

Msako 技術情報

Kowa 製カメラ SC36M への対応

2011/07/27

大金康夫

概要

従来、Msako [1]が対応するカメラは、DirectShow [2]仕様の USB カメラなどのビデオキャプチャデバイスと Motion JPEG [3]仕様のネットワークカメラであった。今回、これらのカメラに加えて、Kowa 製カメラ SC36M [4]にも対応するようになった。

- Kowa 製カメラ SC36M で Msako による動体検知ができるようになった
- 動体を検知して出力基板に TTL 信号が出力されるまでのトータルの遅延時間は、4x4 ビニングの場合 30msec、1x1 ビニングの場合 70msec 程度に抑えられた

なお、この新しい対応を実現するために、このカメラ製品の購入者に提供される同社製の SDK を使用した。

Msako の仕組みと SC36M への対応

この新たな対応を実現するにあたり、まず、これまでの Msako の仕組みについて振り返ってみる。Msako で使用可能な USB カメラなどは、DirectShow 仕様のカメラであり、これらのカメラの映像をパソコン内でキャプチャするには、DirectShow のソフトウェアフレームワークに従うことになる。このフレームワークでは、フィルターグラフと呼ばれる部品化された映像処理用のソフトウェアモジュール同士を、それらが持つピンと呼んでいる入出力端子を連結することによってできる、ダイジェーチェーンによって一連の処理を実現する。Msako の場合、最上流にはビデオキャプチャデバイスのフィルターがあり、その後にビデオフォーマット変換などのフィルターが続く。そして、このフィルターグラフの最下流では、カメラのモデルに依存しない、統一された形式の映像データが得られるようにした。この映像データを使って Msako は動体検知を行う。一方、ネットワークカメラは、TCP/IP 通信を介して映像データを送出するものである。このようなデータを、一般的にビデオストリームと呼んだりもする。このようなタイプのカメラを扱うためには、もちろん DirectShow や Windows Media Format [5]というフレームワークを使う方法も考えられる。しかし、これらのフレームワークは、Windows Media Video (WMV) [6]という Microsoft 社固有の圧縮形式に束縛される。Msako では、これらのフレームワークを使用せず、Motion JPEG という単純で公開された圧縮形式に限定して対応することにした。実際、製品化されている多くのネットワークカメラは、Motion JPEG 圧縮形式である。この Motion JPEG

のビデオストリームをデコードするためには、複雑なフレームワークは必要ない。Msako は、スクラッチで書いた独自のデコードプログラムによって Motion JPEG 形式のカメラの映像を DirectShow タイプのカメラと同一の、統一された形式の映像データを得ている。ここで、統一された形式の映像データとは、24 ビット RGB 形式 (RGB24) のことである。今回は、Kowa の SDK に含まれるライブラリを使って映像データを取得し、さらに Msako 内でこのデータを統一された形式の映像データに変換することにした。これにより、従来からの Msako の仕組みを大幅に変えることなく SC36M へ対応することができた。

