Hochschule Esslingen
University of Applied Sciences

Team Project

Quadrocopter

in the degree course ASM-SB
of the Faculty Graduate School
ASM2

Oliver Breuning
Martin Brodbeck
Jürgen Schmidt
Phillip Woditsch

Periode: Summersemester 2015
Professor: Prof. Dr. Jörg Friedrich

Contents

List of Figures

List of Tables

1 Infrared Sensor

We tried to establish a distance measurement with GP2Y0A60SZLF Infrared Sensor on a Pololu Carrier Board. The Measuring distance of that sensor is 10 to 150 cm. For detailed technical values, please see the sensor chapter or the datasheet.

1.1 First steps

Our first action was to check, weather the sensor is working or not. So we need to active the I²C Bus on the Raspberry Pi and set up the Analog Distance Converter (ADS1011), because the Infrared Sensor provides us only a analog output.
We did a fast test of the sensor using python, just for checking the sensor is not broken.

To gather a bunch of data, we developed I²C and ADC Driver ourselfs in C.

1.1.1 ADC Configuration

First Hex depends on Starting Conversion + the Input, which Pin to read A0-3,
Second Value is PGA (001)=+-4,099V and continuous Mode (0).

These three bytes are written to the ADS1015 to set the config register and start the conversion.

l_writeBuf_rg24[0] = 1;
This sets the pointer register to write two bytes to the config register

l_writeBuf_rg24[1] = l_mux_ui8;
This sets the 8 MSBs of the config register (bits 15-8) to 11000011

l_writeBuf_rg24[2] = 0x23;
This sets the 8 LSBs of the config register (bits 7-0) to 00100011

The following table shows the Hex values in the direction from top to bottom what is needed for reading a conversion value on the specific inputs. At the empty field, there is no change compared to Input A0.

	Input A0	Input A1	Input A2	Input A3
Slave adress+RW	0x49			
PTR register	0x01			
MSB Config	0xC2	0xD2	0xE2	0xF2
LSB Config	0x23			
Slave adress+RW	0x49			
PTR register	0x00			
Data from Slave	0xXX	0xXX	0xXX	0xXX
Data from Slave	0xXX	0xXX	0xXX	0xXX

Table 1.1: ADC Conversion Read

MSB:
The first hexadecimal value is to start the conversion and depends on the Input, which Pin to read A0-3.
The second hexadecimal value is PGA (001)= +-4,099V and continuous Mode (0).

LSB:
The first hexadecimal value is the sample Rate. (001) sets it to 250SPS + Comp Mode (0).
The second hexadecimal value is the Comp. config. (0011) disables the comparator.

1.2 Measured Values

With our own logic getting the data from the sensor, we could store as much data we want. Now we were able to get a impression on the jitter and could thing about a good algorithm to get reliable values.

