

# 複数センサに対応したBluetooth通信のためのフレームワークの提案

橋浦研究室

122D019 榎本真大

122D006 阿部寿樹哉

## 1 はじめに

マイコンから複数のセンサデータを Bluetooth Low Energy (BLE) で送信する際、高頻度の通信周期を要求するセンサを含むと送信遅延が増大し、要求周期に間に合わなくなるという問題が生じる。特に送信処理を同期的に実装した場合、すべての処理が直列に実行されるため、この問題が顕著に表れる。このような問題に対し、非同期処理による実装を用いると CPU がスリープせず処理を並行実行できるため、処理能力の向上が期待できる一方で、多数のセンサごとに異なる送信周期を非同期に管理しつつプログラムを実装・保守することは容易ではないという問題が生じる。そこで、本研究では BLE 通信における非同期送信処理の周期制御をスケジューラによって自動化するフレームワークを提案する。

## 2 研究目的

本研究の目的は、複数センサ混在環境で、高頻度センサを使用可能にすることである。これと同時に、非同期処理による BLE 送信プログラムの実装難易度低下と実装の信頼性を向上させる。特に、非同期処理のプログラミング経験がない初心者を主な対象とする。

## 3 提案手法

実行タイミングの管理をスケジューラに任せ、それをフレームワークとする。フレームワークの概要を図 1 に示す

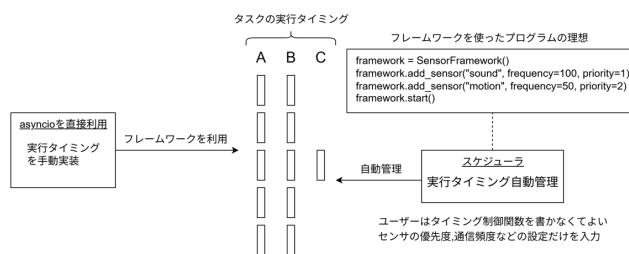


図 1: フレームワークの概要

## 4 実験と評価方法

以下のリサーチクエスチョン (以降 RQ) に回答するため、3つの実験により提案フレームワークの有効性を検証する。

- RQ1: 提案フレームワークは、同周期、同優先度のセンサにおいて、公平性がある通信を実現させるか
- RQ2: 提案フレームワークは、優先度による制御は高負荷環境での高頻度センサの周期の安定性を向上させるか
- RQ3: 提案フレームワークは、複数センサを用いた BLE 非同期通信プログラムの実装難易度を低下させるか

- RQ4: 提案フレームワークは、作成されるコードの信頼性を向上させるか

### 4.1 実験 1: フレームワーク性能調査

RQ1 および RQ2 に回答するため、合計 13 個のセンサタスクを单一のマイコン上で同時に動作させ、BLE 通知の実測周期に基づく性能評価を行った。評価手順として、送信側 (マイコン) にて送信時刻 ( $\mu\text{s}$ ) を付与したデータを Web アプリへ送信して時系列に記録し、連続する通知の到着時刻差から実測周期  $dt_i$  と締め切りミス率 [1] を算出する。評価指標には、実測周期の統計量 (平均、標準偏差、最大値等) に加え、締め切りミス率、理論周期  $P$  に対する無次元量  $dt_i/P$  を用いて分析した。

### 4.2 実験 2: 被験者による実装

被験者を提案フレームワークあり群となし群に分け、BLE センサ通信の実装課題を行わせ、提出された実装コードを比較評価する。評価指標として、MI [2]、初期化用メソッドを無効グラフの頂点から除くように改良した LCOM4 [3] (以降 LCOM4-NI)、非同期処理の難易度を表す指標 APCC を定義して用いた。APCC は OSS を分析し、非同期欠陥との正の相関が確認されている。MI と APCC により RQ3 を LCOM4-NI により RQ4 を評価する。

