QNX Device Driver

The Intellicon-NT960 QNX device driver provides an interface between the QNX operating system (V. 2.XX and V. 4.XX) and an Intellicon-NT960 Host Adapter. The NT960 QNX driver software is a device administrator that executes after the system powers up, (usually from a **sys.init.#** file[QNX 2.XX] or a **sysinit.#** file [QNX 4.XX]) and runs in background continuously.

QNX Version 2.XX

When the **nt960** administrator begins execution, it initializes the hardware then boots the i960 processor on the Host Adapter. This operation is not lengthy but usually takes a second.

Technical Tip:
For QNX Version 2.XX it is advisable that you place a **sleep 3** *command in your* **sys.init.#** *file (where* **#** *is the node numberof your system) following the* **nt960** *command. This will allow the administrator time to initialize. If you do not place a* **sleep** *command after the* **nt960** *command, commands executed from the* **sys.init.#** *file with the adopted serial ports will probably generate* **Invalid Device** *messages because the administrator has not completed its initialization.*

Note:
Due to certain restrictions with QNX Version 2.XX, you can install a maximum of 32 ports using one NT960 Host Adapter.

Installation

To install the QNX device driver follow these procedures:

Note:
The diskettes may also include README files. You will find these files in **/user/960/** *on the diskette. Please examine these files for technical tips or release notes concerning installation and configuration of the device driver*

Technical Tip:
*When you install the NT960 subsystem for the **first** time, you **must** run from QNX the* **ntload** *program with the HEX file* **f2nis.hex**.
*You run this **before** installation of the NT960 QNX 2.XX device driver.*

Example: **ntload f=f2nis**

You will find **ntload in /user/960/cmds/** *on the QNX 2.XX driver diskette. Please refer to the **ntload Program** section on page 3-1 of the **Software Installation** chapter.*

1. Insert the Intellicon-NT960 QNX device driver diskette into a floppy drive and copy all the files on to your system's boot drive.

Example:
For example, if you mount your hard drive as disk 3 and your floppy drive is disk 1, insert the NT960 QNX driver diskette into the floppy drive and type:

```
backup 1:/ 3:/ +a
```

2. Add the following entry to your **sys.init.#** file:

```
nt960 [options] &
```

Where **[options]** are:

a=memory_segment
Shared memory segment.
Default = D0000

+d
Date support. Default = off.

i=interrupt_line
Interrupt Request Line. Default = 10

s=signal_port
Signal port to use. Internally, the nt960 driver requires a signal port to communicate between its interrupt handler and the task time portions of the driver. Default = i + 1

Technical Tip:
We recommend you set the signal port at 19, 20, 21 or 22. You can type **tsk ports** *at the command prompt to see what signal ports are in use*

n=device_name
Base name of devices to create/adopt. By default, **nt960** will create **$nt1, $nt2** . . . etc. This option allows you to override the default and create your own arbitrary name.

t=start_tty

Specifies the starting device number to adopt. This allows you to specify what the **$tty** number will be for the first port on the NT960.

> ***Note:***
> *We do not recommend the use of the* **t=** *parameter. We only included it to gain compatibility with applications having a hard coded* **$tty** *number.*

p=port_address

Specifies the port address of the NT960 Host Adapter. Default = 300

S=signal_port

If you specify this parameter, the nt960 will signal this signal_port each time any port receives data.

> ***Note:***
> *We do not recommend the use of the* **S=** *parameter. We only included it to support older applications.*

+v

Verbose mode. Default = off

+w

Debug wait. You can use this for debugging custom firmware modules. It allows the driver to start but will not timeout waiting for the i960 during startup. Default = off

O=initial_options

Initial options. Allows setting of **tty** options at driver startup. You specify the initial_options as a hex word corresponding to the bits in **get_option()/set_option().**

> ***Note:***
> *See QNX documentation on* **get_option(), set_option()** *and* **stty** *for additional details.*

B=baud_rate Allows setting of initial baud rate during driver startup. By default, the driver starts with all ports set to 1200 baud. This parameter allows you to specify what baud rate to use when the driver starts.

