



PCI Express SDR Board

Version: 2018-07-10

Table of Contents

1	Features	1
2	Installation with the Amarisoft software	2
2.1	Introduction	2
2.2	Automatic Installation	2
2.3	Manual Installation	2
2.4	Driver initialization	3
2.5	Firmware upgrade	3
2.6	Multiple card installation	3
2.7	TRX driver configuration options	4
2.8	TDD support	5
2.9	GPS usage	5
2.10	Oscillator frequency fine tuning	5
3	Troubleshooting	6
4	Miscellaneous utilities	7
4.1	sdr_util	7
4.2	sdr_spectrum	7
4.3	sdr_play	8
5	C API	10
6	Physical specifications	11
6.1	Summary	11
6.2	Connectors	11
6.2.1	SMA connectors	13
6.2.2	PPS/Clock connectors	14
6.2.3	JTAG connector (J1)	14

1 Features

- 2x2 RF transceiver (AD9361) with integrated 12-bit DACs and ADCs
- RF coverage: 70 MHz to 6.0 GHz
- RF bandwidth: < 200 kHz to 56 MHz
- Supports TDD and FDD operation
- Integrated TX/RX switch for TDD operation
- PCIe 1x standard connector
- Integrated GPS for precise time and frequency synchronization
- Clock/PPS inputs and outputs for easy chaining
- Optional sample compression to reduce the PCI bandwidth

2 Installation with the Amarisoft software

2.1 Introduction

If you have bought the OTS (Off-The-Shelf) package, then you don't need to install anything. Everything has already been installed on your PC. Otherwise, please follow through the steps below.

Decompress the `trx_sdr` archive into a convenient directory specified by `<trx_path>`.

```
tar -xzf trx_sdr-linux-YYYY-MM-DD.tar.gz -C <trx_path>
```

You have two ways to install the TRX driver for the PCIe card:

- automatic
- manual

For both cases, the installation requires some specific packages to compile the kernel module. To do this, you need to be root. In Fedora and Cent OS, you need to install *kernel-devel*, *gcc* and *elfutils-libelf-devel* packages by running the following commands:

```
yum install kernel-devel-$(uname -r)
yum install gcc
yum install elfutils-libelf-devel
```

For Ubuntu, use the following commands:

```
apt-get install $(uname -a | awk '{print $3}')
```

```
apt-get install build-essential
```

Note that you'll need equivalent packages for other Linux distributions if you do not use Fedora, Ubuntu or Cent OS.

Once you have finished the installation, you need to initialize (See [Driver initialization], page 3) and upgrade your driver (See [Firmware upgrade], page 3). Please make sure to initialize the driver after each system boot if you have not activated an automatic lte service.

2.2 Automatic Installation

Automatic installation is only available on Fedora, Ubuntu and CentOS distributions. Use manual install for other distributions. To start your automatic install, use the following command where `<path>` is the path to the directory where you have already installed your LTE component (eNB or UE) and `type` should be set to `enb` or `ue` accordingly.

```
./install <path> <type>
```

Notes:

- the script would install some packages, so make sure you have root privileges when you run the script
- this install creates a symlink for the TRX driver so please do not remove this directory afterwards.

2.3 Manual Installation

To manually install the driver, let's note `<path>` the directory where Amarisoft eNB or UE software is installed. Then:

1. Compile kernel driver:

Go to the `kernel/` directory under `<trx_path>` and start compilation:

```
cd kernel
make
```

2. Place driver:

Just copy the compiled driver into `<path>` directory

```
cd ..
cp trx_sdr.so <path>
cp libsdr.so <path>
```

3. Config files:

Copy RF driver config file. Note that there are 2 separate config directories for eNB and UE called `config.enb` and `config.ue` under your `<trx_path>`. As a result, the `<config_dir>` should be set to `config.enb` or `config.ue` accordingly.

```
cp -r <config_dir> <path>/config/sdr
```

Select frontend:

```
<path>/config/rf_select.sh sdr
```

2.4 Driver initialization

Each time you boot your system, you need to perform this initialization. Note that if you are using OTS install, this step is already done by the lte service.

```
cd kernel
./init.sh
```

2.5 Firmware upgrade

Perform the following command to upgrade your PCIe card:

```
./sdr_util upgrade
```

Notes:

- If you have several PCIe cards installed, this will upgrade all cards.
- If the firmware is already up to date, this command will end directly.
- When the upgrade is over, you need your system to be powered off for the changes to take effect (not only reset, real power off).

