Cours 03
Instructions étudiées

Configuration

- **On (interrupt)**
- **Enable Interrupts**
- **Enable [interrupt]**
- **Config Adc**

Programmation

- **Chr()**
- **Declare Sub**
- **Declare Function**
- **ByVal / By Ref**

On Interrupt ...

Exécute un sous-programme quand arrive l'interruption spécifiée.

Syntaxe:

On INTERRUPT ETIQUETTE [NOSAVE]

Interrupt :

INT0 INT1 INT2 INT3 INT4 INT5 TIMER0 TIMER1 TIMER2 ADC
EEPROM CAPTURE1 COMPARE1A COMPARE1B COMPARE1.

On peut aussi utiliser la syntaxe AVR :

OC2 OVF2 ICP1 OC1A OC1B OVF1 OVF0 SPI URXC UDRE UTXE ADCC ERDY ACI

On Interrupt ...

ETIQUETTE : l'étiquette; où commence le sous programme d'interruption.

NOSAVE: Quand on spécifie NOSAVE, les registres ne sont ni sauvés ni restaurés au retour donc il faut sauvegarder et restaurer les registres utilisés (voir explications dans l'aide en ligne).

Le sous-programme doit se terminer par ***RETURN.***

On Interrupt ...

INTERRUPTION	DESCRIPTION
INT0	Interruption externe 0
INT1	Interruption externe 1
OVF0, TIMER0, COUNTER0	Interruption TIMER overflow 0
OVF1, TIMER1, COUNTER1	Interruption TIMER overflow 1
CAPTURE, ICP1	Interruption INPUT CAPTURE TIMER1
COMPARE1A, OC1A	Interruption TIMER1 OUTPUT COMPAREA
COMPARE1B, OC1B	Interruption TIMER1 OUTPUT COMPAREB
SPI	Interruption SPI

On Interrupt ...

INTERRUPTION	DESCRIPTION
URXC	Interruption transmission série RX complète
UDRE	Interruption registre des données série vide
UTXC	Interruption transmission série TX complète
SERIAL	Invalide URXC, UDRE, UTXC
ACI	Interruption Comparateur Analogique
ADC	Interruption convertisseur Analogique/digital
INT2, INT3, etc..	Interruptions dépendantes des µp.

Enable Interrupt(s)

Par défaut, les interruptions sont invalides. Pour invalider ou valider toutes les interruptions; il est obligatoire si on veut les utilisés.

[Enable / Disable] Interrupts

Enable [Interrupt]

Ensuite, on doit configurer les interruptions individuellement.

Enable Urxc

Enable Timer0

Enable Into

Disable Urxc

Disable Timer0

Disable Into

Config Adc

Configurer le convertisseur Analogique/digital. Dépendamment du model de microcontrôleur, il y a de 2 à plus de 8 canaux (**328p - 6**). Selon le modèle, la résolution est de 8 – 9 – **10** Bits.

Syntaxe:

CONFIG ADC=SINGLE, PRESCALER=AUTO, REFERENCE=opt.

ADC: Mode de travail peut être **SINGLE** ou **FREE.**

PRESCALER: pré-diviseur, une constante numérique pour le diviseur d'horloge.
Utiliser **AUTO** pour laisser le compilateur générer la meilleure valeur.

REFERENCE: Des options de références additionnelles. Leur valeur peut être OFF AVCC ou INTERNAL (**1.1 Volts**)...

Chr

Pour convertir une variable ou une constante en une String de 1 caractère représentant la valeur ASCII de la valeur numérique.

Syntaxe:

sVar=**Chr**(var)

sVar : Une String de 1 caractère

Var : Un Byte ou un Integer de 0 à 255

var=52

Print numvar résultat = 52

Print **chr**(var) résultat = 4 (52 est la valeur ASCII du caractère 4)

En Association avec LCD, permet d'afficher les caractères personnalisés.

