AMIT Learning Graduation Project
Smart Home

Ahmed Rady El-Nashar

Abstract:

The Smart Home project is a simple embedded system which performs certain functionalities based on the commands sent from the user's phone via Bluetooth; such as turning lights on/off and alarm mode.

1. Objective:

The aim of this project is to successfully integrate Bluetooth module HC-05 with UART peripheral of an ATmega32 microcontroller which acts as the commander/master to send the data received from the UART to another ATmega32 which acts as controller/slave to perform certain functionalities on three LEDs connected on the first three pins of Port D based on the command sent from the user's phone through its Bluetooth peripheral.

2. Parts list and cost:

The table below shows the parts used to complete this project and its cost:

Item	Quantity	Cost per item (EGP)	Total cost (EGP)
HC-05	1	125	125
ATmega32	2	75	150
220 ohm Resistors	3	0.5	1.5
LED	3	0.25	0.75
Oscillator 8Mhz	2	3	6
Capacitor 22pf	4	0.25	2
Battery 9V	1	18.5	18.5
Battery Clip DC male	1	6.5	6.5
Screw Terminal Block To Female DC Power Adapter	1	6	6
Wire	15	0.25	3.5
Bread board	1	30	30
Smart Phone	1	Use yours	
			350

Table 2.1 Parts list and cost.

3. Design and Implantation:

3.1 Hardware Design

3.1.1 Project circuit diagram.

The project was designed and simulated using Proteus 8.0 as follows:

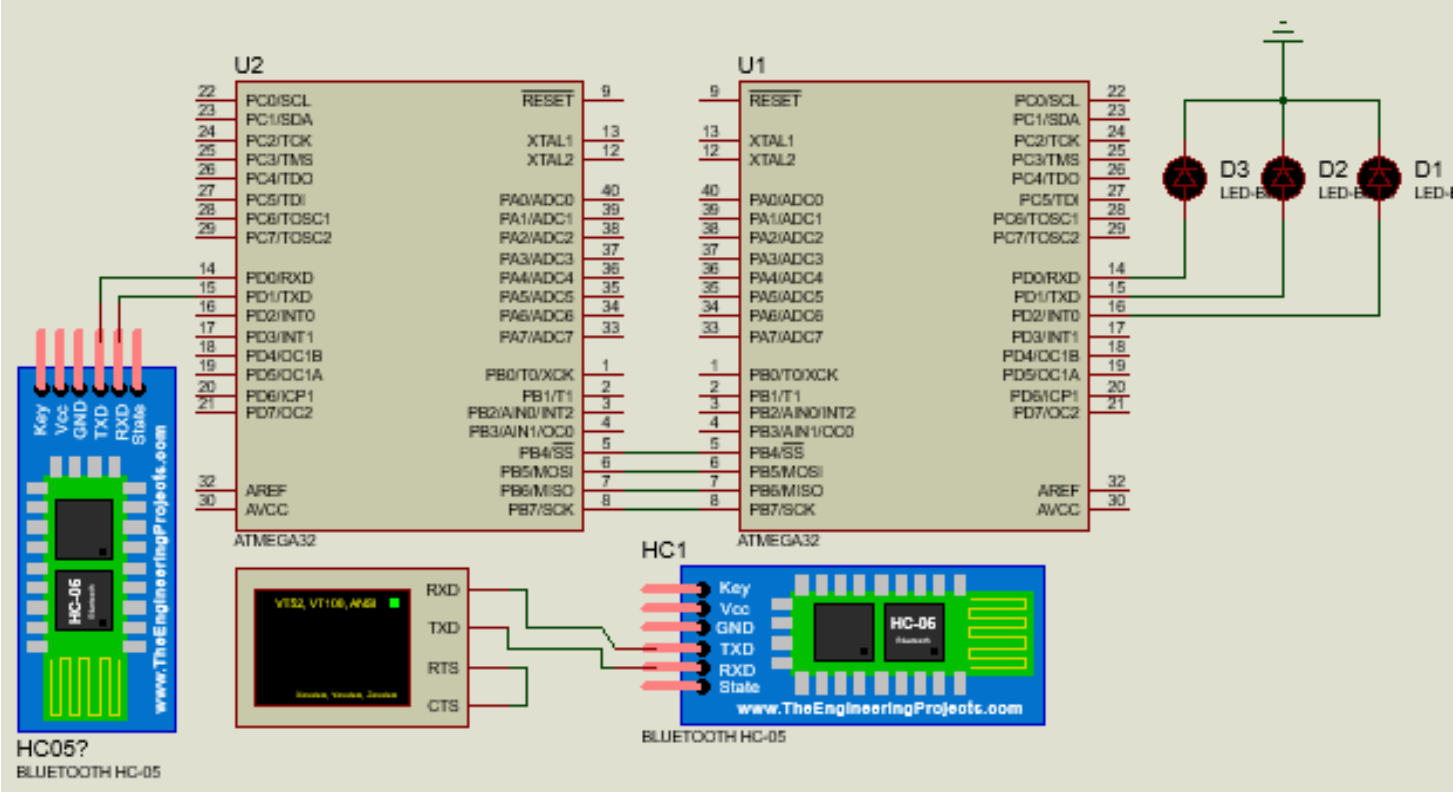


Fig 3.1.1 Project Schematic.

3.1.2 Project block diagram.

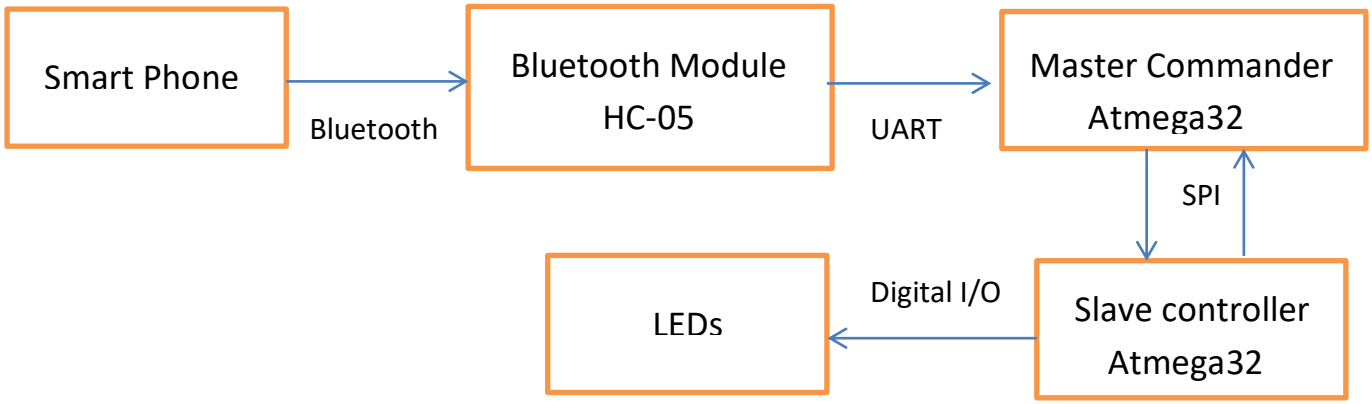


Fig 3.1.2 Project Block diagram.

3.1.3 Hardware Interface:

Since Proteus simulation software was used to simulate the project the following steps was used:

1. Each Bluetooth module was assigned a COM port.

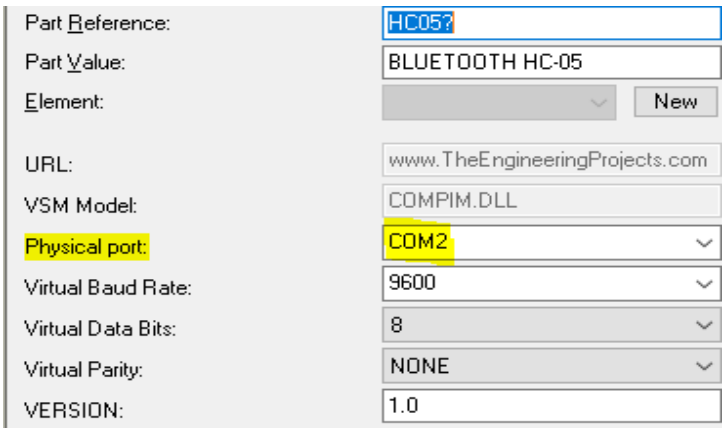


Fig 3.1.3.1 HC-05 (HC2) settings.

