

鉄道模型を使った組込みソフトウェア教育 (中洲線プロジェクト)

九州大学大学院システム情報科学研究院
情報工学専攻
福田・中西研究室

演習のねらい

演習概要

- 内容：
 - ITRONボードを用いて鉄道模型の制御を行うソフトウェアを開発
- 受講生：
 - 研究室に配属された学部4回生 4名
- 受講生の既習事項
 - C言語(技量は個人差かなりあり)
 - 演習(3回生時): CMOSロジックICとFPGAを用いた論理回路設計
 - 演習(3回生時): FPGAを用いたプロセッサ設計
 - 演習(3回生時): オペレーティングシステム実装

演習のねらい

- 組込みに興味を持たせる。
 - 作っていて面白い題材。
 - 一人でできる小さな題材。
 - 前提知識をあまり要求しない題材。
 - 力量に応じて複雑度を変えられる題材。
- C言語の知識を確かなものにする。
- とにかくハードウェアの実物に触れる。
 - まずは半田付けから。
 - トラブル対策はハードウェアからソフトウェアまで。
- 組込みソフトウェア開発に慣れる。
 - クロス開発の体験
 - ハードウェア制御の体験 (I/Oポートへの直接アクセス)
 - オシロスコープ等計測機器の使用体験
 - 状態遷移設計の体験

なぜ鉄道模型なのか

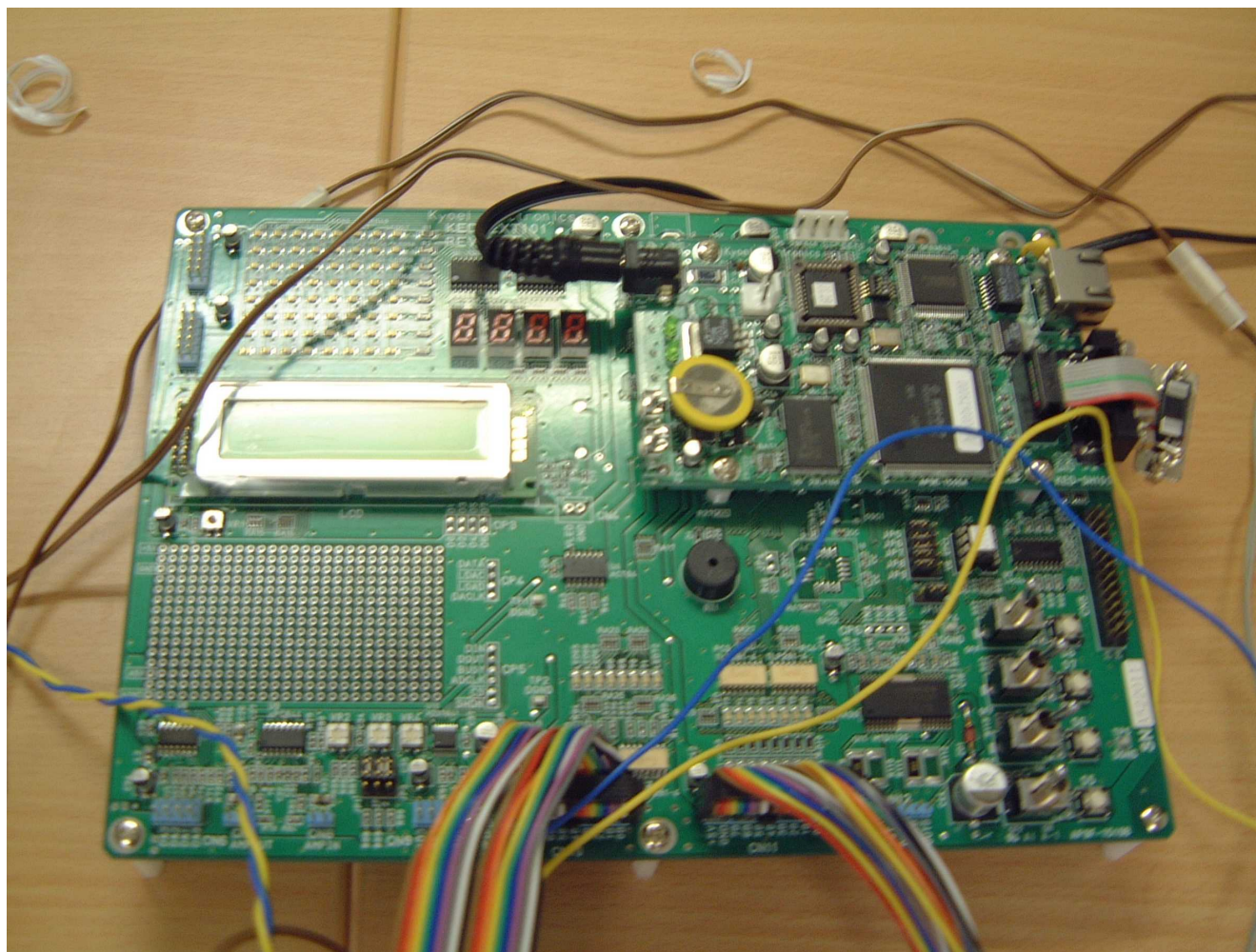
- 高度な前提知識を必要としないから .
- オープンで扱いやすいから .
- 現象を目で確認できるから .
- Variability を出しやすい , 複雑度を変えやすいから .
 - レイアウト
 - 電車の編成数
 - 電車の個体差 (モータ回転数 , 連結数 , 台数)
 - 豊富な制御対象 : 電車 , ポイント , ターンテーブル , 信号機 , 踏切 ...
 - センサの種類
 - デモシナリオ

