

マイクロキャットの製作

岡山県立水島工業高等学校

草野泰秀

マイクロキャットとは、前と左右の3カ所のセンサーにより壁の有無を判定して迷路を脱出する自走式ロボットのことである。これは、アルミ板の上に2個のパルスモーターを制御するためのモータードライブ回路、8255インターフェース回路、バッテリーおよびポケットコンピュータ(ポケコン)を搭載したもので、すべて手作りのものである。今回の発表は、昨年を選択実習の情報技術班において、生徒と共に研究実践したものである。

1. はじめに

選択実習における情報技術は、1学期はポケコンを使用したBASICおよびアセンブラによるLEDの点滅、モーターの回転制御等の基礎的な実習をした。2学期の始めには、ワンボードマイコンのROMにプログラムを書き込んでモーター等の制御実習をした。

マイクロキャットの製作にとりかかったのは2学期の中頃からである。これは機械工作班、電子回路班、ソフトウェア班の3班(各班2~3人)編成で製作した。

2. マイクロキャットの製作

(1) 機械工作

図1にマイクロキャットの側面図を図2に平面図を示す。

機械工作班では、図面どおりにアルミ板にけがきをし、次にシャーで切って型を作った。

また、アルミ棒を旋盤で削ってパルスモーターとタイヤを接続するための軸受けを作った。

当初ワンボードマイコンを搭載したものを考えていたが、プログラム開発の面で、全員が持っているポケコン

に変更した。

ポケコンを搭載した場合、バッテリーの着脱を考慮してポケコンの台が回転するように設計している。

(2) 制御回路

図3にモーター駆動回路を、その回路をプリント基板にしたものを図4に示す。

モーター駆動回路の作製については回路班の各人で分担して、それぞれがオリジナルのプリント基板を作った。

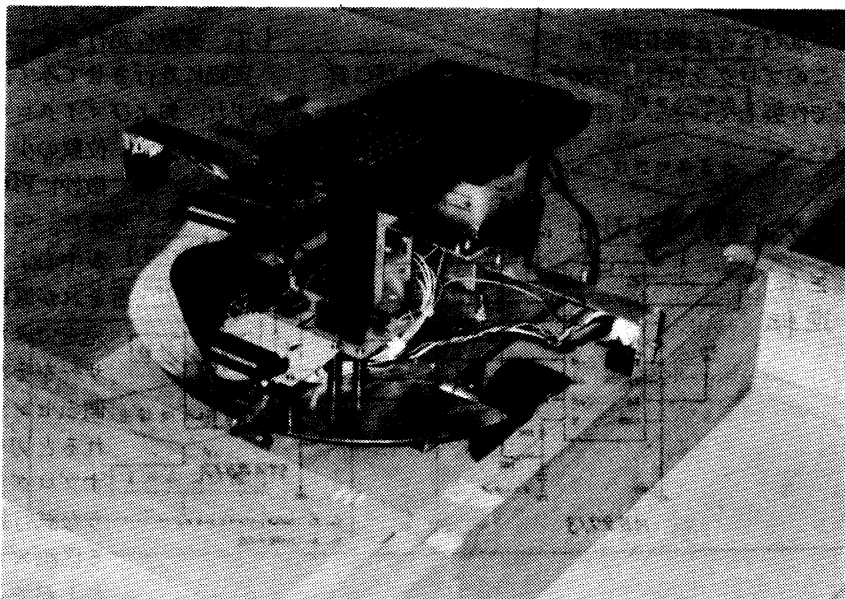


写真1 マイクロキャット

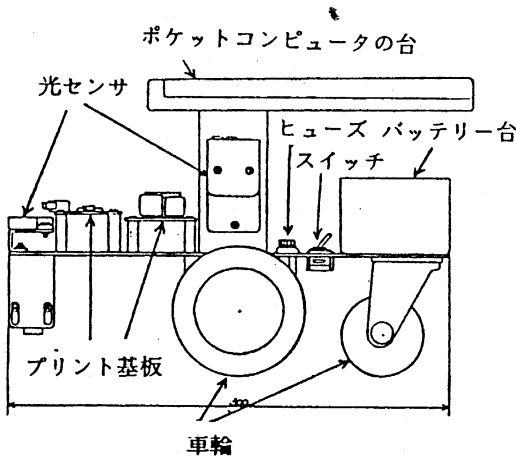


図1 マイクロキットの側面図

そして、個々の回路が完成したとき、ソフトウェア班がポケットコンピュータでパルスモーターを回転させるテストプログラムを作製し、その回路をテストし、動作確認を行い着実に研究を推し進めていった。