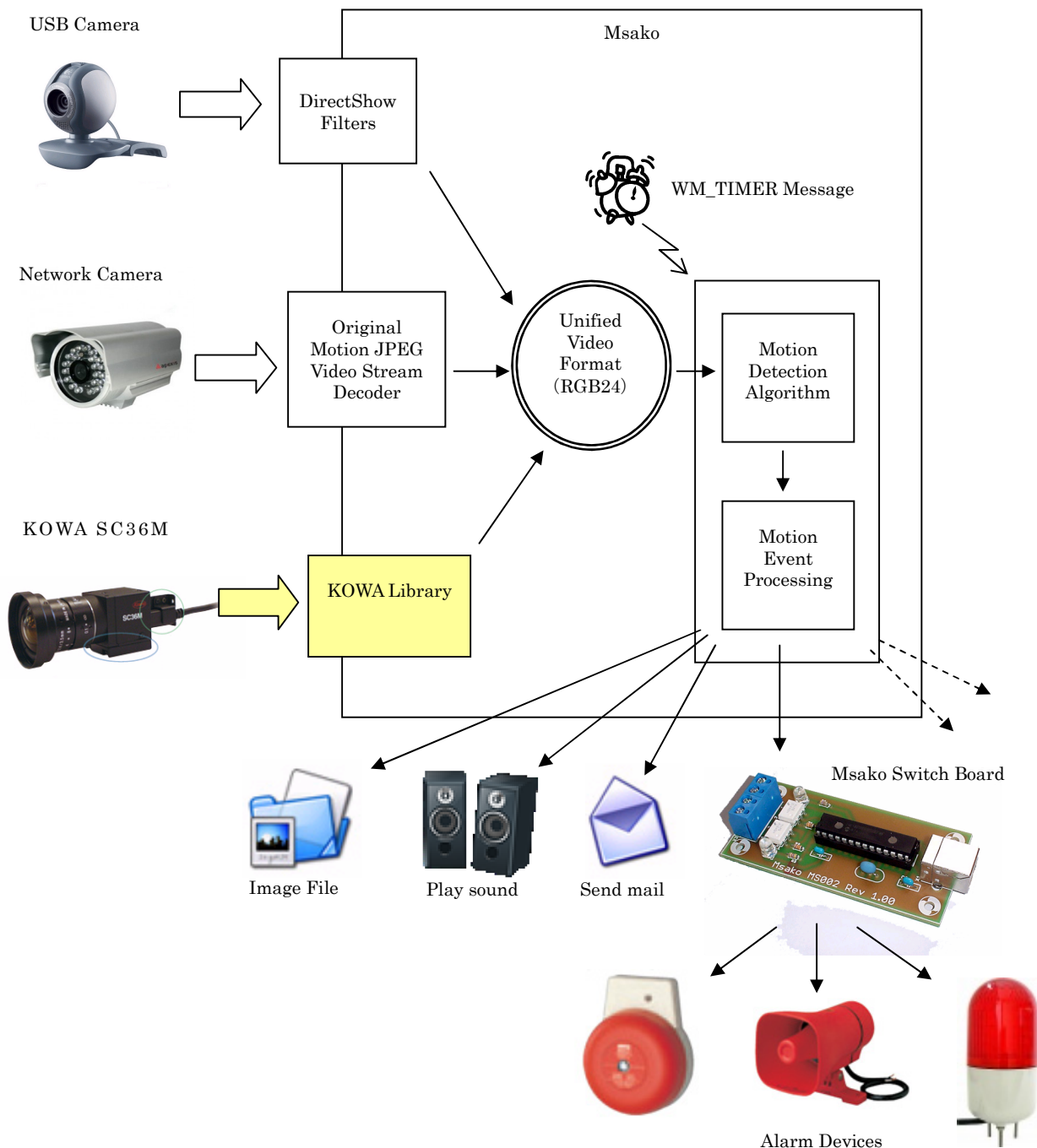


図 1 Msako の仕組み

遅延時間の短縮

今回は、このカメラを使用して単に動体検知ができるようにするだけでなく、動体検知をトリガーとして、できるだけ短い遅延時間で、外部機器を制御したいという、もう1つの目的があった。そこで、下図のように LED の発光をトリガーとして動体検知の遅延時間を測定してみた。

