

マイクロマウス HM StarterKit パート 1

『基礎知識編』



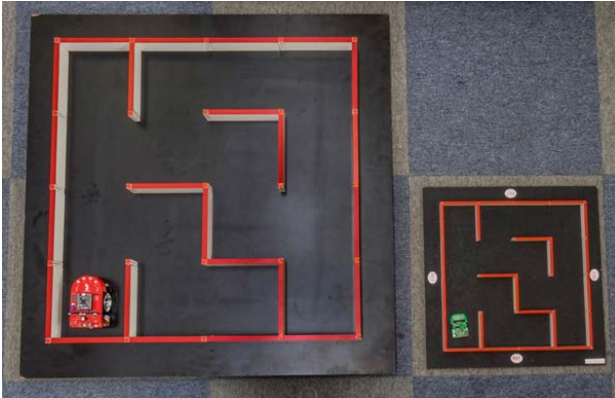
目次

| | |
|-----------------------------------|----|
| 1. マイクロマウス競技とは | 3 |
| 1.1 マイクロマウス競技の歴史 | 3 |
| 1.2 マイクロマウス競技内容 | 3 |
| 2. マイクロマウスの基本的なハードウェア構成 | 4 |
| 2.1 CPU ユニット | 4 |
| 2.2 センサユニット | 4 |
| 2.3 センサの種類 | 4 |
| 2.4 スイッチ | 5 |
| 2.5 赤外線・可視光センサ | 5 |
| 2.6 モータユニット | 6 |
| 2.7 ステッピングモータ仕様 | 6 |
| 2.8 DC モータ仕様 | 7 |
| 2.9 エンコーダ(フォトインタラプタ+エンコーダ板) | 8 |
| 2.10 通信プロトコル | 8 |
| 2.11 電源ユニット | 9 |
| 2.12 バッテリ | 9 |
| 2.13 リチウムイオンポリマ電池 | 9 |
| 2.14 ニッケル水素電池 | 10 |
| 3. 迷路解析の手法と考え方 | 11 |
| 3.1 マッピングについて | 11 |
| 左手法 | 11 |
| 3.3 拡張左手法 | 12 |
| 3.4 求心法 | 13 |
| 3.5 トレモー法 | 13 |
| 3.6 足立法 | 13 |
| 3.7 最短経路の求め方 | 14 |
| 4. 備考 | 15 |
| 5. 改版履歴 | 15 |
| 製造元 | 15 |
| 製品に関するお問い合わせ | 15 |

1. マイクロマウス競技とは

1.1 マイクロマウス競技の歴史

マイクロマウスは、もともと米国 IEEE(米国電気電子学会)が技術チャレンジして 1977 年に提唱され、人工知能とマイクロコンピュータ(以下、マイコン)の実世界応用(要するに体をもってロボットとして動き回る)を教えるために考案されたそうです。(これは、筆者がアメリカに行ったときに、全米で 2 番目に売られている人工知能の教科書で知られている George Luger 先生に、当時実際にマイクロマウスの大会を始めたときの話を聞いた話です)。実際にマイクロマウスは、迷路を解くという形式知(パスプランニングという人工知能)とロボットを動かすという身体知(実体があってはじめて実現できる制御などを含めた人工知能)の両方を考えて融合した内容になっています。日本では 1980 年に第 1 回全日本マイクロマウス大会が開催され 35 年を過ぎてなお、基本ルールが変わっていません。これは、毎年マイコンも進化し、人工知能も進化し続けているので、課題を変える必要がないからです。しかし、1980 年代以降、アメリカではすぐに大型計算機や



ワークステーションでの人工知能にシフトしてしまったこともあり、競技会としてマイクロマウスは、アメリカでは、今はあまり盛んではなく、一部で競技会が行われている程度になっています。海外のマイクロマウスについて詳しい説明は主催団体のニューテクノロジー振興財団に譲りますが、日本の全国大会が事実上の世界大会であるくらい、日本が一番盛んです。

現在では、マイクロマウスは迷路の大きさで競技が分類されています。18×18cm を 1 区画として 16×16 区画の大きさのクラシックマウス競技と、9×9cm を 1 区画として最大 32×32 区画の迷路(通常は 16×16 区画で全国大会のファイナルのみ 32×32 区画)のマイクロマウス競技(2009 年から開催)があります。

1.2 マイクロマウス競技内容

マイクロマウス競技では、完全自立型のロボットが 16×16 区画の迷路を走行します。スタート地点は、迷路の四隅のいずれかが決められた場所であり、ゴールは迷路中心の 4 区画です。競技では、制限時間以内に 5 回走行でき、1 回ごとにスタート地点から最初にゴール地点まで完走するまでのタイムを計測します。途中で止まるなどしてゴールできなかった場合は、その回はリタイアになります。5 回走行するか、制限時間が尽きたときに、走行した中の最短のタイムが記録して採用されます。一度も完走することができなかった場合、記録はリタイアとなります。マイクロマウスは、センサなどを使って迷路を探索し、完全自立走行しなくてはなりません。競技前、もしくは、競技中に迷路の情報をマイクロマウスに入力したり、プログラムを変更したりすることは認められていません。詳しいルールについては主催団体のニューテクノロジー振興財団のホームページ (<http://www.ntf.or.jp>) を参照してください。

