

# Mini Tower Kit for Raspberry Pi 4B

Shanghai Kunlai Electronic Technology Co., Ltd.

## Description

Mini Tower Case is made of ABS material, with very high precision, and the 45-degree OLED screen display angle is more convenient to observe system information.

The screen holder adopts a quick-release design, and the interior of the shell includes a design for arranging cables, which is convenient for arranging cables.

The interface adopts with a fool-proof design to avoid problems caused by wrong wiring, the installation is simpler and more convenient.

The screen driver board contains RGB three-color programmable LED ambient light, which can better render the light color according to your own needs.

The ice tower radiator included in the kit can not only provide good heat dissipation, but also increase the appearance. A non-slip rubber strip is added at the bottom to prevent it from being more stable and firm on a smooth desktop.

## Features

- \* ABS Material and Acrylic
- \* Colorful mood lights
- \* Adjustable fan light
- \* 0.96-inch OLED Display (I2C protocol)
- \* OLED bracket adopts quick release design for easy installation and removal
- \* Precise positioning of holes
- \* Ice tower radiator strong heat dissipation
- \* Inside the housing contains a cable management rack
- \* Easy wiring and easy cable management

## Specifications:

- Edge expansion board: 40Pin
- Ice tower cooler:
  - \* Metal and Copper
- OLED Display:
  - \* Resolution: 128x64 pixel
  - \* Default register address: 0x3c
  - \* INPUT: 3.3V
- Mood Light:
  - \* WS281x programmable LED
  - \* INPUT: 3.3V~5V
- 4010Fan:
  - \* INPUT: 3.3~5.25V
  - \* Consuming: 0.06mA
  - \* Driven by PWM signal
  - \* Freq: 50Hz
  - \* Duty Cycle range: 0~255
- Power supply requirement:
  - \* INPUT: 5V@3A at least
- Case:
  - \* Materials: ABS plastic and Acrylic
  - \* Screws: M2.5



## 40Pin GPIO can be expanded by edge expansion board



Easy to access TF card



- OLED Display system information

NOTE: need to install driver and programming...



- By programming the OLED, the content that can be displayed is variable, such as displaying the Raspberry Pi's IP address, CPU temperature, disk utilization, CPU load, memory status, network card traffic and other system information.

- You can also change the displayed content through your own customization.

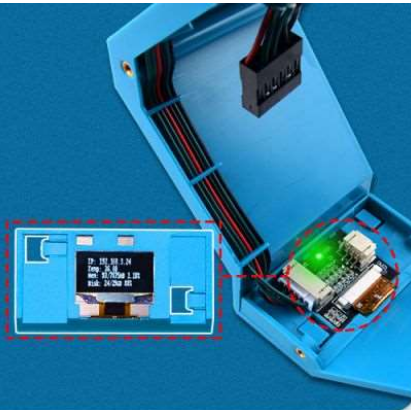
- Good Heat Dissipation with Ice Tower Cooler



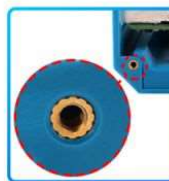
- ABS case with anti-slipper robber pad and flat heat screws.

Easy to tidy up wires.

- The screen holder adopts a quick release design
- The screen driver board contains RGB three-color programmable LED ambient light, which can better render the light color according to your own needs.



► A non-slip rubber strip is added at the bottom to prevent it from being more stable and firm on a smooth desktop.



► Precise positioning of holes



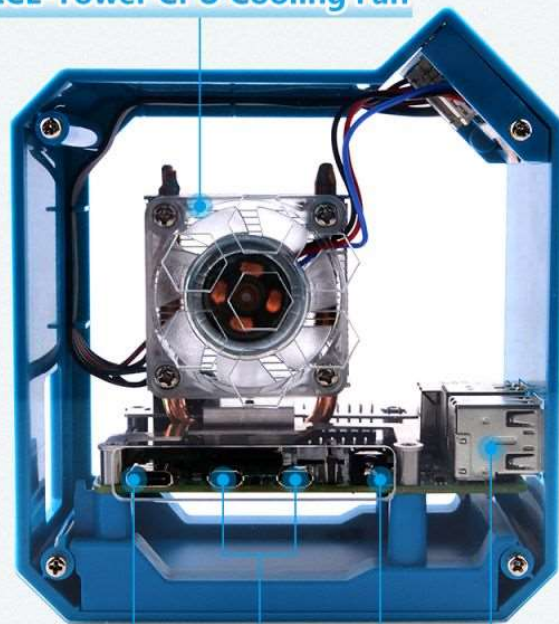
► The laser-cut acrylic cover on the side allows you to observe the running state of the Raspberry Pi.



► The interior of the shell includes a design for arranging cables, which is convenient for arranging cables.

