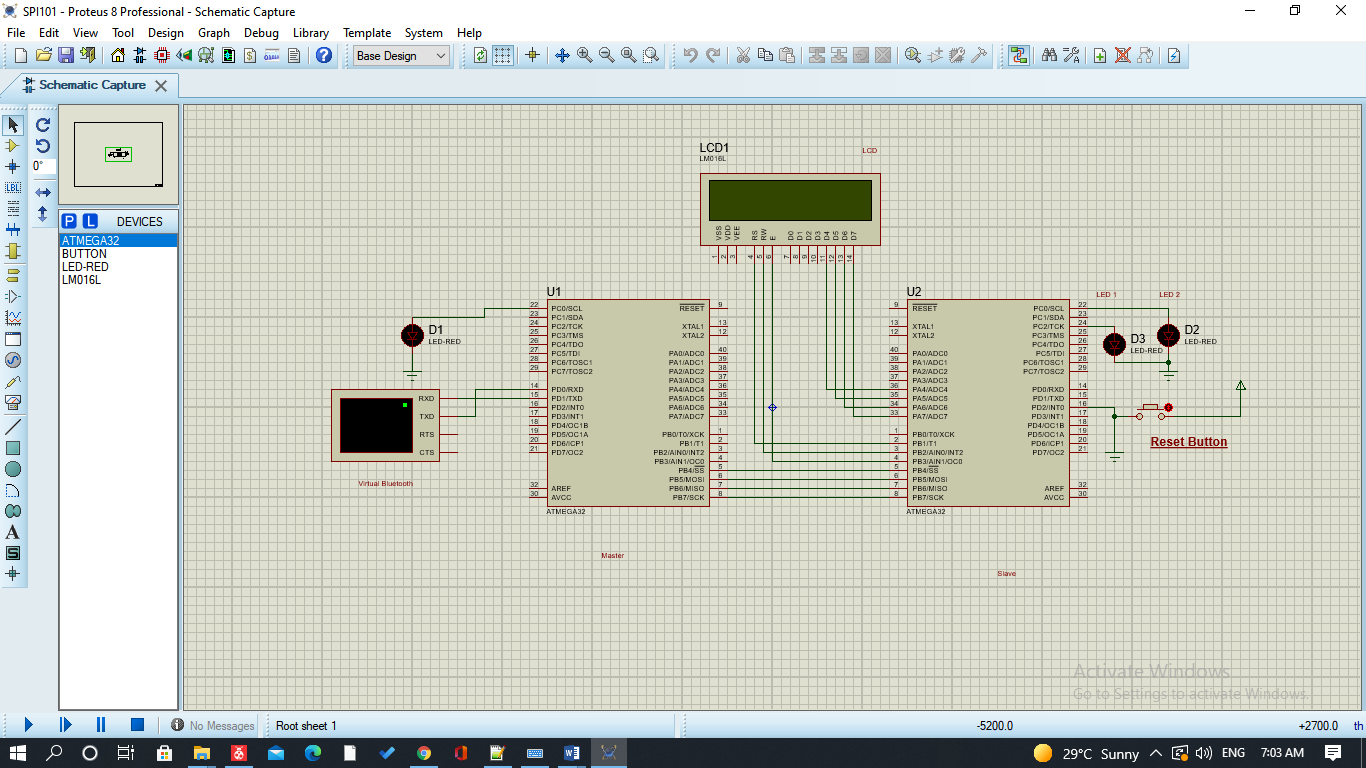
Amit learning   
Graduation Project

short line

# Submitted By: Youmna Khaled Sayed C43

# Introduction

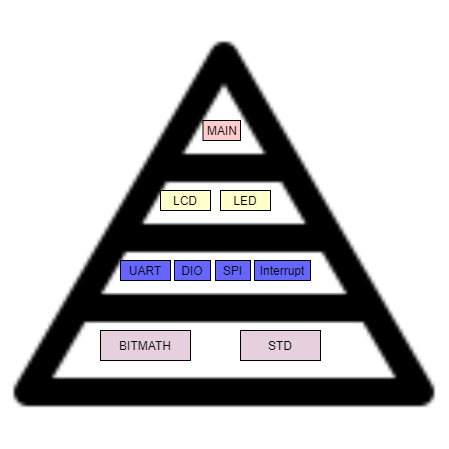
This is a screenshot of the simulation showing the circuit of the smart home that is connected using two atmega32 microcontrollers and communicate through Bluetooth module, or virtual terminal in my case in order to pass commands from master to slave or multiple slaves. In a larger scale but through the same flow of architectures, this flow is used to simulate a smart home. We could say that these two leds are room lights, the system can be more controlled using sensors and keypads and motors to create an advanced smart house.



