

api key
KhrqluzkekBrxbu8qTgzwQLu13VQgOxYOLKN0dea

From <<https://data.nsldata.com/console/apidocumentation.php#sendUplinkToRadioView>>

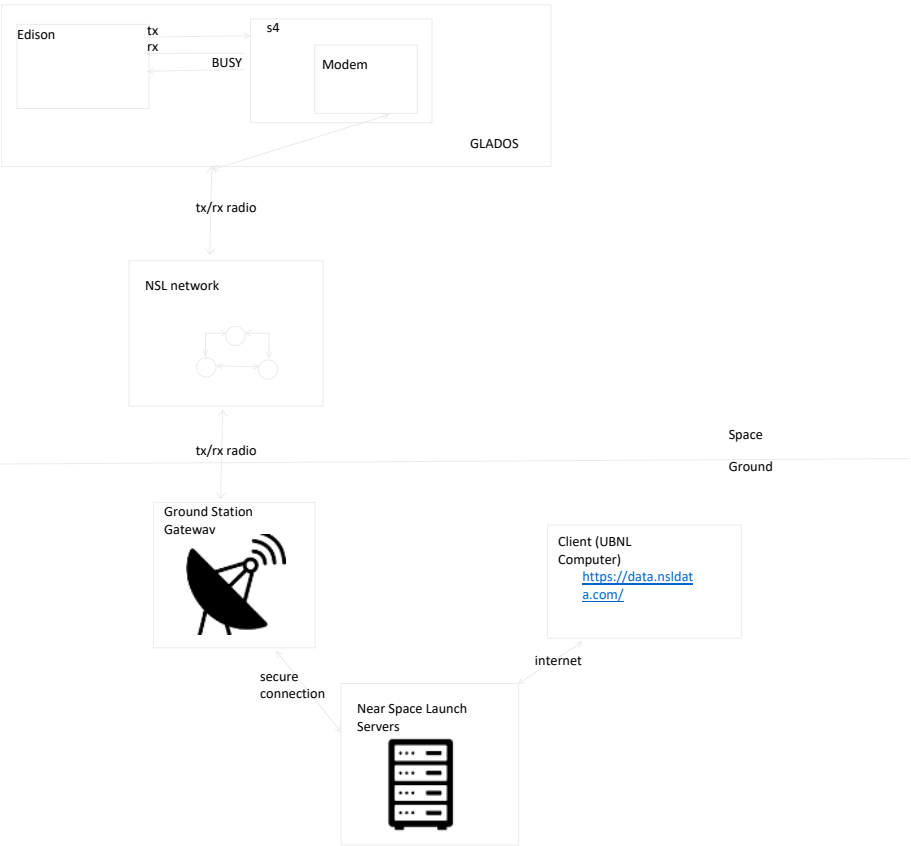
Edison to s4 commands

Payload to S4	Message Format			BUSY Time	Notes
	Header	Function Byte	Data		
Send Payload Data to Ground	50 50 50	F5	Length + 200 payload/ filler bytes	< 30 s	S4 transmits, waits for ACK, timeout after 30 s
Change Configuration	50 50 50	F4	201 data + bytes	< 1 s	Reconfigures S4 parameters. Resets on power-off.
Check Buffer for Uplink Data	50 50 50	48	201 filler bytes	< 1 s	Returns buffer data, then clears buffer
Check Last Serial Packet Status	50 50 50	49	201 filler bytes	< 1 s	Returns last serial packet status and latency
Request UTC Time	50 50 50	4A	201 filler bytes	< 1 s	Returns current UTC time from modem (if available)
Request Health and Safety	50 50 50	F6	201 filler bytes	< 1 s	
Request GPS Packet	50 50 50	F7	201 filler bytes	< 1 s	Optional
Request Mag Packet	50 50 50	0A	201 filler bytes	~ 15 s	Optional
Request PIN Packet	50 50 50	AC	201 filler bytes	~ 16 s	Optional
Request Plasma Packet	50 50 50	AB	201 filler bytes	~ 35 s	Optional

