VIOM TUTORIAL

Introduction

VIOM stands for Versatile Input Output Module. After working through this tutorial I hope that you will agree that the description is accurate.

VIOM has 16 digital inputs, 16 digital open collector outputs, 8 Timer / Counters and a Serial Port. Commands are issued via the Serial Port and the processor will control the outputs according to the commands issued.

VIOM can function either as a stand alone module or it can work in conjunction with a host controller (for example a PC) or a combination of both.

Inputs and outputs can be connected to VIOM either via the screw terminal block or via the IDC headers. Additional output LED and Relay modules are available which connect directly to VIOM on the IDC headers.

Setting up

For this tutorial we will need the following equipment:

1 VIOM
1 VIOM LED Display Module
1 12 Volt Power Supply
1 Null modem lead connected to a PC
3 switches.

Connect one switch between input 1 and supply -ve.
Connect one switch between input 2 and supply -ve.
Connect one switch between input 3 and supply -ve.

Connect the display module to the VIOM output connector 1 (located adjacent to the power terminals).

Connect the power supply to the VIOM power terminals (observing the correct polarity.

If you are using HyperTerminal set it up as follows:

Go to File/Properties.
Set the “Connect using” dialog to “Direct to ComX” where X is the Serial Port you are using.

Go to the “Configure” button and set the following:

“Bits per second”:	9600
“Data Bits”:	8
“Parity”:	None
“Stop Bits”:	1
“Flow Control”:	None

Press OK to return to the File/Properties dialog.

Go to the “Settings” Tab and set the following:

Function arrow and ctrl keys act as :	Terminal keys
Backspace key sends:	Cntrl+H
Emulation:	ANSI
TELNET Terminal:	ANSI

Press ASCII Set-up button.
Check the “Send line ends with line feeds” box.
Check the “Wrap lines that exceed terminal width” box.
Set “Line delay” to 0.
Set “Character delay” to 0.
Other boxes should not be checked.

Press OK to return to the File/Properties dialog.

Press OK to return to the main screen.

Save the new settings.

Hyper Terminal is now set up.

Getting started

Having made all the connections apply power to the VIOM module. In HyperTerminal you should see the message “VIOM Started”. If not check that power has been connected and that it is the right polarity. Check the modem cables (remember that you need to use a Null Modem lead). Check that HyperTerminal has been set to the correct baud rate and other settings.

Assuming that you now have the start up message you can now start to use the VIOM. operate the switch connected to input 1. LED 1 should now start to flash once per second. Open the switch and the LED should extinguish. If you have problems here measure the voltage on the input with reference to the - power terminal. It should read approximately 5 Volts when the switch is open and around 0 Volts when it is closed.

Repeat the operation for input 2 (LED 2 should flash) and input 3 (LED 3 should flash).

Revert to defaults by issuing the command “CPD”.

Changing the Flash Rate

From HyperTerminal type the command “W01C20”. The VIOM is not case sensitive so you could equally well type “wo1c20”. This command tells the VIOM to change the On Time for output 1 to 20 units. The default units are 100mS so this equates to 2 seconds. The VIOM will now respond with a message telling you the new On Time.

Now operate input 1 switch. The LED will now stay on for 2 seconds and go off for 1 second. “W01C30” will put the LED on for 3 seconds, “W01C40” will put the LED on for 4 seconds and so on.

Try changing the time for other periods. Remember that the value should be in the range 1 to 255 (we will come to zero soon).

If you are not sure of the current time you can use the command “R01C”. This command will return the value of the On Time without affecting the stored value. In general for each “W” (write) command there is an equivalent “R” command.

After you have finished playing return the value back to the default value of 10 by using the command “WO1C10”.

By changing the On Time you can turn the LED on for any period between 100mS and 25.5 seconds. Other periods are possible however. As we said earlier the default units are 100mS. We can change this to 10mS by using the command “WO1D0”.

The LED will now be on for a very short period (100mS), but once again the LED off time is not affected. Try changing to other periods. “WO1D2” will change to seconds and the LED will now stay on for 10 seconds. If you need a break try setting the units to minutes (“WO1D3”). For a longer break try “WO1D4” which will set the units to hours and if you are going on holiday try “WO1D5” which will set the units to days.

