DVDO
by Anchor Bay

Serial and IR Automation Specifications and
Programming Guide
for iScan VP20, VP30, VP50 and VP50^{PRO}

Revised - December 2007

Document Contents

0 Preface

Thank you for purchasing a DVDO iScan VP$_{xx}$ Series video processor. We believe the iScan will become a favorite device in your multimedia presentation system due to picture quality, ease of use, and the level of control the iScan gives you or your customer over the processed signal. This document is intended to cover the supplemental control functionality that is available for the iScan VP20, VP30, VP50, and VP50^{PRO}.

0.1   Information Warranty Statement

The information presented within this guide is known to be accurate at the time of publication. However, we at Anchor Bay continually strive to improve our products by offering new functionality and features which may in some cases require modification of or addition to the information contained within this document. As always, one should periodically check our website (www.anchorbaytech.com) for updates to our software and the support documentation. Anchor Bay (Anchor Bay Technologies, Inc.) or its subsidiaries, agents, and/or investors may not be held liable for technical inaccuracies or omissions that affect an installed system or device. Responsibility for correct operation of the iScan product within the installed system lies with the installing or integration party (i.e. a Home Theater Installer or the end-user or "customer").

The iScan VP$_{xx}$ video processors are capable of outputting more types of video signals than many display devices can support. Typically, the menu based user controls have some safety features that prevent most users from executing a command or function that would result in a loss of picture or damage to the display device (typically CRTs fall into this category), or may overwrite settings without any prompt. Direct access to the control system via discrete commands may circumvent these safeties in some cases. Careful planning should be used when configuring the iScan within the system to ensure that it behaves within the design constraints of the installed system and the capabilities of the installed support hardware. If you have just read this and don't understand what it means – *PLEASE* contact an authorized DVDO product installer for consultation and installation help. Not getting a picture from the iScan does not necessarily indicate a failure of the iScan device – the display device may not support the selected output format, or there may be some other circumstance which would need to be investigated and remedied to resume viewing operation of the presentation system.

If you are having trouble with this document, or the operation of the iScan VP$_{xx}$ device, please first refer to the User's Manual included with your device. If you are still not able to resolve your issue, please call our Technical Support Hotline 9AM-5PM Pacific Time, at (U.S. Domestic) 1-866-423-3836 extension 333 or (International) 1-(408)-395-4455 extension 333. Alternatively you may contact our support group at help@anchorbaytech.com.

Download from Www.Somanuals.com. All Manuals Search And Download.

0.2 Document Scope and Limitations

This document will cover the necessary information required to construct and transmit a serial (RS-232) or Infrared (IR) control signal to a DVDO iScan VP$_{xx}$ model video processor. These two basic mediums of control, are intended to convey the intentions of the user or automation system into the processes that operate the iScan. This document will cover the naming conventions, syntax, electrical specifications, and some troubleshooting that may be required for implementation in an installed system.

This document will *NOT* cover specific automation systems such as Crestron, AMX, Control4, Vantage, Elan, Universal Remote, RTI, Xantec, Niles, Russound, etc., or any programming within these systems. Correct selection of the automation system is the responsibility of the installer, and we do not offer troubleshooting for these systems beyond verification of the correct function of our iScan unit, and protocol confirmation.

This means, if a device is able to communicate with an iScan using another software platform (i.e. our firmware update procedure), the unit is deemed to be working correctly and the problem exists beginning at the wiring and proceeding into the code within the automation platform. In this case, contacting the manufacturer of the automation system is required.

Anchor Bay recommends contacting the automation system manufacturer before conducting the installation to see if they have a driver or control module pre-built for our products. If not, asking them to start work on one will help you (as an installer or end-user) by having their Engineers develop a driver or module that is guaranteed to work with their hardware (the more requests they get, the higher a priority it will be for them). If they do not have a complete library, they may have many of our control codes already in their database. Having this information on-hand will greatly ease the installation of our products. If they have any questions, please refer them to our support line, we will be glad to work with them.

0.3 Document Conventions

0.3.1 Model Compatibility

This document is intended to cover the iScan VP20, iScan VP20 with ABT102 daughter card, iScan VP30, iScan VP30 with ABT102 daughter card, the iScan VP50, and the iScan VP50^{PRO}. This document does not cover the iScan Ultra, iScan HD, or iScan HD+.

This document is intended to be used with the latest versions of software for each of the respective models – this is so that the most current features which have been released are listed, and to encourage our customers to use the latest features and bug-fixes that are available (we use the latest version to develop from – please do not report any bugs for old software). Please check our website (www.anchorbaytech.com) for the latest version of software for your product.

0.3.2   Product Introduction

This section is a brief introduction with pictures of each of the models of the iScan VP$_{xx}$ series – it is only intended as a brief "spotters guide" to iScan units. Please refer to your product's user's manual or our website for more in-depth product information at www.anchorbaytech.com/products/systems (replacement user's manuals may be obtained in PDF form at the same website by clicking on the "support" tab and selecting "documentation").

If you are trying to send a command to the iScan and it won't accept it – make sure you possess the model you think you have by using this spotter's guide, and then double-check in the command table in the following chapters, that the command is in fact supported for the model you are attempting to use.

0.3.3   VP20 (MM604)

iScan VP20 Front

iScan VP20 Back

This model is based on our iScan VP30 product, but has one less HDMI input and no analog RGBHV input or analog video out (RGBHV or Component). This device is commonly found in entry-level systems where input count is not as critical as getting the best possible processing with legacy source devices. This device may be further enabled with our ABT102 Deinterlacing add-on card for even better processing of interlaced SD content.

5

0.3.4 VP30 (MM603)

iScan VP30 Front

iScan VP30 Back

This model is our high-end entry-level product with the full four HDMI complement, the RGBHV/Component 3 input and Analog video output – with available options like an SD-SDI input and the ABT102 Deinterlacing add-on card for exceptional reproduction of interlaced SD content.

The VP30 also features more in-depth user controls and greater input flexibility, allowing it to be an excellent addition to a high-end home theater system, corporate media presentation system, or digital signage applications.

0.3.5 VP50 (MM605)

iScan VP50 Front

iScan VP50 Back

The iScan VP50, like the VP30, includes a wide selection of inputs and user controls, while further adding our Anchor Bay VRS processing for HD content (1080i Deinterlacing) and added Gamma adjustment controls.

6

0.3.6 VP50$^{\text{PRO}}$ (MM606)

iScan VP50$^{\text{PRO}}$ Front

iScan VP50$^{\text{PRO}}$ Back

The iScan VP50$^{\text{PRO}}$ is the first Video Processor to achieve the THX certification for Video Processors, setting the benchmark for video processing. This device is also the first HDMI 1.3 compatible video processor with the same outstanding Anchor Bay VRS HD and SD content processing algorithms of the preceding models, while adding even further configuration and calibration controls for ISF calibration and the new HD-SDI inputs (2x) and 12-volt triggers (2x) for driving external devices like anamorphic lenses and screen masking. This makes the iScan VP50$^{\text{PRO}}$ the ultimate in configurable and controllable high-end video processing – all of which can be harnessed through the same automation protocol we have had in the previous models. This makes it easy for systems integrators to upgrade from one iScan VP$_{xx}$ model to the newest to keep their customer's systems at the cutting edge.

7

0.4   How does automation work?

The iScan line of DVDO brand video processors are designed to enable control and flexibility over various input and output signal configurations – as well as our proprietary algorithms to improve several aspects of video quality and enable new capabilities that legacy devices by themselves are not able to achieve. This product has many features (covered in the User's Manual) which are intended to make day-to-day use of our video processing product easier in systems from "simple" up to "complex and fully integrated" home-theaters, or "corporate/industrial" applications. It is up to the user or system's integrator to "turn on" or otherwise set up the unit (and select appropriate auxiliary hardware) to enable this functionality within a given media presentation system. With the exception of some automatic functions which are user selectable (at the time of this writing: Input Selections, Deinterlacing Modes, and Output Profiles), the unit must be prompted by user action to do a specific function or provide a given signal path.

This user function can be initiated by an external device, like a Home Automation controller, Control Sequencer, or Learning/Macro-Infrared-Remote-Control. These execute the "user action" as part of a predefined "routine" or "script". Home Automation controllers, sequencers, or macro-remotes can control many devices at once, making a task like switching from one source device to another on three pieces of equipment occur with one user input action (this also reduces the amount of remote controls a given system has on a table). The iScan can accept either RS-232 based serial automation commands, or infrared remote control commands to enable very precise and "intelligent" control of the unit's behavior.

0.4.1   Interface Compatibility

Our devices have been designed to work with industry standardized control systems based on either "EIA232"-"RS-232C" asynchronous bidirectional serial character data transfers, or NEC or ABT-proprietary based Infra-Red (IR) one-way serial character data transfers operating at a 38.38kHz carrier frequency. The control sets for both methods are based on the same command IDs and control values for the sake of simplicity and ease of overall protocol mastery.

0.4.2   How is data encoded in digital form?

Digital electronics are very good with math and numbers – but they do not know how to "think" or talk in human-readable sentences. Because of this, programmers have created a "look-up-table" of standard characters which humans understand, and numerical equivalents for those characters which the device understands. There are several different ways to place characters in a table, and many different geographic locations which have special characters that need to be encoded. For the sake of standardization and compatibility, we have selected the UTF-8 standard which is backwards compatible with the ASCII standard of encoding characters to a numeric table (ASCII only uses 8-bit values between 0 and 127 - the specifics of these two standards are not covered, as numerous references for these are available at public libraries or the internet).

8

0.4.3 What is Binary?

The digital world is all ones and zeros. By placing ones and zeros in a standardized pattern we can encode data that can be exchanged between multiple devices. The lowest level of encoding data is "binary notation". In this notation, a "bit" represents the "true" or "false" presence of the numeric value at that bit location. Therefore, if the bit representing a "4" was "true", one would add the "4" to the total of the "byte" (the total size of the number). For our systems and the character-set we are using, we have an "8-bit" byte (meaning there are 8 value "places" representing numbers that are added to each other to generate the final number which the "byte" represents).

There are two ways to notate and send binary data – LSB and MSB. These stand for "**L**east **S**ignificant **B**it" and "**M**ost **S**ignificant **B**it" respectively, and these labels refer to which bit in a given byte is sent first (basically this means that data can be notated left-to-right or right-to-left – and the data can be sent with the largest value first, or the smallest value first). In this document, we will use the standard of notating MSB 8-bit bytes for sentence (string) construction (largest-to-smallest, left-to-right), and LSB for the communication scheme (RS-232/IR standards).

As an example, the decimal ("0-9", "10-19", etc.) notation number of "65" is:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1
0	**1**	*0*	*0*	*0*	*0*	*0*	**1**

If you add: 64 + 1, you get "65". This is the basis for all future dialog within this guide.

0.4.4 What is HEX?

So you're probably saying "It's going to take me forever to figure out how to send Binary data from a PC to an iScan," or "Boy, do I have to learn binary notation to use the iScan Automation Protocol?" Well the short answer is "no", you will want this basic ground-work to understand that electronic devices communicate this way – but there is a short-hand for Binary which you *will* need to learn. It reduces the characters you have to type by ¼ (thus you would type only two characters instead of eight to represent an "8-bit byte"). This is the *HEX Notation*. HEX is a different "base" number set – where "binary" has two possibilities for each character (0 and 1), the very familiar "Decimal" has ten possibilities (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), and "Hexadecimal" (or "HEX" for short) has 16 possibilities (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A(10), B(11), C(12), D(13), E(14), F(15)). This "shorthand" was selected since decimal doesn't easily calculate into binary (where each additional bit is a multiplier of two of the previous bit). With HEX, each character represents a "nybble" of a byte (or four bits). Each "byte" is split into two "nybbles" (a high nybble and a low nybble), so that a byte can be conveyed using the same MSB notation with fewer characters to mean the same thing, in a terminal application which accepts HEX.

9

As an example, using "65" again – the HEX equivalent is "41h". So what's the "h" at the end? There are two commonly accepted ways to identify HEX notation in a sentence (or "string"). One is with the use of a "0x00" notation, where the two last zeros represent the two HEX characters, or with "00h" showing that this is a two nybble-byte in HEX notation. This can get confusing the more you learn – so take a moment to highlight this section or put a Post-It flag on this page for future reference.

0.4.5 What is ASCII?

Okay, we started this digital primer with the idea that we wanted to send our data from one place to another in a way that the machines could understand. But now what happens if we (humans) want to read it? Well back to the ASCII Look-Up-Table concept that we brushed on at the beginning. If you remember, we replaced a human-readable character with a number so that the machine can understand it. We use the reverse of that table replacement to "extract" the data that was transferred from one device to another.

Recall that binary, decimal, and HEX - all represent different ways to write **_numbers_**. ASCII characters represent the **_Human-readable equivalent_** of that given number. For example, again using decimal number "65" (binary number: "00100001", HEX number "0x41" or "41h") – the ASCII table equivalent is a capital "**A**". All four of these numbers mean exactly the same thing to a machine using an ASCII table – capital "**A**".

A simple ASCII to HEX conversion table is provided at the end of this document in Appendix A.

0.5   A brief dialog about remote controlling a VPxx series video processor

Please be honest with yourself and ensure that you have understood the previous sections. If you're not confident about how binary = HEX = decimal and relates to ASCII, then you may want to check out the internet for more information on digital information technology – or contact our Technical Support Hotline at (U.S. Domestic) 1-866-423-3836 extension 333, or (International) 1-408-395-4455 extension 333. Alternatively you may contact our support by email group at help@anchorbaytech.com.

- The first thing this writer suggests when learning the following automation protocol – is to realize that this is a machine talking to another machine – not a human talking to another human. The automation protocol is written for maximum efficiency, clarity, and robustness of communication between two machines - all while allowing for future expansion without requiring us to re-write the protocol every time new features/products come out (thus commands that work in the new version of software should work in just about every other previous version/product – which has the exact same functional control).

- The second thing this writer suggests is learning and understanding the HEX notation – and how to convert decimal numbers and basic ASCII characters (0-9 and A-Z capitals) into HEX notation. The serial interface works in bytes, and understands numbers – so the closer you can get to understanding this type of communication – the easier this will be for you.

11

0.6   A dialog about input video memories

Due to the number of inputs and different types of input formats and ever further numerous types of source devices, we at Anchor Bay added input memories, which allow the user/system-integrator to configure very specific "effects" for a specific input format on a specific input connection. This means that a single input can have many different settings within the same control – just based on the input format that it is receiving.

As an example, at the time of this writing, for HDMI on the VP50^{PRO} we support:

VGA	720p-50Hz
SVGA	720p-60Hz
XGA	1080i-50Hz
SXGA	1080i-60Hz
576i-50Hz	1080p-23.98/24Hz
576p-50Hz	1080p-25Hz
480i-60Hz	1080p-50Hz
480p-60Hz	1080p-59.97/60Hz

Each format has its own memory, with individual picture controls, aspect ratios and zooms/pans, processing modes, etc. This can easily make the job of setting up an iScan very involved, as we offer an incredible amount of control over just about every aspect of the processed signal. We have put in functions to our automation protocol which allow an automation controller full access to these parameters – so care must be taken to avoid errors.

