

平成 22 年度  
卒業研究報告

研究題目

IC カードリーダーを使用した電子錠の製作

指導教員

由井 四海

著者

篠川 尚吾

武脇 大樹

平成 23 年 3 月 3 日提出

独立行政法人国立高等専門学校機構  
富山高等専門学校 電子制御工学科

1.	はじめに .....	1
2.	概略.....	1
3.	IC カードリーダ .....	2
3. 1	IC カードとは .....	2
3. 2	IC カードリーダとは.....	2
4.	システム構成に使用した装置.....	4
4. 1	LabVIEW によるプログラミング.....	4
4. 2	RS-232C でデータを通信.....	4
4. 3	レベルコンバータによる変圧 .....	5
4. 4	無線 LAN による遠距離通信.....	5
5.	システム構成.....	7
5. 2	登録者リストとの比較 .....	10
5. 3	電子錠へ命令を送信.....	12
5. 4	IC カードリーダ認識システム .....	15
6.	補助システム .....	16
6. 1	IC カードの情報を登録者リストへ保存.....	16
6. 2	入退出時の時刻を保存 .....	18
7.	電子錠の製作 .....	20
7. 1	SolidWorks の概要.....	20
7. 2	設計図の作成.....	20
7. 3	電子錠の製作.....	21
8.	ステッピングモータ .....	23
8. 1	ステッピングモータの静特性 .....	23
8. 2	ステッピングモータの動特性 .....	25
9.	ステッピングモータ駆動プログラムの作成.....	27
9. 1	MAPLAB の概要.....	27
9. 2	PIC18F2550.....	27
9. 3	PICKIT2 .....	28
10.	ステッピングモータ正・逆転プログラムの作成 .....	30
11.	電子錠のプログラム.....	34
11. 1	可変抵抗 .....	34
11. 2	位置制御システム.....	35
11. 3	駆動プログラム.....	37
11. 4	電子錠の特徴 .....	38
13.	まとめ .....	39
14.	付録.....	40

## 1. はじめに

今日ICカードは携帯電話に内蔵されるようにもなり世間に広く浸透した。ICカードを使用したサービスの1つに入退室管理システムがある。携帯電話をかざすことで解錠・施錠が可能であり、入退室した個人も把握できることから利便性とセキュリティーの向上を実現している。だが、取り付け工事が必要とされるため、即在の建物に普及していない。そこで、後付け設置することが可能な、ICカードを使用した電子錠の製作に取り組んだ。

## 2. 概略

ICカードをICカードリーダーにかざし、読み取った情報と登録してある情報が一致した場合電子錠が動作する。

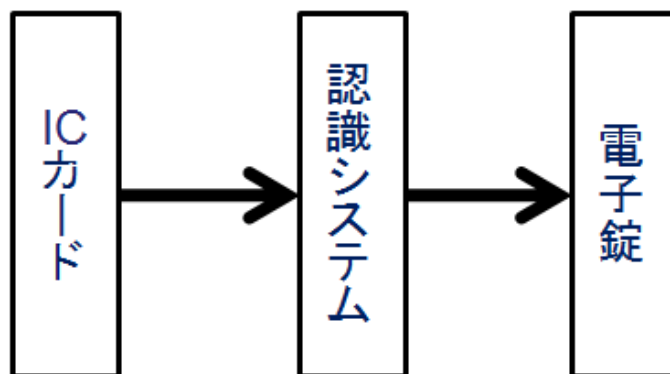


図2. 1 本研究の大きな流れ

### 3. ICカードリーダー

#### 3. 1 ICカードとは

ICカードとはプラスチック製カードに極めて薄いICチップを埋め込み、情報を記録できるようにしたカードのこと。電子マネーやテレホンカードなどに応用されている。磁気カードに比べて100倍近いデータを記録でき、データの暗号化も可能なため偽造にも強い。

本研究で使用するICカードはソニー社製のFeliCaという非接触式ICカードを使用した。FeliCaは誰もが持っている携帯電話のほとんどに内蔵されているため、誰もが所持していることに注目し採用した。FeliCaの特徴について以下に示す。

- ・非接触ICカード技術
- ・不揮発メモリと無線通信チップを内蔵した樹脂製のカード
- ・読み取り端末にかざすだけでデータのやり取りが可能
- ・高速でデータ送受信できる
- ・偽造・変造がしにくいようセキュリティ機構も備えている
- ・ISO/IEC 15408 EAL4標準の認証を受けている
- ・電子マネーや公共交通機関のプリペイドカード、おサイフケータイなどに使われている

#### 3. 2 ICカードリーダーとは

ICカードリーダーとは、ICカードに格納された情報を読むための機器のこと。ICカードリーダーライターとも言う。ICカードの読み込みには、ソニー社製のICカードFeliCa専用読み取り・書き込み機であるPaSoRi、型番RC-S330を使用した。PaSoRiはパソコンに外付けするタイプのFeliCaポートであり、パソコンにはUSBで接続す

る。プレイステーション 3 で使用することもできる。全て型番が FeliCa に対応し、RC-S330 以降は ISO/IEC 14443 (MIFARE) の TypeA・TypeB、これらとの下位互換性をもつ NFC に対応している。

ISO/IEC 14443 とは小電力 IC 通信技術 (RFID) の国際規格のこと。TypeA・TypeB・TypeC がある。NFC とは NXP セミコンダクターズ社とソニーが開発した 13.56MHz 帯の近距離無線通信規格のこと。赤外線通信や Bluetooth、FeliCa、ISO 14443 (MIFARE)、ISO/IEC 18092 などがある。

#### 4. システム構成に使用した装置

システム構成に使用した装置は以下の通り

- LabVIEW
- RS-232C
- レベルコンバータ
- 無線 LAN

##### 4. 1 LabVIEW によるプログラミング

プログラムの構築にはNATIONAL INSTRUMENTS社製のLabVIEWで行った。LabVIEWはグラフィック型言語によってプログラミングすることのできる開発環境である。LabVIEWの特徴について以下に示す。

- G言語と呼ばれるグラフィカルなプログラミング言語
- 関数をテキストではなくアイコンで表す特徴がある
- 他のハードウェアとの協調が可能
- 高度な解析および信号処理が可能

##### 4. 2 RS-232C でデータを通信

IC カードリーダー認識システムと電子錠の通信のやり取りに RS-232C を使用した。RS-232C とは米国電子工業会によって標準化された、シリアル通信方式のインターフェースの1つである。LabVIEW からのデータを RS-232C に変換する USB 変換ケーブルを使用しデータ転送を行う。RS232C のコネクタピン配列を図 4. 1 に示す。本研究では入力に 3 番ピン、GND に 5 番ピンの 2 つを使用した。

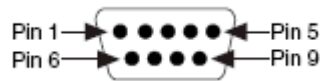


図 4. 1 RS232C のコネクタピン配列

#### 4. 3 レベルコンバータによる変圧

コンピュータの信号電圧は $-5V \sim +5V$ で、受け取り側である電子錠の扱える電圧 $0V \sim +5V$ であるため、変圧するためにレベルコンバータを使用した。レベルコンバータとは信号電圧変換装置のこと。レベルコンバータは Sipex 社製の SP233ACP を使用。SP233ACP の回路図は図 4. 2 に示す。

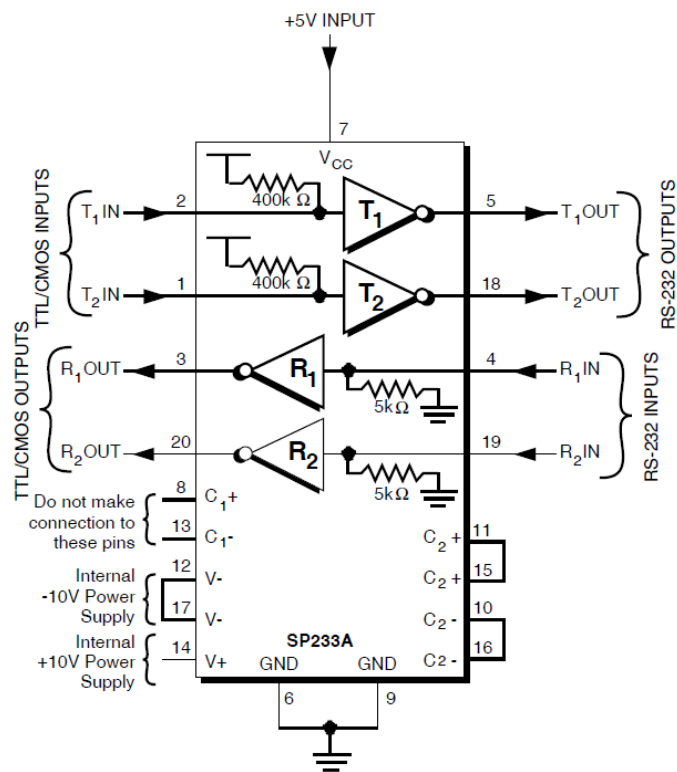


図 4. 2 SP233ACP の回路図

#### 4. 4 無線 LAN による遠距離通信

IC カードリーダーや電子錠はドアに付いているが、パソコンはドアのそばに置くわけ

にはいかない。そのためドアとパソコンを繋ぐケーブルが必要となる。しかし、有線ケーブルでは長くなり邪魔になる。そこでドアとパソコンの遠距離通信を行うために「無線 LAN」を使用した。主に IC カードリーダーとパソコン、パソコンと RS-232C の通信を行うために使用した。



## 5. システム構成

認識システムの全体像について図5.1に示す。ここでは図5.1について詳しく解説する。

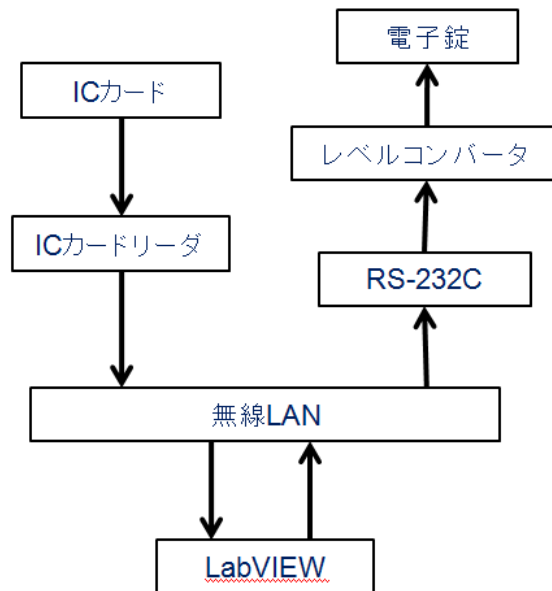


図5.1 認識システムの全体像

### 5. 1 ICカードのデータを取得

ICカードリーダーで取得したICカードはUSBを使って無線LANを経由しPCに送信される。受信したICカードのデータはfeliclibをLabVIEW上で起動して取り込む。feliclibとはUSB接続のFeliCaリーダーPaSoRを使って、FeliCaにアクセスするためのオープンソフトウェアのライブラリ。取り込んだデータのうち、必要なデータだけを抽出して1つの配列に整理する。整理したデータを“3. 2登録者リストとの比較”へと送る。ICカードのデータを取得するために製作したプログラムを図5. 2に、図5. 2のフローチャートを図5. 3に示す。

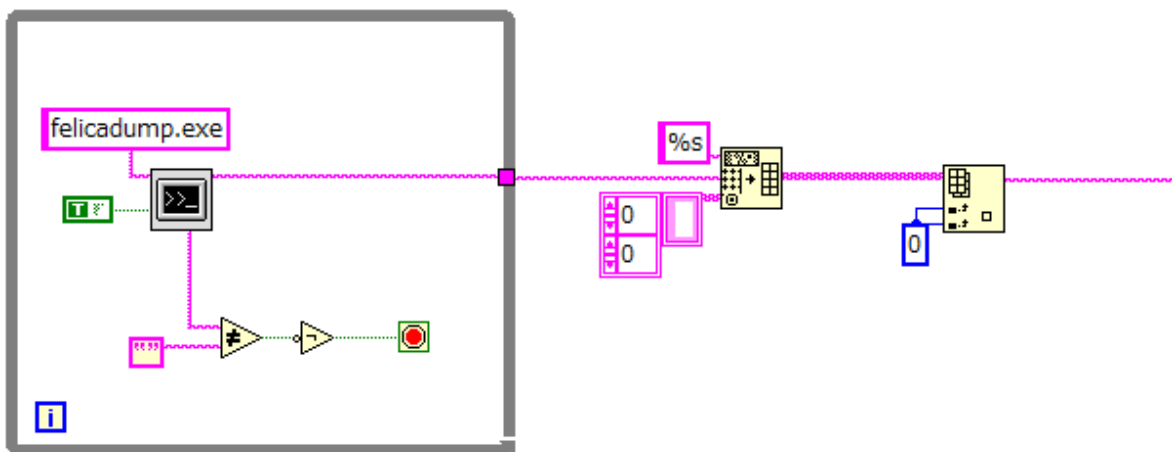


図5. 2 ICカードのデータを取得(プログラム)

