

# **Rete di sensori**

**Ivancich Stefano**

**1 Giugno 2015**

**Istituto tecnico C. Zuccante - Mestre**

## SOMMARIO

Di seguito s'illustra un progetto di un sistema volto alla rilevazione di temperatura di più punti non collegati fisicamente tra loro. Questi comunicano tramite il protocollo ZIGBEE utilizzato dai moduli wireless xBee.

Questo sistema integra al suo interno moduli xBee Serie 2 e sensori di temperatura LM35.

Pur essendo un progetto artigianale e scolastico, non è molto ingombrante (7,7 cm di larghezza x 6,5 cm di lunghezza x 4,8 cm di altezza).

Sono stati realizzati mediante il software SolidWorks e una stampante 3D i contenitori per ogni sensore.



# INDICE

<b>SOMMARIO</b> .....	2
<b>INDICE</b> .....	3
<b>INTRODUZIONE</b> .....	1
<b>SCHEMA A BLOCCHI</b> .....	2
<b>SCHEMA ELETTRICO</b> .....	2
<b>RETE ZIGBEE</b> .....	4
<b>STRUTTURA SENSORE</b> .....	5
<b>ALIMENTAZIONE</b> .....	8
<b>CIRCUITO DI REGOLAZIONE DEI 3.3V LM317</b> .....	8
<b>LED DI POWER-ON</b> .....	9
<b>XBEE SERIE 2</b> .....	10
<b>CONSUMI</b> .....	11
<b>INSTALLAZIONE e MANUTENZIONE DELLA RETE</b> .....	13
<b>PCB (PrintedCircuit Board)</b> .....	14
<b>SOFTWARE</b> .....	15
xBee.....	15
Software sul PC.....	16
Grafica.....	16
Codice.....	16
<b>LISTA COMPONENTI e COSTI</b> .....	20
<b>CONCLUSIONI</b> .....	21

## INTRODUZIONE

Questa rete di sensori è pensata per monitorare la temperatura in un bosco per prevenire un incendio.

È costituita da un punto centrale collegato a un PC e da tanti moduli uguali contenenti un sensore di temperatura.

La comunicazione con il punto centrale avviene tramite i moduli wireless xBee. Il protocollo di comunicazione usato tra di essi è lo ZIGBEE, protocollo implementato nei moduli xBee che permette un basso consumo energetico rispetto alla comunicazione wireless di base.

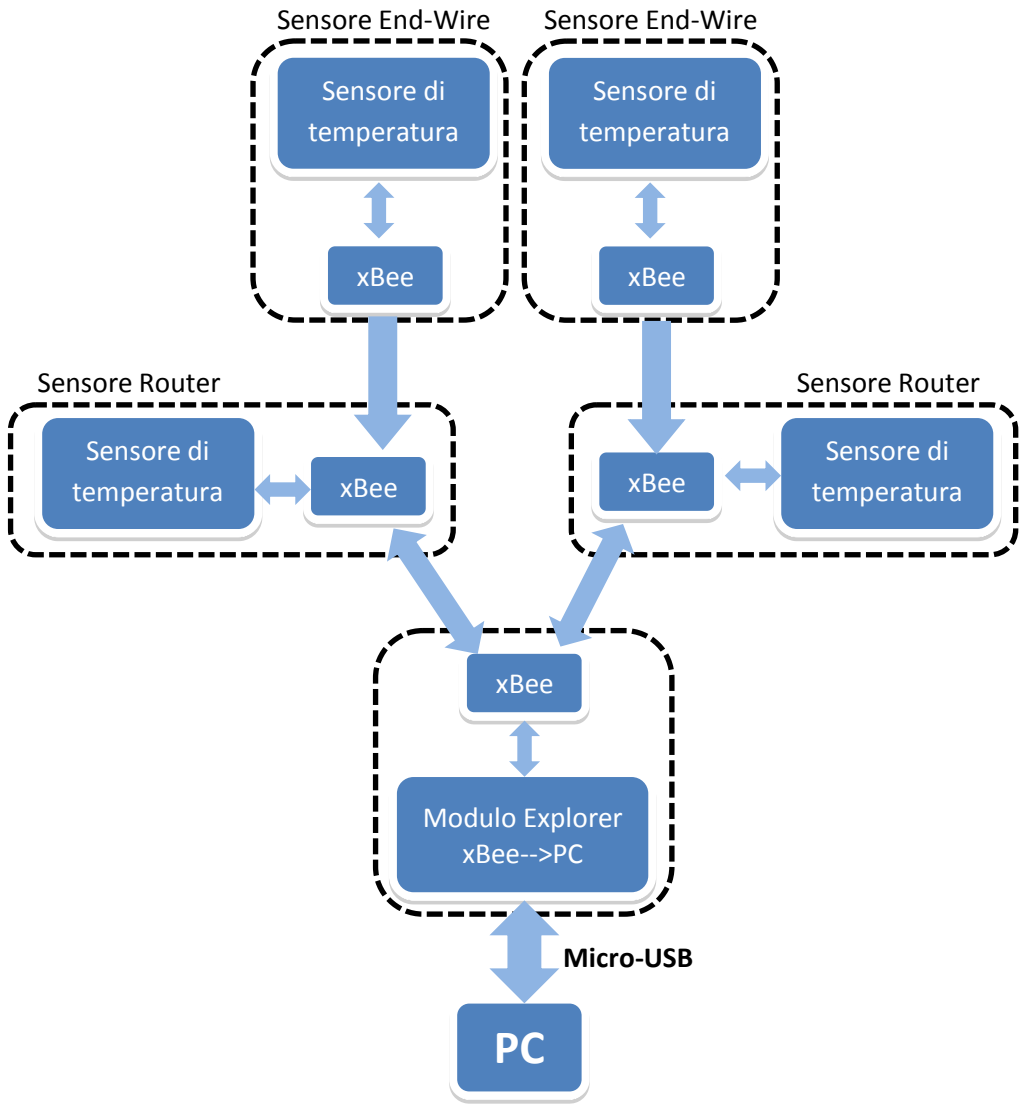
I moduli sono costituiti da: una batteria a 9V, un interruttore, un circuito di regolazione a 3.3V, un led che segnala la presenza di alimentazione, un xBee ed un sensore di temperatura LM35.

L'xBee deve essere alimentato con una tensione compresa tra 3V e 3.4V, il datasheet consiglia 3.3V.

È stato scelto un sensore di temperatura LM35 perché i moduli xBee in modalità ZIGBEE possono accettare in ingresso al loro ADC tensioni da 0V a 1.2V.

LM35, infatti, fornisce in uscita 0V a 0°C e 1.2V a 120°C con un passo di 10mV/°C. Utilizzando questo sensore quindi non serve un circuito di condizionamento del suo segnale.

