



SIM100 CAN Protocol Reference Manual

VERSION: 0.6

SENDYNE SIM100 CAN PROTOCOL IMPLEMENTATION 3

Features 3

GENERAL MESSAGE FORMAT 3

Data types 4

Signals and Signal Groups 4

 Byte ordering 4

HOST REQUESTS MULTIPLEXOR 5

Host request multiplexor values 5

SIM100 RESPONSE MULTIPLEXOR VALUES 6

SIM100 SIGNALS 7

SIM100 signal groups 9

 Status_bits 9

 Error flags 12

 Isolation state 11

 Isolation resistances 11

 Isolation capacitances 11

 Voltages Vp and Vn 11

 Battery voltage 12

MESSAGES 13

DATA REQUESTS FROM HOST TO SIM100 13

Manufacturer’s data requests 14

 Request part name 14

 Request firmware version number 14

 Request serial number 15

Data reporting requests 16

 Request temperature 16

 Request isolation state 16

 Request isolation resistances 17

 Request isolation capacitances 19

 Voltages Vp and Vn 20

 Battery voltage 21

 Error flags 22

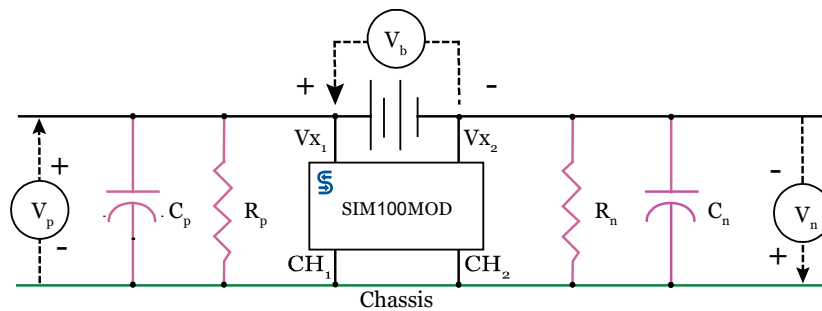
SIM100 STATE CONTROL COMMANDS FROM HOST 23

| | |
|---|----|
| <i>Restart</i> | 23 |
| <i>Turn excitation pulse off</i> | 23 |
| <i>Set Max battery working voltage</i> | 24 |
| SAMPLE SIM100 TRANSACTION..... | 25 |
| <i>Set battery Maximum Working Voltage</i> | 25 |
| <i>Read isolation state</i> | 25 |
| “REQUEST ISOLATION STATE” TYPICAL FLOWCHART | 26 |
| “REQUEST ISOLATION RESISTANCES” TYPICAL FLOWCHART | 27 |
| <i>Revision history</i> | 28 |

Sendyne SIM100 CAN Protocol Implementation

Features

- CAN 2.0B extended frame format
- 500 kbit/s or 250 kbit/s



SIM100 isolation monitoring reference diagram

General message format

The Sendyne SIM100 communicates with the host system through a command-response protocol. Communications are initiated by the host issuing a message with extended ID 0xA100101, followed by a one byte multiplexor (`Request_mux`) indicating the type of operation (read, write or command) to be performed. Depending on the type of operation either more or no data may follow as shown in the following table.

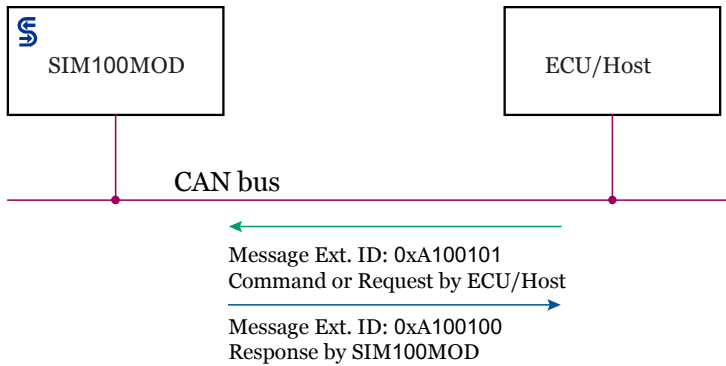
Request from Host:

| Ext. ID | byte 0 | byte 1 | byte 2 |
|-----------|--------------------------|--------|--------|
| 0xA100101 | <code>Request_mux</code> | | |

If the multiplexor value `Request_mux` specifies a request for reading a signal, the SIM100 will respond with Ext. ID 0xA100100 followed by a multiplexor byte `OpCode_mux` with the same value as the multiplexor of the request message and any data pertinent to the transaction. If the value of `Request_mux` specifies a command the SIM100 will execute the command.

SIM100 response:

| Ext. ID | byte 0 | byte 1 | byte 2 |
|-----------|-------------------------|--------|--------|
| 0xA100100 | <code>OpCode_mux</code> | | |



Data types

SIM100 data can have the length of a bit, byte, 2-bytes or 4-bytes depending on the content of the transaction. The data types can be a flag, an ASCII character, a signed or an unsigned integer; they are documented in each signal description.

Signals and Signal Groups

Data communicated in messages are defined as signals. A signal can be a flag, ASCII characters, a signed or an unsigned integer. They are defined in the signal section of this document. Signal groups consist of a collection of signals defined in the signal groups section of this document. Signals and signal group names are represented with a blue Courier font in this document.

Byte ordering

In case of multi-byte data the order of bytes within each message is specified in signal definition as a big endian (Motorola) or little endian (Intel).

Big endian (MOTOROLA) data bytes order

| byte n | byte n+1 | byte n+2 | byte n+3 |
|--------|----------|----------|----------|
| MSB | Data | Data | LSB |

Little endian (INTEL) data bytes order

| byte n | byte n+1 | byte n+2 | byte n+3 |
|--------|----------|----------|----------|
| LSB | Data | Data | MSB |

Host message multiplexor

In a host originating message byte 0 is the multiplexor specifying the type of operation (read, write or command).

