UMONS
Université de Mons

Faculté Polytechnique

Hardware/Software - Humidity sensor HH10D

Project

DESCAMPS Flavian, RIZZO Lisa

POLYTECH MONS

Supervisor : PROF. VALDERAMA Carlos

Academic year 2020-2021

ACADÉMIE UNIVERSITAIRE WALLONIE-BRUXELLES

Contents

Introduction

Chapter 1

Simius Black

Sources pour compteur + Test bench : https://www.fpga4student.com/2017/06/vhdl-code-for-counters-with-testbench.html image présentation : https://sti2d.ecolelamache.org/iii$_c ommunications_i nf o$

Chapter 2

I^2C

Before we start, we have to refresh your memory about the I^2C bus. Philips in order to provide a simple and inexpensive solution to communication between digital integrated circuits inside consumer devices like televisions. The main advantage of the I2C bus is to limit the number of links between integrated circuits[1]. The bus is caracterised by only 2 signals :

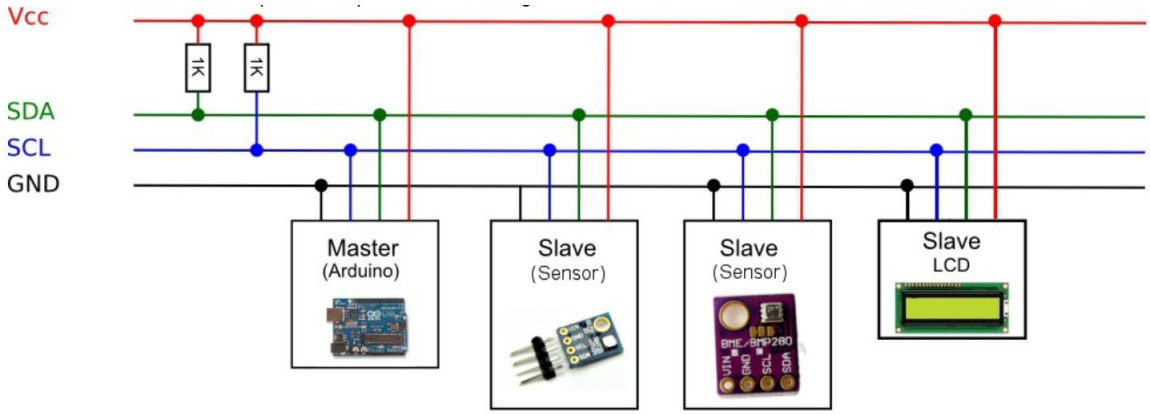

Figure 2.1: I^2C bus conception

- SCL signal that contains the clock of the bus
- SDA signal that contains the date of the bus

The both lines SDA and SCL are pulled to Vcc voltage level (high level) through pull-up resistors. The Master is the emetter and the Slave is the collector. The maximum number of devices is limited by the number of addresses available, 7 bits for the address and one bit to define whether to write or read, i.e. 128 devices, but it also depends on the physical characteristics (capacity) of the bus. It should be noted that some addresses are reserved by the manufacturers, which greatly limits the number of devices There are also some protocols like starting and stopping conditions. As the transmission is in serial form, start and end information must be provided. The start information is called START and the end information STOP [1]

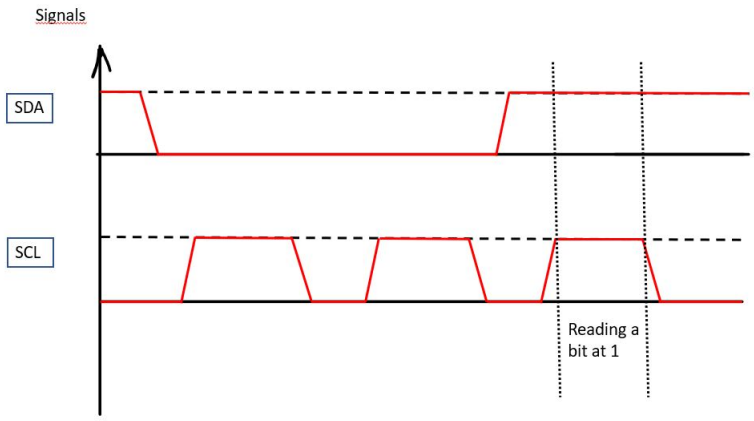

Figure 2.2: SDA and CLAC signals

A master takes control of the bus by performing a START: it sets SDA to 0, while SCL remains at 1. During communication, the SCL clock is sent by the master and SDA can only change its state when SCL is at 0 2.3.

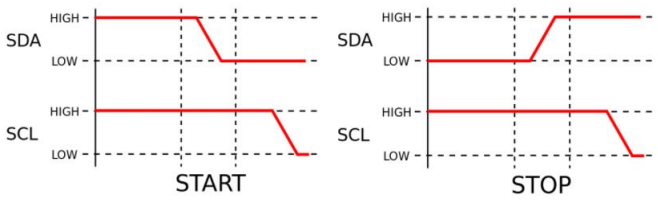

Figure 2.3: Starting and stoping protocols

A important thing to consider is also the acquittment of a task. When a tast is received the Master or Slave confirms at emitter that he has the data by sending an ACK at the end of a frame and sending a Nack if not. Ack and Nack are shown at figure 2.4.

