

PISO-CPM100(U) Library Transformation

How to build the application program by using the new version PISO-CPM100(U) library instead of the old version one

Beside the bug modification, the new version library (version 2.00 later) of the PISO-CPM100(U) offers more functionality such as slave functions, listen mode, non-blocking mode, and so forth. The new version library and firmware are not compatible with the old version library and firmware. Therefore, if you use the PISO-CPM100(U) with the old version firmware, your application program must be built by using the old version library. This document guides you to modify your old application program by using the new version APIs if you want to apply your old program on the PISO-CPM100(U) with the new version firmware.

Because this document only show you how to transfer your application, it only lists the differences between old version library and new version one, and gives an example to demonstrate how to apply the new version API instead of the old version one. About the descriptions of the parameters of these APIs, please refer to the PISO-CPM100U_user_manual.pdf.

In order to make the descriptions more simple and clear, the marks for the old version APIs and the new version ones are given as **[Old]** and **[New]** respectively, as shown in following table.

Keyword	Description
[Old]	The version numbers of the PISO-CPM100(U) library is before 2.00.
[New]	The version numbers of the PISO-CPM100(U) library is version 2.00 or later.

Tables of Content

T1.	CPM100_GetVersion / CPM100_GetVersion	3
2.	CPM100_TotalBoard / CPM100_TotalBoard.....	4
3.	CPM100_GetBoardSwitchNo / CPM100_GetBoardSwitchNo	5
4.	CPM100_GetBoardInf / CPM100_GetBoardInf.....	6
5.	CPM100_ActiveBoard / CPM100_InitMaster.....	7
6.	CPM100_CloseBoard / CPM100_ShutdownMaster	8
7.	CPM100_GetFirmwareVersion / CPM100_GetFirmwareVersion	9
8.	CPM100_GetCANStatus / CPM100_GetCANStatus	10
9.	CPM100_AddNode / CPM100_AddNode	11
10.	CPM100_RemoveNode / CPM100_RemoveNode	12
11.	CPM100_GetNodeList / CPM100_GetNodeList.....	13
12.	CPM100_NMTChangeState / CPM100_NMTChangeState	14
13.	CPM100_NMTGetState / CPM100_NMTGetState	15
14.	CPM100_NMTGuarding / CPM100_NMTGuarding	16
15.	CPM100_SendSYNC / CPM100_SendSYNCMsg.....	17
16.	CPM100_GetSYNCingID / CPM100_GetCyclicSYNCInfo.....	18
17.	CPM100_ChangeSYNCID / CPM100_ChangeSYNCID	19
18.	CPM100_ChaneEMCYID / CPM100_ChangeEMCYID.....	20
19.	CPM100_ReadEMCY / CPM100_GetEMCYData	21
20.	CPM100_SDOAbortTransmit / CPM100_SDOAbortTransmit	22
21.	CPM100_SDOReadData / CPM100_SDOReadData	23
22.	CPM100_SDOWriteData / CPM100_SDOWriteData.....	24
23.	CPM100_ReadPDOMessage / CPM100_GetPDOLastData	25
24.	CPM100_InstallPDO / CPM100_InstallPDO	26
25.	CPM100_DynamicPDO / CPM100_DynamicPDO	27
26.	CPM100_RemovePDO / CPM100_RemovePDO	28
27.	CPM100_PDOWrite / CPM100_PDOWrite	29
28.	CPM100_PDORemote / CPM100_PDORemote	30
29.	CPM100_PDOTxType / CPM100_PDOTxType.....	31
30.	CPM100_SetEventTimer / CPM100_PDORemote	32
31.	CPM100_ChangePDOCobID / CPM100_ChangePDOID.....	33
32.	CPM100_COBIDInfo / CPM100_GetRxDPOID / CPM100_GetTxPDOID / CPM100_GetSYNCID / CPM100_GetEMCYID	34
33.	CPM100_PDOMappingInfo / CPM100_GetPDOMapInfo	35
34.	CPM100_ReadPDOCCount	36
35.	I87123_WriteDO / I87123_ReadDI / I87123_WriteAO / I87123_ReadAI	37

