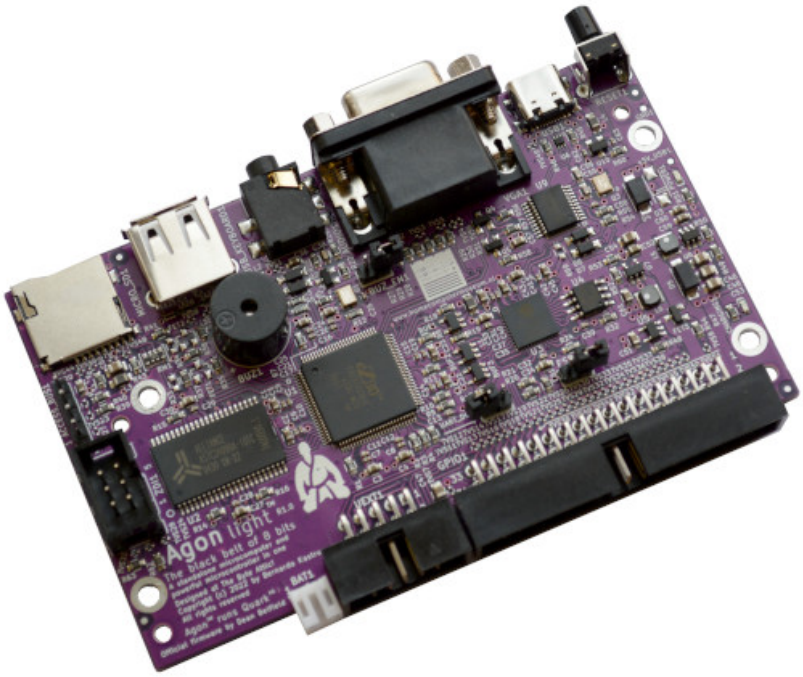


AgonLight2

User Manual

Document revision 1.4., July 2023

www.olimex.com

Table of Contents

INTRODUCTION

What is AgonLight2?

AgonLight2 is a re-design of the original AgonLight eZ80F92 retro computer, designed as a hobby project by Bernardo Kastrup;

AgonLight2 is a complete Single Board Computer (SBC) with VGA display output, PS2 keyboard, and SD card (acting as an external disk). This means that to write and run programs you do not need an external computer like Arduino boards do;

AgonLight2 is BASIC-programmable microcomputer;

AgonLight2 has a GPIO connector with GPIO ports, I2C, SPI, and more signals, which are accessible and can be used to interact with additional hardware (sensors, expansions, etc);

AgonLight and AgonLight2 are Open Source Hardware and officially OSHWA certified;

All CAD source files are available at AgonLight and AgonLight2 GitHub repositories. They are released under CERN Open Hardware License Version 2 – Strongly Reciprocal and allow users to learn, study, edit, modify, produce, and sell same or derivative products based on these designs. The only requirements is to open source their work under the same licensee.

The firmware of AgonLight is written by Dean Belfield and reproduces BBC Basic for Z80. The firmware is also Open Source Software and available on GitHub. The same firmware works on AgonLight2 without any modifications.

AgonLight2 is RoHS, REACH, CE and UKCA compliant.

Order codes and links for AgonLight2 and accessories:

AgonLight2	Single board BBC BASIC Z80 Retro style Computer
MICRO-SD-CLASS	You need a microSD card to store BASIC and software (demos, examples)
AgonLight2-Proto	DIY expansion kit with big proto area; compatible with AgonLight2
BOX-AGONLIGHT2-BLACK	Metal box for AgonLight2
CABLE-USB-A-C-1M	USB-C data and power cable
BATTERY-LIPO1400mAh	Li-po battery 3.7V 1400mAh – note that these batteries can be shipped only by ground so we can deliver only to EU destinations
USB-KEYBOARD-PS2	USB keyboard which supports PS2 and can be used for AgonLight2
BREADBOARD-1	Breadboard for experimenting with AgonLight2
JW-200x10-FM	10 pieces female-male jumper wires for breadboarding
JW-200x10	10 pieces male-male jumper wires for breadboarding
JW-200x10-FF	10 pieces female-female jumper wires for breadboarding
UEXT modules	Different sensors, relays, LCDs, RTC, GSM, GPS etc accessories which can be connected to AgonLight2 UEXT connector
MOD-Wii-UEXT-NUNCHUCK	3 axis joystick, two buttons game controller with UEXT connector

The differences between AgonLight and AgonLight2:

AgonLight2 has these differences from the original design:

- KiCad is used for the CAD, instead of EasyEDA. KiCad is open source software – free to download and use, it is a more fitting CAD tool for an Open Source Hardware project;
- USB-C connector for power supply;
- DCDC power converter with up to 2A output instead of linear power converter;
- Li-Po battery backup power supply, the battery acts like a UPS and will power the computer even if the USB-C gets disconnected. The battery is charged automatically when USB or external power supply is present;
- USB connector for keyboard (but the keyboard must be PS2 compatible!!!);
- SRAM routed with 40 ohm impedance as per part datasheet;
- Fixed some wrong signal naming on ESP32-PICO-D4 GPU co-processor;
- Fixed GPIO and programming connector pin ordering;
- Replaced the naked header 32-pin connector with a plastic boxed 34-pin connector following the same layout and adding two additional signals Vbat and Vin which allow AgonLight2 to be powered by this connector too;
- Added a UEXT connector (https://www.olimex.com/Products/Modules/) which allows AgonLight2 to be connected to: temperature sensors, environmental air quality sensors, pressure, humidity, gyroscope, light, RS485, LCDs, LED matrix, relays, Bluetooth, Zigbee, Lora, GSM, RFID reader, GPS, Pulse, EKG, RTC etc;
- Added 4 grounded mount holes with 3mm diameter, but also kept the original 2.5 mm mount holes.

First time start up or how to prepare the SD card and boot

Important! Notice that AgonLight2 comes completely assembled, tested, and programmed with firmware. The only thing you would need to do is prepare a microSD card with BBC basic and some examples, then connect your display and peripherals and apply power supply. The board comes without micro SD card which has to be purchased separately! You might need extra SD card writer.

1. Open a web browser and navigate to:

https://github.com/OLIMEX/AgonLight2

2. Click on: “Code” → Download ZIP (in “Local” tab);

3. Extract the downloaded ZIP file in your PC;

4. Enter the folders “SOFTWARE” and then “1-SD-card-contents”, then copy the contents of the folder;

5. Insert a class-10 microSD card in your PC (formatted in FAT32 and with a size of maximum 32GB);

6. Paste the contents of the folder “1-SD-card-contents” which you’ve copied in step 4, onto the microSD card. Files should be placed directly into main folder of the card;

7. Eject the microSD card from your PC;

8. Insert the microSD card into the board;

9. Connect a VGA monitor and a PS/2 keyboard (or a PS/2-compatible USB keyboard,

via a PS/2 adapter) to the AgonLight2 unit;

