

五月祭 2018 プログラミング教室

～ロボットカーを動かしてみよう～

工学部機械工学科・機械情報工学科

0. はじめに

プログラミングってなに？

コンピュータを動かすためには、コンピュータにどのように動いてほしいかを教える”プログラム”がなければいけません。このプログラムを作ることが”プログラミング”といいます。たとえば、「マウスを動かすとパソコンの画面上でもマウスの矢印が動く」「スマホのアプリをタッチするとゲームが始まる」「リモコンを押せばエアコンがつく」などの動きにはすべてプログラミングが使われています。プログラミングができるようになれば、自分でゲームを作ったり、地震や火事から人を助けるロボットを作ったりするのに役に立ちます。

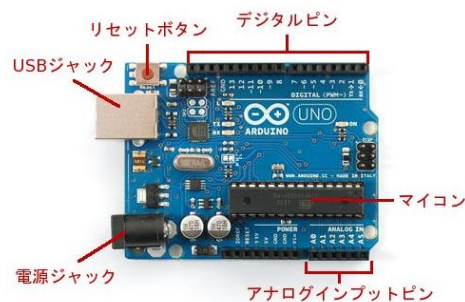
ロボットを動かそう

ロボットを動かすにはどうしたらよいでしょうか。ロボットは「ハードウェア」と「ソフトウェア」の2つの部分で作られています。ハードウェアはロボットの体、ソフトウェアはロボットの頭脳のことです。ハードウェアを動かすには、ロボットの頭脳であるマイクロコンピュータ(マイコンとも)にプログラムを書いていかなければなりません。

五月祭では、両手サイズのミニカーをみなさんに動かしてもらおうと思います、Arduino(アルディーノ)というマイコン(頭脳)にプログラムを書きこんであげることで、ミニカーにどのように動いてほしいかを教えてあげることになります。

Arduino とは

Arduino(アルディーノ)とは、1枚のマイコンボードと、プログラムを開発するための統合開発環境「Arduino IDE」でできているマイコンシステムです。



上にあるのが Arduino とよばれるマイコンボードです. いろんな部品がついていますが, 特に大切なところだけ紹介していきます.

USB ジャックは USB ケーブルでパソコンと Arduino をつなぎ, プログラムを書きこむときに使います.

デジタルピンやアナログピンがありますが, デジタルとアナログのちがいはなんでしょう. アナログは「連続的」に変わるもの, デジタルは「段階的」に変わるものです. たとえばアナログ時計では時計の針がずっと回りつづけるのに対して, デジタル時計は 12:00 の次は 12:01, 12:02 と段階的に変わっていきます. 12:00 と 12:01 の間の時間の長さは表示されません. 「0 から 1 まで」がアナログ, 「0 か 1 のどちらか」がデジタルというイメージです.

アナログ入力ピンは, 連続したアナログ値を読みとりデジタル値(0, 1, 2, ..., 1023 の数字)に変換します.

アナログ出力ピンは, PWM制御とよばれる方法をつかうところです. オンオフの比率を 0~255 の数字にきめることで電圧を変化させることができます. とてもざっくりとした言い方だと, 「0 はずっとオフで電流が流れない, 数字を 255 に増やすにつれて流れる電流も大きくなる」と考えてもらうといいです.

デジタルピンは, 指定したピンに「電流が流れない(0)」か「電流が流れる(1)」かのどちらかの出力を決めたり, どちらかの入力を読みとったりできます.

