

VIOM GPI Interface

GPIVIOM.exe

Written by Andy Woodhouse

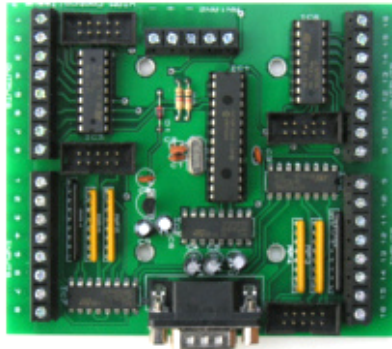
Contents

1	Overview.....	3
1.1	Description.....	3
1.2	CSI Version.....	3
1.3	Resilience and Redundancy.....	4
2	Driver Configuration.....	4
2.1	Ini file settings	4
2.1.1	[System] section	4
2.1.2	[Inverts] section	5
2.1.3	[Database] section	5
3	Tally and Locking States	5
4	Driver GUI	6
5	Mode Configuration File for VIOM.....	7
6	Version History	9
6.1	Driver Version	9
6.2	Document Version	9

1 Overview

1.1 Description

VIOM is the abbreviated name for Versatile Input Output Module, a board manufactured by Phaedrus in the UK and sold directly by them and via Farnell/OneCall. A low-resolution image of the board is shown below.



The board is managed by a PIC controller that provides timers, monitors the input signals, switches output signals, and can provide re-entrant connections for the outputs to reset latches set by input changes.

There are 16 input lines pulled to +5V by internal resistors, and 16 output buffers. The unit communicates serially to a host that can change the operating modes of the device. For the BNCS application, a configuration is downloaded that sets a hardware de-bounce of 40 mS for input lines, and sets all outputs to be controlled by the serial protocol. The board costs around £50 from CPC (July 2016), and add-on modules to provide switched relay outputs are available from the same suppliers. Phaedrus also sells an opto-isolator input board. Extension boards cost about £15 each for 8 controlled contacts.

The low cost of this unit provides a BNCS GPI interfaces for simple applications. The serial protocol can be a speed limitation when many fast changes of output are required.

The GPIVIOM.exe is 32-bit BNCS driver software for the module. It has been tested on Windows XP and Windows 7 (32 and 64 bit) hosts. The driver implements standard GPI BNCS messages, and can be run in simulation mode. It runs in stand-alone mode and does not support redundancy (RX Only) modes. Input contacts are BNCS switches 1 to 16, 17 to 32 are BNCS outputs.

Serial communication uses 9600 baud, 8-bits per character, no parity, one stop bit.

1.2 CSI Version

The driver uses CSI 32-bit messaging, and has been tested with CSI32 and CSI V3.

1.3 Resilience and Redundancy

This driver does NOT support Tx/Rx switching, as there is only a single serial connection to the hardware.

2 Driver Configuration

The driver requires a device number on the task startup command line. It looks for a configuration file dev_xxx.ini where xxx is the driver ID passed on the command line. If the ini file is not found, a file with default settings is created.

The search for the ini file has up to three stages:

1. Look for environment variables CC_ROOT and CC_SYSTEM. These will have values in a version 4 or 4.5 system. If the environment variables are not found stage 2 processing occurs.
2. Look for the relevant entries in c:\bncs_config.ini. This file is used in version 3 systems. If bncs_config.ini is not found, stage 3 processing occurs.
3. The driver assumes the ini file is stored in the windows directory. Standard windows API functions are called to find the path to the directory.

2.1 Ini file settings

There are two data sections used by the driver – **System** and **Inverts**.

2.1.1 [System] section

This section holds values relating to basic operations.

Item	Values and Defaults	Comment
Debug	0 or 1 Default is 0	1 to send debug data to the windows debug display and the list box in the dialog.
Port	1 to nn Default is 1	Windows com port number used to communicate with VIOM module.
Speed	9600 (Default)	Required by VIOM
DataBits	8 (Default)	Required by VIOM
StopBits	1 (Default)	Required by VIOM
Parity	N (Default)	Required by VIOM
TallyDelay	2000 (Default)	This parameter is the delay, in milliseconds, between successive communications check polls of the VIOM.
SaveFileInterval	15	Delay in minutes between saving lock state and I/O states to disk.

