EUTELSAT
Organisation européenne de télécommunications par satellite
European Telecommunications Satellite Organization
70, rue Balard — 75502 PARIS Cedex 15 — France

DiSEqC 1.0 DiSEqC 2.0

Digital Satellite Equipment Control (DiSEqC™)

APPLICATION INFORMATION FOR

TUNER-RECEIVER/IRDS

April 12, 1996

Reference Documents that define the DiSEqC System:

DiSEqC™ Bus Specification Version 4.2 (February 25, 1998)

DiSEqC™ Slave Microcontroller Specification Version 1.0 (February 25, 1998)

DiSEqC™ Logos and Their Conditions of Use (February 25, 1998)

Associated Documents:

Update and Recommendations for Implementation Version 2.1 (February 25, 1998)

Application Information for using a "PIC" Microcontroller in DiSEqC™ LNB and simple switcher Applications Version 1.0 (June 7, 1999)

Application Information for Tuner-Receiver/IRDs (April 12, 1996)

Application Information for LNBs and Switchers Version 2 (February 25, 1998)

Reset Circuits for the Slave Microcontroller (August 12, 1996)

Simple Tone Burst Detection Circuit (August 12, 1996)

Positioner Application Note Version 1.0 (March 15, 1998)

CONTENTS

1. Introduction

The DiSEqC system is intended firstly to provide a method of controlling a wide range of accessories located at, or near, the satellite-receiving antenna (dish), via a single "downlead" cable connected to the Tuner-Receiver, or Integrated Receiver Decoder (IRD). A further aim is to simplify the overall installation process, initially by recommending standardised protocols, and later by permitting automatic installation of accessories on the bus by using two-way communications.

This Application Note is concerned with the organisation of the embedded software and hardware in the Tuner-Receiver/IRD. It deals with how the user /installer might "set up" the system, and how the master Receiver can communicate with the slave accessories on the bus.

The specification of the carrier-tone and data-modulation which the master Receiver should apply to the satellite downlead cable are defined in the DiSEqC Bus Specification, currently Version 4.0. This also includes the present list of standard commands, although the list may be added to, or have deletions, as needs arise. To assist manufacturers, the listed commands are prioritised in their expected usage, from "Mandatory" downwards.

2. User (Installation) Interface

The User-Interface for a DiSEqC - Equipped Tuner-Receiver/IRD need not be very different to existing systems. There are potentially more parameters which need to be set up (e.g. to select a wider range of satellites), but since the DiSEqC system is potentially able to directly interrogate the peripheral devices (for parameters such as the Local Oscillator frequency), there is also the opportunity to reduce the amount of information requested from the installer or user. Furthermore, because DiSEqC commands can co-exist with established signalling methods (i.e. 22 kHz tone and 13/18 volt d.c. levels), it may not be essential for the user even to select the type of signalling employed.

2.1. Typical "Programme-Selection" Data Structure

Before outlining how the "Programme" (or "Channel") data might be handled in a DiSEqC Tuner-Receiver/IRD, an indication will be given of a typical existing data structure:

When each one of up to, say, 200 "Programmes" is to be viewed, the master Receiver processor (micro-controller) retrieves a block of data from a Non-Volatile Random Access Memory (NVRAM). This might be a battery-backed-up RAM or an Electrically Erasable, Programmable, Read-Only Memory (EEPROM), and is needed because the Programme data must be retained when no power is applied to the internal circuitry, or even to the whole Receiver.

The data retrieved from the NVRAM is a large number of data-bits often stored as bytes (groups of 8 bits), although at least some of these bytes need to be sub-divided back to the bit level. The exact structure of this data may vary widely between system designs, depending on the required facilities and the size of the available memory. However, typical data functions can be considered as:

- The Name of the Programme for On-Screen-Display
- A "Number" related to the Frequency that the Programme is transmitted on
- A "Number" related to the Frequency that the Sound is transmitted on (relative to the main carrier)
- Perhaps, further "Numbers" related to Polarisation (Skew) or dish (Positioner) movement
- A group of bits which define how to optimally "Set up" the Path of the signal from the Dish.

It is only the last of these which need be directly affected by introduction of DiSEqC, and even here the changes are only concerned with re-defining, or adding, a few data-bits used for control purposes, as outlined in the following sections.

2.2. Bus Control Flags

Included in the data stored for each Programme in a typical system are two bits, which act as control flags. One of these bits determines whether a 13 or 18 volts power supply is applied to the cable, and the other determines whether a 22 kHz tone is superimposed on this supply voltage. However, there is little consistency as to how the setting of these "flags" is presented to the user or installer.

The 13/18 volt level has a reasonably well established function of controlling whether an LNB is to accept the Vertically or Horizontally polarised component of the signal. Therefore, the user/installer can be presented with a menu option "Polarisation: H / V", and if he wishes to add a new Programme he should be able to find the necessary information, together with the carrier frequency, in published data. It is to be hoped that a professional installer would not be too concerned whether the menu option is presented as "Polarisation: H / V" or "Voltage: 13 / 18", etc.

The 22 kHz tone was originally intended to select between "Low Band" and "High Band" transmissions, i.e. the frequency of the local oscillator in the LNB. Thus the user-interface would be expected to control a functional option presented for example as "Band: Lo / Hi", which again could be presented in published data. However, the 22 kHz tone has been used for other purposes (particularly for operating a switch to select between signals from two satellites at different orbital positions), so users have been presented with a "transport" option selection such as "22 kHz: ON / OFF", rather than a "functional" option. The installation process has thus already become so inconsistent that it cannot be easily handled by reference to simple tables of transponder (Channel) information.

