

# LocoNet Personal Edition の解説

Loconet? は Digitrax 社のトレードマークです。

この内容は桂庵の解釈です。

Draft Version 0.5

## 1. はじめに

ここでは私が理解した LocoNet の要求規格について、説明します。

ただし、私は DCC システムを持ったことも使用したこともないので、とんでもない勘違いをしている可能性が大です。原文と一緒に読んで下さい。

ここに記述した内容はすべて Digitrax 社の所有物であり、原文

( <http://www.digitrax.com/pdfdocs.php#loconetpe> [LocoNet Personal Edition](#) ) には個人での使用のみと明確に書かれております。取り扱いには注意願います。また、この内容を使用する場合は、原文を読み、使用して良い内容をご理解の上、使用願います。説明は原文と異なる場合がありますと思いますが、ご容赦下さい。間違い等があれば、ご連絡ください。

## 2. LocoNet とは

まず、ネットワークとは何か考えて見ましょう。

ちょっと抽象的ですが、ある所からある所 ( 3ヶ所以上でもよい ) へ、物とか情報を流す仕組みをネットワークといいます。

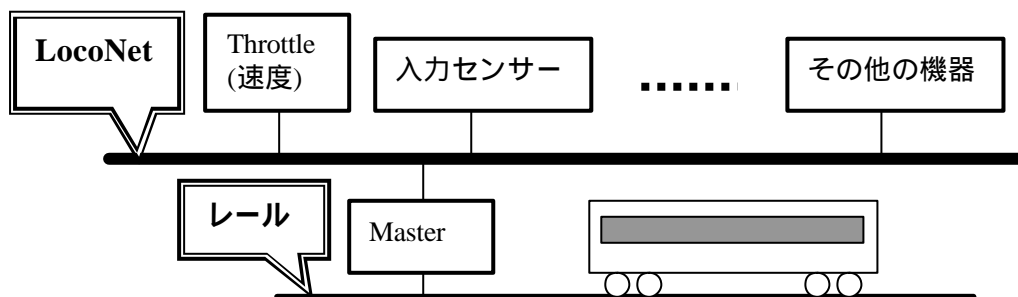
たとえば、コンピュータネットワークとは、複数のコンピュータ間の情報のやり取りを行いますし、物流ネットワークは、宅配便等のやり取りを行っています。今は、色んなネットワークがありますね。

LocoNet とは、Digitrax 社が開発した鉄道模型を制御するため、色々な機器同士をつなぐために決められたネットワークです。

LocoNet 上にある機器 ( ネットワークに接続されている機器はそのネットワーク上にあると言います ) は、LocoNet を用いて、情報のやり取りを行います。

下の図は、この規格に描かれている全体を書いた図 ( 図 1 ) です ( と思います ) 。

図 1



この LocoNet 上に色々な機器から Master に向けて情報が出ます。情報を受け取った Master は情報の解析を行い必要な DCC パケットをレールに出力します。

さて、そうすると LocoNet 上にたくさんの情報が流れて行きます。また、もし、Throttle と入力センサーが同時に情報を Master に出そうとするとどのような事になるでしょう。その情報は混ざってしまい正しい情報を Master に送ることは出来ません。そうなると車両等を制御できなくなります。

そこで LocoNet は CSMA/CD と言う、ネットワーク制御方式を採用して、うまく色々な情報がちゃんと送れるようにしています。CSMA/CD とは Carrier Sense Multiple Access with Collision

Detection の頭文字です。LocoNet Personal Edition の原文を見ると CSMA/CD に関して勘違いしているように見受けられます。(CD を Carrier Detection としている。)簡単に動作を説明すると、ある機器が情報を出そうとする時、まず、LocoNet 上に情報が送られているかいないかを確認する。そして、送られていなければ情報を送り、もし、すでに送られているのであれば、送るのを待ちます。どのくらい待つかは後で説明します。

たまたま、同時に2つ以上の機器が情報を出した場合は、情報の衝突 (Collision) が起こり、情報を出した機器は、衝突を検出し、ジャムと呼ばれる信号を出し、現在情報の衝突が起こった事を、他の機器に知らせます。そして、ある時間待って、情報の再送をします。

以上が CSMA/CD 方式の基本です。ところが、情報を出す最初に、情報の確認のため、Carrier Sense を行いますが、LocoNet の仕様上、Carrier は存在しないのです。通常、Ethernet では、データ"0"、"1"は搬送波と呼ばれる信号により情報を運ばれます。そのため搬送波 (英語で Carrier) を検出することにより、情報の確認が出来るのですが、LocoNet ではそれが出来ません。また、LocoNet 仕様では衝突検出(Collision Detection)に関しても、もし、複数の機器が同じタイミングで情報を出した場合、衝突検出が出来るのは、情報を出した機器のみとなります。

と言うことでちゃんとした CSMA/CD 仕様とは言いがたいのですが、少なくとも、情報を出力する前に、LocoNet 上の情報 (Carrier ではない) の確認を行う。また、衝突検出を行ったら、衝突情報を出す。再送に関して、待ち時間があると言うことで、多重アクセス(Multiple Access) が可能と言うことになってます。仕様としては、もう一歩な仕様ですが、実際には、安くできると思います。原文にもあるようにコストはホビーにとって非常に重要な事です。

### 3. 用語について

この規格内での用語について説明します。原文における用語の使い方は、若干統一性がなく、また、用語の定義もないため、私の独断で説明させていただきます (汗)。

#### I. Master(マスター)

マスターは、各機器からの情報を LocoNet を通して、取得、保持し、DCC パケットを生成し、実際にレール上のデータを作り出す。また、周期的に DCC パケットを出すことも行う。保持している情報により、車両等に適切な制御を行う。

#### II. SLOT (スロット)

マスター内に保持する車両制御用データの割り当ての事。たとえば、車両 A には 55 番が SLOT 番号として割り当てられ、SLOT55 内に車両 A の現時点の情報 (速度とか) を保持する。車両 A の制御は、SLOT55 に対して行う。

#### III. SINGLE END (シングルエンド) と HYSTERESIS (ヒステリシス)

シングルエンドとは、グランドと信号線の 2 本の線で伝送する方式で、正の電圧のみを持ちます。ある閾値以上の電圧で"1"、その電圧以下で"0"を表します。雑音等での誤検出をさけるため、ヒステリシスを設定する場合があります。ヒステリシスとは、いったん"1"と検出した場合、その後"0"となるためには、閾値電圧より下の電圧で検出する事を言います。反対に"0"から"1"の場合は閾値より高い電圧が必要となります。LocoNet の場合、閾値は 4V ですが、ヒステリシスを 1V と設定しているため、"0" -> "1"になるためには 4.5V 以上の電圧が LocoNet 上に必要となっています。また、"1" -> "0"になるためには 3.5V 以下です。

その他の用語については、説明の中で行います。

### 4. 技術仕様

#### I. 形状、ピンレイアウト、線材

フル装備 LocoNet にもちられるコネクタは USOC RJ12 6pin 仕様を用います。

ピンレイアウト(RJ11/6) ->注参照

1. RAIL\_SYNC-                   white
2. SIGNAL GROUND
3. LOCONET+
4. SIGNAL GROUND
5. LOCONET-
6. RAIL\_SYNC+                   blue