Simulation	0 or 1 Default is 0	Setting this flag to 1 sets the driver to simulate mode. In simulate it does not communicate with VIOM hardware
------------	------------------------	---

Although the baud rate needs to be at the values shown above, the driver reads them from the section in case a future updated VIOM is offered with faster serial communications.

2.1.2 [Inverts] section

This section holds the input and output invert flags. These allow a user to adjust the physical level to match a logical level. For example, an open input reports as logic 0, a closing contact reports as logic 0. If we have a power supply that has a closed contact when operating correctly, it may be more logical to invert the input so that the error state reports logic 1.

There are 4 lines in the section in a format similar to:

O/P_01-08=0,0,0,0,0,0,0,0

O/P_09-16=0,0,0,0,0,0,0,0

I/P_01-08=0,0,0,0,0,0,0,0

I/P_09-16=0,0,0,0,0,0,0,0

Replace the 0 in the above comma delimited lists by a 1 to invert the associated input or output. The leftmost entry in the string is the lowest contact number.

2.1.3 [Database] section

This is not used or created by this driver. It appears after CSI has been run on a instance of the driver.

3 Tally and Locking States

The driver writes the tally and lock states for all inputs and outputs to files. These files are stored in the "data" folder for the host system type. For a version 4.5 system the folder is: %CC_ROOT%\%CC_SYSTEM%\data

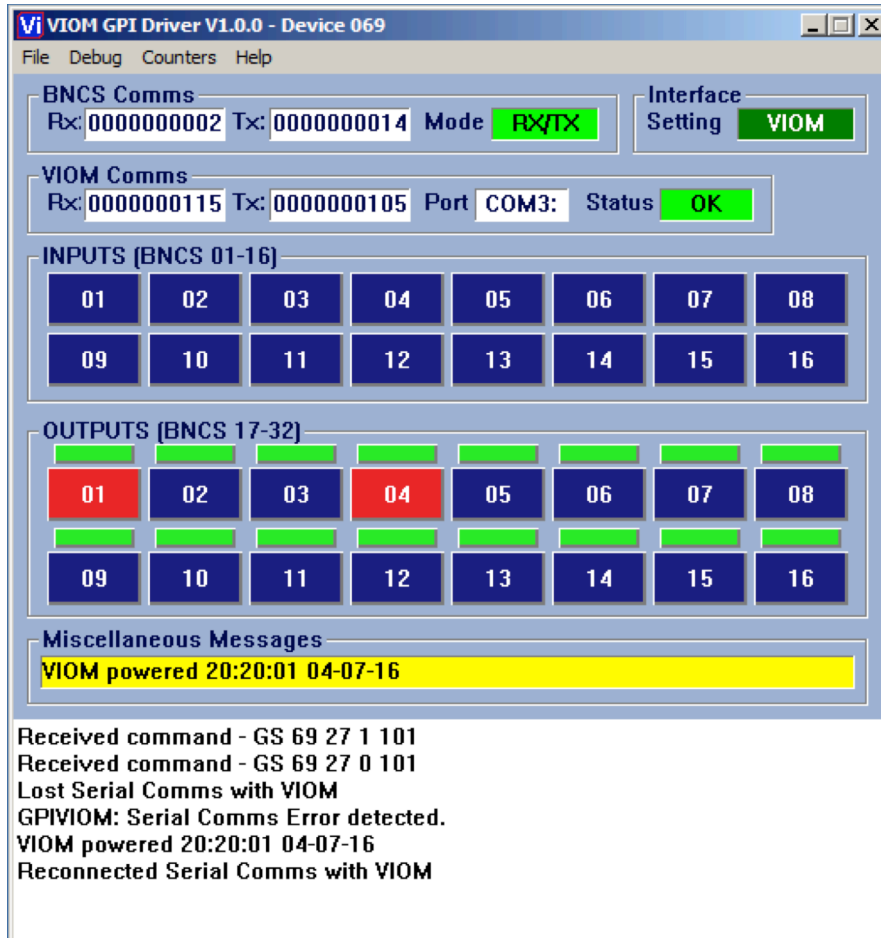
Although the VIOM driver only uses 32 contacts, the files have the ability to hold data for 4096 contacts, following the convention for other BNCS drivers.

