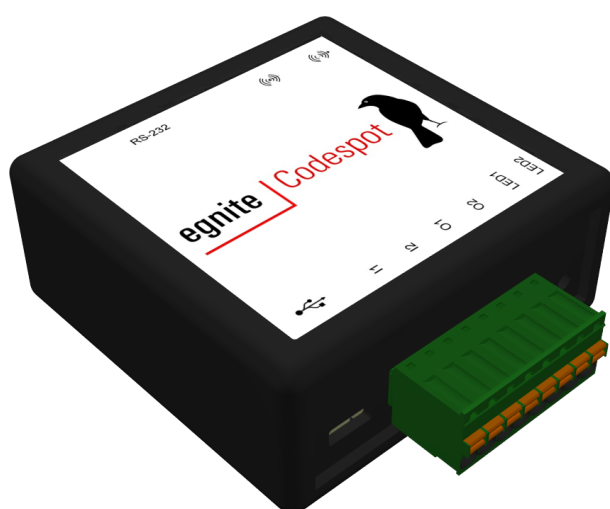


Codespot Dev Manual

Low power DSP box for the development
of audio player applications



Contents

1	Introduction.....	1
1.1	About Codespot Dev	1
1.2	Codespot Dev properties.....	1
1.3	Applications	2
2	Included in delivery.....	3
2.1	Codespot Dev (Article EGN102014)	3
2.2	Codespot Dev Starter Kit (Article EGN102114).....	3
2.3	Furthermore you will need.....	3
3	Codespot Dev at a glance.....	4
3.1	Front side.....	4
3.2	Rear side.....	5
4	Codespot Dev put into operation.....	6
5	Install development environment	8
5.1	Install VSIDE	8
5.2	Install Codespot Dev templates	10
6	Generate applications.....	14
6.1	Connect Codespot Dev with the PC	14
6.2	Hello World template	14
6.3	Audible Hello template	18
6.4	StereoFIR template	18
6.5	Standalone Recorder template	19
7	Codespot Dev hardware.....	21
7.1	Electrical supply	21
7.2	Connect serial interface	21
7.3	Connect audio input and audio output	21
7.4	Connect digital inputs and outputs	22
7.5	Opening the housing.....	24
7.6	Change memory card	24
7.7	Configure boot jumper	25
7.8	Configure reset jumper	25
7.9	Configure jumper for digital inputs	25
8	Attachment	27
8.1	Specifications.....	27
8.2	Assignment of the GPIO ports	28
9	Fittings.....	29
9.1	I/O test board Bintest (Artikel EGN300513)	29
10	More information	30

1 Introduction

1.1 About Codespot Dev

Codespot Dev is a DSP development system for audio applications and it is based on the VS1063 audio codec of the Finnish manufacturer VLSI Solution.

The DSP box has two galvanically isolated digital inputs enabling it to react to external contacts or voltage. Two digital outputs, likewise galvanically isolated, can be used for the control of external devices. Two 3.5 mm jack plugs provide a stereo line input and a stereo line output, whereby the latter is also suitable as a headphone output.

Custom applications are created on a Windows PC in the programming language C or in Assembler with the aid of the VSIDE development environment, which is available free of charge. As a developer you will in the beginning require C skills and you should have at least initial experience of assembler programming for any microcontroller. Codespot Dev offers you the chance to gather practical experience in DSP hardware and in the digital processing of audio data. Prepared example projects facilitate this introduction.

Firmware downloading and in-system debugging occur via a RS-232 interface. This interface can also be used by the application to communicate with external devices.

If required, the firmware can be permanently stored in a serial flash memory and this will then be automatically activated when it is switched on. The flash memory also offers sufficient storage for extensive configuration data.

A microSD card holder is incorporated for larger data volumes e.g. audio files. Files can be exchanged between a PC and microSD card by means of an integrated USB to SD card bridge. Special drivers are not needed for this.

1.2 Codespot Dev properties

- VS_DSP4 Core with high-quality audio DAC / ADC
- 16 kBytes code RAM and 80 kBytes data RAM
- Playback of Ogg Vorbis files
- Playback of MP3 files in the formats MPEG 1&2 Audio Layer III with fixed and variable bitrate
- Playback of WAV files in the formats IMA ADPCM, G.711 μ -law/A-law und G.722 ADPCM
- Playback of other formats available on request, e.g. AAC, WMA or FLAC
- Encoding of Ogg Vorbis, MP3, PCM, IMA ADPCM, G.711 μ -law/A-law und G.722 ADPCM
- Stereo line input and output
- Direct operation of headphones possible
- Two 5000 Vrms isolated inputs, contact or voltage controlled
- Two 1500 Vrms isolated outputs with load capacity up to 400 mA (peak <10 ms 1 A) at 48 V
- Integrated USB microSD card reader
- Serial interface according to RS-232
- LED status display
- 5 V power supply via USB interface or power supply plug

1.3 Applications

- Learning system for audio DSP applications
- Development of event-triggered audio
- Development of digital effect devices such as filter, echo etc.
- Expansion of existing systems with audio output
- Replacement of mechanical recording and playback device

2 Included in delivery

2.1 Codespot Dev (Article EGN102014)

- Codespot Dev
- Connector plug, 8-pin, 3.5 mm
- microSD card (in device)
- Templates for the VLSI Solution Software VSIDE
- Manual
- 2 jumper, RM 2.0

2.2 Codespot Dev Starter Kit (Article EGN102114)

- Codespot Dev
- Connector plug, 8-pin, 3.5 mm
- MicroSD card (in device)
- Templates for the VLSI Solution Software VSIDE
- Manual
- USB power supply
- USB cable, A to micro B
- Serial cable, connector / jack
- 2 jumper, RM 2.0

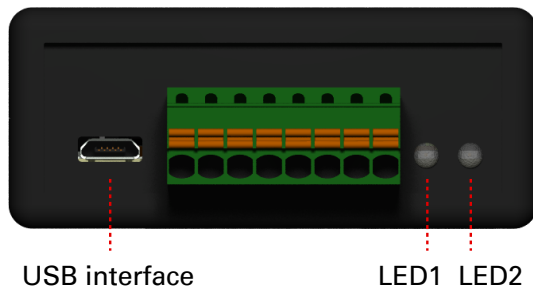
2.3 Furthermore you will need

- PC with Windows operating system (XP or more recent)
- RS-232 interface to PC, e.g. via USB / RS-232 bridge
- USB cable, A to micro B (included in the starter kit)
- Serial cable, connector / jack (included in the starter kit)
- Head phones or play-back device with line input
- Optional audio source with line output, depending on application
- Screwdriver

3 Codespot Dev at a glance

3.1 Front side

On the front of the device connections for the USB interface and the digital inputs and outputs can be found, as well as 2 light-emitting diodes for status indication.



USB

Codespot Dev is supplied with electricity via the USB interface, either via the connected PC or via a 5 V power supply unit with a USB port. Furthermore this interface provides access to the internal microSD card for the management of audio files.

Digital inputs

Two digital inputs detect contact statuses or voltages.

Digital outputs

Two robust galvanically isolated outputs (solid state relay) facilitate the control of external devices.

