

# 中学校・技術科における PIC 搭載センサーカー教材の一検討

西 正明      生活科学教育講座  
北山 遼太    川上村立川上第一小学校

キーワード：PIC, センサーカー, シミュレーション, 技術科教材

## 1. まえがき

中学校・技術科の情報領域において、今までのところ、コンピュータリテラシー、情報リテラシーに関する事項が多く扱われてきている。しかし、学習指導要領の改定に伴って、今後はコンピュータの技術的な要素といえるプログラミング・制御が学習すべき内容として扱われるようになる。したがって、これまで中学校技術科教育においてプログラミングは選択項目であったために学ぶことは少なかったが、今後は多く取り入れられていくと考えられる。プログラミングはプログラミング言語の文法を学ばなければならない、文法に沿わない部分があれば思うように動かすことができないために、難しいものと思われがちである。そのため、生徒がまず興味を持ち、高いモチベーションを持てるようにするため、題材としてロボット、具体的にはセンサーカーを取り上げて、難し過ぎることなく、動作を画面上で確認しながら論理的なプログラミングを学べるような教材の開発を行う。

プログラミング言語には、日本語でプログラミングができる教育用プログラミング言語である、ドリトル<sup>1)</sup>を用いることにした。ドリトルでは細かい文法にとらわれず、比較的自由的な記述が許容され、表示画面内でタートルと呼ばれる物体（通常は亀）を動かして描画することなどができる。実際のものづくりでは、設計の段階でさまざまなシミュレーションを行って、不具合な部分がないか確認することが行われるが、ドリトルではタートルの動きをこのシミュレーション機能の一部見立てることができる。しかし、ロボット制御では、各種のセンサーの働きをプログラムの的に実現していないと十分なシミュレーションができない。技術科の限られた時間数のなかで、ロボットづくりを効果的に授業で進めるためには、シミュレーションは必要な機能であると考えられる。そこで、ドリトルの画面上でタートルの動作を確認しながらプログラムを完成させて、次に簡単な動作記述を追加するだけでセンサーカー用のプログラムに変換できるようにすることで、ドリトルのシミュレーション機能を充実させた。さらに、この教材を用いた自律型ロボットづくりの授業案を試作した。

## 2. センサーカー教材

試作したセンサーカー教材について、まずセンサーカー本体とそれを動かすプログラミング言語について述べる。次にプログラミング言語ドリトルのシミュレーション機能についての問題点とその対策を行ってシミュレーション機能を充実させることを述べ、その場合のプログラミングとセンサーカーにプログラムを転送して動作させる手順を述べる。

### 2.1 教材

現行の平成 10 年中学校学習指導要領―技術・家庭編―では、「B 情報とコンピュータ」で選択項目として、「(6)プログラムと計測・制御について、次の事項を指導する。ア プログラムの機能を知り、簡単なプログラムの作成ができること。イ コンピュータを用いて、簡単な計測・制御ができること。」とある。しかし、現在この項目が学習されることは稀である。「B 情報とコンピュータ」ではマルチメ

ディアの活用など社会的な側面を学ぶことに主眼が置かれており、プログラムと計測・制御はほとんど学習されていない。茅野市立北部中学校の調査<sup>2)</sup>によると、諏訪郡内で平成18年度にプログラムと計測・制御の領域を学習した中学校はなかった。それとは対照的に、この中学校では、「プログラミングはよく分からないがプログラミングをしてみたい」と思っている生徒が半数を超えていた。このことから、現在の情報に関する学習は十分ではないといえる。私たちの身の回りにある家電製品や自動車などはコンピュータで制御されており、コンピュータを利用する機会は日常的なこととなっている。中学生の生活にこのようなコンピュータや機械の制御が深く関わっている現状があるのにも拘わらず、中学生がプログラミングと計測・制御に触れる機会は少なく、中学校での学習は十分ではないと考えられる。プログラミングを学び実際の機械がどのように制御されているのかを知ることは、現代のものづくりを学ぶ上で大変重要なことである。中学校技術・家庭[技術分野]の次期学習内容では、A材料と加工に関する技術、Bエネルギー変換に関する技術、C生物育成に関する技術、D情報に関する技術の4つとされる。D情報に関する内容としては、(1)情報通信ネットワークと情報モラル、(2)デジタル作品の設計・制作、(3)プログラムによる計測・制御の3つとされる。したがって、今まであまり行われていなかった「プログラムによる計測・制御」について、適切な教材の開発が必要である。

