KYOWA
ELECTRONIC INSTRUMENTS CO., LTD.

IM-A-731-C NOVEMBER 2008

SENSOR INTERFACE

PCD SERIES

CONTROL FUNCTION

INSTRUCTION MANUAL

Thank you for purchasing KYOWA's product PCD SERIES SENSOR INTERFACE.
Read this Instruction Manual carefully in order to make full use of the high performance capabilities of the product.
Do not use the product in methods other than described in this Manual.
This Instruction Manual is related to method for controlling the PCD Series.
For details of Hardware and Software, customers are required to refer to the respective Instruction Manuals.

CONTENTS

1. INTRODUCTION

This Instruction Manual is related to necessary control functions for controlling the PCD-300A/320A(hereinafter referred as the PCD-300A Series) and PCD-300B/PCD-300B-F/PCD-330B-F(hereinafter as the PCD-300B Series) Sensor Interface (hereinafter referred to as the PCD) when users are creating their own control software.
Since the PCD is used by communicating with the PC via USB interface, it is essential that the control software should be created with a program operated by Windows 2000/XP/Vista system. For creating the Windows program, refer to the textbook of programming languages in use. All the control functions are included in PCD specified Dynamic Link Library (hereinafter referred to as the DLL). Since no descriptions on methods for using and calling the DLL are written in this Instruction Manual, refer to textbooks of the required programming languages.

2. OPERATIONAL ENVIRONMENT

- OS Windows 2000 (Professional Edition only)
  Windows XP (Home Edition and Professional Edition)
  Windows Vista
- Memory 512 Mbyte or more (1 Gbyte or more is recommended)

3. NECESSARY FILES

1) PCD300B.SYS USB driver file for the PCD
   When the PCD is connected to the PC for the first time, it is required to install the USB driver to the Windows system.
   For installing the USB driver, refer to the PCD Series Instruction Manual for USB Driver Installation.
   Note that this driver would not correctly operate if it is used from multiple application programs at the same time. Always operate the PCD controlling application solely by itself.
2) PCD300B.DLL A file sending and receiving control commands to the PCD.
3) PCD.DLL A file including PCD control functions.
   All the functions called from the application program are included in this file.
   Note that the PCD300B.DLL and PCD.DLL should be installed in the same directory together with the application program. If these files are short, the application program would not correctly operate.

4. CONTROL FUNCTION

The control function is roughly classified into 2 groups; basic function and individual function.
Features of the respective functions are described in the following.

Basic function
- Capable of controlling the PCD to some extent with few control commands.
- Even if multiple PCD units are connected, there is scarcely no change in the control method.
  When multiple PCD units are connected, it is not available to separately control each of the individual PCD units.
- The number of monitor data loaded at one time is limited to 6000 data/CH.

Individual function
- Since control commands can be separately sent to and received from each of the PCD units, detailed controls appropriate for the use are available.
- When multiple PCD units are connected, it is required to create a program by considering the order of sending and receiving the control command.
- There is no limit when receiving the monitor data as the basic function.

4-1 LIST OF FUNCTIONS

4-1-1 Precautions on Functions

With the release of newly developed PCD-300B Series, simple precautions are described in the following.

- The newly developed functions are basically described with under lines to be differentiated from the existing functions.
- Among newly developed functions, those especially related to the PCD-300B Series are as follows.

  Basic Functions
  - Pcd_SetDataType — Set A/D data format
  - Pcd_GetMoniData — Obtain monitor data (Considering data assignment of short and long types)

  Individual Functions
  - Pcd_MOD — Set Meas mode.
  - Pcd_TED — Start loading TEDS information
  - Pcd_TEJ — Obtain TEDS information loaded results.
  - PcdePRR_Float — Load parameters (Return value: float type)
- Existing control functions added with specific functions of PCD-300B Series are as follows.

  Basic Functions
  - PcdSetChParam — Set CH condition
  - PceSetChParamEx — Set CH condition
  - PcdSetSamp — Set sampling frequency
  - PcdGetMoniData — Obtain physical value monitor data
  - PcdeGetMeasData — Obtain monitor data
  - PcdGetType — Load PCD type

  Individual Functions
  - dPcdGetData — Obtaining monitor data
  - PcdRNG — Set range
  - PcdSPD — Set sampling frequency
  - PcdPRR — Load parameters
  - PcdDMS — Set A/D data format

PCD control functions are listed in the following table.

4-1-2 Basic Functions

No.	Function Name	Function	Section
1	PcdOpen	Initialize PCD.DLL.	4-3-1
2	PcdClose	Close PCD.DLL.	4-3-2
3	PcdSetOsc	Set oscillator.	4-3-3
4	PcdSetSync	Set synchronous clock.	4-3-4
5	PcdSetSamp	Set sampling frequency.	4-3-5
6	PcdSetChParam	Set channel condition. (For physical value data)	4-3-6
7	PcdSetChParamEx	Set channel condition. (For physical value data)	4-3-7
8	PcdMoniStart	Start monitor. (For physical value data)	4-3-8
9	PcdMoniStop	Stop monitor. (For physical value data)	4-3-9
10	PcdGetMoniData	Obtain monitor data. (Physical value data)	4-3-10
11	PcdStartAd	Start monitor. (For A/D data)	4-3-11
12	PcdStopAd	Stop monitor. (For A/D data)	4-3-12
13	PcdGetMeasData	Obtain monitor data. (For A/D data)	4-3-13
14	PcdBalance	Conduct balance.	4-3-14
15	PcdTest	Conduct self-test.	4-3-15
16	PcdConnect	Load connected state.	4-3-16
17	PcdReadStatus	Read status.	4-3-17
18	PcdGetType	Read PCD type.	4-3-18
19	Pcd_SetStorageDataType	Set A/D data format of the PCD.	4-3-19
20	Pcd_SetStorageMode	Set storage mode.	4-3-20
21	Pcd_GetStorageDataFloat	Obtain storage data. (Physical value converted value)	4-3-21
22	Pcd_CheckStorageData	Check the number of storage data.	4-3-22
23	Pcd_ClearStorageCount	Initialize the number of storage data.	4-3-23
24	Pcd_GetMoniData	Obtain monitor data (For A/D data: PCD-300B corresponding version)	4-3-24

4-1-3 Individual Functions

No.	Function Name	Function	Section
1	dPcdGetDevices	Check connected state.	4-4-1
2	dPcdStart	Start monitor.	4-4-2
3	dPcdStop	Stop monitor.	4-4-3
4	dPcdGet Data	Obtain monitor data.	4-4-4
5	dPcdGetStatus	Obtain status.	4-4-5
6	PcdSPD	Set sampling frequency.	4-4-6
7	PcdOMS	Set oscillator.	4-4-7
8	PcdCMS	Set synchronous clock.	4-4-8
9	PcdRNG	Set range.	4-4-9
10	PcdFLT	Set low pass filter.	4-4-10
11	PcdCUP	Set high pass filter.	4-4-11
12	PcdTES	Conduct self-test.	4-4-12
13	PcdTER	Load self-test results.	4-4-13
14	PcdBAL	Conduct balance.	4-4-14
15	PcdBAR	Load balance results.	4-4-15
16	PcdSTA	Start A/D conversion.	4-4-16
17	PcdSTP	Stop A/D conversion.	4-4-17
18	PcdPRR	Load parameter.	4-4-18
19	PcdERR	Load error.	4-4-19
20	PcdECL	Clear error.	4-4-20
21	PcdCHK	Conduct hardware check.	4-4-21
22	PcdCHR	Load hardware check results	4-4-22
23	PcdDMS	Set data format.	4-4-23
24	Pcd_MOD	Set Mead mode.	4-4-24
25	Pcd_TED	Start loading TEDS information.	4-4-25
26	Pcd_TEJ	Obtain TEDS information loaded results.	4-4-26
27	PcdPRR_Float	Load parameters	4-4-27

4-2 SPECIFICATIONS COMMON TO VARIOUS FUNCTIONS

4-2-1 Return Value of Function

1) Unless no descriptions are made, return values of various functions are set as follows.

0:	Ends normally
-1:	Command or parameter error
-2:	No PCD is connected.
-3:	No PCD300B.DLL exists.
-4:	Windows system error.
-5:	Communication error between the PCD
-6:	Receive buffer over flow
-7:	Setting not able because during A/D
-8:	PcdOpen is not conducted.
-9:	Execution error because A/D is stopped

2) Return values of part of individual functions (starting with small letter ‘d’) are set as follows.

