

令和元年（ワ）第10940号 損害賠償請求事件

原告 森次 茂廣

被告

## 準備書面 18

令和4年9月9日

大阪地方裁判所第26民事部合議係 御中

被告訴訟代理人弁護士

- 1 御庁からの求釈明（その3）について、別紙のとおり、主張する。
- 2 松谷専門委員から御提案頂いた、実地に、サイレントロボと、規制法に係る公害振動を測定する他機器とを、並列的に設置し、双方の測定値を確認するという検証方法について、被告は承諾する。

以 上

## 別紙

## 求釈明事項（その3）についての回答

## 求釈明事項（1）

騒音計 NL-21、振動計 VM-52 の測定レンジと出力電圧の関係は、準備書面 16 の p 1 - 3 記載のとおりと  
のことですが、各装置の仕様書を提示し、その仕様書を基に説明してください。

## (1) - 1. NL-21 の測定レンジと出力電圧の関係

NL-21 の測定レンジは「NL-21 取扱説明書」の仕様より、図.1 の赤枠で示すように 20~80dB, 20~90dB,  
20~100dB, 20~110dB, 30~120dB, 40~130dB の 6 段階から選択できるとなっている。

測定時間		10秒、1、5、10、15、30分、1、8、24時間及び手動 周長測定時間 200 時間	
測定レベル範囲	A 特性:	28 ~ 130 dB	
	C 特性:	33 ~ 130 dB	
	平坦特性:	38 ~ 130 dB	
	C 特性ピーク許容レベル:	55 ~ 141 dB	
	平坦特性ピーク許容レベル:	60 ~ 141 dB	
自己雑音レベル		NL-21	NL-31
	A 特性:	22 dB 以下	20 dB 以下
	C 特性:	27 dB 以下	25 dB 以下
	平坦特性:	32 dB 以下	30 dB 以下
リニアリティレンジ:	100 dB		
基準の許容レベル:	94 dB		
基準のレベルレンジ:	30 ~ 120 dB		
レベルレンジ切り替え器	10 dB ステップ 6 段		
	20 ~ 80 dB		
	20 ~ 90 dB		
	20 ~ 100 dB		
	20 ~ 110 dB		
	30 ~ 120 dB		
	40 ~ 130 dB		
	1/1、1/3 オクターブフィルタカード NX-21SA またはユニ バーサルフィルタカード NX-21VA (いずれも別売) が ON 状 態の時のみ 10 dB ステップ 7 段		
	10 ~ 70 dB NX-21SA または NX-21VA が ON の時の み選択可能		
	20 ~ 80 dB		
	30 ~ 90 dB		
	40 ~ 100 dB		
	50 ~ 110 dB		
	60 ~ 120 dB		
	70 ~ 130 dB		

サイレントロボでは  
未使用

図.1 測定レンジ

また、出力電圧については「NL-21 取扱説明書」の仕様より、図.2 の矢印で示すように「2.5V (フルスケ

## 別紙

ール)、0.25V/10dB」で出力されると記載されている。

仕様	
出力端子	交流/直流出力端子 キー操作により交流または直流出力機能を選択する
交流出力	選択された周波数特性及びフィルタによる交流出力 出力電圧: 1 Vrms(フルスケール) 出力抵抗: 600 Ω 負荷抵抗: 10 kΩ 以上
直流出力	出力電圧: 2.5 V(フルスケール)、0.25 V/10 dB ← 出力抵抗: 50 Ω 負荷抵抗: 10 kΩ 以上
I/O 端子	RS-232-C を利用したパソコンによる懸待計の制御とデータ出力
RS-232-C	プリンタ DPU-414 / CP-11 / CP-10 へのデータ出力 通信方式: 調歩同期 データ長: 8 ビット ストップビット: 1 ビット パリティ: なし 通信速度: 4800 bps、9600 bps および 19200 bps フロー制御: あり X パラメータ制御か RTS / CTS 制御かを選択
コンパレータ出力	オープンコレクタ出力 設定レベルを超えると ON になる 最大印加電圧 24 V 最大電流 60 mA 設定レベル: 30 ~ 130 dB の間で 1 dB ステップ

図.2 出力電圧

これは出力電圧について次のような関係があることを示す。

- ・測定レンジの最大値が測定されると 2.5V の電圧が出力される
- ・測定値が 10(dB)変化すると出力電圧は 0.25V 変化する

NL-21 は測定レンジが 6 段階あり、サイレントロゴでは現場の状況にあった最適な測定レンジを選択して使用する。選択された測定レンジでの測定値と出力電圧の関係を図示すると図.3~図.8 のようになる。

別紙

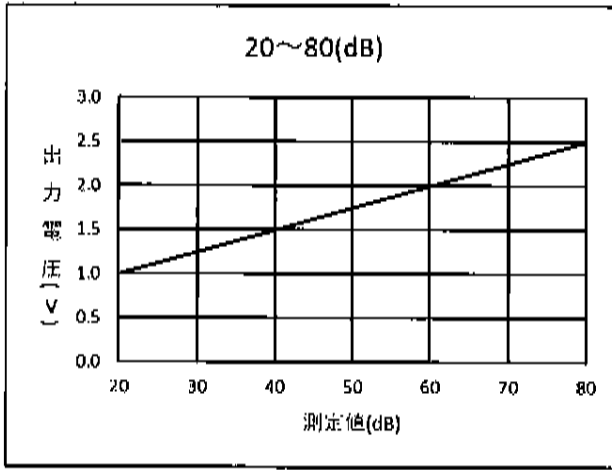


図. 3

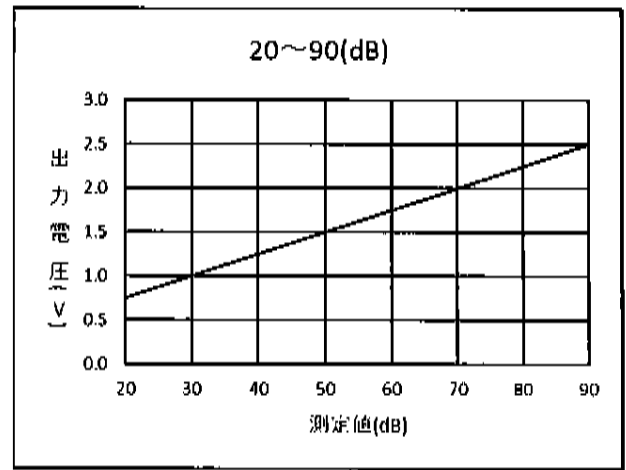


図. 4

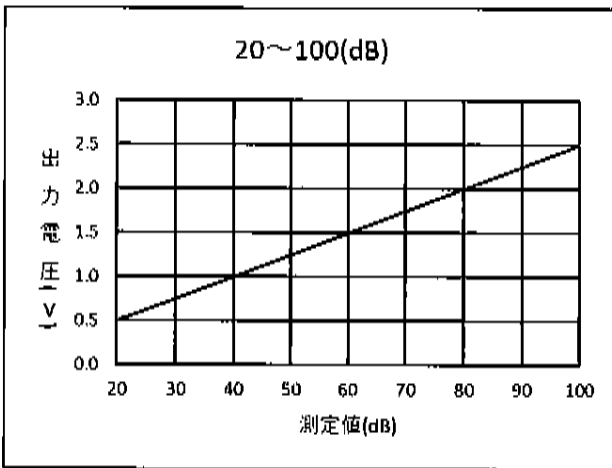


図. 5

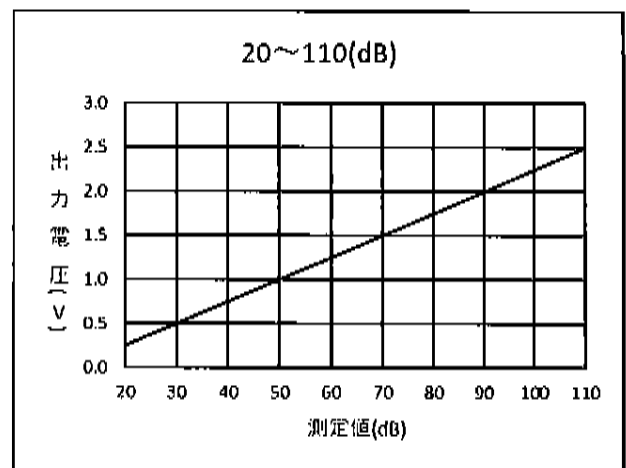


図. 6

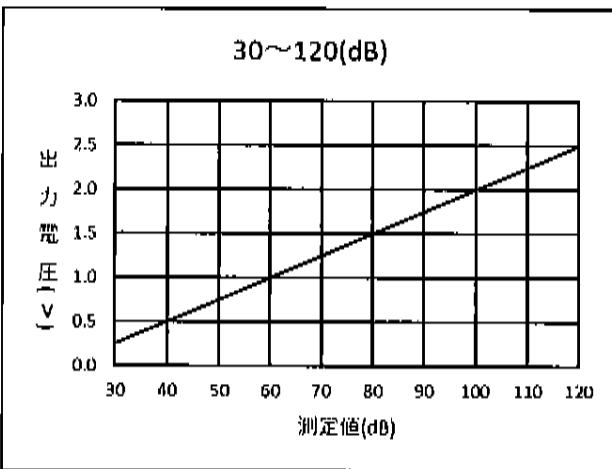


図. 7

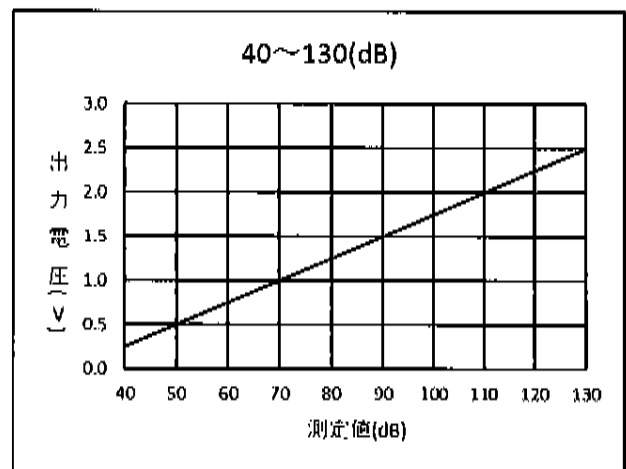


図. 8

つまり、測定レンジと出力電圧の関係は、次の①と②に説明できる。

①測定値と出力電圧の関係は、測定レンジ毎に異なる (ex 測定値が同じ 50dB であっても、20~80dB の測定レンジの出力電圧値と 20~90dB の測定レンジの出力電圧値とは異なる)。

## 別紙

②各測定レンジ内の測定値と出力電圧の関係は、各測定レンジの最大測定値の出力電圧が 2.5Vであることを起点として、測定値が 10 dB 下がるごとに 0.25V 電圧が下がる（各測定レンジの最小測定値を起点としない。測定レンジによって、最小測定値の出力電圧は異なる）。

## 別紙

## (1) - 2. VM-52 の測定レンジと出力電圧の関係

VM-52 の測定レンジは「VM-52 取扱説明書」の仕様より、図.9 の赤枠で示す通り 20~70dB、30~80dB、40~90dB、50~100dB、60~110dB、70~120dB の 6 段階から選択できるとなっている。

仕様	
<b>仕 様</b>	
型式承認番号	第 W941 号
適用規格	計量法・振動レベル計 JIS C 1510-1995
測定機能	振動レベル (L <sub>v</sub> ) 振動加速度レベル (L <sub>va</sub> ) 振動レベル、または振動加速度レベルの時間平均レベル (L <sub>K</sub> , 3 方向同時) 振動レベル、または振動加速度レベルのパワー平均 (L <sub>eq</sub> , 3 方向同時) 振動レベル、または振動加速度レベルの最大値 (L <sub>max</sub> , 3 方向同時)
測定レベル範囲	振動レベル * 30 ~ 120 dB (0dB = 10 <sup>-4</sup> m/s <sup>2</sup> ) 振動加速度レベル 30 ~ 120 dB
周波数範囲	振動レベル 1 ~ 80 Hz 振動加速度レベル 1 ~ 80 Hz
周波数補正回路	振動感覚補正特性: 計量法または JIS による鉛直振動特性と JIS による水平振動特性 平坦ん特性: JIS による平坦ん特性
突効値回路	真の突効値検出回路 動特性: 0.63 s
レベルレンジ切替	10 dB ステップ、6 段 20 ~ 70 dB、30 ~ 80 dB、40 ~ 90 dB、50 ~ 100 dB、 60 ~ 110 dB、70 ~ 120 dB
自己雑音	振動レベル 24 dB 以下 振動加速度レベル 24 dB 以下
出力校正	内蔵発振器 (31.5 Hz 正弦波) による電氣的校正

