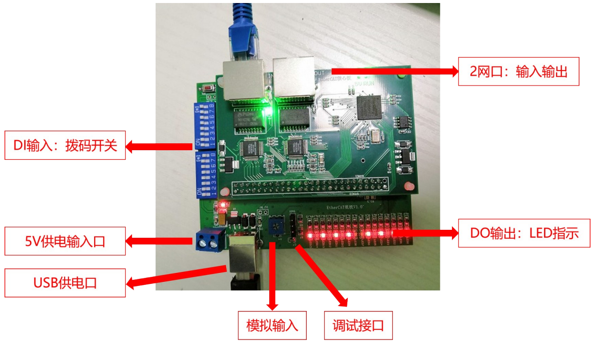
原创 廷华电子 于 2020-03-21 23:45:17 发布 3841 收藏 37 版权
分类专栏： EtherCAT

EtherCAT 专栏收录该内容 88 订阅 14 篇文章 订阅专栏

目的：在EtherCAT开发板上IO程序（基于SSC的IO通讯）基础上进行修改，实现16路拨码开关输入，以及16路LED输出。



EtherCAT从站增加变量的操作包括两个部分：1）XML文件修改；2）底板STM32程序修改。

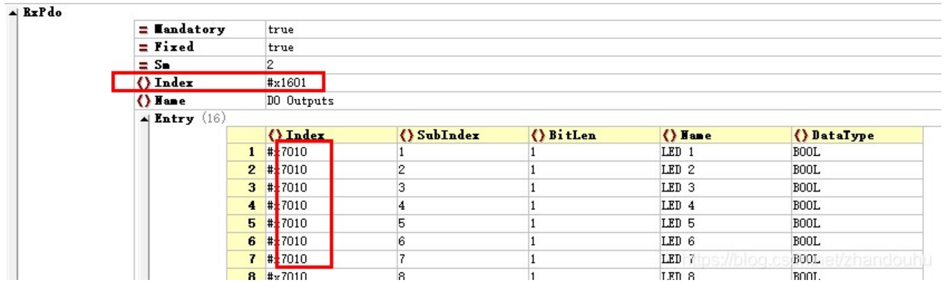
1、XML配置修改

xml作为EtherCAT从站设备描述文件，烧写在EEPROM里，ESC芯片上电后，会从EEPROM加载配置参数，完成启动初始化过程。同时主站设备扫描从站时，会读取EEPORM描述信息，完成对从站的初始配置。XML 里定义的对象字典和PDO 设置，需要和对应底板程序里的对象字典相对应。

XML的基本介绍 参考【EtherCAT分析】三、EtherCAT从站设备描述文件设计，下面介绍根据实验目的进行xml文件修改的过程。

1.1 更改数据类型

原xml文件中，LED输出映射索引0x1601和0x7010，Switch输出映射索引0x1a00和0x6000，这里在这个基础上修改。

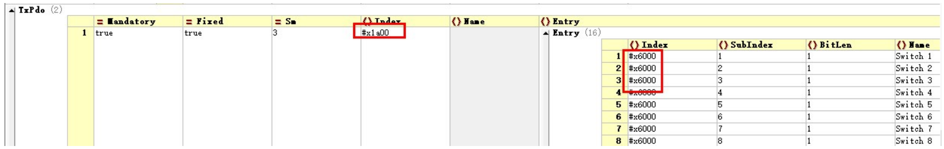


RxPdo

Mandatory	true
Fixed	true
Sm	2
Index	#x1601
Name	DO Outputs

Entry (16)

	Index	SubIndex	BitLen	Name	DataType
1	#x7010	1	1	LED 1	BOOL
2	#x7010	2	1	LED 2	BOOL
3	#x7010	3	1	LED 3	BOOL
4	#x7010	4	1	LED 4	BOOL
5	#x7010	5	1	LED 5	BOOL
6	#x7010	6	1	LED 6	BOOL
7	#x7010	7	1	LED 7	BOOL
8	#x7010	8	1	LED 8	BOOL

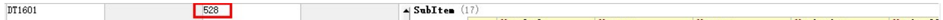


TxPdo (2)

	Mandatory	Fixed	Sm	Index	Name	Entry
1	true	true	3	#x1a00		Entry (16)

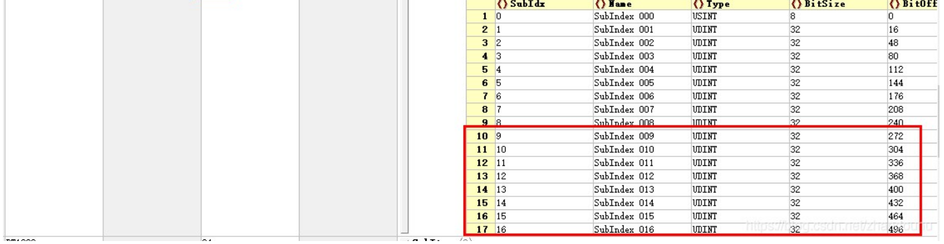
	Index	SubIndex	BitLen	Name
1	#x6000	1	1	Switch 1
2	#x6000	2	1	Switch 2
3	#x6000	3	1	Switch 3
4	#x6000	4	1	Switch 4
5	#x6000	5	1	Switch 5
6	#x6000	6	1	Switch 6
7	#x6000	7	1	Switch 7
8	#x6000	8	1	Switch 8

（1）更改DT1601数据类型，重新修改数据类型的BitSize。



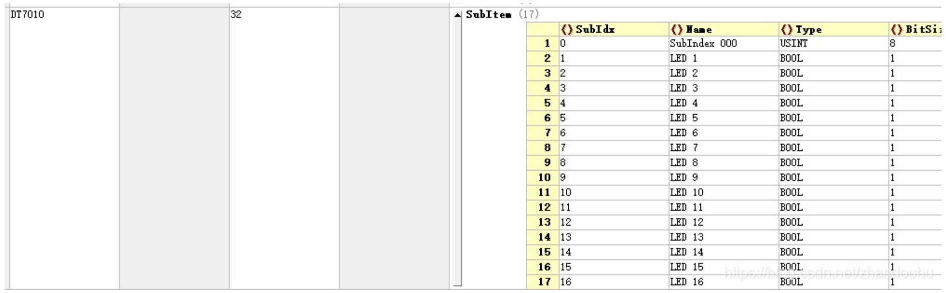
DT1601 528 SubItem (17)

