X-Bee documentation

Created by:

Ranojoy Barua (CST Dept. IIEST Shibpur)

Sourav Das (CST Dept. IIEST Shibpur)

Introduction

It is a device for radio communication. This follows the IEEE 802.15.4 protocol and have a point-to-point baud rate of 250 Kbit/sec. They need a power source of 3.3V but most of its boards can be powered by 5V. X-Bee can be easily combined with Arduino or any other micro-controller. This module have 20 pins which include power pin, ground pin, RX, TX etc. The X-Bee generally works in two modes i.e. AT and API mode.

Difference between AT and API modes are

AT	API
1) Fastest way to transmit data between two X-Bees.	1) More complicated than AT mode.
2) Can handle communications between only two active X-Bees at a time no matter how many X-Bee are there in the network.	2) This mode of communication can handle the communication of multiple X-Bee devices at a time in the network.
3) It is synchronous with "Transparent" mode.	3) API mode stands for "Application Programming Interface".
4) In this mode data is sent to the remote X-Bee module identified by the Destination Address in flash memory of the X-Bee.	4) Data must be formatted in frames or packets with Destination information and payload.
5) No packet information is sent simple serial communication takes place.	5) Information is transmitted in packets so more complicated

6) As no packets are created the destination address and type are fixed as mentioned in the flash memory.	communication can be handled in the network. 6) As packets are there so destination is not fixed as in AT mode. Destination is specified on the packet.

Method to use X-Bee

Step -- 1) Connect the X-Bee to the connector module.

Step – 2) Configure the X-Bee as required.

Step – 3) Use the X-Bee as you need.

Software used for communication

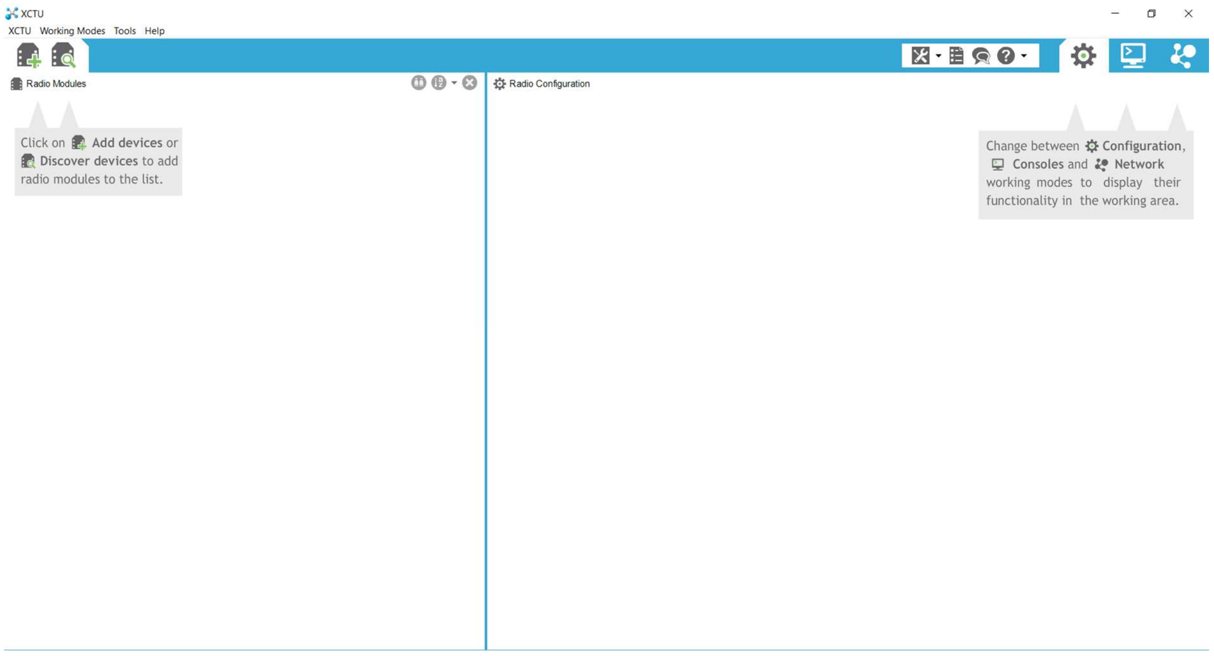
XCTU

PUTTY

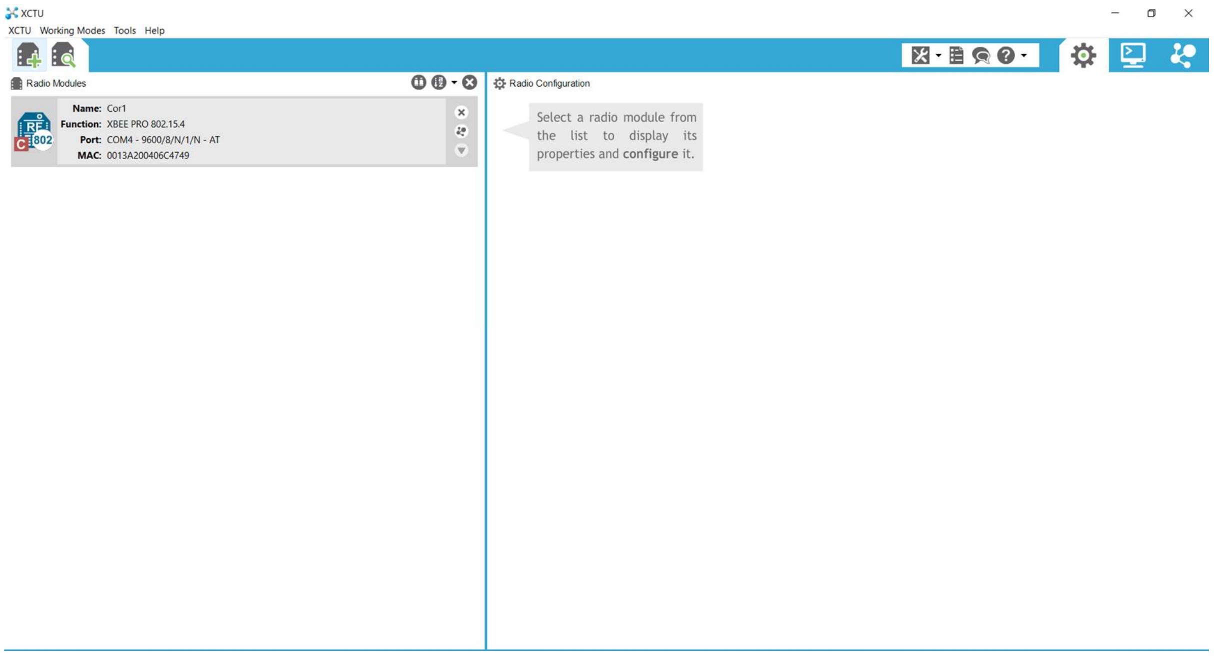
ARDUINO IDE

Configuring the X-Bee using XCTU

Step – 1) Open XCTU


XCTU
XCTU Working Modes Tools Help
Radio Modules
Radio Configuration
Click on Add devices or Discover devices to add radio modules to the list.
Change between Configuration, Consoles and Network working modes to display their functionality in the working area.

