

Contents

1	Terms, Acknowledgments, and Licenses	3
1.1	Terms and Conditions	3
1.2	Acknowledgments and Contributors	3
1.3	Hardware License	3
1.4	Resources and References	3
1.5	Licenses	4
2	Main Pin Map	5
3	How to Generate an Error Report	9
3.1	Steps to Create an Error Report	9
3.2	Review and Follow-Up	9

Note: This documentation is actively evolving. For the latest updates and revisions, please visit the project's GitHub repository.

TERMS, ACKNOWLEDGMENTS, AND LICENSES

1.1 Terms and Conditions

By using, modifying, or distributing the documentation, firmware, or hardware designs included in this repository, you agree to the following terms:

- All materials are provided “**as-is**”, without warranty of any kind.
- The authors and contributors shall not be held responsible for **any damages**, data loss, or legal issues arising from the use of these materials.
- Usage is intended for **educational, development, prototyping**, and other lawful purposes.
- When redistributing or reusing any part of this project, you must **retain attribution** and comply with the corresponding license terms of each component.

1.2 Acknowledgments and Contributors

This project builds upon the work of several open-source developers and projects:

1.2.1 CMSIS-DAP (DAPLink Firmware for CH552)

- **Stefan Wagner** Project: [CH552-DAPLink](#) License: Creative Commons BY-SA 3.0 Description: CMSIS-DAP firmware and hardware design
- **Ralph Doncaster** Source: [nerdralph/ch554_sdcc](#) Description: Original CMSIS-DAP firmware implementation for CH554 (SDCC)
- **Deqing Sun** Source: [CH55xduino](#) Description: CH552/CH554 Arduino-compatible toolchain

1.2.2 USB-Blaster Firmware (CH552G)

- **Vladimir Duan** Project: [CH55x-USB-Blaster](#) License: MIT Description: USB-Blaster JTAG emulation for CH55x
- **Blinkinlabs** SDK Source: [ch554_sdcc](#) Description: SDK for CH552/CH554 (SDCC)
- **Doug Brown** Blog: [Fixing a Knockoff Altera USB Blaster](#) Description: Insights into compatibility and firmware flashing

1.3 Hardware License

All hardware designs (schematics, layouts, and design files) in this repository are released under the **MIT License**, allowing unrestricted use, modification, and distribution, provided the original license and attribution are retained.

1.4 Resources and References

Table 1.1: Source URLs

Project / Tool	Source URL
CH552 DAPLink	https://github.com/wagiminator/CH552-DAPLink
picoDAP	https://github.com/wagiminator/CH552-picoDAP
CH55xDuino	https://github.com/DeqingSun/ch55xduino
CMSIS-DAP Handbook	https://os.mbed.com/handbook/CMSIS-DAP
CH55x USB-Blaster	https://github.com/VladimirDuan/CH55x-USB-Blaster
SDCC Compiler	https://sdcc.sourceforge.net/
CH554 SDK	https://github.com/Blinkinlabs/ch554_sdcc

1.5 Licenses

1.5.1 Documentation & Visual Content

This user guide and its visual content are licensed under:

Creative Commons Attribution-ShareAlike 3.0 Unported License



1.5.2 Firmware Projects

- **CH552-DAPLink:** Creative Commons BY-SA 3.0 — © Stefan Wagner
- **CH55x-USB-Blaster:** MIT License — © Vladimir Duan
- **CH55x SDK / Tools:** MIT License — © Blinkinlabs

1.5.3 Hardware Repository

- All PCB designs and schematics are released under the **MIT License**.

Note: If you distribute this product with third-party firmware (e.g., CMSIS-DAP), you are responsible for ensuring license compliance. Only firmware developed by Unit Electronics and released under the MIT license is supported for commercial redistribution.

1.5.4 Preloaded USB-Serial Firmware

This product may include preloaded firmware based on the project by **Kongou Hikari**: “USB to Serial Converter firmware for CH552T”. Original source: [\[https://github.com/diodep/ch55x_dualserial/tree/master\]](https://github.com/diodep/ch55x_dualserial/tree/master)
License: MIT

Under the terms of the MIT License, users are free to modify or replace the firmware. Unit Electronics provides this firmware for convenience only and does not offer performance guarantees.

MAIN PIN MAP

Table 2.1: Main Pin Map – ESP32-S3 TouchDot

Ar-duino Lily-PAD	UNIT TouchDot S3	ESF S3 GPI	GPIO Function	Type
D13 (SCK)	D13/SCK/T7	GPI0	RTC_GPIO7, GPIO7, TOUCH7, ADC1_CH6	I/O/T
3.3V	3.3V	3.3V	Power supply	P
AREF	•	•	•	•
A0 (Ana-log)	A0/T1	GPI0	RTC_GPIO1, GPIO1, TOUCH1, ADC1_CH0	I/O/T
A1 (Ana-log)	A1/T2	GPI0	RTC_GPIO2, GPIO2, TOUCH2, ADC1_CH1	I/O/T
A2 (Ana-log)	A2/T3	GPI0	RTC_GPIO3, GPIO3, TOUCH3, ADC1_CH2	I/O/T
A3 (Ana-log)	A3/T4	GPI0	RTC_GPIO4, GPIO4, TOUCH4, ADC1_CH3	I/O/T
A4 (SDA)	A4/(SDA)/T5	GPI0	RTC_GPIO5, GPIO5, TOUCH5, ADC1_CH4	I/O/T
A5 (SCL)	A5/(SCL)/T6	GPI0	RTC_GPIO6, GPIO6, TOUCH6, ADC1_CH5	I/O/T
•	A6/D13/SCK	GPI0	ADC1_CH5, LP_UART_TXD, LP_GPIO5, MTDI, FSPIWP, SDIO	I/O/T
•	A7/D12/MIS	GPI0	WS2812B-2020 OUT (DO)	I/O/T
•	A8/D11/MOS	GPI0	WS2812B-2020 OUT (DO)	I/O/T
5V	5V	5V	Power supply	P
RE-SET	RST	EN	High: ON, enables the chip. Low: OFF	I
GND	GND	GND	GND	GND
D0 (RX)	D0/RX	GPI0	U0RXD, GPIO44, CLK_OUT2	I/O/T
D1 (TX)	D1/TX	GPI0	U0TXD, GPIO43, CLK_OUT1	I/O/T
D2	D2/T11	GPI0	RTC_GPIO11, TOUCH11,	I/O/T

