RFID を活用した触れる GIS の操作に関する可能性と限界

中村 暢佑*・石黒 希海・中山 俊*****

Possibilities and Limitations of Touch-Based GIS Operation Using RFID

Yosuke NAKAMURA*, Nozomi ISHIGURO**, Shun NAKAYAMA***

Abstract: In recent years, tangible GIS (TGIS) has gained attention as an interface for spatial data manipulation. However, most systems sequentially process only a single layer, making it difficult to handle multi-layer data such as building-level or block-level information. To address this challenge, we explored RFID technology as a tangible interface for multi-layer operations. In a prototype using an M5Stack Core2 equipped with PaHUB, multiple RFID readers were used in parallel to capture tag IDs across channels in real time. While the system functioned, challenges remained, including recognition instability and the inability to read multiple tags simultaneously within a single reader. These results suggest that while RFID alone cannot fully support multi-layer TGIS, combining it with camera recognition or AR markers can provide a foundation. This research clarifies the potential and limitations of RFID, presenting a direction for hybrid approaches in future development.

Keywords: 触れる GIS（tangible GIS），RFID，並列接続（parallel connections）.

１．はじめに

近年，「触れる GIS（Tangible GIS）」と呼ばれるインタラクティブな意思決定支援ツールの開発が進展している(Francesco Tonini et al，2017)．触れる GIS とは，物理的なモデルをユーザーインタフェースにした地理情報システム(GIS)のことである．映像によるフィードバック等により，検討内容のより深い理解や納得感，共感を提供するものもあり，まちづくりの共創のシーンや意思決定プロセスにおいて大きな期待が寄せられている．

しかし，GIS のエッセンスとも言える「複数レイヤのオーバーレイ解析」だが，複数のレイヤを操作可能な触れる GIS は少ない．多くの実装では，建物や POI などいずれか一つのレイヤを対象とし，他の階層情報を同時に操作・表示することは難しい現状にある．この技術的制約は，ハードウェアの入力方式が一系統に限られることに起因している．

そこで本研究では，複数のレイヤを操作できるように，新しい技術の可用性を検証する．今回はこの課題に対する解決手段として RFID 技術の応用を検討する．RFID タグは小型かつ安価で，任意のオブジェクトや場所へ容易に取り付け可能な特徴を持つ．これを用いて効率的に地物を読み取る仕組みを構築できれば，既存の認識技術と組み合わせることで，複数レイヤを操作可能な触れる GIS の実現可能性が高まる．

２．RFID の検知手法の開発

筆者らが知りうる限り，触れる GIS において RFID が活用されている例はほとんどない．その理由として，単一のリーダーで一枚のタグしか読み取れないという制約が挙げられる．これを克服するには，複数のリーダーを並列的に接続し，それぞれのリーダーが異なるタグを独立に認識できる仕組みが不可欠である．一般に GIS で複数の地物を同時に扱うことは日常茶飯事で，並列処理と干渉回避の双方が必要となる．

本研究では，こうした要請に応えるために，複数のリーダーを I2C（Inter-Integrated Circuit）接続で統合し，その上でソフトウェア的にポーリング制御することで重複

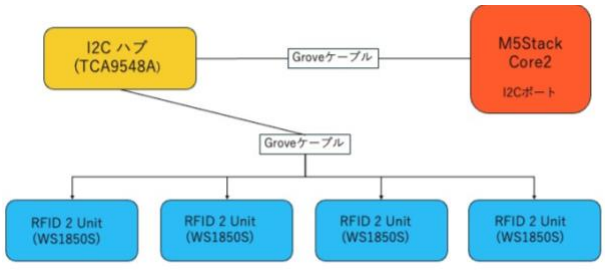


図 1　デバイス構造

*　学生会員　専修大学ネットワーク情報学部 (Senshu University)
〒214-8580　神奈川県川崎市多摩区東三田 2-1-1　Tel：044-712-7772 または E-mail：ne231112@senshu-u.jp
**　学生会員　専修大学ネットワーク情報学部 (Senshu University)
*** 正会員　専修大学ネットワーク情報学部 (Senshu University)

や競合を回避する．I2C 接続とは，フィリップス社が開発した 2 線式シリアル通信プロトコルで，単一のバス上で複数のデバイス（1 対 n)の通信を可能にする．これにより，リアルタイムに並列的に RFID の情報を取得する技術基盤の確立を目指した．

