Core64 Firmware Architecture

The project homepage for this product is at [www.Core64.io](http://www.Core64.io)

The latest version of this document is at <https://github.com/ageppert/Core64> along with the firmware and schematics.

Concept and design by Andy Geppert at [www.MachineIdeas.com](http://www.MachineIdeas.com)

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Description | Author | Date |
| - | Draft concepts and layout | Andy Geppert | 2020-01-08 |
| A | Revise architecture, layer names, add folders, remove RTOS reference, defining the path for next steps. As of today, FWV0.6.3 does not yet fully confirm to this architecture. | Andy Geppert | 2022-01-01 |
|  |  |  |  |

# TABLE OF CONTENTS

[TABLE OF CONTENTS 1](#_Toc91943267)

[Firmware Block Diagram 2](#_Toc91943268)

[Layer Descriptions and Examples 3](#_Toc91943269)

[Folder and File Organization 4](#_Toc91943270)

# Firmware Block Diagram

The User plays over here:

The “Operating System” lives here:

Core64.ino

LOOP(): Superloop Manager

Housekeeping

Heartbeat

Buttons Check

Run Top Level Mode (choose only one)

Memory Monitor

Process Monitor

Power Monitor

CLI Serial Port

Watchdog

Command\_Line\_Handler.ino

SETUP(): Init Hardware

Core64 Communications

Core64\_Communicator.cpp

M01\_TOP\_MENU.cpp

Top Menu (GAUSS)

* Games
* Applications
* Utilities
* Settings
* Special

Game #01 - Pong

Application #01 - Drawing

Settings #01 – Brightness

Utility #01 – Flux Detector

Special #01 – EEPROM CONFIG

Special #02 – FACTORY TEST

Settings #02 – Rotation

Settings #03 – …

Special #03 – …

Utility #02 – …

Application #02 - Calculator

Hardware Configuration Files

Hardware Abstraction Layer (HAL)

Core64 Drivers (Core Memory…)

3rd Party Libraries (FastLED …)

Arduino and Teensy and RP2040 Libraries

MCU

Hardware

# Layer Descriptions and Examples

|  |  |  |  |
| --- | --- | --- | --- |
| LAYERS | SYSTEM | LED ARRAY EXAMPLE | CORE MEMORY EXAMPLE |
| Application | Task\_Top\_Level\_State\_Machine: The user application. Initially this a few different demo and test modes of operation, using different combinations of Middleware Functionality. |  |  |
|  |  |  |  |
| Middleware |  |  |  |
| HAL |  |  | Core\_Mem\_Driver (rename this to Core\_Mem\_HAL): |
| Drivers and Board Support | Arduino hidden includes  HardwareIOMap.h |  |  |
| Hardware | Teensy  MCU  GPIO | Teensy  MCU  GPIO |  |

# Folder and File Organization

CORE64 FOLDER

Core64.ino // The traditional Arduino file, acting as the super loop operating system

CLI\_Handler.ino

MENUS

M01\_TOP\_MENU // Top Level Menu which runs on top of any other game/app/util…

M01.1\_GAMES

M01.2\_APPS

M01.3\_UTILS

M01.4\_SETTINGS

M01.5\_SPECIAL

GAMES

G01\_Pong.cpp // Standalone game logic for each game

G02\_Snakes.cpp

G03\_TicTacToe.cpp

APPS

A01\_Draw.cpp // Standalone application logic for each application

A02\_

A03\_

UTILS

U01\_Calc.cpp

U02\_Flux.cpp

U03\_

SETTINGS

SE01\_Brightness.cpp

SE02\_Rotation.cpp

SE03\_Diagnostics.cpp

SPECIAL

SP01\_EEPROMSet.cpp

SP02\_FactoryTest.cpp

SP03\_

src // Downloaded from other sources, may be tweaked for Core64.

Commandline

Commandline.cpp

Commandline.h

LTR329

LTR329\_Driver.cpp

LTR329\_Driver.h

Si7210

Si7210.c

Si7210.h

DRIVERS // Accessible by any file which includes it.

Core\_Driver\_C64\_\_HWV0.5.cpp

Core\_Driver\_C64\_\_HWV0.5.h

Core\_Driver\_C64c\_HWV0.2.cpp

Core\_Driver\_C64c\_HWV0.2.h

HALS // ONLY accessible from within the driver files.

Core\_HAL.cpp

Core\_HAL.h

CONFIG

Firmware\_Version.h

Hardware\_Version.h

HardwareIOMap.h

FastLED\_Config.h

# Compile-Time Differences and Settings

Arduino IDE

Compile with Teensy 3.2 or RP2040 selected. Pre-processor #ifdefs will confirm a compatible MCU has been selected in the Arduino IDE, and compile in the appropriate combination of drivers.

# Run-Time Start-up

Core64 Power Up

Detect available I2C devices

Check external EEPROM for hardware details and configuration.

Read start-up state of cores.

Display the state of the cores for 2 seconds on the LED Matrix.

Display Core64 Start-up Screen on LED Matrix.

Display instructions on OLED if attached.

Connect to Serial port.

Show the splash text and info.

Since serial port takes a second or two to connect on the computer, the other start-up checks completed and the visual display outputs are initialized first.

Command Prompt.