1. CPM100_GetVersion / CPM100_GetVersion

[Old]:

```
float CPM100_GetVersion(void);
```

[New]:

```
WORD CPM100_GetVersion(void);
```

Examples

[Old]:

```
float Ver;
```

```
Ver = CPM100_GetVersion();  
printf("Ver = %f \n", Ver); // Ver = 1.00
```

[New]:

```
WORD Ver;
```

```
Ver = CPM100_GetVersion();  
printf("Ver = %d \n", Ver); // Ver = 200
```

2. CPM100_TotalBoard / CPM100_TotalBoard

[Old]:

```
int CPM100_TotalBoard(void);
```

[New]:

```
WORD CPM100_TotalBoard(void);
```

Examples

[Old]:

```
int TotalBoard;
```

```
TotalBoard = CPM100_TotalBoard();  
printf("Total Board = %d \n", TotalBoard);
```

[New]:

```
WORD TotalBoard;
```

```
TotalBoard = CPM100_TotalBoard();  
printf("Total Board = %d \n", TotalBoard);
```

3. CPM100_GetBoardSwitchNo / CPM100_GetBoardSwitchNo

[Old]:

```
int CPM100_GetBoardSwitchNo(BYTE BoardCntNo,  
                             BYTE *BoardSwitchNo);
```

[New]:

```
WORD CPM100_GetBoardSwitchNo(BYTE BoardCntNo,  
                              BYTE *BoardNo);
```

Examples

[Old]:

```
int ret;  
BYTE BoardNo;  
  
ret = CPM100_GetBoardSwitchNo(1, &BoardNo);  
printf("Board DIP Switch = %d \n", BoardNo);
```

[New]:

```
WORD ret;  
BYTE BoardNo;  
  
ret = CPM100_GetBoardSwitchNo(1, &BoardNo);  
printf("Board DIP Switch = %d \n", BoardNo);
```

4. CPM100_GetBoardInf / CPM100_GetBoardInf

[Old]:

```
int CPM100_GetBoardInf(BYTE bBoardNo, DWORD *dwVID,  
                      DWORD *dwDID, DWORD *dwSVID,  
                      DWORD *dwSDID, DWORD *dwSAuxID,  
                      DWORD *dwIrqNo);
```

[New]:

```
WORD CPM100_GetBoardInf(BYTE bBoardNo, DWORD *dwVID,  
                        DWORD *dwDID, DWORD *dwSVID,  
                        DWORD *dwSDID, DWORD *dwSAuxID,  
                        DWORD *dwIrqNo);
```

Examples

[Old]:

```
int ret;  
DWORD dwVID, dwDID, dwSVID, dwSDID;  
DWORD dwSAuxID, dwIrqNo  
  
ret = CPM100_GetBoardInf(0, &dwVID, &dwDID, &dwSVID,  
                          &dwSDID, &dwSAuxID, &dwIrqNo);
```

[New]:

```
WORD ret;  
DWORD dwVID, dwDID, dwSVID, dwSDID;  
DWORD dwSAuxID, dwIrqNo  
  
ret = CPM100_GetBoardInf(0, &dwVID, &dwDID, &dwSVID,  
                          &dwSDID, &dwSAuxID, &dwIrqNo);
```

5. CPM100_ActiveBoard / CPM100_InitMaster

[Old]:

```
int CPM100_ActiveBoard(BYTE BoardNo);  
int CPM100_InitMaster(BYTE BoardNo, BYTE Baudrate);
```

[New]:

```
WORD CPM100_InitMaster(BYTE BoardNo, BYTE Node,  
                       BYTE BaudRate, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ret = CPM100_ActiveBoard(0);  
if(ret == 0){  
    ret = CPM100_InitMaster(0, 7);  
    if(ret != 0) return ret; // return error  
}  
else{  
    return ret; // return error  
}
```

[New]:

```
WORD ret;  
  
ret = CPM100_InitMaster(0, 0, 7, 1);  
if(ret != 0) return ret; // return error  
  
/* Initialize PISO-CPM100(U) in board 0 with master ID 0 and  
CAN baud 7 (1 M bps). The mode of the API is 1  
(block-mode). */
```