ArduBlock とは

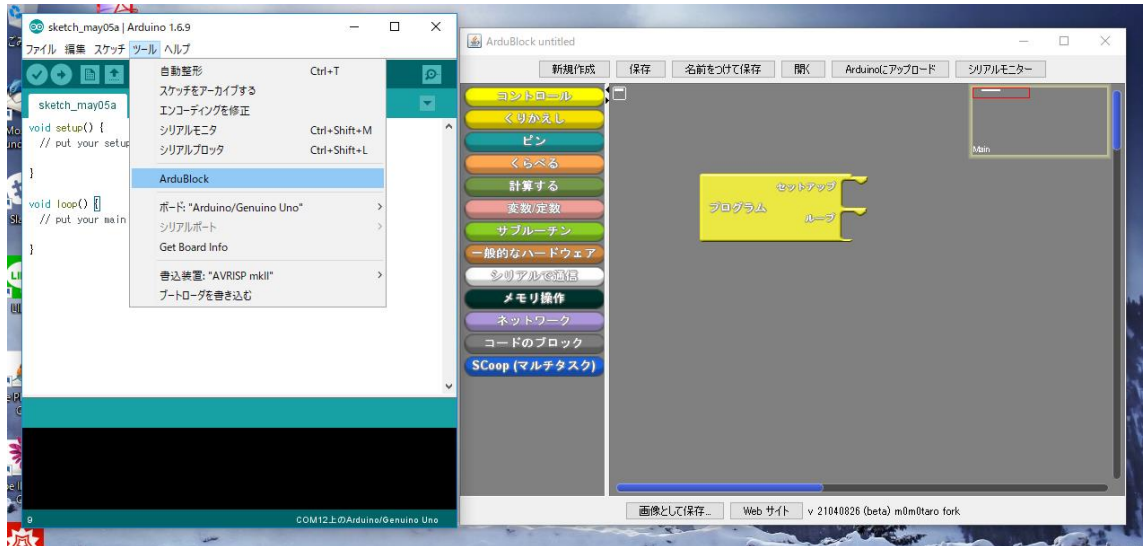
ArduBlock(アルドゥブロック)とは, Arduino のためのビジュアルプログラミング言語で, ブロックをドラッグ(マウスをクリックしたままマウスを動かす)やドロップ(マウスをはなす)を使うだけでプログラムができるので, 小中学生でも学びやすいものになっています. Scratch という有名なビジュアルプログラミング言語があるのですが, その Arduino バージョンだと思ってください.

1. Arduino と ArduBlock のインストール

Arduino と ArduBlock がセットで圧縮された zip ファイル [arduino-1.6.9withArduBlock_m0m0.zip](http://www.bkpoo.net/greatfreesoft/39-ardublock.html) を <http://www.bkpoo.net/greatfreesoft/39-ardublock.html> のサイトの中から見つけて, 適当な場所で解凍しましょう. Arduino のインストール作業はいらないはずです.

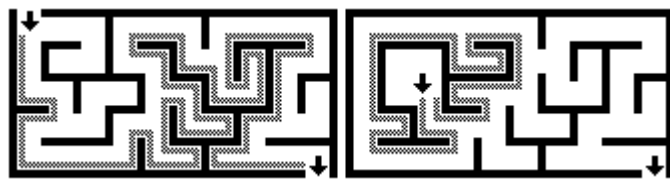
2. Arduino と ArduBlock の起動

arduino-1.6.9withArdublock_m0m0 フォルダの arduino.exe アプリケーションをダブルクリックし, Arduino を起動します。メニューの「ツール」から「ArduBlock」を選択すると, ArduBlock の画面が開けます。Arduino と ArduBlock の両方の画面はずっと立ち上げておきましょう。