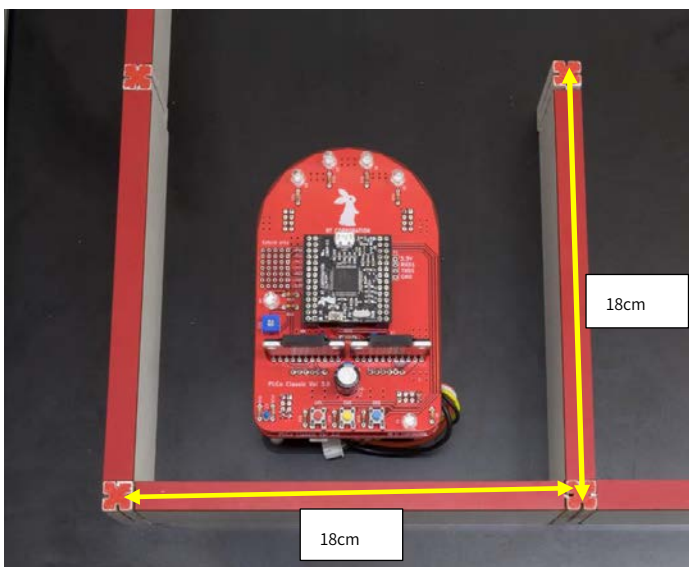


図 クラシック迷路一区間

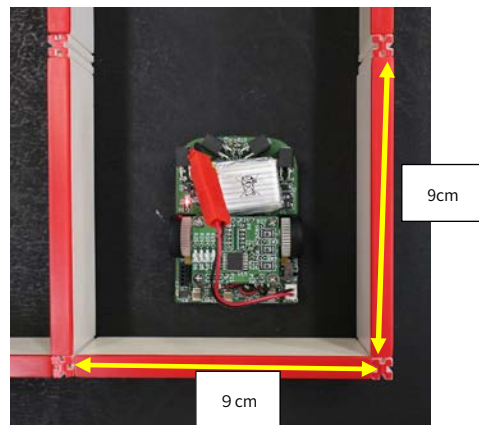


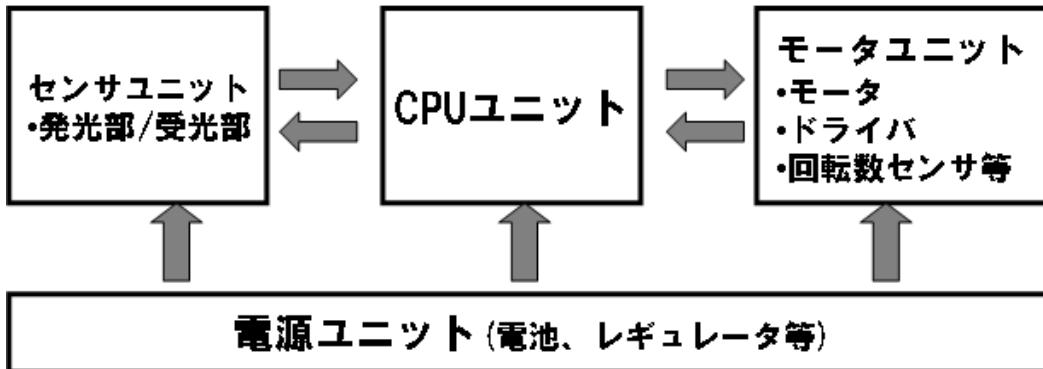
図 マイクロマウス迷路一区間

2. マイクロマウスの基本的なハードウェア構成

マイクロマウスは、コンパクトにまとめられたロボットです。ソフトウェア、ハードウェアの両方がバランスよく組み込まれたマイクロマウスは、現在活躍する技術者が必要とする技術的要素を含んだ構成となっています。

マイクロマウスは、CPU ユニット、センサユニット、モータユニット、電源ユニットなどで構成されています。

図 1 構成図



2.1 CPU ユニット

CPU ユニットでは、センサからの情報入力、迷路の解析、モータへの信号出力をつかさどります。その他、スイッチからの状態変化や電源管理など、さまざまな情報を統制し、制御を行います。ファームと呼ばれるソフトウェアによって動作します。

CPU ユニットに搭載される主な部品は、CPU、RAM、ROMなどのメモリ、スイッチ類などです。電源はCPUの電圧・電流に適合するように調整して供給します。

製作するプログラムはこのCPUごとに開発環境が変わります。HM StarterKitの場合は、R5F5631PDDFL (RENESAS)というCPUを使用しています。

CPU¹ (シーピーユー) : ロボットの指令塔です。ROMに記録されているプログラムを読み出して演算・実行し、センサが感知した値により、モータを動かす指令を出します。

ROM²・RAM³ (ロム・ラム) : メモリという名のとおり、データを記録しておく部分です。ROMは電源を切っても内容が消えないメモリで、プログラムを書き込むために特殊なアプリケーション (ファームライターと呼ばれます) を使います。一方のRAMは、電源を切るとデータがリセット (内容が消える) されるメモリで、プログラムによって読み書きが可能です。従って、迷路探索などアルゴリズムを含めた基本プログラムはROMに書きこみますが、実行中に変化する迷路のデータや変数の値などは、逐次RAMに記録して使用します。

2.2 センサユニット

ロボットが周囲の環境情報を読み取るための装置を、総じてセンサと呼びます。マイクロマウスでは、各区画の壁情報 (壁の有無、壁の切れ目など) を読み取るためにセンサを使用します。マイクロマウスで利用されるセンサの代表的な種類と用途を表にまとめますので、参考にしてください。

2.3 センサの種類

表 1 主なセンサー一覧

| センサ | 用途 |
|-------------------------|--------------------------|
| スイッチ | プログラムのモード変更 壁にぶつかったことの検出 |
| 赤外線・可視光センサ | 壁読み取り用 |
| エンコーダ(フォトインタラプタ+エンコーダ板) | タイヤの回転数検出 |
| ジャイロセンサ | 姿勢制御・方向検出 |

¹ Central Processing Unit の略

² Read Only Memory の略

³ Random Access Memory の略

2.4 スイッチ

CPU ユニット上のスイッチも分類としてはセンサに含まれます。基本的な使い方としてプログラムモードの切り替えなどに使います。また、ロボットの側面に取り付けたりして、壁との衝突を検出するためにも使います。

2.5 赤外線・可視光センサ

壁の読み取り用に使います。発光素子と受光素子やフィルタ回路によって構成されていて、これを数セット使います。

●発光素子と受光素子

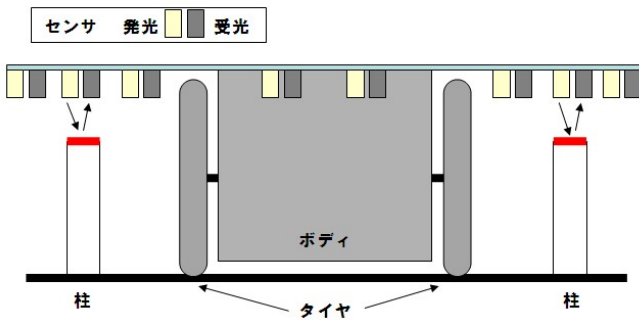
発光素子としては赤外線 LED や可視光 LED が、受光素子としては LED に対応した波長のフォトトランジスタなどがよく使われます。横壁式では、センサの発光部から光を壁の側面に反射させ、受光部で受信します。受信した光の量によって、センサから壁までの距離を測ります。センサで受光した値⁴を元に、壁の有無の検出や姿勢制御⁵を行ないます。

●ハイパスフィルタ

発光素子から赤外線を出しっぱなしにして、その光を受光素子で読もうとすると、なかなか正常に動作しません。これは、受光素子が赤外線だけではなく照明など周りの光（外乱光と呼びます）も一緒に読み取ってしまうことが最大の原因です。この対策として、受光側の回路にハイパスフィルタ回路を組み込む方法があります。ハイパスフィルタとは、電圧の急激な変化のみを通過させる回路です。発光をパルス状にして変化を急にすることで、緩やかに変化する外乱光をカットし、赤外線の値だけを読み取ることができます。