Chr (ASCII)

Ctrl	Dec	Hex	Char	Code
^@	0	00		NUL
^A	1	01	☺	SOH
^B	2	02	☻	SIX
^C	3	03	♥	EIX
^D	4	04	♦	EOI
^E	5	05	♣	ENQ
^F	6	06	♠	ACK
^G	7	07	•	BEL
^H	8	08	◘	BS
^I	9	09	○	HI
^J	10	0A	◙	LF
^K	11	0B	♂	VI
^L	12	0C	♀	FF
^M	13	0D	♪	CR
^N	14	0E	♫	SO
^O	15	0F	☼	SI
^P	16	10	►	SLE
^Q	17	11	◄	CS1
^R	18	12	↕	DC2
^S	19	13	‼	DC3
^T	20	14	¶	DC4
^U	21	15	§	NAK
^V	22	16	▬	SYN
^W	23	17	↨	EIB
^X	24	18	↑	CAN
^Y	25	19	↓	EM
^Z	26	1A	→	SIB
^[	27	1B	←	ESC
^\	28	1C	∟	FS
^]	29	1D	↔	GS
^^	30	1E	▲	RS
^_	31	1F	▼	US

Dec	Hex	Char
32	20	sp
33	21	!
34	22	"
35	23	#
36	24	$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?

Dec	Hex	Char
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_

Dec	Hex	Char
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	⌂†

Dec	Hex	Char
128	80	Ç
129	81	ü
130	82	é
131	83	â
132	84	ä
133	85	à
134	86	å
135	87	ç
136	88	ê
137	89	ë
138	8A	è
139	8B	ï
140	8C	î
141	8D	ì
142	8E	Ä
143	8F	Å
144	90	É
145	91	æ
146	92	Æ
147	93	ô
148	94	ö
149	95	ò
150	96	û
151	97	ù
152	98	ÿ
153	99	Ö
154	9A	Ü
155	9B	¢
156	9C	£
157	9D	¥
158	9E	₧
159	9F	ƒ

Dec	Hex	Char
160	A0	á
161	A1	í
162	A2	ó
163	A3	ú
164	A4	ñ
165	A5	Ñ
166	A6	ª
167	A7	º
168	A8	¿
169	A9	⌐
170	AA	¬
171	AB	½
172	AC	¼
173	AD	¡
174	AE	«
175	AF	»
176	B0	▒
177	B1	▒
178	B2	▒
179	B3	│
180	B4	┤
181	B5	╡
182	B6	╢
183	B7	╖
184	B8	╕
185	B9	╣
186	BA	║
187	BB	╗
188	BC	╝
189	BD	╜
190	BE	╛
191	BF	┐

Dec	Hex	Char
192	C0	└
193	C1	┴
194	C2	┬
195	C3	├
196	C4	─
197	C5	┼
198	C6	╞
199	C7	╟
200	C8	╚
201	C9	╔
202	CA	╩
203	CB	╦
204	CC	╠
205	CD	═
206	CE	╬
207	CF	╧
208	D0	╨
209	D1	╤
210	D2	╥
211	D3	╙
212	D4	╘
213	D5	╒
214	D6	╓
215	D7	╫
216	D8	╪
217	D9	┘
218	DA	┌
219	DB	▒
220	DC	▒
221	DD	▒
222	DE	▒
223	DF	▒

Dec	Hex	Char
224	E0	α
225	E1	ß
226	E2	Γ
227	E3	π
228	E4	Σ
229	E5	σ
230	E6	µ
231	E7	τ
232	E8	Φ
233	E9	Θ
234	EA	Ω
235	EB	δ
236	EC	∞
237	ED	φ
238	EE	ε
239	EF	∩
240	F0	≡
241	F1	±
242	F2	≥
243	F3	≤
244	F4	⌠
245	F5	⌡
246	F6	÷
247	F7	≈
248	F8	°
249	F9	∙
250	FA	·
251	FB	√
252	FC	ⁿ
253	FD	²
254	FE	■
255	FF	

† ASCII code 127 has the code DEL. Under MS-DOS, this code has the same effect as ASCII 8 (BS).
The DEL code can be generated by the CTRL + BKSP key.

