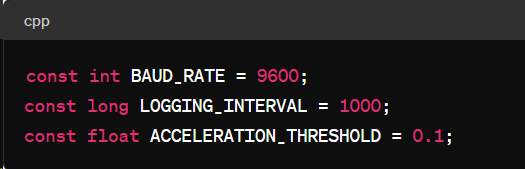
**Embedded System Code Enhancements Report**

**Overview**

This report details the recent enhancements made to the Arduino sketch deployed on the Arduino IoT 33 board, which is based on the SAMD21 microcontroller. The code modifications focus on improving the system's reliability, maintainability, and functionality. Each change is outlined below with its purpose and implementation strategy.

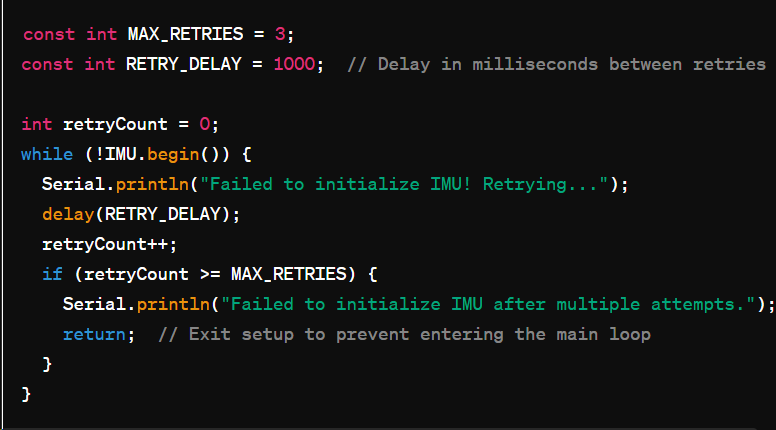
**Enhancements Summary**

1. **Defining Magic Numbers as Constants**
   * **Issue**: The use of magic numbers (e.g., 9600, 1000, 0.1) directly in the code affected readability and maintainability.
   * **Resolution**:
     + Defined these numbers as constants at the top of the code to enhance clarity and facilitate easier modifications in the future.
     + **Constants Introduced**:
       - **BAUD\_RATE** for **9600**, used in **Serial.begin()**.
       - **LOGGING\_INTERVAL** for **1000**, dictates the data logging interval.
       - **ACCELERATION\_THRESHOLD** for **0.1**, determines movement direction based on acceleration.
     + **Code Snippet**



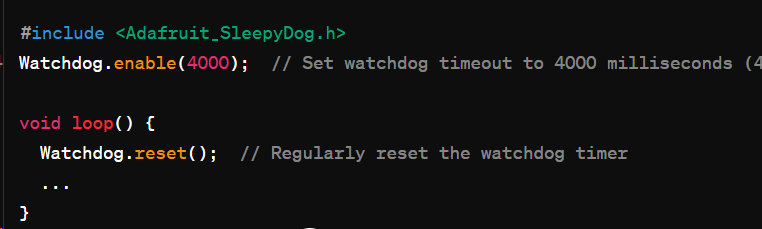
**Implementing Retry Mechanism for IMU Initialization**

* **Issue**: Previous code entered an infinite loop if the IMU failed to initialize.
* **Resolution**:
  + Introduced a retry mechanism that attempts to initialize the IMU multiple times (**MAX\_RETRIES**) before giving up, allowing for recovery or safe shutdown if the hardware isn't responding.
  + **Constants for Retry Mechanism**:
    - **MAX\_RETRIES = 3**
    - **RETRY\_DELAY = 1000** milliseconds
  + **Code Snippet**



**Watchdog Timer Implementation**

* **Issue**: The need for a mechanism to reset the microcontroller in case the program hangs or crashes.
* **Resolution**:
  + Implemented a watchdog timer using the Adafruit\_SleepyDog library, suitable for the SAMD21 microcontroller.
  + Configured to reset the board if the program fails to reset the timer within a specified timeout (4 seconds).
  + **Code Snippet**



**Additional Considerations**

* **Refactoring and Readability**: By defining constants, the code becomes easier to read and modify. For example, adjusting the baud rate now involves updating a single line.
* **Robustness and Reliability**: The retry mechanism and watchdog timer enhance the robustness of the code. They handle temporary failures and prevent the system from remaining in an indefinite hang state.

**Testing and Verification**

* Extensive testing should be conducted to ensure the code behaves as expected under various conditions, including scenarios where the IMU might repeatedly fail to initialize.
* Simulate conditions that might cause the program to hang to verify that the watchdog timer correctly resets the system.

**Conclusion**

These enhancements make the embedded system code more maintainable, understandable, and robust, significantly improving the reliability of deployments on the Arduino IoT 33 board. Future updates should continue to focus on enhancing system stability and operational efficiency.

GitHub link for code: <https://github.com/Bhumika-Chauhan09/redback-orion/blob/main/Acclerometer_secure_code.ino>

### Security Features Implemented:

1. **Input Validation**: The code checks if the raw accelerometer data exceeds a predefined safety threshold before it is processed. This prevents the handling of anomalously high readings that could be indicative of sensor errors or tampering.
2. **Error Handling**: There is robust error handling during sensor initialization. If the sensor fails to initialize, the system will halt, preventing further operations that rely on sensor data. Also, the system checks for sensor availability before attempting to read data, thereby avoiding potential null data errors.
3. **Data Logging Intervals**: The use of a fixed interval for data reading (1 second as defined by **INTERVAL**) ensures that the system does not attempt to process data too frequently, which could overwhelm the processing capabilities or lead to denial of service.

These enhancements aim to protect the Arduino system from potential vulnerabilities related to input data handling and ensure the reliability and integrity of the system's operation.

Github link for the code: <https://github.com/Bhumika-Chauhan09/redback-orion/blob/main/kalman_filter2.ino>