

With the TuneSOne audio module, you can easily add audio and voice output to your project.

## Datasheet

### Datasheet Version History

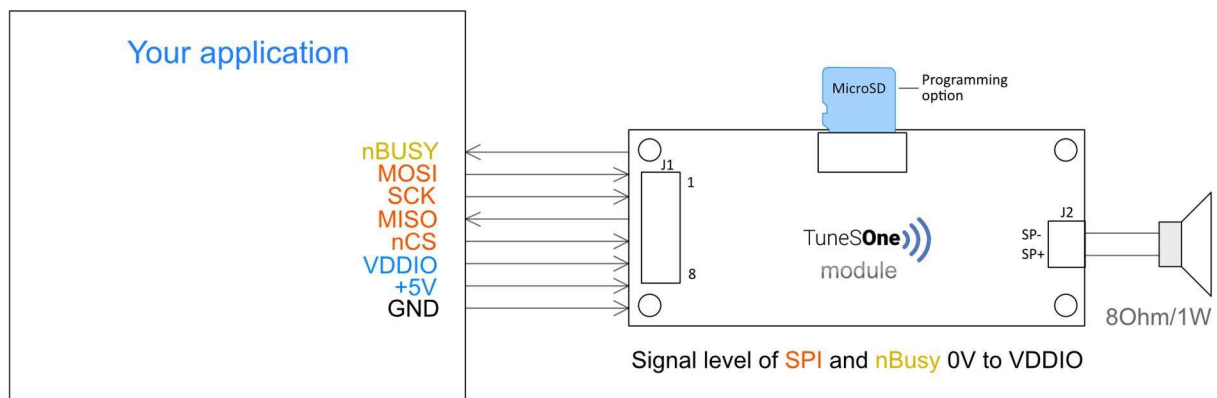
| Version | Date            | Description  |
|---------|-----------------|--|
| 1.0     | July 31, 2025   | Initial Version  |
| 1.1     | August 12, 2025 | Further information added to the chapter "12 Create and program project" |

## Table of contents

|      |   |    |
|------|---|----|
| 1    | Description .....                       | 3  |
| 2    | Absolute Maximum Ratings .....          | 3  |
| 3    | Recommended Operation Conditions .....  | 4  |
| 4    | Electrical Characteristics .....        | 4  |
| 5    | Audio Amplifier .....                   | 5  |
| 6    | Audio Decoding.....                     | 5  |
| 7    | Flash Memory .....                      | 6  |
| 8    | Control Interface and Protocol .....    | 6  |
| 8.1  | Command Play Audio File .....           | 9  |
| 8.2  | Command Play Audio File Language .....  | 10 |
| 8.3  | Command Stop .....                      | 11 |
| 8.4  | Command Play .....                      | 12 |
| 8.5  | Command Pause .....                     | 12 |
| 8.6  | Command Set Volume .....                | 12 |
| 8.7  | Command Read Status.....                | 13 |
| 8.8  | Command Read Firmware Version .....     | 14 |
| 8.9  | Command Read Flash File Version.....    | 14 |
| 8.10 | Command Read Flash File Date Time ..... | 15 |
| 9    | Status LED .....                        | 15 |
| 10   | Dimensions.....                         | 16 |
| 11   | Interface.....                          | 16 |
| 12   | Create and program a project.....       | 17 |
| 13   | Appendix .....                          | 20 |
| 13.1 | Example SPI data transfer.....          | 20 |
| 14   | Legal notice .....                      | 21 |

# 1 Description

The audio decoder module supports MP3 MPEG 1 and 2 Audio Layer III formats with constant and variable bit rates up to a data rate of 320 kBit/s. The integrated Class D audio amplifier can drive an 8-ohm speaker with an output power of 1 W. It features a circuit for reducing clicks and pops, as well as active emission limiting, slew rate limiting, and overshoot control to significantly reduce EMI. The module has an SPI interface with a user-friendly protocol and a level translation function (VDDIO 3.3 V to 5 V). The audio files are stored in a flash memory with 32 MB storage capacity and are located in a robust file container. The audio file container is programmed into the flash memory via a microSD card. The audio file container is created using Windows software. The source audio files for the project are stored in a project directory and managed via Windows Explorer. This makes it easy to port, modify, expand, and duplicate the project.



Block diagram

# 2 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)<sup>(1)</sup>

|                       |  |  | MIN  | MAX                  | UNIT |
|-----------------------|--|--|------|----------------------|------|
| <b>V<sub>CC</sub></b> | Supply voltage range   | VDD, VDDIO                                 | -0,5 | 7,0                  | V    |
| <b>V<sub>I</sub></b>  | Input voltage range  | MOSI, SCK, nCS                             | -0,5 | 7,0                  | V    |
| <b>V<sub>O</sub></b>  | Voltage range applied to any output in the high or low state                 | MISO, nBUSY                                | -0,5 | V <sub>CC</sub> +0,5 | V    |
| <b>I<sub>IK</sub></b> | Input clamp current (V <sub>I</sub> <0)                                      | MOSI, SCK, nCS                             |      | -20                  | mA   |
| <b>I<sub>OK</sub></b> | Output clamp current (V <sub>O</sub> <0 or V <sub>O</sub> >V <sub>CC</sub> ) | MISO, nBUSY                                |      | ±20                  | mA   |
| <b>I<sub>O</sub></b>  | Continuous output current  | MISO, nBUSY                                |      | ±25                  | mA   |
| <b>I<sub>AU</sub></b> | Continuous current audio out   | AUDIO_P, AUDIO_N                           |      | ±1,4                 | A    |
| <b>t<sub>SA</sub></b> | Duration of AUDIO_P short AUDIO_N  | AUDIO_P, AUDIO_N<br>I <sub>LIM</sub> =2,8A |      | continuous           | time |
| <b>T<sub>S</sub></b>  | Storage Temperature Range  |  | -55  | 150                  | °C   |
| <b>T<sub>A</sub></b>  | Operating Temperature Range  |  | -20  | 75                   | °C   |