図.9 測定レンジ

また、出力電圧については「VM-52 取扱説明書」の仕様より、図.10 の矢印で示す通り「3.0V (フルスケールで)、0.5V/10dB」で出力されると記載されている。

## 別紙

仕様		
演算	時間平均レベル： パワー平均： 最大値： 演算時間： サンプリング周期：	$L_a, L_{10}, L_{50}, L_{90}, L_{95}$ を累積度数分布より 3方向同時に演算 指定時間内で振動レベルまたは振動加速度レベルの2乗平均を3方向同時に演算 指定時間内で振動レベルまたは振動加速度レベルの最大値を3方向同時に演算 10秒、500秒、1分、5分、10分、15分、1時間、8時間、24時間及びマニュアル(最長99時間59分59秒) 但し、時間率レベルの演算時間は最長1時間 10ms(パワー平均、最大値) 100ms(時間率レベル但し、演算時間500秒の場合は5秒)
表示バックライト付き液晶表示	数値表示： バーグラフ表示： 警告： 電池電圧： 年、月、日、時刻/経過時間表示：	4けた表示、表示周期1秒、分解能0.1dB X、Y、Z平均切替 目盛範囲50dB、表示周期0.1秒、分解能10dB X、Y、Z同時表示 Over 過大入力または増幅器飽和時に表示 Under 過小入力時に表示(入力信号が各レンジの下限を下回る場合) 残量を4段階表示 切り替えて表示
振動ピックアップ	型式： 構造： 検出方向： 前置増幅器： 寸法、質量： その他：	PV-83B せん形圧電式加速度ピックアップ 3方向(X、Y水平、Z鉛直) 内蔵 $\phi 67 \times 40.7$ mm、約350g 防水性 保護等級2 防滴II形(JIS C 0920)
出力端子	BNCコネクタ 交流/直流 手動切替	3個(3方向)
交流出力	出力電圧、 負荷抵抗	1Vrms(フルスケールで) 10k $\Omega$ 以上
直流出力	出力電圧：	3.0V(フルスケールで)、0.5V/10dB ←

図.10 出力電圧

これは出力電圧について次のような関係があることを示す。

- ・測定レンジの最大値が測定されると3.0Vの電圧が出力される
- ・測定値が10(dB)変化すると出力電圧は0.5V変化する

VM-52は測定レンジが6段階あり、サイレントロボでは現場の状況にあった最適な測定レンジを選択して使用する。選択された測定レンジでの測定値と出力電圧の関係を図示すると図.11～図.16のようになる。

別紙

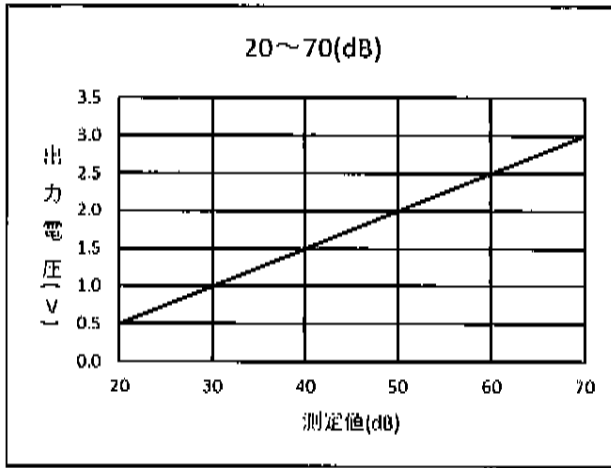


図.1 1

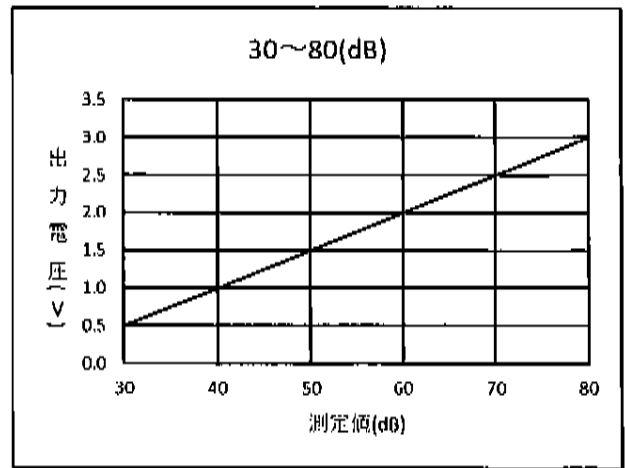


図.1 2

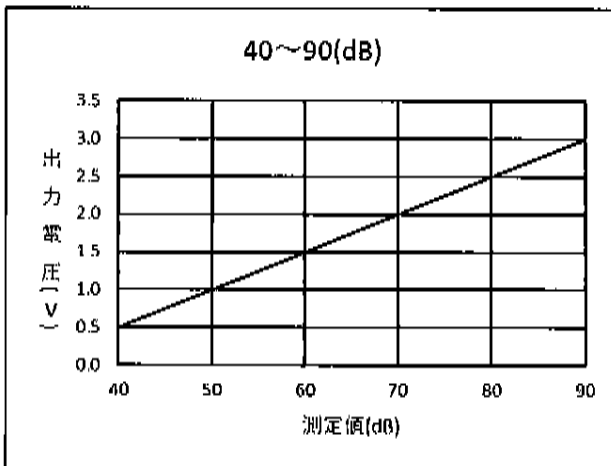


図.1 3

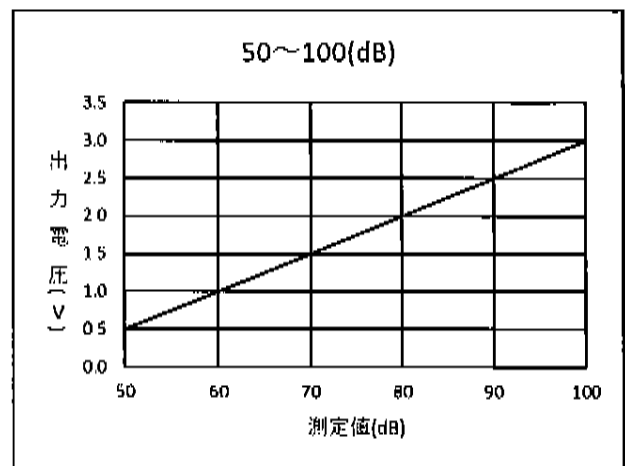


図.1 4

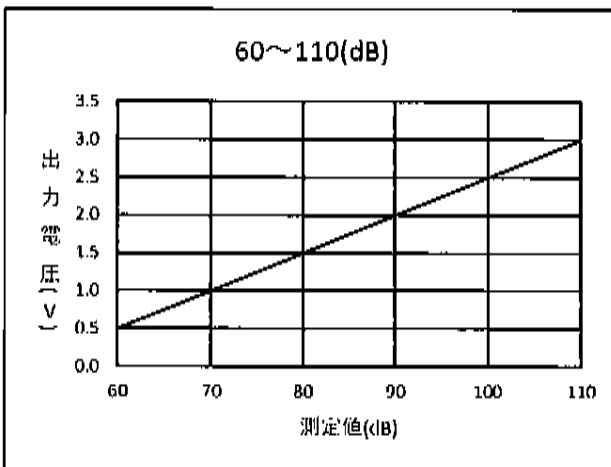


図.1 5

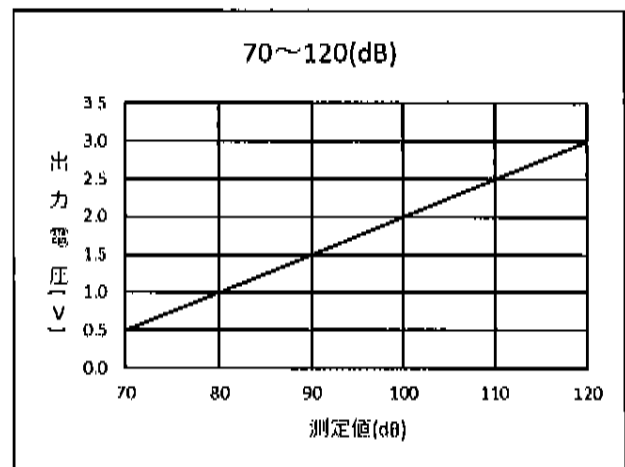


図.1 6

## 別紙

つまり、測定レンジと出力電圧の関係は、次の①と②に説明できる。

①測定値と出力電圧の関係は、測定レンジ毎に異なる（ex 測定値が同じ 50dB であっても、20～70dB の測定レンジの出力電圧値と 30～80dB の測定レンジの出力電圧値とは異なる）。

②各測定レンジ内の測定値と出力電圧の関係は、各測定レンジの最大測定値の出力電圧が 3.0V であることを起点として、測定値が 10 dB 下がるごとに 0.5V 電圧が下がる（各測定レンジの最小測定値を起点としない。測定レンジによって、最小測定値の出力電圧は異なる）。

別紙

求釈明事項（２）

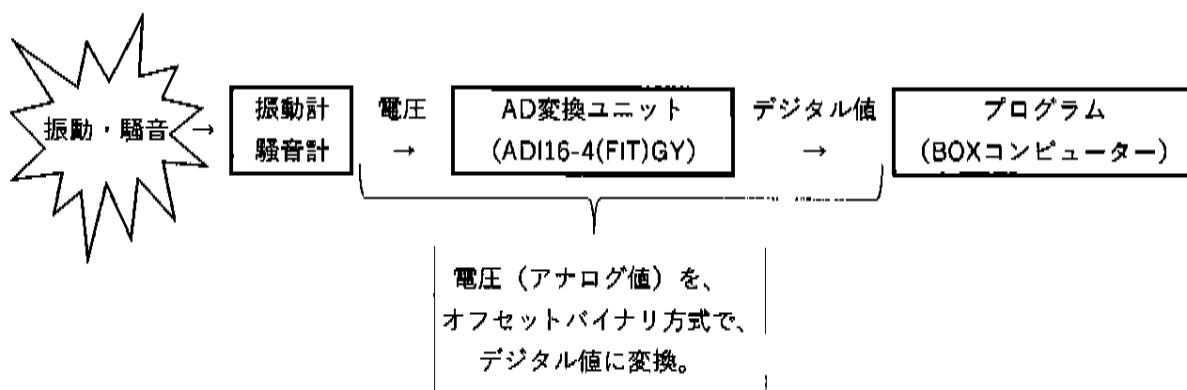
AD変換ユニットにおいて、騒音計 NL-21、振動計 VM-52 から入力される電圧値が、どのようなデジタル値に変換されるのか、乙31を基に説明してください。

振動計又は騒音計は振動測定値又は騒音測定値を電圧の形で外部出力するところ、サイレントロボのプログラムが演算処理するためには、振動計又は騒音計がサイレントロボに向け出力した電圧値をデジタル値に変換する必要があるため、コンテック社のAD変換ユニット（ADI16-4(FIT)GY）を使用している。

この点、裁判所から専門委員へ提示された質問事項（プログラム関連）にて「ボックス・コンピュータのAD変換器から出力されるデジタルデータ変換値 InpBuf は…」と質問が行われているが、これは入出力関係が真逆になっている。

デジタルデータ変換値は、ボックス・コンピュータのAD変換器から出力されるのではなく、AD変換器からボックス・コンピュータへ入力されるものである。これを図Aに示す。

図A



別紙

乙31号証 (ADI16-4(FIT)GY 解説書) の P.32 表 4.5 に、この AD 変換ユニットの入力電圧と変換データの関係 (図.17 の赤矢印部分) が示されている。これは準備書面 12「2. 変換処理について (2)」で説明したように「オフセット・バイナリ方式」という変換方法である。

第4章 ICアドレスマップを使用する場合

表4.5 ±10Vレンジ時の変換データの具体例

入力電圧 (±10Vレンジ)	10ビット変換データ オフセットバイナリ
+9.9997V	FFFFh
⋮	⋮
0.00030V	5001h
0.00000V	5000h
-0.00030V	4FFFh
⋮	⋮
-10.00031V	0000h