# SCHEMA A BLOCCHI

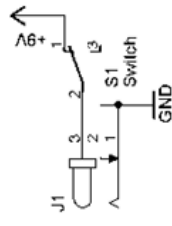


# SCHEMA ELETTRICO

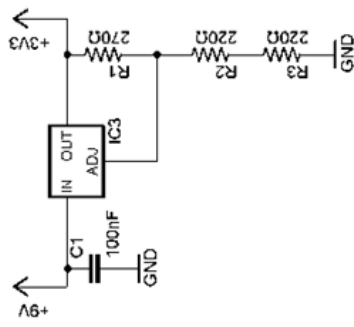
## INTERFACE

Power Supply, Regulators, Connectors, LEDs and Switches

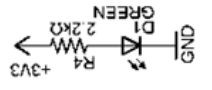
**Battery**  
+5 V ... +12V DC  
Jack 2.1mm



Power Supply Connector & Switch



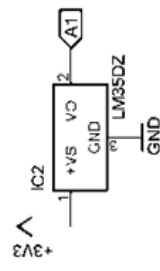
3.3V Voltage Regulator



Power-ON LED

## ANALOG SENSORS

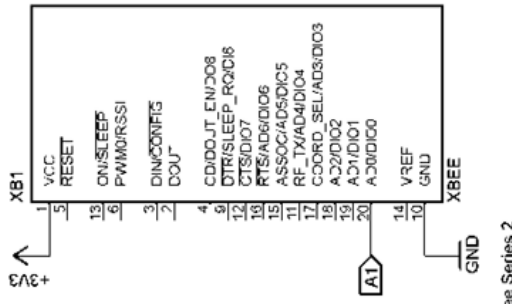
LM35 Thermal Sensor



Temperature Sensor

## WIRELESS

xBee Wireless Module



xBee Series 2

Ivancich Stefano  
ITIS C. Zuccante  
5EA a.s. 2014/15

ReleSensoriXbee

25/03/2015 07:35:50

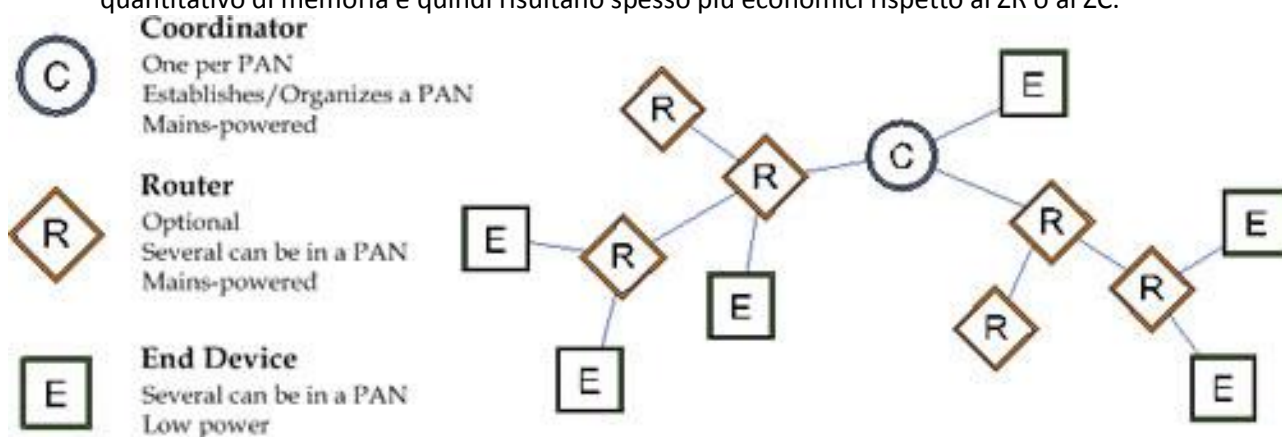
Sheet: 1/1

## RETE ZIGBEE

Nel mondo delle tecnologie senza fili ZigBee rappresenta uno dei principali standard di comunicazione. Attraverso l'uso di piccole antenne digitali a bassa potenza e basso consumo basate sullo standard IEEE 802.15.4.

Ci sono tre differenti tipi di dispositivo ZigBee:

- ZigBee **Coordinator** (ZC): è il dispositivo più "intelligente" tra quelli disponibili, costituisce la radice di una rete ZigBee e può operare da ponte tra più reti. Ci può essere un solo "Coordinator" in ogni rete. Esso è inoltre in grado di memorizzare informazioni riguardo alla sua rete e può agire come deposito per le chiavi di sicurezza.
- ZigBee **Router** (ZR): questi dispositivi agiscono come router intermedi passando i dati da e verso altri dispositivi. Di fatto non vi sono distinzioni hardware tra un ZC e un ZR se non che viene rilasciato al coordinator il ruolo di inizializzare la rete, dopodiché diventano dispositivi identici.
- ZigBee **End-Device** (ZED): includono solo le funzionalità minime per dialogare con il suo nodo parente (Coordinator o Router), non possono trasmettere dati provenienti da altri dispositivi e dunque non partecipano al multi-hop di un messaggio; sono i nodi che richiedono il minor quantitativo di memoria e quindi risultano spesso più economici rispetto ai ZR o ai ZC.