## 5 実験結果と考察

### 5.1 実験 1 の結果と考察

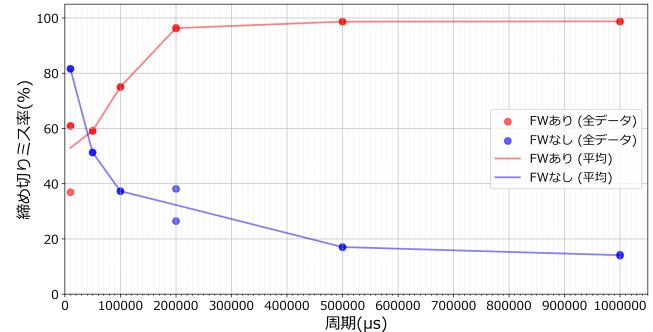


図 2: 締め切りミス率 [1]: FW あり vs FW なし

すべての同周期かつ同優先度のセンサ間の平均実測周期の差に有意差が見られなかったため、優先度制御により公平性が保たれていることが示された。さらに、同周期で異なる優先度のセンサ間の平均実測周期において、優先度の高いセンサの平均実測周期が t 検定の結果、有意に短くなっていたことから優先度による制御が適切に行われていることが示された。また、設計上のトレードオフとして、高頻度センサを優先した際に低頻度センサの締め切りミス率が上昇する傾向が確認されたさらに、同周期センサにおけるフレームワークあり/なしの比較として実測周期の平均値を用

いた分析を行ったところ、フレームワークで優先度を高く設定したセンサは締め切りミス率が低下し、優先度付与の有効性が  $t$  検定により支持された。一方で、高頻度センサを複数接続させた環境では、締め切りミス率が約 37%あり、周期の安定性が低い。そこで高頻度センサ何個までが安定するかを、最高優先度の単一高頻度センサの理想状態が、センサ追加によってどのように締め切りミス率が上昇するかを調べ、結果を図 3 に示した。

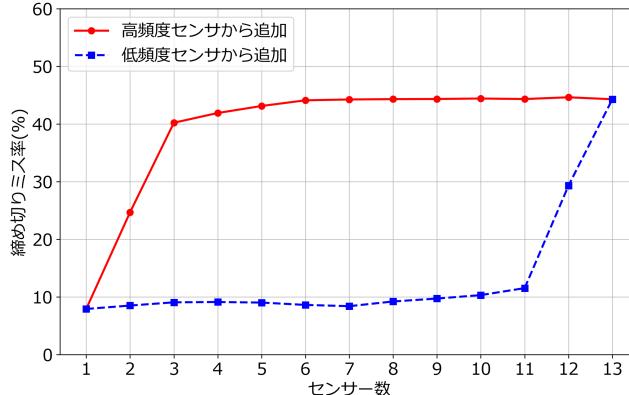


図 3: 最高優先度 (10ms) センサのセンサ追加による締め切りミス率の推移

高頻度センサを追加した場合、締め切りミス率が大幅に上昇し、低頻度センサを複数追加しても締め切りミス率の上昇が小さかったため、フレームワークは高頻度センサ 1 つのと低頻度センサ複数が限界だと考察できる。

## 5.2 実験 2 の結果と考察

表 1: 実験群 (fw あり) の記述統計量 ( $n = 12$ )

#	指標	標本数	平均値	中央値	標準偏差
1	APCC	12	25.0000	25.0000	0.0000
2	MI	12	66.6392	66.7000	0.1125
3	LCOM4(通常)	12	1.0000	1.0000	0.0000
4	LCOM4-NI	12	1.0000	1.0000	0.0000

表 2: 統制群 (fw なし) の記述統計量 ( $n = 12$ )

#	指標	標本数	平均値	中央値	標準偏差
1	APCC	12	39.7500	30.5000	1.5448
2	MI	12	57.8292	55.7100	4.8648
3	LCOM4(通常)	12	1.0000	1.0000	0.0000
4	LCOM4-NI	12	2.5833	2.0000	1.6765

