組込みシステム
開発実習

ビジュアルプログラミング編

本テキストを使用する為に必要なモノ

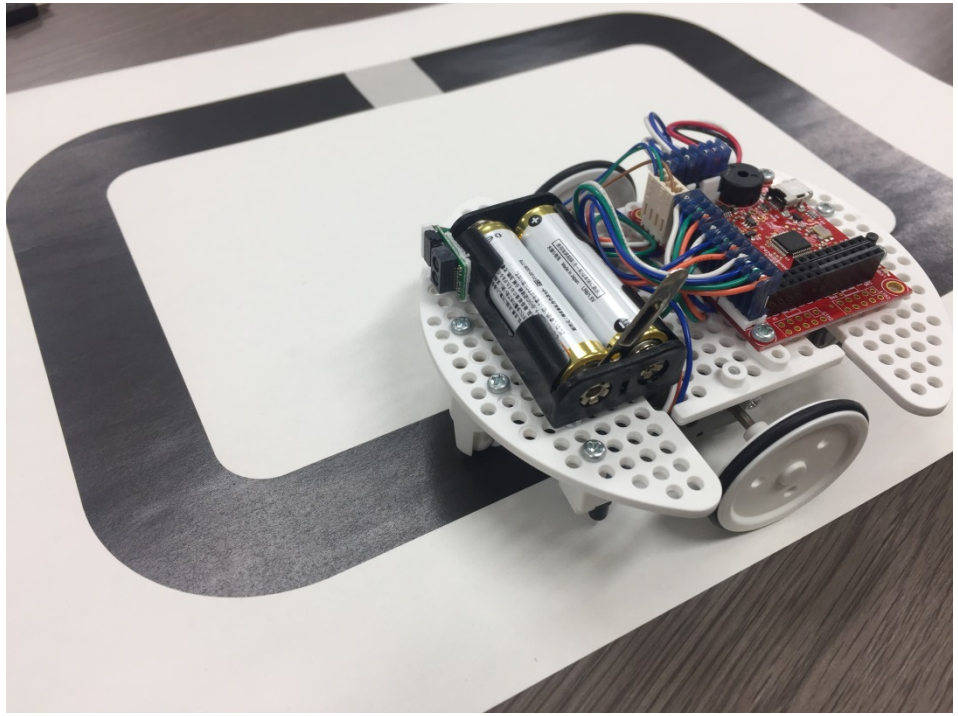
- 関連テキスト
  - ビュートビルダー２取扱説明書(ヴイストン社 著)
- 実習機器
  - パソコン
  - ビュートローバーARM
  - 単三電池×２
  - オーバルコース(A3サイズ)

推奨する動作環境

- OS　　: Windows2000/XP/Vista/7/8/8.1/10
- CPU　　: Pentium-III以降(1GHz推奨)
- RAM　　: 128MB以上
- I/F　　: USBポート(1個以上)
- 画面　　: XGA(1024×768)以上

ライントレースカー(LTC)とは？

- ライントレースカーは、線の上をなぞりながら走る自立制御型のロボットである。
- クラシックな組込み分野・マイコンの学習教材であり、組込み分野の大きな特徴である「論理の世界」と「物理の世界」を繋がりを感じ取る事を学習の目的とする。
- 尚、ライントレースカーは、ライントレースロボットやライントレーサーなどと呼ばれることもある。



動画はコチラ

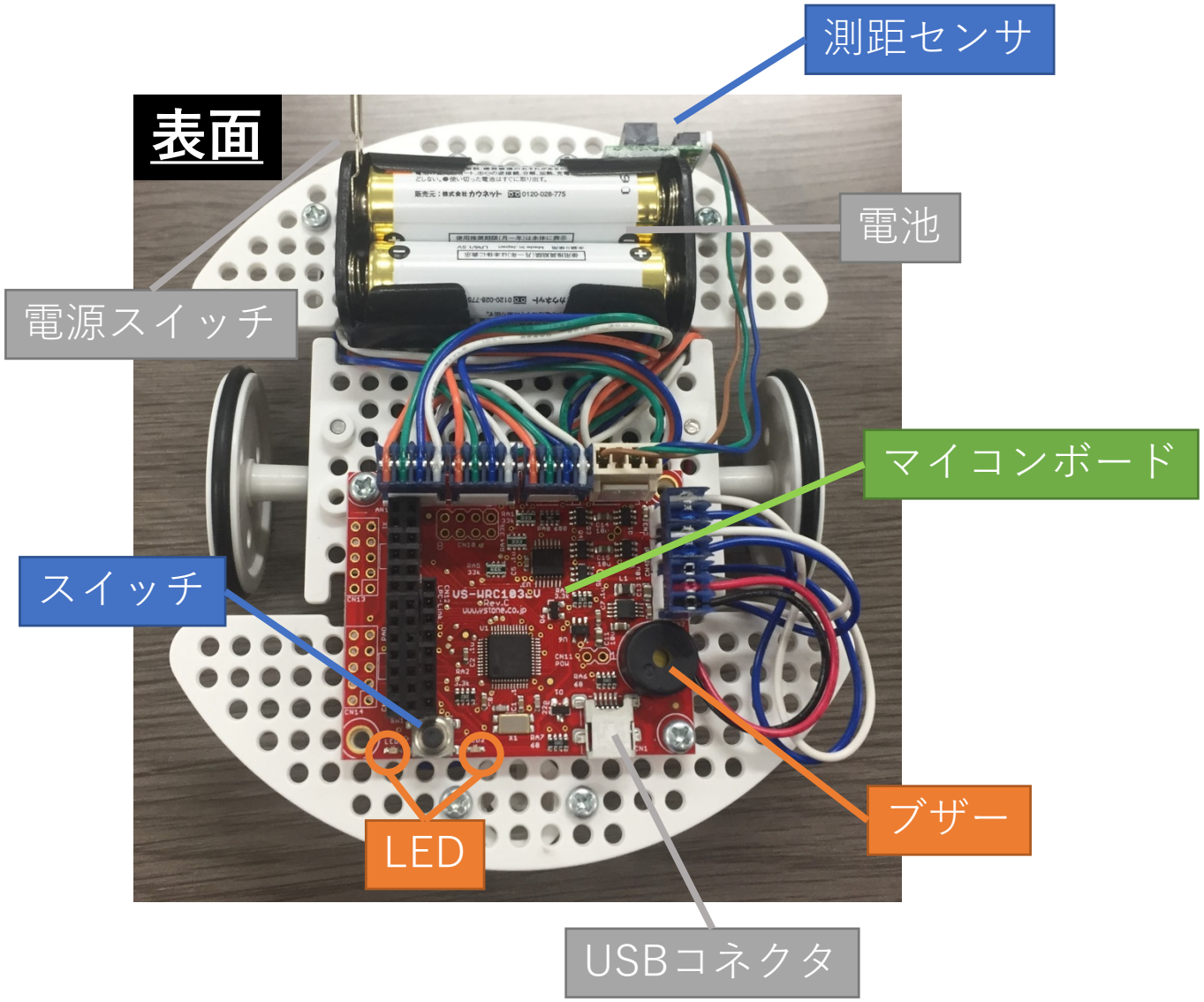
本実習の目標

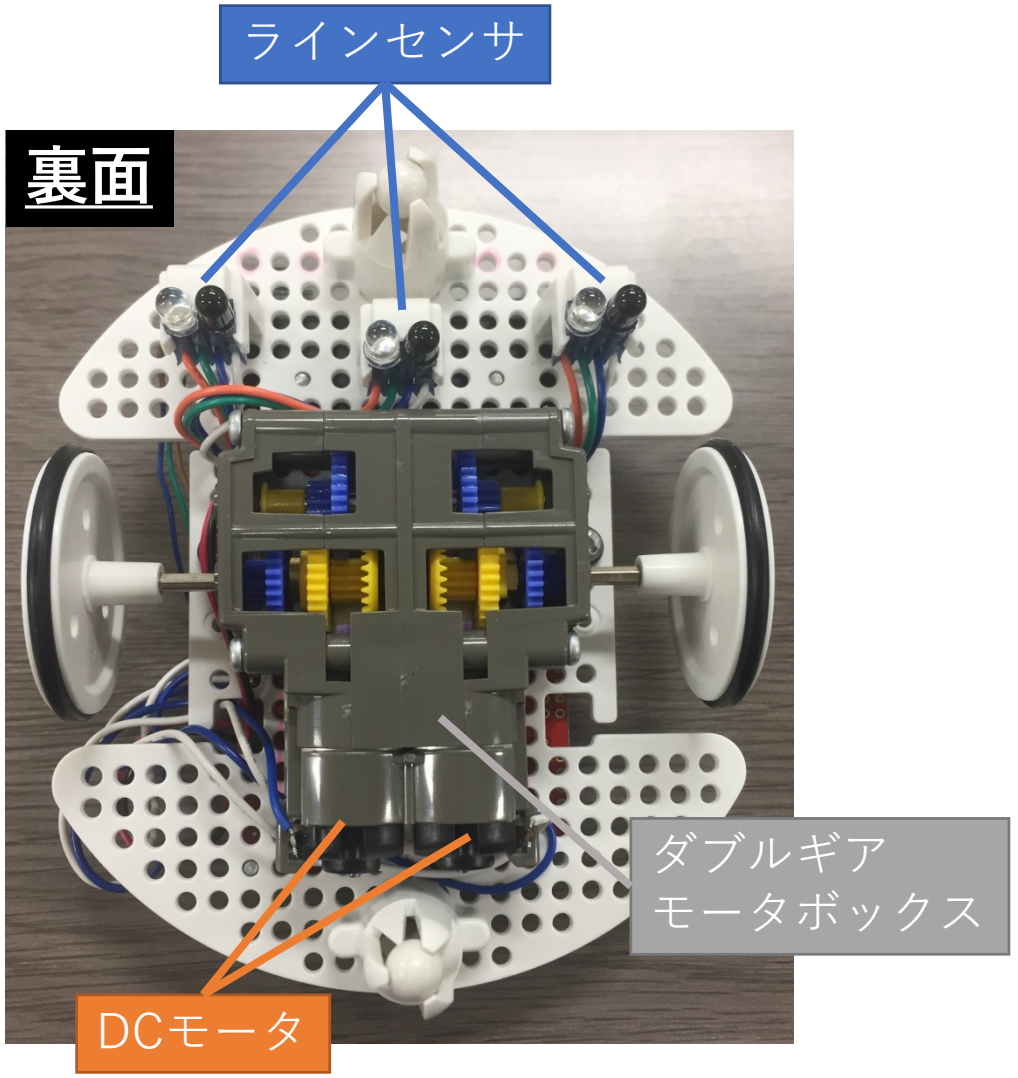
- 教材であるライントレースカー(LTC)の取扱に慣れること
- ビジュアルプログラミングツールを活用し、LTCの制御方法の基本を学ぶこと
- 組込みシステムが物理的要因と密接に関わる事を体感すること