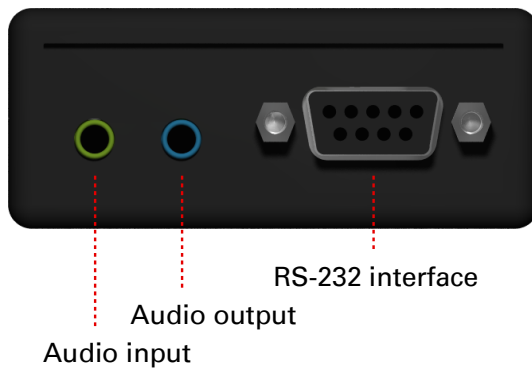
LEDs

LED1 shines red when it is turned on. There is the option of switching the LED by programme from red to green.

LED2 shines green when it is turned on. In card reading operation, access to the internal memory card will be shown by a brief change to red.

3.2 Rear side

To the rear of this device there are connections for the audio output, the audio input and the serial interface.



Audio output

The audio output is used as a stereo line output for connection to an external amplifier. Alternatively you can connect headphones to it with an input impedance of at least 30 ohm.

Audio input

The audio input is designed as a stereo line input.

RS-232 interface

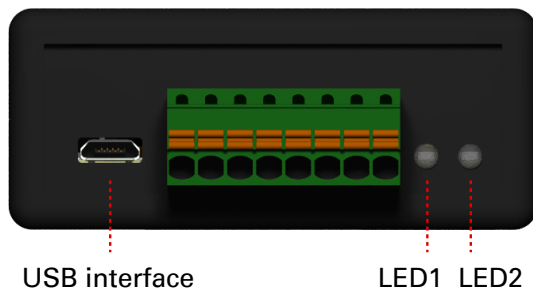
Programming and firmware debugging take place via the serial interface. Applications can use the serial interface to control external devices or to receive commands from external devices.

4 Codespot Dev put into operation

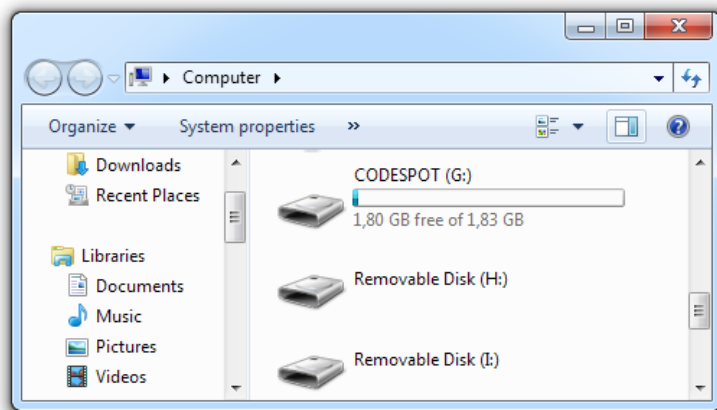
Codespot Dev is fitted with an internal microSD card. All the necessary programmes and documents for development are found on this. To start with you should transfer this card to the hard drive of your PC.

Codespot Dev in card reading operation is utilized for the transfer of the data to your PC. To do this, connect the micro USB jack of Codespot Dev with a free USB port of your PC. An external power supply is not required because Codespot Dev is supplied via the USB interface.

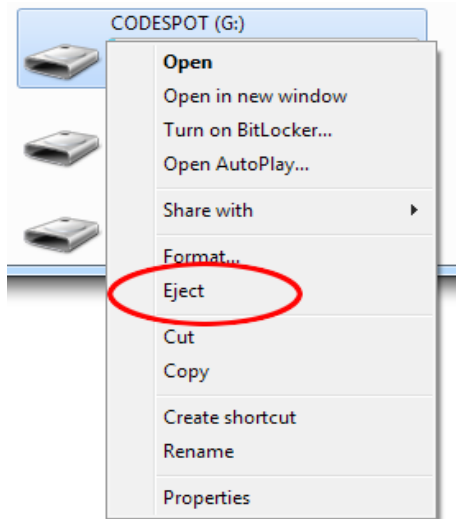
LED1 now shines red and LED2 shines green. Codespot Dev will automatically recognize the PC as a card-reading device and will attempt to access the internal memory card. Every attempt at access causes LED2 to briefly shine red.



As with any conventional card reader, the microSD card appears on the PC as a drive. You can transfer the files in the usual way, e.g. with Windows Explorer.



Log off from the disk drive when you have transferred all files. To do so click in Windows Explorer on the Codespot hard drive with the right mouse button and select *eject*.



Codespot Dev then switches back into the normal mode. If you want to switch back to card reading operation later on then you just need to briefly disconnect the USB connection.

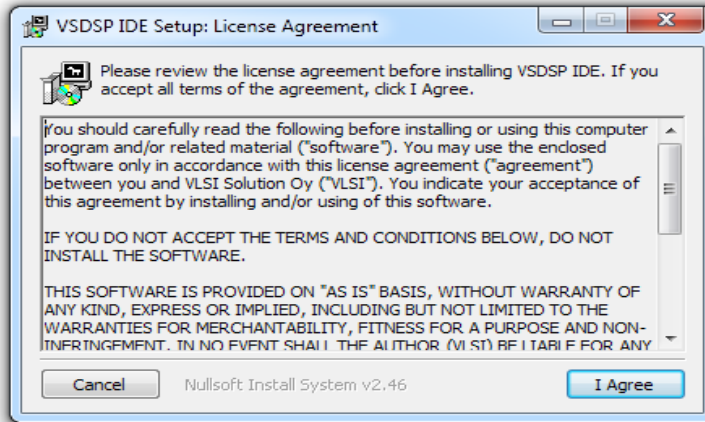
Of course you can also transfer files from the PC to Codespot Dev, delete files or reformat the entire memory card.

5 Install development environment

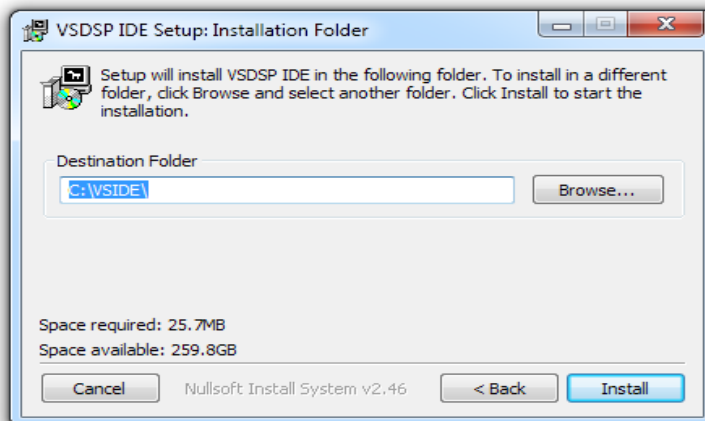
5.1 Install VSIDE

Among the files saved by the microSD card you will find the VSIDE installation with the name *vside_win32.exe*. Start the installation, e.g. with a double click on the file.

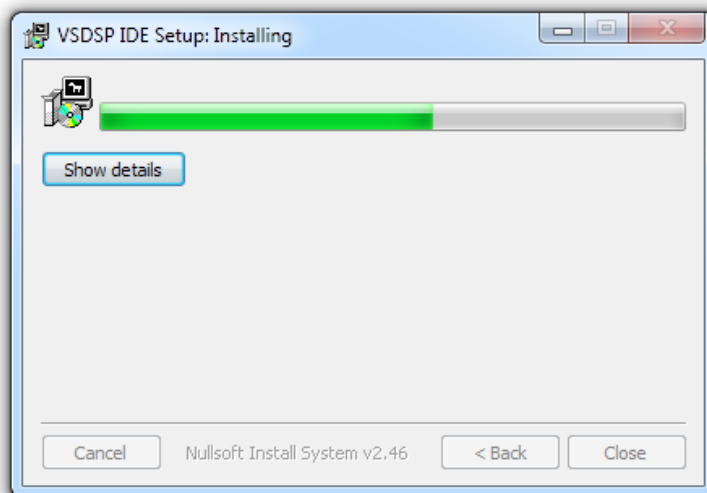
First of all a licence agreement dialogue will appear asking you to accept the terms and conditions of the licence. Click on *I Agree* to consent to this.



