

I-87123 Library Transformation

How to build the application program by using the new version

I-87123 library instead of the old version one

Beside the bug modification, the new version library (version 2.00 later) of the I-87123 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 I-87123 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 I-87123 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 APIs instead of the old version ones. About the descriptions of the parameters of these APIs, please refer to the I-87123_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 I-87123 library is before 2.00.
[New]	The version numbers of the I-87123 library is version 2.00 or later.

Note: If there are other I-87K modules on the PAC, remember that, use the function I87123_Pause to pause the I-87123 before accesses other I-87K modules and use I87123_Start to start the I-87123 after completely accesses.

Tables of Content

1.	I87123_GetVersion / I87123_GetVersion.....	3
2.	I87123_Configure / I87123_InitMaster.....	4
3.	I87123_ShutdownMaster / I87123_ShutdownMaster.....	5
4.	I87123_AddNode / I87123_AddNode	6
5.	I87123_RemoveNode / I87123_RemoveNode.....	7
6.	I87123_ChangeState / I87123_NMTChangeState.....	8
7.	I87123_GetState / I87123_NMTGetState.....	9
8.	I87123_Guarding / I87123_NMTGuarding.....	10
9.	I87123_SendSYNC / I87123_SendSYNCMsg.....	11
10.	I87123_ChaneSYNCCID / I87123_ChangeSYNCCID	12
11.	I87123_ChaneEMCYID / I87123_ChangeEMCYID	13
12.	I87123_AbortSDO / I87123_SDOAbortTransmit	14
13.	I87123_ReadSDO / I87123_SDOReadData	15
14.	I87123_WriteSDO / I87123_SDOWriteData.....	16
15.	I87123_SetPDOResponse / I87123_GetPDOLastData	17
16.	I87123_InstallPDO / I87123_InstallPDO / I87123_DynamicPDO	18
17.	I87123_RemovePDO / I87123_RemovePDO	19
18.	I87123_WritePDO / I87123_PDOWrite	20
19.	I87123_RemotePDO / I87123_PDORemote.....	21
20.	I87123_PDOTxType / I87123_PDOTxType	22
21.	I87123_MsgResponse.....	23
22.	I87123_WriteDO / I87123_ReadDI / I87123_WriteAO / I87123_ReadAI	24

1. I87123_GetVersion / I87123_GetVersion

[Old]:

```
float I87123_GetVersion(void);
```

[New]:

```
WORD I87123_GetVersion(void);
```

Examples

[Old]:

```
float Ver;
```

```
Ver = I87123_GetVersion();
```

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

[New]:

```
WORD Ver;
```

```
Ver = I87123_GetVersion();
```

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

2. I87123_Configure / I87123_InitMaster

[Old]:

```
int I87123_Configure(unsigned char baudrate);
```

[New]:

```
WORD I87123_InitMaster(BYTE SlotNo, HANDLE Port_0, BYTE  
Node, BYTE BaudRate, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;
```

```
ChangeSlotTo87K(0); // Change slot to I-87123  
ret = I87123_Configure(7);
```

[New]:

```
HANDLE hPort;
```

```
WORD ret;
```

```
hPort = uart_Open("COM0:,115200,N,8,1");
```

```
ret = I87123_InitMaster(0, hPort, 0, 7, 1);
```

```
/* Initialize I-87123 in slot 0 with master ID 0 and CAN baud 7  
(1 M bps). The mode of the API is 1 (block-mode). */
```

3. I87123_ShutdownMaster / I87123_ShutdownMaster

[Old]:

```
int I87123_ShutdownMaster(void);
```

[New]:

```
WORD I87123_ShutdownMaster(BYTE SlotNo);
```

Examples

[Old]:

```
int ret;
```

```
ChangeSlotTo87K(0); // Change slot to I-87123  
ret = I87123_ShutdownMaster();
```

[New]:

```
WORD ret;
```

```
ret = I87123_ShutdownMaster(0);
```

4. I87123_AddNode / I87123_AddNode

[Old]:

```
int I87123_AddNode(unsigned char node);
```

[New]:

```
WORD I87123_AddNode(BYTE SlotNo, BYTE Node,  
                    BYTE AddMode, WORD DelayTime,  
                    WORD ResTimeout, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;
```

```
ChangeSlotTo87K(0); // Change slot to I-87123  
ret = I87123_AddNode(1); // Add node 1 slave
```

[New]:

```
WORD ret;
```

```
ret = I87123_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). */
```

