

## Controller CBOX 2003 A Control Box that can be fully operated via RS232. (Firmware versions 3.22)

### Basic philosophy

The CBOX (or controller) can be connected to a remote object (usually a PC) which can behave as:

- Passive monitor of position and status (rotation in progress or not)
- Active controller: send to the CBOX some commands and receive back from the CBOX all related data.

The CBOX is being developed to control up two rotators, one for azimuth and the other for elevation (or whatever you want). As we see in the following, a command can be selectively sent to a specific rotator, and data coming from the controller carries the rotator identifier.

In the following, the words “send” and “receive” will refer to data transfer over RS232.

### RS232 port.

- Speed: 9600 Baud
- Character length: 8-bit
- Parity: none
- Stop bits: 1

### → The commands are CASE SENSITIVE ←

### CBOX status at power-on.

At power-on the CBOX is enabled to send the position and status information whenever one of the two changes. This is the Continuous Position Monitor operating mode (or **CPM**). You can disable the CPM through the **S** (Stop CPM) command, which disables CPM until next power-up or **M** (enable CPM) command.

*At power-on the following happens:*

- “PHH” is displayed on the display for 1 second
- The current position is displayed and the position+status string is sent TWICE, making easier for the remote control program to synchronize, if already running. Note that the status flag should be **R** after the power-on.
- The **CPM** mode is entered: a string is sent whenever the rotor position and/or the operational status change
- The on-board dip-switch status is read: it will be used by the program to handle some parameters
- The CW switch status is checked, to eventually start the **absolute** mode entering procedure
- The CBOX is now ready to accept commands via RS232 or using the CW/CCW switches: there are some limitations in the **absolute** mode

**Warning!** For big antennas requiring a soft start and stop to avoid mechanical failures: rotations through the CW/CCW keys does NOT have any “soft” feature. BOTH START and STOP occur abruptly. Also, if you press any of the two keys during a rotation, an IMMEDIATE STOP (FAST STOP) occurs.

### Commands format:

**<header>** (Chr(2), binary 00000010); it resets the controller command buffer pointer  
**<identifier>** (rotator identifier: ‘A’ for azimuth, ‘E’ for elevation)  
**<cmd>** (see list below)  
**<return>** (Chr(13), binary 00001101)

### <cmd> format:

- **L** : List Parameters
- **W** : Write Parameters
- **V** : Firmware Version. The CBOX reply sending the current firmware version on five digits
- **?** : Fast Position Read. The CBOX will reply sending a string reporting the current position and the current operative status
- **G<angle>**: GOTO command, where the angle is a three digits (hundreds, tens, units) number, leading 0’s are optional, values from 0 to 359. This command tells to the CBOX to turn the antenna from the current position to the <angle> target position. Note that some abnormal angle values are used as special commands, see **G999** and **G977** below.
- **G999** : Fast-stop, immediate motor stop

- **G977 : Soft-stop**, the motor will stop “softly” following the PWM mode behavior ONLY if the PWM mode is ENABLED. If PWM is NOT ENABLED, there will be a **Fast-stop**
- **M** : (Continuous Position Monitor, **CPM**), this command enables the continuous position+status info via RS232, whenever the position and/or the status change. For the data format see the ? command. Command not accepted in **absolute** mode.
- **S** : (**Stop CPM**), disables the **CPM** mode. The ? command can then used to obtain the position+status information.
- **C** : calculate the **Calibration** factor assuming the current position is 360° absolute, command accepted in **absolute** mode only.
- **c** : force the default **calibration** factor (8000), needed to allow a correct position reading, command accepted in **absolute** mode only.
- **K<angle>** : calculate the calibration factor assuming the current position is <angle>° absolute, command accepted in **absolute** mode only.
- **F<factor>** : set the calibration factor value to <factor>, command accepted in **absolute** mode only, minimum allowed value is **1000**, if <1000 error is returned.

The **GOTO** command will be accepted and replies sent even during rotation (when the CBOX is Busy). The new angle target is handled automatically by the CBOX according to the various parameters settings.

#### How the CBOX buffers the commands.

The controller handles the RS232 reception through an interrupt routine. When a character arrives, it is enqueued in a 64-positions circular buffer by the interrupt routine.

The command parsing routine retrieves from the circular buffer one character at time and store it in the command buffer, updating the associated pointer. If a <header> character is retrieved the pointer is reset, so the <header> will be always at position 0 of the command buffer. When a <return> character is retrieved, the command is complete: the command buffer pointer is reset and the command is hence parsed.

