

船体動揺監視 IoT システム動作マニュアル

作成日:2024 年 5 月 1 日

公開日: 2024 年 5 月 20 日

バージョン:0.0.0(仮)

作成者:東京海洋大学大学院 海運ロジスティクス専攻 小森健史

目次

1. 概要.....	3
1.1. 本システムの目的.....	3
1.2. 本システムの構成図と主な機能.....	3
1.3. 本システムの特徴.....	5
1.4. 本システムに必要な物品.....	6
2. 使用方法.....	7
2.1. 船体動揺監視 IoT システム使用前の事前準備	7
2.2. 船体動揺監視 IoT システム動作時の使い方	15
2.3. マイコン端末内部の RTC の時刻合わせ	18
2.4. 使用上の注意.....	20
引用文献	21

1. 概要

1.1. 本システムの目的

本システムは動揺周期と GM 推定値を操船者がリアルタイムに確認できるようにすることで、復原力不足による転覆や激しい動揺による貨物損害事故を未然に防ぐことを目的とする。

1.2. 本システムの構成図と主な機能

本システムの構成図を図 1 に示す。このシステムではマイコン端末によって船体動揺を計測し、動揺周期・GM 推定値などを船内ネットワークもしくはクラウド経由で船員や陸上の運営管理者に共有するようになっている。

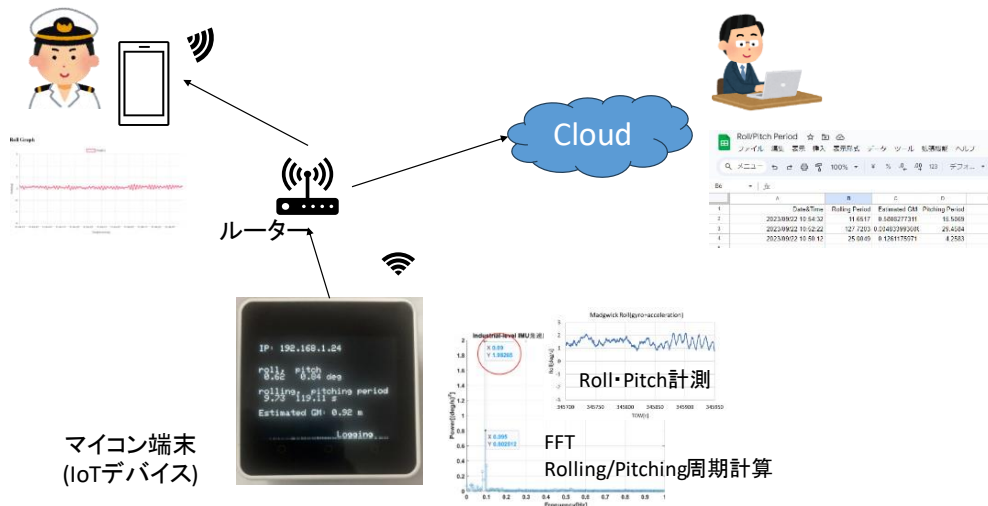


図 1 船体動揺監視 IoT システムの構成

船体動揺監視装置(マイコン端末)のフローチャートを図 2 に示す。船体傾斜(Roll,Pitch)の計測、動揺周期・GM 推定値の計算、クラウド上へのアップロード、WebPage 上での船体傾斜、動揺周期、GM 推定値のグラフ・表での表示は全てマイコン端末内部で行われる。

