

## User Manual v1.5 (dated 26/05/2022)

### AirGlu2™ module

### Wireless Sync and Control

#### Key Features

The AirGlu2™ module highlighted features are listed below:

- Configurable Sync and Control module for sync and control of professional cameras, recorders and audio devices
- Built in Timecode Generator, with 0.5ppm VC TCXO
- Timecode Input and Timecode output modes
- Master or Slave operation modes
- Genlock and Word Clock sync outputs
- Sync and Control wireless using sub-GHz protocol, transmit and receive.
- External antenna port
- Additional connectivity to apps, using 2.4GHz Bluetooth LE.
- 2.0v and 3.3v regulated inputs
- UART to host equipment with buffered data inputs
- Simple serial UART API for all menu and control functions
- Single power and I/O port
- Surface Mount module

# Timecode Systems

GENERATE. SYNC. SHARE

## System Overview

### Introduction

The AirGlu™ module is an energy friendly, compact solution for providing wireless sync and control capabilities to a professional camera, recorder or audio device.

It combines integrated MCU/2.4GHz transceiver, FPGA and Sub GHz radio transceiver.

A highly accurate 0.5ppm TCXO provides stable timecode and sync outputs.

It can communicate via Sub GHz radio to other AirGlu™ and AirGlu2™ modules or to 3<sup>rd</sup> party applications using the 2.4GHz Bluetooth LE radio.

It has an external internal antenna port and can be surface mounted into a host PCB.

It is extremely compact at 22 mm x 16 mm.

## System Overview

### Serial UART API – S1C Commands

#### API V5.4

#### Getting/Changing Configuration Settings of S1C enable devices

All commands are formed normally using text-based strings and communicated to the device by one of the following methods.

- A bi-directional serial link. The link will currently be based around a logic level serial port with or without CTS/RTS handshake with a default baud rate of 57600. Note that TCS standard serial I/O will not tolerate full RS232 voltage levels, also RS232 has inverse logic.
- For Bluetooth Low Energy enabled devices.

The method to be used to read or change values with the System One Controller (S1C) using simple control request messages. These requests can be to retrieve information or to change settings. The S1C will respond back the requested data or acknowledgment of changes made.

All control messages and replies can be checked using simple terminal software, i.e. Teraterm for serial links

#### Command Syntax

Commands and replies are formed as readable ASCII strings.

Commands/queries/replies start with a '#', this start character is followed by the command/query string which is then followed by a '?' for queries or an '=' for commands/replies.

All commands and queries are case sensitive, currently all upper-case and 4 ASCII characters long - excluding '#', '=' or '?'.

All commands and queries must be terminated with a <LF> or a <CR>+<LF> (0x0a or 0x0d,0x0a)['\n' or '\r\n']. All non-binary replies will be terminated with a single <LF>

Command value strings must not exceed 112 bytes to be compatible with the BLink network.

## Binary Commands

Commands or queries requiring binary data are identified with different assignment characters as follows.

'>' Binary set (same as text command's '=')

'<' Binary get (same as text command's '?')

Following the assignment character is a single length byte which denotes the length of the following binary data. If there is no binary data to follow then this length byte should be zero. The length byte should then be followed by the first byte of the binary data. The binary data (or zero length byte) still requires to be followed with a '\n' termination character.

```
"#BCMD>Length+Binary Data\n"
```

## Multiple Commands per line

Multiple commands can be placed on a single command line. This is most useful for BLink commands/queries to a single remote device as multiple commands on a single line will be transmitted to the remote device in a single frame period i.e. quicker. Multiple commands should be separated with the ':' (full colon) character. Each **line**, not each command, should be started with the '#' character and terminated with a '\n'.

```
"#TCTM?:TCUB?:RFTX=1:BINC>\04***\n"
```

Replies to multiple commands may be returned separately.

It must be remembered that there is a line length limit of 112 characters.

## BLink Network Commands

BLink network device addressing is slightly different, please see later section on 'Blink Networks' for full details.

# Timecode Systems

GENERATE. SYNC. SHARE

## Timecode Source

**TCSC?** Returns the current timecode source as a string TCSC=n where n is an ASCII '0' through '2' where;

0 = Internal (free run)

1 = External RF

2 = External RF (Cont) Free run between sync packets.

3 = External LTC

4 = External LTC (Cont) Free run if signal removed

**TCSC=n** Sets the timecode source to the value n as in the list below. The S1C will reply with the currently set source where;

0 = Internal

1 = External RF

2 = External RF (Cont)

3 = External LTC

4 = External LTC (Cont)

5 = Jam once to RF then go to Internal mode

6 = Jam once to LTC then go to Internal mode

## Timecode

**TCTM?** Returns the current timecode as a string

"TCTM=hhmmssff"

**TCTM=hhmmssff** Sets the current Timecode (only if in 'Internal' Mode)

## User-Bits (also see RF Userbits)

**TCUB?** Returns the current User Bits

**TCUB=uuuuuuu** Sets the current User Bits (only if in 'Internal' Mode or RF userbits switch (RFUB) is off)

## Broadcast Timecode \*

**TCBC=n** Will configure the unit to broadcast on the serial output a '#TCTM=hhmmssff' string periodically.

**TCBC=0** Turn off broadcast

**TCBC =1** Send Timecode once a second on frame 10. Frame 10 allows easy offset adjustment to what is used/displayed.

**TCBC=2** Send Timecode every frame.

## Timecode Run/Freeze \*

When the Timecode source is set to a non-continuous mode the output Timecode will 'freeze' until the incoming signal is seen, triggering a running the Timecode.