２．１　ハードウェア構成

本研究で用いたシステムは，M5Stack Core2 を中心に PaHUB を接続し，最大で 4 チャンネルの RFID リーダーを並列に接続可能な構成とした．PaHUB は I2C 信号を分岐し，複数デバイスをアドレス競合なしで接続するための拡張モジュールであり，各ポートを独立したバスのように扱うことができる．これにより，M5Stack Core2 が同一の I2C アドレスを持つ複数のリーダーを順次切り替えて利用できるようになった．

本研究で PaHUB を用いた理由は，RFID リーダーモジュールが固定的な I2C アドレスを持ち，複数台をそのまま直列接続するとアドレス競合が発生するためである．通常の I2C 通信では，デバイスごとに異なるアドレスを割り当てる必要があるが，複数接続すると制御不能に陥る．この制約を回避するために，I2C マルチプレクサである PaHUB を導入した．PaHUB は各ポートを独立した I2C バスとして切り替え可能にすることで，たとえ同一アドレスのデバイスであっても，複数台を順次切り替えて利用することができる．

図 1 に示すように，M5Stack Core2 を配置し，PaHUB を介して複数のリーダーを接続する．各リーダーは独立したチャネルに接続されており，物理的には同時に配置されたタグをそれぞれ担当することになる．

さらに，本研究では M5Stack Core2 のディスプレイを利用し，認識結果をその場で表示する仕組みを導入した．これにより，ユーザは現在認識されているタグの種類や系列を即座に確認できる．図 2 は，実際に 4 基のリーダーを PaHUB に接続した全容を示している．

２．２　ソフトウェア設計

ソフトウェア面においては，複数リーダーからの入力をポーリング制御し，競合や重複を回避する処理を導入した．具体的には，リーダーごとに「カードが置かれた」イベントと「カードが外された」イベントを検出し，状態の変化をトリガーとして記録する方式を採用した．この方式により，単純なループによる一時的な検出漏れを防ぐことができる．

認識されたデータは，M5Stack Core2 のディスプレイに表示されると同時に，シリアル通信を通じて PC に送信される．PC 側では Python スクリプトを用いてログを VV 形式で記録し，読み取りのタイムスタンプを保存する．

３．検証と結果

本研究における提案手法の有効性を検証するため，実際

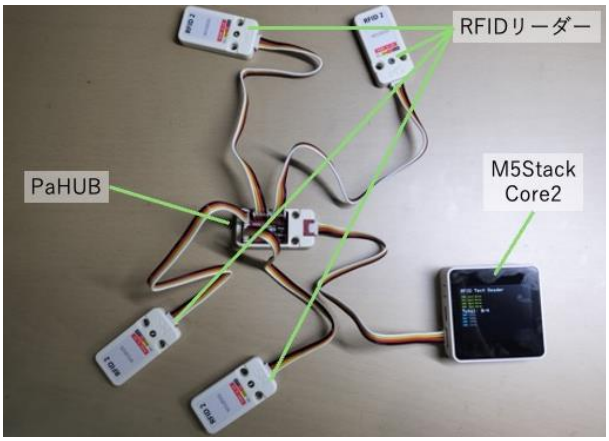


図 2　リーダーの全容

に複数の RFID タグを用いた検証を実施した．本検証は，複数タグを実際に配置した環境において行う．PC 側で保存した CSV ログを解析し，読み取りが時系列的に正しく記録されているかを確認する．特に，二種類以上のタグを近接配置した場合においても，それぞれが誤認識されることなく独立に認識されているかを検証する．