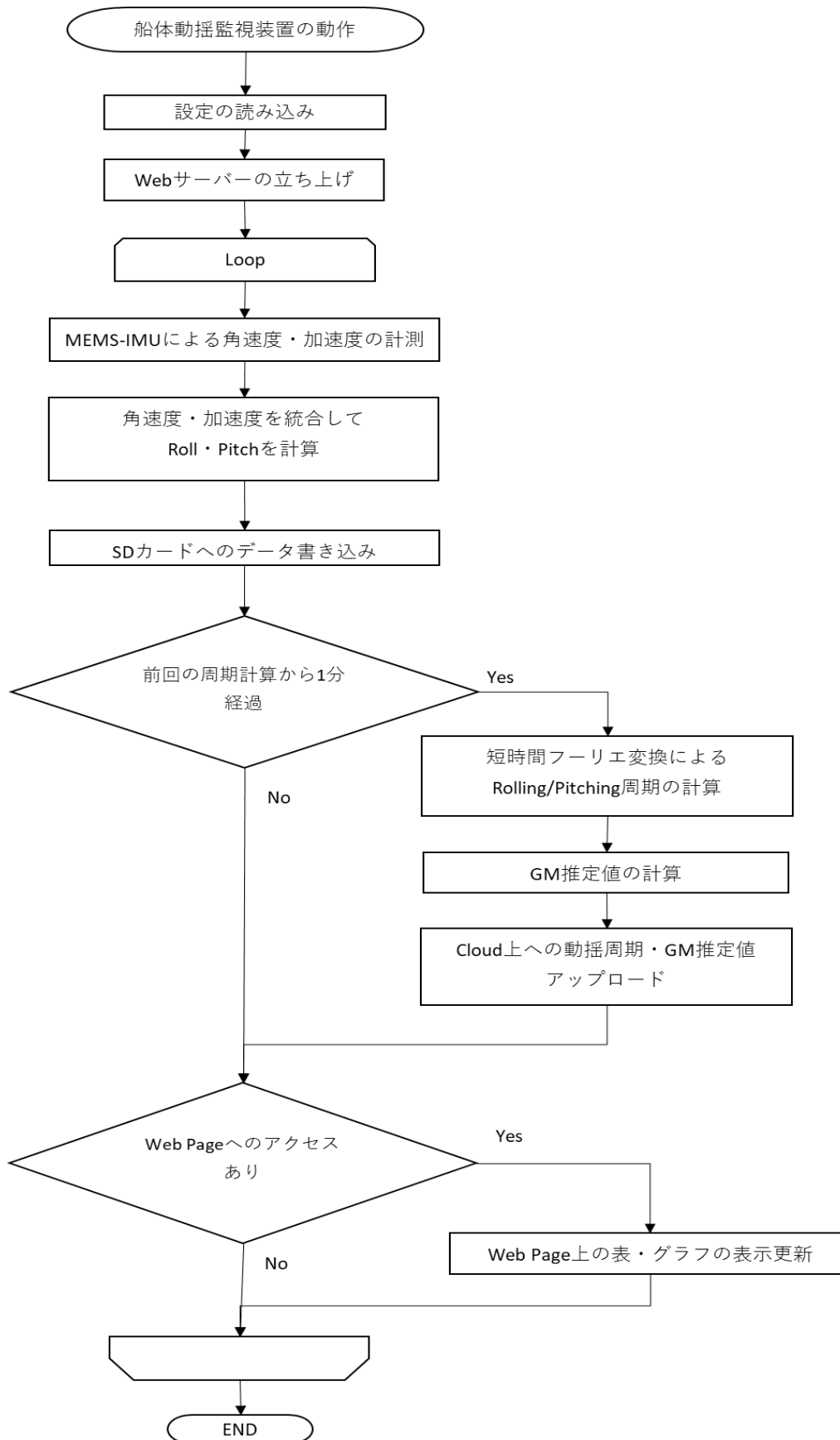


図 2 船体動揺監視装置のフローチャート

1.3 本システムの特徴

本システムの特徴を以下に列挙する。

- ・市販の IoT 開発キット(マイコン端末)とクラウドサービスで動作するようになっており、利用を簡単に開始できる。
- 自前のサーバーを用意する必要がないので、メンテナンスの手間もかからない。
- ・マイコン端末の大きさは 5cm 四方・厚さ 2cm 程であるので、スペースを占有せず、電源・ネットワーク以外は特に必要がないため、設置が簡単である。
- 物だけ用意すればよく、工事の必要はない。
- ・マイコン端末の値段は約 9000 円(2024 年 5 月現在)であり、ルーターや SD カード、付属 GNSS Module 等の費用を考慮しても 5 万円以下で導入できる。
- ただし、インターネット通信費や配送料、電気代は別にかかる。
- ・配布したプログラムはオープンソースであり、それぞれのニーズに合うようにカスタマイズすることも可能。

1.4. 本システムに必要な物品

- ・絶対に必要 (ないとシステム自体が動作しない)

(i)IoT 開発キット M5Stack Core2 (マイコン端末)

-Switch science([SwitchScience — スイッチサイエンス \(switch-science.com\)](https://switch-science.com/))やマルツ([電子部品・半導体パーツの通販 販売 | マルツオンライン \(marutsu.co.jp\)](https://marutsu.co.jp/))から購入可能。

(ii)microSD カード

-容量は 16GB 以下にすること。32GB 以上のものはマイコン端末側が認識できない。

(iii)ルーター

-モバイルルーターでもルーターとして機能すれば(IP アドレスの割り振りができれば)OK。

(iv)PC

-PC で Arduino IDE を立ち上げて、マイコン端末にプログラムを書き込む。

(v)電源

- M5Stack Core2 には 500mAh のバッテリーが搭載されているが、船体動揺監視 IoT システムのプログラム(m5shipmonitorserver.ino)はバッテリーのみだと最大でも 2 時間しか動作しない。長時間使う場合は USB-C から給電してあげる必要がある。モバイルバッテリー・コンセント両方から給電可能。

- ・特定の用途を達成するために必要

(i)インターネット接続

-クラウドへのアップロード及び Web Page 上でグラフを描くためにインターネットへ接続する必要がある。

(ii)Google アカウント

-Drive に Spreadsheet を作成・保存しておくためにアカウントが必要。

-会社や学校のアカウントだと、セキュリティーの仕様上、マイコン端末からのアップロードを受け付けるように設定できない可能性がある。

(iii)付属 GPS+BDS Module([Mini GPS/BDS Unit \(AT6558\)](#) 別売)とバッテリー拡張ボトム ([M5GO Battery Bottom2 \(for Core2 only\)](#) 別売)

-GNSS 測位により時刻合わせしたい場合に必要。

2. 使用方法

本システムは主にマイコン端末によって動作するが、端末の動作は Arduino 言語で記述する。

2.1. 船体動揺監視 IoT システム使用前の事前準備

① プログラムのダウンロード

[Komori-GitHub/ShipMonitor_IoT: 船体動揺監視 IoT システム](#) にアクセスして、プログラム群と設定ファイルをダウンロードする。プログラムの内訳は表 1 に示すとおりである。