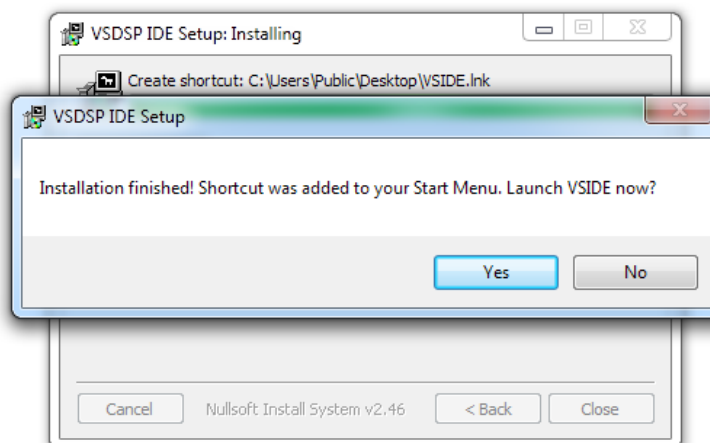
In the next dialogue you can select the installation folder. However it is recommended that you keep the suggested folder with a mouse click on *Install*. If you select another folder then it is important to make sure that the entered path does not include any spaces or umlauts.



The installation programme now starts by copying the data.

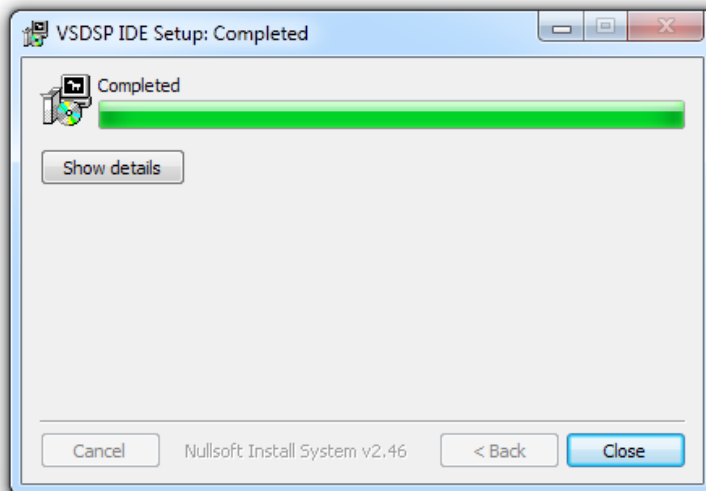


After installation is complete you can select whether you want to directly launch the VSIDE.



As the installation of the Codespot template is still pending, VSIDE should not yet be started. Accordingly you should click on *No*.

Click on *Close* in the last dialogue to end the installation programme.

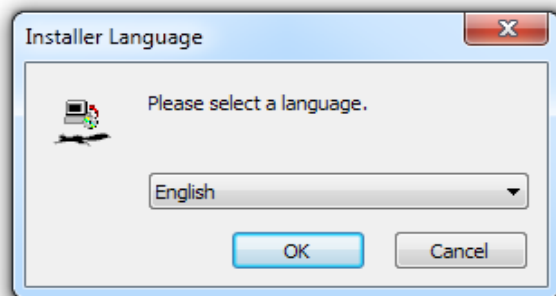


5.2 Install Codespot Dev templates

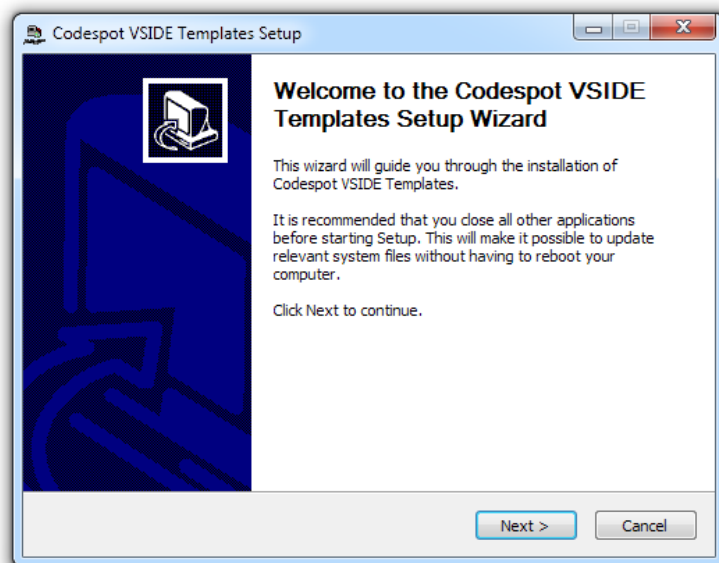
In order to install the Codespot Dev templates you must have already previously installed the VSIDE.

Among the files saved by the microSD card you will find the template installation with the name *codespotvside-setup.exe*. Launch the installation e.g. by double clicking on the file.

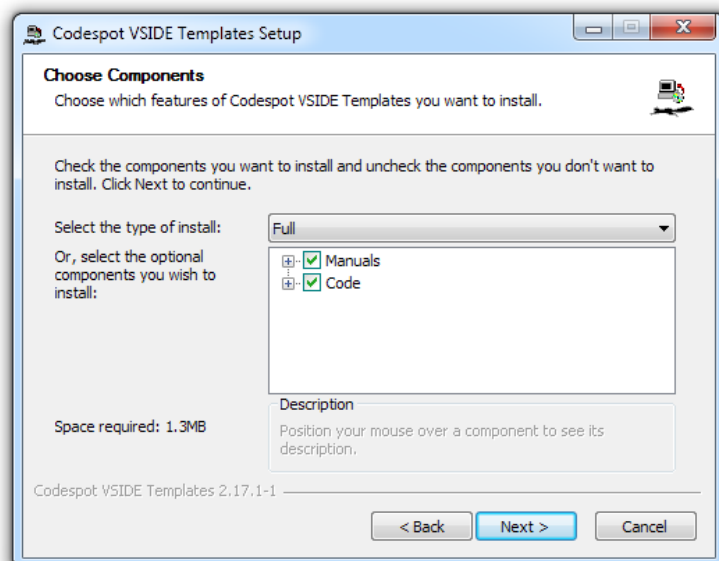
First of all a dialogue about the choice of language will appear. Currently only English is supported.



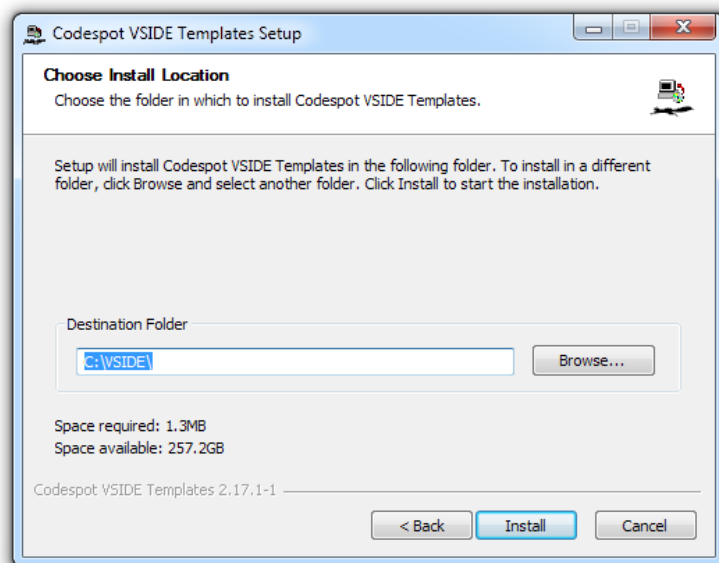
The following welcome dialogue appears in the selected language. Click on *Next* to continue.



