Oxford Cryosystems serial line communication protocols
home | **700 and 800 series** | GM cryocoolers

700 and 800 series: home | **protocols**

Cryostream | Cobra | N-HeliX | PheniX | PheniX-FL | Chimera | Smartstream

status packets | **command packets**

Cryostream command packets

All 700 and 800 series controllers respond to the same set of commands, with minor variations noted below. The structure of a valid command packet is as follows:

```
char    Size;       /* The variable size of the command packet */
char    Id;         /* The command packet identifier */
char    *Params;    /* Possible variable number of parameters */
```

The Size and Id parameters will match one of those listed in the table below. If a command is unrecognised (Id invalid or Size inappropriate), illegal (parameter out of range) or inappropriate (*eg* the machine has shutdown), then it is ignored. No command acknowledgement is issued, instead consult the next status packet for evidence that the command has been received.

Note: this information is available in machine-readable form in Cryostream.xml.

List of commands

Command	Size	Id	Params	Notes
Restart	2	10	none	
Ramp	6	11	HIBYTE(RampRate), LOBYTE(RampRate), HIBYTE(TargetTemp), LOBYTE(TargetTemp)	
Plat	4	12	HIBYTE(Duration), LOBYTE(Duration)	
Hold	2	13	none	
Cool	4	14	HIBYTE(TargetTemp), LOBYTE(TargetTemp)	
End	2	15	none	
Purge	2	16	none	
Pause	2	17	none	
Resume	2	18	none	
Stop	2	19	none	
Turbo	3	20	TurboOn	
SetFormat	3	40	Format	Switch to extended status packets

Parameter units and ranges

Parameter	Units	Min	Max
RampRate	K/hour	1	360
TargetTemp	cK	**[Cryostream]** 8000	**[Cryostream]** 40000 or 50000 (Plus)
Duration	minutes	1	1440 (=24 hours)
TurboOn	none	0	1
Format	none	0 (old format)	1 (extended format)

Example commands

In most of the above cases no parameters are required, and thus the command packet is a simple two-byte

one in which Size=2. For example, a command packet to stop the cooler would be created as follows:

```
/*
 * The Stop command packet, size = 2, id = 19
 */
char buf[2] = { 2, 19 };    /* Create a Stop command packet */
```

For those commands requiring parameters, the Params[] array takes various forms, illustrated by the following examples.

The Turbo command takes a single 1-byte parameter:

```
/*
 * The Turbo command packet, size = 3, id = 20
 * The Params[] array consists of a single char taking the value either
0
 * (switch Turbo off) or 1 (switch Turbo on)
 */
char buf[3] = { 3, 20, 1 };        /* Switches Turbo on */
```

The Ramp, Plat and Cool commands are a little more complicated, because the parameters which are passed are shorts - 16 bit integers. These need to be assembled as illustrated in the following examples, which use the standard HIBYTE and LOBYTE macros reproduced below.

```
/*
 * The Plat command packet, size = 4, id = 12
 * The Params[] array consists of a short containing the duration of
the
 * Plat in minutues
 */
char buf[4] = { 4, 12, HIBYTE(720), LOBYTE(720) }; /* 720 minute
plateau */

/*
 * The Cool command packet, size = 4, id = 14
 * The Params[] array consists of a short containing the end
temperature in centi-Kelvin
 */
char buf[4] = { 4, 14, HIBYTE(17000), LOBYTE(17000) }; /* Cool to 170 K
*/

/*
 * The Ramp command packet, size = 6, id = 11
 * The Params[] array consists of a short containing desired ramp rate
in K/hour,
 * followed by a short containing the end temperature in centi-Kelvin
 */
char buf[6] = { 6, 11, HIBYTE(120), LOBYTE(120), /* Rate = 120 K/hour
*/
                    HIBYTE(25050), LOBYTE(25050), /* Final temp = 250.5
K */ };
```

HIBYTE

```
#define HIBYTE(w) ((unsigned char)(((unsigned short)(w) >> 8) & 0xFF))
```

LOBYTE

```
#define LOBYTE(w) ((unsigned char)(w))
```