分类专栏
EtherCAT 14篇
电机驱动 3篇
pixhawk/px4 2篇
嵌入式设计 1篇



	SubIdx	Name	Type	BitSize	BitOff
1	0	SubIndex 000	USINT	8	0
2	1	SubIndex 001	UDINT	32	16
3	2	SubIndex 002	UDINT	32	48
4	3	SubIndex 003	UDINT	32	80
5	4	SubIndex 004	UDINT	32	112
6	5	SubIndex 005	UDINT	32	144
7	6	SubIndex 006	UDINT	32	176
8	7	SubIndex 007	UDINT	32	208
9	8	SubIndex 008	UDINT	32	240
10	9	SubIndex 009	UDINT	32	272
11	10	SubIndex 010	UDINT	32	304
12	11	SubIndex 011	UDINT	32	336
13	12	SubIndex 012	UDINT	32	368
14	13	SubIndex 013	UDINT	32	400
15	14	SubIndex 014	UDINT	32	432
16	15	SubIndex 015	UDINT	32	464
17	16	SubIndex 016	UDINT	32	496

（2）索引0x1601将输出过程数据映射到0X7010中，因此需要更改DT7010数据类型。

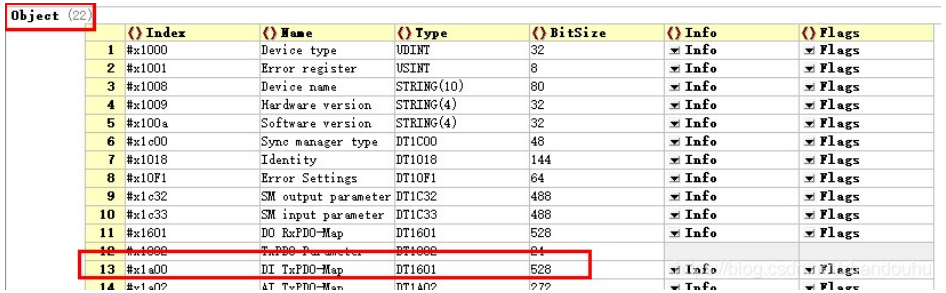


DT7010 32

SubItem (17)

	SubIdx	Name	Type	BitSi
1	0	SubIndex 000	USINT	8
2	1	LED 1	BOOL	1
3	2	LED 2	BOOL	1
4	3	LED 3	BOOL	1
5	4	LED 4	BOOL	1
6	5	LED 5	BOOL	1
7	6	LED 6	BOOL	1
8	7	LED 7	BOOL	1
9	8	LED 8	BOOL	1
10	9	LED 9	BOOL	1
11	10	LED 10	BOOL	1
12	11	LED 11	BOOL	1
13	12	LED 12	BOOL	1
14	13	LED 13	BOOL	1
15	14	LED 14	BOOL	1
16	15	LED 15	BOOL	1
17	16	LED 16	BOOL	1

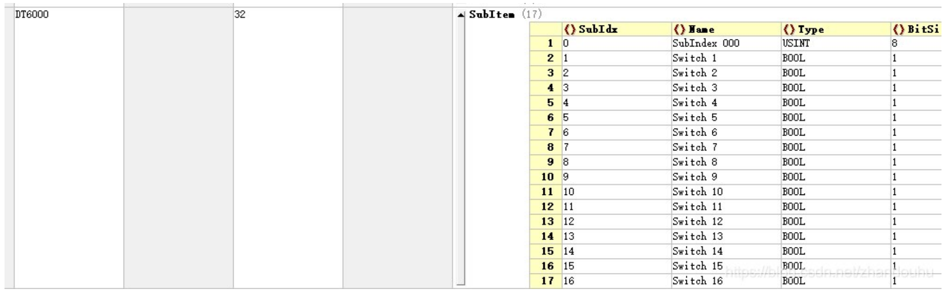
（3）发现数据类型无DT1A00，从后面对象字典可以知0x1A00用得也是DT1601数据，考虑到修改为16路LED输出和16路Switch输入时，两个数据格式相同，因此无需添加。



Object (22)

	Index	Name	Type	BitSize	Info	Flags
1	#x1000	Device type	UDINT	32	Info	Flags
2	#x1001	Error register	USINT	8	Info	Flags
3	#x1008	Device name	STRING(10)	80	Info	Flags
4	#x1009	Hardware version	STRING(4)	32	Info	Flags
5	#x100a	Software version	STRING(4)	32	Info	Flags
6	#x1c00	Sync manager type	DT1C00	48	Info	Flags
7	#x1018	Identity	DT1018	144	Info	Flags
8	#x10F1	Error Settings	DT10F1	64	Info	Flags
9	#x1c32	SM output parameter	DT1C32	488	Info	Flags
10	#x1c33	SM input parameter	DT1C33	488	Info	Flags
11	#x1601	DO RxPDO-Map	DT1601	528	Info	Flags
12	#x1802	TxPDO Parameter	DT1802	24		
13	#x1a00	DI TxPDO-Map	DT1601	528	Info	Flags
14	#x1a02	AI TxPDO-Map	DT1A02	272	Info	Flags

索引0x1a00将输出过程数据映射到0X6000中，因此需要更改DT6000数据类型。



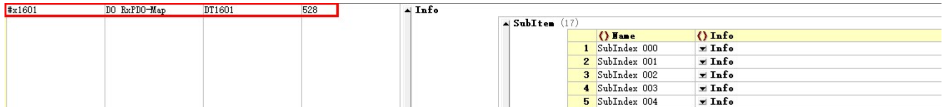
DT6000 32

SubItem (17)

	SubIdx	Name	Type	BitSi
1	0	SubIndex 000	USINT	8
2	1	Switch 1	BOOL	1
3	2	Switch 2	BOOL	1
4	3	Switch 3	BOOL	1
5	4	Switch 4	BOOL	1
6	5	Switch 5	BOOL	1
7	6	Switch 6	BOOL	1
8	7	Switch 7	BOOL	1
9	8	Switch 8	BOOL	1
10	9	Switch 9	BOOL	1
11	10	Switch 10	BOOL	1
12	11	Switch 11	BOOL	1
13	12	Switch 12	BOOL	1
14	13	Switch 13	BOOL	1
15	14	Switch 14	BOOL	1
16	15	Switch 15	BOOL	1
17	16	Switch 16	BOOL	1