Declare Sub

Déclare une procédure utilisateur.

Syntaxe:

DECLARE SUB MaSub[([mode] var as type)]

Mode: Méthode de transfert des variables à la fonction.

BYVAL pour passer une copie de la variable; cette variable ne sera pas modifiée par la SUB.

BYREF pour passer l'adresse de la variable; si cette variable est modifiée par la SUB, elle restera modifiée au retour.

Var: Variable(s) transmise(s) à la SUB.

Type: Type des variables transmises et du résultat, Bytes, Integer, Word, long, Single, String ou tableaux.

Declare Sub

- Les bits sont des variables type global; ils ne peuvent pas être transmis à la Sub.
- La Sub est assimilée à une nouvelle instruction; elle n'est pas dimensionnée.
- En revanche les variables transférées doivent être dimensionnées.
- On termine la rédaction d'une Sub par : End Sub.
- Les fonctions s'écrivent après le End du programme.
- La Sub réalise un ensemble d'instructions répétitives.

Declare Sub

```
  Declare Sub MemErasePage(ByVal Page As Word)
  Declare Sub MemReadBuff(byval AddDebut As Word , Byval Taille As Word , Byval Buffer1 As Byte)


    Do

      Call MemErasePage(3)
      MemReadBuff(3)
    Loop

End

'_____________________________________________
Sub MemErasePage(Page As Word)
    Reset Cs                    'Enable la RAM
    waitms 1
    Spiout Tmeperasepage , 1
    Spiout Tmeppageh , 1
    Spiout Tmeppagel , 1
    Spiout Tmepdontcare , 1
    Set Cs                      'Disable la RAM
    Waitms 12                   'Delais pour le TRF
End Sub


Sub MemReadBuff(byval Adddebut As Word , Byval Taille As Word , Byval Buffer1 As Byte)
Local Readbuff As Byte

    Reset Cs
     waitms 1
    Spiout Readbuff , 1
    Spiout Dontcare , 1
    Spiout Pageh , 1
    Spiout Pagel , 1
    Spiout Dontcare , 1
    waitms 1                    'Disable la RAM
    Set Cs
End Sub
```

Declare Function

Déclare une fonction utilisateur.

Syntaxe:

DECLARE FUNCTION MaFunction([([mode] var as type)]) as Type

La fonction **retourne** une valeur, contrairement à la Sub qui retourne rien.

Mode Méthode de transfert des variables à la fonction.

BYVAL pour passer une copie de la variable, cette variable ne sera pas modifiée par la Fonction.

BYREF pour passer l'adresse de la variable, si cette variable est modifiée par la Fonction, elle restera modifiée au retour.

Var Variable(s) transmise(s) à la Fonction.

Type: Type des variables transmises et du résultat, Bytes, Integer, Word, long, Single, String ou tableaux.

Declare Function

- Les bits sont des variables type global. Ils ne peuvent pas être transmis à la Fonction.

- La Fonction est assimilée à une nouvelle instruction, elle n'est pas dimensionnée.

- En revanche les variables transférées doivent être dimensionnée.

- On termine la rédaction d'une Fonction par : End Function.

- Les fonctions s'écrivent après le End du programme.

- La Fonction réalise un ensemble d'instructions répétitives et retoure une valeur.

