CS/ECE M119 (Fall 2021)

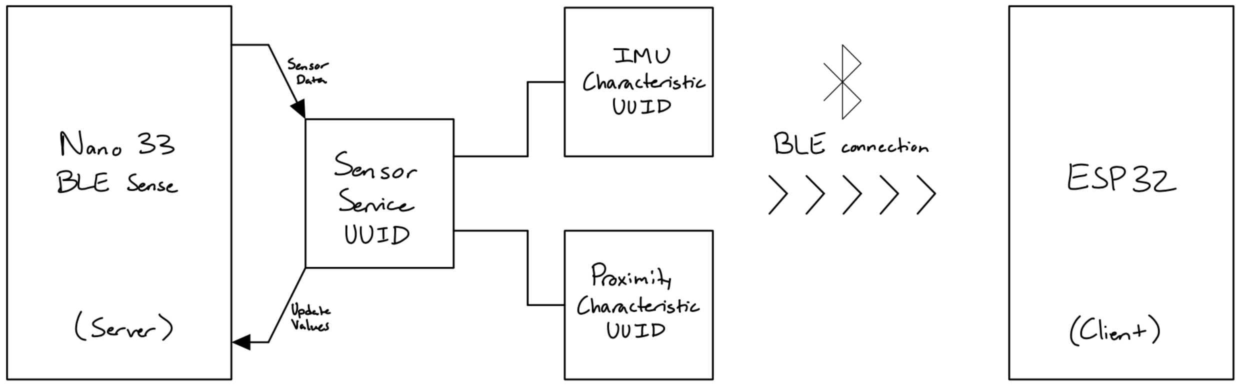
Myles Johnson

205868607

Assignment #2

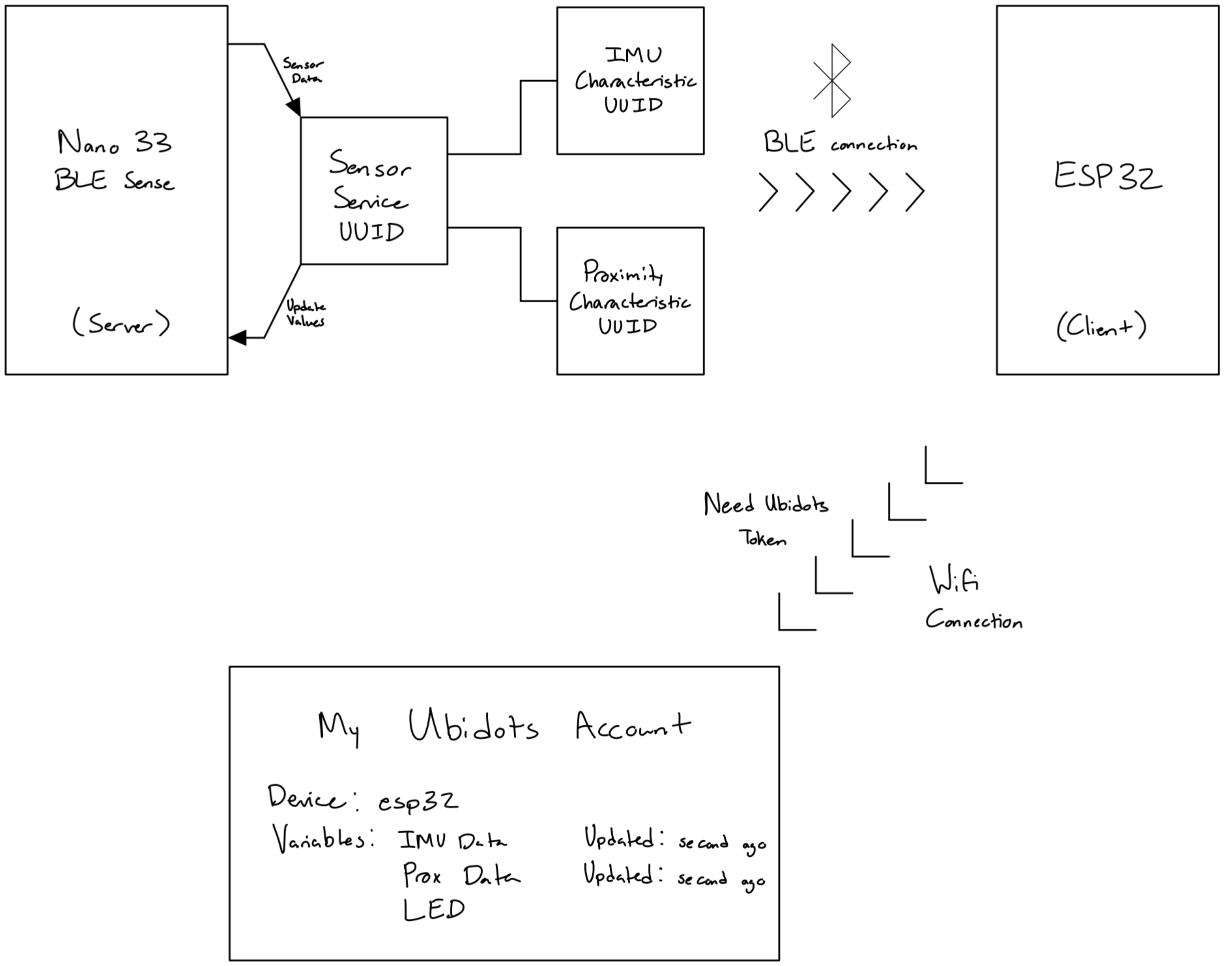
Problem 1: BLE Connection

The BLE Connection between the arduino and esp32 works as such, first the arduino defines a service UUID, SensorService, and 2 characteristic UUIDs, IMUChar and ProxChar, this is how the sensor data is carried over the BLE connection via services that can have a group of capabilities called characteristics. Next, I set up the arduino to write the sensor data of the LSM9DS1 (IMU) and APDS9960 (Proximity Sensor) every second to its respective characteristics. Then I set up the esp32 to connect to the arduino via BLE and search for the SensorService UUID, once that is found the characteristic UUIDs are searched for and found. Lastly the esp32 can now read the values stored in the characteristics live as they are tracked from the arduino sensors.



Problem 2: From Sensors to the Cloud

Now that the devices are connected via BLE Connection the next step of uploading the value to the cloud is quite simple.As I explained before the service UUID, SensorService, and 2 characteristic UUIDs, IMUChar and ProxChar, carry the sensor data of the IMU and Proximity Sensor which the esp32 can now read from the arduino. To get this info to the cloud I have to connect my Ubidots account token and also connect the esp32 to the internet. Once those are connected I can add the same values that the esp32 is reading from the arduino to the cloud and give it a variable name. In my case I used the variable names, Imu Data and Prox Data, then I publish those variables under a device name, "esp32" in my case, and lastly publish that info to my ubidots account. Once that is all completed I can now login online to ubidots and navigate to the "esp32" device and see the two variables, Imu Data and Prox Data, being live updated as they are tracked from the arduino sensors.



Bonus question

It is possible to control the “built_in_LED” on your ESP32 through MQTT protocol. The first step is to create a new variable under the esp32 device tab, I labeled mine “led”. Once that was done I created a new dashboard and added an ON/OFF switch widget. Once the widget was created I assigned it the “led” variable from my esp32 device. Now in my esp32 code I used my same Ubidot token to access my account and enter the wifi info to connect to the internet. I kept my device label as “esp32” and made the variable label “led” to match the one I had created. I then assigned pin 2, the esp32s built in led pin, as an output pin and I used the ubidot subscribe function to get the variable label value from my uboidots account. The switch I created assigns a 1 to the “led” value when on and a “0” when switched off, so in the code I used an if statement to assign the led to digitalwrite “HIGH” when the callback from the ubidots variable was assigned “1” and to digitalwrite the led to “LOW” when a “0” was assigned.