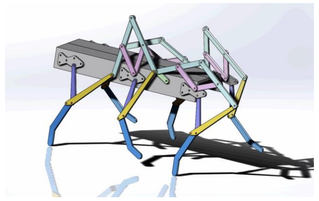
Electronics for Walker Control

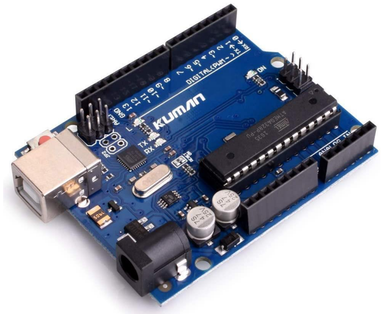
Kinematic Synthesis of Mechanisms

Jiaji Li Spring 2023

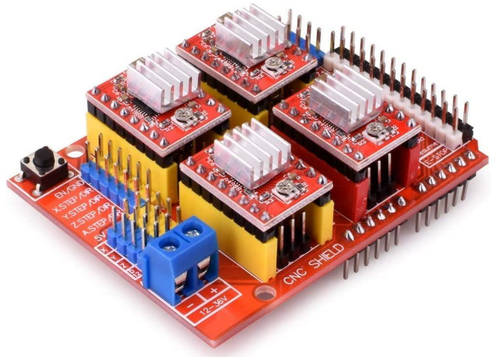


Electronic Components

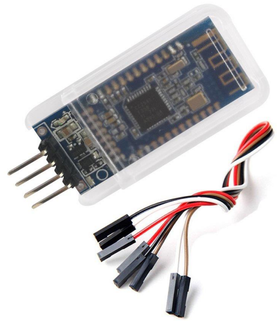
Arduino Uno



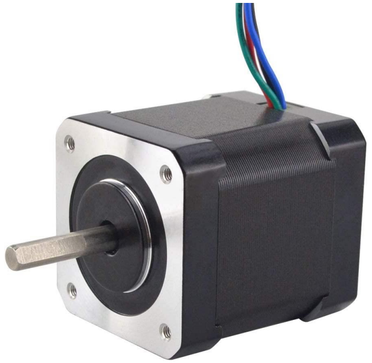
CNC Shield Board



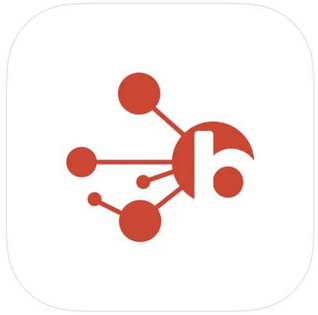
HM-10 Bluetooth Module



Stepper Motor

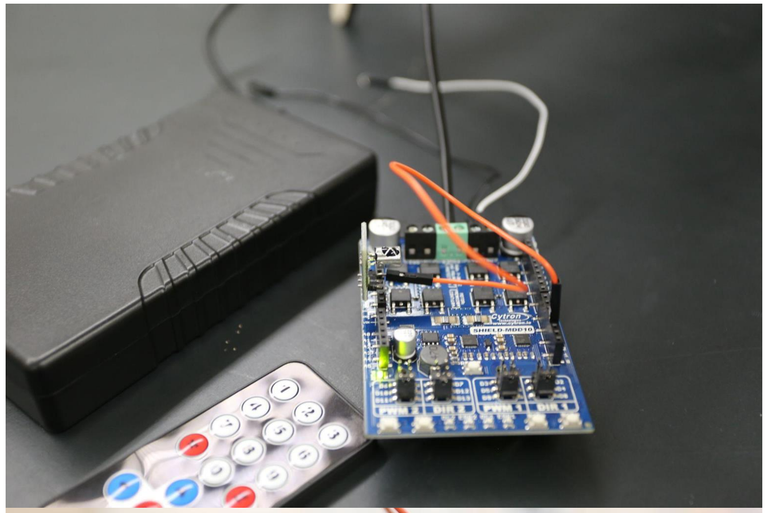


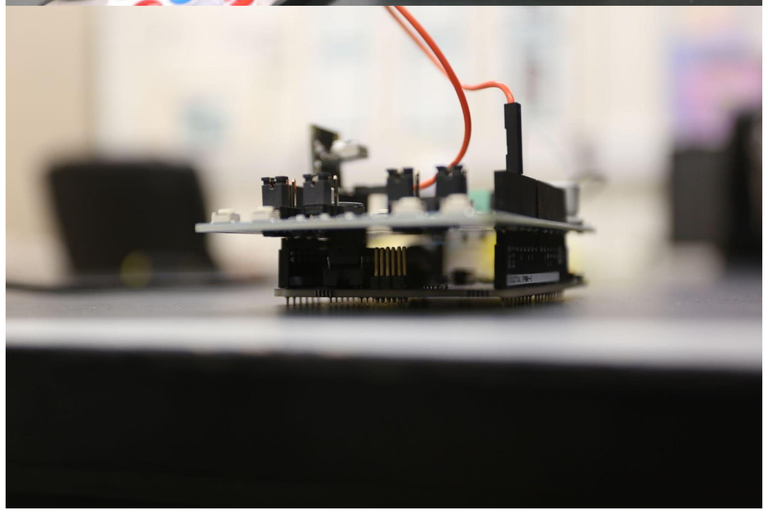
BitBlue app for iOS

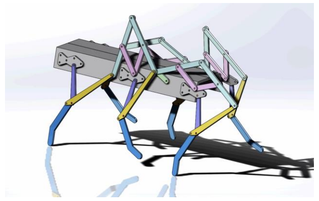


BitBlue 4+
Sung_Han Lee
Designed for iPad
★★★★★ 5.0 • 2 Ratings
Free
View in Mac App Store ↗