表 1 配布したプログラム・設定ファイルの内訳

動作対象/言語	ファイル名	
マイコン端末 (M5Stack Core2) /Arduino	m5shipmonitorserver.ino	船体動揺監視 IoT システムを動かすためのプログラム
	m5shipmonitor_offline.ino	船体動揺監視 IoT システムを専らオフライン環境で用いる場合を想定して、クラウド上へアップロードする機能を省いた
	mpu6886fftcsvfile.ino	マイコン端末の IMU で計測した角速度・加速度や動揺周期の計算結果を csv ファイルに書き込むプログラム
	rtcntp_date_time.ino	NTP サーバーの時刻で M5Stack Core2 内部の <u>Real Time Clock</u> をセットするプログラム
	rtc_comxgps.ino	マイコン端末付属(別売り)品である GNSS 受信機を用いて <u>Real Time Clock</u> をセットするプログラム
マイコン端末 /HTML・Javascript	read/script/*chart*.html table.html	Web Page を描画するためのプログラム
マイコン端末 /C++	MahonyAHRS.cpp MahonyAHRS.h MPU6886.cpp MPU6886.h	それぞれ、Madgwick Filter と IMU のセンサーから値を読み出すプログラム。Madgwick Filter のパラメータの調整を行い&オフセットを取り除いた角速度・加速度を統合するように変更したので、プログラムを上書きする必要がある。
Google サーバー上のコンピューター /Google Apps Script(javascript)	WritePostedData.gas	マイコン端末より POST された動揺周期・GM 推定値のデータ(JSON 形式)を受け取り、これを読みだしてスプレッドシートに書き込むプログラム

マイコン端末	read/ shipparticular.txt wificloud.txt	GM 計算に使う船の幅や、WiFi 接続に使う情報を書き込んでおく設定用ファイル
--------	---	--

② 開発環境の構築

- (i) 統合開発環境”Arduino IDE”([Arduino IDE | Arduino Documentation](#))を PC にダウンロード。
- (ii) ダウンロードした Arduino プロジェクトフォルダ(*.ino が保存されたフォルダー)を適当な場所に保存した後、*.ino ファイルを開く。

Name	Status	Date modified	Type	Size
m5shipmonitorserver.ino		2024/04/14 20:30	INO File	20 KB

図 3 Arduino のソースファイル(.ino)

- (iii) Arduino IDE が立ち上がった後、“Library Manager”を開いて arduinoFFT(ver1.6.2)、M5Core2(ver0.1.9)、ArduinoJson(ver.7.0.4)ライブラリをインストールする。

※これと異なるバージョンのライブラリをインストールすると仕様が異なるため、プログラムコンパイル時にエラーが生じる可能性がある。

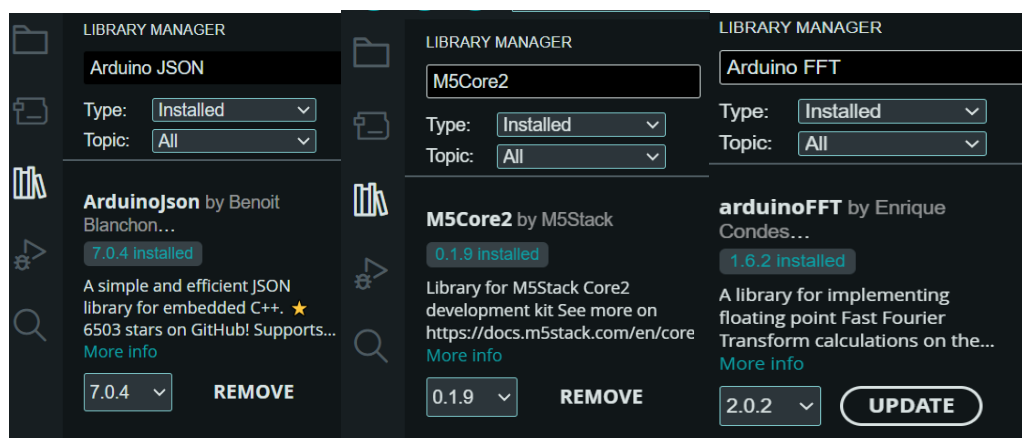
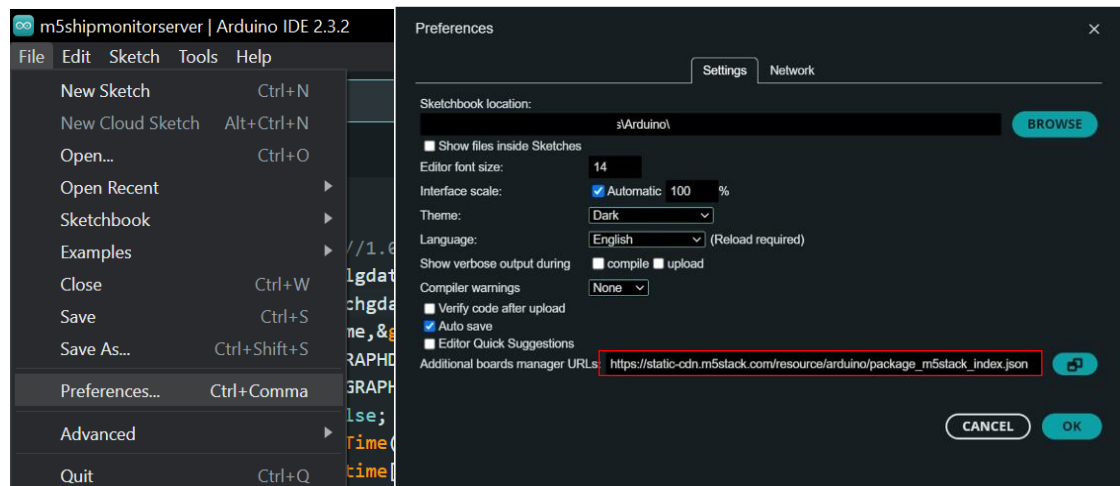


