

I²C Problems in LT: The Solution



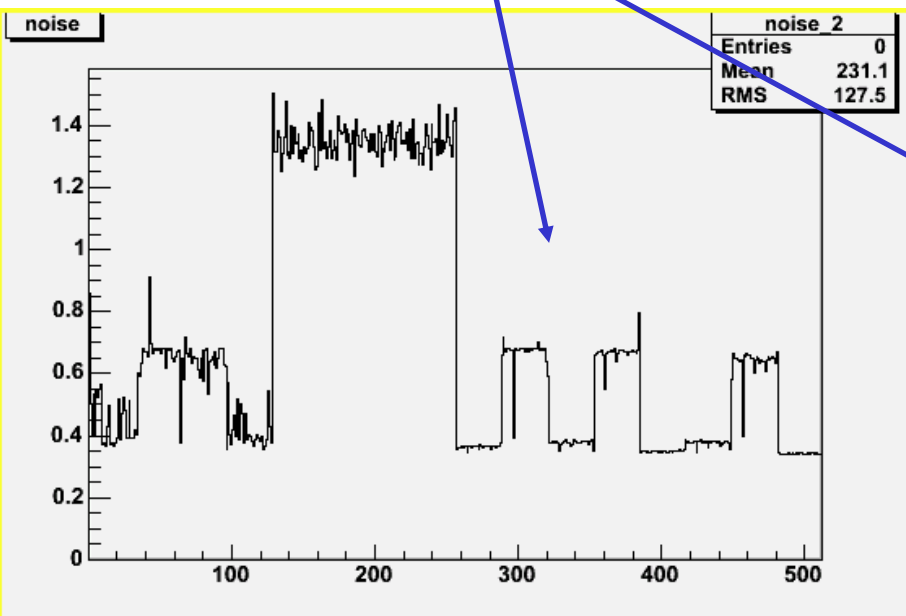
Marco Meschini
INFN Firenze





Introduction: The Problem

- As reported at last ModTest meeting modules were "lost" during LT in cold cycles
- Everything started observing a strange behaviour in APV noise, associated with error messages from FecServer - Modules excluded from DAQ



```
fec Exception: Mon Feb 16 17:42:20
2004
FecDevice::writei2cDevice(single
byte write extended (RAL) mode):
Register acces
FEC Functions error code: 205
Position: FEC 0x0 Ring 0x0 CCU 0x4
Channel 0x6 I2C address 0x3F
```



Introduction

- The same problem has been reported first in TIB centers, and then in TOB too
- **Is I²C responsible?**
- We tried to investigate this hypothesis by all possible software and hardware means
- We looked at almost all analog and digital lines directly with scopes during LT cycles
- **We identified the source of the problem in the APVs not responding to I²C communication under special circumstances**

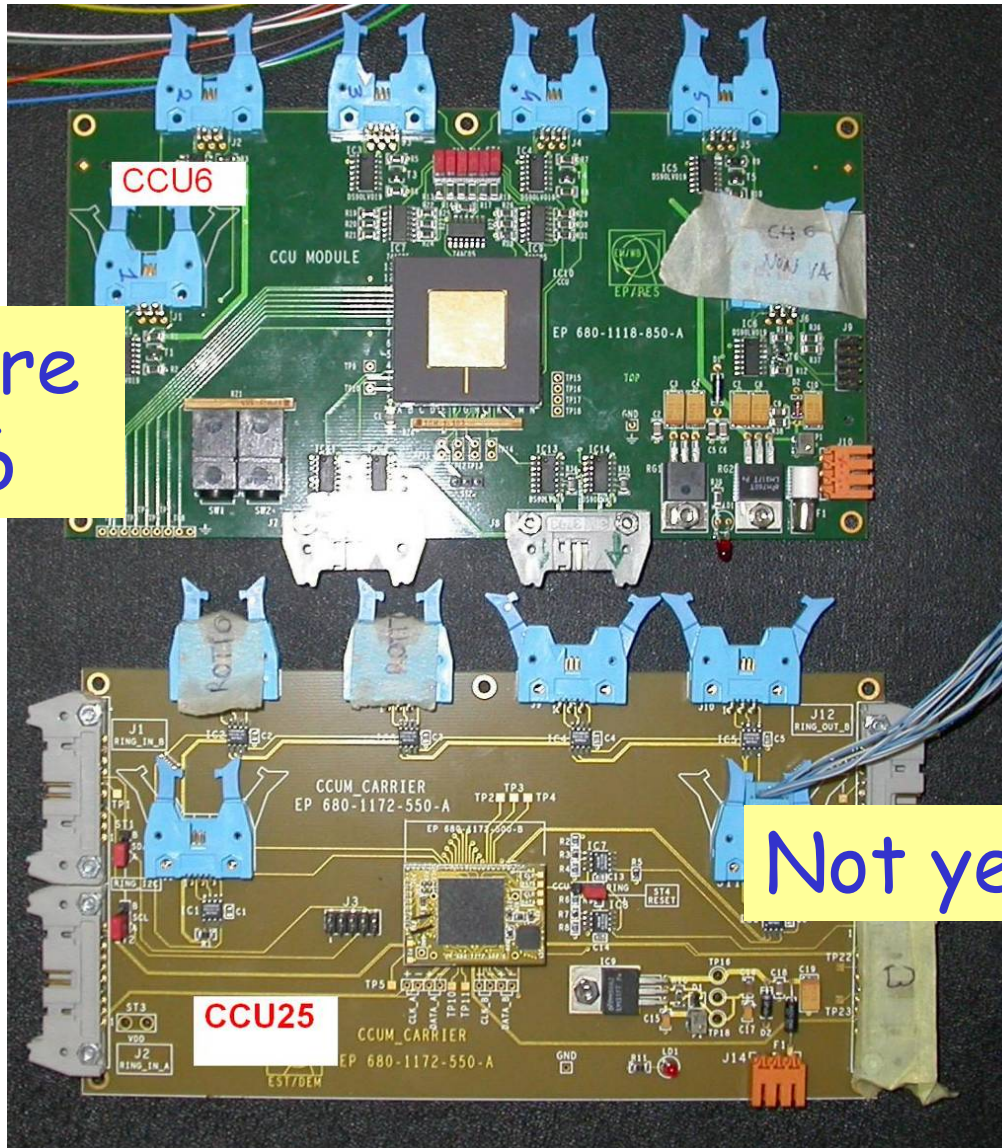


I²C Communication

- All the asics on the module (hybrid) use the same I²C bus to communicate
- Each communication between CCU and asics is a 2 bytes transmission
- Each byte has 9 bits: 8 data + 1 acknowledge (LOW signal) from the device receiving the command
- The I²C LOW level should be below 500 mV (APV User Guide 2.2)

- The first byte is the address of the chip
- The second byte is the register to be read/written

CCUs



In LT we are using CCU6

Not yet CCU25!!!



