

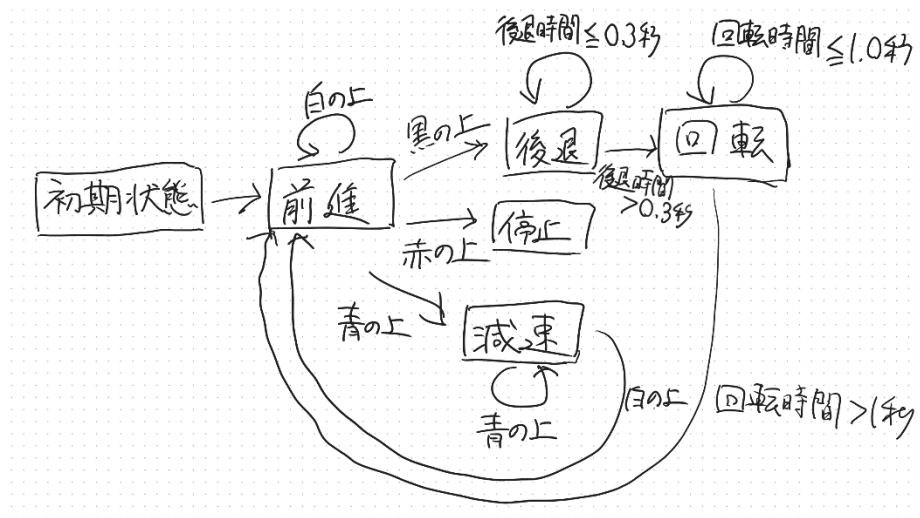
システム設計第13回 ロボット実験2

学籍番号 6122011908

名前 細江健人

令和6年7月14日 作成

レポート 8.2.1 課題 8.2.1 で作成したプログラムを報告せよ。状態遷移図を示せ。



作成した状態遷移図

以下が課題 8.2.1 で作成したプログラムである。

```

1  #include <ZumoMotors.h>          //モータのライブラリ
2  #include <Adafruit_TCS34725.h>  //カラーセンサのライブラリ
3
4  #define WHITE    0 //白
5  #define BLACK    1 //黒
6  #define RED      2 //赤
7  #define BLUE     3 //青
8  #define OTHER    4 //その他
9  #define INIT     10 //初期状態
10 #define FORWARD  11 //直進
11 #define ROTATE    12 //回転
12 #define STOP      13 //停止
13 #define SLOW      14 //減速
14
15 //モータのglobal関数
16 ZumoMotors motors;
17 int motorL, motorR; //モータ速度
18
19 //カラーセンサのglobal関数
20 Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_2_4MS, TCS34725_GAIN_60X);
21 float r, b, g;
22
23 const unsigned int r_min = 58, g_min = 83, b_min = 83;
24 const unsigned int r_max = 255, g_max = 255, b_max = 255;
25 int color; //色
26
27 //状態遷移のglobal関数
28 int mode; //ロボットの状態
29 unsigned long start_time; //時間計測の開始時間
30
31 int identify_color(int r, int g, int b) {
32     if (200 < r && 200 < g && 200 < b) {
33         return WHITE; //白
34     } else if (r < 20 && g < 20 && b < 25) {
35         return BLACK; //黒
36     } else if (90 < r && g < 20 && b < 20) {
37         return RED; //赤
38     } else if (r < 20 && g < 92 && b < 70) {
39         return BLUE; //青
40     } else {
41         return OTHER; //その他
42     }
43 }
44
45 void task() {
46     switch (mode) {
47         case INIT: //初期状態
48             mode = FORWARD;
49             break;
50         case FORWARD: //直進
51             motorL = motorR = 150;
52             if (color == BLACK) { //黒を踏んだら回転
53                 mode = ROTATE;
54                 start_time = millis();
55             }
56             if (color == RED) { //赤を踏んだら停止
