

## 8 canal AD converter using a PIC18F4550



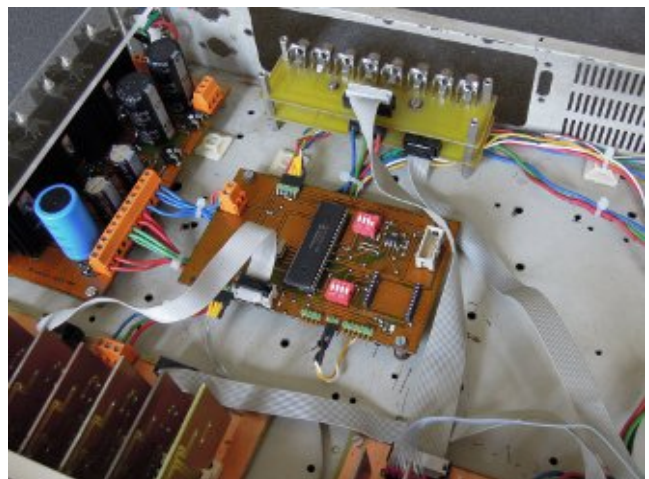
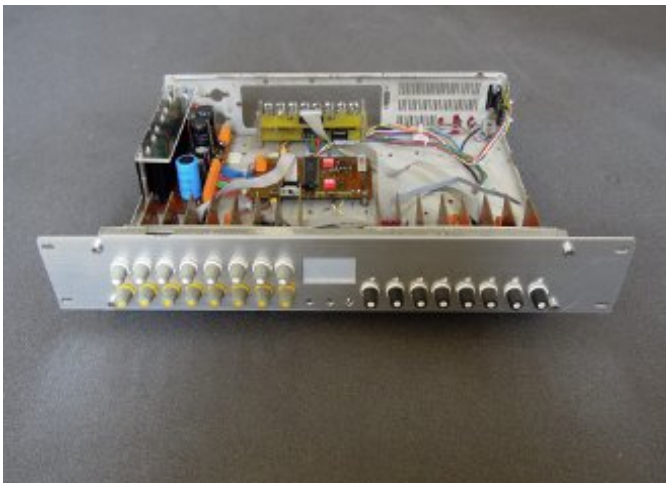
Interface for 2 \* 8 inputs

photo el Ravallero

The electronic circuit converts analogical signals into digital signals. One can plug in microphones, connect line audio or a piezo transducer .

On independent inputs (channels) the unit includes 8 preamplifier, 8 envelope followers and a PCB for a PIC18F4550. The PIC has the function to read the analogue events and to send it digitalized using its USART to the computer.

With the object [comport] of Pure Data the USB o RS232 is read and transformed in MIDI or OSC events.



The circuits are mounted in a chassis (rack standard) with 8 differential preamplifiers, 8 audio preamplifiers, 8 envelope followers and a power supply filter to get a hum-free audio signal.

General description:

The actual project is the successor of the already made unit "[PAD to MIDI](#)". The old unit, made view years ago, uses a PIC16F877. The source code was written in assembling (ASM) uff ... but it works (still) very fast.

The code for the new PIC18F4550 is written in C (BoostC) and it can be downloaded with all other files (electrical schematics, descriptions) at the very end of this page.

Compared to the huge code of the 16F877 the new code for the 18F4550, written in C, is much simpler.

While the "firmware" of the 16F877 project reads 8 analogue inputs and converts the results to a MIDIprotocol the new code for the 18F4550 only reads the 8 analogue inputs and sends the results to its USART to a MAX232 (level shifter) connected by a cable to the RS232 port of the computer.

In case to not have any more a RS232 input (all Laptops) there are cables (RS232 to USB) or one can make a small PCB using a FT232 USART to USB converter. [See here](#).