変換データの入力の制御ポートは次のとおりです。

制御IO アドレス	D7	D6	D5	D4	D3	D2	D1	D0
Input 129 Data	Analog Input Status 0							
	0	0	Sampling (Back Error Check Input)	0	Data Over Write Error	0	Data Read Enable	
Input 116 (10h)	Analog Input (Lower)							
	Conversion Data7	Conversion Data6	Conversion Data5	Conversion Data4	Conversion Data3	Conversion Data2	Conversion Data1	Conversion Data0 (LSB)
Input 117 (11h)	Analog Input (Upper)							
	Conversion Data13 (MSB)	Conversion Data14	Conversion Data15	Conversion Data12	Conversion Data11	Conversion Data10	Conversion Data9	Conversion Data8

図4.22 変換データの入力制御ポート

アナログ入力および制り込みステータスの詳細については後述します。  
高抵抗部でのサンプリング回路の例を以下に示します。

変換データの入力例を以下に示します。

```

while ( !(Inp( ADR122 ) & 2 ) );
CzmemAllData = Inp( ADR116 );
UpperAllData = Inp( ADR117 );

```

CONTEC  
ADI16-4(FIT)GY

図.17 変換データ

別紙

AD変換は、AD変換器にて、乙31号証（ADI16-4(FIT)GY解説書）のP.31に示す式（図.18の赤矢印で示す式）で行われる。

第4章 I/Oアドレスマップを使用する場合

**■サンプリング開始**  
 チャンネルモードがシングルチャンネルモードの場合は、変換するチャンネルを指定します。マルチチャンネルモードの場合は、変換するチャンネルの上限値(1以上の値)を指定します。例えば、3chを指定すると0-3chをサンプリングします。サンプリング開始の制御ポートは、次のとおりです。

先頭I/O アドレス	D7	D6	D5	D4	D3	D2	D1	D0
Output +18 (12h)	Channel Data						Channel Data1	Channel Data0
	0	0	0	0	0	0		

**図4.20 サンプリング開始**  
 サンプリングモードがソフトウェアコマンドの場合、指定チャンネルを1回だけサンプリングします。内部サンプリングクロックを使用する場合、1回目のサンプリングが開始され同時に内部サンプリングクロックがスタートします。サンプリングモードがクロックモードに設定されている状態で、サンプリング開始を実行し再度サンプリング開始を実行するとそれまでの変換データとアナログ入力ステータスは、リセットされ新たにサンプリングが開始されます。モジュールの初期化コマンド、サンプリング条件設定コマンドを実行するとサンプリングが停止します。

サンプリング開始例を以下に示します。  
`outp( ADR+18, 0x3 );`

**■変換データの入力**  
 変換データの入力は、レジスタ内に変換データが格納されていることを確認してから行います。変換動作中であってもレジスタから変換データを入力することができます。変換データの入力の手順を右図に示します。変換データはオフセットバイナリで、変換データと入力電圧(電流)の関係は、次の式で示されます。

$$\text{データ} = \frac{(\text{電圧(電流)} + \text{オフセット})}{\text{スパン}} \times 2^{16}$$

←

入力レンジ	オフセット	スパン	入力レンジ	オフセット	スパン
-10V ~ +10V	10	20	0mA ~ 20mA	0	20

```

        graph TD
            Start([スタート]) --> Analog[アナログ入力ステータス]
            Analog --> DRE{DREステータス  
確認}
            DRE --> Input[/入力データ/]
            Input --> End([終了])
            DRE --> Start
            
```

**図4.21 変換データの  
入力手順**

◎ CONTEC  
ADI16-4(FIT)GY

31

図.18 変換方法

## 別紙

ここで、騒音計 NL-21 の出力電圧をデジタル値に変換した例を表.1 に示す。(測定レンジに 20~100dB を選択した場合)

表.1 NL-21 出力電圧とデジタル値の関係

測定値(dB)	出力電圧(V)	デジタル値
20	0.50	-31130
30	0.75	-30311
40	1.00	-29492
50	1.25	-28672
60	1.50	-27853
70	1.75	-27034
80	2.00	-26215
90	2.25	-25396
100	2.50	-24576

また、振動計 VM-52 の出力電圧をデジタル値に変換した例を表.2 に示す。(測定レンジに 30~80dB を選択した場合)

表.2 VM-52 出力電圧とデジタル値の関係

測定値(dB)	出力電圧(V)	デジタル値
30	0.5	-31130
40	1.0	-29492
50	1.5	-27853
60	2.0	-26215
70	2.5	-24576
80	3.0	-22938

このオフセット・バイナリ方式によるデジタル値が、ボックス・コンピューターへ入力されるため、ボックス・コンピューターでは、乙23の28頁目以下、原告第11準備書面3頁目の図1のコードの中に記載された計算により、振動計・騒音計が出力した電圧値を導く。

具体的には、同準備書面の別図2末尾にある、「A/D 変換器で得た変換値から電圧を求めるプログラム」である①式「 $(\text{InpBuf} - 32768) / 3276.8$ 」、②式「 $(\text{InpBuf} + 32768) / 3276.8$ 」の各「InpBuf」に、前頁の表.1 及び表.2 のデジタル値を入力して計算する。

## 別紙

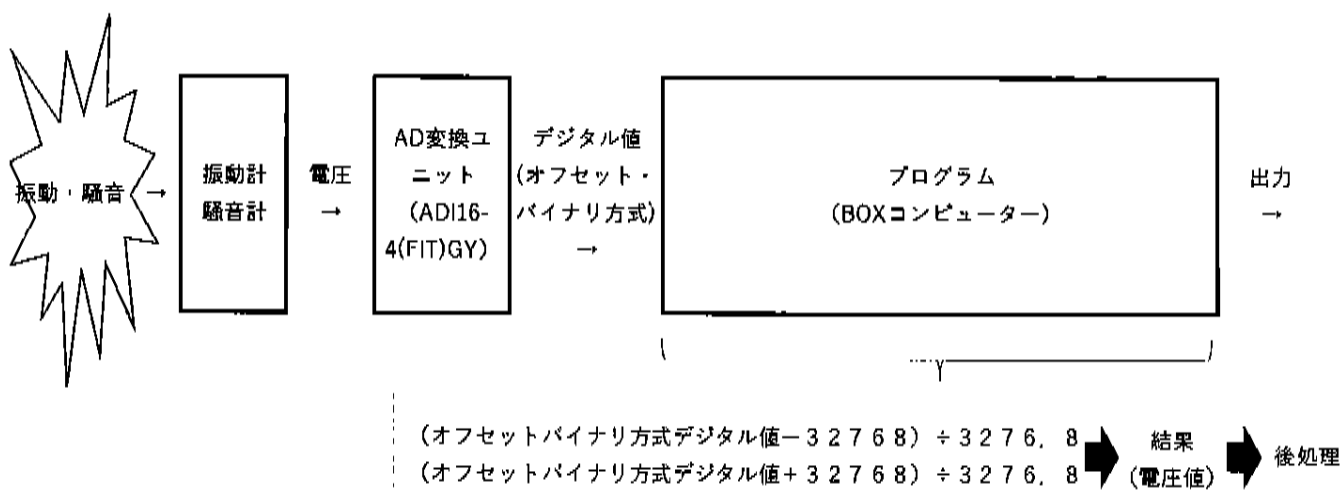
表.1の20dBの場合を例にする。

表.1の20dBの場合、騒音計の出力電圧は0.5Vとなり、AD変換器により、ボックス・コンピューターへはオフセット・バイナリ方式の「-31130」というデジタル値が入力される。

この入力されたデジタル値「-31130」は、原告第11準備書面別図2の「(InpBuf-32768)/3276.8」及び「(InpBuf+32768)/3276.8」の各「InpBuf」に当てはめられ、それぞれ(-31130-32768)/3276.8=-19.5、(-31130+32768)/3276.8=0.5と計算され、if文による採用で、結果は後者の0.5Vとなる。この0.5Vという値を基に、ボックス・コンピューターは処理を行う。

以上を図Bで示す。

図B



↑乙23の28頁目以下に記載

この点、原告第11準備書面3頁目の図1の被告のコード（乙23の28頁目以下）について、専門委員から「ボックス・コンピューターにて、2の補数は当たり前のこと。原告第11準備書面3頁目の図1の被告のコードのような記述は普通行わない。」との御指摘を受け、被告は裁判所から説明を求められている。専門委員の御指摘の主旨は、「2の補数は当たり前」という言葉から、プログラムではデジタル値の表現方法としてコンプリメント・バイナリ方式が当たり前であって、この常識に原告第11準備書面3頁目の図1の被告のコードは反するという主旨と理解した。この専門委員の御指摘は、上述した裁判所の質問事項（プログラム関連）の「ボックス・コンピューターのAD変換器から出力されるデジタルデータ変換値InpBuf」という記載が正の事実であることを前提に行われているのではないかと推察するが（ボックス・コンピューターが

## 別紙

プログラム処理するデジタル値の表現方法はコンプリメント・バイナリ方式であるから、そこから出力されるデジタル値もコンプリメント・バイナリ方式である)、上述のとおり裁判所の質問は入出力関係を真逆に記述しており、事実として、ここでのデジタル値は、ボックス・コンピューターから出力されるのではなく、ボックス・コンピューターへ入力されるものである。そして、その入力される表現方法はオフセット・バイナリ方式である。専門委員の御指摘のとおりボックス・コンピューターは、コンプリメント・バイナリ方式の表現方法でデジタル値を処理するが、AD 変換器よりオフセット・バイナリ方式で入力されるデジタル値から騒音計・振動計が出力した電圧値を正しく導き出すために、原告第11準備書面3頁目の図1の被告のコードが記載されている。

以上が、専門委員のご指摘に対する説明となる。

別紙

求釈明事項（3）

乙23のソースコード（17頁以降）について、乙38、39を参照しながら、各コードがいかなる処理を行っているのか、説明してください。特に、乙3、4によれば、乙23のソースコードは、(i)外部表示装置への電光掲示、(ii)警報出力の設定、(iii)黄色、赤色回転灯（警報）の作動、(iv)10Hzでのサンプリング及び0.1SEC～1SECでのサンプリングタイムの任意設定、(v)各種統計データ（Lp、Leq、L05等）の算出、(vi)測定データ保存の機能を行い得るものとされており、(i)～(vi)の各機能について、ソースコードの該当部分及び関連部分を明示するとともに、該当部分及び関連部分でどのような処理を行っているのか、順を追って、説明してください。

(3) - 1. 乙23の実装機能について

乙23のソースコードは平成16年2月6日時点のものであり、乙3（騒音振動監視システム 製作仕様書 平成15年10月）にしたがって作成されているが、開発途中バージョンであるため一部の機能は実装されていない。外部表示装置への電光掲示について、乙3の9頁のイメージ写真及び外観図のとおり、この時点では予定されていない。また、乙4（サイレントロボカタログ）は現行機のカタログであるため、乙23のソースコードには無い機能が記載されている。質問として挙げられた機能のうち実装されていない機能を以下に示す。

- (i) 外部表示装置への電光掲示 ← 現行機の機能
- (iv) 10Hzでのサンプリング及び0.1SEC～1SECでのサンプリングタイムの任意設定  
← 平成16年2月6日時点では未実装