(3) ソフトウェア

図5に迷路を脱出するプログラムの流れ図を示す。

プログラムの概要は、右手法である。これは、まず右壁があるかどうか判定し、なければ右に進み、あれば前壁の有無を判定し、なければ前に進み、あれば次に左壁の有無を判定する。その左壁がなければ、左に進み、あればUターンして逆方向に進むことを繰り返す。

このプログラムは、Z80アセンブラおよびC言語で作製している。C言語プログラムを付録図1

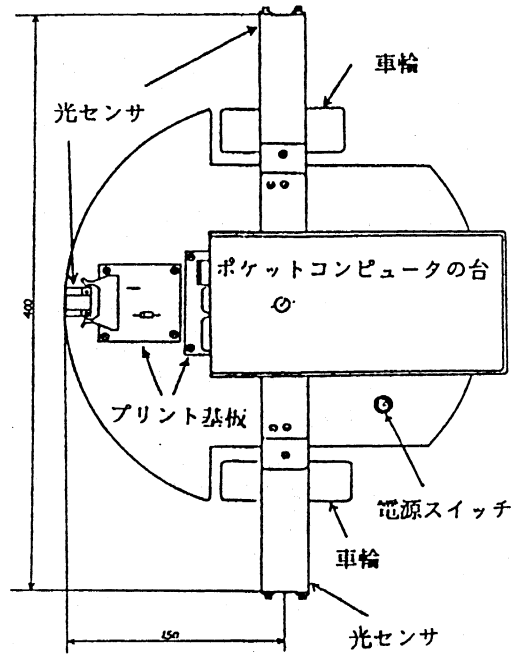


図2 マイクロキットの平面図

に示す。プログラムの作製手順は、トップダウン法で作成した。つまり、右手法のメインプログラムを作成してから個々のサブルーチンを作成していった。サブルーチンとしては、前に進んでいく前進サブルーチン、バックする後進サブルーチン、右に向きを変える右ターンサブルーチン、左に向きを変える左ターンサブルーチン、逆方向に向きを変えるUターンサブルーチンを作成した。そうして、実際に走行させてテストし調整した。

実際に走行させてみて、迷路板の置き方の少しのずれ、およびマイクロキットのスタート時の位置の少しのずれが迷路の脱出に微妙に影響することが分かった。また、走行時、マイクロキットが中心より少しづつずれていき、それが原因でゴールまでなかなか完走できなかったが、両端のセンサーをそれぞれ3個ずつにし、3個のセンサーのうち2個までがはずれると切り返しをして、中央に戻すプログラムを作成した。そのような試行錯誤の結果、図6に示す5区画×5区画の迷路を脱出することができた。

