



IO60-COM4 Device Driver Package

3.0.35 Kernel-Based Linux

1 Introduction

1.1 The IO60-COM4 Device Driver Package consists of two Linux drivers: the actual device driver (io60_com4.ko) and a board configuration driver (io60_com4_cfg.ko). A Makefile is provided to install and uninstall the driver package. It also includes two bash scripts, one that installs both drivers and configures the device and the other to remove both drivers.

1.2 This driver is for use only with the WinSystems' SBC35-C398 series of ARM based single board computers. The driver supports the WinSystems' IO60-COM4 board, which provides four serial ports.

1.3 Both drivers have been built and are included as modules in this package.

1.4 If driver changes are necessary, the driver source code can be edited and the kernel can be rebuilt with the WinSystems' downloadable Linux Target Image Builder (LTIB) kernel package.

1.5 This driver is provided 'as-is' and no warranty as to usability or fitness of purpose is claimed.

1.6 WinSystems does not provide support for the modification of this driver. Bug reports may be sent to linux_drivers@winsystems.com.

1.7 This driver is provided under the terms of the GNU General Public License.

2 Installation and Build

2.1 Since the drivers are already present, no Makefile option is required to build the two IO60-COM4 modules. The included Makefile will install or uninstall the two drivers based on the command line argument.

make install – remove existing drivers, install current drivers, and generate dependencies

make uninstall – remove drivers and dependencies

2.2 Two script files are provided that will allow the modules to be loaded and unloaded as needed. They are called *io60_com4_load* and *io60_com4_unload*.

2.3 The MAJOR number for this device is hard coded at 210. If a different number is desired, the driver will have to be rebuilt (see section 1.4).

2.4 The initialization script *io60_com4_load* (shown below) will reset the device, load the modules, configure the port parameters, provide the required file permissions for the device nodes, and reset the device. The driver can also be loaded manually by the user with the *modprobe* command.

```
#!/bin/sh
module1="io60_com4"
module2="io60_com4_cfg"
mode="666"

# gpio defines
exp_rst="/sys/class/gpio/gpio236/value"
ttyio60com0_slew="/sys/class/gpio/gpio256/value"
ttyio60com0_term="/sys/class/gpio/gpio257/value"
ttyio60com0_m1="/sys/class/gpio/gpio258/value"
ttyio60com0_m0="/sys/class/gpio/gpio259/value"
ttyio60com1_slew="/sys/class/gpio/gpio260/value"
ttyio60com1_term="/sys/class/gpio/gpio261/value"
ttyio60com1_m1="/sys/class/gpio/gpio262/value"
ttyio60com1_m0="/sys/class/gpio/gpio263/value"
ttyio60com2_slew="/sys/class/gpio/gpio264/value"
ttyio60com2_term="/sys/class/gpio/gpio265/value"
ttyio60com2_m1="/sys/class/gpio/gpio266/value"
ttyio60com2_m0="/sys/class/gpio/gpio267/value"
ttyio60com3_slew="/sys/class/gpio/gpio268/value"
ttyio60com3_term="/sys/class/gpio/gpio269/value"
ttyio60com3_m1="/sys/class/gpio/gpio270/value"
ttyio60com3_m0="/sys/class/gpio/gpio271/value"
ttyio60com0_en="/sys/class/gpio/gpio272/value"
ttyio60com1_en="/sys/class/gpio/gpio273/value"
ttyio60com2_en="/sys/class/gpio/gpio274/value"
ttyio60com3_en="/sys/class/gpio/gpio275/value"

# reset expansion bus
echo 0 > $exp_rst
sleep 1
echo 1 > $exp_rst

# invoke modprobe
/sbin/modprobe $module1 protocol="rs232","rs422","rs485","rs232"
/sbin/modprobe $module2

# alter permissions
chmod $mode /dev/ttyio60com[0-3]

# change baud rate (default = 9600)
stty -F /dev/ttyio60com0 19200
stty -F /dev/ttyio60com1 57600
stty -F /dev/ttyio60com2 115200

# set SW flow control on port 1
stty -F /dev/ttyio60com0 -crtcts ixon ixoff

# configure port 0 (default mode bits are rs232)
```

```
echo 1 > $ttyio60com0_en

# configure port 1 (rs422)
echo 1 > $ttyio60com1_en
echo 0 > ttyio60com1_slew
echo 1 > ttyio60com1_term
echo 1 > ttyio60com1_m1
echo 1 > ttyio60com1_m0

# configure port 2 (rs485)
echo 1 > $ttyio60com2_en
echo 0 > ttyio60com2_slew
echo 1 > ttyio60com2_term
echo 1 > ttyio60com2_m1
echo 0 > ttyio60com2_m0

# disable unused ports
echo 0 > $ttyio60com3_en
```

2.5 To load the driver module automatically at boot, add the *io60_com4_load* command to the */etc/rc.local* file using a text editor.