Requests from the host to the SIM100

| Ext. ID | byte 0 | byte 1 | byte 2 |
|-----------|-------------|--------|--------|
| 0xA100101 | Request_mux | Data | Data |

Host request multiplexor values

The following table lists the valid Request_mux values:

SIM100_Request_mux

| Value | Name | Data bytes | Description |
|--------------------------------------|--------------------------------|------------|--|
| <i>Manufacturer's data requests</i> | | | |
| 0x01 | Request Part name 0 | None | Request signal Part_name_0 |
| 0x02 | Request Part name 1 | None | Request signal Part_name_1 |
| 0x03 | Request Part name 2 | None | Request signal Part_name_2 |
| 0x04 | Request Part name 3 | None | Request signal Part_name_3 |
| 0x05 | Request Version 0 | None | Request signal Version_0 |
| 0x06 | Request Version 1 | None | Request signal Version_1 |
| 0x07 | Request Version 2 | None | Request signal Version_2 |
| 0x08 | Request Serial number 0 | None | Request signal Serial_number_0 |
| 0x09 | Request Serial number 1 | None | Request signal Serial_number_1 |
| 0x0A | Request Serial number 2 | None | Request signal Serial_number_2 |
| 0x0B | Request Serial number 3 | None | Request signal Serial_number_3 |
| <i>SIM100 state control commands</i> | | | |
| 0x62 | Turn excitation pulse off | 4 | SIM100 shall turn off excitation voltage pulse |
| 0xC1 | Restart SIM100 | 4 | SIM100 shall restart |
| 0xF0 | Set max battery design voltage | 2 | SIM100 will set max battery design voltage |
| <i>Data reporting requests</i> | | | |
| 0x80 | Request Temperature | None | Request signal Temperature |
| 0xE0 | Request Isolation state | None | Request signal group ≈isolation_state |
| 0xE1 | Request Isolation resistances | None | Request signal group ≈isolation_resistances |
| 0xE2 | Request Isolation capacitances | None | Request signal group ≈isolation_capacitances |
| 0xE3 | Request Voltages Vp and Vn | None | Request signal group ≈voltages_Vp_and_Vn |
| 0xE4 | Request Battery voltage Vb | None | Request signal group ≈battery_voltage |
| 0xE5 | Request Error flags | None | Request signal group ≈Error_flags |

SIM100 response multiplexor values

A message from SIM100 is always transmitted as a response to a message from the host. Byte 0 of SIM100 messages is the `OpCode_mux` multiplexor of the message. Its value is the same value as the host's message multiplexor.

Responses from SIM100 to host

| Ext. ID | byte 0 | byte 1 | byte 2 |
|-----------|-------------------------|--------|--------|
| 0xA100100 | <code>OpCode_mux</code> | Data | Data |

The following table lists the valid `OpCode_mux` values in SIM100 responses. Signals preceded with a double tilde (`≈`) symbol represent signal groups (a collection of signals) which are defined later in this document.

`OpCode_mux`

| Value | Name | Data bytes | Signals (~) and signal groups (≈) |
|----------------------------|----------------------------|------------|--|
| <i>Manufacturer's data</i> | | | |
| 0x01 | Part name 0 | 4 | ~Part_name_0 |
| 0x02 | Part name 1 | 4 | ~Part_name_1 |
| 0x03 | Part name 2 | 4 | ~Part_name_2 |
| 0x04 | Part name 3 | 4 | ~Part_name_3 |
| 0x05 | Version 0 | 4 | ~Version_0 |
| 0x06 | Version 1 | 4 | ~Version_1 |
| 0x07 | Version 2 | 4 | ~Version_2 |
| 0x08 | Serial number 0 | 4 | ~Serial_number_0 |
| 0x09 | Serial number 1 | 4 | ~Serial_number_1 |
| 0x0A | Serial number 2 | 4 | ~Serial_number_2 |
| 0x0B | Serial number 3 | 4 | ~Serial_number_3 |
| <i>Environmental</i> | | | |
| 0x80 | Temperature | 4 | ~Temperature |
| <i>Isolation state</i> | | | |
| 0xE0 | Isolation state | 7 | ≈Status_bits + ≈isolation_state |
| 0xE1 | Isolation resistances | 7 | ≈Status_bits + ≈isolation_resistances |
| 0xE2 | Isolation capacitances | 7 | ≈Status_bits + ≈isolation_capacitances |
| 0xE3 | Voltages Vp and Vn | 7 | ≈Status_bits + ≈voltages_Vp_and_Vn |
| 0xE4 | Battery voltage Vb | 7 | ≈Status_bits + ≈battery_voltage |
| 0xE5 | Error flags | 2 | ≈Status_bits + ≈Error_flags |
| 0xF0 | Max battery design voltage | 2 | ~Max_battery_working_voltage |

SIM100 signals

The following table defines the available signals of SIM100.

SIM100 signals

| <i>Signal Name</i> | <i>Length [Bits]</i> | <i>Byte Order</i> | <i>Value Type</i> | <i>Unit</i> | <i>Value Table</i> | <i>Comment</i> |
|----------------------------------|----------------------|-------------------|-------------------|-------------|--------------------|---|
| Cn | 16 | M | U | nF | - | Estimated value of capacitances Cn. |
| Cn_uncertainty | 8 | M | U | % | - | Cn estimate uncertainty expressed in % |
| Cp | 16 | M | U | nF | - | Estimated values of capacitance Cp. |
| Cp_uncertainty | 8 | M | U | % | - | Cp estimate uncertainty expressed in % |
| Electrical_isolation | 16 | M | U | Ω/V | - | Minimum resistance per Volt isolation path between the IT system and the chassis. The value is calculated based on the battery's Vb_max Voltage. |
| Electrical_isolation_uncertainty | 8 | M | U | % | - | Electrical isolation uncertainty expressed in % |
| Energy_stored | 16 | M | U | mJ | - | This is the maximum energy that can be stored in the Y capacitors between the battery and chassis at the maximum working voltage. |
| Energy_stored_uncertainty | 8 | M | U | % | - | Energy stored uncertainty expressed in % |
| Err_CH | 1 | - | B | | - | 0 - CH1 and CH2 (chassis) connections are good 1 - Connection to chassis broken. |
| Err_Vexi | 1 | - | B | | - | 0 - Excitation voltage level is correct 1 - Excitation voltage level out of range |
| Err_Vpwr | 1 | - | B | | - | 0 - Power supply level is good 1 - Power supply level out of range |
| Err_Vx1 | 1 | - | B | | - | 0 - VX1 connection is good (SIM100 to battery positive terminal connection) 1 - VX1 connection broken |
| Err_Vx2 | 1 | - | B | | - | 0 - VX2 connection is good (SIM100 to battery negative terminal connection) 1 - VX2 connection broken |
| Err_VxR | 1 | - | B | | - | 0 - VX1 and VX2 connections are correct 1 - VX1 and VX2 connections are reversed |
| Excitation_pulse_off | 32 | I | U | | * | Sending data 0xDEADBE1F with SIM_Request_mux = 0x62 disables the excitation pulse of the SIM100. In order to re-enable it a Restart message has to be sent. |
| Hardware_Error | 1 | - | B | | - | 0 - No hardware error 1 - A hardware error was detected |
| High_Battery_Voltage | 1 | - | B | | - | 0 - Observed battery voltage is less than the programmed Max_battery_working_voltage value 1 - Observed battery voltage is higher than Max_battery_working_voltage |
| High_Uncertainty | 1 | - | B | | - | 0 - Uncertainty of calculated values is less than 5% 1 - Uncertainty is higher than 5% |