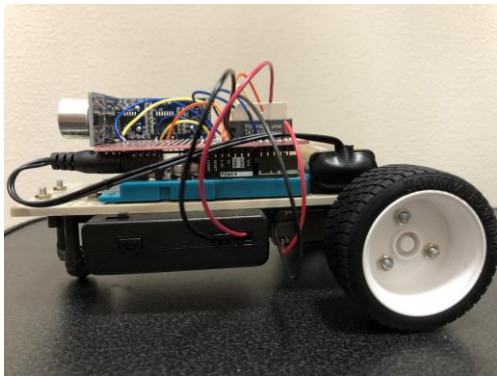


3. 車を動かして迷路をクリアしよう

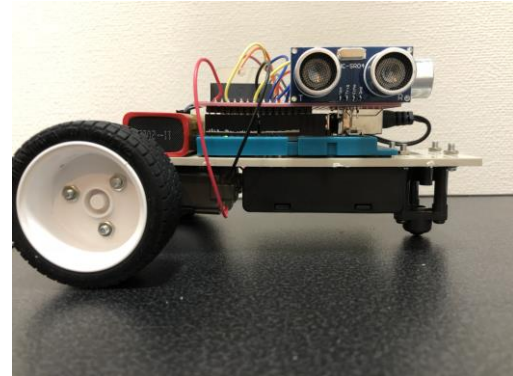
五月祭では, Arduino を乗せた車を動かし, 右手法という方法を使って迷路をクリアすることを目標とします。



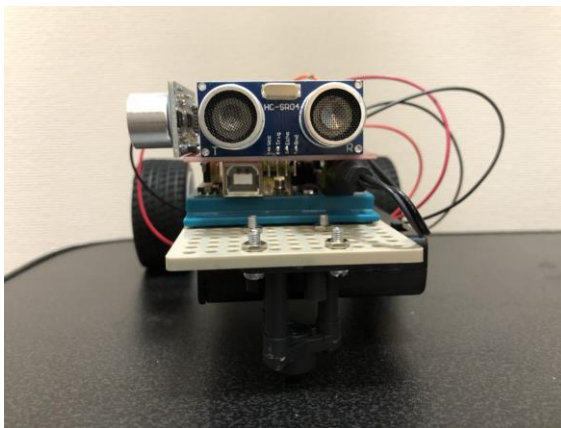
右側のカベに手をつきながらひたすらカベ沿いに進む方法を右手法といい, 図の左上のように迷路をクリアすることができます。ただし図の右上のようにスタートまたはゴールが迷路の中にある場合は無限ループとなってしまう右手法が使えないのですが, 五月祭では右手法が使えるような迷路を用意しているのでここでは考えません。