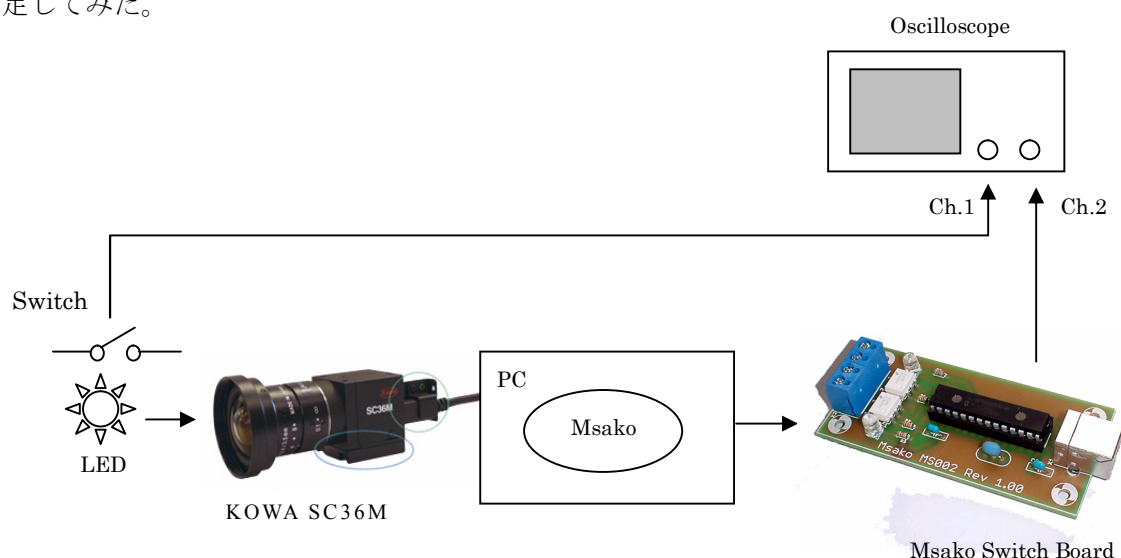


図2 遅延時間の測定

その結果、図3のように、95~135msec程度の遅延時間が観測された。DirectShow仕様の従来のUSBカメラでは、この遅延時間は200msecから300msec程度であった。これは、DirectShowのフィルタ間で行われる映像データの授受のためのバッファリングのために、フィルターの段数に応じた遅延が生じていると考えられる。DirectShow仕様ではないSC36Mは、この要因による遅延時間は生じないはずである。従って、この100msecほどの短縮はDirectShowを使用しないことによる効果であると考えられるが、後の残りの100msecの遅延の要因は依然不明である。

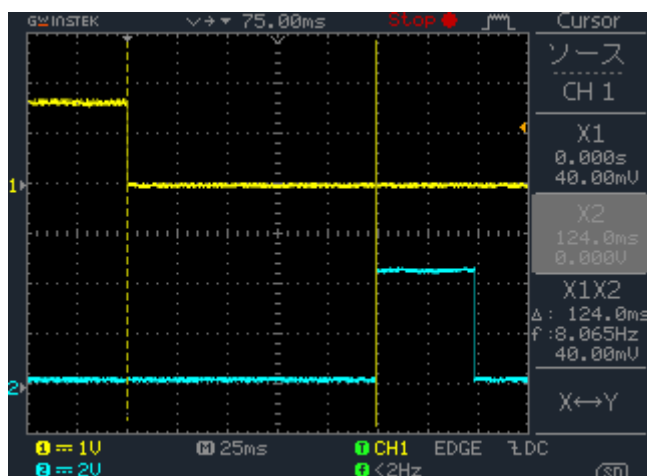


図3 遅延時間
(WVGA 4x4 Binning)

この不明な遅延時間をもう少し詳しく調べるために、Msako が映像のフレーム処理に入るたびに Msako Switch 基板を使って 10msec 幅のパルス信号を出力するよう、一時的に改造した。また同時に、この映像フレームの処理のタイミングとして、従来は Windows から配信される WM_TIMER メッセージ [7] を使用していたものを、より精度の高いマルチメディア用のタイマー [8] を使用するように変更した (変更 1)。