| <i>Signal Name</i> | <i>Length [Bits]</i> | <i>Byte Order</i> | <i>Value Type</i> | <i>Unit</i> | <i>Value Table</i> | <i>Comment</i> |
|--|----------------------|-------------------|-------------------|----------------|--------------------|--|
| <code>Isolation_status_bits</code> | 2 | - | B | | - | 00 – Isolation status OK 10 – Warning 11 - Fault |
| <code>Low_Battery_Voltage</code> | 1 | - | B | | - | 0 – Observed battery voltage higher than 15 V 1 – Observed battery voltage less than 15 V |
| <code>Max_battery_working_voltage</code> | 16 | M | U | V | - | Maximum battery operating voltage (in Volts) written to <code>Vb_max</code> |
| <code>No_New_Estimates</code> | 1 | M | U | | - | 0 – The flag is zero when new and unread isolation values have been calculated 1 – No new estimates |
| <code>Part_name0</code> | 32 | I | A | | - | The first ASCII characters of part name 1/4 |
| <code>Part_name_1</code> | 32 | I | A | | - | ASCII representation of part name 2/4 |
| <code>Part_name_2</code> | 32 | I | A | | - | ASCII representation of part name 3/4 |
| <code>Part_name_3</code> | 32 | I | A | | - | ASCII representation of part name 4/4 |
| <code>Restart</code> | 32 | I | U | | * | Signal to restart the operation of SIM100. Use data value 0x1234567. |
| <code>Rn</code> | 16 | M | U | k Ω | - | Estimate of total resistance between negative rail and chassis |
| <code>Rn_uncertainty</code> | 8 | M | U | % | - | Rn estimate uncertainty in % |
| <code>Rp</code> | 16 | M | U | k Ω | - | Estimate of total resistance between positive rail and chassis |
| <code>Rp_uncertainty</code> | 8 | M | U | % | - | Rp estimate uncertainty |
| <code>Serial_number_0</code> | 32 | I | U | | - | Unit serial number, 1/4 |
| <code>Serial_number_1</code> | 32 | I | U | | - | Unit serial number, 2/4 |
| <code>Serial_number_2</code> | 32 | I | U | | - | Unit serial number, 3/4 |
| <code>Serial_number_3</code> | 32 | I | U | | - | Unit serial number, 4/4 |
| <code>Temperature</code> | 32 | M | S | m $^{\circ}$ C | - | Temperature in milli Celsius |
| <code>Vb</code> | 16 | M | U | V | - | Reports voltage of the monitored IT power system. The reported value is averaged and updated every 100 ms. |
| <code>Vb_max</code> | 16 | M | U | V | - | Maximum value of IT power supply voltage. It is the maximum between <code>Max_battery_voltage</code> and the maximum actual value recorded by SIM100MOD. |
| <code>Vb_max_uncertainty</code> | 8 | M | U | % | - | Vb_max uncertainty in % (if Vb_max is the recorded value) |
| <code>Vb_uncertainty</code> | 8 | M | U | % | - | Vb uncertainty in % |
| <code>Version_0</code> | 32 | I | A | | - | The first 4 ASCII characters of SW version, 1/3 |
| <code>Version_1</code> | 32 | I | A | | - | SW version, 2/3 |
| <code>Version_2</code> | 32 | I | A | | - | SW version, 3/3 |
| <code>Vn</code> | 16 | M | S | V | - | Potential between negative IT system power rail and chassis. Dynamic value includes excitation voltage effect. |
| <code>Vn_uncertainty</code> | 8 | M | S | % | - | Vn uncertainty in % |
| <code>Vp</code> | 16 | M | S | V | - | Potential between positive IT system power rail and chassis. Dynamic value includes excitation voltage effect. |
| <code>Vp_uncertainty</code> | 8 | M | S | % | - | Vp uncertainty in % |

| <i>Signal Name</i> | <i>Length [Bits]</i> | <i>Byte Order</i> | <i>Value Type</i> | <i>Unit</i> | <i>Value Table</i> | <i>Comment</i> |
|---|----------------------|-------------------|-------------------|-------------|--------------------|----------------|
| U – unsigned integer S – signed integer B – Boolean A – ASCII M – Motorola byte order (big endian) I – Intel byte order (little endian) * - indicates that a value table (data) is associated with the signal | | | | | | |

SIM100 signal groups

Status_bits

The `Status_bits` byte is a collection of signal bits that provides concentrated information for the state of the isolation system as well as of the proper operation of SIM100. The `Status_bits` signal group forms the first data byte in the following message reports of the SIM100:

| <code>OpCode_mux</code> | Report |
|-------------------------|------------------------|
| 0xE0 | Isolation state |
| 0xE1 | Isolation resistances |
| 0xE2 | Isolation capacitances |
| 0xE3 | Voltages Vp and Vn |
| 0xE4 | Battery voltage Vb |
| 0xE5 | Error flags |

The layout of the signal bits within the `Status_bits` signal group is shown below:

`≈Status_bits`

| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| HE | NE | HU | - | HV | LV | IS1 | ISO |

Status bits

| <i>bit</i> | <i>Symbol</i> | <i>Signal</i> | <i>Description</i> |
|------------|---------------|-----------------------|--|
| 7 | HE | Hardware_Error | 0 – No hardware error 1 – Hardware error. The host should perform a “Read SIM100 Error Flags” operation in order to resolve the issue |
| 6 | NE | No_New_Estimates | 0 – The flag is zero when new and unread isolation values have been calculated. 1 – Isolation values have not been updated since the last read. This may happen when a host requests to read isolation values while the SIM100 is in the middle of a calculation. In this case the SIM100 will return the most recent calculated values. |
| 5 | HU | High_Uncertainty | 0 – Uncertainty of calculated values is less than 5% 1 – Uncertainty values are higher than 5%. The uncertainty values always accompany reported data. |
| 4 | - | Undefined | Reserved for future use; this bit always has a value of zero |
| 3 | HV | High_Battery_Voltage | 0 – Observed battery voltage is less than the specified <code>Max_battery_working_voltage</code> . 1 – The observed battery voltage value is higher than the specified <code>Max_battery_working_voltage</code> value. The flag will be set if the <code>Max_battery_working_voltage</code> register has not been set, or if the set value is less than the maximum observed battery voltage value. If this flag is set, isolation resistance and stored energy will be calculated based on the maximum observed battery voltage. |
| 2 | LV | Low_Battery_Voltage | 0 – Observed battery voltage higher than 15 V 1 – Observed battery voltage less than 15 V. This flag is also set when battery is disconnected. |
| 1-0 | IS1-IS0 | Isolation_status_bits | 00 – Isolation status OK 10 – Warning. Isolation resistance < 500 Ohm/V limit. 11 – Isolation fault. Isolation resistance < 100 Ohm/V limit. |

