Meridian Automation Interface for 218.
JMB 9th May 2017.

Please confirm you have the latest version of this document before commencing work.

A new automation interface has been designed for the 218 and similar Ethernet connected products. Some products also make this available on RS232. The commands, responses and operating principles are identical in both cases.

The primary intention of the protocol is to provide a graphical user interface for products that only have simple front panel displays themselves. It can either be used to create a stand-alone controller for the product or be integrated into a wider control system that spans various products – how the information is presented is up to the installation designer. At its simplest the interface can drive a small text screen in the form of other Meridian products (like the G or 800 series), with source and volume as the main display and other informative screens for menus and stream status.

Basic format.

The connection to the product is by a raw TCP stream on port 9014. All messages to and from the device are human readable strings terminated by a Line Feed ("\n" 0x0A). Carriage returns are ignored, should the line termination be CR/LF. Up to 5 connections are allowed at once.

There are three types of message:

- Commands
- Queries
- Unsolicited information

Commands are denoted by the '#' character, they will always provoke a reply (denoted by the '*' character):

- A positive acknowledgement "`*ACK\n`"
- A negative acknowledgement "`*NAK\n`"
- An error message "`*ERR\n`"

An ACK means the command has been accepted and will be acted upon. Any changes occurring due to the command will be sent separately as an unsolicited message (to all connected devices).
A NAK will be followed by a string to help explain the reason; this will be more useful to the developer than the customer.
An ERR will be followed by a string to explain the error. This is expected to indicate an interface implementation error, rather than something a customer provokes in normal operation.

Queries are denoted by the '?' character and will always reply, either:

- Positively, with the answer to the query
- Negatively, with a simple descriptive string for the reason "`*NAK\n`"
- With an error "`*ERR\n`"

A query is a request for information, whereby the original request code is part of the positive reply, so, for example, "`?PID\n`" replies "`*PID` <info>".

Information is delivered as Name:String pairs, separated by <space> characters if more than one piece of data is present. Names are unique to each piece of information and do not contain any spaces – they are made up of printable ASCII characters only. Strings are contained within double quotes and contain text suitable for presentation to the customer (which may contain spaces).

Unsolicited information can be a result of either a command from one of the connected interfaces or a change in operating conditions, such as the audio stream changing sample rate. They are denoted by a '!' character, followed by three characters to describe the command and details of the new status.

Maintaining a connection.

To cater for the possibility of users closing/removing control devices the Meridian product will periodically test a control connection with a ‘ping’ command. If there is no reply then the TCP connection is closed. The control interface may also send the ping command to detect the continued presence of the product, if it wishes. For development purposes, this feature can be disabled – it should not be disabled in a customer unit though.

Test/evaluation tools.
The interface is text only, so it is possible to control the product via simple terminal programs such as Putty. Note that disabling the ping requirement will likely be required to use the product via this means.

In addition to the commands described in this document there is a ‘help’ function, which may be useful to developers. This will always reply with the full list of commands that are accepted by the product.

Connecting via RS232.
Some products will present this interface on RS232, either on standard 9 pin D-type connectors or via SpeakerLink. If there are multiple SpeakerLink connections available then there may be a choice for which to use for automation. The baud rate of the connection will also be configurable, higher speeds should give a more responsive interface, but there may be restrictions within other products. Please refer to individual products for their configuration options.

Temporary displays.
Some aspects of the Meridian UI work best by drawing information for a short period of time in order to inform the user of some change in status or to confirm an action. The automation interface asks the UI to do this with the “`!TMP`” command, eg. :

```
“!TMP Display:”Menus Cleared” Period:”3”
```

where the period is measured in seconds.
Note that menu functionality does not use this method as it is up to the UI designer to decide how to draw changes in menu status.

Commands.