0x00000000	Ends normally
0xF0000001	Parameter error
0xF0000002	No PCD is connected
0xF0000003	Not responding from PCD
0xF0000004	Communication error between the PCD
0xF0000005	Receive buffer overflow
0xF0000006	Windows system error
0xF0000007	Windows system error
0xF0000008	Monitor data receive size error

4-2-2 PcdNo

Set PcdNo. of argument of various functions to a value subtracting 1 from the PCD No.

Ex.) PCD No.	PcdNo.
1:	0
2:	1
3:	2
4:	3

4-3 DETAILS OF BASIC FUNCTION

4-3-1 PcdOpen

- Function: Initializes the PCD.DLL.
  Firstly, before using the PCD.DLL, it is required to call this function.
  Or, no other functions are called.
- Definition: __declspec(dllimport) long __stdcall PcdOpen(char *DllPath)
- Argument: *DllPath Pathname of PCD.DLL
  The size of pathname assignment should be 260 bytes or more.
  If you are calling from the C language, please argument to NULL.
- Others: Regarding the application program, when either basic or individual functions are used, it is required to call this function PcdOpen for starting up and PcdClose, for closing.

4-3-2 PcdClose

- Function: Closes the application program.
  Call this function for closing the application program.
- Definition: long_stdcall PcdClose(void)
- Argument: None
- Others: If the application program is closed without calling this PcdClose, Windows system hereafter may become unstable.

4-3-3 PcdSetOsc

- Function: Sets master/slave oscillator.
- Definition: __declspec(dllimport) long __stdcall PcdSetOsc (long PcdNo, long Osc)
- Argument:
  PcdNo Set PCD No. (0 to 3) for setting the oscillator.
  Osc Oscillator
  0: Slave
  1: Master
- Others:
  1) Set the oscillator while stopping monitoring. No oscillator is set during monitoring.
     When multiple PCD-300A units or PCD-300B Series are connected, always set one PCD as a master unit and the rest of the PCD units as slave units. However, by setting only the master PCD, settings of the PCD slaves can be omitted. In addition, it is essential that the oscillator is correctly connected as well as synchronous cables are properly wired.
  2) When the PCD power is turned ON, although the PCD is set as a master unit, it is required to set the oscillator once again before starting monitoring.
  3) Setting of the oscillator is enabled only for the PCD-300A and PCD-300B Series.
  4) When only the PCD-300A or PCD-300B Series is connected, setting of the synchronous clock (PcdSetSync) can be omitted. At this time, the PCD that is set to the master of the oscillator shall be a master of the synchronous clock.
  5) When setting the synchronous clock, always set the oscillator before setting the synchronous clock.
  6) When the PCD-300B Series and PCD-300A Series are mixedly connected, always place the PCD-300B Series ahead of the other PCD units and always set the PCD-300B Series at the head as the master unit.
  7) When PCD-300B Series and PCD-300A Series are mixedly connected, controlling the PCD units are not available if connected by the following order.
     ×: PCD-300A Series → PCD-300B Series
     ×: PCD-300B Series → PCD-300A Series → PCD-300B Series
  8) This function is set only as internal variable of the PCD.DLL and not sending the actual command.

4-3-4 PdSetSync

- Function　Sets synchronous clock.
- Definition　__declspec(dllimport) long __stdcall PcdSetSync (long PcdNo. long Sync)
- Argument　PcdNo　Set PCD No. (0 to 3) for setting the synchronous clock.
  Sync　Synchronous clock
  0:　Slave
  1:　Master
- Others
  1) Set the synchronous clock while stopping monitoring. No synchronous clock is set during monitoring.
     When multiple PCD units are connected, always set one PCD as a master unit and the rest of the PCD units as slave units. However, by setting only the master PCD, settings of the PCD slaves can be omitted. In addition, it is essential that the synchronous clock is correctly connected as well as synchronous cables are properly wired.
  2) When the PCD power is turned ON, although the PCD is set to a master unit, it is required to set the synchronous clock once again before starting monitoring.
  3) Set the oscillator before setting the synchronous clock.
  4) When PCD-300B Series and PCD-300A Series are mixedly connected, always place the PCD-300B Series ahead of the other PCD units and always set the PCD-300B Series at the head as the master PCD unit.
  5) This function is set only as internal variable of the PCD.DLL and not sending the actual command.

4-3-5 PdSetSamp

- Function　Sets sampling frequency.
- Definition　__declspec(dllimport) long __stdcall PcdSet Samp(long Samp)
- Argument　Samp　Sampling frequency

Samp	Sampling frequency
1	1 Hz
2	2 Hz
5	5 Hz
10	10 Hz
20	20 Hz
50	50 Hz
100	100 Hz
200	200 Hz
500	500 Hz
1000	1 kHz
2000	2 kHz
5000	5 kHz
10000	10 kHz (Only for PCD-300BSeries)

- Others
  1) Set sampling frequency while stopping monitoring. No sampling frequency is set during monitoring.
  2) When one PCD or multiple PCD-300B Series units are used and when controlling the PCD units on Windows 2000, up to 5 kHz sampling frequency is available.
  3) This function is only set as the internal variable of the PCD.DLL and not sending the actual command.

4-3-6 PcdSetChParam

- Function: Sets measuring conditions of channels of the PCD-300A and PCD-300B, PCD-300B-T, PCD-300B-H, PCD-300B-HT.
- Definition: __declspec(dllimport) long __stdcall PcdSetChParam (long Ch ,long Range, float Coeff, float, Offset)
- Argument

  Ch: Channel No. (1 to 16)
  Set PCD No. to PCD ID×4.
  Ex.) When PCD ID is 1: Set channel Nos. from 1 to 4.
  When PCD ID is 2: Set channel Nos. from 5 to 8.

  Range: Measuring range
  0: OFF
  1: 200 μm/m
  2: 500 μm/m
  3: 1000 μm/m
  4: 2000 μm/m
  5: 5000 μm/m
  6: 10000 μm/m
  7: 20000 μm/m (Only for PCD-300B,PCD-300B-T, PCD-300B-H,PCD-300B-HT)

  Coeff: CAL coefficient
  $1.0e^{-8}$~$1.0e^{8}$
  When using PcdGetMeasData for obtaining monitor data with A/D data, set '1.0.'

  Offset: Offset
  ±$1.0e^{-8}$ to $1.0e^{8}$
  When using PcdGetMeasData for obtaining monitor data with A/D data, set '0.0.'

- Others
  1) Set channel condition while stopping monitoring. No channel condition is set during monitoring.
  2) Not error if channel conditions of not connected PCD-300A and PCD-300B are set.
  3) If this function is used for the PCD-320A, values of low pass filter and high pass filter are set to OFF and 0.2 Hz, respectively.
  4) This function P is set only as the internal variable of the PCD.DLL and not sending the actual command.

4-3-7 PcdSetChParamEx

- Function　Sets measuring conditions of channels.
- Definition　__declspec(dllimport) long __stdcall PcdSetChParam (long Ch ,long Range, long Filter, long Coupling, float Coef, float Offset)
- Argument

Ch　Channel No. (1 to 16)
Set PCD Nos. to PCD ID×4.
Ex.) When PCD ID is 1: Set channel No. from 1 to 4.
When PCD ID is 2: Set channel No. from 5 to 8.

Range　Measuring range

	PCD-300A PCD-300B PCD-330B Sries(Strain)	PCD-320A PCD-330B Series(Voltage)
0:	OFF	OFF
1:	200 μm/m	1 V
2:	500 μm/m	2 V
3:	1000 μm/m	5 V
4:	2000 μm/m	10 V
5:	5000 μm/m	20 V
6:	10000 μm/m	50 V
7:	20000 μm/m (PCD-300B Series)	

Filter　Low pass filter (For PCD-320A/PCD-300B-F/PCD-330B Series)
For PCD-300A and PCD-300B, set '0.'

	PCD-320A PCD-330B Series(Voltage)	PCD-300B-F PCD-330B Series(Strain)
0:	FLAT	FLAT
1:	10 Hz	10 Hz
2:	30 Hz	30 Hz
3:	100 Hz	100 Hz
4:	300 Hz	

Coupling　High pass filter (For PCD-320A,PCD-330B Series)
For PCD-300A and PCD-300B, set '0.'

0:	OFF
1:	0.2 Hz

Coeff　CAL coefficient
$1.0e^{-8}$ to $1.0e^{8}$
When using PcdGetMeasData for obtaining monitor data with A/D data, set '1.0.'

Offset　Offset
$\pm 1.0e^{-8} \sim 1.0e^{8}$
When using PcdGetMeasData for obtaining A/D value of monitor data, set '1.0.'