LOCONET+-は同相のため、ループバック接続でも大丈夫ですが、RAIL\_SYNC+-は逆相のため、ループバック時の接続はしてはいけません。

ケーブルの太さは 26AWG です。ペアー線（ツイステッドかも？）あるいは、電話に用いられる平行リボン線（私は正式な名称は知りません）が用いられます。

注：原文では、ピンレイアウトの部分で RJ11 と記載されてます。筆者はモジュラーコネクタの知識がないため、どなたか調べて下さい。

## II. 電気特性

LOCONET+-はワイヤードオア接続が出来るような構成となっています。

注：ここでも RJ11 と出ていますが、正しいか間違いかは？です。また、原文では、LocoNet? がワイヤードオアとしていますが RAIL\_SYNC +-はワイヤードオアになり得ないので、LOCONET+-ラインのみワイヤードオアが可能としています。

LOCONET+-における立ち上がり、立下り速度は以下の通りです。

立ち上がり 0.35V/  $\mu$ S 以上 (2.0-7.0 V)

立下り 0.75v/  $\mu$ S 以上

ケーブルは最大 600m まで延ばせます。また、一対一接続では 300m までです。ケーブルで発生する容量は 68000pf と見積もっています。

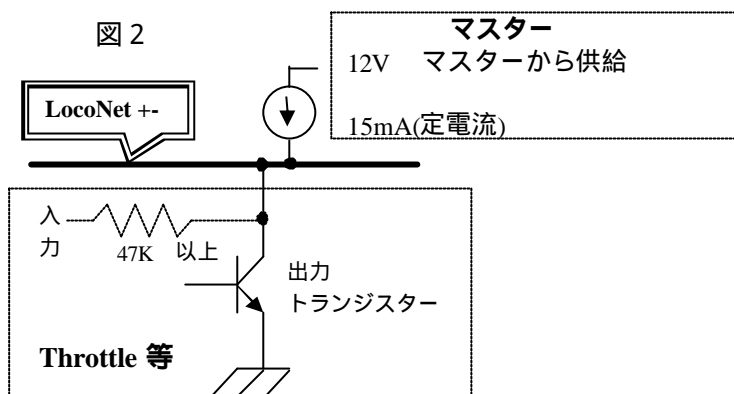
### LOCONET+-規格

- A) 論理値 “1” (MARKとも呼ぶ)  
4.0V 以上 (注：用語ヒステリシス参照)
- B) 論理値 “0” (SPACEとも呼ぶ)  
4.0 V以下 (注：用語ヒステリシス参照)
- C) ヒステリシス  
4Vを中心に 1V
- D) 最大電圧と通常電圧  
最大電圧は 24V, 通常電圧は 12V  
注：24V, 12Vは”1”の場合である。
- E) 入力インピーダンス  
各機器の入力インピーダンスは 47k 以上
- F) 出力形態  
各機器の LocoNet への出力形態はオープンコレクタです。”0”出力時(トランジスターON)の最大電圧は、シンク電流（出力段トランジスターが吸い込める電流）50mA 時、1.6V 以下です。”1”出力時（トランジスターOFF）は 35V 以上です。  
注：”1”出力時の 35V の定義は、基本的にトランジスターOFF 時におけるコレクターエミッタ間の OFF 抵抗値を規定しています。G)のように、”1”の時、15mA の定電流が流れてくるので、出力トランジスターの最小 OFF 抵抗は、35V/15mA =

2.3K となります。この値はトランジスタの OFF 抵抗としては、十分低い抵抗値です。従って、ここでの規定は出力段トランジスタのコレクタ最大定格電圧が 35V 以上と言っていることとなります。

G) プルアップ電流源

オープンコレクタにおけるワイヤードア接続では、プルアップする電流源が必要です。LocoNet にひとつのプルアップ電流源があれば良いので、通常はマスター内部に電流源を設けます。電流源の値としては 12V から 15mA です。



上図 2 に電気仕様の概念図を示す。マスターは LocoNet+- ラインに対して 12V で 15mA の電流を供給します。また、Throttle 等の機器は、出力としてオープンコレクタ出力、入力として 47K 以上を持つ必要があります。

RAIL\_SYNC+-

- A) RAIL\_SYNC+- は少ない電流しか取り出せないが、レール上に出力される DCC パケットデータと同じ出力を出している。この信号は、差動入力を受信し、増幅されレールに出力されても良い。注：マスターが十分な電力をレールに供給できない場合の事だと思ふ。
- B) RAIL\_SYNC+- は最大 15mA の電流を供給できる。最低出力電圧は 7V (注：15mA 時であろう)。無負荷の場合は 12-26V であり、26V 以上は出力してはならない。注：原文には、Throttle Jack について記述があるが、私は全く知識がないので、説明できません (汗)。
- C) ステレオ 3 ピンジャックを用いる場合は、SIGNAL GROUD はスリーブ、LOCONET+- はチップ、そしてスリーブに電源を接続して下さい。また、電源は最大 20mA の電流源として下さい。これは差し込み時ショート危険性があるためです。

III.

基本仕様

シリアル非同期式、半 2 重、16.457K ボー (60  $\mu$  s/bit, 16.666Kbit/s)  
+- 1.5% 以内

スタートビット数	1
データビット数	8
ストップビット数	1
パリティ	なし

IDLE 時 (Loconet 上に信号がない時) Mark ("1")

## 送信制御方式

送信制御について、この仕様書においては、CSMA/CD方式となっておりますが、若干勘違いと思われます。

(注：モデム(DCE)とイーサネットを混乱している感じ。)

取り合えず、解説を進めてみます。

前述したが CSMA/CD 方式の基本から。

### A) CSMA/CD 方式の基本

LocoNet は基本的に CSMA/CD 方式の考え方をベースとして、送信制御方式が決められています。CSMA/CD 方式の基本はデータを送信しようとしている機器において

1. 送信前にネットワーク上にデータがあるかないかの検出。  
データを送信したい機器のみ送信前にネットワーク上のデータの検出を行う。
2. 送信中にデータの衝突があるかないかの検出。
3. 送信不可時、衝突時の再送タイミング制御。  
です。

### B) ネットワーク上のデータ検出

上記 A)-1 のネットワーク上にデータがあるかないかの検出について、原文では CD (Carrier Detection) となっておりますが、本来は CS (Carrier Sense) の部分です。LocoNet にはイーサネットやモデムと異なり Carrier がありません。また、IDLE 時は "1" (Mark) 状態で、"1" データ出力時と区別できません。そのため、LocoNet においては "0" (Space) を検出した場合に、ネットワーク上にデータがあると認識します。"0" はスタートビットに使用されており、1byte のデータをネットワークに出すときは必ず存在します。(注：ストップビットは "1" である。そのため、どのようなデータであっても 1byte 送信時には "0", "1" は存在する)

### C) 送信データの衝突の検出

上記 A)-2 の送信データの衝突検出については、送信した機器が責任を持って検出しなければいけません。もし、全く同じタイミングで複数の機器がデータを送信した場合、LOCONET+ ラインをモニターしていても、まず送信データの衝突は検出できません。可能性としては、送信バイト数が異なる、または、チェックバイトの値が異なるで検出出来るかもしれませんが、検出までの時間がかかってしまいます。従って、データを送信する機器は、データを送信しながら、同時に LOCONET+ ラインを読み込み、自分の出力したデータと比較を行い、送信データの衝突検出を行うことが必要です。出力したデータと読み込んだデータが異なっている時、衝突と判断します。

衝突の検出はビットごとあるいはバイトごとで行います。

注：もちろん衝突の検出はビットごとで行われる方がいいのですが、シリアル半 2 重制御を LSI ですべて操作している場合はビットごとの検出は難しいと思います。(他のポートを使用すれば出来る可能性はある)

### D) データ衝突時の再送タイミング制御

上記 A)-3 再送タイミング制御です。データを送信、再送するためには、ネットワーク上のデータ検出、送信データの衝突検出後の処理を決め、その処理に従って、各機器はデータの再送を試みます。

