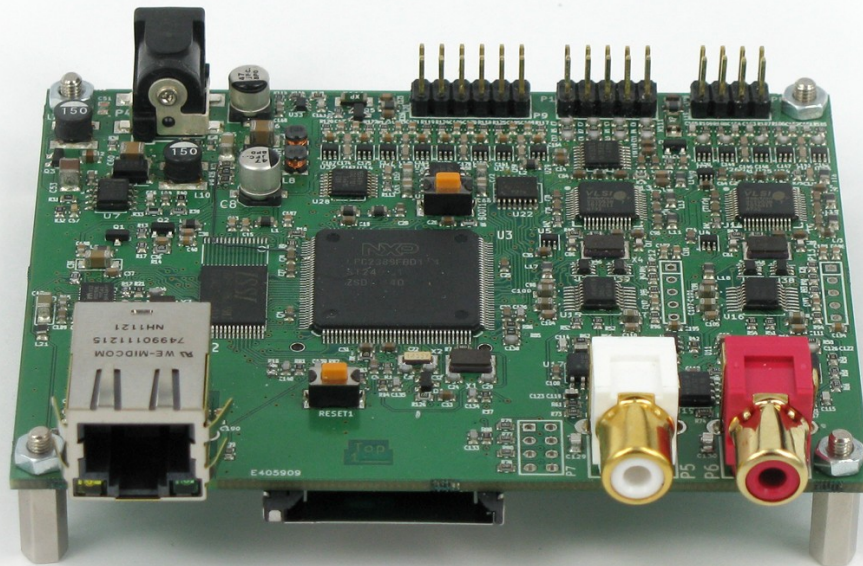


Starling H0440



Programmable Audio & Network Controller for Professional Playback



User Guide

ITB CompuPhase
Eerste Industriestraat 19-21
1401 VL Bussum, The Netherlands

Tel.: +31 (0)35 6939261
Fax.: +31 (0)35 6939293
info@compuphase.com
www.compuphase.com



2012-10-30

Contents

Preliminary.....	2
Anti-static handling precautions.....	2
Power supply voltage range.....	2
Legal disclaimer.....	2
Overview of the Starling, model H0440.....	3
Connecting the Starling, model H0440.....	3
SD/MMC card.....	3
Power connector.....	4
Audio outputs.....	4
Network connector.....	4
Battery connector.....	4
RS232 (serial communication).....	4
SPI (serial communication).....	5
General purpose I/O.....	5
I2S audio output.....	6
Mechanical montage.....	6
Development environment and user-scripts.....	6
Installing the development environment.....	6
Using the ready-made scripts.....	7
Writing scripts.....	7
Audio tracks.....	8
Track duration & seeking to a position.....	8
APE Tags, ID3 Tags and synchronized events.....	8
Adding “synchronized lyrics” to an MP3 file.....	9
Cross-fading.....	9
Encrypted MP3 tracks.....	9
Troubleshooting.....	10
Specifications.....	11

Preliminary

Anti-static handling precautions

Please observe anti-static handling precautions when handling the device, as it contains components that are sensitive to static discharges.

Power supply voltage range

Please verify the power supply voltage to be within the range that the device supports, and that the polarity is correct. See page 11 for the specifications and page 4 for the power connector. Voltages outside the appropriate range may cause malfunction. Voltages above the high limit or with inverted polarity may damage the device.

Legal disclaimer

ITB CompuPhase shall not be liable for the incidental or consequential losses or damage to tangible property, injury or death of a person in connection with the use of this device. Although the examples in this guide have been tested with care, they may contain errors and they are not guaranteed for any particular purpose.