57                 mode = STOP;
58             }
59             if (color == BLUE) { //青を踏んだら減速
60                 mode = SLOW;
61             }
62         break:
63     }
64
65     case ROTATE: //回転
66         if (millis() - start_time < 300) { //0.3秒後退
67             motorL = motorR = -150;
68         } else if (millis() - start_time < 1300) { //1秒回転
69             motorL = -150; motorR = 150;
70         } else {
71             mode = FORWARD;
72         }
73         break;
74     case STOP: //停止
75         motorL = motorR = 0; //スピードを0にする
76         break;
77     case SLOW: //減速
78         motorL = motorR = 100;
79         if (color == BLACK) { //黒を踏んだら回転
80             mode = ROTATE;
81             start_time = millis();
82         }
83         if (color == RED) { //赤を踏んだら停止
84             mode = STOP;
85         }
86         if (color == WHITE) { //白を踏んだら前進
87             mode = FORWARD;
88         }
89         break;
90     }
91 }
92
93 void setup() {
94     Serial.begin(9600);
95     tcs.begin();
96     mode = INIT;

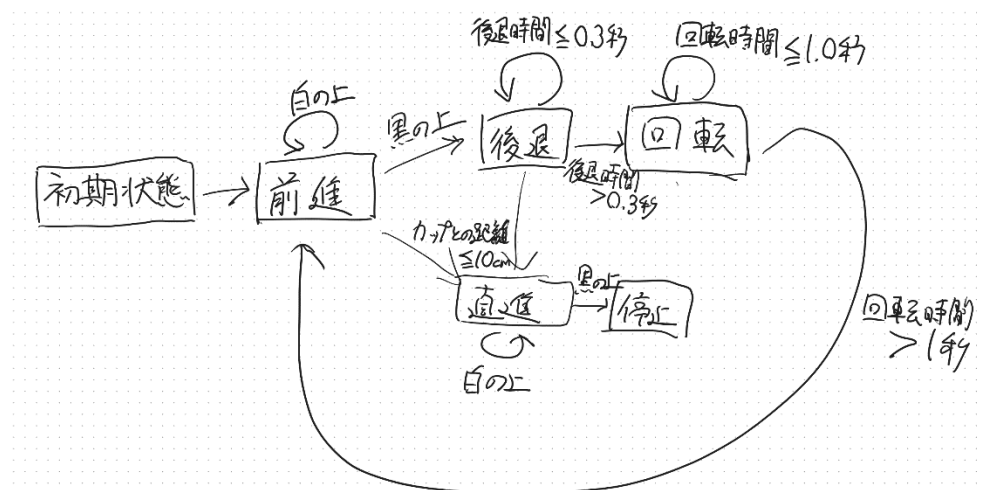
```

```

93 void setup() {
94   Serial.begin(9600);
95   tcs.begin();
96   mode = INIT;
97   motorL = motorR = 0;
98 }
99
100 void loop() {
101   //モータを駆動
102   motors.setSpeeds(motorL, motorR);
103
104   //色を計測
105   tcs.getRGB(&r, &g, &b);
106   r = map(r, r_min, r_max, 0, 255);
107   g = map(g, g_min, g_max, 0, 255);
108   b = map(b, b_min, b_max, 0, 255);
109
110   color = identify_color(r, g, b);
111   //Serial.println(color); //task()をコメントアウトし、ロボットを色の上に乗せて正しく識別できているか確認できる
112   Serial.print("r_min:"); Serial.println(r);
113   Serial.print("g_min:"); Serial.println(g);
114   Serial.print("b_min:"); Serial.println(b);
115   task(); //演習8.2.1の主な処理をする関数
116 }
117

