

# ログファイルの内部構造について

## 1 はじめに

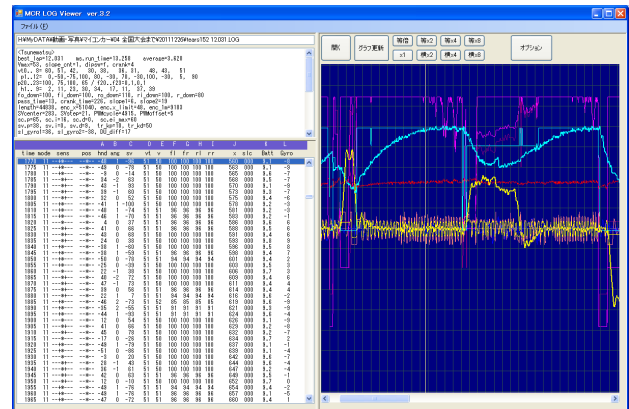
ロボットをプログラムによって制御する場合、内部の様子を知ることが効率的なデバッグに欠かせません。とりわけマイコンカーは高速で移動し、10数秒で動作が完了するので、肉眼による観察のみでは一瞬の間に何が起こったのかを正確に掴むことはほぼ不可能です。よって、走行中の状態を記録しておき、走行後にそれを解析するのが一般的です。

走行中は肉眼で走行を観察し、同時にカメラで動画を撮影しておき、走行が終了すると動画を再生し、コマ送りで何度も挙動を観察しながら、記録されたログと見比べて、「センサのここが反応したときに、右後のブレーキをもう少し強めにかけてみると、こういう動きになるかも。」「このタイミングで前の外側のブレーキを緩めるとグリップが回復するかも。」といった次の一手を考え、それをプログラムに反映して次の走行を開始する。といった感じで、一回一回の走行を無駄にしないことが大切です。これらの一連の作業を繰り返すことによって、「安定した」プログラムに熟成していきます。その際、走行が終了すると一刻も早くログの確認をし、グラフも観察したいところですが、ここで Excel に取り込んでグラフ化するという繁雑な作業が入り込んでくると、思考が中断されてしまうことになるので、できれば簡略化したい部分です。

公式ページではマイクロ SD に関する資料が公開されているので、これを利用して本校では数年前よりマイクロ SD へログを保存し、専用のログビューアで閲覧する環境を構築してきました。これは一走行ごとに作成されたログファイルをエクスプローラからドロップすると即座にグラフ化までできるものです。これによってそれまでの苦労が一気に解決し、快適なデバッグ環境となりました。

ここでは、本校で使用しているログビューアや仕様を公開したいと思います。ログビューアに関しては、全てのソースコードも添付しますので、各マシンに応じて自由に改変していただければと思います。

以下の内容は、本校の生徒向けに作成した資料を、ほとんどそのままの形で掲載したものです。一般的な内容でないものも含まれていますが、内容にまで手を加え始めると、一生公開できない気がするのので、あえてそのままとします。



ログビューアの画面

## 2 走行からログの確認までの流れ

- (1) マシンにマイクロ SD カードを挿入する。
- (2) 走行中は自分の目でマシンの挙動を追いかけ、同時に動画撮影をする。
- (3) 走行が終了したら、マイクロ SD カードを抜き取り、PC へコピーする。(1 走行 1 ファイル)
- (4) コピーしたログファイルをログビューアへ読み込む。

PC へコピーする際、日付ごとのフォルダで分けたり、タイムやどこでコースアウトしたかなどの補足事項をファイル名に追加しておくで後々便利である。

## 3 概要

### (1) 仕様

- ・ 17 バイトの走行データを 5ms 毎に microSD カードに記録する。
- ・ 1 回の走行につき、1 つのファイルに記録する。
- ・ 記録されたファイルはエクスプローラ等で PC にコピーできる。
- ・ 記録されたファイルの中身は専用のログビューア (MCRlogViewer.exe) で見ることができる。
- ・ microSD カードは予め FAT16 でフォーマットしておき、直後に専用のアプリケーション (SDInit2.exe) で空のファイルを 512 個程度作成しておく。(全て 0 を書き込むだけ) これによりファイルの中身も整然と並んだ状態となっている。
- ・ マイコンカーでは走行前に記録するファイルの先頭アドレスのみを取得しておき、走行中はそこから連続して書き込んでいく。(走行中は FAT テーブルの参照はしない)
- ・ 走行前のログファイル消去処理はせず、前回の走行データが残っていても、それに上書きしながら書き込んでいく。走行終了後に最後のレコードを 0 で埋め、ビューアで読み出すときもその場所までを読み込むこととする。走行前の待ち時間がない。

### (2) ログファイルの拡張子

走行時にログが記録されたファイルの拡張子を、走行終了後に以下のように “.LO\_” から “.LOG” に変更する。

001.LOG	}	書き込み済のログファイル
002.LOG		
003.LOG		
004.LO_		←このファイルにログ記録し、走行終了後に 004.LOG と拡張子を変更する。
005.LO_	}	空のログファイル
006.LO_		
007.LO_		

