

OIAS-SR 10月 RFID演習 RFIDとPythonを用いた バックエンドシステムの構築

岡山大学大学院環境生命自然科学研究科
情報セキュリティ工学研究室

奥山 諒

授業構成

- RFID技術の基礎説明
- RFIDタグとリーダーの操作
- Pythonを用いたRFIDデータの取得と処理
- バックエンドシステムの構築

授業構成

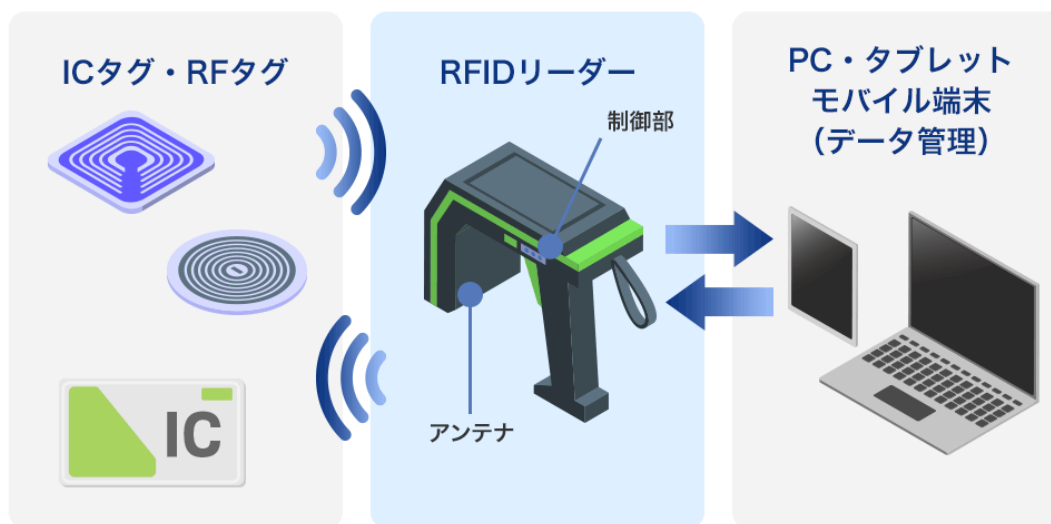
- RFID技術の基礎説明
- RFIDタグとリーダーの操作
- Pythonを用いたRFIDデータの取得と処理
- バックエンドシステムの構築

RFID技術の基礎説明

- **RFID (Radio Frequency Identification)**

- 電波や無線通信を利用して物体の識別や追跡を行う技術
- リーダーでタグを読み取り、PCなどにデータを保存

RFIDリーダーの仕組み



引用:<https://www.e-garde.co.jp/wp-content/uploads/2024/06/leader-works.png>

RFID技術の基礎説明

• 実際の活用例

- ユニクロでの無人会計システム
- ICOCA・SUICA、電子マネー



引用:<https://cdn-ak.f.st-hatena.com/images/fotolife/k/karaage/20220710/20220710183825.jpg>



引用:<https://www.ryutsuu.biz/images/2018/10/20181017uniqlo-1.jpg>

RFID技術の基礎説明

• RFIDの4つのタイプ

通信帯	LF帯	HF帯	UHF帯	マイクロ波帯
周波数	～135KHz	13.56MHz	860～960MHz	2.45GHz
通信範囲	～10cm	～30cm	～20m	～3m
用途例	自動車のスマートキーなど	交通系ICカード	アパレルタグや製品の持出管理など	(旧周波数帯)

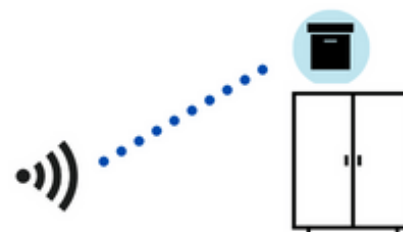
今回の演習ではUHF帯を使用

RFID技術の基礎説明

• RFIDのメリット



一括でデータが読める



遠隔からデータが読める



箱を開けずにデータを読める



タグが汚れてもデータを読める

引用:<https://www.smartmat.io/hs-fs/hubfs/as-to-rfid03.png?width=680&height=346&name=as-to-rfid03.png>

