

CA-210/100plus 技術資料



KONICA MINOLTA

初版	2002年12月18日
第2版	2003年04月24日
第3版	2003年09月15日
第4版	2003年10月01日
第5版	2007年07月01日
第6版(社名変更)	2013年09月18日

コニカミノルタ株式会社

〒590-8551 大阪府堺市堺区大仙西町3-91

<http://sensing.konicaminolta.jp/>

本書に関するご注意

・本書に記載の見出しに「☞ さらに詳しく」のマークが表示されている内容は、非常に詳細な説明をしている部分です。内容によっては、省略して読み進めてください。

・本書の内容の一部または全部を無断で転載することは、禁止されています。

・本書の内容に関しては、将来予告なしに変更することがあります。

・本書は内容について万全を期していますが、万一不審な点や誤り、記載もれなどでお気づきの点がございましたら、ご購入の販売店または“サービスのご案内”に記載のお問い合わせ窓口までご連絡ください。

・本書の内容を運用した結果につきましては、上記にかかわらず責任を負いかねますので、あらかじめご了承ください。

・Windows[®]は、米国マイクロソフト社の商標です。その他、本書に記載の会社名、商品名は各社の商標または登録商標です。

目次

1. 概要	2
1-1 色度と測色について	3
1-2 色彩計について	5
1-3 アナライザモード	10
1-4 マトリクス校正	13
1-5 光学系の特長	17
1-6 回路系の特長	19
1-7 フリッカの測定	21
1-8 PDP (プラズマディスプレイ) の測定	31
2. 確度定格・繰返し性の定義	34
2-1 「輝度・色度」確度定格・繰返し性の定義	35
2-2 「フリッカ」確度定格・繰返し性の定義	39
3. 実測データ (精度定格外の条件での実力)	42
3-1 CA-210 CA-110 CS-1000 の液晶測定値差	43
3-2 CA-210 CA-110 CS-1000 γ 特性比較 (開口角の違い)	48
3-3 CA-210 (測定プローブ 小測定径プローブ), CS-1000 の液晶測定値差	51
3-4 CA-210 のフリッカ測定精度	53
3-5 CA-210 の測定速度	56
3-6 CA-100, CA-100Plus の CRT 測定値差	57
3-7 CA-100Plus, CA-100 の CRT、PDP 測定値差	59
4. CA-SDK ソフト解説	61
4-1 COM とは	62
4-2 CA-SDK による C++アプリケーション作成方法	67
4-3 CA-SDK サンプルソフト用コントロールの利用方法	87
4-4 CA-SDK VB サンプルソフト	100
4-5 CA-SDK による Visual Basic.net アプリケーションの作成方法	107
5. CA-210 アプリケーション例	117
5-1 ホワイトバランス調整	118
5-2 γ 調整	121
6. 関連規格	126
6-1 ED-2522 規格と CA-210 の関係	127
6-2 VESA 規格と CA-210 の関係	131
6-3 sRGB 色空間とは	139

*本書[概要]の部分は、2002年12月号の「月間 Display」に掲載されています。

CA-210/100plus 技術資料

1. 概要

1-1 色度と測色について

1-1-1 色度とは

色を定量的に表すために現在までに様々な表色系が考えられてきました。最も簡単な表色系としてRGB表色系があります。このRGB表色系は、全ての色をRGBの混色比で表わそうとしたもので、国際照明委員会(CIE)はRとして700.0nm,Gとして546.1nm,Bとして435.8nmの単色光を選定しました(この単色光を原刺激と呼びます)。ところが、この表色系では、一部の色を表現する場合に混色する値がマイナスとなることがあり、問題視されていました。この問題解決するため、全ての混色比がプラスとなるように、別の原刺激を選びました。これが、1931年に国際照明委員会(CIE)によって規定されたXYZ表色系で、現在最も広く用いられている表色系です。この表色系の色度図を図1-1-1に示します。

他方、この表色系の問題点として、この表色上での色差(2つの色の差、表色系上での2つの色の距離のこと)と実際に人に認識される色差に相違が生じるということがあります。例えば、LCDの緑単色付近のある色と青単色付近のある色をx、yそれぞれ同じ量ずつ変化させた場合、人は、青色付近の色の方が大きく変化したように認識します。この問題を可決した表色系がu'v'表色系です。このu'v'表色系においては、全ての色範囲において同じ大きさの色差は、人にほぼ等しく認識されるようになっています。この利点から、最近では、この表色系を使用する頻度が高くなる傾向が見られます。この表色系の色度図を図1-1-2に示します。

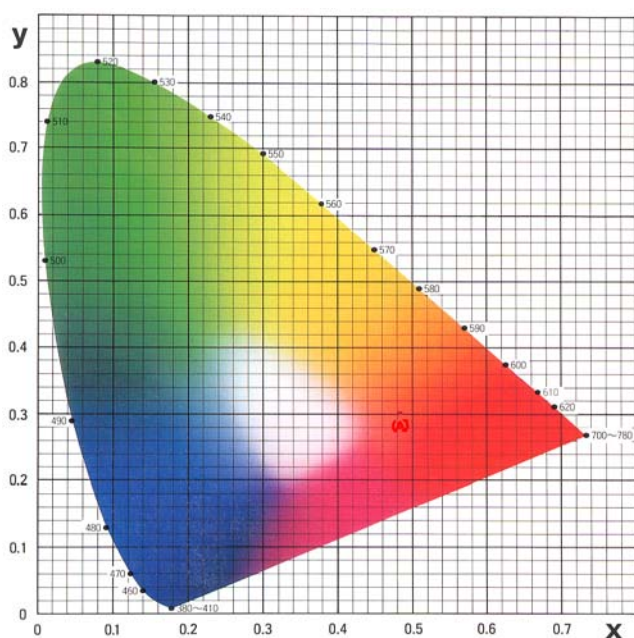


図 1-1-1 XYZ表色系

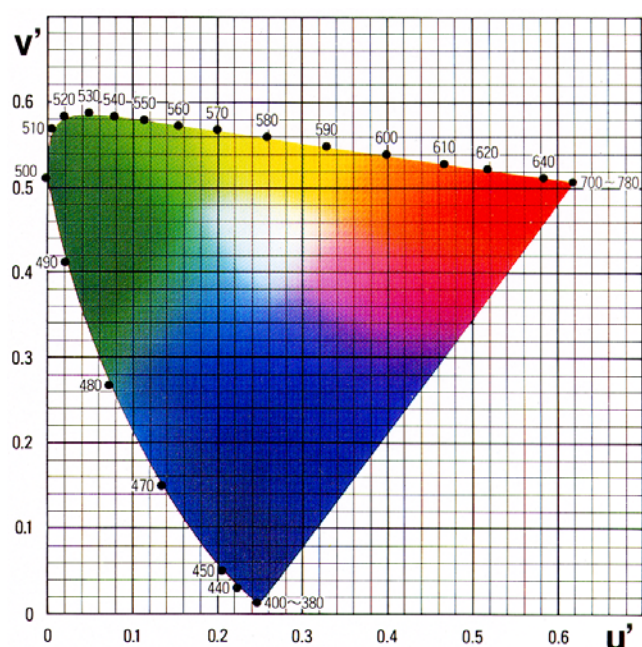


図 1-1-2 u'v'表色系

1-1-2 測色について

XYZ表色系で光源を測色するためには、まず、図 1-1-3 に示すようなx, y, zの等色関数と同じ分光感度をもったセンサの出力(X, Y, Z)を得ます。

この出力を下記式に従って演算する事により、色度を求めます。

$$x = \frac{X}{X + Y + Z}$$

$$y = \frac{Y}{X + Y + Z}$$

u'v'表色系では、X, Y, Zを用いて下記式に従って演算する事により、色度を求めます。

$$u' = \frac{4X}{X + 12Y + 3Z}$$

$$v' = \frac{9Y}{X + 12Y + 3Z}$$

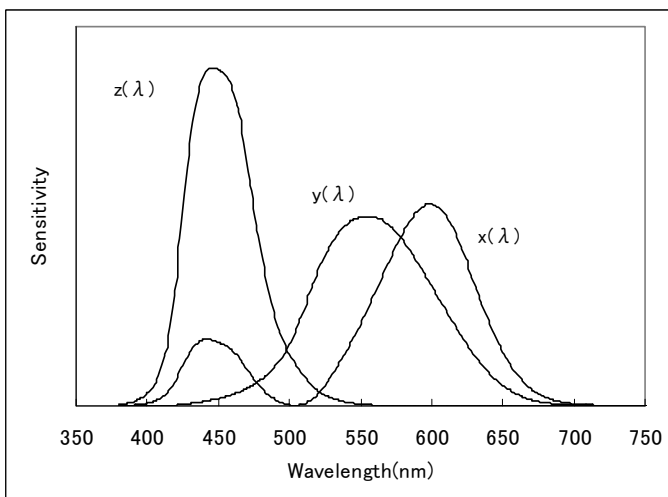


図 1-1-3 等色関数

1-2 色彩計について

色彩計は測色方法の違いによって、大きく刺激値直読式色彩計と分光色彩計に分けることができます。ここでは、それぞれの色彩計の原理、特長の説明を行います。

1-2-1 刺激値直読式色彩計

刺激値直読式色彩計では、CIE1931 で規定された等色関数に近似された分光応答度をもった3つのセンサーの出力を用いて色度、輝度を求めます。図 1-2-1 に、センサーの分光応答度の1例を示しています。これらのセンサーは一般に光学フィルタ、受光素子から構成されています。

被測定光源の分光放射輝度を $S(\lambda)$ 、センサーの各分光応答度を $x'(\lambda)$ 、 $y'(\lambda)$ 、 $z'(\lambda)$ とすると、各センサーの出力 X 、 Y 、 Z は、

$$X = \int S(\lambda) \cdot x'(\lambda) \cdot d\lambda$$

$$Y = \int S(\lambda) \cdot y'(\lambda) \cdot d\lambda$$

$$Z = \int S(\lambda) \cdot z'(\lambda) \cdot d\lambda$$

(ここで、 λ は波長。波長範囲は可視光の波長領域。)

となります。ここで、得られた出力を用いて、下記の演算を行うことにより、色度、輝度を算出します。

$$x = \frac{X}{X + Y + Z}$$

$$y = \frac{Y}{X + Y + Z}$$

$$L_v = Y$$

1-2-2 分光色彩計

分光色彩計では、まず被測定光源の各波長ごとのエネルギー量を求めます。次に、この量にCIE1931 で規定された各等色関数を各波長ごとに乗じる演算処理を行うことにより色度、輝度を求めます。

この測定器は一般に以下のような構成をとります。被測定光源からの光を対物レンズを通して集光し、この光を分光します。この分光された光を、ラインセンサ上に投射し、ラインセンサの各素子の出力により、各波長ごとのエネルギー量を求めます。

ここで、各センサーの出力である分光放射輝度を $S'(\lambda)$ 、等色関数を $x(\lambda)$ 、 $y(\lambda)$ 、 $z(\lambda)$ としますと、

$$X = \sum S'(\lambda) \cdot x(\lambda) \cdot \Delta \lambda$$

$$Y = \sum S'(\lambda) \cdot y(\lambda) \cdot \Delta \lambda$$

$$Z = \sum S'(\lambda) \cdot z(\lambda) \cdot \Delta \lambda$$

(ここで、 λ は波長。波長間隔は、センサーの波長分解能。波長範囲は可視光の波長領域。)

以下、刺激値直読式色彩計と同様の計算式を用いて、色度、輝度を算出します。

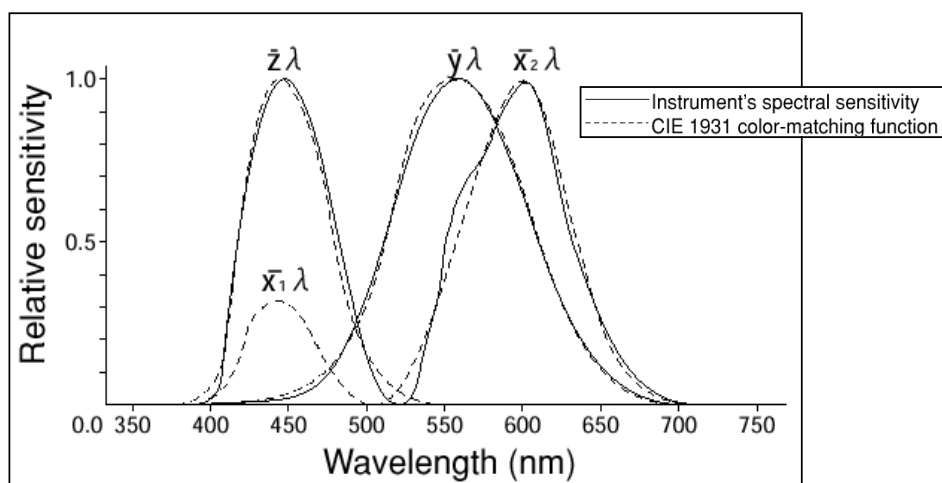


図 1-2-1 等色関数とセンサの分光応答度

1-2-3 刺激値直読式色彩計の絶対値誤差

ここでは、刺激値直読式色彩計に絶対値誤差が発生する理由について説明します。

一般に、刺激値直読式色彩計の各々の分光応答度は、光学フィルタの分光透過率と受光素子の分光応答度の合成で決まります。

この分光応答度が、CIE1931 で規定された等色関数と完全に同じであれば、刺激値直読式色彩計で求めた色度は、任意の光源を測定しても、絶対値誤差は生じません。しかしながら、現在のフィルタ技術では、図 1-2-1 のように、刺激値直読式色彩計の分光応答度を、等色関数に完全に一致させることはできません。この等色関数との差が、絶対値誤差の原因となります。

このメカニズムを、以下に説明します。

ここでは、簡単のため、1つのセンサの出力について考えます。図 1-2-2 に示すように、CIE1931 の等色関数 $y(\lambda)$ (図の破線) に対して、測定器のセンサの分光応答度 $y'(\lambda)$ (図の実線) が、長波長の方向にシフトしている場合を考えます。ここで、短波長域より長波長域で分光放射輝度が大きい光源 $S(\lambda)$ を、異なる分光応答度 $y(\lambda)$, $y'(\lambda)$ をもつ、それぞれのセンサで測定した場合の出力を考えます。これは、

$$Y = \int S(\lambda) \cdot y(\lambda) \cdot d\lambda$$

$$Y' = \int S(\lambda) \cdot y'(\lambda) \cdot d\lambda$$

となり、図 1-2-2 から明らかなように

$$Y < Y'$$

となります。

このケースでは、測定器のフィルタが CIE1931 の特性と異なることにより、センサの出力は真値 (CIE1931 の等色関数から予測される値) より大きくなります。

このような差が生じた場合、この差を無くすために、測定器の分光応答度を CIE1931 の等色関数に近づけることが理想ですが、3刺激値センサの場合、分光応答度は光学フィルタ、素子の分光特性で決まり、これを変化させることは困難です。

このため、センサの出力に適切な係数を乗じ、結果が真値と同じとなるようにする方法が採用されています。これを「校正」といいます。

具体的には、

$$Y / Y' = k_y$$

を求め、以降、 $k_y \cdot Y'$ を用いて色度を算出するようにします。この時の、 Y としては、分光色彩計など、分光応答度が CIE1931 の等色関数に一致している (または非常に近い) 測定器の測定値を用います。

この結果得られる測定器の分光応答度を図 1-2-3 に示します。

ここでセンサーの分光応答度 $y''(\lambda)$ (図の実線) は

$$y''(\lambda) = k_y \cdot y'(\lambda)$$

となり、分光応答度が全体に一定比率分小さくなることで、得られるセンサー出力 Y'' は Y と等しくなります。これを数式で表すと以下のようになります。

$$\begin{aligned} Y'' &= \int S(\lambda) \cdot y''(\lambda) \cdot d\lambda \\ &= \int S(\lambda) \cdot k_y \cdot y'(\lambda) \cdot d\lambda \\ &= k_y \cdot \int S(\lambda) \cdot y'(\lambda) \cdot d\lambda \\ &= (Y / Y') \cdot Y' \\ &= Y \end{aligned}$$

X, Z についても同様に k_x , k_z を求め、以降、この係数を用いて色度を算出するようにすることで、測定値として真値を得ることができます。

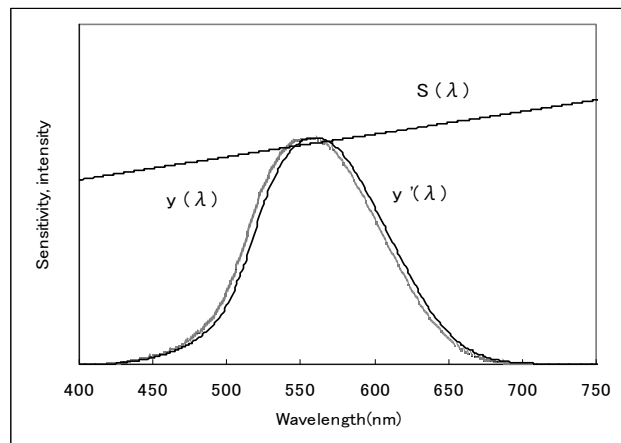


図 1-2-2 光源 $S(\lambda)$ と分光応答度 $y(\lambda)$, $y'(\lambda)$

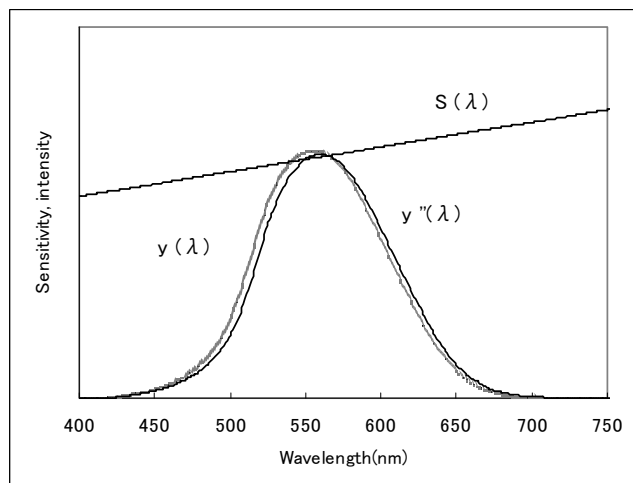


図 1-2-3 光源 $S(\lambda)$ と分光応答度 $y(\lambda)$, $y''(\lambda)$

「校正」を行えば、ある光源を測定する場合には、分光色彩計と同じ絶対値精度を得ることが可能であることを説明しました。ここでは、「校正」を行った光源と異なる光源を測定する場合のことを考えます。

例えば、図 1-2-2 で示した光源に対して「校正」を行った後、長波長側より短波長側で強度が大きい光源 $S'(\lambda)$ を測定する場合を考えてみます(図 1-2-4)。

この場合、両センサの校正前の出力は

$$Y = \int S'(\lambda) \cdot y(\lambda) \cdot d\lambda$$

$$Y'' = \int S'(\lambda) \cdot y''(\lambda) \cdot d\lambda$$

となります。図 1-2-4 より、光源 $S'(\lambda)$ は長波長側より短波長側で強度が大きく、また $y''(\lambda)$ の応答度のピーク値は $y(\lambda)$ のピーク値より小さいのですから、

$$Y \neq Y''$$

となります。

つまり、ある光源に対し「校正」を行った場合、その校正値を用いて、その光源を測定した場合には真値を示しますが、他の光源を測定した場合には真値を示さないことがわかります。

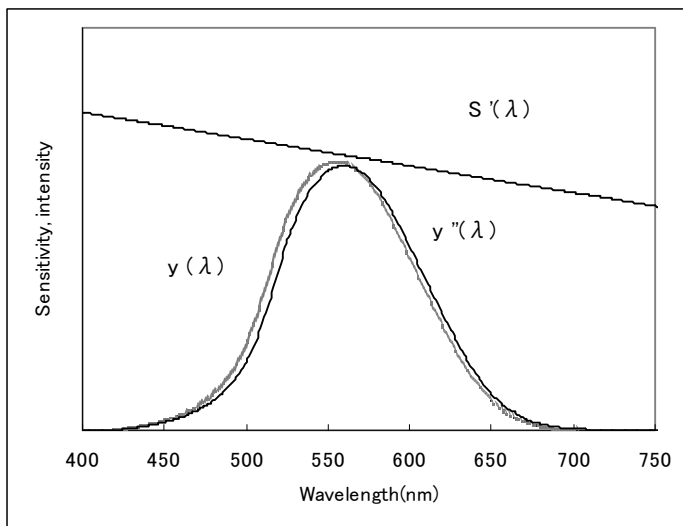


図 1-2-4 光源 $S'(\lambda)$ と分光応答度 $y(\lambda)$, $y''(\lambda)$

ところで、「校正」は、測定器の出荷時に工場内で実施されています。また、仕様として、「校正」に用いた光源が示されている場合が一般的です。

ここで、例えば、校正光源が A 光源である刺激値直読式色彩計があったとします。すると、この色彩計では、A 光源を測定した場合には真値を示しますが、それ以外の光源を測定した場合は真値を示さないことになります。

その時の真値との差が、刺激値直読式色彩計における絶対値誤差となります。

この絶対値誤差は、校正光源と被測定光源の分光放射輝度の差に依存することが容易に推測できます。

言い換えれば、校正光源と被測定光源の分光放射輝度の差が小さい場合、この絶対値誤差は小さいものとなります(※注 1-2-1)。

1-2-4 刺激値直読式色彩計の任意校正

「校正」が、測定器の出荷時に工場内で実施されてることを述べました。

ところで、「校正」については、その校正(工場出荷校正)以外に、測定器を使用される方が光源を選んで実施する校正(任意校正)もあります。

「校正」は、その「校正」に用いた光源を測定するときのみ有効であることを述べました。このことは、任意校正についても言えます。ここでは、この任意校正についての留意点について述べます。

図 1-2-5 に示すように、CIE1931 の等色関数 $y(\lambda)$ (図の破線) に対して、測定器のセンサの分光応答度 $y'(\lambda)$ (図の実線) が、長波長の方向にシフトしている場合を考えます。

また、工場出荷校正に用いた光源の分光放射輝度が、短波長域から長波長域まで等しい $S''(\lambda)$ であったとします。

この場合、センサの分光応答度 $y'(\lambda)$ のピークは図 1-2-5 に示すように、等色関数 $y(\lambda)$ のピークに一致します。

つまり、工場出荷状態でのセンサの分光応答度は図 1-2-5 の $y'(\lambda)$ になっていることになります。

ここで、図 1-2-2 に示した、短波長域より長波長域で分光放射輝度が大きい光源 $S(\lambda)$ を測定した場合の出力を考え

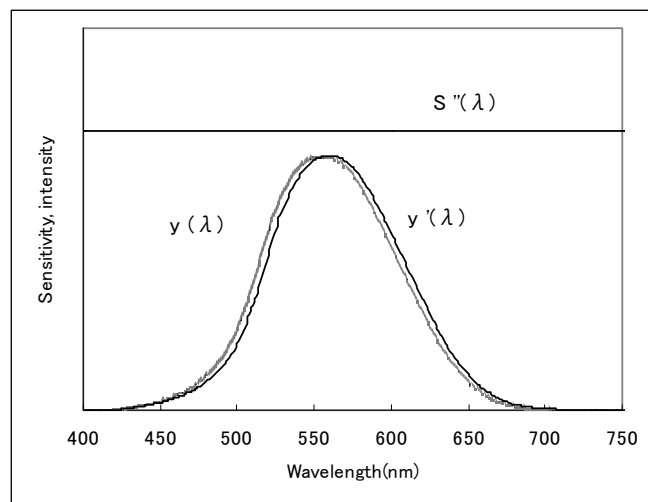


図 1-2-5 光源 $S''(\lambda)$ と分光応答度 $y(\lambda)$, $y'(\lambda)$

ますと、図 1-2-2 から明らかなように、絶対値誤差が発生します。

この誤差をなくす操作が任意校正です。

任意校正も、センサの出力に適当な係数を乗じ、結果が真値と同じとなるようにする点で、工場出荷校正と同じです。従い、図 1-2-2 の光源 $S(\lambda)$ に対して任意校正を行った場合、センサの分光応答度は図 1-2-3 に示した $y'(\lambda)$ となります。

さて、この分光応答度を用いて図 1-2-4 に示した、長波長域より短波長域で分光放射輝度が大きい光源 $S'(\lambda)$ を測定した場合、測定誤差が発生することは、既に説明しました。

ところで、光源 $S'(\lambda)$ を測定した場合、センサの分光応答度のピークが等色関数 $y(\lambda)$ と一致していた場合ですら、センサ出力は真値より小さくなるのですから、光源 $S(\lambda)$ に対して任意校正を行った後では、さらにセンサ出力は小さくなり、真値との差は大きくなります。

- 光源 $S'(\lambda)$ を工場出荷状態で測定した場合に発生する誤差と比較して、光源 $S(\lambda)$ に対して任意校正を行った後、光源 $S'(\lambda)$ を測定した場合、絶対値誤差がむしろ大きくなっていることが判ります。
- 光源 $S'(\lambda)$ を測定する場合にも、光源 $S'(\lambda)$ に対して任意校正を行えば真値が得られます。つまり、刺激値直読式色彩計においても、測定光源ごとに任意校正を行えば、真値を得ることができます。

一般的に、刺激値直読式色彩計では多数の任意校正値をメモリする機能を有し、このメモリ機能を用いて、様々な光源に対して真値を得ることができるようしています(※注 1-2-2)。このメモリ機能を有効活用することで、刺激値直読式色彩計は、精度の高い測定器になります。

1-2-5 刺激値直読式色彩計の器差

刺激値直読式色彩計では、器差が問題にされることがあります。

現在の技術では、機器間の光学フィルタの分光透過率バラツキをなくす事は困難であり、このバラツキが、測定器の器差として現れます。以下に、このことを説明します。

ここに、2台の刺激値直読式色彩計があり、それぞれの分光応答度が図 1-2-3 のようであったとします(光源 $S(\lambda)$ で工場出荷校正された状態、光源 $S(\lambda)$ に対するセンサー出力は2台とも等しい)。

この測定器で、図 1-2-4 の光源 $S'(\lambda)$ を測定した場合、2台のセンサー出力が異なることとなります。

この差が器差となって現れます。

この器差を小さくする方法として、任意校正機能を使用する方法があります。

具体的には、2台の測定器を使用する場合、測定する光源Aに対して、まず1台で測定し、測定値を覚えておきます。

次に、もう1台で光源Aを測定し、任意校正を行い、先の1台と同じ測定値になるようにします。

以上の操作をすれば、以降、光源Aを測定する場合に、器差のない測定をすることができます。

1-2-6 刺激値直読式色彩計と分光色彩計の比較 さらに詳しく

ここでは、分光色彩計と刺激値直読式色彩計の光学系の特長を示した後、両色彩計の長所、短所を項目を示し、その理由について説明します。

まず、両色彩計の光学系の構成概略図を図 1-2-6a(分光色彩計)、図 1-2-6b(刺激値直読式色彩計)に示します。

刺激値直読式色彩計は、被測定光源からの光を対物レンズによりセンサ部に集光するという構成をとります。また、一度集光した光を光ファイバーにより各センサにまで導光することにより、光利用効率をあげる方法がとられることもあります。

分光色彩計は、まず、被測定光源からの光を対物レンズを通して集光した後、照明レンズにより回折格子に照射します。入射光は回折格子で分光され、集光レンズに出射されます。次に、この分光された光を、ラインセンサ上に投射し、ラインセンサの各素子の出力により、各波長ごとのエネルギー量を得ます。

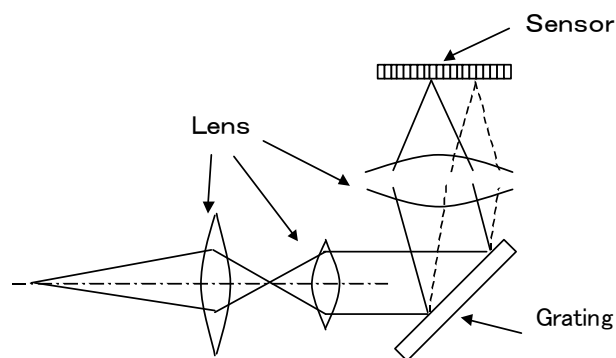


図 1-2-6a 分光色彩計の構成概略図

以上から、光学系の構成を比較すると、分光色彩計が刺激値直読式色彩計に比べて複雑であることが分かります。

次に、両色彩計の長所、短所を表1に示します。

光源を測定する場合には、各特長を把握した上で、刺激値直読式色彩計、分光色彩計を使い分ける必要があります。(※注 1-2-4)

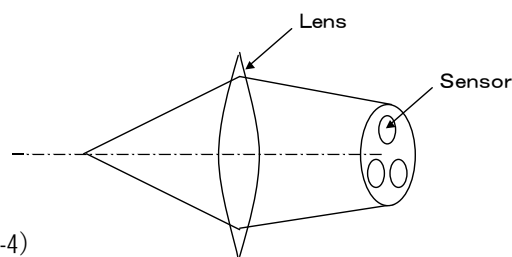


図 1-2-6b 刺激値直読式色彩計の構成概略図

表 1 刺激値直読式色彩計と分光色彩計の比較

	刺激値直読式色彩計	分光色彩計	理由 (刺激値直読式色彩計を刺激値型分光色彩計を分光型として説明)
絶対値精度	低い	高い	分光型の分光応答度が等色関数に一致しているのに対し、刺激値型の分光応答度は等色関数に対し差があるため。(※注 1-2-3)
測定速度	速い	遅い	刺激値型はセンサ数が3個と少ないのに対し、分光型はセンサ数が数100個となり、処理データ量が多くなる
感度	高い	低い	刺激値型はセンサ数が3個と少ないのに対し、分光型はセンサ数が数100個となる。この結果、1センサあたりの受光光量が、刺激値型に対し分光型は非常に小さくなる。
大きさ	センサ部が小型	センサ部が大型	図 1-2-6a、図 1-2-6b に示したように、刺激値型が対物レンズ部とセンサ部から構成されるのに対して、分光型は3つのレンズ部、センサ部に加え、回折格子も必要となり、多くの光学部品が必要となります。さらに各レンズ部では結像のために、一定の空間距離を必要とします。
価格	安い	高い	図 1-2-6a、図 1-2-6b に示したように分光型はセンサ部の構成部品が多く、且つ高価な部品を使用しています
偏光の影響	受けない	受ける	刺激値型には偏光の影響を受ける光学部品は使用していません。他方分光型は、偏光の影響を受ける光学部品である回折格子(Grating)を使用しています。

(※注 1-2-1)

CAシリーズの場合、校正光源として、CA-210はLCDを用いており、CA-100PlusはCRTを用いています。CRT間ではCRTの種類が異なっても、蛍光体の発光特性が似ており分光放射輝度の差は大きくありません。また、LCDについても同様な傾向があります。

(※注 1-2-2)

例えば、CA-210、CA-100Plusでは 99 の任意校正係数をメモリすることができます。

(※注 1-2-3)

分光放射色彩計では分光応答度は等色関数に一致しています。これは、刺激値直読式における絶対値の誤差要因であった、「等色関数との差」がないということを意味します。分光放射色彩計では、被測定光源ごとに校正することなく、絶対値精度の高い測定値を得ることができます。

(※注 1-2-4)

刺激値直読式色彩計としては、CS-100A(コニカミノルタ)、BM-7(トプコン)、フォトマルを使用したBM-5A(トプコン)が有名です。また、分光色彩計としては、CS-1000(コニカミノルタ)、SR-3(トプコン)、PR-704(フォトリサーチ)が有名です。

トプコンは、株式会社トプコンの商標です。フォトリサーチは、Photo Reserch 社の商標です。

1-3 アナライザーモード

1-3-1 アナライザーモードの概要

アナライザーモードは生産ラインで、ディスプレイのホワイトバランスを高速に調整する場合に使用する測定モードです。

ホワイトバランス調整時に、例えばGを調整すればyのみが変化するのであれば、目的のyに対する過不足分だけ、Gを調整すればよいので、調整は簡単に行えます。

ところが、実際にはGを調整すればyだけではなく、xも変化します。これは、図1-3-1に示すように、Gの放射分光範囲に対して、等色関数 $y(\lambda)$ だけでなく、 $x(\lambda)$ の成分が存在するためです。

例えば、目的の色に対してyのみがずれている場合に、Gを調整することで色度yを目的の値にしても、今度はxがずれてしまいます。また、ここで、Rを調整してxを目的の値にしても、今度はyがずれてしまうということになります。

そこで、Gを調整した時に変化するx、yの量を人が概ね覚えておき、それをもとに調整するという方法がとられることがあります。ところが、この方法では熟練を要すること、またディスプレイの種類が異なるとその比率が異なるためディスプレイごとに覚えなおす必要があることなど、課題があります。

アナライザーモードは上述の不具合を解消するために設けられた機能です。このモードを使用すると、xyではなくR、G、Bの強度を直接得ることができるので、ホワイトバランス調整を簡単に行うことができます。

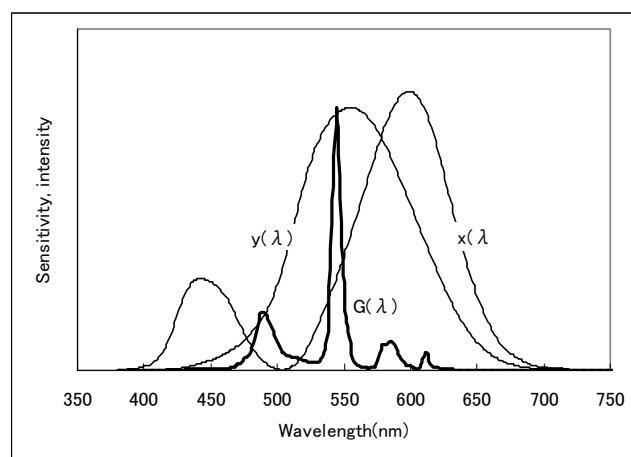


図 1-3-1 Gの分光放射輝度とx、yの等色関数

1-3-2 アナライザーモードの原理・概念の説明

[👉 さらに詳しく](#)

以下にアナライザーモードの原理について説明します。

まず、以下の条件で考えます。

- 簡単のため、センサーはx、yの2種類、ディスプレイの単色はR、Gの2色からなるものとします。
- ある色Wを構成するR、Gの発光強度は、それぞれ k_r 、 k_g で表されるものとします。
ある色Wを測定した場合のx、yセンサーの出力を X_w' 、 Y_w' とします。
- 調整目標とする色Wを構成する、R、Gの発光強度 k_r 、 k_g は1(=100%)とします。
その時のR、Gに対するx、yセンサーの出力をそれぞれ、 X_r 、 Y_r 、 X_g 、 Y_g とします。

ある色Wを測定した場合のx、yセンサーの出力 X_w' 、 Y_w' は、「Rに対するx、yセンサーの出力」と「Gに対するx、yセンサーの出力」の和となりますから、下式のように表わすことができます。

$$\begin{pmatrix} X_w' \\ Y_w' \end{pmatrix} = k_r \begin{pmatrix} X_r \\ Y_r \end{pmatrix} + k_g \begin{pmatrix} X_g \\ Y_g \end{pmatrix} \quad \dots\dots (1-3-1)$$

ここで、ある色Wを測定した時に、 k_r 、 k_g の値を知ることが、すなわち、目的の色W($k_r=k_g=1$)に対するR、Gの過不足分を知ることになります。

以下に、具体的な数値を用いて、任意の色Wを測定した時に、 k_r 、 k_g を知る手順を説明します。

①調整目標とする色Wを構成する、R、Gをそれぞれ表示した時のx、yセンサーの出力 X_r 、 Y_r 、 X_g 、 Y_g を「rgb発光特性の入力」手続きによって求めます。
本例では、その結果が

$X_r=5, Y_r=2$ (R表示の場合のx, yセンサー出力)

$X_g=1, Y_g=4$ (G表示の場合のx, yセンサー出力)

であったとします(これは「rgb発光特性の入力」操作を測定器に対して行うことにより、測定器が自動的に実施するものです)。

②次に色W'をx, yセンサーで測定します(これは通常の測定を意味します)。ここで、センサーから得られた各出力が $X=11, Y=8$ であったとします。(1-3-1)式より、

$$X \text{ については } 11 = k_r \times 5 + k_g \times 1$$

$$Y \text{ については } 8 = k_r \times 2 + k_g \times 4$$

と表すことができます。

この連立方程式を解くと、 $k_r=2, k_g=1$ が得られます。

つまり、色W'は目的の色Wを構成するR, G強度と比較して、Rが2倍、Gが1倍となっていることが分かります。

このことを、ベクトルを用いてさらに説明します。

目的の色W、それを構成する色R, G、及びそれらの色に対するx, yセンサーの出力の関係は図2のように表現することができます。

XY座標上で、色R, Gをベクトルとしてとらえ、その合成がWになると考える訳です。

①の操作は、この図 1-3-2 に示した関係を求めることを意味しています。

次に、②で示した、ある色W'を測定した結果、 $X=11, Y=8$ となるということは、図 1-3-3 に示したW'ベクトルを求めたことを意味します。

ところで、図 1-3-3 から、W'ベクトルはRの2倍とGベクトルの和となっていることが分かります。

つまり、

$$W' \text{ ベクトル} = k_r \times R \text{ ベクトル} + k_g \times G \text{ ベクトル}$$

と考えた場合、 $k_r=2, k_g=1$

となっているということになります。

結局、上記①②は、この図 1-3-2、図 1-3-3 の関係から、 k_r, k_g を求める操作を行ったと言い換えることができます。

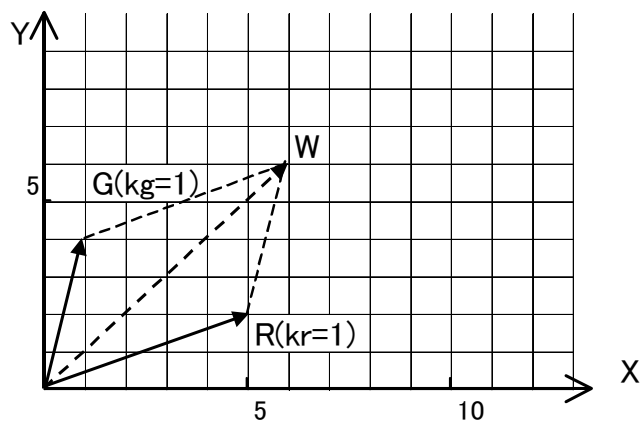


図 1-3-2 目的色Wと構成色R, Gの関係

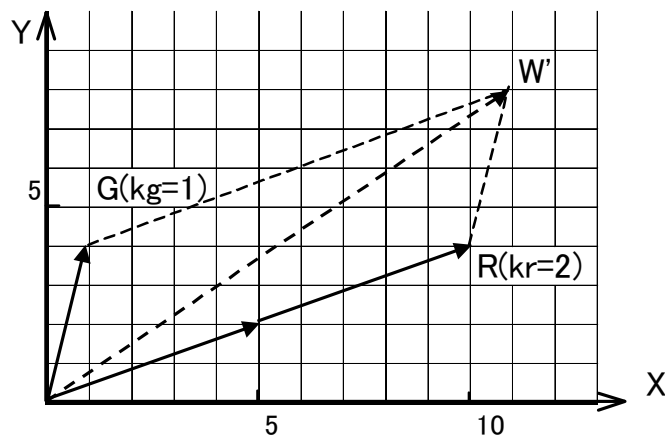


図 1-3-3 ある色W'と構成色R, Gの関係

1-3-3 アナライザーモードの原理・実際の計算説明 さらに詳しく

センサーがx, y, zの3種類、ディスプレイの単色がR, G, Bの3色からなる場合についての操作を数式で示します。
以下のように、変数を定義します。

- ある色W'を構成するR, G, Bの発光強度は、それぞれkr, kg, kbで表されるものとします。
ある色W'を測定した場合のx, y, zセンサーの出力をXw', Yw', Zw'とします。
- 調整目標とする色Wを構成する、R, G, Bの発光強度kr, kg, kbは1(=100%)とします。
その時のR, G, Bに対するx, y, zセンサーの出力をそれぞれ、
Xr, Yr, Zr, Xg, Yg, Zg, Xb, Yb, Zb とします。

ある色W'を測定した場合のx, y, zセンサーの出力Xw', Yw', Zw'は、R, G, Bに対するx, y, zセンサー出力の和となりますから、下式のように表わすことができます。

$$\begin{pmatrix} X w' \\ Y w' \\ Z w' \end{pmatrix} = kr \begin{pmatrix} X r \\ Y r \\ Z r \end{pmatrix} + kg \begin{pmatrix} X g \\ Y g \\ Z g \end{pmatrix} + kb \begin{pmatrix} X b \\ Y b \\ Z b \end{pmatrix}$$

$$= \begin{pmatrix} X r & X g & X b \\ Y r & Y g & Y b \\ Z r & Z g & Z b \end{pmatrix} \begin{pmatrix} k r \\ k g \\ k b \end{pmatrix} \quad \dots\dots(1-3-2)$$

よって、調整目標とする色Wを構成する、R, G, Bの発光強度に対する、各強度比kr, kg, kbは

$$\begin{pmatrix} k r \\ k g \\ k b \end{pmatrix} = \begin{pmatrix} X r & X g & X b \\ Y r & Y g & Y b \\ Z r & Z g & Z b \end{pmatrix}^{-1} \begin{pmatrix} X w' \\ Y w' \\ Z w' \end{pmatrix} \quad \dots\dots(1-3-3)$$

で算出することができます。

このように、ある色W'を測定した時にkr, kg, kbの値を得ることで、目的の色W(kr=kg=kb=1)に対するR, G, Bの過不足分を知ることができます。

1-4 マトリクス校正

1-4-1 マトリクス校正概要

従来、3刺激値色彩計での校正(以下白色1点校正と呼びます)とは、ある光源のある色に対する測定値を既知の値に一致させることを意味していました。

ところで、この白色1点校正では、同じ光源(ディスプレイ)であっても、ある色に対して得た校正値を用いて同じディスプレイの他の色を測定した場合、誤差が生じ得ます。特に、白色に対して得た校正値を用いて単色(R, G, B)を測定した場合の誤差は大きなものとなり得ます。[\(「1-2-3 刺激値直読式色彩計の絶対値誤差」参照\)](#)

マトリクス校正とは、RGBの加法混色(※注 1-4-1)が成り立つディスプレイに対して適用できる校正方法で、1色に対してではなく、RGBWの4色に対して同時に校正を行う方法です。この方法を採用することで、あるディスプレイに対して、広い色範囲において誤差の小さい測定値を得ることができます。

1-4-2 マトリクス校正概念説明

この概念を、図を用いて説明します。

あるディスプレイにおいて測定値と真値(所望の値)の関係が、図 1-4-1a のようであるとします。この例では、ある白色に対する測定値を○印、真値を×印で示しています。この測定値は真値よりxがマイナス方向、yがプラス方向に誤差をもっていることになります。また、ディスプレイの色再現範囲は測定値を用いた場合、実線で囲んだ領域となり、真値では破線で示す領域となることを示しています。

白色1点校正とは、白色1点の測定値(○印)を真値(×印)に一致させることであると言えます。この結果、図 1-4-1a の測定値に対応する実線で示した領域(3角形)は矢印方向に移動し、測定値と真値の関係は図 1-4-1b に示すようになります。この結果、図から分かるように、白色1点においては測定値と真値は一致するものの、白色以外の範囲では、依然、測定値と真値の間に差が残っていることが分かります。図の例では、白色1点校正を行うことによって、R単色においては誤差が拡大しています。

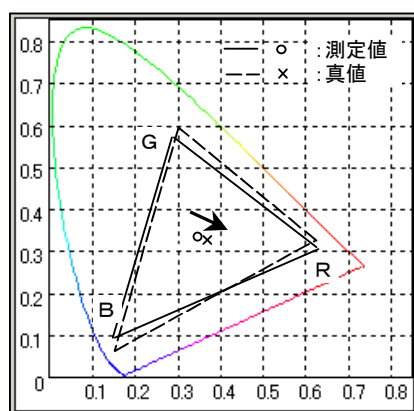


図 1-4-1a 白色1点校正前の測定値と真値

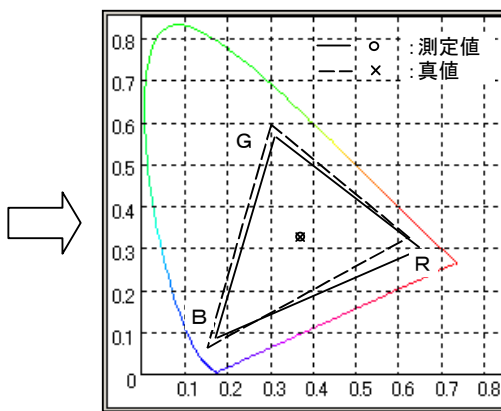


図 1-4-1b 白色1点校正後の測定値と真値

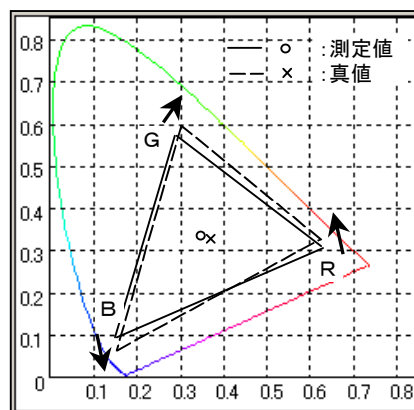


図 1-4-2a マトリクス校正前の測定値と真値

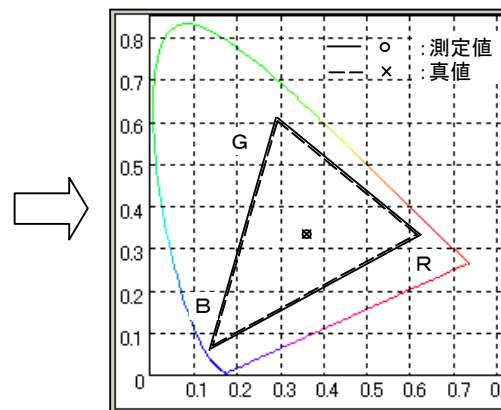


図 1-4-2b マトリクス校正後の測定値と真値

図 1-4-2a, 図 1-4-2b にマトリクス校正前後の測定値と真値の関係を示します。

マトリクス校正では、単色においても測定値と真値を一致させるようにします。すなわち、図 1-4-2a のように、各単色において、測定値と真値に差がある場合に、矢印で示す方向に測定値が変化するように測定値(実線)に対し線形変換(1次変換)を行うことにより、測定値と真値を一致させます(図 1-4-2b)。

言い換えると、マトリクス校正を行うことは、その1次変換係数を求めることを意味します。

刺激値直読式色彩計を用いて、あるLCDに対して、白色1点校正(校正点は白色 6500K)を行った場合と、マトリクス校正を行った場合での色温度の異なる白色測定時の絶対値誤差、及びRGB単色を測定した場合の絶対値誤差を数値実験により算出した結果を表1に示します。

この表から、マトリクス校正では、校正を行ったLCDに対しては理論上、全ての色度範囲において誤差が発生しないことが分かります。これに対し、白色1点校正では、校正点である白色 6500Kでは絶対値誤差が発生しませんが、単色では大きな誤差が発生することが判ります。(※注 1-4-2)

表 1-4-1 各校正方法における絶対値誤差の数値実験結果

	白色1点校正		マトリクス校正	
	Δx	Δy	Δx	Δy
白色6500K	0.0000	0.0000	0.0000	0.0000
白色5000K	0.0017	0.0017	0.0000	0.0000
白色13000K	0.0020	0.0018	0.0000	0.0000
R単色	0.0282	0.0241	0.0000	0.0000
G単色	0.0399	0.0231	0.0000	0.0000
B単色	0.0128	0.0081	0.0000	0.0000

1-4-3 マトリクス校正処理内容説明 1(RGB校正) さらに詳しく

まず、校正対象となるR, G, B(基準R, G, Bと呼びます)に対して校正值x, y, Lvを設定します(マトリクス校正值入力を意味します)。この設定されたx, y, Lv値からX, Y, Zカウントを算出することができます。このR, G, Bに対する校正值から算出されるX, Y, Zカウントを、それぞれ、Xr', Yr', Zr', Xg', Yg', Zg', Xb', Yb', Zb' とします。

さて、アナライザーモード([「1-3-1 アナライザーモードの概要」](#)、[「1-3-2 アナライザーモードの原理・概念の説明」](#)を参照)の原理に従えば、ある色W'を測定した場合、それを構成するR, G, B強度kr, kg, kb(基準R, G, Bに対する比)を算出することができます。

色W'の色度を算出するにあたって用いるべきXw', Yw', Zw'は、その色W'を構成するR, G, Bに対するX, Y, Zカウントの和となりますから、下式のように表わすことができます。

$$\begin{pmatrix} X w' \\ Y w' \\ Z w' \end{pmatrix} = k r \begin{pmatrix} X r' \\ Y r' \\ Z r' \end{pmatrix} + k g \begin{pmatrix} X g' \\ Y g' \\ Z g' \end{pmatrix} + k b \begin{pmatrix} X b' \\ Y b' \\ Z b' \end{pmatrix} \dots \dots \dots (1-4-1)$$

つまり、kr, kg, kbが求められれば、R, G, Bに対する校正值を用いて、任意の色W'に対するX, Y, Zは容易に求められます。

このX, Y, Zを用いて、x, y, Lv値を算出します。

さて、kr, kg, kbを算出するアナライザーモードの原理について説明します。

基準R, G, Bに対するx, y, zセンサーの出力をそれぞれ、Xr, Yr, Zr, Xg, Yg, Zg, Xb, Yb, Zb とします(これは「マトリクス校正」操作を本器に対して行うことにより、測定器が自動的に実施するものです)。

ある色W'を測定した場合

色W'のx, y, zセンサーの出力Xw, Yw, Zwは、R, G, Bに対するx, y, zセンサー出力の和となりますから、下式のように表わすことができます。

$$\begin{pmatrix} X w \\ Y w \\ Z w \end{pmatrix} = k r \begin{pmatrix} X r \\ Y r \\ Z r \end{pmatrix} + k g \begin{pmatrix} X g \\ Y g \\ Z g \end{pmatrix} + k b \begin{pmatrix} X b \\ Y b \\ Z b \end{pmatrix} \dots \dots \dots (1-4-2)$$

$$= \begin{pmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{pmatrix} \begin{pmatrix} k_r \\ k_g \\ k_b \end{pmatrix}$$

よって、基準R, G, Bの発光強度に対する、各強度比kr, kg, kbは

$$\begin{pmatrix} k_r \\ k_g \\ k_b \end{pmatrix} = \begin{pmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{pmatrix}^{-1} \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} \quad \dots \dots (1-4-3)$$