書き込み対象の空ファイルは、FAT テーブルの先頭から “.LO\_” の拡張子を探し、最初に見つかったファイルとなる。

全てのログファイルが記録済の “.LOG” となったときは、単に全ての拡張子を “.LO\_” に書き換えるだけで、見かけ上まっさらの状態となる。これはマイコンカー側のプログラムによって、この初期化処理をしている。

## 3-2 ログ記録～ログ解析の流れ

### (1) 前準備 (最初の 1 回だけ)

- ・ microSD カードを FAT16 形式でフォーマットする。
- ・ SDInit2.exe で 1MB の空ファイルを 512 個作成しておく。(拡張子は \*.LO\_)

### (2) 走行

- ・ マイコンカーに microSD カードを差し込み、走行させる。
- ・ 1 回の走行につき 1 つのファイルに走行ログが記録される。書き込まれたファイル名は走行後液晶に表示される。
- ・ リセットがかかるなど、正常に終了しなかった場合でも、書き込まれる予定の \*.LO\_ ファイルを PC にコピーして、拡張子を .LOG に変更すると、途中までのログが記録されている場合がある。

### (3) PC での読み取り

- ・ PC のカードリーダーに microSD カードを差し込み、\*.LOG ファイルを PC にコピーする。日付別のフォルダを作成しておくといよい。
- ・ ログビューア (MCRlogViewer.exe) にログファイルを読み込ませると中身が確認できる。
- ・ ハードディスクにコピーされたログファイルの場合、ログビューアに読み込まれた時点で、未記入の余分な部分が削除され、ファイルサイズが小さくなる。

(4) microSD カード上のログファイルが全て書き込み済みになった場合、マイコンカー上のメニューでクリア処理をすると、拡張子が全て “\*.LO\_” に変わり、空のファイルとして使用できる。もちろんフォーマットからやり直してもよい。

(5) タイム計測について (補足)

1 周のタイムは、最初のクランクのクロスラインから計測を開始し、あらかじめ設定したクランクの数の分だけクロスラインを通過したときまでのタイムを記録し、一番早いタイムをベストラップとして液晶に表示するとともに、ログファイルにもテキスト形式で記録している。これによって、特別に外部にタイマーなどの装置を必要とせず、1000 分の 1 秒単位でマイコンカー自身にタイム計測させている。よって普段の練習では、クランクの手前からスタートさせることが多い。言うまでもないが、大会や試走の時など、最初に通過したクロスラインに到達する前に停止すると、タイムが記録されないのが難点である。

## 4 FAT 構造を壊さない microSD カードへのログ記録

マイコンカーの走行中は、センサの状態取得やモータの制御に全力を注いでいるので、ログ記録にあまり時間をかけられない。メモリも限られている中で FAT テーブルを参照しながらのログ記録は余り現実的ではない。

そこで、SD カードを PC でフォーマットし、直後に 1MB 単位の空のファイルを 512 個作成しておく、各ファイルの中身は整然と並んだ状態になっているので、走行ログを書き込む際には先頭アドレスのみ取得すればあとはそこから連続して書き込んでいけばよい。ただし、PC で SD カードに対して書込を伴う作業をするとファイルの並びがバラバラになってしまい、正しく書き込めなくなってしまうので、基本的に SD カードは PC では読み込み専用とするよう注意が必要である。

1 レコードの 17 バイトを 5ms 毎に 15 秒程度記録するとしても単純計算で 51kB 程度であるので、1MB のファイルであれば十分過ぎる程のサイズである。空のファイルの拡張子は “\*.LO\_” となっており、書込が終わったファイルの拡張子は走行後に “\*.LOG” に変更される。

プログラムは、マイコンカーラリー実行委員会が作成された microSD 関連ライブラリを基本読み書きプログラムとして利用し、それに被せる形で my\_sdfat08.c を実装している。

### 4-1 ログファイルの構造

(1) 概要

- ・ ログファイルは先頭の 3 セクタ分がテキスト形式で、残りがバイナリ形式となっている。
- ・ 先頭のテキスト形式データがそのままログビューアのテキスト表示エリアに表示される。
- ・ バイナリ形式データは 1 レコード 17 バイト
- ・ 最後のレコードは 17 バイト全てが 0 のデータとする。

ログビューアではバイナリデータを 1 セクタずつバッファに読み込み、そこから 1 レコードずつ読み込む。その際、先頭データが 0 のとき読み込み終了し、-2 のとき 1 セクタ分の残りのデータを無視し、次のセクタの読み込みをする。

よって走行前に全て 0 に初期化する必要はなく、前回の走行ログが残ったままでそれに上書きしながら書き込んでいき、最後のレコードに 17 バイト分の 0 を書き込んでいる。走り出すまでの待ち時間がほぼ 0 である。