```

レポート 8.2.2 課題 8.2.2 で作成したプログラムを報告せよ。 状態遷移図を示せ。



作成した状態遷移図

以下が課題 8.2.2 で作成したプログラムである。

```

1 #include <ZumoMotors.h> //モータのライブラリ
2 #include <Adafruit_TCS34725.h> //カラーセンサのライブラリ
3
4 #define WHITE 0 //白
5 #define BLACK 1 //黒
6 #define RED 2 //赤
7 #define BLUE 3 //青
8 #define OTHER 4 //その他
9 #define INIT 10 //初期状態
10 #define FORWARD 11 //直進
11 #define ROTATE 12 //回転
12 #define STOP 13 //停止
13 #define CATCH 14 //捕獲
14
15 //モータのglobal変数
16 ZumoMotors motors;
17 int motorL, motorR;
18
19 //カラーセンサのglobal変数
20 Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_2_4MS, TCS34725_GAIN_60X);
21 float r, b, g;
22 const unsigned int r_min = 50, g_min = 83, b_min = 83;
23 const unsigned int r_max = 255, g_max = 255, b_max = 255;
24 int color; //色
25
26 //超音波センサのglobal変数
27 const int trig = 2; //Trigピン
28 const int echo = 4; //Echoピン
29 int dst; //距離(cm)
30
31 //状態遷移のglobal変数
32 int mode; //ロボットの状態 (初期状態/直進/回転/停止)
33 unsigned long start_time; //時間計測の開始時間
34
35 int identify_color(int r, int g, int b) //色の判定
36 {
37     if (200 < r && 200 < g && 200 < b) {
38         return WHITE; //白
39     } else if (r < 20 && g < 20 && b < 25) {
40         return BLACK; //黒
41     } else if (90 < r && g < 20 && b < 20) {
42         return RED; //赤
43     } else if (r < 20 && g < 92 && 70 < b) {
44         return BLUE; //青
45     } else {
46         return OTHER; //その他
47     }
48 }
49
50 int distance() //距離を計測する関数
51 {
52     unsigned long interval; //Echoのバース幅(μs)
53     int dst; //距離(cm)
54
55     //10μsのバースを超音波センサのTrigピンに出カ
56     digitalWrite(trig, HIGH);
57     delayMicroseconds(10);
58     digitalWrite(trig, LOW);
59
60     //5767μs(100cm)、超音波が反射してこなければタイムアウトしてエラー値0を返す
61     //Echo信号がHIGHである時間(μs)を計測
62     interval = pulseIn(echo, HIGH, 5767);
63
64     dst = (0.61 * 25 + 331.5) * interval / 10000 / 2; //距離(cm)に変換
65     if (dst == 0) {dst = 100;} //エラー値0 (超音波が反射してこない) は100cmを検出したことにする
66     delay(60); //trigがHIGHになる間隔を60ms以上空ける (超音波センサの仕様)
67     return dst;
68 }
69
70 void task() {
71     switch (mode) {
72         case INIT: //初期状態
73             mode = FORWARD;
74             break;
75         case FORWARD: //直進
76             motorL = motorR = 150;
77             //黒を踏んだら回転
78             if (color == BLACK) {
79                 mode = ROTATE;
80                 start_time = millis();
81             }
82             //距離が15cm未満なら停止
83             if (0 < dst && dst < 10) {
84                 mode = CATCH;
85             }
86             break;
87         case ROTATE: //回転
88             if (millis() - start_time < 300) { //0.3秒後退
89                 motorL = motorR = -150;
90             } else if (millis() - start_time < 1300) { //1秒回転
91                 motorL = -150; motorR = 150;
92             } else {
93                 mode = FORWARD;
94             }
95             break;
96         case STOP: //停止
97             motorL = motorR = 0;
98             break;
99         case CATCH:
100             motorL = motorR = 150;
101             if (color == BLACK) { //黒を踏んだ場合停止
102                 mode = STOP;
103             }
104             break;
105     }
106 }
107
108 void setup() {
109     Serial.begin(9600);
110     tcs.begin();
111     pinMode(trig, OUTPUT); //trigを出力ポートに設定
112     pinMode(echo, INPUT); //echoを入力ポートに設定
113     mode = INIT;
114     motorL = motorR = 0;
115 }
116
117 void loop() {
118     //モータを駆動
119     motors.setSpeeds(motorL, motorR);
120     //色を計測
121     tcs.getRGB(&r, &g, &b);
122     r = map(r, r_min, r_max, 0, 255);
123     g = map(g, g_min, g_max, 0, 255);
124     b = map(b, b_min, b_max, 0, 255);
125     color = identify_color(r, g, b);
126     //Serial.println(color); //task()をコメントアウトし
127     //距離を計測
128     dst = distance();
129     //Serial.println(dst);
130     task(); //演習9.2.2の主な処理をする関数
131 }

