



The Byte Attic's

Agon light™

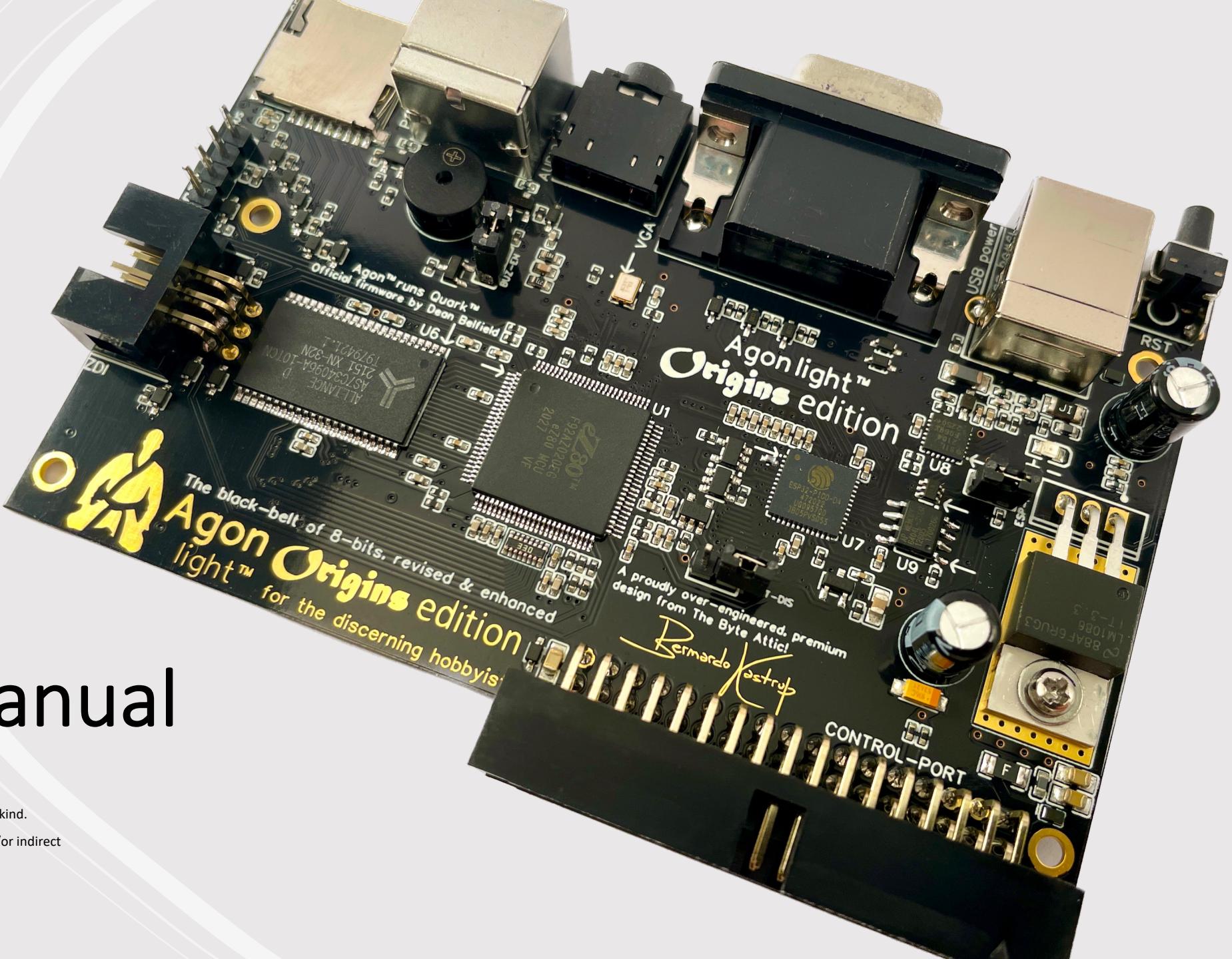
Hardware Manual

© 2022-2023 by Bernardo Kastrup. All rights reserved.

Last updated on: 7/19/23 2:11:17 AM

Provided as-is, expressly without warranties and/or representations of any kind.

The author disclaims all responsibility for damages incurred as a direct and/or indirect result of the use of this manual and/or the system it describes.



What is Agon light™?

- A modern 8-bit microcomputer and microcontroller in one small, low-cost board
- Requires no host PC: Agon light puts out its own video (VGA, various modes, 64 colors), audio (2 identical mono channels), accepts a PS/2 keyboard and has µSD-card storage
- Features a control port with SPI, I²C, 20 distinct GPIOs, a system clock output, as well as power (3.3V and 5V) and ground rails
- Features a separate ACCESS.bus header for e.g. an optional status display
- Aims at the best possible trade-off across performance, cost and flexibility with cutting-edge technology
- There are no FPGAs and no emulation in Agon™: the 'bare metal' is exposed directly to the firmware programmer
- Agon light is powered by USB and runs internally at 3.3V

What is so unique and attractive about it?

- Instant-on, stand-alone, BASIC-programmed* microcontroller: no host PC or sketch compilation required
- Control your whole house from the immediacy of a BASIC prompt! *
- Say goodbye to assembly:
 - C-programmable audio/video coprocessor firmware with freely available tooling
 - C-programmable CPU firmware with freely available tooling
- A hardware canvas for you to make of it your own dream, firmware-customized microcomputer
- A laboratory for computer science experimentation
- The most advanced 8-bit microcomputer to date
- The best balance of cost, performance and programmability
- Agon light is an open-hardware and open-source project, so you get *all* the information about the system

* Requires installation of Quark™ firmware by Dean Belfield



The Byte Attic's
Agon
light™

Technical
overview and
specifications



Architecture and specifications

- Two subsystems:
 - The *processor subsystem*
 - The *terminal subsystem*
- The *processor subsystem* comprises:
 - CPU (eZ80F92 running at 18.432MHz)
 - System memory (512KB, 10ns, parallel SRAM)
 - µSD-card port (as main storage)
 - ZDI port (for programming the firmware of the CPU)
 - Control port (including 20 GPIOs) to control your projects from BASIC*
- The *terminal subsystem* comprises:
 - Audio/video coprocessor (ESP32-PICO-D4 running at 240MHz)
 - Terminal memory (8MB, 133MHz, serial pSRAM)
 - Keyboard port (PS/2)
 - VGA port (various modes, 64 colors)
 - Audio jack (2x mono)
 - USB 2.0 port (for power and programming the ESP32's firmware)
- The two subsystems communicate with each other via full-duplex high-speed serial link (1.152 megabits per second), featuring flow control

