Universiteit Antwerpen

Lab report week 5

Michiel Aernouts michiel.aernouts@student.uantwerpen.be
Mats De Meyer mats.de.meyer@student.uantwerpen.be
Dennis Joosens dennis.joosens@student.uantwerpen.be

April 26, 2017

1 Progress

1.1 I²C

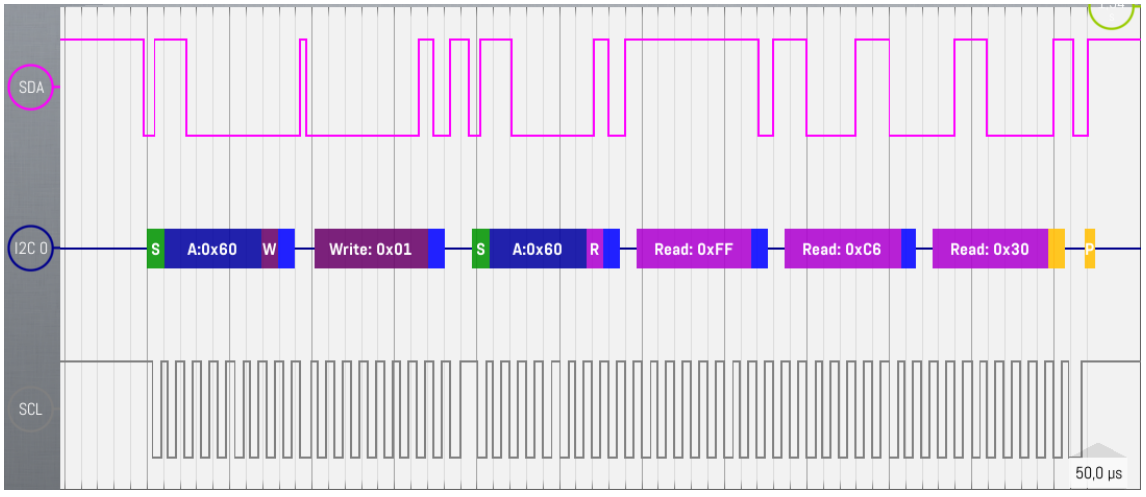

Figure 1: I²C smartscope

Code can be found at `https://github.com/michielaernouts/teammmd/tree/master/I2Cproject`. We had a lot of problems with this topic, as we did not seem to manage to read correct barometer values from the MEMS sensor.
Eventually, we found that we were initiating registers for the altitude sensor instead of the barometer. After fixing this, we obtained correct barometer values over I²C.

1.2 PIR

The PIR sensor needs to be debounced, we do this by adding a low pass filter to the sensor output.

