



Digital Satellite Equipment Control (DiSEqC™)



POSITIONER APPLICATION NOTE

VERSION 1.0

March 15, 1998

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	1
2. General Application Considerations	1
2.1. Power Availability	2
2.2. Direction of Antenna-shaft Rotation	3
2.3. “Endstop” Operation	3
2.4. Mechanical Alignment	4
3. Positioner Commands	5
3.1. “Halt Motor”	6
3.2. “Disable (soft) Limits”	7
3.3. “Set East (soft) Limit”	7
3.4. “Set West (soft) Limit”	8
3.5. “Drive Motor East”	8
3.6. “Drive Motor West”	9
3.7. “Store Satellite Position”	9
3.8. “Goto Satellite Position”	10
3.9. “(Re-) Calculate Satellite Positions”	10
3.10. “Goto Angular Position (degrees)”	11
3.11. Read Positioner Status Byte (Level 2.2)	13
4. User Interface	13
4.1. Typical “Installer” Command Menu	15
4.2. Typical “User” Command Menu	16
5. Positioner Units Embedded Software	17
6. Contact Details	20

1. Introduction

The DiSEqC™ system provides a method of controlling a wide range of accessories usually located at, or near, a satellite-receiving antenna (dish). It uses the existing I.F. coaxial cable (“downlead”) which connects to the Tuner-receiver, or Integrated Receiver Decoder (IRD). This Application Note is concerned with the definition and application of a group of commands, for the control of a “Positioner” Motor Unit, which can automatically rotate the receiving antenna to individually point at one from a range of satellites.

The specification of the basic hardware characteristics, and the method of signalling on the satellite downlead cable are defined in the *Bus Functional Specification Version 4.2*. This version includes a modified list of “Positioner Commands”, which have been adapted to suit “one-way” operation on the Bus. These commands allow the Tuner-receiver/IRD to control the Positioner without any necessity to receive reply or data messages back from the remote motor unit. However, implementation of reply messages can give more reliable operation and simplify installation.

A new DiSEqC™ Implementation Level (level 1.2) has been introduced to support Positioner applications. Tuner-receiver/IRDs must be able to transmit at least the basic set of “mandatory” commands for Positioners to qualify for using the “1.2” Logo. Positioner Motor Units may use the “1.2” Logo if they support the basic positioner commands, and the “2.2” Logo if they return appropriate “Reply” messages when requested by the Master (Tuner-receiver/IRD).

2. General Application Considerations

A minimum DiSEqC™ Positioner system need consists only of a DiSEqC™ Level 1.2 Tuner-receiver/IRD connected at the lower end of the I.F. cable, and a DiSEqC™ Positioner Motor Unit fixed between the satellite antenna (dish) and the antenna mount. The Positioner Motor Unit has two interconnected (unswitched) connectors, one for the downlead and the other for a short cable to the LNB. Thus, I.F. signals pass down from the LNB to the receiver via the Motor Unit. Power and DiSEqC™ (and/or backwards-compatible) control signals pass from the Tuner-receiver/IRD up to, and through, the Motor Unit to the LNB.

Normally, to give backwards compatibility, Tuner-receiver/IRDs will transmit a 13/18 volt d.c. level on the bus in concord with the Polarisation data carried in the DiSEqC™ messages. However, at least for DiSEqC™ Implementation Levels above 1.0, there is an advantage if the d.c. level can be selected independently of the Polarisation. This need only be a “global” option, i.e. the d.c. voltage either follows the Polarisation, or is always 13 volts or always 18 volts, for all channels. For SMATV applications there is a preference to fix the bus d.c. voltage low (i.e. approximately 13 volts) to economise on power dissipation. However, for Positioner applications it may be preferable to use a higher voltage (i.e. approximately 18 volts) to make more power available to the motor, so that it can drive more quickly to the required position.

2.1. Power Availability

DiSEqC™ Tuner-receiver/IRDs are required to be able to deliver at least 500 mA d.c. onto the bus to power all the peripheral components. Typically there may be a maximum current of about 350 mA available to power the Positioner Motor Unit. Motor units intended for this basic type of installation should thus incorporate circuitry to ensure that the current taken from the Bus does not exceed about 400 mA. even when the motor is stalled.

A current limit of 400 mA obviously places a restriction on the speed that the motor unit can move the antenna dish, so alternative configurations may be worthy of consideration. Although the Positioner Motor Unit could remove power from the LNB while traversing, this would only make a small amount of additional current available, and is not recommended. It would make optimisation of the signal strength (manual or automatic) difficult and requires that DiSEqC™-only LNBs must be sent new status (switching) commands when their power is restored.

One solution is to make an additional power feed available to the motor unit. Probably this would be derived from a local (e.g. roof-space) mains (line) power supply. It would need some additional cabling, but not generally as much as is required to route normal Positioner control cables down to the location of the Tuner-receiver/IRD.

