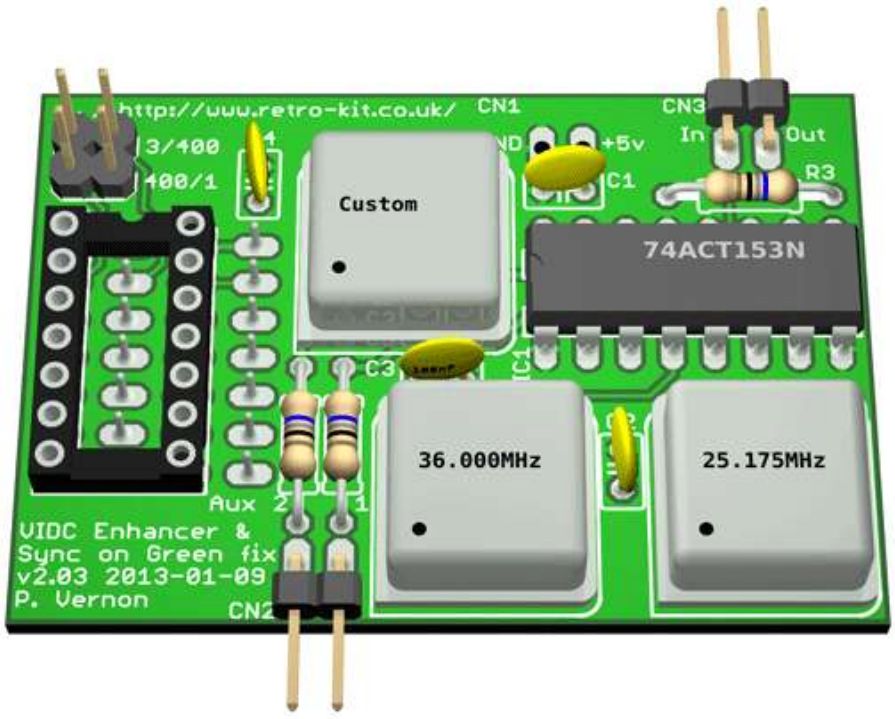
Retro-Kit

AutoVIDC v2.12

VIDC Enhancer Control software

User and Programmer's Reference Manual



Improving monitor compatibility and delivering higher screen resolutions
for the Acorn Archimedes range of computers

Contents

Copyright notice

Disclaimer

AutoVIDC in combination with a VIDC Enhancer provides an easy way for the end user to control the clock speed of the VIDC chip fitted to the Archimedes range of computers.

With a suitable oscillator fitted to a VIDC Enhancer it is possible to significantly over-clock the VIDC chip which could result in damage to, or foreshorten the life of the VIDC Enhancer, VIDC chip or related systems in the Archimedes. If you perform any over-clocking activities using AutoVIDC, you undertake that you will not hold Retro-Kit or Andreas Barth responsible for any damage that you may cause.

The Software is supplied "AS IS". Retro-Kit disclaims all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The user must assume the entire risk of using the Software.

Introduction

AutoVIDC is a re-locatable module used to control a VIDC Enhancer that provides alternative clock frequencies for the Acorn **VID**eo **C**ontroller chip called VIDC which is responsible for the video output in the Archimedes range of computers.

By using faster VIDC clock speeds delivered by a VIDC Enhancer, higher screen resolutions and refresh rates can be achieved by the Archimedes.

The software is intended to work with RISC OS 2.0x and RISC OS 3.xx to provide the ability to control up to two standard VIDC Enhancers at the same time or a single three oscillator VIDC enhancer. The latter can be used to drive the VIDC chip at up to four different clock speeds when including the original Archimedes VIDC clock signal.

How AutoVIDC works

When AutoVIDC is loaded into memory it checks the current MODE against its internal MODE maps to see if it has been defined as requiring an enhanced VIDC clock signal in order that it can set the appropriate VIDC clock signal. Please refer to Appendix A for default MODE map definitions.

During initialisation, AutoVIDC determines the type of hardware being used to control the VIDC Enhancer and cycles through all of the possible VIDC clocks that may be available. This process can cause the screen to go blank for around 1 second.

Once the hardware and available clocks have been determined during initialisation, AutoVIDC waits in the background for **Service_ModeChange** service calls to be broadcast by RISC OS whenever the screen MODE changes. When such a service call is received, AutoVIDC checks to see what VIDC clock frequency the MODE requires and provides it by setting the control lines of the VIDC Enhancer accordingly. This feature us permanently disabled on later Acorn hardware as it is not required. You can refer to Appendix B for more details on the possible combinations of the relevant control lines when AutoVIDC is controlling them.