演習内容

1日目	C言語文法座学 & 小演習
~	
3日目	
4日目	C言語文法座学 & 大演習(万年暦作成)
5日目	モータ駆動回路作成
6日目	組込みボード試用(サンプルプログラム使用)
7日目	電車模型デモプログラム実装(電車速度PWM制御)
8日目	電車模型デモプログラム実装(ポイント切替制御)
9日目	電車通過センサ回路作成
10日目	電車模型デモプログラム実装(一周:加速 最高速度 低速 停止)
11日目	電車模型デモプログラム実装(シナリオ運転)
~	
15日目	

ハードウェア概要

ITRONボード(1)

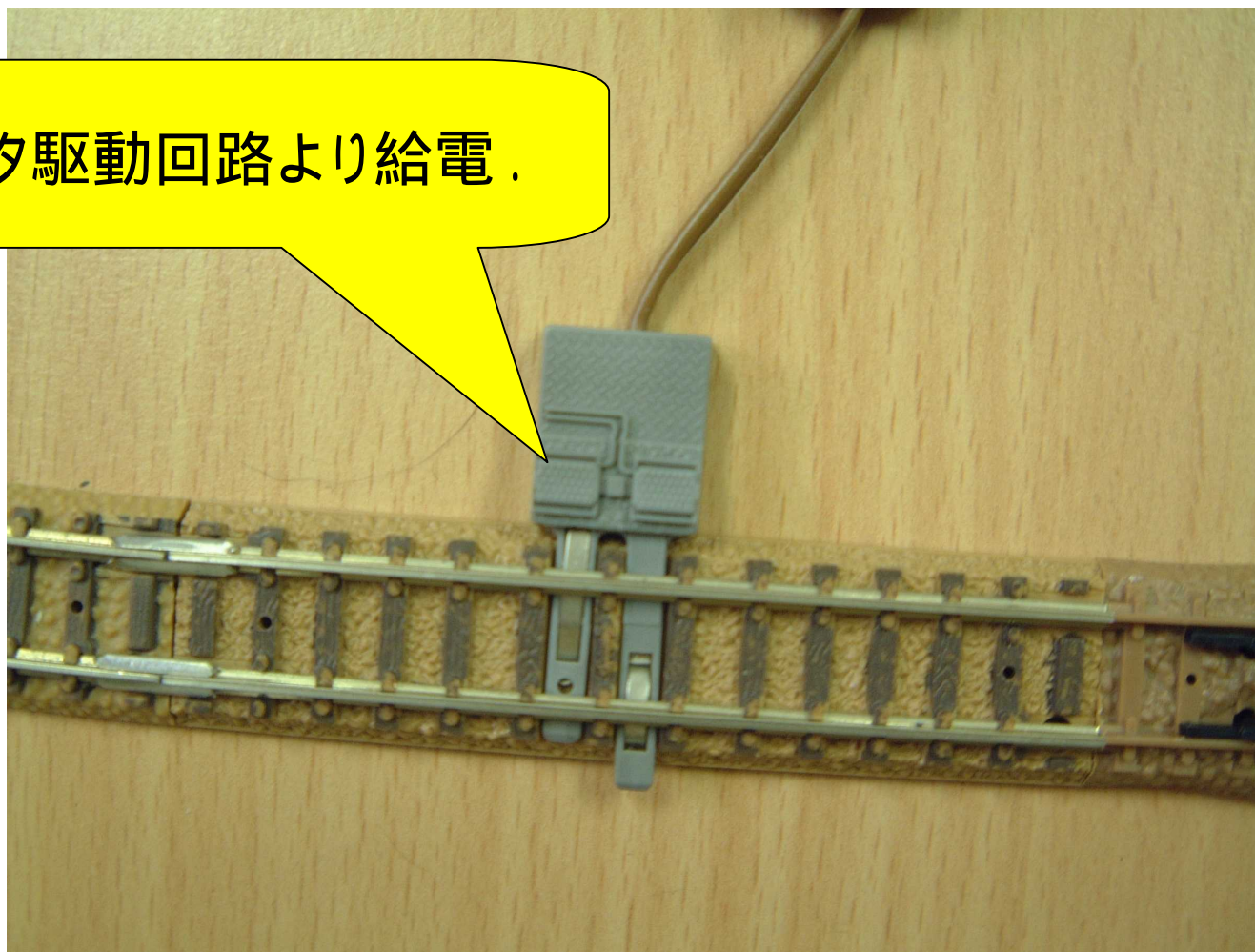


ITRONボード(2)