- (1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

### 3 Recommended Operation Conditions

over operating free-air temperature range (unless otherwise noted)<sup>(1)</sup>

|                       |                             |                | MIN | MAX   | UNIT |
|-----------------------|-----------------------------|----------------|-----|-------|------|
| <b>V<sub>CC</sub></b> | Supply voltage range        | VDD            | 5   | 5     | V    |
|                       |                             | VDDIO          | 3,3 | 5     | V    |
| <b>V<sub>I</sub></b>  | Input voltage range         | MOSI, SCK, nCS | 0   | VDDIO | V    |
| <b>V<sub>O</sub></b>  | Output voltage range        | MOSI, nBUSY    | 0   | VDDIO | V    |
| <b>T<sub>A</sub></b>  | Operating Temperature Range |                | -20 | 75    | °C   |

- (1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

### 4 Electrical Characteristics

at T<sub>A</sub> = 25°C

| VDDIO, SPI and nBUSY  |                           |                            | MIN  | MAX  | UNIT |
|-----------------------|---------------------------|----------------------------|------|------|------|
| <b>V<sub>IH</sub></b> | High-level input voltage  | VDDIO = 3,0V to 3,3V       | 1,37 |      | V    |
|                       |                           | VDDIO = 4,5V to 5,0V       | 2,02 |      | V    |
| <b>V<sub>IL</sub></b> | Low-level input voltage   | VDDIO = 3,0V to 3,6V       |      | 0,8  | V    |
|                       |                           | VDDIO = 4,5V to 5,5V       |      | 2,1  | V    |
| <b>V<sub>OH</sub></b> | High-level output voltage | IOH = -5,5mA, VDDIO = 3,3V | 2,78 |      | V    |
|                       |                           | IOH = -8mA, VDDIO = 5,0V   | 4,49 |      | V    |
| <b>V<sub>OL</sub></b> | Low-level output voltage  | IOL = 5,5mA, VDDIO = 3,0V  |      | 0,33 | V    |
|                       |                           | IOL = 8mA, VDDIO = 4,5V    |      | 0,48 | V    |
| <b>I<sub>L</sub></b>  | Input leakage current     | VDDIO = 0V to 5,5V         |      | 0,1  | μA   |
| <b>I<sub>CC</sub></b> | Static supply current     | I <sub>VDD</sub>           |      | tbd  |      |

## 5 Audio Amplifier

Features:

- Low RF Susceptibility Rejects TDMA Noise from GSM Radios
- Extensive Click-and-Pop Reduction Circuitry
- Robust Short-Circuit and Thermal Protection
- Active emissions-limiting, edge-rate limiting, and overshoot control circuitry greatly reduce EMI

Recommended operating conditions:

VDD = 5V, RL = 8Ohm

|        |             |                      |     | UNIT |
|--------|-------------|----------------------|-----|------|
| P      | Audio Power | VDD = 5V, RL = 8 Ohm | 1   | W    |
| $\eta$ | Efficiency  | VDD = 5V, RL = 8 Ohm | >92 | %    |

## 6 Audio Decoding

MP3 MPEG 1 & 2 audio layer III (CBR + VBR)

CBR = constant bitrate

VBR = variable bitrate

### MPEG 1.0:

Sample rate: 32000Hz, 44100Hz, 48000Hz

Bitrate [kBit/s]:

|    |    |    |    |    |    |    |     |     |     |     |     |     |     |
|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|
| 32 | 40 | 48 | 56 | 64 | 80 | 96 | 112 | 128 | 160 | 192 | 224 | 256 | 320 |
|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|

### MPEG2.0:

Sample rate: 16000Hz, 22050Hz, 24000Hz

Bitrate [kBit/s]:

|   |    |    |    |    |    |    |    |    |    |     |     |     |     |
|---|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|
| 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 80 | 96 | 112 | 128 | 144 | 160 |
|---|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|

### MPEG2.5

Sample rate: 8000Hz, 11025Hz, 12000Hz

Bitrate [kBit/s]:

|   |    |    |    |    |    |    |    |    |    |     |     |     |     |
|---|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|
| 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 80 | 96 | 112 | 128 | 144 | 160 |
|---|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|

Recommended Format:

### MP3 MPEG1

Sample rate: 44100Hz

Bitrate: 128kBit/s

Features:

- Zero-cross detection for smooth volume change
- Quiet power-on and power-off

## 7 Flash Memory

The audio data is stored on the flash memory. The micro SD card is only required for programming the flash memory.

|          |          |     | <b>UNIT</b> |
|----------|----------|-----|-------------|
| <b>S</b> | Capacity | 32M | Byte        |

## 8 Control Interface and Protocol

SPI is used as the control interface. The nBUSY signal is used as a status indicator. As soon as audio output is active, nBUSY is asserted.