When a higher VIDC clock speed is required, the VIDC Enhancer hardware swaps out the original system VIDC Clock signal, replacing it with an alternative VIDC clock signal suitable for the current screen MODE. The VIDC Enhancer could be used to provide a clock signal at any higher rate which is suitable for VGA and SVGA screen mode generation but the extra clock signals are typically 25.175MHz and 36.000MHz. This matches later Acorn machines such as the A5000.

If the VIDC Enhancer does not need to be activated, the original Archimedes 24.000MHz system clock is passed through the VIDC Enhancer to the VIDC chip transparently.

Differences between AutoVIDC v1 and AutoVIDC v2

There are many differences between AutoVIDC v1 and AutoVIDC v2 of which the main additional features in version 2 are described here:

- Fully supports RISC OS 2 and 3.
- Caters for multiple VIDC clock speeds and maps different screen MODEs accordingly.
- Flexible enough to be able to control a single 25.175MHz or 36.000MHz VIDC Enhancer; two VIDC Enhancers fitted in tandem or a single three oscillator VIDC Enhancer design such as the Retro-Kit Ultra VIDC Enhancer.
- AutoVIDC 2.07 and above supports I^2C bus controlled Watford VIDC Enhancers too. For more on this, see the AutoVIDC compatibility section.
- AutoVIDC is Monitor type and OS version aware enabling automatic delivery of true VGA scan rates for re-mapped MODEs when the Archimedes is connected to a VGA or SVGA monitor and RISC OS 3 is present.
- Can be controlled and interrogated by use of various SWI's.
- AutoVIDC 2.01 and above contain a feature that allows RISC OS to indirectly control what clock speed is provided by the VIDC Enhancer.

Installing AutoVIDC

The AutoVIDC module can be installed in several ways as detailed below.

Booting from floppy

AutoVIDC can be used directly from a floppy disc which is bootable and loads the AutoVIDC module as the system boots.

Installing into Uniboot

If you have a hard drive with the Uniboot boot structure, AutoVIDC can be placed into the PreDesk folder to cause it to be loaded early on in the boot sequence prior to the Desktop being launched.

Installing in flash ROM

If a suitable podule is fitted that has a flash ROM fitted, it is possible to add the AutoVIDC module to the podule's ROM image so that the module is loaded immediately after the Archimedes completes its POST function when powered on.

This method of installation is the most advanced but enables all of the VIDC Enhancers capabilities as soon as the machine has powered on and as such, it is the preferred method of installation.

If you intend to install the module into a suitable podule's flash ROM, please refer to the podule manufacturer's manual for more information on how to do that.

Podules known to work with flash ROM installation

Simtec 8-bit and 16-bit IDE podules have a suitable flash ROM and are known to work effectively with AutoVIDC after being flashed with !Snafu from Simtec.

Compatibility with VIDC Enhancer hardware

AutoVIDC should be compatible with any single, dual or triple oscillator VIDC Enhancer hardware that is controlled by the Aux I/O signals available in the Archimedes A300, A400, A400/1 and A3000 series of machines as long as the following information remains consistent.

Atomwide, Beebug and AutoVIDC variants

Traditionally, single oscillator VIDC Enhancers have been controlled by the "Aux I/O 1" signal which is found on pin 3 of the Aux I/O header or pin 17 of IC18 in the A3000.

Typically, these single VIDC Enhancers use a 36.000MHz oscillator and AutoVIDC expects that a 36.000MHz oscillator is controlled by the "Aux I/O 1" signal.

Where a single VIDC Enhancer uses a 25.175MHz oscillator (typically on ARM2 based Archimedes needing to produce VGA compliant refresh rates), AutoVIDC would expect to control the VIDC Enhancer by switching the "Aux I/O 2" signal which is found on pin 2 of the Aux I/O header or pin 18 of IC 18 in the A3000.

Dual VIDC Enhancers

Dual VIDC Enhancers allow an Archimedes to match the video output of the Acorn A5000 MODE for MODE. The pin assignment as described for single VIDC Enhancers carries through to dual VIDC Enhancer configurations where two single VIDC Enhancers are fitted to the same machine. The slower VIDC clock signal is controlled by the "Aux I/O 2" signal and the faster VIDC clock signal is controlled by the "Aux I/O 1" signal.

Retro-Kit Ultra VIDC Enhancer

The Ultra VIDC Enhancer is designed to not only match the Acorn A5000 MODE for MODE[1] but also allow a user to over-clock their VIDC chip and create custom MODEs with higher resolutions and refresh rates than are possible with the Acorn A5000 and related machines

Where a Retro-Kit Ultra VIDC Enhancer is fitted, the Aux I/O header pins aren't used as simple "on/off" selections; rather they're used together as a 2-bit data line giving four possible combinations.

The design of the Ultra VIDC Enhancer is faithful to the hardware mappings that have been set out for earlier VIDC Enhancer hardware designs and it adds in support for a fourth VIDC clock signal when both Aux I/O pins are activated together.

