

# A38X U-Boot

[a38x](#), [u-boot](#), [uboot](#), [bootloader](#)

## Download Binaries

We are automatically building binaries whenever code is pushed to [our U-Boot repository on github](#), currently tracking the **v2018.01-solidrun-a38x** branch. Please find the results at <https://images.solid-build.xyz/A38X/U-Boot/>.

## Installing automatically (SPI, eMMC, M.2 SSD)

**This section assumes that you already have a version of U-Boot >= 2018.01 running on your device! If not, there are two options:**

1. Install U-Boot to removable media first, such as microSD or SATA
2. Boot from UART ([booting from uart](#))

First prepare an sdcard or usb-drive with the u-boot binary that you want to install: - Filesystem should be one of ext2,3,4 and fat, on partition 1 - copy the u-boot binary to the top level directory of your sdcard or usb drive - Eject the drive and plug it into your clearfog board

Now drop to the U-Boot console and run one of these update commands (you might have to substitute the file names \*.kwb by the actual names on your drive):

```
# To install u-boot-spl-spi.kwb from sdcard to spi:
bubt u-boot-spl-spi.kwb spi mmc
# To install u-boot-spl-spi.kwb from usb to spi
bubt u-boot-spl-spi.kwb spi usb
# To install u-boot-spl-mmc.kwb from usb to emmc
bubt u-boot-spl-mmc.kwb mmc usb
# To install u-boot-spl-sata.kwb from usb to m.2 ssd
bubt u-boot-spl-sata.kwb sata usb
# To install u-boot-spl-sata.kwb from emmc/sdcard to m.2 ssd
bubt u-boot-spl-sata.kwb sata mmc
```

bubt does not take care of GPT yet. When installing u-boot to sdcard, sata or emmc data partition, an existing GPT will be broken!

You can configure where u-boot will be installed to on eMMC, the choices are the data partition, boot0 and boot1. Please refer to [this section](#) for instructions.

## Installing over network (TFTP)

**This section assumes that you already have a version of U-Boot >= 2018.01 running on your device! If not, there are two options:**

1. Install U-Boot to removable media first, such as microSD or SATA
2. Boot from UART ([booting from uart](#))

For the purpose of these instructions we make the following assumptions:

|                      |                                                       |
|----------------------|-------------------------------------------------------|
| 192.168.1.1          | TFTP server                                           |
| 192.168.1.20         | IP to be used with the clearfog board                 |
| 255.255.255.0        | netmask of the network                                |
| /u-boot-spl-sata.kwb | name and path of the U-Boot binary on the tftp server |

First power on the device, then “Hit any key to stop autoboot” on the UART console. Finally use below commands to install U-Boot from TFTP to M.2 SSD. Pick the right section for your network setup. Also note that below sata can be replaced by mmc(for sdcard/eMMC) or spi for installing to those.

#### With DHCP and BOOTP

```
dhcp
bubt u-boot-spl-sata.kwb sata tftp
```

#### With DHCP only

```
dhcp
setenv serverip 192.168.1.1
bubt u-boot-spl-sata.kwb sata tftp
```

#### Manually

```
setenv ipaddr 192.168.1.20
setenv netmask 255.255.255.0
setenv serverip 192.168.1.1
bubt u-boot-spl-sata.kwb sata tftp
```

## Installing manually

### microSD

**This section assumes that you have a device running linux, and the target sdcard attached to it. This can be any device!** Optionally create an MBR partition table, and any partitions you may want.

The BootROM searches for U-Boot after the first 512 bytes, so use the dd command to write u-boot to this location on your microSD card. Substitute sdX by the device node of your sdcard.

```
dd if=u-boot-spl-sdhc.kwb of=/dev/sdX bs=512 seek=1 conv=sync
```

Note - Take your time while identifying where your designated SD-Card is mapped on your linux

system. Failure to do so can result in overwriting an arbitrary disk on your system!

## M.2 SSD

**This section assumes that you have a device running linux, and the target ssd attached to it. This can be any device!** Optionally create an MBR partition table, and any partitions you may want.

The BootROM searches for U-Boot after the first 512 bytes, so use the dd command to write u-boot to this location on your SSD. Substitute sdX by the device node of your target SSD.

```
dd if=u-boot-spl-sata.kwb of=/dev/sdX bs=512 seek=1 conv=sync
```

Note - Take your time while identifying where your designated SSD is mapped on your linux system. Failure to do so can result in overwriting an arbitrary disk on your system!