(3) - 2. 乙23の構成

乙23のソースコードは図. 19に示す構成となっている。

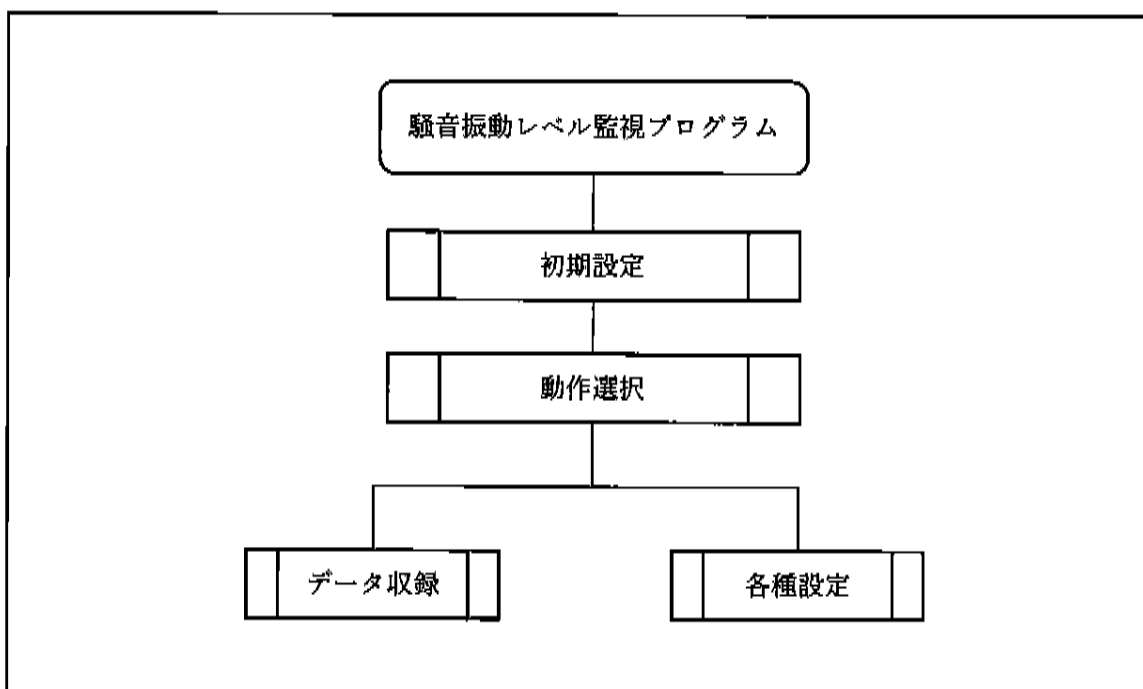


図. 19 システム構成

別紙

(3) - 3. 初期設定

ページ	ソースコード	説明	機能/参考資料
22	<pre> Private Sub Form_Load()     _____     ' Initialized data     _____     ' Text_DeviceName.Text = "D1000"     PortNoInf.Text = 0          ' Set the port number to input/output data (Default)     BitNoInf.Text = 0          ' Set the bit number to input/output data (Default)     ' Centering     Left = (Screen.Width - Width) / 2     Top = (Screen.Height - Height) / 2      Timer1.Enabled = False     Timer1.Interval = 1     MSCOMM1.PortOpen = True     ModeRS1 = 0     lbl騒音計LA = "0.0"     lbl騒音計L05 = "0.0"     KeihoSouonDb1 = GetSetting("騒音警報プログラム", "設定", "警報騒音1", 75#)     KeihoSouonDb2 = GetSetting("騒音警報プログラム", "設定", "警報騒音2", 85#)     KeihoSindouDb1 = GetSetting("騒音警報プログラム", "設定", "警報振動1", 60#)     KeihoSindouDb2 = GetSetting("騒音警報プログラム", "設定", "警報振動2", 75#)     RangeSouonDb = GetSetting("騒音警報プログラム", "設定", "騒音レンジ", 110#)     RangeSindouDb = GetSetting("騒音警報プログラム", "設定", "振動レンジ", 80#)      Hyouka1 = GetSetting("騒音警報プログラム", "設定", "評価時間", 100#)      Command4_Glick ' -----工場出荷に戻す         </pre>	<p>プログラムを起動すると、ここから 実行される</p> <p>警報出力パラメータ初期化</p> <p>画面中央に表示するようにする</p> <p>タイマー初期化</p> <p>騒音レベル表示を0.0にする</p> <p>各種設定で設定した警報管理値を 読み込む</p> <p>外部表示器初期化 (未実装)</p>	<p>(ii) 警報出力の設定</p>

ページ	ソースコード	説明	機能/参考資料
22	<pre> 警報騒音 1.Caption = Format(KeihoSouonDb1, "#00.0") 警報騒音 2.Caption = Format(KeihoSouonDb2, "#00.0") 警報振動 1.Caption = Format(KeihoSindouDb1, "#00.0") 警報振動 2.Caption = Format(KeihoSindouDb2, "#00.0")  評価Time.Caption = Format(Hyouka1, "#000") FormBackColorFlg = False DataWriteFlg = False Timer2.Interval = 1000 Timer2.Enabled = True  On Error GoTo Error MkDir "c:*共有" MkDir "c:*共有*騒音振動監視" MkDir "c:*共有*騒音振動監視\data" OneTimeStartFlag = False  Exit Sub Error: End Sub </pre>	<p>警報管理値の表示</p> <p>「開始」ボタン押し忘れタイマー起 動（100秒）</p> <p>データ保存先フォルダの作成</p> <p>収録中フラグのクリア</p>	

別紙

(3) - 4. 動作選択

動作は画面上のボタンをクリックする事で選択する。(図. 20)

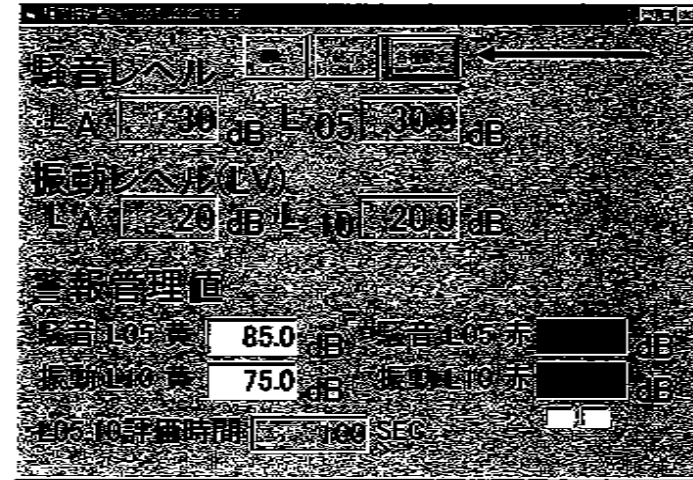


図. 20 動作選択

ページ	ソースコード	説明	機能/参考資料
20	<pre>Private Sub Command1_Click() '-----設定 If Timer1.Enabled = True Then MsgBox "設定変更は計測中は出来ません。" Exit Sub End If  Load Form2 Form2.Show 1   警報騒音 1.Caption = Format(KeihoSouonDb1, "#00.0")   警報騒音 2.Caption = Format(KeihoSouonDb2, "#00.0")    警報振動 1.Caption = Format(KeihoSindouDb1, "#00.0")   警報振動 2.Caption = Format(KeihoSindouDb2, "#00.0")    評価 Time.Caption = Format(Hyouka1, "#000")  End Sub</pre>	<p>「各種設定」ボタン処理</p> <p>データ収録中は設定変更できない</p> <p>各種設定画面を表示</p> <p>変更された警報管理値等の再表示</p>	(ii) 警報出力の設定

別紙

ページ	ソースコード	説明	機能/参考資料
21	<pre>Private Sub Command2_Click() '-----終了 EndFlag = True OneTimeStartFlag = False  Timer1.Enabled = False If Val (PortDataInf.Text) &gt; 0 Then '回転等 消灯 Cmd_DioOpen_Click PortDataInf.Text = "0" Cmd_DioOutByte_Click Cmd_DioClose_Click End If  Exit Sub</pre>	<p>「終了」ボタン処理</p> <p>収録中フラグのクリア</p> <p>タイマーの停止</p> <p>回転灯を消灯する</p> <p>(以下に詳細)</p>	<p>(iii)黄色、赤色回転灯（警報）の作動</p>
19	<pre>Private Sub Cmd_DioOpen_Click() '----- ' Set parameter '----- Ret2 = DioOpenEx("DI000", hDrv2) RetInf = "DioOpenEx Return Code = " &amp; Right("000" + Hex(Ret2), 3) &amp; "h" '----- ' Error Check '-----  If Ret2 &lt;&gt; 0 Then Exit Sub End If  End Sub</pre>	<p>接点出力装置をオープンする</p>	<p>乙39 20ページ</p> <p>DioOpen関数</p>

別紙

ページ	ソースコード	説明	機能/参考資料
21	<pre> Private Sub Cmd_DioOutByte_Click()     ' _____     ' Set parameter     ' _____     PortNo = Val (PortNoInf.Text)      ' Set port number to PortNo     Buf = Val("&amp;H" + PortDataInf.Text) ' Output data     Ret2 = DioOutByte (hDrv2, PortNo, Buf)     RetInf = "DioOutByte Return Code = " &amp; Right("000" + Hex(Ret2), 3) &amp; "h"     ' _____     ' Error Check     ' _____     If Ret2 &lt;&gt; 0 Then         Exit Sub     End If End Sub  Private Sub Cmd_DioClose_Click()     ' _____     ' Close process     ' _____     Ret2 = DioClose (hDrv2)          ' Close process     RetInf = "DioClose Return Code = " &amp; Right("000" + Hex(Ret2), 3) &amp; "h" End Sub </pre>	<p>接点出力装置に回転灯制御データを送る</p> <p>PortDataInf.Text : 制御データ</p> <p>3:赤・黄点灯 2:黄点灯 1:赤点灯 0:消灯</p> <p>接点出力装置をクローズする</p>	<p>(iii)黄色、赤色回転灯（警報）の作動</p> <p>乙39 39ページ</p> <p>DioOutByte 関数</p> <p>乙39 23ページ</p> <p>DioClose 関数</p>



別紙

ページ	ソースコード	説明	機能/参考資料
22	<pre>Dim tempCount As Long tempCount = TimeValue(Time) * 60 * 60 * 24 vDataCount = tempCount KeihoFig = 0 Time0 = (Val(Right(Time, 1)) + 1) Mod 10  Timer2.Enabled = False Cmd_AioOpen_Click  End Sub</pre>	<p>「開始」ボタン押し忘れタイマー停止</p> <p>AD変換装置の初期化</p> <p>(以下に詳細)</p>	
18	<pre>'ドライバを初期化します Private Sub Cmd_AioOpen_Click()      Ret = AioOpenEx(TxtDevice, hDrv) '初期化を行いデバイスハンドルを取得し     ます     '0以外はエラーです     RetInf = "AioOpenEx Return Code = " &amp; Right("000" + Hex(Ret), 3) &amp; "h"      DatDsp = ""  End Sub</pre>	<p>AD変換装置をオープンする</p>	<p>Z38 2ページ</p> <p>AioOpenEx 関数</p>

別紙

(3) - 5. データ収録

ページ	ソースコード	説明	機能/参考資料
22	<pre>Private Sub Timer1_Timer() Static WDTCount As Integer  'Exit Sub  If Right(Time, 1) = Time0 Then     TimePoint = (Val(Left(Right(Time, 5), 2)) * 600 + Val(Right(Time, 2)) * 10) * 2     Cmd_AioInpBack_Click     Stop1     Time0 = (Time0 + 1) Mod 10     WDTCount = 0     'MsgOkFig = False End If WDTCount = WDTCount + 1 If WDTCount &gt; 2000 Then Text2.BackColor = RGB(255, 0, 0)  End Sub</pre>	<p>タイマーにより秒替わりをチェックする</p> <p>秒替わりが発生したら</p> <ul style="list-style-type: none"> <li>・ AD 変換をスタート</li> <li>・ 変換データの処理</li> </ul> <p>を行う</p>	(iv) 10 Hz でのサンプリング
18	<pre>'A/D 変換を行います Private Sub Cmd_AioInpBack_Click()      Dim i As Integer      StsInf = ""     CntInf = ""     ' DatDsp = ""      'ソフトウェアスタート     InpMode = 0</pre>	AD 変換スタート処理	

ページ	ソースコード	説明	機能/参考資料
18	<pre> '変換を行う順番を決定します For i = 0 To 1     InpChNo(i) = i Next i '配列の先頭アドレスを代入します  LpAinpB.InpBuf = LpWord(InpBuf(TimePoint)) LpAinpB.InpChNo = LpWord(InpChNo(0)) '2チャンネル変換します LpAinpB.Channels = 2 '10回変換します LpAinpB.Scan = 10 'システムタイマを使用して変換します LpAinpB.TimType = 0 '100msecで変換します(10 × 10ms) LpAinpB.Tim = 9.3 MsgOK.Message = MEG_OK      '変換完了イベントメッセージ番号 MsgERR.Message = MEG_EXCP   '例外発生イベントメッセージ番号  Ret = AioInpBack(hDrv, InpMode, LpAinpB, MsgOK.hWnd, MsgOK.Message, MsgERR.hWnd, MsgERR.Message) 'リターンコードを表示します RetInf = "AioInpBack Return Code = " &amp; Right("000" + Hex(Ret), 3) &amp; "h"  End Sub </pre>	<p>変換 CH は 0CH と 1CH</p> <p>0CH:騒音 1CH:振動</p> <p>変換 CH 数 = 2</p> <p>変換回数 = 10 回</p> <p>サンプリングピッチ = 9ms</p> <p>AD 変換スタート</p> <p>(AD 変換は本プログラムとは非同 期で実行される)</p>	<p>(iv) 10 Hz でのサンプリング</p> <p>乙 3 8 31 ページ</p> <p>AINPB 構造体</p> <p>乙 3 8 10 ページ</p> <p>AioInpBack 関数</p>
23	<pre> Private Sub Syori1() Dim aaa As String Dim aaaLen As Integer Static tmpDataS(3600) As Single Static tmpDataV(3600) As Single Static tmpDataCountS(3600) As Long Static tmpDataCountV(3600) As Long Static tmpDataNow As Integer Static tmpDataBefore As Integer Static tmpDataSerNo As Integer Static tmpOneLoop As Boolean  Dim tmpMaxInpBuf1Sec0 As Integer Dim tmpMaxInpBuf1Sec1 As Integer </pre>	<p>変換データの処理</p>	<p>(v) 各種統計データ (Lp、Leq、L 05等) の算出</p>