教材紹介：PK-LTC

- PK-LTC：Polytech Center Kumamoto – Line Trace Carの略
- 今回使用する教材は、ヴイストン社製の「ビュートローバーARM」を独自拡張したものを使用します。
- 搭載されるマイコンボードは、VS-WRC103LVでARM Cortext-M3のLPC1343(NXP社製)を搭載しています。
- 開発環境には、NXP社の提供する統合開発環境LPCXPresso及びBeautoBuilder(ヴイストン社製)を使用します。

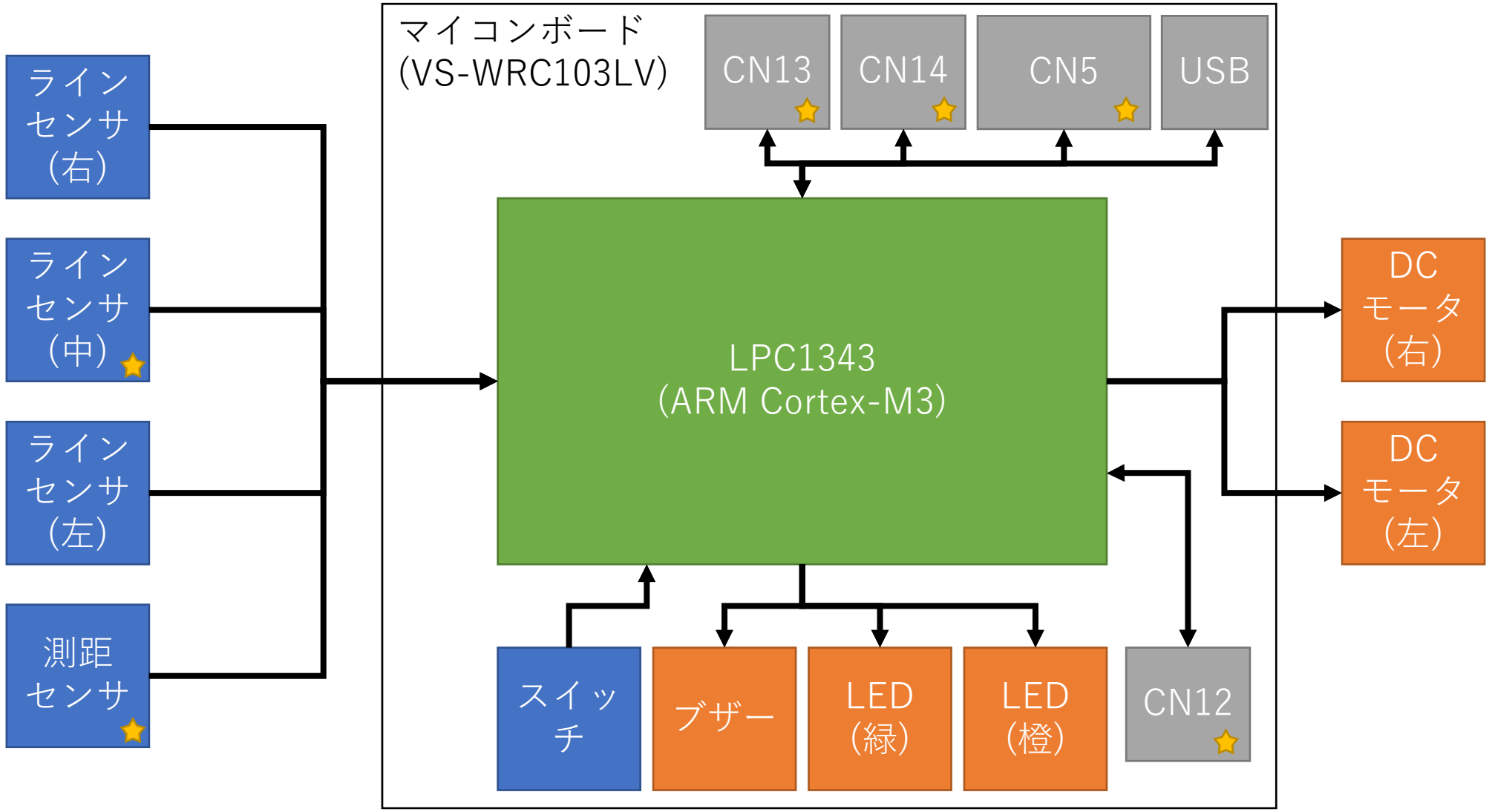
PK-LTCの仕様


表面
測距センサ
電池
電源スイッチ
マイコンボード
スイッチ
LED
ブザー
USBコネクタ


裏面
ラインセンサ
ダブルギア
モータボックス
DCモータ

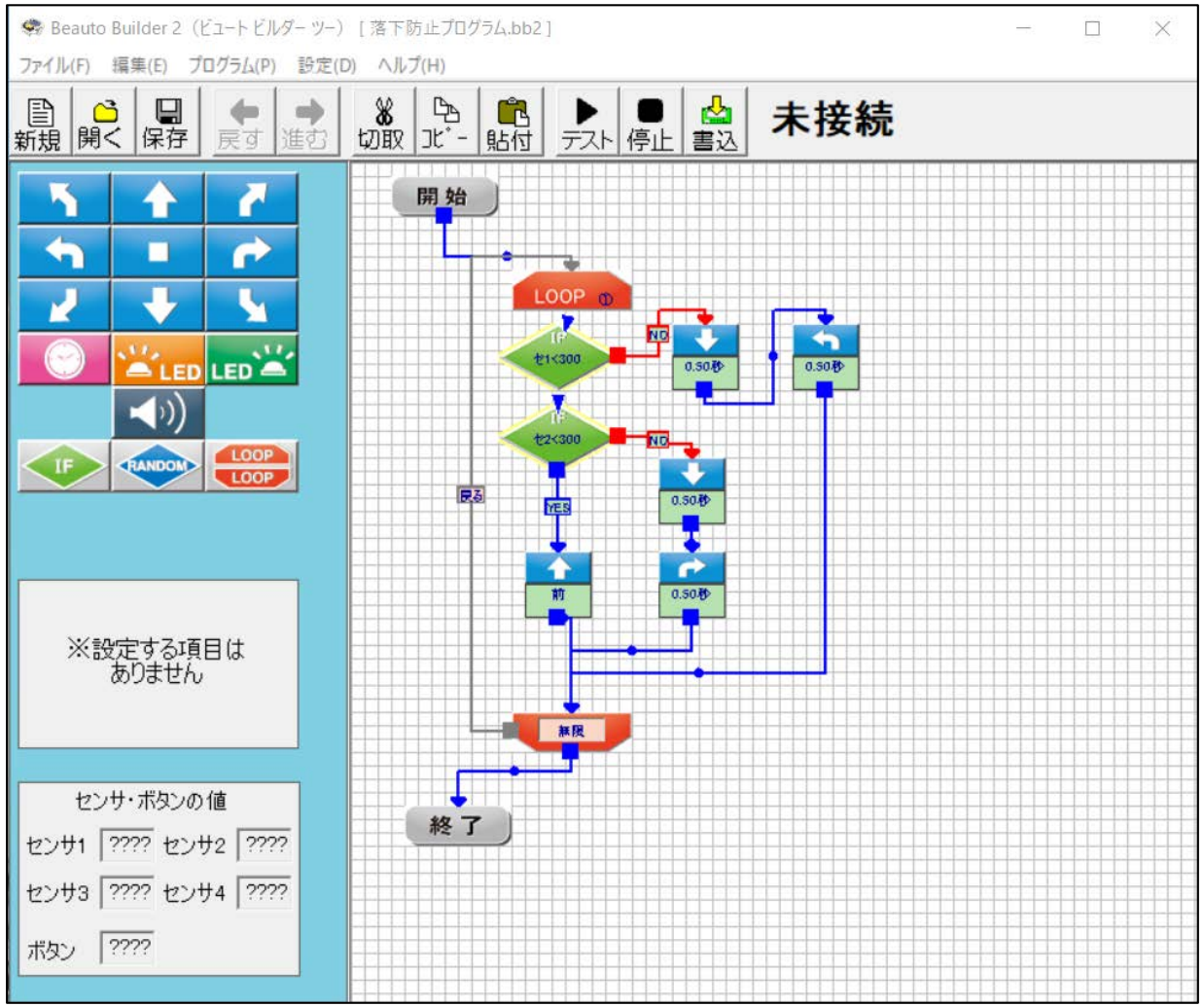
システム構成図

★ …独自拡張の要素

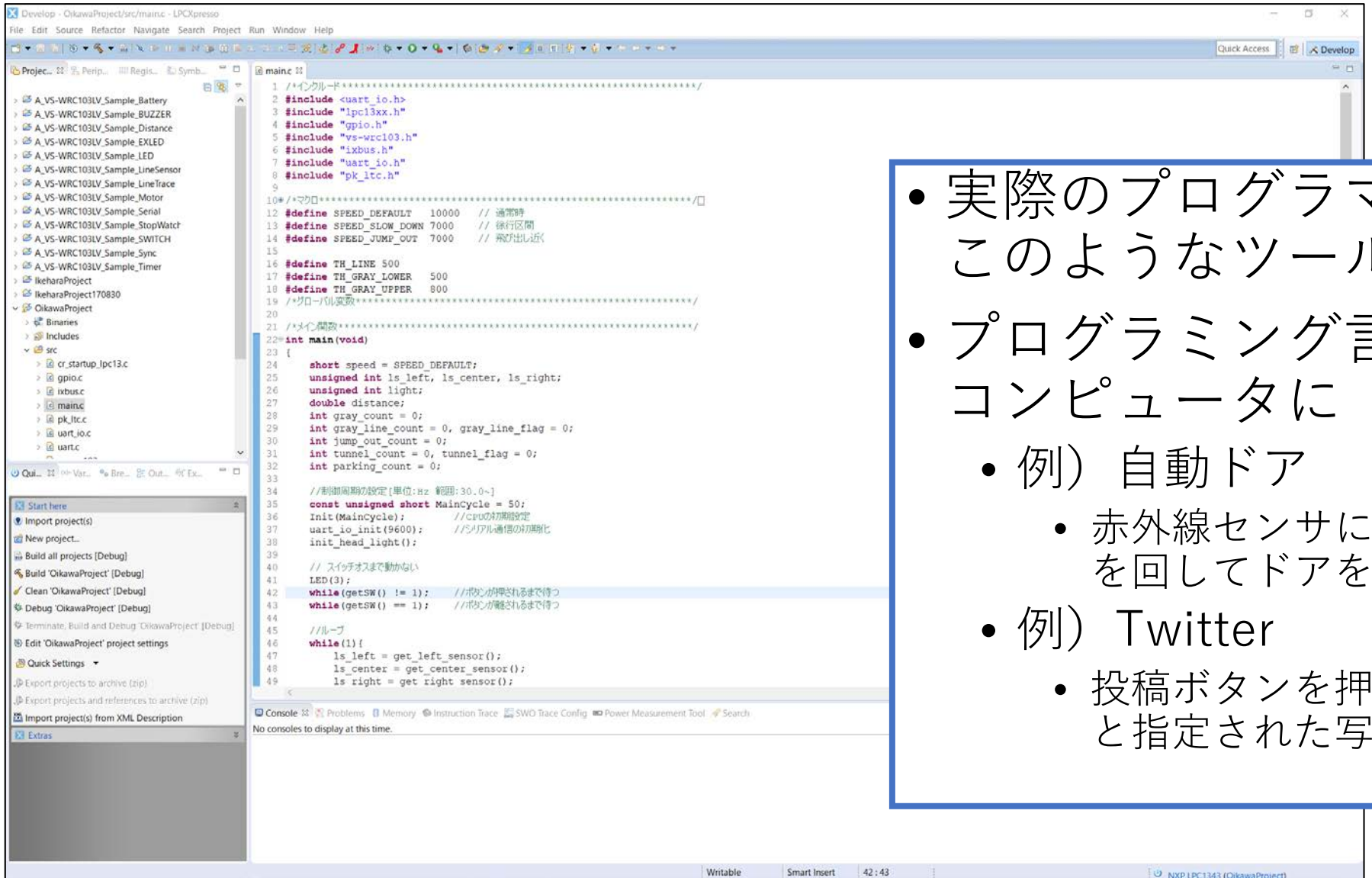


プログラムの開発環境の構築

- ビュートビルダー２取扱説明書「1-3 ソフトウェアのインストール」を参照。(P7)
- 取扱説明書に沿って
- インストーラ版を使用すること



