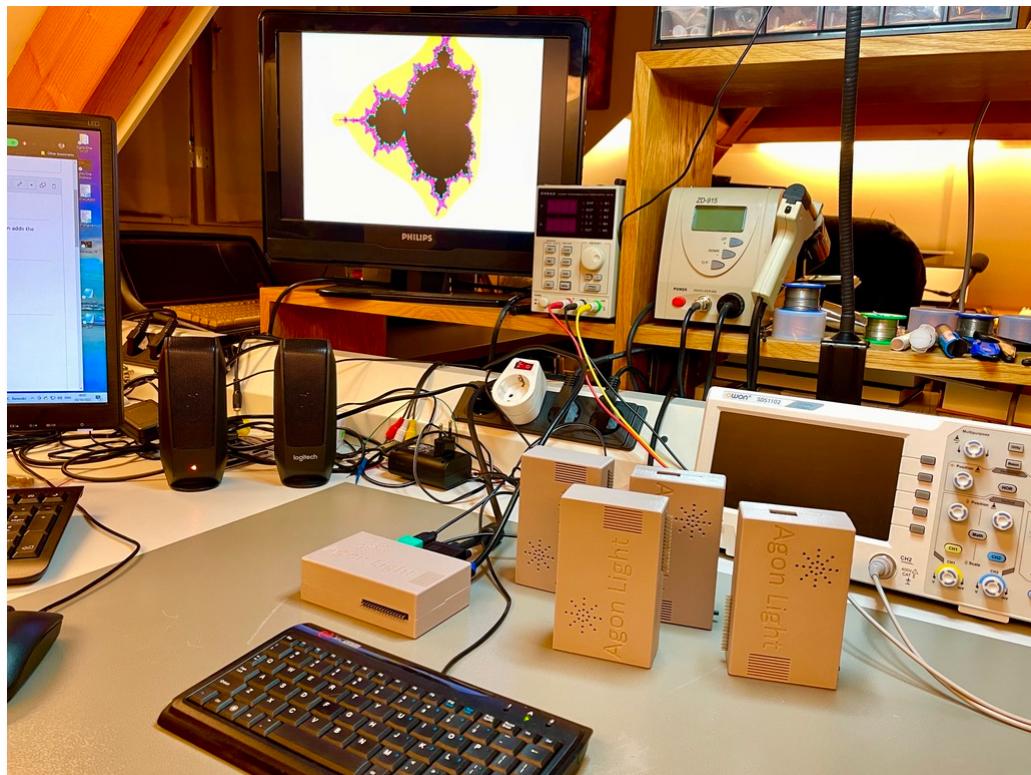


# QUICK START GUIDE

## for Agon light™



*Last update on 2/26/23 11:25:44 PM*

# BEFORE POWER UP

1. *Observe ESD precautions while unpacking, particularly when touching Agon light™'s exposed control port's pins. Agon light™ is heavily protected against ESD, but caution always helps.*
2. *Inside the ESD-safe bag, next to your Agon light™ unit, there is a USB-to-PS/2 keyboard adapter and a µSD card adapter. The µSD card itself is already pre-loaded with software and inserted into your unit.*
3. *A male-to-male USB3 cable is provided to power your unit (and, if you so wish, update the Video Display Processor's—VDP—firmware as per instructions in Appendix C).*
4. *The 'Hardware Manual' included is meant mainly for users who want to write their own firmware. But page 11 is important for you to run the 'gpio.bbc' test program, as it contains the pinout of the control port (more on this shortly).*
5. *Your Agon light™ unit has been tested before shipping. But if you encounter problems, please contact us to get a new unit.*
6. *The firmware pre-loaded to your unit is limited but sufficient for a review. You can always update it by:*
  - a. *Downloading the latest 'bbcbasic.bin' file from:  
<https://github.com/TheByteAttic/AgonLight/tree/main/uSD%20card%20files>  
and replacing (using your regular PC) the file currently loaded in the root directory of the µSD card in your unit (pre-formatted as Fat32 and with a partition of maximum 32GB); and*
  - b. *Downloading the latest Arduino sketch from:  
<https://github.com/breakintoprogram/agon-vdp/releases> (choose the latest release)  
and uploading it to your Agon light™ via the USB cable provided, as per instructions in Appendix C.*
  - c. *Updating the Quark MOST™ firmware of the eZ80 CPU by using this software utility: <https://github.com/envenomator/agon-flash>  
The latest version of Quark MOST™ can be found here:  
<https://github.com/breakintoprogram/agon-mos/releases>  
Further details on uploading the firmware can be found in the Firmware Installation Guide available in the Agon light™ Github repository.*
7. *Your Agon light™ unit is set to US keyboard layout by default. If you are using a UK keyboard, please edit the 'autoexec.txt' file in the root directory of the µSD card in your unit (using your regular PC), so that it states "SET KEYBOARD 0", instead of 1.*