- ・ テキストデータの内容は、ベストタイムや最高速度、平均速度、各設定値など、全体的なデータを printf 関数で出力したもの。
- ・ バイナリデータの内容は、5ms 毎のモードやセンサの値、ハンドル角、目標速度や現在の速度、各輪のモータ出力等。(詳細は後述)

(2) ログファイル (\*\*\*\*\*.LOG) をテキストエディタ等で開いたときの中身

```
#006
017
<Kimura>
best_lap=12.321    ms.run_time=12.562    average=3.775
Vmax=51, slope_cnt=1, dipsw=d, crank=4
vt0.. 8= 60, 49, 38,    30, 37,    37, 33,    45, 39,    46
:
: 省略
:

<END>

bytesPerSector    = 200
sectorsPerCluster = 40
reservedSectors   = 8
numberOfFATs      = 2
:
: 省略
:
```

← ログ形式のバージョン No.  
← 1レコードのバイト数  
← ここから走行データが書き込まれている。  
この部分は、そのままログビューアのテキストデータ表示  
エリアに表示される。

← ログビューアのテキストデータ表示エリアにはここまでが表示される

← FAT 情報など、ログビューアに表示させるまでもないが、テキストエディタ等で  
確認したい情報などは、<END>以降の 3 セクタ目までの領域にテキスト情報を  
書き込むと良い。

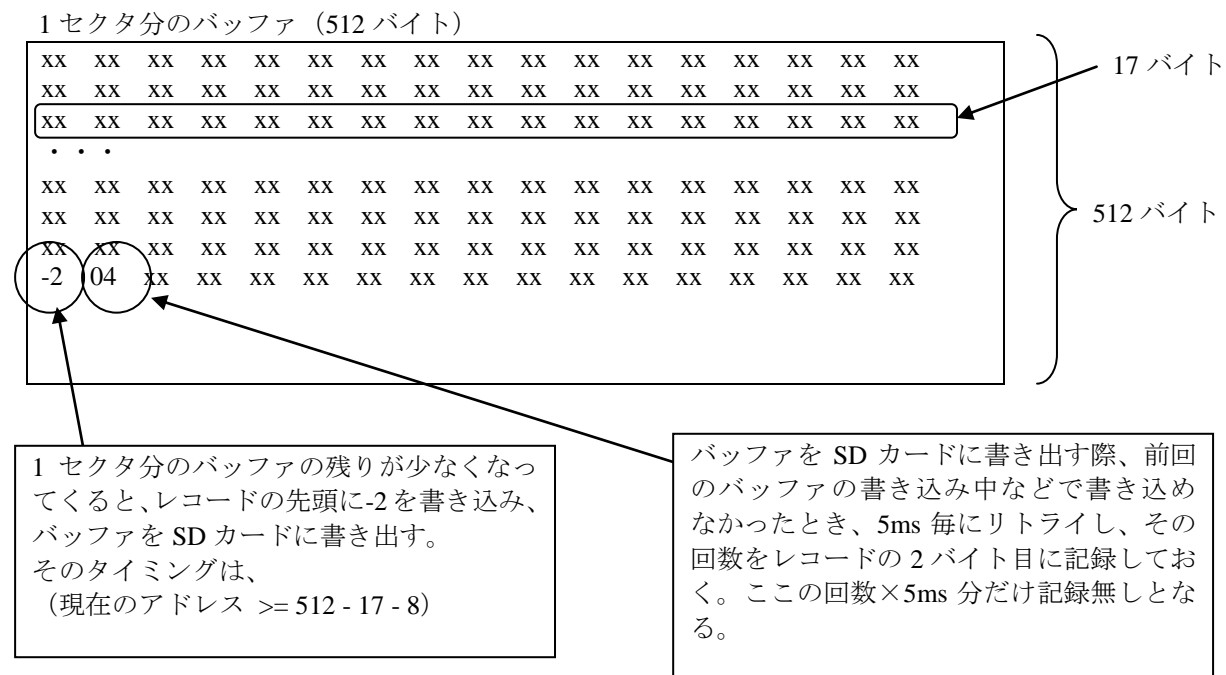
～ ファイルの 3 セクタ目から 1 レコード 17 バイトのバイナリ形式データが続く ～  
(テキストエディタ等で開いた場合は、文字化けしている)  
最後のレコードは 17 バイト全て 0 が書き込まれている。