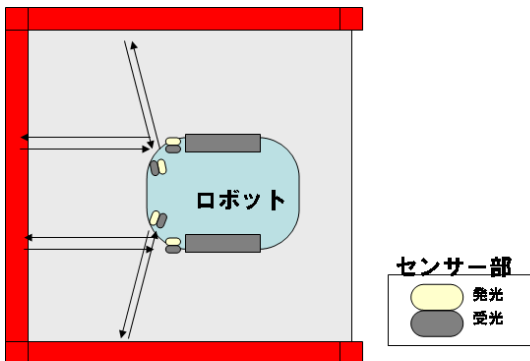
このセンサを設置する方法としては、上壁式と横壁式があります。上壁式は壁の上側の赤い部分を検出するためのもの、横壁式は壁の横側の白い部分を検出するためのものです。

上壁式の設置方法については以下の説明図を参照してください。



赤外線または可視光を上部から照射し、壁を探します。

図 2 上壁式のマウスを迷路正面から見た図



赤外線または可視光を横部から照射し、壁を探します。一般的には前壁方向に2つ、横壁方向に左右1つずつ設置します。

上壁式は、壁をラインのようにみることができるため、わかりやすく初心者向きです。しかしながら、現在では高速なマウスを作る場合は、ほとんど横壁式が採用されています。HM StarterKit も横壁式センサを採用しています。横壁式の設置方法については以下の説明図を参照してください。

横壁用センサは 15°~45°くらいで設置
HM StarterKit は 30°です。

(角度は迷路の区画に対するロボットの大きさによる)

15~45°くらい

図 3 横壁式マウスを上から見た図

⁴ センサの種類にもよりますが、一般的にはセンサが出力する電圧

⁵ ロボットが壁に対してまっすぐ進むようにすること

2.6 モータユニット

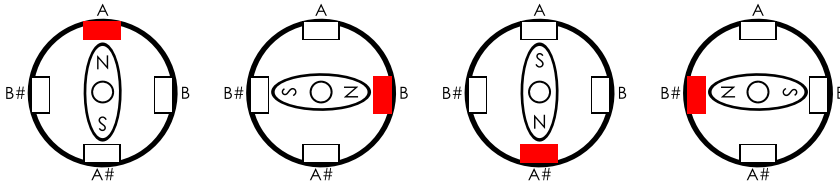
モータは、ロボットを駆動するためのアクチュエータ（駆動装置）の1つです。後述の通り、使用目的によっていくつかの種類を使い分けます。

2.7 ステッピングモータ仕様

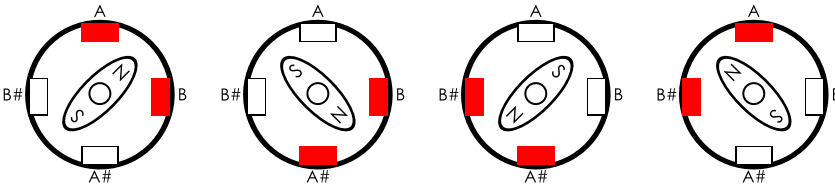
マイクロマウスに広く使われているモータです。パルスを送るたびに決まった角度ずつ軸が回るので、回転角の制御が比較的簡単です。しかし、あまり高トルクには耐えられない、一般的に大きくて重い、などの欠点もあります。モータに指令したパルス数で回転角度がわかるため、回転数を計測するセンサを取り付けずにすむので、初心者には使いやすいモータです。

ステッピングモータは、パルス型の電力に同期して動作するモータで、簡単な構成で正確な位置決めが可能なモータです。巻線への電流の与え方を変えることにより、特性を変えることができ、モータを駆動する励磁の方式も以下のようなものがあります。ステッピングモータのことを、ステップと呼ぶこともあります。

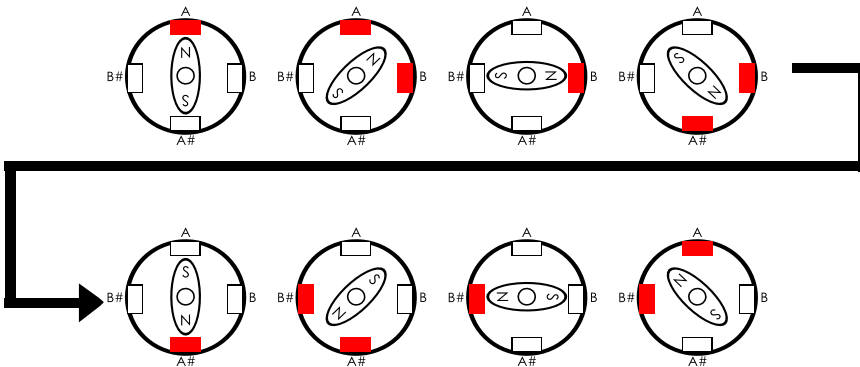
1 相励磁：常に巻線1相のみに電流を流す。位置決め精度が良い。



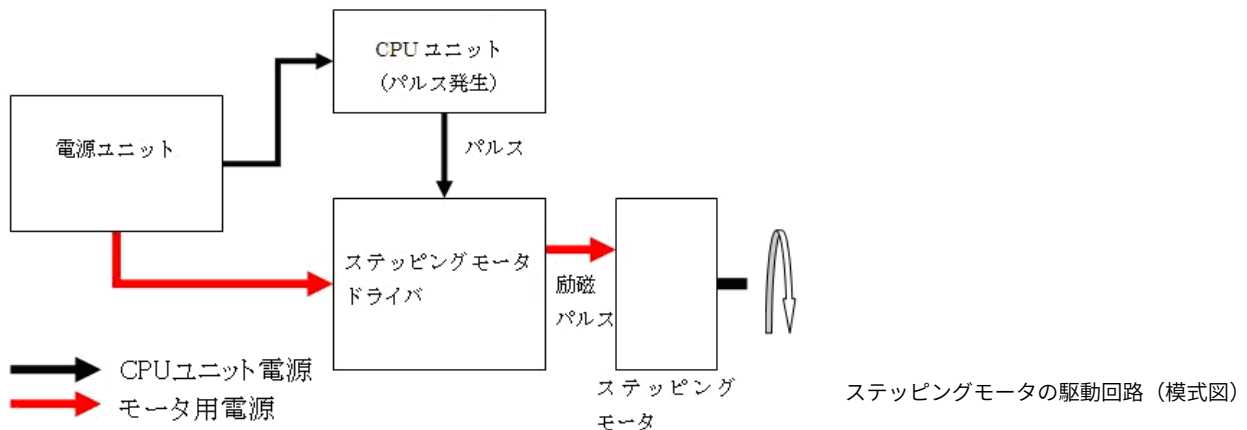
2 相励磁：2相に電流を流す。1相励磁の約2倍の出力トルクが得られます。



1-2 相励磁：1相と2相を交互に切り替えて電流を流します。1相励磁・2相励磁の場合のステップ角度の半分にすることができるので、滑らかな回転が得られます。HM StarterKitはこの1-2 層励磁を採用しています。