For commands following the rules, in the command buffer we have:

<u>position content</u>	
0	<header>
1	<identifier>
2	<cmd>
...	
N	<return> (so the <b>command length</b> is defined to be N)

**Warning:** If the commands arrive at too high rate, the circular buffer could be overfilled and some characters could be missing. It is recommended to wait for the reply before to send another command. The commands are case sensitive.

#### How the CBOX reply (via RS232) to the commands:

<header>	(Chr(2), binary 00000010)
<identifier>	('A' for azimuth, 'E' for elevation; in case of format error: the first character after the <header> of the command string the CBOX received)
<separator>	(';', the comma character)
<cmd>	(see list below; in case of format error: the second character after the <header> of the command string the CBOX received)
<separator>	(';', the comma character)
<data>	
<return>	(Chr(13), binary 00001101)

Where <data> is:

**<data1><separator><data2><separator>...<dataN>**

<identifier> is **A** for Azimuth and **E** for Elevation. In case of command format error, it is the first character after the <header> of the command string the CBOX received.

<cmd> is the same as defined above in **Commands format** for the commands that preview a reply: **L, W, ?, G**. With the **CPM** enabled, the controller send the position+status info using the same format as for **?** command. For format errors it is the third character after the <header> of the command string the CBOX received.

**<data>**:

- **Format errors** (illegal characters or commands to a non-implemented rotator, for example to the Elevation one in the current firmware version, or commands with parameters out of the allowed values – cmd **W** excluded): <data1>=**E**, <data2>=command length coded as decimal number digits (like <data1> in **?** command). Example: if you send <header>qwerty123<return> as command string, you will receive back <header>q,w,E,00010<return>, where 00010 (equivalent to 10) is the command length. The command length is calculated <header> excluded and <return> included. The command buffer has a 64-bytes length.
- Other errors: for CBOX implementations without the Elevation rotator, all commands having <identifier>=**E** will produce a format error reply, as well as all angle values greater than 359 or less than 0, but different from the special values 999 (FAST STOP) and 977 (SOFT STOP).
- Command **W**: **B** (Busy) if a rotation procedure is currently executing, so a parameters changes isn't allowed; **R** (Ready) otherwise, meaning you can send all the required parameters.
- Command **L**: a string containing all the parameters values (see **Parameters List** below).
- Command **?**: <data1>=current angular antenna position (**three digit angle** as hundreds, tens, units); <data2>=**one status character** (**R** = Ready, when the CBOX isn't currently executing a rotation, or **B** = Busy, when a rotation procedure has not ended yet). Example (only <data> are reported): **157,B**, that means position=157° and status=Busy (rotation in progress).
- Command **G**: <data>=**R** if the format is correct, =**E (format error)** if the angle is outside the **UPPER/LOWER LIMITS** (but different from 999 and 977) or consisting of more than **five digits**.
- Command **C, c, K, F**: **B** (Busy, command not executed) if the controller isn't in **absolute** mode or a rotation procedure isn't ended, so a parameters changes isn't allowed; **R** (Ready) otherwise, meaning the command has been executed. For **command F**: if the factor value is less than **1000** a **format error** is issued
- Command **M**: **B** (Busy, command not executed) if the controller is in **absolute** mode, **R** otherwise
- Command **S**: **R**
- Command **V**: <identifier>=**A**, <data>=**Firmware Version**, decimal number on five digits XXXXX, to be decoded as XXX.XX

**An example.**

After the power-on (CPM enabled) you can send a **GOTO** command to rotate the antenna:

- Send a **GOTO** command if you want to turn the antenna to any desired position
- The CBOX replies with a <data>=**R** to say that the command has been accepted. Alternatively it can send a <data>=**E** (Error) if the command you sent is not correct: for example, if you send a H245 as target angle, where the H is clearly not an expected digit. Anyway, after a GOTO command, you will receive back one character (E or R) before the rotation eventually (if you receive R) will start.
- As soon as the rotation starts, the CBOX will begin to send the position+status packets with status=B until the procedure is completed. At the end, on the last sent packet, the status flag will change from B to R.

**IMPORTANT**

The CBOX accepts commands also during a rotation. If the CPM is enabled, it is sending the position+status string. It is NOT guaranteed that after you send a command in such a situation you will receive as first character the CBOX reply for the command sent (it depends on the PC RS232 buffer filling status): you can receive other characters belonging to the position+status strings before the expected reply for the command appears from the RS232 buffer and is seen by the PC program.

