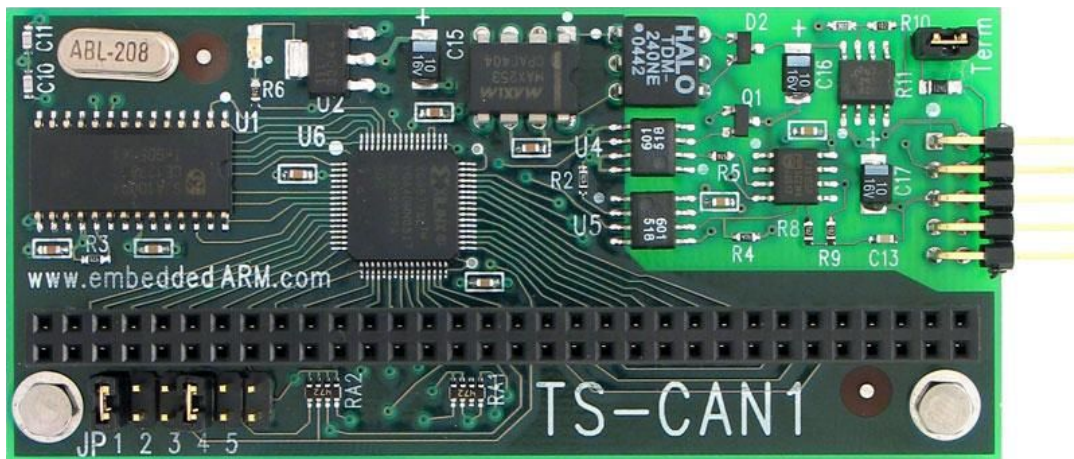


Hardware Overview

The TS-CAN1 is a PC/104 daughter board which provides one single CAN interface channel and enables a PC/104 embedded systems to be connected to CAN1.1 and CAN2.0a networks. It uses a Philips SJA1000 CAN controller which is opto-isolated from the physical CAN network, increasing the system security. On-board jumpers are used to select both the interrupt and the I/O region used. Up to four boards can be installed in any single system. The SJA1000 controller can be configured to communicate with the processor through 8 different 32-bytes I/O regions. TS-CAN1 is compatible with both X86 and ARM architectures.



Jumper settings for base address selection

| I/O Address | JP1 | JP2 |
|-------------|-----|-----|
| 150 - 157 | OFF | OFF |
| 158 - 15F | ON | OFF |
| 160 - 167 | OFF | ON |
| 168 - 16F | ON | ON |

Jumper settings for IRQ selection

| IRQ | JP4 | JP5 |
|------|-----|-----|
| None | OFF | OFF |
| IRQ6 | ON | OFF |
| IRQ7 | OFF | ON |
| IRQ5 | ON | ON |

CAN connection interface

| PIN Number | Signal Name |
|------------|-------------|
| 1 | X |
| 2 | CANL |
| 3 | GND |
| 4 | X |
| 5 | GND |
| 6 | X |
| 7 | CANH |
| 8 | X |
| 9 | X |
| 10 | X |

Register Map: TS-CAN1

| I/O Address | Description | Data | Bits and such |
|--------------------|-----------------------------|-------------|---|
| BASE + 0 | Board identifier #1 | Read only | Fixed value: 0xF6 hex |
| BASE + 1 | Board identifier #2 | Read only | Fixed value: 0xB9 hex |
| BASE + 2 | PLD version register | Read only | Fixed value |
| BASE + 3 | LED control/status register | R/W | Bit 0: LED (1=on, 0=off) |
| BASE + 4 | Page selection register | R/W | Bits 2-7: reserved Bits 0-1: select one-of-four pages for SJA1000 access. This allows all 128 registers to be accessed using only 32-bytes of I/O space. Default to all zeros. (see table "SJA1000 I/O page selection register") |
| BASE + 5 | Mode control register | R/W | Bit 7: reserved Bit 6: If set, it enables SJA1000 to appear in I/O space location determined by mode bits 0-2. Also enables page selection register at [Base + 4] to be used. Bit 5: If set, it allows sharing of IRQ6 and IRQ7. It can only be set for TS ARM platforms and sharing can only be used for IRQ6 and IRQ7. Bits 3-4: reserved Bit 2: MSB of 3-bit address select register. Bit 1: Middle of 3-bit address select register. Bit 0: LSB of 3-bit address select register. (see table "SJA1000 I/O address selection register") |
| BASE + 6 | Jumper status | Read only | Bits 6-7: reserved Bit 5: Jumper 5 (1=on, 0=off) Bit 4: Jumper 4 (1=on, 0=off) Bit 3: Jumper 3 (1=on, 0=off) Bit 2: Jumper 2 (1=on, 0=off) Bit 1: Jumper 1 (1=on, 0=off) Bit 0: reserved |
| BASE + 7 | Reserved | - | |