2.6 Multiple card installation

To make several cards work together, they must be time and frequency synchronized. For that purpose you need to plug the provided black cable between each card. Connect the internal OUT connector (J4 on Figure 6.2) on the first card to the IN connector on the second card (J3 on Figure 6.2, connectors are between RoHS compliant mark and Serial number sticker). Then do the same with OUT of second card and IN of third one, etc...

When you install several PCIe cards, the mapping between the PCI connectors and the Linux devices is not predictable (but it shouldn't change after each boot). To identify the order please do the following:

```
./sdr_util -c 0 led 1
```

Then check inside PC on each board, one of them should have a led blinking. This is card 0 (`/dev/sdr0`).

Switch off the led:

```
./sdr_util -c 0 led 0
```

You can do the same for other cards:

```
./sdr_util -c <n> led 1
```

Where `<n>` is the index of the card.

To use all the cards, update eNB config file (`1chan.cfg`) using `args` parameter:

```
args: "dev0=/dev/sdrA,dev1=/dev/sdrB,..."
```

Note: this allow to change the order of the cards. For instance, if your card are from left to right have following indexes:

```
<1> <2> <0>
```

And you want first cell to be on left, second cell to be on middle and third on right, use the following:

```
args: "dev0=/dev/sdr1,dev1=/dev/sdr2,dev2=/dev/sdr0"
```

2.7 TRX driver configuration options

The default TRX configuration used by your component is called `1chan.cfg`. It could be found under the directory `config/sdr/` of your component (eNB or UE).

The following JSON properties are available:

args String. Set the system device names for the boards. Example:

```
args: "dev0=/dev/sdr1,dev1=/dev/sdr2,dev2=/dev/sdr0"
```

sync Optional enumeration: `none`, `internal`, `gps`, `external` (default = `none`). Set the time synchronization source. `none` and `internal` uses the internal PPS generated from the clock. `internal` synchronizes the RX timestamps on the internal PPS. `none` does not synchronize the RX timestamps on the internal PPS for a faster startup. `gps` uses the internal GPS. `external` uses the PPS from the IN connector (J3 on Figure 6.2).

When several cards are selected (with the `args` property), `sync` only sets the time synchronization source of the first card. The other cards are implicitly set to `external` synchronization, assuming the previous card is used as source.

clock Optional enumeration: `internal`, `external` (default = `internal`). Set the clock source. `internal` uses the internal clock (VCTCXO). If an external time source is used, the internal clock frequency is adjusted by the PPS signal. `external` uses the clock from the IN connector (J3 on Figure 6.2).

When several cards are selected (with the `args` property), `clock` only sets the clock source of the first card. The other cards are implicitly set to `external` clock, assuming the previous card is used as source.

sample_hw_fmt

Optional enumeration (`auto`, `ci16`, `cf8`) (default = `auto`). Set the sample format used on the PCI bus. `auto` selects the best format depending on the available PCI bandwidth. `ci16` selects 16 bit complex integers. `cf8` selects 8 bit complex floats giving a 2:1 compression ratio while retaining the full 12 bit amplitude of the ADCs and DACs.

rx_antenna

Optional enumeration (`auto`, `tx_rx`, `rx`) (default = `auto`). Select the connector on which the RX antenna is connected. By default it is connected on the RX connector for FDD and on the TX/RX connector for TDD. This parameter is useful to force the use of the RX connector in TDD.

The range for the transmit gain (`tx_gain` parameter in eNodeB/UE) is from 0 to 89.75 dB. The range for the receive gain (`rx_gain` parameter in eNodeB/UE) depends on the frequency:

70-1300 MHz

1 to 77 dB

1300-4000 MHz

-4 to 71 dB

4000-6000 MHz

-10 to 62 dB

The maximum sample rate is 61.44 MHz.

2.8 TDD support

The board integrates a TX/RX switch for TDD operation. When TDD operation is configured, the transmit and receive are done on a single RF port (the TX SMA connector). So you only need to connect antennas on the TX1 and TX2 SMA connectors for TDD operation.

