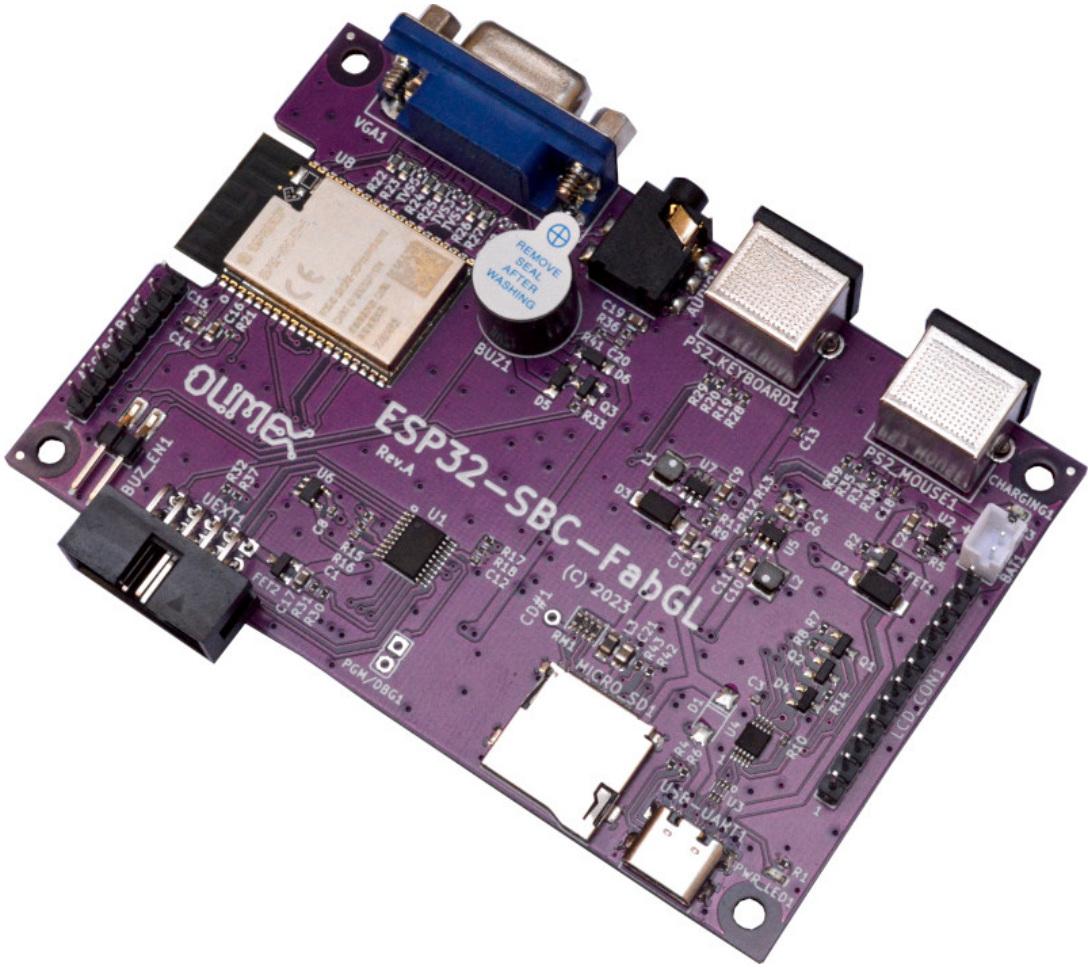


ESP32-SBC-FabGL

User Manual

Document v3.0 **July 2024** **olimex.com**

Table of Contents

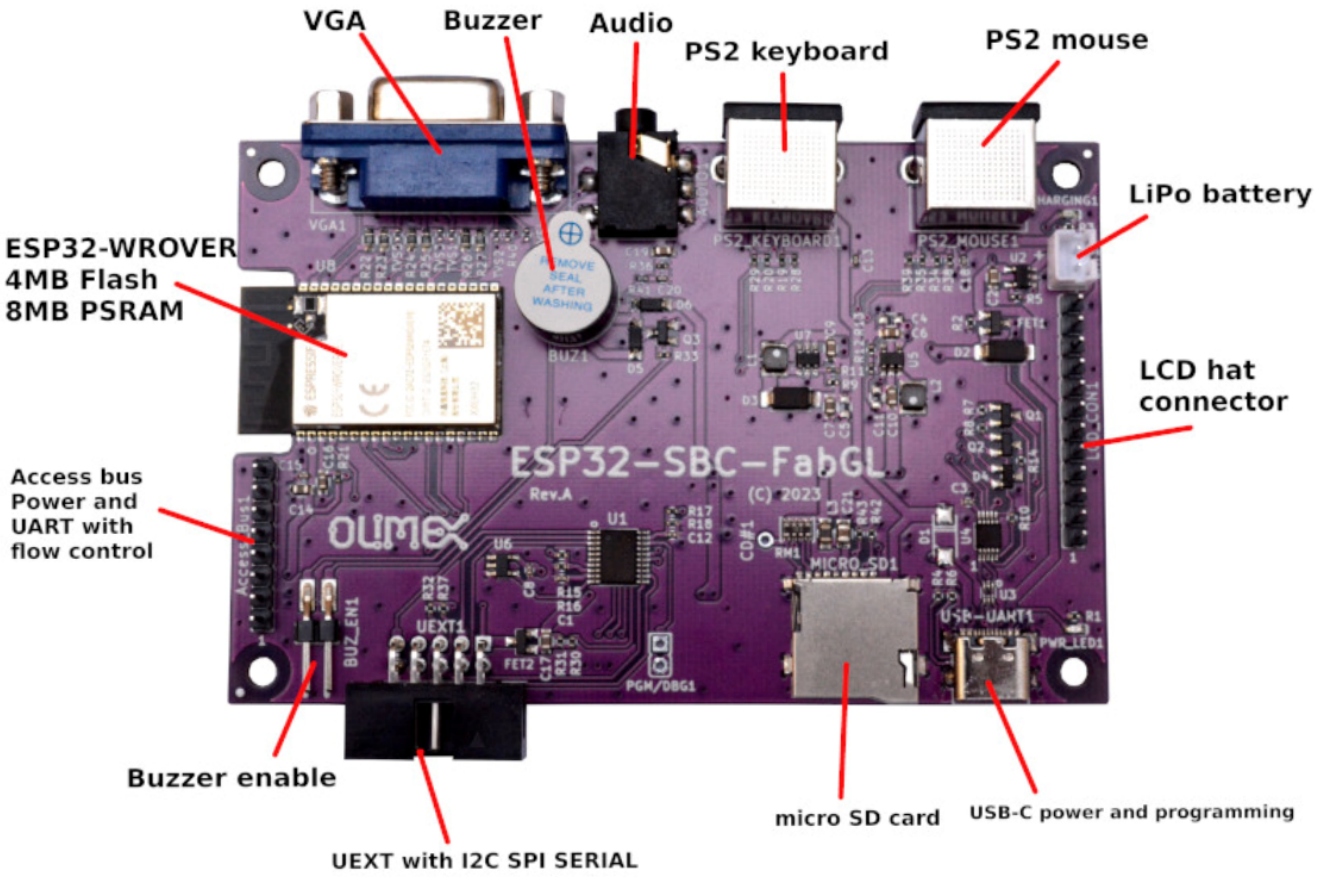
Introduction to ESP32-SBC-FabGL

ESP32-SBC-FabGL is Open Source Hardware board designed to work with FabGL library. The board has the following features:

- ESP32-WROVER-E module
- VGA connector
- PS2 keyboard connector
- PS2 mouse connector
- Buzzer
- Audio 3.5mm connector
- AccessBus connector for Serial port master processor
- LCD connector (FabGL support LCD or VGA output)
- CH32V003 co-processor for UEXT port handling
- USB-C power and programming/debug connector
- SY8089A 3.3V 2A (3A peak) DCDC power supply
- Lipo charger and battery connector
- UEXT connector (pUEXT 1.0 mm step connector)
- Dimensions: (104 x 68)mm

Order codes for ESP32-SBC-FabGL and accessories

ESP32-SBC-FabGL	FabGL board with VGA keyboard mouse
BOX-ESP32-FabGL	Custom plastic box
CABLE-USB-A-C-1M	USB-C cable 1 meter
BATTERY-LIPO1400mAh	Li-po battery 3.7V 1400mAh
UEXT modules	We have temperature, humidity, pressure, magnetic field, light sensors. Modules with LCDs, LED matrix, Relays, Bluetooth, Zigbee, WiFi, GSM, GPS, RFID, RTC, EKG, sensors and etc.

