DATABASE UPLOAD/DOWNLOAD COMMANDS

DATABASE UP/DOWNLOAD COMMANDS

The first five to thirteen bytes of an M Protocol frame are discussed on pages six through 10 of the Pelco Millennium System Protocol Manual, and will not be discussed further here. The following material relates to the data portion of an "M" frame being sent to upload or download database data between the manager software/device and the network manager/switch. Note that this document is an addendum to the manual entitled "Millennium Commands".

Within the data section of the frame, the first two bytes for an up/download command are always $00C2. The first byte denotes the command as a universal command type, and the second as a database up/download sub-type.

The third byte refers to whether the command is a get or set command, or a query of the receiving device's readiness to receive:

$00 = set
$01 = get
$02 = get database size
$03 = are you ready

The fourth and fifth bytes refer to the command type:

$0000 = ports
$0001 = internal alarms
$0002 = external alarms
$0003 = alarm group
$0004 = macros
$0005 = sequences
$0006 = daily timer
$0007 = special timer
$0008 = weekly timer
$0009 = camera information
$000A = monitor information
$000B = time/date information
$000C = keyboard to monitor access
$000D = camera to keyboard access
$000E = camera to monitor access
$000F = keyboard priority
$0010 = auxiliary information
$0011 = physical/logical camera numbers
$0012 = video loss

The sixth and seventh bytes refer to the record number, and are used to denote data specific to the command type in question (discussed later).

The eighth and ninth bytes refer to the record size and these denote the total size in bytes of the data section of the “M” frame in question.

The rest of the bytes prior to the check sum are specific in number and content to the command type and are the subject of the remainder of this document.

3.12.5 PORTS [0000]

Format (byte number for reference only and NOT actual byte number for the frame)

Command Group (the "M" frame data section is further broken into command group, command, and data sections as seen on page 11 of the Pelco Millennium Systems Protocol Manual)

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $0000 = *port command*

Bytes 6 & 7: $xxxx = *record number* (port number in this command)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes within the command)

Byte 10: $00 = 232 (*port communication type*)
$01 = 485
$02 = 422

Byte 11: $00 = ASCII protocol (*protocol type* for the port)
$01 = poled ASCII protocol
$02 = pan&tilt/P protocol
$03 = pan&tilt/D protocol
$04 = MUX protocol
$05 = GPI/P protocol
$06 = alarm/P protocol
$07 = MDA protocol
$08 = M protocol

Byte 12: $00 = 1200 baud (*baud rate* for the port)
$01 = 2400 baud
$02 = 4800 baud
$03 = 9600 baud
$04 = 19200 baud
$05 = 28800 baud
$06 = 36600 baud
$07 = 56k baud

$08 = 115k baud

Byte 13: $00 = even parity (*port parity*)
$01 = odd parity
$02 = no parity

Byte 14: $00 = 7 bits (*data bits*)
$01 = 8 bits

Byte 15: $00 = one (*stop bits*)
$01 = two

3.12.6 INTERNAL ALARMS [0001]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $0001 = *internal alarm command*

Bytes 6 & 7: $xxxx = *record number* (internal alarm number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Byte 10: $00 = disabled (*alarm enabled/disabled*)
$01 = enabled

Byte 11: $00 = auto (*ack type*)
$01 = manual

Byte 12: $00 = normally closed (*contact type*)
$01 = normally open

Byte 13: $00 = no priority (*alarm priority*)
$01 = priority

Byte 14: $xx = *time-out* (number of seconds converted to hexadecimal)

Byte 15: $xx = *number of groups* (maximum number of monitors/8)
(number of bytes for group enable)

Bytes 16 & 17: $xxxx… = *group enable* (bit mapped to monitor group)
(0 = off, 1 = on for each group)
(number of bytes = number of groups)

Byte 18: $xx = *number of steps* (all that follows are part of the step)

(the step is looped this many times)

Bytes *19* & *20*: $xxxx = *physical camera number*

Byte *21*: $xx = *dwell* (dwell time in seconds converted to hex)

Byte *22*: $00 = pattern (*command*)
$01 = preset
$02 = random scan
$03 = frame scan
$04 = stop scan
$05 = null command

Byte *23* & *24*: $xxxx = *command number*

Byte *25*: $00 = null command (*auxiliary*)
$01 = local aux. on
$02 = local aux. off
$03 = global aux. on
$04 = global aux. off
$05 = camera aux. on
$06 = camera aux. off

Bytes *26* & *27*: $xxxx = *auxiliary number*
:
:

