

# 「dsPIC33FJ256GP710 利用メモ」

デモ基板入手方法、FFT 処理、RS-232 通信、CDMA 受信処理、I C D 2 故障修理等を掲載しました。

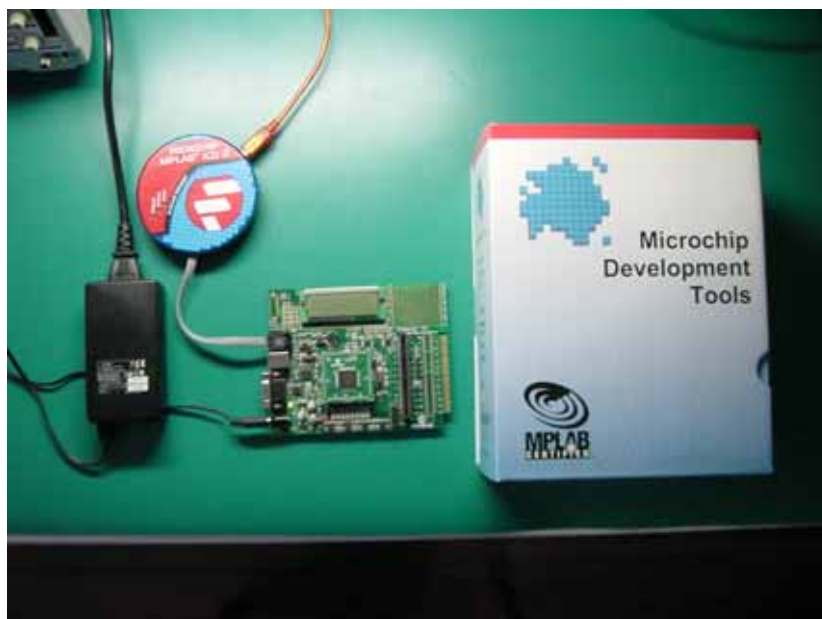
初版      2008年3月19日  
第6版      2009年2月 6日

北吉 均

## 1．はじめに

Microchip 社の dsPIC33F シリーズは、16bit 40MIPS かつ積和演算を 1 cycle で処理できる優れものです。このチップは安い、ROM/RAM/ADC 等内蔵、開発環境も安価に入手可能です。特に、内蔵 12bit の ADC が 2CH 同測できるので I/Q 信号を扱う無線通信の実験には最適です。ただし、まだ発売されて日が浅いせいか解説書の類や H.P.上の解説記事が殆んどありません。私が思うには、通信系の大学院生など計算機シミュレーションにばかり頼っていないで実験をもっとやるべきです。そこで、誰でも安価に実現できる実験装置のブログを書いてみようと思いつきました。ご意見、情報等のメールを頂ければ幸いです。

## 2．入手方法



上の写真は、Microchip 社から私が購入した開発キット (DV164033 Explorer 16 Kit) @299.99USD です。デモ基板、電源、C コンパイラ、デバッガ、サンプルプログラムなどのセットでこれがあればすぐに実験がはじめられます。このキットには PIC24FJ128GA010 と dsPIC33FJ256GP710 が付いていて両方のプログラム開発が可能です。ただし、このボードでは PIC24F の場合 4MIPS の性能しか得られません。また、デモ基板( DM240001 Explorer 16 Demo Board )のみでは@129.99USD、dsPIC33F J256GP710 チップは@7.83-5.56USD で下記 H.P.から購入できます。

<http://www.microchipdirect.com/PriceAvailabilityJump.aspx>

### 3 . dsPIC33F での FFT 処理

内蔵 2 C H 同測 A D C で I / Q 信号を取り込んで FFT 処理してみる。



A D 1、A D 2 とともに入力範囲 0-3.3 V であるために入力は C カップリングで +1.6V の DC オフセットを与えた。ADC の動作スペックは最大 500ksps であったが、SH セトリング時間に余裕を持たせ DMA を使わなかったので 200ksps/2CH の動作であった。FFT 処理は付属のライブラリ関数 FFTComplexIP( ) と BitReverseComplex( ) を用いる。ただし、これらの DSP ライブラリは Q.15 フォーマットの変数を扱うため、Float2Fract( ) および Fract2Float( ) の変数変換が必要になる。また、FFT 用三角関数テーブルの準備も必要で FFT\_factors[ ] に付属 dsPIC30F サンプルプログラム twiddleFactors.c からテーブルのコピーが必要にな

る。

上の写真では、入力信号の周波数と信号レベルを表示している。512 point の複素 FFT 処理時間は約 1ms であった。FFT 処理結果からスペクトルピークサーチを行って入力信号の周波数を求めると周波数分解能が 3 桁に満たないため、周波数スペクトルの補間処理（#1）を行い 5 桁の精度・分解能を得ている。

#1 北吉 均：“ショートタイム周波数スペクトル解析のための高分解能化”，  
信学論 A，vol. J76-A，no. 1，pp. 78-81，Jan. 1993.

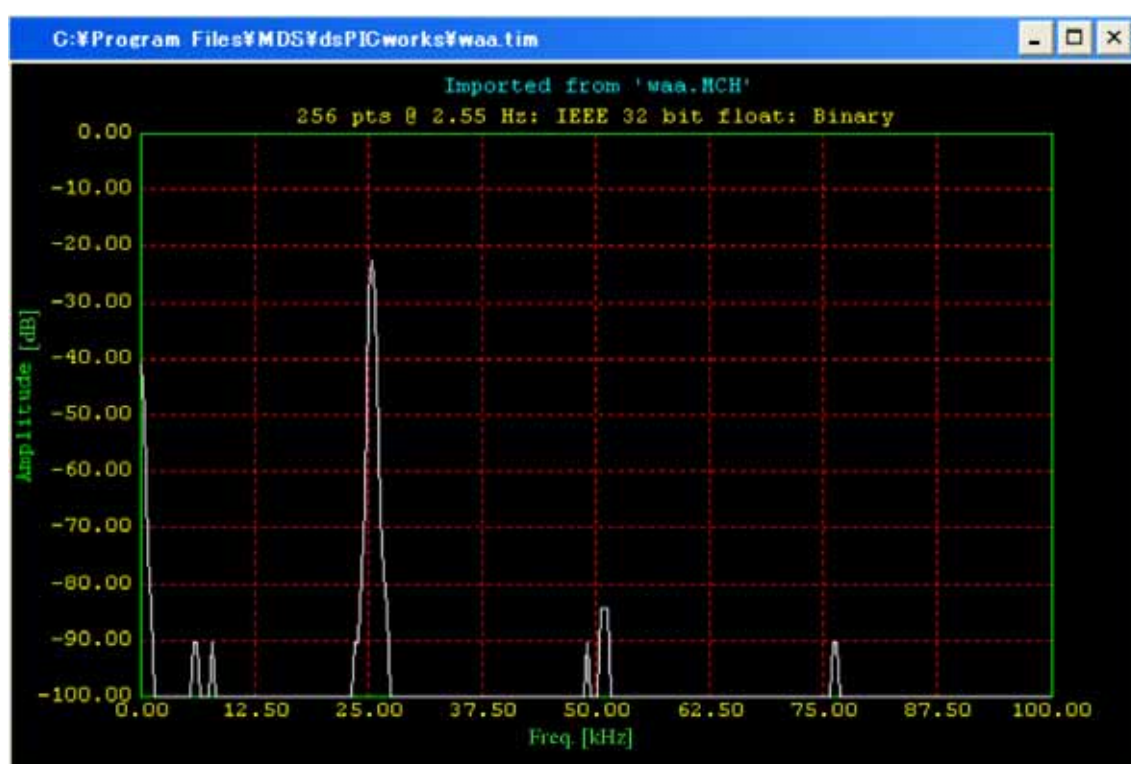


図 1 dsPIC33F を用いた FFT 処理例（25kHz 正弦波を 12bitADC で取り込んで DSP-FFT 処理した結果を dsPICworks で表示、Hanning 窓使用）