We can of course change the LED on period for other outputs as well. “WO2....” will address output 2, “WO2....” will address output 3 and so on.

So much for the LED on timing. What we have done for the on period can be done for the off period. As with the on timing the off timing is a combination of ON Time and Units. “WO1E....” will change the Off Time and “WO1F....” will change the Off Units.

It is worth noting here how the timings are derived and some precautions which are necessary to maintain accuracy. The actual value of the on time will have an accuracy of + or - 1 unit and it is therefore important to choose the value of the period and the units carefully. For example if an on period of 2 seconds is required the value of this parameter could be set to 2 and the on units could be set to seconds. This would give a period in the range 1 to 3 seconds. However if the period was set to 200 and the on units to 10mS then the accuracy would be improved to between 1.99 seconds and 2.01 seconds.

Revert to defaults by issuing the command “CPD”.

Keeping the Output On or Off

Not all applications need the output to go on and off. If we want the output to remain on permanently then we could set the On Time to 255 and the On Units to days. But alas 255 days is not permanent! We can however set the output permanently on by setting the On Time to zero. The On Units don’t matter and no notice is taken of the off periods.

Use the command “WO1C0”. Operate the switch on input 1 and LED 1 will remain on. You can now set the On Time back to the default value of 1 second.

Another scenario might be that the output should come on, stay on for a period, then go off and remain off. This can be achieved by setting the Off Time to zero. “WO1E0” will do this and assuming that the on period has been set up correctly the LED will come on once for a period of 1 second when input 1 switch is operated and then remain off.

Revert to defaults by issuing the command “CPD”.

Output Recognition Period

As well as controlling the on and off periods for outputs we can also insert a delay at the start of the sequence. This delay is referred to as the Recognition Period. As with the on an off periods the recognition period is controlled by two parameters: Recognition Time and Recognition Units.

When you operate input switch 1 you may notice a slight delay before the LED turns on. This is due, in part, to the output recognition period which is set to 1 second as default. Try changing this by altering the Recognition Time to 20 using the command”WO1A20”. Now when you operate the switch on input 1 there is a delay of approximately 2 seconds before the LED is turned on.

The Recognition Units can be changed in the same way as the On and Off Units. To reduce the time use the command “WO1B0” which will set the units to 10mS. You will notice that the LED now comes on almost at once.

If you do not need a recognition period for your application set the Recognition Time to 1 (not zero) and the units to 10mS. This will produce a delay in the range 0 to 10mS and is inherent in the operation of the VIOM, 10mS being the sampling time.

Revert to defaults by issuing the command “CPD”.

Output Active High and Low

Before we get on to the software aspects of active highs and lows it is important to understand how the outputs work electronically. Each output is driven by a Darlington transistor and is referenced to 0 Volts. The load is normally connected between the output terminal and positive voltage supply rail. When the transistor is turned on current is drawn from the load through the transistor to the 0 Volts line.

In order to turn our LED on we must first turn on the transistor, current will then flow from the positive rail, through the LED, through the transistor an so to 0 Volts. Now if we measure the voltage between the 0 Volts line and the output terminal we will measure close to 0 Volts when the LED is on and close to supply when the LED is off.

We can define a low condition as the output being close to 0 Volts and a high condition as when the output is close to positive supply. Further we can define an active condition as being when the LED is on. The output is therefore *active* when it is *low*, that is *active low*.

There may be cases where we need to define the output as being active when the output transistor is off. In this case if we measured the voltage at the output terminal it would be *high* when the output is *active*, that is *active high*.

Now that the hardware is sorted on with the software. Let us set the on period for output 1 LED to 5 seconds and its off period to 1 second:

```
WO1C50
WO1D1
WO1E100
WO1E0
```

Now operate input 1 switch. The LED will now flash on 5 seconds and off 1 second. Turn the switch off and send the new command “WO1H1”. This will set the output to active high. Immediately you should see the LED come on. Now operate the switch again. Once more LED 1 should flash, but this time it will flash on 1 seconds and off 5 seconds. In other words the output has been inverted.

The output can be returned to active low by using the command “WO1H0”.

Revert to defaults by issuing the command “CPD”.