Product identification.

```
?PID
```

Replies:

```
*PID Product:”218” SerialNumber:”100001” VersionNumber:”169” ZoneName:”218 #0024c500a463”
```

Note that when a new TCP connection is made to the product it will send an unsolicited identification string, e.g.

```
!PID Product:”218” SerialNumber:”100001” VersionNumber:”169” ZoneName:”218 #0024c500a463”
```

Ping.

```
#PNG
```

Replies:

```
*PNG
```

Similarily, after 5 minutes without any messages, the product will send `#PNG` and expect `*PNG` in return. To disable this feature send:

```
$DEV
```

This setting is not stored, so will revert to normal behaviour after a reset.

System control – MSR emulation.

The Meridian System Remote has been the principal method of controlling a Meridian system for many years. All of the commands a user can send from the MSR can be emulated on the automation interface, using the same codes as the RS232 interface found on products such as the G or 800 series. A full list of these codes can be retrieved by sending:

```
#MSR help
```

Some examples:

```
#MSR CD
```

Will select the CD source, if it is enabled. The command could reply:

```
*ACK
```

or

```
*NAK "Source not enabled"
```

Following the ACK, the product will send out an unsolicited message to every connected interface:

```
!SRC Source:"0" Legend:"CD" Input:"Digital" Mute:"Demute" Volume:"65"
```

Sending a standby command:

```
#MSR SB
```

Will replay with:

```
*ACK
```

And follow, to all connections, with:

```
!OFF
```

```
#MSR VP
```

Increases the volume (unless in standby). The command will reply:

```
*ACK
```

And lead to:

```
!VMU Mute:"Demute" Volume:"66"
```

The volume will not be changed if the product is in standby, therefore there will not be an unsolicited message.

Commands for source devices, such as:

```
#MSR PL
```

for 'play' will be acknowledged, but any action will depend on the status and configuration of that source.

Command rate.

Commands from a real MSR are separated in time by 114ms, this is significant for the user's experience of the system. There is a two-stage timer built into the interface in order to both enforce this maximum rate and also to assist the smooth running of the GUI. Commands received within 100ms will be rejected:

```
*ERR "Command sent too soon"
```

Commands received between 100ms and 114ms will be delayed (silently) until the 114ms timer has expired, at which point the message will be acknowledged:

```
*ACK
```

Please note that any messages intended to be delivered in this window will still be subject to timing variations of the LAN.

Autosetup.

Within a Meridian system it is normal to designate a single unit as the one that receives commands from an MSR, this unit is called the 'Controller'. This device can either be manually configured, by configuring all the units in the system to 'Not Controller' and 'Controller' as appropriate, or by leaving them in their factory default 'Auto' state and pressing 'Clear' on the MSR (when the system is in standby). This is **not** required for the MSR emulation to function on this interface. If a G12 (or equivalent) were to be plugged into the 218, then the UI might see a message such as:

```
!TMP Display:"Controller" Period:"3"
```

or

```
!TMP Display:"Not Controller" Period:"3"
```

Menu control.

Parameters that a user might want to change at runtime are made available in a menu system, navigated by the MSR up/down/left/right keys.

- `#MSR MP`
- `#MSR MM`
- `#MSR ML`
- `#MSR MR`

Left and right are used to select the previous/next menu, up and down are used to change the menu value. Menus may be enabled or disabled, based on the current configuration. Two ways of driving this system are presented on the automation interface, one allows the designer to create something like a normal Meridian front panel, overlaying the current source & volume with a temporary menu display, the other allows a control panel showing all currently valid menus and their values, with individual controls for increasing/decreasing values.

All menu entries are described by a <Menu> <Value> pair, where each element is presented as Name:String. e.g.

```
Menu:”Treble” Value:”+0.0dB”
```

The menu name and its value are both presented as text strings ready for the user, the value including any units as required. Two different messages are sent from the product when navigating via the MSR – the result of left/right changes are a focus change message:

```
!MFC Menu:”Treble” Value:”+0.0dB”
```

The result of an up/down is a value change message:

```
!MVC Menu:”Treble” Value:”+0.5dB”
```

For the correct user experience the menu display should be shown for 3 seconds, and then revert to the underlying source display.

Note that menu left/right only changes the menu focus within those 3 seconds while the menu display is (expected to be) shown, but menu up/down will always change the value of the menu in focus. This is to allow the user to review, without any changes, the current menu focus and its value.