A fundamental aim of the DiSEqC system was to provide an additional control method so that selection could be made between Hi / Lo Bands and two different satellite orbital locations. All that is required to store this additional parameter is one more data-bit for each Programme. In a Tuner-Receiver/IRD with two IF signal inputs, this additional bit is already present, being used to select which input connector is used for each Programme. Since one of the aims of DiSEqC is to remove the need for more than one downlead and input connector, it may be possible just to re-allocate the control function of this data-bit.

In a DiSEqC system the "method of transport" of the commands is of no relevance to the user, so he can be offered three simple menu choices such as "Polarisation: H / V", Band: "Hi / Lo" and "Satellite Position: A / B" for each Programme. It was originally hoped that "Satellite A" could be rigidly defined as the more Easterly Position (e.g. Astra) and "Satellite B" as the more Westerly (e.g. Eutelsat). In practice, it appears that it is not realistic to achieve this in all applications, but it still remains a recommendation whenever practicable, e.g. for "dual-horn" LNBs.

To permit further expansion of satellite systems, a fourth control function has been included in the basic DiSEqC specification. Since its use has not yet been finalised, it is currently called the "Switching Option", or SW 0, with positions A and B. Thus, an additional installation flag can be offered to the user such as "Option: A / B". However, it is most likely that the two new switches will be used together to select between more than two satellite locations, so it is suggested that the user/installer is offered a multiple selection for satellite location such as "Satellite Position: 1 / 2 / 3 / 4" as shown in the following table:

"Option" switch	"Satellite Position" switch	"Satellite number"
A	A	1
A	B	2
B	A	3
B	B	4

Table 1: Four-input Switcher for use with Twin LNBs

2.3. Tone Burst (Satellite Position) Control

As a result of requests from the industry, a simple method of controlling an individual two-state switch has been devised. In principle, this should be called the "Satellite Position" selection switch, as above. However, if an existing system is being expanded, it is possible that this function already is performed by the continuous 22 kHz tone, so the new function is then "Hi / Lo Band" switching. Thus it seems that the menu-name presented to the user may have to be "transport-derived", such as "Tone Burst: A / B", rather than the true system function.

2.4. Combined (Existing + DiSEqC) Control

DiSEqC commands are able to co-exist on the same cable with the established voltage/tone switching signals (and the new Tone Burst). This means that the Tuner-Receiver/IRD can control both old LNBs and Switchers and new DiSEqC devices at the same time. It even may not be necessary for the user to configure the Tuner-Receiver/IRD for DiSEqC (or not DiSEqC) at the time of installation.

In principle the same system flags can be transmitted both in the DiSEqC commands and by the established voltage/tone commands. However, whilst the functions of the DiSEqC commands are rigidly allocated, the established control methods may differ between systems. In some cases, 13/18 volt levels have even been used for Hi/Lo Band switching (with a separate magnetic or mechanical polariser for H/V selection).

The fully versatile solution is to allow the user/installer independent control of all the existing commands and DiSEqC commands, but each Receiver/ LNB manufacturer must decide whether they wish to provide this level of sophistication. If this versatility is required, then the two types of command can be distinguished by using both the "transport" names and the "functional" names in a menu such as the following:

	Parameter	**Installation Options**
	"Voltage"	"13" / "18"
"Established" functions:	"22 kHz tone"	"OFF" / "ON"
	"Tone Burst"	"A" / "B" (unmodulated / modulated)
	"Band" (L.O.)	"Low" / "High"
DiSEqC functions:	"Polarisation"	"Vertical" / "Horizontal"
	"Position"	"1" / "2" / "3" / "4"
	"SMATV"	"0" (off) / "1" / "2" / . . . . / "16"

Table 2: "Combined" Switching Menu

2.5. SMATV Switching

In the past, some SMATV systems have used the normal voltage and tone control signals for switching between inputs, even though these inputs may not have been directly related to the equivalent functions in an LNB. This can still be done in the future, with the added DiSEqC benefit that up to 16-input switchers can be supported. Thus a SMATV user menu could offer up to 16 "inputs" (or 8 with the established tone/voltage commands) according to the following table:

"Input"	"Tone Burst"	"22 kHz"	"Voltage"	"Option"	"Position"	"H / V"	"Hi / Lo"
1	A	OFF	13	A	A	V	Lo
2	A	OFF	18	A	A	V	Hi
3	A	ON	13	A	A	H	Lo
4	A	ON	18	A	A	H	Hi
5	B	OFF	13	A	B	V	Lo
6	B	OFF	18	A	B	V	Hi
7	B	ON	13	A	B	H	Lo
8	B	ON	18	A	B	H	Hi
9	-	-	-	B	A	V	Lo
10	-	-	-	B	A	V	Hi
11	-	-	-	B	A	H	Lo
12	-	-	-	B	A	H	Hi
13	-	-	-	B	B	V	Lo
14	-	-	-	B	B	V	Hi
15	-	-	-	B	B	H	Lo
16	-	-	-	B	B	H	Hi

Table 3: "Established" and DiSEqC Switch Positions

DiSEqC also offers another set of commands which can be used for SMATV systems with up to 16 inputs. Since these commands, to the "Uncommitted" switches, are completely independent of those already described, they can be used in environments where there is the possibility of DiSEqC commands propagating through to devices such as LNBs, without any risk of inadvertently switching them.

