>Talko_

Die Stimme der Maschinen

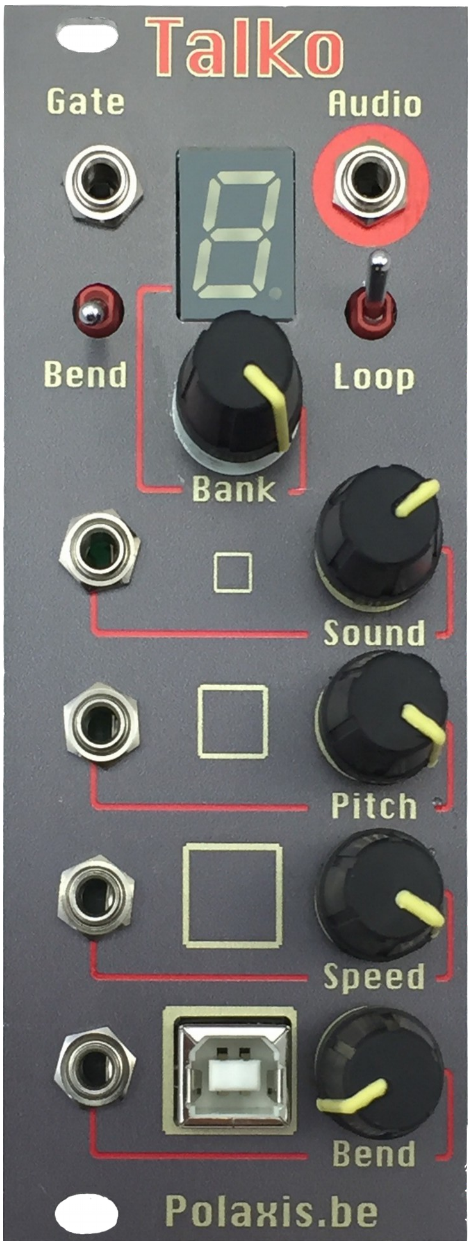


Table of Contents

1 Short description

This module is performing real time LPC synthesis to produce intelligible voice or … not if you choose to.

The module produces sounds in 3 major mode: Speech, Repeat and VCO.

In **Speech** mode, the speech starts with a gate signal and complete before waiting for a new gate signal. The speech has the priority.

In **Repeat** mode the speech starts and stops with the gate signal going high or low. The gate has the priority and the speech repeats while the gate is high. This mode is very useful to create crazy rhythms.

In **VCO** mode, the LPC engine loops while the gate is high, producing steady notes.

The VCO mode can also produce sound using white noise instead of tones, making strange throat sounds.

The sound synthesis is altered via CV signals to affect pitch, speed, bending.

Talko is an Arduino based synthesizer developed under open-source hardware and software. It's firmware can be updated via the onboard USB port, using the standard Arduino IDE.

The module can be powered via USB if needed.

2 Technical details

- format: Eurorack
- width: 10hp
- depth: 30 mm skiff friendly
- power: + 12V: 65ma . Talko can also be powered via USB when using a jumper.
- audio: un-filtered 5V PWM carrier at 62500 Hz, the output is protected against over or under voltage by clipping from 0 to +5V
- inputs: protected against over or under voltage by clipping from 0 to +5V
- CV attenuators potentiometer on each CV entry

3 Banks

A potentiometer is used to change the bank

0: Digits (from 0 to 9) male voice
1: Digits (from 0 to 9) female voice
2: Spell the Alphabet
3: Nato alphabet
4: Vocabulary
5: Big numbers
6: Voltmeter (reading the CV voltage from the Sound entry)
7: Frequency meter (just saying “Hertz” instead of “Volts”)
8: Counter (each gate trigger increments the counter while any change on the Sound entry resets it)
9: Number radio (each gate trigger a 4 number sequence while any change on the Sound entry triggers a new “transmission”)
10: Timer : time elapsed since the last cv reset (each gate triggers telling the time in minutes/seconds while any cv change on the Sound entry resets the clock to zero)
11: Random vocabulary (62 words)
12: Sentences
13: French vowels “a e i o u”

4 Mode operation

The modes are set via a combination of the Bend and Loop Switch:

Bend	**Loop**	**Mode**
Off	Off	Speech
On	Off	Speech + Bending effect
Off	On	Repeat mode
On	On	VCO mode

in VCO mode :

- Speed is not active (unless Bending is turned to zero, see below)
- Bending becomes note pitch
- when Bending (note pitch) is turned to zero, the LPC tone is replaced with the white noise generator and the Speed affects the noise volume.

5 Panel

Please note that all CV pots are also acting as voltage attenuators: when using a CV signal you may want to turn the knob fully clockwise unless you want to attenuate the signal.

5.1 Sound jack & pot

CV signal in allows to choose the words or phrases to be said.
Any change on the Sound entry reset the counter in bank N°8 and start a new transmission in Bank 9.

5.2 Gate jack

A high level high start the complete synthesis in Speech mode. (regardless of its length)

If the mode is set to Repeat mode , the gate will start the sound and repeat it as long as the gate level stays high.

If the mode is set to, VCO, the gate will start a tone and hold it as long as the gate level stays high.

5.3 Loop Switch

Choose between Speech mode Repeat mode or VCO mode.

In Repeat mode, the gate has the priority and the speech is active only while the gate level is high. Playing with the gate length or duty cycle allows making nice shopping effects.