```

レポート 8.2.3 課題 8.2.3 で作成したプログラムを報告せよ。 静止しているものと動いているものを、どのように判定したか説明せよ。

以下が課題 8.2.3 で作成したプログラムである。

```

1  #include <ZumoMotors.h>           //モータのライブラリ
2  #include <Adafruit_TCS34725.h>    //カラーセンサのライブラリ
3
4  #define WHITE    0 //白
5  #define BLACK    1 //黒
6  #define RED      2 //赤
7  #define BLUE     3 //青
8  #define OTHER    4 //その他
9  #define INIT     10 //初期状態
10 #define FORWARD  11 //直進
11 #define ROTATE    12 //回転
12 #define STOP      13 //停止
13 #define CATCH     14 //捕獲
14 #define CHECK     15 //確認
15 #define FINISH    16 //終了
16
17 //モータのglobal変数
18 ZumoMotors motors;
19 int motorL, motorR;
20
21 //カラーセンサのglobal変数
22 Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_2_4MS, TCS34725_GAIN_60X);
23 float r, b, g;
24 const unsigned int r_min = 58, g_min = 83, b_min = 83;
25 const unsigned int r_max = 255, g_max = 255, b_max = 255;
26 int color; //色
27 const int buzzerPin=3;
28 //超音波センサのglobal変数
29 const int trig = 2; //Trigピン
30 const int echo = 4; //Echoピン
31 int dist;          //距離(cm)
32 int checkDist;
33 int moveCup;
34
35 //状態遷移のglobal変数
36 int mode;          //ロボットの状態 (初期状態/直進/回転/停止)
37 unsigned long start_time; //時間計測の開始時間
38
39 int identify_color(int r, int g, int b) //{色の判定の関数
40     if (200 < r && 200 < g && 200 < b) {
41         return WHITE; //白
42     } else if (r < 20 && g < 20 && b < 25) {
43         return BLACK; //黒
44     } else if (90 < r && g < 20 && b < 20) {
45         return RED; //赤
46     } else if (r < 20 && g < 92 && 70 < b) {
47         return BLUE; //青
48     } else {
49         return OTHER; //その他
50     }
51 }
52
53 int distance() {
54     unsigned long interval; //Echoのバース幅(μs)
55     int dst;                //距離(cm)
56
57     //10μsのバースを超音波センサのTrigピンに出力
58     digitalWrite(trig, HIGH);
59     delayMicroseconds(10);
60     digitalWrite(trig, LOW);
61
62     //5767μs(100cm)、超音波が反射してこなければタイムアウトしてエラー値0を返す
63     //Echo信号がHIGHである時間(μs)を計測
64     interval = pulseIn(echo, HIGH, 5767);
65
66     dst = (0.61 * 25 + 331.5) * interval / 10000 / 2; //距離(cm)に変換
67     if (dst == 0) {dst = 100;} //エラー値0 (超音波が反射してこない) は100cmを検出したことにする
68
69     delay(60); //trigがHIGHになる間隔を60ms以上空ける (超音波センサの仕様)
70     return dst;
71 }
72
73 void task() {
74     switch (mode) {
75     case INIT: //初期状態
76         mode = FORWARD;
77         break;
78     case FORWARD: //直進
79         motorL = motorR = 150;
80         //黒を踏んだら回転
81         if (color == BLACK) {
82             mode = ROTATE;
83             start_time = millis();
84         }
85         //距離が15cm未満なら停止
86         if (0 < dist && dist < 15) {
87             mode = CHECK;
88             start_time = millis();
89             checkDist=dist;
90         }
91     }
92     break;
93 case ROTATE: //回転
94     if (millis() - start_time < 300) { //0.3秒後退
95         motorL = motorR = -150;
96     } else if (millis() - start_time < 1300) { //1秒回転
97         motorL = -150; motorR = 150;
98     } else {
99         mode = FORWARD;
100     }