- Others
  1) Set channel condition while stopping monitoring. No channel condition is set during monitoring.
  2) Not error if channel conditions of not connected PCD-300A and PCD-300B are set.

3) This function is set only as the internal variable of the PCD.DLL and not sending the actual command.

4-3-8 PcdMoniStart

- Function　　Starts monitoring. (To obtain physical value data)
- Definition　　__declspec(dllimport) long __stdcall PcdMoniStart (void)
- Argument　　None
- Others　　1) This function is used together with the following functions for obtaining physical value monitor data.
  2) This function is calling various functions such as dPcdStart, PcdOMS, PcdCMS, PcdRNG, PcdFLT, PcdCUP, and PcdSTA inside this function.

4-3-9 PcdMoniStop

- Function　　Stops monitoring. (To obtain physical value data)
- Definition　　__declspec(dllimport) long __stdcall PcdMoniStop (void)
- Argument　　None
- Others　　1) This function is used together with the following functions for obtaining physical value monitor data.
  PcdMoniStart and PcdGetMoniData
  2) This function is calling functions dPcdStop and PcdSTP inside this function.

4-3-10 PcdGetMoniData

• Function　Obtains physical value monitor data.
For application program, call this function within approximately 0.3 sec to obtain the monitor data.

• Definition　__declspec(dllimport) long __stdcall PcdGetMoniData (float *MeasData)

• Argument　*MeasData　Monitor data converted into physical value data from the preset range, CAL coefficient, and offset.
Channel assignment is described in the following.

Ex.1) When only one PCD is connected.
CH01, CH2, CH3, CH4, CH1, CH2, CH3, CH4…..
Ex.2) When two PCD units are connected
CH1, CH2, CH3, CH4, CH5, CH6, CH7, CH8, CH1, CH2, CH3…..

The number of channel data obtained at one time is maximum 6000 data.
The size of float type MeasData alignment to be set on the application side is required to have the 'number of connected PCD units × 4CH × 6000.'

Ex.) When only one PCD is connected.
1 × 4 × 6000 = 24000
Therefore, it is required to define the data assignment to the following.
float MeasData [24000]
If the size of data assignment is less than the above, the program would not correctly operate and error occurs.

• Return Value　Return value of this function slightly differs from those of the other functions.
0 or less: Return value of error. Contents are same as other functions.
-9: Returns '-9' when ring buffer becomes full.
0: Returns '0' when no monitor data exists.
1 or more: The number of data obtained per channel.

• Others
1) This function is used together with the following functions for obtaining physical value monitor data.
PcdMoniStart and PcdfMoniStop
2) This function PcdGetMoniData can be used only during monitoring that is started by PcdMoniStart.
3) When multiple PCD units are connected, this function is obtaining monitor data of all the connected PCD units. Therefore, if there is any PCD not to be measured, turn OFF the power or disconnect the USB cable.
4) No external input information and status of data are obtained from monitor data that is obtained by this function. If it is required to obtain the above items, use functions PcdStartAd, PcdGetMeasData, and PcdStopAd and obtain the monitor data.
5) This function is calling dPcdGetData inside this function.

4-3-11 PcdStartAd

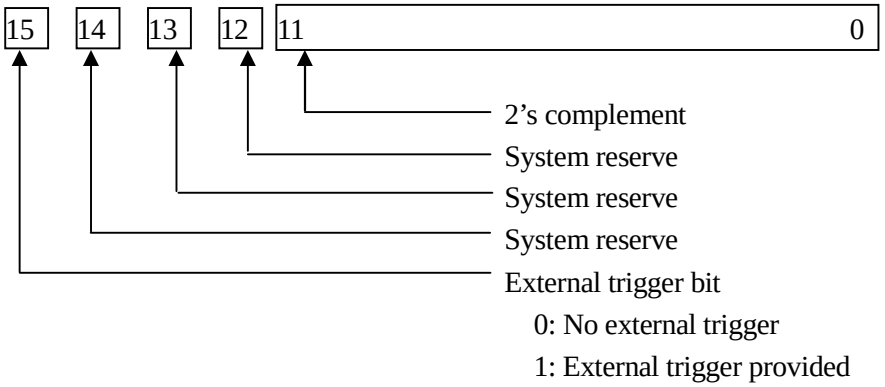
- Function　Starts monitoring. (To obtain A/D data)
- Definition　__declspec(dllimport) long __stdcall PcdStartAd (long PcdNo.)
- Argument　PcdNo.　Set PCD ID No. for starting monitoring.
  Unlike other functions, this function sets the PCD ID No.
  0:　Start monitoring of all the connected PCD units.
  1 to 4:　Start monitoring of only 1 preset PCD.
  If multiple PCD units are connected, the number of channels obtained with function PcdGetMeasData is 4 channels of one PCD.
- Others　1) This function is used together with the following functions for obtaining A/D value monitor data.
  PcdGetMeasData and PcdStopAd
  2) This function is calling various functions such as dPcdStart, PcdSPD, PcdOMS, PcdCMS, PcdRNG, and PcdSTA inside this function.

4-3-12 PcdStopAd

- Function　Stops monitoring (For obtaining A/D data).
- Definition　__declspec(dllimport) long __stdcall PcdStopAd (void)
- Argument　None
- Others　1) This function is used together with the following functions for obtaining A/D value monitor data.
  PcdStartAd and PcdGetMeasData
  2) This function is calling dPcdStop and PcdSTP inside this function.

4-3-13 PcdGetMeasData

- Function　Obtains A/D value monitor data.
  For application program, call this function within an interval of approximately 0.3 sec to obtain the desired monitor data.
- Definition　__declspec(dllimport) long __stdcall PcdGetMeasData (short *MeasData)
- Argument　*MeasData　A/D data outputted from the PCD-300A/PCD-320A.
  Channel assignment is described in the following.
  Ex. 1) When only one PCD is connected
  CH01, CH2, CH3, CH4, CH1, CH2, CH3, CH4·····
  Ex. 2) When two PCD units are connected
  CH1,CH2,CH3,CH4,CH5, CH6, CH7, CH8, CH1, CH2, CH3···
  The number of channel data obtained at one time is maximum 6000 data.
  The number of channel assignments of short type MeasData to be set on application side is required to have the 'number of connected PCD units × 4CHs × 6000.'
  Ex.) When only one PCD is connected
  1 × 4 × 6000 = 24000
  Therefore, it is required to define the number of channel assignments to the following.
  short MeasData [24000]
  If the number of channel assignments is less than the above, the program would not correctly operate and error occurs.
- Return Value　Return value of this function differs from those of other functions.
  0 or less:　Return value when error occurred. Contents are same as other functions.
  0:　Returns '0' when no monitor data exists.
  1 or more:　The number of data obtained per channel.
- Others
  1) This function is used together with the following functions for obtaining A/D monitor data.
  PcdStartAd and PcdStopAd
  2) This function can be used only while monitoring that is started by PcdStartAd.
  3) Monitor data obtained from the PCD is in the following 1-data/2-byte structure.



4) How to convert data into physical value data

Full scale A/D value of the PCD-300A Series is ±2000 digits.

Therefore, physical value can be obtained by calculating range, CAL coefficient, and offset in the following expression.

Physical value = Range/Full scale A/D value (2000)×Meas data×CAL coefficient+Offset

Ex.) Suppose range: 200μm/m, Meas data: 1234, CAL coefficient: 2.0, and offset: 1.5

200/2000 × 1234 × 2.0 + 1.5 = 248.3

5) This function is calling dPcdGetData inside the function.

6) This function is corresponding to the PCD-300B later that the following data formats are available. 12-bit A/D conversion data and Code extension 12-bit A/D conversion data, and 16-bit A/D conversion data.