ビジュアルプログラミングとは？



- 実際のプログラマが開発する際には、このようなツールを用いる
- プログラミング言語を駆使して、コンピュータに「動き」を指示する
  - 例）自動ドア
    - 赤外線センサに反応があったら、モータを回してドアを開ける
  - 例）Twitter
    - 投稿ボタンを押したら、入力された文章と指定された写真を投稿する

敷居が高い

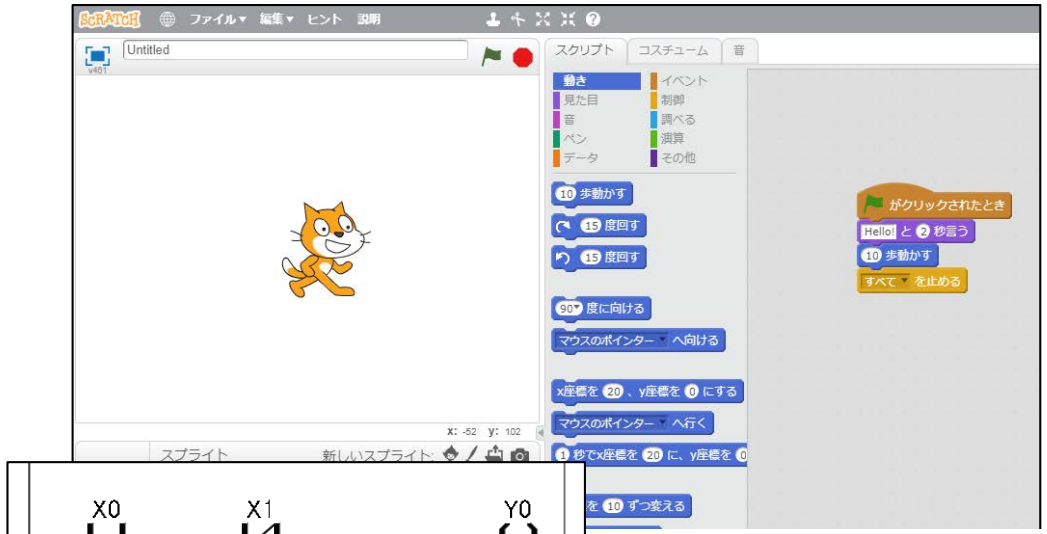
ビジュアルプログラミングとは？

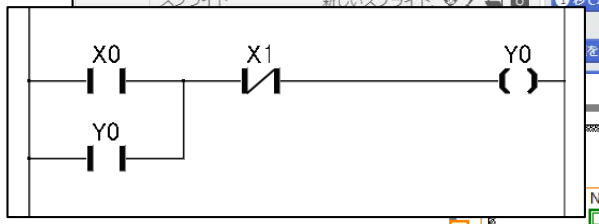
- 実務でビジュアルプログラミングを使用する場面はどれだけあるのか？
  - 教育用途　　：Scratch, プログラミン
  - 計測制御分野　：LabView,　ラダー
  - Web/IoT分野　：Node-Red

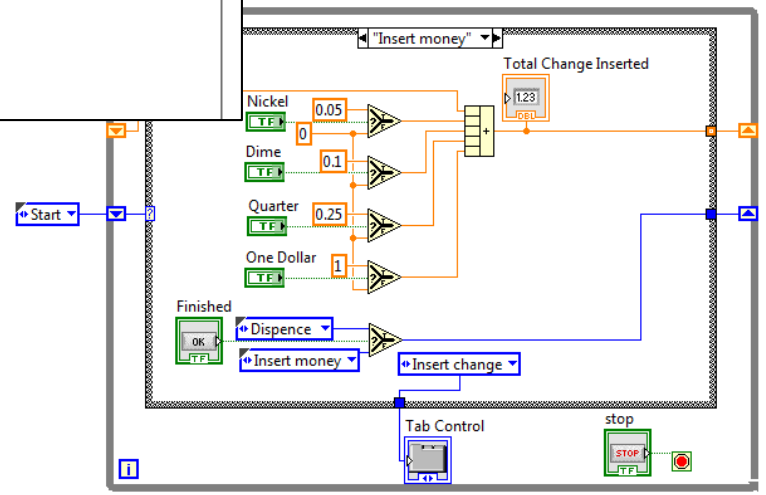
などなど

- 実際の開発現場の用途としては、特定の分野に限られる。
  - ビジュアルプログラミングは、その製品・分野に詳しい専門家が素早く実現・評価を行う為にある
  - 特定の分野向けに特化した作りになりやすく、結果的に高価なツールになりやすい