(3) バイナリデータの 1 レコードの構造

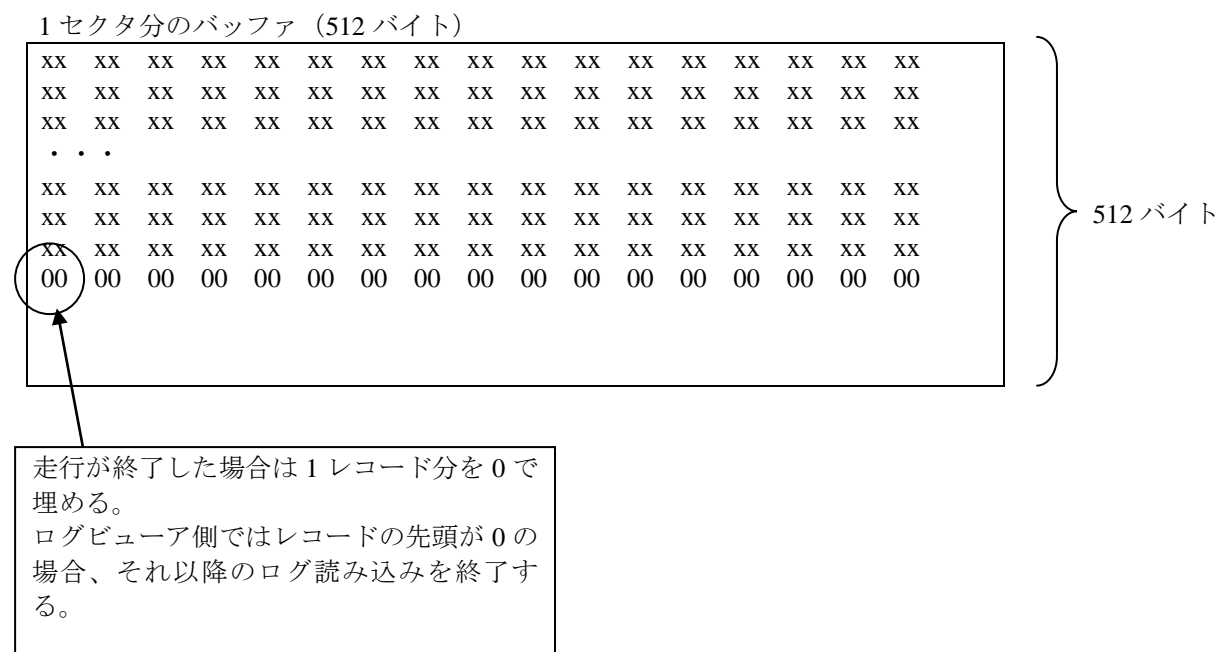
※ 1 レコードのバイト数=17 バイト

byte	内容
0	ms.mode／現在の走行モード
1	ms.sens／デジタルセンサの値
2	トレースモードのとき → ms.anSens >> 4 /アナログセンサの値 ハンドル角指定モードのとき → ms.angle_t /指定ハンドル角
3	ms.angle／ハンドルの角度（中心：0、左方向：－、右方向：＋）
4	ms.powSV／ハンドルのサーボモータ出力（－100～100％）
5	ms.log_vt／目標速度（×10m／s）
6	ms.enc.v／現在の速度（×10m／s）
7	ms.powFL／前左駆動モータ出力（－100～100％）
8	ms.powFR／前右駆動モータ出力（－100～100％）
9	ms.powRL／後左駆動モータ出力（－100～100％）
10	ms.powRR／後右駆動モータ出力（－100～100％）
11	7～6 ビット 坂のモード（0,1,2） 5～4 ビット リミットスイッチの状態（0,1,2） 3～0 ビット 坂の累計数
12	総走行距離[cm]の上位 8 ビット
13	総走行距離[cm]の下位 8 ビット
14	バッテリー電圧
15	ジャイロセンサの値
16	7～6 ビット サイドセンサの状態（2b it） 5 ビット （空き）（1bit） 4～0 ビット ポジションセンサの値（5bit）

(4) 1セクタ分 (512 バイト) のバッファの中身



・走行が終了し、ログ記録をやめるとき



### (5) 1 ファイルの中身

[illegible]

