Tutorials

- **How to make the Raspberry Pi display a web page fullscreen at power on**
- **How to power the Raspberry Pi Pico**
- **How to make two Raspberry Pi Pico communicate via Wifi**
- **How to control Raspberry Pi Pico via Wifi**
- **How to install Node.js on Raspberry Pi**

How to make the Raspberry Pi display a web page fullscreen at power on

This short tutorial should be enough for you to be able to setup a Raspberry Pi with GUI mode and auto login enabled to disable a web page full screen automatically when it turns on.

1. Copy `full-screen.sh` into the home folder of the pi user, and run the following command: `/home/pi/full-screen.sh`

```
sudo chmod +x full-screen.sh
```

full-screen.sh

```
export DISPLAY=:0.0
xset s off
xset -dpms
xset s noblank
chromium-browser --noerrdialogs --kiosk https://google.co.uk --incognito
```

2. Create a folder called autorun inside the `.config` folder of the home folder of the pi user: `/home/pi/.config/autorun/kiosk.desktop`

`cd /home/pi/.config` `mkdir autorun`

kiosk.desktop

```
[Desktop Entry]
Type=Application
Exec=/home/pi/full-screen.sh
Hidden=false
X-GNOME-Autostart-enabled=true
Name=kiosk
```

3. Reboot the Raspberry Pi.

```
sudo reboot now
```

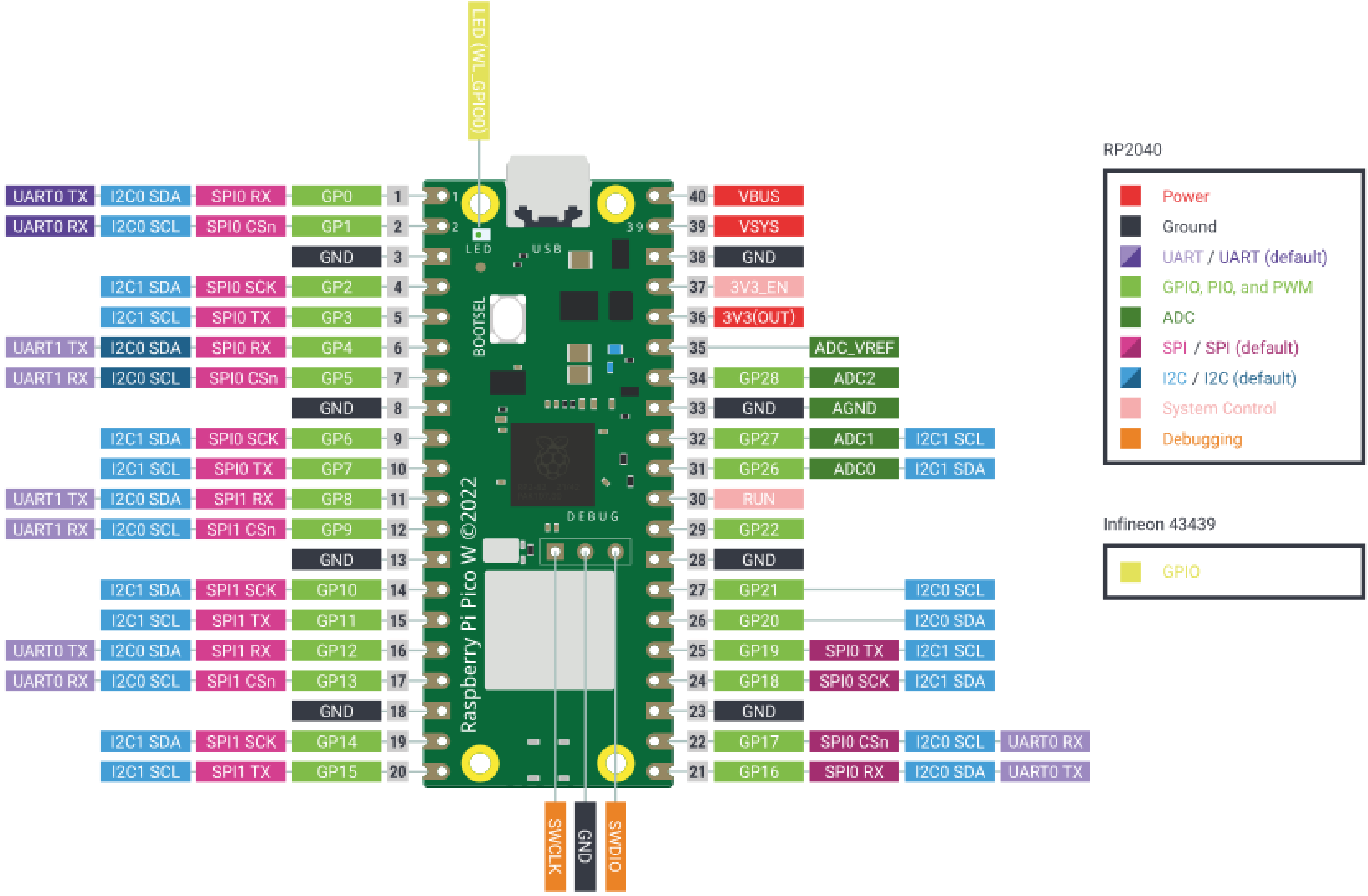
How to power the Raspberry Pi Pico

Powering the Raspberry Pi Pico

The easiest way to power a Pico is through the USB cable either to a computer or a phone charger. If you are going for a wireless design, you can use batteries as well. This tutorial will demonstrate how to power a Pico with a 9V battery or 3 AA batteries.

Raspberry Pi Pico pinout

You have to find out the pinout of the model you are using. The pins you are looking for are `VSYS` and `GND`. In this tutorial, a Pico W is used.



Raspberry Pi Pico's Power Need

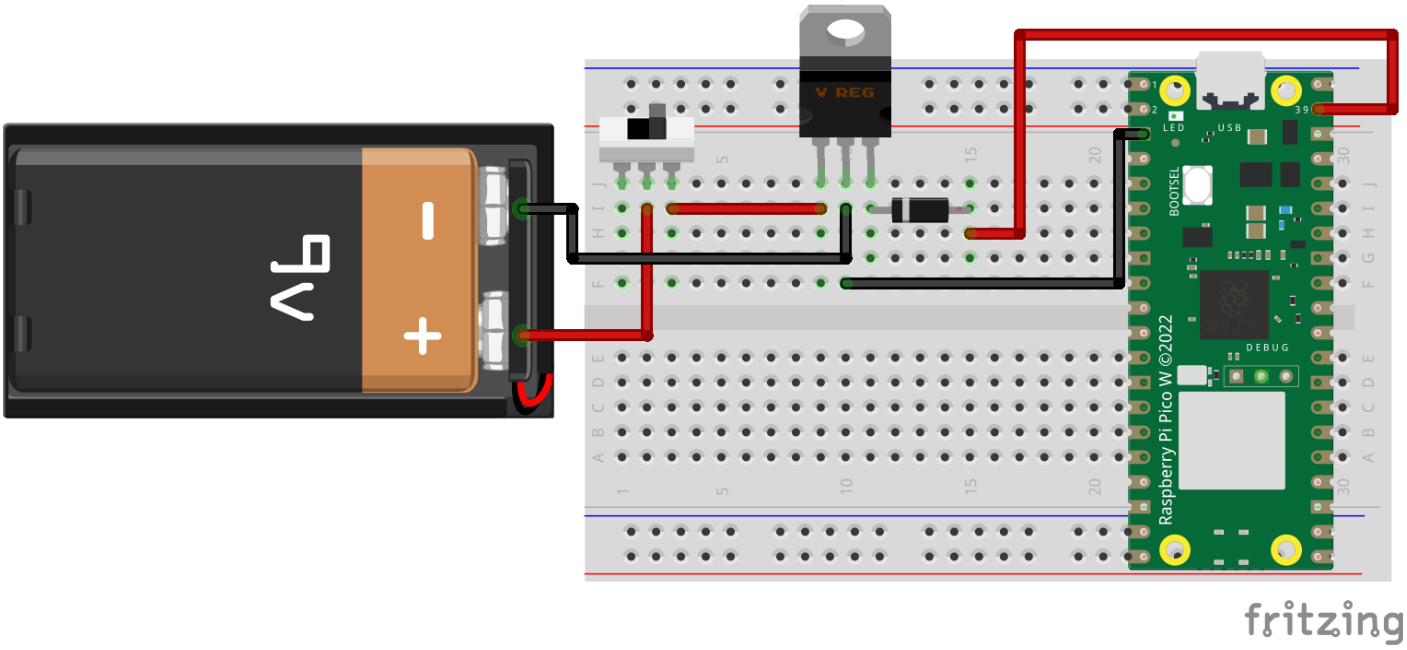
`VSYS` (PIN 39): Pin for main system input voltage. The input voltage can vary between **1.8V to 5.5V**. This voltage is used by the onboard SMPS to generate 3.3V to power the RP2040 microcontroller and GPIOs.

