UART COMMUNICATION INTERFACE FOR SIGNAL GENERATOR

1. UART Communication Features

Implementation:

- Data Bits: 8 bits
- Parity: None
- Start Bit: 1 bit
- Stop Bit: 1 bit
- Data Rate: 115200 bps
- Transfer Mode: Full Duplex

2. UART Communication Protocol

Usage: The protocol defines the structure and format of the messages exchanged between the controller and the signal generator.

Implementation:

- **Headers:** Two headers (0xAA and 0x55) to indicate the start of a message.
- **Message Type (MsgType):** Identifies the type of message being sent (e.g., Write Data, Read Data, Data Response).
- **Length (Len):** Indicates the length of the data portion of the message.
- **Data:** Contains the actual information being transmitted (e.g., waveform type, frequency, phase).
- **Checksum (ChkSum):** Used for error-checking to ensure data integrity.

3. UART Message Types

Usage: Different types of messages facilitate various operations, such as writing data to the signal generator or reading data from it.

Implementation:

- **Write Data (0x99):** Sent from the controller to the signal generator to configure waveform type, frequency, and phase data.
- **Read Data (0x44):** Sent from the controller to request data from the signal generator.
- **Data Response (0x58):** Sent from the signal generator to the controller in response to a read request, containing the requested data.
- **ACK (0x22):** Acknowledgment message sent to confirm the receipt of data.

- **NACK (0x11):** Negative acknowledgment message sent to indicate an error or issue with the received data.

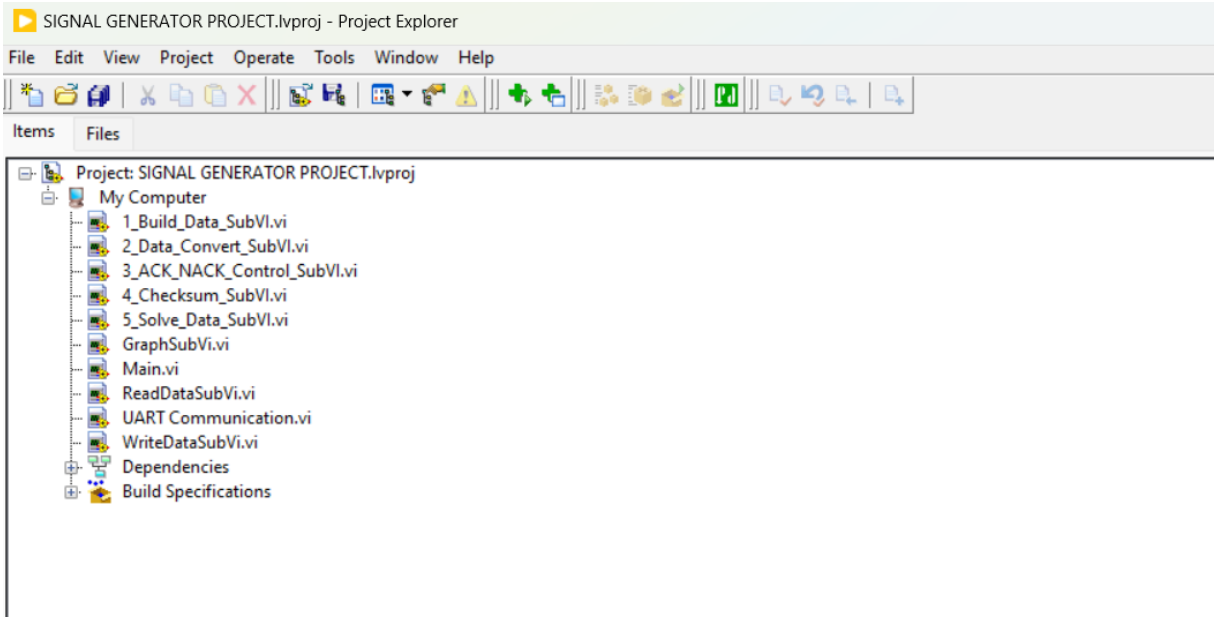
4. UART Message Details

Usage: Specifies the detailed structure of the data being transmitted, ensuring that both the controller and signal generator interpret the data correctly.

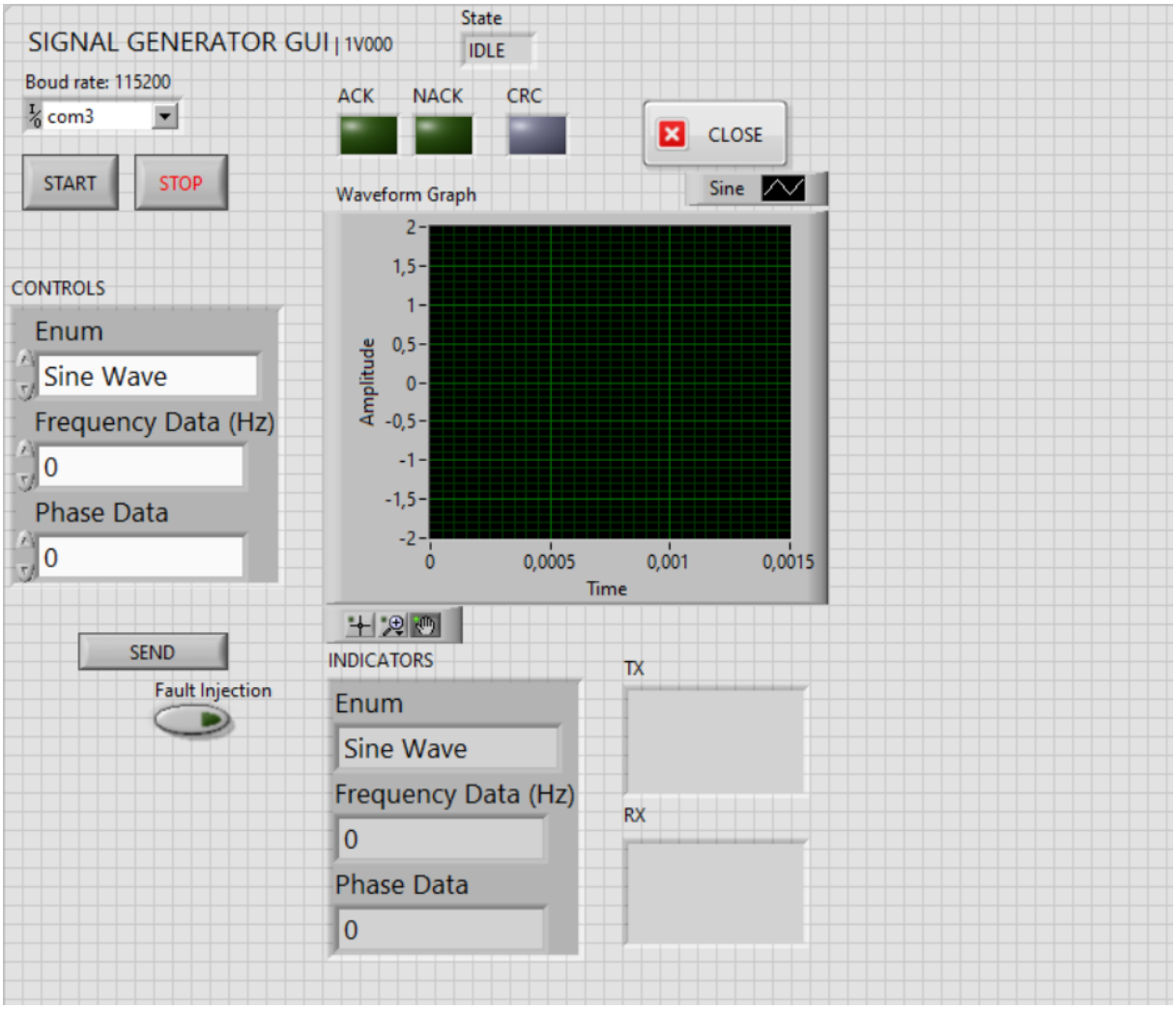
Implementation:

- **Waveform Type (2 bits):**
  - “00”: Sine Wave
  - “01”: Triangular Wave
  - “10”: Square Wave
  - “11”: Sine Wave
- **Frequency Data (28 bits):**
  - Range: 0 to 268435455 Hz
  - Resolution: 0.279 Hz
  - Max Value: 37.5 MHz
  - Min Value: 0 Hz
- **Phase Data (12 bits):**
  - Range: 0 to 360 degrees
  - Resolution: 0.087890625 degrees
  - Max Value: 360 degrees
  - Min Value: 0 degrees

5. PROJECT HIERARCHY



5.1. Main.vi



There are 2 main sections lay on this front panel: CONTROLS and INDICATORS

CONTROLS: Takes wave type, frequency and phase from the user.

INDICATORS: Shows wave type, frequency and phase of data.

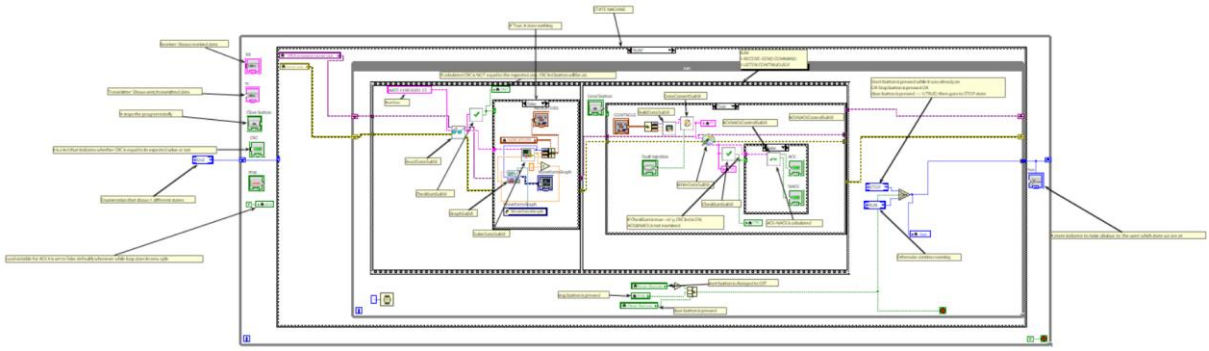
Baud Rate (115200 bps) refers to the rate at which information is transferred in a communication channel. In the context of UART (Universal Asynchronous Receiver/Transmitter) communication, it specifies the number of signal changes or symbols sent per second. It is measured in bits per second (bps).

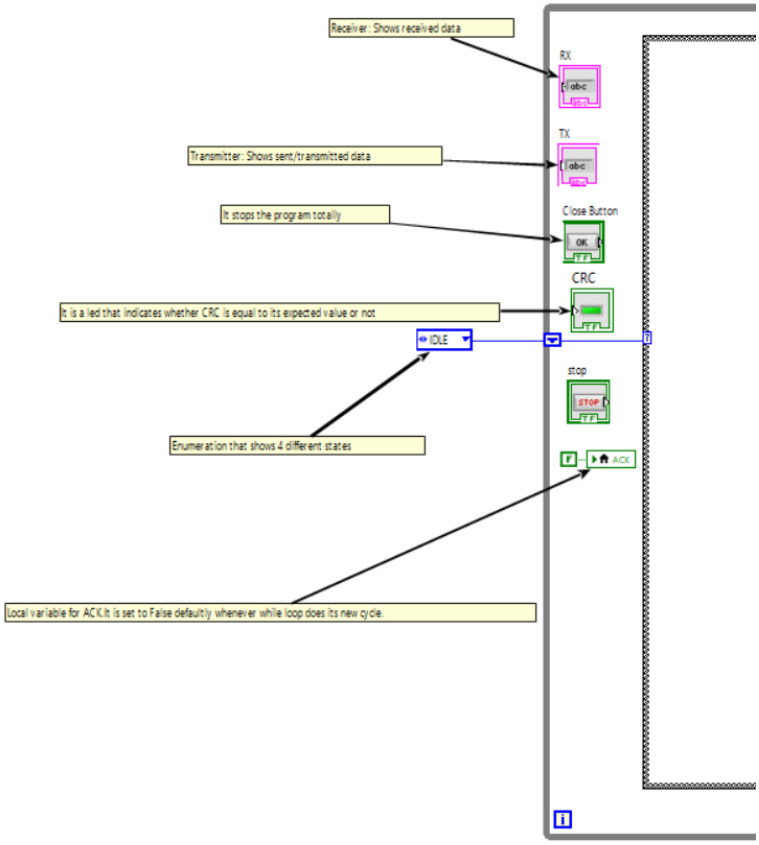
States: IDLE, START, RUN, STOP. State is visible to the user so that it can be investigated which state is being executed at the moment.

ACK (Acknowledgment): ACK is a signal used in communication protocols to indicate that a message has been successfully received and understood.

NACK (Negative Acknowledgment): NACK is a signal used to indicate that a message has been received but there is an error or the message cannot be processed.

CRC (Cyclic Redundancy Check): CRC is an error-detecting code used to detect accidental changes to raw data. It is a type of hash function that generates a short, fixed-length binary sequence, called a checksum, based on the data being transmitted.