3.12.7 EXTERNAL ALARMS [0002]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $0002 = *external alarm command*

Bytes 6 & 7: $xxxx = *record number* (external alarm number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Byte 10: $00 = disabled (*alarm enabled/disabled*)
$01 = enabled

Byte 11: $00 = auto (*ack type*)
$01 = manual

Byte 12: $00 = normally closed (*contact type*)
$01 = normally open

Byte 13: $00 = no priority (*alarm priority*)
$01 = priority

Byte 14: $xx = *time-out* (number of seconds converted to hexadecimal)

Byte 15: $xx = *number of groups* (number of monitors/8)

Byte 16 & 17: $xxxx… = *group enable* (bit mapped to monitor group)
(0 = off, 1 = on for each group)
(number of bytes = number of groups)

Byte 18: $xx = *number of steps* (all that follows are part of the step)

(the step is looped this many times)

Bytes *19* & *20*: $xxxx = *physical camera number*

Byte *21*: $xx = *dwell* (dwell time in seconds converted to hex)

Byte *22*: $00 = pattern (*command*)
$01 = preset
$02 = random scan
$03 = frame scan
$04 = stop scan
$05 = null command

Byte *23* & *24*: $xxxx = *command number*

Byte *25*: $00 = null command (*auxiliary*)
$01 = local aux. on
$02 = local aux. off
$03 = global aux. on
$04 = global aux. off
$05 = camera aux. on
$06 = camera aux. off

Bytes *26* & *27*: $xxxx = *auxiliary number*
:
:

3.12.8 ALARM GROUPS [0003]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $0003 = *alarm group command*

Bytes 6 & 7: $0000 = *record number* (all group info is always sent)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Bytes 10 & 11: $xxxx = *number of monitors* (number of steps)
(all that follows are part of the step)
(the step is looped this many times)

Byte *12*: $xx = *group* (from *a* to *p* in bit position) ($00 = a, $15 = p)

Byte *13*: $00 = B monitor (block) (*group type*)
$01 = Q monitor (queue)
$02 = D monitor (display)
$03 = L monitor
$04 = M monitor
$05 = S monitor (salvo)
:
:

3.12.9 MACROS [0004]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $0004 = *macro command*

Bytes 6 & 7: $xxxx = *record number* (macro number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Bytes 10 & 11: $xxxx = *number of steps* (the step is looped this many times)

Byte 12: $xx = *step size* (the number of bytes in each step loop)
(all that follows are part of the step)

Bytes *13* & *14*: $xxxx = *monitor number*

Bytes *15* & *16*: $xxxx = *physical camera number* ($0000 = no camera)

Byte *17*: $xx = *dwell* (dwell time in seconds converted to hex)

Byte *18*: $00 = local auxiliary on (*command*)
$01 = local auxiliary off
$02 = global auxiliary on
$03 = global auxiliary off
$04 = camera auxiliary on
$05 = camera auxiliary off
$06 = sequence command
$07 = switch command
$08 = arm alarm
$09 = disarm alarm
$0A = pattern command

$0B = preset command
$0C = random scan
$0D = frame scan
$0E = stop scan
$0F = null command

Bytes *19* & *20*: $xxxx = *command number* (pattern or preset number)

:

:

3.12.10 SEQUENCES [0005]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $0005 = *sequence command*

Bytes 6 & 7: $xxxx = *record number* (sequence number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Bytes 10 & 11: $xxxx = *number of steps* (the step is looped this many times)

Byte 12: $xx = *step size* (the number of bytes in each step loop)
(all that follows are part of the step)

Bytes *13* & *14*: $xxxx = *physical camera number* ($0000 = no camera)

Byte *15*: $xx = *dwell* (dwell time in seconds converted to hex)

Byte *16*: $00 = pattern command (*command*)
$01 = preset command
$02 = random scan
$03 = frame scan
$04 = stop scan
$05 = null command

Byte *17* & *18*: $xxxx = *command number* (pattern or preset number)

Byte *19*: $00 = null command (*auxiliary command*)
$01 = local auxiliary on
$02 = local auxiliary off
$03 = global auxiliary on

$04 = global auxiliary off
$05 = camera auxiliary on
$06 = camera auxiliary off

Bytes *20* & *21*: $xxxx = *auxiliary number* ($0000 = no auxiliary)

:

:

3.12.11 DAILY TIMERS [0006]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $0006 = *daily timer command*

Bytes 6 & 7: $xxxx = *record number* (daily timer number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Byte 10: $00 = disabled (*timer enabled/disabled*)
$01 = enabled

Byte 11: $xx = *hour* (00 to 23 converted to hex)

Byte 12: $xx = *minute* (00 to 59 converted to hex)