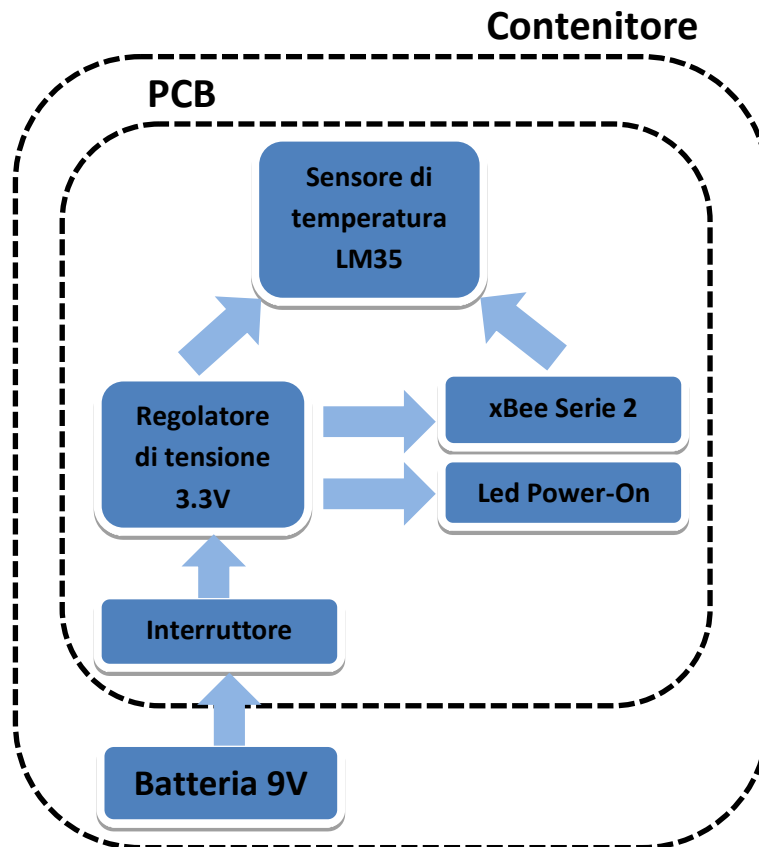


## STRUTTURA SENSORE

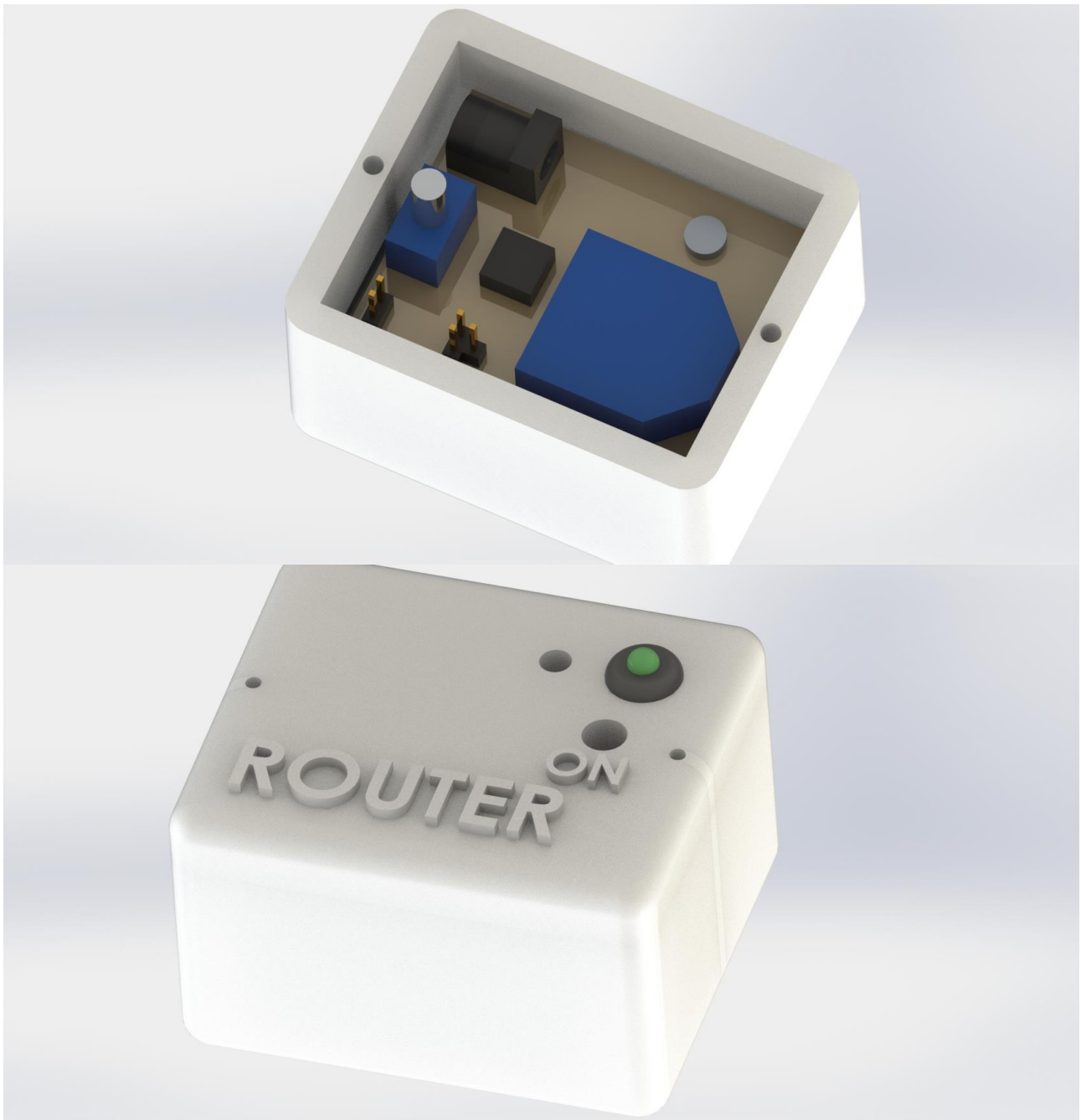
Il contenitore del circuito è stato pensato per essere posto in un bosco, interrato nel suolo con il sensore di temperatura fuoriuscente oppure incastrato in un albero.

E' diviso in 2 parti: una interna che tiene saldi tutti i componenti ed una esterna che funge da "coperchio" che protegge l'interno e fa scorgere l'interruttore, il sensore e il LED.

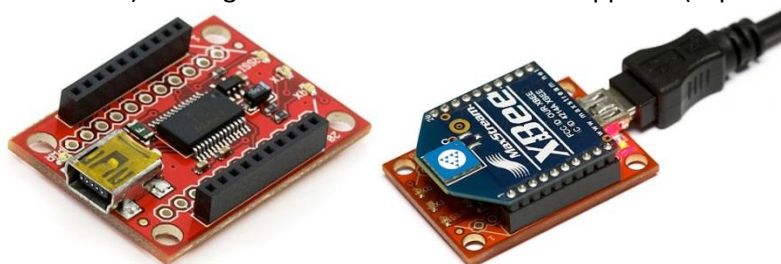
Il contenitore è inoltre pensato per essere impermeabile, sfortunatamente non avendo i mezzi a disposizione ne sono stati realizzati dei prototipi con una stampante 3D.







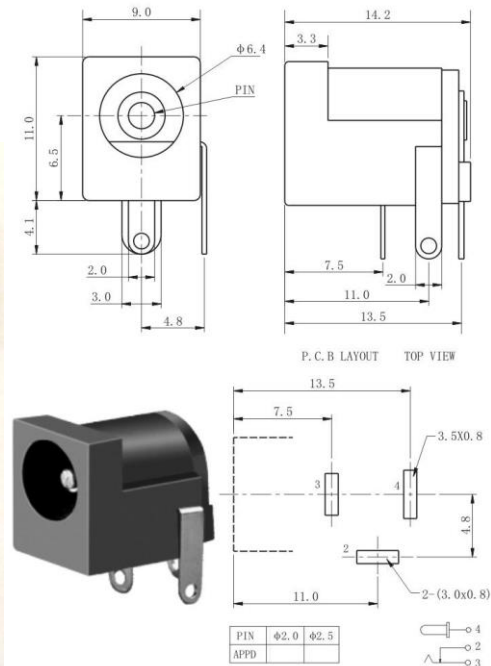
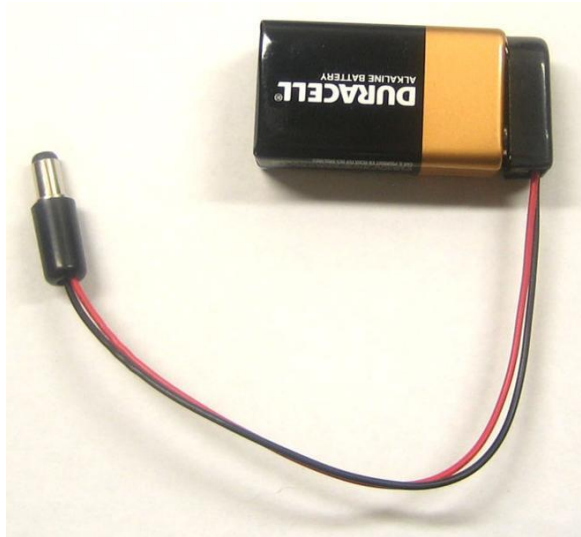
L'xBee centrale (COORDINATOR) è collegato al PC tramite un modulo apposito(Explorer) via USB.