- プロセッサは SH2 .
- OSはITRON(TOPPERS)
- 周辺回路:
 - 8bit 平行出力 * 1
 - 8bit 平行入力 * 1
 - トグルスイッチ * 4
 - プッシュスイッチ * 4
 - 7セグメントLED * 4
 - 液晶ディスプレイ * 1
 - 8 * 8 LEDマトリックス * 1
 - ブザー
 - シリアル入出力 * 1
 - LAN * 1

レール

モータ駆動回路より給電.

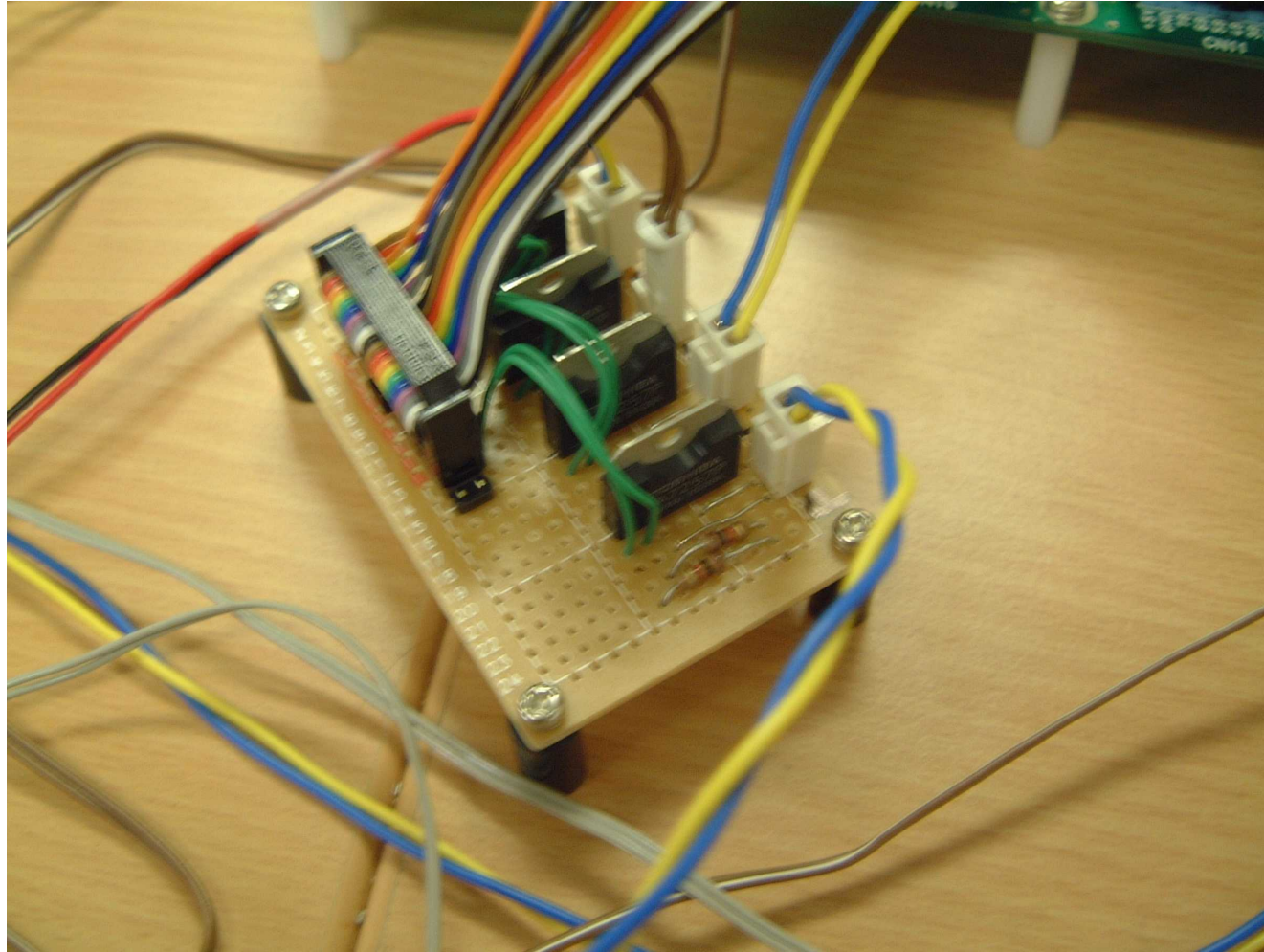


ポイント



モータ駆動回路より給電。

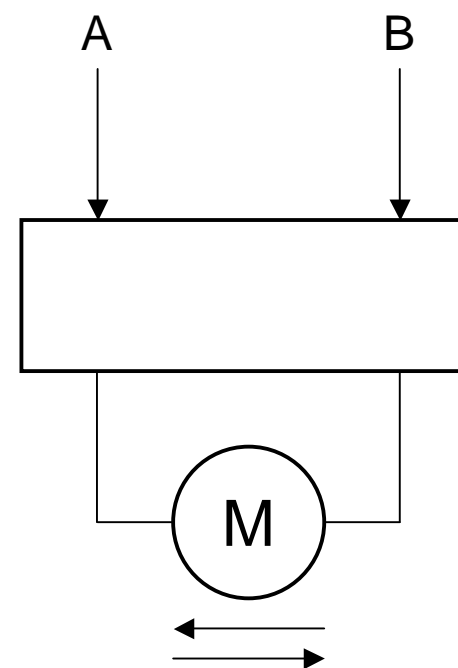
モータ駆動回路(1)



モータ駆動回路(2)

