

# armbian

Espressobin UART recovery

Steps:

1. Use miniterm:

```
miniterm --eol CR /dev/ttyUSB0 115200
```

rather than minicom. Minicom would interfere with the transfer from WtpDownload by trying to reopen the serial port once it had lost it; and then WtpDownload\_linux would get interrupted.

2. Set the jumpers for UART recovery and powered the board.

3. In miniterm hit enter until you got the E prompt, then entered "wtp".

4. Then run the upload:

```
<path>/<to>/WtpDownload_linux -P UART -C 0 -R 115200 -B TIM_ATF.bin -I  
wtmi_h.bin -I boot-image_h.bin -E
```

5. If you have bootable SD card in, the just-downloaded bootloader in RAM will quickly go and start Linux from the SD card. If your EspressoBIN is borked, you probably should just remove any boot device. You need to get into u-boot which can be done by pressing enter a number of times to interrupt the boot sequence.

6. Then plug in a USB stick with an u-boot tailored for your board. (flash-image-2g-2cs-1000\_800.bin for example)

7. Flash to SPI:

```
Marvell>> bibt flash-image-2g-2cs-1000_800.bin spi usb  
Burning U-BOOT image "flash-image-2g-2cs-1000_800.bin" from "usb" to  
"spi"  
USB0: Register 2000104 NbrPorts 2  
Starting the controller  
USB XHCI 1.00  
USB1: USB EHCI 1.00  
scanning bus 0 for devices... 1 USB Device(s) found  
scanning bus 1 for devices... 2 USB Device(s) found  
reading flash-image-2g-2cs-1000_800.bin  
Image checksum...OK!  
SF: Detected mx25u3235f with page size 256 Bytes, erase size 64 KiB,  
total 4 MiB  
  
Updating, 8% 126859 B/s  
Updating, 16% 127704 B/s  
Updating, 31% 167458 B/s  
Updating, 39% 155994 B/s  
262144 bytes written, 593344 bytes skipped in 2.249s, speed 389342 B/s  
  
Done!
```

8. Then power the board off, switch the boot jumpers back to their default position, and power it on again. The board shall boot Armbian from SD card or any other media as usual.