ドリトルによる飛行船組み込みシステムのチーム開発を指向した総合的な学習

高瀬えりか† 江見圭司†

最近の組み込みシステムの大規模化は組み込みシステム開発者にチーム開発を余儀なくしている。通常のソフトウェア開発がコーディングを正しくすれば動作するのに対して，コーディング以外にハードウェアの動作を観察することが重要になる。プログラミング修得が比較的容易なドリトルを用いることによって，飛行船組み込みシステムのチーム開発を指向した総合的な学習を提案し実践した。

'Comprehensive School Hours' That are Oriented for Team Development of Embedded Systems of a Small Airship Robot with Programming Language Dollitle.

Erica TAKASE† Keiji EM †

Embedded systems development has recently become large-sclae,
so team development has become more important than before. In usual software development, we have only to write program coding correctly.
On the other hand, in embedded systems development, the observation of hardware is very important. We use easy-to-learn programming language dolittle for embedded systems of a small airship robot.
We propose 'comprehensive school hours' that are oriented for team development and practiced this.

1. はじめに

1.1.組み込み業界の人材不足

経済産業省による組み込みソフトウェア産業実態調査によれば，2005 年度報告の時点での組み込みソフトウェア技術者は 17 万 5 千人であり，7 万人不足している。[1] 2008 年の報告では，組み込みソフトウェア技術者数は 24 万 2 千人と増加しているが，不足者の数も 8 万 8 千人となっている。[2][3] また，半導体技術の発達に伴い，組み込みシステムの適用範囲は広がり，システム自体の規模も数年の間に急激に拡大している。かつては万能職人的なエンジニアが少人数で開発に従事する傾向が強かったが[4]，システム規模の拡大に伴い，開発の分業化・チーム開発化が進んでいる。

†京都情報大学院大学
The Kyoto College of Graduate Studies for Informatics

文献[2]によると，職種別での不足率は，以下の一覧の通りである。マネジメント系など高度 IT 人材の不足率が高く，現状では多人数での開発に適した開発環境であるとは言い難い。

・プロダクトマネージャ 64.9%
・プロジェクトマネージャ 46.2%
・システムアーキテクト 44.5%
・ソフトウェアエンジニア 25.9%
・テストエンジニア 29.0%
・ドメインスペシャリスト 47.0%
・QA スペシャリスト 55.1%
・開発プロセス改善スペシャリスト 42.6%
・開発環境エンジニア 50.7%
・ブリッジ SE 48.7%

このような状況において，産業界で必要とされているチームでの組み込みソフトウェア開発への興味や意欲を楽しみながら養うとともに，実際に組み込み業界で働く働かないに関わらず，身のまわりにあ

るコンピュータやシステムについて考えるきっかけを提供したいと我々は考える。

1.2.総合的な学習の時間

組み込み開発の流れを理解し，チーム開発を行うことの出来る人材を育成できるような授業は，従来の教科や科目では設定が難しいと考える。

現在，小・中・高等学校等においては，学習指導要領に基づき，「総合的な学習の時間」が実施されている。「総合的な学習の時間」の目標では，
横断的・総合的な学習や探求的な学習を通して，自ら課題を見付け，自ら学び，自ら考え，主体的に判断し，よりよく問題を解決する資質や能力を育成するとともに，学び方やものの考え方を身につけ，問題の解決や探求活動に主体的，創造的，協同的に取り組む態度を育て，自己の在り方生き方を考えることができるようにすることを目標として掲げている。[9]

なお，本稿でとりあげる学習内容が教科「情報」のみならず，「総合的な学習の時間」の内容も含んでいることについては，2.3.「相撲競技の概要とチーム開発」の部分で述べる。

本論文では，大阪府立桃谷高校の生徒を対象とした総合学習での授業実践計画について述べる。この実践においてはラジコン操作することのできる飛行船を教材として用い，実際に動作させながらハードウェアが思い通りに動かないという問題に主体的，創造的，協同的に取り組む授業を提案し，実践する。

2. 実習環境

2.1.プログラミング

実習で用いるプログラミング言語には「ドリトル」を用いる。ドリトルとは，日本語で記述可能な教育用のプログラミング言語である。[7][8]