So, if you want to use the CPM mode and send any commands please take this into account. Alternatively, you can disable the CPM mode (command **S**) and then the CBOX will work on the simpler command-reply handshake behavior.

The CBOX does NOT process commands when a rotation is finished and it is in the REV\_DELAY waiting time. In such a case, the command is received and buffered but not parsed. It will be parsed after the waiting time. So, if your controlling program on the PC implements a timeout on the CBOX reply, the timeout time should be greater than the REV\_DELAY time.

**Display decimal points blinking.**

The decimal points on the LED display blink when:

- you turn the target position knob on the front panel

- rotation begins and is in progress
- after the end of rotation, while the software waits the programmed time before allowing a new rotation (if you are using the CW/CCW keys, such a wait time is zero for successive rotations in the same direction)

### “Err” is displayed

when you turn the target position knob on the front panel to an angle outside the rotation limits.

### Using the keyboard (version with keyboard).

Here the keyboard is used in place of the rotating knob and the two CW/CCW switches.

To enter the target angle simply push the numeric keys followed by '#' as terminator. The display will show the entered number when you will press it. The '\*' key acts as clear (reset to 0 the entered number) or as 'abort' (returning to the position display with no action) when the number displayed is already 0.

If you try to enter a number greater than 359 the display immediately resets to 0.

An example:

- initial condition: the display shows the position, e.g. 185
- push and release (all keys are accepted on the release action) 3: on the display, at the key release, appears a 3 on the leftmost position, so: 3<blank><blank>
- push and release 4: display 34<blank>
- push and release 6: display 346
- push and release # and immediately the rotation starts
- the '\*' key can be used as clear of the entered number when you are entering it. If you push (and release) '\*' when the entered number is 0 the keyboard input routine exits with no actions (input abort).
- When the CBOX is not doing a rotation, '\*' works as CCW key and '#' as CW key.
- during a rotation the '0' key is enabled as FAST-STOP key

### Parameters:

All the parameters needed by the control program are stored in the non-volatile processor's EEPROM. At power up they are copied to RAM locations to allow a faster access. Whenever a **W** command is executed, the EEPROM content is modified and the copy to RAM re-done.

It is possible to control some of such parameters using the on-board dip-switch, see in the following for the details. In case of the dip-switch control is active (switch #1 is ON), the parameters MODE, ENA\_PWM, LOWLIM, UPLIM cannot be modified by the **W** command. Such parameters' values are copied from dip-switch status to the EEPROM at controller power-up.

- **POSITION**: read only, antenna position in degrees, values depending on the **MODE**
- **MODE**: read/write: =0 **north-stop**, =1 **south-stop**, =2 **absolute**
- **LOWLIM**: read/write, lower limit for position, in absolute degrees (less than UPLIM). If POS\_OFFSET is greater than 0, the following action is performed: if  $(LOWLIM - POS\_OFFSET) < 0$  then  $LOWLIM = LOWLIM + POS\_OFFSET - LOWLIM$ . In such a way a rotation below the 0° absolute hardware limit is prevented.
- **UPLIM**: read/write, upper limit for position, in absolute degrees (greater than LOWLIM). If POS\_OFFSET is less than 0, the following action is performed: if  $(UPLIM - POS\_OFFSET) > 500$  then  $UPLIM = 500 + POS\_OFFSET$ . In such a way a rotation above the 500° absolute hardware limit is prevented.
- **REV\_DELAY**: read/write, waiting time, after a stop, before to restart the motor in the opposite direction, in units of 1.64ms. When you turn the position rotating knob, there is the same waiting time from the last knob position changing to the rotation start time.
- **ENA\_PWM**: read/write, to enable (=1) or disable (=0) the PWM
- **PWM\_ON\_VALUE**: read/write. During each PWM signal period, at maximum motor speed, this is the PWM=1 time. By definition,  $PWM\_ON\_VALUE + PWM\_OFF\_VALUE = PWM\_PERIOD$
- **PWM\_OFF\_VALUE**: read/write. During each PWM signal period, at maximum motor speed, this is the PWM=0 time. By definition,  $PWM\_ON\_VALUE + PWM\_OFF\_VALUE = PWM\_PERIOD$
- **RAMP**: read/write, (valid if ENA\_PWM=1). This is the angular distance (measured in degrees) after the rotation starting point and before the ending point, where the PWM is activated to generate an increasing ramp in the starting phase and a decreasing ramp in the arrival phase. This is the soft-start/stop concept.
- **MIN\_ON\_VALUE**: read/write. It is the minimum on time for the PWM, which defines the minimum motor velocity (used only when ENA\_PWM=1), forced at start time and just before the stop time.