### E) ネットワーク上のデータ検出時のタイミング

上述のように、データを送信したい機器が、送信前に "0" (Space) を検出した時、データが LocoNet 上にすでにあると検出します。データを送信したい機器は、すでに別の機器がデータを送信しているため、送信を待たなければなりません。この

待つ時間を待機時間といいます（原文では CD BACKOFF time）。この待機時間は 1.2 ms(20 ビット時間)です。また、この待機時間に加えて、各機器につけられている優先順位による待ち時間（原文では PRIORITY delay）を加えた時間を待ち、再び LocoNet 上のデータ検出を行い、もし、データがなければ、データを送信できます。

注：原文には書いてはないのですが、もう少し考えて見たいと思います。LocoNet に送り出すデータは最低 2 バイトです（メッセージフォーマットを参照してください）。従って、すべての機器がこの規格に従うとすると、バイト毎の送信間隔は、約、 $600 \mu s(10 \text{ ビット秒} \text{ スタートビット}+8 \text{ ビット}+\text{ストップビット})+1.2\text{ms}$  となってしまいます。また、第二バイト送信時、他の機器がデータ送信を先に出来たたとすると、送信バイト列がめちゃくちゃになってしまいます。すなわち、いったん送信できる状態になった機器はこの待機時間 + 優先順位による待ち時間には関係なく、最後のバイトまで送信できる優先権を持っていると解釈できます。原文には、ストップビットに続いてスタートビットを送出できるとは書いてありますが.....

注：待機時間の開始について

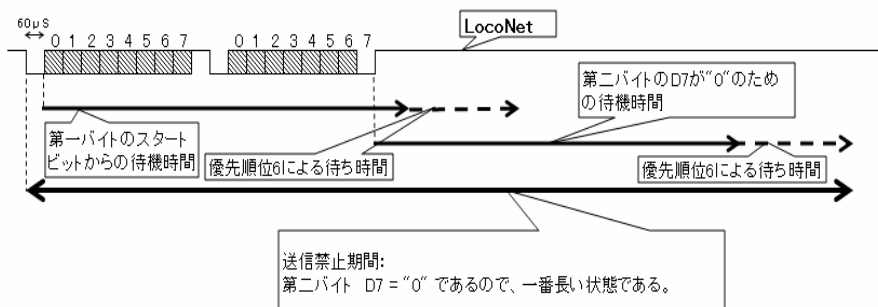
原文は、実は明確のようで明確でないのです。まず、LocoNet 上のデータが"0" になった時に開始するとしている部分と最後の"0"レベルからと言う記述があります。この規格の作者の意図を考えると、後者、すなわち、最後の"0"レベルから"1"になった瞬間がこの待機時間のスタートと考えるのが妥当と思っています。しかし、LocoNet へのデータの送信禁止（検出含め）を規定するためには、待機時間の定義としては、分かりませんね。と言うわけで私の独断と偏見で、この規格の意図に沿ったと思われる内容で定義してみます。

### 送信禁止期間の定義

この期間中には、すでにデータを送信している一つの機器を除き、すべての（P Cは違うようです。P Cアクセス参照）機器は、LocoNet にデータを送信してはならないと言うことです。

1. LocoNet 上のデータが"0"の時
  2. LocoNet 上のデータが"0"から"1"に変化する時から 1.2ms の期間（待機時間）と優先順位による待ち時間の和の期間
  3. 2.の期間中に再び"0"から"1"に変化した場合、待機時間再設定となり、この時点から  $1.2\text{ms} + \text{優先順位による待ち時間}$ 。
- 以上がこの規格の作者が意図した内容かと思っています。

図3は、2 バイトのデータを出す場合の送信禁止期間を示しています。最長を考え、第二バイトの D7 を"0"として考えてます。送信禁止期間は、第一バイトのスタートビットの最初から、第二バイトの D7 ("0")の終わりから始まる待機時間、優先順位による待ち時間までです。送信禁止期間のはじめは Loconet+-上のデータが"1"から"0"になる時です。各機器は、データが"1"から"0"になった後  $3 \mu s$  以内に、送信禁止期間である事を検出できなければいけません。 図 3



### 優先順位による待ち時間

優先順位についてこの原文では明確には定義されていません。定義されている内容は、

- ・優先順位による追加待ち時間は最大 20 ビット時間 (1.2ms) である。
- ・マスターが一番優先される。(優先順位による待ち時間が 0)
- ・入力センサー等の機器の優先順位は 6 以下。

注：原文からすると、

- ・優先順位第一位はマスターで、その PRIORITY は 0 である。
- ・PRIORITY の数字は大きいほど、優先順位が低い。
- ・PRIORITY の数字が待ち時間の一位ビット時間になる。  
たとえば、PRIORITY 6 の機器は  $1.2\text{ms} + 6 \times 60 \mu\text{s} = 1.56\text{ms}$  の待機時間となると私は理解しています。

### 待機時間の精度

この規格上では  $360 \mu\text{s}$  以内と規定されています。また、CR(コンデンサー、抵抗)によるワンショットでの時間も可能と言う記述があり、この場合は  $360 \mu\text{s}$  の精度はコストがかかると思います。従って、精度としては  $360 \mu\text{s}$  ですが、実際はもう少し、精度は出ていないと思います。

### 再送

データを送信しようとしたら、送信禁止期間であった場合、送信を待ちます。送信禁止期間を過ぎ、送信しようとしたら、再び他の機器(マスター等)が先に送信を始めていた場合また待たなければなりません。この再送は 25 回まで出来ます。25 回、送信しようとして無理であったらあきらめましょう。ただ、そうすると、優先順位の低い機器は、いつまでたっても送信できないかもしれません。そのため、送信できなかった場合、優先順位を上げる事が出来ます。一回送信できなかった毎に、優先順位を一つあげます。そうすると次に、データを送信しようとする時、優先順位による待ち時間を 1 ビットの時間 ( $60 \mu\text{s}$ ) 短くできます。ただし、マスターの優先順位を超えることは禁止されています。

注：この規格では、上述したように、優先順位については明確になっていません。通常の機器(Throttle 等)の最上位優先順位は 1 であると思います。また、原文には記述がありませんが、一旦送信できた場合は、優先順位を元に戻すことが必要と思います。

## F) データの衝突時のネットワークタイミング

C) でデータの衝突について説明しました。ここでは、データの衝突を検出した時の機器が行うべきことについて、説明します。

### 1. データを送信していた機器(衝突を検出した機器)

データの衝突を検出した機器は、15 ビット時間 ( $60 \mu\text{s} \times 15 = 900 \mu\text{s}$ ) の間、"0"を連続して出力します。この動作をブレーク(Break)と言い、他の機器にデータの衝突が起こった事を知らせます。

注：ブレーク出力後の動作については原文では記述がありません。ここでは、私の理解で説明します。

データの衝突を検出した機器は、ブレークを出力することにより、LocoNet を "0" にします。この時、送信禁止期間が自動的に発生し、送信禁止期間を過ぎたら、データの再送を試みます。ブレークが終了した時点から、待機時間と優先順位による待ち時間を待ち、データの再送動作になります。データ検

出時と同様な動作になります。

2. データを受信していた機器  
ブレークを検出した受信中の機器は、すでに受信したデータを含め、データを破棄します。その後、再び opcode から、データを受け取ります。

#### PC の LocoNet へのアクセス

PC では、Serial Port が使えます。また、PC であれば、LocoNet 上のデータをモニターしていることにより最後の送信バイトを検出でき、送信禁止期間を待つ必要はなく、最後のバイトの送信終了後、すぐに、LocoNet にアクセスできます。この場合マスターより優先順位が高い状態でのアクセスになります。マスターは BUSY コードを出力でき、この BUSY コードは、送信禁止時間の開始コードです。PC は、このコードを無視しても良いとなっています。

