AMIT - Embedded Systems Project.

Aya Bassel Ibrahim Fahmy

L11

1 Introduction and Overview.

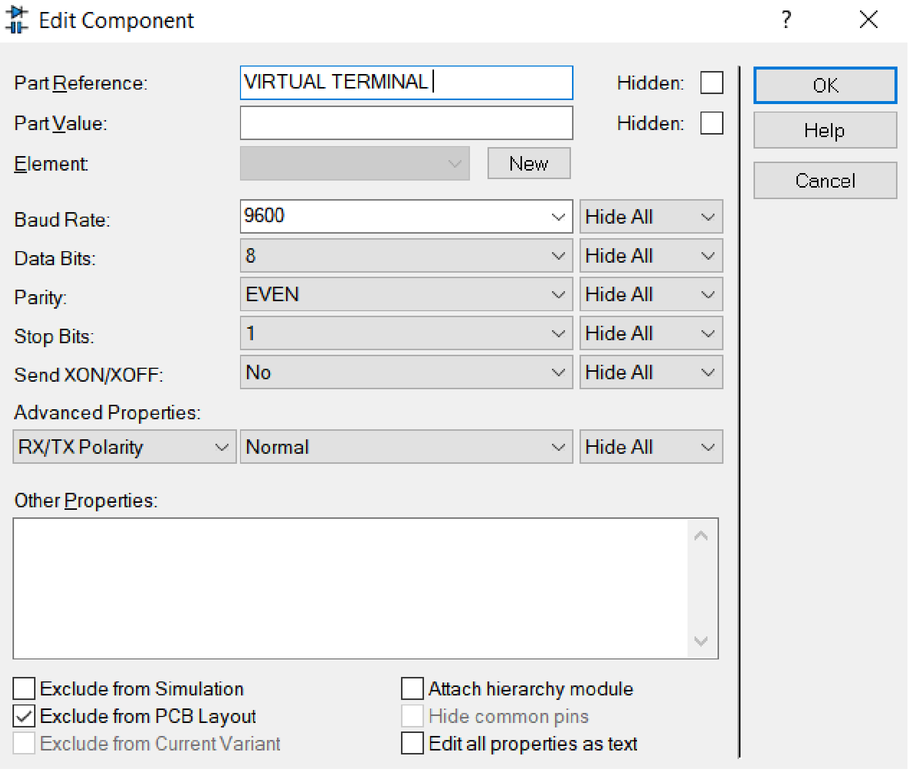
This project aims to design and implement an optimal and feasible structure for a Smart Home application. Firstly, a Bluetooth module takes an input then using UART transmits the input signal to the first controller, The Master. Then the signal is exchanged by the master and the slave using SPI which controls actuators. Consequently, this algorithm can be generically developed for any application that requires communication.

2 Setting Up the Environment.

Regarding the environment, this project simulations was carried out using Proteus 8 professional. with the following set up.

2.1 Bluetooth Modules.

The input signal is transmitted from a virtual terminal to the first Bluetooth module using UART with $BaudRate : 9600$ and $frameformat : 8\ data\ bits$, $Even\ parity$ and 1 $stop\ bit$.



Edit Component

Part Reference: VIRTUAL TERMINAL Hidden: □
Part Value: Hidden: □
Element:

Baud Rate: 9600 Hide All
Data Bits: 8 Hide All
Parity: EVEN Hide All
Stop Bits: 1 Hide All
Send XON/XOFF: No Hide All

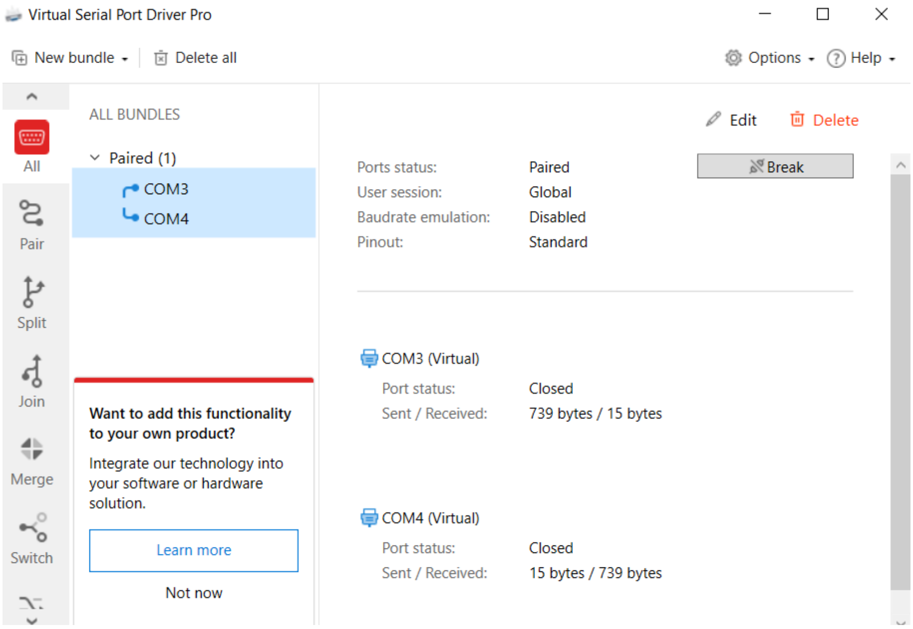
Advanced Properties:
RX/TX Polarity Normal Hide All