で算出することができます。

1-4-4 マトリクス校正処理内容説明 2(WRGB校正) さらに詳しく

ところで、ディスプレイによっては完全には加法混色(※注 1-4-1)が成り立たないものがあります。この場合、R, G, Bに対するマトリクス校正を実施しただけでは、単色での測定値は校正値に一致するものの、白色付近の測定で誤差が発生することがあります。

この問題を解決するために、R, G, Bマトリクス校正により得られた測定値に対してさらに白色1点校正を行うという方法があります。

以下に、この方法における処理内容を説明します。

但し、

R, G, Bマトリクス校正係数を用いて得られた校正用白色に対する測定カウント値をXw'、Yw'、Zw'校正用白色に対する校正設定値x, y, Lvから算出されるカウントをXw1, Yw1, Zw1 とします。

白色1点校正係数は、単純にX, Y, Zの比となりますから、

$$\begin{aligned} k_{x1} &= X_{w1} / X_{w'} \\ k_{y1} &= Y_{w1} / Y_{w'} \\ k_{z1} &= Z_{w1} / Z_{w'} \end{aligned} \quad \dots \dots (1-4-4)$$

となります。

また(1-4-1)式は、

$$\begin{pmatrix} X_{w'} \\ Y_{w'} \\ Z_{w'} \end{pmatrix} = \begin{pmatrix} X_r' & X_g' & X_b' \\ Y_r' & Y_g' & Y_b' \\ Z_r' & Z_g' & Z_b' \end{pmatrix} \begin{pmatrix} k_r \\ k_g \\ k_b \end{pmatrix} \quad \dots \dots (1-4-5)$$

と書けます。

よって、「R, G, Bマトリクス校正係数に白色1点校正を加えた」校正により最終的に得られるカウントをXw2, Yw2, Zw2 としますと、(1-4-4) (1-4-5)式より、

$$\begin{pmatrix} X_{w2} \\ Y_{w2} \\ Z_{w2} \end{pmatrix} = \begin{pmatrix} k_{x1} & 0 & 0 \\ 0 & k_{y1} & 0 \\ 0 & 0 & k_{z1} \end{pmatrix} \begin{pmatrix} X_r' & X_g' & X_b' \\ Y_r' & Y_g' & Y_b' \\ Z_r' & Z_g' & Z_b' \end{pmatrix} \begin{pmatrix} k_r \\ k_g \\ k_b \end{pmatrix} \quad \dots \dots (1-4-6)$$

となります。(※注 1-4-3)

(※注 1-4-1)

2種類以上の原色を混合させることを意味します。

ここでは、原色R, G, Bを用いてカラー画像を表示するディスプレイにおいて、例えば、ある白色画像に対するXYZ値と、その白色を構成する各原色画像に対するXYZ値を合計した値が一致する場合を、加法混色が成り立つと表現しています。

す。

(※注 1-4-2)

数値実験では、加法混色が成り立つとの条件で行っています。

また、分光器で測定した3刺激値色彩計の分光応答度 $x(\lambda)$, $y(\lambda)$, $z(\lambda)$ 、市販のあるLCDの分光放射輝度 $S(\lambda)$ を使用し、センサーの出力が

$$X = \sum S(\lambda) \cdot x(\lambda) \cdot \Delta \lambda$$

$$Y = \sum S(\lambda) \cdot y(\lambda) \cdot \Delta \lambda$$

$$Z = \sum S(\lambda) \cdot z(\lambda) \cdot \Delta \lambda$$

で得られるとして色度誤差を算出しています。

(※注 1-4-3)

CA シリーズではマトリクス校正として、この「R, G, Bマトリクス校正係数に白色1点校正を加えた」方法を採用しています。

参考文献

- ・Y. Ohno, S.W. Brown: Proc. of the IS&T sixth color imaging conference 65--68(1998)
「Four-color Matrix Method for Correction of Tristimulus Colorimeters-Part2」
- ・特許 特開平 6-323910

1-5 光学系の特長

1-5-1 光学系の構成

CA-210は特殊な光学系を用いることで液晶測定に適した測定を実現しています。

光学系の構成を図 1-5-1 に示します。

主な光学構成要素は、対物レンズ、光ファイバブロック、オンチップ・レンズ、センサーです。

光源からの光は、対物レンズにより、光ファイバブロックの入射部へ集光されます。この集光された光は、光ファイバブロック内でミキシングされると同時に3分割され、x, y, z各センサーの受光部前に出射されます。さらに出射光はセンサー上のオンチップ・レンズにより集光され、センサーが受光します。

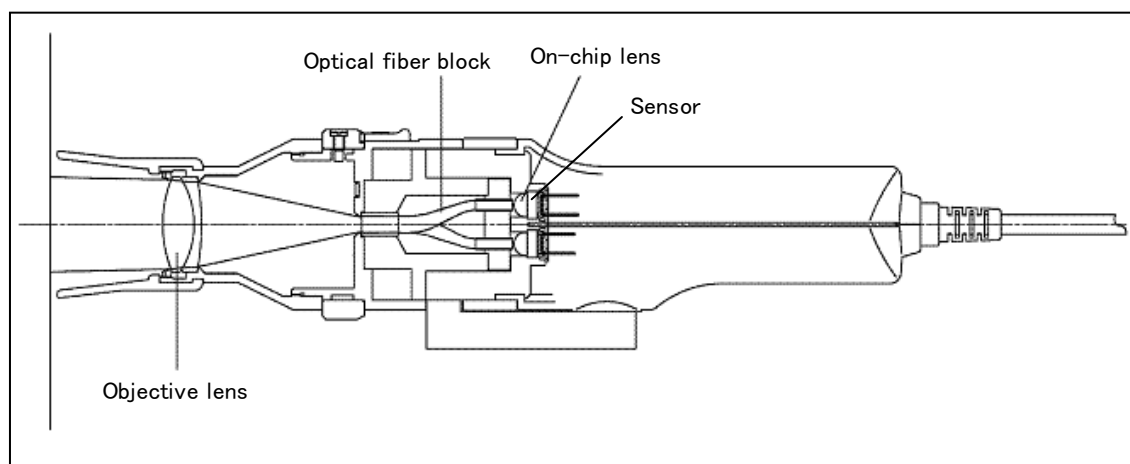


図 1-5-1 CA-210光学系の構成

1-5-2 光学系と測定上の特長

1-5-2-1 低輝度測定の実現

より低輝度までの測定を実現するためには、入射光をセンサ部に導くまでの光量のロスをどれだけ小さくできるかがポイントとなります。

従来は、図 1-5-2 に示すように入射光をレンズを通してそのまま、3つのセンサー(x, y, zセンサー)上に集光する方法などが採られてきました。この方法では、センサのない部分にも光が照射されることになり、損失が大きくなるという問題がありました。

CA-210は光ファイバを用いることにより、従来の方法と比較して光の伝達効率を向上させました。具体的には、図 1-5-1 に示すように、入射光は、レンズを通して光ファイバブロックへ集光します。次に、この光は、光ファイバ内を通過し、ファイバから出射された光は直接オンチップ・レンズによりセンサー上に集光させています。この結果、光の伝達ロスをなくし、低輝度測定が可能になっています。

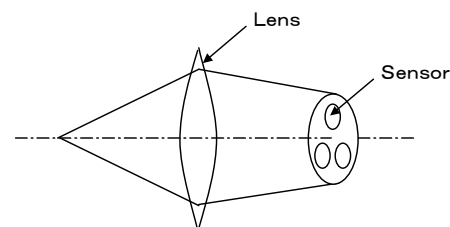


図 1-5-2 従来の測定器の構成

1-5-2-2 狭い開口角 & 均一な開口角

人がディスプレイを見る場合、比較的狭い角度の発光成分のみを視認しています。よって、人が視認する輝度、色度と相関性の高い測定値を得るためには、測定器の開口角も同じように狭い角度であることが必要になります。および、液晶は、視野角度特性を持っているので、異なる開口角度で測定すると、測定値が異なってしまいます。液晶の測定方法の規格である液晶IEC 61747-6においては、液晶を評価する場合の測定器は開口角が 5° 以内であることが示されています(開口角とは、図 1-5-3a の $\theta_1, \theta_2, \theta_3$, 図 1-5-3b の Ψ_1, Ψ_2, Ψ_3 のこと)。

CA-210は開口角が 5° で、この規格を満足しています。

従来の測定器における、「測定エリア内の位置」と「測定器への入射角」の関係を、図 1-5-3a に示します。なお、ここでは、測定器は被測定物発光面に垂直に設置するものとします。図 1-5-3a に示すように、従来の測定器では「測定エリア内の位置」の違いによって、開口角自体(図の $\theta_1, \theta_2, \theta_3$)は大きく変化することはありませんが、被測定光源の法線方向(図の破線)を基準として考えた場合、入射光の最大角度(図の θ'_1, θ'_3)は大きく異なります。測定エリア端部か

らは、開口角を大きく越える角度の光を受光してしまいます。

これに対し、CA-210では、特殊な光学系を採用することで、図 1-5-3b に示すように、測定エリア (φ27) 内のどの位置においても、被測定光源の法線方向に対して対称となる角度の光を受光します。CA-210は開口角が5° となっていますので、被測定光源の法線方向 (図 1-5-3b の破線) を基準として考えた場合、±2.5° 以内の発光成分のみの光を受光しています。

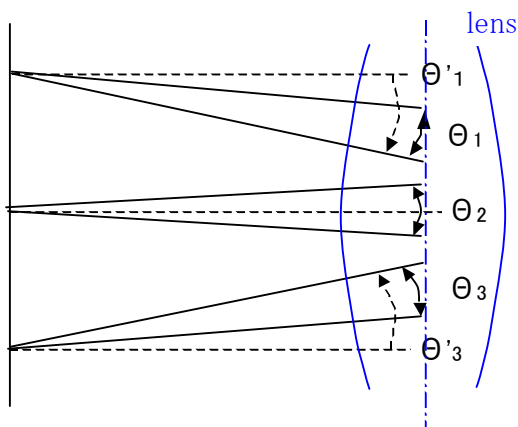


図 1-5-3a 従来測定器の測定位置と入射角度

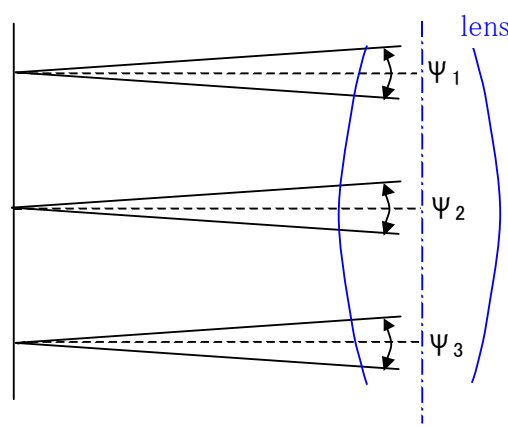


図 1-5-3b CA-210の測定位置と入射角度

1-5-2-3 輝度、色度むらの影響の軽減

一般に測色計の課題として、測定対象物の輝度、色度むらが色度測定値に影響するということがありました。

この影響を軽減するために、光学系に拡散板などを含む構成とする方法などが採られてきました (図 1-5-4b)。ところで、この方法では、輝度むらの影響を軽減できる代わりに、センサに導く光量が減少しその結果感度が低下するという問題がありました。

CA-210では、測定対象物からの光をレンズにより拡散させ (図 1-5-4a に示すように、異なる測定位置からの光を一様に光ファイバブロック入光部に集光させるようにしています)、その拡散光を、そのまま光ファイバブロックを通してセンサが受光する構成を採用しており、感度を低下させることなく、輝度、色度むらの影響を軽減することを実現しています。

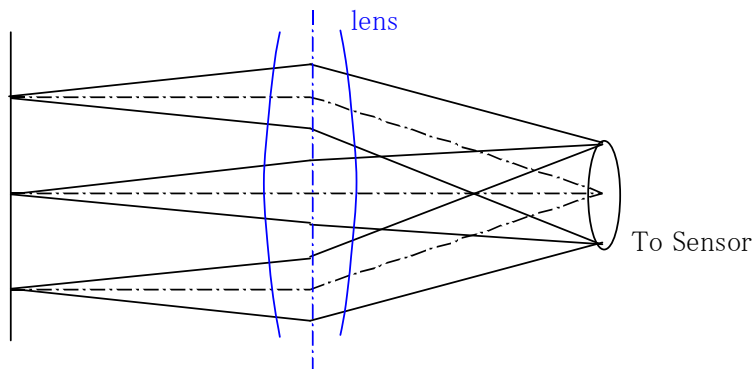


図 1-5-4a CA-210のレンズによる拡散

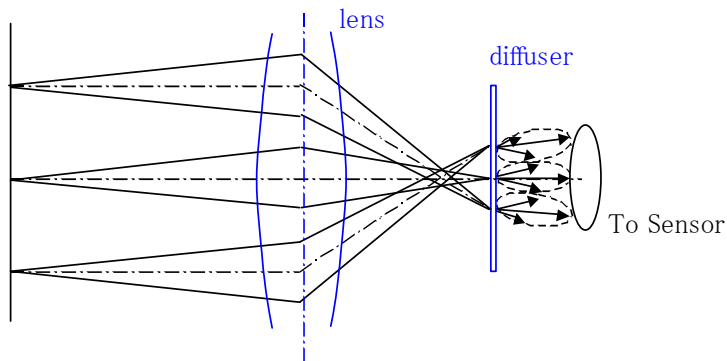


図 1-5-4b 従来器の拡散

1-6 回路系の特長

1-6-1 従来のA/D方式とCA-210との相違点

従来、輝度計、測色計においては、センサーからのアナログ出力信号をA/D変換(アナログ・デジタル変換)する方法として2重積分方式が多く採用されてきました。

この方式を図 1-6-1 を用いて説明します。

時間に対して図 1-6-1(左)のように発光量が変化している光源を測定するものとします。

この光エネルギーは、センサーによって電気エネルギーに変換され、積分器に入力されます。積分器では、一定期間(=t1)充電した後、電荷を放電します。この積分器の出力は図 1-6-1(右)のようになります。ここで、電荷を完全に放電してしまうまでに要する時間(=t2)は、充電された電荷量に比例しますから、この時間を計測することにより、センサー出力に対するデジタル値を得ることができます。

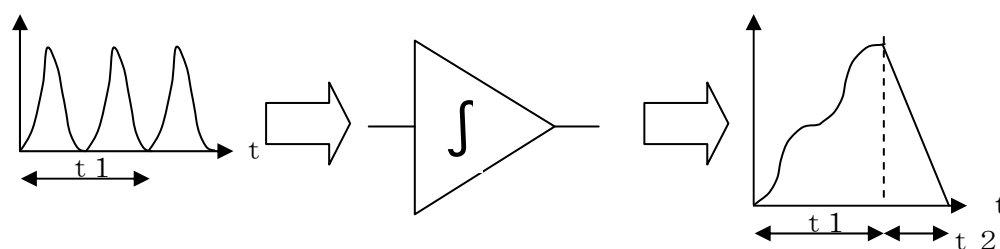


図 1-6-1 2重積分方式の原理

ところで、最近では、A/D変換素子の性能が向上したことから、2重積分方式に代わり、センサーの出力を直接A/D変換する方式が採用されてきています。

(以下、「逐次A/D方式」といいます)

すなわち、図 1-6-2 に示すように、センサーのアナログ出力に対して、一定期間(=t1)、A/D変換を繰返し実行し、得られたデジタル値(図 1-6-2 の●点に相当)、の総和を算出することでセンサー出力に対するデジタル値を得ます。

CA-210はこの逐次A/D方式を採用しています。

この方式の長所を以下に列挙します。

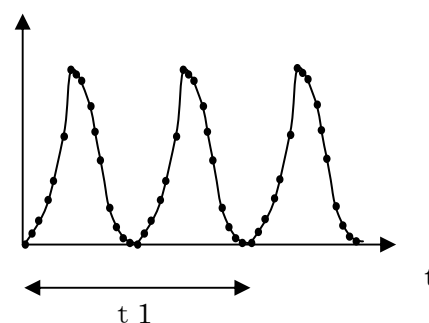


図 1-6-2 逐次A/D方式の原理

1-6-2 逐次A/D方式の長所

1-6-2-1 より低輝度までの測定が可能

低輝度測定を実現するためには、繰返し精度を向上させることが必須条件となります。また、繰返し精度を向上させることと、ノイズに対する有効信号成分(S/N比)を大きくすることは等価と言えます。

S/N比を大きくするためには、

- ① 有効信号成分を増大する
- ② ノイズを軽減させる

の2つの方法があります。

CA-210は②の方法により、S/N比を向上させています。

一般に、A/D値をn回積算することにより、ノイズ(対信号比)は \sqrt{n} に比例して軽減することが知られています。CA-210では、逐次A/D方式により、1回の測定値を得るために数100個のA/D値を得た後、これを積算しています。この結果、大幅なノイズ軽減を実現しています。

1-6-2-2 測定時間が短縮

例えば積分時間が、100msec の場合は

2重積分方式では、「アナログ出力信号を充電する」時間(図 1-6-1 の $t_1=100\text{msec}$)、「充電した電荷を放電する」時間(図 1-6-1 の t_2)が必要となります。これに対して、今回のA/D方式では、A/D変換を繰り返し実行する時間(図 1-6-2 の $t_1=100\text{msec}$)のみで済みます。

このことが測定時間の短縮に寄与します。(なお、これ以外にも、高速CPUの採用による演算時間の短縮、USBを使用することによる通信時間の短縮が寄与しています)

1-6-2-3 フリッカ測定が可能

図 1-6-2 に示すように、対時間でのAD値を得ることができますので、このAD値を演算することにより、フリッカ測定が可能となります。

CA-210 液晶フリッカ測定プローブは、2つの方式でフリッカ値を測定することができますが、両方式とも、上述のAD値を用いています。

AC成分とDC成分の比を算出する「コントラスト方式フリッカ値」は、対時間で得たAD値の最大値と最小値を検出することで、算出できます。

また、被測定光源の周波数成分に人の目の周波数応答度を乗じて算出する「JEITA方式フリッカ値」は、対時間で得たAD値にデジタルフーリエ変換を施し、演算処理することで算出することができます。

1-7 フリッカ測定

1-7-1 液晶ディスプレイにおけるフリッカとは

液晶ディスプレイに、網目状の画面を表示したとき(例えば Windows を閉じるときなど)に、画面に“ちらつき”が発生し、画面が非常に見づらくなることがあります。この“ちらつき”のことを液晶ディスプレイにおけるフリッカ(以下フリッカと記載)といいます。以下にフリッカが発生するメカニズムについて説明します。

1-7-1-1 フリッカ発生のメカニズム

液晶表示デバイスに直流の映像信号を入力し続けると液晶パネルの寿命が短くなることが判っています。また、液晶表示デバイスは正極性電圧でも負極性電圧でも応答します。そこで液晶デバイスに入力する映像信号は、一般に1フレーム(垂直同期周期)ごとに極性を反転させるようにしています。

ここで、同じ画面を表示し続ける場合を考えます。図 1-7-1 に示すように、各フレームの画像が同じレベルとなるためには、基準電位が、映像信号の振幅中心に一致している必要があります。これに対し、図 1-7-2 に示すように、基準電位がずれている場合、正負の映像信号成分が異なることとなります。この結果、フレーム周波数の1/2の周波数で映像信号が変化することになります。

例えば、垂直同期周波数が60Hzの場合、30Hzで映像信号が変化することになり、人間の目に応答する周波数以下となるため、人間に“ちらつき”(フリッカ)として認識されることとなります。

液晶にフリッカが発生すると、画面が非常に見づらくなるので、一般的に、液晶パネルでは、製造工程で基準電位の調整を行っています

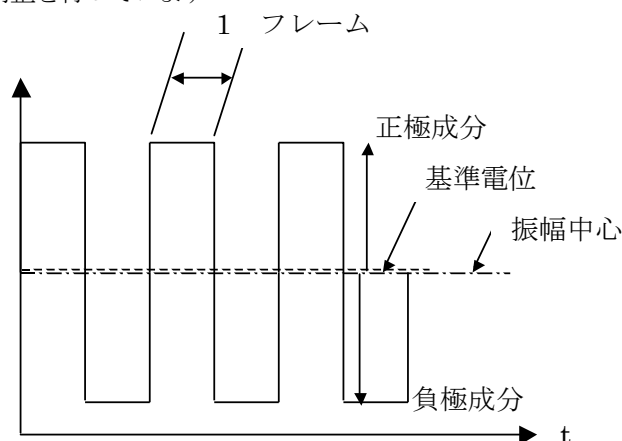


図 1-7-1 各フレームの画像レベルと基準電位
(理想状態)

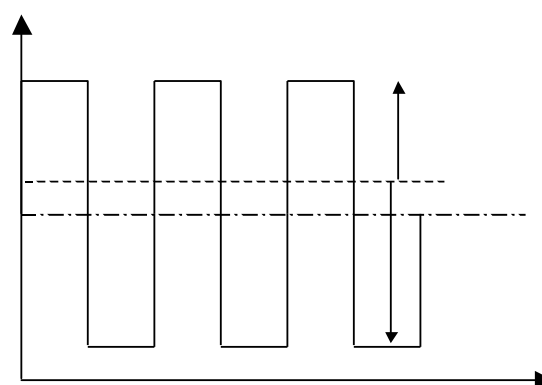


図 1-7-2 各フレームの画像レベルと基準電位
(フリッカ発生時)

1-7-1-2 液晶駆動方式とフリッカの発生し易い画像

近年、画面表示の均一性を高めるために、液晶表示デバイスへ入力する信号の極性を、画素ごとに反転させる液晶駆動素子が開発され、液晶パネルに使用されています。

この反転方式には、以下の2種類があります。

- ・ライン反転駆動方式: 1水平ラインごとに信号の極性を反転させる方式。
現在、小型液晶パネルで多く用いられています。
- ・ドット反転駆動方式: 水平ラインの画素について1画素ごとに信号の極性を反転させる方式。
ある水平ラインの画素が正極、負極、正極、・・・とした場合、次の水平ラインは負極、正極、負極、・・・と並ぶようにします(いわゆる市松模様)。
現在、大型液晶パネルで多く用いられています。

画面全体に均一な画像を表示した場合、この画素反転を行っている液晶パネルでは、上述の基準電位ずれに起因するフリッカは水平ラインごと、あるいは画素ごとに反転する形で発生することになります。この場合は、人の目には均質化されるため、“ちらつき”としてはほとんど認識されません。

ところで、ドット反転を行っている液晶パネルに、1画素ごとに発光/非発光を繰り返す市松模様(図 1-7-3)の画像を表示した

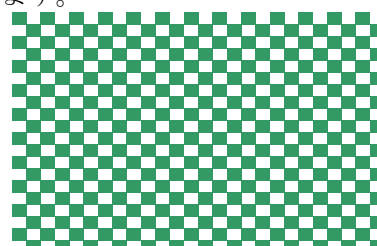


図 1-7-3 フリッカが発生し易い画像例(市松模様)

場合を考えます。この場合は、発光している画素は、あるフレーム期間同じ極性の信号が入力されていることになり、つまり画素ごとの反転が発生しないので、画面全体に“ちらつき”として認識されるようになります。

同様に、ライン反転を行っている液晶パネルに、図 1-7-4 のような画像を表示した場合も、画面全体に“ちらつき”が認識されるようになります。

冒頭、Windows を閉じるときなどに“ちらつき”が発生すると述べましたが、これは、その時に、画面が市松模様になるためです。

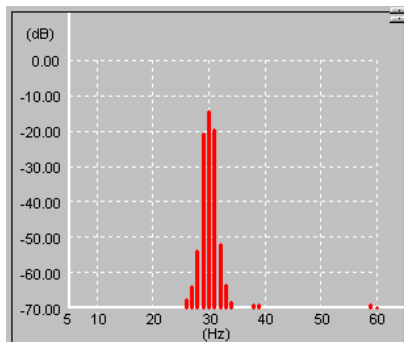


図 1-7-4 フリッカが発生しやすい画像例 (横縞模様)

フリッカ測定例

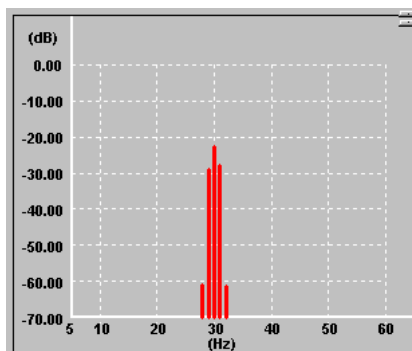
フリッカ量 大

JEITA 測定値
 -14.5 dB
 AC/DC 測定値 68.4 %
 測定画像ファイル F_Big.



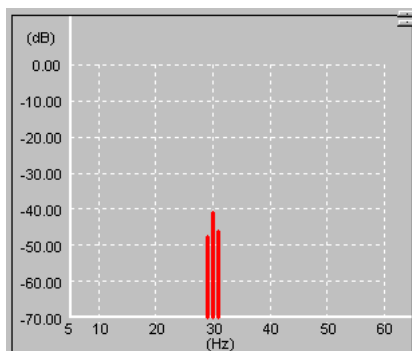
フリッカ量 中

JEITA 測定値
 -22.7 dB
 AC/DC 測定値 27.8 %
 測定画像ファイル F_Medium



フリッカ量 少

JEITA 測定値
 -40.9 dB
 AC/DC 測定値 2.7 %
 測定画像ファイル F_Small



注:測定ファイルの再生には、Media Player が必要です。
 Meida Player はマイクロソフト社の商標登録です。

参考文献

・宅和 幹雄:月刊ディスプレイ6月号, Vol.8 No.6 51(2002)

1-7-2 フリッカの測定方法

1-7-2-1 フリッカの測定

フリッカが発生している時の時間と輝度の関係は図 1-7-5 のようになります。この図からわかるように、輝度レベルは周期的に変化しており、この振幅が大きいほど“ちらつき”としてはっきり認識されます。

また、この輝度レベル変化の周期は、ディスプレイの垂直同期周期の2倍に一致します（「1-7-1 液晶フリッカとは」参照）。

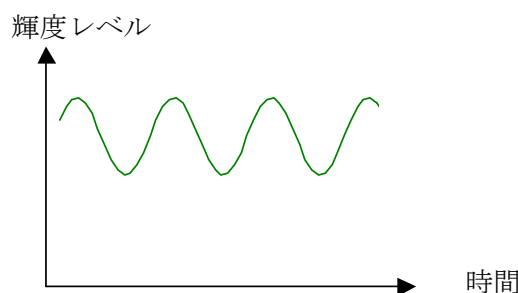


図 1-7-5 フリッカ発生時の時間と輝度の関係

フリッカを測定するための測定方式は、大きく2つの方式に分けることができます。

- ① 図 1-7-5 の輝度レベル変化を直流成分と交流成分の合成ととらえ、両成分の比からフリッカを求める方式
- ② 図 1-7-5 の輝度レベル変化を周波数成分に分解し、直流成分と最大交流成分の比からフリッカを求める方式

CA-210 液晶フリッカ測定プローブは、①の方式として“コントラスト方式”，②の方式として“JEITA方式”を採用し、フリッカの測定を行いません。

それぞれの方式の特長を表1に示します。

表 1-7-1 フリッカの測定方式と特長

各方式は、目的に応じて使い分ける必要があります。一般に、“コントラスト方式”は液晶パネル製造時の調整、検査の使用に適しています。また、“JEITA方式”は液晶パネルの開発・設計での使用に適しています。

各方式について、以下に説明します。

	コントラスト方式	JEITA方式
測定速度	測定速度が速い (16回/秒)	測定速度が遅い (0.5回/秒)
性能の特長	フリッカーの相対的な大小を知ることができる。 人間の目の周波数応答特性は考慮されていない。	フリッカーの絶対値を知ることができる。 人間の目の周波数応答特性が考慮されている。

1-7-2-2 コントラスト方式フリッカ測定の説明

1-7-2-2-1 コントラスト方式フリッカ測定の概要

ディスプレイの輝度レベルが図 1-7-6 のように変化している場合、これを、直流成分の上に交流成分が積み重なっているものと考えます。この交流成分/直流成分をフリッカ量として定義したものが“コントラスト方式”フリッカ値(以降、コントラスト・フリッカ値と言います)です。

コントラスト方式では、交流成分を $V_{max} - V_{min}$ 、直流成分を $(V_{max} + V_{min}) / 2$ であるとし、式(1-7-1)でフリッカ値を求めます。

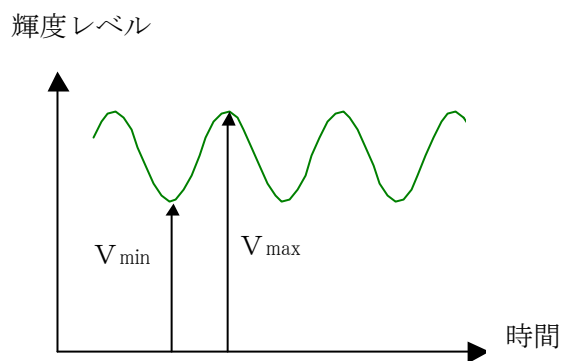



図 1-7-6 コントラスト方式の V_{max} 、 V_{min}

フリッカ値

= 交流成分/直流成分

$$= (V_{max} - V_{min}) / \{ (V_{max} + V_{min}) / 2 \} \times 100 \quad [\%] \quad \dots \quad (1-7-1)$$

1-7-2-2 コントラスト方式のデータ処理詳細  さらに詳しく

センサーの出力データ取得からフリッカ値算出にいたるまでの測定器内の処理内容を図 1-7-7 に示すフローに従い説明します。

- (A)～(C) :輝度センサーの出力信号に対して、高調波成分を除去する電氣的処理を行い、フリッカに対応する信号を取り出します。
- (C)～(D) :高周波成分除去後のアナログ信号に対して、一定期間(=t1)、A/D変換をおこない、演算に必要なデジタル値を得ます(図の●印)。デジタル値の総数は数百個としています。(D)～(E) :デジタル値に対して、最大値、最小値を検出し、それぞれをVmax, Vminと定義します。
- (E)～(G) :この値に対し、(1-7-1)式を適用することで交流成分/直流成分比を算出し、フリッカ値とします。交流成分算出にあたっては、交流成分の周波数に基づき、ローパス処理で減衰した成分を補正しています。

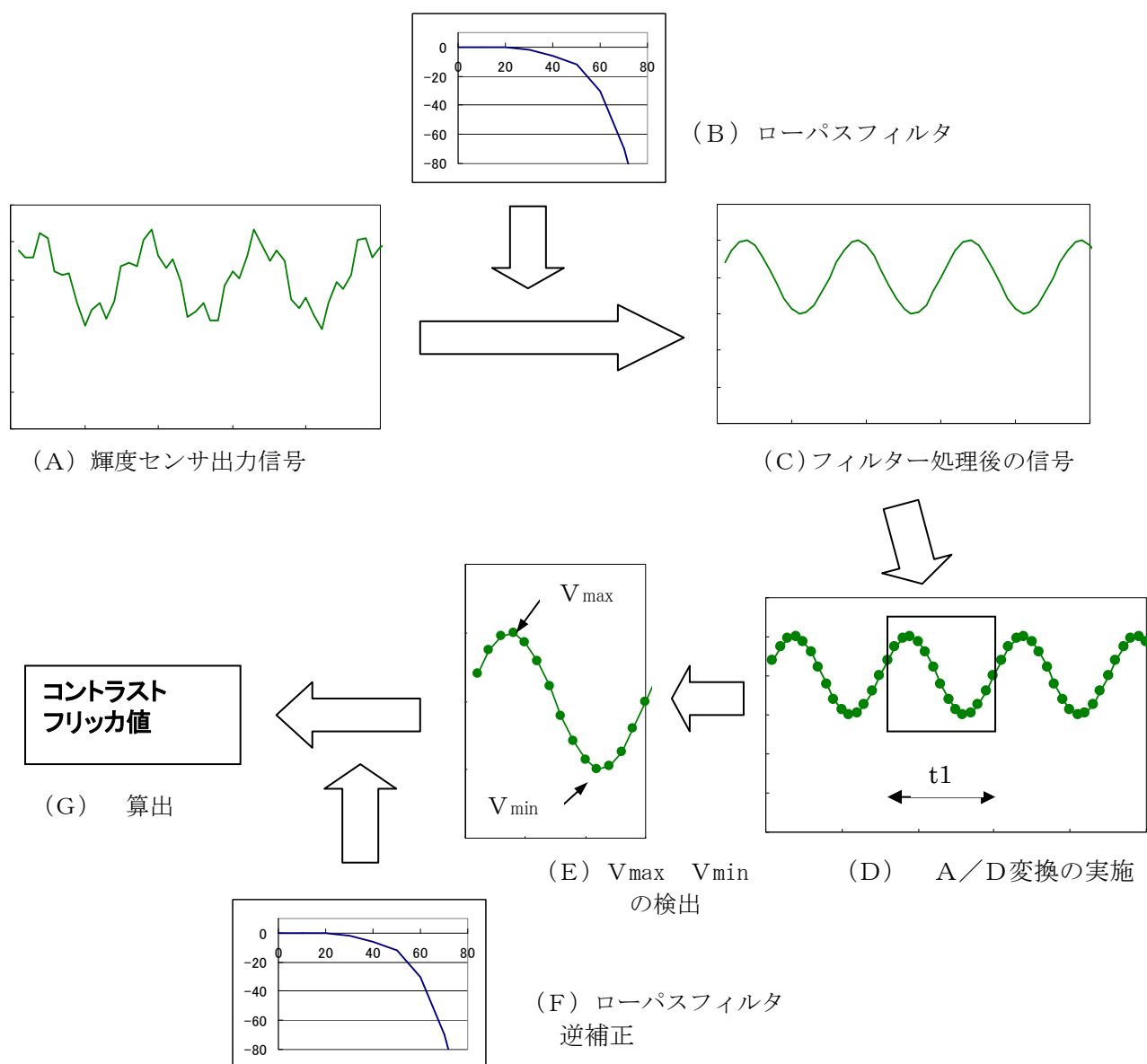


図 1-7-7 コントラスト方式の輝度センサ出力からフリッカ値算出までの処理フロー

1-7-2-2-3 VESA規格のコントラスト方式との相違点  さらに詳しく

VESA 305-5規格に、AC成分とDC成分の比に基づくフリッカ測定方法が規定されています。ここでは、VESA規格のコントラスト方式・フリッカ値と、CA-210 液晶フリッカ測定プローブのコントラスト・フリッカ値の相違点について説明します。

一言でいえば、相違は、VESA規格では人間の目の周波数応答特性を反映させているのに対し、高速測定を実現するために、CA-210 液晶フリッカ測定プローブでは反映させていないことです。

VESA規格では、フリッカ値の算出式として以下の2種が示されています。(※注 1-7-1)

$$\bullet \quad (V_{\max} - V_{\min}) / V_{\max} \times 100 \quad [\%] \quad \dots \dots \quad (1-7-2)$$

$$\bullet \quad (V_{\max} - V_{\min}) / \{(V_{\max} + V_{\min}) / 2\} \times 100 \quad [\%] \quad \dots \dots \quad (1-7-3)$$

ここで定義されている、Vmax , Vmin は人間の目の周波数応答特性(※注 1-7-2)が反映された値であり、また、(1-7-3)式はフリッカ値が小さい場合(13%未満)の場合に適用します。

具体的に、Vmax , Vmin を得るためには、センサーの出力信号を人間の目の周波数応答特性をもったローパスフィルタに通し、その信号の最大、最小レベルを検出することが考えられます。

人間の目の周波数応答特性を図 1-7-8、表 1-7-2 に示します。

CA-210 液晶フリッカ測定プローブのコントラスト・フリッカ値の算出式は、(1-7-3)式と同じです。ただし、人間の目の周波数応答特性を反映させていません。

ところで、フリッカの周波数は液晶ディスプレイの垂直同期周波数の1/2になります ([「1-7-1 フリッカとは」参照](#))。液晶ディスプレイの垂直同期周波数から交流成分の周波数f0を知ることができます。他方、f0の減衰率は表2から算出することができます。従い、コントラスト・フリッカ値から、(1-7-3)式で示したVESA準拠フリッカ値を算出することができます。

例えば、垂直同期周波数が60Hz、コントラスト・フリッカ値が10.0 [%]の場合を考えます。

フリッカ交流成分の周波数は

$$60 \times (1/2) = 30 \quad [\text{Hz}]$$

となります。交流成分に対する、目の周波数応答特性による減衰率は表1より、0.708となります。従い

$$10.0 \times 0.708 \approx 7.1 \quad [\%]$$

が、VESA規格に準拠したフリッカ値になります。

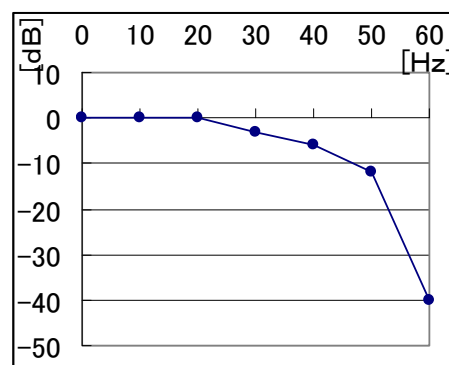


図 1-7-8 人間の目の周波数応答特性

表 1-7-2 人間の目の周波数応答特性

Frquency [Hz]	Factor	
	[dB]	[ratio]
0	0	1.000
10	0	1.000
20	0	1.000
30	-3	0.708
40	-6	0.501
50	-12	0.251
60	-40	0.010

1-7-2-3 JEITA方式フリッカ測定の説明


1-7-2-3-1 JEITA方式フリッカ測定の概要

JEITA方式フリッカ測定は、人間の目の周波数応答特性を正確に反映させた上でフリッカ値を定量化する方式です。

ディスプレイの輝度レベルが図 1-7-6 のように変化している場合、このフリッカ光を、直流成分の上に複数の周波数成分が積み重なっているものと考えます。そこで、まず、この「対時間の輝度変化データ」を、周波数成分(直流成分、及び各交流成分)に分解します(※注 1-7-3)。次に、各周波数成分の出力を“人の目の周波数応答特性”を考慮した値に変換します。

ここで得られた、各周波数成分のうち、0Hz(直流成分)以外で最大となるパワースペクトルを P_x 、直流成分のパワースペクトルを P_0 とすると、式(1-7-4)でフリッカ値(以降、JEITA・フリッカ値と言います)を求めます。

$$\text{フリッカ値} = 10 \times \log_{10} (P_x / P_0) \quad [\text{dB}] \quad \dots \dots (1-7-4)$$

1-7-2-3-2 JEITA方式のデータ処理詳細  さらに詳しく

具体的な例を挙げて、JEITA・フリッカ値算出にいたるまでの測定器内の処理内容を図 1-7-10 に示すフローに従い説明します。ここで、被測定光源は図 1-7-9(4)に示すように、発光周波数の異なる(1)、(2)、(3)3種類の光源が合わさったものであるとします。

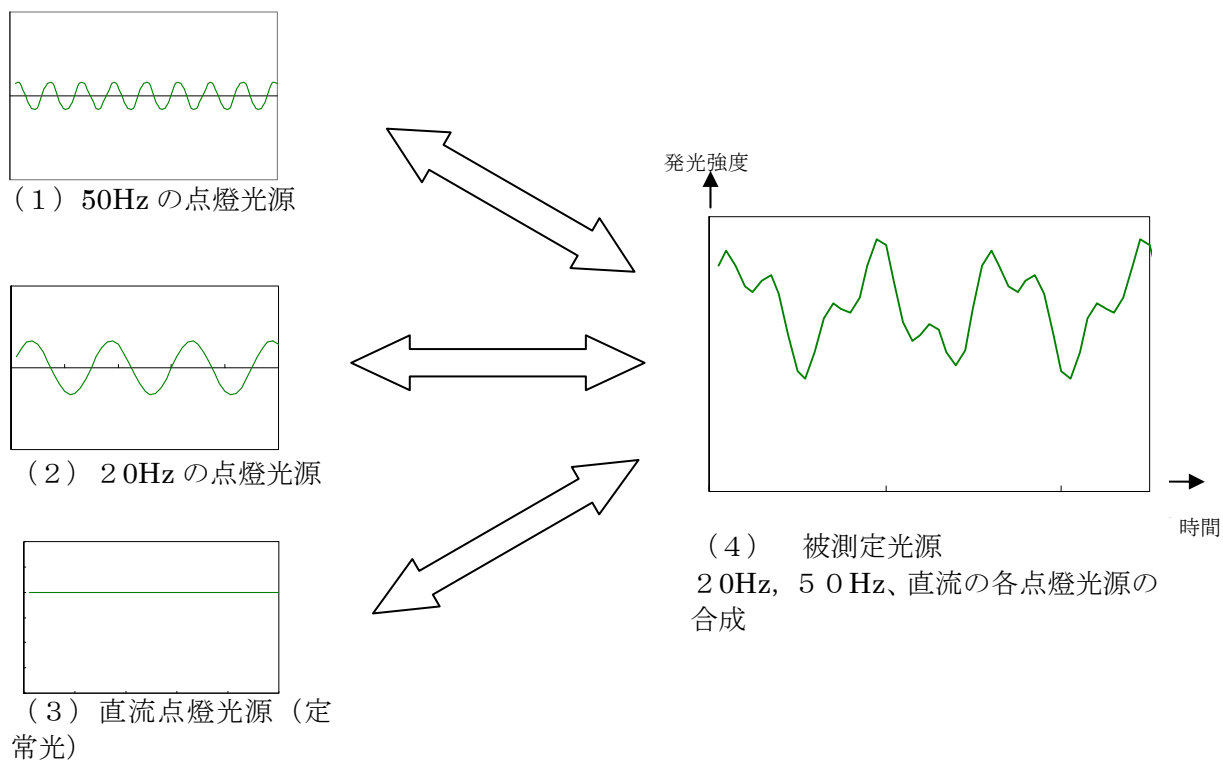


図 1-7-9 被測定光源

- (A) :センサーの出力信号に対して、一定期間(=t2)、A/D変換を繰返し実行し、演算に必要なデジタル値を得ます(図の●印)。(※注 1-7-4)
- (A)～(B):このデジタル値に対して、デジタル・フーリエ変換を行い、各周波数成分に分解します。被測定光源は図 1-7-9 に示した(1)、(2)、(3)3種類の光源を合成したものですから、これを分解するということは、図 1-7-9 の(1)、(2)、(3)光源の周波数と振幅値のデータを得ることと同じことを意味します。この結果、(B)のデータを得ます。
また、デジタル・フーリエ変換特有の誤差を軽減するために、このデジタル値に適当な「窓関数」処理

を行っています。(※注 1-7-5)

(B)～(D):この周波数成分に対して、(C)及び表2に示した人間の目の周波数応答特性(積分器)を反映させます。表2より、0Hz、20Hz 成分は減衰しないのに対し、50Hz 成分は0.251倍になることがわかります。この結果(D)を得ます。(※注 1-7-6)

(D)～(E): (D)から、0Hz以外で最大の成分(この例では20Hz)を交流成分として選択します。

また、0Hzの成分を直流成分とします。

これから直流成分、交流成分のパワースペクトルをそれぞれ求めP0、P1とし、(1-7-4)式によって、JEITA・フリッカ値を算出します。

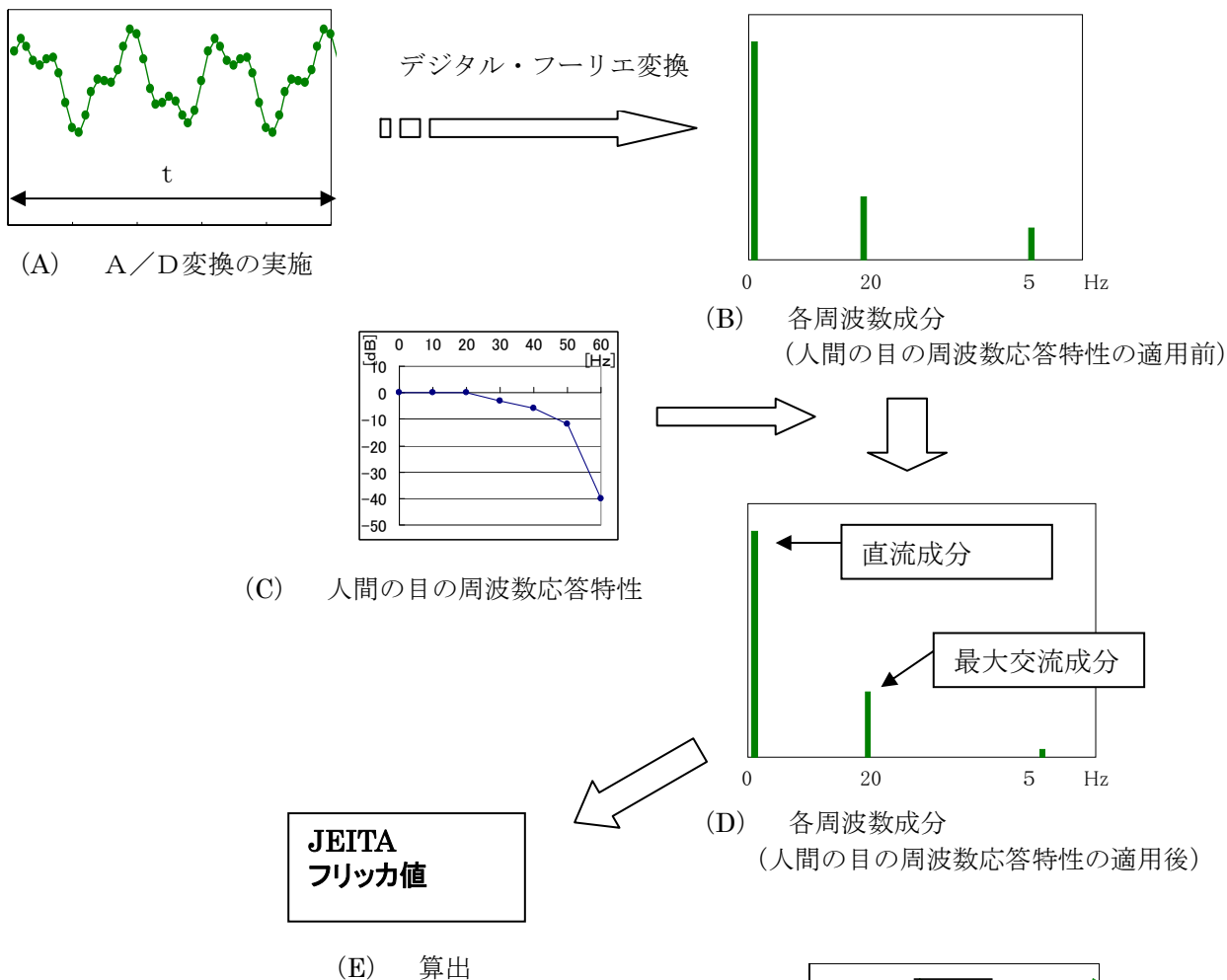


図 1-7-10 JEITA 方式のフリッカ値算出までの処理フロー

最後に、コントラスト方式との違いを説明します。

図 1-7-9 (4)の光源を測定した場合、コントラスト方式ではセンサー出力そのままの最大値、最小値の差(図 1-7-11 上の矢印)を交流成分ととらえます。

これに対して、JEITA 方式では、周波数分解しますので、20Hz のみを取り出して、その最大値、最小値の差(図 1-7-11 下の矢印)を交流成分と捉えることとなります。また、最大交流成分周波数が20Hz 以上の場合、人間の目の周波数応答特性による減衰率が考慮されることも相違点です

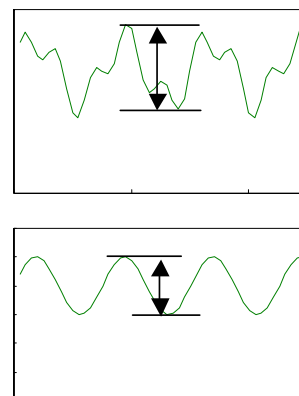


図 1-7-11 JEITA 方式とコントラスト方式の違い

1-7-2-3-3 JEITA方式の CA-210 液晶フリッカ測定プローブのデータ処理内容 [さらに詳しく](#)

ここでは、CA-210 液晶フリッカ測定プローブが、センサ出力取得、デジタルフーリエ変換処理を経て JEITA・フリッカ値を算出するまでの導出内容をに説明します。

1-7-2-3-3-1 JEITA方式の算出式

CA-210 液晶フリッカ測定プローブは、式(1-7-5)に従い、JEITA・フリッカ値を算出しています。

$$20 \times \log \frac{(\sqrt{2} \times \text{weight}(k) \times \text{FFT}(k))}{\text{weight}(0) \times \text{FFT}(0)} \dots\dots (1-7-5)$$

ここで

- k: 周波数
- weight(k): 周波数kの積分器定数
- FFT(k) : 周波数kの FFT 出力のうち、weight(k) × FFT(k)が最大のもの(k>0)。
- FFT(0) : DC 成分の FFT 出力

1-7-2-3-3-2 JEITA方式の算出式の説明

フリッカ JEITA 方式によるフリッカ値は(1-7-6)式で定義されています。

$$10 \times \log (P_x / P_0) \dots\dots (1-7-6)$$

Px: 積分器を通した後の交流成分の最大のパワースペクトル
 P0: 積分器を通した後の直流成分のパワースペクトル
 (EIAJ ED2522 による)

直流成分(x=1)と交流成分(x=sin t)におけるFFT出力、パワースペクトルの比の関係は以下のようになります。

表 1-7-3 FFT出力、パワースペクトルの比の関係

	①直流成分 (x=1)	②交流成分 (x=sin t)
FFT 出力	1	0.5
パワースペクトル	2	1

①と②においては、振幅という概念で捉えた場合、同一であるべきですが、FFT出力は 1:0.5の関係にあり、周波数成分のみ2倍することで、同一となります(逆に言うと、FFT では、交流成分が直流成分の 1/2 となって出力されます)。

すなわち、直流成分と交流成分の整合性をとると

$$\begin{aligned} \text{周波数成分の振幅} & 2 \times \text{FFT}(k) & (=A_k) & \dots\dots (1-7-7) \\ \text{直流成分の振幅} & \text{FFT}(0) & (=A_0) & \dots\dots (1-7-8) \end{aligned}$$

