

A black and white photograph of a Raspberry Pi computer. The board is populated with two large cooling fans mounted on top. A keyboard is connected to the front of the board. Various components like the SD card slot, USB ports, and other surface components are visible. The text 'RGB Matrix Screen' is overlaid in large white letters.

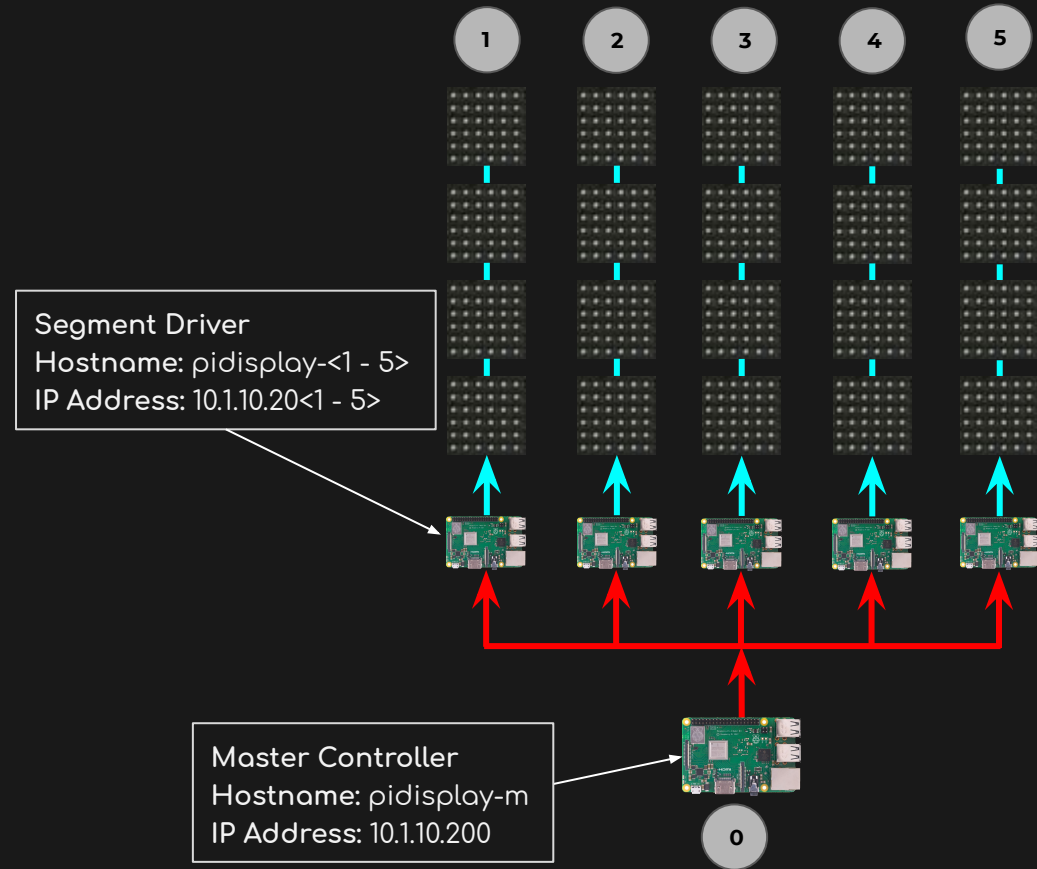
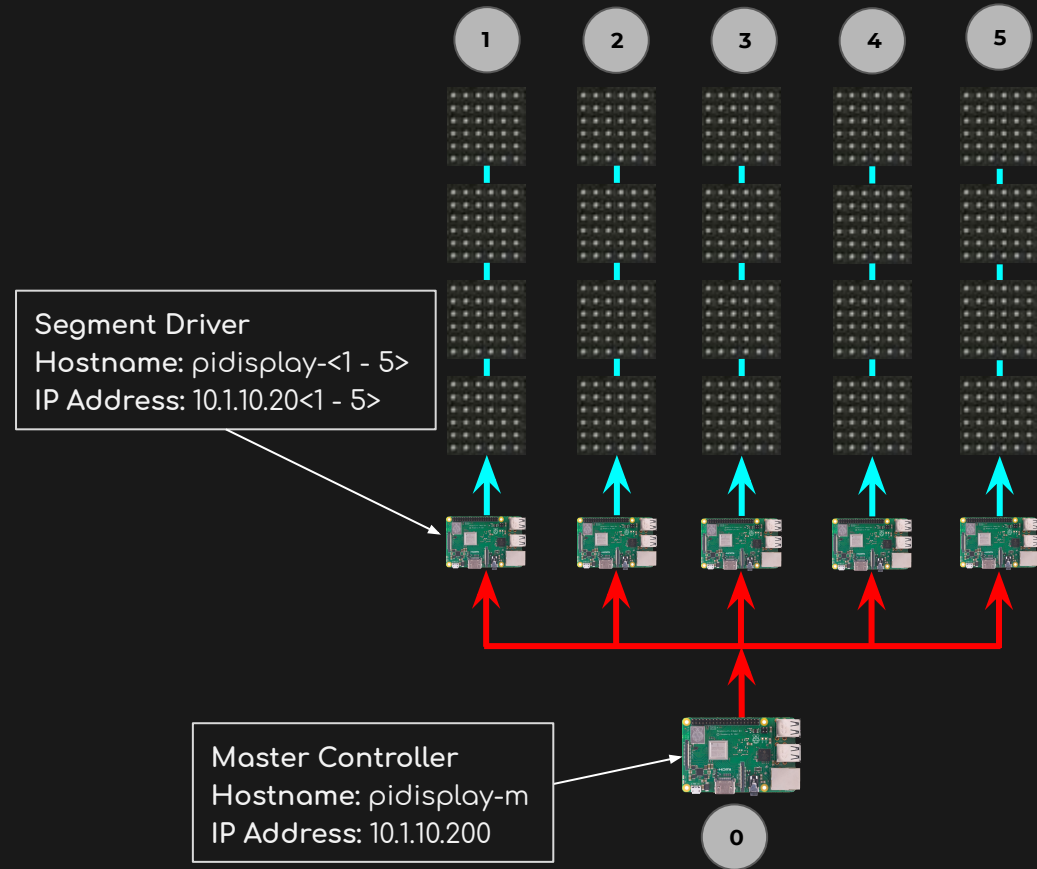
RGB Matrix Screen