Keep in mind that not all inputs support all input types – for example, Composite and S-Video inputs are limited to 480i-60Hz or 576i-50Hz based on the source and the region the iScan is used in.

This page intentionally left blank

1 RS-232 Control

1.1 The RS-232 Physical Connection

RS-232 connections come in several styles which are accepted in the consumer electronics industry. The most common is the 9-pin D-Subminiature connector found on the back of most computers, and is the one that we use on the iScan VP$_{xx}$ products.

The female serial port, found on the back panel of an iScan VP$_{xx}$ video processor.

In this interface, there are a few different signals which ***must*** be supported. These are (all pin numbers are for the iScan):

RX – Data Receive (pin 3)   **CTS – Clear To Send (pin 7)**
TX – Data Transmit (pin 2)   **GND – Signal Ground (pin 5)**
RTS – Request To Send (pin 8)

We do not use the "DSR – Data Set Ready", "DTR – Data Terminal Ready", "CD - Carrier Detect" or "RI – Ring Indicator" pins for the iScan VP$_{xx}$ series.

These signals are associated with specific pin numbers based on what type of device the serial port is attached to. There are two types of serial device **D**ata-**T**erminal-**E**quipment (**DTE**) and **D**ata-**C**ommunications-**E**quipment (**DCE**). A DTE is your computer or an automation system – basically a controlling device. A DCE is a modem, or in this case the iScan. Some manufacturers chose to wire their RS-232 port as a DTE, but we have elected to wire our unit as a DCE. This determines a critical difference in the serial cable wiring to get the unit to communicate with the automation controller or PC. If your automation controller is based on a PC, the serial port is likely to be wired as a DTE port (please check with your automation controller vendor for clarification). This allows the use of a very common straight-through "extension" cable to be used to complete the communication connection (like the type which is shipped with the iScan unit).

When a dissimilar port type is used in a serial connection (for example DTE-to-DCE or vise versa), a straight-through cable is usually all that is needed. However, when similar port types are used, a cross-over cable is required (for example DCE-to-DCE or DTE-to DTE). Please double check the type of connection that you are using before connecting the cable.

14

1.1.1 The Anchor Bay RS-232 Protocol

In this portion of the document, we will discus the three types of control communications that occur between the iScan and the controlling device.

1.1.2 A Dialog on Checksums

Checksums are a way for a receiving device to double check the communication that occurred between the transmitting device and the receiver. In most systems, Checksums are not needed – however some installations absolutely require them (for example: industrial control or corporate teleconference systems). If you don't already know what a checksum is - you probably will not need it for your application. The system will work fine in 99.999% of systems without the use of checksums. If you need to use a checksum due to customer/job requirements, the calculation and checking calculations are provided in the following sections.

1.2 Control Commands

The "Control Command" is probably why you are reading this document right now. This is a sequence of data which tells the iScan to do something. Until the controller or PC sends an instruction to do something, the iScan will happily do its primary job – processing video.

This writer believes that the easiest way to understand what is occurring is to think of a "serial command" as a public address announcement you might hear in an airport:

"May I have your attention please, John Doe, please pickup the white courtesy phone and press 0. Thank you."

Essentially the same thing is done with an automation control sentence (or string):

"*Attention* this *is a command* which is *this long* and the command *controls this function >>pause<<* this is *the value* I want to set *>>pause<<* [*checksum* – optional] *I'm done talking*"

Hopefully this looks easy. However please remember, electronics don't speak in fancy human readable sentences, they speak in numbers. This is where human-readable ASCII character look-up-tables and HEX notation come into play, and a lot of confusion can too. Now in the ASCII table there are some basic "characters" which represent some of the bold words above:

"**Attention**" = Start Text or **STX** in ASCII
>>**pause**<< = Null or **NUL** in ASCII
"**I'm Done Talking**" = End Text or **ETX** in ASCII

Every ASCII character is a single "byte" (one 8-bit number each) which has been specified to mean what is shown above. Now remember that the ASCII table is meant to convert numbers to human readable characters and vise versa.

15

Also, each of the above "characters" has a related HEX notation number to go with it:

"Attention" = **Start Text** or **STX** = 0x*02* in HEX notation
>>pause<< = **Null** or **NUL** = 0x*00* in HEX notation
"I'm Done Talking" = **End Text** or ETX = 0x*03* in HEX notation

   It is up to the individual programmer to determine which method is easiest to understand – but if you haven't chosen your programming style yet, this writer recommends sticking with HEX notation. One thing that should be avoided at all costs is mixing HEX notation with ASCII characters – as you may see in the next set of examples, mixing numbers and ASCII will get you very confused very fast (You're not a computer, so you can't be expected to keep track of it all). This document will be written from here to the end slanted to illustrate HEX notation, as it demands the use of "bytes" and is easiest for new-comers to get used to recognizing characters which need to be converted from human readable text *characters* to machine readable *numbers*.

Let's take another look at that sentence:

   "***Attention*** this ***is a command*** which is ***this long*** and the command ***controls this function*** *>>pause<<* this is ***the value*** I want to set *>>pause<<* [*checksum* – optional] ***I'm done talking***"

Now let's replace the words we know with the HEX notation equivalents:

   "0x**02** this ***is a command*** which is ***this long*** and the command ***controls this function*** 0x**00** this is ***the value*** I want to set 0x**00** [*checksum* – optional] 0x**03**"

   We at Anchor Bay have specified the byte value for the "***is a command***" text's replacement as a portion of our protocol specification. We have defined a <u>command</u> as two ASCII characters of "3" and "0". In HEX notation this comes out to two bytes: 0x**33** and 0x**30** (these must be in this order!). Note that the "***is a command***" is represented by these two bytes (each 8-bits, or two nybbles).

Let's look at the sentence again, replacing what we know:

   "0x**02** 0x**33** 0x**30** which is ***this long*** and the command ***controls this function*** 0x**00** this is ***the value*** I want to set 0x**00** [*checksum* – optional] 0x**03**"

This gives us enough to have a "wrapper" for all RS-232 control commands:

0x**02** 0x**33** 0x**30** [**length byte 1**] [**length byte 2**] [**Command ID byte 1**] [**Command ID byte 2**] 0x**00** [**Value x-Bytes**] 0x**00** [**checksum** – *optional*] 0x**03**

Before we start listing Command ID bytes, lets look at the "*this long*" portion of our sentence. For this, <u>count</u> the two command ID bytes (count the bytes, don't add the values!), add the <u>count</u> of the two NUL bytes (again, don't add the values), add the <u>count</u> of the value bytes (this really should sink in now - <u>don't add the values themselves</u>). This equals the "byte-count" for the command sentence (string) – we are always counting bytes. Below is an example of the bytes we want to count:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Command ID 1	Command ID 2	NUL	Value Byte n	NUL

> *HELPER-RULE: There will always be two command ID bytes and two NUL bytes – and there should always be at least one value byte for a command. This means that you should never have a byte count below "5" for a command. You must also always use two bytes to convey the byte-count value; so an example would be "05" or 0x30 0x35.*

For now let's look at the most simple control of the iScan product – turning its power "on". The Command ID for the power control ("***controls this function***") is "**A**" and "**1**" – hey, if you were reading this from the beginning you'll recognize capital "A" as HEX 0x**41**. The people who wrote the ASCII Look-Up-Tables were nice enough to realize that humans would occasionally use the table – so they lined up decimal numbers to the 0x**30** HEX range (i.e. **0**=0x**30**, **1**=0x**31**, **2**=0x**32**, etc.). This means that the "1" we need is 0x**31**.

So the command ID bytes for the power control are (in HEX) 0x**41** 0x**31**.

Let's look at the sentence again, replacing what we know now:
"0x**02** 0x**33** 0x**30** which is *this long* 0x**41** 0x**31** 0x**00** this is *the value* I want to set 0x**00** [*checksum* – optional] 0x**03**"

Now let's look at the value we want to set this to – in the table in **Section 3** you will see the commands and the values that are possible. Looking up Power, we see that the values for OFF and ON are "0" and "1" respectively. We already know how to convert the "1" to HEX notation and since we do want to turn the unit "on", this is the value we're going to use. "***The value***" = 0x**31**.

Let's look at the sentence again, replacing what we know:
"0x**02** 0x**33** 0x**30** which is *this long* 0x**41** 0x**31** 0x**00** 0x**31** 0x**00** [*checksum* – optional] 0x**03**"

If you've read this far and understand what's happening - Great! Now the only things we are missing are the Checksum and the length-count bytes. Since the checksum must be the last thing we calculate, we'll do the length first: Two bytes for command ID + one byte for NUL + one byte for value + one byte for NUL = 5 bytes or "05". Converting the count to HEX notation we get 0x**30** and 0x**35**.

17

Let's look at the sentence again, replacing what we know now:
   "0x**02** 0x**33** 0x**30** 0x**30** 0x**35** 0x**41** 0x**31** 0x**00** 0x**31** 0x**00** [*checksum* – optional] 0x**03**"

   If you recall, unless your application calls for it specifically – ***YOU DO NOT NEED A CHECKSUM!!!*** If your application doesn't need it, you are done with the sentence construction (just remove the optional placeholder for the "checksum - optional"):

Let's look at the sentence again, with out the optional checksum placeholder:
            "0x**02** 0x**33** 0x**30** 0x**30** 0x**35** 0x**41** 0x**31** 0x**00** 0x**31** 0x**00** 0x**03**"

   Now there is one more detail which ***you*** will need to figure out about your automation system: "How or does it accept HEX notation?" Some systems are smart enough to recognize the "0x" as a prefix for a HEX notation number. Others are not. This writer is aware of an example application called "RS232 Hex Com Tool" which does not recognize the "0x" as a prefix. This means that the ***operator/user/programmer must determine how to enter the data correctly*** – due to the broad spectrum of programming styles across all of the varied automation systems this is not covered in this guide nor is it the responsibility of Anchor Bay to tell you (the reader). Contact your automation system vendor for clarification on data entry to their system.

   As it happens, in the above examples, the byte itself was highlighted with **BOLD** typeface to bring attention to the actual value for the byte. This highlighted data is also what that particular application expects, with a [space] or [comma] to separate the bytes. Thus the same "power-on" command would be:

            02 33 30 30 35 41 31 00 31 00 03 *for "power-on" with no checksum*

   If you are unsure if the automation computer or other machine is working with the serial cable, the "RS232 Hex Com Tool" program is available for download (shareware – free trial for 30 days, purchase for a small fee) on the web at: http://www.rs232pro.com/. Anchor Bay does not warrant the function of this utility or endorse its purchase – this is simply a reference to one of many options available for testing. The open-source Tera Term Pro utility used for upgrading iScan VP$_{xx}$ products is also capable of sending HEX or ASCII strings with some minor programming – but we do not support this use of the program and attempts to use Tera Term Pro as an automation controller should only be taken on by experienced programmers with some basic coding/programming background.

The checksum. This is the last part other than the Command ID Table and Value Table you might need to create a command string. Again, unless your customer/job requirements demand/specify it – **_YOU DO NOT NEED A CHECKSUM!!_** Assuming that you absolutely need to have a checksum due to a customer/job requirement, the checksum is fairly easy - add the *value* of every byte from the beginning of the string (at STX) to the last "NUL" just before the ETX (0x**03**). For the "Power On" command, this would be: **02 33 30 30 35 41 31 00 31 00**

So you would add: 0x02 + 0x33 + 0x30 + 0x30 + 0x35 + 0x41 + 0x31 + 0x00 + 0x31 +0x00 = 0x**16D**

HINT: You can use the scientific calculator in Windows to figure this out in HEX.

Now we only deal with 8-bit values for bytes – and you can see (if you recall the discussion about nybbles and bytes) that the checksum value is three hex characters or three "nybbles". This means the result is a 12-bit value. How we take care of this is very easy – drop (truncate) the nybbles above the two lowest nybbles. If you do this to the 0x**16D** value you get 0x**6D**. If you are writing a software program – an easy way to do this is to "**AND**" the checksum value with 0x**FF** in HEX or "**255"** in decimal.

If you've really been paying attention you'll remember that the checksum is two bytes – we made it easy to figure out these two by simply taking the **6** and the **D** (which are part of a HEX notation number from our calculation) and using them as ASCII stand-ins. So assume these two characters are ASCII and convert them down to HEX ("6" becomes 0x**36**, "D" becomes 0x**44**). This is a form of data expansion – and is intended to reduce the possible valid bit patterns which can be expected at these two byte locations to 16 possibilities.

For a last look at turning on the power for the iScan, let's look at the whole string including the checksum (underlined):

<p align="center">0x02 0x33 0x30 0x30 0x35 0x41 0x31 0x00 0x31 0x00 <u>0x36 0x44</u> 0x03</p>

That is all there is to Command Packets. If you are still unclear on how this is supposed to work, or you believe you are doing this correctly, but still have no success controlling the iScan, please contact our Technical Support group.

1.2.1 Example RS-232 Command Packets

This section contains the most commonly requested automation command-type strings (no checksums are provided):

Power
On
0x**02** 0x**33** 0x**30** 0x**35** 0x**41** 0x**31** 0x**00** 0x**31** 0x**00** 0x**03**
Off
0x**02** 0x**33** 0x**30** 0x**35** 0x**41** 0x**31** 0x**00** 0x**30** 0x**00** 0x**03**

Input
Composite 1
0x**02** 0x**33** 0x**30** 0x**35** 0x**41** 0x**43** 0x**00** 0x**31** 0x**00** 0x**03**
Composite 1
0x**02** 0x**33** 0x**30** 0x**35** 0x**41** 0x**43** 0x**00** 0x**32** 0x**00** 0x**03**
S-Video 1
0x**02** 0x**33** 0x**30** 0x**35** 0x**41** 0x**43** 0x**00** 0x**33** 0x**00** 0x**03**
S-Video 2
0x**02** 0x**33** 0x**30** 0x**35** 0x**41** 0x**43** 0x**00** 0x**34** 0x**00** 0x**03**
Component 1
0x**02** 0x**33** 0x**30** 0x**35** 0x**41** 0x**43** 0x**00** 0x**35** 0x**00** 0x**03**
Component 2
0x**02** 0x**33** 0x**30** 0x**35** 0x**41** 0x**43** 0x**00** 0x**36** 0x**00** 0x**03**
Component 3/RGBHV
0x**02** 0x**33** 0x**30** 0x**35** 0x**41** 0x**43** 0x**00** 0x**37** 0x**00** 0x**03**
HDMI 1
0x**02** 0x**33** 0x**30** 0x**35** 0x**41** 0x**43** 0x**00** 0x**38** 0x**00** 0x**03**
HDMI 2
0x**02** 0x**33** 0x**30** 0x**35** 0x**41** 0x**43** 0x**00** 0x**39** 0x**00** 0x**03**
HDMI 3
0x**02** 0x**33** 0x**30** 0x**36** 0x**41** 0x**43** 0x**00** 0x**31** 0x**30** 0x**00** 0x**03**
HDMI 4
0x**02** 0x**33** 0x**30** 0x**36** 0x**41** 0x**43** 0x**00** 0x**31** 0x**31** 0x**00** 0x**03**
SDI 1
0x**02** 0x**33** 0x**30** 0x**36** 0x**41** 0x**43** 0x**00** 0x**31** 0x**32** 0x**00** 0x**03**
SDI 2
0x**02** 0x**33** 0x**30** 0x**36** 0x**41** 0x**43** 0x**00** 0x**31** 0x**34** 0x**00** 0x**03**
AUTO Input Select
0x**02** 0x**33** 0x**30** 0x**36** 0x**41** 0x**43** 0x**00** 0x**31** 0x**33** 0x**00** 0x**03**