Please refer to Appendix B for the relevant Aux I/O pin combinations and the VIDC clock signal produced.

Watford VIDC Enhancers

Watford Electronics VIDC Enhancers are supported by AutoVIDC 2.07 and above. These can be single 36.000MHz or dual 25.175MHz and 36.000MHz types. AutoVIDC actively detects these VIDC Enhancers by querying the I^2C bus to determine their presence. Setting the Sync. Polarity when using the "Super" VIDC Enhancer is fully supported as of version 2.10.

[1] With the only difference being those caused by the older machines having no Sync. Polarity hardware.

AutoVIDC '*' commands

The AutoVIDC star commands are broken down into two separate groups which are for control and configuration.

AutoVIDC control

The AutoVIDC control commands allow a user to enable, disable and view the status of features that AutoVIDC controls.

AutoVIDCEnable

Enables AutoVIDC where it has been previously disabled at the command line or by the AutoVIDC_DelegateControl SWI.

AutoVIDCDisable

Disables AutoVIDC - When disabled, the VIDC Enhancer is turned off and AutoVIDC ignores all MODE changes service calls. All other features of AutoVIDC are still available.

AutoVIDCEnableOC

Enables control of a third oscillator which is used for over-clocking where fitted to a suitable VIDC Enhancer.

AutoVIDCDisableOC

Disables control of a third oscillator which is used for over-clocking where fitted to a suitable VIDC Enhancer.

AutoVIDCAutoSelOn

Enables the RISC OS VIDC clock selection feature letting AutoVIDC examine the Video Controller's hardware registers for the clock speed it expects to be running at for the current MODE. If AutoVIDC thinks that it can deliver the required clock speed, it will automatically switch to it.

As RISC OS only supports 24MHz, 25.175MHz and 36.000MHz clock speeds, this feature only supports switching to those clock speeds provided by the Ultra VIDC Enhancer. Switching to the over-clocked VIDC clock speed is therefore not supported with this feature.

AutoVIDCAutoSelOff

Disables the RISC OS VIDC clock selection feature so AutoVIDC must rely exclusively on the pre-defined and user modifiable MODE lists.

AutoVIDCTogglePOST

Toggles the RISC OS 3 VIDC Clock check that is performed when the Archimedes is power on during the RISC OS POST.

AutoVIDCStatus

Reports back the current MODE, Monitor type, VIDC clock speed, whether the RISC OS mode mapping is enabled or not and whether the VIDC POST check is enabled or not.

MODE <modenumber>

Allows the user to change the screen mode from the command prompt.

AutoVIDC configuration

AutoVIDC allows the user to configure four different MODE maps which are used to control up to three additional VIDC clock sources at the same time and you can set the speed of the over clocking oscillator in kHz when one is fitted.

Setting the over clocking oscillator speed

AutoVIDCsetOCkHz <speed in kHz>

Set the speed of the over clocking oscillator when fitted allowing RISC OS compliant games to compensate for the faster clock speed when playing audio.

The speed must be provided in kHz. Where a value is set exceeding 60000, the over-clocking facilities of AutoVIDC are completely disabled and the value is ignored.

Configurable MODE maps

Where a MODE is not defined in any MODE map, AutoVIDC now checks to see if the required clock speed can be provided and automatically switches to it. If this feature is disabled, the default 24.000MHz VIDC clock is used.

The first configurable MODE map defines a list of MODEs which AutoVIDC should ignore, leaving them under the control of other compatible software which is VIDC Enhancer aware.

The next three configurable MODE maps allow AutoVIDC to use a custom, 36.000MHz or 25.175MHz clock frequency with MODEs that require enhanced clock speeds.

There is also a fifth MODE map which is non-configurable and comes into play for the enhancement of specific modes when the Archimedes is connected to a VGA or SVGA monitor under RISC OS 3.

All five MODE maps are pre-configured for the most common screen modes making the use of AutoVIDC in a standard Archimedes configuration transparent to the user. Usually modifications to the MODE maps are only required when creating custom MODEs and/or over-clocking the VIDC clock. Please refer to Appendix A for the default MODE map definitions and their order of precedence.

The configurable MODE maps have pairs of commands available to them for setting and reviewing their current status.

AutoVIDCIgnore <ModeNumber><-|+> [...]

Set a specific MODE or list of MODEs that should not be controlled by AutoVIDC.

AutoVIDCIgnoreMap

Lists all 128 MODEs and shows their current configuration with respect to the Ignore map.

AutoVIDCOCset <ModeNumber><-|+> [...]

Set MODEs that require a VIDC clock speed greater than 36.000MHz.

AutoVIDCOCmap

Lists the current configuration of the MODE map controlling the selection of the VIDC clock speed which is greater than 36.000MHz.

AutoVIDC36set <ModeNumber><-|+> [...]