- 2bit でモータを制御.
- 8bit のパラレル出力 = 4系統のモータ駆動回路

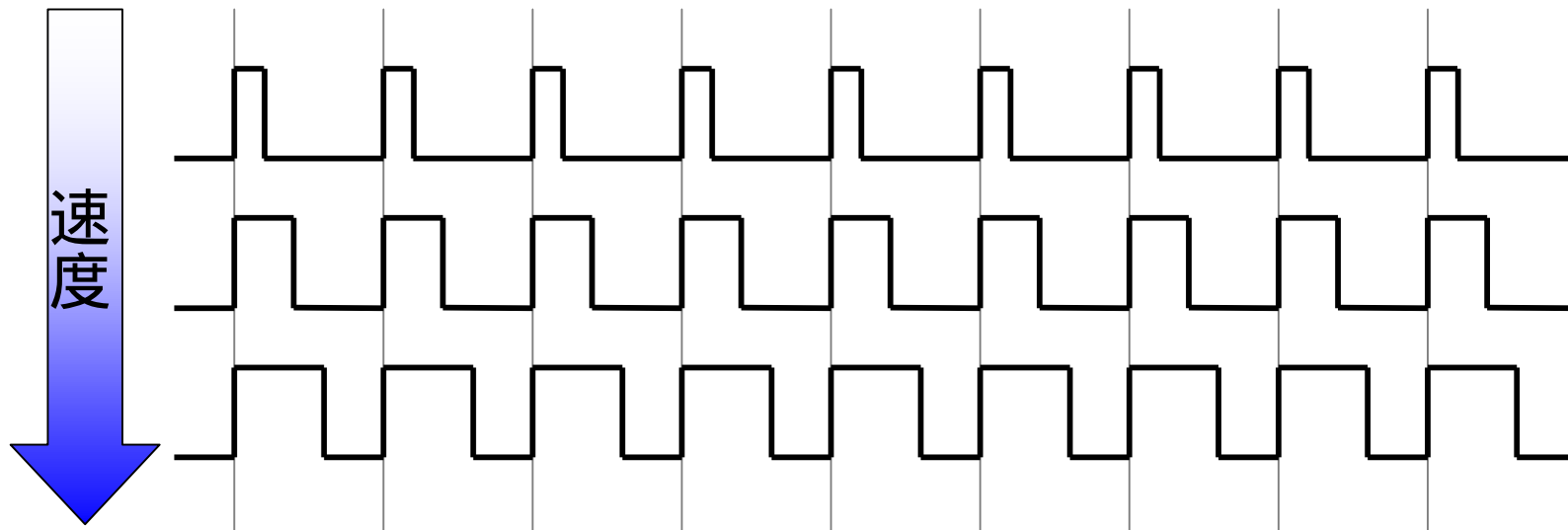
A	B	動作
0	0	停止
1	0	正回転
0	1	逆回転
1	1	制動