中学生が興味をもって取り組める教材にロボットづくりがある。自分が作ったプログラム通りにロボットが動作することは達成感や成就感につながり、挑戦する気持ちを起こさせる。プログラムされた通りにロボットが動くという点からプログラムの意味を実感できる上、実際のものづくりでプログラムが使われていることを体感することができる。そこで、ロボットの教材として、複雑過ぎることなく基本的な制御内容が含まれるものとして、タッチセンサーと制御用マイクロコンピュータPIC(Peripheral Interface Controller)を用いて動きを制御するモーターカーを取り扱うことにした。PICのプログラミング言語としては、文法上の規則が複雑で文法チェックの厳しいプログラミング言語や軌道の印されたタイルを並べるだけで制御プログラムになるようなおおよそプログラミング作業とかけ離れているものは避け、簡単な日本語でプログラムを作成できる教育用プログラミング言語「ドリトル」を使用することにした。ドリトルはプログラムに記述した命令の順番に実行していき、ループやブロックなどのプログラムの各要素を学ぶことができる。ドリトルはオブジェクト指向言語であるため、将来の実用性、発展性が高いと考えられる。

#### (1) センサーカー

現在、さまざまなロボット教材が販売されている。その中には高価で技術科の授業では取り入れられないものがある。例えば、株式会社アルフルのロボット教材「WARP 5500」は文部科学省学習指導要領に準拠しているが、グループで製作するとしてもなお高価であり、中学校で導入することは難しいと考える。これに比べて、図1に示す久富電機産業株式会社の「2軸キャタピラロボット」は、それほど値段は高くないがロボット制御用ソフトウェアが指定され、軌道が印されたタイルを貼り付けていくとセンサーカーの動作プログラムが出来上がる仕組みのもので、プログラミング学習に適しているとは言い難い。出来上がったプログラムをロボットに搭載されたPICに転送するためには、赤外線を用いたコンピュータ接続インタフェース装置を別途購入しなければならず煩わしい。

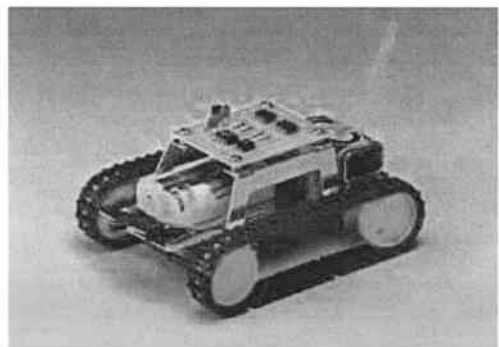


図1 2軸キャタピラロボット

本研究では、ロボット工房スタジオミュウ製のミュウロボ制御基板 MYU-501<sup>3)</sup> を使って、センサーカーを製作することにした。ミュウロボ制御基板 MY-501 を図 2 に示す。これは制御基板だけを購入することが可能で、センサーカー本体は駆動用部品を追加購入すれば比較的安価に製作することができる。この制御基板では、モータドライバーに LB1639, PIC に PIC16F630 を使用している。PIC16F630 の入出力端子を図 3 に示す。ミュウロボ制御基板 MY-501 にモータ、ギヤボックス、タッチセンサーなどを取り付けて製作したセンサーカーを図 4 に示す。