A more versatile approach is to apply a “d.c. Power Booster” somewhere in the download, perhaps close to the Tuner-receiver/IRD. Such a booster would take power from a local mains (line) supply and make additional current available to pass up the cable to the Positioner Motor Unit. The absolute minimum requirement for such a booster is to be “transparent” to I.F. signals passing down the cable (from the LNB) and 22 kHz modulated (DiSEqC™) signals passing up the cable. However, to avoid application problems in the field, it is strongly recommended that any such unit should also support “backwards-compatible” 13/18 volt (polarisation) signalling from the Tuner-

receiver/IRD to the LNB and DiSEqC™ “Reply” messages back from the LNB and/or Positioner Motor Unit to the Tuner-receiver. Note that if currents significantly higher than 500 mA are to be passed up the cable, then it may be necessary to include some (adjustable) pre-compensation to correct for the voltage drop across the resistance of the cable.

Another possibility is to apply a degree of “voltage boosting” on the cable, perhaps up to the 36 volts commonly used by separately-wired positioner systems. In this case the motor unit must drop the voltage to its “LNB” connector back to the normal d.c. level, and various techniques could be used to reproduce the backwards-compatible 13/18 volts signalling, if required. However, it is important that the Positioner Motor Unit, Booster Unit and the user-documentation must carry clear labels that normal DiSEqC™ and other accessories are only to be connected onto sockets marked “LNB”.

2.2. Direction of Antenna-shaft Rotation

For clarity in the DiSEqC™ specification, the direction of movement of antenna Positioners is described as “East” or “West”. In the northern hemisphere Westerly movement corresponds to clockwise rotation of the antenna drive shaft when viewed from the top. For use in the southern hemisphere the equivalent direction of motor rotation would be anti-clockwise, so it will be necessary to modify the embedded software or to reverse the motor connections. If a Positioner Motor Unit is intended to be used in either hemisphere then the preferred solution is to fit a change-over switch, but the provision of alternative alignment tables or charts may be adequate.

2.3. “Endstop” Operation

It is important that Positioner Motor Units should not damage themselves, or any part of the mount or antenna, if instructed to drive continuously in either direction (even under fault or error conditions). If the power available to the motor is limited to that defined for the standard DiSEqC™ Bus (i.e. about 6 watts) then there is unlikely to be sufficient torque to cause damage. If it has a particularly high gear ratio, then it may be beneficial to include some form of slipping clutch mechanism, or limiting mechanism as described below. It is better to locate the clutch in the drive train between the motor and the shaft-rotation detector, so that the system alignment (i.e. the satellite positions) is not disturbed if the clutch slips.

If additional power is available to the motor unit, then an active method of preventing damage to the system may be required. This normally takes the form of moveable switches (e.g. micro-switches) which can be set to disconnect power before the antenna or internal drive mechanism reaches any physical limit such as a wall or the end of a “leadscrew”. These “Hard Limit Switches” are adjusted when the complete antenna and positioner motor assembly are being installed on their mountings.

Another form of endstop is “Soft(ware) Limits” where the positioner motor control system stores two positions (the East and West Soft Limits) past which it will not normally drive the motor. These may be beneficial to prevent the motor unit driving the antenna far away from the normally-used arc of satellites, due to a user or a system error (e.g. a corrupted “Halt” command). To allow these Soft Limits to be changed, they can be “disabled” with a software command, so that the motor can be made to drive past them. It is suggested that disabling the soft limits should require a specific user-action (e.g. confirming the requirement), but still they should not be relied on as the sole method of protecting against damage. This is because the motor unit and antenna normally will not be in the sight of the user when the soft limits are being adjusted.

2.4. Mechanical Alignment

The rotational axis of the Positioner Motor Unit output shaft needs to be aligned accurately with the Earth’s polar axis. Initially, this can be done with a spirit level (or plumb line) and a magnetic compass to find due south, in association with data related to the site latitude and longitude. For smaller dishes, particularly over a fairly small arc of the sky, this may be sufficient. However, for larger dishes sweeping almost from horizon to horizon the alignment needs to be to a fraction of a degree. This requires trimming the tilt adjustment at several positions around the full arc to find the maximum signal strength of appropriate satellites. A method of speeding this up for the professional installer is to release the coupling between the output shaft and the motor so that the dish can be manually swung from horizon to horizon.