RFID技術の基礎説明

・バーコードとの比較

種類	データ 書き換え	同時 読み込み	読み取り 距離	遮蔽物や 汚れの影響	保有情報量	コスト
バーコード 	不可	不可	～数10 cm	見えている 必要があり 汚れにも弱い	バーコード 20文字程度 QRコード 7,089字	RFIDに 比べ安い
RFID 	可能	可能	～数 m	遮蔽物の 影響が少なく 汚れに強い	数1,000字	バーコードに 比べ高い (大ロットで安くて も約10円前後)

引用:https://www.nec-solutioninnovators.co.jp/sp/contents/column/img/20220128/img_main2.png

RFID技術の基礎説明

• RFID

- 電波で物体の識別や追跡
- 一度に複数識別
- タグを取り出す必要がなく、効率的
- データをPCやインターネット上のDBなどに保存可能



今回の目標

リーダの操作、タグの検知

Pythonを用いてリーダーの操作、データの取得
バックエンドシステムでデータの確認

授業構成

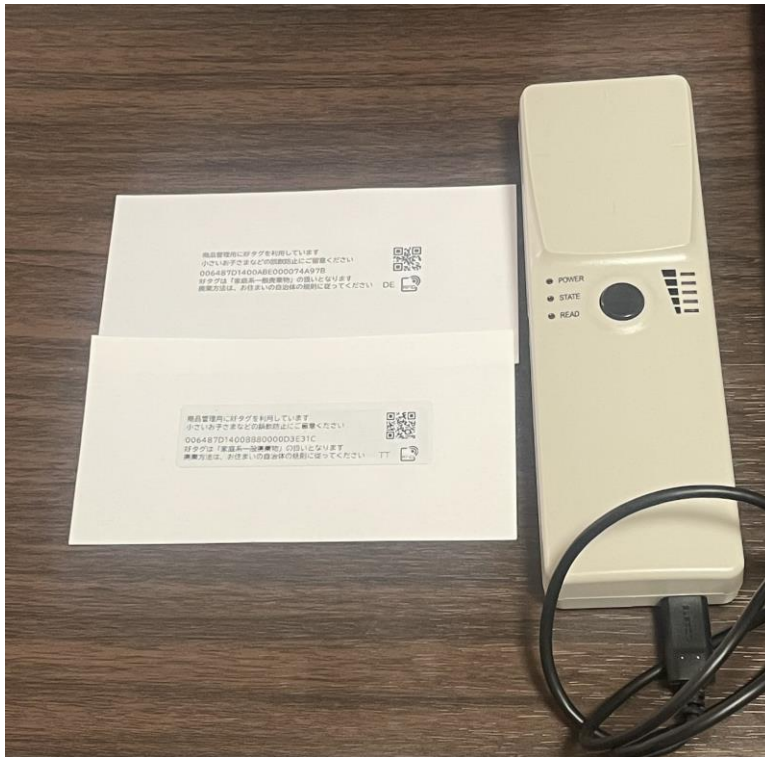
- RFID技術の基礎説明
- RFIDタグとリーダーの操作
- Pythonを用いたRFIDデータの取得と処理
- バックエンドシステムの構築

RFIDタグとリーダーの操作

実践内容

- ✓ リーダとタグ、ソフトの紹介
- ✓ 接続方法
 - USB接続
 - Wi-Fi接続
 - Bluetooth接続
- ✓ コマンドや読み取り方法
- ✓ 読み取ってデータを表示、連続読み取り
- ✓ タグを箱の中に隠し、読み取れるかデモ

