

SA30 OEM Programming Procedure

Draft Version 0.2

Version

Version	Date	By	Description
0.1	24/08/20	Peter Kuell	Initially created
0.2	08/12/20	Peter Kuell	Added certificate, serial number burning procedure

Table of Contents

Version	2
Background Information	3
Host MCU Programming	4
Avatar Bootloader/Recovery Programming	5
Application Layer Programming.....	8
Google Cast Certificate	Error! Bookmark not defined.
Google Cast Certificate, Serial Number, MAC address and Region Programming	11
Locking the Module	13

Background Information

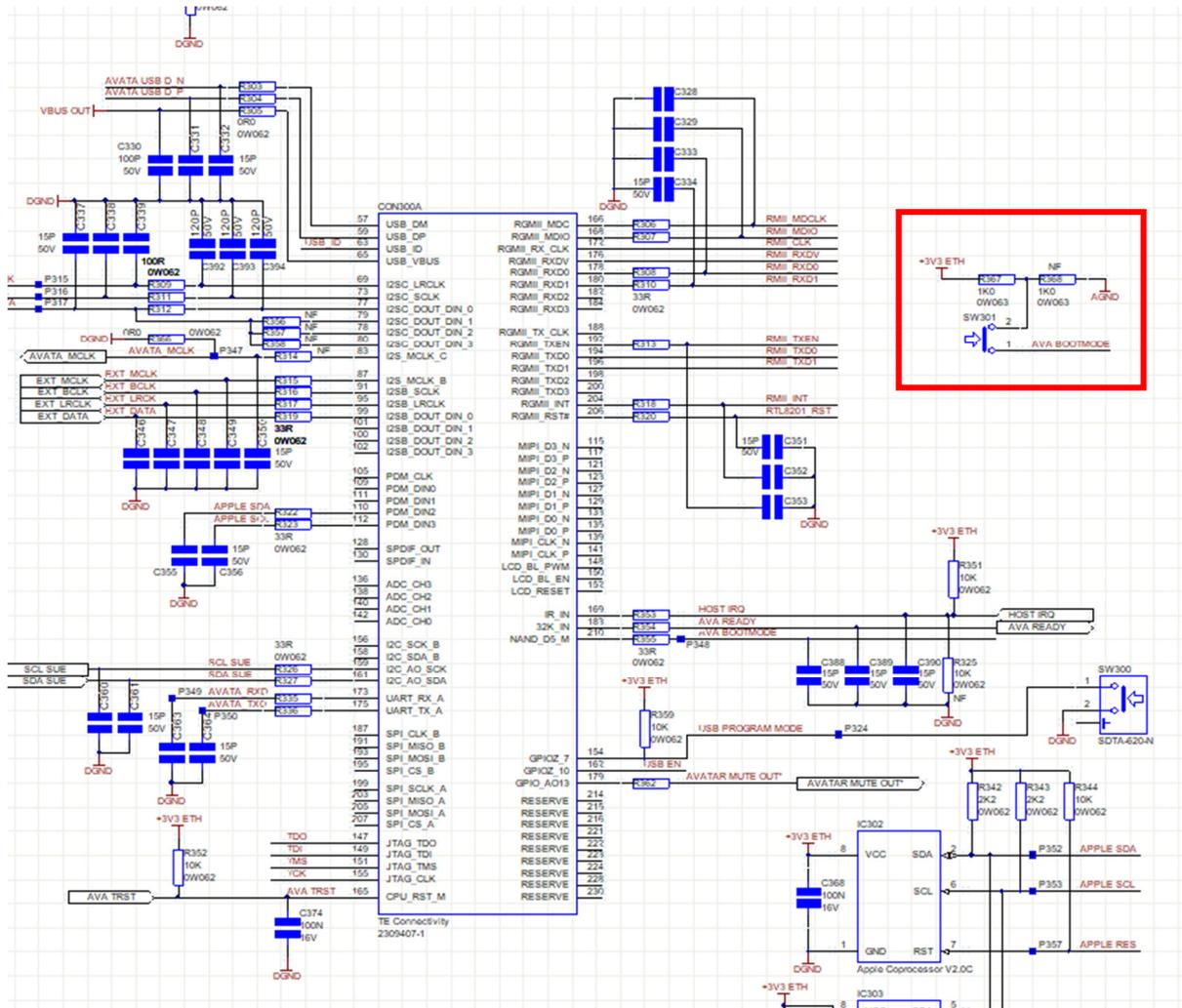
The avatar module requires a bootloader and an application to be loaded in order to work. At the time of writing there is not a reliable method available to program both parts together. This document describes the processes the host MCU and for programming the avatar bootloader followed by the avatar application layer.

