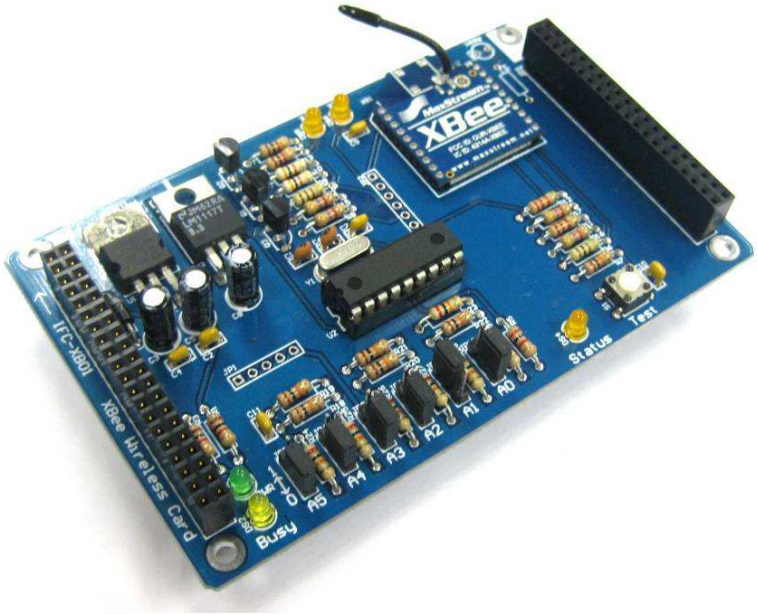
IFC-XB01
Interface Free Controller
XBee Wireless Card



Card Library Functions

V1.0

FEB 2010

IFC-XB01 Card Technical Info

This document explains the function prototype for IFC-XB01 which provide the wireless solution the to IFC system. The function prototype will be called in main program for MB00 in order to control/communicate with IFC-XB01. User can also find the explanation of function prototype in its header file, "iic_xb.h". Table 1 shows the function prototype for IFC-XB01, the example card's address used in the function prototype is 'add_xb'. Please make sure the address on IFC-XB01 is compatible to the program.

Function Prototype	Remark	Parameter Description
void xb_MY(unsigned char **add,** unsigned short MY)	xb_MY(**add_xb,** 0x12AB)	**add** = card address. Set the **16-bit Source Address** of the XBee module (will be written to non-volatile memory of the XBee module)
void xb_DL(unsigned char **add,** unsigned long DL)	xb_DL(**add_xb,** 0x0000FFFF)	**add** = card address. Set the 32-bit **Destination Address Low**. of the XBee module (will be written to non-volatile memory of the XBee module)
void xb_DH(unsigned char **add,** unsigned long DH)	xb_DH(**add_xb,** 0x00000000)	**add** = card address. Set the 32-bit **Destination Address High** of the XBee module (will be written to non-volatile memory of the XBee module)
void xb_CH(unsigned char **add,** unsigned char CH)	xb_CH(**add_xb,** 0x1A)	**add** = card address. Set the channel number used for transmitting and receiving data between IFC-XB01 cards.
void xb_ID(unsigned char **add,** unsigned short ID)	xb_ID(**add_xb,** 0x1234)	**add** = card address. Set the PAN (Personal Area Network) ID used for the IFC-XB01 cards.
void xb_send(unsigned char **add**, unsigned char data0,unsigned char data1,unsigned char data2,unsigned char data3, unsigned char data4, unsigned char data5,unsigned char data6, unsigned char data7)	xb_send(**add_xb**, 0x7E, 0x7D, 0x11, 0x13, 0xAB, 0x12, 0x34, 0x56)	**add** = card address. Send wireless data out of XBee module. 8 bytes of data can be sent out in a packet.

