

2/20/2005

コンテック社の API-AIO ライブラリを使った MATLAB 用 AD/DA ドライバの開発

ここで作成するドライバは、コンテック社の AD/DA カード

ADA16-32/2(CB)F、AD12-8(PM)、それに ADI12-8(USB)GY での動作を確認している。API-AIO を利用しているためそれ以外の AD/DA カードでも一部を変更するだけで応用可能であると考えられる。通常これらのドライバを作成するには、ActiveX ライブラリを使うか、MEX 関数により Visual C 用ライブラリを呼び出すなどの方法が考えられるが、ここでは、MATLAB 6.5 より実装された Generic DLL による方法を紹介する。この方法を使うと、Visual C++ などの言語を用いてコンパイルする必要がなく、MATLAB 言語上でコンパイルすることなく動作するため、デバックやカスタマイズが容易にできる利点がある。ここでは、コンテック社の AD/DA カードの例を示すが、これ以外でも Visual C、Visual Basic 用のドライバを提供している場合には、同様のテクニックを使い MATLAB 上から直接操作できるようになる。

MATLAB 用 ADDA ライブラリ開発例

まず、コンテック社のホームページから API-AIO ライブラリをインストールする。すると、Visual C++、Visual Basic などのサンプルプログラムもインストールされる。Visual C++用のサンプルの中にある、

```
caio.h
```

を MATLAB カレントフォルダにコピーする。Visual Basic 用の caio.vb などの中を見ると、例えば、aioinit 関数の定義を見てみると

```
Declare Function AioInit Lib "CAIO.DLL" (ByVal DeviceName As String, ByVal Id As Short) As Integer
```

と書いてあり、ダイナミックリンクするライブラリは、caio.dll であることがわかる。これらより、loadlibrary 関数を使い、

```
loadlibrary('caio.dll','caio.h');
```

と実行する。このようにすることで、MATLAB 上にライブラリをロードできる。AD 変換のアルゴリズムの汎用性を考え、コンソールサンプルのデバイスバッファを使用した複数チャンネル対応アナログ入力のプログラム

```
AIOWDM¥Sample¥Console¥Ai¥DevBuf¥Ai¥Ai.c
```

をベースに作成に考えてみる。AD カードからのデータの取得をまとめると以下のようになる。

1	デバイスの初期化	AioInit AioResetDevice
---	----------	---------------------------

2	アナログ入力レンジの選択	AioSetAiRangeAll
3	アナログ入力のための条件設定	AioSetAiChannels AioSetAiMemoryType AioSetAiClockType AioSetAiSamplingClock AioSetAiStartTrigger AioSetAiStopTrigger AioSetAiStopTimes
4	アナログ入力を開始	AioResetAiMemory AioStartAi AioGetAiStatus AioGetAiSamplingCount
5	データの取り込みと終了	AioGetAiSamplingDataEx AioExit

見通しを良くするためエラー処理を省き手順どおり MATLAB 用に機能を書き換えるとうぎのようになる。

1	デバイスの初期化	loadlibrary('caio.dll','caio.h'); Id=0;DeviceName='AIO000'; [Ret,DeviceName,Id] = calllib('caio','AioInit',DeviceName, Id); Ret=calllib('caio','AioResetDevice',Id);
2	アナログ入力レンジの選択	PM10=0;%±10V Ret=calllib('caio','AioSetAiRangeAll',Id,PM10);
3	アナログ入力のための条件設定	channel=1 ; Ret=calllib('caio','AioSetAiChannels',Id,channel); Ret=calllib('caio','AioSetAiMemoryType',Id,0); Ret=calllib('caio','AioSetAiClockType',Id,0); Ret=calllib('caio','AioSetAiSamplingClock',Id,(interval*1000000.)); Ret=calllib('caio','AioSetAiStartTrigger',Id,0); Ret=calllib('caio','AioSetAiStopTrigger',Id,0); Ret=calllib('caio','AioSetAiStopTimes',Id,data_num);

4	アナログ入力を開始	<pre>Ret=calllib('caio','AioResetAiMemory',Id); Ret=calllib('caio','AioStartAi',Id); AIS_BUSY=00000001; AiStatus = 1;AiSamplingCount=0; while bitand(double(AiStatus),AIS_BUSY)==AIS_BUSY; [Ret,AiStatus]=calllib('caio','AioGetAiStatus',Id,AiStatus); [Ret,AiSamplingCount]= calllib('caio','AioGetAiSamplingCount',Id, AiSamplingCount); end</pre>
5	データの取り込みと終了	<pre>AiSamplingTimes = data_num; AiData=libpointer('int32Ptr',zeros(1,data_num*channel)); [Ret,AiSamplingTimes,AiData]=calllib('caio','AioGetAiSamplingDataEx',Id, AiSamplingTimes, AiData); data=reshape(data,channel,data_num); Ret = calllib('caio','AioExit',Id); unloadlibrary('caio');</pre>