Host MCU Programming

The combined hex image should be loaded into the MCU via the ICD programmer.

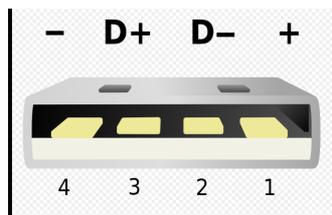
Avatar Bootloader/Recovery Programming

The bootloader (also called recovery partition) is programmed using the AMLogic USB burning tool and can be programmed with the avatar module installed in the SA30 using the USB socket and SW301 on L298.



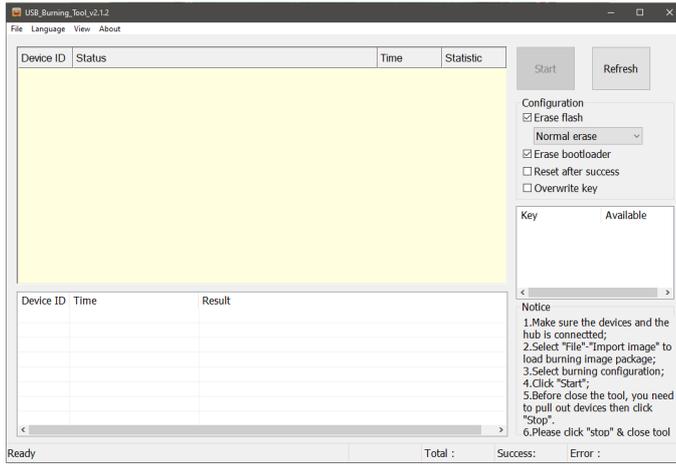
It also requires a custom USB type A to type A cable.

This requires connection of the screen, D+, D- and ground pins (pins 4, 3 and 3 respectively) to each type A plug. **VCC (pin 1) should not be connected.**



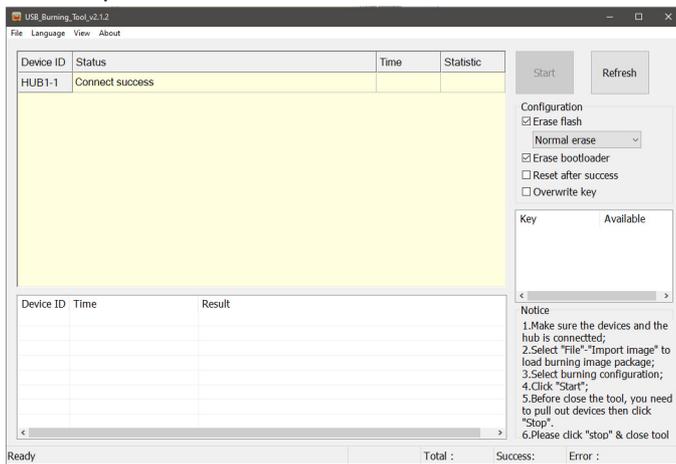
1. The USB cable is connected between the PC and the USB socket on the rear of the unit.
2. Avatar module is installed.