リーダーとタグ、ソフトの紹介



接続方法

- **USB接続**

- 今回のメインの接続
- COMポートで接続

- **Wi-Fi接続**

- TCP/IPを通じて通信
- リーダーはクライアントとして接続

- **Bluetooth接続**

- ペアリングを行い、COMポートで接続
- 物理的な障害がなく使いやすい

コマンドや読み取り方法



- コマンド
- 連続インベントリ
- 連続インベントリリード
- 一枚読み取り
- 消込
- マクロ
- InventoryRead

データの表示、連続読み取り

UTRRWManager Ver.1.5.0.0												
ファイル(F) リードライト制御コマンド(V) リードライト設定コマンド(W) RFタグ通信コマンド(R) ツール(T) 設定(Y) 接続(Z) ヘルプ(H)												
コマンド 連続インベントリ 連続インベントリリード 一枚読取 消込 消込登録 マクロ Clear Count Clear 26 ch												
24 dBm Inventory InventoryRead 電波環境 読取設定 リード設定 マクロ2 0 枚/秒 26 枚 COM3												
受信データ一覧 送受信ログ												
No.	データ数	PC/EPC(U11)	RSSI	-min-	-max-	ANGLE	-min-	-max-	ANT	DATA	CHIP	
1	2	34 00 E2 80 11 70 00 00 02 15 80 31 13 F7	-36.7	-64.3	-36.7	64.6	30.9	64.9	0	E2 80 11 70	Monza R6-P(Impinj)	
2	2	34 00 E2 80 11 70 00 00 02 15 80 31 13 F3	-25.7	-48.8	-25.7	146.2	2.8	146.8	0	E2 80 11 70	Monza R6-P(Impinj)	
3	2	34 00 E2 80 11 70 00 00 02 15 80 31 13 F4	-24.7	-47.9	-24.7	33.7	2.8	33.8	0	E2 80 11 70	Monza R6-P(Impinj)	
4	1	34 00 00 64 87 D1 40 0B 88 00 00 D3 E3 1C	-24.7	-24.7	-24.7	135.0	135.0	135.0	0	E2 80 11 90	M750(Impinj)	
5	1	30 00 00 64 87 D1 40 0A BE 00 00 E3 C5 2D	-24.7	-24.7	-24.7	132.1	132.1	132.1	0	E2 80 69 95		
6	1	30 00 E2 80 68 94 00 00 50 27 09 A3 71 3F	-30.7	-30.7	-30.7	171.5	171.5	171.5	0	E2 80 68 94	UCode 8(NXP)	
7	1	30 00 E2 80 68 94 00 00 40 27 09 A3 91 3F	-29.0	-29.0	-29.0	14.0	14.0	14.0	0	E2 80 68 94	UCode 8(NXP)	
8	1	30 00 E2 80 68 94 00 00 50 27 09 A3 99 3F	-25.7	-25.7	-25.7	14.0	14.0	14.0	0	E2 80 68 94	UCode 8(NXP)	
9	1	34 00 E2 80 11 70 00 00 02 15 80 31 13 F0	-23.8	-23.8	-23.8	33.7	33.7	33.7	0	E2 80 11 70	Monza R6-P(Impinj)	
10	1	34 00 00 64 87 D1 40 0D AC 00 00 78 B9 0C	-28.2	-28.2	-28.2	171.5	171.5	171.5	0	E2 80 11 90	M750(Impinj)	
11	1	34 00 00 64 87 D1 40 0B 88 00 00 15 03 70	-30.7	-30.7	-30.7	112.5	112.5	112.5	0	E2 80 11 90	M750(Impinj)	
12	1	34 00 E2 80 11 70 00 00 02 15 80 31 12 FF	-28.2	-28.2	-28.2	70.3	70.3	70.3	0	E2 80 11 70	Monza R6-P(Impinj)	
13	1	34 00 E2 80 11 70 00 00 02 15 80 31 13 F5	-24.7	-24.7	-24.7	47.8	47.8	47.8	0	E2 80 11 70	Monza R6-P(Impinj)	
14	1	34 00 E2 80 11 70 00 00 02 15 80 31 13 F2	-26.3	-26.3	-26.3	151.8	151.8	151.8	0	E2 80 11 70	Monza R6-P(Impinj)	
15	1	34 00 00 64 87 D1 40 0B 88 00 01 50 68 18	-27.5	-27.5	-27.5	16.8	16.8	16.8	0	E2 80 11 90	M750(Impinj)	
16	1	30 00 00 64 87 D1 40 0C B2 00 01 88 41 52	-36.7	-36.7	-36.7	174.3	174.3	174.3	0	E2 80 69 95		
17	1	30 00 00 64 87 D1 40 0A BE 00 00 A1 F2 7A	-27.5	-27.5	-27.5	11.2	11.2	11.2	0	E2 80 69 95		
18	1	34 00 00 64 87 D1 40 0B 88 00 01 76 02 DF	-27.5	-27.5	-27.5	165.9	165.9	165.9	0	E2 80 11 90	M750(Impinj)	
19	1	30 00 E2 80 68 94 00 00 40 27 09 A4 39 3F	-31.7	-31.7	-31.7	165.9	165.9	165.9	0	E2 80 68 94	UCode 8(NXP)	
20	1	34 00 00 64 87 D1 40 0B 88 00 02 59 74 90	-27.5	-27.5	-27.5	2.8	2.8	2.8	0	E2 80 11 90	M750(Impinj)	
21	1	30 00 E2 80 68 94 00 00 50 27 09 A3 39 3F	-32.9	-32.9	-32.9	165.9	165.9	165.9	0	E2 80 68 94	UCode 8(NXP)	
22	1	30 00 E2 80 68 94 00 00 40 27 09 A3 9D 3F	-47.0	-47.0	-47.0	47.8	47.8	47.8	0	E2 80 68 94	UCode 8(NXP)	
23	1	34 00 E2 80 11 70 00 00 02 15 80 31 13 F6	-32.3	-32.3	-32.3	174.3	174.3	174.3	0	E2 80 11 70	Monza R6-P(Impinj)	
24	1	30 00 E2 80 68 94 00 00 50 27 09 A3 95 3F	-44.3	-44.3	-44.3	39.3	39.3	39.3	0	E2 80 68 94	UCode 8(NXP)	
25	1	34 00 E2 80 11 70 00 00 02 15 80 31 13 F1	-27.5	-27.5	-27.5	14.0	14.0	14.0	0	E2 80 11 70	Monza R6-P(Impinj)	
26	1	30 00 E2 80 68 94 00 00 50 27 09 A3 2D 3F	-53.0	-53.0	-53.0	22.5	22.5	22.5	0	E2 80 68 94	UCode 8(NXP)	