<p>23</p>	<pre>Dim tmpDataSerNo0 As Integer Dim tmpData0 As Single Dim tmpData1 As Single Dim tmpDataCount0 As Long Dim tmpDataCount1 As Long  Dim tmpLOSCount As Integer Dim tmpL10Count As Integer  ' MSComm1.Output = Chr(2) &amp; Chr(1) &amp; "C" &amp; "D000?" &amp; Chr(&amp;H3) &amp; Chr(0) &amp; vbCrLf &amp; vblf ' aaa = "" ' Do While MSComm1.InBufferCount &gt; 0 '   aaa = aaa + MSComm1.input ' Loop If TimePoint = 0 Then TimePoint = 72000 - 20 ' If TimePoint = 20 Then TimePoint = 72000 tmpMaxInpBuf1Sec0 = InpBuf(TimePoint - 20) For i = 2 To 18 Step 2   If tmpMaxInpBuf1Sec0 &gt; InpBuf(TimePoint - 20 + i) Then tmpMaxInpBuf1Sec0 = InpBuf(TimePoint - 20 + i) Next i tmpMaxInpBuf1Sec1 = InpBuf(TimePoint - 20 + 1) For i = 3 To 19 Step 2   If tmpMaxInpBuf1Sec1 &gt; InpBuf(TimePoint - 20 + i) Then tmpMaxInpBuf1Sec1 = InpBuf(TimePoint - 20 + i) Next i</pre>	<p>テストプログラムの残骸 （コメントアウト）</p> <p>1秒前の変換データから1秒間の最大値を探す（バグ）</p> <p>tmpMaxInpBuf1Sec0:騒音データ tmpMaxInpBuf1Sec1:振動データ</p>	
<p>24</p>	<pre>If tmpMaxInpBuf1Sec0 &gt;= 0 Then   lbl騒音計LA = Int(((tmpMaxInpBuf1Sec0 - 32768) / 3276.8 * 20 + (RangeSouonnDb - 70)) * 10) / 10 Else   lbl騒音計LA = Int(((tmpMaxInpBuf1Sec0 + 32768) / 3276.8 * 20 + (RangeSouonnDb - 70)) * 10) / 10 End If  If tmpMaxInpBuf1Sec1 &gt;= 0 Then   lbl振動計LA = Int(((tmpMaxInpBuf1Sec1 - 32768) / 3276.8 * 20 + (RangeSindouDb - 60)) * 10) / 10 Else</pre>	<p>変換データを(dB)に変換する</p> <p>lbl騒音計LA:騒音レベル(LA)</p> <p>lbl振動計LA:振動レベル(LA)</p>	<p>準備書面12「2. 変換処理について (2)」</p>

<p>24</p>	<pre> lbl 振動計 LA = Int(((tmpMaxInpBuf1Sec1 + 32768) / 3276.8 * 20 + (RangeSindouDb - 60)) * 10) / 10 End If  ' If Val(lbl 騒音計 LA) = 0 Then '     MSComm1.Output = Chr(2) &amp; Chr(1) &amp; "C" &amp; "D000?" &amp; Chr(&amp;H3) &amp; Chr(0) &amp; vbCr &amp; vbCrLf '     aaa = "" '     Do While MSComm1.InBufferCount &gt; 0 '         aaa = aaa + MSComm1.Input '     Loop '     lbl 騒音計 LA = Mid(aaa, 4, 5) ' End If vData(vDataCount, 0) = Val(lbl 騒音計 LA) vData(vDataCount, 1) = Val(lbl 振動計 LA)  ' If vDataCount &lt; 100 Then Exit Sub  -----騒音 L05L10 計算 Dim LastCount As Integer If OneTimeStartFlag = True Then '1 回目だけの処理     tmpDataSerNo = 0     OneTimeStartFlag = False     tmpDataS(tmpDataSerNo) = vData(vDataCount, 0)     tmpDataCountS(tmpDataSerNo) = vDataCount     tmpOneLoop = True     tmpDataNow = Right(Time, 1)     GoTo INCLINENT End If ' tmpDataNow = (tmpDataNow) Mod 10 ' If Right(TimePoint / 20, 1) &lt;&gt; tmpDataNow Then '     Text1.BackColor = RGB(255, 0, 0) ' Else '     Text1.BackColor = &amp;H8000000F ' End If </pre>	<p>テストプログラムの残骸 (コメントアウト)</p> <p>1 秒毎のデータを配列に保存する</p> <p>騒音レベル(L05)を求める</p> <p>最初のデータは無条件に配列に追加する。</p> <p>テストプログラムの残骸 (コメントアウト)</p>	<p>(v)各種統計データ (Lp、Leq、L05等) の算出</p>
-----------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------

24	<pre> If tmpOneLoop = True Then '1 周回のみ処理   tmpDataS(tmpDataSerNo) = vData(vDataCount, 0)   tmpDataCountS(tmpDataSerNo) = vDataCount   tmpDataSerNo0 = tmpDataSerNo   For i = tmpDataSerNo - 1 To 0 Step -1     If tmpDataS(i) &lt; tmpDataS(i + 1) Then       tmpData0 = tmpDataS(i)       tmpData1 = tmpDataS(i + 1)       tmpDataS(i) = tmpData1       tmpDataS(i + 1) = tmpData0       tmpDataCount0 = tmpDataCountS(i)       tmpDataCount1 = tmpDataCountS(i + 1)       tmpDataCountS(i) = tmpDataCount1       tmpDataCountS(i + 1) = tmpDataCount0      End If   Next i Else   tmpDataCount0 = tmpDataCountS(0)   For i = 0 To Hyouka1 - 1 - 1     If tmpDataCount0 &gt; tmpDataCountS(i + 1) Then       tmpDataCount0 = tmpDataCountS(i + 1)       LastCount = i + 1     End If   Next i   tmpDataS(LastCount) = vData(vDataCount, 0)   tmpDataCountS(LastCount) = vDataCount   tmpDataSerNo0 = tmpDataSerNo    If LastCount = 0 Then     i = i   ElseIf tmpDataS(LastCount) = tmpDataS(LastCount - 1) Then     i = i   ElseIf tmpDataS(LastCount) &gt; tmpDataS(LastCount - 1) Then     For i = LastCount - 1 To 0 Step -1       If tmpDataS(i) &lt; tmpDataS(i + 1) Then         tmpData0 = tmpDataS(i)         tmpData1 = tmpDataS(i + 1)         tmpDataS(i) = tmpData1         tmpDataS(i + 1) = tmpData0       End If     Next i   End If </pre>	<p>騒音レベル(L05)の処理に必要なだけのデータが配列に保存されていないならば、配列にデータを保存する</p> <p>保存したデータを昇順に並べ替える</p> <p>騒音レベル(L05)の処理に必要なだけのデータが配列に保存されていたら一番古いデータを収録データと入れ替える</p> <p>収録データを入れ替えた事により昇順の並びが崩れた場合は、再度昇順に並び替える</p>	
----	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

25	<pre> tmpDataCount0 = tmpDataCountS(i) tmpDataCount1 = tmpDataCountS(i + 1) tmpDataCountS(i) = tmpDataCount1 tmpDataCountS(i + 1) = tmpDataCount0 End If Next i ElseIf tmpDataS&gt;LastCount) &lt; tmpDataS&gt;LastCount - 1) Then For i = LastCount To Hyouka1 - 1 If tmpDataS(i) &lt; tmpDataS(i + 1) Then tmpData0 = tmpDataS(i) tmpData1 = tmpDataS(i + 1) tmpDataS(i) = tmpData1 tmpDataS(i + 1) = tmpData0 tmpDataCount0 = tmpDataCountS(i) tmpDataCount1 = tmpDataCountS(i + 1) tmpDataCountS(i) = tmpDataCount1 tmpDataCountS(i + 1) = tmpDataCount0 End If Next i End If tmpL05Count = Int(Hyouka1 * 0.05) tmpL10Count = Int(Hyouka1 * 0.1)  b 騒音計 L05 = Format(tmpDataS(tmpL05Count), "#00.0")  b 騒音計 L10 = Format(tmpDataS(tmpL10Count), "#00.0") '-----振動 L05L10 計算 If OneTimeStartFlag = True Then '1 回目だけの処理 tmpDataSerNo = 0 OneTimeStartFlag = False tmpDataV(tmpDataSerNo) = vData(vDataCount, 1) tmpDataCountV(tmpDataSerNo) = vDataCount tmpOneLoop = True GoTo INCLMENT End If tmpDataNow = (tmpDataNow + 1) Mod 10 If tmpOneLoop = True Then '1 周回だけの処理 tmpDataV(tmpDataSerNo) = vData(vDataCount, 1) tmpDataCountV(tmpDataSerNo) = vDataCount tmpDataSerNo0 = tmpDataSerNo For i = tmpDataSerNo - 1 To 0 Step -1 If tmpDataV(i) &lt; tmpDataV(i + 1) Then tmpData0 = tmpDataV(i) </pre>	<p>騒音レベル(L05)のデータを取り出し表示する (昇順に並んだデータの上位 5%の位置にあるデータ) 振動レベル(L10)を求める</p> <p>最初のデータは無条件に配列に追加する。</p> <p>振動レベル(L10)の処理に必要なだけのデータが配列に保存されていなければ、配列にデータを保存する</p>	<p>(v)各種統計データ (Lp、Leq、L05等)の算出</p>
----	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------

<pre> tmpData1 = tmpDataV(i + 1) tmpDataV(i) = tmpData1 tmpDataV(i + 1) = tmpData0 tmpDataCount0 = tmpDataCountV(i) tmpDataCount1 = tmpDataCountV(i + 1) tmpDataCountV(i) = tmpDataCount1 tmpDataCountV(i + 1) = tmpDataCount0  End If Next i Else Dim LastCount As Integer tmpDataCount0 = tmpDataCountV(0) For i = 0 To Hyouka1 - 1 - 1 If tmpDataCount0 &gt; tmpDataCountV(i + 1) Then tmpDataCount0 = tmpDataCountV(i + 1) LastCount = i + 1 End If Next i tmpDataV(LastCount) = vData(vDataCount, 1) tmpDataCountV(LastCount) = vDataCount tmpDataSerNo0 = tmpDataSerNo  If LastCount = 0 Then i = i ElseIf tmpDataV(LastCount) = tmpDataV(LastCount - 1) Then i = i ElseIf tmpDataV(LastCount) &gt; tmpDataV(LastCount - 1) Then For i = LastCount - 1 To 0 Step -1 If tmpDataV(i) &lt; tmpDataV(i + 1) Then tmpData0 = tmpDataV(i) tmpData1 = tmpDataV(i + 1) tmpDataV(i) = tmpData1 tmpDataV(i + 1) = tmpData0 tmpDataCount0 = tmpDataCountV(i) tmpDataCount1 = tmpDataCountV(i + 1) tmpDataCountV(i) = tmpDataCount1 tmpDataCountV(i + 1) = tmpDataCount0 End If Next i ElseIf tmpDataV(LastCount) &lt; tmpDataV(LastCount - 1) Then For i = LastCount To Hyouka1 - 1 </pre>	<p>保存したデータを昇順に並べ替える</p> <p>振動レベル(L10)の処理に必要なだけのデータが配列に保存されていたら一番古いデータを収録データと入れ替える</p> <p>収録データを入れ替えた事により昇順の並びが崩れた場合は、再度昇順に並び替える</p>	
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------	--