SPI Configuration:

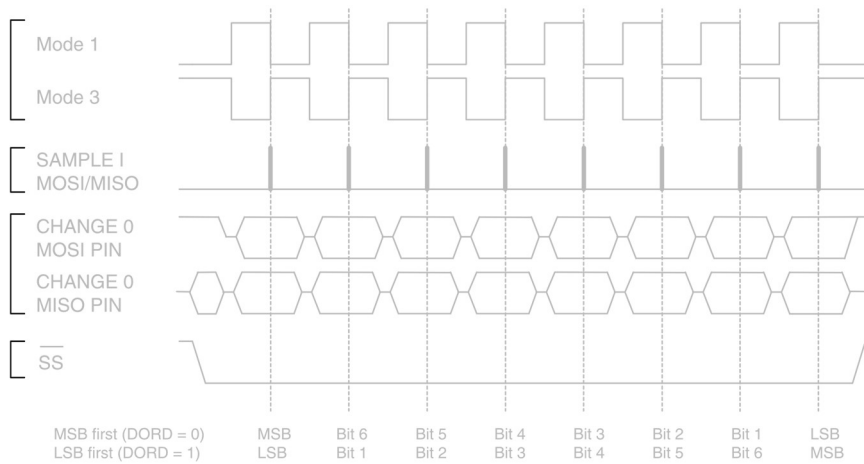
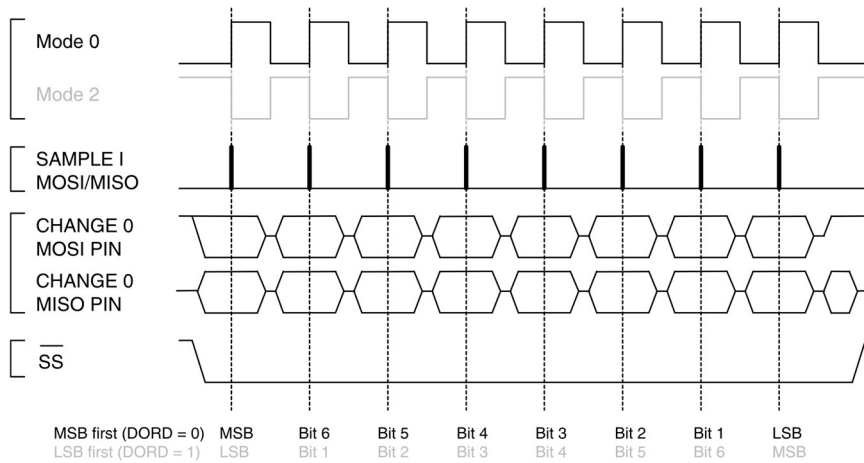
- MSB first (DORD=0)
- Mode 0 (CPHA=0, CPOL=0)
- 8Bit Frame
- SPI Clock (SCK) max. 5MHz

| <b>Mode</b> | <b>CPOL</b> | <b>CPHA</b> | <b>Leading Edge</b> | <b>Trailing Edge</b> |
|-------------|-------------|-------------|---------------------|----------------------|
| <b>0</b>    | 0           | 0           | Rising, sample      | Falling, setup       |
| <b>1</b>    | 0           | 1           | Rising, setup       | Falling, sample      |
| <b>2</b>    | 1           | 0           | Falling, sample     | Rising, setup        |
| <b>3</b>    | 1           | 1           | Falling, setup      | Rising, sample       |

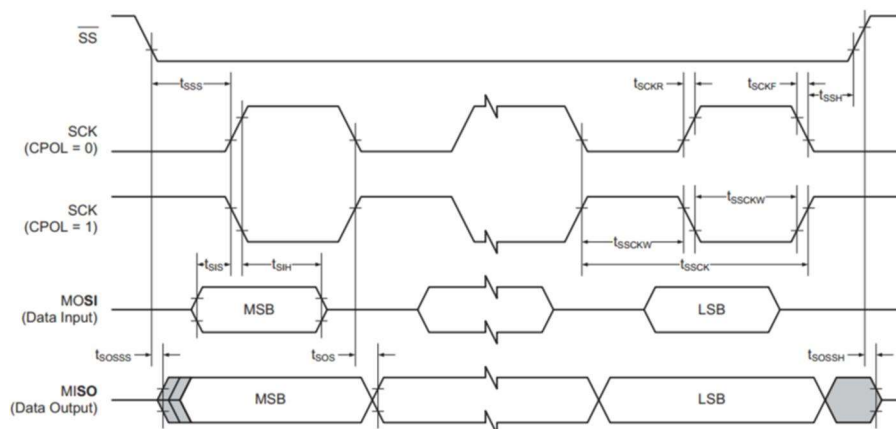
Note:

Leading edge is the first clock edge in a clock cycle.

Trailing edge is the second clock edge in a clock cycle.



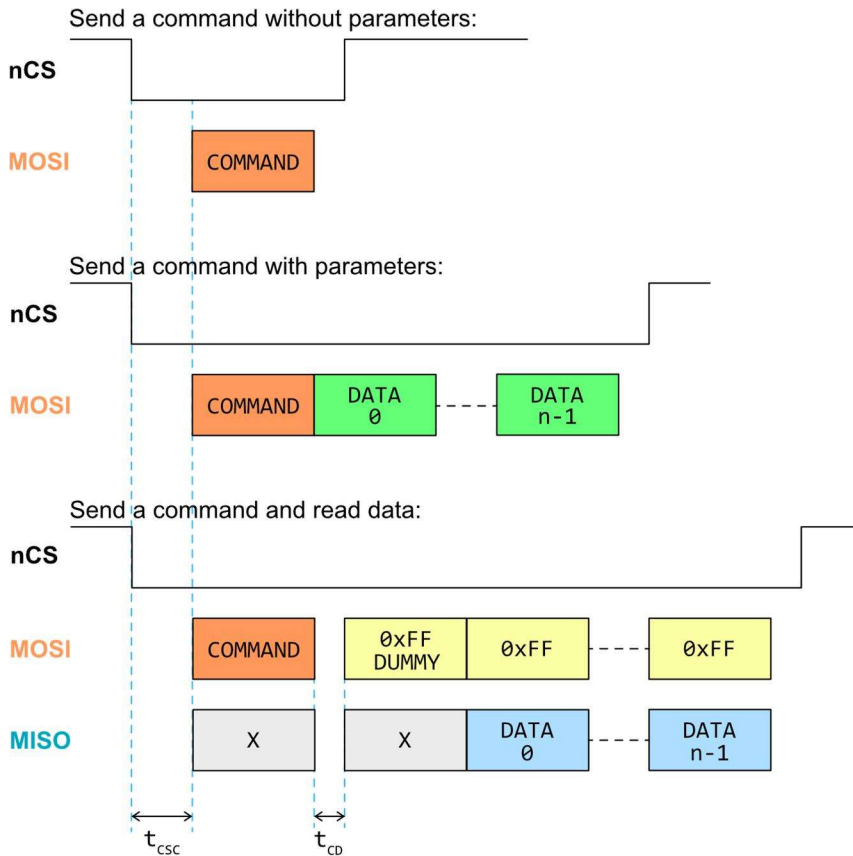
**Timing Requirements:**



|             |                               | MIN | TYP          | UNIT    |
|-------------|-------------------------------|-----|--------------|---------|
| $t_{ssck}$  | SCKmax = 1MHz<br>Clock period | 1   |              | $\mu$ s |
| $t_{ssckw}$ | SCK high/low width            |     | $t_{ssck}/2$ |         |
| $t_{sss}$   | SS (nCS) setup to SCK         | 1   |              | $\mu$ s |
| $t_{ssh}$   | SS (nCS) hold after SCK       | 200 |              | ns      |

## Typical protocol structure

- Command without parameters
- Command with parameters
- Command with read data



|           |                             | MIN | UNIT    |
|-----------|-----------------------------|-----|---------|
| $t_{csc}$ | Chip select to command time | 1   | $\mu s$ |
| $t_{cd}$  | Command to data time        | 10  | $\mu s$ |

## Interface signals

| Signal       | Description  |
|--------------|--|
| <b>nCS</b>   | Audio module chip select in, low active (nSS, select slave)  |
| <b>MOSI</b>  | Audio module data in   |
| <b>MISO</b>  | Audio module data out  |
| <b>SCK</b>   | Audio module SPI clock in  |
| <b>nBUSY</b> | Audio Play Status out, low active. As soon as audio output is active, nBUSY is asserted. When the flash memory is programmed, nBUSY is asserted. |

Signal nCS synchronizes the command frame, it does not switch MISO to high-impedance status. The control interface cannot therefore be used directly together with other SPI slaves

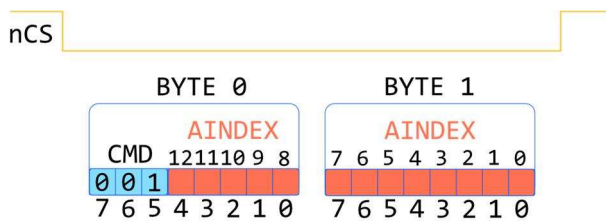
## Commands

| Value Decimal | Command                      | Description                           | Parameter                      |
|---------------|------------------------------|---------------------------------------|--------------------------------|
| 0             | CMD_STOP                     | Play stop                             | -                              |
| 1             | CMD_PLAY                     | Play start active audio index         | -                              |
| 2             | CMD_PAUSE                    | Play pause                            | -                              |
| 3             | CMD_SET_VOLUME               | Set volume                            | volume (0..255)                |
| 5             | CMD_READ_STATUS              | Read status                           | -                              |
| 6             | CMD_READ_FWVERSION           | Read firmware version                 | -                              |
| 7             | CMD_READ_FLASH_FILE_VERSION  | Read flash file version number (word) | -                              |
| 8             | CMD_READ_FLASH_FILE_DATETIME | Read flash file date time             | -                              |
| 32            | CMD_PLAY_AUDIO_FILE          | Play start an audio index             | audio index                    |
| 64            | CMD_PLAY_AUDIO_FILE_LANG     | Play start an audio index             | audio index and language index |

### 8.1 Command Play Audio File

Play an audio file.

The command uses parameters in the command byte to minimize the total number of bytes to be sent.



AINDEX = Audio index

#### Command:

CMD\_PLAY\_AUDIO\_FILE      32 (=0b001000000, 1<<5)

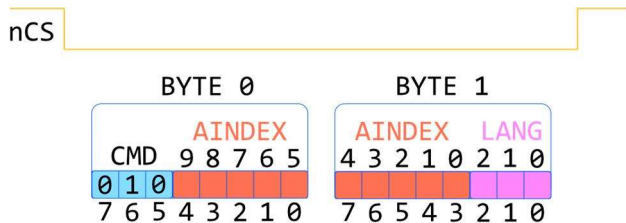
#### Parameter:

AUDIO INDEX [12:0]      Index of the audio file (0..7999), max. 8000 audio files

## 8.2 Command Play Audio File Language

Play an audio file with language mode.

The command uses parameters in the command byte to minimize the total number of bytes to be sent.



X = don't care  
AINDEX = Audio index  
LANG = Language index

### Command:

CMD\_PLAY\_AUDIO\_FILE\_LANG 64 (=0b01000000, 1<<6)

### Parameter:

AUDIO INDEX [9:0]                      Index of the audio file (0..999), max. 1000 audio files  
LANGUAGE INDEX [2:0]                    Index of the language (0..7), max. 8 languages

### Example:

0b01000000 + 0b00000000 → Play Audio index 0 language 0  
0b01000010 + 0b00000000 → Play Audio index 64 language 0  
0b01000000 + 0b00100010 → Play Audio index 4 language 2

## Sending Command and Parameter example in ANSI C:

```
bool audio_PlayFile(uint16_t index)
{
    bool result = false;

    if (index < 7999)
    {
        SPICONTROL_CS_LOW
        spi_transmit(CMD_PLAY_AUDIO_FILE | (uint8_t)((index >> 8) & 0x1F));
        spi_transmit((uint8_t)index);
        SPICONTROL_CS_HIGH

        result = true;
    }

    return result;
}

bool audio_PlayFile_Language(uint16_t index, uint8_t lindex)
{
    bool result = false;

    if ((index < 1000) && (lindex < 8))
    {
        SPICONTROL_CS_LOW

        // First Byte (Byte 0)
        spi_transmit(CMD_PLAY_AUDIO_FILE | (uint8_t)((index >> 5) & 0x1F));

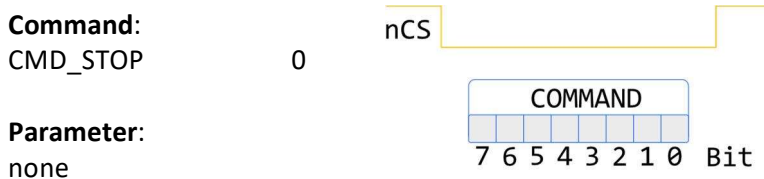
        // Second Byte (Byte 1)
        spi_transmit((uint8_t)((index << 3) | (lindex & 0x07)));

        SPICONTROL_CS_HIGH
        result = true;
    }

    return result;
}
```

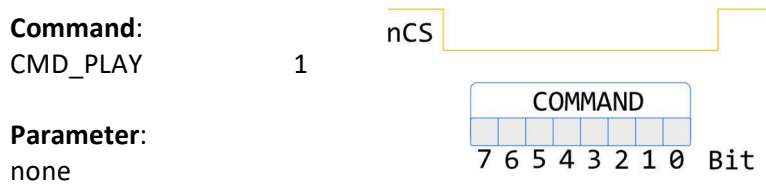
### 8.3 Command Stop

If audio is currently playing, the audio output is stopped.



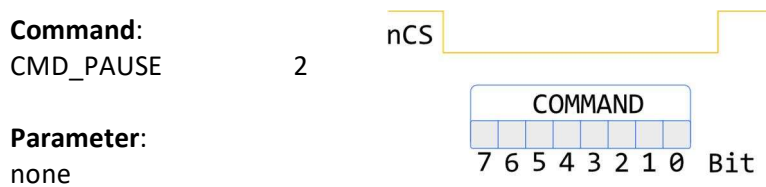
## 8.4 Command Play

The audio output with the current index is started. If the audio index exists, nBusy is asserted.



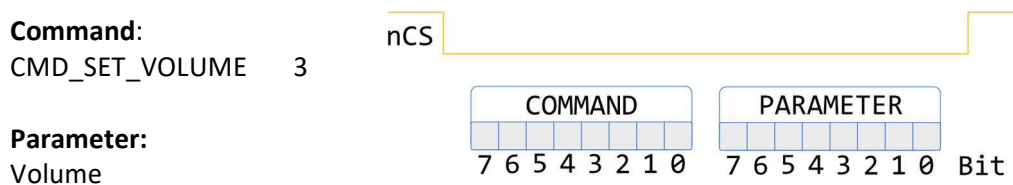
## 8.5 Command Pause

Audio output is paused at the current track position.



## 8.6 Command Set Volume

The command sets the current volume of the audio output.



Volume:

values between 0 and 255 (0x00..0xFF)

maximum volume is 0x00 and total silence is 0xFE

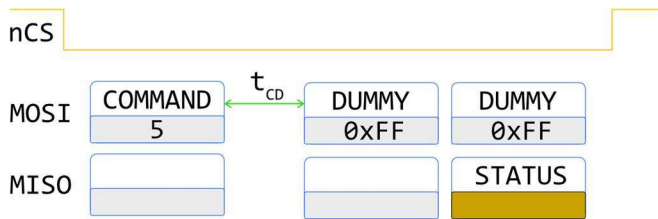
0xFF will activate analog power down mode

**Example:**

0b00000011 + 0b00000000 → Maximum Volume

## 8.7 Command Read Status

The command reads the status byte.

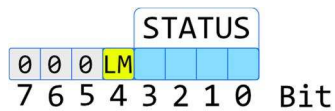


### Command:

CMD\_READ\_STATUS 5

### Response:

Status byte



Status:

Bit 0..3 decimal value

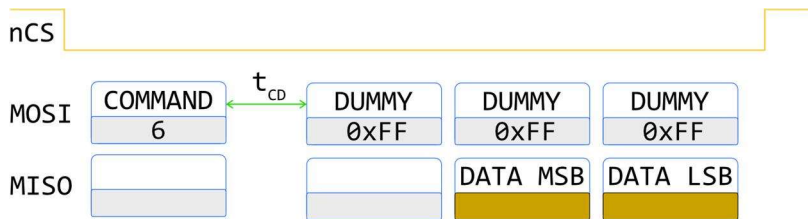
| Value | Status                           | Description   |
|-------|----------------------------------|---|
| 0     | STATUS_INIT                      | System initializing process   |
| 1     | STATUS_FLASH_EMPTY               | No audio data in flash memory (unprogrammed)  |
| 2     | STATUS_PLAY_AUDIO                | Playing audio (nBUSY asserted)  |
| 3     | STATUS_PLAY_READY                | Stop or paused playing audio (nBUSY deasserted)   |
| 4     | STATUS_FLASH_INDEX_NOT_FOUND     | The audio index used with the play command is not present in the flash data.                |
| 5     | STATUS_FLASH_PROGRAMMING         | Programming flash memory  |
| 6     | STATUS_FLASH_PROGRAMMING_SUCCESS | Programming flash memory successful   |
| 7     | STATUS_FLASH_PROGRAMMING_ERROR   | Programming flash memory error  |
| 8     | STATUS_MSD_READ_ERROR            | Micro SD Read data error (for example, because of wrong file system, source file not found) |

Bit 4: LM LanguageMode (is 1 if data file created with option languagemode)

Bit 5..7: for future use

## 8.8 Command Read Firmware Version

The command reads the firmware version number (word).



### Command:

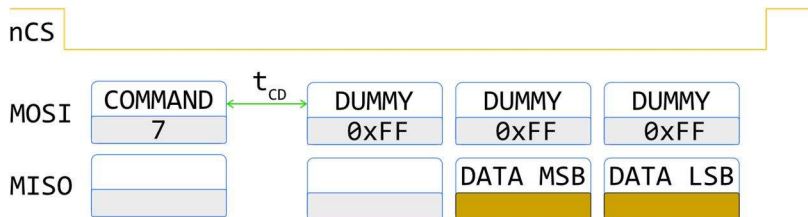
CMD\_READ\_STATUS 6

### Response:

Firmware version number (word, MSB first).

## 8.9 Command Read Flash File Version

The command reads the flash file version number (word).



### Command:

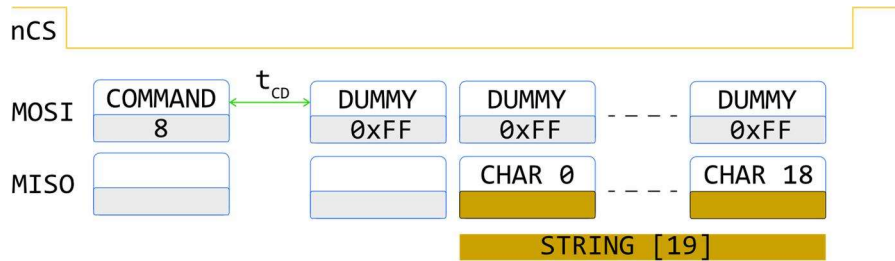
CMD\_READ\_FLASH\_FILE\_VERSION 7

### Response:

Flash file version number (word, MSB first). If unknown or invalid, data is 0xFFFF.

## 8.10 Command Read Flash File Date Time

The command reads the flash file date time.  
Date time is the date and time when the flash file was created.



### Command:

`CMD_READ_FLASH_FILE_DATETIME` 8

### Response:

Date time string (19 byte)

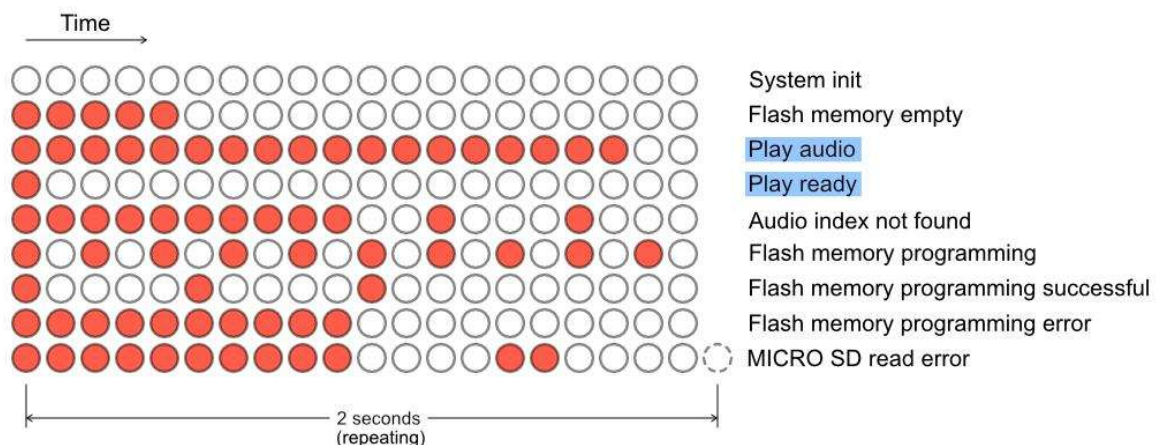
Represents date and time as an ASCII character string in the following format:

`2025-10-22 09:25:48`

Char 0 begins with the year. If unknown or invalid, all characters have the char value '?' (63 decimal).