1.2更改数据字典对象表

（1)更改0x1601数据对象字典--输出映射表AI TxPDO-Map

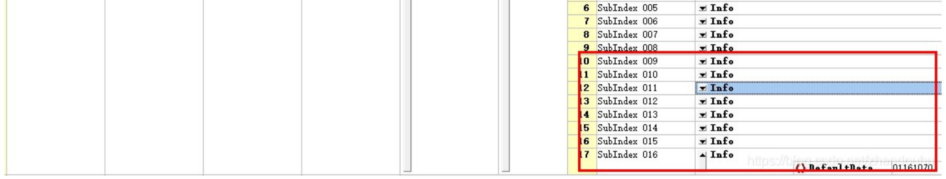


#x1601 DO RxPDO-Map DT1601 528

Info

SubItem (17)

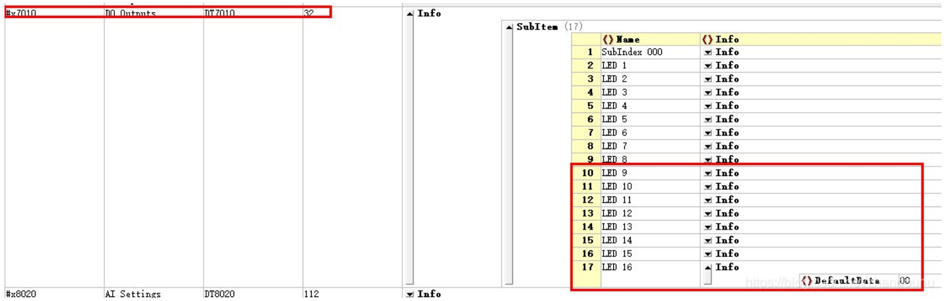
	Name	Info
1	SubIndex 000	Info
2	SubIndex 001	Info
3	SubIndex 002	Info
4	SubIndex 003	Info
5	SubIndex 004	Info



	Name	Info
6	SubIndex 005	Info
7	SubIndex 006	Info
8	SubIndex 007	Info
9	SubIndex 008	Info
10	SubIndex 009	Info
11	SubIndex 010	Info
12	SubIndex 011	Info
13	SubIndex 012	Info
14	SubIndex 013	Info
15	SubIndex 014	Info
16	SubIndex 015	Info
17	SubIndex 016	Info

DefaultData 01161070

(2) 更改0X7010对象字典



#x7010	DO Outputs	DT7010	32

Info

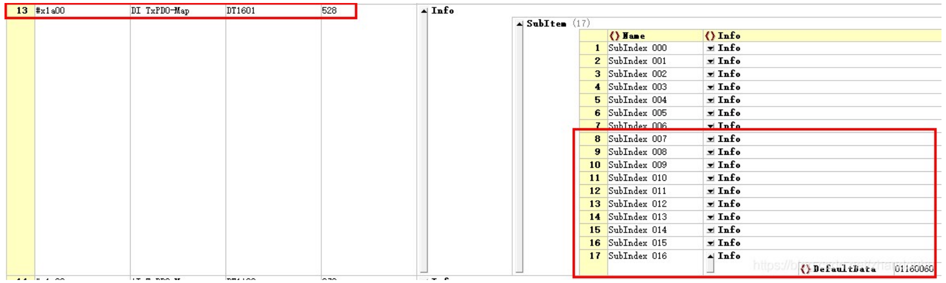
SubItem (17)

	Name	Info
1	SubIndex 000	Info
2	LED 1	Info
3	LED 2	Info
4	LED 3	Info
5	LED 4	Info
6	LED 5	Info
7	LED 6	Info
8	LED 7	Info
9	LED 8	Info
10	LED 9	Info
11	LED 10	Info
12	LED 11	Info
13	LED 12	Info
14	LED 13	Info
15	LED 14	Info
16	LED 15	Info
17	LED 16	Info

DefaultData 00

#x8020	AI Settings	DT8020	112	Info

（3）更改0X1a00对象字典



13	#x1a00	DI TxPDO-Map	DT1601	528

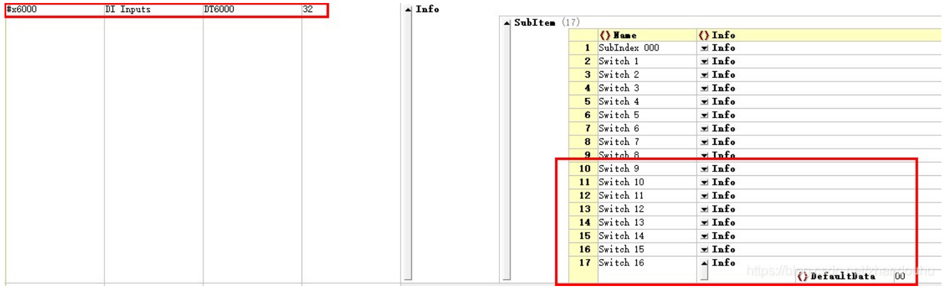
Info

SubItem (17)

	Name	Info
1	SubIndex 000	Info
2	SubIndex 001	Info
3	SubIndex 002	Info
4	SubIndex 003	Info
5	SubIndex 004	Info
6	SubIndex 005	Info
7	SubIndex 006	Info
8	SubIndex 007	Info
9	SubIndex 008	Info
10	SubIndex 009	Info
11	SubIndex 010	Info
12	SubIndex 011	Info
13	SubIndex 012	Info
14	SubIndex 013	Info
15	SubIndex 014	Info
16	SubIndex 015	Info
17	SubIndex 016	Info

DefaultData 01160060

(4) 更改0X6000对象字典



#x6000	DI Inputs	DT6000	32

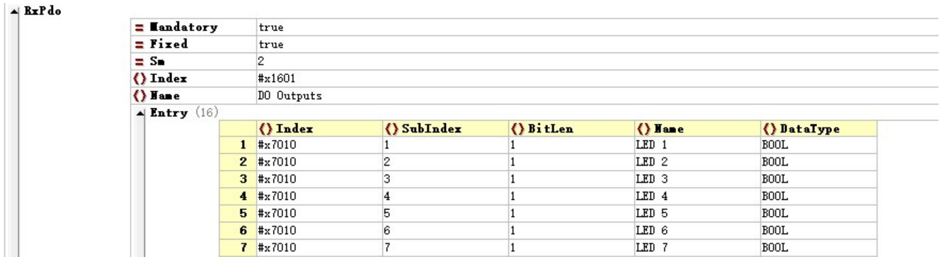
Info

SubItem (17)

