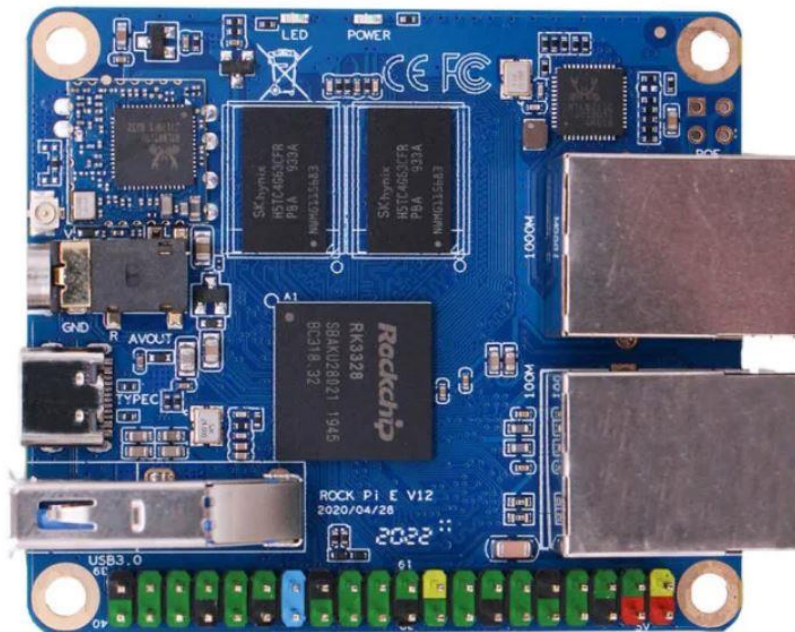


MaxAir Technical – Using the Rock Pi E Board

The Rock Pi E is based on the Rockchip RK3328 SoC (system on a chip), it can be purchased in a number of configurations.



Rock Pi E specifications:

- SoC – Rockchip RK3328 quad-core Arm Cortex-A53 processor @ up to 1.3 GHz
- System Memory – 256MB, 512MB, 1GB, or 2GB DDR3
- Storage – MicroSD card slot up to 128GB, eMMC module support (16GB / 32GB / 64GB / 128GB)
- Video & Audio – 3.5mm AV output jack (Not 100% sure whether composite video is supported)
- Connectivity
 - 1x Gigabit Ethernet port with PoE support via an additional HAT
 - 10/100M Ethernet
 - Wireless
 - 2.4 GHz 802.11 b/g/n Wifi 4 and Bluetooth 4.2 via RTL8723DU **OR**
 - Dual-band 802.11b/g/n/ac WiFi 5 and Bluetooth 4.2 via RTL8821CU
- USB – 1x USB 3.0 Type-A host port
- Expansion – 40-pin expansion header with 1x USB 2.0, 1x I2C, 1x SPI, 2x UART, 1x I2S, 5V, 3.3V and GND signals
- Misc – Reset key, Maskrom key, LEDs
- Power Supply – 5V via USB-C port
- Dimensions – 56 x 65mm

A significant plus for this board is that it can be equipped with an eMMC memory module using sockets on the reverse side of the board.

Typical cost for the 1MB RAM version with a 16GB eMMC module and dual-band wireless is around £40 and the performance is similar to the Raspberry Pi 4.

Setting up MaxAir using the Armbian operating system image did raise a number of issues relating the SPI, 1-Wire and UART interfaces.

SPI Issues

After installing Armbian from image file 'Armbian_21.05.1_Rockpi-e_buster_current_5.10.35.img', the device tree overlays spi-spidev and w1-gpio were found to be non-operational.

The issues were due to the fact that these overlays had been originally developed to support the RK3399 rather than the RK3328 used by the Rock Pi E.

The overlay files can be found in directory **/boot/dtb/rockchip/overlay/** these files are in dtbo format eg. **rockchip-spi-spidev.dtbo**, they need to be converted to dts format in order that they can be edited. To edit the dtbo files, install the Device-Tree-Compiler using '**sudo apt-get install device-tree-compiler**'.

spidev

There are two issues with this overlay, the first is that it references RK3399 and not RK3328, the second is that it maps spi0 to ff1c0000, whereas it should be ff190000.