An alternative approach is to allow small adjustments of the tilt (elevation) of the antenna axis to be made remotely with a small motor (or, of course, a true azimuth-elevation mount system can be used). This is supported in the DiSEqC™ system by defining two addresses for commands to Positioner motors. Address ‘31’ (hexadecimal) is for the normal equatorial rotation or azimuth, and address ‘32’ (hexadecimal) for trimming the tilt, or the “elevation”, of the antenna. More details are given in *section 3.* and *section 4.*

3. Positioner Commands

Like all DiSEqC™ command messages, the Positioner control messages consist of at least three bytes, a ‘Framing’ byte, an ‘Address’ byte and a ‘Command’ byte. Some commands have one or more ‘Data’ bytes to transfer additional information in the message. Each byte has an associated (odd) parity bit for error-detection purposes.

The Framing byte is normally ‘E0’ (hexadecimal) if the master (Tuner-receiver/IRD) does not require (or is not able to receive) a “Reply” message. It is usually ‘E2’ to request a Reply.

The normal Address byte for Positioner motors which drive on a polar, or azimuthal, axis is ‘31’ (hexadecimal). Motors which drive to determine the “elevation” (or tilt) of the antenna have an address ‘32’ (hexadecimal). The DiSEqC™ protocol is that a ‘0’ nibble is a “wildcard” (or “don’t care”) so that address ‘30’ (hexadecimal) sends commands to all Positioner addresses (i.e. including ‘31’ and ‘32’). Clearly, commands which are specifically intended to produce movement in one axis of a two-axis system must use the relevant address byte. Commands which are relevant to both axes can be sent to the general (wildcard) address (‘30’). Simple Tuner-receiver/IRDs which do not support any vertical (tilt) movement should generally use the specific address ‘31’ to avoid malfunctioning with a Positioner Motor Unit which incorporates remote control of tilt. However, very few additional commands are required to support two-axis control, so it is recommended to use all the commands with their correct associated addresses, as shown in *section 5*.

All the Positioner Command bytes defined so far have a ‘high’ hexadecimal nibble ‘6’ and are shown in *Table 1*. It is Recommended that all Slaves (including Positioners) should support “Level 2.x ” reply messages (e.g. “OK”, “Unrecognised Command”, etc.) when requested, but the basic commands do not demand the return of actual data values for the system to operate successfully.

Column 1 in *Table 1* indicates which commands are mandatory for inclusion in Positioner Motor Units (‘M’) and Tuner-receiver/IRDs (‘T’) to qualify for the ‘1.2’ (or higher) DiSEqC™ Logo. The equivalent lower-case letter ‘r’ indicates commands which are recommended for inclusion in these units. The penultimate column (“Total transmitted bytes”) indicates whether the message uses just the basic 3 bytes, or whether one or more additional data bytes are included in the message. Only one (non-mandatory) command requests the return of data (the status byte) from the slave, as indicated in the final column of *Table 1*.

The following sub-sections describe each command in detail.

Mandatory for (M)otor / (T)uner	Hex. Byte	Command Name	Command Function	Total trans. Bytes	Reply Data byte(s)
M / T	60	Halt	Stop Positioner movement	3	
-	61		Reserved		
-	62		Reserved		
M / T	63	Limits Off	Disable Limits	3	
-	64	PosStat	Read Positioner Status Register	3	Pos Status
-	65		Reserved		
M / T	66	Limit E	Set East Limit (& Enable recommended)	3	
M / T	67	Limit W	Set West Limit (& Enable recommended)	3	
M / T	68	Drive East	Drive Motor East (with optional timeout/steps)	4	
M / T	69	Drive West	Drive Motor West (with optional timeout/steps)	4	
M / T	6A	Store nn	Store Satellite Position & Enable Limits	4	
M / T	6B	Goto nn	Drive Motor to Satellite Position nn	4	
-	6C		Reserved		
-	6D		Reserved		
-	6E	Goto x.x°	Drive Motor to Angular Position (°)	5	
- / r	6F	Set Posns.	(Re-) Calculate Satellite Positions	(4) / 6	

Table 1: DiSEqC™ Positioner Commands

3.1. “Halt Motor”

Command byte ‘60’ causes the motor to stop driving immediately. Execution of the command by Positioner Motor Units is obvious, but the command may be generated by the Tuner-receiver/IRD software on occasions not directly related to the user pressing a remote control key. For example the “Drive Motor” commands (see *section 3.5.* and *section 3.6.*) include an optional “timeout” parameter, but this always should be supported (after an appropriate delay) by a “Halt” command from the Tuner-receiver/IRD.

A common User-Interface function is to drive the positioner motor whilst a (remote control) key is held depressed. This facility can be obtained by generating a “Drive Motor” command when the key is initially pressed and then sending a “Halt” command when the key is released. Note that this function could be implemented in the Remote Control protocol (Infra-Red or Radio), by sending separate commands for “start” and “stop” when respectively pressing and releasing a key. Alternatively, the onset and termination of a stream of repeated Remote Control “Drive Motor” commands might be converted to the DiSEqC™ “Drive” and “Halt” commands in the Tuner-receiver/IRD embedded software.