	Name	Info
1	SubIndex 000	Info
2	Switch 1	Info
3	Switch 2	Info
4	Switch 3	Info
5	Switch 4	Info
6	Switch 5	Info
7	Switch 6	Info
8	Switch 7	Info
9	Switch 8	Info
10	Switch 9	Info
11	Switch 10	Info
12	Switch 11	Info
13	Switch 12	Info
14	Switch 13	Info
15	Switch 14	Info
16	Switch 15	Info
17	Switch 16	Info

DefaultData 00

1.3定义PDO传输

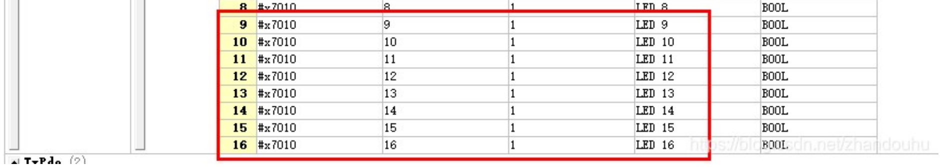


RxPdo

Mandatory	true
Fixed	true
Sm	2
Index	#x1601
Name	DO Outputs

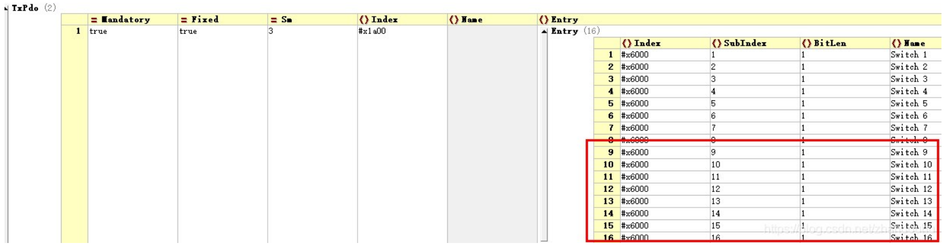
Entry (16)

	Index	SubIndex	BitLen	Name	DataType
1	#x7010	1	1	LED 1	BOOL
2	#x7010	2	1	LED 2	BOOL
3	#x7010	3	1	LED 3	BOOL
4	#x7010	4	1	LED 4	BOOL
5	#x7010	5	1	LED 5	BOOL
6	#x7010	6	1	LED 6	BOOL
7	#x7010	7	1	LED 7	BOOL



8	#x7010	8	1	LED 8	BOOL
9	#x7010	9	1	LED 9	BOOL
10	#x7010	10	1	LED 10	BOOL
11	#x7010	11	1	LED 11	BOOL
12	#x7010	12	1	LED 12	BOOL
13	#x7010	13	1	LED 13	BOOL
14	#x7010	14	1	LED 14	BOOL
15	#x7010	15	1	LED 15	BOOL
16	#x7010	16	1	LED 16	BOOL

TxPdo (2)

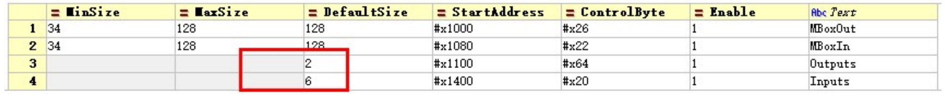


TxPdo (2)

	= Mandatory	= Fixed	= Sm	() Index	() Name	() Entry
1	true	true	3	#x1a00		Entry (16)

	() Index	() SubIndex	() BitLen	() Name
1	#x6000	1	1	Switch 1
2	#x6000	2	1	Switch 2
3	#x6000	3	1	Switch 3
4	#x6000	4	1	Switch 4
5	#x6000	5	1	Switch 5
6	#x6000	6	1	Switch 6
7	#x6000	7	1	Switch 7
8	#x6000	8	1	Switch 8
9	#x6000	9	1	Switch 9
10	#x6000	10	1	Switch 10
11	#x6000	11	1	Switch 11
12	#x6000	12	1	Switch 12
13	#x6000	13	1	Switch 13
14	#x6000	14	1	Switch 14
15	#x6000	15	1	Switch 15
16	#x6000	16	1	Switch 16

1.4修改SM配置



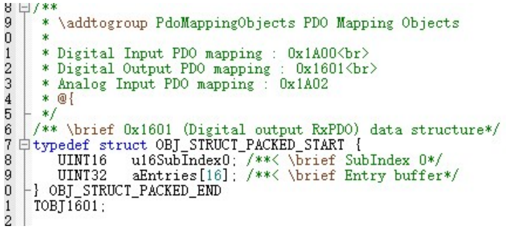
	= MinSize	= MaxSize	= DefaultSize	= StartAddress	= ControlByte	= Enable	Abc Text
1	34	128	128	#x1000	#x26	1	MBoxOut
2	34	128	128	#x1080	#x22	1	MBoxIn
3			2	#x1100	#x64	1	Outputs
4			6	#x1400	#x20	1	Inputs

2、STM32程序更改

在el9800appl.h中修改对象描述，包括更改本地存储变量、表项描述、对象名字以及对象描述。

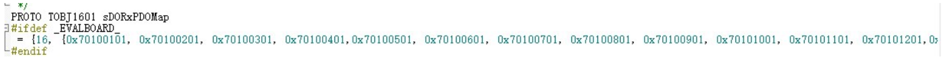
2.1更改本地存储变量

（1）更改0x1601本地存储变量



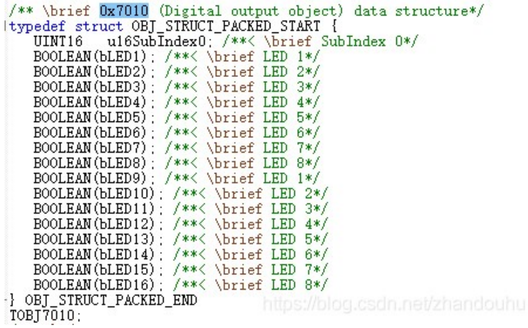
```
/**
 * \addtogroup PdoMappingObjects PDO Mapping Objects
 *
 * Digital Input PDO mapping : 0x1A00<br>
 * Digital Output PDO mapping : 0x1601<br>
 * Analog Input PDO mapping : 0x1A02
 * @{
 */
/** \brief 0x1601 (Digital output RxPDO) data structure*/
typedef struct OBJ_STRUCT_PACKED_START {
   UINT16   u16SubIndex0; /**< \brief SubIndex 0*/
   UINT32   aEntries[16]; /**< \brief Entry buffer*/
} OBJ_STRUCT_PACKED_END
TOBJ1601;
```