5. I87123_RemoveNode / I87123_RemoveNode

[Old]:

```
int I87123_RemoveNode(unsigned char node);
```

[New]:

```
WORD I87123_RemoveNode(BYTE SlotNo, BYTE Node,  
                        BYTE BlockMode);
```

Examples

[Old]:

```
int ret;
```

```
ChangeSlotTo87K(0); // Change slot to I-87123  
ret = I87123_RemoveNode(1); // Remove node 1 slave
```

[New]:

```
WORD ret;
```

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

6. I87123_ChangeState / I87123_NMTChangeState

[Old]:

```
int I87123_ChangeState(unsigned char node,  
                      unsigned char state);
```

[New]:

```
WORD I87123_NMTChangeState(BYTE SlotNo, BYTE Node,  
                           BYTE State, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ChangeSlotTo87K(0); // Change slot to I-87123  
ret = I87123_ChangeState(1, 0x01);
```

[New]:

```
WORD ret;  
  
ret = I87123_NMTChangeState(0, 1, 0x01, 1);
```


7. I87123_GetState / I87123_NMTGetState

[Old]:

```
int I87123_GetState(unsigned char node,  
                   unsigned char* state);
```

[New]:

```
WORD I87123_NMTGetState(BYTE SlotNo, BYTE Node,  
                        BYTE* State, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
unsigned char state;  
  
ChangeSlotTo87K(0); // Change slot to I-87123  
ret =I87123_GetState(1, &state);
```

[New]:

```
WORD ret;  
BYTE state;  
  
ret = I87123_NMTGetState(0, 1, &state, 1);
```

8. I87123_Guarding / I87123_NMTGuarding

[Old]:

```
int I87123_Guarding(unsigned char node,  
                   unsigned short guardtime, unsigned char lifftime);
```

[New]:

```
WORD I87123_NMTGuarding(BYTE SlotNo, BYTE Node,  
                        WORD GuardTime, BYTE LiftTime, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ChangeSlotTo87K(0); // Change slot to I-87123  
ret = I87123_Guarding(1, 1000, 3);
```

[New]:

```
WORD ret;  
  
ret = I87123_NMTGuarding(0, 1, 1000, 3, 1);
```

9. I87123_SendSYNC / I87123_SendSYNCMsg

[Old]:

```
int I87123_SendSYNC(unsigned short cobid,  
                   unsigned char cyclically, unsigned short timer);
```

[New]:

```
WORD I87123_SendSYNCMsg(BYTE SlotNo, WORD Cobid,  
                        WORD Timer, DWORD Times, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
unsigned char cyclic = 1;  
  
ChangeSlotTo87K(0); // Change slot to I-87123  
  
if(cyclic) // Send SYNC message per 1000 ms  
    ret = I87123_SendSYNC(0x80, 1, 1000);  
else // Send SYNC message once  
    ret = I87123_SendSYNC(0x80, 0, 1);  
  
/* Set parameter cyclically to 0 (0: send sync once, 1:  
send sync continuously) */
```

[New]:

```
WORD ret;  
BYTE cyclic = 1;  
  
if(cyclic) // Send SYNC message per 1000 ms  
    ret = I87123_SendSYNCMsg(0, 0x80, 1000, 0, 1);  
else // Send SYNC message once  
    ret = I87123_SendSYNCMsg(0, 0x80, 1, 1, 1);  
  
/* Set parameter times to 1 time (0: send sync  
continuously, others: send sync by the specific times) */
```

10. I87123_ChaneSYNCID / I87123_ChangeSYNCID

[Old]:

```
int I87123_ChaneSYNCID(unsigned char node,  
                      unsigned short cobid);
```

[New]:

```
WORD I87123_ChangeSYNCID(BYTE SlotNo, BYTE Node,  
                        WORD Cobid, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ChangeSlotTo87K(0); // Change slot to I-87123  
  
ret =I87123_ChaneSYNCID(1, 0x90);
```

[New]:

```
WORD ret;  
  
ret = I87123_ChangeSYNCID(0, 1, 0x90, 1);
```

11. I87123_ChaneEMCYID / I87123_ChangeEMCYID

[Old]:

```
int I87123_ChaneEMCYID(unsigned char node,  
                      unsigned short cobid);
```

[New]:

```
WORD I87123_ChangeEMCYID(BYTE SlotNo, BYTE Node,  
                          WORD Cobid, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ChangeSlotTo87K(0); // Change slot to I-87123  
  
ret =I87123_ChaneEMCYID(1, 0x91);
```