4-3-14 PcdBalance

- Function　Conducts balance.
- Definition　__declspec(dllimport) long __stdcall PcdBalance (long *BalCh,
  long TotalBalCh, long *BalResult)
- Argument
  - *BalCh　Set balance channel assignment.
    - 0: No balance is conducted.
      Or no PCD-300A or PCD-300BSeries is connected.
    - 1: Conduct balance.
  - TotalBalCh　Set the number of preset balance conducting channels.
  - *BalResult　Balance results of the preset BalCh.
    - -1: No balance is conducted.
    - 0: Balance OK
    - 1: Resistance balance over
    - 2: Range OFF
    - 3: Capacity balance over
    - 4: Connection failure
- Others
  1) Conduct balance while stopping monitoring. No balance is conducted during monitoring.
  2) PcdBalance would not return the processing to the application program unless balance is completed or until error occurred.
     Take care not to have the application program become time out.
  3) Example of use 1) When only PCD ID = 1 is connected and balance is conducted for only CHs 2 and 3.

```
long Balch[4]:          ← Maximum PCD ID × 4 are required.
long TotalBalCh:
long BalResult [4]:     ← Same size as BalCh

BalCh[0] = 0; BalCh[1] = 1; BalChj[2] = 1; BalCh[3] = 0;
TotalBalCh = 4;          ← Set maximum PCD ID × 4
PcdBalance(&BalCH[0], TotalBalCh, &BalResult[0]);
```

  4) Example of use 2) When PCD ID = 2 and 4 are connected and balance is conducted only for CHs 5 and 16

```
long Balch[16]:         ← Maximum PCD ID × 4 are required.
long TotalBalCh:
long BalResult [16]:    ← Same size as BalCh

BalCh[0]  = 0; BalCh[1]  = 0; BalCh[2]  = 0, BalCh[3]  = 0;
BalCh[4]  = 1; BalCh[5]  = 0; BalCh[6]  = 0, BalCh[7]  = 0;
BalCh[8]  = 0; BalCh[9]  = 0; BalCh[10] = 0, BalCh[11] = 0;
BalCh[12] = 0; BalCh[13] = 0; BalCh[14] = 0, BalCh[15] = 1;
TotalBalCh = 16
PcdBalance(&BalCh[0], &TotalBalCh, &BalResult[0]);
```

  5) Before conducting balance, set PcdSetOsc and PcdSetChParam.
  6) This function is calling various functions such as PcdSetOMS, PcdRNG, PcdBAL, and PcdBAR inside the function.

4-3-15 PcdTest

- Function　Conducts self-test.
- Definition　__declspec(dllimport) long __stdcall PcdTest (long *TesNo, long TotalTesNo, long *TesResult)
- Argument

*TesNo　Set whether or not self-test is conducted for every PCD No.
0:　No self-test is conducted
Or no PCD is connected.
1:　Conduct self-test.

TotalTesNo　Set the number of preset PCD units.

*TesResult　Results of the self-test conducted for every channel.
-1:　No self-test is conducted or no PCD is connected.
0:　Normally completed.
1 or more　Error occurred.
Since the self-test results are outputted for every channel, the number of the self-test results assignments is required to be TotalTesNo × 4.

- Others

1) Conduct self-test while stopping monitoring. No self-test is conducted during monitoring.
2) The PcdTest would not return the processing to the application unless the self-test is completed or error occurs.
Take care not to have the application program become timeout.
3) Example of use 1) When only PCD ID = 1 is connected.

```
long TestNo[1]:          ← Maximum PCD ID
long TotalTestNo;
long TesResult[4];       ← TotalTesNo. × 4 are required

TestNo[0] = 1
TotalTestNo = 1          ← Set maximum PCD ID.
PcdTest (&TestNo.[0], TotalTesNo, &TesResult[0]);
```

4) Example of use 2) Although PCD ID = 2 and 4 are connected, self-test is conducted only for PCD ID4.

```
long TesNo.[4];          ← Maximum PCD ID
long TotalTestNo;
long TesResult[16];      ← Total TestNo × 4 are required

TesNo[0] = 0; TesNo[1] = 0; TesNo[2] = 0, TesNo[3] = 1; TotalTesNo = 4;
PcdTest(&TesNo[0], TotalTesNo, &TesResult[0]);
```

5) Before conducting self-test, set PcdSetOsc.
6) This function is calling functions PcdOms, PcdTES, and PcdTER inside the function.

4-3-16 PcdConnect

- Function　Checks connected state.
- Definition　__declspec(dllimport) long __stdcall PcdConnect (long PcdNo, long *Connect)
- Argument
  - PcdNo　Set PCD No. (0 to 3) for checking the connected state.
  - *Connect　PCD connected state
    - 0:　Not connected
    - 1:　Connected
- Others
  1) Check the connected state while stopping monitoring. If connection check is conducted during monitoring, correct connected state may not be obtained.
  2) This function is calling dPcdGetDevice inside this function.

4-3-17 PcdReadStatus

- Function　Obtains the PCD status.
- Definition　__declspec(dllimport) long __stdcall PcdReadStatus (long PcdNo, long *Status)
- Argument
  - PcdNo　Set PCD No. (0 to 3) for obtaining the PCD status.
  - *Status　PCD status.
    - 0:　Stopping A/D conversion
    - Other than 0:　Not defined
- Others
  1) Although the PCD status can be obtained either while stopping monitoring or during monitoring, if it is frequently obtained during monitoring, some CPU clock may not be in time for obtaining the monitor data and error occurs.
  2) This function is calling dPcdGetStatus inside the function.

4-3-18 PcdGetType

- Function　Obtains the PCD type.
- Definition　__declspec(dllimport) long __stdcall PcdGetType (long PcdNo, long *PcdType)
- Argument　PcdNo　Set.PCD No. (0 to 3) for obtaining the PCD type.
  *PcdType

-1:	Not connected
1:	PCD-300A
2:	PCD-320A
3:	PCD-300B
4:	PCD-300B-F
5	System reserve
6	System reserve
7	System reserve
8	System reserve
9	System reserve
10	System reserve
11	PCD-330B-F
12	System reserve
13	System reserve
14	System reserve

- Others　1) Obtain the PCD type while stopping monitoring.
  2) This function is calling PcdPRR inside the function.

4-3-19 Pcd_SetDataType

- Function　Sets A/D data format of PCD-300B/PCD-300B-F.
- Definition　__declspec(dllimport) long __stdcall Pcd_SetDataType (long lDataType)
- Argument　lDataType　Set A/D data format.
  - 0: 12-bit A/D conversion (2000 digits)
  - 1: Code extension 12-bit A/D conversion (2000 bits)
  - 2: 16-bit A/D conversion (32000 digits) (*: PCD-300B Series)
  - 3: 24-bit A/D conversion (8200000 digits) (*: P CD-300B Series)
  - 4: Code extension 24-bit A/D conversion (8200000 digits)(*: PCD-300B Series)
- Others
  1) This function is set to A/D data format of all the connected PCD units.
  2) When this function is used in PCD-300A Series, relation between lDataType and PCD-300A Series shall be as described in the following.

lDataType	Setting in PCD-300A Series
0	0
1	1
2	1
3	0
4	1

  3) This function is calling PcdDMS inside the function.

4-3-20 Pcd_SetStorageMode

- Function　Sets storage mode and the number of storage data (per channel)
- Definition　__declspec(dllimport) long __stdcall PcdSetStorageMode (long lDataNum)
- Argument　The number of storage data to set lDataNum.
  Maximum 10,000,000-Byte memory is secured in DLL..
  Maximum number of setting available data is as follows.
  10,000,000/Meas CH/2 (For 12-bit to 16-bit A/D conversion)
  10,000,000/Meas CH/4 (For 24-bit A/D conversion)
  However, if '0' is specified, storage mode is cancelled.
- Return value　Return value of this function differs from those of other functions.
  - 0 or less: Error
  - 0: Cancel the storage mode
  - 1 or more: Preset number of storage data
    Or the number of setting available storage data
- Others
  1) When obtaining data in storage mode, always set the storage mode and the number of storage data using this function.
  2) The storage data is enabled until conducting the function Pcd_ClearStorageCount.

4-3-21 Pcd_GetStorageDataFloat

- Function Obtains storage data acquired from the DLL.
- Definition __declspec(dllimport)long __stdcall Pcd_GetStorageDataFlat (long lStartPoint, long lDataNum, float *vfData)
- Argument lStartPoint Load start point (0 or more)
  The number of loaded lDataNum data (per channel)
  *vfData Data obtained from the PCD
- Return value Return value of this function differs from those of other functions.
  0 or less: Error
  0 or more: The number actually obtained data (per channel)

4-3-22 Pcd_CheckStorageData

- Function Checks the number of current storage data.
- Definition __declspec(dllimport) long __stdcall Pcd_CheckStorageData (void)
- Argument None
- Return value Return value of this function differs from those of other functions.
  0 or more: The number of actually obtained data (per channel)
  0: No storage data
  0 or less: Error
- Others Before conducting this function, conduct Pcd_SetStorageMode and start the measurement.

4-3-23 Pcd_ClearStorageCount

- Function Initializes the number of storage data.
- Definition __declspec(dllimport) long __stdcall Pcd_ClearStorageCount (void)
- Argument None
- Return value If this function is conducted, storage data obtained on the DLL side is erased.