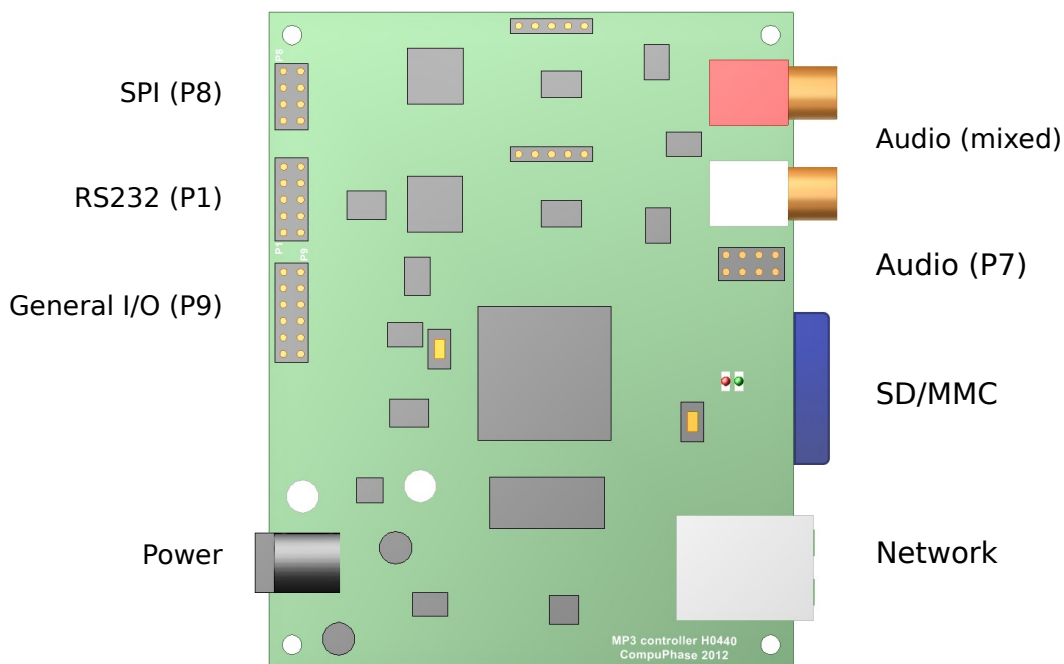
Overview of the Starling, model H0440

The Starling is a compact and solid state general purpose controller with a double digital-audio decoder and network interface, that runs under the control of a scripting language. The audio fragments and the script that controls the operation and behaviour of the audio player are stored on an SD/MMC card. A network interface includes a built-in TCP/IP stack. Auxiliary devices, such as an LCD, (electronic) switches or data terminals may be attached through an 8-bit general purpose I/O bus, an SPI bus, or a dual RS232 line. The audio outputs are present on a pin header for both audio decoders separately, as well as mixed on two RCA outputs.

This booklet introduces the scripting capabilities of the Starling audio controller. For technical details and a programming reference, please see the documentation that is provided in Adobe Acrobat format on the CD-ROM that comes with the product.

The script controls the operation of the Starling controller and the peripherals. The script must be written in the PAWN language and be stored (in compiled form) on the SD/MMC card. In absence of a script, the Starling plays all audio files on the SD/MMC card. To react on switches or I/O pins and to communicate over the network interface, or drive other peripherals, a PAWN script is required.

Connecting the Starling, model H0440



SD/MMC card

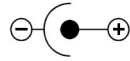
The SD/MMC card connector is of the push/push type. To insert a card, push it in until it clicks and then release it. To remove a card, push the card again until it clicks and release it; it will then be ejected.

The memory card connector accepts SD cards, SD/HC cards and MMC cards. The memory cards must use the standard FAT file system (both FAT16 and FAT32 are supported). The firmware supports subdirectories, encrypted volumes (partitions) and encrypted audio tracks, using strong encryption with a 128-bit key.

The SD/MMC card must contain all audio tracks. Typically, it will also contain a programmed "script" that determines how the device responds to various inputs—for example, by playing different audio tracks for each switch that is pressed.

Power connector

The power connector is a standard low voltage power connector with an outer diameter of 5.5 mm and a pin size of 2.1 mm. The pin is the plus pole.

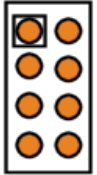


The operating voltage must be in the range of +7.5 V DC to +28 V DC.

Audio outputs

Two RCA connectors provide analogue stereo output. The red connector is for the right channel. RCA connectors are also called “phono” connectors, or “cinch” connectors.

On model H0440, the output on the RCA connectors is the mixed stereo signal of both decoders. If you need the audio output of both decoders separately, you need to use the pin header P7. The pin lay-out of P7 is:

	1	Left channel, decoder 1	2	Left channel, mixed
	3	Right channel, decoder 1	4	Ground
	5	Left channel, decoder 2	6	Right channel, mixed
	7	Right channel, decoder 2	8	Ground

Network connector

The Starling can be connected to an Ethernet network using a standard CAT5 cable with an RJ-45 connector. The network connection supports 100 Mb/s, full duplex and auto-negotiation.

A few standard protocols of the TCP/IP protocol suite are built-in, other protocols can be implemented with PAWN scripts. Please see the Reference Guide for details.