Input Preset (*recall – not save*)
4x3 Full Frame
0x**02** 0x**33** 0x**30** 0x**35** 0x**43** 0x**31** 0x**00** 0x**31** 0x**00** 0x**03**
4x3 Letterbox
0x**02** 0x**33** 0x**30** 0x**35** 0x**43** 0x**31** 0x**00** 0x**32** 0x**00** 0x**03**
16x9 Full Frame
0x**02** 0x**33** 0x**30** 0x**35** 0x**43** 0x**31** 0x**00** 0x**33** 0x**00** 0x**03**
4x3 Stretch
0x**02** 0x**33** 0x**30** 0x**35** 0x**43** 0x**31** 0x**00** 0x**34** 0x**00** 0x**03**

Preset 1
0x**02** 0x**33** 0x**30** 0x**35** 0x**43** 0x**31** 0x**00** 0x**35** 0x**00** 0x**03**
Preset 2
0x**02** 0x**33** 0x**30** 0x**35** 0x**43** 0x**31** 0x**00** 0x**36** 0x**00** 0x**03**
Preset 3
0x**02** 0x**33** 0x**30** 0x**35** 0x**43** 0x**31** 0x**00** 0x**37** 0x**00** 0x**03**
Preset 4
0x**02** 0x**33** 0x**30** 0x**35** 0x**43** 0x**31** 0x**00** 0x**38** 0x**00** 0x**03**
Preset 5
0x**02** 0x**33** 0x**30** 0x**35** 0x**43** 0x**31** 0x**00** 0x**39** 0x**00** 0x**03**
Preset 6
0x**02** 0x**33** 0x**30** 0x**36** 0x**43** 0x**31** 0x**00** 0x**31** 0x**30** 0x**00** 0x**03**
Preset 7
0x**02** 0x**33** 0x**30** 0x**36** 0x**43** 0x**31** 0x**00** 0x**31** 0x**31** 0x**00** 0x**03**
Preset 8
0x**02** 0x**33** 0x**30** 0x**36** 0x**43** 0x**31** 0x**00** 0x**31** 0x**32** 0x**00** 0x**03**
Preset 9
0x**02** 0x**33** 0x**30** 0x**36** 0x**43** 0x**31** 0x**00** 0x**31** 0x**33** 0x**00** 0x**03**
Preset 10
0x**02** 0x**33** 0x**30** 0x**36** 0x**43** 0x**31** 0x**00** 0x**31** 0x**34** 0x**00** 0x**03**

Deinterlacing Mode
Auto
0x**02** 0x**33** 0x**30** 0x**35** 0x**34** 0x**39** 0x**00** 0x**36** 0x**00** 0x**03**
Film
0x**02** 0x**33** 0x**30** 0x**35** 0x**34** 0x**39** 0x**00** 0x**30** 0x**00** 0x**03**
Video
0x**02** 0x**33** 0x**30** 0x**35** 0x**34** 0x**39** 0x**00** 0x**31** 0x**00** 0x**03**
Forced 3:2
0x**02** 0x**33** 0x**30** 0x**35** 0x**34** 0x**39** 0x**00** 0x**38** 0x**00** 0x**03**
Forded 2:2
0x**02** 0x**33** 0x**30** 0x**36** 0x**34** 0x**39** 0x**00** 0x**31** 0x**30** 0x**00** 0x**03**
2:2 Odd
0x**02** 0x**33** 0x**30** 0x**35** 0x**34** 0x**39** 0x**00** 0x**33** 0x**00** 0x**03**
2:2 Even
0x**02** 0x**33** 0x**30** 0x**35** 0x**34** 0x**39** 0x**00** 0x**32** 0x**00** 0x**03**
Game Mode 1
0x**02** 0x**33** 0x**30** 0x**35** 0x**34** 0x**39** 0x**00** 0x**34** 0x**00** 0x**03**
Game Mode 2
0x**02** 0x**33** 0x**30** 0x**35** 0x**34** 0x**39** 0x**00** 0x**35** 0x**00** 0x**03**

Mosquito Noise Reduction
Off
0x**02** 0x**33** 0x**30** 0x**35** 0x**43** 0x**41** 0x**00** 0x**30** 0x**00** 0x**03**
Level 1
0x**02** 0x**33** 0x**30** 0x**35** 0x**43** 0x**41** 0x**00** 0x**31** 0x**00** 0x**03**
Level 2
0x**02** 0x**33** 0x**30** 0x**35** 0x**43** 0x**41** 0x**00** 0x**32** 0x**00** 0x**03**
Level 3
0x**02** 0x**33** 0x**30** 0x**35** 0x**43** 0x**41** 0x**00** 0x**33** 0x**00** 0x**03**

Output Display Profile (recall – not save)
Display Profile 1
0x**02** 0x**33** 0x**30** 0x**35** 0x**45** 0x**30** 0x**00** 0x**31** 0x**00** 0x**03**
Display Profile 2
0x**02** 0x**33** 0x**30** 0x**35** 0x**45** 0x**30** 0x**00** 0x**32** 0x**00** 0x**03**
Display Profile 3
0x**02** 0x**33** 0x**30** 0x**35** 0x**45** 0x**30** 0x**00** 0x**33** 0x**00** 0x**03**
Display Profile 4
0x**02** 0x**33** 0x**30** 0x**35** 0x**45** 0x**30** 0x**00** 0x**34** 0x**00** 0x**03**
Display Profile 5
0x**02** 0x**33** 0x**30** 0x**35** 0x**45** 0x**30** 0x**00** 0x**35** 0x**00** 0x**03**
Display Profile 6
0x**02** 0x**33** 0x**30** 0x**35** 0x**45** 0x**30** 0x**00** 0x**36** 0x**00** 0x**03**
Display Profile 7
0x**02** 0x**33** 0x**30** 0x**35** 0x**45** 0x**30** 0x**00** 0x**37** 0x**00** 0x**03**
Display Profile 8
0x**02** 0x**33** 0x**30** 0x**35** 0x**45** 0x**30** 0x**00** 0x**38** 0x**00** 0x**03**
Display Profile 9
0x**02** 0x**33** 0x**30** 0x**35** 0x**45** 0x**30** 0x**00** 0x**39** 0x**00** 0x**03**
Display Profile 10
0x**02** 0x**33** 0x**30** 0x**36** 0x**45** 0x**30** 0x**00** 0x**31** 0x**30** 0x**00** 0x**03**

1.3 Query Commands

Query commands allow an external device to determine the setting of a given control. Building on the information presented in the previous section on constructing Command Packets, we will discuss the method for building a Query Packet. We'll use the example of querying the "power" state of the unit (Command ID **A1** used in the previous section).

Again, using the example of a sentence, the dialog would be:
"***Attention*** *this* ***is a query*** which is ***this long*** and I want to know the setting of ***this function*** *>>pause<<* [***checksum*** – optional] ***I'm done talking***"

There are some fairly obvious similarities which can be seen between Commands and Queries. These values are still the same for queries as they are based on ASCII look-up equivalents:

"Attention" = **Start Text** or **STX** = 0x*02* in HEX notation
>>pause<< = **Null** or **NUL** = 0x*00* in HEX notation
"I'm Done Talking" = **End Text** or ETX = 0x*03* in HEX notation

If you recall that the "is a command" bytes are **3** and **0**, we have defined "is a query" bytes as **2** and **0** – or 0x**32** and 0x**30** in HEX notation. So, looking at the sentence and replacing what we know, we would get:
"*0x02 0x32 0x30* which is ***this long*** and I want to know the setting of ***this function*** *0x00* [***checksum*** – optional] *0x03*"

To query "power", we use the same command ID found in Section 3, which we used to set the state – "A" and "1", or 0x**41** and 0x**31**. Again looking at the sentence and replacing what we know, we would get:
"*0x02 0x32 0x30* which is ***this long*** *0x41 0x31 0x00* [***checksum*** – optional] *0x03*"

We would then calculate the length (which in this type of packet is always "3" or 0x30 and 0x33 since it must be two bytes). With this value, and if you do not need a checksum the final packet would be:
"*0x02 0x32 0x30 0x30 0x33 0x41 0x31 0x00 0x03*"

To calculate the checksum, we take all of the values and add them up, then truncate the result to an 8-bit number (two nybbles):
0x02 + 0x32 + 0x30 + 0x30 + 0x33 + 0x41 + 0x31 + 0x00 = 0x139
If you truncate the result by only keeping the right most two "nybbles" and you should get 0x39 – remember these two nybbles are then assumed to be ASCII stand-in which must be converted to HEX notation (for data expansion). The two bytes for the checksum would be 0x33 and 0x39 so that your final query packet with checksum would be:

"*0x02 0x32 0x30 0x30 0x33 0x41 0x31 0x00 0x33 0x39 0x03*"

23

1.4 Responses

Responses (aka "feedback") is arguably what really make RS-232 a powerful interface. As opposed to infrared control, the RS-232 port allows for bi-directional communication, so that the controlling device can get information from the controlled unit to make decisions based on the actual state of the unit. Response packets are about the same as Command or Query packets – with some minor differences is the data they contain.

First, there are only three types of response the iScan can give (and remember that the iScan will not just start transmitting data without first being "asked" to do something).

- Acknowledge – This means the control you just sent was accepted and valid
- Query Response – the value for the control you would have just asked about
- Error Response – Something went wrong, this packet you what

Just like "Commands" and "Queries" have two bytes signifying those communications, these response packets each have their own:

- Command = "3" and "0" or 0x**33** and 0x**30** in HEX notation
- Query = "2" and "0" or 0x**32** and 0x**30** in HEX notation
- Acknowledge Response = "0" and "1" or 0x**30** and 0x**31** in HEX notation
- Query Response = "2" and "1" or 0x**32** and 0x**31** in HEX notation
- Error Response = "0" and "2" or 0x**30** and 0x**32** in HEX notation

For acknowledge, you will only ever see one packet:
"***Attention*** this ***is an acknowledge*** which ***is this long*** the data was ***accepted >>pause<<*** the packet ***was a command >>pause<<*** [***checksum*** – always included in replies] ***I'm done talking***"

We won't spend a great deal of time on the "acknowledge" except replacing the known items above with the values:
"0x**02** 0x**30** 0x**31** 0x**30** 0x**35** 0x**31** 0x**00** 0x**33** 0x**30** 0x**00** 0x**35** 0x**43** 0x**03**"

Since you will only ever get an "Acknowledge" packet for a "command", this is the only variant you should ever expect (using these exact above values). However should something go wrong, you will get an error reply:

"***Attention*** this ***is an Error*** which ***is this long*** this is ***the Error*** >>***pause***<< [***checksum*** – always included in replies] ***I'm done talking***"

Replacing the items which should be starting to get familiar, we get:
"0x**02** 0x**30** 0x**32** [count byte 1 0x**30**] [count byte 2 – either 0x**32** or 0x**33**] [error byte n (there may be up to two bytes based on the error)] 0x**00** [checksum – always included in replies] 0x**03**"

The values you may get in an error reply are on the next page.

Error "1" –   Invalid checksum. This error means either the checksum you sent was wrong or the transmission was bad due to interference (double check your checksum calculation or your serial link).

Error "2" –   Invalid Incoming Packet ID (i.e. Command = "3"&"0", Query = "2"&"0", others are invalid when **_sent_** to the iScan)

Error "3" –   Invalid Setting (i.e. Power = "A"&"1") if you get this error, make sure that the command is supported by the model you are using.

Error "4" –   Range Error (i.e. Power on = "1", power off = "0") if you get this error you tried to set a value to the control which is either out of range or not supported.

Error "5" –   Bad Packet Character (i.e. STX, ETX, NUL) a valid ASCII character value may have been used in the wrong place – double check your syntax. Otherwise, ensure that only numbers, or punctuation (".", "+", or "-") was used.

Error "6" –   Last byte of packet was not received within 100 milliseconds – if this happens, first make sure that the link is good. Then, ensure that your control device is waiting for a complete response packet before sending another packet. If your controller does not "listen" to the flow control pins (DSR/DTR look at section 2.1) the buffer may over-flow causing bytes to be lost. If no RS-232 return path is being used, pace your commands to about 10 commands every second.

Error "7" –   Unterminated Data Value. This means you missed a "NUL" after a value and went straight to the "ETX" – check your syntax.

Error "8" –   Bad Data – If you get this response, first check your serial link, then check the table in Section 3 to ensure you sent the right type of value. If you send a "5E" for a control expecting a number like "1.453", you will get this type of error response.

Error "9" –   Too many or too few data characters. This error appears if your packet has the wrong byte counts value, or you don't have all of the data in the string.

Error "10" – The setting is not writable (i.e. command for "Device Name"), this will be your response if you attempt to write to a query only Command ID

Error "11" – The packet is larger than the maximum packet size. You should never see this error – we do not have any controls which are at the time of this writing even close to the maximum size. If this error comes back – check your serial link and syntax. If you are transmitting more that 50 bytes in a single command you are probably doing something wrong!!

The query response is the most involved response packet you will get in reply. This packet can have *any* data in the "value" bytes (although it will still be ASCII characters in HEX notation). Note that commands like "Model Name" will reply with text, while commands which are controlled by numbers will reply with numbers.

The response to a query for power state (if the unit is "on") would be:
0x**02** 0x**32** 0x**31** 0x**30** 0x**35** 0x**41** 0x**31** 0x**00** 0x**31** 0x**00** 0x**36** 0x**44** 0x**03**

Working in reverse of building a packet (assuming you read the previous sections), you should be starting to see patterns:

0x**02** = STX, 0x**32** = "2", 0x**31** = "1", 0x**30** = "0", 0x**35** = "5", 0x**41** = "A", 0x**31** = "1", 0x**00** = NUL, 0x**31** = "1", 0x**00** = NUL, 0x**36** = "6", 0x**44** = "D", 0x**03** = ETX

From this you can see:
- The STX which means "Attention".
- The "2" and "1" which identifies the packet as a query response type.
- A "0" and "5" which shows that the byte count is 5 bytes long.
- An "A" "1" for the command ID which decodes to "Power" in Section 3
- A NUL before the value of the command
- A "1" showing the state to be "On" as decoded in Section 3
- A NUL after the value of the command
- A checksum of "6D" which if we check the math;
  0x02 + 0x32 + 0x31 + 0x30 + 0x35 + 0x41 + 0x31 + 0x00 + 0x31 + 0x00 = 16D
  and if we truncate the value to only two "nybbles" (or two hex characters) we get 6D which matches the checksum value – showing the checksum and packet is good.
- An ETX which means "I'm done talking"

26

2 IR Control

We have provided a reprint of Barry Gordon's paper on IR interfacing in Appendix B at the end of this document. If you feel the information provided in the next few sections is a bit confusing, please take a moment to read that document.

