



## Conversione testo in ottetti per messaggi SMS (seconda parte)

[http://www.vbsimple.net/activity/act\\_02\\_2.htm](http://www.vbsimple.net/activity/act_02_2.htm)

**Richiesta di: Marcello Gaidone - 6 Novembre 2000**

**Difficoltà:** 5 / 5

*Devo trasformare (ad esempio) la parola **hellohello** in una stringa formata da 8 bit, in modo da poterla utilizzare per mandare SMS sui cellulari.*

[<< Continua dalla parte 1](#)

Dopo aver visto l'alfabeto da utilizzare per effettuare la conversione, ci serviremo di una funzione che converte un numero decimale in numero binario ed un'altra funzione che effettua l'operazione inversa.

Cominciamo a definire le due funzioni appena citate:

```
1. Public Function Dec2Bin(ByVal NUMERO As Integer, Optional ByVal LUNGHEZZA As Integer) As String
2.     While NUMERO > 0
3.         Dec2Bin = IIf(NUMERO Mod 2 = 0, "0", "1") & Dec2Bin
4.         NUMERO = Int(NUMERO / 2)
5.     Wend
6.     If (LUNGHEZZA > 0) And (Len(Dec2Bin) < LUNGHEZZA) Then
7.         Dec2Bin = String(LUNGHEZZA - Len(Dec2Bin), Asc("0")) & Dec2Bin
8.     End If
9. End Function
10.
```

La prima funzione si chiama Dec2Bin, che pronunciato in inglese suona come Dec To Bin, ovvero Decimale a binario. Questa effettua una serie di divisioni e di moduli, secondo il sistema di conversione in binario. Questa funzione verifica che la stringa binaria risultante sia lunga almeno un certo numero di caratteri (valore di LUNGHEZZA). Se la lunghezza è minore di tale valore inserisce degli zeri alla sinistra del numero.

```
11. Public Function Bin2Dec(ByVal NUMERO As String) As Integer
12.     Dim TEMP As Integer
13.     For TEMP = 1 To Len(NUMERO)
14.         Bin2Dec = Bin2Dec + (Mid(NUMERO, TEMP, 1) * 2) ^ (Len(NUMERO) - TEMP)
15.     Next TEMP
16.     If Right(NUMERO, 1) = "0" Then Bin2Dec = Bin2Dec - 1
17. End Function
18.
```

Quest'ultima funzione effettua l'operazione inversa della funzione precedente. Da una stringa binaria restituisce il valore intero corrispondente.

Prima di iniziare con la vera funzione di conversione spieghiamo con calma il procedimento da seguire.

Prendiamo innanzitutto in esame la stringa **hellohello** da convertire. Convertiamo i singoli valori ASCII di ogni carattere in binario:

h	e	l	l	o	h	e	l	l	o
104	101	108	108	111	104	101	108	108	111
1101000	1100101	1101100	1101100	1101111	1101000	1100101	1101100	1101100	1101111

Come possiamo vedere dalla tabella sopra, ogni carattere di testo è formato da un massimo di 7 bit. Per questo ogni carattere semplice verrà chiamato *Settetto*. Il processo di conversione per l'invio via SMS richiede che il testo sia convertito in *ottetti*, in una maniera particolare.

Il primo settetto viene convertito in ottetto prendendo il bit all'estrema destra del secondo carattere e mettendolo alla sinistra del primo bit del primo carattere. Otterremmo così il primo ottetto: **1**1101000. Il primo bit in rosso è l'ultimo bit del secondo carattere.

Il secondo settetto perde l'ultimo bit e rimangono sei bit. Aggiungiamo a questi sei bit gli ultimi due (8 - 6) bit del terzo carattere. Otteniamo: **00**110010. Due bit del terzo carattere e sei bit del secondo, perché l'ultimo bit del secondo carattere è stato utilizzato per il primo.

Procediamo con il terzo ottetto: abbiamo 5 bit del settetto e 3 bit del carattere successivo: **100**11011. E così via...

Otterremmo quindi il seguente risultato:

- 11101000 = 232 = E8 esadecimale
- 00110010 = 50 = 32 "
- 10011011 = 155 = 9B "
- 11111101 = 253 = FD "
- 01000110 = 70 = 46 "
- 10010111 = 151 = 97 "
- 11011001 = 217 = D9 "
- 11101100 = 236 = EC "
- 00110111 = 55 = 37 "

Così i 10 settetti verranno convertiti in 9 ottetti.

