

AUTODESK
Instructables

Build a Key for the Copenhagen Wheel With an ESP32 Microcontroller

By [slviajero](#) in [CircuitsReuse](#)
Published Jun 5th, 2024



Introduction: Build a Key for the Copenhagen Wheel With an ESP32 Microcontroller



The Copenhagen Wheel was a big thing when it came out in 2017. Crowdfunded, the stylish red wheel came out as a solution for urban mobility. An existing bike could be reused as an electric city bike. The user's smartphone was the control panel and the battery was in the wheel. Putting the wheel into a bike was done quickly.

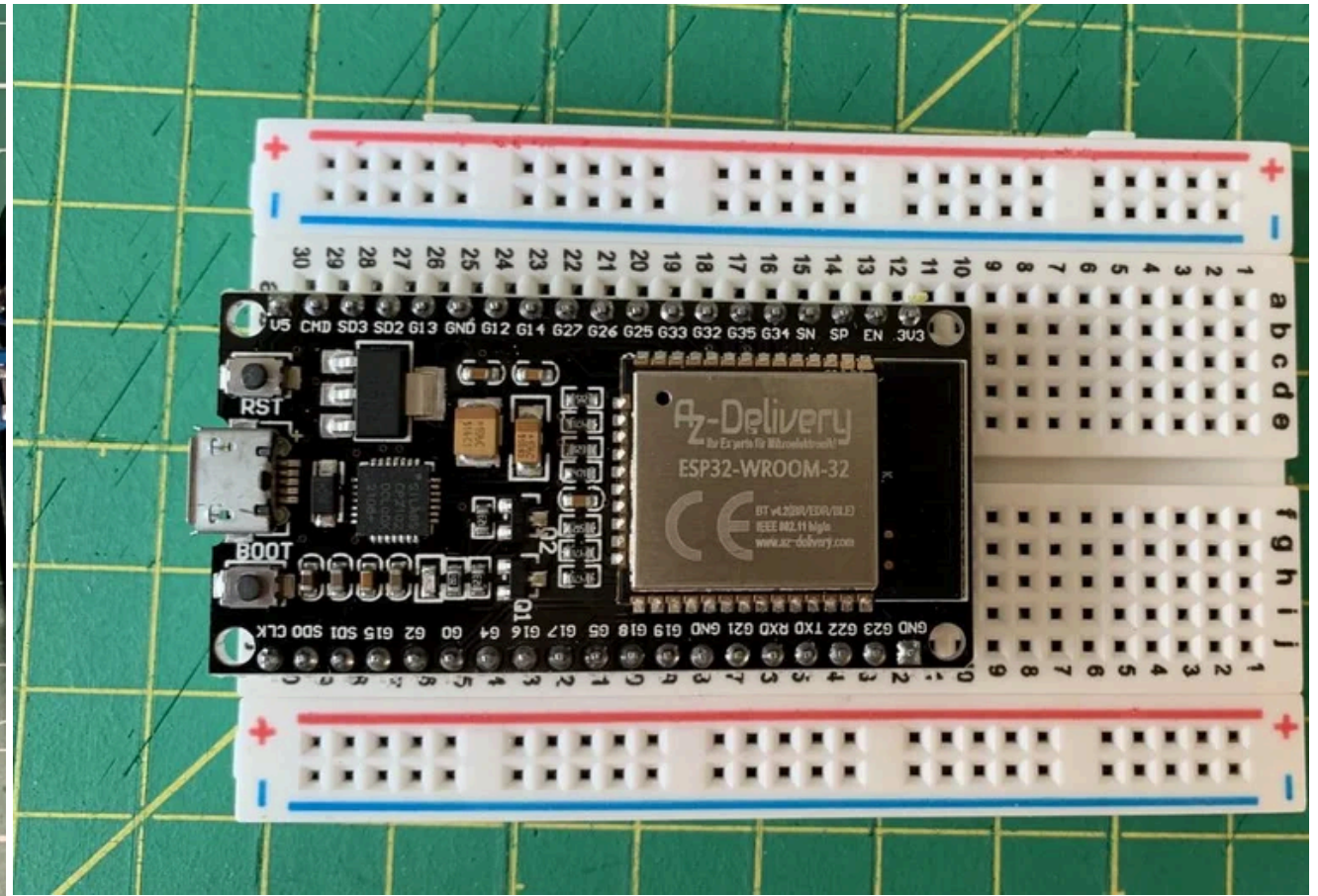
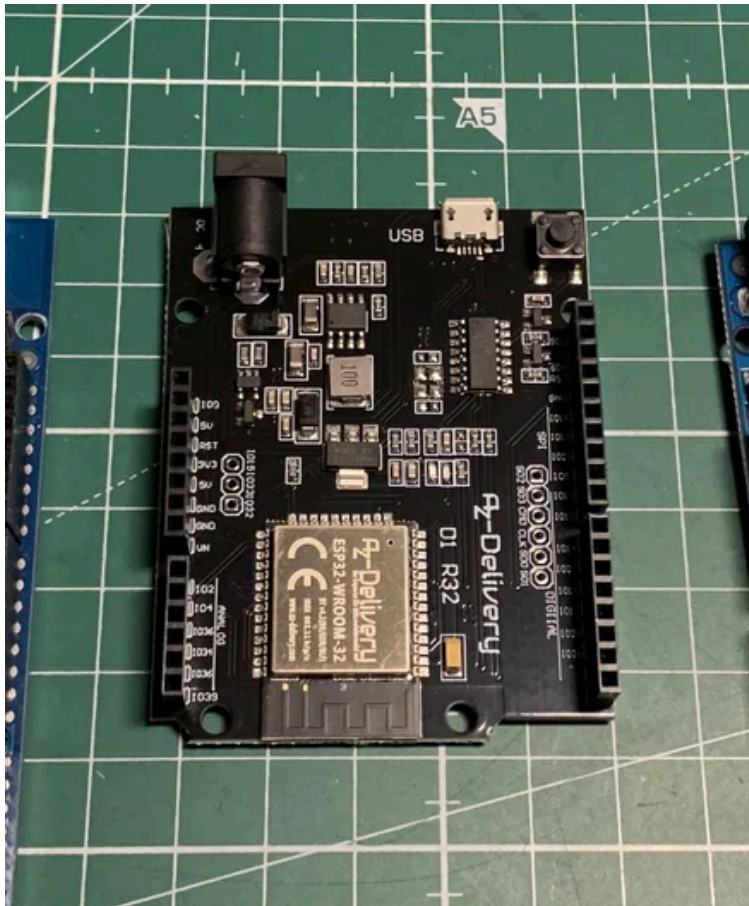
I liked the sustainability of the concept. An existing bike would be used to build something new. That's why I contributed early in the crowdfunding campaign.