* Requires installation of Quark™ firmware by Dean Belfield

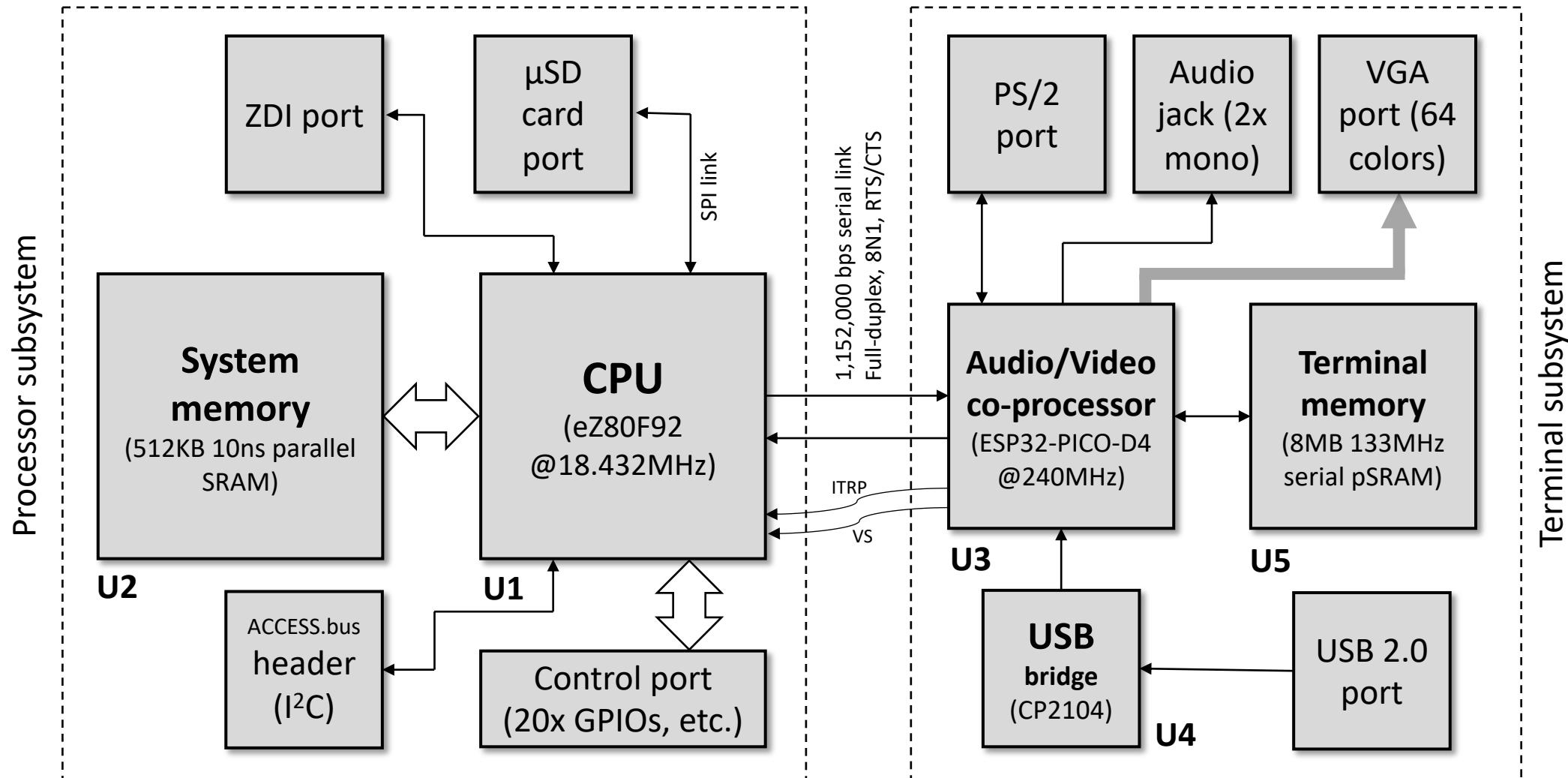
Theory of operation

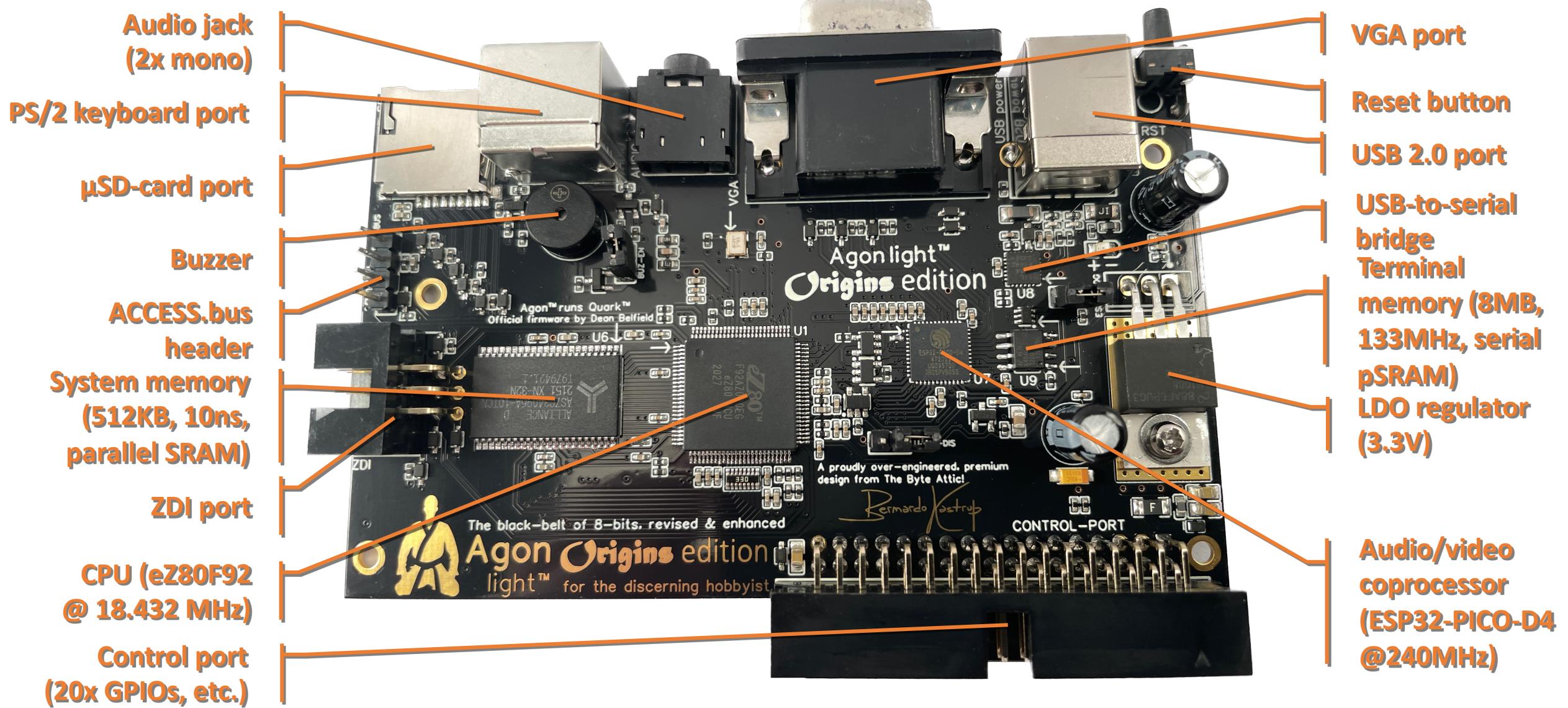
- The *terminal subsystem*:
 - Reads out the (PS/2) keyboard and sends the corresponding keypress tokens to the CPU via a high-speed serial link
 - Generates the screen based on display-list commands issued by the CPU and sent to the ESP32 via a high-speed serial link
 - Produces the VGA & audio signals
 - Supports the FabGL™ library
 - Sends the vertical synch signal (**VS**, from pin 21/IO15) both to the VGA port *and the CPU*
 - Sends a general-purpose, firmware-programmable signal (**ITRP**, from pin 28/SD2) to the CPU
- The *processor subsystem*:
 - Runs the BIOS and BASIC interpreter*
 - Executes application code
 - Drives the GPIOs based on the application code
 - Drives the *terminal subsystem* by issuing display-list and audio-related commands to the ESP32 via a high-speed serial link
 - Manages storage (μ SD-card)
 - The eZ80F92 CPU receives the vertical synch (**VS**, in pin 89/PB1/T1_IN) and a general-purpose firmware-programmable signal (**ITRP**, in pin 88/PB0/T0_IN) from the ESP32, both of which can be used by the eZ80F92 as interrupts