マイクロマウスでは、細かいステップ角とトルクを得るために1-2相励磁を用いるのが一般的です。マイクロマウス本体の重量も考慮にいて、脱出トルクの適切なステッピングモータを選びましょう。



一般的にステッピングモータを駆動する場合は、パルスを発生する CPU ユニット、ステッピングモータドライバ、ステッピングモータ、電源ユニットで回路を構成します。電源供給はステッピングモータドライバと CPU ユニットに別々に行い、パルスは CPU で発生させます。パルス発生の間隔を調整することで、加減速を行い、滑らかに駆動させることが可能です。(台形駆動とも言います)

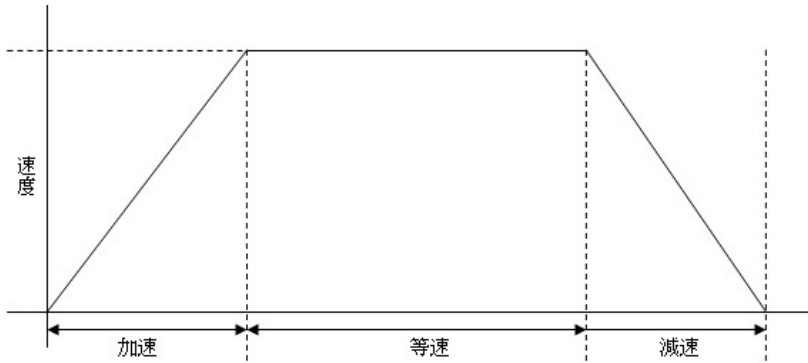


図 4

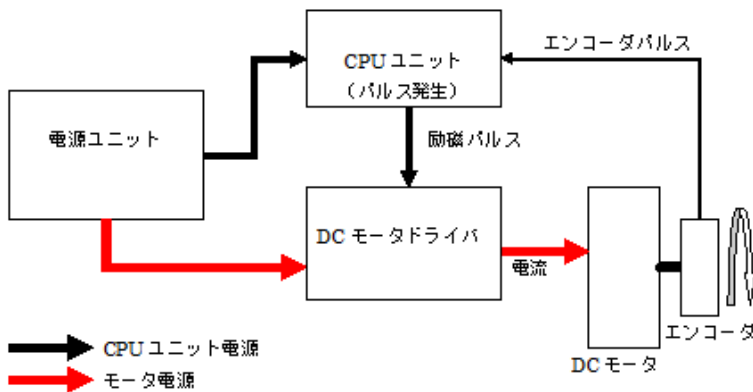
パルスによりローターが回転する方式ですので、スイッチを ON/OFF させる間隔でスピードを変えることが可能です。そのため、加速部分でパルス発生数を多く(最初の間隔よりどんどん短い間隔にして)してモータの回転速度を早めます。等速部分ではパルス数は一定間隔で発生させ、回転を維持します。そして、減速部分ではパルス発生数を少なく(間隔を間遠に)し、最終的にはパルス発生を 0 にします。

2.8 DC モータ仕様

DC モータは、一般的に、モータドライバ、DC モータ、エンコーダなどで構成されます。マイコンから PWM 信号という周期的な信号をモータドライバに送り、モータドライバから DC モータに電流を流すことによって制御しています。DC モータをマイクロマウスで使うには、位置決めをするために回転数や角度を別のセンサ(エンコーダなど)で取得して制御する必要があり、ステッピングモータに比べると構成するパーツ数が増えるために難しいといわれています。しかしながら、一度技術を習得してしまえば、ステッピングモータよりもはるかに使いやすいモータです。

DC モータは、電流を流す向きによって軸の回転方向が変わります。モータドライバには 2 つ以上の入力信号ポートがあり、それぞれに PWM 信号を与えることによって正方向(CW (Clockwise) クロックワイズ、「シーダブリュー」と呼ぶのが一般的)、逆方向(CCW(Counter ClockWise)カウナークロックワイズ、「シーシーダブリュー」と呼ぶのが一般的)に回転を制御することができます。

また、誤差を少なく位置決めするためには、PID 制御と呼ばれる制御方法が用いられます。P は比例に応じた変化量を示す変数で、I は積分的要素、D は微分要素をさします。積分的要素を除いた PD 制御などもよく使います。PID 制御については専門書がいくつか出ていますので、適切な参考書を探すことをお勧めします。



DC モータの駆動原理

2.9 エンコーダ(フォトインタラプタ+エンコーダ板)

●エンコーダ

車輪やモータの回転数を計測するセンサです。1相式、2相式などの読み取り方式や、光学式、磁気式など様々なタイプがあります。

●フォトインタラプタ+エンコーダ板

エンコーダ板等を使用して、スリットの数を読み取ります。フォトインタラプタの種類によって読める速度が異なります。モータやタイヤの回転数に適合する速度のものを選ぶと良いでしょう。

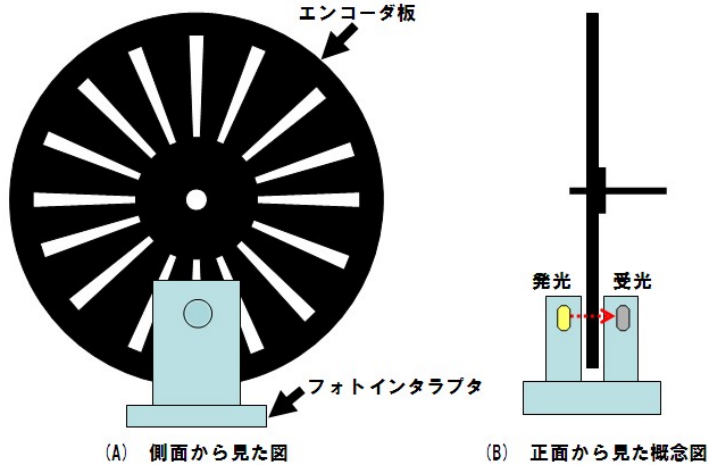


図5 フォトインタラプタとエンコーダ板による回転数計測

スリットが通過する回数をカウント（受光部が発するパルスをマイコンでカウントする）

●ホール素子+磁石（磁気式エンコーダ）

モータの回転軸や車軸などに磁石を取り付け、磁石の磁気をホール素子で検知することで、軸の回転数や角度を測定する方式です。一般に販売されている磁気式エンコーダはサイズがマウスに対して大きいため、多くの参加者が磁気式エンコーダを制作して使用している。HM StarterKitではこの方式を採用している。また、磁石を使用するため、モータやほかの機械部品、対となるホール素子との磁力干渉等が問題となることがある。