## Front Side

### ICE-Tower CPU Cooling Fan



USB-C Power Supply Port

Micro HDMI Ports

Audio Port

**Raspberry Pi 4**  
(Not include motherboard)

How to assemble

## Installation Steps

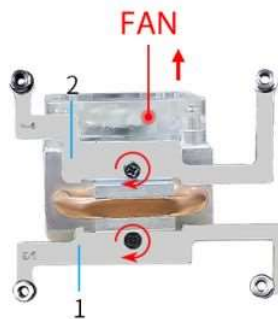


Install the thermal pad



For easy installation, please install the TF card on the motherboard first

1 2  
3 4

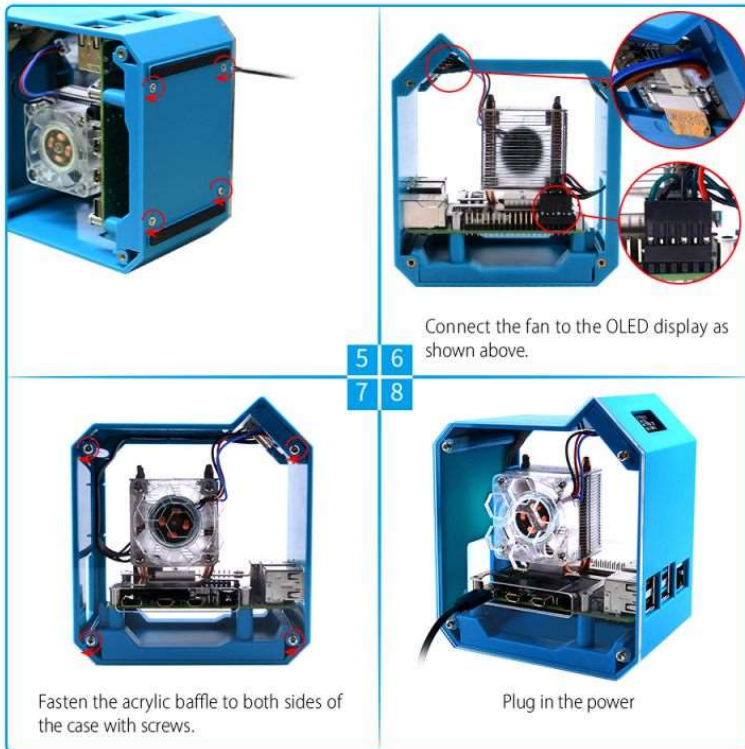


Fix the metal bracket of low profile ice tower cooler with M2.5 screws.



Please hold down the top of the hot end lightly and install the screws from the bottom to secure the Raspberry Pi and tower cooler.

## Installation Steps



5 6  
7 8

Connect the fan to the OLED display as shown above.

Fasten the acrylic baffle to both sides of the case with screws.

Plug in the power

## Package Includes

- 1 \* Mini Tower Kit for Raspberry Pi 4B.

NOTE: Raspberry Pi 4B mainboard is not included, Additional purchase required



## What you will get?



- ABS Mini Tower CASE
- ICE-Tower CPU Cooling Fan
- Acrylic covers
- 0.96 inch OLED Display (I2C protocol)
- Screen holder
- Assembly manual
- GPIO Edge Extension

## How to enable OLED Display?

- We assume you are using Raspberry Pi OS, (32bit/64bit).
1. Turn on `i2c function` by using `sudo raspi-config` -> `interface options` -> `i2c` -> `enable` -> `yes`. 2. Check if the screen has been recognized by Raspberry Pi

```
i2cdetect -y 1
```

if encounters `command not found` error, please install `i2c-tools` by using `sudo apt update && sudo apt -y install i2c-tools`.

3. Install dependencies libraries:

```
sudo apt -y install python3 python3-pip python3-pil libjpeg-dev  
zlib1g-dev libfreetype6-dev liblcms2-dev libopenjp2-7 libtiff5
```

4. Grant privileges to user `pi`

```
sudo usermod -a -G gpio,i2c pi
```

5. Download sample code from this repo:

```
git clone https://github.com/rm-hull/luma.examples.git
cd luma.examples/
```