0x70100x01与xml文件对应，其中索引也要改为16。



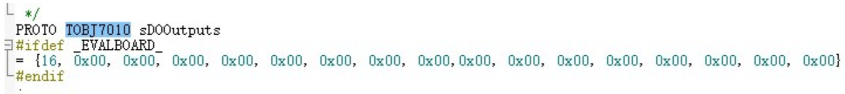
```
 */
PROTO TOBJ1601 sDORxPDOMap
#ifdef _EVALBOARD_
 = {16, {0x70100101, 0x70100201, 0x70100301, 0x70100401,0x70100501, 0x70100601, 0x70100701, 0x70100801, 0x70100901, 0x70101001, 0x70101101, 0x70101201,0x
#endif
```

（2）更改0x7010本地存储变量



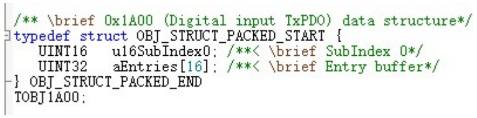
```
/** \brief 0x7010 (Digital output object) data structure*/
typedef struct OBJ_STRUCT_PACKED_START {
   UINT16   u16SubIndex0; /**< \brief SubIndex 0*/
   BOOLEAN(bLED1); /**< \brief LED 1*/
   BOOLEAN(bLED2); /**< \brief LED 2*/
   BOOLEAN(bLED3); /**< \brief LED 3*/
   BOOLEAN(bLED4); /**< \brief LED 4*/
   BOOLEAN(bLED5); /**< \brief LED 5*/
   BOOLEAN(bLED6); /**< \brief LED 6*/
   BOOLEAN(bLED7); /**< \brief LED 7*/
   BOOLEAN(bLED8); /**< \brief LED 8*/
   BOOLEAN(bLED9); /**< \brief LED 1*/
   BOOLEAN(bLED10); /**< \brief LED 2*/
   BOOLEAN(bLED11); /**< \brief LED 3*/
   BOOLEAN(bLED12); /**< \brief LED 4*/
   BOOLEAN(bLED13); /**< \brief LED 5*/
   BOOLEAN(bLED14); /**< \brief LED 6*/
   BOOLEAN(bLED15); /**< \brief LED 7*/
   BOOLEAN(bLED16); /**< \brief LED 8*/
} OBJ_STRUCT_PACKED_END
TOBJ7010;
```

索引改为16：

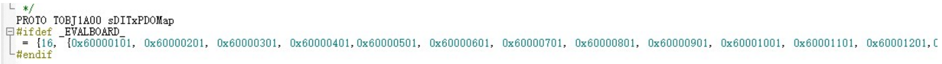


```
 */
PROTO TOBJ7010 sDOOutputs
#ifdef _EVALBOARD_
= {16, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}
#endif
;
```

（3）更改0x1a00本地存储变量

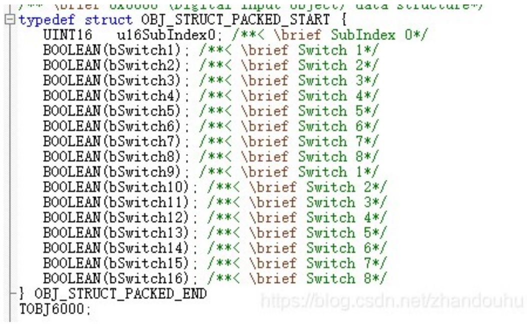


```
/** \brief 0x1A00 (Digital input TxPDO) data structure*/
typedef struct OBJ_STRUCT_PACKED_START {
   UINT16   u16SubIndex0; /**< \brief SubIndex 0*/
   UINT32   aEntries[16]; /**< \brief Entry buffer*/
} OBJ_STRUCT_PACKED_END
TOBJ1A00;
```

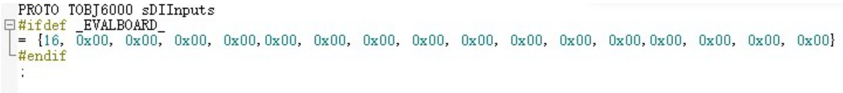


```
 */
PROTO TOBJ1A00 sDITxPDOMap
#ifdef _EVALBOARD_
 = {16, {0x60000101, 0x60000201, 0x60000301, 0x60000401,0x60000501, 0x60000601, 0x60000701, 0x60000801, 0x60000901, 0x60001001, 0x60001101, 0x60001201,C
#endif
;
```

