LTCP(Line Trace Car Project)
設計仕様の策定と、考え方の流れ

１．設計要件

並列処理 CPU の性能を最大に引き出すように心掛ける。

２．動作仕様

① 内部の判定論理が確認できるように配慮する。
② 手動モード、自動モードと、プログラムモードが、切り替えられるものとする。
   1) 手動モードでは、直進、停止、右回転、左開展を切り替えられるものとする。リモコンカーとして動作する。
   2) 自動モードでは、濃淡が明確なラインの上をトレースするように動作させる。
   3) 自動モードでは、ライントレースのロジックを継承したまま、特定の図形では方向転換して進むものとする。
   4) プログラムモードでは、「記録された動作を逆にたどる」動作が可能なものとする。
③ 自動モード、手動モードの動作を記録できるものとする。
④ 自動モードで、「記録された動作を逆にたどる」動作が可能なものとする。

３．表示

動作指定に利用できるインディケータは、全部で 6 個ある。
① X-MOS 上のインディケータ　　×4
② ライントレースカー上のインディケータ　　×2
これらのインディケータの組み合わせによって、内部の状態が判定

４．入力

① ボタン
システムの入力は、X-MOS ボート上の二つのボタンである。
② ラインセンサ
「自動制御」の入力として、ライントレースカーの左右の「光センサ」を入力とする。
これにより、システム全体への「入力」は、４種類となる。

５．動作モードの詳細

① 動作モードの遷移

「ライントレースモード」(以下、モードLとする)を「基本状態」とする。
「プログラムモード」を、以下モードPとする。
「手動モード」を、以下モードMとする。

基本的には、モードLとモードPでは、ライントレースの動作を継続するため、ボタンの入力は不要である。モード切り替えのためのみに、ボタンを受け付ける。

1) モードL/Pで、左ボタンを長押しされたら、モードMに移行する。
2) モードLで右ボタンを長押しされたら、モードPに移行する。
3) モードPで右ボタンを長押しされたら、モードLに移行する。
4) モードLで、両ボタンが短押しされたら、「動作記録」をON/OFFする。

モードMでは、二つのボタン入力はそのまま左右の車輪制御に利用される。
このため、モードMからの状態遷移は、「左右どちらのボタンも押されないまま、５秒以上の時間が経過したら、モードLに移行する」ものとする。

モードMでは、自動的に「動作記録」をONにするものとする。

② ライントレース状態の定義

・ライントレースカーを改造する前のハードウエアでは、「左右両方の車輪が回転している」か、「右の車輪だけが回転している」か、「左の車輪だけが回転している」の、3つの状態しかなかった。
このままでは、「常に動いている」こととなるため、全体の動作を次のように定義する。

1) 左右とも、ラインセンサが「ライン」を検知していない状態では、以下のように動作判定を行う。
・モードLに切り替わった直後(ライントレースモードになったばかり)に、左右とも「ライン」を検知していない場合には、「静止」する。(基本状態とする)
・モードLに切り替わった直後ではなく、左右ともラインを検知していない場合には、「直進」する。
2) 左右両方のラインセンサが「ライン」を検知している状態では、「直進」する。
3) 右のセンサのみが「ライン」を検出した場合は、右側の車輪を停止して、左側の車輪のみを回転させる。
4) 左のセンサのみが「ライン」を検出した場合は、左側の車輪を停止して、右側の車輪のみを回転させる。