- **OPTIMIZE**: read/write. If =1 the rotation path is optimized, looking for alternative paths on the extra-rotation ranges (if available) in order to find the shortest one between the start and the end positions. If =0 the rotation take place only in the “standard” one-turn range (between 70 and 430 absolute degrees).
- **PRESET\_DELAY**: read/write, waiting time from the last changement of the preset knob to the rotation start. It cannot be less than 1 second or greater than 5 seconds. If you try to program out of limits values, they are forced to the lowest or highest limits.
- **POS\_OFFSET**: read/write, angular value in degrees to be added to (if positive) or to be subtracted from (if negative) the current angular position. Usefull in case of not well centered antennas, or when a re-centering is needed. Minimum value is **-70**, maximum is **70**, values outside these boundaries are software clipped to the boundaries. See also LOWLIM and UPLIM.
- **ROTATION\_CHECK**: read/write. If =1 the rotation direction (seen through the position readout changes) is continously checked. The rotation stops immediately if the position changes in the wrong direction of CHECK\_ANGLE degrees (default=2). If =0, the check isn't performed. **WARNING**: disable the ROTATION\_CHECK feature only if YOU KNOW WHAT ARE YOU DOING.
- **CHECK\_ANGLE**: read/write. See **ROTATION\_CHECK**.
- **ROTATION\_TIMEOUT**: read/write. The rotation stops immediately if the position doesn't change at least of 3 degrees in a ROTATION\_TIMEOUT seconds time interval. Minimum value measured in CBOX timer ticks (about 1.64 ms each) is 1, maximum is 65535 (corrspoding to 107 seconds). Default value is 3049 (5 seconds).
- **DIP\_SWITCH**: read only. It flags if the on-board dip-switch is enabled (=1) or not (=0)
- **CAL\_FACT**: read (modified through the commands **c**, **C**, **K**, **F** in **absolute** mode), calibration factor, used by the CBOX control program to map the antenna position sensor voltage to the 0-5V ADC range. Default value (set by command **c**) is 8000.

See **List Parameter** and **Write Parameters** for how to read-back and modify the parameters.

## Entering in Absolute mode

The controller uses an ADC which accepts 0-5V voltages. The antenna position voltage readout is mapped to be inside the 0-5V range. In the **absolute** mode the display shows directly the rotator position expressed in absolute degrees: you read values from 0 to 500. In the other operational modes (north/south-stop) the absolute readings are converted into different angular position numbers, but only for input/display purposes: internally the program works with absolute angles.

The **absolute** mode can be **only** entered turning-on the CBOX with the CW key pressed until the display shows a blinking “---“ (three minus): now you can release the key and the **absolute** mode is entered. You can move the motor using CW and CCW keys between the rotation limits. To exit from such a mode you must recycle CBOX power (forcing another mode by RS232 command isn't effective).

In the **absolute** mode the rotor can be only operated by the CW/CCW keys with the **PWM DISABLED**. The **CPM** is also disabled and cannot be enabled.

The **absolute** mode is usefull for CBOX calibration (see **CALIBRATION** paragraph). It is NOT recommended to turn big antennas which need to be started and stopped softly.

## The on-board dip-switch

There is a on-board dip-switch that allows the user to control a bit the operating mode without the intervention of a personal computer.

Remember that the dip-switch status is read only at power-up and copied into the EEPROM: you must recycle the power in order a dip-switch status changement becomes effective.

There are four switches:

- #1: select the source of the three following parameters: if ON the source is the other three switches status, otherwise the relative EEPROM contents modifiable through the **W** command
- #2: operating mode: if ON is north-stop, otherwise is south-stop
- #3: rotation range: if ON the rotation range is 0 to 500 absolute degrees, otherwise is 70 to 430 absolute degrees
- #4: PWM: if ON the PWM is enabled, otherwise is disabled

The above parameters (ENA\_PWM, LOWLIM, UPLIM, MODE) are not modifiable by RS232 commands if the switch #1 is ON (dip-switch selected). LOWLIM and UPLIM change accordingly with the POS\_OFFSET values.