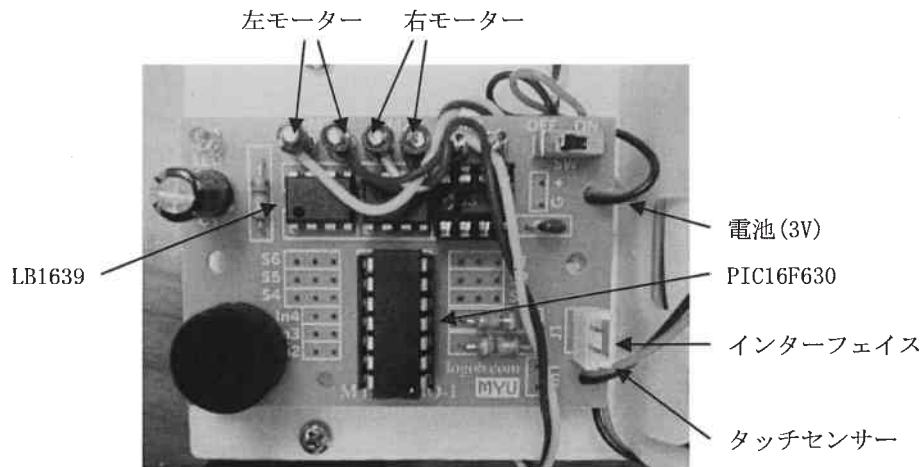
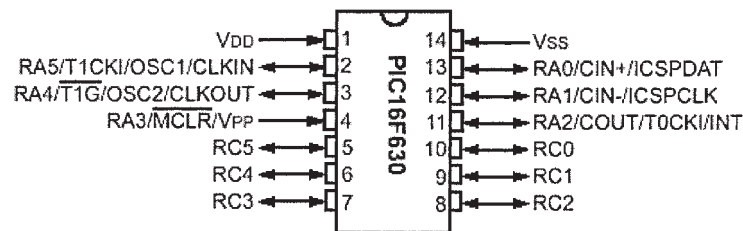


図 2 ミュウロボ制御基板 MYU-501



- 1:VDD(+)
- 2:RA5(出力) LED と圧電ブザー用
- 3:RA4(入力) In1(入力1・リミットスイッチ)
- 4:RA3(入力) シリアル信号入力用
- 5:RC5(出力) M2R(モーター2R), S3(サーボ3)
- 6:RC4(出力) M3L(モーター3L), S2(サーボ2)
- 7:RC3(出力) M3R(モーター3R), S1(サーボ1)
- 8:RC2(出力) M1L(モーター1L), S6(サーボ6)
- 9:RC1(出力) M1R(モーター1R), S5(サーボ5)
- 10:RC0(出力) M2L(モーター2L), S4(サーボ4)
- 11:RA2(入力) In4(入力4)
- 12:RA1(入力) In3(入力3)
- 13:RA0(入力) In2(入力2)
- 14:VSS(-)