2.9 GPS usage

You can check the GPS operation when the eNodeB/UE is stopped with

```
./sdr_util gps_state
```

The GPS takes a few minutes to lock if the GPS antenna is connected. Any active GPS antenna accepting a 3.3V DC supply can be used, for example: <http://www.mouser.fr/Search/ProductDetail.aspx?R=ANT-GPS-SH-SMAvirtualkey59000000virtualkey712-ANT-GPS-SH-SMA>

2.10 Oscillator frequency fine tuning

If you don't have a GPS, it is still possible to manually fine tune the VCTCXO (Voltage Controlled, Temperature Controlled Crystal Oscillator) frequency provided you have a way to know the offset:

```
./sdr_util clock_tune n
```

where n is the offset in PPM (parts-per-million) from the nominal TCXO frequency. Note: the PPM offset n to voltage law is only approximative, so you should adjust it by successive approximation.

3 Troubleshooting

Below are a few tips on how to handle different errors.

- The eNodeB does not start and the output of `dmesg` linux command resembles the following error message:

```
sdr: version magic '4.12.9-200.fc25.x86_64 SMP mod_unload' should be  
'4.12.11-200.fc25.x86_64 SMP mod_unload'
```

To resolve it, you need to recompile the SDR driver on the PC (See [Manual Installation], page 2).

4 Miscellaneous utilities

4.1 sdr_util

usage: sdr_util [options] cmd [args...]

Options:

-h	help
-c device_num	select the device (default = all)

Available commands:

version	dump the FPGA version
sync_state	dump the synchro and clock state
gps_state	dump the GPS state
temp	dump the temperature of the board components
led [0 1]	enable/disable led blinking
clock_tune n	tune TCXO frequency offset to n ppm
upgrade [options]	upgrade the FPGA firmware
upgrade options are:	
-force	force upgrade even if identical or previous version

4.2 sdr_spectrum

sdr_spectrum is a real time spectrum analyzer, sample viewer and I/Q file recorder. It handles several channels at the same time. It can be invoked without arguments. The following options are available:

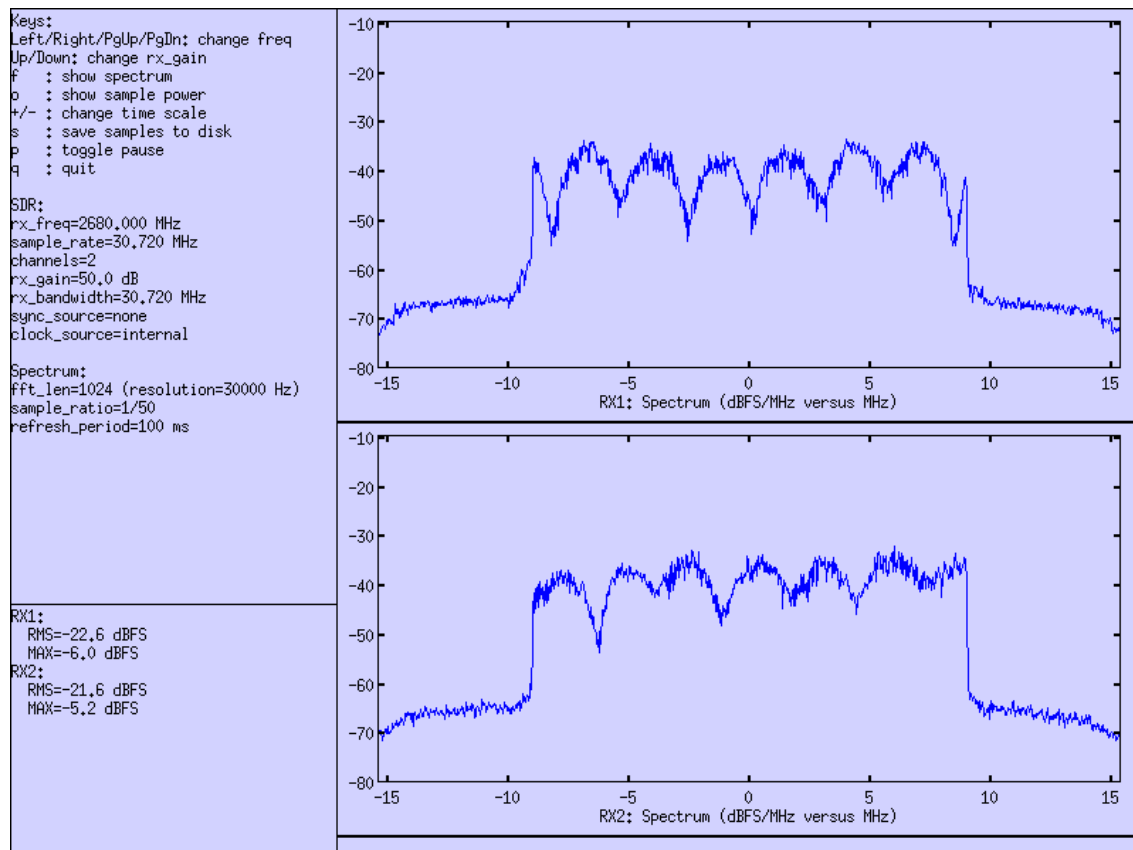
-h	help
-args str	set the device arguments (default="/dev/sdr0")
-rx_freq freq	set the RX frequency in Hz (default=2400000000)
-rate rate	set the sample rate to 'rate' Hz (default=30720000)
-rx_gain gain	set the RX gain in dB (default=60)
-channels c	set the number of RX channels to 'c' (default=1)
-rx_bw	set the analog receive bandwidth in Hz (default=same as sample rate)
-sync source	set the sync source to 'source' (none, internal, gps, external) (default=none)
-clock source	set the clock source to 'source' (internal, external) (default=internal)
-save_path path	set the directory where the recorded samples are saved (default=/tmp)
-duration d	set the recorded sample file duration in seconds when saving (default=1.0)