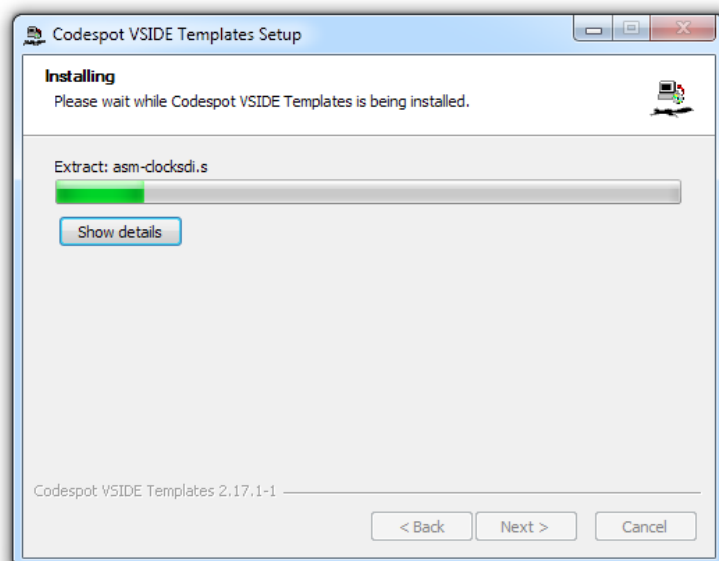
In the following dialogue you have the option of installing only part of the package. However we recommend installing the entire package.



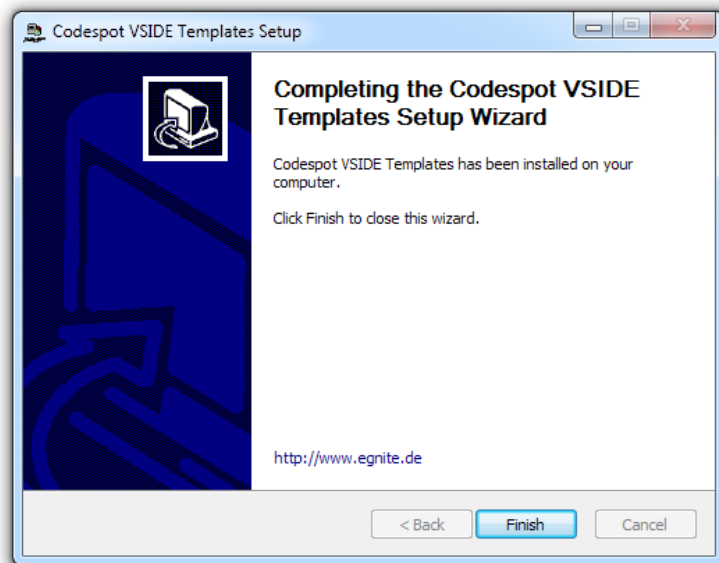
Select the installation folder from the next dialogue. Select the same folder here as selected for installation of VSIDE. Then finally click on *Install*.



The installation programme now begins with the files being copied.



Click on *finish* in the last dialogue to end the installation programme.



6 Generate applications

The development of the firmware occurs on a Windows PC with the assistance of VSIDE and the corresponding Codespot Dev templates. Among others the source code of an audio player is available which is suited as a basis for customized applications.

6.1 Connect Codespot Dev with the PC

The RS-232 interface is used for communication between the PC and Codespot Dev. You must connect Codespot Dev with the RS-232 interface (COM port) of the PC.

In the event that the PC does not have such an interface then you can alternatively make use of a so-called USB / RS-232 bridge. This is available from computer specialist stores. One side of the USB / RS-232 bridge is plugged into a free USB port on the PC while the other side is connected with an RS-232 cable to the RS-232 interface of Codespot Dev.

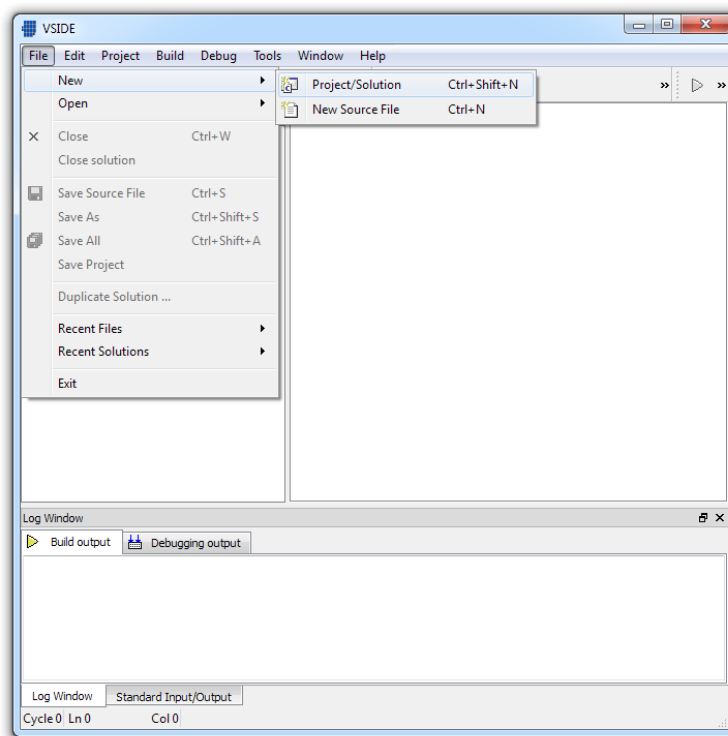
Supply Codespot Dev with electricity via the micro USB jack, preferably with a USB power supply.

Note: There is also the option of providing Codespot Dev with power via the USB port of the PC. However in the process Codespot Dev will be switched to card reading operation, which inhibits communication via the VSIDE. To change to normal operation you must eject the appropriate disk drive in Windows Explorer again, see #Put Codespot Dev into operation.