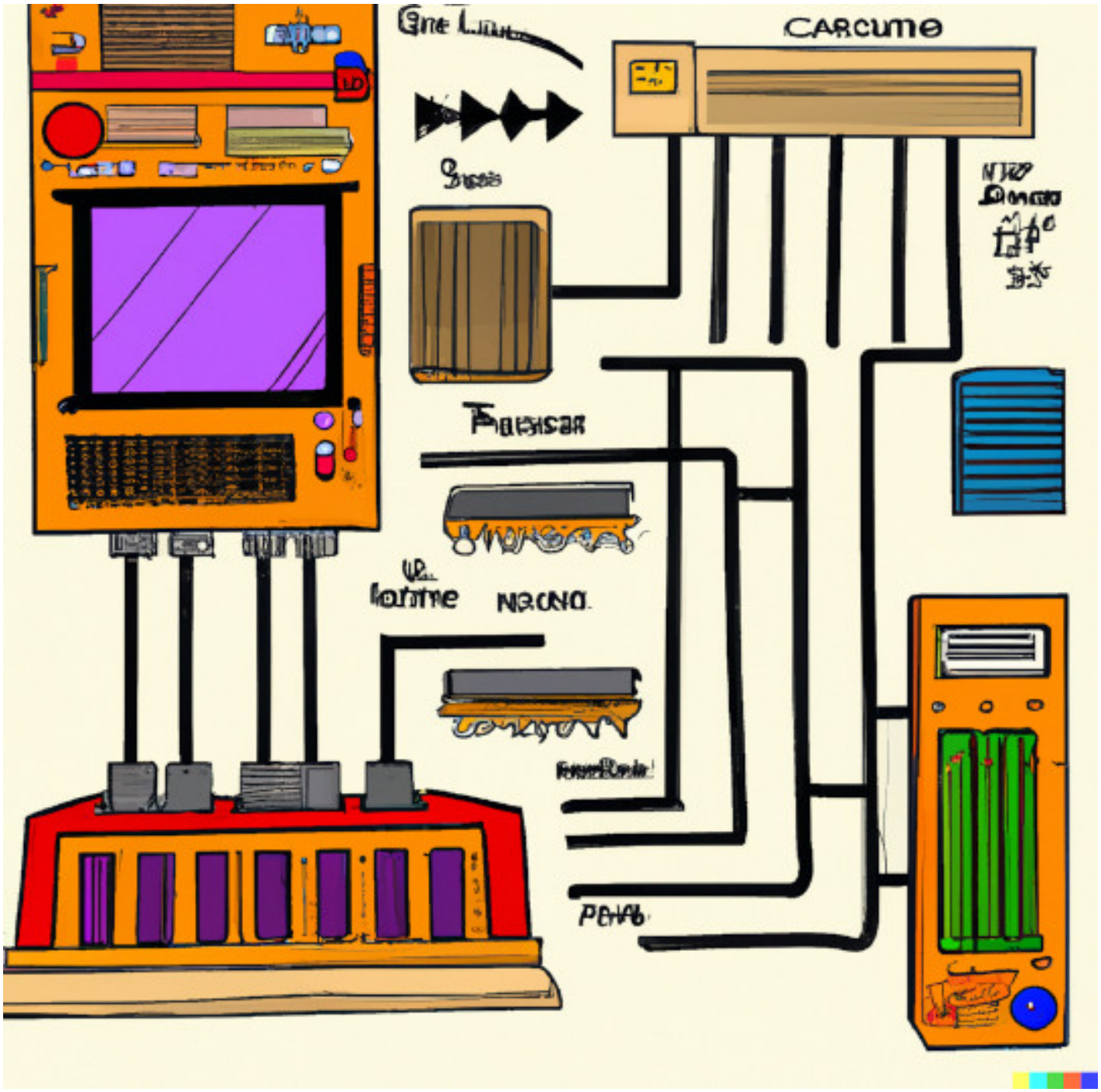
10. Turn the AgonLight2 board on by connecting it to the USB cable.

More detailed info about the software can be found further below in the document.

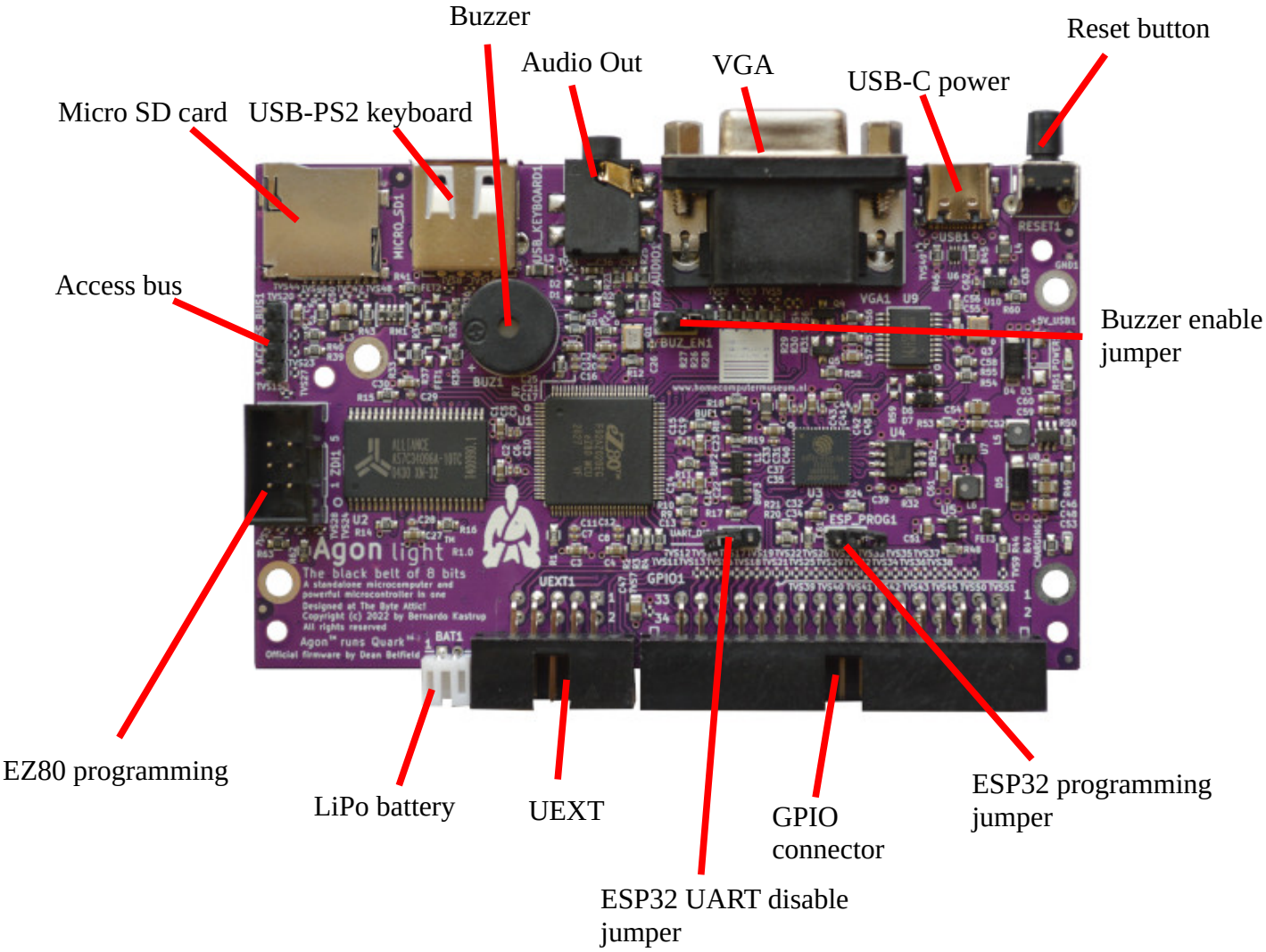
AgonLight references documents:

Links to original AgonLight manuals: AgonLight Hardware Manual, QuickStart Guide, and Firmware Installation Guide.