Whereas continuous modes will continue to run the Timecode in the absence of and external signal and then 'soft-lock' when the signal is re-established. These functions can override the standard behaviour.

**TCRN?** Returns whether Timecode is running (=1) or frozen (=0).

# Timecode Systems

GENERATE. SYNC. SHARE

TCRN=n Sets the Timecode to run (n=1) or freeze (n=0).

## Frame Rate \*

TCFR? Returns the current frame rate (Frames per Second) and whether using Drop-Frame encoding of the timecode data. Returns TCFR=n,d where n is the FPS value\*1001 and d is a flag of whether using drop-frame encoding or not;

24000,0 = 23.98 fps (Non Drop-Frame)

24024,0 = 24 fps

25025,0 = 25 fps

30000,0 = 29.97 fps

30000,1 = 29.97 fps Drop-Frame

30030,0 = 30 fps

30030,1 = 30 fps Drop-Frame

The function will accept double frame rates, as follows but will return the standard rate value in response and for subsequent queries.

48000,0 = 47.96 fps (Non Drop-Frame)

48048,0 = 48 fps

50050,0 = 50 fps

60000,0 = 59.94 fps

60000,1 = 59.94 fps Drop-Frame

60060,0 = 60 fps

60060,1 = 60 fps Drop-Frame

TCFR=n,d Sets the current frame rate & drop frame to the values n,d as in the list above. The return value must be checked as illegal values will be ignored or changed to the nearest value.

If changing the frame rate then the TV Sync standard may also change. It is important that the TV Sync value must be read and displayed after any changes to the FPS.

## Genlock or Sync Standard

GLSD? Returns the current TV sync (Genlock) method as a string GLSD=n where n is an ASCII 0 through 13 where;

0 = Off

1 = PAL

2 = NTSC

3 = 720p

4 = 720p x2 (double rate)

5 = 1080i

6 = 1080p

7 = 1080p x2

8 = LTC (Timecode)

9 = 44.1 KHz Wordclock

10 = 88.2 KHz Wordclock

11 = 48 KHz Wordclock

12 = 96 KHz Wordclock

13 = 192 KHz Wordclock

# Timecode Systems

GENERATE. SYNC. SHARE

GLSD=n Sets the current TV Sync method to the value n as in the list above. If the sync standard is not supported for the current frame rate the reply may not be the same as the request. The reply must be decoded and displayed as certain TV sync methods cannot be used with certain frame rates.

## Genlock or Sync Level

GLOP? Returns the current TV sync level GLOP=n where n is an ascii character '0' or '1' where;

0 = Normal (75 Ohm)

1 = High (2 x 75 Ohm = 37.5 Ohm for 3D camera rigs)

GLOP=n Sets the TV Sync output as per the table above

## RF Channel Number \*

RFCH? Returns the current RF channel number as "RFCH=n". Channel number range is 1 through to a maximum of 14, but will depend on country/area.

RFCH=n Sets the current RF channel number. The reply must be decoded and displayed as certain channels cannot be used within some areas.

## Country/Area Setting \*

RFCN? Return Current Country/Area setting for the ISM band transceiver

RFCN=n Sets the current Country/Area setting. If changing the country then the RF channel number may also change, so the RF channel number must be read and displayed after any changes to the country/area.

0 = CEPT (EU/UK) 865.050-868.050MHz

1 = FCC (US/AU) 915.050-918.650MHz

2 = ARIB (JP) 920.600-923.600MHz

## RF Signal Strength

RFSI? Returns the current signal strength (0 to 99) and whether the unit is locked to an RF timecode signal.

The return will be something like;

#RFSI=68,1

Means the current signal strength is 68 and the unit is locked.

Whereas;

#RFSI=14,0

Means the current signal strength is 14 and the unit is unlocked.

## RF Userbits

RFUB? Returns the 'RF Receive User-Bits' on/off switch value, 0(zero) if off or not sync'd to master, 1 if on i.e. synchronised.

RFUB=n Sets the RF User-Bits on/off switch, where n = 0(off) or 1(on)

## Transmitter On/Off \*

RFTX? Returns the status of the Transmitter, 0(zero) if off, 1 if on.

RFTX=n Sets the Transmitter on/off, where n = 0(off) or 1(on)

# Timecode Systems

GENERATE. SYNC. SHARE

## Note:

The transmitter will be automatically turned off if the Timecode Source is switched to External RF mode. It will not be automatically re-enabled when switched back.

## Device Type Name

STTN? Returns the current unit's device type as a text string such as "SystemOne OEM".

## Device Definable Name

STNM? Returns the current unit's text name string. This string is user definable and is used to represent a 'friendly' name for the unit i.e. "Camera 1".

STNM= Sets the text name of the unit. The maximum length is 11 characters.

## Display Brightness (only useful if used with display)

STBT? (Deprecated – Use STBR) Returns the current brightness setting as "STBT=n" where n is an ascii character representing the current display brightness value '0' to '7'

STBT=n Sets the display brightness.

## Revision

STVS? Returns the current version as a string "STVS=<string>" where <string> lists the program type/revision and the FPGA type/revision separated by a semi-colon ';' followed by the S1C Protocol revision.

STVP? Returns the Program revision \* 100

STVF? Returns the FPGA revision \* 100

STVC? Returns the S1C Protocol revision

## Device Status

STST? (Deprecated – use STSS/STSP)

Returns the current state of the device.