Bytes 13 & 14: $xxxx = *macro number* ($0000 = no macro)

Bytes 15 & 16: $xxxx = *number of monitors* (number of steps)
(all that follows are part of the step)

Byte *17*: $xxxx = *sequence number* (for the monitor # matching the current step #)
:
:

3.12.12 SPECIAL TIMERS [0007]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $0007 = *special timer command*

Bytes 6 & 7: $xxxx = *record number* (special timer number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Byte 10: $00 = disabled (*timer enabled/disabled*)
$01 = enabled

Byte 11: $xx = *hour* (00 to 23 converted to hex)

Byte 12: $xx = *minute* (00 to 59 converted to hex)

Byte 13: $xx = *day of the month* (01 to 31 converted to hex)

Byte 14: $xx = *month* (01 to 12 converted to hex)

Byte 15: $xx = *year* (00 to 99 converted to hex)

Bytes 16 & 17: $xxxx = *macro number* ($0000 = no macro)

Bytes 18 & 19: $xxxx = *number of monitors* (number of steps)
(all that follows are part of the step)

Byte *20*: $xxxx = *sequence number* (for the monitor # matching the current step #)
:
:

3.12.13 WEEKLY TIMERS [0008]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $0008 = *weekly timer command*

Bytes 6 & 7: $xxxx = *record number* (weekly timer number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Byte 10: $00 = disabled (*timer enabled/disabled*)
$01 = enabled

Byte 11: $xx = *hour* (00 to 23 converted to hex)

Byte 12: $xx = *minute* (00 to 59 converted to hex)

Bytes *13* & *14*: $xxxx = *macro number/day* ($0000 = no macro)
: (loops 7 times) (Sunday through Saturday)
:

Bytes 27 & 28: $xxxx = *number of monitors* (step size)
(the following is a nested loop)
(outer loop is 7 times, one for each day)
(inner loop runs to match the number of monitors)

Bytes *29* & *30*: $xxxx = *sequence number/day* (for the monitor # matching the step #)
:
:

3.12.14 CAMERAS [0009]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $0009 = *camera command*

Bytes 6 & 7: $xxxx = *record number* (physical camera number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Byte 10: $00 = off (*vertical drive*)
$01 = on

Byte 11: $00 = extended coaxial (*R/X type*)
$01 = standard coaxial
$02 = PTZ A
$03 = PTZ B

Byte 12: $xx = Port Address (*1 – 32 converted to hex*)

Bytes *13 – 32*: $xx = (*camera title*) (20 characters)(the terminator is assumed, not sent)
(allowable: $30 - $39 = 0 – 9; $41 - $5A = A – Z; $20 = blank; $00 = terminator)

3.12.15 MONITORS [000A]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $000A = *monitor command*

Bytes 6 & 7: $xxxx = *record number* (monitor number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Byte 10: $00 = 1 (*brightness*)
$01 = 2
$02 = 3
$03 = 4
$04 = 5
$05 = 6
$06 = 7
$07 = 8

Byte 11: $xx = number of titles (*number of steps*) (titles and their x and y positions)

Byte *12*: $00 = disabled (*time/date display*)
$01 = enabled

Byte *13*: $xx = xpos (*position of title on x-axis*)

Byte *14*: $xx = ypos (*position of title on y-axis*)

Byte *15*: $00 = disabled (*monitor title*)
$01 = enabled

Byte *16*: $xx = xpos (*position of title on x-axis*)

Byte *17*: $xx = ypos (*position of title on y-axis*)

:

:

*For Project 68 the title enable/disables and x and y positions will be as follows:
Camera number (byte 12), xpos (13), ypos (14); **Camera Title** (15), xpos (16), ypos (17); **Rx Type** (18), xpos (19), ypos (20); **Monitor number** (21), xpos (22), ypos (23); **Monitor Status** (24), xpos (25), ypos (26); **Time** (27), xpos (28), ypos (29); **Date** (30), xpos (31), ypos (32).

Note: In Project 68 only the first xpos and ypos are actually used for positioning the titles. In future products each title may be able to be positioned separately, but not in Project 68. Bytes 16, 17, 19, 20, 22, 23, 25, 26, 28, 29, 31, and 32 will all be set to $00, and all of these bytes will then be skipped in parsing for Project 68.