注：BUSY コードを PC は無視すべきではないと思っています。マスターが BUSY を出す場合は、マスター内での処理が忙しく、LocoNet 上のデータを読み込めない可能性がある為です。もし、PC そのものがマスターであれば、この内容はなしです。

複数の PC が LocoNet にアクセスする場合は、20 ビット時間の待機時間をうまく、各 PC に分割し、データの衝突を避けて下さい。衝突検出は必ず実装して下さい。

注：PC でのアクセスも他の機器と同じようにすればいいと思いますが、原文ではこのようになっています。ただし、待機時間である 20 ビット時間の分割の仕方に気をつけないといけません。すでに LocoNet 上に送信されている最後の送信バイトを受信後、10 ビット時間内でそれぞれの PC に対して、待ち時間を設定するのが妥当な設定と思います。

#### LocoNet への接続と切り離し

各機器は、100ms 以上続けて"0"であった場合、LocoNet から切断されたと認識する必要があります。再び、その機器が LocoNet へ接続された場合、LocoNet へのアクセスは 250ms 以上待つ必要があります。これは、その機器のパワーオン時も要求される内容です。この待ち時間のことをスタートアップ待ち時間 (STARTUP backoff time) と言います。

再接続した機器は、機器の内部情報を初期化する必要はありません。ただし、マスター内部の SLOT 情報と合っていることを確認する必要があります。SLOT 情報と合っていることを確認後、SLOT を使うことができます。

注：切断後の再接続時は、その機器が SLOT データを読み出し、機器内部に保存すべきと思っています。

マスターは、ある機器からパージ(PURGE)時間内に S L O T をアクセスされなかった場合、その機器は切断されたと認識し、その SLOT はコモン (COMMON)状態にします。

通常のパージ時間は DT200 の場合約 200 秒です。ユーザが Throttle 等の操作をしなくても、機器は 100 秒ごとに SLOT をアクセスすることが好ましいです。

注：マスター自身で、LocoNet に接続されている機器を確認には行きません。各機器は自分でマスター内の SLOT に定期的にアクセスする必要があります。



#### IV. メッセージフォーマット

##### メッセージの基本構造

メッセージは複数のバイト列で構成されます。最低バイト列は2バイトであり、4バイト、6バイト、可変バイトメッセージがあります。最初のバイトは命令コード (op code) です。

命令コードは最上位ビットが必ず"1"であり、命令以外のバイトは最上位ビットを"0"にしなければなりません。

2バイト命令	<op code> <checksum>
4バイト命令	<op code> <arg1> <arg2> <checksum>
6バイト命令	<op code> <arg1> <arg2> <arg3> <arg4> <checksum>
可変長命令	<op code> <n> <arg1> <arg2> <arg3> .....<checksum>
<op code>	命令コードバイト
<n>	メッセージの総バイト数を示すバイト
<arg>	引数 (命令に付随する命令に関連した情報) バイト
<checksum>	エラー検出バイト、通信のエラーを検出するためのバイト。

##### checksumの生成方法

checksum バイトを除くすべてのバイトの排他的論理和を取る。

その結果を1の補数にし、checksum バイトとする。

受信時は checksum バイトを含めすべてのバイトに対して、排他的論理和を取った結果が 255 (10進) あるいは FF(16進)であれば、データは正常に受信できた事とします。

注：通常 checksum の場合は、各バイトの足し算 (sum の意味) で checksum バイトを生成します。排他的論理和で同じ結果となるかは分かりませんが、原文の用語使用の間違いであると思います。ここではこのまま checksum あるいは<CHK>と記述します。ソフトを実装する人は気をつけて下さい。

op code

D7 D6 D5 D4 D3 D2 D1 D0	
1 0 0 F D C B A	2バイトメッセージ
1 0 1 F D C B A	4バイトメッセージ
1 1 0 F D C B A	6バイトメッセージ
1 1 1 F D C B A	可変長メッセージ

D5,D6 によりメッセージバイト数の確認ができる。

##### SLOTについて

マスターは最大 120 個の SLOT を持ち、各 SLOT は車両、コントロールに関連した 10 バイトのデータ領域からなります。SLOT は車両にアサインされます。SLOT 番号 (アドレス) 120-127 はマスター、システムが使用します。アドレス 124 は DCS100 が使用しており、通常の内容と異なります。

注：凡例内のスロットの説明と矛盾しているような気がしてます？

##### メッセージの基本的な使い方 (動かしたい車両の設定)

DCC システムは複数の車両を同時に制御できるシステムです。そのため、動かしたい車両を自分のコントローラ (Throttle, スロットル, 速度制御機器) で使えるようにしなければなりません。 のメッセージの説明も参考にして下さい。

- A) 動かしたい車両のデコーダアドレスを含む OPC\_LOCO\_AD 命令メッセージを、マスターに送ることにより SLOT の確保ができます。このメッセージは OPC\_LOCO\_ADR 命令

- + 車両デコーダの上位 7 ビットアドレス (ADR2)
- + 車両デコーダの下位 7 ビットアドレス (ADR)
- + エラー検出バイト

です。

- B) マスターから、SLOT 番号の応答があります。この応答には、この車両の状態（現在の速度等）が含まれています。なにも問題がなければ、この SLOT を使用しますとマスターに通知します。これで設定は終了です。
- C) マスターから来た SLOT 番号を用いて、車両の制御が出来ます。例えば、速度を変えたい時は、OPC\_LOCO\_SPD 命令を用いて OPC\_LOCO\_SPD 命令
- + SLOT 番号
  - + 車両の速度
  - + エラー検出バイト
- メッセージであなたの車両は速度を変えます。

#### SLOT の確保（車両アドレスの選択）

車両制御の基本である SLOT の確保について説明します。車両のデコーダアドレスを OPC\_LOCO\_ADR 命令でマスターに送ることにより、何も問題がなければ SLOT の確保ができます。車両デコーダアドレスには 2 種類あります。ひとつは、標準の 7 ビット NMRA アドレスで、もうひとつは 14 ビットアドレスです。14 ビットアドレスは DCC 拡張パケットに対応しているアドレスで現在のほとんどのデコーダは 14 ビットに対応しています。（詳細はデコーダマニュアルを確認願います） マスターは OPC\_LOCO\_ADR 命令を受け取った時、ADR2 の内容をしらべ、もし、ADR2 が 0x00 でなければ 14 ビットアドレスを持ったデコーダとして処理をします。（DT200 は 14 ビットアドレスに対応していません、）

マスターが OPC\_LOCO\_ADR 命令を受け取ると、マスターは SLOT に関する情報を含め応答します。この応答の中に STAT1 と呼ばれる SLOT 状態を示す情報があります。SLOT を要求した機器はこの STAT1 の内容を確認しなければなりません。STAT1 の内容は各種ありますので、の凡例を見てください。もし、STAT1 の内容が共有 (COMMON) であれば、OPC\_MOVE\_SLOT 命令を使用して、この SLOT を使用中 (IN\_USE) にできます。他の人がこの車両を制御しようとして SLOT の確認をした場合、STAT1 が IN\_USE 状態であるので、他の人はこの SLOT がすでに使用中であることが分かります。

注：マスターは他の人がこの SLOT を使用することを禁止してはしません。

#### 連結（CONSIST）動作について

連結 (CONSIST) とは、例えば、機関車を重連で操作したい場合、一つの機関車 (SLOT) に対して、速度等の操作を行うと、もう一方の機関車も同じように動く動作のことを言います。（連結の設定については DT400 等のマニュアルを参照して下さい。） マスターからの応答の一部である STAT1 内に CONSIST (連結) 状態が含まれています。もし、マスターからの STAT1 内で D6 (SL\_CONUP) = "1" であった場合、この SLOT を使用してはいけません。この SLOT は他の SLOT に連結しており、他の SLOT でこの車両 (SLOT) を制御します。この車両 (SLOT) を制御したい場合は、まず連結の切り離しを行った後、この車両 (SLOT) を制御します。