3.2. “Disable (soft) Limits”

Command byte ‘63’ disables both the East and West “Soft(ware) Limits”. This allows the positioner motor to drive over the full (mechanically-defined) arc of its movement, when commanded by the normal “Drive Motor” commands. It is not mandatory to support this command in Positioner Motor Units, but the command must be available from Tuner-receiver/IRDs to ensure that motor units which do support soft limits can be aligned over their full mechanical range.

It is recommended that the “Disable Limits” command is “protected” in Tuner-receiver/IRDs to prevent accidental transmission, particularly by the normal user (as opposed to the installer). In On-Screen-Display systems, this may be by including it in an “Installer Menu” and/or with an “Are You Sure?” prompt or a “PIN” code number.

3.3. “Set East (soft) Limit”

Command byte ‘66’ stores the present position of the motor shaft as the East (Soft) Limit, or “endstop”. To avoid accidental changing of the Limit, it is recommended that this command is only executed by the Positioner Motor Unit when the Limit is at present disabled, and this should then automatically enable the Limit. Whether the other (West) Limit is also enabled by this command is optional in the implementation of the Positioner Motor Unit embedded software. If both Limits are enabled by the command then it will be necessary for another “Disable Limits” command to be sent before the other (West) Limit can be set. Alternatively, if only the associated Limit is enabled then it would be necessary to transmit an “Enable Limits” command (see command byte ‘6A’) if only one Limit was to be changed.

If limits are “Enabled”, then whenever the motor is driving, it must stop on reaching the positions which were stored by the Set East and Set West Limit commands.

3.4. “Set West (soft) Limit”

Command byte ‘67’ produces exactly the same behaviour as command byte ‘66’ except that the West Soft Limit is affected instead of the East Soft Limit.

3.5. “Drive Motor East”

Command byte ‘68’ is used with a following data byte to determine the length of time that the motor drives to the East. The default value for the timing parameter byte is ‘00’ for which the Positioner Motor drives continuously in the required direction. It is mandatory for Master controllers (Tuner-receiver/IRDs) to transmit at least this data value (or any other positive value).

Positive (non-zero) values of the parameter byte (i.e. ‘01’ to ‘7F’ hexadecimal) indicate a “timeout” value in seconds, but Positioner Motors not supporting a timing function should drive continuously if receiving these parameter values. A typical use would be for the Master controller to transmit a “Go East for 4 seconds” command and repeat it, say, every 3 seconds whilst movement is required. This is “fail safe” if a “Halt” message becomes corrupted (invalidated) on the bus. Note however, that Master controllers must still transmit a “Halt” command when movement is to stop, (they must not rely on the timeout alone), because the timeout may not be supported by all Positioner Motor Units.

Negative parameter values (i.e. ‘80h’ to ‘FFh’) indicate the number of “steps” the motor should make, and Positioner Motors should only drive if they support this movement process. Thus, Positioner Motor Units which do not support this function should not drive if the Most Significant Bit of the parameter data byte is a ‘1’ (the “negative flag”).

Each “step” will typically correspond to the smallest angular movement of the antenna shaft which the associated dish can usefully resolve, i.e. perhaps 5% to 10% of the beam width of the antenna. This is not necessarily the same as the pulse resolution from the shaft rotation sensor (reed relay, Hall-effect switch, etc.), but these pulses can easily be divided down by the Motor Unit embedded software to the appropriate step value.

The number of steps to make is given by the additional count needed to make the parameter byte reach zero (or overflow to zero if the byte is considered as unsigned). Thus the byte 'FF' (hexadecimal) requests only one step, 'FE' two steps, and for example 'F9' requests 7 steps. The advantage of the "step" format of this command is that it is much easier and repeatable to "nudge" the alignment when installing the antenna (or a new satellite) than simply driving the motor for a short time.

In all the above circumstances, if the East (Soft) Limit has been Set and Enabled, then the motor should stop driving if it reaches the stored limit point before completing the defined task.

3.6. "Drive Motor West"

Command byte '69' with a data byte 'xx' is used exactly as command '68', but to drive the Positioner Motor to the West and with reference to the West (Soft) Limit (if Set and Enabled).

3.7. "Store Satellite Position"

Satellite Positions are defined by positive binary number parameters from 1 ('01' hexadecimal) upwards. Thus the command and parameter bytes '6A 01' store the present motor position as "Satellite Position 1". In addition, the Positioner Motor Unit embedded software should Enable both (soft) Limits if they have been set. The maximum number of positions to be stored is not defined, but it should not be less than 8. 30 or 40 satellites may be more typical, particularly if the motor unit has pre-stored satellite position data in its ROM (for automatic installation).