3.12.16 TIME/DATE [000B]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $000B = *daily timer command*

Bytes 6 & 7: $0000 = *record number* (system-wide, so always $0000)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Byte 10: $xx = *second* (00 to 59 converted to hex)

Byte 11: $xx = *minute* (00 to 59 converted to hex)

Byte 12: $xx = *hour* (00 to 23 converted to hex)

Byte 13: $xx = *day* (01 to 31 converted to hex)

Byte 14: $00 = Sunday (*day of week*)
$01 = Monday
$02 = Tuesday
$03 = Wednesday
$04 = Thursday
$05 = Friday
$06 = Saturday

Byte 15: $xx = *month* (01 to 12 converted to hex)

Byte 16: $xx = *year* (00 to 99 converted to hex)

Byte 17: $xx = *century* (19, 20, 21 etc. converted to hex)

Byte 18: $00 = MMDDYY (*date format*)
$01 = DDMMYY
$02 = YYMMDD
$03 = MMMDDYY
$04 = DDMMMYY
$05 = YYMMMDD

Byte 19: $00 = 24 hr./military (*time format*)
$01 = am/pm

3.12.17 KEYBOARD TO MONITOR ACCESS [000C]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $000C = *keyboard to monitor access command*

Bytes 6 & 7: $xxxx = *record number* (keyboard port number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Bytes 10 & 11: $xxxx = *number of monitors* (max. monitors) (number of steps)

Byte 12: $xx = *keyboards per port* (at maximum) (number of bits per step)

Byte *13*: $xx… = *monitor to keyboard access* (keyboards are bit mapped to monitors, where monitor number = step number) (0 = no access, 1 = access for that keyboard)

:
:

3.12.18 CAMERA TO KEYBOARD ACCESS [000D]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $000D = *camera to keyboard access command*

Bytes 6 & 7: $xxxx = *record number* (physical camera number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Byte 10: $xx = *keyboards per port* (at maximum) (number of steps)

Byte 11 & 12: $xxxx = *port number* (for camera number matching the record number)

Byte *13*: $00 = no access (*access information*) (for keyboard number matching
$01 = full access the step number)
$02 = view video only

:
:

3.12.19 CAMERA TO MONITOR ACCESS [000E]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $000E = *camera to monitor access command*

Bytes 6 & 7: $xxxx = *record number* (physical camera number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Bytes 10 & 11: $xxxx = *number of monitors* (at maximum) (max. mon./8 = bytes for bitmap)

Bytes *12* & *13*: $xxxx… = *camera access information*
(monitors are bit mapped to cameras, where lower byte = mon. 1-8, higher byte = mon. 9-16, etc.)
(0 = no access, 1 = access for that camera)

3.12.20 KEYBOARD PRIORITY [000F]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $000F = *keyboard priority command*

Bytes 6 & 7: $xxxx = *record number* (keyboard port number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Byte 10: $xx = *keyboards per port* (at maximum) (number of steps)

Byte *11*: $xx = *keyboard priority* (from $00 to $xx, where $xx = kbds/port)
($01 = highest priority, $00 = no priority)

:
:

3.12.21 AUXILIARIES [0010]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $0010 = *auxiliaries command*

Bytes 6 & 7: $xxxx = *record number* (auxiliary number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Byte 10: $00 = alarm (*mode*)
$01 = latched
$02 = momentary

Bytes 11 & 12: $xxxx = *monitor number* (from $0000 to max. monitors)
($0000 = no monitor)

Byte 13: $xx = *dwell* (00 to 99 seconds converted to hex) ($00 = no dwell)

3.12.22 PHYSICAL/LOGICAL CAMERA NUMBERS [0011]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $0011 = *physical/logical camera numbers command*

Bytes 6 & 7: $xxxx = *record number* (physical camera number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Bytes 10 & 11: $xxxx = *logical camera number* (for camera matching the record number)

3.12.23 VIDEO LOSS [0012]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $0012 = *video loss command*

Bytes 6 & 7: $xxxx = *record number* (physical camera number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Byte 10: $00 = off (*enabled*)
$01 = on

Byte 11: $00 = auto (*ack type*)
$01 = manual

Byte 12: $xx = *time-out* (00 to 99 seconds converted to hex)

Byte 13: $xx = *number of camera groups* (max. monitors/8)
(number of bytes for group enable)

Bytes 14 & 15: $xxxx… = *group enable* (bit mapped to monitor group)
(0 = off, 1 = on for each group)

3.12.24 KEYBOARD BUTTONS [0013]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $0013 = *keyboard buttons database*

Bytes 6 & 7: $xxxx = *record number* (physical key number)

Bytes 8 & 9: $0006 = *record size* (number of data bytes in record)

Byte 10: $xx = *key definition* **(see chart below) (example = RELAY)**

Byte 11,12: $xxxx = *key number* **(unsigned word)**
Supports key definition (from the example above 128 = RELAY 128)