For the alternative layout, where the designer wants to create a control panel that draws all menus at once there are three commands, one to read and two to change the values.

```
?MGV
```

Replies with all the menus that could be shown as a <Menu> <Value> <Show> triplet, e.g.:

```
*MGV Menu:"Treble" Value:"+0.5dB" Show:"Yes" Menu:"Bass" Value:"+0.0dB" Show:"Yes"
Menu:"Balance" Value:"<0>" Show:"Yes" Menu:"Axis" Value:"-1" Show:"Yes" Menu:"Phase"
Value:"+ve" Show:"Yes" Menu:"EBA" Value:"Yes" Show:"No" Menu:"Location" Value:"Free" Show:"No"
Menu:"EBC" Value:"Off" Show:"No" Menu:"Sub Mode" Value:"Music" Show:"No" Menu:"Sub Filter"
Value:"Off" Show:"No" Menu:"Sub Gain" Value:"+0dB" Show:"No" Menu:"Loudness" Value:"Off"
Show:"No"
```

The ‘show’ element is required because some menus can be disabled by another setting, for example, the EBA menu is not relevant when SpeakerLink is disabled, so will be marked as `Show:”No”`. (Note that when navigating the menus via the MSR interface only valid menus will be shown.)

To change any menu at random, use the command MVP or MVM followed by the menu name, e.g.

```
#MVP Treble
```

Is used to increase the Treble value, and

```
#MVM Bass
```

Is used to decrease the Bass value.

Note that some menus will ‘wrap’ from largest to smallest value and vice versa (e.g. phase), and some will not.

If the menu is not allowed (i.e. ‘Show’ is “No”) then the product will respond:

```
*NAK “Menu not allowed”
```

It will send

```
*ERR “Unknown menu”
```

if there is an error in matching the menu name.

Otherwise it will send `*ACK` and invoke an unsolicited message, e.g.:

```
!MVC Menu:”Bass” Value:”-0.5dB”
```

Note that the menu system is not available unless a source is selected. Trying to access the menus directly while in other states will return:

```
*NAK ”No source selected”
```

There are occasions where the menu system is reset or changed, such as source selection or configuration change. When this occurs, the product does not resend all the menu data, it sends:

```
!MRE
```

to indicate that the menu system has been reset. It is up to the developer whether to subsequently ask for the new menu data using `$MGV` or not. If the developer merely wants to find out which menu is now in focus then sending:

```
?MGF
```

will return the current menu focus:

```
*MGF Menu:”Treble” Value:”+3.0dB”
```

Default menu contents.

When a product is turned on it loads the menus with data retrieved from non-volatile storage. When first unpacked this will be the same as the factory reset. The product will not change these defaults when the menus are adjusted during normal operation, but the user may want to change the default away from the factory settings. There are various ways to ‘Store’ the user’s preference in the non-volatile memory, including two via this automation interface. The first is a two-step process based on MSR emulation of the ‘Store’ key.

```
$MSR SR
```
Begins the process, and will reply:
```
*ACK
```
Followed by an unsolicited:
```
!TMP Display:‟Store Menus?” Period:”3”
```
Which tells the UI to draw the given text for three seconds. Sending another `$MSR SR` within 3 seconds will save the settings, send `*ACK` and reply:
```
!TMP Display:‟Menus stored” Period:”3”
```
Note that the store process could be started from another interface, or even a real MSR.

The second method allows a designer to add a store button for this express purpose. Sending
```
#MST
```
Will store all menu data immediately and reply:
```
*ACK
```
It will then follow, as in other methods, with:
```
*TMP Display:‟Menus Stored” Period:”3”
```

The menus can be returned to their factory settings using the ‘Clear’ key on the MSR (when not in standby):
```
#MSR CL
```
Will reply with an `*ACK` and will invoke an unsolicited message:
```
!TMP Display:‟Clear Menus?” Period:”3”
```
Sending another `$MSR CL` within 3 seconds will clear the settings, send `*ACK` and reply:
```
!TMP Display:‟Menus cleared” Period:”3”
```
It is also possible to do this in one step by sending:
```
#MCL
```
Which will reply immediately with `*ACK` and follow with:
```
*TMP Display:‟Menus cleared” Period:”3”
```

System control – additional non-MSR features.