Tx: Transmitted/Sent Data

Rx: Received Data

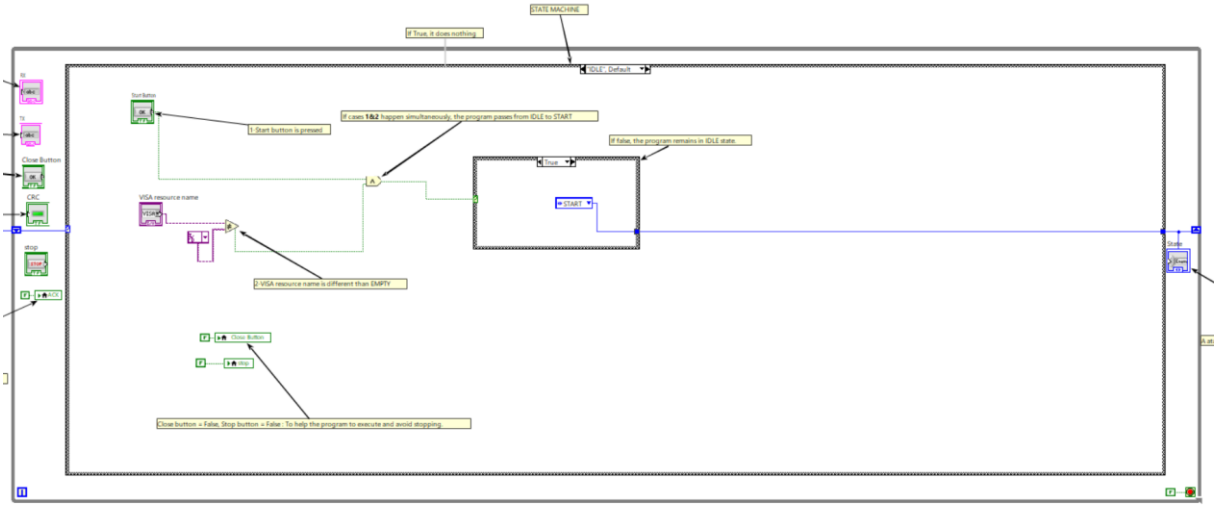
START: The program starts running whenever it is pressed. (mechanical action: switch when released)

SEND: It sends the input data and activate the VISA operations.

STOP: Stops the program as it is clicked.

CLOSE: It completely ends the program and bring it to IDLE state.

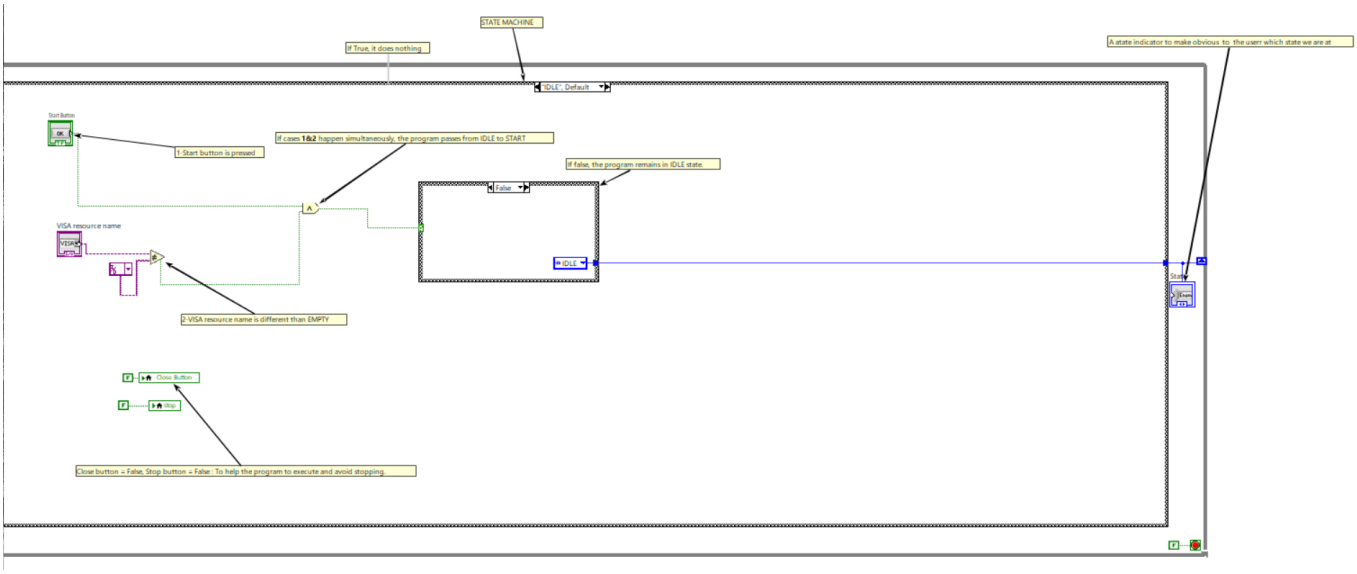
5.1.1.1 IDLE(True)



Start button is pressed and VISA resource name is different than EMPTY. If these two criterias meet at the same time, it is true condition. In this case, state goes from IDLE to START.

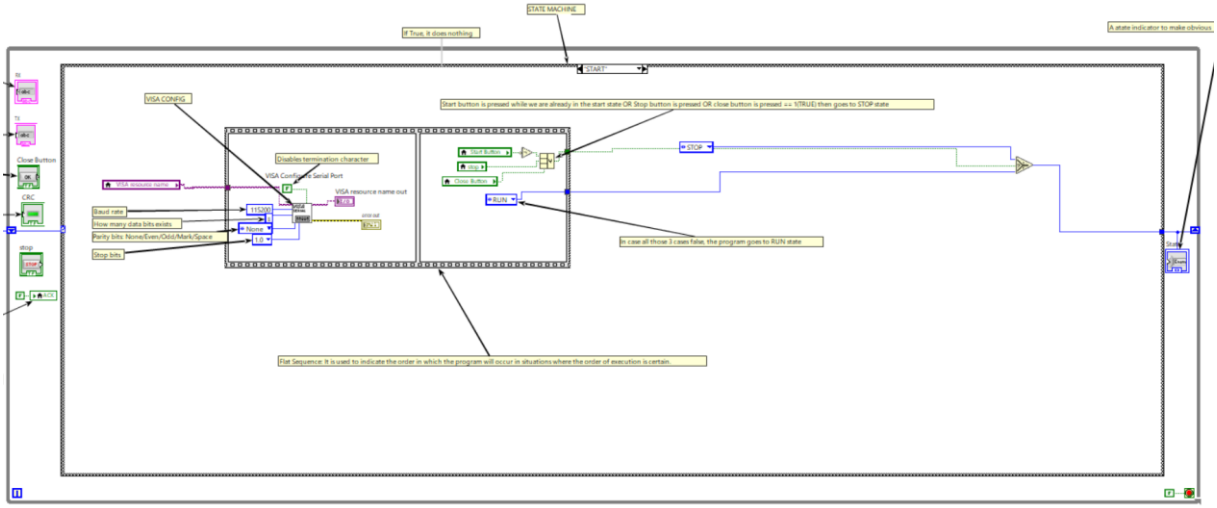
Close button = False and Stop button = False are set. If even one of them was true,the state would move to STOP.

5.1.1.2 IDLE(False)



In case start button is not hit or VISA resource name is empty or both of them, the False condition holds, which means we still at IDLE state.

5.1.2. START



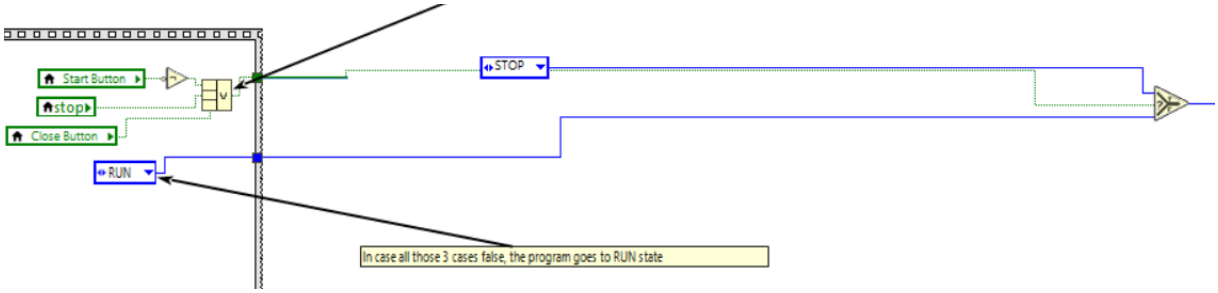
Start and COMPORT operations are held here: com1, com2, com3, com4, com5, com6 etc.