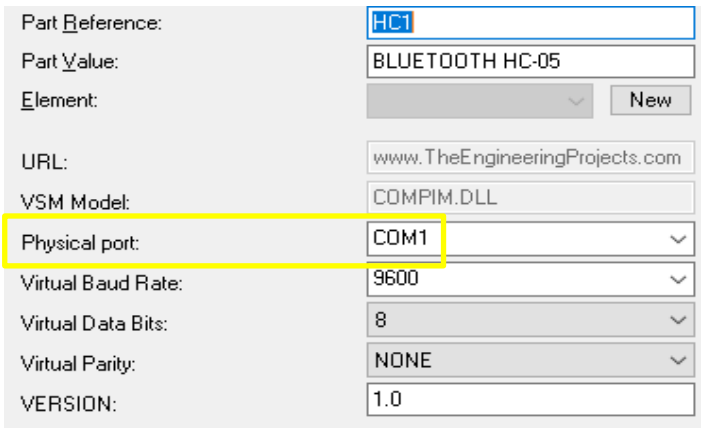


Fig 3.1.3.2 HC-05 (HC1) settings.

2. Both COM Ports was connected virtually using Virtual Serial Port Driver 8.0

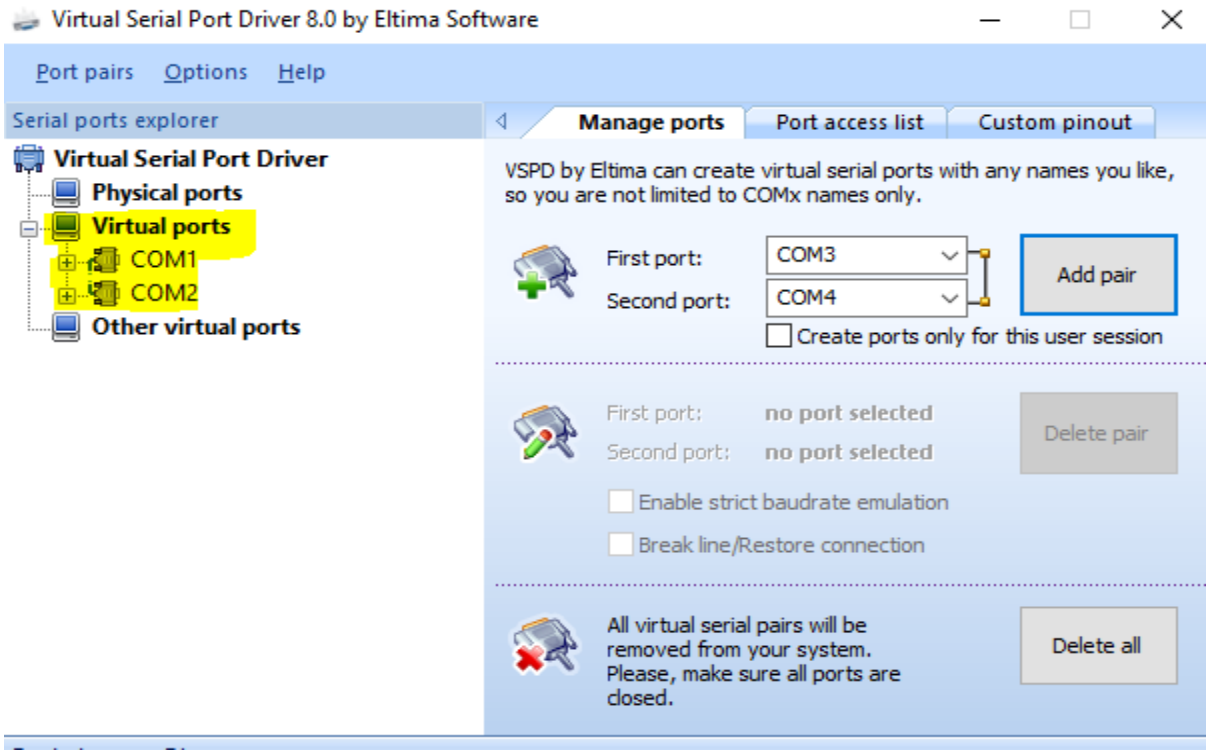


Fig 3.1.3.2COM Ports Virtual settings.

3. Virtual terminal from Proteus SW connected to HC-05 (HC-1) via UART pins was used to simulate sending data via Bluetooth from a smart phone.