注：STAT1 D6 は SL\_CONUP である（凡例参照してください）。SL\_CONUP が "1" の場合ここでは UP\_CONSISTED と記述してありますが、凡例内では uplinked と記述されていて、かつ、ここでは D6 = "1" の場合は、この SLOT は使用不可としていますが、凡例内では D6/D3 = 10 の場合は TOP SLOT と説明しています。すなわち、この SLOT

は連結の最上位 SLOT となり、制御できるはず..... ?????? 私の勘違いかも (汗)

DISPATCH (ディスパッチ)について

DISPATCHは、英単語としては、”急いで送る”と言う意味です。LocoNetでは、設定した車両(SLOT)の制御権を他の機器に渡すことを意味しています。BT20等、それ自身で車両(SLOT)の設定が出来ない機器に対して、DT400等で車両の設定を行い、BT20にその車両(SLOT)の制御権を渡すことが出来ます。DISPATCHは OPC\_MOVE\_SLOT 命令を用いて設定が出来ます。(メッセージの説明を参照して下さい)

メッセージの説明

凡例( )を参照しながらメッセージを見て下さい。

なお、各命令は

命令名、op dode(16進)、

メッセージ列                      この命令のメッセージ列

この命令の内容説明

発行                      この命令を発行する機器。

受取                      この命令を受け取る機器

応答                      この命令を受け取った機器からの応答  
で説明しています。

A) 2 バイト命令

<op code>, <checksum>

OPC\_IDLE 0x85

メッセージ列                      <0x85>, <CHK>

レール IDLE 要求命令。緊急停止状態にし、レールの状態を待機状態とする。

発行                      マスター以外の機器

受取                      マスター

応答                      なし

OPC\_GPON 0x83

メッセージ列                      <0x83>, <CHK>

システムパワーオン要求命令。DCC パケット出力可能状態になる

発行                      マスター以外の機器

受取                      マスター

応答                      なし

OPC\_GPOFF 0x82

メッセージ列                      <0x82>, <CHK>

システムパワーオフ要求命令。DCC パケット出力不可状態になる

発行                      マスター以外の機器

受取                      マスター

応答                      なし

OPC\_BUSY 0x81

メッセージ列                      <0x81>, <CHK>

マスター-BUSY宣言命令。各機器は待機時間をスタートする。

(又は再スタートする)

発行                      マスター

受取                      マスター以外のすべての機器

応答 なし

B) 4 バイト命令

<op code>, <arg1>, <arg2>, <checksum>

OPC\_LOCO\_ADR 0xBF

メッセージ列 <0xBF>, <ADR2>, <ADR>, <CHK>

車両アドレス(ADR2,ADR)にアサインされた SLOT 番号要求命令。

マスターは ADR2, ADR の車両を探し、存在していた場合はその SLOT 番号(応答では SLOT1)を含めた SLOT の状態を返す...応答 1。もし、存在おらず、SLOT に空きがあった場合初期化された SLOT の状態を返す...応答 2。もし、SLOT に空きがない場合、OPC\_LONG\_ACK 命令をフェール状態にし戻す..応答 3

発行 マスター以外の機器

受取 マスター

成功応答 1,2 <OPC\_SL\_RD\_DATA>, <0x0E>, <SLOT1>, <STAT>, <ADR>, <SPD>  
<DIRF>, <TRK>, <STAT2>, <ADR2>, <SND>, <ID1>, <ID2>, <CHK>  
応答 1 と 2 の違いは SLOT1 が存在していたか、新しく設定されたかの違いである。OPC\_SL\_RD\_DATA は可変長命令である。

失敗応答 3 <OPC\_LONG\_ACK>, <0x3F>, <0x00>, <CHK>

OPC\_SW\_ACK 0xBD

メッセージ列 <0xBD>, <SW1>, <SW2>, <CHK>

SW 動作要求命令 ( 応答あり )。<SW1>, <SW2> の内容で SW 作動させる。

発行 マスター以外の機器

受取 マスター

成功応答 <OPC\_LONG\_ACK>, <0x3D>, <0x7F>, <CHK>

失敗応答 <OPC\_LONG\_ACK>, <0x3D>, <0x00>, <CHK>

この応答は DCS100 の場合である。失敗時は DCS100 内部処理の問題  
注：他のマスターによる応答については記述がない。

OPC\_SW\_STATE 0xBC

メッセージ列 <0xBC>, <SW1>, <SW2>, <CHK>

SW の状態確認要求命令

発行 マスター以外の機器

受取 マスター

応答 <OPC\_LONG\_ACK>, <0x3C>, < ?? >, <CHK>

応答に SW の状態が含まれていないといけません、記述されていませんでした。

OPC\_RQ\_SL\_DATA 0xBB

メッセージ列 <0xBB>, <SLOT1>, <0x00>, <CHK>

SLOT1 状態確認要求命令

発行 マスター以外の機器

受取 マスター

応答 <OPC\_SL\_RD\_DATA>, <0x0E>, <SLOT1>, <STAT>, <ADR>, <SPD>  
<DIRF>, <TRK>, <STAT2>, <ADR2>, <SND>, <ID1>, <ID2>, <CHK>

OPC\_MOVE\_SLOT 0xBA

メッセージ列 <0xBA>, <SLOT1>, <SLOT2>, <CHK>

SLOT1 の内容の SLOT2 への移動要求命令

特別な場合として

SLOT1 = 0x00 時 ( このとき SLOT2 は何でもよい )、DISPATCH 取得命令となり、  
応答 1 をマスターは返す。

SLOT2 = 0x00 時、DISPATCH 設定命令となり、SLOT1 は DISPATCH 用 SLOT としてマスターに管理される。応答 2 を返す。

SLOT1 = SLOT2 の時、SLOT 使用中要求命令となり、マスターはこのスロットを使用状態(IN\_USE)にする。

もし、DISPATCH を含め、失敗した場合 (SLOT1 or 2 = 0x7F を指定等) は失敗応答をマスターは戻す。

受取            マスター

応答 1         <OPC\_SL\_RD\_DATA>, <0x0E>, <SLOTX>, <STAT>, <ADR>, <SPD>  
<DIRF>, <TRK>, <STAT2>, <ADR2>, <SND>, <ID1>, <ID2>, <CHK>

注 : SLOTX はこの命令以前に DISPATCH 設定されていた SLOT だと思えます。

応答 2         <OPC\_SL\_RD\_DATA>, <0x0E>, <SLOT1>, <STAT>, <ADR>, <SPD>  
<DIRF>, <TRK>, <STAT2>, <ADR2>, <SND>, <ID1>, <ID2>, <CHK>

失敗応答       <OPC\_LONG\_ACK>, <0x3A>, <0x00>, <CHK>

注 : すでに IN\_USE あるいは DISPATCH になっている SLOT に再度同じ命令を発行した場合のマスターの動作については、記述がありません。

#### OPC LINK SLOTS 0xB9

メッセージ列     <0xB9>, <SLOT1>, <SLOT2>, <CHK>

SLOT1 の SLOT2 への連結要求命令。マスターはこの命令を受けるとリンク処理を行い、STAT1 内の SL\_CONUP/DN ビットを適切に設定する。

発行            マスター以外の機器

受取            マスター

成功応答       <OPC\_SL\_RD\_DATA>, <0x0E>, <SLOT1>, <STAT>, <ADR>, <SPD>  
<DIRF>, <TRK>, <STAT2>, <ADR2>, <SND>, <ID1>, <ID2>, <CHK>

