

# InfoCube / ArtCube Project Manual

Globe, InfoCube, SolarCube, GalaxyCube, MatrixCube, BannerCube, Life, FaceID, orientation utilities, service automation, and build notes.

Manual version	2.0 draft
Source basis	Project zip uploaded May 9, 2026
Hardware target	Raspberry Pi with 6 x 64x64 RGB LED panels
Matrix library	rpi-rgb-led-matrix
Validated face order	0 TOP, 1 RIGHT, 2 REAR, 3 LEFT, 4 FRONT, 5 BOTTOM

Security note: the supplied project includes a private Google Calendar ICS URL in config/infocube.conf. This manual intentionally redacts that token and shows only the configuration pattern.

# Table of Contents

1. Project overview
  2. Hardware and runtime assumptions
  3. Repository layout
  4. Build process
  5. Common LED arguments
  6. Cube face orientation
  7. Application guide
  8. InfoCube data services
  9. Service automation and playlist
  10. Tuning notes
  11. Blog/GitHub release checklist
- Appendix A. Command reference
- Appendix B. Source files included

# 1. Project Overview

This project turns a six-panel RGB LED matrix cube into a small rendering platform. The project is not just one display program. It is a collection of renderers that share the same physical cube mapping and, increasingly, the same orientation correction layer.

The globe renderer became the visual baseline. Once Earth, the Moon, and the planet maps wrapped correctly across the six panels, the same face order and orientation assumptions could be used by InfoCube, SolarCube, GalaxyCube, MatrixCube, BannerCube, Life, and FaceID.

## 2. Hardware and Runtime Assumptions

Item	Current project assumption
Panel geometry	64 rows x 64 columns per panel
Panel count	6 chained panels
Logical strip width	384 x 64
GPIO mapping	adafruit-hat
Typical slowdown	6 for general use, 4 for improved photo/video refresh in some tests
Typical brightness	60 to 75 depending on exhibit conditions
Privileges	Most rpi-rgb-led-matrix runs require sudo/root for GPIO/PWM access

## 3. Repository Layout

The zip contains source, build output, maps, small static images, service files, and older backups. For publication, the source files and assets matter; the compiled binaries and object files can usually be omitted from a GitHub release.

Path	Purpose
*.cc / *.h	Primary application and support source files
Makefile	Production build rules for all applications
maps/	Planetary texture maps used by globe
images/	Static banner/top/bottom images for BannerCube
config/infocube.conf	Runtime config. Contains CAL_ICS_URL and should be redacted before publishing
cube-cycle.sh	Main service playlist and app rotation script
cube-cycle.service	Current systemd unit for root-run cube cycle
systemd/infocube-cycle.service	Older or alternate systemd unit
backup/	Older copies and development snapshots
bin/ and build/	Compiled output. Useful locally, but normally excluded from source release

## 4. Build Process

The project uses a production Makefile with C++17, release and debug modes, version stamping, and per-app linker requirements. The source assumes the rpi-rgb-led-matrix project is located one directory above this project directory.

```

# Build all applications
make

# Build debug versions
make MODE=debug

# Build one application
make globe
make infocube
make bannercube

# Clean generated objects and binaries
make clean

```

Application	Source objects	Extra libraries
infocube	infocube.cc, cube_orientation.cc, weather.cc, calendar.cc	curl
globe	globe.cc	jpeg
life	life.cc	none beyond rgbmatrix common libs
solarcube	solarcube.cc, cube_orientation.cc	none beyond rgbmatrix common libs
galaxycube	galaxycube.cc, cube_orientation.cc	none beyond rgbmatrix common libs
matrixcube	matrixcube.cc, cube_orientation.cc	none beyond rgbmatrix common libs
bannercube	bannercube.cc, cube_orientation.cc	jpeg
faceid	face_id.cc, cube_orientation.cc	none beyond rgbmatrix common libs

## 5. Common LED Arguments

Most applications accept the standard rpi-rgb-led-matrix command-line flags. The cycle script centralizes these into LED\_ARGS so the shared hardware settings stay consistent.

```

--led-rows=64
--led-cols=64
--led-chain=6
--led-gpio-mapping=adafruit-hat
--led-pwm-dither-bits=1
--led-pwm-lsb-nanoseconds=130
--led-pwm-bits=10
--led-slowdown-gpio=6
--led-brightness=75

```

If the Pi audio module conflicts with PWM, the script notes that --led-no-hardware-pulse can be added as a workaround.