６．スイッチ入力の処理

モードＬとモードＰでは、スイッチの状態は以下の図のように処理を行い、「入力」を取りだす。

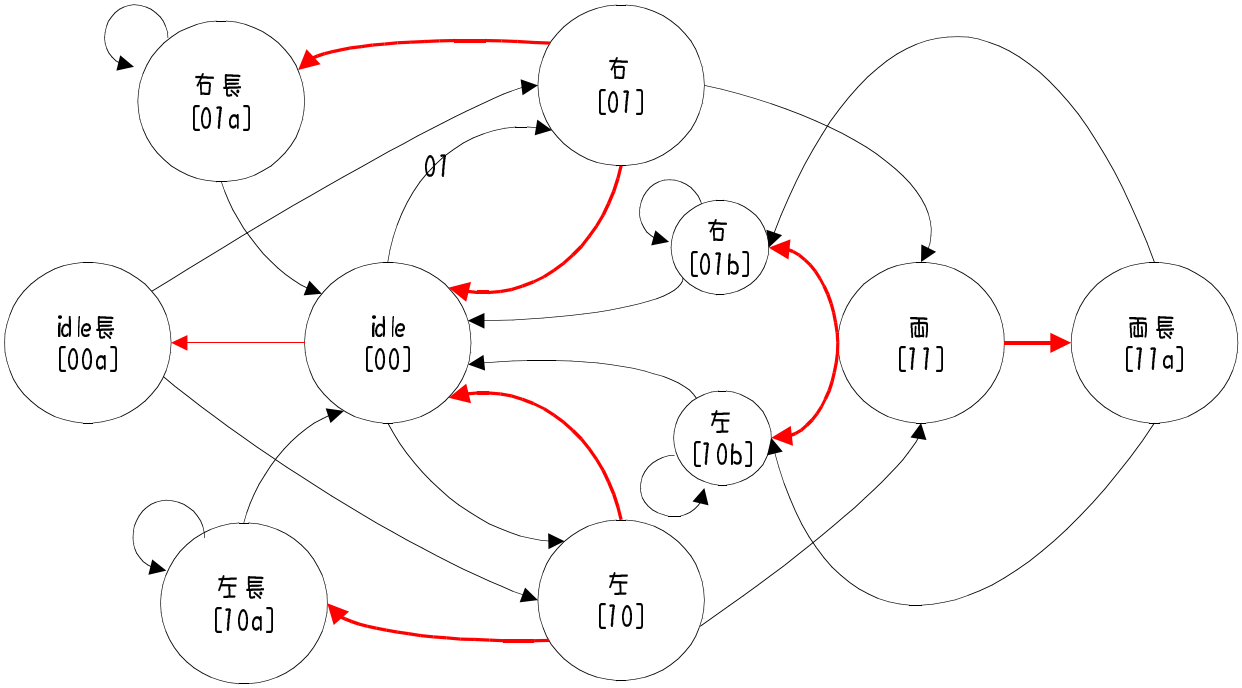

図１：SW 入力処理の状態遷移図

すなわち、スイッチの状態から判断される「入力」は、以下の７通りである。

① idel 状態が５秒以上続いた。[通知番号 1]
② 右が短く押された。[通知番号 2]
③ 右が長く押された。[通知番号 3]
④ 左が短く押された。[通知番号 4]
⑤ 左が長く押された。[通知番号 5]
⑥ 両方の SW が短く押された。[通知番号 6]
⑦ 両方の SW が長く押された。[通知番号 7]

図１において、「入力」が確定するのは赤い矢印の状態遷移が発生したタイミングである。例えば、idle 状態から「左」が押された場合には[10]の状態に遷移するが、まだ「入力」の確定には至らない。そのまま「左」が離された場合には、「左短押」の入力が確定する。また、そのまま３秒以上押し続けられた場合には、「左長押」の入力が確定となる。また、さらに右が押された場合には[11]の状態に遷移するが、[11]ではまだ入力は確定しない。

この状態遷移を表に表すと、表１の通りとなる。

表１：キー入力処理の状態遷移表（ライントレースモード/プログラムモード）

	idle [00]	idle長 [00a]	右押 [01]	右長押 [01a]	右戻り [01b]	左押 [10]	左長押 [10a]	左戻り [10b]	両押 [11]	両長押 [11a]
右押	→[01] Time	→[01]				→[11] Time	→[10a] 無視	→[10a] 無視		
右離			→[00] (右)	→[00] ※	→[00] ※				→[10b] (両)	→[10b]
左押	→[10] Time	→[10]	→[11] Time	→[01a] 無視	→[01b] 無視					
左離						→[00] (左)	→[00] ※	→[00] ※	→[01b] (両)	→[01b]
Timer	→[00a] (idle長)	無視	→[01a] (右長)	無視	無視	→[10a] (左長)	無視	無視	→[11b] (両長)	無視

この表で、例えば右が押された状態[01]で、右が離された場合には、「入力・右短押」が確定し、状態は[00]に移ることがわかる。

また、「何もキーが押されていない状態」では、「右のキーを離す」イベントは発生し得ない。こうしたあり得ない「状態」と「イベント」の組み合わせの升目は、塗りつぶしてある。