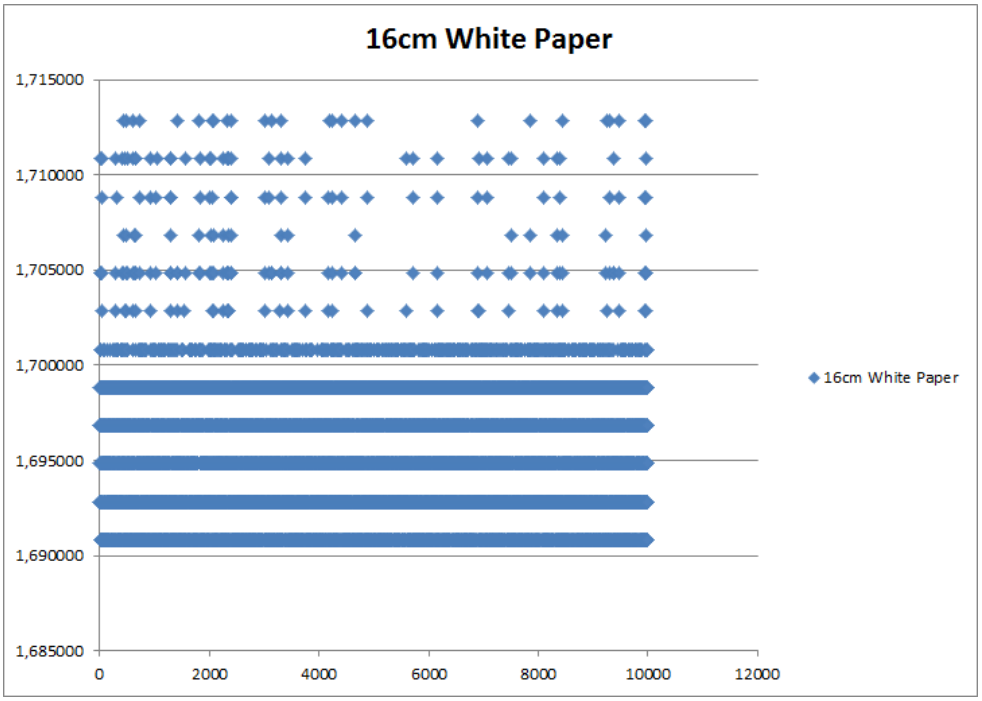

Figure 1.1: IR: 16 cm to white paper

Gathering 10 000 values while measuring the distance of 16cm on a white paper. We used the white paper to compare the values with them in the datasheet.

As we changed the distance and the surface, we discover a astonishing fact.

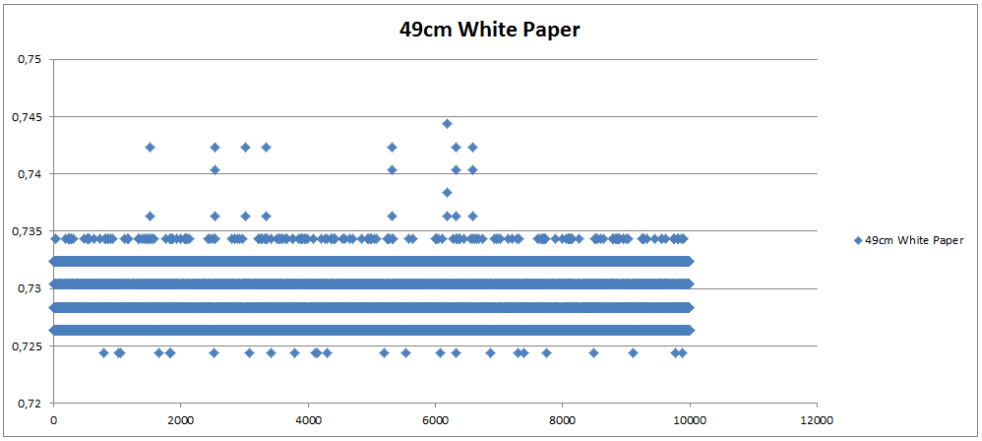

Figure 1.2: IR: 49 cm to white paper

Same measurement on the surface of the table:

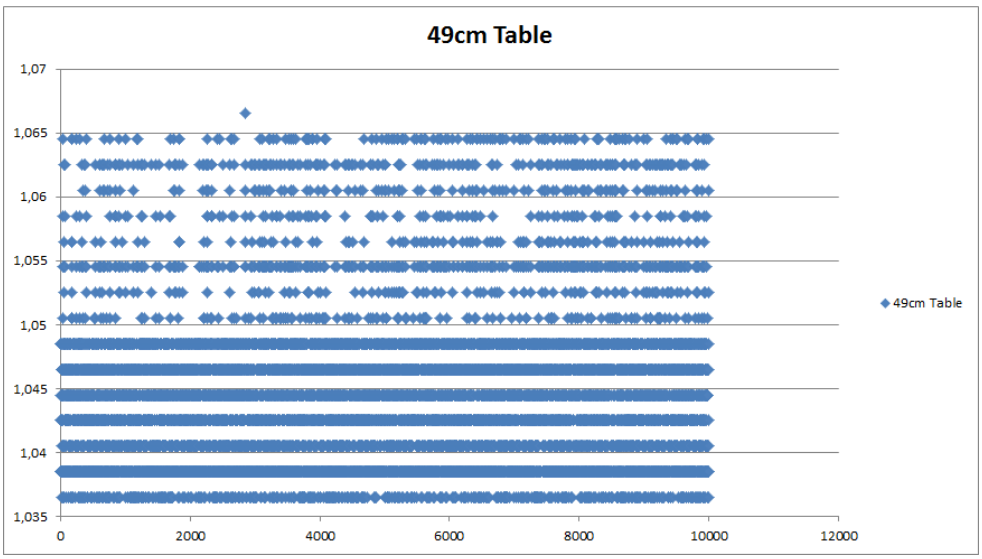

Figure 1.3: IR: 49 cm to table

As you see, the measurements on different surfaces are highly varying. At 49cm we got the following rough values:

White Paper: 0,73 V
Desk Surface: 1,42 V

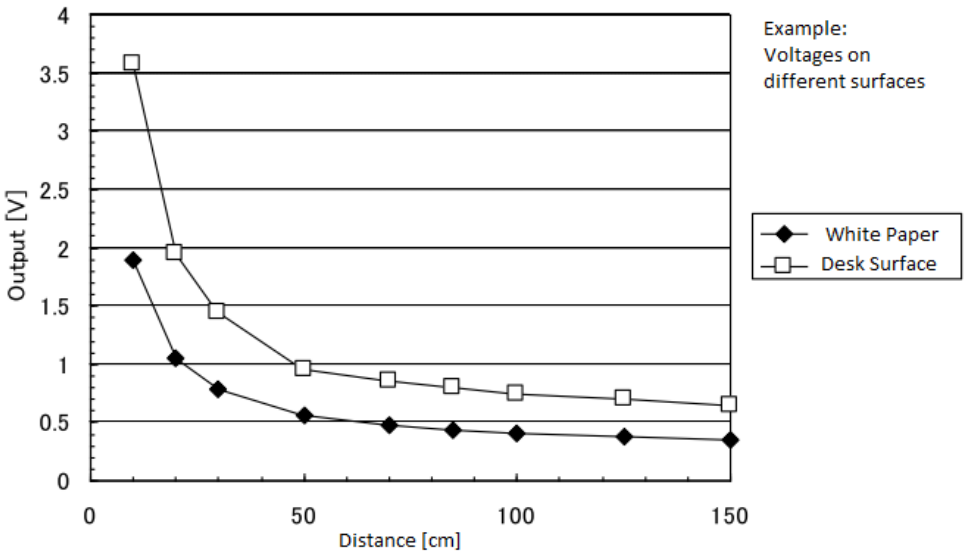

Figure 1.4: IR: Sensor values (voltages) on different surfaces

2 Surfaces and 2 Distances:

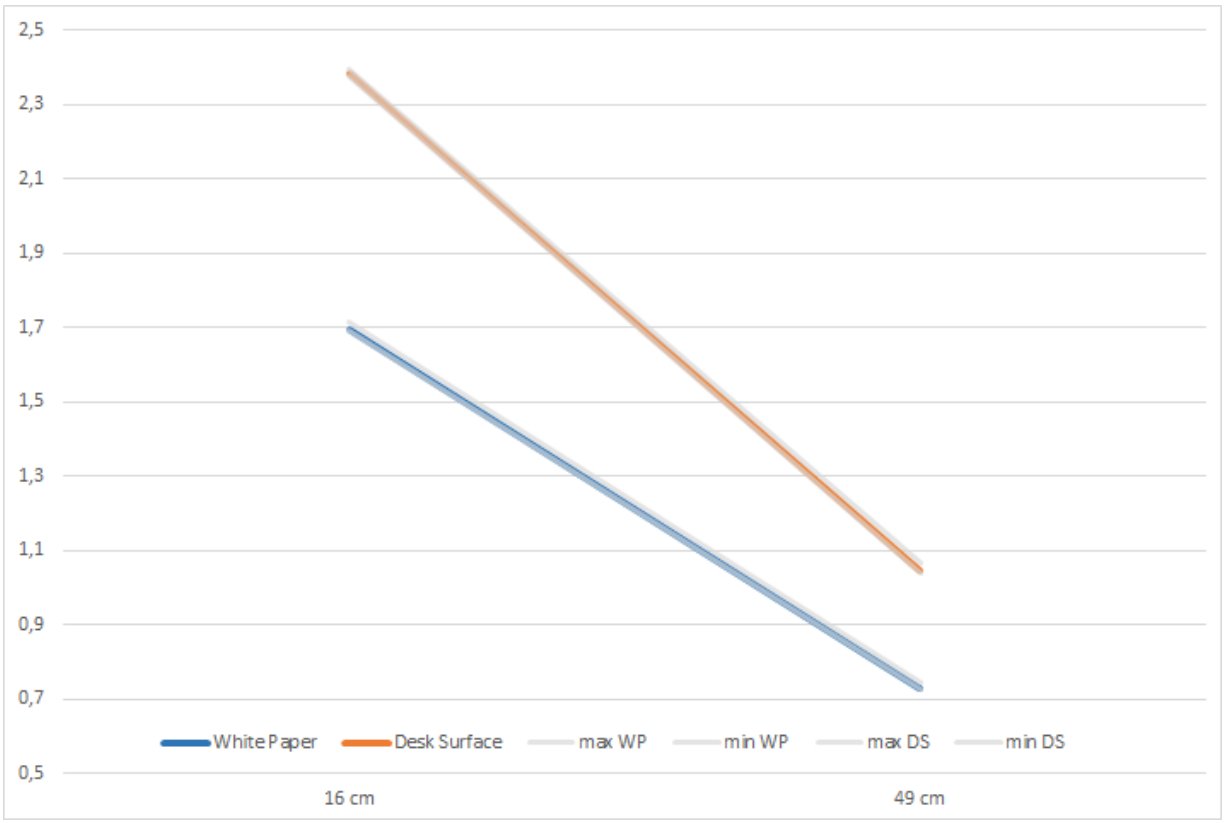

Figure 1.5: IR: Compared Voltages on two surfaces

We only measured 2 different distances in order to check the functionality of the sensor. The grey lines shows the jitter which we measured.

1.3 Conclusion

As we could not guarantee a good flying with changing surfaces, we decided to stop chasing a solution with the IR Sensor.

2 Ultrasonic Distance Sensor

Our next approach was, measuring the distance with a ultrasonic distance sensor. In a meeting we agreed on using 3 ultrasonic sensor to measure the distance to ground, because a single sensor is imprecise. We hoped that a sensor fusion of 3 ultrasonics will bring us better values.

While searching for theses sensors, we found a relatively cheap Laser Sensor. After a short consultation, we decided to go with that one. So Ultrasonic was discarded, before start.

3 LIDAR Laser Sensor

LIDAR-Lite Laser Distance Sensor
Model LL-905-PIN-01

Performance
Range: 0-20m LED Emitter
Range: 0-60m Laser Emitter

Interfaces
- I^2C
- PWM

For detailed technical values, please see the sensor chapter or the datasheet.

3.1 Connecting the sensor with I^2C

Because there is no proper input for a PWM signal, we used the I^2C Bus to connect the laser sensor. As the sensor does not support fast I^2C mode, which we need, we decided to use the second I^2C on the Raspberry Pi. This bus is orginally located on the camera port with an FPC (flexible printed circuit) Connector. We managed it to redirect it to the normal Pins on the board. See Chapter xxxxxx.