## 6. Cube Face Orientation

Face ID	Physical face	3D meaning in Solar/Galaxy comments
0	TOP	+Y
1	RIGHT	+X
2	REAR	-Z
3	LEFT	-X
4	FRONT	+Z
5	BOTTOM	-Y

The shared cube\_orientation module applies per-face rotation correction for a cube laid out as one 384 x 64 strip. The current default correction array is based on the observed FaceID test: face 0 compensates 90 degrees, face 1 compensates 270 degrees, face 2 compensates 0, face 3 compensates 90, face 4 compensates 0, and face 5 compensates 270.

BannerCube intentionally uses a side-face strip order of LEFT -> FRONT -> RIGHT -> REAR, which corresponds to faces 3 -> 4 -> 1 -> 2. That creates a continuous 256 x 64 belt around the sides.

# 7. Application Guide

## Globe

**Source:** globe.cc. UV maps JPEG planetary textures onto the cube. It supports normal flat-face orientation and a pointy orientation where a cube vertex is used as the pole.

```
sudo ./bin/globe -i maps/earth.jpg --led-rows=64 --led-cols=64 --led-chain=6  
--led-gpio-mapping=adafruit-hat --led-slowdown-gpio=6 -s 60 -v 2
```

Option	Meaning
-i <file>	JPEG texture map. Defaults to Earth if omitted
-v or -v <0-7>	Pointy orientation. Optional vertex chooses top pole; default is 3 if -v is supplied without a number
-s <value>	Spin speed used by the Adafruit-derived globe code

## InfoCube

**Source:** infocube.cc. Rotating information display with clock, weather, forecast, calendar, status, and system diagnostics.

```
sudo ./bin/infocube --led-rows=64 --led-cols=64 --led-chain=6  
--led-gpio-mapping=adafruit-hat --led-slowdown-gpio=6 -s 60
```

Face/card	Description
CLOCK	Time-oriented face
WEATHER	Current weather from Open-Meteo
FORECAST	Today/tomorrow forecast and precipitation probability
CALENDAR	Next event from a Google Calendar ICS feed
STATUS	CPU, temperature, memory, network, host/IP, and health flags
DIAG	Version/build/status information

## MatrixCube

**Source:** matrixcube.cc. Cinematic digital rain mapped across the cube surface. Drops spawn near the top center, spill across side faces, and converge toward the bottom center.

```
sudo ./bin/matrixcube --led-rows=64 --led-cols=64 --led-chain=6  
--led-gpio-mapping=adafruit-hat --led-slowdown-gpio=6 -s 60
```

Tuning area	Notes
Drop speed	Current source uses $d.speed = 0.45f + dist01(rng) * 0.85f$
Motion model	Per-drop accumulated speed with sub-steps
Hardware pulse	Use --led-no-hardware-pulse if Pi audio conflicts with PWM

## SolarCube

**Source:** solarcube.cc. Animated stars, sun, planets, orbit trails, and Saturn ring projected onto the cube surface.

```
sudo ./bin/solarcube --led-rows=64 --led-cols=64 --led-chain=6
--led-gpio-mapping=adafruit-hat --led-slowdown-gpio=7 -s 60 --yaw=270
```

Option	Meaning
--yaw=<deg>	Rotates scene around +Y. 0, 90, 180, and 270 are most useful
--reverse	Reverses orbit direction