N=stop_bits Allows setting of initial stop bits during driver startup. Possible settings are:
N=1 (default)
N=2

D=data_bits Allows setting of initial data bits during driver startup. Default = 8

P=parity Allows setting of initial parity settings during driver startup. The initial parity setting is none. Possible settings are:
P=even
P=odd

Example:

```
nt960 a=E0000 i=7 s=19 &
```

The above is an example of a **sys.init.#** *statement for one NT960 host adapter in your system.*

Where:

memory segment is:	E000(hex)
interrupt request line is:	7

The non-specified parameters will use the following defaults:

+d(date support):	off
signal_port:	s=19
device_name:	n=$nt
t=start_tty:	calculated by QNX
port_address:	p=300
S=signal_port:	off
+v(verbose mode):	off
+w(debug wait):	off
O=initial_options:	off
baud_rate:	B=1200
stop_bits:	N=1
data_bits:	D=8
parity:	P=none

Example:

```
nt960 a=E000 B=38400 D=8 N=1 P=even s=19 &
```

The above is an example of a **sys.init.#** *statement for one NT960 host adapter in your system.*

Where:

memory segment is:	*E000(hex)*
baud_rate:	*B=38400*
stop_bits:	*N=1*
data_bits:	*D=8*
parity:	*P=none*

The non-specified parameters will use the following defaults:

+d(date support):	*off*
interrupt_line:	*I=10*
signal_port:	*s=19*
device_name:	*n=$nt*
t=start_tty:	*calculated by QNX*
port_address:	*p=300*
S=signal_port:	*off*
+v(verbose mode):	*off*
+w(debug wait):	*off*
O=initial_options:	*off*

Utilities

nt_tty

The **nt_tty** utility sets or queries the **edit** and **raw** FIFO timeout levels on the ACM/16 or ACM/Flex16. When a port's options include QNX **EDIT**, the port is in **edit** mode, otherwise, the port is in **raw** mode.

Technical tip:
You should only use the **nt_tty** *utility if timing is critical (i.e., a polling application where a program waits for an event to occur in a fixed time period. Otherwise the default setting should be adequate.*

Each port or channel on the ACM/16 or ACM/Flex16 has a 12 byte RX Data FIFO to buffer incoming data. You can program each FIFO to generate interrupts to the i960 when the inter-character time reaches a specified timeout level.

The nt960 firmware sets the fill level according to the baud rate as follows:

baud rate > 76800	Fill level = 4
baud rate > 19200	Fill level = 6
baud rate > 4800	Fill level = 8
other baud rates	Fill level = 10

The nt960 firmware sets the timeout levels to 20 ms and 20 ms respectively, for **raw** and **edit** timeouts. You set timeout levels in 5 millisecond increments, with the smallest level being 5 milliseconds.

If used with no parameters, **nt_tty** reports the current settings of the port from its **stdin**. When used with parameters, **nt_tty** effects the changes on its **stdin** (i.e., **nt_tty raw=5 <$nt1**).

> ***Note:***
> *You will find* **nt_tty** *in* **/cmds/** *on the QNX diskette.*

Syntax

The syntax for the **nt_tty** statement is:

nt_tty [options] < [NT960 port]

Where **[options]** are:

r=raw_time	Set the **raw** timeout level
raw=raw_time	Set the **raw** timeout level
e=edit_time	Set the **edit** timeout level
edit=edit_time	Set the **edit** timeout level

and **[NT960 port]** is the Intellicon-NT960 subsystem serial port (i.e., $nt2)

Examples:
The first example shows the **nt_tty** *utility used with no parameters: The* **nt_tty** *utility reports the current settings of* **device_name $nt1** *from its* **stdin.**

```
nt_tty < $nt1
```

The second example shows the **nt_tty** *utility used with parameters that set the* **raw** *and* **edit** *FIFO timeout levels to 5 milliseconds on* **device_name $nt1**.

```
nt_tty raw=5 edit=5 < $nt1
```

The third example shows the **nt_tty** *utility used with another set of parameters that set the* **raw** *and* **edit** *FIFO timeout levels to 5 milliseconds on* **device_name $nt12**.

```
nt_tty r=5 e=5 <$nt12
```