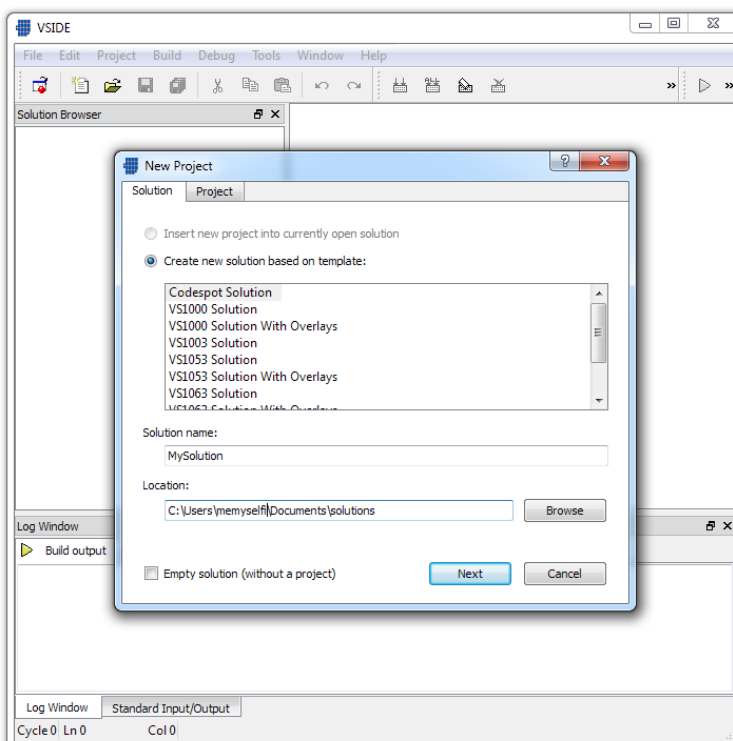
6.2 Hello World template

Using the example of the simple programme "Hello world" it will be demonstrated how a customized application can be generated with the help of the VSIDE and how it can be got running on Codespot Dev.

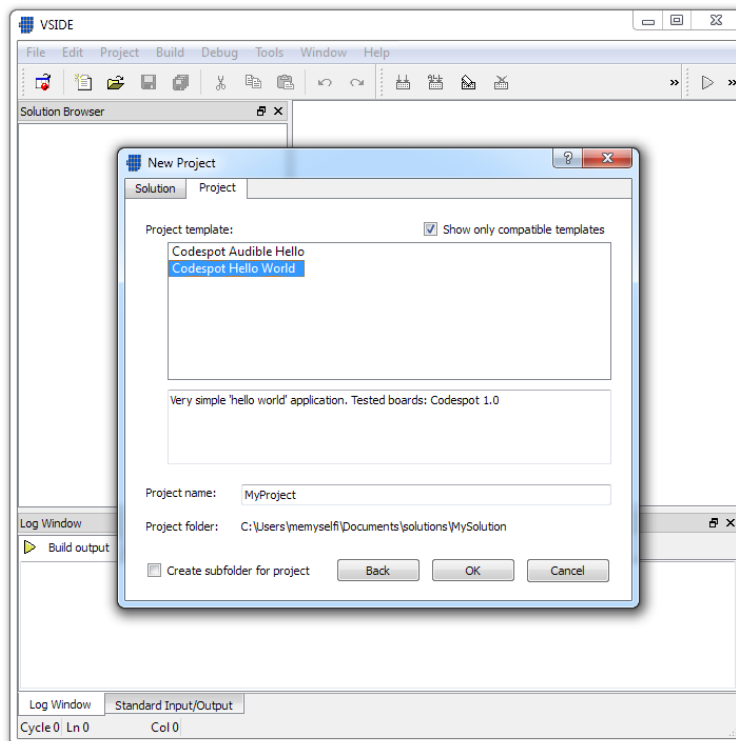
Start the VSIIDE and select *File -> New -> Project/Solution* from the main menu.



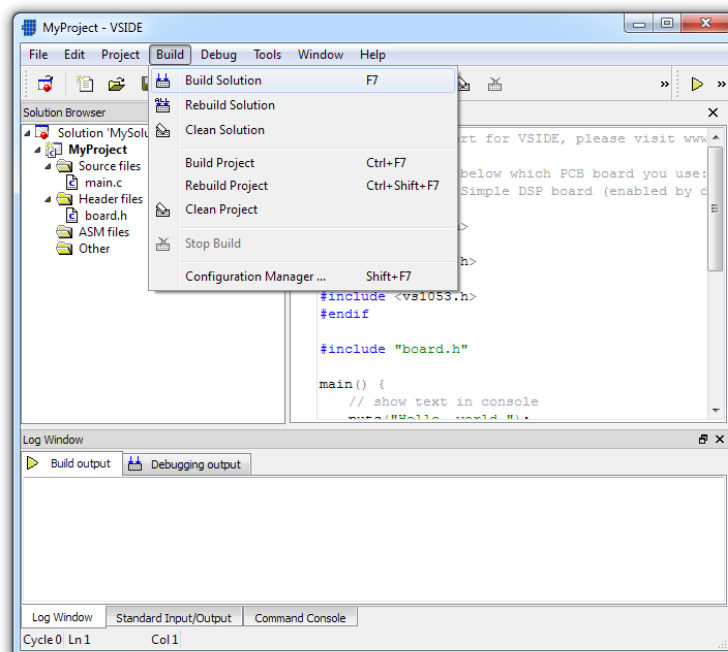
The following dialogue lists all prepared solutions. Select *Codespot Solution* and then click on *Next*.



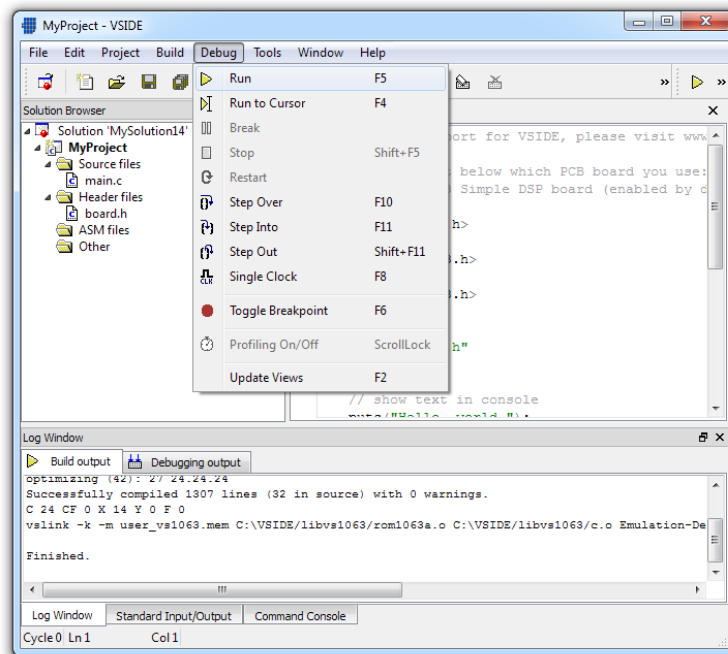
The dialogue moves to the next page on which all the available project templates for the selected solution are listed. Select the project *Codespot Hello World* and click on *OK*.



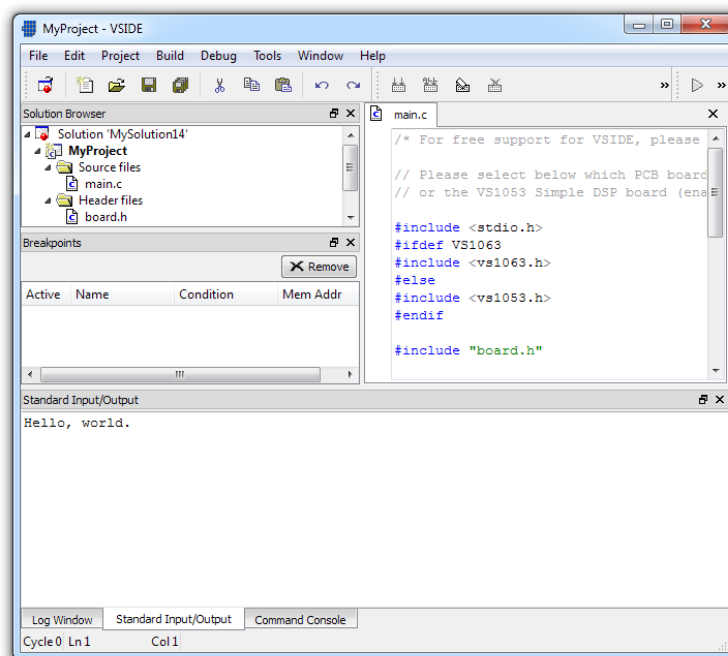
The project with the therein-contained source files now appears in the main window. Select *Build -> Build Solution* from the main menu to compile and to link the source data.