上の図 1 は、付属のデバッガを使ってプログラム変数配列に入っている周波数スペクトルをグラフ表示した結果である。信号処理プログラムでは処理途中の結果を確認する場合が多く便利である。この周波数スペクトル（入力 25kHz 正弦波、ただし 2 次 3 次歪あり）を見ると約 70dB のダイナミックレンジが得られていることが分かる。

#### 4. デモ基板とPC間接続 (RS-232)

デモ基板にはシリアル I/O ポート (RS-232) と接続ケーブルが付いており簡単に PC とデータ・コマンドのやり取りもできる。dsPIC33F のサンプルプログラムにはこの部分が含まれていないので PIC24F のサンプルから uart2.c と uart2.h をコピーしてくる必要がある。PIC24F はクロック 8MHz 設定になっているので SYSCLK を 80MHz に変更してボーレートが正しく設定されるようにしなければならない (dsPIC33F の PLL は 160MHz に設定: PLLFBD=0x00C6, CLKDIV=0x0008)。

RS-232 を使うには最初に UART2Init( ) をコールする。PC からコマンドを取り込む場合、関数 UART2IsPressed( ) の戻値が 1 になるまで待ってから関数 UART2GetChar( ) の戻値として 1 文字ずつ受け取ることができる。また、PC 側へ 1 文字送るには関数 UART2PutChar( ) を使えば良い。

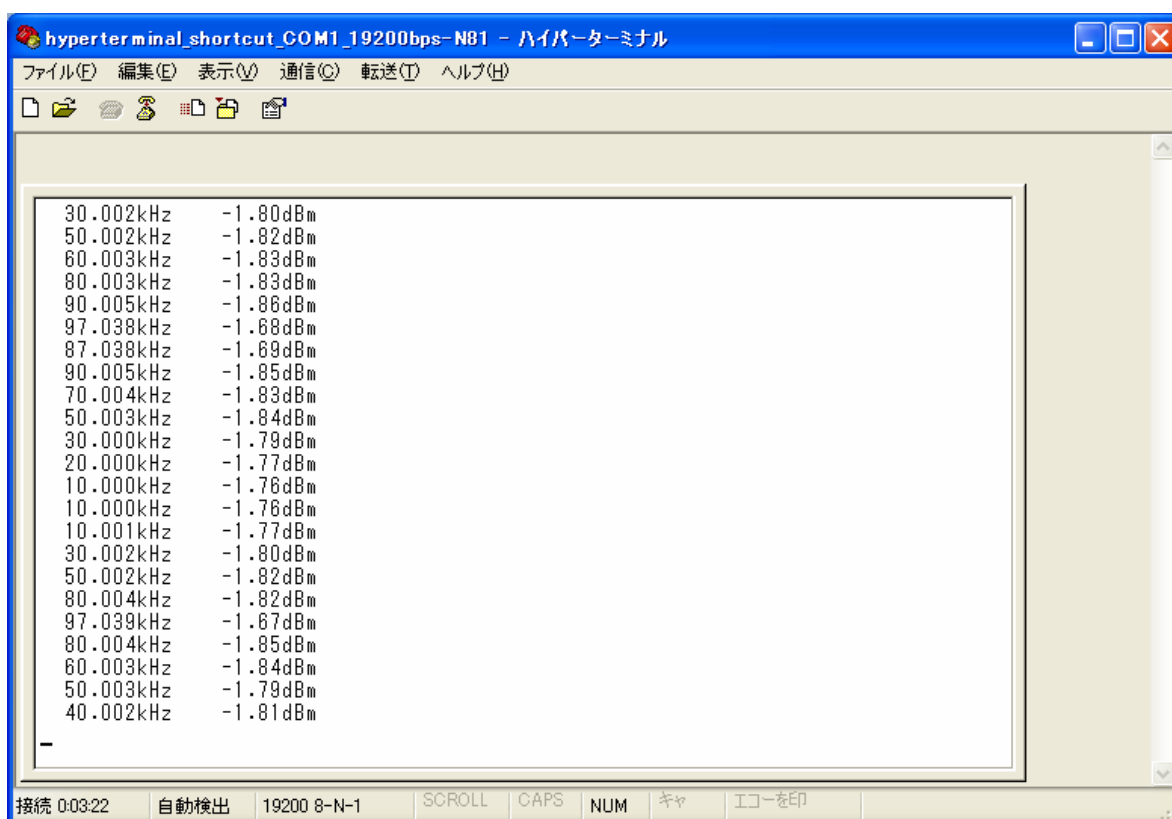


図2 dsPIC33F と PC 間の RS-232 による通信例 (FFT 補間ピークスpekトルの周波数とレベルを PC に取り込んで hyperterminal に表示)

上の図2はデモ基板で測定した FFT 補間ピークスpekトルの周波数とレベルを PC に取り込んで hyperterminal に表示した例である。入力信号は SG で与え

周波数は手動で変化させた。このようにデモ基板と P C 間の通信も比較的簡単に実現することができる。

第 3 版 2008 年 4 月 14 日

## 5 . C D M A の受信 ( スペクトル逆拡散 )

デモ基板を用いて C D M A 信号の受信を行ってみた。送信信号は、PIC16F684 を用いて M 系列を発生させてキャリア ( 2.45GHz ) 信号を BPSK 変調した。拡散率 512 で実験を行った。PIC16F684 は外部クロック 34MHz で動作、スペック保証値から外れてしまうが 60MHz 位までは余裕で動作する。デモ基板 ( Explorer 16 Kit ) 側では、受信してベースバンド I ・ Q に変換した信号を 12bit/200ksps で取り込みスペクトル逆拡散処理を行った。

下の図 3 は、受信信号複素 512 point の FFT 処理結果  $S_r(f)$  である。逆拡散処理は、送信 M 系列の FFT 処理結果  $S_o(f)$  ( 参照信号として固定する ) と前記受信信号の FFT 処理結果のクロススペクトル  $S_r(f) \cdot S_o^*(f)$  を得て IFFT 処理する。

