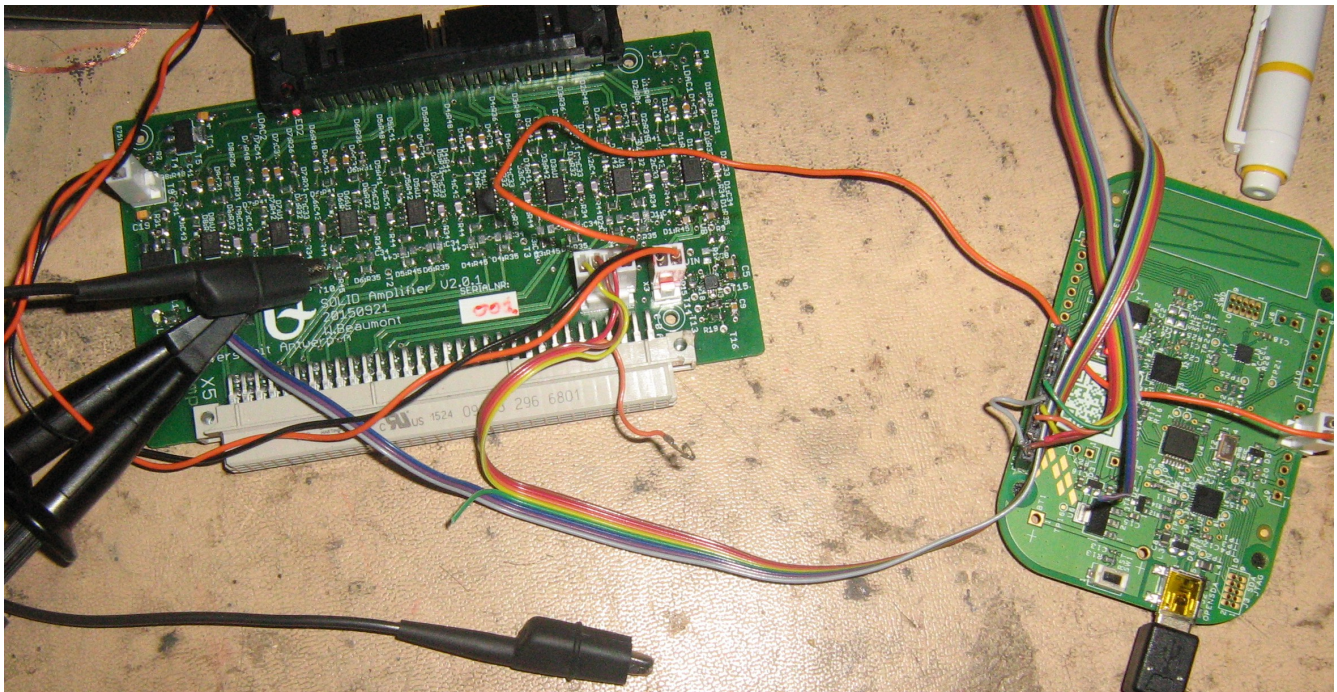


LV Power supply (low noise) between 6.5 and 7.2 V molex two pins

Bias voltage $< 80\text{ V}$, take $\sim 75\text{ V}$. Molex two pins "red" connector

Power the uP board with the USB cable of the computer.

Connect the cable to the board and uP as indicated.



I2C lines are pulled up to 3.3 V by a voltage divider to 5 V.

The pull ups (R33 , R32) are connected to the Vext. For the uP configuration this comes from the V5USB . (the single orange cable on the picture).

The gray cable on the connector on the processor is used to count the clocks and the green cable can control the LDAC line. This is used for setting the address of the MCP4728 . But the voltage on the line is too low (the trick was to lower the voltage on the analogue board and then put it back to 5V.)

The two resistors to ground are soldered on the SDA and SCL test pin on the back (on the picture the scope probes are connected to it.) .

For operations with the ADC boards :

- remove the two resistors at the SDA and SCL pins (on the solder side) (cross check that R37, R38 are not placed and R14 is placed).

The voltage of the ADC board (assumed to be 3.3 V) will be used.

In case of read /write errors one could lower the voltage on the I2C devices .
This can be done by changing the resistor R3.
- this has no consequences for the analogue path.
-But it will limit the voltage control range of the DAC's.

Addresses :

- DAC general bias adjust (MCP4725) 0b1100111 (production will be 0b110001)
- DAC trim channel 1..4 (MCP4728) 0b1100011
- DAC trim channel 5..8 (MCP4728) 0b1100101
- Temperature probe MCP9808 0b0011000

Functionality loaded software :

It sets different values for the DAC and then start increasing it by steps of one.
So each channel has an other voltage (see code).
The output gives the set values (in counts and voltage) and read back.

The temperature is read back only in values (no temperature conversion done up to now).

uP Firm ?/ Software

Main program.

I reused the "command code" that was also used in SM1.

For development there is some flexibility to send commands and read them back without making a new command interface. But this is only for "now".

But this is not tested.

Only the "heartbeat" part is tested.

What should be compatible is the device part of the software.

Check for the software the following links :

the used test environment :

<https://developer.mbed.org/users/wbeaumont/code/SolidAnalogueBrdctrl/>

<https://developer.mbed.org/users/wbeaumont/code/DevInterfaces/>

A simple application how to use the MCP4728 class :

<https://developer.mbed.org/users/wbeaumont/code/MCP4728test/>

The device classes

<https://developer.mbed.org/users/wbeaumont/code/MCP4725/>

<https://developer.mbed.org/users/wbeaumont/code/MCP4728/>

<https://developer.mbed.org/users/wbeaumont/code/I2Cinterfaces/>