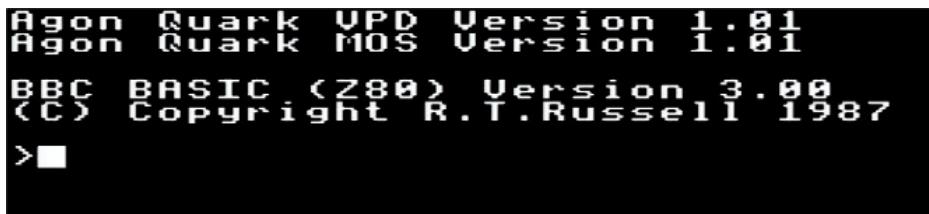
# POWERING UP YOUR UNIT

1. *Make sure the µSD card is inserted in the unit (pre-formatted as Fat32, with a partition of maximum 32GB, and the appropriate files pre-loaded to it).*
2. *Connect a PS/2 keyboard or a PS/2-compatible USB keyboard, in this latter case using the adapter provided.*
3. **[OPTIONAL]** *Connect stereo PC speakers. Make sure the audio plug is inserted all the way in. The audio hole in Agon light™'s case exactly matches the size of regular 3.5mm audio plugs, so you may have to wiggle it in to insert it fully. If the plug is not correctly inserted, you will hear a buzz from the speakers. If you do not connect speakers at all, Agon light™ will automatically use its on-board buzzer instead.*
4. *Connect your unit to a standard VGA monitor and turn the monitor on.*
5. **[OPTIONAL]** *If you are going to use Agon light™ to control any external circuit, connect said circuit to Agon light™'s control port before powering it up.*
6. *Power your unit up by connecting it, through the USB3 cable provided, to any standard USB charger or power source (there is no power switch). Once powered up, you should be able to see the internal power LED lit up, through the ventilation grill on the top-right of the case.*