Other Properties:

□ Exclude from Simulation
☑ Exclude from PCB Layout
□ Exclude from Current Variant
□ Attach hierarchy module
□ Hide common pins
□ Edit all properties as text

Figure 1: Virtual Terminal Settings

Then the Signal is transmitted to the second Bluetooth module using a Virtual Serial Driver Port which pairs virtual physical ports $COM3$ and $COM4$ for the first and second Bluetooth modules respectively.



Paired (1)
- COM3
- COM4

Ports status: Paired
User session: Global
Baudrate emulation: Disabled
Pinout: Standard

COM3 (Virtual)
Port status: Closed
Sent / Received: 739 bytes / 15 bytes

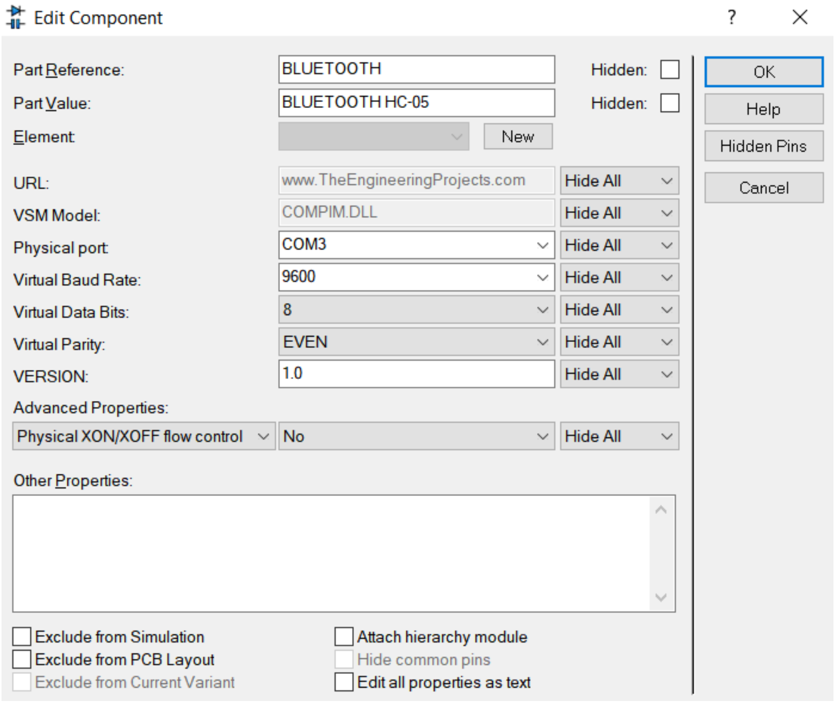
COM4 (Virtual)
Port status: Closed
Sent / Received: 15 bytes / 739 bytes

Want to add this functionality to your own product?

Integrate our technology into your software or hardware solution.

Figure 2: Virtual Serial Port Driver Pro Pair

Figures (3,4) shows the settings for both Bluetooth modules with the Baud Rate, Frame Formate and the Virtual Physical Ports.