In the foreseeable future, it seems unlikely that systems will need to use both "Committed" and "Uncommitted" commands at the same time, so the installer-interface might offer the choice of "Committed" or "Uncommitted" switching groups as a simple "flag" option. Note particularly with the "Port Group" commands (*see section 3.2.*), the difference between the Uncommitted and Committed commands is very simple, just employing a '39h' command byte in place of '38h', with the same data byte. In two-way DiSEqC systems the Tuner-Receiver/IRD software can automatically interrogate each slave Switcher to determine which group of commands it supports.

Where the DiSEqC Uncommitted commands are used, the switch numbers are related to the individual outputs according to the following table.

"Input"	"SW 4"	"SW 3"	"SW 2"	"SW 1"
1	A	A	A	A
2	A	A	A	B
3	A	A	B	A
4	A	A	B	B
5	A	B	A	A
6	A	B	A	B
7	A	B	B	A
8	A	B	B	B
9	B	A	A	A
10	B	A	A	B
11	B	A	B	A
12	B	A	B	B
13	B	B	A	A
14	B	B	A	B
15	B	B	B	A
16	B	B	B	B

Table 4: "Uncommitted" Switch Positions

2.6. Local Oscillator Frequency data

An important piece of data mentioned in section 2.1 is the "Programme Frequency", which is usually required in two forms. It is normally presented as a true numerical value (in MHz or GHz) in an On-Screen-Display Menu, and it is also needed to control the "tuning" of the receiver section. However, the numerical value required for tuning is not the same as the "published" value because the frequency has been offset by the Local Oscillator in the LNB. Furthermore, the offset is not constant for all systems. Not only is it different for Low and High Bands, but various Local Oscillator frequencies have been used as satellite broadcasting has developed.

All (versatile) Tuner-Receiver/IRDs thus require an OSD menu option to set up the LNB Local Oscillator frequency (or frequencies). For compatibility with existing LNBs this will have to remain in future systems for some time, but a two-way DiSEqC system is able to interrogate the LNB to determine the L.O. frequency. Thus there is the opportunity to make this menu option "Automatic" if the Tuner-Receiver/IRD software is able to obtain the necessary data directly from the LNB via the bus.

3. Use of DiSEqC Commands

Although the set of DiSEqC commands is relatively simple, it may not be clear exactly how errors are handled, or why there are several types of command capable of performing similar functions.

3.1. Error Handling

The error handling is fundamentally concerned with reporting functions which cannot be performed, or cannot be validated, rather than actions which may not be useful. Thus, if a command is sent to select Standby (command '02h'), then an error report ('E5h') is returned if the 'Standby' pin is connected to ground (indicating that there is no standby hardware installed). However, if standby hardware is available and a standby command is sent, then an 'OK' reply ('E4h') is returned even if it is already in the standby condition, so the command has no effect.

To avoid excessive software complexity, the Slave does not test the validity of individual bits within the Port Groups, even if individual pins are fixed high or low. However, by using additional commands, the Master can determine the state and status of all of the individual pins, before sending the actual command, and/or after the command is executed.

3.2. Writing to Port-Pins

There are three different types of command capable of controlling the individual "Committed" and "Uncommitted" Port-pins. These are the individual Port-bit 'set' and 'clear' commands ('20h' to '2Fh'), the Write Port-Group commands ('38h' and '39h') and the "Analogue" Write-Port command ('48h'). Although any of these commands may be able to produce a desired effect, it is strongly recommended to use the appropriate command for each application.

The "Write to Port Group" commands ('38h' for the Committed port and '39h' for the Uncommitted port) are the preferred commands, and are the only ones considered mandatory for inclusion in all (switching) Slave devices. Although splitting the data byte into two nibbles may appear rather complex, the actual application can be very simple. In practice, each time a new Programme (Channel) is selected by the user, a data nibble (4 bits) is fetched from memory (or built up from less bits in simple, memory-limited systems) to define the Committed parameters (or maybe the Uncommitted parameters in SMATV applications). Since all four bits are to be updated (it doesn't matter whether they actually change or not), the new data are written into the low nibble of the data byte and four '1's (i.e. 'Fh') written to the high nibble. This forces all the '0's and '1's to be transferred to the Port-Group pins. Thus it is possible to update up to four control lines with a single 4-byte command string (Frame + Address + Command + Data) in less than 60 ms.

The Write Port pin commands ('20h' to '2Fh') are only 3-bytes long (Frame + Address + Command), so if only a single port-line is to be changed, they are slightly more efficient, taking approximately 40 ms to send. However, if more than one pin-level is to be changed, they are less efficient than the Port-Group commands. It is likely that all Slave microcontrollers will implement these commands (probably converting them internally to the Write-Port format), so Tuner-Receiver/IRD software designers can use either or both forms of command to their personal preference.

The Write (complete) Port command ('48h') is basically intended for use when all 8 bits of the microcontroller port are to carry an "analogue" value on 8 parallel lines. The command could have been made "not valid" for controlling the port when used in a switching application, but the current view is that there may be uses in exceptional applications, yet to be defined. However, this command is **not recommended** for general use in signal-path switching applications.

4. Receiver/IRD Hardware Required to implement DiSEqC

The main requirement to implement simple "one-way" DiSEqC 1.0 in a Tuner-Receiver/IRD is the ability to switch on and off (modulate) the 22 kHz tone with a time resolution of about 100 is to be sent, typically when the viewer requests a change of Programme. Thus it may be convenient to allocate the resident microcontroller full-time to the message task, using timing loops to determine the DiSEqC bit periods, or even generating the 22 kHz cycles as well. Alternatively, it may be preferred to use interrupts to generate the tone-keying events, maybe with a small amount of dedicated hardware support.

