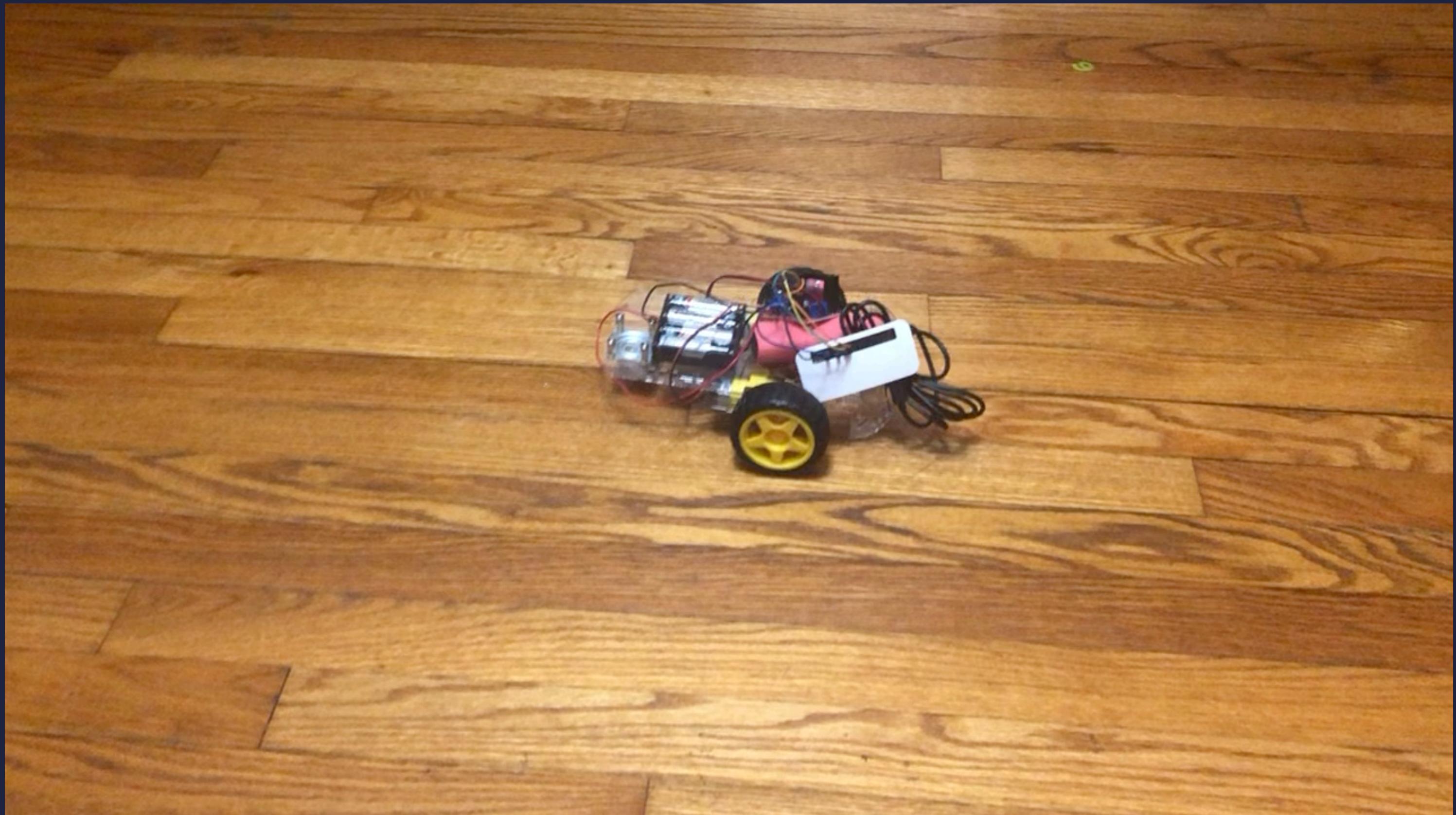


# SWIFT WITHOUT SCREENS - POWERING CONNECTED DEVICES

MARC AUPONT - IOS ENGINEER @LICKABILITY





Elon Musk

@elonmusk

Tell me more about the try!Bot model 3!  
@ [digimarktech](#)

4:35 PM · 10 Sep 19 · [Twitter For iOS](#)

---

**2,504 Retweets 8.9K Likes**

---



# WWDC 2014

# OBJECTIVE-C WITHOUT THE C





**SWIFT MADE DEVELOPMENT A  
POSSIBILITY FOR ME**



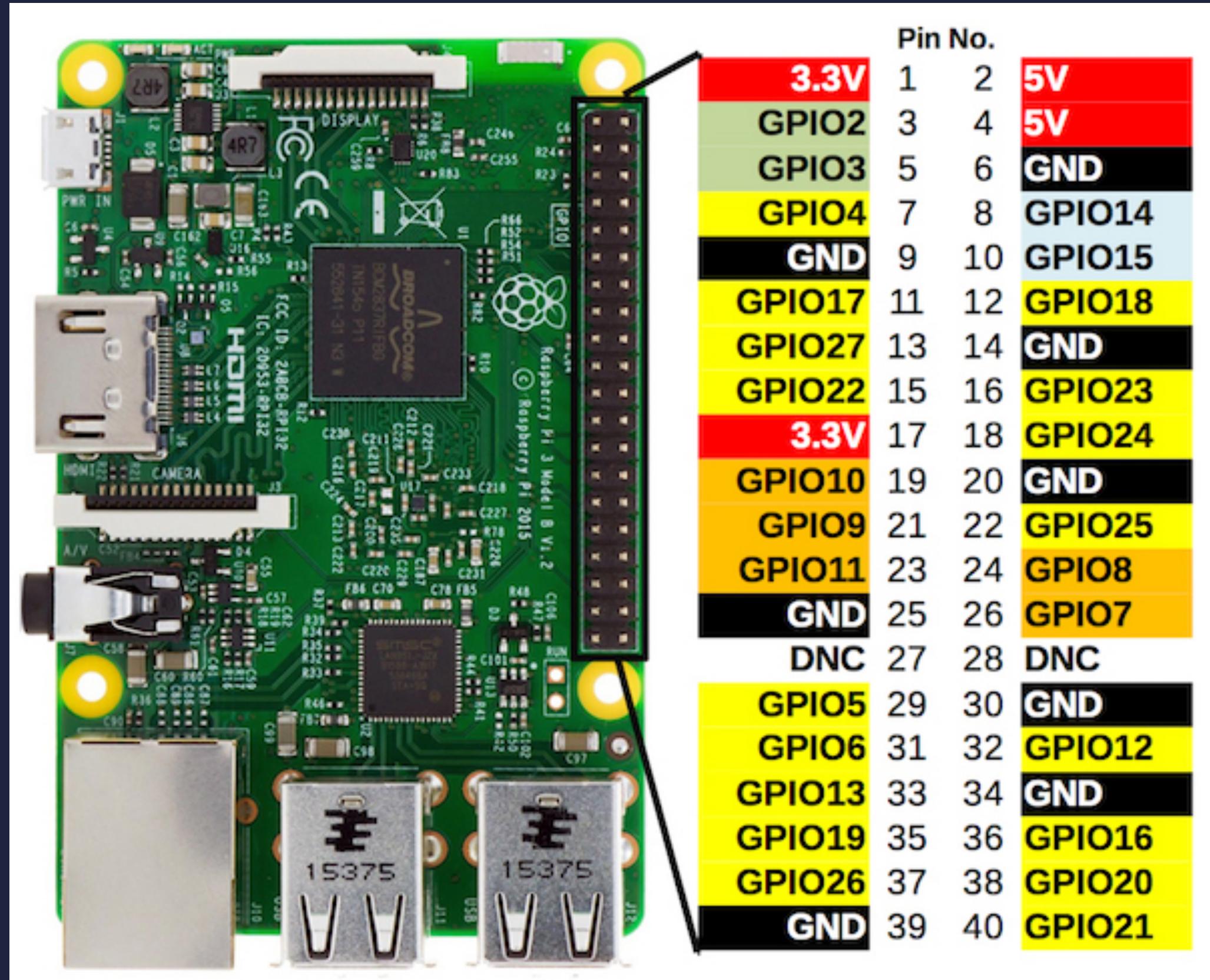
# SWIFT BECAME OPEN SOURCE

# SWIFT AVAILABLE ON LINUX

**BECAUSE SWIFT CAN RUN ON  
LINUX, SWIFT CAN RUN ON  
THE RASPBERRY PI**

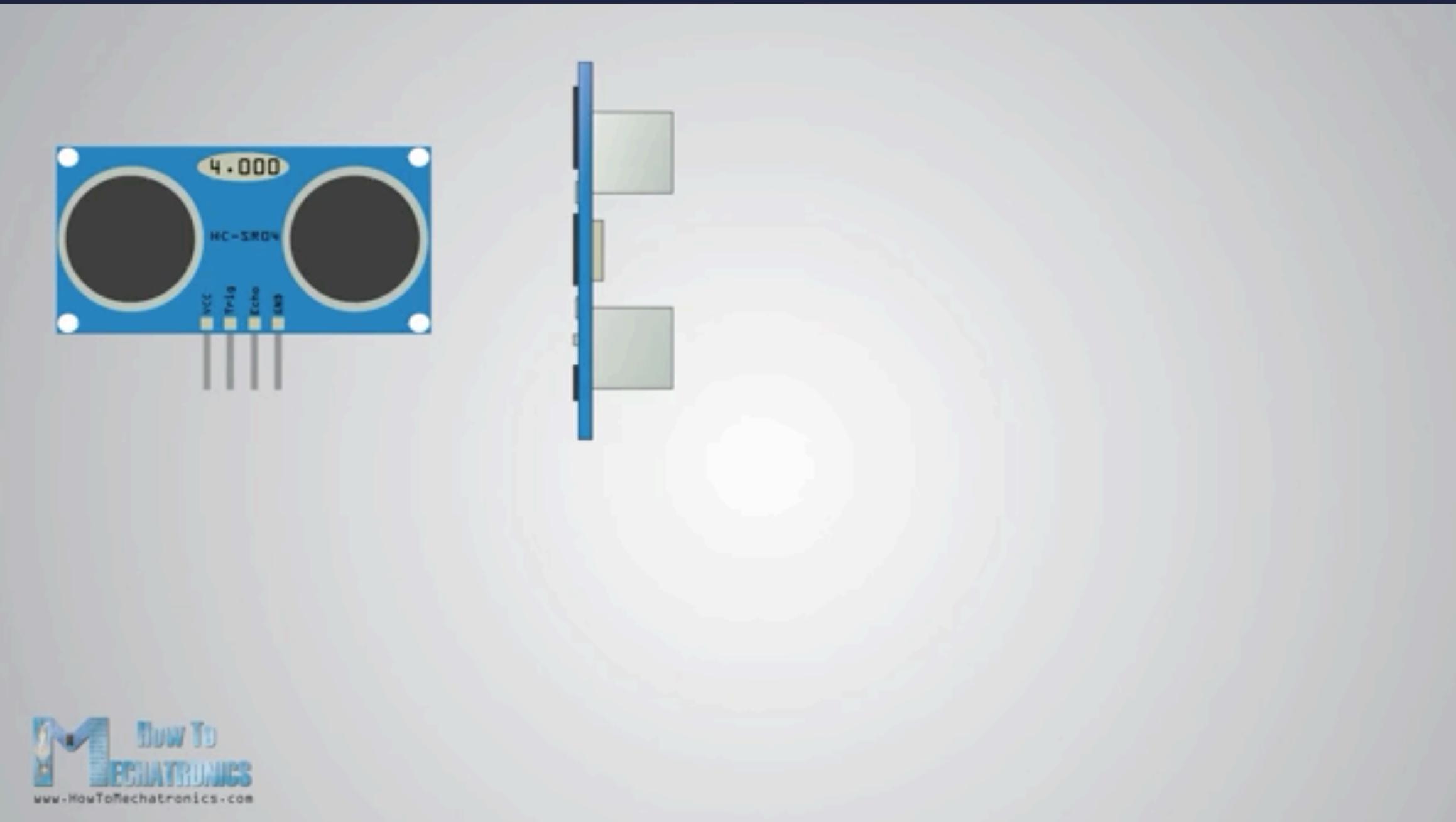
# WHAT IS RASPBERRY PI?



