

## 電子制御工学実験報告書

実験題目 : オートマトンのプログラミング  
報告者 : 3 年 32 番 平田 蓮  
提出日 : 2019 年 10 月 29 日  
実験日 : 2019 年 10 月 7 日, 10 月 21 日, 10 月 28 日  
実験班 : 第 4 班  
共同実験者 : 4 番 石橋那起  
8 番 小林歩夢  
12 番 小室弦太  
15 番 佐藤貴幸  
20 番 関晋一郎  
24 番 高橋祐己哉  
28 番 外川諒太郎  
36 番 本多充稔

### ※指導教員記入欄

評価項目	配点	一次チェック ・ ・	二次チェック ・ ・
記載量	20		
図・表・グラフ	20		
見出し, ページ番号, その他体裁	10		
その他の減点	—		
合計	50		

コメント :

## 1 目的

本実験では、擬似自動販売機回路のプログラムを作成し、プログラミングを通してオートマトンの考え方を理解する。

## 2 有限オートマトン (Finite Automaton: FA)

オートマトンとは自動機械という意味であるが、工学で用いられる場合は、離散的な入力及び出力を持つ機械のモデルのことであり、状態とその遷移という考え方で捉える。ある装置の動作を実現することを考えた場合に、入出力をまず考えるが、それだけでは動作を実現することはできない。出力を決定する要素として内部状態という考えが必要である。

装置の取り得る内部状態の数が有限個の場合、その装置を有限オートマトンといい、その動作は次の 5 個の集合と関数で記述できる。

■FA に必要な集合と関数 上で述べた集合と関数を示す。

- $X$ : 入力集合
- $Q$ : 状態集合
- $Z$ : 出力集合
- $\sigma$ : 状態遷移関数  $\sigma(X, Q) \rightarrow Q$
- $\omega$ : 出力関数  $\omega(X, Q) \rightarrow Z$  または  $\omega(Q) \rightarrow Z$

### 2.1 状態遷移図

FA の動作を図で表すには状態遷移図を用いると良い。

例として 10 円硬貨だけが使える 30 円切手自動販売機を考える。Cancel ボタンを押すと払い戻しとする。

- $X$ : {10[円], Cancel}
- $Q$ : {0[円], 10[円], 20[円]} (初期状態: 0[円])
- $Z$ : {1[枚], 10[円], 20[円]}

この FA の状態遷移図を図 1 に示す。

## 3 練習問題

実験テキストの練習問題の状態遷移図を示す。

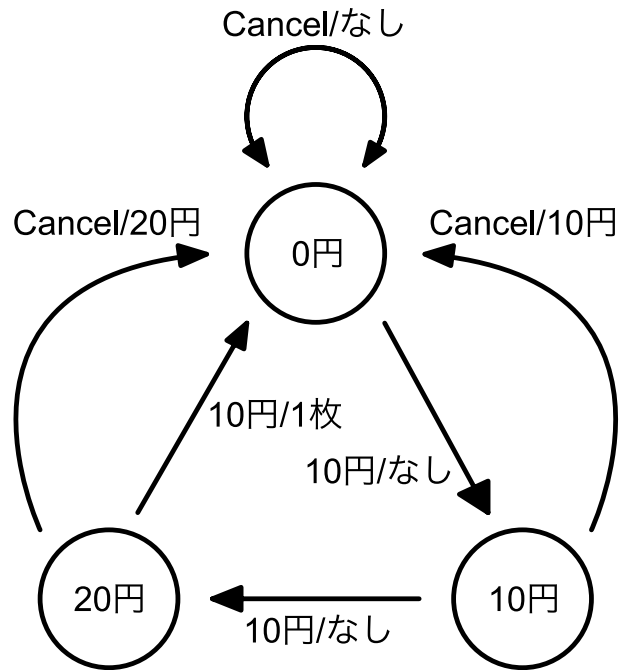


図1 30円切手自動販売機

(1) 10円硬貨だけが使用できる40円切手自動販売機. Cancelを押すと払い戻し.