注 : どちらの SLOT の状態が読み出されるかは記述していませんが、SLOT1 と思っています。

失敗応答       <OPC\_LONG\_ACK>, <0x39>, <0x00>, <CHK>

注 : リンクの失敗の条件が示されていませんが、たとえば、まだ車両が設定されていない SLOT を使用したとか、SLOT1,2 が規定されている範囲外の SLOT を選んだとかがあると思えます。

#### OPC UNLINK SLOTS 0xB8

メッセージ列     <0xB8>, <SLOT1>, <SLOT2>, <CHK>

SLOT1 を SLOT2 からの連結を解除する。マスターはリンク解除処理を行い、新しい SLOT を戻す。

発行            マスター以外の機器

受取            マスター

応答            <OPC\_SL\_RD\_DATA>, <0x0E>, <SLOTX>, <STAT>, <ADR>, <SPD>  
<DIRF>, <TRK>, <STAT2>, <ADR2>, <SND>, <ID1>, <ID2>, <CHK>

SLOTX は新しくマスターより設定された SLOT 番号。

注 : もし、この命令が実行不可能な場合 (例 SLOT1 が連結状態になっていない) におけるマスターの応答が規定されていない。フェール LACK かな???

#### OPC CONSIST FUNC 0xB6

メッセージ列     <0xB6>, <SLOT1>, <DIRF>, <CHK>

機能ビットをアップリンクの連結に設定する。

発行            マスター以外の機器

取込            マスター

応答            なし

#### OPC\_SLOT\_STAT1 0xB5

メッセージ列 <0xB5>, <SLOT1>, <STAT1>, <CHK>

SLOT1 に STAT1 の内容を書き込む

発行 マスター以外の機器

受取 マスター

応答 なし

#### OPC\_LONG\_ACK 0xB4

メッセージ列 <0xB4>, <LOPC>, <ACK1>, <CHK>

長い応答、通常、マスターが機器からの命令に対して応答のために発行する。

<LOPC>は機器から発行された命令の D7 を "0" にしたバイト、<ACK1>は応答に対応した内容が入られる。<ACK1> = 0x00 の場合はフェール、<LOPC> = 0x00 で応答する場合もある。この場合もフェールである。

発行 マスター

受取 直前に命令を発行した機器

応答 なし

LOPC は以下の場合 0x7F になる。

システムのコンフィギュレーション（設定）が、終わっていない場合、アドバンスドアクノレッジ（デコーダ側？）が搭載されていない場合。

#### OPC\_INPUT\_REP 0xB2

メッセージ列 <0xB2>, <IN1>, <IN2>, <CHK>

一般センサー状態レポート命令。センサーの状態をレポートする。

発行 センサーを装備している機器

受取 マスター

応答 なし

#### OPC\_SW\_REP 0xB1

メッセージ列 <0xB1>, <SN1>, <SN2>, <CHK>

ターンアウト（スイッチ、ポイント）状態レポート命令。ターンアウトの状態をレポートする。

発行 ターンアウト（スイッチ、ポイント）の状態を検出できる機器

受取 マスター

応答 なし

#### OPC\_SW\_REQ 0xB0

メッセージ列 <0xB0>, <SW1>, <SW2>, <CHK>

SW 動作要求命令（応答なし）。<SW1>, <SW2>の内容で SW 作動させる。

発行 マスター以外の機器

受取 マスター

応答 なし

#### OPC\_LOCO\_SND 0xA2

メッセージ列 <0xA2>, <SLOT1>, <SND>, <CHK>

SLOT1 の車両に対して <SND>の内容のサウンドを制御する。

発行 マスター以外の機器

受取 マスター

応答 なし

#### OPC\_LOCO\_DIRF 0xA1

メッセージ列 <0xA1>, <SLOT1>, <SND>, <CHK>

SLOT1 の車両に対して <DIRF>の内容の方向、機能を制御する。

発行        マスター以外の機器  
受取        マスター  
応答        なし

OPC\_LOCO\_SPD        0xA0

メッセージ列        <0xA0>, <SLOT1>, <SPD>, <CHK>

SLOT1 の車両に対して <SPD> の内容の速度を設定する。

発行        マスター以外の機器  
受取        マスター  
応答        なし

- C) 6 バイト命令  
規定されていません。使用不可

- D) 可変長命令  
<op code> <n> <arg1> <arg2> <arg3> .....<checksum>

OPC\_WR\_SL\_DATA        0xEF

メッセージ列

<0xEF>, <0x0E>, <SLOT1>, <STAT1>, <ADR>, <SPD>, <DIRF>,  
<TRK>, <STAT2>, <ADR2>, <SND>, <ID1>, <ID2>, <CHK>

SLOT1 に各内容の書き込み要求命令

発行        マスター以外の機器  
受取        マスター  
応答        なし

OPC\_SL\_RD\_DATA        0xE7

メッセージ列

<0xE7>, <0x0E>, <SLOT1>, <STAT1>, <ADR>, <SPD>, <DIRF>,  
<TRK>, <STAT2>, <ADR2>, <SND>, <ID1>, <ID2>, <CHK>

SLOT1 の情報応答命令。SLOT1 の情報を要求された時、その要求が正しい場合に  
応答される。SLOT1 = 0x00 の時はマスターの設定情報となる。

発行        マスター  
受取        直前に SLOT1 情報を必要とする命令を発効した機器  
応答        なし

その他

*OPC\_PEER\_XFER(0xE5)*, *OPC\_IMM\_PACKET(0xED)* 命令について記述されていま  
すが、これらの命令は未確定のため、省略します。

凡例

数字、アルファベット(A-F)の前の 0x は、16 進表示を示している。例：10 進表示で 12  
は、16 進では C、2 進では 1100 である。

A(10)、B(11)、C(12)、D(13)、E(14)、F(15) なお、()内は 10 進表示。基本的に 16 進  
での表示とする。16 進表示の右側に()で 10 進表示をする場合があります。2 進表示(0  
または 1)はバイト内の各ビットを表す時に用いています。

各内容はアルファベット順に並べてあります。

<ADR>、<ADR2>        車両アドレスバイト

ADR は NMRA 規格の標準の 7 ビットアドレスを示す。ADR2 は 14 ビットアドレスの

上位 7 ビットである。ADR2=0x00 でない場合、マスターは 14 ビットアドレスの DCC パケットを生成できなくてはならない。ADR = 0x00 かつ ADR2= 0x00 の場合は、アナログ車両を示す。

<CHK> checksum バイト ( - メッセージの基本構造参照)  
メッセージの最後につけられるエラー検出用バイト

<DIRF> 方向、ファンクションバイト

車両の方向と、機能状態バイト

D7		"0"
D6	SL_XCLT	"0" 使用不可
D5	SL_DIR	"1"..前進、"0"..後退
D4	SL_F0	"1"..方向ライトオン、"0"..方向ライトオフ
D3	SL_F4	"1"..F4 オン "0"..F4 オフ
D2	SL_F3	"1"..F3 オン "0"..F3 オフ
D1	SL_F2	"1"..F2 オン "0"..F2 オフ
D0	SL_F1	"1"..F1 オン "0"..F1 オフ

<ID1>、<ID2> 機器固有番号

ID1 / ID2 は機器につけられる固有の番号であり、各 7 ビット、計 14 ビットの ID となる。

ID1 / ID2

0x00 / 0x00 ID はつけられていない。

0x01 / 0x00 – 0x7F / 0x01 PC ID 番号、低いほうの 7 ビットは PC 番号 ???

