Contact information:

Mail address:

Elektron ESI AB
Stena Center 1C
412 92 Göteborg
Sweden

Email: info@sidstation.com
Website: http://www.sidstation.com

Fax: +46 31 7728111

Using the unit safely

CAUTION!

- Before using this unit make sure that you have read the instructions below.
- Do not perform any modifications to the unit itself.
- Make sure that you use the correct power supply unit (PSU). If you don’t use the one supplied by Elektron ESI AB, please check that it fits according to the requirements found in the PSU section of this manual.
- Please protect the SidStation from strong electronic discharges. These can make the SidStation malfunction.
- Turn down the volume on your amplifier before switching on the SidStation. On powerup the SidStation can emit a sharp click that could harm your equipment.
- Do not connect or disconnect cables to the unit with the power turned on.
- Never climb on top of, nor place heavy objects on the unit.
- Avoid nuclear explosions. The SID-chip might be harmed.

Contents

Introduction

A word from the creators

Thank you for buying the SidStation synthesizer module. The instrument that brings back the unique sound of the SID6581-chip originally found in the Commodore 64 home computer. Using the full power of the SID-chip combined with innovative synthesis techniques it gives musicians access to a whole new world of exciting sounds. We have put many hours of hard work in this product, a synthesizer that reintroduces the spirit and sound from the 80’s.

By using the famous “MOS 6581 Sound Interface Device” chip also known as the SID-chip from the Commodore 64 computer, we bring you a classic sound, which can not be reproduced by any other music equipment. Integrating it into the SidStation unleashes the full potential of this chip and puts you in control. Freeing it from the limitations of the C64 architecture and providing new methods to control the SID sound gives possibilities yet to be explored.

"The SID is the classic synthesizer that never had a case built around it."

The secret of the SidStation sound is only partially explained by the actual SID6581 chip. The SidStation operating system stretches the chip to its limits by updating all SID6581 parameters up to 200 times per second. This implies results of a previously unheard nature.

We in the SidStation design team wish you good luck with your new instrument and hope that it will inspire you and your music.

The MOS6581 SID-chip

In 1983, Commodore released the Commodore 64. It was a home computer based around the 6510 microprocessor and a couple of custom chips. It soon became very popular and played a major part in shaping the computer gaming industry. During the development of the Commodore 64, Bob Yannes (who later founded Ensoniq) was approached by Commodore to help develop a custom sound chip for their home computers. He created possibly the most flexible sound system ever to be used in a home computer. The SID-chip is capable of three-note polyphony, with one programmable filter, ring modulation and oscillator hard synchronisation. Each channel can use one of 4 waveforms: Triangle, Sawtooth, Pulse and Noise (it was soon found that the Triangle and Pulse waveforms could be combined to a different strange sound – in the SidStation this waveform is called “Mixed”). But the characteristics of the SID-chip are much more than can be seen in specifications. It’s wonderful lo-fi distorting filter, beautiful pulsesweep and dramatic ringmodulation effects – it all has to be experienced.

Features

- MOS 6581 synthesis
- Powerful SidStation OS operating system
- Special arpeggiator
- Unique waveform and notenumber table function
- Flexible LFO modulation system
- Hands on modulation capabilities through direct controllers
- External audio in connector routed through the filter
- C64 SID song player support

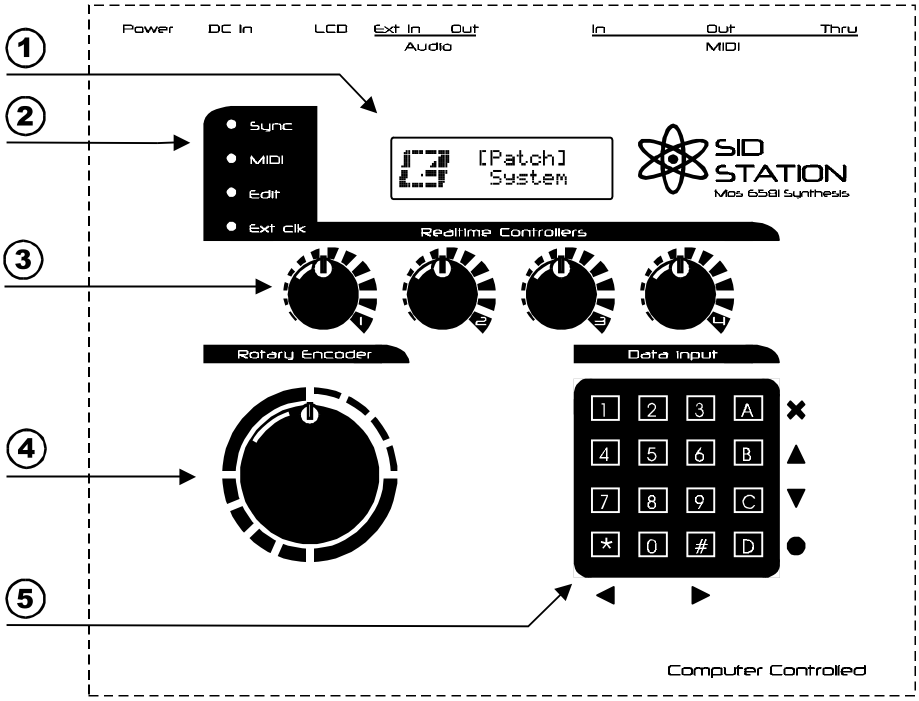
Conventions in this manual

- When a button or knob is discussed the name is enclosed in brackets with bold style. For instance the real-time control knob 1 is referred to as **[KNOB 1]**.

- When a specific mode is discussed the name is written in italic and bold style. The ***play mode*** is an example of this.

- Parameters in the LCD menus are written in bold style, like the parameter **sync**.

A brief overview of SidStation



The SidStation interface

① LCD

The backlit liquid crystal display (LCD) is your source of information of what's happening in your SidStation.

② Indicator LEDs

- **Sync** – Indicates the current update speed of the SidStation. Each blink represents one hundred updates.
- **MIDI** – Indicates incoming MIDI note events
- **Edit** – This LED indicates that a patch is currently being edited.
- **Ext clk** – Indicates when the SidStation is slaved to external MIDI clock.

③ Real-time control knobs

These knobs are used for changing parameters both in edit mode and in play mode. By turning a knob clockwise the corresponding value on the LCD increases. Doing the opposite decreases the value. This allows quick and easy editing in the SidStation menu system. In this manual we refer to them as **[KNOB 1]**, **[KNOB 2]**, **[KNOB 3]** and **[KNOB 4]**.

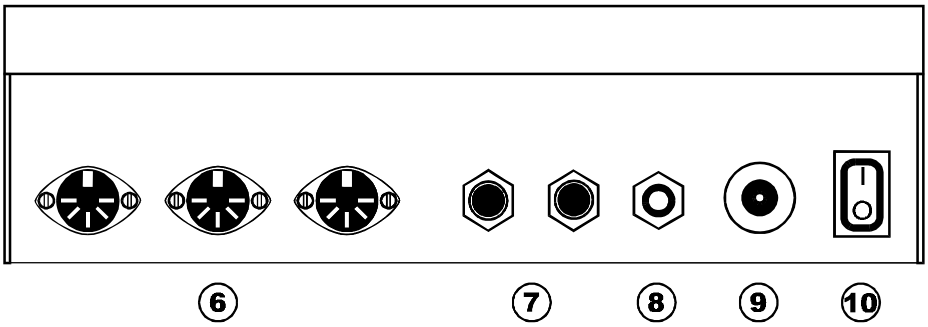
④ Rotary encoder wheel

Used to stepwise increase or decrease a value. By turning the rotary encoder one step clockwise the value is increased by one unit. Turning it counter-clockwise decreases the value by one unit. In this manual we refer to it as the **[ENCODER]**.

⑤ Keyboard

This is where most data input is done. The symbols on the right side and below the keyboard indicate the directions and functions that happen when the corresponding key is pressed. The cross (A key) is the symbol for **[EXIT]** or “up one level”. The **[UP]** and **[DOWN]** (B and C key) arrows allows a value to be increased or decreased. In some menus a blinking cursor is shown. Use the **[LEFT]** and **[RIGHT]** (* and # key) arrows together with the **[UP]** and **[DOWN]** arrows to move it around (note: some menus only allow left and right movement). The dot (D key) is the symbol for **[ENTER]**. The number keys have different functions in different areas of the SidStation OS. Please see section “Getting around and basic editing” for more information.

Connections at the rear panel



⑥ MIDI connectors

MIDI Thru – Connect this output to the next keyboard in your MIDI chain.
MIDI Out – Connect this output to the MIDI In port of your sequencer or keyboard.
MIDI In – Connect this input to the MIDI Out port of your sequencer or keyboard.

⑦ Audio connectors

Out – Line level audio output to connect to your mixerboard or amplifier.

Ext in –Accepts an unbalanced line level signal that is routed through the SIDs filter.

WARNING: Never connect any signals to the “Ext in” input that can be carrying hazardous voltage (signals of too high voltage). This may damage your SidStation and possibly harm the SID chip. “Ext in” is designed for normal line level signals.

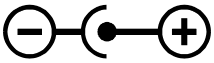
⑧ LCD Contrast

This knob controls the contrast of the LCD. Please set the contrast to an appropriate value. The display is a “super-twist” model, which means that it has a wider observer angle to maximise visibility. The contrast should be set so that no “ghosting” appears from inactive character cells.

⑨ Power connector

Connect the DC power adapter here. Check the specifications below carefully if you fit a PSU not supplied by Elektron ESI.

The PSU should be stabilised, deliver a voltage of 6V up to 7V DC and be able to supply at least 400mA to power the SidStation. The connector has a diameter of 2.1 mm. The connector should have the positive voltage on the inner core, and ground on the outer ring. Please check that this applies to your PSU before powering up the SidStation as both this standard and the opposite exists.



WARNING: Using an improper type of power adapter may result in malfunction of the SidStation and could even damage it. Please consult your dealer if you are unsure of which kind of adapter you should use.

⑩ Power switch

Turns your SidStation on or off.

Care instructions & important notes

To ensure many years of troublefree operation some advice about taking care of your SidStation could be in place:

- Never use any aggressive cleaners on the casing or display. Remove dust, dirt and fingerprints with a dry soft cloth. If needed, damp the cloth slightly with water to remove more persistent dirt.

- Never use sharp objects near the display to avoid scratches or damage. Also avoid applying any pressure to the display itself.

- When transporting the SidStation from one place to another please use the box within which it was shipped (with padding) or use equivalent packaging.

About the memory

The data contained in the SidStation memory may be corrupted or lost during a repair, upgrade or after a temporary malfunction. To be sure that your sounds (patches) are preserved in these cases, please use the SYSEX patch dump feature to transfer the data over MIDI. No liability is assumed for lost data by Elektron ESI AB.

About noise

The SID6581 chip is designed unlike any other chip. This design has given the SidStation a unique sound. The price to pay for this special design technology is that the noise-level unfortunately is quite high. We have done all we can to help to keep the noise-level down, and compared to the original Commodore 64 it is dead silent. What is left is the noise generated by the SID-chip itself.

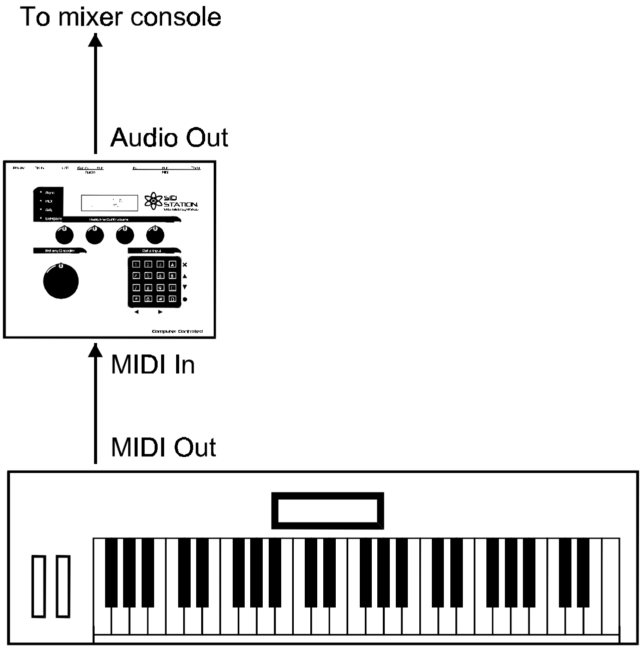
There are several different sources of noise. The most distinct source is the oscillators that never keep totally silent, even when the envelope is done. This means that after the sound is supposed to have died out it can still be heard in the background. Using the gated envelope will cure this bug, but will introduce other strange SID effects (see the oscillator section for details).

Another source of noise is that the address and data bus leaks noise into the audio output. To restrict this problem we have put the SID-chip on an isolated address/data-bus. However, we still have to update the chip, so there are some noise from this source. To locate this type of noise, pump up the volume, vary the update speed from low to high, and you will possibly hear a slight pitch change.

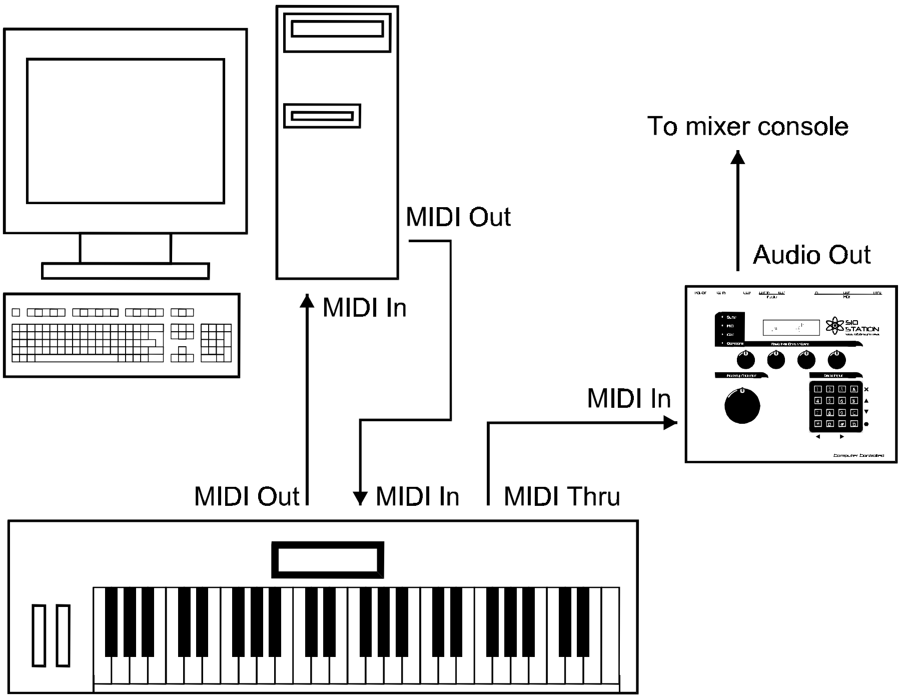
For best results we recommend using a noise-gate, such as the Behringer Autocom.