4-3-24 Pcd_GetMoniData

- Function: Obtains A/D value monitor data. (Corresponding to the PCD-300B Series)
  For application program, call this function within interval of approximately 0.3 sec and obtain the monitor data.
- Definition: __declspec(dllimport) long __stdcall Pcd_GetMoniData (void *vMoniData)
- Argument: *vMoniData — A/D data outputted by the PCD.
  It is required to separate data assignment on the application side according to A/D data format preset by Pcd_SetDataType.

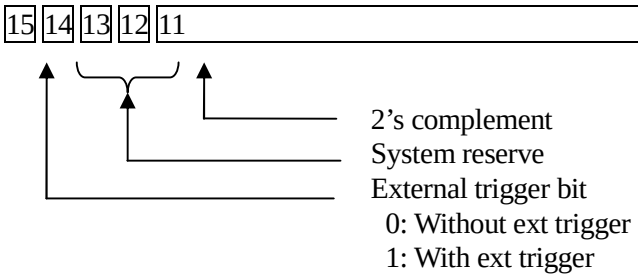
Value set by Pcd_SetDataType		Corresponding assignment type
0:	12-bit A/D conversion data	short
1:	Code extension 12-bit A/D conversion data	short
2:	16-bit A/D conversion data	short
3:	24-bit A/D conversion data	long
4:	Code extension 24-bit A/D conversion data	long

  Channel assignment is described in the following.
  Ex.1) When only one PCD is connected
  CH01, CH2, CH3, CH4, CH1, CH2, CH3, CH4….,
  Ex.2) When two PCD units are connected
  CH1, CH2, CH3, CH4, CH5, CH6, CH7, CH8, CH1, CH2, CH3…..
  The number of data that can be obtained at one time is maximum 6000 data.
  The number of channel assignments of short type vMoniData to be prepared on the application side is required to be ‘the number of connected PCD units × 4 CHs × 6000.’
  Ex.) When only one PCD is connected
  1 × 4 × 6000 = 24000
  Therefore, it is required to define the channel assignment as follows.

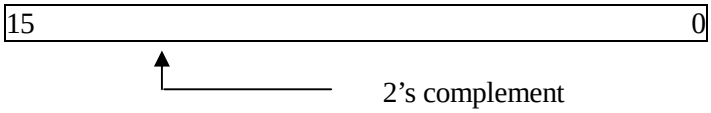
```
short   vsMoniData [24000];
long    vlMoniData [24000];
```

  The application program with the channel assignment less than the above may not properly operate and error occurs.
- Return value: Return value of this function differs with those of other functions.
  0 or less: Return value when error occurred. Contents are same as other functions.
  1: Returns ‘0’ when no monitor data.
  1 or more: The number of data obtained per channel.
- Others
  1) This function is used together with the following functions for obtaining the following A/D monitor data.
  PcdStartAd and PcdStopAd
  2) This function can be used only while monitoring that is started by the PcdStartAd.
  3) Monitor data obtained from the PCD is in the following 1-data/2-byte or 1-data/ 4-byte structure with A/D data format set by Pcd_SetPDataType function.

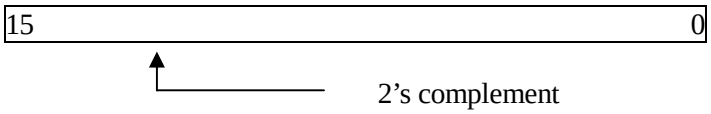
0: External trigger information + System reserve: 3 bits + : 12-bit A/D data



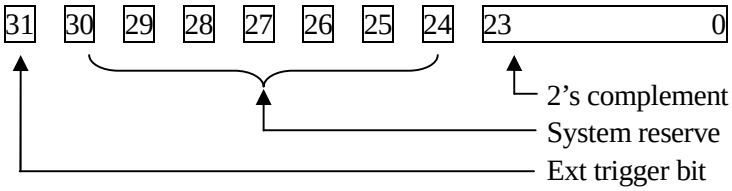
1: Code extension: 12-bit A/D data



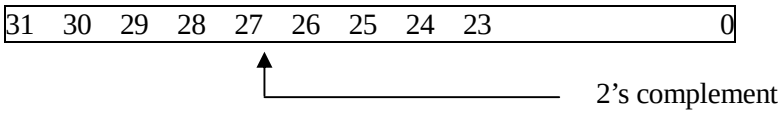
2: 16-bit A/D data (PCD-300B series)



3: External trigger information + 7-bit System reserve + 24-bit A/D data (PCD-300BSeries)



4: Code extension 24-bit A/D data (PCD-300B series)



4) How to convert data into physical value data

Full scale A/D value of the PCD-300B Series is described in the following.

A/D data format set by the Pcd_SetDataType	Corresponding full scale A/D value
0: 12-bit A/D conversion data	±200 digits
1: Code extension 12-bit A/D conversion data	±2000 digits
2: 16-bit A/D conversion data	±32000 digits
3: 24-bit A/D conversion data	±8200000 digits
4: Code extension 24-bit A/D data	±8200000 digits

Therefore, physical value can be obtained by calculating range, CAL coefficient, and offset values in the following expression.

Physical value =

Range/Full scale A/D conversion data × Meas data × CAL coefficient +Offset

Ex.) Suppose Full scale A/D conversion value: 2000, range: 200μm/m, Meas data: 1234, CAL coefficient: 2. 0, and offset: 1.5

200/2000 × 1234 × 2.0 + 1.5 = 248.3

5) This function is calling dPcdGetData inside the function.

4-4 DETAILS OF INDIVIDUAL FUNCTIONS

4-4-1 dPcdGetDevices

- Function　Checks connection state.
- Definition　__declspec(dllimport) DWORD __stdcall dPcdGetDevices (DWORD PcdNo, DWORD *Connect)
- Argument
  - PcdNo: Set PCD Nos. (0 to 3) for checking connection.
  - *Connect: PCD connected state
    - 0: Unconnected
    - 1: Connected
- Others　Always check the connected state while stopping monitoring. Checking during monitoring may fail to obtain the proper connected state.

4-4-2 dPcdStart

- Function　Starts loading monitor data from the PCD via USB driver.
- Definition　__declspec(dllimport) DWORD __stdcall dPcdStart (DWORD PcdNo)
- Argument　PcdNo: Set PCD No. (0 to 3) for obtaining monitor data.
- Others
  1) To start monitoring data, after starting the PCD A/D with the PcdSTA function, it is required to have the USB driver to be in data receive state by the dPcdStart function
  2) When multiple PCD units are connected, after setting the slave PCD to monitor start state, it is required to set the master PCD to monitor start state. When multiple slave PCD units exist, any slave PCD can be set in monitor start state.
     Ex.) Monitor start order when two PCD units ID Nos. 1 and 2 are connected and PCD ID 1 is set as the master PCD
     (1) dPcdSTA ← Start A/D conversion of the slave PCD.
     (2) dPcdStart (1) ← Start loading USB driver for the slave PCD.
     (3) PcdSTA (0) ← Start A/D conversion of the master PCD.
     (4) PcdStart (0) ← Start loading USB driver for the master PCD.
  3) Before starting monitoring, set the following items.
     (1) Oscillator ← PcdOSC
     (2) Clock ← PcdCMS
     (3) Sampling frequency ← PcdSPD
     (4) Range ← PcdRNG
     (5) Low pass filter ← PcdFLT (Should be set for the PCD-320A /PCD-330B Series.)
     (6) High pass filter ← PcdCUP (Should be set for the PCD-320A /PCD-330B Series.)
  4) Commands that can be sent to the PCD while monitoring are only PcdSTP and dPCDGetStatus. If any other commands are sent, error occurs.

4-4-3 dPceStop

- Function　　USB driver stops loading monitor data from the PCD.
- Definition　　__declspec(dllimport) DWORD __stdcall dPcdStop (DWORD PcdNo)
- Argument　　PcdNo:　　Set PCD No. (0 to 3) for stopping loading monitor data.
- Others
  1) To stop monitoring data, after stopping the A/D conversion of the PCD by the PcdSTP function, it is required to set the slave PCD to monitor stop state.
  2) When multiple PCD units are connected, after setting the master PCD to monitor stop state, it is required to set the salve PCD to monitor stop state.