The statuses will be returned as an ASCII string with each status vertical bar ('|', 0x7C) delimited on each line. The status' to be returned will be:-

1. UserBits "UUUUUUUU"
2. Sync Mode (0=INT, 1=Ext RF.. see TCRC command)
3. Ext Sync Std (0=Off, 1=PAL... see GLSD command)
4. FPS (0=25,1=23.98,2=24,3=29.97, 4=30)
5. Country (see RFCN command)
6. RF Channel (see RFCH command)
7. Brightness level (see STBT command)
8. Battery (1-5)
9. On PSU (0 or 1)
10. Device Name (if any)
11. SSID (string) (if any)

A typical return from this request will be:-

```
STST=12ABCD78|0|0|23,030|1|14|60|3|0|D Name|Wave077<LF>
```

# Timecode Systems

GENERATE. SYNC. SHARE

## Individual Device – Static Data

STSS? (For use with individual device – Use BLSS=n if through BLink master)

The current status will be returned as an ASCII string with each status comma delimited on each line. The status' to be returned will be:-

1. Unit Type
2. Unit Device Type Name
3. Firmware Revision \* 100
4. FPGA Revision \* 100
5. Ext Revision \* 100
6. S1C Revision
7. Device Capability Flags – See Appendix

The device capability flags state what functions can be used with the device, such as if a display is fitted, has Wifi capability, or external Sync port etc. This is denoted using a 32bit hexadecimal number.

A typical return from this request will be:-

```
STSS=11,UltraSyncBLU,201,106,0,5,E0F07040<LF>
```

## Individual Device – Poll Changing Data

STSP? (For use with individual device – Use BLSP=n if through BLink master)

Poll for changing status information from the current device.

The status' will be returned as an ASCII string with each status comma delimited on each line. The status' to be returned will be:-

1. Current Assigned Network Group Number – '0'/'A' – 'F'
2. Current UserBits "UUUUUUUU"
3. Data changed flags (8bit hex) - see below
4. Ext Sync Std (0=Off, 1=PAL... see GLSD command)
5. Battery Percentage
6. On PSU (0 or 1)
7. RF Signal Strength (0 to 99)
8. Friendly name

Data changed flags – Bit definitions;

0. Main Static data Changed – use BLSS? to get changes
1. Main Polloing data Changed – use BLSP? to get changes
2. Attached device static data changed – use GCSS?
3. Attached device polling data changed – use GCSP?
- 4-7 Undefined

A typical return from this request will be:-

```
BLSP=1,B,65,CAFEF00D,05,40,0,65,UltraSync 1<LF>
```

STSP=0 Clear Data Changed flags

## Bluetooth Low Energy (BLE) Commands

BTPR= Set BLE into pairing mode

Return values;

BTPR=0 Stop any pairing sequence

BTPR=1 Initiate pairing sequence – look for devices wishing to pair with us

BTPR=2 Connect to device

BTPR? Obtain current pairing information

Return Values

BTPR=0 Pairing sequence is idle

BTPR=1 Searching

BTPR=2,<Device> <Device> is waiting to pair

BTST? Obtain current status of BLE devices

This request returns the number of available connection slots followed by a list of currently connected device names and signal strengths.

i.e.

BTST=1,name1,85,name2,80,,0,,0

Signifying currently two devices connected and just one slot available for a new pairing to another currently unknown device.

BTST= Delete pairing information

BTST=0 Delete whole pairing table

# Timecode Systems

GENERATE. SYNC. SHARE

## Device Control and Status – Generic Commands

### GCMD?

This command is a method of getting connected device. As well as returning the device 'Family' type it also returns the model number.

As BLST only returns the 'Family Type' this function returns both this and the model connected as a string.

The query returns a CSV string where the first value is the Family Type of the connected device followed by a string denoting the devices model number. If only one value is returned, i.e. no comma and no second value, then a 'family' connecting lead is plugged in but no device was detected connected to the lead.

### Device Types

- 1 = Nothing connected
- 0 = Sound Devices
- 1 = Canon OK
- 2 = GoPro

### GCCT? Request the current transport control status

Returned values:-

- 1 = Nothing connected/Offline
- 0 = Unknown
- 1 = Stopped
- 2 = Recording
- 3 = Playback
- 4 = Pause
- 5 = Rewind
- 6 = Fast Forward

### GCCT=n

As above but command to change the state of the device to 'n' status as per list above.

### GCVR?

Requests the firmware version for the attached device and returned as an ASCII string

GCBT? Returns the battery level in percentage units 0 through 100

GCON? Power status 1=On, 0=Off

GCDV? Media Status. GCDV=<drive\_id>,<value>,<units>

<drive\_id>

- 0 = CF Card
- 1 = SD Card
- 2 = Internal Hard Disk
- 3 = External drive

<units>

- 0 = minutes
- 1 = percentage

# Timecode Systems

GENERATE. SYNC. SHARE

2 = Mbytes

## Attached Device Status and Control

These new commands form a combination of the old individual commands but split into just two functions, one to get static data that rarely changes, and one to get data that can change or be set.

**GCXS?** Get attached device's static data.

Returns a series of fields that define the attached device and its capabilities. The fields are returned comma delimited.