## ALIMENTAZIONE

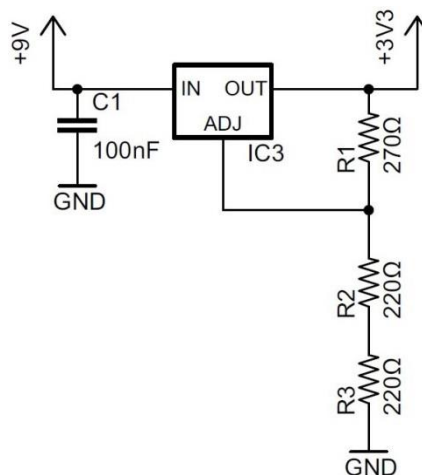
L'alimentazione di ciascun sensore avviene tramite una batteria a 9V 1200mA con un apposito connettore jack da 2.1mm. Possono essere utilizzate batterie anche con tensione maggiore ad esempio 12V, 15V, 18V.



## CIRCUITO DI REGOLAZIONE DEI 3.3V LM317

Questo circuito trasforma la tensione fornita dalla batteria in una tensione stabile di 3.3V necessari per alimentare l'xBee e il sensore di temperatura LM35.

Schema preso dal datasheet.



### Dimensionamento:

Da datasheet:

- $V_O = V_{REF} * \left(1 + \frac{R_2}{R_1}\right) + (R_2 * I_{ADJ})$
- $V_{REF}=1.25V$  è la tensione di riferimento tra il pin OUT e il pin ADJ, quindi la tensione ai capi di  $R_1$ .
- $I_{OUT(MIN)}=5mA$  è la minima corrente di uscita necessaria a mantenere la regolazione. Valore tipico 3.5mA, valore massimo 10mA. Si è scelto 5mA per semplificare i calcoli.
- $I_{ADJ}=100\mu A$  è la corrente massima che può uscire dal pin ADJ.
- $C_1=100nF$  non è necessario ma è raccomandato se non è presente un condensatore di bypass o il regolatore non è posto vicino ad esso.

Quindi:

$$R_1 = \frac{V_{REF}}{I_{OUT(MIN)}} = \frac{1.25V}{5mA} = 250\Omega$$

Si è scelta  $R_1=270\Omega$ .

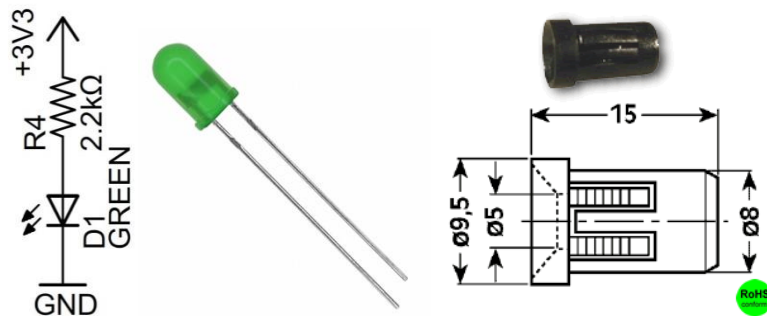
Quindi  $I_{OUT(MIN)}=4.6mA$ .

$$R_2 = \frac{V_O - V_{REF}}{\frac{V_{REF}}{R_1} + I_{ADJ}} = \frac{3.3 - 1.25}{\frac{1.25}{270} + 100 \cdot 10^{-6}} = 442.75\Omega$$

Si è scelta  $R_2=440\Omega$ , quindi di usare 2 resistori in serie da  $220\Omega$ .

## LED DI POWER-ON

Il led verde indica che il dispositivo è acceso.



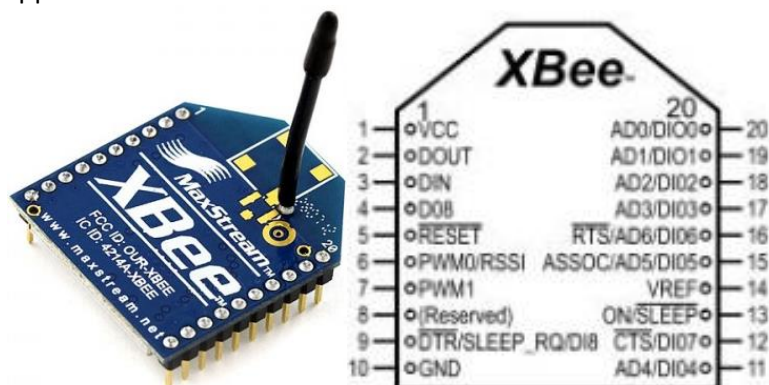
Dimensionamento:

- $V_{CC}=3.3V$  Tensione fornita dal regolatore LM317.
- $V_{LED}=2.2V$
- $I_{LED}=500\mu A$  E stato provato in laboratorio che il led ha una buona luminosità anche con solo  $500\mu A$ .

$$R_4 = \frac{V_{CC} - V_{LED}}{I_{LED}} = \frac{3.3 - 2.2}{500\mu A} = 2.2k\Omega$$

## XBEE SERIE 2

Il modulo XBee Series 2 è una soluzione compatibile con lo standard ZigBee/IEEE 802.15.4 che soddisfa la necessità di una rete a basso costo e a basso consumo, pensata soprattutto per l'utilizzo con sensori. I moduli sono semplici da utilizzare, richiedono pochissima energia e costituiscono una soluzione efficace ed affidabile per la trasmissione di dati critici. Le dimensioni estremamente compatte, fanno risparmiare spazio prezioso nelle applicazioni.



### Caratteristiche tecniche:

- Frequenza operativa 2.4 GHz
- Potenza RF 2 mW (+3 dBm)(boost mode abilitato)
- Portata fino a: 120mt al chiuso e 2500mt all'aperto
- Velocità RF: 250.000bps
- Velocità interfaccia seriale: 1200 ... 230400 bps
- Possibilità di antenna filo, Chip oppure connettore U.FL. RF e RPSMA
- Range di Temperatura Industriale (-40 – 85° C)
- Approvato per utilizzo in Europa, USA e Canada
- Supportate modalità di rete avanzate e basso consumo

Molti persone, quando considerano un'applicazione wireless con i microcontrollori, comparano i moduli XBee a 2.4GHz con dei moduli radio, a basso costo, ad esempio da 433MHz o simili. Sebbene i moduli a 433MHz costano meno e possono essere impiegati in alcune semplici applicazioni, i moduli XBee offrono notevoli vantaggi.