Output Latching

Until now, when we have used the switch to turn the output on and off. We can also use one switch to turn an output on and another to turn it off. The first change we need to make is to make output 1 latching. We do this with the command “WO1K1”. Now we need to define an input to reset this output. Let us use input 2 to reset output 1. The command is “WO1G2”.

Now when we operate switch 1, LED 1 will flash and when we turn off the switch the LED will continue to flash. To turn off the LED we can operate switch 2 (LED 2 will flash but this is immaterial to this test). LED 1 will now extinguish.

We can also reset an output from the Serial Port. Again operate the switch on input 1 and LED will start to flash. release the switch. Now issue the command “WO1U”. the LED should now go out.

As with the recognition period on the outputs which allows for a delay before the output is full activated, so there is an equivalent for the Reset Input. As with other timings the reset period is in two parts: the Reset Time and the Reset Units. To change the Reset Time on output 1 use the command “WO1S...” and the Reset Units are “WO1T...”.

The reset period is applied to the output. This means that if you are using a single input to reset several outputs you can have different periods for each output which is being reset. Starting from defaults use the following code:

```
WO1K1
WO1G3
WO1S10
WO1T0
WO2K1
WO2G3
WO2S50
WO2T1
```

This will define outputs 1 and 2 as latching with input 3 acting as a reset. Input 1 will have a reset period of 100mS while input 2 will have a reset period of 5 seconds. Now momentarily operate switches 1 and 2 to set LEDs 1 and 2 flashing. When you reset the outputs by operating switch 3 notice the difference in timing.

The reset period is only effective when the output is reset from one of the inputs. it is not effective when the output is reset via the Serial Port. Again latch outputs 1 and 2 but this time reset the outputs via the Serial Port with the commands “WO1U” and “WO2U”. you will notice that both of the outputs reset immediately.

Revert to defaults by issuing the command “CPD”.

Output One-Shot Mode

Send the following commands to the VIOM:

```
WO1K1
WO1G2
WO1C50
WO1E0
```

These commands will set output 1 conditions to latching, reset input 2 and output on period 5 seconds once only.

Now operate the switch on input 1. The LED will come on for a period of 5 seconds, go off and remain off. even if the switch on input 1 is re-operated the LED will remain off until two events take place. First the output needs to be reset and then the output needs to be activated again. Try this to prove it to yourself.

But what if we need the output to be re-triggered without a reset being used? We can do this by putting the output into One-Shot mode. First disable latching with “WO1K0” then enable One-Shot with “WO1W1”. Now repeat the procedure with the switch. This time after the LED has gone out it can be turned on again without waiting for a reset.

Revert to defaults by issuing the command “CPD”.

Using Multiple Inputs

Up to this point we have always had input 1 controlling output 1, input 2 controlling output 2 and so on. This, it must be said, has its limitations. VIOM can of course provide much greater flexibility than that. In fact any input can be assigned to any output and more than one input can be assigned to an output.

VIOM maintains a list, or map, of which inputs are used to control each output. Commands are available to add or remove inputs from these lists. First let us add input 2 to the list of output 1 by using the command “WO1L2”. Inputs can be either ANDed or ORed together. As it is the default we will start with them being ORed.

Now when either the switch on input 1 or on input 2 is activated LED 1 will flash (LED 2 will flash but this is immaterial to this test). This is because the inputs are ORed. That is that output will be active when input 1 is active OR when input 2 is active. We can also AND inputs.

To change the way an output processes is list of inputs from ORing to ANDing use the command “WO1J0”. Now LED 1 will only flash when both the switch on input 1 AND the switch on input 2 are operated.

We can also remove an input from the output list. The command “WO1M2” will remove input 2 from the list of output 1. If, for example, we want to stop LED flashing when input 2 switch is operated we can issue the command “WO2M2”.

All the previous commands are still viable when we use multiple inputs. We can still latch outputs, change on and off periods etc.

Revert to defaults by issuing the command “CPD”.

Virtual Outputs

As well as the 16 real outputs that we have been using up to now VIOM also has another 16 virtual outputs. These exist only within the processor and have no physical presence. They have many of the features of the real outputs but do not have any of the timing functions.

While we generally refer to these as outputs they are in fact both outputs and inputs. All the real outputs can use the virtual outputs as inputs. Confused?