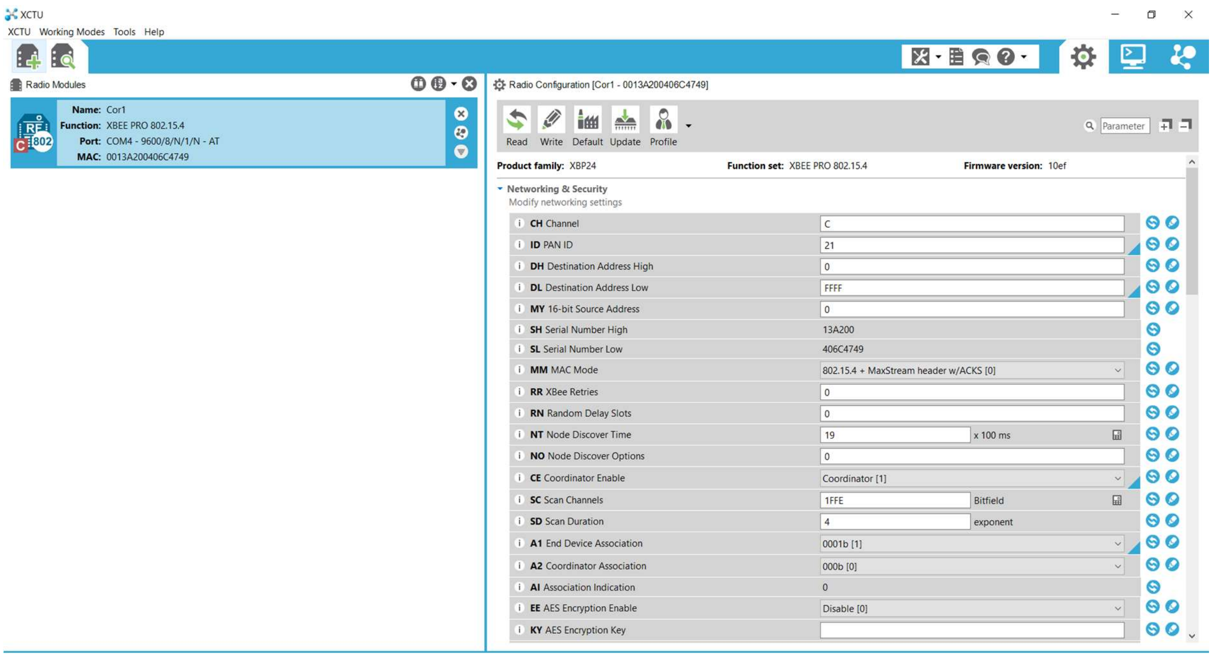
Step – 2) Connect the X-Bee module to the computer via the connector and click the Discover Device and add it to the terminal.


XCTU
XCTU Working Modes Tools Help
Radio Modules
Radio Configuration
Select a radio module from the list to display its properties and configure it.

Name: Cor1
Function: XBEE PRO 802.15.4
Port: COM4 - 9600/8/N/1/N - AT
MAC: 0013A200406C4749

Notice that the device is already configured here but we will re-configure this device. Click on the device to open the device.

Step – 3) Open the device and start configuring the device.



Name: Cor1
Function: XBEE PRO 802.15.4
Port: COM4 - 9600/8/N/1/N - AT
MAC: 0013A200406C4749

Radio Configuration [Cor1 - 0013A200406C4749]

Product family: XBP24 Function set: XBEE PRO 802.15.4 Firmware version: 10ef

Networking & Security
Modify networking settings

CH Channel	C	
ID PAN ID	21	
DH Destination Address High	0	
DL Destination Address Low	FFFF	
MY 16-bit Source Address	0	
SH Serial Number High	13A200	
SL Serial Number Low	406C4749	
MM MAC Mode	802.15.4 + MaxStream header w/ACKS [0]	
RR XBee Retries	0	
RN Random Delay Slots	0	
NT Node Discover Time	19	x 100 ms
NO Node Discover Options	0	
CE Coordinator Enable	Coordinator [1]	
SC Scan Channels	1FFE	Bitfield
SD Scan Duration	4	exponent
A1 End Device Association	0001b [1]	
A2 Coordinator Association	000b [0]	
AI Association Indication	0	
EE AES Encryption Enable	Disable [0]	
KY AES Encryption Key		

Setting up the co-ordinator:

Select a **communication channel** as the communication will take place via a communication channel. Here we have selected it as '**C**' you can choose any channel as you want.

Select a **PAN (Personal area network) ID** as the X-Bees will create a personal area network.

Set **DH (Destination High address)** to '**0**', **DL (Destination Low address)** to '**FFFF**' and **MY address** to '**0**'.

Serial Number High holds the first 8-bits of the MAC address of the device and **Serial Number Low** holds rest of the 8-bits of the MAC address of the device as shown in the device.

In **CE (Co-ordinator Enable)** select **Coordinator[1]** as we are configuring this X-Bee as co-ordinator.

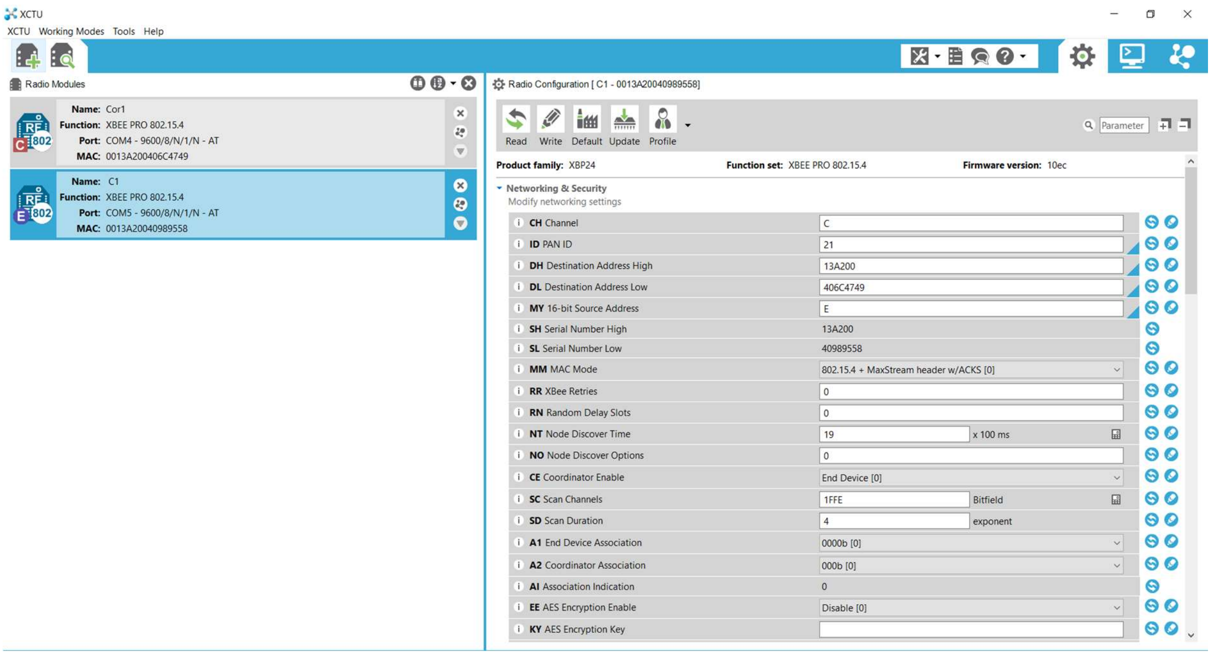
In **EE (Enable AES Encryption)** select **Enable[1]** if you want the communication to be encrypted or else select **Disable[0]**.

In **NE (Node Identifier)** give the name of the device you want here we have used the name '**Cor1**'.

<u>Note:</u> After configuring always write it to the flash memory by clicking the pencil type icon on the right hand side.

Step -4) Configure another X-Bee as end-device.

End device will send data and it will be received by the co-ordinator.



