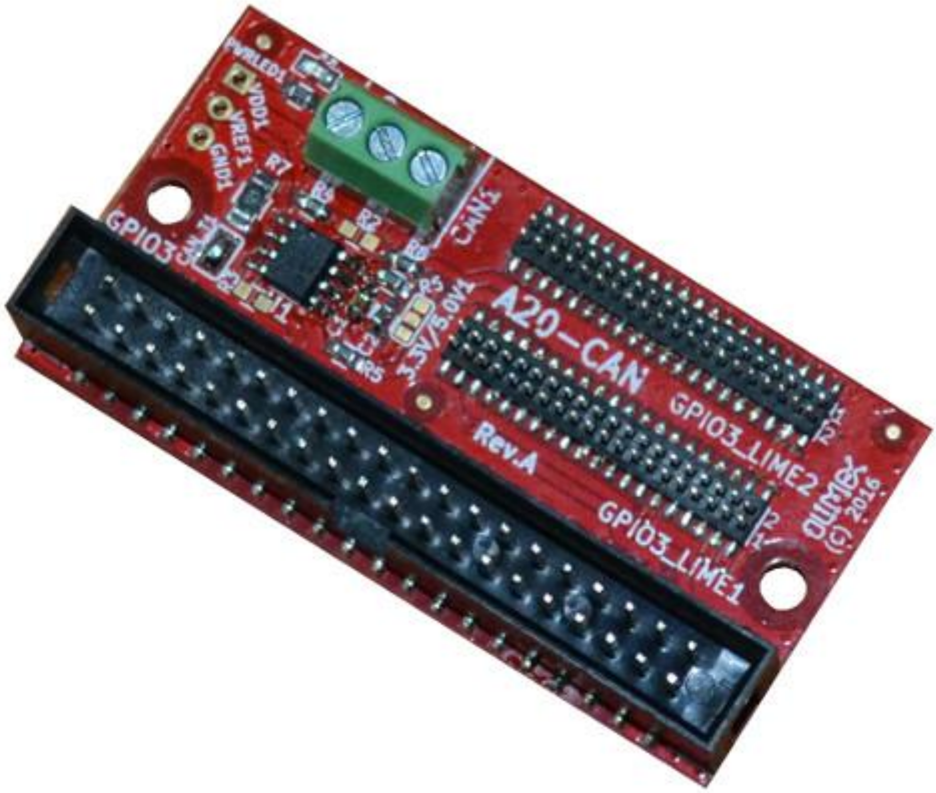
A20-CAN



A20-CAN is CAN driver board for A20 Allwinner SOCs. A20-CAN features are:

- compatible with A20-OLinuXino-LIME, A20-OLinuXino-LIME2, A20-OLinuXino-MICRO, A20-SOM-EVB
- connects to GPIO3 connector
- allow CAN networking of A20 boards running Linux
- Power LED
- PCB dimensions: 60x30mm

Contents

Building Linux with CAN support

Get sources

Download kernel tree from:

```
git clone https://github.com/linux-sunxi/linux-sunxi.git
```

For Linux-Sunxi kernel (3.4)

The can driver is not in the kernel tree. This is done with the following steps:

Download the can driver:

```
clone https://github.com/btolfa/sunxi-can-driver.git
```

Copy file to kernel tree:

```
cp -v sunxi-can-driver/sun7i_can.* linux-sunxi/drivers/net/can
```

Add Kconfig selection:

```
vi linux-sunxi/drivers/net/can/Kconfig
```

Add the following:

```
config CAN_SUN7I
tristate "Sun7i CAN bus controller"
default n
help
This is the Sun7i CAN BUS driver for android system by peter chen.
```

Modify Makefile:

```
vi linux-sunxi/drivers/net/can/Makefile
```

Add:

```
obj-$(CONFIG_CAN_SUN7I) += sun7i_can.o
```

NOTE: The example build is with default sun7i_defconfig. You can use your own. Now you device tree is ready for build. However the new module is not selected.

Modify kernel:

```
cd linux-sunxi
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- -j4 sun7i_defconfig
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- -j4 menuconfig
```

Select from:

```
[*] Networking support  --->
```

```
    <*>   CAN bus subsystem support  --->
    <*>   CAN Device Drivers   --->
         [M]   Sun7i CAN bus controller
```

Save current configuration and build:

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- -j4 uImage modules
```

Modify can section of script.bin file:

```
[can_para]
can_used = 1
can_tx = port:PH20<4><default><default><default>
can_rx = port:PH21<4><default><default><default>
```

Copy kernel and modules to SD-CARD and boot the system.

For Mainline kernel

In the mainline kernel there is support for CAN bus. However we need to declare that it's gonna be used.

First of all checkout **__sunxi_next__** branch:

```
cd linux-sunxi
git checkout -b sunxi-next origin/sunxi-next
```

Select **__sunxi_defconfig__**:

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- -j4 sunxi_defconfig
```

NOTE: By default can module is build as part of the kernel.
NOTE: In the example we are gonna use A20-OLinuxIno-MICRO, but all other A20 based boards can be used.
NOTE: See device-tree binding in:

```
   /compile/A20/linux-
sunxi/Documentation/devicetree/bindings/net/can/sun4i_can.txt
```

Modify common device-tree sources:

```
vi arch/arm/boot/dts/sun7i-a20.dtsi
```

Append at line 1216:

```
can0_pins_a: can0@0 {
    allwinner,pins = "PH20","PH21";
    allwinner,function = "can";
    allwinner,drive = <0>;
    allwinner,pull = <0>;
};
```

Append at line 1442:

```
   can0: can@01c2bc00 {
       compatible = "allwinner,sun4i-a10-can";
       reg = <0x01c2bc00 0x400>;
       interrupts = <0 26 4>;
       clocks = <&apb1_gates 4>;
       status = "disabled";
   };
```

Open arch/arm/boot/dts/sun7i-a20-olinuxino-micro.dts:

```
   vi arch/arm/boot/dts/sun7i-a20-olinuxino-micro.dts
```

Append at the bottom:

```
   &can0 {
       pinctrl-names = "default";
       pinctrl-0 = <&can0_pins_a>;
       status = "okay";
   };
```

Build everything needed:

```
   make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- -j4 zImage dtbs modules
```

Copy kernel, module and dtb to sd-card

CAN bus usage

Now when everything is ready we need to start using the CAN bus.

You should see __can0__ device with ifconfig:

```
   ifconfig -a

   can0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-
00-00-00-00
             NOARP  MTU:16  Metric:1
             RX packets:0 errors:0 dropped:0 overruns:0 frame:0
             TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:10
             RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
             Interrupt:51
```

If you don't make sure that module is loaded:

```
   modprobe sun7i_can
```

Get some tools:

```
   apt-get install can-utils
```

Configure can bus

```
   ip link set can0 down
   ip link set can0 type can bitrate 100000 triple-sampling on loopback off
   ip link set can0 up
```

Now CAN is set to 100kbps. Connect to some you other can devices.

Send simple message:

```
   cansend can0 5A1#A5
```

For more infomation about can-utils usage:
http://www.armadeus.com/wiki/index.php?title=CAN_bus_Linux_driver