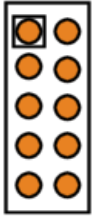
Battery connector

The battery connector is on the reverse side of the Starling board. Use a common 3 V CR2032 battery. The battery is used for keeping the time while the board is powered off and for powering a small region of memory, so that settings are remembered when the power is off.

If you do not need clock functions and/or storing settings in the device's memory, the battery is optional. Note that a script can also store device settings on the SD/MMC card.

RS232 (serial communication)

Two serial ports are provided on a 5×2 pin header marked P1. The signals on these pins have the standard voltage RS232 levels for serial communication. The pin lay-out has been chosen so that the RxD and TxD pins for the first port match with an IDC D-subminiature socket pressed on ribbon wire, were the RxD and TxD lines for the second port get mapped to the DTR and DSR lines.

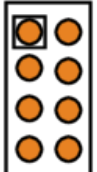
	1	N.C. not connected	2	TxD2 Transmit, port 2 / DTR
	3	TxD1 Transmit, port 1	4	not connected
	5	RxD1 Receive, port 1	6	not connected
	7	RxD2 Receive, port 2 / DSR	8	not connected
	9	Ground	10	Ground

All common Baud rates are supported (1200, 2400, 4800, 9600, 19200, 28800, 38400, 57600 or 115200); special Baud rates, such as 31250 for MIDI, are supported as well. The number of data bits and stop bits, and the parity is configurable in the user program.

The RS232 interface supports the XON/XOFF protocol (software handshaking), but no hardware handshaking protocols. The use of a handshaking protocol is configurable.

SPI (serial communication)

The 4×2 pin header marked P8 provides a general purpose SPI communication port with two “slave select” signals. In addition, an output power pin is provided on this connector; this power output pin must be enabled by the script, it is disabled by default.

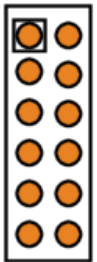
	1	Master-out/Slave-in (MOSI)	2	Slave select 1
	3	Slave select 0	4	Master-in/Slave-out (MISO)
	5	SPI clock (max. 10 MHz)	6	Ground
	7	Power output (400 mA max.)	8	Ground

The voltage levels of the SPI signal pins is the configurable external voltage, which is 3.3 V by default. The maximum SPI clock frequency is 10 MHz.

The voltage of the power output pin (if enabled) is the same of that of the I/O pins. The power output has a self-resetting fuse for short-circuit protection. The maximum current that can be drawn from the power pin is 400 mA.

General purpose I/O

The 6×2 pin header marked P9 has 8 general purpose digital pins for communicating with other devices that can be configured as input or output, and one analogue output-only pin. In addition, an output power pin is provided on this connector; this power output pin must be enabled by the script, it is disabled by default.

	1	IO0 general purpose I/O	2	IO1 general purpose I/O
	3	IO2 general purpose I/O	4	IO3 general purpose I/O
	5	IO4 general purpose I/O	6	IO5 general purpose I/O
	7	IO6 general purpose I/O	8	IO7 general purpose I/O
	9	Analogue out	10	Ground
	11	Power output (400 mA max.)	12	Ground

The I/O pins are EMC and ESD filtered, and slew-rate limited to 1 MHz. The analogue output pin is limited to 100 kHz.

When the I/O pins are configured as inputs, the pins are 5 V-tolerant. Each pin has a pull-up resistor of 10 kΩ.

When the I/O pins are configured as outputs, the high-level voltage for the pins is

configurable between 0 V to 5 V (on start-up, it is 3.3 V). The analogue output pin has a range from 0 V to the I/O voltage.

Each digital I/O output pin has a series resistor of 120Ω for current limiting and short-circuit protection. The output pins can directly drive a LED (or a similar load, such as an opto-coupler), usually without requiring an external current-limiting resistor. The current limit depends on the forward voltage of the LED and the configured voltage of the output pins.

The analogue output has a series resistor of 220Ω for current limiting and short-circuit protection.

The power output has a self-resetting fuse for short-circuit protection. The maximum current that can be drawn from the power pin is 400 mA. If enabled, it has the same voltage as the I/O pins (the power pin is disabled by default).

I²S audio output

In addition to analogue output on the RCA connectors, the Starling model H0440 can provide the audio in I²S format, for an external DAC. Please see the datasheet for details on the I²S connectors and specifications.