* Requires installation of Quark™ firmware
by Dean Belfield

System diagram

↔ Serial link
↔ Parallel link



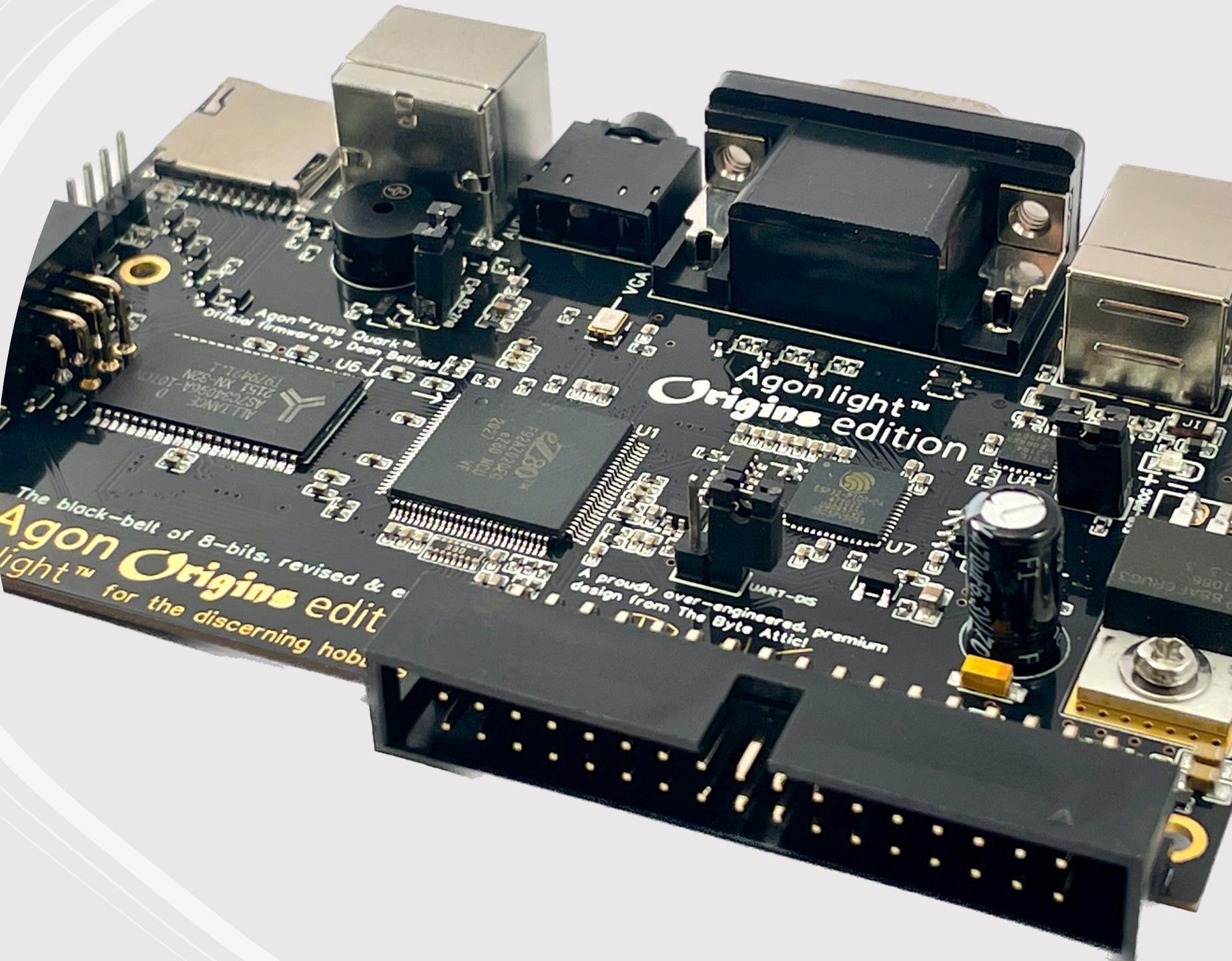




The Byte Attic's

Agon light™

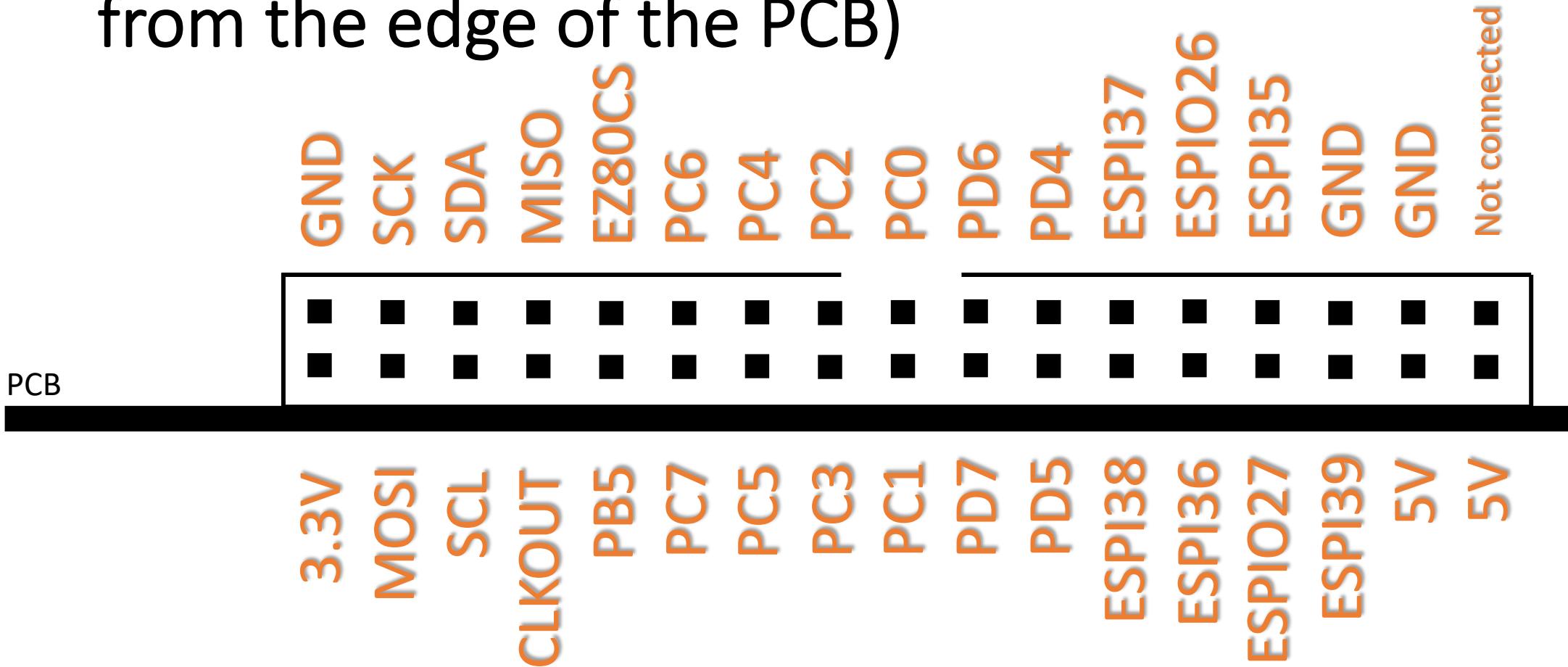
User's
guide



Control port signal descriptions

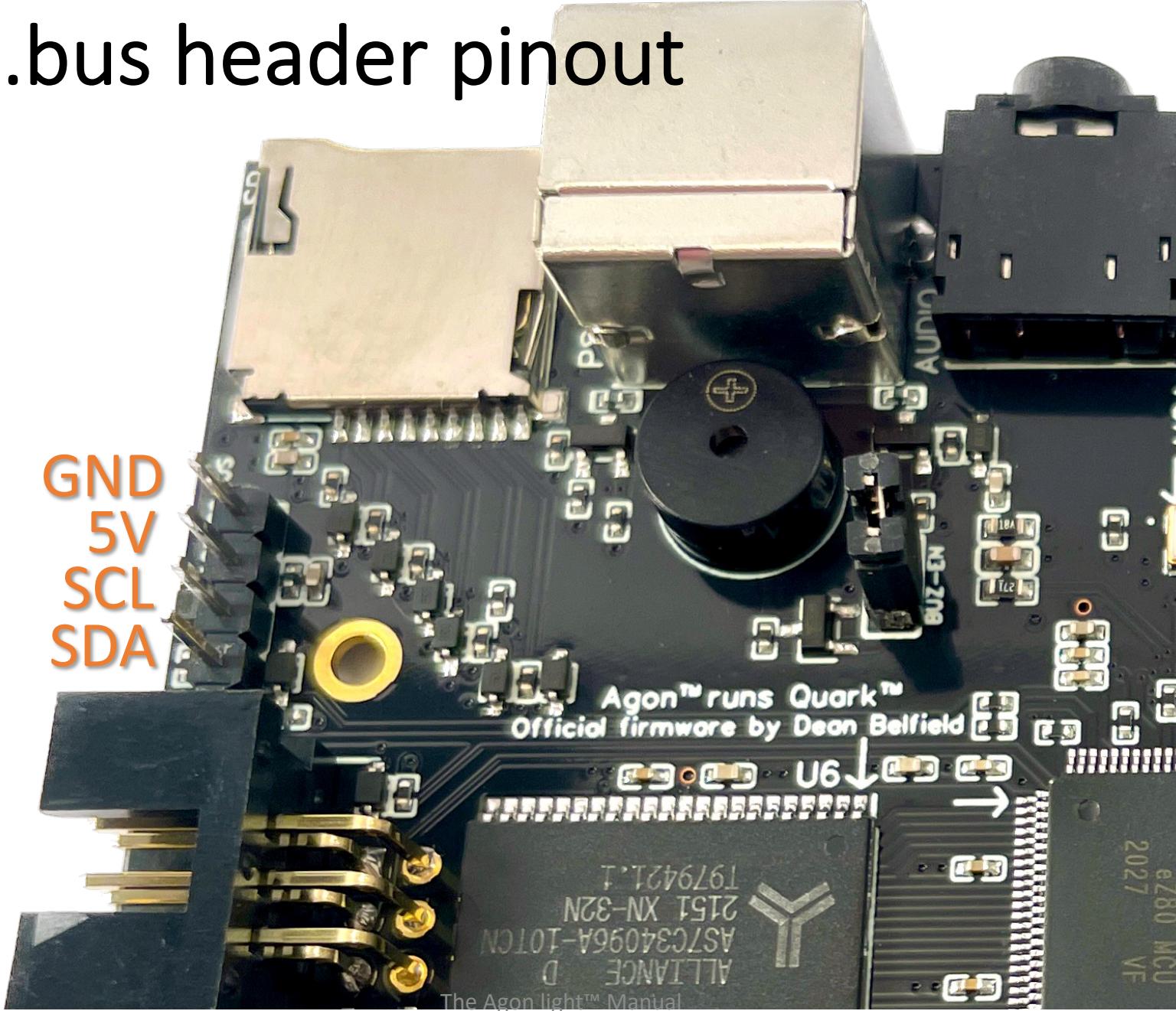
- **CLKOUT**: System clock (18.432MHz) buffered by the eZ80F92 CPU (PHI)
- ESP32-PICO-D4 *bidirectional* GPIOs:
(see datasheet for clarifications)
 - **ESPIO26** (pin 15, IO26) and **ESPIO27** (pin 16, IO27), both pulled up by 22KΩ resistors
- ESP32-PICO-D4 GP *inputs*:
(see datasheet for clarifications)
 - **ESPI39** (SENSOR_VP), **ESPI38** (SENSOR_CAPP), **ESPI37** (SENSOR_CAPN), **ESPI36** (SENSOR_VN), **ESPI35** (IO35)
- **MOSI** (pin 95, PB7), **SCK** (pin 91, PB3), **MISO** (pin 94, PB6), **EZ80CS** (pin 90, PB2/!SS): SPI signals of the eZ80F92
- **SDL**, **SCA**: I²C signals of the eZ80F92
- eZ80F92 *multi-functional, bidirectional* GPIOs:
(see datasheet for clarifications)
 - **PB5/T5_OUT**
 - **PC0/TxD1**, **PC1/RxD1**, **PC2/!RTS1**, **PC3/!CTS1**, **PC4/!DTR1**, **PC5/!DSR1**, **PC6/!DCD1**, **PC7/!RI1**
 - **PD4/!DTR0**, **PD5/!DSR0**, **PD6/!DCD0**, **PD7/!RI0**

Control port pinout (looking into the connector from the edge of the PCB)



See schematics and eZ80F92 and ESP32-PICO-D4
datasheets for more comprehensive signal descriptions

ACCESS.bus header pinout



Pinout of serial link between CPU and ESP32

- On the *eZ80F92*'s side:
 - Pin 68 (**PD0/TXD0/IR_TXD**) is the transmitter
 - Pin 69 (**PD1/RXD0/IR_RXD**) is the receiver
 - Pin 70 (**PD2/!RTS0**) is RTS (signal '*eZ80RTS*' in the schematics)
 - Pin 71 (**PD3/!CTS0**) is CTS (signal '*eZ80CTS*' in the schematics)
- On the *ESP32-PICO-D4*'s side:
 - Pin 10 (**IO34**) is the receiver (connected to signal '*eZ80TxD*' in the schematics)
 - Pin 22 (**IO2**) is the transmitter (connected to signal '*eZ80RxD*' in the schematics)
 - Pin 17 (**IO14**) is CTS (connected to signal '*eZ80RTS*' in the schematics)
 - Pin 20 (**IO13**) is RTS (connected to signal '*eZ80CTS*' in the schematics)

Recommended configuration of serial link between CPU and ESP32

Channel: full duplex, asynchronous

Baud rate: 1,152,000 bits per second

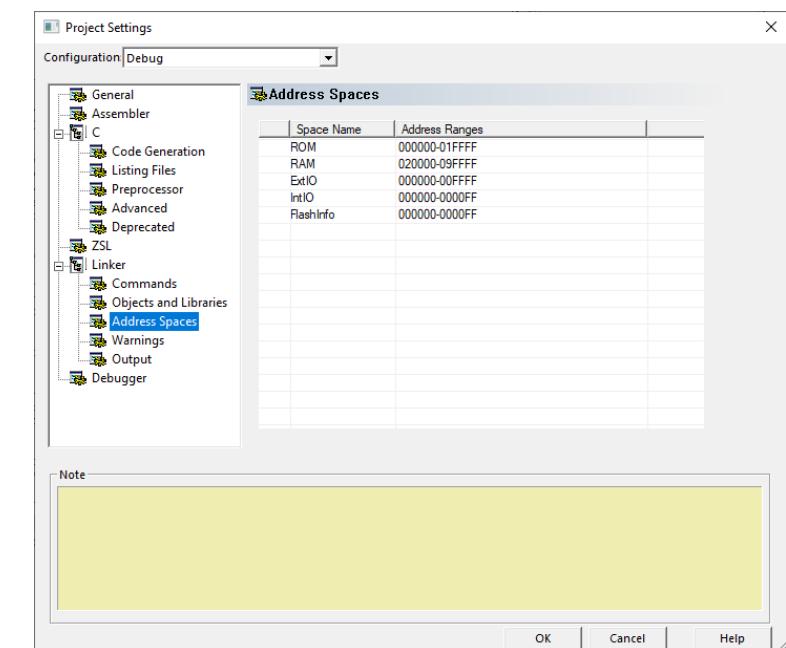
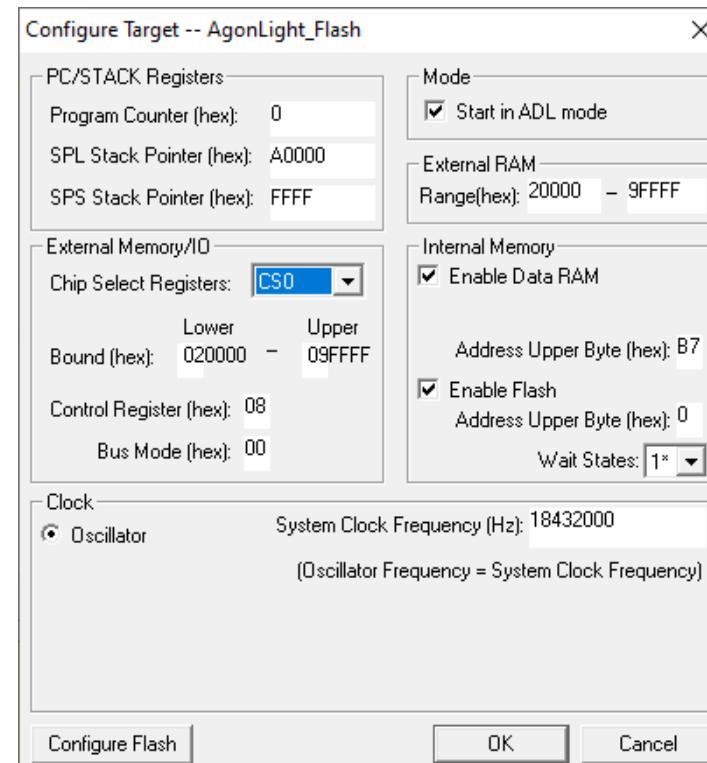
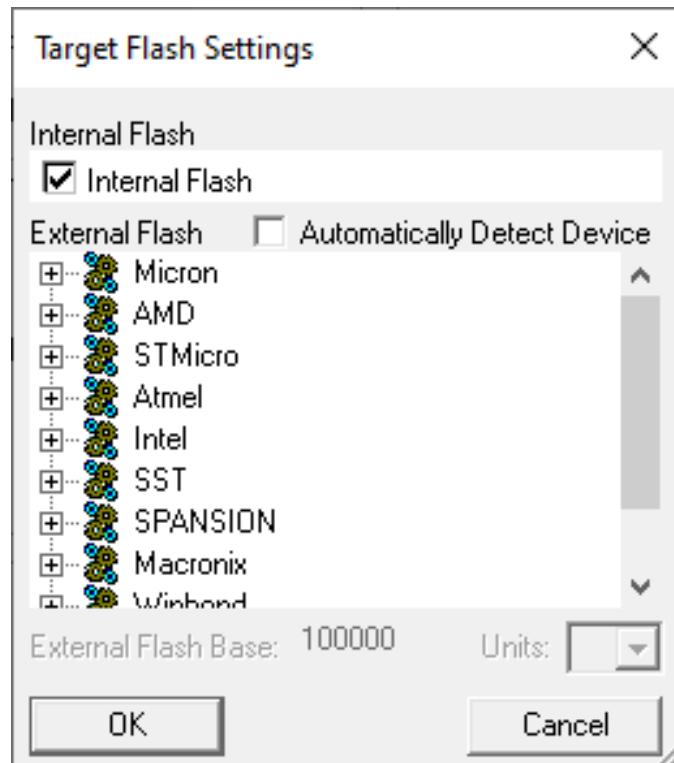
Signal structure: 1 start bit, 8 data bits, 1 stop bit,
no parity bit (8N1)

Flow control: CTS/RTS

Other possible baud rates are: 115200, 128000, 144000, 192000, 230400, 288000,
384000, and 576000 bps

Developing firmware for the eZ80F92

- Use the freely-available Zilog ZDS-II™ IDE, downloadable from:
https://www.zilog.com/index.php?option=com_zcm&task=view&soft_id=54&Itemid=74
- Configure your project as per the figures below (CS1, CS2 and CS3 are *not* used in Agon light, so their settings don't matter)



Required programming/debugging USB smart capable

- To upload firmware into the eZ80F92 CPU, from within the ZDS-II IDE, you will need a Zilog opto-isolated *USB Smart Cable*

- Zilog product numbers:

ZUSBSC00100ZACG (discontinued)

ZUSBASC0200ZACG (current; requires v5.3.5 or later of Zilog's **ZDS-II IDE**)

ATTENTION: the cable with product number ZUSBESC0200ZACG is **NOT** suitable for Agon light™!