以上を踏まえ、使いやすい形で MATLAB 関数としてまとめてみると、次のようになる。

contec_ad.m

```
function AiData=contec_ad(interval, data_num, AiChannels)
% contec_ad.m
% The calling syntax is:
% AiData = contec_ad(interval, data_num, AiChannels);
% Kazuyuki Kobayashi Feb 21, 2005
% Copyright (c) 2005 Geroc(c)
% All Rights Reserved
% for Windows版高機能アナログ入出力ドライバ API-AIO(WDM)
%これを使用する際には、コンテック社のドライバに含まれるcaio.hが必要です。
% ADA16-8/2(CB)L AD112-8(USB)GY AD128PM でテスト済み
% Windows版高機能アナログ入出力ドライバ API-AIO(WDM)がサポートしているカードで動作可
% ただし、入力レンジなどの設定をチューニングする必要あり、
if exist('caio.h')~= 2;
disp('これを使用する際には、コンテック社のドライバAPI-AIO(WDM)に含まれるcaio.hが必要です。');
return;
end
if ~libisloaded('caio');loadlibrary('caio.dll','caio.h');end
Id=0;DeviceName='AI0000';
AiData=[];
[Ret, DeviceName, Id]=calllib('caio','AioInIt', DeviceName, Id);
if Ret ~= 0
% [Ret, ErrorString]=calllib('caio','AioGetErrorString', Ret, char(zeros(1, 256)));
% fprintf('AioInIt = %d : %s\n', Ret, ErrorString);disp('error');
fprintf('AioInIt = %d %n', Ret);
Ret = calllib('caio','AioExit', Id );
unloadlibrary('caio');
return;
end
Ret=calllib('caio','AioResetDevice', Id);
if Ret~=0;
% [Ret, ErrorString]=calllib('caio','AioGetErrorString', Ret, char(zeros(1, 256)));
```

```
% fprintf(' AioResetDevice = %d : %s\n', Ret, ErrorString);
  fprintf(' AioResetDevice = %d\n', Ret);
  unloadlibrary(' caio');
  return;
end
% ±10V      : 0 |0~10V      : 50
% ±5V       : 1 |0~5V       : 51
% ±2.5V     : 2 |0~4.095V : 52
% ±1.25V    : 3 |0~2.5V    : 53
% ±1V       : 4 |0~1.25V   : 54
% ±0.625V   : 5 |0~1V     : 55
% ±0.5V     : 6 |0~0.5V   : 56
% ±0.3125V  : 7 |0~0.25V  : 57
% ±0.25V    : 8 |0~0.1V   : 58
% ±0.125V   : 9 |0~0.05V  : 59
% ±0.1V     :10 |0~0.025V : 60
% ±0.05V    :11 |0~0.0125V : 61
% ±0.025V   :12 |0~20mA    :100
% ±0.0125V  :13 |4~20mA    :101
% 1~5V      :150
PM10=0;%±10V
Ret=calllib(' caio', ' AioSetAiRangeAll', Id, PM10);
if Ret~=0;
% [Ret, ErrorString]=calllib(' caio', ' AioGetErrorString', Ret, char(zeros(1, 256)));
% fprintf(' AioSetAiRangeAll = %d : %s\n', Ret, ErrorString);
  fprintf(' AioSetAiRangeAll = %d\n', Ret);
  Ret = calllib(' caio', ' AioExit', Id );
  unloadlibrary(' caio');
  return;
end
Ret=calllib(' caio', ' AioSetAiChannels', Id, AiChannels);
if Ret~=0;
% [Ret, ErrorString]=calllib(' caio', ' AioGetErrorString', Ret, char(zeros(1, 256)));
% fprintf(' AiSetAiChannels = %d : %s\n', Ret, ErrorString);
  fprintf(' AiSetAiChannels = %d\n', Ret);
  unloadlibrary(' caio');
  return;
end
Ret=calllib(' caio', ' AioSetAiMemoryType', Id, 0);
if Ret~=0;
% [Ret, ErrorString]=calllib(' caio', ' AioGetErrorString', Ret, char(zeros(1, 256)));
% fprintf(' AioSetAiMemoryType = %d : %s\n', Ret, ErrorString);
  fprintf(' AioSetAiMemoryType = %d\n', Ret);
  unloadlibrary(' caio');
  return;
end
Ret=calllib(' caio', ' AioSetAiClockType', Id, 0);
if Ret~=0;
% [Ret, ErrorString]=calllib(' caio', ' AioGetErrorString', Ret, char(zeros(1, 256)));
% fprintf(' AioSetAiClockType = %d : %s\n', Ret, ErrorString);
  fprintf(' AioSetAiClockType = %d\n', Ret);
  unloadlibrary(' caio');
  return;
end
Ret=calllib(' caio', ' AioSetAiSamplingClock', Id, (interval*1000000.));
if Ret~=0;
% [Ret, ErrorString]=calllib(' caio', ' AioGetErrorString', Ret, char(zeros(1, 256)));
% fprintf(' AioSetAiSamplingClock = %d : %s\n', Ret, ErrorString);
  fprintf(' AioSetAiSamplingClock = %d\n', Ret);