The following signal groups are defined and used in SIM100 messages

Isolation state

`≈isolation_state`

| <i>Start byte</i> | <i>Signal</i> |
|-------------------|----------------------------------|
| 2 | Electrical_isolation |
| 4 | Electrical_isolation_uncertainty |
| 5 | Energy_stored |
| 7 | Energy_stored_uncertainty |

Isolation resistances

`≈isolation_resistances`

| <i>Start byte</i> | <i>Signal</i> |
|-------------------|----------------|
| 2 | Rp |
| 4 | Rp_uncertainty |
| 5 | Rn |
| 7 | Rn_uncertainty |

Isolation capacitances

`≈isolation_capacitances`

| <i>Start byte</i> | <i>Signal</i> |
|-------------------|----------------|
| 2 | Cp |
| 4 | Cp_uncertainty |
| 5 | Cn |
| 7 | Cn_uncertainty |

Voltages Vp and Vn

`≈voltages_Vp_and_Vn`

| <i>Start byte</i> | <i>Signal</i> |
|-------------------|----------------|
| 2 | Vp |
| 4 | Vp_uncertainty |
| 5 | Vn |
| 7 | Vn_uncertainty |

Battery voltage

≈battery_voltage

| Start byte | Signal |
|------------|--------------------|
| 2 | Vb |
| 4 | Vb_uncertainty |
| 5 | Vb_max |
| 7 | Vb_max_uncertainty |

Error flags

The `Error_flags` byte is a collection of one bit signals which are updated during the continuous self-checking of SIM100. If any of these flags is set, the signal bit `Hardware_error` in the `Status_bits` will be set.

≈Error_flags

| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---------|---------|--------|---------|----------|----------|-------|-------|
| Err_Vx2 | Err_Vx1 | Err_CH | Err_VxR | Err_Vexi | Err_Vpwr | - | - |

Error flags

| Bit | Symbol | Signal | Description |
|-----|------------------|----------|---|
| 7 | V _{X2} | Err_Vx2 | 0 – V _{X2} connection is good (SIM100 to battery negative terminal) 1 – V _{X2} connection is broken |
| 6 | V _{X1} | Err_Vx1 | 0 – V _{X1} connection is good (SIM100 to battery positive terminal) 1 – V _{X1} connection is broken |
| 5 | CH | Err_CH | 0 – CH ₁ and CH ₂ connections are good (chassis connections) 1 – CH ₁ or CH ₂ connection is broken |
| 4 | V _{XR} | Err_VxR | 0 – V _{X1} and V _{X2} connection are correct 1 – V _{X1} and V _{X2} connection are reversed |
| 3 | V _{EXI} | Err_Vexi | 0 – Excitation voltage level is correct 1 – Excitation voltage level is out of specs |
| 2 | V _{PWR} | Err_Vpwr | 0 – Power supply level is good 1 – Power supply level is out of range |
| 1-0 | | Not used | Reserved |

Messages

Data requests from host to SIM100

This group consists of single byte messages issued by the host in order to poll the SIM100 for data. The SIM100 will respond to each one of these requests by sending a multiplexed message with the same multiplexor value as the multiplexor of the request followed by the signal group data requested.

| | |
|-----------|-----------------------------|
| Ext. ID | byte 0 |
| 0xA100101 | Request_mux |

SIM100_Request_mux

| <i>Value</i> | <i>Name</i> | <i>Data bytes</i> | <i>Description</i> |
|-------------------------------------|----------------------------------|-------------------|--|
| <i>Manufacturer's data requests</i> | | | |
| 0x01 | Request Part name 0 | None | Request signal Part_name_0 |
| 0x02 | Request Part name 1 | None | Request signal Part_name_1 |
| 0x03 | Request Part name 2 | None | Request signal Part_name_2 |
| 0x04 | Request Part name 3 | None | Request signal Part_name_3 |
| 0x05 | Request Version 0 | None | Request signal Version_0 |
| 0x06 | Request Version 1 | None | Request signal Version_1 |
| 0x07 | Request Version 2 | None | Request signal Version_2 |
| 0x08 | Request Serial number 0 | None | Request signal Serial_number_0 |
| 0x09 | Request Serial number 1 | None | Request signal Serial_number_1 |
| 0x0A | Request Serial number 2 | None | Request signal Serial_number_2 |
| 0x0B | Request Serial number 3 | None | Request signal Serial_number_3 |
| <i>Data reporting requests</i> | | | |
| 0x80 | Request Temperature | None | Request signal Temperature |
| 0xE0 | Request Isolation state | None | SIM100 will report "Isolation state" |
| 0xE1 | Request Isolation resistances | None | SIM100 will report "Isolation resistances" |
| 0xE2 | Request Isolation capacitances | None | SIM100 will report "Isolation capacitances" |
| 0xE3 | Request Voltages V_p and V_n | None | SIM100 will report "Voltage V_p and V_n " |
| 0xE4 | Request Battery voltage V_b | None | SIM100 will report "Battery Voltage" |
| 0xE5 | Request Error flags | None | SIM100 will report "Error flags" |

Manufacturer's data requests**Request part name**

SIM100 part name consists of 16 ASCII characters. The host can retrieve the part name through four message transactions. Each of the four `Part_name_N` signals is 32 bits (4 characters) arranged in Intel byte order.

| <i>Request_mux</i> | <i>Name</i> | <i>Data bytes</i> | <i>Description</i> |
|--------------------|---------------------|-------------------|---|
| 0x01 | Request Part name 0 | None | Request signal <code>Part_name_0</code> |
| 0x02 | Request Part name 1 | None | Request signal <code>Part_name_1</code> |
| 0x03 | Request Part name 2 | None | Request signal <code>Part_name_2</code> |
| 0x04 | Request Part name 3 | None | Request signal <code>Part_name_3</code> |

Request from host:

| Ext. ID | <code>Request_mux</code> |
|-----------|--------------------------|
| 0xA100101 | 0x0N |

Where N can be 1, 2, 3 or 4

Response from SIM100:

| Ext. ID | <code>OpCode_mux</code> | byte 1-4 |
|-----------|-------------------------|------------------------------|
| 0xA100101 | 0x0N | <code>Part_name_(N-1)</code> |