Raspberry Pi Setup
[Software from base image]

Overview

When complete, this procedure will help prepare the 6 Raspberry Pis for assembly.

We use a numeric scheme that helps organize our devices & communication topology.



Overall Steps

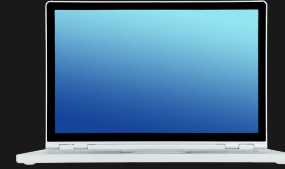
1. Gather Parts
2. Solder Daughterboard
3. Insert Pis into bottom half of Case
4. Assemble 5x Display controller Pis
5. Assemble 1x Driver Pi

Note: this process takes ~60 minutes for each Pi.

Required tools & Materials



Fully assembled Raspberry Pi 3B+



Mac or Linux Laptop & SD Card Reader / Writer



Anker PowerPort



USB Keyboard And Mouse



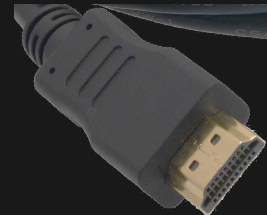
2.4Amp Micro USB Cables



Micro SD Card & Adapter



Network Switch & Cat 5 cables

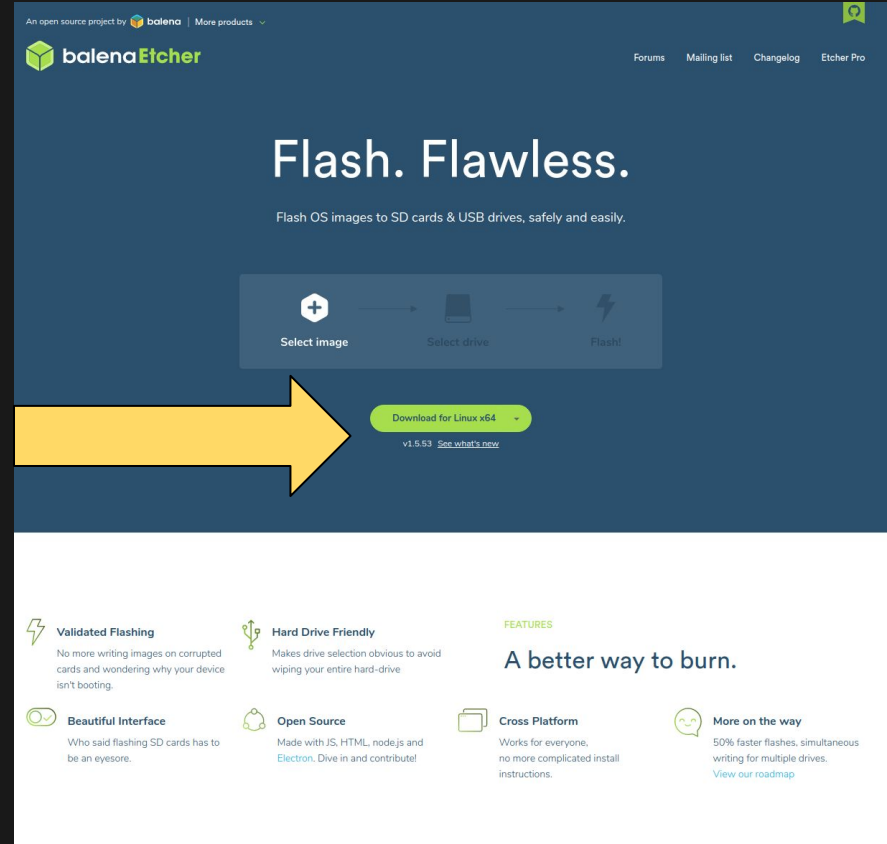


