Interim Report For Final Project Of ECE540

For Rutuja Muttha

We have created a shared repository on GitHub and have planned to push all the work on that as per we complete our part.

About my work till now:

1. For connection between ESP32 and Android App:

As per commitment given in the proposal I have figure out connection between Android app and ESP32. I am using Bluetooth wireless connection for the communication between app and the microcontroller which is nothing but using the Bluetooth Low Energy(BLE).
Using BLE is a good option because it is low-power, easy to use, and widely supported on both Android devices and the ESP32.
I have done with the implementation of connections for ESP32 on Arduino IDE and have done the git push for the implementation. For implementing connections of ESP32 I have setup BLE communication configuration.
I have configured the connections of BLE in the android app using libraries and the connections will be established by same SERVICE_UUID and CHARACTERISTIC_UUID which will advertise the connections of ESP32 to the Android app and have done with git push.

2. For connection between ESP32 and RVFPGA:

For this part, I was thinking of using the SPI protocol. I found, this is the easiest way of communication but I am still working on it.
In this, I have figured out following steps to establish this connection.
Step1: Choose the SPI pins on the ESP32 and RVFPGA: The ESP32 and RVFPGA have dedicated pins for SPI communication. Choosing the SPI pins on both the ESP32 and RVFPGA that you will use for communication.
Step2: Configure the ESP32 SPI as the master: The ESP32 will act as the master in the SPI communication protocol. Configuring the ESP32 SPI as the master and set the clock speed, data mode, and bit order.
Step3: Configure the RVFPGA SPI as the slave: The RVFPGA will act as the slave in the SPI communication protocol. Configuring the RVFPGA SPI as the slave and set the clock speed, data mode, and bit order.
Step4: Writing SPI communication code on ESP32: Need to write the SPI communication code on the ESP32 that will send commands to RVFPGA and receive game updates. The ESP32 will send commands to RVFPGA such as move left, move right, fire, etc. The ESP32 will also receive game updates from RVFPGA such as the position of the player, the position of the aliens, and the score.
Step5: Write SPI communication code on RVFPGA: Need to write the SPI communication code on the RVFPGA that will receive commands from the ESP32 and send game updates. The RVFPGA will receive commands from ESP32 such as move left, move right, fire, etc. The RVFPGA will also send game updates to the ESP32 such as the position of the player, the position of the aliens, and the score.

As per above steps, I have done with the coding part requirement. I have done the git push for the above part.

3. Regarding the app layout:

I have come up with the rough layout for the android app but I want to modify it more and implement it on the .xml file to see the final look of the app. I am working on it and will be get done till next deadline.

4. Problems faced while doing the above part of the project:

There were different ways for communication, and I have to figure out which one is the easiest and fastest way for the communication as the communication is the main part of the entire project. There were different ways for the communication between RVFPGA, Android app and ESP32 like WiFi Direct, Bluetooth, Ethernet connection, Zigbee, USB, etc.