

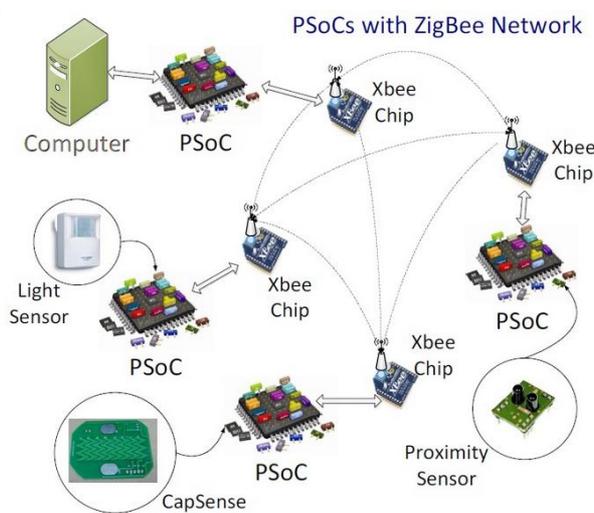
## 1. XBEE PSoC

### Sommaire

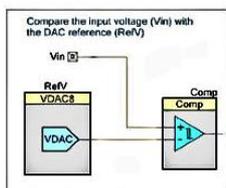


Configuration & Test Utility Software

A Digi International Inc. product.



<b>Communications logiques .....</b>	<b>2</b>
<i>Liste du materiel requis : .....</i>	<i>2</i>
<i>Configuration générale.....</i>	<i>3</i>
1) Pour le coordonnateur API .....	3
2) Pour les deux end device.....	3
<i>Fonctionnement : .....</i>	<i>4</i>
<b>Commmade à distance .....</b>	<b>7</b>
<i>Liste du materiel requis : .....</i>	<i>7</i>
<i>Configuration.....</i>	<i>7</i>
<i>Mise en oeuvre utilisation de trame API .....</i>	<i>7</i>
<i>Une vue générale du réseau star 3 xbees .....</i>	<i>9</i>
<i>Commande à distance d'un noeud avec un programme windows (Delphi) .....</i>	<i>9</i>
<i>Organisation du code Delphi la partie émission de trame.....</i>	<i>10</i>
<b>Le PSoC entre en scène.....</b>	<b>11</b>
<i>Ajout d'un module supplémentaire.....</i>	<i>11</i>
<i>Transmission du coordinator vers le PSoC (Router).....</i>	<i>12</i>
<i>Transmission d'une trame depuis le PSoC vers le coordonnateur .....</i>	<i>13</i>
<i>Connexion du module XBee sur le PSoC5 (Kit014).....</i>	<i>14</i>
<b>Descriptions des trames API utilisées .....</b>	<b>15</b>
<i>Description générale d'une trame .....</i>	<i>15</i>
<i>La liste des trames possibles .....</i>	<i>15</i>
<i>Le calcul du checksum :.....</i>	<i>16</i>
<i>ZigBee requête de transmission de données 0x10.....</i>	<i>17</i>
<i>ZigBee requête de réception de paquet 0x90.....</i>	<i>19</i>
<i>ZigBee indicateur d'IO distant 0x92.....</i>	<i>20</i>
<b>Schéma général du réseau XBee .....</b>	<b>22</b>



## Communications logiques

### Liste du materiel requis :

- ↳ 2 platines XBee explorer régulée Sparkfun WRL-11373
- ↳ 1 platine XBee explorer dongle USB Sparkfun-11697
- ↳ 3 modules XBee

### Chapitre 3

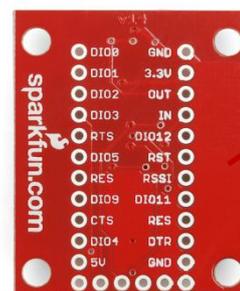
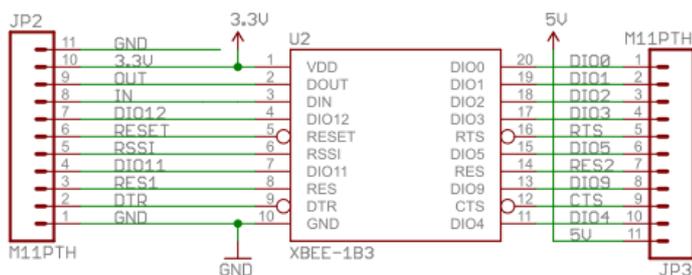


### Sparkfun-



### Sparkfun-

- ↳ une alimentation 5V : la platine WRL-11373 délivre la tension de 3,3V au module XBee.



Vue de  
dessus

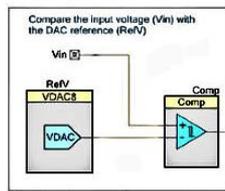
- ↳ L'adaptateur breakout board de sparkfun redistribue les broches :

GND  
3.3  
OUT  
IN  
DIO12  
RST  
RSSI  
DIO11  
RES  
DTR  
GND



DIO0  
DIO1  
DIO2  
DIO3  
RTS  
DIO5  
RES  
DIO9  
CTS  
DIO4  
5V





## Configuration générale

### 1) Pour le coordonnateur API

↳ régler le numéro de netid :

#### Networking

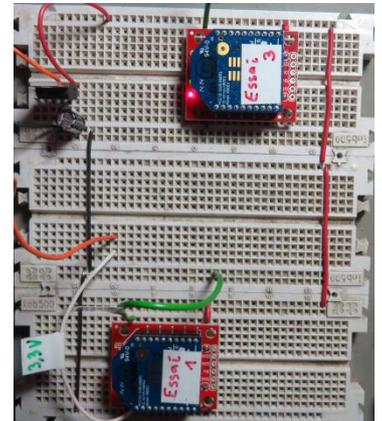
Change networking settings

**ID PAN ID**

#### Serial Interfacing

Change modem interfacing options

**AO API Output Mode**



### 2) Pour les deux end device

Quand le réseau fonctionne XCTU à accès à la configuration de n'importe quel noeud du réseau. Donc on peut observer ou modifier la configuration sans avoir à les débrancher.

↳ Les deux end device sont câblés sur une platine labdec.

↳ Les deux configurations sont identiques sauf le délai de transmission IR (12C soit 300 ms)

#### Networking

Change networking settings

**ID PAN ID**

#### Addressing

Change addressing settings

**DH Destination Address High**

**DL Destination Address Low**

#### I/O Settings

Modify DIO and ADC options

**DO AD0/DIO0 Configuration**

**PR Pull-up Resistor Enable**

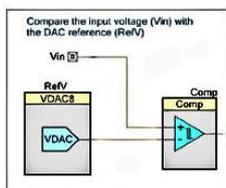
#### I/O Sampling

Configure IO sampling parameters

**IR IO Sampling Rate**  x 1 ms

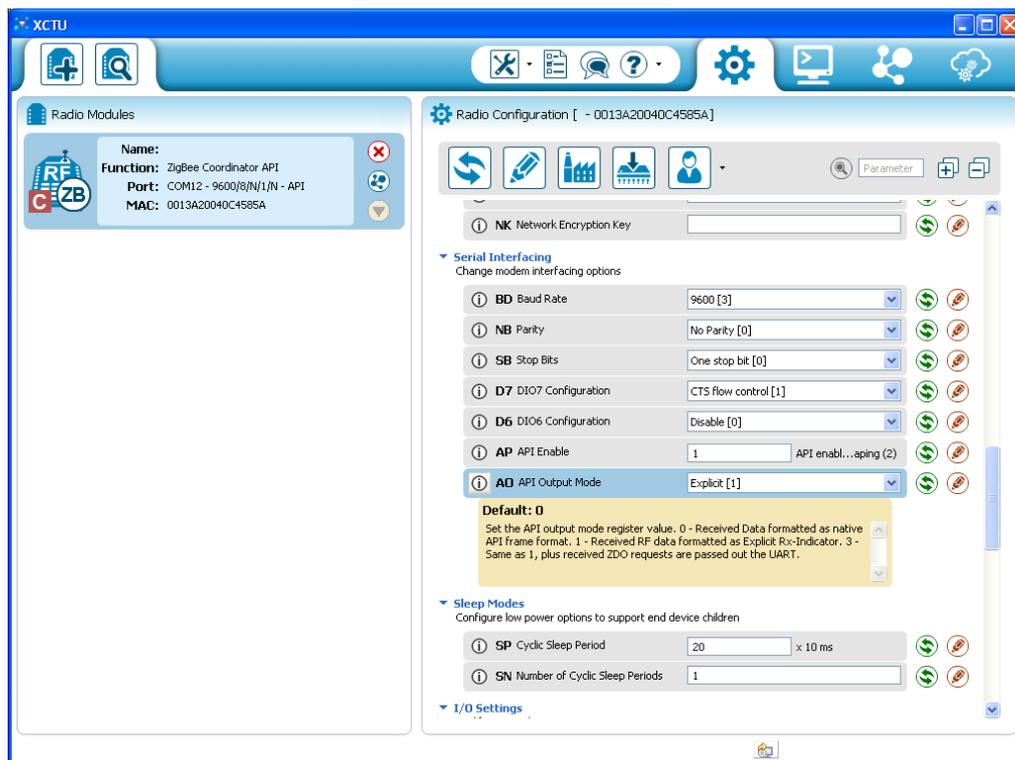
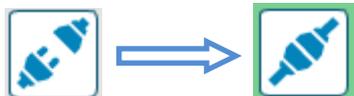