```

```

101         break;
102     case STOP: //停止
103         motorL = motorR = 0;
104         if (dist > 15) {
105             mode = ROTATE;
106         }
107         break;
108     case CATCH: //捕獲
109         motorL = motorR = 150;
110         if (color == BLACK) {
111             mode = FINISH;
112         }
113         if (dist > 15) {
114             mode = ROTATE;
115         }
116         break;
117     case CHECK: //動いていないかの確認
118         motorL = motorR = 0;
119         if (millis() - start_time < 1000) { //1秒間
120             moveCup = dist - checkDist;
121             if (moveCup < -3 || moveCup > 3) { //距離が3cm変わっていた場合
122                 tone(buzzerPin, 200); //ブザーを鳴らす
123                 mode = STOP;
124             }
125         }
126         else {
127             mode = CATCH;
128         }
129         break;
130     case FINISH: //停止
131         motorL = motorR = 0;
132         break;
133     }
}

131     motorL = motorR = 0;
132     break;
133 }
134 }
135 }
136 void setup() {
137     Serial.begin(9600);
138     tcs.begin();
139     pinMode(trig, OUTPUT); //trigを出力ポートに設定
140     pinMode(echo, INPUT); //echoを入力ポートに設定
141     mode = INIT;
142     motorL = motorR = 0;
143 }
144 }
145 void loop() {
146     //モータを駆動
147     motors.setSpeeds(motorL, motorR);
148
149     //色を計測
150     tcs.getRGB(&r, &g, &b);
151     r = map(r, r_min, r_max, 0, 255);
152     g = map(g, g_min, g_max, 0, 255);
153     b = map(b, b_min, b_max, 0, 255);
154     color = identify_color(r, g, b);
155     //Serial.println(color); //task()をコメントアウト
156
157     //距離を計測
158     dist = distance();
159     //Serial.println(dist);
160
161     task(); //演習9.2.2の主な処理をする関数
162 }
163 }

```

静止しているものと動いているものの判定としては、カップを発見したとき停止し、1秒間カップまでの距離を計測し続けた。そして距離が判定前と3cm以上変化した場合、動いていると判定し、そのまま停止を続け、変化しなかった場合は捕獲モードに移行するようにプログラムした。

レポート 8.2.4 発展課題 8.2.4 で作成したプログラムを報告せよ。 効率良くカップを運び出すために行った工夫について説明せよ。

以下が課題 8.2.4 で作成したプログラムである。

```

1  #include <ZumoMotors.h>           //モーターのライブラリ
2  #include <Adafruit_TCS34725.h>    //カラーセンサのライブラリ
3
4  #define WHITE    0 //白
5  #define BLACK    1 //黒
6  #define RED      2 //赤
7  #define BLUE     3 //青
8  #define OTHER    4 //その他
9  #define INIT     10 //初期状態
10 #define FORWARD  11 //直進
11 #define ROTATE    12 //回転
12 #define STOP      13 //停止
13 #define CATCH     14 //捕獲
14 #define CHECK     15 //確認
15 #define FINISH    16 //終了
16
17
18 //モータのglobal変数
19 ZumoMotors motors;
20 int motorL, motorR;
21
22 //カラーセンサのglobal変数
23 Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_2_4MS, TCS34725_GAIN_60X);
24 float r, b, g;
25 const unsigned int r_min = 50, g_min = 83, b_min = 83;
26 const unsigned int r_max = 255, g_max = 255, b_max = 255;
27 int color; //色
28
29 //超音波センサのglobal変数
30 const int trig = 2; //Trigピン
31 const int echo = 4; //Echoピン
32 const int buzzerPin=3;
33 int dist;          //距離(cm)

```

```

34 int checkDist;
35 int moveCup;
36 int count=0;
37
38 //状態遷移のglobal変数
39 int mode;          //ロボットの状態（初期状態／直進／回転／停止）
40 unsigned long start_time; //時間計測の開始時間
41
42 int identify_color(int r, int g, int b) { //色の判定の関数
43     if (200 < r && 200 < g && 200 < b) {
44         return WHITE; //白
45     } else if (r < 20 && g < 20 && b < 25) {
46         return BLACK; //黒
47     } else if (90 < r && g < 20 && b < 20) {
48         return RED; //赤
49     } else if (r < 20 && g < 92 && 70 < b) {
50         return BLUE; //青
51     } else {
52         return OTHER; //その他
53     }
54 }
55
56 int distance() {
57     unsigned long interval; //Echoのパルス幅(μs)
58     int dst;                //距離(cm)
59
60     //10μsのパルスを超音波センサのTrigピンに出力
61     digitalWrite(trig, HIGH);
62     delayMicroseconds(10);
63     digitalWrite(trig, LOW);
64
65     //5767μs(100cm)、超音波が反射してこなければタイムアウトしてエラー値0を返す
66     //Echo信号がHIGHである時間(us)を計測