<pre> If tmpDataV(i) &lt; tmpDataV(i + 1) Then     tmpData0 = tmpDataV(i)     tmpData1 = tmpDataV(i + 1)     tmpDataV(i) = tmpData1     tmpDataV(i + 1) = tmpData0     tmpDataCount0 = tmpDataCountV(i)     tmpDataCount1 = tmpDataCountV(i + 1)     tmpDataCountV(i) = tmpDataCount1     tmpDataCountV(i + 1) = tmpDataCount0 End If Next i End If End If *tmpL05Count = Int(Hyouka1 * 0.05) tmpL10Count = Int(Hyouka1 * 0.1)  b  振動計 L05 = Format(tmpDataV(tmpL05Count), "#00.0")  b  振動計 L10 = Format(tmpDataV(tmpL10Count), "#00.0") *tmpDataCountS(3600) = vDataCount  ** If vDataCount &lt; Hyouka1 Then **   For i = 0 To vDataCount **     tmpDataS(i) = vData(86400 - vDataCount + i, 0) **   Next i **   For i = vDataCount + 1 To Hyouka1 - 1 **     tmpDataS(i) = vData(vDataCount - Hyouka1 - 1 + i, 0) **   Next i ** Else **   For i = 0 To Hyouka1 - 1 **     tmpDataS(i) = vData(vDataCount - Hyouka1 - 1 + i, 0) **   Next i ** End If ** Dim L05Count As Integer ** L05Count = Int((Hyouka1 - 1) * 0.05) ** ** For i = 0 To Hyouka1 - 2 **   If tmpDataS(i) &gt; tmpDataS(i + 1) Then **     tmpData0 = tmpDataS(i) **     tmpData1 = tmpDataS(i + 1) **     tmpDataS(i) = tmpData1 **     tmpDataS(i + 1) = tmpData0 **   End If ** Next i </pre>	<p>振動レベル(L10)のデータを取り出し表示する (昇順に並んだデータの上位 10%の位置にあるデータ)</p> <p>テストプログラムの残骸 (コメントアウト)</p>	
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------	--

26	<pre> 警報判別  If Val ( b  騒音計 L05) &gt;= KeihoSouonDb2 Then   KeihoFlg = KeihoFlg Or &amp;H1 Else   KeihoFlg = KeihoFlg And &amp;HE End If If Val ( b  振動計 L10) &gt;= KeihoSindouDb2 Then   KeihoFlg = KeihoFlg Or &amp;H2 Else   KeihoFlg = KeihoFlg And &amp;HD End If If Val ( b  騒音計 L05) &gt;= KeihoSouonDb1 Then   KeihoFlg = KeihoFlg Or &amp;H4 Else   KeihoFlg = KeihoFlg And &amp;HB End If If Val ( b  振動計 L10) &gt;= KeihoSindouDb1 Then   KeihoFlg = KeihoFlg Or &amp;H8 Else   KeihoFlg = KeihoFlg And &amp;H7 End If </pre>	<p>警報判定</p> <p>騒音レベル(L05)、振動レベル(L10)と警報管理値を比較する。</p> <p>警報管理値を超えた場合 KeihoFlg の該当ビットを ON にする。</p> <p>Bit 0:騒音レベル ≧ 管理値(赤)  1:振動レベル ≧ 管理値(赤)  2:騒音レベル ≧ 管理値(黄)  3:振動レベル ≧ 管理値(黄)</p>	(iii)黄色、赤色回転灯(警報)の作動
27	<pre> If (KeihoFlg &gt; 3) And ((KeihoFlg And &amp;H3) &gt; 0) Then   Cmd_DioOpen_Click   PortDataInf.Text = "3"   Cmd_DioOutByte_Click   Cmd_DioClose_Click ElseIf (KeihoFlg &gt; 3) And ((KeihoFlg And &amp;H3) = 0) Then   Cmd_DioOpen_Click   PortDataInf.Text = "2"   Cmd_DioOutByte_Click   Cmd_DioClose_Click ElseIf KeihoFlg And &amp;H10 &gt; 0 Then   Cmd_DioOpen_Click   PortDataInf.Text = "1"   Cmd_DioOutByte_Click   Cmd_DioClose_Click Else   Cmd_DioOpen_Click   PortDataInf.Text = "0"   Cmd_DioOutByte_Click </pre>	<p>KeihoFlg のビットの状態によって回転灯を制御する。</p> <ul style="list-style-type: none"> <li>・(Bit 2 又は 3 が ON)かつ(Bit0 又は 1 が ON) → 黄・赤回転灯 点灯</li> <li>・(Bit 2 又は 3 が ON)かつ(Bit0、1 が OFF) → 黄回転灯 点灯</li> <li>・(Bit 2、3 が OFF)かつ(Bit0 又は 1 が ON) → 赤回転灯 点灯</li> <li>・それ以外は黄・赤回転灯 消灯</li> </ul>	<p>回転灯制御は本資料 14～15 ページ参照</p>

<pre> Cmd_DioClose_Click  End If ' If FormBackColorFlg = False Then '   Main1.BackColor = &amp;H8000000F '   FormBackColorFlg = True ' Else '   Main1.BackColor = RGB(255, 0, 0) '   FormBackColorFlg = False ' End If ' If Val (PortDataInf.Text) &lt; 3 Then '回転等 赤黄点灯 '   Cmd_DioOpen_Click '   PortDataInf.Text = "3" '   Cmd_DioOutByte_Click '   Cmd_DioClose_Click ' End If  ' If FormBackColorFlg = False Then '   Main1.BackColor = &amp;H8000000F '   FormBackColorFlg = True ' Else '   Main1.BackColor = RGB(255, 255, 0) '   FormBackColorFlg = False ' End If ' If Val (PortDataInf.Text) &lt; 1 Or Val (PortDataInf.Text) = 3 Then '回転等 黄 点灯 '   Cmd_DioOpen_Click '   PortDataInf.Text = "1" '   Cmd_DioOutByte_Click '   Cmd_DioClose_Click ' End If ' Else '   Main1.BackColor = &amp;H8000000F '   FormBackColorFlg = False ' If Val (PortDataInf.Text) &gt; 0 Then '回転等 消灯 '   Cmd_DioOpen_Click '   PortDataInf.Text = "0" '   Cmd_DioOutByte_Click '   Cmd_DioClose_Click ' End If ' End If </pre>	<p>テストプログラムの残骸 (コメントアウト)</p>	
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------	--

27	<pre> If Val (Right (Time, 1)) &lt;&gt; (TimePoint / 20) Mod 10 Then     Text1.BackColor = RGB (255, 0, 0) End If '-----data保存 毎正時 If (Right (Time, 5) = "59:50" Or Right (Time, 5) = "59:51") And DataWriteFlg = False Then     Dim FileNameCSV As String     Dim FileNo As Integer     Dim wData (36000, 1) As Single     Dim FileNameCSV As String     FileNameCSV = Format (Date, "yyyymmdd")     FileNameCSV = "C:\共有\騒音振動監視\data*" &amp; FileNameCSV &amp; Format (Time, "hh")     &amp; ".CSV"     FileNo = FreeFile ()     Open FileNameCSV For Output As #FileNo     For ii = 0 To 35899         If InpBuf (ii * 2) &gt;= 0 Then             wData (ii, 0) = Int (((InpBuf (ii * 2) - 32768) / 3276.8 * 20 + (RangeSouonDb - 70)) * 10) / 10         Else             wData (ii, 0) = Int (((InpBuf (ii * 2) + 32768) / 3276.8 * 20 + (RangeSouonDb - 70)) * 10) / 10         End If          If InpBuf (ii * 2 + 1) &gt;= 0 Then             wData (ii, 1) = Int (((InpBuf (ii * 2 + 1) - 32768) / 3276.8 * 20 + (RangeSindouDb - 60)) * 10) / 10         Else             wData (ii, 1) = Int (((InpBuf (ii * 2 + 1) + 32768) / 3276.8 * 20 + (RangeSindouDb - 60)) * 10) / 10         End If          Next ii         For ii = 0 To 35899             DoEvents             Write #FileNo, wData (ii, 0), wData (ii, 1)         Next ii         Close #FileNo     End For      If TimeValue (Time) = 0 Or vDataCount &gt; 86400 Then vDataCount = 0 'カウントク リア </pre>	<p>データ保存</p> <p>xx 時 59 分 50 秒又は xx 時 59 分 51 秒でデータが未保存であれば保存処理を行う</p> <p>ファイル名は"年月日時.CSV"</p> <p>保存するデータ数は 35900</p> <p>(秒 10 データ×59 分 50 秒)</p> <p>変換データを(dB)に変換する</p> <p>wData(ii, 0):騒音レベル(dB)</p> <p>wData(ii, 1):振動レベル(dB)</p> <p>ファイルに保存する</p> <p>日替わり発生でデータ保存カウンタリア</p>	<p>(vi)測定データ保存</p> <p>準備書面 1 2 「2. 変換処理について (2)」</p>
----	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------

<pre> DataWriteFlg = True For i = 0 To 3600     tmpDataCountS(i) = 0     tmpDataS(i) = 0 Next i  Elseif Left(Right(Time, 5), 4) = "00:0" And DataWriteFlg = True Then      DataWriteFlg = False End If  INCLIMENT: '-----各種カウンタをインクリメント If tmpDataSerNo &gt;= Hyouka! - 1 Then     If tmpOneLoop = True Then         tmpOneLoop = False         tmpDataSerNo = 0     Else         tmpDataSerNo = 0     End If Else     tmpDataSerNo = tmpDataSerNo + 1 End If vDataCount = vDataCount + 1 Exit Sub  ERROR: MsgBox "error"  'Label5.Caption = Format(Rnd * 100, "##0.0") 'Label6.Caption = Format(Rnd * 100, "##0.0") End Sub </pre>	<p>毎正時のデータが保存されてい れば保存フラグをクリア</p> <p>統計処理用配列にデータが溜まっ たら tmpOneLoop = False</p> <p>1 秒毎のデータ保存カウンタを進め る</p>	
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------	--

## 別紙

## (3) - 6. 各種設定

ページ	ソースコード	説明	機能/参考資料
9	<pre>Private Sub Form_Load0  tmpKeihoSouonnDb1 = KeihoSouonnDb1 tmpKeihoSouonnDb2 = KeihoSouonnDb2 tmpKeihoSindouDb1 = KeihoSindouDb1 tmpKeihoSindouDb2 = KeihoSindouDb2 tmpRangeSouonnDb = RangeSouonnDb tmpRangeSindouDb = RangeSindouDb  tmphyouka1 = Hyouka1 txt 警報騒音 L05.Text = KeihoSouonnDb1 txt 警報騒音 L10.Text = KeihoSouonnDb2 txt 警報振動 L05.Text = KeihoSindouDb1 txt 警報振動 L10.Text = KeihoSindouDb2  txt 騒音計レンジ.Text = RangeSouonnDb txt 振動計レンジ.Text = RangeSindouDb  txt 評価時間.Text = Hyouka1  End Sub Private Sub Command1_Click0 `OK  KeihoSouonnDb1 = tmpKeihoSouonnDb1 KeihoSouonnDb2 = tmpKeihoSouonnDb2 KeihoSindouDb1 = tmpKeihoSindouDb1 KeihoSindouDb2 = tmpKeihoSindouDb2 RangeSouonnDb = tmpRangeSouonnDb RangeSindouDb = tmpRangeSindouDb  Hyouka1 = tmphyouka1  Unload Form2 End Sub</pre>	<p>各種設定画面を開いて警報管理値 他パラメータを表示する</p> <p>「OK」ボタン</p> <p>警報管理値他パラメータを保存し て各種設定画面を閉じる</p>	(ii)警報出力の設定