表 3: Mann-Whitney の U 検定結果

#	指標	$U$	$p$ 値 (片側)	Cliff's $\delta$
1	APCC	0.0	0.0054	-1.0000
2	MI	144.0	0.0095	1.0000
3	LCOM4(通常)	72.0	1.0000	0.0000
4	LCOM4-NI	30.0	0.0014	-0.5833

APCC の低下により、実装難易度が低下したことが示された。MI の上昇により、保守性の観点からフレームワークが実装を改善する傾向が確認された。凝集度指標に関しては、通常の LCOM4 では明確な低下は観測されなかった一方、初期化メソッドの影響を除外

した LCOM4-NI では値の低下が確認された。統制群のコードを確認すると、LCOM4-NI が高い実装はクラス内部に (1) 送信機能と (2) 周期のタイミング制御の二つの責務が同居していた。その例を図 4 に示す。

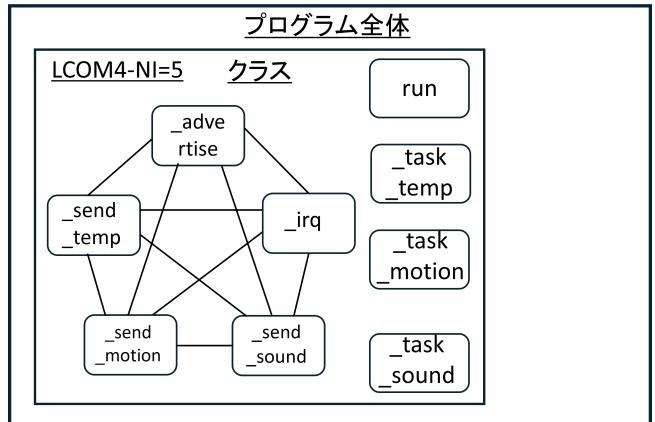


図 4: LCOM4=4.5 のプログラム構造 (責務の混在)

一方、LCOM4-NI が低い実装ではクラス内部に送信機能のみを配置し、タイミング制御はクラス外に記述されており、責務が分割されていた。したがって LCOM4-NI は、非同期周期通信実装における責務分割の有無を反映できる指標であることが示唆される。さらに、フレームワークなし群では責務分離の有無にばらつきが見られたが、フレームワークあり群ではタイミング制御がフレームワーク側により自動化されるため、タイミング制御をクラス内部に実装する必要がなく、結果として責務分割が構造的に達成され、信頼性が向上した可能性があることが明らかになった。

## 6 結論

フレームワークレベルでの優先度制御を導入することにより、タスクの締め切りミス率が低下することを確認した。特に、高頻度センサと低頻度センサが混在する環境においても、高頻度センサの処理を優先することで、リアルタイム性の要求に対して有効に機能することが示された。ただし、高頻度センサを複数同時使用する場合には、締め切りミス率が約 37%あり、安定性が低い。高頻度センサ 1 つと低頻度センサ複数なら運用可能である。また、フレームワーク化により実装難易度は低下し、信頼性も向上した可能性が高い。

## 参考文献

- [1] K.-H. Chen, Georg Von Der Brüggen, and J.-J. Chen, “Analysis of deadline miss rates for uniprocessor fixed-priority scheduling,” 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) (online), pp.168–178, 2018. (DOI:10.1109/RTCSA.2018.00028)
- [2] D. Coleman, D. Ash, B. Lowther, and P. Oman, “Using metrics to evaluate software system maintainability,” Computer (online), Vol.27, No.8, pp.44–49, 1994. (DOI:10.1109/2.303623)
- [3] S.R. Chidamber and C.F. Kemerer, “A metrics suite for object oriented design,” IEEE Transactions on Software Engineering (online), Vol.20, No.6, pp.476–493, 1994. (DOI:10.1109/32.295895)