  Ex.) Monitor stopping order when PCD IDs 1 and 2 are connected and ID 0 is set as the master PCD
  (1) PcdSTP (0)　　← Stop A/D conversion of the master PCD
  (2) dPcdStop (0)　　← Stop loading USB driver of the master PCD
  (3) PcdSTOP (1)　　← Stop A/D conversion of the slave PCD
  (4) dPcdStop (1)　　← Stop loading USB driver of the slave PCD

4-4-4 dPcdGetData

- Function: Load monitor data via USB driver for the PCD.
- Definition: __declspec(dllimport) DWORD __stdcall dPcdGetData (DWORD PcdNo, void *pSizeStruct, char *pReciveData)
- Argument:
  - PcdNo: Set PCD No. (0 to 3) for loading monitor data.
  - *pSizeStruct: Pointer of SIZE_INFO structure

```
Typedef struct _size_info {
  DWORD dwReqSize:      The number of desired bytes
  DWORD dwBGetSize:     The number of actually obtained bytes
                        (Set multiple number of 8.)
  DWORD dwBufferSize:   The number of bytes remained on the ring
                        buffer.
}SIZE_INFO;
```

Set 0 to dwReqSize and call dPcdGetData to obtain the number of bytes remained in the current ring buffer.

Ring buffer size in the PCD's UB driver is capable of loading maximum 200 kbytes monitor data per PCD.

  - *pReceiveData: Ring buffer size of the USB driver is 200 kbytes per PCD (which may change without prior notice.)
- Others:
  1) This function loads monitor data of one PCD. When multiple PCD units are connected, it is required to load monitor data of the connected PCD units while changing the PCD No. on the application program.
  2) Monitor data outputted from the PCD is in 1-data/2-byte (PCD-300A Series) or 1-data/2-byte or 1-data/4-byte (PCD-300B Series) structure. For formatting data or sorting channels, see items in PcdGetMeasData.

     Example in use When loading data of PCD ID = 1

```
short ReceiveData [100000];
Size_INFO SizeStruct;

Ring buffer size checking
When multiple PCD units are connected, always check the ring buffer size with the master PCD of the clock.
SizeStruct.dwReqSize = 0;
dPcdGetData (0, &SizeStruct, (char*)&ReceiveData[0];

When data exist in ring buffer
If (SizeStruct.dwReqSize >0) {
SizeStruct.dwReqSize = SizeStruct.dwBufferSize;
dPcdGetData (0,&SizeStruct, (char*)&ReceiveData[0]);
}
```

  3) Since the PCD is simultaneously recording data from CH 1 to CH 4, the number of output bytes shall always be multiple number of 8. In addition, when outputting data, since a certain amount of data is stored and outputted, it is not available to use such method as real-time processing by interlocking with sampling frequency.

4-4-5 dPcdGetStatus

- Function　Obtains PCD status.
- Definition　__declspec(dllimport) DWORD __stdcall dPcdGetStatus (DWORD PcdNo, DWORD *Status)
- Argument

PcdNo:	Set PCD No. (0 to 3) for obtaining PCD status.	
*Status	PCD status	
	0:	Stopping A/D conversion
	1:	Not defined

- Others　None

4-4-6 PcdSPD

- Function　Sets sampling frequency.
- Definition　__declspec(dllimport) long __stdcall PcdSPD (long PcdNo, long Spd)
- Argument

PcdNo: Set PCD No. (0 to 3) for setting sampling frequency.

Spd: Sampling frequency

1	1 Hz
2	2 Hz
5	5 Hz
10	10 Hz
20	20 Hz
50	50 Hz
100	100 Hz
200	200 Hz
500	500 Hz
1000	1 kHz
2000	2 kHz
5000	5 kHz
10000	10 kHz (Only for PCD-300B Series)

- Others
  1) Do not forget to set the sampling frequency before starting monitoring.
  2) Once the sampling frequency is set, it is stored in the PCD internal memory even if the power is turned OFF.
  3) When connecting multiple PCD units, irrespective of master or slave PCD, set the same sampling frequency to all the PCD units.
  4) Assuming that a single or multiple PCD-300B Series units are used, when controlling the PCD using Windows 2000, sampling frequency of up to 5 kHz is available.

4-4-7 PcdOMS

- Function　Sets oscillator.
- Definition　__declspec(dllimport) long __stdcall PcdOMS (long PcNo, long Oms)
- Argument
  - PcdNo　Set PCD No445. (0 to 3) for setting oscillator.
  - OMS　Oscillator
    - 0:　Slave
    - 1:　Master
- Others

1) Oscillator and synchronous clock of the PCD-300A and PCD-300B Series become master units when the PCD units are turned ON and when the USB cable is connected/disconnected. Therefore, when connecting multiple PCD units, do not forget to set the oscillator and synchronous clock before starting monitoring.
2) When connecting multiple PCD-300A or PCD-300B Series units, it is required to set master/slave to all the connected PCD-300A and PCD-300B Series units. At this time, always set the 1st PCD as the master and the rest of other PCD-300A and PCD-300B Series units as the slaves.
3) Setting of the oscillator is enabled only for the PCD-300A and PCD-300B Series.
   If it is set to the PCD-320A, it is not error but no signal is sent from the oscillator.
   When PCD-300A and PCD-320A are mixedly connected, it is required to set one PCD-300A as the master unit and the rest of other PCD-300A and PCD-320A units as slave units.

Ex. 1) Assuming ID=1: PCD-300A, ID=2: PCD-300A, ID=3: PCD-320, and ID=4: PCD-320A, when ID=1 is set as the master unit

ID	Oscillator	Synchronous clock
1	Master	Master
2	Slave	Slave
3	Slave	Slave
4	Slave	Slave

Ex.2) Assuming ID=1: PCD-320A, ID=2: PCD-300A, ID=3: PCD-320, and ID=4: PCD-320A, when ID=1 is set as the master unit

ID	Oscillator	Synchronous clock
1	*1	Master
2	Master	Slave
3	Slave	Slave
4	Slave	Slave

*1: No error even if PcdOMS is set. No need t to set the PcdOMS.

4) When PCD-300A and PCD-320A are mixedly connected, considering the control line of the synchronous cables of the PCD-300B/PCD-300B-F, always set the PCD-300B/PCD-300B-F to the front.

Ex. 1) Assume ID=1: PCD-300B, ID=2: PCD-300B-F, ID=3: PCD-300A, and ID=4: PCD-320A

ID	Oscillator	Synchronous clock
1	Master	Master
2	Slave	Slave
3	Slave	Slave
4	Slave	Slave

5) When PCD-300B Series and PCD-300A Series are mixedly connected, no control operation is available with the following connection method.
   ×: PCD-300A Series → PCD-300B Series
   ×: PCD-300B Series → PCD-300A Series → PCD-300B Series

4-4-8 PcdCMS

- Function　Sets synchronous clock.
- Definition　__declspec(dllimport) long __stdcall PcdCMS (long PcNo., long Cms)
- Argument
  PcdNo: Set PCD No (0 to 3) for setting synchronous clock.
  OMS Synchronous clock
  0: Slave
  1: Master
- Others
  1) When the PCD is turned ON or when the USB cable is connected/disconnected. The oscillator and synchronous clock become master. Therefore, when connecting multiple PCD units, do not forget to set the oscillator and synchronous clock before starting monitoring.
  2) When connecting multiple PCD units, it is required to set master/slave to all the connected PCD units. At this time, set one PCD as the master and the rest of the PCD units to slave units.
  3) When the PCD-300B Series, PCD-300A Series are mixedly connected, considering the control line of the synchronous cable of the PCD-300B Series, always set the PCD-300B Series at the head.

4-4-9 PcdRNG

- Function　Sets ranges of channels.
- Definition　__declspec(dllimport) long __stdcall PcdRNG (long PcNo, long *sRng)
- Argument
  PcdNo: Set PCD No. (0 to 3) for setting rang.
  *sRng Pointer of channel assignment set with channel ranges.

	PCD-300A PCD-300B/PCD-300B-F PCD-330B Series(Strain)	PCD-320A PCD-330B Series(Voltage)
0	OFF	OFF
1	200 μm/m	1 V
2	500 μm/m	2 V
3	1000 μm/m	5 V
4	2000 μm/m	10 V
5	5000 μm/m	20 V
6	10000 μm/m	50 V
7	20000 μm/m (PCD-300B Series)	

  Since 4 channels are provided per PCD, it is required to set 4 channel assignments.
  Ex.) PCD-300A
  Set channels in due order CH1 = 200 μm/m, CH2 = OFF, CH3 = 1000 μm/m, and CH4 = 500μm/m

```
long sRng[4];
sRng[0] = 1;
sRng[1] = 0;
sRng[2] = 3;
sRng[3] = 2;
```

  Range should be set for 4 channels.
- Others　Setting range is only available while stopping monitoring.