The parameter byte '00' (i.e. Satellite Position 0) is reserved as a "reference position" which is not valid for storing satellite position data. However, some applications may require a specific command to Enable (Soft) Limits, so all Masters (i.e. Tuner-receiver/IRDs) should be capable of transmitting command '6A' with a data byte '00', either specifically as an "Enable Limits" user-command, or by allowing "Position 0" to be selected for the "Store Satellite Position" installation command.

3.8. “Goto Satellite Position”

Command byte and parameter ‘6B xx’ causes the Positioner Motor Unit to drive back at any later time to the antenna shaft position stored by the associated command ‘6A xx’. The parameter byte ‘00’ (i.e. Satellite Position 0) is reserved as a “Reference Position” which will normally correspond to due South, but may be the easterly “Hard Limit” (i.e. defined by physical switching within the motor unit) if so implemented in the Positioner Motor Unit. The reference position is primarily intended for use during system installation, but may be required for antenna alignment at subsequent times.

All Masters (i.e. Tuner-receiver/IRDs) should be capable of transmitting command ‘6B’ with a data byte ‘00’, either specifically as a “Goto Reference Position” user-command, or by allowing “Position 0” to be selected with the “Goto Satellite Position” user-command.

3.9. “(Re-) Calculate Satellite Positions”

Command byte ‘6F’ with various data parameters can give the user/installer various types of “short-cut” to setting up or aligning the system. The basic command has the (first) parameter byte set to zero, i.e. command ‘6F 00’ which can initiate any appropriate function, defined by the manufacturer of the Positioner Motor Unit. In general, the user/installer will manually drive the antenna to receive a strong signal from a suitable satellite and then the command ‘6F 00 {00 00}’ will cause the embedded software in the Positioner Motor Unit to calculate the (approximate) positions of all the other satellites in its memory (ROM) relative to the known position. The user/installer may then need to optimise each position in turn with reference to the received signal strength.

It is preferred that Tuner-receiver/IRDs with Menu systems should support 3 (signed) binary parameter bytes for the “Calculate Satellite Positions” command. Again, their exact function will be defined in the documentation for the Positioner Motor Unit, but they typically may be used for, “Satellite Number”, “X-Value” (e.g. Site Longitude in degrees) and “Y-Value” (e.g. Site Latitude in degrees). Thus if the user/installer can receive any (identifiable) satellite signal, then the command can contain sufficient data to calculate the positions of all other satellites stored in the Positioner Motor Unit’s ROM.

Strictly, the “X-Value” data byte cannot define a unique Longitude in the World because its range is only 256 degrees. However, a suitable range can be inferred from the range of satellites for which data is stored (i.e. those expected to be above the horizon in the relevant continental region). In Europe and Africa the “X-Value” can be signed binary (positive to the East) so that it defines longitudes from 128° West to 127° East. In the Middle East and Asia the “X-Value” can be unsigned binary so that it defines longitudes from 0° to 255° East (i.e. to 105° West). In the Americas the “X-Value” generally would be unsigned binary to the West, so that it defines longitudes from 0° to 255° West (i.e. to 105° East).

3.10. “Goto Angular Position (degrees)”

Command byte ‘6E’ is not yet fully defined, but will normally use two data parameter bytes. At present the command is defined for control of a terrestrial antenna rotator, e.g. for the azimuth of a horizontally-directed Yagi or microwave antenna.

The high nibble of the first data byte indicates how the remainder of the angular data is to be interpreted. The values ‘F’, ‘0’ and ‘1’ are allocated to support full rotation in a horizontal plane (e.g. for terrestrial sources). The nibble ‘0’ references the angle defined in the remainder of the data to North (0°), ‘1’ adds 256° clockwise to the angle and ‘F’ subtracts 256°. The low nibble of the first data byte indicates the clockwise rotation angle in increments of 16°. Thus a first data byte of ‘00’ defines due North, ‘01’ defines an azimuth of 16° East, ‘0A’ (hexadecimal) an azimuth of 160°, and ‘10’ (hexadecimal) a rotation of 256° clockwise from north.

The high nibble of the second data byte defines the angle in degrees, and the low nibble indicates the fractional part of a degree, if required. Note that the low nibble is a true binary fraction, and is not Binary Coded Decimal. Thus the most significant bit (of the low nibble) defines half a degree, the next bit a quarter degree and the least significant bit a sixteenth. If the Tuner-receiver User-interface is to show a decimal value, then the ten closest fractional values can be employed, as given in the following table.