### How it works with and without the PWM.

#### *Without PWM (ENA\_PWM=0)*

The motor start and stop happens immediately, abruptly applying and breaking the power supply to the motor. The PWM output is always =1. The **Soft-stop** command acts as a **Fast-stop**.

#### *With PWM (ENA\_PWM=1)*

The motor start and stop begins and ends with a ramp, from the minimum/maximum speed to the maximum/minimum ones, using the PWM output in the start/stop phases. The parameters involved with the PWM algorithm are: the repetition period, the minimum duty-cycle, the maximum duty-cycle, the RAMP interval (in degrees).

The controller software uses an interrupt routine (executed every 1.64ms) to handle the PWM (and the display too). So, all times defined in the following are measured using such an interrupt and are hence expressed in 1.64ms units.

The following parameters characterize the PWM:

- period (PWM\_PERIOD)
- max. duty-cycle (PWM\_MAX\_DUTY)
- min. duty-cycle (PWM\_MIN\_DUTY)

In the controller program are instead used the following parameters: PWM\_ON\_VALUE, PWM\_OFF\_VALUE, MIN\_ON\_VALUE and the ENA\_PWM flag. These are the parameters that must be written to (and can be read-back from) the controller.

How they are related to the above parameters:

- $PWM\_ON\_VALUE + PWM\_OFF\_VALUE = PWM\_PERIOD$
- $PWM\_ON\_VALUE = PWM\_MAX\_DUTY * PWM\_PERIOD$
- $MIN\_ON\_VALUE = PWM\_MIN\_DUTY * PWM\_PERIOD$
- ENA\_PWM: =1 PWM enabled, =0 PWM disabled

The PWM is active from the starting rotation point for **RAMP** degrees (with initial duty-cycle = PWM\_MIN\_DUTY, and final duty-cycle = PWM\_MAX\_DUTY), and from **RAMP** degrees before the arrival point until it (with initial and final duty-cycles swapped with respect to the starting situation).

In the first interval [start, start+RAMP] a linear increasing duty-cycle ramp is generated. The minimum duty-cycle at beginning must allow the motor to start.

In the other interval [arrival-RAMP, arrival] a linear decreasing duty-cycle ramp is generated. The final duty-cycle value will be PWM\_MIN\_DUTY but immediately the program will remove the power supply to the motor because the target angle has been reached.

Example: from 154° to arrive at 200° with ENA\_PWM=1 and RAMP=10. We start with the PWM active with a pulse MIN\_ON\_VALUE\*1.64 ms long. As the position approaches the 164° (154°+10°) the increasing duty-cycle arrives at its maximum value (PWM\_MAX\_DUTY) when the position is 164°. From here to 190° (200°-10°) the rotation proceed at maximum speed. From 190° to 200° the dutycycle will decrease until PWM\_MIN\_DUTY is reached at the arrival point (200°) with immediate stop and eventually small correction pulses if the position at stop differs more than 1° from the requested one.

### Parameters list

As answer to the L command, the CBOX send 19 <data>'s:

- <data1>: current angular position, (POSITION)
- <data2>: STATUS, B (rotation in progress, Busy) or R (Ready)
- <data3>: working mode (0 north-stop, 1 south-stop) (MODE)
- <data4>: lower rotation limit, absolute degrees (LOWLIM)
- <data5>: upper rotation limit, absolute degrees (HILIM)
- <data6>: after-stop/reverse delay, in 1.64ms units (REV\_DELAY)
- <data7>: PWM enabled if =1, PWM disabled if =0 (ENA\_PWM)
- <data8>: maximum PWM\_ON time, in 1.64ms units (PWM\_ON\_VALUE)
- <data9>: maximum PWM\_OFF time, in 1.64ms units (PWM\_OFF\_VALUE)

- <data10>: RAMP, distance in degrees from the start/stop points where the PWM must work (if enabled) generating two increasing/decreasing (RAMP)
- <data11>: minimum PWM\_ON value (MIN\_ON\_VALUE)
- <data12>: OPTIMIZE enabled if =1, OPTIMIZE disabled if =0
- <data13>: from last preset knob change to rotation start delay, in 1.64ms units (PRESET\_DELAY)
- <data14>: angular value offset (positive or negative) to be added to the current angular position value, min and max t.b.d. (expecting no more than a few tens of degrees). It is coded as 16-bit signed number, the sign being the MSB, so numbers from -32768 to 32767 can be represented (POS\_OFFSET)
- <data15>: ROTATION CHECK is enabled if =1, disabled if =0
- <data16>: CHECK\_ANGLE, suggested >=2
- <data17>: ROTATION\_TIMEOUT value
- <data18>: DIP-SWITCH status read-only flag: if =1 the on-board dip-switch is enabled, if =0 it is disabled
- <data19>: calibration factor (CAL\_FACT) (see commands **c**, **C**, **K**, **F** and **Calibration** paragraph)

The digits belonging to each parameter are sent starting from the most significant (ex: if LOWLIM=124 <data4> will be 00124). Five digits for numeric parameters are used because they are all 16-bit numbers as required by the controller program. POSITION and STATUS are the same read with the ? command.