There are some additional commands available that are not possible from the MSR.

Direct volume control, for implementing a volume control of the developer’s choice. The command:
```
#SVN 45
```
will reply:
```
*ACK
```
and set the volume to 45 (if not in standby), informing all connected devices:
```
!VMU Mute:”Demute” Volume:”45”
```

Logical source selection.

A ‘source’ in the Meridian world is a collection of settings defining an audio input selection, a legend, communication settings and some other defaults. The root of this is the ‘logical’ source, a number from 0 to 11. The default settings for all of these sources are set such that a system in its factory default will present a unified interface to the user. This default extends to the text drawn on the MSR and is used in the literature to describe the products. Each of these sources can be customised to suit the installation however, to the extent that they bear little resemblance to the defaults; with the automation interface this can also be extended to the UI. We include the logical source number to help the developer integrate their design into the Meridian functionality.

This includes selecting a source by number, rather than needing to know its name, or which button it might be on an MSR. For example, to select the first source:
```
#SRC 0
```
will reply (if the source is enabled):
```
*ACK
```
and provoke an unsolicited message to all devices:
```
!SRC Source:”0” Legend:”CD” Input:”Digital” Mute:”Demute” Volume:”65”
```

Some products in the Meridian range have a “Source” key that selects the next valid source (or may have a remote control that has a “Source” key). Sending `#SRC` without a number will either:

- Bring the unit out of standby on the last used source
- Bring the unit out of standby on the ‘Startup Source’ (if the unit has just been powered up)
- Select the next logical source that is enabled

The product will reply with `*ACK` and send an unsolicited message to all devices:

```
!SRC Source:"2" Legend:"SLS  " Input:"Sooloos" Mute:"Demute" Volume:"65"
```

System information.

Source legends.

Where products have customisable legends, they can be discovered by sending:

```
?GSL
```

The product will reply:

```
*GSL Source:"0" Legend:"CD" Enabled:"Yes" Source:"1" Legend:"Radio" Enabled:"Yes" Source:"2"
Legend:"SLS" Enabled:"Yes" Source:"3" Legend:"TV" Enabled:"Yes" Source:"4" Legend:"Tape"
Enabled:"Yes" Source:"5" Legend:"Sat" Enabled:"Yes" Source:"6" Legend:"Disc" Enabled:"Yes"
Source:"7" Legend:"Cable" Enabled:"Yes" Source:"8" Legend:"DVD" Enabled:"Yes" Source:"9"
Legend:"PVR" Enabled:"Yes" Source:"10" Legend:"USB" Enabled:"Yes" Source:"11" Legend:"Game"
Enabled:"Yes"
```

Which covers:

- The logical source number
- The user configurable legend
- Whether the source is currently enabled or not

When the user changes the source legends (e.g. via webpage setup) the product will send an unsolicited message:

```
“!SLC”
```

If the UI wishes to discover the new legends then it should send “`?GSL`” again.

Source status.

To get the current source status, send:

```
?PGS
```

The content of the reply will be:

```
*PGS Status:"On" Source:"2" Legend:"SLS  " Input:"Sooloos" Mute:"Demute" Volume:"65"
```

Which covers:

- The standby state
- The logical source number
- The source legend
- Which audio input is selected
- Whether the product is in mute or not
- The current volume

Audio stream status.

The product will provide information about the currently received audio stream whenever it changes, as well as on request from the GUI.

To request the current status, send:

```
?AGS
```

The product will answer with:

```
*AGS Format:"PCM" SampleRate:"44100Hz" Error:"None" Audio:"Yes"
```

During normal operation the product will send unsolicited messages when the stream changes, such as:

```
!ASC Format:"PCM" SampleRate:"48000Hz" Error:"None" Audio:"No"
```

Sooloos identification.