一部では特定用途に特化して活用されている



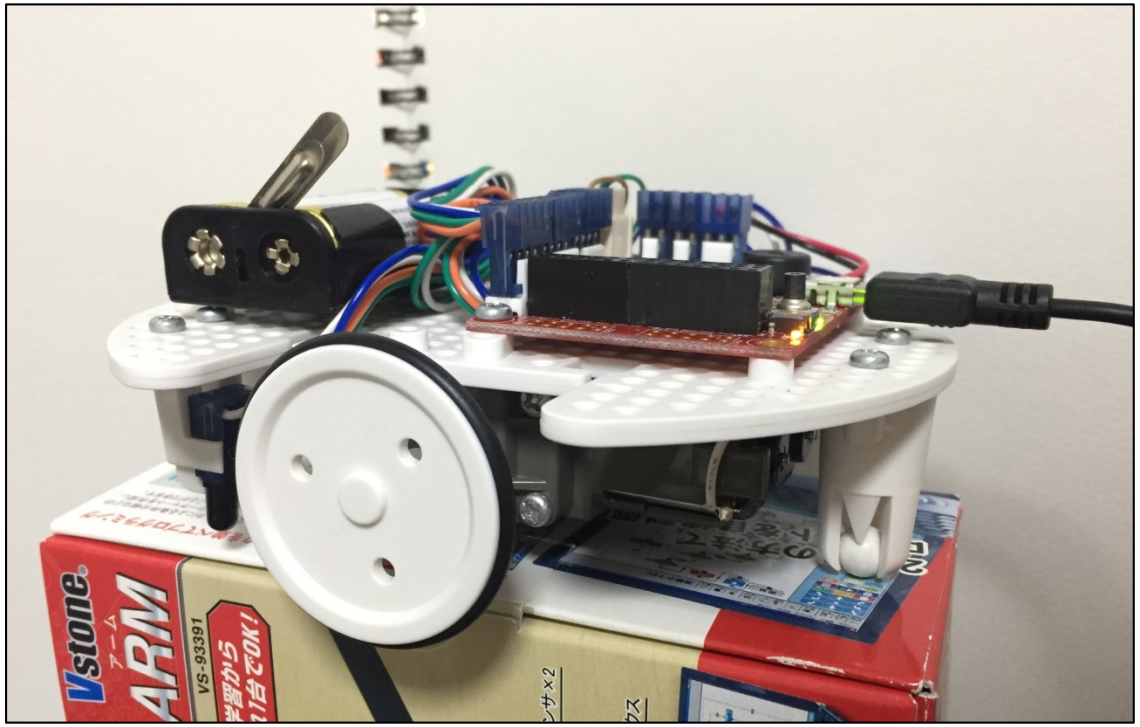




【画像引用元】
・https://en.wikipedia.org/wiki/State_machine_(LabVIEW_programming)
・https://ja.wikipedia.org/wiki/ラダー・ロジック

実習を始める前に

- デバッグ時は「箱の上」か「床」で実施すること
  - 机の上でデバッグを行うと誤ってLTCを落下させて破損する可能性があります。
  - そこで、「箱の上」もしくは「床」で動作させて下さい。
    - 箱の上：モータを空転させて動作確認
    - 床　　：広い場所で動作確認



実習①
センサ・アクチュエータの制御

ビジュアルプログラミング

- ビュートビルダー２取扱説明書「2. プログラムを作ってみよう」を参照。(P10～29)
- 実施内容
  2. プログラムを作ってみよう
  3. いろいろな機能を使ってみよう
     1. モータブロックの使い方
     2. ブザーとLEDの使い方
     3. センサを使った条件分岐

3-1 練習①　90度の右旋回にかかる時間

- 自分の作成したプログラムの旋回時間　　「　　秒」
- 隣の人の作成したプログラムの旋回時間　　「　　秒」

- 双方の旋回時間に差異はでましたか？　→　はい　・　いいえ
- 差異が出た場合、その理由を考えて書き出しましょう？

同じプログラムでも動作が変わる

- 組込み系のシステムでは、同じ仕様のハードウェアに全く同じプログラムを書き込んでも挙動が変わる事がある
- なぜか？
  - センサやモータなどに個体差がある
  - メカ的な構造にも、組み付け精度など起因の差が出る
  - バッテリ駆動の場合、バッテリの残量(電圧)の差
- PCやスマホ上だけで動作するアプリのような、「論理空間(デジタル空間)」で完結するアプリとは異なり、「物理空間」を相手にする組込み・制御系のシステムならではの特徴
- このように機体ごとの「個体差」を考慮したプログラムを記述する必要がある

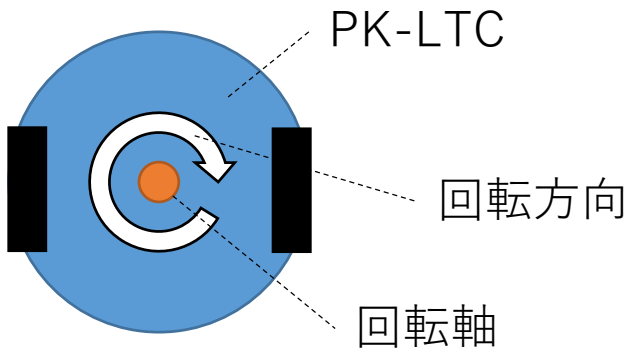
【練習①】モータブロック

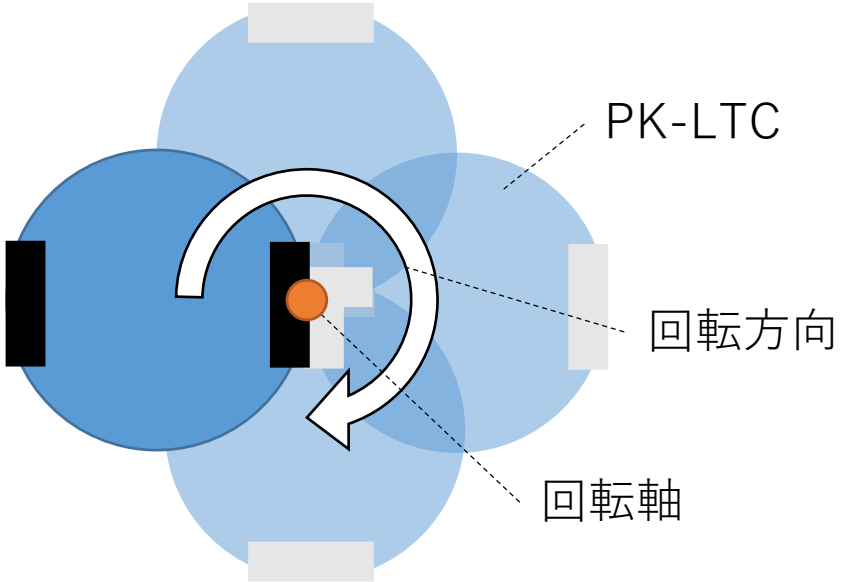
P21まで終わったら…

- 練習①-2　：180度ターン
  - LTCを1.5sec直進後、180度ターンさせ元の位置まで戻る。戻ったら再度180度ターンして最初の向きに戻る。

- 練習①-3　：その場で右旋回
  - LTCをその場でグルグル右旋回させる

- 練習①-4　：右タイヤを軸に右回転
  - LTCの右タイヤを軸に右方向へ回転させる





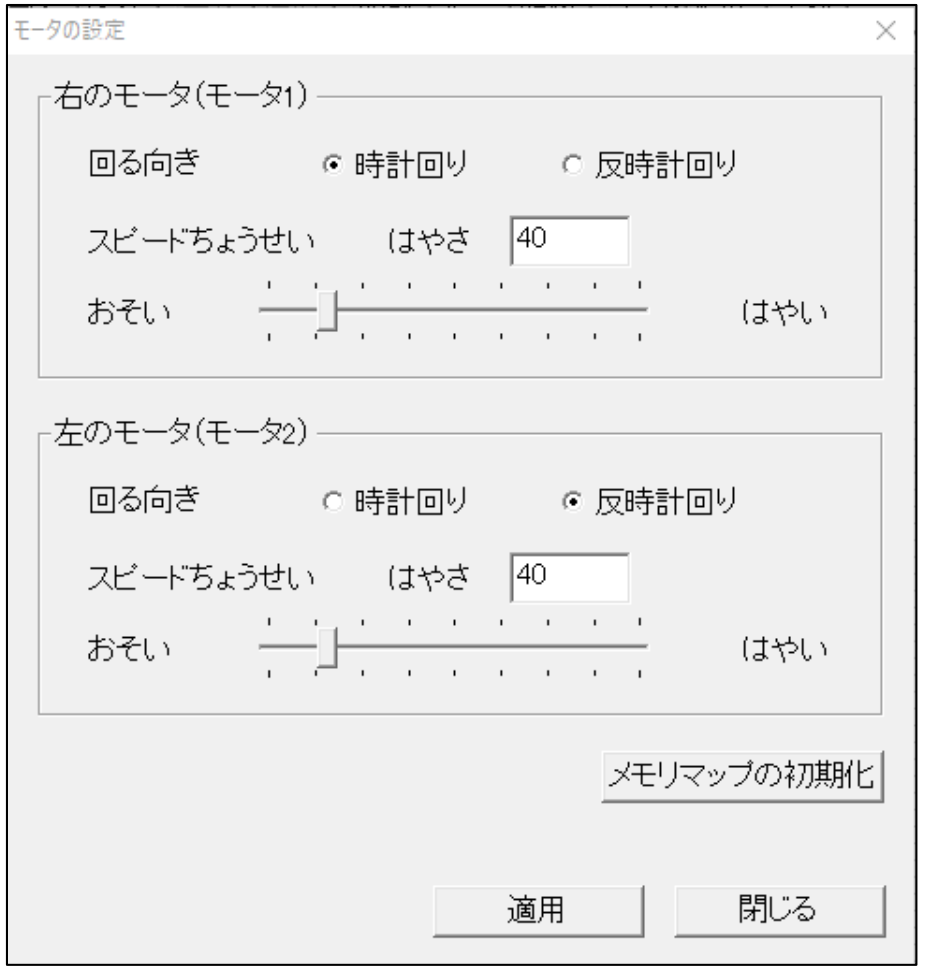
【発展】まっすぐ走らせよう

- 直進時に曲がる事象を防ぐ・補正する

- そもそも、なぜ曲がるのか？
  →モータごとに持つ出力の差や組み立て精度の違いなど。これも1つの「個体差」

- モータの出力パラメータを変更する
  - [設定]>[モータの設定]から実施
  - [スピードちょうせい]の[はやさ]を任意の値に設定する
    - 初期値：80
    - はやさは80以上に設定しないこと