```

```

67   interval = pulseIn(echo, HIGH, 5767);
68
69   dst = (0.61 * 25 + 331.5) * interval / 10000 / 2; //距離(cm)に変換
70   if (dst == 0) {dst = 100;} //エラー値0（超音波が反射してこない）は100cmを検出したことにする
71   delay(60); //trigがHIGHになる間隔を60ms以上空ける（超音波センサの仕様）
72
73   return dst;
74 }
75
76 void task() {
77   switch (mode) {
78     case INIT: //初期状態
79     mode = FORWARD;
80     break;
81     case FORWARD: //直進
82     motorL = motorR = 150;
83     //黒を踏んだら回転
84     if (color == BLACK) {
85       mode = ROTATE;
86       start_time = millis();
87     }
88     //距離が15cm未満なら停止
89     if (0 < dist && dist < 15) {
90       mode = CHECK;
91       start_time = millis();
92       checkDist=dist;
93     }
94   }
95   break;
96   case ROTATE: //回転
97   if (millis() - start_time < 300) { //0.3秒後退
98     motorL = motorR = -150;
99   } else if (millis() - start_time < 1300) { //1秒回転
100
101     motorL = -150; motorR = 150;
102   } else {
103     mode = FORWARD;
104   }
105   break;
106   case STOP: //停止
107   tone(buzzerPin,600);
108   motorL = motorR = 0;
109   if (dist > 15) {
110     mode = ROTATE;
111   }
112   break;
113   case CATCH://カップを捕獲するモード
114   tone(buzzerPin,800);
115   motorL = motorR = 150;
116   if (color == BLACK) {//色が黒だった場合
117     mode = FINISH;
118     start_time = millis();
119     count++;//カップの捕獲数を増加
120   }
121   if (dist > 15) {
122     mode = ROTATE;
123   }
124   break;
125   case CHECK://カップが静止しているか確認するモード
126   tone(buzzerPin,200);
127   motorL = motorR = 0;
128   if (millis() - start_time < 200) { //1秒間
129   }
130   else if (millis() - start_time < 1000) { //1秒間
131     moveCup=dist-checkDist;
132     if(moveCup<-2||moveCup>2){//カップが2cm以上動いていた時
133       mode=STOP;
134     }
135   }
136   else{
137     mode=CATCH;//捕獲モード
138   }
139   break;
140   case FINISH://カップを全て捕獲したか判定するモード
141   if(count==3){
142     motorL = motorR = 0; //停止
143     tone(buzzerPin,900);
144   }
145   else if (millis() - start_time <500) { //0.5秒後退してカップの捕獲を継続する
146     motorL = motorR = -150;
147   } else if (millis() - start_time < 1000) { //1秒回転
148     motorL = -150; motorR = 150;
149   } else {
150     mode = FORWARD;
151   }
152 }
153
154 void setup() {
155   Serial.begin(9600);
156   tcs.begin();
157   pinMode(trig, OUTPUT); //trigを出力ポートに設定
158   pinMode(echo, INPUT); //echoを入力ポートに設定
159   mode = INIT;
160   motorL = motorR = 0;
161 }
162
163 void loop() {
164
165   //モータを駆動
166   motors.setSpeeds(motorL, motorR);
167
168   //色を計測
169   tcs.getRGB(&r, &g, &b);
170   r = map(r, r_min, r_max, 0, 255);
171   g = map(g, g_min, g_max, 0, 255);
172   b = map(b, b_min, b_max, 0, 255);
173   color = identify_color(r, g, b);
174   //Serial.println(color); //task()をコメントアウト
175
176   //距離を計測
177   dist = distance();
178   //Serial.println(dist);
179
180   task(); //演習9.2.2の主な処理をする関数
181 }

```

工夫した点

カップを3つ捕獲するまで動作を終了しないようにカップを黒線の外に運んだあと後退して回転し、次のカップを探索するようなプログラムを実装した。回転時にカップにロボットの先端部分が当たってしまい、ロボットがカップが動いていると判断して停止してしまったことで遅延してしまうことがあったため、回転時は動いている判定を行わないようにした。