0x00 / 0x20 – 0x7F / 0x03 使用不可

0x00 / 0x40 – 0x7F / 0x7E 通常 Throttle で使用

<IN1>、<IN2> 入力センサーバイト

<IN1>

D7	0	
D6	A6	
D5	A5	
D4	A4	
D3	A3	A0-A6 入力センサーデコーダアドレス下位 7 ビット
D2	A2	
D1	A1	A1, A0 は DS54 の場合は 4 つのセルに対応する。
D0	A0	

<IN2>

D7	0	
D6	X	"1"..コントロールビット、"0"..使用不可
D5	I	"1".."スイッチ"入力、"0"..外部入力(aux)
D4	L	"1"..>=6[V]、"0"..0[V]
D3	A10	
D2	A9	
D1	A8	A7-A10 入力センサーデコーダアドレス上位 4 ビット
D0	A7	

<SLOT1>、<SLOT2> スロット番号バイト

スロット番号をあらわすバイト

0x00 - 0x7F (127)までの 128 個のスロットがある。0x00 のスロットは特殊スロットであり、また、0x70(112)-0x7F(127)は DIGITRAX 社が使用する。



### <SN1>、<SN2> ターンアウトセンサーバイト

#### <SN1>

D7	0	
D6	A6	
D5	A5	
D4	A4	
D3	A3	A0-A6 入力センサーデコーダアドレス下位 7ビット
D2	A2	
D1	A1	A1, A0 は DS54 の場合は 4 つのセルに対応する。
D0	A0	

#### <SN2>

D7	0	
D6	1	
D5	I	"1".."スイッチ"入力、"0"..外部入力(aux)
D4	L	"1"..>=6[V]、"0"..0[V]
D3	A10	
D2	A9	
D1	A8	A7-A10 入力センサーデコーダアドレス上位 4ビット
D0	A7	

又は

#### <SN2>

D7	0	
D6	0	
D5	C	"1"..直進出力オン、"0"..直進出力オフ
D4	T	"1"..分岐出力オン、"0"..分岐出力オフ
D3	A10	
D2	A9	
D1	A8	A7-A10 入力センサーデコーダアドレス上位 4ビット
D0	A7	

### <SND> SLOT サウンドバイト

SLOT のサウンドあるいはファンクションモード 2 パケット(F5-F8)バイト

D7	使用不可	
D6	使用不可	
D5	使用不可	
D4	使用不可	
D3	SL_SND4 / F8	"1" SLOT のサウンドまたは機能オン、"0"でオフ
D2	SL_SND3 / F7	"1" SLOT のサウンドまたは機能オン、"0"でオフ
D1	SL_SND2 / F6	"1" SLOT のサウンドまたは機能オン、"0"でオフ
D0	SL_SND1 / F5	"1" SLOT のサウンドまたは機能オン、"0"でオフ

### <SPD> SLOT 速度バイト

#### 速度バイト

0x00	慣性をもって速度 0 へ
0x01	緊急停止 (速度 0)
0x02-0x7F	速度データ 0x02 が一番遅く、0x7F が一番速い。

### <STAT1>、<STAT2> SLOT の状態情報バイト

指定された SLOT の状態(status)情報を保持するバイト。SLOT の指定は命令による。

#### <STAT1 >

D7	SL_SPURGE	マスター内部動作で使用。LocoNet 上は "0"
D6	SL_CONUP	
D5	SL_BUSY	

D4 SL\_ACTIVE  
D3 SL\_CONDN  
D2 SL\_SPDEX  
D1 SL\_SPD14  
D0 SL\_SPD28

各ビットの組み合わせで、状態が決まる。

D6 SL\_CONUP / D3 SL\_CONDN

D6/D3

- 11 この SLOT は上位 SLOT、下位 SLOT 両方へ連結操作状態(中間の位置)
- 10 この SLOT が下位 SLOT からのみ連結操作状態(トップ位置)
- 01 この SLOT が下位 SLOT として上位 SLOT に連結操作状態
- 00 この SLOT は単独操作状態

注：この内容に関しては、私の理解が間違っている可能性が大了。

D5 SL\_BUSY / D4 SL\_ACTIVE

D5/D4

- 11 使用中(IN\_USE)、すでに誰かがこの SLOT に設定された車両を使用中
- 10 休止中(IDLE)、この SLOT は誰かに設定されているが、休止状態
- 01 共有(COMMON)、この SLOT に設定された車両を誰も使用していない。
- 00 空き(FREE)、この SLOT に車両が設定されていない。

IN\_USE, COMMON 状態にある SLOT は、DCC パケットの生成のため、常に処理されます。注：原文は REFRESH と記述してあります。これはマスター内部の処理の話であらうと思ってます。SLOT 内容の変化、あるいは、定期的にマスターが DCC パケットを出力するための処理の一部の事と考えられます。

D2 SL\_SPDEX / D1 SL\_SPD14 / D0 SL\_SPD28

D2/D1/D0

- 111 128 speed step、連結操作可能
- 110 規定されていない
- 101 規定されていない
- 100 28 speed step、連結操作可能
- 011 128 speed step モード
- 010 14 speed step モード
- 001 28 speed step Trinary パケット出力
- 000 28 speed step DCC 標準パケット出力

<STAT2>

- D7 規定されていない
- D6 規定されていない
- D5 規定されていない
- D4 規定されていない
- D3 “1” 拡張入力 ID1/2、“0” 符号化別名
- D2 “1” 拡張 ID1/2 は使用されていない。
- D1 規定されていない
- D0 “1” この SLOT は SUPPRESSED ADV consist ???

注： 関連として EXPANSION RESERVED ID1, ID2 がある。その説明は、STAT2.4 “=”1”の時、これらの各 7 ビットの ID は PC あるいは THROTTLE により書き込まれる。とあるが STAT2.4 の意味がはっきりしない。もし、STAT2 の D4 であっても、規定されていないビットです。STAT2 については、理解していません。

### <SW1>、<SW2> スイッチ用バイト

#### <SW1>

D7	0	
D6	A6	
D5	A5	
D4	A4	
D3	A3	A0-A6 スイッチ用デコーダアドレス下位 7ビット
D2	A2	
D1	A1	A1, A0 は DS54 の場合は 4 つのセルに対応する。
D0	A0	

#### <SW2>

D7	0	
D6	0	
D5	DIR	"1"..直進 / 緑、"0"..分岐 / 赤
D4	ON	"1"..出力オン、"0"..出力オフ
D3	A10	
D2	A9	
D1	A8	A7-A10 スイッチ用デコーダアドレス上位 4ビット
D0	A7	

### <TRK> トラックバイト

レール上あるいは全体的な状態を表すバイト

D7	使用不可	
D6	使用不可	
D5	使用不可	
D4	使用不可	
D3	GTRK_PROG_BUSY	"1" マスターがプログラム中で忙しい
D2	GTRK_MLOK1	"1" LocoNet 1.1 が使用可能、"0" マスターは DT200
D1	GTRK_IDLE	"0" レールへの制御が一旦停止状態、緊急停止
D0	GTRK_POWER	"1" DCC パケット出力状態可能状態

## 5. デコーダプログラム

### I. デコーダプログラムの開始

OPC\_WR\_SL\_DATA命令を用い、SLOT 0x7C(124) に、プログラム用データを書き込むことにより、開始する。SLOT = 0x7C にした場合、OPC\_WR\_SL\_DATA命令はプログラム開始要求命令となる。 の凡例を参照して下さい。

#### メッセージ列

<0xEF>, <0x0E>, <SLOT = 0x7C(124)>, <PCMD>, <0x00>, <HOPSA>, <LOPSA>, <TRK>, <CVH>, <CVL>, <DATA7>, <0x00>, <0x00>, <CHK>