HARDWARE

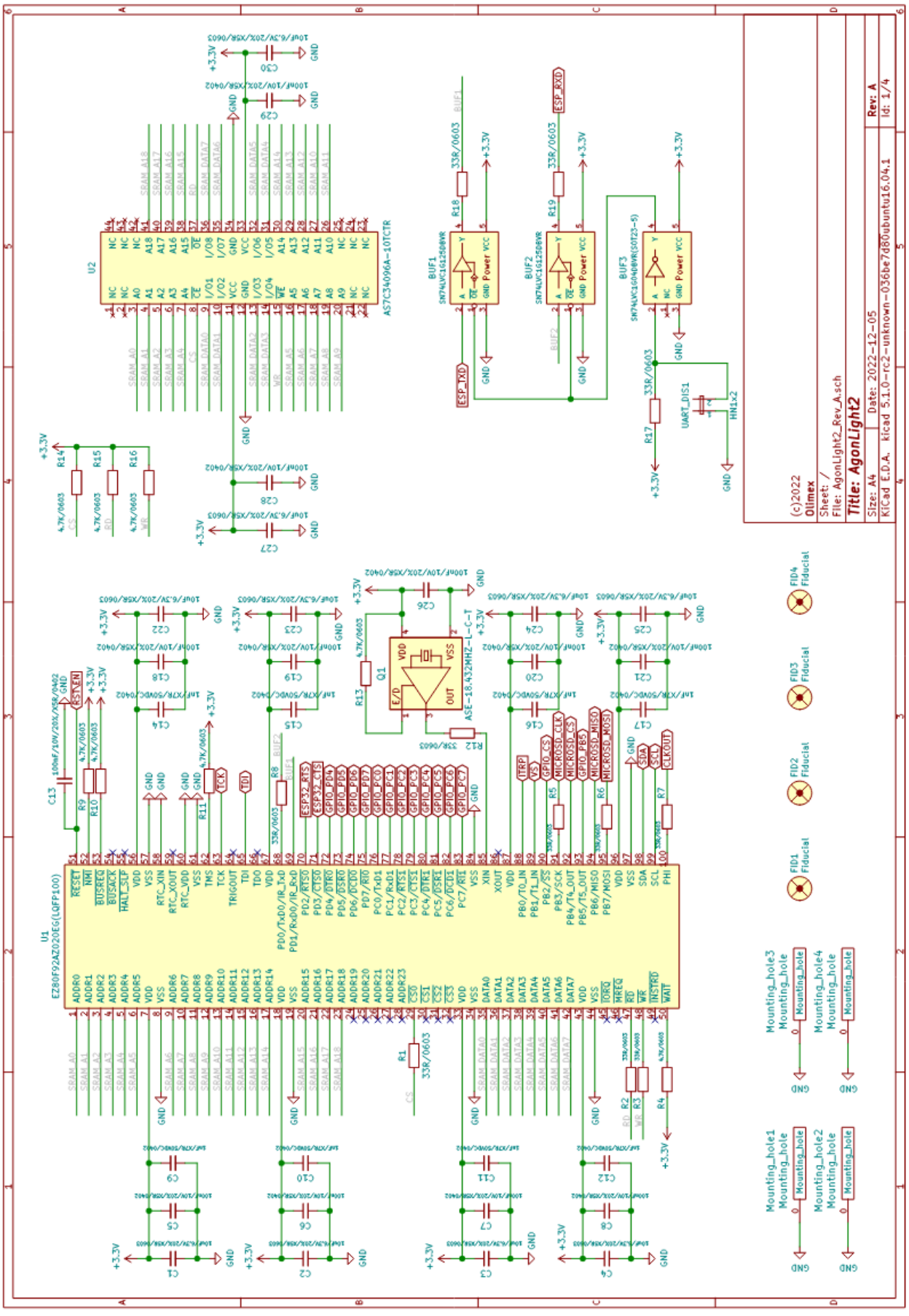


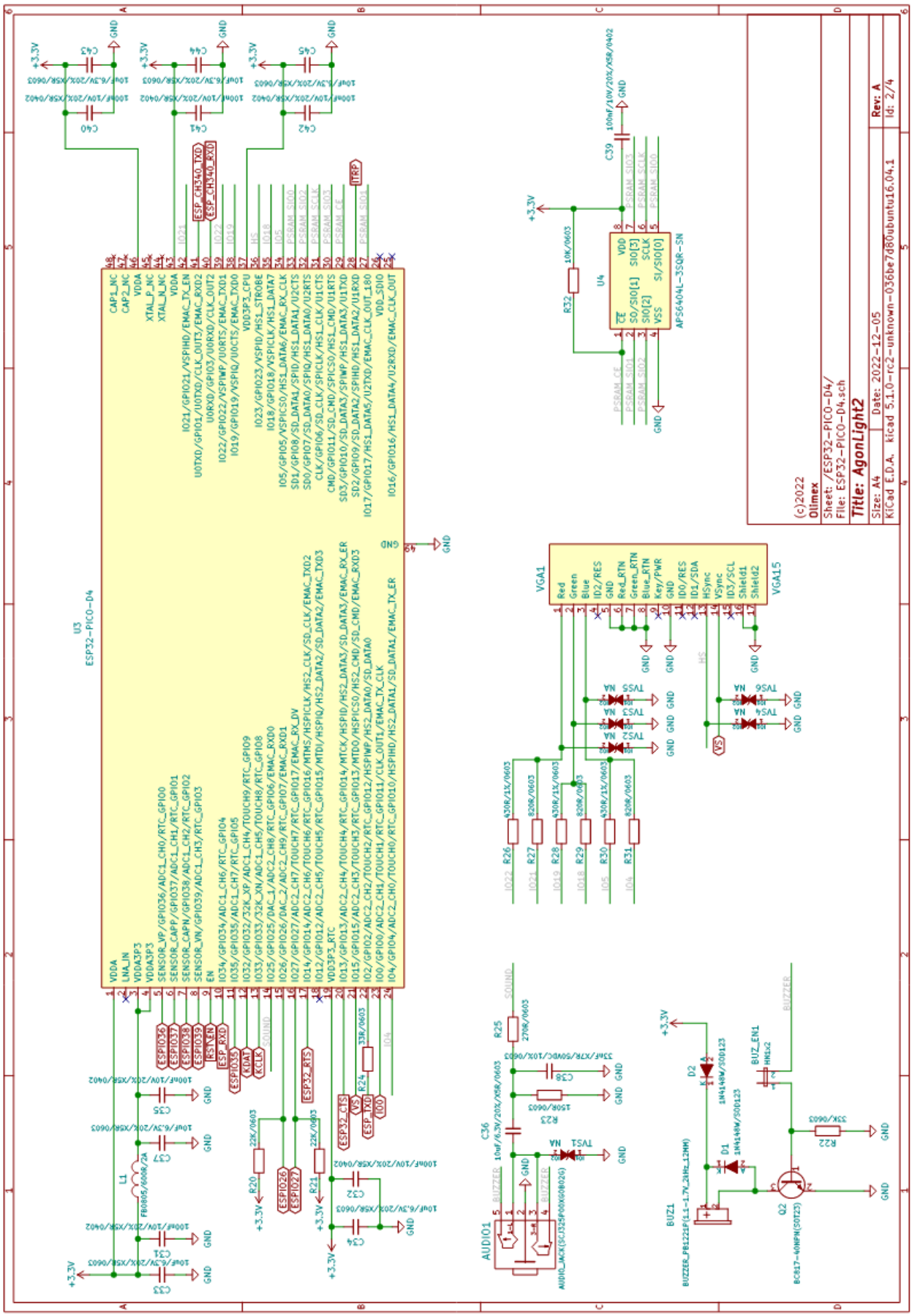
AgonLight2 layout:

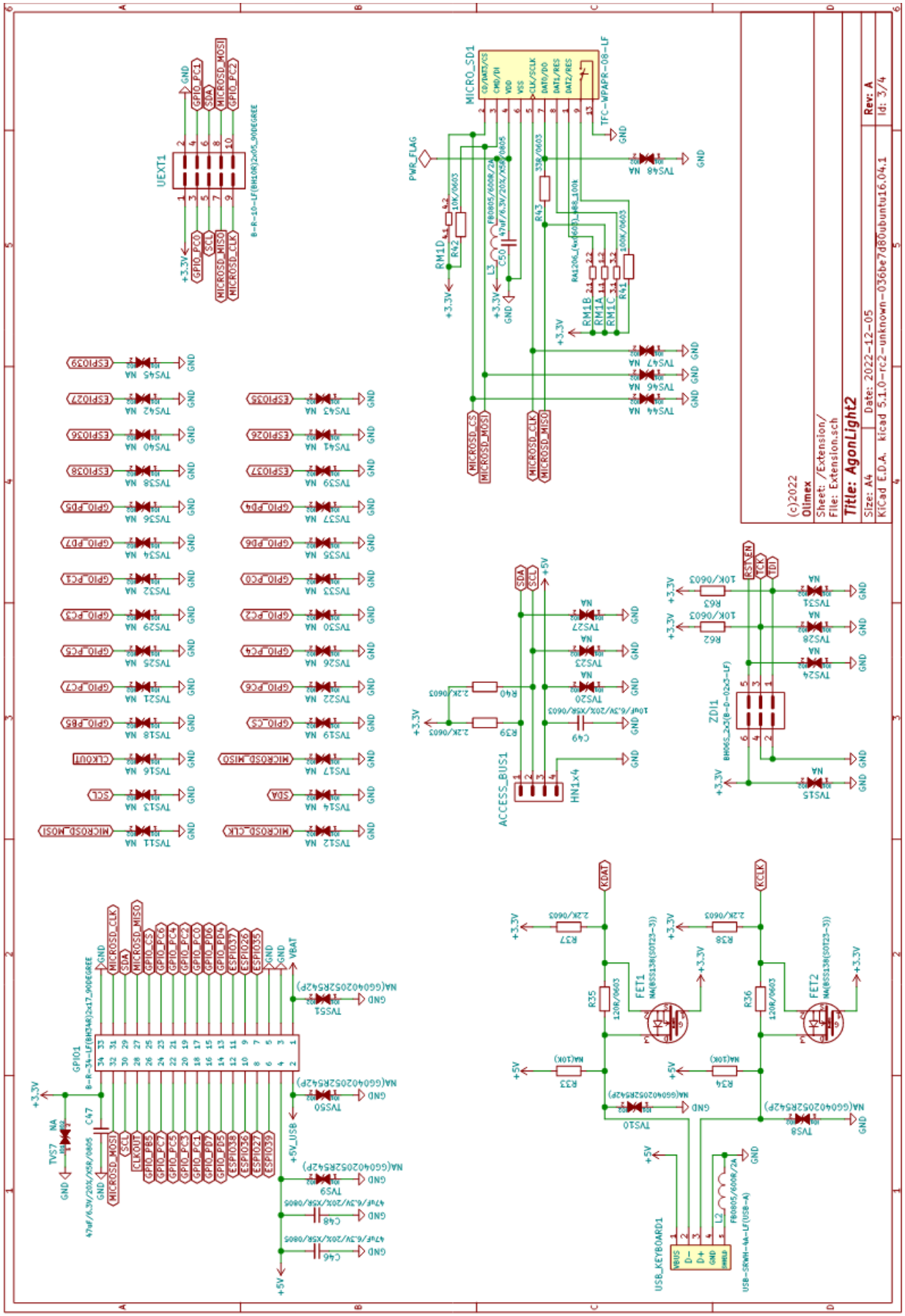


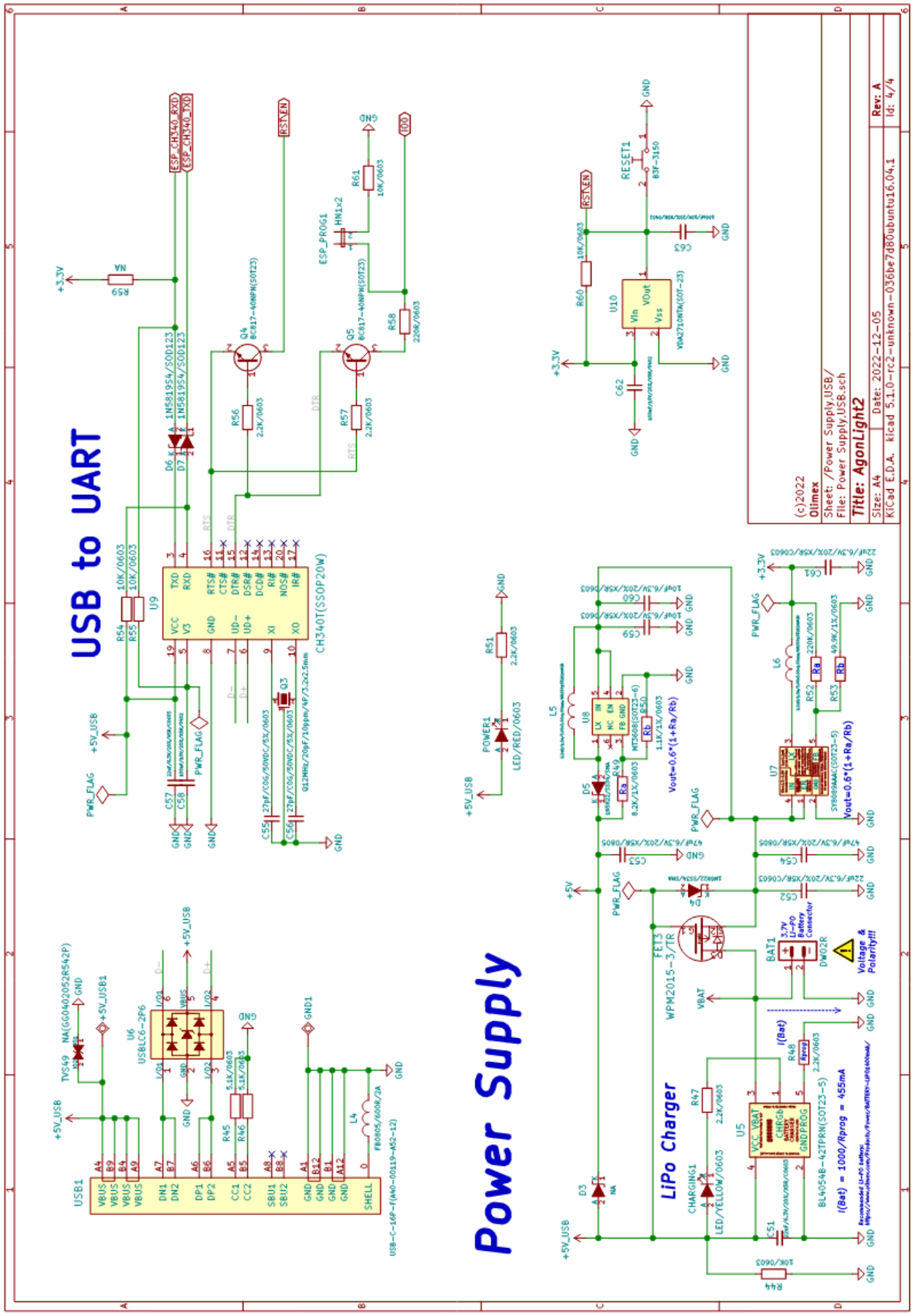
AgonLight2 schematic:

The current AgonLight2 schematic is available for download and print on GitHub


(c)2022
Olimex
Sheet: /
File: AgonLight2_Rev_A.sch
Title: AgonLight2
Size: A4
Date: 2022-12-05
Rev: A
KiCad E.D.A. kicad 5.1.0-rc2-unknown-036be7d80ubuntu16.04.1
Id: 1/4


(c)2022
Olimex
Sheet: /ESP32-PICO-D4/
File: ESP32-PICO-D4.sch
Title: AgonLight2
Size: A4
Date: 2022-12-05
Rev: A
KiCad E.D.A. kicad 5.1.0-rc2-unknown-036be7d80ubuntu16.04.1
Id: 2/4
U3
ESP32-PICO-D4
U4
APS6404L-3SQR-SN
VGA1
VGA15
AUDIO1
BUZ1
D1
D2
Q2
BUZ_EN1
L1
ESP_CH340_TXD
ESP_CH340_RXD
ESP_RXD
ESP_TXD
ESP32_RTS
ESP32_CTS
PSRAM_CE
PSRAM_SIO0
PSRAM_SIO1
PSRAM_SIO2
PSRAM_SIO3
PSRAM_SCLK
ESPIO36
ESPIO37
ESPIO38
ESPIO39
ESPIO35
ESPIO26
ESPIO27
RSTKEY
KDAT
KCLK
SOUND
BUZZER
TRIP
GND
+3.3V


Title: AgonLight2
Size: A4
Date: 2022-12-05
Rev: A
KiCad E.D.A. kicad 5.1.0-rc2-unknown-036be7d80ubuntu16.04.1
Id: 3/4
File: Extension.sch
Sheet: /Extension/
(c)2022
Olimex
MICRO_SD1
TFC-WPAPR-08-LF
UEXT1
PWR_FLAG
ACCESS_BUS1
HN1x4
USB_KEYBOARD1
GPIO1
FET1
FET2
ZD1
MICROSD_CS
MICROSD_MOSI
MICROSD_CLK
MICROSD_MISO
KDAT
KCLK
+3.3V
+5V
+5V_USB
GND
VBAT


USB to UART
Power Supply
LiPo Charger
CH340T(SSOP20W)
WPM2015-3/TR
BL4054B-42TPRN(SOT23-5)
MT3608(SOT23-6)
SY8089AAAC(SOT23-5)
USBLC6-2P6
USB-C-16P-F(A40-00119-A52-12)
(c)2022
Olimex
Sheet: /Power Supply,USB/
File: Power Supply,USB.sch
Title: AgonLight2
Size: A4
Date: 2022-12-05
Rev: A
KiCad E.D.A. kicad 5.1.0-rc2-unknown-036be7d80ubuntu16.04.1
Id: 4/4

AgonLight2 power supply and consumption:

AgonLight2 can be powered by 3 sources:

- USB-C connector;
- Li-Po battery connector;
- GPIO.pin2 (5V_USB) note that this wire is connected to the 5V of the USB so when you power AgonLight2 from GPIO.pin2 – you should not connect have USB cable attached at the same time!!!

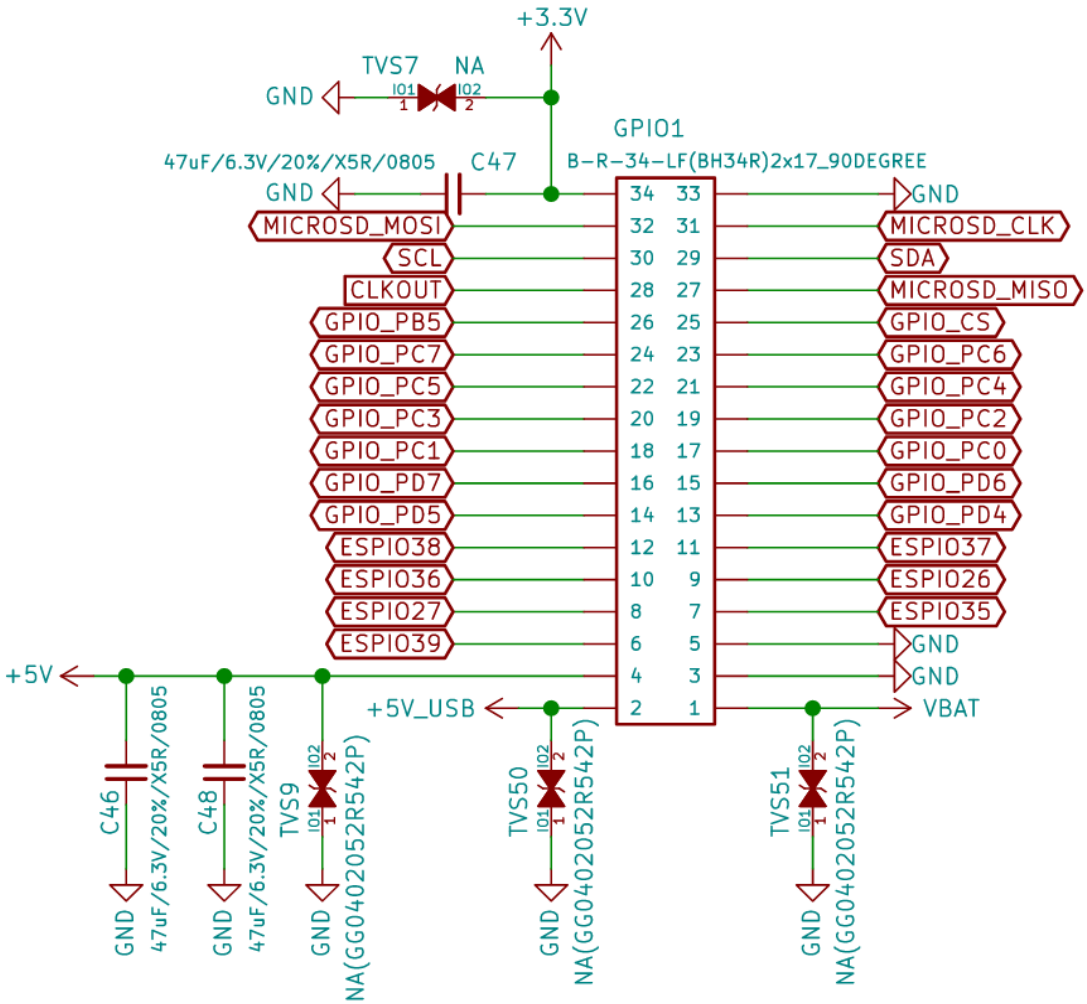
The power consumption of AgonLight2 is approximately 130mA without a keyboard and around 180mA with a keyboard connected.