User-interface value	Fractional nibble data value	Actual angle	User-interface value	Fractional nibble data value	Actual angle
0.0°	0000 (0h)	0.0000°	0.5°	1000 (8h)	0.5000°
0.1°	0010 (2h)	0.1250°	0.6°	1010 (Ah)	0.6250°
0.2°	0011 (3h)	0.1875°	0.7°	1011 (Bh)	0.6875°
0.3°	0101 (5h)	0.3125°	0.8°	1101 (Dh)	0.8125°
0.4°	0110 (6h)	0.3750°	0.9°	1110 (Eh)	0.8750°

Table 2: Fractional Coding for Decimal Angles

Examples of complete commands for the main points of the compass are:

Compass Direction	Backwards Rotation from North	First Rotation from North	Second Rotation from North
North (0°)		E0 31 6E 00 00 (0°)	E0 31 6E 16 80 (104+256=360°)
East (90°)		E0 31 6E 05 A0 (90°)	E0 31 6E 1C 20 (194+256=450°)
South (180°)	E0 31 6E F4 C0 (76-256= -180°)	E0 31 6E 0B 40 (180°)	
West (270°)	E0 31 6E FA 60 (166-256= -90°)	E0 31 6E 10 E0 (256+14=270°)	

Table 3: Sample Azimuth Commands

3.11. Read Positioner Status Byte (Level 2.2)

The Positioner Status Byte is returned in response to command byte ‘64’ and contains individual flag bits to indicate the operational conditions of the software and peripheral hardware related to the Positioner Motor Unit. It is only relevant in DiSEqC™ Level 2.2 “two-way” systems.

Bit Number	Positioner Function
.7	Movement command has been completed
.6	Software Limits are Enabled
.5	Movement direction is (or last was) West
.4	Motor is running
.3	Software Limit (end-stop) has been reached
.2	Power is not available
.1	Hardware switch (Limit or Reference) is activated
.0	Position reference data has been lost or corrupted

Table 4: Positioner Status Byte flag allocations

4. User Interface

The interface that the User has to control the Positioner Motor Unit via the Tuner-receiver/IRD will depend on the architecture and design of the Tuner-receiver, and the (Infra Red or Radio) Remote Control protocol. However, this section suggests how the required User-functions can be mapped on to a simple numeric keypad, or an On-Screen Display Menu with about 10 options per screen. The command numbers allocated here are such that the directional commands correlate reasonably with the layout of a normal “telephone style” keypad, i.e. a square with ‘1’ in the top left corner, ‘7’ in the bottom left corner and ‘0’ below the ‘8’.

Rather more than 10 commands may need to be offered to the “User”, but the interface can be simplified and protected by subdividing the functions into a special “Installer” group and a general-purpose “Viewer” group. However, even the “Viewer” commands only need to be used occasionally, perhaps if the system has become mis-aligned, or to introduce new satellites into the

memory. Normally, the Tuner-receiver/IRD embedded software generates the required commands automatically when the viewer changes the programme channel.

The “Installer” group of commands includes those concerned with the setting of (Software) Limits and the “Automatic Calculation” of satellite positions (relative to a pre-defined reference). Incorrect use of these commands can potentially upset the functionality of the complete Positioner, perhaps taking a considerable time to restore, so it is suggested that either the whole menu, and/or each of these commands individually, should be protected. Possible techniques are a “PIN” (Personal Identification Number), an “Are You Sure?” message, or a special operation of the Remote Control Keypad such as a long keypress, or the combined depression of more than one key.

In DiSEqC™ Level 1.2 “one way” systems (i.e. without a reply) with a “Soft Limits” facility it can sometimes be unclear whether the limits are actually enabled or not. It is suggested that commands concerned with these limits should be transmitted twice by the Tuner-receiver/IRD, to avoid unnecessary confusion if a message is lost due to noise or interference. Similarly, the “Store Satellite” command is worthy of repetition in all “one way” systems.

The following sections give a pair of sample menus which include all the recommended commands. There is some duplication of commands, but this avoids unnecessary switching between menus, and permits different versions of the same command to be included (e.g. “Start” and “Nudge” versions of the “Drive Motor” commands).

4.1. Typical “Installer” Command Menu

Key	Command Description	Typical DiSEqC™ Command
1	Start Motor Driving West	E1 31 69 40
2	Halt Motor	E0 30 60
3	Start Motor Driving East	E1 31 68 40
4	Set West (soft) Limit	E1 30 67
5	Disable (soft) Limits	E0 30 63
6	Set East (soft) Limit	E1 30 66
7	Go to Reference Position	E0 30 6B 00
8	Enable (soft) Limits	E0 30 6A 00
9	(Re-) Calculate Positions	E0 30 6F 00 {00 00}
0	Select “User” Menu	

Table 5: Typical “Installer” Menu

The “Start Motor” commands normally cause the Positioner Motor to drive in the selected direction until cancelled by the “Halt” command. However, the particular data byte shown for the DiSEqC™ Command in *Table 5* is actually non-zero to give a “timeout” (if supported by the Positioner Motor Unit), but this is nominally set to approximately 64 seconds (‘40’ hexadecimal) so would not normally be reached. Note that only the commands referring directly to East and West motion have the specific address ‘31’. All the other commands have the “wildcard” address ‘30’ since they might equally apply to a motor controlling the vertical (tilt) axis.