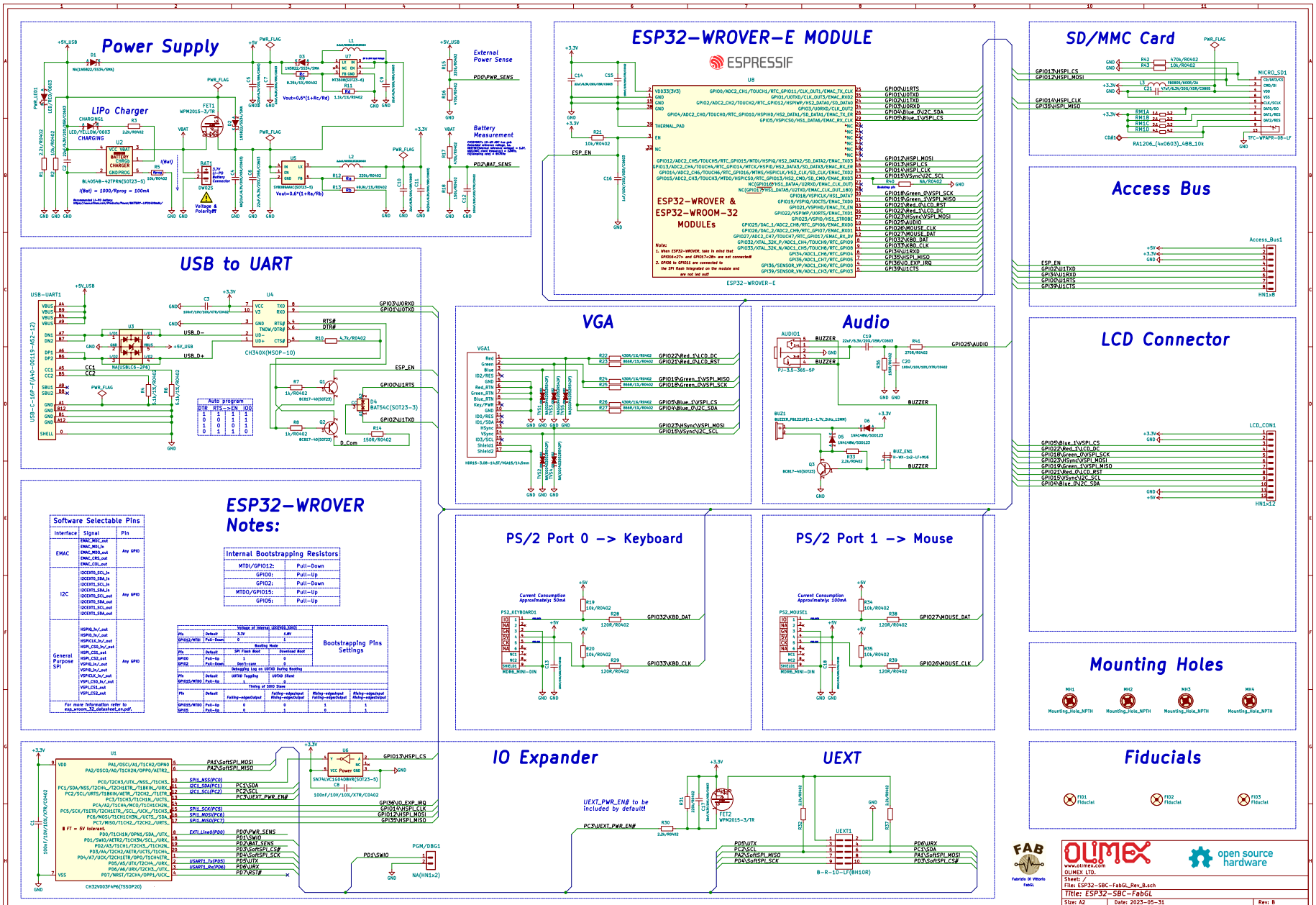


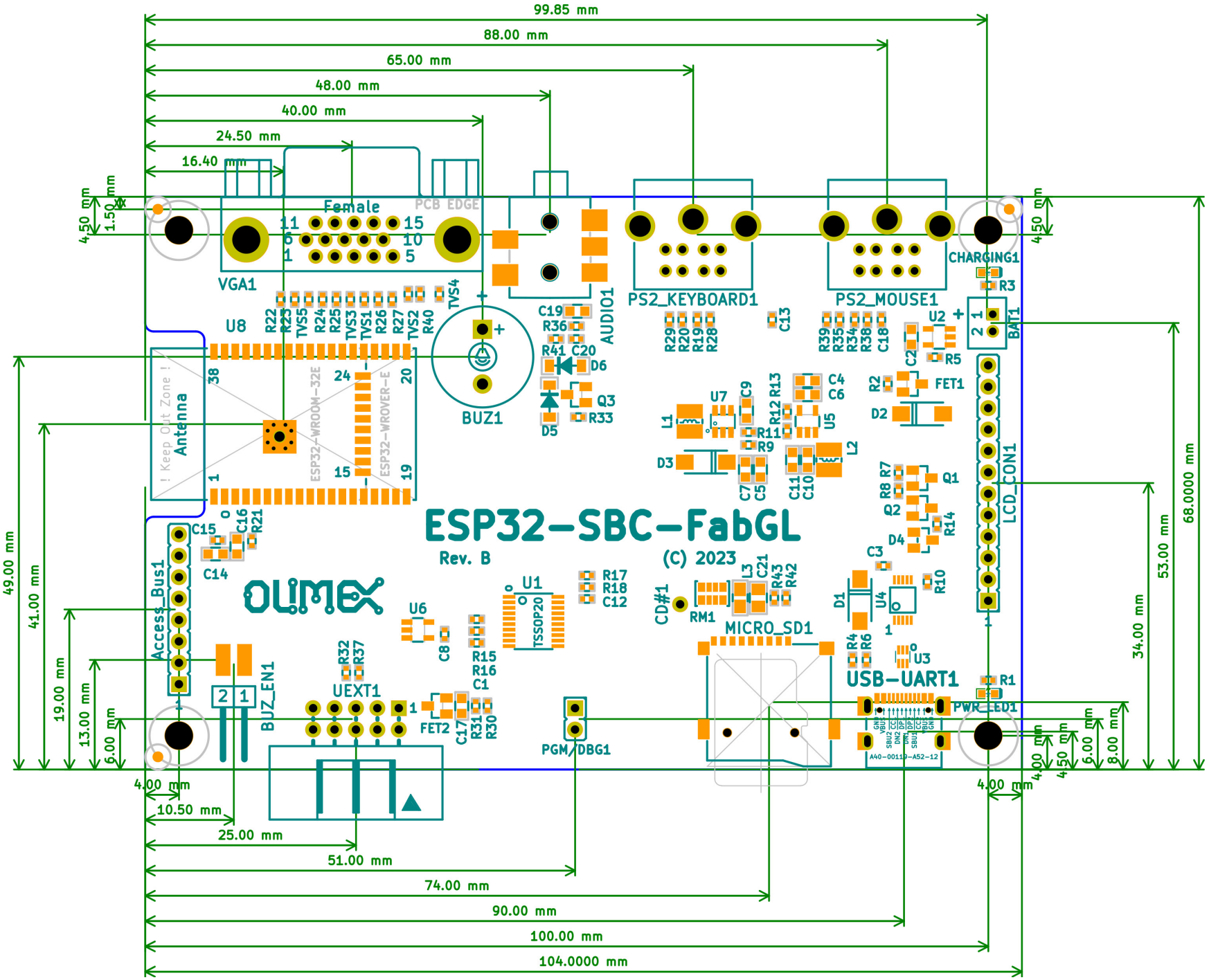
HARDWARE

ESP32-SBC-FabGL schematic and sources

Latest schematic is available on next page for viewing. Sources and PDF export are for ESP32-SBC-FabGL are available at GitHub. The page after the schematic shows the dimensions.