2.10 通信プロトコル

マウスをはじめ、多くのロボットではセンサ部やモータの駆動部など主要な機能がユニット単位で管理されています。CPUユニットから各ユニットに対して制御・命令を行うためにはユニット同士が何らかの通信手段で接続されていなければなりません。また多くの場合モジュール同士は電気回路上で結線されており、0(0V)と1(3.3V)の電気信号で通信が行われています。もちろん通信方法にも約束事があり、これを通信プロトコル(通信フォーマット)といいます。代表的なプロトコルとしてはRS232、I2C、SPIが用いられます。

●I2C

I2C(アイ・スクエアド・シー、アイツージー)は

図6のように1台のマスター(図中CPUユニット)とスレーブ(図中各ユニットA, B)で構成された1対多のユニット接続で通信が行えるプロトコルです。また、通信に用いる線は2本(SDAとSCL)で済むことが特徴です。携帯電話などさまざまな組み込み機器に使用されています。1本の線で多数のユニット同士を接続し、各々双方向の入出力を行うのでI2Cを扱う入出力ピンはオープンドレインという特別な状態とあるため、プルアップ抵抗が必要となります。

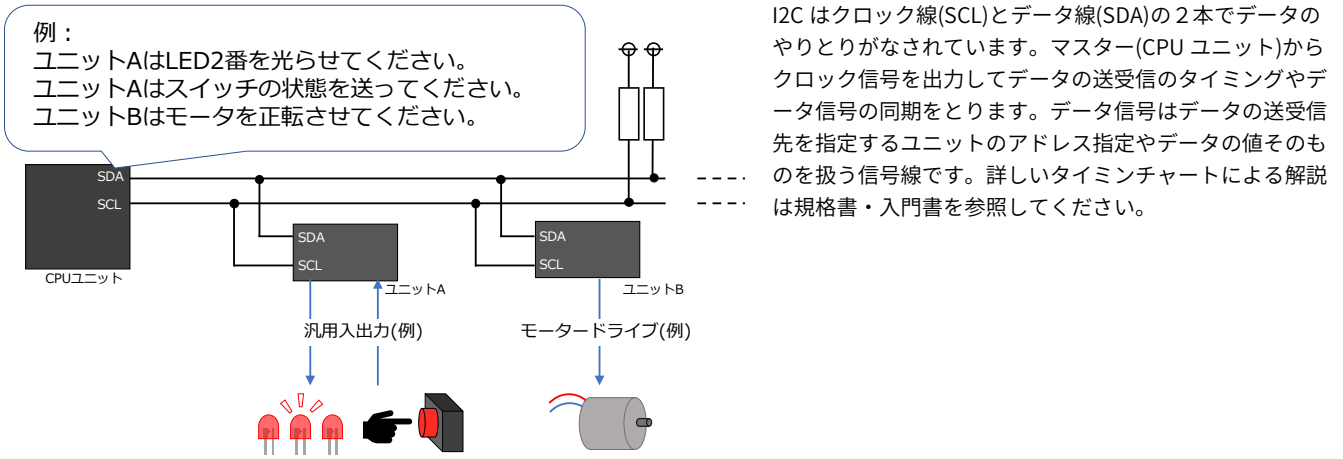


図 6 I2C によるユニット同士の接続例

● SPI

SPI も前述した I2C と同じように 1 台のマスターと複数のスレーブによる 1 対多のユニット接続が可能な通信プロトコルです。I2C と異なり 2 本のデータ線(MOSI と MISO)でデータの送受信を行います。また、通信ユニットの選択は SS(スレーブ・セレクト)ピンを HIGH または LOW の状態にさせることで指定を行うため、ユニットが 1 台増えるごとに 1 本信号線が増えますが、比較的高速な通信が可能となります。

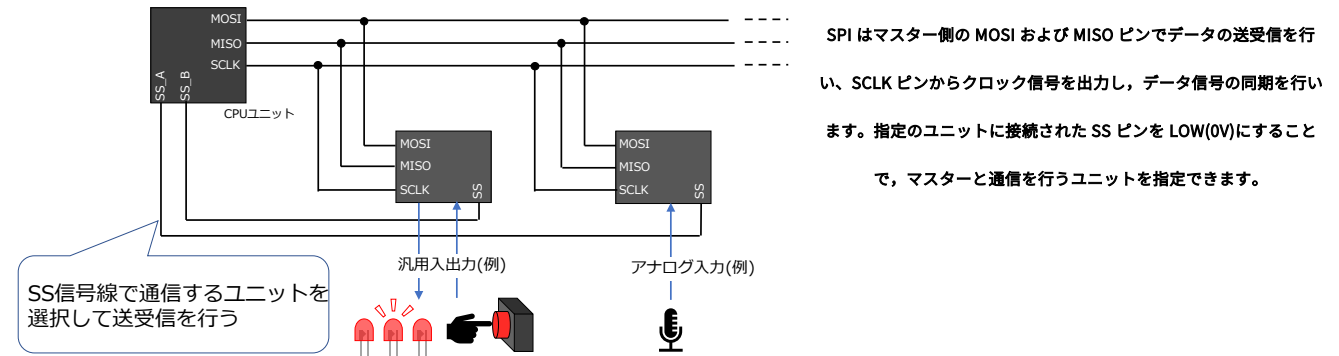


図 7 SPI によるユニット接続の例

2.11 電源ユニット

電源ユニットにはレギュレータが搭載されています。レギュレータは、電圧を変換するための IC、または回路です。一般的に、モータに供給する電源とマイコンに電源を供給するのは、バッテリーが同じでも電圧を変えるために系統を変えることがほとんどです。DC-DC コンバータを使う場合もありますが、一般的には三端子レギュレータと呼ばれる IC を使います。供給する電源電圧は、CPU やモータの定格を守りましょう。IC が焼けたり、モータの焼損の原因となります。

電源の系統を分ける場合は、CPU などの信号系統とモータ系統のグランドレベルを合わせます。しかしながら、モータ系統はノイズが発生しやすいので、ノイズ対策をする必要があります。(グランドレベルがノイズでゆれると信号にもノイズがのってしまいます。)

2.12 バッテリ

ロボットは電源が供給されなければ動きません。マウスの場合には外部電源を持つことは認められていませんので、本体に電池を積んで走行することになります。効率を考えれば 2 次電池⁶が良いでしょう。以下に、マウスにおいてよく使われる電池を紹介します。