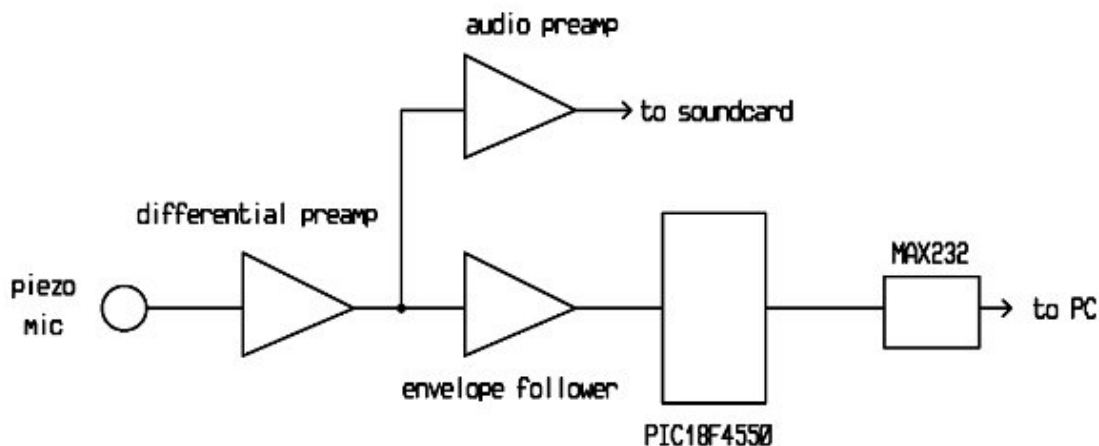
The reason to use a PIC184550 was principal to have a PIC with the ability to connect the PIC directly to a USB port using the its internal register. Still I didn't program the source code to have this ability of the PIC.

For further use of the incoming (raw) data sent from the PIC, I use Pure Data (the download file includes a patch). With Pd one has a lot of possibilities to interpret the events. They can be used as MIDI, or OSC events to manage other programs (SuperCollider, Resolume) or one can send the data with [netsend] to another computer. Interesting is the use of tables, [tablewrite], [tableread] to change incoming data in may different ways.

If you prefer to program in Prozessing or SuperCollider environment instead of Pure Data one can obtain the results of course.

The firmware of the PIC18F4550 works indeed very fast. In the Pd patch there id a "cluster" button allowing to send data without any stop byte. With a simple MIDIinterpreter (MIDI\_OX for example) one can generate 50 MIDINote-On messages with only one stroke. This is a tremendous sound cloud achieved only with a simple MIDItool.

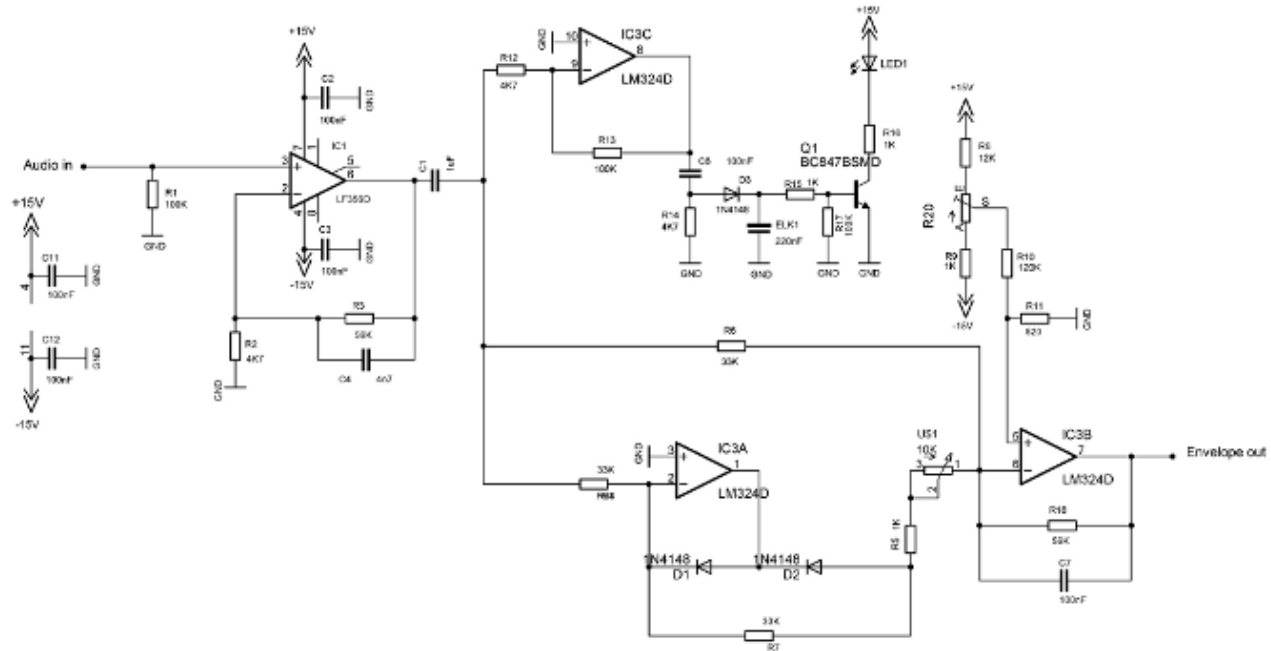
Block-Scheme of the circuitry:



The above Block-Scheme shows the signal flow of one channel, but of course there is only one

PIC18F4550 and one Level-Shifters (MAX232).

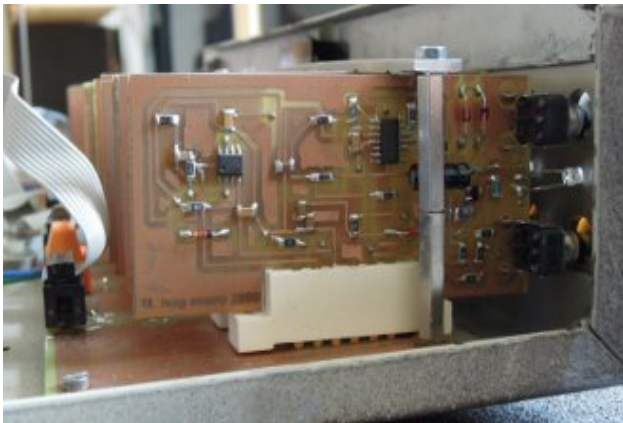
The envelope follower (for one channel)



The original design of the envelope follower you find [here](#). I added a potentiometer (R20) to get the ability to adjust the output voltage, pin 7, of the LM324. The “free” IC3C I use to have a visual event indicator with LED1. The indicator could be designed significantly simpler with only one resistance of 1k and a LED connected to pin 8 of the IC3C.

Together with the differential pre-amplifier I get quit good results. It is a good idea to put a 100 ohm resistance with a 5,1V Zener diode to the output (pin 7) of the IC3B to protect the input of the 18F4550. (the reason is the to high +15V, -15V of the envelope follower for the inputs of the PIC)

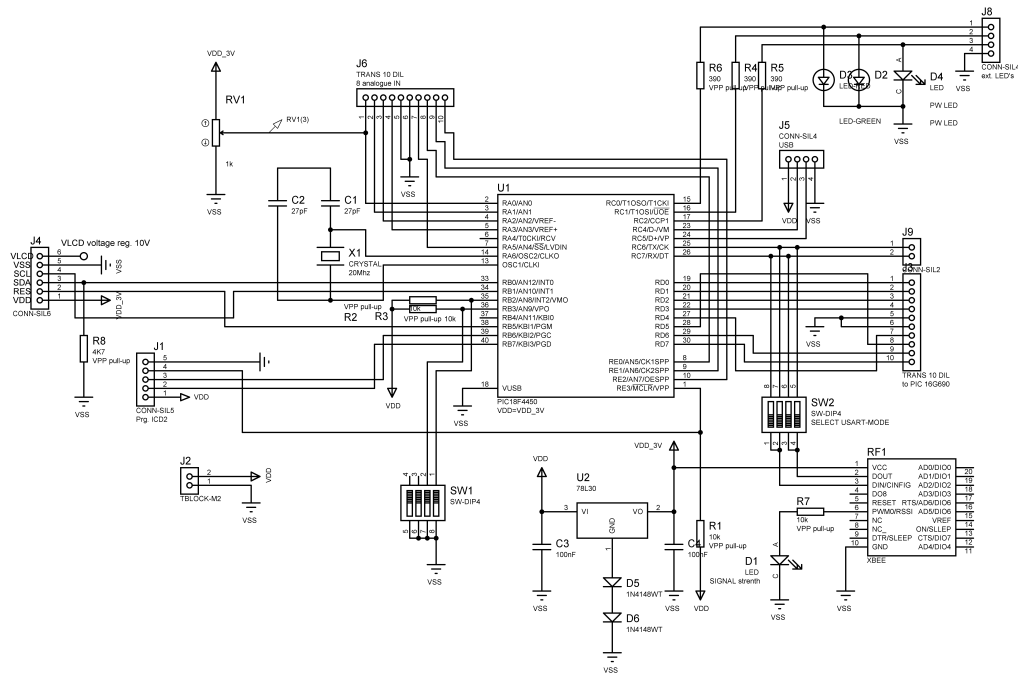
The 8 envelope follower units are designed for easy plug into a base PCB. (see photo).



