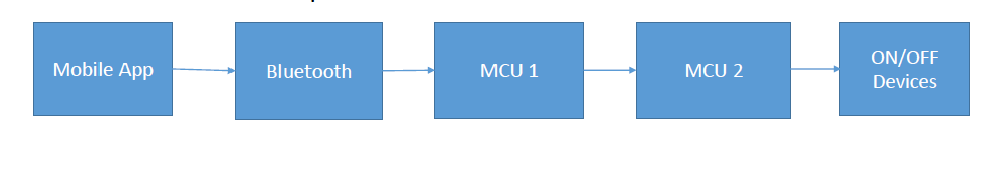
AMIT Graduation Project

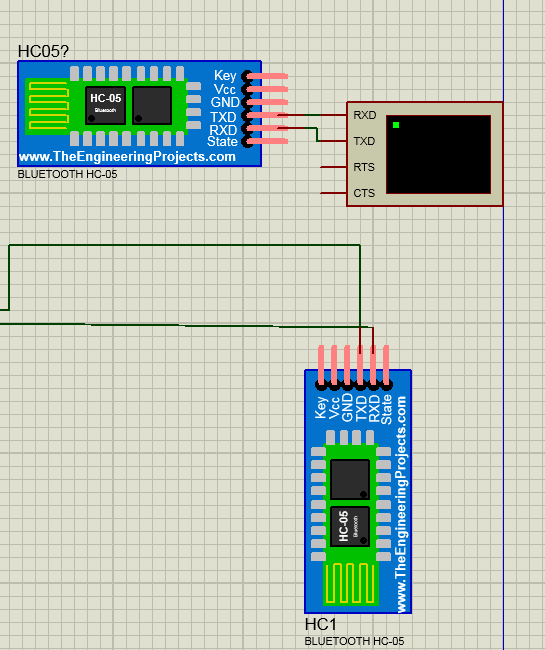
**Mohamed Amr Sayed B26 Online**

The problem at hand can be represented with the following figure:

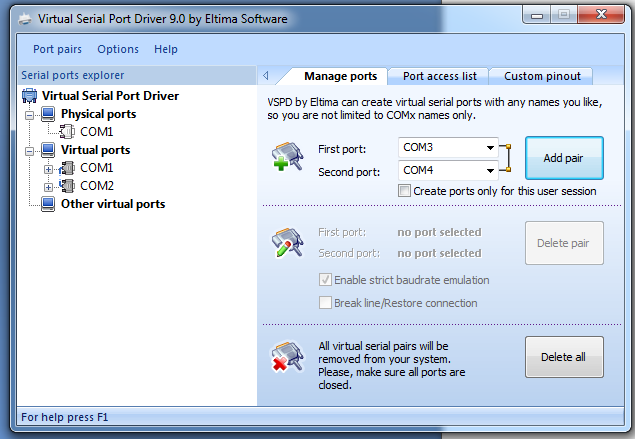


The first problem encountered is that I have only one MCU, therefore I had to use Proteus simulator.

Now that Proteus simulator is used. The mobile app will be replaced by a virtual terminal to input to the first Bluetooth HC-06 Module using RXD and TXD. Which can be seen in the following figure:

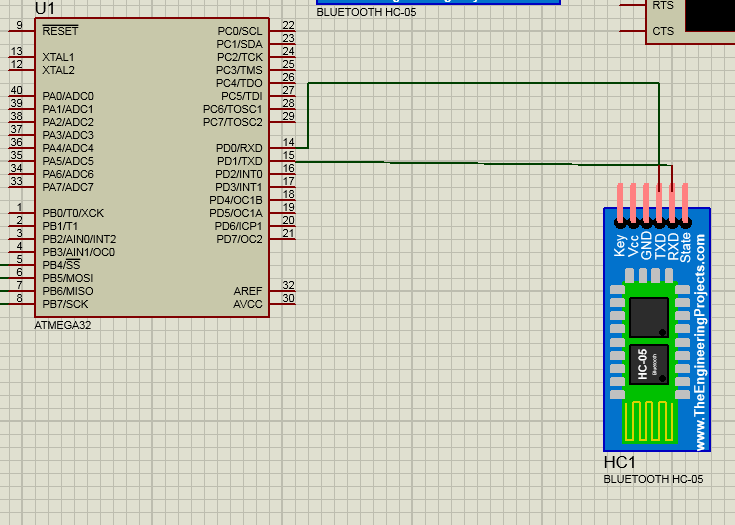


The First Bluetooth module is paired with the second Bluetooth module also HC-06 using a program called Virtual Serial Port Driver. This program recognizes the two modules as COM1 and COM2 and pairs them. This can be seen in the following figure:

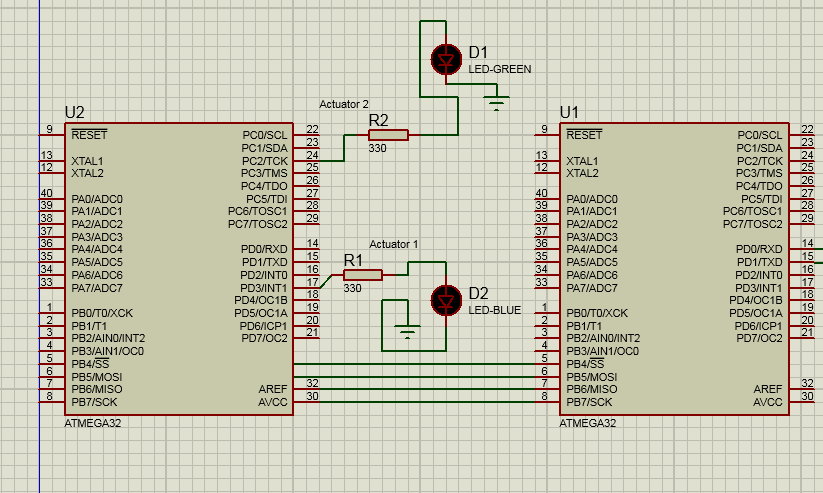


Now that communication between the Bluetooth has been done.