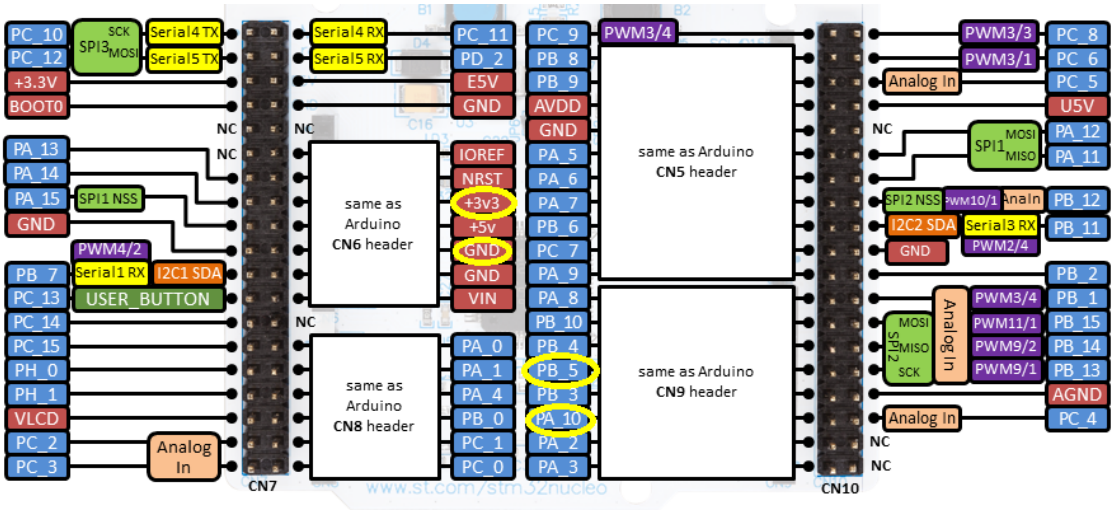

Figure 2: Used pins for PIR sensor

GPIO pin A10 is used as the sensor input pin.

Through some measurements we executed, we need to overlap a time frame of about 200 ms to make the sensor output stable.
Following the data sheet of the PIR sensor, the output of the sensor can have an average current of 170 μA and can withstand a current up to max 300 μA.

We know:

V = 3.3 V
I = 170 μA

So:

$R = \frac{3.3V}{170\mu A} = 19.4\ k\Omega$

$R = \frac{3.3V}{300\mu A} = 11\ k\Omega$

So the resistor should be at least 11 kΩ.

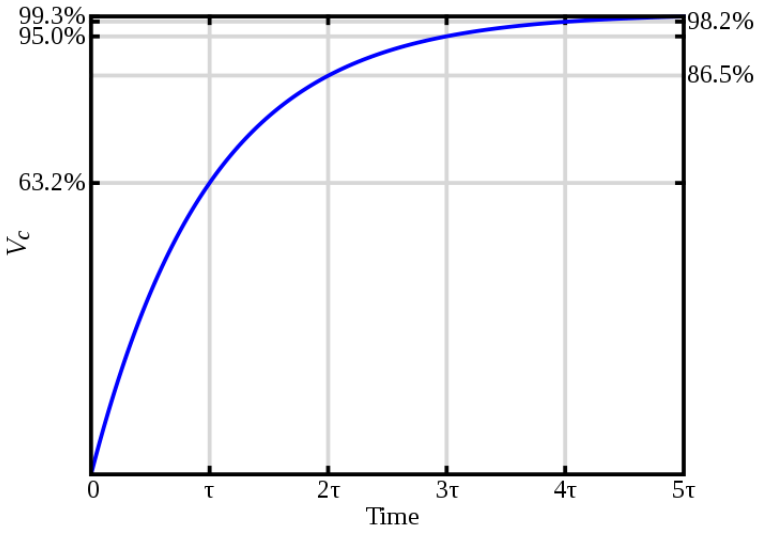

Figure 3: Charging cycle of a capacitor

About the last 3τ is essential to keep the signal "high".
3τ = +/- 500 ms
$\rightarrow \tau$ = 166.67 ms

τ = R * C
$\rightarrow$ C = $\frac{166.67ms}{22k\Omega}$ = 7.6 μF

We select a bigger resistor to lower the current. For a **22** $k\Omega$ we have 150 μA.and choose a **6.8** μ**F** capacitor.