I²C Bus Configuration

Pull-up resistors

Hybrid side

CCU side

Level shifter

Scope

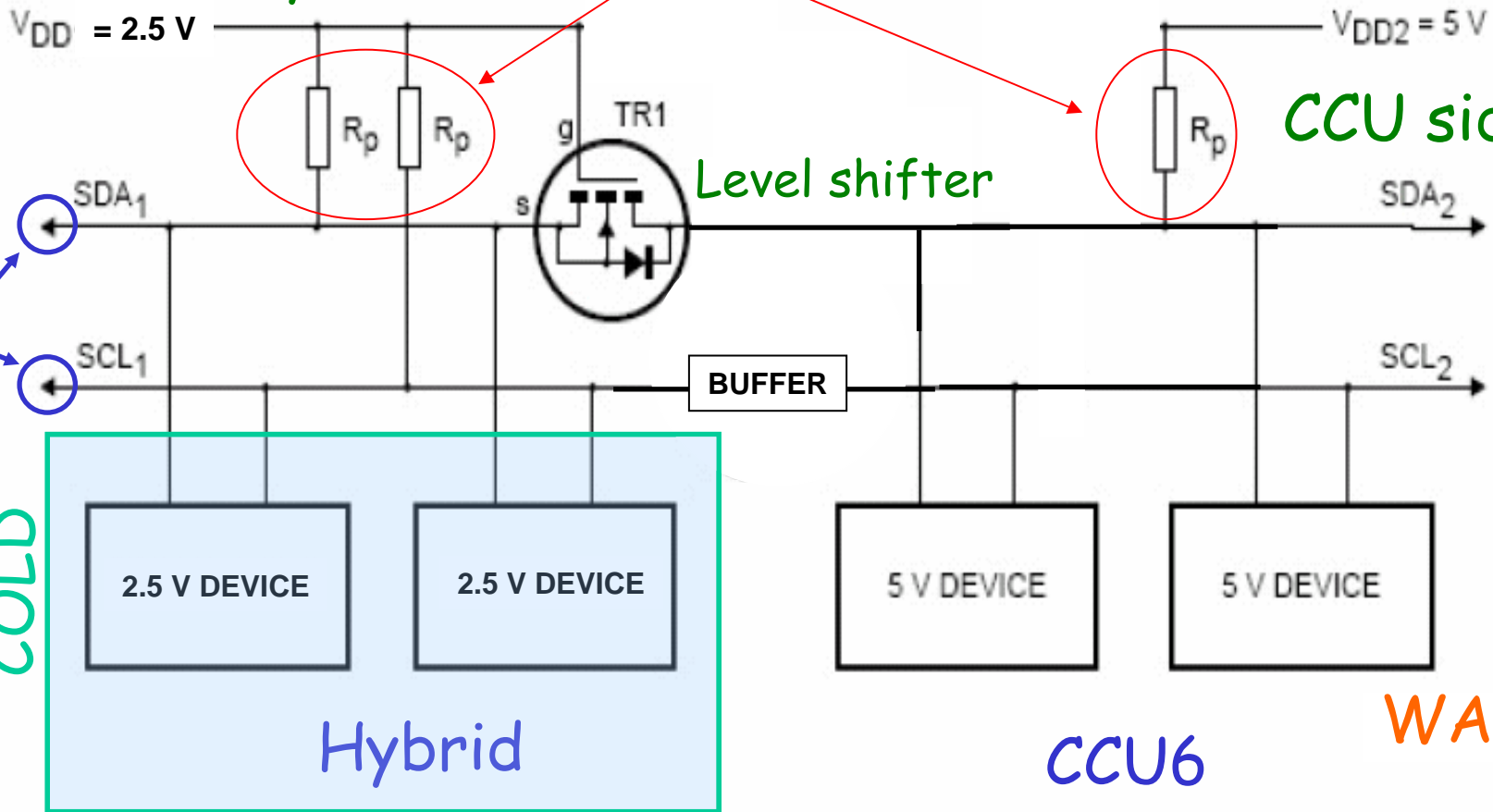
COLD

Hybrid

CCU6