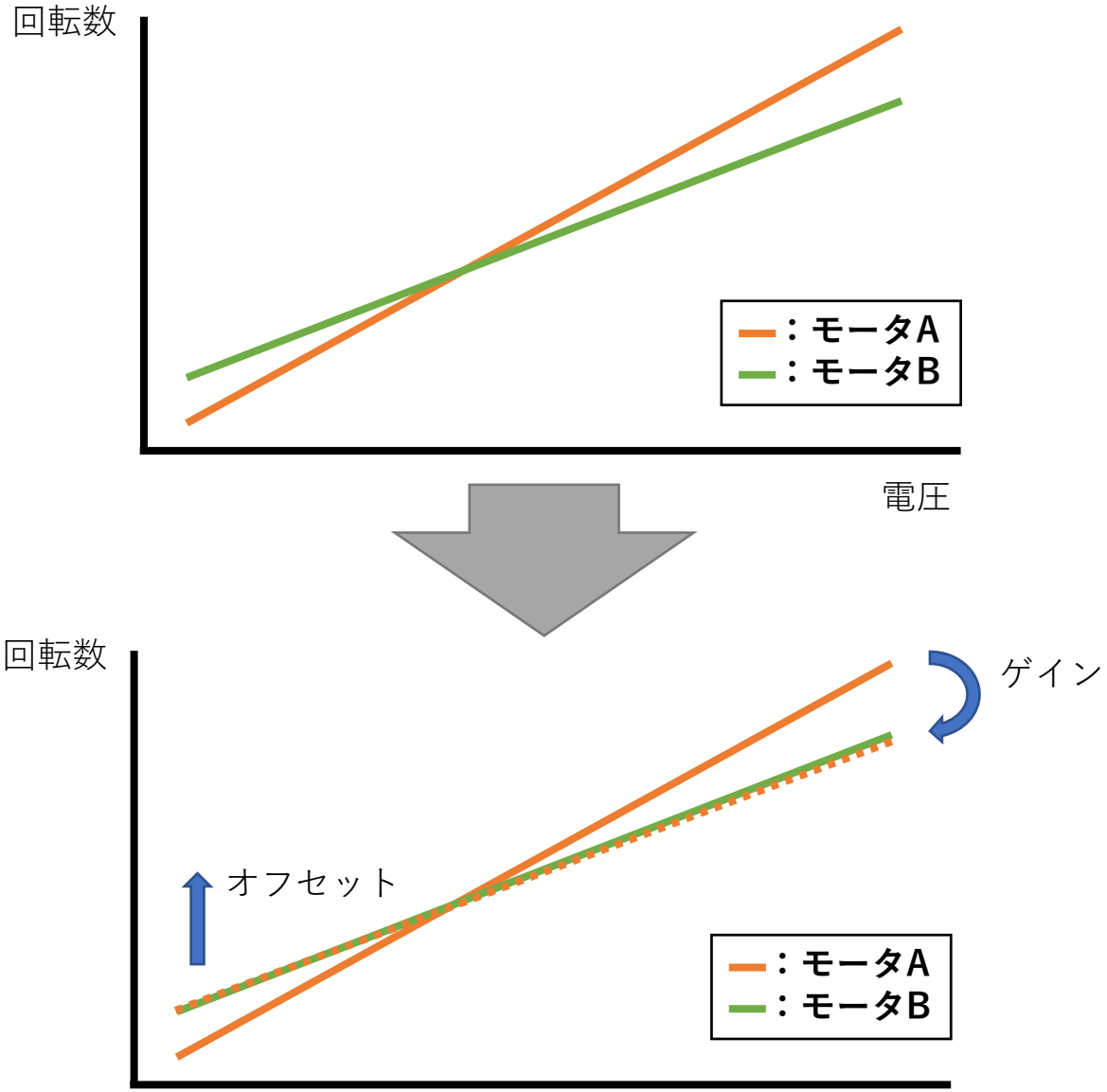
- 調整および動作検証には、初めに作成した「前進・後進」のプログラムを使用する



後進(バック)時の補正が効かない場合

- ビュートビルダーからは、モータブロック全体に対する出力調整しか出来ない
- 構造的な歪み等に起因する場合、前進と後進では出力調整の値が異なる事がある。
  - よって、前進の時の調整パラメータと後進の時の調整パラメータで分ける事が理想。
  - C言語による開発時に挑戦してみて下さい。

[発展]一次関数を用いた出力調整の考え方



- そもそも、低速時と高速時で同じ出力差が出るとは限らない
  - 左右のモータの出力は、低速時は右＞左でも、高速時には右＜左と変化する可能性も
- このような場合、複数の箇所で計測を行い、各箇所での差分からゲインおよびオフセット算出し、設定する事になる
  - ゲイン　　：傾きの調整
  - オフセット　：切片の調整

← **モータAをモータBの出力特性に近づけるように補正する時のイメージ**

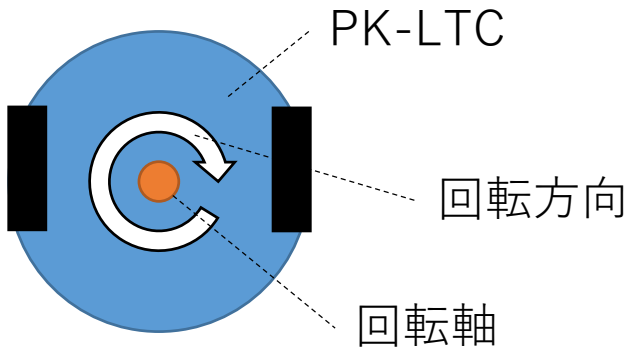
$y = ax+b$

(y:回転数　x:電圧　a:ゲイン　b:オフセット)