（4）更改0x6000本地存储变量



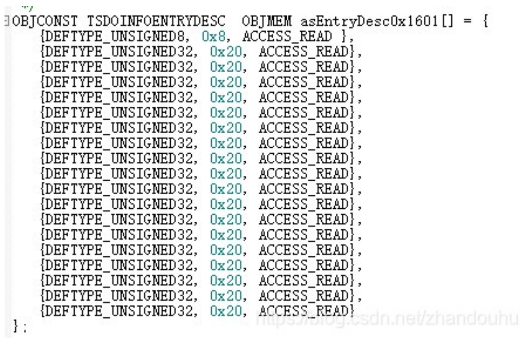
```
/** \brief 0x6000 (Digital input object) data structure*/
typedef struct OBJ_STRUCT_PACKED_START {
   UINT16   u16SubIndex0; /**< \brief SubIndex 0*/
   BOOLEAN(bSwitch1); /**< \brief Switch 1*/
   BOOLEAN(bSwitch2); /**< \brief Switch 2*/
   BOOLEAN(bSwitch3); /**< \brief Switch 3*/
   BOOLEAN(bSwitch4); /**< \brief Switch 4*/
   BOOLEAN(bSwitch5); /**< \brief Switch 5*/
   BOOLEAN(bSwitch6); /**< \brief Switch 6*/
   BOOLEAN(bSwitch7); /**< \brief Switch 7*/
   BOOLEAN(bSwitch8); /**< \brief Switch 8*/
   BOOLEAN(bSwitch9); /**< \brief Switch 1*/
   BOOLEAN(bSwitch10); /**< \brief Switch 2*/
   BOOLEAN(bSwitch11); /**< \brief Switch 3*/
   BOOLEAN(bSwitch12); /**< \brief Switch 4*/
   BOOLEAN(bSwitch13); /**< \brief Switch 5*/
   BOOLEAN(bSwitch14); /**< \brief Switch 6*/
   BOOLEAN(bSwitch15); /**< \brief Switch 7*/
   BOOLEAN(bSwitch16); /**< \brief Switch 8*/
} OBJ_STRUCT_PACKED_END
TOBJ6000;
```



```
PROTO TOBJ6000 sDIInputs
#ifdef _EVALBOARD_
= {16, 0x00, 0x00, 0x00, 0x00,0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,0x00, 0x00, 0x00, 0x00}
#endif
;
```

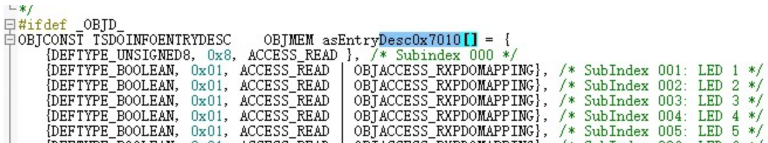
2.2更改表项描述

（1）更改0x1601表项描述

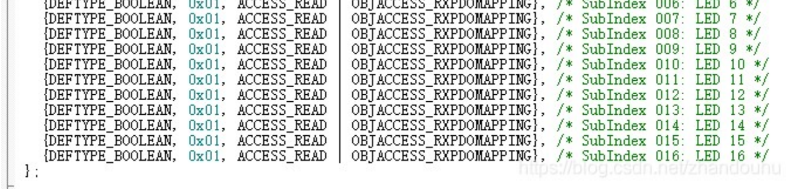


```
OBJCONST TSDOINFOENTRYDESC  OBJMEM asEntryDesc0x1601[] = {
   {DEFTYPE_UNSIGNED8, 0x8, ACCESS_READ },
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
   {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ}
};
```

（2）更改0x7010表项描述

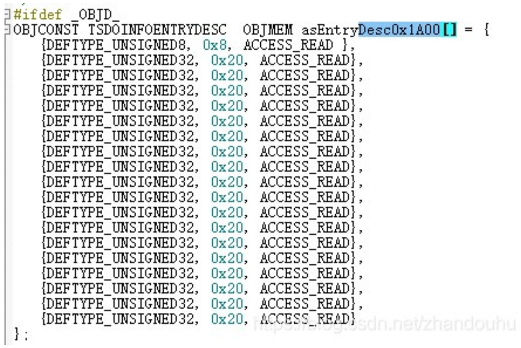


```
 */
#ifdef _OBJD_
OBJCONST TSDOINFOENTRYDESC    OBJMEM asEntryDesc0x7010[] = {
    {DEFTYPE_UNSIGNED8, 0x8, ACCESS_READ }, /* Subindex 000 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 001: LED 1 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 002: LED 2 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 003: LED 3 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 004: LED 4 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 005: LED 5 */
```



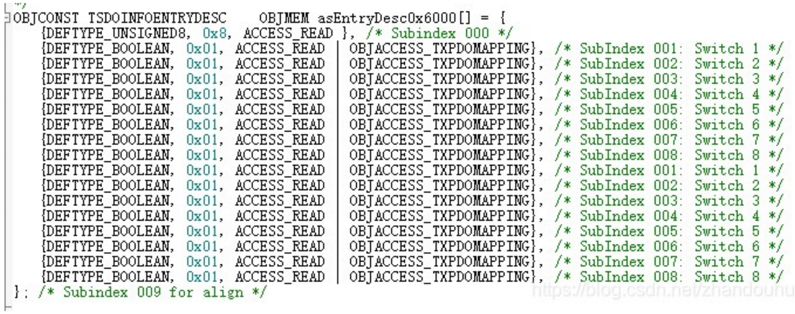
```
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 006: LED 6 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 007: LED 7 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 008: LED 8 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 009: LED 9 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 010: LED 10 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 011: LED 11 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 012: LED 12 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 013: LED 13 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 014: LED 14 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 015: LED 15 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_RXPDOMAPPING}, /* SubIndex 016: LED 16 */
 };
```

（3）更改0x1a00表项描述



```
#ifdef _OBJD_
OBJCONST TSDOINFOENTRYDESC  OBJMEM asEntryDesc0x1A00[] = {
    {DEFTYPE_UNSIGNED8, 0x8, ACCESS_READ },
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ},
    {DEFTYPE_UNSIGNED32, 0x20, ACCESS_READ}
};
```

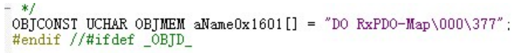
（4）更改0x6000表项描述



```
*/
OBJCONST TSDOINFOENTRYDESC    OBJMEM asEntryDesc0x6000[] = {
    {DEFTYPE_UNSIGNED8, 0x8, ACCESS_READ }, /* Subindex 000 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 001: Switch 1 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 002: Switch 2 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 003: Switch 3 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 004: Switch 4 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 005: Switch 5 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 006: Switch 6 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 007: Switch 7 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 008: Switch 8 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 001: Switch 1 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 002: Switch 2 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 003: Switch 3 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 004: Switch 4 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 005: Switch 5 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 006: Switch 6 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 007: Switch 7 */
    {DEFTYPE_BOOLEAN, 0x01, ACCESS_READ | OBJACCESS_TXPDOMAPPING}, /* SubIndex 008: Switch 8 */
}; /* Subindex 009 for align */
```