1. Manufacturer's name - ASCII text value
2. Manufacturer's model - ASCII text value
3. Capability Flags – 32bit hexadecimal number – See appendix
4. Storage unit #1 Name (ASCII string)
5. Storage unit #1 Capacity (integer value)
6. Storage unit #1 Units 0 = Empty/Unknown, 1 = MB, 2 = GB, 3=TB
7. Storage unit #2 Name
8. Storage unit #2 Capacity
9. Storage unit #2 Units
10. Storage unit #3 ... etc.

**GCXP?** Get the attached device's changing data

Returns a series of fields denoting the attached devices current status

The fields are comma delimited.

1. Current transport status – See appendix
2. Current frame rate numerator
3. Current frame rate denominator
4. Rate Flags (8bit Hex) – Bit 0 = Drop Frame, others unused
5. Current battery level percentage
6. External power applied/battery charging flag
7. Current clip/file name
8. Storage unit currently selected or in use
9. Storage unit #1 media remaining time in minutes
10. Storage unit #1 media remaining size
11. Storage unit #2 media remaining time in minutes
12. Storage unit #2 media remaining size
13. Storage unit #3... etc.

**GCXP=** Used to set the attached device's transport status or frame rate.

This command can only change the transport control state or the frame rate. The command need only supply the first field if just controlling the transport state or optionally two fields if also controlling the frame rate.

1. Current transport status – See appendix
2. Current frame/bit rate Numerator (optional)
3. Current frame/bit rate Denominator (optional)

# Timecode Systems

GENERATE. SYNC. SHARE

## BLink networks

BLink is a method of connecting client/slave devices to a master controlling device via the TCS proprietary long-range RF link. The master device **MUST** be the device that the clients are connected to for RF Timecode synchronisation.

All S1C commands are initially sent to the master device and then relayed to the BLink client via the RF link. Responses from the BLink client are then replied back through the RF link to the master and then back to the command originator.

Generally, all commands/requests are replied to but it must be noted some replies may take over a second to be returned. If no reply is heard within, say 2 seconds, the original request should be re-sent.

The master device continually polls the BLink network both obtaining basic status information which is stored in a local table for fast responses to requests

The BLink master also looks for new devices being connected to the network. New devices are auto allocated a new BLink ID which corresponds to a table entry of status information. New clients will generally be allocated an empty table position but if these get used up then entries from clients 'not heard from' will be re-used. This table may change as BLink devices come and go from the network.

- New devices may re-use table entries of devices that have disappeared.
- Table entries may exist of devices that have not been heard from for a long time.
- Returning devices may be allocated a new BLink network ID and therefore a different position in the table, therefore reference to Unique ID must be maintained.

The BLink system can be configured for in excess of 1000 clients but currently restricted to around 50 clients depending upon the TCS device used as BLink master. It must be considered that more BLink clients do reduce the speed of the network.

### Command use with BLink network

Example addressing of individual unit within BLink network

"#@1;GLSD?\n" Requests Sync standard of Blink device 1

"#@42;GLSD=0\n" Sets Sync standard of Blink device 42 to

Off.

Note; "#@0;GLSD?\n" Requests the Sync standard of the Blink Master and will result in the same as "#GLSD?\n"

Broadcast commands to all units will be for a restricted set of commands/settings only (not requests) and will be addresses using "#@;" prefix i.e. device number missing. Broadcast messages are NOT responded to by the clients and there is no way of knowing without checking that a command has been heard or actioned.

There are also Group addressing commands that can address groups of units that are assigned to a particular group. There are 6 groups named 'A' through to 'F' and should be addressed as #@A; for instance, to address group 'A' devices.

# Timecode Systems

GENERATE. SYNC. SHARE

S1C is designed to address either the 'Master' device or any of the devices listening to the 'Master' device. The master device will be the device that all S1C communications are directed for access to the BLink network.

S1C may also be used directly with devices that are currently a client within another master device's BLink network, in this instance access to the BLink network is not possible. This would be used locally to ask for current information of that particular device, such as Timecode etc.

BLID? Retrieves the current BLink ID for the device. The BLink ID is determined by the BLink master and cannot be set otherwise.

BLGR? Returns the BLink group the slave belongs. Slave devices can be addressed individually or within groups. There are six groups so the reply will be an 'A' through 'F' dependant on current group or will return '0' (zero) if slave does not belong to any group.

BLGR= Sets the slave unit to belong to a particular group of slaves. Values should be an ASCII 'A' through to 'F' or a '0' if to remove the slave from all groups.

## Sending user defined data to BLink slave devices

DASP=<string>

Send a user-defined <string> to the serial port on the slave devices.

The command should be used along with the formatting required to address slave units individually or globally. i.e.

"#@7;DASP=Hello\n" Will send "DASP=Hello" to slave ID #7

"#@;DASP="To All\n" Will send "DASP=To All" to every slave

This command will also work with binary data such as

"#@3;DASP>Length+Data\n" Will send Length+Data to slave ID #3

The overall length of the command should be kept as short as reasonably possible and must not exceed 112 bytes.

## Sending user defined data from slave devices back to the master.

DAMP=<string>

Send a user-defined <string> to the serial port on the master device.

i.e. "DAMP=Hello\n"

Will arrive at the master as "#@n;DAMP=Hello" - where n = slave ID

**All the following commands should be directed to the master device only**

BLST? (Deprecated – use BLSS/BLSP)

Returns the number of known BLink devices on the network and a number or sequence of numbers that represent a bitmap of which slave devices are active. A reply of BLST=5, 91 means there are currently 5 devices online, and if the 91 (decimal) if converted to a binary value i.e. 1011011 will give a bitmap representation of active devices where bit 0 represents device 1, bit 1 represents device 2, and so on. In this instance devices 1, 2, 4, 5, & 7 are all active. Each bitmap number represents 8 devices so a third number in the reply represents the bitmap of devices 9 through to 16 and so on. This request should then be followed with individual requests for a particular device's current status.