Set MODEs that require the 36.000MHz VIDC clock speed.

AutoVIDC36map

Lists the current configuration of the MODE map controlling the selection of the 36.000MHz VIDC clock speed.

AutoVIDC25set <ModeNumber><-|+> [...]

Set MODEs that require the 25.175MHz VIDC clock speed.

AutoVIDC25map

Lists the current configuration of the MODE map controlling the selection of the 25.175MHz VIDC clock speed.

The non-configurable MODE map

AutoVIDCVGAmap

Lists the MODE map configuration for the 25.175MHz VIDC Enhancer used only when connected to a VGA or SVGA monitor type under RISC OS 3. This map is used in conjunction with the standard 25.175MHz MODE map.

MODE map configuration syntax

Using any of the "set" commands allows custom configuration of each MODE map that AutoVIDC provides.

To configure a MODE to use a specific VIDC clock speed, use the following syntax:

*AutoVIDC***xx**set <$MODE_1$>(+/-) <$MODE_2$>(+/-) ... <$MODE_n$> (+/-)

e.g.

*AutoVIDC36set 31+

*AutoVIDC25set 28-

*AutoVIDC36set 31+ 29-

Automatic RISC OS VIDC clock selection

If the RISC OS VIDC clock selection feature is enabled, after checking the MODE lists, AutoVIDC interrogates the VIDC registers to see what clock speed the VIDC chip thinks it should be clocked at for the given MODE. If the clock speed is recognised as being provided by the VIDC Enhancer fitted, AutoVIDC automatically switches the actual clock speed to the correct one for that MODE.

The RISC OS VIDC clock selection feature brings the earlier Archimedes models into line with later models and significantly improves games compatibility.

For example, this feature is particularly useful where a game creates a custom MODE and assigns it to the first available MODE number. As the MODE number is dynamically assigned, it could change every time the game is launched subject to other custom MODEs being installed in memory. By allowing RISC OS to indirectly control the VIDC Enhancer, the correct clock speed is always provided with no pre-configuration required of the user.

The RISC OS VIDC clock selection feature is available in all versions of RISC OS from RISC OS 2.01 onwards and is enabled by default.

Over-clocking the VIDC chip with AutoVIDC

WARNING

Any over-clocking of the VIDC chip is done at your own risk and Retro-Kit does not advise that it is done unless you know exactly what you are doing.

Any form of over-clocking of the VIDC chip will require a suitable custom cooling solution to be fitted to your Archimedes.

The over-clocking features of AutoVIDC are disabled by default. This is to stop the accidental selection of a third oscillator where one is not fitted.

AutoVIDC behaviour when over-clocking is disabled

Whilst the over-clocking feature is disabled, the over-clocking MODE map is skipped when a **Service_ModeChange** event is received and neither can the SWI call AutoVIDC_SetClock be used to set the VIDC Enhancer to select the third oscillator.

Controlling the over-clocking features

To enable and disable the over-clocking features of the AutoVIDC module, use the commands AutoVIDCEnableOC, AutoVIDCDisableOC and AutoVIDCsetOCkHz.

AutoVIDC behaviour when over-clocking is enabled

Once enabled, the over-clocking features of AutoVIDC are automatically made available. By default, the AutoVIDC over-clocking MODE map has no MODEs defined as requiring an enhanced VIDC clock so any MODEs that need to be enhanced must be added to the over-clocking MODE map using the command AutoVIDCOCset.

Permanently enabling the over-clocking features

If you wish to permanently enable the over-clocking facilities and use a set of pre-defined MODEs requiring enhancement using an over-clocked VIDC, this can be scripted using the * commands available. The script can then be placed in the PreDesk folder of the Uniboot boot structure or your existing !BOOT sequence could be modified to incorporate this.

For example, to enable over-clocking with a 40MHz oscillator with MODE 96, the following script could be used[2].

```
AutoVIDCEnableOC
AutoVIDCsetOCkHz 40000
AutoVIDCOCset 96+
```

[2] If !CustomVDU were modified to support the same clock speed as the over-clocked Ultra VIDC Enhancer the AutoVIDCOCSet command would not be required.

Programming your VIDC Enhancer

AutoVIDC SWI's

AutoVIDC provides support for programmers to control the VIDC Enhancer hardware through several SWI calls.

AutoVIDC_SetClock (SWI & 59380)

Set's the VIDC clock speed output by a VIDC Enhancer

On entry

R0 = Clock speed (0-3) - other values will return the current clock speed.

On exit

R0 = Current clock speed (0-3)

Interrupts

Interrupts are disabled
Fast interrupts are disabled

Processor Mode

Processor is in SVC mode

Re-entrancy

SWI is not re-entrant as all interrupts are disabled.