3. Open the USB burning tool.

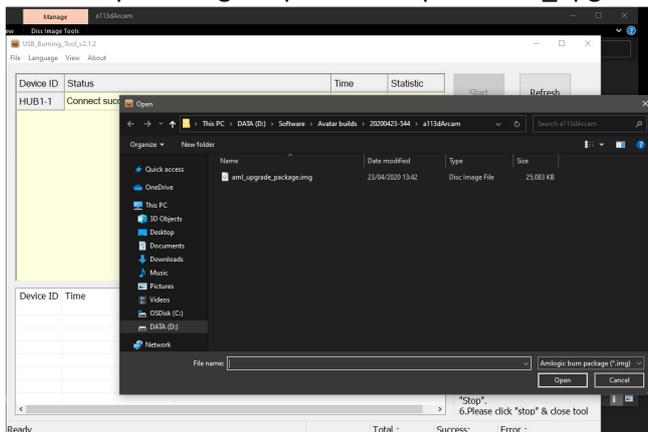


4. Press and hold SW301 and power up the unit.

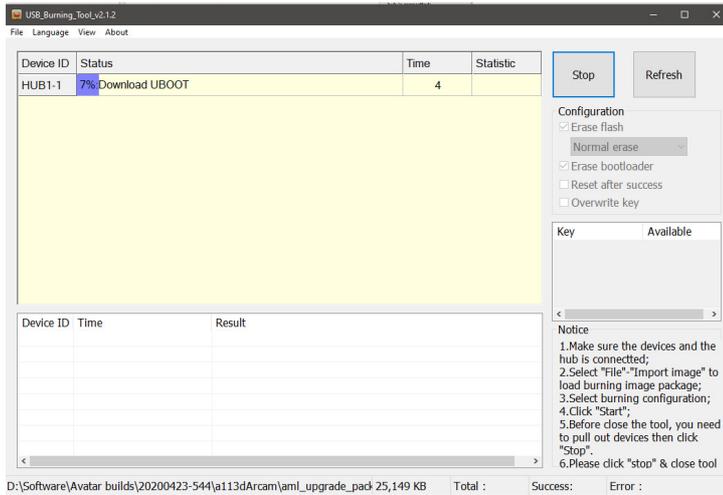
- a. Note the host MCU will need to be programmed as this will be in control of enabling power to the avatar.



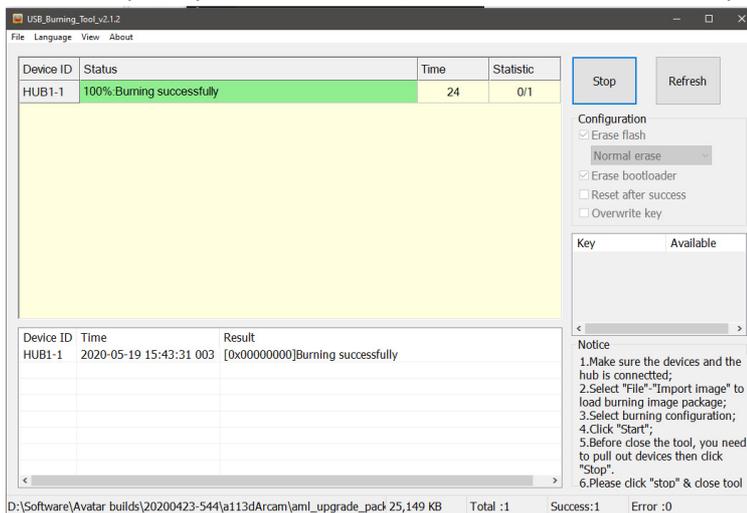
5. File -> Import image import the required aml_upgrade_package.img file.