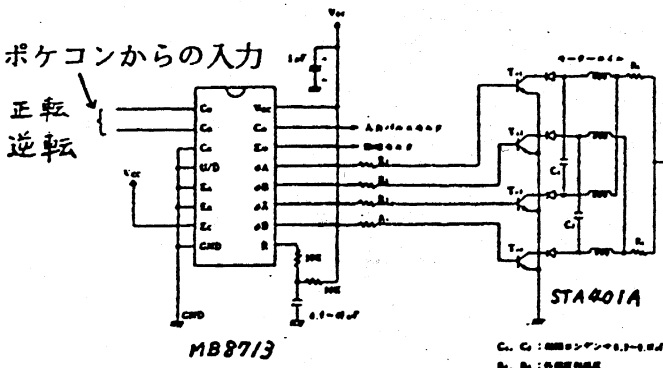


図3 パルスモーター駆動回路

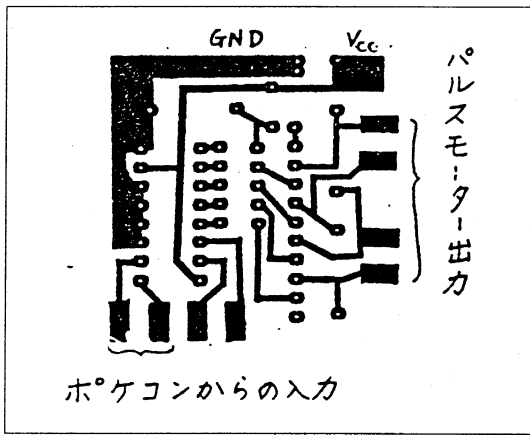


図4 モータ駆動回路のプリント基板

3. おわりに

今後は一度迷路を右手法でゆっくり走行し、各区画を記憶しておき、次に、最初の位置からスタートすれば最短経路をスピードアップして迷路を脱出するハードウェアとソフトウェアを検討してい