XCTU

XCTU Working Modes Tools Help

Radio Modules

Name: Cor1
Function: XBEE PRO 802.15.4
Port: COM4 - 9600/8/N/1/N - AT
MAC: 0013A200406C4749

Name: C1
Function: XBEE PRO 802.15.4
Port: COM5 - 9600/8/N/1/N - AT
MAC: 0013A20040989558

Radio Configuration [C1 - 0013A20040989558]

Product family: XBP24 **Function set:** XBEE PRO 802.15.4 **Firmware version:** 10ec

Networking & Security
Modify networking settings

CH Channel	C	
ID PAN ID	21	
DH Destination Address High	13A200	
DL Destination Address Low	406C4749	
MY 16-bit Source Address	E	
SH Serial Number High	13A200	
SL Serial Number Low	40989558	
MM MAC Mode	802.15.4 + MaxStream header w/ACKS [0]	
RR XBee Retries	0	
RN Random Delay Slots	0	
NT Node Discover Time	19	x 100 ms
NO Node Discover Options	0	
CE Coordinator Enable	End Device [0]	
SC Scan Channels	1FFE	Bitfield
SD Scan Duration	4	exponent
A1 End Device Association	0000b [0]	
A2 Coordinator Association	000b [0]	
AI Association Indication	0	
EE AES Encryption Enable	Disable [0]	
KY AES Encryption Key		

Add the device to the XCTU and open the device as for the pervious device. Now again configure the device with the end-device configuration.

<u>Configuring as the end-device:</u>

Set **CH (Channel)** and **PAN (Personal Area Network) ID** same as that of the co-ordinator.

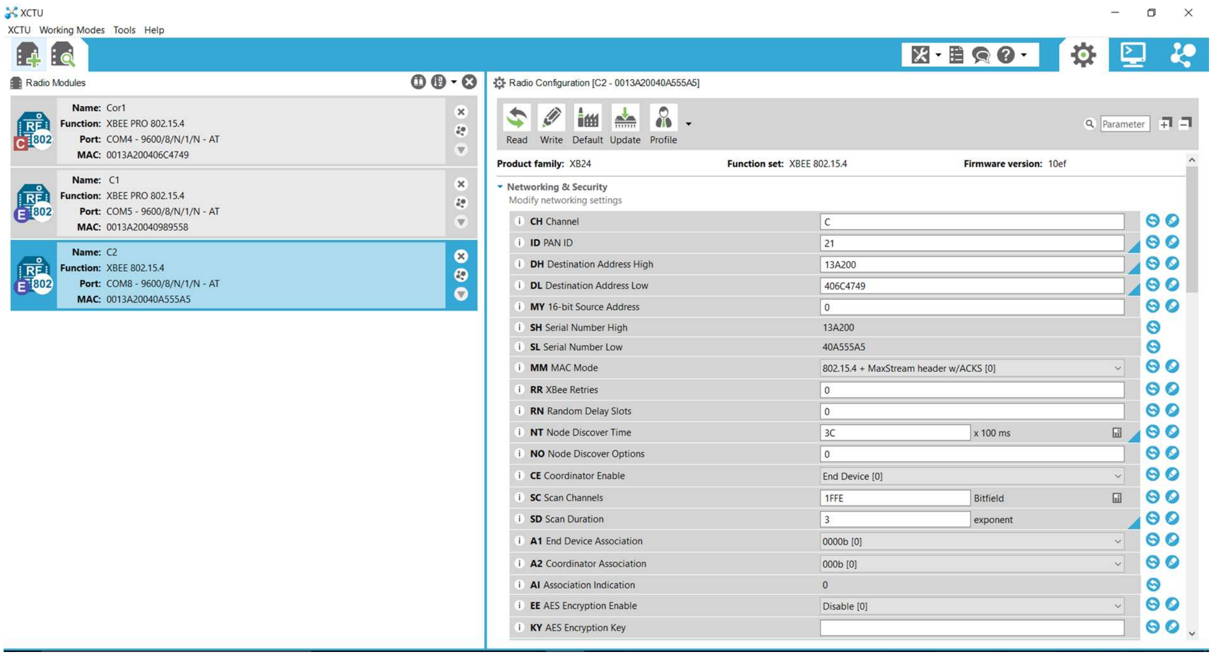
DH (Destination High) holds the **first 8-bits of the MAC address of the co-ordinator** and **DL (Destination Low)** will hold the **last 8-bits of the MAC address of the co-ordinator**.

In **CE (Co-ordinator enable)** select **End Device[0]** to set the X-Bee as the end device.

In **NI (Node Identifier)** set it as the name you want to identify the X-Bee for us we selected it as C1.

Step – 5) Add another X-Bee and configure it as another end-device as that of in Step- 4.

<u>Note</u>: Refresh ports to find the new devices.


XCTU
XCTU Working Modes Tools Help
Radio Modules
Radio Configuration [C2 - 0013A20040A555A5]
Read Write Default Update Profile
Parameter

Name:	Cor1
Function:	XBEE PRO 802.15.4
Port:	COM4 - 9600/8/N/1/N - AT
MAC:	0013A200406C4749

Name:	C1
Function:	XBEE PRO 802.15.4
Port:	COM5 - 9600/8/N/1/N - AT
MAC:	0013A20040989558

Name:	C2
Function:	XBEE 802.15.4
Port:	COM8 - 9600/8/N/1/N - AT
MAC:	0013A20040A555A5

Product family: XB24 **Function set:** XBEE 802.15.4 **Firmware version:** 10ef

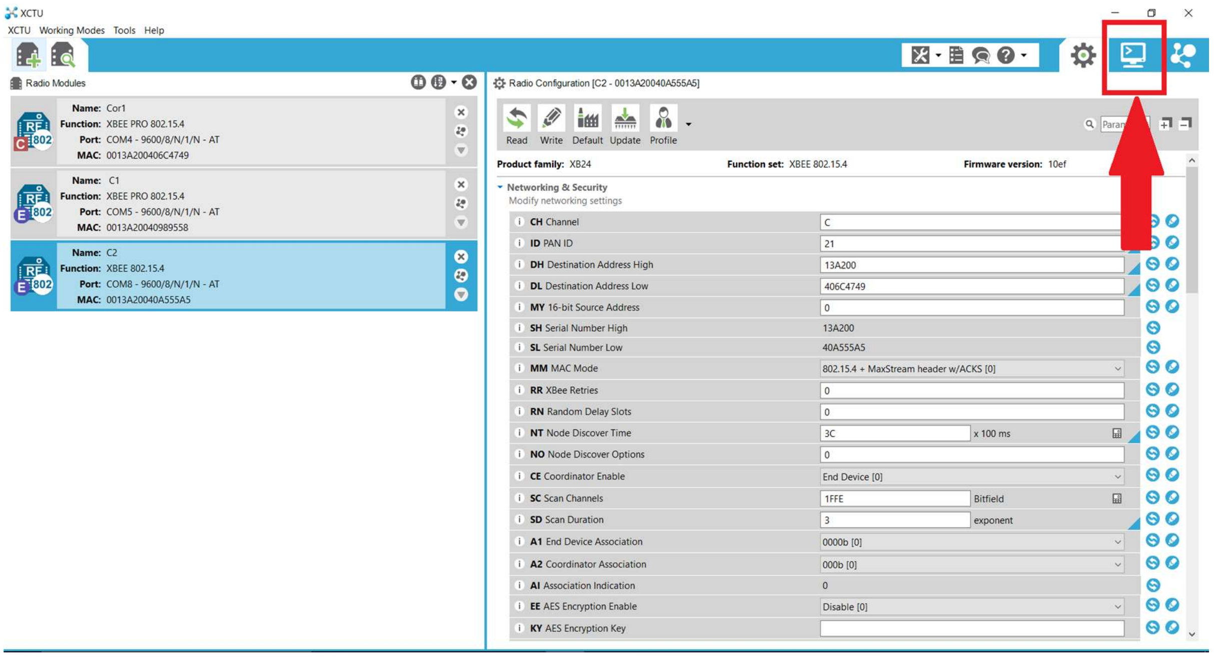
Networking & Security
Modify networking settings

CH Channel	C
ID PAN ID	21
DH Destination Address High	13A200
DL Destination Address Low	406C4749
MY 16-bit Source Address	0
SH Serial Number High	13A200
SL Serial Number Low	40A555A5
MM MAC Mode	802.15.4 + MaxStream header w/ACKS [0]
RR XBee Retries	0
RN Random Delay Slots	0
NT Node Discover Time	3C x 100 ms
NO Node Discover Options	0
CE Coordinator Enable	End Device [0]
SC Scan Channels	1FFE Bitfield
SD Scan Duration	3 exponent
A1 End Device Association	0000b [0]
A2 Coordinator Association	000b [0]
AI Association Indication	0
EE AES Encryption Enable	Disable [0]
KY AES Encryption Key	

Configure this X-Bee as done for the pervious X-Bee.