6. Press start.



7. Once complete power down EVB, disconnect USB and close program.



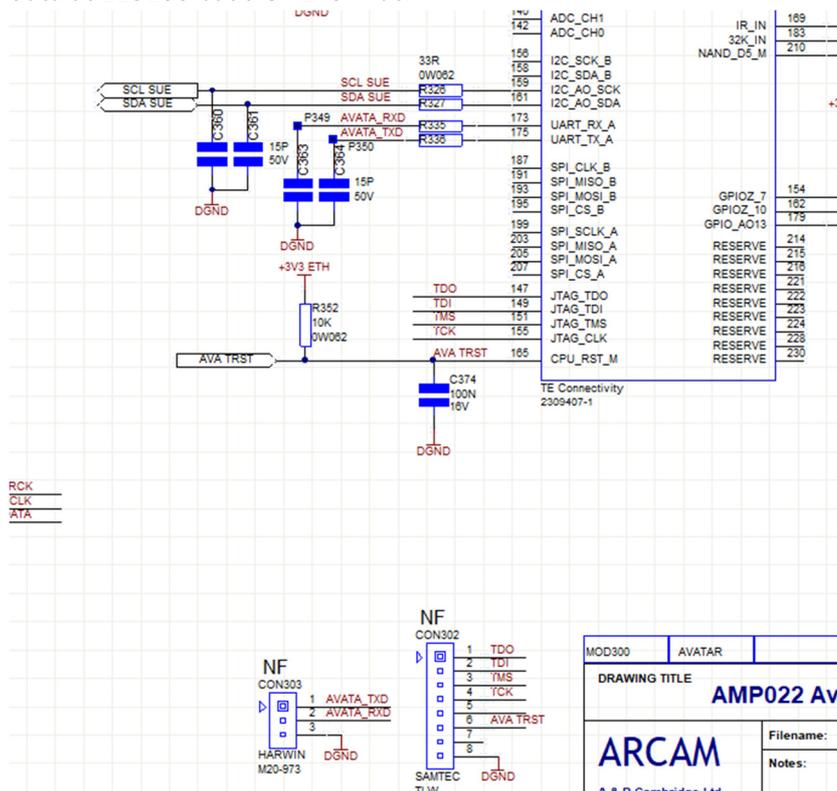
This completes the process of programming the bootloader.

Application Layer Programming

Once the bootloader has been installed the application layer is installed via the USB socket on the unit.

1. The only issue with this method is the avatar bootloader cannot communicate with the host so the unit cannot give any status updates.

The easiest thing to do is to add an RS232 level shifter or similar to pins 1/2 of CON303 and monitor its output via docklight or similar RS232 logging program. The avatar will output data at 115200 baud 8N1 format.



2. Copy the required image.swu to a USB stick.
3. Insert USB stick into the USB socket on the rear of the unit.
4. Whilst applying power to the unit press and hold the “service” push switch on the rear of the unit. Keep the switch held for 20 seconds then release it. The log should be displaying “Arcam login: Connected”

```

NOTIFY] : SWUPDATE running : [extract_sw_description] : Found file:CR<LF>
HT> filename sw-description.sig<CR><LF>
HT> size 256<CR><LF>
HT> checksum 0x77ba VERIFIED<CR><LF>
CR<LF>
NOTIFY] : SWUPDATE running : [swupdate_verify_file] : Verify signed image: Read 1838 bytes<CR><LF>
CR<LF>
NOTIFY] : SWUPDATE failed [0] ERROR corelib/verify_signature.c : verify_final : 107 : EVP_DigestVerifyFinal failed, error 0x407806a 0<CR><LF>
CR<LF>
NOTIFY] : SWUPDATE running : [swupdate_verify_file] : Error Verifying Data<CR><LF>
CR<LF>
NOTIFY] : SWUPDATE running : [parse] : Ignore signature errors is enabled, ignoring image verification failure<CR><LF>
longpose web server v. 3.8 with pid 2476 started on port(s) 80 with web root [/www]<CR><LF>
Version 0.100.544.0xCR09A7-3d60939a<CR><LF>
NOTIFY] : SWUPDATE running : [parse_hw_compatibility] : Accepted Hw Revision : Generic<CR><LF>
NOTIFY] : SWUPDATE running : [parse_hw_compatibility] : Accepted Hw Revision : a113dArcam<CR><LF>
NOTIFY] : SWUPDATE running : [parse_images] : Found Image uboot 0.100.131114.0-3d60939a: u-boot.tar.gz in device : /dev/mtd0 for handler uboot-a113 Version must be checked<CR><LF>
CR<LF>
NOTIFY] : SWUPDATE running : [parse_images] : Found compressed Image rootfs 0.100.544.0xCR09A7-3d60939a: yocto-ndk-ip-image-swu-a113dArcam.ubifs in volume : nsdk-rootfs for handler ubivol Version must be checked<CR><LF>
CR<LF>
NOTIFY] : SWUPDATE running : [parse_images] : Found Image fit 0.100.131113.0-3d60939a: fitImage-a113dArcam-signed in device : /dev/mtd4 for handler flash Version must be checked<CR><LF>
CR<LF>
NOTIFY] : SWUPDATE running : [parse_images] : Found Image swu-fit 0.100.31.0-3d60939a: fitImage-swu-a113dArcam-signed in device : /dev/mtd3 for handler flash Version must be checked<CR><LF>
CR<LF>
NOTIFY] : SWUPDATE running : [parse_scripts] : Found Script: swupdate_postinstall.sh<CR><LF>
CR<LF>
NOTIFY] : SWUPDATE running : [check_field_string] : Configuration Key is empty<CR><LF>
CR<LF>
NOTIFY] : SWUPDATE running : [parse_scripts] : Found Script: swupdate_preinstall.sh<CR><LF>
CR<LF>
NOTIFY] : SWUPDATE running : [check_hw_compatibility] : Hardware default Revision: Generic<CR><LF>
NOTIFY] : SWUPDATE running : [check_hw_compatibility] : Hardware compatibility verified<CR><LF>
NOTIFY] : SWUPDATE running : [epio_scan] : Found file:CR<LF>
HT> filename u-boot.tar.gz<CR><LF>
HT> size 1486747<CR><LF>
HT> REQUIRED<CR><LF>
CR<LF>
NOTIFY] : SWUPDATE running : [epio_scan] : Found file:CR<LF>
HT> filename yocto-ndk-ip-image-swu-a113dArcam.ubifs<CR><LF>
HT> size 102796468<CR><LF>
HT> REQUIRED<CR><LF>
CR<LF>
CR>
CR<LF>
Wdy [Yocto Project Reference Distro] 2.1.3 a113dArcam /dev/ttyS0<CR><LF>
CR>
CR<LF>
CR>
CR<LF>
a113dArcam login: Connected<CR><LF>