6. CPM100_CloseBoard / CPM100_ShutdownMaster

[Old]:

```
int CPM100_CloseBoard(BYTE BoardNo)
int CPM100_ShutdownMaster(BYTE BoardNo);
```

[New]:

```
WORD CPM100_ShutdownMaster(BYTE BoardNo);
```

Examples

[Old]:

```
int ret;

ret = CPM100_ShutdownMaster(0);
if(ret == 0){
    ret = CPM100_CloseBoard(0);
    if(ret != 0) return ret; // return error
}
else{
    return ret; // return error
```

[New]:

```
WORD ret;

ret = CPM100_ShutdownMaster(0);
if(ret != 0) return ret; // return error
```


7. CPM100_GetFirmwareVersion / CPM100_GetFirmwareVersion

[Old]:

```
int CPM100_GetFirmwareVersion(BYTE BoardNo, WORD *Version);
```

[New]:

```
WORD CPM100_GetFirmwareVersion(BYTE BoardNo,  
                                WORD *Fir_Ver, BYTE BlockMode);
```

Examples

[Old]:

```
Int ret;  
WORD Ver;  
  
ret = CPM100_GetFirmwareVersion(0, &Ver);  
printf("Ver = %d \n", Ver);           // Ver = 100
```

[New]:

```
WORD ret;  
WORD Ver;  
  
ret = CPM100_GetFirmwareVersion(0, &Ver, 1);  
printf("Ver = %d \n", Ver);           // Ver = 200
```

8. CPM100_GetCANStatus / CPM100_GetCANStatus

[Old]:

```
int CPM100_GetCANStatus(BYTE BoardNo, BYTE *Status)
```

[New]:

```
WORD CPM100_GetCANStatus(BYTE BoardNo, BYTE *Status);
```

Examples

[Old]:

```
int ret;  
BYTE Status  
  
ret = CPM100_GetCANStatus(0, &Status);
```

[New]:

```
WORD ret;  
BYTE Status  
  
ret = CPM100_GetCANStatus(0, &Status);
```

9. CPM100_AddNode / CPM100_AddNode

[Old]:

```
int CPM100_AddNode(BYTE BoardNo, BYTE Node);
```

[New]:

```
WORD CPM100_AddNode(BYTE BoardNo, BYTE Node,  
                    BYTE AddMode, WORD DelayTime,  
                    WORD ResTimeout, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ret = CPM100_AddNode(0, 1); // Add node 1 slave in board 0
```

[New]:

```
WORD ret;  
  
ret = CPM100_AddNode(0, 1, 1, 1, 200, 1);  
/* Add a slave with node 1 by using automatic mode. The  
time interval of the CAN message sent to the slave is 1 ms.  
The response timeout value of the CAN message is 200 ms.  
The mode of the API is 1 (block-mode). */
```

10. CPM100_RemoveNode / CPM100_RemoveNode

[Old]:

```
int CPM100_RemoveNode(BYTE BoardNo, BYTE Node);
```

[New]:

```
WORD CPM100_RemoveNode(BYTE BoardNo, BYTE Node,  
                        BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ret = I87123_RemoveNode(0, 1); // Remove node 1 slave
```

[New]:

```
WORD ret;  
  
ret = I87123_RemoveNode(0, 1, 1);
```

11. CPM100_GetNodeList / CPM100_GetNodeList

[Old]:

```
int CPM100_GetNodeList(BYTE BoardNo, BYTE *NodeList)
```

[New]:

```
WORD CPM100_GetNodeList(BYTE BoardNo, BYTE *NodeList,  
                        BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
BYTE NodeList[16] = {0};  
  
ret = CPM100_GetNodeList(0, NodeList);
```

[New]:

```
WORD ret;  
BYTE NodeList[16] = {0};  
  
ret = CPM100_GetNodeList(0, NodeList, 1);
```

12. CPM100_NMTChangeState / CPM100_NMTChangeState

[Old]:

```
int CPM100_ChangeState(BYTE BoardNo, BYTE Node,  
                       BYTE State);
```