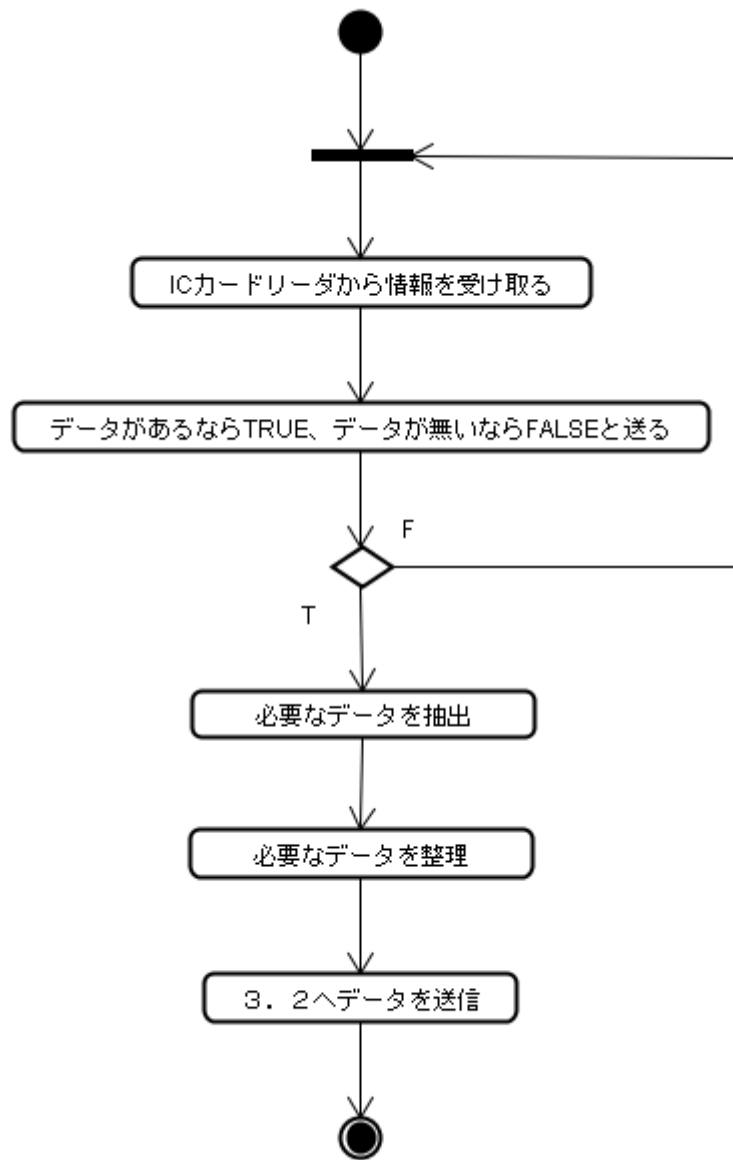


図5. 3 ICカードのデータを取得(フローチャート)

## 5. 2 登録者リストとの比較

ここでは登録されているリストに5. 1で読み込んだデータと一致しているかを判断する。用意してある登録者リストを LabVIEW に取り込む。このままでは3. 1のデータと比較できないので5. 1のデータと同じ配列にデータを変換する。3. 1で読み込んだデータと登録者リストを比較し、一致か登録者リストの最後のデータならば TRUE、一致しなければデータなら FALSE とする。FALSE なら登録者リストの次のデータと比較し、TRUE ならループを停止。ループが停止したらデータが一致していれば TRUE、一致していなければ FALSE と“3. 3電子錠へ命令を送信”に送る。登録者リストとの比較をするために製作したプログラムを図5. 4に、図5. 4のフローチャートを図5. 5に示す。

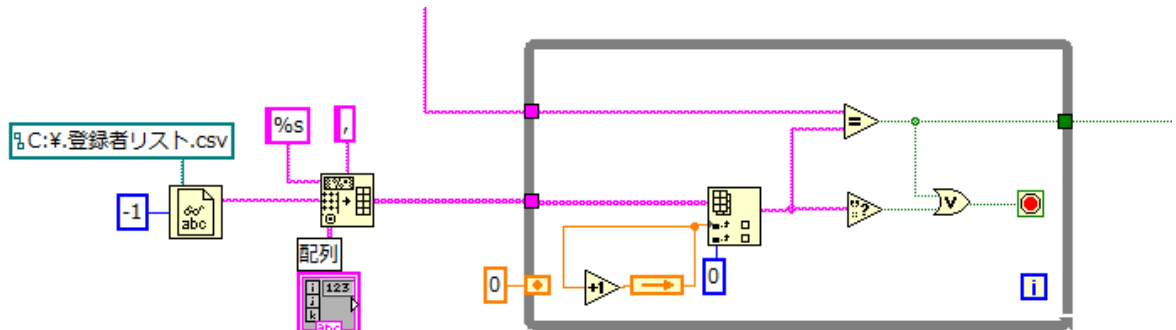


図5. 4 登録者リストとの比較(プログラム)

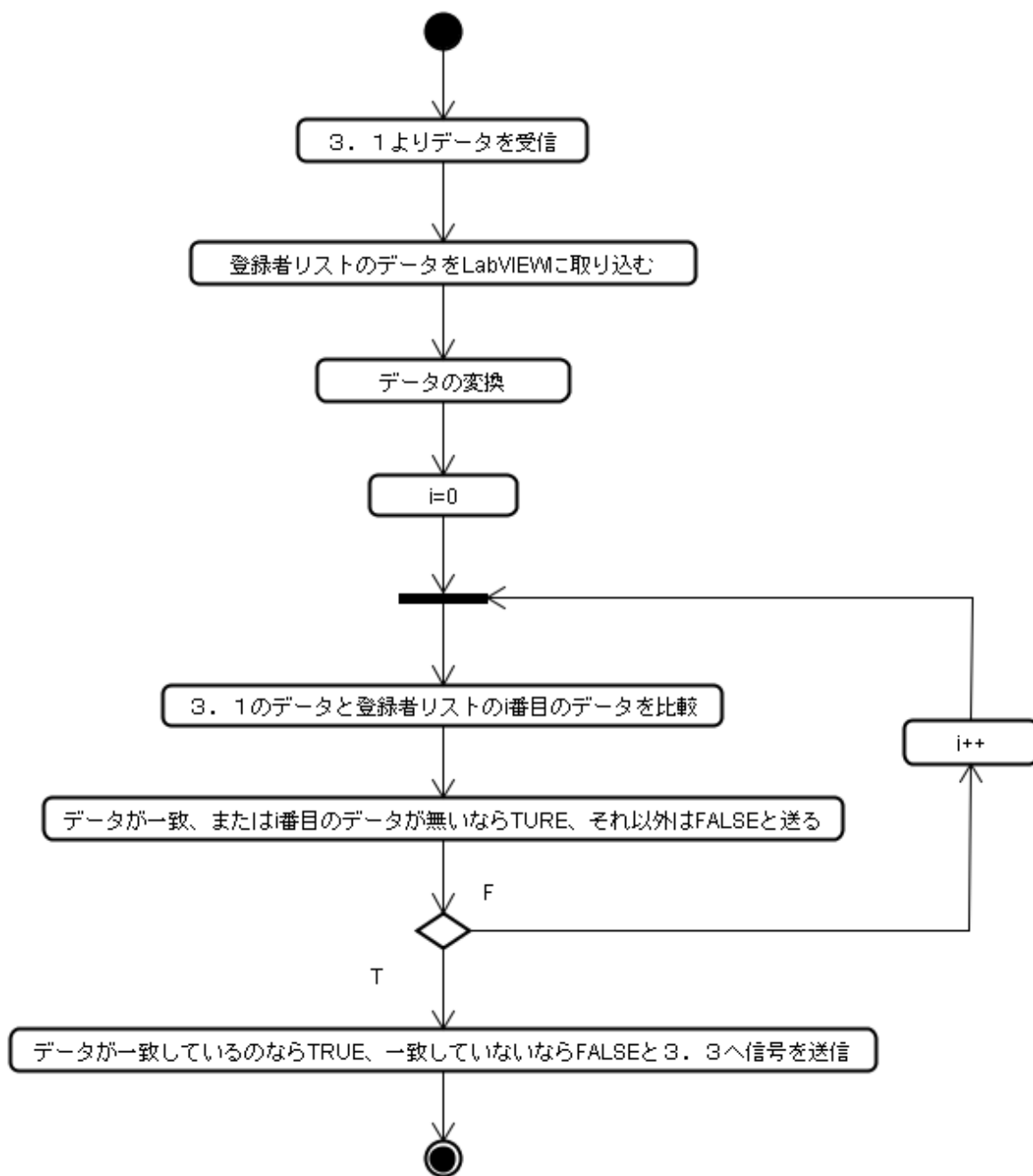


図5. 5 登録者リストとの比較(フローチャート)

### 5. 3 電子錠へ命令を送信

5. 2から受け取った判断が TRUE だった場合 USB より信号を送る。FALSE なら 5. 1に戻る。このプログラムは LabVIEW のサンプルプログラム“Basic Serial Write and Read.vi”を手本に製作した。ここで送信した信号は無線 LAN を経由し USB 変換ケーブルを使用して RS-232C に変換。しかしこのままではコンピュータの信号電圧は  $-5V \sim +5V$  で、受け取り側である電子錠の扱える電圧  $0V \sim +5V$  であるため、電子錠が壊れてしまう。そこでこの間にレベルコンバータを使用した変圧回路を挟む。これによってコンピュータの信号電圧  $-5V \sim +5V$  を  $0V \sim +5V$  に変圧する。変圧した波形は反転してしまうが、ここで必要なのは波形の有無だけなので問題無い。RS-232C を使用して信号を送信するために製作したプログラムを図 5. 6 に、レベルコンバータで変圧するのに使用した回路を図 5. 7 に、図 5. 6 および図 5. 7 のフローチャートを図 5. 8 に示す。

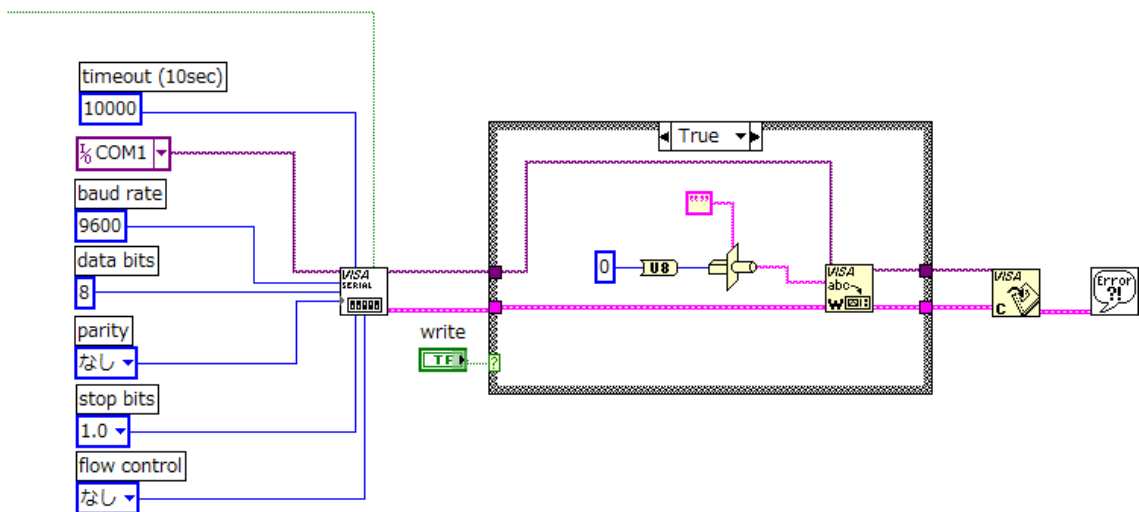


図 5. 6 RS-232C を使用して信号を送信(プログラム)

