BUILD the Bipedal Walking ROBOT

A DIY Guide to 3D Printing, Assembly, and Coding with Arduino and Raspberry Pi

Nix Robot

Build the Bipedal Walking Robot

A DIY Guide to
3D Printing, Assembly, and Coding
with Arduino and Raspberry Pi

Nix Robot

First Edition
ISBN: 979-8-27822-202-6

About This Book

All assembly, electrical, 3D printing, programming, and other sections of this book are based on actual working models of the robot, originally developed at Columbia University in the City of New York as an academic project.

To achieve successful results, please read the chapters and their content sequentially in the order that they appear in the book. The content is arranged so that the robot is built step by step, and performing previous steps is necessary for the subsequent ones.

For tutorials and new books on robotics, circuits, and electronics, visit:

www.youtube.com/@nixrobot

Table of Contents

Chapter 1. The Machine We Build

Do you wish to have a two-legged robot that can walk, kick, speak, and do some other moves and actions? Can it be upgraded to have arms, vision, and speech recognition? If you are reading this book, it is assumed that you answered yes to some or all of these questions.

Before we begin, let us delve into the world of robotics and engineering and discuss some developments in the field.

Bipedal robots are robots that have two legs. They are considered especially hard to engineer and program. Yet using this book, anyone who can assemble building brick toys can make a robot, upload software, and run it. It is all thanks to detailed instructions and drawings.

No special engineering knowledge is needed to build your own first robot using this guide. If you want to upgrade it further – add various sensors, cameras, or arms – you could learn some aspects of engineering from this book and supplemental materials – just what you need, step by step.

So, this book is a first step from having just a wish to build a robot to becoming a roboticist with no limits and unlimited horizons.

Robots might soon be present everywhere. That means, for example, that in homes, factories, hospitals, and construction sites, robots could do useful work. Repetitive tasks are especially easy to robotize. That is why learning robotics is important. Maybe in the future, people will spend more time managing robots, maintaining them, and, of course, engineering them, rather than simply doing repetitive work.

Robotics = creativity. Making a robot, and in particular this robot according to this book, may sound like doing something very straightforward and mechanistic. But it is not. You will see that in the process, and even during

preparation for your project, you can make engineering choices, and your final result may be slightly different. For instance, you may omit some components, like speakers. Alternatively, these speakers could have different shapes. But the legs will be the same.

Figure 1.1 shows the robot you will build. As one student pointed out in a comment on the picture shown to the class, “He looks strong! Definitely did not skip leg days.” Indeed, he can do exercise moves.

Figure 1.1: Cool robot photo.

Let us now look at a general schematic view of the robot, which can be found in Figure 1.2. We will review many parts of the machine in detail, so this is only to observe some key structural blocks.

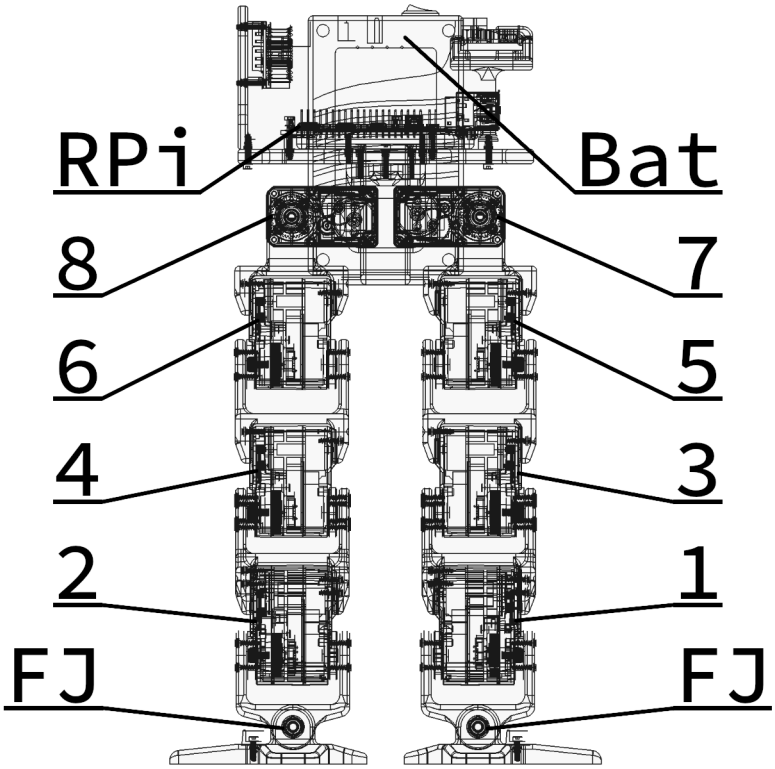

Figure 1.2: General robot drawing.

Let us explain what the numbers and letters in Figure 1.2 mean.

Numbers from **1** to **8** represent the motors in the legs that control the movements of the robot. As you can notice, they are numbered from bottom to top, and the right leg has even numbers, while the left leg has odd numbers. This is important for our robot, and it will be explained later in the book.

FJ means free joint. A joint is a connection of two parts. This joint does not have a motor to control it, and by default, there is nothing that prevents it from freely rotating. In this robot, the feet can rotate freely and there is no motor to control them.

Bat means battery; it is positioned on the back of the robot and is easily detachable to allow transportation without the battery.

RPi means Raspberry Pi – a single-board computer that runs programs to control the robot. It could be replaced with an Arduino at later stages of your project.

Sure enough, this is not the complete list of components of the robot, but the list of the most important ones.

All right. So that is how it will look when fully assembled and ready to walk.

But what about tech specs? How strong is this robot, and what are its key parameters? Well, here is the list in the format of the "parameter name: parameter value" scheme:

- **Motors:** LX-16A
- **Main board:** Raspberry Pi 4/5 or Arduino
- **Height:** 12 inches (30 cm)
- **Width:** 7 inch (18 cm)
- **Mass:** 3.3 pounds (1.5 kg)
- **Maximum speed:** 4 inch/sec (10 cm/s)
- **Power consumption:** 64 W

These values above are rough estimates. Variations depend on the specific plastic, screws, wires, and electronics used.

Let us discuss some key definitions and concepts that are needed to better understand robotics and this book. If you are familiar with certain concepts, just skip them. No need to memorize them: you can always refer to this list later. Some concepts will be easier to understand during certain building stages.

Robotics – an interdisciplinary engineering field that combines skills and knowledge from mechanical, electrical, and software engineering.

Engineering – a field of human activity that assumes that some knowledge is applied to solve real-world problems and particular business needs. Designing and assembling a certain mechanism is an example of engineering.

Actuators – mechanical devices that perform some action. A motor is an example of an actuator.

CAD – computer-aided design. It is the process of creating models of mostly mechanical parts. For example, there are applications like FreeCAD, Fusion, and SolidWorks. They provide a user interface to design 3D models.

Chapter 2. Parts and Tools

You may already want to start building. Assembling itself could be done as quickly as in one week or less. A lot more time will be spent on looking for components and making parts. The first step is preparation: obtaining all necessary components.

Reading this chapter, you might notice that some components are optional; some might not fit into your design without additional workarounds; some components allow alternatives. Thus, here the elements of creativity in engineering work start to manifest themselves.

To build the robot, we need the components, materials, and tools that are listed in this chapter. The following categories of components are required: (a) electrical parts and electronics; (b) mechanical parts that will be 3D printed; (c) fasteners and tools. Each category is listed below.

In addition to the components listed in this chapter, to set up and control the Raspberry Pi, a Wi-Fi or Ethernet network is needed. To write programs and upload them into the robot, a computer is required.

In engineering, this kind of list, which we are creating here, is called the bill of materials (abbreviated BOM). So let us look at the BOM for the robot.

Tables follow the format: Name; Tech specs, part numbers, or other descriptive data; Quantity. Additional notes and pictures (where needed) are provided below the tables.