発行 マスター以外の機器

受取 マスター

応答 OPC\_LONG\_ACK命令<0xB4>にてマスターがすぐに応答する。

<0xB4>, <0x7F>, <0x7F>, <CHK> プログラム機能は搭載していない

<0xB4>, <0x7F>, <0x00>, <CHK> プログラマービジーのためプログラム不可

<0xB4>, <0x7F>, <0x01>, <CHK> プログラム開始、結果は OPC\_SL\_RD\_DATA命令よりマスターから通知される。

<0xB4>, <0x7F>, <0x40>, <CHK> プログラム開始、結果は通知されない。

プログラムを開始しようとして、マスターから、プログラム不可と応答が来た場合、少し待ってから再び、プログラム開始を要求して下さい。プログラム開始の要求前に OPC\_SL\_RD\_DATA命令により、プログラム状態を確認して下さい。<TRK>バイト、D3

GTRK\_PROG\_BUSYビットで現在のプログラムの状態を確認出来ます。プログラミングを中止するには、<PCMD = 0x00>とし、OPC\_WR\_SL\_DATAを発行して下さい(SLOT = 0x7C)。マスターは OPC\_SL\_RD\_DATA 命令にて、プログラミングの結果を通知します。注：原文では<0xE6> RD コマンドにより、プログラミングの中止のエコーが返ってくると記述がありますが、<0xE7>コマンドにより、プログラミングの応答があると思っています。

## II. プログラム終了応答

プログラム開始要求命令をマスターに発行した後、マスターからの応答が<0xB4>, <0x7F>, <0x01>, <CHK>であった場合、プログラム終了時に、マスターから OPC\_RD\_SL\_DATA 命令にて、プログラム結果の情報の通知がされます。

### メッセージ列

<0xE7>, <0x0E>, <SLOT = 0x7C(124)>, <PCMD>, <PSTAT>, <HOPSA>, <LOPSA>, <TRK>, <CVH>, <CVL>, <DATA7>, <0x00>, <0x00>, <CHK>

発行                    マスター  
受取                    プログラム開始要求を行った機器  
応答                    なし

プログラムが成功した場合、DATA, CHV に読み出した結果が含まれます。

## III. 凡例

### PCMD

D7	"0"	
D6	Write / Read	"1" ..書き込み、"0" ..読み込み
D5	Byte Mode	"1" ..バイト動作、"0" ..ビット動作（可能であれば）
D4	TY1	プログラム方法選択ビット
D3	TY0	プログラムタイプ選択ビット
D2	Ops Mode	"1" 通常のルール上でのプログラミング(Ops モード) "0" サービスルールでのプログラミング
D1	使用不可	
D0	使用不可	

### プログラムモードの組み合わせ

ByteMode	OpsMode	TY1	TY0	
1	0	0	0	サービスルール、バイト、ページドモード
1	0	0	1	サービスルール、バイト、ダイレクトモード
0	0	0	1	サービスルール、ビット、ダイレクトモード
x	0	1	0	サービスルール、ビット、physical Register モード
x	0	1	1	サービスルール、使用不可
1	1	0	0	Ops モード、バイト、応答なし
1	1	0	1	Ops モード、バイト、応答あり
0	1	0	0	Ops モード、ビット、応答なし
0	1	0	1	Ops モード、ビット、応答あり

### HOPSA, LOPSA

Ops モードで使用されるデコーダアドレスを示す。

	HOPSA	LOPSA
D7	"0"	"0"
D6	A13	A6
D5	A12	A5
D4	A11	A4
D3	A10	A3
D2	A9	A2

D1	A8	A1
D0	A7	A0

サービスモードでは HOPSA, LOPSA とともに 0x00 にします。

#### CVH

CVナンバーの上位 3 ビットと、書き込み、比較用データの最上位ビット(D7)

D7	"0"
D6	"0"
D5	CV9
D4	CV8
D3	"0"
D2	"0"
D1	D7
D0	CV7

#### CVL

CVナンバーの下位 7 ビット

D7	"0"
D6	CV6
D5	CV5
D4	CV4
D3	CV3
D2	CV2
D1	CV1
D0	CV0

#### DATA7

書き込み、比較用データの低位 7 ビット(D0-D6)。CVH ビット 2 が最上位となり、一バイトのデータになる。

#### PSTAT

プログラム結果状況エラーフラグ

D7	使用不可
D6	使用不可
D5	使用不可
D4	使用不可
D3	"1" ユーザがプログラムを中止
D2	"1" デコーダから読み出し比較エラー検出
D1	"1" デコーダから書き込み応答なし
D0	"1" サービスモードでデコーダがなかった

## 6. 早送り時計

### I. 現在の時計の状態の確認

OPC\_RQ\_SL\_DATA 命令(0xBB)で、SLOT 番号 0x7B(123)を読み出すことにより、現在の時計の状態が分かります。

メッセージ列

<0xBB>, <SLOT = 0x7B(123)>, <0>, <CHK>,

発行                    マスター以外の機器

受取                    マスター

応答

<0xE7>, <0x0E>, <SLOT = 0x7B(123)>, <CLK\_RATE>, <FRAC\_MINSL>, <FRAC\_MINSH>, <256-MINS\_60>, <TRK>, <256-HRS\_24>, <DAYS>, <CLK\_CNTRL>, <ID1>, <ID2>, <CHK>

## II. 早送り時計の設定

OPC\_WR\_SL\_DATA命令を用い、SLOT 0x7B(123) に、時計用データを書き込むことにより、設定を行う。

メッセージ列

<0xEF>, <0x0E>, <SLOT = 0x7B(123)>, <CLK\_RATE>, <FRAC\_MINSL>, <FRAC\_MINSH>, <256-MINS\_60>, <TRK>, <256-HRS\_24>, <DAYS>, <CLK\_CNTRL>, <ID1>, <ID2>, <CHK>

発行                    マスター以外の機器

受取                    マスター

応答

<0xE7>, <0x0E>, <SLOT = 0x7B(123)>, <CLK\_RATE>, <FRAC\_MINSL>, <FRAC\_MINSH>, <256-MINS\_60>, <TRK>, <256-HRS\_24>, <DAYS>, <CLK\_CNTRL>, <ID1>, <ID2>, <CHK>

## III. 凡例

### CLK\_RATE

早送りの速さの設定値

0x00(0)            時計は停止状態

例 0x0A(10)        普通の10倍の早さで時計が進みます。最大は0x7F(127)倍まで。

注：原文は7F/128となっておりますが、127の間違いと思います。

### FRAC\_MINSL / FRAC\_MINH

これらは、マスター内部で使用される補助用分カウンタです。SYNC命令<0xE6>, <0x7B>..によりこのカウンタはリセットされます。

注：私にはなんのことが???です。0xE6で始まる命令はこのLocoNet Personal Edition内には定義されていません。

### 256-MINS\_60

早送り時計の分を表す。256から引かれた値である。

### 256-HRS\_24

早送り時計の時間を表す。256から引かれた値である。

### DAYS

24時間の繰り返された回数を示す。日の意味

### CLK\_CNTRL

時計制御バイト

D6しか定義されていない。D6="1"でこの時刻は正しい。"0"で正しくない(無視すべき)

### ID1 / ID2

時計を設定した機器のIDを示す。ID1 / ID2 =0x00の場合はまだ設定されていない。

注：256-MINS\_60, 256-HRS\_24の記述の中にModulo 0-59, Modulo 0-23とあるが、Moduleとは、ある数を他の数で割った時の余りを意味している。当然0で割ることは出来ないので、Modulo 0-59, Modulo 0-23はあり得ない。

以上