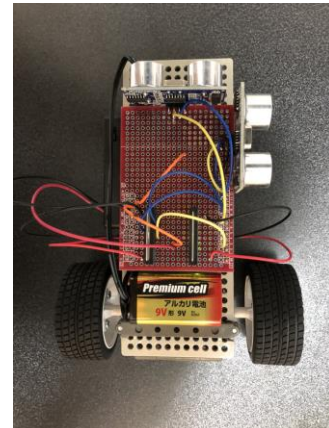
車を左から見た図



車を右から見た図



車を前から見た図

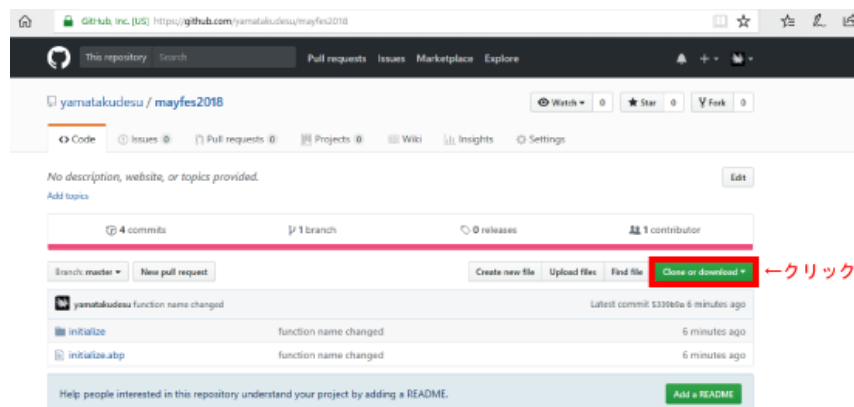


車を上から見た図

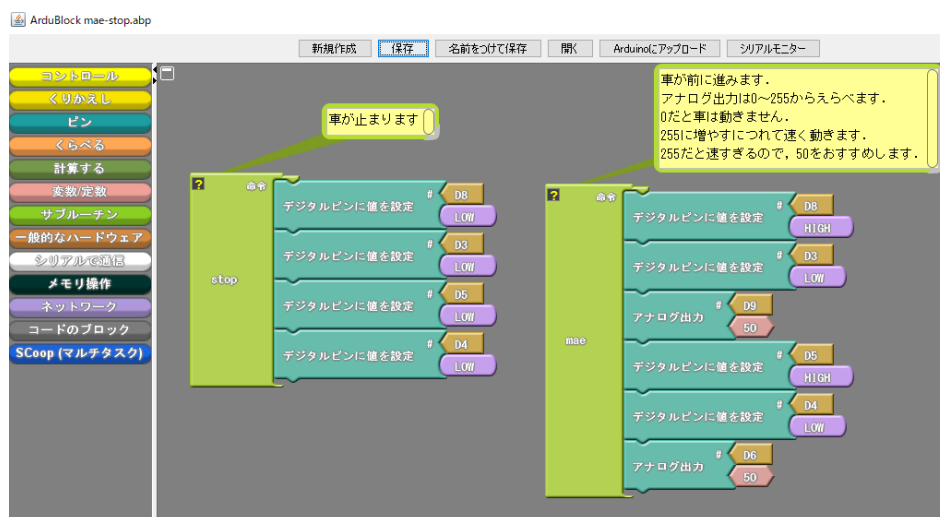
五月祭で使う車が上のようになります(配線やはんだ付けが下手なのは許してください). 前と右に1つずつ超音波距離センサがついており、センサから読み取った距離情報に応じてモータドライバでモーターを制御し車を動かす仕組みになっています. センサの仕組みとしては、超音波を発射し、それがモノにあたって返ってくるまでにかかった時間と、音の進むスピードからモノまでの距離が計算できるようになります.

とりあえず車を動かしてみよう

はじめに、「車が1秒前にすすみ、1秒とまるのを交互にくりかえす」プログラムをつくっていきましょう.



上のように <https://github.com/yamatakudesu/mayfes2018> から「Clone or download」のボタンをクリックし, Download ZIP をおして zip ファイルをダウンロードして解凍してください. 保存先は「arduino-1.6.9withArdublock_m0m0」フォルダ下だと分かりやすくていいとです. Arduino を起動→「ツール」メニューから「ArduBlock」を選択し ArduBlock を起動→「開く」メニューを選択→「mae-stop.adp」のファイルを選択, という風にして Arduino と ArduBlock を立ち上げましょう.



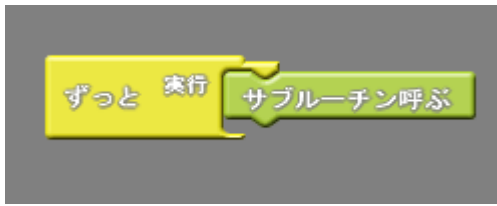
上のような画面が出たら OK です. 「mae」関数は前進する関数になっています. 8, 3, 9 ピンが左側のモーター, 5, 4, 6 ピンが右側のモーターを操作するピンになっておりそれぞれ「HIGH」と「LOW」の組み合わせでモーターの回転方向を決め, アナログ出力の数値(0～255)で回転の速さを決めています. 全てを「LOW」に設定した「stop」関数も用意しています. サブルーチン(関数)とは, まとまった動きをきめて, 必要ときにそれを呼び出せるようにする機能です.



まず、「コントロール」ボタンをクリックすると使える部品一覧が表示されるので、「ずっと実行」の部品を右のスケッチパネルにドラッグして置きましょう。部品を間違えておいてしまった場合は、その部品を部品パネルのところまでドラッグ&ドロップしてあげると消えます。「ずっと実行」は、プログラムのメインになる部分で、Arduinoの電源が切れるまで「ずっと実行」の中に入った動きがずっと無限にくり返されることを意味します。次に「ずっと実行」の中の動きを作っていきます。