Once the procedure has been completed without fault then select *Debug -> Run* from the main menu.



The application is transferred to Codespot Dev and then started. After a few seconds the edition *Hello, world.* should appear in the lower window.



As an additional function the program shows the status of the first digital input via LED1. More detailed information about the digital inputs can be found in the section *#Connect digital inputs and outputs*.

Congratulations you have just successfully created and started your first Codespot Dev application.

Next you can for example change the issued text by modifying the *main* function's command *puts*. Furthermore there is the option of interrupting the code via the debugger, of showing variables or memory areas and much more. All necessary information about the VSIDE can be found in the related manual.

6.3 Audible Hello template

A further application that is available as a Codespot Dev template is "Audible Hello." This time only *Hello* is issued, but not as text rather as an audio sample. In addition to this, there is a short MP3 file saved in a C array, which is played after the start of the programme.

The loading of the relevant template in the VSIDE and the transfer of the binary code in Codespot Dev occurs in the same way as in the previous example #Hello world, therefore no further explanation is given here.

The application "Audible Hello" differs from "Hello world" not only in its output format, audio instead of text, but it also has a completely different internal structure.

Whereas "Hello world" is a C application with a *main* function, "Audible Hello" is a so-called stand-alone player. Instead of the function *main* this has the function *MyMain*. Although this will also be launched after the start, it then handles the routines provided in the internal ROM, which decode the MP3 code in the background and cater for the audio output. Whenever the CPU idles, the function *UserHook* is called up. In our example the MP3 files in the array are transferred into the decoder buffer of this function. Real applications are used by the *UserHook* for controlling the player, e.g. to get the status of keys, to read audio files from the memory card etc.

6.4 StereoFIR template

In this application audio files are read from the audio input, filtered and then finally reissued via the audio output. To be able to understand this example you ought to be familiar with calculating digital filters. Furthermore you ought to have a basic idea of how to process audio data with the aid of a DSP.

We have already described how to create the project from a template in the "Hello world" example.

After the code has been created in the VSIDE and loaded in Codespot Dev you can feed the audio input with an external source and receive the result of the filter at the audio output. However the filter will only be turned on when digital input 1 has been activated, see #Connect digital inputs and outputs.

To be able to make use of the DSP's special set of commands, the filter itself is programmed in assembler language. The coefficients of the FIR filters realized here are located in a memory area of the assembler code, starting with the label *_firCoeffs* in the file *firs.s*. Various sets of coefficients for different frequency responses are available and can then be selected via the *FIR_LEN* macro in the header file *stereofir.h*.

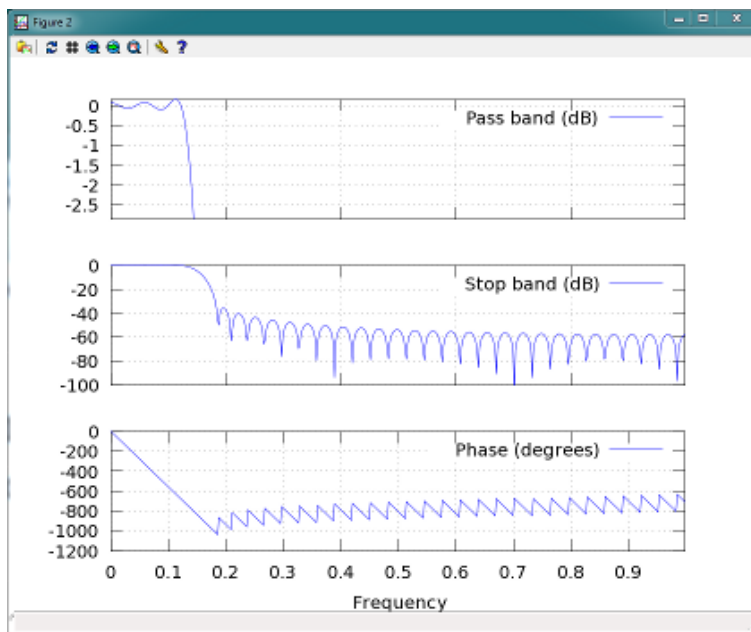
In order to be able to calculate your own filter coefficients, you can use the free programme GNU Octave or alternatively MATLAB, Scilab or similar. When

installing Octave make sure that the *signals* package is also installed and then select *gnuplot* for the graphic output.

You can calculate a low pass FIR filter in the entry field of Octave with the following commands:

```
pkg load all
fs=48000; fsp2=fs/2; f1=3100; f2=4300; deg=62;
b=firls(deg, [0 f1/fsp2 f2/fsp2 1], [1 1 0 0]);
freqz(b);
```

You should then be shown the following output:



Because our CPU cannot support any floating-point numbers the coefficients must still be converted into 16 bit values.

```
b2=round(b'*32768);
save 'coeff.txt' b2
```

After this the coefficients are located in the *coeff.txt* file and can be taken into the assembler code *firs.s* from there. Of course the values for this must be transmitted in a form that the assembler can understand. So this means in exactly the same form as the prepared coefficients that are already available in the file.

6.5 Standalone Recorder template

This example is a one-to-one porting of the VLSI application. Only the GPIO ports to Codespot Dev are adapted here.

This involves a complete audio player with the additional option of recording

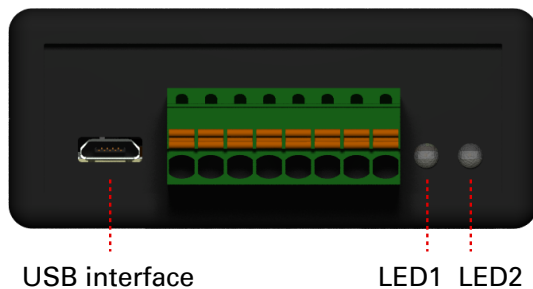
audio signals at input on the memory card. It is controlled either via commands to the serial interface or via keys, although with Codespot Dev the latter is restricted to its 2 digital inputs.

The application is very extensive and makes use of nearly all the available features of the VS1063. Accordingly a certain amount of experience in dealing with VLSI codecs is required. A detailed description can be found in the original application.

7 Codespot Dev hardware

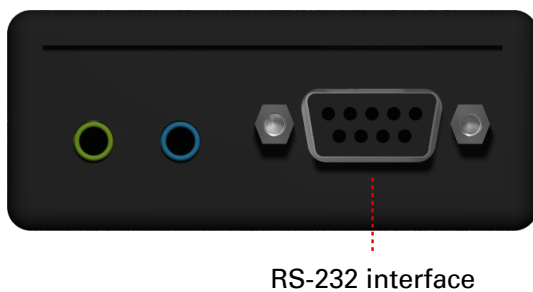
7.1 Electrical supply

The electrical supply is provided via the micro USB jack on the front side, either via the connected PC or via an external USB power supply unit.



7.2 Connect serial interface

The serial interface is used for programming and for firmware debugging. Your applications can equally make use of the serial interface (UART) to control external devices or to receive commands from these.