[New]:

```
WORD ret;  
  
ret = I87123_ChangeEMCYID(0, 1, 0x91, 1);
```

12. I87123_AbortSDO / I87123_SDOAbortTransmit

[Old]:

```
int I87123_AbortSDO(unsigned char node, unsigned short index,  
                  unsigned char subindex);
```

[New]:

```
WORD I87123_SDOAbortTransmit(BYTE SlotNo, BYTE Node,  
                             WORD Index, BYTE SubIndex,  
                             DWORD TData, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ChangeSlotTo87K(0); // Change slot to I-87123  
  
ret = I87123_AbortSDO(1, 0x1008, 0x0);
```

[New]:

```
WORD ret;  
  
ret = I87123_SDOAbortTransmit(0, 1, 0x1008, 0x0,  
                              0x00000000,1);  
// 0x00000000 is an example of the abort data
```

13. I87123_ReadSDO / I87123_SDOReadData

[Old]:

```
int I87123_ReadSDO(unsigned char node, unsigned short index,  
                 unsigned char subindex, unsigned char* len,  
                 unsigned char* rdata);
```

[New]:

```
WORD I87123_SDOReadData(BYTE SlotNo, BYTE Node,  
                       WORD Index, BYTE SubIndex, WORD* RDLen,  
                       BYTE* RData, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
unsigned char len;  
unsigned char rdata[256];  
  
ChangeSlotTo87K(0); // Change slot to I-87123  
  
ret = I87123_ReadSDO(1, 0x1000, 0x0, &len, rdata);
```

[New]:

```
WORD ret;  
WORD len;  
BYTE rdata [256];  
  
ret = I87123_SDOReadData(0, 1, 0x1000, 0x0, &len, rdata ,1);
```

14. I87123_WriteSDO / I87123_SDOWriteData

[Old]:

```
int I87123_WriteSDO(unsigned char node, unsigned short index,
                   unsigned char subindex, unsigned char len,
                   unsigned char* tdata, unsigned char* rlen,
                   unsigned char* rdata);
```

[New]:

```
WORD I87123_SDOWriteData(BYTE SlotNo, BYTE Node,
                          WORD Index, BYTE SubIndex, WORD TDLen,
                          BYTE *TData, WORD *RDLen, BYTE *RData,
                          BYTE BlockMode);
```

Examples

[Old]:

```
int ret;
unsigned char rlen;
unsigned char rdata [256];
unsigned char tdata[4] = {0xFF, 0, 0, 0};
```

```
ChangeSlotTo87K(0); // Change slot to I-87123
```

```
ret =I87123_WriteSDO(1, 0x6200, 0x1, 1, tdata, &rlen, rdata);
```

[New]:

```
WORD ret;
WORD rlen;
BYTE rdata [256];
BYTE tdata[4] = {0xFF, 0, 0, 0};
```

```
ret = I87123_SDOWriteData(0, 1, 0x6200, 0x1, 1, tdata,
                           &rlen, rdata ,1);
```


15. I87123_SetPDOResponse / I87123_GetPDOLastData

[Old]:

```
int I87123_SetPDOResponse(unsigned char node,  
                          unsigned short cobid, unsigned char mode);
```

[New]:

```
WORD I87123_GetPDOLastData(BYTE SlotNo, WORD Cobid,  
                           BYTE *IsNew, BYTE *DLen,  
                           BYTE *RData, BYTE BlockMode);
```

[Old]:

```
int ret;  
unsigned char node, len;  
unsigned char rdata[8];  
unsigned short cobid;  
  
ChangeSlotTo87K(0); // Change slot to I-87123  
  
ret = I87123_SetPDOResponse(1, 0x181, 1);  
.....  
ret = I87123_MsgResponse(&node, &cobid, &len, rdata);  
if(cobid == 0x181){  
.....  
}
```

[New]:

```
WORD ret;  
BYTE new, len;  
BYTE rdata[8];  
  
ret = I87123_GetPDOLastData(0, 0x181, &new, &len, rdata, 1);  
if(new == 1){ // Handle the new obtained message  
.....  
}
```

Note: If the mode parameter is 0 in the old function, there will do nothing in the new library.