  unloadlibrary(' caio');
  return;
end
Ret=calllib(' caio', ' AioSetAiStartTrigger', Id, 0);
```

```
if Ret~=0;
% [Ret, ErrorString]=calllib(' caio', ' AioGetErrorString', Ret, char(zeros(1, 256)));
% fprintf(' AioSetAiStartTrigger = %d : %s\n', Ret, ErrorString);
  fprintf(' AioSetAiStartTrigger = %d\n', Ret);
  unloadlibrary(' caio');
  return;
end
Ret=calllib(' caio', ' AioSetAiStopTrigger', Id, 0);
if Ret~=0;
% [Ret, ErrorString]=calllib(' caio', ' AioGetErrorString', Ret, char(zeros(1, 256)));
% fprintf(' AioAiStopTrigger = %d : %s\n', Ret, ErrorString);
  fprintf(' AioAiStopTrigger = %d\n', Ret);
  unloadlibrary(' caio');
  return;
end
Ret=calllib(' caio', ' AioSetAiStopTimes', Id, data_num);
if Ret~=0;
% [Ret, ErrorString]=calllib(' caio', ' AioGetErrorString', Ret, char(zeros(1, 256)));
% fprintf(' AioSetStopTimes = %d : %s\n', Ret, ErrorString);
  fprintf(' AioSetStopTimes = %d\n', Ret);
  unloadlibrary(' caio');
  return;
end
Ret=calllib(' caio', ' AioResetAiMemory', Id);
if Ret~=0;
% [Ret, ErrorString]=calllib(' caio', ' AioGetErrorString', Ret, char(zeros(1, 256)));
% fprintf(' AioResetAiMemory = %d : %s\n', Ret, ErrorString);
  fprintf(' AioResetAiMemory = %d\n', Ret);
  Ret = calllib(' caio', ' AioExit', Id );
  unloadlibrary(' caio');
  return;
end
Ret=calllib(' caio', ' AioStartAi', Id);
if Ret~=0;
% [Ret, ErrorString]=calllib(' caio', ' AioGetErrorString', Ret, char(zeros(1, 256)));
% fprintf(' AioStartAi = %d : %s\n', Ret, ErrorString);
  fprintf(' AioStartAi = %d\n', Ret);
  Ret = calllib(' caio', ' AioExit', Id );
  unloadlibrary(' caio');
  return;
end
AIS_BUSY=00000001;
AiStatus = 1;AiSamplingCount=0;
while bitand(double(AiStatus), AIS_BUSY)==AIS_BUSY;
  [Ret, AiStatus]=calllib(' caio', ' AioGetAiStatus', Id, AiStatus);
  if Ret~=0;
% [Ret, ErrorString]=calllib(' caio', ' AioGetErrorString', Ret, char(zeros(1, 256)));
% fprintf(' AioGetAiStatus = %d : %s\n', Ret, ErrorString);
    fprintf(' AioGetAiStatus = %d\n', Ret);
    Ret = calllib(' caio', ' AioExit', Id );
    unloadlibrary(' caio');
    return;
  end
  [Ret, AiSamplingCount]= calllib(' caio', ' AioGetAiSamplingCount', Id, AiSamplingCount);
  if Ret~=0;
% [Ret, ErrorString]=calllib(' caio', ' AioGetErrorString', Ret, char(zeros(1, 256)));
% fprintf(' AioGetAiSamplingCount = %d : %s\n', Ret, ErrorString);
    fprintf(' AioGetAiSamplingCount = %d\n', Ret);
    Ret = calllib(' caio', ' AioExit', Id );
    unloadlibrary(' caio');
    return;
  end
end
end
```

```
AiData=libpointer('singlePtr',zeros(1,data_num*AiChannels));
AiSamplingTimes = data_num;
[Ret,AiSamplingTimes,AiData]=calllib('caio','AioGetAiSamplingDataEx',Id,AiSamplingTimes,AiData);
if Ret~=0;
% [Ret,ErrorString]=calllib('caio','AioGetErrorString',Ret,char(zeros(1,256)));
% fprintf('AioGetAiSamplingDataEx = %d : %s\n',Ret,ErrorString);
fprintf('AioGetAiSamplingDataEx = %d\n',Ret);
Ret = calllib('caio','AioExit',Id);
unloadlibrary('caio');
return;
end
AiData=reshape(AiData,AiChannels,data_num);
Ret = calllib('caio','AioExit',Id);
if Ret~=0;
% [Ret,ErrorString]=calllib('caio','AioGetErrorString',Ret,char(zeros(1,256)));
% fprintf('AioExit = %d : %s\n',Ret,ErrorString);
fprintf('AioExit = %d\n',Ret);
unloadlibrary('caio');
return;
end
end
if libisloaded('caio');unloadlibrary('caio');end
```