Where N can be 1, 2, 3 or 4

The SIM100 part name can be formed by concatenating the four signals

| SIM100 Part Name |
|--|
| <code>Part_name_3 Part_name_2 Part_name_1 Part_name_0</code> |

Request firmware version number

SIM100 version number consists of 12 ASCII characters. The host can retrieve the version number through three message transactions. Each of the three `Version_N` signals is 32 bits (4 characters) arranged in Intel byte order.

| <i>Request_mux</i> | <i>Name</i> | <i>Data bytes</i> | <i>Description</i> |
|--------------------|-------------------|-------------------|---------------------------------------|
| 0x05 | Request version 0 | None | Request signal <code>Version_0</code> |
| 0x06 | Request version 1 | None | Request signal <code>Version_1</code> |
| 0x06 | Request version 2 | None | Request signal <code>Version_2</code> |

Request rom host:

| Ext. ID | Request_mux |
|-----------|-------------|
| 0xA100101 | 0x0N |

Where N can be 5,6, or 7

Response from SIM100:

| Ext. ID | OpCode_mux | byte 1-3 |
|-----------|------------|---------------|
| 0xA100101 | 0x0N | Version_(N-5) |

Where N can be 5,6, or 7

The SIM100 firmware version can be formed by concatenating the three signals

| SIM100 Version | | |
|----------------|-----------|-----------|
| Version_2 | Version_1 | Version_0 |

Request serial number

Serial number is unique for every SIM100 MCUs and consists of 128 bits. The host can retrieve the serial number through four message transactions. Each of the four `Serial_number_N` signals is 32 bits arranged in Intel byte order.

| Request_mux | Name | Data bytes | Description |
|-------------|-------------------------|------------|---|
| 0x08 | Request Serial number 0 | None | Request signal <code>Serial_number_0</code> |
| 0x09 | Request Serial number 1 | None | Request signal <code>Serial_number_1</code> |
| 0x0A | Request Serial number 2 | None | Request signal <code>Serial_number_2</code> |
| 0x0B | Request Serial number 3 | None | Request signal <code>Serial_number_3</code> |

Request rom host:

| Ext. ID | Request_mux |
|-----------|-------------|
| 0xA100101 | 0x0N |

Where N can be 8, 9, A or B

Response from SIM100:

| Ext. ID | OpCode_mux | byte 1-4 |
|-----------|------------|---------------------|
| 0xA100101 | 0x0N | Serial_number_(N-8) |

Where N can be 8, 9, A or B

The SIM100 serial number can be formed by concatenating the four signals as follows:

```
SIM100 Serial number
Serial_number_3 Serial_number_2 Serial_number_1 Serial_number_0
```

Data reporting requests

Request temperature

SIM100MOD monitors environmental temperature and can communicate it to the host through the `Temperature` 32 bit signed integer signal (mCelsius units). The `Temperature` signal byte order is Motorola (Big endian).

To read `Temperature` the host sends:

Request from host:

| Ext. ID | Request_mux |
|-----------|-------------|
| 0xA100101 | 0x80 |

SIM100 response:

| Ext. ID | OpCode_mux | byte 1-4 |
|-----------|------------|-------------|
| 0xA100101 | 0x80 | Temperature |

Request isolation state

The “Request isolation state” and its response is intended to provide in a single message an overview of the safety state of the isolation system. The “Request isolation state” message of the host is as follows:

Request from Host:

| Ext. ID | Request_mux |
|-----------|-------------|
| 0xA100101 | 0xE0 |

On “Request isolation state” the SIM100 will respond with a message composed of two “signal groups”, the `≈Status_bits` and `≈isolation_state`.

Response from SIM100:

| Ext. ID | OpCode_mux | byte 1 | bytes 2-7 |
|-----------|------------|--------------|------------------|
| 0xA100101 | 0xE0 | ≈Status_bits | ≈isolation_state |

The ≈Status_bits signal group is a collection of flags described in the “SIM100 signal groups” section of this document. They provide information on whether a warning or fault condition has occurred, on the success or failure of SIM100’s self-check, as well as other information related to the quality of the estimates and the voltage conditions of the IT system and should be checked in each communication in order to validate the estimates provided.

The ≈isolation_state signal group provides the following information:

| Signal | Byte# | Units | Description |
|----------------------------------|-------|-------|--|
| Electrical_isolation | 2-3 | Ω/V | This value corresponds to the <i>minimum</i> resistance path between the positive or negative rail and chassis. It is calculated as: $\min(R_p, R_n)/V_{b_max}$, where $\min(R_p, R_n)$ is the minimum isolation resistance between the positive or negative rail and chassis. If the “warning” or “fault” <i>Isolation_status_bits</i> are set the host should check the ≈isolation_resistances signal group for the presence of a hazardous symmetrical fault. |
| Electrical_isolation_uncertainty | 4 | % | Uncertainty of <i>Electrical_isolation</i> estimate |
| Energy_stored | 5-6 | mJ | Maximum stored energy at <i>Vb_max</i> in mJ. <i>Energy_stored</i> is calculated as: $0.5 * (C_p + C_n) * V_{b_max}^2$ |
| Energy_stored_uncertainty | 7 | % | Uncertainty of <i>Energy_stored</i> estimate |

Request isolation resistances

The “Request isolation resistances” and its response is intended to provide individual estimates for the isolation resistance values between the positive and negative power rails and the chassis. Besides cross-checking these estimates against pre-programmed fault values, in case of a warning or fault condition, by checking these values the host can determine if there is a symmetrical fault. A symmetrical fault can lead to high temperatures and power loss and unlike single faults is not controllable. The “Request isolation resistances” format is as follows:

Request from host:

| Ext. ID | Request_mux |
|-----------|-------------|
| 0xA100101 | 0xE1 |

On “Request isolation resistances” the SIM100 will respond with a message composed of two “signal groups”, the `≈Status_bits` and `≈isolation_resistances`.

Response from SIM100:

| Ext. ID | OpCode_mux | byte 1 | Byte 2-7 |
|-----------|------------|---------------------------|-------------------------------------|
| 0xA100101 | 0xE1 | <code>≈Status_bits</code> | <code>≈isolation_resistances</code> |

The `≈Status_bits` signal group is a collection of flags described in the “SIM100 signal groups” section of this document and as with any isolation state related message they should be checked to validate the estimates, the condition of the SIM100 and the presence of any warnings or alerts.