表２：キー入力処理の状態遷移表（マニュアルモード）

	idle [00]	idle長 [00a]	右押 [01]	左押 [10]	両押 [11]
右押	→[01] (右)	→[01] (右)		→[11] (両)	
右離			→[00] (無) Time		→[10] (左)
左押	→[10] (左)	→[10] (左)	→[11] (両)		
左離				→[00] (無) Time	→[01] (右)
Timer	→[00a] (idle長)	無視	無視	無視	無視

マニュアルモードでは、キーが押されたらすぐに通知する。長押しの判定を行うのはIDLE 時のみとし、右長押しなどの遷移は処理しない。

「IDEL 時間が長い」遷移状態はモードが切り替わるため存在しえないが、トラップとして処理を残す。

７．内部状態の表示

４つの LED を用いて、内部の状態を表示する。

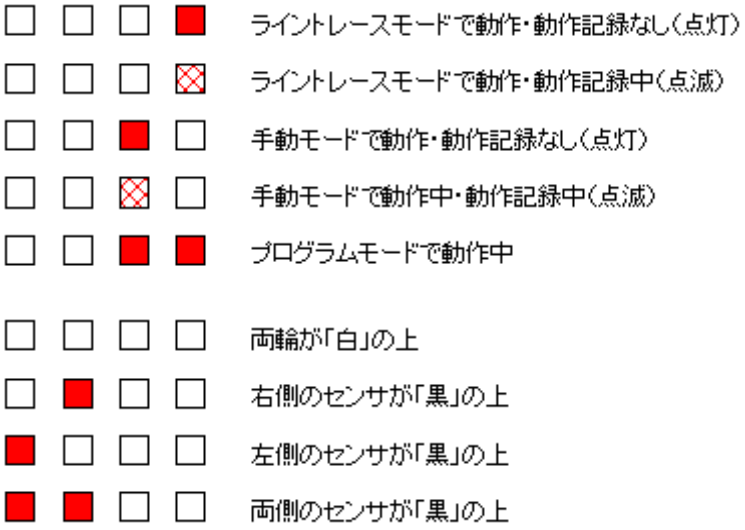

図２：LED による状態表示パターン

ライントレースカー上の LED は、右の車輪が ON 状態で右の LED を、左の車輪が ON

状態で左の LED を点灯させる。

８．全体の構成

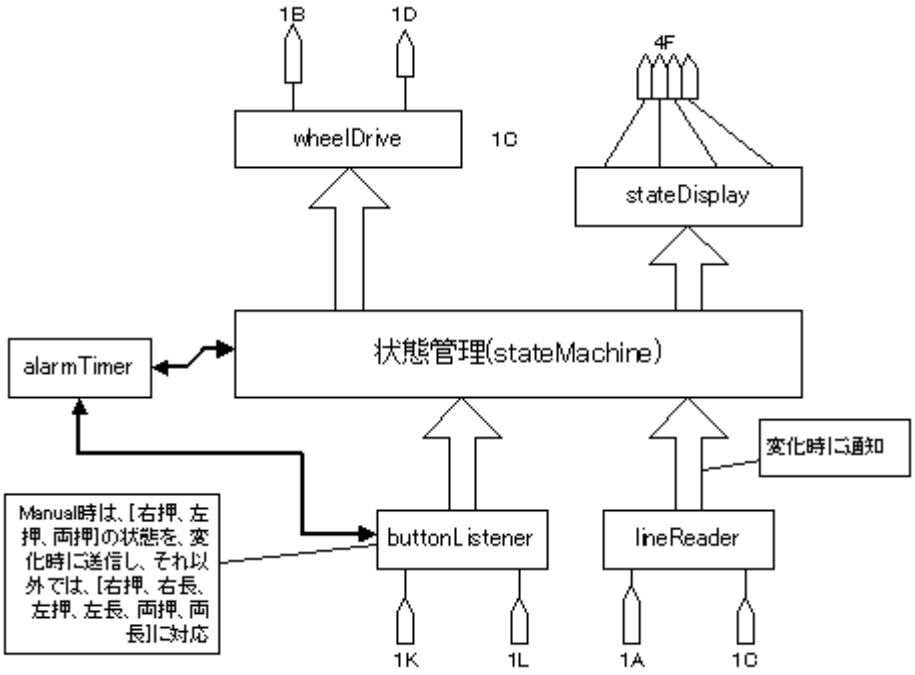

表３：全体の構成

それぞれのモジュールに機能を割り振り、チャンネルを経由したデータの受け渡しにより、連携動作させる。

９．通信チャンネルの仕様

① buttonListener から stateMachine へのデータ

コード	モード L, モード P	モード M
1(001)	idle 状態が 5 秒以上続いた。	両方押されていない
2(010)	右が短く押された。	右が押された
3	右が長く押された	
4(100)	左が短く押された	左が押された
5	左が長く押された	
6(110)	短く両押しされた	両押しされた
7(111)	長く両押しされた	モード L への移行 (idle5 秒)