- Data Bits: 8 bits
- Parity: None
- Start Bit: 1 bit
- Stop Bit: 1 bit
- Data Rate: 115200 bps
- Transfer Mode: Full Duplex

ATTENTION: Since this laptop does not have any port, VSPE(Virtual Serial Ports Emulator), and Docklight applications was installed in order to create virtual ports to function this Project.

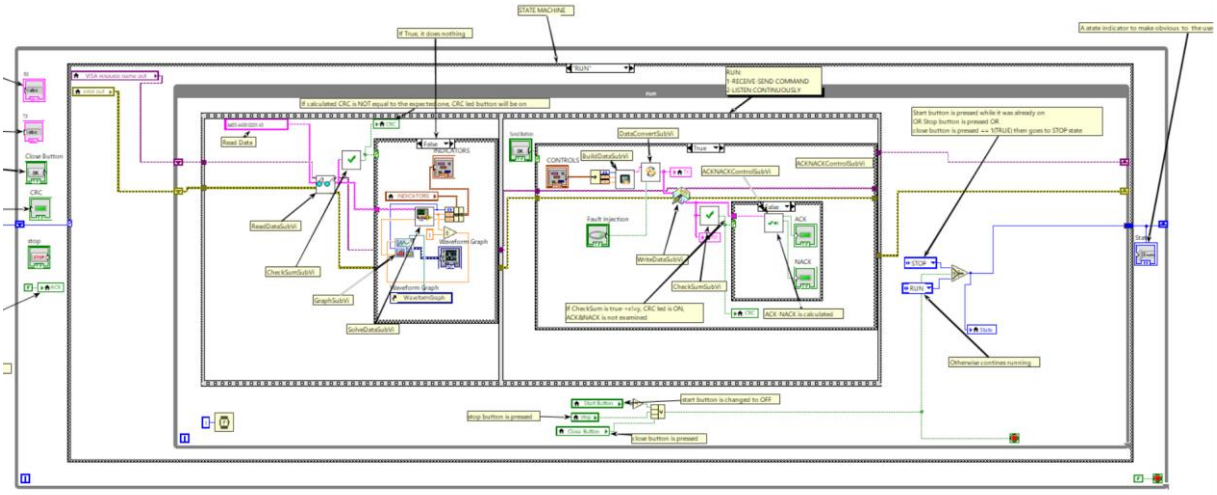
STATE TRANSITIONS:

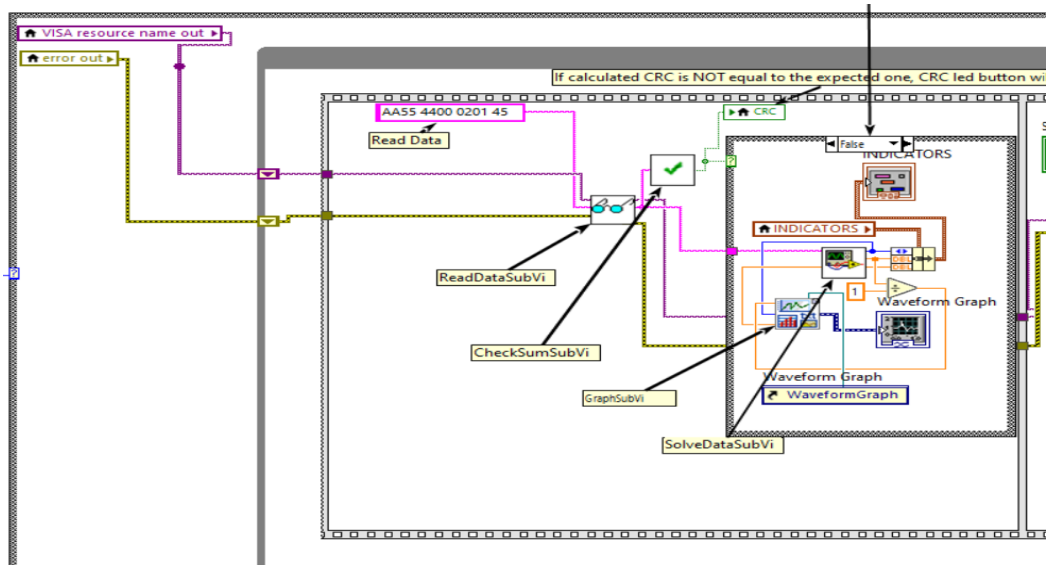
A. RUN: If nothing happens, the state will automatically move to RUN state.
B. STOP: A select structure is used.The first condition is that either start button clicked once more (OFF situation) or stop button is clicked or close button is clicked. As a result the state moves to STOP condition.



The second condition is that the program passes RUN state. If there is no any stimulus, program will remain in this phase.

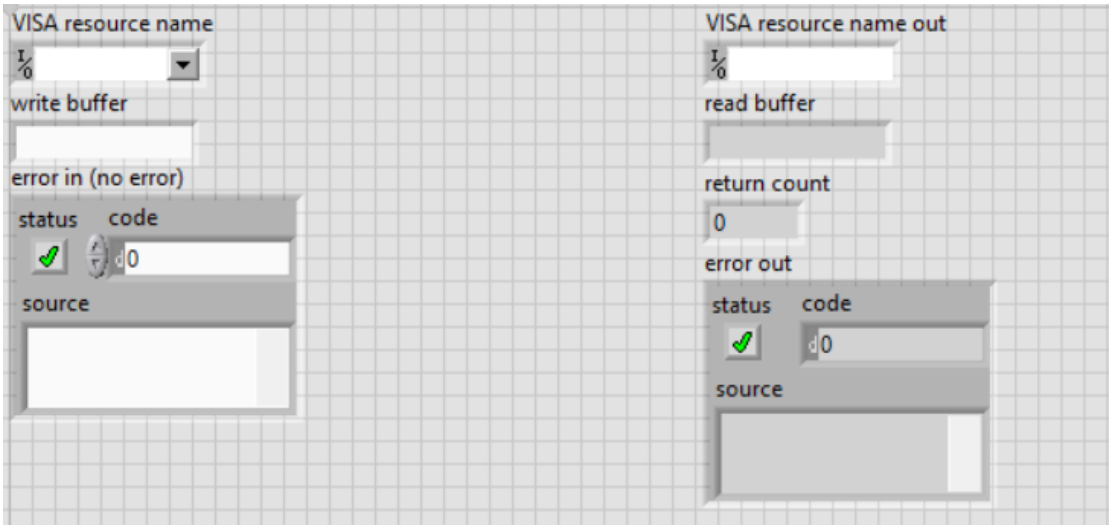
5.1.3 RUN



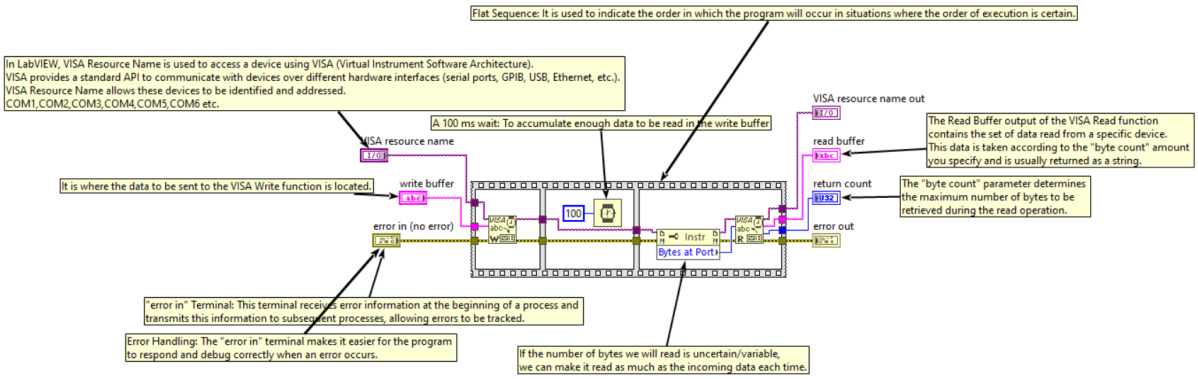


Different subvi's were used in this state. The first one is ReadDataSubVi

ReadDataSubVi-Front Panel



ReadDataSubVi-Block Diagram



- VISA Resource Name: This input specifies the device to communicate with, identified by its VISA resource name (e.g., COM1, COM2).

- Write Buffer: Before reading data, a write operation may be performed to send a command to the device. The "write buffer" holds the data to be sent.

- Error In: This terminal carries error information into the sub-VI, allowing error handling and propagation through the VI.

- Flat Sequence Structure: The flat sequence ensures that operations occur in a specific order. First, data is written to the device, followed by a 100 ms wait to ensure the device has time to process the command and generate a response.

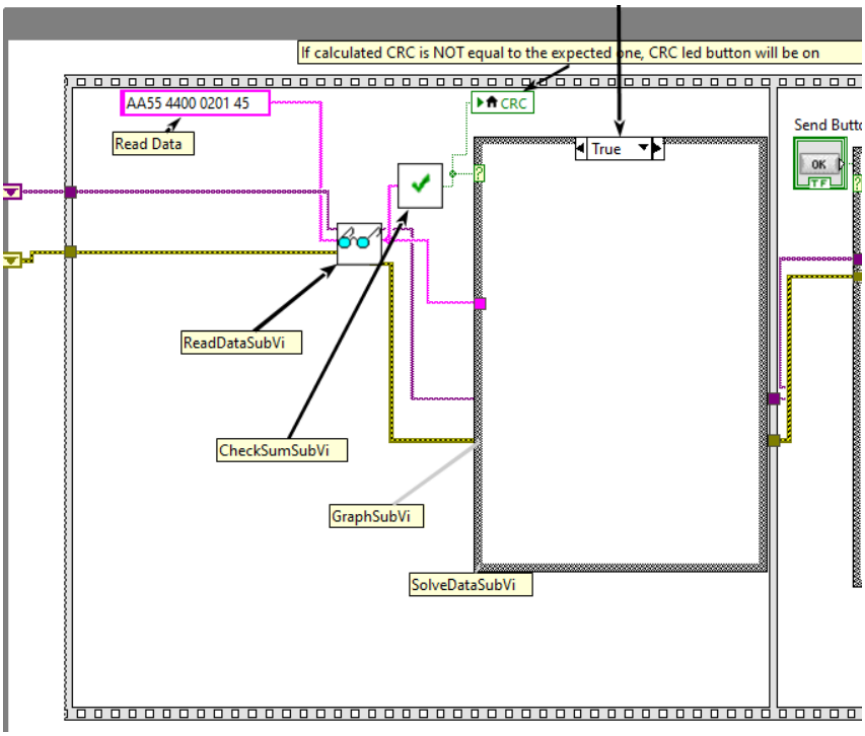
- VISA Write: This function sends the data in the write buffer to the device specified by the VISA resource name.

- Wait Function: The 100 ms wait accumulates enough time for the device to prepare the data to be read. This delay ensures that the subsequent read operation captures complete and accurate data.

- VISA Read: This function reads data from the device. The parameters include:
    - VISA Resource Name Out: Passes the resource name for continuity.
    - Read Buffer: The output of this function, containing the data read from the device.
    - Return Count: Indicates the number of bytes read.

- Error Out: Carries error information out of the sub-VI.
- Bytes at Port: This property node determines the number of bytes available to read. It ensures that the read operation retrieves the correct amount of data.

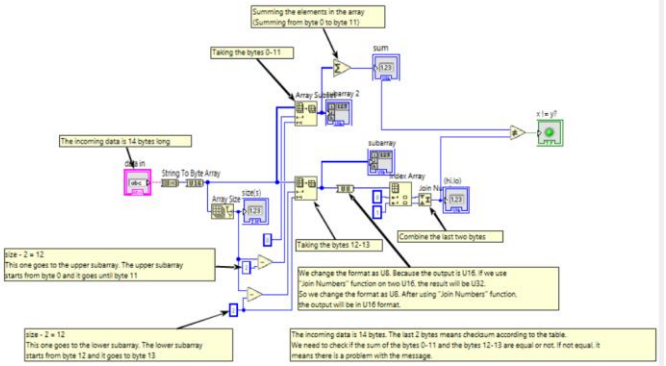
- Error Handling: The error in and error out terminals ensure any issues are properly managed, allowing for debugging and reliable operation.

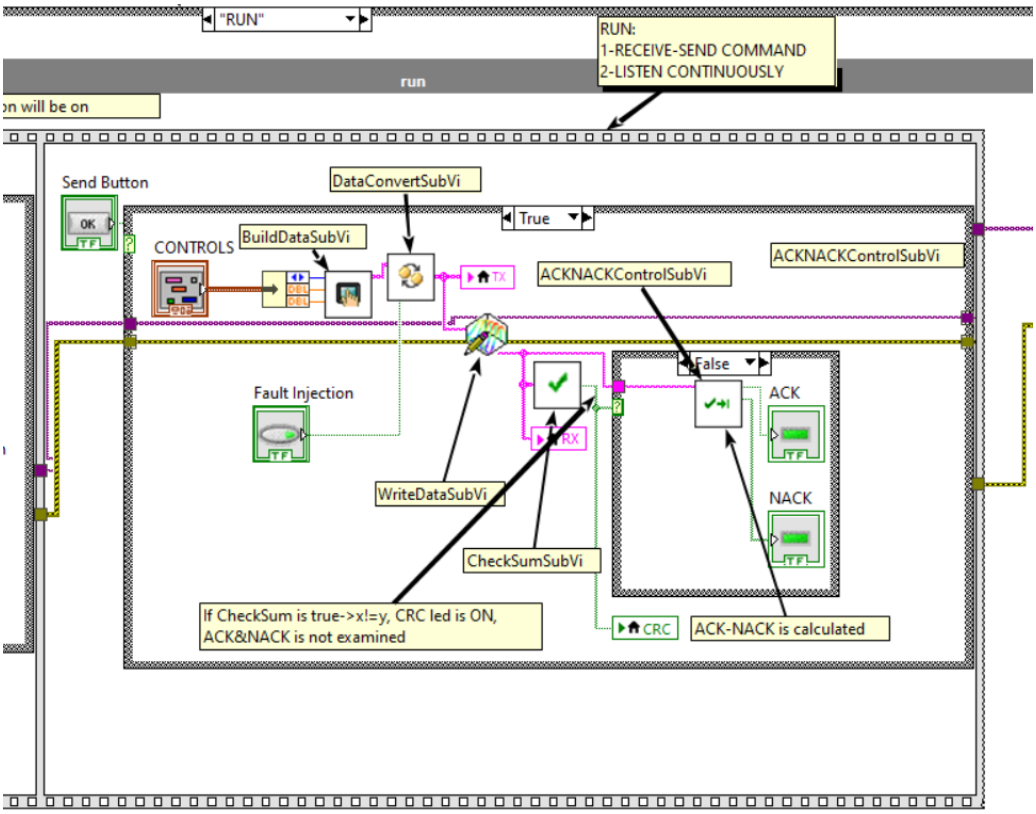


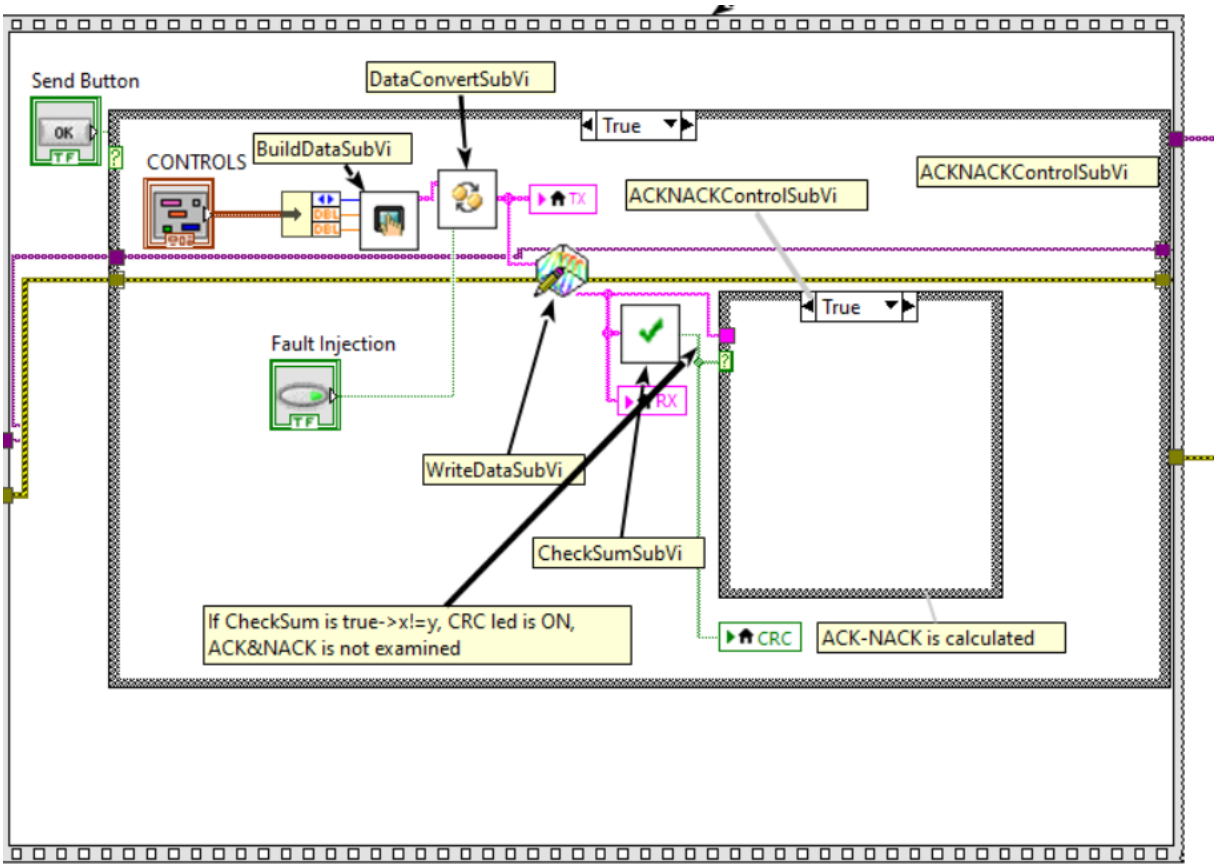
The string coming from the read buffer is taken into the check sum process via CheckSumSubVi

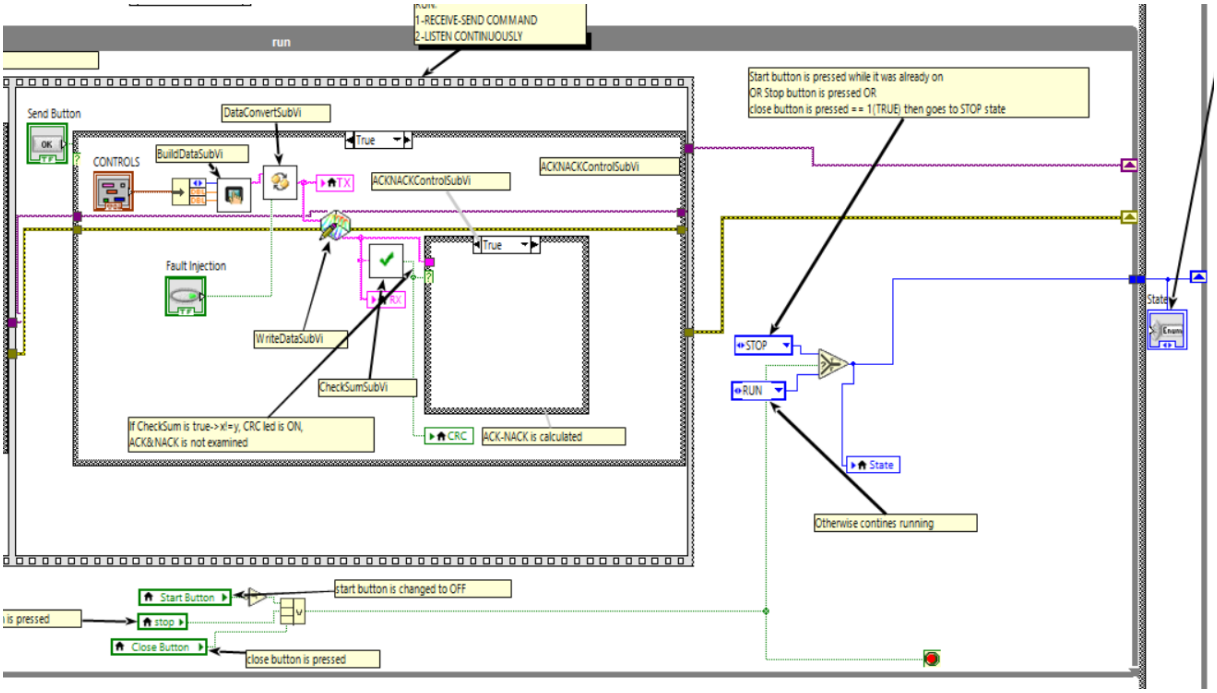
CheckSumSubVi


data in
x != y?
size(s)
0
(hi.lo)
0
subarray 2
0
sum
0000
subarray
0

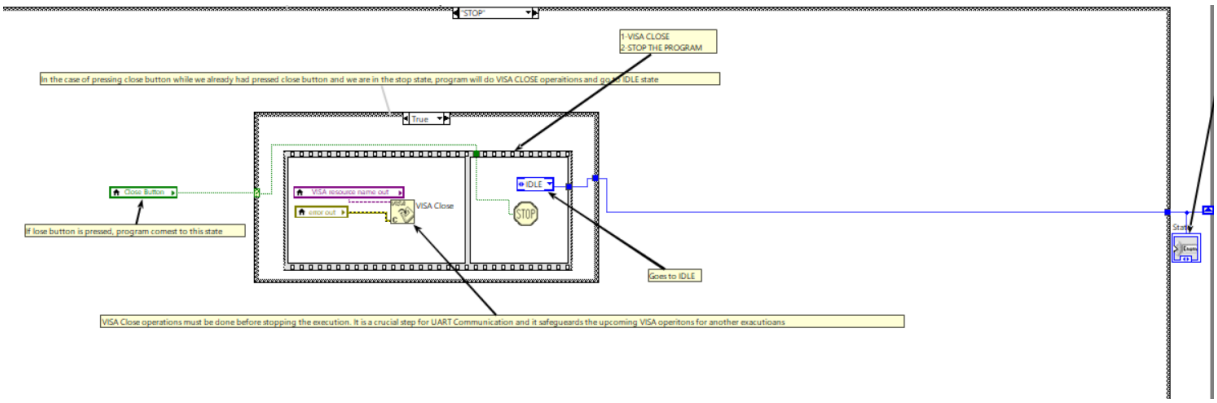

Summing the elements in the array (Summing from byte 0 to byte 11)
Taking the bytes 0-11
sum
Array Subset
subarray 2
x != y?
The incoming data is 14 bytes long
data in
String To Byte Array
size(s)
Array Size
subarray
Index Array
Join Numbers
(hi.lo)
Taking the bytes 12-13
Combine the last two bytes
size - 2 = 12
This one goes to the upper subarray. The upper subarray starts from byte 0 and it goes until byte 11
We change the format as U8. Because the output is U16. If we use "Join Numbers" function on two U16, the result will be U32. So we change the format as U8. After using "Join Numbers" function, the output will be in U16 format.
size - 2 = 12
This one goes to the lower subarray. The lower subarray starts from byte 12 and it goes to byte 13
The incoming data is 14 bytes. The last 2 bytes means checksum according to the table. We need to check if the sum of the bytes 0-11 and the bytes 12-13 are equal or not. If not equal, it means there is a problem with the message.


"RUN"
RUN:
1-RECEIVE-SEND COMMAND
2-LISTEN CONTINUOUSLY
run
on will be on
Send Button
OK
DataConvertSubVi
True
CONTROLS
BuildDataSubVi
TX
ACKNACKControlSubVi
ACKNACKControlSubVi
Fault Injection
False
ACK
RX
NACK
WriteDataSubVi
CheckSumSubVi
If CheckSum is true->x!=y, CRC led is ON, ACK&NACK is not examined
CRC
ACK-NACK is calculated





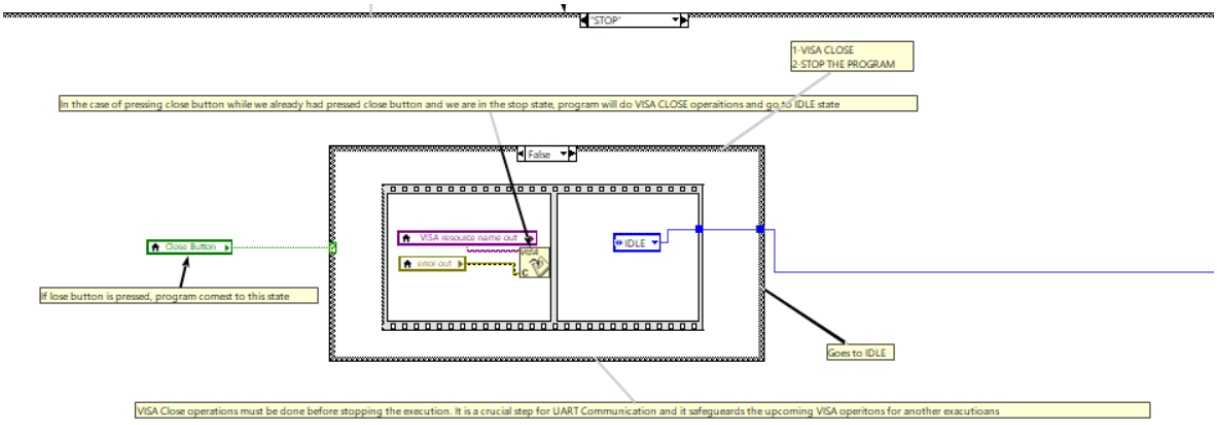
5.1.4.1. STOP(True)



In the case of pressing close button while we already had pressed close button and we are in the stop state, program will do VISA CLOSE operations and go to IDLE state.

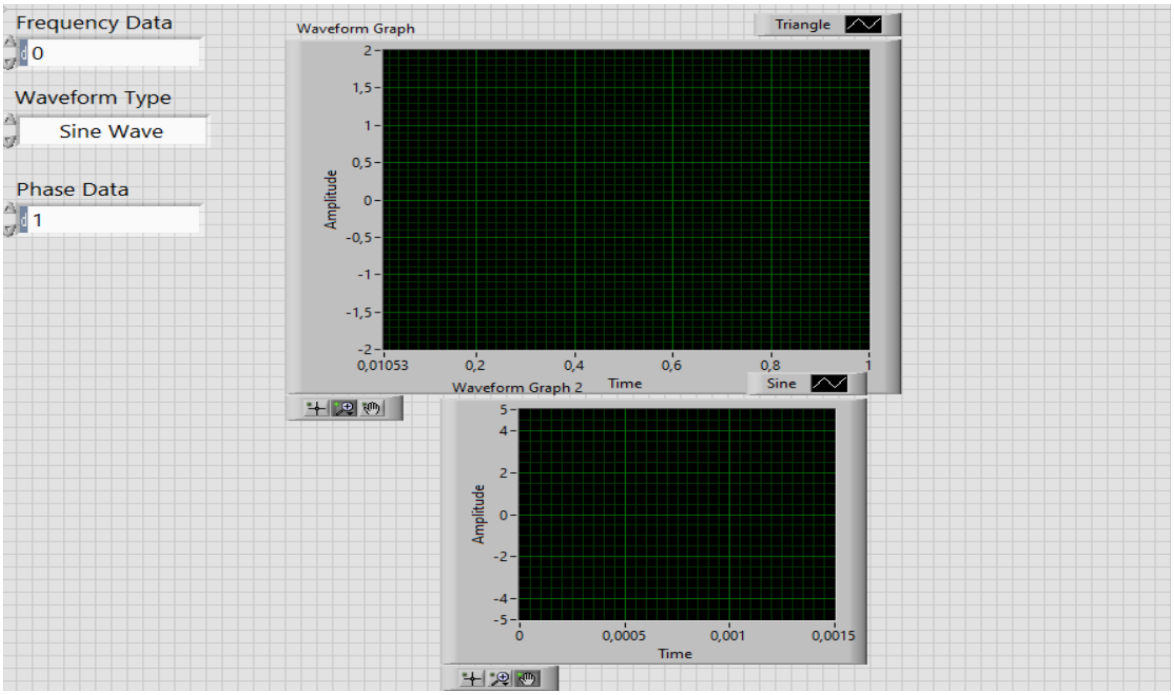
VISA Close operations must be done before stopping the execution. It is a crucial step for UART Communication and it safeguards the upcoming VISA operitons for another exacutioans

5.1.4.2. STOP(False)



VISA Close operation is done and goes to IDLE state without completely cut the execution. A stimulus may restart the operations again.

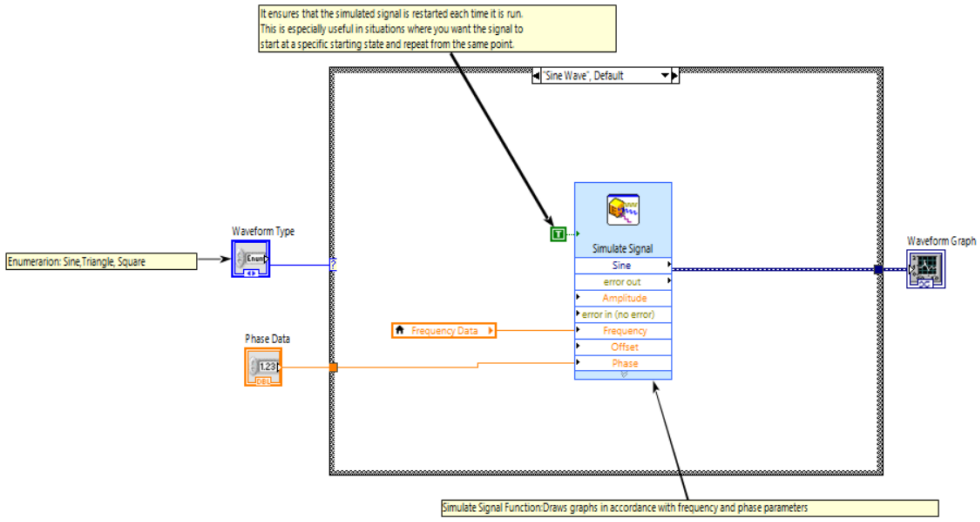
GraphSubVi-Front Panel



This subvi contains three inputs (frequency data, waveform type, phase data) and one output (waveform graph).

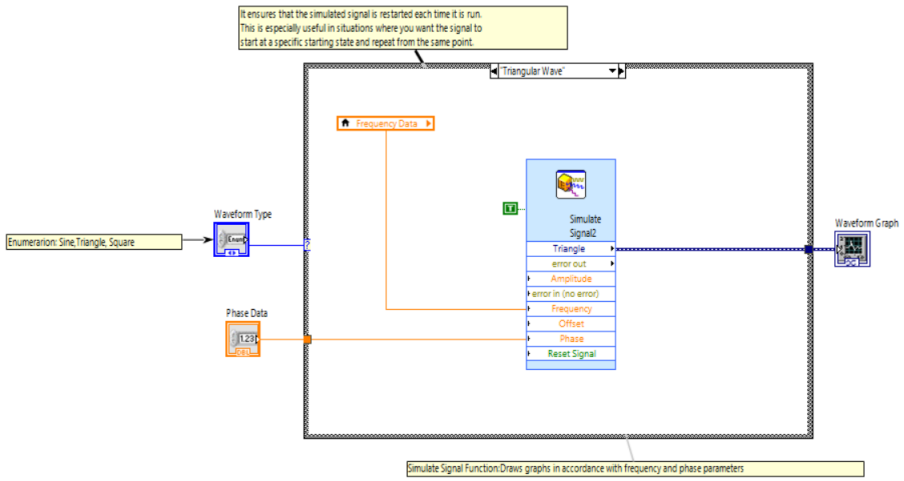
GraphSubVi-Block Diagram

a) **Sine Wave**

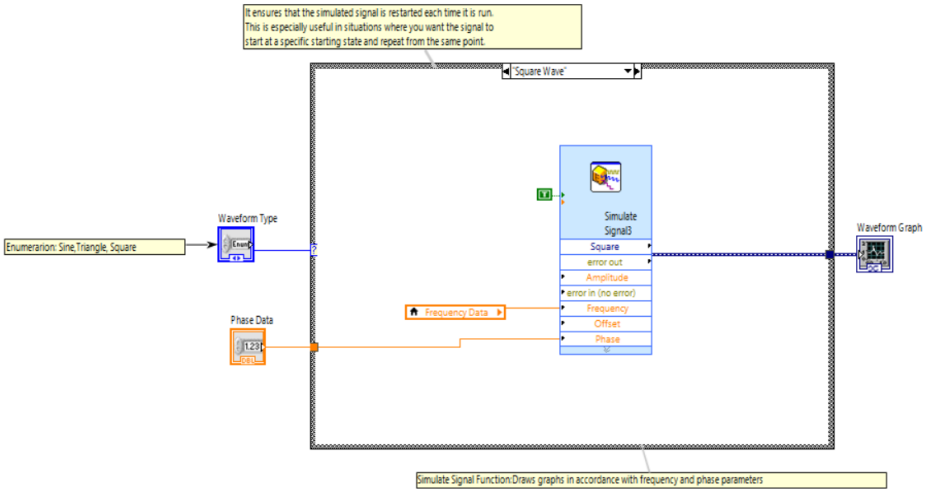


Simulate Signal closed function was used. Waveform type of enumeration, frequency data and phase data is given as input manually. A waveform graph is created in accordance with these inputs and shown on the screen to the user.

b) Triangle Wave



c) Square Wave


It ensures that the simulated signal is restarted each time it is run.
This is especially useful in situations where you want the signal to start at a specific starting state and repeat from the same point.
Square Wave
Waveform Type
Enumeration: Sine,Triangle, Square
Phase Data
Frequency Data
Simulate Signal3
Square
error out
Amplitude
error in (no error)
Frequency
Offset
Phase
Waveform Graph
Simulate Signal Function:Draws graphs in accordance with frequency and phase parameters