## SPI

**This section assumes that you already have a version of U-Boot >= 2018.01 running on your device! If not, there are two options:**

1. Install U-Boot to removable media first, such as microSD or SATA
2. Boot from UART ([booting from uart](#))

The BootROM loads U-boot from the start of SPI flash, offset=0. U-Boot expects to have the first 1M for itself, the environment lives at 0x0fe000-0x100000.

Drop to the U-Boot console, and execute these command for loading the u-boot binary to memory, and then writing it to the spi flash. This sample only covers eMMC/sdcard partition 1 as **source**, but network or usb are also usable.

```
ext4load mmc 0:1 0x200000 /u-boot-spl-spi.kwb
sf probe
# you may want to erase the first 1M, or just the environment:
# sf erase 0 0x100000
sf write 0x200000 0 $filesize
```

## eMMC

**This section assumes that you already have Linux running on your device! If not, there are three options:**

1. Install a system to a removable sdcard first
2. Boot a system via network from existing u-boot on your device
3. Boot U-Boot from UART ([booting from uart](#)), then boot a system via network

First remove write protection from boot0 partition:

```
echo 0 | sudo tee /sys/block/mmcblk0boot0/force_ro
```

Now write U-Boot to the start of boot0:

```
sudo dd if=u-boot-spl-mmc.kwb of=/dev/mmcb1k0boot0 conv=sync
```

It is still possible that the device will not boot from eMMC boot0. Usually in such case the eMMC has been configured to boot from boot0 or the data partition. The next section explains how this can be changed.

## Configure eMMC Boot Partition

The previous section described how U-Boot can be installed to boot0. However there are also the options to install it on the data partition, or boot1.

Responsible for changing this setting is the **mmc partconf** command. It takes either 1, or 4 paramaters:

```
# print configuration of mmc 0
mmc partconf 0
BOOT_ACK: 0x0
BOOT_PARTITION_ACCESS: 0x0
PARTITION_ACCESS: 0x0
```

The most relevant piece here is the **BOOT\_PARTITION\_ACCESS** field. It takes one of the following values:

- **0**: do not boot from eMMC
- **1**: boot from boot0
- **2**: boot from boot1
- **7**: boot from data partition

PARTITION\_ACCESS is supposed to control access to the boot partitions where 0 means no access, 1 means read-write for boot0 and 2 read-write for boot1. However this currently does not appear to have any effect. It is suggested to set this to 1 when booting from boot0, 2 when booting from boot1 and 0 when booting from the data partition.

So finally this is how a new configuration is set:

```
# configure mmc 0 to boot from, and enable access to, boot0
mmc partconf 0 0 1 1
# configure mmc 0 to boot from, and enable access to, boot1
mmc partconf 0 0 2 2
# configure mmc 0 to boot from the data partition, and disable access to
boot partitions
mmc partconf 0 0 7 0
```

## Booting from UART

**This section assumes that you have downloaded and compiled the U-Boot sources so that**

### the binary tools/kwboot exists!

First configure the board to boot from UART. There are two options to do so:

1. Set the Boot DIP switches to boot from UART ([Bootdevice](#)).
2. Set the Boot DIP switches to something that is **not** connected, e.g. SD/eMMC, or SATA. The BootROM will fall back to UART after a few seconds

After connecting the serial uart to your PC using a micro-USB-cable, run this command to send U-Boot to the board for execution:

```
./tools/kwboot -t -b u-boot-spl-uart.kwb -B 115200 /dev/ttyUSB0
```