Step – 6) Communication: For communication we will be sending data from end-devices named as C1 and C2 to the co-ordinator named Cor1 one at a time.

Click on the sign as shown


XCTU
XCTU Working Modes Tools Help
Radio Modules
Radio Configuration [C2 - 0013A20040A555A5]
Read Write Default Update Profile
Paran

Name:	Cor1
Function:	XBEE PRO 802.15.4
Port:	COM4 - 9600/8/N/1/N - AT
MAC:	0013A200406C4749

Name:	C1
Function:	XBEE PRO 802.15.4
Port:	COM5 - 9600/8/N/1/N - AT
MAC:	0013A20040989558

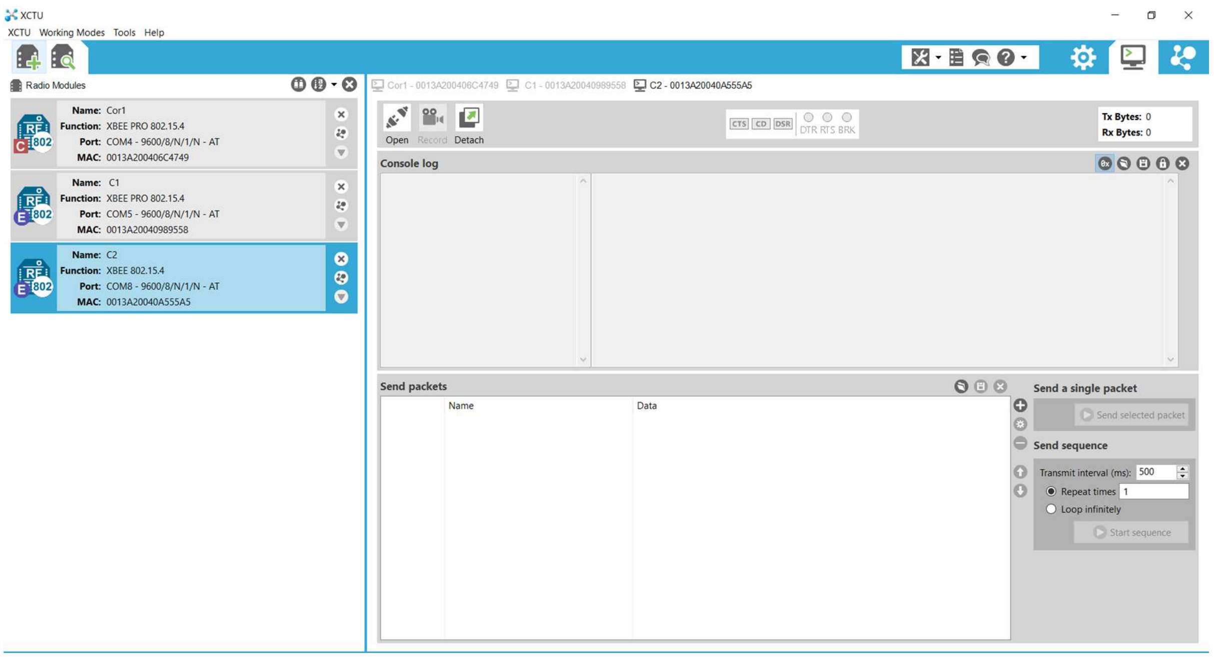
Name:	C2
Function:	XBEE 802.15.4
Port:	COM8 - 9600/8/N/1/N - AT
MAC:	0013A20040A555A5

Product family: XB24 **Function set:** XBEE 802.15.4 **Firmware version:** 10ef

Networking & Security
Modify networking settings

CH Channel	C
ID PAN ID	21
DH Destination Address High	13A200
DL Destination Address Low	406C4749
MY 16-bit Source Address	0
SH Serial Number High	13A200
SL Serial Number Low	40A555A5
MM MAC Mode	802.15.4 + MaxStream header w/ACKS [0]
RR XBee Retries	0
RN Random Delay Slots	0
NT Node Discover Time	3C x 100 ms
NO Node Discover Options	0
CE Coordinator Enable	End Device [0]
SC Scan Channels	1FFE Bitfield
SD Scan Duration	3 exponent
A1 End Device Association	0000b [0]
A2 Coordinator Association	000b [0]
AI Association Indication	0
EE AES Encryption Enable	Disable [0]
KY AES Encryption Key	

This page will open



Name: Cor1
Function: XBEE PRO 802.15.4
Port: COM4 - 9600/8/N/1/N - AT
MAC: 0013A200406C4749

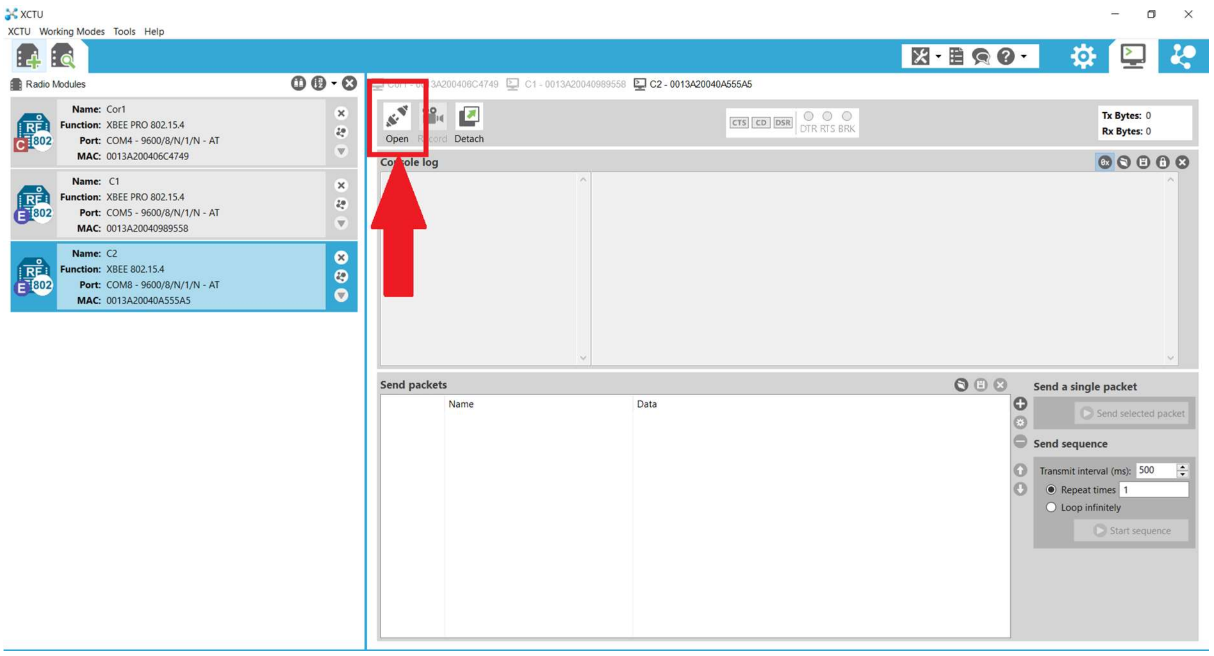
Name: C1
Function: XBEE PRO 802.15.4
Port: COM5 - 9600/8/N/1/N - AT
MAC: 0013A20040989558