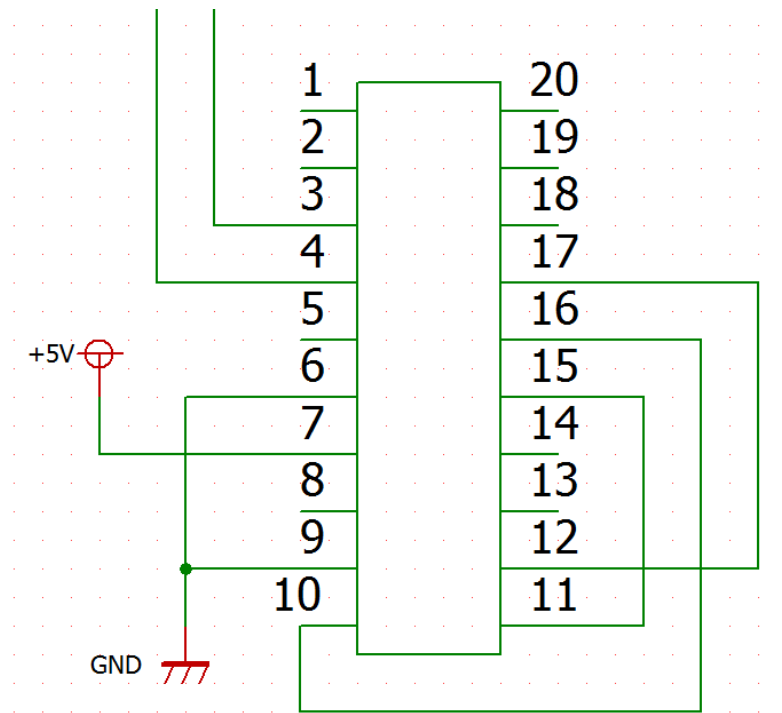


図 5. 7 変圧回路の回路図

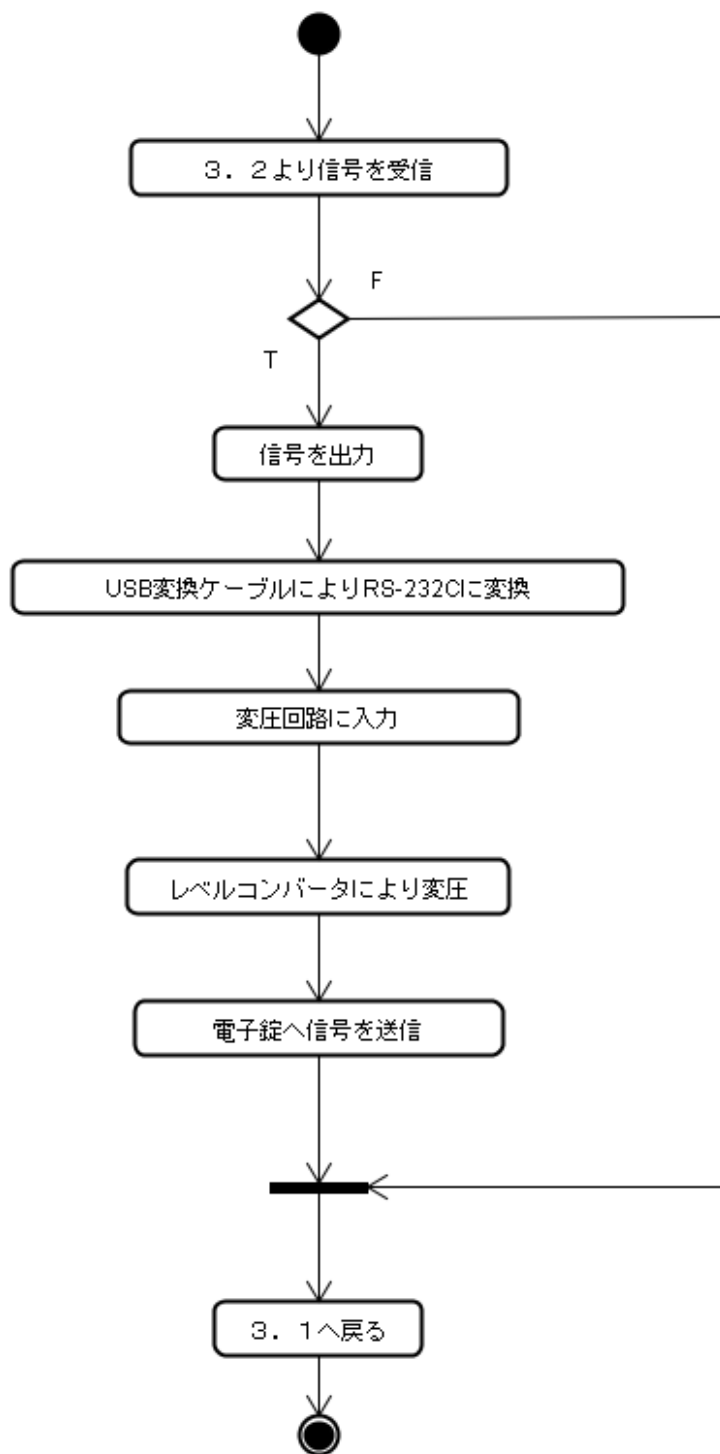


図5.8 電子錠へ命令を送信(フローチャート)



#### 5. 4 IC カードリーダー認識システム

5. 1、5. 2、5. 3 の中で LabVIEW で行ったシステムの全体像を図 5. 9 に示す。このシステムを簡単に表すと以下の動作を行う。

- ①IC カードリーダーの情報を読み取る
- ②読み取った情報と登録者リストを比較する
- ③一致していた場合電子錠に信号を送る
- ④ ①～③の動作を繰り返す

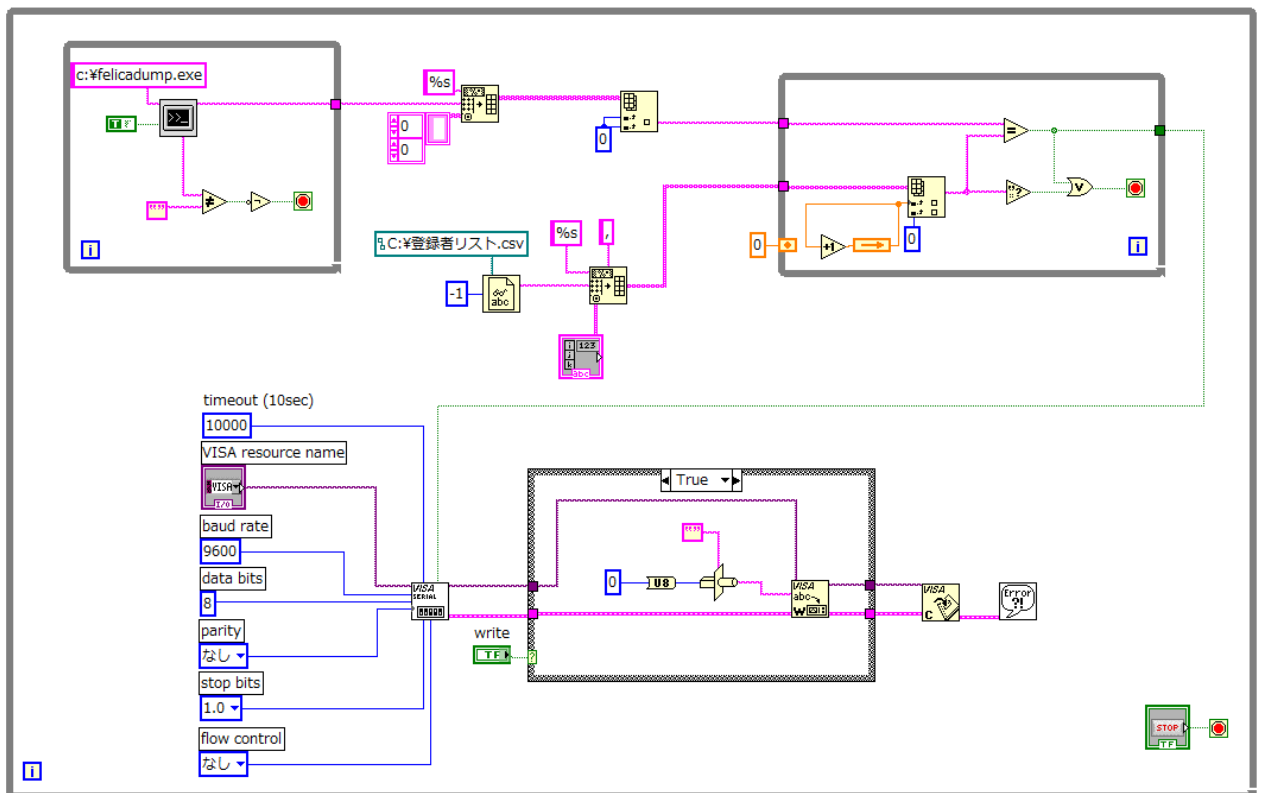


図 5. 9 IC カードリーダー認識システム

## 6. 補助システム

“5. システム構成”は本研究の大本となるシステムについて述べた。ここではそれとは別に“3. システム構成”を補助するために製作したシステムについて述べる。

### 6. 1 ICカードの情報を登録者リストへ保存

このシステムは5. 1で読み込んだデータを登録者リストに保存するシステムである。

5. 1で読み込んだデータが登録者リストにあるかを判断、もし無ければ登録者リストにデータを追記する。ICカードの情報を登録者リストへ保存をするために製作したプログラムを図6. 1に、図6. 1のフローチャートを図6. 2に示す。

このシステムは“3. システム構成”に組み込まず、独立したシステムとして製作した。理由としては、組み込むとこのシステムの動作にON、OFFをつけなくてはならず、人間側がシステムをOFFにすることを忘れるのを防ぐためである。

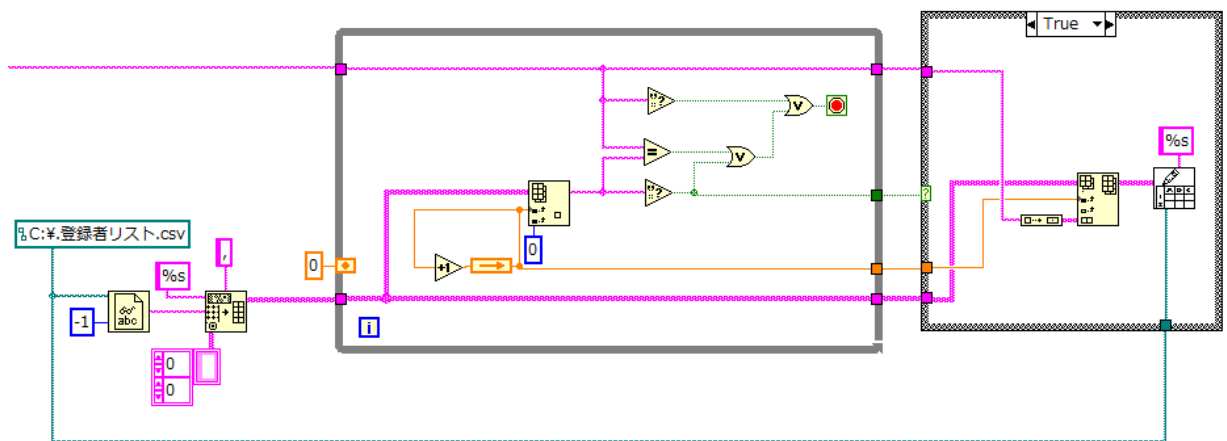


図6. 1 ICカードの情報を登録者リストへ保存(プログラム)