Use

This SWI is used to control the clock rate output by a VIDC Enhancer where multiple clock rates are available. The SWI is currently only applicable to Archimedes machines fitted with a VIDC1 chip.

The value of R0 corresponds to the possible available clock speeds that VIDC Enhancers usually supply which are:

0 - 24.000MHz (default Archimedes VIDC clock speed)
1 - 25.175MHz (VGA compatible clock speed)
2 - 36.000MHz (SVGA compatible clock speed)
3 - Undefined custom speed for over-clocking purposes

Depending on the VIDC hardware available, the values of R0 are used to set the Aux I/O pins Aux 1 and Aux 2 which are then used to control a suitable VIDC Enhancer board.

Where a value of 3 is specified, it is only acted upon if the over-clocking features of AutoVIDC are enabled.

Please see Appendix B to confirm the Aux I/O pin selections.

AutoVIDC_DelegateControl (SWI & 59381)

Delegates control of the VIDC Enhancer software entirely to a third party application.

On entry

R0 = 0 or 1

On exit

R0 is preserved

Interrupts

Interrupt status is not altered

Fast interrupts are not altered

Processor Mode

Processor is in SVC mode

Re-entrancy

SWI is re-entrant

Use

Allows a third party application to stop the AutoVIDC module from acting upon the **Service_ModeChange** (Service &46) service call implicitly delegating all responsibility to the third party application for setting the VIDC clock signal.

The values of R0 allow a third party application to disable and enable AutoVIDC's control of the VIDC Enhancer hardware entirely.

0 - Enable AutoVIDC to handle **Service_ModeChange** service calls

1 - Stop AutoVIDC from handling **Service_ModeChange** service calls

Notes

This SWI is functionally similar to the * commands AutoVIDCEnable and AutoVIDCDisable however the current VIDC clock setting is only affected when the delegation is rescinded by the calling application.

AutoVIDC_IsDelegated (SWI & 59382)

Returns the state of the AutoVIDC module's delegation status which indicates whether it has delegated control of the VIDC Enhancer to another application.

On entry

No parameters required

On exit

R0 = 0 or 1

Interrupts

Interrupt status is not altered

Fast interrupts are not altered

Processor Mode

Processor is in SVC mode

Re-entrancy

SWI is re-entrant

Use

Allows the calling program to determine whether AutoVIDC has delegated control of the VIDC Enhancer to another application or not.

The value returned in R0 corresponds to:

0 - Not delegated

1 - Delegated

AutoVIDC_IsOverClockingEnabled (SWI & 59383)

Returns the state of the AutoVIDC module's over-clocking features indicating whether they are available for use.

On entry

No parameters required

On exit

R0 = 0 or 1

Interrupts

Interrupt status is not altered
Fast interrupts are not altered

Processor Mode

Processor is in SVC mode

Re-entrancy

SWI is re-entrant

Use

Allows the calling program to determine whether the over-clocking facilities of AutoVIDC are enabled or not. This allows the calling program to intelligently use the AutoVIDC_SetClock service call with regards to using the over-clocking VIDC clock speed.

The value returned in R0 corresponds to:

0 - Disabled
1 - Enabled

Returns the default Sync.Polarity for any given MODE and MonitorType combination provided.

On entry

R0 = ModeNumber (0-127)

R1 = MonitorType (0-5)

On exit

R0 = 0-3 (See table below)

Interrupts

Interrupt status is not altered

Fast interrupts are not altered

Processor Mode

Processor is in SVC mode

Re-entrancy

SWI is re-entrant

Use

Used to retrieve the default Sync.Polarity for the requested MODE and Monitor Type combination.

The value returned in R0 corresponds to:

	V.Sync	H.Sync	VGA MODE
0	+ve	+ve	-
1	+ve	-ve	VGA 400
2	-ve	+ve	VGA 350
3	-ve	-ve	VGA 480

Notes

1. Under RISC OS 2, this SWI always returns a value of 0.
2. There is no way to determine the ***current*** Sync.Polarity from RISC OS, only the Sync. Polarity as it is configured for a given MODE and monitor type. If the Sync.Polarity were to be altered dynamically, this change would not be detectable by the system.
3. When running RISC OS 3, the MODE's Sync.Polarity is always returned, however for upgraded A300, A400, A400/1 and A3000 series computers, regardless of the Sync.Polarity returned, in reality, the actual Sync.Polarity is always 0 due to the hardware constraints of those machines.

This call simulates a Service_ModeChange call in order to re-initialise the AutoVIDC module after delegation has been returned to it regardless of whether a Mode change has actually occurred or not.

On entry

No parameters required

On exit

Registers are preserved.

Interrupts

Interrupt status is not altered
Fast interrupts are not altered

Processor Mode

Processor is in SVC mode

Re-entrancy

SWI is re-entrant

Use

This SWI is used internally by the AutoVIDC_DelegateControl SWI in order to re-initialise AutoVIDC after a 3rd party application has given up control of AutoVIDC.