これをパワースペクトルに変換する場合、直流成分は振幅の2乗、交流成分は振幅の2乗の1/2とすればよいので、式(1-7-6)は

$$\begin{aligned} & 10 \times \log \frac{(2 \times \text{wFFT}(k))^2 / 2}{\text{wFFT}(0)^2} \\ = & 10 \times \log \frac{(\sqrt{2} \times \text{wFFT}(k))^2}{\text{wFFT}(0)^2} \\ = & 20 \times \log \frac{(\sqrt{2} \times \text{wFFT}(k))}{\text{wFFT}(0)} \dots\dots (1-7-9) \end{aligned}$$

ここで、wFFT(k)=weight(k) × FFT(k), wFFT(0)=weight(0) × FFT(0) によって、式(1-7-5)が導出されます。

1-7-2-3-3-3 JEITA方式の算出式の補足

実効値の考え方から、(1-7-6)式は、交流成分のみを(√2)で除すればよいので

$$10 \times \log \frac{(A_k / \sqrt{2})^2}{A_0^2} \dots \dots \dots (1-7-10)$$

ここで、Ak、A0は交流成分、直流成分の振幅と書けます。ここで(1-7-7)式、(1-7-8)式を適用すると

$$\begin{aligned} &= 10 \times \log \frac{(\sqrt{2} \times \text{wFFT}(k))^2}{\text{wFFT}(0)^2} \\ &= 20 \times \log \frac{(\sqrt{2} \times \text{wFFT}(k))}{\text{wFFT}(0)} \dots \dots \dots (1-7-11) \end{aligned}$$

となり、式(1-7-1)に一致します。

つまり、(1-7-5)式は実効値の考え方と整合性がとれていると言えます。

1-7-2-3-3-4 VESA規格との相違点 [さらに詳しく](#)

VESA 305-4規格に、交流成分の周波数分解によるに基づくフリッカ測定方法が規定されています。そこでは、フリッカ値が

$$20 \times \log \frac{(2 \times \text{wFFT}(k))}{\text{wFFT}(0)} \dots \dots \dots (1-7-12)$$

と定義されています。(交流成分、直流成分の振幅を単純に2乗しています)今回提示した式(1-7-1)と比較すると、log内の係数が異なります。JEITA・フリッカ値を用いて、VESA・フリッカ値を算出するためには

$$\begin{aligned} &(\text{VESA} \cdot \text{フリッカ値}) \\ &= 20 \times \log \frac{(2 \times \text{wFFT}(k))}{\text{wFFT}(0)} \\ &= 20 \times \log \frac{(\sqrt{2} \times \text{wFFT}(k))}{\text{wFFT}(0)} + 20 \times \log \sqrt{2} \\ &= (\text{JEITA} \cdot \text{フリッカ値}) + 20 \times \log \sqrt{2} \dots \dots \dots (1-7-13) \end{aligned}$$

つまり、JEITA・フリッカ値に約3.01[dB] (=20×log√2)を加えればVESA・フリッカ値になることがわかります。

(※注 1-7-1)

(1-7-3)式は、正確には、(Vmax-Vmin)/Vdc × 100 [%] と規定されています。

ここでは、Vdc = (Vmax + Vmin)/2 としています。

(※注 1-7-2)

人間の目の点滅に対する視覚のこと。人間の目は、30Hz付近以上の周波数より徐々に感度が低下し、60Hz以上では知覚できなくなる特性をもっています。(図 1-7-8, 表 1-7-2)

(※注 1-7-3)

これは、フリッカ成分が、垂直同期周波数の1/2の周波数の成分のみであることが条件となります。これ以外の周波数成分が含まれる場合は、それが誤差要因となることがあります。

(※注 1-7-4)

デジタル値は、総サンプリング期間:1秒(=t2)、 総サンプリング数 512 個です。

(※注 1-7-5)

デジタル・フーリエ変換特有の誤差とは、フーリエ変換が無限長さ時間のデータを扱うことを前提にしているのに対して、デジタル・フーリエ変換が有限長さ時間のデータを扱い演算していることに起因する誤差のことです。

「データ取得期間t2」と「交流成分の周期の整数倍」の差が大きい場合に誤差が大きくなる傾向があります。この誤差を軽減するために一般に「窓関数」が使用されます。

この結果、交流成分の検出精度は向上しますが、他方、検出分解能は低下します。

例えば、図 1-7-12 のようにある交流成分(例えば30Hz)が存在する場合、窓関数を使用せずにデジタル・フーリエ変換を行うと、図 1-7-12(B)のように、真値より小さい値が算出されることがあります。他方、窓関数を使用してデジタル・フーリエ変換を行うと、30Hzの成分は真値を得ますが、その成分の前後にもある成分が出力されてしまいます。ただし、この場合、30Hzの成分の真値を得ているため、JEITA・フリッカ値を正確に求めることができます。

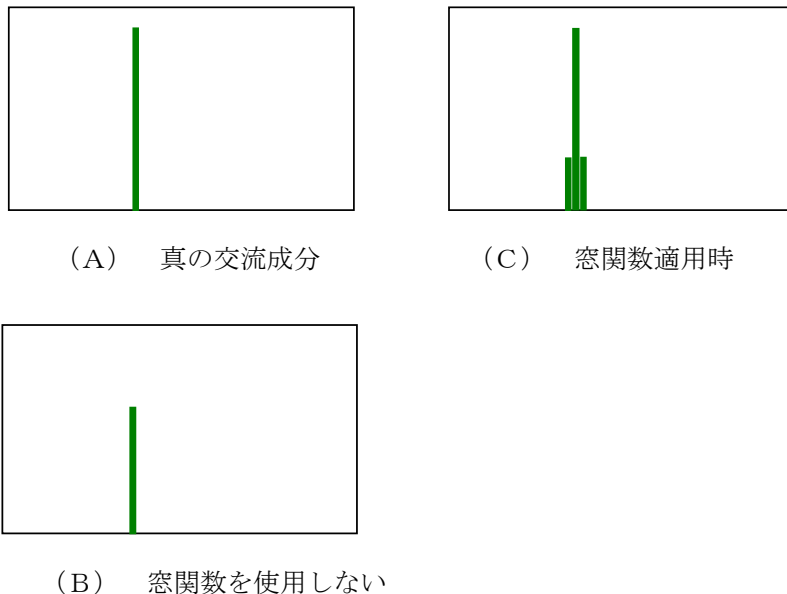


図 1-7-12 窓関数の影響

(※注 1-7-6)

フリッカ値算出手順は、

「JEITA規格」：「輝度計信号取得」→「積分器処理」→「FFTアナライザ処理」→「パワースペクトル演算」
 CA-210 液晶フリッカ測定プローブ：「輝度計信号取得」→「FFTアナライザ処理」→「積分器処理」→「パワースペクトル演算」

となっており、両方で「積分器処理」、「FFTアナライザ処理」の順序が入れ替わっています。

しかし、いずれの順序をとっても、「パワースペクトル演算」に入力されるデータは同じとなるため、両者は同じフリッカ値を得ることになります。言い換えると、等価な演算処理を行っていることになります。

参考文献

- VESA 305-4/305-5
- EIAJ ED-2522
- 南茂夫「科学計測のための波形データ処理」(CQ出版)
- 中村尚五「デジタルフーリエ変換」(東京電機大学出版局)

1-8 PDP(プラズマディスプレイ)の測定 (CA-100plus/ CA-210 標準プローブ)

1-8-1 PDPとは

PDPとはプラズマディスプレイの略で自発光型の表示デバイスです。

PDPの発光原理は、蛍光灯と同じです。蛍光灯は、蛍光管の中で放電し、管内の白色蛍光体に紫外線を当てることで、光を發します。PDPも、同じように気体放電によって紫外線が生成され、この紫外線がパネル内の赤(R)、緑(G)、青(B)の蛍光体に当たることで、發光します。

図1-8-1にプラズマパネルの基本構造を示します。プラズマのパネルは、2枚のガラス板を平行に重ねたもので、その間にはセルと呼ばれる小部屋が作られており、それぞれ、赤(R)、緑(G)、青(B)の蛍光体が塗られています。

セルには希ガス(キセノンが主)が封入されており、+と-の電極間に電圧を印加するとガス放電が起こり励起されたガスからは紫外線が発生します。この紫外線が、赤(R)、緑(G)、青(B)のセルに当たることで、蛍光体が光り、可視光線を發光します。

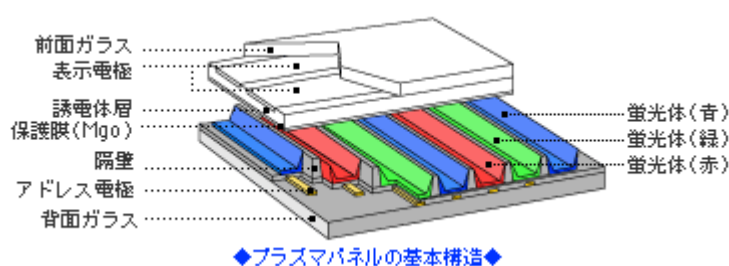


図 1-8-1 プラズマパネルの基本構造

他の表示パネルと同様にR, G, Bの集合体を最小単位として画素(ピクセル)と呼びます。

この画素が縦横に規則正しく並ぶことにより、パネル全体を構成しています。画素ピッチは1mm×1mm程度の大きさであり、40インチ前後のパネルではその解像度はVGAが主流であるが最近では高精細のXGAモデルも出てきています。

表 1-8-1 にTVの伝送の映像フォーマットの種類を示します。

添え字のi, pは走査方式を現しており、各々インターレス(飛び越し走査)、プログレッシブ(順次走査)に対応しています。現行では伝送フォーマットはインターレスであり、480iがNTSC、1080iがHDTVです。

PDPではインターレス方式をプログレッシブ方式に変換して表示しています。その際に、PDPの解像度に合わせて画像信号処理も行っており、これらをフォーマット変換と言います。

表 1-8-1 TVの伝送の映像フォーマット

	走査線数/有効走査線数	有効画素数 (横×縦)	アスペクト比	走査方式	フレーム f
1080i	1125/1080	1920×1080	16:9	飛び越し	29.97Hz
480p	525/480	720×480	16:9	順次	59.94Hz
480i	525/480	720×480	16:9,4:3	飛び越し	29.97Hz
720p	750/720	1280×720	16:9	順次	59.94Hz
*1080p	1125/1080	1920×1080	16:9	順次	59.94Hz

またPDPはPC入力に対して画像情報を圧縮・拡大して表示しています。(注 1-8-1)これを解像度変換と言います。

さらにPDPではTV信号に合わせており、(50/60Hz)、それ以外のPC入力も垂直周波数は固定されます。前述のフォーマット変換と合わせてマルチスキャンコンバータと言います。PDPではこの機能を内蔵しています。

次に階調表示方式について説明します。

PDPでは輝度を増加させるために、セルのメモリ機能を利用して線順次アドレス/全面同時發光を行っています。(注 1-8-2)

図 1-8-2 に階調表示方式の例を示します。この例ではTVの1垂直走査期間(=1TVフィールド)を8つのサブフィールドに分けています。ここで發光時間の長さを2のべき乗1:2:4:8:16:32:64:128にすれば、これらの組み合わせで25

6階調を現すことが出来ます。

つまりPDPでは発光時間のパルス幅を変えることにより階調を変えています。(注 1-8-3)

図の横軸は時間、縦軸は走査・放電維持電極の番号を示します。この番号はパネルの縦方向の画素数に対応しており、本例はVGAパネルの場合です。

各サブフィールドはどの画素を発光させるか指定選択するアドレス期間と、選択された画素のみ所定の輝度比に応じた

パルス幅で発光させる放電維持期間(サスティーン期間)で構成されています。

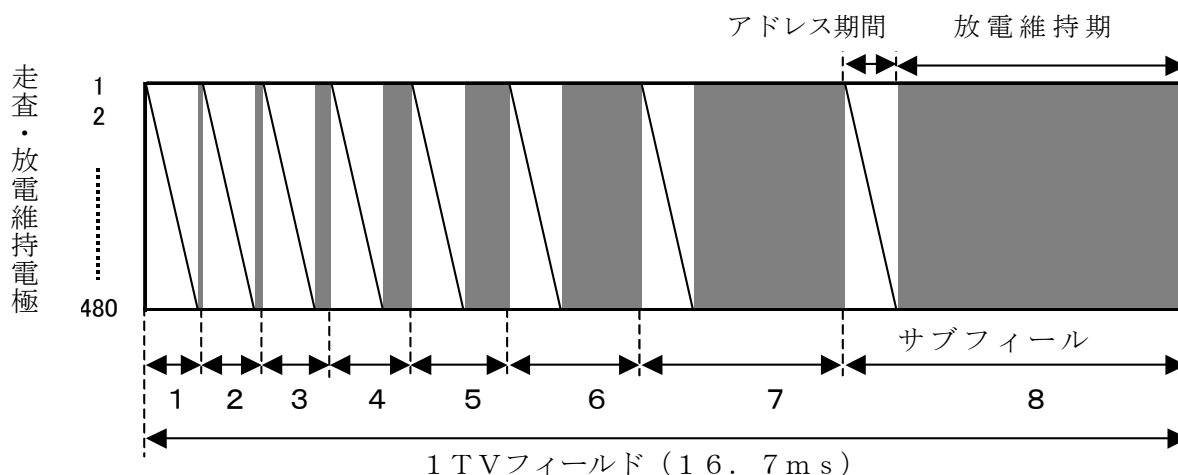


図 1-8-2 PDPの階調表示方式の例

このような階調表示方式をサブフィールド駆動法と言いますが、これにはいくつかの特有の画質障害があり、代表的な画質障害に動画擬似輪郭があります。(注 1-8-4)このような画質障害を軽減するため種々の駆動方式が存在し、実際は図 1-8-2のような単純なモデルではなく、今後も変わっていくと考えられます。

1-8-2 色彩輝度計との関係

パルス幅変調であるPDPをセンサで受光する場合、仮に受光処理回路の時定数を0とすると図 1-8-2 のような信号レベルが0と1の矩形波形になり、正確な測定ができません。(注 1-8-5) つまり、この場合ピーク値と平均値との差が格段に大きくなり、回路のダイナミックレンジを有効に使うことが出来ないためS/Nが悪化してしまいます。またピーク値が飽和してしまうため、絶対値誤差が生じます。

従い、CA-100Plusシリーズ/ CA-210 標準プローブ(注 1-8-6)では回路の時定数を適量にすることにより、適正なアナログ波形にしています。

1. 項でPDPの発光原理について気体放電による紫外線が蛍光体に当たることにより発光すると述べました。

しかしながら、現時点での PDP は、発光効率が低い為、エネルギーの大半が熱になってしまい、輝度最大の状態では、40°C以上になります。この温度上昇の影響を除去するために、CA-100Plusシリーズ/ CA-210 標準プローブでは測定プローブ内の温度センサにより正確な温度補正を行っており、管面温度の影響を受けずに測定ができます。

(注 1-8-1)

例)画素数VGAの PDP では SVGA 信号以上は圧縮して表示しています。

(注 1-8-2)

セルは一旦点灯信号を与えると信号を除いた後も点灯状態を続け、消灯信号を与えると信号を除いた後も消灯状態を続ける、いわゆる双安定状態を実現することができます。

(注 1-8-3)

CRTでは画素毎の輝度を電子ビームの強さにより変えることができますが(アナログ量制御)、PDPでは紫外線の飽和により、例えば電流による輝度の制御ができないため、発光時間の長さにより輝度を変えます。(デジタル量制御)

(注 1-8-4)

動画の場合、人間の目がある画面を追従した場合にその画面前後のサブフィールドの発光部分と非発光部分とを融合して認識してしまうため、本来存在しない色の帯が見えてしまいます。

(注 1-8-5)

実際はサステーン期間は数十kHzの周波数でON/OFFされており、周波数及びデューティを変えることにより階調によってピーク輝度を変える等制御されています。この結果、受光波形はもつと複雑になります。

(注 1-8-6)

測定輝度範囲の異なる 2 種類のプローブ (標準、高輝度) があります。

※標準プローブ、高輝度プローブの仕様

項目		CA-100Plus (標準プローブ)	CA-100PlusH (高輝度プローブ)
輝度	測定範囲	0.05~1000 cd/m2	0.05~2000 cd/m2
	確度	0.10~1000 cd/m2 ±2%±1digit 校正CRT (D65)	0.05~2000 cd/m2 ±2%±1digit 校正CRT (D65)
	繰返性	0.10~1000 cd/m2 0.2%+1digit (2σ)	0.05~2000 cd/m2 0.2%+1digit (2σ)
色度	測定範囲	0.05~1000 cd/m2	0.05~2000 cd/m2
	確度	0.05~0.19 cd/m2 ±0.006 校正CRT (D65) 0.20~0.49 cd/m2 ±0.004 校正CRT (D65) 0.50~1000 cd/m2 ±0.003 校正CRT (D65) 校正CRT (D65, 40cd/m2) ±0.002 白色 ±0.004 単色	0.05~0.09 cd/m2 ±0.008校正CRT (D65) 0.10~0.39 cd/m2 ±0.006 校正CRT (D65) 0.40~0.99 cd/m2 ±0.004 校正CRT (D65) 1.00~2000 cd/m2 ±0.003 校正CRT (D65) 校正CRT (D65, 40cd/m2) ±0.002 白色 ±0.004 単色
	繰返性	CRT D65 0.05~0.19 cd/m2 0.006 (2σ) 0.20~0.49 cd/m2 0.002 (2σ) 0.50~1000 cd/m2 0.001 (2σ)	CRT D65 0.05~0.09 cd/m2 0.009 (2σ) 0.10~0.39 cd/m2 0.006 (2σ) 0.40~0.99 cd/m2 0.002 (2σ) 1.00~2000 cd/m2 0.001 (2σ)

CA-210/100plus 技術資料

2. 確度定格・繰り返し性の定義

2-1 「輝度・色度」確度定格・繰り返し性の定義

CA-210の確度仕様を表 2-1-1 に示します。
この仕様に対する定義内容を以下に説明します。

表 2-1-1 CA-210の確度

輝度	測定範囲	0.10~1000 cd/m ²		
	確度	0.10~1000 cd/m ²	±2%±1digit 校正液晶 (6500K, 9300K)	
	繰返し性	0.10~0.99 cd/m ²	0.2%±1digit (2σ)	
		1.00~1000 cd/m ²	0.1%±1digit (2σ)	
色度	測定範囲	0.10~1000 cd/m ²		
	確度	校正液晶 (6500K, 9300K)		
		0.10~4.99 cd/m ²	±0.005	
		5.00~19.99 cd/m ²	±0.004	
		20.00~1000 cd/m ²	±0.003	
		160 cd/m ²	±0.002 (単色 ±0.004)	
	繰返し性	液晶 (6500K, 9300K)		
	0.10~0.19 cd/m ²	0.010	(2σ)	
	0.20~0.49 cd/m ²	0.005	(2σ)	
	0.50~0.99 cd/m ²	0.002	(2σ)	
	1.00~1000 cd/m ²	0.001	(2σ)	

2-1-1-1 確度

2-1-1-1 確度の定義

CA-210は国家標準へ遡ることが可能なトレーサビリティ体系を有しています(「[2-1-4 トレーサビリティ](#)」参照)。
確度は、CA-210トレーサビリティ体系に基づき値付けされたCA-210基準測定器からの差で定義しています。但し、確度を定義するにあたっては、繰返し性誤差は含まないものとし、複数回測定した場合の平均値を用いています。

2-1-1-2 確度保証輝度範囲

CA-210は0.1 cd/m²までの輝度について精度保証しています(※注 2-1-1)。
各輝度範囲での精度保証光源は、輝度に変化しても分光放射輝度、配光特性が変化しない光源としています。

2-1-1-3 確度保証光源 さらに詳しく

刺激値直読式色彩計では、ある光源(ディスプレイ)を測定した場合に絶対値誤差がない場合でも、異なる光源を測定した場合には絶対値誤差が生じ得ます。また、同じ光源であっても、測定色が異なれば誤差が生じ得ます。(「[1-2-3刺激値直読式色彩計の絶対値誤差](#)」参照)

従い、刺激値直読式色彩計では、確度保証するための光源をカタログに記載しています。CA-210はコニカミノルタが基準とする液晶モニタで確度保証しています。

確度保証光源と被測定光源の分光放射輝度の差が小さい場合、絶対値誤差は小さいものとなります。また、液晶モニタは異なる機種間でもよく似た分光特性をもっています(図 2-1-1)。従い、CA-210は液晶モニタ測定時に、絶対値精度の高い(絶対値誤差の小さい)測定が可能です。

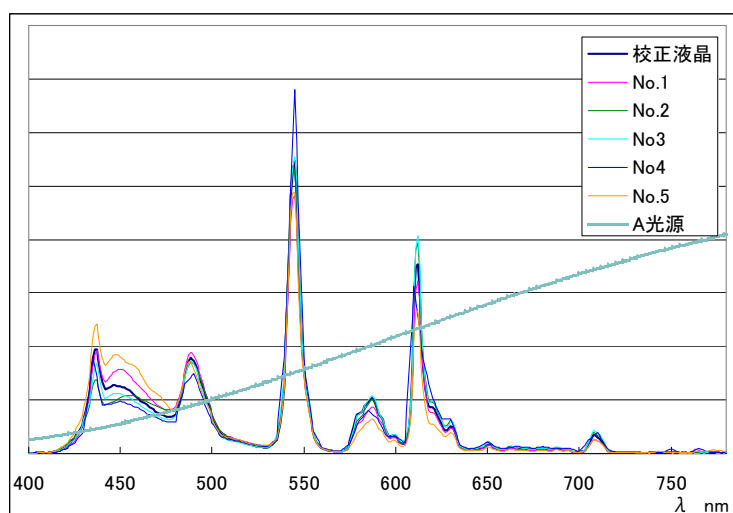


図 2-1-1 液晶とA光源の分光放射輝度

一般的な刺激値直読式色彩計は、A光源を確度保証光源としています。他方、A光源と液晶モニタの分光特性は大きく異なるため(図 2-1-1)、A光源で確度保証した色彩計で液晶モニタを測定した場合は、絶対値誤差、器差が大きくなる傾向が見られます(※注 2-1-2)。

なお、ここで述べた内容は、輝度、色度の両方について言えることです。

2-1-2 繰返し性

2-1-2-1 繰返し性の定義

CA-210の繰返し性は、同一条件下での複数回測定値の標準偏差の2倍(=2σ)で定義しています。

2-1-2-2 繰返し性の保証光源 さらに詳しく

表 2-1-2 単色繰返し性の実機評価結果

CA-210の繰返し性の保証する測定光源は液晶モニタとしています。

測定色度については、6500K、9300Kとしています。これ以外の色を測定した場合は、繰返し性が異なります。白色(6500K)と、それ以外の色である単色に対する繰返し性の実測結果を表2に示します(評価台数3台)。

この結果から、白色(W)に対してR、Gの繰返し性がやや悪化、Bは向上していることが判ります。(※注 2-1-3)

測定色	本体No	繰返し性 (2σ)		測定時の色度、輝度		
		x	y	x	y	Lv
R	No1	0.0003	0.0003	0.6318	0.3274	1.09
	No2	0.0005	0.0005	0.6322	0.3269	1.09
	No3	0.0003	0.0002	0.6326	0.3257	1.09
G	No1	0.0002	0.0005	0.2868	0.5861	1.38
	No2	0.0002	0.0004	0.2869	0.5860	1.38
	No3	0.0002	0.0004	0.2871	0.5853	1.38
B	No1	0.0001	0.0000	0.1464	0.0494	1.01
	No2	0.0001	0.0000	0.1465	0.0496	1.02
	No3	0.0001	0.0001	0.1461	0.0493	1.02
W	No1	0.0001	0.0002	0.3089	0.3243	1.38
	No2	0.0002	0.0002	0.3092	0.3241	1.38
	No3	0.0002	0.0002	0.3096	0.3239	1.39

2-1-3 測定精度とは

確度には、繰返し性誤差は含まないことを述べました。つまり、確度と繰返し性は独立して定義しています。

従い、1回測定した場合、測定値は真値に対し、図 2-1-2 に示すように「確度+繰返し」精度内にあることになります。

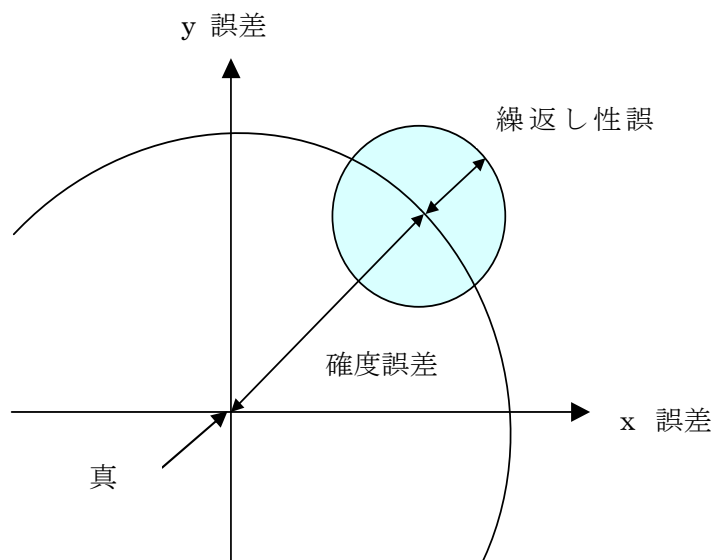


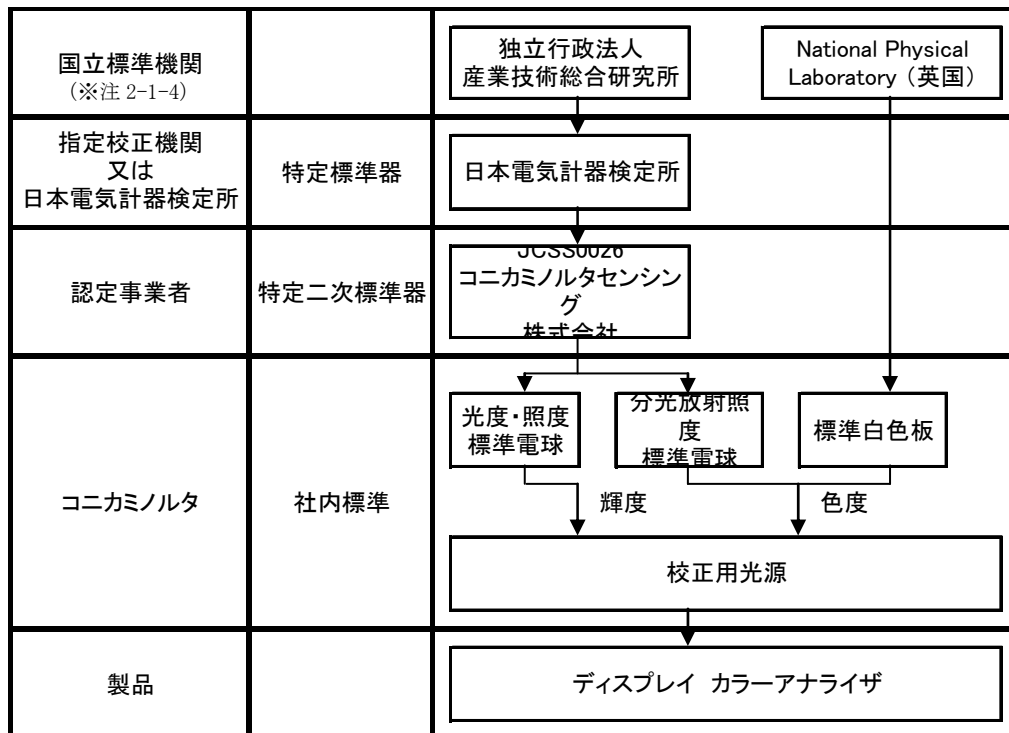
図 2-1-2 測定精度

2-1-4 トレーサビリティ

CA-210は国家標準へ遡ることが可能なトレーサビリティ体系を有しています。

図 2-1-3 にトレーサビリティ体系図を示します。

今まで、測定器がトレーサビリティ体系を有するか否かは、一部のディスプレイ、光源の生産現場でのみ留意されてきました。他方、最近、世界的に「測定器には厳密なトレーサビリティ体系を有すること」が求められる傾向があります。つまり、測定器がトレーサビリティ体系を有することの重要性が高まっています。(※注 2-1-5)



校正周期： 光度・照度標準電球（1年又は累計10時間の使用毎）
 分光放射照度標準電球（1年又は累計10時間の使用毎）、標準白色板（5年）
 校正用光源（製品校正毎）

図 2-1-3 トレーサビリティ体系図

(※注 2-1-1)

液晶モニタのコントラスト測定時の最低輝度は一般に0.5cd/m²程度です。これに対して、十分低い輝度まで精度保証しているため、液晶モニタ検査、評価に対して必要十分な性能を有していると言えます。

(※注 2-1-2)

液晶モニタ(コニカミノルタ基準液晶モニタ)で校正した場合と、A光源で校正した場合のそれぞれにおいて、5種類の液晶モニタを測定した場合に発生する絶対値誤差を評価した数値実験結果を表 2-1-3a、表 2-1-3b に示します。ここで、数値実験は実測により得た各光源の分光放射輝度に基づいて行っています。液晶モニタの色度は白色(6500K~9300K)としています。

この結果から、絶対値誤差が、液晶モニタで校正した場合にはxyで約0.002以下と小さいのに対し、A光源で測定した場合では0.009と大きいことが判ります。

表 2-1-3a A光源で校正した場合の他5種液晶測定誤差

	Δx	Δy
A光源	0.0000	0.0000
液晶 A	-0.0085	-0.0027
液晶 B	-0.0087	-0.0048
液晶 C	-0.0090	-0.0022
液晶 D	-0.0084	-0.0017
液晶 E	-0.0081	-0.0042

表 2-1-3b 液晶で校正した場合の他5種液晶測定誤差

	Δx	Δy
コニカミノルタ基準液晶	0.0000	0.0000
液晶 A	-0.0004	0.0016
液晶 B	-0.0004	-0.0010
液晶 C	-0.0005	0.0015
液晶 D	-0.0001	0.0017
液晶 E	-0.0016	0.0012

(※注 2-1-3)

繰返し性を決める主要因は、X、Y、Zセンサー出力の大きさです。各出力が大きい方が、繰返し性は向上します。

表2の結果では、各色測定においてLv値(=Yセンサー出力)をほぼ等しくしています。従い、残りのX、Zセンサー出力の違いが繰返し性の違いに表れます。この各色測定時のセンサー出力比を表4に示します。

R色測定の場合、W(白色)と比較して、Xセンサー出力は大きくなりますが、Zセンサー出力が極端に小さくなるため、繰返し性がWに比べ低下します。

G色測定の場合、W(白色)と比較して、X、Zセンサー出力が共に小さくなるため、繰返し性がWに比べ低下します。

B色測定の場合、YセンサのB色に対する感度が低いために、Lvを1cd/m²とした時にはX、Zセンサーの出力は共にW測定時より大きくなり、繰返し性はWに比べ向上します。

表 2-1-4 各色測定時のセンサー出力比

	センサー出力比		
	X	Y	Z
R	1.9	1.0	0.1
G	0.5	1.0	0.2
B	2.9	1.0	16.1
W	1.0	1.0	1.1

(※注 2-1-4)

- 国立標準機関として
- 産業技術総合研究所(日本)
- National Physical Laboratory(英国)
- National Institute of Standards and Technology(米国)
- 等があります。

(※注 2-1-5)

例えば、品質マネジメントシステムを規定した規格 ISO9001:2000(JIS Q 9001:2000)には、その7.6項において「測定値の正当性が保証されなければならない場合には、測定機器に関し、次の事項を満たすこと。

a)定められた間隔又は使用前に、国際または国家計量標準にトレース可能な計量標準に照らして校正又は検証する。そのような標準が存在しない場合には、校正又は検証に用いた基準を記録する。

(以下省略)

と記載されています。

つまり、国家計量標準が存在する計測器を使用する場合、その測定器が国家計量標準にトレースされていることが義務付けられています。

2-2 「フリッカ」確度定格・繰り返し性の定義 [さらに詳しく](#)

CA-210 液晶フリッカ測定プローブの確度仕様を表1に示します。
この仕様に対する定義内容を以下に説明します。

表 2-2-1 CA-210 液晶フリッカ測定プローブの確度仕様

フリッカ・コントラスト方式	測定範囲	5 cd/m ² 以上 (小径プローブは 15 cd/m ² 以上)
	表示範囲	0.0~100.0%
	確度	±1% (30Hz AC/DC 10%正弦波) (※注 2-2-1) ±2% (60Hz AC/DC 10%正弦波) (※注 2-2-1)
	繰返し性	1% (2σ) (AC/DC 10%正弦波) (※注 2-2-1)
フリッカ・JEITA方式	測定範囲	5 cd/m ² 以上 (小径プローブは 15 cd/m ² 以上)
	確度	±0.5% dB (30Hz AC/DC 10%正弦波)
	繰返し性	0.3 dB (2σ) (30Hz AC/DC 10%正弦波)

2-2-1 確度の定義

フリッカ値は直流成分と交流成分の比で表され、単位は無次元となります。従い、フリッカ値にはトレーサビリティ体系はありません。

仮に、フリッカの絶対値が値付けられた光源が存在すれば、その光源を測定することで絶対値誤差(確度)を定義することができます。しかし、現実にはそのような光源は存在していません。そこで、CA-210 液晶フリッカ測定プローブでは、以下の考え方に基づいて確度を定義しています。

2-2-1-1 基本的な考え方

CA-210は、本体部、プローブ部から構成されています(図 2-2-1)。
その各構成部ごとに確度を算定し、その合計で確度を定義します。

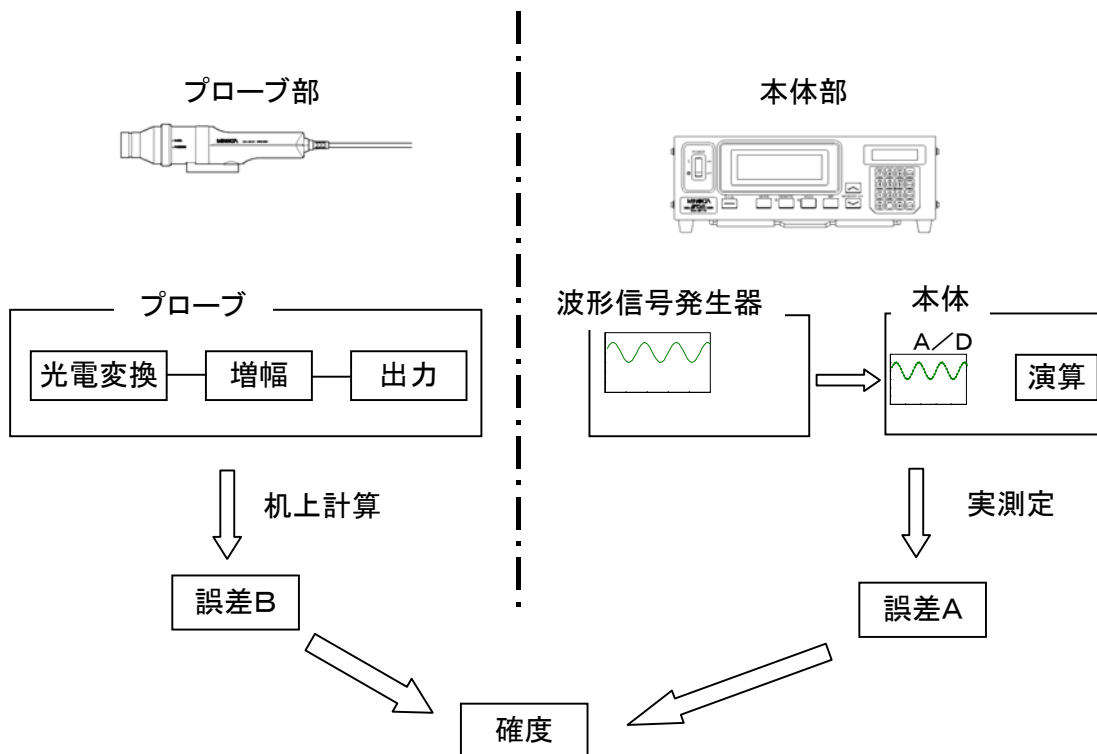


図 2-2-1 CA-210の構成と確度

2-2-1-2 本体部について

以下の実測に基づいて誤差を算出します。

プローブ部から入力されるセンサー信号の代わりに、波形信号発生器から信号を本体に入力します。

波形信号発生器入力信号から予測されるフリッカ値(※注 2-2-2)と出力されるフリッカ測定値の差を絶対値誤差(誤差A)とします。この時の波形信号発生器からの入力信号は、

交流成分/直流成分比	10%
交流周波数	30Hz、60Hz(JEITA方式は30Hzのみ)
信号波形	sinカーブ

としています。また、直流成分信号レベルは輝度5cd/m²相当以上としています。

2-2-1-3 プローブ部について

以下の、机上計算結果に基づいて誤差を算出します。

この構成部は、センサーを含む光電変換回路、増幅回路、出力回路(インターフェース回路)から構成されています。これら構成要素ごとに誤差因子を抽出し、その合計をもって絶対値誤差とします。

特に、増幅回路部はローパスフィルターを形成しており、素子バラツキが交流成分増幅比に影響を与えることで誤差が発生します。プローブ部の誤差は主に、この部分に起因しています。

設計上予測される素子バラツキから机上計算により算出される誤差の最大値を、絶対値誤差(誤差B)と定義しています。

2-2-1-4 CA-210の確度

CA-210の確度は、「誤差A+誤差B」で定義しています。

確度を定義するにあたっては、繰返し性誤差は含まないものとし、複数回測定した場合の平均値を用いています。コントラスト方式、JEITA方式とも同じ考え方で確度を定義しています。

2-2-2 繰返し性の定義

CA-210の繰返し性は、以下の①②の繰返し性に差がないことを、実機検証確認した上で、①に対して定義しています。繰返し性誤差は、その条件下での複数回測定値の標準偏差の2倍(=2 σ)で定義しています。

①プローブ部から入力されるセンサー信号の代わりに、波形信号発生器から信号を本体に入力した時の測定値
 交流成分/直流成分比 10%
 交流周波数 20~75Hz(JEITA方式は30Hzのみ)
 信号波形 sinカーブ

②以下のように調整した液晶を測定した時の測定値
 交流成分/直流成分比 10%(CA-210コントラスト・フリッカ値)
 垂直同期周波数 60Hz(フリッカ交流成分周波数30Hz)

2-2-3 測定精度

確度には、繰返し性誤差は含まないことを述べました。つまり、確度と繰返し性は独立して定義されています。

従い、1回測定した場合、測定値は真値に対し、図2-2-2に示すように「確度+繰返し」精度内にあることとなります。

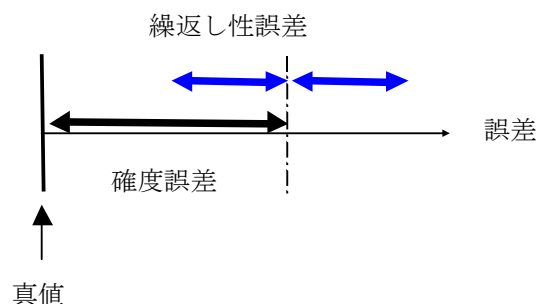


図 2-2-2 測定精度

(※注 2-2-1)

フリッカ・コントラスト方式の“確度”と“繰返し性”の単位である“%”は、交流成分／直流成分比のことです。
 例えば、フリッカ値10%において、確度±1%とは、確度が9%～11%以内であることを意味します。
 同様に、フリッカ値10%において、繰返し性1%(2σ)とは、繰返し測定した場合に、測定値が9%～11%の範囲でバラツキ(2σ)をもつことを意味します。

(※注 2-2-2)

波形信号発生器入力信号から予測されるフリッカ値を算出する、交流成分、直流成分は波形信号発生器の設定値を用いています。例えば、

交流成分／直流成分比	10%、
交流周波数	30Hz、60Hz (JEITA方式は30Hzのみ)
信号波形	sinカーブ

の信号波形に対して、コントラスト方式、JEITA方式のフリッカ値(真値)は以下のようになります。

・コントラスト方式では、
 交流成分／直流成分比がそのままフリッカ値になるので、
 フリッカ値は 10[%]

・JEITA方式では、この例の場合
 直流成分(0Hz)のFFT出力は400 (=FFT(0Hz))
 交流成分(30Hz)のFFT出力は10 (=FFT(30Hz))
 直流成分の積分器の増幅比(減衰比)は 100% (=weight(0Hz))
 交流成分30Hzの積分器の増幅比(減衰比)は 70.8% (=weight(30Hz))
 となります。

以上より、JEITA方式定義式に基づき、フリッカ値は

$$20 \times \log \frac{(\sqrt{2} \times \text{weight}(30\text{Hz}) \times \text{FFT}(30\text{Hz}))}{\text{weight}(0\text{Hz}) \times \text{FFT}(0\text{Hz})}$$

$$= -32.03 \text{ [dB]}$$

となります。(「[1-7 フリッカ測定](#)」参照)

CA-210/100plus 技術資料

3. 実測データ

3-1 CA-210, CA-110, CS-1000 の液晶測定値差 [さらに詳しく](#)

3-1-1 白色測定(高輝度)

高輝度(100cd/m²以上)、白色表示した液晶を各測定器(CA-210, CA-110, CS-1000)で測定した結果を表 3-1-1 に示します。ここで、白色とは6500K付近の色のことです。測定した液晶は4機種です。測定に使用したCA-210は液晶ブリッカ測定プローブで、以降 CA-210 と記載します。

分光器CS-1000の測定値を真値と考えると、CA-210, CA-110のCS-1000との差は、CA-210, CA-110の絶対値誤差と考えることができます。

CA-210とCS-1000の差は、色度で0.002以下、輝度で3%以下となっています。つまり、CA-210は異なる種類の液晶を測定した場合にも絶対値精度が高いと言えます。

これに対し、CA-110とCS-1000の差は、色度で0.007以下、輝度で4%以下となっています。特に、色度の絶対値誤差が、CA-210と比較して悪くなります。

表 3-1-1 高輝度、白色表示液晶を測定した結果

液晶の種類	CS-1000			(CA-210) - (CS-1000)			(CA-110) - (CS-1000)			(CA-110) - (CA-210)		
	x	y	Lv	Δx	Δy	ΔLv	Δx	Δy	ΔLv	Δx	Δy	ΔLv
液晶F	0.313	0.331	156.9	0.001	0.000	0.8%	-0.007	0.001	-3.2%	-0.008	0.001	-3.9%
液晶G	0.313	0.333	158.8	0.000	-0.002	2.7%	-0.007	-0.002	0.1%	-0.007	0.001	-2.6%
液晶H	0.312	0.331	103.9	0.002	-0.002	1.7%	-0.002	-0.004	-2.8%	-0.003	-0.002	-4.5%
液晶I	0.309	0.314	236.2	0.002	0.000	2.1%	-0.005	0.003	0.4%	-0.007	0.003	-1.7%

(CA-110は近接条件で測定しています。)

これらの差が発生する要因としては

①校正光源の差

②測定器の光学系の差(開口角は、CS-1000が1° , CA-210 が5° , CA-110が10°)があります。(※注 3-1-1、※注 3-1-2)

①について、刺激値直読式色彩計(※注 3-1-3)では、その分光応答度がCIE等色関数からわずかに異なるために、校正光源以外の光源を測定した場合、誤差が発生します。CA-210はLCD で校正しているため、異なる種類の液晶を測定した場合、この誤差が小さいのに対し、CA-110はC光源で校正しているため、誤差が大きくなります。[\(「1-2 色彩計について」参照\)](#)(※注 3-1-4)

②について、液晶は、機種によって配光特性が異なります。これが原因となり、開口角が異なる測定器間では輝度測定値に差が生じます。これを図 3-1-1 を用いて説明します。

ここでは2種類の液晶(液晶P、液晶Qと呼びます)の配光特性をそれぞれ示しています。また、ここで液晶Pと液晶Qの紫色部分の面積は等しいものとします。

図 3-1-1 に示すように、液晶の種類が異なると、配光特性は、“広がり”、“向き”が異なります。

この液晶の発光面に対し測定器を正対させて測定する場合は考えます。開口角が1° の場合、測定器は紫色で示した光束を受光することになります。また開口角が10° の場合は、紫色、水色で示した両方の光束を受光することになります。

開口角が1° の場合、液晶Pと液晶Qの紫色部分の面積は等しいから、輝度測定値は等しくなります(輝度が100cd/m²であるとして)。ところが、紫色、水色部分の和の面積は液晶Pと液晶Qでわずかに異なります(この図では液晶Pの方が小さい)。従い、開口角が10° の場合、液晶Pで輝度が100cd/m²となるように校正しても、液晶Qを測定した場合は、100cd/m²とはなりません。つまり、開口角1° の場合と測定値差が生じます。

これが、測定器の光学系の差に起因して測定値差が生じるメカニズムです。今回評価した液晶の配光特性を実測し、上記測定差を数値計算すると、開口角1° に対し、開口角5° の場合は、最大で約1%、開口角10° の場合は約2%となります。

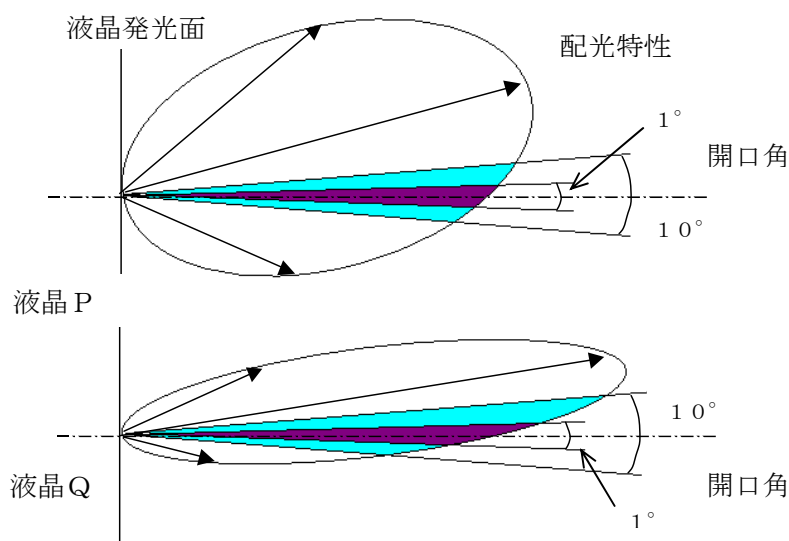


図 3-1-1 液晶の配光特性と開口角

3-1-2 白色測定(低輝度)

輝度(20cd/m²付近)、白色表示した液晶を各測定器(CA-210, CA-110, CS-1000)で測定した結果を表 3-1-2 に示します。ここで、白色とは6500K付近の色のことです。測定した液晶は3機種です。

低輝度では、CA-210とCS-1000の差は、色度で0.003以下、輝度で6%以下となります。つまり、CA-210は、低輝度においても色度の精度が高いと言えます。他方、輝度については高輝度測定の場合と比較して、CS-1000との差が大きくなっています。

これは、受光系の差が大きいことによります。このメカニズムは「1. 白色測定(高輝度)」で説明した内容と同じです。ここでは、さらに具体例を示して、説明します。

図 3-1-2 に低輝度でCS-1000との差が大きい液晶Gの低輝度、高輝度での配光特性を示します。この配光特性は発光面に垂直な発光成分が等しくなるように正規化しています。

開口角5° の場合、高輝度では図の“黄色+緑色+赤色”の部分に相当する光を受光し、低輝度では“黄色+緑色+青色”の部分に相当する光を受光します。他方、開口角が非常に小さい場合(例えば1°)、高輝度、低輝度とも“黄色”の部分に相当する光を受光します。

図 3-1-2 より、“緑色+青色”の部分の面積は、“緑色+赤色”の部分の面積より大きいので、「開口角1° 受光量」に対する「開口角5° 受光量」の比は高輝度より低輝度の方が大きくなります。

従い、高輝度で開口角1° と開口角5° の測定器の測定値が一致している時に、低輝度で測定した場合、開口角5° の測定器の測定値は開口角1° の測定器の測定値より大きくなります。

図 3-1-2 に示した液晶Gの場合、図の配光特性から計算した差は約3%になります。

表 3-1-2 低輝度、白色表示液晶を測定した結果

液晶の種類	CS-1000			(CA-210) - (CS-1000)			(CA-110) - (CS-1000)			(CA-110) - (CA-210)		
	x	y	Lv	Δx	Δy	ΔLv	Δx	Δy	ΔLv	Δx	Δy	ΔLv
液晶F	0.313	0.328	22.4	0.000	0.000	1.7%	-0.008	0.005	1.4%	-0.008	0.005	-0.3%
液晶G	0.312	0.327	22.0	0.001	-0.002	5.8%	-0.008	0.004	6.0%	-0.009	0.004	0.2%
液晶H	0.310	0.331	17.5	0.003	-0.001	2.1%	-0.001	-0.002	-1.3%	-0.004	-0.002	-3.3%

液晶発光面

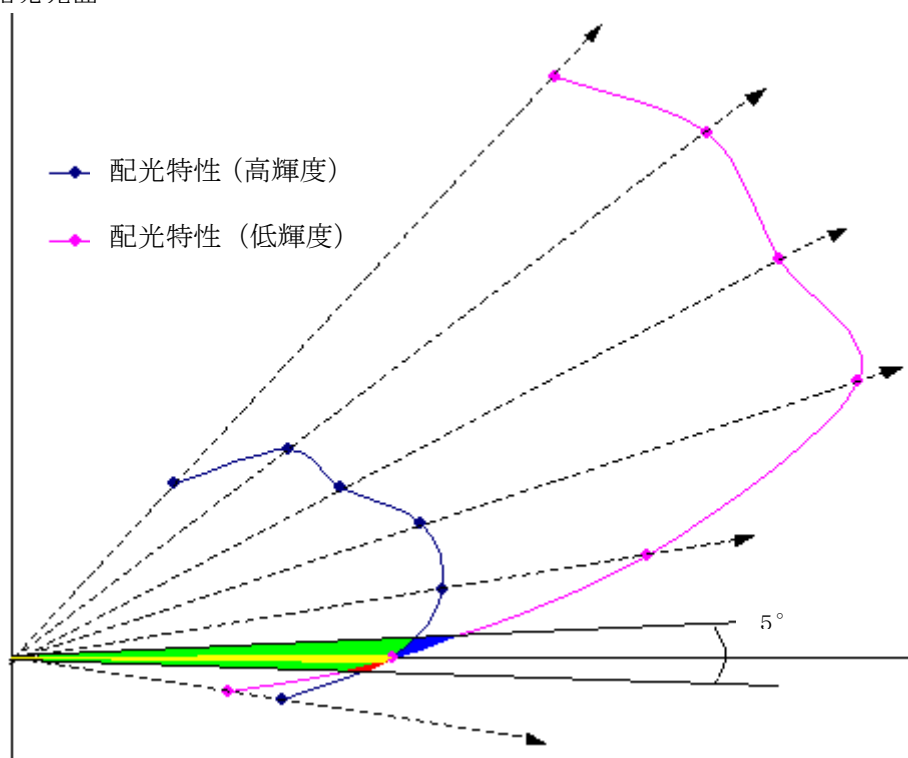


図 3-1-2 液晶の輝度-配光特性

3-1-3 単色測定

単色表示した液晶を各測定器(CA-210, CA-110, CS-1000)で測定した結果を表 3-1-3 に示します。測定した液晶は3機種です。また、この結果の誤差最大値(Δx または Δy のうちの大きい方)を抽出した結果を表 3-1-4 に示します。