送受信ログ

UTRRWManager Ver.1.5.0.0

ファイル(F) リーダライタ制御コマンド(V) リーダライタ設定コマンド(W) RFタグ通信コマンド(R) ツール(T) 設定(Y) 接続(Z) ヘルプ(H)

コマンド 連続インベントリ 連続インベントリリード 一枚読取 消込 消込登録 マクロ Clear Count Clear

24 dBm Inventory InventoryRead 電波環境 読取設定 リーダ設定 マクロ2 0 枚中あと 4

受信データ一覧 送受信ログ

```
10:08:38.089 [cmd] /* リーダライタ動作モードの読み取り */
10:08:38.089 [TX] 02 00 4F 01 00 03 55 0D
10:08:38.168 [RX] 02 00 30 09 00 00 00 10 00 01 03 08 00 03 58 0D
10:08:38.168 [cmd] /* UHF_InventoryRead */
10:08:38.168 [TX] 02 00 55 07 14 02 00 00 00 00 02 03 79 0D
10:08:38.311 [RX] 02 00 6C 19 0A FE 74 40 0E 34 00 00 64 87 D1 40 0B B8 00 01 50 68 18 04 E2 80 11 90 00 03 1F 0D
10:08:38.329 [RX] 02 00 6C 19 0A FE 0E 01 0E 34 00 E2 80 11 70 00 00 02 15 80 31 13 F6 04 E2 80 11 70 00 03 7E 0D
10:08:38.345 [RX] 02 00 6C 19 0A FE 02 3F 0E 34 00 E2 80 11 70 00 00 02 15 80 31 13 F0 04 E2 80 11 70 00 03 AA 0D
10:08:38.354 [RX] 02 00 6C 19 0A FE 8D 40 0E 34 00 00 64 87 D1 40 0B B8 00 00 03 E3 1C 04 E2 80 11 90 00 03 19 0D
10:08:38.370 [RX] 02 00 30 05 14 00 04 00 1A 03 6C 0D
10:08:38.371 [cmt] マクロ実行終了
10:08:40.894 [cmd] /* リーダライタ動作モードの書き込み */
10:08:40.896 [TX] 02 00 4E 07 00 00 00 10 00 00 00 03 6A 0D
10:08:40.977 [RX] 02 00 30 00 03 35 0D
10:08:48.088 [cmd] /* UHF_Inventory */
10:08:48.091 [TX] 02 00 55 01 10 03 68 0D
10:08:48.190 [RX] 02 00 6C 13 09 FE 7B 3B 0E 34 00 00 64 87 D1 40 0B B8 00 01 50 68 18 03 13 0D
10:08:48.199 [RX] 02 00 6C 13 09 FD EE 04 0E 30 00 00 64 87 D1 40 0C B2 00 01 88 41 52 03 90 0D
10:08:48.208 [RX] 02 00 6C 13 09 FE AA 3B 0E 34 00 00 64 87 D1 40 0B B8 00 00 03 E3 1C 03 43 0D
10:08:48.209 [RX] 02 00 6C 13 09 FE 31 00 0E 30 00 00 64 87 D1 40 0A BE 00 00 74 A9 7B 03 56 0D
10:08:48.221 [RX] 02 00 30 05 10 00 04 00 1A 03 68 0D
10:08:50.663 [cmd] /* リーダライタ動作モードの書き込み */
10:08:50.665 [TX] 02 00 4E 07 00 00 00 10 00 00 00 03 6A 0D
10:08:50.746 [RX] 02 00 30 00 03 35 0D
10:08:50.748 [cmd] /* UHF_InventoryRead */
10:08:50.748 [TX] 02 00 55 07 14 02 00 00 00 00 02 03 79 0D
10:08:50.869 [RX] 02 00 6C 19 0A FE 3F 31 0E 34 00 00 64 87 D1 40 0B B8 00 00 03 E3 1C 04 E2 80 11 90 00 03 DC 0D
10:08:50.887 [RX] 02 00 6C 19 0A FE 54 39 0E 34 00 00 64 87 D1 40 0B B8 00 01 50 68 18 04 E2 80 11 90 00 03 F8 0D
10:08:50.931 [RX] 02 00 6C 19 0A FE 3F 00 0E 30 00 00 64 87 D1 40 0A BE 00 00 74 A9 7B 04 E2 80 89 95 00 03 CF 0D
10:08:50.949 [RX] 02 00 6C 19 0A FE 0E 02 0E 30 00 00 64 87 D1 40 0C B2 00 01 88 41 52 04 E2 80 89 95 00 03 1A 0D
10:08:50.967 [RX] 02 00 30 05 14 00 04 00 1A 03 6C 0D
```