Each unit is connected using “very” old computer plugs. I got a lot of them without even paying them, but for my purpose they work good enough.

The envelope followers are made with SMD devices.

## The PCB with the PIC184550



for better quality see the schematics attached.

When I designed the PCB I put view connectors without use to the PCB. With this connectors one could connect a xBee unit or a serial LCD to the I2C bus of the PIC. What we really need for the project is only J6 for 8 analogue inputs, J1 to connect a PIC programmer and the 3 LED's to have some visual control. I use a ICD2 clone to program the PIC18F4550.

Same important code of the PIC18F4550 firmware.  
The crystal is external and runs with 20Mhz.

```
//initial function for the RX/TX registers
void init_TX_RX()
{
    char dump; // aux Var to empty RX_buffer

    rcsta = 10000000b; // set SPEN bit,serial port enabled & CREN disabled.
    txsta = 00100100b; // TXEN enabled, BRGH = 1, TX9 8-bit, Async mode.

    baudcon = 010001000b; // BRG16 = 1 (010001000).
    spbrg = 42; // 42 for 116,279Kbaud)
    dump = rcreg; // empty RX_buffer.
    dump = rcreg; // empty RX_buffer.
}
```

//extract of code for channel 3. Only if the expression of the first line is "true" data is sent to the USART of the PIC

```
if (!test_bit(on_off,3)) && (Vel_3 >= TH_3) || (Vel_3 >= aux_3 + TH_3))
{
    del_3 = del_3 + 1;
}
```

```

if (del_3 >= del_On_3) // short delay to adjust the envelope knee at the beginning.
{
    set_bit (portc,1); // white_LED on.

    aux_3 = Vel_3; // aux_3 used to re-trigger Note. if value is higher than Vel_3.

    set_bit (upper_3,4); // select Ch_3: 10000b = 16.

    init = 0x00; // init byte, sets in Pure Data the gate_object.
    txreg = init; // 1st byte =0x00 initialize Pure Data.
    while(!trmt);
    txreg = upper_3;
    while(!trmt);
    txreg = lower_3;
    set_bit (on_off,3); // indicates that data is sent.
    del_3 = 0;
    clear_bit (portc,1); // white_LED off.
    Vel_3 = 0x00;
    return;
}

```

Uff ... it looks weird but it isn't ...

The function shows how 10bits of channel 3 are sent. The transmission includes in total 3 bytes: The `init` byte, always 0x00, the bytes `upper_3` and `lower_3`.

