## Richard's stuff

**ELECTRONICS**

# HONEYWELL GALAXY ETHERNET ADAPTER RS485 PROTOCOL

3RD MAY 2020 | RICHARD | 3 COMMENTS

This is a belated follow-up to my previous posts, this time about the Honeywell Galaxy RS485 protocol, covering the Ethernet adapter. The adapter normally provides reports to an Alarm Receiving Centre (ARC). It can also be used to configure the alarm panel through Honeywell software. It does not, as far as I know, provide any direct interface to the alarm to allow it to be set/unset/etc. Devices like the LCE-01 provide this via a virtual keypad interface, by emulating a keypad on the RS485 bus (i.e. it does not interact with the panel through the normal panel<->ethernet interface, but instead through the panel<->keypad interface).

All data values below are in hex. The last byte of each command/response is a checksum (generally replaced below with XX).

### General replies

```
<- 11 EC 03 C0 00 00 00 00 6C (ok)
<- 11 EC 03 E0 00 00 00 00 8C (error, "ARC Comms fail")
```

When an error occurs the Ethernet device will reply with an error response to several (5?) polls in a row, presumably to ensure the message gets through to the panel (as the Ethernet device does not get confirmation from the panel that it has received the error).

### Initial device poll

```
-> 25 00 0E 0F 00 31 35 36 89
<- 11 FF 0A 00 DD A3
```

As previously described the alarm panel will poll for devices on startup and when exiting engineer mode. 25 is the only valid device id for the Ethernet adapter. 31 35 36 is ascii for “156” and my panel is running v1.56 firmware, so this is probably to allow the Ethernet card to ensure the panel has a compatible level of firmware before trying to interact with it. I have only seen one response (as above) so I do not know what a failure response would look like.

### Status poll

```
-> 25 1A 00 E9
```

Regular poll for device events, send a generic ok/error reply (see above).

### Device configuration

```
-> 25 1C 10 31 39 32 2E 31 36 38 2E 37 2E 35 0 0 27 12 XX
-> 25 1C 14 01 03 XX
-> 25 1C 18 01 XX
-> 25 1C 1A 31 39 32 2E 31 36 38 2E 37 2E 32 39 00 XX
-> 25 1C 1B 31 39 32 2E 31 36 38 2E 37 2E 31 00 XX
-> 25 1C 1C 32 35 35 2E 32 35 35 2E 32 35 35 2E 30 00 00 XX
-> 25 1C 1D 00 1E 04 00 00 00 00 XX
-> 25 1C 1E 31 39 32 2E 31 36 38 2E 37 2E 36 0 27 12 XX
-> 25 1C 1F 00 00 XX - 11 EC 03 C0 00 00 00 00 6C
-> 25 1C 20 35 34 2E 32 34 37 2E 31 31 32 2E 38 32 0 27 15 XX
```

Some/all of these commands are sent on device startup, periodically and on change (via the engineer menu).

25 1C 10… Primary reporting ip address and port. Ip address is null terminated ascii string e.g. 31 39 32 2E 31 36 38 2E 37 2E 35 0 = “192.168.7.35”. There is an additional 00 (purpose unknown) and then the port number in hex e.g. 27 12 = 0x2712.

25 1C 1E… Secondary reporting address, format as for primary.

25 1C 14… Reporting format and level, in the following format:

```
25 1C 14 fm lv XX
fm - 01=microtech/02=sia
lv - 00-03 for sia, always 03 for microtech
```

25 1C 18… Comms retries, in the following format:

```
25 1C 18 nn XX
nn - number of retries
```

25 1C 1A… Ethernet device ip address, null terminated string e.g. 31 39 32 2E 31 36 38 2E 37 2E 32 39 00 = “192.168.7.1”

25 1C 1B… Ethernet device gateway, null terminated string e.g. 31 39 32 2E 31 36 38 2E 37 2E 31 00 = “192.168.7.1”

25 1C 1C… Ethernet device subnet, null terminated string e.g. 32 35 35 2E 32 35 35 2E 32 35 35 2E 30 00 = “192.168.7.29”

25 1C 1D… Various settings, in the following format:

```
25 1C 1D 00 hb sp en lf rm ip XX
```

```
hb - heartbeat timer (mins)
sp - line fail signal path 00=primary/01=secondary/02=alarm mon/03=any/04=all
en - encrypt bitmask 01=alarm report | 02=remote access | 04=alarm mon
lf - line fail network 00=off/01=available
rm - receiver mode 00=alt/01=dual
ip - protocol 00=udp/01=tcp
```

25 1C 20… Monitoring ip address and port. Ip address is null terminated ascii string e.g. 35 34 2E 32 34 37 2E 31 31 32 2E 38 32 0 = “54.247.112.82”. (There is **NO** additional 00 in this setting.) Then the port number in hex e.g. 27 15 = 0x2715. Note this setting is only sent when it is changed and before a report is sent, not as part of the block of settings routinely sent on startup.