For one-way DiSEqC the tone may be superimposed on the power supply voltage in the same way as the present 22 kHz continuous tone method, i.e. typically by modulating the transistor or regulator used to select the 13 / 18 volt levels for polarisation control.

4.1. Power-on (Reset) Requirements

The "Reset" circuits used in most DiSEqC slave accessories need to have a relatively short time constant to ensure that the microcontroller is functional within 100 ms of power being applied (for example by an intermediate Switcher during "channel changing"). This means that the supply voltage to the bus must rise reasonably rapidly, typically in less than 5 ms. Therefore, if the Tuner-Receiver/IRD uses a "slow start" main power supply, it may be necessary to ensure that the bus supply is switched on at a later time, when the main supply rail is fully established.

4.2. Two-Way DiSEqC 2.0 Implementation

Two-way DiSEqC requires the bus power supply feed to have a nominal source impedance of 15Ω (at 22 kHz), typically provided by a small inductor. Then, the supply modulation may be performed by a simple low-power shunt transistor, like that used in the slave circuits. Detection of the reply tone will generally need a simple transistor interface to produce 'TTL' levels for the controller hardware, or perhaps a slightly more elaborate tone-demodulator to reduce the processing load on the host microprocessor.

Since "replies" are only received from accessories immediately following a transmitted message, the host microcontroller need not be interrupt-driven, but can wait to detect the reply modulation (or demodulate the carrier itself, like the slave microcontroller) in real-time. However, since it may be necessary to wait for up to 150 ms for certain replies (or to discover that a reply is not being sent because of an error), the use of an interrupt may be preferred. Because the first five bits of current DiSEqC messages do not carry any information (except contributing to the parity calculation and framing) it may be possible to compensate for a slow interrupt response time by using a more sophisticated message-recognition process.

5. Typical One-Way (DiSEqC 1.0) Protocol

One-way DiSEqC operation is necessary if the slave accessories do not support reply-signalling, or if the master Tuner-Receiver/IRD is unable to detect the reply modulation. In this section, the protocol where the master is the limiting factor will be considered. The situation when one or more slave accessories are only "one-way" (or are only voltage/tone controlled) must be handled as a "fall-back" of the two-way master protocol, as discussed in Section 6.

In a one-way DiSEqC system, the master Tuner-Receiver/IRD is unable to obtain any installation information from the slave accessories, nor can it confirm what effects its commands are producing (except very indirectly by user or signal-strength feedback). Therefore, some degree of user or installer setting-up may be required to indicate the type of accessories (whether DiSEqC or not) which are present on the bus. However, the ability of DiSEqC devices to operate in a mixed "backwards compatible" environment, and the use of "wildcard" (or "don't care") device addressing, should allow the installation procedure to be kept to a minimum.

Although the full hardware configuration of a DTH or SMATV system may take many forms, the Tuner-Receiver/IRD should be able to handle most requirements with about five basic command structures. Because these structures are basically hierarchical, the "top-level" version (*section 3.2.*) can also support the others, so it could be used as a "default" to simplify the installation process. However, the transmission of unnecessary commands obviously places an additional loading on the master controller microprocessor, and may increase the time that the system takes to Programme-change, so manufacturers may prefer to provide a user-selection for the command structure. The basic structures are described in ascending sophistication in the following sections.

5.1. Pure "One Stage" DiSEqC System

This configuration applies when only true DiSEqC slaves (not DiSEqC Tone Burst) are present on the bus, and all remain continuously powered whilst the system is in operation (i.e. no loop-through switching). Since most switchers are expected not to feed power through to non-selected inputs, this control protocol applies mainly to single LNBs (including "multiple horn" monoblocks), and SMATV switches where the LNBs have a fixed "personality", or an auxiliary power supply.

The command protocol can be very simple because only one DiSEqC message is needed to set up almost any signal path (i.e. up to 16 combinations of either the 4 Committed or 4 Uncommitted control lines). Since there is no need to distinguish between different switch addresses on the bus, the "wildcard" address ’10h’ (all LNBs Switchers and SMATV) can be used. In general there will be only one DiSEqC slave on the bus, but even if there are more, there is no risk of replies "colliding" because a reply is not (and must not be) requested if the master only supports one-way DiSEqC communications. If there is more than one slave on the bus, then each will act on the appropriate control bits within the message (providing that every one is already powered before the message starts). A minimum sequence to control the system could be as follows:

After the Receiver/IRD has powered up the bus (> 100 ms delay) :

"Standby Off" : **E0 10 03** (No reply, address 10h, command 03h)

and whenever a Channel (Programme) is changed :

"Port Group data X" : **E0 10 38 FX** (As above, command 38h, data FX)

Where the data byte, FX, takes the value from the "Port Group" column in the following table:

"Switch Number"	"Switch Option"	"Satellite Position"	Polarisation	Frequency	"Port Group" data byte
1	A	A	V	Lo	F0
2	A	A	V	Hi	F1
3	A	A	H	Lo	F2
4	A	A	H	Hi	F3
5	A	B	V	Lo	F4
6	A	B	V	Hi	F5
7	A	B	H	Lo	F6
8	A	B	H	Hi	F7
9	B	A	V	Lo	F8
10	B	A	V	Hi	F9
11	B	A	H	Lo	FA
12	B	A	H	Hi	FB
13	B	B	V	Lo	FC
14	B	B	V	Hi	FD
15	B	B	H	Lo	FE
16	B	B	H	Hi	FF

In most of the data bytes above, the first nibble 'F' actually contains one or more "don't care" bits (because they are over-written by a '1' bit in the second nibble), so the first nibble may validly take alternative values. However, the consistent use of the 'F' seems the simplest structure to program. Note that the "bit format" command bytes shown in the final columns of Tables 10 and 11 of the Slave Microcontroller Specification Versions 0.1 and 0.2 are not correct and will be corrected in future issues.