In teoria è stato semplice. Ora viene la parte del codice:

```

19. Public Function CodificaSMS(ByVal MESSAGGIO As String) As String
20.     Dim TEMP As Integer
21.     Dim LUNGHEZZADAESAMINARE As Integer
22.     Dim TEMPSTR As String
23.     Dim OTTETTO As String
24.     InizializzaAlfabeto
25.     For TEMP = 1 To Len(MESSAGGIO)
26.         LUNGHEZZADAESAMINARE = TrovaCarattereAlfabeto7(Mid(MESSAGGIO, TEMP, 1))
27.         If LUNGHEZZADAESAMINARE > 127 Then
28.             TEMPSTR = TEMPSTR & Dec2Bin((LUNGHEZZADAESAMINARE And &HFF00) / &H100, 7)
29.             TEMPSTR = TEMPSTR & Dec2Bin((LUNGHEZZADAESAMINARE And &HFF), 7)
30.         Else
31.             TEMPSTR = TEMPSTR & Dec2Bin(LUNGHEZZADAESAMINARE, 7)
32.         End If
33.     Next TEMP
34.     TEMPSTR = TEMPSTR & String(7, Asc("0"))

```

```

35.     LUNGHEZZADAESAMINARE = 7
36.     For TEMP = 1 To Int(Len(TEMPSTR) / 8) + 1
37.         OTTETTO = Mid(TEMPSTR, TEMP * 7 - 6, LUNGHEZZADAESAMINARE)
38.         OTTETTO = Mid(TEMPSTR, (TEMP + 1) * 7 - (7 - LUNGHEZZADAESAMINARE), 8 -
LUNGHEZZADAESAMINARE) & OTTETTO
39.         LUNGHEZZADAESAMINARE = LUNGHEZZADAESAMINARE - 1
40.         If LUNGHEZZADAESAMINARE = 0 Then
41.             LUNGHEZZADAESAMINARE = 7
42.             TEMP = TEMP + 1
43.         End If
44.         OTTETTO = Hex(Bin2Dec(OTTETTO))
45.         If Len(OTTETTO) = 1 Then OTTETTO = "0" & OTTETTO
46.         CodificaSMS = CodificaSMS & OTTETTO
47.     Next TEMP
48. End Function

```

Le prime 4 righe dopo l'intestazione della funzione dichiarano 4 variabili:

- TEMP, usata per effettuare i cicli e calcoli generici
- LUNGHEZZADAESAMINARE, valore numerico che indica il numero di bit da prendere per volta. Per il primo settetto sono 7, per il secondo sono 6, etc.
- TEMPSTR è una stringa dentro la quale sarà memorizzata la sequenza di bit.
- OTTETTO è un ottetto generato.

Alla riga 24 inizializziamo l'alfabeto creato nel modulo precedente con la Sub **InizializzaAlfabeto**.

Subito dopo abbiamo un ciclo for che costruisce la stringa TEMPSTR.

```

25.     For TEMP = 1 To Len(MESSAGGIO)
26.         LUNGHEZZADAESAMINARE = TrovaCarattereAlfabeto7(Mid(MESSAGGIO, TEMP, 1))
27.         If LUNGHEZZADAESAMINARE > 127 Then
28.             TEMPSTR = TEMPSTR & Dec2Bin((LUNGHEZZADAESAMINARE And &HFF00) / &H100, 7)
29.             TEMPSTR = TEMPSTR & Dec2Bin((LUNGHEZZADAESAMINARE And &HFF), 7)
30.         Else
31.             TEMPSTR = TEMPSTR & Dec2Bin(LUNGHEZZADAESAMINARE, 7)
32.         End If
33.     Next TEMP

```

Per ogni singolo carattere della stringa MESSAGGIO estrae la stringa di 7 bit con Dec2Bin e la concatena alla stringa TEMPSTR.

L'unico problema che può sorgere è con i caratteri speciali composti da due bytes. Pertanto viene controllato se il valore estratto (**LUNGHEZZADAESAMINARE**) è maggiore di 127. In tal caso vengono processati i due caratteri in maniera separata (righe 28-29) dopo averli estratti con delle operazioni di AND. Se il numero non è superiore a 127, possiamo elaborare tale numero traducendolo in sequenze di bit.

```

34.     TEMPSTR = TEMPSTR & String(7, Asc("0"))

```

Questa riga aggiunge 7 zeri finali alla stringa TEMPSTR che verranno utilizzati dall'ultimo ottetto.

La funzione String genera una stringa di un certo numero di caratteri, tutti uguali.

Richiede due parametri: il primo è il numero di caratteri da generare ed il secondo è il codice ASCII del carattere da replicare.