Declare Function

```
  Declare Function MemErasePage(ByVal Page As Word) As Word
  Declare Function MemReadBuff(byval AddDebut As Word , Byval Taille As Word , Byval Buffer1 As Byte) As Byte

    Do
      If MemErasePage(3) = 4 Then
        Timer1 = 36700
      End If

      If MemReadBuff(3) = 124 Then
        Timer1 = 0
      End If
    Loop
End

'_____________________________________________
Function MemErasePage(Page As Word) As Word
    Reset Cs                        'Enable la RAM
    waitms 1
    Spiout Tmeperasepage , 1
    Spiout Tmeppageh , 1
    Spiout Tmeppagel , 1
    Spiout Tmepdontcare , 1
    Set Cs                          'Disable la RAM
    Waitms 12                       'Delais pour le TRF
    MemErasePage = Tmeppageh
End Function

Function MemReadBuff(byval Adddebut As Word , Byval Taille As Word , Byval Buffer1 As Byte) As Byte
Local Readbuff As Byte
    Reset Cs
     waitms 1
    Spiout Readbuff , 1
    Spiout Dontcare , 1
    Spiout Pageh , 1
    Spiout Pagel , 1
    Spiout Dontcare , 1
    waitms 1                        'Disable la RAM
    Set Cs
    MemReadBuff = Readbuff
End Function
```

By VAL / By Ref

Spécifie que la variable est transmise par valeur(Byval) ou par référence(Byref).

Syntaxe:

Sub test(Byval var)

Sub test(*Byref* var)

Sub test(var) ‘ byRef est utilisé par default si pas spécifiée.

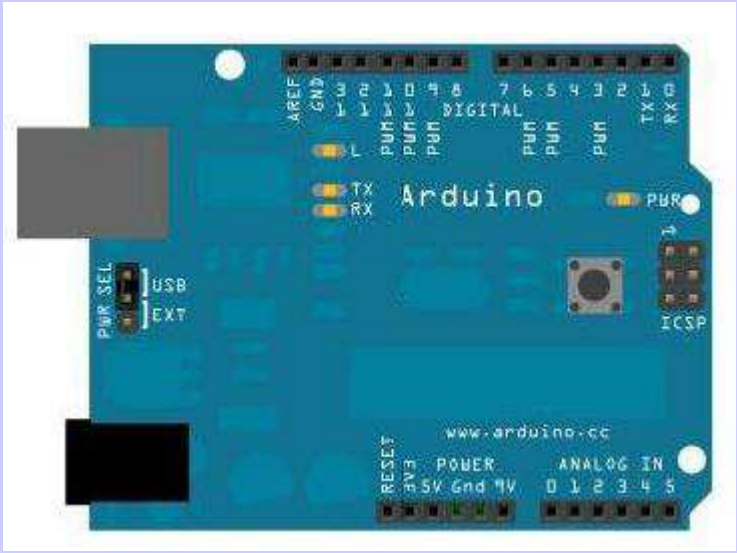
Var: Nom de variable

- **BYVAL** Var n'est pas transformée par la SUB ou la Function.

- **BYREF** : c'est par l'intermédiaire de son adresse (la référence) que la variable est transmise, donc elle sera transformée par la SUB ou la Function. C’est l’adresse mémoire qui est utilisée.

Exercise #7

Recevoir des données sur le port série en utilisant l'Interruption. Un par un, les caractères devront être retournés sur le port série et ce, dès que ceux-ci sont reçus avec un entête.



Aide #7

Nouvelles commandes à utiliser :

Configuration :
On Urxc ...
Enable ...
Enable ...