In the highly competitive market of electric bikes, the Copenhagen wheel never became popular. It was discontinued. Then, by the end of 2023, the manufacturer Superpedestrian went bankrupt and the app backend was switched off. All users were locked out of their wheels as the app was needed to start the wheel.

This instructable is how to build a key to start the wheel again with an ESP32 microcontroller board. The software is not mine. Very knowledgeable people built the code. Big kudos to them. I just took the ESP32 code and tried it out.

The intention of this text is a step by step tutorial how to implement the key for people with a bit of Arduino know how. I will write more software for the wheel on the basis of the existing code. This will be published later.

Supplies



To build this you need of course a bicycle with a Copenhagen wheel.

For the key you need a ESP32 microcontroller board. I started with the ESP32 Wemos D1 board. It is rather big for a key but I had it lying around. Any other board will do as well. Later in this instructable there will be an idea for a small form factor build.

You need the Arduino IDE. I use version 1.8.x here which is outdated. Still, it is so common and so many people have it, that it is still standard for Arduino projects. The IDE 2.x works fine as well. See here: [Software | Arduino](#)

The ESP32 controller needs the current ESP32 core. This can be downloaded from the Expressif website. There is also a detailed instruction on how to download this: [Installing -- — Arduino ESP32 latest documentation \(espressif.com\)](#)

Then, you also need the sketch [slviajero/cphw-esp32 \(github.com\)](https://github.com/slviajero/cphw-esp32) with the key software.

A power bar and a cable can be used as power supply.

Step 1: Get the Arduino and ESP32 Software

Basis for using Arduino type microcontrollers is the Arduino IDE. There is an old 1.8.19 version with a simple user interface and the modern 2.x version. Both can be downloaded from the official Arduino website.

Go to [Software | Arduino](#) and download the version for your OS. Windows, Mac and Linux are supported.

After this you will need the ESP32 core. In the past it was not included in the Arduino distribution. As Arduino now also supports ESP32 boards, this might change in the future. If you want to use one of the many ESP32 boards on the market, you will need the core anyway.

Follow the instructions in [Installing - - — Arduino ESP32 latest documentation \(espressif.com\)](#).

In the 2.x IDE the ESP32 core can be installed from the IDE directly. Select the core in the board menu and tell the IDE to install.

After a successful install, you should see the many ESP32 board in the ESP32 boards menu in the IDE.

Some platforms still need a COM: or Serial driver install. You might need a UART driver like this [CP210x USB to UART Bridge VCP Drivers - Silicon Labs \(silabs.com\)](#).

Step 2: Get the Key Software

Once you have a working IDE, you can import the sketch. Currently it is just one file in the repo

[slviajero/cphw-esp32 \(github.com\)](https://github.com/slviajero/cphw-esp32).

Download *BLE_client.ino* and place it in a folder BLE_client. Then open the IDE and open this folder.

You should be able to compile the sketch now. If this works, connect the board, select the right port and upload the sketch.

In the serial monitor of the Arduino IDE you should see the status messages of the sketch IF you have set the board rate correctly to 115000 baud.

Step 3: Find Your Wheel Characteristic

Once the sketch is uploaded, start the wheel and then reset the controller. It should report a connection to the wheel and then you a string called wheel characteristic.

The wheel characteristic is a number like 00403160970202 or J144800550. It identifies your wheel.

Look into the file passcodes.csv in the repo and search for your wheel. You will find lines like this

```
00403160970202,E3EFD15276F93B7ECE0C532558214FAF
```

The code behind the comma is the passcode of your wheel.

Open the sketch again and modify these lines

```
// REPLACE THIS WITH YOUR PASSCODE  
#define PASSCODE "101112131415161718191A1B1C1D1E1F"
```

adding your passcode.

Then upload the software again on the controller and reset it.

Step 4: Use the Key



Now you have a key that unlocks your wheel. I use a power bar to start the controller.

Power it up, bring it close to the wheel and switch it on. The wheel should start now with the default run mode. Maybe a reset of the board is needed sometimes.

Once the wheel is stated you can power off the controller and enjoy riding. As long as the wheel moves, the key is not needed any more.

Step 5: What's Next

Currently the app is very basic. I am working on a better app with a display and buttons to change the wheel run mode. I will also build a small for m factor device.

Wheel users, join us in our facebook group [Copenhagen Wheel Users | Facebook](#)