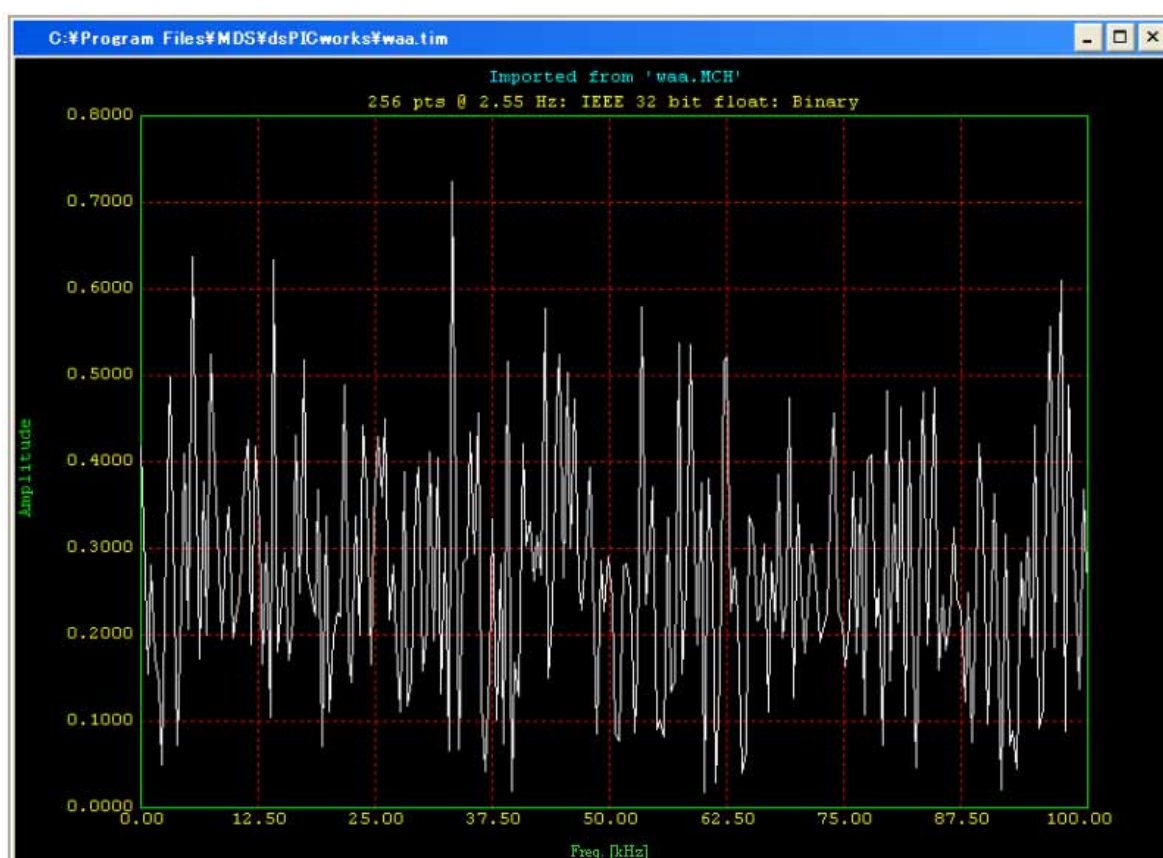


図 3 dsPIC33F による拡散率 512 の C D M A 受信信号周波数スペクトル解析例



下の図 4 は、逆拡散処理した受信 C D M A 信号の例である。この例では、受信信号レベルが十分でなかったためにノイズが比較的多いがノイズレベルに対して十分に大きな逆拡散インパルス信号が観測できる。

逆 拡 散 処 理 プ ロ グ ラ ム 中、IFFT 処 理 は 付 属 の ラ イ ブ ラ リ 関 数 IFFTComplexIP()と BitReverseComplex()を用いることができるが、クロススペクトル演算ライブラリが無く C 言語で記述したために処理時間が 100ms を超えてしまった。特に、Float2Fract()は 1 回当たり 36  $\mu$  s 掛かってしまう。また、FFT 及び IFFT 処理は固定小数点となってしまうために演算による桁落ちが深刻でクロススペクトルの振幅不足が生じないように演算 AGC (ブロック・スケーリング) 処理も行っている。

以上、dsPIC33F を高速信号処理に利用するためにはアセンブラによるプログラミングが不可欠であると実感した。

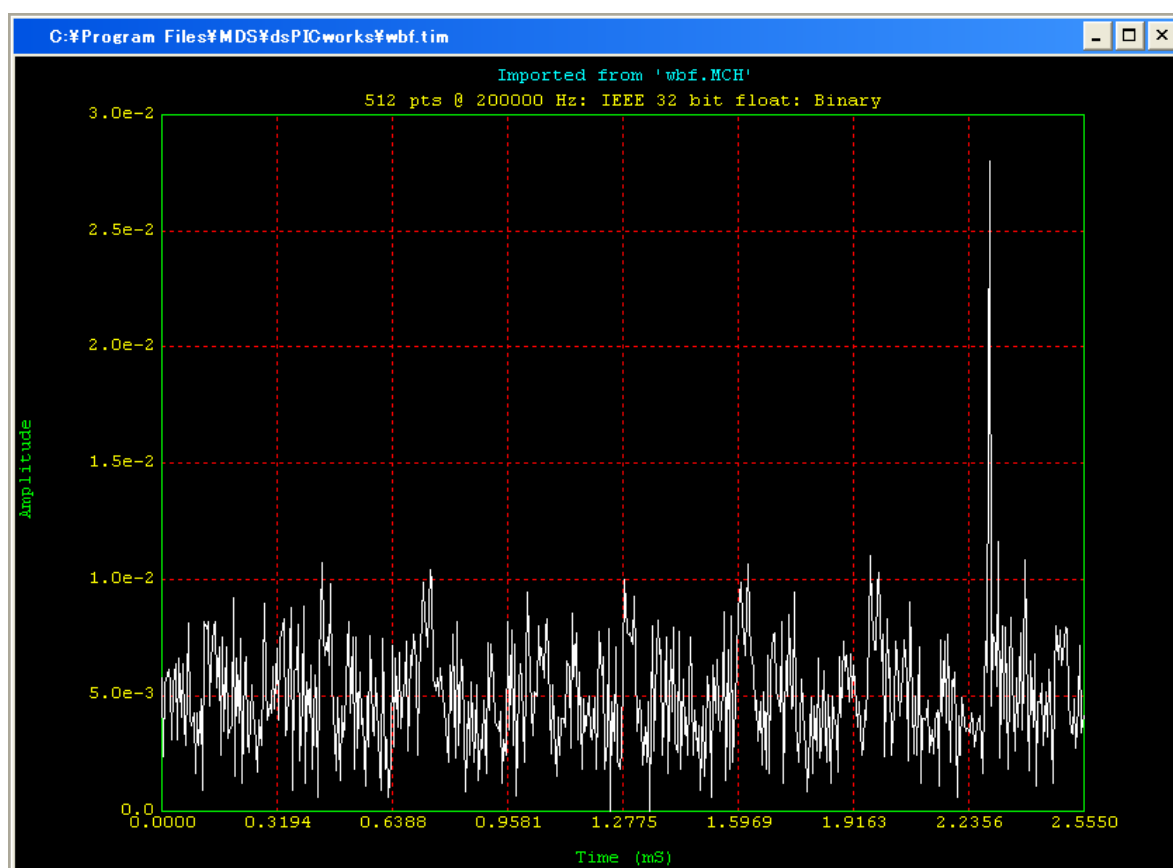


図 4 dsPIC33F による拡散率 512 の C D M A 受信信号の逆拡散処理例

## 6. CDMA受信信号の色々（妨害波やキャリア周波数オフセットの影響）

デモ基板を用いたCDMA受信機を使って色々な条件でのCDMA受信信号の解析を行った。CW妨害波が混入受信された場合や送受信でキャリア周波数にオフセットがある場合を想定して実空間伝送実験を行い dsPIC33F を用いたCDMA受信機での逆拡散解析結果を示す。