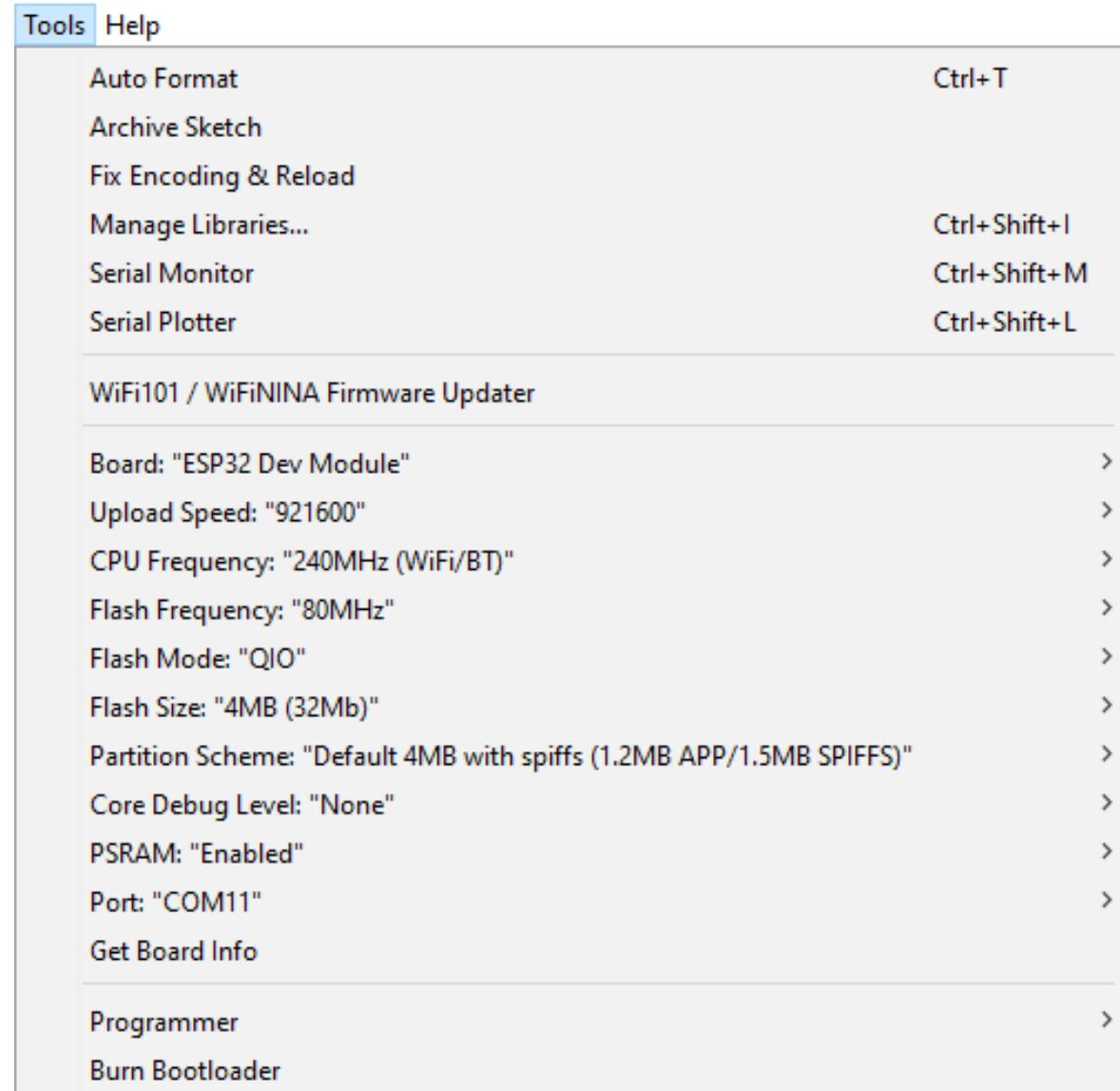
- This cable is only needed for programming the eZ80 *for the first time*. If you bought a pre-programmed Agon light™, you can update the firmware *without* this cable, by using the software utility here:

<https://github.com/envenomator/agon-flash>



Developing firmware for the ESP32-PICO-D4

- Use the freely-available Arduino™ IDE
- Install the FabGL™ library as per instructions available online
 - Link to the FabGL library: <http://www.fabgl.org/index.html>
 - Link to installation tutorial: <https://youtu.be/8OTaPQISTas>
- The figure to the right illustrates a suitable configuration for loading an Arduino sketch into the ESP32
 - Change the port number to the one active in your case



Power supply and signal level considerations

- Agon light can be powered (5V) *either* from its USB port *or* from the 5V pin in its control port
- If Agon light is powered from the USB port, then the 5V pin in the control port can be used to power an external circuit connected to Agon light
- Similarly, the 3.3V pin in the control port can be used to power an external circuit, *but it cannot be used to power Agon light*
- The on-board LDO regulator can provide up to 1.5A of current at 3.3V
 - This is the maximum *total* current for Agon light's internal use *and* devices powered from the 3.3V pin in the control port
 - It assumes that the USB device powering Agon light can deliver 1.5A; otherwise, that device becomes the bottleneck
- All GPIO/I²C/SPI logic signals on the control and ACCESS.bus ports are referenced to 3.3V and, therefore, are *not* TTL-level
 - You must use (two-way) level-shifters if you plan to integrate those signals with external circuitry running at 5V-level
- The GPIO/I²C/SPI logic signals on the control and ACCESS.bus ports are NOT buffered
 - Those signals have the current and fanout limitations described in the eZ80F92 and ESP32-PICO-D4 datasheets
 - It is recommended that you buffer those signals before driving external circuits with them, particularly for larger fanouts
 - If you use an unbuffered signal to drive an external LED, a 1KΩ current-limiting resistor, in series with the LED, is (highly) recommended

Power through USB

- For powering Agon light ORIGINS edition™ alone, a USB 2.0 cable with a *male USB-B* connector will suffice (it will deliver up to 500mA at 5V)
- For powering Agon light ORIGINS edition™ *and* another circuit attached to Agon light's control port, a USB 3.0 cable with a *male USB-B* connector is recommended (it will deliver close to 1A at 5V)