BLST Command must be used direct to the connected master device.

Addressing a slave device such as #@3:BLST? will return nothing.

BLST=n Retrieve basic status information from a BLink network device. Where n is the index into the table of stored devices.

For instance; BLST=1 will reply with the details of the first device in the table, and BLST=2 will reply with the details of the second device in the table, and so on. The statuses will be returned as an ASCII string with each status comma delimited on each line. The status' to be returned will be:-

1. ID (Table Index)
2. Unique ID(hex form of 32 bit number i.e. 8 characters)
3. Unit Type
4. Last Reply (how long since last reply from client [x10mS])
5. Replied Timecode "HHMMSSFF"
6. Replied UserBits "UUUUUUUU"
7. Sync Mode (0=INT, 1=Ext RF.. see TCSC command)
8. FPS (0=25,1=23.98,2=24,,,etc)
9. Ext Sync Std (0=Off, 1=PAL... see GLSD command)
10. Locked to Timecode Source (0 or 1)
11. Battery (1-5)
12. On PSU (0 or 1)
13. UserBits Lock (1=User Settable 0=Locked to RF Master)
14. Attached Device Type – See GCMD command
15. Current Assigned Group Number – 0 if none
16. Attached Device Flags. Value should be broken into binary bits

A typical return from this request will be:-

BLST=1,12ABCD78,6,121,00043410,00000000,1,1,0,1,3,0,0,2,0,3<LF>

We may wish to expand this list in the future so currently ignore any subsequent statuses.

# Timecode Systems

GENERATE. SYNC. SHARE

The statuses are only updated around twice per second maximum. So this command should only be used infrequently.

## Initial Connect – Static Data

**BLSS?** Returns a bitmap of known BLink devices on the network.

The bitmap is broken down into 32-bit hexadecimal values that represent a bitmap of which slave devices are active. If say there is the reply BLST=0000005B if now the reply is broken into binary, each bit represents a device present or not (1 or 0). Thus, the hex value of 0000005B breaks down to binary 0...0001011011 and shows there are 5 bits set, thus there are currently 5 devices online, also the bits that are set gives a representation of active devices where bit 0 represents device 1, bit 1 represents device 2, and so on. In this instance devices 1, 2, 4, 5, & 7 are all active. Each bitmap number represents 32 devices so additional values in the reply represents the bitmap of devices 33 through to 64 and so on.

The BLSS Command must be used direct to the connected master device.

Addressing a slave device such as #@3:BLSS? will return nothing.

**BLSS=n** Returns the slave device's static data, where n is the index into the table of stored devices.

The statuses will be returned as an ASCII string with each status comma delimited on each line. The status' to be returned will be:-

1. Blink ID
2. Unique ID (32bit Hexadecimal)
3. Unit Type
4. Unit Device Type Name
5. Firmware Revision \* 100
6. FPGA Revision \* 100
7. Ext Revision \* 100
8. S1C Revision
9. Device Capability Flags (See appendix)

The device capability flags will let the Hub know what functions can be used with the device, such as if a display is fitted, has Wifi capability, or external Sync port etc. It is proposed to use a 32bit hexadecimal number.

A typical return from this request will be:-

```
BLSS=1,12ABCD78,11,UltraSyncBLU,201,106,0,5, E0F07040<LF>
```

There could be confusion when parsing the returned values from the above two requests as they both return "BLSS=<reply>". To differentiate between the two the length of the first <reply> field will determine which request the reply was from. If the first field has a length of 8 characters it can be determined that the reply is for the "BLSS?" query, otherwise it is a reply to the "BLSS=n" query.

# Timecode Systems

GENERATE. SYNC. SHARE

## Slave Poll

**BLSP?** Same as 'BLSS?' command – see above.

**BLSP=n** Retrieve status information from a BLink network device.

Where n denotes the index into the table of stored devices.

For instance; BLSP=1 will reply with the details of the first device in the table, and BLSP=2 will reply with the details of the second device in the table, and so on.

The statuses will be returned as an ASCII string with each status comma delimited on each line. The status' to be returned will be:-

1. Blink ID
2. Current Assigned Group Number – '0'-'A' – 'F'
3. Last Reply (time since last reply x10mS). Zero if unheard.
4. Current UserBits "UUUUUUUU"
5. Data changed flags (8bit hex) - see below
6. Ext Sync Std (0=Off, 1=PAL... see GLSD command)
7. Battery Percentage
8. On PSU (0 or 1)
9. RF Signal Strength (0 to 99)
10. Friendly name

Field 5 – Data changed flags – Bit definitions;

- 0 Main Static data Changed – use BLSS? to get changes
- 1 Main Polling data Changed – use BLSP? to get changes
- 2 Attached device static data changed – use BLXS?
- 3 Attached device polling data changed – use BLXP?
- 4-7 Undefined

The changed flags are automatically reset after this query.

A typical return from this request will be:-

BLSP=1,B,65,CAFEF00D,05,40,0,65,UltraSync 1<LF>

## Devices with changed information

**BLCH?** Similar to 'BLCG' command but only flags changes once the master device has updated its records with the changes.

