IMU_Module

Module Responsibility :

- Read raw sensor data from the **MPU6050** using **I2C**
- Parse and store **accelerometer**, **gyroscope**, and **temperature** raw values

Output :

- Provide the accelerometer and gyroscope data
- Provide the **temperature** data

Module Design

<<block>>
MPU_Sensor

<<Parameters>>

```
typedef struct {
    int16_t accel_x;
    int16_t accel_y;
    int16_t accel_z;
    int16_t gyro_x;
    int16_t gyro_y;
    int16_t gyro_z;
    int16_t temperature
    } Imu_DataType;
```

<<Function>>

~Imu_Init(void) : void

~Imu_Main(Imu_DataType *data) : void

~out: Imu_ReadAccel(int16_t *ax, int16_t *ay, int16_t *az) : int

~out: Imu_ReadGyro(int16_t *gx, int16_t *gy, int16_t *gz) : int

~out: Imu_ReadTemp(float *temperature) : int

Module Flowchart

```
                    ┌─────────┐
                    │  Start  │
                    └────┬────┘
                         │
                    ┌────▼────┐
                    │ IMU_Init│
                    └────┬────┘
                         │
              ┌──────────▼──────────┐
        Yes   │  WHO_AM_I == 0x68 ? │   No
      ◄───────┤                     ├────────►
              └─────────────────────┘
```

- Start
- IMU_Init
- WHO_AM_I == 0x68 ?   (Yes / No)
- Write PWR_MGMT_1 (Wake Up IMU)
- I2C WriteRead (Read 14 Bytes from MPU)
- Parse Accelerometer Data
- Parse Temperature Data
- Parse Gyroscope Data
- Store Values in Imu_DataType Structure
- End