If at the installation stage, the user/installer had selected control of the "Uncommitted" switches, then the command '38h' above would be replaced by '39h'.

If there was a particular desire to economise on transmission time, then sometimes the 3-byte commands, '20h' to '2Fh', could be used (*see section 6.1.*). However, this is not generally recommended because of the additional master software complexity and the inability to verify command success or slave switch status.

5.2. Multi-Stage (Loop-through) DiSEqC System

This operating mode applies not only to "Loop-through" LNBs, but any switching system where there is a "distant" DiSEqC Slave which only receives power (and/or commands) when a "nearby" Switcher selects it as a source. In this case, the transmission of a DiSEqC command may cause the "nearby" switch to operate, which then applies power to the "distant" slave, which cannot have seen the command. Therefore, the command needs to be repeated after not less than 100 ms, when the "distant" slave has had time to power-up and initialise.

It is probably not necessary to consider multiple loop-through configurations, but the master could transmit the command message for a third time, which would also give some error-recovery capability if a command had been missed. Furthermore, if no "backwards-compatible" signalling is required, it even may be appropriate to continually repeat the same message, provided that the master microcontroller has sufficient processing resources.

5.3. Combined (Backwards-Compatible) DiSEqC System

A DiSEqC Switcher may be used to select between I.F. signals from two or more existing (22 kHz/voltage) LNBs. In this case, a DiSEqC command is sent first (at least 10 ms after terminating any previous 22 kHz continuous tone), and then after a pause of 10 ms the 22 kHz tone (and the appropriate 13/17 volt level) may be transmitted continuously to select the appropriate function from the LNB.

Because of the slave microcontroller's backwards-compatible capability, to be explained in the next section, it is important in this mode that the "messages" carried in the DiSEqC flags and the tone/voltage levels are consistent with each other. Thus, the master Tuner-Receiver/IRD must "know" how the continuous 22 kHz tone will be interpreted (i.e. as Hi/Lo frequency , or Satellite Position switching). This is necessary because when a backwards-compatible DiSEqC LNB is first powered up (e.g. by operation of a "nearby" Switcher), the LNB is controlled by the tone/voltage levels. However, any subsequent DiSEqC message which does not cause the Switcher to change-over, will be received by the LNB and used in preference to any subsequent tone/voltage levels.

5.4. DiSEqC Switcher with Mixed LNB types

It is possible that many DiSEqC LNBs (at least during the introductory phase) will be fully backwards-compatible with the established switching signals, i.e. they will respond to continuous tone and voltage levels on the bus. Thus, they may be controlled by the signalling described in Section 5.3. However, this is not a satisfactory general method because backwards-compatibility is only a short-term characteristic which cannot be relied on, for two reasons:

Firstly, the DiSEqC slave microcontroller must abandon the backwards-compatible mode as soon as it recognises a valid DiSEqC command. This is essential because otherwise the normal operating state of the bus (i.e. a voltage of typically 12 volts and no continuous 22 kHz tone) would be interpreted as Vertical polarisation and Low frequency band (or satellite ’A’) and could over-ride any previous DiSEqC message.

Secondly, the 13/17 volt signalling method carries cost and power-dissipation penalties for most components in the system (certainly for LNBs and Receivers), so manufacturers may wish to abandon it as soon as practicable. Generation of the continuous 22 kHz tone also requires some dedicated hardware in the Tuner-Receiver/IRD, whereas it is possible to generate short DiSEqC messages purely with real-time software in the master microcontroller.

Therefore, the preferred method for controlling a system which may contain both DiSEqC and tone/voltage LNBs is to combine the protocols described in Sections 5.2 and 5.3. Thus the basic DiSEqC switching data-burst should be transmitted two or three times, spaced by at least 100 ms, and then the continuous tone and voltage levels as appropriate for the LNB.

5.5. Inclusion of "Tone Burst" Commands

To provide a method of controlling a simple additional two-state switch, a "Tone Burst" signal has been added to the DiSEqC specification. The Tone Burst is described in the DiSEqC Bus Specification 4.0 onwards, and has a nominal duration of 12½ ms. Normally, it will be inserted after the last DiSEqC message of any Programme-change sequence, and before the continuous tone (if relevant), with a minimum of 15 ms "quiet" pauses each side of the Tone Burst.

Because the transient behaviour of continuous tone Switchers, and to a lesser extent Tone Burst Switchers, is not predictable when full DiSEqC messages are transmitted, it is not recommended to use these types of switch to select between DiSEqC LNBs. Clearly, if any Switcher changes position (even for a brief period), it will corrupt the transmission of any DiSEqC message to any more "distant" slave on the bus. Although it is possible to design continuous tone and Tone Burst detectors which will ignore true DiSEqC messages (by using long time constants, modulation detectors and/or time-gating) it is unlikely that existing devices, or those designed to minimum costs, will meet the required criteria.

The best recommendation which can be given to Tuner-Receiver/IRD manufacturers who wish to minimise the risks of unsatisfactory behaviour with these simple Switchers, is to always change (and then restore, if necessary) the position of tone and Tone Burst switches, whenever a Programme-change is requested. This should ensure that any DiSEqC slaves are reset (by power-down) and thus respond to backwards-compatible signalling.