The map returned is the same as the 'BLSS?' command above i.e. hexadecimal not decimal.

**BLCH=0** Reset BLCH bits.

**BLCG?** (Deprecated – use BLCH)

Returns the number of BLink network slave devices that have changed details plus a bitmap of the affected devices.

**BLCG=0** Reset BLCG bits.

# Timecode Systems

GENERATE. SYNC. SHARE

## Slave's Attached Device Status

These commands should be used to get information regarding a device attached to a slave unit, directly from the master device. There is one query to get static data that rarely changes, and one to get data that can frequently change.

**BLXS=n** Get the slave device's static data where n = slave's BLink ID.  
Returns a series of CSV fields that define the attached device and its capabilities.  
The fields are returned comma delimited.

1. Blink ID
2. Manufacturer's name - ASCII text value
3. Manufacturer's model - ASCII text value
4. Capability Flags – 32bit hexadecimal number – See appendix
5. Storage unit #1 Name (ASCII string)
6. Storage unit #1 Capacity (integer value 0 - 65535)
7. Storage unit #1 Units 0 = Empty/Unknown, 1 = KB, 2 = MB, 3 = GB
8. Storage unit #2 Name
9. Storage unit #2 Capacity
10. Storage unit #2 Units
11. Storage unit #3 ... etc.

**BLXP=n** Get the slave device's changing data where n = slave's BLink ID.  
Returns a series of CSV fields denoting the attached devices current status  
The fields are comma delimited.

1. Blink ID
2. Current transport status – See appendix
3. Current frame rate numerator
4. Current frame rate denominator
5. Rate Flags (8bit Hex) – Bit 0 = Drop Frame, others unused
6. Current battery level percentage
7. External power applied/battery charging flag
8. Current clip/file name
9. Storage unit currently selected or in use
10. Storage unit #1 media remaining time in minutes
11. Storage unit #1 media remaining size
12. Storage unit #2 media remaining time in minutes
13. Storage unit #2 media remaining size
14. Storage unit #3... etc.

# Timecode Systems

GENERATE. SYNC. SHARE

## Updating Firmware

UFCK>length(=0x0A)+First ten bytes of update file+'\\n'

Binary command with an ASCII response to check the version of an update file. The update type and version are held within the first ten bytes of the update file so these need to be included in the query.

The response will be UFCK=type, version, flag

type – 0=unknown, 1=main FW, 2=FPGA, 3=Ext

version = update version \* 100

flag – 0=Can't use, 1=Ok to proceed, 2=same version, 3=older version than current

UFST=n

Initiate a firmware update where n = overall size of 'update file' to be transferred.

The response will give the maximum size of update data that can be transferred in each 'UFDA' block. i.e. UFST=106. *Currently this is limited to a 106 byte maximum but, in the future and if transferred locally using say the serial interface, this may be increased.*

This command may take up to 3 seconds to complete depending on the size of the update.

Please use this command sparingly as each call forces an erase of the internal Flash memory update buffer.

UFDA>length + address + data-block + checksum + '\\n'

This is a binary command where blocks of data are sent sequentially starting at address zero and progressing through until all the file is transferred. The data is sent in binary format, where;

- length is a single byte length of the entire message to follow (not including the final '\\n'). Therefore, it is the length of the data block plus five (address and checksum bytes). *Currently the length value should not exceed 111(0x6f). With the final '\\n' character this is the current maximum total of 112bytes of data following the length byte.*
- address is a three-byte, little endian, relative address of the data block from start of file.
- data-block is the block of data from the file
- checksum is a 2's compliment 16bit modular checksum of each of the address and data-block bytes. This is computed by successive addition of each of the (8-bit) bytes starting with the address bytes and ending with the final data byte and discarding any overflow beyond 16 bits. The resultant 16bit word is then two's complimented and appended, little endian, to the end as the last two bytes of the message.

The responses are as follows

UFDA=0 All fine – ready for next block

UFDA=1 All fine – Transfer complete

UFDA=2 Checksum error - Please Repeat message

UFDA=3 Address out of sequence

UFDA=4 Length error – or UFST has not been issued

# Timecode Systems

GENERATE. SYNC. SHARE

UFGO=checksum

This command initiates the update process once all the data has been transferred where checksum is a 2's compliment 16 bit checksum of the entire file and presented in decimal. See UFDA command for method of calculation.

Responses are as follows

UFGO=0 Only for main firmware update. Flag to specify that all was fine and now proceeding with update. The communications will be lost for a short time and when complete a UFGO=1 will be sent;

UFGO=1 Update has been completed. There may be many seconds delay before this is issued.

UFGO=2 Data is not yet complete

UFGO=3 Checksum does match stored data

UFGO=4 Old or same update version

UFGO=-n Negative return values during update represent update failure.

Each type of update should be done separately i.e. Main Code, FPGA, and BLE updates, we always need to update main code first. We will be providing updates as three separate files where one or more may be needed to be updated at any one time. The AirGlu can determine what each update is for from the contents, you do not have to retain the filenames.