Mechanical montage

Four mounting holes with a diameter of 3.1 mm (suitable for M3 screws or 1/8" screws) are spaced at 93 mm horizontally and 73 mm vertically. Ensure at least 6 mm of clearance below the PCB. See the datasheet for details and drawings.

The mounting holes are electrically isolated from the digital and analogue ground of the device.

Development environment and user-scripts

A full development environment for user-scripts is freely available on the CompuPhase web site. It is also included on the CD-ROM that comes with the product. The development environment contains:

- Several ready-to-run and fully documented scripts that can be configured through an "INI" file and a "playlist" file. No programming is required to run these scripts.
- The PAWN development tools: the compiler, the debugger and configuration files.
- An assortment of example scripts in source code form, to get you starting with programming the Starling audio controller.
- Auxiliary utilities, such as a serial monitor program.
- Extensive documentation in Adobe Acrobat format: the PAWN language manual, the Starling programming reference, and the datasheet with detailed electrical and mechanical information.
- Example MP3 tracks, for testing purposes.

Installing the development environment

On Microsoft Windows, run the application called "Setup" on the CD-ROM or the self-installing setup downloaded from the CompuPhase web site. Then follow the directions of the installation utility. The installer allows you to choose various optional components for installation.

The PAWN compiler is also available for Linux, on the CD-ROM and on-line. For installation under Linux, please refer to the documentation of the development environment.

Using the ready-made scripts

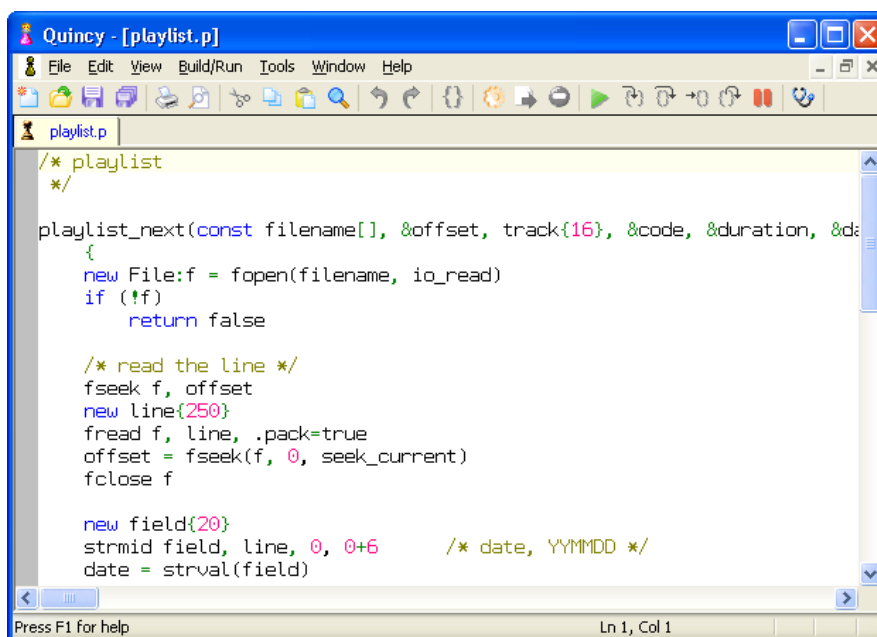
You may first want to try whether one of the standard, ready-made scripts fits your requirements. The ready-made scripts are documented in HTML pages, and links to these pages appear in the “Program Files” menu (after installing the development environment).

The ready-made scripts read the audio tracks to play from a playlist that conforms to the WinAmp M3U specification. The scripts may optionally be configured through a standard text file in “INI” file format. All configuration options are described in the aforementioned HTML pages.

The ready-made scripts may also serve as more advanced examples for coding your own scripts. Each of the ready-made scripts comes with the full and commented source code. To use the scripts, no programming or build step is required, because the scripts come in a form that is directly usable by the Starling controller (“pre-compiled”).

Writing scripts

Scripts for the Starling are in the PAWN language. To type in and build the scripts, you use the “Quincy IDE” that is part of the PAWN development tools, or any standard text editor. When saving the script's source code, we suggest the extension “.p” or “.pawn”.



The Quincy IDE

A minimal script, that plays one file on power-on and then stops, is:

```
@reset()
{
    play "track1.mp3"
}
```

The Starling controller does not interpret the source code of the script directly, but it executes a compiled form of the script. You can compile directly from the Quincy IDE, or use the separate PAWN compiler in a console window or “DOS box”.