Byte 13: $xx = *key flags*
Bit 0 **1 = Key Enabled, 0 = Key Disabled**
Bit 1 **1 = LED always ON, 0 = LED always OFF**
Bit 2 **1 = LED lit/unlit when key pressed**
Bit 3 **1 = BEEP when key pressed**
Bit 4 **1 = Suppress release, 0 = Act on release of key**
Bit 5 **1 = Suppress press, 0 = Act on press of key**

Key Definition Chart (Preliminary – This may change)

CONSTANT	NAME	Description
0 (0x00)	NOTDEF	Key is not defined
1 (0x01)	NUM0	Numerical value 0
2 (0x02)	NUM1	Numerical value 1
3 (0x03)	NUM2	Numerical value 2
4 (0x04)	NUM3	Numerical value 3
5 (0x05)	NUM4	Numerical value 4
6 (0x06)	NUM5	Numerical value 5
7 (0x07)	NUM6	Numerical value 6
8 (0x08)	NUM7	Numerical value 7
9 (0x09)	NUM8	Numerical value 8

10	(0x0A)	NUM9	Numerical value 9
11	(0x0B)	ESC	Escape – acts like exit key
12	(0x0C)	BS	Backspace when editing ASCII strings or time/date
13	(0x0D)	ENTER	Enter – has various functionality
14	(0x0E)	MON	Monitor Menu
15	(0x0F)	MONX	Select Monitor X
16	(0x10)	CAM	Camera Menu
17	(0x11)	CAMG	?
18	(0x12)	AUX	Defines a key as one of the icon keys, uses numbers 1 – 8
19	(0x13)	PRESET	Go to preset menu
20	(0x14)	PRESETX	Call preset X
21	(0x15)	LOCK	Monitor Lock
22	(0x16)	GPI	GPI menu
23	(0x17)	CAMFWD	Switches next camera to current monitor
24	(0x18)	CAMBWD	Switches previous camera to current monitor
25	(0x19)	ALM	Go to alarm menu
26	(0x1A)	ALMX	Go to alarm menu, current alarm X
27	(0x1B)	RSTALLALM	Send Reset all alarms
28	(0x1C)	CLEAR	Clears preset or resets alarm (depending upon menu)
29	(0x1D)	MAC	Go to macro menu
30	(0x1E)	MACX	Go to macro menu, select macro X
31	(0x1F)	MAC_PAUSE	Pause a macro
32	(0x20)	DEFINE	Goes to define PIN menu
33	(0x21)	FLIP	Calls preset 33.
34	(0x22)	CAMX	Switch camera x to current monitor.
35	(0x23)	GPIX	Select GPI X and go to GPI menu.
36	(0x24)	ZOOM_IN	Zoom current camera lens in.
37	(0x25)	ZOOM_OUT	Zoom current camera lens out.
38	(0x26)	FOCUS_NEAR	Focus current camera lens near.
39	(0x27)	FOCUS_FAR	Focus current camera lens far.
40	(0x28)	IRIS_OPEN	Open current camera iris.
41	(0x29)	IRIS_CLOSE	Close current camera iris.
42	(0x2A)	MENU	?
43	(0x2B)	ARM_DISARM	Arm or disarm an alarm
44	(0x2C)	VCR	Go to VCR menu
45	(0x2D)	TXTCONCEAL	?
46	(0x2E)	IDONOFF	?
47	(0x2F)	RELAY	Turn relay X on
48	(0x30)	RECALL	Recall previous camera on monitor
49	(0x31)	ALT	?
50	(0x32)	TURBO	Turbo pan speed when pressed
51	(0x33)	DATE_TIME	?
52	(0x34)	FNC	?
53	(0x35)	MENUFWD	Next page of menu
54	(0x36)	MENUBWD	Previous page of menu or exit to top menu
55	(0x37)	CAMAUX	Send a camera aux
56	(0x38)	CAMPATTERN	Call a pattern
57	(0x39)	RSTALM	? (must be in alarm menu)
58	(0x3A)	MACG	?
59	(0x3B)	ALTBWD	?

3.12.25 ADD CHARACTERS [0014]

Format

Command Group

Byte 1: $00 = *universal command type*

Command

Byte 2: $C2 = *database up/download command*

Data

Byte 3: $00 = *set*
$01 = *get*
$02 = *get database size*

Bytes 4 & 5: $0014 = *Add characters command*

Bytes 6 & 7: $0000 = *record number* (Always 0000, no record number)

Bytes 8 & 9: $xxxx = *record size* (number of data bytes)

Byte 10 & 11: $xxxx = *Number of characters*. (Maximum $FF for Project 68)

Byte *12*: $xx = *ASCII code for the characters.*
:
: