BLUE CONTROL

Course : EE536 : Internet of Things
Course Instructor : Dr. Padmanabhan Ranjan & Dr. Siddharth Sarma
Evaluated By : Dr.Srikanth Suvaganam & Dr. Siddharth Sarma

Team Members :
Sandeep N Kundalwal (T22051)
Mahima Gupta (T22055)

Project Description:

Create a network where a basic Control Unit can send commands to a Central Hub. The Central Hub will execute a series of commands on the attached instruments. SCPI commands will be sent from the Control Unit to the Central Hub via Bluetooth. Those SCPI Commands will be executed on respective instruments using USB Cables.

Here,
Control Unit : Raspberry Pi 3B+
Central Hub : Raspberry Pi 3B+

Github Repository : https://github.com/SKundawal/BlueControl

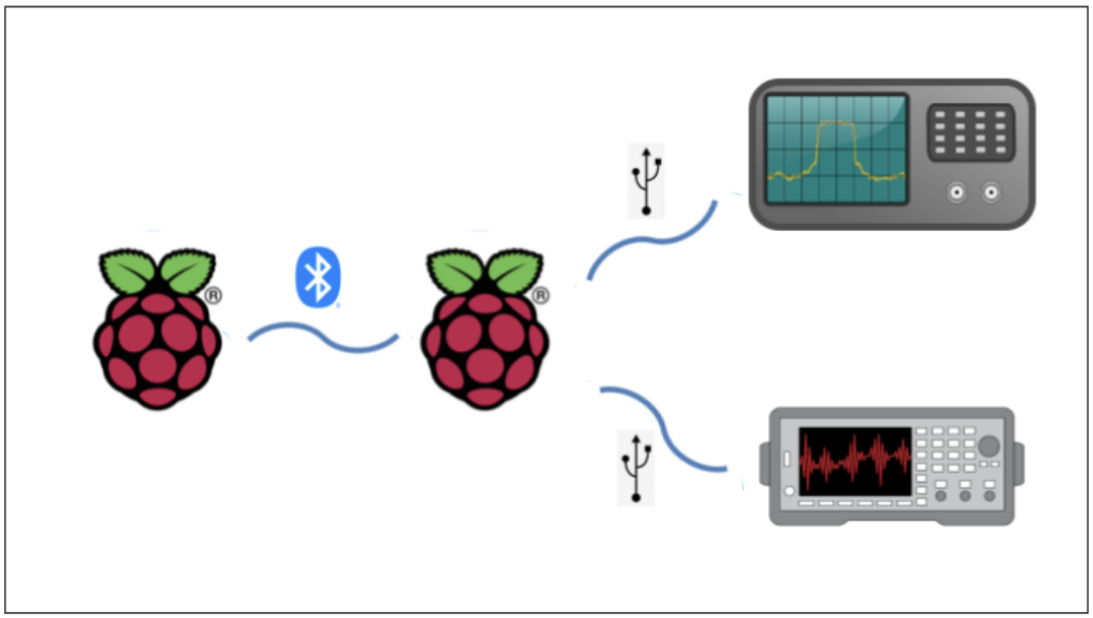


Figure 1 : Graphical Representation of the System

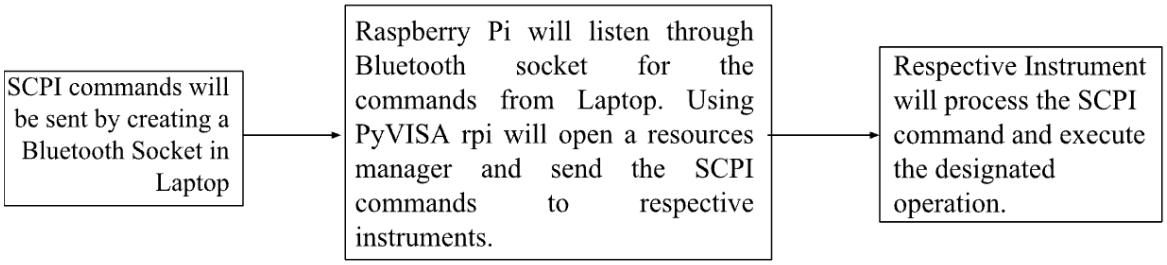


Figure 2: Data Flow Chart depicting the flow of scpi commands from laptop to devices.