3.2 First steps

Quick Start Guide

1. Make Power and I2C Data Connections as per J1 connector pin out diagram. Pins 2 & 3 are optional connections and not required.
2. Initialization: Apply Power to the Module. The sensor operates at 4.75-5.5V DC Nominal, Maximum 6V DC.
3. Measurement: Write register 0x00 with value 0x04 (This performs a DC stabilization

cycle, Signal Acquisition, Data processing). Refer to the section "‘I2C Protocol Summary"’ in this manual for more information about I2C Communications.
4. Periodically poll the unit and wait until an ACK is received. The unit responds to read or write requests with a NACK when the sensor is busy processing a command or performing a measurement. (Optionally, wait approx. 20 milliseconds after acquisition and then proceed to read of high and low bytes)
5. Read: register 0x0f, returns the upper 8 bits of distance in cm, register 0x10, returns the lower 8 bits of distance in cm. (Optionally a 2-Byte read starting at 0x8f can be done)

3.3 Measured Values

3.4 conclusion

Bibliography

[1] Thomas Nonnenmacher, LaTeX Grundlagen - Setzen einer wissenschaftlichen Arbeit Skript, 2008,http://www.stz-softwaretechnik.de; *(Bei STZ Internetseite unter Publikationen - Skripte) [V. 2.0 26.02.08]*

[Gun04] Karsten Günther, LaTeX2 — Das umfassende Handbuch, Galileo Computing, 2004, http://www.galileocomputing.de/katalog/buecher/titel/gp/titelID-768; *1. Auflage*