### ＜テキスト形式データ＞

走行終了後にテキスト形式でタイムや設定値などの情報を書き込む。

レコードの先頭が-2のとき、次のセクタへ移る。

レコードの2バイト目が0以外のとき、マイクロSDへのセクタ書き込みが失敗した回数を意味する。

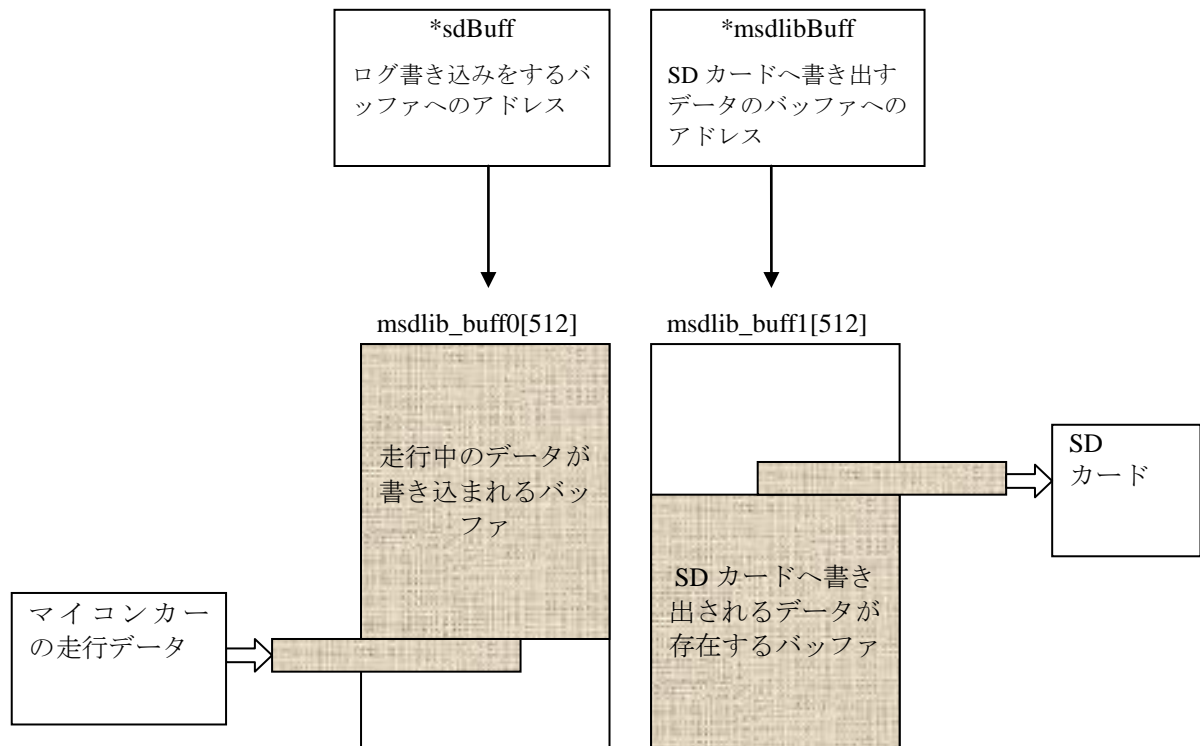
レコードの先頭が0のとき、データ終了。

1 セクタ分 (512 バイト)

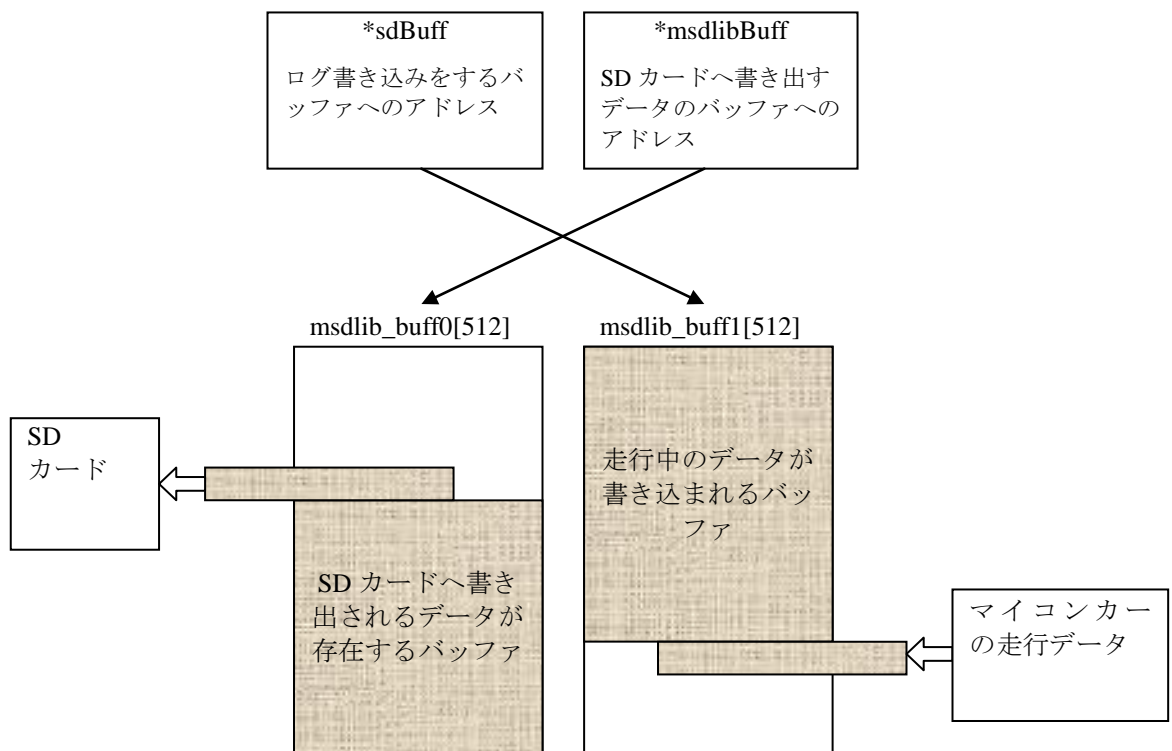
←全体で 1MB

(6) 二重バッファについて

- 5ms 毎の走行データが書き込まれる書き込み用バッファと、SD カードへ書き出すための読み出しバッファが2つ存在し、2つが平行して処理される。
- バッファの実体は static char msdlib\_buff0[512] と static char msdlib\_buff1[512] の2つの配列であり、その先頭アドレスを static char \*sdBuff と static char \*msdlibBuff の2つのポインタで示す。
- 書き込みバッファと読み出しバッファの切り替えは、ポインタの入れ替えで済みます。