Java や C 言語など汎用的なプログラミング言語では，用意されている基本的なソフトウェアに手を加えるといえ，基礎的な知識のない生徒であれば言語の理解に注力せざるを得ず，単にプログラミング言語習得のための授業で終わってしまう可能性がある。

一方，ドリトルは日本語でソースコードを記述することが可能である。今回のようにプログラミング初心者で，かつ日本語のネイティブスピーカーを対象とした授業，ドリトルはソースコードが理解しやすく，総合的な理解を深めることが可能となると考える。

実習の段階ではラジコン操作で飛行船を動作させる為の基本的なソフトウェアは既に用意している。

2.2.ハードウェア

ハードウェアは，基地局 PC，地上 MPU（Micro Processing Unit,図 1 参照），2 つ合わせて地上局という。飛行船はバルーンと（図 2 参照），機体搭載 MPU から成っている。基地局 PC からの入力は地上 MPU を通して無線で機体搭載 MPU に送信され，飛行船に取り付けられたプロペラが回転し，飛行船が動作する。[5]

飛行船は，ヘリウムガスによって浮かんでいるが，上昇・下降・前進や旋回などの動作はプロペラの回転によって変化する。飛行船には，左プロペラ，上下プロペラ，右プロペラの 3 つのプロペラが取り付けられている。（図 3 参照）実習の段階では，飛行船を操作するためのハードウェア環境は既に用意されており，生徒が直接手を加えることはない。

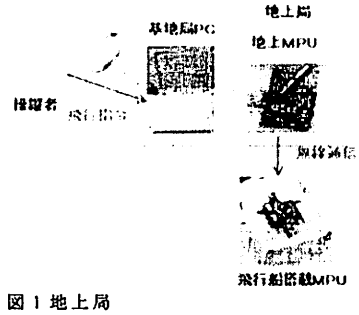

図 1 地上局


図 2 飛行船

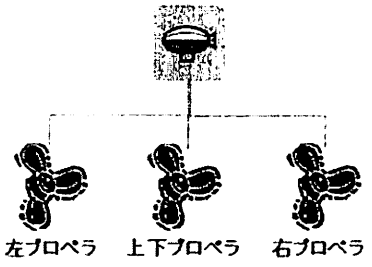

図 3 飛行船とプロペラ

2.3.相撲競技の概要とチーム開発

相撲競技とは，飛行船をラジコンで操作して1対1で対戦し，勝敗を競うものである。（図４参照）この競技は，情報処理学会　組込みシステム研究会の組込みシステムシンポジウム（ESS）で実施されているコンテストの競技の１つである。この相撲競技のエンターテイメント性に着目し，生徒の意欲的な学習を促すためにこの競技を授業内で用いたいと考えている。

ESS で実施されている競技規定では，相手の飛行船を壁に押しつけると勝ちというルールの他に，各チームの飛行船の端にひもをつけ，そのひもが床につくと失格，などのルールもあるが，今回の授業でははは相手の飛行船を壁に押しつけた時点で勝ち，というルールのみで行う。

また，操作はコントローラやマウス，キーボード，ゲームのコントローラなどを用いて行う。ESSでの競技規定では，飛行船の仕様も各チームで異なっているが，今回の授業ではマウスを用いて操作を行い，飛行船も同じ仕様の機体同士で対戦を行い，各チームで改変する部分はソフトウェアのソースコード，そしてキーボードやマウスなどの操作部分のみとする。

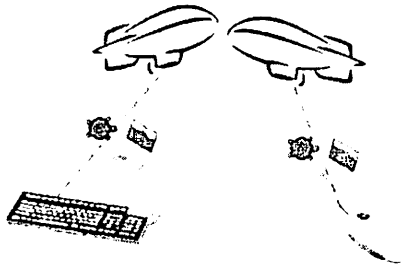

図４　相撲競技

従来の情報系の教育では，「観察する」という行為が少ない。[6]組み込み開発の場合は，成果物の動作確認をする場合に，対象となるハードウェアの動きを観察することが必要となる。観察が不十分な場合，動作の不具合がソフトウェア・ハードウェアのどちらに起因するものであるか判断することが難しい。また，次のプロセスの基となる要求を見つけ出せず，開発に支障をきたす場合もある[10]。