## Appendix

### TCS Device Capabilities Flags as used in BLSS=n and STSS? requests

Bit	Capability
0	Has LTC Output function
1	Has LTC Input function
2	Has Genlock Output function
3	Has Word-Clock Output function
4	Has Wi-Fi capability
5	Has BLE capability
6	Has Serial port
7-11	Currently unallocated with zero default
12	Has a built-in display and can control brightness
13	Has button control
14	Has internal rechargeable battery
15-19	Currently unallocated with 0 default
20-23	Currently unallocated with 1 default
24	Can do zero brightness display setting
25	Can do zero brightness LED flash
26	Can do BLink master
27-28	Currently unallocated with 0 default
29-31	Currently unallocated with 1 default

### External Device Capability Flags as used in BLXS=n and GCXS? Requests

Bit	Capability
0/1	Number of fitted Media Storage units
2	Transport state can be controlled
3	Can be controlled On or Off
4	Predominantly Video device using FPS (=0) or Audio Bitrate device (=1)
5-9	Currently unallocated default =0
10-15	Unallocated default =1
16-23	Unallocated default =0
24-31	Unallocated default =1

## Integration Instructions

### 2.2 List of applicable FCC / ISED rules

FCC:	ISED:
47CFR 15.247	RSS-247

### 2.3 Specific operational use conditions

Not applicable

### 2.4 Limited module procedures

Not applicable

### 2.5 Trace antenna designs

Not applicable

### 2.6 RF exposure considerations

To comply with FCC RF Exposure requirements, the OEM must ensure that only antennas are installed which are listed with this module antenna details. The AirGlu2 module should be used in such a manner that the potential for human contact during normal operation is minimized. OEM integrators and end users must be provided with transmitter operating conditions for satisfying RF exposure compliance.

Radio frequency radiation exposure Information: this equipment complies with FCC radiation exposure limits prescribed for an uncontrolled environment for fixed use conditions. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter except in accordance with FCC procedures and as authorized in the module certification filing. Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

### 2.7 Antennas

The external antenna approved for use is the Taoglas TG.09.0113 SMA (F) monopole. The maximum Gain (free space) allowed is 2.0 dBi (900MHz) and -6dBi (2.4GHz). No other antenna type or higher gain of antenna design should be used. The unique antenna connection used on the AirGlu2 module is the ECT (Electric Connector Technology Co Ltd) part no. 818000157.

### 2.8 Label and compliance information

The host product manufacturer is responsible to provide physical or e-labelling stating 'Contains FCC ID: AYV-AGLU02 or Contains IC: 10427A-AGLU02' with the finished product. See Guidelines for Labeling and User Information for RF Devices – KDB Publication 784748.

### 2.9 Information on test modes and additional testing requirements

The AirGlu2 module comes complete with a Test Mode API within the core operation firmware, to configure all required test modes as a standalone transmitter in a host

### 2.10 Additional testing, Part 15 Subpart B disclaimer

The AirGlu2 modular transmitter is only FCC authorized for the specific rule parts (i.e., FCC transmitter rules) listed on the grant, and that the host product manufacturer is responsible for compliance to any other FCC rules that apply to the host not covered by the modular transmitter grant

# Timecode Systems

GENERATE. SYNC. SHARE

of certification. The final host product requires Part 15 Subpart B compliance testing once the modular transmitter installed.

## Certifications

### CE

The AirGlu2™ module is in conformity with the essential requirements and other relevant requirements of the Radio Equipment Directive (RED) (2014/53/EU). Please note that every application using the AirGlu2™ module will need to perform the radio EMC tests on the end product according to EN 301 489-17. The conducted test results can be inherited from the modules test report to the test report of the end product using AirGlu2™ module. EN 300 328 radiated spurious emission test must be repeated with the end product assembly. Test documentation and software for the EN 300 328 radiated spurious emissions testing can be requested from the Timecode Systems support.

# Timecode Systems

GENERATE. SYNC. SHARE

## FCC

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

- This device may not cause harmful interference, and
- This device must accept any interference received, including interference that may cause undesirable operation.

Any changes or modifications not expressly approved by Timecode Systems could void the user's authority to operate the equipment.

### **FCC RF Radiation Exposure Statement:**

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. End users must follow the specific operating instructions for satisfying RF exposure compliance. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter except in accordance with FCC multi-transmitter product procedures. OEM Responsibilities to comply with FCC Regulations: The transmitter module must not be co-located or operating in conjunction with any other antenna or transmitter except in accordance with FCC multi-transmitter product procedures. Each new host will require reassessment of radiated spurious emissions and a permissive change to the certification. For AirGlu2™ module the minimum separation distance to human body is 20cm. OEM integrator is responsible for testing their end-product for any additional compliance requirements required with this module installed (for example, digital device emissions, PC peripheral requirements, etc.). Important Note: In the event that this condition cannot be met (for certain configurations or co-location with another transmitter), then the FCC authorization is no longer considered valid and the FCC ID cannot be used on the final product. In these circumstances, the OEM integrator will be responsible for re-evaluating the end product (including the transmitter) and obtaining a separate FCC authorization.

### **End Product Labeling**

The AirGlu™ module not labelled with its own FCC ID due to its physical size. If the FCC ID is not visible when the module is installed inside another device, then the outside of the device into which the module is installed must also display a label referring to the enclosed module. In that case, the final end product must be labeled in a visible area with the following:

"Contains Transmitter Module FCC ID: AYV-AGLU02"

Or

"Contains FCC ID: AYV-AGLU02"

The OEM integrator must not provide information to the end user regarding how to install or remove this RF module or change RF related parameters in the user manual of the end product.

## ISED Canada

### ISED Canada (English)

This radio transmitter has been approved by Industry Canada to operate with its embedded antenna. Other antenna types are strictly prohibited for use with this device. This device complies with Industry Canada's license-exempt RSS standards. Operation is subject to the following two conditions:

1. This device may not cause interference; and
2. This device must accept any interference, including interference that may cause undesired operation of the device.