## GalaxyCube

**Source:** galaxycube.cc. Rotating spiral galaxy and starfield projected across the six faces.

```
sudo ./bin/galaxycube --led-rows=64 --led-cols=64 --led-chain=6
--led-gpio-mapping=adafruit-hat --led-slowdown-gpio=7 -s 60 --yaw=90
--brightness-scale=0.6
```

Option	Meaning
--yaw=<deg>	Rotates scene around +Y
--brightness-scale=<0..1>	Scales rendered RGB values; clamped from 0.05 to 1.0

## BannerCube

**Source:** bannercube.cc. Scrolling text around the side faces with static images on the top and bottom faces.

```
sudo ./bin/bannercube --led-rows=64 --led-cols=64 --led-chain=6
--led-gpio-mapping=adafruit-hat --led-slowdown-gpio=6 --text "MusicMaster Rocks"
--top-image images/jp_cube_premium_dual_top.jpg --bottom-image
images/jp_cube_premium_dual_bottom_rot180.jpg --speed 15 --brightness-scale 0.7
--text-color 5,52,243 --bg-color 221,224,237
```

Option	Meaning
--image <file>	Use same JPEG on top and bottom
--top-image <file>	Top face JPEG
--bottom-image <file>	Bottom face JPEG
--text <message>	Scrolling banner text
--speed <N>	Pixels per second, default 24
--brightness-scale <X>	0.05 to 1.0, default 1.0
--text-color R,G,B	Default 255,180,40
--bg-color R,G,B	Default 0,0,0

## Life

**Source:** life.cc. Game of Life style cube animation derived from the Adafruit example.

```
sudo ./bin/life --led-rows=64 --led-cols=64 --led-chain=6 --led-gpio-mapping=adafruit-hat
--led-slowdown-gpio=6 -s 60
```

Option	Meaning
-k	Color map / palette option from Adafruit-derived code
-t	Runtime option from Adafruit-derived code
-f	Fade timing option from Adafruit-derived code

## FaceID

**Source:** face\_id.cc. Diagnostic renderer. Draws face colors, large face numbers, and world-axis arrows so physical mapping can be verified.

```
sudo ./bin/faceid --led-rows=64 --led-cols=64 --led-chain=6 --led-gpio-mapping=adafruit-hat
--led-slowdown-gpio=6 -s 60
```

Use	Why it matters
Initial wiring check	Confirms panel chain order
Orientation debugging	Confirms which faces need rotation correction
Regression test	Run after refactoring cube_orientation or moving panels

## 8. InfoCube Data Services

InfoCube combines local system status with network data. Weather comes from Open-Meteo. Calendar data comes from an ICS feed, typically Google Calendar secret iCal format. Both services use libcurl with short connection and total timeouts so the display does not hang indefinitely.

Service	Source file	Behavior
Weather	weather.cc / weather.h	Fetches current and daily forecast data from Open-Meteo. Supports Fahrenheit. Parses current temperature, high/low, today/tomorrow codes, precipitation probability, and hourly arrays.
Calendar	calendar.cc / calendar.h	Fetches an ICS feed and selects the next relevant event. Stores title, start, end, all-day flag, formatted when text, and fetched timestamp.
Config	infocube.cc	Reads CAL_ICS_URL from /etc/infocube/infocube.conf/infocube.conf or from environment variable CAL_ICS_URL.
Cache	infocube.cc	Uses weather_cache.json and calendar_cache.json paths under /etc/infocube/infocube.conf in the current source. This path works only if the directory exists and is writable.