4-4-10 PcdFLT

- Function: Sets low pass filter of channels.
  This function is dedicated for PCD-320A and PCD-300B-F and PCD-330B Series.
- Definition: __declspec(dllimport) long __stdcall PcdFLT (long PcNo., long *sFlt)
- Argument: PcdNo Set PCD No. (0 to 3) for setting low pass filter.

	PCD-300A PCD-330B Series(Voltage)	PCD-300B-F PCD-330B Series(Strain)
0:	FLAT	FLAT
1:	10 Hz	10 Hz
2:	30 Hz	30Hz
3:	100 Hz	100 Hz
4:	300 Hz	

Since 4 channels are provided per PCD, it is required to set 4 channel assignments

Ex.) PCD ID = 1

Set channels in due order CH1 = FLAT, CH2 = 10 Hz, CH3 = 30 Hz, and CH4 = 100 Hz.

```
long sFlt[4];
sFlt[0] = 0;
sFlt[1] = 1;
sFlt[2] = 2;
sFlt[3] = 3;
```

Low pass filter should be set for 4 channels.

- Others: Setting low pass filter is only available while stopping monitoring.

4-4-11 PcdCUP

- Function: Sets high pass filter of channels.
  This function is dedicated for the PCD-320A and PCD-330B Series.
- Definition: __delcspec(dllimport) long __stdcall PcdCUP (long PcNo, long *sCup)
- Argument: PcdNo: Set PCD No. (0 to 3) for setting low pass filter.

0:	OFF
1:	0.2 Hz

Since 4 channels are provided per PCD, it is required to set 4 channel assignments.

Ex.) PCD ID = 1

Set channels in due order CH1 = DC, CH2 = AC, CH3 = AC, and CH4 = DC.

```
long sCup[4];
sCup[0] = 0;
sCup[1] = 1;
sCup[2] = 1;
sCup[3] = 0;
PcdFLT (0, &sCup[0]);
```

High pass filter should be set for 4 channels.

- Others: Setting high pass filter is only available while stopping monitoring.

4-4-12 PcdTES

- Function　Conducts self-test.
- Definition　__declspec(dllimport) long __stdcall PcdTES (long PcdNo)
- Argument　PcdNo　Set PCD No. (0 to 3) for conducting self-test.
- Others
  1) This function is only available while stopping monitoring.
  2) For the PCD-300A and PCD-300B Series, before starting the self-test, do not forget to set the PCdOMS.
  3) The PcdTED is a function that only send the self-test command to the PCD. After sending this command, obtain the PCD status by function PcdLoadStatus. When the PCD status becomes '0,' load the self-test results by function PcdTER.

4-4-13 PcdTER

- Function　Loads the self-test results.
- Definition　__declspec(dllimport) long __stdcall PcdTER (long PcdNo, long *rTer)
- Argument　PcdNo　Set PCD No. (0 to 3) for loading the self-test results.
  *rTer　Self-test results of channels.
  0: Normally finished
  1: Error occurred
  Since 4 channels are provided for per PCD, it is required to set 4 channel assignments.
  rTer[0] CH1 self-test results
  rTer[1] CH2 self-test results
  rTer[2] CH3 self-test results
  rTer[3] CH4 self-test results
- Others
  1) Loading the self-test results is only available while stopping monitoring.
  2) Once loaded self-test results are maintained in the PCD while the PCD is turned ON but it is initialized when the PCD is turn OFF.

4-4-14 PcdBAL

- Function: Conducts balance.
  This function is dedicated for PCD-300A and PCD-300B Series.
- Definition: __declspec(dllimport) long __stdcall PcdBAL (long PcdNo, long *sBal)
- Argument:
  PcdNo: Set PCD No. (0 to 3) for conducting balance.
  *sBal Set balance channel.
  0: Conduct balance.
  1: No balance is conducted
  Set balance conducting channel for 4 channels.
  Ex.) When conducting balance of CH1 and CH3

```
long sBal[4];
sBal[0] = 1;
sBal[1] = 0;
sBal[2] = 1;
sBal[3] = 0;
PcdBAL (0, &sBal[0]);
```

- Others
  1) Conducting balance is only available while stopping monitoring.
  2) Before conducting balance, do not forget to set functions PCdOMS and PcdRNG.
  3) This function PcdBAL only sends balance command to the PCD-300A/PCD-300B Series.
     After sending this command, obtain status of the PCD-300A/PCD-300B Series with function PcdReadeStatus.
     If the status of the PCD-300A/PCD-300B Series becomes '0,' load the balance results with function PcdBAR.

4-4-15 PcdBAR

- Function: Loads balance results.
  This function is dedicated for PCD-300A and PCD-300B Series.
- Definition: __declspec(dllimport) long __stdcall PcdBAR (long PcdNo, long *sBar)
- Argument:
  PcdNo: Set PCD No. (0 to 3) for loading balance results.
  *sBar Balance results of channels.
  -1: Balance not conducted
  0: Balance OK
  1: Resistance balance over
  2: Range OFF
  3: Capacity balance over
  4: Connection failure
  Since 4 channels are provided for the PCD, it is required to set 4 channel assignments.
  rBar[0] CH1 balance results
  rBar[1] CH2 balance results
  rBar[2] CH3 balance results
  rBar[3] CH4 balance results
- Others
  1) Loading balance results is only available while stopping monitoring.
  2) Once loaded balance results are stored in the PCD while the PCD is turned ON but initialized when the PCD is turned OFF.

4-4-16 PcdSTA

- Function Starts A/D conversion of the PCD.
- Definition __declspec(dllimport) long __stdcall PcdSTA (long PcdNo)
- Argument PcdNo: Set PCD No. (0 to 3) for starting A/D conversion.
- Others See description on dPcdStart.

4-4-17 PcdSTP

- Function Stops A/D conversion of the PCD.
- Definition __declspec(dllimport) long __stdcall PcdSTP (long PcdNo)
- Argument PcdNo: Set PCD No. (0 to 3) for stopping A/D conversion.
- Others See description on dPcdStop.

4-4-18 PcdPRR

- Function　Loads PCD parameters.
- Definition　__declspec(dllimport) long __stdcall PcdPRR (long PcdNo, long sPrr, long *Prr)
- Argument　PcdNo:　Set PCD No. (0 to 3) for loading parameters.
  sPrr　Set desired parameter types.

0x00000001	Type	
0x00000002	Software version	
0x00000003	FPGA version	
0x00000004	Sampling frequency	
0x00000005	Oscillator	
0x00000006	Synchronous clock	
0x00000007	Range	
0x00000008	Filter	
0x00000009	High pass filter	
0x0000000A	Meas CH mode	(*: See Pcd_Mod.)
0x00000201	TEDS unit	(*: See Pcd_TED.)
0x00000202	TEDS range	(*: See Pcd_TED.)
0x00000203	TEDS filter	(*: See Pcd_TED.)
0x00000204	TEDS ε or	(*: See Pcd_TED.)
0x00000205	TEDS CH mode	(*: See Pcd_TED.)

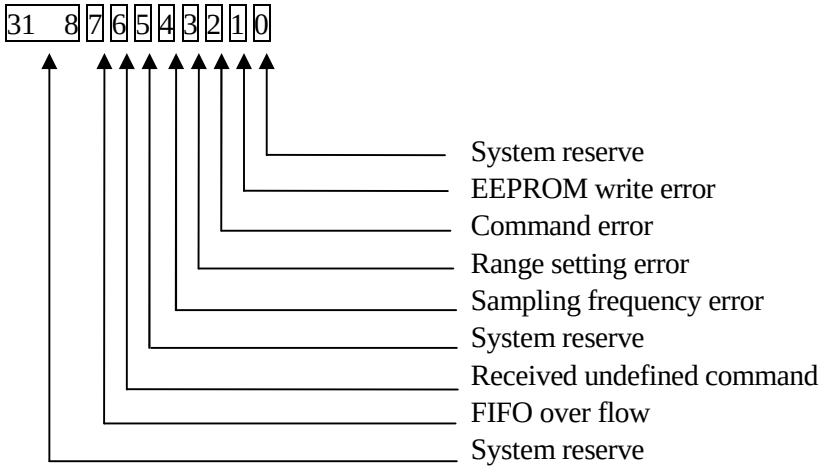
  *rPrr　Loaded parameters
  Number of the required rPrr assignments varies with a value preset in the sPrr.

sPrr	The number of rPrr assignments
0x00000001	1
0x00000002	1
0x00000003	1
0x00000004	1
0x00000005	1
0x00000006	1
0x00000007	4
0x00000008	4
0x00000009	4
0x0000000A	4
0x00000201	4
0x00000202	4
0x00000203	4
0x00000204	4
0x00000205	4

- Others　Loading parameters is only available while stopping monitoring.