4. Both ATmega32 Clock Frequency was set to 8MHz.

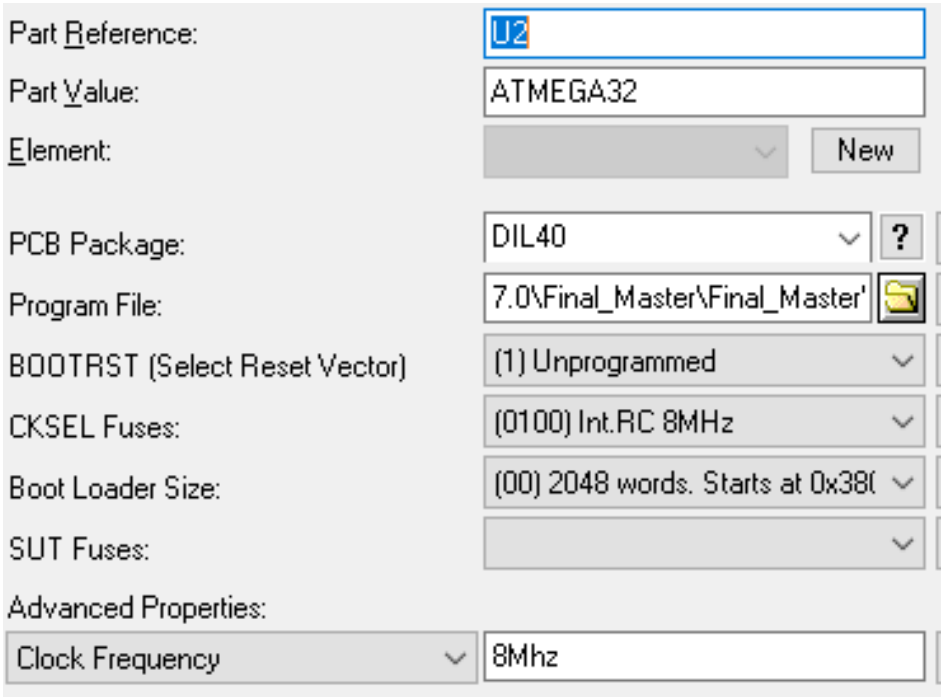


Fig 3.1.3.2 AT-mega32 settings.

5. SPI was used as Communication protocol between both ATmega32 where the master Atmega32 acts as a commander by sending the user commands received from the virtual terminal via master UART to the Slave ATmega32 which acts as controller and executer of these commands.

