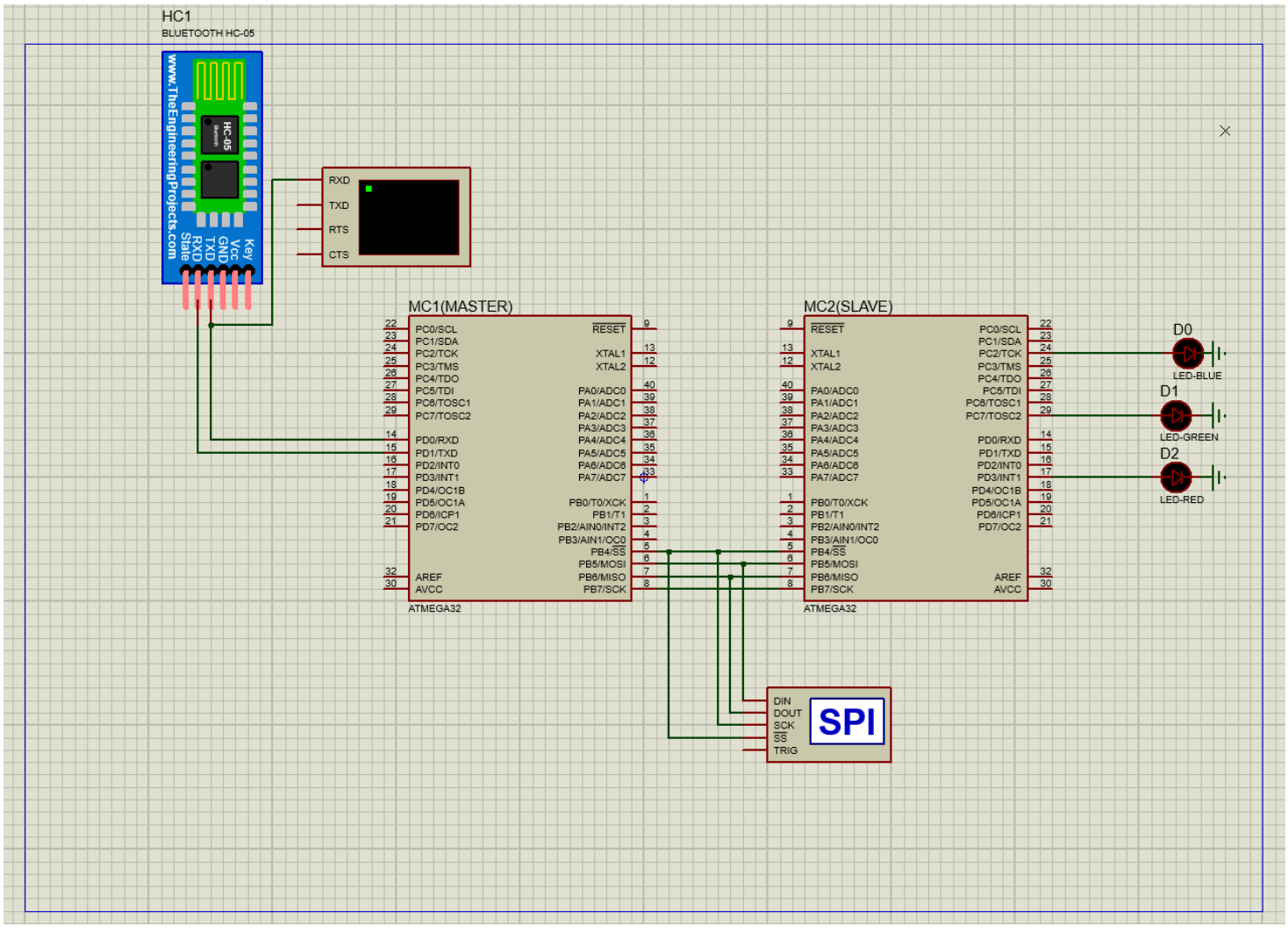
Smart Home based Bluetooth : based on 2 ATmeaga16 microcontrollers that communicates using SPI protocol and Bluetooth module (HC-05) communicates with first MCU (MASTER) using UART protocol, first MCU (MASTER) take a signal from the user on mobile phone sent it via Bluetooth , second MCU (SLAVE) take that signal from the first MCU (MASTER) to control and handles all the system logic and choose which led can be turn on, All drivers are implemented using the configurations technique, different design methods used like interrupts, call backs, and periodic polling, and software layers for portability and maintainability purpose with high quality and readable code with good comments. Drivers designed for the project: UART, SPI, LED Sequence of the Application as follow:

At first time to run the program: first MCU (MASTER) will wait to take input from Bluetooth via UART protocol in order to send it to the second MCU (SLAVE) ,the user will have the choice to turn on any of three led through three character , for the first led to turn on the user will have to enter character ‘a’ the first led will turn on for one second and then turned off again for second led the user should enter character ‘b’ as input through mobile to turn on the second led which takes only 1 second and return back to off states and the last led which will be turned on for one second if the user send a input of character ‘c’, every led can be turned on individually, after the led turned on for a second and back again to off state user should wait 0.5 second to turn another one. UART Interrupt is used as soon as the user send data via Bluetooth, so data received in the interrupt service routine and transmitted via SPI communication to the second MCU (SLAVE) which enable the SPI Receive interrupt enabled as soon as the SPIF flag is set and the chosen led is turned on for a while.



Proteus 8 Professional Release 8.9 SP2 (Build 28501) is used to simulate the whole Project.

Using 2 ATMEGA32 MCU one act as Master give the order to the second one which act as slave to control the system, Using Virtual Terminal to verify the input data coming from Bluetooth module, the SPI DEUBGGER used here to validate the input coming from the user.