The PAWN compiler generates a binary file with the extension “.amx” from the input file. The next steps are to copy this “.amx” file to the SD/MMC card and to insert this card into the Starling. In the Quincy IDE you can also adjust the output path so that the source file gets compiled directly onto the SD/MMC card.

Typically, a script must do a lot more than play just a single file. At the very least, it should allow to re-start playing one or more audio files without needing to power-cycle the device. More commonly, the script must react on (user) input in the form of switches or digital levels from sensors. Please look at to the example scripts for various interfacing techniques.

As stated above, compiling the script creates a new file with the extension “.amx”. This file needs to be copied to the SD/MMC card. It can be in one of two locations:

- a directory called “system” on the SD/MMC card.
- the root directory of the SD/MMC card

The Starling first checks the root directory and, if no compiled script is found in the root, then checks for a script in the “system” directory.

If you have multiple scripts on a card, the compiler script that must run on power-up (or after reset) should be called “startup.amx”. This script can also be in either the root directory or the “system” directory.

Audio tracks

The Starling supports the following audio file formats:

- MPEG version 1 layer 3 (“MP3”), as well as MPEG version 2 and 2.5 layer 3;
- Ogg Vorbis;
- Uncompressed “WAV” files (RIFF header), 8-bit and 16-bit;
- ADPCM files with a RIFF header;
- Optionally HE-AAC (AAC LC with SBR), level 3; this option requires a license.

Track duration & seeking to a position

Within the MP3 file format, there are the encoding varieties “constant bit rate” (CBR), “variable bit rate” (VBR) and “average bit rate” (ABR). VBR and ABR files must have a “Xing” header to support “direct jumps” to a specific position in the MP3 track. Without this header, jumps to a position may be inaccurate for VBR and ABR files and the reported track duration may be inaccurate as well. Reported durations and direct jumps are always accurate for CBR files.

Reported durations and direct jumps are also accurate for uncompressed PCM files (“WAV” files).

Ogg Vorbis files are always “variable bit rate” and no meta-header with information on the full file is defined. The Starling therefore relies on the APE tag for the duration of the track. Jumps to a position in an Ogg Vorbis track may not be accurate.

APE Tags, ID3 Tags and synchronized events

The scripting language gives the programmer access to the information in an APE or ID3 tag (version 2), with information on the artist, album & track, full title, copyright and cue/segue times.

The supported frames in the APE and ID3 tags are:

ID3 frame	APE frame	Description
TIT2	Title	Song title

TPE1	Artist	Performing artist
TALB	Album	Album title
COMM	Comment	Comment
TCOP	Copyright	Copyright
TSRC	ISRC	ISRC data, a code to identify sound/music recordings
TLEN	Duration	Duration of the audio track in milliseconds
SYLT		Synchronized lyrics
	Cue	Cue time in milliseconds (silence at the start of the track)
	Segue	Segue time in milliseconds (silence at the end of the track)

The APE and ID3 tags both contain similar data, but both also have fields that are absent from the other. The ID3 tag supports events to be synchronized to time-stamps inside the track (“synchronized lyrics”) and the APE tag provides fields that are useful for the purpose of cross-fading.

Adding “synchronized lyrics” to an MP3 file

In addition to the standard “authoring” information, the Starling supports the optional “synchronized lyrics” frame in the ID3 tag. The contents in this frame will cause scriptable events to be “fired” on the time stamps that are attached to each line of lyrics. Although the synchronized lyrics frame was originally designed for purposes of karaoke and sub-titling songs, another use would be to tag time-stamped cues or commands at specific positions in a song or audio fragment.

The first step in making a script react on synchronized events in an MP3 file is to attach an ID3 tag with the “SYLT” frame to that file. A suitable SYLT editor is freely available from the CompuPhase web site.

Synchronized lyrics is a set of plain text strings where each string has a time-stamp, in milliseconds from the start of the audio track. After having added the public function “@synch()” to your script, this function will be invoked (or called) at the precise time-stamps of each string and with the text of the string as a parameter.

For purposes of Karaoke, the text strings will indeed be the lyrics of the piece. For other applications, the time-stamped text may consist of strings like “Light #1 off”, or “Next slide”. It is up to your script to interpret the text strings that it receives.

Cross-fading

The Starling model H0440 has two audio decoders and can read and play two tracks at the same time. One typical use of this is to cross-fade two tracks so that they follow each other seamlessly.