表 3-1-3、表 3-1-4 の結果から、CA-210, CA-110の絶対値誤差を比較すると、CA-210の絶対値誤差が小さくなっていることがわかります。CA-210はCA-110に対して、誤差がG色に対して1/2以下に、B色に対して1/3以下に、Rに対しては1/10以下に縮小されています。

この主な理由は、CA-110が「白色1点校正」を採用しているのに対し、CA-210が「マトリクス校正」を採用していることにあります。

「液晶F」はCA-210の校正基準液晶です。この液晶では、CA-210の色度絶対値誤差が0.002以下になっています。この結果から、ある液晶に対してマトリクス校正を行えば、その液晶に対して、単色領域においても絶対値誤差の非常に小さい測定が可能であることがわかります(※注 3-1-5)。

液晶は、種類が異なっても、分光放射輝度の差は小さい傾向があります。従い、ある液晶に対して、マトリクス校正を行えば、他機種の液晶測定においても、従来の1点校正された測定器で測定した場合と比較すると、絶対値精度の高い測定が可能です。表 3-1-3、表 3-1-4 の結果がそのことを示しています。

表 3-1-3 単色表示液晶を測定した結果

液晶の種類	色	CS1000		(CA-210) - (CS-1000)		(CA-110) - (CS-1000)		(CA-110) - (CA-210)	
		x	y	Δx	Δy	Δx	Δy	Δx	Δy
液晶 F	R	0.635	0.331	-0.001	0.001	-0.017	0.023	-0.016	0.021
	G	0.292	0.591	0.002	-0.001	-0.019	-0.015	-0.021	-0.014
	B	0.146	0.054	0.001	0.001	-0.005	-0.011	-0.006	-0.012
液晶 G	R	0.627	0.348	-0.002	0.002	-0.021	0.024	-0.019	0.021
	G	0.289	0.594	0.004	-0.011	-0.017	-0.023	-0.021	-0.012
	B	0.147	0.086	0.002	0.001	-0.004	-0.011	-0.006	-0.013
液晶 H	R	0.595	0.346	-0.002	0.000	-0.009	0.010	-0.007	0.010
	G	0.301	0.561	0.008	0.003	-0.010	-0.007	-0.018	-0.010
	B	0.150	0.134	0.004	0.002	-0.002	-0.014	-0.006	-0.016

表 3-1-4 単色表示液晶を測定した最大色度誤差 Δx , Δy

色	(CA-210) - (CS-1000)	(CA-110) - (CS-1000)
R	0.002	0.024
G	0.011	0.023
B	0.004	0.014

3-1-4 中間色測定

中間色、及び単色表示した液晶を各測定器(CA-210, CA-110, CS-1000)で測定した結果を表 3-1-5 に示します。

表 3-1-5 の結果から、CA-210の中間色、及び単色における測定値を比較すると、中間色測定時の絶対値誤差の方が単色測定時の絶対値誤差より小さいことがわかります。つまり、マトリクス校正点内の色度空間内では誤差は単色誤差以内となります。

CA-110も、同様に中間色測定時の絶対値誤差の方が単色測定時の絶対値誤差より小さくなりますが、その誤差はCA-210と比較すると大きくなります。(「[3-1-3 単色測定](#)」参照)

上記結果は図 3-1-3a～図 3-1-3d から理解することができます。CA-110は白色1点校正を行っており、この結果、デ

スプレイの表色色度空間は図 3-1-3c の三角形のエリアから図 3-1-3d に線形変換されます。つまり、真値と測定値の差は線形関係であるため、図の校正点が一致する場合、三角形の頂点で誤差が最大となり、中間色では、誤差が単色(三角の各頂点)の場合より小さくなります。

マトリクス校正では、R, G, Bによる校正後は、ディスプレイの表色色度空間は図 3-1-3a の三角形のエリアから図 3-1-3b に線形変換され、測定値はディスプレイの表色色度空間内全てで真値となります。ところが、CA シリーズでは、液晶の加法混色からのズレを考慮してマトリクス校正として、この「R, G, Bマトリクス校正係数に白色1点校正を加えた」方法を採用しています(「[1-4マトリクス校正](#)」参照)。この結果、誤差は白色1点校正の場合と同様に、三角形の頂点で誤差が最大となり、中間色では、誤差が単色(三角の各頂点)の場合より小さくなります。

表 3-1-5 中間色、及び単色表示液晶を測定した結果

色	CS1000		(CA-210) - (CS-1000)		(CA-110) - (CS-1000)		(CA-110) - (CA-210)	
	0.3132	0.3307	0.0005	0.0002	-0.0072	0.0013	-0.0077	0.0011
W	0.3132	0.3307	0.0005	0.0002	-0.0072	0.0013	-0.0077	0.0011
R	0.6354	0.3305	-0.0015	0.0014	-0.0174	0.0225	-0.0159	0.0211
G	0.2919	0.5911	0.0017	-0.0012	-0.0189	-0.0151	-0.0206	-0.0139
B	0.1460	0.0540	0.0006	0.0008	-0.0050	-0.0110	-0.0056	-0.0118
Ye (R+G)	0.4203	0.4913	-0.0001	-0.0009	-0.0153	-0.0043	-0.0152	-0.0034
Cy (G+B)	0.2191	0.3369	0.0015	-0.0003	-0.0111	-0.0049	-0.0126	-0.0046
Mg (B+R)	0.2885	0.1342	0.0000	0.0010	0.0025	0.0058	0.0025	0.0048

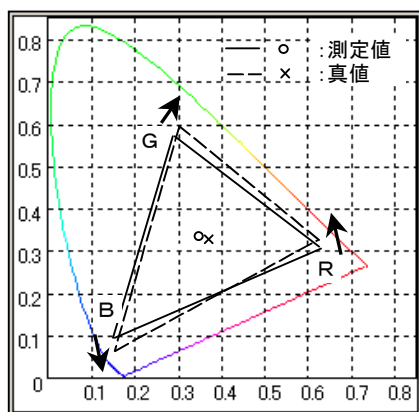


図 3-1-3a R, G, Bマトリクス校正前の測定値と真値

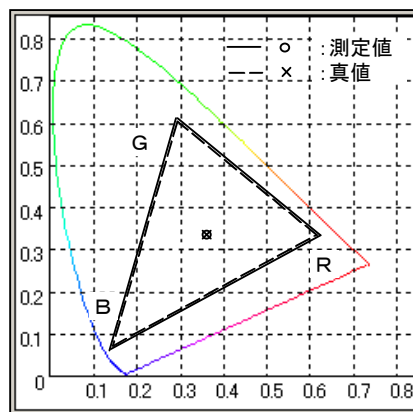


図 3-1-3b R, G, Bマトリクス校正後の測定値と真値

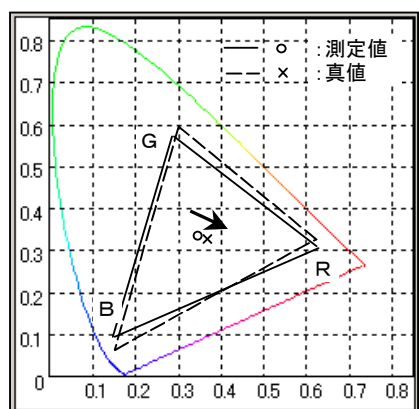


図 3-1-3c 白色1点校正前の測定値と真値

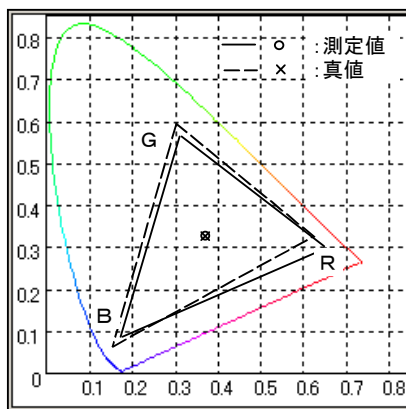
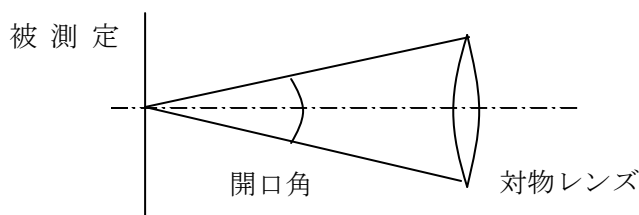


図 3-1-3d 白色1点校正後の測定値と真値

(※注 3-1-1)

開口角とは、被測定物から出た光で測定器に取り込まれる光線の範囲を表した角度のことです。



(※注 3-1-2)

CS-1000は測定距離約1.2m、CA-110は近接条件で測定する場合の開口角です。

(※注 3-1-3)

刺激値直読式色彩計とは、CIE1931で規定された等色関数に近似された分光応答度をもった3つのセンサの出力を用いて色度、輝度を求める色彩計のことです。

(※注 3-1-4)

参考として、刺激値直読式色彩計の分光応答度、C光源、複数種類液晶の分光放射輝度のデータから、数値実験により最大誤差を算出した結果を表 3-1-6 に示します。この差は、校正光源が液晶であるかC光源であるかに起因するものであると言えます。今回の実測結果(表 3-1-1)と数値実験結果(表 3-1-6)は近い値を示しています。

表 3-1-6 刺激値直読式色彩計における数値実験結果

校正光源： C 光源			校正光源： コニカミノルタ基準液晶		
	Δx	Δy		Δx	Δy
C 光源	0.000	0.000	コニカミノルタ基準液晶	0.000	0.000
被測定液晶	-0.008	0.002	被測定液晶	-0.002	0.001

(※注 3-1-5)

マトリクス校正を行えば、加法混色が成立している光源に対して理論上、絶対値誤差は発生しません。ここで、誤差が0.002発生している理由は、液晶がクロストーク等により、完全には加法混色が成り立っていない(非常にわずかですが、ズレがあります)ことにあります。

3-2 γ 特性測定比較

図 3-2-1、表 3-2-1 に液晶の γ 特性を CA-210 液晶フリッカ測定プローブ (以降 CA-210 と記載します)、CA-110、CS-1000 で測定した結果を示します。(※注 3-2-1)

RGB の各単色において、CA-210 の測定結果と CS-1000 の測定結果は近い特性となります。

また、CA-110 の測定結果は、CA-210、CS-1000 の測定結果より緩やかなカーブを描くことがわかります。

CA-210、CA-110、CS-1000 で測定した γ 特性が異なるのは、各測定器の受光系の差に起因しています。これを以下に説明します。

図 3-2-2 に液晶の低輝度、高輝度での配光特性を示します。この配光特性は発光面に垂直な発光成分が等しくなるように正規化しています。

開口角 (※注 3-2-2) 10° の場合、高輝度では図の“緑色 + 黄色 + 赤色”の部分に相当する光を受光し、低輝度では“緑色 + 黄色 + 青色”の部分に相当する光を受光します。他方、開口角が非常に小さい場合 (例えば 1°)、高輝度、低輝度とも“黄色”の部分に相当する光を受光します。

図 3-2-2 より、“緑色 + 黄色 + 青色”の部分の面積は、“緑色 + 黄色 + 赤色”の部分の面積より大きいので、「開口角 1° 受光量」に対する「開口角 10° 受光量」の比は高輝度より低輝度の方が大きくなります。

従い、開口角 10° の方が開口角 1° と比較して、相対的に、低輝度での測定値が大きくなります。

開口角は、CS-1000 が 1° 、CA-210 が 5° 、CA-110 が 10° (近接条件) です。

このため、CA-110 で測定した γ 特性が、より緩やかなカーブを描くこととなります。

また、B 色では液晶に見られるブルーシフトによって、低輝度のレベルが浮く傾向が見られます。(※注 3-2-3)

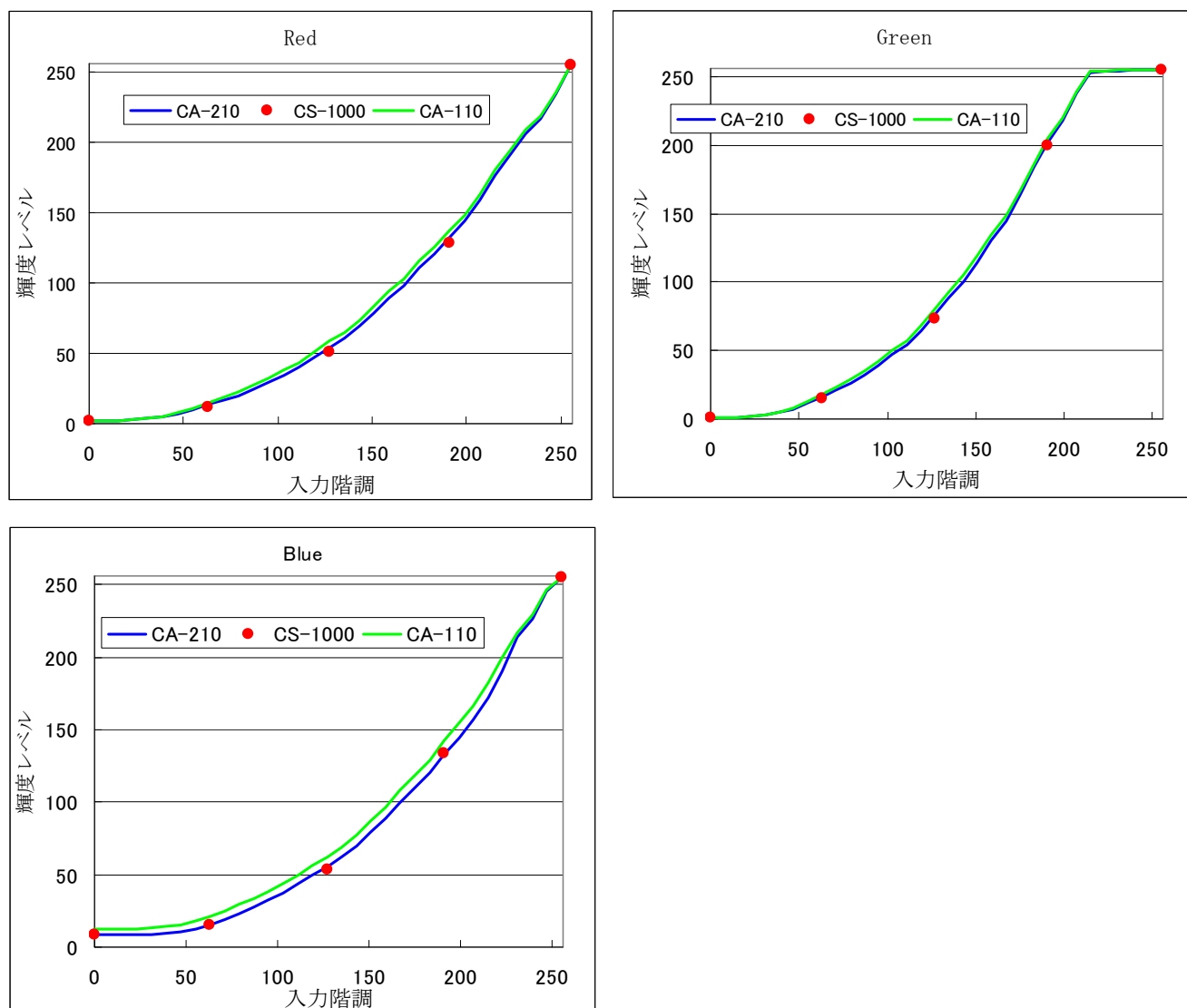


図 3-2-1 液晶の γ 特性測定結果

表 3-2-1 液晶の γ 特性測定結果

入力階調	輝度レベル								
	Red			Green			Blue		
	CS-1000	CA-210	CA-110	CS-1000	CA-210	CA-110	CS-1000	CA-210	CA-110
0	2.2	2.3	2.1	0.8	0.8	0.9	8.2	8.4	12.5
7		2.3	2.1		0.8	0.9		8.4	12.5
15		2.4	2.3		0.9	1.1		8.4	12.5
23		2.8	2.9		1.5	1.8		8.5	12.8
31		3.7	4.0		2.7	3.1		9.0	13.4
39		4.8	5.3		4.6	5.3		9.7	14.2
47		7.0	7.8		7.4	8.4		10.9	15.5
55		9.8	11.0		11.6	12.9		12.8	17.9
63	12.2	13.5	15.1	14.8	16.3	18.0	14.9	15.4	20.6
71		16.3	18.2		20.7	22.7		18.8	24.7
79		19.9	22.1		26.6	28.9		23.1	29.5
87		24.8	27.5		32.5	35.1		27.9	33.4
95		29.7	32.8		39.2	42.2		32.6	38.5
103		34.6	38.1		47.0	50.3		37.5	43.6
111		39.9	43.5		54.1	57.5		43.5	49.9
119		47.1	50.8		64.2	68.0		49.3	55.9
127	51.1	54.3	58.4	72.8	76.7	80.6	53.9	55.5	62.5
135		60.7	64.9		87.9	92.0		61.8	69.1
143		69.4	74.0		100.6	104.9		70.0	77.4
151		78.3	83.1		114.8	119.1		79.0	86.9
159		89.3	94.3		130.9	135.0		88.7	96.9
167		98.2	103.4		144.7	148.9		99.2	107.7
175		110.4	115.3		163.4	166.6		109.6	118.4
183		121.1	125.9		183.4	185.7		120.1	129.0
191	128.0	132.2	137.1	199.3	202.2	204.9	133.9	132.5	142.0
199		143.9	148.4		218.1	219.6		144.6	154.3
207		159.1	163.2		237.5	238.8		156.6	166.6
215		176.6	180.3		253.2	253.5		172.3	182.3
223		190.9	194.1		253.6	253.5		189.7	199.8
231		205.9	208.6		254.2	255.0		214.2	216.8
239		216.9	218.7		254.6	255.0		226.5	229.5
247		234.1	235.4		255.0	255.0		245.1	246.5
255	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0

液晶発光面

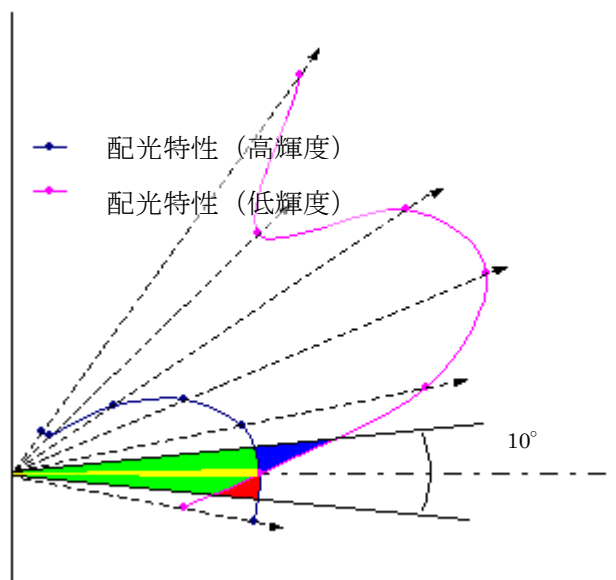


図 3-2-2 液晶の輝度－配光特性

(※注 3-2-1)

ディスプレイの映像入力信号に対する発光強度の関係は、それぞれを V_{in} , V_{out} とすると

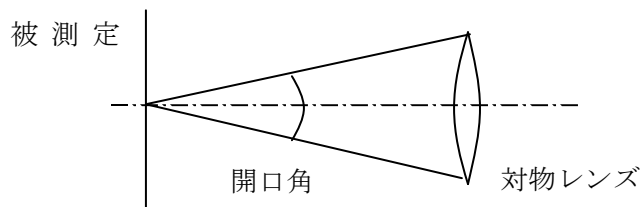
$$V_{out} = A \cdot V_{in}^\gamma \quad (A \text{ は定数})$$

と表すことができます。

この関係のことを、「 γ 特性」といいます。

(※注 3-2-2)

開口角とは、被測定物から出た光で測定器に取り込まれる光線の範囲を表した角度のことです。



(※注 3-2-3)

液晶は、漏れ光が存在し、この光は長波長より短波長成分をより多く含みます。低輝度においては、この影響が顕在化し、表示色は青みがかった色になります。このことを、「ブルーシフト」といいます。

今回測定した例では、Rの γ 特性はX値を、Gの γ 特性はY値を、Bの γ 特性はZ値を使用しています(IEC-61966準拠の評価方法)。X、Yセンサーは短波長成分に対する感度が小さく、ブルーシフトの影響をあまり受けませんが、Zセンサーは短波長成分に対する感度が大きいため、ブルーシフトの影響を受けます。このため、R、Gの γ 特性測定値はブルーシフトの影響を受けず、B色のみ大きく影響を受けることになります。

参考文献

・Yasuhiro Yoshida , Yoichi Yamamoto: 映像情報メディア学会誌Vol.56 No.8 pp.1279(2002)

3-3 CA-210(液晶フリッカ測定プローブ、小径液晶フリッカ測定プローブ), CS-1000の液晶測定値差 [さらに詳しく](#)

3-3-1 白色測定

高輝度(100cd/m²以上)、低輝度(20cd/m²付近)白色表示した液晶を各測定器
 CA-210小径液晶フリッカ測定プローブ(以降CA-210Sといいます)
 CA-210液晶フリッカ測定プローブ(以降CA-210といいます)
 CS-1000(分光色彩輝度計)

で測定した結果をそれぞれ表 3-3-1, 表 3-3-2 に示します。ここで、白色とは6500K付近の色のことです。測定した液晶は3機種です。分光器CS-1000の測定値を真値と考えるとCA-210S, CA-210のCS-1000との差は、CA-210S, CA-210の絶対値誤差と考えることができます。

高輝度、低輝度ともにCA-210S, CA-210とCS-1000の差は、色度で0.002以下、輝度で3%以下となっています。今回評価した液晶モニタはそれぞれ分光特性が異なりますが、このデータからCA-210S, CA-210は、それらの液晶モニタに対して高い精度で測定する可以说。 [「1-2 色彩計について」参照](#)

表 3-3-1 高輝度、白色表示液晶を測定した結果

液晶の種類	CS-1000			(CA-210S) - (CS-1000)			(CA-210) - (CS-1000)			(CA-210) - (CA-210S)		
	x	y	Lv	Δx	Δy	ΔLv	Δx	Δy	ΔLv	Δx	Δy	ΔLv
液晶 L	0.313	0.329	155.0	0.000	0.000	0.1%	0.000	0.000	0.0%	0.000	0.000	-0.1%
液晶 M	0.315	0.333	156.8	-0.001	-0.001	0.4%	-0.001	-0.001	-2.3%	0.000	0.001	-2.6%
液晶 N	0.313	0.330	99.1	0.001	-0.001	-1.2%	0.001	-0.001	-1.2%	0.000	0.000	-0.1%

表 3-3-2 低輝度、白色表示液晶を測定した結果

液晶の種類	CS-1000			(CA-210S) - (CS-1000)			(CA-210) - (CS-1000)			(CA-210) - (CA-210S)		
	x	y	Lv	Δx	Δy	ΔLv	Δx	Δy	ΔLv	Δx	Δy	ΔLv
液晶 L	0.311	0.328	20.1	0.000	0.001	2.8%	0.000	0.000	0.1%	0.000	-0.001	-2.6%
液晶 M	0.314	0.328	20.4	-0.002	0.000	-0.1%	-0.001	0.000	0.3%	0.001	0.000	0.4%
液晶 N	0.312	0.329	20.5	0.001	0.000	-0.2%	0.001	0.000	0.3%	0.000	0.000	0.5%

また、CA-210SとCA-210の差は、色度で0.001以下、輝度で3%以下となっています。表 3-3-3 に示したように、CA-210SとCA-210は開口角、測定径が異なりますが、白色液晶評価については、差異を与える大きな要因とはなっていないと言えます。

表 3-3-3 CA-210SとCA-210仕様比較

	CA-210S	CA-210
開口角[deg]	10	5
測定径[mm]	10	27

3-3-2 単色測定

単色表示した液晶を各測定器(CA-210S, CA-210, CS-1000)で測定した結果を表 3-3-4 に示します。測定した液晶は3機種です。

CA-210S, CA-210ともCS-1000との色度差は0.010以下になっています。従来の刺激値直読式色彩計では、単色の精度は0.020~0.030であったことと比較すると、大きく精度が向上していることがわかります。これは、CA-210S, CA-210がマトリクス校正を実施していることによります [「1-4 マトリクス校正」参照](#)。

また、CA-210SとCA-210の差は、色度で最大0.003発生しており、白色に比べてやや大きい値となっています。CA-210S, CA-210は「RGBマトリクス校正に白色1点校正を加えた」校正を実施しており、白色確度±0.002に対し単色確度は±0.004とやや悪くなっています [「1-4 マトリクス校正」参照](#)。上記差異はこのことが要因と考えられます。

表 3-3-4 単色表示液晶を測定した結果

液晶の種類	色	CS-1000		(CA-210S) - (CS-1000)		(CA-210) - (CS-1000)		(CA-210) - (CA-210S)	
		x	y	Δx	Δy	Δx	Δy	Δx	Δy
液晶 L	R	0.634	0.331	0.001	-0.001	0.001	0.000	-0.001	0.001
	G	0.293	0.591	0.001	-0.001	0.000	0.000	-0.001	0.001
	B	0.146	0.055	0.000	0.000	0.000	0.000	0.000	0.000
液晶 M	R	0.626	0.349	0.000	-0.001	0.000	0.000	-0.001	0.001
	G	0.289	0.594	0.000	-0.007	0.000	-0.007	0.000	0.000
	B	0.147	0.087	0.002	0.002	0.002	0.001	0.000	-0.001
液晶 N	R	0.596	0.346	-0.001	-0.001	0.002	-0.004	0.003	-0.002
	G	0.302	0.561	0.003	0.008	0.003	0.010	0.000	0.002
	B	0.150	0.136	0.003	0.002	0.003	0.003	-0.001	0.001

3-4 フリッカ測定精度 さらに詳しく

3-4-1 コントラスト方式

フリッカ測定精度(コントラスト方式)の仕様は、下記の条件において各々定義されています。

確度

30Hz AC/DC 10%正弦波 ±1%

60Hz AC/DC 10%正弦波 ±2%

繰返し

20~65Hz AC/DC 10%正弦波 1%

(※注 3-4-1)

上記条件外で、評価用治具を用いて実測した、フリッカ測定精度(コントラスト方式)の評価結果を、表 3-4-1~表 3-4-1 に示します ([「2-2 「フリッカ」確度定格・繰返し性の定義」参照](#)) (※注 3-4-2)。なお、下記の“周波数”とは、本体部に入力する信号の周波数のことです。フリッカ発生時の発光周波数はディスプレイの垂直同期周波数の1/2になります。

[「1-7-1 液晶ディスプレイにおけるフリッカとは」参照](#)

表 3-4-1~表 3-4-2 の結果から以下のことが言えます。

確度誤差は、

- フリッカ値の大小の影響を受ける。フリッカ値が大きい程、確度誤差が大きくなる(ただし、全ての条件において、確度誤差はフリッカ値の1/10以下になっている)。

- 周波数の影響を受ける。つまり、周波数が高くなるほど、確度誤差が大きくなる(回路がローパスフィルターを構成しており、高周波成分の方が、素子の特性バラツキの影響を受けるため)。

- 輝度の影響を受けない。

繰返し性は、フリッカ値、周波数、輝度の影響を、余りうけていない。

表 3-4-1 各輝度、周波数におけるフリッカ測定値の確度誤差 単位[%]

5cd/m ²						70cd/m ²					
真値	20Hz	30Hz	40Hz	50Hz	58Hz	真値	20Hz	30Hz	40Hz	50Hz	58Hz
5.7	0.1	0.0	-0.1	0.2	0.0	5.8	-0.1	-0.1	0.0	0.1	-0.1
10.0	0.0	0.0	0.0	0.1	-0.1	10.0	0.0	-0.1	-0.4	0.0	0.1
20.0	0.3	0.0	0.0	0.3	-0.3	19.8	-0.2	-0.1	-0.2	-0.1	0.3
40.0	0.4	0.3	0.2	0.5	-0.6	39.5	-0.2	-0.2	-0.4	-0.1	0.8
80.0	1.0	0.6	-0.2	0.6	-1.2	79.1	-0.3	-0.6	-1.0	-0.3	1.3

表 3-4-2 各輝度、周波数におけるフリッカ測定値の繰返し性 単位[%]

5cd/m ²						70cd/m ²					
真値	20Hz	30Hz	40Hz	50Hz	58Hz	真値	20Hz	30Hz	40Hz	50Hz	58Hz
5.7	0.3	0.2	0.2	0.1	0.2	5.8	0.0	0.0	0.0	0.0	0.0
10.0	0.1	0.1	0.2	0.3	0.4	10.0	0.3	0.1	0.1	0.1	0.1
20.0	0.2	0.2	0.2	0.1	0.2	19.8	0.2	0.1	0.1	0.1	0.1
40.0	0.2	0.2	0.2	0.1	0.2	39.5	0.3	0.1	0.1	0.1	0.2
80.0	0.1	0.2	0.2	0.2	0.2	79.1	0.4	0.0	0.2	0.0	0.1

3-4-2 JEITA方式

フリッカ測定値精度 (JEITA方式) の仕様は、下記の条件において各々定義されています。

確度 30Hz AC/DC 10%正弦波 ±0.5 [dB]
 繰返し性 30Hz AC/DC 10%正弦波 0.3 [dB]
 (※注 3-4-1)

上記条件外で、評価用治具を用いて実測した、フリッカ測定値精度 (JEITA方式) の評価結果を、表 3-4-3～表 3-4-5 に示します ([「2-2 「フリッカ」確度定格・繰返し性の定義」参照](#)) (※注 3-4-3)。

表 3-4-3～表 3-4-5 の結果から以下のことが言えます。

確度誤差は、

- ・フリッカ値の大小の影響を受けない。
- ・周波数の影響を受ける。周波数が高くなるほど、確度誤差が大きくなる (回路がローパスフィルターを構成しており、高周波成分の方が、素子の特性バラツキの影響を受けるため)。
- ・輝度の影響を受けない。

繰返し性は、フリッカ値、周波数、輝度の影響を余りうけていない。

表 3-4-3 各輝度、周波数におけるフリッカ測定値 (JEITA方式) の真値 単位[dB]

5cd/m ²						70cd/m ²					
AC/DC比 [%]	2 0Hz	3 0Hz	4 0Hz	5 0Hz	5 8Hz	AC/DC比 [%]	2 0Hz	3 0Hz	4 0Hz	5 0Hz	5 8Hz
5.7	-33.97	-37.06	-40.18	-46.34	-61.23	5.8	-33.85	-36.94	-40.07	-46.23	-61.12
10.0	-29.11	-32.20	-35.32	-41.48	-56.37	10.0	-29.11	-32.20	-35.32	-41.48	-56.37
20.0	-23.09	-26.18	-29.30	-35.46	-50.35	21.1	-22.64	-25.73	-28.86	-35.01	-49.90
40.0	-17.06	-20.16	-23.28	-29.44	-44.33	39.5	-17.18	-20.27	-23.40	-29.55	-44.44
80.0	-11.04	-14.14	-17.26	-23.42	-38.31	78.9	-11.16	-14.25	-17.38	-23.53	-38.42

(AC/DC比については (※注 3-4-1) 参照)

表 3-4-4 各輝度、周波数におけるフリッカ測定値 (JEITA方式) の確度誤差 単位[dB]

5cd/m ²						70cd/m ²					
AC/DC比 [%]	2 0Hz	3 0Hz	4 0Hz	5 0Hz	5 8Hz	AC/DC比 [%]	2 0Hz	3 0Hz	4 0Hz	5 0Hz	5 8Hz
5.7	0.13	0.28	0.47	0.74	1.17	5.8	0.10	0.24	0.44	0.71	1.14
10.0	0.11	0.25	0.45	0.72	1.15	10.0	0.08	0.23	0.42	0.69	1.12
20.0	0.11	0.26	0.45	0.72	1.15	21.1	0.18	0.32	0.52	0.80	1.22
40.0	0.11	0.26	0.45	0.73	1.16	39.5	0.17	0.31	0.51	0.79	1.21
80.0	0.10	0.25	0.44	0.72	1.15	78.9	0.16	0.31	0.51	0.78	1.21

(AC/DC比については (※注 3-4-1) 参照)

表 3-4-5 各輝度、周波数におけるフリッカ測定値 (JEITA方式) の繰返し性 単位[dB]

5cd/m ²						70cd/m ²					
AC/DC比 [%]	2 0Hz	3 0Hz	4 0Hz	5 0Hz	5 8Hz	AC/DC比 [%]	2 0Hz	3 0Hz	4 0Hz	5 0Hz	5 8Hz
5.7	0.01	0.01	0.02	0.01	0.01	5.8	0.01	0.01	0.01	0.01	0.01
10.0	0.01	0.01	0.01	0.01	0.01	10.0	0.01	0.01	0.01	0.01	0.00
20.0	0.00	0.00	0.00	0.01	0.01	21.1	0.00	0.00	0.00	0.00	0.00
40.0	0.00	0.00	0.00	0.00	0.00	39.5	0.00	0.00	0.00	0.00	0.00
80.0	0.01	0.00	0.00	0.00	0.00	78.9	0.00	0.00	0.00	0.00	0.00

(AC/DC比については (※注 3-4-1) 参照)

なお、周波数30Hzの信号をCA-210本体に入力した場合の、30Hz以外の周波数成分は、-80[dB]以下となります (各周波数成分にフリッカ定義式適用した場合の値)。これは、AC/DC比で0.1%以下に相当します。

(※注 3-4-1)

「AC/DC 10%正弦波」とは図 3-4-1
に示す信号のことです。

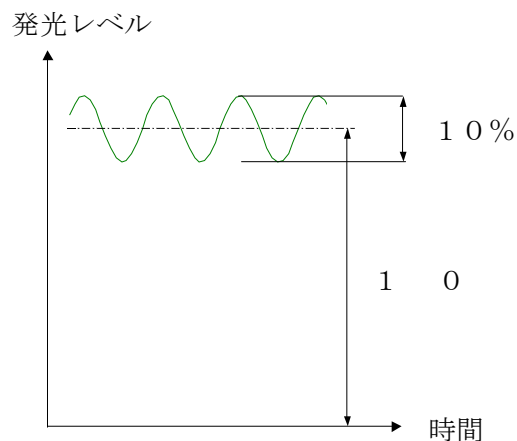


図 3-4-1 AC/DC 10%正弦波の定義

(※注 3-4-2)

「2-2「フリッカ」確度定格・繰り返し性の定義」において、絶対値精度は①プローブ部の精度(机上計算)、②本体部の精度に分け、両者の和で定義すると説明しています。ここでは、②の結果を記載しています。

①については、

30Hz、10%AC/DCにおける最大誤差 0.09 [%]

60Hz、10%AC/DCにおける最大誤差 0.32 [%]

であり、確度仕様の1/10~1/5未満であり、非常に小さい値となっています。

(※注 3-4-3)

「2-2「フリッカ」確度定格・繰り返し性の定義」において、絶対値精度は①プローブ部の精度(机上計算)、②本体部の精度に分け、両者の和で定義すると説明しています。ここでは、②の結果を記載しています。

①については、

30Hz、10%AC/DCにおける最大誤差 0.08 [dB]

であり、確度仕様の1/5未満であり、非常に小さい値となっています。

3-5 測定速度 さらに詳しく

CA-210の測定速度は、下記の各モード、条件によって異なります。

- ・測定スピードモード (FASTモード, SLOWモード)
- ・測定モード (xyLV, TΔuvLV, フリッカなど)
- ・測定同期モード (NTSC, PAL, など)
- ・測定プローブ数 (1~5)
- ・使用インターフェースの種類 (USB, RS232C)

表 3-5-1~表 3-5-3 に各モード、条件下での測定速度を示します。

表 3-5-1 USB使用時の測定スピードモード、測定同期モードと測定速度の関係 [回/秒]

測定スピードモード	測定同期モード	プローブ1本接続時		プローブ5本接続時	
		x y LV	フリッカ コントラスト方式	x y LV	フリッカ コントラスト方式
FAST	NTSC	20	16	13	7
	PAL	17	14	11	6
	EXT (60Hz)	20	16	13	7
	UNIV	8	※1	6	※1
	INT (60Hz)	20	16	13	7
SLOW	NTSC	5	※2	3.5	※2
	PAL	4	※2	3	※2
	EXT (60Hz)	5	※2	3.5	※2
	UNIV	1.5	※2	1.5	※2
	INT (60Hz)	5	※2	3.5	※2

表 3-5-2 RS232C使用時の測定スピードモード、測定同期モードと測定速度の関係 [回/秒]

測定スピードモード	測定同期モード	プローブ1本接続時		プローブ5本接続時	
		x y LV	フリッカ コントラスト方式	x y LV	フリッカ コントラスト方式
FAST	NTSC	17	16	7	6
	PAL	15	14	6	5
	EXT (60Hz)	17	16	7	6
	UNIV	7	※1	4	※1
	INT (60Hz)	17	16	7	6
SLOW	NTSC	4.5	※2	3	※2
	PAL	4	※2	2.5	※2
	EXT (60Hz)	4.5	※2	3	※2
	UNIV	1.5	※2	1	※2
	INT (60Hz)	4.5	※2	3	※2

表 3-5-3 測定モードと測定速度の関係 [回/秒]

測定同期モード 測定スピードモード	測定モード	プローブ1本接続時		プローブ5本接続時	
		USB	RS232C	USB	RS232C
NTSC FAST	x y LV	20	17	13	7
	u' v' Lv	19	17	12	6
	アナライザ [®]	20	17	13	7
	XYZ	20	17	13	7
	T Δ uv	19	17	12	5.5
	フリッカ・コントラスト方式	16	16	7	6

(※1) 設定不可

(※2) フリッカ測定は FAST モードのみ (SLOW モードにはならない)

なお、評価条件は、 PC: Pentium II (300MHz) OS: Windows 2000、Windows 98、Windows Me
RS232Cのボーレート: 38400bps

3-6 CA-100, CA-100PlusのCRT測定値差

3-6-1 白色測定

4種類のCRTを、異なる輝度(色温度は6500K)にて、CA-100、CA-100Plusで測定した結果を、表3-6-1に記載します。CA-100PlusとCA-100は白色では色度、輝度の差は、最大で0.002、0.5%以内であり、CA-100Plusの仕様範囲内です。

表 3-6-1 白色表示したCRTを測定した結果(高輝度)

CA-100Plus and CA-100 measurement data of CRTs W(6500K) high-luminance

	CA-100Plus			CA-100			(CA-100) - (CA-100Plus)		
	x	y	Lv	x	y	Lv	Δx	Δy	ΔLv
CRT 1	0.314	0.330	39.5	0.313	0.329	39.7	-0.001	-0.001	0.5%
CRT 2	0.314	0.329	74.6	0.312	0.328	74.4	-0.002	-0.001	-0.3%
CRT 3	0.314	0.330	86.8	0.313	0.329	87.0	-0.001	-0.001	0.2%
CRT 4	0.315	0.329	70.8	0.313	0.329	70.8	-0.002	0.000	0.0%

3-6-2 単色測定

4種類のCRTを、異なる輝度(色温度は6500K)に設定した後、単色表示に変更して、CA-100、CA-100Plusで測定した結果を、表3-6-2に記載します。

4種類のCRTは表3-6-1に示すように白色の輝度が異なるため、単色の輝度も異なっています。

CA-100Plusの測定値はCA-100に対して色度で0.011、輝度で12%前後の差があり、いずれも青色(B)で最大誤差が発生しています。

表 3-6-2 単色表示したCRTを測定した結果

CA-100Plus and CA-100 measurement data of CRTs RGB

	color	CA-100Plus			CA-100			(CA-100) - (CA-100Plus)		
		x	y	Lv	x	y	Lv	Δx	Δy	ΔLv
CRT 1	R	0.626	0.340	9.8	0.634	0.333	10.0	0.008	-0.007	2.5%
	G	0.288	0.607	27.5	0.278	0.607	27.9	-0.010	0.000	1.5%
	B	0.153	0.069	3.4	0.142	0.062	3.0	-0.011	-0.007	-12.5%
CRT 2	R	0.605	0.342	19.1	0.612	0.336	19.3	0.007	-0.006	1.0%
	G	0.292	0.588	50.7	0.283	0.587	51.0	-0.009	-0.001	0.6%
	B	0.161	0.087	8.0	0.151	0.080	7.1	-0.010	-0.007	-10.8%
CRT 3	R	0.633	0.328	24.3	0.641	0.322	24.9	0.008	-0.006	2.5%
	G	0.285	0.597	76.2	0.275	0.596	77.2	-0.010	-0.001	1.3%
	B	0.155	0.071	9.5	0.144	0.065	8.5	-0.011	-0.006	-11.0%
CRT 4	R	0.630	0.335	15.7	0.637	0.329	16.1	0.007	-0.006	2.5%
	G	0.299	0.605	46.9	0.289	0.604	47.5	-0.010	-0.001	1.3%
	B	0.154	0.072	6.0	0.143	0.065	5.3	-0.011	-0.007	-11.7%

単色で差が生じる原因は校正方法が異なることです。

CA-100は白色1点校正であるのに対して、CA-100Plusはマトリクス校正(R, G, Bマトリクス校正係数に白色1点校正を加えた方法)を採用しています。 ([「1-4マトリクス校正」参照](#))

これは次のように説明できます。

CA-100Plusでは図3-6-1bのようにW, R, G, Bマトリクス校正により測定値は白色、単色共にほぼ真値と等しくなります。またCA-100では図3-6-1dのように白色1点校正により測定値は白色では真値と等しくなりますが、単色では校正されていないため真値からずれたままです。

ここで図3-6-1aと図3-6-1cよりCA-100PlusとCA-100の校正前の真値と測定値の差が異なっていますが、これは分光感度のばらつきによるものです。刺激値直読式色彩計では分光感度のばらつきによる単色誤差が大きく、これは白色1点校正では軽減されません。

表3-6-3に分光感度のばらつきを考慮した場合の各校正方法による、任意のCRTで校正した場合の他21種類のCR

Tに対する単色の数値実験結果の最大値を示します。白色1点校正では色度で0.027と非常に誤差が大きく、マトリクス校正では色度で0.0016と一桁小さいことが分かります。

つまり、CA-100PlusはCA-100に対して白色の色度、輝度は互換性を保ちながら、単色の色度精度は大幅に改善しています。

CA-100Plusの場合

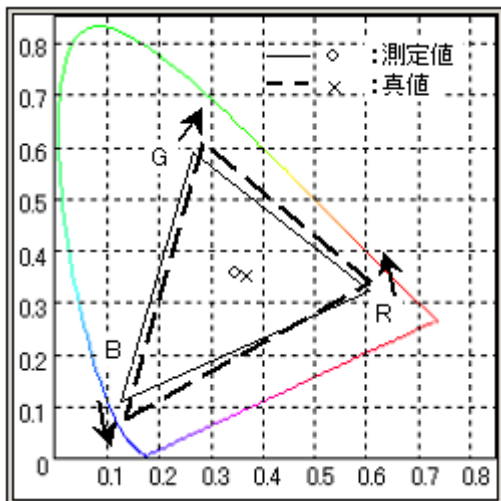


図 3-6-1a W,R,G,Bマトリクス校正前の測定値と真値

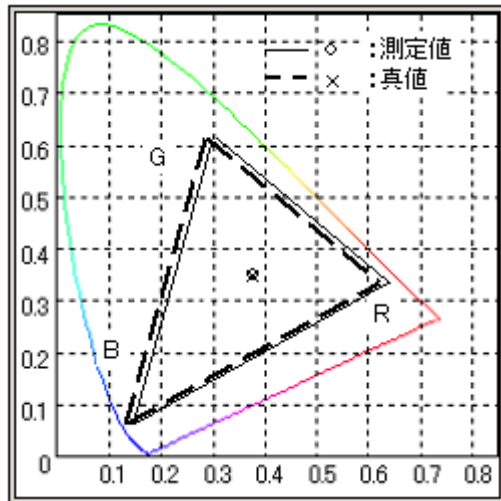


図 3-6-1b W,R,G,Bマトリクス校正後の測定値と真値

CA-100の場合

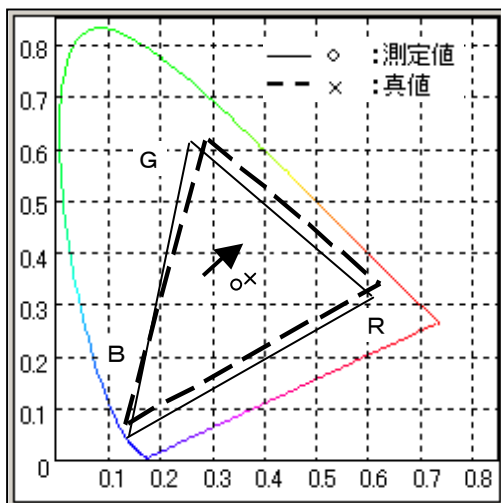


図 3-6-1c 白色1点校正前の測定値と真値

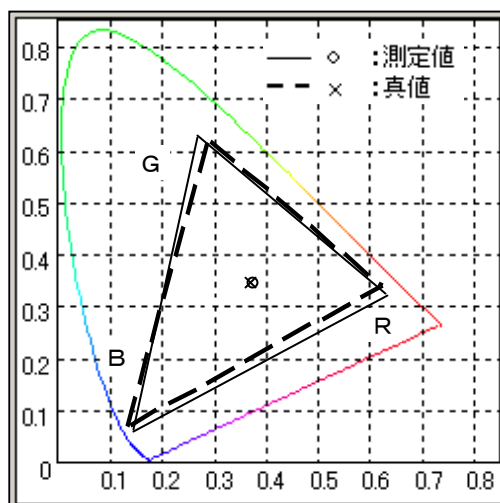


図 3-6-1d 白色1点校正後の測定値と真値

表 3-6-3 CRTでの単色の絶対値誤差の数値実験結果(最大値)

単色での真値との差						
	白色1点校正			マトリクス校正		
	Δx	Δy	$\Delta Lv(\%)$	Δx	Δy	$\Delta Lv(\%)$
絶対値誤差	0.027	0.021	14.8	0.0016	0.0016	1.6

3-7 CA-100Plus, CA-100のCRT、PDP測定値差

3-7-1 白色高輝度測定(CRT)

CA-100PlusはCA-100と測定値に互換性を持たせるために同じ分光応答度のセンサを用い、また同じ同じ分光放射特性をもつCRTで校正を行っています。

高輝度、白色表示したCRTを各測定器(CA-100Plus, CA-100)で測定した結果を表1に示します。ここで、白色とは6500K付近の色のことです。測定したCRTは4機種です。(CRT1はCA-100Plusの校正CRTです)

両測定器の差は、全てのCRTにおいて色度で0.002以下、輝度で1%以下となっており、CA-100Plus, CA-100の確度仕様誤差(色度 ± 0.002 , 輝度 $\pm 2\%$)以内の数値となっています。

先に述べたように、CA-100PlusはCA-100と同じ分光放射特性をもつCRTに対して校正されており、またセンサの分光応答度も等しいため測定値に互換性があると言えます。

表 3-7-1 高輝度、白色表示CRTを測定した結果

	CA-100Plus			CA-100			(CA-100) - (CA-100Plus)		
	x	y	Lv	x	y	Lv	Δx	Δy	ΔLv
CRT1	0.314	0.330	39.5	0.313	0.329	39.7	-0.001	-0.001	0.5%
CRT2	0.314	0.329	74.6	0.312	0.328	74.4	-0.002	-0.001	-0.3%
CRT3	0.314	0.330	86.8	0.313	0.329	87.0	-0.001	-0.001	0.2%
CRT4	0.315	0.329	70.8	0.313	0.329	70.8	-0.002	0.000	0.0%

3-7-2 白色低輝度測定(CRT)

低輝度、白色表示したCRTを各測定器(CA-100Plus, CA-100)で測定した結果を表2に示します。測定したCRTは4機種です。(CRT1はCA-100Plusの校正CRTです)

ここでも、両測定器の差は、全てのCRTにおいて色度で0.002以下、輝度で1%以下となっており、CA-100Plus, CA-100の確度仕様誤差以内の数値となっています。

低輝度での測定においても、CA-100PlusはCA-100と測定値互換性があると言えます。

表 3-7-2 低輝度、白色表示CRTを測定した結果

	CA-100Plus			CA-100			(CA-100) - (CA-100Plus)		
	x	y	Lv	x	y	Lv	Δx	Δy	ΔLv
CRT1	0.314	0.329	4.9	0.312	0.329	4.9	-0.002	0.000	0.8%
CRT2	0.312	0.329	5.3	0.314	0.329	5.3	0.002	0.000	0.0%
CRT3	0.314	0.329	5.0	0.313	0.329	5.0	-0.001	0.000	-0.8%
CRT4	0.314	0.330	5.0	0.312	0.330	5.1	-0.002	0.000	1.0%

3-7-3 白色PDP(Plasma Display)測定

低輝度から高輝度まで白色表示したPDPを各測定器(CA-100Plus, CA-100)で測定した結果を表 3-7-3 に示します。両測定器の差は、低輝度から高輝度まで色度で0.002以下、輝度で1%以下となっており、CRTと同様、CA-100Plus, CA-100の確度仕様誤差以内の数値となっています。

表 3-7-3 白色表示PDPを測定した結果

	CA-100Plus			CA-100			(CA-100) - (CA-100Plus)		
	x	y	Lv	x	y	Lv	Δx	Δy	ΔLv
luminance A	0.313	0.328	228.0	0.311	0.327	227.0	-0.002	-0.001	-0.4%
luminance B	0.313	0.329	62.2	0.311	0.328	62.0	-0.002	-0.001	-0.3%
luminance C	0.312	0.332	5.3	0.310	0.332	5.3	-0.002	0.000	-0.2%

