

Escuela de Ingeniería Informática

Karl Deilmann
Dani Quintero

Internet de las Cosas

Report 2

21 of Sep - 29 of Dec

Contents

0.1 I ² C and Serial communication	1
A	2
References	2

0.1 I²C and Serial communication

The goal for this Assignment was to establish a serial connection between two *Arduino MKR 1310* μ -controllers and send commands from one to another using a custom communication protocol. The commands should then make the second *Arduino* take a distance measurement with a ultrasonic sensor and return the data to the first one. Furthermore we have an OLED screen to display data from the sensor reading. The protocol should have the following options:

1	- help	-> Show available commands
2	- us <srf02> {one-shot on <period_ms> off}	-> Start a single measurement, a continuous measurement
3		for <period_ms> duration, stop periodic measurements.
4		<srf02> refers to the sensors address
5	- us <srf02> unit {inc cm ms}	-> Set the unit for the measurements < inc cm ms >
6	- us <srf02> delay <ms>	-> Set delay between consecutive measurements
7	- us <srf02> status	-> Return configuration of sensor (address, state, units, delay)
8	- us	-> Return all available sensors (addresses)

Slave

To communicate with the sensor and the display, we use the **wire** library which in turn uses I²C to communicate via a bus wire with all connected devices (here: sensor and display). Both devices are given a unique address to not conflict while using the communication bus. We refresh the displays content every 100ms to make it responsive to ongoing measurements or incoming commands. The display show the last measurement results (**distance** and **minDistance**) as well as the unit used for measuring and whether there is an ongoing measuring and for how long it will keep measuring. The itself sensor is only activated on receiving the command from the other *Arduino*. To implement the the periodic reading of the sensor, we use an expression to measure the time elapsed since the last reading and if the specified delay has been reached, we perform the next reading. The data from each reading is automatically send to the master after each read.

Master

The master receives commands from the Serial interface command line of the *Arduino IDE*. It parses the string to recognise the command issued and sends a header for the corresponding command followed by a payload with further information such as **address**, or **period_ms**. It then waits for a response from the slave.

Appendix A

Git repository: https://github.com/Puvogel/Internt_de_las_Cosas