[New]:

```
WORD CPM100_NMTChangeState(BYTE BoardNo, BYTE Node,  
                             BYTE State, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ret = CPM100_ChangeState(0, 1, 0x01);
```

[New]:

```
WORD ret;  
  
ret = CPM100_NMTChangeState(0, 1, 0x01, 1);
```

13. CPM100_NMTGetState / CPM100_NMTGetState

[Old]:

```
int CPM100_NMTGetState(BYTE BoardNo, BYTE Node,  
                       BYTE *State);
```

[New]:

```
WORD CPM100_NMTGetState(BYTE BoardNo, BYTE Node,  
                        BYTE *State, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
BYTE state;  
  
ret =CPM100_NMTGetState(0, 1, &state);
```

[New]:

```
WORD ret;  
BYTE state;  
  
ret = CPM100_NMTGetState(0, 1, &state, 1);
```

14. CPM100_NMTGuarding / CPM100_NMTGuarding

[Old]:

```
int CPM100_NMTGuarding(BYTE BoardNo, BYTE Node,  
                       WORD GuardTime, BYTE LiftTime);
```

[New]:

```
WORD CPM100_NMTGuarding(BYTE BoardNo, BYTE Node,  
                        WORD GuardTime, BYTE LiftTime, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ret =CPM100_NMTGuarding(0, 1, 1000, 3);
```

[New]:

```
WORD ret;  
  
ret = CPM100_NMTGuarding(0, 1, 1000, 3, 1);
```


15. CPM100_SendSYNC / CPM100_SendSYNCMsg

[Old]:

```
int CPM100_SendSYNC(BYTE BoardNo, WORD Cobid,  
                   DWORD SyncCycle);
```

[New]:

```
WORD CPM100_SendSYNCMsg(BYTE BoardNo, WORD Cobid,  
                        WORD Timer, DWORD Times, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
BYTE cyclic = 1;  
  
if(cyclic) // Send SYNC message per 1000 ms  
    ret = CPM100_SendSYNC(0, 0x80, 10000);  
else // Send SYNC message once or Stop SYNC  
    ret = CPM100_SendSYNC(0, 0x80, 0);  
  
/* The unit of the parameter SyncCycle is 0.1 ms */
```

[New]:

```
WORD ret;  
BYTE cyclic = 1;  
  
if(cyclic) // Send SYNC message per 1000 ms  
    ret = CPM100_SendSYNCMsg(0, 0x80, 1000, 0, 1);  
else // Send SYNC message once  
    ret = CPM100_SendSYNCMsg(0, 0x80, 1, 1, 1);  
/* The unit of the parameter SyncCycle is 1 ms */  
/* Set parameter times to 1 time (0: send sync  
continuously, others: send sync by the specific times) */
```

16. CPM100_GetSYNCingID / CPM100_GetCyclicSYNCInfo

[Old]:

```
int CPM100_GetSYNCingID(BYTE BoardNo, BYTE *IdNum,  
                        WORD *SYNCIdList, DWORD *SyncCycle);
```

[New]:

```
WORD CPM100_GetCyclicSYNCInfo(BYTE BoardNo,  
                                WORD *Cobid WORD *Timer, DWORD *Times,  
                                BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
BYTE IdNum = 0;  
WORD IdList[5] = {0};  
DWORD SyncCycle[5] = {0};  
  
ret = CPM100_GetSYNCingID(0, &IdNum, IdList, SyncCycle);  
/* The unit of the parameter SyncCycle is 0.1 ms */
```

[New]:

```
WORD ret;  
WORD IdList[5] = {0};  
WORD SyncCycle[5] = {0};  
DWORD Times[5] = {0};  
  
ret = CPM100_GetCyclicSYNCInfo(0, IdList, SyncCycle,  
                                Times, 1);  
/* The unit of the parameter SyncCycle is 1 ms */
```

17. CPM100_ChangeSYNCID / CPM100_ChangeSYNCID

[Old]:

```
int CPM100_ChangeSYNCID(BYTE BoardNo, BYTE Node,  
                        WORD Cobid);
```

[New]:

```
WORD CPM100_ChangeSYNCID(BYTE BoardNo, BYTE Node,  
                          WORD Cobid, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ret = CPM100_ChaneSYNCID(0, 1, 0x90);
```