External display with HDMI and HDMI Cable

Download & Install Balena Etcher

On your Mac or Linux machine, download and install Balena Etcher

<https://www.balena.io/etcher/>



Download & Extract Diet Pi

On your Mac or Linux machine, download and extract Diet Pi Linux distro for the Raspberry Pi.

You'll need 7zip installed to extract the image.

MacOS: Use UnArchiver: <https://apps.apple.com/us/app/the-unarchiver/id425424353?mt=12>

https://dietpi.com/downloads/images/DietPi_RPi-ARMv6-Buster.7z



CPU Performance Rating:	
Filesystem Performance Rating (Micro SD):	
Network Performance (actual bandwidth, 1Gbit scale):	
Heat Efficiency Rating:	

Raspberry Pi

CPU:	Various
RAM:	Various

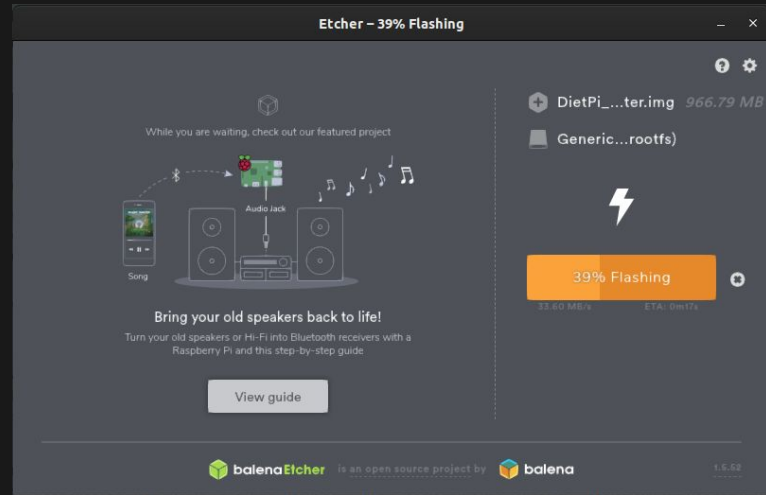
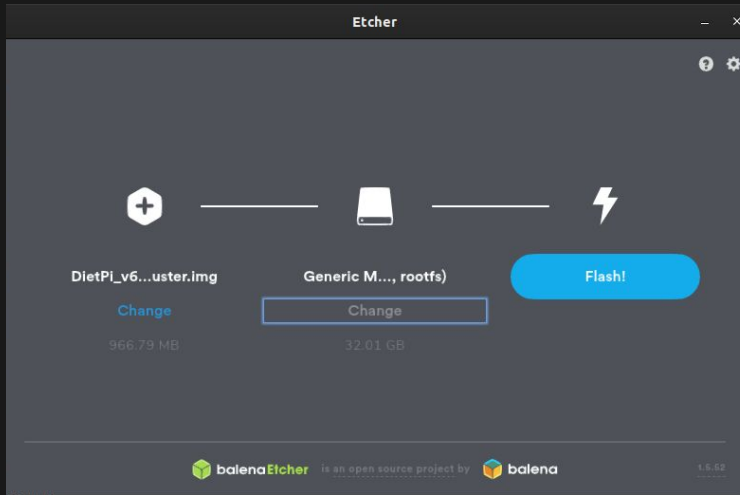
Download Image:  [Download](#)