図 4 インストールするライブラリ

- (iv) “Preferences”を開いて”Additional boards manager URLs”のところに、“https://static-cdn.m5stack.com/resource/arduino/package_m5stack_index.json”と入力して、OKを押す。



(v) “Boards Manager”を開いて M5Stack のボードをインストール

図 5 boards の設定

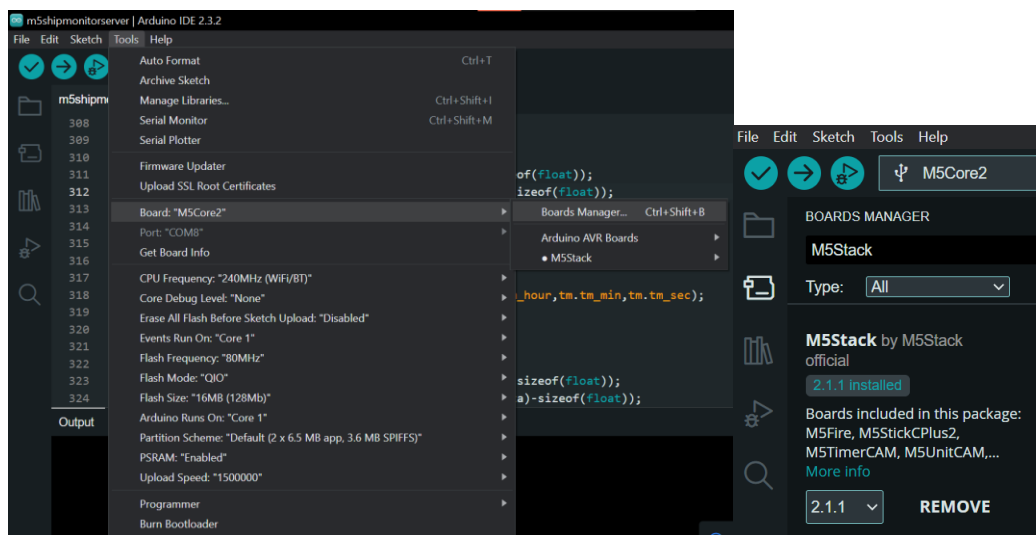


図 6 インストールするボードマネージャー

なお、Arduino IDE で M5Stack Core2 の開発環境をセットアップする方法についてはこちら [m5-docs \(m5stack.com\)](https://m5stack.com/docs) を参考にしてください。

③ ライブラリファイルの入れ替え

Arduino のライブラリがダウンロードされたフォルダーを見つけて、/libraries/M5Core2/src/utility 内にある MahonyAHRS.cpp・.h、MPU6886.cpp・.h をダウンロードしたものの上に書きする。

-ライブラリは大抵、Arduino プロジェクトと同じディレクトリもしくは 1 つ上のディレクトリに保存されている。

Documents > Arduino > libraries > M5Core2 > src > utility					Search utility
Name	Status	Date modified	Type		
In_eSPI_Setup.h	✓	2024/04/09 12:08	C Header ソースファイル		
M5Button.cpp	✓	2024/04/09 12:08	C++ ソースファイル		
M5Button.h	✓	2024/04/09 12:08	C Header ソースファイル		
M5Timer.cpp	✓	2024/04/09 12:08	C++ ソースファイル		
M5Timer.h	✓	2024/04/09 12:08	C Header ソースファイル		
MahonyAHRS.cpp	✓	2023/10/22 10:28	C++ ソースファイル		
MahonyAHRS.h	✓	2023/07/31 13:28	C Header ソースファイル		
MPU6886.cpp	✓	2023/08/20 16:27	C++ ソースファイル		
MPU6886.h	✓	2023/08/20 16:11	C Header ソースファイル		
pngle.c	✓	2024/04/09 12:08	C ソースファイル		
pngle.h	✓	2024/04/09 12:08	C Header ソースファイル		

図 7 ライブラリのソースファイル

④ SD Card の用意。

ダウンロードした“read”フォルダーをそのまま SD Card(別売り)に保存する。

⑤ Google Spread Sheet の作成とサーバーApp Script の書き込み

この IoT システムは Google Drive 上の Spread Sheet に動揺周期と GM 推定値を書き込む。

- (i) PC 上の Google Drive で、(使用したい)Google アカウントにログインをし、スプレッドシートを新規作成。

※注 学校や会社のアカウントだと(iii)のところで共有設定を「全員」にできず、マイコン端末から Google Drive へのアップロードができなくなる恐れがある。