The `≈isolation_resistances` signal group provides the following information:

| <i>Signal</i> | <i>Byte#</i> | <i>Units</i> | <i>Description</i> |
|-----------------------------|--------------|--------------|---|
| <code>Rp</code> | 2-3 | kΩ | Estimate for the total resistance between the positive power rail and chassis |
| <code>Rp_uncertainty</code> | 4 | % | Uncertainty of <code>Rp</code> estimate |
| <code>Rn</code> | 5-6 | kΩ | Estimate for the total resistance between the negative power rail and chassis |
| <code>Rn_uncertainty</code> | 7 | % | Uncertainty of <code>Rn</code> estimate |

Request isolation capacitances

The “Request isolation capacitances” message and the SIM100 response is intended to provide individual estimates for the isolation capacitance values between the positive and negative power rails and the chassis. SIM100 utilizes these values to calculate potentially hazardous energy stored. The “Request isolation capacitances” format is as follows:

Request from host:

| Ext. ID | Request_mux |
|-----------|-------------|
| 0xA100101 | 0xE2 |

On “Request isolation capacitances” the SIM100 will respond with a message composed of two “signal groups”, the `~Status_bits` and `~isolation_capacitances`.

SIM100 response:

| Ext. ID | OpCode_mux | byte 1 | Byte 2-7 |
|-----------|------------|---------------------------|--------------------------------------|
| 0xA100101 | 0xE2 | <code>~Status_bits</code> | <code>~isolation_capacitances</code> |

The `~Status_bits` signal group is a collection of flags described in the “SIM100 signal groups” section of this document and as with any isolation state related message they should be checked to validate the estimates, the condition of the SIM100 and the presence of any warnings or alerts.

The `~isolation_capacitances` signal group provides the following information:

| Signal | Byte# | Units | Description |
|-----------------------------|-------|-------|---|
| <code>Cp</code> | 2-3 | nF | Estimate for the total capacitance between the positive power rail and chassis. |
| <code>Cp_uncertainty</code> | 4 | % | Uncertainty of <code>Cp</code> estimate |
| <code>Cn</code> | 5-6 | nF | Estimate for the total capacitance between the negative power rail and chassis. |
| <code>Cn_uncertainty</code> | 7 | % | Uncertainty of <code>Cn</code> estimate |

Voltages Vp and Vn

The “Request voltages Vp and Vn” message and the SIM100 response is intended to provide individual measurements for the voltages between the positive and negative power rails and the chassis. Vp and Vn values are updated and can be sampled every 10 ms. If they are sampled at a lower frequency the voltage values will represent the average value between successive reads. The sampled values of Vp and Vn include the effect of the excitation voltage pulse of SIM100. The sum of Vp+Vn provides the battery voltage Vb. The “Request voltages Vp and Vn” format is as follows:

Request from host:

| Ext. ID | Request_mux |
|-----------|-------------|
| 0xA100101 | 0xE3 |

On “Request voltages Vp and Vn” the SIM100 will respond with a message composed of two “signal groups”, the `≈Status_bits` and `≈voltages_Vp_and_Vn`.

SIM100 response:

| Ext. ID | OpCode_mux | byte 1 | bytes 2-7 |
|-----------|------------|---------------------------|----------------------------------|
| 0xA100101 | 0xE3 | <code>≈Status_bits</code> | <code>≈voltages_Vp_and_Vn</code> |

The `≈Status_bits` signal group is a collection of flags described in the “SIM100 signal groups” section of this document and as with any isolation state related message they should be checked to validate the estimates, the condition of the SIM100 and the presence of any warnings or alerts.

The `≈voltages_Vp_and_Vn` signal group provides the following information:

| <i>Signal</i> | <i>Byte#</i> | <i>Units</i> | <i>Description</i> |
|----------------|--------------|--------------|--|
| Vp | 2-3 | V | Measured voltage between the positive power rail and chassis |
| Vp_uncertainty | 4 | % | Uncertainty of Vp measurement |
| Vn | 5-6 | V | Measured voltage between the negative power rail and chassis |
| Vn_uncertainty | 7 | % | Uncertainty of Vn measurement |

Battery voltage

The “Request battery voltage” message and the SIM100 response is intended to provide a measurements for the battery voltage and its maximum value. The battery voltage value `Vb` is updated every 100 ms and corresponds to the average of the measurements over this period. The maximum battery voltage value `Vb_max` is the maximum value between the `Max_battery_working_voltage` value and the maximum actual `Vb` value recorded by SIM100 since power-on or reset. Default value of `Max_battery_working_voltage` is zero. If the `Max_battery_working_voltage` is not set by the host then `Vb_max` will be tracking the maximum value measured by SIM100. The `Vb_max` value is utilized by SIM100 to determine a warning or fault condition and set the appropriate flags in `≈Status_bits`. The “Request voltages `Vp` and `Vn`” format is as follows:

Request from host:

| Ext. ID | Request_mux |
|-----------|-------------|
| 0xA100101 | 0xE4 |

On “Request voltages `Vp` and `Vn`” the SIM100 will respond with a message composed of two “signal groups”, the `≈Status_bits` and `≈battery_voltage`.

SIM100 response:

| Ext. ID | OpCode_mux | byte 1 | bytes 2-7 |
|-----------|------------|---------------------------|-------------------------------|
| 0xA100101 | 0xE4 | <code>≈Status_bits</code> | <code>≈battery_voltage</code> |

The `≈Status_bits` signal group is a collection of flags described in the “SIM100 signal groups” section of this document and as with any isolation state related message they should be checked to validate the estimates, the condition of the SIM100 and the presence of any warnings or alerts.

The `≈battery_voltage` signal group provides the following information:

| Signal | Byte# | Units | Description |
|---------------------------------|-------|-------|--|
| <code>Vb</code> | 2-3 | V | Measured DC power supply voltage |
| <code>Vb_uncertainty</code> | 4 | % | Uncertainty of <code>Vb</code> measurement |
| <code>Vb_max</code> | 5-6 | V | Maximum between <code>Max_battery_working_voltage</code> and measured <code>Vb</code> voltage since power-on or reset. |
| <code>Vb_max_uncertainty</code> | 7 | % | Uncertainty of <code>Vb_max</code> if it represents measured value |

Error flags

The “Request error flags” message and the SIM100 response is intended to provide diagnostic information derived during the self-test of SIM100. This message should be invoked by the host anytime the `Hardware_Error` flag in the `≈Status_bits` signal group is set. The “Request error flags” format is as follows:

Request from host:

| Ext. ID | Request_mux |
|-----------|-------------|
| 0xA100101 | 0xE5 |

On “Request error flags” the SIM100 will respond with a message composed of two “signal groups”, the `≈Status_bits` and `≈Error_flags`.

SIM100 response:

| Ext. ID | OpCode_mux | byte 1 | Byte 2 |
|-----------|------------|---------------------------|---------------------------|
| 0xA100101 | 0xE5 | <code>≈Status_bits</code> | <code>≈Error_flags</code> |

The `≈Status_bits` signal group is a collection of flags described in the “SIM100 signal groups” section of this document and as with any isolation state related message they should be checked to validate the estimates, the condition of the SIM100 and the presence of any warnings or alerts.