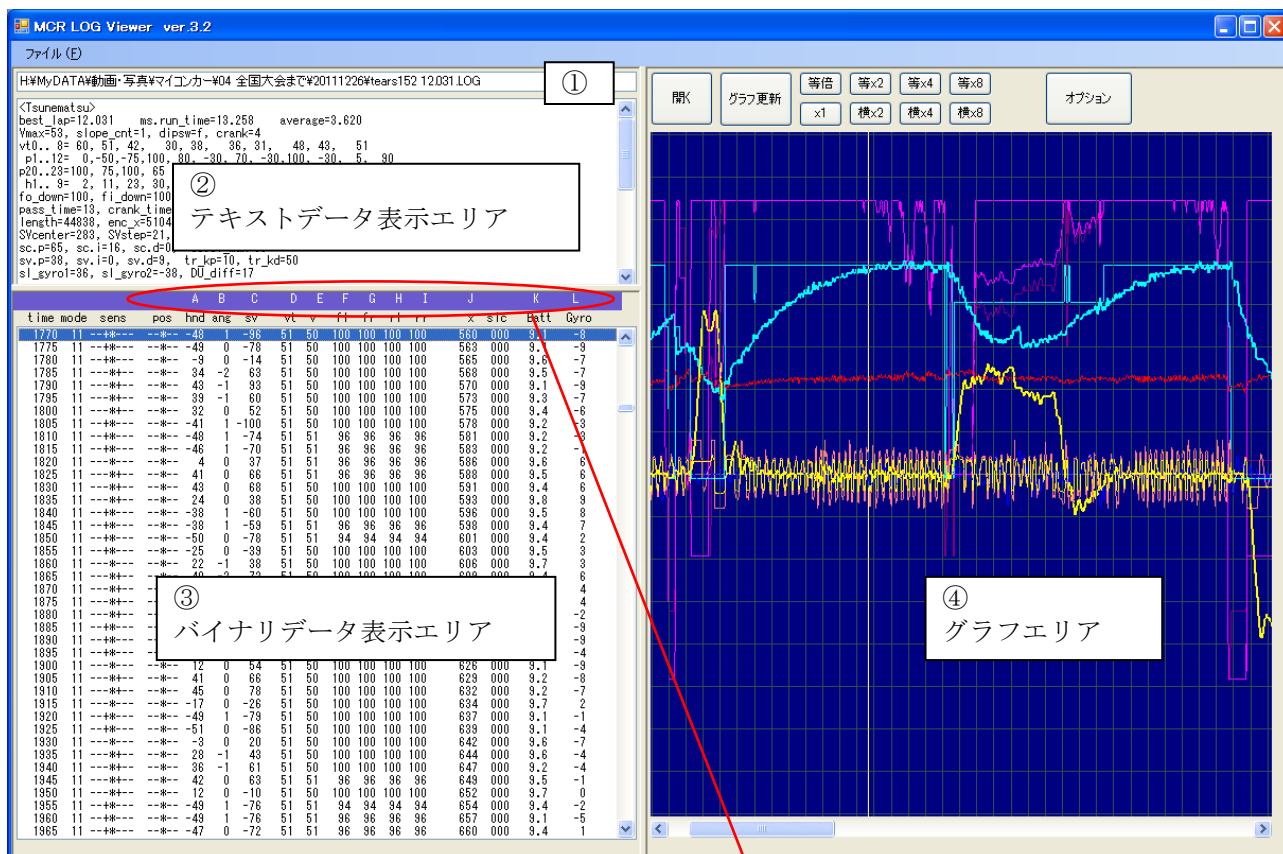
\*sdBuff のバッファが一杯に近くなってくると、ポインタを入れ替える。そのとき、\*msdlibBuff の SD への書き出し処理がまだ終わってなかったら、5ms 後まで入れ替えを待つ。5ms 後も同様だと更に 5ms 後まで待つというように、SD への書き出し処理が終了するまで待つ。その期間の走行ログは破棄される。(実際に破棄されることは余りない)



## 5 ログビューアについて

- ・実行ファイル名：MCRlogViewer.exe
- ・開発環境：Microsoft Visual C# 2010 Express
- ・.NET フレームワーク：.NET Framework 3.5