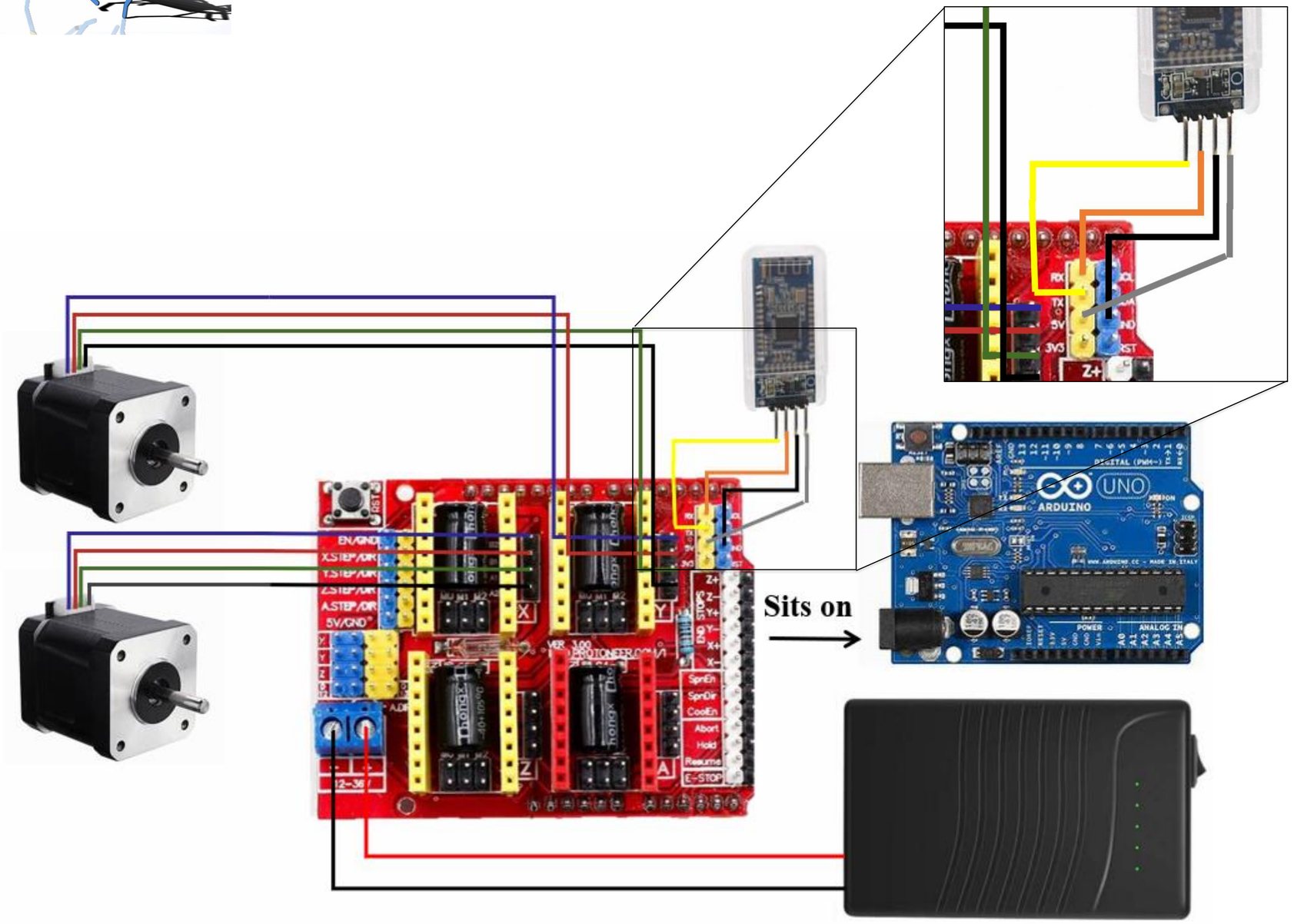
12V Battery



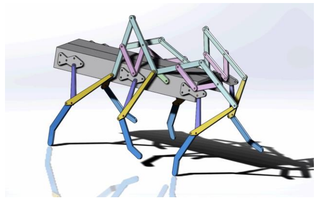




Electronics Schematic



1. Stepper Motor Shield sits on top of the Arduino.
2. 12V Battery is connected to the motor shield.
3. Connect the motors to the motor shield as shown.
4. Connect 5V (grey wire) and ground (black wire) from the motor shield to the Bluetooth module.
5. Before connecting the Rx and Tx wires, upload the code to the Arduino. Every code is uploaded to the Arduino, disconnect the Rx and Tx wires.
6. Connect the Rx of the Blue tooth module (yellow wire) to Tx of the motor shield, and the Tx of the Bluetooth module (orange wire) to the Rx of the motor shield.



Stepper Motor Shield

Step 3

Set-Up

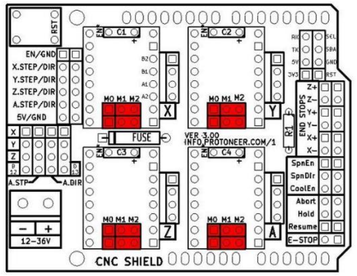
You can access the stepper motor shield documentation at the link

CNC Stepper Motor Shield

Step 1

1. Configuring Micro Stepping for Each Axis

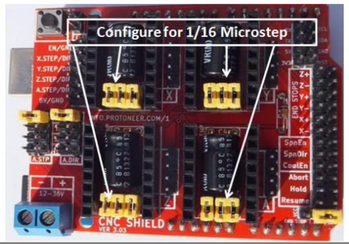
Each axis has 3 jumpers that can be set to configure the micro stepping for the A4988 plug-in driver board.



Micro-stepping jumper location, before inserting A4988.

In the tables below 'High' indicates that a jumper is insert and 'Low' indicates that no jumper is inserted.

MS0	MS1	MS2	Microstep Resolution
Low	Low	Low	Full Step
High	Low	Low	½ Step
Low	High	Low	¼ Step
High	High	Low	1/8 Step
High	High	High	1/16 Step