2.1 The NEC IR Protocol (Factory Remote)

In this section, we will provide the basic values needed for building a Pronto HEX compatible iScan factory remote control code. The factory remote strictly adheres to the NEC IR protocol, while the discrete commands are often too long (more than one byte) or too numerous to fit within one "device code" under the NEC protocol. For discrete commands, please see the next section. Below we give you the basic items required to replicate the factory remote buttons. If you are not familiar with the NEC IR protocol, please take the time to read the article in Appendix B by Barry Gordon on Pronto HEX and NEC IR protocols. :

Carrier Frequency =   38.38kHz
Device ID Code =   0x2084
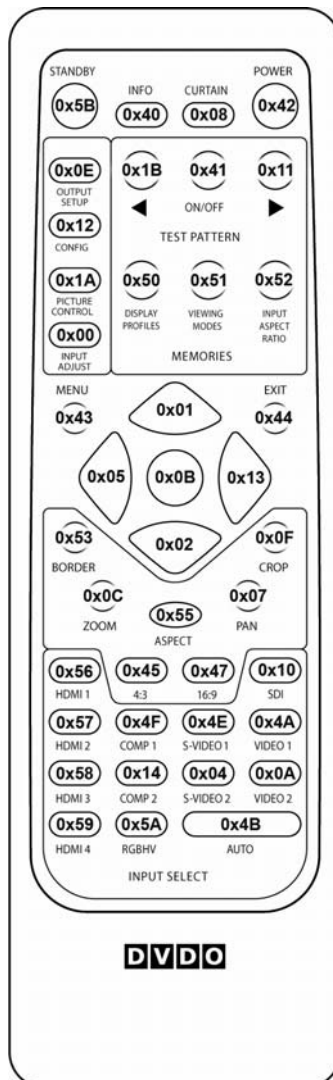
2.2 The Anchor Bay IR Protocol (Discrete Control)

As stated before, the discrete controls may be longer than the NEC protocol will allow. The NEC protocol only allows for one byte of "control/value" data to be transferred from the remote control to the controlled device. The Pronto HEX format does not have specific length limits, and since it is a common interchange format and is fairly easy to use – we have constructed a discrete control command system based on the RS-232 Command IDs and Values, which may be programmed into an advanced learning macro remote control which can interpret the Pronto HEX structure.

Note: Not all learning remotes are made equal! Some are limited to NEC compliant codes only and are <u>incompatible</u> with the discrete functions provided in this section. Check with your installer or remote reseller to find out if your learning remote is NEC compliant only. Also, your remote must be able to understand and transmit 38.38kHz IR signals – not every remote or IR repeater system can do this, again double check with your vendor.

The basic sequence for building a discrete code is very simple:

- Decide which Command you wish to control and find the "command ID" for that control
- Decide what you would like to set the control to and look-up (if necessary) the appropriate value for that setting on that control
- Build the IR command using the below methodology
- Test the IR command before leaving the job-site or publishing the codes publicly

The method for building a code is very similar to what you may have already read (in the previous sections) – if you have not read section 2.1, this is a good time to go back and read it before we *really* confuse you. Assuming that you have read and understood the 2.1 section information, here are some fixed values which you will want to know for discrete Anchor Bay commands (*Note: if you "learn" a discrete function from one remote to another you may get slightly different values – but these may not work reliably*):

Pronto HEX Carrier Frequency of 38.38kHz =	006C
Pronto HEX Start bit pulse width =	0064 0064
Pronto HEX Logic "1" bit pulse width =	0016 0041
Pronto HEX Logic "0" bit pulse width =	0016 0015
Pronto HEX Stop bit Pulse width =	0044 0044
End of defined command string "bit" =	0016 0001

The format for data which must be adhered to is:

0000	Always "zero" to mark the beginning of the code header
006C	Carrier Frequency of 38.38kHz
nnnn	Number of "bit bursts" in the transmission
0000	Always "zero" to mark the end of the code header
0064 0064	Start "bit" (beginning of command transmission)

nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn *8-bit Command ID*

nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn *8-bit Value byte 1*

……………

nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn *8-bit Value byte n*

nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn *8-bit Checksum*

0044 0044	Stop "bit" (end of command transmission)
0016 0001	End of defined command string "bit"

Remember that the bit order we discuss when readin the numbers is MSB – yet when it is written in Pronto HEX, because the IR transmitter sends out the LSB first, the bits must be flipped from left to right. Let's try this with a discrete power "off" command starting on the next page.

If you recall from the RS-232 section, the command ID for the power control is "A" and "1", but if you look at the Pronto HEX format, the command ID can only be one byte. This is why we made the Command IDs out of "HEX compatible" naming – so that the same data when represented as a byte will look very familiar. Thus the HEX byte for controlling power via IR is 0x**A1**.

If you were to open the calculator program in Windows, and enter this number by selection the "Hex" radio button in scientific view mode – then by clicking the "Bin" radio button the calculator will automatically convert it from HEX to binary for you:
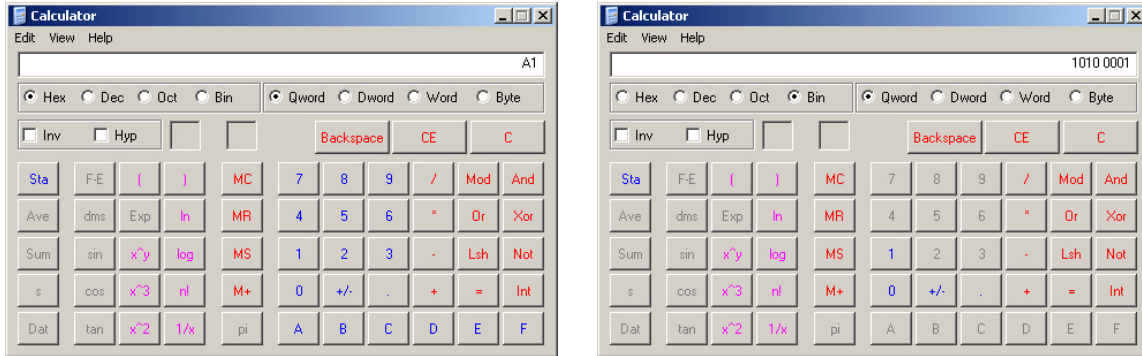
You get the result of: 1010 0001 as the binary number for A1.
Remember that you must fip the number left-to-right for IR to work correctly, so you would get: 1000 0101

Since we want to turn this control "off", we look up the value for off and find that it is "0" or 0x30 in HEX. Since this value is only a byte – it is the only "value" byte we need to transmit. By using the Windows calculator, we can convert this to binary: 0011 0000 – then flip it from MSB-to-LSB: 0000 1100

So we have two of the three parts needed to make a control command – the third and last byte to transmit in this case is the checksum. The checksum is easier in IR than it is in RS-232, one simply adds the command ID byte value and the setting value(s). For power off, this is 0xA1 + 0x30 = 0xD1. We can again use Calculator to convert the HEX value to binary: 1101 0001 then flip it from MSB-to-LSB: 1000 1011

Now we have the IR command in binary (we'll show the command parts for reference):

0000	Always "zero" to mark the beginning of the code header
006C	Carrier Frequency of 38.38kHz
nnnn	Number of "bit bursts" in the transmission
0000	Always "zero" to mark the end of the code header
0064 0064	Start "bit" (beginning of command transmission)
1000 0101	*8-bit Command ID of 0xA1*
0000 1100	*8-bit Value byte 1 of 0x30*
1000 1011	*8-bit Checksum of 0xD1*
0044 0044	Stop "bit" (end of command transmission)
0016 0001	End of defined command string "bit"

29

This writer finds this to be the best step to explain the "bit burst" count in the Pronto HEX format. The bit burst is every "bit" from the after last "0000" in the header to the last "bit" in the command defenition. The bits being referred to are in bold and larger above. There is one start bit, 8 command ID bits, 8 value bits, 8 checksum bits, one stop bit, and an end of string bit. The bit count is then: $1 + 8 + 8 + 8 + 1 + 1 = 27$. In the Pronto HEX format this number must be conveyed in a 16-bit HEX notation. You can again use the Windows calculator to convert this from decimal to HEX: 0x001B.

We're almost done with this except that in Pronto HEX, every bit must be represented by a "burst pair" (on-time and off-time). Since we've previously defined what these values are for Logic "1" and Logic "2", all you need to do now is replace "1"s and "0"s with the appropriate predefined burst pair.

Pronto HEX Logic "1" bit pulse width =   0016 0041
Pronto HEX Logic "0" bit pulse width =   0016 0015

1   0   0   0   0   1   0   1 *Command ID of 0xA1*
0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 *in* Pronto HEX
burst pairs

0   0   0   0   1   1   0   0 *Value byte 0x30*
0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 *in* Pronto HEX
burst pairs

1   0   0   0   1   0   1   1 *Checksum of 0xD1*
0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 *in* Pronto HEX
burst pairs

This makes the final command for discrete power off:

0000   *Always "zero" to mark the beginning of the code header*
006C   *Carrier Frequency of 38.38kHz*
001B   *Number of "bit bursts" in the transmission (in this case 27 "bits" or HEX 0x001B)*
0000   *Always "zero" to mark the end of the code header*
0064 0064   *Start "bit" (beginning of command transmission)*
0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 *Command ID of 0xA1*
0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 *Value byte 1 of 0x30*
0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 *Checksum of 0xD1*
0044 0044   *Stop "bit" (end of command transmission)*
0016 0001   *End of defined command string "bit"*

Remember that the Pronto HEX string must not have the above descriptive text, extra [ENTER]s, or other formatting characters. The final CCF command you should end up with is:

0000 006C 001B 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0044 0044 0016 0001

We have provided for your reference some of the most common IR commands we have had requested, beginning in the next section. Since we are constantly adding features and new automation commands – it will be faster if you learn how to construct these commands yourself, rahter than waiting for enough people to request codes to be built for you. Of course, if you are having trouble, with this and need a code for an installation imediately, please do not hesitate to call our Technical Support team.

30

2.2.1 Discrete IR Control Examples

Below is a partial list of commonly used discrete commands in Pronto HEX syntax format (Pronto HEX is a common Home-Automation Interchange format):

Power

On:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041
0044 0044 0016 0001

Off:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041
0044 0044 0016 0001

Inputs

Video 1:
0000 006d 0024 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0015 0015 0040 0016 0015 0015 0040 0016 0015 0015 0015 0016 0040 0015 0015
0016 0040 0015 0015 0016 0040 0015 0015 0016 0040 0015 0040 0016 0015 0015 0040 0016 06c0
0156 0055 0016 00ab

Video 2
0000 006d 0024 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0015 0015 0040 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015
0015 0040 0016 0015 0015 0040 0016 0015 0015 0040 0016 003f 0016 0040 0015 0040 0016 06c0
0156 0055 0016 00ab

S-Video 1:
0000 006d 0024 0000 0156 00aa 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0015 0015 0040 0016 003f 0016 0040 0015 0015 0016 0015 0016 003f 0016 0015
0016 003f 0016 0015 0016 0015 0015 0015 0016 0040 0015 0040 0016 0015 0015 0040 0016 06c0
0156 0055 0016 00aa

S-Video 2:
0000 006d 0024 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0015 0015 0015 0016 0040 0015 0015 0016 0015 0016 0015 0015 0015 0016 0015
0016 003f 0016 0040 0015 0015 0016 0040 0015 0040 0016 003f 0016 0040 0015 0040 0016 06c0 0156
0055 0016 00ab

Component 1:
0000 006d 0024 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 003f 0016 0040 0015 0040 0016 003f 0016 0015 0016 0015 0015 0040 0016 0015
0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0040 0016 0015 0015 0040 0016 06c0
0156 0055 0016 00ab

Component 2:
0000 006d 0022 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0015 0015 0015 0016 0040 0015 0015 0016 0040 0015 0015 0016 0015 0016 0015
0015 0040 0016 003f 0016 0015 0016 003f 0016 0015 0016 003f 0016 0040 0015 0040 0016 00ab

RGBHV/Component 3:
0000 006d 0046 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0015 0015 0040 0016 0015 0015 0040 0016 003f 0016 0015 0016 003f 0016 0015
0016 003f 0016 0015 0016 003f 0016 0015 0016 0015 0015 0040 0016 0015 0015 0040 0016 06c1 0156
0055 0016 0641 0156 00aa 0016 0015 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016
0015 0016 0015 0015 0040 0016 0015 0015 0040 0016 003f 0016 0015 0016 003f 0016 0015 0016
0015 0015 0015 0016 0040 0015 0015 0016 0040 0015 0040 0016 0015 0015 0040 0016 0015 0015
0040 0016 0015 0015 0040 0016 0015 0015 0015 0016 0040 0015 0015 0016 0040 0015 00ab

HDMI 1:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0015 0015 0041 0016 0040 0016 0015 0016 0040 0016 0015 0016 0040 0016 0015
0016 0040 0016 0015 0016 0015 0015 0041 0016 0015 0015 0041 0016 0015 0015 0040 0016 06cf
0156 0056 0016 00ac

HDMI 2:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0041 0015 0015
0016 0015 0016 0040 0016 0041 0015 0040 0016 0015 0015 0041 0016 0015 0015 0041 0016 0015
0015 0015 0016 0015 0016 0015 0015 0040 0016 0015 0015 0041 0016 0015 0015 0041 0016 06cf
0156 0056 0016 00ac

HDMI 3:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0041 0015 0015
0016 0015 0016 0015 0015 0015 0016 0015 0016 0040 0016 0040 0015 0015 0016 0041 0015 0015
0016 0041 0015 0041 0016 0040 0016 0015 0016 0015 0015 0041 0016 0015 0015 0041 0016 06cf
0156 0056 0016 00ac

HDMI 4:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0041 0015 0015
0016 0015 0016 0040 0016 0015 0016 0015 0015 0040 0016 0040 0016 0015 0016 0040 0016 0015
0016 0015 0015 0041 0016 0040 0016 0015 0016 0015 0015 0040 0016 0015 0015 0041 0016 06cf
0156 0056 0016 00ac

32

SDI:
0000 006d 0024 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0015 0015 0015 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 003f 0016 0040 0015 0040 0016 003f 0016 0015 0016 003f 0016 0040 0015 0040 0016 06c0 0156
0055 0016 00ab

Auto:
0000 006d 0024 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 003f 0016 0040 0015 0015 0016 0040 0015 0015 0016 0015 0016 003f 0016 0015
0016 0015 0015 0015 0016 0040 0015 0015 0016 0040 0015 0040 0016 0015 0015 0040 0016 06c0
0156 0055 0016 00ab

Navigational Keys

Menu:
0000 006d 0024 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 003f 0016 0040 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0015 0015 0040 0016 003f 0016 0040 0015 0040 0016 0015 0015 0040 0016 06c0
0156 0055 0016 00ab

Exit:
0000 006d 0024 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0015 0015 0015 0016 0040 0015 0015 0016 0015 0016 0015 0015 0040 0016 0015
0015 0040 0016 003f 0016 0015 0016 003f 0016 0040 0015 0040 0016 0015 0015 0040 0016 06c0 0156
0055 0016 00ab