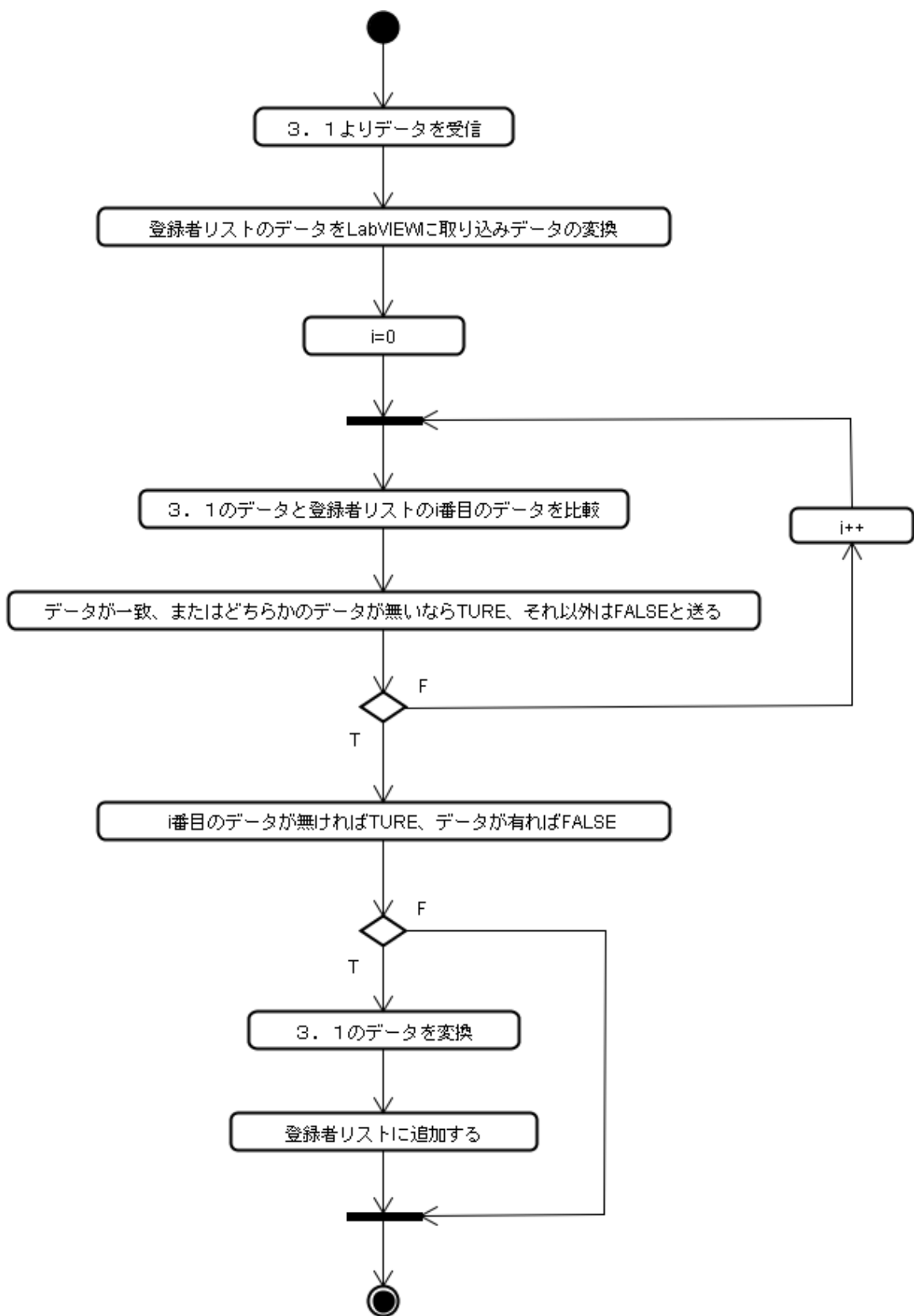


図6. 2 ICカードの情報を登録者リストへ保存(フローチャート)

## 6. 2 入退出時の時刻を保存

このシステムは5. 1でデータを読み込んだ時刻とデータを入退出時刻リストに登録するシステムである。3. 1でデータを読み込んだ時刻を呼び出し、5. 1で読み込んだデータと結合する。結合したデータを入退出時刻リストに保存する。入退出時の時刻を保存するために製作したプログラムを図6. 3に、図6. 3のフローチャートを図6. 4に示す。

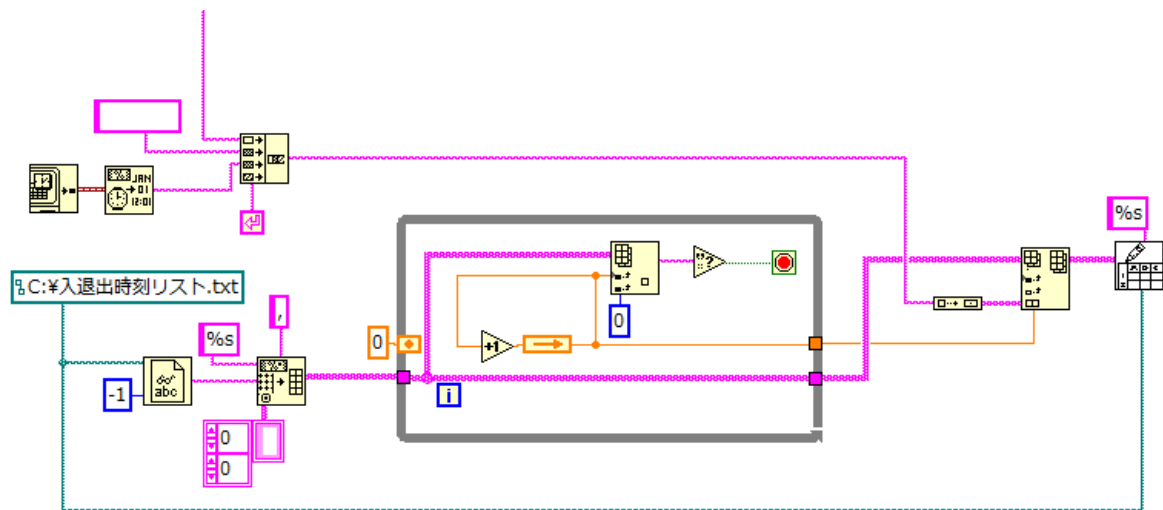


図6. 3 入退出時の時刻を保存(プログラム)

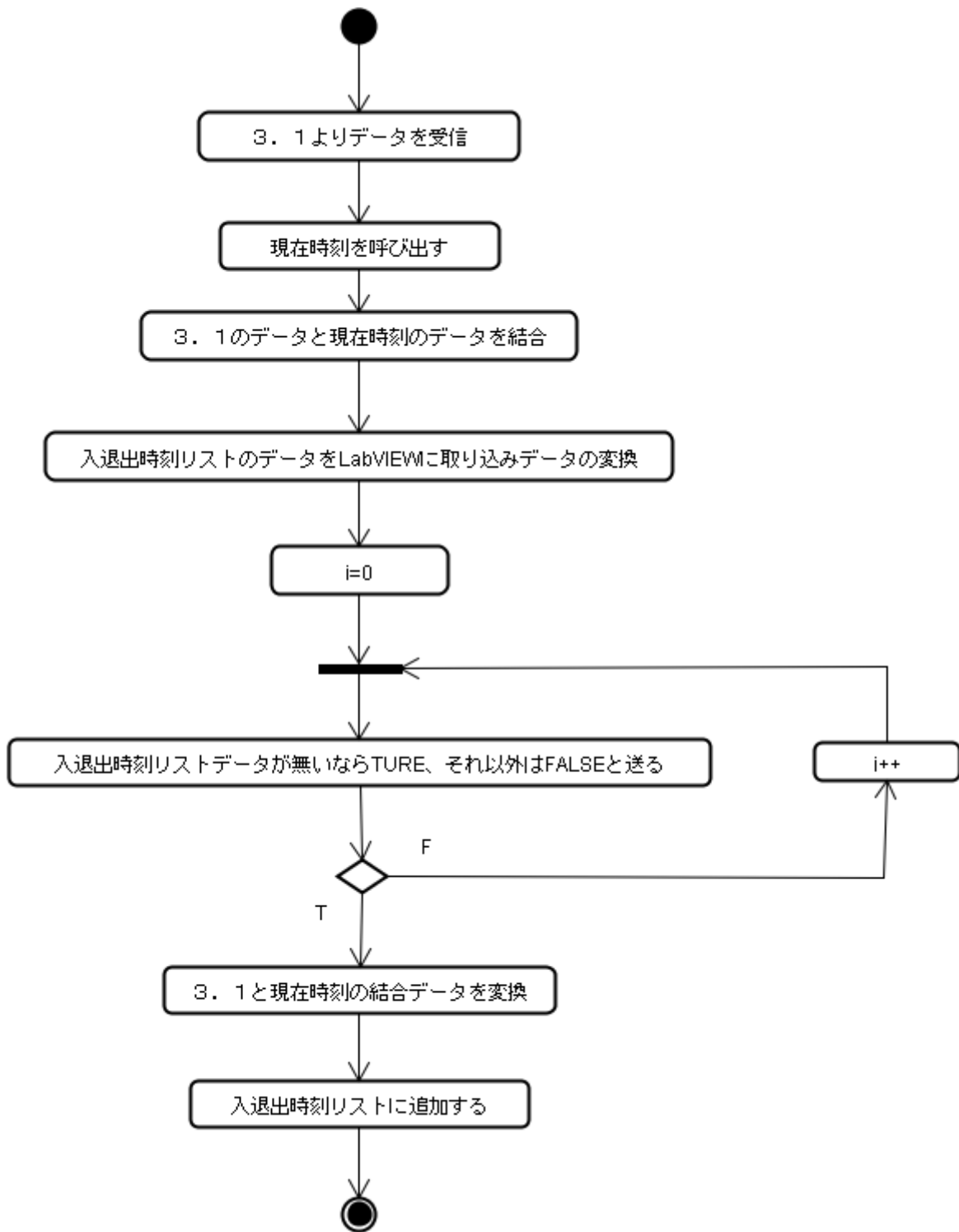


図6. 4 入退出時の時刻を保存(フローチャート)

## 7. 電子錠の製作

### 7. 1 SolidWorks の概要

SolidWorks とは、SolidWorks 社が販売している 3 次元 CAD ソフトである。このソフトは、3 次元設計ツールである CAD ソフトウェア、設計検証のためのシミュレーション・ソフトウェア、製品データ管理ソフトウェア、および製品ドキュメント作成ソフトウェアという、3 次元製品開発環境を包括的に支援し、3 次元データを効率的に活用するための製品である。部品などをドラッグなどによって配置するだけで直観的に設計を行うことができるソフトウェアである。

### 7. 2 設計図の作成

はじめに SolidWorks を使用し、電子錠の設計図を作成した。その電子錠の設計図を図 1-1 に示す。本研究では MISUMI 製の部品を用いて電子錠を設計した。錆が発生しないよう MISUMI 製の部品は表面加工を施したものを使用する。また、独自に製作した部品はジュラルミン材で製作した。

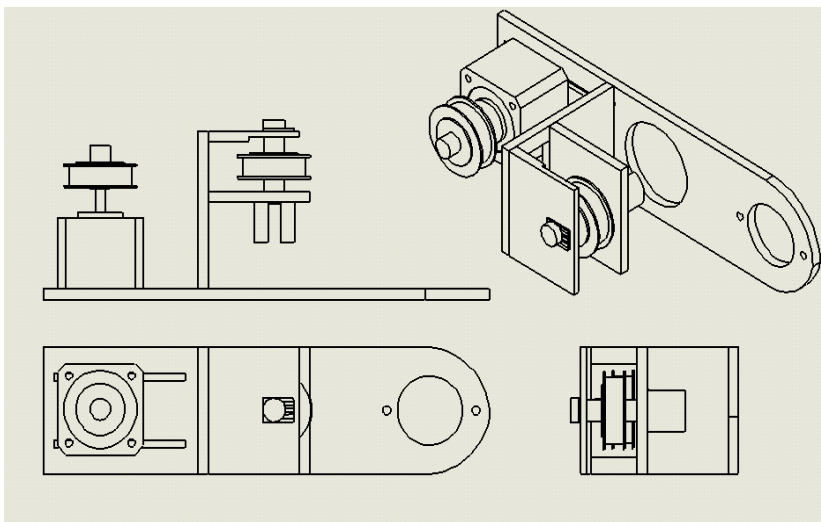


図 7. 1 電子錠の設計図

### 7. 3 電子錠の製作

SolidWorks で作成した設計図から電子錠を製作した。図 7. 2 に製作した電子錠を示す。ステッピングモータにより発生した動力を、プーリーとベルトを用いてアームに伝達する。ベルトを張るためにステッピングモータはレール上を移動が可能とした。可変抵抗をアームの軸に設置しアームの回転角度を測定する。また、ドアノブに電子錠の設置を行った。ドアノブの構造を図 7. 3 に示す。ドアとセットプレートとの間に電子錠を挟みビスで固定する。完成した電子錠をドアに設置させた写真を図 7. 4 に示した。