This call allows a third party program to determine if a specific clock speed is available or whether a particular clock "slot" is available.

On entry

R0 = 0 to 3 or Clock speed in kHz (e.g. 25175)

On exit

R0 = -1 or 0 to 3

R1 = Clock speed in kHz

Interrupts

Interrupt status is not altered

Fast interrupts are not altered

Processor Mode

Processor is in SVC mode

Re-entrancy

SWI is re-entrant

Use

This SWI allows a third party application to determine whether a specific clock speed is available or not and act accordingly.

Example:

```
MOV R0,#25175                  ; Set the clock speed in kHz
SWI AutoVIDC_ClockAvailable    ; Is it there?
CMN R0,#1                      ; If it isn't there
MOVEQ R0,#0                    ; choose the 24MHz clock (always in slot 0)
SWI AutoVIDC_SetClock          ; pass R0 back into AutoVIDC to set the clock.
```

This would set the VIDC clock to 25.175MHz if the clock is available or to 24MHz if it is not present.

Typically where a 25.175MHz oscillator is fitted, the following code could be used also.

```
MOV R0,#1                      ; Set the hardware slot we're interested in.
SWI AutoVIDC_ClockAvailable    ; Is an oscillator fitted there?
CMN R0,#1                      ; If it isn't
MOVEQ R0,#0                    ; choose the 24MHz clock (always in slot 0)
SWI AutoVIDC_SetClock          ; pass R0 back into AutoVIDC to set the clock.
```

This second example assumes that the number one slot has a 25.175MHz oscillator fitted whereas the initial example does not assume any such detail and the 25.175MHz oscillator could be in any available slot on the VIDC Enhancer hardware. Also, if your software requires a 25.175MHz oscillator and a user had replaced it with a 31.5 or 32MHz oscillator, the first example will correctly drop down to using the built in 24MHz clock whereas the second example will switch to the 31.5 or 32MHz clock speed.

This call allows a third party program to determine if a VIDC Enhancer has been detected by AutoVIDC.

On entry

No parameters required

On exit

R0 = 0 (not-present) or 1 (present)

Interrupts

Interrupt status is not altered

Fast interrupts are not altered

Processor Mode

Processor is in SVC mode

Re-entrancy

SWI is re-entrant

Use

This SWI allows a third party application to determine whether a VIDC Enhancer is fitted or not.

Example:

```
SWI AutoVIDC_EnhancerPresent  ; Is an Enhancer present?
CMP R0,#1                     ; Check the result
BEQ R0,EnhancerDetected       ; Branch to a function if it's there.
BNE R0,NoEnhancer             ; Branch to a different function if not...
```

NOTE: If AutoVIDC cannot detect a VIDC Enhancer but one is fitted, AutoVIDC will report that the VIDC Enhancer is not present.

In the source code of !ArmSI it is apparent that it was written to identify different VIDC Enhancer configurations and the source code uses a variable called VIDCType% to track what sort of VIDC environment is configured however the detection code is incomplete.

A VIDCType% value of 2 indicates that a VIDC Enhancer is fitted so ArmSI can be modified to use AutoVIDC to detect VIDC Enhancers by adding the following lines of code near to the PCATS detection code as shown below:

```
REM *** Test for PCATS enhancer card (SWI Enhancer_HardwarePresent)
SYS &62A4D TO ;flags%
IF (flags% AND 1) = 0 THEN VIDCType%+=128

REM *** Test for AutoVIDC enhancer controlled hardware (SWI
AutoVIDC_EnhancerPresent)
SYS &59387 TO flag%
IF flag% THEN VIDCType%=2
```

AutoVIDC_EnhancerType (SWI & 59388)

This call allows a third party program to determine the type of VIDC Enhancer that has been detected by AutoVIDC.

On entry

No parameters required

On exit

R0 = 0, 1 or 2 where

- 0 = Acorn (A540, A30x0, A4000, A5000)
- 1 = Aux I/O (Atomwide, Beebug, Retro-Kit etc.)
- 2 = I2C bus (Watford Electronics)

Interrupts

Interrupt status is not altered

Fast interrupts are not altered

Processor Mode

Processor is in SVC mode

Re-entrancy

SWI is re-entrant

Use

This SWI allows a third party application to determine what type of VIDC Enhancer is fitted.

Example:

```
SYS "AutoVIDC_EnhancerPresent" TO present%

IF present% THEN

    SYS "AutoVIDC_EnhancerType" TO type%
    IF type% = 0 THEN PRINT "Native Acorn Enhancer."
    IF type% = 1 THEN PRINT "Aux I/O VIDC Enhancer present."
    IF type% = 2 THEN PRINT "I2C VIDC Enhancer present."

ELSE

    PRINT "No VIDC Enhancer detected."
```

Appendices

Appendix A – Default MODE map definitions

With the advent of v2.01 of AutoVIDC, the MODE lists are much less important than they once were for most versions of RISC OS. They are only really required for support of RISC OS 2.00 and where a user wishes to force the VIDC clock to a specific clock speed for a given screen mode that is not normal for that screen mode.