下の図5は、キャリア周波数 2.45GHz、拡散率 512、チップレート 200kcps、拡散コード周期 2.56ms のCDMA受信信号の逆拡散処理結果であり、妨害波及び送受信でのキャリア周波数オフセットが無い場合である。逆拡散インパルスピークレベル 0.36 に対してノイズレベル 0.025 は非常に低い。

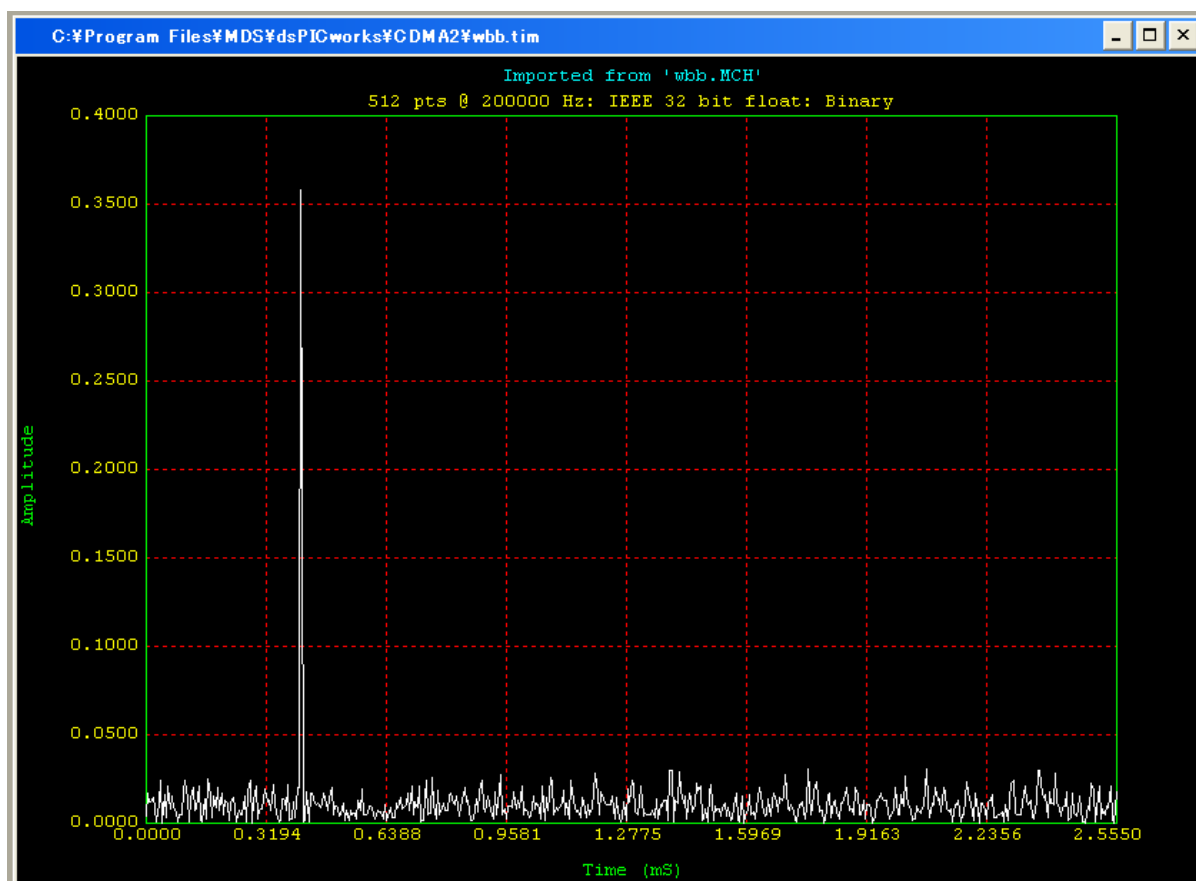


図5 妨害波及びキャリア周波数オフセットが無い場合のCDMA受信信号の逆拡散解析結果

下の図6は、図5で示した実験においてC D M A受信信号振幅の約2倍の振幅でキャリア周波数のC W妨害波を印加受信した場合の逆拡散解析結果である。C D M A受信信号のレベルは図5の場合と同じであるがC W妨害波レベルが大きいために受信信号に演算 AGC スケーリング処理が適応された結果、図5の場合よりもC D A M受信信号振幅は約 1/2 に小さくなっている。結果、逆拡散インパルスピークレベル0.18 に対してノイズレベル0.015 となり、相関ピークは下がったもののS N比は殆んど劣化していない。