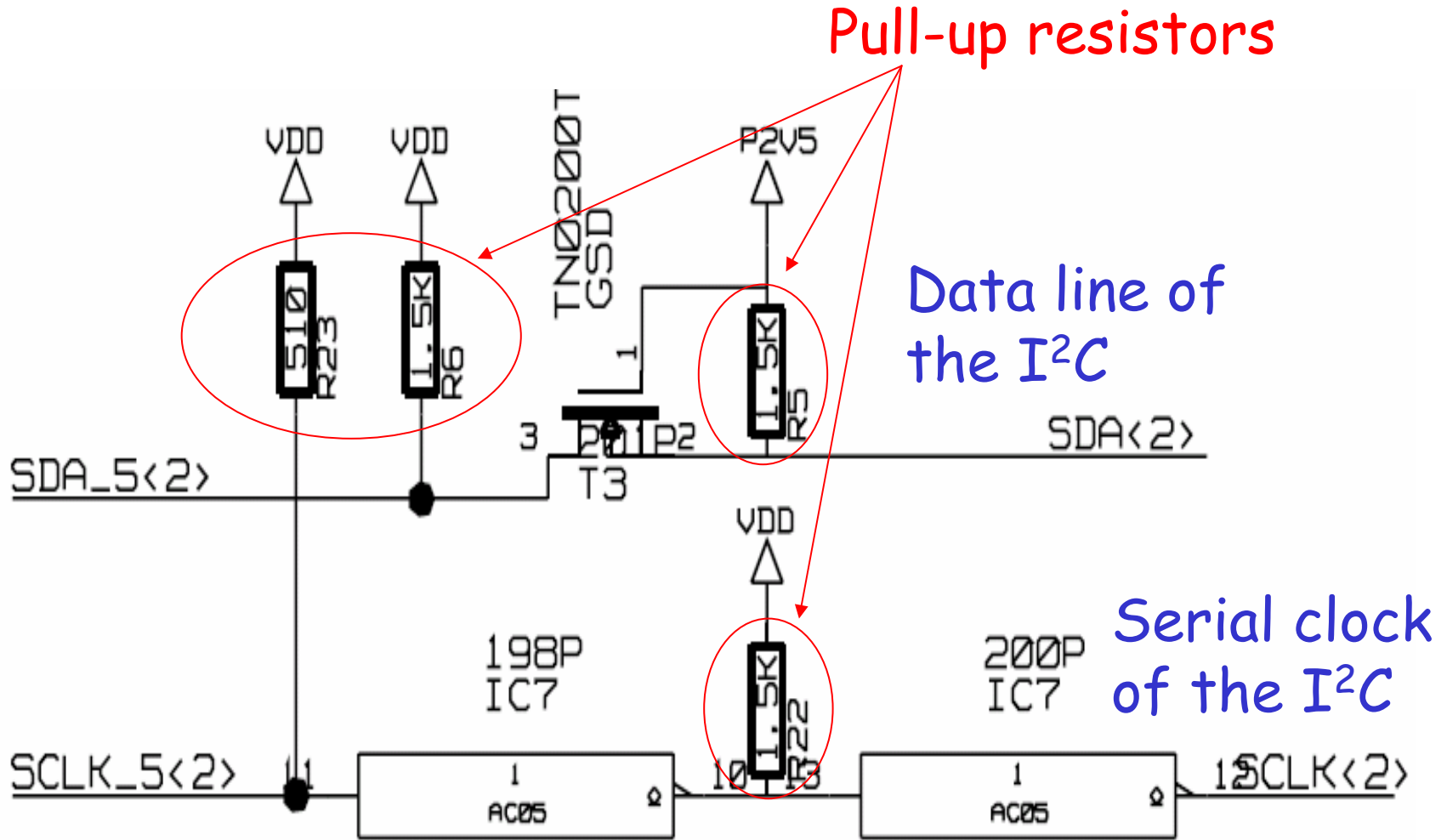
WARM

From Philips I²C Bus Specification





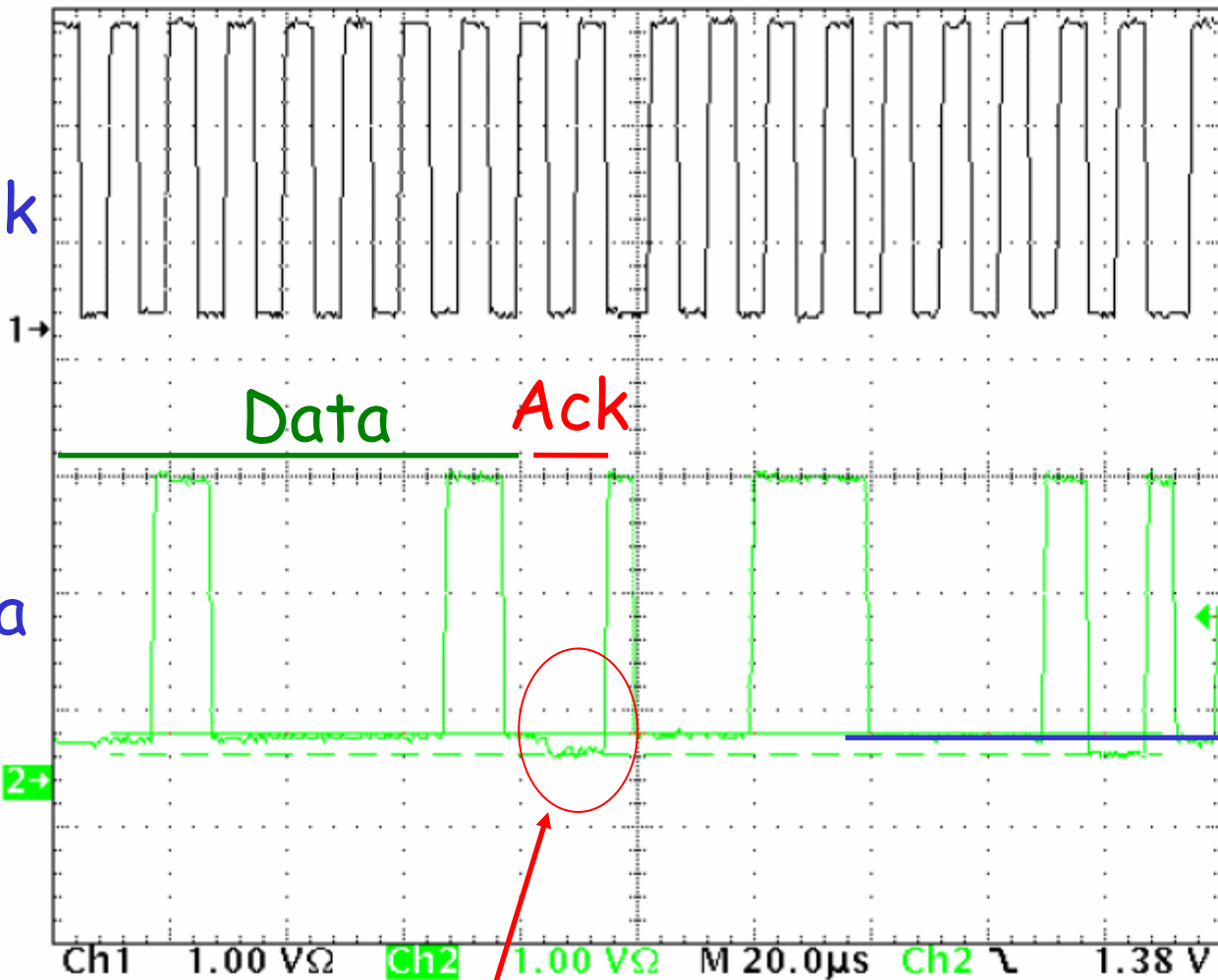
CCU6 Schematics