If a Li-Po battery is connected it will get charged automatically when power supply is attached with up to 100mA.

When the LiPo battery is attached and external power supply is missing internal DCDC step-up converter and switching circuit automatically powers AgonLight2 from the battery. 1400mAh battery will provide about 8 hours of stand-alone operation.

+ The LiPo battery connector is JST 2.0 mm connector and with Olimex's battery polarity. If you use batteries from other manufacturers please make PLUS and MINUS are connected properly as you may damage the board!!!

GPIO connector:



Pin.1 is connected to the Li-Po battery PLUS you can connect external battery on this pin or to use battery voltage to external circuits.

Pin. 2 is 5V power supply connected to USB-C +5V signal, you can power the board from this signal if the USB-C is not connected. It must be regulated 5V power supply, applying more than 5V will damage the board.

Pin.3, **Pin.5**, **Pin.33** are GND

Pin.34 is +3.3V output capable to source up to 2A note that 200mA are used by AgonLight2

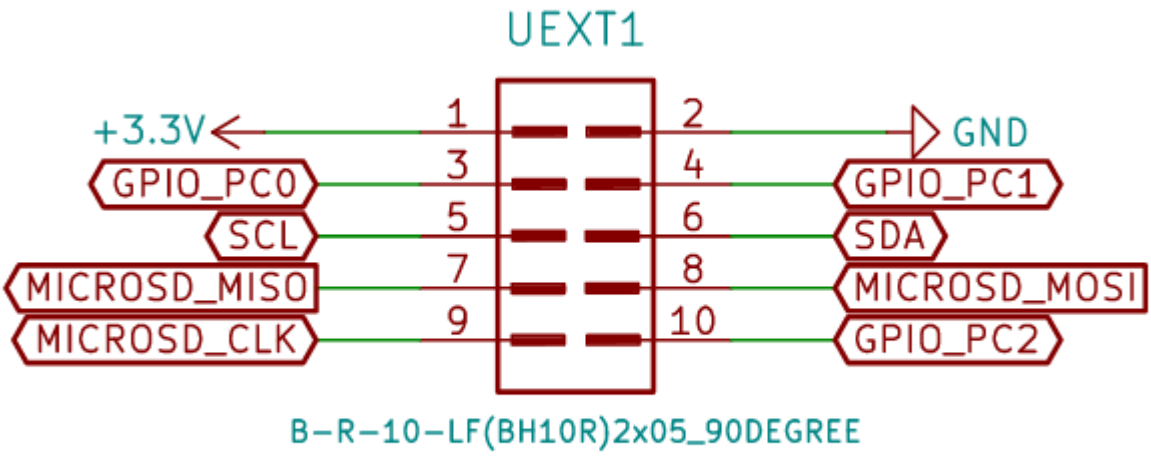
Pin.4 is +5V output capable to source up to 2A (1.8A + AgonLight2 0.2A), it's backed by LiPo UPS so even if there is interruption on power supply if LiPo battery is attached there will be 5V on this pin.

All GPIOs are operating at +3.3V levels. This means you should not connect signals above 3.3V on these ports as this will damage the board.

UEXT connector:

UEXT connector stands for Universal EXTension connector and contain +3.3V, GND, I2C, SPI, UART signals:

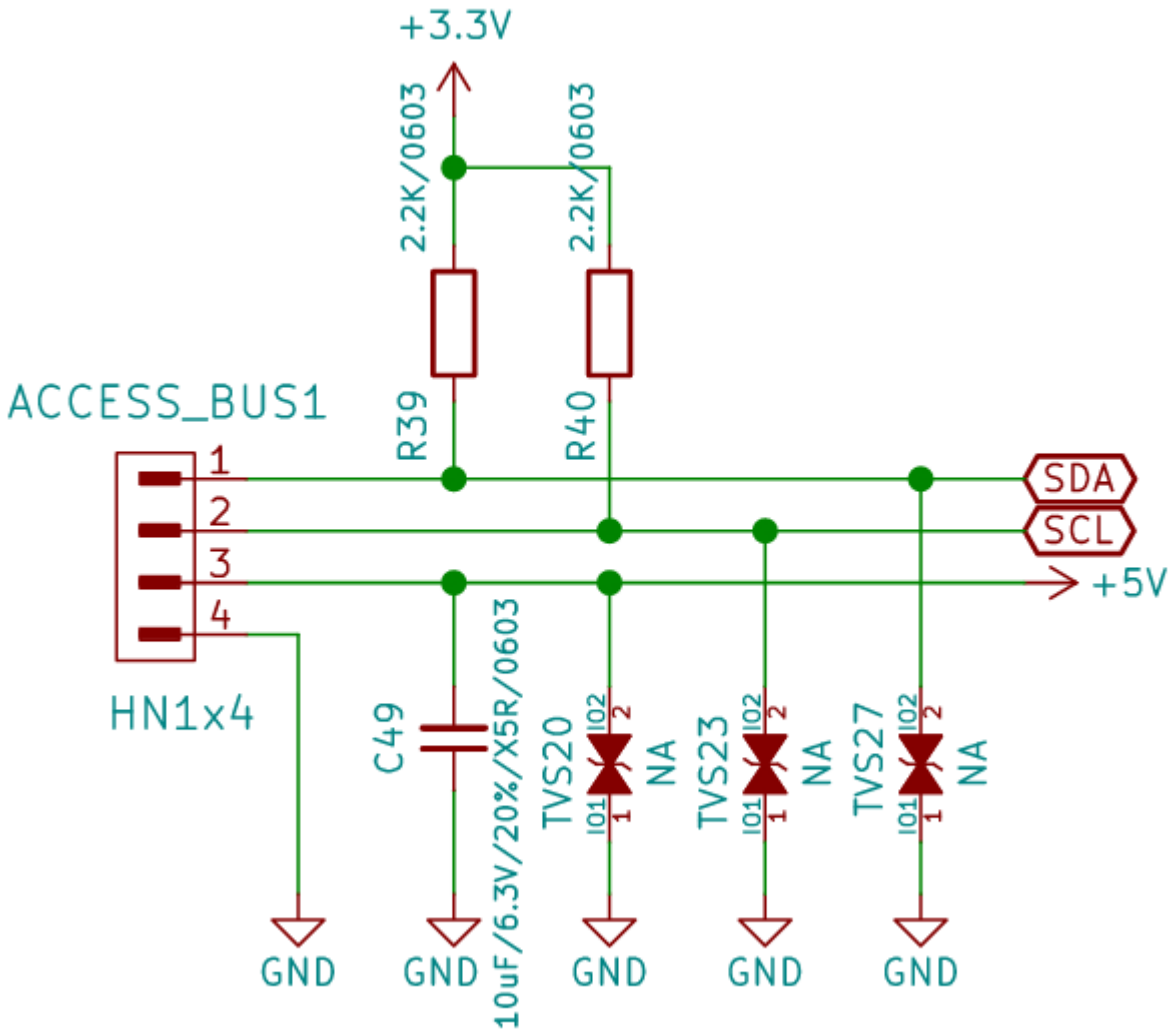
UEXT is 0.1” 2.54mm step boxed plastic connector. All signals are with 3.3V levels.