# PSoC PSoC



## Fonctionnement :

les trames XBee reçues peuvent être visualisées avec XCTU. Il faut connecter la liaison série au module radio avec le bouton **Open the serial connection with the radio module.**



Trame reçue DIO0 valeur 1 (en l'air) ou 0 (relié à GND), analyse par XCTU

### Zigbee IO Data Sample RX Indicator

7E 00 12 92 00 13 A2 00 40 C1 2A BF 69 CF 41 01 00 01 00 00 01 52

- Start delimiter: 7E
- Length: 00 12 (18)
- Frame type: 92 (Zigbee IO Data Sample RX Indicator)
- 64-bit source address: 00 13 A2 00 40 C1 2A BF
- 16-bit source address: 69 CF
- Receive options: 41
- Number of samples: 01
- Digital channel mask: 00 01
- Analog channel mask: 00
- Digital samples: 00 01
- Checksum: 52

### Zigbee IO Data Sample RX Indicator

7E 00 12 92 00 13 A2 00 40 C1 2A BF 69 CF 41 01 00 01 00 00 00 53

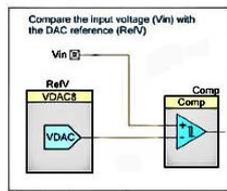
- Start delimiter: 7E
- Length: 00 12 (18)
- Frame type: 92 (Zigbee IO Data Sample RX Indicator)
- 64-bit source address: 00 13 A2 00 40 C1 2A BF
- 16-bit source address: 69 CF
- Receive options: 41
- Number of samples: 01
- Digital channel mask: 00 01
- Analog channel mask: 00
- Digital samples: 00 00
- Checksum: 53

Trame vue par testcom (entrée à gnd) longueur 22 [API output mode 0]:

7E 00 12 92 00 13 A2 00 40 C1 2A BF 69 CF 41 01 00 01 00 00 00 53



# PSoC PSoC



**Radio Modules**

- Name: ZigBee ...tor API
- Function: ZigBee ...tor API
- Port: COM12 ... -API
- MAC: 0013A...4585A

API Console Status: Connected Tx Frames: 0 Rx Frames: 505

ID	Time	Lo...	Frame
496	14:52:49...	18	Zigbee IO Data Sample RX Indicator
497	14:52:50...	18	Zigbee IO Data Sample RX Indicator
498	14:52:50...	18	Zigbee IO Data Sample RX Indicator
499	14:52:50...	18	Zigbee IO Data Sample RX Indicator
500	14:52:50...	18	Zigbee IO Data Sample RX Indicator
501	14:52:50...	18	Zigbee IO Data Sample RX Indicator
502	14:52:51...	18	Zigbee IO Data Sample RX Indicator
503	14:52:51...	18	Zigbee IO Data Sample RX Indicator
504	14:52:51...	18	Zigbee IO Data Sample RX Indicator

Frame details: Zigbee IO Data Sample RX Indicator  
7E 00 12 92 00 13 A2 00 40 C1 2A BF 65 7D 41 01 00 01 00 00 A9

- Start delimiter: 7E
- Length: 00 12 (18)
- Frame type: 92 (Zigbee IO Data Sample RX Indicator)
- 64-bit source address: 00 13 A2

**Radio Modules**

- Name: ZigBee Coordinator API
- Function: ZigBee Coordinator API
- Port: COM12 - 9600/8/N/1/N - API
- MAC: 0013A20040C4585A

2 remote modules

- Name: ZigBee End Device API
- Function: ZigBee End Device API
- MAC: 0013A20040C12ABF
- Fessai

- Name: ZigBee End Device API
- Function: ZigBee End Device API
- MAC: 0013A20040C12AC2
- Fessai

API Console Status: Connected Tx Frames: 10 Rx Frames: 1566

ID	Time	L.	Frame
1...	14:59:...	18	Zigbee IO Data Sample RX I...
1...	14:59:...	18	Zigbee IO Data Sample RX I...
1...	14:59:...	18	Zigbee IO Data Sample RX I...
1...	14:59:...	18	Zigbee IO Data Sample RX I...
1...	14:59:...	18	Zigbee IO Data Sample RX I...
1...	14:59:...	18	Zigbee IO Data Sample RX I...
1...	14:59:...	18	Zigbee IO Data Sample RX I...
1...	14:59:...	18	Zigbee IO Data Sample RX I...
1...	14:59:...	18	Zigbee IO Data Sample RX I...
1...	14:59:...	18	Zigbee IO Data Sample RX I...

Frame details: Zigbee IO Data Sample RX Indicator  
7E 00 12 92 00 13 A2 00 40 C1 2A BF 65 7D 41 01 00 01 00 00 00 A9

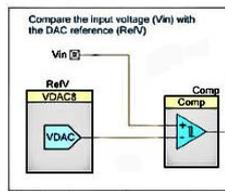
- Start delimiter: 7E
- Length: 00 12 (18)
- Frame type: 92 (Zigbee

Trame reçue toutes les secondes longueur 28 [API output mode 1]:

7E 00 18 91 00 13 A2 00 40 C1 2A BF 3A 8A E8 E8 00 92 C1 05 41 01 00 01 00 00 01 9F  
 7E 00 18 91 00 13 A2 00 40 C1 2A BF 69 CF E8 E8 00 92 C1 05 41 01 00 01 00 00 00 2C



# PSoC PSoC



↳ Penser à régler le pull-up si besoin :

**PR Pull-up Resistor Enable**

**Range: 0x0 - 0x3FFF (Default: 1FFF)**

Set/read bitfield to configure internal pullup resistors status for I/O lines. 1=internal pullup enabled, 0=no internal pullup. Bitfield map: (13)-DIO7/CTS, (12)-DIO11, (11)-DIO10/PWM0, (10)-DIO12, (9)-On/Sleep, (8)-Associate, (7)-DIN/Config, (6)-Sleep\_Rq, (5)-RTS, (4)-AD0/DIO0, (3)-AD1/DIO1,

**IC Digital IO Change Detection**

**Range: 0x0 - 0xFFFF (Default: 0)**

Bitfield that configures which digital IO pins should be monitored for change detection. If a change is detected on an enabled digital IO pin, a digital IO sample is immediately transmitted to the address specified by DH+DL.

Ajout d'un deuxième End Device (Essai 3) configuré sur un envoi de l'entrée DIO0 toutes les 300 ms IR=0x12C

```

7E 00 12 92 00 13 A2 00 40 C1 2A BF 8F 81 41 01 00 01 00 00 00 7B
7E 00 12 92 00 13 A2 00 40 C1 2A C2 0F 6B 41 01 00 01 00 00 01 0D
7E 00 12 92 00 13 A2 00 40 C1 2A C2 0F 6B 41 01 00 01 00 00 01 0D
7E 00 12 92 00 13 A2 00 40 C1 2A C2 0F 6B 41 01 00 01 00 00 01 0D
7E 00 12 92 00 13 A2 00 40 C1 2A BF 8F 81 41 01 00 01 00 00 00 7B
  
```

Quelques questions :

- ↳ comment identifier l'origine du message ?
- ↳ comment envoyer un état à tous le réseau ?
- ↳ revoir la récupération de trame avec PSoC et Delphi pour le maître.