The `≈Error_flags` signal group provides the following information:

| Signal | Bit# | Units | Description |
|-----------------------|------|---------|--|
| <code>Err_Vx2</code> | 7 | Boolean | Set if connection to negative power rail is broken |
| <code>Err_Vx1</code> | 6 | Boolean | Set if connection to positive power rail is broken |
| <code>Err_CH</code> | 5 | Boolean | Set if connection to Chassis is broken |
| <code>Err_VxR</code> | 4 | Boolean | Set if connection to power rails is reversed |
| <code>Err_Vexi</code> | 3 | Boolean | Set if excitation voltage is out of range |
| <code>Err_Vpwr</code> | 2 | Boolean | Set if SIM100 power supply is out of range |
| Reserved | 1-0 | | Reserved |

SIM100 state control commands from Host

Restart

The “Restart” command forces the SIM100 to enter a power-on state. Specifically:

- Clears all flags in \approx Status_bits signal group
- Clears all flags in \approx Error_flags signal group
- Clears all isolation state estimates
- Reloads from flash memory Max_battery_working_voltage into Vb_max

After a “Restart” command the SIM100 will perform self-check and will produce new estimates and update flags within 5 s.

The “Restart” command is as follows:

| Ext. ID | Request_mux | Byte 1 | Byte 2 | Byte 3 | Byte 4 |
|-----------|-------------|--------|--------|--------|--------|
| 0xA100101 | 0xC1 | 0x67 | 0x45 | 0x23 | 0x01 |

Turn excitation pulse off

The “Turn excitation pulse off” disables the excitation pulse of SIM100 and suspends its isolation monitoring function. The purpose of this command is to prevent SIM100 from interfering with another insulation monitoring device which is currently active. For example, if the SIM100 operates in a vehicle the “Turn excitation pulse off” command shall be used when and while the vehicle is attached to a DC quick charging station. While the excitation pulse is turned off measurements will not be valid and the relevant error flags will be set. The SIM100 shall resume its isolation monitoring function through the issuance of a “Restart” command.

The “Turn excitation pulse off” command is as follows:

| Ext. ID | Request_mux | Byte 1 | Byte 2 | Byte 3 | Byte 4 |
|-----------|-------------|--------|--------|--------|--------|
| 0xA100101 | 0x62 | 0x1F | 0xBE | 0xAD | 0xDE |

Set Max battery working voltage

The “Set Max battery working voltage” sets the `Vb_max` to the provided `Max_battery_working_voltage` value, which is used for estimating warning and fault conditions in the isolation system. The `Vb_max` value will change upwards to the maximum measured `Vb` value. The “`Max_battery_working_voltage`” value is stored in flash and it is restored into `Vb_max` after power-on or a “Restart” message. After a new `Max_battery_working_voltage` is set, a “Restart” command has to be issued in order for the new value to take effect.

The transaction is as follows:

Request from Host:

| Ext. ID | Request_mux | Bytes 1-2 |
|-----------|-------------|-----------------------------|
| 0xA100101 | 0xF0 | Max_battery_working_voltage |

Response from SIM100:

| Ext. ID | Request_mux | Bytes 1-2 |
|-----------|-------------|-----------------------------|
| 0xA100100 | 0xF0 | Max_battery_working_voltage |

Sample SIM100 transaction

Set battery Maximum Working Voltage

In this sample transaction the host sets the Maximum Working Voltage of the battery to 600 Volts

Request from Host

| Ext. ID | byte 0 | byte 1 | byte 2 |
|-----------|--------|--------|--------|
| 0xA100101 | 0xF0 | 0x02 | 0x58 |

Response from SIM100

| Ext. ID | byte 0 | byte 1 | byte 2 |
|-----------|--------|--------|--------|
| 0xA100100 | 0xF0 | 0x02 | 0x58 |

Read isolation state

In this example the host requests the isolation state of the IT system. The SIM100 responds with new data indicating minimum electrical isolation of 550 Ω/V with uncertainty of 2% and maximum energy stored in capacitors under maximum working voltage of 80 mJ with uncertainty of 4%.