SJA1000 I/O page selection register

| Page and registers | Bit 1 | Bit 0 |
|---------------------------|--------------|--------------|
| Page 1: 00-1F | 0 | 0 |
| Page 2: 20-3F | 0 | 1 |
| Page 3: 40-5F | 1 | 0 |
| Page 4: 60-7F | 1 | 1 |

SJA1000 I/O address selection register

| Address range | Bit 2 | Bit 1 | Bit 0 |
|----------------------|--------------|--------------|--------------|
| 100-11F | 0 | 0 | 0 |
| 120-13F | 0 | 0 | 1 |
| 180-19F | 0 | 1 | 0 |
| 1A0-1BF | 0 | 1 | 1 |
| 200-21F | 1 | 0 | 0 |
| 240-25F | 1 | 0 | 1 |
| 280-29F | 1 | 1 | 0 |
| 320-33F | 1 | 1 | 1 |

PROGRAMMING FOR SJA1000

The TS-CAN1 was designed using the Philips SJA1000 CAN controller. The SJA1000 provides support for CAN2.0 networks and supports BasicCAN and PeliCAN modes. For further information about SJA1000 and how to interact with it through its registers, please check the specific Datasheet and Application Notes:

- <http://www.semiconductors.philips.com/pip/SJA1000.html>
- http://www.semiconductors.philips.com/acrobat/datasheets/SJA1000_3.pdf
- <http://www.semiconductors.philips.com/acrobat/applicationnotes/AN97076.pdf>

DRIVER FOR LINUX

The Linux device driver for TS-CAN1 was developed using the OCERA framework for CAN, which is composed of LINCAN, VCA, CAN/CANOpen Monitor and CANOpen Device. LINCAN is a generic linux device driver implementation that supports many boards. It is an evolution of other CAN open-source projects for Linux. The Virtual CAN API (VCA) is an interface that hides LINCAN driver specifics from user space programmer, increasing the development speed and compatibility. CAN/CANOpen Monitor is a tool (application server and Java GUI client) that enables monitoring and interaction with a CAN network using client/server applications. CANOpen Device is a library based on VCA that provides a CANOpen high-level protocol for device integration over LINCAN using CANOpen master and slave components. For further information on OCERA/LINCAN framework, please find out the links:

- <http://www.ocera.org>
- <http://cmp.felk.cvut.cz/~pisa/can/doc/lincandoc-0.3.pdf>

NOTE: TS-CAN1 AND TS-7KV

This guide was written to cover both TS-CAN1 and TS-7KV hardware, as the Lincan driver is compatible with both. You can use the same available files and described steps for both TS-CAN1 and TS-7KV, with the exception that TS-7KV is only compatible with ARM.

INSTALLING THE OCERA/LINCAN FILES

The files are available at the ftp server running on ftp.embeddedarm.com. The CAN directory is /can:

- **oceralincan_arm.tar.gz** : tarball with ocera/lincan files for TS ARM boards
- **lincan_arm.tar.gz** : tarball with last version of the lincan driver for TS ARM boards
- **oceralincan_x86.tar.gz** : tarball with ocera/lincan files for TS X86 boards
- **lincan_x86.tar.gz** : tarball with last version of the lincan driver for TS X86 boards

If you need the source code of OCERA/LINCAN, please follow the links below to get access to OCERA at CVS:

- <http://cvs.sourceforge.net/viewcvs.py/ocera/ocera/components/comm/can/lincan/>
- http://switch.dl.sourceforge.net/sourceforge/ocera/ocera_install_howto.pdf