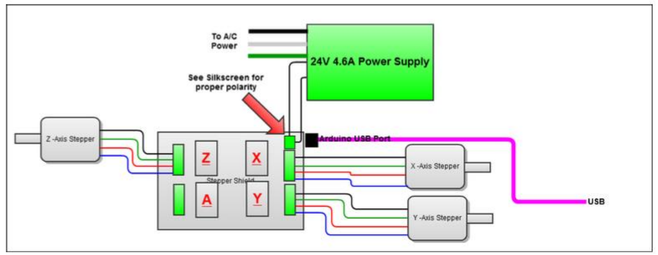
Step 2

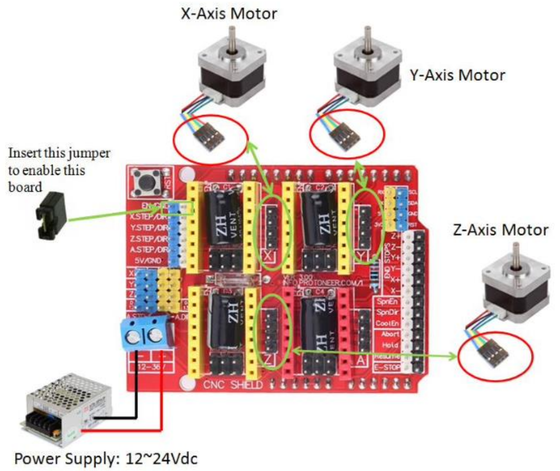
After setting the microstep jumper, you can plug-in A4988 driver boards as shown in the photo also shown this CNC sit nicely on top of Arduino Uno board, without any external

!!! Beware of the orientation of the A4988 driver boards! You will des driver board if plug-in with wrong orientation.

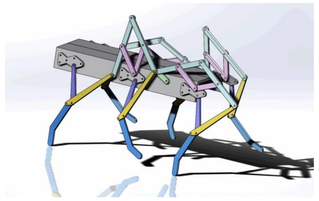
3. Hooking Up the Stepper Motor to CNC Shield

Connect steppers motor to CNC Shield board as the below block diagram. of the CNC Shield connected to 3-stepper motor:





Your CNC Shield board is now ready to go for a test run, let's try to turn the motor as to our instruction !!



Software for Motor Shield

Download to the Arduino

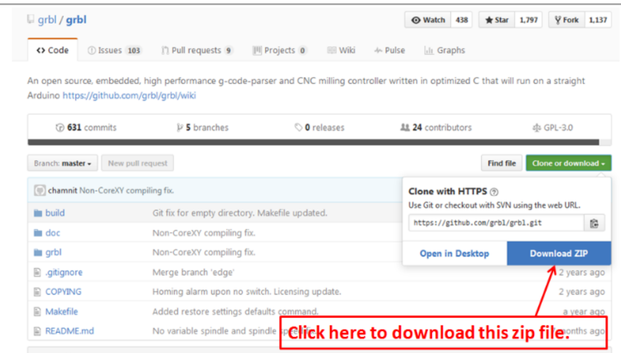
2. GRBL Control Software/Firmware for Arduino

Before you can use this CNC shield with Arduino, a control firmware need to be downloaded into Arduino board. We are going to use 'GRBL' to accomplish our job. GRBL is open-source software that runs on an Arduino Uno that takes G-Code commands via Serial and turns the commands into motor signals. Grbl is a no-compromise, high performance, low cost alternative to parallel-port-based motion control for CNC machine. It accepts standards-compliant g-code and has been tested with the output of several CAM tools with no problems. Arcs, circles and helical motion are fully supported, as well as, all other primary g-code commands. Macro functions, variables, and most canned cycles are not supported, but we think GUIs can do a much better job at translating them into straight g-code anyhow.

A copy of this open-source firmware can be downloaded from the below link:

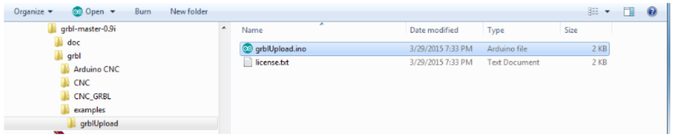
Following the below steps to prepare this CNC Shield board to function properly:

1. Download a copy of GRBL from: https://github.com/grbl/grbl

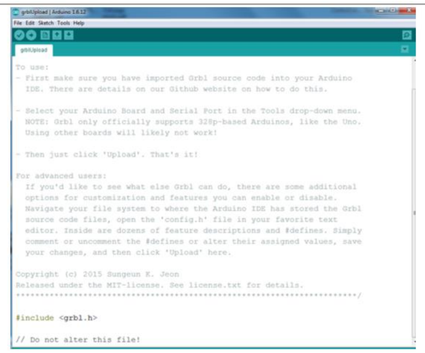


Unzip this file into your local hardisk location, you may want to create a special new folder for this purpose for easy locating of all files needed in your project. Locate an Arduino sketch '***grblUpload.ino***' in this folder where you have unzip the files.

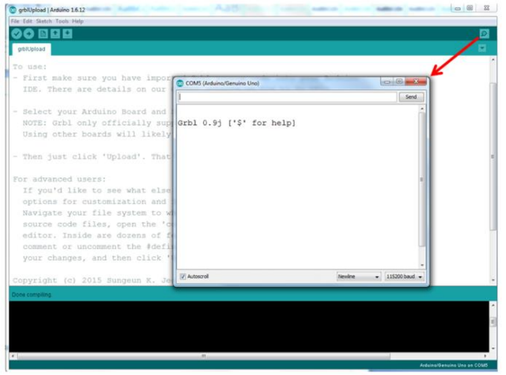
Below is the files structures located in my local hardisk:



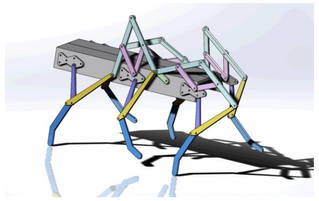
Open up this sketch '***grblUpload.ino***' and you should see the screen as below:



Click the upload icon as usual to 'compile/upload' as you normally upload Arduino sketch. When you see 'done uploading', click the 'Serial Monitor' on Arduino IDE as shown below:



If you can receive response message " ***Grbl 0.9j ['$' for help]*** " from your Serial Monitor, congratulation! You have successful uploaded the 'GRBL' firmware into your Arduino board.



Software

1. Setup code
2. Loop code
3. Control Functions

Gonk Code folder: [MAE_183_Code]

1

```
#include <math.h>

#define PUL_PIN 2   // Pulse Pin (sometimes called a step pin)
#define DIR_PIN 5   // Direction Pin
#define PUL_PIN2 3  // Pulse Pin (sometimes called a step pin)
#define DIR_PIN2 6  // Direction
#define enable 8    //enable pin

// Steps per revolution of my motor/driver combo
//SoftwareSerial BTserial(0,1);
int Motor1Interval;  //Defined step interval for each stepper motor
int Motor2Interval;
int FullRightRot;  // Rotation count rounded down by defining it as an integer
int FullLeftRot;
int k = 1;
int s = 2000;  //Time interval between steps in microseconds

float LeftStepCount = 0;  //Step count variable keeping track of how many steps have been taken for each motor
float RightStepCount = 0;
float leftdist;  // Amount of distance traveled by each motor calculated
float rightdist;

unsigned long previousMotor1Time = micros();  //Time variable of when the motor stepped last
unsigned long previousMotor2Time = micros();


char BTvar = '0';  // character variable

bool B_executed = false;

//SETUP

void setup() {
  //Start serial to read both leftdist and right dist, and for the HM-10 BT module
  pinMode(PUL_PIN, OUTPUT);  //LeftMotor Output pins
  pinMode(DIR_PIN, OUTPUT);
  pinMode(PUL_PIN2, OUTPUT);  //Right Motor Pins
  pinMode(DIR_PIN2, OUTPUT);
  pinMode(enable, OUTPUT);
  digitalWrite(DIR_PIN, LOW);  //Direction outputs required for forward movement
  digitalWrite(DIR_PIN2, HIGH);
  digitalWrite(enable, HIGH);
  Serial.begin(115200);



}
```

2

```
if (BTvar == 'B' && !B_executed) {  //speed up
  s = constrain(s - 100, 300, 2000);
  B_executed = true;
  BTvar = 'F';
}
if (BTvar != 'B') {
  B_executed = false;
}
if (BTvar == 'X' && !X_executed) {  //slow down
  s = constrain(s + 100, 300, 2000);
  X_executed = true;
  BTvar = 'F';
}
if (BTvar != 'X') {
  X_executed = false;
}
if (BTvar == 'S') {
  digitalWrite(enable, LOW);
  Serial.println("System Unlocked");
  /*
  Reset();
  BTvar = '0';*/
}
if (BTvar == 'C') {
  digitalWrite(enable, HIGH);
  Serial.println("System Locked");
}
if (BTvar == 'F') {
  Forward();
}
if (BTvar == 'L') {
  Left();
}
if (BTvar == 'R') {
  Right();
}
if (BTvar == 'T') {
  Reverse();
}
if (BTvar == 'Y') {
  TurnAroundR();
}
  if (BTvar == 'A') {
  TurnAroundL();
  }
}
```

3

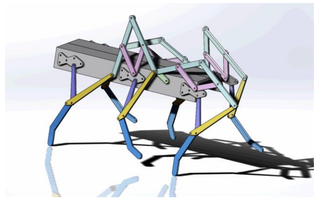
```
void Forward() {
  digitalWrite(DIR_PIN, LOW);
  digitalWrite(DIR_PIN2, HIGH);
  k = 1;
  Motor1Interval = s;
  Motor2Interval = s;
}

void Left() {
  digitalWrite(DIR_PIN, LOW);
  digitalWrite(DIR_PIN2, HIGH);
  k = 1;
  Motor1Interval = 2 * s;
  Motor2Interval = s;
}
void Right() {
  digitalWrite(DIR_PIN, LOW);
  digitalWrite(DIR_PIN2, HIGH);
  k = 1;
  Motor1Interval = s;
  Motor2Interval = 2 * s;
}

void Reverse() {
  digitalWrite(DIR_PIN, HIGH);
  digitalWrite(DIR_PIN2, LOW);
  k = 1;
  Motor1Interval = s;
  Motor2Interval = s;
}

void TurnAroundR() {
  digitalWrite(DIR_PIN, LOW);
  digitalWrite(DIR_PIN2, LOW);
  k = -1;
  Motor1Interval = s;
  Motor2Interval = 2 * s;
}

void TurnAroundL() {
  digitalWrite(DIR_PIN, HIGH);
  digitalWrite(DIR_PIN2, HIGH);
  k = -1;
  Motor1Interval = s;
  Motor2Interval = 2 * s;
}
```



Serial Interface to Arduino

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3); // RX, TX

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port
only
  }

  // set the data rate for the SoftwareSerial port
  mySerial.begin(9600);