### 5-1 画面の説明



A～L のアルファベットが対応する

- ① ログファイルのパス
- ② テキストデータ
- ③ 5ms 毎のバイナリデータ (1 レコード 17 バイト)
- ④ ③のグラフ (各項目データとグラフの色の対応関係はオプション画面で確認できる)

「開く」：ログファイルを開く  
「グラフ更新」：グラフ再描画  
「等倍」～「等 x8」：時間軸を等倍に。  
「x1」：全体を 1 画面に納める。  
「横 x2」～「横 x8」：横 2 倍～横 8 倍へ拡大

### 5-2 操作方法

(1) ログファイルを開く方法は以下の 3 通り

- ・「開く」ボタンからログファイルを選択する。
- ・エクスプローラからログファイルをドロップする。
- ・エクスプローラでログファイルをログビューアに関連付けて、ダブルクリックする。

※ハードディスクにコピーされたログファイルの場合、ログビューアに読み込まれた時点で、未記入の余分な部分が削除され、ファイルサイズが小さくなる。



## (2) グラフの拡大／縮小方法

マウスのホイール操作により、横方向のみの拡大／縮小ができる。グラフ上部のボタン操作によっても同じ事ができる。

## (3) グラフのスクロール方法

グラフを拡大表示すると横方向にスクロールできるようになる。スクロールバーでスクロールする以外にも、マウスの右ボタンを押したままドラッグしてもスクロールできる。

## (4) グラフエリアでクリックすると・・・

その地点に対応するデータが③のエリアの行が選択される。逆に③の行をクリックしても④のグラフにも縦ラインカーソルが表示される。

## 5-3 オプション

「オプション」ボタンを押すと、右の画面が表示され、以下の操作ができるようになる。A～Lの各項目のデータは、③の上部に表示されたものと対応する。

- ・各項目のグラフ表示のオン／オフ
- ・スケール変更
- ・色変更
- ・線幅変更

※バッテリー電圧の a,b の値は、AD 変換値から電圧値[V]に変換するためのパラメータ。AD 変換値とテストで計測した実測値を元に Excel で求めた。

表示	スケール	線色	線幅
<input checked="" type="checkbox"/> A	1	-----	1
<input checked="" type="checkbox"/> B	12	*****	2
<input checked="" type="checkbox"/> C	1	-----	1
<input checked="" type="checkbox"/> D	12	-----	1
<input checked="" type="checkbox"/> E	12	*****	2
<input checked="" type="checkbox"/> F	8	-----	1
<input checked="" type="checkbox"/> G	8	-----	1
<input checked="" type="checkbox"/> H	8	-----	1
<input checked="" type="checkbox"/> I	8	-----	1
<input type="checkbox"/> J	1	-----	1
<input checked="" type="checkbox"/> K	30	-----	1
<input checked="" type="checkbox"/> L	4	-----	1

OK  
キャンセル  
全選択  
全解除

バッテリー電圧  $y = ax + b$   
a 0.0143  
b 0.0338

## 5-4 ファイルサイズの圧縮について

ここで言う「圧縮」とは、1MB の空ファイルの先頭から書き込まれている実際のデータ以外の部分を削除することである。5ms 毎に記録されたログファイルは 12 秒の走行で 50kB 前後のファイルサイズとなる。

- |   |                               |                |
|---|-------------------------------|----------------|
| { | ・ハードディスクにコピーしたログファイル          | → 読み込まれると圧縮する。 |
|   | ・マイクロ SD カードのログファイルをそのまま開いた場合 | → 圧縮しない。       |

といった動作のため、まずはハードディスクに日付別のフォルダを作成し、そこにコピーしたものを開くようにすると良い。

ハードディスクにまだ読み込まれていないログファイルが大量にあると、圧縮されないままなのでハードディスクの空き容量を圧迫してしまう。そのときは、エクスプローラで未圧縮のファイルを全て選択し、ログビューアにまとめてドロップすると、一つずつ読み込まれ、順次圧縮されていくので、覚えておくと便利である。

## 5-5 ログデータが示す意味

time	mode	sens	pos	hnd	ang	sv	vt	v	fl	fr	rl	rr	x	slc	Batt	Gyro
1770	11	--+*---	--*--	-48	1	-96	51	50	100	100	100	100	560	000	9.1	-8
1775	11	--+*---	--*--	-49	0	-78	51	50	100	100	100	100	563	000	9.1	-9
1780	11	--+*---	--*--	-9	0	-14	51	50	100	100	100	100	565	000	9.6	-7
1785	11	---*+--	--*--	34	-2	63	51	50	100	100	100	100	568	000	9.5	-7
1790	11	---*+--	--*--	43	-1	93	51	50	100	100	100	100	570	000	9.1	-9
1795	11	---*+--	--*--	39	-1	60	51	50	100	100	100	100	573	000	9.3	-7
1800	11	---*+--	--*--	32	0	52	51	50	100	100	100	100	575	000	9.4	-6
1805	11	--+*---	--*--	-41	1	-100	51	50	100	100	100	100	578	000	9.2	-3
1810	11	--+*---	--*--	-48	1	-74	51	51	96	96	96	96	581	000	9.2	-3
1815	11	--+*---	--*--	-46	1	-70	51	51	96	96	96	96	583	000	9.2	-1
1820	11	---*+--	--*--	4	0	37	51	51	96	96	96	96	586	000	9.6	6
1825	11	---*+--	--*--	41	0	66	51	51	96	96	96	96	588	000	9.5	6
1830	11	---*+--	--*--	43	0	68	51	50	100	100	100	100	591	000	9.4	6
1835	11	---*+--	--*--	34	0	38	51	50	100	100	100	100	593	000	9.8	8