unsigned short xb_readMY(unsigned char **add**)	xb_readMY(**add_xb**)	**add** = card address. Read the **16-bit Source Address** of the XBee module.
unsigned long xb_readDL(unsigned char **add**)	xb_readDL(**add_xb**)	**add** = card address Read the 32-bit **Destination Address Low**. of the XBee module
unsigned long xb_readDH(unsigned char **add**)	xb_readDH(**add_xb**)	**add** = card address Read the 32-bit **Destination Address High** of the XBee module
unsigned char xb_readCH(unsigned char **add**)	xb_readCH(**add_xb**)	**add** = card address Read the channel number used for transmitting and receiving data of IFC-XB01 card.
unsigned short xb_readID(unsigned char **add**)	xb_readID(**add_xb**)	**add** = card address. Read the PAN (Personal Area Network) ID of IFC-XB01 card.
unsigned long xb_readSH(unsigned char **add**)	xb_readSH(**add_xb**)	**add** = card address. Read 32 bits **Serial Number High** of the XBee module's unique IEEE 64-bit address. 64-bit source address is always enabled.
unsigned long xb_readSL(unsigned char **add**)	xb_readSL(**add_xb**)	**add** = card address. Read 32 bits **Serial Number Low** of the XBee module's unique IEEE 64-bit address. 64-bit source address is always enabled.
void xb_receive(unsigned char **add**)	xb_receive(**add_xb**)	**add** = card address. Read received data from the XBee module.

unsigned char xb_RSSI(unsigned char **add**)	xb_RSSI(**add_xb**)	**add** = card address. Read RSSI of the received data packet.

Table 1 Function Prototype for IFC-XB01

This document also shows the examples of function prototype usage. With these examples, hopefully user can get more in-depth understanding of IFC-XB01's function prototype.

void xb_MY(unsigned char **add,** unsigned short MY)

This function is used to set the 16-bit source address of the XBee module. The address will be written to non-volatile memory of the XBee module. The address is in hexadecimal format and the range is 0 - 0xFFFF.

```
xb_MY(add_xb, 0x12AB);          //set the source address to 0x12AB
```

void xb_DL(unsigned char **add,** unsigned long DL)

This function is used to set the lower 32 bits of the 64-bit destination address of the XBee module. The address will be written to non-volatile memory of the XBee module. The address is in hexadecimal format. For the transmission using 16-bit address, set DH parameter to zero and DL less than **0xFFFF**. 0x000000000000FFFF is the broadcast address for the PAN.

```
xb_DL(add_xb, 0x0000FFFF);    //set the destination address  low to 0xFFFF(broadcast address
                              for the PAN) to broadcast to all the IFC-XB01 cards which are
                              having same channel(CH) and same PAN ID.

xb_DL(add_xb, 0x000012AB);    //set the destination address to match the source address (MY)
                              for communication with that particular MY address matching IFC-
                              XB01 card
```

void xb_DH(unsigned char **add,** unsigned long DH)

add = card address.

This function is used to set the higher 32-bits of 64 bits destination address of the XBee module. The address will be written to non-volatile memory of the XBee module. Set DH to 0 for 16-bits addressing communication. (default value of DH is zero, this function is **reserved) :**

```
xb_DH(add_xb, 0x00000000);          //set DH to 0 for 16 bits addressing
```


void xb_CH(unsigned char **add,** unsigned char CH)

This function is used to set the channel number used for transmitting and receiving data between IFC-XB01 cards. (Default value of CH is 0x0C). Users are advised to set CH to be different to avoid interference from other XBee systems that might operate in the same channel. Allowed channels are **0x0B - 0x1A** (XBee) and **0x0C - 0x17** (XBee-PRO)

```
xb_CH(add_xb, 0x1A);          //set channel to 0x1A
```

void xb_ID(unsigned char **add,** unsigned short ID)