```

5. Wait for approx. 20 seconds more and the avatar will then start upgrading from the USB stick.

The log output will look something like this once it actually starts upgrading.

```

Set volume size to 411783168<CR><LF>
Volume ID 1, size 3243 LEBs (411783168 bytes, 392.7 MiB), LEB size 126976 bytes (124.0 KiB), dynamic, name "nsdk-rootfs", alignment 1<CR><LF>
preinstall: Creating partitions complete<CR><LF>
[NOTIFY] : SWUPDATE running : [execute_shell_script] : Calling shell script /tmp/swupdate_preinstall.sh : return with 128<CR><LF>
scan for ubi devices<CR><LF>
[NOTIFY] : SWUPDATE running : [scan_ubi_volumes] : mtd7:<HT> Volume found : <HT> nsdk-settings<CR><LF>
[NOTIFY] : SWUPDATE running : [scan_ubi_volumes] : mtd7:<HT> Volume found : <HT> nsdk-rootfs<CR><LF>
[NOTIFY] : SWUPDATE running : [install_single_image] : Found installer for stream fitImage-swu-a113dArcam-signed flash<CR><LF>
<CR><LF>
Update started !<CR><LF>
[NOTIFY] : SWUPDATE running : [get_mtd_from_device] : mtd name [/dev/mtd3] resolved to [/dev/mtd3]<CR><LF>
Interface: UNKNOWN<CR><LF>
<CR><LF>
<CR><LF>
INFO : <CR><LF>
<CR><LF>
<CR><LF>
[-----] 1 of 4 0% (fitImage-swu-a113dArcam-signed)<CR>
[NOTIFY] : SWUPDATE running : [install_flash_image] : Copying fitImage-swu-a113dArcam-signed into /dev/mtd3<CR><LF>
[-----] 1 of 4 1% (fitImage-swu-a113dArcam-signed)<CR>
[-----] 1 of 4 2% (fitImage-swu-a113dArcam-signed)<CR>
[-----] 1 of 4 3% (fitImage-swu-a113dArcam-signed)<CR>
[-----] 1 of 4 4% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 5% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 6% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 7% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 8% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 9% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 10% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 11% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 12% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 13% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 14% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 15% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 16% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 17% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 18% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 19% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 20% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 21% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 22% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 23% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 24% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 25% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 26% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 27% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 28% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 29% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 30% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 31% (fitImage-swu-a113dArcam-signed)<CR>
[=====] 1 of 4 32% (fitImage-swu-a113dArcam-signed)<CR>

```

6. The upgrade will take around 90 seconds and the avatar will then restart and attempt to communicate with the host. If it is successful the host will report a firmware version for the avatar. The avatar log will output something like this:

```

[ 29.75750981] bcm5dh_oob_intr_register: enable_irq_wake failed with -6<CR><LF>
[ 29.76662980] dhd_preinit_ioctls: Set tcapack_sup_mode 1<CR><LF>
[ 29.76946380] dhd_apply_default_clm: Ignore clm file /etc/wifi/4359/clm.blob<CR><LF>
[ 29.77672480] Firmware up: op_mode=0x0002, MAC=c8:84:7d:41:fd:7a<CR><LF>
[ 29.78661880] Driver: 1.363.59.144.15 (r)<CR><LF>
[ 29.78661880] Firmware: w10: Aug 6 2018 15:52:15 version 9.87.51.11.18 (4ce23ed@shgit) (r) FWD 01-7de9b5d0<CR><LF>
[ 29.78661880] CLM: 9.7.5 <CR><LF>
[ 29.79968880] dhd_txglom.enable: enable 1<CR><LF>
[ 29.80282280] dhd_conf_set_txglom_params: txglom_mode=copy, use_rxchain=0<CR><LF>
[ 29.80954780] dhd_conf_set_txglom_params: txglomsize=36, deferred_tx_len=0<CR><LF>
[ 29.81634980] dhd_conf_set_txglom_params: tx_in_rx=1, txlnrx_thres=128, dhd_conlimax=1<CR><LF>
[ 29.82429680] dhd_conf_set_txglom_params: tx_max_offset=0, txctl_tmo_fix=300<CR><LF>
[ 29.83127580] dhd_conf_get_disable_proptx: fw_proptx=0, disable_proptx=1<CR><LF>
[ 29.83987880] dhd_pno_init: Support Android Location Service<CR><LF>
[ 29.86322280] dhd_conf_set_country: set country US, revision 988<CR><LF>
[ 29.86689380] Country code: US (US/988)<CR><LF>
[ 29.86819780] dhd_conf_set_bw_cap: set bw_cap 5g 1<CR><LF>
[ 29.87762780] wl_android_wifi_on: Success<CR><LF>
[ 29.87881783] wl_create_event_handler(): threadw1_event_handler:b34 started<CR><LF>
[ 29.87882581] task Enter, task = 0xffffffff01ce2ecd8<CR><LF>
[ 29.92439180] dhd_open: Exit ret=0<CR><LF>
Interface wlan0 is up<CR>
<CR><LF>
Adding interface uap0<CR>
<CR><LF>
[ 34.95474783] ignore event 54, not interested<CR><LF>
[ 34.95586780] Register interface [uap0] MAC: c2:84:7d:41:fd:7a<CR><LF>
[ 34.95586780] <CR><LF>
Error: no such command: input linein<CR>
<CR><LF>
.<CR>
.<CR>
logger.sh(/tmp/log-socketd) has been started, current/max size (bytes): 0/2097152.<CR><LF>
<CR>
<CR><LF>
Poky (Yocto Project Reference Distro) 2.1.3 st68-41fd7a /dev/tty50<CR><LF>
<CR>
<CR><LF>
<CR>
st68-41fd7a login: [ 37.08390780] meson6-dwmac ff3f0000.ethernet eth0: fail to init PTP.<CR><LF>
[ATOS] TDK-v2.4<CR><LF>
[ATOS] TA verified (2048-2048)<CR><LF>
ERR [2713] TEEs:compute_hmac:335: Cannot compute MAC (key not set)<CR><LF>
ERR [3055] TEEs:compute_hmac:[ATOS] "WARNING" TA Version Table not ready!<CR><LF>
335: Cannot compute MAC (key not set)<CR><LF>
[ 42.31663881] i2c i2c-3: [aml_i2c_xfer] error ret = -5 (-EIO)<CR><LF>
[ 42.31675981] i2c i2c-3: token 1, master_no(3) 300k addr 0x10<CR><LF>
[ 42.46027280] Disconnect STA : ffffffff:ffff scb_val.val 3<CR><LF>
[ 42.47713480] netdev_ifidx(7), chan_type(1) target channel(2) <CR><LF>
[ 42.51816180] ** AP/GO Link up event **<CR><LF>

