# C-Bus Application Messages & Behaviour

# Chapter 8 – Enable Control

Document Number: CBUS-APP/08
Issue: 1.3
Date: 13 February 2007

**Comments on this document should be addressed to:**

**Engineering Manager**
**Clipsal Integrated Systems**
**PO Box 103 Hindmarsh**
**South Australia 5007**

## TABLE OF CONTENTS

# 8 ENABLE CONTROL APPLICATION

## *8.1 Application ID*

$CB

Note that to ensure consistent operation of C-Bus networks, this Application ID shall not be reassigned.

## *8.2 Description*

The Enable Control Application is used to set one or more shared C-Bus Network Variables. Devices on the bus (and which can accept Enable Control Messages) take some defined set of actions, based on the value of the shared Network Variable(s). These actions are dependent on the device using the Network Variable. They may lead to the generation of other C-Bus messages.

Devices which respond to Enable Control Application Messages can generally be programmed in some manner, so that exact values of the Network Variables and the actions taken can be set up at the time the device is installed.

A typical device using the Event Control Application is a scheduling system, where different schedules can be selected using the Network Variables. Complex scheduling can be created using several of these devices on the same network.

For compatibility with C-Bus Lighting Application messages, the messages used for the Enable Control Application look like a Lighting Application instantaneous ramp to level command.

## *8.3 Document Convention*

Numbers are shown in decimal (base ten) with no other special prefixes or indications.

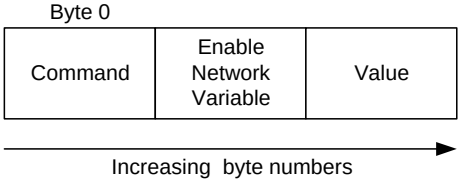
Binary numbers (base 2) are shown with the prefix %.

Hexadecimal numbers (base 16) are shown with the prefix $.

Example: 157 = %10011101 = $9D

## *8.4 Message Structure*

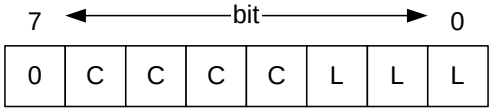
Enable Control Application messages are always fixed at 3 bytes long, and have the form:



### *8.4.1 Command*

The command is a byte, broken into bit-fields to support encoding of a command and the number of bytes following as parameters.

The command format is:




Where “C” represents a bit of a command, and “L” represents a bit of the length.

This command format provides compatibility with the C-Bus Lighting Application, and is therefore suitable for backward compatibility with older devices and interoperability with lighting units.

For the Event Control Application, the Command byte format is:

SET NETWORK VARIABLE[1]: %0 xxxx 010

The bits “xxxx” can have any value, however 0000 is recommended for all new designs. There is no guarantee that the use of values other than 0000 will be supported in future.

The (3 bit) length field reflects the number of arguments, which is always 2.

All other possible encodings of the Command byte are reserved.

### *8.4.2 Enable Network Variable*

The Enable Network Variable[2] is a byte, used to represent the number of the Network Variable being set.

Setting Network Variables not used by a device has no effect.

The number of Enable Network Variables is device dependent.

The following convention is used:

Enable Network Variable:

| | |
|---|---|
| Size: | 8-bit byte |
| Range: | $00 .. $FF |
| Special Cases: | The value $FF is reserved as a wildcard. If used, all Network Variables in all devices that accept Enable Control messages shall be set to the same value. |

[1] This command corresponds to a Lighting Application RAMP command.

[2] Corresponding to the Lighting Application Group Address.

### *8.4.3 Value*

The Value[3] field is used to set the value of the specified Enable Network Variable.

Devices use the values of the Network Variables they support to perform actions, as determined by the device and its configuration.

The following convention is used:

Value:

| | |
|---|---|
| Size: | 8-bit byte |
| Range: | $00 .. $FF |
| Special Cases: | None. |

## *8.5 Programming of Devices*

C-Bus devices that accept Enable Control Application messages should have some means of configuring the Network Variable numbers, and values, to which they respond and the various responses they can generate.