Now turn on, or reset the board. You should see output similar to the following after less than 10 seconds:

```
./tools/kwboot -t -b u-boot-spl-uart.kwb -B 115200 /dev/ttyUSB2
Sending boot message. Please reboot the target...\
Sending boot image...
 0 %
[.....]
 2 %
[.....]
 4 %
[.....]
 6 %
[.....]
 8 %
[.....]
 9 %
[.....]
11 %
[.....]
13 %
[.....]
15 %
[.....]
17 %
[.....]
19 % [.....]
U-Boot SPL 2018.01-00060-g2fde99bd69-dirty (Apr 03 2018 - 16:43:31)
High speed PHY - Version: 2.0
Detected Device ID 6828
board SerDes lanes topology details:
| Lane # | Speed | Type      |
-----|-----|-----|
| 0      | 3     | SATA0    |
| 1      | 0     | SGMII1   |
| 2      | 5     | PCIE1    |
| 3      | 5     | USB3 HOST1 |
| 4      | 5     | PCIE2    |
```

```
| 5 | 0 | SGMII2 |
-----
:** Link is Gen1, check the EP capability
PCIe, Idx 1: remains Gen1
PCIe, Idx 2: detected no link
High speed PHY - Ended Successfully
DDR3 Training Sequence - Ver TIP-1.29.0
DDR3 Training Sequence - Switching XBAR Window to FastPath Window
DDR3 Training Sequence - Ended Successfully
...]
21 %
[.....]
23 %
[.....]
25 %
[.....]
27 %
[.....]
29 %
[.....]
31 %
[.....]
33 %
[.....]
35 %
[.....]
37 %
[.....]
39 %
[.....]
41 %
[.....]
43 %
[.....]
45 %
[.....]
47 %
[.....]
49 %
[.....]
51 %
[.....]
53 %
[.....]
55 %
[.....]
57 %
[.....]
59 %
```

```
[.....]
61 %
[.....]
63 %
[.....]
65 %
[.....]
67 %
[.....]
69 %
[.....]
71 %
[.....]
73 %
[.....]
75 %
[.....]
77 %
[.....]
79 %
[.....]
81 %
[.....]
83 %
[.....]
85 %
[.....]
87 %
[.....]
89 %
[.....]
91 %
[.....]
93 %
[.....]
95 %
[.....]
97 %
[.....]
99 % [.....]
[Type Ctrl-\ + c to quit]
```

U-Boot 2018.01-00060-g2fde99bd69-dirty (Apr 03 2018 - 16:43:31 +0200)

SoC: MV88F6828-A0 at 1600 MHz  
DRAM: 1 GiB (800 MHz, 32-bit, ECC not enabled)  
MMC: mv\_sdh: 0  
Using default environment

PCI:

```
00:01.0      - 168c:003c - Network controller
Model: SolidRun Clearfog
Board: SolidRun ClearFog Pro
SCSI: MVEBU SATA INIT
SATA link 0 timeout.
AHCI 0001.0000 32 slots 2 ports 6 Gbps 0x3 impl SATA mode
flags: 64bit ncq led only pmp fbss pio slum part sxs
Net:
Warning: ethernet@70000 (eth1) using random MAC address - d2:63:b4:9c:1d:0e
eth1: ethernet@70000
Warning: ethernet@30000 (eth2) using incremented MAC address -
d2:63:b4:9c:1d:0f
, eth2: ethernet@30000
Warning: ethernet@34000 (eth3) using incremented MAC address -
d2:63:b4:9c:1d:10
, eth3: ethernet@34000
Hit any key to stop autoboot:  0
```

To leave the mini terminal of the kwboot command, press Control and backspace (Ctrl + \), then c.

## Compiling from source

**This section assumes that you have git, make and a cross-compiler targeting 32-bit arm available on your system!**

These are the instructions to fetch the code, and build a binary:

```
git clone --branch v2018.01-solidrun-a38x
https://github.com/SolidRun/u-boot.git u-boot-clearfog
cd u-boot-clearfog
export CROSS_COMPILE=<Set toolchain prefix to your toolchain>
# optionally add options to configs/clearfog_defconfig
make clearfog_defconfig
# optionally configure u-boot graphically
# make menuconfig
make
```

This will generate u-boot-spl-sdhc.kwb to be used on the Clearfog Pro when booting from an sdcard. To target the Clearfog Base and/or other boot media, set the following options in `configs/clearfog_defconfig` or through menuconfig:

- Clearfog Pro (default)

```
CONFIG_TARGET_CLEARFOG=y
```

- Clearfog Base

```
CONFIG_TARGET_CLEARFOG_BASE=y
```

- SD-Card (default)



```
CONFIG_MVEBU_SPL_BOOT_DEVICE_SDHC=y
```

- SPI

```
CONFIG_MVEBU_SPL_BOOT_DEVICE_SPI=y
```

- eMMC

```
CONFIG_MVEBU_SPL_BOOT_DEVICE_MMC=y
```

- UART

```
CONFIG_MVEBU_SPL_BOOT_DEVICE_UART=y
```

- M.2 SSD

```
CONFIG_MVEBU_SPL_BOOT_DEVICE_SATA=y
```

Note: The resulting binaries will carry the respective -sdhc/-spi/-emmc/-uart suffixes in the name.

## Reconfigure PCIe as SATA, and SFP speed

These settings are exposed via the u-boot configuration system, and can be set in configs/clearfog\_defconfig **before running `make clearfog_defconfig`**, or **afterwards using `make menuconfig`**.