6. Typical Two-Way (DiSEqC 2.0) Protocol

Most of the protocols described in Section 5. are applicable to a full two-way DiSEqC system. The main difference is that when the master Receiver/IRD requires a Reply, the first (framing) byte of the DiSEqC command must have the 'Reply' flag set (e.g. 'E2h'), and the address in the second byte should be specific to only one slave on the bus, so "wildcard" addressing must be used with more caution. However, the master Tuner-Receiver/IRD can determine the true addresses of all slave accessories on the bus during an initialisation or installation procedure as outlined in Section 7.

The Reply facility can be used in various ways. The simplest application is just as a confirmation that a command has been received by the slave accessory. Since the messages in both directions include parity checking, the receipt of an "OK" (reply 'E4h') gives a good validation that the command was received successfully by the slave. If no reply, or an unexpected error reply ('E5h', 'E6h' or 'E7h'), is received, then the command generally should be repeated once.

In "loop-through" DiSEqC applications (which includes most separate Switchers), where the master Tuner-Receiver/IRD has to wait for at least 100 ms for the switch to operate and the "distant" accessory to become active, the master might choose to interrogate the "nearby" switch position for verification whilst it is waiting. An alternative strategy, particularly where it is beneficial to set up the complete path as soon as possible, would be to transmit a "global" command (e.g. defining all Committed switch positions, to the wildcard address, but not requesting a reply), and then interrogate the status of each slave accessory in turn to verify that it has operated correctly.

Some typical two-way command protocols are outlined in the following subsections:

6.1. Two-position DiSEqC switch with tone/voltage LNBs

If there is only one DiSEqC slave on the bus, the master could use "don't care" addresses such as '00h' (all slaves) or '10h' (all LNBs, switchers and SMATV). However, ideally it should use a specific Switcher address such as '14h' or '15h' (the automatic- or user-installation procedure can determine which). The switch control line probably would be the "Satellite Position B/A", but the master can determine this also at installation (e.g. by reading the "switching resources" registers with commands '14h' and '15h'). It is unlikely that a simple Switcher would have a "Standby" (power-down) facility, but this also could be determined at installation (e.g. by commanding "Standby" or "Power on" and waiting for an OK/error reply).

When the Receiver is switched on it could use the following commands:

"Clear Reset Flag": **E2 10 01** Reply: **E4 (OK)**

This ensures that all replies will be returned without delay (i.e. after approximately 10 ms). This command should only be used after the master is sure that there are no other slaves on the bus with a corresponding address.

Unless it knows that the slave has no Standby facility it should send:

"Power On" : **E2 10 03** Reply: **E4** (OK) or **E5** (no standby)

Then, each time it wishes to change the switch position it can use commands:

"Set Position A"	**E2 10 22**	Reply: **E4** (OK)
"Set Position B"	**E2 10 26**	Reply: **E4** (OK)

OR the "Port Group" commands:

"Set Position A"	**E2 10 38 40**	Reply: **E4** (OK)
"Set Position B"	**E2 10 38 04**	Reply: **E4** (OK)

The "OK" reply should be adequate confirmation, but the Committed switches status could be requested with:

"Return Port status": **E2 10 14**	Reply: **E4 04** (OK, all Committed pins low (0), "Position" available (4))
if 'Position A', or if 'Position B':	Reply: **E4 44** (OK, "Position" pin high (4), "Position" available (4))

If it is an "Options" Switcher, then the short commands would use **23** / **27** in place of **22** / **26** and the data nibbles would be **8** (bit.3) in place of **4** (bit.2).

If the LNBs are 13/18 v.- 22 kHz types, the DiSEqC command(s) are followed by the continuous tone and/or a change in voltage level to select the required characteristic of the LNB as shown in section 5.3 of the DiSEqC Bus Specification version 4.0.

6.2. Four-position DiSEqC switch with tone/voltage LNBs

A four-position switch is almost exactly the same as the two-position type. The switch would probably use the four permutations of the "Satellite Position" and "Options" control lines. The Tuner-Receiver/IRD could send 1 or 2 individual commands for each control line (it need only change the lines which are wrong), but it is probably simpler to use a single "Port-Group" command. The data for the two command types are as follows:

LNB number	Individual command #1	Individual command #2	"Port Group" command
1	E2 10 22	E2 10 23	E2 10 38 C0
2	E2 10 26	E2 10 23	E2 10 38 C4
3	E2 10 22	E2 10 27	E2 10 38 C8
4	E2 10 26	E2 10 27	E2 10 38 CC

6.3. "Dual-Horn" LNB or Monoblock

The 3 control lines for this type of monoblock are the Hi/Lo (frequency), H/V (polarisation) and SB/SA (satellite position), and can be controlled according to the following table. Like the 4-position switch, it is possible to use (up to 3) individual commands, but the single "Port Group" command should be much simpler to implement:

Sat. Pos.	Pol.	Freq.	Individual command #1	Individual command #2	Individual command #3	"Port Group" command
A	V	Lo	E2 10 22	E2 10 21	E2 10 20	E2 10 38 70
A	V	Hi	E2 10 26	E2 10 21	E2 10 24	E2 10 38 71
A	H	Lo	E2 10 22	E2 10 25	E2 10 20	E2 10 38 72
A	H	Hi	E2 10 26	E2 10 25	E2 10 24	E2 10 38 73
B	V	Lo	E2 10 22	E2 10 21	E2 10 20	E2 10 38 74
B	V	Hi	E2 10 26	E2 10 21	E2 10 24	E2 10 38 75
B	H	Lo	E2 10 22	E2 10 25	E2 10 20	E2 10 38 76
B	H	Hi	E2 10 26	E2 10 25	E2 10 24	E2 10 38 77