This programming may performed using:

a. some kind of C-Bus data load facility;

b. panel programming buttons or keys;

c. participation in C-Bus Lighting Learn mode;

d. listening for C-Bus network traffic as part of a dedicated learn function;

e. a direct connection from a programming device (for example, a PC); or

f. some other technique at the discretion of device manufacturer.

## *8.6 Message Priority*

Enable Control Application messages shall always be transmitted at Class 4 (lowest) priority.

## *8.7 Internetwork Routing*

Enable Control Application messages may be routed via one or more C-Bus bridges or gateway devices. Such messages will be received with a message type indicating point-multipoint, but will have a non-zero Network routing.

The Network routing information is irrelevant, as responses are not required for Enable Control Application messages.

## *8.8 Application Behaviour*

A device that receives and handles Enable Control Application messages will perform some kind of actions, either immediately or in the future. These actions depend on the nature of the device.

[3] Corresponding to a Lighting Application Level, when used in a RAMP command.

There can be many Enable Control Application devices on a C-Bus network. These devices shall support at least:

a. sharing of common Network Variables;

   In this case, several devices would be programmed to respond on the same settings of a Network Variable. This permits extension of the number of actions beyond whatever limits are imposed by a single device

b. separation of Network Variables;

   Devices in the Enable Control Application would respond to different Network Variable numbers. The devices are separate and completely unrelated.

### *8.8.1 Concatenated Commands*

An Enable Control Application device may receive a message containing more bytes than a single command. This permits a single C-Bus transmission to contain multiple commands for a single application.

Devices using C-Bus Enable Control Application messages must process all received bytes. This is achieved by placing the received bytes in a buffer, and using the following simple algorithm:

```
WHILE the buffer contains bytes LOOP
    The first byte defines the command type and argument
    count (refer section 8.4).
    Process the first (command) byte and its arguments
    Once processed, remove the command and argument bytes
    from the buffer
END LOOP
```

### *8.8.2 State Preservation*

Devices which respond to Enable Control Application messages shall include capabilities to preserve the setting of Enable Network Variables across power failures. On power failure, these devices shall store the value of the network variable(s) they respond to. On power up, these devices shall restore and use the preserved network variable value(s).

Devices which transmit Enable Control Application messages, but do not respond to them or use the Enable Network Variable values, do not need to preserve the network variable values. However, on power-up, if a condition exists that would result in the transmission of an Enable Control Application message, then this message shall be sent.

## *8.9 Status Reporting*

Devices using Enable Control Application messages shall not respond to C-Bus status requests (MMI) issued against the Enable Control Application Address.

### *8.10 Limitations*

It is not possible to retrieve the value of a Network Variable which has been set on the Enable Control Application. This means that any device making use of the Enable Control Application Network Variables must maintain its own state.

### *8.11 Examples*

Refer to section 8.4 (Page 3).

These examples assume that a device is causing operating an Enable Control by issuing a message. This device interfaces to C-Bus using the C-Bus Serial Interface, which is described in more detail in CBUS-SIUG.

The examples assume the Serial Interface SRCHK option is set, so that data transfer both to and from the Serial Interface uses a checksum.

To set the value of Enable Control Network Variable $37 to $82, a device would issue:

To PCI: \05CB0002378275

To perform the same operation on a remote network (through a single bridge with unit address $56 on the side of the sending device, and unit address $37 on the side of the meter measurement device), a device would issue:

To PCI: \03**5609**CB02378216

The internetwork routing bytes ($5609) would be modified by the bridge as the message passed through, to construct the reverse route. The receiving device(s) can ignore this, as the reverse route is not important.

### *8.12 Notes*

Because an Enable Control Application message is the same format as a Lighting Application instantaneous Ramp to Level message, standard C-Bus Lighting input devices can be used to set operation of devices which use the Enable Control Application.