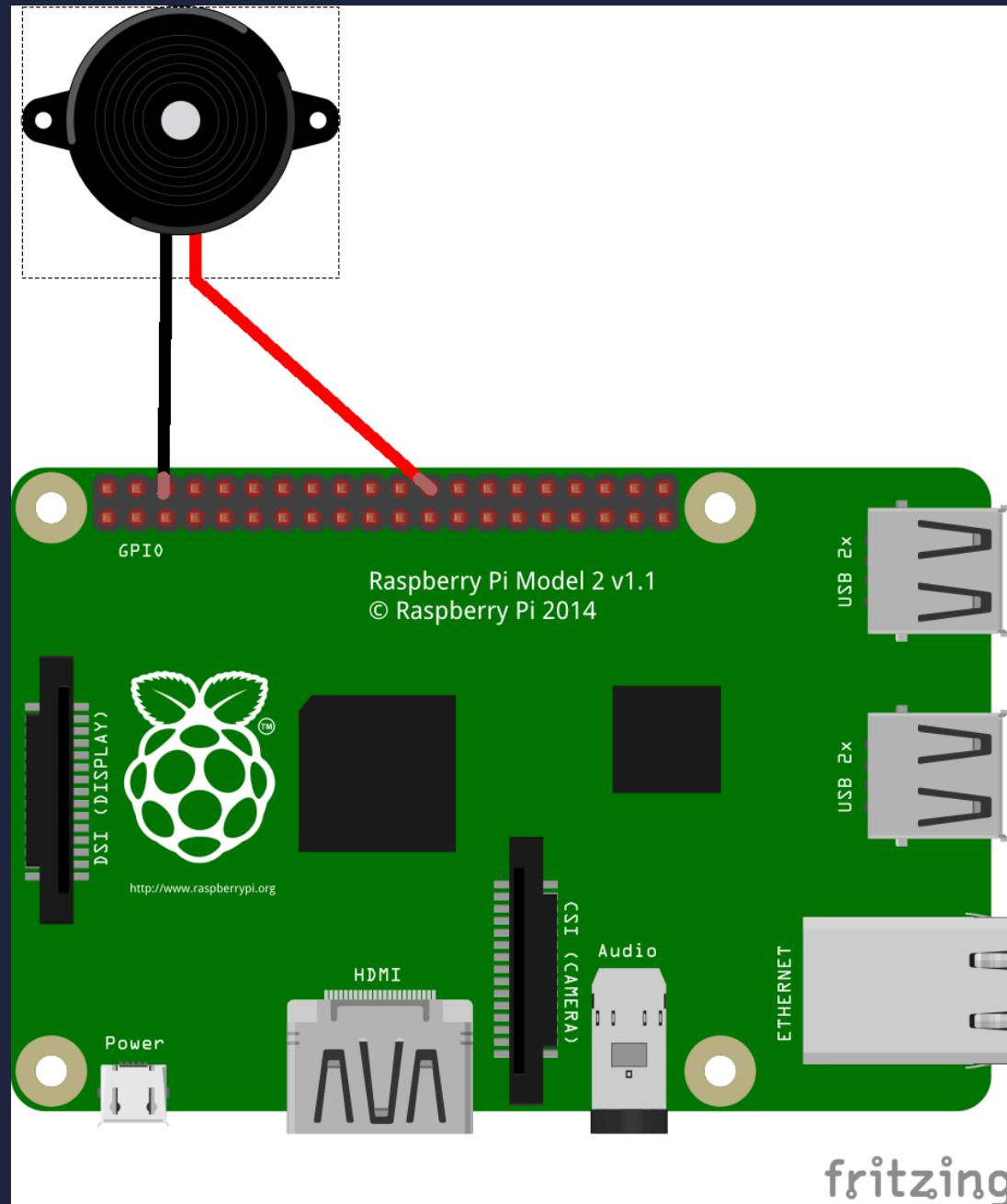


# SENSORS/COMPONENTS

# HC-SR04 ULTRASONIC SENSOR



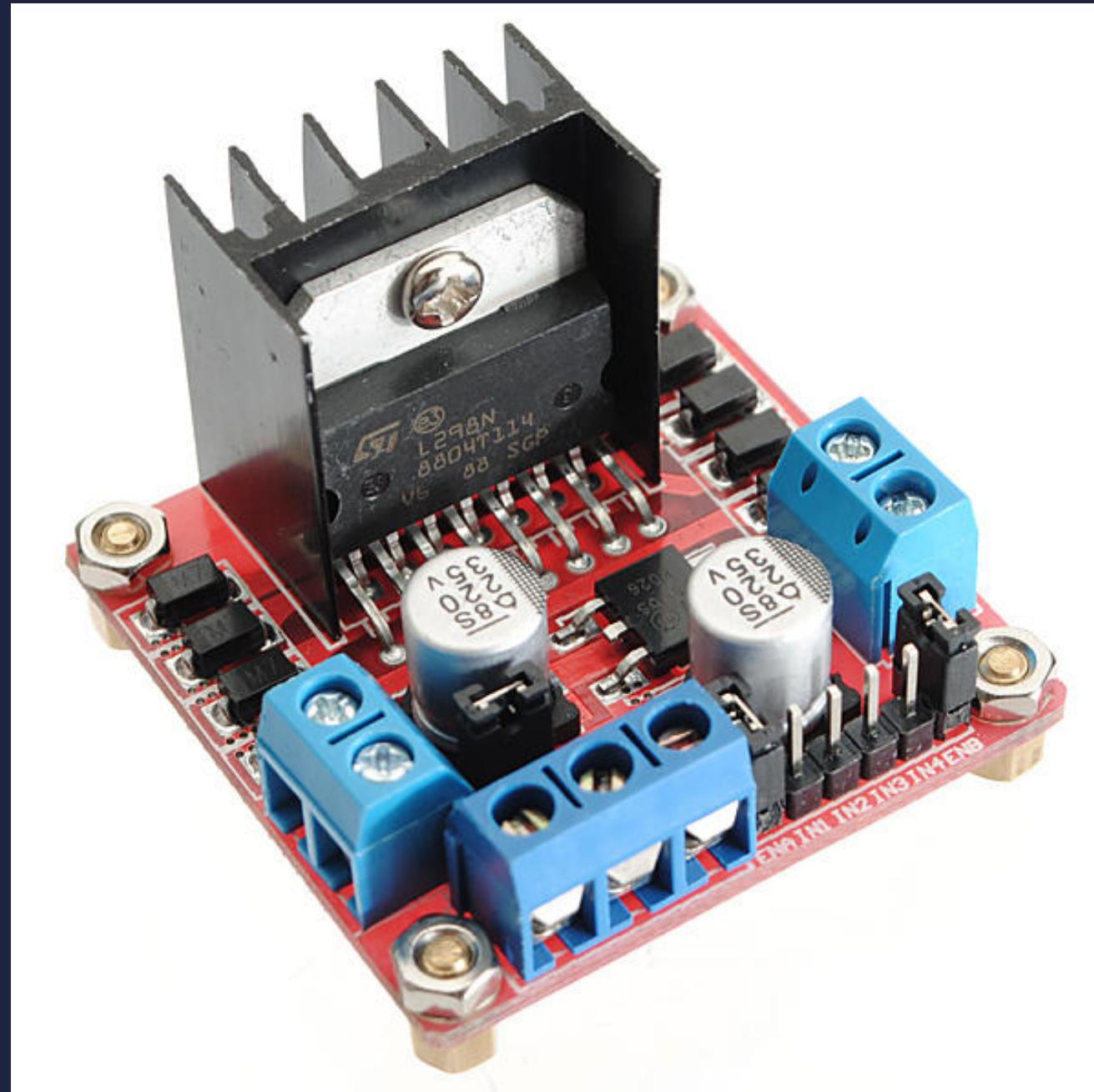
# ACTIVE BUZZER (PIEZO)



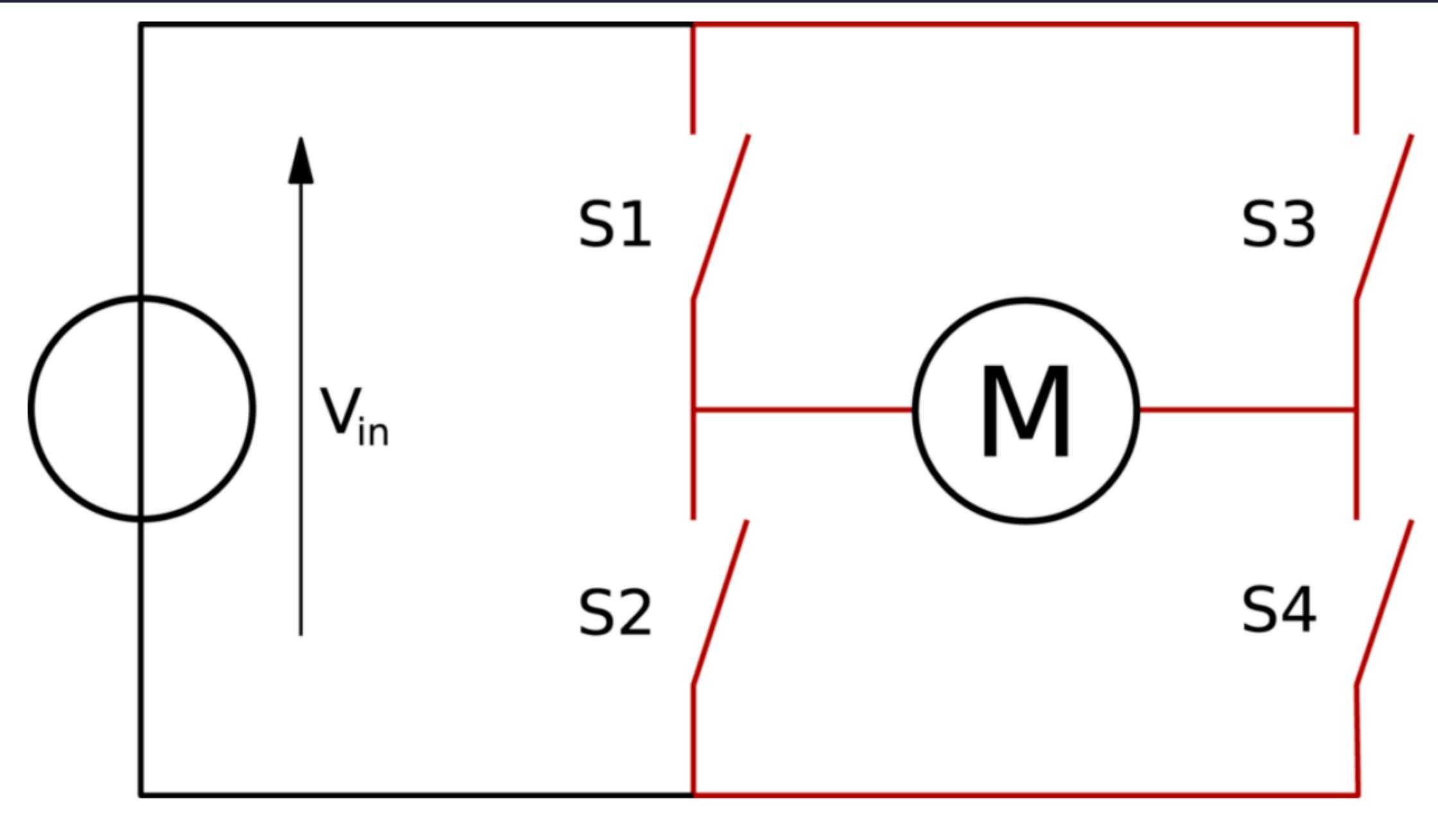
# DC GEARBOX MOTOR

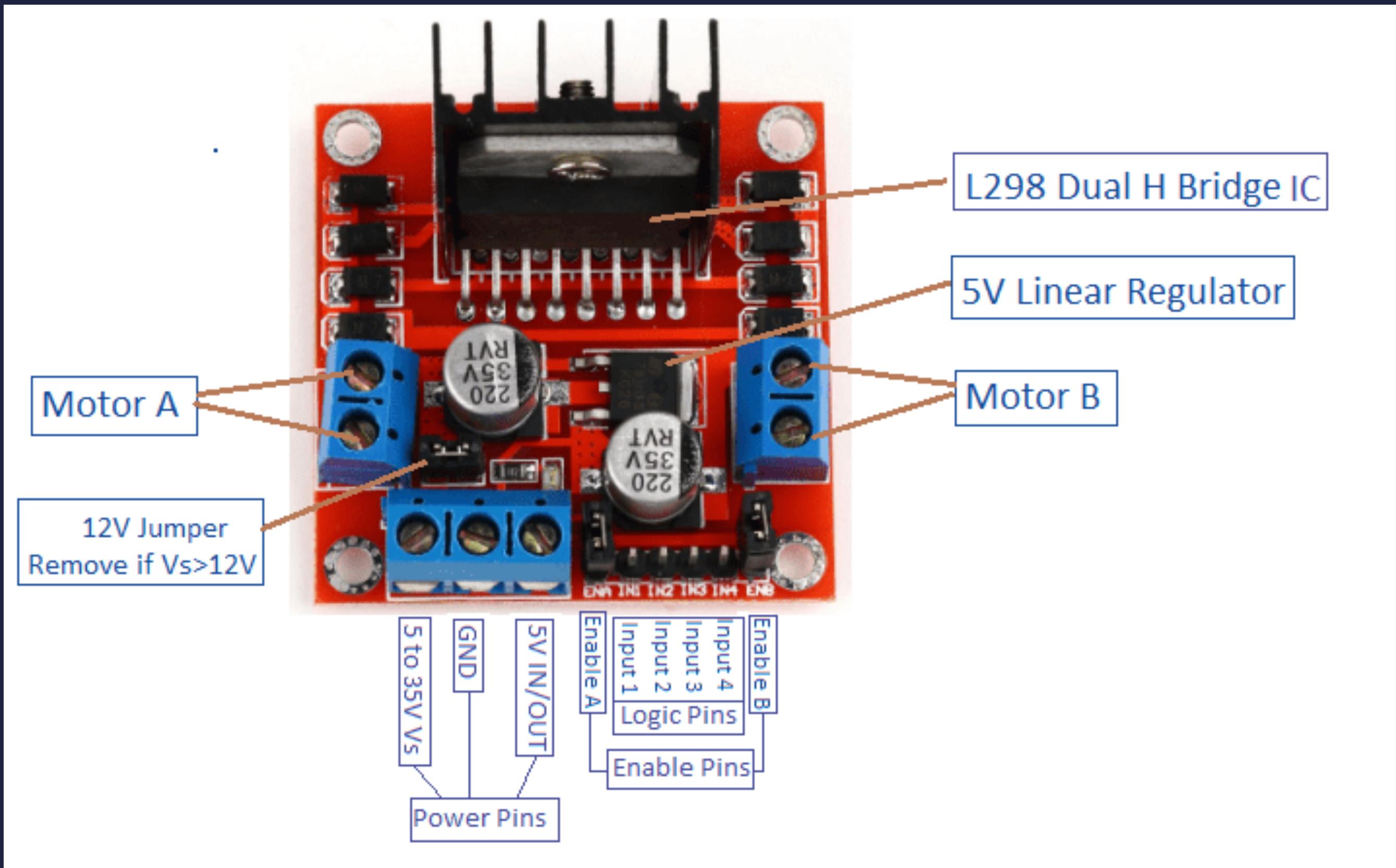


# L298N MOTOR CONTROLLER



# H BRIDGE DIAGRAM





# L298N TRUTH TABLE

ENA	IN1	IN2	MOTOR STATE
0	N/A	N/A	Off
1	0	0	Stopped (brakes)
1	0	1	Backwards
1	1	0	Forwards
1	1	1	Stopped (brakes)

# SWIFTYGPIO

uraimo / **SwiftyGPIO**

Code Issues Pull requests Actions Wiki Security Insights

A Swift library for hardware projects on Linux/ARM boards with support for GPIOs/SPI/I2C/PWM/UART/IWire.

spi-interface gpio iot swift pwm i2c-display uart serialport i2c i-wire led-strip neopixel spi led raspberry-pi

349 commits 5 branches 54 releases 19 contributors MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

uraimo	Don't try to print .pull in the CustomStringConvertible	✓ Latest commit f7dbe1c 25 days ago
.github	Update ISSUE_TEMPLATE.md	9 months ago
.travis.d	Add files to test build in Travis (#68)	last year
Examples	Update swift-tools-version for all examples	3 months ago
Sources	Don't try to print .pull in the CustomStringConvertible	25 days ago
docs	Update i2c-debugging.md	3 months ago
images	Updated README.md	3 years ago
.gitignore	Updated .gitignore for SPM	4 months ago
.travis.yml	Add files to test build in Travis (#68)	last year
CONTRIBUTING.md	Update CONTRIBUTING.md	3 years ago
LICENSE	Update LICENSE	3 years ago
Package.swift	Updated package name, new release	3 years ago

# INSTALLING SWIFT ON RASPBERRY PI

## Swift on Raspberry Pi

Controlling hardware is easy as Pi



Marc D Aupont  
Aug 26 · 7 min read



Raspberry PI 3 Model B+

# ADD/INSTALL THE SWIFT-ARM REPO

```
curl -s <https://packagecloud.io/install/repositories/swift-arm/release/script.deb.sh> | sudo bash
```

```
pi@raspberrypi:~ $ curl -s https://packagecloud.io/install/repositories/swift-arm/release/script.deb.sh | sudo bash
```

# INSTALL SWIFT

```
sudo apt-get install swift5=5.0.2-v0.4
```

```
pi@raspberrypi:~ $ sudo apt-get install swift5=5.0.2-v0.4
```

# PROJECTS BUILT WITH THE PI

# MAGIC MIRROR

The open source modular smart mirror platform.

dependencies up to date devDependencies up to date cii best practices passing license MIT build error vulnerabilities 0



## Open Source

Open Source, free and maintained by a big group of people. Got a nice idea? Send us a pull request and become a part of the big list of contributors.

[Go to the Repository »](#)



## Modular

The core of MagicMirror<sup>2</sup> contains a strong API which allows 3rd party developers to build additional modules. Modules you can use. Modules you can develop.

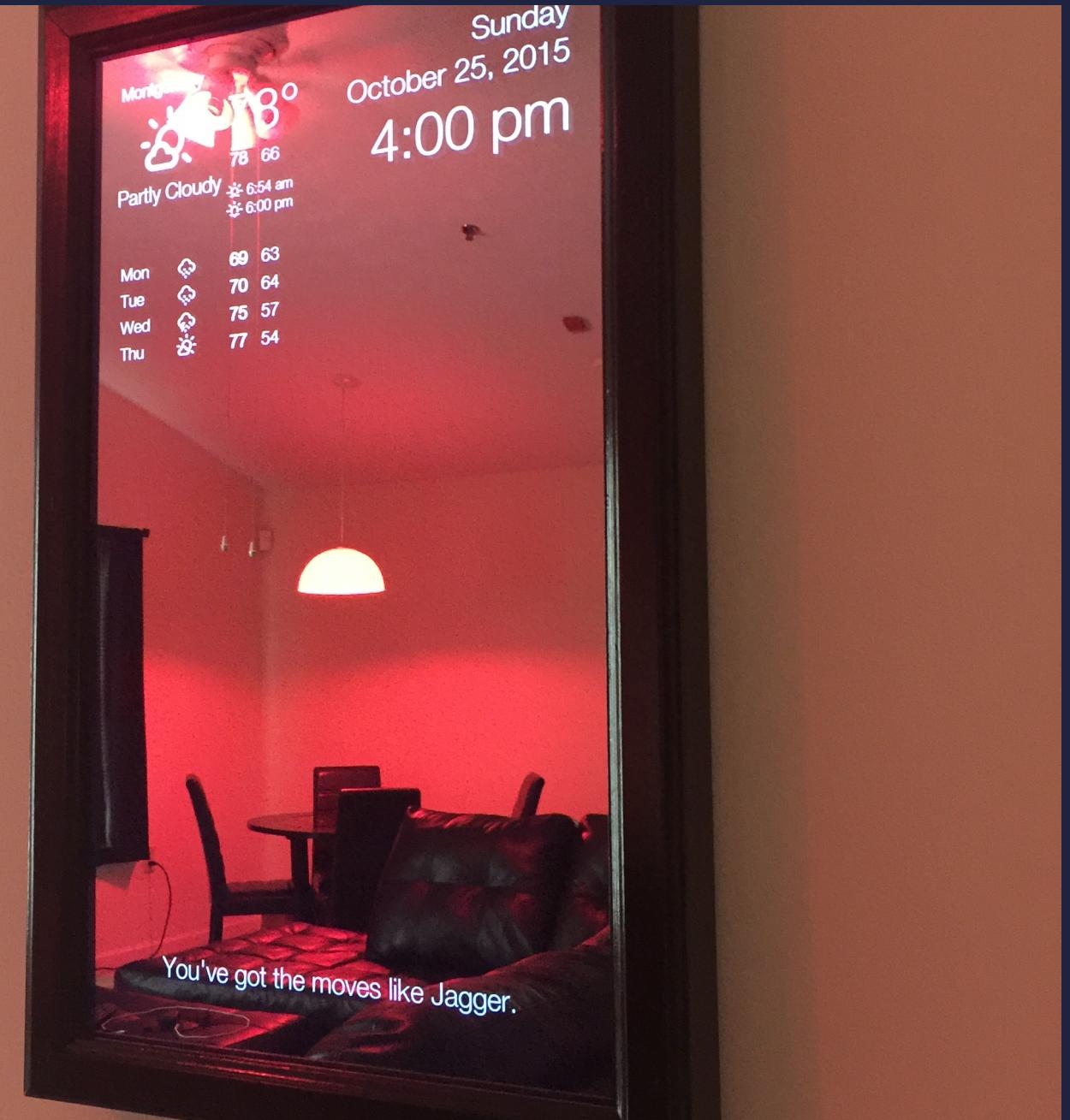
[Check out the Modules »](#)



## Community drive

On the forum you will find a big list of MagicMirror users. Share your ideas, ask your questions and get answers from the perfect place for you to start.

[Visit the Forum »](#)



# TRY!BOT MODEL 3

# PARTS LIST 1



The perseids DIY Robot Smart Car Chassis Kit with Speed Encoder, Wheels and Battery Box (2 Wheels)

by [The perseids](#)

7 customer reviews | 17 answered questions

List Price: \$14.99  
Price: **\$13.99** & FREE Returns  
You Save: \$1.00 (7%)

Thank you for being a Prime member. Get \$70 off instantly: Pay \$0.00 upon approval for the Amazon Prime Rewards Visa Card. No annual fee.

Size: **2 Wheels**

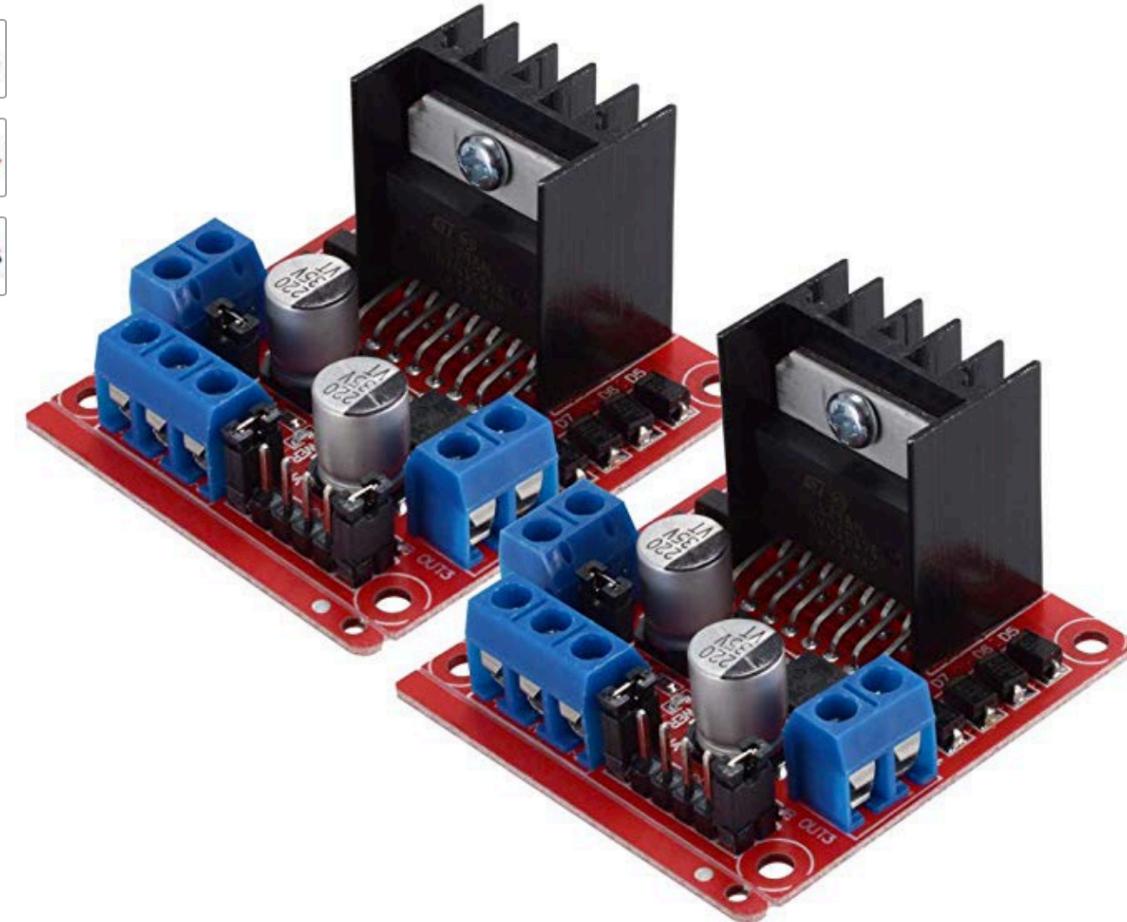
**2 Wheels**  
**\$13.99**

**4 Wheels**  
\$17.99

- Intelligent Tracing Robot car chassis, high degree of freedom, ideal for DIY subject, science education, development of hobbies and interest, educational toys for kids.
- Mechanical structure is simple, 2 DC gear motors, flexible turning, eco-friendly chassis is stability and very easy to expand.
- No add-ins is very convenient, the entire system is highly integrated. Can be used to measure distance and speed.
- Educational learning kit for beginners (kids) to get hands-on experience about Robotics & Can extend electronics system like Raspberry Pi, or Arduino etc. Realizing functions of tracing, obstacle avoidance, distance testing, speed testing, and wireless remote control.
- Size: 21 x 10cm (LxW). Wheel size: 6 x 2.7cm (Dia.xH). Motor power supply is 3V ~ 6V. All the parameter above is tested without load.

New (1) from \$13.99

# PARTS LIST 2



Roll over image to zoom in

PChero 2Packs L298N Motor Drive Controller Board Module, Dual H Bridge DC Stepper For Arduino Smart Car Robot  
by PChero  
 3 customer reviews  
Price: \$8.99  & FREE Returns  
Thank you for being a Prime member. Get \$70 off instantly: Pay \$0.00 upon approval for the Amazon Prime Rewards Visa Card. No annual fee.  
Size: 2 Packs  

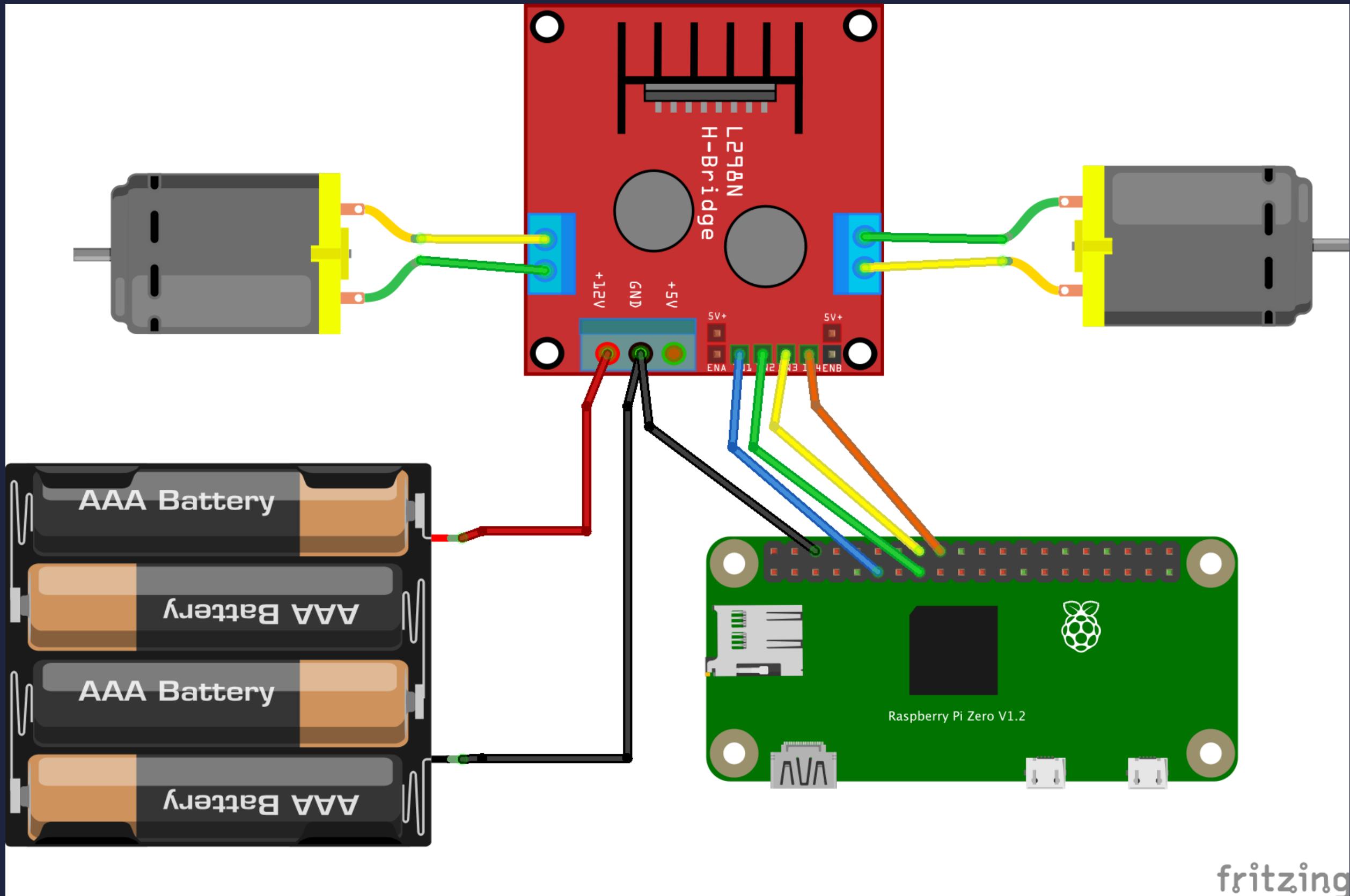
1 Pack \$6.49 	2 Packs \$8.99 
--	---

- ★ Package Included - 2pcs L298N Motor Driver Controller Board Modules.
- ★ Item Size - External size 43\*43\*27mm / 1.69\*1.69\*1.06 inch. Drive voltage 5V-35V, high working power to 46V. Large current can reach 3A MAX and continue current is 2A. Power to 25w.(NOTE: To avoid damage, please use an external 5V logic supply when using more than 12V driving voltage.)
- ★ High Quality - Use L298N as the main driving chip, so the module has such features as strong driving ability, strong anti-interference and low calorific value. And it also use large-capacity filter capacitor and diode with freewheeling protection function, thereby improve the reliability of module.
- ★ High Efficiency - Dual-channel H-bridge design, can drive one 2-phase stepper motor, one 4-phase stepper motor or two DC motors.
- ★ Easy of Use - No assembly required. This L298N board comes ready to go. Just wire up motors and power, this thing will work wonders.

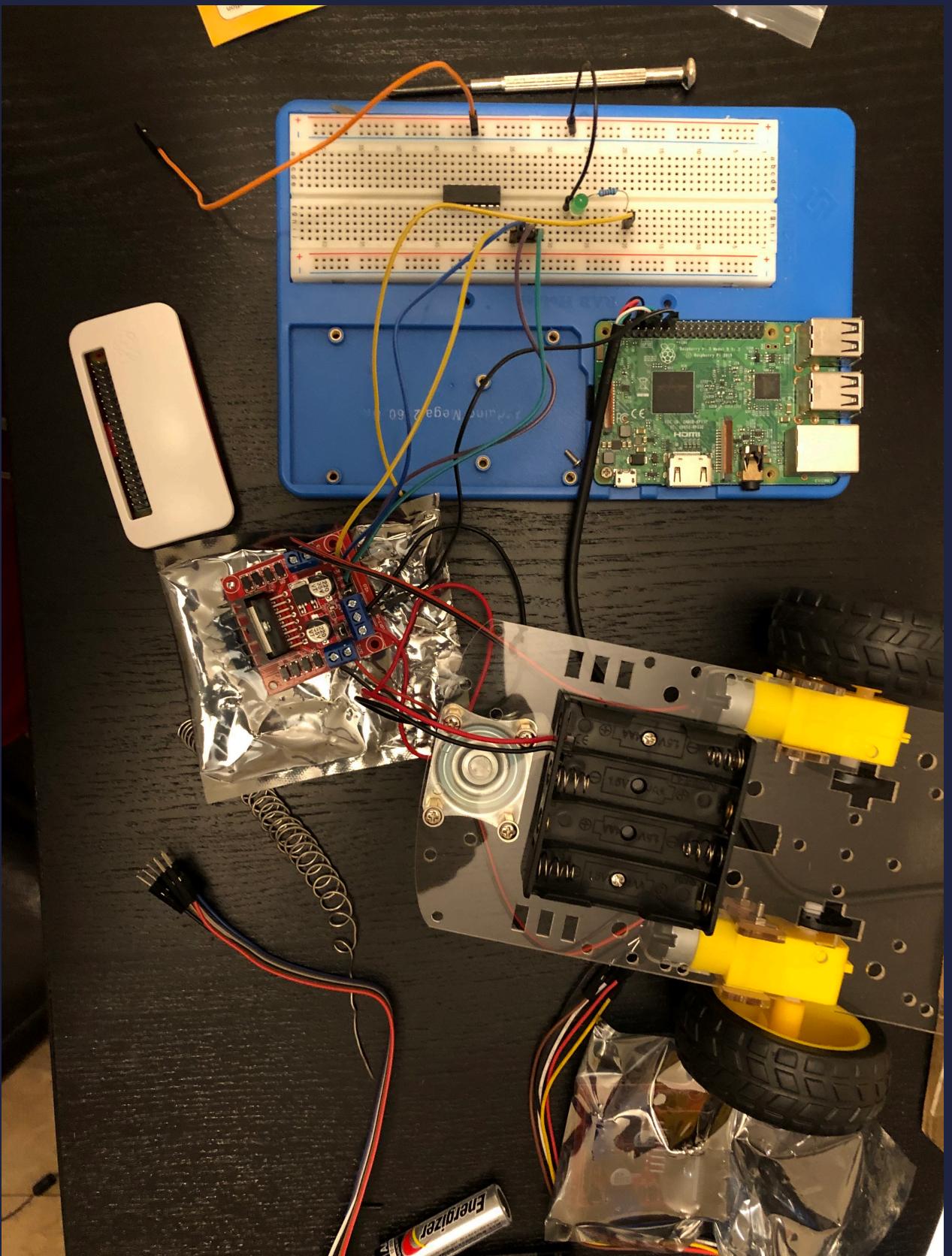
[See more product details](#)

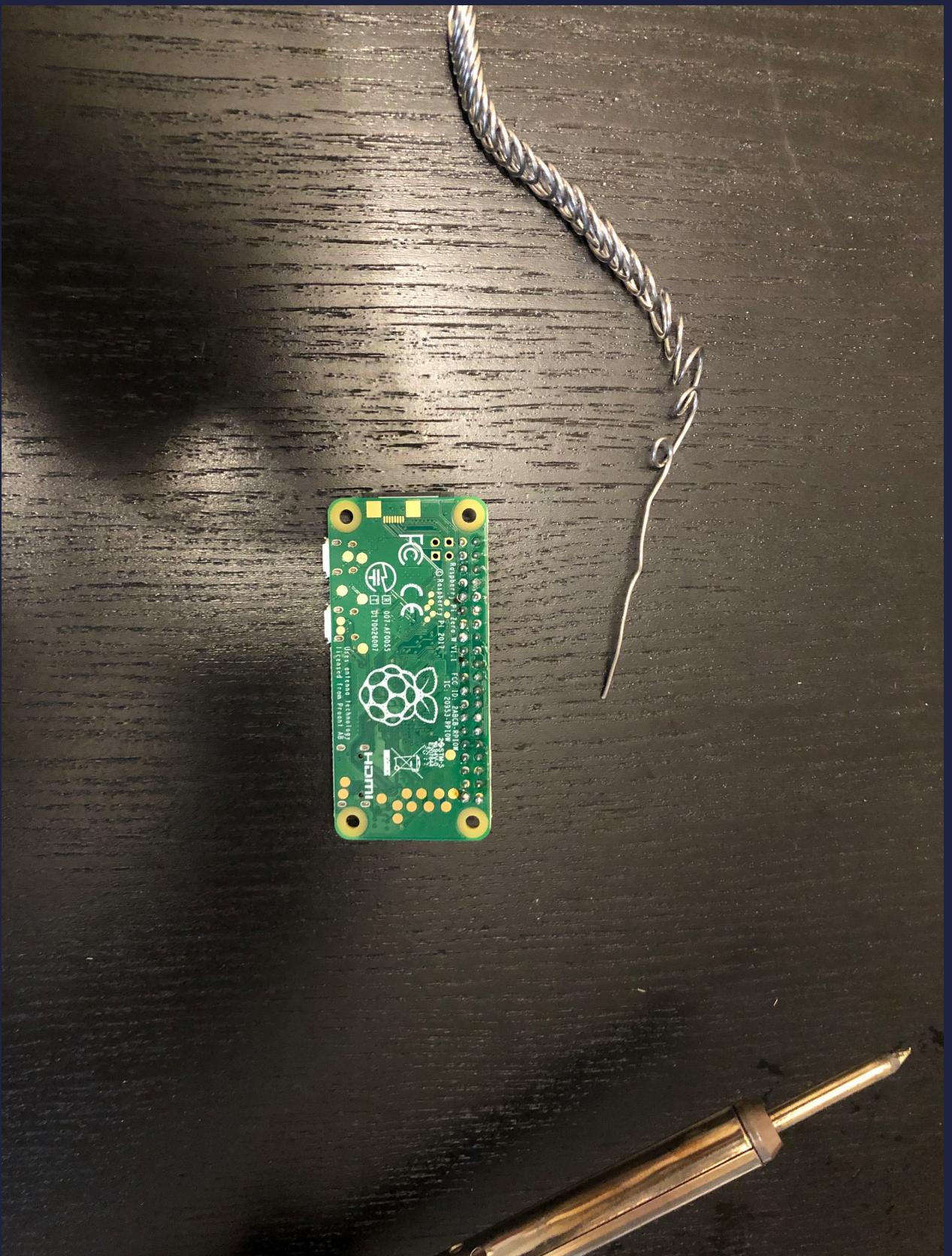
New (2) from \$8.99 

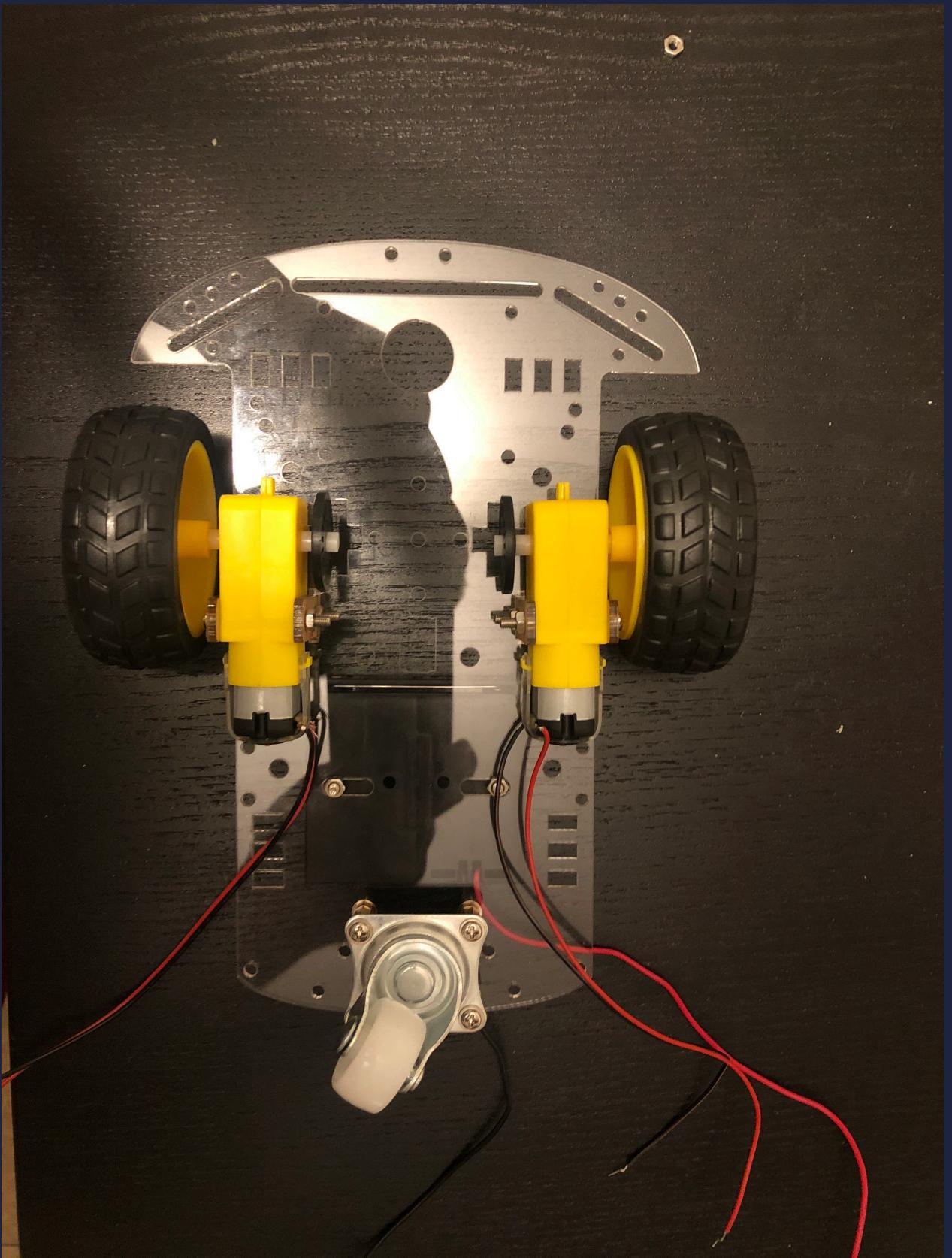
[Report incorrect product information.](#)



fritzing









# LET'S SEE THE CODE!

# THE CODE

```
import SwiftyGPIO
import Foundation

// SwiftyGPIO configuration
let gpios = SwiftyGPIO.GPIOs(for: .RaspberryPiPlusZero)

// Initialize pins
guard let pin17 = gpios[P17],
      let pin22 = gpios[P22],
      let pin23 = gpios[P23],
      let pin24 = gpios[P24] else {
    fatalError("Could not initialize pins.")
}

// Setup pin direction
pin17.direction = .OUT
pin22.direction = .OUT
pin23.direction = .OUT
pin24.direction = .OUT

// Setup pin values
print("Going Forward")
pin17.value = 1
pin22.value = 0
pin23.value = 1
pin24.value = 0
sleep(2)

print("Stopping")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 0
sleep(2)

print("Going backwards")
pin17.value = 0
pin22.value = 1
pin23.value = 0
pin24.value = 1
sleep(2)

print("Stopping")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 0
sleep(2)

print("Right Turn")
pin17.value = 0
pin22.value = 1
pin23.value = 0
pin24.value = 0
sleep(2)

print("Stopping")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 0
sleep(2)

print("Left Turn")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 1
sleep(2)

print("Stopping")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 0
```

```
import SwiftyGPIO
import Foundation

// SwiftyGPIO configuration
let gpios = SwiftyGPIO.GPIOs(for: .RaspberryPiPlusZero)

// Initialize pins
guard let pin17 = gpios[.P17],
      let pin22 = gpios[.P22],
      let pin23 = gpios[.P23],
      let pin24 = gpios[.P24] else {
    fatalError("Could not initialize pins.")
}
```

```
import SwiftyGPIO
import Foundation

// SwiftyGPIO configuration
let gpios = SwiftyGPIO.GPIOs(for: .RaspberryPiPlusZero)

// Initialize pins
guard let pin17 = gpios[.P17],
      let pin22 = gpios[.P22],
      let pin23 = gpios[.P23],
      let pin24 = gpios[.P24] else {
    fatalError("Could not initialize pins.")
}
```

```
// Setup pin direction  
pin17.direction = .OUT  
pin22.direction = .OUT  
pin23.direction = .OUT  
pin24.direction = .OUT
```

```
// Setup pin values  
print("Going Forward")  
pin17.value = 1  
pin22.value = 0  
pin23.value = 1  
pin24.value = 0  
sleep(2)
```

```
// Setup pin direction  
pin17.direction = .OUT  
pin22.direction = .OUT  
pin23.direction = .OUT  
pin24.direction = .OUT
```

```
// Setup pin values  
print("Going Forward")  
pin17.value = 1  
pin22.value = 0  
pin23.value = 1  
pin24.value = 0  
sleep(2)
```

```
print("Stopping")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 0
sleep(2)
```

```
print("Going backwards")
pin17.value = 0
pin22.value = 1
pin23.value = 0
pin24.value = 1
sleep(2)
```

```
print("Stopping")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 0
sleep(2)
```

```
print("Going backwards")
pin17.value = 0
pin22.value = 1
pin23.value = 0
pin24.value = 1
sleep(2)
```

```
print("Stopping")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 0
sleep(2)
```

```
print("Right Turn")
pin17.value = 0
pin22.value = 1
pin23.value = 0
pin24.value = 0
sleep(2)
```

```
print("Stopping")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 0
sleep(2)
```

```
print("Right Turn")
pin17.value = 0
pin22.value = 1
pin23.value = 0
pin24.value = 0
sleep(2)
```

```
print("Stopping")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 0
sleep(2)
```

```
print("Left Turn")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 1
sleep(2)
```

```
print("Stopping")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 0
```

```
print("Stopping")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 0
sleep(2)
```

```
print("Left Turn")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 1
sleep(2)
```

```
print("Stopping")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 0
```

```
print("Stopping")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 0
sleep(2)
```

```
print("Left Turn")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 1
sleep(2)
```

```
print("Stopping")
pin17.value = 0
pin22.value = 0
pin23.value = 0
pin24.value = 0
```

# SETUP REMOTE BUILD

[Code](#)[Issues 5](#)[Pull requests 0](#)[Projects 0](#)[Security](#)[Insights](#)

Build Swift Projects on a Remote Machine within Xcode.

44 commits

2 branches

6 releases

1 contributor

MIT

Branch: master ▾

New pull request

Create new file

Upload files

Find File

Clone or download ▾



thomaspaulmann Merge pull request #5 from thomaspaulmann/connection-unexpectedly-closed ...

Latest commit 7de02bb on Nov 8, 2016

Design

Add squared logo

3 years ago

Example

Rename external build target to Swish in example

3 years ago

Scripts

Fix unexpectedly closing of connection to remote

3 years ago

.gitignore

Initial commit

3 years ago

Formula.rb

Add template for homebrew formula

3 years ago

LICENSE

Initial commit

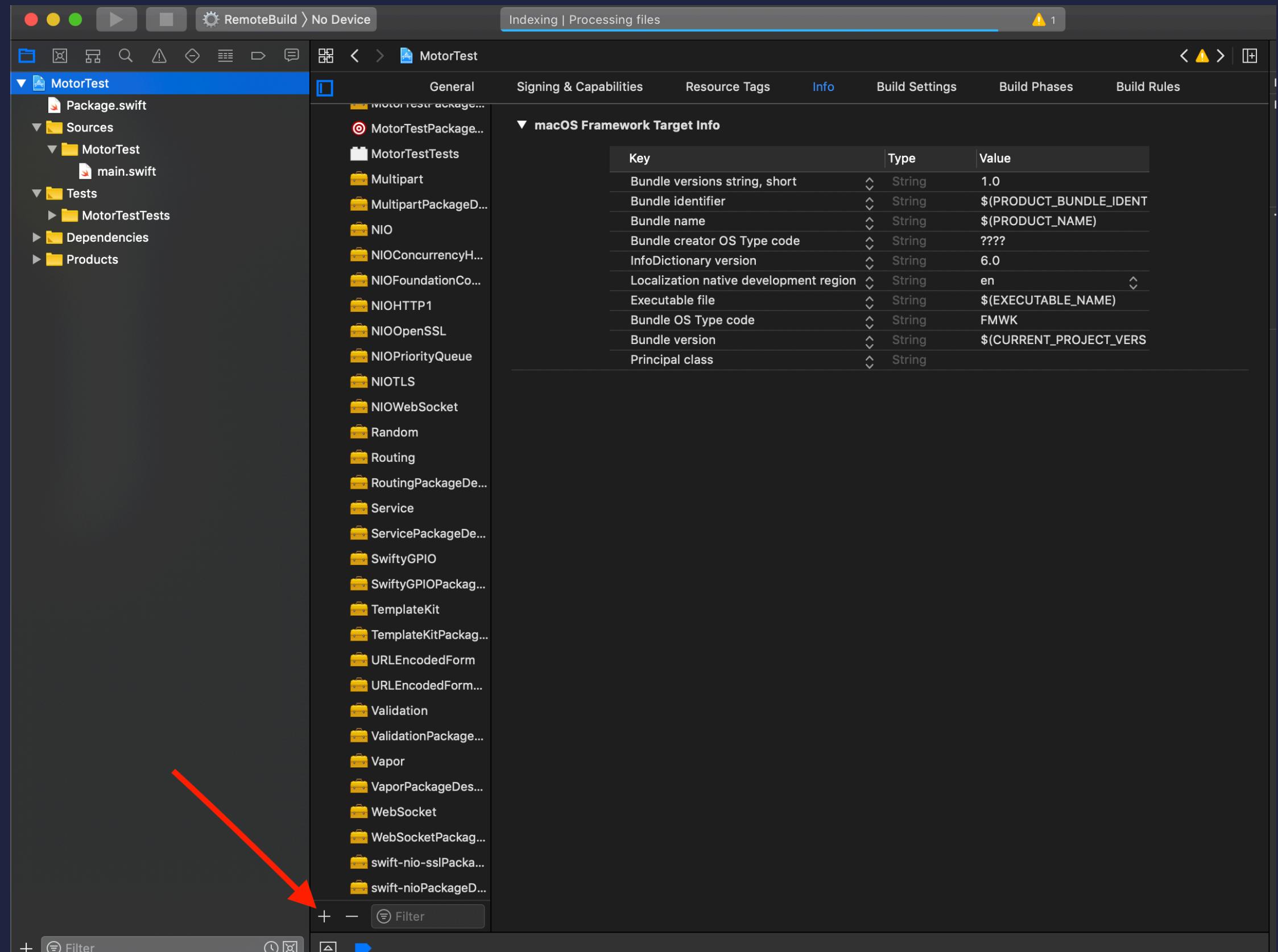
3 years ago

README.md

Update README.md

3 years ago

README.md



Choose a template for your new target:

iOS

watchOS

tvOS

macOS

Cross-platform

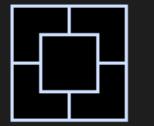
Filter

**Application**



Cross-platform  
Game

**Other**



Aggregate



External  
Build System



In-App Purchase  
Content

Cancel

Previous

Next

Choose options for your new target:

Product Name: **RemoteBuild**

Team: **Marc Aupont (Personal Team)** 

Organization Name: **Marc Aupont**

Organization Identifier: **com.digimarktech**

Bundle Identifier: **com.digimarktech.RemoteBuild**

Build Tool: **/bin/bash**

Project: **MotorTest** 

**Cancel**

**Previous**

**Finish**

MotorTest

< > | [ ]

Resource Tags Info Build Settings Build Phases

MOTORTEST package...

MotorTestTests

Multipart

MultipartPackageD...

NIO

NIOConcurrencyH...

NIOFoundationCo...

NIOHTTP1

NIOOpenSSL

NIOPriorityQueue

NIOTLS

NIOWebSocket

Random

Routing

RoutingPackageDe...

Service

ServicePackageDe...