３．１　検証手順

実験の流れは以下のように設定した．

１．複数の RFID タグを用意し，それぞれを異なるリーダーに配置する．

２．M5Stack Core2 を起動し，各リーダーからのタグ検出を監視する．

３．Core2 のディスプレイに表示されるタグ ID を確認する．

４．並行して，シリアルモニタ出力を PC で受信・記録する．

５．PC 側に生成された CSV ファイルを開き，タグ ID とタイムスタンプが正しく記録されているかを検証する．

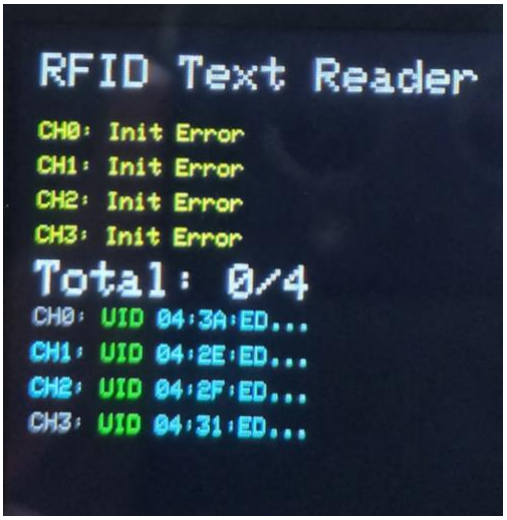


図 3　Core2 ディスプレイに表示されたタグ ID

３．２　検証結果

図 3 に示すように，のディスプレイ上には「UID XX:XX:XX…」といった形式でタグの固有 ID が表示される．これにより，個々のタグが確かに検出されていることが目視で確認できる．検証の結果，複数のリーダーに異なるタグを設置した場合でも，Core2 がそれぞれのタグ ID を正しく読み取り，順次ディスプレイに表示することが確認された．また，シリアルモニタを通じて出力された情報を確認すると，各タグが検出されたタイミングで ID が出力されていることも確認できた．

図 4 に示す通り，例えば「Card detected – UID: XX:XX:XX:XX:XX:XX」といった形式で逐次出力され，複数タグが同時に存在してもそれぞれの検出イベントが記録されていた．

さらに，PC 側に保存された CSV ファイルには，タグ ID とともに検出時刻がタイムスタンプとして記録されていた．図 5 に示すように，CSV には「datetime，timestamp，channel，action，card_uid」といった形式でデータが並び，後から実験履歴を分析できる状態が確保できた．

４．考察

４．１　RFID の認識技術における新規性

本技術は既に Web 記事でも紹介されている，一般的な方法をベースにしている．同記事では，MFRC522_I2C ライブラリと I2C ハブ（TCA9548A）を使用してシンプルに RFID リーダーを制御し，複数チャネルからの読み取りを可能にする構成が紹介されている(@misoca12，2024)．しかし，Web 記事に基づいて RFID 読み取りの試みを行ったところ，読み取れない・不安定という問題が生じたため，本研究で再構築を試みた．

同手法による読み取りが失敗した理由は二つ考えられる．一つ目は，GPIO ピンエラーだ．元のコードではリセットピンに GPIO12 が指定されていた．しかし M5Stack

```
========================================
🟢 CARD PLACED - Channel 0
   Card UID: 04:3A:ED:0A:C1:2A:81
   Timestamp: 427
========================================
EVENT:CH0:PLACED:04:3A:ED:0A:C1:2A:81:427
STATE:04:3A:ED:0A:C1:2A:81,-,-,-

========================================
🔴 CARD REMOVED - Channel 0
   Previous UID: 04:3A:ED:0A:C1:2A:81
   Timestamp: 432
========================================
EVENT:CH0:REMOVED:04:3A:ED:0A:C1:2A:81:432
STATE:-,-,-,-
```

図 4　シリアルモニタの出力

	A	B	C	D	E	F
1	datetime	timestamp	channel	action	card_uid	
2	0:03:43	868	0	PLACED	04:3A:ED:0A:C1:2A:81	
3	0:03:48	874	0	REMOVED	04:3A:ED:0A:C1:2A:81	
4	0:03:52	878	1	PLACED	04:2E:ED:0A:C1:2A:81	
5	0:04:01	887	0	PLACED	04:3A:ED:0A:C1:2A:81	

図 5　CSV ファイルの内容

Core2 では GPIO12 は利用できず，実行時にエラーが発生した．このため，読み取り動作自体が停止してしまった．そこで解決策として，リセットピンを無効化するコードに書き換えた．

二つ目は，カード検出の不安定さだ．元コードではカードの検出処理が 1 回しか行われないため，タイミングによってはタグを検出できないまま処理が終了してしまう．その結果，読み取り漏れが頻発した．そこで解決策として，検出を 2 回以上繰り返す処理を導入した．これにより，読み取りの冗長性を確保できた．