Part Reference:	BLUETOOTH	Hidden: □	
Part Value:	BLUETOOTH HC-05	Hidden: □	
Element:			
URL:	www.TheEngineeringProjects.com	Hide All	
VSM Model:	COMPIM.DLL	Hide All	
Physical port:	COM3	Hide All	
Virtual Baud Rate:	9600	Hide All	
Virtual Data Bits:	8	Hide All	
Virtual Parity:	EVEN	Hide All	
VERSION:	1.0	Hide All	

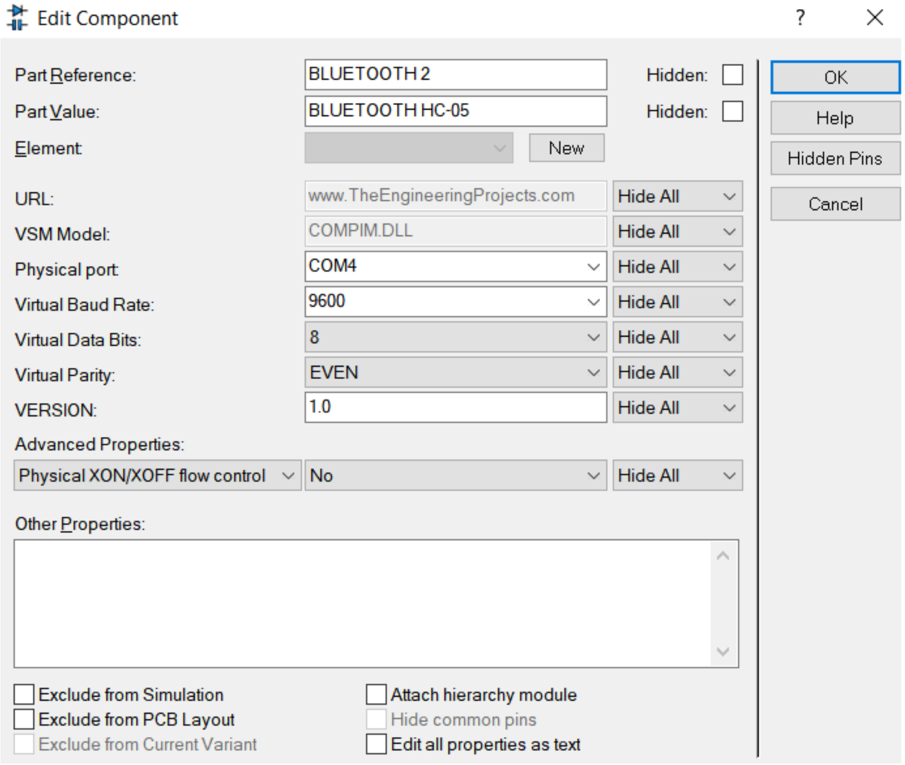
Advanced Properties:

Physical XON/XOFF flow control	No	Hide All

Other Properties:

□ Exclude from Simulation
□ Exclude from PCB Layout
□ Exclude from Current Variant
□ Attach hierarchy module
□ Hide common pins
□ Edit all properties as text

Figure 3: The First Bluetooth Module.



Edit Component

Part Reference:	BLUETOOTH 2	Hidden: □	
Part Value:	BLUETOOTH HC-05	Hidden: □	
Element		New	
URL:	www.TheEngineeringProjects.com	Hide All	
VSM Model:	COMPIM.DLL	Hide All	
Physical port	COM4	Hide All	
Virtual Baud Rate:	9600	Hide All	
Virtual Data Bits:	8	Hide All	
Virtual Parity:	EVEN	Hide All	
VERSION:	1.0	Hide All	

Advanced Properties:

Physical XON/XOFF flow control: No — Hide All

Other Properties:

□ Exclude from Simulation
□ Exclude from PCB Layout
□ Exclude from Current Variant
□ Attach hierarchy module
□ Hide common pins
□ Edit all properties as text

Figure 4: The Second Bluetooth Module.

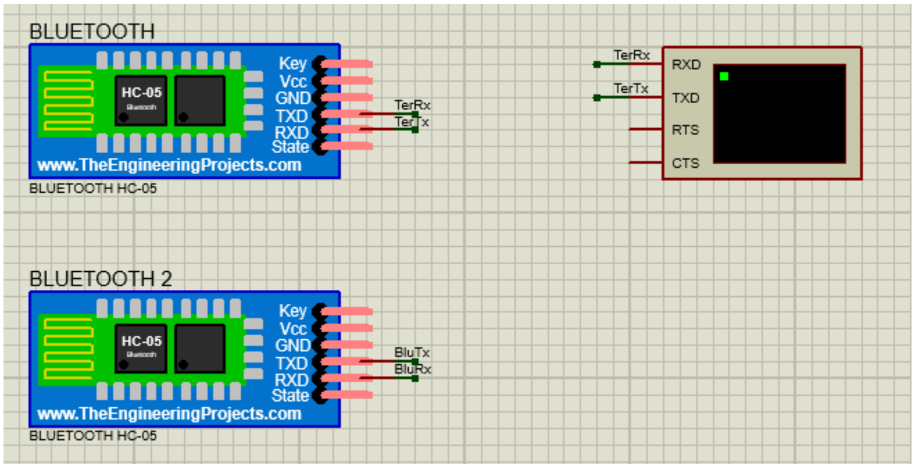


Figure 5: Bluetooth Communication: Proteus 8 Professional

Figure 5 shows the communication of the virtual terminal with the Bluetooth module by which the terminal Tx is connected to the Bluetooth Rx and vise verse.