ページ	ソースコード	説明	機能/参考資料
9	<pre>Private Sub Command2_Click() 'キャンセル Unload Form2 End Sub  Private Sub txt 警報騒音 L05_LostFocus() If 30 &gt; Val(txt 警報騒音 L05.Text) Or Val(txt 警報騒音 L05.Text) &gt; 120 Then MsgBox "警報管理値は 30~120 の範囲で入力してください。" txt 警報騒音 L05.Text = tmpKeihoSouonnDb1  Exit Sub End If tmpKeihoSouonnDb1 = Val(txt 警報騒音 L05.Text)  End Sub Private Sub txt 警報騒音 L10_LostFocus() If 30 &gt; Val(txt 警報騒音 L10.Text) Or Val(txt 警報騒音 L10.Text) &gt; 120 Then MsgBox "警報管理値は 30~120 の範囲で入力してください。" txt 警報騒音 L10.Text = tmpKeihoSouonnDb2  Exit Sub End If tmpKeihoSouonnDb2 = Val(txt 警報騒音 L10.Text)  End Sub</pre>	<p>「キャンセルボタン」</p> <p>何もせず各種設定画面を閉じる</p> <p>警報管理値（騒音レベル1）編集処理</p> <p>警報管理値（騒音レベル2）編集処理</p>	
10	<pre>Private Sub txt 警報振動 L05_LostFocus() If 30 &gt; Val(txt 警報振動 L05.Text) Or Val(txt 警報振動 L05.Text) &gt; 100 Then MsgBox "警報管理値は 30~100 の範囲で入力してください。" txt 警報振動 L05.Text = tmpKeihoSindouDb1  Exit Sub End If tmpKeihoSindouDb1 = Val(txt 警報振動 L05.Text)  End Sub</pre>	<p>警報管理値（振動レベル1）編集処理</p>	

ページ	ソースコード	説明	機能/参考資料
10	<pre> Private Sub txt 警報振動 L10_LostFocus() If 30 &gt; Val(txt 警報振動 L10.Text) Or Val(txt 警報振動 L10.Text) &gt; 100 Then MsgBox "警報管理値は 30~100 の範囲で入力してください。" txt 警報振動 L10.Text = tmpKeihoSindouDb2  Exit Sub End If tmpKeihoSindouDb2 = Val(txt 警報振動 L10.Text)  End Sub Private Sub txt 評価時間_LostFocus() If 100 &gt; Val(txt 評価時間.Text) Or Val(txt 評価時間.Text) &gt; 3600 Then MsgBox "警報管理値は 100~3600(100 ピッチ)の範囲で入力してください。" txt 評価時間.Text = tmpyoukai1 Exit Sub End If Dim sAAA As Single sAAA = 100 * Int(Val(txt 評価時間.Text) / 100) txt 評価時間.Text = sAAA  tmpyoukai1 = Val(txt 評価時間.Text) Private Sub txt 騒音計レンジ_LostFocus() If 80 &gt; Val(txt 騒音計レンジ.Text) Or Val(txt 騒音計レンジ.Text) &gt; 140 Then MsgBox "騒音計レンジ値は 80~140 の範囲で入力してください。" txt 騒音計レンジ.Text = tmpRangeSouonnDb Exit Sub End If tmpRangeSouonnDb = Val(txt 騒音計レンジ.Text)  End Sub Private Sub txt 振動計レンジ_LostFocus() If 70 &gt; Val(txt 振動計レンジ.Text) Or Val(txt 振動計レンジ.Text) &gt; 120 Then MsgBox "振動計レンジ値は 70~120 の範囲で入力してください。" txt 振動計レンジ.Text = tmpRangeSindouDb Exit Sub End If tmpRangeSindouDb = Val(txt 振動計レンジ.Text)  End Sub </pre>	<p>警報管理値（振動レベル2）編集処理</p> <p>L05、L10 評価時間編集処理</p> <p>騒音計測定レンジ編集処理</p> <p>振動計測定レンジ編集処理</p>	

## 別紙

## 求釈明事項(4)

上記(1)～(3)を踏まえ、騒音計 NL-21、振動計 VM-52 から出力される具体的な一連の相異なる電圧値に対して(例えば、NL-21 の測定レンジが 20～80dB である場合において、20dB から 80dB の一連の相異なる騒音値が 1.0V から 2.5V の一連の相異なる電圧値として出力される場合等)、各電圧値が、AD 変換ユニットで、どのようなデジタル値に変換され、(i)～(vi)の機能において、どのように処理されていくのか、ソースコード内の各過程を辿って、具体的に説明してください。

「テストデータを用いて処理の過程を追って説明せよ」とあるが、一連の動作を確認するためには 36000 セット(騒音レベル、振動レベル)以上のテストデータを用意する必要があり、それらのデータについて(3)～5で説明した処理がどの様に行われるかを、書面で説明するのは煩雑になるため説明を省略し(i)～(vi)の機能について処理内容を説明する。

## (4) - 1. 騒音計、振動計の出力電圧がどのようなデジタル値になるか

本書(1)に測定値と出力電圧の関係を、本書(2)に出力電圧とデジタル値の関係を示す。

また、AD 変換に関するプログラムは以下のようになる。

## (1). Cmd\_AioOpen\_Click()関数を呼び出し初期化する

ドライバが初期化され AD 変換器が使用できるようになる。

Cmd\_AioOpen\_Click()関数では次の API 関数を呼び出している。

(乙 3 8 2 ページ、AioOpenEx 関数参照)

```
Ret = AioOpenEx(TxtDevice, hDrv) '初期化を行いデバイスハンドルを取得します
```

## (2). Cmd\_AioInpBack\_Click()関数を呼び出し AD 変換を行う

AD 変換に必要なパラメータを設定して API 関数を呼び出すことにより AD が行われ指定した配列にデジタルデータが格納される。

## ● AD 変換パラメータ (乙 3 8 31 ページ、AINPB 構造体参照)

```
' ソフトウェアスタート
InpMode = 0 ' コマンドにより AD 変換を開始する
' 変換を行う順番を決定します
For i = 0 To 1
    InpChNo(i) = i ' 入力 CH は 0CH と 1CH を使用する
Next i
' 配列の先頭アドレスを代入します
```

## 別紙

LpAinpB.InpBuf = LpWord(InpBuf(TimePoint))	デジタルデータの格納先を設定する
LpAinpB.InpChNo = LpWord(InpChNo(0))	入力CHを設定する
'2チャンネル変換します	
LpAinpB.Channels = 2	AD変換のCH数は2
'10回変換します	
LpAinpB.Scans = 10	10回(秒10データAD変換を行う)
'システムタイマを使用して変換します	
LpAinpB.TimType = 0	
'100msecで変換します(10 × 10ms)	
LpAinpB.Tim = 9.3	サンプリングピッチは90ms (9.3を指定しているがTimメンバーは整数型のため9となる)

## ● AD変換の実行(乙38 10ページ、AioInpBack関数参照)

```
MsgOK.Message = MEG_OK      '変換完了イベントメッセージ番号
MsgERR.Message = MEG_EXCP   '例外発生イベントメッセージ番号
Ret = AioInpBack(hDrv, InpMode, LpAinpB, MsgOK.hWnd, MsgOK.Message, MsgERR.hWnd, MsgERR.Message)
```

0CHと1CHのアナログ値を90ms毎に10回AD変換を行い、変換結果を配列に保存する。

## (4) - 2. (i)外部表示装置への電光掲示

乙23のプログラムではサポートされていない。乙23のソースコードは平成16年2月6日時点のものであり、乙3(騒音振動監視システム 製作仕様書 平成15年10月)にしたがって作成されているが、開発途中バージョンであるため一部の機能は実装されていない。乙3の9頁のイメージ写真及び外観図のとおり、外部表示装置への電光掲示は予定されていない。乙4(サイレントロボカタログ)は現行機のカタログであるため、乙23のソースコードには無い機能が記載されている。

## (4) - 3. (ii)警報出力の設定

本書(3) - 6で示すプログラムで管理値の設定・変更を行う。管理値はプログラム終了時に次のプログラムによりレジストリに保存される。

```
SaveSetting "騒音警報プログラム", "設定", "警報騒音1", KeihoSouonnDb1
SaveSetting "騒音警報プログラム", "設定", "警報騒音2", KeihoSouonnDb2
SaveSetting "騒音警報プログラム", "設定", "警報振動1", KeihoSindouDb1
SaveSetting "騒音警報プログラム", "設定", "警報振動2", KeihoSindouDb2
SaveSetting "騒音警報プログラム", "設定", "騒音レンジ", RangeSouonnDb
SaveSetting "騒音警報プログラム", "設定", "振動レンジ", RangeSindouDb
SaveSetting "騒音警報プログラム", "設定", "評価時間", Hyouka1
```

また、システム起動時に次のプログラムにより管理値は読み込まれる。

```
KeihoSouonnDb1 = GetSetting("騒音警報プログラム", "設定", "警報騒音1", 75#)
KeihoSouonnDb2 = GetSetting("騒音警報プログラム", "設定", "警報騒音2", 85#)
KeihoSindouDb1 = GetSetting("騒音警報プログラム", "設定", "警報振動1", 60#)
KeihoSindouDb2 = GetSetting("騒音警報プログラム", "設定", "警報振動2", 75#)
RangeSouonnDb = GetSetting("騒音警報プログラム", "設定", "騒音レンジ", 110#)
RangeSindouDb = GetSetting("騒音警報プログラム", "設定", "振動レンジ", 80#)
```

## 別紙

## (4) - 4. (iii)黄色、赤色回転灯（警報）の動作

回転灯（警報）の動作は以下に示す段階により決定される。

## (1).騒音レベル(L05)、振動レベル(L10)の取得（本書（4） - 5. 参照）

統計処理により騒音レベル(L05)、振動レベル(L10)を取得する。

```

tmpL05Count = Int(Hyouka1 * 0.05)
lbl 騒音計 L05 = Format(tmpDataS(tmpL05Count), "#00.0")

tmpL10Count = Int(Hyouka1 * 0.1)
lbl 振動計 L10 = Format(tmpDataV(tmpL10Count), "#00.0")

```

## (2).騒音レベル(L05)、振動レベル(L10)と管理値を比較し、フラグ変数をセットする。

```

If Val(lbl 騒音計 L05) >= KeihoSouonnDb2 Then
    KeihoFlg = KeihoFlg Or &H1
Else
    KeihoFlg = KeihoFlg And &HE
End If
If Val(lbl 振動計 L10) >= KeihoSindouDb2 Then
    KeihoFlg = KeihoFlg Or &H2
Else
    KeihoFlg = KeihoFlg And &HD
End If
If Val(lbl 騒音計 L05) >= KeihoSouonnDb1 Then
    KeihoFlg = KeihoFlg Or &H4
Else
    KeihoFlg = KeihoFlg And &HB
End If
If Val(lbl 振動計 L10) >= KeihoSindouDb1 Then
    KeihoFlg = KeihoFlg Or &H8
Else
    KeihoFlg = KeihoFlg And &H7
End If

```

## (3).フラグ変数の状態により回転灯を点灯する。フラグ変数の状態については本書 26 ページを、回転灯の制御は本書 14～15 ページ参照

```

If (KeihoFlg > 3) And ((KeihoFlg And &H3) > 0) Then
    Cmd_DioOpen_Click
    PortDataInf.Text = "3"
    Cmd_DioOutByte_Click
    Cmd_DioClose_Click
ElseIf (KeihoFlg > 3) And ((KeihoFlg And &H3) = 0) Then
    Cmd_DioOpen_Click
    PortDataInf.Text = "2"
    Cmd_DioOutByte_Click
    Cmd_DioClose_Click
ElseIf KeihoFlg And &H10 > 0 Then
    Cmd_DioOpen_Click
    PortDataInf.Text = "1"
    Cmd_DioOutByte_Click
    Cmd_DioClose_Click
Else
    Cmd_DioOpen_Click
    PortDataInf.Text = "0"
    Cmd_DioOutByte_Click

```

## 別紙

Ond\_DioClose\_Click

(4) - 4. (iv) 10 Hz でのサンプリング及び 0.1SEC~1SEC でのサンプリングタイムの任意設定

乙 23 のプログラムではサポートされていない。

(4) - 5. (v) 各種統計データ (Lp、Leq、L05 等) の算出

乙 23 のプログラムでは騒音レベル(L05)、振動レベル(L10)をサポートしており以下の手順により算出される。

(1). 秒 10 個の電圧値を取得し、最大値を求める。(乙 23 のプログラムにはバグがあり最小値になっている)

```
If TimePoint = 0 Then TimePoint = 72000 - 20
tmpMaxInpBuf1Sec0 = InpBuf(TimePoint - 20)
For i = 2 To 18 Step 2
    If tmpMaxInpBuf1Sec0 > InpBuf(TimePoint - 20 + i) Then tmpMaxInpBuf1Sec0 = InpBuf(TimePoint - 20 + i)
Next i
tmpMaxInpBuf1Sec1 = InpBuf(TimePoint - 20 + 1)
For i = 3 To 19 Step 2
    If tmpMaxInpBuf1Sec1 > InpBuf(TimePoint - 20 + i) Then tmpMaxInpBuf1Sec1 = InpBuf(TimePoint - 20 + i)
Next i
```