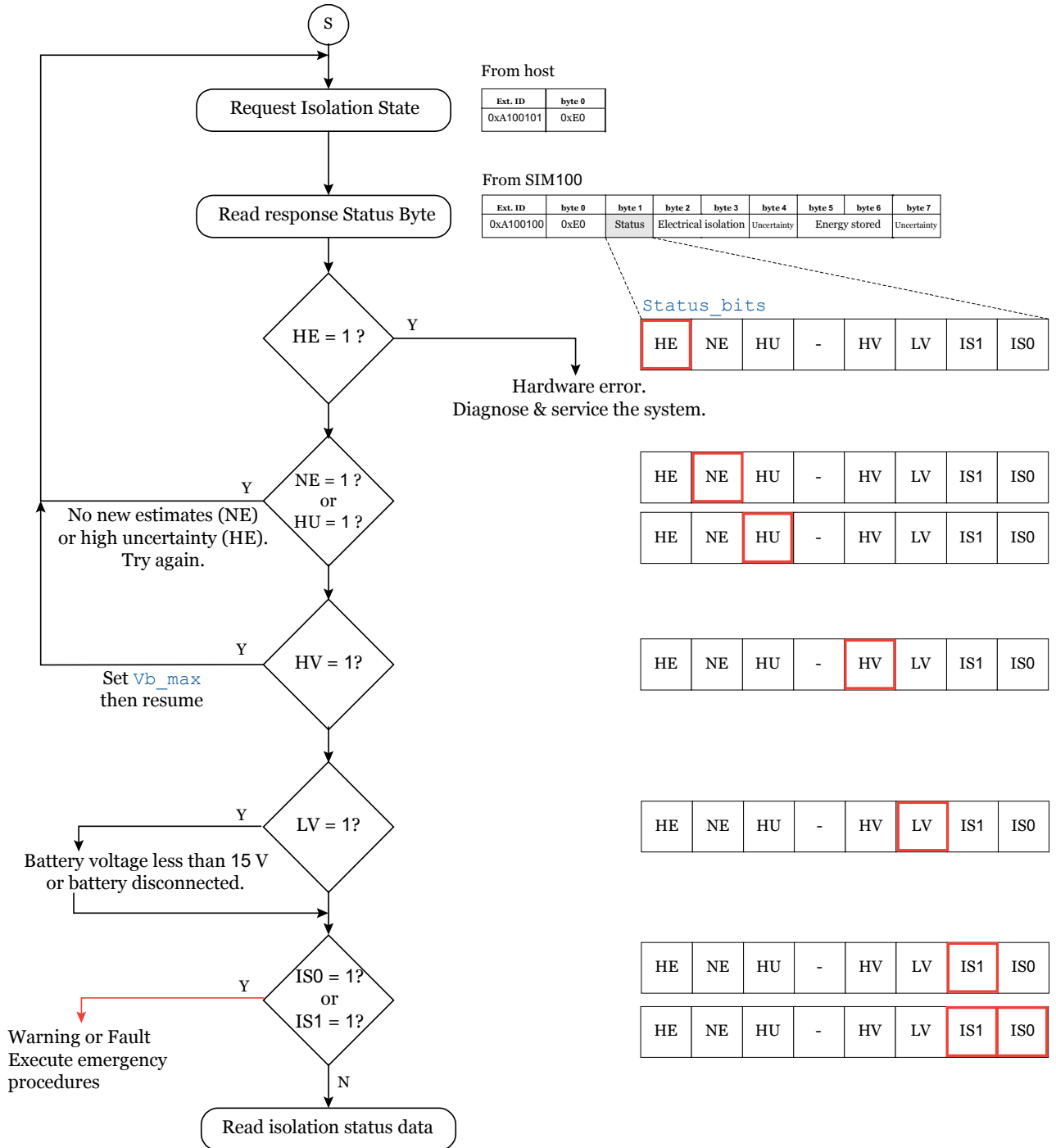
Request from Host

| Ext. ID | byte 0 |
|-----------|--------|
| 0xA100101 | 0xE0 |

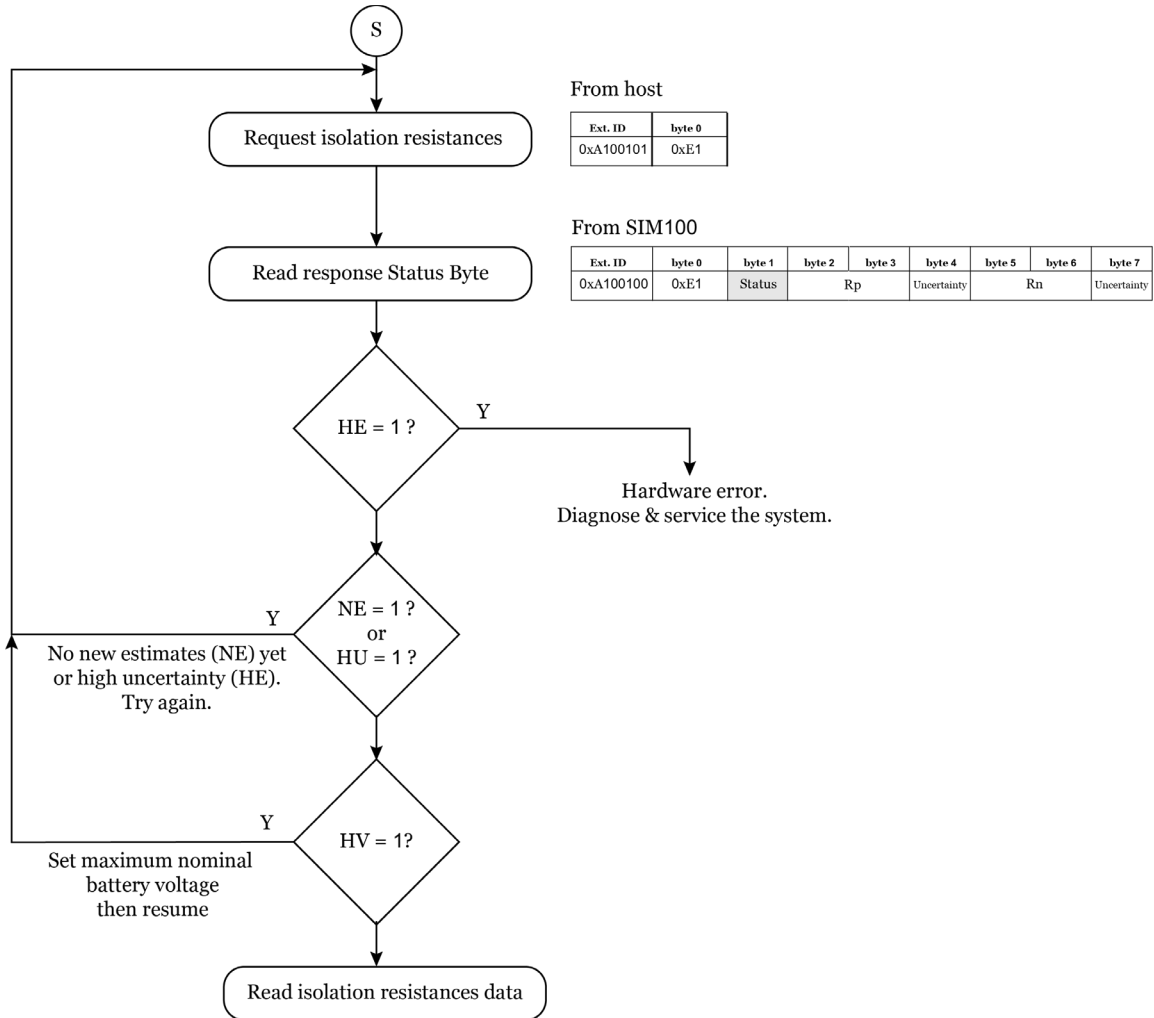
Response from SIM100

| Ext. ID | byte 0 | byte 1 | byte 2 | byte 3 | byte 4 | byte 5 | byte 6 | byte 7 |
|-----------|--------|--------|--------|--------|--------|--------|--------|--------|
| 0xA100100 | 0xE0 | 0x00 | 0x02 | 0x26 | 0x02 | 0x00 | 0x50 | 0x04 |

Read Status_bits typical flowchart



“Request isolation resistances” typical flowchart



Revision history

| | |
|---------------|---|
| V 0.6 | Documented Excitation pulse off command. Reorganized the protocol document |
| V 0.5 | Corrected subscripts of Cs & Rs in Fig 1. |
| V 0.4 | SIM100 reports isolation resistances estimates with low voltage present. Prior versions were reporting parallel combination. Clarify behavior when a short is detected (all estimates set to zero). |
| V 0.3 | Defined isolation status bits into Status Byte |
| V 0.2a | Corrected Serial number encoding description to Hex. |
| v 0.2 | Added flowcharts for typical operations. Description of “Request isolation resistances” when battery is disconnected or below 15 Volts. |
| v 0.1 | Initial release |