Why these parts are needed

- **Switch**: When we are using batteries to power the microcontroller, a switch is preferred. Some battery holder has a built-in switch which will be more convenient for you. With a switch, you can turn off the Pico when not in use and not drain the battery.
- **LM7805**: The LM7805 is a popular voltage regulator integrated circuit (IC). It is a positive voltage regulator that provides a stable and fixed output voltage of +5 volts. It is widely used in electronic circuits to regulate the voltage and ensure a consistent and reliable power supply. It can accept an input voltage in the range of 7V to 35V, depending on the specific model, so you will need this when you are using a power supply with a voltage higher than 5V.
- **Diode**: - A diode is a semiconductor device that allows current to flow in one direction while blocking it in the opposite direction. In other words, it acts as a one-way valve for electric current. We need a diode for safety to prevent one power source from back-feeding the other. Raspberry Pi can have two power sources at the same time and there is a risk of unwanted power flow between the two power sources. Schottky diode is preferred for its low voltage drop, but any others should work with a bit of voltage drop around 0.2-0.6V.

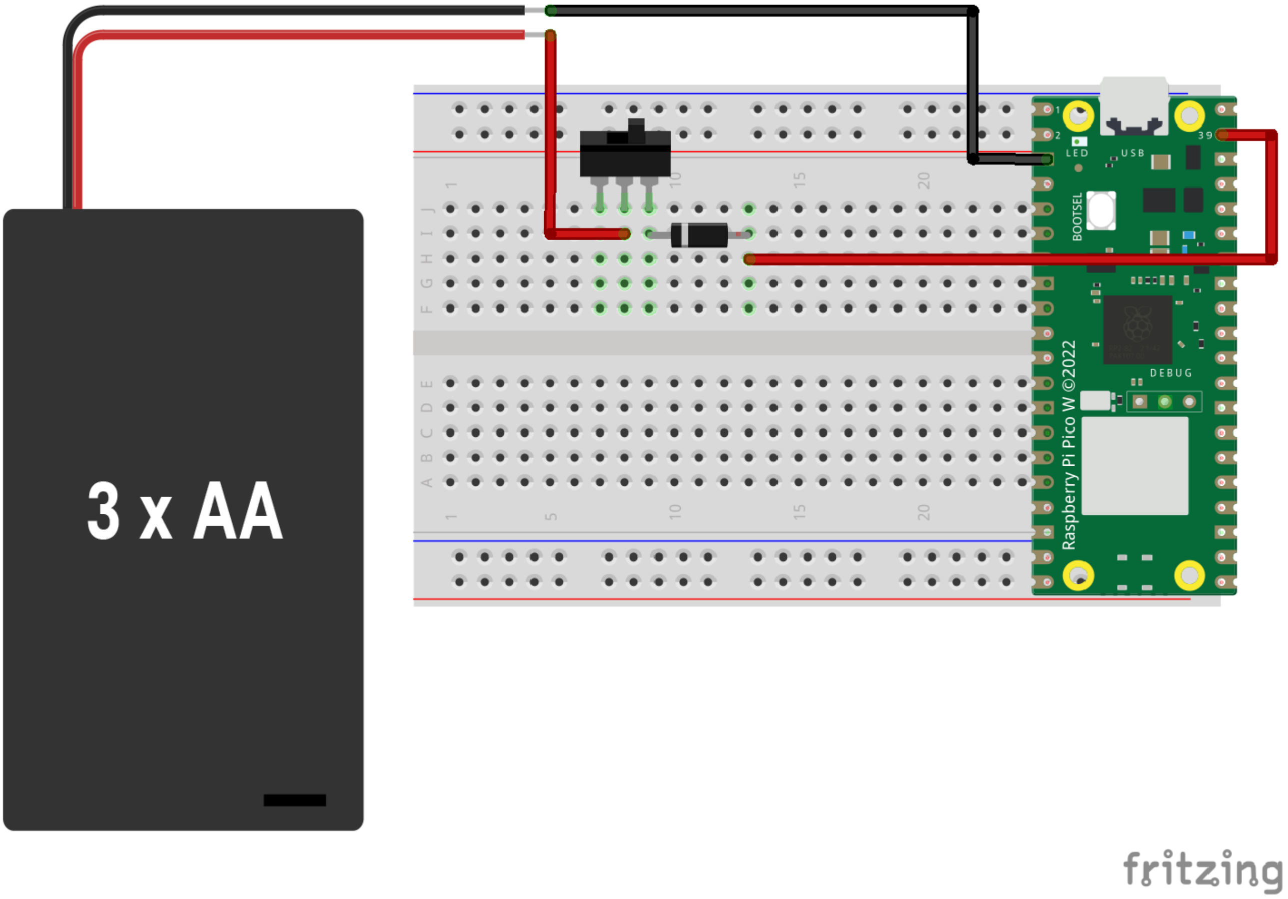
9V battery

Part needed: switch, LM7805 & Any diode



3 AA batteries

Part needed: switch & Any diode



How to make two Raspberry Pi Pico communicate via Wifi

Wifi Communication

Before heading to communication between two pico, you can try controlling pico via wifi with your computer first, see **tutorial here**.