る。また、新規に相撲ロボットに取り組んでいる状況である。

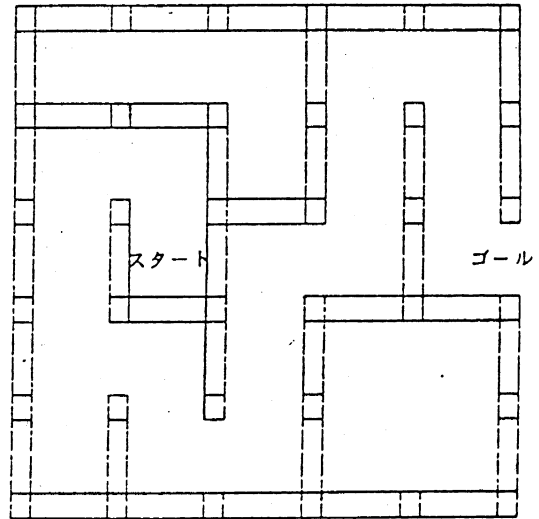


図6 迷路

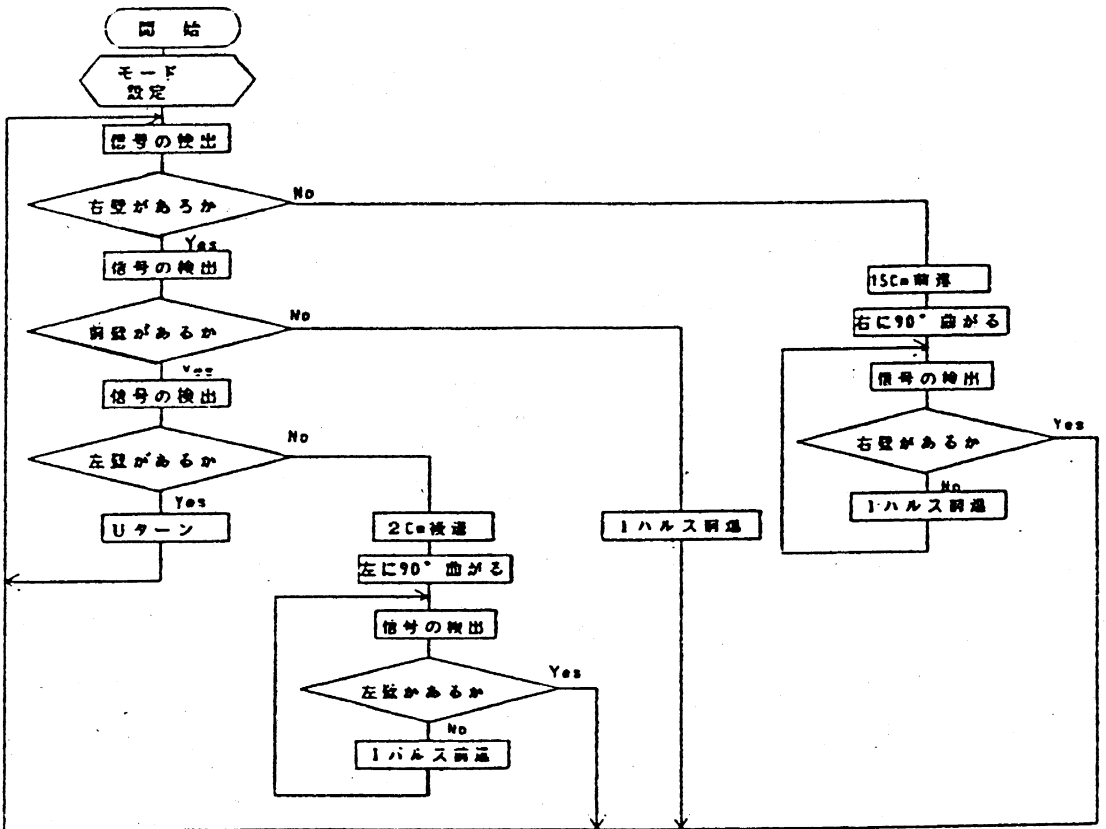


図5 迷路を脱出する流れ図

```

5/* CATK3.C*/
10#define PA 0x20
20#define PB 0x21
30#define PC 0x22
40#define CR 0x23
45 int right(int);
50 void timer(void);
60 void hozen(void);
70 int zensin(int);
80 int left(int);
90 void uturn(void);
100int bac(int);
110main()
120{
130int t,i,j,k,sw;
140outPort(CR,0x90);
150while(1){
160i=inPort(0x20);
170j=i & 0x7F;
180if((j & 0x01) || (j &
0x02) || (j & 0x04))
190{
200 if(j & 0x08)
210 {
220 if((j & 0x10)
|| (j & 0x20)|| (j & 0x
40))
230 {
240 bac(20);
250 uturn();}
260 else
270 {
280 bac(20);
290 left(179);
292 while(1){
294 j=inPort(0x
20);
296 if((j & 0x1
0)|| (j & 0x20)|| (j & 0
x40))
298 {break;}
300 else
302 {zensin(1
);}
304 }
306 }
310 }
312 else
314 {zensin(1);}
320 }
330 else
340 {
360 zensin(150);
370 right(179);
372 while(1){
374 j=inPort(0x20);
376 if((j & 0x01)|| (
j & 0x02)|| (j & 0x04))
378 {break;}
380 else
382 {zensin(1);}
384 }
390 }
400 zensin(1);
402 j=inPort(0x20);
404 if((j & 0x01)&& !((j
& 0x02)|| (j & 0x04)))
410 {left(90);
412 bac(15);
414 right(90);}
416 j=inPort(0x20);
418 if((j & 0x04)&& !((j
& 0x02)|| (j & 0x01)))
420 {right(90);
422 bac(15);
424 left(90);}
426 }
428void timer(void)
430{
440int t;
450for(t=0; t<7; t++)
460{ }
465 return;
470}
480int zensin(1)
490{
500int ki;
510for(k=0; k<1; k++)
520{
530 outPort(PB,0x00);
540 timer(); outPort(PB,
0x42);
550 timer();
560 timer();
570}
580return;
590}
600int left(1)
610{
620int ki;
630for(k=0; k<1; k++)
640{
650 outPort(PB,0x00);
660 timer();
670 outPort(PB,0x82);
680 timer();
690}
700return;
710}
720int right(1)
730{
740int ki;
750for(k=0; k<1; k++)
760{
770 outPort(PB,0x00);
780 timer();
790 outPort(PB,0x41);
800 timer();
810}
820return;
830}
840int bac(1)
850{
860int ki;
870for(k=0; k<1; k++)
880{
890 outPort(PB,0x00);
900timer();
910 outPort(PB,0x81);
920 timer();
930}
873return;
950}
960void uturn(void)
970{
972int ki;
974for(k=0; k<360; k++)
1000{
1110outPort(PB,0x00);
1120timer();
1130outPort(PB,0x41);
1140timer();
1150}
1160return;
1170}
1180 void hozen(void)
1190 {
1200 int k;
1210 for(k=0; k<15; k++)
1220 {
1230 outPort(PB,0x00);
1240 timer();
1250 outPort(PB,0x42);
1260 timer();
1270 }
1280 return;
1290}

```

付録図1 プログラム