Code :
Chr()
Print

Code #7

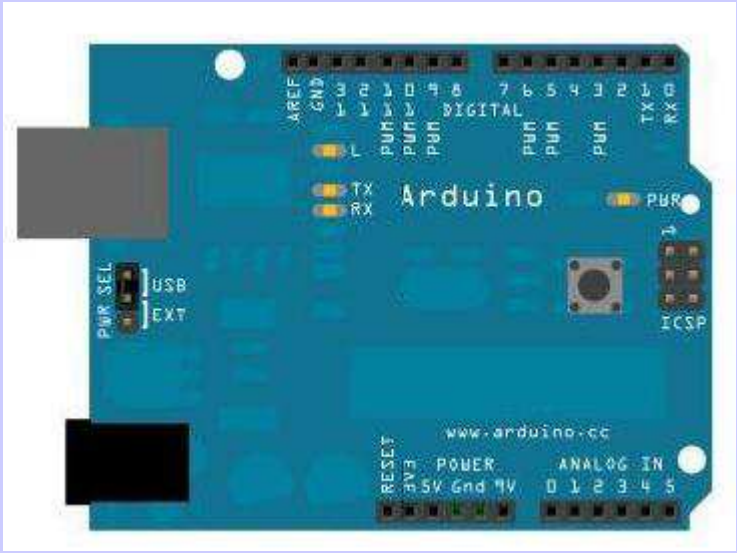
```
'////////////////////////////////////////////////////////////////////
'
'  By Steve Tremblay
'
'  Cours Arduino Uno R3
'
'  Revision 1.00
'
'  ____________
'
'
'
'////////////////////////////////////////////////////////////////////


   $regfile = "m328pdef.dat"                    'Assigne le DEF du Micro

   $HwStack = 48                               'Valeur par default
   $SwStack = 32                               'Valeur par default
   $FrameSize = 32                              'Valeur par default
   $Crystal = 16000000                            'Vitesse du crystal

   $Baud = 9600                                'Vitesse du port serie



'------------------------------------------------------------
' Configuration le PORT SERIE
'_____________________________
   On Urxc RxData                              'RX data du Port Serie
'------------------------------------------------------------


'------------------------------------------------------------
' Declare les VARIABLES
'_____________________________
   Dim bByteRcv as Byte
   Dim bByteData as Byte
'------------------------------------------------------------
```

```
'------------------------------------------------------------
' Demarre les peripheriques
'_____________________________
   Enable Interrupts
   Enable Urxc

'------------------------------------------------------------
'Fin de la configuration du Micro
'_____________________________

     Do
        If bByteRcv = 1 Then
          bByteRcv = 0
          Print "Rcv Byte : " ;  Chr(bByteData) ; "  Valeur : " ; bByteData
        End If
     Loop


   End
'************************************************************



'////////////////////////////////////////////////////////////////////
'////////////////////////////////////////////////////////////////////
'////////////////////////////////////////////////////////////////////

'_____________________________
RxData:
   bByteRcv = 1
   bByteData = UDR
Return
```

Exercise #8

Recevoir des données sur le port série en utilisant l'Interruption. Après avoir reçu 5 caractères, nous devrons les retournés ensemble sur le port série avec un entête.

Tips : Dimensionner une variable String d'au minimum la taille nécessaire.



Aide #8

Nouvelle commande à utiliser :

Configuration :
Pas de nouvelle

Code :
+ pour l'ajout des caractères dans une String.