- (ii) スプレッドシートを開いたら、Apps Script の画面を開き、“WritePostedData.gas”に記載されたプログラムを editor に貼り付けた後、“新しいデプロイ”をクリック。

- (iii) 新しいデプロイ画面でアクセスできるユーザーを全員に設定してデプロイをクリック。

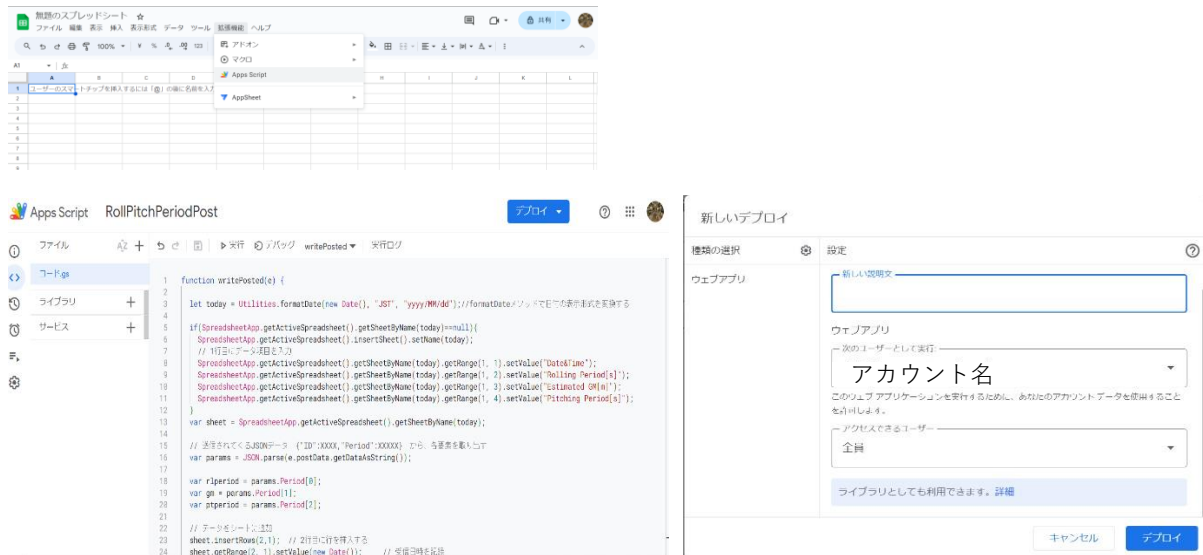


図 8 Google Spreadsheet の画面と Apps script

⑥ M5Stack の設置

表 2 M5Stack 設置時の要件

要件		左の要件を満たさなかった場合に生じる弊害
分類	要件	
端末の設置場所	・ ほぼ水平なところ	Roll・Pitch が正しく計算できない
	・ なるべく振動が少ないところ	姿勢は加速度も使って計算しているため、振動によって誤差が生じる
	・ 使用したいネットワークの無線 LAN に接続できるところ ※できればインターネットに接続できることが良い	ネットワーク(ルーター)に接続しないとこのシステムは動作しない(ずっとルーターに接続しようとして先に進まない)。 インターネットに接続しないと、Web Page 上のグラフの描画と Google Drive への接続ができない。
設置する向き	図 9 に示した 2 通りのパターンのうちどちらか 1 つに合わせて置く。端末の辺が船首尾線に水平になるように注意する。なお、置いた向きについては⑦で後述するように設定ファイルで指定する。	端末の辺が船首尾線に平行でない場合は Roll・Pitch が正しく計算できない。 設定を反対にすると、Roll と Pitch の値が入れ替わってしまう。

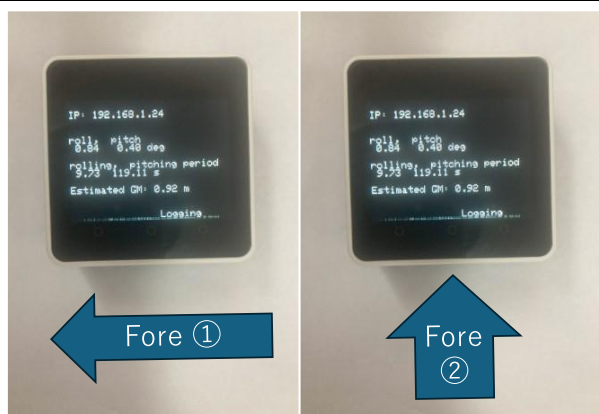


図 9 M5Stack Core2 を置く向き

⑦ 設定の書き込み

テキストファイルに表 3 に示す設定を書き込んでおく。これらの設定は WiFi・クラウドサービスへの接続や GM 計算に必要となる。なお、プログラム内では何行目に何の設定が書き込まれているかを規定しているため、ファイルの行を追加・削除すると正常に動作しなくなる。図 10 に示すように、SD Card 内