Il primo grande vantaggio consiste nel fatto che i moduli XBee sono bidirezionali, parecchi sistemi economici a 433MHz sono unidirezionali, con questo sistema il trasmettitore non ha idea se il ricevitore stia ricevendo i dati o meno. I moduli XBee trasmettono e ricevono i dati in entrambe le direzioni, in questo modo è possibile testare facilmente (da entrambe i lati) se il sistema sta funzionando correttamente.

Il secondo vantaggio consiste nell'indirizzamento univoco di questi moduli. Ogni XBee ha un numero seriale univoco. Questo significa che 2 o più unità possono essere settate per parlare esclusivamente tra loro, ignorando tutti i segnali di altri moduli. Questo non è facilmente ottenibile con i moduli a 433MHz.

Il terzo vantaggio consiste nella logica precostruita all'interno dei moduli XBee, infatti sono già implementati tutti i necessari controlli tipici di una trasmissione wireless, quali ad esempio l'error checking. Infine il protocollo XBee consente diversi numeri di canali, settando differenti unità in differenti canali, possono essere minimizzate eventuali interferenze.

## CONSUMI

Circuito di regolazione a 3.3V:  $I_{3.3V} = I_{OUT(MIN)} + I_{ADJ} = 4.6mA + 0.1mA = 4.7mA$

Led di power-on:  $I_{LED} = 500\mu A$

LM35:  $I_{LM35} = 500\mu A$  da Datasheet

Quindi si ha un consumo teorico continuo di 5.7mA ma tenendo conto delle tolleranze dei componenti ci si può mettere nel caso peggiore di un consumo di 6mA.

xBee:

- $I_{SLEEP} = 15mA$  Consumo a riposo
- $I_{TX} = 170mA$  Consumo in modalità trasmissione.
- $I_{RX} = 45mA$  Consumo in modalità ricezione.

Come si può notare l'xBee consuma molto, quindi non tutti gli xBee posso essere di tipo ROUTER, ovvero che trasmettono e ricevono continuamente, ma anche un END-WIRE non lo si può tenere continuamente in modalità trasmissione.

Eseguendo dei test in laboratorio si è visto che il segnale per essere inviato può metterci fino ad 1s.

Quindi un modulo END-WIRE dovrà stare a riposo per un certo lasso di tempo e poi trasmettere per 1s.

Sfortunatamente i moduli ROUTER dovranno stare in ricezione continuamente per far arrivare al punto centrale i dati degli END-WIRE e ROUTER più lontani.

Ogni quanto far trasmettere un dato può essere deciso dall'utente a seconda delle proprie esigenze.

Calcolo corrente media assorbita dai moduli:

END-WIRE:

$$I_{M(END)} = \frac{(I_{SLEEP} * T_{SLEEP}) + (I_{TX} * T_{TX})}{T_{SLEEP} + T_{TX}}$$

$T_{SLEEP}$  è l'intervallo di tempo deciso dall'utente per ogni quanto vuole fare una misura.

$T_{TX} = 1s$  è il lasso di tempo massimo impiegato dall'xBee per trasmettere un dato.

ROUTER:

$$I_{M(ROUT)} = \frac{(I_{RX} * T_{SLEEP}) + [(I_{TX} * T_{TX}) * n_{DISP}]}{T_{SLEEP} + (T_{TX} * n_{DISP})}$$

Dove  $n_{DISP}$  è il numero di dispositivi da qui il ROUTER deve ricevere i dati.

Durata di vita della batteria:

$$T_{LIFE} = \frac{C_{BATT}}{I_M}$$

Dove  $C_{BATT}$  è la capacità in mA/h che la batteria può fornire e  $I_M$  è la corrente media in mA assorbita dal ROUTER o l'END-WIRE.

### Esempio

Avendo a disposizione una batteria comune da 9V 1000mA/h per ogni modulo, assumendo che ogni ROUTER debba ricevere dati da altri 3 ROUTER e 10 END-WIRE e si vuole prendere una misura ogni 10min(600s).

Ogni quanto si devono effettuare dei cambi di batterie?

$$I_{M(END)} = \frac{(I_{SLEEP} * T_{SLEEP}) + (I_{TX} * T_{TX})}{T_{SLEEP} + T_{TX}} = \frac{(15mA * 600s) + (170mA * 1s)}{600s + 1s} = 15.26mA$$

$$T_{LIFE(END)} = \frac{C_{BATT}}{I_{M(END)}} = \frac{1000mA/h}{15.26mA} = 65.53 h$$

$$I_{M(ROUT)} = \frac{(I_{RX} * T_{SLEEP}) + [(I_{TX} * T_{TX}) * n_{DISP}]}{T_{SLEEP} + (T_{TX} * n_{DISP})} = \frac{(45mA * 600s) + [(170mA * 1s) * 13]}{600s + (1s * 13)} = 47.65mA$$

$$T_{LIFE(ROUT)} = \frac{C_{BATT}}{I_{M(ROUT)}} = \frac{1000mA/h}{47.65mA} = 20.98 h$$

Quindi la batteria per un END-WIRE va sostituita circa ogni 3 giorni mentre quella di un ROUTER ogni giorno.

Come si nota dai risultati è chiaro che i moduli xBee pur essendo tra i dispositivi di comunicazione wireless di minor consumo energetico sul mercato, assorbono tanta corrente.

Quindi per aumentare l'efficienza di questa rete è necessario utilizzare una batteria con capacità superiore. E anche aumentare l'intervallo di tempo tra un campione di temperatura e l'altro, così facendo però la parte di bosco incendiata può venire rilevata tempo dopo e quindi il fuoco si propaga di più. Per ovviare a questo si può aumentare il numero di dispositivi.

Ad esempio si può utilizzare una batteria da 100A/h, campionare 1 volta all'ora e pretendendo che un ROUTER riceva dati da altri 10 ROUTER e 400 END-WIRE. Questo significa che su un'area quadrata di 100x100m si ha un sensore ogni 20m e la durata delle batterie raggiunge un anno.

## **INSTALLAZIONE e MANUTENZIONE DELLA RETE**

Per utilizzare questa rete è necessario sapere in precedenza quanti sensori si vogliono collocare e a quale distanza l'uno dall'altro.

Infatti, prima dell'installazione deve esserci delle fasi di preparazione del prodotto:

1. Settaggio degli xBee.
2. Preparazione dei circuiti e dei contenitori.
3. Preparazione del software grafico da installare sul PC (Cartina, disposizione dei sensori e varianti proposte dal cliente).

L'installazione avviene in più fasi:

1. Installazione del software grafico sul PC del cliente, eventualmente fornendogliene uno.
2. Avvio del software e del coordinator.
3. Gli operatori collocando ogni sensore lo accendono uno alla volta e lo comunicano all'operatore che è situato nel punto centrale davanti al monitor che vede il nuovo xBee acceso e lo posiziona nella cartina.