So what are their use? Essentially they allow us to create far more complex rules than would normally be possible. Let us suppose that we want output 1 to be activated if either input 1 switch is made OR input 2 switch is made but only if input 3 switch is made. We can make use of virtual output 17 to achieve this.

First assign inputs 1 and 2 to (virtual) output 17:

```
WO17L1
WO17L2
```

Tell output 17 to use an OR function

```
WO17J1
```

Remove input 1 from output 1 list (assuming that we are starting from defaults):

```
WO1M1
```

Add input 3 and output 17 to output 1 list:

```
WO1L3
WO1L17
```

Finally tell output 1 to use the AND function

```
WO1J0
```

Now LED 1 will only flash when input 3 AND output 17 are active and output 17 will only be active when either input 1 OR input 2 switches are made.

Revert to defaults by issuing the command “CPD”.

Inverting Inputs

VIOM also has the ability to invert inputs when used on specific outputs. As with the multiple input lists VIOM will maintains a list of inputs which need to be inverted for a particular output.

As an example let us suppose that we want LED 1 to flash when the switch on input 1 is made and the switch on input 2 is not made, but not when switch 2 is made. Additionally we want LED 2 to flash when both switch 1 and switch 2 are made. Setting up output 2 is straightforward:

```
WO2L1
WO2L2
WO2J0
```

Thus output 2 will be active when both inputs 1 AND 2 are active.

Output 1 starts the same:

```
WO1L1
WO1L2
WO1J0
```

However we now need to add input 2 to the list of inputs to be inverted when used by output 1:

```
WO1P2
```

Now operate switch 1 and LED 1 should flash. Operate switch 2 and LED 2 should flash and LED1 should go out.

To remove the inversion from input 2 on output 1 use the command “WO1Q2”.

Revert to defaults by issuing the command “CPD”.

Concluding Outputs

This concludes the section on output control at least by the inputs. We can also control outputs from the Timer / Counters and via the Serial Port but we will leave that to later sections.

If you have worked through this section and can understand it then you are a long way towards creating complex applications which involve the logical manipulation of inputs.

The following sections, dealing with the parameters which can be assigned to inputs can add even more versatility to your applications.

Input Recognition Period

When we were working with the outputs we used a recognition period to delay the output activating. We can also insert a delay on an input using similar commands: “WI1A25” will set the Recognition Time to 25 and “WI1B1” will set the Recognition Units to 100mS. You will see that the only difference in the command is that the second letter has changed from a ‘O’ to an ‘I’.

This recognition period will be applied to all outputs which use this input. To see how the recognition periods affect operation try the following code:

```
WO1L1
WO1L2
WO1J0

WO2L1
WO2L3
WO2M2
WO2J0
WO2A30
WO2B1

WI2A30
WI2B1
```

The first part sets up output 1 to be the AND of inputs 1 and 2. The second part sets up output 2 to be the AND of inputs 1 and 3 with an output recognition period of 3 seconds. The third part sets the recognition period for input 2 to 2 seconds.

As you operate the switches note that the recognition delay on output 2 is always 3 seconds irrespective of the order of operation of the switches. Output 1 will activate a lot quicker if switch 1 is operated after switch 2. This is because switch 2 has a recognition period of 3 second while input 1 will have the default of 100mS.

Now make outputs 1 and 2 latching using the commands “WO1K1” and “WO2K1”. Operate switch 1 and then operate switch 2 for a period less than 3 seconds. Notice that output 1 does not activate. Now operate switch 3 for less than 3 seconds and notice that output 2 does activate.

Revert to defaults by issuing the command “CPD”.

A Word about Anti-Bounce

This is one feature of VIOM that is hard to see the effect of. Sometimes mechanical switches, relay contact and the like can “bounce”, that is operate then break then operate again. This can happen several times. This kind of behaviour is particularly bad when you are using a counter as it may, indeed will, produce an erroneous result. Similar problems can exist if you are working in a noisy (electrically speaking) environment.

Enabling the anti-bounce feature will help to eliminate these problems by ensuring that when the input changes it does so for an extended period. Of course this will mean a slower response to changes but where there are problems anti-bounce may provide an answer.

Input Latching

As with outputs, inputs can be latched and reset. This results in a different form of behaviour of the affected outputs. The commands for latching are of the same form as for the outputs with “WI...” replacing “WO...”.

To show the effects of latching an input enter the following code:

```
WO1L1
WO1L2
WO1J0

WO2L1
WO2L3
WO2M2
WO2J0

WI3K1
```

The first part sets up output 1 to be the AND of inputs 1 and 2. The second part sets up output 2 to be the AND of inputs 1 and 3. The third part sets input 3 to be latching.

Operate switches 1 and 2. LED 1 should now flash. When you break either switch 1 or switch 2 the LED should extinguish. Now when you operate switches 1 and 3 LEDs 1 and 3 should flash (LED 3 will flash because by default it is controlled by input 3). Release switch 3 and the two LEDs should continue to flash. Release switch 1 and this time LED 1 should go out (because it is controlled by inputs 1 AND 3) but LED 3 should continue flashing.

Finally send the command “WI3U” to reset input 3.

Although we reset input 3 via the Serial Port, we could equally well have defined another input as a reset. Issue the command “WI3G2” to set input 2 as the reset for input 3. Now repeat the test above but this time operate switch 2 to reset input 3 instead of using the Serial Port.

Revert to defaults by issuing the command “CPD”.

Input Active High and Low

Inputs as well as outputs can be defined as either active high or active low.

Each input is fed into a CMOS gate and is pulled up to the VIOM 5 Volts rail by a resistor. In the case of our test arrangement we have a switch wired from the input to 0 Volts. Therefore when the switch is open (inactive) the input will be at 5 Volts (high) and when it is closed (active) the input will be at 0 Volts (low). Thus we define our inputs as being active low.

We can demonstrate active high operation with the following code:

```
WO1L1
WO1L2
WO1J0

WI1H1
WI2H1
```

The first part sets up output 1 to be the AND of inputs 1 and 2. The second part sets inputs 1 and 2 as active high. The outputs will now be active when switches 1 and 2 are open. That is LED 1 will flash when both switches are open and it will be extinguished when either are closed. notice as well that LED 2 will flash when switch 2 is open (because of the default condition that output 2 is controlled by input 2).

Revert to defaults by issuing the command “CPD”.

Concluding Inputs

This concludes the section on input control. Combined with the operations possible on the outputs this allows great flexibility for stand alone systems.

We will next deal with reading from the VIOM and controlling outputs directly from the host.

Reading Inputs - Overview

We can read the current status of the inputs directly from the host via the Serial Port. There are three methods of reading: on demand, periodically and whenever an input changes. The form of the returned information will depend on the method employed and this will be explained as we go along.

While it is certainly possible to read the information with a program such as HyperTerminal for many applications will require special software and so the responses are designed with that in mind.

It is possible to program VIOM to provide information by more than one method at a time. For example a program may need information whenever an input changes but request periodic information in order to make sure it stays in sync with VIOM. To enable the program to differentiate the source of the data it is possible to program prefixes for each method of reading.

Reading Inputs On Demand

This is the simplest way of reading the status of the inputs. The command is “CIN”. When you use this command you will notice that the response is either “Active” or “Inactive”. This is not the state of the input terminal itself but is the logical value as processed with the active high, active low and latching commands discussed earlier.

Try using the active high and active low commands on the inputs and see the new responses.

The response generated here is in a form we will refer to as Long Messages (we will discuss this in more detail later). For software engineers this is not particularly easy to interface with. We therefore have another form of response which we refer to as Short Messages. In this form the returned data is a string of 1s and 0s (the first character relates to input 1, the last character to input 16). To switch to short Messages use the command “CMS”. Now re-issue the “CIN” command an note the difference.

Revert to defaults by issuing the command “CPD”.

Reading Inputs Periodically

VIOM can be programmed to give periodic updates of the data. The period can be every 100mS, once per second, once per minute, once per hour or once per day.

In detail the commands are:

CIR0	No periodic update (default)
CIR1	Update every 100mS
CIR2	Update every second
CIR3	Update every minute
CIR4	Update every hour
CIR5	Update every day

Note that the form of these reports is always Short Messages.

Use the command “CIR” to read the current report period.

Reading Inputs On Change

VIOM can also report whenever an input changes state. To enable this function use the command “CIC1”. To disable the function use “CIC0”. To find if the function is enabled use “CIC”. Note that the form of these reports is always Short Messages.

It is worth trying this function by enabling it and then operating and releasing the switches.