16. I87123_InstallPDO / I87123_InstallPDO / I87123_DynamicPDO

[Old]:

```
int I87123_InstallPDO(unsigned char node, unsigned short cobid,  
                    unsigned char txrxtype, unsigned char channel,  
                    unsigned char* tdata);
```

[New]:

```
WORD I87123_InstallPDO(BYTE SlotNo, BYTE Node,  
                      WORD Cobid, BYTE RxTx,  
                      WORD PDO_No, BYTE BlockMode);
```

```
WORD I87123_DynamicPDO(BYTE SlotNo, BYTE Node,  
                      WORD Cobid, BYTE RxTx, BYTE Entry,  
                      DWORD EntryData, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
unsigned char map_tdata[4] = {0x8 0x1, 0x0, 0x062};  
  
ChangeSlotTo87K(0); // Change slot to I-87123  
// Create a RxPDO with 0x333 ID  
// and set map_data to entry 1 of 0x333 ID  
ret = I87123_InstallPDO(1, 0x333, 0, 1, map_tdata);
```

[New]:

```
WORD ret;  
DWORD map_tdata = 0x62000108;  
  
// Create a RxPDO5 with 0x333 ID  
ret = I87123_InstallPDO(0, 1, 0x333, 0, 5, 1);  
// Set map_data to entry 1 of 0x333 ID  
ret = I87123_DynamicPDO(0, 1, 0x333, 0, 1, map_tdata, 1);
```

17. I87123_RemovePDO / I87123_RemovePDO

[Old]:

```
int I87123_RemovePDO(unsigned char node,  
                    unsigned short cobid, unsigned char txrxtype,  
                    unsigned char channel);
```

[New]:

```
WORD I87123_RemovePDO(BYTE SlotNo, BYTE Node,  
                     WORD Cobid, BYTE Entry, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ChangeSlotTo87K(0); // Change slot to I-87123  
  
ret = I87123_RemovePDO(1, 0x201, 0, 1);  
  
// 0 is RxPDO type
```

[New]:

```
WORD ret;  
  
ret = I87123_RemovePDO(0, 1, 0x201, 1, 1);
```

18. I87123_WritePDO / I87123_PDOWrite

[Old]:

```
int I87123_WritePDO(unsigned short cobid, unsigned char offset,  
                  unsigned char dlen, unsigned char *tdata);
```

[New]:

```
WORD I87123_PDOWrite(BYTE SlotNo, WORD Cobid,  
                    BYTE Offset, BYTE DLen, BYTE *Data,  
                    BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
unsigned char tdata[2] = {0x55, 0xFF};  
  
ChangeSlotTo87K(0); // Change slot to I-87123  
  
ret = I87123_WritePDO(0x201, 0, 2, tdata);
```

[New]:

```
WORD ret;  
BYTE tdata[2] = {0x55, 0xFF};  
  
ret = I87123_PDOWrite(0, 0x201, 0, 2, tdata, 1);
```

19. I87123_RemotePDO / I87123_PDORemote

[Old]:

```
int I87123_RemotePDO(unsigned short cobid, unsigned char *len,  
                    unsigned char *rdata);
```

[New]:

```
WORD I87123_PDORemote(BYTE SlotNo, WORD Cobid,  
                     BYTE *DLen, BYTE *RData,  
                     BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
unsigned char len;  
unsigned char rdata[8];
```

```
ChangeSlotTo87K(0); // Change slot to I-87123
```

```
ret = I87123_RemotePDO(0x181, &len, rdata);
```

[New]:

```
WORD ret;  
BYTE len;  
BYTE rdata[8];
```

```
ret = I87123_PDORemote(0, 0x181, &len, rdata, 1);
```

20. I87123_PDOTxType / I87123_PDOTxType

[Old]:

```
int I87123_PDOTxType(unsigned char node,  
                    unsigned short cobid, unsigned char txtype);
```

[New]:

```
WORD I87123_PDOTxType(BYTE SlotNo, WORD Cobid,  
                    BYTE Tx_Type, BYTE BlockMode);
```

Examples

[Old]:

```
int ret;  
  
ChangeSlotTo87K(0); // Change slot to I-87123  
  
ret =I87123_PDOTxType(1, 0x181, 50);
```

[New]:

```
WORD ret;  
  
ret = I87123_PDOTxType(0, 0x181, 50, 1);
```

21. I87123_MsgResponse

[Old]:

```
int I87123_MsgResponse(unsigned char* node,  
                      unsigned short* cobid, unsigned char* len,  
                      unsigned char* rdata);
```

[New]:

None.

22. 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]:

Use **I87123_SDOReadData** and **I87123_SDOWriteData** instead above old functions.