## Write Parameters

There is a particular protocol to follow in order to update the parameters into the controller's EEPROM, this is because some parameters (rotation limits, PWM enabling, etc.) could be critical from the mechanical point of view and we should also avoid communication hang-ups.

As first answer to the **W** command, the controller sends:

<header><identifier><separator>**W**<separator>**B**<return> (it is Busy) if a rotation is in progress and hence no parameter modifications can be done,

or an

<header><identifier><separator>**W**<separator>**R**<return> if it is ready to proceed.

After the **R** reply the writing procedure must comply with the following procedure, the PC sends each parameter coded as: the sign "-" (minus) if required, plus five digits (leading zero's not required) followed by <return>, starting from the most significant digit (ex.: 23 can be sent as 23 or 023 or 00000023, but numbers greater than 65535 aren't allowed):

1- <header><identifier><separator>**W**<separator>**R**<return> is received

2- CBOX: sends a **N** (Next) as next parameter query

3- PC: sends the MODE value followed by <return>

The steps from 2 to 3 are repeated for the other parameters, in the order: LOWLIM, HILIM, REV\_DELAY, ENA\_PWM, PWM\_ON\_VALUE, PWM\_OFF\_VALUE, RAMP, MIN\_ON\_VALUE, OPTIMIZE, PRESET\_DELAY, POS\_OFFSET, ROTATION\_CHECK, CHECK\_ANGLE, ROTATION\_TIMEOUT.

At the end, if and only if the process has been successful for all parameters, the EEPROM content is updated with the new values. The CBOX send at the end a <return> to close the transaction.

To the handshake protocol is added a timeout: if the PC doesn't answer in about 1.64 seconds after the CBOX send the **N** the procedure is aborted at the controller side and no parameters are updated.

In the case of timeout, the controller sends a **T** followed by a <return>.

In case of the dip-switch control is active (switch #1 is OFF), the parameters MODE, ENA\_PWM, LOWLIM, UPLIM cannot be modified by the **W** command.

## Calibration

First-time calibration:

1. Enter in **absolute** mode.
2. Force the default calibration factor: command **c**.
3. Write all default-value parameters (important: rotation limits 0-500, position offset 0).
4. Rotate the motor until you read 1.400 Volts at position sensor board input. You could also apply such a voltage using other methods. This is the position voltage at 0 absolute degrees.
5. Regulate the on-board OFFSET trimmer in order to read 500mV at V1 test point TP1.
6. Regulate the on-board GAIN trimmer in order to read 1.329V at VADC test point TP2.
7. Regulate on-board OFFSET trimmer in order to read 000 on the position display.
8. Rotate the motor until a complete and precise turn is completed (use some precise references).
9. Check if the position display indicates 360: if not so, send a command **C** and now the display must show 360.

Note: you can use the command **K** if you need to calibrate at angles different from 360°. If you need to modify the calibration factor directly, without the above calibration commands **c**, **C**, you can use the **F** command.

Whenever an alignment is required (using the PC control program):

1. Turn the antenna in a well known direction: if there is some disalignment the position readout differs from the expected one.
2. Calculate:  $POSITION\_OFFSET = EXPECTED\_POSITION - READOUT\_POSITION$ . Take care with readouts in south/north-stop modes: one degree more or less change "a lot" the readout values.
3. Run the control program. Enter the  $POSITION\_OFFSET$  above calculated in the  $POS\_OFFSET$  parameter field and perform a command **W**: now the display should show the "aligned" value (check against the values noted on step 2.). Note that the rotation limits could be auto-modified in order to avoid rotation at dangerous (for the rotator hardware) angles.

Whenever an alignment is required (using the on-board OFFSET trimmer):

1. Turn the antenna in a well known direction: if there is some disalignment the position readout differs from the expected one.
2. Regulate the on-board OFFSET trimmer until the readout reach the expected value

Figure 1: PWM Ramps