[New]:

```
WORD ret;  
  
ret = CPM100_ChangeSYNCID(0, 1, 0x90, 1);
```

18. CPM100_ChaneEMCYID / CPM100_ChangeEMCYID

[Old]:

```
int CPM100_ChaneEMCYID(BYTE BoardNo, BYTE Node,  
                      WORD Cobid);
```

[New]:

```
WORD CPM100_ChangeEMCYID(BYTE BoardNo, BYTE Node,  
                          WORD Cobid, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ret = CPM100_ChaneEMCYID(0, 1, 0x91);
```

[New]:

```
WORD ret;  
  
ret = CPM100_ChangeEMCYID(0, 1, 0x91, 1);
```

19. CPM100_ReadEMCY / CPM100_GetEMCYData

[Old]:

```
int CPM100_ReadEMCY(BYTE BoardNo, BYTE *Node,  
    WORD *CobId, BYTE *EMCY_Data, WORD *Year,  
    WORD *Month, WORD *DayOfWeek, WORD *Day,  
    WORD *Hour, WORD *Minute, WORD *Second,  
    WORD *Milliseconds);
```

[New]:

```
WORD CPM100_GetEMCYData(BYTE *BoardNo, BYTE *Node,  
    BYTE *Data);
```

Examples

[Old]:

```
int ret;  
BYTE Node,  
WORD CobId;  
BYTE Data[8] = {0},  
WORD Year, Month, DayOfWeek, Day, Hour, Minute;  
WORD Second, Milliseconds;  
  
ret = CPM100_ReadEMCY(0, &Node, &CobId, Data, &Year,  
    &Month, &DayOfWeek, &Day, &Hour, &Minute,  
    &Second, &Milliseconds);
```

[New]:

```
WORD ret;  
BYTE BoardNo, Node;  
BYTE Data[8] = {0},  
  
ret = CPM100_GetEMCYData(&BoardNo, &Node, &Data);  
if(ret == 0){  
    // Get the system time in here  
}
```

20. CPM100_SDOAbortTransmit / CPM100_SDOAbortTransmit

[Old]:

```
int CPM100_SDOAbortTransmit(BYTE BoardNo, BYTE Node,  
                             WORD Index, BYTE Subindex);
```

[New]:

```
WORD CPM100_SDOAbortTransmit(BYTE BoardNo, BYTE Node,  
                              WORD Index, BYTE SubIndex,  
                              DWORD TData, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ret = CPM100_SDOAbortTransmit(0, 1, 0x1008, 0x0);  
  
// The abort data is always 0x00000000
```

[New]:

```
WORD ret;  
  
ret = CPM100_SDOAbortTransmit(0, 1, 0x1008, 0x0,  
                              0x00000000,1);  
  
// 0x00000000 is an example of the abort data
```

21. CPM100_SDOReadData / CPM100_SDOReadData

[Old]:

```
int CPM100_SDOReadData(BYTE BoardNo, BYTE Node,  
                      WORD Index, BYTE Subindex,  
                      WORD *Len, BYTE *Rdata);
```

[New]:

```
WORD CPM100_SDOReadData(BYTE BoardNo, BYTE Node,  
                        WORD Index, BYTE SubIndex, DWORD* RDLen,  
                        BYTE* RData, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
WORD len;  
BYTE rdata[256];  
  
ret = CPM100_ReadSDO(0, 1, 0x1000, 0x0, &len, rdata);
```

[New]:

```
WORD ret;  
DWORD len;  
BYTE rdata [256];  
  
ret = CPM100_SDOReadData(0, 1, 0x1000, 0x0, &len, rdata ,1);
```

22. CPM100_SDOWriteData / CPM100_SDOWriteData

[Old]:

```
int CPM100_SDOWriteData(BYTE BoardNo, BYTE Node,  
                        WORD Index, BYTE Subindex, WORD Len,  
                        BYTE *Tdata, WORD *Rlen, BYTE *Rdata);
```