The Standard Situation

SCL - Clock



Δ: 180mV
@: 380mV

SDA - Data

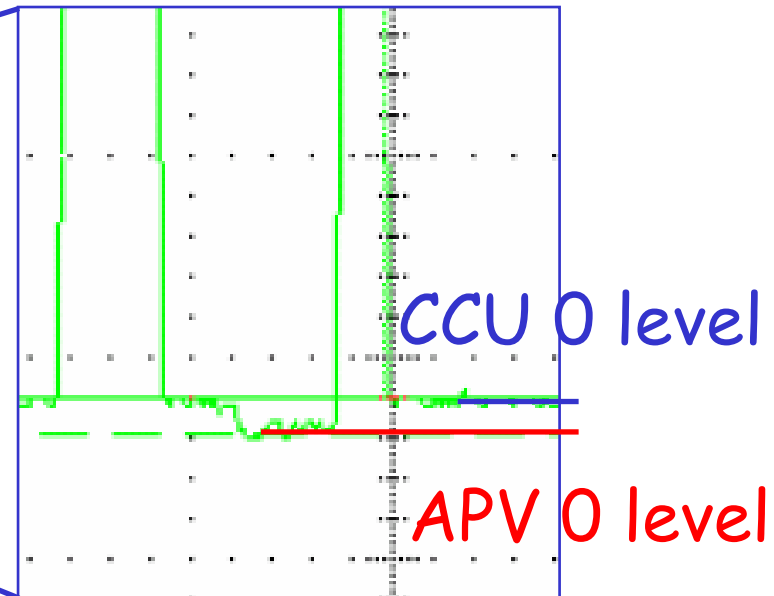
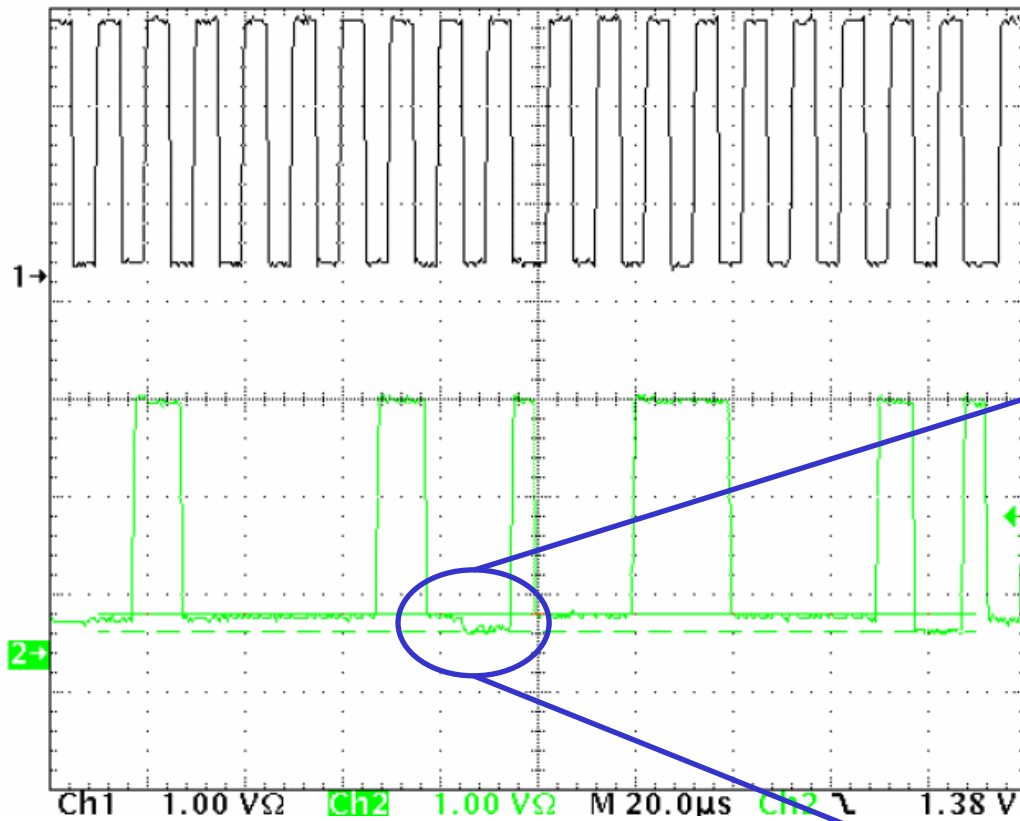
380 mV