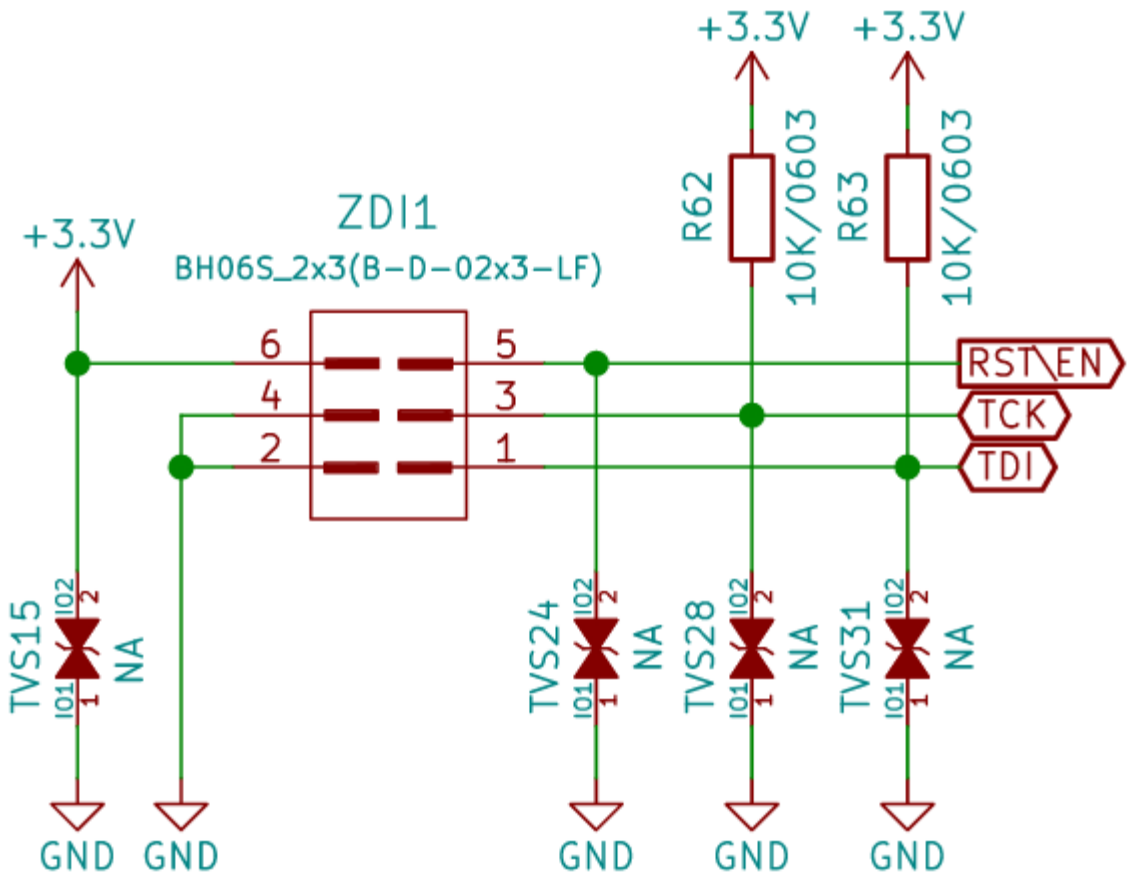
Olimex has developed number of MODULES compatible with this connector. There are temperature, humidity, pressure, magnetic field, light sensors. Modules with LCDs, LED matrix, relays, Bluetooth, Zigbee, WiFi, GSM, GPS, RFID, RTC, EKG, etc.

Access bus connector:

This is I2C and power supply with 3.3V levels.



eZ80 programming connector:



This connector is used for initial programming of eZ80 microcontroller. AgonLight2 comes with pre-programmed firmware so you do not need to use this connector nor programmer for it.

If you want to develop your own firmware Jeroen Venema wrote flash utility which allows you to flash new firmware without need of external programmer.

Jumpers:

Typically you don’t need to change the positions of the jumpers to use the board.

BUZ_EN1 – buzzer enable jumper – when closed the buzzer plays the sound commands.

Default state: Open (Buzzer disabled)

ESP_PROG1 – ESP programming enabled when closed ESP32 enters bootloader mode at reset and can be programmed over the USB connector.

Default state: Open (ESP programming handled by software)

UART_D – UART disable – when closed the UART connection between eZ80 and ESP32 is disconnected. This is necessary when ESP32 is programming.

Default state: Open (communication between eZ80 and ESP32 enabled)

SOFTWARE:

Official AgonLight Firmware

The three modules of AgonLight official firmware, called Quark™, can be found here:

- Quark™ BBC BASIC – BBC Basic interpreter
- Quark™ MOS - command line machine operating system, similar to CP/M or DOS
- Quark™ VDP - The VDP is a serial graphics terminal that takes a BBC Basic text output stream as input. The output is via the VGA connector on the Agon.

Extra info about AgonLight2 firmware:

A properly operating AgonLight2 system has three software components **Quark™ VDP** + **Quark™ MOS** + **Quark™ BBC BASIC**. By default each Olimex AgonLight2 board comes with VDP and MOS already programmed. **You only need to download the basic and the examples to an SD card to start using the AgonLight2 board.** For the BASIC, MOS updater, CP/M and other examples and test programs you need a micro SD card in FAT or FAT32 format and a SD card writer. Micro SD cards are purchased separately.

Notice that certain combinations of versions of VDP, MOS and basic might be incompatible. Meaning that you might need to upgrade or downgrade MOS, VDP, basic in order to run the board. If in doubt, use the versions we provide.

It is always recommended to get the latest files and instructions from the official VDP, MOS, BASIC locations listed below:

• Quark™ BBC BASIC – https://github.com/breakintoprogram/agon-bbc-basic/releases – BBC Basic interpreter

• Quark™ MOS – https://github.com/breakintoprogram/agon-mos/releases – command line machine operating system, similar to CP/M or DOS

• Quark™ VDP – https://github.com/breakintoprogram/agon-vdp/releases – the VDP is a serial graphics terminal that takes a BBC basic text output stream as input. The output is via the VGA connector on the Agon.

However, if you encounter problems use the versions we have provided at our GitHub, we have tested them and confirmed them to work:

https://github.com/OLIMEX/AgonLight2/tree/main/SOFTWARE

The "1-SD-card-contents" folder contains sub-folders with files for the SD card, also mos files that can be flashed via bootloader without programmer (if possible);

The "2-AgonLight2-VDP-XXX" folder contains VDP;

The "3-AgonLight2-MOS-XXX" folder contains MOS that can be flashed via programmer.

BASIC and demo installation

Brand new AgonLight2 board requires only the contents of folder “1-SD-card-contents” to start. Copy the contents from the "1-SD-card-contents" into the root directory of the card.