[New]:

```
WORD CPM100_SDOWriteData(BYTE BoardNo, BYTE Node,  
                          WORD Index, BYTE SubIndex, DWORD TDLen,  
                          BYTE *TData, WORD *RDLen, BYTE *RData,  
                          BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
WORD rlen;  
BYTE rdata [256];  
BYTE tdata[4] = {0xFF, 0, 0, 0};  
  
ret = CPM100_SDOWriteData(0, 1, 0x6200, 0x1, 1, tdata,  
                           &rlen, rdata);
```

[New]:

```
WORD ret;  
WORD rlen;  
BYTE rdata [256];  
BYTE tdata[4] = {0xFF, 0, 0, 0};  
  
ret = CPM100_SDOWriteData(0, 1, 0x6200, 0x1, 1, tdata,  
                           &rlen, rdata, 1);
```


23. CPM100_ReadPDOMessage / CPM100_GetPDOLastData

[Old]:

```
int CPM100_ReadPDOMessage(BYTE BoardNo, WORD *CobId,  
                           BYTE *RDataLen, BYTE *RData,  
                           DWORD *U_time, DWORD *L_time);
```

[New]:

```
WORD CPM100_GetPDOLastData(BYTE BoardNo, WORD Cobid,  
                            BYTE *IsNew, BYTE *DLen,  
                            BYTE *RData, BYTE BlockMode);
```

[Old]:

```
int ret;  
BYTE len, rdata[8];  
WORD cobid = 0;  
DWORD u_time, l_time;  
  
ret = CPM100_ReadPDOMessage(0, &cobid, &len, rdata,  
                             &u_time, &l_time);  
  
if(cobid == 0x181){  
    .....  
}
```

[New]:

```
WORD ret;  
BYTE new, len;  
BYTE rdata[8];  
  
ret = CPM100_GetPDOLastData(0, 0x181, &new, &len,  
                             rdata, 1);  
  
if(new == 1){ // Handle the new obtained message  
    .....  
}
```

24. CPM100_InstallIPDO / CPM100_InstallIPDO

[Old]:

```
int CPM100_InstallIPDO(BYTE BoardNo, BYTE Node,  
                      WORD Cobid, WORD PDOIndex);
```

[New]:

```
WORD CPM100_InstallIPDO(BYTE BoardNo, BYTE Node,  
                        WORD Cobid, BYTE RxTx,  
                        WORD PDO_No, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
// Install a RxPDO5 with 0x333 ID  
ret = CPM100_InstallIPDO(0, 1, 0x333, 0x1405);
```

[New]:

```
WORD ret;  
  
// Install a RxPDO5 with 0x333 ID  
ret = CPM100_InstallIPDO(0, 1, 0x333, 0, 5, 1);
```

25. CPM100_DynamicPDO / CPM100_DynamicPDO

[Old]:

```
int CPM100_DynamicPDO(BYTE BoardNo, BYTE Node,  
                     WORD Cobid, BYTE RxTxType,  
                     BYTE PDOEntry, BYTE *MappingData);
```

[New]:

```
WORD CPM100_DynamicPDO(BYTE BoardNo, BYTE Node,  
                       WORD Cobid, BYTE RxTx, BYTE Entry,  
                       DWORD EntryData, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
BYTE map_tdata[4] = {0x8 0x1, 0x0, 0x062};  
  
// Set map_data to entry 1 of 0x333 ID  
ret = CPM100_DynamicPDO(0, 1, 0x333, 0, 1, map_tdata);
```

[New]:

```
WORD ret;  
DWORD map_tdata = 0x62000108;  
  
// Set map_data to entry 1 of 0x333 ID  
ret = CPM100_DynamicPDO(0, 1, 0x333, 0, 1, map_tdata, 1);
```

26. CPM100_RemovePDO / CPM100_RemovePDO

[Old]:

```
int CPM100_RemovePDO(BYTE BoardNo, BYTE Node,  
                    WORD Cobid, BYTE RxTxType,  
                    BYTE PDOEntry);
```

[New]:

```
WORD CPM100_RemovePDO(BYTE BoardNo, BYTE Node,  
                    WORD Cobid, BYTE Entry, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ret = CPM100_RemovePDO(0, 1, 0x201, 0, 1);  
  
// 0 is RxPDO type  
// Remove RxPDO 0x201 1st entry
```