Usually, the Limits will be Disabled with key ‘5’ and then automatically re-Enabled as each Limit is set. However, the User may accidentally Disable the Limits, in which case key ‘8’ can be used to restore them. Note that this command is also available on the other (User) menu. Similarly, key ‘7’ (Goto Reference Position) can be supported by the general “Goto Position Number” in the User Menu, with position number ‘00’.

The “(Re-) Calculate Positions” command is shown with the basic default parameters, but a more sophisticated Menu system might open a “window” to allow the three parameters to be adjusted to suitable decimal values. It is suggested that the first data byte may take any value defined by the decimal range 0 to 255, the second byte takes values from -180 to +359 (i.e. each byte can be derived from more than one decimal value) and the third byte takes a value from -128 to +127 decimal. The designer of the embedded software in the Positioner Motor Unit will have to decide to what extent this command should be allowed to “overwrite” data which has previously been stored.

4.2. Typical “User” Command Menu

Key	Command Description	Typical DiSEqC™ Command
1	Nudge Motor West	E1 31 69 xx
2	Toggle Step/Timed Mode	[xx = 256 - xx]
3	Nudge Motor East	E1 31 68 xx
4	Nudge Tilt Motor Up	E2 32 68 xx
5	Store Present Satellite Number	E0 30 6A nn
6	Increase Nudge Size	[xx = xx + (sign xx)]
7	Nudge Tilt Motor Down	E2 32 69 xx
8	Go to Satellite Number	E0 30 6B nn
9	Decrease Nudge Size	[xx = xx - (sign xx)]
0	Select “Installer” Menu	
+	Increment Satellite Number	[nn = nn + 1]
-	Decrement Satellite Number	[nn = nn - 1]

Table 6: Typical “User” Menu

Table 6 shows a typical set of commands which may be offered to the general User/Viewer. Note that only six of these actually generate DiSEqC™ messages for the bus, the remainder adjust internal constants which are used by the commands.

The “Nudge Motor” commands are basically the same as the “Start Motor” commands but have the data parameter set to give a “Step” or “Timeout” function (if supported by the Positioner Motor Unit). These commands would be used by the Viewer to optimise picture quality (signal strength) manually, or to gradually search for a new satellite. If the data byte is positive (Most Significant Bit = ‘0’) then the motor drives for a few seconds, and if negative (Most Significant Bit = ‘1’) then the motor moves for a few “steps”.

With the user-functions shown here, the timeout or step range can be adjusted with keys ‘6’ and ‘9’, perhaps over a range of 1 to 9. Key ‘2’ toggles between stepping and timeouts (perhaps described as “Fine” and “Coarse” Nudges) and can be implemented by changing the sign of the decimal value (or the “two’s complement”) of the data byte. A value of ‘0’ should be used with caution because it gives an “infinite” timeout.

Alternatively, the Nudge Range keys could be arranged to give a continuous sequence through the “step” and “timeout” ranges. A typical sequence, increasing in step size could be:

‘FF’ ‘FE’ ‘FD’ ‘FC’ ‘FB’ ‘01’ ‘02’ ‘03’ ‘04’ ‘05’ ‘00’.

The other commands permit the User to store the present antenna position with reference to the “Satellite Number”, and subsequently cause the Positioner Motor Unit to drive back to this stored position. In practice, the “Go to Satellite Number” DiSEqC™ Command normally will be generated by the Tuner-receiver/IRD embedded software when the Viewer selects a new channel to watch (or perhaps only when the software determines that the new channel is on a different satellite to that being viewed). The “Store Satellite Number” command will usually be used after the antenna has been nudged to a new position, either to optimise the signal quality, or to install a new satellite. To prevent accidental corruption of an existing setting, the Tuner-receiver/IRD embedded software might issue an “Are you sure?” prompt, if the new location is significantly different (say more than a few degrees) from the previously stored value.

The Satellite Number is adjusted in this example menu by the ‘+’ and ‘-’ keys which respectively increment and decrement the Satellite Number displayed in the On-Screen Menu, or on the front panel of the Tuner-receiver/IRD. The highest value normally will be defined by the memory capacity (number of “satellite position” bits) supported by Tuner-receiver. The lowest limit is 1, or perhaps 0 if the Installer Menu does not include the explicit commands with a 00 data byte. If ‘+’ and ‘-’ keys are not available, then the ‘2’ key could initiate entry of the “Satellite Number” as a numeric value, or the “Increase/Decrease Satellite Number” function could replace the “Step Size” or “Tilt” key functions.