Enter:
0000 006d 0024 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 003f 0016 0040 0015 0015 0016 0040 0015 0015 0016 0015 0016 0015 0015 0015
0016 0015 0016 0015 0015 0040 0016 0015 0015 0040 0016 003f 0016 0040 0015 0040 0016 06c0
0156 0055 0016 00ab

Up:
0000 006d 0024 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 003f 0016 0015 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 003f 0016 0040 0015 0040 0016 003f 0016 0040 0015 0040 0016 06c0 0156
0055 0016 00ab

Down:
0000 006d 0024 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0015
0016 003f 0016 0015 0016 003f 0016 0040 0015 0040 0016 003f 0016 0040 0015 0040 0016 06c0 0156
0055 0016 00ab

Left:
0000 006d 0024 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 003f 0016 0015 0016 003f 0016 0015 0016 0015 0015 0015 0016 0015 0016 0015
0015 0015 0016 0040 0015 0015 0016 0040 0015 0040 0016 003f 0016 0040 0015 0040 0016 06c0
0156 0055 0016 00ab

Right:
0000 006d 0024 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 003f 0016 0040 0015 0015 0016 0015 0016 003f 0016 0015 0016 0015 0015 0015
0016 0015 0016 0015 0015 0040 0016 003f 0016 0015 0016 003f 0016 0040 0015 0040 0016 06c0 0156
0055 0016 00ab

Test Patterns

On:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041
0044 0044 0016 0001

Off:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041
0044 0044 0016 0001

Test Pattern On/Off:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0041 0015 0015
0016 0015 0016 0015 0015 0041 0016 0015 0015 0015 0016 0040 0015 0015 0016 0041 0015 0015
0016 0041 0015 0015 0016 0041 0015 0040 0016 0015 0015 0040 0016 0015 0015 0041 0016 06cf
0156 0056 0016 00ac

Previous Test Pattern:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0040 0016 0041 0015 0015 0016 0041 0015 0041 0016 0015 0015 0015 0016 0015
0016 0015 0015 0015 0016 0041 0015 0015 0016 0015 0016 0040 0016 0040 0015 0040 0016 06cf
0156 0056 0016 00ac

Next Test Pattern:
0000 006c 0026 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0041 0015 0015
0016 0015 0016 0040 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015 0016 0015 0016 0015
0015 0015 0016 0041 0015 0041 0016 0040 0016 0015 0016 0040 0016 0041 0015 0041 0016 06cf
0156 0056 0016 0e60 0156 0056 0016 00ac

User Mode

Advanced:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015
0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0041 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015 0016 0041
0044 0044 0016 0001

Normal:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015
0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0041 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015 0016 0041
0044 0044 0016 0001

34

Cue

Off:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041
0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015 0016 0041 0016 0015
0044 0044 016 0001

On:
0000 006c 001b 0000 0064 00640016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041
0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015 0016 0041 0016 0015
0044 0044 0016 0001

VCR Mode

On:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

Off:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

Input Aspect Ratio - Frame:

4:3:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0041 0016 0041 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

16:9
0000 006c 001b 0000 0064 0064 0016 0015 0016 0041 0016 0041 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0044 0044 0016 0001

Input Aspect Ratio Presets

4:3 Full Frame:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0041 0016 0041 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015
0044 0044 0016 0001

Letterbox:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0041 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015
0044 0044 0016 0001

16:9 Full Frame:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0041 0016 0041 0016 0041 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015
0044 0044 0016 0001

IAR Preset 1:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0041 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015
0044 0044 0016 0001

IAR Preset 2:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0041 0016 0041 0016 0041 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0041 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015
0044 0044 0016 0001

IAR Preset 3:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0041 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0041 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015
0044 0044 0016 0001

IAR Preset 4:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 0016 0015
0044 0044 0016 0001

User:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0041 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 0016 0015
0044 0044 0016 0001

Film Mode

Off:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

Film Bias:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

Auto:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0041 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

Active Input AR

1.33:1
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015
0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0044 0044 0016 0001

36

1.55:1
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015
0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0044 0044 0016 0001

1.66:1
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015
0016 0041 0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0044 0044 0016 0001

1.78:1
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015
0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0044 0044 0016 0001

1.85:1
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015
0016 0041 0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0044 0044 0016 0001

2.35:1
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015
0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0044 0044 0016 0001

User:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015
0016 0041 0016 0015 0016 0041 0016 0041 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0041 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0044 0044 0016 0001

Audio Input

Off:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

Audio 1:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0041 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

Audio 2:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

Audio 3:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

Audio 4:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

Analog:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

HDMI:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0044 0044 0016 0001

Display Profiles

Profile 1:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0041 0016 0041 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015
0044 0044 0016 0001

Profile 2:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0041 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015
0044 0044 0016 0001

Profile 3:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0041 0016 0041 0016 0041 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015
0044 0044 0016 0001

Profile 4:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0041 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015
0044 0044 0016 0001

User:
0000 006c 001b 0000 0064 0064 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041
0016 0041 0016 0041 0016 0041 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015
0044 0044 0016 0001

Display Profile – Auto

Off:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0041 0016 0041 0016 0015 0016 0015 0016 0041
0016 0041 0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0041 0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015
0044 0044 0016 0001

On:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0041 0016 0041 0016 0015 0016 0015 0016 0041
0016 0041 0016 0041 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 0016 0015
0044 0044 0016 0001

Deinterlacing Modes

Auto:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

Film Bias Mode:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0015 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

Video Mode:
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

2:2 Even Mode
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0041 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

2:2 Odd Mode
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0041 0016 0041 0016 0015 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

Game Mode 1
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0015 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

Game Mode 2
0000 006c 001b 0000 0064 0064 0016 0041 0016 0015 0016 0015 0016 0041 0016 0015 0016 0015
0016 0041 0016 0015 0016 0041 0016 0015 0016 0041 0016 0015 0016 0041 0016 0041 0016 0015
0016 0015 0016 0015 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 0015
0044 0044 0016 0001

Buttons from Remote Control:

Information:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0041 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0041 0015 0015
0016 0015 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0015 0016 0040 0016 0015
0016 0040 0016 0041 0015 0041 0016 0040 0016 0040 0015 0041 0016 0015 0015 0041 0016 06cf
0156 0056 0016 00ac

Curtain:
0000 006d 0024 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0015 0015 0015 0016 0015 0016 003f 0016 0015 0016 0015 0015 0015 0016 0015
0016 003f 0016 0040 0015 0040 0016 0015 0015 0040 0016 003f 0016 0040 0015 0040 0016 06c1 0156
0055 0016 00ab

4:3
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0040 0016 0015 0016 0040 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0040 0016 0015 0016 0040 0016 0041 0015 0041 0016 0015 0015 0040 0016 06cf
0156 0056 0016 00ac

16:9
0000 006d 0026 0000 0156 00ab 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 003f 0016 0040 0015 0040 0016 0015 0015 0015 0016 0015 0016 003f 0016 0015
0016 0015 0015 0015 0016 0015 0016 003f 0016 0040 0015 0040 0016 0015 0015 0040 0016 06c1
0156 0055 0016 0e43 0156 0055 0016 00ab

Aspect:
0000 006c 0026 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0040 0016 0015 0016 0040 0016 0015 0016 0040 0016 0015 0016 0040 0016 0015
0016 0015 0015 0041 0016 0015 0015 0041 0016 0015 0015 0040 0016 0015 0015 0040 0016 06cf
0156 0056 0016 0e60 0156 0056 0016 00ac

Border:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0041 0015 0015
0016 0015 0016 0040 0016 0041 0015 0015 0016 0015 0016 0040 0016 0015 0016 0040 0016 0015
0016 0015 0015 0015 0016 0041 0015 0040 0016 0015 0015 0040 0016 0015 0015 0041 0016 06cf
0156 0056 0016 00ac

Crop:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0041 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0041 0015 0015
0016 0015 0016 0015 0015 0015 0016 0041 0015 0015 0016 0040 0015 0015 0016 0041 0015 0015
0016 0041 0015 0041 0016 0015 0015 0040 0016 0015 0015 0041 0016 0015 0015 0041 0016 06cf
0156 0056 0016 00ac

Zoom:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0041 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0041 0015 0015
0016 0015 0016 0015 0015 0015 0016 0041 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015
0015 0041 0016 0040 0016 0015 0016 0015 0015 0040 0016 0040 0016 0041 0015 0041 0016 06cf
0156 0056 0016 00ac

Pan:
0000 006c 0026 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0041 0015 0015
0016 0015 0016 0040 0016 0041 0015 0040 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015
0016 0015 0016 0015 0015 0015 0016 0041 0015 0041 0016 0040 0016 0041 0015 0041 0016 06cf
0156 0056 0016 0e60 0156 0056 0016 00ac

Display Profiles:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0015 0015 0015 0016 0015 0016 0015 0015 0041 0016 0015 0015 0041 0016 0015
0015 0040 0016 0040 0016 0041 0015 0041 0016 0015 0015 0041 0016 0015 0015 0040 0016 06cf
0156 0056 0016 00ac

40

Viewing Modes:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0041 0015 0015
0016 0015 0016 0040 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015 0016 0041 0015 0015
0016 0015 0016 0040 0016 0041 0015 0041 0016 0015 0015 0040 0016 0015 0015 0041 0016 06cf
0156 0056 0016 00ac

Output Setup:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0040 0015 0015
0016 0015 0016 0015 0015 0041 0016 0040 0016 0041 0015 0015 0016 0015 0016 0015 0015 0015
0016 0041 0015 0015 0016 0015 0016 0015 0015 0041 0016 0040 0016 0040 0015 0040 0016 06cf
0156 0056 0016 00ac

Configuration:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0041 0015 0015
0016 0015 0016 0015 0015 0041 0016 0015 0015 0015 0016 0040 0015 0015 0016 0015 0016 0015
0015 0041 0016 0015 0015 0041 0016 0040 0016 0015 0016 0040 0016 0041 0015 0041 0016 06cf
0156 0056 0016 00ac

Picture Control:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0041 0015 0015
0016 0015 0016 0015 0015 0041 0016 0015 0015 0040 0016 0040 0016 0015 0016 0015 0015 0015
0016 0041 0015 0015 0016 0041 0015 0015 0016 0015 0016 0040 0016 0041 0015 0041 0016 06cf
0156 0056 0016 00ac

Input Adjust:
0000 006c 0026 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0041 0015 0015
0016 0015 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015
0016 0041 0015 0041 0016 0040 0016 0041 0015 0040 0016 0040 0016 0041 0015 0041 0016 06cf
0156 0056 0016 0e60 0156 0056 0016 00ac

Input Aspect Ratio:
0000 006c 0024 0000 0156 00ac 0016 0015 0016 0015 0015 0040 0016 0015 0015 0015 0016 0015
0016 0015 0015 0041 0016 0015 0015 0015 0016 0015 0016 0015 0015 0015 0016 0041 0015 0015
0016 0015 0016 0015 0015 0041 0016 0015 0015 0015 0016 0040 0015 0015 0016 0041 0015 0015
0016 0041 0015 0015 0016 0041 0015 0040 0016 0015 0015 0040 0016 0015 0015 0041 0016 06cf
0156 0056 0016 00ac

3 Automation Command IDs and Values

This section beginning on the next page, contains the entire list of Control/Query commands available with the iScan VP*xx* line of video processors. The two character Command ID is in **bold-underline** (example: **A1** for "Power"). The possible values are given for each control in **bold** (example: **1.000**). We have presented the list in the same layout as the OSD starting on the next page, to allow for quick location of the control you are seeking.

Some commands are not supported with certain models, the models which are supported for a given command or value will be identified with an icon: VP20 VP30 VP50 VP50PRO or ALL . If no icon is shown for a value, the icon for the command is correct for the value as well.

There are some automation functions which are not available as a direct item in the OSD, these are:

Power – **A1** ALL
  Off - **0**
  On - **1**

Navigation Buttons – **A2** ALL
  Left - **1**
  Right - **2**
  Up - **3**
  Down - **4**
  Menu - **5**
  Enter - **6**
  Exit – **7**

Curtain – **A4** ALL
  Open - **0**
  Closed – **1**
Note: This control will not override HDCP blanking because of a link failure (indicated by flashing power LED).

Product Name – **A8** *(QUERY ONLY)* ALL

Software Version – **A9** *(QUERY ONLY)* ALL

System Reset – **AE** *(Use value "0")* ALL

42

Complete OSD Menu Tree

Input Select – **4C** `ALL`
       Video 1 - **1**
       Video 2 - **2**
       S-Video 1 - **3**
       S-Video 2 - **4**
       Component 1 - **5**
       Component 2 - **6**
       RGBHV/Component - **7** `VP30` `VP50` `VP50PRO`
       HDMI 1 - **8**
       HDMI 2 - **9**
       HDMI 3 - **10**
       HDMI 4 - **11** `VP30` `VP50` `VP50PRO`
       SD/HD-SDI 1 - **12** `VP30` `VP50` `VP50PRO` (with SD/HD-SDI module installed)
       SD/HD-SDI 2 - **14** `VP50PRO` (with HD-SDI module installed)
       Auto - **13**
Input Aspect Ratio (OSD MENU ONLY)
       Frame AR – **4E** `ALL`
          4:3 - **1**
          16:9 - **2**
       Active AR – **50** `ALL`
          1.33:1 - **1**
          1.55:1 - **2**
          1.66:1 - **3**
          1.78:1 - **4**
          1.85:1 - **5**
          2.35:1 - **6**
          User - **7**
       Panorama – **A6** `VP50` `VP50PRO`
          On - **1**
          Off - **0**
       Zoom (see below)
          Horizontal – **40** `ALL`
             Range: **1.000-2.000**
          Vertical – **41** `ALL`
             Range: **1.000-2.000**
       Pan (see below)
          Horizontal – **42** `ALL`
             Range: **0-100**
          Vertical – **43** `ALL`
             Range: **0-100**

Borders (see below)

    Horizontal – **44** `ALL`

        Range: **0-200**

    Vertical – **45** `ALL`

        Range: **0-200**

Preset – **E1** `ALL`

    4:3 Full Frame - **1**

    Letterbox - **2**

    16:9 Full Frame - **3**

    4:3 Stretch - **14** `VP50` `VP50PRO`

    Preset 1 - **4**

    Preset 2 - **5**

    Preset 3 - **6**

    Preset 4 - **7**

    Preset 5 - **8**

    Preset 6 - **9**

    Preset 7 - **10**

    Preset 8 - **11**

    Preset 9 - **12**

    Preset 10 - **13**

    User - **0**

Save User to – **53** `ALL` *(there is no safety for this function)*

    Preset 1 - **1**

    Preset 2 - **2**

    Preset 3 - **3**

    Preset 4 - **4**

    Preset 5 - **5**

    Preset 6 - **6**

    Preset 7 - **7**

    Preset 8 - **8**

    Preset 9 - **9**

    Preset 10 - **10**

Input Adjust (OSD MENU ONLY)