Instruments Required:

- ➔ (Raspberry Pi 3 B+ & Power Adapter) ✗ 2
- ➔ DSOX1102A KeySight Technologies (Digital Storage Oscilloscope) & Power Cable
- ➔ AFG1062 Tektronix (Arbitrary Function Generator) & Power Cable
- ➔ (USB Cables) ✗ 2
- ➔ Laptop / PC (For remote access of Raspberry Pi)

Python Libraries:

- ➔ Library for sending SCPI commands to instruments from RaspberryPi - **PyVISA (https://pypi.org/project/PyVISA-py/)**
- ➔ Library for sending SCPI commands from laptop to RaspberryPi using Bluetooth - **PyBluez (https://pypi.org/project/PyBluez/)**
- ➔ Library for USB Communication between RaspberryPi and Instruments - **PySerial (https://pypi.org/project/pyserial/)**
- ➔ Library for discovery and service registration - **ZeroConf (https://pypi.org/project/zeroconf/)**

Setup :

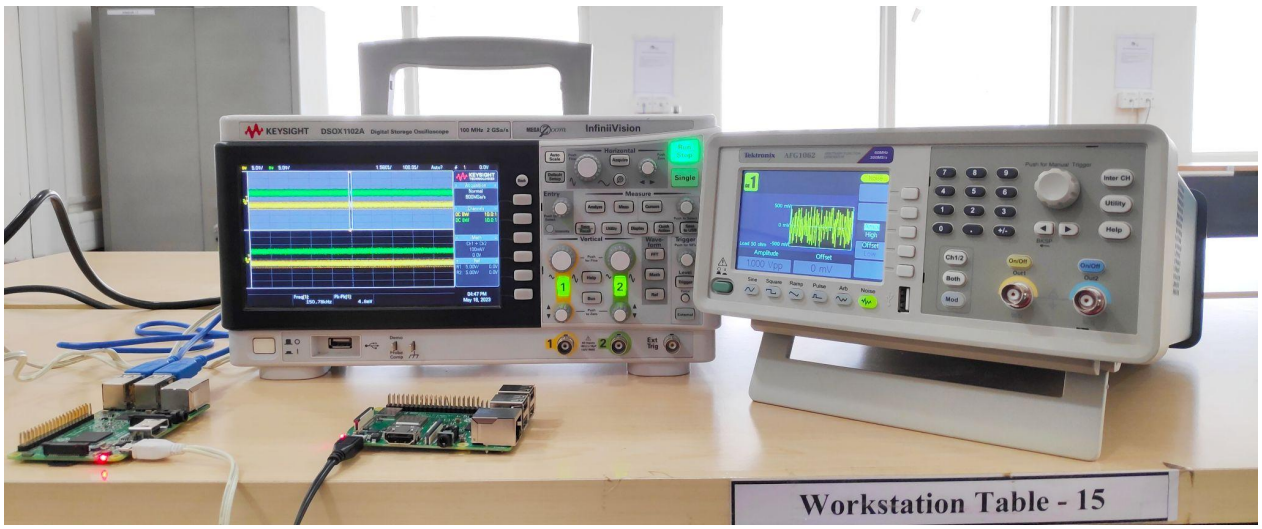


Figure 3 : Front view of the manual setup

Procedure:

Pair both Raspberry Pis using bluetooth from GUI/Terminal. Make the Control Hub discoverable using the following command :

```
sudo hciconfig hci0 piscan
```

Write Python Script to connect both Raspberry Pi units. We have implemented a **Client-Server Architecture for bluetooth connection.**