図 3 PIC16F630 の入出力端子

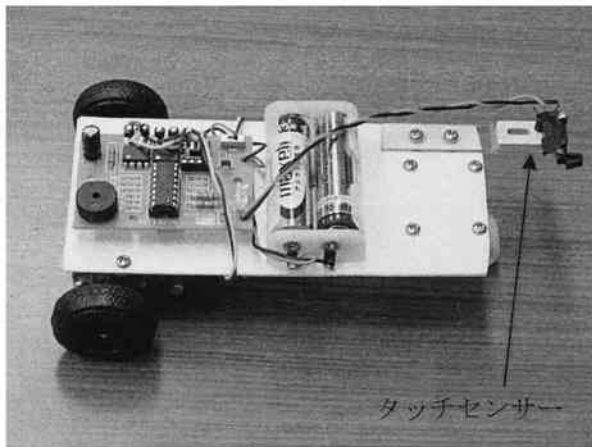


図4 製作したセンサーカー

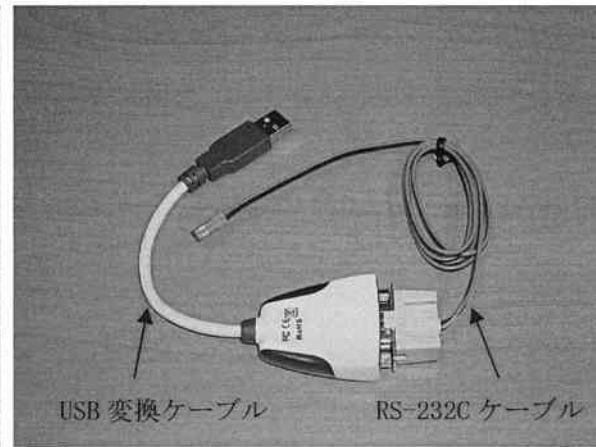


図5 接続ケーブル

## (2) ドリトル

図4に示したセンサーカーを動かすためには、図2に示したミュウロボ制御基板に搭載しているPICにプログラムを転送することが必要である。パソコン側がRS-232C端子であれば付属のRS-232Cケーブルで制御基板と接続すればプログラムを転送できる。パソコン側がUSB端子の場合には、図5に示すように付属のRS-232CケーブルにUSB変換ケーブルをさらに接続すれば、プログラムを転送できるようになる。センサーカーを動かすプログラムをPICに転送すると、制御基板のPICにあらかじめ書き込まれているファームウェアというプログラムによってマシン語に変換保存されて、センサーカーが動作可能になる。ミュウロボ制御基板の対応しているプログラミングソフトには、ドリトル、myurobo, MYU BASICの3種類がある。myuroboはセンサーカーの動きをボタンで指定するように単純化されたもので、中学校・技術科のプログラミング学習に適しているとは言い難い。MYU BASICはアルファベットによる命令の記述が中心になるため、理解が容易とは言えない。そこで、日本語で記述できるドリトルを用いることにする。

ドリトルはオブジェクト指向言語である。すなわち、対象についての情報と操作を別々に記述する手続き的言語ではなく、対象についての情報とそれに対する操作をまとめて記述するようにした進歩的なプログラミング言語である。オブジェクトすなわち具体的な対象を表現するのに、一般的にはクラス方式を用いるが、ドリトルではプロトタイプ方式をとっている。クラス方式ではクラス、インスタンス、継承など、理解しなければならない概念が多いために、どうしても理解し難くなってしまふ。これに対して、プロトタイプ方式ではクラス概念がなく、継承はオブジェクト間の委譲関係で定義され、委譲関係はオブジェクト生成時に自動的に定まるようにできるため、多くの概念を学ばなくてもよく、簡潔にプログラムを記述することができる。このようにドリトルは簡潔で、高度な概念を学習する必要がない。それにもかかわらず、ドリトルでは、手続き的なプログラミングの基礎概念として、逐次実行、代入による値の変化、サブルーチン、制御構造による実行制御ということや、記述された通りにプログラムが動作するというプログラミングの本質を学ぶことができるようになっている。なお、従来からよく使用されているLogoは、再帰やリストといった高度な概念を学ぶ必要があり、対話的なタートルグラフィックス以降の学習が難しい。

次に、タートルに障害物を回避して移動させる課題についてのドリトルのプログラム例を図6に示す。図6のプログラムのドリトル実行画面を図7に示す。プログラムの上から10行目までは障害物の描画部分であり、12行目以降は亀の形をしたタートルの動きを記述している部分である。カメ太と名