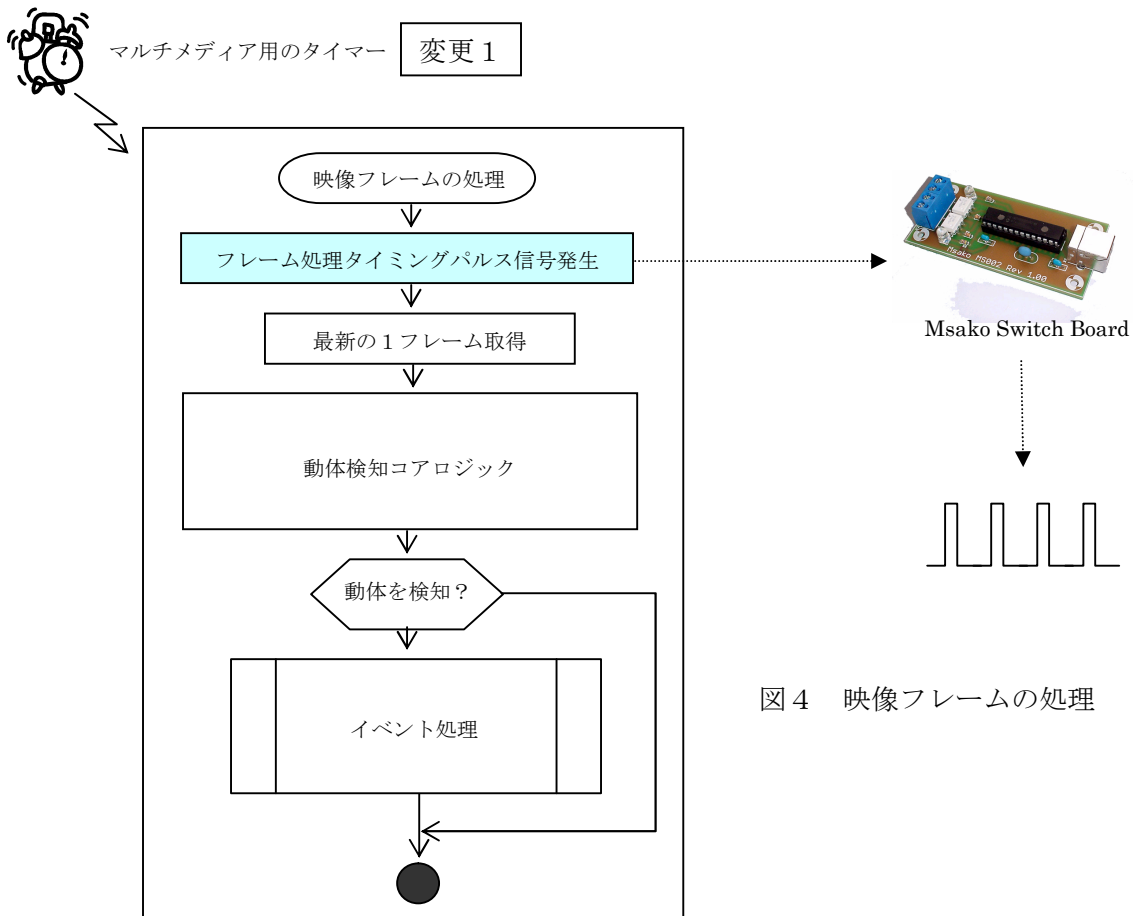


図4 映像フレームの処理

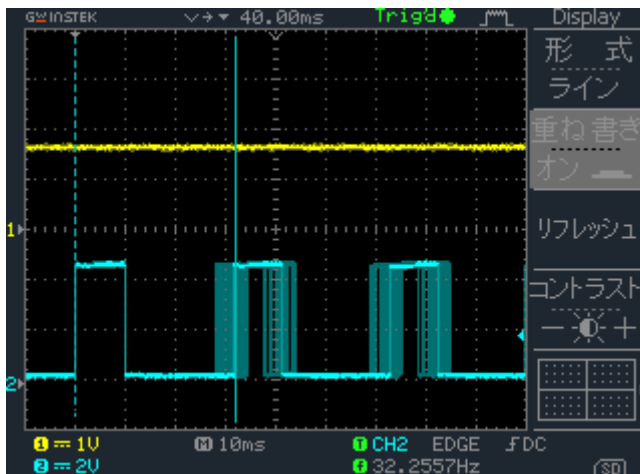


図5 動体を検知しない状態で基板から出力されるパルス信号 (30fps)

この結果、図5のとおり、動体を検知しない状態では、ほぼ設定したフレームレートで動体検知コアロジックが実行されていることが確認できた。次に、従来からある Msako の機能を使って、動体を検知した時に、この基板から 20msec 幅のパルスを出力するよう設定した。この設定は Msako の警報タブの外部機器制御にある。外部機器制御は、動体検知コアロジックで動体が検知された時、イベント処理の中で実行される。イベント処理では、まずイベント発生時刻を取得、保存し、その後、設定画面で設定した内容に応じてファイル保存などの動体検知発生時に行うべき処理が実行される。これらの処理の多くは、負荷分散の理由からスレッドで行うようにしている。(図6)