The content involves:

1) bbcbasic.bin (current version is 1.04, for the latest one check: https://github.com/breakintoprogram/agon-bbc-basic)

2) autoexec.txt which runs the BASIC on restart

3) basic_examples_tests – various BASIC examples/test/games programs

4) Agon-CPM2.2 – contains CPM

5) mos – contains files suitable for updating mos without programmer

Agon Quark VDP installation

For the Agon Quark VDP on the ESP32 chip you have to install IDE, package and libraries:

1) Arduino IDE

1.1) Go to: https://www.arduino.cc/en/software

1.2) While this will most likely work with 2.x.x we recommend you to download 1.8.9 which is shown a little bit below in the section "Legacy IDE (1.8.x)"

1.3) Download, install and run it

2) Install the ESP32 package

2.1) Go to "Main menu --> Preferences" (CTRL+,)

2.2) In the "Additional Boards Manager URLs" add in a new line:
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

2.3) Go to "Main menu --> Tools --> Board --> Boards manager..."

2.4) Look for the package "ESP32" (since there are lots of packages you can filter them by typing "ESP32" in the search bar on the top)

2.5) Install the package (it is tested with 2.0.9v but later are likely to work)

3) Install the FabGL library

3.1) Go to "Main menu --> Sketch --> Include Library --> Manage Libraries..." (CTRL+SHIFT+I);

3.2) Look for library FabGL (just like the packages earlier you can filter all the libraries by typing "FabGL" in the search bar);

3.3) Install it (it is tested with 1.0.8, newer versions of the library may not work);

4) Compile and Upload the firmware (Agon Quark VDP);

4.1) Open the sketch 2-AgonLight2-VDP-104-RC1.ino(the provided one is version 1.04RC1, for the latest one check https://github.com/breakintoprogram/agon-vdp);

4.2) Go to "Main menu --> Tools --> Board --> ESP32 Arduino" and select "ESP32 Dev Module";

4.3) Plug the USB to the AgonLight2 board;

4.4) In Device manager you can see the COM# at which our device has connected;

4.5) Go to "Main menu --> Tools --> Port" select the one ESP32 has connected;

4.6) Go to "Main menu --> Tools --> PSRAM" select Enabled;

4.7) Compile and Upload (every time after opening the project in Arduino the first compilation is VERY slow – be patient).

Agon Quark MOS installation on the eZ80 chip

1) If there is no MOS at all or it's older than 1.02 you must have "ZILOG eZ80F92" programmer and have to install IDE:

1.1) Download "ZDS II - eZ80Acclaim!" from here: https://zilog.com/index.php?option=com_zcm&task=view&soft_id=54&Itemid=74 (tested with 5.3.5 but should work with newer versions)

1.2) Install the IDE and run it

1.3) Go to "Main menu -> File -> Open Project" and navigate to "3-AgonLight2-MOS-104-RC1/MOS.zdsproj" (the provided version is 1.04-RC1, for the latest one check https://github.com/breakintoprogram/agon-mos)

1.4) Build the project (F7)

1.5) Connect the programmer to the ZDI connector on the AgonLight2 board

1.6) Go to "Main menu -> Debug -> Download the code"

1.7) When the download is complete go again to "Main menu --> Debug --> Stop Debugging" (SHIFT + F5)

1.8) Disconnect the programmer from the board

2) If there is MOS 1.02 and you want to update it to a newer version you can still use the method described above but you can also do it from the console on the AgonLight2 board itself without the need of the programmer and IDE

Olimex tested firmware and examples

Can be found in the repository here:

https://github.com/OLIMEX/AgonLight2/tree/main/SOFTWARE/

Disclaimer: Most of the examples located in the repository above are NOT written by Olimex. We have just tested them and confirmed them working at the time of release, then re-hosted them. The repository contains: examples taken from AgonLight repository, solutions to our weekly programming challenge, and other interesting examples we found online.

MOS commands:

LOAD “”

SAVE “”

*CAT *DIR

*CD /

*MKDIR

*BYE *CPM

*ERA file

*ESC [ON OFF]

*EXEC file

*OPT

*REN oldfile ewfile

*SAVE file aaaa bbbb

*SPOOL file

*TYPE file

*| comment

BBC Basic commands reference:

Variables:
X – float
X% - integer
X$ - string
&X – hex

Pressing ESC key breaks the code execution

This is list of the BBC Basic commands click on the command name for more info and use.

ABS	EOR	LOMEM	REPORT
ACS	ERL	MID$	RESTORE
ADVAL	ERR	MOD	RETURN
AND	ERROR	MODE	RIGHT$(
ASC	EVAL	MOVE	RND
ASN	EXP	NEW	RUN
ATN	EXT	NEXT	SAVE
AUTO	FALSE	NOT	SGN
BGET	FN	OFF	SIN
BPUT	FOR	OLD	SOUND
CALL	GCOL	ON ERROR	SPC
CHAIN	GET	ON ERROR OFF	SQR
CHR$	GET()	ON X%	STEP
CLEAR	GET$	OPENIN	STOP
CLG	GOSUB	OPENOUT	STR$
CLOSE	GOTO	OPT	STRING$(
CLS	HIMEM	OR	TAB(X)
COLOUR	IF	PAGE	TAB(X,Y)
COS	INKEY	PAGE	TAN
COUNT	INKEY$	PI	THEN
DATA	INPUT	PLOT	TIME
DEF	INSTR(	POINT(	TO
DEG	INT	POS	TRACE
DELETE	LEFT$	PRINT	TRUE
DIM	LEN	PROC	UNTIL
DIV	LET	PTR	USR
DRAW	LINE	PUT A, X	VAL
ELSE	LIST	RAD	VDU
END	LN	READ	VPOS
ENDPROC	LOAD	REM	WIDTH
ENVELOPE	LOCAL	RENUMBER	
EOF	LOG	REPEAT	

Software access to GPIOs:

eZ80 ports are described in the eZ80-specification.

Port direction register set if the port is INPUT or OUTPUT. Upon RESET all GPIOs are set as Inputs.

Direction registers are 8 bit and each bit corresponds to port signal. If 1 is written on the corresponding port will be set as Input if 0 as Output.

Port B data direction register is at address: 009Bh = 153 decimal

Port C data direction register is at address: 009Fh = 159 decimal

Port D data direction register is at address: 00A3h = 163 decimal

If we read the Port C direction register we can PRINT GET(159) and it should return 255

If we want to set GPIO_PC0 as output we should write 0 in bit 0 with the command PUT 159,254.

Data registers are 8 bit and each bit corresponds to port signal. If 1 is written on the corresponding port will be set 3.3V if 0 is written 0V will be set.

Port B data register is at address: 009Ah = 152 decimal

Port C data register is at address: 009Eh = 158 decimal

Port D data register is at address: 00A2h = 162 decimal

If we read data register with GET() command, for instance PRINT GET(158) it will return the state of Port C.

We can write the data register with PUT command, for instance PUT 158,1 will set GPIO_PC0 to +3.3V and PUT 158,0 will set it to 0V.

Software access to I2C:

TBD

Software access to SPI:

TBD

Software access to VPU:

TBD

DOCUMENT REVISION

Revision 1.0 January 2023

- initial release

Revision 1.1 January 2023

- added “How to prepare the SD card and boot the first time”, taken and edited from “Agon light™ Firmware Installation Guide.pdf”

Revision 1.2 June 2023

- general improvements to the structure of the document; added more info about first steps with the board

Revision 1.3 July 2023

- general improvements to the structure of the document; fixed links; added info about firmware and firmware update

Revision 1.4 July 2023

- bug fixes around names of folders and instructions;