付けられたタートルが (250, -40) の位置で初期状態が右向きだったのを左向きに変えて 20 歩進める。進んでいる途中で障害物と衝突すると、50 歩後ずさりして右回りに 90 度向きを変える。次に 20 歩進んで左回りに 90 度向きを変える。最後に 20 歩進んで停止する、といった動きをタイマーと連携して記述している。その結果、図 7 に示されるようなコースを辿って、タートルが時々刻々と移動していく。このようにドリトルでは、日本語で記述するので比較的わかりやすく、タートルを動かすプログラムを簡単に作ることができる。

```

タートル:カベ=「|長さ 厚み|
!ペンあり。
「!(長さ)歩く 90 左回り (厚み) 歩く 90 左回り。」!2 繰り返す。
!図形にする。
!ペンなし (長さ) 歩く。」。
大工=タートル!作る ペンなし 消える 250 200 位置。
「大工!90 右回り 400 2 カベ。 大工!90 右回り 600 2 カベ。」!2 繰り返す。
大工=タートル!作る ペンなし 消える 50 0 位置。
「大工!90 右回り 100 2 カベ。
大工!90 右回り 150 2 カベ。」!2 繰り返す。
//////////////////// 上部:障害物 下部:タートル動作 //////////////////////
カメ太=タートル!作る ペンなし。カメ太!250 -40 位置。カメ太!180度 向き。
時計=タイマー!作る 0.3秒 間隔 10秒 時間。時計!「カメ太!20 歩く。」実行。
カメ太:衝突=「時計!中断。
カメ太!50歩 戻る。
カメ太!90度 右回り。
」。実行。

時計!待つ。
時計=タイマー!作る 0.3秒 間隔 1秒 時間。時計!「カメ太!20 歩く。」実行。
時計!待つ。
時計=タイマー!作る 1秒 間隔 1秒 時間。時計!「カメ太!90度 左回り。」実行。
.....中略.....
時計=タイマー!作る 0.3秒 間隔 2秒 時間。時計!「カメ太!20 歩く。」実行。