さらに，複数人での観察やその後の議論によって，問題や要求が見つけやすくなる。

このことより，チームでの観察によって分析力・設計力を養うような授業を行う必要があり飛行船を用いることによって，このような授業を行うのに適していると考えられる。

ソフトウェアを正しく作り，正しく入力すれば飛行船が思い通りに動作するように考えがちである。しかし実際にはヘリウムガスで浮かび小さなプロペラで動く飛行船は，空気の流れや室内の温度，周囲にあるものの動きなど，外的環境に動作がかなり左右される。そのため，成果物であるソフトウェアを正しく作っても，ハードウェアの観察無しには思い通りに動かない。

例えば，「飛行船を下降」させたい場合には下降の入力を行えば良いが，その時の気温や飛行船の重さによっては，「全停止」の入力を行っても「飛行船の下降」という結果が得られることは観察によって見つけることができるだろう。

また，操作者の操作の腕によっても飛行船の動作は左右される。

思い通りに動かない飛行船を，いかにして思い通りに動かし，面白い動作をさせたり，相撲競技に勝てるのかを考え，チームで議論することで生徒が楽しみながら学ぶことが出来るのではないかと考える。

また，この飛行船は制御速度が遅いため，入力に対する反応がリアルタイムで現れない。この反応の遅さが，生徒にとっては入力に対する反応を観察しやすいという利点となる。

3. 授業実践計画

3.1.はじめに

授業実践の計画にあたり，まず授業の構成を 1.ハードウェアと飛行船の飛ばし方，2.プログラミングとドリトルの概要，3.相撲競技の練習とチーム開発の３つに分割した。

これまで飛行船プロジェクトで，学園祭で飛行船を多くの方に操縦していてただいた結果，半数の方が飛行船の操縦が全く行えかなった。その経験をふまえて飛行船の飛行特性や操縦方法を習得するための時間が必要であると考えた。

ドリトルに関して京都コンピュータ学院の春期講習会（１週間の集中講義）で授業実践経験から，１回分の授業（１時間～３時間）を必要とすると判断した

3.2.ハードウェアと飛行船の飛ばし方

この授業では，飛行船の動作を図や実際に飛ばして簡単に説明し，どのように飛行船が飛ぶのか理解させ，実際の開発への意欲を高めたい。

1)まず，教師は飛行船を教室で浮かべ，この飛行船が，地上局 PC から操作画面で命令を入力されることによって，地上 MPU から無線で機体搭載 MPU に命令が伝わり，プロペラを回転させることを解説する。

前述したように，飛行船には３つのプロペラが取り付けられている。（図３参照）このプロペラを指しながら，教師はプロペラの回転と推力の方向の原理的な関係を解説する。

例えば，前進であれば左プロペラと右プロペラが回転し，空気を後ろ方向に加速して推力を得る。つまり，プロペラが押し出した空気と反対方向に飛行船は進む。（図５参照）

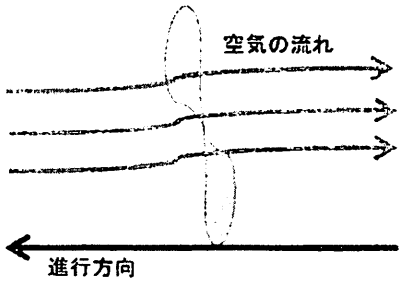

図５飛行船の回転と推力

2)次に，教師は用意した飛行船の操縦プログラムを起動し，画面を示す。（図６参照）

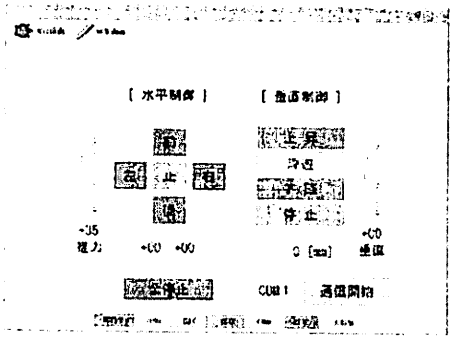

図６操作画面例

3)画面を操作しながら，プロペラの回転と飛行船の動きについて解説する。まず，[前]と記されてるボタンをクリックする。すると，左右それぞれのプロペラが，後ろ方向へ空気を加速するような回転を始め，飛行船が前進を始める。この場合の，空気を後ろ方向に加速する回転を＋方向への回転へとすると，飛行船が前進する場合のプロペラの回転は