Table 2.2: QWIIC-Compatible JST Connector

Pin	JST Function	Ar-duino Com-patibility	ESP S3 GPIO	GPIO Function
1	GND	GND	GND	GND
2	3.3V	3.3V	3.3V	3.3V
3	SDA / MUX IO	A4	GPIO5	RTC_GPIO5, GPIO5, TOUCH5, ADC1_CH4
4	SCL / MUX IO	A5	GPIO6	RTC_GPIO6, GPIO6, TOUCH6, ADC1_CH5

Table 2.3: JTAG Test Points

Function	Ar-duino Pin	ESP S3 GPIO	GPIO Function	Type
MTCK	D21	GPIO39	MTCK, CLK_OUT3, SUB-SPICS1	I/O/T
MTDO	D22	GPIO40	MTDO, CLK_OUT2	I/O/T
MTDI	D23	GPIO41	MTDI, CLK_OUT1	I/O/T
MTMS	D24	GPIO42	MTMS, GPIO42	I/O/T
GND1	GND	GND	GND	GND
TP_3V3	3.3V	Power Supply	P (3.3V)	P

Table 2.4: Serial Programming Header (1x6)

Pi	JST Function	Ar-duino Com-patibility	ESP S3 GPIO	GPIO Function	Type
1	GND	GND	GND	GND	GND
2	EN	RESET	EN	High: ON, enables chip; Low: powers off	I
3	3.3V	3.3V	3.3V	3.3V	P
4	TX0	D1	GPIO43	U0TXD, GPIO43, CLK_OUT1	I/O/T
5	RX0	D0	GPIO44	U0RXD, GPIO44, CLK_OUT2	I/O/T
6	BOOT	D29	GPIO0	RTC_GPIO0, GPIO0	I/O/T

Table 2.5: Expansion Header (2x6)

P	Function	Ar-duino Pin	ESP S3 GPIO	GPIO Function	Type
1	3.3V	3.3V	•	Power Supply	P
2	GND	GND	•	Ground	GND
3	GPIO33	D15	GPIO	SPIIO4, GPIO33, FSPiHD, SUB-SPIHD	I/O/T
4	GPIO34	D16	GPIO	SPIIO5, FSPICS0, SUBSPICS0	I/O/T
5	GPIO35	D17	GPIO	SPIIO6, FSPID, SUBSPID	I/O/T
6	GPIO36	D18	GPIO	SPIIO7, FSPI-CLK, SUBSPI-CLK	I/O/T
7	GPIO37	D19	GPIO	SPIDQS, FSPIQ, SUBSPIQ	I/O/T
8	GPIO38	D20	GPIO	GPIO38, FSPiWP, SUBSPiWP	I/O/T
9	GPIO47 / PDM_D ⁺	D27	GPIO	SPICLK_P_DIFF, SUBSPI-CLK_P_DIFF	I/O/T
10	GPIO48 / PDM_CI	D28	GPIO	SPI-CLK_N_DIFF, SUBSPI-CLK_N_DIFF	I/O/T
11	5V	5V	•	Power Supply	P
12	GND	GND	•	Ground	GND

Table 2.6: microSD Connector (SPI Mode)

P	mi-croSD Pin Name	SPI Function	ESP S3 GPIO	GPIO Function	Type
1	DAT2	Not used in SPI	•	•	•
2	CD / DAT3	CS (Chip Select)	GPIO	RTC_GPIO2, GPIO21	I/O/T
3	CMD	MOSI	GPIO	RTC_GPIO9, TOUCH9, ADC1_CH8, FSPiHD, SUB-SPIHD	I/O/T
4	VDD	3.3V	3.3V	Power Supply	P
5	CLK	SCLK	GPIO	RTC_GPIO7, TOUCH7, ADC1_CH6	I/O/T
6	VSS	GND	GND	Ground	GND
7	DAT0	MISO	GPIO	RTC_GPIO8, TOUCH8, ADC1_CH7, SUBSPICS1	I/O/T
8	DAT1	Not used in SPI	•	•	•

HOW TO GENERATE AN ERROR REPORT

This guide explains how to generate an error report using GitHub repositories.

3.1 Steps to Create an Error Report

1. Access the GitHub Repository

Navigate to the [GitHub repository](#) where the project is hosted.

2. Open the Issues Tab

Click on the “Issues” tab located in the repository menu.

3. Create a New Issue

- Click the “New Issue” button.
- Provide a clear and concise title for the issue.
- Add a detailed description, including relevant information such as:
 - Steps to reproduce the error.
 - Expected and actual results.
 - Any related logs, screenshots, or files.

4. Submit the Issue

Once the form is complete, click the “Submit” button.

3.2 Review and Follow-Up

The development team or maintainers will review the issue and take appropriate action to address it.