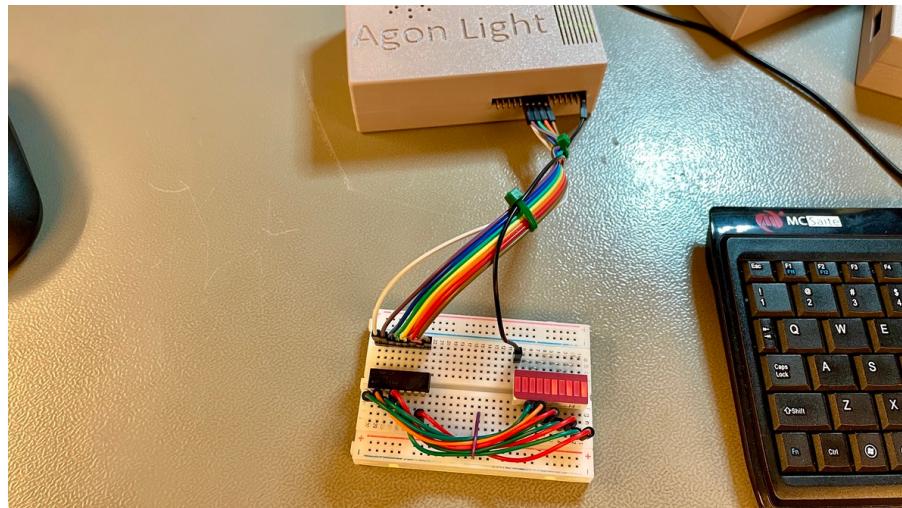
# USING YOUR UNIT

1. Upon turning your unit on, you should be greeted with this screen:



2. Adjust your monitor settings to properly display the image (mainly vertical and horizontal alignment, so the image isn't cropped).
3. There are three components to the firmware pre-installed in your unit:
  - a. Quark™ MOS is the low-level operating system, similar to CP/M and DOS.
  - b. Quark™ VDP is the software running on Agon's Video Display Processor (an ESP32-PICO-D4), which generates the screen, sound, and controls the keyboard.
  - c. BBC BASIC is the BASIC language interpreter that runs on top of Quark™ MOS. MOS commands can be issued from within BBC BASIC by prefixing them with the "\*" character, as in "\*CAT" (which lists the files in the currently active directory of the µSD card inserted in the unit). See Appendix A for a list of MOS commands.
4. Now type "\*CAT" to see the files present in the "tests" directory of your µSD card.
5. Here is an overview of the test programs you should find there:
  - a. "triangles.bbc" renders colorful random triangles on the screen.
  - b. "cube.bbc" renders a rotating cube on the screen.
  - c. "sprites.bbc" renders moving square sprites on the screen.
  - d. "udg.bbc" illustrates how to change character definitions.
  - e. "sound.bbc" illustrates how to produce simple sounds from within BBC BASIC. Agon light™'s sound capabilities, however, far exceed what can be done from within BASIC.
  - f. "benchmNUM.bbc" are a series of simple benchmarks, meant to measure the BASIC performance of Agon light™.
  - g. "noelrlb.bbc" is Noel's Retro Lab's 8-bit BASIC benchmark.
  - h. "gpio.bbc" implements an 8-bit binary counter on 8 pins of Agon light™'s control port (namely, the port C pins illustrated on page 11 of the Hardware Manual included). To use this test, you need to connect the respective pins to in-series 1KΩ resistors, and then to the positive leads of LEDs (which, in turn,

*(should be connected to common ground via the other leads). In the following picture, the appropriate circuit in illustrated, in which a  $1K\Omega$  resistor bank and an LED bar are used:*



6. These test programs are meant solely to illustrate how to do different things in Agon light™’s BBC BASIC, not to show the full hardware capabilities of Agon light™, which go much beyond these tests.
7. You can now load any of the “filename.bbc” files by typing “LOAD “filename.bbc”” and hitting ENTER. Notice that:
  - a. BASIC commands are case-sensitive and should be typed in CAPITAL LETTERS, or they won’t be recognized.
  - b. File names are not case-sensitive, so “GPIO.BBC” is the same as “gpio.bbc”.
8. The file you have just loaded can be inspected by doing “LIST”.
9. Optionally, you can list only a range of lines, by doing: “LIST firstLine,lastLine”, as in “LIST 10,100”.
10. You can now execute the loaded program by doing “RUN”.
11. To abort a running program, press ESC.
12. Agon light™ uses three different screen modes: 0, 1 and 2. They are specified as follows:  
 MODE 0: 640 x 480 pixels @ 60Hz  
 MODE 1: 512 x 384 pixels @ 60Hz (this is the default mode on boot)  
 MODE 2: 320 x 200 pixels @ 75Hz  
 64 simultaneous colors can be displayed.
13. To change screen modes, simply type e.g. “MODE 0” followed by ENTER.
14. When you’ve aborted a BASIC program that has changed the screen mode, you can return to the default mode by doing “MODE 1”.
15. To clear the screen, do “CLS”.
16. If a BASIC program has left sprites on the screen after being aborted, “CLS” won’t clear the sprites; only a screen mode change will. If you ‘change’ the screen mode to the same mode already in use, that will clear the sprites as well.

17. If you modify or create a new BASIC program, you can save it to the currently active directory of the µSD card by typing “SAVE “filename””. In the current version of the firmware, you cannot overwrite an existing file, so changes to a program must be saved with a different file name.
18. See appendix B for a summary of BASIC commands.
19. See the project’s official web page for updates, links, online documentation, etc.: <https://www.thebyteattic.com/p/agon.html>.
20. Have fun and please be kind to us! We are aware that the firmware still has bugs and other shortcomings, and are working hard to improve it a.s.a.p.. Please also remember that Agon light™ is a fully open-source project, both as far as the hardware and the software are concerned, and that the project team does not sell Agon light™ units themselves. We make precisely zero money with all this (in fact, we just spend a lot of money).