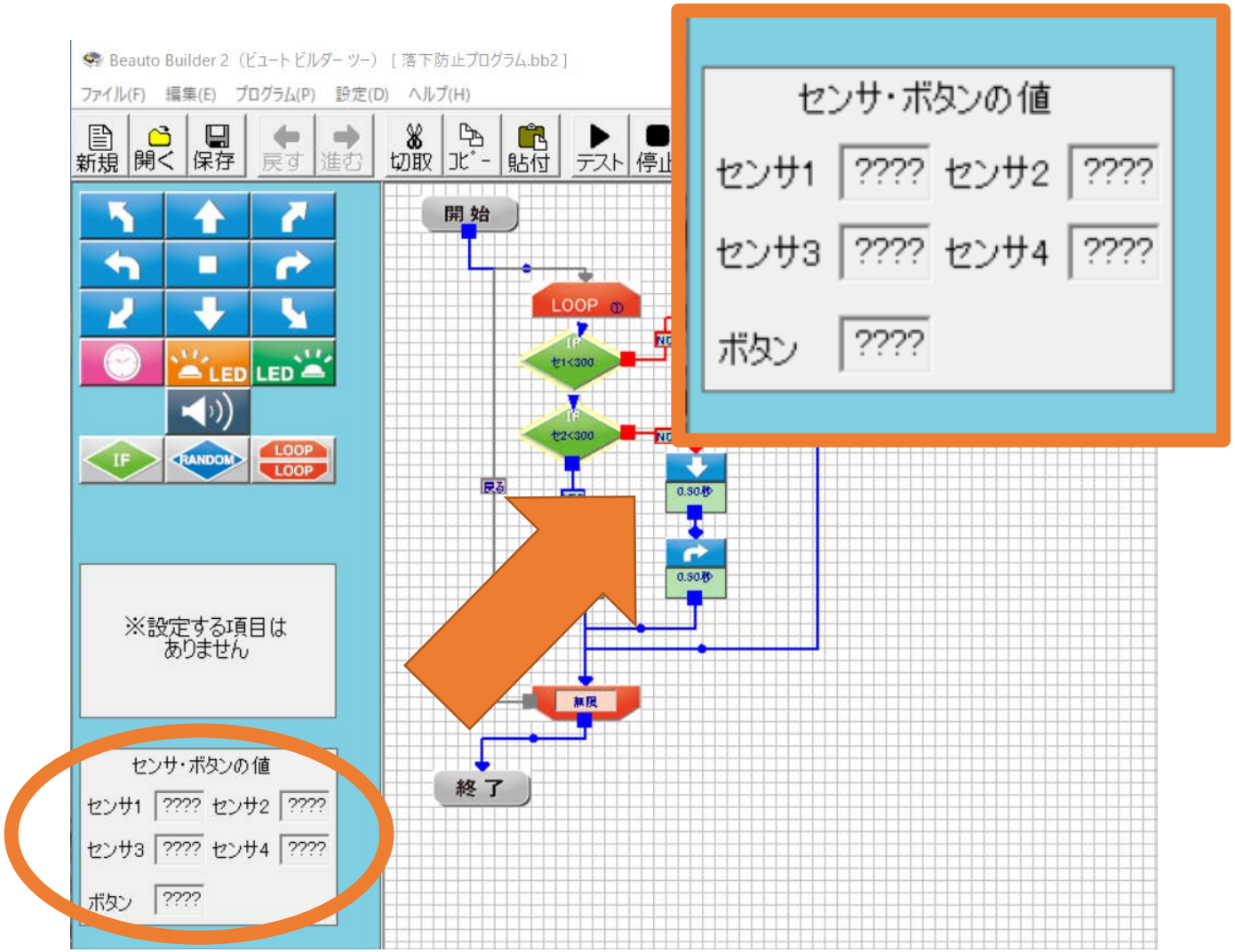
【練習②】ブザーとLED

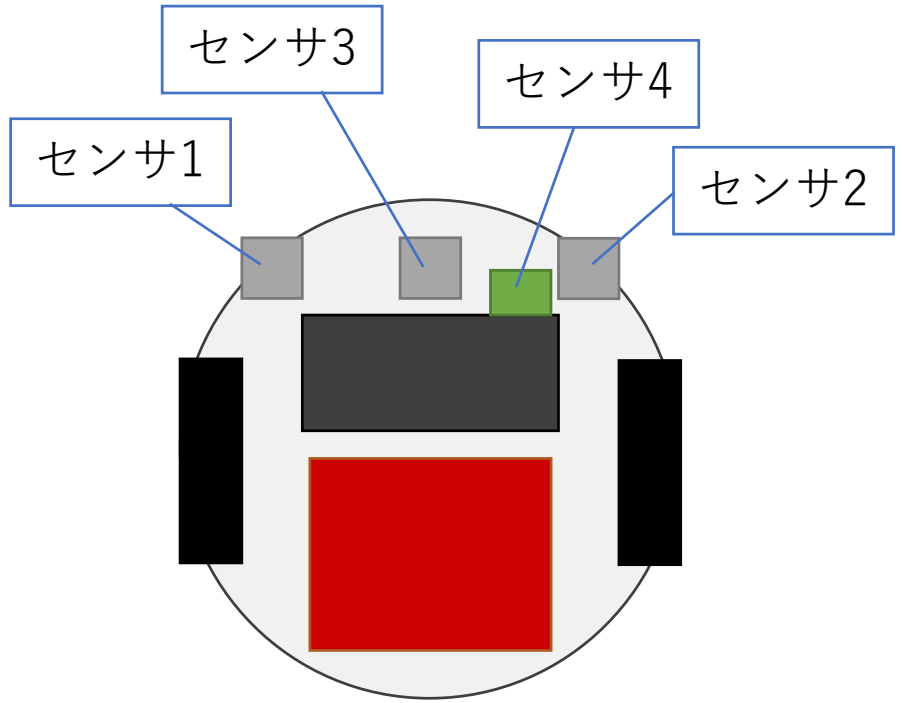
P25まで終わったら…

- 練習②-2　：右旋回しながらLED点滅
  - 1回転したら緑LEDを点灯、もう1回転したら緑LEDを消灯。これを繰り返す。
- 練習②-3　：右旋回しながらブザー鳴らす
  - 1回転したら「ド」の音を鳴らす。もう1回転したら音を消す。これを繰り返す。
- 練習②-4　：左旋回しながらLED点滅
  - 緑LEDを１秒間の点灯の後、１秒間の消灯。これを繰り返す。
  - 「ウェイトブロック」の活用推奨。



センサのチートシート



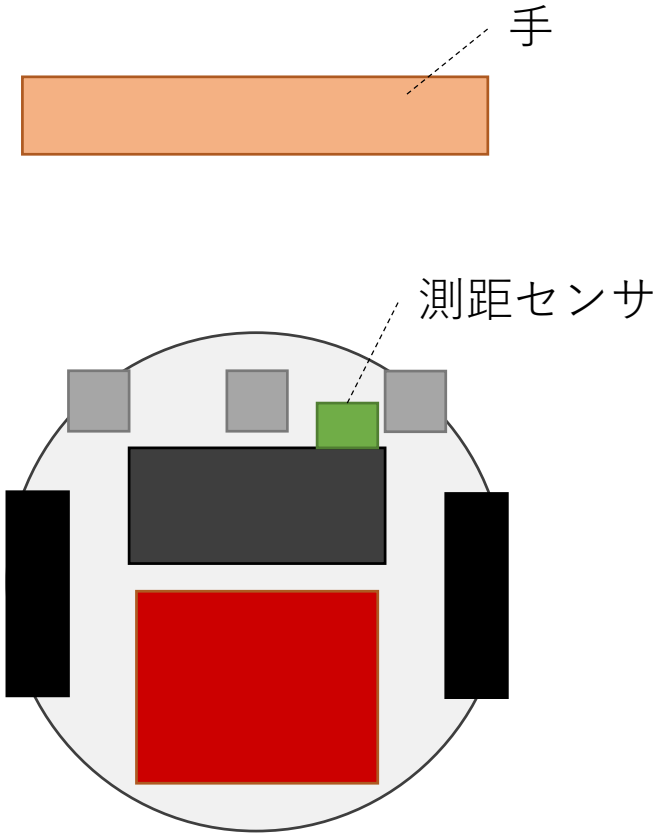


PK-LTC 上面図

ビュートビルダー	PK-LTC
センサ1	ラインセンサ(左)
センサ2	ラインセンサ(右)
センサ3	ラインセンサ(中)
センサ4	測距センサ

【練習③】ブザーとLED

P29まで終わったら…



- 練習③-2　：PK-LTCの前に手をかざし、測距センサの値を計測する
  - 手をかざした時のセンサ値　：「＿＿＿＿＿＿」
  - 手を離した時のセンサ値　　：「＿＿＿＿＿＿」

- 練習②-3　：PK-LTCの前に手をかざしたら、ブザーを鳴らす。手を離したらブザーを止めるプログラムを作成する。

組込み分野のアプリ・システムの入出力

- PCやスマホで動くアプリにおける入出力機器は…
  - 入力：マウス、＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿
  - 出力：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿
- 今回使用したのライントレースカー(組込みシステム)における入出力機器は…
  - 入力：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿
  - 出力：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

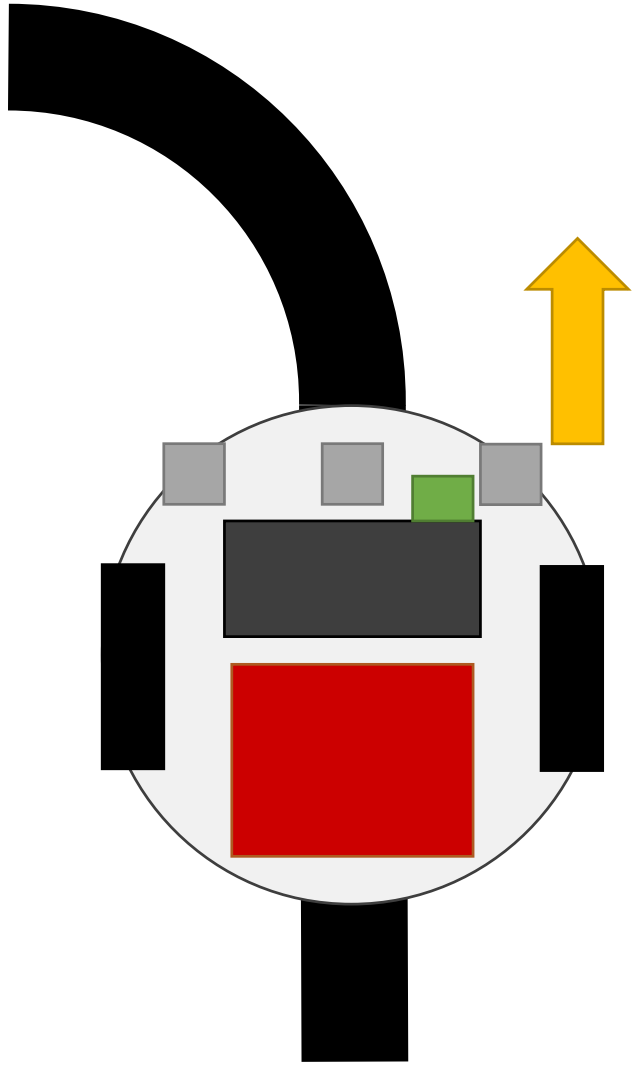
- 光や音、位置などの物理量を電気信号に変える様々な電子部品を「　　」と呼ぶ
- 電気信号を物理量に変える様々な電子部品を「　　」と呼ぶ
- 組込みシステムは、「　　」や「　　」をプログラムから制御して目的の動作・振る舞いを実現する

実習② ライントレース

ラインとレースに挑戦①

- ラインセンサを用いて線の上を走る
- 今回は中央のラインセンサは使わない

- ラインとレースの基本アルゴリズム
  - 左のラインセンサがラインに反応した場合
    - 舵を右に切りながら走行する
  - 右のラインセンサがラインに反応した場合
    - 舵を左に切りながら走行する
  - どちらのセンサを反応しない場合
    - 直進する
  - 以上をひたすら繰り返す



ラインントレースに挑戦②

1. ライン(白・黒)を識別しよう
   - ラインセンサが黒のライン上にある時とそれ以外の場所(白の上)にある時でセンサ値を計測してみましょう！
2. ラインの識別方法の検討がついたら…
   - ラインセンサの値に合わせてLTCの動作(振舞い)を定めよう！
3. Let’s Programing!!