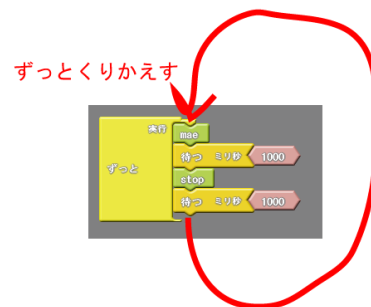
「サブルーチン」ボタンから「サブルーチン呼ぶ」の部品をスケッチパネルにある「ずっと実行」の右上にはまるようにドラッグすると、以下のように「サブルーチン呼ぶ」が「ずっと実行」の中にセットされます。



「サブルーチン呼ぶ」の文字の部分をクリックすると名前を変えることができますので、「mae」とうち Enter キーをおしてあげると、右上の図のようになります。ここでは、「mae」関数を呼び出しているので、「ずっと前に進む」というプログラムができたことになります。

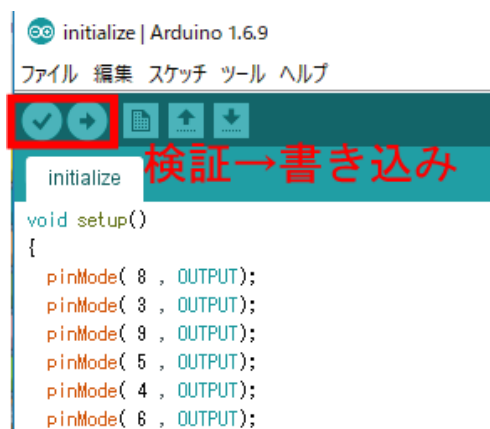


今度は、「コントロール」ボタンから「待つミリ秒」の部品を選び「ずっと実行」の中の「mae」の下までドラッグし、はなしてあげると上の図の右のようになります。



同様に、上の図の左のようなスケッチを完成させてみましょう。「待つミリ秒」は横についている数字の分だけ Arduino を待たせることを意味しています。ここでは、「mae」が実行されたまま 1000 ミリ秒、つまり 1 秒待つということになるので「1 秒間前に進む」ことになります。そして、「stop」が実行されたまま 1 秒待つので「1 秒間止まる」ことになります。これを合わせると「1 秒前に進み、1 秒止まる」となって、これが「ずっと実行」の部品の中に入っているので、結局このプログラムは「1 秒前に進み、1 秒とまるのを交互にくりかえす」ことを表しています。

プログラムができたので車を動かしてみましょう。Arduino の画面を開いた状態で ArduBlock の画面の「Arduino にアップロード」をクリックすると、さっき作ったものが Arduino にも^{ほんえい}反映されます。



パソコンとマイコンボードを USB ケーブルで^{せつぞく}接続してから、Arduino の画面のチェックマークをおすと、そのプログラムに^{まちが}間違い(エラー)がないかを確認してくれます。^{まちが}間違いがあると下の黒い画面に表示してくれます。次に^{やじるし}矢印マークをおすと、プログラムがマイコンに書きこまれ、車が動くようになります。動かすときは車の電源を on にしましょう。「マイコンボードに書きこもうとしましたが、エラーが発生しました」と言われた場合は、Arduino の「ツール」→「シリアルポート」で「COM○(数字)」と書かれた部分がクリックできているかたしかめてください。

前にカベがあったら車をとまるプログラムを作ろう

次に「車がまっすぐ進み、前にカベがあったら車をとまる」プログラムを作ってみましょう。ArduBlock の「開く」をクリックし、保存されるか聞かれた時は「いいえ」をクリックして、「mae-kabe-stop.adp」ファイルを選択しましょう。



上のような画面が出れば OK です。「ずっと実行」の中身についての説明です。^{へんすう}変数

「migikyori」に右についた超音波距離センサから読み取った距離、変数「maekyori」に前についたセンサから読み取った距離が整数値として入ります。「待つマイクロ秒」を挟んでいるのは、挟まないとなぜか2つのセンサ情報を読み取れないのでそのためのおまじないだと思ってもらっていいです。

ではプログラムを作っていきます。 「前にカベがある」ことを認識するためにはどうすればいいでしょうか。「前との距離が10cmより小さければ前にカベがある」と考えてみるとよいです。