左プロペラ：＋方向への回転

右プロペラ：＋方向への回転

となると言える。

後退であれば，プロペラの回転は

左プロペラ：＋方向への回転

右プロペラ：＋方向への回転

となると言える。

このように，まず飛行船の前進・後退について動作例を説明する。

4)上記に加えて，左右旋回や上昇の場合の操作方法とプロペラの関係も説明する。

左右旋回の場合は，左右プロペラがそれぞれ逆回転の動作をするため，プロペラと動作の関係をすぐに理解しずらい。この動作の解説の前には，生徒に予想させ，考えさせる時間を取ることで理解を深めたい。

5.全体をまとめると，それぞれのプロペラの回転方向と飛行船の動作の対応が下記の表のように表すことができると解説する。（図７参照）

上昇	＋	
下降	－	
前進	＋	＋
後退	－	－
左旋回	－	＋
右旋回	＋	－
全停止		

図７プロペラ回転と飛行船動作の対応

5)教師が[前]と記されているボタンをクリックしても，実際の教室ではエアコンなどの空気の流れや周りの生徒の動きなどに影響されて，飛行船は前方向に直進し続けることは少ない。これは飛行船が故障しているためではなく現実の世界では原理に加えて様々な要因されて動作すること，それを考慮して，実際の飛行船の動きを観察しながらドリトルのプログラムをチームで改良することが実習の目標であると説明する。

3.3.プログラミングとドリトルの概要

次に，実習の前段階として，飛行船操作のためのソフトウェアに手を加えるためのプログラミング言語，ドリトルの最低限の講義を行いたい。基本的な飛行船操作のソースコードは別紙参照されたい。このソースコードを用いて，ドリトルでのオブジェクトの作り方や命令など基本的なコードの書き方，ハードウェアとの通信の方法などについて講義を行いたい。白紙のエディタから作成していく方法の講義

も考えられるが，今回は，プログラミング言語取得が最優先課題ではないこと，限られた時間内での授業実践であることから，ソースコードを用いた方法での講義を行う。

3.4.相撲競技の練習とチーム開発

飛行船の飛ぶ仕組みと，ドリトルの基礎知識が理解できれば，実際の競技のために偶数チームに分けた後，相撲競技のためのチーム開発と練習を行いたい。

実習の流れは，プログラミング→成果物の観察→チームでのディスカッション→プログラミング，としたい。

まずはそれぞれのチームで，操作画面（図6参照）を見ながら上昇，下降，前進，後退，旋回など基本的な操作を体験し，飛行船の飛ばし方の項目で学習したように飛行船が動作しているか観察させる。

飛行船の入力に対する動作の様子を観察した後，チームでどのような飛行をさせたいか，また，どのような動作が出来れば相撲競技に勝てるか，面白い動作は出来るのか，競技の作戦や目標などを考えさせる。

ある程度方向性が決まれば，実際に基本のソフトウェアのソースコードに手を加えていく。ソースコードに手を加えながら，実機を動かし，コードの変更が反映されているか，不具合があればそれは外部環境とソースコードのどちらに問題があるのか観察させる。

開発が進む中で個々の得意分野に応じて，チームの中で「ソースコードを書き換える者」と，「飛行船の動作を観察し，案を出す者」，「飛行船を操縦する者」など，分業されていくと考えられる。このような流れの中で，チームでの組み込み開発を楽しんで理解できることをねらいとしたい。

3.5.相撲競技と授業のまとめ

授業の最後に，各チーム毎に相撲競技を行い勝負させる。成果物は，それぞれのチームが改変した飛行船操作のソフトウェアとする。成果物は，思い通りに動くソフトウェアが作れたか，チームで効率よく分担して作業を進められたか，改善点があるとすれば何かなど，競技終了後に各チーム自身で評価を行わせたい。

最後に，実際の組み込みソフトウェア開発などの例も紹介し，身のまわりの組み込みシステムや，開発に対しての興味や意欲を促したい。

4. 実践結果とまとめ

4.1.概要

授業実践計画に基づき，7月7日に桃谷高等学校にて授業を実施した。

授業時間は，45分×4コマであり，当初予定していた授業計画の全てを行うことが出来ず，内容を一部変更して行った。

対象とする生徒数は24人であった。

4.2.授業の流れ

1)まず，授業前に生徒にアンケートを行った。

2)次に，3.2.節の通り，ハードウェアと飛行船の飛ばし方について解説を行いった。実際に飛行船を動かしながら，生徒には実際の動作とプロペラの動作について観察するよう促した。