TABLE 2.1. ELECTRICAL PARTS

Name	**Tech Specs**	**Quantity**
LX-16A Serial Bus Servomotors [see notes below the table]	0-240 degrees rotation; 6-8 V; 16-19 kg·cm torque	8
Bus Linker for LX-16A. Also known as TTL / USB Debugging Board	Micro-USB port; serial port pins; servo cable ports	1
USB-A to Micro-USB Data Cable	4 inch (10 cm) length	1
USB-C to Micro-USB Data Cable or USB-C to USB-A socket [optional]	Any length. Needed if your laptop does not have USB-A ports	1
Barrel DC (*direct current*) Power Plug Connector, Male	Standard 2.1 × 5.5 mm barrel (coaxial) plug; male	1
DC Power Adapter [optional]	5.5-8 V, Barrel DC power plug	1
Barrel DC Power Plug Connector, Female [optional]	Standard 2.1 × 5.5 mm barrel (coaxial) plug; female	
Step-Down Voltage Regulator DC-DC [see notes below the table]	Used in this design: DROK-90481 10A DC-DC 4-30V to 1.2-30V 12V	1
Talentcell Battery YB1203000-USB	3000mAh, DC 12V/5V	1
Raspberry Pi 4 Model B or Raspberry Pi 5	Any RAM size	1
MicroSD Card	32 GB or more	1
USB-A to USB-C Power Cable	10 inch (25 cm) length	1
Black Wire	22 AWG (diameter: 0.0253 in / 0.644 mm), copper, solid/single core	3-6 feet (1-2 m)
Red Wire	22 AWG (diameter: 0.0253 in / 0.644 mm), copper, solid/single core	3-6 feet (1-2 m)
Arduino UNO or its clone [optional]	Standard Form Factor	1
USB-A to USB-B Data Cable [optional, for Arduino]	10 inch (25 cm) length	1
LEDs [optional]	5 mm, 3 V	2
Resistor [optional, for LEDs]	100-150 Ω for Raspberry Pi, 250-400 Ω for Arduino	2
Breadboard Jumper Cables [optional, for Arduino and LEDs]	Female to Male, 20 cm; Female to Female, 20 cm	4

Insulation [optional, for LED legs]	Either thin electrical tape or small heat shrink tubes (wrap sleeves).	For 4 legs
USB Mini Speaker [optional, but makes the robot very cool]	Examples: HONKYOB H002, Adafruit product ID: 3369	1
USB Mini Microphone [optional]	Examples: Adafruit product ID: 3369, Youmi, Estiq	1
Multimeter [optional, see note below]	Should be able to measure DC Voltage in the range 1-12 V	1
Soldering Iron and Soldering Materials	Alloy of tin (Sn), silver (Ag), and copper (Cu) is recommended as it is lead-free; and rosin to be used as the flux	1
Screwdriver	Flat-head (-), 2.5-3 mm (3/32-1/8 in)	1
Safety Glasses or Goggles	Any for eye protection	1
Wire Stripping Tool [optional]	10-22 AWG	1

Notes to Table 2.1:

These parts should be purchased online, or in local shops, or in makerspaces. Examples of online stores for electronics: Adafruit, Amazon, SparkFun.

If you get cables longer than needed, they can be cut and reconnected. If you get cables that are too short, they can be cut and extended.

Each **LX-16A servo** should include a 3-pin cable for serial bus communication and two plastic shaft disks (adapters). No metal brackets are needed.

Step-Down Voltage Regulator DC-DC could be omitted in some designs, but that will lead to very weak motor performance. It needs to have mounting holes positioned at the vertices of the rectangle: 54.8 × 22.8 mm. If you cannot find the same model or a device with the same positioning, there is a workaround that will be discussed in the assembling chapter. In that case, simply obtain the smallest voltage regulator you can find, preferably with an LCD screen, which is capable of converting 12 V to 7 V.

DC Power Adapter and **Barrel DC Power Plug Female** (socket barrel plug) are optional but can simplify the initial setup by replacing the battery and voltage regulator during testing.

A **multimeter** is needed if you are using a voltage regulator that does not have an LCD screen.

A **wire stripping tool** could be replaced with scissors, but scissors are very inconvenient.

Figures for Table 2.1:

Servomotors, as shown in Figure 2.1 below, are typically supplied with a 3-pin cable and two shaft adapters (plastic disks). They also usually include 4 mm screws, but we will not use these screws as they are too short for our parts.

After you have obtained these servomotors, do not place the shaft adapters on them. In the next chapters, we will first do some basic setup and calibration on them.

Figure 2.1: LX-16A servo.

The Bus Linker board is tiny and does not even have mounting holes (see Figure 2.2).

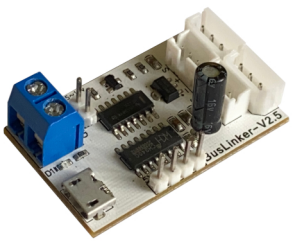

Figure 2.2: Bus Linker.