How MODE maps work

When AutoVIDC receives a **Service_ModeChange** service call, it processes its current MODE maps to check if the newly selected MODE requires an enhanced VIDC clock speed. The first time a MODE is encountered as requiring a higher clock speed in the MODE maps; AutoVIDC sets that clock speed and does not process any further MODE map definitions.

The default MODE maps are listed in their order of precedence as processed by AutoVIDC.

Ignore MODE map

This MODE map is provided to allow configurable compatibility with other VIDC Enhancer tools which can create their own MODE definition modules and control a VIDC Enhancer independently of AutoVIDC.

No MODEs are set to be ignored by AutoVIDC.

Over-clocking MODE map

This MODE map is only processed if over-clocking has been enabled. Refer to the over-clocking section of this manual for more information.

No modes are pre-configured to enable the VIDC clock signal capable of over-clocking the VIDC chip as it is anticipated that this will be down to the users' oscillator choice and custom MODE configurations.

36.000MHz – SVGA MODE map

MODEs 29 to 31 are pre-configured to provide a 36.000MHz VIDC clock speed.

25.175MHz – VGA MODE map

This mode map has different pre-configured settings depending on the version of RISC OS you have running.

The reason for this is that the screen geometry for VGA modes in RISC OS 2.00 delivers a 60Hz refresh rate without the need of the additional speed provided by the 25.175MHz VIDC Enhancer. Although RISC OS 2 achieves the 60Hz refresh rate, the screen modes are not VGA/VESA compliant.

RISC OS 3

MODEs 25 to 28 are pre-configured to provide a 25.175MHz VIDC clock speed.

RISC OS 2.00

No modes are pre-configured to provide a 25.175MHz VIDC clock speed.

RISC OS 3 - VGA and SVGA MonitorType MODE map

This final MODE map is only processed when a VGA or SVGA monitor is connected to an Archimedes running RISC OS 3.xx. It is used to enhance the MODEs that RISC OS 3 re-maps to make them available on a VGA capable monitor.

This MODE map is not configurable by the user.

MODEs 0 to 15 and 41 to 46 are configured to require a 25.175MHz VIDC clock speed.

Automatic RISC OS VIDC clock selection

AutoVIDC 2.01 and above provides support for versions of RISC OS from version 2.01 onwards to tell AutoVIDC which clock speed is required for the correct display of the selected MODE. This feature causes the clock speed to be switched to the requested speed regardless of whether the MODE appears in any of the MODE maps listed above.

There are two scenarios where this switch does not occur. The first is when a MODE is listed in any of the MODE maps listed above which override RISC OS. The second is where the VIDC clock selection feature itself is disabled in AutoVIDC.

Appendix B – AutoVIDC truth tables

The truth table shows the different combinations of the Aux I/O signals that are produced by AutoVIDC.

Aux 1	Aux 2	Clock speed	SWI SetClock value	Notes
L	L	24.000 MHz (System clock)	0	Standard VIDC clock
L	H	25.175 MHz (VGA)	1	Improves VGA monitor compatibility
H	L	36.000 MHz (SVGA)	2	Required to display MODEs 29-31
H	H	Over-clocked VIDC clock	3	Only available when over-clocking is enabled in AutoVIDC and a suitable VIDC Enhancer is fitted.

The following truth table shows similar details for the Watford Electronics I^2C bus controlled VIDC Enhancers.

p0	p1	Clock speed	SWI SetClock value	Notes
L	L	36.000 MHz (SVGA)	2	Required to display MODEs 29-31
L	H	Not Used	Not Used	Not Used
H	L	25.175 MHz (VGA)	1	Improves VGA monitor compatibility
H	H	24.000 MHz (System clock)	0	Standard VIDC clock

If a Watford VIDC Enhancer only has a 36.000MHz oscillator, AutoVIDC substitutes the values that are normally used for the 25.175MHz clock selection with values for the selection of the 24MHz clock. This ensures that a VIDC clock is always provided to the VIDC chip regardless of the mode selection or VIDC clock SWI request.

Appendix C – VIDC Enhancer detection

As of AutoVIDC version 2.07, the software supports all known VIDC clock selection circuitry. When first loaded, AutoVIDC goes through a specific process to determine the type of VIDC Enhancer circuitry fitted to the Archimedes range of computers which is as follows.

