

```

/** This is GPIB sample program to get Histogram for SIA-3000 ***/

#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>

#include <windows.h>
#include "decl-32.h"

#define BDINDEX                0                // Board Index
#define PRIMARY_ADDR_OF_SIA    5                // SIA-3000 GPIBADD=5
#define NO_SECONDARY_ADDR      0                // Secondary address of device
#define TIMEOUT                T10s            // Timeout value = 10 seconds
#define EOTMODE                1                // Enable the END message
#define EOSMODE                0                // Disable the EOS mode
#define ESB_BIT                (0x20)

#define BUFFERSIZE              1024            // Size of read buffer
#define Null                    '\0'
#define NG                      0

char    DataBuff[BUFFERSIZE + 1];
char    ErrorMnemonic[21][5] = {"EDVR", "ECIC", "ENOL", "EADR", "EARG",
                                "ESAC", "EABO", "ENEB", "EDMA", "",
                                "EOIP", "ECAP", "EFSO", "", "EBUS",
                                "ESTB", "ESRQ", "", "", "", "ETAB"};

void    GPIB_Cleanup (int Dev, char* ErrorMessage);
void    SendGPIB (int Dev, char* pString);
void    ReceiveGPIB (int Dev);

/** main start ***/

int     main()
{
    int     SIA, Fitting_OK;

    /** Initialize Device ***/
    SIA = ibdev (BDINDEX, PRIMARY_ADDR_OF_SIA, NO_SECONDARY_ADDR,
                TIMEOUT, EOTMODE, EOSMODE);
    if (ibsta & ERR) {
        printf ("Unable to open device\nibsta = 0x%x iberr = %d\n", ibsta, iberr);
        return 1;
    }

    // SIAの初期化を行います。
    SendGPIB (SIA, ":SYST:COMPOFF::SYST:HEADOFF;*ESE255;*SRE255");
    // :SYST:COMPOFF      DTSとのコンパチブル・モードをオフにします。
    // :SYST:HEADOFF     SIAからのデータにヘッダを付けないようにします。

    // Histogram測定の各パラメータを設定します。
    SendGPIB (SIA, ":HIST:DEFAULT");
    // 各パラメータの初期化を行います。GigaViewソフトウェアでHistogramツールを
    // 開いた時と同じ状態です。

    // 以下の設定は初期状態と同じですが、参考のため設定します。実際に変更の必要がない
    // 場合には測定時間のために実行する必要はありません。
    SendGPIB (SIA, ":HIST:PARAM:CHAN1");
    // 測定チャンネルを1に設定します。
    SendGPIB (SIA, ":HIST:PARAM:FUNC PER+");
    // 測定する対象をPeriod+ (立上がりから立上がりまでの周期) に設定します。
    SendGPIB (SIA, ":HIST:PARAM:SAMP 10000");
    // 測定回数を1万回に設定します。
    SendGPIB (SIA, ":HIST:PARAM:ARM:MODE STOP");
    // Arming ModeをArm On Stopモードに設定します。
    SendGPIB (SIA, ":HIST:PARAM:THRESHOLD 5050");
    // 閾値をAUTO(50%)に設定します。

    // DJ/RJ/TJを測定するためにTail FitをEnableにします。
    SendGPIB (SIA, ":HIST:TAIL:MODE ON");

    // ここから測定を開始します。
    do {
        SendGPIB (SIA, ":HIST:ACQ;*OPC");
        // 測定の実行
        SendGPIB (SIA, ":HIST:TAIL:COMP?");
        ReceiveGPIB (SIA);
    }
}

```

```

        Fitting_OK = atoi (DataBuff);
        // Tail Fitが完了したかの確認
    } while (Fitting_OK == NG);
    // Tail Fitが完了するまでループします。

    // 測定結果の取得
    SendGPIB (SIA, ":HIST:DJ?");
    ReceiveGPIB (SIA);
    printf ("DJ: %s\n", DataBuff);
    // DJの所得と表示
    SendGPIB (SIA, ":HIST:RJ?");
    ReceiveGPIB (SIA);
    printf ("RJ: %s\n", DataBuff);
    // RJの取得と表示
    SendGPIB (SIA, ":HIST:TJ?");
    ReceiveGPIB (SIA);
    printf ("TJ: %s\n", DataBuff);
    // TJの取得と表示

    printf("\n\nStrike ENTER to continue...");
    gets(DataBuff);

    ibonl (SIA, 0);

    return 0;
}

/** main end **/

/** Function Group **/
void GPIB_Cleanup (int Dev, char* ErrorMsg)
{
    printf ("Error : %s\nibsta = 0x%x iberr = %d (%s)\n",
        ErrorMsg, ibsta, iberr, ErrorMnemonic[iberr]);
    if (Dev != -1) {
        printf("Cleanup: Taking device offline\n");
        ibonl (Dev, 0);
    }
}

/** Send command to SIA **/
void SendGPIB (int Dev, char* pString)
{
    ibwrt (Dev, pString, strlen(pString));
    if (ibsta & ERR) {
        GPIB_Cleanup (Dev, "Unable to send command to SIA");
    }
}

/** Receive Data from SIA **/
void ReceiveGPIB (int Dev)
{
    ibrd (Dev, DataBuff, BUFFERSIZE);
    if (ibsta & ERR) {
        GPIB_Cleanup (Dev, "Unable to read data from SIA");
    }
    DataBuff[ibcntl-1] = Null;
}

```