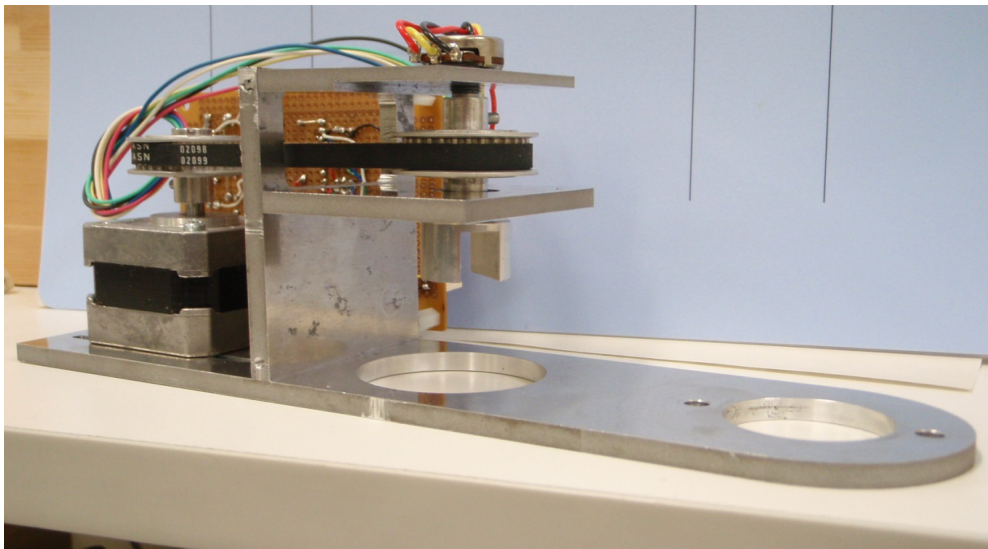


図 7. 2 製作した電子錠

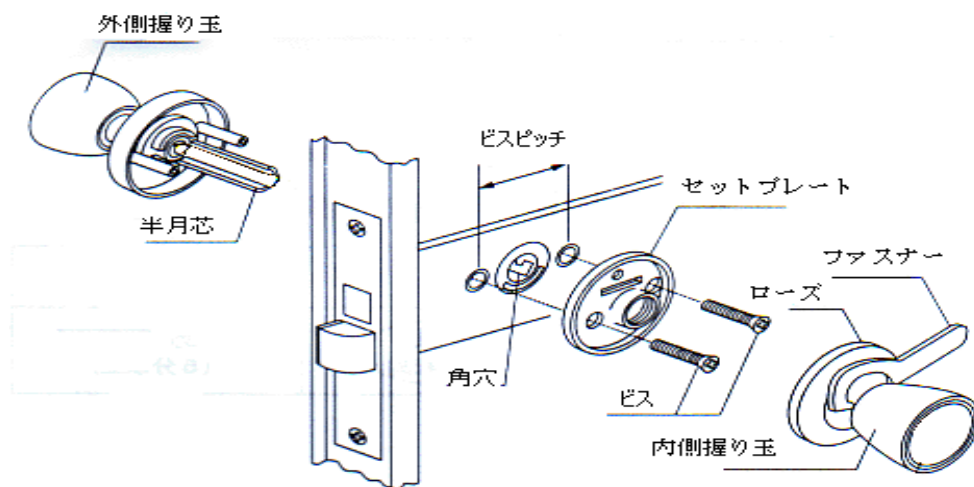


図 7. 3 ドアノブの構造

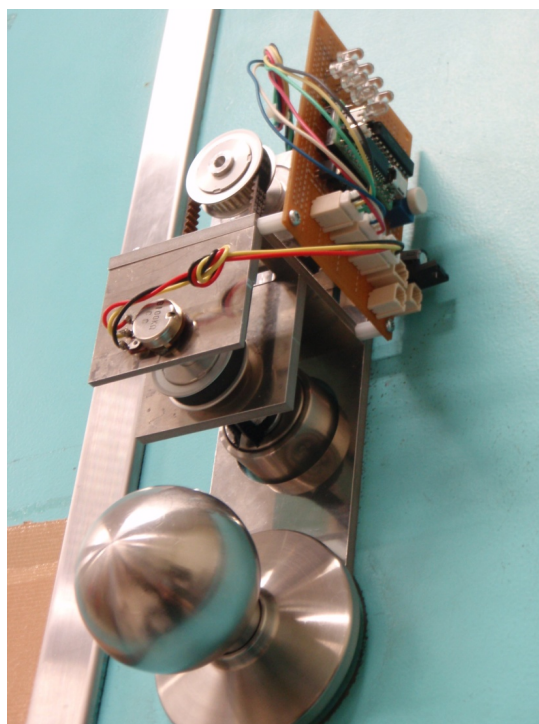


図 7. 4 ドアに設置した電子錠



## 8. ステッピングモータ

### 8. 1 ステッピングモータの静特性

錠を回転させるための動力としてステッピングモータを用いた。ステッピングモータはパルスモータとも呼ばれていて、モータ内部の複数のコイルに適切な順番でパルス信号を与えることで回転する。パルス信号を与えるとモータの軸がステッピングモータ固有のステップ角で回転する。回転角度と回転速度が与えるパルス信号の回数と周期によって決する。

$$\text{回転角度} = \text{ステップ角}(\text{°}) \times \text{パルス数}$$

$$\text{回転速度}[\text{rpm}] = \text{ステップ角}[\text{°}] \times \text{パルス速度}[\text{Hz}] \times 60[\text{s}]/360[\text{°}]$$

通電状態で大きな静止トルクが得られ、通電しない状態(無励磁状態)でも小さな静止トルクが得られる。また、ギアを取り付けることで許容トルクや最大トルクが大きくなる他、基本ステップ角を細かくすることが可能である。このような正確な回転角度制御のできる特性があるため位置制御の用途で多用されている。パルス信号での動作を行うためにデジタル制御との相性が良い。パルス信号とは直流電圧が ON/OFF を繰り返す電気信号である。ON/OFF の 1 サイクルを 1 パルスと数え、1 パルスが命令されると基本ステップ角だけモータ軸が回転する。

今回使用したステッピングモータはユニポーラ型で基本ステップ角は  $1.8^{\circ}$  であり 2 相励磁方式で駆動させた。200 パルス与えるとモータの軸が一回転することとなる。ユニポーラ型は 2 本のコイルに電流を流すことにより回転磁界を発生させ磁化したローターが回転する。図には仕様したステッピングモータとその基本構造を示した。表 8.1 に使用したステッピングモータの仕様を示した。

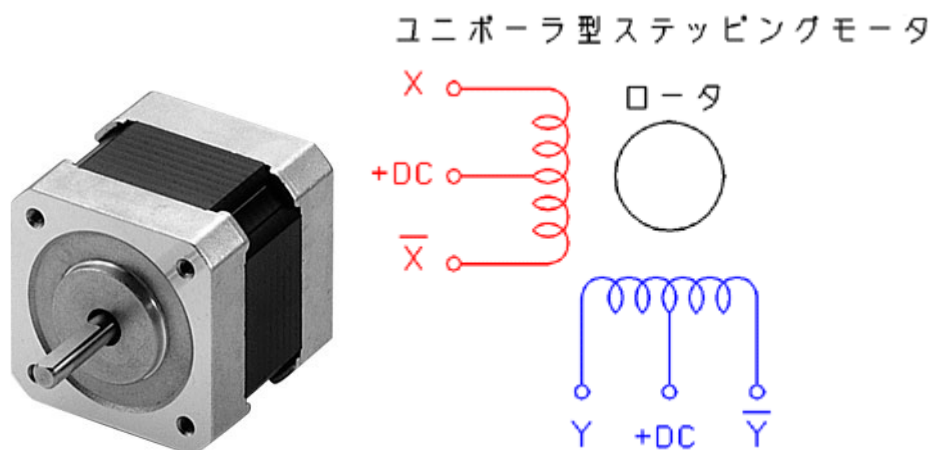


図 8. 1 使用したステッピングモータと構造

表 8. 1 使用したステッピングモータの使用

オリエンタルモータ株式会社	
2相ステッピングモータ(6本リード線)	
形式	PK243-01A
励磁最大トルク	0.16Nm
ローター慣性モーメント	$35 \times 10^{(-7)} \text{ kg} \cdot \text{m}^2$
定格電流	0.95A/相
電圧	4V
巻線抵抗	4.2Ω/相
基本ステップ角	1.8°
質量	0.21kg

## 8. 2 ステッピングモータの動特性

電氣的なパルス信号を回転させたい分送ることによりモータ軸の回転角度がその送られたパルス数だけ基本ステップ回転し止まる。複数のコイルに位相が異なる周期的な電流を流すと、合成させる磁束の向きを回転させることができる。これが回転磁界でありこれを発生させ、磁力によりローターを引き付けて回転する。ローターが回転することによりモータ軸も回転する。回転軸を正転(軸を上から見て時計まわり)、逆転(軸を上から見て反時計回り)させたい時は各コイルに流すパルス信号の順番を変えることにより簡単に制御できる。正転の場合は図の励磁のように  $XY, Y\bar{X}, \bar{X}\bar{Y}, \bar{Y}X$  の順にパルス信号を入力する。逆転の場合は図のように  $XY, \bar{Y}\bar{X}, X\bar{Y}, \bar{Y}X$  という順にパルス信号を入力する。この励磁方式は二相励磁方式のステッピングモータ励磁シーケンスであり、図 2-2 示した流れのとおりステッピングモータドライブ回路に電圧を印可することによりステッピングモータを正転・逆転することができる。このようにパルス信号を与えるたびに決められたステップ角だけ回転する。今回採用した 2 相磁励方式は二相の信号を同時に励磁するので消費電力は一相励磁の 2 倍であるが、得られるトルコも 2 倍となる。

パルス数	1	2	3	4	5	6
回転角度[°]	1.8	3.6	5.4	7.2	9.0	10.8
X	1	1	0	0	1	1
X	0	1	1	0	0	1
Y	0	0	1	1	0	0
Y	1	0	0	1	1	0

表 8. 2 正転の場合の励磁シーケンス

パルス数	1	2	3	4	5	6
回転角度[°]	1.8	3.6	5.4	7.2	9.0	10.8
X	1	0	0	1	1	0
X	0	0	1	1	0	0
Y	0	1	1	0	0	1
Y	1	1	0	0	1	1

表 8. 3 逆転の場合の励磁シーケンス

## 9. ステッピングモータ駆動プログラムの作成

### 9. 1 MAPLAB の概要

MPLAB は米マイクロ社が提供する、組み込みシステム用統合開発環境ソフトウェアのことである。Windows95/98/Me/2000/XP/Vista/7 上で動作するエディタとコンパイラを持ち合わせている。全 PIC シリーズに対応したプログラムの書き込みが可能。開発言語はアセンブラや MAPLIB そして C 言語がある。今回は C 言語を用いてプログラムの作成を行った。

### 9. 2 PIC18F2550

本研究では、ワンチップマイコンのなかで A/D 変換が行え USB ポートを搭載している秋月電子通信社製の PIC18F2550 使用 USB マイコンボード(MICROCHIP18f2550)を使用した。図 2 にマイコンボードの回路を示し特徴を以下に示す。