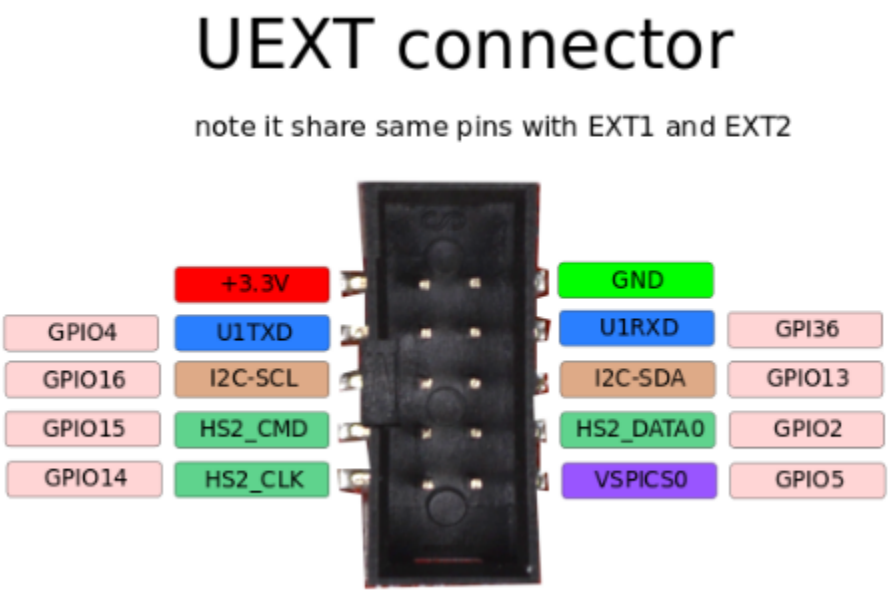
At the GitHub pages for the board you can also find schematics for older hardware revisions and also hardware revision change notes (detailing notable changes between revisions).


Power Supply
LiPo Charger
External Power Sense
Battery Measurement
ESP32-WROVER-E MODULE
ESP32-WROVER & ESP32-WROOM-32 MODULEs
SD/MMC Card
Access Bus
USB to UART
VGA
Audio
LCD Connector
ESP32-WROVER Notes:
PS/2 Port 0 -> Keyboard
PS/2 Port 1 -> Mouse
Mounting Holes
IO Expander
UEXT
Fiducials
Software Selectable Pins
Internal Bootstrapping Resistors
Bootstrapping Pins Settings
OLIMEX
open source hardware
Title: ESP32-SBC-FabGL
File: ESP32-SBC-FabGL_Rev_B.sch
Size: A2
Date: 2023-05-31
Rev: B
KiCad E.D.A. kicad (5.1.10)-1
Id: 1/1


ESP32-SBC-FabGL
Rev. B
(C) 2023
OLIMEX
99.85 mm
88.00 mm
65.00 mm
48.00 mm
40.00 mm
24.50 mm
16.40 mm
4.50 mm
1.50 mm
Female
PCB EDGE
VGA1
PS2_KEYBOARD1
PS2_MOUSE1
CHARGING1
BAT1
AUDIO1
BUZ1
U8
ESP32-WROOM-32E
ESP32-WROVER-E
Antenna
! Keep Out Zone !
LCD_CON1
Access_Bus1
BUZ_EN1
UEXT1
PGM_DBG1
CD#1
MICRO_SD1
USB-UART1
PWR_LED1
FET1
FET2
U1
TSSOP20
U2
U3
U4
U5
U6
U7
RM1
49.00 mm
41.00 mm
19.00 mm
13.00 mm
6.00 mm
4.00 mm
10.50 mm
25.00 mm
51.00 mm
74.00 mm
90.00 mm
100.00 mm
104.0000 mm
53.00 mm
34.00 mm
68.0000 mm
8.00 mm

UEXT connector:

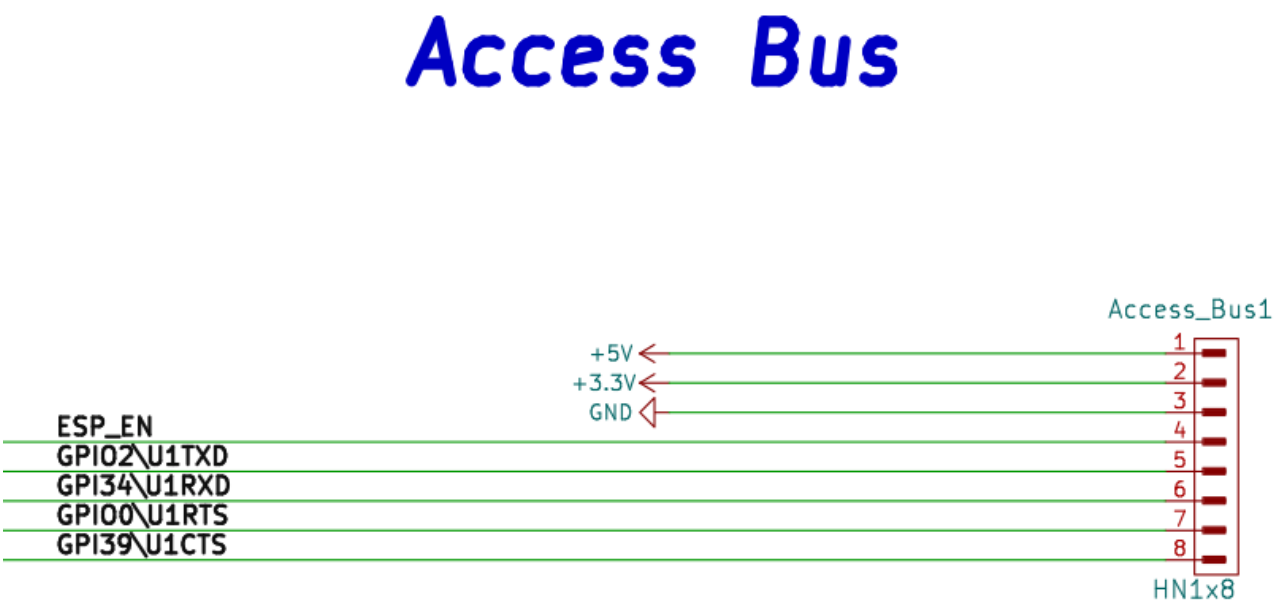
UEXT connector stands for Universal EXTension connector and contain +3.3V, GND, I2C, SPI, UART signals. UEXT connector can be in different shapes and sizes but the original UEXT connector is 0.1” 2.54mm step boxed plastic connector. All signals are at 3.3V levels.



As the boards become smaller and smaller some smaller packages were introduced (beside the original UEXT connector):
- mUEXT is 1.27 mm step boxed header connector which is with same layout as UEXT
- pUEXT is 1.0 mm single row connector (this is the connector used in RP2040-PICO30)