2.13 リチウムイオンポリマ電池

リチウムイオンポリマ電池は、リポ電池と略されます。小さくて軽く、瞬間的に流せる電流も他の電池に比べて大きく、さらにメモリ効果もほとんど無いためロボット用途にはむいています。しかし他の電池に比べるとやや高価です。過充電やショートを起こすと発火・爆発することもあります。一般的なリチウムイオンポリマ電池は保護回路が入っていますが、取り扱いには細心の注意が必要です。リポ電池はその Cell の数で電圧の大きさが決まります。1Cell 平均 3.7V で、2Cell で 7.4V、3Cell で 11.1V と電圧が大きくなっていきます。(また、全ての電池の放電に関する単位に C を使います。C は容量に対する放電の比率を表し、1C ですと 1 倍、2C ですと 2 倍、3C ですと 3 倍の電流

⁶ 充電して何度も使える電池

を放電するという意味になります。) 充電するには専用の充電器を使ってください。(リポ電池は1Cで充電するのが良いでしょう。) 保管するには充電容量約80%の状態での保管が良いとされています。

また、1Cell 3.3Vを切ると過放電となり、使えなくなってしまいます。HM StarterKit で使われているリポ電池は1Cellなので、3.3V以下には絶対しないでください。サンプルプログラムでは、3.2V以下になるとブサがるように組み込まれていますので、ブサがなったら、速やかに開発を中断し、充電してください。

リチウムポリマ電池を利用するときは、注意事項をよく守って、過放電、過充電にならないように取り扱しましょう。

2.14 ニッケル水素電池

容量が大きく、マイクロマウスに積みやすいサイズであることから、よく使われています。

充電には専用の充電器を使いますが、完全に放電してから充電しないとメモリ効果⁷を起こしてしまうという欠点もあります。しかし、最近の充電器にはリフレッシュ機能⁸が付いているものもあるので、それと併せて買えば問題ありません。

⁷ 電池を使い切らないうちに充電すると放電電圧が低下する現象

⁸ 一度電池を完全に放電させてから充電を開始する機能

3. 迷路解析の手法と考え方

マイクロマウスは、スタートからはじまって、ゴールまで迷路を解析し、最短距離を求めて走行する人工知能を駆使した競技でもあります。もちろん、全ての壁についてマッピングしなくてもゴールにたどり着くことはできます。しかし、マイクロマウス競技ではたいていゴールまでの道のりは複数用意されていて、制限時間がある中で最短経路を探し出して好タイムをマークするには、やはり広範囲のマッピングを効率的に行うことが重要です。このために様々な手法が考え出されており、ここでは代表的なものを紹介します。

3.1 マッピングについて

競技開始時点では、マイクロマウスにとって迷路は未知です。このまま闇雲にゴールを目指して走ってもよいのですが、それでは効率的にゴールへの最短経路を見つけることができません。したがって、マイクロマウスにはまず迷路全体を探索して、迷路の地図を作ることが要求されます。ルールによればマイクロマウスは競技中に複数回走行でき、その中で最短の記録が採用されますから、試合戦略としてまっすぐにゴールを目指すのはロボットの頭の中に地図を作ってからでも遅くありません。

この「ロボットが迷路を探索して頭の中に地図を作ること」をマッピングと呼びます。マッピングの手法は様々ですが、基本的には次のような考え方に基づいています。次の図を参照してください。

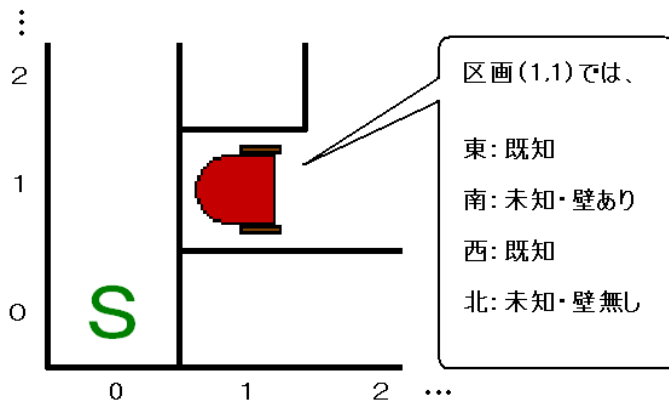


図 8 マッピング

ルールでは、スタート位置は迷路の左端がスタートであり、2018年にゴール位置が変更になりますが、スタートとゴールだけは既知です。しかし、それ以外の区画はどのようなになっているのかマイクロマウスにはわかりません。スタートを出て1区画目から、マイクロマウスは、その区画の周りの壁の有無を調べた事があるかどうかを判定し、まだ調べていなかった場合は、センサーを使って壁の有無を判定します。迷路を1区画進むごとにこれを繰り返していくことにより、マッピングが行われます。まるでダンジョンゲームでマップを作っていくかのような作業となります。全区画を踏破した後はマップが完成しますので、このマップに基づいて最短距離を計算します。次からは効率が良いとされる迷路の探査方法について説明します。

左手法⁹

最も簡単な迷路探索法です。図を参照してください。

左手法とは、例えば遊園地などにある巨大迷路に挑戦する時に、左手をずっと壁に付けたままたどっていけばいつか必ずゴールにたどり着ける可能性が高い、という考え方に基づいた探索法です。マイクロマウスで言えば、

左側に壁が無ければ左へ進む

左側に壁があり、正面に壁が無ければ直進する

左側と正面に壁があり、右側に壁が無ければ右へ進む

左側、正面、右側に壁がある場合は来た道を引き返す

という動作の繰り返しです。（図は簡単に左下がスタート、右上がゴールとして描いています）

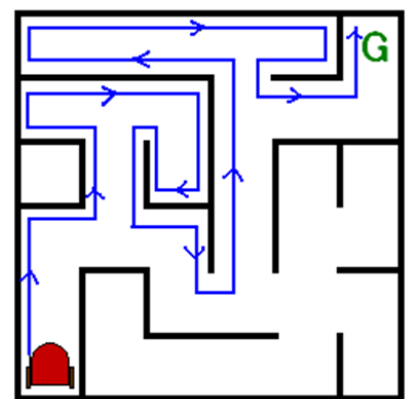


図 9 .2 左手法

⁹左右を完全に逆にすれば右手法、拡張右手法になります

3.3 拡張左手法

左手法には大きな弱点があります。図 11 を参照してください。この迷路では、真ん中がゴールになっています。このような迷路の場合、左手法でコースをたどっても絶対にゴールにはたどり着けないということが分かります。

この左手法の弱点を克服しようと考え出されたのが、拡張左手法です。拡張左手法では、1度通ったコースを再び通らないように配慮したものです。詳しく知るために、図 10 を参照してください。この図では、ロボットが左手法に従って左に曲がると、同じコースをぐるぐると回り続けることになってしまいます。よって、これを防ぐために、仮想壁という概念を導入します。

仮想壁とは、実際には壁がない部分にデータの上でのみ存在すると仮定した壁です。もちろん仮想ですから、最短経路を求めるような場合には無視しなければいけません。しかし探索中において、仮想壁は大きな役割を果たします。