```

Once it shows the SA30-xxxx message it has successfully upgraded and started and talked to the host to find out what unit it is in.

- Without using the RS232 log there is no real indication of the process but once it is complete the unit will start normally and you can check the avatar version in the system info menu.

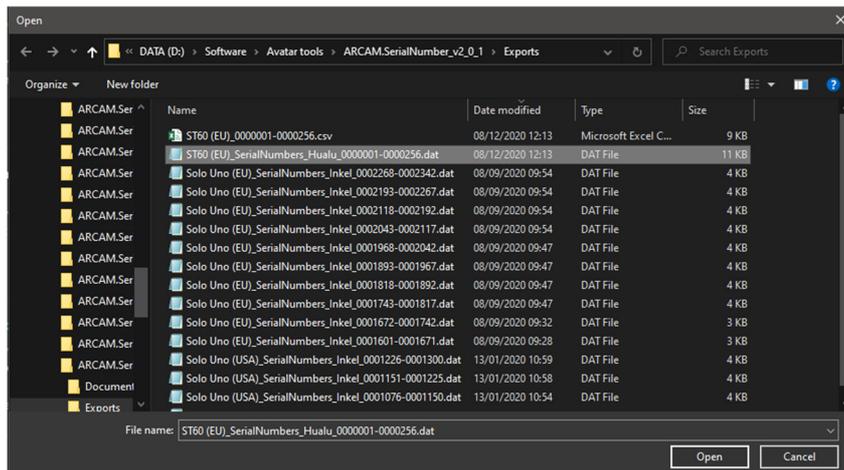
This completes the application layer programming.

Google Cast Certificate, Serial Number, MAC address and Region Programming

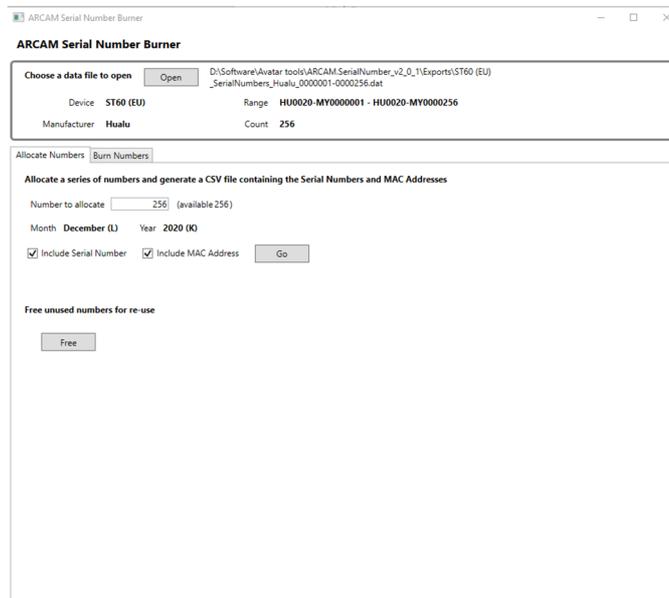
The unit needs a serial number and MAC address programming, assigned a region and a google cast certificate installed.

This is done using the Arcam Serial Number Burner application.