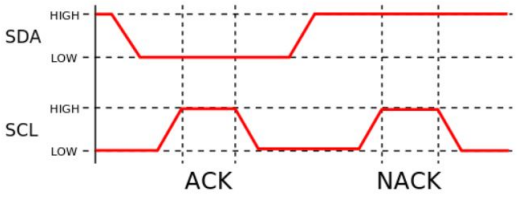

Figure 2.4: Acquittment - Ack and Nack

0.1 I^2C Master state

Before showing the code, you must know how a Master state machine works in I^2C. We begin in the ready state and wait until the ena signal. Then the fist slave slv-ack1 check the acknowledgement to prepare commands of RW (reading) and WR (writing) to emits data or

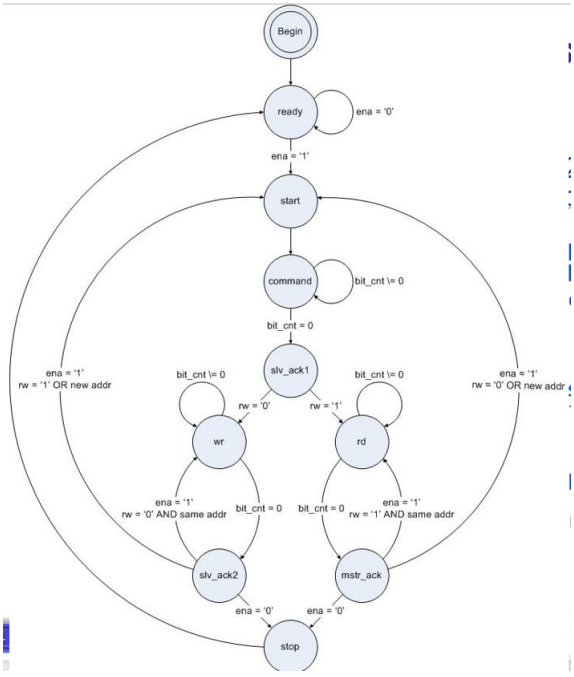

Figure 2.5: Master State Machine in I^2C

receives data in/from slave. To finish, the master check the slave's response (so the slv-ack2 state) to see if the data will come or not.

Chapter 3

Code

The purpose of the code is to determine the humidity level in the room of the humidity sensor HH10D. The main way to arrive at that result is to implement a VHDL code that will determine frequency of the sensor during a period of time and will translate it into a humidity level given by the maker. This is given by the following table 3.1.

HH10D Humidity Module Characteristic				
Parameter	min	nominal	max	unit
humidity range	1		99	%
accuraccy	-3		+3	%
temperature range	-10		+60	C
working voltage	2.7	3	3.3	V
stability versus time		1%		per year
power consumption	120	150	180	uA
Output Frequency Range	5.0	6.5	10	KHZ

Figure 3.1: HH10D characteristics

We can see that we have to find a frequency between 5 and 10 kHz to translates it into a humidity level also given in the table 3.1 [2]. In practice, we can show you the physical implementation of the Humidity sensor in 3.2 and its application in 3.3 [2].

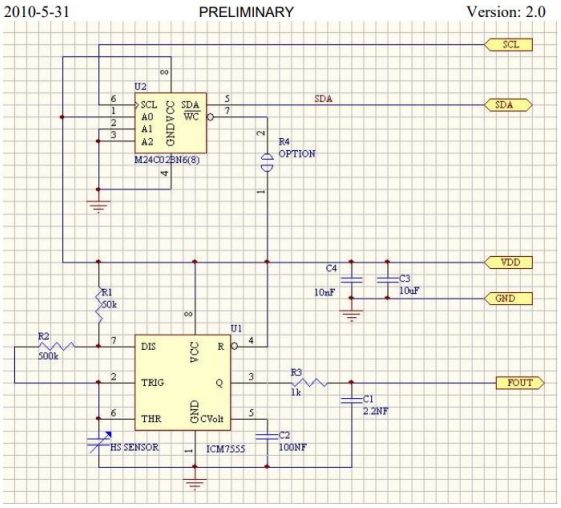

Figure 3.2: Circuit diagram

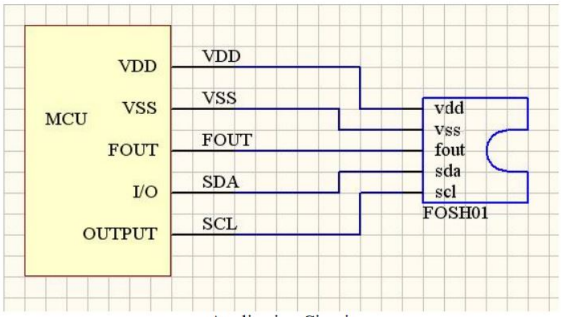

Figure 3.3: Circuit application

1 VHDL

1.1 Quartus

1.2 I^2C Master

1.3 I^2C Master Test Bench

1.4 I^2C Counter

1.5 I^2C Counter Test Bench

2 ModelSim

In this section, we will visualise all the previous VHDL code and will see if all the connections work.

Conclusion

Appendix

Step1 : code for each functions

ua function

```
function ua = fcn(Iref, ia, theta, ua_bis)
Vd = 307;
delta = 4;
stroke = 15;
theta_on = -20;
theta_off = theta_on + stroke;

if (theta_on <= theta) && (theta < theta_off)
    if ia <= Iref-delta/2
        ua = Vd;
    elseif ia >= Iref+delta/2
        ua = -Vd;
    else
        ua = ua_bis;
    end
else
    if ia >= 0
        ua = -Vd;
    else
        ua = 0;
    end
end
```

Listing 3.1: ua function

Bibliography

[1] *Le bus I2C*. URL: https://sti2d.ecolelamache.org/iii_communications_informatiques_3_exemple_n2_le_bus_i2c.html (visited on 05/16/2021).

[2] LTD HOPE MICROELECTRONICS CO. "HUMIDITY SENSOR MODULE". In: (2010).