の”read”というフォルダ内に設定ファイルが保存されている。

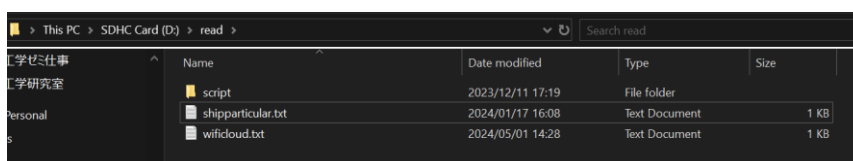


図 10 SD Card に保存した設定ファイル

表 3 ファイルに書き込むデータ

ファイル名	読み込むデータ	
wificloud.txt	行目	内容
	1	WiFi の SSID
	2	WiFi の password
	3	時差 1 時間単位で -12~+12(+/-が UTC に対して進んで/遅れていることを意味する。)の範囲で入力。
	4	Google Spread Sheet にアクセスするための URL
Shipparticular.txt	行目	内容
	1	船幅[m]
	2	慣動半径の係数 ※0.40 が良く用いられるが、復原性資料に記載されているより 厳密な値を用いたい場合はそれを記入。
	3	M5Stack Core2 を置く向き

⑧ プログラムの書き込み

船体動揺監視システムを動かしたい場合は、”m5shipmonitorserver.ino”を Arduino IDE で開く。シリアルのドロップダウンを開いて Upload する Port を選択した後に、Upload ボタンをクリック。クリック後、ビルド&コンパイルが始まる。

※コンパイル・upload 完了まで 10 分~1 時間かかります。特に初回は時間がかかります。

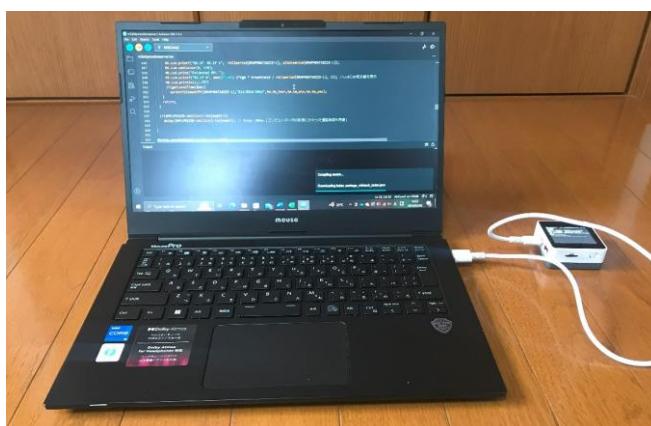


図 11 PC によるマイコン端末へのプログラムの書き込み

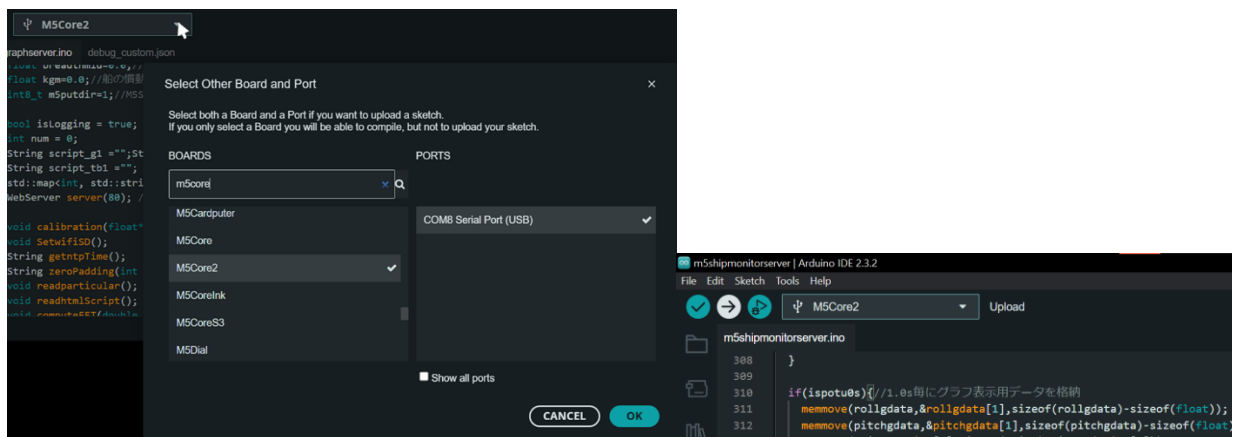
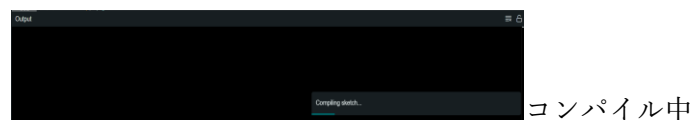


図 12 port の選択及び upload ボタン



コンパイル中

```
Sketch uses 986533 bytes (15%) of program storage space. Maximum is 6553600 bytes.
Global variables use 64144 bytes (1%) of dynamic memory, leaving 4457840 bytes for local variables. Maximum is 4521984 bytes.
esptool.py v4.5.1
Serial port COM8
Connecting...
```