４．２　他の物体検知技術との連携

本研究において，RFID 技術を用いた複数地物の同時操作を試みた結果，一定の有効性が確認できた．では RFID 単独で複数レイヤを扱うことは可能だろうか．

RFID はタグごとにユニークな ID を持ち，それを用いて「建物」「街区」といった異なる階層を識別すること自体は可能である．しかし，現在の M5Stack Core2＋PaHUB 構成では少なくとも三つの制約が確認された．一つ目は，一つのリーダーで一度に読み取れるタグは一枚のみであり，複数のタグが同時に存在すると競合や検出漏れが発生することである．二つ目は，位置情報を得られない点である．すなわち，RFID は「どのタグが存在するか」は判定できるが，「どこにあるか」「どの向きか」といった GIS に重要な空間的属性を直接扱えないという点である．三つ目は，リーダーサイズの制約である．複数のレイヤを一度に操作できるほどのスケールにリーダーを小型化・多数配置することは現段階では容易でない．

したがって，RFID のみで複数レイヤを操作可能な触れる GIS を実現するのは困難であり，複数リーダーの配置

表 1　触れる GIS に用いられてきた技術と RFID との共存可能性

技術	主な用途・特徴	RFIDとの共存可否	コメント
カメラ画像認識	映像から物体位置・動きを識別.	高い共存性	RFIDが取りにくい範囲・動きの把握を補完.
ARマーカー	2Dマーカーを読み取り，高精度な位置認識が可能.	高い共存性	RFIDが識別，ARが空間位置を補完.
タッチセンサー	物理接触を通じて位置や入力を検出.	中程度	非接触不可.RFIDと機能が重複する部分あり.
投影型UI（プロジェクションマッピング）	投影による操作・フィードバック.	中程度	光条件に依存.装置構成が複雑.

＋他技術の補完が必須である．そのため，他の物体検知技術との連携を検討する必要がある．これまでの国内外の触れる GIS の研究や実装において，様々な物体検知技術が導入されている．画像認識，や AR マーカーなどはその代表例である．

これらは物体の上方からの検知という点で RFID とは異なる特性を持つ．その点，下方から物体を検知可能な RFID は，触れる GIS に導入されてきた物体検知と競合するのではない．今後このような要素技術との連携手法の確立が求められる．

５．おわりに

従来の触れる GIS では，ハードウェアの入力方式が一系統に限られるため，単一のレイヤしか操作できず，GIS の本質的機能を十分に活用できていなかった．そこで本研究では，複数レイヤを操作可能な触れる GIS の開発を目的に，RFID 技術を応用したアプローチを提案した．

M5Stack Core2 と PaHUB を用いて複数の RFID リーダーを I2C 接続で統合し，ソフトウェア的なポーリング制御により競合や重複を回避する方式を開発した．最大 4 チャンネルの並列 RFID 読み取りシステムを構築し，複数の RFID タグを異なるリーダーに配置した検証を実施した結果，各リーダーが独立してタグを認識し，時系列的に正しく記録されることが確認された．

複数 RFID リーダーの並列制御という基盤技術の確立に成功し，従来の単一系統入力の制約を部分的に克服できた．一方，複数レイヤ操作の実現という観点では，一つのリーダーで同時に読み取れるタグが一枚のみであること，タグの位置や向きといった空間的属性を取得できないこと，複数レイヤを効果的に操作できる規模までのリーダー小型化・多数配置が困難であることなどの制約が明らかになった．これらの制約により，RFID 単独での複数レイヤ操作は困難であることが判明した．

しかし，既存の触れる GIS で活用されてきた画像認識や AR マーカーなどは，上方からの物体検知技術である．これは上方からの検知に拘らない RFID との連動に親和性があると言える．このような要素技術との連携手法の確立により，触れる GIS がより多様な GIS 機能を有する可能性が見えた．

謝辞

慶應義塾大学総合政策学部の春日裕信氏には本研究の遂行にあたり，多大なるご指導とご協力を賜った．また，株式会社サトーの山内隆宏氏は RFID 活用法の提供の面でご尽力していただいた．そして，専修大学中山研究室の各位にも多くのご助力をいただいたことをここに記し，深く感謝の意を表す．

参考文献

@misoca12 (2024) M5Stack で複数の RFID タグを同時に読み取る. <https://qiita. com/misoca12/items/e7bfda1d454382e986ff>.

Francesco, T.，Anna, P.，Douglas, A，S.，& Brendan, H. (2017)Tangible geospatial modeling for collaborative solutions to invasive species management. *ResearchGate*，177.

Yun, W.，Ying, Z.，LingYu, Z. (2024) Developing a digital management system for museum collections using RFID and enhanced GIS technology. *PeerJ Computer Science*，18.