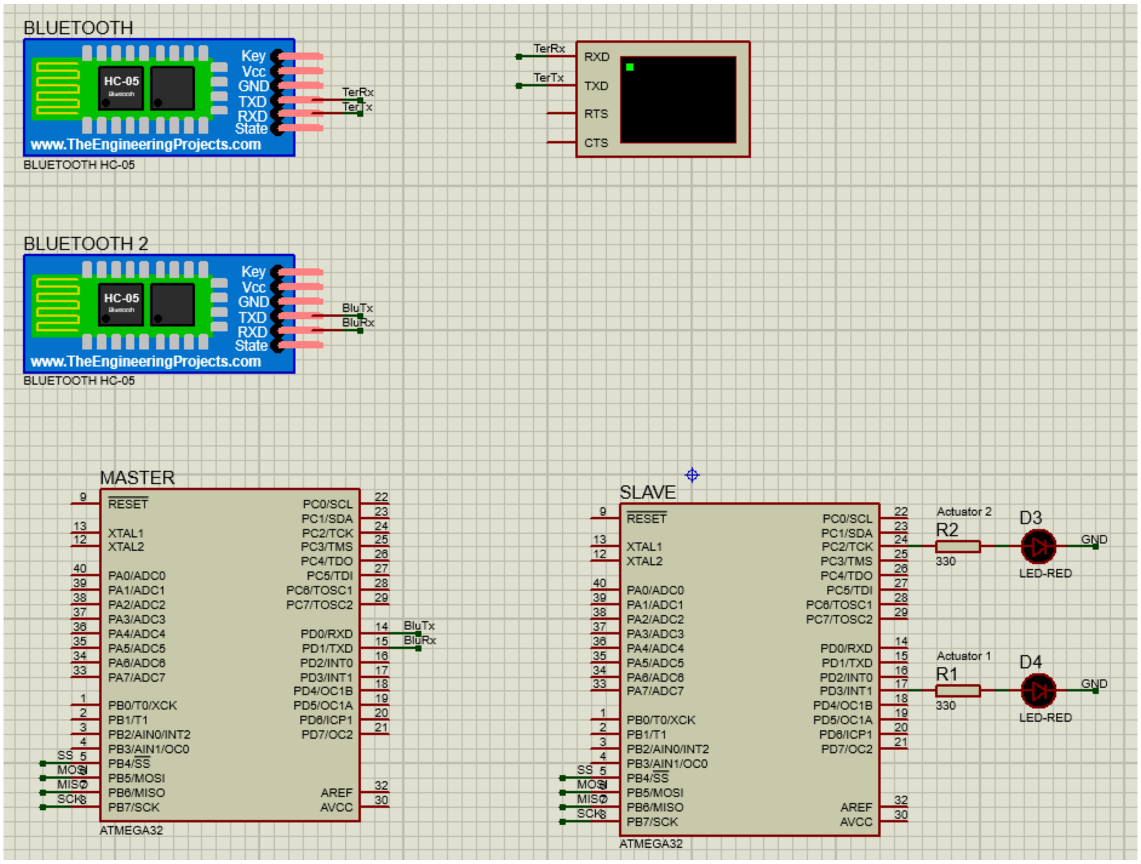


Figure 6: Project Environment on Proteus 8 Professional

Furthermore, figure 6 shows the wiring between the Second Bluetooth module and the Master Atmega32 where the Tx of the module is connected to the microcontroller's Rx and vise verse. Regarding the second stage of communication, The master Micro-controller is communicating with a Second, Slave, Atmega32 using SPI where the SS, $MOSI$, $MISO$, SCK is connected. Finally the Slave Atemga32 manipulates 2 LED outputs as an Acturators for the desired functionality.

3 Algorithm Structure.

The algorithm was developed based on modular programming and the Layered Architecture mentality which is illustrated in the following figure.