Quick start

This is a very simple MIDI setup that allows you to quickly get started with your SidStation:



The following setup could be used to connect the SidStation to a computer sequencer:



Note: The computer sequencer must be configured to echo incoming MIDI data. Please consult your sequencer documentation for more information.

Using the SidStation

Getting around and basic editing

The SidStation is controlled using three different types of controls. The keyboard can be used to do everything, but realtime controllers and a rotary encoder have been added to make sound editing smoother.

The menu system is made up of a hierarchy of menus and parameters. Use the arrow keys (**[UP]**, **[DOWN]**, **[LEFT]** and **[RIGHT])** to direct the focus on the display. The current focus is indicated with flashing brackets or a flashing arrow. By pressing the **[ENTER]** key you select the item and enter a new sub-menu. To exit to a higher level in the menu system you press the **[EXIT]** key.

To change a value of a parameter, select it by moving the focus over it and press the **[UP]** key to increase the value or **[DOWN]** to decrease it. The **[ENCODER]** wheel can also be used to change parameter values in steps. Turn it clockwise to increase or counter-clockwise to decrease.

If any values on screen, some or all of the four realtime controllers can be used to set a new value. The values are assigned from left to right, giving the **[KNOB 1]** access to the first parameter, **[KNOB 2]** to the second and so on for the number of values presented on screen.

The numbers on the keypad can be used for quick access to menus. Pressing 1 moves the focus to the first option and selects it. Pressing 2 moves to the second option and select it etc.

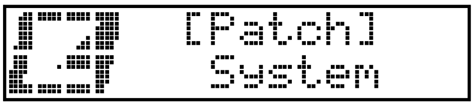
Some menus are two screens wide. This is represented by a small animated arrow that points in the direction the next screen is located. Simply move the focus in that direction to access the screen.

Zero menu selection

While editing an oscillator in single mode, or an LFO, it is possible to quickly access other menus by using the zero menu selection feature. As an example we can say that we are currently editing the oscillator 1 waveform. You can now move to the waveform menu of another oscillator by using the zero menus. First press the “0” key on the keypad and the screen will show rotating directional arrows. Now you can press either arrow right/left for accessing the next or previous waveform oscillator menu. You can also press keys one through three to directly access a specific oscillator. This work equivalent for LFO menus, only difference is that there are four LFO menus in total.

The menu system

When you switch on your SidStation you are greeted by a short welcoming message and then the following screen appears:

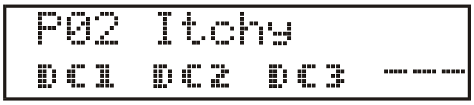


[Patch]
System

This is the top-level menu from where you start exploring the SidStation. On this screen there are three menu items to choose from, controlling three very different aspects of the SidStation. Choose using the **[UP]** and **[DOWN]** keys. Press **[ENTER]** to enter the menu.

Patch play mode

This is where you select and edit a patch in the SidStation. The patch is a set of parameters which can be named and stored in the SidStation memory bank. The parameters control the nature of the sound that is played on the audio output when MIDI data is received. In the SidStation memory you can store up to one hundred different patches. The patch selection screen is shown below:



```
 P02 Itchy
 DC1 DC2 DC3 ---
```

The **[UP]** and **[DOWN]** keys along with the **[ENCODER]** wheel lets you browse the patches stored in memory. To select a patch press the **[ENTER]** key. A patch that is selected but not yet activated with **[ENTER]** will flash.

The four values at the bottom of the screen are user assignable parameters called direct controllers. This allows realtime control of up to four parameters that can be selected by the patch programmer. The SidStation knobs are mapped in the following order:

[KNOB 1] – DC1
[KNOB 2] – DC2
[KNOB 3] – DC3
[KNOB 4] – DC4

When a patch is selected you can play it by sending MIDI notes from your keyboard or other MIDI compatible equipment. The SidStation is now in ***play mode***.
By pressing **[RIGHT]** you enter the ***edit mode*** where you edit the parameters of the patch currently playing. Please see the “Programming the SidStation” section for more information on how to edit a patch.

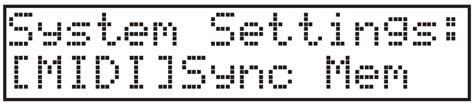
> **Note:** When changing patches there may be a short loud static noise. This is normal and is a result of changing the filter settings between different patches. This is not because of the SidStation, but depends on the way the SID-chip is designed.

Direct Controllers and MIDI

The four direct controllers (DC1 to DC4) all send and receive MIDI control data. They are mapped to MIDI control number 16 through 19. Direct controllers that are not used (shows as --- in the patch selection menu) still output MIDI control data for the corresponding MIDI control number. This means that the SidStation also can be used as an external control box for other synthesizers.

System parameters

The system menu screen contains the parameters that control MIDI, synchronisation and memory integrity. It is accessed from the top-level menu and is seen below:

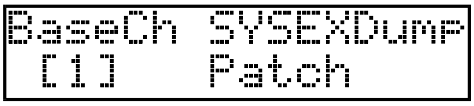


```
System Settings:
[MIDI]Sync Mem
```

The system parameters are divided into three submenus called **MIDI**, **Sync** and **Mem**.

MIDI settings and SYSEX patch dump menu (MIDI)

The **MIDI** submenu lets you set the MIDI base channel of your SidStation and also control the SYSEX patch dump function:



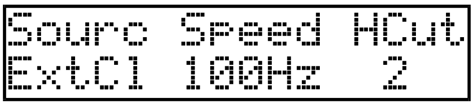
```
BaseCh SYSEXDump
 [1]   Patch
```

By adjusting the **BaseCh** value you set the MIDI channel which is monitoring incoming MIDI data. If your connected MIDI equipment sends note data to your SidStation on channel 1, you must set the base channel to 1. Any other data on channels 2-16 is ignored. The MIDI thru port echoes all incoming data.

The **SYSEXDump** switch controls the system exclusive (SYSEX) patch dump functions. It is easy to make a backup of your patches in your SidStation to a computer sequencer or a MIDI data recorder. See section "SYSEX patch transfers" for more information.

Synchronisation menu (Sync)

The **Sync** sub-menu controls how the SidStation is being updated.



```
Sourc Speed HCut
ExtCl 100Hz  2
```

Sourc sets the source for the update frequency. There are three choices:

- Patch - the update frequency and HCut settings are taken from the local patch settings.
- Globl - the global settings for Speed and HCut found at the right of this screen are used for all patches.
- ExtCl - the update speed is locked to external MIDI clock (when available). HCut setting is taken from the global value found to the right on this screen.

When locked to an external clock, the SidStation makes 48 updates per quarter note (2 updates per MIDI clock). This can be used to time various parts of the SidStation to the external beat. The arpeggiators, square LFO, sample & hold LFO and oscillator delay are obvious targets for such timings. But actually all sound updating parts of the SidStation (except for the VCA envelope will be locked to the external clock. See small table below for various note signatures:

Desired note length	Speed value
½	96
¼	48
1/8	24
1/16	12
1/32	6
1/64	3

If the clock source is set to "Globl" then the **Speed** parameter sets the timing. The timing of the SidStation arpeggiator (as well as all other functions in the SidStation) is then directly derived from the **Speed** parameter. A speed value is given in the unit Hertz, which is how many times per second the sound settings are updated. A value of 100 means that the sound parameters are updated 100 times per second, and an arpeggiator speed value of 20 would mean that the arpeggiator is updated 5 times per second.

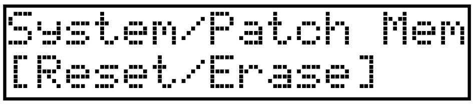
The "Hard cut" **HCut** parameter controls a function that compensates for a bug in the envelopes of the 6581 sound chip. When a new note is triggered, the first few milliseconds of the envelope sometimes are left out. This means that the notes played sometimes suffer from a slightly bad timing. With the **HCut** setting you can choose to force the oscillators to keep silent for a certain amount of time when a new note is triggered. With a setting high enough the envelope bug will not be noticed. However, this function will make the SidStation to react more slowly to new notes (often referred to as "lag"), so this function may not be appreciated in all situations. But when working with a sequencer this is no problem as you can program the sequencer to send notes to the SidStation a certain time in advance. The time the envelope will be kept quiet can be calculated from the following formula: ***Time = HCut/speed*** *(s)*

We recommend a setting of around 20ms. (HCut = 2 at Speed = 100 Hz, or HCut = 4 at Speed = 200 Hz)

For patches using the “gate”-envelope, there is no need for HCut The gate will override the SID envelope, and the related bug will not be present anymore.

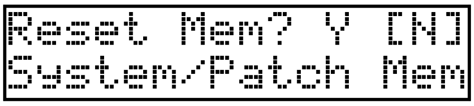
SidStation system/patch memory control menu (Mem)

The **Mem** sub-menu is where you restore the original factory settings:



```
System/Patch Mem
[Reset/Erase]
```

This is a non-recoverable operation that will reset all settings to default, also erasing all patch data. This is very useful if you’re experiencing problems with your machine. This could also need to be done after some system upgrades. Also, if you have problems with the SidStation locking up when entering the patch menu this could mean that the memory is corrupted and a reset is needed. If you decide to proceed with this operation by pressing **[ENTER]**, you are prompted with the question:



```
Reset Mem? Y [N]
System/Patch Mem
```

Selecting “Y” and pressing **[ENTER]** will restore the factory settings from ROM. To abort and keep the current data in memory simply select “N” and **[ENTER]**.

Note: Since SidStation only stores patches in RAM no patches will be present after a memory reset operation. To restore the factory patches you will have to download them online from www.sidstation.com and transfer them by SYSEX dump to your SidStation.

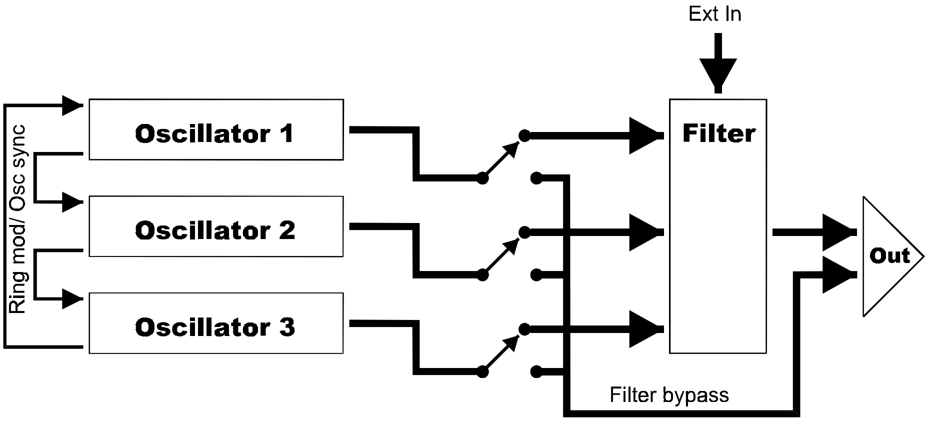
The ASID C64 song player

This selection sets up the SidStation for direct access of the SID chip. This allows the ASID software (or compatible) on your computer to play Commodore 64 SID-songs directly on your SidStation. After selecting ASID the screen will go blank and the SidStation waits for data from host computer. Press a key on the keypad to exit this mode and restart the SidStation. Please refer to the documentation that is included in the host application for your computer platform for more information.

Programming the SidStation

Anatomy of the synthesis in SidStation

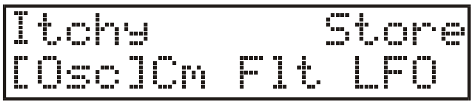
The following picture shows the synthesis architecture in SidStation:



In this manual we speak of an oscillator as a complete set of synthesis components. This means that inside every oscillator we have a tone generator, envelope, pulse width modulation and pitch settings among other things. If the above picture and synthesis terminology seems alien to you, don't be alarmed. Everything will be explained, as we now take an in-depth look at the programming of the SidStation.

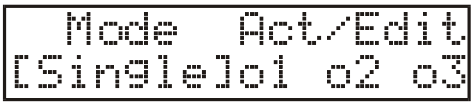
Patch editing

To start editing a patch in SidStation simply press **[ENTER]** in ***play mode***. See section "The menu system" for more info about the ***play mode***. The Edit LED will light up and the following screen is shown which is the ***edit mode*** screen:



```
Itchy      Store
[Osc]Cm Flt LFO
```

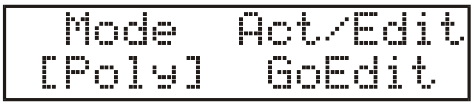
You are now ready to start making changes to your patch. First we enter the **Osc** sub-menu to set the mode for the patch. This is where the three oscillators in your SidStation can be edited:



```
  Mode  Act/Edit
[Single]o1 o2 o3
```

The current mode selected is the ***single mode***. In this mode the SidStation is monophonic. This means that when you play a note all oscillators are trigged simultaneously to form a three-layer patch. All oscillators can be edited individually. This is accomplished by moving the focus to the desired oscillator and pressing **[ENTER]**. A single oscillator can also be deactivated to make a 1 or 2-layer patch. To do so, move the cursor to it and press **[DOWN]** to deactivate it (activate it by pressing **[UP]**).

The second oscillator mode is the ***poly mode***:



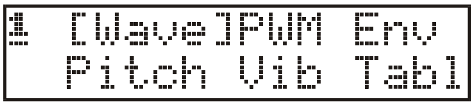
```
  Mode  Act/Edit
 [Poly]  GoEdit
```

This is a three note polyphonic mode. In this mode up to three notes can be played simultaneously or in sequence without interrupting notes playing. This allows you to form three-note-chords and other polyphonic musical structures. To edit the oscillator simply move the cursor to the **GoEdit** sub-menu and press **[ENTER]**. A poly-sound is made from only one oscillator with three-note polyphony.

> **Note:** In the following text we use ***single mode*** as basis for our discussion as the difference to ***poly mode*** when editing is very small. In ***single mode*** we edit up to three oscillators but in ***poly mode*** only one.