[New]:

```
WORD ret;  
  
ret = CPM100_RemovePDO(0, 1, 0x201, 1, 1);  
// Remove RxPDO 0x201 1st entry
```

27. CPM100_PDOWrite / CPM100_PDOWrite

[Old]:

```
int CPM100_PDOWrite(BYTE BoardNo, WORD Cobid, BYTE Offset,  
                   BYTE TDataLen, BYTE *TData);
```

[New]:

```
WORD CPM100_PDOWrite(BYTE BoardNo, WORD Cobid,  
                    BYTE Offset, BYTE DLen, BYTE *Data,  
                    BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
BYTE tdata[2] = {0x55, 0xFF};  
  
ret = CPM100_PDOWrite(0, 0x201, 0, 2, tdata);  
// Output 2-byte PDO data with 0x55 and 0xFF
```

[New]:

```
WORD ret;  
BYTE tdata[2] = {0x55, 0xFF};  
  
ret = CPM100_PDOWrite(0, 0x201, 0, 2, tdata, 1);  
// Output 2-byte PDO data with 0x55 and 0xFF
```

28. CPM100_PDORemote / CPM100_PDORemote

[Old]:

```
int CPM100_PDORemote(BYTE BoardNo, WORD Cobid,  
                    BYTE *RDataLen, BYTE *RData,  
                    DWORD *U_Time, DWORD *L_Time);
```

[New]:

```
WORD CPM100_PDORemote(BYTE BoardNo, WORD Cobid,  
                    BYTE *DLen, BYTE *RData,  
                    BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
BYTE len;  
BYTE rdata[8];  
DWORD u_time, l_time;  
  
ret = CPM100_PDORemote(0, 0x181, &len, rdata,  
                      &u_time, &l_time);
```

[New]:

```
WORD ret;  
BYTE len;  
BYTE rdata[8];  
  
ret = CPM100_PDORemote(0, 0x181, &len, rdata, 1);
```

29. CPM100_PDOTxType / CPM100_PDOTxType

[Old]:

```
int CPM100_PDOTxType(BYTE BoardNo, BYTE Node,  
                    WORD Cobid, BYTE TxType);
```

[New]:

```
WORD CPM100_PDOTxType(BYTE BoardNo, WORD Cobid,  
                    BYTE Tx_Type, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ret = CPM100_PDOTxType(0, 1, 0x181, 50);
```

[New]:

```
WORD ret;  
  
ret = CPM100_PDOTxType(0, 0x181, 50, 1);
```

30. CPM100_SetEventTimer / CPM100_PDODoEventTimer

[Old]:

```
int CPM100_SetEventTimer(BYTE BoardNo, BYTE Node,  
                        WORD Cobid, WORD EventTimer);
```

[New]:

```
WORD CPM100_PDODoEventTimer(BYTE BoardNo, WORD Cobid,  
                             WORD Timer, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;
```

```
ret = CPM100_SetEventTimer(0, 1, 0x181, 1000);  
// TxPDO 0x181 message of node 1 responses per 1000 ms.
```

[New]:

```
WORD ret;
```

```
ret = CPM100_PDODoEventTimer(0, 0x181, 1000, 1);  
// TxPDO 0x181 message responses per 1000 ms.
```


31. CPM100_ChangePDOCobID / CPM100_ChangePD OID

[Old]:

```
int CPM100_ChangePDOCobID(BYTE BoardNo, BYTE Node,  
                           WORD Old_Cobid, WORD New_Cobid);
```

[New]:

```
WORD CPM100_ChangePD OID(BYTE BoardNo,  
                          WORD Old_Cobid, WORD New_Cobid,  
                          BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ret = CPM100_ChangePDOCobID(0, 1, 0x181, 0x222);  
// Change node 1 PDO ID 0x181 to 0x222.
```

[New]:

```
WORD ret;  
  
ret = CPM100_ChangePD OID(0, 0x181, 0x222, 1);  
// Change PDO ID 0x181 to 0x222.
```