    Mosquito Noise Reduction – **CA** `VP50PRO`

        Off - **0**

        Low - **1**

        Medium - **2**

        High - 3

Deinterlacing – **49** `ALL`

     Auto - **6**

     Film Bias Mode - **0**

     Video Mode - **1**

     Forced 3:2 Mode - **8** `VP50` `VP50PRO` or `VP20` `VP30` with ABT102 card

     Forced 2:2 Mode - **10** `VP50` `VP50PRO` or `VP20` `VP30` with ABT102 card

     2:2 Even Mode - **2** `VP50` `VP50PRO` or `VP20` `VP30` with ABT102 card

     2:2 Odd Mode - **3** `VP50` `VP50PRO` or `VP20` `VP30` with ABT102 card

     Game Mode 1 - **4** `VP50` `VP50PRO` or `VP20` `VP30` with ABT102 card

     Game Mode 2 - **5** `VP50` `VP50PRO` or `VP20` `VP30` with ABT102 card

     Field-Scale - **9** `VP50`

PReP – **B6** `VP50` `VP50PRO`

     Off - **0**

     On - **1**

Cadence Detect – **BB** `VP50PRO`

     Off - **0**

     On - **1**

Pass Through – **A7** `ALL`

     Off - **0**

     On - **1**

Overscan – **46** `ALL`

     Range: **0-20**

Image Shift (see below)

     Horizontal - **54** `ALL`

       Range: **0-30**

     Vertical – **47** `ALL`

       Range: **0-50**

Color Space – **87** `ALL`

     RGB - **1**

     YPbPr - **2**

     YCbCr 4:2:2 - **3**

     YCbCr 4:4:4 - **4**

     Auto - **5**

Input Level – **F0** `ALL`

     Video - **1**

     PC - **2**

VCR Mode – **48** `ALL`

     Off - **0**

     On - **1**

     Auto - **2**

HDMI Config. (see below)

     HDCP Mode – **86** `ALL`

       Off - **0**

       On - **1**

Auto AR – **B0** `ALL`

       Off - **0**

       On - **1**

Auto Color Space – **B1** `ALL`

       Off - **0**

       On - **1**

Auto Priority – **81** `ALL`

       Range: **1-13**

Audio Input – **4A** `ALL`

       Audio 1 - **1**

       Audio 2 - **2**

       Audio 3 - **3**

       Audio 4 - **4**

       Stereo - **5**

       HDMI - **6**

       Off - **0**

AV Lipsync – **4B** `ALL`

       Range: **0-200**

Picture Control (OSD MENU ONLY)

Fine Detail – **C8** `VP50PRO`

       Range: **(-100)-(+100)**

Edge Enhancement – **C9** `VP50PRO`

       Range: **(-100)-(+100)**

Brightness – **21** `ALL`

       Range: **(-100)-(+100)**

Contrast – **22** `ALL`

       Range: **(-100)-(+100)**

Saturation – **23** `ALL`

       Range: **(-100)-(+100)**

Hue – **24** `ALL`

       Range: **(-100)-(+100)**

Y/C Delay – **27** `ALL`

       Range: **(-100)-(+100)**

CUE Correction – **28** `ALL`

       Off - **0**

       On - **1**

       Auto - **2**

Configuration (OSD MENU ONLY)

Test Patterns – **80** `ALL`

    Off - **0**
    Frame Geometry - **1**
    Brightness/Contrast - **2**
    Checker board - **3**
    Vertical Lines - **4**
    Horizontal Lines - **5**
    Judder - **6**
    Color8 Bars75 - **7**
    Color8 Bars100 - **8**
    Window IRE10 - **9**
    Window IRE20 - **10**
    Window IRE30 - **11**
    Window IRE40 - **12**
    Window IRE50 - **13**
    Window IRE60 - **14**
    Window IRE70 - **15**
    Window IRE80 - **16**
    Window IRE90 - **17**
    Window IRE100 - **18**
    Gray Ramp - **19**
    XHatch Coarse - **20**
    XHatch Fine - **21**
    Focus - **22**
    Half B/W - **23**
    H-Clr7 Bars75 - **24**
    H-Clr7 Bars100 - **25**
    H-Clr8 Bars75 - **26**
    H-Clr8 Bars100 - **27**
    Black - **35**
    White - **28**
    Red - **29**
    Green - **30**
    Blue - **31**
    Cyan - **32**
    Magenta - **33**
    Yellow - **34**

Auto Standby – **83** `ALL`

    Off - **0**
    On - **1**

LED Brightness (OSD MENU ONLY)

    Navigation – **EC** `ALL`
        Range: **0-3**
    Normal – **ED** `ALL`
        Range: **0-3**

User Mode – **85** `ALL`
    Normal - **1**
    Advanced - **2**
Serial Port Rate – **A3** `ALL` *(Unit will reply with acknowledge, then switch to new baud-rate)*
    4800bps - **1**
    9600bps - **2**
    14400bps - **3**
    19200bps - **4**
    38400bps - **5**
    57600bps - **6**
Factory Default – **AC** `VP30` `VP50` `VP50PRO` *(Use value "0" – there is no safety for this function)*
Software Update – **AD** `VP30` `VP50` `VP50PRO` *(Use value "0" – there is no safety for this function)*
12V Trigger Levels (OSD MENU ONLY)
    Trigger #1 – **B8** `VP50PRO`
        Normal - **1**
        Negative - **2**
    Trigger #2 – **B9** `VP50PRO`
        Normal - **1**
        Negative - **2**
Information – **A5** `ALL`
    Off - **0**
    On - **1**
Output Setup (OSD MENU ONLY)
    Analog/Digital – **60** `VP30` `VP50` `VP50PRO`
        BNC (Analog) - **1**
        HDMI (Digital) - **2**
    Format - **61** `ALL`
        480p - **1**
        540p - **2**
        576p - **3**
        720p-50 - **4**
        720p-60 - **5**
        1080i-50 - **6**
        1080i-60 - **7**
        1080p-24 - **37**
        1080p-25 - **38**
        1080p-30 – *Not Defined Yet* `VP50PRO`
        1080p-48 - **30** `VP50PRO`
        1080p-50 - **8**
        1080p-60 - **9**
        640x480 (VGA) - **10**
        800x600 (SVGA) - **11**
        1024x768 (XGA) - **12**
        1280x1024 (SXGA) - **13**
        848x480 - **34**
        852x480 - **14**

1365x768 - **35**

852x576 - **15**

1366x768 (1) - **16**

1366x768 (2) - **33**

1360x768 (1) - **31**

1360x768 (2) - **32**

1280x768 - **17**

1024x1024 - **18**

1024x852 - **19**

1024x768 - **36**

1024x576 - **20**

848x600 - **21**

1365x1024 - **22**

1400x1050 - **23**

1400x788 - **24**

960x540 - **25**

1280x960 - **26**

1440x960 - **27**

1440x1152 - **28**

User - **29**

USER RESOLUTION CONTROLS:

    Horizontal Shift (OSD MENU ONLY, SET FRONT PORCH AND BACK PORCH)

    Horizontal Size – **62** `VP30` `VP50` `VP50ᴾᴿᴼ`

        Range: **640-2000** *(Limited Pixel clock, must not exceed 180MHz)*

    Horizontal Front Porch – **63** `VP30` `VP50` `VP50ᴾᴿᴼ`

        Range: **0-512** *(See VESA timing specifications for guidance)*

    Horizontal Sync – **64** `VP30` `VP50` `VP50ᴾᴿᴼ`

        Range: **0-512** *(See VESA timing specifications for guidance)*

    Horizontal Back Porch – **65** `VP30` `VP50` `VP50ᴾᴿᴼ`

        Range: **0-512** *(See VESA timing specifications for guidance)*

    Vertical Shift (OSD MENU ONLY, SET FRONT PORCH AND BACK PORCH)

    Vertical Size – **66** `VP30` `VP50` `VP50ᴾᴿᴼ`

        Range: **480-2000** *(Limited Pixel clock, must not exceed 180MHz)*

    Vertical Front Porch - `VP30` `VP50` `VP50ᴾᴿᴼ`

        Range: **0-512** *(See VESA timing specifications for guidance)*

    Vertical Sync - `VP30` `VP50` `VP50ᴾᴿᴼ`

        Range: **0-512** *(See VESA timing specifications for guidance)*

    Vertical Back Porch - `VP30` `VP50` `VP50ᴾᴿᴼ`

        Range: **0-512** *(See VESA timing specifications for guidance)*

Aspect Ratio (OSD MENU ONLY)

Display – **6A** `ALL`

 4:3 - **1**

 5:4 - **2**

 16:9 - **3**

 2.35:1 - **4**

 User - **5**

  Display User Value – **88** `ALL`

   Range: **1.00-3.00**

Lens – **B7** `VP50PRO`

 Mode 1 - **1**

 Mode 1 "Auto" - **2**

 Mode 2 - **3**

 None - **0**

Screen – **89** `ALL`

 4:3 - **1**

 5:4 - **2**

 16:9 - **3**

 2.35:1 - **4**

 User - **5**

  Screen User Value – **8A** `ALL`

   Range: **1.00-3.00**

Image Shift (OSD MENU ONLY)

 Vertical – **8C** `VP30` `VP50` `VP50PRO` *(some underscan must be set first)*

  Range: **(-30)-(+30)**

 Horizontal – **8D** `VP30` `VP50` `VP50PRO` *(some underscan must be set first)*

  Range: **(-30)-(+30)**

Underscan – **8B** `VP30` `VP50` `VP50PRO`

 Range: **0-100**

Sync Type – **6B** `ALL`

 Bi-Level - **1**

 Tri-Level - **2**

 Composite - **3**

 +H/+V - **4**

 +H/-V - **5**

 -H/+V - **6**

 -H/-V - **7**

Color Space – **6C** `ALL`

 RGB - **1**

 YPbPr - **2**

 YCbCr 4:2:2 - **3**

 YCbCr 4:4:4 - **4**

  Color Gamut – **E5** `VP50PRO` *When 4:2:2 or 4:4:4*

   BT.601 - **1**

   BT.709 - **2**

50

Output Level – **E6** `ALL`
      Video - **1**
      PC - **2**

Framerate (OSD MENU ONLY)
    When **input** is: 24Hz - NOT YET DEFINED `VP50PRO`
        24Hz Lock - **1**
        48Hz Lock - **2**
        60Hz Lock - **3**
        72Hz Lock - **4**
        Unlock - **0**
            24Hz input, Unlocked output framerate – NOT YET DEFINED `VP50PRO`
            Range: **24.00-80.00**
    When **input** is: 25Hz - NOT YET DEFINED `VP50PRO`
        25Hz Lock - **1**
        50Hz Lock - **2**
        75Hz Lock - **3**
        Unlock - **0**
            25Hz input, Unlocked output framerate – NOT YET DEFINED `VP50PRO`
            Range: **24.00-80.00**
    When **input** is: 30Hz - NOT YET DEFINED `VP50PRO`
        30Hz Lock - **1**
        60Hz Lock - **2**
        Unlock - **0**
            30Hz input, Unlocked output framerate – NOT YET DEFINED `VP50PRO`
            Range: **24.00-80.00**
    When **input** is: 50Hz – **6D** `ALL`
        25Hz Lock - **1**
        50Hz Lock - **2**
        75Hz Lock - **3**
        Unlock - **0**
            50Hz input, Unlocked output framerate – `ALL`
            Range: **24.00-80.00**
    When **input** is: 60Hz – **6E** `ALL`
        24Hz Lock - **1**
        48Hz Lock - **2**
        60Hz Lock - **3**
        72Hz Lock - **4**
        Unlock - **0**
            60Hz input, Unlocked output framerate – **6F** `ALL`
            Range: **24.00-80.00**

Border Level – **4F** `ALL`
        Range: **(-16)-(+100)**

Output Picture Controls (OSD MENU ONLY)
        Presets – **C4** `VP50PRO`
            ISF Day Normal - **1**
            ISF Day Bright - **2**
            ISF Night - **3**
            Preset 1 - **4**
            Preset 2 - **5**
        Brightness – **C0** `VP50PRO`
            Range: **(-100)-(+100)**
        Contrast – **C1** `VP50PRO`
            Range: **(-100)-(+100)**
        Saturation – **C2** `VP50PRO`
            Range: **(-100)-(+100)**
        Hue – **C3** `VP50PRO`
            Range: **(-100)-(+100)**
HDCP Mode - **EA** `ALL`
        Off - **0**
        On - **1**
12V Trigger #2 – **C7** `VP50PRO`
        Lens - **2**
        On - **1**
        Off - **0**
Audio Select - **BA** `VP50PRO`
        S/PDIF - **1**
        HDMI - **2**

Display Profile (OSD MENU ONLY)
        Select – **E0** `VP30` `VP50` `VP50PRO`
            Profile 1 - **1**
            Profile 2 - **2**
            Profile 3 - **3**
            Profile 4 - **4**
            Profile 5 - **5**
            Profile 6 - **6**
            Profile 7 - **7**
            Profile 8 - **8**
            Profile 9 - **9**
            Profile 10 - **10**
        Save – **52** `VP30` `VP50` `VP50PRO` *(there is no safety for this function)*
            Range: **1-10**
        Auto – **E7** `VP30` `VP50` `VP50PRO`
            Off - **0**
            On - **1**

Appendix A.   Decimal to Binary to HEX to ASCII Conversion Table

Some ASCII Characters will not be used ever in the iScan communication – these are grayed out for clarity (the entire list is published for the sake of completion).