# Modularity

I have used the computer architecture that we have been taught during the course to ensure modularity and abstraction. Code is separated in a hierarchy that ensures that different complexity of codes and kept in order and that the logic of each module is separated. This could help us in a way that keeps code clean as well as maintain the ease of the editing process if needed.

The architecture of the code is as shown below



The lower most level contains the library files that we need for the layer called Micro Controller Abstraction level, MCAL, that is the layer exactly above it .The library files contain the std types and the function that are used to process bits. This layer has direct access on the microchip and is considered software module of the peripherals needed. It also contains the protocol implementations.

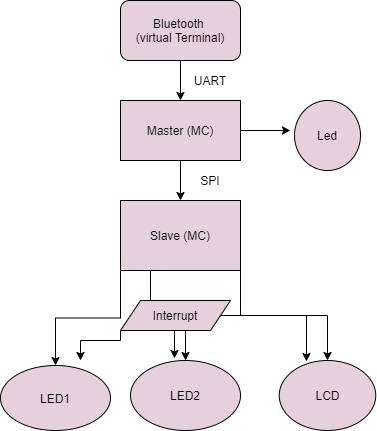
HAL, hardware abstraction layer, contains the logic needed to configure the hardware used. In my case it is the lcd and the leds.Many more hardware devices can be used in this layer.

The application layer, is the one that is simpler and easier to understand and is written in a code that is very easy to understand as all the configuration details are hidden from the one who reads the code and it contains the integration logic of the whole system. Here comes another advantage of the Modularity.

Flow of the system:

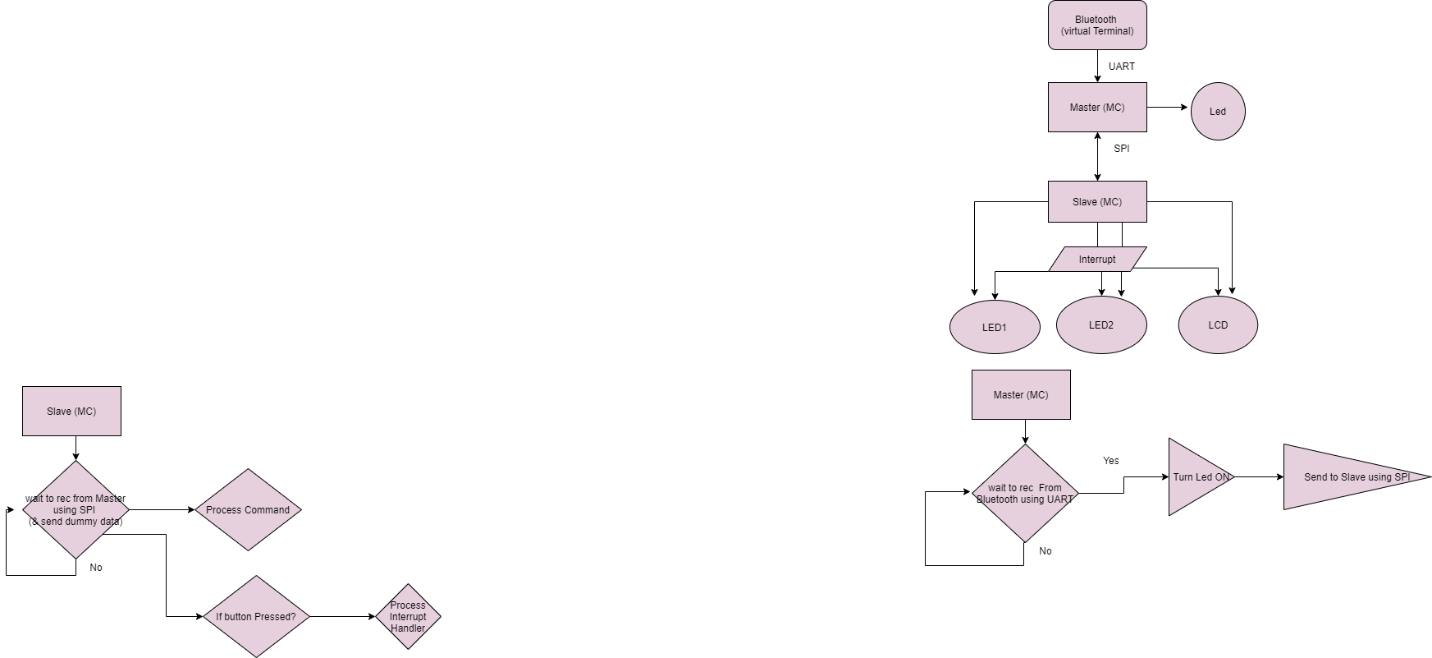
The system relies on two communication protocols. One between the Bluetooth module and the maser microcontroller that is UART. The second is SPI, which is a serial protocol to ease the communication between the Master and the Slaves. This protocol enables sending and receiving at the same time, which means that in cases where there is only one that sends and the other receives, we send dummy data.