Adding a Prefix to Input Reports

As we noted in the introduction to reports it can often be useful to include a prefix to the reports. The commands for setting and reading these prefixes are:

CIP0	Read the current prefixes
CIP1	Set the prefix for change reporting
CIP2	Set the prefix for periodic reporting
CIP3	Set the prefix for on demand reporting

If the last three commands are followed by a character then that character will be used as the prefix. If no character is used then no prefix is used.

Try setting some prefixes and then remove them. Try reading them with the “CIP0” command. when you have finished revert to defaults by issuing the command “CPD”.

Reading Outputs

Unlike the input reports, output reports can only be generated on demand. However as well as reading all outputs (real and virtual) it is possible to read just real outputs, just virtual outputs or individual outputs.

Relevant commands are:

COA	Read all outputs
COR	Read all real outputs
COR	Read all virtual outputs
YC	Read individual output (the command is followed by the number of the output).

Notice that the report tells you exactly the state of the output, including the which part of the on and off period cycle the output is in. This would correspond to the state of the LED in our earlier tests.

After playing with these commands and setting the outputs in various states try switching to Short Messages and comparing the results with the reports issued by the inputs. Because the outputs can take on more states the report string will include the numbers 0 to 4. These numbers correspond to:

0	Inactive
1	recognition Period
2	On Period
3	Off Period
4	Off and waiting or reset

Writing to Outputs Overview

All outputs, real and virtual can be controlled via the Serial Port. However when controlled directly from the Serial Port none of the timing parameters are available. The output is under total control of the host. It is possible to make use of the output timing parameters by writing to a virtual output and then using that to control a real output. This will be explained later.

It is possible to write to individual outputs or to either all real or all virtual output simultaneously.

Before an output can be controlled by the Serial Port it must have its source of control changed. For example use the command “WO1N3” to control output 1 via the Serial Port. To revert its control back to the inputs use the command “WO1N1” (for the curious “WO1N2” used by the Timer / Counters).

Writing to a Single Output

Use the command “WO1N3” to control output 1 via the Serial Port. Output 1 can now be made active by using the command “XA1” and made inactive by using the command “XB1”.

You can take advantage of the timing, latching and other output functions by controlling a virtual output from the Serial Port and then using the virtual output to control the real output. First revert to defaults by issuing the command “CPD” then use the following code:

```
WO17N3
WO1M1
WO1L17
```

This code will put virtual output 17 under the control of the Serial Port, remove input 1 from the output 1 list of controlling inputs and finally use output 17 to control output 1. Now when you issue the command “XA17” the LED will flash and it will be extinguished with the command “XB17”.

This technique is also useful for combining local and Serial Port control. If we re-map input 1 to the output 1 list with “WO1L1” and AND the inputs with “WO1J0” we will now only be able to control output 1 when switch 1 is made.

Revert to defaults by issuing the command “CPD”.

Writing to a Multiple Outputs

Four commands are available for multiple writes, two for real outputs and two for virtual outputs. these are summarised below:

Commands starting with YA will write to the virtual outputs and will return their status (in Short Message form).

Commands starting with Y will write to the virtual outputs but will not return their status.

Commands starting with ZA will write to the real outputs and will return their status (in Short Message form).

Commands starting with Z will write to the real outputs but will not return their status.

After the first character or two, there should follow a string of sixteen 1s and 0s to specify the new set of conditions, with a 1 indicating active and a 0 indicating inactive. the first character of the string is equivalent to either real output 1 or virtual output 17 (depending on which command is used).

As an example “ZA1000100010001000” will activate real outputs 1,5,9,and 13 and deactivate all the other real outputs.

Before an output will act upon these commands its control must be allocated to the Serial Port. Thus in the example above if only outputs 1 to 8 have had their control allocated to the Serial Port the outputs 1 and 5 will be activated and outputs 2,3,4,6,7 and 8 will be deactivated. Real outputs 9 to 16 will be unaffected.

If are used they will return the current status of all the outputs irrespective of whether they are controlled by the Serial Port or not. these commands are useful where the VIOM is being controlled by a host which can then verify that the command has been received and correctly acted upon.