28 Feb 2004
23:20:10

Acknowledge bit = 0 means that the APV is understanding the I²C communication

APV Answer

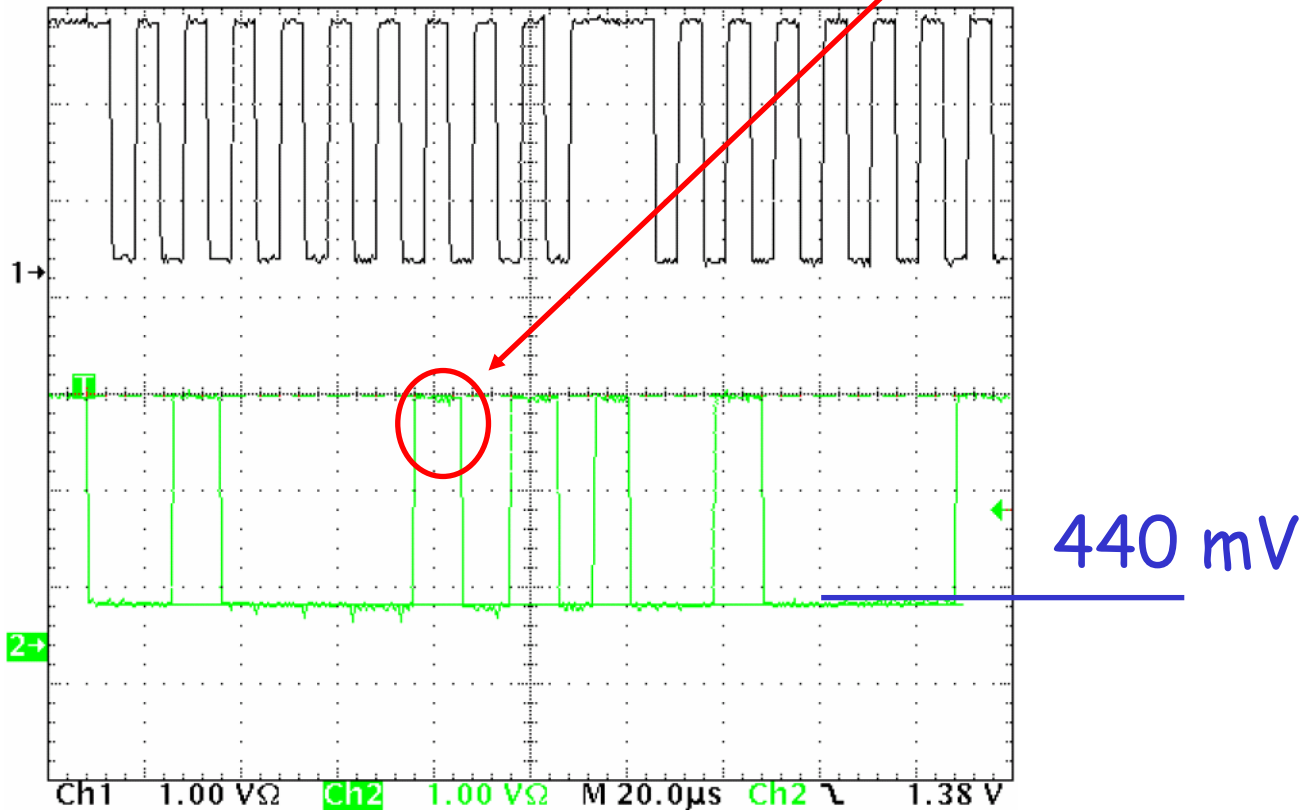
The LOW level of the I²C data line is lower when the APVs are asserting on the bus



Depending on the pull-up R_p it can happen that the CCU6 does not pull down enough the I²C line

A Bad Situation

- In this case the acknowledge bit is HIGH
- The APV is not understanding the I²C signals and then it goes into an unpredictable state

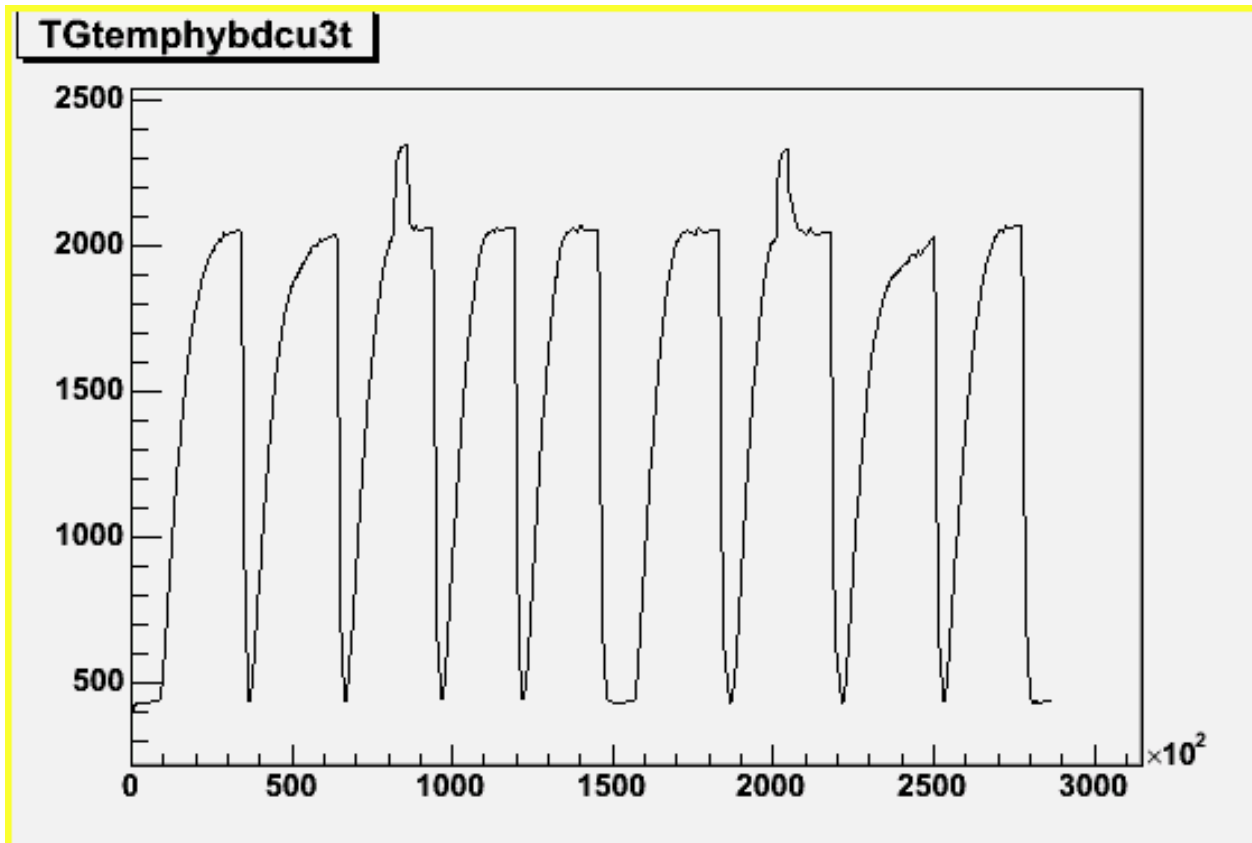




But I²C on DCU works!