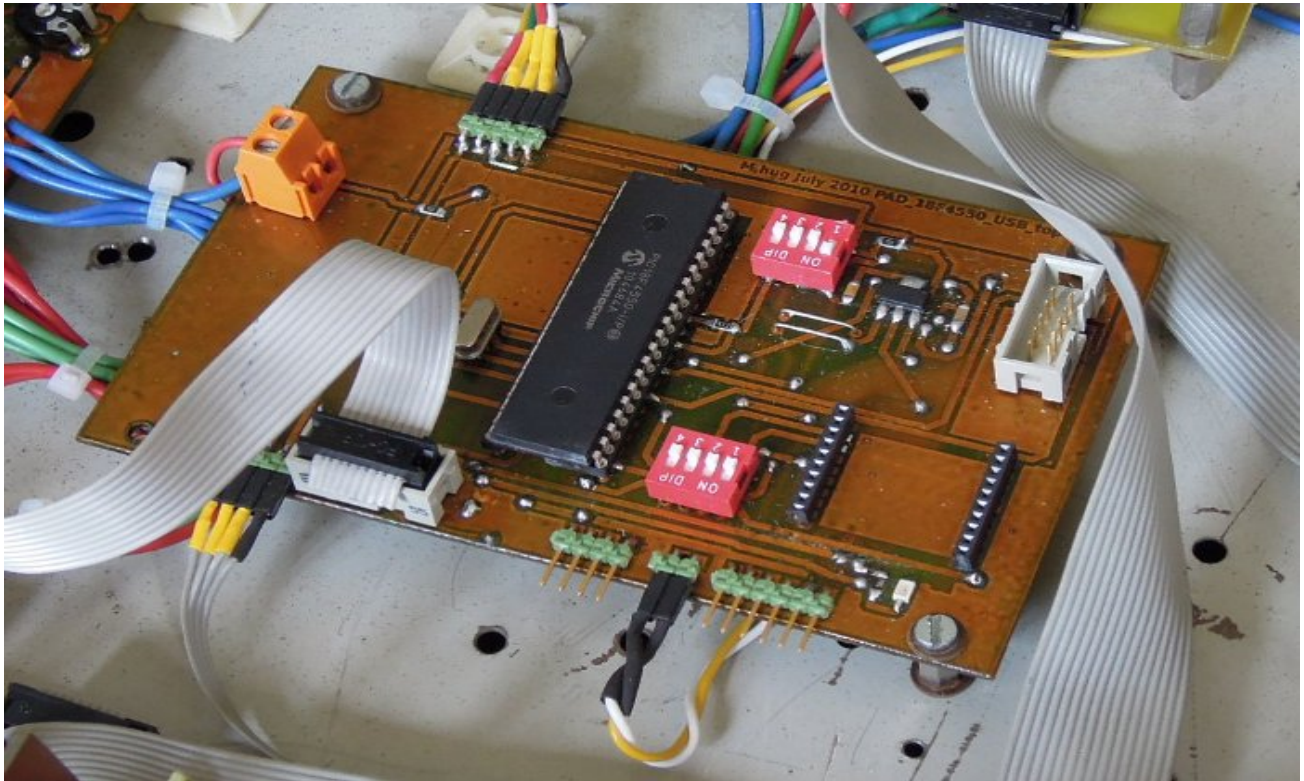
The 6 unused bits of the upper byte of the 10bit AD conversion are used for direction purpose. So, looking at the code above the "set\_bit (upper\_3,4)" writes the address, so that Pure Data (or Processing) can extract the address included in the `upper_3` byte and assign the corresponding channel.

To determinate an event a forth byte is sent, As soon as the 18F4550 reads a 0V at the analogue input a byte with its corresponding channel address is sent.

In this way I avoid that the PIC sends permanently data as soon the voltage rises above 0V. The effect is similar to a MIDI-Keyborad: on pressing a key its corresponding MIDI Note-On message is sent and on releasing the same key a MIDI Note-Off message is sent.

The source code is written in C (BoostC) and is included in the download-ZIP at the bottom of the page.





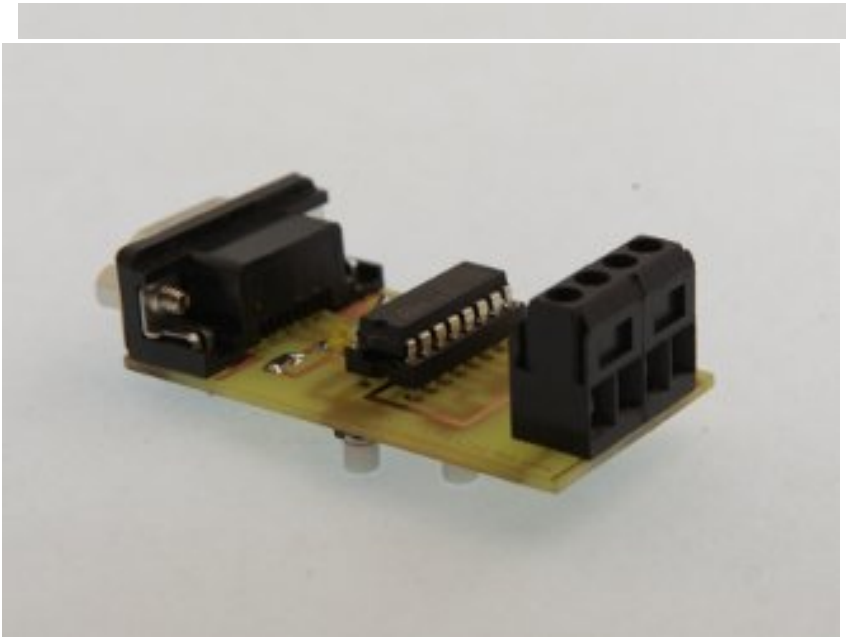
PCB with the 18F4550 and the “zocal” to put and xBee module if there is a need.