2.6 The initialization script *io60_com4_unload* (shown below) will unload both of the driver modules.

```
#!/bin/sh
module1="io60_com4"
module2="io60_com4_cfg"

# remove modules
/sbin/modprobe -r $module1
/sbin/modprobe -r $module2
```

3 Driver Architecture

3.1 The *io60_com4* driver is accessed as an SPI-bus device using the IO60 Expansion Bus. The driver is implemented using the Linux Serial Driver layer. This requires the driver to register itself with the serial core and then register the individual serial ports in the system. The *uart_ops* pointer is used to define a list of functions that the serial core uses to call back into the port-specific serial driver.

3.1.1 The module parameter *protocol* is used to select the serial port interface for all four ports. Valid selections are rs232, rs485, and rs422. The default values if this parameter is not included is rs232. An example of usage is here:

```
modprobe io60_com4 protocol="rs232","rs485","rs422","rs232"
```

3.2 The *io60_com4_cfg* driver is accessed as an I2C-bus device using the IO60 Expansion Bus. The driver uses the *i2c_new_probed_device* function to instantiate the I/O devices required to configure the device. The specific configuration bits can be accessed using the echo command to

write the appropriate bit value to each bit. For example to reset the device you would do the following:

```
echo 0 > /sys/class/gpio/gpio236/value
sleep 1
echo 1 > /sys/class/gpio/gpio236/value
```

3.2.1 The complete list of configurable parameters is as follows. Short user-friendly names have been created. Multiple usage examples are shown in section 2.4.

```
Board Reset:      exp_rst = /sys/class/gpio/gpio236/value

COM0 Enable:      ttyio60com0_en  = /sys/class/gpio/gpio272/value
COM0 Slew:        ttyio60com0_slew = /sys/class/gpio/gpio256/value
COM0 Termination: ttyio60com0_term = /sys/class/gpio/gpio257/value
COM0 Mode 0:      ttyio60com0_m0  = /sys/class/gpio/gpio258/value
COM0 Mode 1:      ttyio60com0_m1  = /sys/class/gpio/gpio259/value

COM1 Enable:      ttyio60com1_en  = /sys/class/gpio/gpio273/value
COM1 Slew:        ttyio60com1_slew = /sys/class/gpio/gpio260/value
COM1 Termination: ttyio60com1_term = /sys/class/gpio/gpio261/value
COM1 Mode 0:      ttyio60com1_m0  = /sys/class/gpio/gpio262/value
COM1 Mode 1:      ttyio60com1_m1  = /sys/class/gpio/gpio263/value

COM2 Enable:      ttyio60com2_en  = /sys/class/gpio/gpio274/value
COM2 Slew:        ttyio60com2_slew = /sys/class/gpio/gpio264/value
COM2 Termination: ttyio60com2_term = /sys/class/gpio/gpio265/value
COM2 Mode 0:      ttyio60com2_m0  = /sys/class/gpio/gpio266/value
COM2 Mode 1:      ttyio60com2_m1  = /sys/class/gpio/gpio267/value

COM3 Enable:      ttyio60com3_en  = /sys/class/gpio/gpio275/value
COM3 Slew:        ttyio60com3_slew = /sys/class/gpio/gpio268/value
COM3 Termination: ttyio60com3_term = /sys/class/gpio/gpio269/value
COM3 Mode 0:      ttyio60com3_m0  = /sys/class/gpio/gpio270/value
COM3 Mode 1:      ttyio60com3_m1  = /sys/class/gpio/gpio271/value
```

4 Serial Port Configuration

4.1 Before any of the serial ports can be used, each one needs to be configured for the proper speed and flow control. The Linux utility *stty* can be used to change and display these parameters. Any of the four serial ports can be specified.

- Serial Port 1 = /dev/ttyio60com0
- Serial Port 2 = /dev/ttyio60com1
- Serial Port 3 = /dev/ttyio60com2
- Serial Port 4 = /dev/ttyio60com3

4.1.1 Current Settings

The following command line is used to display the current state of a serial port.

```
stty -F /dev/ttyio60com3 -a
```

4.1.2 Baud Rate

The following command line is used to set the baud rate of a serial port. The default baud rate of all ports is 9600 bps. The maximum baud rate supported for RS-232 mode is 115,200 bps.

```
stty -F /dev/ttyio60com1 19200
```

4.1.3 Disable Flow Control

The following command line is used to disable all flow control for a serial port.

```
stty -F /dev/ttyio60com2 -crtcts -ixon -ixoff
```

4.1.4 Hardware Flow Control

The following command line is used to select hardware flow control for a serial port. For this setting the signals RTS and CTS are used to throttle the flow of data.

```
stty -F /dev/ttyio60com0 crtcts -ixon -ixoff
```

4.1.5 Software Flow Control

The following command line is used to select software flow control for a serial port. For this setting the signals RTS and CTS are used to throttle the flow of data.

```
stty -F /dev/ttyio60com3 -crtcts ixon ixoff
```