- ・ 基盤サイズ 40mm×18mm で、28 ピン IC ソケットに装着可能
- ・ プログラムメモリはフラッシュメモリ 32 キロバイト。
- ・ 20MHz の発振器を搭載、内部 PLL は 48MHz で動作
- ・ 電源は、USB からのバスパワー、直流 5V 電源、基盤搭載の 5V レギュレータ使用に対応
- ・ 動作電圧範囲は 2~5.5V
- ・ 通常の命令は 1 サイクルで実行される。分岐命令のみ 2 サイクル必要
- ・ ソフトの開発は、Microchip Technology 社の無償版 C18 コンパイラ (StudentEdition)で可能
- ・ ソフトの書き込みは、AKI-PIC プログラム Ver4 に対応
- ・ 直接的、間接的あるいは相対的なアドレッシングモードを持つ

- ・ 28 ピンで IC ソケットに装着可能で、40mm×18mm
- ・ AD 変換の分解能は 10bit

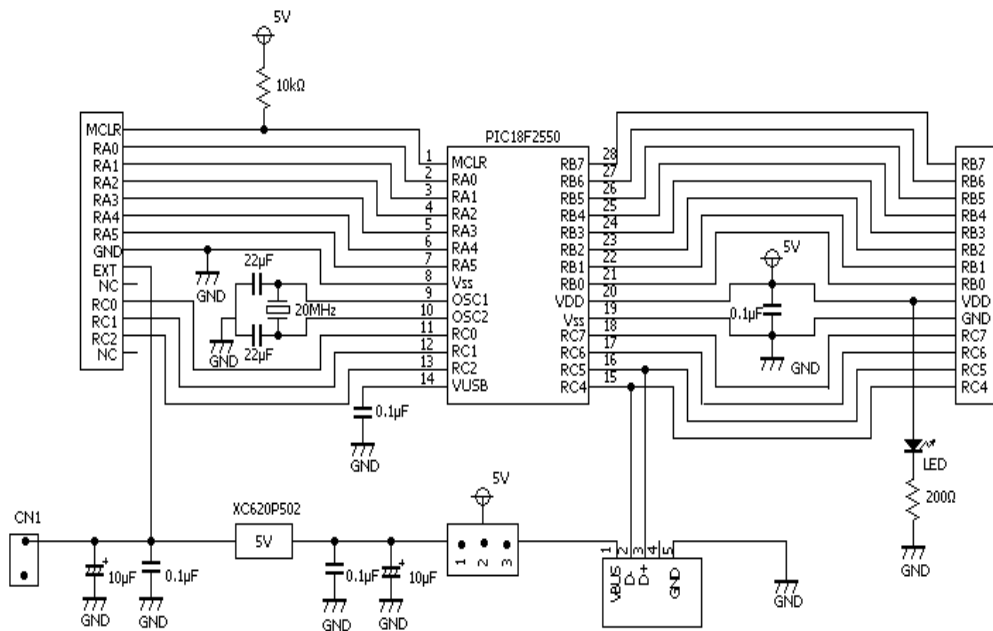


図 9. 1 PIC18F2550

### 9. 3 PICKIT2

PIC18F2550 にプログラムを書き込むため、MICROCHIP 社製の PICKit2 を使用した。図 9. 2 には使用した PICKit2、図 9. 3 にはプログラムを書き込むためにブレッドボード上に製作した回路を示した。また、以下には PICKit2 の機能を示す。

- ・ インサーキット書き込みでユーザー基盤に直接書き込み可能
- ・ USB インターフェイスによる書き込み品装備
- ・ 専用プログラム(MAPLAB とは別)

- ・独立な書き込み器としても使用可能
- ・電源は USB で供給可能
- ・多種のワンチップマイコンをサポート
- ・モジュール変換コネクタで MAPLAB IDE2 コネクタにダイレクトライン
- ・MAL のほとんどのボードに対応



図 9. 2 使用したPICkit2

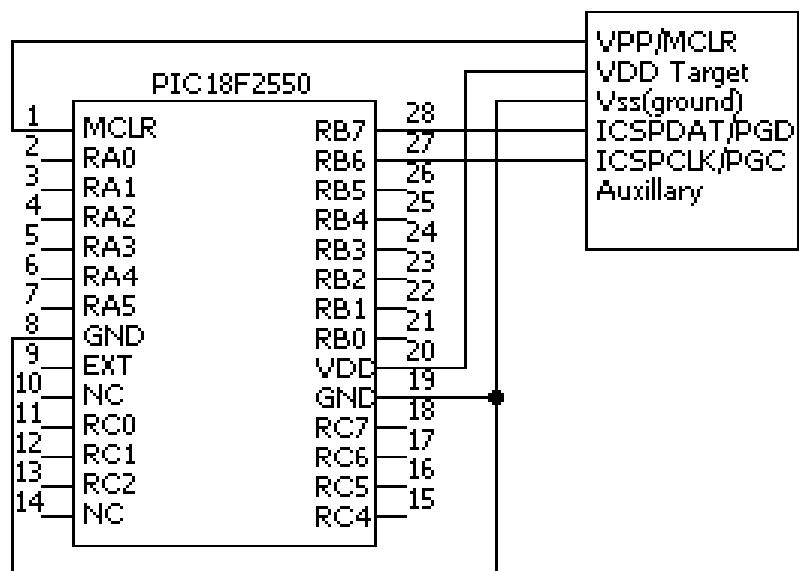


図 9. 3 PICkit2とマイコンボードの接続

## 10. ステッピングモータ正・逆転プログラムの作成

ステッピングモータを回転させるためには電圧出力をステップ信号としなければならない。正転させるためにフローチャートを図10.1に示す。動作確認のためにLED点灯回路を製作した。LEDの点灯から、2相励磁の特徴である2点同時点灯と正しい順序でLEDが点灯するかを確認した。図10.2にLED点灯回路を示す。

このフローチャートのPORTA=12の出力とPORTA=24の出力を入れ替えると逆転を行うフローチャートとなる。ステッピングモータを正転させるプログラムを14. 付録のプログラムリスト1に示す。

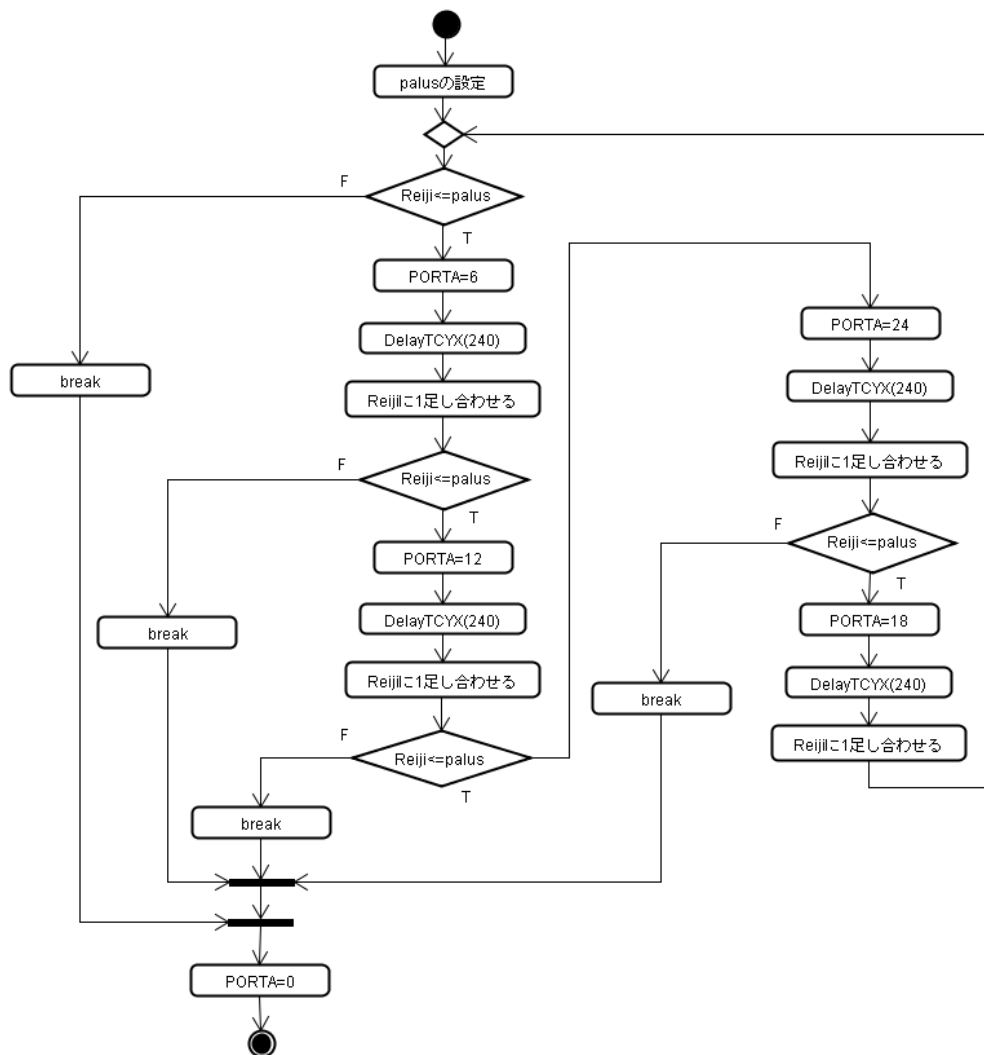


図10.1 正転用パルス信号確認のフローチャート



LED を点灯させるフローチャートの SENTENT では、与えられたパルス信号数が正の数であるところに入る。これはステッピングモータ軸を正転させる用のフローチャートとなる。Reiji ではステッピングのコイルが何回励磁を行ったかを示す変数である。これを与えられた pulse の数だけ励磁を行いステッピングを回転させる。PORTA=9などで各ピンに出力を行わせている。図の正転の場合の励磁シーケンスピンの 0 時間を見ると 1001 というように出力を行っていて、それを PIC のピンの出力で行わせた。2 進数の 1001 を 10 進数に変換すると 9 となる。PORTA=9 は RA0 と RA3 を出力するということになる。Delay10KTCYX(240)という関数では、時間待機をさせるものでこれを用いてパルス幅を設定することができる。10KTCY(120)では 200msec 待機できる。10KT=0.05  $\mu$  sec  $\times$  4  $\times$  1000=2msec となる。Break は関数であり、while 文の繰り返し処理を終わらせるために使用した。

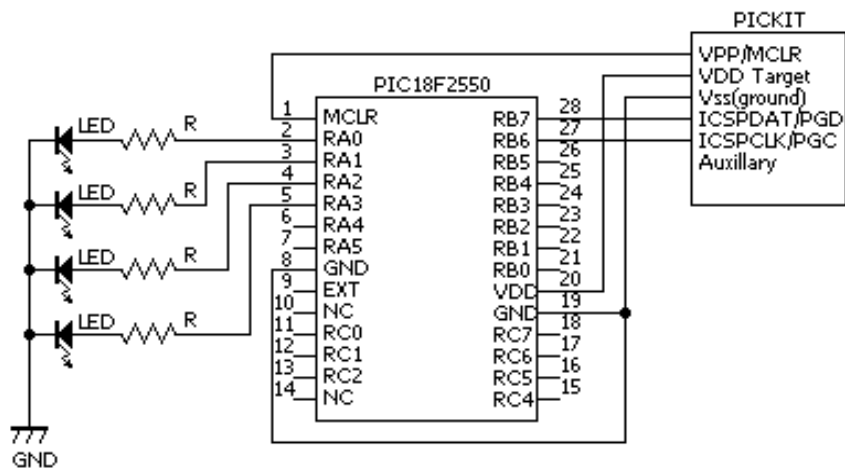


図 10. 2 LED 点灯回路

フローチャートのプログラムを実行した結果により点灯した LED は図のようになった。黒く塗りつぶしてある LED が点灯したということとする。この結果より正転の場合の励磁シーケンス通りに LED を点灯させることができた。