The commands “Z...” or “Y...” will not return any value. These commands can be used to communicate between two VIOMs connected via a RS232 link so the outputs of one VIOM will mimic the inputs of the other. Set the input reporting to periodic, set the prefix for this reporting to ‘Z’ or ‘Y’ and enable control of the outputs to the Serial Port. Now connect the two VIOMs via a null modem lead and the state of the inputs on one will be mimicked by the other.

Revert to defaults by issuing the command “CPD”.

Timer / Counters Overview

VIOM has eight 32 bit Timer / Counters. Each can be individually configured as either a timer or counter. While their value may only be read via the Serial Port they are able to activate an output when their level reaches a predetermined level.

Each Timer / Counter is controlled by a single input which can also be a virtual output, thus making it possible to control them by a combination of inputs.

Timing Functions

The mode of the Timer / Counters can be set to either timer or counter. For this section we need to set Timer / Counter 1 to counter mode with the command “WT1C1”. We will also need a controlling input. let us choose input 2 so the command is “WT1A2”. Enter these commands and then operate the switch on input 2. Timer / Counter 1 will now start to run.

We can read the Timer / Counter with the command “RT1G”. The timer has a resolution of 10mS and the returned value is in mS. therefore you will notice that the last character of the returned value is always 0. If you break switch 2 and read the timer you will notice that the time remains constant. If you now re-operate switch 2 the timer starts to increment again.

Once again open switch 2. This time we will reset the timer by using the command “WT1F”. If you read the timer now you will find the returned value is 0.

In this mode the timer will accumulate the active periods for its assigned input. If you need the individual active times for an input you would need to reset the Timer / Counter prior to each new activation. This can be done automatically by changing from the default reset mode to reset on start with the command “WT1D1”. Now the counter will be reset each time the input becomes active.

After issuing the command try the following. Operate switch 2 and leave it operated for several seconds. Read the timer to verify that the timer is actually running. Open the switch, read and note down the returned time. Now re-operate the switch briefly. Again read the timer and compare the new time with the old. It should be shorter showing that the timer has indeed reset.

Revert to defaults by issuing the command “CPD”.

Counting Functions

We set the mode of the Timer / Counter to counter mode with the command “WT1C0”. as with the previous section we use switch 2 as the controlling input with the command “WT1A2”.

The register is cleared in the same way as if it were a timer with the command “WT1F” and read in the same way with “RT1G”. Having reset the register and verifying that it has been reset with a read, operate switch 2 several times. Now read the register an check that the counter is counting.

As well as counting up the counter can count down. We can designate input 3 to count down with the command “WT1H3”. Apply the command and operate switch 3. Again use command “RT1G” to read the register and note that the count has been reduced by 1. Operate switch 3 several more times so that the count is reduced to zero.

Having got to zero operate switch 3 again. note that the count does not go negative but remains fixed at zero. This is intentional, the count will never go below zero. One other point to remember is that the count down function is only enabled when there is a controlling (or count up) input designated.

Note that there is no equivalent of the reset on start as used in timer mode. This parameter is ignored in counter mode.

Revert to defaults by issuing the command “CPD”.

Controlling Outputs

Timer / Counters can be used to control outputs. In this mode of operation a level is set in a register and when this value is exceeded the output (real or virtual) is activated. When controlled in this manner none of the timing parameters of real outputs are directly available. However they can be used if the Timer / Counter controls a virtual output which in turn controls a real output. This is very similar to the technique used with the Serial Ports.

As with control from the Serial Port, we need to assign the control of the output to Timer / Counters.

Let us control output 3 from Timer / Counter 1 We will need to assign the control of output 3 to Timer / Counters with the command “WO3N2”. The mode will be set to timer, the threshold to 5 seconds and the control input to 2:

```
WO3N2
WT1A2
WT1B3
WT1C1
WT1E5000
```

Now operate switch 2. After 5 seconds LED 3 will come on and stay on. You can now release the switch. Issue the command “WT1F” to reset the timer and the LED should go out.

Now try changing the reset mode to reset on start with the command “WT1D1”. Operate switch 3 and again LED 3 will come on and stay on. Now release the switch and re-operate it. The LED will go out when the switch is re-operated (because the timer is reset) and will come on again after a further 5 seconds.