4-4-19 PcdERR

- Function　　Loads PCD error information.
- Definition　　__declspec(dllimport) long __stdcall PcdERR (long PcdNo, long *Err)
- Argument　　PcdNo:　Set PCD No. (0 to 3) for loading error information.
  *Err　Set the loaded error information.
  0:　Normal
  Other than 0:　Error occurred (Error contents are corresponding to bit)



- Others　　Loading error information is available only while stopping monitoring.

4-4-20 PcdECL

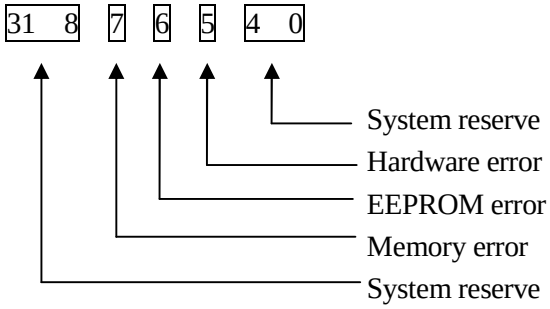
- Function　　Initializes PCD error information.
- Definition　　__declspec(dllimport) long __stdcall PcdECL (long PcdNo)
- Argument　　PcdNo:　Set PCD No. (0 to 3) for initializing error information.
- Others　　Initializing error information is only available while stopping monitoring.

4-4-21 PcdCHK

- Function　　Checks PCD hardware.
- Definition　　__declspec(dllimport) long __stdcall PcdCHK (long PcdNo)
- Argument　　PcdNo:　Set PCD No. (0 to 3) for checking the hardware.
- Others　　Checking the hardware is only available while stopping monitoring.

4-4-22 PcdCHR

- Function　　Loads PCD hardware check results.
- Definition　　__declspec(dllimport) long __stdcall PcdCHR (long PcdNo, long *Ch4)
- Argument　　PcdNo:　Set PCD No. (0 to 3) for initializing hardware check results.
  *Chr　Hardware check result (Corresponding to bit)

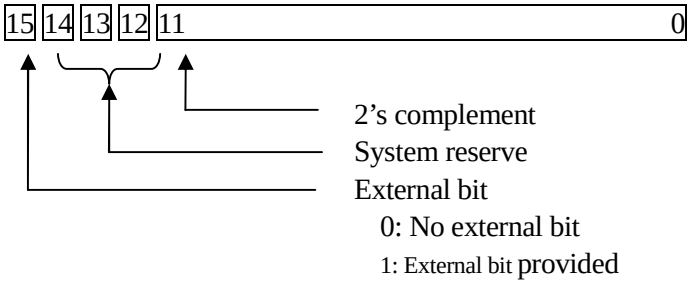


- Others　　1) Loading hardware check results is only available while stopping monitoring.
  2) The PCD automatically conducts hardware check with the power turned ON. When error occurred in hardware check, power LED flickers.

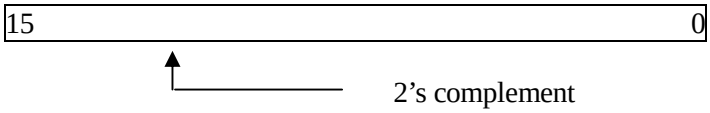
4-4-23 PcdDMS

- Function　Sets A/D data format.
- Definition　__declspec(dllimport) long __stdcall PcdDMS (long PcdNo, long Dms)
- Argument　PcdNo:　Set PCD No. (0 to 3) for setting A/D data format.
  Dms　A/D data format

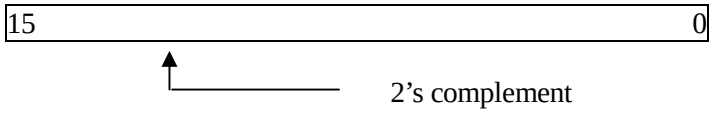
  0　External trigger information + 3-bit system reserve + 12-bit A/D conversion data



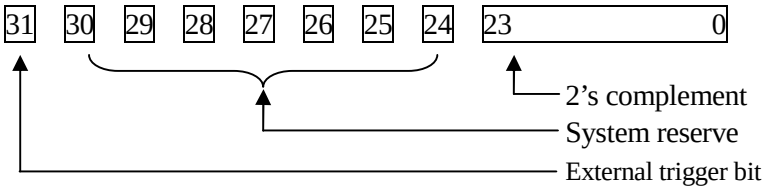
  1　Code extension 12-bit A/D conversion data



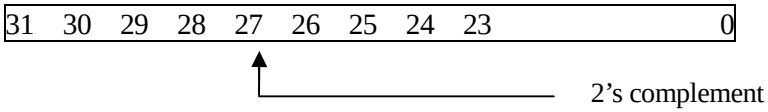
  2　16-bit A/D conversion data (PCD-300B series)



  3　External trigger information + 7-bit system reserve + 24-bit A/D conversion data (PCD-300B series)



  4　Code extension 24-bit A/D conversion data (PCD-300B series)



- Others
  1) Setting A/D data format is only available while stopping monitoring.
  2) When the PCD power is turned ON, Dms=0(PCD-300ASeries),Dms=3(PCD-300B Series).
  3) Dms=2 to 4 is dedicated for the PCD-300B series.

4-4-24 Pcd_MOD

- Function Sets Meas mode of PCD channels. (Dedicated for the PCD-300B Series)
- Definition __declspec(dllimport) long __stdcall Pcd_MOD (long PcdNo, long *ChMod)
- Argument PcdNo Set PCD No. (0 to 3) for setting Meas mode.
  *ChMod Pointer of channel assignment preset with Meas mode of channels.
  0: 1G mode
  1: 2G mode
  2: 4G mode
  3: Voltage mode(PCD-330B Series)
  Since 4 channels are provided for the PCD, it is required to set 4 channel assignments.
  Ex.) Assume Meas mode is set in due order CH1 = 1G mode, CH2 = 2G mode, CH3 = 4G mode, and CH4 = 4G mode.
  long ChMod[4];
  ChMod[0] = 0;
  ChMod[1] = 1;
  ChMod[2] = 2;
  ChMod[3] = 2;
  Setting the Meas mode should be set for 4 channels.
- Others This function is dedicated for the PCD-300B series.

4-4-25 Pcd_TED

- Function Starts loading TEDS information to the PCD. (Dedicated for the PCD-300B Series)
- Definition __declspec(dllimport) long __stdcall Pcd_TED (long PcdNo)
- Argument PcdNo Set PCD No. (0 to 3) for start loading the TEDS information.
- Others This function is dedicated for the PCD-300B series.

4-4-26 Pcd_TEJ

- Function Obtains TEDS information loaded results in the PCD. (Dedicated for the PCD-300B Series)
- Definition __declspec(dllimport) long __stdcall Pcd_TEJ (long PcdNo, long *rTED)
- Argument PcdNo Set PCD No. (0 to 3) for obtaining TEDS information loaded results.
  *rTED Pointer of channel assignment for obtaining TEDS information of channels.
  0: Succeeded in loading the TEDS information
  1: Error in loading the TEDS information
  Since PCD is provided with 4 channels, it is required to set 4 channel assignments.
  rTED[0] TEDS information loaded results of CH1
  rTED[1] TEDS information loaded results of CH2
  rTED[2] TEDS information loaded results of CH3
  rTED[3] TEDS information loaded results of CH4
- Others This function is dedicated for the PCD-300B series.

4-4-27 PcdPRR_Float

- Function: Loads PCD parameters. (When return value is float data type.)
- Definition: __declspec(dllimport) long __stdcall PcdPRR_Float (long PcdNo, long sPrr, float *Prr)
- Argument:
  - PcdNo: Set PCD No. (0 to 3) for loading parameters.
  - sPrr: Set the desired parameter type to be loaded

0x00000200	TEDS calibration coefficient
Oxoooo11oo	Cable compensation coefficient

  - *rPrr: Loaded parameters

    The number of required rPrr assignments varies according to the value preset in rPrr.

sPrr	The number of rPrr assignments
0x00000200	4
0x00001100	4

- Others: Loading parameters is only available while stopping monitoring.