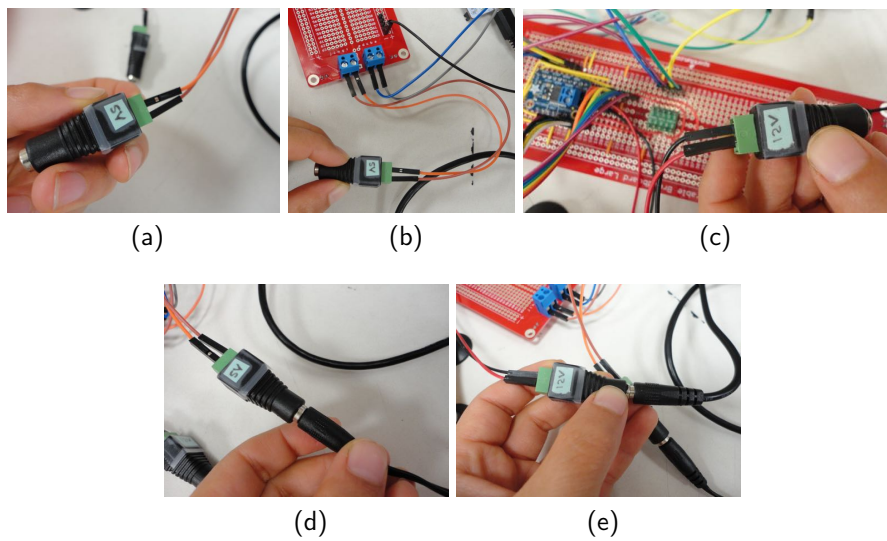
6. **Connect the USB cables to your Nucleo Board.** There are two USB cables in your kit. One 3ft and another 6 ft in length. (a) Plug in the shorter USB cable to the programming side of your board and then to a USB port on your computer. (b) Then plug in the longer USB cable to the USB side of your board and then to your computer.



7. **Download a sample Hapkit program following the Programming Instructions given in class.** (a) When you first plug in your board to the computer, LED4 will be RED. (b) Once there is a program loaded into it, it will turn green.



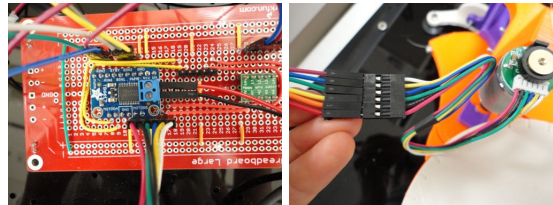
8. **Plug in the 12V and 5V power supply to your electronics board** It is important that you do this step after you have downloaded some code to your board. Otherwise you may be driving pins that should be inputs and you can damage your electronics. (a)-(b) The 5V power socket is connected to your board through a header and it distributes 5V to the entire bottom power rail. (c) The 12V power socket only goes to the motor driver. (d) Connect the 5V power supply to the 5V socket. (e) Connect the 12V power supply to the 12V power socket.



You are now ready to download different virtual environments for your Hapkit. Congratulations!

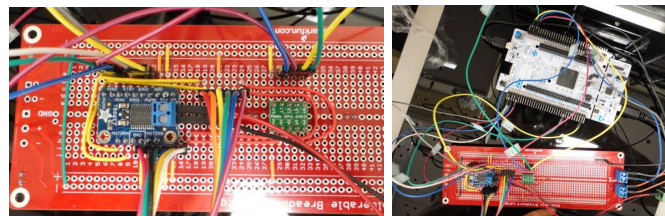
9. **Connect Haplink to your electronics board.** Once you have Haplink assembled following the Assembly document, (a) you should connect Motor 1 to Header one like you did in

Step 5. Following this, you will notice that motor 2 is too short to be connected straight into the electronics board. (b) Use some Male to Female jumper wires to connect to the motor terminals. We recommend that you either follow all the color patterns or at least make sure that the red motor wire is connected to a red wire. (c) That way you can still line up the red wire all the way to the left as you did on Step 5 (but now for header 2) and you will have your Motor 2 interfacing to the right pins on the board (d).



(a)

(b)



(c)

(d)

Congrats! you have completed the electronics assembly of Haplink! You should download different virtual environments to your device and see how it feels.