The product identification includes the Sooloos Zone name; if the user changes the name then the endpoint will inform the UI with a separate message, like this:

```
!ZNC ZoneName:”Dining Room”
```

Closing the connection.

Devices may close the TCP connection at any time without ill effects. The product will normally send a short message before closing any or all connections, possibly including a string, such as:

```
!ARV “PNG timeout”
```

Appendix.

Here is a list of message descriptors that may exist on this interface. Lists of information parameters will be product dependant, to an extent, so the presence of a parameter is not guaranteed. Nor is the order in which they appear guaranteed.
Within this appendix the lists include the information from a 218, as of the current release.

Commands.

```
help            Help (special case for development purposes)
        <all the automation commands>
#PNG            Ping (can originate from either UI or product)
        *ACK
#DEV            Select Development mode (stops ping and auto-disconnect)
        *ACK
#MSR xx         MSR emulation
        *ACK, *NAK or *ERR
#MSR help       List MSR commands
        < all the MSR commands>
#MVP            Menu Value Plus
        *ACK, *NAK or *ERR
#MVM            Menu Value Minus
        *ACK, *NAK or *ERR
#MST            Menus Store
        *ACK, *NAK or *ERR
#MCL            Menus Clear
        *ACK, *NAK or *ERR
#SVN xx         Set Volume number
        *ACK, *NAK or *ERR
#SRC x          Select Logical Source
        *ACK, *NAK or *ERR
```

Queries.

```
?PID            Product Identification
        *PID <Product> <SerialNumber> <VersionNumber> <ZoneName>
?MGV            Menu Get Value
        *MGV <Menu> <Value> <Show>
?MGF            Menu Get Focus
        *MGF <Menu> <Value>
?PGS            Product Get Status
        *PGS <Standby> <Source> <Legend> <Input> <Mute> <Volume>
?AGS            Audio Get Status
        *AGS <Format> <SampleRate> <Error> <Audio>
?GSL            Get Source Legends
        *GSL <Source> <Legend> <Enabled>
```

Unsolicited messages.

```
!PID            Product Identification
        <Product> <SerialNumber> <VersionNumber> <ZoneName>
!SRC            New source
        <Source> <Legend> <Input> < Mute> <Volume>
!OFF            Standby
        <no data>
!VMU            New Volume/Mute
        <Mute> < Volume>
!MFC            Menu Focus Changed
        <Menu> <Value>
!MVC            Menu Value Changed
```

<Menu> < Value>

!MRE Menu System Reset

<no data>

!TMP Draw Temporary Display

<Display> < Period>

!ASC Audio Status Changed

<Format> <SampleRate> <Error> <Audio>

!SLC Source Legends Changed

<no data>

!ZNC Zone Name Changed

<ZoneName>

!ARV Au Revoir

Parameters.
These are the parameters in the current version of 218.

<Product>
The product name, e.g. “218”

<SerialNumber>
For identification, e.g. “100042”

<VersionNumber>
The software version number, e.g. “1.0.170”

<ZoneName>
The Sooloos zone name, e.g. “218 #0024c5000368”

<Menu>
The menu name, as a string, e.g. “Bass”, “Treble”

<Value>
The menu value, as a string, e.g. “+2.5dB”

<Show>
Whether the menu is currently valid or not, can be “Yes” or “No”

<Standby>
Can be “Standby” or “On”

<Source>
The logical source number, from “0” to “11”

<Legend>
The customer facing legend, e.g. “CD” or “Game”

<Enabled>
Whether a source is enabled or not, can be “Yes” or “No”

<Input>
The audio input, e.g. “Digital”, “Sooloos”

<Mute>
Can be “Mute” or “Demute”

<Volume>
The Meridian volume number, on a scale from “1” to “99”

<Format>
The audio format, e.g. “PCM”

<SampleRate>
The digital audio sample rate, e.g. “48000Hz”

<Error>
The digital audio error status, e.g. “None”, “No Lock”

<Audio>
Indicates “Audio” or “Data”

<Display>
A text string to draw on the UI, e.g. “Controller”

<Period>
How long to draw the display for, in seconds, e.g. “3”