If this is a full DiSEqC system, then the Tuner-Receiver/IRD can interrogate the LNB (at installation) to determine the Local Oscillator frequencies. Although there are 4 different "frequency" commands defined in the Bus Specification, the one recommended to use is '51h', "Return present frequency". The following command sequence could be used:

"Set Low frequency": **E2 10 20** Reply: **E4** (OK)

"Return present frequency": **E2 10 51** Reply: **E4 02** (OK, table entry 2)
OR : **E5** (not supported)

"Set High frequency": **E2 10 24** Reply: **E4 (OK)**

"Return present frequency": **E2 10 51** Reply: **E4 04** (OK, table entry 4)

6.4. SMATV Switcher

A SMATV system can be exactly the same as the monoblock in the previous section (except that frequency data will not generally be available). Thus, simple Tuner-Receiver/IRDs should not need to distinguish between these two configurations. The full allocation of Input Numbers to pin levels is defined in the table in *section 5.1*. Note how the last digit (nibble) of the Port Group command relates very simply to the switch input number (minus 1).

It is recommended that SMATV installations of this type should allocate the Hi/Lo, H/V and Satellite Position A/B signals in the sequence as shown, so that Programmes may be "pre-installed" in Tuner-Receiver/IRDs.

If a SMATV installation has a less well-defined mapping of input signals to input numbers (perhaps with more satellites, or other frequency bands, etc.) then the preferred approach is to use the "Uncommitted" switching commands. With these, the installer can select up to 16 inputs in any way he chooses, and the Port Group commands (in this case to the SMATV address '18h') would be:

Input Number	"Uncommitted Port Group" command
1	E2 18 39 F0
2	E2 18 39 F1
3	E2 18 39 F2
4	E2 18 39 F3
5	E2 18 39 F4
6	E2 18 39 F5
7	E2 18 39 F6
8	E2 18 39 F7
9	E2 18 39 F8
10	E2 18 39 F9
11	E2 18 39 FA
12	E2 18 39 FB
13	E2 18 39 FC
14	E2 18 39 FD
15	E2 18 39 FE
16	E2 18 39 FF

6.5. "Loop-through" switched systems

It still has to be finalised exactly how "Loop-through" accessories will be implemented, but the current proposals are outlined below:

When power is applied to a Loop-through LNB (or a reset command is received) it selects its own "internal" source and does not feed power (or receive any IF) to the loop-through input. The master (Tuner-Receiver/IRD) can thus "talk" to it without any risk of another device responding as well. When the master "knows" that a loop-through device is present (e.g. by reading the flag in the status register at first installation), it changes the address to a special reserved address such as '71h'. Then, once this first device has been initialised, the master commands it to select its loop-through input and the master then interrogates the next device on the bus.

This structure means that "distant" devices do not receive any power until they are selected, and so must be initialised each time the "nearby" device switches. To keep the response time of the "distant" device within reasonable limits, it is recommended that slaves should be fully operational within 100 ms of power being applied. In addition, new simple "address" commands ('07h' and '09h') have been introduced to reduce the number of instructions which have to be sent. Thus a typical command sequence might be:

When power is first applied to the Bus :

"Return address":	**E2 10 07**	Reply: **E4 12** (OK, address 12h)
"Clear reset flag":	**E2 12 01**	Reply: **E4** (OK)
"Change address to 71h":	**E2 12 09 71**	Reply: **E4** (OK)

When the Tuner-Receiver/IRD wants to select the "distant" LNB:

"Select satellite position B":	**E2 71 26**	Reply: **E4** (OK)

If the "distant" LNB is a voltage/22 kHz type, then the appropriate continuous levels can be sent, but if it is a DiSEqC device then it is necessary to wait about 120 ms for the "distant" device to initialise and then send, for example:

"Clear Reset Flag":	**E2 10 01**	Reply: **E4** (OK)
"Select Hi L.O., Horizontal":	**E2 10 38 33**	Reply: **E4** (OK)

and if it has a standby facility:

"Power On" :	**E2 10 03**	Reply: **E4** (OK)

Note that the address of the first "nearby" device only has to be changed when power is applied to the system (and strictly only if a "distant" device may have the same address). It only would be necessary to change the address of a "distant" device during Programme-changing if both the second and a third device had the same address (which should be very unlikely).

6.6. DiSEqC switch selecting DiSEqC LNBs

This is similar to that in Sections 6.1 and 6.2, but the Switch is followed by one or more DiSEqC LNBs, instead of voltage/22 kHz LNBs. The additional consideration is that the LNBs need time to initialise when they have their power restored after disconnection.

If the switch has an address ’14h’ and the LNBs each have an address ’11h’, then typical control sequences if the Satellite Position switch has to be changed-over could be, for example:

"Select satellite position B": **E2 14 26** Reply (from switch): **E4 (OK)**

Wait >100 ms for the LNB to initialise, then:

"Clear Reset Flag": **E2 11 01** Reply: **E4 (OK)**

"Select Hi L.O., Horizontal": **E2 11 38 33** Reply (from LNB): **E4 (OK)**

If the Satellite Position switch is not changed-over then the only command required is, for example :

"Select Hi L.O., Horizontal": **E2 11 38 33** Reply (from LNB): **E4 (OK)**

Note that the DiSEqC slave microcontroller is fully backwards-compatible, so a DiSEqC LNB can be operated by 13/18 V. and 22 kHz tone commands exactly as in Section 6.1, provided that no DiSEqC commands are addressed to it after the power-on or a software reset.