#### **RF Exposure Statement**

AirGlu2™ module meets the given requirements when the minimum separation distance to human body is more than 20cm. RF exposure or SAR evaluation is not required when the separation distance is 20cm or more. AirGlu2™ module has been tested for worst case RF exposure.

#### **OEM Responsibilities to comply with IC Regulations**

The transmitter module must not be co-located or operating in conjunction with any other antenna or transmitter. Radiated emission must be tested with each new host product and ISEDC must be notified with a Class 4 Permissive Change. OEM integrator is responsible for testing their end-product for any additional compliance requirements required with this module installed (for example, digital device emissions, PC peripheral requirements, etc.).

#### **Important note**

In the event that these conditions cannot be met (for certain configurations or co-location with another transmitter), then the IC authorization is no longer considered valid and the IC ID cannot be used on the final product. In these circumstances, the OEM integrator will be responsible for re-evaluating the end product (including the transmitter) and obtaining a separate IC authorization.

#### **End Product Labelling**

The AirGlu™ module is not labelled with IC ID because of its small physical size. The final end product must be labelled in a visible area with the following:

“Contains Transmitter Module IC: 10427A-AGLU02”

Or

“Contains IC: 10427A-AGLU02”

The OEM integrator has to be aware not to provide information to the end user regarding how to install or remove this RF module or change RF related parameters in the user manual of the end product.

## ISED Canada (Français)

# Timecode Systems

GENERATE. SYNC. SHARE

Cet émetteur radio (IC : 10427A-AGLU02) a reçu l'approbation d'Industrie Canada pour une exploitation avec l'antenne puce incorporée. Il est strictement interdit d'utiliser d'autres types d'antenne avec cet appareil. Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes: 1. L'appareil ne doit pas produire de brouillage; et 2. L'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible de provoquer un fonctionnement non désiré de l'appareil. Déclaration d'exposition aux RF

Le module AirGlu2™ répond aux exigences données lorsque la distance minimale de séparation avec le corps humain est supérieure à 20 cm. L'exposition RF ou l'évaluation SAR n'est pas nécessaire lorsque la distance de séparation est de 20 cm ou plus. Le module AirGlu2™ a été testé pour le pire cas d'exposition RF. Responsabilités du FEO ayant trait à la conformité avec les règlements IC Le Module AirGlu a été certifié pour une intégration dans des produits uniquement par les intégrateurs FEO dans les conditions suivantes: . • Le module émetteur ne doit pas être installé au même endroit ou fonctionner conjointement avec toute autre antenne ou émetteur. Dès lors que les deux conditions ci-dessus sont respectées, aucun test supplémentaire de l'émetteur n'est obligatoire. Cependant, il incombe toujours à l'intégrateur FEO de tester la conformité de son produit final vis-à-vis de toute exigence supplémentaire requise avec ce module installé (par exemple, émissions de dispositifs numériques, exigences relatives aux matériels périphériques PC, etc). Note: S'il s'avère que ces conditions ne peuvent être respectées (pour certaines configurations ou la colocation avec un autre émetteur), alors l'autorisation IC n'est plus considérée comme valide et l'identifiant IC ne peut plus être employé sur le produit final. Dans ces circonstances, l'intégrateur FEO aura la responsabilité de réévaluer le produit final (y compris l'émetteur) et d'obtenir une autorisation IC distincte. Étiquetage du produit final L'étiquette du Module AirGlu porte son propre identifiant IC. Si l'identifiant IC n'est pas visible quand le module est installé à l'intérieur d'un autre appareil, alors l'extérieur de l'appareil dans lequel le module est installé doit aussi porter une étiquette faisant référence au module qu'il contient. Dans ce cas, une étiquette comportant les informations suivantes doit être apposée sur une partie visible du produit final. "Contient le module émetteur IC: 10427A-AGLU02" ou "Contient IC : 10427A-AGLU02" L'intégrateur FEO doit être conscient de ne pas fournir d'informations à l'utilisateur final permettant d'installer ou de retirer ce module RF ou de changer les paramètres liés aux RF dans le mode d'emploi du produit final.

## JAPAN

The AirGlu2™ module is certified in Japan with certification number 008-220415

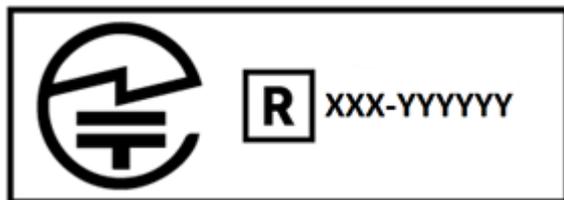
UK Timecode Systems Ltd. Unit 6, Elgar Business Centre, Moseley Road, Hallow, Worcester.  
WR26NJ. UK

# Timecode Systems

GENERATE. SYNC. SHARE

## Important

The module does not have a Japan certification mark and ID because of its small physical size. The manufacturer who integrates a radio module into their host equipment must place the certification mark and certification number on the outside of the host equipment.



The certification mark and certification number must be placed close to the text in the Japanese language which is provided below.

当該機器には電波法に基づく、技術基準適合証明等を受けた特定無線設備を装着している。

## Translation:

“This equipment contains specified radio equipment that has been certified to the Technical Regulation Conformity Certification under the Radio Law.”