Chapter 5. Assembling

After accomplishing so many steps, the assembly should be a straightforward procedure. Now you need to unpack fasteners and put mechanical parts and electronics in your physical workspace. It might look as in Figure 5.1. We are ready to assemble.

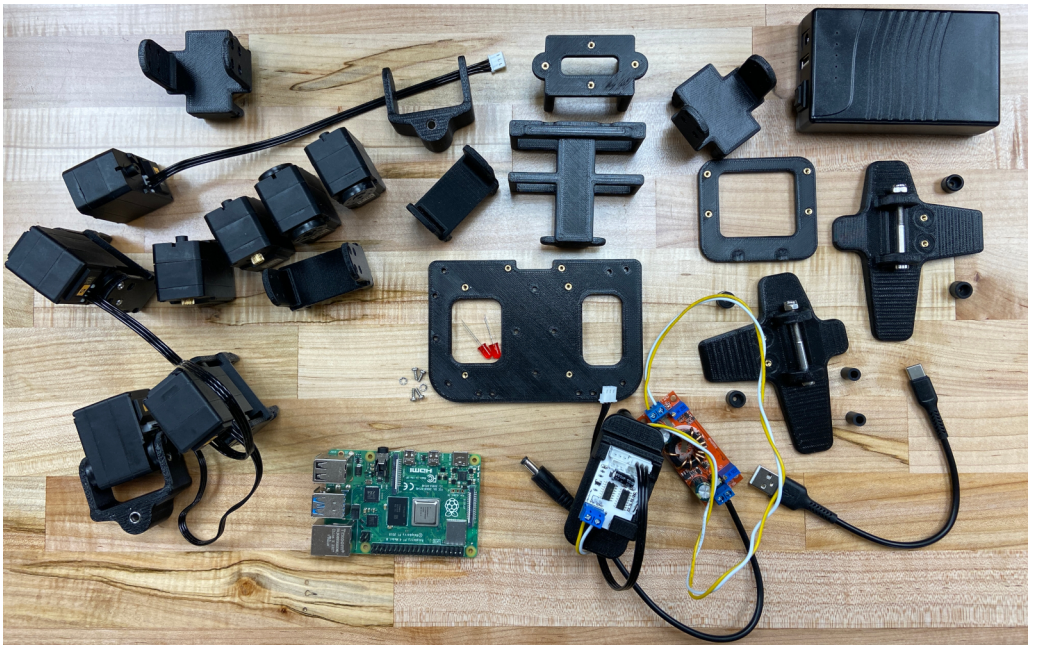

Figure 5.1: Ready to assemble.

We will build the robot from the feet to the top. Therefore, this chapter will begin by describing the assembly of one leg. The pictures show the left leg, so it is recommended to begin assembling with the left leg.

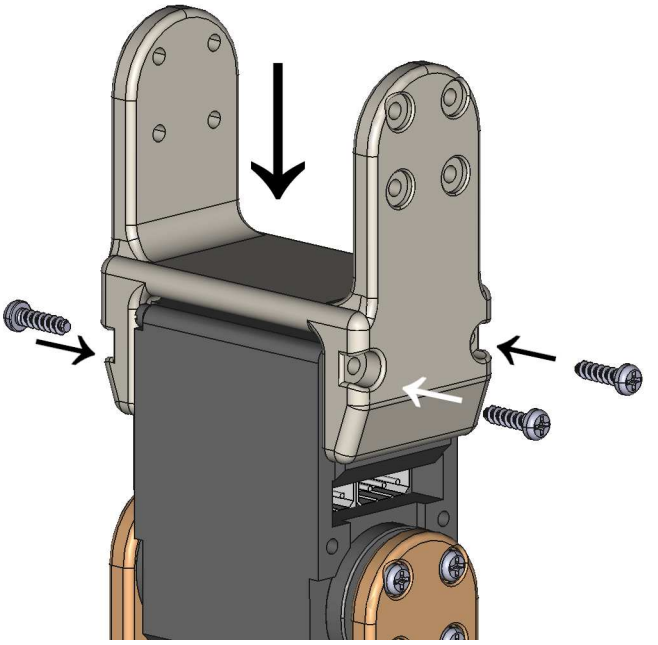

Figure 5.6: Leg segment assembly.

At this moment, you have assembled a leg that contains three servomotors, the foot, the ankle, and two leg segments. Time to install the hip. It is done in a similar way to the other leg components, as shown in Figure 5.7. Again, use four M2.2 8 mm Phillips thread-forming screws to secure the hip to the last added servomotor.

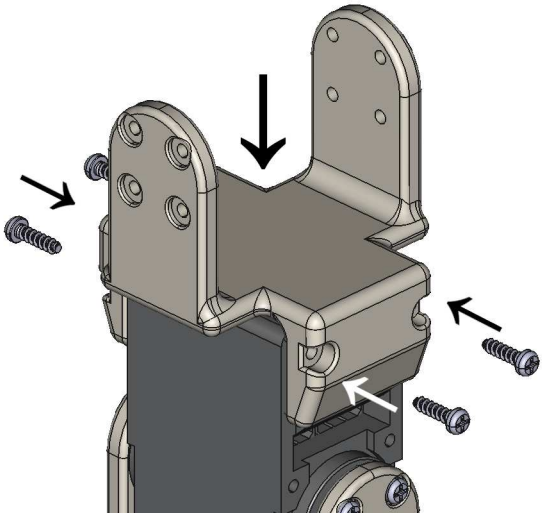

Figure 5.7: Hip installation.

Chapter 6. Programming with Raspberry Pi

Before using Raspberry Pi, its initial setup is required. If you already know how to set up Raspberry Pi, you can skip the first pages of this chapter and go directly to coding.

Download the official Raspberry Pi Imager from www.raspberrypi.com. It is an app that creates a bootable microSD with an operating system for Raspberry Pi.

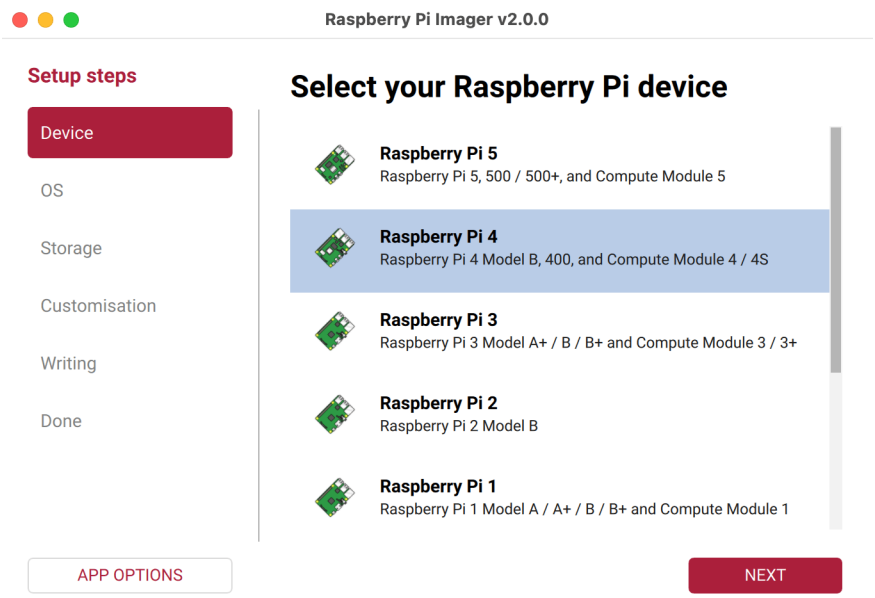

Figure 6.1: Selecting device.

You can combine portions of going forward with turning left or right to program complex paths, or combine them with other movements.

One interesting example is robot sliding, which is provided in Listing 6.8. This resembles the robot skating or something similar. In this way, the robot can move forward faster than in the regular *robot_go.py* script. It also looks fun. However, large slides are less precise, and sometimes it accidentally turns.

Listing 6.8 robot_slide.py

```
# Robot slides forward (free joints)
from robot import NixRobot


STEPS = 4
DURATION = 600

robot = NixRobot("/dev/ttyUSB0")

homing = (178, 20, 165, 110, 62, 220, 75, 130)
robot.homing_position(homing)

turn_angles = (190, 75, 98, 108, 75, 240, 36, 120)

turn_angles_reverse = tuple(
    [240-v for v in (turn_angles[4:] + turn_angles[:4])]
    )

i = 0
while True:
    robot.move_all_servos(turn_angles, DURATION)
    robot.homing_position(homing, DURATION*2)
    i += 1
    if i >= STEPS:
        break
    robot.move_all_servos(turn_angles_reverse, DURATION)
    robot.homing_position(homing, DURATION*2)
    i += 1
    if i >= STEPS:
        break

robot.disable_torque()
```

Chapter 7. Programming with Arduino

If you built the robot with the base plate v2, you can mount an Arduino board on it. To create programs with Arduino, two software pieces are needed: the Arduino IDE and the LX16A-bus servo library.

Get the Arduino IDE from the official Arduino website: www.arduino.cc

Install the IDE and open it. Select the Arduino Library Manager and enter the words "**LX16A-bus**" in the search box. You will see a picture similar to what is in Figure 7.1. Be careful: install only the *LX16A-bus* library; the other servo library does not support the Bus Linker and is not compatible with the Nix Robot.

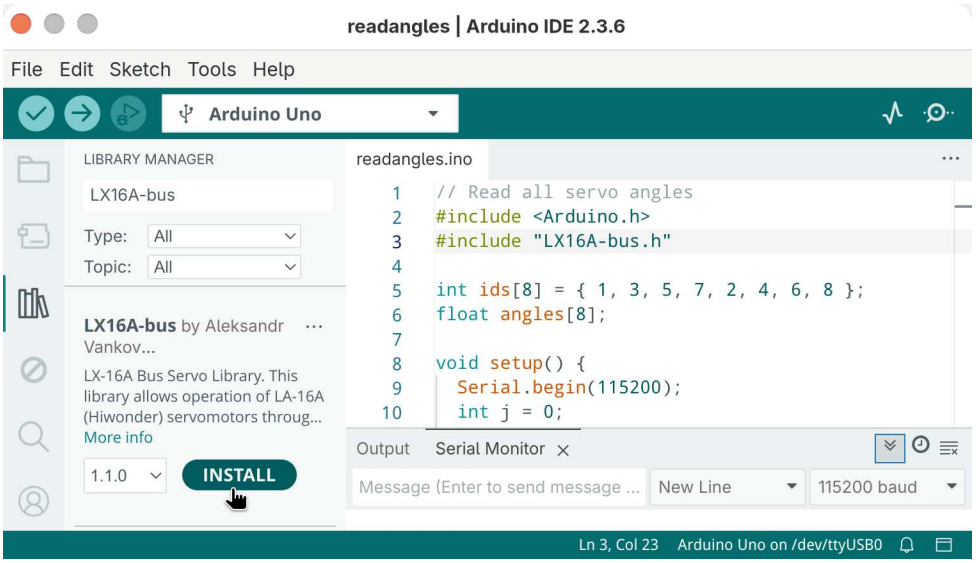

Figure 7.1: Arduino Library Manager: LX16A-bus library.

Listing 7.3: NixRobot.h

```
#ifndef NIX_ROBOT_H
#define NIX_ROBOT_H

#include <Arduino.h>
#include <LX16A-bus.h>

enum class Cmd {
    pause,
    homing,
    all,
    leftLeg,
    rightLeg,
    straight,
    tiltLeft,
    tiltRight,
};

struct Command {
  Cmd cmd;
  float* angles;
  uint16_t duration;
};

class NixRobot {
public:
  NixRobot(HardwareSerial& serial, unsigned long baud = 115200);
  void homing(uint16_t duration = 1500);
  void homing(float* angles, uint16_t duration = 1500);
  void moveAllServos(float* angles, uint16_t duration = 0);
  void moveLeftLeg(float* angles, uint16_t duration = 0);
  void moveRightLeg(float* angles, uint16_t duration = 0);
  void straight(uint16_t duration = 0);
  void tiltLeft(uint16_t duration = 0);
  void tiltRight(uint16_t duration = 0);
  void enableTorque();
  void disableTorque();
  bool isTorqueOn();
  bool isRunning();
  void runCommand(const Command& c);

  LX16A broadcaster;
  LX16A ankleLeft;
  LX16A kneeLeft;
  LX16A hipLowLeft;
  LX16A hipHighLeft;
```