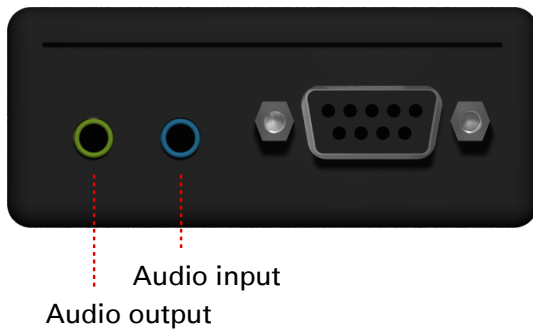


In certain circumstances it can be useful to restart Codespot Dev externally without having to cut the supply. This can be done optionally via the DTR pin of the serial interface, see #Configure reset jumper. To do this the housing needs to be opened up.

7.3 Connect audio input and audio output

You can connect the audio output (green jack) to regular headphones or with the line output of an external amplifier.

The audio input (blue jack) can be connected with the line output of an external amplifier. You will need an additional pre-amplifier if you want to connect a microphone.

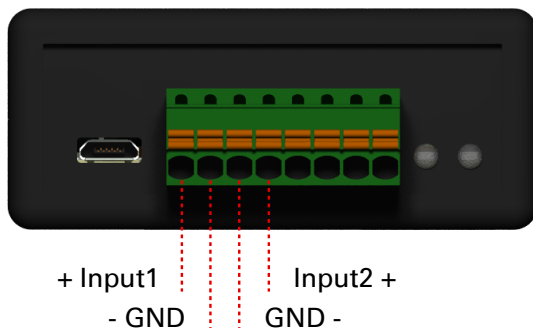


Both the input and the output are designed for stereo operation, whereby the following connector assignment is expected:

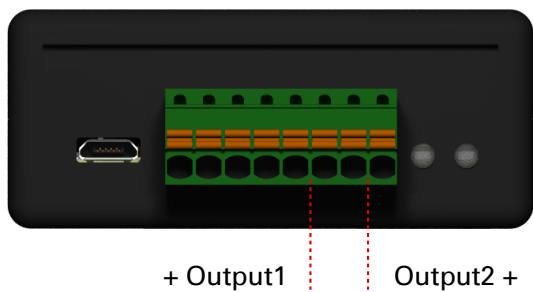
7.4 Connect digital inputs and outputs

The digital inputs are configured as contact inputs at the time of delivery. Please note, that the minus poles of both inputs are connected internally to each other, which means that they are not galvanically isolated.

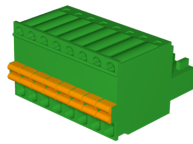
Alternatively every input can be re-configured individually as a voltage input, see #Configure jumper for digital inputs. In this way a galvanic isolation of both inputs can be achieved.



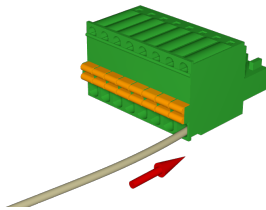
Both digital outputs are available as contact outputs and are open in their basic state. They allow to drive up to 400mA, permitting a voltage of up to 48 V.



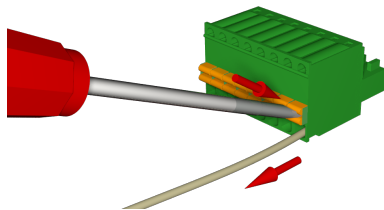
To enable the simple attachment of connecting wires, the inputs and outputs are connected via a plug-in terminal. This is fixed quite firmly. Take hold of the terminal on both sides with your thumb and index finger and loosen this from the housing by gently moving it from side to side as you pull it out.



Remove the connection wire insulation by about 8 mm (0,3 "). You can fit the wires with ferrules or insert them directly into the larger openings of the contact plug. The connection wires are held in place by a spring clamp. Carefully pull on the wires to assess whether they are securely attached. When using stranded wires or very thin wires make sure when inserting them that you push against the orange lever with a screwdriver (2.5 mm / 0.1 " head).



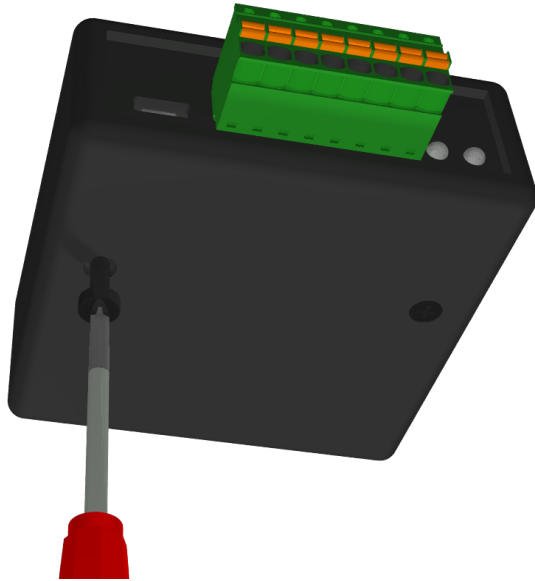
To remove a wire you also use a screwdriver to push against the orange lever.



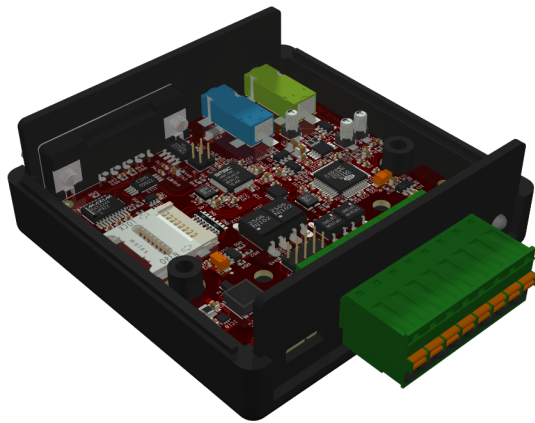
7.5 Opening the housing

You will need to open the housing to change the memory card, the function of the digital inputs or to determine behaviour after switching on.

Open the device by loosening and removing both screws on the under side with a screwdriver.



You can then lift off the upper part of the housing.



7.6 Change memory card

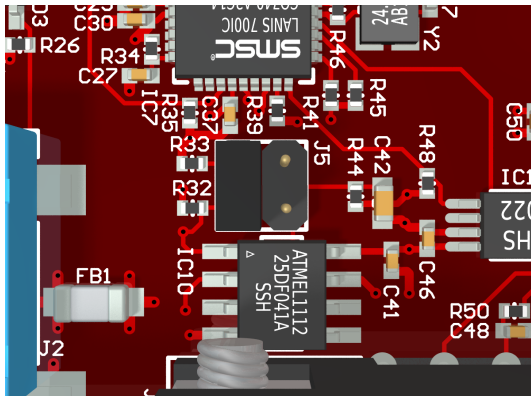
Push lightly with your finger on the silver cover of the memory cardholder and push this away from the edge of the housing in the direction of the device's centre by approximately 1 to 2 mm. Now it should be easy to lift the cover and to remove the memory card.

After inserting the new card shut the lid of the housing, then push this back in the original direction until it locks into place.