3-7-4 繰返し性の比較

低輝度・白色 (6500K付近) のCRTを測定した場合の繰返し性について、CA-100とCA-100Plus の比較結果を表 3-7-4 に示します (CA-100, CA-100Plus とも、測定速度が 5[回/秒]となる条件で評価)。この結果から、低輝度での繰返し性はCA-100Plus が、CA-100に対して約2倍向上していることがわかります。

表 3-7-4 CA-100とCA-100Plus の繰返し性の比較

評価時Lv [cd/m ²]	CA-100Plus		CA-100	
	x	y	x	y
0.05	0.0018	0.0018	0.0037	0.0030
0.10	0.0009	0.0010	0.0025	0.0016

CA-210/100plus 技術資料

4. CA-SDK ソフト解説

4-1 COMとは

ここでは、CA-SDK を利用して、Visual C/C++ (以降、VC++ と記す) アプリケーションを作成する際に必要な COM 技術の概略説明をします。

なお、CA-SDK は“32ビットモード”のアプリケーションにのみ対応しています。CA-SDK を使用してアプリケーションを開発する場合は、“32ビットモード”を指定してください。

4-1-1 COM インターフェース

4-1-1-1 インターフェース

ソフト開発の現場で、オブジェクト指向技術がポピュラーになってきています。オブジェクト指向は、高品質なソフトを効率よく開発する手法であるといわれていますが、その説明の中で、“ソフトウェア開発をするのに一番よい方法は、ソフトウェアを作らないことだ”という言い方がよくされます。VC++などのオブジェクト指向言語を利用し、オブジェクトの長所を“うまく”ひきだすような形でプログラミングできさえすれば、いわゆるソフトウェアの部品化/再利用という長年の“夢”が実現できる…ということを指しています。

では、“うまく”プログラミングするにはどうしたらよいのでしょうか。この点で重要とされている技術の一つに、“インターフェース”プログラミングというものがあります。新規にソフトを作らずにソフト開発するためには、“標準”部品の機能とその使用法・組み合わせ方～インターフェース、を適切に規定し、その“標準”部品を組み合わせる形でプログラミングする、実際のソフトウェアは、“標準”部品の仕様を満たす既存の品質確認済みの部品～インターフェースを実装したものをプラグインし、積み木細工をするように構築する、といったイメージです。

“インターフェース”という言葉はいろいろな意味で使用されますが、COM におけるインターフェースは、上の意味をもちます。オブジェクト指向で中心的な役割をもつ“クラス”は、COM においては実装にかかわる概念を指し、“インターフェース”プログラミングということからすれば、それに依存しない形でプログラミングできるようにする必要があります。COM は、この点で徹底して“インターフェース”プログラミングの考え方にたつプログラミング手法であり、インターフェースが全て、といわれています。

また、VC++などのオブジェクト指向プログラミング言語における部品化/再利用は一般にはそのソースコードを対象としています。従って、それは使用する言語/ツールに依存するものとなります。これに対し、COM はその仕様でインターフェース実装のバイナリ構造～メモリー上の構造を規定し、言語/ツール/プラットフォームに依存しないインターフェースを実現することを目指しています。COM 仕様に合致したインターフェースバイナリの処理さえできれば、バイナリレベル～.DLL/.EXE ファイルにパッケージした形で、言語などに依存しない部品化/再利用が可能になっています。

4-1-1-2 COM インターフェース

COM インターフェースの定義は、それを実装する C++ファイルとは別の.IDL(.ODL)ファイルでなされます。

MFC の COM サポート機能を使用して COM (オートメーション) サーバーを作成する場合、ツールが自動的に.ODL ファイルを作成してくれます。また、インターフェースの実装用クラスをプロジェクトに挿入すると、自動的に、対応するインターフェース定義が.ODL ファイルに挿入されます。従って、通常、インターフェース定義ファイルを直接操作することはほとんどありません。ただし、CA-SDK の C++サンプルソフトではイベント(アプリケーション側で COM インターフェースを定義することになります)のプログラミングで、一部.ODL ファイルを修正する必要があります。このためも含め、ここで、簡単に COM インターフェース定義(ファイル)を説明します。

1-1. に記したとおり、COM はインターフェースがすべて、です。インターフェース定義さえあれば、そのクライアントやサーバーを実装することができます。

CA-SDK を利用してソフト開発する際、最初に、アプリケーションプロジェクトに CA-SDK コンポーネントの参照設定/追加をします。下に示すリストは、その操作によってプロジェクトに取り込まれる実体(タイプライブラリと呼ばれているもの)に対応するもので、これがインターフェース定義になります。

リストにおいて、

①library ステートメント以降の部分がタイプライブラリに反映される部分です。最初に、②Ca200 COM クラスが、③ICa200 COM インターフェースを実装するクラスであることが記述されています。④からその ICa200 COM インターフェースの定義が記述されています。

⑤と次の行で、Cas という Icas COM インターフェースを取得するプロパティ、⑥で SetConfiguration()というメソッドがあり、long 型・BSTR 型・long 型・long 型の4つの入力引き数をとること、⑦で AutoConnect()という引き数のないメソッド、⑧と次の行

で、SingleCa という ICa COM インターフェースを取得するプロパティが定義されています。

また、ここにはサーバー/クライアントのプログラムを作成する上で必要なもう1つの情報が入っています。GUID といわれる ID 値です。リストでは、②Ca200 COM クラスの定義の上に、その CLSID(クラス ID。リストでは、“uuid()”の中にある部分)、④ICa200 COM インターフェース定義の上の部分に、その IID(インターフェース ID が定義されています。

後で説明しますオートメーションの場合を除いて、インターフェース定義中にある名前は、ソースプログラムの中でしか意味を持ちません。実行(ファイル)においては、この ID 値で識別し、処理されます。

このように、タイプライブラリー中にはインターフェースプログラミングをする上で必要な情報が全て含まれています。COM をサポートする開発ツールは、これをもとにそれぞれのやり方で開発者用の C++ ファイルを作成します(“CA-SDK による VC++ アプリケーションの作成方法”、“CA-SDK サンプルソフト用コントロールの使用法”でその例を説明しています)。何らかの理由でツールのサポートを利用しないでプログラミングする必要がある場合、これを MIDL コンパイラにかければ開発上最低限必要な C ファイルを得ることができます。

```
// Generated .IDL file (by the OLE/COM Object Viewer)
//
// typelib filename: CA200Srvr.dll
...
library CA200SRVRLib ①
{
    [
        uuid(006B0650-AF9A-4EE1-B18F-B5740004D7CE),
        ...
    ]
    coclass Ca200 { ②
        [default] interface ICa200; ③
    };

    [
        ...
        uuid(DB87A8F6-FAF3-433A-B7F3-31BB4D759361),
        ...
    ]
    interface ICa200 : IDispatch { ④
        [..., propget, ...] ⑤
        HRESULT Cas([out, retval] ICas** CasVal);
        ...
        HRESULT SetConfiguration( ⑥
            [in] long CaNumberVal,
            [in] BSTR ConnecStringVal,
            [in] long PortVal,
            [in, optional, defaultvalue(38400)] long BaudRateVal);
        ...
        HRESULT AutoConnect(); ⑦
        [..., propget, ...] ⑧
        HRESULT SingleCa([out, retval] ICa** SingleCaVal);
    };
    ...
}
```

4-1-1-3 COM インターフェースプログラミング

COM コンポーネントを利用するクライアントソフトウェアでは

① 所要の COM インターフェースを実装し公開する COM クラスのインスタンス～COM オブジェクトを生成し、IUnknown インターフェースを取得する

② IUnknown から所要の COM インターフェースを取得する

③ 取得した COM インターフェースを利用して、クライアント固有の必要な処理を行う

④ インターフェースが不要になったら、解放する

という手順を実行することになります。

この説明中の“IUnknown インターフェース”は1-2. で示した定義中には現れませんでした。

CA-SDK は、いわゆるデュアルインターフェース(オートメーション機能をあわせ持つインターフェース)として定義されています。1-2. のリストの④ICa200 COM インターフェースの定義では、ICa200 は IDispatch インターフェースを継承しています。この IDispatch がオートメーション機能を実現するインターフェースです。そして、IDispatch 定義では(このリストにはありませんが)IUnknown インターフェースを継承しています。

COM の仕様では、全ての COM インターフェースは IUnknown インターフェースを継承する必要があります。従って、全ての COM オブジェクトが IUnknown インターフェースを実装していることになります。

この IUnknown インターフェースは以下の3つのメソッドを持ちます。

- QueryInterface
- AddRef
- Release

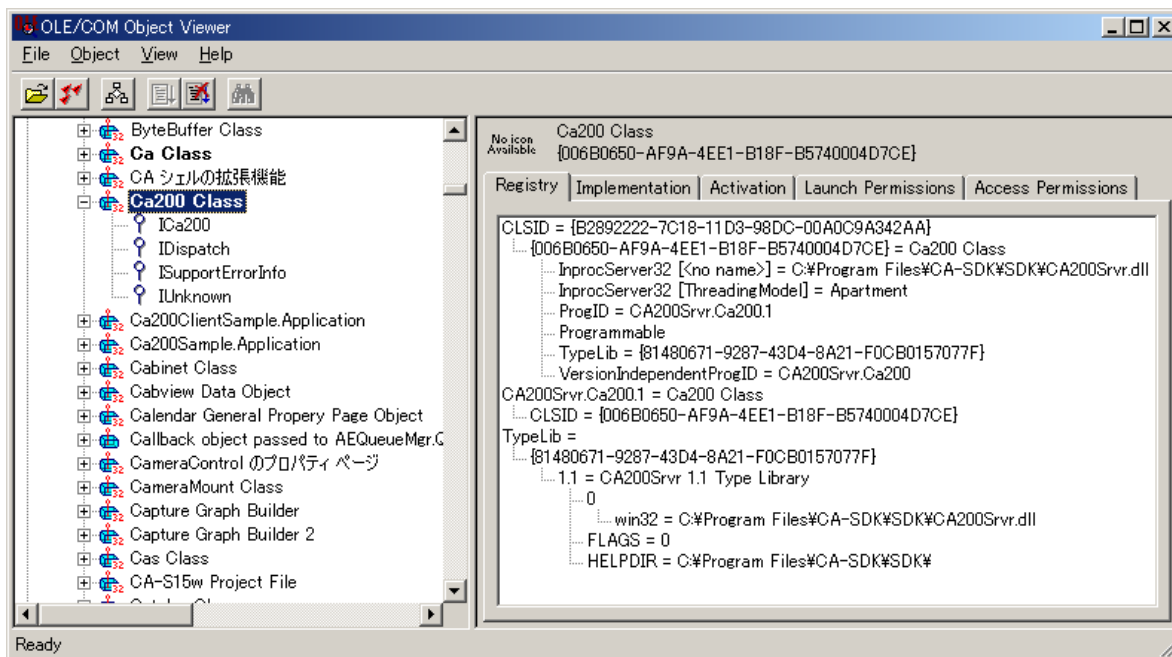
QueryInterface メソッドは、上記手順の②インターフェースのクエリー処理をします。

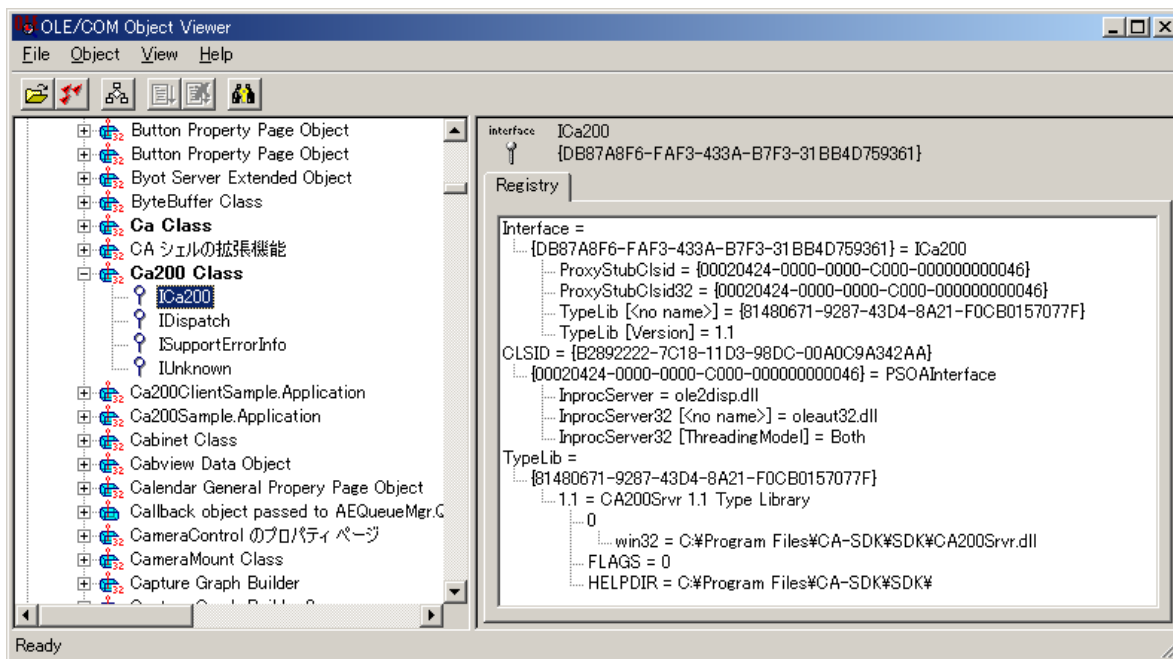
AddRef・Release メソッドは、COM オブジェクトの参照カウントの制御をするもので、上記④インターフェースの解放時に使用します。COM コンポーネントは“部品”として、いろいろなクライアントから同時に使用される可能性があります。その寿命管理、リソース管理は簡単ではありません。COM では当然、この機能も IUnknown により“インターフェース”プログラミングします。

ツールのサポート機能を利用する場合、この IUnknown の操作は通常、ツールが生成する C++クラスが隠蔽し、シンプルにプログラミングができるようになっていますが、ツールのサポートを受けない場合はユーザーが適切に IUnknown を使用してプログラミングする必要があります。

①の部分は、COM API として実装されています。下に、CA-SDK コンポーネントのレジストリー登録データの一部を示します。CA-SDK をインストールすると、コンポーネントファイル“CA200Srvr.dll”中にある登録用リソースを使用して、レジストリー登録されます。COM API などはこちらを参照して、クライアントソフト実行時、コンポーネント～ソフトウェア“部品”をリンクし、オブジェクトを生成します。全ての COM クラスは IUnknown インターフェースを実装しているため、他に問題がなければ、かならず IUnknown インターフェースを取得できます。そして、それを利用し、必要なインターフェースを獲得できます。

上記 COM API を使用するためにはシステムの COM ライブラリーの初期化が必要です。ツールサポートを受ける場合、ツールが自動生成したコードにより、適切に実行されます。サポートを受けない場合、この初期化もユーザーが明示的に行う必要があります。COM ではスレッド対応のため、スレッドモデルという概念があります。CA-SDK は STA モデルで動作するように設計されています。プログラムで COM ライブラリーの初期化を行う際には、アプリケーションのメインスレッドにおいてこのモデルを指定して初期化すると、最も効率よく動作させることができます。





4-1-2 オートメーション

スクリプト言語プログラムで COM コンポーネントを利用する場合、それ自体ではバイナリーコードを持たないため、個々のインターフェースのバイナリー構造にアクセスできません。これに対応するため、オートメーションという機能が準備されています。(純粋な、デュアルインターフェースによるものでない)オートメーションは、インターフェース定義では dispinterfadce というステートメントで定義しますが、実装は前記の IDispatch によって行われます。

VC++ の MFC は(この純粋な)オートメーションをサポートしています。オートメーションではオブジェクトの作成、インターフェースのメソッド/プロパティへの参照はスクリプトに対応するため、その“名前”を使用します。それを専用のインターフェース IDispatch が処理し、コンポーネント呼び出しを間接的に実行します。

CA-SDK のイベントは、この dispinterfadce で定義されたものとしてプログラミングされています。イベント処理に MFC を利用したアプリケーション作成例が、“CA-SDK による VC++アプリケーションの作成方法”に説明してあります。

ツールサポートを利用しない場合は、手作りで IDispatch インターフェースのプログラミングをすることになります。いろいろな実装法がありますが、カスタムインターフェースに比べると複雑です。新たにいちから作成する場合は、通常のオートメーション機能でよければ、できるだけツールのサポートを利用したほうがよいかと考えます。

COMのお話

オーディオアンプ・サウンドプロセッサ・SP・CD/DVD プレーヤ・ビデオモニターなどいろいろなコンポーネントを組み合わせて、自分だけの AV システムを構築する、私の夢のひとつです。残念ながら、まず、先立つものがありません。また、場所的な問題もあり、夢は当分かなわないでしょう。でも、そうした現実的な問題を抜きにすれば、夢はすぐにも実現できます。専門店に行けば、かなりマニアックなコンポーネントでも簡単に手に入れることができます。一通りそろったら、ケーブルで接続します。ケーブルといっても、びんからきりまで、端子のめっきは…、芯線は…。しかし、びんのケーブルでも、きりのケーブルでも機能は同じ、“標準”のケーブルです。コンポーネントを標準のケーブルで接続すれば、それでOK、どんなCD/DVD メディアでも、プレーヤに突っ込めば、それですばらしい音楽/映像が再現できます。各コンポーネントは、標準の入力を標準の出力に変換、それが標準のケーブルを介して伝えられ、そして、すべてがうまくいきます。もし、気に入らないところがあれば、コンポーネントをこだわりの一品に差し替えるだけです。

前置きが長くなりました。COM を説明します。ソフトウェアの分野でも、上のような形でプログラムが構築できたら、“ソフトウェア危機”といわれる状況は改善されるかもしれません。ソフトウェアをコンポーネント化…部品化する。コンポーネントを利用…部品を再利用して構築する、このような形で作成するソフトウェアを、“コンポーネント”ウェアと呼ぶことがあります。COM(Component Object Model)は、MS が仕様を定めたコンポーネントウェアを実現するための仕組み(モデル)であり、一連のサービス(Windows が実装し、提供する)を指します。COM を利用して、作成したコンポーネントは COM コンポーネントと呼ばれます。

意識する・しないは別として、Windows 環境でコンピュータを利用している私たちは、COM の恩恵を幅広く受けています。オブジェクトのリンクと埋め込み、クリップボード転送、ドラッグ&ドロップといった OLE 関連機能、オートメーションでサポートされるマクロ機能などはいまや、コンピュータでの作業には必須のものとなっています。プログラムを組む人は、MS などが提供する GUI やデータベース利用のためのコントロールなどを利用して、高度な機能を容易に実現しています。これらは、みな COM をベースとしています。このように、COM 自体が表舞台に現れることはありませんが、縁の下で、Windows 実行環境を支えています。

従来、Windows システムを含めて、外部に機能を公開するには、それを関数化し、DLL にパッケージして提供していました。DLL には、一連の公開関数がフラットに配置され、アプリケーションプログラムを組む場合、直接、その関数を呼び出していました。

COM は、オブジェクト指向のパラダイム(枠組み)を採用しています。COM コンポーネントは、クラスを公開し、クラスはインターフェースを公開します。インターフェースはメソッド～関数の集まりです。オブジェクト指向プログラミングでは、現実にある世界をソフトでシミュレーションする形で、プログラムを設計します。その世界に登場し、活動する個々の“もの”を表すのがクラス～オブジェクトであり、それが他の“もの”との関係で、どう動かかを規定するのがそのインターフェース・メソッドになります。Excel でマクロを組んだことのある方は、Excel のオブジェクト(～クラス)階層図をみたことがあるかもしれません。Excel のデータはブックという単位にまとめられます。ブックは、シートから構成され、シートはセルから構成され…、という構造をもっています。私たちは、シートやセルを選択し、ある操作を実行して、データを処理します。Excel のオブジェクト階層は、こうした構造を反映して構築されており、シートオブジェクトやセルオブジェクトには、私たちが行う操作に対応するメソッドがまとめられています。そこには、私たちが目にし、操作している Excel の世界があります。このオブジェクト階層を利用し、ソフトウェアで所要のデータ処理をシミュレートする、こういう形でソフトウェアを設計できれば、ソフトの都合に振り回されることなく、ソリューションを得るための作業に集中できます。COM は、従来のフラットな構造の API でなく、上記のような現実の世界の構造～クラス階層をもった API を、使用する言語・ツールに依存することなしに提供することを可能にしてくれます。

COM が出現して以来、MS は、Windows システムの拡張に、COM コンポーネントを利用してきたといわれています。Windows2000 になって、COM は COM+ といわれるものに進化しました。Windows2000 では、OS システムと COM の結びつきは、今まで以上に、緊密になり、Windows2000 で実現された重要な機能は、COM+ の進化した仕組みに深く組み込まれているといわれています。

この進化を理解するキーワードに、Windows DNA、DCE といったものがあります。当初、ネットワークは、コンピュータ間の通信の手段でしかありませんでした。プログラムで実現されるシステムの活躍の場は、ネットワークの大海に浮かぶ個々のコンピュータの独立した島の中だけでした。この状況はインターネットの普及で、大きく様変わりしました。クライアント/サーバーという 2 層モデルは、ごく普通、今では、ネットワーク環境がシステムの活動する場となってきています。

COM は、イントラネット環境でみた場合、優れた実行環境を提供するといわれています。COM コンポーネントがネットワーク上のどこにあらうと、プログラムはその位置に依存することなくアクセスし、利用できます。そして、Windows2000 において、COM は COM+ に進化し、企業のいわゆるミッションクリティカルな分散システムを構築するに十分なインフラを提供するまでになったといわれています。

4-2 CA-SDKによるVC++アプリケーションの作成方法

ここでは、VC++コンパイラの拡張機能…#import ディレクティブを使用して、CA-SDKアプリケーションを作成する方法を説明します。

MFCのCOMサポート機能を使用したアプリケーション作成法については、[“4-3 CA-SDK サンプルソフト用コントロールの使用法”](#)の中で、一部説明しています。

サンプルソフト作成時の開発ツールの操作等については、この説明書の [4-2-2 SDK アプリケーション作成法\(補足説明\)](#)以降で説明しています。

VC++サンプルソフトは、CA-SDKが“C:\Program Files\CA-SDK”フォルダーにインストールされているものとして作成されています。これ以外のフォルダーにインストールされている場合、“Ca200SampleDlg.h”、“Ca200SampleDlg.cpp”の#import文のパス記述部分を修正する必要があります。

4-2-1 SDK アプリケーション作成法

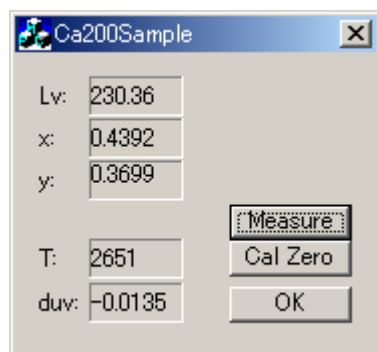
4-2-1-1 アプリケーションプロジェクトの作成

サンプルソフトのプロジェクトを下記の内容で作成します。

- サンプルソフトはMFCを利用したダイアログベースのアプリケーションとします。
- プロジェクト名は、“Ca200Sample”、SDKのイベント対応とするため、オートメーションのサポートの設定をONとします。

4-2-1-2 UIの作成

- UIは次の図のようにします。スタティックテキストを10(ラベル用・測定値表示用各5)、プッシュボタンを3つ利用します。“OK”はデフォルトでセットされているものを利用します。



- 測定値表示用スタティックテキスト・プッシュボタンには下記のIDを設定します。

Lv 表示 : IDC_STATIC_LV
 x 表示 : IDC_STATIC_X
 y 表示 : IDC_STATIC_Y
 T 表示 : IDC_STATIC_T
 duv 表示 : IDC_STATIC_DUV

Measure ボタン : IDC_BUTTON_MSR
 Cal Zero ボタン : IDC_BUTTON_CAL0

- Measure ボタンは、“無効”の設定、キャンセルボタンは、“不可視”の設定にします。

4-2-1-3 UIオブジェクトとコードの対応付け

- Measure ボタン・Cal Zero ボタンに対して、BN_CLICKED メッセージハンドラーを追加します。
- 測定値テキスト表示用スタティックテキストに対して、下記の DDX 用メンバーを追加します。

```

Lv 表示 :      CString m_strLv;
T 表示 :      CString m_strT;
x 表示 :      CString m_strx;
y 表示 :      CString m_stry;
duv 表示 :    CString m_strduv;

```

4-2-1-4 CA-SDK のプログラミング**4-2-1-4-1 CA-SDK オブジェクト生成**

ここから、CA-SDK を利用して CA を制御するコードの作成に入ります。

まず、CA-SDK コンポーネントが提供するオブジェクトを生成・初期化して CA-SDK を利用できる状態にします。VC++ の COM サポート機能である #import ディレクティブを利用します。

CA-SDK コンポーネント“CA200Srvr.dll”には、タイプライブラリーがリソースとしてパッケージされています。 #import ディレクティブのタイプライブラリー指定は、このファイル名でします。

- #import ディレクティブ文を 2 箇所、“Ca200SampleDlg.h”と“Ca200SampleDlg.cpp”リストで示すとおり追加します。

Ca200SampleDlg.h

```

...
#if !defined(AFX_CA200SAMPLEDLG_H__85F30F2C_42D9_48BB_8323_005EF636DB07__INCLUDED_)
#define AFX_CA200SAMPLEDLG_H__85F30F2C_42D9_48BB_8323_005EF636DB07__INCLUDED_

// CA-SDK
#import "C:\Program Files\CA-SDK\SDK\CA200Srvr.dll" no_namespace ¥
        no_implementation
...

```

Ca200SampleDlg.cpp

```

...
// CA-SDK
#include "Const.h"
#include "CaEvent.h"
#import "C:\Program Files\CA-SDK\SDK\CA200Srvr.dll" no_namespace ¥
        implementation_only

class CCa200SampleDlgAutoProxy;
...

```

ここで、ビルドすると、“CA200Srvr.tlh”・“CA200Srvr.tli”ファイルが自動生成されます。

“CA200Srvr.tli”には、“CA200Srvr.dll”が提供するインターフェース ICa200・ICas…ごとに C++ のクラス ICa200Ptr・ICasPtr…が作成されます(以降、これらのクラスをスマートポインターと記します)。

- 自動生成されるスマートポインタ型の Public メンバー変数をメインウィンドクラス Ca200SampleDlg に追加します。

```

IOutputProbesPtr m_pOutputProbesObj;
IProbesPtr m_pProbesObj;
ICasPtr m_pCasObj;
IMemoryPtr m_pMemoryObj;
IProbePtr m_pProbeObj;

```

```
ICaPtr m_pCaObj;
ICa200Ptr m_pCa200Obj;
```

- CA-SDK オブジェクトの生成と初期化コードを作成をします。メインウィンドーの初期化メンバー関数 OnInitDialog() 中に、リストに示す文を追加します。

コードでは、多点 CA・多点プローブに対応できるように、Ca200 オブジェクトの SetConfiguration メソッドを使用した初期化をします。

- ・ Ca200 オブジェクト生成 (①)
- ・ その SetConfiguration() メソッドにより CA 測定器 1 台使用の下記構成を設定 (②)
 - ・ CA 測定器の CA 番号 = 1
 - ・ プローブは 1 本接続、プローブ番号 = 1
 - ・ USB 接続

③以降で、CA-SDK のオブジェクト階層中の CA 測定器を制御するためのオブジェクトを設定します。

- ・ Ca200 オブジェクトから、Cas コレクションを取得。
- ・ Cas オブジェクトから、Ca オブジェクトを取得。
- ・ Ca オブジェクトから OutputProbes コレクションを取得
- ・ OutputProbes コレクションをリセット
- ・ OutputProbes コレクションにプローブ番号 1 のプローブを追加、出力プローブに設定。
- ・ OutputProbes コレクションから Probe オブジェクトを取得。
- ・ Ca オブジェクトから Memory オブジェクトを取得。

次に、④以降で、各オブジェクトにより以下の CA 測定器の初期化・メモリーチャンネルの設定をします。

- ・ 同期モードを NTSC に設定。
- ・ FAST/SLOW モードを FAST に設定
- ・ アナログ表示のレンジを 2.5%・2.5% に設定
- ・ 出力表示を Lvxy に設定
- ・ 表示桁数を 4 桁に設定
- ・ メモリーチャンネルは 0 (コニカミノルタ校正) に設定

(この部分の詳細は、“プログラミングガイド”の 3. SDK によるプログラム作成法-3.1 基本編を参照してください)

Ca200SampleDlg.cpp

```
BOOL CCa200SampleDlg::OnInitDialog()
{
    ...
    // TODO: 特別な初期化を行う時はこの場所に追加してください。
    // CA-SDK

    long lcan = 1;
    _bstr_t strenfig(_T("1"));
    long lprt = PORT_USB;
    long lbr = 38400;
    _bstr_t strprbid(_T("P1"));
    _variant_t vprbid(_T("P1"));

    try{
```

```

        m_pCa2000bj = ICa200Ptr(__uuidof(Ca200));①
        m_pCa2000bj->SetConfiguration(lcan, strenfig, lprt, lbr);②
    }
    catch(_com_error e) {
        CString strerr;
        strerr.Format(_T("HR:0x%08x¥nMSG:%s"), e.Error(), ¥
            (LPCSTR)e.Description());
        AfxMessageBox((LPCSTR)strerr);
        return TRUE;
    }

    m_pCasObj = m_pCa2000bj->Cas;③
    m_pCaObj = m_pCasObj ->ItemOfNumber[lcan];
    m_pOutputProbesObj = m_pCaObj ->OutputProbes;
    m_pOutputProbesObj ->RemoveAll();
    m_pOutputProbesObj ->Add(strprbid);
    m_pProbeObj = m_pOutputProbesObj ->Item[vprbid];
    m_pMemoryObj = m_pCaObj->Memory;

    m_pCaObj->SyncMode = SYNC_NTSC;④
    m_pCaObj->AveragingMode = AVRG_FAST;
    m_pCaObj->SetAnalogRange(2.5, 2.5);
    m_pCaObj->DisplayMode = DSP_LXY;
    m_pCaObj->DisplayDigits = DIGT_4;
    m_pMemoryObj->ChannelNO = 0;
    ...
}

```

4-2-1-4-2 CA-SDK オブジェクト利用

サンプルソフトでは、

- Cal Zero ボタンがクリックされたら、OCal を実行する。
- Measure ボタンがクリックされたら、10 回測定し、Lv・x・y、T、 $\angle uv$ の値を測定の都度、メインウィンドーに表示するという動作とします。

- Cal Zero ボタンのハンドラー CCa200SampleDlg::OnButtonCal0() に、リストに示す OCal 実行のコードを追加します。

コードでは①で、Ca オブジェクトの CalZero メソッドを呼び出し、OCal を実行します。OCal が正常に終了した場合、Measure ボタンを有効にして、測定できる状態にします。

Ca200SampleDlg.cpp

```

void CCa200SampleDlg::OnButtonCal0()
{
    ...
    // TODO: この位置にコントロール通知ハンドラ用のコードを追加してください
    // CA-SDK
    try{
        m_pCaObj->CalZero();①
    }
    catch(_com_error e) {
        CString strerr;
        strerr.Format(_T("HR:0x%08x¥nMSG:%s"), e.Error(), ¥

```



```

        (LPCSTR)e.Description());
    AfxMessageBox((LPCSTR)strerr);
    return;
}

CButton* pb;

pb = (CButton *)GetDlgItem(IDC_BUTTON_MSR);
pb->EnableWindow(TRUE);
...
}

```

- Measure ボタンのハンドラー CCa200SampleDlg::OnButtonMsr() に、リストに示す測定実行コードを追加します。

コードでは①で、Ca オブジェクトの Measure() メソッドを呼び出し、測定を実行します。

②以降で、Probe オブジェクトのプロパティにより、測定結果を得て、測定結果表示に関連付けられた DDX メンバーに設定し、最後にコントロールに反映させています。

Ca200SampleDlg.cpp

```

void CCa200SampleDlg::OnButtonMsr()
{
    ...
    // TODO: この位置にコントロール通知ハンドラ用のコードを追加してください
    // CA-SDK
    int i;
    float fLv;
    float fx;
    float fy;
    long lT;
    float fduv;

    for (i = 0; i < 10; i++) {
        m_pCaObj->Measure(0); ①
        fLv = m_pProbeObj ->Lv; ②
        fx = m_pProbeObj ->sx;
        fy = m_pProbeObj ->sy;
        lT = m_pProbeObj ->T;
        fduv = m_pProbeObj ->duv;

        m_strLv.Format("%4.2f", fLv);
        m_strx.Format("%1.4f", fx);
        m_stry.Format("%1.4f", fy);
        m_strT.Format("%4d", lT);
        m_strduv.Format("%1.4f", fduv);
        UpdateData(FALSE);
    }
    ...
}

```

4-2-1-4-3 シンクオブジェクトの作成

CA-SDK は、CA 測定器で OCal が必要な状況になるとイベントにより、知らせる機能があります。これを利用する場合、アプリケーション側で、イベントを受け取るオブジェクト(以降シンクオブジェクトと記します)を持つ必要があります。

まず、サンプルソフトにシンクオブジェクトのクラスを作成します。

- シンクオブジェクトのクラスをプロジェクトに追加します。
 - ・クラス名:CCaEvent
 - ・基本クラス:CcmdTarget
 - ・オートメーションサポートを ON とする。
- サンプルソフトの odl ファイル中で、追加された dispinterface の uuid 指定値を、#import ディレクティブで生成された“CA200Srvr.tlh”中にある_IcaEvents 宣言部の uuid 指定値から、コピー・貼り付けし、リストにあるように変更します。
- クラス CcaEvent に、オートメーションメソッドを追加します。
 - ・外部名:ExeCalZero()
 - ・戻り値:void

これで、CA-SDK が受け付け可能なシンクインターフェース～シンクオブジェクトクラスとなります。

Ca200Sample.odl

```

...
[
    // CA-SDK
    // uuid(D497071B-3F9F-41C2-9E04-1E573571BF0D)
    // uuid(f7663750-5900-45eb-905f-78c5d5378481)
]
dispinterface ICaEvent
{
    properties:
        // メモ - ClassWizard はここでプロパティの情報を追加します。
        //   このセクションの編集は慎重に行ってください。
        //{{AFX_ODL_PROP(CCaEvent)
        //}}AFX_ODL_PROP

    methods:
        // メモ - ClassWizard はここでメソッドの情報を追加します。
        //   このセクションの編集は慎重に行ってください。
        //{{AFX_ODL_METHOD(CCaEvent)
        [id(1)] void ExeCalZero();
        //}}AFX_ODL_METHOD

};
...

```

- 新たに作成したシンククラス CCaEvent の実装ファイル中の IID_ICaEvent 定義も同じように変更(フォーマットが少し異なるので注意)します。また static 宣言子を外します。
- CCaEvent 定義ファイル中に、IID_IcaEvent の extern 宣言を追加します。

これで、これから作成するコードからこの IID を参照できるようになります。

CaEvent.cpp

```

...
// CA-SDK
// {D497071B-3F9F-41C2-9E04-1E573571BF0D}
// static const IID IID_ICaEvent =

```

```
// { 0xd497071b, 0x3f9f, 0x41c2, { 0x9e, 0x4, 0x1e, 0x57, 0x35, 0x71, 0xbf, 0xd } };
// {f7663750-5900-45eb-905f-78c5d5378481}
const IID IID_ICaEvent =
{0xf7663750, 0x5900, 0x45eb, {0x90, 0x5f, 0x78, 0xc5, 0xd5, 0x37, 0x84, 0x81}};
...
```

CaEvent.h

```
...
// CA-SDK
extern const IID IID_ICaEvent;

class CCaEvent : public CCmdTarget
{
    DECLARE_DYNCREATE(CCaEvent)
```

- CCaEvent::ExeCalZero() にリストに示すイベント発生時のコードを追加します。

サンプルソフトでは、イベント発生時、Measure ボタンを無効に設定します。

CaEvent.cpp

```
...
void CCaEvent::ExeCalZero()
{
    // TODO: ここにテキストパッチハンドラコードを追加してください。
    // CA-SDK
    CWinApp* papp = AfxGetApp();
    CCa200SampleDlg* pdlg = reinterpret_cast<CCa200SampleDlg*>(papp -> m_pMainWnd);

    CButton* pb;

    pb = (CButton *) (pdlg -> GetDlgItem(IDC_BUTTON_MSR));
    pb->EnableWindow(FALSE);
}

```

- シンククラスのコンストラクタを public なメンバーに変更します。

これで、これから作成するコードから直接、シンクオブジェクトを生成できるようになります。

CaEvent.h

```
class CCaEvent : public CCmdTarget
{
    DECLARE_DYNCREATE(CCaEvent)

    // CA-SDK
    // CCaEvent(); // 動的生成に使用されるプロテクト コンストラクタ

    // アトリビュート
public:

    // オペレーション
```

```
public:
    // CA-SDK
    CCaEvent();
    ...
}
...
```

4-2-1-4-4 シンクオブジェクトの生成・登録…イベント処理

- CCa200SampleDlg::OnInitDialog() 中の前に作成した初期化コードの後に、リストにあるシンクオブジェクトの生成・登録コードを追加します。

IConnectionPointPtr, IConnectionPointContainerPtr は、CA-SDK のイベント処理を実行するオブジェクトのスマートポインターです。これらは、MS が定義したいわゆる標準インターフェースに対応するものです。標準インターフェースのスマートポインターは、定義済みであり、そのヘッダーファイルが“CA200Srvr.tlh”に include されています。

コードでは①で、シンクオブジェクトを生成し、②でそのシンクインターフェースを得ています。④で、シンクインターフェースを登録するオブジェクト(IConnectionPoint インターフェース～コネクションポイント)を検出し、⑤でそのオブジェクトに登録しています。⑥では、後で登録を解除するときに必要な ID を保存します。

Ca200SampleDlg.cpp

```
BOOL CCa200SampleDlg::OnInitDialog()
{
    ...

    m_dwCk = 0;

    CCaEvent* pevntobj;

    if (NULL != (pevntobj = new CCaEvent)) { ①
        m_pIDispatch = pevntobj ->GetIDispatch(FALSE); ②

        IConnectionPointContainerPtr pcpcobj;
        DWORD dwck;

        pcpcobj = m_pCaObj; ③
        pcpcobj -> FindConnectionPoint(IID_ICaEvent, ¥
            &m_pIConnectionPointObj); ④
        m_pIConnectionPointObj ->Advise(m_pIDispatch, &dwck); ⑤
        m_dwCk = dwck; ⑥
    }
    ...
}
```

- CCa200SampleDlg::OnOK() 中に、リストに示す終了のコードを追加します。

ここでは、①で、CA-SDK の登録を解消し、②で、サンプルソフトのシンクオブジェクトの Release()を実行します。これにより、シンクオブジェクトは自らを削除します。サンプルソフトで保持している CA-SDK のオブジェクトは、そのスマートポインターが自動的に削除のための処理を実行してくれます。③では、Ca オブジェクトの RemoteMode プロパティをリモート OFF に設定し、CA がリモートモードを抜けるようにしています。

Ca200SampleDlg.cpp

```
void CCa200SampleDlg::OnOK()
```

```

{
    if (CanExit()){
        // CA-SDK
        if (m_dwCk != 0){
            m_pIConnectionPointObj ->Unadvise(m_dwCk);           ①
        }
        m_pIDispatch ->Release();                               ②
        m_pCaObj ->RemoteMode = 0;                             ③
        CDialog::OnOK();
    }
}

```

4-2-1-4-5 エラー処理

#import ディレクティブで生成されるスマートポインターは、COMのエラー処理もラップしています。CA-SDK 呼び出しが失敗した場合、_com_error タイプの例外オブジェクトをスローします。

サンプルでは、例外オブジェクトからエラー発生時の HRESULT の値、及び CA-SDK のエラーメッセージを取得し、メッセージボックスに表示します。

Ca200SampleDlg.cpp

```

BOOL CCa200SampleDlg::OnInitDialog()
{
    ...
    try{
        m_pCa200Obj = ICa200Ptr(__uuidof(Ca200));
        m_pCa200Obj->SetConfiguration(lcan, strenfig, lprt, lbr);
    }
    catch(_com_error e){
        CString strerr;
        strerr.Format(_T("HR:0x%08x¥nMSG:%s"), e.Error(), ¥
            (LPCSTR)e.Description());
        AfxMessageBox((LPCSTR)strerr);
        return TRUE;
    }
    ...
}

```

4-2-2 SDK アプリケーション作成法(補足説明) さらに詳しく

現在、多くのアプリケーション開発環境で、COM コンポーネントがサポートされていますが、そのサポートの仕方はいろいろであり、それに従って、アプリケーションの作成法も変わってきます。

MS の Visual Basic (以降、VB と記します) でプログラミングする場合、COM コンポーネントを使用するにあたり、特別意識することはありません。VB の COM サポート機能によって、VB 固有のオブジェクトを利用するのと同じ方法で、プログラムできます。それに対し、C/C++ でプログラミングする場合、他のアプリケーションを組む場合と同様に、いくつかの組み方があり、それを設計者が選択することになります。

MS Visual C++ (以降、VC++ と記します) の場合、その選択肢のひとつとして、VC++ コンパイラの拡張機能… #import ディレクティブを利用する方法があります。この方法により、VB と同程度に簡単にプログラムが組めます。また、これはコンパイラの機能であるため、MFC/ATL 等のツールが提供するフレームワークを使用しないアプリケーションでも、利用できます。VC++ で作成した既存の“手作り” VC++ アプリケーションを再利用し、CA-SDK を組み込む場合、プログラミングを容易にしてくれます。

4-2-2-1 アプリケーションプロジェクトの作成

サンプルソフトは MFC を利用した“ダイアログベース”のアプリケーションとします。

MFC アプリケーションの開発手順は、概略以下のとおりと説明されています。

手順1. プロジェクトの生成

手順2. UI の作成

手順3. UI オブジェクトとコードの対応付け

手順4. アプリケーション固有のコードの作成

手順5. コンパイル・デバッグ

ここでは、VC++ のツールであるアプリケーションウィザードを利用して、最初の手順…プロジェクトの作成をします。ウィザードの質問に答えていくと、所要のオプションが設定されたコードのスケルトンが自動生成されます。

サンプルソフトの場合、設定は以下のとおりとしました。

① VC++ を起動、**ファイル** - **新規作成** を選択 (アプリケーションウィザードが起動)、表示される“新規作成”のダイアログで、

1. “プロジェクト”タブを選択

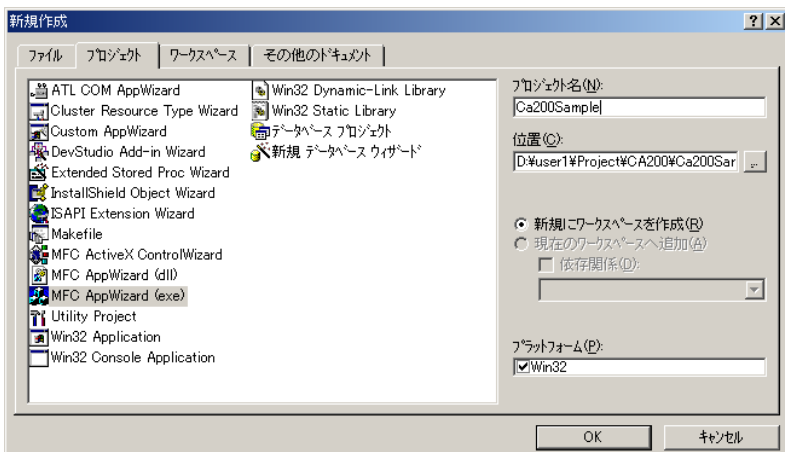
2. プロジェクトの種類一覧表示から、“MFC AppWizard(exe)”を選択

3. “位置”に、作成するプロジェクトのフォルダーを指定

4. “プロジェクト名”に、作成するプロジェクトの名前をして

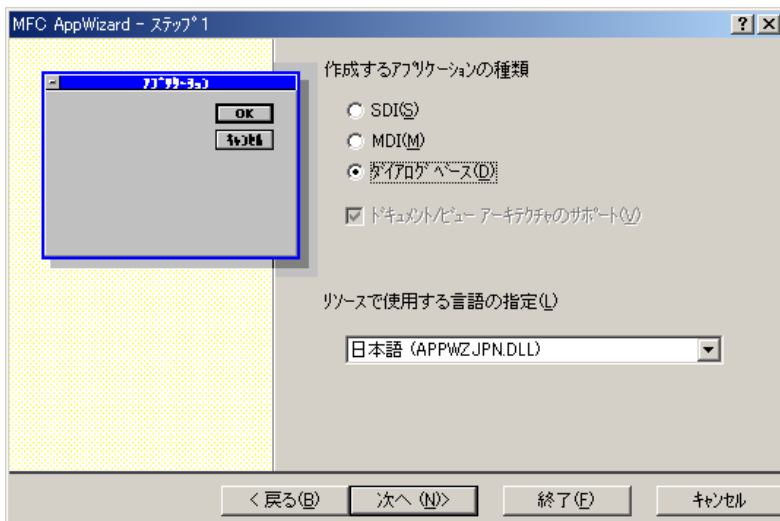
5. その他の設定は、デフォルト (“新規にワークスペースを作成”選択、“プラットフォーム”の“Win32”チェック状態) のまま

を設定し、**OK** を選択。

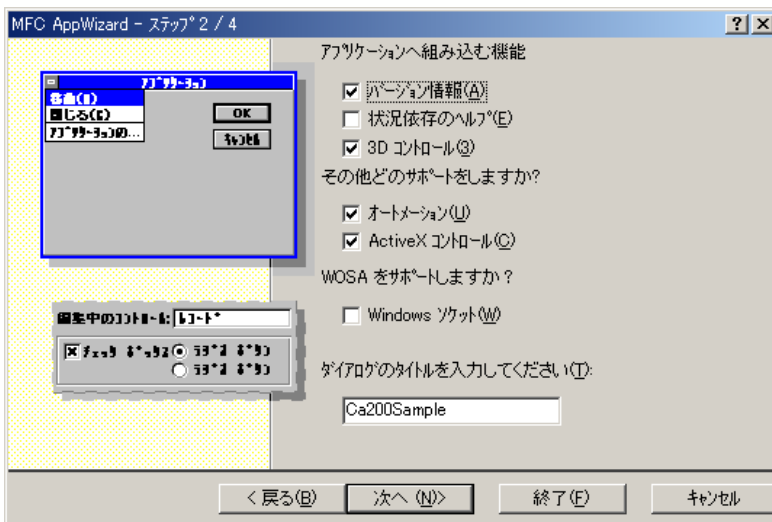


- ②表示される“MFC AppWizard-ステップ1”で
- “作成するアプリケーションの種類”を“ダイアログベース”に設定。
 - “リソースで使用する言語”はデフォルト(日本語)のまま。

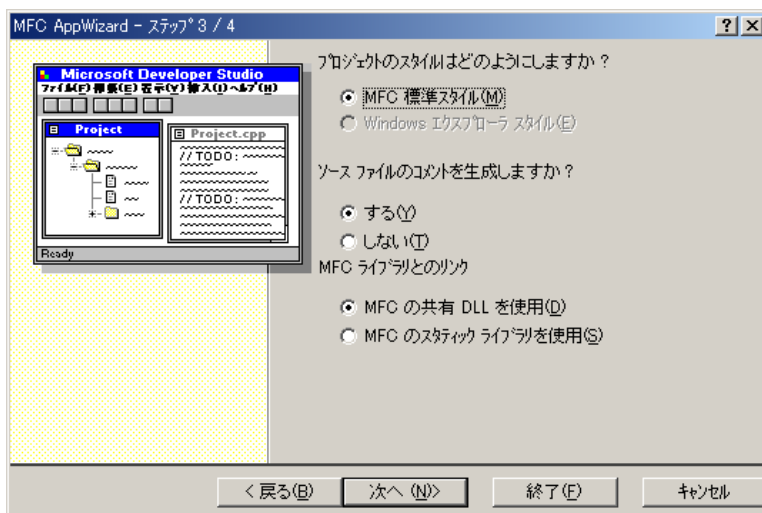
にして、**次へ**を選択。



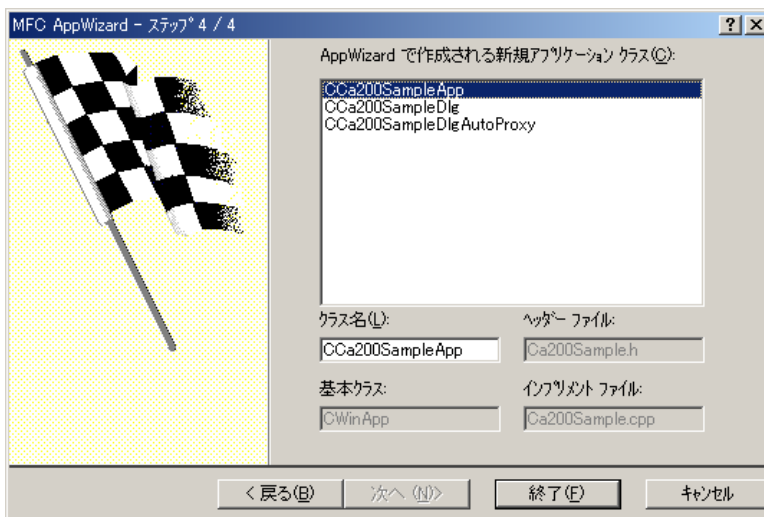
- ③表示される “MFC AppWizard-ステップ 2/4” では、
- “その他のサポートをしますか？” で、“オートメーション” のチェックを入れる。
 - それ以外、デフォルト (“バージョン情報”、“3D コントロール”、“ActiveX コントロール” のチェック、ダイアログタイトルは、自動生成されたもの)。の設定をして、**次へ**を選択。



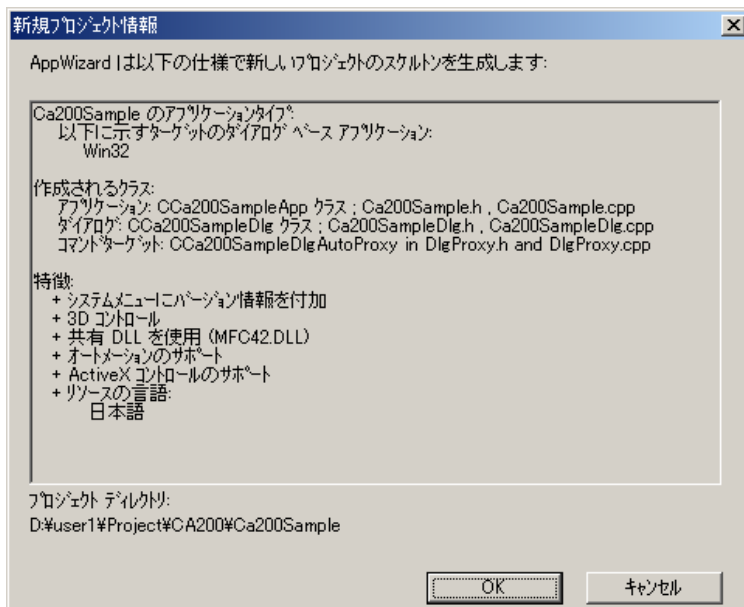
④表示される“MFC AppWizard-ステップ 3/4”では、デフォルトの設定(“プロジェクトのスタイルはどのようにしますか?”は“MFC 標準スタイル”、“ソースファイルのコメントを生成しますか”は“する”、“MFC ライブラリーとのリンク”は“MFC の共有 DLL を使用”)のままとし、**次へ**を選択。