② lineReader から stateMachine へのデータ

コード	意味
0(00)	両輪とも「白」
1(01)	右輪が「黒」線検出
2(10)	左輪が「黒」線検出
3(11)	両輪が「黒」線検出

③　stateMachine から wheelDrive へのデータ

コード	意味
0(00)	両輪とも停止
1(01)	右輪を駆動
2(10)	左輪を駆動
3(11)	両輪を駆動

④　stateMachine から stateDisplay へのデータ

データ	意味
00 000X	１ビット目の値を X とする
00 00X0	２ビット目の値を X とする
00 0X00	３ビット目の値を X とする
00 X000	４ビット目の値を X とする
01 0001	１ビット目の値を点滅とする。
10 0010	２ビット目の値を点滅とする。

１０．モード L での動作

モード L (ライントレースモード) での動作は、下記のとおりとする。

①　右輪側のみが「黒」線を検出している時
　　左車輪が「黒」線からはみ出しているため、右輪の回転を停止し、左輪の駆動のみを ON とする。

②　左輪側のみが「黒」線を検出している時
　　右車輪が「黒」線からはみ出しているため、左輪の回転を停止し、右輪の駆動のみを ON とする。

③　両輪が「黒」線を検出している時
　　両輪の駆動を ON とする。
④　両輪が「黒」線を検出していない時
　　ライントレースモードに切り替わった直後でない場合は、
　　両輪の駆動を ON として、そのまま直進する。

１１．動作の記録

モード L (ライントレースモード) と、モード M (手動モード) において、「両ボタンの長押し」があった際には、「動作記録」を ON/OFF する。

(1)　動作記録の回数
　　配列のサイズの上限を 16 とし、16 回までの動作を記録するものとする。
　　MAX_TRACE_COUNT　　16 として定義する。

(2)　動作記録用の配列・変数
　①　traceCounter　　配列に格納された「動作」の個数
　　MAX_TRACE_COUNT を上限とする。
　②　eventType[]　　発生した動作の種類を、以下のコードで記録する。

コード	意味
0(00)	両輪とも停止
1(01)	右輪を駆動
2(10)	左輪を駆動
3(11)	両輪を駆動

　③　eventPeriod[]　　発生した動作の持続時間を記録する。

(3)　動作記録処理
　①　「動作記録」ON 時の処理
　・「動作記録」ON では、それまでの動作記録をクリアして、新たに記録を開始する。
　・引き続き、最初の動作として、「動作記録」処理を行う。

　②　「動作記録」の処理
　インクリメントした際に、traceCouonter が MAX_TRACE_COUNT に

達する場合（traceCounter==MAX_TRACE_COUNT-1）は、
「動作記録」OFF の処理に移る。

最初の動作でなかったら(if (traceCounter>0))、以下の操作を行う。

- 「経過時間」を計算する。
- eventPeriod[traceCounter]に、動作の経過時間を格納する。
  （これで、eventPeriod と eventType の組みのデータが完成する。）
- traceCounter をインクリメントする。（１加算する。）

共通で、以下の操作を行う。

- 「経過時間」計算のため、現在時刻を記録する。
- 記録される動作を、eventType[traceCounter]に格納する。

③ 「動作記録」OFF 時の処理

- 「経過時間」を計算する。
- eventPeriod[traceCounter]に、動作の経過時間を格納する。
  （これで、eventPeriod と eventType の組みのデータが完成する。）
- traceCounter をインクリメントする。（１加算する。）