32. CPM100_COBIDInfo / CPM100_GetRxPDOID / CPM100_GetTxPDOID / CPM100_GetSYNCID / CPM100_GetEMCYID

[Old]:

```
int CPM100_COBIDInfo(BYTE BoardNo, BYTE NodeID,  
                    WORD *SYNCID, WORD *EMCYID,  
                    BYTE *RxPDONum, WORD *RxPDOID,  
                    BYTE *TxPDONum, WORD *TxPDOID);
```

[New]:

```
WORD CPM100_GetRxPDOID(BYTE BoardNo, BYTE Node,  
                      BYTE *PDO_Cnt, WORD *Id_List,  
                      BYTE BlockMode);  
WORD CPM100_GetTxPDOID(BYTE BoardNo, BYTE Node,  
                      BYTE *PDO_Cnt, WORD *Id_List,  
                      BYTE BlockMode);  
WORD CPM100_GetSYNCID(BYTE BoardNo, BYTE Node,  
                     WORD *Cobid, BYTE BlockMode);  
WORD CPM100_GetEMCYID(BYTE BoardNo, BYTE Node,  
                     WORD *Cobid, BYTE BlockMode);
```

Examples

[Old]:

```
WORD RPDO[8], TPDO[8], SYNC, EMCY;  
BYTE RPDONum, TPDONum;  
  
CPM100_COBIDInfo(0, 1, &SYNC, &EMCY, &RPDONum,  
                RPDO, &TPDONum, TPDO);
```

[New]:

```
WORD RPDO[8], TPDO[8], SYNC, EMCY;  
BYTE RPDONum, TPDONum;  
  
CPM100_GetRxPDOID(0, 1, &RPDONum, RPDO, 1);  
CPM100_GetTxPDOID(0, 1, &TPDONum, TPDO, 1);  
CPM100_GetSYNCID(0, 1, &SYNC, 1);  
CPM100_GetEMCYID(0, 1, &EMCY, 1);
```

33. CPM100_PDOMappingInfo / CPM100_GetPDOMapInfo

[Old]:

```
int CPM100_PDOMappingInfo(BYTE BoardNo, WORD Cobld,  
    BYTE *RxTxType, BYTE *PDONo, BYTE *PDOEntry,  
    BYTE *Len, BYTE *RxData, BYTE *MappingData);
```

[New]:

```
WORD CPM100_GetPDOMapInfo(BYTE BoardNo, WORD Cobid,  
    WORD *PDONo, BYTE *RxTx, BYTE *Tx_Type,  
    WORD *EventTimer, BYTE *EntryCnt,  
    DWORD *Map_Data, BYTE BlockMode);
```

Examples

[Old]:

```
Int ret;  
BYTE RxTx, PDONo, Entry, Len;  
BYTE RxData[8] = {0}, MapData[4] = {0};  
  
ret = CPM100_PDOMappingInfo(0, 0x181, &RxTx, &PDONo,  
    &Entry, &Len, RxData, MapData);  
// Get PDO 0x181 information.
```

[New]:

```
WORD ret;  
BYTE RxTx, PDONo, Entry, TxType;  
WORD EventTimer;  
DWORD MapData;  
  
ret = CPM100_GetPDOMapInfo(0, 0x181, &PDONo, &RxTx,  
    &TxType, &EventTimer, &Entry, &MapData, 1);  
// Get PDO 0x181 information.
```

34. CPM100_ReadPDOCount

[Old]:

int CPM100_ReadPDOCount (**BYTE** BoardNo);

[New]:

Not support.

35. I87123_WriteDO / I87123_ReadDI / I87123_WriteAO / I87123_ReadAI

[Old]:

```
int I87123_WriteDO(unsigned char node,  
                  unsigned char dochannel, unsigned char value);
```

```
int I87123_ReadDI(unsigned char node,  
                  unsigned char dichannel, unsigned char* value);
```

```
int I87123_WriteAO(unsigned char node,  
                   unsigned char aochannel, unsigned short value);
```

```
int I87123_ReadAI(unsigned char node,  
                  unsigned char aochannel, unsigned short* value);
```

[New]:

Not support.