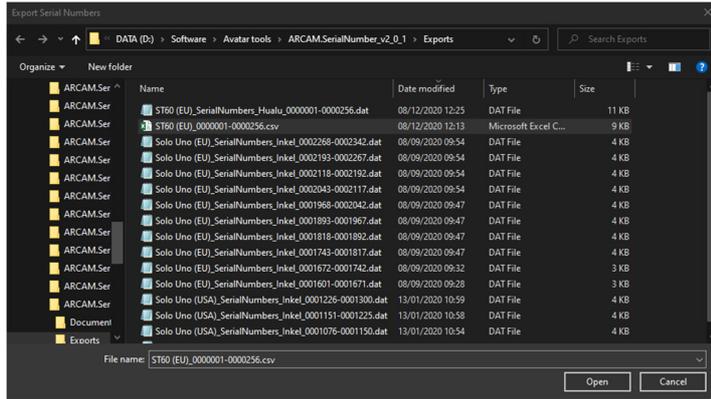
1. When opening the program you will be prompted to choose a serial number dat file.



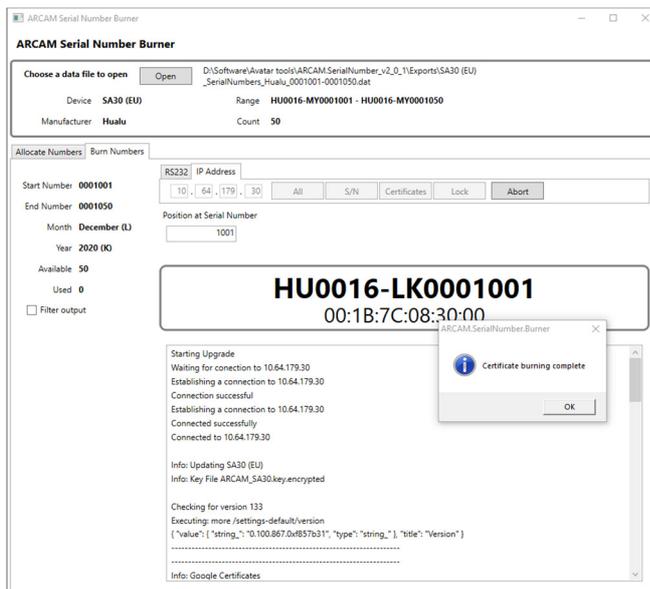
2. After selecting the dat file allocate the required number of serial numbers and click go.



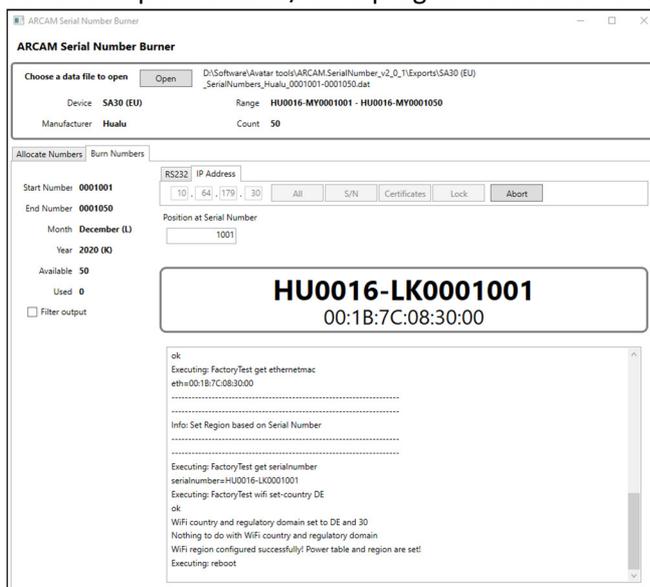
3. Select the output .csv file.



4. Connect either via RS232 or IP and click “Certificates” to install the google certificate.



5. When complete click “S/N” to program the serial number, MAC address and wifi region.



Note that when the MAC address is programmed the IP address will be changed so the burner may not reconnect.

Locking the Module

This procedure is irreversible so must only be done after ALL previous steps have been completed.

1. Once all previous steps are complete click “Lock” this will lock the module ready for shipping.