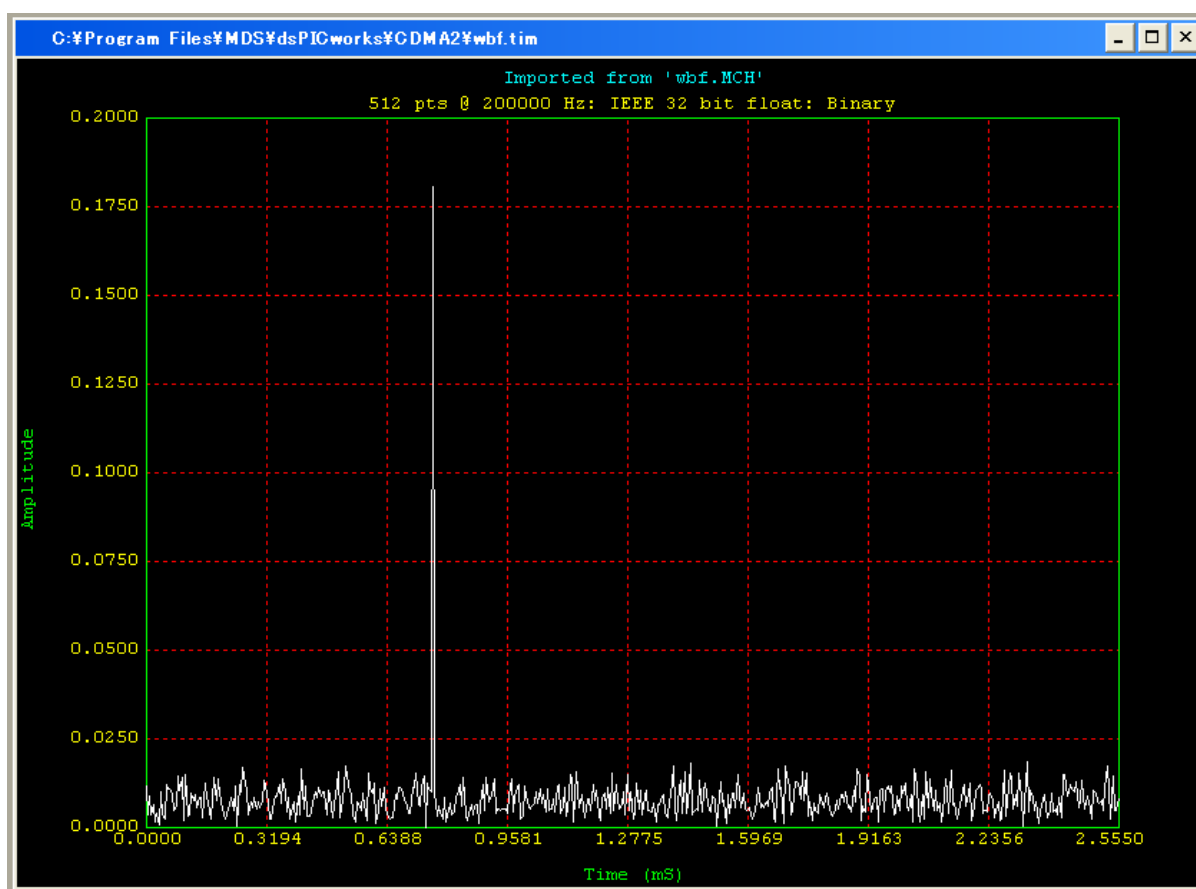


図6 C D M A受信信号振幅の約2倍の振幅でキャリア周波数のC W妨害を印加した場合のC D M A逆拡散解析結果



下の図7は、図5で示した実験においてC D M A受信信号振幅の約4倍の振幅でキャリア周波数のC W妨害波を印加受信した場合の逆拡散解析結果である。C D M A受信信号のレベルは図5の場合と同じであるがC W妨害波レベルが大きいため受信信号に演算 AGC スケーリング処理が適応された結果、図5の場合よりもC D A M受信信号振幅は約 1/4 に小さくなっている。結果、逆拡散インパルスピークレベル0.1に対してノイズレベル0.015 となり、相関ピークは更に下がりS N比も劣化している。これは、12bitADC やFFT 及び IFFT 演算の量子化雑音がノイズレベルに影響したためだと考えられる。

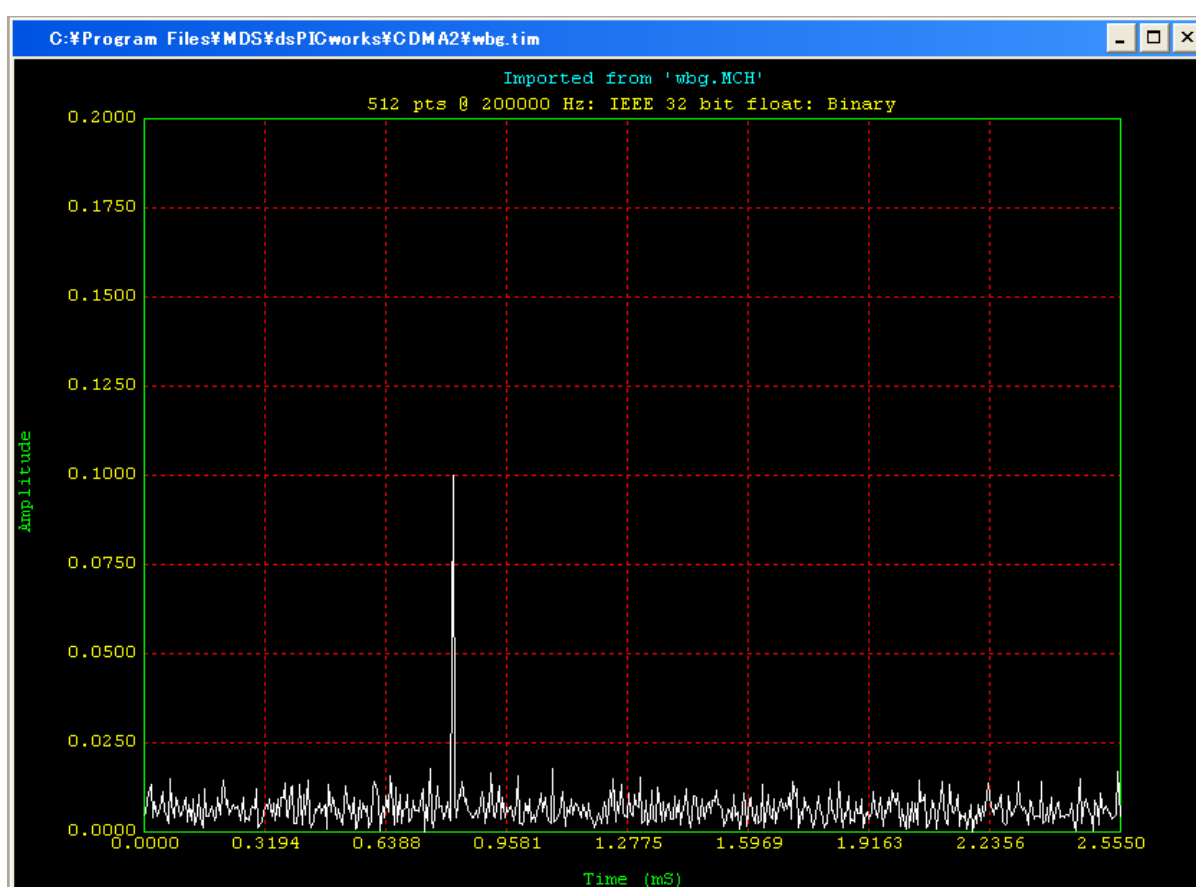


図7 C D M A受信信号振幅の約4倍の振幅でキャリア周波数のC W妨害を印加した場合のC D M A逆拡散解析結果

下の図 8 は、図 5 で示した実験において送信と受信のキャリア周波数に 300Hz の差を与えた場合の C D M A 受信信号の逆拡散解析結果である。C D M A 受信信号のレベルは図 5 の場合と同じであるがキャリア周波数に差がある結果、逆拡散インパルスピークレベル 0.11 に対してノイズレベル 0.03 となり、相関ピークが下がり S N 比も劣化している。これは、拡散コード周期 2.56ms 中に 1 サイクルに満たない位相回転が影響したためだと考えられる。