6. Install the dependencies

```
sudo -H pip3 install -e .
```

7. Entering into example folder and test it.

```
cd examples/
python3 clock.py
```

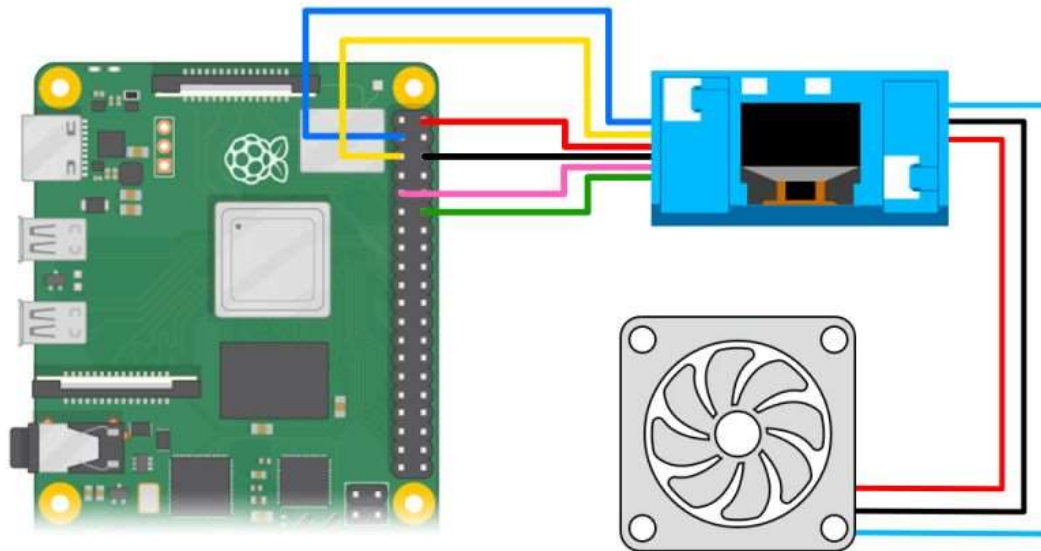
8. At this time, the OLED screen is displaying a clock.

If there is black screen or have nothing display on the screen, please check whether the cable is connected properly, and then check if you have enable the I2C function, and you can just typing: `i2cdetect -y 1` in a terminal and check if there is an address like "3C" on screen. if not, please reconnect the cable and reboot raspberry pi. If you cannot download the repository from github, please check the internet connection and please make sure you can access to github.com. If you have issue with using the OLED display, please contact us first.



# How to enable fan light and mood light?

## Connection Details



NOTE: The RGB lights in the fan are connected to the ambient lights on the screen driver board. Turn on any one of the lights is equivalent to turn on the entire light group.

- 1. The lights are connected to GPIO18 which can found by typing: pinout in a terminal.

J8:

3V3	( 1 )	( 2 )	5V
GPI02	( 3 )	( 4 )	5V
GPI03	( 5 )	( 6 )	GND
GPI04	( 7 )	( 8 )	GPI014
GND	( 9 )	( 10 )	GPI015
GPI017	( 11 )	( 12 )	GPI018
GPI027	( 13 )	( 14 )	GND
GPI022	( 15 )	( 16 )	GPI023
3V3	( 17 )	( 18 )	GPI024
GPI010	( 19 )	( 20 )	GND
GPI09	( 21 )	( 22 )	GPI025
GPI011	( 23 )	( 24 )	GPI08
GND	( 25 )	( 26 )	GPI07
GPI00	( 27 )	( 28 )	GPI01
GPI05	( 29 )	( 30 )	GND
GPI06	( 31 )	( 32 )	GPI012
GPI013	( 33 )	( 34 )	GND
GPI019	( 35 )	( 36 )	GPI016
GPI026	( 37 )	( 38 )	GPI020

- 2. Make sure your Raspberry Pi can access internet.
- 3. Download demo code projects sources from github.

```
cd ~  
git clone https://github.com/jgarff/rpi_ws281x
```

- 4. Build:

## Build with SCons

### Install Scons (on raspbian)

```
sudo apt update && sudo apt -y install scons
```

- Make sure to adjust the parameters in `main.c` to suit your hardware.
- Signal rate (400kHz to 800kHz). Default 800kHz.
- `ledstring.invert=1` if using an inverting level shifter.
- Width and height of LED matrix (height=1 for LED string).
- Type scones from inside the source directory.

```
cd rpi_ws281x/  
sudo scones
```

## Build and install with CMake

- Install CMake
- Configure your build:

For example:

```
mkdir build  
cd build  
cmake -D BUILD_SHARED=OFF -D BUILD_TEST=ON ..
```

See also for available options in [CMakeLists.txt](#).

Type

```
cmake --build .
```

to build

- To install built binaries and headers into your system type:

```
sudo make install
```

- Running:

Type

```
sudo ./test
```

(default uses PWM channel 0).

That's it. You should see a moving rainbow scroll across the display.