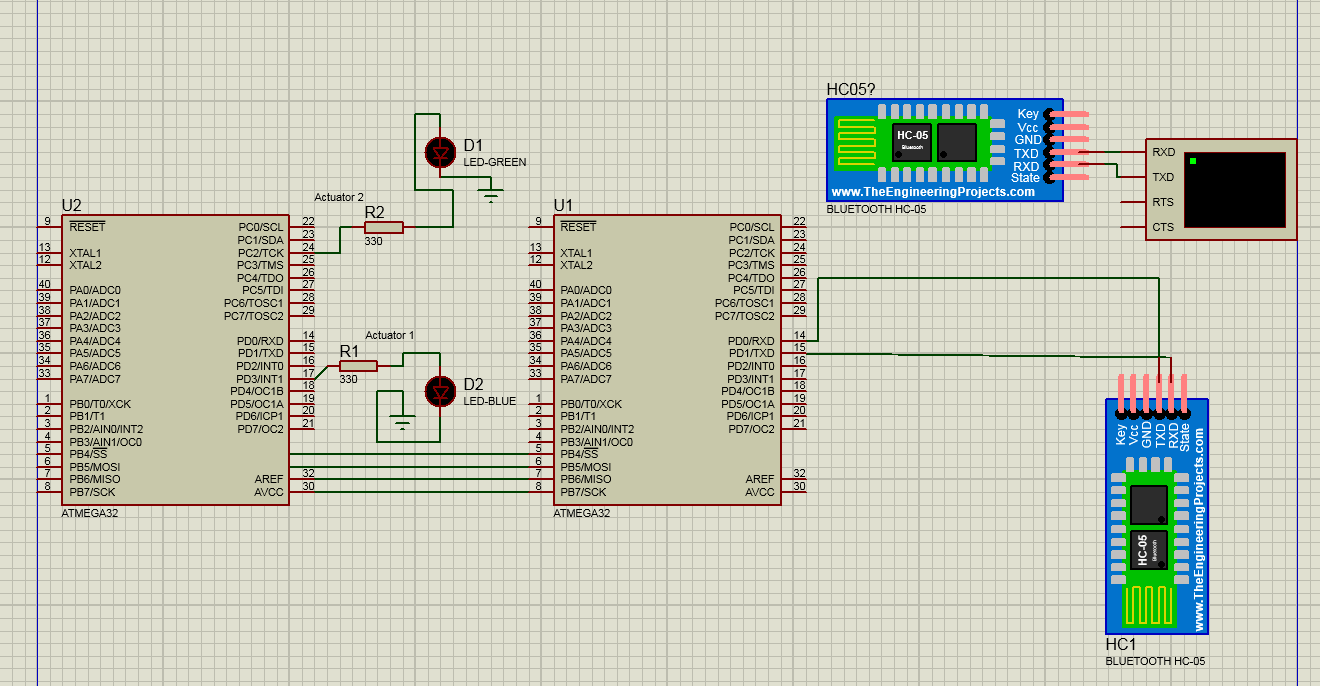
The next step is connecting the second Bluetooth module to the Master Atmega32 MCU. This communication is done using UART communication protocol. Where he RXD and TXD of the MCU is connected to RXD and TXD in the Bluetooth.



The third step is connecting both MCU's using SPI communication, where the first MCU that is connected to the Bluetooth is the Master while the Second that is connected to the actuator's or LED's is the slave. In order for SPI communication to occur SS,MOSI,MISO and SCK are connected to each other. Which can be seen in the following figure:

 Finally, the Slave receives the Data that is transmitted from the master and checks if it's a value of char 2 or char 1, if it's a signal of '1' then the actuators connected to PC2 and PD3 are "activated" , if it's a value of '2' then it's "deactivated" .

The whole Image can be seen as follows:



**The Code:**

Regarding the Code and the Architecture used. The MCAL LAYER for the Master IS DIO,UART and SPI. While, the MCAL layer for the slave is DIO and SPI.

The master had to initialize UART to interact with Bluetooth module through TXD and RXD channels. In addition, initialize SPI to connect with the second MCU(Slave).

The slave needed SPI to receive data and check on this data, if it's a value of char '1' then we activated the LED's using DIO, if it's a value of '2' then we deactivate it. For being more professional it's defined using #define for the code to be more readable.

Mainly it was figuring out how to include Bluetooth devices in Proteus then using a virtual serial channel to connect both modules by pairing them while using a virtual terminal for input, then integrating the UART for Bluetooth communication then SPI for communication between MCU'S and then finally ON/OFF the actuators using DIO.