```

図 6 障害物回避課題のタートルプログラム例

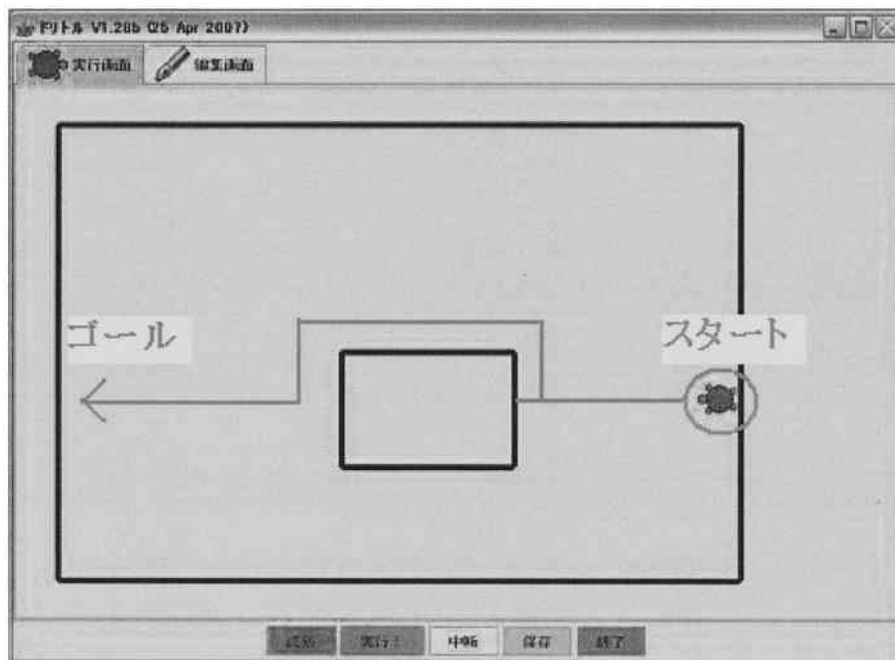


図 7 障害物回避課題のドリトル画面例

## 2.2 問題点と対策

図6に示したドリトルのプログラムはタートルを動かすことはできるが、これをそのまま図4のセンサーカーにプログラムを転送しても思うように動かすことはできない。向きの変更について、タートルの場合は回転角度で指定しているが、センサーカーではその指定はできず左右のモータを回転させる時間で指定しなければならない。また、衝突という事象についても、タートルの場合は“衝突”と記述すれば座標値の比較で判別するが、センサーカーの場合にはリミットスイッチ（タッチセンサー）に関する記述にしなければならないためである。図7と同じ動きを図4のセンサーカーにさせるためには、ドリトル上のプログラムを図8に示す記述内容にしてセンサーカーのPICに転送しなければならない。

```

ロボ太=MYU!"com5"作る。
ロボ太:転送命令="!"はじめロボット
100 前進 "!"前進
      リミットスイッチ
      10 後退 10 左前
      ]実行
10 前進 10 左後
40 前進 10 右前
10 前進 10 左前
10 前進 おわりロボット。
ロボ太!転送命令。
    
```

図8 障害物回避のセンサーカープログラム

ドリトル上でのプログラムを図8に示す記述内容にしてセンサーカーのPICに転送しなければならない。ドリトルには、図2の制御基板にプログラムを転送する命令があり、図8では“転送命令”で囲まれた記述部分がPICに転送される。図6に示したプログラムはドリトルの表示画面上で動作を確認しながらプログラミングすることができる。しかし、図8のプログラムはドリトルの表示画面上で動作を確認しながらプログラミングすることはできない。そのため、センサーカーにプログラムを送ってみるまでセンサーカーがどのような動作をするのかわからないという問題がある。

そこで、画面上でタートルを動かすプログラムの中で、センサーカーを意識した細かい動作をコメント部分に記述して、この部分の記述に基づいてセンサーカー用のプログラムに変換するようにした。そのための変換プログラムを製作することにした。タートルとセンサーカーの命令対応を表1に示すようにした。表1に示すように、タートルでの「歩く」「中断」「戻る」の命令は、センサーカーでは「前進」「停止」「後退」の命令にそれぞれ対応する。タートルを回転させるには回転命令と回転角度を指定すればよいが、センサーカーでは左右の駆動輪の回転方向と回転時間を指定しなければならない。例えばタートルの「右回り」の命令は、次の3つの場合が考えられ、それぞれの回転方法を//の後にコメントとして記述するようにして指定した。

- (1)左後輪を前進 → //前
- (2)右後輪を後進 → //後
- (3)左後輪を前進, 右後輪を後進 → //中

これらの3つの場合に対するセンサーカー用の命令は、表1に示すように、(1)の場合は「//前」の記述を識別して「左前」、同様にして(2)の場合は「//後」の記述を識別して「右後」、(3)の場合は「//中」の記述を識別して「右回り」とする。タートルの「左回り」の命令に対しても同様に、表1に示すように3つの場合を扱えるようにした。

表1 タートルとセンサーカーの命令対応

右車輪回転	正	正	×	×	逆	逆	×	正	逆
左車輪回転	正	×	正	×	逆	×	逆	逆	正
タートル	歩く	左回り	右回り	中断	戻る	右回り	左回り	左回り	右回り
識別記述		//前	//前			//後	//後	//中	//中
センサーカー	前進	右前	左前	停止	後退	右後	左後	左回り	右回り

回転時間は、使用するセンサーカーの実態に合わせなければならないが、図 4 に示したセンサーカーでは回転角度 10 度につき 1 秒として、プログラム変換時に回転時間を付加するようにした。例えば、次のタートル用プログラム「10 度 右回り」の場合には、「//前」を記述することによってセンサーカー用プログラムとして「1 左前」、すなわち 1 秒間左車輪を正回転させる命令に変換されるようになる。

