PicoLamp

Łukasz Kolber

Introduction

Pico lamp is an LED lamp, using an addressable LED (ws2812b) the whole device is using a 5V power source. In default it uses 3 neopixel ring-s (24 LED, 16 LED, 12 LED) and the whole device needs at least 4A power source.

Lamp designation is to be table lamp and an ornament.

Lamp has 10 modes:

- Constant Color No Animation
- Constant Color
- Loading
- Spinning
- RGB-animation
- Fade
- Blink
- Instant Blink
- Wifi Quality - shows wifi quality as colors red yellow red base od RSSI value.

Device communicates via MQTT Broker.

Devices can use other numbers of led-s / ring-s but this is not tested, also requires changes in config.

Required Parts

Required parts:

- neopixel ring 24 LED (wsb2812b)
- neopixel ring 16 LED (wsb2812b)
- neopixel ring 12 LED (wsb2812b)
- DC Socket
- DC 5V/4A Power Source
- Wires
- Raspberry Pi Pico W
- 3D Printed Model
- Some bolts
- Articulating Arm

Configs

All configs should be in config folder in driver

Colors

Contains default colors for the device should contain at least 2 colors. Best way is to use default colors.

Strip

Here are field defining where ring-s are connected how many of them there is and how many ring-s device is using.
Strip contains 3 field:

- "size": Number - total number of led-s
- "pin": Number - pin number where led-s are connected
- "groups": [list] - group sizes

Secrets [Must Be Configured By User]

Filed are required to connect device to wifi (must be 2.4 GHz):

- "ssid" : string - wifi name (full name)
- "password" : string - password to wifi

If device cannot connect to wifi, run this code and check if you can connect to it (using Thonny IDE):

```
import network
def scan_wifi_networks():
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    networks = wlan.scan()
    wlan.active(False)
    return networks
wifi_networks = scan_wifi_networks()
for network in wifi_networks:
    ssid = network[0].decode('utf-8')
    rssi = network[3]
    print("SSID: {}, RSSI: {}".format(ssid, rssi))
```

Hivemq [Must Be Configured By User]

Required data to connect to MQTT broker - HiveMQ:

- "client_id" : string
- "server" : string
- "port" : number
- “user”: string
- “password” : string
- “keepalive”: int
- “ssl”: int
- “ssl_params_server_hostname” : string

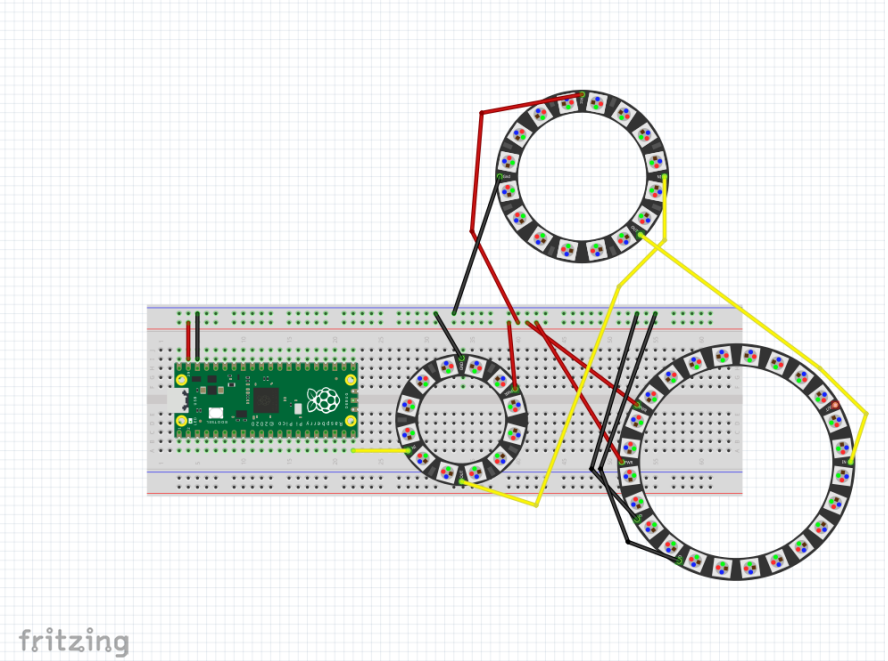
All data can be found in GettingStarted (the select python example).

Pilot

Not used in the current version required to start the device.

Wiring Example Schematic

Connection can be different, but it needs to be reconfigured in strip config. Default connection (Img 1.) requires no changes in strip config.



Img 1. Example connection schematic in Fritzing.

Starting

Configuration Steps

1. Plug in Raspberry Pi Pico to Computer.
2. Open Thonny IDE and transfer files to Raspberry Pi Pico.
3. Create and configure Configs.
4. Run Device in DEBUG (set it to true) Mode.
5. Test device by sending example json.
6. If no problems then change DEBUG to false.
7. Disconnect device from computer.
8. Connect Neopixels and put raspberry pi pico in a 3D model.
9. Power it up and use it.

Error Finding

Easiest way to find error-s is to connect raspberry pi pico via usb to the computer and run the program in Thonny in DEBUG mode.

Communication Via HiveMQ

Overview

You can communicate to the device by sending json objects to provided topics.
Example json objects can be found in the repository.

Topics

state

State contains a single field and switches led’s off (0) and on (1).

update_led

This topic contains single field-s - ‘states’ and indicates which LED rings should be on (1) or off (0).

update_mode

Provides information for actual mode and updates it, if in data transfer all information was provided. Each mode contains other fields that can be updated(check example_json).

ping

Ping indicates if the device is working and prints whole data provided in transfer.

new_mode

Provides information about the new mode to start (cannot start the same mode you should use update_mode). Each mode has an ‘extend’ field where you can provide additional data for the mode(leaving it empty means default data will be loaded).