Create the dts file by executing the command

```
dtc -I dtb -O dts /boot/dtb/rockchip/overlay/rockchip-spi-spidev.dtbo -o /boot/dtb/rockchip/overlay/rockchip-spi-spidev.dts
```

Edit the file **/boot/dtb/rockchip/overlay/rockchip-spi-spidev.dts**, delete the contents and replace with the following :

```
/dts-v1/;

/ {
    compatible = "rockchip,rk3328";

    fragment@0 {
        target-path = "/aliases";

        __overlay__ {
            spi0 = "/spi@ff190000";
        };
    };

    fragment@1 {
        target = < 0xffffffff >;

        __overlay__ {
            #address-cells = < 0x01 >;
            #size-cells = < 0x00 >;

            spidev {
                compatible = "spidev";
                status = "disabled";
                reg = < 0x00 >;
                spi-max-frequency = < 0x989680 >;
            };
        };
    };

    __fixups__ {
        spi0 = "/fragment@1:target:0";
    };
};
```

Save the file ***/boot/dtb/rockchip/overlay/rockchip-spi-spidev.dts*** and recompile the dtbo using the command:

```
dtc -I dts -O dtb /boot/dtb/rockchip/overlay/rockchip-spi-spidev.dts -o
/boot/dtb/rockchip/overlay/rockchip-spi-spidev.dtbo
```

It is also necessary to update the file `/boot/dtb/rockchip/overlay/rockchip-fixup.scr`

Copy file /boot/dtb/rockchip/overlay/rockchip-fixup.scr to
/boot/dtb/rockchip/overlay/rockchip-fixup.script

Edit file /boot/dtb/rockchip/overlay/rockchip-fixup.script

Delete everything from the top of the file before **# overlays fixup script** ie.

[illegible]

Becomes

```
# overlays fixup script
# implements (or rather substitutes) overlay arguments functionality
# using u-boot scripting, environment variables and "fdt" command
```

Change every occurrence of spi@ff1c0000 to spi@ff190000

Save the file `/boot/dtb/rockchip/overlay/rockchip-fixup.script` and recompile using

```
mkimage -A arm -T script -C none -d /boot/dtb/rockchip/overlay/rockchip-fixup.script
/boot/dtb/rockchip/overlay/rockchip-fixup.scr
```

Finally edit file `/boot/armbianEnv.txt` and add the following two lines:

```
overlays=spi-spidev
param_spidev_spi_bus=0
```

1-Wire Issues

w1-gpio

There are two issues with this overlay, the first is that it references RK3399 and not RK3328, the second is that it maps to the wrong GPIO pin.

The following instructions will enable the 1-wire interface on same physical pin used by the Raspberry Pi, ie physical pin 7, which is GPIO1_D4 on the Rock Pi E. This pin is located on gpio bank 1 and is the 28th pin on that bank.

Create the dts file by executing the command :

```
dtc -I dtb -O dts /boot/dtb/rockchip/overlay/rockchip-w1-gpio.dtbo -o  
/boot/dtb/rockchip/overlay/rockchip-w1-gpio.dts
```

Edit the file **/boot/dtb/rockchip/overlay/rockchip-w1-gpio.dts**, delete the contents and replace with the following :

```
/dts-v1/;  
  
/{  
    compatible = "rockchip,rk3328";  
  
    fragment@0 {  
        target-path = [ 2f 00 ];  
  
        __overlay__ {  
  
            onewire@0 {  
                compatible = "w1-gpio";  
                pinctrl-names = "default";  
                gpios = < 0xffffffff 0x1c 0x00 >;  
                status = "okay";  
                phandle = < 0x01 >;  
            };  
        };  
    };  
  
    __symbols__ {  
        w1 = "/fragment@0/__overlay__/onewire@0";  
    };  
  
    __fixups__ {  
        gpio1 = "/fragment@0/__overlay__/onewire@0:gpios:0";  
    };  
};
```

Save the file **/boot/dtb/rockchip/overlay/rockchip-w1-gpio.dts** and recompile using:

```
dtc -I dts -O dtb /boot/dtb/rockchip/overlay/rockchip-w1-gpio.dts -o  
/boot/dtb/rockchip/overlay/rockchip-w1-gpio.dtbo
```

Finally edit file **/boot/armbianEnv.txt** and add change the following line:

overlays=spi-spidev to overlays=spi-spidev w1-gpio

UART Issues

Uart2

Uart2 TXD is found on pin8 and RXD on pin10, which is compatible with the Raspberry Pi. By default these pins are used by the serial console, hence if you want to use Uart2, for example, to interface with a 'Serial Gateway', then the serial console needs to be disabled. In order to do this:

1. Edit file `/boot/armbianEnv.txt` and delete the line **console=serial**
2. Execute the command **systemctl mask serial-getty@ttyS2.service**
3. Reboot