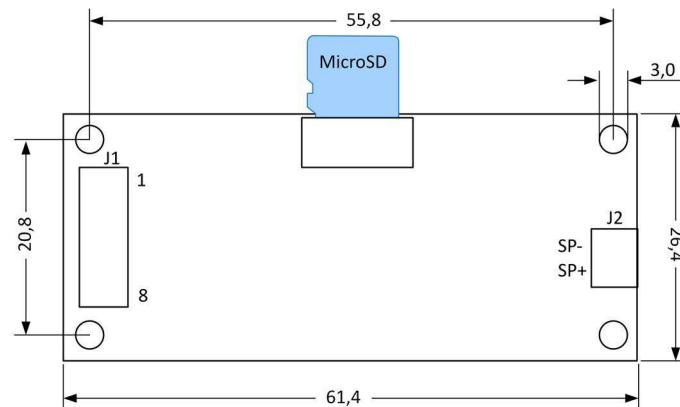
## 9 Status LED

The status LED on the module indicates different system states, which is particularly helpful during the development process.



The filled circles indicate a lit LED.

## 10 Dimensions



All dimensions in millimeters.

## 11 Interface

### Control and Power Supply (Connector J1)

| No | Signal | Typ | Description                                 |
|----|--------|-----|---|
| 1  | nBUSY  | OUT | Busy status (low active)                    |
| 2  | MOSI   | IN  | SPI IN                                      |
| 3  | SCK    | IN  | SPI Clock IN                                |
| 4  | MISO   | OUT | SPI OUT                                     |
| 5  | nCS    | IN  | SPI Chip Select (low active)                |
| 6  | VDDIO  | PWR | IO VDD for interface (typical 3,3V to 5,0V) |
| 7  | GND    | PWR |   |
| 8  | VDD    | PWR | Supply voltage 5V                           |

#### Assembled connector J1:

Würth 690367290876

Micromatch 8pin, 2,54mm

WR-MM Female SMT Connector with Latch & Polarization

#### Recommended connector J1:

Würth 690157000872

Micromatch 8pin, 2,54mm

WR-MM IDC Mini Module Male Connector with Polarization

### Audio output (Connector J2)

| No | Signal  | Typ | Description           |
|----|---------|-----|-----------------------|
| 1  | AUDIO_P | OUT | Audio positive output |
| 2  | AUDIO_N | OUT | Audio negative output |

#### Assembled connector J2:

Würth 648102131822

2pin

WR-WTB 1.50 mm SMT Male Horizontal Shrouded Header

#### Recommended connector J2:

Würth 648002113322

2pin

WR-WTB 1.50 mm Female Terminal Housing

### Programming interface (Connector SD1)

Connector for MicroSD card.

## 12 Create and program a project

### Data structure:

The source audio files for the project are stored in a project audio directory and managed via Windows Explorer. This makes it easy to port, modify, expand and duplicate the project.

You can create a project by simply copying MP3 audio files to an audio source directory and assigning them a track index prefix. Optionally, you can also specify an index for the language if you want to use multiple languages in your project.

Using the Windows command line program AAA, you can create an audio file container from the project audio files, which you can then program into the flash memory (32Mbyte) of the TuneSOne module via a MicroSD card (Use the FAT32 file system on the MicroSD card).

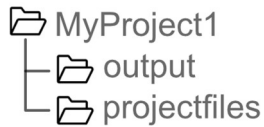
Depending on whether you want to use multiple languages in your project or not, there are different rules for naming the files.

- Without languages, you can manage up to 8.000 audio indexes.
- With languages, you can manage up to 1.000 audio indexes with 8 language indexes each.

### Console Application [AudioXFlash](#)

This console application is a Windows 64-bit application and has been specifically designed for x64 processor architectures. The Microsoft .NET 8.0 Framework is required to run the application.

## Project Directory structure:



**MyProject1** is your project directory, which you can name as you wish.

Copy the **AudioXFlash** program and the associated program files to your **MyProject1** project directory. The source audio files are located in the **projectfiles** directory.

The **AudioXFlash** program creates the file container **flash.dat** in the **output** directory (If the output directory does not exist, it will be created). In addition, a C/C++ header file **database.h** is created with project information.

## Rules for naming the audio source files:

### A

You do not want to use the audio files in multiple languages.

Example:

Audio main index

007\_Welcome.mp3  
008\_Starting.mp3  
142\_Success.mp3

Track naming

Audio main index is used to address the individual audio files.

**Audio main index: Value from 0 to 7999**

### B

You want to use the audio files in **multiple languages**.

Use parameter **LanguageMode** in the command line.

Example:

Audio main index

Language index

007\_0\_Welcome.mp3  
007\_1\_Welcome.mp3  
007\_2\_Welcome.mp3

Track naming

Audio main index and language index are used to address the individual audio files.