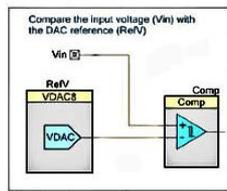
Exemple d'envoi d'info digital après un arrêt et une remise sous tension, environ 1mn d'attente avant le départ du réseau :

```

7E 00 12 92 00 13 A2 00 40 C1 2A C2 0F 6B 41 01 00 01 00 00 01 0D
7E 00 12 92 00 13 A2 00 40 C1 2A C2 0F 6B 41 01 00 01 00 00 01 0D
7E 00 12 92 00 13 A2 00 40 C1 2A BF 8F 81 41 01 00 01 00 00 00 7B
7E 00 12 92 00 13 A2 00 40 C1 2A C2 0F 6B 41 01 00 01 00 00 01 0D
7E 00 12 92 00 13 A2 00 40 C1 2A C2 0F 6B 41 01 00 01 00 00 01 0D
7E 00 12 92 00 13 A2 00 40 C1 2A C2 00 FC 41 01 00 01 00 00 01 0D
7E 00 12 92 00 13 A2 00 40 C1 2A C2 00 FC 41 01 00 01 00 00 01 8B
7E 00 12 92 00 13 A2 00 40 C1 2A BF 00 7E 41 01 00 01 00 00 00 0D
7E 00 12 92 00 13 A2 00 40 C1 2A C2 00 FC 41 01 00 01 00 00 01 8B
7E 00 12 92 00 13 A2 00 40 C1 2A C2 00 FC 41 01 00 01 00 00 01 8B
  
```

↳ Arrêt et redémarrage





## Commande à distance

### Liste du matériel requis :

↳ Le matériel est inchangé, voir chapitre précédent.

### Configuration

↳ la sortie DIO1 est configurée, une led est connectée via une résistance de 120Ω

↳ Il faut modifier la configuration avec XCTU. A noter qu'il est possible de reconfigurer n'importe quel noeud du réseau à distance avec ce logiciel.

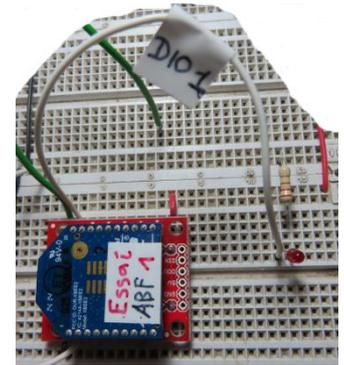
Chapitre 12



#### I/O Settings

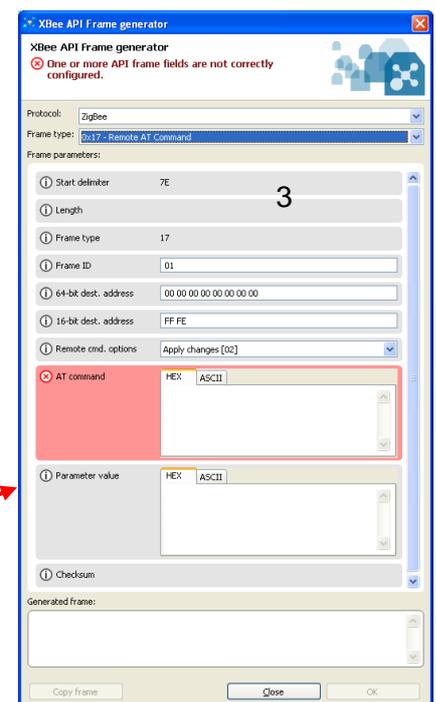
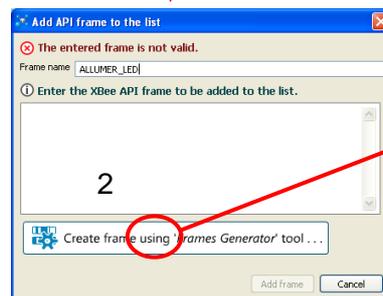
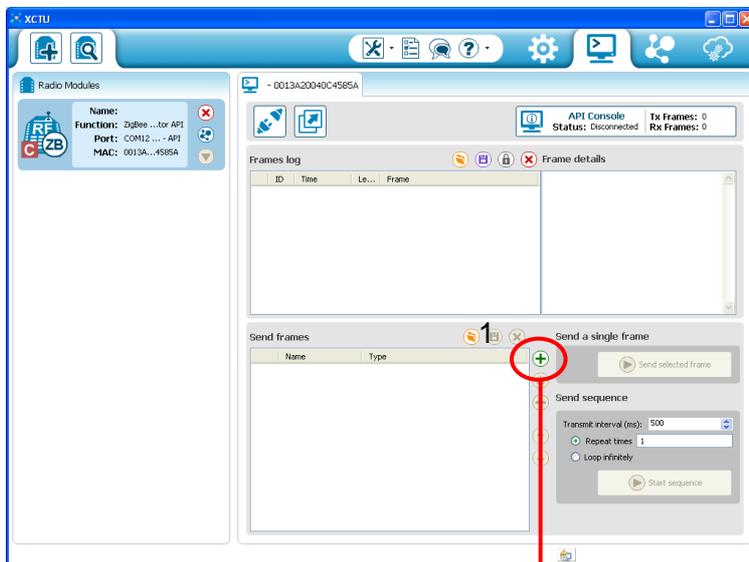
Modify DIO and ADC options

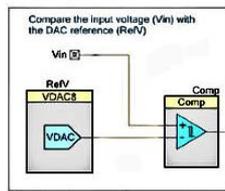
<b>D0</b> AD0/DIO0 Configuration	Digital Input [3]
<b>D1</b> AD1/DIO1 Configuration	Digital Out, High [5]



### Mise en oeuvre utilisation de frame API

↳ nous allons utiliser une frame API pour transmettre notre commande AT. Le logiciel XCTU permet de construire cette frame :





↳ Saisie et envoi de la trame avec XCTU :

**XBee API Frame generator**

Protocol: ZigBee  
Frame type: 0x17 - Remote AT Command

Frame parameters:

- Start delimiter: 7E
- Length: 00 10 **3**
- Frame type: 17
- Frame ID: 01
- 64-bit dest. address: 00 13 A2 00 40 C1 2A BF
- 16-bit dest. address: FF FE
- Remote cmd. options: Apply changes [02]
- AT command: HEX ASCII  
44 31
- Parameter value: HEX ASCII  
04
- Checksum: D0

Generated frame:  
7E 00 10 17 01 00 13 A2 00 40 C1 2A BF FF FE 02 44 31 04 D0

Buttons: Copy frame, Close, **OK**

Trame extinction de la led

**Edit selected frame**

Enter the new name of the frame.  
Frame name: Eteint\_Led\_Essai\_1

The specified XBee API packet is correct!

7E 00 10 17 01 00 13 A2 00 40 C1 2A BF FF FE 02  
44 31 05 CF

Buttons: Apply changes, Cancel

Trame allumage de la led

**Add API frame to the list**

Enter the name of the XBee API frame to be added to the list.  
Frame name: ALLUMER\_LED

The specified XBee API packet is correct!

7E 00 10 17 01 00 13 A2 00 40 C1 2A BF FF FE 02  
44 31 04 D0

Buttons: Add frame, Cancel

4

**XCTU**

Radio Modules: ZigBee ... for API, COM12 ... - API, MAC: 0013A...4585A

API Console Status: Connected, Tx Frames: 2, Rx Frames: 839

Frames log:

ID	Time	L...	Frame
831	23:23:30...	18	Zigbee IO Data Sample RX Indicator
832	23:23:30...	18	Zigbee IO Data Sample RX Indicator
833	23:23:30...	18	Zigbee IO Data Sample RX Indicator
834	23:23:31...	18	Zigbee IO Data Sample RX Indicator
835	23:23:31...	18	Zigbee IO Data Sample RX Indicator
836	23:23:31...	18	Zigbee IO Data Sample RX Indicator
837	23:23:31...	18	Zigbee IO Data Sample RX Indicator
838	23:23:31...	18	Zigbee IO Data Sample RX Indicator
839	23:23:32...	18	Zigbee IO Data Sample RX Indicator
840	23:23:32...	18	Zigbee IO Data Sample RX Indicator

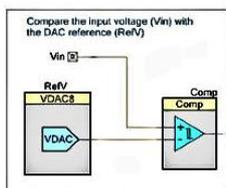
Send frames:

Name	Type
ALLUMER_LED	Remote AT Command Request
ETEINDRE_LED	Remote AT Command Request

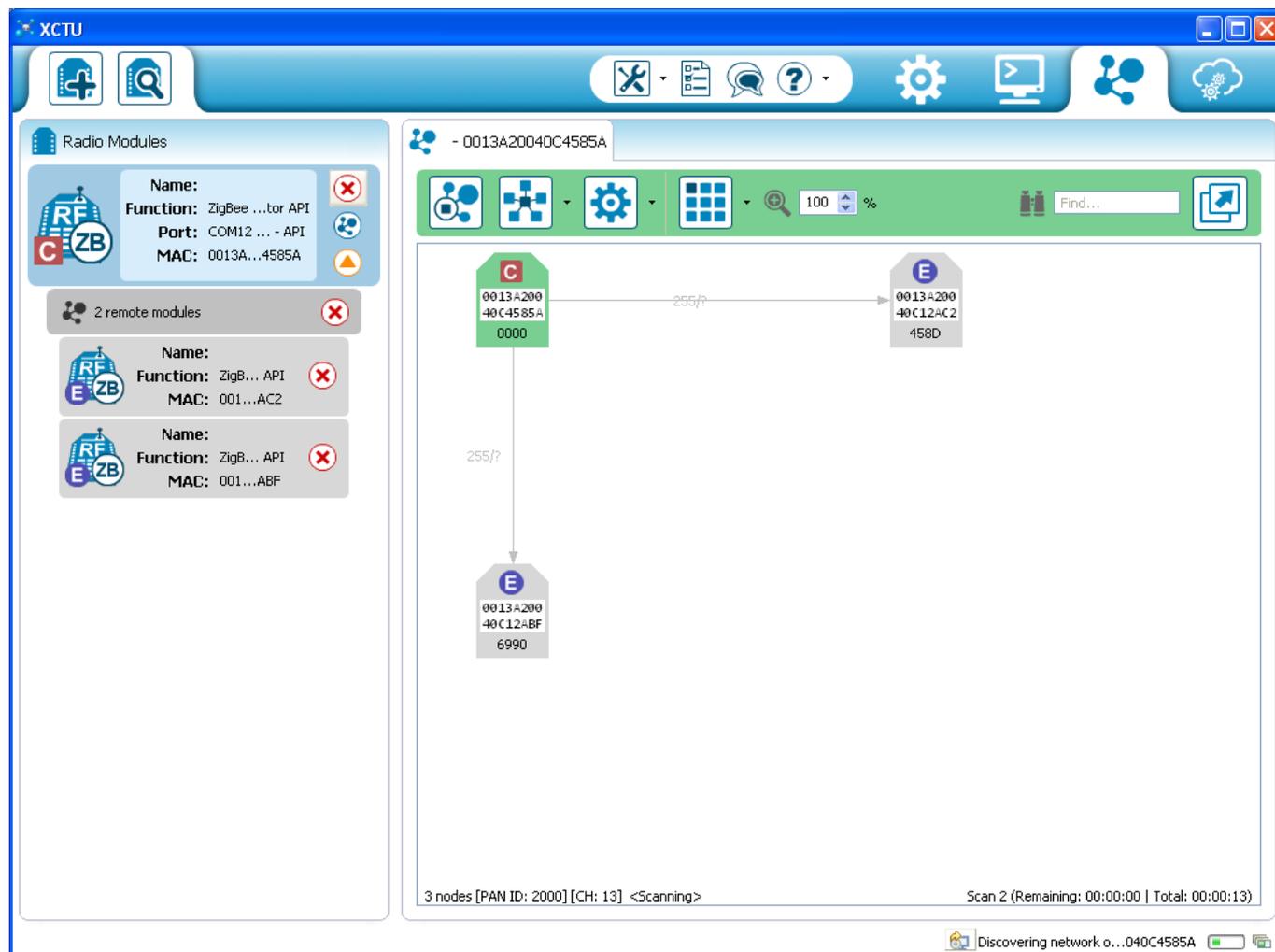
Buttons: Send selected frame, Start sequence

5





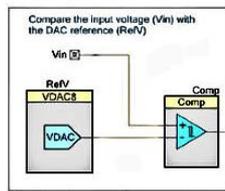
## Une vue générale du réseau star 3 xbees



## Commande à distance d'un noeud avec un programme windows (Delphi)

- ↳ Nous connaissons la structure des deux trames API qui assurent les commandes AT d'allumage et d'extinction de la LED connectée sur la broche DIO1 de l'END DEVICE d'adresse : 00 13 A2 00 40 C1 2A BF
- ↳ Ces trames peuvent être générées avec un programme utilisant la liaison série. Pour ce faire Delphi6 est utilisé. Simple de mise en oeuvre langage Pascal Objet en français, avec une aide en français il permet de construire rapidement des interfaces windows agréable et fonctionnelle. Delphi6 est gratuit.
- ↳ La mise en oeuvre est très simple :
  - choix du port com sur lequel est connecté le coordinateur (dans notre cas com12)
  - connection sur ce port
  - les trames reçues qui proviennent des deux end device (voir paragraphe 1)
  - l'appui sur le bouton allumage ou extinction provoque l'action concernée sur la led distante.
  - comme un écho est demandé le end device acquitte la bonne exécution de la commande.





## Action sur le réseau xbee à partir de l'application Delphi

## Organisation du code Delphi la partie émission de trame

↳ Les octets des deux trames sont envoyés par la commande ComPort1.WriteStr

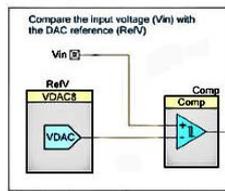
```

procedure TForm1.Button3Click(Sender: TObject);
begin
    if ( Comport1.Connected = true ) then
        ComPort1.WriteStr(chr($7E)
            + chr($00)
            + chr($10)
            + chr($17)
            + chr($01)
            + chr($00)
            + chr($13)
            + chr($A2)
            + chr($00)
            + chr($40)
            + chr($C1)
            + chr($2A)
            + chr($BF)
            + chr($FF)
            + chr($FE)
            + chr($02)
            + chr($44)
            + chr($31)
            + chr($05)
            + chr($CF)
        );
    else ShowMessage('Port com non connecté');
end;
    
```

```

procedure TForm1.Button1Click(Sender: TObject);
begin
    if ( Comport1.Connected = true ) then
        ComPort1.WriteStr(chr($7E)
            + chr($00)
            + chr($10)
            + chr($17)
            + chr($01)
            + chr($00)
            + chr($13)
            + chr($A2)
            + chr($00)
            + chr($40)
            + chr($C1)
            + chr($2A)
            + chr($BF)
            + chr($FF)
            + chr($FE)
            + chr($02)
            + chr($44)
            + chr($31)
            + chr($04)
            + chr($D0)
        );
    else ShowMessage('Port com non connecté');
end;
    
```





## Le PSoC entre en scène

### Ajout d'un module supplémentaire

↳ Choix Firmware Router API

↳ Pan ID : 2000

↳ Adresse de destination adresse du coordinateur : 00 13 A2 00 40 C4 58 5A

Radio Modules

Name: ZigBee Coordinator API  
Function: ZigBee Coordinator API  
Port: COM12 - 9600/8/N/1/N - API  
MAC: 0013A20040C4585A

3 remote modules

Name: ZigBee Router API  
Function: ZigBee Router API  
MAC: 0013A20040C45883

Name: ZigBee End Device API  
Function: ZigBee End Device API  
MAC: 0013A20040C12AC2

Name: ZigBee End Device API  
Function: ZigBee End Device API  
MAC: 0013A20040C12ABF

↳ description du réseau : Le coordinateur est en liaison avec le programme XBEE\_STAR, celui-ci affiche les trames reçues et envoyées.