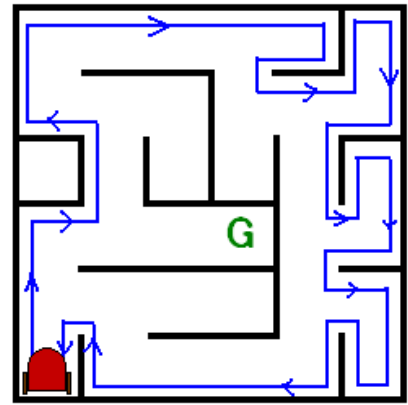


図 10 左手法ではゴールできない迷路もあります

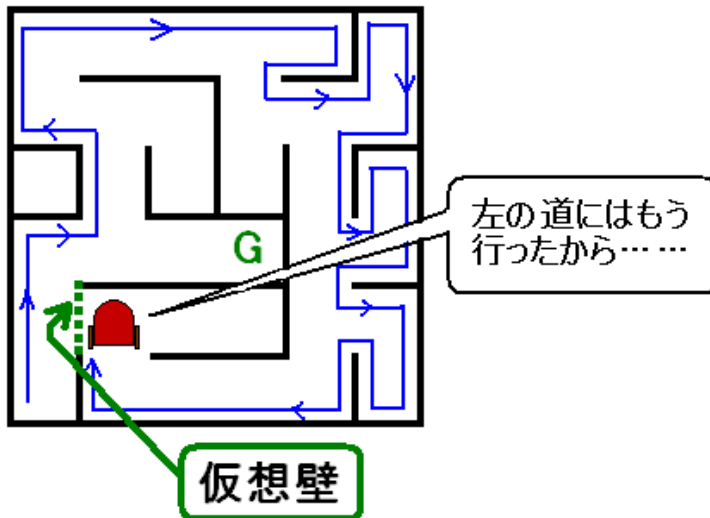


図 11 ループ防止の仮想壁を作ります

図 10 に戻ります。この時、左に旋回する前に左の区画の壁が探索済みであるかどうかを判定します。そしてすでに探索済みなら、この探索済みルートへ入る部分に仮想壁を作るのです。探索中はこの仮想壁も壁として認識しますから、もうここから探索済みルートへと迷い込むことはありません。また、図 11 のように袋小路に行き当たったら、一度分岐点まで戻って袋小路の入り口に仮想壁で蓋をしていきます。

このように、進む先が探索済みであればそこに仮想壁をつくり、その上で左手法に従う、という方法が、拡張左手法です。

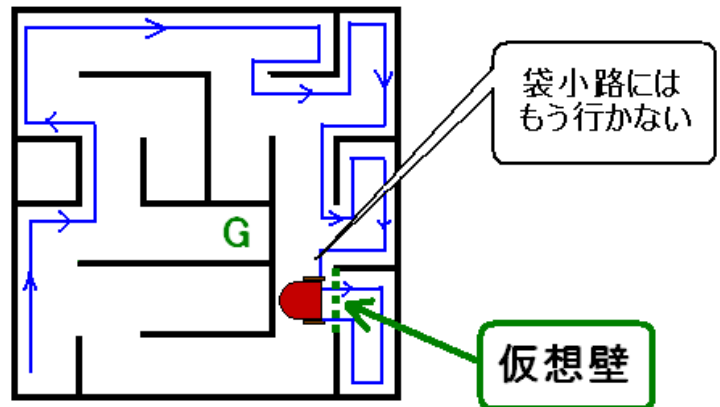


図 12 袋小路にも仮想壁で蓋をします

3.4 求心法

いくらマップをまんべんなく探索しても、ゴールにたどり着くのに時間がかかってしまっても効率がよくありません。そのことを考慮された手法が考え出されました。その1つが求心法です。

左手法では、分岐点に来たときには左→前→右の順に優先して路を選んでいました。しかし求心法では、必ずゴールに近いほうの路を選ぶように工夫がなされています。その工夫が、図14の重みマップです。

このマップ中の各数字は、壁を無視した場合にゴールからその区画までにかかる最短歩数（進む区画数）です。未知の経路同士の分岐点にさしかかったときには、この数字を比べて小さいほうを選んで進みます。これによって、ロボットは、迷路を探索しながらも少しずつゴールに近づいていくことができます。人工知能の基本的な探索法として、よく知られた手法の一つで、別名として勾配法とも呼ばれます。

| | | | | | |
|---|---|---|---|---|---|
| 4 | 3 | 2 | 2 | 3 | 4 |
| 3 | 2 | 1 | 1 | 2 | 3 |
| 2 | 1 | 0 | 0 | 1 | 2 |
| 2 | 1 | 0 | 0 | 1 | 2 |
| 3 | 2 | 1 | 1 | 2 | 3 |
| 4 | 3 | 2 | 2 | 3 | 4 |

※ 中心の4区画をゴールとしています

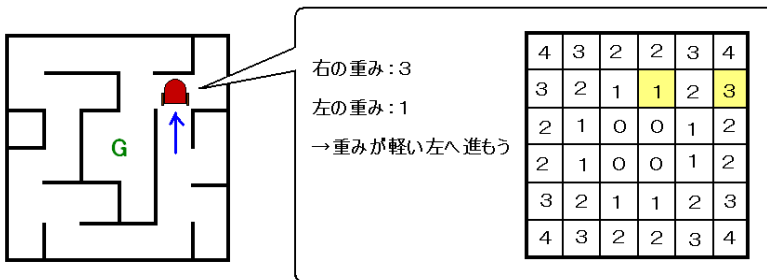
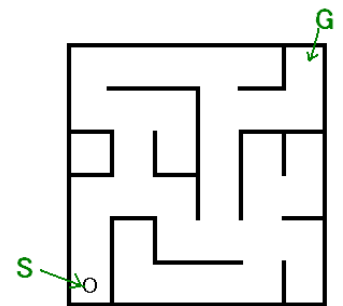


図14 重みマップを使って進む方向を決めます

3.5 トレモー法

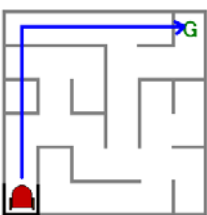
トレモー法は拡張左手法と似た考え方をしますが、「その地点を通った回数」を使うことで効率的に探索しようというものです。トレモー法は以下の5つの条件に従ってマイクロマウスを走行させ、マップを作っていきます。（5つの条件は優先順位順に書いてあります。）

1. 左右とも壁がある地点は直進する
2. 未通過の角では好きなように進む
3. 未通過の地点から通過済みの角についた場合、引き返す
4. 通過済みの地点から通過済みの角についた場合、通った回数の少ない方へ進む
5. 行き止まりの地点は引き返す