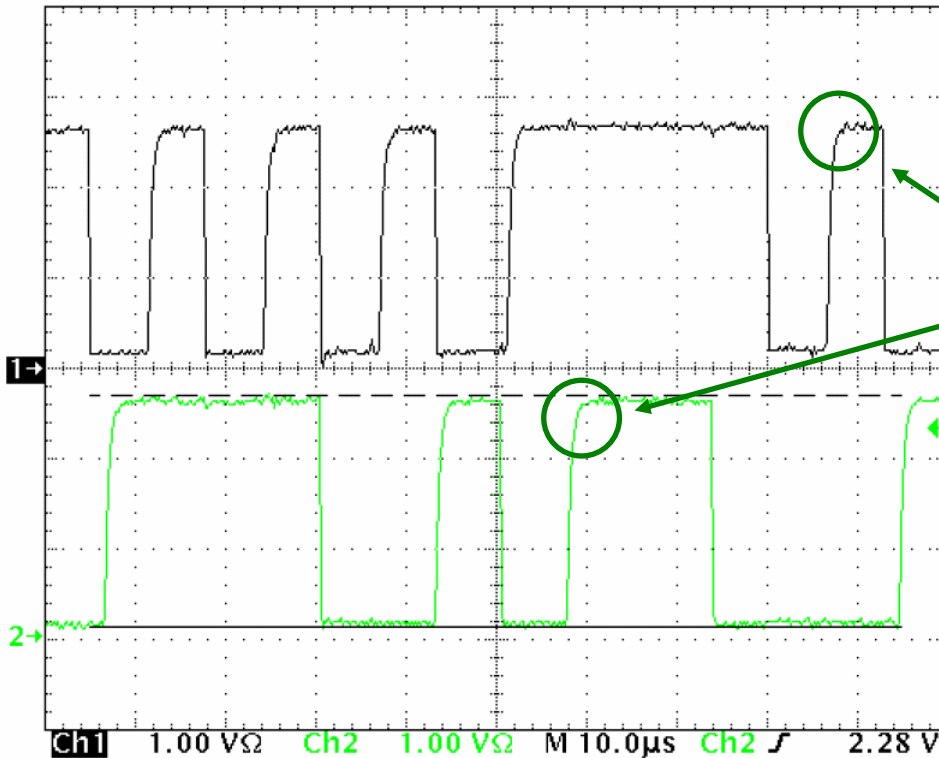
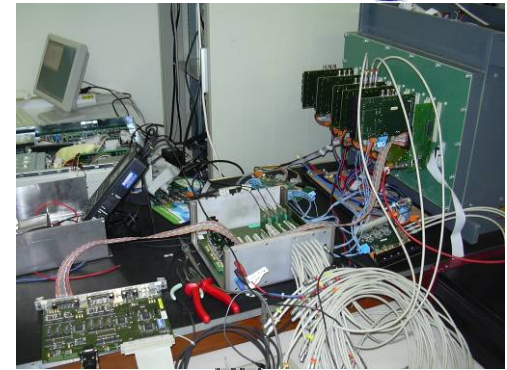
The DCU readout via I²C shows no problem both at +20 °C and -20 °C during a full 3 days scenario in the cold box



Readout of the NTC thermistor on the silicon

And then with ARC...

- ARC connected to the cold box
- Modules at $-20\text{ }^{\circ}\text{C}$
- Different I²C driving
- I²C signals are recognized correctly



Even a slower rise time wrt the CCU6 setup has no bad influence



Minimum Pull-up Resistor

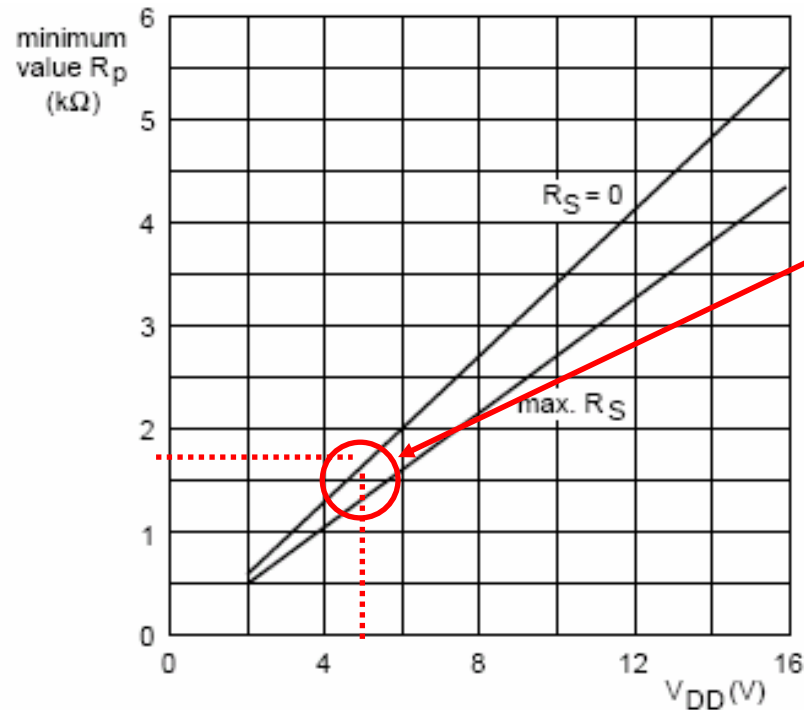


Fig.37 Minimum value of R_p as a function of supply voltage with the value of R_S as a parameter.