１２．モード P での動作

モード P（プログラムモード）では、以下の動作を行う。
tracePointer を、traceCounter-1 からまで変化させ、以下の処理を反復する。

(1) 記録された動作の「再生」処理

① eventPeriod[tracePointer]の値（動作の持続時間）を alarmTimer に設定する。

② eventType[tracePointer]の値から、以下の変換を行い、ライントレースカーに設定する。

※ 「右輪を駆動」(01)のコードの場合は、「左輪を駆動」(10)のコードをライントレースカーに送信する。（左折の逆動作は右折）

※ 「左輪を駆動」(10)のコードの場合は、「右輪を駆動」(01)のコードをライントレースカーに送信する。（右折の逆動作は左折）

(2) アラームから、「時間経過」の通知を受けた際の処理

① tracePointer が 0 の場合は、そのまま「静止」状態で待機する。

② tracePointer が 0 以外の場合 (次の動作がある場合) は、デクリメント (１減算) する。

③ デクリメント後、次の動作の「再生」処理 ((1)の処理) を行う。

１３．開発の手順 / テストの項目

作業を 1 ステップずつ進める。

① LED の点灯処理

alarmTimer と stateDisplay を仮実装し、１秒ごとに LED が順次点灯する処理を作成する。

【テスト内容】１秒ごとに LED が順次左側に点灯する。

② SW の入力処理

buttonListener を作成する。

SW の入力により、LED の点灯を一つずつ左に巡回してずらす。

【テスト内容】左の SW を 1 回押す (100 受信) ごとに、LED 点灯が左に一つずつ移動することを確認する。また、右の SW を 1 回押す (010 受信) ごとに、LED 点灯が右に一つずつ移動することを確認する。

③ lineReader の入力処理

lineReader の入力を、LED の状態 display に反映させる。

【テスト内容】ライントレースカーを黒い線の上にかざし、左や右のセンサーがそれぞれ黒い線に反応していることを確認する。

④ alarmTimer と buttonListener の連携動作の確認

buttonListener の状態遷移図を実装し、各ボタンの 3 秒長押し状態を確認する。

【テスト内容】[９通信チャンネル]で記述した、buttonListener から stateMachine へのキー入力コードの番号を、そのまま LED に表示して、buttonListener の動作を検証する。

⑤ stateMachine を実装する。

ボタンを押すことによって、「動作モード」を変える。

stateDisplay へ出力を実装し、内部モードの変化が LED 表示できるようにする。

【テスト内容】ボタンを押すと、モード切り替えができることを確認する。

手動モード、自動モード、プログラムモードのモード切り替えが、仕様書通りになっているか、確認する。

⑥　モードM(手動モード)を実装する。
手動モードでは、左右のボタンを押すと、それぞれのボタンに対応する車輪を駆動させる。
【テスト内容】手動モードで、左右のボタンを押すと、対応する車輪が駆動されることを確認する。

⑦　モードL(ライントレースモード)を実装する。
ライントレースモードで、ライントレースカーのセンサーが線を読み込んだならその内容に応じた動作([10 モードL の動作]で記述した動作)をすることを確認する。
【テスト内容】ライントレースモードで、ライントレースカーが、ライントレースの動作をすることを確認する。

⑧　動作の記録を行う。
動作記録のアルゴリズムを組み込む。手動モードで動作を記録させ、動作記録OFFと同時に、１秒ごとに「記録された内容」をLED表示させて、配列の内容を目視確認する。
【テスト内容】手動モードで、「動作記録」をON/OFFさせると、LEDのインジケータが、点灯から点滅に切り替わることを確認する。
LEDのインジケータ点滅時に、記録させた動作が、「動作記録OFF」になった後に、LEDに再生表示させる「開発用モジュール」を作成して、記録が正しいことを確認する。LEDに再生表示させる内容は、eventTypeのみとし、eventPeriodは再生しないものとする。

⑨　モードP(プログラムモード)を作成する。
記録された動作を逆に辿る。
【テスト内容】手動モードで動作を記録させ、180度回転させた後、プログラムモードに切り替えると、それまで走行させたルートを、そのまま逆に辿る動作をすることを、確認する。

１４．改版履歴

初版：(V1.0) 2011.01.05 執筆
V1.1：2011.01.08　：　alarmTimerをモジュールからtimer変数に変更した。
内部状態のLED表示を変更した。
key入力の状態遷移表で、時刻チェックの記録を修正した。

V1.2：2011.01.08　：　マニュアルモードでのキー判定状態遷移表を追加した。
マニュアルモードでは、実行最初から動作記録を行うものとした。
動作記録は timeKeeper を使わない方法とした。
全体構成から timeKeeper を削除した。