This picture below illustrates the whole communication process:

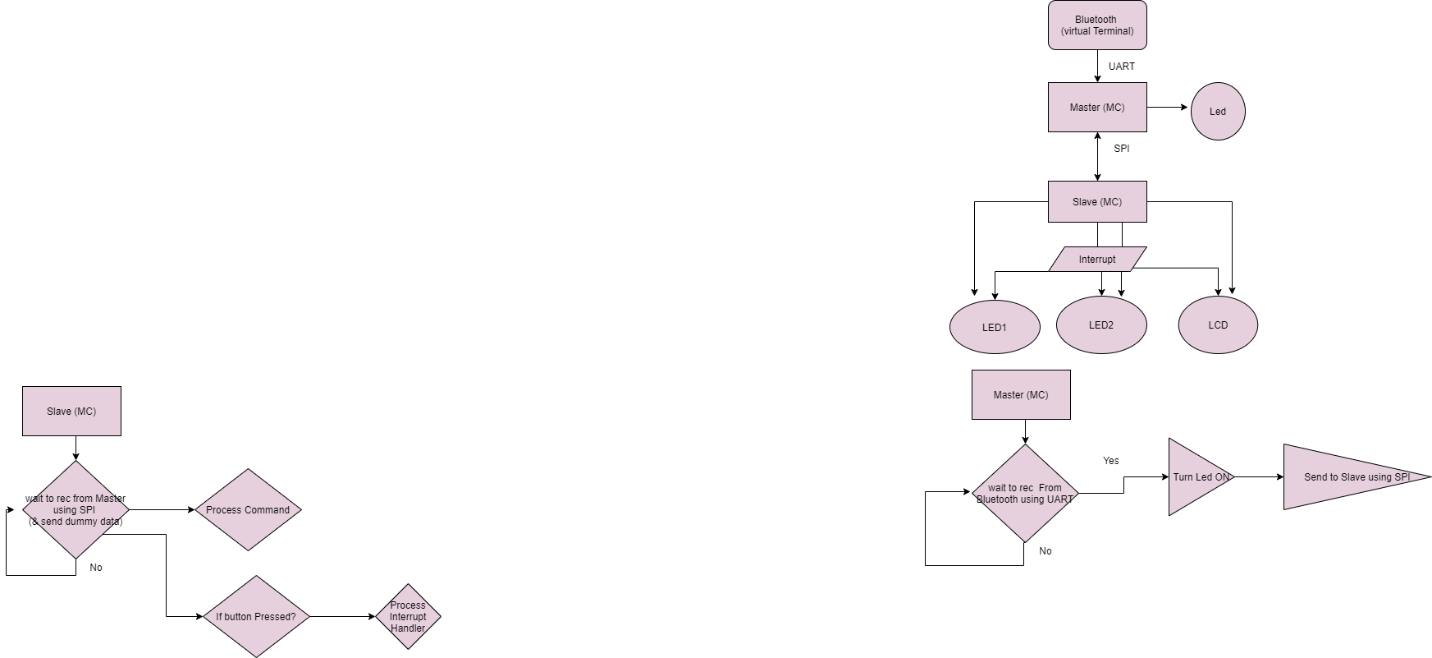


The interrupt is used where there’s a reset button that is connected such that if pressed the system resets ,this is simulated by showing a resetting message on the Lcd , in addition to turning the two leds off .

The flowchart of the Master:



The flowchart of the slave alone is also illustrated as below:

%3CmxGraphModel%3E%3Croot%3E%3CmxCell%20id%3D%220%22%2F%3E%3CmxCell%20id%3D%221%22%20parent%3D%220%22%2F%3E%3CmxCell%20id%3D%222%22%20value%3D%22Process%20Command%22%20style%3D%22rhombus%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3BfillColor%3D%23E6D0DE%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22-800%22%20y%3D%22710%22%20width%3D%22130%22%20height%3D%22100%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3C%2Froot%3E%3C%2FmxGraphModel%3E