In Speech mode, the speech has the priority and will complete before starting again on the new gate signal

5.4 Bend Switch

If the bend is on, the bend pot … well, distorts the sound.
Setting Bend and Loop together activates the VCO mode (see table Mode Operation)

5.5 Speed jack & pot

Change the speed of the voice

5.6 Pitch jack & pot

Change the pitch of the voice and change the noise volume in VCO mode

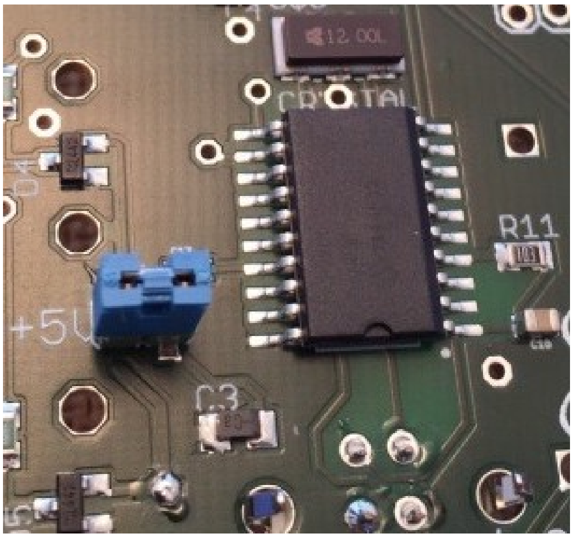
5.7 Bend jack & pot

Change the amount of bending or the pitch of the tone in VCO mode.

6 Hardware connections

6.1 USB

The USB port can be used to power the module.
In order to to so, the jumper has to be placed like this :



6.2Connection to the modular system (Doepfer Bus)

The module is powered via a ribbon cable.
Please make sure to respect the polarity and place the redline as marked on the PCB. (there are no protecting diodes)

7 Software upload

7.1 Connect COM Port

The com port MUST to be recognized by the computer. (If not, you won't be able to communicate to the module and upload anything.)

On Mac you just plug the module and a new com port shows up.

On PC you must install the MCP 2200 driver to create a com port.

The Windows driver can be found here :
http://ww1.microchip.com/downloads/en/DeviceDoc/MCP2221%20Windows%20Driver%202014-10-09.zip

You will need to unzip that file and start the driver installation

To do this : Double-click on the executable file within the appropriate folder for your system (x86 for 32-bit OS and x64 for 64-bit OS).

After installation, you must have a new COM showing up when you plug the module.

You can verify this by opening your control panel and plugging unplugging the module

If this works, leave the module connected and move to the next step.

7.2 Install the Arduino IDE

Windows : Install the old Arduino IDE **1.0.6** from the Arduino web site.
Mac : Install the Arduino IDE 1.6.X from the Arduino web site

Close the Aduino IDE for the moment.

7.3 Power the module

The module needs to be powered to have the MCU programed. You can do this using the Eurorack cable OR using the +5V temporary jumper that will provide power from the USB.

Note that the 7 segment's display may show random digits. This is normal.

First test : Upload the blink example.

Under the Arduino IDE, set this module as an Arduino Uno board using the "Tools/Board" menu

Pick the appropriate USB port discovered at Step 7.1 using the "Tools/Ports" menu

Verify that the programmer is set to AVRISPMK2 under Tools/Programmer

Open the blink code provided by the Arduino IDE.

It is located under Files/Examples/01.Basics/Blink

This code ALWAYS compile if the IDE is properly installed.

Now try to upload the code to the module.
The 7 segment's dot MUST blink at 1 sec rate.

If it does, you can move to the next step

7.4 Install the Time library

Close the Arduino IDE first
The code uses a time library to tell the time since the last gate trigger in bank 10.
The library is located here : https://github.com/PaulStoffregen/Time/archive/master.zip

Unzip and rename the Time-Master folder to Time before moving to your libraries folder
Copy the entire Time folder under your Arduino/libraries folder

If you forgot how to install a library, check this out : https://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use/arduino-libraries

Compile/upload a Time library example

Start the Arduino IDE
Create a new empty sketch (File/New)
Paste the following code I made for this test :

```
#include <Time.h>

void setup() {
Serial.begin(9600);
}

void loop(){
digitalClockDisplay();
delay(1000);
}

void digitalClockDisplay(){

Serial.print("elapsed time since reset ");
Serial.print(hour());
printDigits(minute());
printDigits(second());
Serial.println();
}

void printDigits(int digits){
// utility function for digital clock display: prints preceding colon and
leading 0
Serial.print(":");
if(digits < 10)
Serial.print('0');
Serial.print(digits);
}
```

The code must compile without error

Try now to upload that code.
Open the Arduino's serial monitor (Tools/Serial Monitor)
It should display something like :

```
elapsed time since reset 0:00:04
elapsed time since reset 0:00:05
elapsed time since reset 0:00:06
```

If this works, close the Arduino IDE.

7.5 Upload main code

Unzip the Talko code from here : https://github.com/deladriere/euro-modules/blob/master/Talko/Software/ TalkoVCO/TalkoVCO.zip

Create a TalkoVCO folder under your Arduino/sketches directory

Copy TalkoVCO.ino, talko.cpp, talko.h to that folder
Open the Arduino IDE
Open Talko's code : File/Sketchbook/TalkoVCO
Compile it
Upload it

8 Appendix

8.1 List of phrases

```
0: “set the control for the start of the one”
1: “C time slow”
2: “start the fire
3: “smoke the circuit”
4: “turn past the time”
5: “stop the machine”
6: “too low for you”
7: “fire and ice”
8: “heavy machine”
9: “new ground circuit”
10: “repair the machine”
11: “Wir Sind die Roboter”
12: “Error Instruments”
```