読み取りデモ

箱の中にいれたタグを読み取ってみよう
金属や物の間など、タグの性質を体感しよう

授業構成

- RFID技術の基礎説明
- RFIDタグとリーダーの操作
- Pythonを用いたRFIDデータの取得と処理
- バックエンドシステムの構築

Pythonを用いたデータの取得

- ソフトで行っていたこと

- COMポートで接続
- 読み取りなどの命令（バイナリコマンド）
- 出力を表示

- Pythonを用いた実行

- 上3つの内容をコードで行うことができる
- クリックしなくても時間設定や無限ループで行える
- データベースに保存可能
- ほかのサービスを増やすことができる

送受信ログ

UTRRWManager Ver.1.5.0.0

ファイル(F) リーダライタ制御コマンド(V) リーダライタ設定コマンド(W) RFタグ通信コマンド(R) ツール(T) 設定(Y) 接続(Z) ヘルプ(H)

コマンド 連続インベントリ 連続インベントリリード 一枚読取 消込 消込登録 マクロ Clear Count Clear

24 dBm Inventory InventoryRead 電波環境 読取設定 リーダ設定 マクロ2 0 枚中あと 4

受信データ一覧 送受信ログ

```
10:08:38.089 [cmd] /* リーダライタ動作モードの読み取り */
10:08:38.089 [TX] 02 00 4F 01 00 03 55 0D
10:08:38.168 [RX] 02 00 30 09 00 00 00 10 00 01 03 08 00 03 58 0D
10:08:38.168 [cmd] /* UHF_InventoryRead */
10:08:38.168 [TX] 02 00 55 07 14 02 00 00 00 00 02 03 79 0D
10:08:38.311 [RX] 02 00 6C 19 0A FE 74 40 0E 34 00 00 64 87 D1 40 0B B8 00 01 50 68 18 04 E2 80 11 90 00 03 1F 0D
10:08:38.329 [RX] 02 00 6C 19 0A FE 0E 01 0E 34 00 E2 80 11 70 00 00 02 15 80 31 13 F6 04 E2 80 11 70 00 03 7E 0D
10:08:38.345 [RX] 02 00 6C 19 0A FE 02 3F 0E 34 00 E2 80 11 70 00 00 02 15 80 31 13 F0 04 E2 80 11 70 00 03 AA 0D
10:08:38.354 [RX] 02 00 6C 19 0A FE 8D 40 0E 34 00 00 64 87 D1 40 0B B8 00 00 03 E3 1C 04 E2 80 11 90 00 03 19 0D
10:08:38.370 [RX] 02 00 30 05 14 00 04 00 1A 03 6C 0D
10:08:38.371 [cmt] マクロ実行終了
10:08:40.894 [cmd] /* リーダライタ動作モードの書き込み */
10:08:40.896 [TX] 02 00 4E 07 00 00 00 10 00 00 00 03 6A 0D
10:08:40.977 [RX] 02 00 30 00 03 35 0D
10:08:48.088 [cmd] /* UHF_Inventory */
10:08:48.091 [TX] 02 00 55 01 10 03 68 0D
10:08:48.190 [RX] 02 00 6C 13 09 FE 7B 3B 0E 34 00 00 64 87 D1 40 0B B8 00 01 50 68 18 03 13 0D
10:08:48.199 [RX] 02 00 6C 13 09 FD EE 04 0E 30 00 00 64 87 D1 40 0C B2 00 01 88 41 52 03 90 0D
10:08:48.208 [RX] 02 00 6C 13 09 FE AA 3B 0E 34 00 00 64 87 D1 40 0B B8 00 00 03 E3 1C 03 43 0D
10:08:48.209 [RX] 02 00 6C 13 09 FE 31 00 0E 30 00 00 64 87 D1 40 0A BE 00 00 74 A9 7B 03 56 0D
10:08:48.221 [RX] 02 00 30 05 10 00 04 00 1A 03 68 0D
10:08:50.663 [cmd] /* リーダライタ動作モードの書き込み */
10:08:50.665 [TX] 02 00 4E 07 00 00 00 10 00 00 00 03 6A 0D
10:08:50.746 [RX] 02 00 30 00 03 35 0D
10:08:50.748 [cmd] /* UHF_InventoryRead */
10:08:50.748 [TX] 02 00 55 07 14 02 00 00 00 00 02 03 79 0D
10:08:50.869 [RX] 02 00 6C 19 0A FE 3F 31 0E 34 00 00 64 87 D1 40 0B B8 00 00 03 E3 1C 04 E2 80 11 90 00 03 DC 0D
10:08:50.887 [RX] 02 00 6C 19 0A FE 54 39 0E 34 00 00 64 87 D1 40 0B B8 00 01 50 68 18 04 E2 80 11 90 00 03 F8 0D
10:08:50.931 [RX] 02 00 6C 19 0A FE 3F 00 0E 30 00 00 64 87 D1 40 0A BE 00 00 74 A9 7B 04 E2 80 89 95 00 03 CF 0D
10:08:50.949 [RX] 02 00 6C 19 0A FE 0E 02 0E 30 00 00 64 87 D1 40 0C B2 00 01 88 41 52 04 E2 80 89 95 00 03 1A 0D
10:08:50.967 [RX] 02 00 30 05 14 00 04 00 1A 03 6C 0D
```

送受信ログ

InventoryReadの送受信ログ

```
10:10:56.801 [cmd] /* UHF_InventoryRead */  
10:10:56.804 [TX] 02 00 55 07 14 02 00 00 00 00 02 03 79 0D  
10:10:56.920 [RX] 02 00 6C 19 0A FE AA 14 0E 34 00 00 64 87 D1  
40 0B B8 00 01 50 68 18 04 E2 80 11 90 00 03 29 0D  
10:10:56.937 [RX] 02 00 30 05 14 00 01 00 1A 03 69 0D
```

TX(Transmit)

- リーダーが送信したコマンド

RX(Receive)

- リーダーが受信したデータ

Pythonを用いたデータの取得

```
import serial

# シリアルポートを開く
ser = serial.Serial(
    port='COM3', # 使用しているポート名に変更 (WindowsではCOMポート、Linux/macOSでは/dev/ttyUSBxなど)
    baudrate=115200, # ボーレート
    timeout=1 # タイムアウト (必要に応じて調整)
)

# バイナリデータの送受信
binary_data = b'¥x02¥x00¥x55¥x07¥x14¥x02¥x00¥x00¥x00¥x00¥x02¥x03¥x79¥x0D' # ログに対応するデータ
# データを送信
ser.write(binary_data)

# データを受信 (32バイトを読み取る場合)
response_3 = ser.read(32) # 32バイト分のデータを受信
# 受信データを16進数形式で連続表示 (空白なし)
hex_string = response_3.hex() # 16進数の文字列に変換
print("Received data:", hex_string)

# ポートを閉じる
ser.close()
```

Pythonを用いたデータの取得

Pythonでの実行結果

```
C:\Users\ryou2\Documents\RFID\RFID_python>python a.py  
Received data:  
02006c190aff1b2f0e3400006487d1400bb8000150681804e280  
11900003b60d
```

ソフトでの読み取り結果

```
10:10:56.801 [cmd] /* UHF_InventoryRead */  
10:10:56.804 [TX] 02 00 55 07 14 02 00 00 00 00 02 03 79 0D  
10:10:56.920 [RX] 02 00 6C 19 0A FE AA 14 0E 34 00 00 64 87 D1  
40 0B B8 00 01 50 68 18 04 E2 80 11 90 00 03 29 0D  
10:10:56.937 [RX] 02 00 30 05 14 00 01 00 1A 03 69 0D
```

様々なバイナリデータ

InventoryRead以外にもほかのコマンドを試してみよう！
コードを改良して読み取りの無限ループをしてみよう！
複数のタグを表示できるようにしてみよう！

ヒント

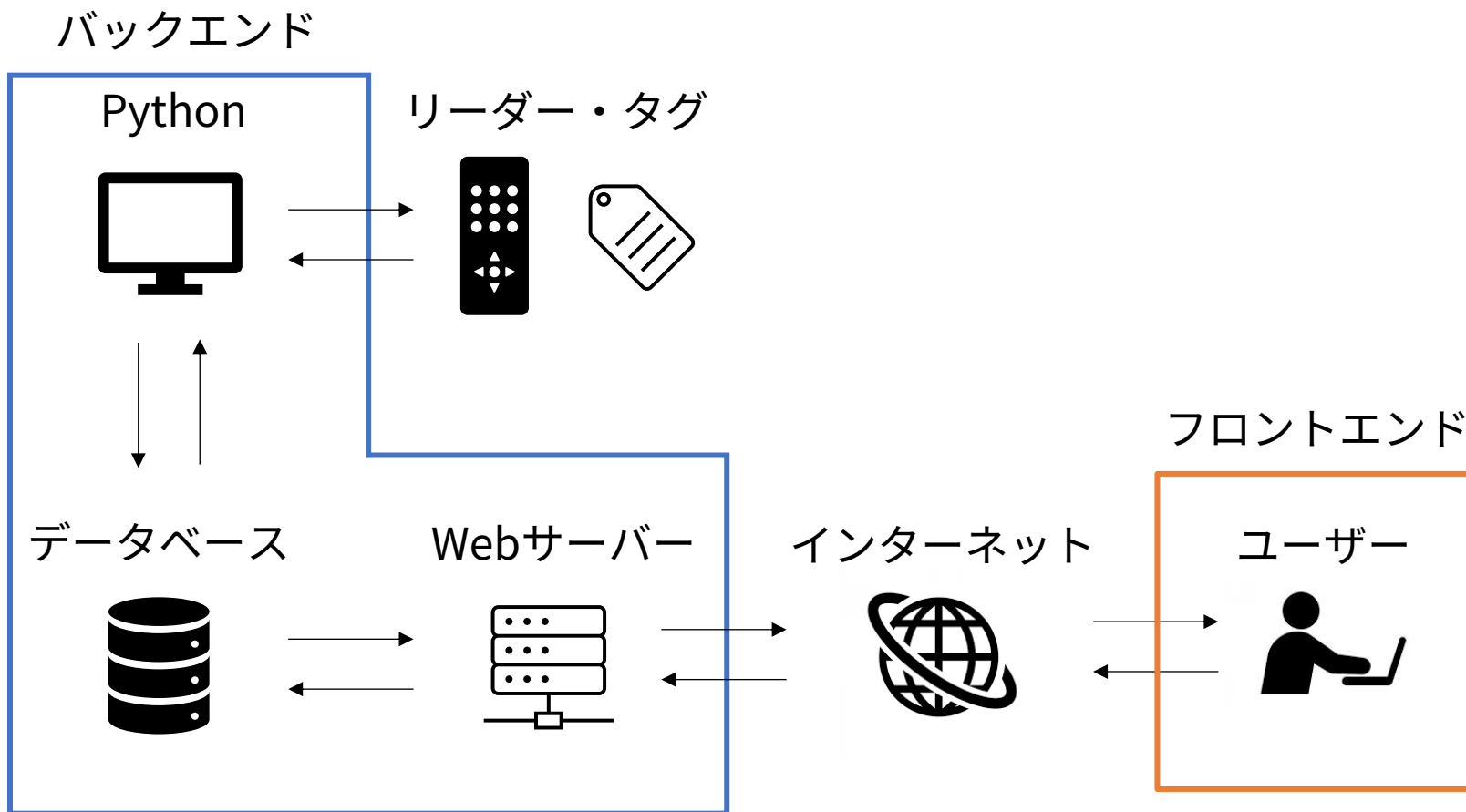
欲しい部分の出力は64文字で、タグのIDは22番目から46番目に書かれている

```
10:10:56.920 [RX] 02 00 6C 19 0A FE AA 14 0E 34 00 00 64 87 D1  
40 0B B8 00 01 50 68 18 04 E2 80 11 90 00 03 29 0D
```


授業構成

- RFID技術の基礎説明
- RFIDタグとリーダーの操作
- Pythonを用いたRFIDデータの取得と処理
- バックエンドシステムの構築

システムの概要



システムの概要

倉庫管理システム

ID	existence
006487d1400abe000074a97b	True
006487d1400cb20001884152	False
006487d1400bb80000d3e31c	True
006487d1400bb800017602df	False

システム概要

- バックエンド
 - タグの読み取り
 - タグの情報をデータベースに保存
 - 貸し出し返却の表示
- フロントエンド
 - データベースの表示
 - True,Falseの表示

RFID_system

RFID.py

```
while True:

    print("READ")

    # RFIDへのREAD命令
    RFIDread(ser)

    # 受信データの読み取り
    rcv_data_list = serialReadLines(ser)

    # 全てのデータの取り出し
    for rcv_data in rcv_data_list:

        # 読み取ったデータのIDがすでに存在するか
        check_str = df[df["ID"].isin([rcv_data])]
        # print("CHECK_STR")
        # print(check_str)
        # print(check_str[1])

        # rcv_dataで得たIDがデータベースに存在しない場合
        if(check_str.empty):
            # IDをデータベースに登録
            new_data = pd.DataFrame({"ID": [rcv_data], "existence": [True]})
            df = pd.concat([df, new_data], ignore_index=True)
            print("ID: " + rcv_data + "をデータベースに登録しました。")

        # rcv_dataで得たIDがデータベースに存在する場合
        # 中でも倉庫内に保管されている場合
        elif(check_str.iloc[0]["existence"] == True):
            # 指定IDの保管データをFalseに(貸し出し処理)
            df.loc[df["ID"] == rcv_data, "existence"] = False
            print("ID: " + rcv_data + "を貸し出しました。")
```

RFID_system

Index.html

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>在庫管理</title>
  <link rel="stylesheet" href="style.css">
  <script>
    // 5秒ごとにページをリロード(表の更新のため)
    setTimeout(function() {
      window.location.reload();
    }, 5000);
  </script>
</head>
<body>
  <div class="container">
    <h1>倉庫管理システム</h1>
    {{ table | safe }}
  </div>
</body>
</html>
```

さらなるシステム概要

- タイムスタンプの追加
- 状態を比較して返却・貸出の出力
- 音声出力

RFID_system_voice

RFID.py

```
# 2. すべての existence を False に設定し、読み取ったデータを True に設定
df['existence'] = False
for rcv_data in rcv_data_list:
    current_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    check_str = df[df["ID"].isin([rcv_data])]

    if check_str.empty:
        # 新しいタグの場合、True に設定して追加
        new_data = pd.DataFrame({
            "ID": [rcv_data],
            "existence": [True],
            "previous_existence": [False], # 初期値は False
            "timestamp": [current_time]
        })
        df = pd.concat([df, new_data], ignore_index=True)
        print("ID: " + rcv_data + " をデータベースに登録しました (時間: " + current_time + ") 。")
    else:
        # 既存のタグの場合、True に設定
        df.loc[df["ID"] == rcv_data, ["existence", "timestamp"]] = [True, current_time]

# 3. previous_existence と existence を比較して、状態が変わった場合に音声で通知
for index, row in df.iterrows():
    if row['previous_existence'] == True and row['existence'] == False:
        print(f"ID: {row['ID']} を貸し出しました (時間: {row['timestamp']}) 。")
        speak(row['ID'][-4:] + "が貸し出しされました")
    elif row['previous_existence'] == False and row['existence'] == True:
        print(f"ID: {row['ID']} を返却しました (時間: {row['timestamp']}) 。")
        speak(row['ID'][-4:] + "が返却されました")
```


さらなるシステム概要

- フロントエンドでの検索機能
- フロントエンドでの音声出力
- IDと商品名の連携などなど

まだまだできることはたくさんあります！

授業構成

- RFID技術の基礎説明
- RFIDタグとリーダーの操作
- Pythonを用いたRFIDデータの取得と処理
- バックエンドシステムの構築