When a track is recorded, however, there is often silence at the start and the end of the track. These silent periods should be skipped when cross-fading one track into the other. The APE tag can carry the required information in its “Cue” and “Segue” frames.

There are a few programs that can analyze audio tracks and write/update the cue and segue settings automatically. We advice “TrackTool” from StationPlaylist.com.

Encrypted MP3 tracks

The Starling supports MP3 tracks that are encrypted with 128-bit device key, plus an optional 32-bit user key. The device key is hard-coded in the device; only Starling units with the correct key can play back the encrypted tracks. When ordering a series of

MP3 players, a customer may request a unique device key for that series. In addition to the key, the customer also receives a programs for encrypting MP3 tracks.

ID3 tag information (version 2) is preserved and non-encrypted. However, the ID3 tag should not be modified after encrypting the file.

Troubleshooting

If the device does not run correctly, please first try these few tests.

As a first test, remove power, disconnect all cables and devices attached to the Starling and remove the SD/MMC card. Then (re-)connect the power and observe the two LEDs.

- The green LED should stay on. If it remains off, please check the power and the polarity of the power. See page 11 for the voltage level and page 4 for the polarity. If the green LED blinks, there is a problem with the firmware, please contact us for a replacement or option to restore the firmware.
- The red LED should blink roughly twice per second.

After passing the above test, create an SD/MMC card with only MP3 tracks in the root directory (delete all script files and other files with a “.amx” extension). It is best to format the card first and then put a few MP3 tracks on it (at least, verify that you can read this SD/MMC card on a PC and that it uses the FAT16 or FAT32 file formats).

- You can press the switch labeled “FUNC” on the Starling to skip to the next track. On the Starling model H0440, this switch also toggles the decoder and the green LED. That is, after pressing the FUNC switch, the next track will play on the secondary decoder and the green LED goes off. Another press on the FUNC switch toggles back to the primary decoder and turns the green LED back on.
- After inserting the card, audio should start playing. The red LED indicates read access to the SD/MMC card, it typically blinks so fast that it appears full on, but dimmed. The green LED stays full on.

If these tests pass, you can now copy one of the pre-compiled scripts onto the SD/MMC card. It is suggested that you first copy the “Play playlist” example, which is a complete example of a simple script that just plays tracks in the order that they are specified in the playlist file.

For diagnosing network problems, please make sure that you have a DHCP server active in your LAN, and then run one of the pre-compiled network scripts.

Specifications

General

Operating voltage.....+7.5 V DC to +28 V DC.
Power consumption.....1.35 Watt to 1.6 Watt, depending on peripheral use.
Dimensions.....100 mm × 80 mm.
Weight.....0.070kg.

Audio

Supported formats.....MPEG layer III (“MP3”); Ogg Vorbis; 8-bit & 16-bit RIFF WAV; MIDI format 0; AAC (AAC support requires a license).
Frequency response.....20 Hz to 20 kHz, ±0.5dB.
Dynamic range.....90 dB (typical).
Distortion+Noise.....THD+N < 0.03% at 1 kHz +0 dBu.
Noise.....SNR 93 dB at 1kHz +0 dBu (typical).
Output impedance.....100Ω.
Max. output level.....+6 dBu (European studio level).

Integrated peripherals

Temperature sensor.....Measures -40 °C to +125 °C; typical accuracy 0.5 °C.
Real-time clock.....Inaccuracy < ±1.7 seconds per 24 hours.
Pseudo-random numbers...Cycle length 2^{32} , initialized with true a random seed.
RS232 interfaces.....Two interfaces conforming to 3-wire RS232 ports; standard signal levels.
SPI interface.....Standard SPI interface, with programmable voltage level and two slave select lines
Network.....10/100 Mbit/s, full/half-duplex with auto-negotiation.
Digital I/O.....8 general-purpose I/O pins, with programmable voltage level (2.5 V to 5.0 V); max. 50 mA source or sink; all pins are ESD/EMC filtered.
Analogue out.....1 software-controlled analogue output pin.

Operating conditions

Temperature.....Designed for -40 °C to +85 °C.
Humidity.....5% to 95% non-condensing.
Vibration.....Full solid-state device, no moving parts.

Conformity

EMC (emission/immunity). Compliant with EU Directive 2004/108/EC: EN 55022 and EN 55024 + A1 (2001) + A2 (2003).
Electrical safety.....Compliant with EU Directive 2006/95/EC: EN 60950-21
RoHS.....Compliant with EU Directive 2002/95/EC.