Oscillators

The oscillator edit menu is shown below:

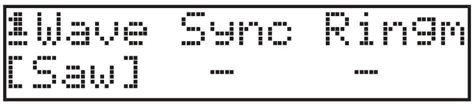


```
1 [Wave]PWM Env
  Pitch Vib Tabl
```

To determine the oscillator currently being edited please look at the underlined number in the upper left corner of the screen. In the case above we are editing oscillator 1. Every oscillator has 6 sub-menus with parameters. We will now explain them one at a time:

Waveform (Wave)

This is where you select the waveform for the current oscillator and edit related parameters:



```
1Wave Sync Ringm
[Saw]  -    -
```

The first parameter is the waveform. The available waveforms are:

- Triangle (Tri)
- Sawtooth (Saw)
- Pulse (Puls)
- Mixed (Mix)
- Noise (Nois)

The waveform is the essential building block of the synthesis in SidStation. Every type of waveform has its own distinctive timbre.

For all waveforms, synchronisation and ringmodulation can be selected to make the current oscillator coupled to the previous one. We call the previous oscillator *the carrier*.

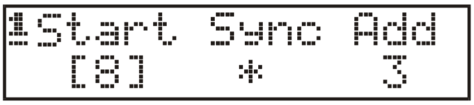
Synchronisation is an effect better experienced than explained. It restarts the modulated waveform every time the carrying waveform crosses zero. This however says nothing about what it really sounds like. Try it out by enabling the switch **Sync**.

Ringmodulation is the product of two signals, the carrier wave and the modulator. If the switch **Ringm** is set the current oscillator will modulate the previous one. So if ring-modulation is switched on for oscillator 2 then it will ring-modulate oscillator 1. Oscillator 1 modulates oscillator 3. Please see the figure in the “Anatomy of the synthesis in SidStation” section above to find out how the oscillators are connected.

Triangle is usually the waveform that produces the most audible results, both for synchronisation and ringmodulation (but do experiment with other waveforms as well – the ringmodulator can do some wild things with the mixed waveform). Also, try different pitch settings. The effect of the synchronisation / ringmodulation varies very much depending on the pitchlevel of the two coupled oscillators. The best effects are often found when changing the pitch of the modulating oscillator in realtime (with pitchbend or LFO).

Pulse Width Modulation (PWM)

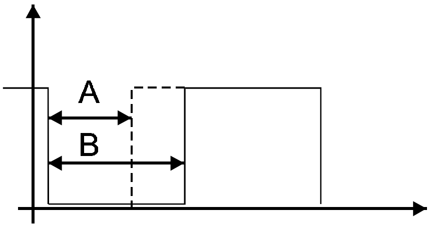
This is the first screen of the menu that controls the pulse width modulation:



```
1Start Sync Add
  [8]   *    3
```

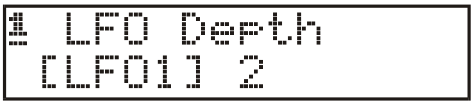
The most characteristic waveform of the SID-chip is probably the pulse sound. Its shape can be varied wildly from a needle-sharp thin sound to full rich square. The parameter that control the basic setting of the pulse width sound is **Start**. Try first setting **Sync** and **Add**=0. You can now vary the pulse width from the most sharp pulse sound (1) to square (64). This is all very well, we can vary the type of pulse sound, but wouldn't it be fun to have the pulse animated from sharp to square? To do this, set the start parameter to 1 and increase the add value to around 3. Let the note play a while and notice the difference.

The **Sync** switch stands for *Synchronise to note on*. If set it will restart the pulse settings to the start value each time you press a new note. If not selected it will run through the pulse values without regarding new notes. This renders the start value obsolete, as the pulse values will never restart from it.



The figure above shows a theoretic visualisation of a pulse sound. The A-pulse shows a sharp pulse sound and B a square. If we start with a pulse like the one in A (**Start**=16) and set add to a value higher than zero it will gradually turn into a more square-like shape, and then become sharper again. This makes the sound animated and "alive".

Moving the focus to the extreme right accesses the second screen that holds the pulse width modulation LFO control settings:



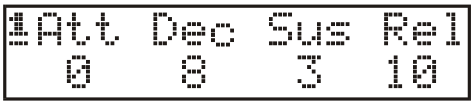
```
1 LFO Depth
 [LFO1] 2
```

The **LFO** parameter lets you choose an LFO to modulate the pulse width. Any of the four LFO's in SidStation can be used. The **Depth** value sets the modulation depth (0 to 127).

Note: For any of the above settings to have any effect, you must select Pulse as the waveform of the current oscillator. The pulse settings will also affect the Mixed waveform, but the effect can not be explained logically (the sound gradually disappears with higher pulse width settings).

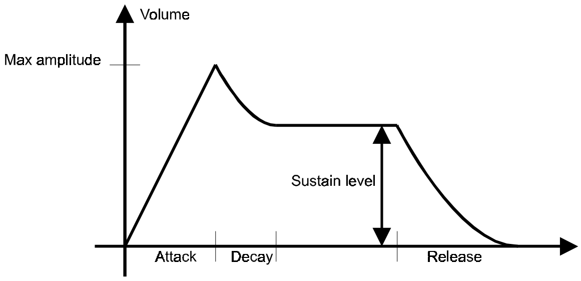
Envelope (Env)

The first screen of the parameter menu for the envelope is seen below:



```
1Att Dec Sus Rel
  0   8   3  10
```

The envelope controls the variation of the oscillator volume over time. A string sound fades in with a slow attack but a piano or a guitar string has a fast attack time. In this manner we can control the characteristics of our sound. This can be described by the figure below:



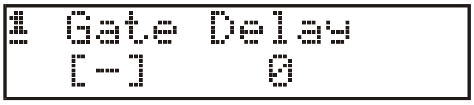
The parameters are:

A (Attack): The time the envelope takes to reach full level.
D (Decay): The time it will take to reach the sustain level.
S (Sustain): The volume level to keep when a key is held.
R (Release): The time it takes for the sound to fade out after the key is released.

All parameter values are defined from 0 to 15.

Value	Attack Rate	Decay/Release Rate
0	2 ms	6 ms
1	8 ms	24 ms
2	16 ms	48 ms
3	24 ms	72 ms
4	38 ms	114 ms
5	56 ms	168 ms
6	68 ms	204 ms
7	80 ms	240 ms
8	100 ms	300 ms
9	240 ms	750 ms
10	500 ms	1.5 s
11	800 ms	2.4 s
12	1 s	3 s
13	3 s	9 s
14	5 s	15 s
15	8 s	24 s

Moving the focus to the extreme right accesses the second screen:



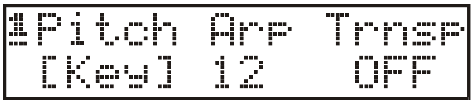
```
1 Gate Delay
  [-]      0
```

The **Gate**-switch, when active, replaces the SID chip hardware envelope found on screen 1 with a software gate with amplitude of the Sustain. Since the SID envelope now is not used this gives perfect timing without any need for the **HCut** delay (se section regarding the System menu and the sync sub-menu). It also keeps the SID chip oscillators totally quiet, reducing the background noise. However, while curing the SID chip envelope bug, it introduces a new peculiar effect, courtesy of the SID chip. A few seconds after a sound is played and the SID is forced to be quiet, it lets go of a "ghost sound". Choose for yourself if you think that this envelope mode is useful.

The **Delay** value makes it possible to delay the output of an oscillator with a number of update cycles. Can be used either to fatten up a layered sound by giving the different sounds a slight delay, or it can be used for "echo effects" found on some C64 game tunes using greater values.

Pitch

The pitch settings are controlled from the two-screen menu below:



```
1Pitch Arp Trnsp
 [Key] 12   OFF
```

All oscillators can be tuned separately (in ***poly mode*** there's only one oscillator to edit). The pitch can be locked to a specific note by the **pitch** parameter. To lock the pitch, select the note you want the oscillator to hold no matter what key is pressed on the keyboard. To make the pitch follow the incoming MIDI data set the parameter to **Key**.

The arpeggiator in SidStation is quite different from the ones found in other synthesizers. This arpeggiator implements the "broken chord" style found in Commodore C64 music. The speed of the arpeggiator is divided down from the frequency set in the system menu or, if selected, relative to the patch. Different arpeggiator speeds can be set for the three oscillators. It is also possible to synchronise the arpeggiator to external MIDI clock messages using the following table:

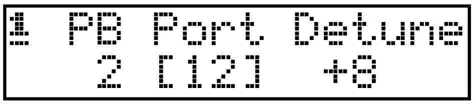
Desired note length	Speed value
½	96
¼	48
1/8	24
1/16	12
1/32	6
1/64	3

Note:

- For the arpeggiator to synchronise to external MIDI clock, the system sync source must be set to **ExtCl**. Please see section "The menu system" for more info about the system settings.
- The order the chords are pressed on the keyboard controls the order they will be replayed. As long as one note is held down the arpeggiator is held. This is made to help simple chord changes without restarting the arpeggio (in other words: feature – not bug).
- The arpeggiator have no function in poly mode.

The pitch is transposed by the **Trnsp** parameter. Transpose allows you to move the pitch range of your oscillator up or down. For example, this can be used to let an oscillator act as a sub-oscillator when making bass sounds so that the sound gets more depth. The transpose interval is –24 to +24 semitones.

By accessing the second screen to the right you can edit the pitch wheel and portamento settings:



```
1 PB Port Detune
   2 [12]  +8
```

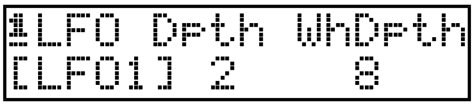
The range of the pitchbend wheel is controlled by the **PB** parameter. This parameter gives the interval of the wheel. If the **PB** parameter is set to 12 then the pitch can be varied from –12 to +12 semitones by the pitchbend. The maximal pitchbend range is –24 to +24 semitones.

Portamento controls the sliding between two consecutive notes played. A slow portamento speed gives the effect of a slow pitch-bend up to (or down to) the next note. The portamento speed is controlled by the **Port** parameter. The portamento of the SidStation is the same that can be found on for example the TB-303, e.g. it uses the same amount of time to slide between any two notes.

The fine tune is controlled by the **Detune** parameter. Detune is useful when the oscillators are “too much in phase” with each other and the result sound dull. To avoid this and make the sound more alive, detune one or two oscillators a bit. The interval is –63 to +64. The maximum settings moves the pitch one half note higher or lower.

Vibrato (Vib)

Vibrato is used to give the sound a bit more “expression”. This can be explained by listening to a violinist playing. On some notes he or she vibrates the tone so that some more “feeling” is introduced. In SidStation we do this by using the vibrato settings. Using more extreme vibrato-settings crazy video game sounds can easily be done.



```
1LFO Dpth WhDpth
[LFO1] 2     8
```

First you have to select which LFO of the four available in SidStation that will generate the vibrato modulation. The **LFO** parameter selects this. Please see the “Low Frequency Oscillator” section for more information on the LFO in SidStation.

The **Dpth** parameter controls the depth of the vibrato. The range is 0-127.

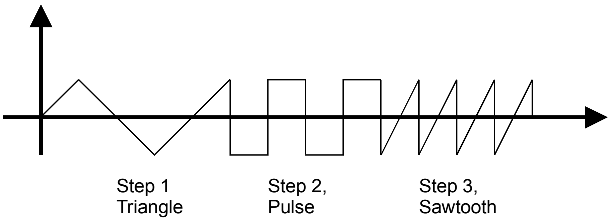
The **WhDpth** parameter sets the amount of which the modulation wheel should add to the **Dpth** value. The range is 0-127.

Waveform table (Tabl)

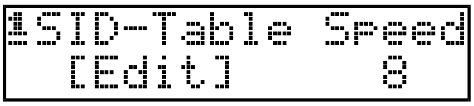
A waveform table is a concept unknown to most people unknown of music creation techniques of the Commodore 64. It is a bit similar to what is used in the Wave-synthesizers by Waldorf, but with the SID it is usually not used to do sweet sweeping pads. SID tables are often harsh transitions for dramatic effects. The difference from "wave sequences" known from the Korg Wavestation, is that the SidStation tables are usually played through much quicker to create percussive effects. Or, it can be used at a lower speed as a conceptual sequencer, with control over waveform, synchronisation, ringmodulation notenumber (transpose). Viewed on the whole, it sounds very different from what is known from other synthesizers.

If you’ve searched for the secret on how to do those Rob Hubbard effects from classic Commodore C64 games like Commando or Delta, or how the cool drums of SID-songs are made - tables are the answer. It takes some practice to master the technique, but the results can be radical.

So what is a table? As a simple description it is a table of waveforms and frequencies that are replayed quickly, one at a time from beginning to end. Either as a one time shot at the beginning of a note on, or as loop. Take a look at the following figure for a simple example. A table in SidStation has 65 steps that contain the data much like a sequencer. Each oscillator in SidStation has an individual table so in ***single mode*** you can play up to three tables simultaneously.



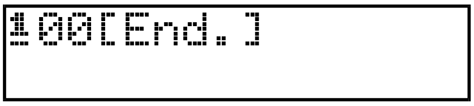
The table menu is shown below:



```
1SID-Table Speed
   [Edit]      8
```

The **Speed** value determines the speed of the table when played. If the **Speed** value is Off then the table is not active. Like in the case of the arpeggiator the table speed can be synchronised to MIDI clock messages. See the section “Oscillators, Pitch” for a speed table and general considerations when using external sync.

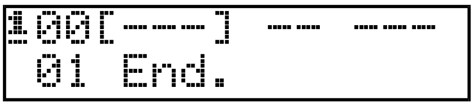
To start edit a new table we enter the sub-menu called Edit:



```
100[End.]
```

This is the editor for SidStation tables. Use the arrow keys to move around. Use the numbers from one and up to enter data. Press **[ENTER]** to insert and the “0” key to delete one step. The knobs have no function in this menu. The list is longer than the screen, so use **[UP]** and **[DOWN]** to access steps outside the screen. A table can have a maximum of 32 steps.

Press **[ENTER]** to insert a new step:



```
100[---] --- ---
 01 End.
```

The number to the left is the current step index (0...31). As you can see each step has four focus positions. Move the focus from one to another by using the **[LEFT]** and **[RIGHT]** arrows.

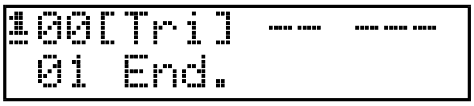
The first focus position can either be a waveform or a Loop/End command. To insert a command we make use of the keypad with the following mapped keys:

Key	Command	Description
1	Tri	Triangle waveform
2	Saw	Sawtooth waveform
3	Puls	Pulse waveform
4	Mix	Mixed waveform
5	Nois	Noise waveform
6	Off	Output is disabled for this table step
7	---	Waveform is left untouched from the Osc->Wave setting[1]

8	Loop	Loop from step xx (xx is edited in focus position 2)[2]
9	End	Stop playing the table (continue with normal mode)

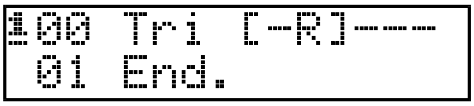
1: Unless the step command Off or a waveform is selected the original waveform for the current oscillator will play. Any changes in oscillator sync, ringmodulation and pitch will affect the original waveform when using tables.
2: A loop can not be initiated from step 00, since this would give an infinite loop.