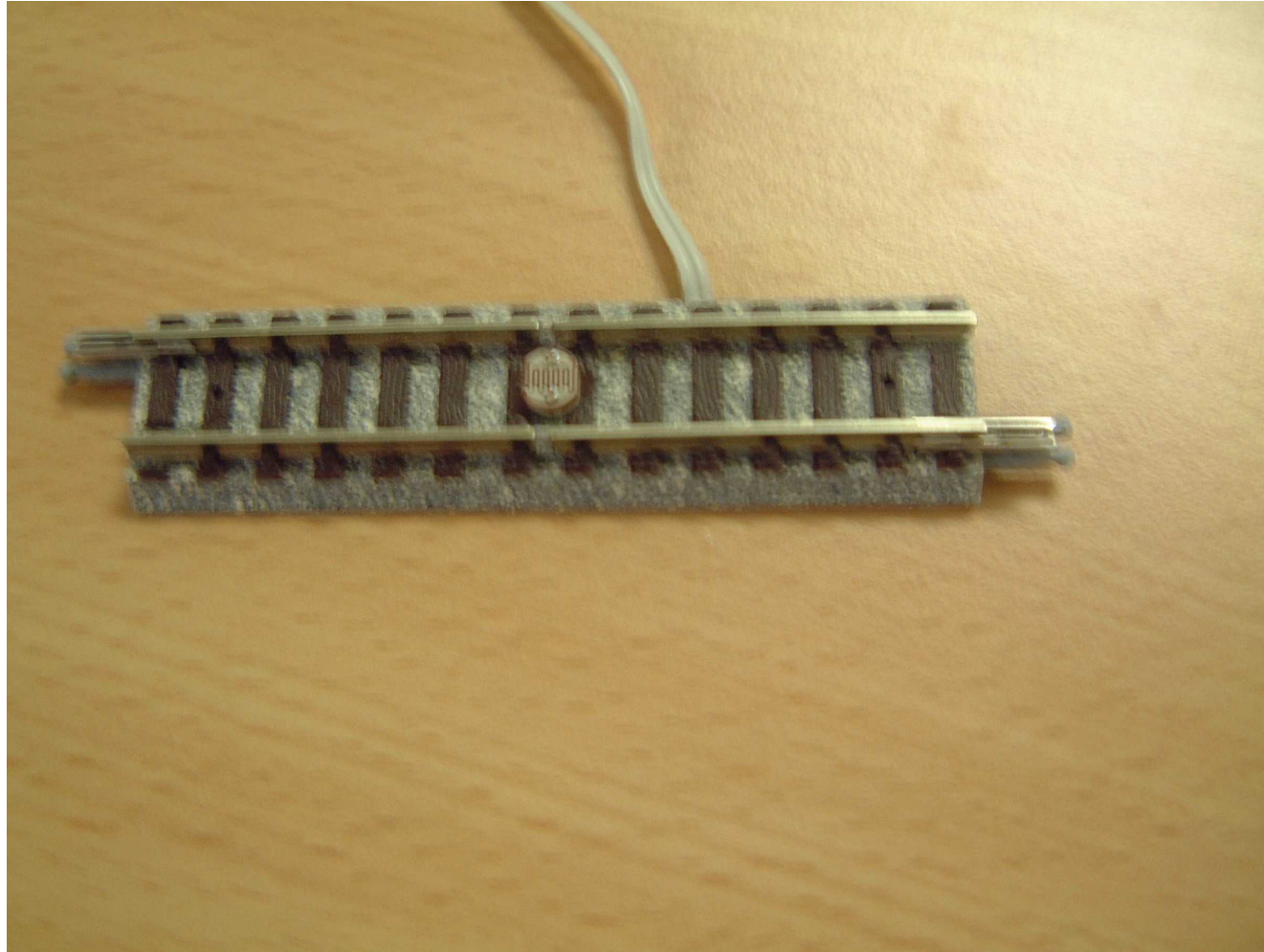
モータ駆動回路(3)

PWM制御：

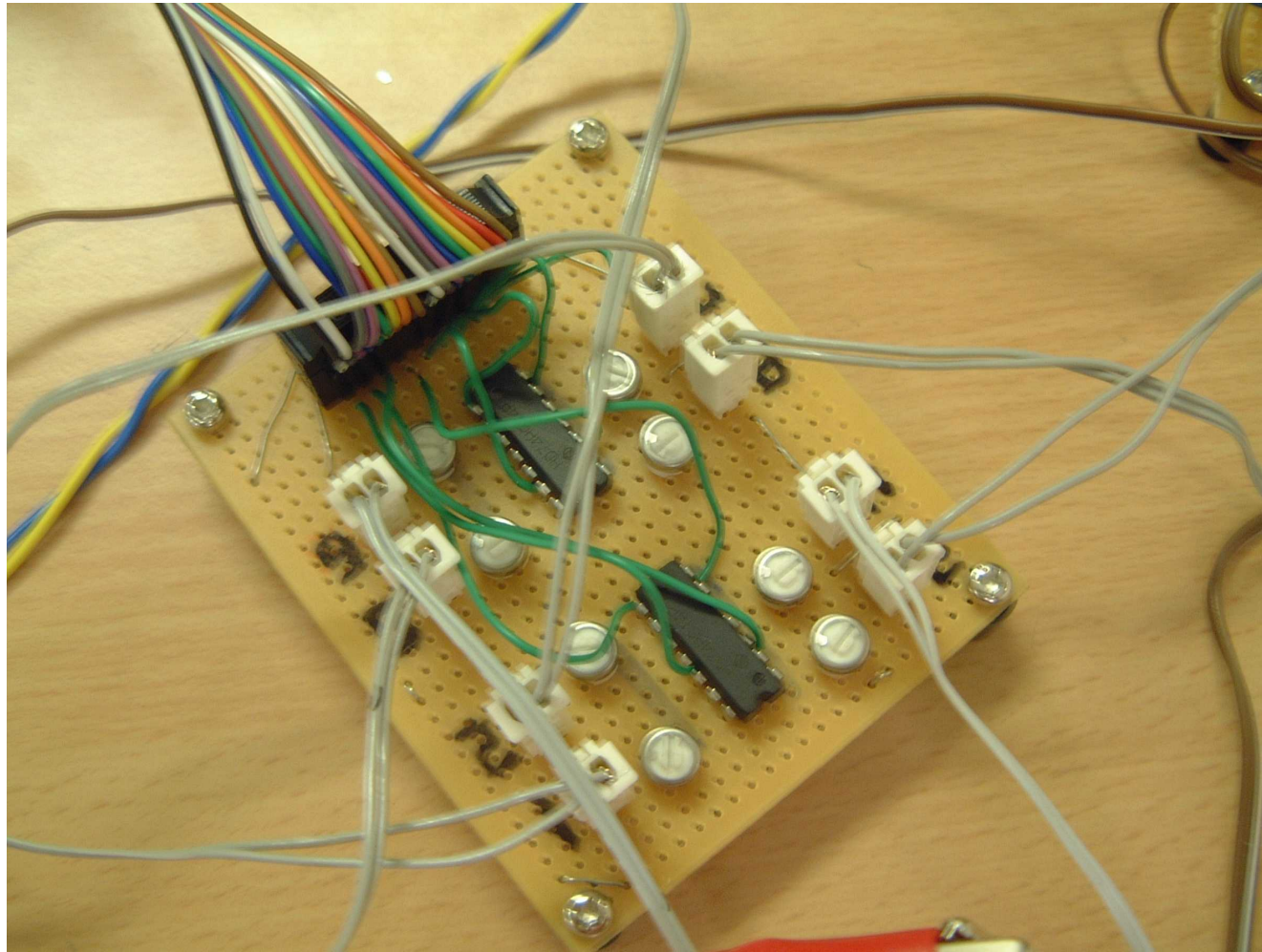
- モータにパルス状に電流を流す．
- パルスのデューティ比を変えて速度制御する．