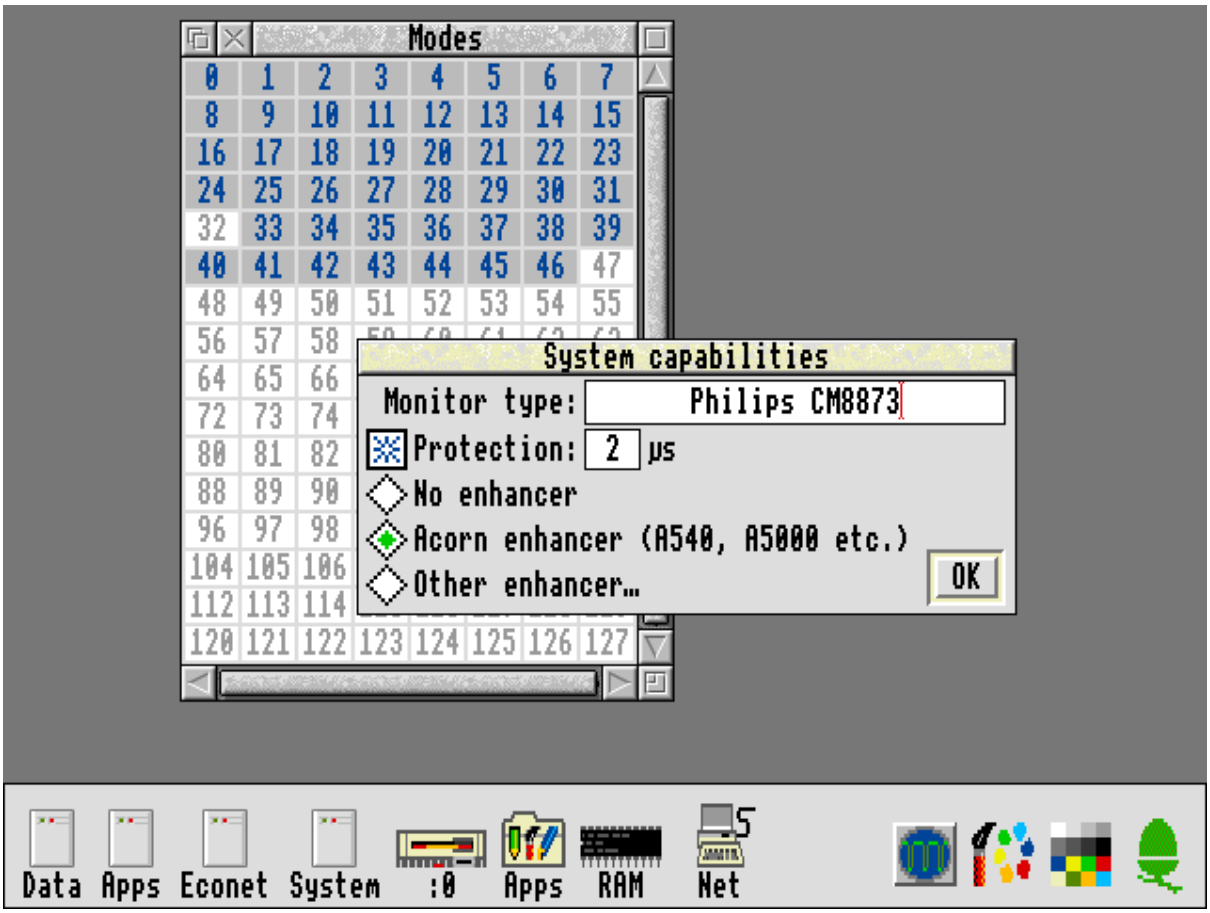
1. Determine if Native Acorn VIDC clock selection circuitry is present
2. Determine if a Watford Electronics VIDC Enhancer is fitted and what type it is
3. Assume that if neither the Native Acorn nor Watford Electronics clock selection circuitry is fitted that an Aux I/O controlled VIDC Enhancer is present.

Aux I/O VIDC Enhancer support can only ever be assumed whereas Native support and WE VIDC Enhancers can be reliably detected.

Appendix D – Configuring !CustomVDU to create your own screen modes

With the advent of AutoVIDC v2.01, when using a VIDC Enhancer with RISC OS 3, there is no need to tell !CustomVDU which Aux I/O pin to select when configuring the program. This feature also allows CustomVDU to support Watford Electronics VIDC Enhancer hardware which uses the I^2C bus to control clock selection.

When configuring !CustomVDU, set the VIDC Enhancer setting to “Acorn (A540, A5000 etc.)” as shown in the image below and allow AutoVIDC to take care of the rest.



By doing this, AutoVIDC’s RISC OS clock selection routines control the VIDC Enhancer and therefore, the MODEs created by !CustomVDU are fully supported with the correct clock speed being switched to automatically regardless of the VIDC clock circuitry fitted.

For further information on how to create your own custom screen modes, please see the !CustomVDU documentation.