This function is used to set the PAN (Personal Area Network) ID. Use 0xFFFF to broadcast messages to all PANs.

```
xb_ID(add_xb, 0x1234);          //set the PAN ID to 1234
```

void xb_send(unsigned char **add**, unsigned char data0,unsigned char data1,unsigned char data2,unsigned char data3, unsigned char data4, unsigned char data5,unsigned char data6, unsigned char data7)

This function is used to send wireless data out of XBee module. 8 bytes of data can be sent out in a packet. API mode 2 is used in IFC-XB01 for send function.

```
xb_send(add_xb, 0x7E, 0x7D, 0x11, 0x13, 0xAB, 0x12, 0x34, 0x56)        //sending 8 bytes of data
```

unsigned short xb_readMY(unsigned char **add**)

This function is used to read the 16-bit source address of the XBee module.

```
myadd=xb_readMY(add_xb);    //read source address of the XBee module and save to variable
                             myadd.
```

unsigned long xb_readDL(unsigned char **add**)

This function is used to read the lower 32 bits of the 64-bit destination address of the XBee module.

```
dest_L=xb_readDL(add_xb);   //read destination address low of the XBee module and save
                             to variable dest_L.
```

unsigned long xb_readDH(unsigned char **add**)

This function is used to read the higher 32-bits of 64 bits destination address of the XBee module.

```
dest_h=xb_readDH(add_xb) ;   // read destination address high of the XBee module and save to
                               variable dest_h
```

unsigned char xb_readCH(unsigned char **add**)

This function is used to read the channel number used for transmitting and receiving data of the IFC-XB01 card.

```
Channel=xb_readCH(add_xb);  // read  CH of the XBee module and save to variable Channel
```

unsigned short xb_readID(unsigned char **add**)

This function is used to read the PAN (Personal Area Network) ID of the XBee module.

```
PANID=xb_readID(add_xb);      //read the PAN ID of the XBee module and save to variable PANID
```

unsigned long xb_readSH(unsigned char **add**)

This function is used to read 32 bits **Serial Number High** of the XBee module's unique IEEE 64-bit address. 64-bit source address is always enabled.

```
s_high=xb_readSH(add_xb);     //read serial number high and save to variable s_high
```

unsigned long xb_readSL(unsigned char **add**)

This function is used to read 32 bits **Serial Number Low** of the XBee module's unique IEEE 64-bit address. 64-bit source address is always enabled.

```
s_low=xb_readSL(add_xb);              //read serial number low and save to variable s_low
```

void xb_receive(unsigned char **add**)

This function is used read received data from the XBee module. User need to poll IFC_XB01 from main board using this function to keep update and to enable IFC_XB01 read received wireless data. The received data are saved to predefined array, rec_data[8], user will need to read this array to retrieve the received data.

```
xb_receive(add_xb);          //read 8 bytes received data
```

Note: Be cautious! If user does not read the received wireless data packet, the data packet will be buffered until the buffer full (200 bytes). Wireless data received after the buffer full will be **LOST**.

unsigned char xb_RSSI(unsigned char **add**)

This function is used to read RSSI of the received data packet. RSSI is Received Signal Strength Indicator. The one byte RSSI value is hexadecimal equivalent of (-dBm) value. (For example: If RX signal strength = -40 dBm, “0x28” (40 decimal) is returned).

```
rssi=xb_RSSI(add_xb);        //read RSSI of the receive data packet and save to variable rssi
```


Note: User needs to add header file (iic.h and iic_xb.h) and object file (iic.o and iic_xb.o) for IFC-MB00 and IFC-XB01 each time user opens a new project for IFC. User also needs to include card's header file at the beginning of the program. Please refer sample source code for the example to include card's header file.

Prepared by
Cytron Technologies Sdn. Bhd.
19, Jalan Kebudayaan 1A,
Taman Universiti,
81300 Skudai,
Johor, Malaysia.

Tel: +607-521 3178
Fax: +607-521 1861

URL: www.cytron.com.my
Email: support@cytron.com.my
sales@cytron.com.my