コンパイル完了後

```
Writing at 0x000ff6c7... (100 %)
Wrote 992832 bytes (630996 compressed) at 0x00010000 in 9.4 seconds (effective 847.2 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

upload 終了後

図 13 コンパイル中/後のターミナル(Output)の表示

⑨ プログラムの動作開始

ビルド&コンパイルが終わると、船体動揺監視システムが動作し始める。正常にネットワークにつながった場合、次のような順序で動作する。Calibration 中は端末を動かしたり、衝撃を与えないようにすること。そうでないと、姿勢計測に誤差が生じてしまう。



設定読み込み



キャリブレーション中



キャリブレーション終了後(最初の 1 分は動揺周期・GM が表示されない。)

図 14 M5Stack の動作開始時の画面

2.2. 船体動揺監視 IoT システム動作時の使い方

① マイコン端末上に立ち上げた Web Page での確認

- (i) PC やスマートフォンをマイコン端末と同じネットワークに接続する。
- (ii) 図 15 に示すような動揺周期や GM 推定値の表を確認したい場合には”<http://m5shipmonitor.local/table>”にアクセス。船体傾斜や動揺周期のグラフを確認したい場合には”<http://m5shipmonitor.local/graph>”にアクセス。場合によっては DNS がうまく機能しないため、その時は代わりに IP アドレス(マイコン端末の LCD に表示してある)を”<http://192.168.1.24/graph>”のように打つことでアクセスできる。

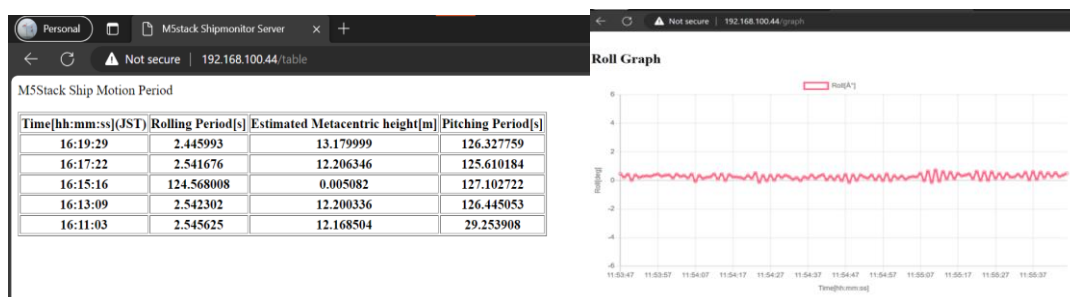


図 15 WebPage での表示

左:動揺周期・GM 推定値の表 右:船体傾斜のグラフ

② Google Drive のスプレッドシートでの確認

- ①で作成した Spread sheet を開くと、図 16 のように動揺周期・GM 推定値を確認できる。値はシステム(マイコン端末)の動作時に自動で書き込まれるが、値が更新されていないと思われる場合は Reload して頂きたい。

Figure 16 shows a Google Spread Sheet titled "Roll/Pitch Period". The sheet contains the following data:

	A	B	C	D
1	Date&Time	Rolling Period	Estimated GM	Pitching Period
2	2023/09/22 10:56:42	4.2538	4.357845186	23.5088
3	2023/09/22 10:54:32	11.6517	0.5808277311	15.5869
4	2023/09/22 10:52:22	127.7203	0.004833993606	29.4584
5	2023/09/22 10:50:12	25.0049	0.1261175971	4.2583
6	2023/09/22 10:48:03	29.0829	0.09322888959	14.1127

図 16 Google Spread Sheet に書き込まれた値

③ SD Card に保存された csv ファイルでの確認

このように SD Card へ”imu_data_観測日時.csv”というファイル名で保存されているので、SD Card を開けば IMU 計測値などが確認できる。Csv ファイルに書き込まれるデータは表 4 の通りである。

表 4 csv ファイルに書き込まれるデータ

列	A	B	C	D	E	F
項目	JST	ms	num	gyroX	gyroY	gyroZ
説明と[単位]	[hh:mm:ss]	経過秒[ms]	通し番号	IMU X 軸の 角速度[° /s]	IMU Y 軸の 角速度	IMU Z 軸の 角速度
列	G	H	I	J	K	L
項目	accX	accY	accZ	roll	pitch	yaw
説明と[単位]	IMU X 軸の 加速度[G]	IMU Y 軸の 加速度	IMU Z 軸の 加速度	角速度と加 速度を統合 して求めた Roll[°]	角速度と加 速度を統合 して求めた Pitch[°]	角速度を積 分して求め た Heading[°]
列	M	N	O	P	Q	
項目	temperature	roll period	freq	Pitch period	freq	
説明と[単位]	IMU 内部温 度[°C]	Rolling 周期 [s]	Rolling 周波 数[Hz]	Pitching 周 期[s]	Pitching 周 波数[Hz]	

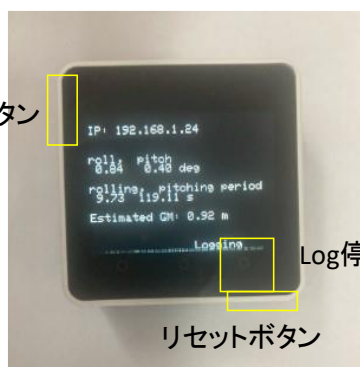
The image shows a file explorer view of an SD card with several CSV files. To the right, a preview of the data is shown, displaying columns for time, gyro data, acceleration, and other sensor readings.

図 17 SDCard に保存された csv data

④ システムの終了と再開

- ・LCD ディスプレイ下の一番右のボタン(○)をクリックすると、SD カードへの記録が停止する。
 - ・M5Stack の電源ボタン(側面にある)を 6 秒以上長押しすれば端末の電源を落とせる。電源を入れるときはもう 1 回電源ボタンを押せばよい。
 - ・M5Stack の Reset ボタン(側面)を押せばプログラムを再起動できる。
- たまに動作がフリーズすることがあります。その時は再起動してください。