More options are available,

```
./test -h
```

should show them:

```
./test version 1.1.0
Usage: ./test
-h (--help)    - this information
-s (--strip)   - strip type - rgb, grb, gbr, rgbw
-x (--width)   - matrix width (default 8)
-y (--height)  - matrix height (default 8)
-d (--dma)     - dma channel to use (default 10)
-g (--gpio)    - GPIO to use
                If omitted, default is 18 (PWM0)
-i (--invert)  - invert pin output (pulse LOW)
-c (--clear)   - clear matrix on exit.
-v (--version) - version information
```

- Reference URL: [ [https://github.com/DeskPi-Team/rpi\\_ws281x](https://github.com/DeskPi-Team/rpi_ws281x) ]

## How to Install All Drivers Automatically

---

- Clone this Repository: [ <https://github.com/geekpi/absminitowerkit> ]

```
cd
git clone https://github.com/geekpi/absminitowerkit.git
cd absminitowerkit/
sudo ./install.sh
```

## How to change display information

If you want to change the display information, please follow the steps below.

- Open a terminal and navigate to `/lib/systemd/system/` folder.
- Edit 'minitower\_oled.service' file and adding your script.

```
pi@raspberrypi:/lib/systemd/system $ pwd
/lib/systemd/system
pi@raspberrypi:/lib/systemd/system $ sudo vim.tiny /lib/systemd/system/minitower_oled.servic
e █
```

And the demo codes are located at: `/usr/local/luma.examples/examples/`, or you can put your own code in the same location and change the parameter of `ExecStart` variable and

restart the service.

```
14:15:15 192.168.3.7 (pi)
[Unit]
Description=Minitower oled Service

[Service]
Type=forking
User=root
ExecStart=/bin/bash -c 'python3 /usr/local/luma.examples/examples/animated_gif.py &'
# ExecStart=/bin/bash -c 'python3 /usr/local/luma.examples/examples/sysinfo.py &'
Restart=always
RestartSec=30

[Install]
WantedBy=multi-user.target
~
```

For example, Default ExecStart's parameters is called /usr/local/luma.examples/examples/animated\_gif.py file, we can just comment it out with # (hash tag) and remove the hast tag before ExecStart=/bin/bash -c 'python3 /usr/local/luma.examples/examples/sysinfo.py &' and save it and quit.

```
[Unit]
Description=Minitower oled Service

[Service]
Type=forking
User=root
# ExecStart=/bin/bash -c 'python3 /usr/local/luma.examples/examples/animated_gif.py &'
ExecStart=/bin/bash -c 'python3 /usr/local/luma.examples/examples/sysinfo.py &'
Restart=always
RestartSec=30

[Install]
WantedBy=multi-user.target
~
```

you can also replace the file name with the file names in folder:  
/usr/local/luma.examples/examples/

- Reload systemd service and minitower\_oled.service

```
sudo systemctl daemon-reload
sudo systemctl restart minitower_oled.service
```

- You will find the content of OLED display has been changed.
- More demo code please access your local folder in:  
/usr/local/luma.examples/examples/ folder, there are plenty funny code inside the folder.

```
pi@raspberrypi:/usr/local/luma.examples/examples $ ls
3d_box.py          dotmatrixtool.py  picamera_photo.py  sys_histogram.py
animated_gif.py   etc               picamera_video.py  sys_info.py
bitstamp_realtime.py font_awesome.py  pi_logo.py         sysinfo.py
bitstamp_ticker.py game_of_life.py   proc              terminal.py
bounce.py         greyscale.py     __pycache__       tv_snow.py
carousel.py      image_composition.py root              tweet_scroll.py
chroma.py        images           runner.py         usr
clock.py         invaders.py      savepoint.py      var
colors.py        jetset_willy.py scrolling_pixelart.py video.py
crawl.py         laron_hue.py   sevensegment_demo.py weather.py
demo_opts.py     matrix.py       sprite_animation.py welcome.py
demo.py          maze.py         starfield.py      sys
dev             perfloop.py     sys
pi@raspberrypi:/usr/local/luma.examples/examples $ █
```

## YouTuber Feedback

---

- Notenoughtech.com
- <https://notenoughtech.com/raspberry-pi/turning-raspberrypi-4-into-a-mini-tower-pc/>

## Keywords

---

- Mini tower case, mini tower kit for Raspberry Pi 4B case, abs case for Raspberry Pi

## FAQ

---

- Why does my fan lights up when I was booting up my Raspberry Pi?

1. Please check if the cable is connected properly as wiki instructions assembling steps.
2. please check if you have installed the driver and make it running at booting time?
3. Could you please try to setup your lights by following this repo: <https://github.com/geekpi/absminitowerkit>

- How to check if the OLED display is dead or not configured well?

1. Please check the cable's connection, make sure the cable connected to raspberry pi's GPIO in right position and direction.
2. Please make sure you have enable I2C by using 'raspi-config' tool.
3. Please typing 'i2cdetect -y 1' in a terminal and check if there is '3c' mark in the address table.



4. Please download demo examples code from :<https://github.com/rm-hull/luma.examples.git> and set it up according to wiki instructions.