**Reports**

```
-> 25 1C 50 23 31 30 30 31 30 33 XX                                "#10
-> 25 1C 52 4F 74 69 30 30 3A 34 34 2F 69 64 32 34 2F 52 58 XX     "Oti
-> 25 1C 51 41 43 4F 4D 20 54 45 53 54 20 20 45 6E 67 2E 20 20 XX  "ACC
```

```
-> 25 1C 52 4F 74 69 30 30 3A 30 39 2F 70 69 30 31 2F 54 52 XX  "Oti00:
-> 25 1C 51 41 2D 20 54 41 4D 50 45 52 20 20 4B 50 44 20 31 XX  "A- TAM
```

Reports are sent in 2 or 3 messages, each of which needs a generic ok reply, depending on settings. I can’t remember which format and level options, setting a monitor ip, etc, results in what output – you’ll have to play…, I seem to remember that sometimes the first message above (25 1C 50…) is not sent.

25 1C 50… Account number, ascii (not null-terminated).

25 1C 52… Report code, ascii (not null-terminated). I don’t know much about the content of the messages, but searching the web for *sia alarm codes* will tell you more.

25 1C 51… Human readable report, ascii (not null-terminated).

Scroll right in the code boxes above to see the ascii of these examples. When I used SelfMon to receive these messages the first characters were removed in the output. It’s possible that the “O” & “A” at the start of the messages are actually binary flags that just coincidentally are all reasonable ascii characters on this occasion or, probably more likely, are ascii flags. If you are implementing a network adapter you would need to send the messages on to the ARC at this point. This is likely to need to be done asynchronously and any errors reported to the panel on a subsequent status poll.

ALARM GALAXY HONEYWELL RS485 S2

## 3 THOUGHTS ON “HONEYWELL GALAXY ETHERNET ADAPTER RS485 PROTOCOL”

**stuartmcg**

2ND MARCH 2022 AT 7:56 AM

“Devices like the LCE-01 provide this via a virtual keypad interface, by emulating a keypad on the RS485 bus (i.e. it does not interact with the panel through the normal panelethernet interface, but instead through the panelkeypad interface).”

This isn’t correct. The LCE device, like the E080 device communicates Microtech protocol through the Ethernet device. The emulated keypad only performs keypad operations. The newer LCE-K3 device also supports encrypted Ethernet comm’s. The easy part is emulating the keypad, the Ethernet device, which supports full panel control and status, is much more difficult to implement.

---

**stuartmcg**

13TH JUNE 2020 AT 8:18 AM

From an old header… Can’t believe I wrote this almost ten years ago. The address and command bytes change depending on panel variant – and firmware version (hence the firmware versions being passed around at the start), so what works on the G2 will not always be the same on classic / G3 / Flex / Dimension. The O and A are message blocks, with O being the data block and A being the ASCII block. With Microtech protocol, you receive O or N data blocks, with N for prioritised / actionable events. On the RS485 bus, the Galaxy master passes these out in three chunks so, depending on the IP protocol, they need stitched together and encapsulated for passing to an ARC. See the latest Flex+ manual page 213, as it has a more up to date description of the message block.

```
// 25 1C XX (where XX is byte 2 below)

#define ETH_MOD_REQUEST 0x0F // Module request expects reply with version info.
#define ETH_SEC_REPORT 0x1E // IP:port
#define ETH_REPORT_FORMAT 0x14 // Microtech 0x02 SIA 0x01 – We’ll ignore and keep m-tec
#define ETH_REPORT_FAIL 0x18 // Time before reporting comms fail 0 to 20 minutes hex 0 to 0x14
#define ETH_IP 0x1A // IP address only
#define ETH_GATEWAY 0x1B // IP address only
#define ETH_MASK 0x1C // Mask only
#define ETH_RECEIVER 0x1D // Config – See documentation.
#define ETH_PRI_REPORT 0x10 // IP:port
#define ETH_SIA_CONTROL 0x1F // IP address only
#define ETH_ALARM_MON 0x20 // IP:port

#define ETH_REPORT_1 0x50 // The site ID. #111111
#define ETH_REPORT_1_G3_P1 0x23 // When primary alarm report is set on G3 the event comes from this addr
#define ETH_REPORT_2 0x52 // The ‘O/N’ String ‘/’ separated. 0ti02:02/id23/CL
#define ETH_REPORT_2_G3 0x53
```

#define ETH_REPORT_2_G3_P1T 0x4E // When primary alarm report is set on G3 the event comes from this addr for tag related events.
#define ETH_REPORT_2_G3_P1 0x4F // When primary alarm report is set on G3 the event comes from this addr
#define ETH_REPORT_3 0x51 // The 'A' additional info string. AFULL SET MASTER
#define ETH_REPORT_3_G3_P1 0x41 // When primary alarm report is set on G3 the event comes from this addr

**ferz**

22ND APRIL 2021 AT 1:32 PM

Hi Thank you all for sharing this information, it has been very useful to connect my home alarm to the internet.

Do you happen to know if there's a command that can be sent to request a status update? Currently with the ethernet report messages it is difficult to identify the current state of the alarm.
It would be great if the ethernet module supported such command, or if this could be obtained via another module protocol (e.g. RIO).

Thanks again