Recommended publish-safe config example:

```
# /etc/infocube/infocube.conf/infocube.conf
# Do not publish your private Google Calendar token.
CAL_ICS_URL=https://calendar.google.com/calendar/ical/YOUR_PRIVATE_CALENDAR_ID/basic.ics
```

Current source location defaults:

Setting	Value in source
Weather latitude	30.5299299
Weather longitude	-98.145384
Temperature unit	Fahrenheit
Weather hours requested	6
Card rotation	15 seconds
Weather stale threshold	20 minutes
Calendar stale threshold	20 minutes
Fonts	7x13.bdf, 6x10.bdf, 5x8.bdf under /home/jerry/rpi-rgb-led-matrix/fonts/

## 9. Service Automation and Playlist

The active service file is cube-cycle.service. It runs as root, starts after network-online.target, uses /home/jerry/InfoCube as the working directory, launches /home/jerry/InfoCube/cube-cycle.sh, and restarts automatically.

```
sudo cp cube-cycle.service /etc/systemd/system/cube-cycle.service
sudo systemctl daemon-reload
sudo systemctl enable cube-cycle.service
sudo systemctl start cube-cycle.service
sudo systemctl stop cube-cycle.service
sudo journalctl -u cube-cycle.service -f
```

The cycle script uses a run\_for helper that starts a child process, lets it run for a specified duration, then sends SIGINT followed by a cleanup kill if needed. This is important because the matrix should be cleared cleanly when each renderer exits.

Playlist item	Duration in script
Sun, Mercury, Venus, Earth, Earth2, Moon, Mars, Jupiter, Neptune	15 seconds each
SolarCube	30 seconds
GalaxyCube	30 seconds
InfoCube	90 seconds
MatrixCube	30 seconds
FacelD	5 seconds
BannerCube	15 seconds

## 10. Tuning Notes

Refresh artifacts and camera banding were reduced by experimenting with PWM timing and GPIO slowdown. The following globe command was captured as a high-quality photo/video test configuration.

```
sudo ./bin/globe -i maps/sun.jpg --led-rows=64 --led-cols=64 --led-chain=6
--led-gpio-mapping=adafruit-hat --led-pwm-dither-bits=1 --led-pwm-lsb-nanoseconds=130
--led-pwm-bits=10 --led-slowdown-gpio=4 -s 60 -v=2
```

Knob	Effect
--led-slowdown-gpio	Higher values can improve signaling stability; lower values may improve refresh and reduce visible artifacts
--led-pwm-bits	Controls PWM depth. Higher can improve gradients but may affect refresh
--led-pwm-lsb-nanoseconds	Adjusts PWM timing
--led-pwm-dither-bits	Can smooth perceived brightness and reduce banding
--led-brightness	Global panel brightness; good exhibit range is often 60 to 75

## 11. Blog and GitHub Release Checklist

Before publishing the project files with the blog, the following cleanup is recommended.

Action	Reason
Remove or redact config/infocube.conf	It contains a private calendar URL/token
Exclude bin/ and build/	Compiled artifacts are platform-specific and not needed for source distribution
Consider excluding backup/	Backups may confuse builders unless clearly labeled as historical
Add README.md	Point readers to this manual and give a quick-start build/run path
Add LICENSE file	Several files derive from Adafruit MIT examples; keep license notices intact
Add asset credits	Planet maps and images should be credited if redistributed
Document hardware wiring	Panel chain order and cube physical assembly are as important as the code

# Appendix A. Command Reference

## Earth globe

```
sudo ./bin/globe --led-rows=64 --led-cols=64 --led-chain=6 --led-gpio-mapping=adafruit-hat
--led-slowdown-gpio=6 -s 60 -v 2
```

## Jupiter globe

```
sudo ./bin/globe -i maps/jupiter.jpg --led-rows=64 --led-cols=64 --led-chain=6
--led-gpio-mapping=adafruit-hat --led-slowdown-gpio=6 -s 60
```

## Moon globe

```
sudo ./bin/globe -i maps/moon.jpg --led-rows=64 --led-cols=64 --led-chain=6
--led-gpio-mapping=adafruit-hat --led-slowdown-gpio=6 -s 60
```

## InfoCube

```
sudo ./bin/infocube --led-rows=64 --led-cols=64 --led-chain=6
--led-gpio-mapping=adafruit-hat --led-slowdown-gpio=6 -s 60
```

## MatrixCube

```
sudo ./bin/matrixcube --led-rows=64 --led-cols=64 --led-chain=6
--led-gpio-mapping=adafruit-hat --led-slowdown-gpio=6 -s 60
```