Technical Tip: basic uses of the* stty *command
You can use the **stty** *command to set and query the parameters of a NT960 port.*

Examples:

Query a port:

```
stty < $nt1
```

Set baud rate (available baud rates = 50 bps to 57.6 Kbps):

```
stty baud=9600 > $nt1
```

Set line parameters:

```
stty bits=7 stop=2 par=even > $nt1
```

Please refer to QNX 2.XX documentation on **stty** *for additional information*

Error Messages

Fatal Errors

talk_to_960: Failed (%d:%d): Timeout waiting for last complete (cs_busy=%x, last=(%d:%d))

talk_to_960: Failed (%d:%d): Timeout waiting for accept

talk_to_960: Failed (%d:%d): Timeout waiting for complete (cs_busy=%x)

talk_to_960: Failed(%d:%d): Error code %d

talk_to_960: Null reason ??

talk_to_960: Failed(%d:%d): Error code %d

high_speed: Wait failed Last (%d:%d) Current (%d:%d)

high_speed: Null reason ??

low_speed: Wait cs_busy failed Last (%d:%d) Current (%d:%d)

low_speed: (%d) Wait complete 1 failed Last (%d:%d) Current (%d:%d)
These fatal error messages indicate an internal **nt960** driver or firmware failure: Please contact Connect Tech Customer Support.

Startup/Initialization Errors

Memory verify error on DPR.

NT960 did NOT start

NT960 did NOT complete initialization
The **nt960** driver cannot verify reads and writes to the DPR on the NT960 Host Adapter. This can result from a hardware failure or a hardware conflict. Please check for I/O port address and base memory address conflicts between the NT960 adapter and other adapters in the system.

This version of QNX (counter=%04x) is not supported by this driver. Please contact CTI Tech Support for further information. Voice: 519-836-1291 FAX: 519-836-4878
The **nt960** driver does not support extremely old versions of QNX.. Please contact Connect Tech Customer Support.

QNX port %d in use by tid %4.4x
The Signal port specified using **s=**, or the port calculated from the IRQ number, is already in use. To correct this problem, use the **s=** parameter on the command line to specify a different signal port.

Waiting for interrupt
You are using the **+w** parameter on the command line. The **nt960** driver will wait indefinitely for an interrupt. You should only use this with hardware test and debugging equipment.

No interrupt from card %d: port=%3.3x int=%d
No interrupt sent by the NT960 Host Adapter. Check your interrupt jumpers and command line settings and try again. Also please check for possible IRQ conflicts with other adapters in the system.

Operational Errors

NT960: unknown last msg type (%d).
The **nt960** driver did not understand the last message type received. This results from a task sending a "bad" message to **nt960**. When this occurs, **nt960** will not reply to the task sending the message. This will leave the offending task in a "REPLY" blocked state; by using **tsk** you can find the **tid**, name, etc. of the offending task.

Exception received on channel %d (invalid)
The **nt960** driver detected a channel/port number that is invalid. Internal **nt960** driver or firmware failure. Contact Connect Tech Customer Support.

empty: not open for write
A task attempted a write on a channel/port that it does not have open for write.

function out of range %d
The **nt960** driver detected an internal function number that is invalid. Internal **nt960** driver or firmware failure. Contact Connect Tech Customer Support.

hw_timeout: type %d
The **nt960** driver detected an internal function number that is invalid. Internal **nt960** driver or firmware failure. Contact Connect Tech Customer Support.

device_control: unknown function %d.
The **nt960** driver detected an internal function number that is invalid. Internal **nt960** driver or firmware failure. Contact Connect Tech Customer Support.

Break_begin: function not implemented.
Break_end: function not implemented.
The **nt960** driver detected a function not supported in this release. Internal **nt960** driver or firmware failure. Contact Connect Tech Customer Support.

Device out of range (%d)
The **nt960** driver detected a channel/port number that is invalid. Internal **nt960** driver or firmware failure. Contact Connect Tech Customer Support.

Illegal interrupt reason (%d).
The **nt960** driver detected an interrupt code that is invalid. Internal **nt960** driver or firmware failure. The **nt960** driver detected a channel/port number that is invalid. Contact Connect Tech Customer Support.