Name: C2
Function: XBEE 802.15.4
Port: COM8 - 9600/8/N/1/N - AT
MAC: 0013A20040A555A5

Tx Bytes: 0
Rx Bytes: 0

Transmit interval (ms): 500
[✓ Repeat times] 1
Loop infinitely

Click on this button



Name: Cor1
Function: XBEE PRO 802.15.4
Port: COM4 - 9600/8/N/1/N - AT
MAC: 0013A200406C4749

Name: C1
Function: XBEE PRO 802.15.4
Port: COM5 - 9600/8/N/1/N - AT
MAC: 0013A20040989558

Name: C2
Function: XBEE 802.15.4
Port: COM8 - 9600/8/N/1/N - AT
MAC: 0013A20040A555A5

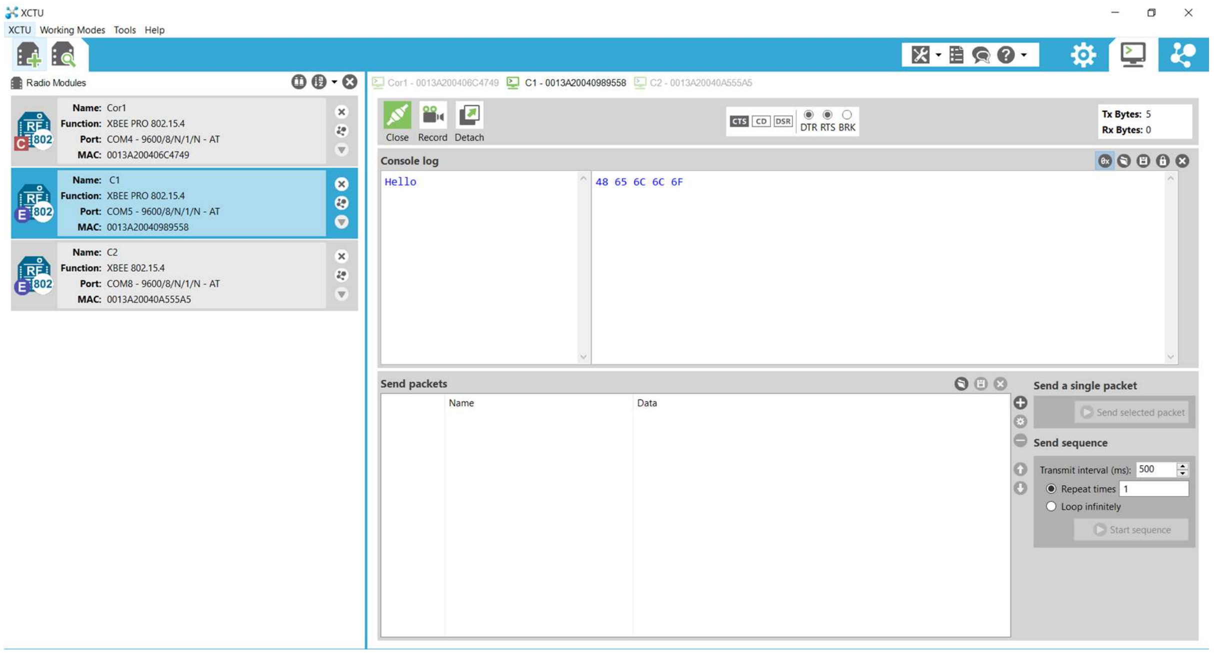
Tx Bytes: 0
Rx Bytes: 0

Transmit interval (ms): 500
[✓ Repeat times] 1
Loop infinitely

Do this for all the X-Bees and then in the console log of the end-devices C1 and C2 type the message you want to send to the co-ordinator.

From C1 we send the message “Hello” and from C2 we entered the message “World”.

From C1:


XCTU

XCTU Working Modes Tools Help

Radio Modules

Name: Cor1
Function: XBEE PRO 802.15.4
Port: COM4 - 9600/8/N/1/N - AT
MAC: 0013A200406C4749

Name: C1
Function: XBEE PRO 802.15.4
Port: COM5 - 9600/8/N/1/N - AT
MAC: 0013A20040989558

Name: C2
Function: XBEE 802.15.4
Port: COM8 - 9600/8/N/1/N - AT
MAC: 0013A20040A555A5

Cor1 - 0013A200406C4749 | C1 - 0013A20040989558 | C2 - 0013A20040A555A5

Close Record Detach | CTS CD DSR | DTR RTS BRK

Tx Bytes: 5
Rx Bytes: 0

Console log

```
Hello        48 65 6C 6C 6F
```

Send packets

Name | Data

Send a single packet

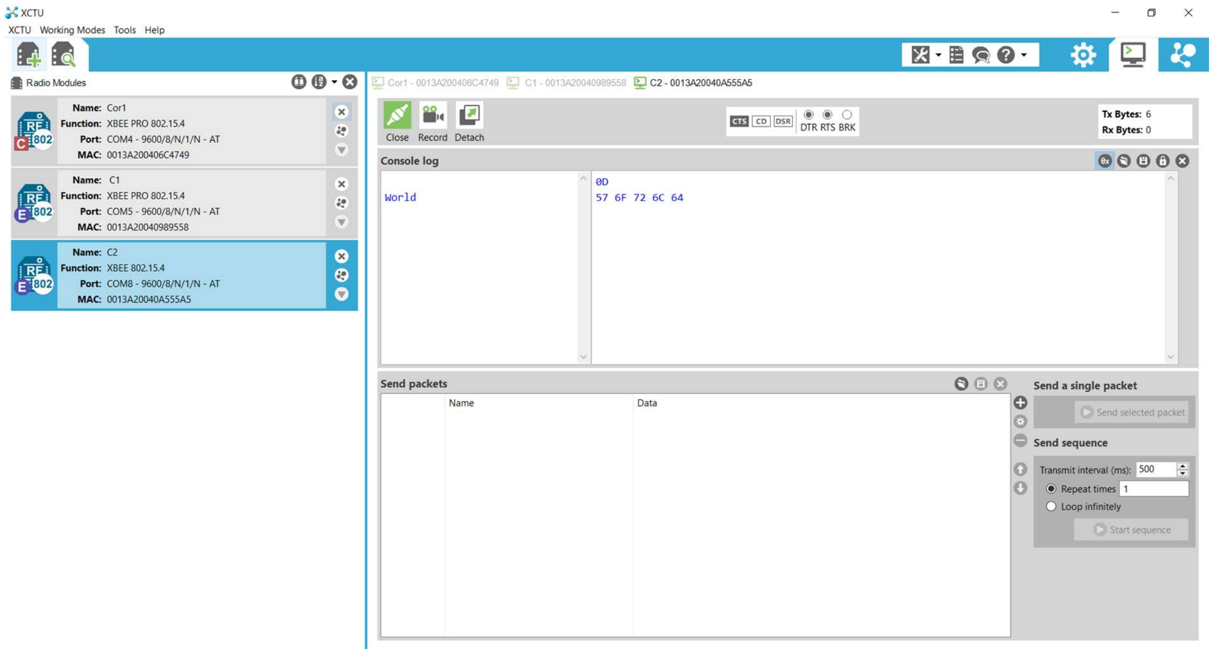
Send selected packet

Send sequence

Transmit interval (ms): 500
[✓ Repeat times] 1
Loop infinitely

Start sequence

From C2:


XCTU

XCTU Working Modes Tools Help

Radio Modules

Name: Cor1
Function: XBEE PRO 802.15.4
Port: COM4 - 9600/8/N/1/N - AT
MAC: 0013A200406C4749

Name: C1
Function: XBEE PRO 802.15.4
Port: COM5 - 9600/8/N/1/N - AT
MAC: 0013A20040989558

Name: C2
Function: XBEE 802.15.4
Port: COM8 - 9600/8/N/1/N - AT
MAC: 0013A20040A555A5

Cor1 - 0013A200406C4749 | C1 - 0013A20040989558 | C2 - 0013A20040A555A5

Close Record Detach | CTS CD DSR | DTR RTS BRK

Tx Bytes: 6
Rx Bytes: 0

Console log

```
             0D
World        57 6F 72 6C 64
```

Send packets

Name | Data

Send a single packet

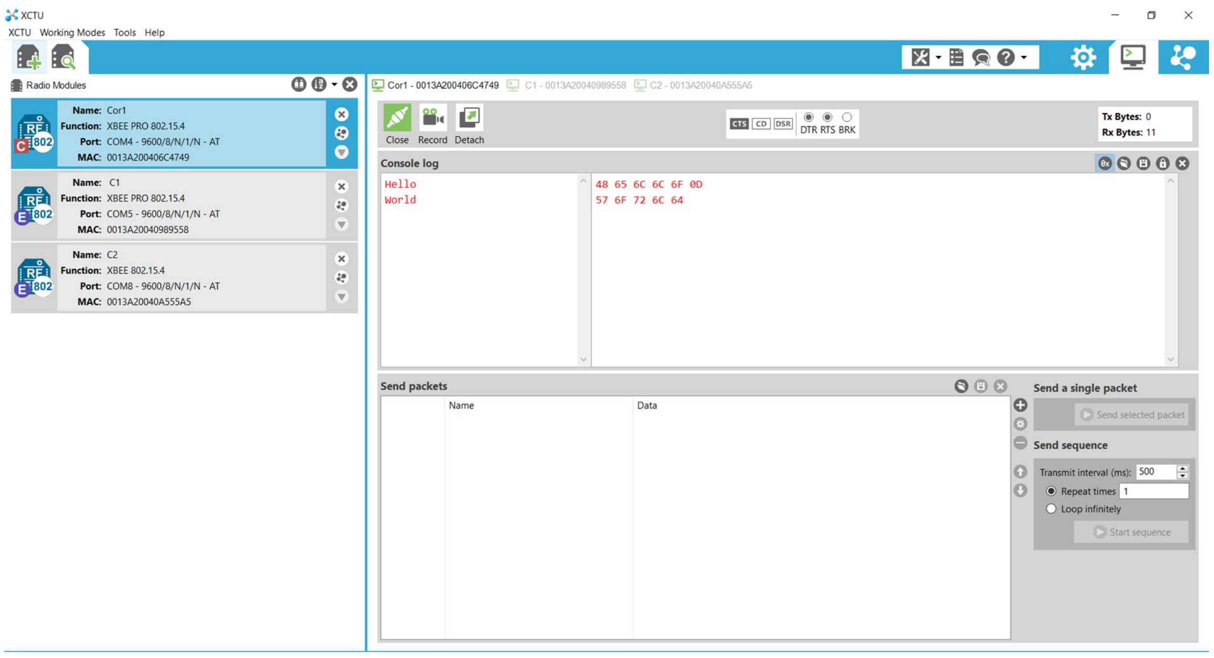
Send selected packet

Send sequence

Transmit interval (ms): 500
[✓ Repeat times] 1
Loop infinitely

Start sequence

The message received in the co-ordinator is:



Name: Cor1
Function: XBEE PRO 802.15.4
Port: COM4 - 9600/8/N/1/N - AT
MAC: 0013A200406C4749

Name: C1
Function: XBEE PRO 802.15.4
Port: COM5 - 9600/8/N/1/N - AT
MAC: 0013A20040989558

Name: C2
Function: XBEE 802.15.4
Port: COM8 - 9600/8/N/1/N - AT
MAC: 0013A20040A555A5

Cor1 - 0013A200406C4749 C1 - 0013A20040989558 C2 - 0013A20040A555A5

Tx Bytes: 0
Rx Bytes: 11

Console log

```
Hello    48 65 6C 6C 6F 0D
World    57 6F 72 6C 64
```

Name	Data

Transmit interval (ms): 500
Repeat times 1
Loop infinitely

So in this way you can send data from multiple X-Bees to one X-Bee.

X-Bee communication via Arduino

Component used

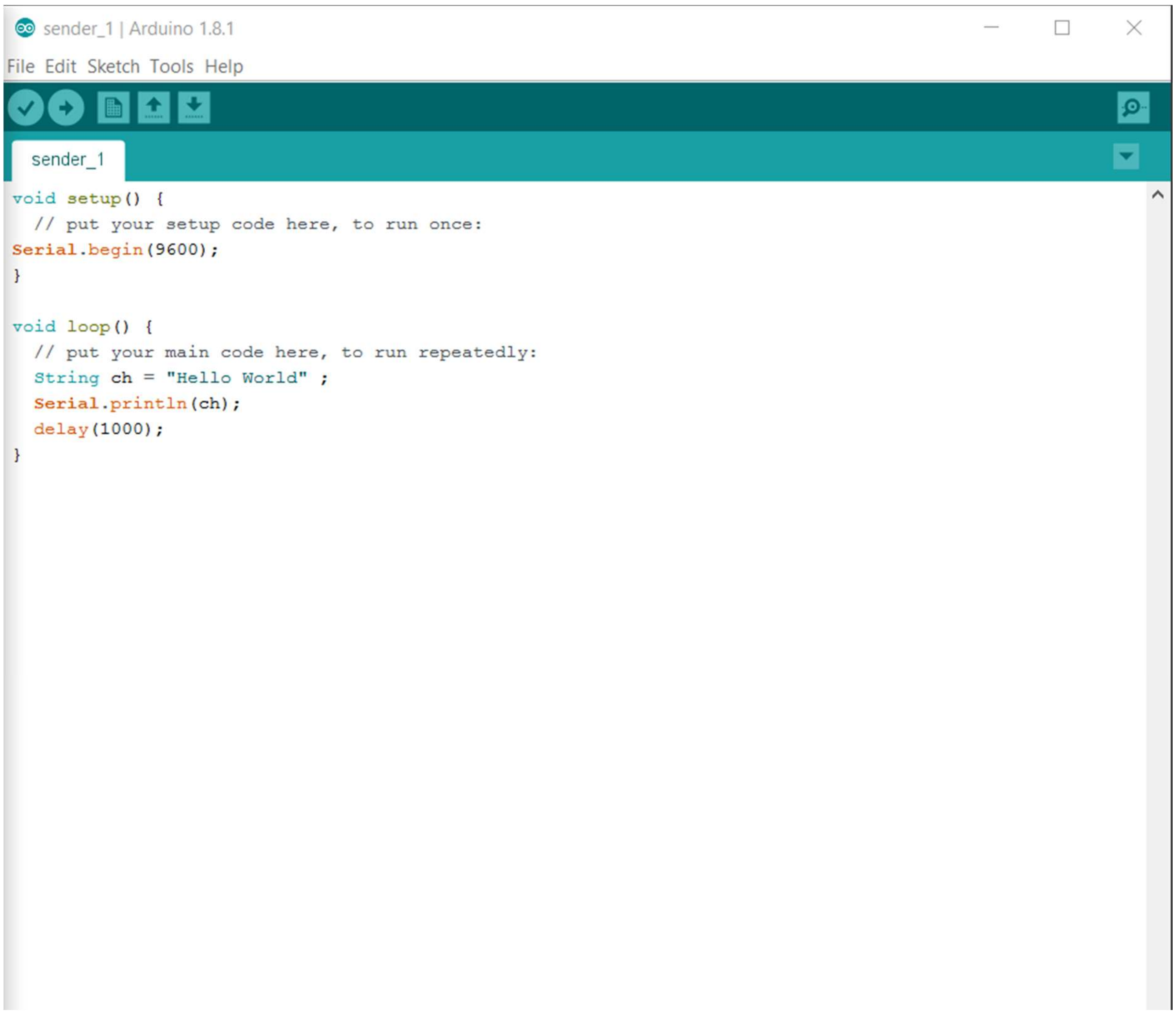
1) Arduino (2 pcs.)
2) X-Bee shield (2 pcs.)
3) X-Bee (2 pcs.)