Pros:	Large community, highly affordable, accessible
Cons:	CPU/Network performance, heat

The Raspberry Pi is quite possibly the single SBC that started the IoT era. While the Raspberry Pi may be a few years behind other SBC manufacturers in terms of performance, its widely available, affordable, has a large community and will always have a place in our hearts.

Download & Extract Diet Pi

Insert SD Card into your laptop & burn the image using balena Etcher. Perform this step for each of the 6 Micro SD Cards.



Preparing your workstation

1. Set up an area where you can work on multiple work streams simultaneously.
2. Grab a power strip, you'll need it. The following items will be powered by it.
3. Place the network switch in your work environment & connect the office LAN and your laptop to the switch.
4. Add the the Anker Powerport to your workstation. This will supply power to the Pis.
5. Place the HDMI display, keyboard and mouse on your workstation.

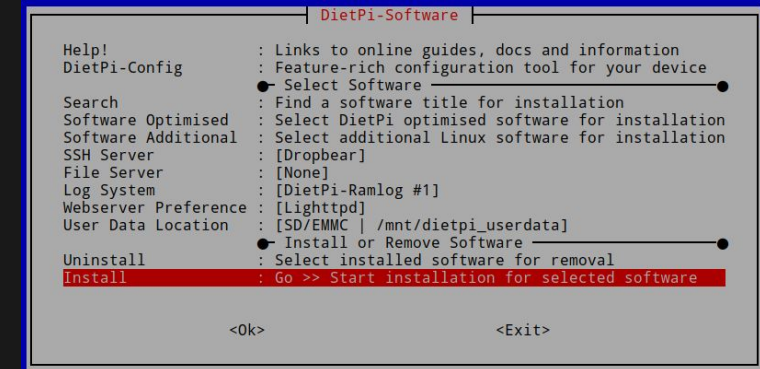
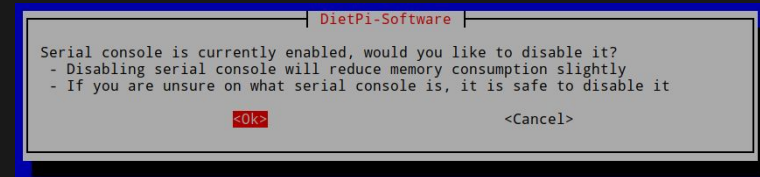
Setting up Pi workflow

The following workflow should be performed for each Pi

1. Gently insert a freshly burned SD Card
2. Attach Pi to Network, Keyboard, Mouse, Display
3. Apply power to the Pi to boot.
4. Update distribution
5. Install required libs & tools
6. Download & compile LibBoost and CMake
7. Install our library & configure for auto-execution
8. Configure the Pi for assembly (Hostname, IP Address, etc...)

Provisioning the Pi base software

1. Gently insert an SD Card, connect peripheral devices & apply power.
The pi will boot, resize it's filesystem and reboot again.
2. Log in with the default credentials
 - User: root
 - Pass: dietpi
3. Accept the license agreement (ENTER and TAB + ENTER)
4. Change the default passwords when prompted for both accounts to ********* (do this twice)
5. Choose **<OK>** to disable serial console
6. Navigate to **Install** and press ENTER & select **<OK>** to only do a pure minimal image".
The pi will install some stuff.
7. Choose **<Opt Out>** when prompted. The Pi will land in a prompt.