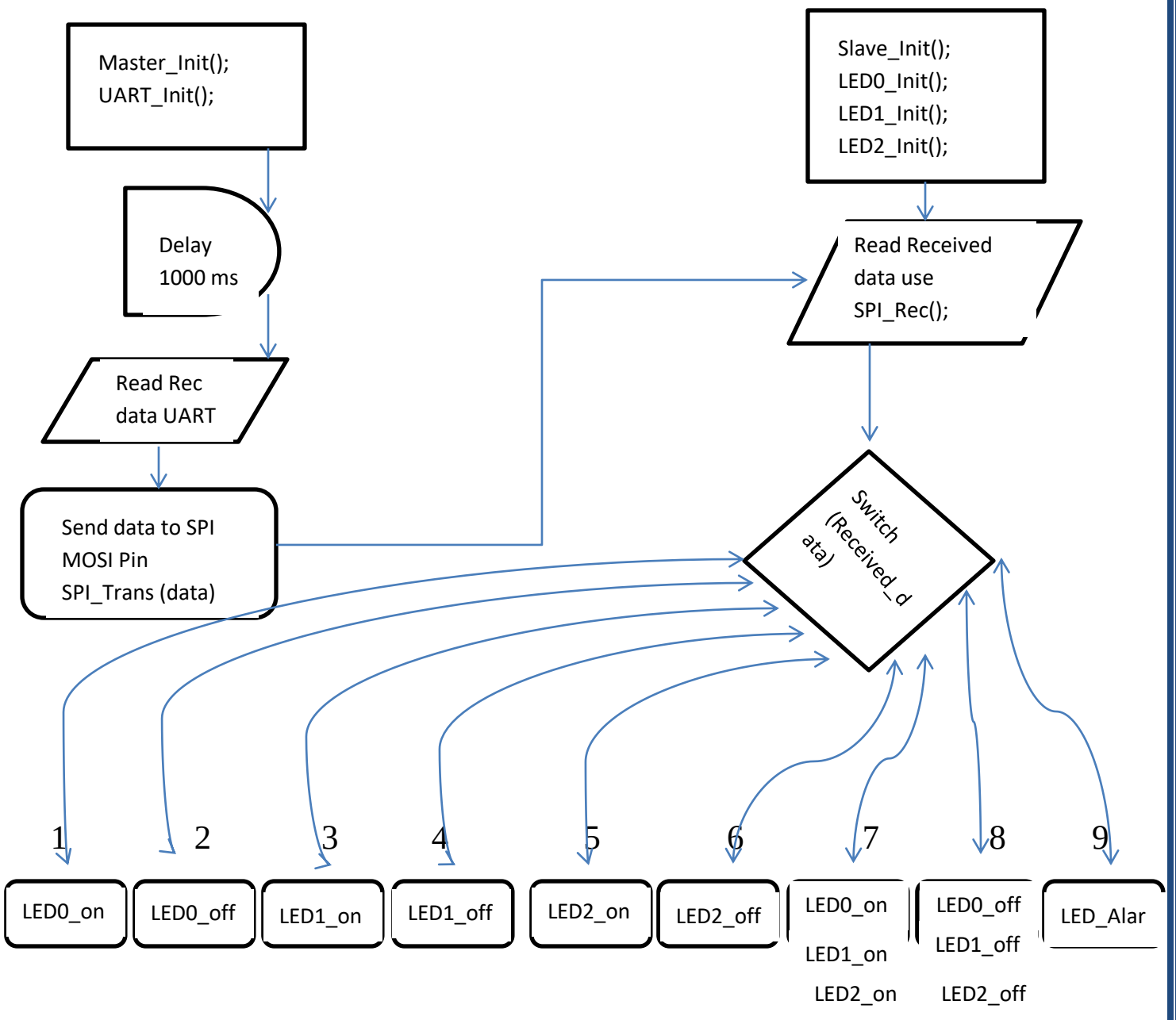
6. Digital I/O pins (0, 1 and 2) of Port D of the Slave ATmega32 was used to act as an output pins.

3.2 Software Design

3.2.1 Architecture Layers

	Master ATmega32	Slave ATmega32
Application Layer	Commander.c	Executer.c
HAL	Bluetooth_UART.h	LED.h
MCAL	SPI.h, STD.h, ATMEGA32_Regs.h, DIO.h, BIT_Math.h	

3.2.2 SW Flow Chart



4. Testing and Technical Challenges:

4.1 Testing

First Test:

Input	Expected output	Actual Output	Comment
1	LED0_ON	All remained Off (Idle state)	Test UART
2	LED0_OFF		
3	LED1_ON		
4	LED1_OFF		
5	LED2_ON		
6	LED2_OFF		
7	All LEDs On		
8	All LEDs Off		
9	LEDs flashing		

To test UART one LED was added to the Master ATmega32 to turn on if data received.

Second Test:

Input	Expected output (Slave)	Expected output (Master)	Actual Output	Comment
1	LED0_ON	LED_ON	All remained Off (Idle state)	Troubleshoot UART
2	LED0_OFF	LED_ON		
3	LED1_ON	LED_ON		
4	LED1_OFF	LED_ON		
5	LED2_ON	LED_ON		
6	LED2_OFF	LED_ON		
7	All LEDs On	LED_ON		
8	All LEDs Off	LED_ON		
9	LEDs flashing	LED_ON		

4.2 Technical Challenge Solved

After troubleshooting baud rate of both HC-05 Bluetooth module and Master ATmega32 was the same at 9600 but the clock frequency of the Atmega32 was 1MHz instead of 8MHz as mentioned in the Hardware interface section.

Third Test:

Input	Expected output (Slave)	Expected output (Master)	Actual Output	Comment
1	LED0_ON	LED_ON	LED0_ON	LED on master stated that the UART started working fine
2	LED0_OFF	LED_ON	LED0_OFF	
3	LED1_ON	LED_ON	LED1_ON	
4	LED1_OFF	LED_ON	LED1_OFF	
5	LED2_ON	LED_ON	LED2_ON	
6	LED2_OFF	LED_ON	LED2_OFF	
7	All LEDs On	LED_ON	All LEDs On	
8	All LEDs Off	LED_ON	All LEDs Off	
9	LEDs flashing	LED_ON	LEDs flashing	