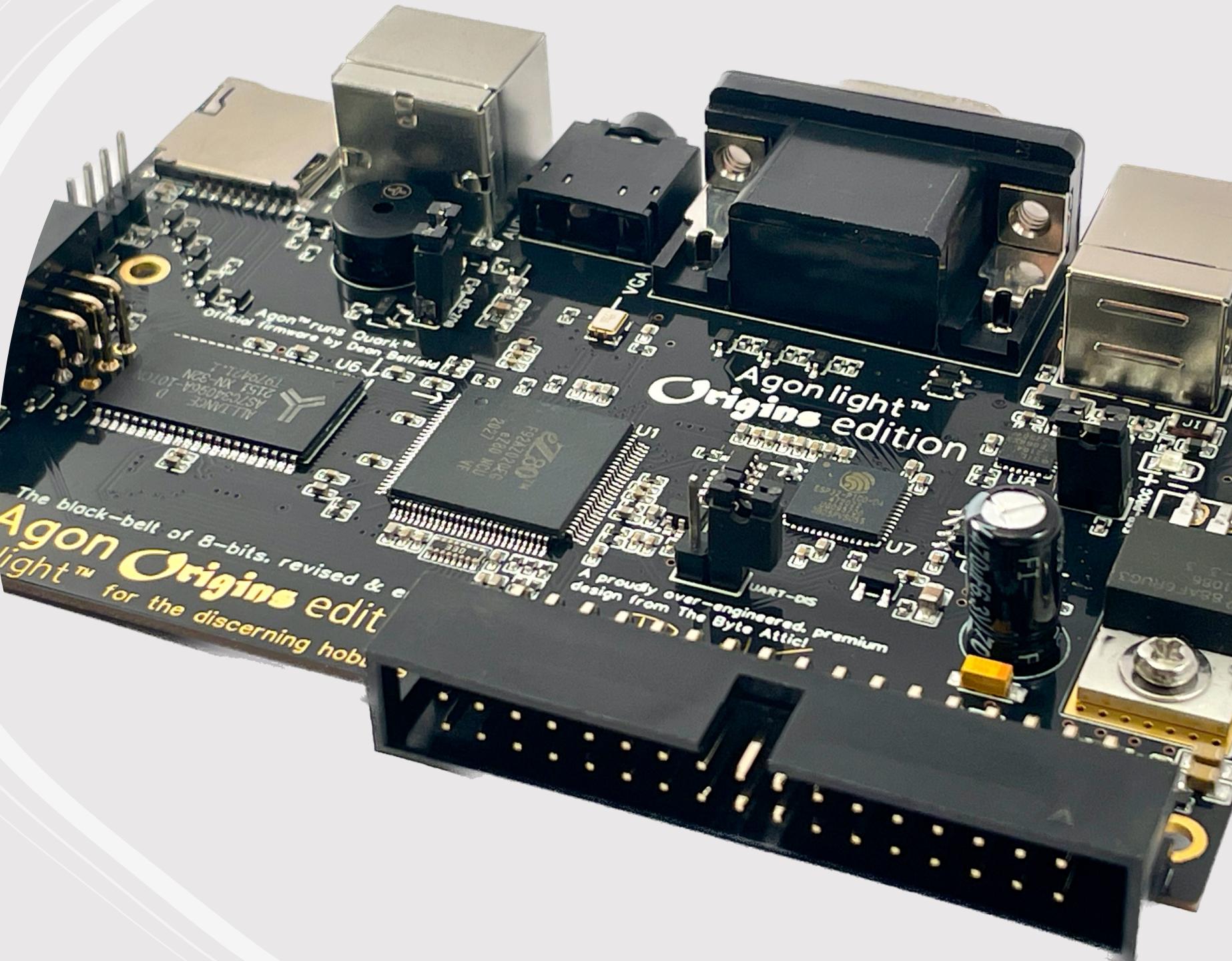




The Byte Attic's

Agon light™

Assembly
guide

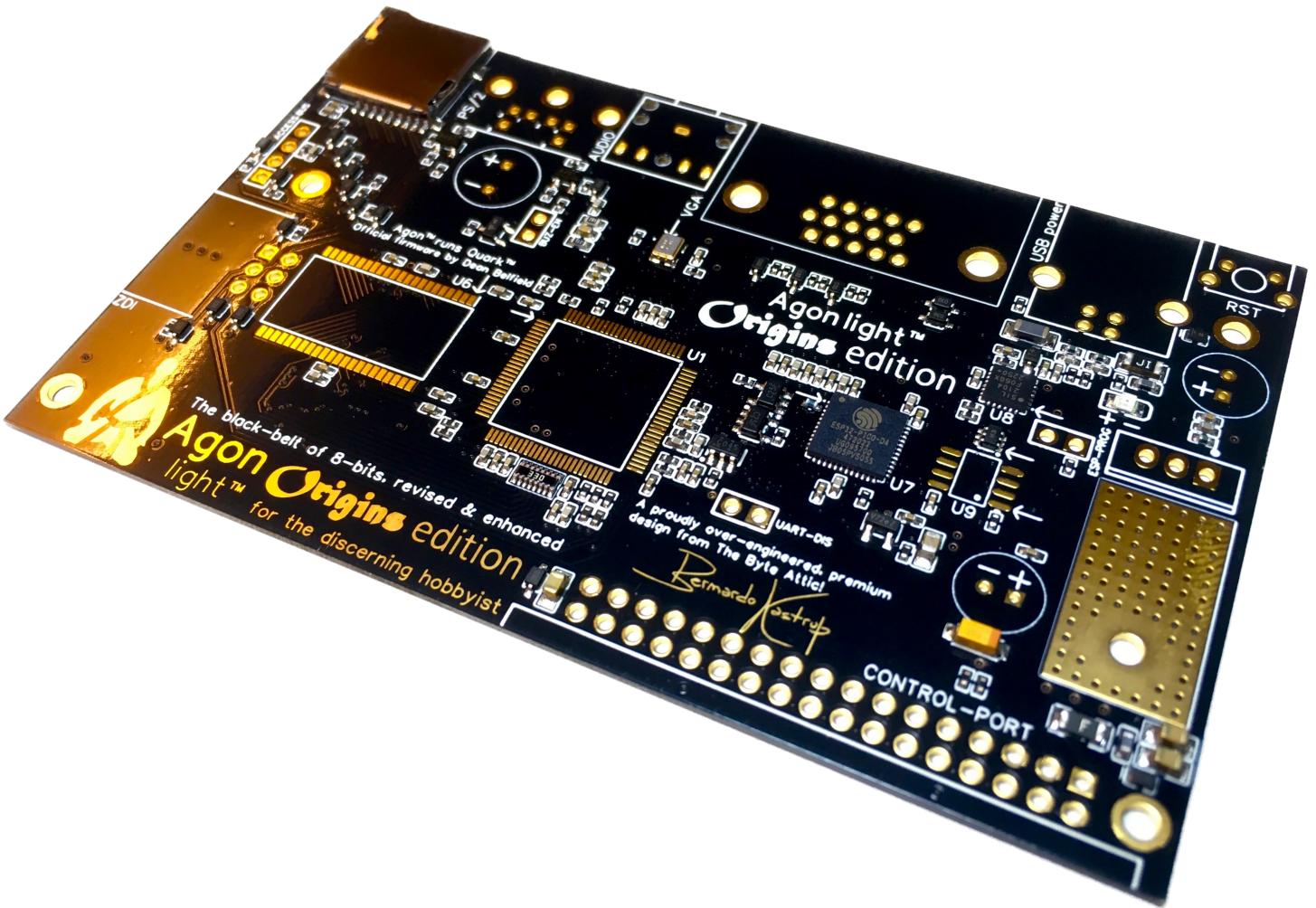


Assembly options

- There are three options:
 1. You buy the bare PCB and fully populate it yourself (requires a stencil and reflow oven)
 2. You buy a PCB minimally populated with the small parts and the two QFN ICs (the ESP32 and the pSRAM), which are hard to solder by hand
 3. You buy a fully-populated board, so you need not do any soldering yourself
- Options (1) and (3) will not be discussed further: if you choose option (1) you know what you are doing, and option (3) requires nothing of you
- Reasons for choosing option (2): PCB makers charge a premium (usually 50% of the parts' costs) for procuring parts for you, and there are multiple import fees involved. It's cheaper (and better, if you know how to do it) to buy and populate the most expensive parts yourself, especially if you are building Agon light to sell it commercially

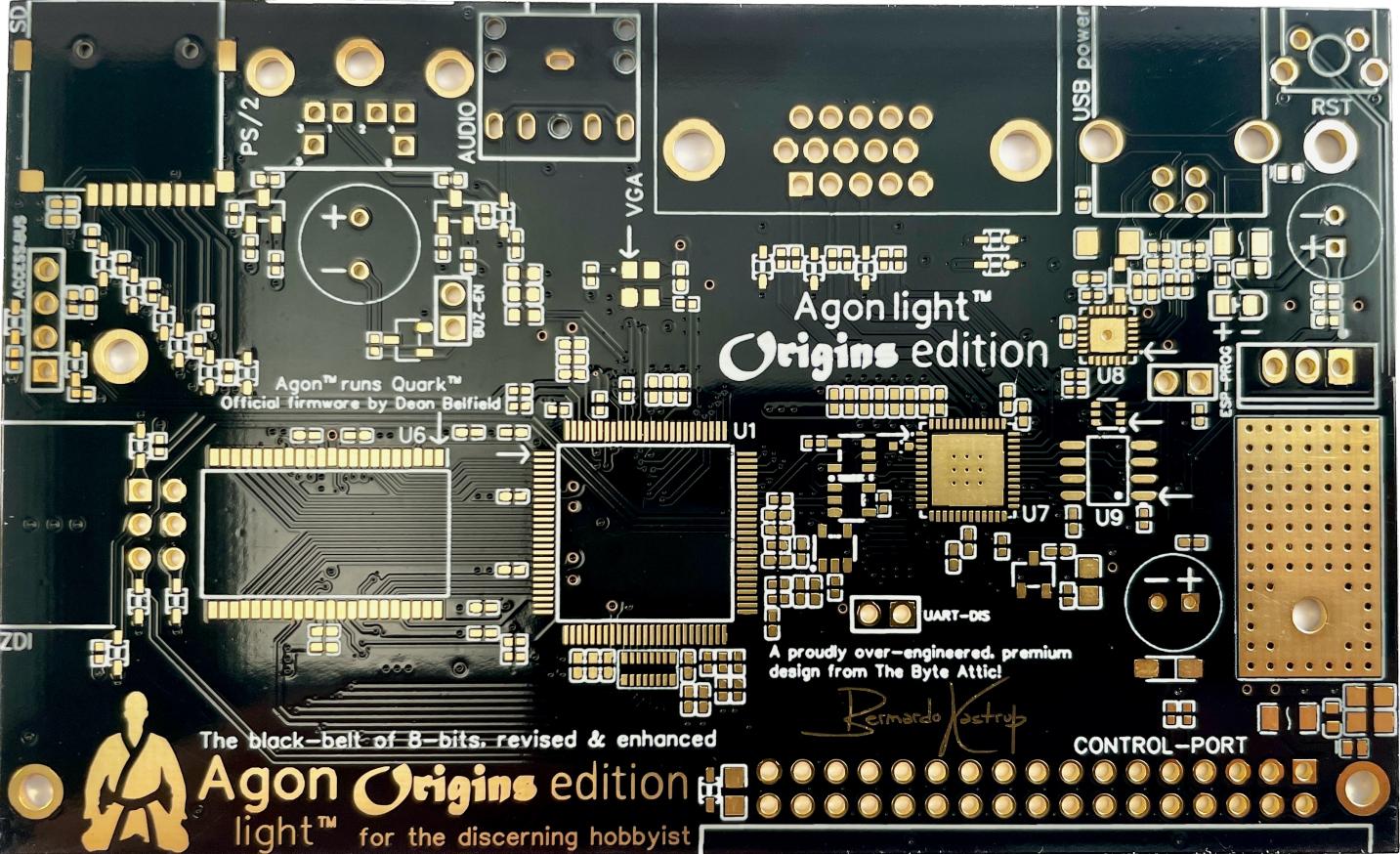
Manufacturing files

- All files are available in the /Manufacturing directory of Agon light's Github repository at: <https://github.com/TheByteAttic/AgonORIGINS>
- For option (3), send the following files to your PCB manufacturer, next to the Gerber file (Agon ORIGINS Gerber PCB.zip):
 - Agon ORIGINS Pick & Place complete.csv
 - Agon ORIGINS BoM complete.csv
- For option (2), send these:
 - Agon ORIGINS Pick & Place minimal.csv
 - Agon ORIGINS BoM minimal.csv
- The board's silkscreen indicates pin 1 of each chip with a small arrow



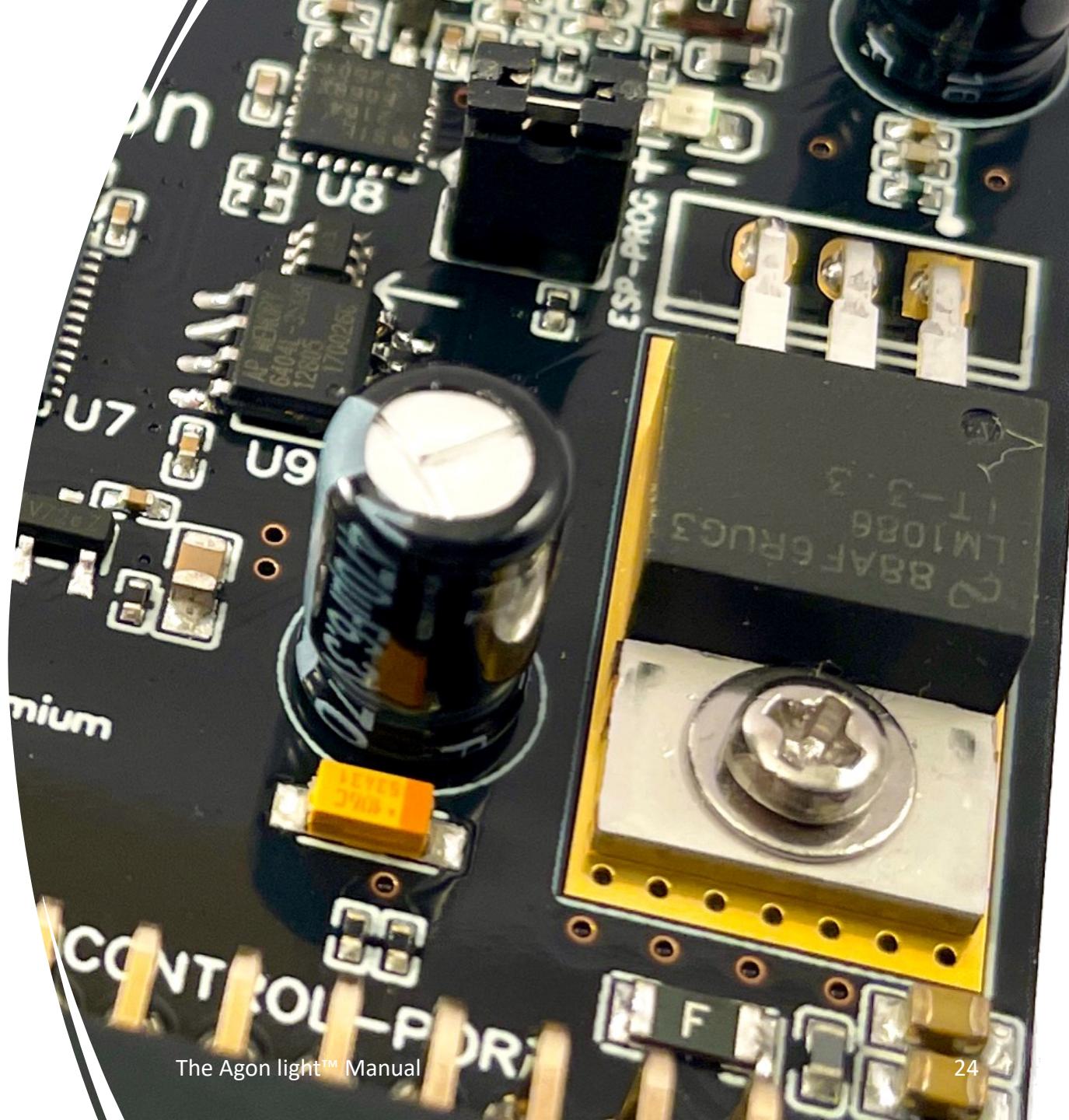
PCB layer stack

- Agon light's PCB has four layers:
 - Two signal layers (top and bottom)
 - Two inner planes (GND and 3.3V)
- The stack is as follows:
 - *TopLayer* (signals + GND copper fill)
 - *Inner1* (3.3V plane)
 - *Inner2* (GND plane)
 - *BottomLayer* (signals + 3.3V copper fill)
- Agon light has tiny VIAs: **0.4mm** diameter with **0.205mm** drill holes, so choose a compatible process with your manufacturer
- Total PCB thickness of **1mm** is recommended, so to improve signal integrity
- The board's silkscreen indicates pin 1 of each chip with a small arrow



Mounting the LDO regulator

- Agon light's 3.3V V_{cc} rail is provided by a Low-DropOut (LDO) linear regulator
- The regulator must be mounted flush against the corresponding exposed metal area on the top of the PCB (see photo)
- The regulator's tab (chassis) is at 3.3V, as is the exposed copper area on which it is to be mounted
- Use *no thermal paste or insulating spacers*; simply clean the tab and the exposed metal area with IPA before mounting
- Affix the regulator with a 2mm-diameter bolt, a regular and a lock washer on the top, and a nylon (or other soft material) washer and nut at the back

