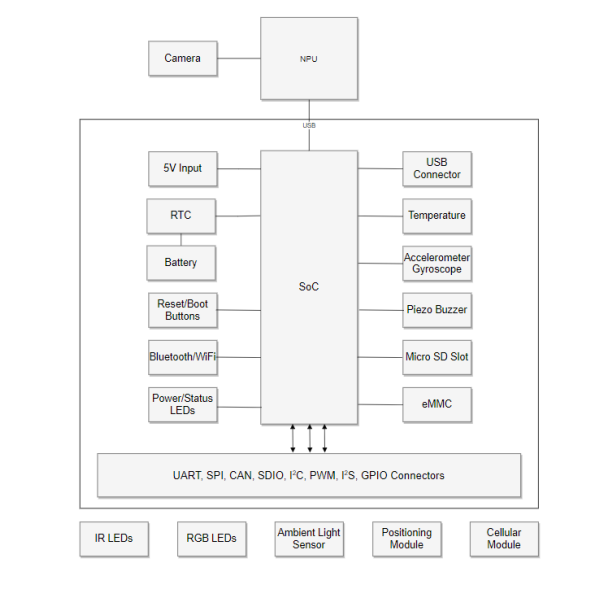
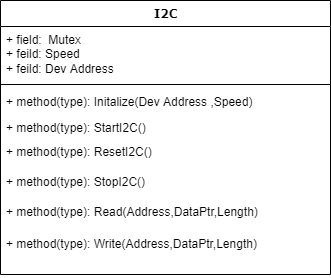
**Architecture for IoT project.**

**Reference block diagram:**

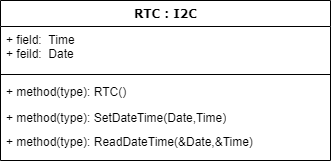


**Software drivers need to be implemented:**

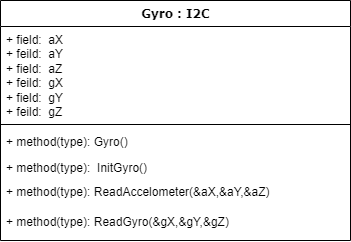
1. RTC (I2C)
2. Temperature sensor (I2C)
3. Accelerometerter/Gyroscope module (I2C)
4. Ambient light sensors (I2C)
5. Positioning modules – GPS (I2C)
6. Wifi module (UART)
7. GSM module (Cellular) – (UART).
8. Buzzer (GPIO)
9. Micro SD card (SPI)
10. IR LEDs (PWM)
11. RGB LEDS (PWM)
12. USB
13. I2S
14. CAN
15. SDIO
16. Parent Class: I2C



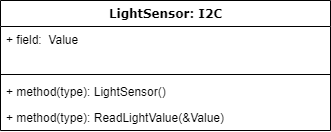
* All functions are guarded by mutex locks.
  1. Child class: RTC