1.2.1 Schematic PIR

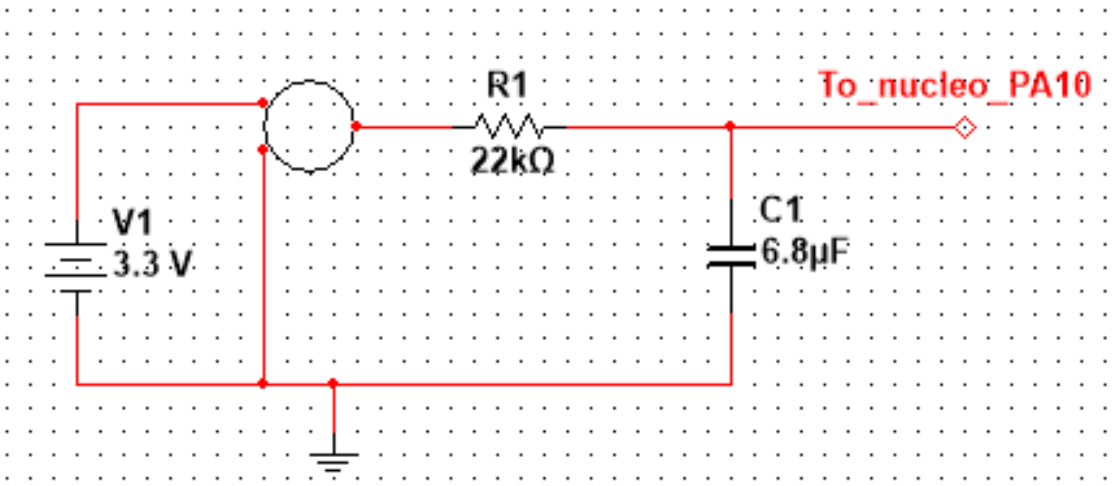

Figure 4: Schematic PIR connection

To make it work we used a 5 kΩ resistor and LED to show if there is motion detected, this depends on the LED of course. The LED is connected to **GPIO pin B5** and ground.

