

---

# Hide sidebar nav and TOC on the home page

---

Welcome to the **Cinemate** documentation.

*Edit `docs/index.md` and hit save – the site updates automatically when running `mkdocs serve`.*

 July 7, 2025



# Overview

## Overview

CineMate scripts is a way for users to implement and customize manual controls for their [cinepi-raw](#) build.

Project aims at offering an easy way to build a custom camera. For basic operation and experimentation, Raspberry Pi, camera board and monitor is needed. For practical use, buttons and switches can easily be added.

A ready made disk image can be found [here](#).

Join the CinePi Discord [here](#).

## Hardware requirements

- Raspberry Pi 4/5 or CM4 module
- Official HQ or GS camera
- HDMI monitor or device (phone or tablet) for monitoring

*For recording, use a high speed NVME drive or [CFE Hat](#) by Will Whang. Drive needs to be formatted as ext4 and named "RAW".*

*CineMate is also compatible with [OneInchEye](#) (Sony IMX 283) and [StarlightEye](#) (Sony IMX 585) by Will Whang. Works with CM4 module and Pi 5B.*

## Quickstart guide

- 1) Burn image to ssd card. 16 GB or larger.
- 2) Connect Pi and camera sensor board

**! When connecting the camera module to the Pi, make sure it is the Pi is not powered. It is not advised to hot-swap the camera cable.**

- 3) Boot up the Pi. CineMate should autostart.



## 7.3 CineMate “Pseudo-CLI”

CineMate doesn’t use a real command-line parser or shell. Instead, it implements a “pseudo-CLI” inside the running Python process to let you type or send simple, human-readable commands over an SSH session and USB/serial. Here’s how it works:

### Available Commands

Command	Argument(s)	What it does
<code>rec / stop</code>	<i>none</i>	Toggle recording on/off
<code>set iso &lt;100–3200&gt;</code>	<code>int</code>	Set ISO (clamps to nearest valid step)
<code>inc iso / dec iso</code>	<i>none</i>	Step ISO up/down
<code>set shutter a &lt;float&gt;</code>	<code>float</code>	Set shutter angle (snaps unless in free/sync mode)
<code>inc shutter a / dec shutter a</code>	<i>none</i>	Step shutter angle through dynamic list
<code>set shutter a nom &lt;float&gt;</code>	<code>float</code>	Set nominal shutter (motion-blur target)
<code>inc shutter a nom / dec shutter a nom</code>	<i>none</i>	Step nominal shutter angle
<code>set fps &lt;float&gt;</code>	<code>float</code>	Set frames-per-second (snaps or free)
<code>inc fps / dec fps</code>	<i>none</i>	Step FPS up/down
<code>set wb [&lt;Kelvin&gt;]</code>	<code>int</code> or <i>none</i>	Set or cycle white balance
<code>inc wb / dec wb</code>	<i>none</i>	Cycle WB up/down
<code>set resolution [&lt;mode&gt;]</code>	<code>int</code> or <i>none</i>	


## How “inverse” (1-0-1) buttons are auto-detected

Many latching push-buttons are wired closed = logic 1 at rest and open = 0 when pressed. At start-up each SmartButton performs:

```
if self.button.is_pressed:      # high at rest → treat as “inverse”  
    self.inverse = True
```

If so, the framework swaps the handlers:

```
when_pressed ← on_release  
when_released ← on_press
```

 July 7, 2025



# settings.json Cheat Sheet

A quick-reference table of every setting in `settings.json`, what it does, and its allowed values.

```
{
  "pin": 27,
  "state_on_action": {"method": "set_all_lock", "args": [1]},
  "state_off_action": {"method": "set_all_lock", "args": [0]}
},
```

```
if self.button.is_pressed:      # high at rest → treat as “inverse”
    self.inverse = True
```

## Geometry

JSON Path	Description	Values
<code>geometry.camX.rotate_180</code>	Rotate image 180° on startup	<code>true</code> / <code>false</code>
<code>geometry.camX.horizontal_flip</code>	Flip image horizontally on startup	<code>true</code> / <code>false</code>
<code>geometry.camX.vertical_flip</code>	Flip image vertically on startup	<code>true</code> / <code>false</code>

## Output

JSON Path	Description	Values
<code>output.camX.hdmi_port</code>	Select DRM connector for HDMI output ( <code>cinepi-raw --hdmi-port</code> )	<code>0</code> , <code>1</code> , or <code>-1</code> (auto)