s4 to Edison

S4 to Payload	Message Format			Data	Notes
	Header	Function Byte 1	Function Byte 2		
Mirrored data transmitted OTA	50 50 50	FD	Console Fn Byte	37 – 201 data bytes	Sent whenever S4 transmits OTA
S4 ID and Set Up Info	NA	NA	NA	~152 bytes	ASCII ID Packet sent at power up
Data received from buffer	50 50 50	48	Sequence Count	17 payload bytes	Won't change if nothing received from queue
UTC Time	50 50 50	4A	Sequence Count	17 time bytes	Current UTC Time from modem in ASCII
Serial Health and Safety Packet	50 50 50	F6	Sequence Count	19 data bytes	Sent after collecting data
Serial GPS Packet	50 50 50	F7	Sequence Count	56 data bytes	Optional. Sends current GPS solution.
Serial Mag Packet	50 50 50	0A	Sequence Count	19 data bytes	Optional. Sent after collecting data
Serial PIN Packet	50 50 50	AC	Sequence Count	17 data bytes	Optional. Sent after collecting data
Serial Plasma Packet	50 50 50	AB	Sequence Count	17 data bytes	Optional. Sent after collecting data
ACK - Good Command	AA 05	00	NA	NA	Sent after good command
NAK - Invalid Command	AA 05	FF	NA	NA	Sent after invalid command
PASS - Successful Transmission	F5	Latency	NA	NA	Sent after successful data transfer to Iridium
FAIL - Failed Transmission	F5	FF	NA	NA	Sent after failed data transfer to Iridium

Command Data Handling Architecture



- Implementation Notes
- Make sure to remove padding in c structs
 - change to chars to uint_8
 - omitting gps, mag, pin, and plasma implementations

C API

CONSTANTS

SIMULATION 0

enums

```
EyeStarS4_COM_RESP
{
    ACK
    NAK
    PASS
    FAIL
}
EyeStarS4_COM_TYPE
{
}
EyeStarS4_REQ_TYPE
{
    UTC
    H&S
    LAST_STATUS
}
```

structs

```
EyestarS4_handle
{
    union EyestarS4_port port
}
EyestarS4_header
{
    char header[3]
```

Datasheet notes

- Data can be sent to iridium network via webapi

- uplink data can only be received after a connection is initiated and network acknowledgement as been received
 - this only happens by transmitting a downlinked packet (H&S or payload serial)
- s4 sends ASCII ID packet at power up (simple string of chars ~152 bytes)

Receiving Uplinks

- Only one uplinked packet (to iridium) can be received (by the s4) at a time
 - After (the s4) recieveing an uplinked packet from the (iridium) netowrk, the s4 (modem) will place packet in the buffer to be requested by the payload
 - Any requests made for data in the buffer (by the payload) will receive the last packet to be uplinked
 - previous packets are not stored
 - Must request packets frequently (prolly faster than H&s packet)
 - S4 Must request the buffer data before sending downlink data
 - A sequence counter byted is included in the requested packet which increments in each new pakcet

```

        union EyeStarS4_header_etc

    }

EyeStarS4_req_packet
{

    char header[3]

    char InstrumentId[1]

    char filler[201]

}

EyeStarS4_send_downlink_packet
{
    EyeStarS4_header header

    char length[1]

    char payload[ (data + filler) = 200 ]

}

EyeStarS4_send_change_config
{
    EyeStarS4_header header

    char h&s_period[1]

    char gps_period[1]

    char filler[205-6]

}

EyeStarS4_rcv_utc_packet
{

    EyeStarS4_header header

    char data[17]  //may be able to be read as pure text in form

}

EyeStarS4_rcv_uplink_packet
{

    EyeStarS4_header header
    char data[17]

}

EyeStarS4_rcv_powerup_packet
{

    char data[17]

}

EyeStarS4_rcv_h&s
{
    EyeStarS4_header header

    char sequence[1]

    char utc_seconds[1]

    char temp [1]

    char bus+_volt[2]

    char analog_input1[2]

    char analog_input2[2]

    char analog_input3[2]

    char analog_input4[2]

    char analog_input5[2]

    char analog_input6[2]

    char digital_inputs[1]

    char uplink_count[1]

    char h&s_downlink_count[1]

}

