## RC4 Encryption and Decryption Functions

When sending data between units, the data should be encrypted, no matter how trivial. It seems there are miscreants who like to create havoc although it is similarly trivial in impact on the “big picture” - the mental challenge seems to be the goal.

The RC4 Encryption Algorithm, is a shared key method requiring a secure exchange of a shared key i.e. both sides of the encode/decode must possess the key. RC4 is termed “symmetrical” - that is the same key is used to both encrypt and decrypt. It is now considered a deprecated method for really strong encryption but is still viable with a suitably complex key (at least 30 characters). To further harden things, it is possible to have different keys. You could easily modify the below code to have one used for decrypting a packet and another for sending. Further you could have separate packets for each recipient, e.g. in a network of 10 units, each could have a pair of keys that it uses when talking to each of the other devices. 20 keys in total but very easy to keep track of which to use. Even with a deprecated encryption, this would make it almost impossible for any snooper to make head or tail of what he was seeing on the network.

RC4$() returns a hexadecimal string of the original input expression. Note: The encrypted string has a 100% overhead in size over the original i.e. it will be twice the length of the input string.

UnRC4$() decodes a previously encrypted hexcadecimal string and returns the original string.

**Syntax**:

```
  RC4$(expression)
  UnRC4$(expression)
```

**Example**:

```
  a$=RC4$("The quick brown fox")
  b$=UnRC4$(encrypted_data$)
\\
```

**Code**:

```
    OPTION BASE 0

'optimized version 12NOV2021
'change this key to whatever you like but it should be complex
'at least 30 characters long and don't lose it! To increase security,
'you could use Base64 encoding to obfuscate it in your code.
'It won't stop a determined snooper but why make it easy for them?
'Suitable B64 routines are available in this namespace
    CONST
RC4KEY$=">4!1x4q3z4+7%4{9?5\3HhH^5$9=6@1~6,7_7|1)7'3]7[9:8<3*8S9I9l7Z1eT0r1"

    Function RC4$(z$)
        Local String o
        Local Integer i,j,x,y,t,t1,s(255),k(255)
        For i=0 To 255:s(i)=i:Next
        j=1:t=Len(RC4key$)
```

```
        For i=0 To 255
            if j>t Then j=1
            k(i)=Peek(Var RC4key$,j):j=j+1
        Next
        j=0
        For i=0 To 255
            j=(j+s(i)+k(i)) Mod &h100
            t=s(i):s(i)=s(j):s(j)=t
        Next
        i=0:j=0
        For x=1 To Len(z$)
            i=(i+1) Mod &h100:j=(j+s(i)) Mod &h100
            t1=s(i):s(i)=s(j):s(j)=t1
            t=(s(i)+(s(j) Mod &h100)) Mod &h100:y=s(t)
            o=o+Chr$(Peek(Var z$,x) Xor y)
        Next
        RC4$=""
        For x=1 To Len(o)
            RC4$=RC4$+Hex$(Peek(Var o,x),2)
        Next
    End Function

    Function UnRC4$(z$)
        Local String c
        Local Integer n
        For n=1 To Len(z$) Step 2
            c=c+Chr$(Val("&h"+Mid$(z$,n,2)))
        Next
        c=RC4$(c)
        For n=1 To Len(c) Step 2
            UnRC4$=UnRC4$+Chr$(Val("&h"+Mid$(c,n,2)))
        Next
    End Function
```