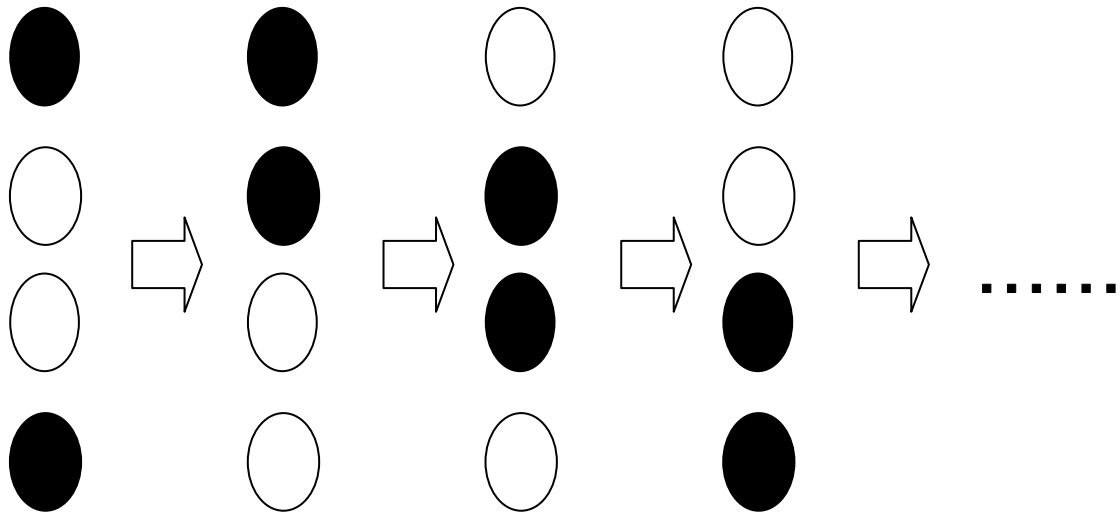


図 1 0 . 3 正転用の led の点灯順番

次に逆転用のプログラムで led 点灯を確認したところ、逆転の場合の励磁シーケンス通りに led が点灯できた。

この結果から正、逆回転を行わせる場合の PIC の各ピンが行う電圧出力を横軸を経過時間、縦軸を出力電圧として図 1 0 . 4 に正転の場合、図 1 0 . 5 に逆転の場合を示す。

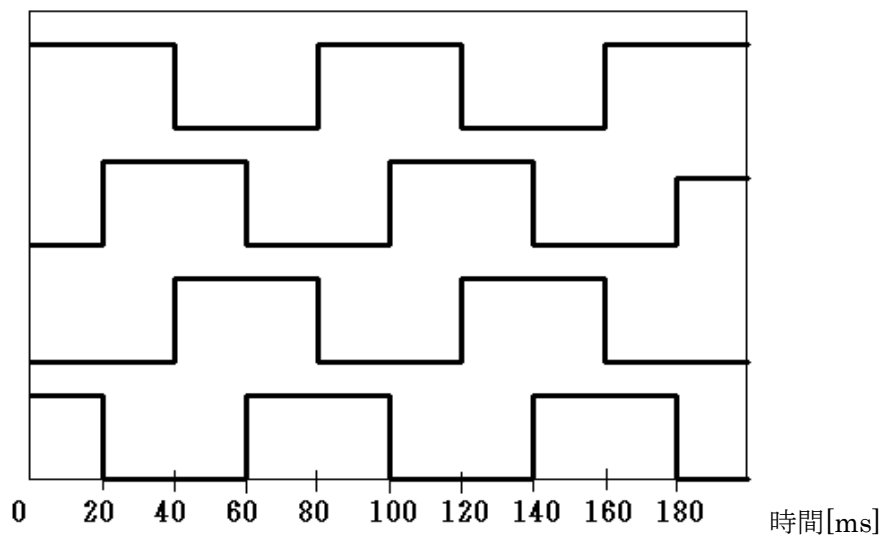


図 1 0 . 4 正転の場合の各ピンの電圧出力

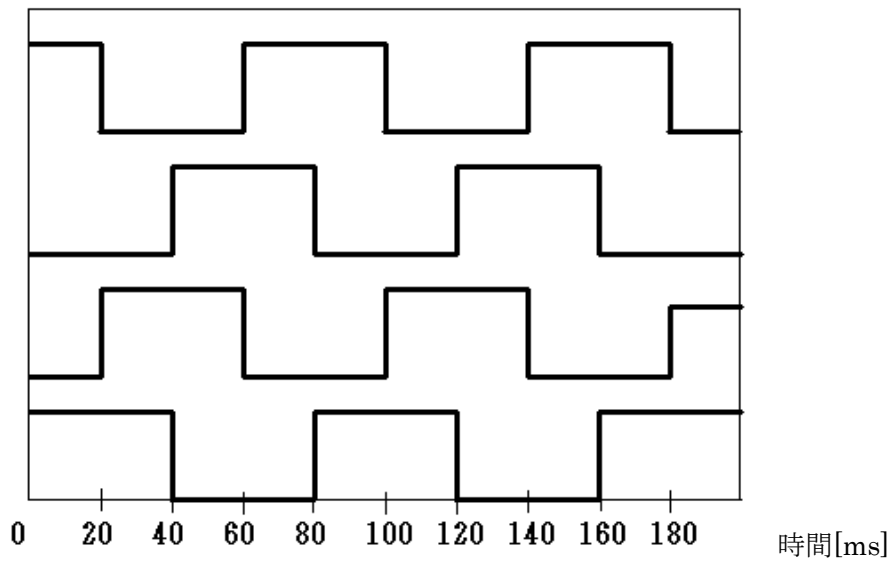


図10.5 逆転の場合の各ピンの電圧出力

以上の結果から電氣的なステップ信号を出力できた。また、位相の異なったパルス信号を作り出すことができ、2相磁励方式の信号を作成することができた。このプログラムを利用して次のにステッピングモータを回転させた。モーター駆動には1Aの電流を必要とするためパワーMOSFETを使用した。他のパワーデバイスと比較してもスイッチング速度が速い。なおかつ、低電圧領域での変換効率が低い。

## 1 1. 電子錠のプログラム

### 1 1. 1 可変抵抗

施錠・解錠を行うためには錠の回転角度を制御することが必要である。そこで可変抵抗を用いて錠つまみの位置を判断する。使用した可変抵抗の規格を以下に示す。

- ・ 抗値 : 10K  $\Omega$
- ・ カーブ特性 : B カーブ
- ・ 定格電力 : 0.125W
- ・ 最高使用電圧 : 200V
- ・ 回転角度 : 300°  $\pm 5^\circ$
- ・ 軸 :  $\phi 6\text{mm}$  (ネジ部 M7)、長さ 13.5mm
- ・ 取付部品が付属 (ネジ、ワッシャ)

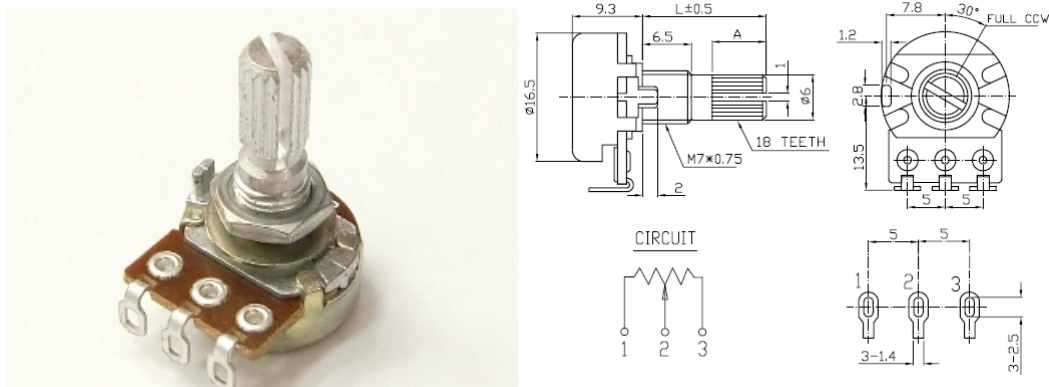


図 1 1. 1 使用した可変抵抗

可変抵抗を図 1 1. 2 ように制御回路に結線した。錠回転軸の回転に比例して PIC18F2550 の RA0 端子に入力される電圧は変化する。この電圧を A/D 変換を行い、その結果を RB0~7 の LED を点灯させて 2 進数で表示した。A/D 変換の分解能は  $V_{ref+}$  と  $V_{ref-}$  の電圧の間を 256 等分した値となる。 $V_{ref+}$  と  $V_{ref-}$  の間の電圧は  $V_{dd}$  と GND 間の

電圧になるよう設定した。また、可変抵抗の両端にも Vdd の電圧を印加した。そのため、 $1.172^\circ$  間隔での回転軸の回転角度を把握することができる。

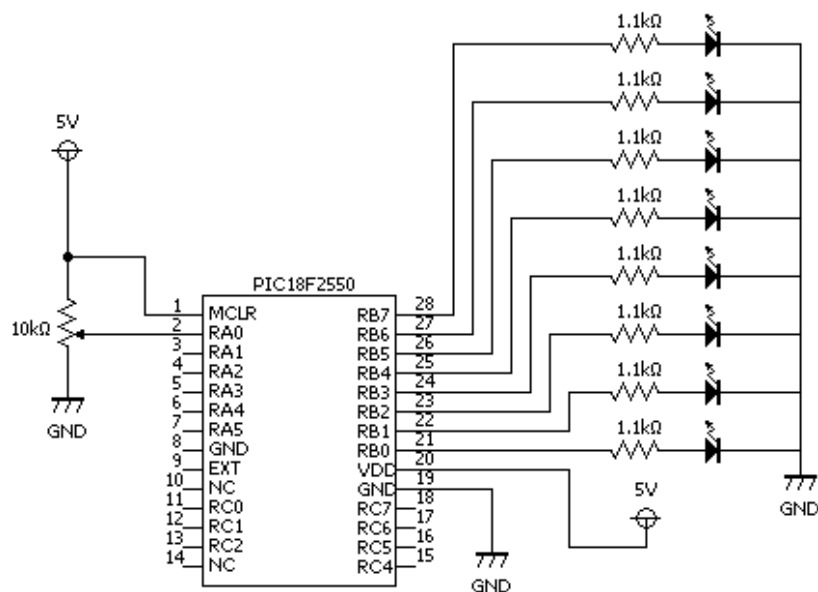


図 1 1 . 2 位置制御回路

### 1 1 . 2 位置制御システム

電子錠をドアに設置した。基盤上の A/D 変換の結果を表示する LED より、施錠と解錠のしきり値、施錠命令がある場合に停止する値と解錠命令がある場合に停止する値を決め測定した。その結果を表 1 1 . 1 にまとめた。また、図 1 1 . 3 には実験錠をどの角度で停止させたいかをまとめた。