電車通過センサ

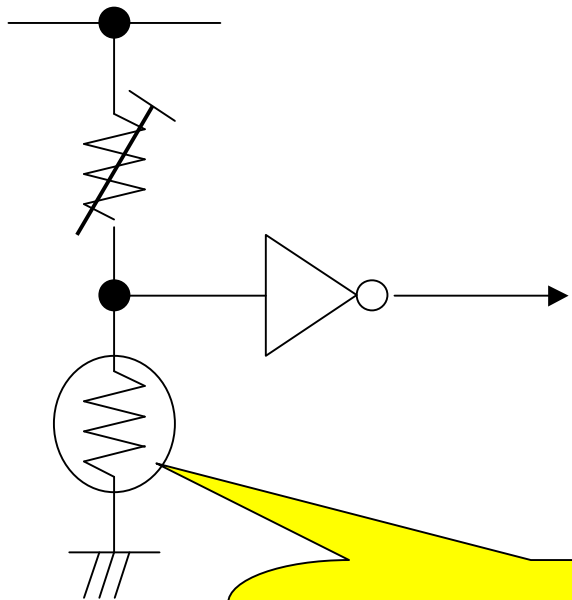


センサ回路(1)

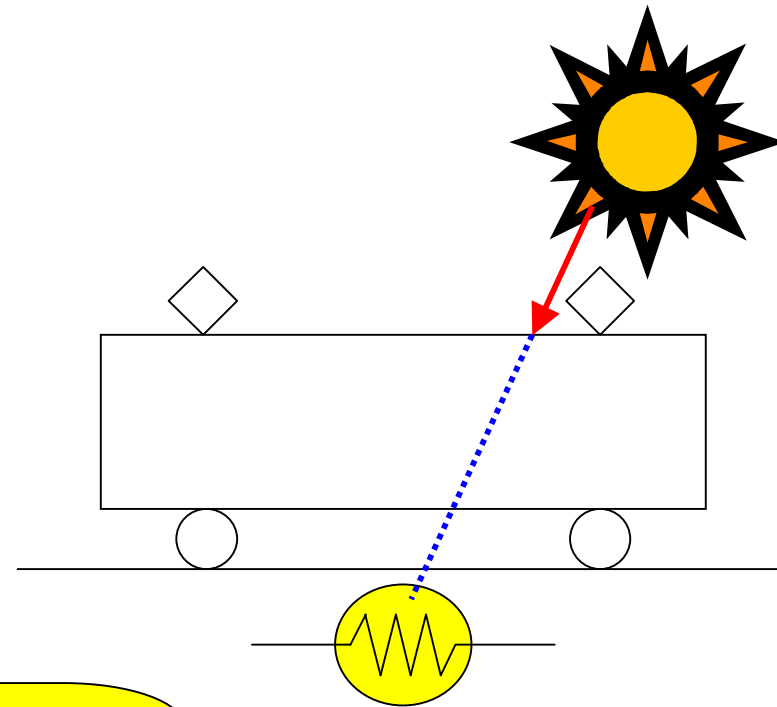


センサ回路(2)