Method

1) Use the per-configured X-Bees for the communication using the Arduino.
2) You don't need to again configure the X-Bees using Arduino as they are already configured using the XCTU.
3) Connect the X-Bee on the shield.
4) Place the shield on the Arduino and connect the Arduino to the Arduino IDE via a USB cable.
5) NOTE: Connect the X-Bee, X-Bee Shield and the Arduino with proper pin configuration.
6) Now switch the shield to the USB mode.

<u>Code for Sender Side</u>

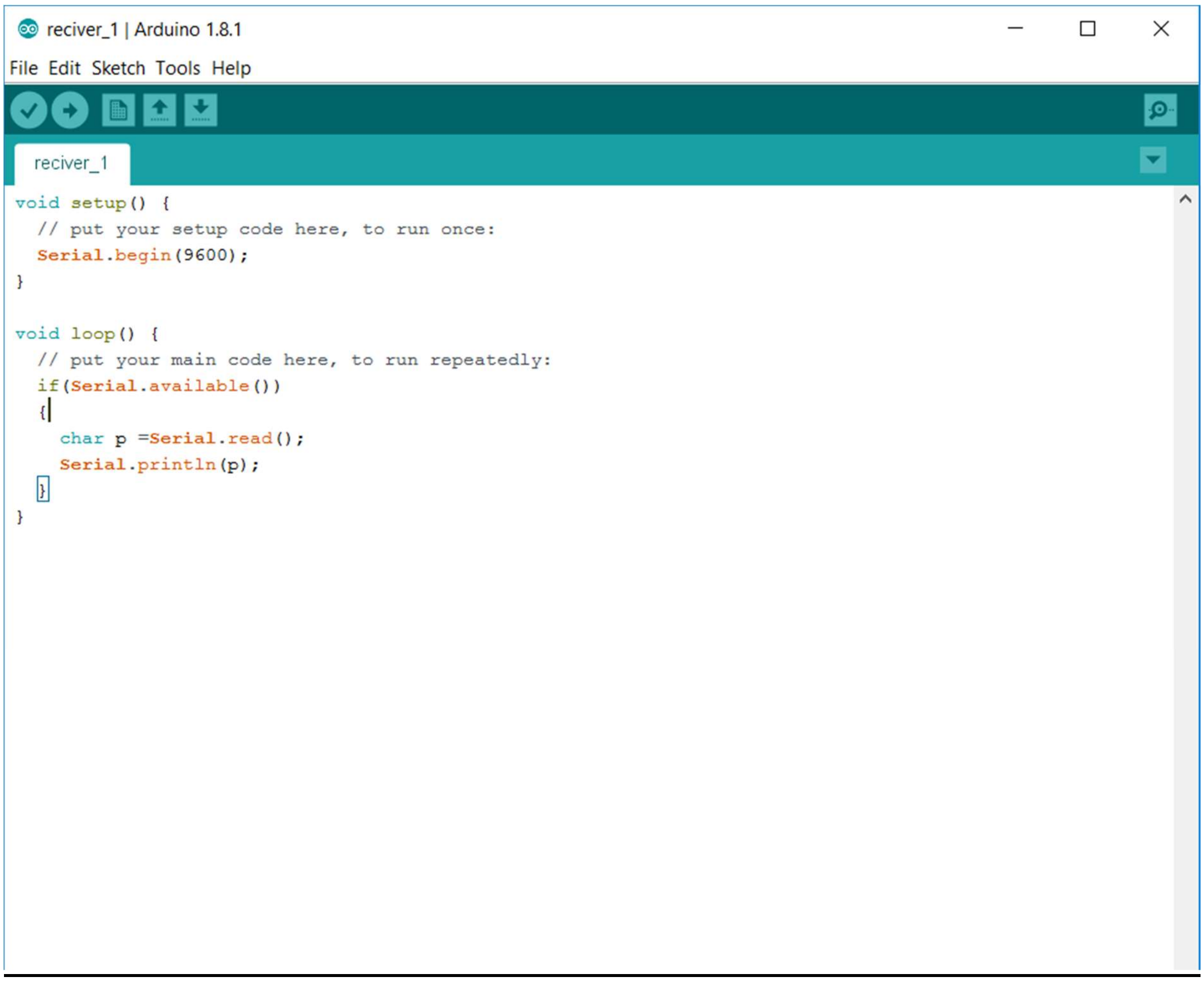
```
void setup() {
Serial.begin(9600);              // starting the serial communication
 }


void loop() {
 String ch = "Hello World" ;             // the string that you want to transmit in case of
 Serial.println(ch);                     // sensors it will be data from the sensors
 delay(1000);                            // applying a delay of 1000 milli-second.
}
```



```
void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  String ch = "Hello World" ;
  Serial.println(ch);
  delay(1000);
}
```

Code for receiver side

```
void setup() {
  Serial.begin(9600);                    // setting up the serial communication
}


void loop() {
  if(Serial.available())                 // checking if something is available on serial monitor
  {
    char p =Serial.read();               // reading the data available in serial monitor and
    Serial.println(p);                   // assigning it in a 'char' type variable and printing it
  }                                      // on serial monitor
}
```

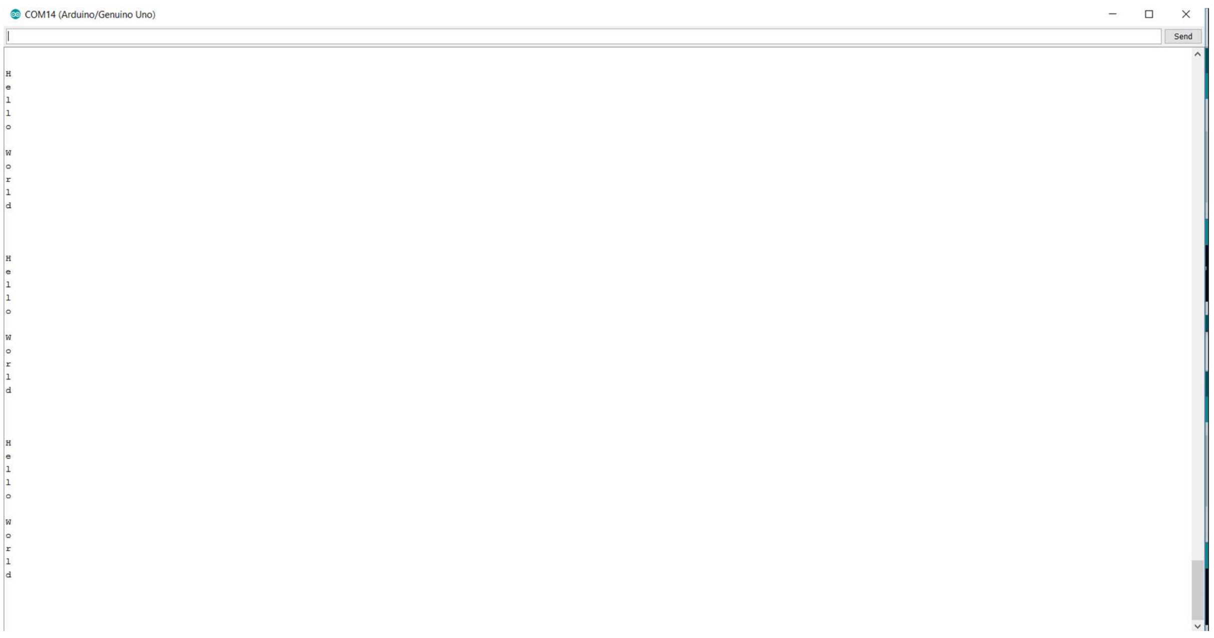


```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  if(Serial.available())
  {
    char p =Serial.read();
    Serial.println(p);
  }
}
```

Uploading

1) Now switch to USB mode.
2) Compile the programmes and upload the code.
3) Switch to X-Bee mode and open the serial monitor of the receiver to see the message that the sender have sent.

