PROJECT 2 (MK 1)

Kunal Mukherjee

UNIVERSITY OF EVANSVILLE

STATEMENT OF THE PROJECT

The project was assigned to give us experience with I2C transmission protocol. I2C is one of the widely used protocol for sensors and inter-device communication. Therefore, having a strong grasp of the I2C protocol was necessary, and this project where we had to make I2C1 configured as master had to communicate with I2C3 which was configured as a slave.

POSSIBLE UTILITY OF THE PROJECT

The possible utility of the project is in sensor communication and game development. For EE 380, on the sensors is using I2C protocol to communicate. By having completed the code for this project, the I2C master and slave code can be easily modified to be used with the sensor for EE 380.

IDENTIFICATION OF SPECIFICATION

The specification was that the user should be able to give any slave address. Then by using the slave address, the user should be able to request the slave to blink a led some number of times that is specified by the user. Also, the master should be able to request a value from the slave, after which it will blink an led corresponding the value sent by the slave.

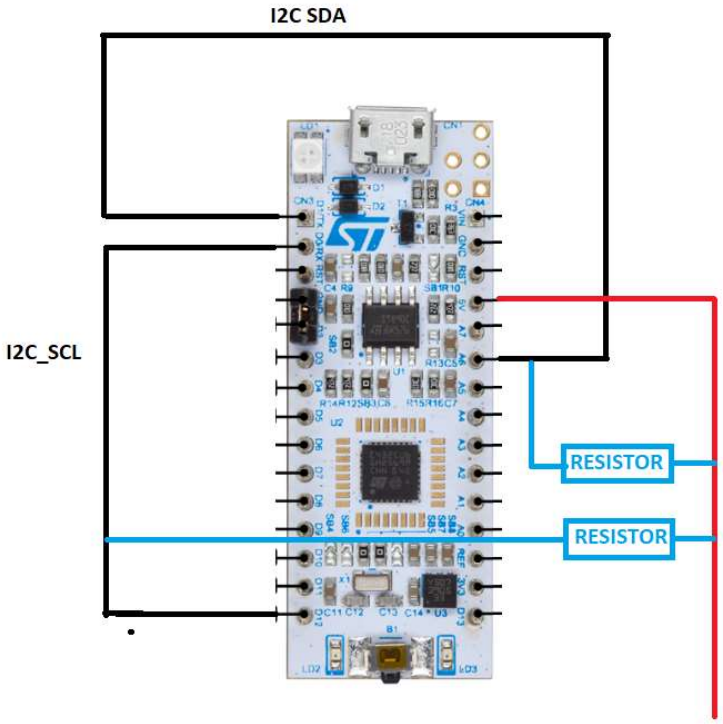
IDENTIFICATION OF DESIGN ISSUES AND SOLUTION

One of the biggest design issues that I faced was, using the correct interrupt. The slave had to look at his own interrupt to determine if it can receive data or can send data, but, the interrupt routine has to look at the TC (transfer complete flag) to see, if the transfer has been completed. The second issue was to determine which interrupt to enable, because if we enabled

all the interrupts, then we won't be able to leave the interrupt routine. We will be continuously stuck in the routine.

Another, place where I struggled was to notice that the slave I2C never gets control of the bus, so the idle, wait, start, stop function is useless. As, the slave cannot start or stop a transmission but rather send a acknowledge or a no acknowledge flag. In the beginning, I did not understand this concept, and I tried giving the slave control, and I my code would not work. But, after I understood the problem, I was able to work around it.

SCHEMATIC OF COMPONENTS EXTERNAL TO THE STM32F446 BOARD



INFORMATIONAL RESOURCES USED

- https://www.youtube.com/watch?v=Wdac8FnZvQA [58. How to: I2C Circuit and Initialization - ARM STM32 Microcontroller Tutorial]
- https://www.youtube.com/watch?v=XvakWfRHrCY&list=PL6PplMTH29SHgRPDufZhfMRoFwRAIrzOp&index=57 [59. How to Use I2C to Read a Device's Register Part 1 - ARM STM32 Microcontroller Tutorial]
- Embedded System Book
- https://i2c.info/ [I2C Info – I2C Bus, Interface and Protocol]