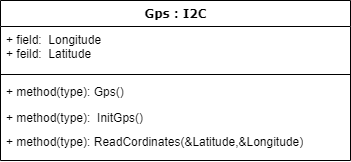
* 1. Child class : Gyro



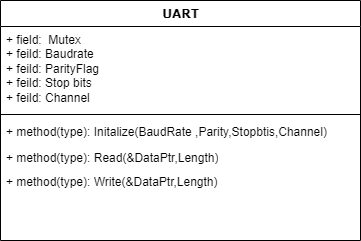
* 1. Child class : LightSensor



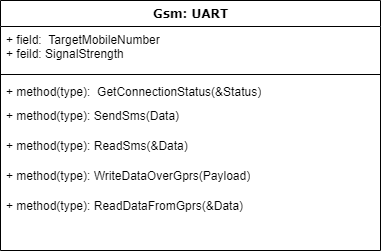
* 1. Child class : GPS



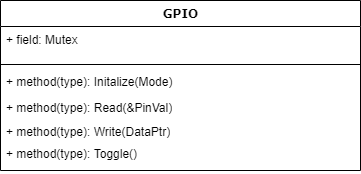
1. Parent Class: UART



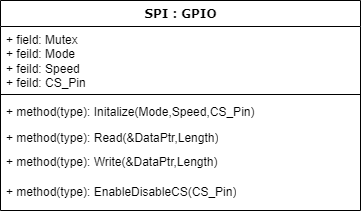
1. Child class : Gsm



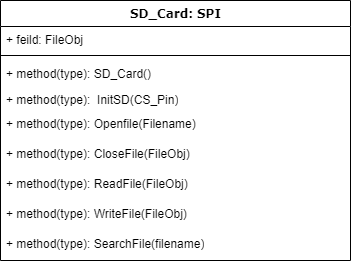
1. Parent class: GPIO



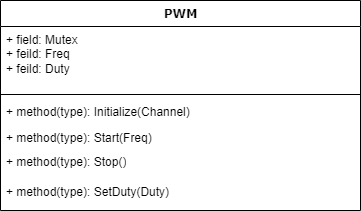
1. Parent class: SPI



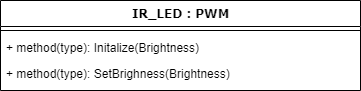
1. Child class : SD\_Card



1. Parent class: PWM



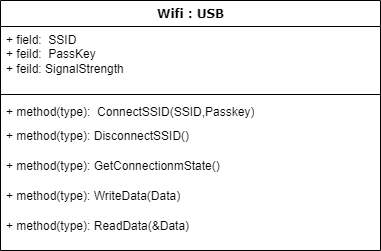
1. Child class : IR\_LED



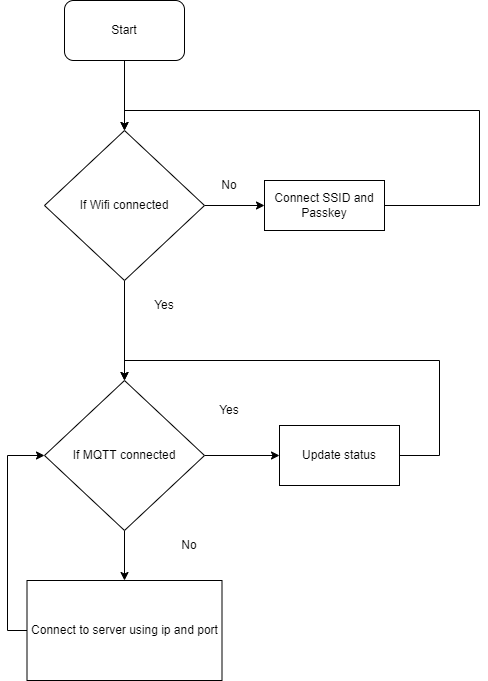
1. Child class : RGB\_LED



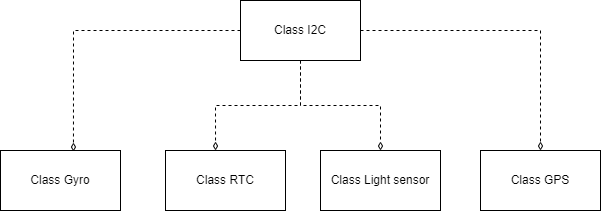
1. Class Wifi

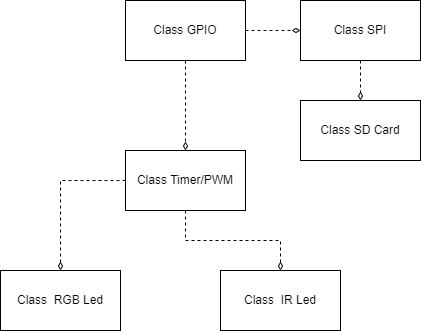


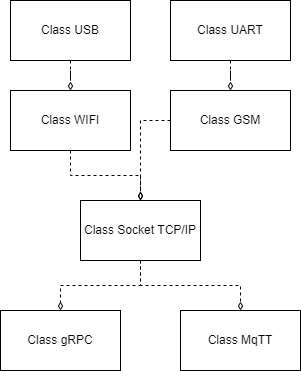
**Handler for MQTT connection**



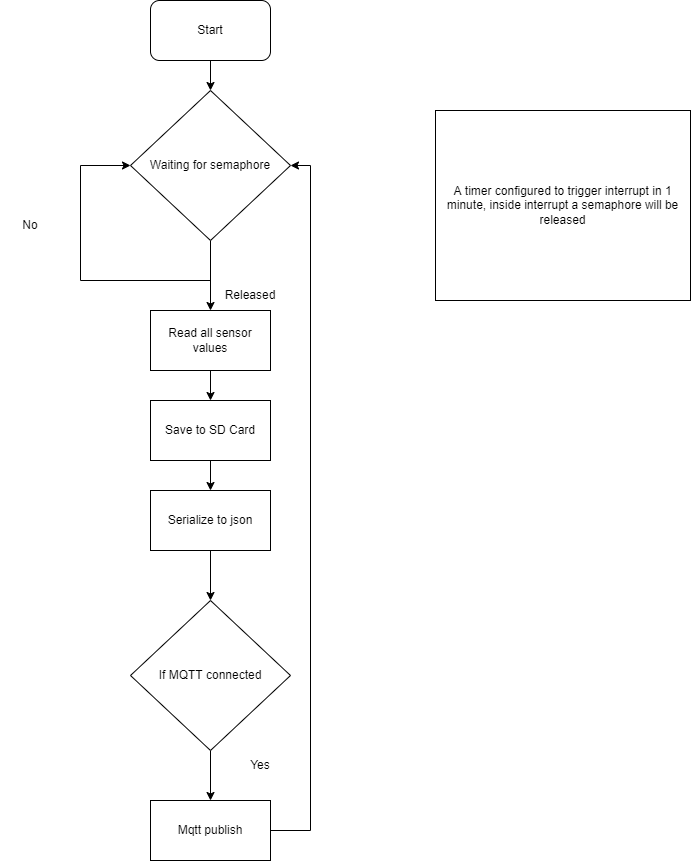
**Relationship between classes**

****

****

****

**Fetching sensor parameters, Saving and Sending data to a cloud**

****

**Communication between application to Mobile APP.**

For ease of communication, we are using Google remote procedure calls (GRPC) over TCP IP.

Using gRPC a client app can directly call a method in the server app. Hence all parameters can be read from server directly.