## SolarCube

```
sudo ./bin/solarcube -i maps/jupiter.jpg --led-rows=64 --led-cols=64 --led-chain=6
--led-gpio-mapping=adafruit-hat --led-slowdown-gpio=7 -s 60 --yaw=270
```

## GalaxyCube

```
sudo ./bin/galaxycube -i maps/jupiter.jpg --led-rows=64 --led-cols=64 --led-chain=6
--led-gpio-mapping=adafruit-hat --led-slowdown-gpio=7 -s 60 --yaw=90
--brightness-scale=0.6
```

## BannerCube

```
sudo ./bin/bannercube --led-rows=64 --led-cols=64 --led-chain=6
--led-gpio-mapping=adafruit-hat --led-slowdown-gpio=6 --text "MusicMaster Rocks"
--top-image images/jp_cube_premium_dual_top.jpg --bottom-image
images/jp_cube_premium_dual_bottom_rot180.jpg --speed 15 --brightness-scale 0.7
--text-color 5,52,243 --bg-color 221,224,237
```

## FaceID

```
sudo ./bin/faceid --led-rows=64 --led-cols=64 --led-chain=6 --led-gpio-mapping=adafruit-hat
--led-slowdown-gpio=6 -s 60
```

## Appendix B. Source Files Included

File	Size
Makefile	4,507 bytes
Makefile.bak	3,181 bytes
bannercube.cc	13,549 bytes
calendar.cc	10,758 bytes
calendar.h	724 bytes
config/infocube.conf	131 bytes
cube-cycle.service	472 bytes
cube-cycle.sh	3,701 bytes
cube_orientation.cc	1,362 bytes
cube_orientation.h	1,431 bytes
face_id.cc	7,631 bytes
face_id.cc.bak	3,787 bytes
galaxycube.cc	10,953 bytes
globe.cc	20,380 bytes
globe.cc.bak	16,520 bytes
globe_new.cc	16,520 bytes
images/jp_cube_cyan.jpg	1,736 bytes
images/jp_cube_premium_cyan_bottom_rot180.jpg	1,769 bytes
images/jp_cube_premium_cyan_top.jpg	1,772 bytes
images/jp_cube_premium_dual_bottom_rot180.jpg	1,777 bytes
images/jp_cube_premium_dual_top.jpg	1,775 bytes
images/jp_cube_premium_warm_bottom_rot180.jpg	1,727 bytes
images/jp_cube_premium_warm_top.jpg	1,724 bytes
images/jp_cube_warm.jpg	1,692 bytes
images/mm_cube_blue_v2.jpg	1,211 bytes
images/mmlogo.jpg	2,247 bytes
infocube.cc	55,064 bytes
infocube.cc.bak	54,651 bytes
life.cc	14,535 bytes
maps/adafruit.jpg	32,877 bytes
maps/boing.jpg	401 bytes
maps/earth.jpg	112,939 bytes
maps/earth2.jpg	4,565,076 bytes
maps/jupiter.jpg	54,249 bytes

File	Size
maps/jupiter1.jpg	3,080,980 bytes
maps/mars.jpg	8,396,951 bytes
maps/mercury.jpg	15,035,740 bytes
maps/moon.jpg	98,910 bytes
maps/moon2.jpg	15,030,356 bytes
maps/neptune.jpg	241,580 bytes
maps/neptune_title.jpg	150,096 bytes
maps/stars.jpg	1,905,513 bytes
maps/sun.jpg	822,427 bytes
maps/venus.jpg	12,523,129 bytes
matrixcube.cc	14,830 bytes
matrixcube.cc.bak	14,830 bytes
matrixcube.cc.new	15,080 bytes
scripts/infocube_cycle.sh	1,008 bytes
solarcube.cc	11,595 bytes
systemd/infocube-cycle.service	304 bytes
weather.cc	7,152 bytes
weather.h	877 bytes