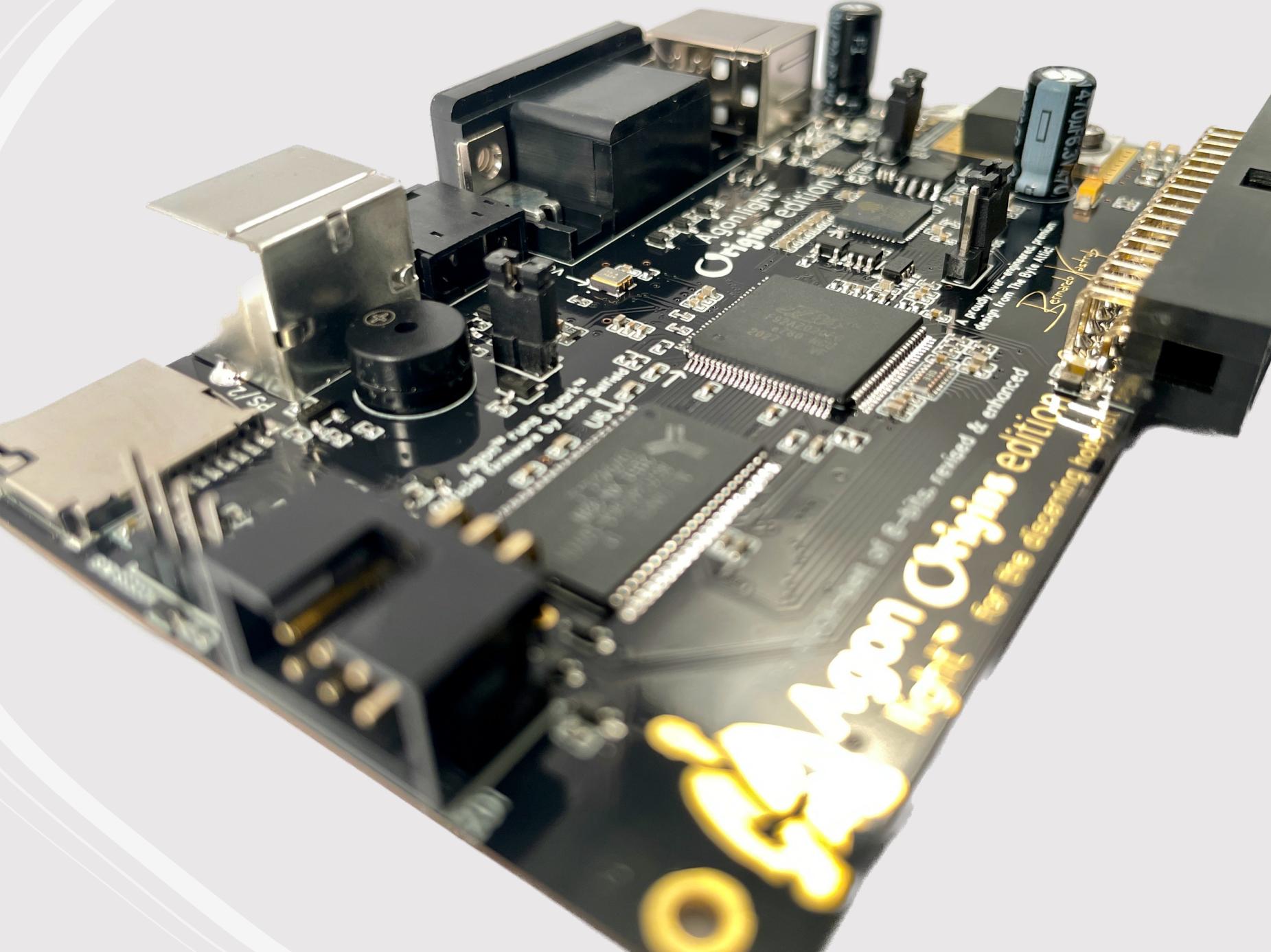


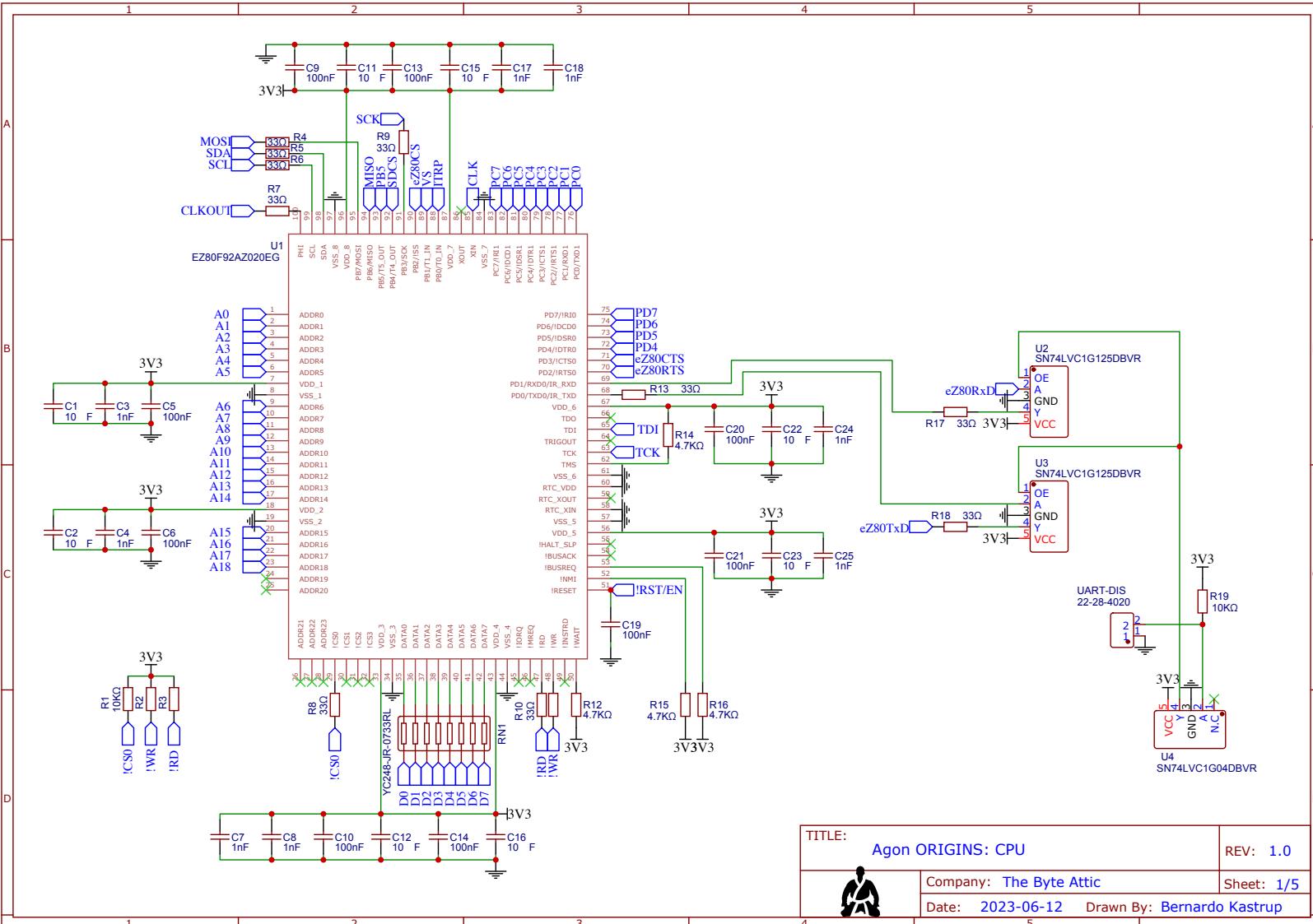
Default settings for the jumpers

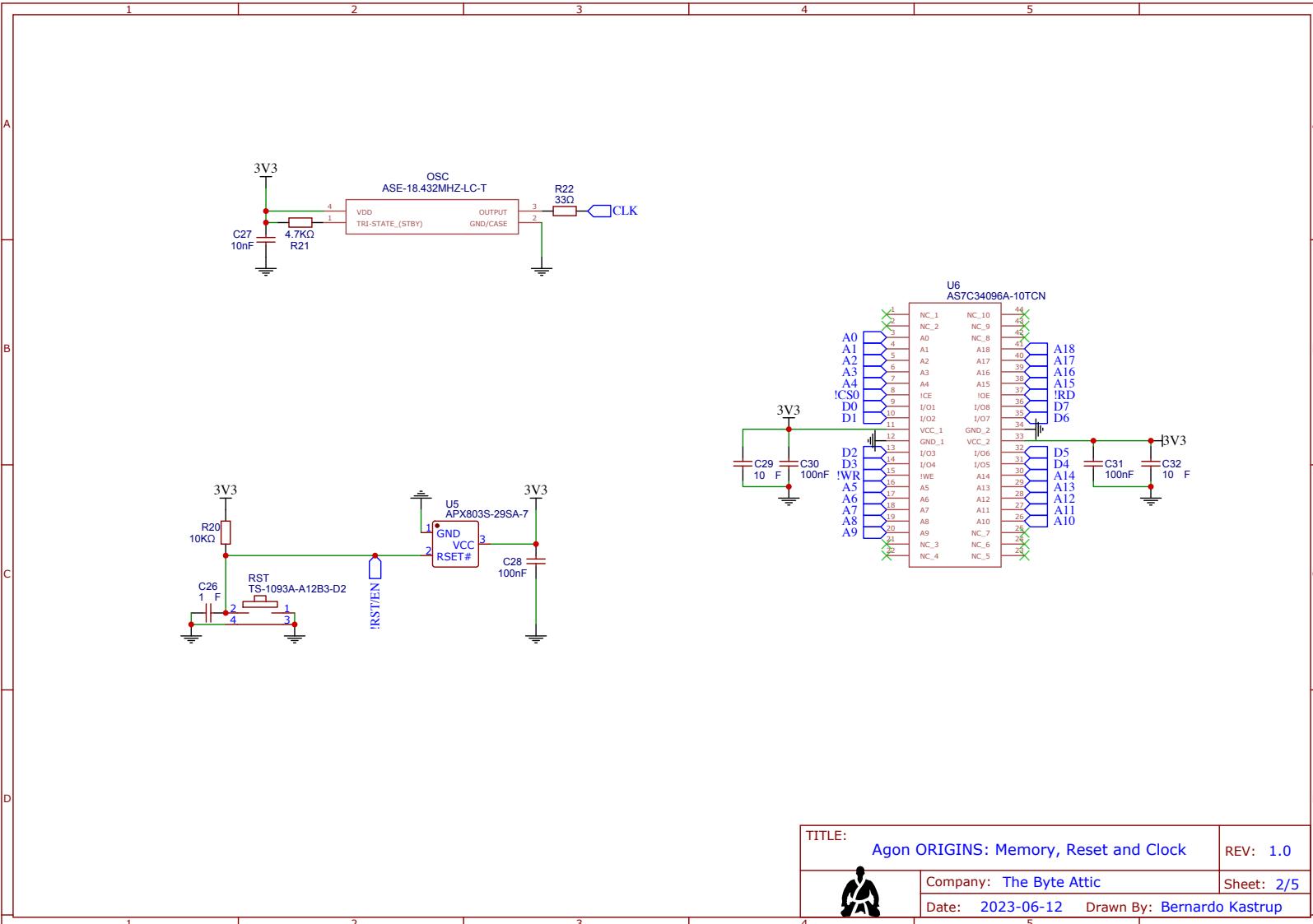
- **BUZ-EN**: shunted (if shunted, enables the buzzer; remove the shunt if the buzzer annoys you)
- **UART-DIS**: open (if shunted, disables communication between eZ80 and ESP32, which may be useful if you are having difficulties uploading firmware into the ESP32; *remember to remove the shunt after programming, or Agon will not work!*)
- **ESP-PROG**: shunted (this shunt should only be removed if the ESP32 is spontaneously going into programming mode during operation; this behavior has never been observed as of this writing)