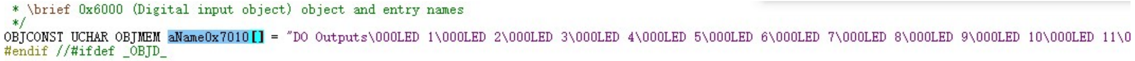
2.3更改对象名字

（1）0x1601对象名字无需更改



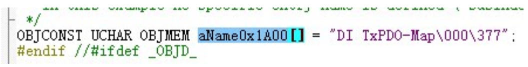
```
 */
OBJCONST UCHAR OBJMEM aName0x1601[] = "DO RxPDO-Map\000\377";
#endif //#ifdef _OBJD_
```

（2）更改0x7010对象名字



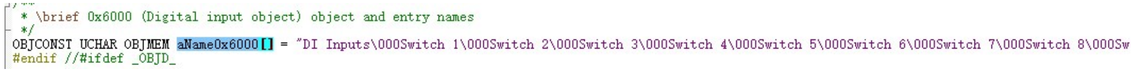
```
 * \brief 0x6000 (Digital input object) object and entry names
 */
OBJCONST UCHAR OBJMEM aName0x7010[] = "DO Outputs\000LED 1\000LED 2\000LED 3\000LED 4\000LED 5\000LED 6\000LED 7\000LED 8\000LED 9\000LED 10\000LED 11\0
#endif //#ifdef _OBJD_
```

（3）0x1a00对象名字无需更改



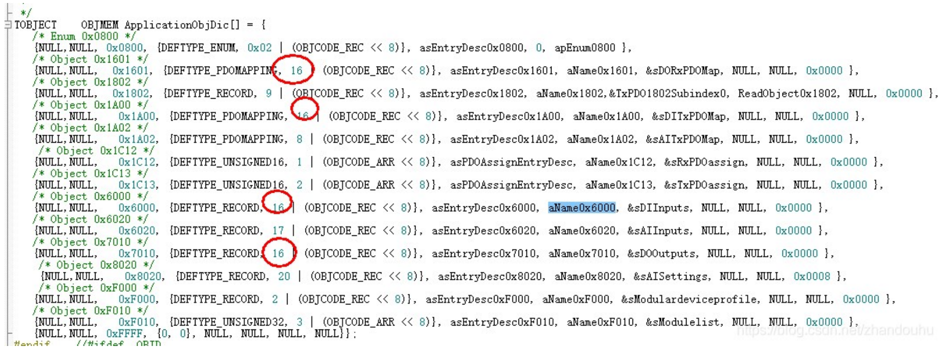
```
 */
OBJCONST UCHAR OBJMEM aName0x1A00[] = "DI TxPDO-Map\000\377";
#endif //#ifdef _OBJD_
```

（4）更改0x6000对象名字



```
/**
 * \brief 0x6000 (Digital input object) object and entry names
 */
OBJCONST UCHAR OBJMEM aName0x6000[] = "DI Inputs\000Switch 1\000Switch 2\000Switch 3\000Switch 4\000Switch 5\000Switch 6\000Switch 7\000Switch 8\000Sw
#endif //#ifdef _OBJD_
```

2.4更改对象描述

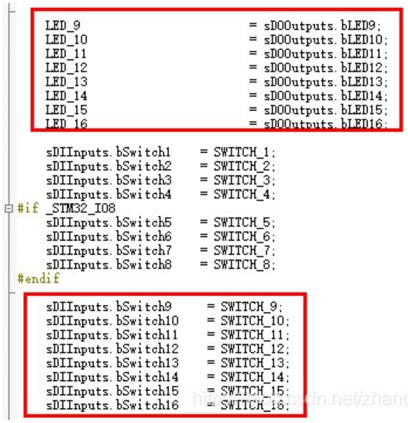


```
 */
TOBJECT    OBJMEM ApplicationObjDic[] = {
    /* Enum 0x0800 */
    {NULL,NULL, 0x0800, {DEFTYPE_ENUM, 0x02 | (OBJCODE_REC << 8)}, asEntryDesc0x0800, 0, apEnum0800 },
    /* Object 0x1601 */
    {NULL,NULL,  0x1601, {DEFTYPE_PDOMAPPING, 16 | (OBJCODE_REC << 8)}, asEntryDesc0x1601, aName0x1601, &sDORxPDOMap, NULL, NULL, 0x0000 },
    /* Object 0x1802 */
    {NULL,NULL,  0x1802, {DEFTYPE_RECORD, 9 | (OBJCODE_REC << 8)}, asEntryDesc0x1802, aName0x1802,&TxPDO1802Subindex0, ReadObject0x1802, NULL, 0x0000 },
    /* Object 0x1A00 */
    {NULL,NULL,   0x1A00, {DEFTYPE_PDOMAPPING, 16 | (OBJCODE_REC << 8)}, asEntryDesc0x1A00, aName0x1A00, &sDITxPDOMap, NULL, NULL, 0x0000 },
    /* Object 0x1A02 */
    {NULL,NULL,   0x1A02, {DEFTYPE_PDOMAPPING, 8 | (OBJCODE_REC << 8)}, asEntryDesc0x1A02, aName0x1A02, &sAITxPDOMap, NULL, NULL, 0x0000 },
     /* Object 0x1C12 */
    {NULL,NULL,   0x1C12, {DEFTYPE_UNSIGNED16, 1 | (OBJCODE_ARR << 8)}, asPDOAssignEntryDesc, aName0x1C12, &sRxPDOassign, NULL, NULL, 0x0000 },
    /* Object 0x1C13 */
    {NULL,NULL,   0x1C13, {DEFTYPE_UNSIGNED16, 2 | (OBJCODE_ARR << 8)}, asPDOAssignEntryDesc, aName0x1C13, &sTxPDOassign, NULL, NULL, 0x0000 },
    /* Object 0x6000 */
    {NULL,NULL,   0x6000, {DEFTYPE_RECORD, 16 | (OBJCODE_REC << 8)}, asEntryDesc0x6000, aName0x6000, &sDIInputs, NULL, NULL, 0x0000 },
    /* Object 0x6020 */
    {NULL,NULL,   0x6020, {DEFTYPE_RECORD, 17 | (OBJCODE_REC << 8)}, asEntryDesc0x6020, aName0x6020, &sAIInputs, NULL, NULL, 0x0000 },
    /* Object 0x7010 */
    {NULL,NULL,   0x7010, {DEFTYPE_RECORD, 16 | (OBJCODE_REC << 8)}, asEntryDesc0x7010, aName0x7010, &sDOOutputs, NULL, NULL, 0x0000 },
     /* Object 0x8020 */
     {NULL,NULL,   0x8020, {DEFTYPE_RECORD, 20 | (OBJCODE_REC << 8)}, asEntryDesc0x8020, aName0x8020, &sAISettings, NULL, NULL, 0x0008 },
     /* Object 0xF000 */
    {NULL,NULL,   0xF000, {DEFTYPE_RECORD, 2 | (OBJCODE_REC << 8)}, asEntryDesc0xF000, aName0xF000, &sModulardeviceprofile, NULL, NULL, 0x0000 },
    /* Object 0xF010 */
    {NULL,NULL,   0xF010, {DEFTYPE_UNSIGNED32, 3 | (OBJCODE_ARR << 8)}, asEntryDesc0xF010, aName0xF010, &sModulelist, NULL, NULL, 0x0000 },
    {NULL,NULL, 0xFFFF, {0, 0}, NULL, NULL, NULL, NULL}};
#endif   //#ifdef _OBJD_
```