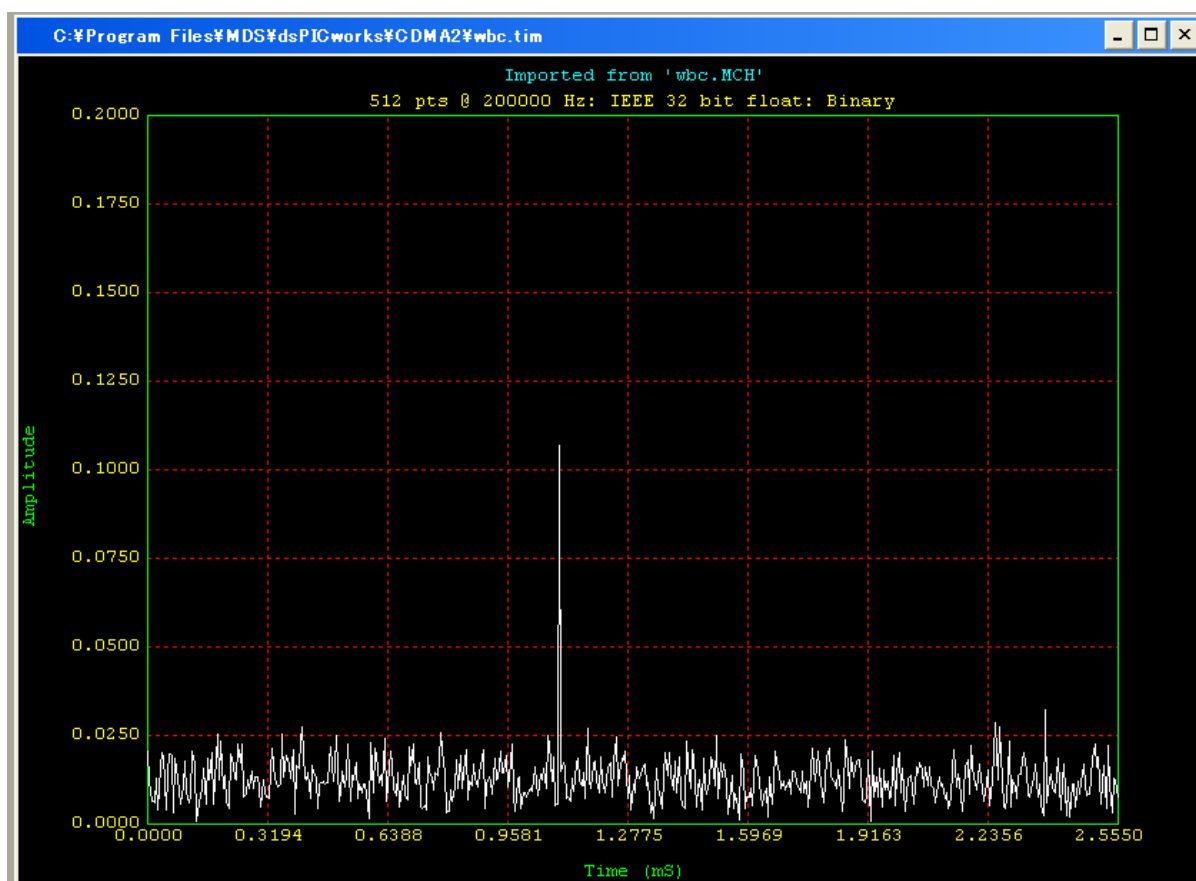


図 8 送受信キャリア周波数に 300Hz の周波数オフセットを与えた場合の C D M A 逆拡散解析結果

下の図9は、図5で示した実験において送信と受信のキャリア周波数に600Hzの差を与えた場合のCDMA受信信号の逆拡散解析結果である。CDMA受信信号のレベルは図5の場合と同じであるがキャリア周波数に差がある結果、逆拡散インパルスピークレベル0.063に対してノイズレベル0.03となり、相関ピークが更に下がりSN比も更に劣化している。これは、拡散コード周期2.56ms中に1サイクルを超える程度の位相回転が影響したためだと考えられる。

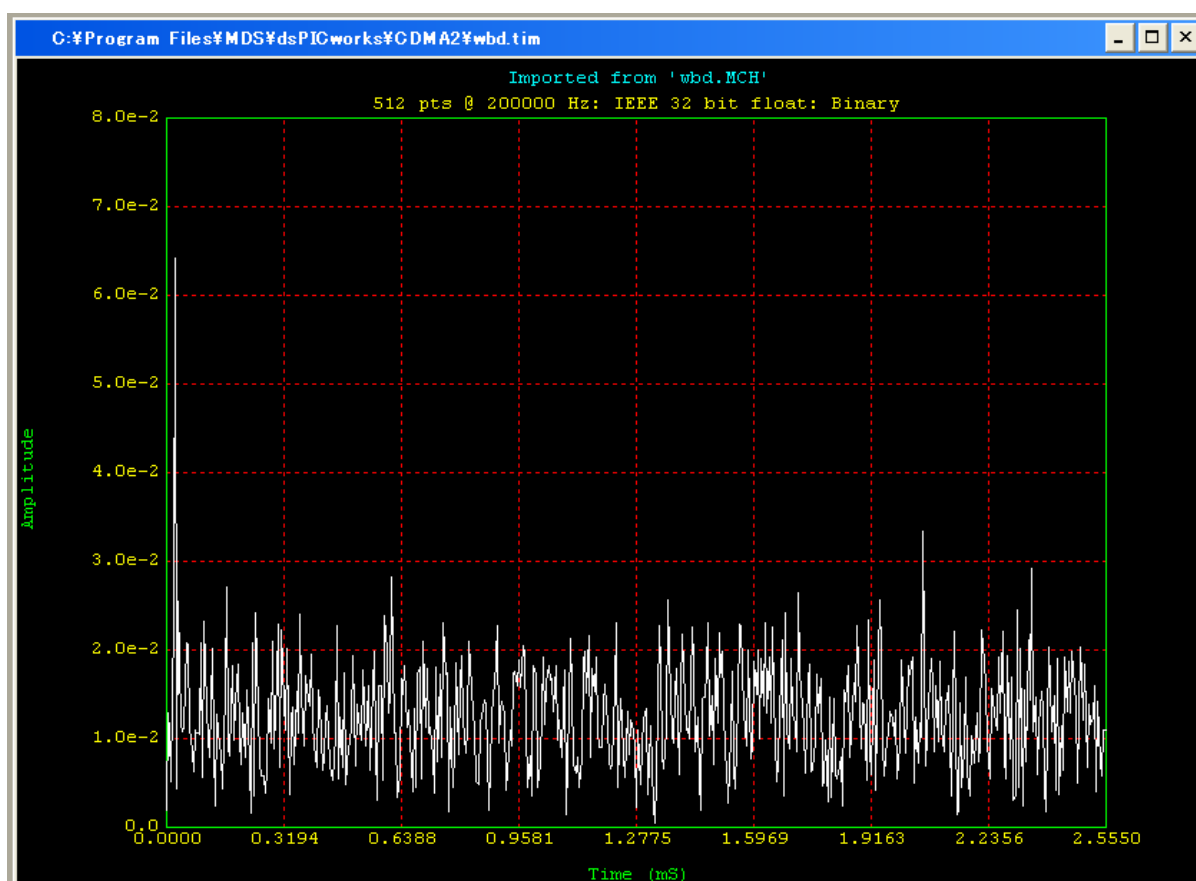


図9 送受信キャリア周波数に600Hzの周波数オフセットを与えた場合のCDMA逆拡散解析結果

ところで、拡散コード周期2.56ms中に1サイクルを超える位相回転が存在すればM系列周波数スペクトルの周波数軸における相関が無いとしてクロススペクトルの逆フーリエ変換結果はランダムノイズとなって相関ピークは存在しない筈である。しかし、実際の実験系では送信拡散コード周期と受信窓時間に僅かな差異が存在し、かつ、方形窓によるスペクトル広がりによって周波数のずれたスペクトル間での相関が評価されたものと考えられる。

下の図 10 は、図 5 で示した実験において送信と受信のキャリア周波数に 1kHz の差を与えた場合の C D M A 受信信号の逆拡散解析結果である。C D M A 受信信号のレベルは図 5 の場合と同じであるがキャリア周波数に差がある結果、明確な逆拡散インパルスピークは観測されなくなった。これは、拡散コード周期 2.56ms 中に数サイクル程度の位相回転が影響したためだと考えられる。