- Defaults: both ports PCIe, SFP at 1Gbps

```
CONFIG_CLEARFOG_CON2_PCI=y
CONFIG_CLEARFOG_CON3_PCI=y
CONFIG_CLEARFOG_SFP_1GB=y
```

- both ports SATA:

```
CONFIG_CLEARFOG_CON2_SATA=y
CONFIG_CLEARFOG_CON3_SATA=y
```

- SFP at 2.5Gbps

```
CONFIG_CLEARFOG_SFP_25GB=y
```

Any combinations are valid depending on your particular needs.

## Verified Boot

Verified Boot is a way to ensure that only authenticated code will be executed on a machine. This page provides instructions on setting this up for the startup phase from u-boot to Linux.

Read more here: [Verified Boot](#)

## Setup Mac-Address

The A38x MicroSoms haven't got any fixed or prefused Mac-Addresses. On each power-on a random Mac Address is generated:

```
U-Boot 2018.01 (Mar 19 2018 - 15:48:44 +0000)

SoC:   MV88F6828-A0 at 1600 MHz
DRAM:  1 GiB (800 MHz, 32-bit, ECC not enabled)
MMC:   mv_sdh: 0
SF: Detected w25q32bv with page size 256 Bytes, erase size 4 KiB, total 4 MiB
*** Warning - bad CRC, using default environment

Model: SolidRun Clearfog A1
Board: SolidRun ClearFog
SCSI:  MVEBU SATA INIT
Target spinup took 0 ms.
AHCI 0001.0000 32 slots 2 ports 6 Gbps 0x3 impl SATA mode
flags: 64bit ncq led only pmp fbss pio slum part sxs
Net:
Warning: ethernet@70000 (eth1) using random MAC address - d2:63:b4:96:1c:cb
eth1: ethernet@70000
Warning: ethernet@30000 (eth2) using incremented MAC address -
d2:63:b4:96:1c:cc
, eth2: ethernet@30000
Warning: ethernet@34000 (eth3) using incremented MAC address -
d2:63:b4:96:1c:cd
, eth3: ethernet@34000
Hit any key to stop autoboot:  0
```

The easiest way to avoid this is by saving the U-Boot environment **once** from the U-Boot console:

```
saveenv
```

If you instead want to use specific MAC addresses, they can be set per interface using these U-Boot commands:

```
setenv eth1addr c2:d8:c5:2d:92:0e
setenv eth2addr 2e:d7:af:12:e1:96
setenv eth3addr 92:67:67:88:6d:13
saveenv
reset
```

The next bootlog should show up like this:

```
Model: SolidRun Clearfog A1
Board: SolidRun ClearFog
```

```

SCSI: MVEBU SATA INIT
Target spinup took 0 ms.
AHCI 0001.0000 32 slots 2 ports 6 Gbps 0x3 impl SATA mode
flags: 64bit ncq led only pmp fbss pio slum part sxs
Net: eth1: ethernet@70000, eth2: ethernet@30000, eth3: ethernet@34000
Hit any key to stop autoboot: 0

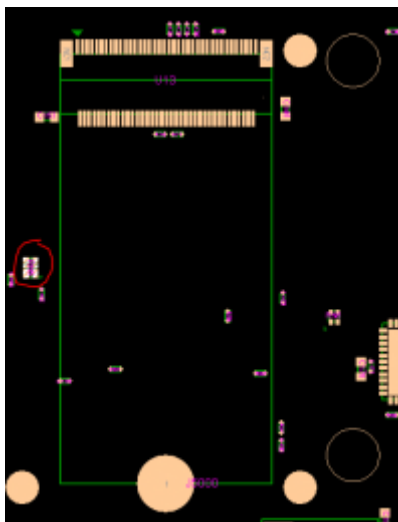
```

By removing the RTC-Battery the U-Boot environment, and with it the mac addresses are reset!

# Modifications

## Hardware Mod for Clearfog-A1 Rev-2.0 M.2 SSD

1. remove RN5 (resistor array number 5) from the board. RN5 can be found near the M.2 connector on the bottom side of the board.



2. RN5 is there first place in order to force pull-up B2B\_MPP57 (SPI Clock) on a pin on Mikrobus since one of it's pins is also used as boot select reset strap. If Mikrobus is not used then this can be removed without worry. If Mikrobus is used then make sure that B2B\_MPP57 is not pulled up or down.

## Modifying a 32bit DDR bus A388 MicroSOM to utilize only 16bit DDR

A developer that wants to evaluate the performance when using 16bit DDR bus width (like in the base MicroSOM) then the following patch on u-boot can accomplish that.

