



写真1 6輪マイクロマウス「紫苑」

マイクロマウス「紫苑」大解剖！

ソフトウェア編

なかしま ちかたか
中島 史敬

はじめに

前回のハードウェア編に引き続き、今回は「紫苑」(写真1)のソフトウェアについて解説する。しかし、そのすべてをこの数ページで説明することはできないため、全体の大まかな構成と特徴的な部分、そしてハードウェアの制御について解説していくこととする。

アセンブラ

紫苑のソフトウェアは、前回も書いたように、すべてアセンブラで記述しており、その開発ツールには、ポーランド社の「Turbo Assembler」を使用している。

アセンブラといっても、C言語などと同じで、テキストエディタで作成したソースファイルをコンパイルして、実行ファイルを作成するのであるが、Turbo Assemblerが出力する実行ファイルは、基本的にMS-DOS上で動作するEXE形式であるため、EXE2BINコマンドにてバイナリファイルにコンパイルした後、マウスに搭載しているオリジナルのモニタプログラム(これもアセンブラで記述)を使用して、開発用PCからマウス上のメモリに、シリアル通信で転送している。

ソフトウェアの構成

このような方法にて作成されている紫苑のソフトウェアは、図1に示すように、2つの大きなブロックに分けることができる。

つまり、未知の迷路を探索しながら走行す

る迷路探索部や、探索済みの迷路を用いて目的地までの最短経路を算出して走行する最短走行部、さらには各種走行コマンドのテストやセンサの調整を行うテストプログラムなど、マウスにどのような動作をさせるのかを決めるメインルーチンと、それらのメインルーチンから出力される走行コマンドの解析や、実際のハードウェアの制御、マウスの各種パラメータを「ブラックボックス」という記録領域に保存しておく処理を行うBIOSの2つから構成されている。

このように、迷路探索や最短経路の計算といったハードウェアに依存しない部分と、モーターやセンサの制御といったマウスごとに変更しなくてはならない部分とを分けることによって、新しいマウスを製作してもプログラムをすべて書き換えるのではなく、必要な箇所のみを変更すればいいので、ソフト開発の手間を省くことができるのである。

なお、ブラックボックスとは、マイクロマウス界における方言であって、一般的には、フライトレコーダーやドライブレコーダーと言われるものである。つまり、走行中のセンサ値や速度値などを一定時間ごとに記録しておき、走行後にそのデータをPCに転送して確認することによって、目で見るだけではわからない細かい挙動のチェックを可能とするものである。

迷路探索アルゴリズム

迷路探索方法としては、求心法、トレモーフ法、足立法などがあるが、これらは、過去の

ロボコンマガジンやさまざまなWebサイトに解説されているので、ここで詳しくは解説しない。

紫苑では、基本的には全迷路探索をしており、探索時の進行方向を決める手段として、トレモーフと呼ばれる求心法をベースとした探索法に、「枝刈り」という改良を加えたものを用いている。この枝刈りとは、迷路で袋小路となっている部分の外周壁がすべて検出済みである場合に、その入り口に仮想壁を作ることによって袋小路内に入り込まないようにして、無駄な迷路探索をしないようにする方法である(図2)。

しかし、近年のエキスパートクラスの迷路では、このように袋小路となる部分が少ないため、その効果はほとんどないのが実状でもある。たとえば、2005年のエキスパートクラス予選迷路を探索し終えた紫苑からダウンロードした迷路図(図3)を見ると、枝刈りで生成した青い壁と検出済みの赤い壁にて囲まれた袋小路はわずか数ブロックしかなく、より賢い探索アルゴリズムを採用したほうがよいかもしれない。

なお、この例示した迷路図では、壁検出のミスを起こしており、そのため、本当の最短経路を選択することができていない。

最短経路の算出

最短経路の求め方は、まず、出発点からの距離(コマ数)の等高線を描き、その後、目標点を始点として、そこから距離値の低いコ

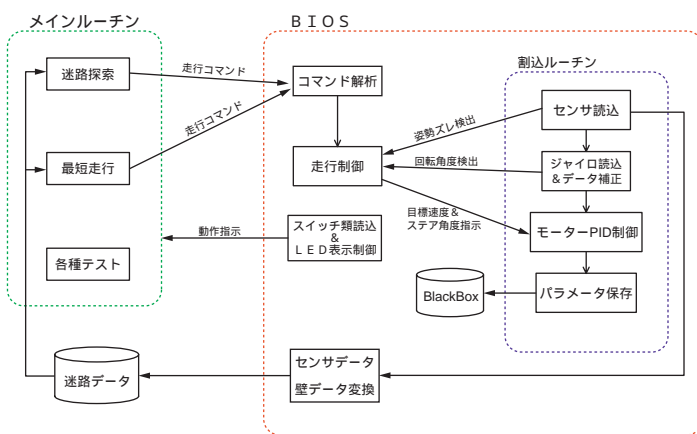


図1 ソフトウェアの構成

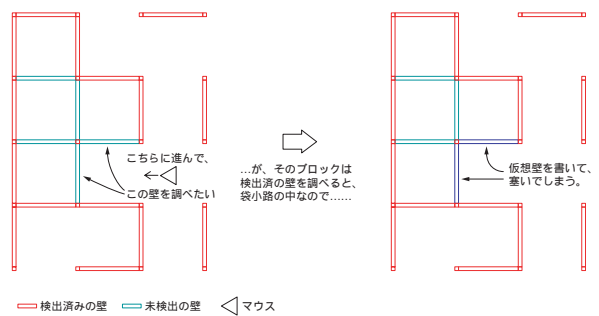


図2 枝刈りの例