Serial Monitor of the Receiver

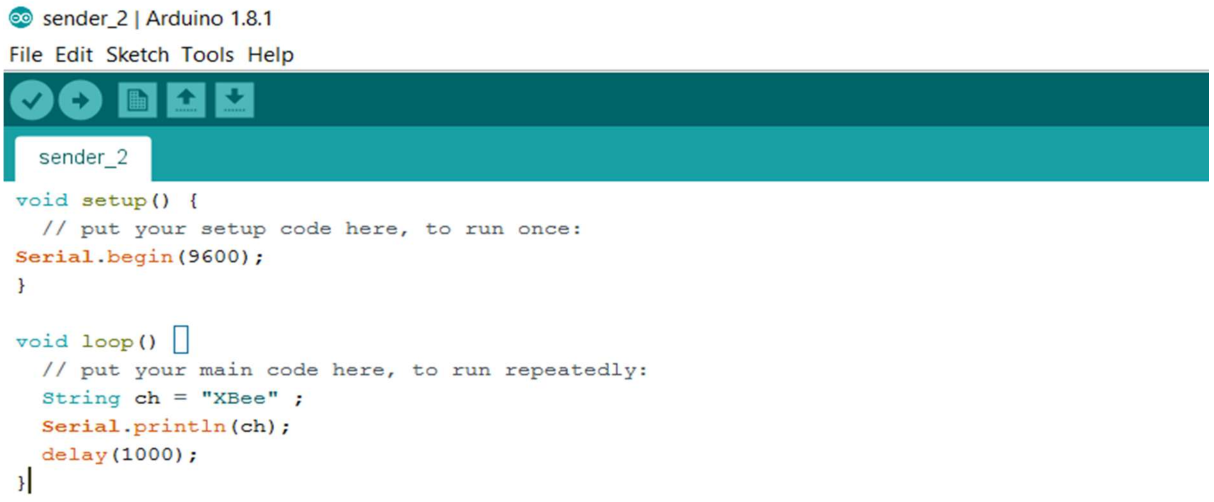


```
H
e
l
l
o

W
o
r
l
d


H
e
l
l
o

W
o
r
l
d


H
e
l
l
o

W
o
r
l
d
```

Now we add another Arduino in sender end to make the network more complicated i.e. there are 2 senders and one receiver in the network.

Code for sender – 2

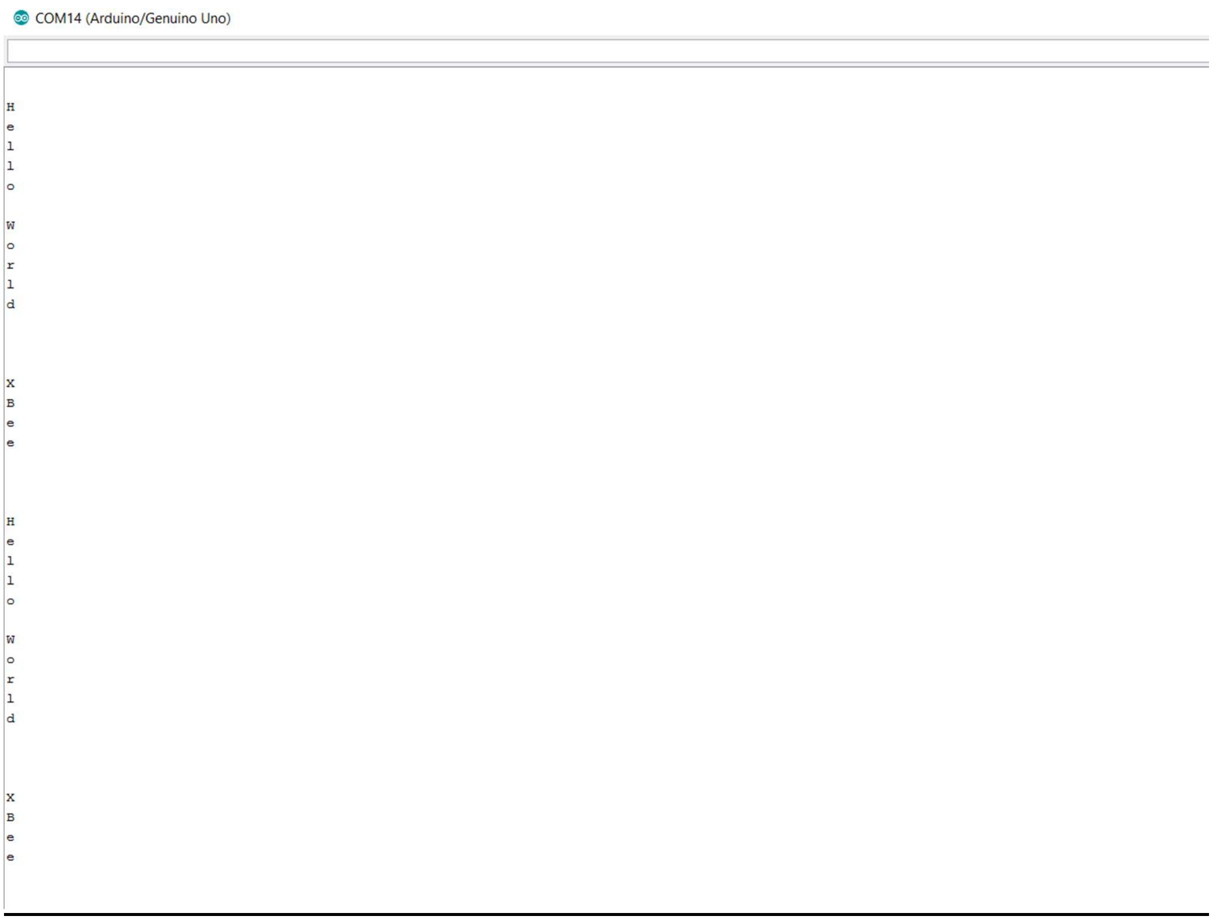
```
void setup() {
Serial.begin(9600);                // starting the serial communication
 }


void loop() {
 String ch = "Xbee" ;              // the string that you want to transmit in case of
 Serial.println(ch);                     // sensors it will be data from the sensors
 delay(1000);                            // applying a delay of 1000 milli-second.
 }
```



```
void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  String ch = "XBee" ;
  Serial.println(ch);
  delay(1000);
}
```

Now there are two X-Bee one sending “Hello World” and the other is sending “Xbee”.

Out-put in the serial monitor of the receiver


COM14 (Arduino/Genuino Uno)

```
H
e
l
l
o

W
o
r
l
d



X
B
e
e



H
e
l
l
o

W
o
r
l
d



X
B
e
e
```

So, as we can see that “Hello World” and “Xbee” comes separately so the communication is successful. But this communication will be more reliable if API mode is used for communication.

THANK-YOU