(2). 求めた最大値を騒音レベル(LA)、振動レベル(LA)に換算する。

```
If tmpMaxInpBuf1Sec0 >= 0 Then
    |b| 騒音計 LA = Int(((tmpMaxInpBuf1Sec0 - 32768) / 3276.8 * 20 + (RangeSouonnDb - 70)) * 10) / 10
Else
    |b| 騒音計 LA = Int(((tmpMaxInpBuf1Sec0 + 32768) / 3276.8 * 20 + (RangeSouonnDb - 70)) * 10) / 10
End If

If tmpMaxInpBuf1Sec1 >= 0 Then
    |b| 振動計 LA = Int(((tmpMaxInpBuf1Sec1 - 32768) / 3276.8 * 20 + (RangeSindouDb - 60)) * 10) / 10
Else
    |b| 振動計 LA = Int(((tmpMaxInpBuf1Sec1 + 32768) / 3276.8 * 20 + (RangeSindouDb - 60)) * 10) / 10
End If
```

(3). 毎秒データとして騒音レベル(LA)、振動レベル(LA)を保存する。

```
vData(vDataCount, 0) = Val(|b| 騒音計 LA)
vData(vDataCount, 1) = Val(|b| 振動計 LA)
```

(4). 騒音レベル(L05)を求めるため配列に騒音レベル(LA)を追加し昇順に並び替える。

配列には L05、L10 評価時間で設定した時間だけのデータを保存する。評価時間より古いデータは

## 別紙

新しいデータで上書きされる。

```

If OneTimeStartFlag = True Then '1 回目だけの処理
    tmpDataSerNo = 0
    OneTimeStartFlag = False
    tmpDataS(tmpDataSerNo) = vData(vDataCount, 0)
    tmpDataCountS(tmpDataSerNo) = vDataCount
    tmpOneLoop = True
    tmpDataNow = Right(Time, 1)
    GoTo INCLIMENT
End If
If tmpOneLoop = True Then '1 周回だけの処理
    tmpDataS(tmpDataSerNo) = vData(vDataCount, 0)
    tmpDataCountS(tmpDataSerNo) = vDataCount
    tmpDataSerNo0 = tmpDataSerNo
    For i = tmpDataSerNo - 1 To 0 Step -1
        If tmpDataS(i) < tmpDataS(i + 1) Then
            tmpData0 = tmpDataS(i)
            tmpData1 = tmpDataS(i + 1)
            tmpDataS(i) = tmpData1
            tmpDataS(i + 1) = tmpData0
            tmpDataCount0 = tmpDataCountS(i)
            tmpDataCount1 = tmpDataCountS(i + 1)
            tmpDataCountS(i) = tmpDataCount1
            tmpDataCountS(i + 1) = tmpDataCount0
        End If
    Next i
Else
    tmpDataCount0 = tmpDataCountS(0)
    For i = 0 To Hyouka1 - 1 - 1
        If tmpDataCount0 > tmpDataCountS(i + 1) Then
            tmpDataCount0 = tmpDataCountS(i + 1)
            LastCount = i + 1
        End If
    Next i
    tmpDataS(LastCount) = vData(vDataCount, 0)
    tmpDataCountS(LastCount) = vDataCount
    tmpDataSerNo0 = tmpDataSerNo
    If LastCount = 0 Then
        i = i
    ElseIf tmpDataS(LastCount) = tmpDataS(LastCount - 1) Then
        i = i
    ElseIf tmpDataS(LastCount) > tmpDataS(LastCount - 1) Then
        For i = LastCount - 1 To 0 Step -1
            If tmpDataS(i) < tmpDataS(i + 1) Then
                tmpData0 = tmpDataS(i)
                tmpData1 = tmpDataS(i + 1)
                tmpDataS(i) = tmpData1
                tmpDataS(i + 1) = tmpData0
                tmpDataCount0 = tmpDataCountS(i)
                tmpDataCount1 = tmpDataCountS(i + 1)
                tmpDataCountS(i) = tmpDataCount1
                tmpDataCountS(i + 1) = tmpDataCount0
            End If
        Next i
    ElseIf tmpDataS(LastCount) < tmpDataS(LastCount - 1) Then
        For i = LastCount To Hyouka1 - 1
            If tmpDataS(i) < tmpDataS(i + 1) Then
                tmpData0 = tmpDataS(i)
                tmpData1 = tmpDataS(i + 1)
                tmpDataS(i) = tmpData1
                tmpDataS(i + 1) = tmpData0
                tmpDataCount0 = tmpDataCountS(i)
                tmpDataCount1 = tmpDataCountS(i + 1)
                tmpDataCountS(i) = tmpDataCount1
                tmpDataCountS(i + 1) = tmpDataCount0
            End If
        Next i
    End If

```

別紙

```

        End If
      Next i
    End If
  End If

```

(5).騒音レベル(L05)を求める。

騒音レベル L05 とは、評価時間分の騒音レベル(LA)を昇順に並べた時、最大値より - 5 % の位置にあるデータの事である。(図. 21)

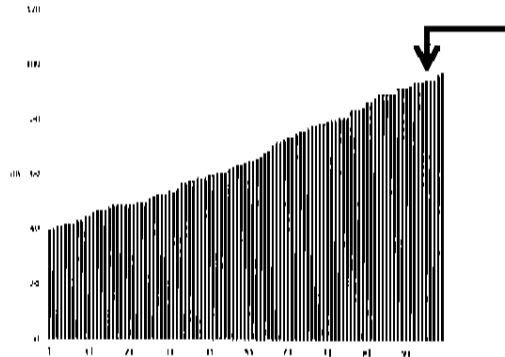


図. 21 騒音レベル(L05)

最大値 - 5 % の位置の騒音レベル(LA)を配列から取り出し、騒音レベル(L05)として表示する。

```

tmpL05Count = Int(Hyouka1 * 0.05)
|b| 騒音計 L05 = Format(tmpDataS(tmpL05Count), "#0.0")

```

(6).振動レベル(L10)を求めるため配列に振動レベル(LA)を追加し昇順に並び替える。

配列には L05、L10 評価時間で設定した時間だけのデータを保存する。評価時間より古いデータは新しいデータで上書きされる。

```

If OneTimeStartFlag = True Then '1 回目だけの処理
  tmpDataSerNo = 0
  OneTimeStartFlag = False
  tmpDataV(tmpDataSerNo) = vData(vDataCount, 1)
  tmpDataCountV(tmpDataSerNo) = vDataCount
  tmpOneLoop = True
  GoTo INCLIMENT
End If
If tmpOneLoop = True Then '1 周回だけの処理
  tmpDataV(tmpDataSerNo) = vData(vDataCount, 1)
  tmpDataCountV(tmpDataSerNo) = vDataCount
  tmpDataSerNo0 = tmpDataSerNo
  For i = tmpDataSerNo - 1 To 0 Step -1
    If tmpDataV(i) < tmpDataV(i + 1) Then
      tmpData0 = tmpDataV(i)
      tmpData1 = tmpDataV(i + 1)
      tmpDataV(i) = tmpData1
      tmpDataV(i + 1) = tmpData0
      tmpDataCount0 = tmpDataCountV(i)
      tmpDataCount1 = tmpDataCountV(i + 1)
      tmpDataCountV(i) = tmpDataCount1
    End If
  Next i
End If

```

## 別紙

```

        tmpDataCountV(i + 1) = tmpDataCount0
    End If
Next i
Else
    tmpDataCount0 = tmpDataCountV(0)
    For i = 0 To Hyouka1 - 1 - 1
        If tmpDataCount0 > tmpDataCountV(i + 1) Then
            tmpDataCount0 = tmpDataCountV(i + 1)
            LastCount = i + 1
        End If
    Next i
    tmpDataV(LastCount) = vData(vDataCount, 1)
    tmpDataCountV(LastCount) = vDataCount
    tmpDataSerNo0 = tmpDataSerNo
    If LastCount = 0 Then
        i = 1
    ElseIf tmpDataV(LastCount) = tmpDataV(LastCount - 1) Then
        i = i
    ElseIf tmpDataV(LastCount) > tmpDataV(LastCount - 1) Then
        For i = LastCount - 1 To 0 Step -1
            If tmpDataV(i) < tmpDataV(i + 1) Then
                tmpData0 = tmpDataV(i)
                tmpData1 = tmpDataV(i + 1)
                tmpDataV(i) = tmpData1
                tmpDataV(i + 1) = tmpData0
                tmpDataCount0 = tmpDataCountV(i)
                tmpDataCount1 = tmpDataCountV(i + 1)
                tmpDataCountV(i) = tmpDataCount1
                tmpDataCountV(i + 1) = tmpDataCount0
            End If
        Next i
    ElseIf tmpDataV(LastCount) < tmpDataV(LastCount - 1) Then
        For i = LastCount To Hyouka1 - 1
            If tmpDataV(i) < tmpDataV(i + 1) Then
                tmpData0 = tmpDataV(i)
                tmpData1 = tmpDataV(i + 1)
                tmpDataV(i) = tmpData1
                tmpDataV(i + 1) = tmpData0
                tmpDataCount0 = tmpDataCountV(i)
                tmpDataCount1 = tmpDataCountV(i + 1)
                tmpDataCountV(i) = tmpDataCount1
                tmpDataCountV(i + 1) = tmpDataCount0
            End If
        Next i
    End If
End If

```

## (7).振動レベル(L10)を求める。

振動レベル L10 とは、評価時間分の振動レベル(LA)を昇順に並べた時、最大値より -10%の位置にあるデータの事である。(図. 22)

## 別紙

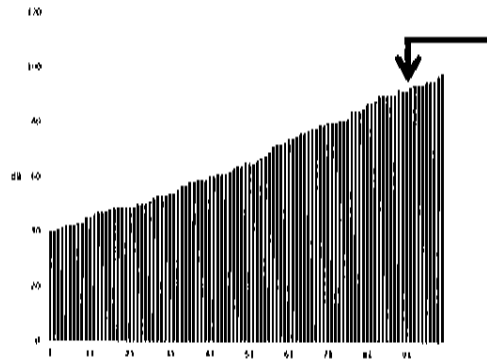


図. 2 2 振動レベル(L10)

最大値 - 10%の位置の振動レベル(LA)を配列から取り出し、振動レベル(L10)として表示する。

```
tmpL10Count = Int(Hyouka1 * 0.1)
|b| 振動計 L10 = Format(tmpDataV(tmpL10Count), "#00.0")
```

## (4) - 6. (vi)測定データ保存

毎正時 59 分 50 秒又は 59 分 51 秒になった時、59 分 50 秒分の騒音レベル(LA)、振動レベル(LA)のデータ (35900 個) を保存する。

データ保存先フォルダは"C:\共有\騒音振動監視\data\"固定であり、ファイル名は保存時の日時から決定する。

```
If (Right(Time, 5) = "59:50" Or Right(Time, 5) = "59:51") And DataWriteFlg = False Then
Dim FileNameCSV As String
Dim FileNo As Integer
Dim wData(36000, 1) As Single
Dim FileNameCSV As String
FileNameCSV = Format(Date, "yyyymmdd")
FileNameCSV = "C:\共有\騒音振動監視\data\" & FileNameCSV & Format(Time, "hh") & ".CSV"
FileNo = FreeFile()
Open FileNameCSV For Output As #FileNo
For ii = 0 To 35899
If InpBuf(ii * 2) >= 0 Then
wData(ii, 0) = Int(((InpBuf(ii * 2) - 32768) / 3276.8 * 20 + (RangeSouonnDb - 70)) * 10) / 10
Else
wData(ii, 0) = Int(((InpBuf(ii * 2) + 32768) / 3276.8 * 20 + (RangeSouonnDb - 70)) * 10) / 10
End If

If InpBuf(ii * 2 + 1) >= 0 Then
wData(ii, 1) = Int(((InpBuf(ii * 2 + 1) - 32768) / 3276.8 * 20 + (RangeSindouDb - 60)) * 10) / 10
Else
wData(ii, 1) = Int(((InpBuf(ii * 2 + 1) + 32768) / 3276.8 * 20 + (RangeSindouDb - 60)) * 10) / 10
End If

Next ii
For ii = 0 To 35899
DoEvents
Write #FileNo, wData(ii, 0), wData(ii, 1)
Next ii
Close #FileNo
```