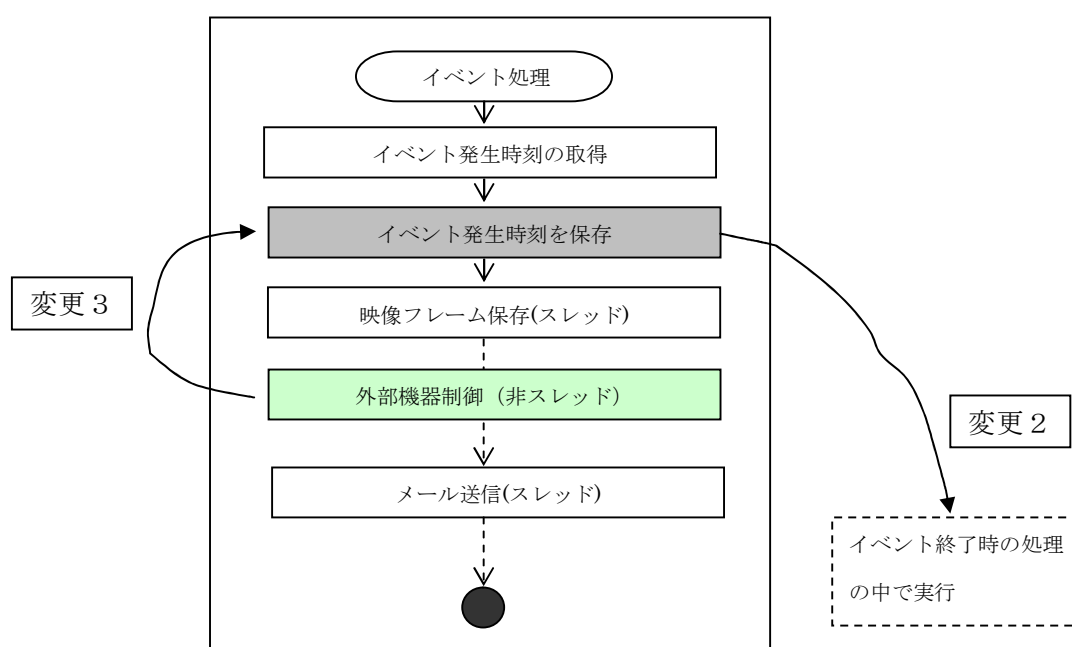


図6 イベント処理

ここで、イベント発生時刻を保存する処理は、XML形式でファイルに保存する処理であり、時間を要するXMLファイルのパーズやHDDアクセスを伴うので、イベント終了処理の中で実行するよう移動した(変更2)。さらに、スレッドで処理していた外部機器制御を非スレッドで処理、つまり即時に処理するように変更し、イベント処理の先頭に移動した(変更3)。

これらの変更の結果、図7のとおり、LED 発光後、最初に処理したフレームの中で外部機器制御が実行されるようになり、その遅延時間は約 30msec と劇的に減少した。

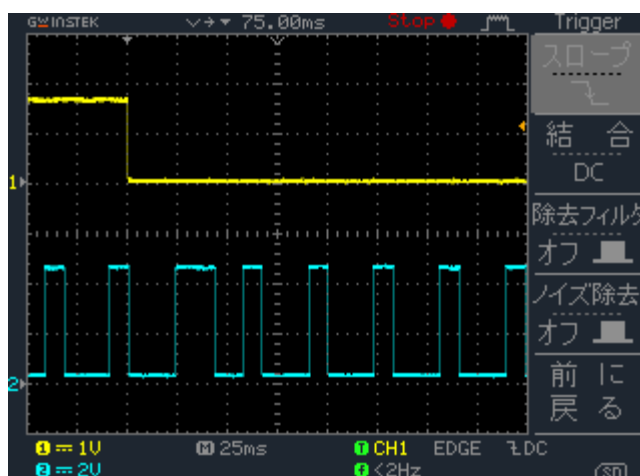


図7 イベント処理変更後の遅延時間 (WVGA 4x4 Binning)