Connection PORT  
COM12

Radio Modules

Name: ZigBee Coordinator API  
Function: ZigBee Coordinator API  
Port: COM12 - 9600/8/N/1/N - API  
MAC: 0013A20040C4585A

3 remote modules

Name: ZigBee End Device API  
Function: ZigBee End Device API  
MAC: 0013A20040C12AC2

Name: ZigBee Router API  
Function: ZigBee Router API  
MAC: 0013A20040C45883

Name: ZigBee End Device API  
Function: ZigBee End Device API  
MAC: 0013A20040C12ABF

.58 5A

.2A C2

.2A BF

Eteindre la LED

Allumer la LED

.58 83

Vers PSoC

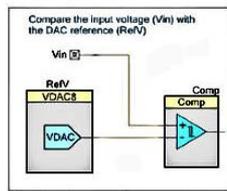
Effacement

Allumage extinction de la led sur le End Device

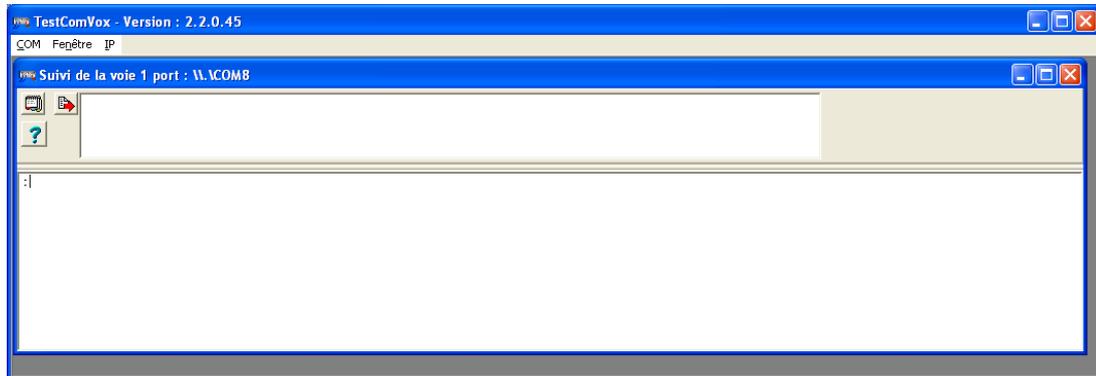
Envoi d'une trame au Router



# PSoC PSoC



↳ le module XBee router est connecté sur la carte PSoC. Le PSoC retransmet tous les octets reçus ou envoyés sur la liaison série PC. Le sniffer TestCom permet de visualiser les trames.

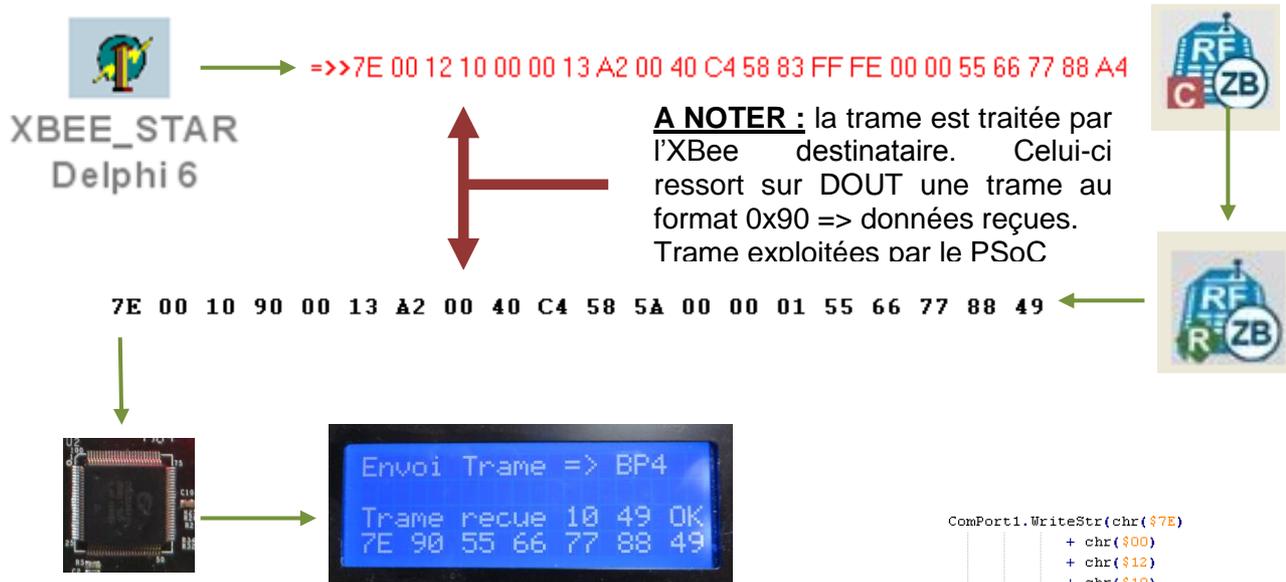


## Transmission du coordinator vers le PSoC (Router)

↳ le PSoC attend la réception d'une trame



↳ la trame est envoyée :

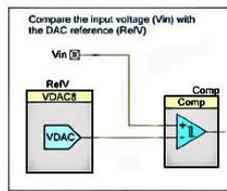


```
ComPort1.WriteStr(chr($7E)
+ chr($00)
+ chr($12)
+ chr($10)
+ chr($00)
+ chr($00)
+ chr($13)
+ chr($A2)
+ chr($00)
+ chr($40)
+ chr($C4)
+ chr($58)
+ chr($83)
+ chr($FF)
+ chr($FE)
+ chr($00)
+ chr($00)
+ chr($55)
+ chr($66)
+ chr($77)
+ chr($88)
+ chr($A4)
);
```

↳ Programme d'envoi de la trame par appui sur le BP dans Delphi6 :

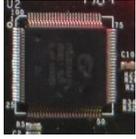


# PSoC PSoC



## Transmission d'une trame depuis le PSoC vers le coordonnateur

↳ Le PSoC transmet une trame à destination du coordonnateur



→ 7E 00 0F 10 00 00 13 A2 00 40 C4 58 5A FF FE 00 00 24 63



XBEE\_STAR  
Delphi 6

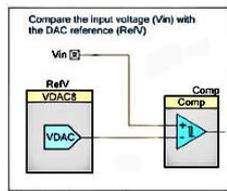
← 7E 00 0D 90 00 13 A2 00 40 C4 58 83 CA 98 01 24 54 ←



↳ l'envoi de la trame dans le PSoC

```
if (BP4_Push==1) {  
    XBEE_PutArray(trame_API_Transfert_1_Octet, longueurTrame) ;  
    PC_PutArray(trame_API_Transfert_1_Octet, longueurTrame) ;  
}
```





## Connexion du module XBee sur le PSoC5 (Kit014)

Deux composants UART sont utilisés le premier pour la liaison avec le module XBee et le second pour la liaison vers un port COM de PC via la l'interface RS232.