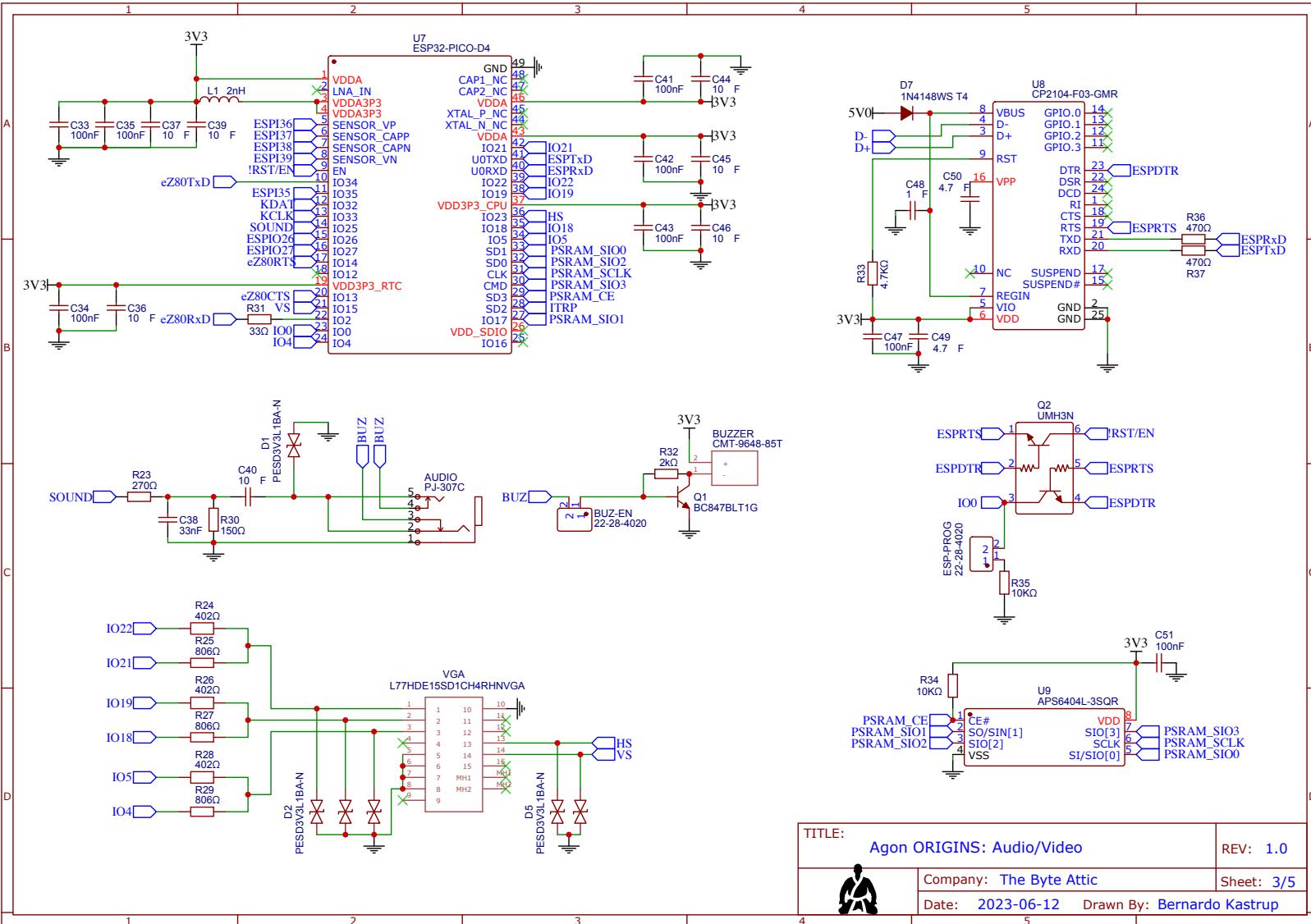


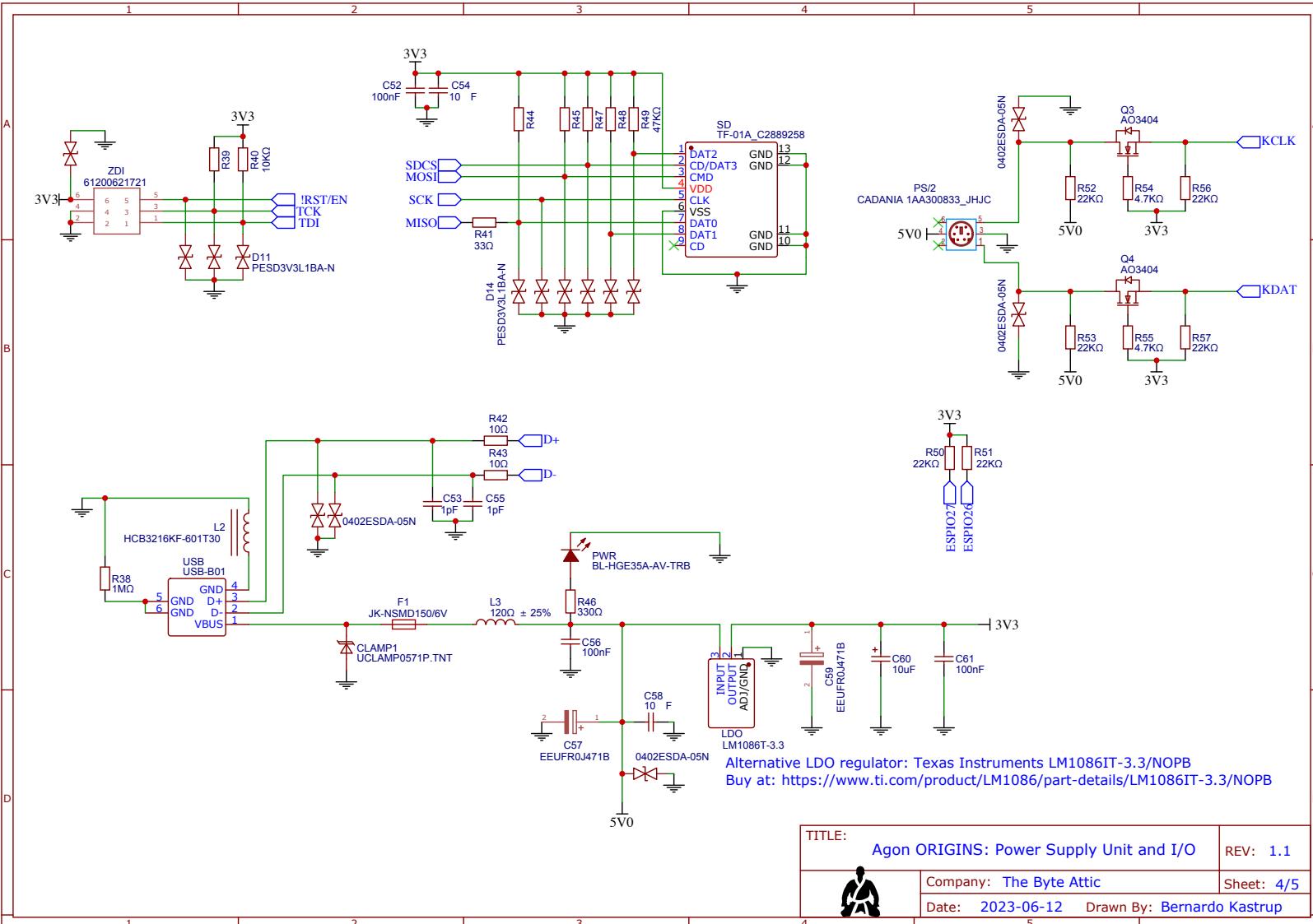
The Byte Attic's
Agon
light™
Schematics

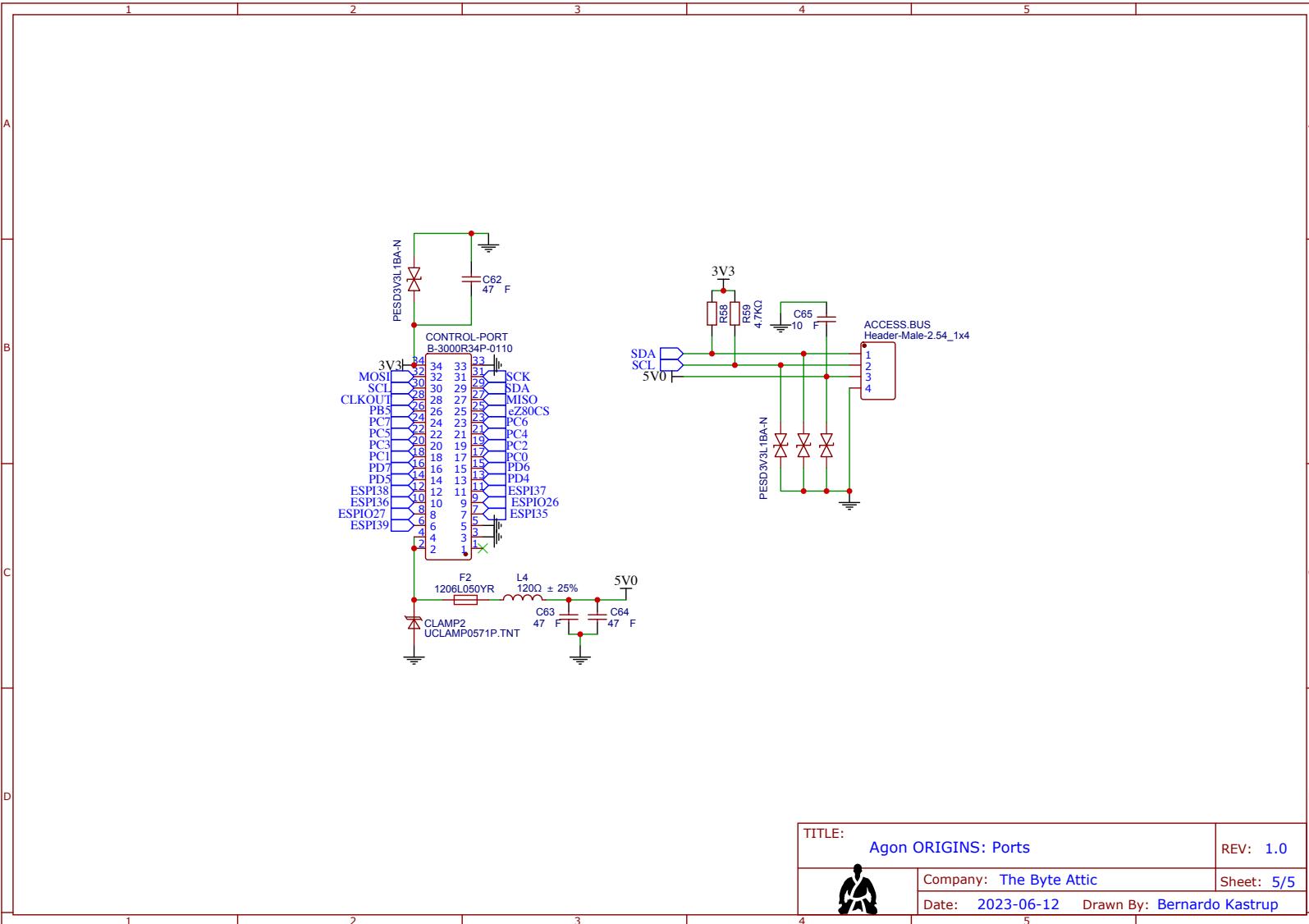










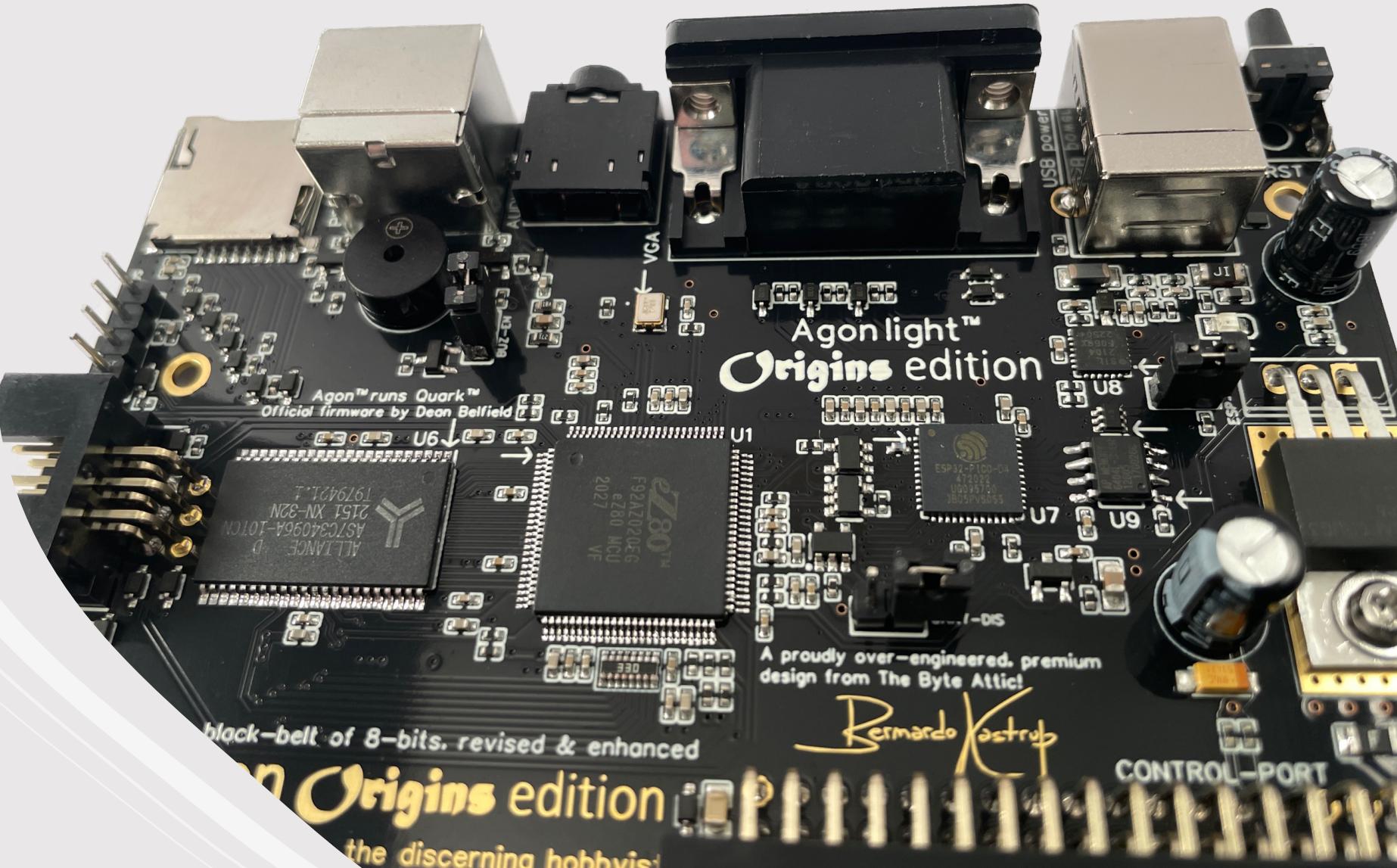




The Byte Attic's

Agon light™

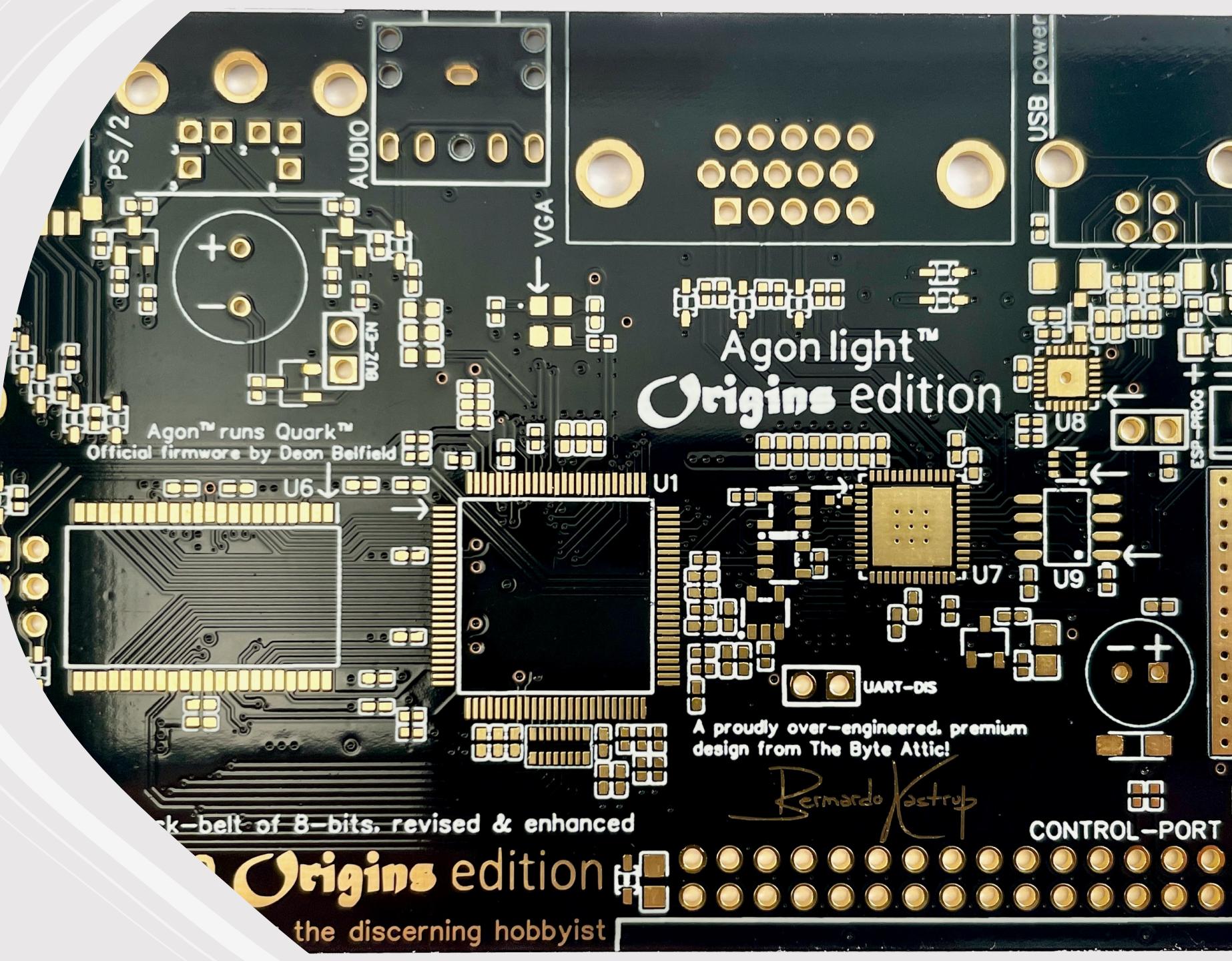
Bill of
Materials



ID	Name	Designator	Quantity	Manufacturer Part
1	APX8035-295A-7	U5	1	APX8035-295A-7
2	33Q	R41,R5,R6,R7,R8,R9,R10,R11,R13,R17,R18,R22,R31,R4	14	0402WGF1301TCE
3	D1	D12	1	0402ESDA-05N
4	0402ESDA-05N	D13,D26,D21,D22,D26	5	0402ESDA-05N
5	PESD3V3L1BA-N	D1,D2,D3,D4,D5,D6,D8,D9,D10,D11,D14,D15,D16,D17,D18,D19,D23,D24,D25	19	PESD3V3L1BA-N
6	CADANIA_1AA300833_JHC	P5/2	1	
7	A5E-18.432MHz-LC-T	OSC	1	A5E-18.432MHz-LC-T
8	47KΩ	R44,R45,R47,R48,R49	5	0402WGF4702TCE
9	4.7KΩ	R12,R14,R15,R16,R21,R33,R54,R55,R58,R59	10	0402WGF4701TCE
10	10KΩ	R1,R2,R3,R19,R20,R34,R35,R39,R40	9	0402WGF1002TCE
11	270Ω	R23	1	0603WA1270075E
12	40Ω	R24,R25,R28	3	PTR0402B402RN9
13	806Ω	R25,R27,R29	3	TC022SB8060TCE
14	150Ω	R30	1	0603WA150075E
15	2KΩ	R32	1	0402WGF2001TCE
16	47Ω	R36,R37	2	0402WGF47001TCE
17	1MΩ	R38	1	0402WGF1004TCE
18	100Ω	R42,R43	2	0402WGF1001TCE
19	33Ω	R46	1	0402WGF33001TCE
20	22KΩ	R50,R51,R52,R53,R56,R57	6	0402WGF2202TCE
21	B-3000R34P-0110	CONTROL-PORT	1	B-3000R34P-0110
22	10μF	C1,C2,C11,C12,C15,C22,C23,C29,C32,C36,C37,C39,C40,C44,C45,C46,C54,C58,C65	20	CL05A106M050UNIC
23	1mF	C3,C4,C7,C8,C17,C18,C24,C25	8	0402B12K500NT
24	100nF	C5,C6,C9,C10,C13,C14,C19,C20,C21,C28,C30,C31,C33,C34,C35,C41,C42,C43,C47,C51,C52,C56,C	23	CL05B104K050NNNC