With this patch only one DDR device is being used as x16 instead of two DDR devices being used as x32 -

```

diff --git a/board/solidrun/clearfog/clearfog.c
b/board/solidrun/clearfog/clearfog.c
index 34dc50d94b..e2633c52af 100644
--- a/board/solidrun/clearfog/clearfog.c
+++ b/board/solidrun/clearfog/clearfog.c
@@ -106,7 +106,7 @@ static struct hws_topology_map board_topology_map = {
     HWS_TEMP_LOW,          /* temperature */

```

```
        HWS_TIM_DEFAULT} },          /* timing */
    5,                                /* Num Of Bus Per Interface*/
-   BUS_MASK_32BIT                    /* Busses mask */
+   BUS_MASK_16BIT                    /* Busses mask */
};

struct hws_topology_map *ddr3_get_topology_map(void)
```

## Supporting 2GByte memory configuration

It is possible to order from SolidRun 2GByte memory configuration where the support is using twin die memory configuration.

Twin die is a configuration of DDR components where there are two DDR dies in the same package and each gets it's own chip-select control.

```
diff --git a/board/solidrun/clearfog/clearfog.c
b/board/solidrun/clearfog/clearfog.c
index 34dc50d94b..aed274e941 100644
--- a/board/solidrun/clearfog/clearfog.c
+++ b/board/solidrun/clearfog/clearfog.c
@@ -93,11 +93,11 @@ int hws_board_topology_load(struct serdes_map
**serdes_map_array, u8 *count)
    static struct hws_topology_map board_topology_map = {
        0x1, /* active interfaces */
        /* cs_mask, mirror, dqs_swap, ck_swap X PUPs */
-   { { { {0x1, 0, 0, 0},
-         {0x1, 0, 0, 0},
-         {0x1, 0, 0, 0},
-         {0x1, 0, 0, 0},
-         {0x1, 0, 0, 0} },
+   { { { {0x3, 0, 0, 0},
+         {0x3, 0, 0, 0},
+         {0x3, 0, 0, 0},
+         {0x3, 0, 0, 0},
+         {0x3, 0, 0, 0} },
        SPEED_BIN_DDR_1600K, /* speed_bin */
        BUS_WIDTH_16,       /* memory_width */
        MEM_4G,             /* mem_size */
```

# Legacy U-Boot (2013.01)

## Customizing u-boot for a custom board

The easiest method to customize u-boot for a custom board using the A38x MicroSOM is to start with

the clearfog board configuration and modifying it to the custom board.

The following are high level instructions how to do that -

- Read the pin muxing from the [A38X Documents](#)
- Every MPP can be configured to the required function in the following code - [https://github.com/SolidRun/u-boot-armada38x/blob/u-boot-2013.01-15t1-clearfog/board/mv\\_ebu/a38x/armada\\_38x\\_family/boardEnv/mvBoardEnvSpec38x.h#L102](https://github.com/SolidRun/u-boot-armada38x/blob/u-boot-2013.01-15t1-clearfog/board/mv_ebu/a38x/armada_38x_family/boardEnv/mvBoardEnvSpec38x.h#L102)
- Modifying the high speed SERDES lines is done here - [https://github.com/SolidRun/u-boot-armada38x/blob/u-boot-2013.01-15t1-clearfog/tools/marvell/bin\\_hdr/src\\_phy/a38x/mvHighSpeedTopologySpec-38x.c#L89](https://github.com/SolidRun/u-boot-armada38x/blob/u-boot-2013.01-15t1-clearfog/tools/marvell/bin_hdr/src_phy/a38x/mvHighSpeedTopologySpec-38x.c#L89)
- Modifying the L2 switch (if it's available), number of Ethernet ports, I/O expander etc... is done here - [https://github.com/SolidRun/u-boot-armada38x/blob/u-boot-2013.01-15t1-clearfog/board/mv\\_ebu/a38x/armada\\_38x\\_family/boardEnv/mvBoardEnvSpec38x.c#L214](https://github.com/SolidRun/u-boot-armada38x/blob/u-boot-2013.01-15t1-clearfog/board/mv_ebu/a38x/armada_38x_family/boardEnv/mvBoardEnvSpec38x.c#L214)

From:  
<https://wiki.solid-run.com/> - **Wiki | SolidRun**

Permanent link:  
<https://wiki.solid-run.com/doku.php?id=products:a38x:software:development:u-boot>

Last update: **2018/08/07 10:09**