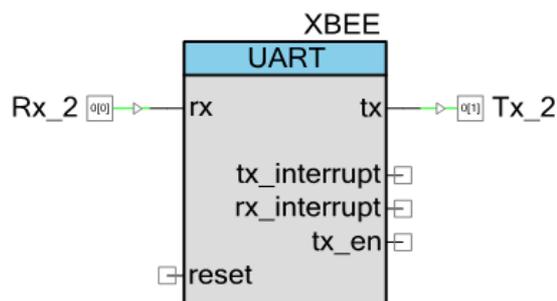
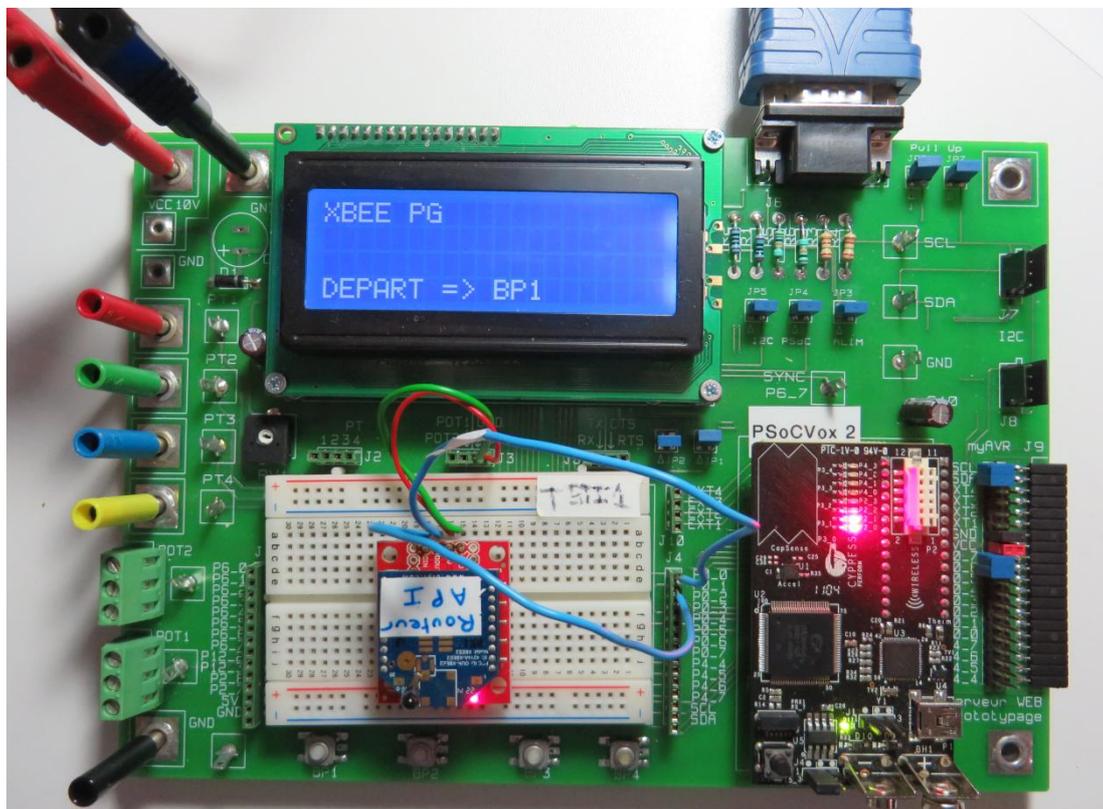
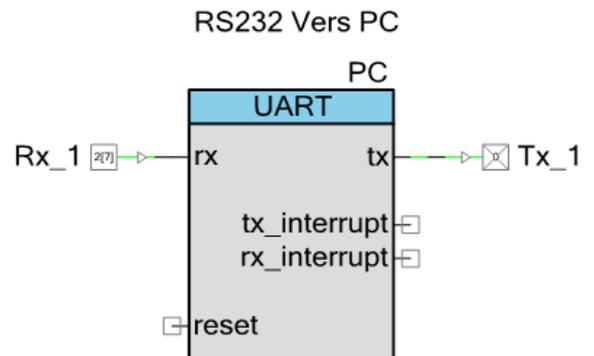
### L'UART XBee

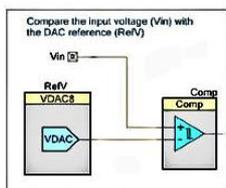
DIN xbee => P0\_1 (Tx XBEE UART component)

DOUT xbee => P0\_0 (Rx XBEE UART component)

### L'UART PC (PSoC vers RS232 carte PSoCVox) utilise

les liaisons RX => P2\_7 et Tx => P2\_6

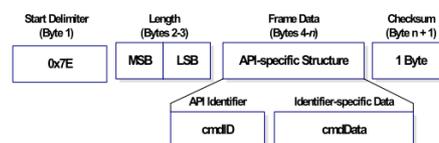




## Descriptions des trames API utilisées

↳ ZigBee RF Modules User Guide.pdf

[ftp1.digi.com/support/documentation/90000976\\_W.pdf](ftp1.digi.com/support/documentation/90000976_W.pdf)



### Description générale d'une trame



Attention certains caractères ne doivent pas se trouver dans ces séquences :

#### Data bytes that need to be escaped:

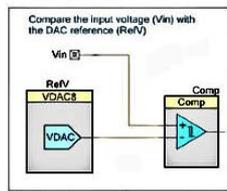
- 0x7E – Frame Delimiter
- 0x7D – Escape
- 0x11 – XON
- 0x13 – XOFF

Voir à ce sujet la page 113 du document ressource.

### La liste des trames possibles

API Frame Names	API ID
AT Command	0x08
AT Command - Queue Parameter Value	0x09
ZigBee Transmit Request	0x10
Explicit Addressing ZigBee Command Frame	0x11
Remote Command Request	0x17
Create Source Route	0x21
AT Command Response	0x88
Modem Status	0x8A
ZigBee Transmit Status	0x8B
ZigBee Receive Packet (AO=0)	0x90
ZigBee Explicit Rx Indicator (AO=1)	0x91
ZigBee IO Data Sample Rx Indicator	0x92
XBee Sensor Read Indicator (AO=0)	0x94
Node Identification Indicator (AO=0)	0x95
Remote Command Response	0x97
Over-the-Air Firmware Update Status	0xA0
Route Record Indicator	0xA1
Many-to-One Route Request Indicator	0xA3





## Le calcul du checksum :

Le checksum se calcul sur les octets de Data Frame. Tous ces octets sont additionnés et la somme obtenue est soustraite à 0xFF :

**Example:** Create an API AT command frame to configure an XBee to allow joining (set NJ to 0xFF). The frame should look like:

```
0x7E 0x00 0x05 0x08 0x01 0x4E 0x4A 0xFF 5F
```

Where

0x0005 = length

0x08 = AT Command API frame type

0x01 = Frame ID (set to non-zero value)

0x4E4A = AT Command ('NJ')

0xFF = value to set command to

0x5F = Checksum

The checksum is calculated as  $[0xFF - (0x08 + 0x01 + 0x4E + 0x4A + 0xFF)]$

↳ la procedure dans le projet PSoC : La trame est dans la variable de type tableau `table[]`, comme indiqué dans la norme le checksum est calculé à partir de l'octet n°3, (rappel le décompte commence à 0, l'octet n°3 est le premier de DataFrame). La somme obtenue ramenée sur les 8 bits de poids faible est soustraite à 0xFF :

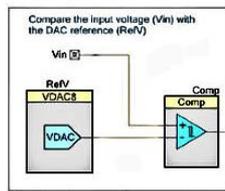
```
/* Calcul du Checksum Cypress */
/* To calculate: Not including frame delimiters and length,
 * add all bytes keeping only the lowest 8
 * bits of the result and subtract the result from 0xFF.
 * Returns checksum of contents in TxPacket based on the packet length entries */
uint8 CalculDuChecksum(uint8 table[], uint8 nombreElements)
{
    uint16 tempchecksum = 0;
    uint16 i;

    for(i = 3; i < nombreElements-1; i++) {
        tempchecksum += table[i];
    }

    tempchecksum &= 0x00FF;

    return (uint8)(0xFF - tempchecksum);
}
```



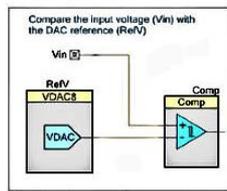


## ZigBee requête de transmission de données 0x10

Cette trame permet de transférer des données d'un point à un autre du réseau.