Customising the VIOM Interface

VIOM has a number of features which are purely concerned with the user interface. We have already mentioned long and short messages. Long messages are intended to interface with human beings via a terminal type program such as HyperTerminal. They are returned from the VIOM in a verbose form which can be directly read from the program

Short messages on the other hand are intend to interface to user programs. The returned values consist of a simple acknowledgement or the requested data in ASCII format. All messages whether short or long are terminated by a carriage return and line feed.

To switch to short messages use the command “CMS”. To revert to long messages use the command “CML”.

In some cases it is desirable to positively identify error messages as distinct from information messages. This can be achieved by prefixing error messages with a character. Use the command “CEA...” followed by a character to set the prefix. A useful character is ‘<‘ which should be used when two VIOMs form part of a data link. The command for this would look like “CEA<“.

To turn off prefixes use the command “CEA”. you can read the currently used prefix by issuing the command “CER”.

VIOM can be programmed to echo back any characters it receives via the Serial Port back to the host. This is the default condition. The facility can be turned off with the command “CR0” and re-enabled with the command “CR1”. You can interrogate the current condition without affecting it with the command “CR”.

At start up VIOM issues the message “VIOM Started”. This can be inhibited with the command “CSM0” and next time that VIOM is powered up the message will not be sent. It can be re-enabled with the command “CSM1”. You can interrogate the current condition without affecting it with the command “CSM”.

Finally, while not really part of the interface, you can read the current software version of VIOM with the command “CSV”. You cannot of course change this value.

Transferring Files

When writing complex programs for VIOM or you need to program multiple units it is useful to be able to download the complete code without having to type it all in again. VIOM is capable of reading text files provided it is aware that the file is about to be downloaded. It needs to know this because each time a command is issued it will store the new data in non-volatile memory. If the commands are issued too quickly it will not have finished processing one command when the next arrives.

To overcome this problem the command “FDL” warns the VIOM that a file download is about to begin. VIOM will now take two actions. It will inhibit any storing of data and it will not respond to any commands with return messages. When it does not receive any further messages for a period of approximately half a second it will assume that the download is complete. At this point it will store all the new data and re-enable return messages.

It can also be useful to comment your code. Any line that begins with ‘<‘ is totally ignored by VIOM until it detects the carriage return and line feed characters.

Taking an earlier example a text file download may look like this:

```
FDL
<Set up output 1
WO1L1
WO1L2
WO1J0
<set up output 2
WO2L1
WO2L3
WO2M2
WO2J0
<Set up input 3
WI3K1
```

Note that there is no need to take any action at the end of the file download as VIOM will automatically revert to normal operation.

Reading Multiple Parameters

When debugging code for VIOM it can be tedious to have to interrogate each parameter of interest individually. VIOM therefore has the ability to report on groups of parameters.

If you are interested in a particular parameter on all inputs you can use the command “DA...”. For example if you need to know which inputs are latched use the command “DAK”. The last letter is the same as the letter used to read or write the value individually. This command will now return the latching parameter for each of the outputs.

A similar command exists for the inputs (“DB...”) and the Timer / Counters (“DC...”).

On the other hand you may be interested in all the parameters for a particular output. These can be obtained with the command “DO...”. For example the command “DO1” will produce a list of parameters for output 1.

A similar command exists for the inputs (“DI...”) and the Timer / Counters (“DT...”).

Writing Multiple Parameters

When programming the same parameter to all outputs there is no need to use multiple commands. It is possible to write the same parameter to all outputs simultaneously.

For example suppose that you wanted to make all outputs latching. You could issue a series of commands “WO1K1”, “WO2K1” , “WO3K1” etc. Alternatively you could use the command “UAK1” and this would have the same effect. The command will work with any parameter for all outputs real and virtual. If the parameter does not exist for a virtual output it will be applied only to the real outputs and ignored for the virtual outputs.

Other useful commands are:

UOM will clear the list of controlling inputs on all outputs.
UOQ will clear the list of inverting inputs on all outputs.
UTF will clear all Timer / Counters.

Conclusion

By now you should have a good idea of the capabilities of VIOM. The best way forward is to experiment with it, think up your own projects and enjoy playing with VIOM.

Phaedrus Limited
Unit 1
Darwen Enterprise centre
Railway Road
Darwen, England
+44 (0)1254 772622
tech@phaedrusltd.co.uk
www.phaedrusltd.co.uk