Select a triangle waveform using the "1" key:



```
100[Tri] --- ----
 01 End.
```

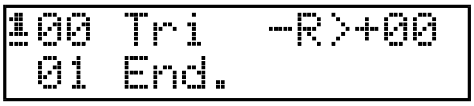
The second focus position lets you either select the loop point or set synchronise and ringmodulation on/off:



```
100 Tri [-R]----
 01 End.
```

If the current step is a loop command then this is where the loop point is entered (only loop points with and index lesser than the current step is accepted). Once the table is played it will play until the loop command and then restart at the index specified by the loop point. Since we have entered a waveform we can select oscillator synchronisation on/off for this step using the "1". The "2" key selects ring modulation on/off for the current step.

The third focus position sets the step note pitch mode using the keypad:

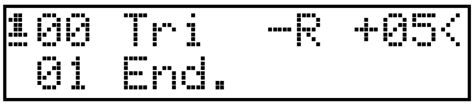


```
100 Tri  -R>+00
 01 End.
```

Key	Mode	Description
1	F	Sets the note pitch to fixed mode[1]
2	-	Subtracts the note pitch in from the base note currently played
3	+	Adds the note pitch to the base note currently played
4	---	No change to the original base note pitch

1: When the F mode is selected, the table step ignores the pitch of the original MIDI note. This is good for fixed drums, and for adding some noise at the beginning of a sound.

The fourth focus position sets the actual note pitch:



```
100 Tri  -R +05<
 01 End.
```

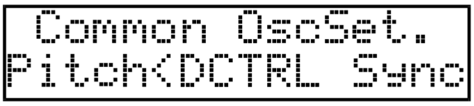
For the fixed note pitch mode (F) you can enter a value from 00 to 81 which will be the note number that will be played for this row, no matter what MIDI key that is received.

For positive and negative values a value from 00 to 47 semitones can be entered. This will be added or subtracted from the current base note pitch when this step is replayed.

This is the whole procedure to program one step in a table. Now it is just a matter of adding more steps to build up your table. As said once before, it takes some practise to master it. See section "Programming tips" for more information about how to use tables.

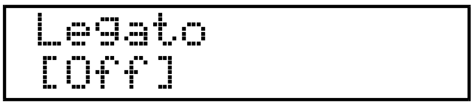
Common parameters

The global patch parameters are located in the common sub menu. To access the common sub menu you have to go back up to the ***edit mode*** screen. The selection **Cm** opens the screen below:



```
 Common OscSet.
Pitch<DCTRL Sync
```

The **Pitch** sub-menu opens up the following screen:



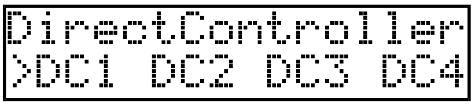
```
 Legato
 [Off]
```

Here you set the legato mode for the notes played.

With legato off, portament o is always active. Legato on means that portamento is only activated when a note is initiated while another note is held. In other words with legato on portamento is only activated for notes played legato.

Direct controls (CTRL)

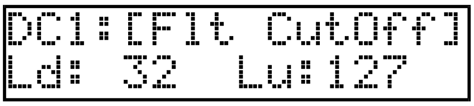
The **CTRL** sub-menu holds the direct control routing information for the four realtime controllers:



```
DirectController
>DC1 DC2 DC3 DC4
```

In SidStation it is possible for the user to assign a direct control destination to a number of parameters in the SidStation synthesis engine. It is then possible to tweak the corresponding realtime controller in patch ***play mode*** to directly control the parameter selected. See the section “The menu system, Patch play mode” for more information about the ***play mode***.

To assign a direct control, simply select one from the screen above to enter its sub-menu:



```
DC1:[Flt CutOff]
Ld: 32  Lu:127
```

The direct controllers will appear in the order 1 to 4 in the ***play mode*** screen. In the screen above we are editing direct control number 1.

The **DC1** parameter is the destination selected for control with direct control 1. "No Control" means that no destination has been chosen for this Direct Controller (will be seen as "---" in the ***play mode*** screen). If No Control is chosen continuos MIDI control data is still output when the knobs are turned, so that the SidStation can be used as an external MIDI control box.

It is possible to limit the output range for a direct control, which can be useful in some cases. An example is a bass sound with a filter cut-off direct control. The patch programmer might want to limit the cut-off so that the user that plays his patch doesn’t turn down the cut-off too much. The **Lu** and **Ld** parameters are used for this.

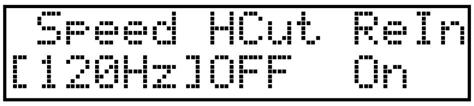
Ld is the lower limit value selected for this direct control.

Lu is the upper limit value selected for this direct control.

Local patch synchronisation (Sync)

The **Sync** sub-menu contains the local synchronisation parameters for the current patch:

Each patch in SidStation can have its own unique synchronisation setting. See section “The menu system, System” for more information about the following parameters:



```
 Speed HCut ReIn
[120Hz]OFF  On
```

Speed sets the local update speed for the patch.

HCut sets the local hardcut setting for the patch.

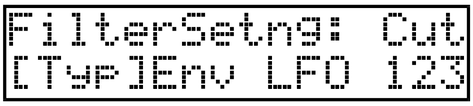
ReIn is a switch that when set for a single-patch forces a new note to be initiated (envelopes, LFO's and tables restarted) for all new notes pressed down – even if the previous wasn't let up before the new one. In Poly-mode with the ReIn-function activated the envelopes for all three oscillators are retriggered for every note on.

> **Note:** The Speed and HCut local sync settings only apply if the sync source is set to "Patch" in the system synchronisation menu. See section "The menu system, System" for more information.

The Filter

A filter can change the timbre of a sound very dramatically. Thumping analogue bass sounds or needle sharp leads can be produced with the help of a filter. The filter of the SID-chip is unlike any other filter.

The filter sub menu controls the filter in SidStation. To access this menu you have to go to the ***edit mode*** screen. The selection **Flt** opens the following screen:



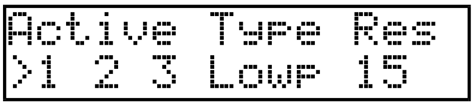
```
FilterSetng: Cut
[Typ]Env LFO 123
```

The filter controls are divided into three submenus called **Typ**, **Env** and **LFO**.

The **Cut** parameter controls the cut-off frequency. This frequency is where the filter will start to affect the sound depending on its selected type. The range is 0 to 127.

The filter type menu (Typ)

To select the actual filter type enter the first **Typ** submenu screen below:



```
Active Type Res
>1 2 3 Lowp 15
```

All oscillators in SidStation are routed through the same filter. The filter can however be bypassed for each oscillator. To bypass the filter for an oscillator simply move the cursor to the desired oscillator number and press **[DOWN]**. Press **[UP]** to enable it again.

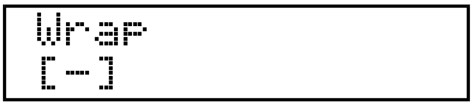
The filter characteristics can be chosen with the **Type** parameter from the following:

- **Lowp** – Low pass filter which cuts frequencies above the cut-off frequency.
- **Band** – Band pass filter that cuts frequencies above and below a region around the cut-off frequency.
- **L/B** – Low pass and band pass combined forming a (sort of) low pass filter cutting higher than just Lowp.
- **High** – High pass filter which only cuts frequencies lower than the cut-off frequency.
- **L/H** – Low and high pass filter combined forming a band-reject filter that removes frequencies around the cut-off frequency.

- **B/H** – Band pass and high pass filter combined forming a (sort of) high pass filter that lets through frequencies lower than just high pass.
- **All** – All filters combined, which does provide some strange filtering yet to be understood by the human being.

Resonance specifies the degree to which the frequencies in the region of the cut-off frequency will be emphased. In SidStation you can set the resonance between 0-15 with the **Res** parameter.

Moving the focus to the extreme right accesses the second screen:

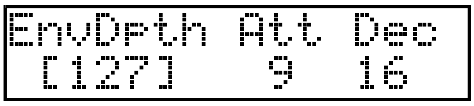


```
 Wrap
 [-]
```

The filter normally doesn't wrap round between open and closed. Driving the cut-off above 127 with a LFO modulation only results in a maximum value of 127. But if **Wrap** is activated the filter will wrap around from highest to lowest value above 127. (Some might say this is not very useful)

The filter envelope (Env)

The filter has it's own envelope which is controlled by the two screens wide **Env** submenu:



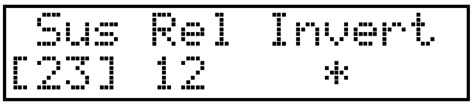
```
EnvDpth Att Dec
 [127]   9  16
```

EnvDpth controls how much the envelope affects the cut-off value. Setting **EnvDpth** to 0 will leave the Envelope unused. An EnvDpth value of 127 and setting the **Cut** (on the main filter screen) parameter to 0 makes the envelope span all over the cut-off range.

Att controls the filter envelope attack speed. A low value gets the envelope up to full value quicker than a higher value.

Dec controls the filter envelope decay speed. A low value gets the envelope from full value to sustain value quicker than a higher value.

Accessing the second screen to the right shows the remaining parameters of the envelope:



```
 Sus Rel Invert
[23] 12     *
```

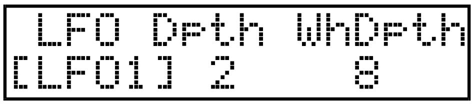
Sus controls the filter envelope sustain value. After the attack and decay phase the filter envelope will stay at the sustain value until the key is released.

Rel controls the filter envelope release speed. After all keys are released the release phase will begin, and the envelope value will slope down to zero.

Invert will invert the filter envelope when activated, the envelope will be subtracted from the cut-off value rather than added.

Filter modulation (LFO)

The **LFO** submenu controls LFO cut-off modulation of the filter:



```
 LFO Dpth WhDpth
[LFO1] 2    8
```

First, select which LFO of the four available in SidStation that will generate the cut-off frequency modulation. The **LFO** parameter selects this. Please see the “Low Frequency Oscillator” section for more information on the LFO in SidStation.

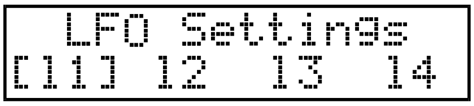
The **Dpth** parameter controls the depth of the vibrato. The range is 0-127.

The **WhDpth** parameter controls how much depth the modulation wheel should add to the **Dpth** value. The range is 0-127.

Low Frequency Oscillators (LFO)

The low frequency oscillators (LFO) are used for parameter modulation. This allows parameters in SidStation to be modulated during the length of which the note is played. A typical example is to modulate the filter cut-off frequency. LFO modulation is simply a variation of a parameter value along with the amplitude of a relatively slow propagating waveform. This is very useful for making evolving changes in the sound as the note is playing. In SidStation you can find one of the most flexible LFO implementation ever made.

The SidStation has four LFOs that can be operated independently from each other. The LFO select sub menu is accessed from the ***edit mode*** screen with the option **Lfo**:



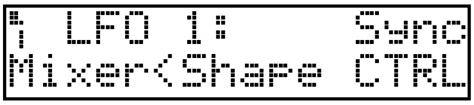
```
  LFO Settings
[l1] l2  l3  l4
```

Here you select edit the LFO. Selecting any of the four options **I1**, **I2**, **I3** and **I4** to edit the LFO of your choice.

Note: The individual LFO menus can also be accessed directly from some of the parameter menus of an oscillator. In the vibrato menu for instance, the vibrato is modulated by a LFO. Pressing [ENTER] on the selected LFO brings you directly to the edit menu of that LFO.
You can switch between different LFO’s by accessing the zero-menu (press 0 on the keyboard). See “Zero menu selection” in “Using the SidStation” for more information.

Editing the selected LFO

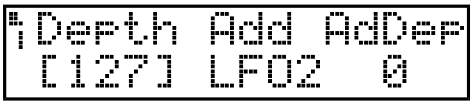
Selecting the LFO you want to edit presents you the LFO main menu:



```
  LFO 1:    Sync
Mixer<Shape CTRL
```

The symbol in the upper left corner indicates that we are currently editing LFO 1. This symbol is present in all the LFO sub menus. These menus are **Mixer**, **Shape**, **CTRL** and **Sync**. The various submenus will be explained one at a time.

The LFO mixer menu (Mixer)



```
 Depth Add AdDep
 [127] LFO2   0
```

The mixer menu controls the mixing of two LFOs. This means that you can add another LFO to the current and create a more complex waveform. The parameters are:

Depth - controls the outgoing depth of the current LFO.

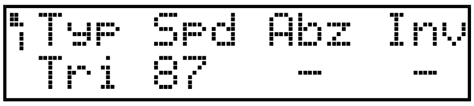
Add - selects the LFO to be added to the current LFO.

AdDep - controls the depths of the added LFO, in this case LFO 1. Setting **AdDep** to 0 means that no other LFO is added (default).

> **Note:** For a walk on the wild side, try adding the output of an LFO to itself! The LFO will start to have a life of it's own.

The LFO shape menu (Shape)

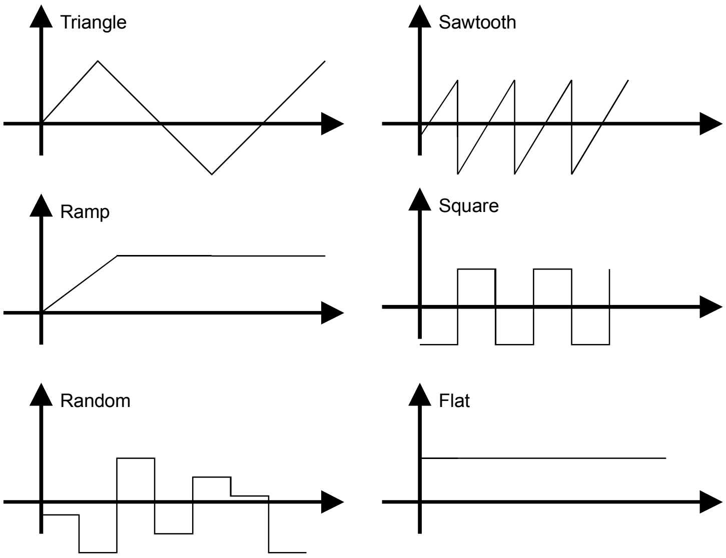
This menu is two screens wide and the first screen is seen below:



```
 Typ Spd Abz Inv
 Tri 87   -   -
```

The LFO base waveform is selected by the **Typ** parameter from the following:

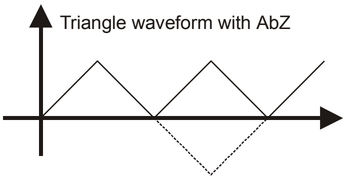
- Tri - Triangle
- Saw - Sawtooth
- Rmp - Ramp
- Sqr - Square
- Rnd - Random
- Flat – LFO output is set to a steady maximum value



The Flat waveform is a non-oscillating flat LFO, which output is always set to a steady maximum. This is useful as a base for external MIDI control messages.