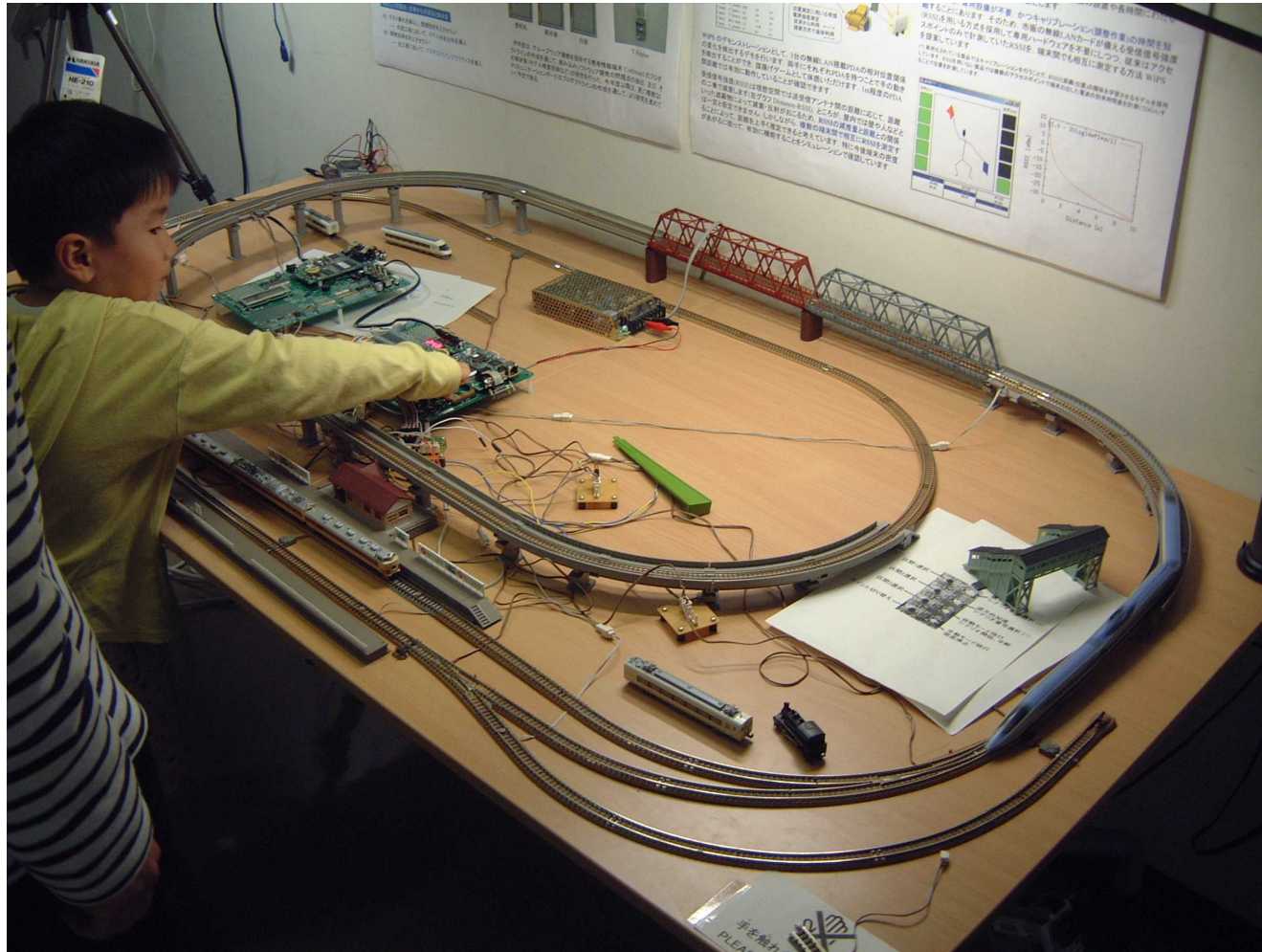
センサの原理：



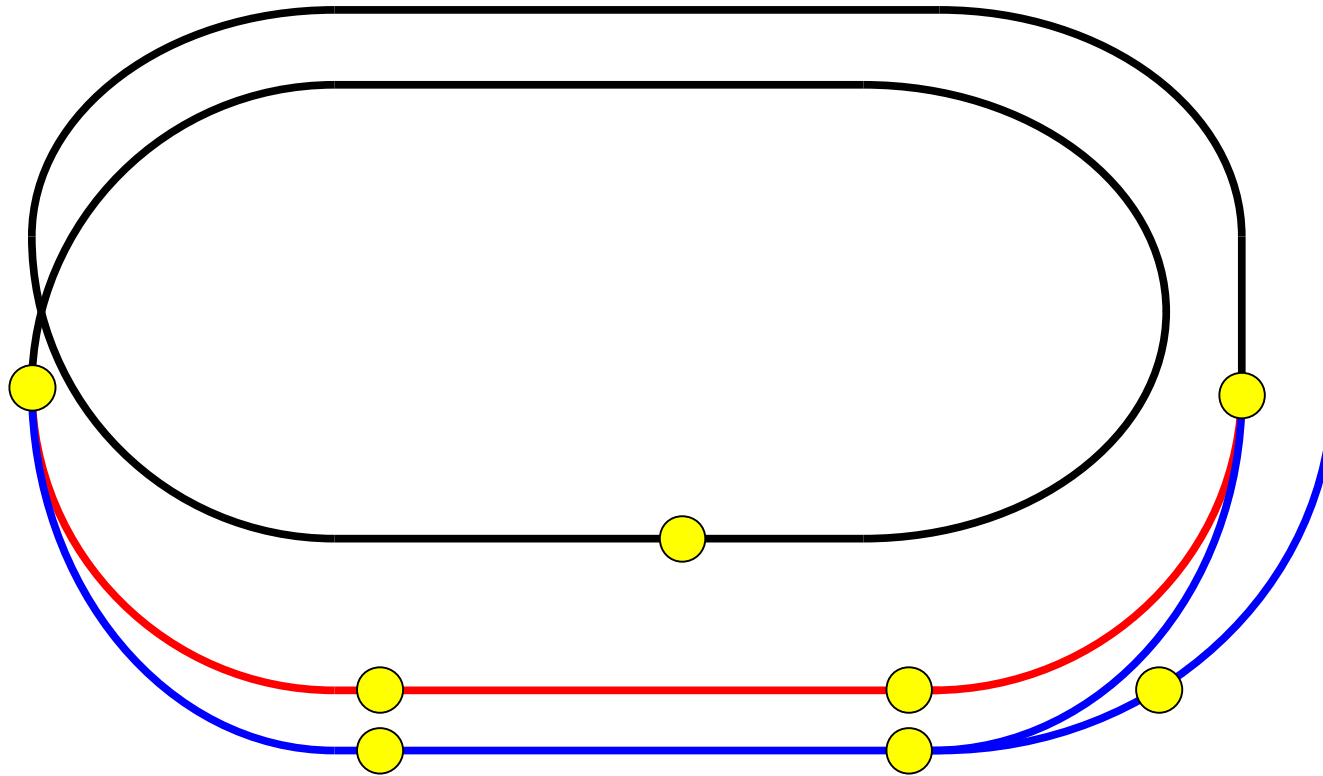
CdSセル： 光の強度
で抵抗値が変わる素子。



レイアウト全容(1)



レイアウト全容(2)



- 各送電区間(赤, 青, 黒)への送電により電車を制御.
- 3つのポイントを同時制御.
- 8ヶ所のセンサで電車の通過を検知.(イベント源)

演習課題

● 要件

- 当該システムは手動運転モードと自動運転モードを有する。
- 送電区間は3区間とし、3つのモータ駆動回路で制御する。
- 3つのポイントは全て同じひとつのモータ駆動回路で制御する。
- 手動運転モード： トグルスイッチで選択した送電区間に存在する電車の速度を、時計回り加速、反時計周り加速プッシュボタンで制御する。
- 自動運転モード： プッシュボタンで選択したシナリオに従って、2台の電車を自動運転する。
- 緊急停止ボタンを押すと全列車が停止する。

● 指示

- 自動運転のシナリオを作成せよ。
- 状態遷移図または状態遷移表を設計してから実装せよ。

シナリオ例

1. 普通列車が駅を出発する。
2. 普通列車は順次加速，その後最高速度で走行する。
3. 普通列車が半周回ると，特急列車は普通列車を追って駅を出発する。