When saving the recorded samples, one file is generated per channel. The filename contains the UTC date, the frequency and the channel number.

Invocation example:

```
./sdr_spectrum -channels 2 -rx_gain 50 -rx_freq 2680e6 -rate 30.72e6
```

Resulting output:



4.3 sdr_play

`sdr_play` is an I/Q file player. Several channels can be played at the same time and they can be time synchronized on an external clock. One file must be provided per channel. The following options are available:

- `-h` help
- `-args str` set the device arguments (default="dev0=/dev/sdr0")
- `-tx_freq freq` set the TX frequency in Hz (default=2400000000)
- `-rate rate` set the sample rate to 'rate' Hz (default=23040000)
- `-tx_gain gain` set the TX gain in dB (default=70)
- `-channels c` set the number of channels to 'c' (default=1)
- `-tx_bw bw` set the analog transmit bandwidth in Hz (default=same as sample rate)
- `-loop` play the file in loop

-sync source
set the sync source to 'source' (none, internal, gps, external) (default=none)

-clock source
set the clock source to 'source' (internal, external) (default=internal)

-time_offset n
offset the output by n samples (default=0)

5 C API

The PCIe SDR board can be used in other projects with its C API. The C API allows to send and receive I/Q samples and to change the various parameters (frequency, sample rate, bandwidth, gains, ...). The Amarisoft TRX driver, `sdr_play` and `sdr_spectrum` are built using this API.

The C API is described in `libsdr.h`. The corresponding Linux x86_64 dynamic library is `libsdr.so`.

Amarisoft does not provide any support for this API and can modify it without notice.

6 Physical specifications

6.1 Summary

- RF power output < 10 dBm
- Max RF input power: -10 dBm
- 4 SMA female (TX1, TX2, RX1, RX2), AC coupled
- 1 SMA female (GPS antenna with 3.3V DC power supply)
- 2 internal 5-pin connectors for inter-card time synchronization
- PCIe full height, short length
- Uses single 12V power supply from PCI connector
- ADC/DAC Sample Rate: 61.44 MS/s
- ADC/DAC Resolution: 12 bits
- Frequency Accuracy: 2 ppm
- LTE 20MHz 64QAM EVM < 2 %rms (f < 2.6 GHz)
- LTE 20MHz 64QAM EVM < 4 %rms (f < 3.5 GHz)
- Full size (L x W x H): 128mm x 115mm x 20mm
- Weight: 0.100 kg

6.2 Connectors

The following figure depicts the location and functionality of each connector in the PCIe board.