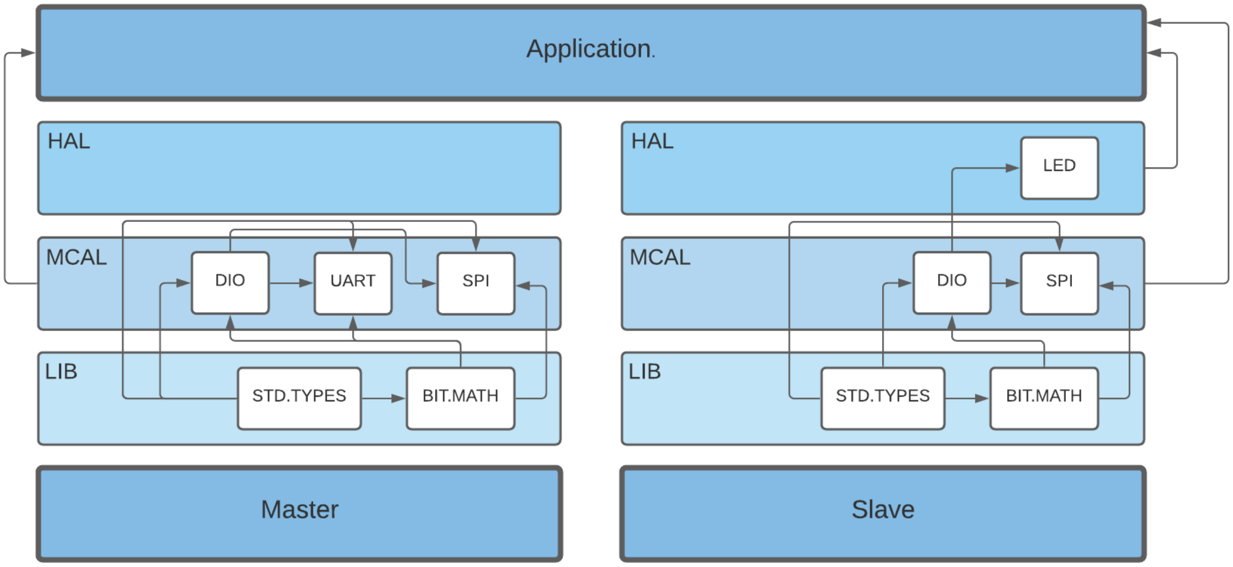


Figure 7: Layered Architecture

3.1 Flowchart.

The following flowcharts shows the algorithm for the master and slave micro controllers based on modular programming. Please refer to the code comments for exact explanation and details of each module.