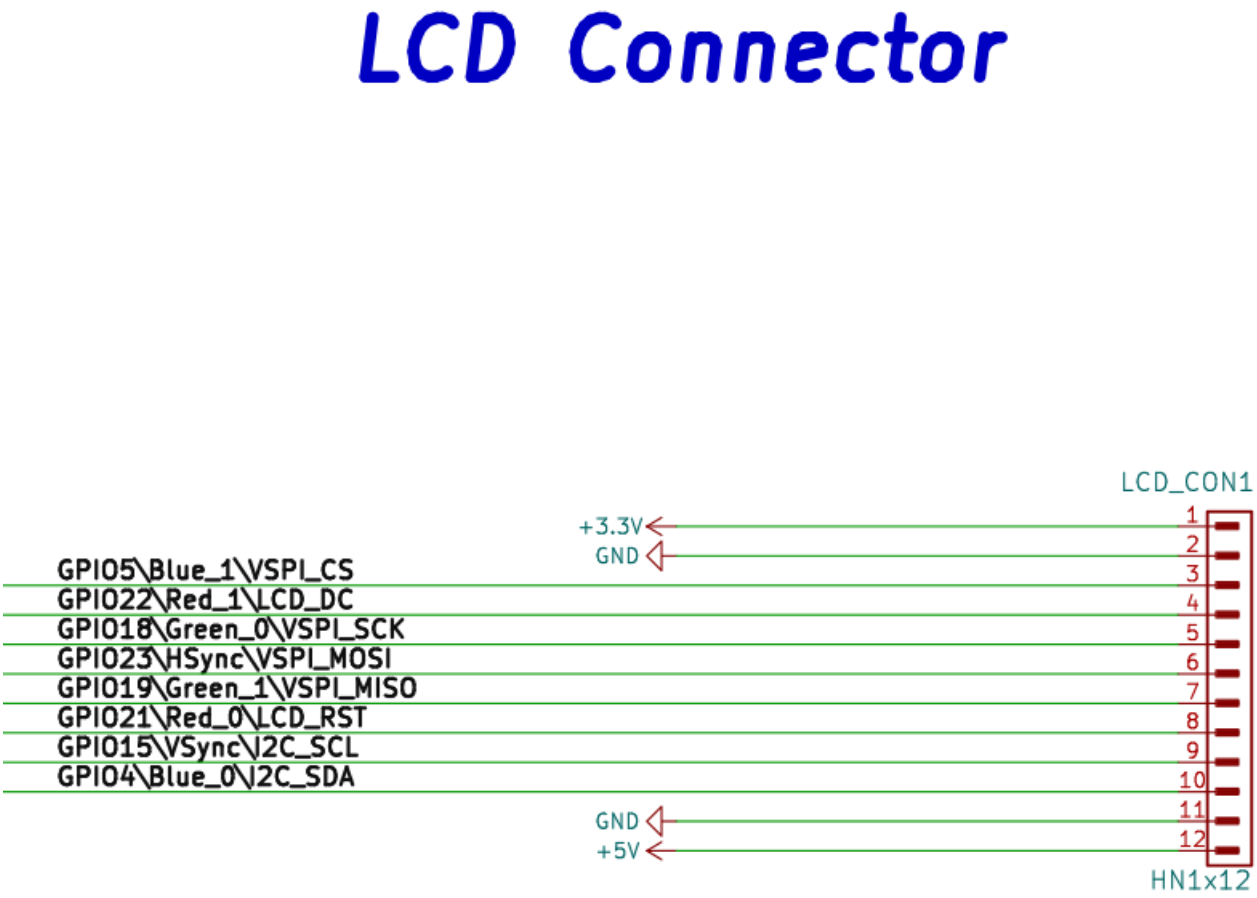
Olimex has developed number of MODULES with this connector. There are temperature, humidity, pressure, magnetic field, light sensors. Modules with LCDs, LED matrix, Relays, Bluetooth, Zigbee, WiFi, GSM, GPS, RFID, RTC, EKG, sensors and etc.

AccessBus connector:



Pin1. is marked with a square pad, all others are round pads.

LCD hat connector:



Pin1. is marked with a square pad, all others are round pads.

SOFTWARE

ESP32-SBC-FabGL is tested and works with many projects, start with the Olimex fork of FabGL project: https://github.com/OLIMEX/FabGL and make sure to check the resources here:

https://github.com/OLIMEX/ESP32-SBC-FabGL/tree/main/SOFTWARE

and here:

https://github.com/OLIMEX/FabGL

How to get the board working in Arduino IDE?

Set up the Arduino environment

1. Install and start Arduino IDE, get it from: https://www.arduino.cc/en/software

2. Add the esp32 board support by Espressif using the Board Manager (Tools > Board > Board Manager and search for "esp32"). Use the package called “esp32” by Esoressif Systems and use the version selector to load a version 2.x.x! IMPORTANT! At the time of writing this, the FabGL library is NOT compatible with version 3.x.x or newer of the package. From the drop-down menu select version 2.0.x, we used 2.0.11.