また，ミュウロボを用いてコンピュータの入力と実際の動作について解説し，ロボットがコンピュータで制御出来ることを協調した。

3)今回は，教室後方のゴール地点飛行教室内のスタート地点からゴール地点を目指して着地し，スタートからゴールまで到達する時間を競うという競技を行い，飛行船の制御を体験させた。スタート地点からゴール地点までの距離は，約6メートルである。（図8参照）

競技はチーム単位で行い，4人前後で1チームとして役割分担を行った。まず，教室前方に飛行船を操作するパイロットとパイロットの隣でアドバイスを行うサブパイロット，そして教室後方では，ゴール地点に近づいた飛行船の位置や進むべき方向をパイロット側に伝えるナビゲータ2人を配置する様に役割を分担した。

その後，チーム毎に競技を行った。

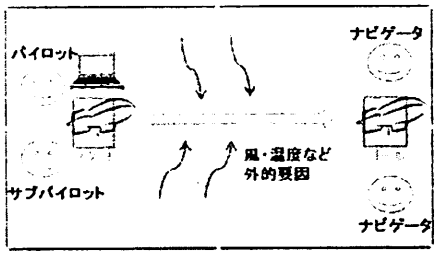

図8競技の概要

4)競技が終わった後は，以下の2点について解説を行い，生徒の理解を促した。

(1)飛行船やミュウロボなどは組み込み制御で動作しており，ソフトウェアの不具合などの内的要因だけではなく周囲の環境などの外的要因にも左右さ

れること。

(2)このような組み込み製品は，例えば携帯電話や炊飯器など，生徒達の身の回りにたくさん存在していること。

5)最後に生徒にアンケートを行い，この授業を終了した。

当初の計画では，3.3.「プログラミングとドリトルの概要」，3.4.「相撲競技の練習とチーム開発」に基づいて授業を行う予定であったが時間の都合上割愛した。

図 9 飛行船を飛ばし始め，競技を行っている様子

4.3.競技のまとめ

競技でのスタート地点からゴール地点に到達するまでの所用時間は各チームで 5 分～ 10 分程度であった。

飛行船の動きとプロペラの動きとの対応や，微かな空気の流れで流されてしまう飛行船の特性を良く観察出来ていたチームではメンバー同士で操作や飛行船の進行方向について指示を出し合っており，その為ゴール地点までの所要時間も比較的短かった。

4.4.アンケート

授業前と，授業後に行ったアンケートでは以下の結果が得られた。

[受講前アンケート　有効回答数 24 人　]

1-1 飛行船を制御する授業を受けたことはありますか？　ある:4 人　ない:20 人

1-2.ミュウロボや，飛行船などのロボットがコンピュータで制御できることを知っていますか？あるいは聞いたことがありますか？

よく知っている:0 人　知っている:4 人　聞いたことはある:7 人　あまり知らない:3 人　全く知らない:10 人

1-3.コンピュータ制御は，自分にとって役立つと思いますか

思う:2 人　やや思う:4 人　どちらでもない:13 人　あまり思わない:1 人　全く思わない:4 人

[受講後アンケート　有効回答数 24 人]

2-1.ミュウロボや，飛行船などのロボットがコンピュータで制御できることを理解できましたか？

思う:9 人　やや思う:7 人　どちらでもない:7 人　あまり思わない:0 人　全く思わない:1 人

2-2.コンピュータ制御は，自分にとって役に立つと思いますか。

思う:1 人　やや思う:5 人　どちらでもない:10 人　あまり思わない:5 人　全く思わない:3 人

アンケート 1-2.，2-1.の結果から，授業前にミュウロボや飛行船などのロボットがコンピュータで制御出来ることを聞いたことがある程度でも知っていた生徒の割合は[よく知っている:0 人　知っている:4 人　聞いたことはある:7 人]と生徒 24 人のうち半数以下であるが，授業後にこれらの事を理解できたと考えている生徒の割合は[思う:9 人　やや思う:7 人]となった。このアンケート項目についての生徒の理解は，この授業を受けたことによるものか，あるいは元々あった知識を深めるのに役立ったと考えられる。