25	1μF	C26	1	CL21B10K5FNNNE
26	10nF	C27	1	CL05B10K5B5NNNC
27	33nF	C38	1	CL10B33K5B8NNNC
28	1μF	C48	1	CL05A10K50K05NNNC
29	4.7μF	C49,C50	2	CL05A1075M5PNRNC
30	10F	C53,C55	2	0402CG15RC500NT
31	ELFR0471B	C57,C59	2	EEU-FR0471B
32	10μF	C60	1	TAJA106K016RNJ
33	47μF	C63,C65,C64	3	CL21A475M50YHNNNE
34	YC248-JR-0733RL	RN1	1	YC248-JR-0733RL
35	1N4148WS T4	D7	1	1N4148WS T4
36	2mH	L1	1	A5C-0402-2m0J-T
37	HCB3216K-601T30	L2	1	HCB3216K-601T30
38	32.18Ω ±2.5%	L3,L4	2	BLM18KG121TN1D
39	BC847BLT1G	Q1	1	BC847BLT1G
40	UMH3N	Q2	1	UMH3N
41	A03404	Q3,Q4	2	A03404
42	SN74LVC1G125DBVR	U2,U3	2	SN74LVC1G125DBVR
43	SN74LVC1G04DBVR	U4	1	SN74LVC1G04DBVR
44	A57C34096A-10TCN	U6	1	A57C34096A-10TCN
45	ESP32-PICO-D4	U7	1	ESP32-PICO-D4
46	CP2104-F03-GMR	U8	1	CP2104-F03-GMR
47	AP56404L-35QR-SN	U9	1	AP56404L-35QR-SN
48	USB-B01	USB	1	AP56404L-35QR-SN
49	Header-Male-2.54-1x4	ACCESS-BUS	1	120S-1*P1 L=11.5MMgold-plated black
50	Pj-307C	AUDIO	1	Pj-307C
51	22-28-4020	BUZZER-EN,ESP-PROG,UART-DUS	3	22-28-4020
52	CMT-9648-85T	BUZZER	1	
53	UCLAMPMP571P-TNT	CLAMP1,CLAMP2	2	UCLAMPMP571P-TNT
54	JK-NSMD150/6V	F1	1	JK-n5MD150/6V
55	1206L050YR	F2	1	1206L050YR
56	LM1086T-3.3	LOO	1	LM1086T-3.3
57	BL-HGE35A-AV-TRB	PWR	1	BL-HGE35A-AV-TRB
58	TF-01A, C2889258	SD	1	TF-01A
59	EZ80F929A2020EG	U1	1	EZ80F929A2020EG
60	L7710DE15SD10HERRINGVA	VGA	1	
61	61200621721	ZDI	1	
62	T5-1093A-A12B3-D2	RST	1	TS-1093A-A12B3-D2



The Byte Attic's
Agon
light™
PCB



PCB dimensions (diagram *not* to scale)