SwiftyGPIO

SwiftGPIO

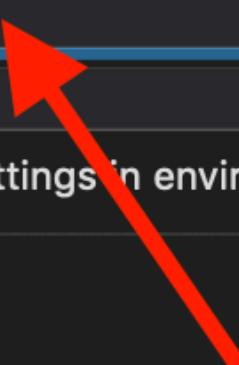
▼ External Build Tool Configuration

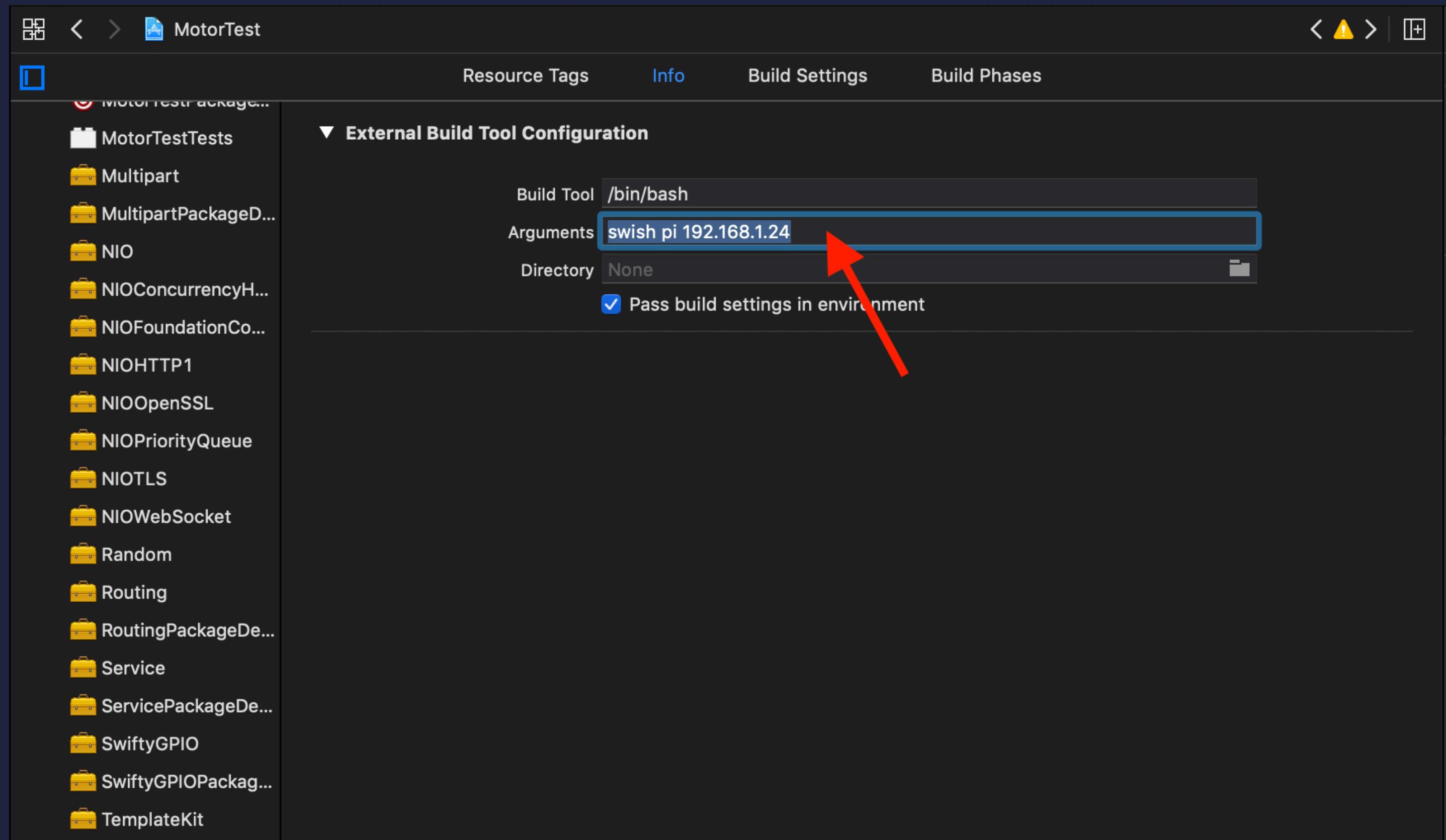
Build Tool /bin/bash

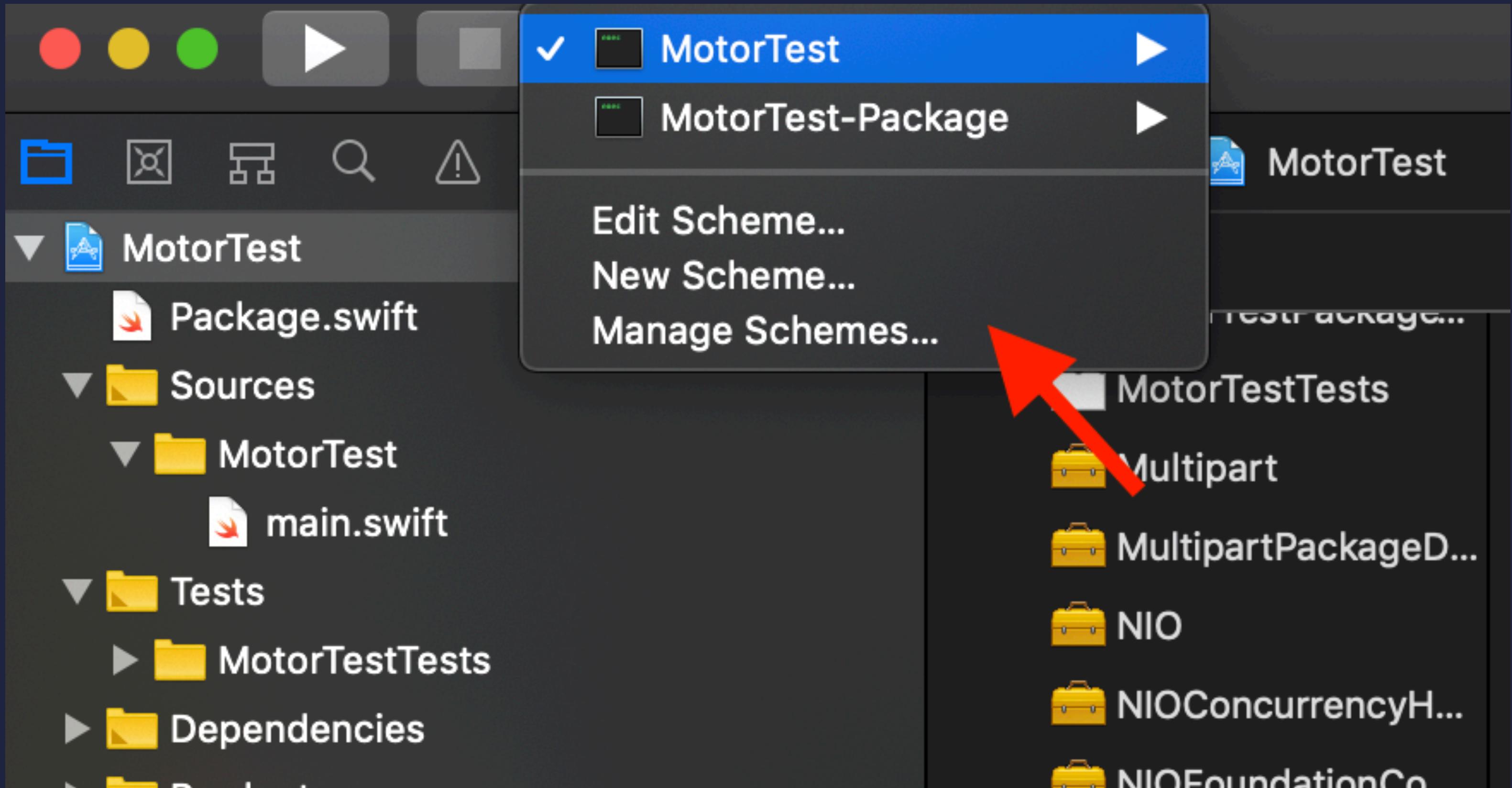
Arguments **\$(ACTION)**

Directory None

Pass build settings in environment







Autocreate schemes

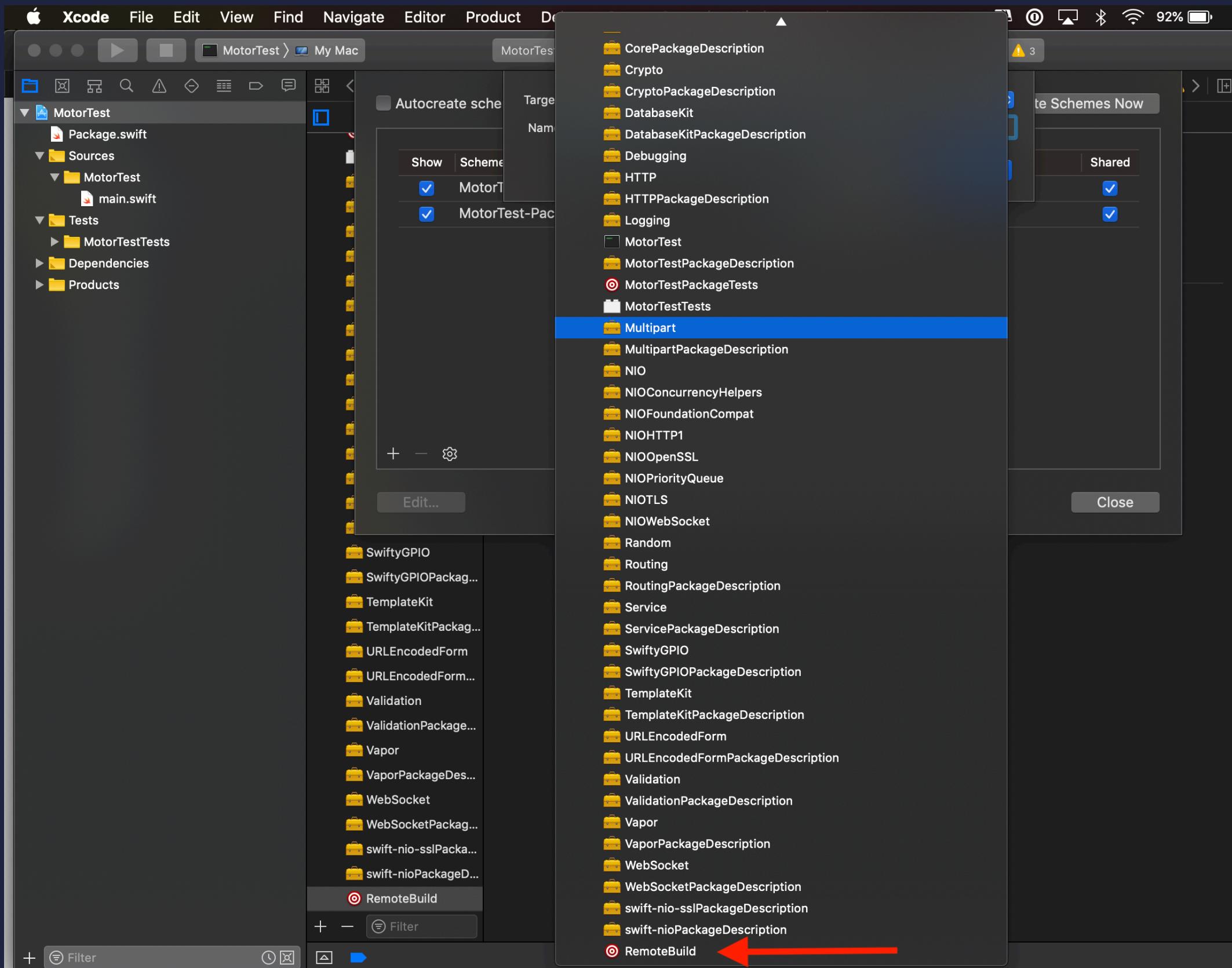
Autocreate Schemes Now

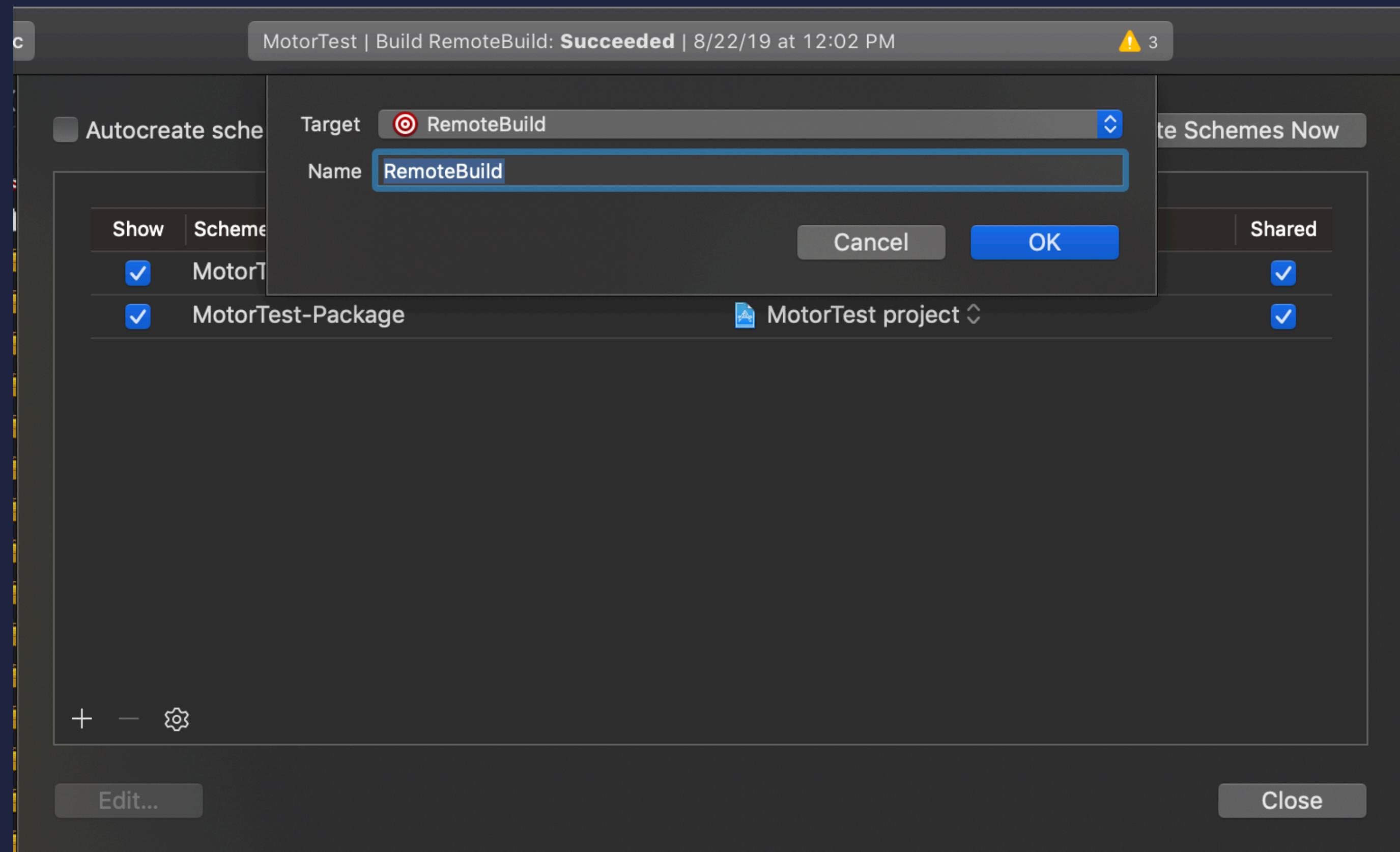
Show	Scheme	Container	Shared
<input checked="" type="checkbox"/>	MotorTest	 MotorTest project ◇	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	MotorTest-Package	 MotorTest project ◇	<input checked="" type="checkbox"/>



Edit...

Close





Mac

MotorTest | Build RemoteBuild: **Succeeded** | 8/22/19 at 12:02 PM

⚠ 3

 Autocreate schemes**Autocreate Schemes Now**

Show	Scheme	Container	Shared
<input checked="" type="checkbox"/>	MotorTest	MotorTest project ▾	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	MotorTest-Package	MotorTest project ▾	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RemoteBuild	MotorTest project ▾	<input checked="" type="checkbox"/>



+ - ⚙

**Edit...****Close**

▼  **Prepare build**  
✓ Workspace MotorTest | Scheme RemoteBuild | Destination My Mac  
Using new build system  
Planning build  
Using build description from memory

▼  **Build target RemoteBuild**  
✓ Project MotorTest | Configuration Debug | Destination My Mac | SDK macOS 10.14  
✓ Run external build tool 18.2 seconds

-----  
Configuration:  
-----

Username: pi  
Hostname: 192.168.1.24  
Destination: ~/Swish/MotorTest

-----

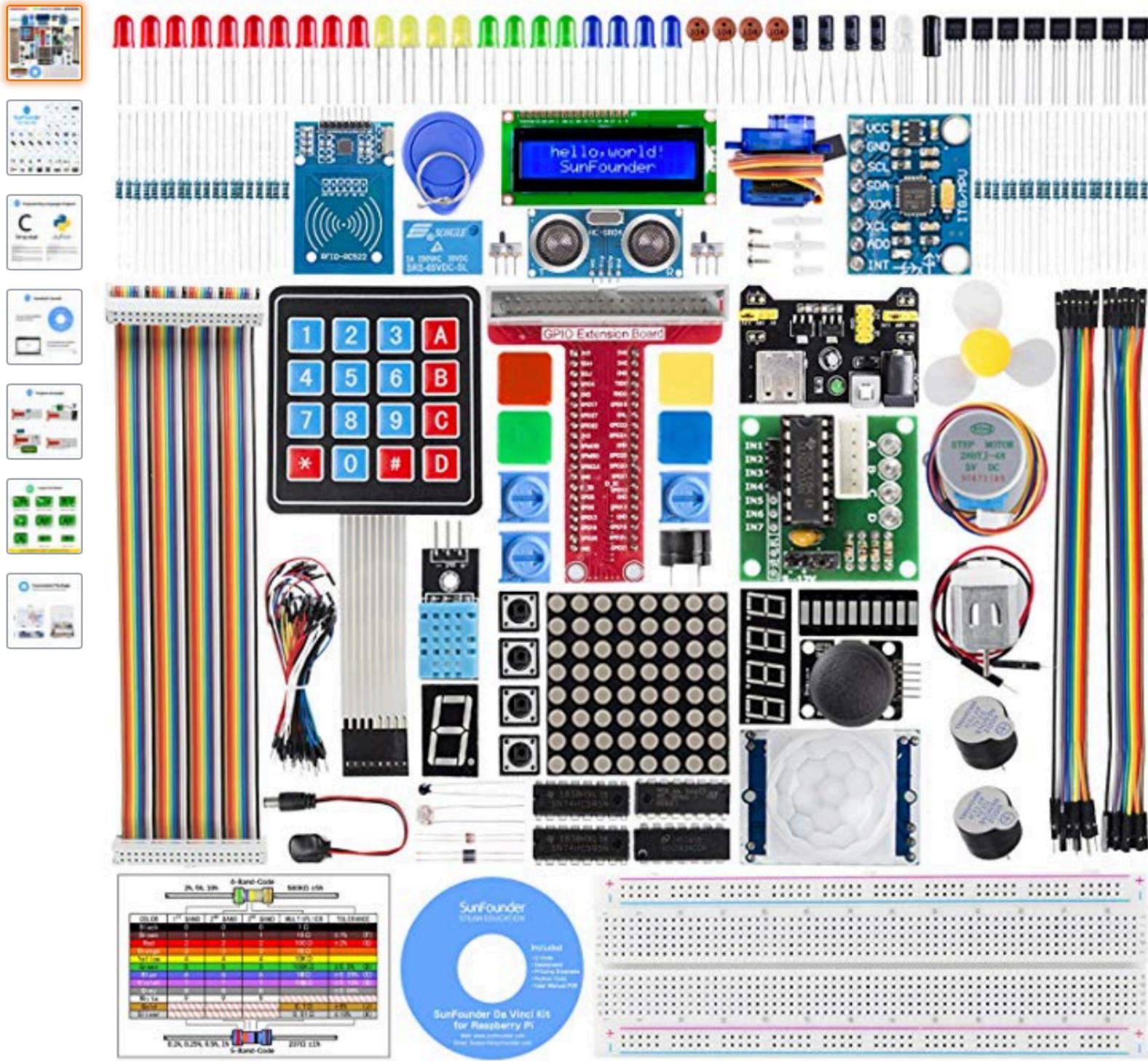
🚀 Start syncing project to pi@192.168.1.24:~/Swish/MotorTest  
building file list ...  
0 files...  
. /  
Package.resolved  
700 14% 0.00kB/s 0:00:00  
Package.swift  
829 100% 809.57kB/s 0:00:00

sent 995 bytes received 124 bytes 2238.00 bytes/sec  
total size is 8052 speedup is 7.20  
💻 Did sync project to remote machine  
🔨 Start building project  
/home/pi/.bash\_profile: No such file or directory  