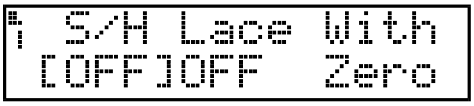
To set the speed of the LFO, use the **Spd** parameter. The value is a divisor from from the system sync speed (see section "The menu system" for more info). A higher value consequently means that the oscillator is slower.

AbZ inverts all LFO output values below zero. The LFO waveform output is forced into the positive domain so that no negative values are allowed which can be useful with some modulation parameters.



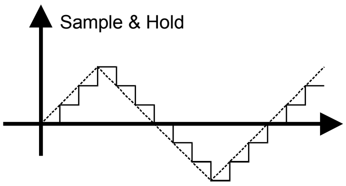
The **Inv** switch selects the inversion of the waveform. For instance a square waveform goes from low to high and back to low. If **Inv** is set the sequence is now high to low and back to high again.

By moving the focus to the extreme right you access the second screen:



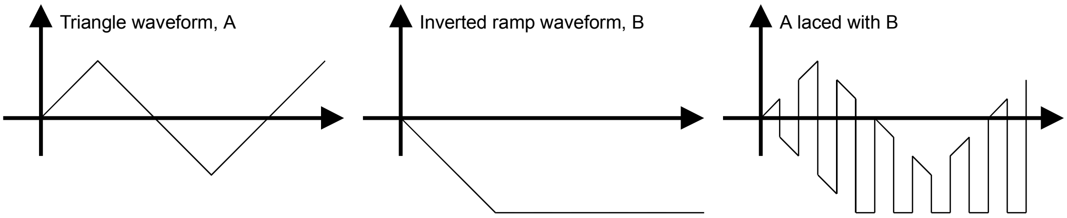
```
  S/H Lace With
 [OFF]OFF  Zero
```

S/H sets the value for the Sample and Hold. Sample and hold holds the output of the LFO fixed for a certain amount of time. A low value holds the value a shorter time. OFF disables the Sample and hold function. See below for an example of sample & hold on a triangle waveform:



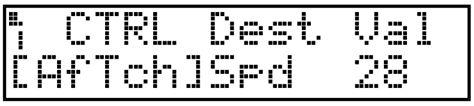
Lace controls the interlace speed of the current LFO output. Interlace is a special feature that allows the current LFO to be interleaved with another source. This is done by rapidly inserting values from another source into the output of the LFO at the rate specified by **Lace**. The result is a non-continuos or “chopped-up” LFO waveform. A low value gives a quick interleave function. OFF disables the interlace function.

-With selects the source of which the current LFO output is interlaced with. The source can be a zero value or the output of another LFO. See figure below for a typical example.



The LFO modulation control source menu (CTRL)

A LFO parameter in SidStation can be modulated by an external MIDI source event.



```
  CTRL Dest Val
[AfTch]Spd  28
```

CTRL sets the source of the LFO parameter modulation. The different modulation sources are:

- ModWh - Modulation Wheel
- PBend - Pitch bend
- Veloc - Velocity
- AfTch - Aftertouch

- CTRL1 to CTRL4 - Mapped MIDI control messages (20 to 23)
- LFO1 to LFO4 - Output of LFO1 to 4 is used to modulate the selected parameter.

Dest selects the destination LFO parameter to be modulated by the external control. The different destinations are:

- None - No external control for this LFO
- Dpth - LFO Depth
- Spd - LFO Speed
- S/H - LFO Sample and hold
- Lace - LFO Interlace rate.

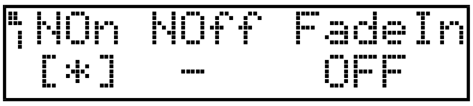
Val sets the maximum value of the external control modulation influence on the selected LFO parameter. A minimal value of the external control sets the original value for the destination.

An example to clear this out a bit:

CTRL=ModWh, Dest=Spd, Val=0 (In CTRL menu) Spd=127 (In Shape menu)

This will make the Modulation Wheel control the speed of the LFO from 127 (slow) (Modulation wheel at zero) to 0 (fast) (Modulation wheel at its maximum).

The LFO synchronisation control menu (Sync)



```
 NOn NOff FadeIn
 [*]  -     OFF
```

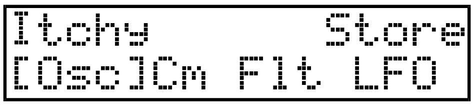
The **NoteOn** switch selects re-synchronisation of the LFO when a new note is played. If sync is not selected the LFO runs independent from any new notes played. The ramp waveform is always synced to Note On, as it would not make any sense to have it non-synchronised

The **NOff** switch controls if the LFO should be restarted each time a new Note Off message is received.

The **FadeIn** parameter controls the time of the LFO fade in effect. This allows a LFO modulation to start at zero and then gradually fade in during the time specified by **FadeIn**. A very useful feature for making automatic smooth transitions on Note On without having to use the modulation wheel to control the LFO depth manually.

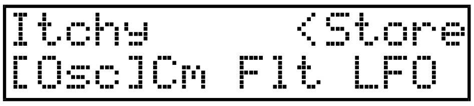
Naming and managing patches

When you are done with your patch editing, it is time to name the patch. This is done from the edit mode screen:



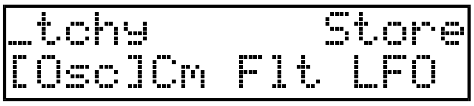
```
Itchy      Store
[Osc]Cm Flt LFO
```

To start editing the patch move the focus up so that that it changes to an arrow pointing right:



```
Itchy     <Store
[Osc]Cm Flt LFO
```

Pressing **[ENTER]** changes the focus marker into the character ***input mode*** cursor:



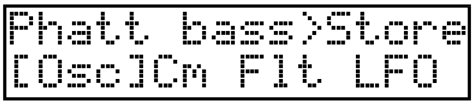
```
_tchy      Store
[Osc]Cm Flt LFO
```

The keyboard and the **[ENCODER]** are used for character input. The **[LEFT]** and **[RIGHT]** arrows moves the cursor left and right. To enter or change a character, turn the **[ENCODER]** wheel or use the **[UP]** and **[DOWN]** arrows. You can also use the number keys on the keyboard (1-9). Each number key has a couple of characters in the alphabet assigned to them. Each time you press the key the character toggles to the next out of the three assigned to this key. Once the lowercase characters has been displayed you can continue to press this key and get the uppercase characters. Note that to input numbers or other non-letter characters you would have to use the **[ENCODER]** or the arrow keys. To enter a space you can use the “0” key.

When you are done exit the ***input mode*** with the **[EXIT]** key

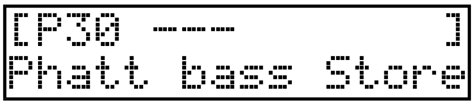
Saving the patch

To save the patch in memory, move the cursor to the store option on the ***edit mode*** screen:



```
Phatt bass>Store
[Osc]Cm Flt LFO
```

Pressing **[ENTER]** once opens up the following screen:

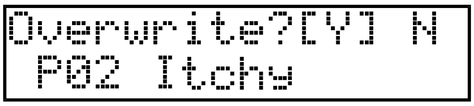


```
[P30 ---       ]
Phatt bass Store
```

The number and name between the flashing brackets is the target memory slot for the store operation. Empty memory slots are marked by three “-“ in a row. To save the patch “Phatt bass” in an empty memory slot press **[ENTER]** once. Then press the **[EXIT]** button to go back up the ***edit mode*** screen.

Overwriting a patch

Overwriting an existing patch in a memory slot is similar the procedure for empty slots, but you get the following request:



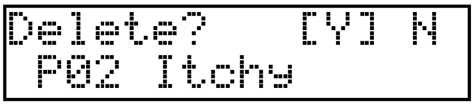
```
Overwrite?[Y] N
 P02 Itchy
```

Selecting the **Y** option with **[ENTER]** will overwrite the patch “Itchy” in memory slot 2. Selecting **N** will abort the overwrite operation.

> **Note:** The overwriting of a patch is a non-recoverable operation. Any patch overwritten is gone forever so be sure that you have saved important patches using SYSEX dump.

Deleting a patch

It is possible to delete a patch and thus free up the memory it is using. This is done from the ***edit mode*** screen by pressing **[LEFT]**. The following screen is then shown:



```
Delete?   [Y] N
 P02 Itchy
```

Selecting the **Y** option with **[ENTER]** will delete the patch “Itchy” in memory slot 2. Selecting **N** will abort the overwrite operation.

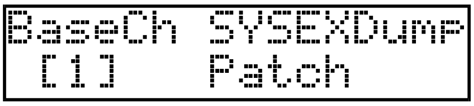
Note: The deletion of a patch is a non-recoverable operation. Any deleted patch is gone forever so be sure that you have saved important patches by using SYSEX dump.

SYSEX patch transfers

SYSEX transfers are ideal for backups of your patch data. It is also a gateway to exchange patches with other SidStation owners around the world by downloading them from the Internet. Visit www.sidstation.com frequently for new patch collections by users and professional sound programmers.

Dumping a single patch from SidStation

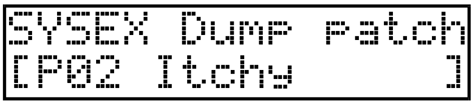
Go to the system menu and select the **MIDI** submenu:



```
BaseCh SYSEXDump
 [1]    Patch
```

Select the dump mode you want with the **SYSEXDump** switch and start the MIDI SYSEX receive function in your sequencer or MIDI recording hardware. Consult the manual for these on how to do this.

Select “Patch” to dump a single patch. You will be asked what patch you want to dump:



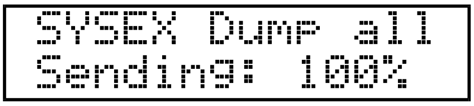
```
SYSEX Dump patch
[P02 Itchy     ]
```

Press **[ENTER]** to dump the patch you have selected.

Dumping the entire patch memory

It is possible to dump the entire patch memory in your SidStation. The procedure is the same as above for a single patch but select “AllPatch” as the dump mode with the **SYSEXDump** switch.

Press **[ENTER]** to start the dump:



```
 SYSEX Dump all
 Sending: 100%
```

When the counter is showing 100% the transfer is complete.

Transferring patch data to the SidStation

Simply load your sequencer or MIDI recorder with the patch data you want to send. Then initiate the transfer according to the manual for these products.

You should be able to receive SYSEX patch data any time (but you’d better leave the SidStation alone while it processes the incoming data). The best menu to be located in when receiving patch data should be the ***play mode*** menu, where you can verify that the patches actually are received in realtime.

When receiving single patches they will be placed in current position selected in the patch choose menu. If that position is occupied the next free one will be used.

Note: When receiving complete banks that are stored with the "AllPatch" message the old patches in the memory will be lost. The patch memory will first be reset before the transferred patches are stored in their original positions. Be sure to save important patches before you perform this operation.

Programming tips

Programming the SidStation is like programming many other synthesizers in some aspects but very different in others. In order to get a deeper understanding the only way is experimenting and listening to what happens. Below we have gathered a few tips for both the inexperienced and the experienced synthesizer programmer.

- A good idea is to start with a single oscillator when starting editing a new patch. Disable the other two oscillators and build up your patch gradually. This is extra important when you are using ring modulation and synchronisation of the oscillators, so that you can hear the effect of every new parameter set.

- It is often good to start off editing an existing patch if you are new to synthesizer programming.

- Try to use the speed setting on the arpeggiator for interesting effects. By setting the arpeggio speed of each oscillator to half the speed of the one before, you might get some exciting patterns.

- Experiment with the routing of the filter. Try to filter only one or two oscillators to get some new timbres.

- When using ***poly mode*** with filter, or routing all of the oscillators of a single sound through the filter, you can get beautiful cutoff-frequency-depending distortion from the SID-filter.

- The LFO is an essential part of a synthesizer. Use this to get everything from wild arcade game sounds to subtle vibratos.

- Experiment with filtering the noise function and tables played at high speeds to get percussive sounds. Try out the inverted ramp LFO on the vibrato modulation to get a good percussive pitch sweep for bass drums.

- When using waveform tables remember that you can always turn off the oscillator waveform and only use the waveforms specified in the table. The table can be used as a small sequencer to make small looping phrases that can be synched to MIDI clock events. Think of it as an extended arpeggiator.

- Ring modulation and synchronisation are perhaps the most characteristic features of the SID chip. It can be used to create lead sounds that cuts through any mix. Even if the triangle waveform gives the best results try out the other ones as modulators. Also try to use the oscillator sync with ring modulation together. When using tables you can turn these features on for some steps to create exciting timbre changes.

- The audio input is routed through the filter. This allows you to get external signals to run through the SID-chip lo-fi filter distortion. The filter envelope can be triggered with external note-on messages and control-changes.

- For patches not using the filter, set the cutoff-frequency to zero to get the noiselevel down slightly.

- Use the zero-menus in the oscillator and LFO-menus for quick navigation.

Finally it is up to you to explore the SidStations capabilities to find new exciting sound textures.

Trouble shooting

In the event of problems with your SidStation you can find some solutions to some common problems in this section.

The nature of the problem	**Suggested solution**
My SidStation does not start up at power on.	Have you checked that your power supply is one that is compliant with the specifications in the “Brief overview” section?
When I start up my SidStation the LCD screen is blank but the indicator LEDs are working.	Adjust the LCD contrast with the contrast knob on the back of the SidStation.
When I play notes on my MIDI keyboard nothing happens. The MIDI indicator LED indicates that data is received.	Check your audio cables and routing. Try changing sound. When the MIDI indicator LED is blinking the SidStation should be playing sounds.
When I play notes on my MIDI keyboard nothing happens. The MIDI indicator LED is never lit up.	Check the MIDI base channel setting in the System->MIDI menu. This is the only channel that the SidStation accepts MIDI data on. Please refer to the section “The menu system” for more info. Also, check your MIDI cables and routing.
The SidStation boots up fine, but crashes immediately when I enter the patch menu.	The memory of your SidStation appears to have become corrupted. Try resetting the memory by accessing the System-Memory-Reset option.
After resetting the memory there are no patches available.	This is normal. If you want the preset patches you will have to download them from the SidStation website and transfer them to your SidStation.