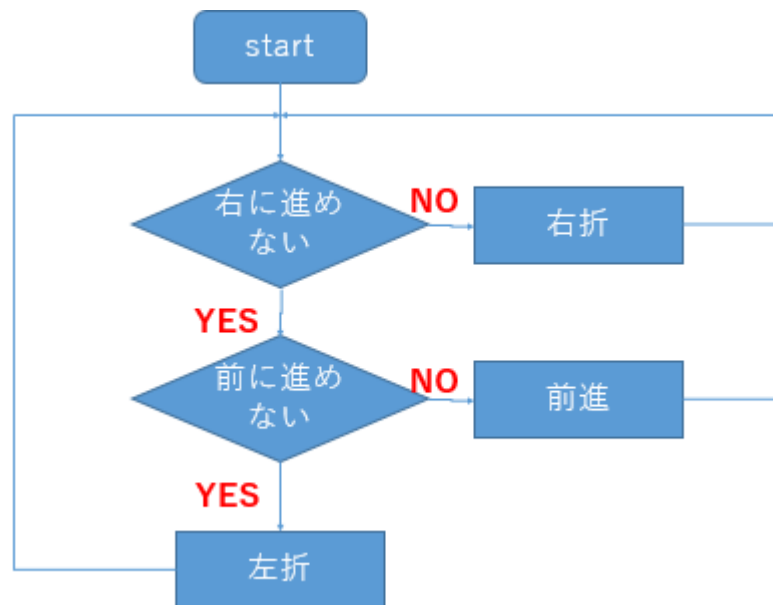


「くらべる」ボタンや「変数/定数」ボタンから部品をえらんで上のように作ってみましょう。これは前側の超音波センサから読み取った値、つまり前にあるモノとの距離が10(cm)より小さいということを表しています。

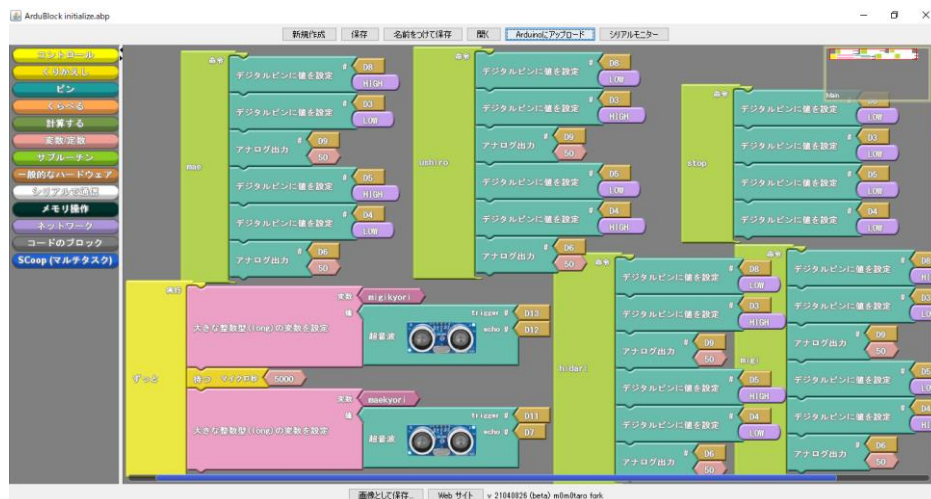


「コントロール」ボタンから上にあるような部品を、「サブルーチン」ボタンから「サブルーチン呼ぶ」部品をえらんで上のように作ってみましょう。左でも右でもどちらでもいいです。ここではプログラミングでとても大事な「条件分岐」というものを説明します。条件分岐とは、ある条件を満たす場合と満たさない場合でそれぞれ異なる条件を処理することをいいます。上の図の右の場合だと、条件「前との距離が10cmより小さい」に対してOKつまり前との距離が10cmより小さいなら車が止まり、NGつまり前との距離が10cm以上なら車が前に進むことを表しています。上の図の左のようにすると、条件を満たさない場合の動きは書かなくても大丈夫です。では、これを書き込んで車を動かしてみましょう。「Arduinoにアップロード」をクリックし、Arduinoの画面の「検証」→「書き込み」で車を動かしましょう。前に紙や手などをかざして車が止まればOKです。