### **Manutenzione**

Quando un sensore non invia più dati, un operatore si reca per verificarne il funzionamento.

Questi deve aprire il modulo controllare tramite un tester il circuito elettronico la tensione in vari punti.

La batteria va sostituita periodicamente, poco prima del tempo calcolato dalle formule indicate nel paragrafo dei consumi.

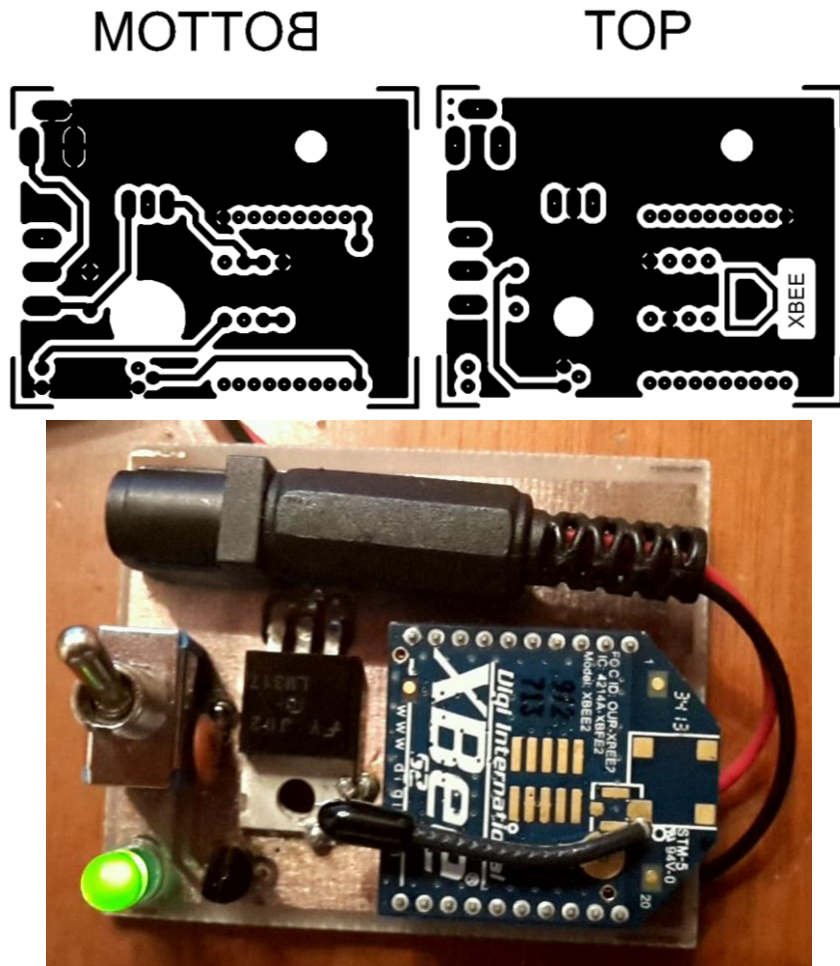
Sia quando si cambia batteria che quando sostituisce un sensore si deve riaccendere e ripetere il passo 3 dell'installazione.



## PCB (Printed Circuit Board)

È stato realizzato un PCB contenente tutti i componenti precedentemente elencati.

Il pcb e lo schema elettrico sono stati realizzati mediante l'utilizzo del software EAGLE CAD 7.1.



# SOFTWARE

## xBee

L'xBee non ha bisogno di essere programmato con un linguaggio di programmazione, ma possiede un firmware da caricare e dei valori da settare tramite il software XCTU fornito dalla DIGI.

Il firmware serve ad indicare il tipo di dispositivo ZigBee(Coordinator, Router o End-Device).

Vediamo il firmware da caricare e i parametri da impostare.

### Coordinator:

- **Firmware:** COORDINATOR API.
- **PAN ID:** indirizzo desiderato per il coordinator

**Firmware information**

Product family: XB24-ZB  
Function set: ZigBee Coordinator API  
Firmware version: 21A7

Written and default  
Written and not default  
Changed but not written  
Error in setting

▼ **Networking**  
Change networking settings

ID PAN ID	1234	
SC Scan Channels	FFFF	Bitfield
SD Scan Duration	3	exponent
ZS ZigBee Stack Profile	0	
NJ Node Join Time	FF	x1 sec
OP Operating PAN ID	1234	
OI Operating 16-bit PAN ID	4873	
CH Operating Channel	E	
NC Number of Re...ing Children	A	

### Router e End-Device:

- **Firmware:** ROUTER AT oppure END-DEVICE AT
- **PAN ID:** lo stesso del coordinator
- **JV:** Enabled
- **D0:** ADC[2]
- **IR sampling:** periodo di campionamento desiderato in ms.

**Firmware information**

Product family: XB24-ZB  
Function set: ZigBee Router AT  
Firmware version: 22A7

Written and default  
Written and not default  
Changed but not written  
Error in setting

▼ **Networking**  
Change networking settings

ID PAN ID	1234	
SC Scan Channels	FFFF	Bitfield
SD Scan Duration	3	exponent
ZS ZigBee Stack Profile	0	
NJ Node Join Time	FF	x1 sec
NW Network Watchdog Timeout	0	x1 minute
JV Channel Verification	Enabled [1]	
JN Join Notification	Disabled [0]	

▼ **I/O Settings**  
Modify DIO and ADC options

D0 AD0/DIO0 Configuration	ADC [2]	
D1 AD1/DIO1 Configuration	Disabled [0]	
D2 AD2/DIO2 Configuration	Disabled [0]	
D3 AD3/DIO3 Configuration	Disabled [0]	
D4 DIO4 Configuration	Disabled [0]	
D5 DIO5/Assoc Configuration	Associated indicator [1]	
P0 DIO10/PWM0 Configuration	RSSI PWM Output [1]	
P1 DIO11 Configuration	Disabled [0]	
P2 DIO12 Configuration	Disabled [0]	

▼ **I/O Sampling**  
Configure IO sampling parameters

IR IO Sampling Rate	3E8	x1 ms
IC Digital IO Change Detection	0	
V+ Supply Voltage Threshold	0	

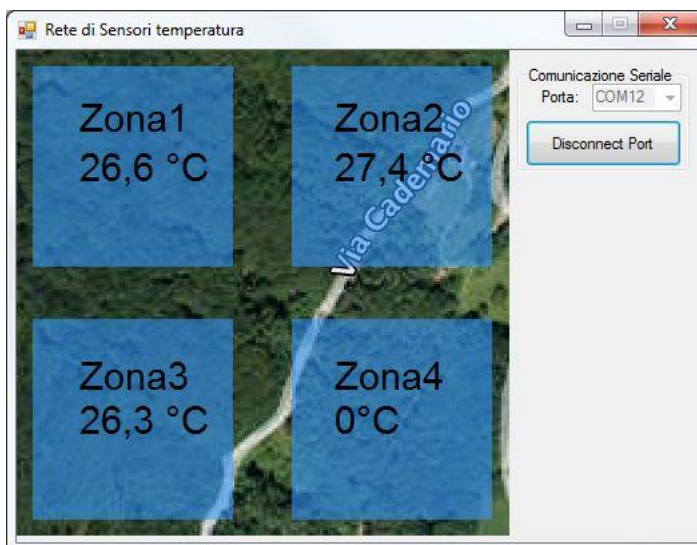
## Software sul PC

Il software su pc è realizzato con **Visual Basic .NET** per consentire una maggiore velocità e facilità in fase di programmazione.