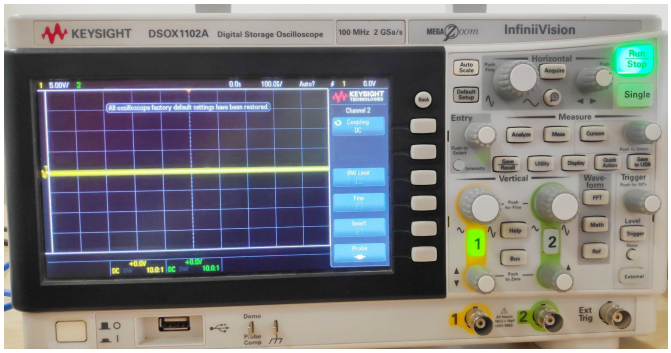
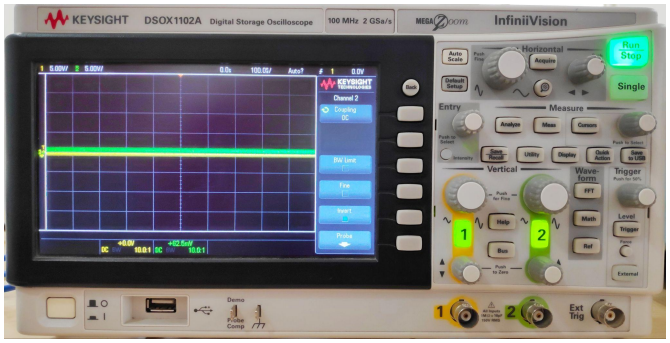
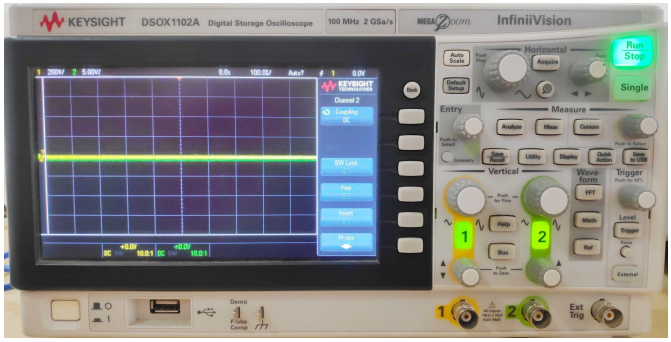
Control Unit → Client (Raspberry Pi 3B+)
IP Address : 192.168.43.113
Bluetooth Mac Address : B8:27:EB:85:6C:C3
Central Hub → Server (Raspberry Pi 3B+)
IP Address : 192.168.43.39
Bluetooth Mac Address : B8:27:EB:32:71:7B

Add Python Script for executing SCPI commands on instruments and send response for query commands back to the client. Capture the response of Set commands on the instrument. Add Logger for storing all info and error messages in a log file.

```
rpi@raspberrypi:~/Desktop/IoT-Mini-Project/scripts $ python3 client.py
Connected to server at: B8:27:EB:32:71:7B

Available Devices:
1.Device: AFG1062-TEKTRONIX, Address: USB0::1689::851::2140058::0::INSTR
2.Device: DSO-X 1102A-KEYSIGHT TECHNOLOGIES, Address: USB0::10893::6023::CN59014300::0::INSTR

In order to send a SCPI Command, write <Sr No.> <SCPI Command>
To QUIT, press 'Q' or 'q'

Command: 1 *RST; 2 *RST
Command: 1 SOUR1:FUNC:SHAP?; 1 SOUR1:FUNC:SHAP SQU; 1 SOUR1:FUNC:SHAP?
Response: SIN
Response: SQU
Command: 2 :CHAN2:DISP?; 2 :CHAN2:DISP ON; 2 :CHAN2:DISP?
Response: 0
Response: 1
Command: 1 SOUR1:FUNC:SHAP SIN; 2 :CHAN2:DISP OFF
Command: 1 SOUR1:FUNC:SHAP?; 2 :CHAN2:DISP?
Response: SIN
Response: 0
Command: q
Connection Closed.
```

Figure 4: Command line interface for sending commands at Client End.

SCPI Commands executed in *Digital Storage Oscilloscope* :

Sno.	SCPI Command	Description	Response
1.	*RST	Places the oscilloscope in a known state. This command loads the factory default setup.	
2.	:CHANnel2:INVert ON	Turns a channel's input signal inversion ON/OFF.	
3.	:CHANnel2:DISPlay ON	Turns the display of the specified channel ON/OFF.	
4.	:CHANnel2:INVert?	Returns the current state of the channel inversion.	Command: 2 :CHANnel2:INVert? Response: 1
5.	:CHANnel2:DISPlay?	Returns the current display condition for the specified channel.	Command: 2 :CHAN2:DISP? Response: 0

SCPI Commands executed in *Arbitrary Function Generator* :

Sno.	SCPI Command	Description	Response
1.	*RST	Resets the instrument to the factory default setting.	
2.	SOUR1:FUNCtion: SHAPe PRN	Sets the shape of the output waveform.	
3.	SOURce1:FM:STATe ON	Enables/ Disables FM Modulation.	
4.	SOURce1:FUNCtion:SHAPe?	Queries the shape of the output waveform.	Command: 1 SOURce1:FUNCtion:SHAPe? Response: PRN
5.	SOURce1:FM:STATe?	Queries whether the current FM Modulation is enabled/disabled.	Command: 1 SOURce1:FM:STATe? Response: 1