**Audio main index: Value from 0 to 999**

**Language index: Value from 0 to 7**

Note:

The track names within the same audio main index must be identical.

Track naming: Your mp3 file name. Please do not use spaces or special characters, as the name is used in the C/C++ header file.

Separator: Use the “\_” character as a separator

## Tips and hints:

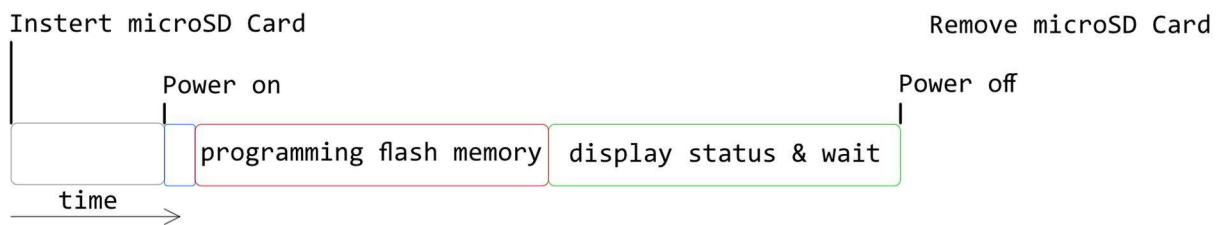
If you want to use more than one language, select the options parameter `LanguageMode`.

Use leading zeros in the audio main index so that Windows Explorer sorts alphabetically in the best possible way.

Always use different track names, as the name is used in the C/C++ header file and must not occur more than once. Please do not use spaces or special characters.

When creating a project for the first time, start by using an existing sample project. This will help you understand and learn the naming rules more easily.

## Programming

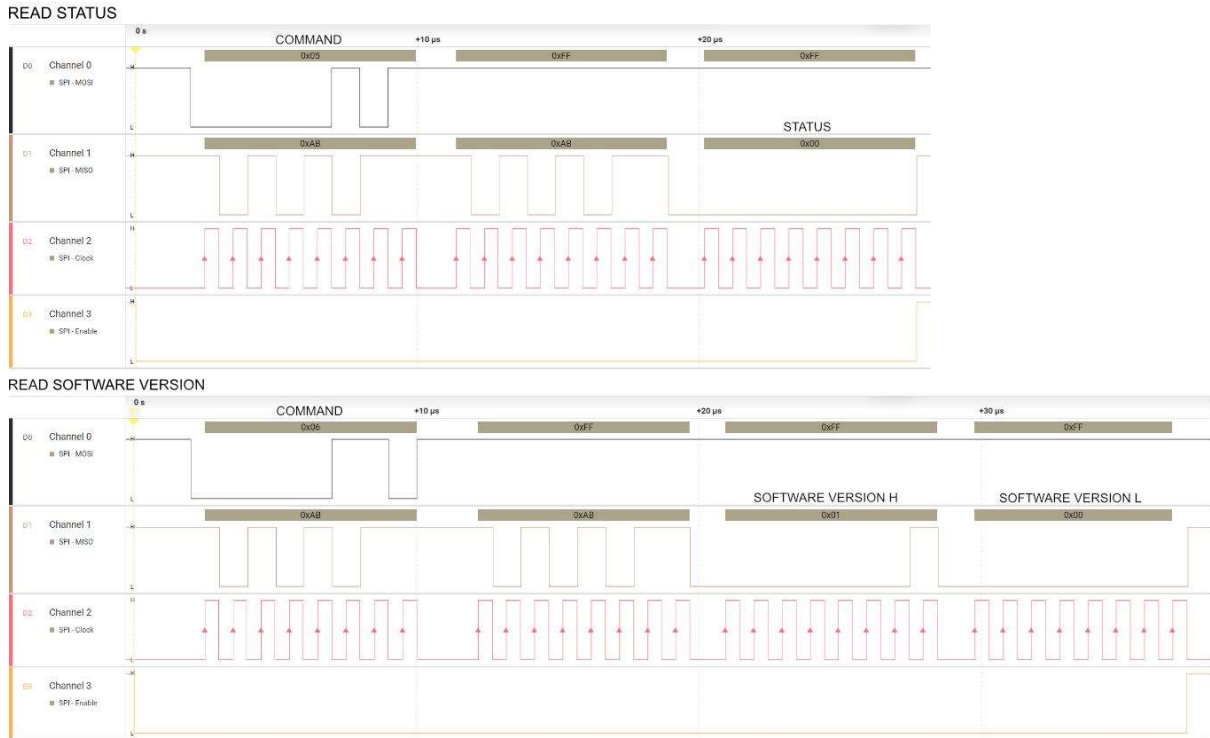


Filesystem MicroSD Card: FAT32  
Programming file: flash.dat

# 13 Appendix

## 13.1 Example SPI data transfer

The following diagrams show a logic analyzer recording.



## 14 Legal notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. RODENHAUSEN MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. RODENHAUSEN disclaims all liability arising from this information and its use. Use of RODENHAUSEN devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless RODENHAUSEN from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any RODENHAUSEN intellectual property rights unless otherwise stated.



[www.displayone.de](http://www.displayone.de)  
[www.rodenshausen.de](http://www.rodenshausen.de)

Rodenshausen Electronic  
Berggartenstrasse 5  
35457 Lollar-Odenhausen  
GERMANY