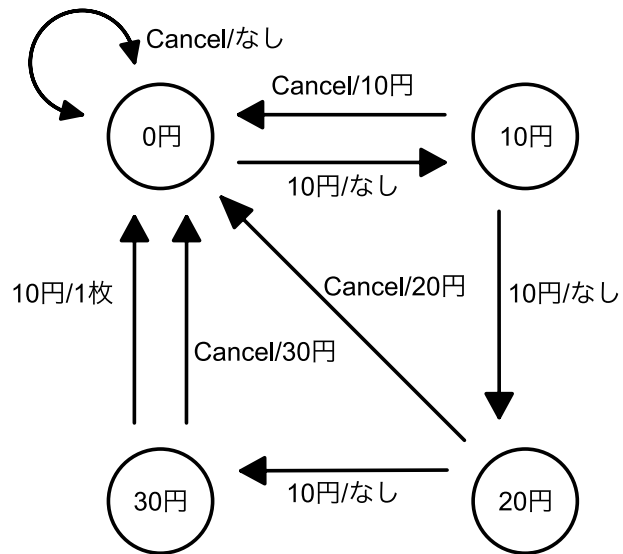


図2 40円切手自動販売機

(2) 10円硬貨と50円硬貨だけ使える30円切手自動販売機. Cancelを押すと払い戻し. (お釣りはCancelを押さないと出てこない.)

(3) 10円, 50円, 100円硬貨が使える20円切手自動販売機. Cancelを押すと払い戻し. (お釣りはCancelを押さないと出てこない.)

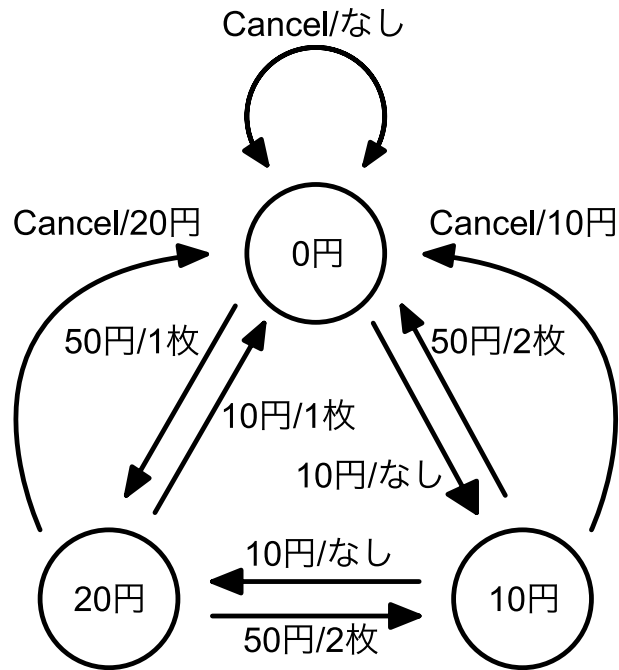


図3 30円切手自動販売機

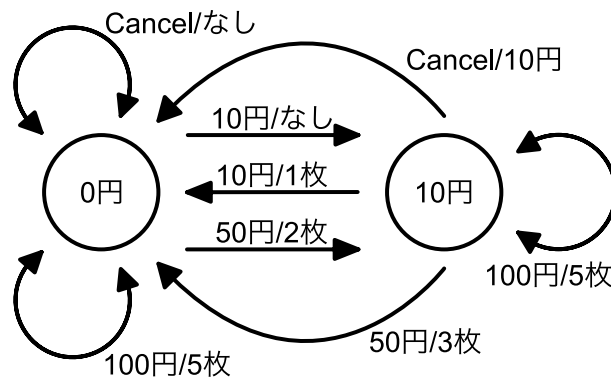


図4 20円切手自動販売機

## 4 仮想自動販売機作成実習

今回は, 10円と50円が使える20円切手にした. 10円入ってるときに50円を入れると60円になり切手が3枚出力されるので, 3枚目は100円のランプを使うこととした. また, Cancelを押すと払い戻しをする.

図5に状態遷移図を示す.

### 4.1 ソースコード

今回変更を加えた部分のソースコードを以下に示す.

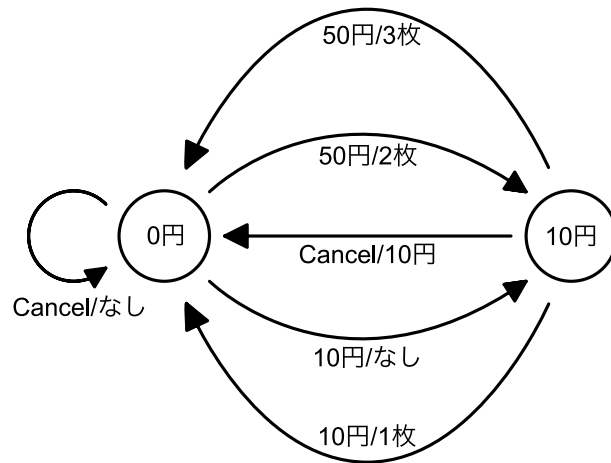


図5 20円切手自動販売機状態遷移図

---

ソースコード 1 piclib.c

---

```

1 void StampOut(int num) {
2     if (num > 2) led_on(LED100);
3     if (num > 1) led_on(STAMP2);
4     if (num > 0) led_on(STAMP1);
5 }
  