遅延時間の測定では、一貫して取得した映像フレームのファイルへの保存は行っていないので、この 100msec から 30msec への 70msec もの遅延時間の減少は、イベント発生時刻を保存する処理を後回しにしたこと、つまり、変更 2 による効果としか考えられない。言い換えれば、XML ファイルへの書き込みによって 70msec の遅延が生じていたと考えられる。

最終的に、調査のために出力していた処理タイミングパルス信号を出力しないように戻して、もう 1 度遅延時間を測定してみた。その結果、間違いなく 4x4 ビニングの場合 30ms (図 8) 程度であった。同様に、2x2 ビニングの場合 50msec (図 9)、1x1 ビニングの場合 70msec (図 10) 程度であることが確認できた。

図 8
WVGA
4x4 Binning

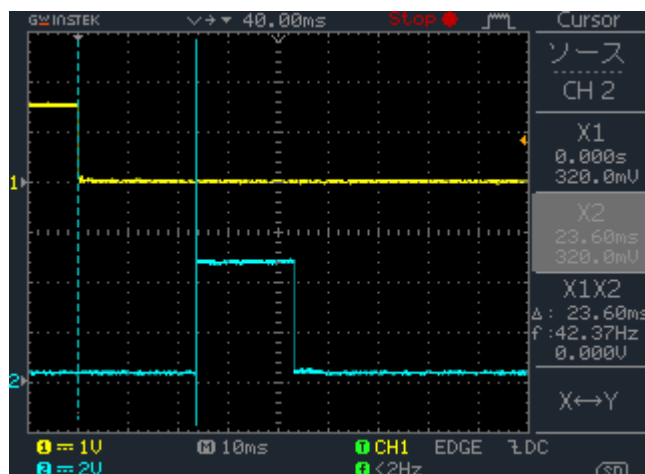


図 9
WVGA
2x2 Binning

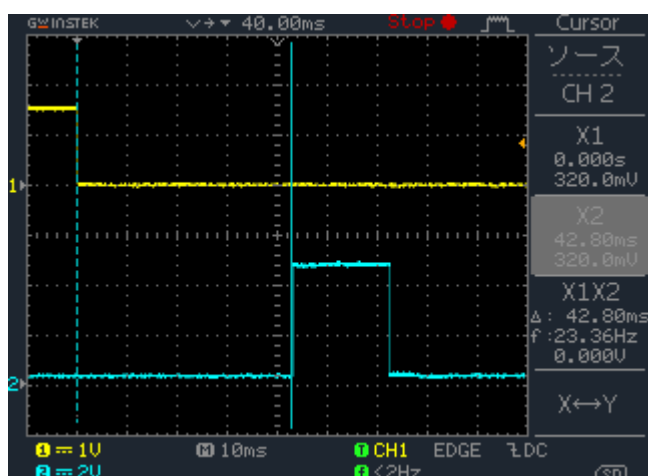
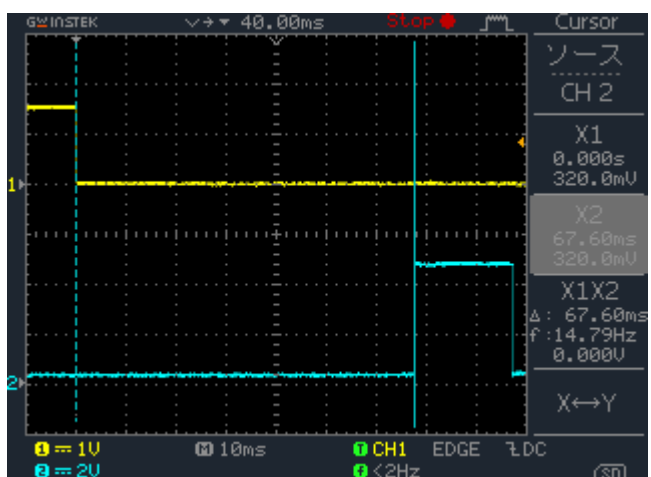