### Grafica

La parte grafica visualizzabile da un operatore posto sul punto centrale della rete, si presenta come una cartina della zona, con sopra dei quadrati posti sulle aree in cui sono presenti i corrispondenti sensori.

Questi quadrati sono colorati, il loro colore cambia a seconda della temperatura rilevata dal corrispondente sensore: azzurro (temperatura<45°C), arancio (45°C<temperatura<60°C) e rosso (temperatura>60°C). Inoltre è presente il valore di temperatura, la distinzione dei colori è utile all'operatore per riconoscere facilmente a vista una temperatura anomala in caso siano presenti molti sensori.



### Codice

Ogni xBee invia un frame di 22byte in esadecimale all'xBee centrale, il primo byte rappresenta il byte di start che vale 7E, i byte 12 e 13 indicano l'indirizzo dell'xBee invece i byte 20 e 21 rappresentano il valore letto dall'ADC dell'xBee. Quindi il software quando legge il byte 7E si prepara a leggere altri 21 byte e preleva solo il 20esimo e il 21esimo, dopo di che gli converte in decimale e si ricalcola il valore originale.

$$Temperatura = \frac{ADC}{1023} * k$$

Dove **k=100**.

Il valore di k è impostato di norma a 100 per ogni sensore ma può essere modificato dall'operatore poiché i sensori non sono precisi e col tempo si usurano e potrebbero non restituire una temperatura corretta.

#### Zigbee IO Data Sample RX Indicator

7E 00 12 92 00 13 A2 00 40 B4 17 32 4F 28 01 01 00 00 01 00 D9 28

- **Start delimiter:** 7E ← **Byte 0: inizio frame**
- **Length:** 00 12 (18)
- **Frame type:** 92 (Zigbee IO Data Sample RX Indicator)
- **64-bit source address:** 00 13 A2 00 40 B4 17 32
- **16-bit source address:** 4F 28 ← **Byte 12 e 13: indirizzo**
- **Receive options:** 01
- **Number of samples:** 01
- **Digital channel mask:** 00 00
- **Analog channel mask:** 01
- **Analog sample:** 00 D9 ← **Byte 19 e 20: Valore ADC**
- **Checksum:** 28

```

Public Class Form1
    Dim porteDisponibili As Array          'Array che conterrà la lista delle porte COM disponibili
    Dim DatoRicevuto As String              'Variabile che conterrà un Byte ricevuto dalla comunicazione seriale
    Dim stringaRicevuta(22) As String       'Array che conterrà una frame ricevuto, ovvero 22 Byte

    Dim valoreHEX As String                 'Variabile che conterrà il valore in esadecimale
    Dim valoreDEC As Integer                'Variabile che conterrà il valore in decimale
    Dim valoreVolt As Double                'Variabile che conterrà il valore di tensione acquisita dall'ADC di
    un xBee

    Dim indirizzo As String
    Dim indirizzi(10) As Integer

    Dim gradi(10) As Double                'Array che conterrà le temperature di tutte le zone

    Dim i As Integer                       'Variabile utilizzata come contatore per acquisire i 22 byte dalla Serial Port
    Dim e As Boolean                        'Flag utilizzato per l'abilitazione dell'acquisizione del Frame di 22 Byte dalla
    SerialPort

    Private Sub Form1_Load() Handles MyBase.Load 'Funzione eseguita all'avvio del programma
        SerialPort1.Close()                 'Chiude la porta seriale se era già aperta
        porteDisponibili = IO.Ports.SerialPort.GetPortNames() 'Mette nell'array le porte seriali
    disponibili
        ComboBoxPort.Items.AddRange(porteDisponibili) 'Mette nella ComboBox(lista) le porte seriali
    disponibili
        i = 0                                'Inizializza il contatore a 0
        e = False                            'Inizializza il Flag di abilitazione
    dell'acquisizione del Frame a FALSO(Disabilitato)

        Panel1.BackColor = Color.FromArgb(150, 51, 153, 255) 'Inizializza il colore dei sensori come
    azzurro
        Panel2.BackColor = Color.FromArgb(150, 51, 153, 255)
        Panel3.BackColor = Color.FromArgb(150, 51, 153, 255)
        Panel4.BackColor = Color.FromArgb(150, 51, 153, 255)
    End Sub

    Private Sub SerialPort1_DataReceived() Handles SerialPort1.DataReceived 'Funzione eseguita ogni volta che
    la porta seriale riceve un dato
        'In questo caso la porta seriale non riceverà 1 Byte alla volta ma 22Byte tutti insieme,
        'quindi questo evento(Data_Received) viene richiamato 1 Volta sola ogni 22Byte ricevuti.
        DatoRicevuto = SerialPort1.ReadByte() 'Assegna alla variabile 1 Byte letto dalla coda della porta
    seriale

        If i > 20 Then i = 0 'Se il contatore è >20 (ovvero 21, cioè sono state eseguite 21 letture oltre a
    quella iniziale)
        If DatoRicevuto = 126 Then e = True '126 in decimale = 7E in esadecimale, ovvero il byte di inizio
    Frame

        If e Then
            stringaRicevuta(i) = DatoRicevuto
            'Questa Funzione(Data_Received) non può agire sulla parte grafica, usando invoke(...) può
            Me.Invoke(Sub()
                For j As Integer = 0 To 20 Step 1 'Leggo i 21 Byte rimasti
                    i = i + 1
                    DatoRicevuto = SerialPort1.ReadByte()
                    stringaRicevuta(i) = DatoRicevuto
                    TextBox1.Text = TextBox1.Text + " " + stringaRicevuta(j)
                Next
                'Unione dei 2 byte del valore dell'ADC e calcolo valore in Volt:
                Dim valoreHex20 As String
                If Hex(stringaRicevuta(20)).Length < 2 Then
                    valoreHex20 = "0" & Hex(stringaRicevuta(20))
                Else
                    valoreHex20 = Hex(stringaRicevuta(20))
                End If
                valoreHEX = Hex(stringaRicevuta(19)) & valoreHex20 'Hex(stringaRicevuta(20)) 'Il
    valore della misura dell'ADC è composto da 2 Byte
                valoreDEC = Convert.ToInt32(valoreHEX, 16) 'Converto il valore esadecimale in un
    Intero a 16 bit
                valoreVolt = (valoreDEC / 1023) * 1.2 'Calcolo il valore dei volt misurati

                indirizzo = Hex(stringaRicevuta(12)) & Hex(stringaRicevuta(13)) 'Ricavo
    l'indirizzo dell'xBee
                Dim indirizzoDec As Integer