Extracting the two tarballs for your specific architecture (ARM or X86) at the / directory of your target system will create the /usr/lincan/ directory filled with all necessary files (All the examples in this guide assume that / is your working directory). Use the following commands replacing xxx with the appropriate architecture name:

```
# tar zxvf oceralincan_xxx.tar.gz -C /  
# tar zxvf lincan_xxx.tar.gz -C /
```

LOADING THE DRIVER

The lincan driver for TS-CAN1 implements a plug and play system. So, the driver is able to detect the hardware, I/O address and IRQ line being used. To take advantage of the implemented plug and play feature, load the driver into the kernel by executing the command below:

```
# cd /usr/lincan/modules/  
# insmod lincan.o hw=tscan OR  
# insmod lincan.o hw=ts7kv
```

TESTING THE CAN NETWORK

You can use the testing tools available at /usr/lincan/bin-utils directory. For example, to set up your CAN hardware to listen to messages from a working CAN network, use the following command:

```
# ./readcan /dev/can0
```

To send messages out to a CAN network, you can use the sendburst tool. For example, to send out #3 can messages with identifiers #1 on each #1 second of period, do:

```
# ./sendburst --help  
# ./sendburst -d/dev/can0 -w1 -b3 -i1
```

To find further information about either the binary tools or LIBVCA and CAN/CANOpen, please view the OCERA documentation:

- <http://www.ocera.org/archive/deliverables/ms4-month24/WP7/D7.4.pdf>
- <http://cmp.felk.cvut.cz/~pisa/can/doc/ocera-canopen-ug.pdf>

MORE CONTROL OVER THE DRIVER

If you need more control over the driver configuration, you can skip the plug and play feature and use the module parameters. Get more information about the lincan driver using the modinfo tool. For TS-CAN1, the relevant driver parameters are:

- **hw:** Name of the TS hardware being used. It is tscan1 if using TS-CAN1 card or ts7kv if using TS-7KV card.
- **io:** The I/O address used by the card. It must be in accordance with the jumper selection. Use io=0x150 if jumper selection is all OFF for ARM I/O, for example.
- **irq:** The IRQ address used by the card. It must be in accordance with the jumper selection. For ARM platform, use irq=33 when jumper selection is IRQ 6, and irq=40 when jumper selection is IRQ 7.
- **canio:** Use this parameter for setting a specific I/O address space to allocate the SJA1000 registers. It is only available with TS-CAN1. See the table SJA1000 I/O address selection register in previous section.
- **clock:** Clock frequency of the on-board crystal/oscillator in step of 1kHz. The default is clock=16000 for 16Mhz on TS-CAN1 and clock=24000 for 24Mhz on TS-7KV.
- **baudrate:** Bit rate for the SJA1000 communication with the CAN network in step of 1kHz. The default is baudrate=1000 for 1Mbps.

USING ADDITIONAL TS-CAN1 CARDS IN A SINGLE SYSTEM

By using more than one TS-CAN1 card on a single system, you can set up a CAN network for testing and development with a single target embedded hardware. The Lincan device driver supports up to 8 CAN boards. You can load a single driver to give support to all cards by setting the appropriate parameters on driver loading. The parameters of the driver can be extended by using index commas. For example, to set up the Lincan driver to support 3 CAN boards and use plug and play functionality:

```
# insmod lincan.o hw=tscan1,tscan1,ts7kv
```

If you want to set specific values to the other parameters, it is possible to skip the plug and play functionality by assigning values others than -1. Notice that the -1 value tells the driver to use plug and play or default values to the specific parameter.

The next example loads the LINCAN driver to support two TS-CAN1 boards and one TS-7KV. The I/O space of the first board is 0x150, while the others are plug and play configured. The same methodology can be applied for IRQ and baudrate configurations.

```
# insmod lincan.o hw=tscan1,tscan1,ts7kv io=0x150,-1,-1 irq=-1,-1,33 baudrate=1000,500,-1
```

TS ARM based systems can be set up with a maximum of 4 TS-CAN1 boards and 4 TS-7KV boards. Thus, you are able to have up to 8 CAN interfaces running simultaneously.

SUPPORT

Email: support@embeddedarm.com

DOCUMENT HISTORY

=====

09.20.2005 – Created