# APPENDIX A

## QUARK™ MOS COMMANDS

1. **CAT**: *Directory listing of the current directory. Aliases include DIR and .*
2. **CD path**: *Change current directory*
3. **LOAD filename <addr>**: *Load a file from the SD card to the specified address*
4. **MKDIR filename**: *Make a folder on the SD card*
5. **SAVE filename addr size**: *Save a block of memory to the SD card*
6. **RUN <addr>**: *Call an address in memory (switching to Z80 mode - ADL=0)*
7. **DEL filename**: *Delete a file or folder (must be empty). Aliases include ERASE*
8. **REN filename1 filename2**: *Rename a file*
9. **JMP addr**: *Jump to the specified address in memory*
10. **SET option value**: *Set a system option*

# APPENDIX B

## QUARK™ BASIC

*In addition to the core BBC BASIC for Z80 core language (details of which can be found here: <https://github.com/breakintoprogram/agon-bbc-basic/blob/main/bbcbasic.txt>), BBC BASIC for Agon light™ adds the following functionality.*

## Editor

*BBC BASIC currently uses the MOS editor.*

- *Use the cursor keys to navigate around the current edit line.*
- *Line lengths must be less than 256 characters.*

## Star commands

*The following \* commands are supported:*

- **\*BYE**  
*Exit BASIC and return to MOS.*
- **\*CAT (Alias: \*.)**  
*List the current directory on the μSD card.*
- **\*CD path**  
*Change the current directory on the μSD card.*
- **\*EDIT linenum**  
*Pull a line into the editor for editing.*
- **\*ERASE filename**  
*Erase a file or folder from the μSD card. If the directory is not empty, then an error message will be displayed.*
- **\*FX osbyte, params**  
*Execute an OSBYTE command.*
- **\*MKDIR PATH**  
*Create a folder on the μSD card.*
- **\*REN oldfilename newfilename**  
*Rename a file on the μSD card.*

# BASIC

The following statements differ from the BBC BASIC standard:

- `MODE n`  
*Three modes are currently supported by the VDP. These can be tweaked by modifying the function `set_mode` in the VDP code.*
- `MODE 0: 640 x 480 pixels @ 60Hz`  
`MODE 1: 512 x 384 pixels @ 60Hz (this is the default mode on boot)`  
`MODE 2: 320 x 200 pixels @ 75Hz`
- `GCOL mode, r,g,b`  
*Set the graphics color to the specified RGB color. Each component is a number between 0 and 255.*
- `VDU`  
*The VDU command is a work-in-progress with a handful of mappings implemented:*

`VDU 8: Backspace`  
`VDU 9: Advance one character`  
`VDU 10: Line feed`  
`VDU 11: Move cursor up one line`  
`VDU 12: CLS`  
`VDU 13: Carriage return`  
`VDU 16: CLG`  
`VDU 18,mode,r,g,b: GCOL mode,r,g,b`  
`VDU 22,n: Mode n`  
`VDU 25,mode,x;y,: PLOT mode,x,y`  
`VDU 29,x;y,: Set graphics origin to x,y`  
`VDU 30: Home cursor`  
`VDU 31,x,y: TAB(x,y)`

*Examples:*

`VDU 25,64,128;88;: Plot point in middle of screen`  
`VDU 22,1: Change to Mode 1`

# APPENDIX C

## QUARK™ VDP

The VDP (Video Display Processor) is a serial graphics terminal that takes a BBC BASIC text output stream as input. The output is via the VGA connector on Agon light™. It will process any valid BBC BASIC VDU commands (starting with a character between 0 and 31).

For example:

VDU 25, mode, x; y; is the same as PLOT mode, x, y

## Building

The VDP is programmed via the USB connector at the back of the case using the standard Arduino IDE. Use version 1.8.19 or later.

### Arduino IDE settings

In order to add the VDP as a supported board in the Arudino IDE, you will need to add a new URL into the board manager. Here are the steps:

1. Select Preferences from the File menu.
2. In the Additional Board Manager URLs text box, enter the following URL:  
[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)
3. In the Board Manager (from the Tools menu), select the esp32 board, making sure version 2.0.4 is installed.
4. Now the board can be selected and configured.

Board: “ESP32 Dev Module”  
Upload Speed: “921600”  
CPU Frequency: “240Mhz (WiFi/BT)”  
Flash Frequency: “80Mhz”  
Flash Mode: “QIO”  
Flash Size: “4MB (32Mb)”  
Partition Scheme: “Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)”  
Core Debug Level: “None”  
PSRAM: “Enabled”

5. And for the Port, you will need to determine the COM port that Agon light™ is assigned from your OS after it is connected.
6. Now a third party library will need to be installed from the Library Manager in the Tools menu: FabGL version 1.0.8.