Decimal	Binary (MSB)	HEX	ASCII
0	0000 0000	0x00	NUL – Null
1	0000 0001	0x01	SOH – Start of Heading
2	0000 0010	0x02	STX – Start of Text
3	0000 0011	0x03	ETX – End of Text
4	0000 0100	0x04	EOT – End of Transmission
5	0000 0101	0x05	ENQ – Enquiry
6	0000 0110	0x06	ACK – Acknowledge
7	0000 0111	0x07	BEL – Bell
8	0000 1000	0x08	BS – Backspace
9	0000 1001	0x09	HT – Horizontal Tab
10	0000 1010	0x0A	LF – Line Feed/New Line
11	0000 1011	0x0B	VT – Vertical Tab
12	0000 1100	0x0C	FF – Form Feed/New Page
13	0000 1101	0x0D	CR – Carriage Return
14	0000 1110	0x0E	SO – Shift Out
15	0000 1111	0x0F	SI – Shift In
16	0001 0000	0x10	DLE – Data Link Escape
17	0001 0001	0x11	DC1 – Device Control 1
18	0001 0010	0x12	DC2 – Device Control 2
19	0001 0011	0x13	DC3 – Device Control 3
20	0001 0100	0x14	DC4 – Device Control 4
21	0001 0101	0x15	NAK – Negative Acknowledge
22	0001 0110	0x16	SYN – Synchronous Idle
23	0001 0111	0x17	ETB – End of Transmission Block
24	0001 1000	0x18	CAN – Cancel
25	0001 1001	0x19	EM – End of Medium
26	0001 1010	0x1A	SUB – Substitute
27	0001 1011	0x1B	ESC – Escape
28	0001 1100	0x1C	FS – File Separator
29	0001 1101	0x1D	GS – Group Separator
30	0001 1110	0x1E	RS – Record Separator
31	0001 1111	0x1F	US – Unit Separator
32	0010 0000	0x20	SPC - Space
33	0010 0001	0x21	!
34	0010 0010	0x22	"
35	0010 0011	0x23	#
36	0010 0100	0x24	$
37	0010 0101	0x25	%
38	0010 0110	0x26	&

39	0010 0111	0x27	'
40	0010 1000	0x28	(
41	0010 1001	0x29	)
42	0010 1010	0x2A	*
43	0010 1011	0x2B	+
44	0010 1100	0x2C	,
45	0010 1101	0x2D	-
46	0010 1110	0x2E	.
47	0010 1111	0x2F	/
48	0011 0000	0x30	0
49	0011 0001	0x31	1
50	0011 0010	0x32	2
51	0011 0011	0x33	3
52	0011 0100	0x34	4
53	0011 0101	0x35	5
54	0011 0110	0x36	6
55	0011 0111	0x37	7
56	0011 1000	0x38	8
57	0011 1001	0x39	9
58	0011 1010	0x3A	:
59	0011 1011	0x3B	;
60	0011 1100	0x3C	<
61	0011 1101	0x3D	=
62	0011 1110	0x3E	>
63	0011 1111	0x3F	?
64	0100 0000	0x40	@
65	0100 0001	0x41	A
66	0100 0010	0x42	B
67	0100 0011	0x43	C
68	0100 0100	0x44	D
69	0100 0101	0x45	E
70	0100 0110	0x46	F
71	0100 0111	0x47	G
72	0100 1000	0x48	H
73	0100 1001	0x49	I
74	0100 1010	0x4A	J
75	0100 1011	0x4B	K
76	0100 1100	0x4C	L
77	0100 1101	0x4D	M
78	0100 1110	0x4E	N
79	0100 1111	0x4F	O
80	0101 0000	0x50	P
81	0101 0001	0x51	Q
82	0101 0010	0x52	R
83	0101 0011	0x53	S

84	0101 0100	0x54	T
85	0101 0101	0x55	U
86	0101 0110	0x56	V
87	0101 0111	0x57	W
88	0101 1000	0x58	X
89	0101 1001	0x59	Y
90	0101 1010	0x5A	Z
91	0101 1011	0x5B	[
92	0101 1100	0x5C	\
93	0101 1101	0x5D	]
94	0101 1110	0x5E	^
95	0101 1111	0x5F	_
96	0110 0000	0x60	`
97	0110 0001	0x61	a
98	0110 0010	0x62	b
99	0110 0011	0x63	c
100	0110 0100	0x64	d
101	0110 0101	0x65	e
102	0110 0110	0x66	f
103	0110 0111	0x67	g
104	0110 1000	0x68	h
105	0110 1001	0x69	i
106	0110 1010	0x6A	j
107	0110 1011	0x6B	k
108	0110 1100	0x6C	l
109	0110 1101	0x6D	m
110	0110 1110	0x6E	n
111	0110 1111	0x6F	o
112	0111 0000	0x70	p
113	0111 0001	0x71	q
114	0111 0010	0x72	r
115	0111 0011	0x73	s
116	0111 0100	0x74	t
117	0111 0101	0x75	u
118	0111 0110	0x76	v
119	0111 0111	0x77	w
120	0111 1000	0x78	x
121	0111 1001	0x79	y
122	0111 1010	0x7A	z
123	0111 1011	0x7B	{
124	0111 1100	0x7C		
125	0111 1101	0x7D	}
126	0111 1110	0x7E	~
127	0111 1111	0x7F	DEL
128	1000 0000	0x80	No ASCII Character at This Value

129	1000 0001	0x81	No ASCII Character at This Value
130	1000 0010	0x82	No ASCII Character at This Value
131	1000 0011	0x83	No ASCII Character at This Value
132	1000 0100	0x84	No ASCII Character at This Value
133	1000 0101	0x85	No ASCII Character at This Value
134	1000 0110	0x86	No ASCII Character at This Value
135	1000 0111	0x87	No ASCII Character at This Value
136	1000 1000	0x88	No ASCII Character at This Value
137	1000 1001	0x89	No ASCII Character at This Value
138	1000 1010	0x8A	No ASCII Character at This Value
139	1000 1011	0x8B	No ASCII Character at This Value
140	1000 1100	0x8C	No ASCII Character at This Value
141	1000 1101	0x8D	No ASCII Character at This Value
142	1000 1110	0x8E	No ASCII Character at This Value
143	1000 1111	0x8F	No ASCII Character at This Value
144	1001 0000	0x90	No ASCII Character at This Value
145	1001 0001	0x91	No ASCII Character at This Value
146	1001 0010	0x92	No ASCII Character at This Value
147	1001 0011	0x93	No ASCII Character at This Value
148	1001 0100	0x94	No ASCII Character at This Value
149	1001 0101	0x95	No ASCII Character at This Value
150	1001 0110	0x96	No ASCII Character at This Value
151	1001 0111	0x97	No ASCII Character at This Value
152	1001 1000	0x98	No ASCII Character at This Value
153	1001 1001	0x99	No ASCII Character at This Value
154	1001 1010	0x9A	No ASCII Character at This Value
155	1001 1011	0x9B	No ASCII Character at This Value
156	1001 1100	0x9C	No ASCII Character at This Value
157	1001 1101	0x9D	No ASCII Character at This Value
158	1001 1110	0x9E	No ASCII Character at This Value
159	1001 1111	0x9F	No ASCII Character at This Value
160	1010 0000	0xA0	No ASCII Character at This Value
161	1010 0001	0xA1	No ASCII Character at This Value
162	1010 0010	0xA2	No ASCII Character at This Value
163	1010 0011	0xA3	No ASCII Character at This Value
164	1010 0100	0xA4	No ASCII Character at This Value
165	1010 0101	0xA5	No ASCII Character at This Value
166	1010 0110	0xA6	No ASCII Character at This Value
167	1010 0111	0xA7	No ASCII Character at This Value
168	1010 1000	0xA8	No ASCII Character at This Value
169	1010 1001	0xA9	No ASCII Character at This Value
170	1010 1010	0xAA	No ASCII Character at This Value
171	1010 1011	0xAB	No ASCII Character at This Value
172	1010 1100	0xAC	No ASCII Character at This Value
173	1010 1101	0xAD	No ASCII Character at This Value

174	1010 1110	0xAE	No ASCII Character at This Value
175	1010 1111	0xAF	No ASCII Character at This Value
176	1011 0000	0xB0	No ASCII Character at This Value
177	1011 0001	0xB1	No ASCII Character at This Value
178	1011 0010	0xB2	No ASCII Character at This Value
179	1011 0011	0xB3	No ASCII Character at This Value
180	1011 0100	0xB4	No ASCII Character at This Value
181	1011 0101	0xB5	No ASCII Character at This Value
182	1011 0110	0xB6	No ASCII Character at This Value
183	1011 0111	0xB7	No ASCII Character at This Value
184	1011 1000	0xB8	No ASCII Character at This Value
185	1011 1001	0xB9	No ASCII Character at This Value
186	1011 1010	0xBA	No ASCII Character at This Value
187	1011 1011	0xBB	No ASCII Character at This Value
188	1011 1100	0xBC	No ASCII Character at This Value
189	1011 1101	0xBD	No ASCII Character at This Value
190	1011 1110	0xBE	No ASCII Character at This Value
191	1011 1111	0xBF	No ASCII Character at This Value
192	1100 0000	0xC0	No ASCII Character at This Value
193	1100 0001	0xC1	No ASCII Character at This Value
194	1100 0010	0xC2	No ASCII Character at This Value
195	1100 0011	0xC3	No ASCII Character at This Value
196	1100 0100	0xC4	No ASCII Character at This Value
197	1100 0101	0xC5	No ASCII Character at This Value
198	1100 0110	0xC6	No ASCII Character at This Value
199	1100 0111	0xC7	No ASCII Character at This Value
200	1100 1000	0xC8	No ASCII Character at This Value
201	1100 1001	0xC9	No ASCII Character at This Value
202	1100 1010	0xCA	No ASCII Character at This Value
203	1100 1011	0xCB	No ASCII Character at This Value
204	1100 1100	0xCC	No ASCII Character at This Value
205	1100 1101	0xCD	No ASCII Character at This Value
206	1100 1110	0xCE	No ASCII Character at This Value
207	1100 1111	0xCF	No ASCII Character at This Value
208	1101 0000	0xD0	No ASCII Character at This Value
209	1101 0001	0xD1	No ASCII Character at This Value
210	1101 0010	0xD2	No ASCII Character at This Value
211	1101 0011	0xD3	No ASCII Character at This Value
212	1101 0100	0xD4	No ASCII Character at This Value
213	1101 0101	0xD5	No ASCII Character at This Value
214	1101 0110	0xD6	No ASCII Character at This Value
215	1101 0111	0xD7	No ASCII Character at This Value
216	1101 1000	0xD8	No ASCII Character at This Value
217	1101 1001	0xD9	No ASCII Character at This Value
218	1101 1010	0xDA	No ASCII Character at This Value

219	1101 1011	0xDB	No ASCII Character at This Value
220	1101 1100	0xDC	No ASCII Character at This Value
221	1101 1101	0xDD	No ASCII Character at This Value
222	1101 1110	0xDE	No ASCII Character at This Value
223	1101 1111	0xDF	No ASCII Character at This Value
224	1110 0000	0xE0	No ASCII Character at This Value
225	1110 0001	0xE1	No ASCII Character at This Value
226	1110 0010	0xE2	No ASCII Character at This Value
227	1110 0011	0xE3	No ASCII Character at This Value
228	1110 0100	0xE4	No ASCII Character at This Value
229	1110 0101	0xE5	No ASCII Character at This Value
230	1110 0110	0xE6	No ASCII Character at This Value
231	1110 0111	0xE7	No ASCII Character at This Value
232	1110 1000	0xE8	No ASCII Character at This Value
233	1110 1001	0xE9	No ASCII Character at This Value
234	1110 1010	0xEA	No ASCII Character at This Value
235	1110 1011	0xEB	No ASCII Character at This Value
236	1110 1100	0xEC	No ASCII Character at This Value
237	1110 1101	0xED	No ASCII Character at This Value
238	1110 1110	0xEE	No ASCII Character at This Value
239	1110 1111	0xEF	No ASCII Character at This Value
240	1111 0000	0xF0	No ASCII Character at This Value
241	1111 0001	0xF1	No ASCII Character at This Value
242	1111 0010	0xF2	No ASCII Character at This Value
243	1111 0011	0xF3	No ASCII Character at This Value
244	1111 0100	0xF4	No ASCII Character at This Value
245	1111 0101	0xF5	No ASCII Character at This Value
246	1111 0110	0xF6	No ASCII Character at This Value
247	1111 0111	0xF7	No ASCII Character at This Value
248	1111 1000	0xF8	No ASCII Character at This Value
249	1111 1001	0xF9	No ASCII Character at This Value
250	1111 1010	0xFA	No ASCII Character at This Value
251	1111 1011	0xFB	No ASCII Character at This Value
252	1111 1100	0xFC	No ASCII Character at This Value
253	1111 1101	0xFD	No ASCII Character at This Value
254	1111 1110	0xFE	No ASCII Character at This Value
255	1111 1111	0xFF	No ASCII Character at This Value

Appendix B.   Infrared Control White Paper by Barry Gordon

Note: This article in Sections 2.1 through 2.5 is reprinted with permission from Barry Gordon. The original article was printed circa 1998, although the information it contains is still very pertinent. We wish to thank Barry for allowing us to reprint the article. Anchor Bay is not responsible for the information presented within the below article.

Infrared Signaling and how it works

Acknowledgments: This document could not have been possible for me to write without the assistance of a lot of people who contributed their time and effort to helping me understand the various parts of the ProntoEdit IR display format. I would like to thank; AHP (A Helpful Person), Jack Schultz, Manu Duarte, Timm, CDecker, and others. Please pardon my use of the BBS handles, but in many cases that is the only way I know them.

Warning: This document will give the reader enough information to develop and hand enter IR codes rather than learning them from a remote. That is not the intent of the document, merely a byproduct of the knowledge you can gain. Many devices controlled by IR remotes, in Particular TV's, have undocumented IR code sequences used for servicing the equipment by factory trained technicians in possession of detailed service manuals and test equipment. By causing a service code to be sent to your TV or other device, you may place it in a state where it no longer operates as desired, or at all. An example of this might be the resetting of all convergence offsets, or altering the width or height of the picture. Be careful, if you are not sure of what the outcome might be, perhaps you should not do it.

IR remotes operate by modulating (turning on and off) an infra red (IR) light source. When the IR light source (the IR emitter) is "on" it is actually turning itself on and off thousands of times per second, too fast for the human eye to follow. The rate at which this occurs is called the carrier frequency. The terminology comes from the metaphor that the "carrier" carries the "information". This is done to provide a better transmission system and allow the overall IR system (transmitter and receiver) to operate in noisy (with respect to light) environments. It is important to understand that the IR receiver for a given remote is tuned to IR "carrier" frequency for that remote and will effectively not see IR signals sent on a different carrier frequency such as from other remotes. [Note: The human eye can never see an infrared transmission, so the concept of on and off is not with regards to visible light. Some equipment has a "telltale", a little red light that visibly flashes when the equipment receives IR signals. That is what we can see]

The "information" is placed on the "carrier" using several different techniques. The most common technique is Pulse Width Modulation. In Pulse Width Modulation the duration of the ON (carrier present, light flashing thousands of times per second), or Off (no light at all coming out of the IR emitter) periods is made to vary. Lets assume, because this is what is done, that we wish to send numbers representing what key has

been pressed (and perhaps even what device this key is for). We first need to simplify the problem so that we don't have deal with too many "Pulse widths". We can easily do this by representing the number in base 2, or binary. (I apologize if this now gets a little technical, but in reality it already has). In binary there are only two digits to worry about not ten as in decimal. Therefore we only need to have two distinct "pulse widths". If you think about it, the periods of on and off will need to alternate. If they didn't it would be hard to judge their width. [Note: Other modulation schemes in particular RC5 do not use PWM. RC5 uses Phase modulation. Luckily for us we never have to decode or figure out the RC5 patterns because Philips has provided them as pure clean data.] Only one of the widths needs to vary. Either the width of the ON period or the Width of the Off period.

In summary, IR transmission most often takes place by varying the on off times of an IR emitter to represent binary numbers according to some well established pattern.

[Note: At this point I am going to assume that the reader has a basic understanding of the binary numbering system. Not detailed enough to add, subtract or multiply them, but enough to be able to form the decimal value of a binary number.]

Each manufacturer has the option of deciding just how big a number he wishes to send to his equipment, and what meaning is given to that number (or numbers) when they are received. Remember the environment through which the IR signals are passing (the air) is noisy in a light sense. Bright sunlight, Fluorescent lights, all contribute to the noise. Some manufacturers add additional "redundant" information such as sending the numbers twice to ensure that they get to the equipment correctly. Some do not. I will discuss those details when I discuss some of the more common manufacturer's products.

The Philips ProntoEdit HEX Format

This discussion is only completely valid for IR transmissions using Pulse Width Modulation. Keep in mind that the sole purpose of the HEX data is to represent a series of ON and OFF times for the IR emitter, and when the IR emitter appear to be solidly ON its is rapidly flashing. The ProntoEdit HEX format uses a pair of numbers to represent an on/off sequence. We will call this a "Burst Pair" (thanks to AHP). The first digit represents an ON time and the second an Off time.