反面，1-3.，2-2.の結果から，コンピュータ制御は自分の役に役立ちそうか思うかとの問いに対し，授業前は[どちらでもない:13 人　あまり思わない:1 人　全く思わない:4 人]から，授業後での質問では「」どちらでもない:10 人　あまり思わない:5 人　全く思わない:3 人]となった。この結果からは，この授業で組み込みなどのコンピュータ制御を理解することによって，自分にとっては有用ではないと判断した生徒が増えたためはないかと考えられる。

5. 終わりに

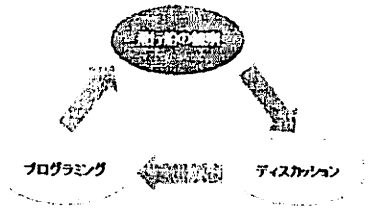

図 10　我々の想定する問題解決のプロセス

産業界での組み込みソフトウェア人材不足の状況は今後もしばらく続いていくと考えられる。

上記の表の問題解決プロセスは組み込み開発に欠かせない，実世界の観察とチームでのディスカッションを含めており，このプロセスを用いて組み込みソフトウェア開発への興味や意欲を楽しみながら養

えるような教育を行いたいと考えている。今回は時間の都合上授業内で割愛せざるを得ない部分が多かった。今回の授業の反省点を踏まえ，次回の授業実践に活かしていくことを今後の課題としたい。

謝辞

ドリトルの使用にあたり，大阪電気通信大学の兼宗進教授，大阪府立桃谷高校での授業にあたり，野辺緑教諭，相撲競技という競技種目・名称を用いるにあたり，MDD ロボットチャレンジ 2009 実行委員会技術顧問　二上貴夫氏，飛行船操作用のソフトウェア作成にあたり，京都情報大学院大学の村上智史氏，ハードウェアの開発にあたり，株式会社ヒューマンエンジニアリングロボティックスの岡村勝 氏，西村憲二 氏，高橋嘉也 氏，竹内勇貴 氏，松井委宏樹 氏，岡部拓矢 氏，古川彬 氏，のご協力を賜りましたことを感謝いたします。