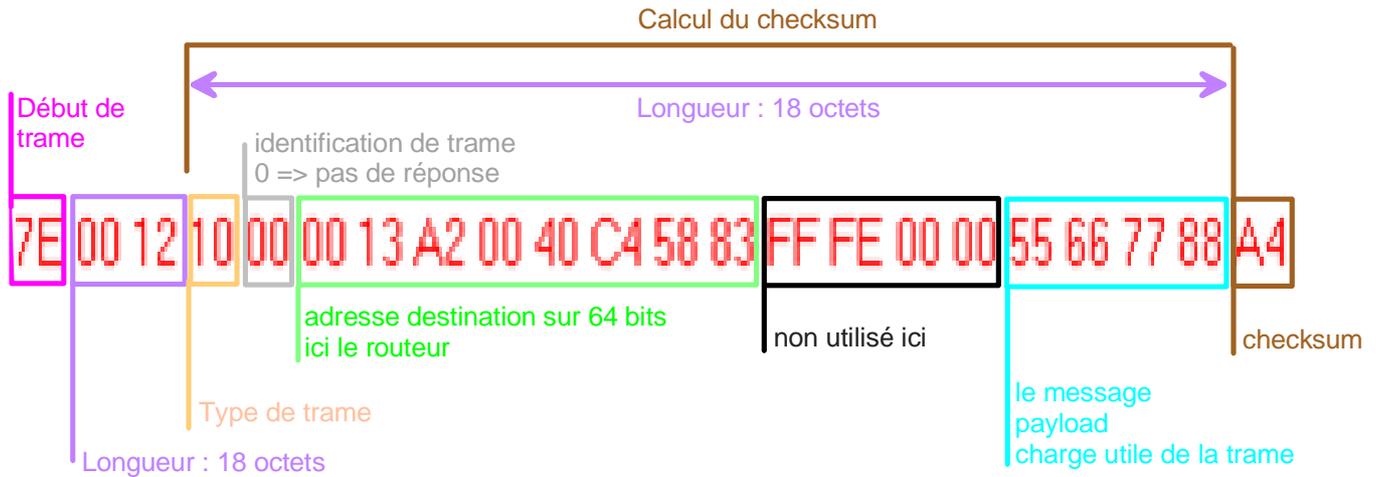
Frame Fields	Offset	Example	Description	
Start Delimiter	0	0x7E		
Length	MSB 1	0x00	Number of bytes between the length and the checksum	
	LSB 2	0x16		
Frame-specific Data	Frame Type	3	0x10	
	Frame ID	4	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgment). If set to 0, no response is sent.
	64-bit Destination Address	MSB 5	0x00	Set to the 64-bit address of the destination device. The following addresses are also supported: 0x0000000000000000 - Reserved 64-bit address for the coordinator 0x000000000000FFFF - Broadcast address
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x40	
		10	0x0A	
		11	0x01	
	LSB 12	0x27		
16-bit Destination Network Address	MSB 13	0xFF	Set to the 16-bit address of the destination device, if known. Set to 0xFFFE if the address is unknown, or if sending a broadcast.	
	LSB 14	0xFE		
Broadcast Radius	15	0x00	Sets maximum number of hops a broadcast transmission can occur. If set to 0, the broadcast radius will be set to the maximum hops value.	
Frame-specific Data	Options		Bitfield of supported transmission options. Supported values include the following: 0x01 - Disable retries and route repair 0x20 - Enable APS encryption (if EE=1) 0x40 - Use the extended transmission timeout Enabling APS encryption presumes the source and destination have been authenticated. It also decreases the maximum number of RF payload bytes by 4 (below the value reported by NP). The extended transmission timeout is needed when addressing sleeping end devices. It also increases the retry interval between retries to compensate for end device polling. See <a href="#">Transmission timeouts</a> on page 77 for a description. Unused bits must be set to 0.	
		16		0x00
	RF Data	17	0x54	Data that is sent to the destination device
		18	0x78	
		19	0x44	
		20	0x61	
		21	0x74	
		22	0x61	
23		0x30		
24	0x41			
Checksum	25	0x13	0xFF - the 8 bit sum of bytes from offset 3 to this byte.	





↳ Analyse d'une trame :

7E 00 12 10 00 00 13 A2 00 40 C4 58 83 FF FE 00 00 55 66 77 88 A4

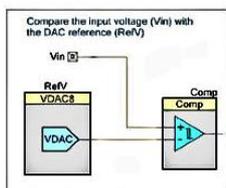


↳ l'adressage utilisé est en 64 bits l'adresse sur 16 bits n'est donc pas utilisée d'où le code FF FE

↳ les deux octets suivants sont à 00

↳ le message utile est la suite 55 66 77 88





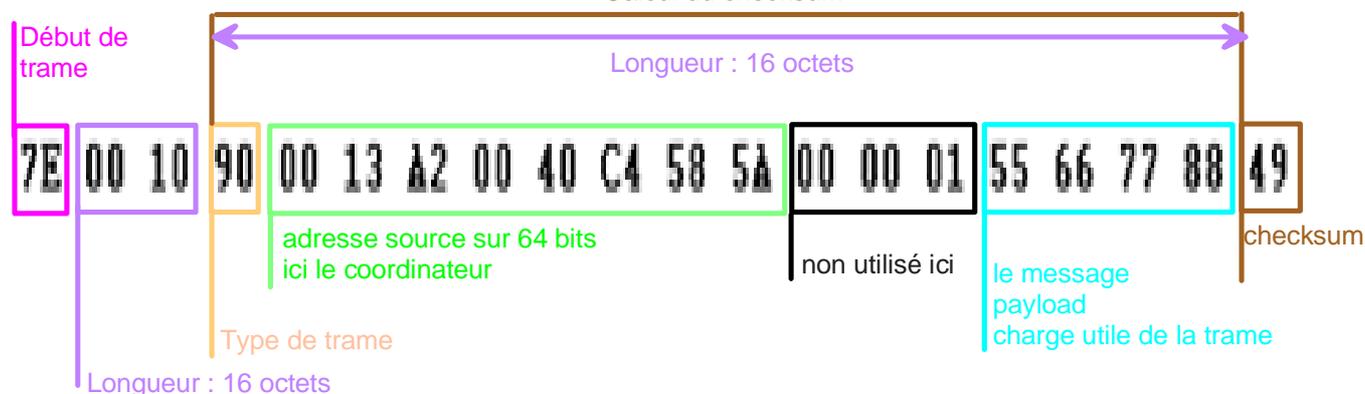
## ZigBee requête de réception de paquet 0x90

Frame Fields	Offset	Example	Description				
Start Delimiter	0	0x7E					
Length	MSB 1	0x00	Number of bytes between the length and the checksum				
	LSB 2	0x11					
	Frame Type	3	0x90				
	64-bit Source Address	MSB 4	0x00	64-bit address of sender. Set to 0xFFFFFFFFFFFFFFFF (unknown 64-bit address) if the sender's 64-bit address is unknown.			
		5	0x13				
		6	0xA2				
		7	0x00				
		8	0x40				
		9	0x52				
		10	0x2B				
	16-bit Source Network Address	MSB 12	0x7D	16-bit address of sender			
LSB 13		0x84					
Receive Options	14	0x01	0x01 - Packet Acknowledged 0x02 - Packet was a broadcast packet 0x20 - Packet encrypted with APS encryption 0x40 - Packet was sent from an end device (if known)  <b>Note</b> Option values can be combined. For example, a 0x40 and a 0x01 will show as a 0x41. Other possible values 0x21, 0x22, 0x41, 0x42, 0x60, 0x61, 0x62.				
				Received Data	15	0x52	Received RF data
					16	0x78	
					17	0x44	
					18	0x61	
					19	0x74	
20	0x61						
Checksum	21	0x0D	0xFF - the 8 bit sum of bytes from offset 3 to this byte.				

### ↳ analyse d'une trame

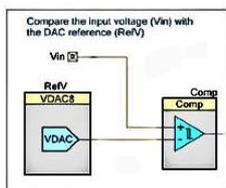
7E 00 10 90 00 13 A2 00 40 C4 58 5A 00 00 01 55 66 77 88 49

Calcul du checksum



↳ cette trame est délivrée par le module XBee lorsqu'il reçoit les données par la trame 0x10. Il indique de ce fait l'adresse du module d'origine qui a envoyé les données, ici le coordinator.





## ZigBee indicateur d'IO distant 0x92

Frame Fields	Offset	Example	Description
Start Delimiter	0	0x7E	
Length	MSB 1	0x00	Number of bytes between the length and the checksum
	LSB 2	0x14	
Frame-specific Data	3	0x92	Frame Type
	MSB 4	0x00	64-bit address of sender
	5	0x13	
	6	0xA2	
	7	0x00	
	8	0x40	
	9	0x52	
	10	0x2B	
LSB 11	0xAA		
16-bit Source Network Address	MSB 12	0x7D	16-bit address of sender
	LSB 13	0x84	
Receive Options	14	0x01	0x01 - Packet Acknowledged 0x02 - Packet was a broadcast packet
Number of Samples	15	0x01	Number of sample sets included in the payload (Always set to 1)
Digital Channel Mask*	16	0x00	Bitmask field that indicates which digital IO lines on the remote have sampling enabled (if any)
	17	0x1C	
Analog Channel Mask**	18	0x02	Bitmask field that indicates which analog IO lines on the remote have sampling enabled (if any)
Digital Samples (if included)	19	0x00	If the sample set includes any digital IO lines (Digital Channel Mask > 0), these two bytes contain samples for all enabled digital IO lines. DIO lines that do not have sampling enabled return 0. Bits in these 2 bytes map the same as they do in the Digital Channels Mask field
	20	0x14	
Analog Sample	21	0x02	If the sample set includes any analog input lines (Analog Channel Mask > 0), each enabled analog input returns a 2-byte value indicating the A/D measurement of that input. Analog samples are ordered sequentially from AD0/DIO0 to AD3/DIO3, to the supply voltage
	22	0x25	
Checksum	23	0xF5	0xFF - the 8 bit sum of bytes from offset 3 to this byte

