

7Q-6

C#による FeliCa ライブラリ

原田 和貴[†] 齋藤 祐輔[‡] 並木 美太郎[§]

東京農工大学工学部情報コミュニケーション工学科[†]

東京農工大学大学院工学教育部情報コミュニケーション工学専攻[‡] 東京農工大学大学院共生科学技術研究部[§]

1 はじめに

近年 RFID は様々な場所で使われてきている。その中でも FeliCa は Suica や Edy などの電子マネーとして使われたり、携帯に搭載され、携帯のアプリケーションと連携した様々なサービスに使われたりしている。

FeliCa が普及してくるにつれて、自分が開発するソフトウェアに FeliCa を用いたものを作りたいという要望が出てくる。しかし、FeliCa にはエリア、サービスといったデータ管理やデータへのアクセス方式や、ブロックといったデータ格納領域など独自の仕組みが多く、仕様の理解に手間取る。また、複数の開発者がカードを用いる場合必ずいつも同じエリアコードを用いることができないため、自分の管理する領域を見つける工夫を開発者が行う必要がある。そこで本発表では、カードのアクセスを通常のディレクトリ名、ファイル名を用いて扱えるライブラリの作成を行った。

使用する言語として、C#を選択した。C#や Java には入出力を統一して行うシステムとして Stream が存在する。Stream に沿う形で作成することによりファイルと同じように扱うことができる。また、C#は.NETに対応しているため、.NET に対応した他の言語で利用することが可能である。

2 目標

FeliCa を用いたソフトウェアを開発する際に障害となるエリア、サービスといった独自の仕組みを隠蔽し、ディレクトリやファイルに置き換えて扱えるようにする。また、ブロックの管理を考えずに行えるようにして、C#の Stream に沿った形で FeliCa を扱え、また他の言語でも使えるようなライブラリの作成を目指す。

3 全体構成

全体の構成は図 1 のようになる。

FeliCa Control クラスでは既存のライブラリを機能ごとに処理をまとめたものを提供する。このクラスでは他のクラスから実際に FeliCa へのアクセスを行うための機能を有するとともに、FeliCa 独自仕様によるエリア、サービスコードを用いたアクセス方法も提供する。

Directory クラスではディレクトリ管理を行う。ディレクトリの作成や削除の他、ディレクトリ下のファイルの参照などを行う。エリアコードとディレクトリ名の対応の管理はこのクラスで行い、ディレクトリ管理情報を格納したブロックにアクセスして読書きを行う。

File クラスではファイル管理を行う。ファイルの作成、削除の他、ファイルの属性、サイズや更新日時取得などを行う。サービスコードとファイル名の対応

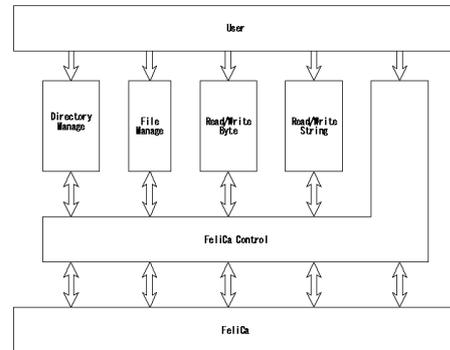


図 1: 全体の構成

の管理はこのクラスで行い、ディレクトリごとに用意されたファイル管理情報を格納したブロックにアクセスして読書きを行う。

Read Byte クラスはバイト列の読み込みを、Write Byte クラスではバイト列の書き込みを行う。このクラスではブロックではなくサイズを指定することにより、バイト列の読書きを行う。

Read/Write String クラスでは Read/Write Byte クラスと同様に文字列の読書きを行う。なお Read/Write Byte/String クラスでは従来の FeliCa の入出力仕様も提供している。

4 ライブラリの設計

ライブラリは図 2 のようなクラス構成になる。

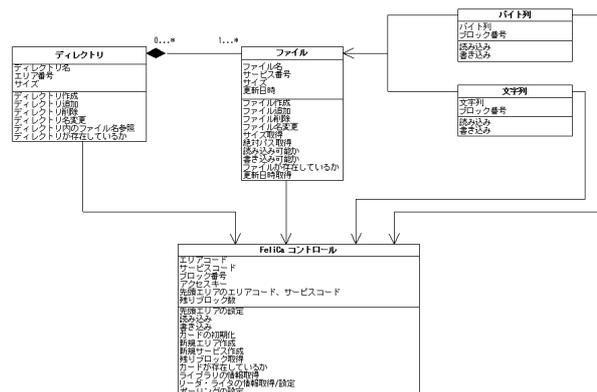


図 2: クラス図

4.1 FeliCa コントロールクラス

FeliCa コントロールでは、実際に FeliCa カードとのアクセスを行うためのメソッドを用意する。ディレクトリ管理情報を格納したエリア（以下先頭エリア）の設定や読み込み、書き込み、カードの初期化、新規エリア作成、新規サービス作成、各種情報の取得、設定を行うメソッドが存在する。

C# Library for FeliCa

[†] Kazutaka Harada

Department of Computer, Information and Communication Sciences, Tokyo University of Agriculture and Technology

[‡] Yusuke Saito

Department of Computer, Information and Communication Sciences, Tokyo University of Agriculture and Technology

[§] Mitaro Namiki

Department of Computer, Information and Communication Sciences, Tokyo University of Agriculture and Technology

4.2 ディレクトリ管理

ディレクトリ管理には、ディレクトリの作成、削除、既存のエリアに名前をつける追加、名前変更、参照ディレクトリ内のファイル名取得、ディレクトリが存在しているかの確認を行うメソッドが存在し、ディレクトリ操作を行う。

ディレクトリ管理では FeliCa コントロールクラスのメソッドを用いて処理を行う。ディレクトリの作成や削除はカードの初期化や新規エリア作成を用いる。ただし、カードの初期化はカード全てを初期化するため、カード全体を管理するエリア 0000 にアクセスする権限必要となる。権限が無い場合はディレクトリの管理情報を書き換えることにより削除を行う。また、この方法で削除されたディレクトリは新たにディレクトリを作成するときに再利用される。追加や名前変更はディレクトリ管理情報を書き換えることにより行う。ディレクトリ内のファイル名は、ファイルの管理情報を読み込むことにより行う。

4.3 ファイル管理

ファイル管理には、ファイルの作成、削除、既存サービスに名前をつける追加、名前変更、サイズ取得、絶対パス取得、読み込みが可能か、書き込みが可能かの確認、ファイルの存在確認、更新日時を取得を行うメソッドが存在し、ファイル操作を行う。

ファイル管理では FeliCa コントロールクラスのメソッドを用いて処理を行う。ファイル作成、削除、追加や名前の変更はディレクトリ管理と同様に行う。サイズや読み込み可能か、書き込み可能かの確認、ファイルの存在確認、更新日時の取得は管理情報を読み込むことにより行う。絶対パスは呼び出しもとから取得する。なお相対パスに関しては、本ライブラリではディレクトリの下にはファイルしか置けない仕様としたので、扱わない。

4.4 バイト列読書き

バイト列読み込みには、バイト列の読み込みを行うメソッドが、バイト列書き込みには、バイト列を書込むメソッドが存在する。

バイト列読書きは、FeliCa コントロールクラスのメソッドを用いて処理を行う。ファイル名を用いる場合は、管理情報を読み込み、エリアコード、サービスコードに変換してから実際のデータの読書きを行う。サービスコード、エリアコードを用いる場合はそのまま実際の読書きを行う。このクラスの読書きはサイズを指定して行うが、指定されたサービスのサイズが足りない場合は最後のブロックに次のサービスコードを記述することにより連続したデータとして読書きを行う。

4.5 文字列読書き

文字列読み込みには、文字列の読み込みを行うメソッドが、文字列書き込みには、文字列を書込むメソッドが存在する。文字列列読み書きも、バイト列読書きと同様の動作を行う。

4.6 ディレクトリ、ファイル管理

FeliCa ではエリアコード、サービスコード、ブロック番号によってデータを特定している。エリアコードでは使用可能なブロック数と権限などが定義されている。サービスコードではアクセスする方法とアクセスするブロックが定義されている。ブロックはデータを格納する領域である。

エリアコードをディレクトリ、サービスコードをファイルとみなすことにより通常のファイルと同じように扱うことができるようにした。

カード内のデータ管理は図 3 のとおりである。

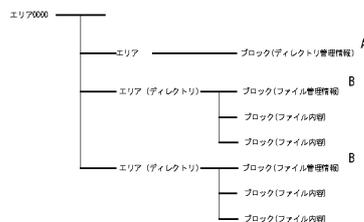


図 3: カード内データ管理

保存方法としては、他の方法も考えられるがカードのメモリを節約し、かつ通常のディレクトリ、ファイルと同じようにアクセスするためにはこの方法が良い。

図 3 の A のディレクトリの管理情報としては、次のものをブロック内に記録する。

```
ディレクトリ名 エリア番号 サイズ
```

図 3 の B のファイルの管理情報としては、次のものをブロック内に記録する。

```
ファイル名 サービス番号 サイズ 属性 更新日時
```

4.7 アクセス例

アクセス例を次に示す。

4.7.1 ディレクトリ作成

```
Open(byte[] id);
SetKey(byte[] key, int type);
CreateDirectory(string directory_name, int size);
Close(byte[] id);
```

まず先頭エリアを含む ID を引数としてオープンする、次にエリア 0000 の鍵を設定、そして、ディレクトリ名とディレクトリの大きさを引数にして作成を行う。処理が終了したらクローズする。

4.7.2 ファイルをバイト列で読み込み

```
Open(byte[] id);
ReadByte(string path, byte[] dat);
Close(byte[] id);
```

まず先頭エリアを含む ID を引数としてオープンする、次にディレクトリ名/ファイル名を引数にして読み込みを行う。処理が終了したらクローズする。読み込むファイルが暗号化されている場合は、読み込みの前に SetKey で暗号化の鍵を設定する。

5 おわりに

本論文では FeliCa を他のデバイスと同じように扱うようにするための、ライブラリの作成について述べた。

FeliCa の独自仕様による開発者に負担を軽減し、ファイルアクセスのインタフェースにすることによりエリアコード、サービスコードを固定する必要がなく柔軟に管理することができる。

参考文献

- [1] ソニー株式会社 SDK for FeliCa Version 1.21
- [2] ハーバート・シルト著、株式会社スリー・イー・システムズ訳 独習 C# 株式会社翔泳社 (2002)
- [3] 河合昭男著 まるごと図解最新 UML がわかる 株式会社技術評論社 (2002)
- [4] <http://www.microsoft.com/japan/msdn/library/>