```

---



---

ソースコード 2 piclib.h extern.h

---

```

1 void StampOut(int num); /* スタンプ表示 */
  
```

---



---

ソースコード 3 stamp.c メインループ

---

```

1 do {
2     it = sw_read();
3     if (it == 0) continue;
4     outfg = outcoin = 0;
5     s = Transition(it, &st, &a, &b);
6     DispStatus(status_bit[st]);
7     if (a | b) outfg = 1;
8     StampOut(s);
9     if (a) DispCoinA(a);
10    if (b) DispCoinB(b);
11    while(it = sw_read()){ /* スイッチ入力監視 */
12        if (it == Exit) break;
13        timer(30);
14    }
15 } while (st != stExit);
  
```

---



---

ソースコード 4 stamp.c 遷移関数

---

```

1 int Transition (char it, Status* st, int* a, int* b) {
2     *a = *b = 0; /* コイン A, の出力枚数をにしておく B0 */
  
```

---

```

3     if (it == Exit) *st = stExit;
4     if (it == Cancel) {
5         *a = *st;
6         *st = stEmpty;
7     }
8     if (it == CoinA) {
9         ++*st;
10        *st = *st % 2;
11        if (!*st) {
12            return 1;
13        } else {
14            return 0;
15        }
16    }
17    if (it == CoinB) {
18        *st += 5;
19        *st = *st % 2;
20        if (*st) {
21            return 2;
22        } else {
23            return 3;
24        }
25    }
26    return 0;
27 }

```

---

まず、ソースコード 1 にあるように、切手の出力をする関数を複数枚出力ができるように変更した。StampOut 関数は引数で切手の枚数を受け取り、その値の個数だけ LED を光らせる。また、それに伴い piclib.h と extern.h にある Stampout 関数のプロトタイプ宣言を変更した。

ソースコード 4 には、変更後の遷移関数を示した。遷移関数 Transition は受け付ける入力を 2 種類に増やし、それぞれの入力に対し適切な切手枚数を返すように変更した。

最後に、これらの変更に伴い、ソースコード 3 のように stamp.c のメインループ内を変更した。

## 5 調査課題「ワンチップマイコンについて調査せよ」

コンピュータを形成するのに必要な要素は入出力装置、記憶装置、処理装置などである。これらを一つの IC にまとめたものがワンチップマイコンである。

ワンチップマイコンの特徴として、以下のものが挙げられる。

- 小さい
- 安い
- 多種類

- プログラムの書き換えが可能

現在主流のワンチップマイコンは、Microchip 社製の PIC シリーズと、現在は Microchip 社に買収された Atmel 社製の AVR シリーズがある。今回の実験では PIC シリーズを使用した。

AVR には PIC と比べて以下のような特徴がある。

- 高速
- 操作が簡単
- 知名度が低く、情報が少ない

それぞれに特徴があり、用途によって使い分ける必要がある。

## 6 考察

今回は練習問題で入力集合の大きさが 4 の自販機が最大だったが、実際の自販機を考えると、入力集合の大きさはより大きくなる。

そのため、実装するには状態遷移図、ソフトウェア共に工夫が必要になってくると考えられる。

今回はソフトウェア面での工夫を考えてみた。今回の実験で使用したコードでは、一つの遷移関数 `Transition()` の中で状態の遷移を行なったが、構造体やクラスを使って硬貨や自販機の処理との関係性をシミュレーションをすることで簡潔なコードで多種の入力がある自販機を実装することができると考えた。この方法をとることで出力種類の増加にも対応させることができる。

## 7 感想

今回の実験では、前期のデジタル論理回路の授業内容が活かせた。また、今まで培ってきたプログラミング能力を駆使して比較的早く課題を終わらすことができた。発展課題には挑戦しなかったのが、今後機会があったら調べてみたい。

## 参考文献

[1] Electro-ワンチップマイコン <http://laboratory.sub.jp/ele/13.html/>

[2] 第 1 章 ワンチップマイクロチップコンピュータとは

<https://www.nmri.go.jp/oldpages/eng/khirata/mcon/ch01.html/>

[3] さらに可能性を秘めた PIC, AVR マイコン

<https://www.rs-online.com/designspark/unlimited-possibilities-of-pic-and-avr-microcontrollers-jp/>