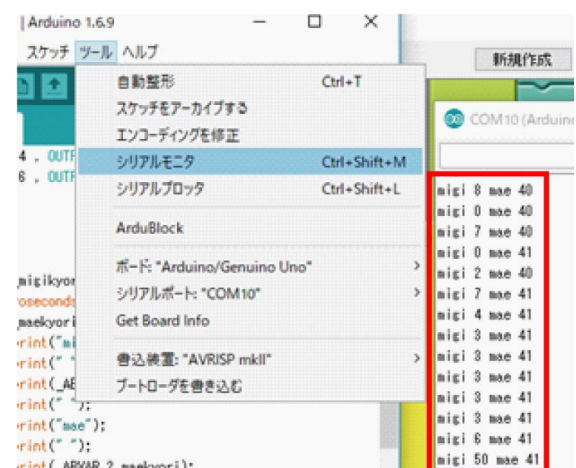
迷路の中で車を走らせてみよう



ようやく迷路を解くという本題に入ります。右手法のフローチャートが上ようになります。前と右の超音波距離センサ2つのだけを使って前進・右折・左折を切り替えていくことになり、みなさんにはこの部分のプログラムを作ってもらおうこととなります。それでは実際に Arduino と ArduBlock で車を操作していきたいと思います。



ArduBlockで「開く」から「right-hand-method.adp」をえらんで開いてください。上のようなスケッチが出ればOKです。



上のような「シリアルに出力」部品を使うと、Arduino の「ツール」→「シリアルモニタ」を選んだときにセンサから読みとった距離の数値がどうなっているのかをパソコン上で確認できます(USB ケーブルで接続している必要あり)。右上の赤枠でかこった部分に「migi 8 mae 40」などと書かれていますが、これは右のセンサから読み取った値が 8cm、前のセンサから読み取った値が 40cm であることを表しています。“前との距離が何 cm で右との距離が何 cm のときに左に曲がればいいのかなあ”などと思ったときにシリアルモニタの情報を参考にすることができますね。

では、右手法のフローチャートを見ながら、「条件分岐」やシリアルモニタの情報をつかって迷路をクリアするプログラムをつくってみよう！分からないことがあったら近くにいる学生にどんどん質問してください。

4. おわりに

みなさん楽しめたでしょうか。今日やったように、プログラミングができるようになれば車などのモノを自分の思い通りに動かすことができるようになります。もし、車をつくることから始めたいと思ったときは、タミヤの楽しい工作シリーズにいろいろなものがあるので調べてみるといいです。今回の車の材料を以下に紹介します。

- ・ Arduino マイコンボード
- ・ ダブルギヤボックス(左右独立 4 速タイプ) ータミヤ楽しい工作シリーズ
- ・ スポーツタイヤセット(56mm 径)ータミヤ楽しい工作シリーズ
- ・ ボールキャスター(2 セット入)ータミヤ楽しい工作シリーズ
- ・ ユニバーサルプレート(2 枚セット) ータミヤ楽しい工作シリーズ
- ・ 超音波センサ HC-SR04 2 個
- ・ モータードライバ TA7291P 2 個

- ・ユニバーサル基板
- ・ピンヘッダ
- ・バッテリースナップ
- ・電池ボックスリード線スイッチ付(単三4本)
- ・9V 電池
- ・単三電池 4 本
- ・その他(はんだ, はんだごて, ブレットボード, ジャンパワイヤ…)

他にも Arduino の使い方や車の組み立て, はんだ付けについてなど何か質問がありましたら, プログラミング教室の企画担当の山田 yamatakudesu123@gmail.com に気兼ねなく質問してください.

(右手法のヒント)

フローチャートを見ながら, 赤い四角の中をうめてみてね!(ただしこれではおそらく迷路をクリアすることができません)