4. 特急列車は順次加速，その後最高速度で走行する。
5. 普通列車は駅の待避線に進入し，順次減速，停止する。
6. 普通列車停止後，特急列車は駅の通過線に進入し，駅を通過する。(普通列車を追越し.)
7. 特急通過後，普通列車はスタックに進入，停止する。
8. 特急は1周後，駅に停車する。
9. 普通はスタックから駅に進入，停止する。

学生たちの感想

学生たちの感想

- 今回のソフトウェアは非常に複雑だったので、各タスクの関係や入力ポート、出力ポートのビット割当などを図や表にまとめてから作成に取り掛かった。考えたことを書き加えていくことで考えが整理され、非常にわかりやすかった。これまではこのようなプログラムの仕方はしなかった。
- ほとんどドントケアで(状態遷移)表が埋まってしまっているのだが、ドントケアでないほんの数箇所を考えるだけで長い時間を必要とした。
- 実際に動かさないとわからない、列車が区間を通過するのに必要な時間などの要素すべてに対処するような状態遷移を設計する必要性と難しさを感じたものの、時間が足りずどうしようもなかった。
- 作成するプログラムが動作するのを、列車の動きによって見ることができるのは楽しくもあり、充実感もありました。
- マルチタスクプログラミングをどのようにやるかがわかってきた。
- 今までよりビット演算やシフト演算に習熟したと思う。