表11.1 位置制御結果

	施錠と解錠境界	施錠命令停止位置	解錠命令停止位置
A/D 変換結果	168	45	172
角度°	90	2	92

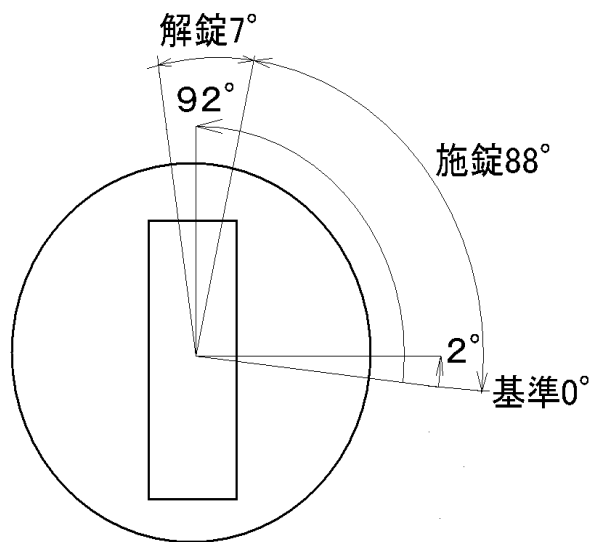


図11.3 実験錠の可動域

### 11.3 駆動プログラム

以下に電子錠が駆動命令を受信した場合のフローチャートを図11.4に示した。

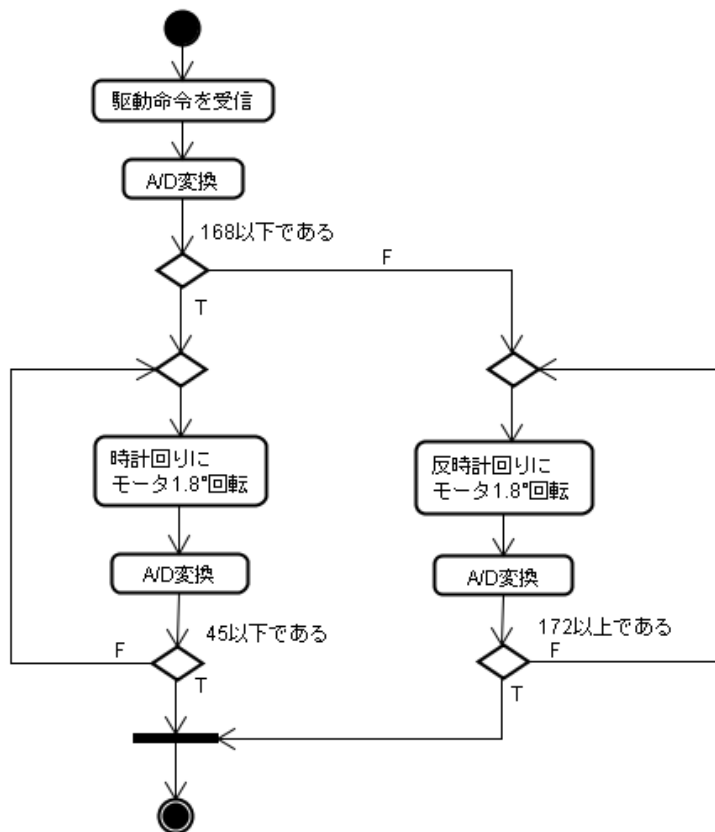


図11.4 電子錠駆動の流れ

駆動命令を受信した場合、A/D変換の結果が168以下であれば施錠、168以上であれば解錠と判断する。施錠であれば、反時計回りに1.8°回転してからA/D変換を行う。この動作をA/D変換の値が45以下になるまで繰り返す、解錠にする。解錠であれば、時計回りに1.8°回転してからA/D変換を行う。この動作をA/D変換の値が172以上になるまで繰り返す、施錠にする。よって、施錠であれば解錠(2°以下)、解錠であれば施錠(92°以上)といった動作を行う。ステッピングモータを使用しているためステップ角の間隔での駆動しかできない。そのため、停止角度よりも進行方向に最大1.8°の誤差が生じる。

#### 1 1. 4 電子錠の特徴

以下に製作した電子錠の機能及び特徴を示す。電子錠のプログラムを1 4. 付録のプログラムリスト 2 に示した。また、電子錠の回路図を図 1 1. 5 に示した。

- ・施錠状態であれば解錠、解錠状態であれば施錠を行う
- ・基盤錠のリミットスイッチを押すと駆動
- ・登録された IC カードの信号を受信すると駆動
- ・IC カードリーダに IC カードをかざすと、電子錠が処理を終えるまで1 秒
- ・1. 1 7 2° 間隔での位置計測が可能
- ・LED の点灯で A/D 変換の結果表示
- ・受信機の LED で施錠または解錠状態の表示
- ・駆動トルクは 80mNm
- ・駆動速度 1800rpm
- ・既存の鍵も使用可能
- ・ギアの変更によりギア比を変えることで、錠回転軸のステップ角を変更可能
- ・プログラムで回転速度・トルクを変更可能



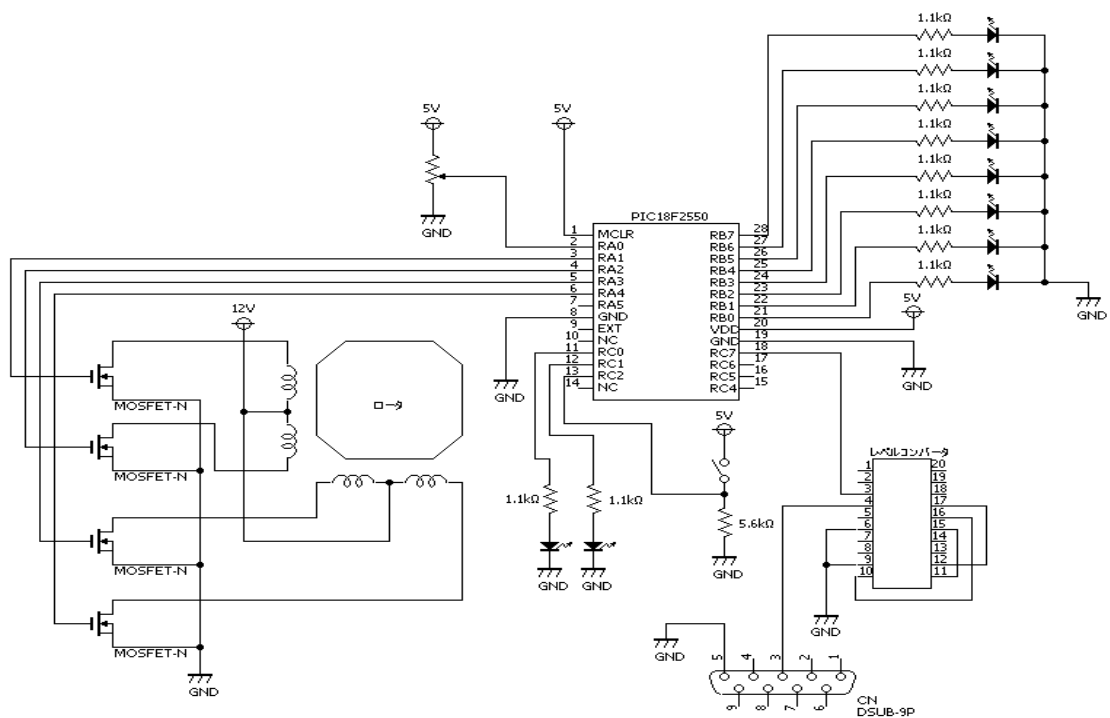


図 1 1 . 5 電子錠の回路図

### 1 3 . まとめ

ICカードリーダー認識システムと電子錠を接続した。また、ICカードをICカードリーダーにかざすことによって電子錠が動作することを確認した。これによりICカードを使用したドアの解錠、施錠を行うことができた。

## 1 4 . 付録

### 1 4 . 1 プログラムリスト 1

```
#include<stdio.h>
```

```
#include<p18f2550.h>
```

```
#include<delay.h>
```

```
#include<adc.h>
```

```
#include<timers.h>
```

```
#program config FOSC=HSSL-HS //PIC18F2550 の動作設定
```

```
#program config WDT=OFF
```

```
#program config PLLDIV=5
```

```
#program config CPUDIV=OSC_PLL2
```

```
#program config USBDIV=OSC1_PLL2
```

```
#program config USBDIV=2
```

```
#program config PWRT=ON
```

```
#program config BOR=ON
```

```
#program config LVP=OFF
```

```
#program config VREGEN=ON
```

```
#program config MCLR=ON
```

```
#program config PBADEN=OFF
```

```

void main(void)
{

    int i,paruse;

    int=0;

    parusu=0;

    TRISA=0;           //PORTA の入出力設定(0:出力,1:入力)
    PORTA=0;          //PORTA 全ピン 0
    Delay10KTCY(240); //一秒待ち

    while(parusu <= 100) //コイルに 100 回励磁を行う
    {
        //正転をする場合の PORTSA の出力の流れ

        PORTA= 18;      //PORTA の出力
        Delay10KTCY(240); //一秒間待機
        parusu += i;    //励磁をした回数に 1 足し合わせる
        if(parusu == 100) //励磁を何回行ったか判断する
        {
            break; //100 回励磁をおこなっていたら while 文から抜け出す
        }

        PORTA= 6;
        Delay10KTCY(240);

        parusu += i;

        if(parusu == 100)
        {
            break;
        }
    }
}

```

```

    }

    PORTA= 12;

    Delay10KTCY(240);

    parusu += i;

    if(parusu == 100)
    {
        break;
    }

    PORTA= 12;

    Delay10KTCY(240);

    parusu += i;

    if(parusu == 100)
    {
        break;
    }

}

PORTA= 0;          //PORTA をリセット
}

```

#### 14.2 プログラムリスト2

```
#include<p18cxxx.h>

#include<delays.h>

#include<adc.h>

#include<usart.h>

#pragma config FOSC=HSPLL_HS

#pragma config WDT=OFF

#pragma config PLLDIV=5

#pragma config CPUDIV=OSC1_PLL2

#pragma config USBDIV=2

#pragma config PWRT=ON

#pragma config BOR=ON

#pragma config BORV=3

#pragma config LVP=OFF

#pragma config VREGEN=ON

#pragma config MCLRE=ON

#pragma config PBADEN=OFF

float ADhenkan(void);

void sejou(int,int);           //施錠を行う動作

void kaijou(int,int);         //解錠を行う動作

void LED(float);              //室外の点等

void move(float);             //鍵の状態判断してモータを駆動させる
```

```

void main(void)
{
    float kakudo;

    char cmd;

    OpenUSART(USART_TX_INT_OFF & USART_RX_INT_OFF &
              USART_ASYNC_MODE & USART_EIGHT_BIT &
              USART_CONT_RX & USART_BRGH_LOW,77);

    TRISA=1;

    TRISB=0;

    TRISC=0b11111100;

    PORTA=0;

    while(1)
    {
        kakudo=ADhenkan();

        LED(kakudo);

        PORTB=kakudo;

        if(DataRdyUSART || PORTCbits.RC2== 1)
        {
            move(kakudo);

            cmd=ReadUSART();
        }
    }
}

```

```

float ADhenkan(void)
{
    float Result,kakudo;

    OpenADC(ADC_FOSC_32&ADC_RIGHT_JUST
            &ADC_12_TAD,ADC_CH0&ADC_INT_OFF
            &ADC_REF_VDD_VSS,ADC_7ANA);

    SetChanADC(ADC_CH0);

    ConvertADC();

    while(BusyADC());
    {
        Result=ReadADC();

        kakudo=Result/1024*256;
    }

    return(kakudo);
}

```

```

void kaijou(int j,int b)
{
    float kakudo;

    kakudo=ADhenkan();

    PORTB=kakudo;

    while(b>=kakudo)
    {
        PORTA=18;
    }
}

```

```
Delay10KTCYx(j);  
kakudo=ADhenkan0;  
PORTB=kakudo;  
if(b<=kakudo)  
{  
    break;  
}  
PORTA=6;  
Delay10KTCYx(j);  
kakudo=ADhenkan0;  
PORTB=kakudo;  
if(b<=kakudo)  
{  
    break;  
}  
PORTA=12;  
Delay10KTCYx(j);  
kakudo=ADhenkan0;  
PORTB=kakudo;  
if(b<=kakudo)  
{  
    break;  
}  
PORTA=24;
```



```

        Delay10KTCYx(j);

        kakudo=ADhenkan();

        PORTB=kakudo;

    }

}

```

```

void sejou(int j,int a)
{

    float kakudo;

    kakudo=ADhenkan();

    while(a<=kakudo)
    {

        PORTA=24;

        Delay10KTCYx(j);

        kakudo=ADhenkan();

        PORTB=kakudo;

        if(a>=kakudo)
        {

            break;

        }

        PORTA=12;

        Delay10KTCYx(j);

        kakudo=ADhenkan();

        PORTB=kakudo;
    }
}

```

```

    if(a>=kakudo)
    {
        break;
    }

    PORTA=6;

    Delay10KTCYx(j);

    kakudo=ADhenkan();

    PORTB=kakudo;

    if(a>=kakudo)
    {
        break;
    }

    PORTA=18;

    Delay10KTCYx(j);

    kakudo=ADhenkan();

    PORTB=kakudo;

    }
}

```

```

void move(float kakudo)
{

```

```

    int j;

    int a=65;

    int b=65;

```

```

j=5;
if(a>=kakudo)
{
    kaijou(j,b);
}
else
{
    sejou(j,a);
}
}

```

```

void LED(float kakudo)
{
    if(65>=kakudo)
    {
        PORTCbits.RC0=1;
        PORTCbits.RC1=0;
    }
    if(65<=kakudo)
    {
        PORTCbits.RC0=0;
        PORTCbits.RC1=1;
    }
}

```