⑤表示される“MFC AppWizard-ステップ 4/4”の表示内容を確認の上、**終了**を選択。



下に示す“新規プロジェクト情報”ダイアログが表示されます。**OK**を選択すると、アプリケーションウィザードにより、指定したオプションに従って、プロジェクトが生成されます。



4-2-2-2 UI の作成

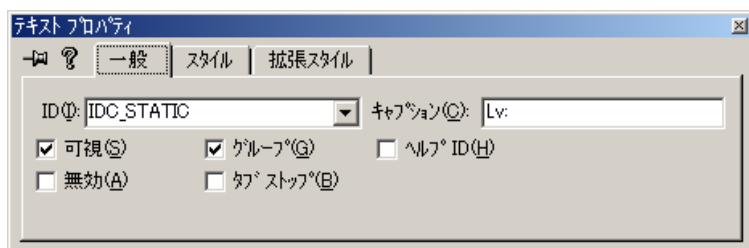
手順2. UI の編集には、リソースエディターを使用します。

ワークスペースの“ResourceView”を選択、“Dialog”フォルダーを展開し、アプリケーションのメインウィンドーとなるダイアログ (IDD_CA200SAMPLE_DIALOG) をダブルクリックすると、“OK”・“キャンセル”ボタンを備えたデフォルトのダイアログ外観が表示されます。

編集は、VB と同じようにできます。エディター画面のツールコントロールボックスから、所要のコントロールをドラッグ & ドロップします。コントロールの位置・サイズの変更の仕方も VB とかわりません。

大体の設定を終えたら、コントロール・イメージ上で右クリックし、表示されるメニューから「プロパティ」を選択、表示されるダイアログで、プロパティの編集をします。

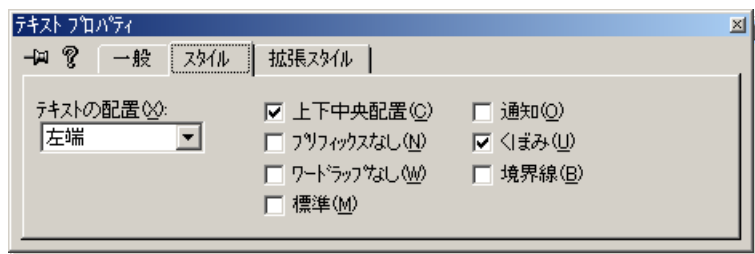
“Lv:”のラベル表示用のコントロールでは、“キャプション”に“Lv:”を設定します。



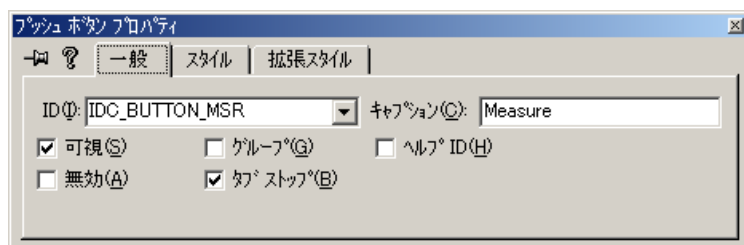
他のラベル用コントロールも同じように設定します。

測定値表示用のコントロールは、後でコードからアクセスするため、分かりやすい ID に変更しました。

また、枠にくぼみを持たせる外観になるよう設定しています。



プッシュボタンも、ID とキャプションを変更します。



最後は、メニューの「レイアウト」を利用し、コントロールのサイズ・配置を調整、全体の外観を整えます。

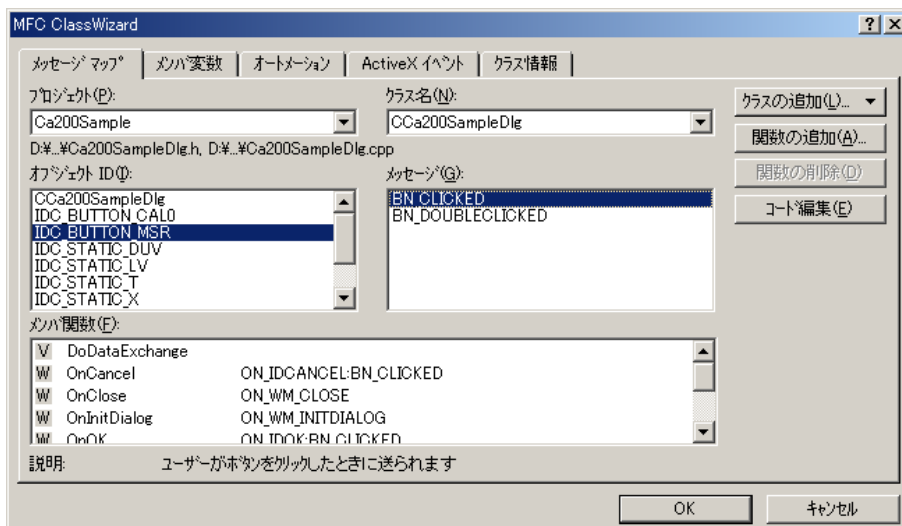
4-2-2-3. UI オブジェクトとコードの対応付け

VB の場合、UI オブジェクトとコードとの対応付けは容易です。フォーム (モジュール) を選択、コード編集モードにすると、

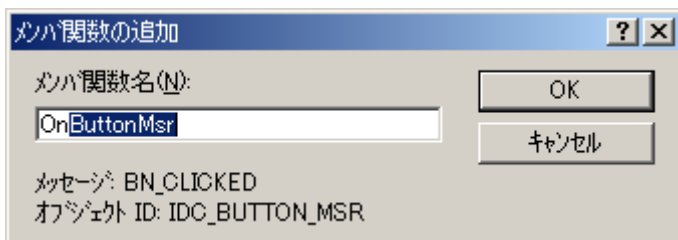
オブジェクト・ボックスとプロシージャ・ボックスがあらわれます。オブジェクト・ボックスで、対象のオブジェクトを選択すると、プロシージャ・ボックスにオブジェクトが処理可能なイベントがリストアップされ、イベントを選択すれば、それを処理するプロシージャの定義が、モジュールファイル中に自動生成されます。あとは、そのプロシージャの中味を作成するだけです。

MFC の場合、これと同じようなことを“クラスウィザード”を使って行います。

メニュー **表示** - **ClassWizard** を選択すると下に示すクラスウィザードが表示されます。



“メッセージマップ”タブを選択し、“クラス名”でメッセージを処理するオブジェクトクラス…サンプルソフトでは、CCa200SampleDlg を選択します。“オブジェクト ID”にコントロール (ID) などのオブジェクトリスト、“メッセージ”に VB のイベントに対応するメッセージリストが表示されます。VB 同様、オブジェクト…Measure ボタンの場合 IDC_BUTTON_MSR、を選択し、その後、処理するメッセージ…BN_CLICKED、を選択してから **関数の追加ボタン** をクリックすると、下の確認のダイアログが表示されます (メッセージリスト中で、強調表示されているメッセージは既に、関数が存在することを示します)。



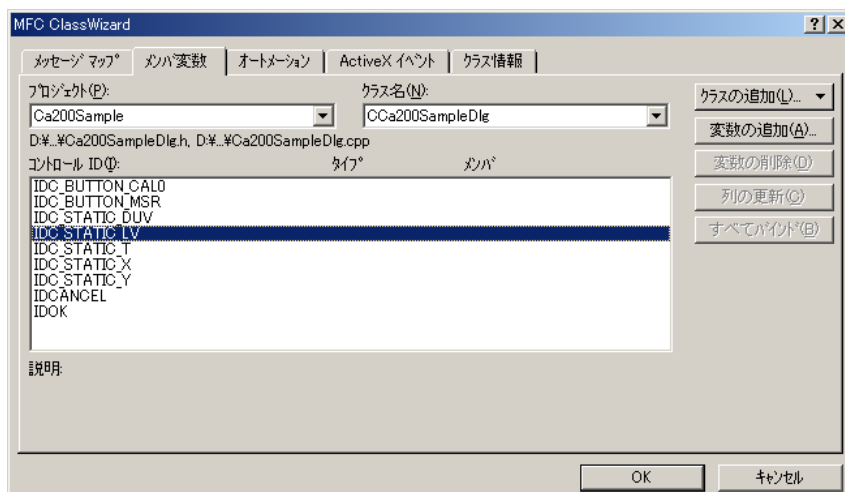
OK をクリックすると、必要なコードが自動的に関連するファイルに挿入されます。あとは、VB 同様、生成された関数の中味を作成します。

Cal Zeor ボタンも同様に BN_CLICKED メッセージハンドラーを追加します。

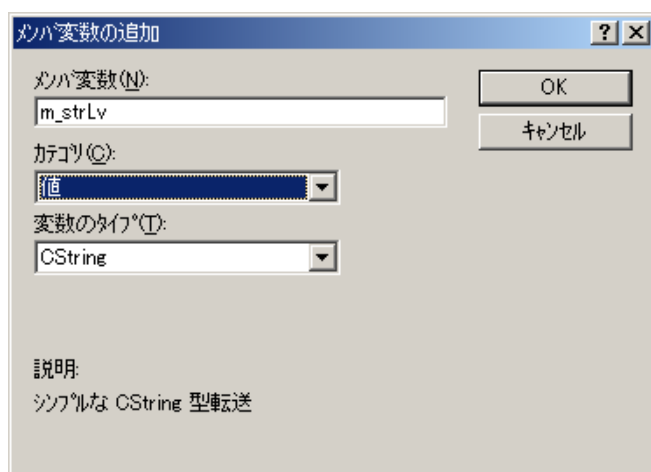
クラスウィザードは、上に記したメッセージハンドラーの作成の他にも、様々なケースで使用します。サンプルソフトでは、MFC の DDX (Dialog Data Exchange) の機能を利用します。これは、ダイアログにのせたコントロールをダイアログのメンバー変数に対応付けるものです。この設定も、クラスウィザードを使用します。

サンプルソフトでは測定値表示用のコントロールにメインウィンドとなるダイアログのメンバー変数を対応付けます。クラスウィザードを表示し、“メンバー変数”タブを選択します。“クラス名”で、設定対象の CCa200SampleDlg クラスを選択します。“表示されるコントロール ID”から、Lv 測定値表示用のコントロール ID “IDC_STATIC_LV” を選択し、**変数の追加** をクリックしま

す。



下に示す“メンバ変数の追加”ダイアログが表示されます。“メンバ変数”のボックスに、メンバー変数名 `m_strLv` を設定、“カテゴリ”を“値”、“変数のタイプ”を“CString”として、**OK** をクリックします。



これにより、`IDC_STATIC_LV` の ID を持つスタティックテキストコントロールの値～キャプションに/をサンプルソフトのダイアログに対応する C++ クラス `CCa200SampleDlg` のメンバー `m_strLv` の値を/に設定/取得する機能が自動生成されます。

他の測定値表示用コントロールにも、同様に設定します。

4-2-2-4 CA-SDK のプログラミング

4-2-2-4-1 CA-SDK オブジェクト生成

VC++ では COM コンポーネントのクラス・インターフェースの定義には MIDL が使用されます。これで記述した定義ファイルを専用のコンパイラでコンパイルすると、COM コンポーネントを実装するのに必要ないくつかのファイルが作成されます。その一つがタイプライブラリです。タイプライブラリは MIDL で定義したクラス/インターフェース定義のバイナリに対応します。タイプライブラリは単独のファイルである場合もあれば、リソースとしてコンポーネントファイルに含まれる場合もあります(コンポーネント情報がレジストリに登録されている場合、“OLE View”を使って、タイプライブラリファイルを得ることができます)。COM コンポーネントが“ActiveX コンポーネント”と呼ばれるようになってからは、リソースとして含むことが前提となっています。

このタイプライブラリを利用して、開発ツールはいろいろな COM サポート機能を実現しています。VC++ のインテリセンス機能もその一つです。(アプリケーション開発者にとって、インテリセンス機能は大きな手助けとなります。これと MSDN ライブラリの膨大なヘルプさえあれば、それなりの効率でプログラミングを進めていくことができます)。

VC++ コンパイラの `#import` ディレクティブもタイプライブラリを利用します。タイプライブラリをアプリケーションのソ

ースに取り込み、コンポーネントを利用するための C++クラスを自動生成します。

#import ディレクティブにはいろいろなオプションがあります。サンプルソフトでは、名前空間なし…no_namespace、とし、また、ヘッダー部…no_implementation、と実装部…implementation_only、を別々に生成するオプションとしています。

#import ディレクティブで生成される C++クラス…ICa200Ptr 等のスマートポインターは、COM インターフェースポインターと同じように使用できます。ただし、通常の COM インターフェースポインターによる操作と違って、AddRef()/ReleasRef()/QueryInterface()等を明示的にコーディングする必要はありません。

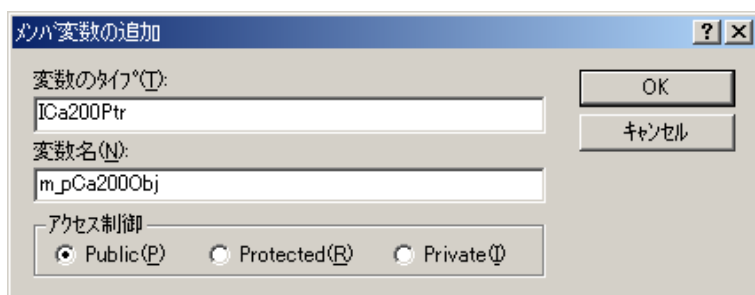
HRESULT 型の戻り値の明示的な処理も必要ありません。メソッドの戻り値は、インターフェースのメソッド定義で retval と指定したパラメータを戻すものとして使用できます。

propget/propput と指定した“プロパティ”は、VB と同様の記述で、プロパティとして使用できます。

これらのおかげで、VB と同程度にシンプルで、自然な形で記述できます。特に、1-4-5. で説明したエラーサポート機能とあいまって、アプリケーションのエラー処理を大変シンプルにしてくれます。ネイティブなインターフェースポインターを使用する場合のコードを想像すれば、よさが実感できるでしょう。

サンプルソフトでは、#import ディレクティブで生成される C++クラスのメンバーをメインウィンドクラス CCa200SampleDlg に追加します。既存のクラスに単に関数/データメンバーを追加するにはクラスビューの機能を使用します。

ワークスペースで“Class View”を選択します。対象のクラス…CCa200SampleDlg 上で右クリックするとポップアップメニューが表示されます。**メンバー変数の追加**をクリックすると下の“メンバ変数の追加”ダイアログが表示されます。“変数のタイプ”に追加する変数のタイプ…ICa200Ptr、“変数名”に m_pCa200Obj を入力、“アクセス制御”を Public に設定して **OK** ボタンをクリックします。



これで、指定したメンバー変数 m_pCa200Obj がメインウィンドクラス CCa200SampleDlg に追加されます。他のメンバー変数も同様に追加します。

4-2-2-4-2 CA-SDK オブジェクト利用

CA-SDK オブジェクトの生成/初期化が終了したら、それを利用して、測定/校正等が実行できます。

CA-SDK オブジェクトの利用法は、CA-SDK の“プログラミングガイド”に説明されています。プログラミングガイドでは VB を使用していますが、本サンプルソフトのコードを参考にいただければ、C++コードにすることは容易かと思えます。[\(「4-4 CA-SDK VB サンプルソフト」参照\)](#)

4-2-2-4-3 シンクオブジェクトの作成

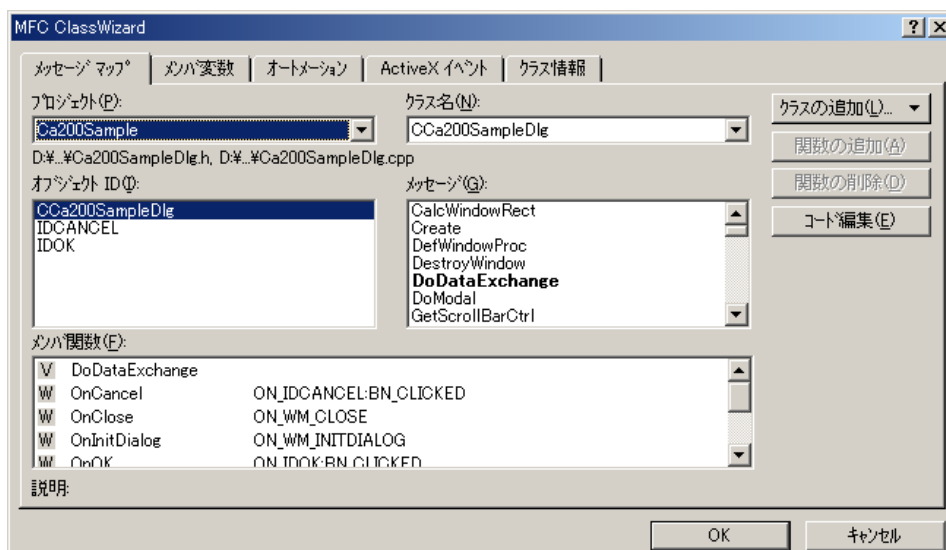
COM イベントは、COM サーバーからクライアントが公開するメソッドを呼び出すことによって実現されます。これにより、クライアント・サーバー間に、双方向の通信が確立されます。イベント機能は ActiveX コントロールなど GUI をもつ COM コンポーネントでは不可欠の機能となります。イベントという機能の観点にたつと、COM サーバーはクライアントであり、アプリケーションがサーバーになります。

CA-SDK のイベントは、ディスプレイインターフェースを使用します。MFC はオートメーションをサポートしており、オートメーションサーバも比較的容易に実現できます。サンプルソフトでは MFC のこのサポート機能を利用して、シンクオブジェクトを実装します。

シンクオブジェクトクラスの作成にはクラスウィザードを使用します。

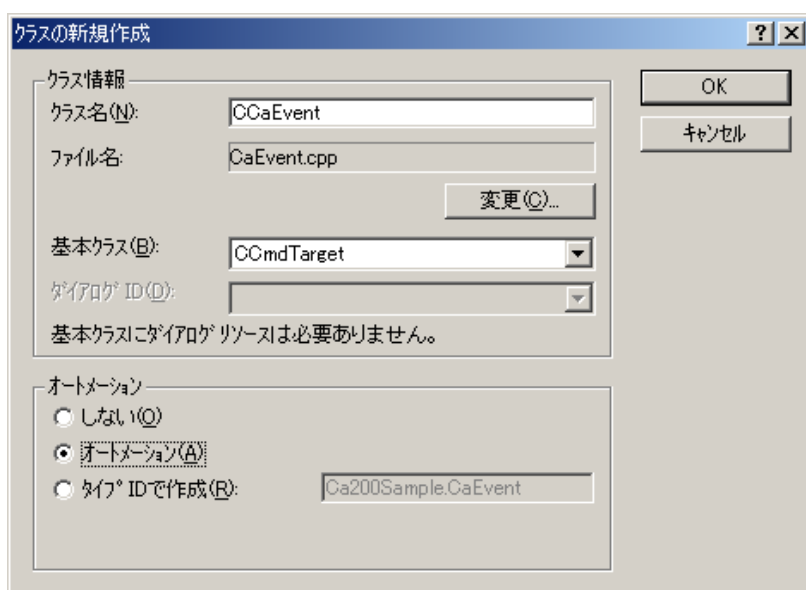
メニュー **表示** **ClassWizard** を選択し、クラスウィザードを表示します。**クラスの追加**をクリックし、表示される項目から、**新**

規をクリックします。



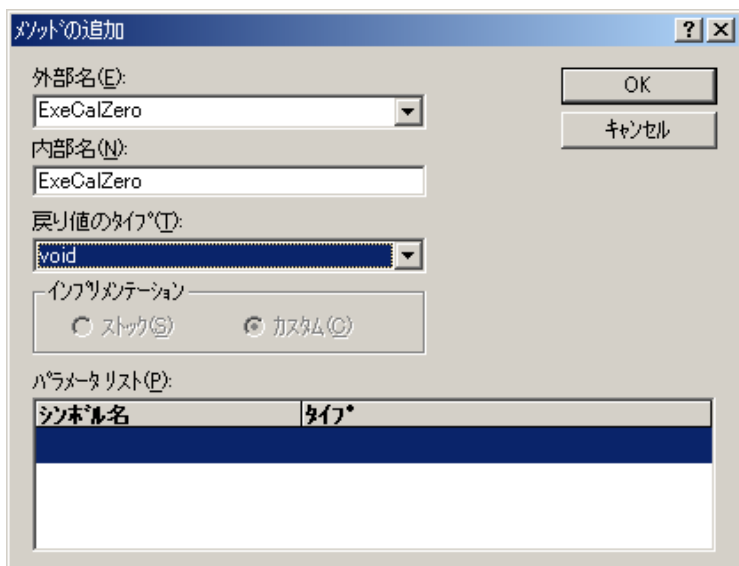
次に示す“クラスの新規作成”ダイアログが表示されます。“クラス名”にオートメーションクラス名…CCaEvent、“基本クラス”に CCmndTarget を設定します。基本クラスを CCmndTarget とすると、“オートメーション”のオプションが有効になります。“オートメーション”を選択し、**OK** ボタンをクリックします。

これで、ツールがオートメーション実装クラス…CCaEvent を生成してくれます。アプリケーションのインターフェース定義ファイル(“Ca200Sample.odl”)には、coclass(～COM クラス)CCaEvent と、そのデフォルトの dispinterface ICaEvent の定義が追加されます。それぞれに自動生成されたクラス ID・インターフェース ID が割り付けられます。



引き続き、クラスウィザードを利用して、イベント・メソッドを追加します。“オートメーション”タブを選択、“クラス名”で今生成した CCaEvent を選択、**メソッドの追加** ボタンをクリックします。次に示す“メソッドの追加”ダイアログが表示されます。

外部名に、CA-SDK のイベント名…ExeCalZeroを入力、戻り値・パラメータリストはイベント定義どおり…void・パラメータなし、に設定して **OK** ボタンをクリックします。



これで、ツールによるシンクオブジェクトの定義は一応完了です。

ツールが必要な。odl ファイルと C++ファイル・コードを生成してくれます。

生成したシンクオブジェクトを CA-SDK のイベントシンクにするためには、自動生成された定義を手作業で一部修正する必要があります。ツールが自動的に割り当てた IID を CA-SDK 規定の値に設定します。COM インターフェースにとって、そのアイデンティティは GUID～uuid でしか識別できません。

ツール割り当ての IID を検索でリストアップし、全て CA-SDK が認識するイベントの IID に置き換えます。これにより、生成されたオブジェクトは、CA-SDK が認識するイベントシンクになります。

その他、1-4-3. で説明されている ExeCalZero()の実装以外の残りの変更は、このシンクオブジェクトをローカルに生成し利用するのに必要な変更です。通常、サーバーとして動作する場合、公開するオブジェクトは、クラスファクトリーという匿名のオブジェクトによって生成します。MFC の COM クラスの実装もそれに対応するものになっています。サンプルソフトではローカルなオブジェクトとして操作するため、自動生成された定義/実装を一部変更しています。

4-2-2-4-4 シンクオブジェクトの生成・登録…イベント処理

CA-SDK のイベントは、MS がイベント処理用に定義した標準のインターフェース IConnectionPoint…コネクションポイントを利用しています。コネクションポイントオブジェクトにシンクインターフェースを登録すると、対応するイベントがサーバから発行…シンクインターフェースのメソッドが呼び出し、されます。

シンクインターフェースの登録/解除は、通常、下記の手順でなされます。

① イベントをサポートするオブジェクトからコネクションポイントのコンテナオブジェクト～IConnectionPointContainer を得る。

② IConnectionPointContainer の FindConnectionPoint()メソッドにより、シンクインターフェースをサポートするコネクションポイントオブジェクト～IConnectionPoint を得る。

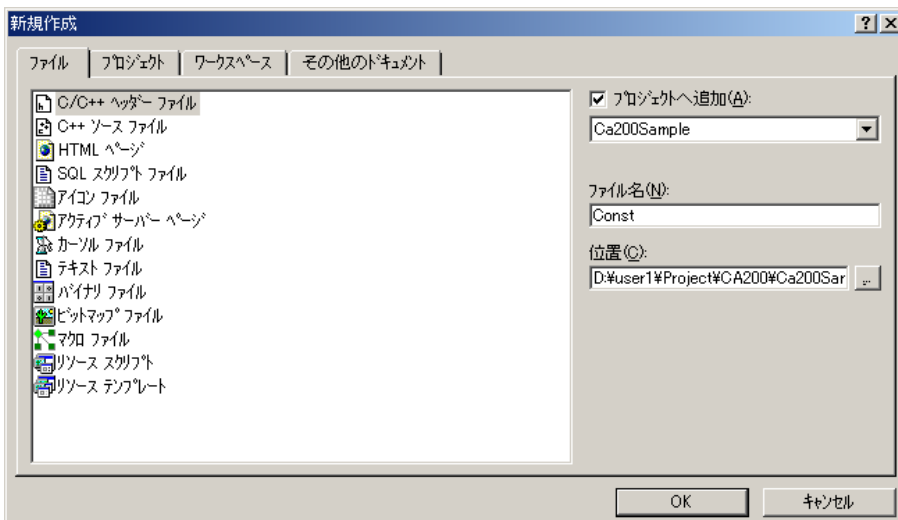
③ IConnectionPoint の Advise()メソッドにより、シンクオブジェクトを登録する。

④ イベント処理が不要になったら、IConnectionPoint の UnAdvise()メソッドにより、シンクオブジェクトの登録を解除する。

4-2-2-4-5 その他

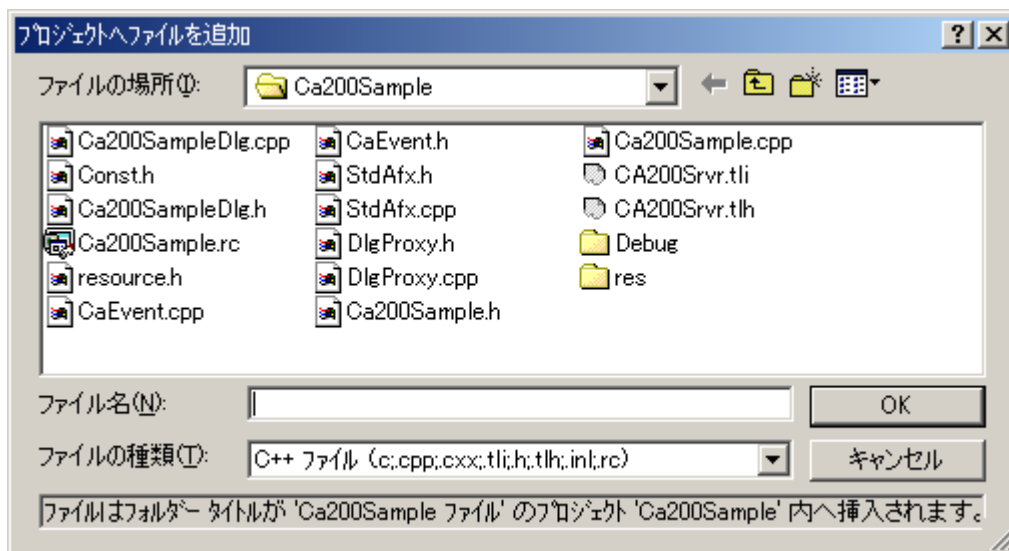
サンプルソフトでは、コードで使用するシンボルを“Const.h”で定義しプロジェクトに追加しています。

プロジェクトに新規ファイルを作成する場合、ファイル—新規作成で表示される“新規作成ダイアログ”を使用します。



“ファイル”タブを選択、作成するファイルタイプをリストから選択して、“プロジェクトへ追加”をチェック、サンプルソフトのプロジェクト…“Ca200Sample”、を選択し、“ファイル名”に追加ファイルの名前～“Const”を入力して **OK** をクリックします。

既存のファイルをプロジェクトに追加する場合は、ワークスペースの“File View”を選択、プロジェクト“Ca200Sample”を右クリックし、表示されるメニューから **ファイルをプロジェクトへ追加** をクリックします。表示される“プロジェクトへファイルを追加”ダイアログボックスで、追加するファイルを選択、**OK** をクリックします。



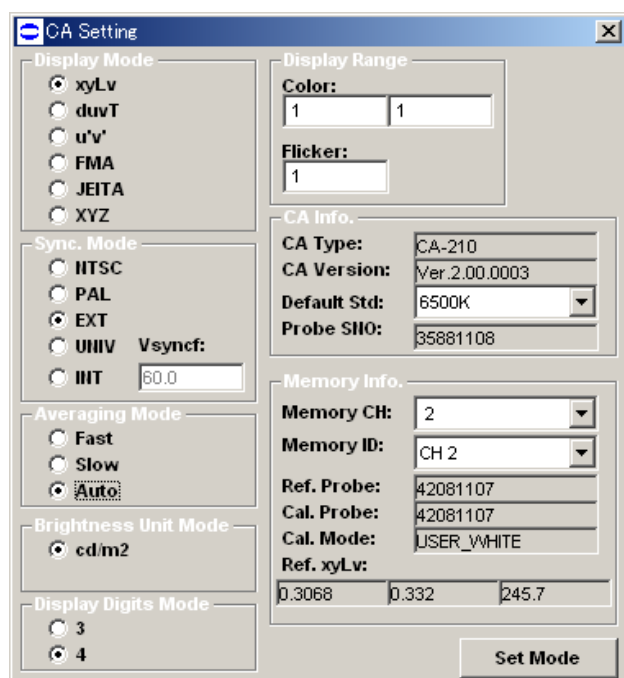
4-3 CA-SDK サンプルソフト用コントロールの使用法

CA-SDK の VB サンプルソフトには3つのサンプル ActiveX コントロールが添付されています。いずれもサンプルソフト用に作成されたものですが、他のソフトで利用することもできます。

ここでは、そのうちの CA 測定器の各種セッティングを行うコントロール CaControl と、測定結果をxy色度図表示するコントロール xyControl について、使用法を説明します。

4-3-1 CaControl (CA 測定器セッティング用コントロール)

CaControl の "... ボタンをクリックすると、下に示す”CA Setting”ダイアログが表示されます。このダイアログを使って、CA 測定器の各種設定をすることができます。



4-3-1-1 CaControl のプロパティ・メソッド

Ca プロパティ

設定/取得プロパティです。

セッティング対象のCaオブジェクトを設定/取得します。

コントロールを使用する前に、このプロパティにより、制御するCaオブジェクトを設定します。

構文:

```
Dim objCaControl as CaControl
```

```
Dim objCa as Ca
```

```
Set objCaControl.Ca = objCa
```

Probe プロパティ

設定/取得プロパティです。

セッティング対象のProbeオブジェクトを設定/取得します。

コントロールを使用する前に、このプロパティにより、制御するProbeオブジェクトを設定します。

構文:

```
Dim objCaControl as CaControl
```

```
Dim objProbe as Probe
```

```
Set objCaControl.Probe = objProbe
```

Memory プロパティ

設定/取得プロパティです。

セッティング対象のMemoryオブジェクトを設定/取得します。

コントロールを使用する前に、このメソッドによりコントロールを使用して制御するMemoryオブジェクトを設定する。

構文:

```
Dim objCaControl as CaControl
```

```
Dim objMemory as Memory
```

```
Set objCaControl.Memory = objMemory
```

UpdateCaInfo メソッド

コントロールに現在のCA測定器の状態(メモリチャンネルを除く)を反映させます。

コントロールに必要なCA-SDKオブジェクトを設定した後、コントロールを使用する前に、一度はこのメソッドを実行してください。

構文:

```
Dim objCaControl as CaControl
```

```
objCaControl.UpdateCaInfo
```

UpdateMemoryInfo メソッド

コントロールに現在のCA測定器のメモリチャンネルの状態を反映させます。

コントロールに必要なCA-SDKオブジェクトを設定した後、コントロールを使用する前に、一度はこのメソッドを実行してください。

構文:

```
Dim objCaControl as CaControl
```

```
objCaControl.UpdateMemoryInfo
```

Update イベント

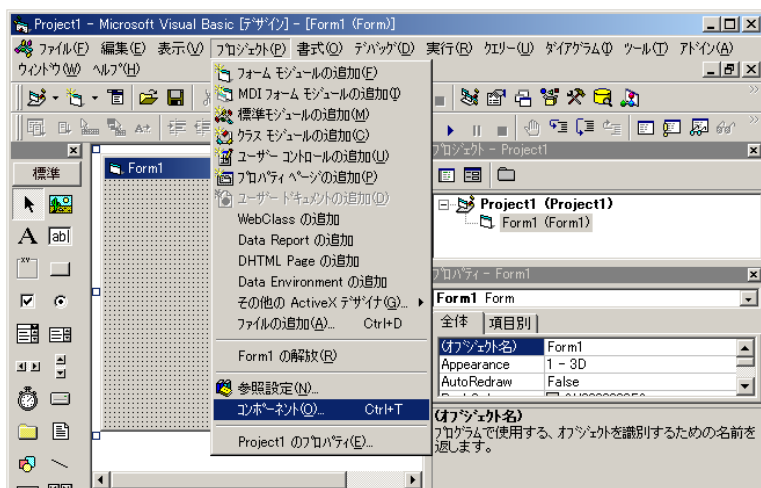
コントロールにより、CA測定器のセッティングが変更されたとき、発行されます。

4-3-1-2 VB での利用法

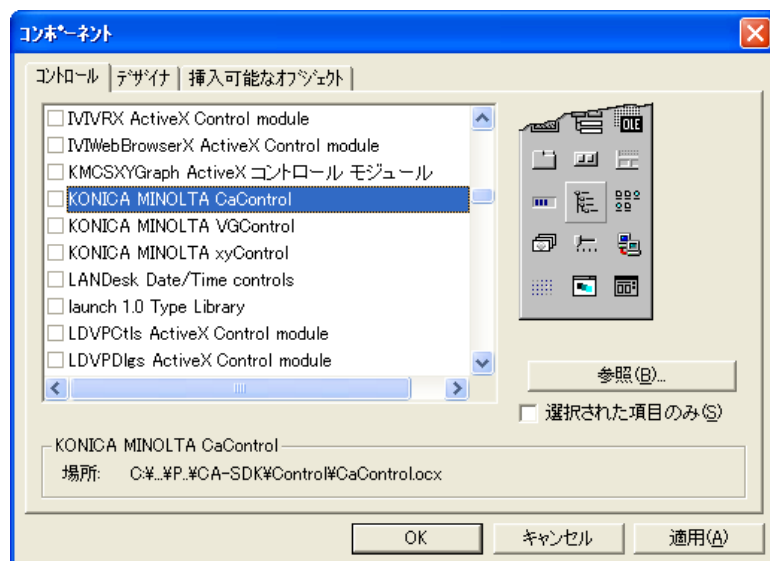
最初に、CA-SDK コンポーネントの参照設定により、VB プロジェクトに CA-SDK コンポーネントを追加します(参照設定の方法は CA-SDK プログラミングマニュアルにあります)。

次に、以下の手順で、コントロールを VB プロジェクトに追加します。

①メニューでプロジェクトコンポーネントを選択



②表示される“コンポーネント”ダイアログの“コントロール”タグで、“KONICA MINOLTA CaControl”をチェックし、“OK”をクリック



コントロールの追加が終わると、フォームエディターのツールボックスに、コントロールアイコンが表示され、編集できる状態になります。

下に、CA-SDK の色測定サンプルソフトから抜き出したソースリストを示します。

SDK のオブジェクトを生成した後、コントロールのプロパティにより、①②③で CA 測定器を制御するためのオブジェクトをコントロールに設定します (FormMeasurement は、コントロールを貼り付けたフォームです)。

その後、④UpdateCaInfo⑤UpdateMemoryInfo で、コントロールに現在の CA 測定器の状態により、コントロールを初期化します。

以上の操作で、コントロールが使用できる状態になります。

Update イベントを利用する場合、コントロールを貼り付けたフォームのモジュール宣言部で、WithEvents キーワード付で

コントロール変数を宣言します。コード編集モードのオブジェクトボックスでコントロールオブジェクトを選択し、プロシージャボックスで、“Update”プロシージャが選択できる状態になります。CaControl は、CA 測定器のセッティングを変更すると、この Update ハンドラーを呼び出します。

Ca200Sample.bas- Main()

```

...
Set objCa200 = New Ca200
objCa200.AutoConnect
...
Set objCa = objCa200.SingleCa
Set objProbe = objCa.SingleProbe
Set objMemory = objCa.Memory
...
Set FormMeasurement.objCaControl.Ca = objCa ①
Set FormMeasurement.objCaControl.Probe = objProbe ②
Set FormMeasurement.objCaControl.Memory = objMemory ③
...
FormMeasurement.objCaControl.UpdateCaInfo ④
FormMeasurement.objCaControl.UpdateMemoryInfo ⑤
...

```

FormMeasurement.frm

```

Public WithEvents objCaControl As CaControl

Private Sub CaControl1_Update()
...
End Sub

```

4-3-1-3 VC++(MFC)での利用法

MFC アプリケーションでの利用法をサンプルソフトで説明します。

“CA-SDK のアプリケーション作成方法”では、VC++コンパイラーの COM サポート機能を使用しましたが、ここでは、MFC のサポート機能を利用して作成することにします。

サンプルソフトは下に示す GUI をもつダイアログベースのアプリケーションとします。“OK”/“キャンセル”ボタンの左にあるのが今回利用する CaControl です。”…“ボタンをクリックすると、前記の”CA Setting“ダイアログが表示されます。

プロジェクト生成時、ActiveX コントロールのサポートオプションをつけてください。

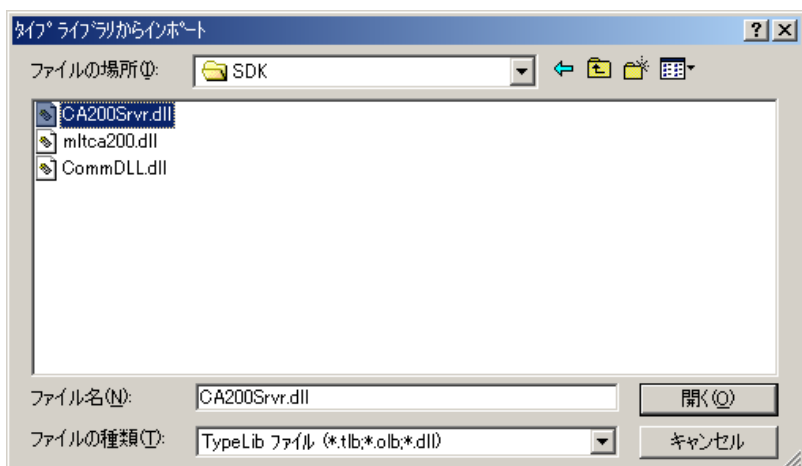


最初に、CA-SDK コンポーネントをプロジェクトに追加します。
以下の手順で、CA-SDK コンポーネントクラスをプロジェクトに追加します。

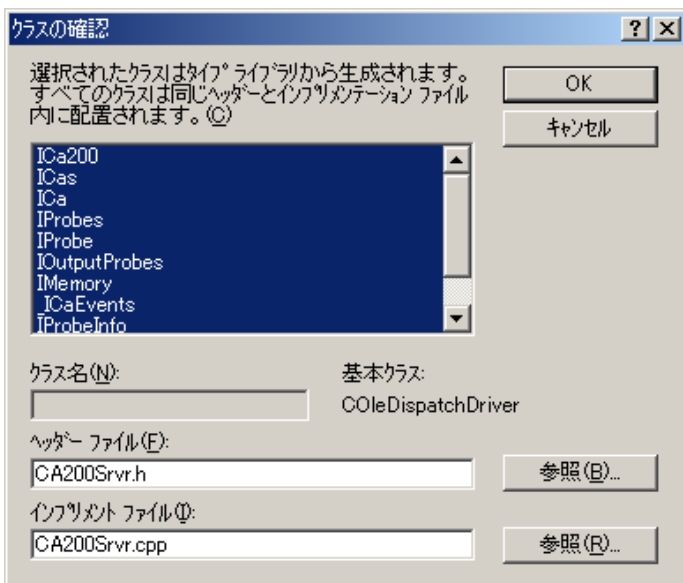
①メニュー **表示** - **ClassWizard** を選択、クラスウィザードを表示する。

②クラスウィザードの“クラスの追加”ボタンをクリック、ポップアップメニューから **タイプライブラリから** を選択。

③表示される“タイプライブラリからインポート”ダイアログで、CA-SDK のコンポーネントファイル“CA200Srvr.dll”を選択、“開く”をクリック。



④下のクラスの確認ダイアログが表示されるので、確認の上、“OK”をクリック。②のクラスウィザード表示画面に戻るので、“OK”をクリック。

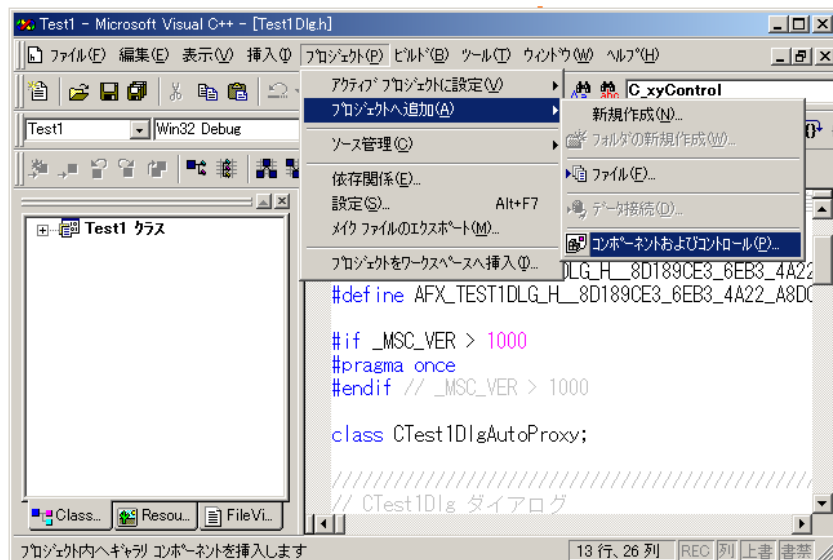


これで CA-SDK コンポーネントの追加は終了です。CA-SDK オブジェクトを利用するための C++ラップクラス ICa200/ICas...が自動生成されます。

ここでビルドすると、ツールが自動生成した“ca200Srvr.h”で定義されている Imemory クラスで、SetChannelID()が2重定義されているというエラーが発生します。これは、CA-SDK の Memory オブジェクトの ChannelID プロパティに対して、ツールが自動生成した設定用のラップメソッドの名前(SetChannelID)がもともとある Memory オブジェクトの SetChannelID メソッド名と衝突するためです。どちらかの名前を変更してください。サンプルソフトでは、もともとあるメソッドの名前を SetID に変更しています。

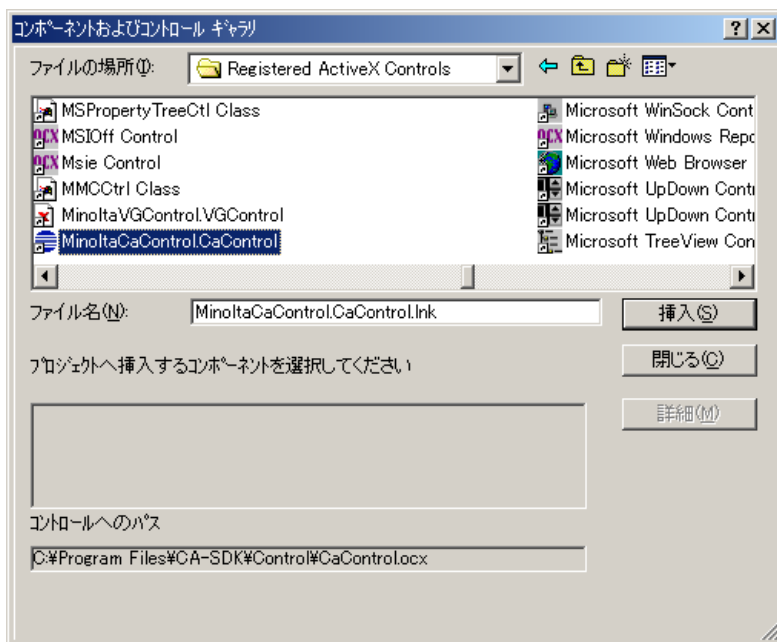
次に、MFC アプリケーションでコントロールを利用するため、以下の手順で、MFC プロジェクトにコントロールを追加します。

①メニュー「プロジェクト」→「コンポーネントおよびコントロール」を選択。

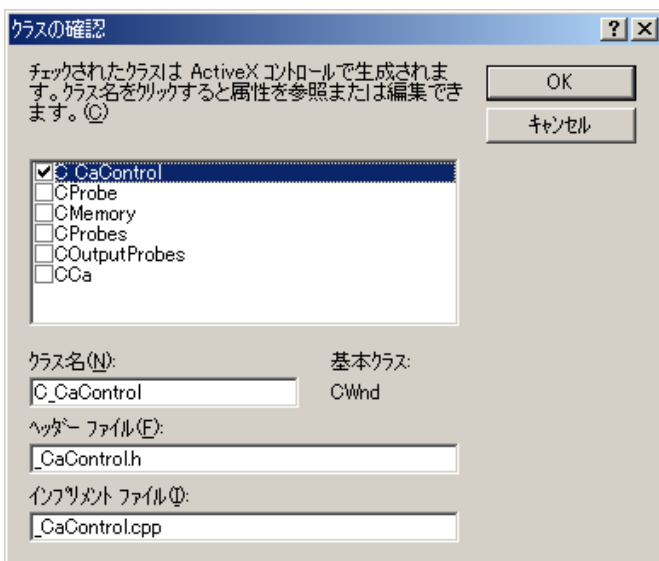


②表示される“コンポーネントおよびコントロールギャラリー”ダイアログで“Registered ActiveX Controls”フォルダーから

“MinoltaCaControl.CaControl”を選択、挿入をクリック。



③挿入確認のダイアログで、OKをクリックすると、下のクラスの確認ダイアログが表示される。C_CaControl だけチェック状態にして、OKをクリックする。②の状態に戻るので、閉じるをクリックする。



これで、CaControl が使用できる状態になります。

リソースエディターのツールボックスには CaControl のアイコンが追加されます。また、③で確認したコントロールを制御するための C++クラス C_CaControl が自動生成されます。

メインウィンドーにコントロールを貼り付け、GUI を作成します。

この後、

- ①コントロールクラスの DDX 用メンバー変数 m_CCaControl をダイアログクラスに追加
- ②CA-SDK オブジェクトクラスのメンバ変数 m_ICa200、m_ICa、m_Iprobe、m_IMemory をダイアログクラスに追加して、コー

デイングの準備は終わりです。

下に、コントロール利用のためのサンプルソフトのソースリストを示します。

①で、Ca200 オブジェクトを生成します。

②で、生成した Ca200 オブジェクトの (SingleCa プロパティから) GetSingleCa メソッドにより Ca オブジェクトを得、③で、Ca オブジェクトに対応するラップクラス ICa のインスタンスにアタッチします。以降、ICa のインスタンスを Ca オブジェクトのように操作できます。

同様に、Probe オブジェクト、Memory オブジェクトの設定をします。ここまでが、CA-SDK オブジェクトの初期化の部分です。

その後、④⑤⑥で、CaControl のプロパティに Ca オブジェクト、Probe オブジェクト、Memory オブジェクトを設定します。

最後に、コントロールの⑦UpdateCaInfo、⑧UpdateMemoryInfo メソッドにより、CaControl コントロールを初期化します。

これで、コントロールが使用できる状態になります。

Import ディレクティブを利用する方法に比べると、

- プロパティの構文が使用できない
- CA-SDK オブジェクトと C++ クラスを明示的にリンクする操作が必要

などがあり、多少、複雑になります。また、CA-SDK 呼び出しは全てディスプレイインターフェースにより行われるため、多少、効率が悪くなります。

Test1Dlg.cpp

```

BOOL CTest1Dlg::OnInitDialog()
{
    ...
    // TODO: 特別な初期化を行う時はこの場所に追加してください。
    m_ICa200.CreateDispatch("CA200Srvr.Ca200.1"); ①
    m_ICa200.m_bAutoRelease = TRUE;
    m_ICa200.AutoConnect();
    LPDISPATCH pICa = m_ICa200.GetSingleCa(); ②
    m_ICa.AttachDispatch(pICa); ③
    LPDISPATCH pIProbe = m_ICa.GetSingleProbe();
    m_IProbe.AttachDispatch(pIProbe);
    LPDISPATCH pMemory = m_ICa.GetMemory();
    m_IMemory.AttachDispatch(pMemory);
    m_ICa.CalZero();

    m_CCaControl.SetRefCa(&m_ICa.m_lpDispatch); ④
    m_CCaControl.SetRefProbe(&m_IProbe.m_lpDispatch); ⑤
    m_CCaControl.SetRefMemory(&m_IMemory.m_lpDispatch); ⑥
    m_CCaControl.UpdateCaInfo(); ⑦
    m_CCaControl.UpdateMemoryInfo(); ⑧

    return TRUE; // TRUE を返すとコントロールに設定したフォーカスは失われません。
}