```

```

        indirizzoDec = Convert.ToInt32(indirizzo, 16)

        'Inserisco nell'array dei valori i rispettivi valori dei sensori
        If indirizzi(0) = indirizzoDec Then
            indirizzi(0) = indirizzoDec
            gradi(0) = valoreVolt * NumericUpDown1.Value
        Else
            If indirizzi(0) = 0 Then
                indirizzi(0) = indirizzoDec
                gradi(0) = valoreVolt * NumericUpDown1.Value
            Else
                If indirizzi(1) = indirizzoDec Then
                    indirizzi(1) = indirizzoDec
                    gradi(1) = valoreVolt * NumericUpDown2.Value
                Else
                    If indirizzi(1) = 0 Then
                        indirizzi(1) = indirizzoDec
                        gradi(1) = valoreVolt * NumericUpDown2.Value
                    Else
                        If indirizzi(2) = indirizzoDec Then
                            indirizzi(2) = indirizzoDec
                            gradi(2) = valoreVolt * NumericUpDown3.Value
                        Else
                            If indirizzi(2) = 0 Then
                                indirizzi(2) = indirizzoDec
                                gradi(2) = valoreVolt * NumericUpDown3.Value
                            End If
                        End If
                    End If
                End If
            End If
        End If

        disegna()      'Richiama la funzione disegna() che farà visualizzare graficamente i
dati
    End Sub)
End Sub

Private Sub ButtonConnectSerialPort_Click() Handles ButtonConnectSerialPort.Click
    On Error Resume Next      'Se accadono errori ad una istruzione, la salta per evitare che il
programma si arresti
    If ButtonConnectSerialPort.Text = "Connect Port" Then      'Se il testo del bottone ConnectSerialPort è
"Connect Port":
        SerialPort1.PortName = ComboBoxPort.Text      'Assegna al portname la porta COM del
ComboBoxPort selezionata
        SerialPort1.Open()      'Apre la porta seriale
        ButtonConnectSerialPort.Text = "Disconnect Port"      'La scritta sul bottone connetti diventa
"Disconnect Port"
        ComboBoxPort.Enabled = False      'La ComboBoxPort non può essere più
selezionata
    Else
        SerialPort1.Close()      'Chiude la porta seriale
        ButtonConnectSerialPort.Text = "Connect Port"      'La scritta sul bottone connetti diventa
"Connect Port"
        ComboBoxPort.Enabled = True      'La ComboBoxPort può essere di nuovo
selezionata
        e = False
    End If
End Sub

Private Sub disegna()      'Funzione richiamata per rappresentare graficamente le temperature
    'Disegno ZONA 1:
    If gradi(0) < 40 Then
        Panel1.BackColor = Color.FromArgb(150, 51, 153, 255)
        'FromArgb(AlphaValue,RedValue,GreenValue,BlueValue) Azzurro
    ElseIf gradi(0) < 55 Then
        Panel1.BackColor = Color.FromArgb(150, 255, 128, 0)      'Arancione
    Else
        Panel1.BackColor = Color.FromArgb(150, 255, 0, 0)      'Rosso
    End If
    LabelGradi0.Text = gradi(0).ToString("0.0") + " °C"

    'Disegno ZONA 2:
    If gradi(1) < 40 Then

```

```

        Panel12.BackColor = Color.FromArgb(150, 51, 153, 255)
'FromArgb(AlphaValue,RedValue,GreenValue,BlueValue) Azzurro
    ElseIf gradi(1) < 55 Then
        Panel12.BackColor = Color.FromArgb(150, 255, 128, 0) 'Arancione
    Else
        Panel12.BackColor = Color.FromArgb(150, 255, 0, 0) 'Rosso
    End If
    LabelGradi1.Text = gradi(1).ToString("0.0") + " °C"

    'Disegno ZONA 3:
    If gradi(2) < 40 Then
        Panel13.BackColor = Color.FromArgb(150, 51, 153, 255)
'FromArgb(AlphaValue,RedValue,GreenValue,BlueValue) Azzurro
    ElseIf gradi(2) < 55 Then
        Panel13.BackColor = Color.FromArgb(150, 255, 128, 0) 'Arancione
    Else
        Panel13.BackColor = Color.FromArgb(150, 255, 0, 0) 'Rosso
    End If
    LabelGradi2.Text = gradi(2).ToString("0.0") + " °C"
End Sub
End Class

```

## LISTA COMPONENTI e COSTI

### Coordinator:

COMPONENTE	Q/TA	COSTO
Cavo USB-microUSB	1	2.00€
Explorer xBee	1	15.00€
xBee Serie 2	1	23.00€
Contenitore	1	5.00€

Costo totale coordinator: **45€**

### Router o End-Wire:

COMPONENTE	Q/TA	COSTO TOT
Batteria 9V 1200mA	1	1.50€
Connettore per Pila da 9V	1	0.31€
Jack 2.1mm	1	1.00€
Connettore coassiale 2,1mm	1	0.40€
Interruttore	1	0.80€
Strip pin 2mm	20	1.00€
Condensatore 100nF	1	0.10€
LM317	1	0.30€
Resistore 270	1	0.01€
Resistore 220	2	0.02€
Led verde 5mm	1	0.05€
LM35DZ	1	1.00€
xBee Serie 2	1	23.0€
PCB	1	1.00€
Contenitore	1	5.00€

Costo totale Router o End Wire: **35€**

Come si vede il costo è molto basso, ad esempio supponendo di volere una rete di 100 sensori, distanti 100m l'uno dall'altro, quindi comprendo un area quadrata di 1Km x 1Km, il costo totale sarà di 3500€ se non di meno, visto che comprando in grandi quantità i componenti si spende meno.



## CONCLUSIONI

Il progetto si è concluso con successo, anche se non si era partiti nel modo migliore, visto la scarsità di componenti reperibili nei laboratori dell'istituto.

Questo progetto rappresenta quindi un prototipo realizzato artigianalmente di un progetto molto più ampio.

Alcuni miglioramenti che sicuramente devono essere effettuati avendo più risorse e mezzi sono: utilizzo di una batteria che abbia una capacità maggiore, ad esempio si può implementare la tecnologia dei carica batterie portatili per gli smartphone, che posseggono capacità dell'ordine delle decine di migliaia di mA. Anche un contenitore impermeabile, poiché nei boschi c'è sicuramente molta umidità, e prima o poi piove. Utilizzo di un sensore di temperatura migliore, che magari abbia un range di temperature superiore ai 150°C.

Questo progetto risulta applicabile in altri ambiti, ad esempio all'interno di una serra per monitorare la temperatura ambientale in vari punti, o aggiungendo altri sensori, si può rilevare anche l'umidità e altre caratteristiche sulle piante e ambientali nei boschi, nelle serre e in qualsiasi altro ambiente.