The question is how much time? What the burst pair really contains is the number of cycles of the carrier for which to turn the light on and off. The carrier frequency therefore acts as the clock (not totally true, but good enough for this discussion). To illustrate the point, let us assume a carrier frequency of 40 kilohertz (that is 40,000 cycles per second). This is a very common IR carrier frequency. One cycle of that carrier takes 1/40000 units of time or 25 microseconds. A "burst pair" of 48,24 would turn the IR emitter on for 48*25 Microseconds, and off for 24*25 microseconds. A "burst Pair" of 24,24 would turn the IR emitter on for 24*25 Microseconds, and off for 24*25 microseconds. Because we are using binary numbers we only have two digits to represent (0,1) as opposed to decimal where we would need 10 unique burst pair patterns to represent the 10 decimal digits. We could for example decide the encoding of a "1" will be represented by having the On period twice as long as the Off period, and a "0" by

60

having them equal. We might choose 48,24 for the "1" and 24,24 for the "0". In fact this is what Sony has done in its IR remotes. [Note: If you work through the numbers you will find that Sony IR signaling uses a sequence of 1200 microseconds of light followed by 600 microseconds of no light to represent a "1"; and a sequence of 600 microseconds of light followed by 600 microseconds of no light to represent a "0"]. In general all IR equipment is forgiving and operates with in a timing tolerance of +/- 10%.

A full IR key code as encoded in the ProntoEdit Hex display contains three discrete parts.

| Preamble | Burst Pair Sequence 1 | Burst Pair Sequence 2 |

Either one of the burst pair sequences is optional so we will actually have three different patterns of IR encoding.

Preamble	Burst Pair Sequence 1	Burst Pair Sequence 2
Preamble	Burst Pair Sequence 1	
Preamble	Burst Pair Sequence 2	

The preamble does not contain Burst Pairs but rather four (4) hexadecimal (HEX, base 16) numbers, each of which has a precise meaning. I will only discuss them in the context of Learned IR codes. Each Hex number consists of 4 digits.

The first number is always a zero (0000) it indicates that the IR pattern is raw data, which means it was learned.

The second number is the frequency of the IR carrier in terms of the Pronto internal clock. The following formula where N represents the decimal value of this hex number will give you the frequency of the carrier in Kilohertz: Frequency = 1000000/(N * .241246)

A Sony remote will usually have a value for N of 103 (this shows as 67 Hex). Doing the arithmetic we have Freq=1000000/(103*. 241246)= 40,244 or approximately 40,000 cycles per second (well within a tolerance of 40,000 +/- 10%)

The third number is the number of Burst Pairs in Burst Pair Sequence #1. Each Burst pair consists of two 4 digit Hex numbers representing the On and Off time of that burst (single binary Bit).

The fourth number is the number of Burst Pairs in Burst Pair Sequence #2.

Burst Pair Sequence #1 starts at word 5 if it is present and is immediately followed by the digits of Burst Pair Sequence #2 if it is present (word 4>0000). If Sequence #1 is missing (word 3=0000), then Burst Sequence Number 2 starts at word 5.

61

A Burst Pair Sequence usually looks as follows:

Lead in Burst Pair	Data Burst Pairs	Lead Out Burst Pair
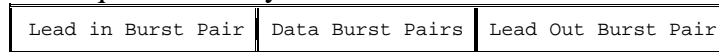
The Lead In Burst pair can be thought of as the hello or wake up burst. It tells the receiver to start listening (or rather looking) very closely as what is coming. It is usually of different timing duration than the Burst Pairs in the data part. Technically it is also used to set the receivers AGC level, a factor related to how much the receiver will amplify the IR light it sees.

The Lead Out burst pair marks the end of the message and usually has a long OFF time period to guarantee that two IR messages can't be sent too close together. It may actually be incorporated as part of the last data bit if the ON period is what carries the information (that is, the off time is constant in the data portion and the On time varies between two values). Once again, Sony does exactly that.

Remember all data in the IR Hex display is in Hex and to properly interpret these values you must convert them to decimal. Two values should be considered equal if they are within about 10% of each other. They don't have to be exactly the same.

[Digression to convert a 4 digit Hex "WXYZ" number to decimal, the following formula will work W*4096+X*256+Y*16+Z. W,X,Y,Z represent HEX digits in the range 0-15 where a=10, b=11,c=12,d=13,e=14,f=15. A hex value of 0067 is therefore 16*6+7=103)].

If you understand all of what has been discussed so far you have based the basics of Binary Signaling 101. Go take a breather.

Before we look at some actual IR codes and their detailed formats, we should understand why there might be two burst sequences in the code and not always just one. The first burst sequence is the Once sequence. It is sent if you tap the button on the Pronto which has learned this code. The second burst sequence is called the Repeat sequence. It is sent repeatedly as long as you hold the key on the Pronto down. If you have used an IR remote you already know that all buttons do not "repeat". The two sequences do not have to be the same. In many cases they are, in others they are not. This is generally manufacturer dependent.

IR Codes

The world of IR remotes has become a commodity world. IR remotes (simple ones, not the Pronto) are relatively inexpensive. I bought 5, credit card sized, universal remotes for $10. They are three times as thick as a credit card but the same height and width. Fits nicely in a shirt pocket. (A true couch potato must NEVER EVER be without a remote!).

This has happened because there has been a large degree of standardization on the chips that generate the IR codes and receive them. In fact there are only about 5 or 6 such chips being used. Sony, Sharp, Toshiba, Philips and NEC are the most popular, with the NEC one being the most popular of all. The majority of the Asian rim manufacturers (except for Sony, Sharp, Toshiba, and Philips) use NEC chips and therefore NEC format.

I will discuss the exact coding of two of these systems, Sony and NEC. I believe Pioneer, Onkyo, Akai, Canon, Goldstar (now LG), Hitachi, Kenwood, NEC, Teac, and Yamaha all use the NEC chip.

[Note: IR data is always transmitted least significant bit first so the first data bit sent is lowest order and in a real binary representation it would be the rightmost bit having a weight of 1.]

SONY IR CODING

Parameter	Decimal Value	HEX Value
Carrier Frequency	40kHz	
Unit of Burst Time	25 cycles of the carrier	
Lead In Burst	96 24	0060 0018
"1" Burst Pattern	48 24	0030 0018
"0" Burst Pattern	24 24	0018 0018
Lead Out	X, 1024	0018 03f6 or 0030 03f6

The lead out pattern in the Sony code is added to the last bit by increasing the off time. It is NOT a separate burst of data.

Sony data consists of a different number of bits in the message. The first seven bits (the first seven burst pairs after the lead in burst) always represent the key pressed on the remote. The next N bits where in is 5, 8, or 13 represents a device code. Older Sony devices like a TV (no matter what its true model age, it is a device made by Sony for a long time so it is "old") uses a 12 bit code. A newer one like the DVD S7000 uses a 20 bit code. Some remotes can control more than 1 device so they can send codes of different lengths.

Here is an example from a Sony DVD S7000 as it appears in the ProntoEdit Hex Display

```
    0000 0067 0000 0015 0060 0018 0018 0018 0030 0018 0030
0018 0030 0018 0018 0018 0030 0018 0018 0018 0018 0018 0030
0018 0018 0018 0030 0018 0030 0018 0030 0018 0018 0018 0018
0018 0030 0018 0018 0018 0018 0018 0030 0018 0018 03f6
```

Let us break it up to decipher it.

Preamble	0000 0067 000 0015
Word 1	0 so it is a learned IR code
Word 2	103 decimal which when plugged into the formula already given yields an IR Carrier frequency of about 40kHz.
Word 3	0000 is the length of the One Time Burst. There is no one time burst
Word 4	Decimal 21 is the length of the repeat burst. There are 21 bits (Burst pairs) in this code. The code length is 20 bits plus 1 more pair for the Lead in.
Word 5,6	0060 0018 (96,24 decimal) The lead in Burst . 4 units of on followed by 1 unit of off, where a unit is 600 microseconds
Word 7,8	0018 0018 (24,24 decimal) Burst pair 1, bit 1 = "0"
Word 9,10	0030 0018 (48,24 decimal) Burst Pair 2, bit 2 = "1"
Word 11,12	0030 0018 (48,24 decimal) Burst Pair 3, bit 3 = "1"
Word 13,14	0030 0018 (48,24 decimal) Burst Pair 4, bit 4 = "1"
Word 15,16	0018 0018 (24,24 decimal) Burst Pair 5, bit 5 = "0"
Word 17,18	0030 0018 (48,24 decimal) Burst Pair 6, bit 6 = "1"
Word 19,20	0018 0018 (24,24 decimal) Burst Pair 7, bit 7 = "0"

The above is the function code as transmitted it is 0111010. Reversing the string so it is a true binary number with the least significant digit on the right we get 0101110 which in decimal is 46.

Continuing on to the device code we have:

Word 21,22	0018 0018 (24,24 decimal) Burst Pair 8, bit 1 = "0"
Word 23,24	0030 0018 (48,24 decimal) Burst Pair 9, bit 2 = "1"
Word 25,26	0018 0018 (24,24 decimal) Burst Pair 10, bit 3 = "0"
Word 27,28	0030 0018 (48,24 decimal) Burst Pair 11, bit 4 = "1"
Word 29,30	0030 0018 (48,24 decimal) Burst Pair 12, bit 5 = "1"
Word 31,32	0030 0018 (48,24 decimal) Burst Pair 13, bit 6 = "1"
Word 33,34	0018 0018 (24,24 decimal) Burst Pair 14, bit 7 = "0"
Word 35,36	0018 0018 (24,24 decimal) Burst Pair 15, bit 8 = "0"
Word 37,38	0030 0018 (48,24 decimal) Burst Pair 16, bit 9 = "1"
Word 39,40	0018 0018 (24,24 decimal) Burst Pair 17, bit 10 = "0"
Word 41,42	0018 0018 (24,24 decimal) Burst Pair 18, bit 11 = "0"
Word 43,44	0030 0018 (48,24 decimal) Burst Pair 19, bit 12 = "1"
Word 45,46	0018 03fc (24,24 decimal) Burst Pair 20, bit 13 = "0"

The device code as transmitted is 0101110010010. Reversing the order to make it a binary number we get 0100100111010. Converting it to decimal we get 2362.

This means that the Sony DVD S7000 has a device code of 2362 and this key has a function code of 46. This is the discrete Power ON key. If a Sony device has a discrete Power on Code it is normally 46. Note the dead time on the second half of the last data burst pair. Sony does not use a unique lead out, but rather adds the inter-message minimum time to the last data burst's off period

Sony codes are fairly simple. Sony builds a lot of power into the IR senders, and good noise rejection in their receivers. They use no redundancy or error checking in the code

65

NEC IR Code Format

Parameter	Decimal Value	HEX Value
Carrier Frequency	40kHz	
Unit of Burst Time	22 cycles of the carrier	
Lead In Burst	341 171	0156 00ab
"1" Burst Pattern	22 96	0016 0060
"0" Burst Pattern	22 24	0016 0016
Lead Out	22, 1427	0016 0593

Doing the arithmetic we see that this code uses a base time of 550 microseconds. The lead in is a unique burst as is the lead out. It is a pulse width modulation system where the information is carried in the length of the off time with a fixed duration of on time. The NEC message format is quite a bit more complicated then that of Sony. It is always a 32-bit code. Which consists of 16 bits of data and 16 bits of error checking.

The code is divided into four 8-bit fields.

Device Code	Device code Compliment	Function Code	Function Code Compliment

A device code will be in the range of 0 to 255 or 256 discrete device codes. The same is true of the function code. The compliment fields are the 1's compliment of the code they represent. The device code and the device code compliment must add up to 255 or else there is an error. The same is true of the function code and the function code compliment. NEC uses a discrete lead in and a discrete lead out, so the total code length will take 34 burst pairs to represent as a Burst Pair Sequence.

The following as an example of a Pioneer IR sequence for the CLD79 Elite Laser Disk Player.

```
    0000 0067 0000 0022 0156 00ab 0016 0060 0016 0060 0016
0060 0016 0016 0016 0060 0016 0016 0016 0060 0016 0016 0016
0016 0016 0016 0016 0016 0016 0060 0016 0016 0016 0060 0016
0016 0016 0060 0016 0060 0016 0016 0016 0060 0016 0016 0016
0016 0016 0060 0016 0060 0016 0060 0016 0016 0016 0060 0016
0016 0016 0060 0016 0060 0016 0016 0016 0016 0016 0016 0016
0593
```

If you work out all of the detailed analysis in a manner similar to that shown for the Sony you should determine that the carrier frequency is indeed 40kHz, there are 34 total burst pairs in the one burst sequence used, and the burst sequence is repeatable. The actual 32 bits of data is: 00010101 11101010 01011000 10100111 Looking at the adjacent fields (1 & 2, 3 & 4) we see they are compliments of each other. A short way of checking for compliments is that ones become zeros and zeros become ones.

The device code as transmitted is 00010101. Reversing it we get the binary value 10101000. This is the decimal value of 128+32+8=168.

The function code is transmitted as 01011000. Reversing it we get the binary number 00011010. This is the decimal value 16+8+2=26.

This is the discrete Power On Code for the CLD 79.

Conclusion

Let me once again say thank you to all those who helped me with the deciphering of these codes. I used to do it for a living but that was for some government agency and that is a whole other story. All the help made it much faster and much more enjoyable. If you are interested in finding out more about IR codes search the WWW. Sci.Electronics FAQ is a good search parameter along with the word "IR code". An article by Scott Coleman of Xanadu consulting sheds a lot of light on the Sony Control-S protocol. An excellent article by Juergen Putger describes decoding IR remotes in general. Once you find a couple of them, they will have links to the others.

Appendix C.   Help and Support

Thanks for taking the time to read this document. We have tried to cover in easy-to-understand terms, every facet of automation the iScan supports – while attempting at answer every question we've ever been asked by customers and installers.

However – if after reading this document you have questions which are left unanswered, please call or email us to get an answer. We are located in California (U.S Pacific Time Zone), and run a phone call-center between the hours of 9AM and 5PM. Please remember that we follow Daylight Savings Time at our office. If you are unable to reach us with a phone call or it is not convenient to call us, we recommend that you send us an email.

Our Phone number within the US (Toll-free):   1-866-423-3836 extension 333

Our Phone number outside the U.S. (Toll):   1-(408)-395-4455 extension 333

Our Email address for Technical Support: help@anchorbaytech.com

Our Mailing Address:
              Anchor Bay
              983 University Ave.
              Building A
              Los Gatos, CA 95032

Free Manuals Download Website

http://myh66.com

http://usermanuals.us

http://www.somanuals.com

http://www.4manuals.cc

http://www.manual-lib.com

http://www.404manual.com

http://www.luxmanual.com

http://aubethermostatmanual.com

Golf course search by state

http://golfingnear.com

Email search by domain

http://emailbydomain.com

Auto manuals search

http://auto.somanuals.com

TV manuals search

http://tv.somanuals.com