図 10
WVGA
1x1 Binning



カメラ設定画面の実装

従来 Msako が対象としてきたカメラがそうだったように、Kowa SC36M もモデル固有のカメラ属性を有する。今回のこのカメラへの対応でも、このモデル固有のカメラ属性の設定を Msako の画面から行えるようにした。この画面は、従来と同じ操作、つまり映像表示領域でマウスの右ボタンクリックで「カメラのプロパティ」メニューを選択することにより表示される。この画面で設定できるカメラ属性は、この製品に同梱されているアプリケーションとほぼ同等である。加えて、このアプリケーションにはない、自動露出のための測光エリアの重み付けができるようにした。

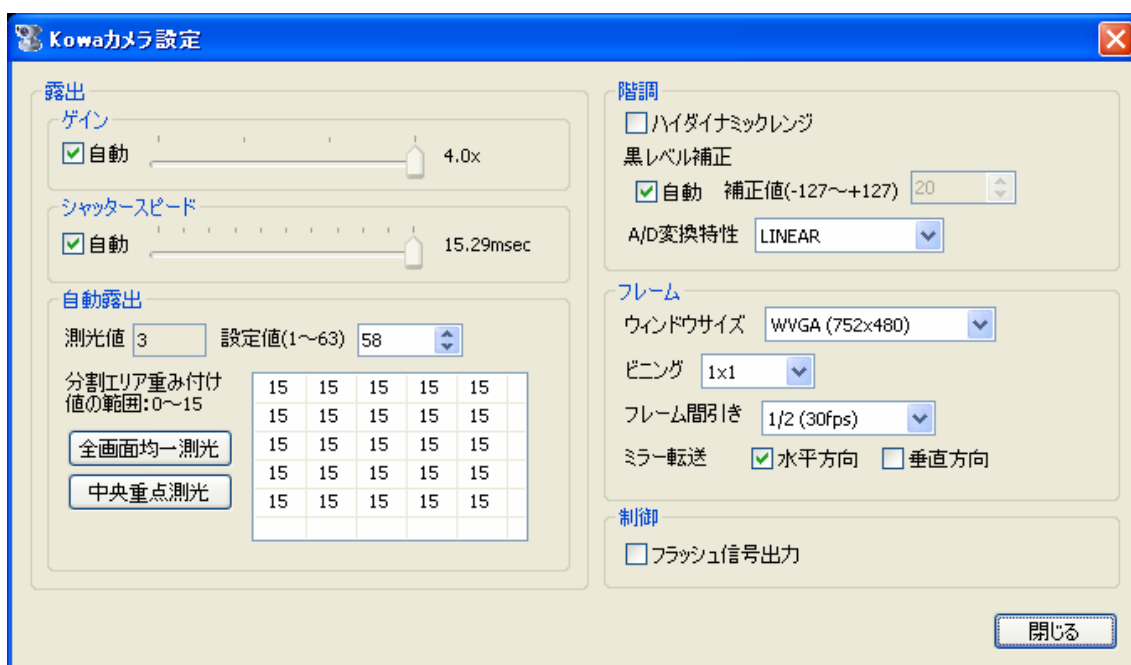


図 1 1 SC36M 用のカメラ設定画面

用語の解説

[1] Msako

<http://homebrew.jp/show?page=1262>

[2] DirectShow

<http://ja.wikipedia.org/wiki/DirectShow>

[3] Motion JPEG

http://ja.wikipedia.org/wiki/Motion_JPEG

[4] Kowa SC36M

<http://www.kowa.co.jp/opto/products/sc36m.htm>

[5] Windows Media Format

<http://msdn.microsoft.com/ja-jp/library/dd148612.aspx>

[6] Windows Media Video (WMV)

http://ja.wikipedia.org/wiki/Windows_Media_Video

[7] WM_TIMER Message

[http://msdn.microsoft.com/en-us/library/ms644902\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms644902(v=vs.85).aspx)

[8] Multimedia Timers

[http://msdn.microsoft.com/en-us/library/dd743609\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/dd743609(v=vs.85).aspx)