需要特别注意蓝色圈圈中数据要与子索引最大值相对应。

2.5应用层程序更改

在APPL_Application()更改



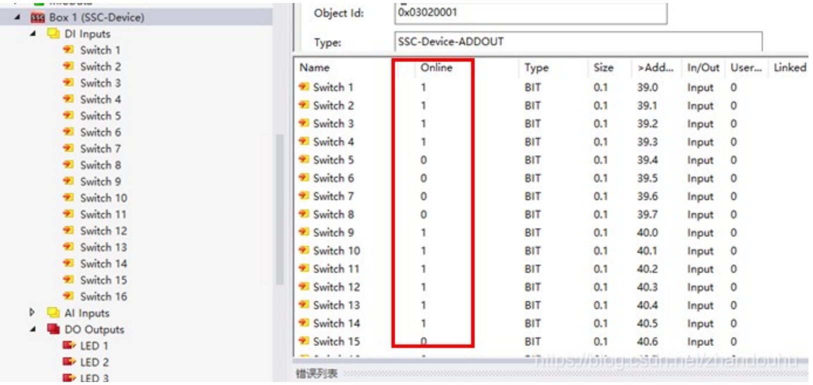
```
    LED_9                          = sDOOutputs.bLED9;
    LED_10                         = sDOOutputs.bLED10;
    LED_11                         = sDOOutputs.bLED11;
    LED_12                         = sDOOutputs.bLED12;
    LED_13                         = sDOOutputs.bLED13;
    LED_14                         = sDOOutputs.bLED14;
    LED_15                         = sDOOutputs.bLED15;
    LED_16                         = sDOOutputs.bLED16;

    sDIInputs.bSwitch1    = SWITCH_1;
    sDIInputs.bSwitch2    = SWITCH_2;
    sDIInputs.bSwitch3    = SWITCH_3;
    sDIInputs.bSwitch4    = SWITCH_4;
#if _STM32_IO8
    sDIInputs.bSwitch5    = SWITCH_5;
    sDIInputs.bSwitch6    = SWITCH_6;
    sDIInputs.bSwitch7    = SWITCH_7;
    sDIInputs.bSwitch8    = SWITCH_8;
#endif

    sDIInputs.bSwitch9     = SWITCH_9;
    sDIInputs.bSwitch10    = SWITCH_10;
    sDIInputs.bSwitch11    = SWITCH_11;
    sDIInputs.bSwitch12    = SWITCH_12;
    sDIInputs.bSwitch13    = SWITCH_13;
    sDIInputs.bSwitch14    = SWITCH_14;
    sDIInputs.bSwitch15    = SWITCH_15;
    sDIInputs.bSwitch16    = SWITCH_16;
```

3、测试验证

通过下载STM32修改后的程序，并更新xml文件，在Freerun模式下进行测试，可以发现16路拨码开关和16路LED都可以正常响应。



Object Id: 0x03020001

Type: SSC-Device-ADDOUT

Name	Online	Type	Size	>Add...	In/Out	User...	Linked
Switch 1	1	BIT	0.1	39.0	Input	0	
Switch 2	1	BIT	0.1	39.1	Input	0	
Switch 3	1	BIT	0.1	39.2	Input	0	
Switch 4	1	BIT	0.1	39.3	Input	0	
Switch 5	0	BIT	0.1	39.4	Input	0	
Switch 6	0	BIT	0.1	39.5	Input	0	
Switch 7	0	BIT	0.1	39.6	Input	0	
Switch 8	0	BIT	0.1	39.7	Input	0	
Switch 9	1	BIT	0.1	40.0	Input	0	
Switch 10	1	BIT	0.1	40.1	Input	0	
Switch 11	1	BIT	0.1	40.2	Input	0	
Switch 12	1	BIT	0.1	40.3	Input	0	
Switch 13	1	BIT	0.1	40.4	Input	0	
Switch 14	1	BIT	0.1	40.5	Input	0	
Switch 15	0	BIT	0.1	40.6	Input	0	

EtherCAT开发学习板：https://shop461235811.taobao.com

廷华电子
码龄11年 暂无认证

19	10万+	160万+	14万+	
原创	周排名	总排名	访问	等级

1502	331	127	24	946
积分	粉丝	获赞	评论	收藏

私信 关注

搜博主文章

最新评论

【BLDC理论篇】无刷直流电机与永磁同…
caihuici: 写得真好啊，nice

CSDN如何转载别人的文章
修己达人: 多谢分享

【EtherCAT实践篇】四、TwinCAT 3实…
weixin_43699335: 软件怎么汉化的

【EtherCAT实践篇】八、更改XML示例3…
m0_55104861: 加油呀，整理的挺好的！！！

【EtherCAT实践篇】一、TwinCAT3安装
烟雨成云: 我也是一直这样的bug,兄弟这个问题你解决了吗?

您愿意向朋友推荐“博客详情页”吗?

强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章

【EtherCAT实践篇】八、更改XML示例3，增加16位模拟DAC输出

【EtherCAT实践篇】七、更改XML示例2，增加16位模拟输入

【SOEM主站】一、SOEM主站环境搭建及连接板子测试

2020年 7篇　2019年 12篇

2018年 2篇

廷华电子 关注 👍 4 0 ☆ 37 ¥ 专栏目录

参与评论 您还未登录，请先 登录 后发表或查看评论

举报