3. The Olimex fork of FabGL adds support for the ESP32-SBC-FabGL board and needs to be installed as a local library. **You'll also need to first uninstall Fabrizio's FabGL library if it's already installed.**

- Go to the Olimex FabGLrepo at GitHub: https://github.com/OLIMEX/FabGL
 - Click "Download ZIP" from the green "Code" drop-down button, top-right. Save the ZIP file as "Olimex-FabGL.zip" so you don't get confused with Fabrizio's library and repo.
 - Unzip and copy the contents to a new folder in the "Documents\Arduino\libraries" folder (e.g. "C:\Users\username\Documents\Arduino\libraries\Olimex-FabGL").

4. Close Arduino IDE. The next time the IDE is started, the local Olimex FabGL library will be available for use.

Compile and download examples to ESP32-SBC-FabGL

1. Connect the ESP32-SBC-FabGL board to your desktop PC using a USB cable. The USB-C port on the board serves as both power and data.

2. Start Arduino IDE.

3. Verify the Olimex FabGL library is loaded by:
 - Opening the Library Manager (Tools > Manage Libraries).
 - Typing "FabGL" in to the "filter" textbox.
 - Changing the "Type" to "Installed".
 - It should list "FabGL 1.0.9" (at the time of this writing) as one of the installed libraries.

4. Select a FabGL demo from (File > Examples > FabGL). The FabGL examples will be at the bottom of a lengthy list.

5. Configure the board.
 - Select the "ESP32 Dev Module" board (Tools > Board > esp32 > ESP32 Dev Module).
 - Set the board port to upload to (Tools > Port > COM#).
 - Set the partition scheme (Tools > Partition > Huge APP).
 - Disable PSRAM (Tools > PSRAM > Disabled).
 - Set the Upload Speed (Tools > Upload Speed > 921600).
 - If an upload error occurs, lower the transfer speed.

6. Edit the demo if needed. Most demos are well-commented on what has to be edited. Some demos require also preparing an SD card or else in specific manner.

7. Compile and upload to the board (Sketch > Upload).

 - If all goes well, the ESP32-SBC-FabGL board will reboot after the compilation and upload complete.
 - If the wireless parameters were not set, a prompt asking to configure the wireless connection will appear.
 - After that, the boot menu should show.

If you have compilation problems with most of the examples you probably installed latest version of espressif package for ESP32. **The most important part is to use ESP32 package version older than 3.x.x ESP32 package (we used Espressif 2.0.11). Also do NOT use the “Arduino ESP32 Boards” package! Use the “esp32” package by espressif systems!**

Wordpress articles

Also make sure to check these FabGL articles at our wordpress:

- https://olimex.wordpress.com/2023/06/27/esp32-sbc-fabgl-update-now-you-can-play-invaders-with-wii-nunchuck-via-the-uext-port/

- https://olimex.wordpress.com/2023/08/01/esp32-sbc-fabgl-how-to-work-with-the-expander-module-gpios-from-fabgl-library-and-apps/

- https://olimex.wordpress.com/2023/05/31/zx-espectrum-emulator-tested-with-esp32-sbc-fabgl-the-mensch-computer-with-w65c256-got-vga-display-and-keyboard/

- https://olimex.wordpress.com/2023/05/25/new-open-source-hardware-design-is-verified-esp32-sbc-fabgl-single-board-computer-based-on-fabgl-library-with-ps2-keyboard-mouse-and-vga-display-with-many-retro-computer-emulators/

Other projects that we haven't tested fully but should work:

https://github.com/uho/ESP32forthStation
https://github.com/OulanB/OricVGA32
https://github.com/guidol70/RunCPM_VGA32
https://github.com/EremusOne/ZX-ESPectrum-IDF
https://github.com/rpsubc8/ESP32TinyMCUMEesp81
https://github.com/rpsubc8/ESP32TinyNesMaster
https://github.com/rpsubc8/ESP32TinyZXSpectrum
https://github.com/rpsubc8/ESP32TinyFairChild
https://github.com/rpsubc8/ESP32TinyChip8
https://github.com/rpsubc8/ESP32TinyFake86
https://github.com/rpsubc8/ESP32TinyCPC
https://github.com/rpsubc8/ESP32TinyC64

Document revision history

Revision 1.0 May 2023
Revision 2.0 April 2024
Revision 3.0 July 2024
- added Arduino IDE installation instructions
- added dimensions