Nel nostro esempio abbiamo usato la funzione Asc("0") per farci tornare il codice ASCII del numero 0. Abbiamo creato allora una stringa di 7 zeri e l'abbiamo concatenata alla

stringa di bit.

```
35.          LUNGHEZZADAESAMINARE = 7
```

Sulla riga 35 memorizziamo il numero iniziale di bit da prendere in considerazione per ogni settetto. All'inizio prenderemo l'intero settetto, per cui il valore di **LUNGHEZZADAESAMINARE** sarà 7.

```
36.      For TEMP = 1 To Int(Len(TEMPSTR) / 8) + 1
37.          OTTETTO = Mid(TEMPSTR, TEMP * 7 - 6, LUNGHEZZADAESAMINARE)
38.          OTTETTO = Mid(TEMPSTR, (TEMP + 1) * 7 - (7 - LUNGHEZZADAESAMINARE), 8 -
    LUNGHEZZADAESAMINARE) & OTTETTO
39.          LUNGHEZZADAESAMINARE = LUNGHEZZADAESAMINARE - 1
40.          If LUNGHEZZADAESAMINARE = 0 Then
41.              LUNGHEZZADAESAMINARE = 7
42.              TEMP = TEMP + 1
43.          End If
44.          OTTETTO = Hex(Bin2Dec(OTTETTO))
45.          If Len(OTTETTO) = 1 Then OTTETTO = "0" & OTTETTO
46.          CodificaSMS = CodificaSMS & OTTETTO
47.      Next TEMP
```

Ora inizia il vero ciclo di conversione. Per ogni settetto della stringa eseguiamo certe operazioni.

Alla riga 37 memorizziamo la prima parte dell'ottetto ovvero i caratteri rimasti del settetto preso in esame. Il numero di bit da memorizzare in OTTETTO è specificato dalla variabile **LUNGHEZZADAESAMINARE**.

```
38.          OTTETTO = Mid(TEMPSTR, (TEMP + 1) * 7 - (7 - LUNGHEZZADAESAMINARE), 8 -
    LUNGHEZZADAESAMINARE) & OTTETTO
```

La riga 38 è il centro di questa funzione. Essa fa precedere a OTTETTO i bit finali del settetto successivo.

Il numero di bit da prendere dal settetto successivo è dato dalla differenza di 8 - **LUNGHEZZADAESAMINARE**.

A partire da dove deve prendere quei bit?

TEMP è il carattere attuale, quindi TEMP+1 è il carattere successivo. Moltiplichiamo questo valore per 7, il numero di bit per settetto e troviamo il punto dove finisce il settetto che ci serve. Da lì torniamo indietro di 7-LUNGHEZZADAESAMINARE caratteri.

Infatti al primo ciclo TEMP avrà valore 1 e LUNGHEZZADAESAMINARE 7.

La riga dice di partire da  $(1 + 1) * 7$ , quindi 14, la fine del secondo settetto; dice pure di tornare indietro di  $(7 - 7)$  caratteri. Quindi il risultato di quest'operazione è il 14 carattere perché lo spostamento all'indietro è di 0 caratteri.

Dal 14° carattere abbiamo bisogno di un certo numero di caratteri per generare un ottetto, quindi 8 - LUNGHEZZADAESAMINARE. Prenderemo allora 1 carattere dal 14 bit.

In seguito di questo diminuiamo LUNGHEZZADAESAMINARE di 1. Dopo il primo ciclo allora sarà 6 il suo valore.

Se mediante l'ultima operazione LUNGHEZZADAESAMINARE giunge a 0 dice di ricominciare il conteggio da 7 e saltare un carattere poiché tutti i bit di questo settetto

saranno già stati presi dal ottetto precedente.

Quest'operazione avviene al completamento del 7° ottetto.

```
44.          OTTETTO = Hex(Bin2Dec(OTTETTO))
45.          If Len(OTTETTO) = 1 Then OTTETTO = "0" & OTTETTO
```

La riga 44 prende l'ottetto binario e lo converte in esadecimale mediante due operazioni. Prende l'ottetto binario, lo converte in decimale utilizzando la funzione Bin2Dec. Il risultato di quest'operazione sarà un numero decimale che andrà ad essere convertito in esadecimale dalla funzione di VB Hex. Il risultato di Hex sarà memorizzato nella variabile OTTETTO.

Dopo questo momento OTTETTO conterrà un valore esadecimale e non più un valore binario.

La riga 45 effettua un controllo d'obbligo. Se la lunghezza di OTTETTO è di un carattere, aggiunge alla sinistra di tale carattere uno zero ottenendo una cifra esadecimale completa di 2 caratteri.

```
46.          CodificaSMS = CodificaSMS & OTTETTO
```

L'ultima riga del ciclo memorizza i singoli ottetti nella variabile CodificaSMS che sarà il valore restituito all'uscita della funzione.

Questo processo di conversione termina qui.

Il consiglio a chi legge questo codice è di provarlo ed utilizzare gli strumenti di debug forniti in VB. Il codice è alquanto complesso. Una valida comprensione può essere data solo da una serie di prove e controlli pratici.

[Fibia FBI](#)

7 Novembre 2000

Corretto il 28 Marzo 2001



[Torna all'introduzione delle Richieste dei lettori](#)

---