  // Ensure the HM-10 is in AT mode
  mySerial.print("AT");
  delay(100);
}

void loop() { // run over and over
  if (mySerial.available()) {
    Serial.write(mySerial.read());
  }
  if (Serial.available()) {
    mySerial.write(Serial.read());
  }
}
```

1.Test the AT mode:

- Open the serial monitor in the Arduino IDE (Tools > Serial Monitor)
- Set the baud rate to 9600 and the line ending to "Both NL & CR" or "Carriage return".
- Type "AT" (without quotes) and press Enter. If the HM-10 module is in AT mode, it should respond with "OK".

2.Configure the HM-10 module:

- Use the AT commands listed in the HM-10 datasheet to configure the module according to your needs. For example, you can change the name, baud rate, and other settings of the module.

Remember that not all HM-10 modules have the same default settings, so consult the documentation for the specific HM-10 module you are using for the appropriate AT commands and default values.
Link to the data sheet

1.Verify the baud rate: It is important to check the baud rate set on the HM-10 module you have received. While it is not necessary to change the baud rate of the module, ensure that the Arduino's baud rate matches this value, as we will use the Arduino's TX and RX pins to communicate with the Bluetooth module.
2.Set a custom name for the Bluetooth module: Assigning a unique name to the Bluetooth module is recommended to avoid confusion when connecting to other devices. This will ensure that you do not accidentally connect to another device with a similar name.

50. Query/Set Module name

Send	Receive	Parameter
AT+NAME?	OK+NAME[P1]	P1: module name, Max length is 12. Default: HMSoft
AT+NAME[P1]	OK+Set[P1]	

e.g.

change module name to bill_gates

Send: AT+NAMEbill_gates

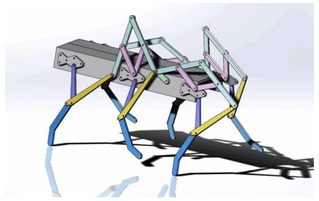
Receive: OK+SetName:bill_gates

1.Ensure proper alignment of the pins when connecting the Arduino to the shield.

2.If the power system is 12V or below, connect the power supply directly to the Arduino Uno.

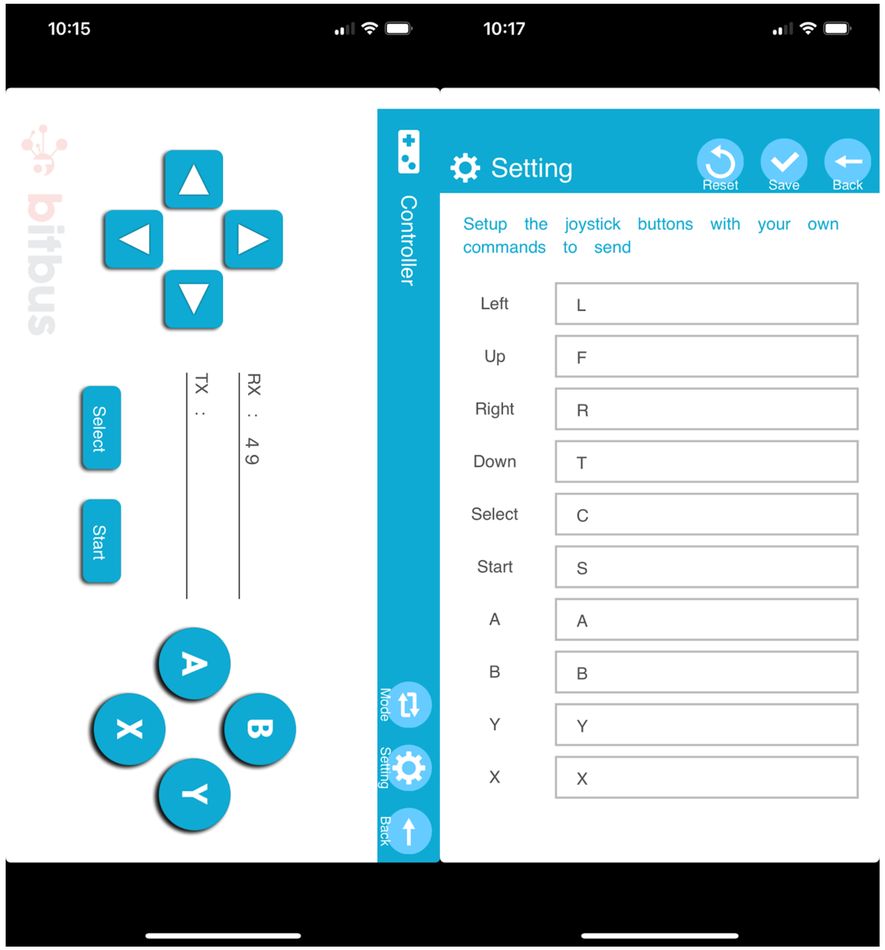
3.Avoid unplugging the stepper motor or driver while the shield is powered to prevent potential damage.

4.Adjust the current potentiometer according to the recommended motor current specified in the stepper motor's datasheet.