Code:
`https://github.com/michielaernouts/teammmd/tree/master/pir`

1.3 mag3110 pins

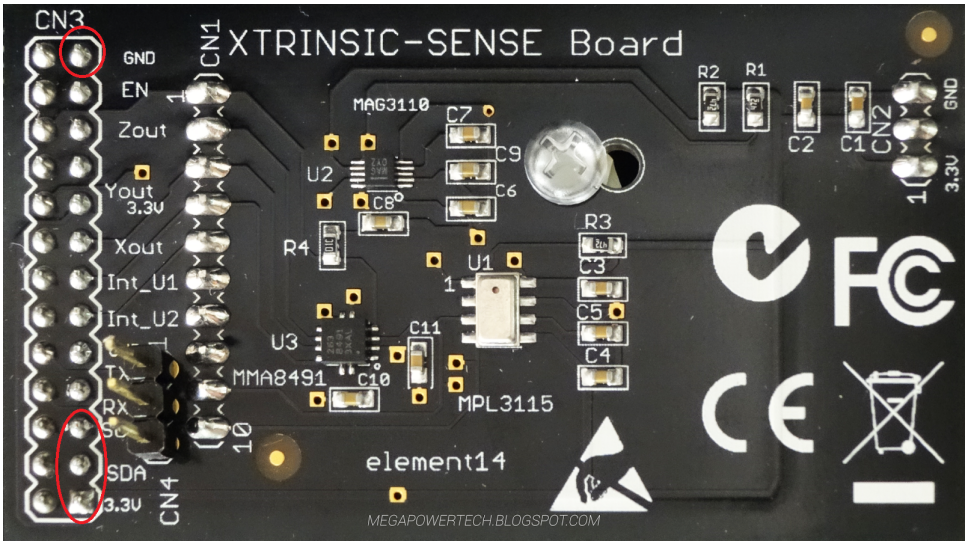

Figure 5: Used pins for MAG3110 sensor

Code:
`https://github.com/michielaernouts/teammmd/tree/master/magneto`

1.4 mma8491q pins

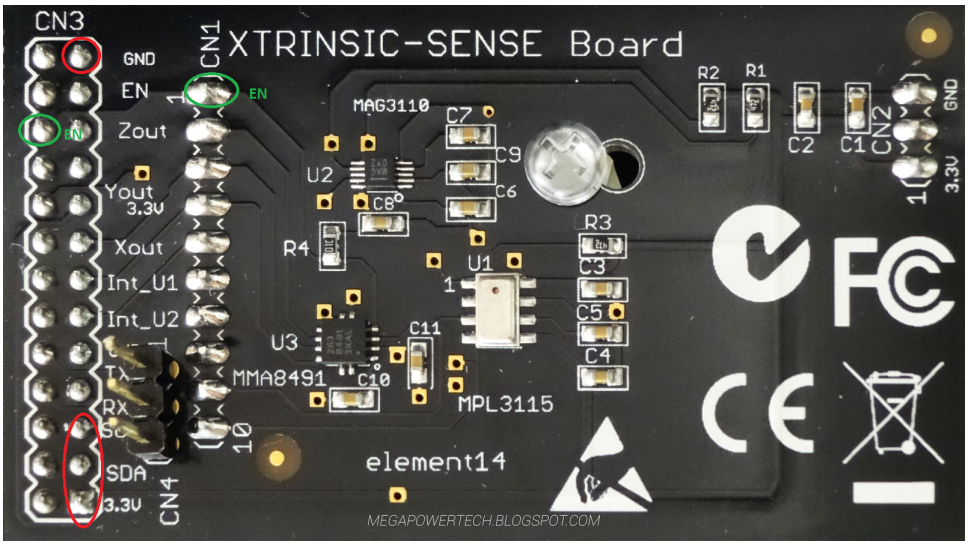

Figure 6: Used pins for PIR sensor

The enable pin is essential to make the mma8491q work. However it is shown wrong on the board.

Code:
`https://github.com/michielaernouts/teammmd/tree/master/accelerometer`

1.5 ADXL345

As an extra we took another accelerometer and got it working.

Code:
`https://github.com/michielaernouts/teammmd/tree/master/ADXL345`

1.6 DASH7

Setup is one sensor node which is sending sensor data at a set interval and multiple gateways receiving this data, filtering it and publishing to MQTT topic to display in openhab.

Data is sent using ALP command wrapped in a serial frame. Last byte of serial frame before the ALP command is the length of the ALP frame, because an ALP command is variable in length. When the hexadecimal serial frame is created, it can be sent over UART. This is tested over USB, with serial terminal realterm. The following frame is sent as an example from a DASH7 node:
`0x41 0x54 0x24 0x44 0xc0 0x00 0x0e 0xb4 0xaf 0x32 0xd7 0x01 0x00 0x10 0x01 0x20 0x01 0x00 0x02 0x00 0x01`
This serial frame is a wrapper for the following ALP frame (with length 0x0e=14):
`0xb4 0xaf 0x32 0xd7 0x01 0x00 0x10 0x01 0x20 0x01 0x00 0x02 0x00 0x01`
The gateway is connected to a raspberry pi and receives the command as seen in Figure 7

```
pi@raspberrypi:~/pyd7a $ PYTHONPATH=. python examples/modem_example.py -d /dev/ttyUSB0
connected to /dev/ttyUSB0, node UID b57000009125d running D7AP v1.1, application "gatewa" with git sha1 46f48
1f
Command with tag 228 (executing)
        actions:
                action: ReturnFileData: file-id=1, size=1, offset=0, length=2, data=[0, 1]
        interface status: interface-id=215, status=unicast=False, nls=False, retry=False, missed=False, fifo_
token=124, rx_level=75, seq_nr=0, target_rx_level=80, addressee=ac=1, id_type=IdType.UID, id=0xb57000009126dL
, response_to=exp=3 mant16, link_budget=85, channel_header=coding=ChannelCoding.PN9, class=ChannelClass.NORMA
L RATE, band=ChannelBand.BAND 868
```

Figure 7: D7 command received at the gateway.

To send data with the STM microcontroller, the D7 node is connected with Vcc, ground, RX and TX. UART1 is used, and TX is connected with TX as well as RX with RX. To send the data consisting of hexadecimal bytes, an array of uit8_t is used.

```
void Dash7Send(void)
{
  uint8_t message[] = {0x41, 0x54, 0x24, 0x44, 0xc0, 0x00, 0x0e, 0xb4,
  HAL_UART_Transmit(&huart1, (uint8_t*)message, sizeof(message), HAL_MA
}
```

2 Planning

- Calibrating and testing MEMS sensor
- Add MEMS sensor data to openHAB (ok? uitbreiding?)
- Get OpenCV working with camera (busy)
- Backup SD card on blanco SD

- Optimizing PIR sensor
- check 2's complement conversion (sensors)