```

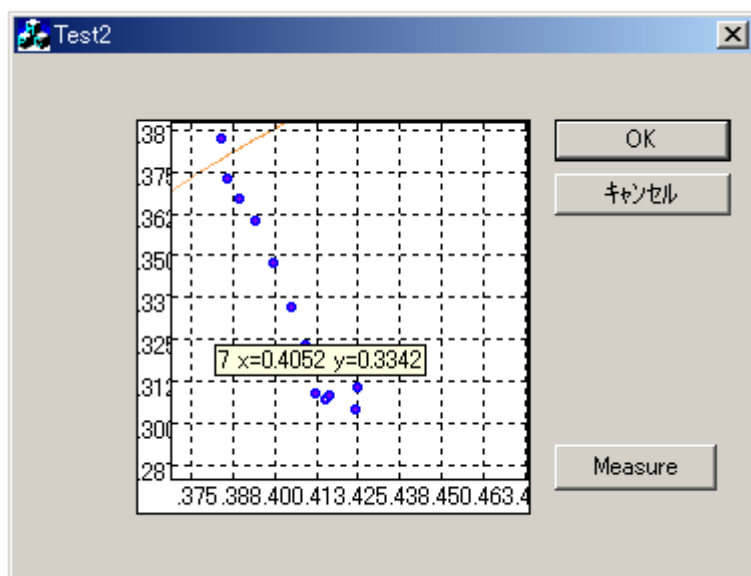

4-3-2 xyControl(xy色度図表示コントロール)

xyControl は CA 測定器の色測定結果をxy色度図にプロットします。

xyControl には約1000個のデータを登録できます。

自動的にデータ登録/表示する機能のほか、表示するデータを、インデクスで指定する機能(複数表示可)があります。

最大で、8倍の拡大表示機能があります(マウス右クリックで倍率メニューが表示されます)。データプロットにマウスを近づけると、データの通し番号(インデクス+1の値)と色度値を表示します(下図は8倍拡大表示にし、色度値表示をした状態です)。



4-3-2-1 プロパティ/メソッド

Ca プロパティ

設定プロパティです。

測定結果を表示するCaオブジェクトを設定します。

コントロールを使用する前に、設定します。

構文：

```
Dim objxyControl as xyControl
```

```
Dim objCa as Ca
```

```
Set objxyControl.Ca = objCa
```

Probe プロパティ

設定プロパティです。

測定結果を表示するProbeオブジェクトを設定します。

コントロールを使用する前に、設定します。

構文：

```
Dim objxyControl as xyControl
```

```
Dim objProbe as Probe
```

```
Set objxyControl.Probe = objProbe
```

SetXYGraphData メソッド

Probe オブジェクトから現在の x y 測定結果を取得・データ登録し、x y 色度図に表示します。
データのインデクスはインクリメントされます。
それ以前のデータの表示は不可視に設定されます。

構文：

```
Dim objxyControl as xyControl
```

```
objxyControl.SetXYGraphData
```

AddXYGraphData メソッド

Probe オブジェクトから現在の x y 測定結果を取得、指定したインデクスのデータとして登録し、x y 色度図に表示します。

インデクスは指定したインデクスに設定されます。

他のインデクスのデータの表示状態を変更することはありません。

構文：

```
Dim objxyControl as xyControl
```

```
Dim lIndex as long
```

```
objxyControl.AddXYGraphData lIndex
```

SetXYData メソッド

指定したインデクス、x・y 色度値でデータを登録し、x y 色度図に表示します。

インデクスは指定したインデクスに設定されます。

他のインデクスのデータの表示状態を変更することはありません。

構文：

```
Dim objxyControl as xyControl
```

```
Dim lIndex as long
```

```
Dim fx as float
```

```
Dim fy as float
```

```
objxyControl.SetXYData lIndex, fx, fy
```

SetVisible メソッド

指定したインデクスのデータ表示を可視に設定します。

構文：

```
Dim objxyControl as xyControl
```

```
Dim lIndex as long
```

```
objxyControl.SetVisible lIndex
```

SetVisibleAll メソッド

指定した Boolean の値 True/False にしたがって、全データの表示を可視/不可視に設定します。

構文：

```
Dim objxyControl as xyControl
```

```
Dim lIndex as long
```

```
Dim bVisible as Boolean
```

```
objxyControl.SetVisibleAll bVisible
```

ClearData メソッド

登録されている全データの内容を削除し、インデクスを初期化します。

構文：

```
Dim objxyControl as xyControl
```

```
objxyControl. ClearData
```

4-3-2-2 VBでの使用法

1-2. で説明した手順で、CA-SDK コンポーネント、xyControl コントロールをVB プロジェクトに追加してください。

下に、CA-SDK の色測定サンプルソフトから抜き出したソースリストを示します。

SDK のオブジェクトを生成した後、コントロールのプロパティにより、①②でCA 測定器を制御するためのオブジェクトをコントロールに設定します (FormMeasurement は、コントロールを貼り付けたフォームです)。

その後、③ClearData でコントロールのデータを初期化します。

以上の操作で、コントロールが使用できる状態になります。

CA-SDK サンプルソフトでは、色測定を実行すると測定結果のxy値を色度図にプロットします。これは、添付リスト④ SetXYGraphData で実行しています。SetXYGraphData は、最新のデータだけを可視に設定するため、それ以前の測定結果のプロット表示はされません。

また、サンプルソフトでは色測定結果をグリッド・コントロールに保存し、ユーザーがコントロールのデータ行を選択すると、その範囲のxy色度を色度図にプロットします。これは、添付リスト⑤⑥で実行します。最初に FALSE を引き数の値として⑤SetVisibleAllを実行し、全表示を不可視に設定します。その後、選択された範囲のこのデータに対して⑥SetVisibleを実行し、表示を ON しています。

Ca200Sample.bas- Main()

```
...
Set objCa200 = New Ca200
objCa200. AutoConnect
...
Set objCa = objCa200. SingleCa
Set objProbe = objCa. SingleProbe
Set objMemory = objCa. Memory
...
Set FormMeasurement. xyControl1. Probe = objProbe    ①
Set FormMeasurement. xyControl1. Ca = objCa    ②
...
```

Sub GridInit()

```
...
xyControl1. ClearData    ③
...
End Sub
```

FormMeasurement.frm

```
Sub SetSingleData(ByVal LisNo As Integer)
...
xyControl1. SetXYGraphData    ④
...
End Sub
```

```

Private Sub grdDataList_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
...
xyControl1.SetVisibleAll False ⑤
lSelectedRow1 = grdDataList.MouseRow
If lSelectedRow0 > lSelectedRow1 Then
    For L = lSelectedRow1 To lSelectedRow0
        If L <= ListNo Then
            xyControl1.SetVisible L ⑥
        Else
            Exit Sub
        End If
    Next L
Else
    For L = lSelectedRow0 To lSelectedRow1
        If L <= ListNo Then
            xyControl1.SetVisible L
        Else
            Exit Sub
        End If
    Next L
End If
End Sub

```

4-3-2-3 VC++(MFC)での利用法

サンプルソフトは2-1. で示した GUI をもつダイアログベースのアプリケーションとします。

最初に、CA-SDK コンポーネント・xyControl コントロールをプロジェクトに追加します。方法は、[4-3-1-3 VC++\(MFC\)での利用法](#)を参照してください。

ここでビルドすると、ツールが自動生成した“ca200Srve.h”で定義されている Imemory クラスで、SetChannelID()が2重定義されているというエラーが発生します。これは、CA-SDK の Memory オブジェクトの ChannelID プロパティに対して、ツールが自動生成した設定用のラップメソッドの名前(SetChannelID)がもともとある Memory オブジェクトの SetChannelID メソッド名と衝突するためです。どちらかの名前を変更してください。サンプルソフトでは、もともとあるメソッドの名前を SetID に変更しています。

メインウィンドーにコントロールと Measure ボタンを貼り付け、GUI を作成します。

この後、

- ①コントロールクラスの DDX 用メンバー変数 m_CxyControl をダイアログクラスに追加
 - ②CA-SDK オブジェクトクラスのメンバ変数 m_ICa200、m_ICa、m_lprobe をダイアログクラスに追加
 - ③Measure ボタンのハンドラーOnMsr()をダイアログクラスに追加
- して、コーディングの準備は終わりです。

サンプルアプリケーションでは、Measure ボタンをクリックすると、色測定し、xyControl にそのxy色度値をプロットしていくものとします。

下に、サンプルソフトのソースコードを示します。

- ①で、Ca200 オブジェクトを生成します。
- ②で、生成した Ca200 オブジェクトの (SingleCa プロパティから) GetSingleCa メソッドにより Ca オブジェクトを得、③で、Ca オブジェクトに対応するラップクラス ICa のインスタンスにアタッチします。以降、ICa のインスタンスを Ca オブジェクトのように操作できます。

同様に、Probe オブジェクトの設定をします。ここまでの、CA-SDK オブジェクトの初期化の部分です。

その後、④SetRefCa()⑤SetRefProbe()で、xyControlのプロパティにCaオブジェクト・Probeオブジェクトを設定、最後に、コントロールの⑥ClearData()メソッドにより、xyControlコントロールのデータを初期化します。

これで、コントロールが使用できる状態になります。

測定・表示の部分は、Measure ボタンのクリックハンドラーOnMsr()にあります。

添付リストで、Ca オブジェクトの⑦Measuref()により、測定を実行します。後は、xyControl の⑧AddXYGraphData()により、コントロールにデータの測定結果の取得・登録・表示を全て委ねます。AddXYGraphData()は、データプロットを単純に追加していくため、それまでの全ての測定結果が色度図にプロット表示されます。

Test2Dlg.cpp

```

BOOL CTest2Dlg::OnInitDialog()
{
    ...
    // TODO: 特別な初期化を行う時はこの場所に追加してください。
    m_ICa200.CreateDispatch("CA200Srvr.Ca200.1");           ①
    m_ICa200.m_bAutoRelease = TRUE;
    m_ICa200.AutoConnect();
    LPDISPATCH pICa = m_ICa200.GetSingleCa();             ②
    m_ICa.AttachDispatch(pICa);                            ③
    LPDISPATCH pIProbe = m_ICa.GetSingleProbe();
    m_IProbe.AttachDispatch(pIProbe);
    m_ICa.CalZero();

    m_CxyControl.SetRefCa(&m_ICa.m_lpDispatch);           ④
    m_CxyControl.SetRefProbe(&m_IProbe.m_lpDispatch);     ⑤
    m_CxyControl.ClearData();                             ⑥
    m_lIndex = 0;

    return TRUE; // TRUE を返すとコントロールに設定したフォーカスは失われません。
}

void CTest2Dlg::OnMsr()
{
    // TODO: この位置にコントロール通知ハンドラ用のコードを追加してください
    m_ICa.Measure(0);                                     ⑦
    m_CxyControl.AddXYGraphData(&m_lIndex);               ⑧
    m_lIndex++;
}

```

4-4 CA-SDK VB サンプルソフト

VB サンプルソフトはソースコードを全て公開しています。本サンプルソフトと VB サンプルソフトのコードを比較すれば、CA-SDK オブジェクトを利用する際のコーディングの仕方には僅かな違いしかなく、ご理解いただけると思います。CA-SDK を利用して C++アプリケーションソフトを組む際の参考にしてください。

4-4-1 色/フリッカ測定

- CA-SDK-Sample-Color フォルダに色/フリッカ測定の VB サンプルソフトのソースがあります。
- Ca200Sample. bas モジュールの Main() プロシージャで CA-SDK オブジェクトの生成・初期化を行っています。VB サンプルソフトの場合、CA1 台・プローブ 1 本・USB 接続の構成を前提としており、この場合、添付のリストにあるように簡便な構成の設定/初期化ができます (②③④)。

添付ソースリストでは、

- ①CA-SDK オブジェクト～Ca200 オブジェクト生成
- ②Ca200 オブジェクトの AutoConnect() メソッドにより、上記構成の設定
- ③Ca200 オブジェクトの SingleCa プロパティにより、接続 Ca オブジェクト取得
- ④Ca オブジェクトの SingleProbe プロパティにより、接続 Probe オブジェクト取得
- ⑤Ca オブジェクトの Memory プロパティにより、Memory チャンネルオブジェクト取得
- ⑥Ca オブジェクトの CAType プロパティにより、CA 測定器のタイプ取得
- ⑦Ca オブジェクトの CalZero() メソッドにより、0 Cal 実行
- ⑧Memory オブジェクトの GetReferenceColor() メソッドにより、接続プローブ ID、基準色設定値を取得

を実行しています。

- 色/JEITA リッカー/FMA フリッカ測定操作のサンプルは、FormMeasurement. frm モジュールの SingleMeasure() プロシージャにあります。いずれも、添付リストで、

- ①Ca オブジェクトの Measure() メソッドにより測定
- ②Probe オブジェクトのプロパティにより、測定結果のステータス/測定値を取得

という操作になります。Measure() メソッドにより測定される項目 (色/JEITA フリッカ/FMA フリッカ) は、Ca オブジェクトの DisplayMode プロパティに何が設定されているかによります。

- JEITA フリッカ測定をした場合、Probe オブジェクトの GetSpectrum(i) メソッドにより引き数 i に与えた周波数の成分が取得できます (③)。

4-4-2 ガンマ測定

- CA-SDK-Sample-Gamma フォルダに校正 VB サンプルソフトのソースがあります。
- ガンマ測定部分は FormGamma. frm モジュールの MeasureGamma() プロシージャで実行しています。ビデオジェネレータを制御し、各原色及び白色の入出力特性～諧調特性を測定しています。

4-4-3 コントラスト測定

- CA-SDK-Sample-Contrast フォルダにコントラスト/ユニフォーミティの VB サンプルソフトのソースがあります。
- コントラスト/ユニフォーミティ測定は、FormContrast. frm モジュールの MeasureUniformity() プロシージャで行っています。

4-4-4 校正

- CA-SDK-Sample-Cal フォルダに校正 VB サンプルソフトのソースがあります。
- 校正ソフトでは、最初に全メモリーチャンネルの校正データを読み出し、表示します。これは、FormCaCal.frm モジュールの Form_Activate() プロシージャで行っており、本説明にそのリストを添付してあります。リストでは

- ①Memory オブジェクトの ChannelNO プロパティにメモリーチャンネル NO を設定
- ②Memory オブジェクトの GetMemoryStatus() により、現在の接続プローブシリアル NO、校正時のプローブシリアル NO、基準色設定時のプローブシリアル NO、校正モード（白色/マトリクス）を取得
- ③Memory オブジェクトの GetReferenceColor() メソッドにより、接続プローブ ID、基準色設定値を取得

の操作によって各チャンネル NO の情報を取得し、表示しています。

- 校正は、FormCaCal.frm モジュールの CommandCal_Click () プロシージャで行っています。添付リストで、UserCal のコメント以降のブロックがマトリクス校正、WhiteCal のコメントのある部分が白色校正、WhiteSet のコメントのある部分が測定値による基準色設定、WhiteDataSet のコメントのある部分が CH00 の数値指定による基準色設定を実行している部分です。
- 測定値による基準色設定以外の校正/基準色設定では、あらかじめ校正/基準色設定に必要な $x \cdot y \cdot Lv$ 値を設定しておきます。また、校正は、Ca オブジェクトの DisplayProbe プロパティに ID が設定してあるプローブが対象となります。
- マトリクス校正は、添付リストにある通り、以下の手順で行います。

- ①Memory オブジェクトの ChannelNO プロパティに校正チャンネル NO を設定
- ②Ca オブジェクトの SetLvxyCalMode() メソッドにより、校正実行モードに設定

R・G・B・White 各校正画像に対し、

- ③Ca オブジェクトの Measure() メソッドにより、校正画像測定
- ④Ca オブジェクトの SetLvxyCalData() メソッドにより校正値設定

R・G・B・White 各校正画像に対し②～④を実行後

- ⑤Ca オブジェクトの Enter() メソッドを実行

- 白色校正は、White 画像に対してだけ、上記操作を実行します。
- 基準色設定は、White 画像に対して、③と⑤の操作を実行します。
- CH00 の基準色設定は、④だけを実行します。

CA-SDK-Sample-Color-Ca200Sampl.bas

Public Sub Main()

```
Dim i As Integer
```

```
'=====
```

```
' Set Error Trap
```

```
'=====
```

```
On Error GoTo E
```

```
'=====
```

```
' Create SDK/Application Object
```

```
'=====
```

```
Set objCa200 = New Ca200 ①
```

```
'=====
' Set Configuration
'=====
```

```
objCa200.AutoConnect ②
```

```
'=====
' Initialize SDK Object
'=====
```

```
Set objCa = objCa200.SingleCa ③
```

```
Set objProbe = objCa.SingleProbe ④
```

```
Set objMemory = objCa.Memory ⑤
```

```
CA_Type = objCa.CAType ⑥
```

```
'=====
' Initialize CA and Application
'=====
```

```
' 0 Calibration
```

```
MsgBox "0-Cal", vbOKOnly
```

```
ObjCa.CalZero ⑦
```

```
Screen.MousePointer = vbHourglass
```

```
objMemory.GetReferenceColor objProbe.ID, typCurrentRefereceData.sRefx, typCurrentRefereceData.sRefy,
typCurrentRefereceData.sRefLv ⑧
```

```
...
```

```
End Sub
```

```
CA-SDK-Sample-Color-FormMeasurement.frm
```

```
Private Sub SingleMeasure()
```

```
Dim i As Integer
```

```
On Error Resume Next
```

```
Select Case SelectDataName
```

```
Case "COLOR"
```

```
'=====
```

```
' Color Measurement
```

```
'=====
```

```
objCa.Measure ①
```

```
typCurrentMeasurementData..dateColorData = Date
```

```
typCurrentMeasurementData.timeColorData = Time
```

```
typCurrentMeasurementData.lColorStatus = objProbe.RD ②
```

```
typCurrentMeasurementData.ssx = objProbe.Sx
```

```
typCurrentMeasurementData.ssy = objProbe.Sy
```

```
typCurrentMeasurementData.sLv = objProbe.Lv
```

```
typCurrentMeasurementData.sLvFL = objProbe.LvFL
```

```
typCurrentMeasurementDataSx = objProbe.X
```

```
typCurrentMeasurementData.Sy = objProbe.Y
```

```
typCurrentMeasurementData.Sz = objProbe.Z
```

```
typCurrentMeasurementData.sud = objProbe.ud
```

```
typCurrentMeasurementData.svd = objProbe.vd
```

```
typCurrentMeasurementData.sdud = objProbe.dud
```



```
typCurrentMeasurementData.LT = objProbe.T
```

Case "FMA"

```
=====
' FMA Measurement
=====
objCa.Measure      ①
typCurrentMeasurementData.dateFMAData = Date
typCurrentMeasurementData.timeFMAData = Time
typCurrentMeasurementData.lFMAStatus = objProbe.RFMA ②
typCurrentMeasurementData.sFMA = objProbe.FlckrFMA
```

Case "JEITA"

```
=====
' JEITA Measurement
=====
objCa.Measure      ①
typCurrentMeasurementData.dateJEITAData = Date
typCurrentMeasurementData.timeJEITAData = Time
typCurrentMeasurementData.lJEITASStatus = objProbeRJEITA ②
typCurrentMeasurementData.sJEITA = objProbe.FlckrJEITA
SpectMin = 0
SpectMax = -100
For i = 6 To 65
    typCurrentMeasurementData.Gspect(i) = objProbe.GetSpectrum(i) ③
    If typCurrentMeasurementData.Gspect(i) <> -999 Then
        If SpectMin > typCurrentMeasurementData.Gspect(i) Then
            SpectMin = typCurrentMeasurementData.Gspect(i)
        End If
        If SpectMax < typCurrentMeasurementData.Gspect(i) Then
            SpectMax = typCurrentMeasurementData.Gspect(i)
        End If
    End If
Next i
End Select

SetCurrentData
End Sub
```

CA-SDK – Sample – Cal – FormCaCal.frm

```
Private Sub Form_Activate()
```

```
...
```

```
Dim i As Integer
```

```
For i = 1 To 100
```

```
DoEvents
```

```
With atypCaCalInfo(i)
```

```
    objMemory.ChannelNO = i - 1 ①
```

```
    objMemory.GetMemoryStatus objProbe.Number, .lCalProbeSNO, .lRefProbeSNO, .lCalMode ②
```

```
    objMemory.GetReferenceColor objProbe.ID, .sRefx, .sRefy, .sRefLv ③
```

```
...
```

```

End With
Next i
...

```

End Sub

Private Sub CommandCal_Click()

```

...
'=====
' UserCal
'=====

objCa.DisplayMode = DSP_LXY
objMemory.ChannelNO = ListNo - 1    ①
objCa.SetLvxyCalMode                ②
bSetMode = True

'-----
' Red
'-----

Result = MsgBox("Red Measure", vbOKCancel)
If Result = vbCancel Then
    objCa.ResetLvxyCalMode
    Exit Sub
End If

objCa.Measure                        ③
objCa.SetLvxyCalData CLR_RED,      typCalData(CLR_RED).sRefx,  typCalData(CLR_RED).sRefy,
typCalData(CLR_RED).sRefLv          ④

'-----
' Green
'-----

Result = MsgBox("Measure Green", vbOKCancel)
If Result = vbCancel Then
    objCa.ResetLvxyCalMode
    Exit Sub
End If

objCa.Measure                        ③
objCa.SetLvxyCalData CLR_GREEN,    typCalData(CLR_GREEN).sRefx,  typCalData(CLR_GREEN).sRefy,
typCalData(CLR_GREEN).sRefLv       ④

'-----
' Blue
'-----

Result = MsgBox("Measure Blue", vbOKCancel)
If Result = vbCancel Then
    objCa.ResetLvxyCalMode
    Exit Sub
End If

objCa.Measure                        ③
objCa.SetLvxyCalData CLR_BLUE,     typCalData(CLR_BLUE).sRefx,  typCalData(CLR_BLUE).sRefy,

```

typCalData(CLR_BLUE).sRefLv ④

```

'-----
'White
'-----
Result = MsgBox("Measure White", vbOKCancel)
If Result = vbCancel Then
    objCa.ResetLvxyCalMode
    Exit Sub
End If

```

objCa.Measure ③

objCa.SetLvxyCalData CLR_WHITE, typCalData(CLR_WHITE).sRefx, typCalData(CLR_WHITE).sRefy,
typCalData(CLR_WHITE).sRefLv ④
objCa.Enter ⑤

ElseIf OptionRefCal.Value = True Then

```

If objMemory.ChannelNO = 0 Then
    MsgBox "CH00 cannot be calibrated", vbOKOnly
    Exit Sub
End If

```

```

=====
' White Cal
=====

```

```

objCa.DisplayMode = DSP_LXY
objMemory.ChannelNO = ListNo - 1 ①
objCa.SetLvxyCalMode ②
bSetMode = True

```

```

'-----
'White
'-----
Result = MsgBox("Measure White", vbOKCancel)
If Result = vbCancel Then
    objCa.ResetLvxyCalMode
    Exit Sub
End If

```

objCa.Measure ③

objCa.SetLvxyCalData CLR_WHITE, typCalData(CLR_WHITE).sRefx, typCalData(CLR_WHITE).sRefy,
typCalData(CLR_WHITE).sRefLv ④
objCa.Enter ⑤

ElseIf OptionRefSet.Value = True Then

```

=====
' White Set
=====

```

```

If objMemory.ChannelNO = 0 Then
    MsgBox "CH00 cannot be set", vbOKOnly
    Exit Sub

```

```
End If
objCa.DisplayMode = DSP_LXY
objMemory.ChannelNO = ListNo - 1    ①
Result = MsgBox("White Measure", vbOKCancel)
If Result = vbCancel Then
    Exit Sub
End If
objCa.Measure    ③
objCa.Enter    ⑤
```

```
ElseIf OptionRefData.Value = True Then
```

```
'=====
' White Data Set(CH00)
'=====
```

```
If objMemory.ChannelNO <> 0 Then
    MsgBox "Only For CH00", vbOKOnly
    Exit Sub
End If
```

```
objCa.SetLvxyCalData CLR_WHITE, typCalData(CLR_WHITE).sRefx, typCalData(CLR_WHITE).sRefy,
typCalData(CLR_WHITE).sRefLv    ④
```

```
Else
    ...
```

```
End Sub
```

4-5 CA-SDK による Visual Basic.net アプリケーションの作成方法

Visual Basic .net は、Microsoft (以降、MS と記します) の新しいアプリケーション開発/実行環境です。MS のアプリケーション統合開発環境 (以下、IDE と記す) である Visual Studio も .net 対応版に移行し、CA-SDK のアプリケーションを開発する場合、この新しい環境でプログラムを作成することになります。CA-SDK は、COM を利用したコンポーネントとして、実現しています。

以前の Visual Basic (以降、VB と記します) では、COM コンポーネントに対する参照の設定をするだけで、そのアプリケーションを簡潔に記述できました。

VB は、.net 版になり、仕様が大きく変更されていますが、IDE により、VB と同様な形で開発が可能です。

ここでは、VB.net を使用して、CA-SDK アプリケーションソフトを作成する方法を説明します (なお、本説明では、Visual Studio.net の 2002 バージョンを使用しています。バージョンが異なる場合、本説明で示した表示とは異なる場合があります)。

アプリケーションは以下の手順で作成します。

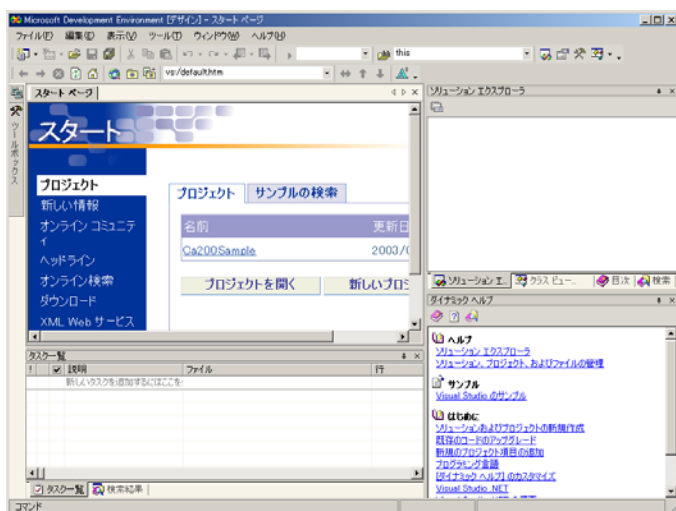
1. CA-SDK をインストールする。
2. IDE で VB.net のプロジェクトを作成する。
3. CA-SDK の参照設定をする。
4. アプリケーションの GUI/コードを作成する。

VB.net での作業を始める前に、“1. CA-SDK をインストールする。”を行ってください。インストールの方法は、CA-SDK のマニュアルに記載されています。

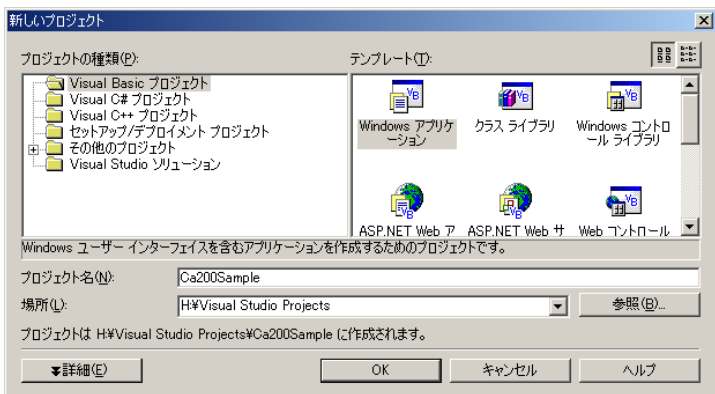
以下、上記の“2. IDE で VB.net のプロジェクトを作成する。”以降の説明をします。

4-5-1 VB のプロジェクト生成

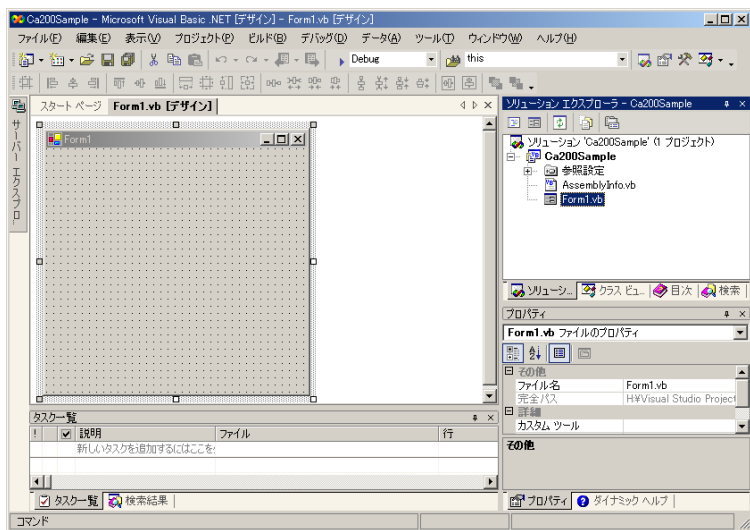
①VS のスタート画面で、新しいプロジェクトを開く (あるいは、**ファイル** **新規作成** **プロジェクト**) を選択します。



②新しいプロジェクトダイアログで、プロジェクトの種類として、Visual Basic プロジェクト、テンプレートとして、Windows アプリケーションを選択、プロジェクトを作成する場所、プロジェクト名~サンプルソフトでは、“Ca200Sample”、を入力して、OK をクリックします。

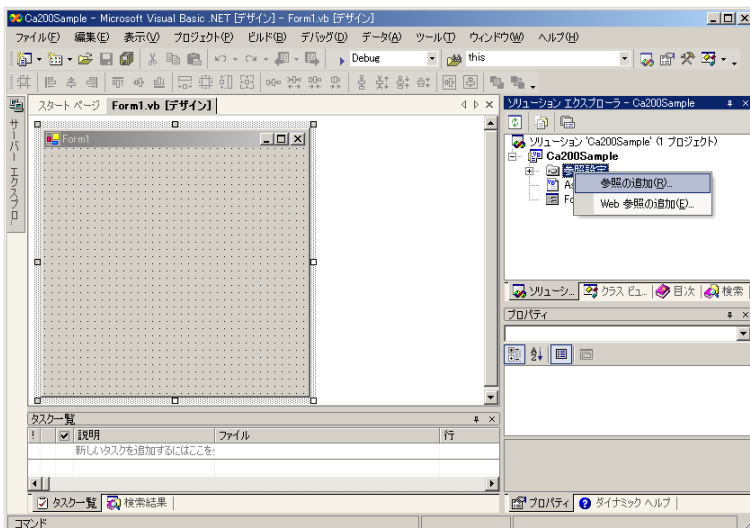


プロジェクトが作成され、フォームのデザイン画面が表示されます。

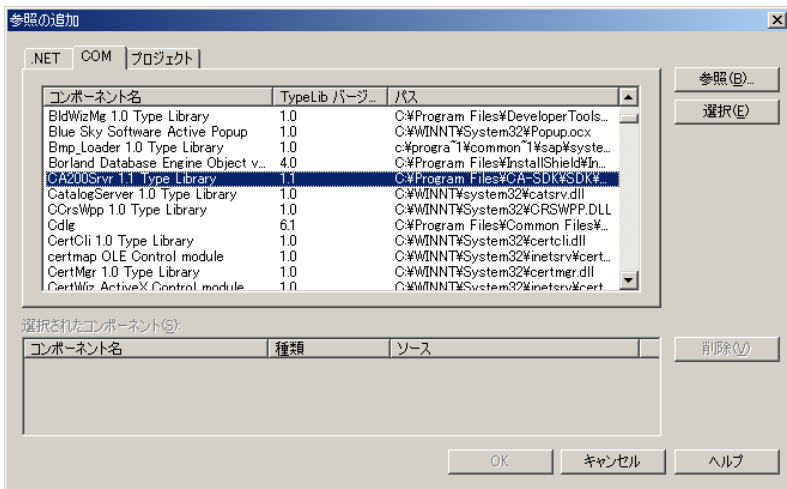


4-5-2 CA-SDKの参照設定

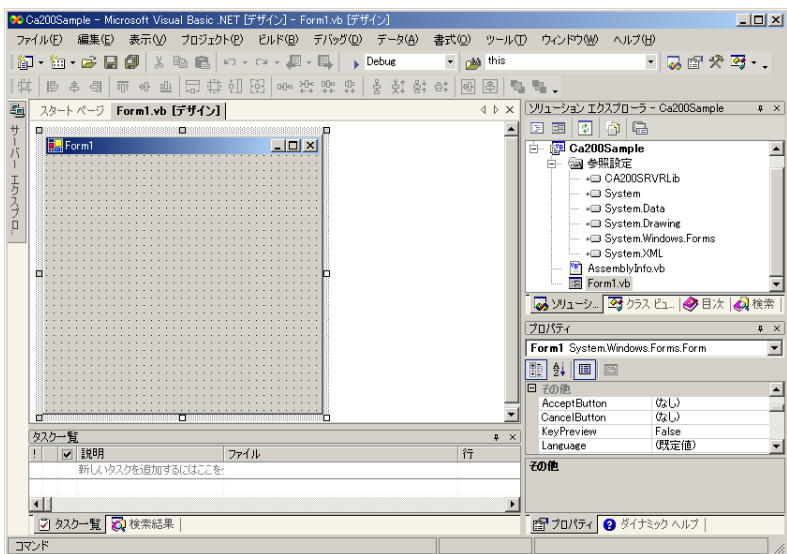
①ソリューションエクスプローラで参照設定を右クリック、ポップアップメニューの参照の追加を選択（あるいは、メニューのプロジェクト参照の追加を選択）します。



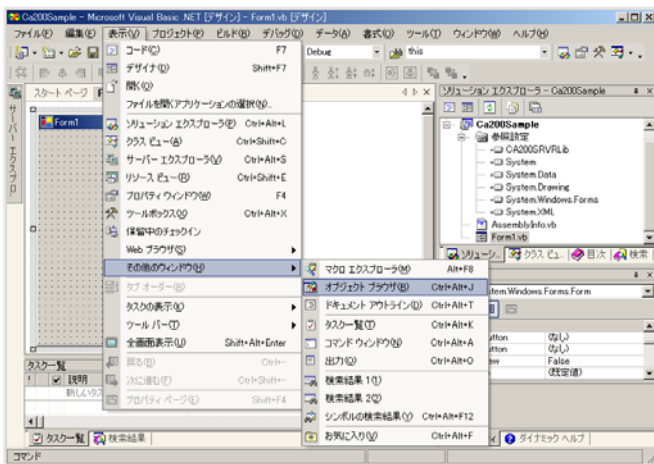
②表示された参照の追加ダイアログで、COM タブを選択、表示されるリストからコンポーネント名 CA200Srvr 1.1 Type Library を選択し、選択ボタンをクリック、“選択されたコンポーネント”に表示されたことを確認して、OK ボタンをクリックします。



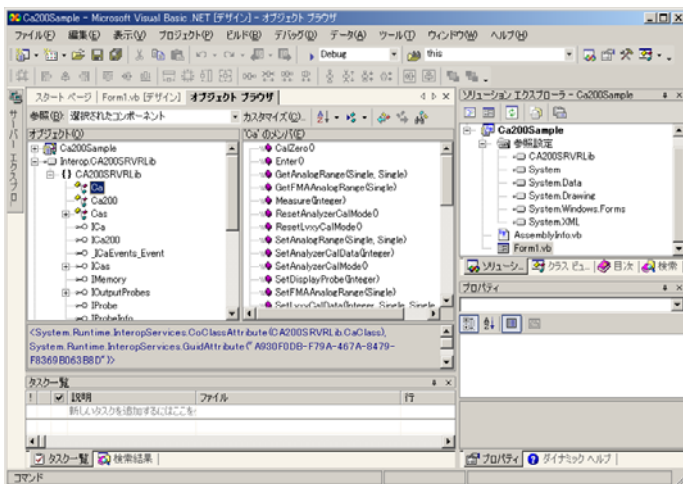
IDE は CA-SDK (のタイプライブラリー) から CA-SDK を使用するための .net アセンブリ (.net のコンポーネント) ~CA200SRVRLib を自動生成し、参照設定に追加します。これで、アプリケーションから CA-SDK が使用できる状態になります。



メニューで、**表示** **その他のウィンドウ** **オブジェクトブラウザ** を選択すると、オブジェクトブラウザが表示されます。



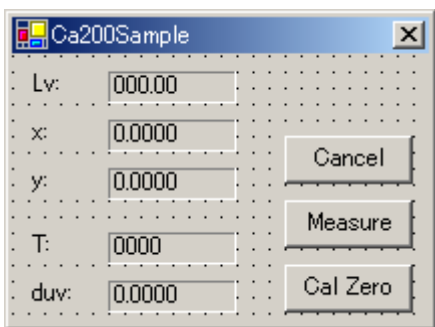
オブジェクトのリストから Interop.CA200SRVLib を選択すると利用できるクラス・インターフェースが確認できます。



4-5-3 アプリケーションの GUI/コードの作成

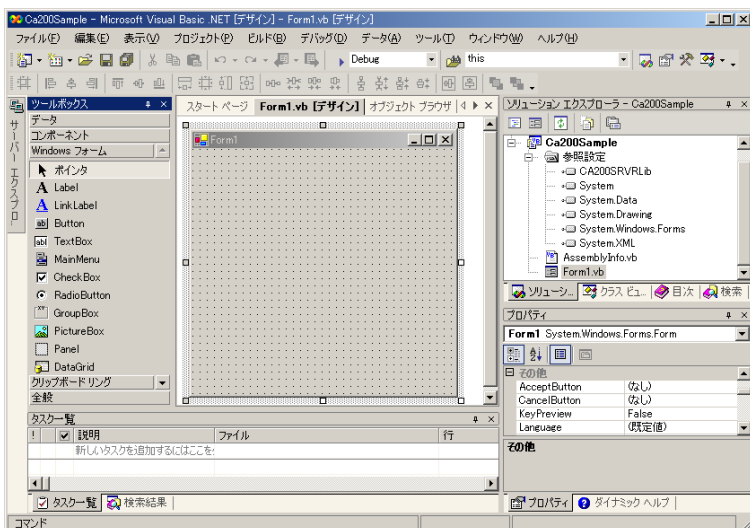
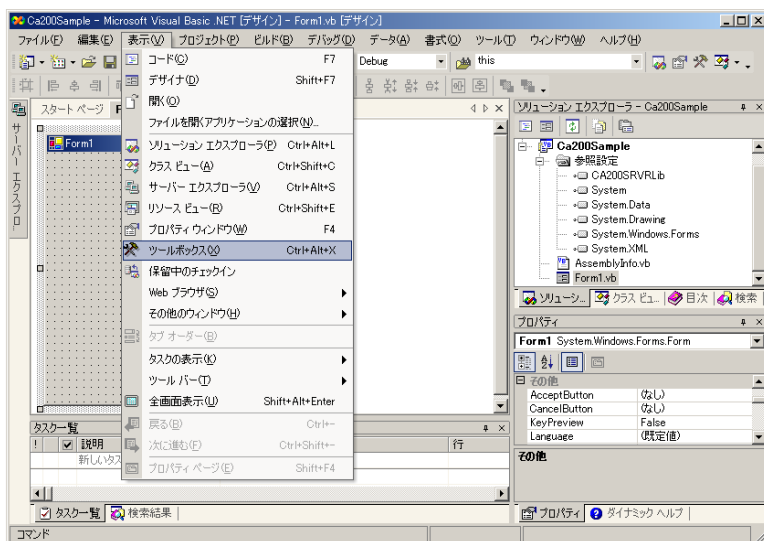
サンプルソフトのGUIを以下のようにします。
 ラベルを 10 個（ラベル用・測定値表示用各 5 個）、ボタンを 3 個利用します。

Cal Zero ボタンをクリックすると、CA 本体で、ゼロ校正を実行します。
 Measure ボタンをクリックすると、20 回連続測定し、測定結果を測定値表示用ラベルに表示します。
 Cancel ボタンをクリックすると、測定を途中で中止します。

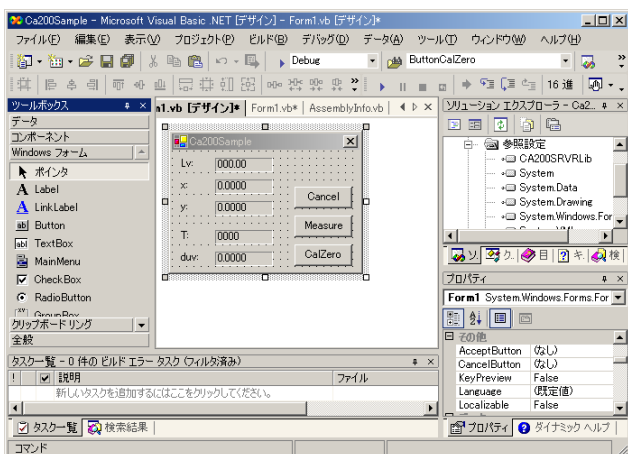


GUI は、ツールボックスを利用、VB と同じ方法で作成します。

ツールボックスはメニュー「表示」>「ツールボックス」を選択、デザイン画面の左のツールバーのツールボックスアイコン上にマウスポインターを移動すると表示されます。



従来と同様、各コントロールのプロパティを設定し、GUI 画面を設計します。



コードも、VB とほぼ同じ方法で記述できます。

以下、サンプルソフトのコードを説明します（本説明の最後に、サンプルソフトのコードを添付しています）。

1. CA-SDK 公開クラスメンバーの宣言

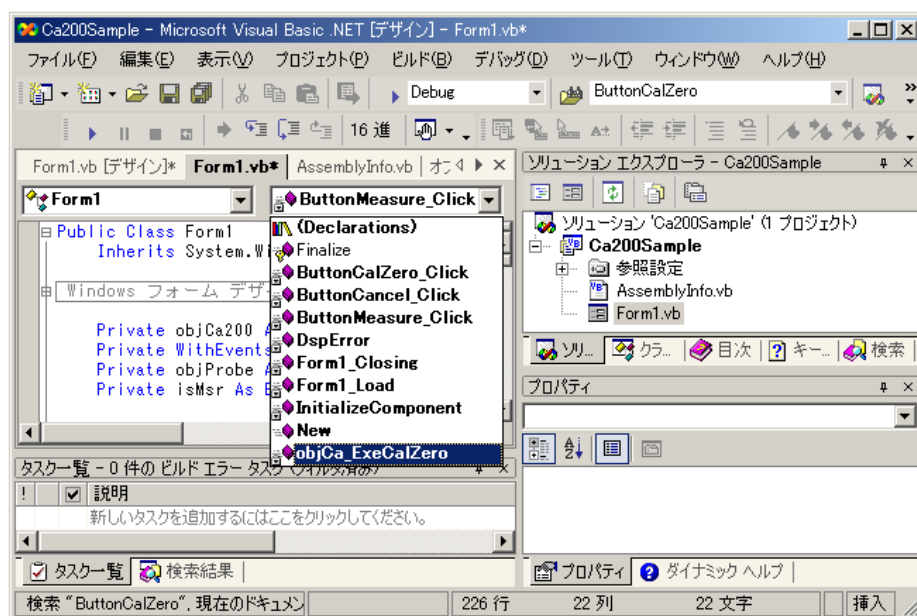
VB は、.net 版になって、完全なオブジェクト指向言語になりました。プログラムを構成する基本単位は、クラスになりました。

サンプルコードにおいて、①～の部分で、Form1 クラスのカスタムプロパティとして CA-SDK の .net クラス型のメンバーを宣言しています。②の部分では、VB と同様、Ca クラス型のメンバーを WithEvents キーワードつきにしています。これにより、CA-SDK が公開するイベントのハンドラを記述できるようになります。Form1 上に置いた各コントロールメンバーの宣言は、“Windows フォーム デザイナで生成されたコード”の中に折りたたまれています。

2. イベントハンドラの作成

各ボタンをクリックしたときのイベントハンドラを記述します。方法は、VB と同様、コード編集画面上部のクラス名ボックスでボタンを選択、イベント名ボックスでイベントを選択、自動生成されたイベントハンドラの定義中に、所要のコードを記述します。

CA-SDK の公開するイベントも同様に記述します。クラス名ボックスで、Ca クラス型のメンバ変数～サンプルソフトでは objCa、を選択、イベント名ボックスで、CA-SDK の公開するイベント～ExeCalZero を選択し、自動生成されたハンドラ～objCa_ExeCalZero() の定義中に、所要のコードを記述します。これにより、プローブ温度がある値以上変化したとき、CA-SDK から発行されるイベントに応じて、ハンドラが実行されます。



サンプルソフトでは、Form1 の Load イベントで、CA-SDK の Ca200 オブジェクトを生成～④、構成の設定～⑤をし、それにより以降、アプリケーションで使用する CA-SDK オブジェクトを取得します。

測定ボタン～ButtonMeasure のクリックイベントで、CA-SDK の Ca オブジェクトにより測定を実行、結果を Probe オブジェクトにより取得し、ラベルに表示します～⑥。

0 校正ボタン～ButtonCalZero のクリックイベント・CA-SDK からの ExeCalZero イベント～⑨で、CA オブジェクトにより 0 校正を実行します～⑧⑩。

フォームの閉じるボタンをクリックした時の Closing イベントでは、Ca オブジェクトをリモートオフに設定します～⑪（Closing イベントハンドラーは、クラス名ボックスのリストで“(Base Class Events)”を選択すると、イベントボックスで選択できる状態になります）。

3. エラー処理

VB.net では、従来と同様のエラー処理が記述できます～③⑩。それに加えて、構造化例外処理も記述できます～⑦。

サンプルソフトのコードから分かる通り、VB.net で CA-SDK のオブジェクトを操作するには、VB とほぼ同じコードで記述できます。

しかし、説明中に一部記したとおり、VB.net は VB とは、全く別の言語といってよいくらいに“中味”は大きく変化しています。VB.net が提供する新しい機能を利用することにより今まで以上に複雑なアプリケーションを簡潔に、効率よく開発できるようになります。

```
Public Class Form1
```

```
    Inherits System.Windows.Forms.Form
```

```
#Region " Windows フォーム デザイナで生成されたコード "
```

```
    ...省略...
```

```
#End Region
```

```
    Private objCa200 As CA200SRVLib.Ca200Class ... ①
```

```
    Private WithEvents objCa As CA200SRVLib.CaClass ... ②
```

```
    Private objProbe As CA200SRVLib.ProbeClass
```

```
    Private isMsr As Boolean
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
        On Error GoTo er ... ③
```

```
        objCa200 = New CA200SRVLib.Ca200Class() ... ④
```

```
        objCa200.AutoConnect() ... ⑤
```

```
        objCa = objCa200.SingleCa
```

```
        objProbe = objCa.SingleProbe
```

```
        Exit Sub
```

```
er:
```

```
    DspError()
```

```
    End
```

```
End Sub
```

```
    Private Sub ButtonCancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonCancel.Click
```

```
        isMsr = False
```

```
        ButtonCancel.Enabled = False
```

```
        ButtonMeasure.Enabled = True
```

```
        ButtonCalZero.Enabled = True
```

```
End Sub
```

```
    Private Sub ButtonMeasure_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonMeasure.Click
```

```
        Dim i As Integer
```

```
        On Error GoTo er
```

```
        isMsr = True
```

```
        ButtonCancel.Enabled = True
```

```
        ButtonMeasure.Enabled = False
```

```

ButtonCalZero.Enabled = False
For i = 1 To 20
    objCa.Measure() ... ⑥
    Labellv.Text = objProbe.Lv.ToString("###.##")
    Labelx.Text = objProbe.sx.ToString("#.####")
    Labely.Text = objProbe.sy.ToString("#.####")
    LabelT.Text = objProbe.T.ToString("####")
    Labelduv.Text = objProbe.duv.ToString("#.####")
    Application.DoEvents()
    If isMsr = False Then
        ButtonCancel.Enabled = False
        ButtonMeasure.Enabled = True
        Exit Sub
    End If
Next

```

```

ButtonCancel.Enabled = False
ButtonMeasure.Enabled = True
ButtonCalZero.Enabled = True
Exit Sub

```

er:

```

DspError()
End

```

End Sub

Private Sub ButtonCalZero_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonCalZero.Click

retry:

```

ButtonMeasure.Enabled = False
ButtonCalZero.Enabled = False
Try ... ⑦
    objCa.CalZero() ... ⑧
Catch er As Exception

```

If MessageBox.Show("Zero Cal Error" + Chr(13) + "Retry?", "CalZero",
MessageBoxButtons.OKCancel) = DialogResult.Cancel Then

```

    objCa.RemoteMode = 0
    End

```

End If

GoTo retry

End Try

```

ButtonMeasure.Enabled = True
ButtonCalZero.Enabled = True

```

End Sub

Private Sub objCa_ExeCalZero() Handles objCa.ExeCalZero ... ⑨

If MessageBox.Show("CalZero?", "CalZero", MessageBoxButtons.OKCancel) = DialogResult.Cancel
Then

```
        Exit Sub
    End If
    ButtonMeasure.Enabled = False
    ButtonCalZero.Enabled = False
retry:
    Try
        objCa.CalZero() ... ⑩
    Catch er As Exception
        MessageBox.Show("Zero Cal Error!" + Chr(13))
        GoTo retry
    End Try
    ButtonMeasure.Enabled = True
    ButtonCalZero.Enabled = True

End Sub

Private Sub DspError()

    Dim msg As String

    msg = "Error from" + Err.Source + Chr(10) + Chr(13) ... ⑪
    msg = msg + Err.Description + Chr(10) + Chr(13)
    msg = msg + "HR:" + (Err.Number - vbObjectError).ToString
    MessageBox.Show(msg)

End Sub

Private Sub Form1_Closing(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles MyBase.Closing

    objCa.RemoteMode = 0 ... ⑫

End Sub

End Class
```

CA-210/100plus 技術資料

5. CA-210 のアプリケーション例

5-1 CA-210使用アプリケーション例(ホワイトバランス調整システム)

5-1-1 ホワイトバランス調整

複数台の液晶モニターに白色を出した場合、色が違って見える場合があります。
この原因は、液晶モニターを構成する部品(フィルタ、回路等)にバラツキがあり、完全に同じ白色にならないからです。