From Philips I²C Bus Specification

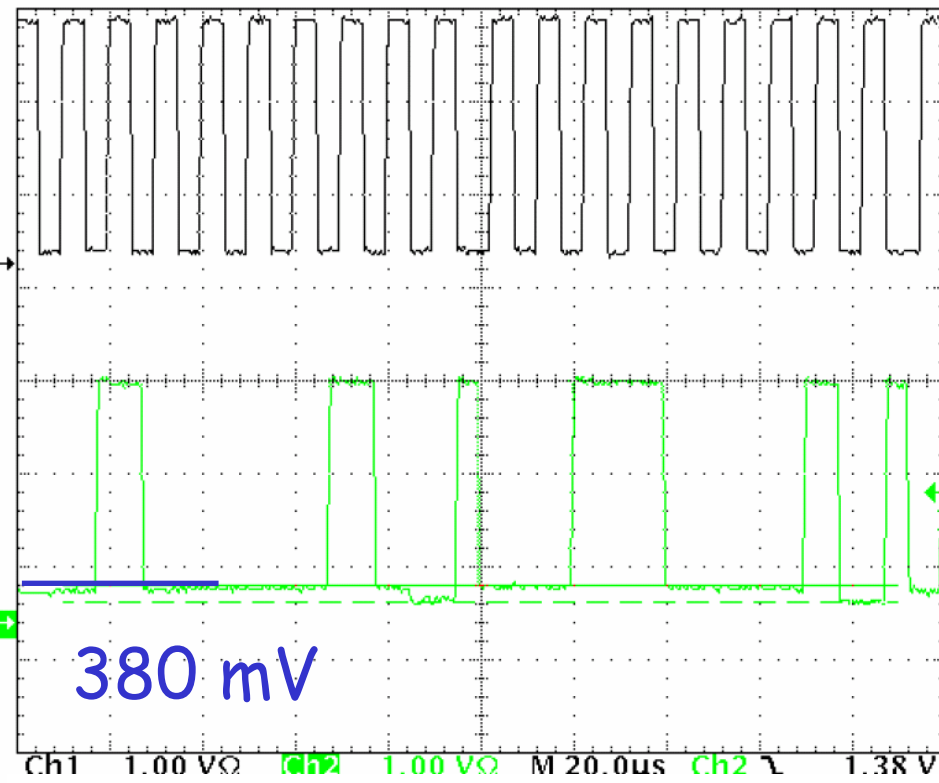
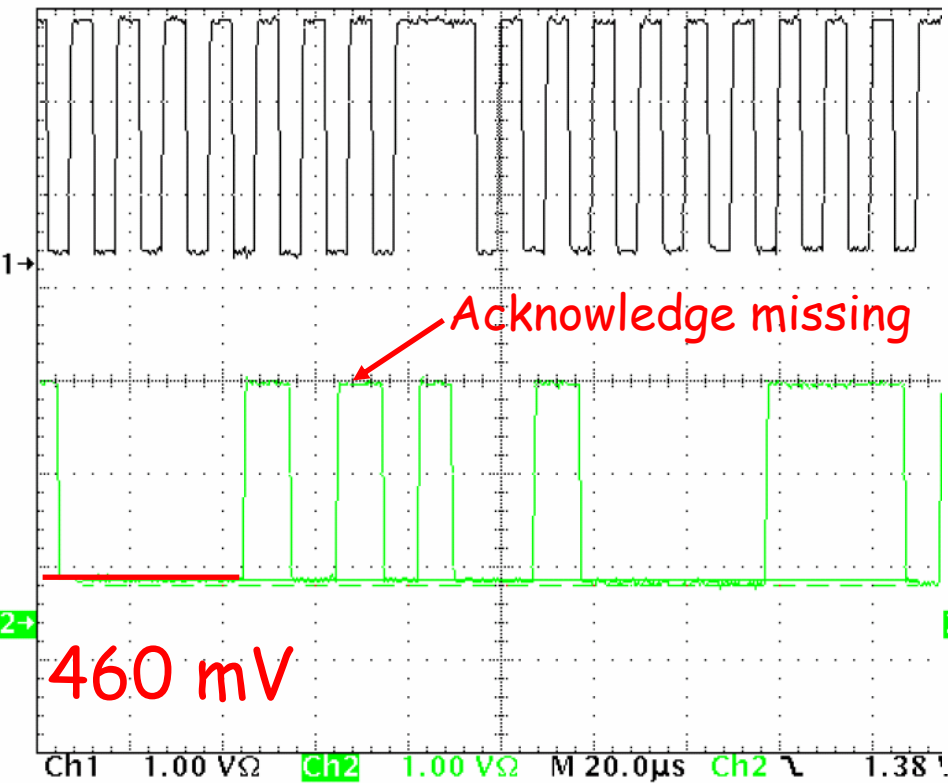
- In our case $R_S = 0$ so the minimum R_p value is higher than 1500Ω (original pull-up resistor on CCU6)

So we decided to try changing the pull-up resistors (using trimmers) to understand the LOW level behaviour