7.7 Configure boot jumper

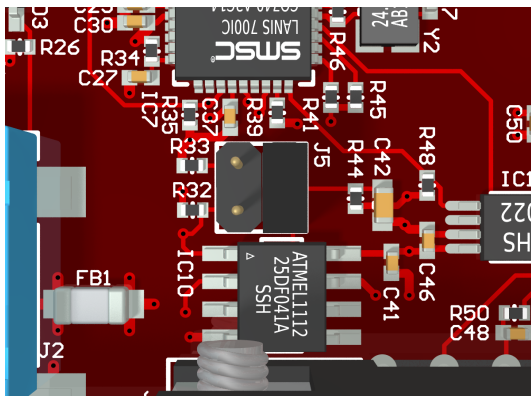
In its delivery state the firmware in ROM takes over control of the CPU once the device has been switched on and enables the loading and debugging of programs via the serial interface.

Alternatively you can store the firmware that you have created in the serial flash memory. To activate this straight after you have turned on the device you need to remove the plug-in jumper that connects pins 1 and 2 of the jumper field J5.



7.8 Configure reset jumper

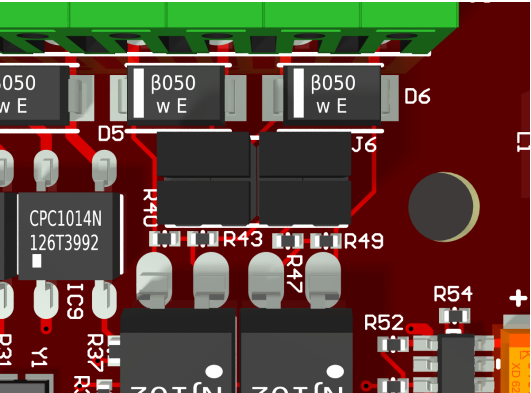
There is the possibility of resetting Codespot via the DTR pin of the serial interface. To do this, connect pins 3 and 4 of the jumper field J5 with a plug-in jumper, as demonstrated in the picture.



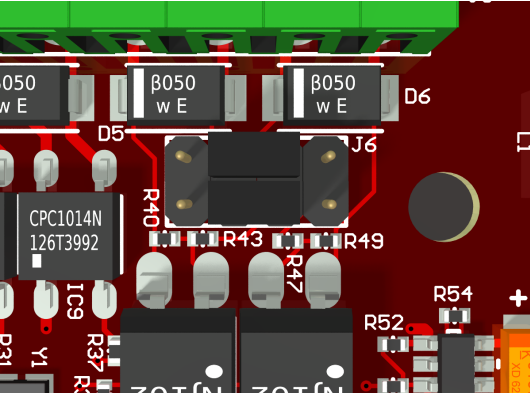
7.9 Configure jumper for digital inputs

On delivery the inputs are configured as contact inputs with a common negative terminal. If you want to control Codespot Dev with external voltage instead of via contacts then you must change the jumper setting inside the device.

The following picture shows the standard configuration as contact input:



The configuration as a voltage input looks like this:



8 Attachment

8.1 Specifications

Processor and memory	
CPU	VS1063 VS_DSP4 core
Static RAM	96 kBytes
Serial flash	4 MBytes
Memory card	microSD, 4 GBytes, replaceable
USB Interface	
Mechanical	Micro-B
Class	USB 2.0 Device
Function	Card reader and power supply
Digital outputs	
Voltage	0 V to 48 V
Current	400 mA constant, 1 A for 10 ms
Isolation	1500 V _{RMS}
Digital inputs (voltage triggered)	
Voltage	0 V to 48 V, non-polarized
Inactive	0 V to 1 V
Active	3 V to 48 V
Current	0.5 mA to 12.4 mA
Isolation	5000 V _{RMS}
Digital inputs (contact triggered)	
Positive polarity	4.9 V to 5.1 V
Negative polarity	0 V
Current	ca. 0.6 mA
Isolation	None
Indicators	
Supply	LED red
Status	LED red/green, programmable
Audio output	
Mechanical	3.5 mm stereo jack
Usage	30 Ω headphone oder line out (stereo)
Audio input	
Mechanical	3.5 mm stereo jack
Impedance	80 kΩ
Amplitude	max. 2.8 V _{pp}
RS-232 interface	
Mechanical	9-pin DCE, 2-wire
Data format	2400 Baud to 230400 Baud, 8N1
ESD protection	1500 V _{RMS}
Power supply	
Voltage	5 V via USB connector
Max. consumption	70 mA (0.35 W) standalone

	200 mA (1 W) USB transfer
Environmental	
Operating temperature	-25 bis 85 °C (-13 bis 185 °F)
Storage temperature	-40 bis 85 °C (-40 bis 185 °F)
Humidity	5 to 95 %, non-condensing
Approvals	
Safety	PCB UL94-V-0 listed Enclosure UL94-HB listed
RoHS compliance	EU directive 2002/95/EC
Metrics	
Dimensions (L x W x H)	82.00 x 66.22 x 28.00 mm (3.228 x 2.607 x 1.102 in)
Weight	75 g (0.17 lb)
Product identification	
PCB revision	Written in copper on the PCB's backside
Serial number	Barcode sticker

8.2 Assignment of the GPIO ports

Warnig! False programming of the data direction register can lead to destruction of the hardware.

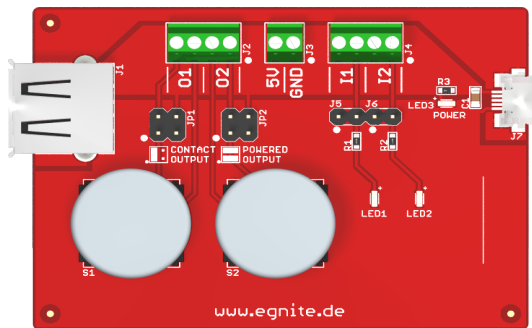
Input-Register	Output-Register	Data direction Register	Application
GPIO_IDATA[0]	GPIO_ODATA[0]	GPIO_DDR[0]	Low output activates the serial data flash, equipped with 100 kΩ pull-up.
GPIO_IDATA[1]	GPIO_ODATA[1]	GPIO_DDR[1]	SPI clock output for serial data flash, equipped with 10 kΩ pull-down.
GPIO_IDATA[2]	GPIO_ODATA[2]	GPIO_DDR[2]	SPI data input.
GPIO_IDATA[3]	GPIO_ODATA[3]	GPIO_DDR[3]	SPI clock output for memory card.
GPIO_IDATA[4]	GPIO_ODATA[4]	GPIO_DDR[4]	Low output activated memory card.
GPIO_IDATA[5]	GPIO_ODATA[5]	GPIO_DDR[5]	Digital relay output 2.
GPIO_IDATA[6]	GPIO_ODATA[6]	GPIO_DDR[6]	Digital relay output 1.
GPIO_IDATA[7]	GPIO_ODATA[7]	GPIO_DDR[7]	Low output switches LED1 to green, High switched LED to red.
GPIO_IDATA[8]			Equipped with 100 kΩ pull-down.
GPIO_IDATA[9]			Equipped with 100 kΩ pull-down.
GPIO_IDATA[10]			Digital optocoupler input 1.
GPIO_IDATA[11]			Digital optocoupler input 2.
	GPIO_ODATA[8]	GPIO_DDR[8]	Low output activates voltage supply of the memory card.

9 Fittings

9.1 I/O test board Bintest (Artikel EGN300513)

With the I/O test board Bintest you can comfortably test your application for digital in- and outputs.

Bintest has two buttons that are connected to both inputs and two LEDs, which show the status of both the outputs.



10 More information

Find out more about Codespot. Visit us at www.egnite.de.

www.egnite.de | E-Mail: info@egnite.de
Tel.: +49 (0)2305 441256 | Fax: +49 (0)2305 441487
egnite GmbH | Erinstrasse 9, 44575 Castrop-Rauxel, Germany