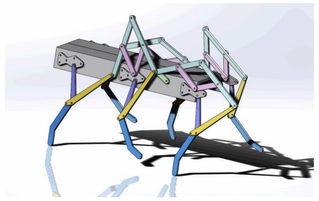
How to Operate the Remote

[Link to Arduino Code]
Bluetooth Controller
App: [BitBlue] on the Apple store

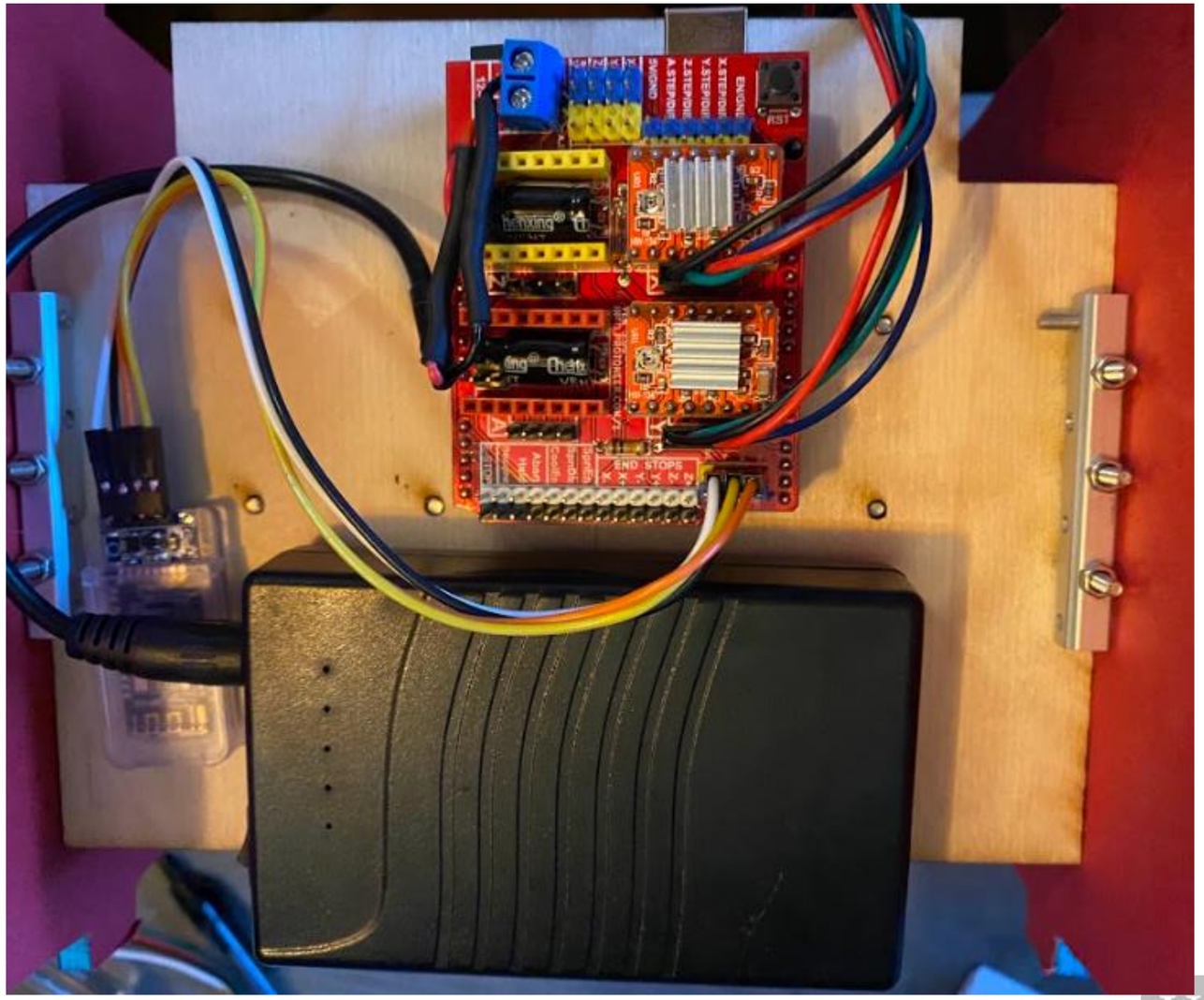


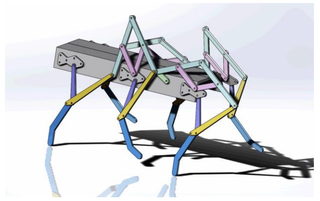
The controls for the walker are as follows:

- Start: Start
- B: Increase Speed
- A: Rotate CCW
- Y: Rotate CW
- X: Decrease Speed
- ▲: Forward Movement
- ◀: Turn Left
- ▶: Turn Right
- ▼: Reverse
- Select: Stop



Demonstration





Electronics Parts List

Electronics Parts List				
Description	Amount	Unit	Cost/Unit	Cost
STEPPERONLINE Nema 17 Stepper Motor	1	Each	$13.99	$13.99
weideer 2Pcs 42mm Stepper Motor Mounting Bracket	2	Each	$9.99	$19.98
kuman CNC Shield Expansion Board V3.0 +UNO R3 Board + A4988 Stepper Motor Driver	1	Each	$25.99	$25.99
TalentCell Rechargeable 12V DC Output Lithium ion Battery Pack	1	Each	$27.99	$27.99
DSD TECH HM-10 Master and Slave Bluetooth 4.0 LE iBeacon Module	1	Each	$11.49	$11.49
Total				99.44

Data sheets

StepperOnline NEMA 17 59 Ncm
A 4988 Stepper Motor Driver
CNC Stepper Motor Shield
HM-10 Bluetooth Module