If your problem still exists please contact technical support at:

Email: support@sidstation.com
Fax: +46 31 772 81 11

Please state your full name and return email address / fax number. Also include the OS version of your SidStation (as seen on startup) and the serial number.

Reference section

Factory patches

These are the authors of the patches that are distributed as default with the SidStation synthesizer v1.1.

Patch name	Author	Patch name	Author
P00 Anpanman	Daniel Hansson	P45 Pulsation	Klaus P. Rausch
P01 Krutong	Daniel Hansson	P46 Coming In	Klaus P. Rausch
P02 Floating-7	Daniel Hansson	P47 A Starpego	Klaus P. Rausch
P03 P Itchy	Daniel Hansson	P48 DualFunctn	Klaus P. Rausch
P04 Paddan	Daniel Hansson	P49 RM-Synth1	Klaus P. Rausch
P05 VeloBass	Daniel Hansson	P50 P Britepad	Klaus P. Rausch
P06 Tel-Lead	Daniel Hansson	P51 P IcySynth	Klaus P. Rausch
P07 Acid ModWh	Daniel Hansson	P52 P Ghosty	Klaus P. Rausch
P08 Mental FX	Daniel Hansson	P53 Jarrism	Klaus P. Rausch
P09 8bitOrchst	Izik Levy	P54 Fatburger	Klaus P. Rausch
P10 C64 Stabs	Mikael Räim	P55 PulsePorta	Klaus P. Rausch
P11 Body drive	Mikael Räim	P56 EvolvingSy	Klaus P. Rausch
P12 Spaceducks	Mikael Räim	P57 Fluttering	Klaus P. Rausch
P13 Nebula pad	Mikael Räim	P58 ThinBandPs	Klaus P. Rausch
P14 Distorted	Mikael Räim	P59 64’Floor	Klaus P. Rausch
P15 Killer	Mikael Räim	P60 NewAger	Klaus P. Rausch
P16 Dreamy Arp	Mikael Räim	P61 Chaser	Klaus P. Rausch
P17 Hellcat	Mikael Räim	P62 EchoeSynth	Klaus P. Rausch
P18 Tracers DA	Daren Ager	P63 Electron Ω	Klaus P. Rausch
P19 JV.AirMod	Joeri Vankeirsbilck	P64 FlatΣ-Type	Klaus P. Rausch
P20 JV.BabyBas	Joeri Vankeirsbilck	P65 RMG Tri	Klaus P. Rausch
P21 JV.Basis	Joeri Vankeirsbilck	P66 ShrpIntLd	Klaus P. Rausch
P22 JV.BeaBass	Joeri Vankeirsbilck	P67 Matricky	Klaus P. Rausch
P23 JV.Cheesy	Joeri Vankeirsbilck	P68 Synched	Klaus P. Rausch
P24 JV.Crunchy	Joeri Vankeirsbilck	P69 Intervaler	Klaus P. Rausch
P25 JV.DropFX	Joeri Vankeirsbilck	P70 CuttinLead	Klaus P. Rausch
P26 JV.Electro	Joeri Vankeirsbilck	P71 Vengasynth	Klaus P. Rausch
P27 JV.LaVache	Joeri Vankeirsbilck	P72 Syncrubber	Klaus P. Rausch
P28 JV.LoFi	Joeri Vankeirsbilck	P73 Pyramid	Klaus P. Rausch
P29 JV.OberPWM	Joeri Vankeirsbilck	P74 AnalogBase	Klaus P. Rausch
P30 JV.PaddyFx	Joeri Vankeirsbilck	P75 Octabase	Klaus P. Rausch
P31 JV.Quadran	Joeri Vankeirsbilck	P76 Wha Bass	Klaus P. Rausch
P32 JV.SIDHype	Joeri Vankeirsbilck	P77 SparkleBas	Klaus P. Rausch
P33 JV.Whisper	Joeri Vankeirsbilck	P78 Heartbeat	Klaus P. Rausch
P34 A Fly High	Klaus P. Rausch	P79 Subloop	Klaus P. Rausch
P35 A WalkLoop	Klaus P. Rausch	P80 Aliensign	Klaus P. Rausch
P36 A Minimals	Klaus P. Rausch	P81 Digicrickt	Klaus P. Rausch
P37 A Dipeggio	Klaus P. Rausch	P82 Translava	Klaus P. Rausch
P38 SelfPlayr1	Klaus P. Rausch	P83 Formula X1	Klaus P. Rausch
P39 NoisyNeedl	Klaus P. Rausch	P84 Weird One	Klaus P. Rausch
P40 Doorbeller	Klaus P. Rausch	P85 BassDront	Daniel Hansson
P41 P Autochrd	Klaus P. Rausch	P86 BassDrTech	Daniel Hansson
P42 Eurodance	Klaus P. Rausch	P87 SomeHat	Daniel Hansson
P43 FilmScore1	Klaus P. Rausch	P88 Snare^Lego	Daniel Hansson
P44 Trance it	Klaus P. Rausch	P89 Snare Std	Daniel Hansson

Technical specifications

Model name:		SidStation, MOS 6581 synthesis module
Polyphony:		3 voices (poly mode), 1 voice (single mode)
Maximum number of patches:		100
SID chip characteristics:		
	3 Tone Oscillators	Range: 0-4 kHz
	4 Waveforms/ Oscillator	Triangle, Sawtooth, Variable Pulse, Noise
	3 Amplitude Modulators	Range: 48 dB
	3 Envelope Generators	Exponential response Attack Rate: 2 ms - 8 s Decay Rate: 6 ms - 24 s Sustain Level: 0 - peak volume Release Rate: 6 ms - 24 s
	Oscillator Synchronization Ring Modulation	
	Programmable Filter	Cutoff range: 30 Hz - 12 kHz 12 dB/octave rolloff Low pass, bandpass, high pass, Notch Variable Resonance
Display:		2 x 16 character back lit LCD
Connectors:		
	Audio Out	1 x ¼-inch phone (Unbalanced)
	Ext In	1 x ¼-inch phone (Unbalanced)
	MIDI	3 x DIN5P (In, Out, Thru)
	DC In	1 x 2.1 mm standard Euro DC connector
Power requirements:		6Volts DC, 400 mA
Dimensions:		240W x 70H x 200D mm
Weight:		Approx 1.2kg

Credits

Initial hardware design:	Anders Gärder Daniel Hansson Kristoffer Johansson Jens Wall
SidStation programming:	Daniel Hansson
PCB layout:	Mikael Räim
ASID emulator:	Anders Gärder
Flash upgrading:	Anders Gärder
Manual author:	Mikael Räim
Addition manual writing:	Daniel Hansson

We would like to thank the following people for help on the project:

Lennart Hansson,	Computer Engineering Chalmers, for project supervision.
Hans Bergstrand,	Computer Engineering Chalmers, for sharing his experience while designing the printed circuit board.
Henrik Sundberg,	for assistance with the development & production of the casing.
Ingvar Anderson,	CEO Chalmers Invest, for supporting our project.
Tomas Danko Jr.	for betatesting.
ACNE International	for being cool in general

Assistance for the user manual:

Andreas Varga	('SID Homepage' http://stud1.tuwien.ac.at/~e9426444/sidpage.html)
Tim Forsyth	(http://www.soton.ac.uk/~tf/)

Patch programming:

Klaus P. Rausch
Joeri Vankeirsbilck
Daren Ager
Izik Levy
Mikael Räim
Daniel Hansson
Daniel Eriksson

Big thanks to everyone who have helped to make this project reality!

SidStation MIDI implementation

MIDI CONTROLLER ASSIGNMENTS

MIDI_CTRL	Name	Trn	Rec	Type
1	Modulation wheel		x	CONT
16	Direct Controller 1	x	x	CONT/BOOL
17	Direct Controller 2	x	x	CONT/BOOL
18	Direct Controller 3	x	x	CONT/BOOL
19	Direct Controller 4	x	x	CONT/BOOL
20	LFO CTRL1		x	CONT
21	LFO CTRL2		x	CONT
22	LFO CTRL3		x	CONT
23	LFO CTRL4		x	CONT
24	Oscillator1 active		x	BOOL
25	Oscillator2 active		x	BOOL
26	Oscillator3 active		x	BOOL
27	Filter Cutoff		x	CONT
28	Filter Envelope Depth		x	CONT
29	Filter LFO Depth		x	CONT
30	Filter Envelope Attack		x	CONT
31	Filter Envelope Decay		x	CONT
32	Filter Envelope Sustain		x	CONT
33	Filter Envelope Release		x	CONT
34	Osc1 Arpeggiator Speed		x	CONT
35	Osc1 Pitch/Track		x	CONT
36	Osc1 Transpose		x	CONT
37	Osc1 Vibrato Depth		x	CONT
38	Osc1 Detune		x	CONT
39	Osc1 Portamento Speed		x	CONT
40	Osc1 Synchronize		x	BOOL
41	Osc1 RingModulation		x	BOOL
42	Osc1 PWM Startvalue		x	CONT
43	Osc1 PWM Addvalue		x	CONT
44	Osc1 PWM LFO Depth		x	CONT
45	Osc1 Delay		x	CONT
46	Osc1 Attack		x	CONT
47	Osc1 Decay		x	CONT
48	Osc1 Sustain		x	CONT
49	Osc1 Release		x	CONT
50	Osc2 Arpeggiator Speed		x	CONT
51	Osc2 Pitch/Track		x	CONT
52	Osc2 Transpose		x	CONT
53	Osc2 Vibrato Depth		x	CONT
54	Osc2 Detune		x	CONT
55	Osc2 Portamento Speed		x	CONT
56	Osc2 Synchronize		x	BOOL
57	Osc2 RingModulation		x	BOOL
58	Osc2 PWM Startvalue		x	CONT
59	Osc2 PWM Addvalue		x	CONT
60	Osc2 PWM LFO Depth		x	CONT
61	Osc2 Delay		x	CONT
62	Osc2 Attack		x	CONT
63	Osc2 Decay		x	CONT
70	Osc2 Sustain		x	CONT
71	Osc2 Release		x	CONT
72	Osc3 Arpeggiator Speed		x	CONT
73	Osc3 Pitch/Track		x	CONT
74	Osc3 Transpose		x	CONT
75	Osc3 Vibrato Depth		x	CONT
76	Osc3 Detune		x	CONT
77	Osc3 Portamento Speed		x	CONT
78	Osc3 Synchronize		x	BOOL
79	Osc3 RingModulation		x	BOOL
80	Osc3 PWM Startvalue		x	CONT
81	Osc3 PWM Addvalue		x	CONT
82	Osc3 PWM LFO Depth		x	CONT
83	Osc3 Delay		x	CONT

84	Osc3 Attack		x	CONT
85	Osc3 Decay		x	CONT
86	Osc3 Sustain		x	CONT
87	Osc3 Release		x	CONT
88	LFO1 Speed		x	CONT
89	LFO1 Depth		x	CONT
90	LFO1 AddDepth		x	CONT
91	LFO1 Fade In		x	CONT
92	LFO1 Sample And Hold		x	CONT
93	LFO1 Lace Speed		x	CONT
94	LFO2 Speed		x	CONT
95	LFO2 Depth		x	CONT
102	LFO2 AddDepth		x	CONT
103	LFO2 Fade In		x	CONT
104	LFO2 Sample And Hold		x	CONT
105	LFO2 Lace Speed		x	CONT
106	LFO3 Speed		x	CONT
107	LFO3 Depth		x	CONT
108	LFO3 AddDepth		x	CONT
109	LFO3 Fade In		x	CONT
110	LFO3 Sample And Hold		x	CONT
111	LFO3 Lace Speed		x	CONT
112	LFO4 Speed		x	CONT
113	LFO4 Depth		x	CONT
114	LFO4 AddDepth		x	CONT
115	LFO4 Fade In		x	CONT
116	LFO4 Sample And Hold		x	CONT
117	LFO4 Lace Speed		x	CONT

SYSEX messages:

(All data starting with a dollar sign ($) is printed in hexadecimal format.)

SYSEX init:

All SYSEX messages start with this sequence.

MIDI Byte	Purpose
$f0	SYSEX Start
$00	Europe/USA ID
$20	Europe ID
$3c	Elektron ESI ID
$01	SidStation ID
$00	Base channel (Padding)

SYSEX Patch all clear:

Clears all patch positions.

MIDI Byte	Purpose
(SYSEX init)	
$01	Dump Patch memory all clear ID
$45	Magic byte
$2d*14	Pad
$f7	SYSEX end

SYSEX Patch dump:

This is the data that is sent and received for patch dump.

MIDI Byte	Purpose
(SYSEX init)	
$02	Dump Patch ID
$00	Dump Patch version
%000000aa	aa = upper bits of patch size (in bytes)
%0aaaaaaa	aaaaaaa = lower bits of patch size (inbytes)
$2d*24	Pad
$45	Start of Patch Data bytes

```
   ...      | Patch Data bytes ( See separate documentation )
   $f7      | SYSEX end
```

SYSEX Skip Patch (Clear):

This message is sent/received for increasing the current patch position.

```
 MIDI Byte | Purpose
-----------+------------------------------------------
 (SYSEX init)
   $03     | Dump Skip Patch (Clear) ID
   $f7     | SYSEX end
```

SYSEX Direct Program (DP):

This message is recognised, and intended for directly program all available parameters of the current SidStation patch (see separate document for specification).

```
 MIDI Byte | Purpose
-----------+------------------------------------------
 (SYSEX init)
   $04     | Dump Skip Patch (Clear) ID
 %00000aaa | Memory position (high 3-bits)
 %0aaaaaaa | Memory position (low 7-bits)
 %0mmmmmmm | Mask
 %00000sss | Positions to shift up
 %0ddddddd | Data to program
   $f7     | SYSEX end
```

SYSEX Patch data bytes:

The first ten patch data bytes are the patchname in ASCII format.