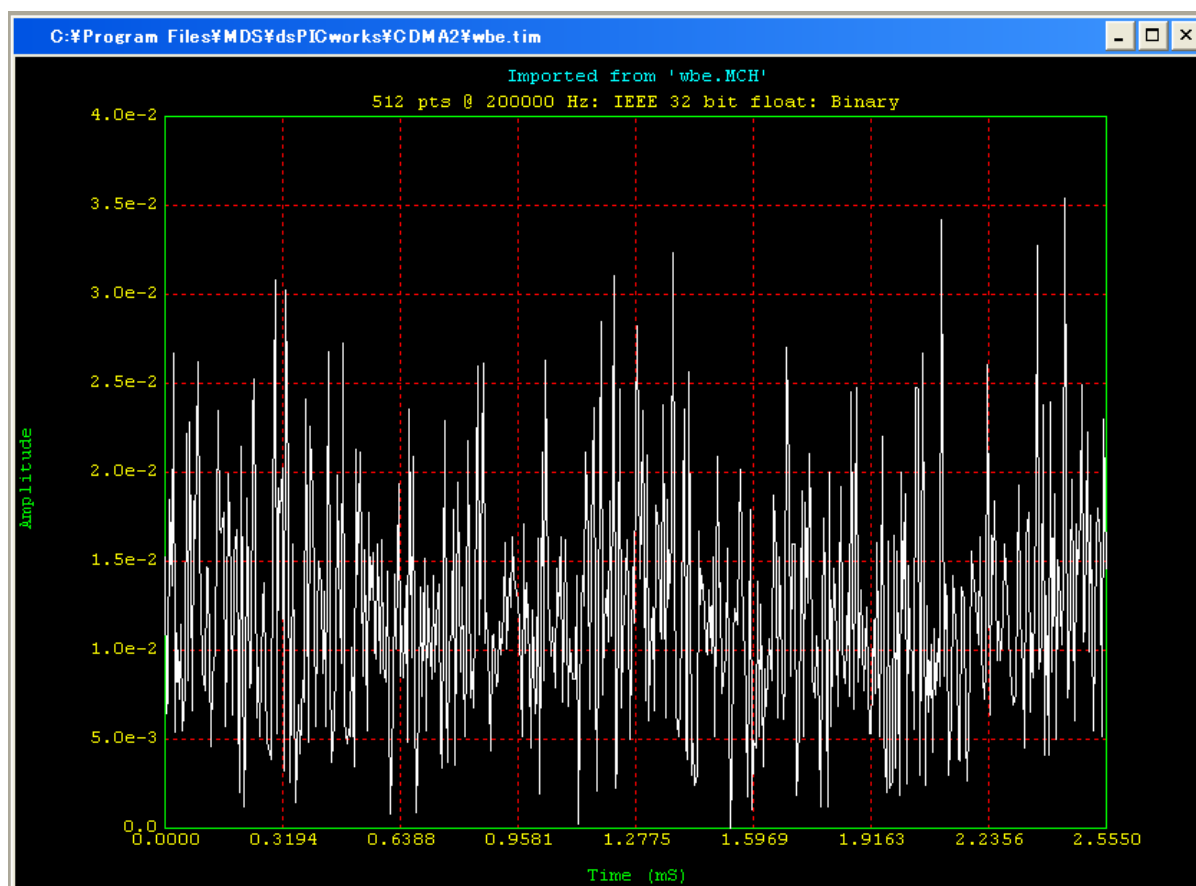


図 10 送受信キャリア周波数に 1kHz の周波数オフセットを与えた場合の C D M A 逆拡散解析結果

以上の実験結果から C D M A 方式の対妨害波耐性は十分に確認されたが、ドップラ周波数シフト等の影響を含む送信側と受信側でのキャリア周波数の差異が通信品質に大きく影響することが確認された。ここで、相関ピーク最大化による AFC 動作を考える場合に送信側の M 系列周期と受信側の窓時間幅を厳密に一致させてしまうと周波数引き込み幅が狭くなってしまいうデメリットがあり、少なくとも引き込み動作中は若干の差を持たせることが有効であると考えられる。これは、従来の時間軸でのコンボルバを用いた逆拡散方式でも同じである。

## 7. DSPのアセンブラ・プログラミング(クロススペクトル演算)

デモ基板を用いたCDMA受信信号の逆拡散処理でクロススペクトルなどの処理をc言語で記述したために100ms以上(512 point complex)掛かってしまったことを「5. CDMAの受信(スペクトル逆拡散)」で紹介した。今回は、スケーリングやクロススペクトル演算をアセンブラで記述し、CDMA受信信号の逆拡散処理を2.8msに短縮した。

下のc言語プログラムは、CDMA受信信号の逆拡散処理部分である。ここでは、ADCの取り込みデータを12 bit integer形式として実部 adbuf1[]及び虚部 adbuf2[]に取り込んで、アセンブラで記述した関数 MSMCpInt2Fract()を用いて fractcomplex 形式の FFT\_result2[]に代入する。このとき、ADCコード中心(2048)を0.0として、 $\pm 0.5$ に入るようにオフセットとスケーリング処理を行い、AGCのための最大 power を戻り値 fa1 とした。尚、ADCの取り込みデータを直接 Q.15 形式指定することも可能であるが、オフセットやスケーリング処理に便利であるために integer 形式を用いた。FFTComplexIP()及び BitReverseComplex()はライブラリで提供された関数である。VectorScale2()は、アセンブラで記述したAGCのためのスケーリング関数(FFT\_result2[] = FFT\_result2[] \* gs2)である。ここで、gs2 は unsigned int 形式とした。提供されたライブラリにもスケーリング関数はあったが Q.15 形式同士の積となるために値の減衰処理のみに限定されるものであった。CrossCplxV()は、アセンブラで記述したクロススペクトル関数であり、FFT\_result[]には送信 M 系列の複素スペクトルが入っていて FFT\_result2[] とのクロススペクトルを得て FFT\_result2[] に格納する。CrossCplxV()処理後の VectorScale2()によるスケーリングはクロススペクトル演算後 IFFTComplexIP()による演算桁落ちを防止するためである。尚、IFFTComplexIP()はライブラリで提供された関数である。

```
fa1=MSMCpInt2Fract(MaxSize,&FFT_result2[0],&adbuf1[0],&adbuf2[0]);
gs2=Fract2Float(fa1); gs2=1.0/sqrtf(gs2);
FFTComplexIP(Power2N,&FFT_result2[0],&FFT_factors[0],COEFFS_IN_DATA);
BitReverseComplex(Power2N,&FFT_result2[0]);
gsf=gs2;
VectorScale2(MaxSize*2,&FFT_result2[0],&FFT_result2[0],gsf);
```

```

CrossCplxV(MaxSize,&FFT_result2[0],&FFT_result[0],&FFT_result2[0]);
gsf=64;
VectorScale2(MaxSize*2,&FFT_result2[0],&FFT_result2[0],gsf);
IFFTComplexIP(Power2N,&FFT_result2[0],&FFT_factors[0],COEFFS_IN_DATA);
BitReverseComplex(Power2N,&FFT_result2[0]);