例えば、部品バラツキにより、 x 、 y の色度が各々0.02異なると、下図のような色となります。



$x=0.310, y=0.332, L_v=172.5$
目標の色度、輝度



$x=0.292, y=0.356, L_v=170.3$
調整されていない状態の色度、輝度

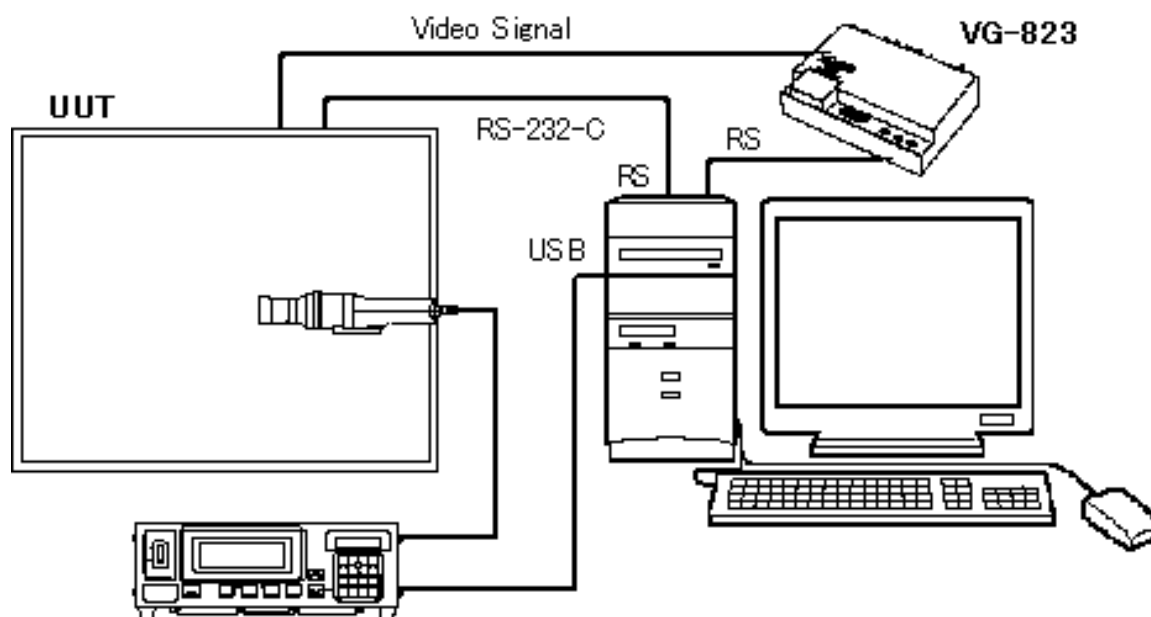
他方、最近、液晶モニターで画像をみるケースが増えてきており、従来以上に色度、輝度に対する要求が厳しくなってきました。このバラツキを少なくするために、液晶モニターのホワイトバランス調整をおこなうケースが増えてきています。



$x=0.315, y=0.337, L_v=169.4$
調整後のパターン図

以下に、CA-210を使用した白色の輝度、色度を高速に、精度よく調整、及びコントラストを検査するシステム例を説明します。

5-1-2 システム構成ブロック図



CA-210 + 1 probe 輝度、色度測定用
 UUT(Unit Under Test) 調整対象液晶モニター
 外部から、校正定数を書き込み可能タイプ 本例ではRS-232-Cを使用
 PC 測定器、液晶モニター、パターンジェネレータ制御用パソコン
 USB port × 1(for CA-210)
 RS-232-C Port × 2 (for LCD Control & VG control)
 PentiumⅢ 660MHz 以上
 解像度：XGA
 VG-823,813,812 被測定モニターへのパターン制御で使用
 アストロデザイン社製パターンジェネレータ
 注：I²C使用時は、RS-232-CからI²C変換基板が必要。

5-1-3 ホワイトバランス調整ソフトの機能

- 液晶モニターを設定された色度、輝度になるように、液晶モニターのゲインを変更して、調整を行います。
- 最大6種類の設定を、続けて行うことができます。
- コントラスト検査
- 合否判定機能

5-1-4 ホワイトバランス調整時間

調整精度が、
 輝度Lv ±10cd/m² 色度±10/1000 以内 なら、5秒以下
 輝度Lv ±5cd/m² 色度±5/1000 以内 なら、10秒以下

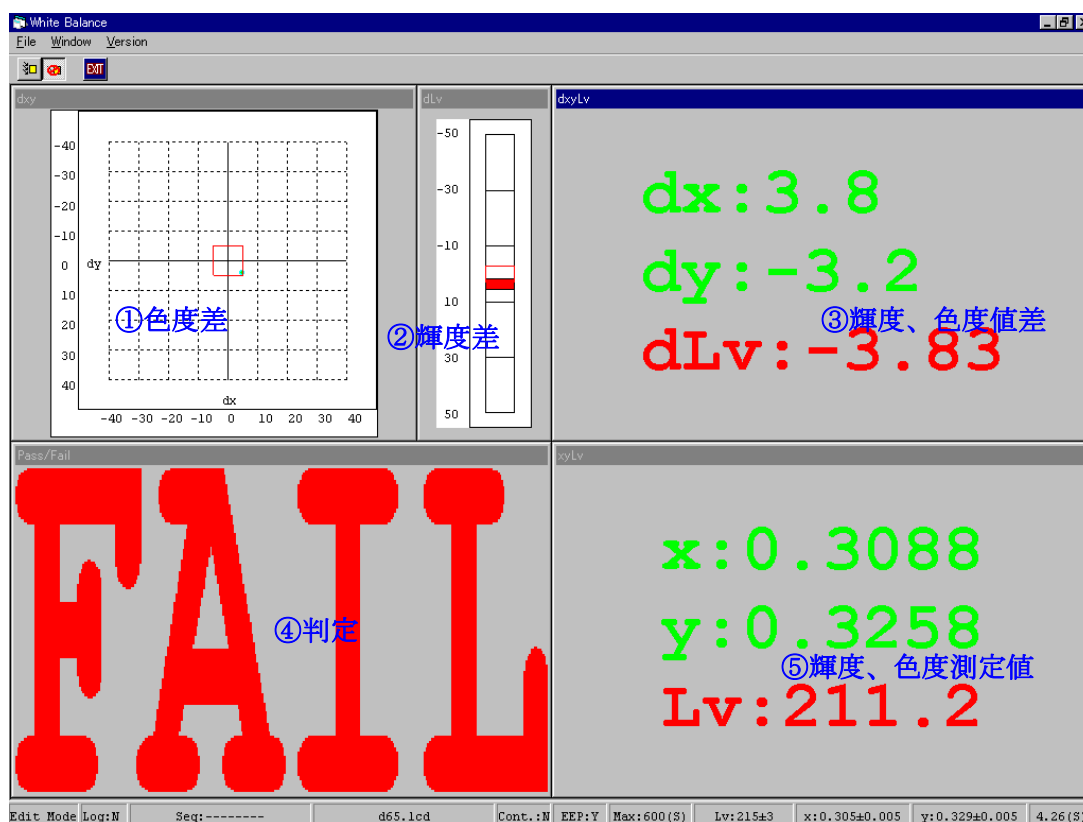
動作環境：富士通FMV-6600MF8/X PⅢ 660MHz メモリー128M時。

注) 液晶モニターにより、調整時間は変化します。

5-1-5 GUI

調整画面

調整の状態を視覚的に表示します。



⑥ステータスバー：

図の見方

設定されている目標値 $Lv: 215 \pm 3$ 、 $x: 0.305 \pm 0.005$ 、 $y: 0.329 \pm 0.005$

図番号と対応しています。

- ①色度差：設定されている目標値が赤い枠で表示されています。その枠内に入るように自動調整を行います。
- ②輝度差：設定されている目標値が赤い枠で表示されています。その枠内に入るように自動調整を行います。
- ③輝度、色度差：設定されている目標値に対しての差を表示します。目標値内時、文字が緑色になります。
- ④判定：輝度、色度ともに目標値内で、Passとなります。Pass時は、文字が緑色になります。
- ⑤輝度、色度測定値：測定値そのものを表示します。目標値内時、文字が緑色になります。
- ⑥ステータスバー：目標輝度、色度などの設定状態、調整時間などの表示を行います。

上記のように、CA-210を用いれば、高速、高精度でのホワイトバランス調整を行うことが可能となります。

5-2 CA210使用アプリケーション例(γ 調整システム)

5-2-1 γ 調整

複数台の液晶モニターに同じカラー画像を出した場合、中間階調の色が、違って見える場合があります。

この原因は、液晶モニターを構成する部品(フィルタ、回路等)にバラツキがあり、完全に同じ中間階調の色にならないからです。

例えば、部品バラツキにより、R, G, Bの中間調の明るさが違うと下図のような色あいとなります。



原画

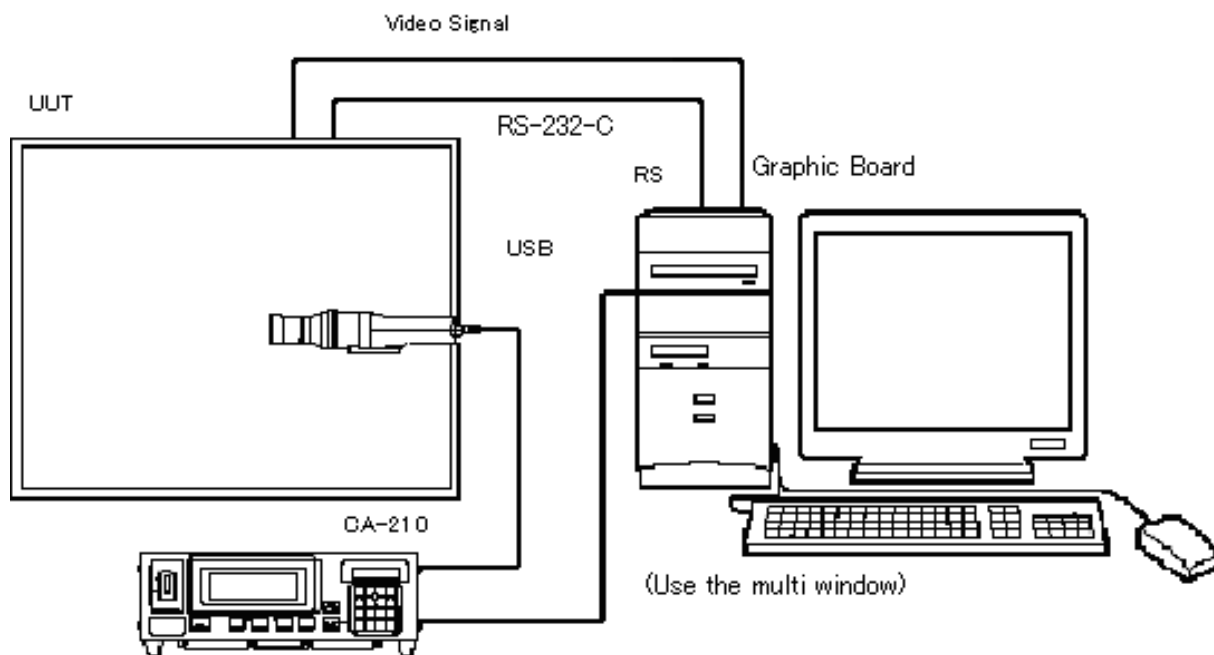


中間調のBlueが強い状態

他方、最近、液晶モニターで画像をみるケースが増えてきており、従来以上に中間調に対する要求が厳しくなっています。このバラツキを少なくするために、液晶モニターの γ 調整をおこなうケースが出てきています。

以下に、CA-210を使用して、この中間階調の各色のバランスを高速に、精度よく調整するシステム例を説明します。

5-2-2 システム構成ブロック図



- CA-210 + 1 probe 輝度、色度測定用
- UUT(Unit Under Test) 調整対象液晶モニター
- 外部から、校正定数を書き込み可能タイプ 本例ではRS-232-Cを使用
- PC 測定器、液晶モニター制御用パソコン
- USB port × 1(for CA-210)
- RS-232-C Port × 1 (for LCD)
- PentiumIII 660MHz 以上
- 解像度：XGA、True Color 24bit 以上
- Graphic Board True color (24Bit以上) Graphic Board 型番指定します。
- あるいは、VG-823,813,812 アストロデザイン社製パターンジェネレータ

5-2-3 調整方法

被測定モニターへのパターン表示は、PC用のグラフィックカードから、出力します。
PCの設定をマルチウィンドウに設定し、PC本体側にソフト表示、グラフィックカードから、被測定パターンを表示します。

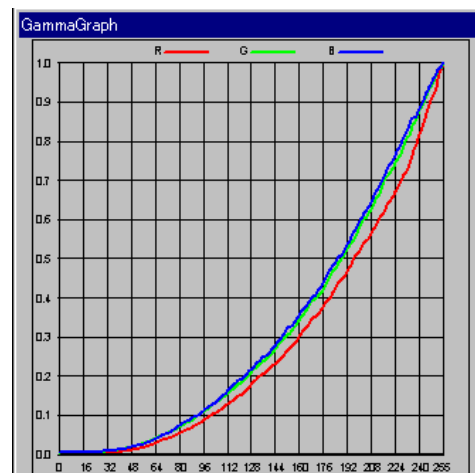
測定の方法は、IEC 61966-4に
規定された3刺激値により計算をしています。(※注 5-2-1)

横軸：0-255階調
縦軸：各色最大値での正規化

① γ 測定

まず、液晶本来の γ 特性を任意の階調で、測定します。
(階調：8、16、32、64、128、256)

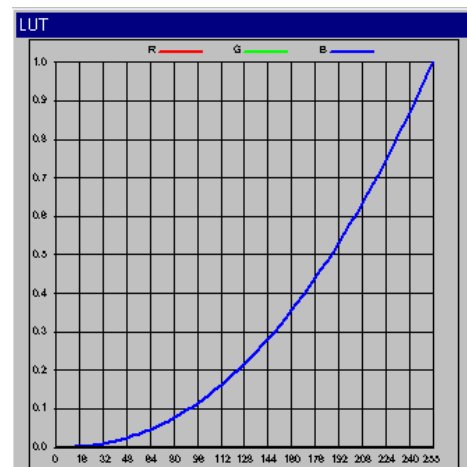
①



② γ 値設定

γ 特性を設定します。
右図は、 $\gamma = 2.2$ の例です。
 $\gamma \text{ curve} = \text{step}^{\wedge} 2.2$ の計算から、設定データを作成します。

②



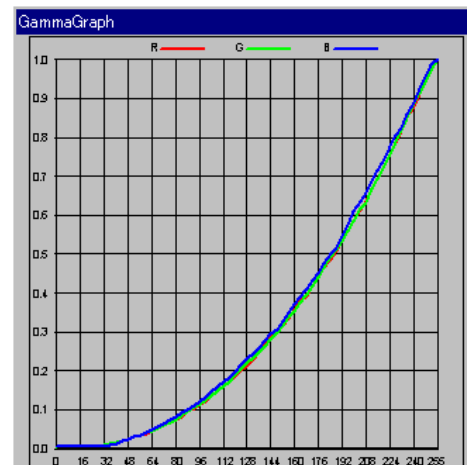
①で測定したデータから、右図の設定したデータになるように
LUT (L o o k U p T a b l e) を計算し、
液晶モニターに書き込みます。

LUTの計算方法は、例えば、階調16での
データの場合、 $(16^{\wedge} 2.2) / (255^{\wedge} 2.2)$ (: 値 / 最大値) の値が
測定したデータのどこにあるかを調べて、
その上下2点のデータから、線形補間して、
階調16でのLUT値を求めます。

③ γ 検査

LUTを書き込んだ液晶をもう一度測定し、
正確に反映されている事を検証します。
右図がその例です。

③



5-2-4 GUI

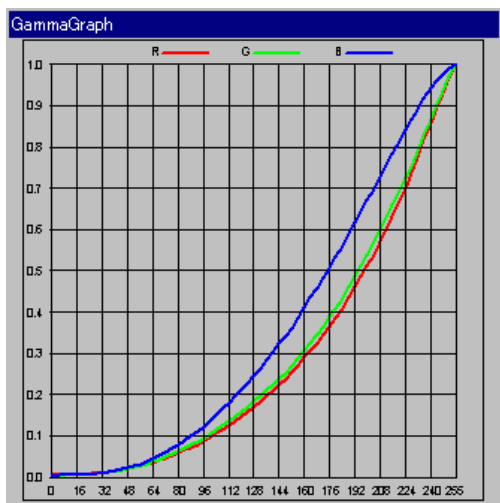
調整画面例

測定時の状態を視覚的に表示します。

① γ 測定画面

設定された測定階調に応じて、液晶モニターの γ 特性を測定、表示する画面です。

ています。

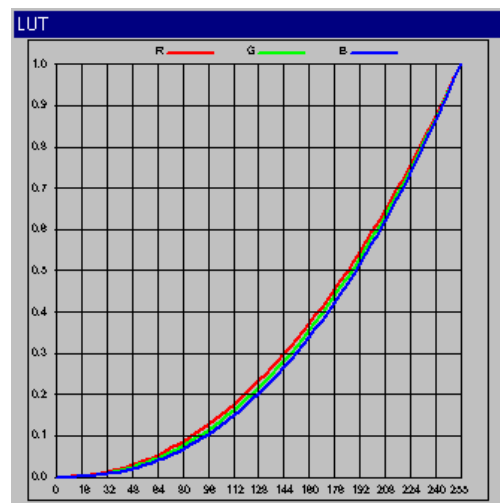


② LUT 設定表示画面

液晶モニターへ書き込む設定データを

表示します。この場合、R : 2. 1

G : 2. 2, B : 2. 3 に設定され



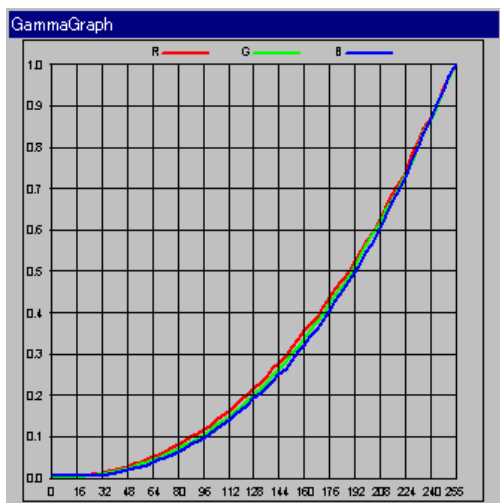
③ γ 確認画面

LUT を書き込んだ液晶モニターを

ます。

測定し、その γ 特性を表示します。

又は、任意の画像を表示し、確認し



5-2-5 γ 調整時間

CA-210を用いて、R/G/B 3色の γ 調整を実施した場合の調整時間を表 5-2-1 に示します。

注:表示を切り替えてから、液晶の応答安定に100msのウェイトを入れています。

液晶モニターへのLUT書き込み時間は含まれていません。

動作環境:FMV-6600MF8/X PIII 660MHz メモリー128M

表 5-2-1 γ 調整時間 [秒]

階調数	8	16	32	64	128	256
調整時間	6	8	12	19	34	64

上記のように、CA-210を用いれば、高輝度から低輝度までの γ 調整を高速に行え、中間色も正確に表現可能になります。

(※注 5-2-1)IEC-61966-4

Method of measurement

The centred colour patches shall be displayed for equally stepped values of input data from

$$\frac{1}{m} 2^N \text{ to}$$

$M=2^N-1$, where m is the number of data and should be at least 32, and N is the number of bits per

channel. For the red channel measurement, $D_G = D_B = 0$; for the green channel, $D_R = D_B = 0$, and for

the blue channel, $D_R = D_G = 0$ shall be kept, respectively.

The readings of the colorimeter for each colour patch on the LCD shall be recorded successively and noted as X^i_c, Y^i_c, Z^i_c where the subscript C shall be replaced by R, G and B, for the red, green, and blue channels, respectively; and the subscript i corresponds to measurement steps, $i = 1, 2, \dots, m$.

The measured tristimulus values shall be normalized by the values corresponding to the maximum excitation for the last step m with input data $M=2^N-1$,

$$X''ic = \frac{X^ic}{X^m_c}$$

$$Y''ic = \frac{Y^ic}{Y^m_c}$$

$$Z''ic = \frac{Z^ic}{Z^m_c}$$

where the subscript C shall be replaced by R, G and B.

X, Y, Z : Measured raw data using spectroradiometers and colorimeters corresponding to tristimulus values.

CA-210/100plus 技術資料

6. 関連規格

6-1 ED-2522 規格と CA-210 の関係

6-1-1 はじめに

液晶パネルの評価方法を規定した規格としては、以下の3つの規格が有名です。

- JEITA (Japan Electronics and Information Technology Industries Association) による、ED-2522
- VESA (Video Electronics Standards Association) による Flat Panel Display measurements Standard
- IEC (International Electrotechnical Commission) による IEC61966-4

また、これ以外に人間工学の観点からの規格としての ISO-13406 があります。

ここでは、ED-2522 規格の概要を説明するとともに、CA-210 と規格に規定された評価方法との関係について述べます。(※注 6-1-1)

6-1-2 評価規格項目

ED-2522 (JEITA) に規定されている評価項目は表 6-1-1 に示すように14項目あります。

表 6-1-1 ED-2522 の評価項目

	評価項目
○	コントラスト比の測定方法
○	応答時間の測定方法
●	モジュール構成ブロックの消費電流及び消費電力の測定方法
○	白色色度の測定方法
○	色の再現範囲の測定方法
○	垂直視野角の測定方法
○	水平視野角の測定方法
○	比反転視野角の測定方法
○	輝度と輝度ムラの測定方法
○	輝度の始動特性の測定方法
●	解像度の測定方法
○	クロストークの測定方法
○	フリッカの測定方法
○	鏡面反射率の測定方法

この項目のうち、○ 印のものは色彩計(輝度計) ([「1-2 色彩計について」参照](#)) を用いて評価する項目です。

6-1-3 評価項目と CA-210 の関係

色彩計の仕様上の特長から、評価項目によっては測定に不適な項目があります。表 6-1-1 の評価項目のうち、○ 印の評価項目に対して、適/不適を判断する色彩計の性能仕様上の特長としては以下の点があります。

- 白色を高精度に測定できる(輝度、色度)
- 単色を高精度に測定できる(輝度、色度)
- 低輝度で高精度に測定できる(※注 6-1-2)
- 測定距離、深度が大きい(液晶パネルを傾けても妨げとならない)
- msオーダーの測定追随性がある
- 多点測定が可能である

色彩計に要求される仕様と、色彩計で測定すべき評価項目の関係を表 6-1-2 に示します。表 6-1-1 の○印は、その評価項目を測定するために、色彩計に必要とされる仕様であることを意味します。特に、CA-210 が適している評価項目、備えている仕様を **青字** で示し、適さない評価項目、備えていない仕様を **赤字** で示しています。(CA-210 の主な性能仕様を表 6-1-3 に示します。)

表 6-1-2 色彩計の仕様と評価項目の関係

評価項目	白色輝度	単色	低輝度	距離	msの追随性	多点
コントラスト比	○		○			○
応答時間	○				○	
白色色度	○					○
色の再現範囲		○				
垂直視野角	○			○		
水平視野角	○			○		
非反転視野角	○			○		
輝度と輝度ムラ	○					○
輝度の始動特性	○					○
クロストーク	○		○			○
フリッカ	○※1					
鏡面反射率	○			○		

※1：液晶フリッカ測定プローブは○、標準測定プローブは不可。

表 6-1-3 CA-210 の主な仕様

測定項目	xyLv, u' v' Lv, TΔuvLv, XYZ, RGB, フリッカ(コントラスト方式, JEITA方式)※2	
測定範囲	0.1~1000 cd/m2	
確度	輝度	±2%±1digit (0.1~1000 cd/m2)
	色度	白色 ±0.002 単色 ±0.004 (基準液晶, 160cd/m2) 白色 ±0.003 (基準液晶, 20.00~1000 cd/m2)
繰返し	輝度	0.2%+1digit (0.1~0.99 cd/m2) 0.1%+1digit (1.00~1000 cd/m2)
	色度	0.001 (液晶白色, 1.00~1000 cd/m2)
測定速度	20 回/秒	
測定エリア	Φ27mm	
測定距離	30mm±10mm	
開口角	5°	
その他	最大5点測定まで多点拡張可能	

※2：測定項目の「フリッカ(コントラスト方式, JEITA方式)」は、液晶フリッカ測定プローブでのみ可能

「コントラスト比」「クロストーク」の評価には、低輝度測定が必須です。また、「色の再現範囲」の評価には、単色で高精度に測定する必要があります。CA-210は、これらの項目の評価が可能な仕様を備えており、表 6-1-2の評価項目青字で示したように標準測定プローブは6項目、液晶フリッカ測定プローブはフリッカ測定機能を含む計7項目の評価を行うことができます。

ED-2522では、標準的な液晶パネルを色彩計で評価する場合には、評価エリアは500ドット以上含むものであると規定しています。CA-210は測定径がΦ27であり、この条件を満たしています。さらに照準機能により測定場所が容易にわかります。

表 6-1-2に示した、色彩計を用いて評価する項目は全て、「暗室条件」(照度1lx以下)で行うこととしています。CA-210は遮光筒を装着可能なので、「暗室条件」を実現し易くなっています。

また、多点測定が可能のため、複数点測定を必要とする場合に効率的な評価が可能です。

ところで、規格のうち、CA-210の評価に適さない項目もあります。例えば「応答時間」の評価には、1ms(ミリ秒)単位での光量変化に追随した出力を得る必要があります。CA-210はその機能を有していません。また、視野角関係(鏡面反射率含む)の評価についても、液晶に対して垂直方向からの測定は可能ですが、斜めの深い角度からの測定には、測定距離が30mmであること、その前後方向の許容値は±10mmであることから、測定プローブが測定の妨げとなったり、測定深度が不足することが考えられ測定できない可能性があります。

6-1-4 評価項目の概要説明

以下に、ED-2522に規定された評価項目のうち、CA-210で測定可能な項目について概要を説明します。(詳細については、ED-2522規格書をお読み下さい(※注 6-1-3))

コントラスト比

画面全体に白色、黒色を表示させ、それぞれの輝度 L_{w1} 、 L_{w2} を画面中央において測定し、コントラスト比を評価します。

評価する表示パターンは、画面全体表示、またはウインドウ（全画面の1/6）表示のいずれかとなります。

特に、ウインドウ評価時には、白部分と黒部分にまたがって測定しないように留意する必要があります。

コントラスト = L_{w1} / L_{w2}

で定義します。

白色色度

白色の色度、色バラツキを評価します。

評価する表示パターンはコントラスト比を最大にした時の全白表示パターンとし、測定点は、1点、5点あるいは9点のいずれかとなります。

色バラツキは画面中央の測定値からの差で定義します。

色再現範囲

コントラスト比が最大となる条件で液晶にR、G、B単色を表示、測定することで色再現が可能な色度範囲を評価します。

輝度と輝度ムラ

表示面の位置による輝度むらを評価します。

評価する表示パターンはコントラスト比を最大にした時の全白表示パターンとし、測定点は、1点、5点あるいは9点のいずれかとなります。

輝度むら = $(B - B_i) / B$ [%] B : 輝度平均 B_i (各点の輝度)

で定義します。

輝度の始動特性

最大輝度表示となる条件での、通電後の輝度変化を経時時間に応じて（数分間隔で、輝度が十分安定するまで）測定します。通電前に温度を標準測定状態に十分合わせることに留意する必要があります。

周辺部分の変化が必要な場合は、多点測定をします。

クロストーク

ウインドウ画面を表示し、駆動信号レベルを変化させて最もクロストークが目立つ所にセットします。

この条件の下で、バックグラウンドの輝度 (L_b)、ウインドウの輝度 (L_w)、クロストークの輝度 (L_i) を測定します。

クロストークレベル = $|L_b - L_i| / |L_b - L_w|$ [%]

で定義します。

(あるエリアに表示している表示パターンの輝度・色度が、他の領域の表示パターンの輝度・色度の影響を受けて変化することをクロストークといいます)

フリッカ

フリッカ発生時の輝度変化特性を測定し、その周波数成分を求めた後、DC成分パワースペクトル (P_0) と最大周波数成分パワースペクトル (P_x) の比に基づいて下式に基づいてフリッカ値を

算出します。

$$\text{フリッカ値} = 10 \log (P_x / P_0) \quad [\text{dB}]$$

なお、輝度変化特性は「人間の目の周波数応答特性」が反映されたものとします。[\(「1-7 フリッカの測定」参照\)](#)

(※注 6-1-1)

現在、ED-2522規格をベースに、国際規格である IEC 61747-6 として審議中です。

(※注 6-1-2)

液晶の表示可能な最低輝度は約0.5cd/m²なので、色彩計にはそれ以下の測定範囲仕様が必要であると言えます。

(※注 6-1-3)

規格本文は以下の HOME Page から購入することができます。

http://www.jeita.or.jp/japanese/public/standard/device/dev_04.htm

6-2 VESA 規格と CA-210 の関係

6-2-1 はじめに

液晶パネルの評価方法を規定した規格としては、以下の3つの規格が有名です。

- JEITA (Japan Electronics and Information Technology Industries Association) による、ED-2522
- VESA (Video Electronics Standards Association) による Flat Panel Display measurements Standard
- IEC (International Electrotechnical Commission) による IEC61966-4

また、これ以外に人間工学の観点からの規格としての ISO-13406 があります。

ここでは、Flat Panel Display measurements Standard (以降 VESA 規格という) の概要を説明するとともに、CA-210 と規格に規定された評価方法との関係について述べます。(※注 6-2-1)

表 6-2-1 CA-210 の主な仕様

測定項目	xyLv, u' v' Lv, TΔuvLv, XYZ, RGB, フリッカ(コントラスト方式, JEITA 方式) ※3	
測定範囲	0.1~1000 cd/m ²	
確度	輝度	±2%±1digit (0.1~1000 cd/m ²)
	色度	白色 ±0.002 単色 ±0.004 (基準液晶, 160cd/m ²) 白色 ±0.003 (基準液晶, 20.00~1000 cd/m ²)
繰返し	輝度	0.2%+1digit (0.1~0.99 cd/m ²) 0.1%+1digit (1.00~1000 cd/m ²)
	色度	0.001 (液晶白色, 1.00~1000 cd/m ²)
測定速度	20 回/秒	
測定エリア	Φ27mm	
測定距離	30mm±10mm	
開口角	5°	
その他	最大 5 点測定まで多点拡張可能	

※3: 測定項目の「フリッカ(コントラスト方式, JEITA 方式)」は、液晶フリッカ測定プローブでのみ可能

6-2-2 評価条件とCA-210の関係

VE SA規格には評価について、基本的な評価条件、測定器（色彩計）に要求される仕様が定義されています。この内容をそれぞれ表6-2-2、表6-2-3に示します。ただし、評価条件はこれに拘束されることはなく、これ以外の条件で評価することも許容しています。ただし、その場合には、評価結果に、その条件を特記することとしています。

表 6-2-2 基本的な評価条件

条件項目	条件内容
温度	20°C±5°C
気圧	86~106 Pa
湿度	25~85 RH
ウォームアップ時間	20分以上
外光	1 lx以下
視認する角度誤差	±0.3°
評価する画素数	500 pixels以上
中心位置誤差	±3%（表示可能エリアを100%として）

表 6-2-3 測定器に要求される仕様

仕様項目	要求内容
開口角, 測定角	2° 以下
輝度確度	±5%
輝度繰返し性	0.50%
色度確度	±0.002（A光源, 10cd/m ² ）
積分時間	これに起因する誤差が繰返し性の2倍以内

“基本的な評価条件”とCA-210の仕様を比較すると、以下のことが言えます。

- ・遮光筒を装着可能なので、1Lx以下を実現し易い。
- ・測定径がΦ27であり、評価する画素数500ドット以上の条件を満たしている。

“測定器に要求される仕様”とCA-210仕様を比較すると、「開口角、測定角」「色度確度」の2点で、CA-210仕様が合致しません。この点について、液晶評価という点から説明します。

- ・「開口角、測定角」について（※注6-2-2）

VE SA規格では2°以下を推奨しているのに対し、CA-210は開口角が5°（測定プローブ）です。6500K、160cd/m²の白色均一パターンを表示した液晶を分光色彩計により測定することで、開口角と色度の関係を求めた結果を表6-2-4に示します。ここでは、開口角1°の測定値を基準とした時の各開口角の測定値差を示しています。この結果から、液晶測定において、開口角が1°～5°では、色度差x_yが0.001以下、輝度差が0.3%以下であり、開口角が5°でも開口角2°以下場合とほぼ等しい確度で測定することが可能であることがわかります。つまり、CA-210で評価しても測定器に許される測定誤差範囲内であり、実質上問題がないと言えます。

- ・「色度確度」について

VE SA規格では色度確度について“A光源で±0.002以内”を推奨しているのに対し、CA-210は色度確度を液晶に対して定義しており、A光源に対しては定義していません。ところで、刺激値直読式色彩計においては、その分光応答特性がCIE等色関数とわずかに異なることに起因して、測定器がA光源で高精度に確度保証されていても、液晶を測定した場合には、精度が大きく低下します。表6-2-5に、A光源に対して校正した刺激値直読式色彩計を用いて、5種類の液晶（白色）を測定した場合の数値実験結果を示します。A光源で誤差がない場合にも、液晶測定時には色度誤差が0.009も発生することがわかります。CA-210の確度仕様は表6に示すとおりであり、±0.002より大きいですが、液晶測定に対して、VE SAが推奨する“A光源で±0.002に確度保証された測定器”に、最大0.009の誤差要因があることを考慮すると、より高精度であると言えます。[（「1-2 色彩計について」参照）](#)

表 6-2-4 液晶評価における開口角と測定値

開口角 [deg]	Δx	Δy	ΔLv
1	0.0000	0.0000	0.0%
2	0.0000	0.0001	-0.1%
5	0.0000	0.0006	0.2%

表 6-2-6 CA-210の色度確度

輝度範囲	色度確度
0.1~4.99cd/m2	±0.005
5.00~19.99cd/m2	±0.004
20.00~1000cd/m2	±0.003
160cd/m2	±0.002

表 6-2-5 刺激値直読式色彩計における数値実験結果

	Δx	Δy
A光源	0.000	0.000
液晶 A	-0.008	-0.003
液晶 B	-0.009	-0.005
液晶 C	-0.009	-0.002
液晶 D	-0.008	-0.002
液晶 E	-0.008	-0.004

6-2-3 評価項目とCA-210の関係

VE SA規格には11の大項目の下に、合計54の評価項目が規定されています。このうち、強度関係、電力関係を除くほとんどの項目が、色彩計（輝度計）を用いて評価する項目です。この項目を表6-2-7に示します。

色彩計の仕様上の特長から、評価項目によっては不適な項目があります。この適/不適を判断する色彩計の性能仕様上の特長としては以下の点があります。

- ・ 白色を高精度に測定できる（輝度、色度）
- ・ 単色を高精度に測定できる（輝度、色度）
- ・ 低輝度で高精度に測定できる（※注6-2-3）
- ・ 測定距離、深度が大きい（液晶パネルを傾けても妨げとまらない）
- ・ msオーダーの測定追随性がある
- ・ μmオーダーの測定解像度がある
- ・ 多点測定が可能である

色彩計に要求される仕様と、色彩計で測定すべき評価項目の関係を表6-2-7に示します。表6-2-7の○印は、その評価項目を測定するために、色彩計に必要とされる仕様であることを意味します。特に、CA-210が適している評価項目、備えている仕様を青字で示し、適さない評価項目、備えていない仕様を赤字で示します。（CA-210の主な性能仕様は表6-2-1に示します。）

表 6-2-7 色彩計の仕様と評価項目の関係

大項目	評価項目	白色輝度	単色	低輝度	距離	msの追随性	μmの解像度	多点
i	Luminance and Color of Full-screen White	○						
i	Luminance and Color of Full-screen Black	○		○				
i	Contrast Ratio of Full Screen	○		○				
i	Gamut and Colors of Full Screen		○					
i	Gray Scale of Full Screen	○		○				
i	Color Scales of Full Screen		○	○				
i	Full-Screen Gray-Scale Color Changes	○		○				
i	Luminance Adjustment Range	○		○				
i	Luminous Flux	○			○			
ii	Luminance&Contrast of Centered Box		○					

ii	Centered Box On-Off Luminance&Contrast	○		○				
ii	Tranverse Contrast of Centered Box	○		○				
ii	Gray Scale of Centered Box	○		○				
ii	Color Gamut of Centered Box	○		○				
ii	Color Scales of Centered Box		○	○				
ii	Halation	○		○				
ii	Luminance Loading	○		○				
ii	Checkerboard Luminance&Contrast (n × m)	○		○				○
iii	Response Time					○		
iii	Residual Image	○		○				○
iii	Warm-up-time Measurement	○						
iii	Dominant Flicker Component	○						
iii	Flicker Modulation Amplitude	○						
iv	Sampled Uniformity&Color of White	○						○
iv	Sampled Uniformity of Black	○		○				○
iv	Sampled Uniformity of Contrast Ratio	○		○				○
iv	Sampled Uniformity of Colors		○					○
iv	Sampled Uniformity of Dark Gray	○						○
iv	Anomalous nonuniformity	○						
v	Four-Point Viewing Angle	○				○		
v	Threshold-based H&V Viewing Angles	○				○		
v	Gray-Scale Inversion	○				○		
v	Viewing-Cone Thresholds	○				○		
v	Gray-Scale Inversion Viewing Cone	○				○		
vi	Reflectance With Diffuse Illumination			○		○		
vi	Ambient Contrast Ratio	○				○		
vi	Large-Source Diffuse Reflectance	○				○		
vi	Large-Source Specular Reflectance	○				○		
vi	Small-Source Specular Reflectance	○				○		
vii	Line Luminance and Contrast							○
vii	N×N Grill Luminance and Contrast							○
vii	Pixel Fill Factor							○
vii	Shadowing(Gray-Scale Artifacts)							○
vii	Intracharacter Luminance and Contrast							○
vii	Defective Pixel Analysis							○
vii	Resolution From Contrast Modulation							○

上表の大項目は、それぞれ以下の分類となります。

- (i) Center Measurements of Full Screen
- (ii) Box-Pattern Measurements
- (iii) Temporal Performance
- (iv) Uniformity
- (v) Viewing Angle Performance
- (vi) Reflection
- (vii) Detail, Resolution, and Artifacts

6-2-4 評価項目の概要説明

以下に、VESA規格に規定された評価項目のうち、CA-210で測定可能な項目について概要を説明します。(詳細については、VESA規格書をお読みください(※注 6-2-4))

Luminance and Color of Full-screen White

画面全体に白色を表示させ、その中央部の輝度、色度を評価します。

Luminance and Color of Full-screen Black

画面全体に黒表示させ、その中央部の輝度、色度を評価します。

Contrast Ratio of Full Screen

画面全体に白色、黒色を交互に表示させ、それぞれの輝度 L_{w1} 、 L_{w2} を測定し、コントラストを評価します。

なお、画面全体に表示しない場合、あるいは垂直方向から測定していない場合には、その条件をコントラスト値に付記することとしています。

コントラスト = L_{w1} / L_{w2}

で定義します。

Gamut and Colors of Full Screen

画面全体にRGB単色(あるいは中間色)を表示させ、その中央部の輝度、色度を評価します。

Gray Scale of Full Screen

画面全体に8段階のグレー色(あるいは16段階)を順次表示させ、その中央部の輝度を各々評価します。

この結果、 γ 特性(入力信号—発光強度特性)を得ることができます。

Color Scales of Full Screen

画面全体に8段階のRGB単色(あるいは16段階)を順次表示させ、その中央部の輝度を各々評価します。この結果、RGB単色の γ 特性(入力信号—発光強度特性)を得ることができます。

Full-Screen Gray-Scale Color Changes

画面全体に8段階のグレー色(あるいは16段階)を順次表示させ、その中央部の色度を各々測定します。

この結果、輝度変化に伴う色度変化を評価することができます。

色度変化量は u' v' 座標系上の距離で定義します。

Luminance Adjustment Range

輝度調整機能を具備しているディスプレイにおいて、調整可能な最大輝度 (L_{max})、最小輝度 (L_{min}) を表示させ、測定します。

調整可能輝度幅 = $(L_{max} - L_{min}) / L_{max}$ [%]

で定義します。

Luminance&Contrast of Centered Box

黒色を背景色とし、その中央に白色ウインドウ(全画面の1/5~1/6)を表示させ、ウインドウ内の1点の輝度 (L_{w1})、ウインドウ外の周辺部8点の平均輝度 (L_{w2}) を測定し、コントラストを評価します。

コントラスト = L_{w1} / L_{w2}

で定義します。

Centered Box On-Off Luminance&Contrast

黒色を背景色とし、その中央に白色ウインドウ(全画面の1/5~1/6)、及び黒色を表示させ、各表示毎に中央の輝度を測定します。この結果、最大輝度、最小輝度、コントラストを評価します。

最大輝度を L_{w1} 、最小輝度を L_{w2} とすると
 コントラスト = L_{w1} / L_{w2}
 で定義します。

Transverse Contrast of Centered Box

黒色を背景色とし、その中央に白色ウインドウ（全画面の $1/5 \sim 1/6$ ）を表示させ、ウインドウ内の1点の輝度（ L_{w1} ）、ウインドウ左右2点の平均輝度（ L_{w2} ）を測定します。
 コントラスト = L_{w1} / L_{w2}
 で定義します。

Gray Scale of Centered Box

黒色を背景色とし、その中央のウインドウ部（全画面の $1/5 \sim 1/6$ ）に8段階のグレー色（あるいは16段階）を順次表示させ、その中央部の輝度を各々評価します。
 この結果、 γ 特性（入力信号—発光強度特性）を得ることができます。

Color Gamut of Centered Box

黒色を背景色とし、その中央のウインドウ部（全画面の $1/5 \sim 1/6$ ）にRGB単色（あるいは中間色）を順次表示させ、その中央部の輝度、色度を評価します。

Color Scales of Centered Box

黒色を背景色とし、その中央のウインドウ部（全画面の $1/5 \sim 1/6$ ）に8段階のRGB単色（あるいは16段階）を順次表示させ、その中央部の輝度を各々評価します。
 この結果、RGB単色の γ 特性（入力信号—発光強度特性）を得ることができます。

Halation

白色を背景色とし、その中央の黒色ウインドウ部の大きさを順次変化させ、その中央部の輝度を各々測定します。
 この結果、黒色ウインドウ部の大きさと輝度の特性を得ることができます。
 上記測定で得た最大輝度を L_{max} 、画面全体に黒色表示した場合の輝度を L_b 、画面全体に白色表示した場合の輝度を L_w とすると、
 $\text{ハレーション} = (L_{max} - L_b) / L_w$
 で定義します。

Luminance Loading

黒色を背景色とし、その中央の白色ウインドウ部の大きさを順次変化させ、その中央部の輝度を各々測定します。
 この結果、白色ウインドウ部の大きさと輝度の特性を得ることができます。
 上記測定で得た最大輝度を L_{ext} 、画面全体に白色表示した場合の輝度を L_w とすると、
 $\text{ローディング} = (L_{ext} - L_w) / L_w$
 で定義します。

Checkerboard Luminance&Contrast (n×m)

市松模様（図 6-2-1）を表示し、その白色部、黒色部の輝度、コントラスト比を評価します。
 コントラストの定義は「Luminance&Contrast of Centered Box」に同じです。

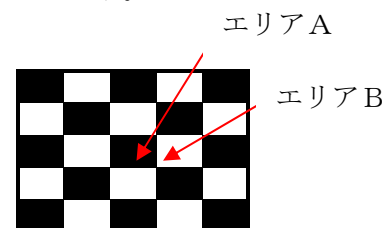


図 6-2-1 市松模様

Residual Image

市松模様を表示した後の、残像を評価します。以下の段階で行います。
 ①画面全体に白色を表示させ、図 6-2-1 のエリア A、B の輝度を測定します（それぞれ L_A 、 L_B と

する)。

②表示パターンを切換え、図 6-2-1 に示す市松模様を一定時間表示させます。

③その後、再び、画面全体に白色を表示させ、エリア A、B の輝度を測定します。

(それぞれ KA, KB とします)。

残像因子 = $1 - (L_B / L_A) / (K_A / K_B)$

で定義します。

Warm-up-time Measurement

白色表示において、通電後の経時時間に対する輝度変化を逐次測定することにより、時間あたりの輝度変化率が 5% 以下 (安定時輝度を基準とする) となるまでの時間を評価します。

通電前に温度を標準測定状態に十分合わせることに留意する必要があります。

Dominant Flicker Component

フリッカ発生時の輝度変化特性を測定し、その周波数成分を求めた後、DC 成分と最大周波数成分の比に基づいてフリッカ値を算出します。輝度変化特性は「人間の目の周波数応答特性」が反映されたものとします。 ([「1-7 フリッカの測定」参照](#))

Flicker Modulation Amplitude

フリッカ発生時の輝度変化特性を測定し、DC 成分と振幅の比に基づいてフリッカ値を算出します。輝度変化特性は「人間の目の周波数応答特性」が反映されたものとします。 ([「1-7 フリッカの測定」参照](#))

Sampled Uniformity&Color of White

画面全体に白色を表示させ、画面上の 5 点 (または 9 点) の輝度、色度を測定、比較することにより輝度ユニフォーミティ、色度ユニフォーミティを評価します。

全測定値に対する最大輝度を L_{max} , 最小輝度を L_{min} とすると

輝度ユニフォーミティ = $(L_{max} - L_{min}) / L_{max}$ [%]

で定義します。

色度ユニフォーミティは $u' v'$ 座標系上の測定値間の最大距離で定義します。

Sampled Uniformity of Black

画面全体に黒色を表示させ、画面上の 5 点 (または 9 点) の輝度を測定、比較することによりユニフォーミティを評価します。

全測定値に対する最大輝度を L_{max} , 最小輝度を L_{min} とすると

輝度ユニフォーミティ = $(L_{max} - L_{min}) / L_{max}$ [%]

で定義します。

Sampled Uniformity of Contrast Ratio

画面全体に黒色を表示させ、画面上の 5 点 (または 9 点) の輝度、色度を測定、比較することによりコントラスト・ユニフォーミティを評価します。

各測定点 i の最大輝度を L_{max} , 最小輝度を L_{min} とすると

コントラスト $C_i = L_{min} / L_{max}$ [%]

で定義します。

C_i の最大値を C_{max} , 最小値を C_{min} とすると、

コントラスト・ユニフォーミティ = $(C_{max} - C_{min}) / C_{max}$ [%]

で定義します。

Sampled Uniformity of Colors

画面全体にカラー色を表示させ、画面上の 5 点 (または 9 点) の輝度、色度を測定、比較することにより輝度ユニフォーミティ、色度ユニフォーミティを評価します。

輝度ユニフォーミティ、色度ユニフォーミティの定義は、「Sampled Uniformity&Color of White」に同じです。

Sampled Uniformity of Dark Gray

画面全体にダークグレーを表示させ、画面上の 5 点 (または 9 点) の輝度、色度を測定、比較することにより輝度ユニフォーミティ、色度ユニフォーミティを評価します。

輝度ユニフォーミティ、色度ユニフォーミティの定義は、「Sampled Uniformity&Color of White」に同じです。

Anomalous nonuniformity

画面全体に白色を表示させ、画面上の最大輝度エリア、最小輝度エリアを抽出し、それぞれのエリアの輝度測定することにより輝度ユニフォーミティを評価します。

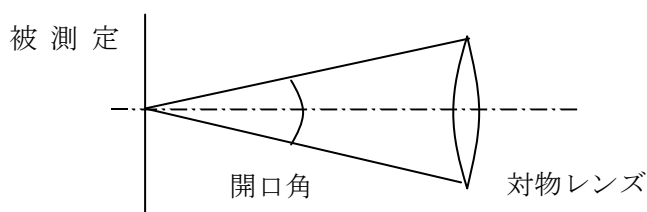
輝度ユニフォーミティの定義は、「Sampled Uniformity&Color of White」に同じです。

(※注 6-2-1)

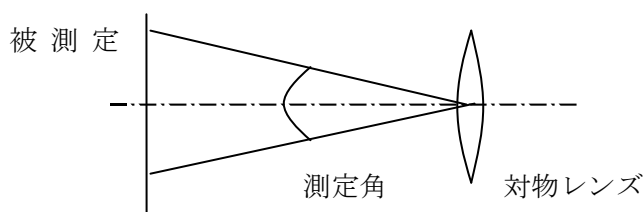
既に、Version 2.0 が発行されていますが、ここでは Version 1.0 についての記載とします。

(※注 6-2-2)

開口角とは、被測定物から出た光で測定器に取り込まれる光線の範囲を表した角度のことです。



測定角とは、測定器が被測定物を見ている角度のことです。



(※注 6-2-3)

液晶の表示可能な最低輝度は約 0.5 cd/m^2 なので、色彩計にはそれ以下の測定範囲仕様が必要であると言えます。

(※注 6-2-4)

規格本文は以下の HOME Page から購入することができます。

<http://www.vesa.org/>

6-3 sRGBとは

従来より、画像出力機器(ディスプレイ、プリンター)が出力する画像と実物の色に差があることが課題となっていました。

これを解決する手段として、以下の2つの方法が提言されてきました。

- ① 画像入出力機器が、それぞれ色補正プロファイルによる補正を行う。
- ② 画像入出力機器が、それぞれが共有する色空間(標準色空間)をもつ。

①の方式としては、ICC(International Colour Consortium)による方法が有名です。この方法は、忠実な色再現が可能であるものの、対象画像に入出力色空間のプロファイルを添付する必要があり、機器に求められる処理負荷が重くなるという問題があります。

②の方式では、画像入出力機器が標準色空間に基づき画像情報を管理することにより、機器に大きな負荷をかけることなく実物に忠実な色再現を行うことができます。

sRGBとは、この②の方式に基づく標準色空間のひとつで、IEC-61966-2-1として制定されました。現在、デジタルカメラ、液晶、CRT、プリンターなど、代表的な画像入出力機器においてsRGB準拠の製品が発売されています。また、多くのWeb言語、OSにもデフォルト色空間として採用されるに至っています。

sRGBで定義された標準色空間を表 6-3-1、図 6-3-1 に示します。また、ディスプレイが目標とすべきリファレンス条件を表2に示します。

表 6-3-1 sRGBの色空間座標

	R	G	B
x	0.6400	0.3000	0.1500
y	0.3300	0.6000	0.0600

図 6-3-1 には従来よりCRTにおいて定義されているNTSC色空間も示しています。

sRGB色空間はNTSC色空間より小さいことが分かります。

sRGB色空間は、その表色範囲が比較的小さいため、銀塩写真が表現可能な色を再現できないなどの課題も残されており、この解決をめざした規格制定が審議されています。

表 6-3-2 sRGBのリファレンス条件

輝度レベル	80cd/m2
白色点	D65 (x=0.3127, y=0.3291)
ガンマ特性	$\gamma = 2.2$

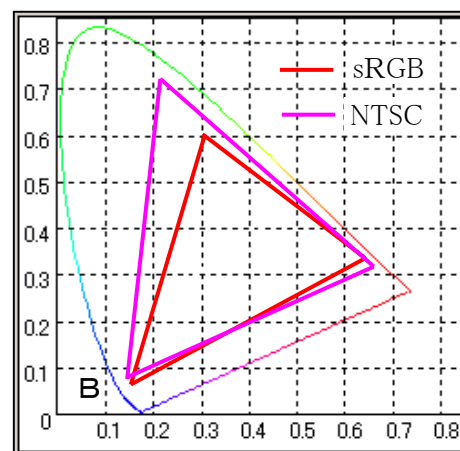


図 6-3-1 sRGBの色空間

参考文献

- ・杉浦博明 : 映像情報メディア学会誌, Vol.56 No.8 1247-1248(2002)