Subsequent bytes are sent and received in high/low-nibble format as seen below:

```
%0000aaaa - highnibble
%0000bbbb - lownibble
```

Together they form an eightbit byte as: %aaaabbbb

Patchdata:

```
--------+------------------+---------------------------------------------
 Index  | Name             | Function
--------+------------------+---------------------------------------------
 0..9   | PATCH_NAME       | Patch name in ASCII
  10    | PATCH_DCTRL1     | Direct controller 1 (0..96)
  11    | PATCH_DCTRL1_LD  | DCTRL1 limit down (0..127 - must be lower than
        |                  | PATCH_DCTRL1_LU)
  12    | PATCH_DCTRL1_LU  | DCTRL1 limit up (0..127 - must be higher than
        |                  | PATCH_DCTRL1_LD)
  13    | PATCH_DCTRL2     | Direct controller 2 (0..96)
  14    | PATCH_DCTRL2_LD  | DCTRL2 limit down (0..127 - must be lower than
        |                  | PATCH_DCTRL2_LU)
  15    | PATCH_DCTRL2_LU  | DCTRL2 limit up (0..127 - must be higher than
        |                  | PATCH_DCTRL2_LD)
  16    | PATCH_DCTRL3     | Direct controller 3 (0..96)
  17    | PATCH_DCTRL3_LD  | DCTRL3 limit down (0..127 - must be lower than
        |                  | PATCH_DCTRL3_LU)
  18    | PATCH_DCTRL3_LU  | DCTRL3 limit up (0..127 - must be higher than
        |                  | PATCH_DCTRL3_LD)
  19    | PATCH_DCTRL4     | Direct controller 4 (0..96)
  20    | PATCH_DCTRL4_LD  | DCTRL4 limit down (0..127 - must be lower than
        |                  | PATCH_DCTRL4_LU)
  21    | PATCH_DCTRL4_LU  | DCTRL4 limit up (0..127 - must be higher than
        |                  | PATCH_DCTRL4_LD)
  22    | PATCH_MODE       | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | |  7  |  6  |  5  |  4  |  3  |  2  |  1  |  0  |
        |                  | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | |FEINV|FWRAP|LEGAT|     | POLY| OSC3| OSC2| OSC1|
        |                  | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | FEINV - Filter Envelope Invert
        |                  | FWRAP - Filter wrap
        |                  | LEGAT - Portamento Legato On/Off
        |                  | POLY  - Poly mode On/Off
        |                  | OSC3  - Oscillator 3 active
        |                  | OSC2  - Oscillator 2 active
        |                  | OSC1  - Oscillator 1 active
  23    | PATCH_MODE2      | Not used - set to zero
  24    | PATCH_FLT_ACTRS  | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | |  7  |  6  |  5  |  4  |  3  |  2  |  1  |  0  |
        |                  | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | |    RESONANCE (0..15)  |     | FLT3| FLT2| FLT1|
        |                  | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | FLT3 - Oscillator 3 routed through filter
        |                  | FLT2 - Oscillator 2 routed through filter
        |                  | FLT1 - Oscillator 1 routed through filter
  25    | PATCH_FLT_TYPLFO | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | |  7  |  6  |  5  |  4  |  3  |  2  |  1  |  0  |
        |                  | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | |     |     | FilterLFO |     | Filtertype(0..7)|
        |                  | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | FilterLFO - What LFO to control filter cutoff (0..3)
  26    | PATCH_FLT_CUTOFF | Cutoff value for filter (0..127)
  27    | PATCH_FLT_ENVDEP | Filter envelope depth (0..127)
  28    | PATCH_FLT_ENVATT | Filter envelope attack (0..127)
  29    | PATCH_FLT_ENVDEC | Filter envelope decay  (0..127)
  30    | PATCH_FLT_ENVSUS | Filter envelope sustain(0..127)
  31    | PATCH_FLT_ENVREL | Filter envelope release(0..127)
  32    | PATCH_FLT_LFODEP | LFO depth for filter (0..127)
  33    | PATCH_FLT_LFOWHD | Amount the modulation wheel adds to filter LFO (0..127)
  34    | PATCH_SYNC_SPEED | Local update speed for patch (50..200)
  35    | PATCH_SYNC_HCUT  | Local Hardcut setting for patch (0..15)
36..56  | PATCH_OSC1       | Settings for Oscillator 1 (21 bytes)
57..77  | PATCH_OSC2       | Settings for Oscillator 2 (21 bytes)
78..98  | PATCH_OSC3       | Settings for Oscillator 3 (21 bytes)
99..109 | PATCH_LFO1       | Settings for LFO 1 (11 bytes)
110..120| PATCH_LFO2       | Settings for LFO 2 (11 bytes)
121..131| PATCH_LFO3       | Settings for LFO 3 (11 bytes)
132..142| PATCH_LFO4       | Settings for LFO 4 (11 bytes)
143..   | PATCH_TABLE1     | Table data (see below for documentation)
   ..   | PATCH_TABLE2     | ..
   ..   | PATCH_TABLE3     | ..
--------+------------------+---------------------------------------------
```

Oscillator data:

```
--------+------------------+---------------------------------------------
 Index  | Name             | Function
--------+------------------+---------------------------------------------
  x+0   | OSC_FLAGS        | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | |  7  |  6  |  5  |  4  |  3  |  2  |  1  |  0  |
        |                  | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | |     |     |     |     |     |     |GATE |SPWM |
        |                  | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | SPWM - Synchronize PWM-add, ie restart PWM-value from
        |                  |        PWM_START on every NOTE_ON received.
        |                  | GATE - Use gate or SID-envelope
  x+1   | OSC_TRACK        | 0 = Keyboardtracking,
        |                  | 1..99 = fixed note
  x+2   | OSC_ARPSPEED     | 0 = No arpeggiator
        |                  | 1..127 = arpeggiator speed
  x+3   | OSC_TRANSPOSE    | Oscillator Transpose (-24..+24)
  x+4   | OSC_DETUNE       | Oscillator Detune  (-64..+63)
  x+5   | OSC_PBRANGE      | Oscillator pitchbend range (0..24)
  x+6   | OSC_ATTACK       | SID-envelope Attack (0..15)
  x+7   | OSC_DECAY        | SID-envelope Decay  (0..15)
  x+8   | OSC_SUSTAIN      | SID-envelope Sustain / Gate volume (0..15)
  x+9   | OSC_RELEASE      | SID-envelope Release(0..15)
  x+10  | OSC_DELAY        | Oscillator delay (0..127)
  x+11  | OSC_PWM_START    | Pulsewidth start (0..127)
  x+12  | OSC_PWM_ADD      | Pulsewidth modulation add (PWM-sweep) (0..127)
  x+13  | OSC_PWM_LFO      | Pulsewidth LFO number (0..3)
  x+14  | OSC_PWM_LFODPTH  | Pulsewidth LFO depth (0..127)
  x+15  | OSC_WAVE         | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | |  7  |  6  |  5  |  4  |  3  |  2  |  1  |  0  |
        |                  | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | | Oscillator Waveform   |     |RINGM|SYNC |     |
        |                  | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | Oscillator Waveform:
        |                  | $1 = Triangle
        |                  | $2 = Saw
        |                  | $4 = Pulse
        |                  | $5 = Mixed
        |                  | $8 = Noise
        |                  | RINGM = ringmodulation On/Off
        |                  | SYNC = synchronization On/Off
  x+16  | OSC_PORTSPEED    | Oscillator portamento speed (0..127)
  x+17  | OSC_VIB_LFO      | Oscillator vibrato LFO (0..3)
  x+18  | OSC_VIB_DEPTH    | Oscillator vibrato depth (0..127)
  x+19  | OSC_VIB_WHD      | Oscillator vibrato modulation wheel depth (0..127)
  x+20  | OSC_TABLESPEED   | 0 = Table not activated
        |                  | 1..127 = Table speed
--------+------------------+---------------------------------------------
```

LFO data:

```
 Index  | Name             | Function
--------+------------------+---------------------------------------------
  x+0   | LFO_CTRL_TYPE    | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | |  7  |  6  |  5  |  4  |  3  |  2  |  1  |  0  |
        |                  | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | |     |  CTRL_Source    |     |     | LFO_TYPE  |
        |                  | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | CTRL_Source (0..11): 0 = ModWheel
        |                  |                      1 = PitchBend
        |                  |                      2 = Velocity
        |                  |                      3 = Aftertouch
        |                  |                      4 = LFO_CTRL1 (MIDI_CTRL)
        |                  |                      5 = LFO_CTRL2 (MIDI_CTRL)
        |                  |                      6 = LFO_CTRL3 (MIDI_CTRL)
        |                  |                      7 = LFO_CTRL4 (MIDI_CTRL)
        |                  |                      8 = LFO1
        |                  |                      9 = LFO2
        |                  |                      10 = LFO3
        |                  |                      11 = LFO4
        |                  | LFO_TYPE (0..4)      0 = Triangle
        |                  |                      1 = Saw
        |                  |                      2 = Ramp
        |                  |                      3 = Pulse
        |                  |                      4 = Random
        |                  |                      7 = Flat
  x+1   | LFO_OPTIONS      | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | |  7  |  6  |  5  |  4  |  3  |  2  |  1  |  0  |
        |                  | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | |     |    CTRL_Dest    |SNOFF| ABZ | INV |SYNC |
        |                  | +-----+-----+-----+-----+-----+-----+-----+-----+
        |                  | SYNC = sync note on
        |                  | INV = invert LFO
        |                  | ABZ = (Above Zero) Output of LFO is strctly positive
        |                  | SNOFF = sync to note off
        |                  | CTRL_Dest = Destination of control:
        |                  |             0 = None
        |                  |             1 = LFO Depth
        |                  |             2 = LFO Speed
        |                  |             3 = LFO S/H
        |                  |             4 = LFO Lace
  x+2   | LFO_SPEED        | Speed of LFO (0..127)
  x+3   | LFO_SAMPHOLD     | Sample and hold (0..127) 0=off
  x+4   | LFO_DEPTH        | Depth of LFO (0..127)
  x+5   | LFO_ADDLFO       | Add output of this LFO (0..2)
  x+6   | LFO_LACE         | Speed of interlace (0..127)
  x+7   | LFO_LACEWITH     | Use this to lace with: (0..3)
        |                  |             0 = Zero
        |                  |             1 = LFO1
        |                  |             2 = LFO2
        |                  |             3 = LFO3
        |                  |             4 = LFO4
  x+8   | LFO_ADDDEPTH     | Amount of added LFO (0..127)
  x+9   | LFO_CTRLVAL      | Amount the CTRL data can affect the depth (0..127)
  x+10  | LFO_FADEIN       | Speed of Fade in.
        |                  | 0 = no fade in
        |                  | 1..127 = fade in speed
--------+------------------+---------------------------------------------
```

Table data:

```
 Index  | Name             | Function
--------+------------------+---------------------------------------------
  x+0   | TABLE_DATA1      | $FF = TABLE_END, no TABLE_DATA2 follows after.
        |                  |       No more data for this table.
        |                  | $FE = TABLE_LOOP, TABLE_DATA2 tells looppoint.
        |                  |       Only TABLE_DATA2 follows for this table
        |                  | Other values: Waveform (see OSC_WAVE for
        |                  | specification).
  x+1   | TABLE_DATA2      | $00..$7F = Fixed notenumber
        |                  | $80..$BF = Add TABLE_DATA2-$80 to notenumber.
        |                  | $C0..$FF = Subtract TABLE_DATA2-$C0 from notenumber.
        |                  | TABLE_LOOPPOINT if TABLE_DATA1 was TABLE_LOOP
  ...   | TABLE_DATA1 ...
```

TABLE_DATA1 and TABLE_DATA2 follows after each other, building up the steps in the table. The table can be of maximum size of 32. After that the table is ended even if TABLE_END or TABLE_LOOP were found. Normally a table is ended with TABLE_END or TABLE_LOOP+TABLE_LOOPPOINT.

SYSEX Direct Program messages:

All Direct Program messages start with the general SidStation Direct Program-sequence:
F0 00 20 3c 01 00 04
...continues with one (not more) Direct Program-sequence as specified below in the chart below, and ends with the standard F7 SYSEX-end-byte.

The x/y-characters refers to data as specified in the Function row.

Only use the data specified in the Function-menu. Sending data other than what is specified can make the SidStation crash and invalidate the patchmemory.