```

下のアセンブラ言語プログラムは、DSPを用いたクロススペクトル演算処理部である。ここでは、 $dstV[n] = srcV2[n] * srcV1[n]$  の演算処理を行っている。DSPでは、do 命令によって do 行から L0: 行までを w0+1 回くり返し処理する。また、mpy、mac、msc 命令などでは、それぞれ、積、積和、積差演算を行っているが同時に次に使う値のフェッチも 1 サイクル内で処理している。DSP の命令は多重化パイプライン処理に向いているが使用できるレジスタの組み合わせに制限が多いので「dsPIC30F\_dsPIC33F Programmer's Reference Manual\_DS70157B.pdf」を確認しながらプログラムを作成しなければならない。

```

; Prepare operation.
    mov        w2, w10                ; w10 is ptr of srcV1
    mov        w3, w11                ; w11 is ptr of srcV2
    mov        w1, w8                 ; w8 is ptr of dstV
    dec        w0, w0                 ; w0 = numElems-1
    clr        a,[w10]+=2,w4          ; a=0, w4 = Re(srcV1[0])
    clr        b,[w11]+=2,w7          ; b=0, w7 = Re(srcV2[0])
;.....
do      w0, L0
    mpy        w4*w7,a,[w11]+=2,w6    ; a = w4*w7, W6 = Im(srcV2[n])
    mpy        w4*w6,b,[w10]+=2,w5    ; b = w4*w6, W5 = Im(srcV1[n])
    mac        w5*w6,a,[w10]+=2,w4    ; a=a+w5*w6, w4= Re(srcV1[n+1])
    msc        w5*w7,b                ; b = b-w5*w7
    sac.r      a,#0,[w8++]             ; Re(dstV[n]) = accA
    sac.r      b,#0,[w8++]             ; Im(dstV[n]) = accB
L0:      mov        [w11++],w7        ; w7 = Re(srcV2[n+1])

```



## 8．ICD2故障修理

MPLAB ICD2が急に動かなくなりました。本体のPOWER表示は正常でERRORの表示無し。MPLAB IDEでデバッガやプログラマにICD2を接続しようとするとき”Unable to connect with MPLAB ICD2”のメッセージが表示されて全く使用できない。そこで、ICD2のフタを開けてチェックしたところPIC16F877の20MHzのXtalが発信していなかった。基板からC1を外して8pFに交換したところ正常動作に復帰した。

## 9．Explorer16によるリアルタイムFFTモニタ

前記「3．dsPIC33FでのFFT処理」および「4．デモ基板とPC間接続(RS-232)」でExplorer16を用いたFFT周波数スペクトル解析データのPC取り込み例を紹介した。ここでは、VBを使ってExplorer16で解析したFFT周波数スペクトルのリアルタイム表示例を示す。

Explorer16にdsPIC33Fを搭載してI・Q信号を12bitで取り込んでHanning窓をかけた時間波形を512ポイントの複素FFT処理した後に周波数(kHz)とレベル(dBm)を計算しながらスペクトル情報をRS232C経由でPCへ転送する。ここで、周波数は時間波形の取り込みサンプリング周期に依存するがExplorer16のSWによって切り替えられるようにした。PC上のVBでは、RS232Cで取り込んだデータをx軸周波数、y軸レベルに対応させてプロット描画している。

dsPIC33Fの処理速度に対して19kbps動作のRS232C転送速度が遅く約2秒に1回のフレーム更新スピードに制限されてしまった。下の図11はVBによる周波数スペクトルの表示例であり、25kHzの正弦波を入力している。この装置では、入力信号に対してアンチエイリアジング処理が行われていないために100kHzを超える周波数成分を持った信号が入力されると偽スペクトルが表示されてしまうことに注意しなければならない。

ベースバンドに変換したI・Q無線通信信号の周波数スペクトル評価では妨害電波の影響等dsPICworksでのバッチ処理では対応が難しい事象も多く、多少表示スピードは遅いが連続モニタが可能なこの方法は有益である。

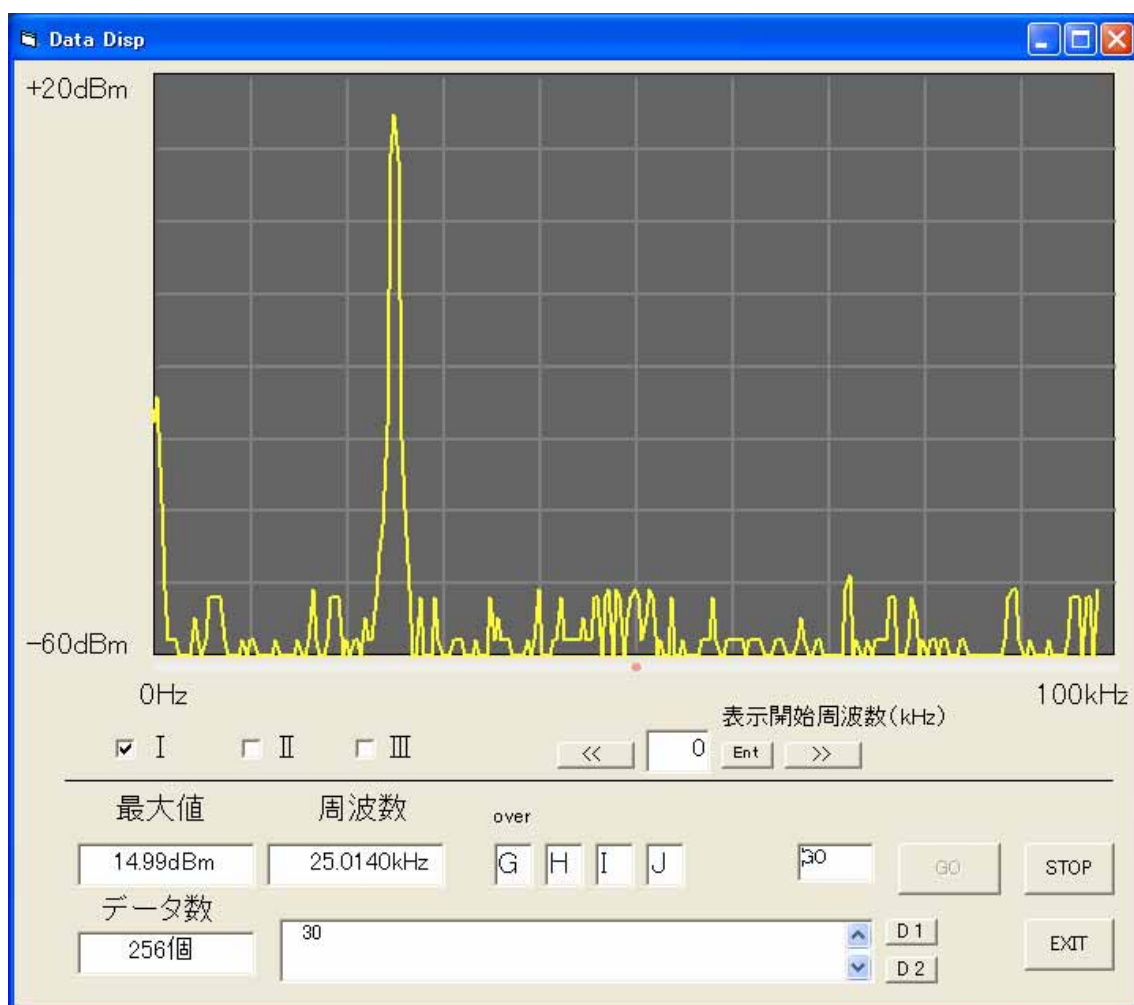


図 1 1 Explorer 16 による FFT 周波数スペクトルを V B でリアルタイム表示した例