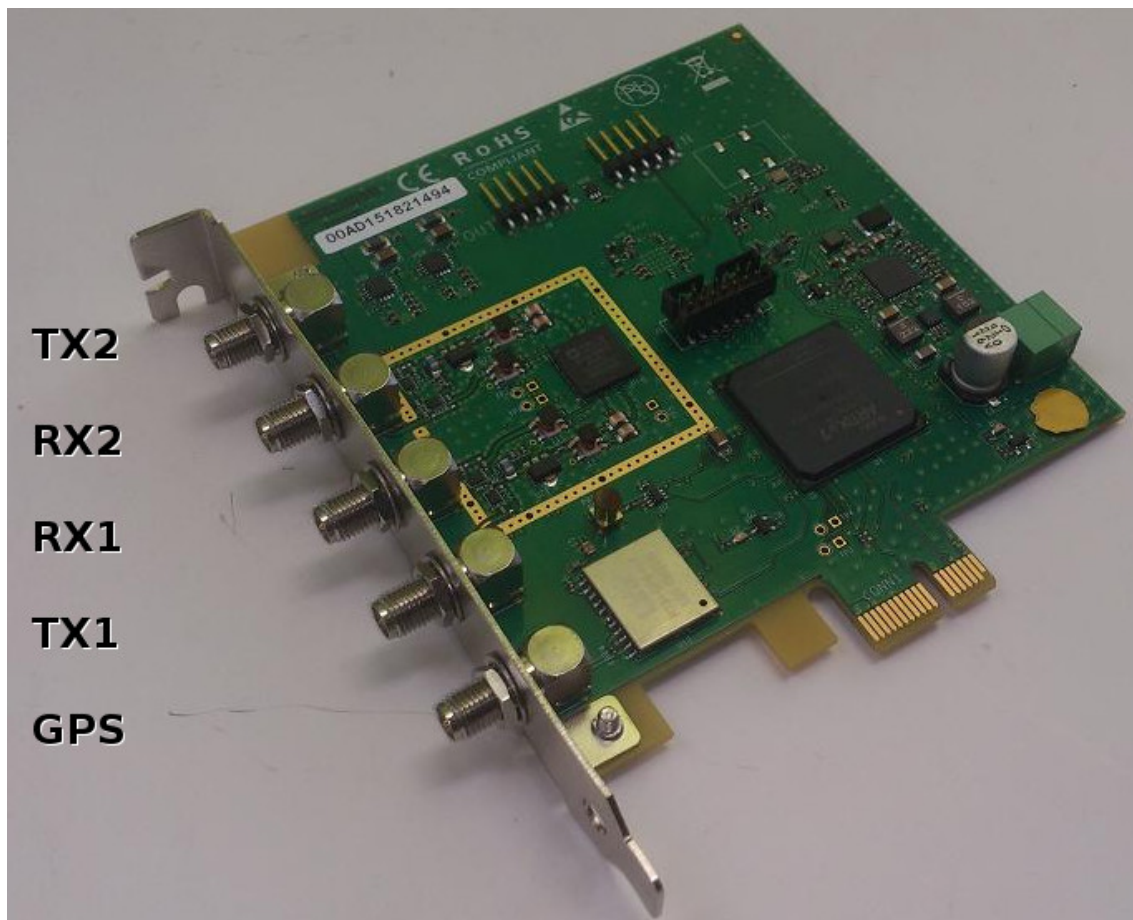


Figure 6.1

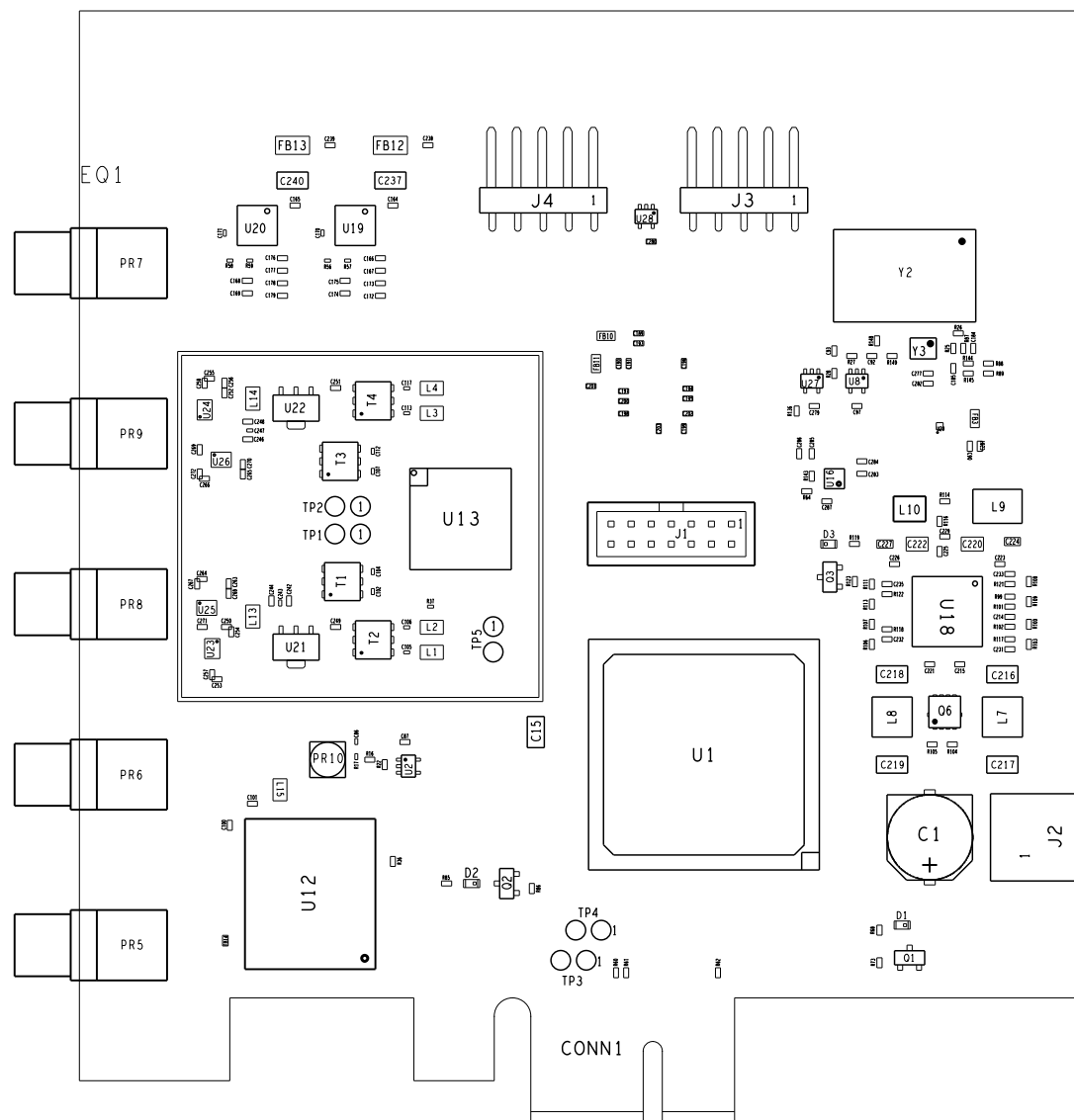


Figure 6.2

6.2.1 SMA connectors

See Figure 6.1 for the location.

On Figure 6.2, the connector names are:

PR7	TX2 (and RX2 for TDD)
PR9	RX2 (for FDD)
PR8	RX1 (for FDD)
PR6	TX1 (and RX1 for TDD)
PR5	GPS

6.2.2 PPS/Clock connectors

OUT connector (J4):

1	PPS_OUT
2	REFCLK_OUT_N
3	REFCLK_OUT_P
4,5	GND

PPS_OUT is a LVCMOS 3.3V output. REFCLK_OUT is a standard LVDS signal with AC coupling.

IN connector (J3):

1	PPS_IN
2	REFCLK_IN_N
3	REFCLK_IN_P
4,5	GND

PPS_IN is a LVCMOS 3.3V input (> 3.3V input is not accepted). REFCLK_IN is a standard LVDS signal with 100 ohm impedance and AC coupling.

The reference clock is at 38.4 MHz. 5 or 4 pin USB2 motherboard PC cables can be plugged in J3 and J4. When using a 4 pin connector, the pin 5 can be left unconnected.

6.2.3 JTAG connector (J1)

JTAG connector:

1,3,5,7,9,11,13	GND
2	VCC (3.3V)
4	JTAG_TMS
6	JTAG_TCK
8	JTAG_TDO
10	JTAG_TDI
12,14	NC