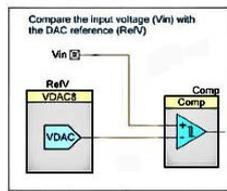
## Identification de l'entrée tout ou rien IO transmise 0x16 et 0x17:

N/A	N/A	N/A	CD/DIC12	PWM/DI011	RSSI/DIO10	N/A	N/A
CTS/DI07	RTS/DI06	ASSOC/DI05	DI04	AD3/DI03	AD2/DI02	AD1/DIO1	AD0/DIO0

## Identification de l'entrée analogique transmise 0x18

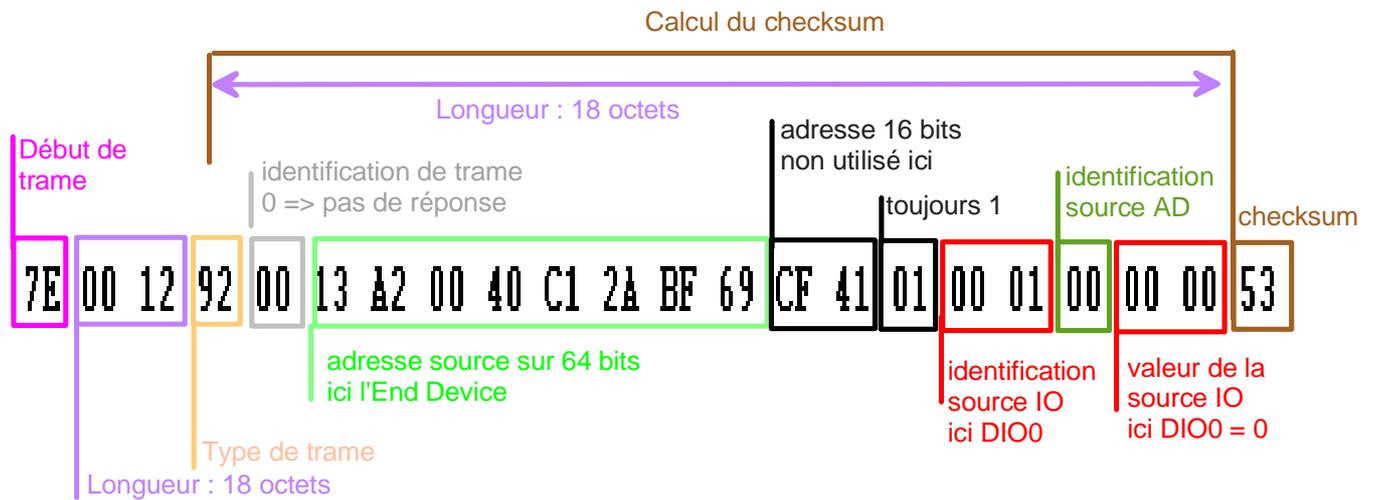
Supply Voltage	N/A	N/A	N/A	AD3	AD2	AD1	AD0
----------------	-----	-----	-----	-----	-----	-----	-----





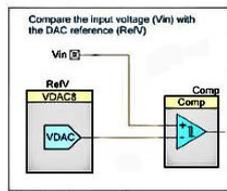
↳ analyse d'une trame

7E 00 12 92 00 13 A2 00 40 C1 2A BF 69 CF 41 01 00 01 00 00 00 53

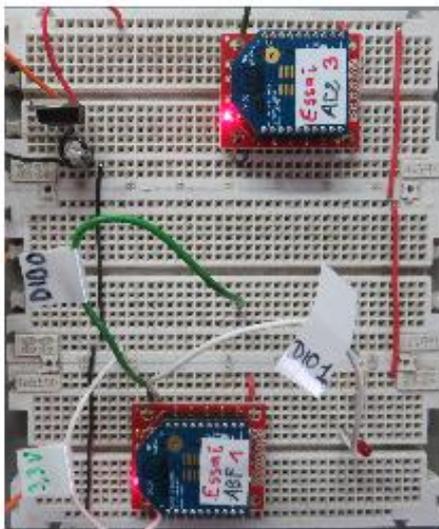
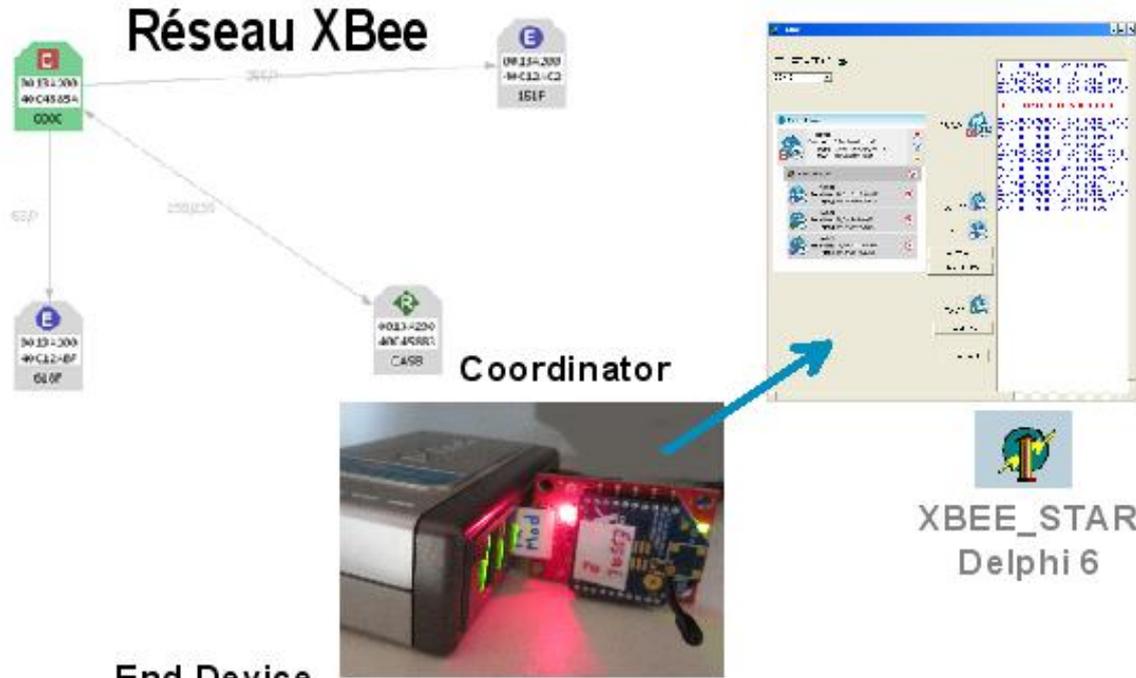


↳ il n'y a pas d'octet reçu pour les valeurs analogiques car aucune n'a été définie identification source AD = 00.

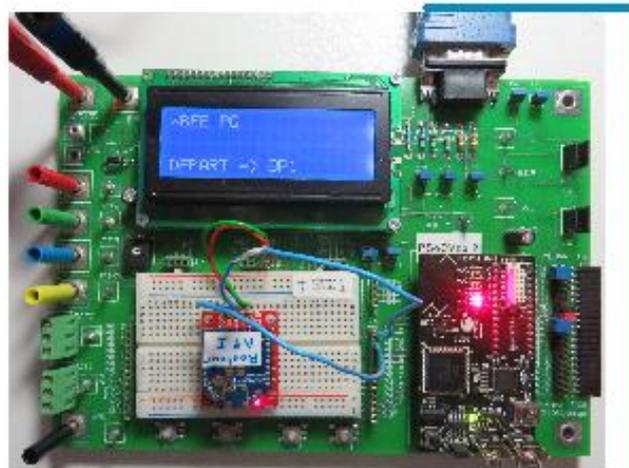




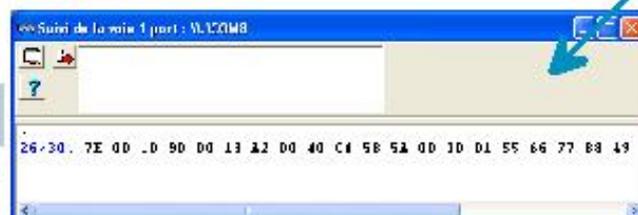
## Schéma général du réseau XBee



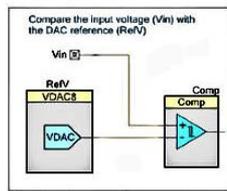
**End Device**



**Router**



# PSoC PSoC



Indique un document ressource



Retour au sommaire



Retour à la page courante