All the circuitry is mounted in a standard 19inch chassis. The unit can be screwed into a rack for easy transport.

The “level shifter”.

I made a small print to convert the data from the 18F4550 to a RS232 format. The PCB is done using a MAX232.

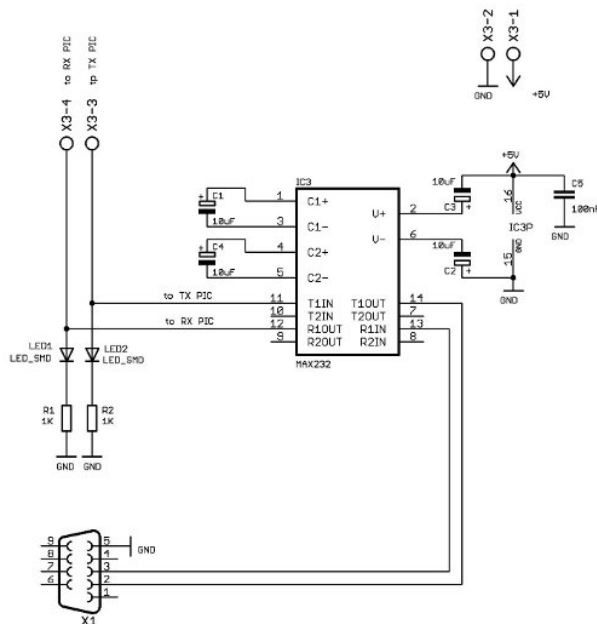


I reused the PCB with the MAX232 because I had it ready mad from another project.

But I recommend using a newer design with a FT232RL to connect directly to an USB port of the computer.  
<http://asxcasas/>

Small PCB with a MAX232. Nowadays it is recommended to use a circuit with a FT232RL.

## MAX232



TITLE: MAX232\_UART

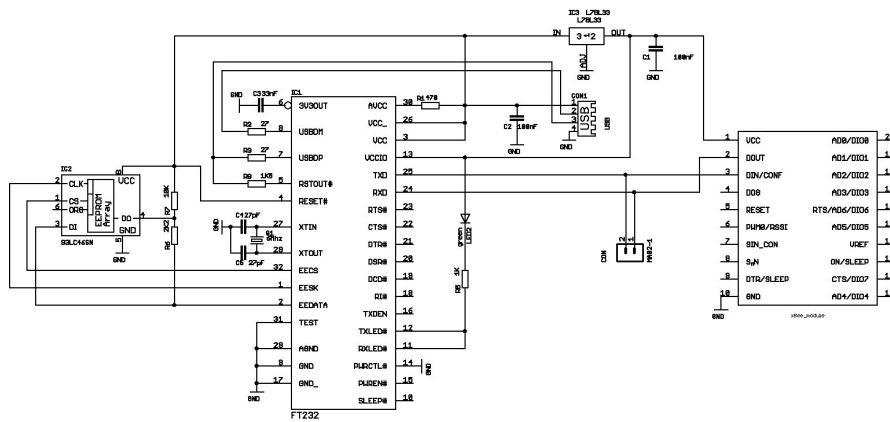
Document Number:

REV:

Date: 9/09/2009 07:53:14p

Sheet: 1/1

## FT232



TITLE: FT232\_USB

Document Number:

REV:

Date: 9/24/2009 04:50:55p

Sheet: 1/1

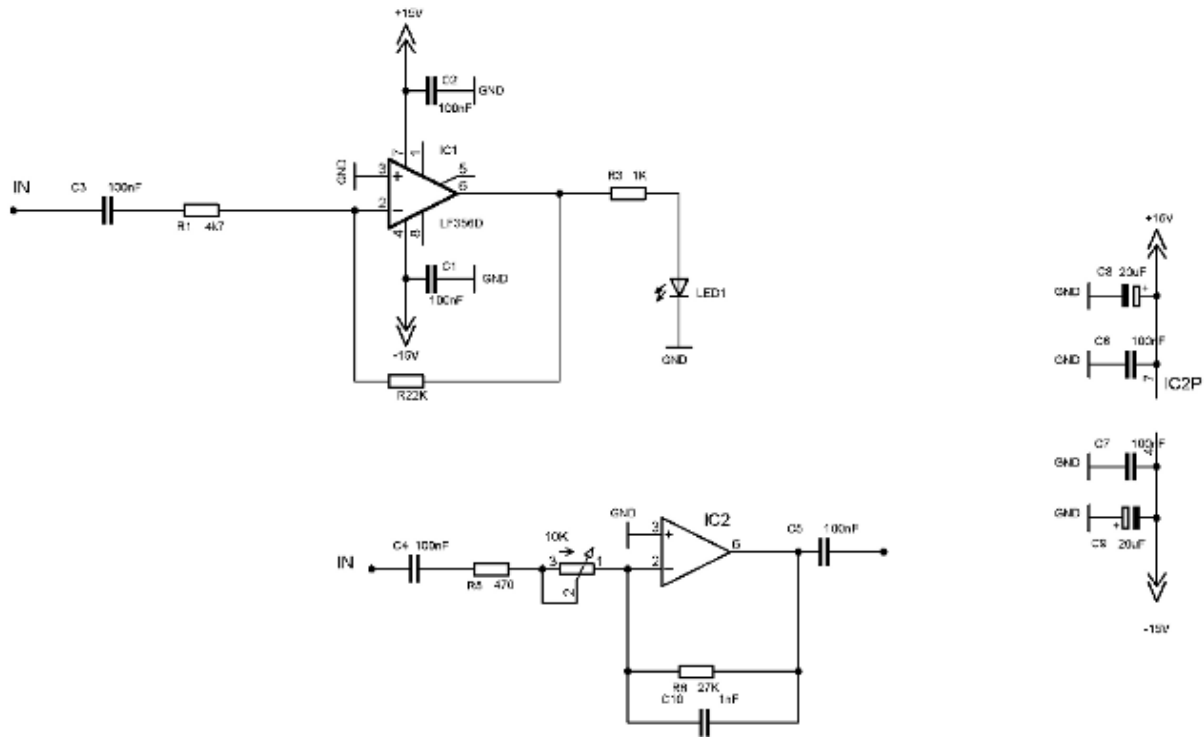
If the circuit is used without connecting a xBee the TXD/RXD of the FT232 is connected to the RX/TX of the PIC18F4550

The audio pre amplifier.

The schematics is really simple. The gain can be adjusted with the potentiometer of 10k between 2,7 minimum and 57 maximum. The pot. Is connected to pin 2 of the IC2.

The LED1 is an event indicator of the input values.





For sure in a future project I'll do all this circuitry above very small and the audio pre-amp will be omitted and the audio adjustments will be made only by software.,

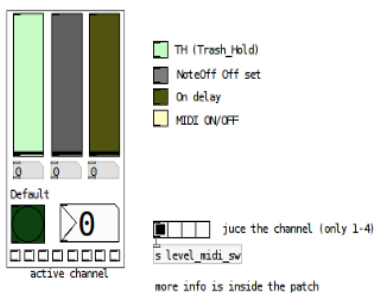
## Pure Data patch.

With the program of Pd one can adjust several parameters of the 18F4550.

Adjustable are:

- TH: minimum voltage that generates events. (like Trash Hold)
- NoteOff: below this voltage a final byte is sent.
- On delay: after how many program cycles a data package is sent?
- MIDI ON/OFF: not activated.

juice the channel:



The above values are assigned to its corresponding channel. Until now only channel 1-4 is programmed.