Server Code

Client Code

How to control Raspberry Pi Pico via Wifi

What is Wifi

WiFi, short for Wireless Fidelity, is a technology that enables devices like smartphones, laptops, and other electronic gadgets to connect to the internet or communicate with each other without the need for physical cables. It operates by using radio frequency signals to transmit data between devices and a wireless router. The router, connected to the internet via a wired connection, acts as a central hub that facilitates communication between devices within its range. WiFi has become a ubiquitous and convenient way for people to access the internet and share information wirelessly, contributing to the proliferation of wireless connectivity in homes, businesses, and public spaces.

You can create a local wifi network with just powering up a wifi router. When two or more devices (computers, microcontrollers...) are connected to the same wifi network, they can communicate with each other, without accessing the internet. Not all microcontrollers come with a built-in Wifi module so we will be using Pico W (with a built-in wifi module) in this tutorial. But there are plenty of add-ons options you can add to your microcontroller for wifi.

External read for fun:)

What is IP address

An IP address, or Internet Protocol address, is a unique numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication. It consists of a series of numbers separated by periods, like **xxx.xxx.xxx.xxx**. Think of it as a digital address for your device on the internet. It allows devices to identify and communicate with each other on a network, much like a home address allows mail to be delivered to a specific location. IP addresses can be dynamic, meaning they change periodically, or static, where they remain constant. They play a crucial role in routing data packets across the internet, ensuring that information reaches the correct destination.

How to find my IP address

1. Google Search - "What's my IP address?"
2. Go to Network Preferences (MAC), Wifi Properties (Windows)
3. Go to **Terminal** (MAC) or **Command Prompt** (Windows) and type in `ipconfig getifaddr en1`

Get started

The example code will turn on/off the onboard LED on Pico via wifi without an external circuit. See **this tutorial** to learn how to set up your Pico. Before you go ahead, there are some things that you need to change for your wifi.

1. Change `ssid`, `password` in the code
2. Run your code in Thonny
3. Find your `IP address` in the Thonny Shell, something like `ip = 10.3.15.120`
4. open up a web browser
5. go to `http:// **IP address**/light/on` to turn the LED on
6. go to `http:// **IP address**/light/off` to turn the LED off

```
import network
import socket
import time

from machine import Pin

led = Pin("LED", Pin.OUT)

ssid = 'YOUR NETWORK NAME'
password = 'YOUR NETWORK PASSWORD'

wlan = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.connect(ssid, password)

html = """<!DOCTYPE html>
<html>
    <head> <title>Pico W</title> </head>
    <body> <h1>Pico W</h1>
        <p>%s</p>
    </body>
</html>
"""

max_wait = 10
while max_wait > 0:
    if wlan.status() < 0 or wlan.status() >= 3:
        break
    max_wait -= 1
    print('waiting for connection...')
    time.sleep(1)

if wlan.status() != 3:
    raise RuntimeError('network connection failed')
else:
    print('connected')
    status = wlan.ifconfig()
    print( 'ip = ' + status[0] ) //print your address

addr = socket.getaddrinfo('0.0.0.0', 80)[0][-1]

s = socket.socket()
s.bind(addr)
s.listen(1)

print('listening on', addr)

# Listen for connections
while True:
    try:
        cl, addr = s.accept()
        print('client connected from', addr)
        request = cl.recv(1024)
        print(request)

        request = str(request)
        led_on = request.find('/light/on')
        led_off = request.find('/light/off')
        print( 'led on = ' + str(led_on))
        print( 'led off = ' + str(led_off))

        if led_on == 6:
            print("led on")
            led.value(1)
            stateis = "LED is ON"

        if led_off == 6:
            print("led off")
            led.value(0)
            stateis = "LED is OFF"

        response = html % stateis

        cl.send('HTTP/1.0 200 OK\r\nContent-type: text/html\r\n\r\n')
        cl.send(response)
        cl.close()

    except OSError as e:
        cl.close()
        print('connection closed')
```

How to install Node.js on Raspberry Pi

Installing Node.js on Raspberry Pi is very simple for those with basic command line experience.

In the terminal or via SSH:

1. Add the package source: `curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -`
2. Install Node using: `sudo apt-get install -y nodejs`
3. Confirm package is installed: `node -v`

Extra steps

You may also wish to install the development tools to build native addons: `sudo apt-get install gcc g++ make`

And you may wish to install Yarn package manager to replace NPM:

```
curl -sL https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -
echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/sources.list.d/yarn.list
sudo apt-get update && sudo apt-get install yarn
```