Direct Program sequences:

```
---------------------------+---------------------------------------------
 Sequence                  | Function
---------------------------+---------------------------------------------
 00 00 7f 00 %0yyyyyyy     |
     ..                    | Patch name in (extended) ASCII (positions 0..9)
 00 09 7f 00 %0yyyyyyy     |

 00 0a 7f 00 %0yyyyyyy     | Direct Controller 1 (0=OFF,1..82=DCTRL_x)
 00 0b 7f 00 %0yyyyyyy     | DCTRL1 limit up
 00 0c 7f 00 %0yyyyyyy     | DCTRL1 limit down
 00 0d 7f 00 %0yyyyyyy     | Direct Controller 2 (0=OFF,1..82=DCTRL_x)
 00 0e 7f 00 %0yyyyyyy     | DCTRL2 limit up
 00 0f 7f 00 %0yyyyyyy     | DCTRL2 limit down
 00 10 7f 00 %0yyyyyyy     | Direct Controller 3 (0=OFF,1..82=DCTRL_x)
```

```
00 11 7f 00 %0yyyyyyy      | DCTRL3 limit up
00 12 7f 00 %0yyyyyyy      | DCTRL3 limit down
00 13 7f 00 %0yyyyyyy      | Direct Controller 4 (0=OFF,1..82=DCTRL_x)
00 14 7f 00 %0yyyyyyy      | DCTRL4 limit up
00 15 7f 00 %0yyyyyyy      | DCTRL4 limit down
00 16 01 00 %0000000y      | Osc1 ON/OFF
00 16 01 01 %0000000y      | Osc2 ON/OFF
00 16 01 02 %0000000y      | Osc3 ON/OFF
00 16 01 03 %0000000y      | Poly/Single
00 16 01 04 %0000000y      | Filter sync to note_on
00 16 01 05 %0000000y      | Legato switch
00 16 01 06 %0000000y      | Filter wrap switch
00 16 01 07 %0000000y      | Filter Envelope Invert
00 18 07 00 %00000yyy      | Filter Active (Osc 3/2/1)
00 18 0f 04 %0000yyyy      | Filter Resonance
00 19 07 00 %00000yyy      | Filtermode (high/band/lo)
00 19 03 04 %000000yy      | LFO to control filter cutoff
00 19 01 06 %0000000y      | Force NoteOn ReInit
00 1a 7f 00 %0yyyyyyy      | Filter Cutoff position
00 1b 7f 00 %0yyyyyyy      | Filter Envelope Depth
00 1c 7f 00 %0yyyyyyy      | Filter Envelope Attack
00 1d 7f 00 %0yyyyyyy      | Filter Envelope Decay
00 1e 7f 00 %0yyyyyyy      | Filter Envelope Sustain
00 1f 7f 00 %0yyyyyyy      | Filter Envelope Release
00 20 7f 00 %0yyyyyyy      | Filter LFO Depth
00 21 7f 00 %0yyyyyyy      | Filter LFO Wheel Depth
00 22 7f 00 %0yyyyyyy      | Pitch Sync Speed (50..200)
00 23 0f 00 %0000yyyy      | Pitch Sync HCut

00 47 01 00 %0000000y      | Osc1: Sync PWM-add to key-on
00 47 01 01 %0000000y      | Osc1: Envelope gate ON/OFF
00 48 7f 00 %0yyyyyyy      | Osc1: Pitch track
                           |       (0 = track keyboard, other = fixed note)
00 49 7f 00 %0yyyyyyy      | Osc1: Arpeggiator Speed (0 = no arpeggio on osc)
00 4a 7f 00 %0yyyyyyy      | Osc1: Transpose U/D (signed 7-bit, +/-24)
00 4b 7f 00 %0yyyyyyy      | Osc1: Detune U/D (signed 7-bit)
00 4c 7f 00 %0yyyyyyy      | Osc1: Pitch bend range (0..24)
00 4d 0f 00 %0000yyyy      | Osc1: Attack
00 4e 0f 00 %0000yyyy      | Osc1: Decay
00 4f 0f 00 %0000yyyy      | Osc1: Sustain
00 50 0f 00 %0000yyyy      | Osc1: Release
00 51 7f 00 %0yyyyyyy      | Osc1: Delay
00 52 7f 00 %0yyyyyyy      | Osc1: Pulsewidth start
00 53 7f 00 %0yyyyyyy      | Osc1: Pulsewidth addvalue
00 54 03 00 %000000yy      | Osc1: Pulsewidth LFO1..4 [0..3]
00 55 7f 00 %0yyyyyyy      | Osc1: Pulsewidth LFO-depth
00 56 0f 04 %0000yyyy      | Osc1: Waveform(1=Tri,2=Saw,4=Pulse,5=Mixed,8=Noise)
00 57 7f 00 %0yyyyyyy      | Osc1: Portamento speed (0..99)
00 58 03 00 %000000yy      | Osc1: Vibrato LFO1..4 [0..3]
00 59 7f 00 %0yyyyyyy      | Osc1: Vibrato LFO-depth
00 5a 7f 00 %0yyyyyyy      | Osc1: Vibrato LFO-wheel-depth
00 5b 7f 00 %0yyyyyyy      | Osc1: Tablespeed (0 = no table)

00 5c 01 00 %0000000y      | Osc2: Sync PWM-add to key-on
00 5c 01 01 %0000000y      | Osc2: Envelope gate ON/OFF
00 5d 7f 00 %0yyyyyyy      | Osc2: Pitch track
                           |       (0 = track keyboard, other = fixed note)
00 5e 7f 00 %0yyyyyyy      | Osc2: Arpeggiator Speed (0 = no arpeggio on osc)
00 5f 7f 00 %0yyyyyyy      | Osc2: Transpose U/D (signed 7-bit, +/-24)
00 60 7f 00 %0yyyyyyy      | Osc2: Detune U/D (signed 7-bit)
00 61 7f 00 %0yyyyyyy      | Osc2: Pitch bend range (0..24)
00 62 0f 00 %0000yyyy      | Osc2: Attack
00 63 0f 00 %0000yyyy      | Osc2: Decay
00 64 0f 00 %0000yyyy      | Osc2: Sustain
00 65 0f 00 %0000yyyy      | Osc2: Release
00 66 7f 00 %0yyyyyyy      | Osc2: Delay
00 67 7f 00 %0yyyyyyy      | Osc2: Pulsewidth start
00 68 7f 00 %0yyyyyyy      | Osc2: Pulsewidth addvalue
00 69 03 00 %000000yy      | Osc2: Pulsewidth LFO1..4 [0..3]
00 6a 7f 00 %0yyyyyyy      | Osc2: Pulsewidth LFO-depth
00 6b 0f 04 %0000yyyy      | Osc2: Waveform(1=Tri,2=Saw,4=Pulse,5=Mixed,8=Noise)
00 6c 7f 00 %0yyyyyyy      | Osc2: Portamento speed (0..99)
00 6d 03 00 %000000yy      | Osc2: Vibrato LFO1..4 [0..3]
00 6e 7f 00 %0yyyyyyy      | Osc2: Vibrato LFO-depth
00 6f 7f 00 %0yyyyyyy      | Osc2: Vibrato LFO-wheel-depth
00 70 7f 00 %0yyyyyyy      | Osc2: Tablespeed (0 = no table)

00 71 01 00 %0000000y      | Osc3: Sync PWM-add to key-on
00 71 01 01 %0000000y      | Osc3: Envelope gate ON/OFF
00 72 7f 00 %0yyyyyyy      | Osc3: Pitch track
                           |       (0 = track keyboard, other = fixed note)
00 73 7f 00 %0yyyyyyy      | Osc3: Arpeggiator Speed (0 = no arpeggio on osc)
00 74 7f 00 %0yyyyyyy      | Osc3: Transpose U/D (signed 7-bit, +/-24)
00 75 7f 00 %0yyyyyyy      | Osc3: Detune U/D (signed 7-bit)
00 76 7f 00 %0yyyyyyy      | Osc3: Pitch bend range (0..24)
00 77 0f 00 %0000yyyy      | Osc3: Attack
00 78 0f 00 %0000yyyy      | Osc3: Decay
00 79 0f 00 %0000yyyy      | Osc3: Sustain
00 7a 0f 00 %0000yyyy      | Osc3: Release
00 7b 7f 00 %0yyyyyyy      | Osc3: Delay
00 7c 7f 00 %0yyyyyyy      | Osc3: Pulsewidth start
00 7d 7f 00 %0yyyyyyy      | Osc3: Pulsewidth addvalue
00 7e 03 00 %000000yy      | Osc3: Pulsewidth LFO1..4 [0..3]
00 7f 7f 00 %0yyyyyyy      | Osc3: Pulsewidth LFO-depth
01 00 0f 04 %0000yyyy      | Osc3: Waveform(1=Tri,2=Saw,4=Pulse,5=Mixed,8=Noise)
01 01 7f 00 %0yyyyyyy      | Osc3: Portamento speed (0..99)
01 02 03 00 %000000yy      | Osc3: Vibrato LFO1..4 [0..3]
01 03 7f 00 %0yyyyyyy      | Osc3: Vibrato LFO-depth
01 04 7f 00 %0yyyyyyy      | Osc3: Vibrato LFO-wheel-depth
01 05 7f 00 %0yyyyyyy      | Osc3: Tablespeed (0 = no table)

01 06 07 00 %00000yyy      | LFO1: Type (0=Tri,1=Saw,2=Ramp,3=Pulse,4=DC)
01 06 0f 04 %0000yyyy      | LFO1: CTRL_source (0=ModWheel,1=PitchBend,2=Velocity,
                           |       3=Afttouch,4..7=CTRL1..4,8..11=LFO1..4)
01 07 01 00 %0000000y      | LFO1: Sync to note on
01 07 01 01 %0000000y      | LFO1: Invert
01 07 01 02 %0000000y      | LFO1: Above zero
01 07 01 03 %0000000y      | LFO1: Sync to note off
01 07 07 04 %00000yyy      | LFO1: CTRL destination (0=None,1=LFO Depth,2=LFO Speed,
                           |       3=LFO S/H,4=LFO Lace)
01 08 7f 00 %0yyyyyyy      | LFO1: Speed
01 09 7f 00 %0yyyyyyy      | LFO1: Sample and hold (0=OFF)
01 0a 7f 00 %0yyyyyyy      | LFO1: Depth
01 0b 03 00 %000000yy      | LFO1: Add LFO (0..3=LFO1..4)
01 0c 7f 00 %0yyyyyyy      | LFO1: Lace speed (0=OFF)
01 0d 07 00 %00000yyy      | LFO1: Lace with (0=Zero,1..4=LFO1..4)
01 0e 7f 00 %0yyyyyyy      | LFO1: Add LFO depth
01 0f 7f 00 %0yyyyyyy      | LFO1: CTRL value
01 10 7f 00 %0yyyyyyy      | LFO1: Fade In (0=OFF)

01 22 07 00 %00000yyy      | LFO2: Type (0=Tri,1=Saw,2=Ramp,3=Pulse,4=DC)
01 22 0f 04 %0000yyyy      | LFO2: CTRL_source (0=ModWheel,1=PitchBend,2=Velocity,
                           |       3=Afttouch,4..7=CTRL1..4,8..11=LFO1..4)
01 23 01 00 %0000000y      | LFO2: Sync to note on
01 23 01 01 %0000000y      | LFO2: Invert
01 23 01 02 %0000000y      | LFO2: Above zero
01 23 01 03 %0000000y      | LFO2: Sync to note off
01 23 07 04 %00000yyy      | LFO2: CTRL destination (0=None,1=LFO Depth,2=LFO Speed,
                           |       3=LFO S/H,4=LFO Lace)
01 24 7f 00 %0yyyyyyy      | LFO2: Speed
01 25 7f 00 %0yyyyyyy      | LFO2: Sample and hold (0=OFF)
01 26 7f 00 %0yyyyyyy      | LFO2: Depth
01 27 03 00 %000000yy      | LFO2: Add LFO (0..3=LFO1..4)
01 28 7f 00 %0yyyyyyy      | LFO2: Lace speed (0=OFF)
01 29 07 00 %00000yyy      | LFO2: Lace with (0=Zero,1..4=LFO1..4)
01 2a 7f 00 %0yyyyyyy      | LFO2: Add LFO depth
01 2b 7f 00 %0yyyyyyy      | LFO2: CTRL value
01 2c 7f 00 %0yyyyyyy      | LFO2: Fade In (0=OFF)

01 3e 07 00 %00000yyy      | LFO3: Type (0=Tri,1=Saw,2=Ramp,3=Pulse,4=DC)
01 3e 0f 04 %0000yyyy      | LFO3: CTRL_source (0=ModWheel,1=PitchBend,2=Velocity,
                           |       3=Afttouch,4..7=CTRL1..4,8..11=LFO1..4)
01 3f 01 00 %0000000y      | LFO3: Sync to note on
01 3f 01 01 %0000000y      | LFO3: Invert
01 3f 01 02 %0000000y      | LFO3: Above zero
01 3f 01 03 %0000000y      | LFO3: Sync to note off
01 3f 07 04 %00000yyy      | LFO3: CTRL destination (0=None,1=LFO Depth,2=LFO Speed,
                           |       3=LFO S/H,4=LFO Lace)
01 40 7f 00 %0yyyyyyy      | LFO3: Speed
01 41 7f 00 %0yyyyyyy      | LFO3: Sample and hold (0=OFF)
01 42 7f 00 %0yyyyyyy      | LFO3: Depth
01 43 03 00 %000000yy      | LFO3: Add LFO (0..3=LFO1..4)
01 44 7f 00 %0yyyyyyy      | LFO3: Lace speed (0=OFF)
01 45 07 00 %00000yyy      | LFO3: Lace with (0=Zero,1..4=LFO1..4)
01 46 7f 00 %0yyyyyyy      | LFO3: Add LFO depth
01 47 7f 00 %0yyyyyyy      | LFO3: CTRL value
01 48 7f 00 %0yyyyyyy      | LFO3: Fade In (0=OFF)

01 5a 07 00 %00000yyy      | LFO4: Type (0=Tri,1=Saw,2=Ramp,3=Pulse,4=DC)
01 5a 0f 04 %0000yyyy      | LFO4: CTRL_source (0=ModWheel,1=PitchBend,2=Velocity,
                           |       3=Afttouch,4..7=CTRL1..4,8..11=LFO1..4)
01 5b 01 00 %0000000y      | LFO4: Sync to note on
01 5b 01 01 %0000000y      | LFO4: Invert
01 5b 01 02 %0000000y      | LFO4: Above zero
01 5b 01 03 %0000000y      | LFO4: Sync to note off
01 5b 07 04 %00000yyy      | LFO4: CTRL destination (0=None,1=LFO Depth,2=LFO Speed,
                           |       3=LFO S/H,4=LFO Lace)
01 5c 7f 00 %0yyyyyyy      | LFO4: Speed
01 5d 7f 00 %0yyyyyyy      | LFO4: Sample and hold (0=OFF)
01 5e 7f 00 %0yyyyyyy      | LFO4: Depth
01 5f 03 00 %000000yy      | LFO4: Add LFO (0..3=LFO1..4)
01 60 7f 00 %0yyyyyyy      | LFO4: Lace speed (0=OFF)
01 61 07 00 %00000yyy      | LFO4: Lace with (0=Zero,1..4=LFO1..4)
01 62 7f 00 %0yyyyyyy      | LFO4: Add LFO depth
01 63 7f 00 %0yyyyyyy      | LFO4: CTRL value
01 64 7f 00 %0yyyyyyy      | LFO4: Fade In (0=OFF)

01 76 0f xx %0000yyyy      | Table1 (see below for details)
.. ..
02 77 0f xx %0000yyyy

02 78 0f xx %0000yyyy      | Table2 (see below for details)
.. ..
03 79 0f xx %0000yyyy

03 7a 0f xx %0000yyyy      | Table3 (see below for details)
.. ..
04 7b 0f xx %0000yyyy
```

Table Direct Program Details

Each row of the tables consist of one 16-bit word, which in direct programming breaks up in four four bit nibbles. Each row has to be programmed in one sequence (four Direct Program messages in sequence).

```
Nibble one:
 Ph Pl   0f 04 %0000yyyy   | Waveform (0=Off,1=Tri,2=Saw,4=Pulse,
                           |           5=Mixed,8=Noise,9=---)
Nibble two:
 Ph Pl   0f 00 %00000xy0   | x=Ringmod, y=Sync
Nibble three:
 Ph Pl+1 0f 04 %0000xxxx   | Pitch, high nibble
Nibble four:
 Ph Pl+1 0f 04 %0000yyyy   | Pitch, low nibble
```

The Pitch-nibbles form a byte with the following meaning:

```
00..51 - Fixed note
80..bf - Subtract value-80 from current note
c0..ff - Add value-c0 to current note
```

The nibbles can also form loop and end statements as follows:

Loop from pos XX: FE XX

End of table: FF FF

Again - all four nibbles must be programmed in sequence.

Example for programming the first table:

```
DP: 01 76 0f 04 08  ; Program first table position with Noise
DP: 01 76 0f 00 00  ; Ringmod and Sync deactivated
DP: 01 77 0f 04 02  ; The note of this position will be fixed to note
DP: 01 77 0f 00 0d  ; with index 45 ($2d) = A-3

DP: 01 78 0f 04 02  ; Program second table position with Sawtooth
DP: 01 78 0f 00 04  ; Ringmod activated, Sync deactivated
DP: 01 79 0f 04 0c  ; The note of this position will be three index
DP: 01 79 0f 00 03  ; over what is received from the keyboard.

DP: 01 7a 0f 04 0f  ; End of table (FF FF)
DP: 01 7a 0f 00 0f  ;
DP: 01 7b 0f 04 0f  ;
DP: 01 7b 0f 00 0f  ;
```

Notes