時計=タイマー!作る 1 秒 間隔 1 秒 時間。

時計!「カメ太!10 度 右回り。」実行。 //前 ⇒ 1 左前

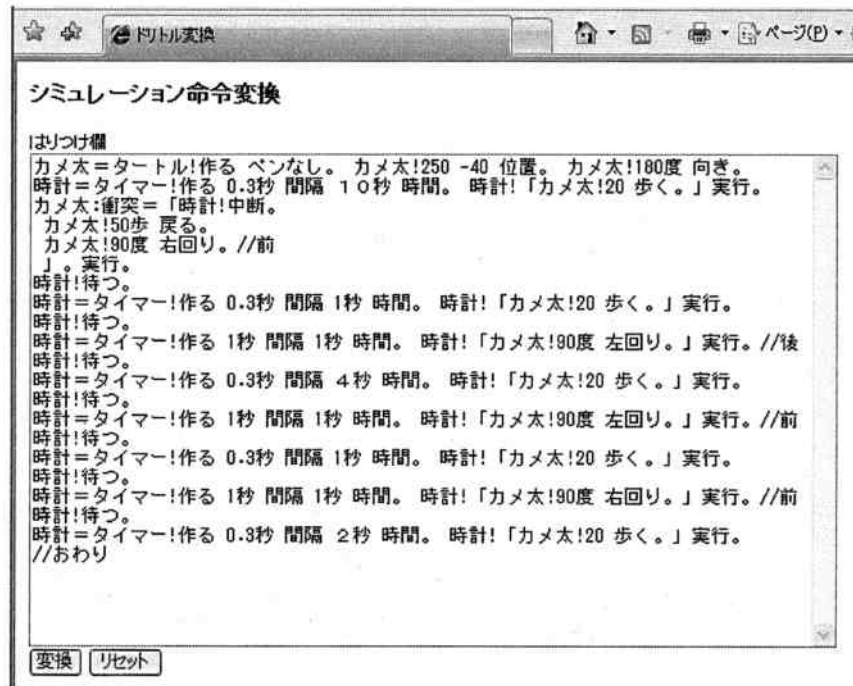


図 9 プログラム変換の入力画面

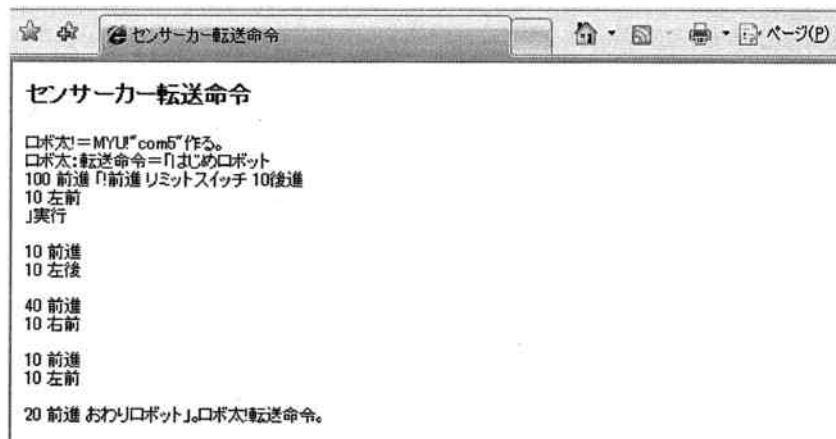


図 10 プログラム変換の出力画面

以上によりプログラムを変更する度にセンサーカーに転送しなくても、画面上でタートルの動きを確認しながらプログラムを完成させてセンサーカー用の動作記述を追加すれば、センサーカーのプログラムに変換出来るようになった。これによって、全体の動作でどこが間違っているのか確認しやすくなり、プログラミング学習をより進めやすいものにとできると考える。動作のシミュレーションを体感することにより、シミュレーションで基本部分を作り、そこから微調整していくという実際のものづくりを学ぶこともできる。プログラム変換は Perl 言語で製作したので、Web 上で操作することができる。図 9 にプログラム変換前の入力画面、図 10 にプログラム変換後の出力画面をそれぞれ示す。そ

れぞれ、コピー&ペーストで操作できるようにして使いやすくしている。

この教材でのプログラミングと動作確認の手順をまとめると、以下のようになる。

- (1) タートル用のプログラム作成
- (2) センサーカー用の動作記述の追加
- (3) プログラム変換
- (4) プログラム転送
- (5) センサーカーの動作確認