```

Payload to S4	Message Format				
Description	Header	Function Byte	Data	BUSY Time	Notes
Send Payload Data to Ground	50 50 50	F5	Length + 200 payload/ filler bytes	< 30 s	S4 transmits, waits for ACK, timeout after 30 s
Change Configuration	50 50 50	F4	201 data + bytes	< 1 s	Reconfigures S4 parameters. Resets on power-off.
Check Buffer for Uplink Data	50 50 50	48	201 filler bytes	< 1 s	Returns buffer data, then clears buffer
Check Last Serial Packet Status	50 50 50	49	201 filler bytes	< 1 s	Returns last serial packet status and latency
Request UTC Time	50 50 50	4A	201 filler bytes	< 1 s	Returns current UTC time from modem (if available)
Request Health and Safety	50 50 50	F6	201 filler bytes	< 1 s	
Request GPS Packet	50 50 50	F7	201 filler bytes	< 1 s	Optional
Request Mag Packet	50 50 50	0A	201 filler bytes	~ 15 s	Optional
Request PIN Packet	50 50 50	AC	201 filler bytes	~ 16 s	Optional
Request Plasma Packet	50 50 50	AB	201 filler bytes	~ 35 s	Optional

S4 to Payload	Message Format				
Description	Header	Function Byte 1	Function Byte 2	Data	Notes
Mirrored data transmitted OTA	50 50 50	FD	Console Fn Byte	17 – 201 data bytes	Sent whenever S4 transmits OTA
S4 ID and Set Up Info	NA	NA	NA	~152 bytes	ASCII ID Packet sent at power up
Data received from buffer	50 50 50	48	Sequence Count	17 payload bytes	Won't change if nothing received from queue
UTC Time	50 50 50	4A	Sequence Count	17 time bytes	Current UTC Time from modem in ASCII
Serial Health and Safety Packet	50 50 50	F6	Sequence Count	19 data bytes	Sent after collecting data
Serial GPS Packet	50 50 50	F7	Sequence Count	19 data bytes	Optional. Sends current GPS solution.
Serial Mag Packet	50 50 50	0A	Sequence Count	19 data bytes	Optional. Sent after collecting data
Serial PIN Packet	50 50 50	AC	Sequence Count	17 data bytes	Optional. Sent after collecting data
Serial Plasma Packet	50 50 50	AB	Sequence Count	17 data bytes	Optional. Sent after collecting data
ACK - Good Command	AA 05	00	NA	NA	Sent after good command
NACK - Invalid Command	AA 05	FF	NA	NA	Sent after invalid command
PASS - Successful Transmission	F5	Latency	NA	NA	Sent after successful data transfer to Iridium
FAIL - Failed Transmission	F5	FF	NA	NA	Sent after failed data transfer to Iridium

unions

```
EyeStarS4_port
{
    sp_port *serial;

    struct sockaddr_in addr;

}

EyeStarS4_com
{
    sp_port *serial;

    struct sockaddr_in addr;

}

EyestarS4_header_etc
{
    uint8_t command
    uint8_t sequence

}

EyeStarS4_req
{
}
```

methods

```
EyestarS4* EyestarS4_init( )

int EyestarS4_send_downlink(EyestarS4* s4,
                           uint8_t len,
                           void data[200])

int EyestarS4_recv_uplink(EyestarS4_recv_uplink_packet* )

int EyestarS4_change_config(EyestarS4_send_change_config* )

int EyestarS4_req_h&s(EyestarS4_recv_h&s*)

int EyestarS4_req_utc(EyestarS4_recv_utc_packet* )

//int EyestarS4_req_data(REQ_TYPE type, *)

int EyestarS4_close(EyestarS4*)
```