3.6 足立法

今までに紹介した探索法では、ロボットは迷路を探索してマッピングを終えてから後述の最短経路を算出します。しかし足立法では、ロボットは1区画進むごとに最短経路を算出します。

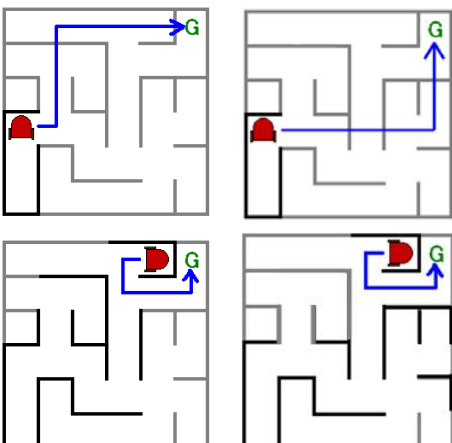


スタート時、ロボットは壁の有無を知りません。しかし、未知の部分には壁が無いものと仮定してとりあえずの最短経路を求め、その経路にそって走行します。やがて壁に阻まれて仮定した最短経路を進めなくなると、既知の壁を考慮しての最短経路を求めなおします。これを繰り返しながらスタートとゴールの往復を繰り返し、求めた最短経路中に未知の区画が無かったら探索終了です。具体的な例を図16～図18に示します。

※黒い壁：探索済み 灰色の壁：未探索

図15

探索していない部分には壁がないと仮定して、最短経路を求めます



求めた最短経路の通りに走行しようとしたら壁に阻まれてしまいました。こうなったらもう一度最短経路を求めなおします。

図16

これでゴールです。

図 17

3.7 最短経路の求め方

これまで紹介してきたアルゴリズムのいずれにおいても、最短経路を求めるために、歩数マップという考えを導入します。

歩数マップとは、迷路状の各区画までロボットがたどり着くまでに要する歩数を表したマップです(図 19)。この歩数マップの作り方の原理を説明します。図 13 を参照してください。

まず、スタート地点の歩数は0です。そして、そこから壁の無い方向(=進める方向)に1歩進んだ部分の歩数は1です。

このように、隣り合い、かつ進める区画にむかって1ずつ数字を増やしていきます。隣の区画にすでに数字が書き込まれている場合は、小さいほうを優先します。求めたいのは最短歩数だからです。このようにしてマップを数字で埋めていって、ゴールの区画に数字が書かれた時点で歩数マップの作成を完了します。

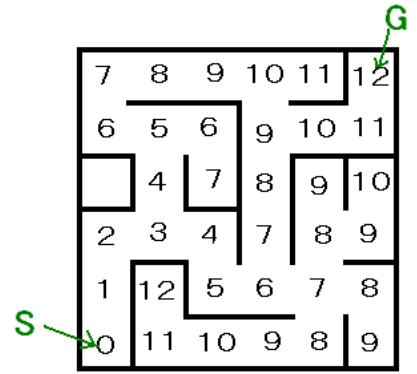


図 18 歩数マップ (Sはスタート、Gはゴール)

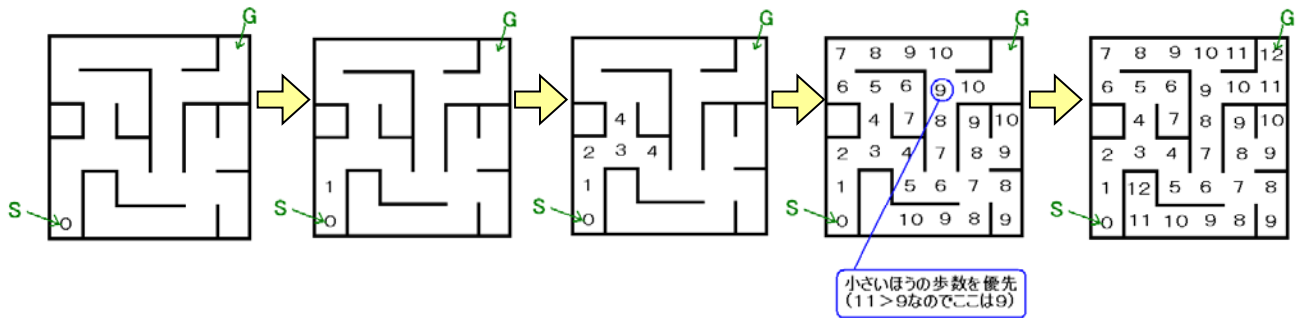


図 19 歩数マップの作り方

あとは、ゴールから数字が1ずつ減っていくようにたどっていくと、そこが最短経路になっています。

また、マイクロマウスのような複雑な迷路になると、最短経路が複数あることもよくあります。そのような場合は、直進経路が多い方のルートに優先させるのが良いでしょう※。歩数マップを作る段階で、旋回する場合は1歩余計に歩数がかかるとして計算するという方法もあります。

※機体の特性を考慮して優先度を決めます。直進性がよい機体は直進を、直進よりスラロームが得意な場合は曲がるルートをとるのも問題はありません。

いずれにしても、独自のアルゴリズムを考えるのがマイクロマウス競技の醍醐味です。ここで紹介したものを参考にいろいろなアルゴリズムを考案してどんどん知的に制御してってください。

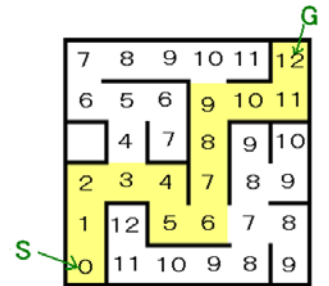


図 20 最短経路を求める

4. 備考

●参考文献

マイクロマウス 20 周年座談会

http://www.ntf.or.jp/archives/zadan/taidan_01.pdf

http://www.ntf.or.jp/archives/zadan/taidan_02.pdf

http://www.ntf.or.jp/archives/zadan/taidan_03.pdf

●著作権について

本取扱説明書で紹介、または記載されている会社名、製品名は、各社の登録商標または商標です。

本取扱説明書に掲載している文書、写真、イラストなどの著作物は、日本の著作権法及び国際条約により、著作権の保護を受けています。インターネット等の公共ネットワーク、構内ネットワーク等へのアップロードなどはおよび株式会社アールティの許可無く行うことはできません

5. 改版履歴

| 発行日 | ページ | 改訂内容 |
|--------|-----|------|
| 2018/7 | - | 新規発行 |

製造元

株式会社アールティ

〒101-0021 東京都千代田区外神田 3-2-13 山口ビル 3F TEL 03-6666-2566

URL <https://www.rt-net.jp/>

製品に関するお問い合わせ

本製品に関するお問い合わせは、下記までお願いします。

お問い合わせは電子メールにて受け付けております。

E-mail: support@rt-net.jp