Update locale and keyboard layout

dietpi-config

5:Language/Regional Options

Locale: en_US.UTF-8

OK to Reboot

```
LC_ALL='en_US.UTF-8' dpkg-reconfigure keyboard-configuration
Select Generic 101-key PC
Select 'Other'
Select 'English (US)'
Select 'English (US)'
Select 'The default for the keyboard layout'
Select 'No compose key'
reboot
```

Install Boost & CMake (on servers only)

Install core libraries, some utilities and our RGB Matrix library by issuing command:

```
curl https://raw.githubusercontent.com/ModusCreateOrg/network-rgb-matrix-display/master/bin/install.sh | bash
```

This will take about an hour to complete the process. Some steps may seem to pause (curl output). As long as you see the green Storage I/O LED flashing, the Pi is working.

Safely unplug the Keyboard, mouse & display and move on to the next Pi to setup base software.

Do this for all of the Raspberry Pis.

After every Pi has entered this step, we can move on.

Configure Regional info

We'll need to configure a few things to allow our devices to communicate. Substitute **N** for the number of the Pi you are provisioning. *****After these steps, the Pi will no longer be able to access the internet.**


1. Run `dietpi-config`
2. Select `Advanced Options`, then `Time Sync Mode` and change to `0 : Custom`
3. Select `Security-Options`, then `Change Hostname`
4. Type `pidisplay-N` and press ENTER
5. Press ENTER to `<CANCEL>` to abort the reboot request.
6. Select `<BACK>` to return to the menu root
7. Choose `Network Options: Misc`
8. Toggle `Boot Net Wait` to `[Off]`
9. Choose `Network Options: Adapters`
10. Toggle `IPV6` to `[OFF]`
11. Select `Ethernet` then change the following attributes
 - o `Change Mode` to `[STATIC]`
 - o `Static Ip` to `10.1.10.20N`
 - o `Static Gateway` to `10.1.10.1`
12. Select `Apply | Save all changes`
13. Select `<OK>` to reboot system.

Disable Core 4 on Matrix controllers only !!!

For the Matrix controllers to work as effectively as possible, we'll need to tell Linux not to use Core 4 for it's operating system.

We do this by adding `isolcpus=3` to `/boot/cmdline.txt`.

Log into the Pi and make the change via VI or Nano and reboot or shut the system down.



```
1 dwc_otg.lpm_enable=0 console=serial0,115200 console=tty1
  root=PARTUUID=cb7b86f7-02 rootfstype=ext4 elevator=deadline fsck.repair=yes
  isolcpus=3 rootwait
```

2



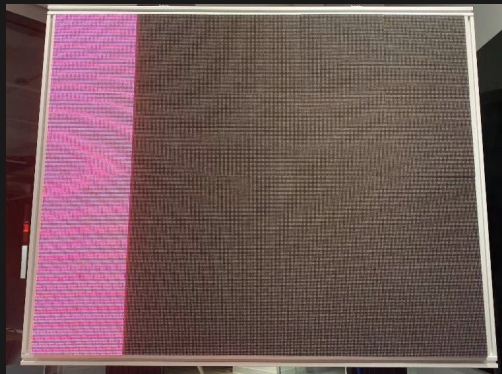
Test Matrix Controllers

We need to test the matrix controllers and to do this, we'll need to clone the repository, compile & execute. Execute the following commands from /root.

```
cd network-rgb-matrix-display/  
make -j 3
```

```
cd examples-api-use
```

```
./demo -t 4 -D 4 --led-rows=64 --led-cols=64 --led-slowdown-gpio=4 --led-chain=4
```



Setup master Pi

The Master Pi is what runs our Video game and we have to install the code to drive the matrix strips. Connect Audio amplifier and controller USB dongle to the Master Pi and issue the following commands from /root

```
git clone https://github.com/ModusCreateOrg/genus-matrix-display.git
./install.sh
./make-genus-build.sh
cd build/
./Genus
```

```
[pidisplay-0:~# ./Genus
RemoteMatrixSegment 10.1.10.201 (id = 0)
RemoteMatrixSegment 10.1.10.202 (id = 1)
RemoteMatrixSegment 10.1.10.203 (id = 2)
RemoteMatrixSegment 10.1.10.204 (id = 3)
RemoteMatrixSegment 10.1.10.205 (id = 4)
```