場所	ラインセンサ(左)の値	ラインセンサ(右)の値
ライン上(黒)にセンサがある時		
それ以外(白)にセンサがある時		

ラインセンサ(左)	ラインセンサ(右)	LTCの動作
白	白	
黒	白	
白	黒	

【実験】日の当たる場所で動作させる

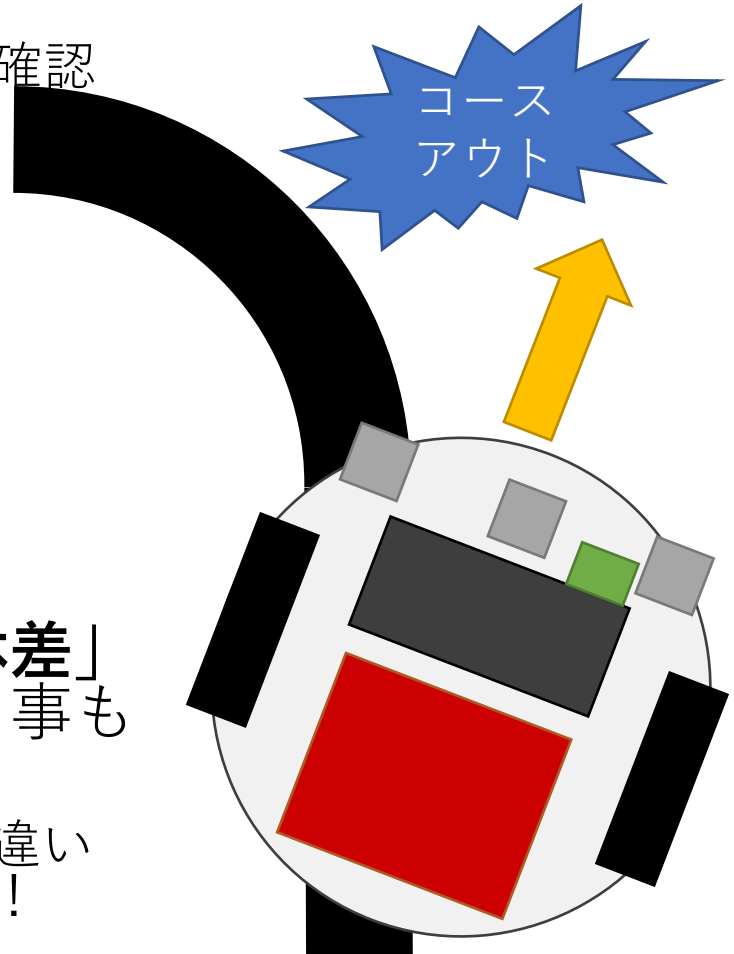
ライントレース制御ができたら…

- カーテンやブラインドを開放して、日光があたる場所でライントレースをさせてみて下さい。
- この時、ライントレースカーの挙動はどのようになるでしょうか？
  - 確認した挙動
  - この挙動に至った原因考察

「外乱」による挙動の変化

- 太陽光をあてると、ライントレースカーはコースを外れてしまう。
- これは、太陽光が光センサに影響を与えている為です
  - ライントレースカーにUSBケーブルを繋いでラインセンサの値を確認してみて下さい
  - 先ほどまでのセンサ値と比べてみて下さい

場所	ラインセンサ(左) の値	ラインセンサ(右) の値
ライン上(黒)に センサがある時		
それ以外(白)に センサがある時		



- 組込みシステムに影響を及ぼす物理要因は、機体の「**個体差**」だけではなく、外界からのノイズ「**外乱**」などが影響する事もあります。
  - デバッグ・テストを行った時の動作環境と、実際の動作環境との違いにより、振る舞いに影響がでることもあります。注意しましょう！

【応用】手をかざしたらLTCを停止

- 測距センサで前方に障害物を検知したらPK-LTCを停止させるプログラムを作成してください。

手	測距センサ
手を かざす	
手を離す	

ライン センサ(左)	ライン センサ(右)	測距センサ	LTCの動作
白	白	－	
黒	白	－	
白	黒	－	
－	－	手をかざす	
－	－	手を離す	

【発展】センサ１つでライントレース

- ラインセンサ(中)だけを用いてライントレースするプログラムを実現しよう。

場所	ラインセンサ(中)
ライン上(黒)	
それ以外(白)	

ラインセンサ(中)	LTCの動作
白	
黒	

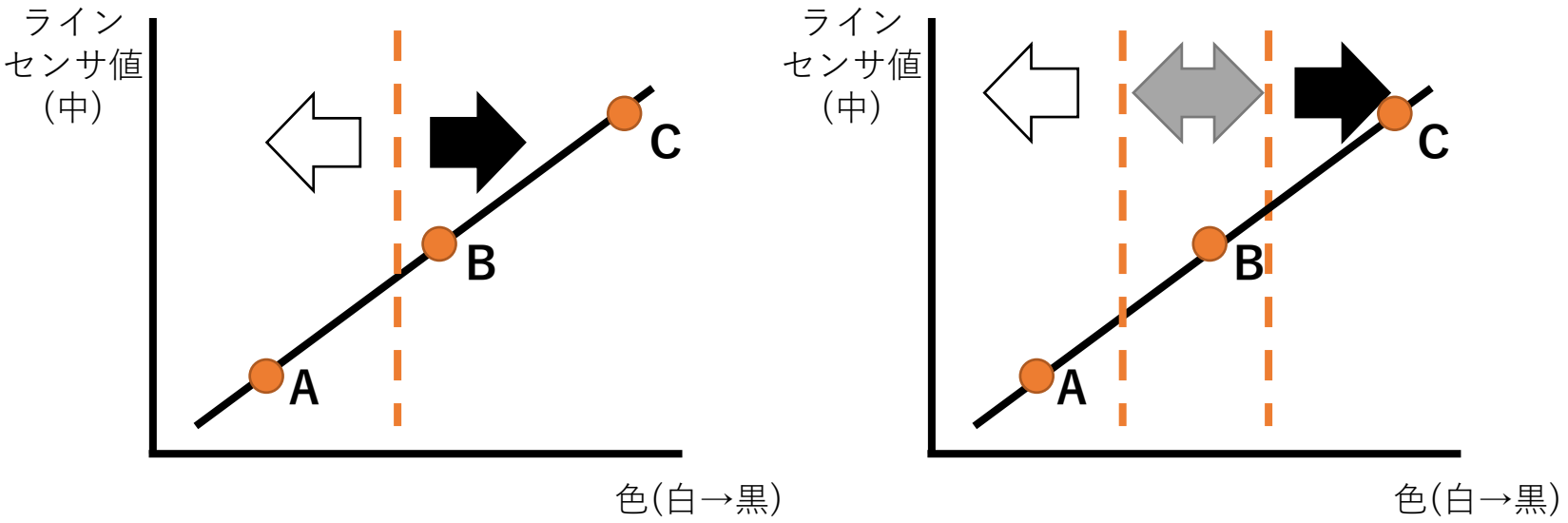
【発展②】センサ１つでライントレース＋手をかざしたらLTC停止

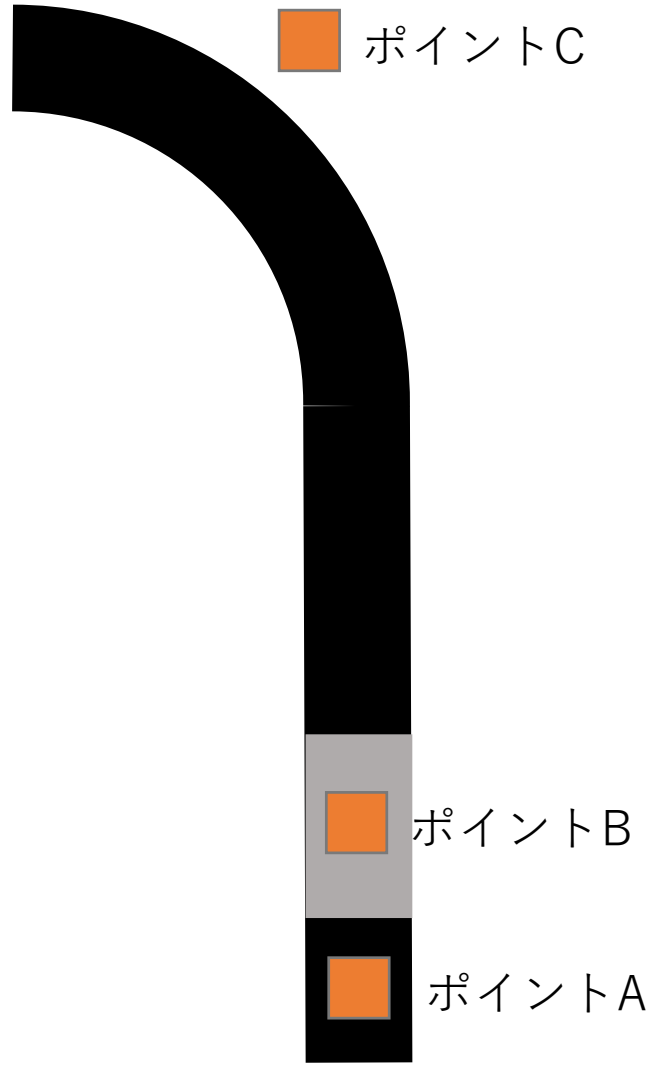
- 距離センサを用いてライントレースするプログラムに「手をかざしたらLTC停止」の機能を追加しよう