### 3. センサーカー教材を用いた授業案

第2節では、プログラミングの部分を中心に述べたが、実際にはセンサーカーを組み立てる段階で、電気、機械、材料加工など複数の学習領域が含まれてくる。中学3年生向けに、この題材を用いた全17時間扱いの授業案を試作してみた。表2に学習内容別のつける力と時間数の内訳を示す。

表2 学習内容別のつける力と時間数の内訳

学習内容	つける力	授業の概要	時間
導入・イメージづくり		教師がつくったシミュレーション画面や、自律型ロボットの動きを見て授業の目的を把握するとともに、挑戦する意欲をもつ。	1
電気回路	電気回路図の読み取り，電子部品の組立て	電気回路図の読み取り	1
		はんだづけの練習	1
		基板上の電子部品の配置を決め，はんだづけする際の注意点について学習する。	1
		基板に電子部品をはんだづけする。	3
材料の加工・組み立て	材料の加工技術，部品と全体の組み立て	動力伝達とギヤ比の学習 ギヤボックスの組み立て	2
		プラボードの加工 センサーカー組み立て	1
		シミュレーションの学習	1
プログラミング・制御	プログラムの機能の理解，プログラムの作成，機械の制御	画面上で動くドリトルについての学習	1
		ドリトルを用いたプログラミング シミュレーションを使ってプログラムの修正	3
		誤差の修正	1
		自分の作品の反省や、他の生徒と認め合うことでの振り返り。 学んだことを発表しまとめをする。	1



全体的には以下の3つの部分になる。

(1)最初の7時間

全体の説明と電気回路を学び、回路基板づくりや、はんだづけを行う。回路図を見て部品を正確に取り付けて配線組み立てを行う。ここで、正確なはんだづけや回路図を読む力、電気回路の知識を身につける。

(2)次の3時間

加工・組み立てについて学ぶ。ギヤボックスを組み立てながらエネルギーの伝達を理解し、ボードにタッチセンサーや電池ボックスを取り付けて組み立てていく。使用目的や使用条件に即して製作品の機能と構造を考え、ものを加工する力をつける。

(3)後半6時間

プログラミングとセンサーカーの制御を学ぶ。ドリトルを用いて動作確認しながらプログラミングを行い、PICに送るプログラムを完成させていくものづくり方法を体感する。

最後に評価であるが、興味関心、作業工程の理解、創意工夫、センサーカーの出来具合、プログラムの出来具合と理解度などから評価することになると考える。

#### 4. むすび

中学校・技術科の情報領域において、プログラミング・制御が学習すべき内容として重要になる。その題材としてセンサーカーを取り上げて、プログラミング言語にドリトルを使用した教材を検討した。ドリトルでは、タートル用のプログラムはタートルの動きを画面上で見ながら作成していきけるが、センサーカー用のプログラムはPICに転送して実際に動かさないと動作確認することができない。そこで、タートル用のプログラムにリミットスイッチ（タッチセンサー）に関する記述をコメントとして追加すれば、センサーカー用のプログラムに変換できるような変換プログラムを製作した。その結果、ドリトルの表示画面上でセンサーカーの動作をタートルの動きで確認しながらプログラミングすることが可能になり、プログラミング学習をしやすくした。

今後は、PICのファームウェアについても取り組む必要があると考える。また、本教材を用いた授業実践に取り組んでいきたい。

#### 参考文献

- 1) プログラミング言語「ドリトル」, <http://dolittle.eplang.jp/> (2007).
- 2) 倉島宏和, “一人一人が自ら拓く技術・家庭科の学習 (PICとLEDを使った制御)”, 平成19年度長野県家庭科 技術・家庭科教育研究大会 技術・家庭科 学習指導案<技術分野> (2007. 6. 1).
- 3) ミュウロボ制御基板 MY-501, ロボット工房, スタジオミュウ (2007).

(2008年5月15日 受理)