7. Automatic Installation

It is too early to be able to predict the "best" strategy for a Tuner-Receiver/ IRD to interrogate the bus, to find out what accessories are present, so a general indication of a possible procedure is as follows:

The master needs to be able to "talk" to each device separately, so it must find out the address of every slave. There are two possible techniques:

1. The master can send a command to each address in turn and wait to see if there is a reply. A typical command would be to switch "Standby" off or on, or to request the "Configuration Register" data. When a slave responds, then the master knows its address, and any additional data can be requested as required. This process should not take too long because the master only needs to test addresses which it is able to support, initially perhaps some of the LNB/Switcher family such as '11h', '12h', '14h', '15h' and '18h'.

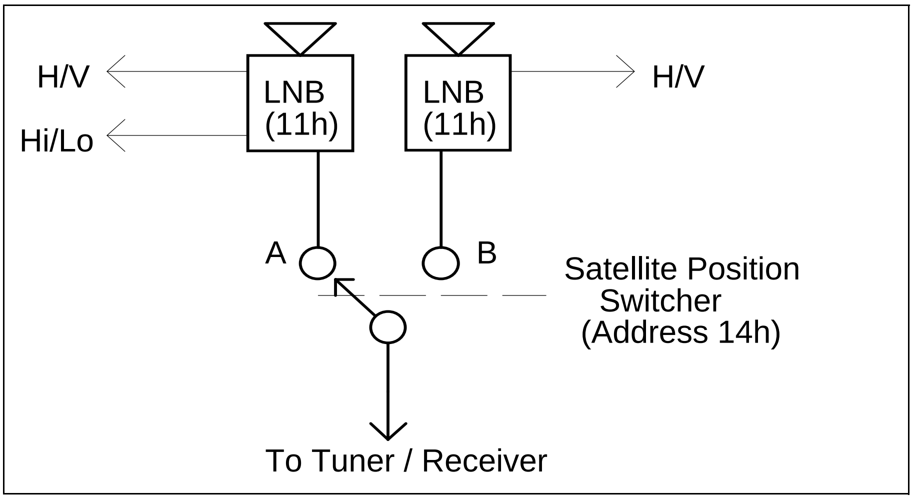
2. The master can send a "Return address" command ('07h') to a "wildcard" address such as '10h' or even '00h'. The slave microcontroller is designed to initially wait a random time of between approximately 15 and 115 ms before sending a reply, so one device should start replying first. All slaves "listen" to the bus before replying, and any which lose the race do not attempt to transmit, but set their internal "contention" flag.

The master should now know the address of one slave on the bus and can interrogate it for any data it requires. In the unlikely event that two replies do collide, they will almost certainly corrupt each other and the master can try again.

Although the master could use the "Return address if contention flag set" command ('05h'), there may be more than one other device on the bus, so it is better to use a more general procedure. The slave which has already responded needs to be "put to sleep" either by allocating it a reserved address (outside the range of the "wildcard" address), or by setting its contention flag (with command '06h') after all the others have been cleared by a "global" command '04h' (**E0 00 04**). The interrogation can then continue with the remaining slave(s).

After using one of the above procedures, the master should know about all the slaves at present on the bus. However, it then needs to determine whether there are any more accessories on the far side of switches which are open, and whether any of the slaves it has found will "disappear" (and probably be reset by losing power) when any of the switches are operated. It thus needs to repeat the interrogation procedure with all switches in each of their possible positions. This includes all LNBs with loop-through capability, identified by the flag in their "Status Register", or with a modified address ('12h' instead of '11h' in the case of the DiSEqC slave microcontroller software version 0.2).

A typical interrogation procedure for a Switcher + two LNBs, as shown in the following diagram, could be:



"Return Resources" (address 18h)	**E2 18 14**	Reply: none
"Return Resources" (address 15h)	**E2 15 14**	Reply: none
"Return Resources" (address 14h)	**E2 14 14**	Reply: **E4 04** ("Position" switch available, 'A' selected)
"Clear Reset flag" (address 14h)	**E2 14 01**	Reply: **E4** (OK)
"Return Uncommitted Resources" (address 14h)	**E2 14 15**	Reply: **E4 00** (No switches available)
"Return Resources" (address 12h)	**E2 12 14**	Reply: none (no loop through LNB)
"Return Resources" (address 11h)	**E2 11 14**	Reply: **E4 03** (LNB with Hi/Lo & H/V available, currently Lo & V selected)
"Clear Reset flag" (address 11h)	**E2 11 01**	Reply: **E4** (OK)

Request frequencies as in section 6.3

"Select Position B" (address 14h)	**E2 14 26**	Reply: **E4** (OK)
Wait >100 ms		
"Return Resources" (address 11h)	**E2 11 14**	Reply: **E4 12** (LNB, H/V switchable, Fixed Hi, V selected)
"Clear Reset flag" (address 11h)	**E2 11 01**	Reply: **E4** (OK)
Request frequencies as in section 6.3		
"Select Position A" (address 14h)	**E2 14 22**	Reply: **E4** (OK)
Wait >100 ms		
"Return Status Register" (address 11h)	**E2 11 10**	Reply: **E4 01** ("Reset" flag indicates that power has been lost)
"Return address if contention" (any address)	**E2 00 05**	Reply: none (No duplicate addresses have caused replies to conflict)