ChatGPT Prompts:

# Prompt:

I'm working on a project in the arduino ide 2.0, using the esp32 dev module board. Using MQTT Pubsubclient I'm publishing sensor data from the mpu6050 in every void loop(). Now I'd like to publish the temperature on a topic called "temp/K05", but only every second. How can I make sure, that the temperature topic is published only every second, whilst retaining the constant publishing of sensor data of the gyro and accelerometer?

# Answer:

unsigned long lastTempPublishTime = 0;

const unsigned long tempPublishInterval = 1000; // 1000 milliseconds = 1 second

unsigned long currentMillis = millis();

// Publish temperature only once every second

if (currentMillis - lastTempPublishTime >= tempPublishInterval) {

// Code to read the temperature from the sensor and publish it

float temperature = readTemperature(); // Replace with your temperature reading logic

// Convert the temperature value to a string

char temperatureStr[10];

sprintf(temperatureStr, "%.2f", temperature);

// Publish the temperature value to the topic "temp/K05"

client.publish("temp/K05", temperatureStr);

// Update the last publish time

lastTempPublishTime = currentMillis;

}

// Code to read and publish sensor data from the MPU6050

// ...

// Remaining code in the loop

// ...

# Prompt:

are you sure that this code ascertains that the temperature is published EXACTLY every seond?

# Answer:

You're correct, the code I provided doesn't guarantee that the temperature will be published exactly every second. The use of millis() and the comparison currentMillis - lastTempPublishTime >= tempPublishInterval ensures that the temperature is published approximately every second, but it may not be exact due to other processing delays in the void loop() or other parts of the code.

If you require precise timing, especially in scenarios where the void loop() function may have varying execution times, you can use a Timer library to achieve more accurate timing. The library allows you to schedule tasks to run at specific intervals.

Here's an example of how you can use the SimpleTimer library to publish the temperature exactly every second:

Install the SimpleTimer library in the Arduino IDE (Sketch -> Include Library -> Manage Libraries -> Search for "SimpleTimer" -> Install).

Include the library in your code:

cppCopy code

#include <SimpleTimer.h>

Define a timer object:

cppCopy code

SimpleTimer timer;

In the void setup() function, initialize your MQTT client and connect it to the broker.

Set up a timer to execute the temperature publishing function every second:

cppCopy code

void publishTemperature() { // Code to read the temperature from the sensor and publish it float temperature = readTemperature(); // Replace with your temperature reading logic // Convert the temperature value to a string char temperatureStr[10]; sprintf(temperatureStr, "%.2f", temperature); // Publish the temperature value to the topic "temp/K05" client.publish("temp/K05", temperatureStr); } void setup() { // ... // Set up the timer to execute the publishTemperature function every 1000 milliseconds (1 second) timer.setInterval(1000, publishTemperature); // ... }

In the void loop() function, add the following line to allow the timer to run:

cppCopy code

void loop() { // ... timer.run(); // ... }

By utilizing the SimpleTimer library and setting the timer interval to 1000 milliseconds, the publishTemperature() function will be called approximately every second with better accuracy compared to relying on the void loop() execution time alone.

Please note that the exact timing also depends on the overall code complexity and the resources consumed by other tasks running on the ESP32.