The tally and lock states are automatically saved at intervals set by the SaveFileInterval token of the [System] section. The states are also saved when the driver closes down.

When a driver starts, it checks for the existence of these files. If they are not found, default files are created. If the files exist, they are read and the VIOM interface is set to match the values. In simulate mode this includes the state of the inputs as well as the outputs.

4 Driver GUI

The picture below shows the driver GUI with “Debug” and “Debug Show” enabled.



The interface setting at the top right indicates the operating mode – Simulated or VIOM.

In VIOM mode the Inputs buttons operate as mimics only. They are blue for switch logic 0, red for logic 1. The states are post invert controls. In Simulate mode the buttons act as switches to enable the user so simulate external contacts changing.

The output buttons are active in simulate and VIOM modes. Colour indications are the same as input mimics. The green rectangles above the output boxes are the lock indicators and switches. These change to red backgrounds when the lock is set active.

The miscellaneous Messages box starts empty with a white background. If the VIOM is powered whilst connected the VIOM module the box goes yellow and it shows the time and date when the unit reported power up.

5 Mode Configuration File for VIOM

VIOM has many complex programming options. For this driver we need a simple mode. This is implemented by downloading a series of commands over the serial interface. These settings are stored in non-volatile memory. The settings can be held in a text file, and be downloaded to an instance of VIOM using Hyperterm or similar communications tool. The settings are listed below. The lines that start with a "<" character are comments that are ignored by the VIOM.

```
< Set viom module to mode required for BNCS GPIVVIOM
FDL
CMS
CSM1
CEAE
CIR0
CR0
< Set outputs to respond to RS232
WO1N3
WO2N3
WO3N3
WO4N3
WO5N3
WO6N3
WO7N3
WO8N3
WO9N3
WO10N3
WO11N3
WO12N3
WO13N3
WO14N3
WO15N3
WO16N3
WO17N3
WO18N3
WO19N3
WO20N3
WO21N3
WO22N3
WO23N3
WO24N3
WO25N3
WO26N3
```

BBC

```
WO27N3
WO28N3
WO29N3
WO30N3
WO31N3
WO32N3
< Change input reporting prefixes to "i" "j" "k"
CIP1i
CIP2j
CIP3k
< Set input debounce on
WI1C1
WI2C1
WI3C1
WI4C1
WI5C1
WI6C1
WI7C1
WI8C1
WI9C1
WI10C1
WI11C1
WI12C1
WI13C1
WI14C1
WI15C1
WI16C1
< Set debounce units to 10ms
WI1B0
WI2B0
WI3B0
WI4B0
WI5B0
WI6B0
WI7B0
WI8B0
WI9B0
WI10B0
WI11B0
WI12B0
WI13B0
WI14B0
WI15B0
WI16B0
< Set 4 units of time
WI1A4
```


BBC

WI2A4
WI3A4
WI4A4
WI5A4
WI6A4
WI7A4
WI8A4
WI9A4
WI10A4
WI11A4
WI12A4
WI13A4
WI14A4
WI15A4
WI16A4
< Enable Input Change Reporting
CIC1
< End of VIOM GPI config file

6 Version History

6.1 Driver Version

Version No	Date	Details	Name
V1.0	07/07/2016	Initial release	Andy Woodhouse

6.2 Document Version

Version No	Date	Details	Name
V1.0	07/07/2016	First release	Andy Woodhouse