- この課題も終わってしまったら…？
  - ラインセンサ（左・右）をもちいたライントレースのプログラムを改変し、「ラインセンサ(中)でグレーラインを検出したらブザーを鳴らす」機能を追加しよう
  - （この課題は意図した通りに動きません。グレーラインの誤検知が発生するはずです。誤検知を確認出来たら、なぜ誤検知したのか？原因を考えてみましょう）

グレーライン検出の考え方

- グレーライン検出の考え方

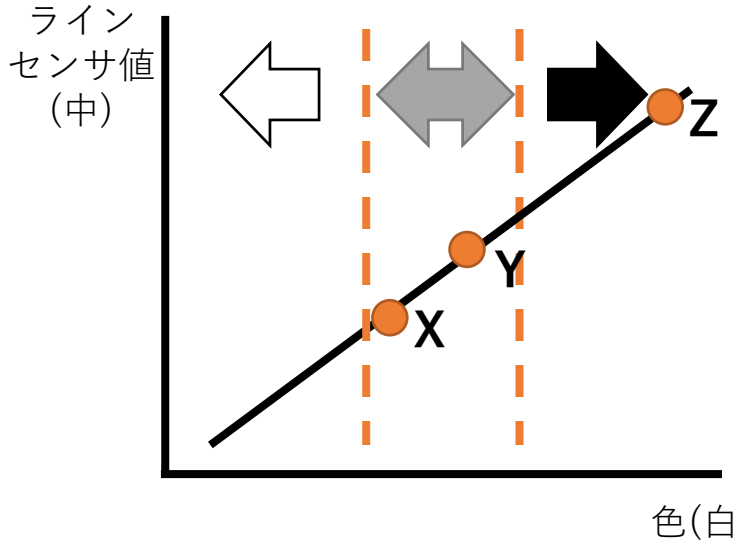


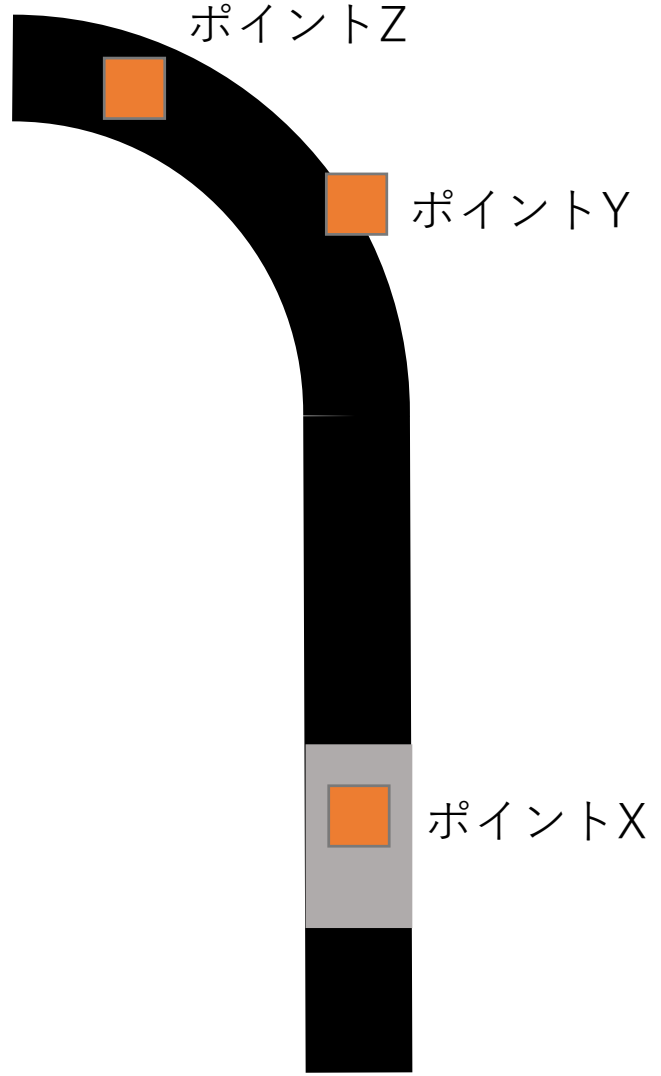


- 白・黒の識別は、**ある値**よりセンサ値が…
  - ある値未満　　：白
  - ある値以上　　：黒　　と判断する
- ある値　＝　**閾値(スレッショルド)**　と呼ぶ

グレーラインの誤検知の理由

- どんな場面で誤検知しましたか？
  - 真ん中のラインセンサが黒ラインと白の境界あたりに位置した時ではありませんでしたか？





- 黒と白の間にラインセンサが位置すると…
  - 反射する光量がグレーと同じぐらいの値を示すことがある
  - 物理的な要因がついて回る為に発生する問題
- 対策方法については、また別途。

まとめ

- マイコンのプログラムも基本は「　　」「　　」「　　」
- 従来学んだC言語のプログラムのフローとは異なり、「無限ループ」を作成し、その中で「　　」や「　　」を制御する処理を記述する。
- ロボットを制御する時には、ロボット(機体)ごとに「　　」や動作させる環境からの「　　」などがある為、それらの要因を考慮したプログラムを設計・開発する必要がある

関連リンクほか

- ビュートローバーARM（ヴイストン社）
  - https://www.vstone.co.jp/products/beauto_rover/index.html
  - 本テキストに関する問い合わせについて、ヴイストン社は受け付けておりませんのでご注意ください。
- 画像引用元
  - いらすとや (https://www.irasutoya.com)

指導員向け資料

3-1 練習①　90度の右旋回にかかる時間

- 自分の作成したプログラムの旋回時間　　「　　秒」
- 隣の人の作成したプログラムの旋回時間　　「　　秒」

- 双方の旋回時間に差異はでましたか？　→　はい　・　いいえ
- 差異が出た場合、その理由を考えて書き出しましょう？

バッテリの残量の差異、モータの出力の差、組立時の誤差

組込み分野のアプリ・システムの入出力

- PCやスマホで動くアプリにおける入出力機器は…
  - 入力：マウス、キーボード、マイク、タッチパネルなど
  - 出力：ディスプレイ、スピーカーなど
- 今回使用したのライントレースカー(組込みシステム)における入出力機器は…
  - 入力：光センサ、ボタン、距離センサ
  - 出力：LED、モータ、ブザー

- 光や音、位置などの物理量を電気信号に変える様々な電子部品を「　センサ　」と呼ぶ
- 電気信号を物理量に変える様々な電子部品を「アクチュエータ」と呼ぶ
- 組込みシステムは、「　センサ　」や「　アクチュエータ　」をプログラムから制御して目的の動作・振る舞いを実現する

ラインとレースに挑戦②

1. ライン(白・黒)を識別しよう
   - ラインセンサが黒のライン上にある時とそれ以外の場所(白の上)にある時でセンサ値を計測してみましょう！
2. ラインの識別方法の検討がついたら…
   - ラインセンサの値に合わせてLTCの動作(振舞い)を定めよう！
3. Let's Programing!!

場所	ラインセンサ(左)の値	ラインセンサ(右)の値
ライン上(黒)にセンサがある時		
それ以外(白)にセンサがある時		

ラインセンサ(左)	ラインセンサ(右)	LTCの動作
白	白	直進
黒	白	左旋回
白	黒	右旋回

【実験】日の当たる場所で動作させる

ライントレース制御ができたら…

- カーテンやブラインドを開放して、日光があたる場所でライントレースをさせてみて下さい。
- この時、ライントレースカーの挙動はどのようになるでしょうか？
  - 確認した挙動

黒ラインを外れてコースアウトする

  - この挙動に至った原因考察

ブラインドを開けたことで太陽光が入り、ラインセンサで黒・白を読んだ時の値が
ブラインドを開ける前と大きく変化した為

【応用】手をかざしたらLTCを停止

- 測距センサで前方に障害物を検知したらPK-LTCを停止させるプログラムを作成してください。

手	測距センサ
手を かざす	
手を離す	

ライン センサ(左)	ライン センサ(右)	測距センサ	LTCの動作
白	白	–	直進
黒	白	–	左旋回
白	黒	–	右旋回
–	–	手をかざす	停止
–	–	手を離す	何もしない (再度動き出す)

【発展】センサ１つでライントレース

- ラインセンサ(中)だけを用いてライントレースするプログラムを実現しよう。

場所	ラインセンサ(中)
ライン上(黒)	
それ以外(白)	

ラインセンサ(中)	LTCの動作
白	左斜め前進
黒	右斜め前進

【発展②】センサ１つでライントレース＋手をかざしたらLTC停止

- 距離センサを用いてライントレースするプログラムに「手をかざしたらLTC停止」の機能を追加しよう

- この課題も終わってしまったら…？
  - ラインセンサ（左・右）をもちいたライントレースのプログラムを改変し、「ラインセンサ(中)でグレーラインを検出したらブザーを鳴らす」機能を追加しよう
  - （この課題は意図した通りに動きません。グレーラインの誤検知が発生するはずです。誤検知を確認出来たら、なぜ誤検知したのか？原因を考えてみましょう）

黒と白の境目をグレーを誤検知している

まとめ

- マイコンのプログラムも基本は「逐次」「分岐」「繰り返し」

- 従来学んだC言語のプログラムのフローとは異なり、
  「無限ループ」を作成し、その中で「センサ」や
  「アクチュエータ」を制御する処理を記述する。

- ロボットを制御する時には、ロボット(機体)ごとに「個体差」や
  動作させる環境からの「外乱」などがある為、それらの要因を考
  慮したプログラムを設計・開発する必要がある