電源ボタン



Log停止ボタン

リセットボタン

図 18 操作ボタン

2.3. マイコン端末内部の RTC の時刻合わせ

本システムではマイコン端末搭載の RTC の時刻を用いるが、RTC は月日を経つと誤差が蓄積するため、時刻合わせのプログラムで誤差を補正してあげる必要がある。誤差の蓄積量は著者が自分で確かめた限り 1 週間に 2~3 秒程度だったが、個体・環境によって異なる。

① “rtcntp_date_time.ino”での時刻合わせ

このプログラムは NTP サーバーの時刻を用いて、RTC を正しい時刻にセットするプログラムである。マイコン端末がインターネット(無線 LAN)につながる場所において、このプログラムを Arduino IDE でマイコン端末に書き込むことによって、時刻合わせのプログラムが動作する。WiFi の SSID やパスワード、時差は“wificloud.txt”に書き込む。

② “rtc_comxgps.ino”での時刻合わせ

このプログラムは GNSS 測位解として得られる時刻を用いて、RTC を正しい時刻にセットするプログラムである。使用手順を以下に示す。

(事前準備) GPS モジュール([Mini GPS/BDS Unit \(AT6558\)](#) 別売) とバッテリーボトム([M5GO Battery Bottom2 \(for Core2 only\)](#) 別売)を用意しバッテリーボトムをマイコン端末の底面にはめる。ボトムは六角ねじで着脱ができる。

(i)PortC に GPS モジュールのケーブルを差し込み、プログラムを書き込む。



図 19 マイコン端末 PortC

(ii)GPS モジュールを衛星からの電波が受信できる屋外や屋内であれば窓側に置く。



図 20 GPS モジュールとマイコン端末

2.4. 使用上の注意

① ライブラリのバージョンについて

2.1 に関して、ライブラリの新バージョンがリリースされると、Update するか聞かれますが、Update した場合にライブラリの仕様変更のために、プログラムが正常に動作しなくなる(コンパイル時にエラーが出る)可能性があります。もしエラーが出た際は図 4 のようにライブラリマネージャーを開いて 1 回 Update したライブラリをアンインストールしてから、指定のバージョンのライブラリを再度インストールしてください。



図 21 Update を促すメッセージ

② IMU の計測値について

マイコン端末に搭載されている低コスト(Consumer level) MEMS-IMU は温度によって誤差が蓄積し、著者が恒温槽で行った研究では 40°C 程度の温度変化で、Roll, Pitch に最大 2° 弱の誤差が生じました。この端末は一般商船に搭載されている 電子傾斜計の性能基準を満たしません。Roll, Pitch の角度はあくまでも参考程度にお使いください。

③ GM 推定値の正確性について

著者が汐路丸で行った実験の結果として、横揺れ周期から誤差 10% 以内で G_0M (自由水影響を考慮した横メタセンタ高さ)を推定できましたが、他の船でどれくらい正確に求められるかは自身でご確認下さい。

④ 今後のアップデート方針について

アップデートは GitHub 上に新しいプログラム群(バージョン)をアップロードする形で行いたいと思います。フィードバック等が得られましたらそれに基づいた修正を行うつもりですが、著者の時間・能力の制約上、ご要望に応えられないことがございます。あらかじめご了承ください。

⑤ ライセンスについて

GitHub 上で公開されている船体動揺監視 IoT システム関連のプログラムはオープンソースです。引用元をソースファイル内などで明記していただければ、商用利用・変更したうえでの再配布も可能です。ただし、本システムはオープンソースの Arduino ライブラリを含んでおります。

引用文献

M5Stack. (2023 年 12 月). Document Core2 ver1.1. 参照日: 2024 年 3 月 12 日, 参照先: M5Stack Developer Center: <https://docs.m5stack.com/en/core/Core2%20v1.1>

Tomosoft. (2021 年 12 月 27 日). M5Stack Core2 内蔵のジャイロ加速度計「MPU6886」の入力. 参照日: 2023 年 2 月 8 日, 参照先: Tomosoft official ブログ: <https://tomosoft.jp/design/?p=45508>

toolware. (2020 年 5 月 20 日). M5Stack から Google スプレッドシートに送信. 参照日: 2023 年 8 月 20 日, 参照先: toolware.jp のブログ: <https://toolware.jp/m5stack-spreadsheet/>

下島健彦. (2022). みんなの M5Stack 入門 (第 第 1 版). リックテレコム.

利用ライブラリ

M5Stack:M5Core2, ver0.1.9, <https://github.com/m5stack/M5Core2.git>,2023.1.30.

Enrique Condes: arduinoFFT, ver2.0.2, <https://github.com/kosme/arduinoFFT.git>,2023.10.13.

Benoit Blanchon: ArduinoJson, ver7.0.4, <https://arduinojson.org/,2023.2.18>.

Mikal Hart: TinyGPSPlus,ver1.0.3, <https://github.com/mikalhart/TinyGPSPlus.git>,2022.1.25.s

Chart.js, ver.2.9.4, <https://www.chartjs.org/,2023.12.7>.