本研究は，上月スポーツ・教育財団より，第６回「エデュテイメント開発研究助成事業」を受けています。

付録

```
// 基本設定 //
ポート番号＝『COM1』。　制限時間＝180秒。推力＝35。
シリアル＝シリアルポート！作る。
シリアル：送信＝「」。
シリアル：送信処理＝「
  送信＝「」。
  ！(ポート番号)開く
    (『S』！(！(垂直)文字化)(！(左)文字化)(！(右)文字化)『$』連結)出力
    閉じる。
  送信＝送信処理」。
文字化＝「｜ｖ｜ｖ＝round(ｖ)%100。
  「ｖ＞＝０」！なら「『+』！((ｖ＜10)！なら「『0』」そうでなければ「『』」実行)(ｖ)連結」
      そうでなければ「『-』！((ｖ＞-10)！なら「『0』」そうでなければ「『』」実行)(ｖ＊-1)連結」実行」。
垂直＝０。左＝０。右＝０。
制限時間表示＝ラベル！("残り時間："！(制限時間)『秒』連結) 作る　-95 -105 位置　300 50 大きさ。
終了音＝楽譜！『ピアノ』作る　(メロディ！作る『C8C1～』追加)設定。
開始＝ボタン！"開始" 作る　192 192 192 色　-200 -110 位置　90 40 大きさ。
開始：動作＝「
  シリアル：送信＝シリアル：送信処理。
  ！80 80 80 色。
  時計＝タイマー！作る　1秒　間隔　(制限時間)　時間
  「｜回｜制限時間表示！(『残り時間：』！(制限時間－回)『秒』連結)書く」実行
    1 回数 「制限時間表示！『時間切れ！』書く。全停止：動作！実行。シリアル：送信＝「」。終了音！演奏」実行。
  動作＝「」。」。
前進＝ボタン！"前" 作る　128 255 255 色　-130 85 位置　60 50 大きさ。
前進：動作＝「ルート：左＝推力。ルート：右＝推力。シリアル！送信」。
後進＝ボタン！"後" 作る　128 255 255 色　-130 -35 位置　60 50 大きさ。
後進：動作＝「ルート：左＝推力＊-1。ルート：右＝推力＊-1。シリアル！送信」。
左旋回＝ボタン！"左" 作る　128 255 255 色　-200 25 位置　60 50 大きさ。
左旋回：動作＝「ルート：左＝推力＊-1。ルート：右＝推力。シリアル！送信」。
右旋回＝ボタン！"右" 作る　128 255 255 色　-60 25 位置　60 50 大きさ。
右旋回：動作＝「ルート：左＝推力。ルート：右＝推力＊-1。シリアル！送信」。
止＝ボタン！"止" 作る　255 128 128 色　-130 25 位置　60 50 大きさ。
止：動作＝「ルート：左＝0。ルート：右＝0。シリアル！送信」。
上昇＝ボタン！"上 昇" 作る　55 90 位置　155 40 大きさ。
上昇：動作＝「ルート：垂直＝90。シリアル！送信」。
下降＝ボタン！"下 降" 作る　55 40 位置　155 40 大きさ。
下降：動作＝「ルート：垂直＝-90。シリアル！送信」。
停止＝ボタン！"停 止" 作る　255 128 128 色　55 -45 位置　155 40 大きさ。
停止：動作＝「ルート：垂直＝０。シリアル！送信」。
全停止＝ボタン！"全停止" 作る　255 80 80 色　-80 -180 位置　160 40 大きさ。
全停止：動作＝「ルート：垂直＝０。ルート：左＝０。ルート：右＝０。シリアル！送信」。
ラベル左右＝ラベル！『[ 水平制御 ]』作る　-175 160 位置。
ラベル垂直＝ラベル！『[ 垂直制御 ]』作る　60 160 位置。
// デバッグ用 //
ダンプリスト＝リスト！作る　消える　-380 240 位置　200 200 大きさ。
ｐ＝「ダンプリスト！現れる　(自分)書く。自分」。
```

参考文献

[1]経済産業省:2005 年版組み込みソフトウェア産業実態調査
http://sec.ipa.go.jp/download/200506es.php

[2]経済産業:2008 年版組み込みソフトウェア産業実態調査報告書の公表について
http://www.meti.go.jp/policy/mono_info_service/joho/2008software_resarch.html

[3]京都コンピュータ学院:特集 IT 業界が求める人材 IT 人材の不足～現状と課題，京都コンピュータ学院校友会機関誌 Accumu,Vol17(2009)

[4]中小企業基盤整備機構:平成 19 年度ナレッジリサーチ事業
http://www.smrj.go.jp/keiei/chosa/032268.html

[5] 中村州男，ほか:教育用プログラミング言語を用いたプロジェクト工数削減と多段階開発の実践，情報処理学会，Vol.55,pp.123-130(2008).

[6] 山口淳一，ほか:オブジェクトモデリング(UML)を用いた組み込みソフトウェア開発技術者養成プロジェクト，JPSJ Sympsoium Series，Vol.2004，(2004).

[7](a)兼宗進，久野 靖:ドリトルで学ぶプログラミング-グラフィックス、音楽、ネットワーク、ロボット制御，イーテキスト研究所(2008)
(b)兼宗進:プログラミング言語「ドリトル」
http://dolittle.eplang.jp/

[8]文部科学省:高橋修司，ほか:ドリトルを用いたモデル化・シミュレーション・オブジェクト指向開発の自学自習実践,第 94 回コンピュータと教育研究発表会 (2008)

[9]文部科学省:小・中・高校教育に関すること（総合的な学習の時間）
http://www.mext.go.jp/a_menu/shotou/sougou/main14_a2.htm

[10]Erica TAKASE, Kunio NAKAMURA, Shuji TAKAHASHI, Satoshi MURAKAMI Keiji EMI:The Practice of Project-based Learning on IT Engineer Education by Using Japanese -based Programming Language "Dolittle", JeLa 2009 年度国際シンポジウム(2009)

著者紹介

高瀬えりか（学生会員）
京都情報大学院大学
応用情報技術研究科
ウェブビジネス技術専攻
修士課程
専修学校
京都コンピュータ学院勤務

江見圭司（正会員）
京都情報大学院大学
応用情報技術研究科
准教授..