としてなる。使用方法は、例えば、0.01秒間隔でデータ1024点、1チャンネルのデータを取得したい場合には、

```
data=contec_ad(0.01,1024,1);
```

とする。この場合、時間は、 0.01×1024 秒なので、約10秒程度かかるので、

```
tic ;data= contec_ad(0.01,1024,1) ;toc
```

とすると、実行できているか確認することができる。ダイナミックライブラリをロードし実行するため、一般には、サンプリング時間10秒+ α だけ時間がかかるはずである。この要領でDA、DIOなどのデバイスドライバもMATLAB言語のみで開発することができる。

アナログ出力の実装例

ここでは、ADA16-8/2(CB)L AD128PMの2つのAD/DAカードにより検証した例を示す。コンテック社のC言語で記述されたコンソールバージョンをベースに

```
AIOWDM¥Sample¥Console¥Ao¥MultiAo¥MultiAo.c
```

MATLAB用に書き換えてみる。

contec_multiao.m

```
function contec_multiao(AoData);
% multiao.m
% The calling syntax is:
% contec_multiao(AoData);
% Kazuyuki Kobayashi Feb 21, 2005
% Copyright (c) 2005 Gerox(c)
% All Rights Reserved
% ADA16-8/2(CB)LAD128PM でテスト済み
% Windows 版高機能アナログ入出力ドライバ API-AIO(WDM)がサポートしているカードで動作可
% ただし、出力レンジなどの設定をチューニングする必要あり、
if exist('caio.h')~= 2;
disp('これを使用するには、コンテック社のドライバAPI-AIO(WDM)に含まれる caio.h が必要です。');
return;
end
loadlibrary('caio.dll','caio.h');
Id=0;DeviceName='AIO000';
```

```
AoChannels=length(AoData);
[Ret,DeviceName,Id]=calllib('caio','AioInit',DeviceName, Id);
if Ret ~= 0
% [Ret,ErrorString]=calllib('caio','AioGetErrorString',Ret,char(zeros(1,256)));
% fprintf('AioInit = %d : %s¥n',Ret,ErrorString);
  fprintf('AioInit = %d ¥n',Ret);
  unloadlibrary('caio');
  return;
end
AoRange=0;
Ret = calllib('caio','AioSetAoRangeAll',Id,AoRange);
if Ret~=0;
% [Ret,ErrorString]=calllib('caio','AioGetErrorString',Ret,char(zeros(1,256)));
% fprintf('AioSetAiRangeAll = %d : %s¥n',Ret,ErrorString);
  fprintf('AioSetAiRangeAll = %d¥n',Ret);
  unloadlibrary('caio');
  return;
end
MaxAoChannels = 0;
[Ret,MaxAoChannels]=calllib('caio','AioGetAoMaxChannels',Id,MaxAoChannels);
if Ret~=0;
% [Ret,ErrorString]=calllib('caio','AioGetErrorString',Ret,char(zeros(1,256)));
% fprintf('AioGetAoMaxChannels = %d : %s¥n',Ret,ErrorString);
  fprintf('AioGetAoMaxChannels = %d ¥n',Ret);
  Ret = calllib('caio','AioExit',Id );
  unloadlibrary('caio');
  return;
end
if(MaxAoChannels < AoChannels) AoChannels = MaxAoChannels;end
pAoData=libpointer('singlePtr',AoData);
[Ret,AoData] = calllib('caio','AioMultiAoEx',Id, AoChannels, AoData);
if Ret~=0;
% [Ret,ErrorString]=calllib('caio','AioGetErrorString',Ret,char(zeros(1,256)));
% fprintf('AioAoData = %d : %s¥n',Ret,ErrorString);
  fprintf('AioAoData = %d ¥n',Ret);
  unloadlibrary('caio');
  return;
end
Ret = calllib('caio','AioExit',Id );
if Ret~=0;
% [Ret,ErrorString]=calllib('caio','AioGetErrorString',Ret,char(zeros(1,256)));
% fprintf('AioExit = %d : %s¥n',Ret,ErrorString);
  fprintf('AioExit = %d ¥n',Ret);
  unloadlibrary('caio');
  return;
end
if libisloaded('caio');unloadlibrary('caio');end
```

この場合には、

```
contec_multiao([1 2]);
```

この場合、1チャンネルに1V、2チャンネルに2Vを出力する例である。