5. Positioner Units Embedded Software

The Positioner Motor Unit probably will use a single microcontroller to perform all the necessary functions within the assembly. Typically, it will need to monitor the DiSEqC™ bus for messages, monitor the motor position (e.g. by counting pulses from the drive-train) and check for hardware and/or software “endstops”. At appropriate times it may also need to write data to internal or external Non-Volatile Memory, and to transmit reply messages back to the Master (Tuner-receiver/IRD).

Which of the above functions are initiated by an interrupt and which by “polling” (i.e. repeatedly testing the state of a pin or flag, to determine when it changes) will depend on the architecture of the microcontroller, the embedded software and the hardware. This section just gives a few suggestions for the implementation of the DiSEqC™-specific functions.

It is possible to detect the DiSEqC™ data bits modulated onto the 22 kHz carrier using fairly simple hardware. However, the microcontroller normally will not be very heavily loaded (in timing or memory size terms), so it should be generally advantageous to detect the data directly from the carrier tone, using real-time software embedded in the microcontroller, as outlined below.

The amplitude of the DiSEqC™ carrier tone on the bus is generally too small to detect directly on a “TTL” or “CMOS” compatible pin on the microcontroller, so usually a “comparator” input, or a simple external (one-transistor) amplifier, will be required. In either case, it is important not to make the input too sensitive to small-amplitude signals which may be “noise” or interference. It is recommended that the smallest amplitude normally detected is about 300 millivolts peak-peak, which can be achieved either with hysteresis (positive feedback applied around the comparator/amplifier) or with a d.c. bias offset (equivalent to about 150 millivolts) applied to the input of the amplifier/comparator. Hysteresis (if symmetrical) can maintain a reasonably constant 50% duty cycle for the detected carrier tone, whilst the d.c. offset approach may generate a rather asymmetric (pulse) waveform when the carrier amplitude approaches the lower limit.

DiSEqC™ messages can be demodulated directly from the carrier using either an interrupt or a polling structure. Direct polling needs to be performed at above a 50 kHz rate to avoid missing any half-cycles of the upper-limit carrier frequency (26.4 kHz), even if the detected carrier is near the optimum 50% duty cycle. Servicing direct interrupts at the carrier rate can introduce a significant time overhead (entering and exiting the service routine), and also needs a “timer” to be available to measure the pauses between bursts of carrier, and to exit at the end of each message.

With some microcontrollers it is possible to use a compromise polling/interrupt arrangement. The carrier tone on the bus is arranged to drive an edge-triggered interrupt input. However, the interrupt is not directly serviced (i.e. it is disabled), but a polling routine tests the associated interrupt flag at about a 17.5 kHz rate (i.e. just below the minimum carrier frequency). Thus, if the carrier is present then the routine always finds the flag set (regardless of the carrier frequency), and if there is no carrier then the flag is clear. Of course, the interrupt flag must be repetitively cleared, either by software in the polling routine, or automatically when the flag is read. An advantage of this continuous polling technique is that the other status information such as driveshaft rotation and endstop operation can be economically included in the polling loop.

Sometimes it is necessary to perform an action which is incompatible with the continuous polling loop, for example decoding a DiSEqC™ message, sending a reply, or sending data to the Non-Volatile memory (perhaps external, via an IIC Bus). However, these actions are normally required immediately after receiving a DiSEqC™ message, when the bus can be assumed to be “quiet” for at least five milliseconds, and usually much longer.

When actually transmitting a reply message, it is not necessary to read data from the bus, so a separate, accurately-timed, program loop can be arranged to transmit the data and to poll any urgent status data (i.e. that which may change again during the 13 or 27 ms period of a typical reply message).

The basic requirements and characteristics of receiving and transmitting messages over the DiSEqC™ bus are described in the *Bus Functional Specification Version 4.2*. Also, an indication of a typical set of commands to be supported, their validation, and particularly the protocol for transmitting Reply messages, may be found in the specification for the *Slave microcontroller version 1.0*, published by EUTELSAT. However, the likelihood of there being multiple Positioners, with the same address, on the bus is low, so it is probably not necessary to include the random delay arbitration protocol incorporated in the general-purpose Slave. A simple fixed delay of typically 10 ms between the end of the command and the start of the reply may be satisfactory.

6. Contact Details

For further information, please contact:

Guy Wilkinson

Tel: + 33 1 53 98 37 07

E-mail: gwilkins@eutelsat.fr

Laurence Bailly

Tel: + 33 1 53 98 46 42

E-mail: lbailly@eutelsat.fr

Network Development

70 rue Balard

F - 75502 PARIS CEDEX 15

France

Fax: + 33 1 53 98 40 23

Note: All documents are available on the EUTELSAT web site :

<http://www.eutelsat.com/docs/diseqc>

If you are not already receiving regular DiSEqC™ mailings, please remember to complete the registration form named “Technical Information Service” on the web site.