[1/3] Compiling Swift Module 'SwiftyGPIO' (10 sources)  
[2/3] Compiling Swift Module 'MotorTest' (1 sources)  
[3/3] Linking ./build/armv6-unknown-linux-gnueabihf/debug/MotorTest  
🎉 Did build project on your remote machine

Build succeeded 9/3/19, 9:43 PM 18.3 seconds  
✓ No issues

# FUTURE IMPROVEMENTS TO ENHANCE PROJECT

# HOW DO YOU GET STARTED?



Roll over image to zoom in

# SunFounder Raspberry Pi Starter Kit for Raspberry Pi 4B 3 B+, 331 Pages Detailed Tutorials Support Python C 210 Items 30 Projects Included, Learn Electronics and Programming for Raspberry Pi Beginners

by SunFounder

★★★★★ 5 139 customer reviews | 32 answered questions

Price: **\$46.99** ✓prime & FREE Returns

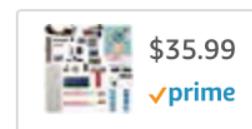
Coupon  Save an extra 10% when you apply this coupon. [Details](#)

Get \$70 off instantly: Pay ~~\$0.00~~ **\$46.99** upon approval for the Amazon Prime Rewards Visa Card. No annual fee.

Color: **DavinciKit**



**\$46.99**  
✓prime



**\$35.99**  
✓prime



**\$79.98**  
✓prime



**\$36.79**  
✓prime

- **PROGRAMMING LANGUAGES SUPPORT:** C Code, Python Code(compatible with 2&3) are provided to learn raspberry pi for beginners
- **26 MODULES:** There are 26 commonly used input and output components and modules and some basic electronic devices which can provide powerful assistance in programming learning.
- **30 PROJECTS PROVIDED:** This Kit provides you with 30 lessons for reference and learning, including 26 basic I/o lessons and 4 simple practical examples.
- **331 PAGES DETAILED MANUAL** (include basic electronics knowledge) and complete code -> Download Link: <https://github.com/sunfounder/davinci-kit-for-raspberry-pi>
- **COMPATIBLE MODELS:** Raspberry Pi 4B / 3B+ / 3B / 3A+ / 2B / 1B+ / 1A+ / Zero W / Zero. (NOT included in this kit), to compatible with Raspberry Pi 4 Model B Manual Link: <https://github.com/sunfounder/davinci-kit-for-raspberry-pi>

# WHY IS THE PI IMPORTANT?

# WHY SWIFT ON RASPBERRY PI?

Python is a trash language.  
– Paige Bailey

# RECAP

# THANKS