APV Behaviour

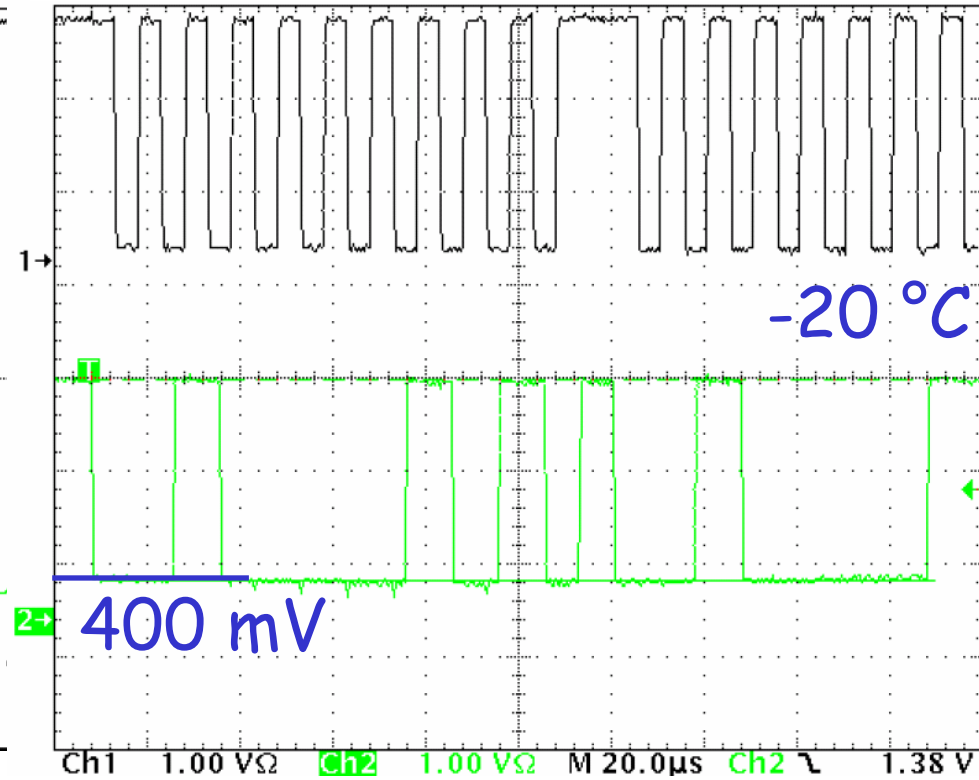
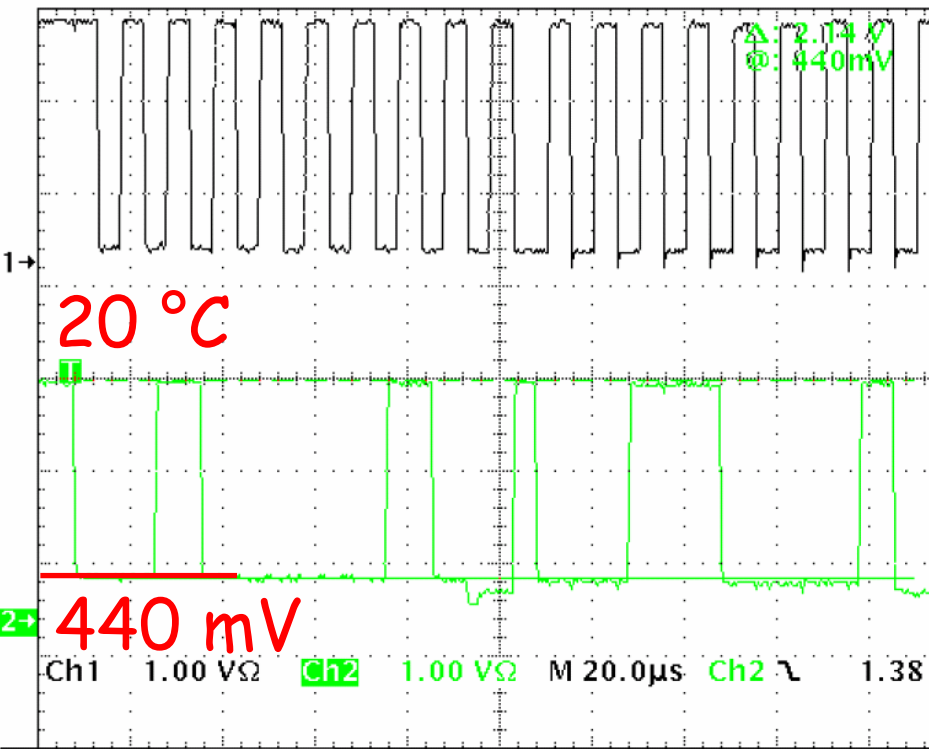
- Warm temperature, same APV at different I²C LOW levels
- Even if below 500 mV, there are problems in communication





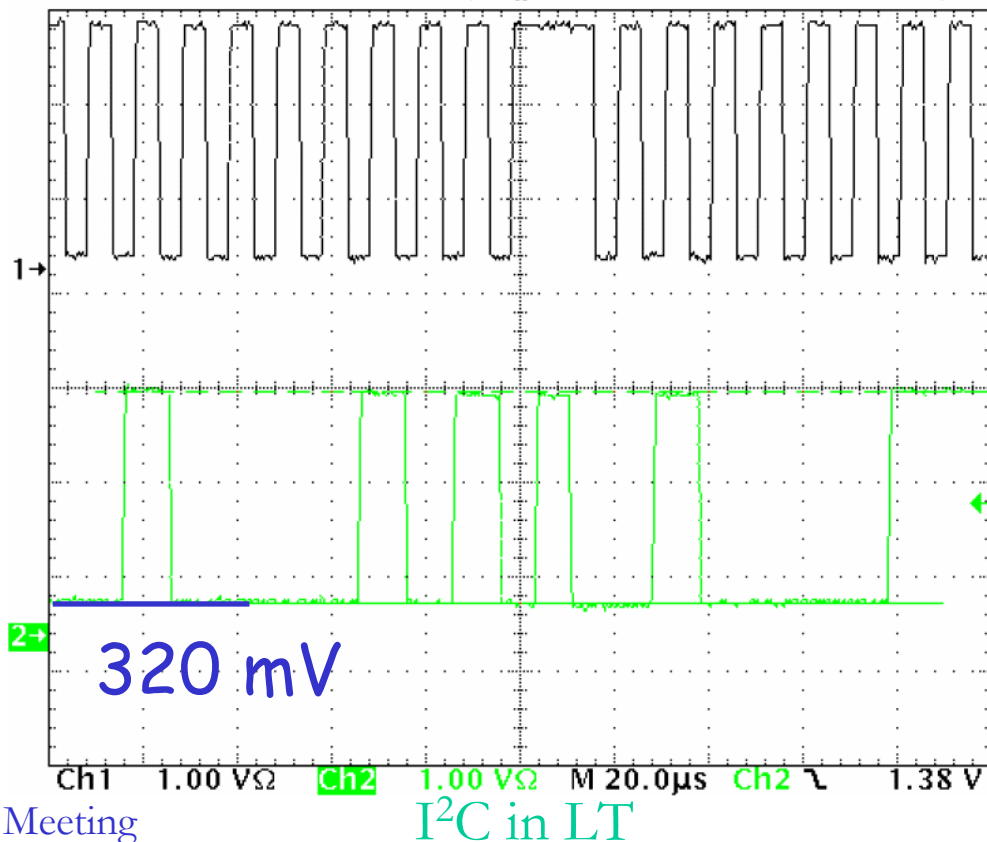
Warm and Cold

- Varying the working temperature, the APV doesn't recognize the same LOW level



Problems in Cold

- One chip at low temperature doesn't work even pulling the LOW level down to 320 mV
- With 2200 Ω pull-up resistors even this chip is completely recovered



Conclusions

- Not all APVs behave in the same way: different sensitivity to LOW thresholds leads to loss of communication → **WE WERE RIGHT!**
- To avoid this problem we chose to increase the pull-up resistors on the data line of the CCU6 from 1500 Ω to 2200 Ω . **All R1 through R12 on the CCU6 boards have to be replaced**
- **With this modification no I²C errors appear, both on new tested modules and on previously failing modules which are then recovered**
- The line speed of the I²C bus in our case is 100 kHz. Is it safe going to higher speed?