time	スタートしてからの経過時間 [ms]
mode	現在のモード。これを見ると、プログラムが今どこを実行しているかが分かる。  1:起動時 , 2:通常 Start 待ち , 5:LCD_menu , 6:PC_menu , 7:print 8:TestMode , 9:リセット Start , 10:通常 Start 11:直進 , 12:ターボモード , 20:指定速度以下でのコーナー , 30:指定速度以上でのコーナー 50:クランクのクロスライン , 60:右クランク , 70:左クランク , 80:ハーフクロスライン , 90:レーンチェンジ 100:no_line
sens	デジタルセンサの状態。(アナログセンサもデジタルとして 0,1 で表している) * : デジタルセンサ + : アナログセンサ
pos	ポジションセンサの状態。
hnd	トレースモードのときは (左アナログセンサー右アナログセンサ) の値 ハンドル角度指定モードのときは指定ハンドル角度 [°]
ang	現在のハンドル切れ角 [°]
sv	サーボモータの出力 [%]
vt	指定速度 [×10 m/s] この値が -1 のときは、プログラムで直接駆動モータを制御している場合であり、0 以上のときは、駆動モータの出力は自動で計算されている。
v	現在の速度 [×10 m/s]
fl, fr, rl, rr	駆動モータの出力 [%] fl:左前、fr:右前、rl:左後、rr:右後
x	進んだ距離 [cm]
slc	s : 現在の slope_mode。0=通常走行、1=坂検出後の空走状態、2=減速状態。 l : 坂検出センサの状態 c : スタートしてから現在までの、坂を検出した回数。
Batt	バッテリー電圧 [V]
Gyro	ジャイロセンサの状態。 この値を参考にして、坂検出のしきい値を設定する。 sl_gyro1=上り坂の設定値 sl_gyro2=下り坂の設定値

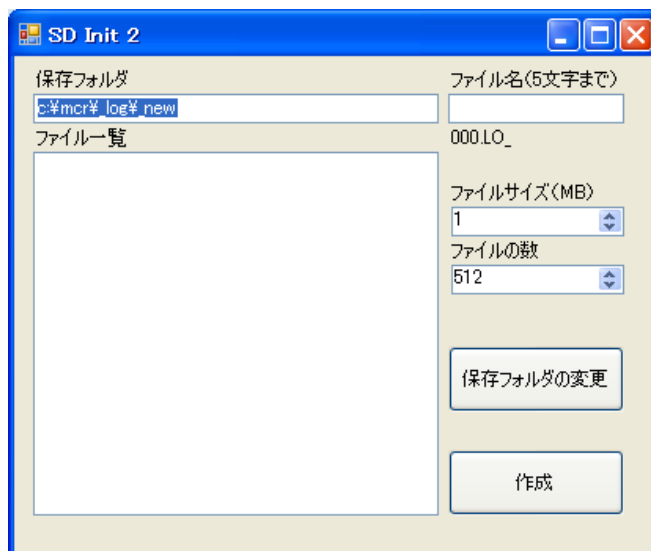
## 6 空のログファイル作成ソフトについて

マイコンカーでは先頭アドレスから順次闇雲にデータを書き込んでいくため、予めファイルが整然と並んだ状態となっている必要がある。そのためカードをフォーマットした直後に空のファイルを作成しておく必要があるが、その際使用するソフトがこれである。やっていることは、単に全て 0 で埋められた指定サイズの空ファイルを指定数個作成しているだけである。

- 実行ファイル名：SDInit2.exe
- 開発環境：VisualC#2010Express
- .NET フレームワーク：.NET Framework 3.5

### <使用手順>

1. microSD カードを PC でフォーマットする。(FAT16)
2. その直後に本アプリケーションを起動する。
3. 「保存フォルダの変更」ボタンにて、microSD カードのドライブを指定する。
4. ファイル名に半角 5 文字以内で任意の文字列を付ける。
5. ファイルサイズは 1MB で、ファイルの数は 512 個とする。  
(FAT16 では、ルートディレクトリに作成できるファイル数は 512 個まで)
6. 作成ボタンを押す。(終了まで 1 分以上の時間がかかる)
7. 終了メッセージが出れば作成終了。



SDInit2.exe の画面