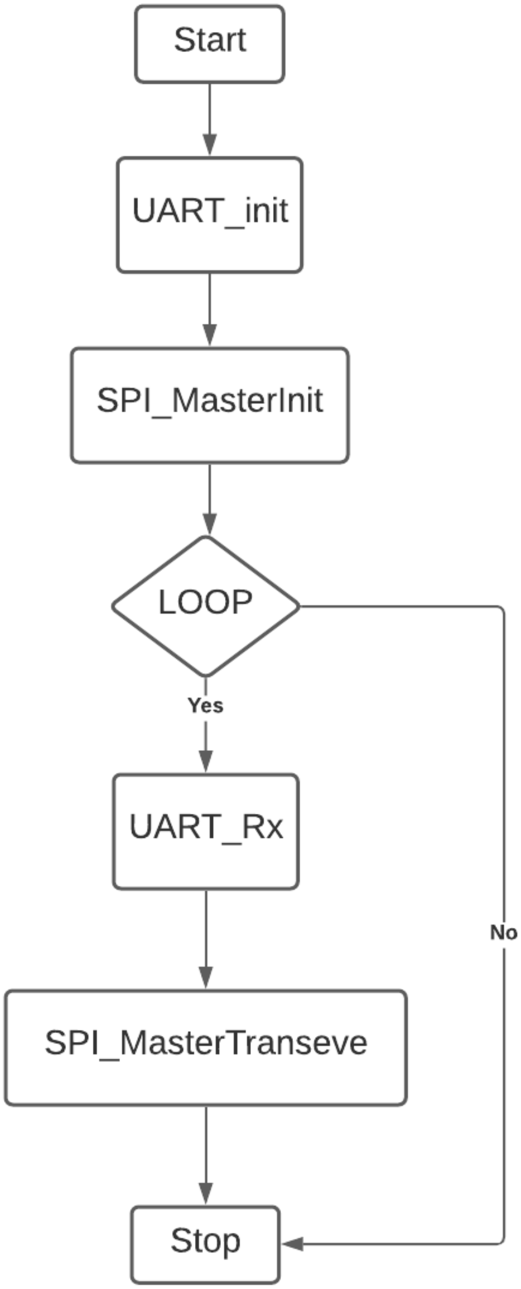


Figure 8: Master Flowchart

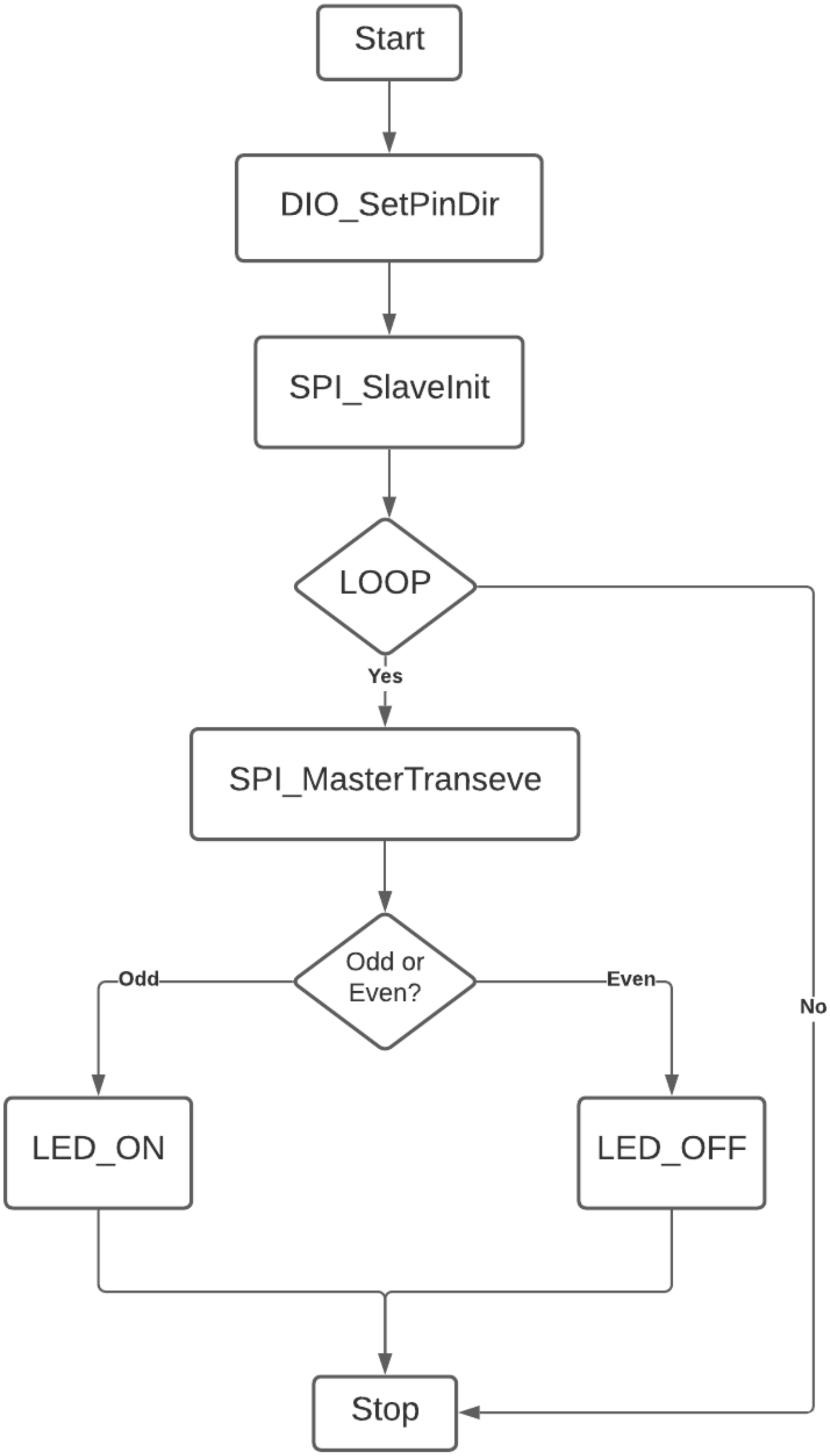


Figure 9: Slave Flowchart

4 Git Hub Link

https://github.com/AyaBassel3/AMITproject.git