Code #8

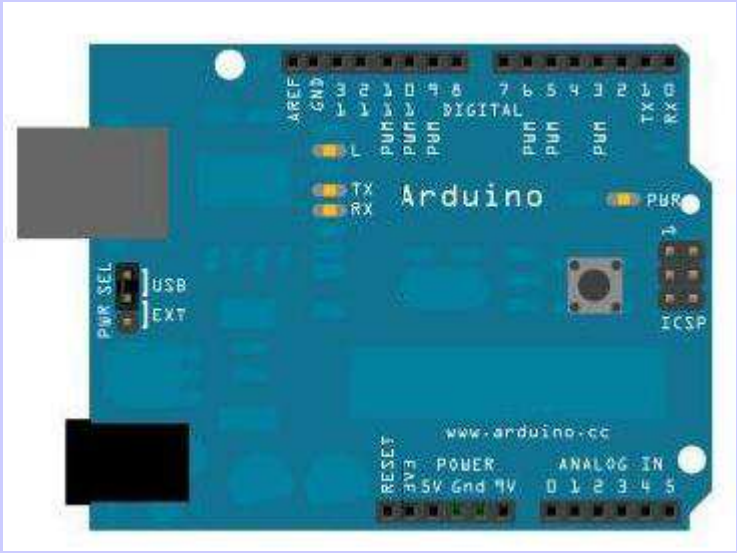
```
  $regfile = "m328pdef.dat"                    'Assigne le DEF du Micro

  $HwStack = 48                                'Valeur par default
  $SwStack = 32                                'Valeur par default
  $FrameSize = 32                              'Valeur par default
  $Crystal = 16000000                          'Vitesse du crystal

  $Baud = 9600                                 'Vitesse du port serie

'------------------------------------------------------------
' Configuration le PORT SERIE
'____________________________
  On Urxc RxData                               'RX data du Port Serie
'------------------------------------------------------------

'------------------------------------------------------------
' Declare les VARIABLES
'____________________________
  Dim bByteRcv as Byte
  Dim bByteData as Byte
  Dim bLenStr as Byte
  Dim sStrData as string * 20                  '
'------------------------------------------------------------

'------------------------------------------------------------
' Demarre les peripheriques
'____________________________
  Enable Interrupts
  Enable Urxc

'Fin de la configuration du Micro
'____________________________
```

```
    Do
        If bByteRcv = 1 Then
          bByteRcv = 0
          bLenStr = bLenStr + 1
          sStrData = sStrData + Chr(bByteData)
            If bLenStr = 5 Then
              bLenStr = 0
              Print "Voici les donnees : " ; sStrData
              sStrData = ""
            End If
        End If
    Loop


  End
'************************************************************



'____________________________
RxData:
  bByteRcv = 1
  bByteData = UDR
Return
```

Exercise #9

Utiliser une **SUB** afin d'envoyer les caractères reçus sur le port série. Fournir le nombre de caracères à la **SUB** afin de déclancher l'envoie des **BYTES**.

Raison : Simplifier le **CODE** et améliorer le déverminage (débugage).



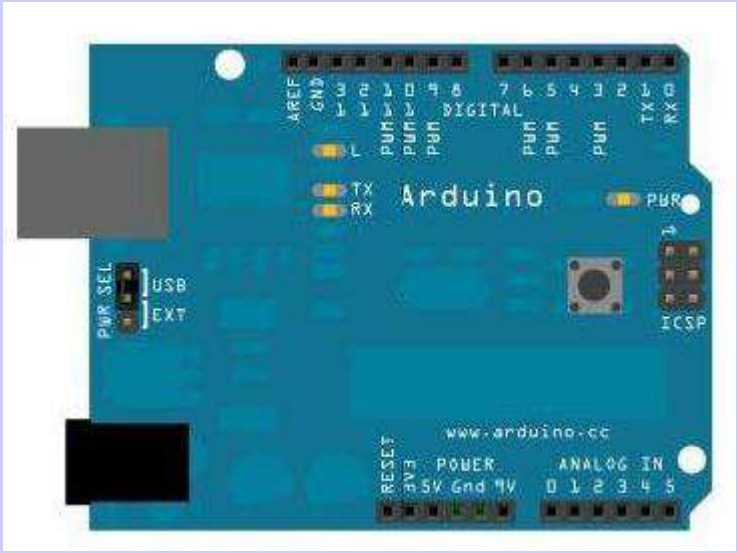
Code #9

```
'////////////////////////////////////////////////////////////////
  $regfile = "m328pdef.dat"                    'Assigne le DEF du Micro

  $HwStack = 48                                'Valeur par default
  $SwStack = 32                                'Valeur par default
  $FrameSize = 32                              'Valeur par default
  $Crystal = 16000000                          'Vitesse du crystal

  $Baud = 9600                                 'Vitesse du port serie

'================================================================
' Configuration le PORT SERIE
'----------------------------------------------------------------
  On Urxc RxData                               'RX data du Port Serie

' Declare les VARIABLES
'----------------------------------------------------------------
  Dim bByteRcv as Byte
  Dim bByteData as Byte
  Dim bLenStr as Byte
  Dim sStrData as string * 20                  '

' Declare les SUB
'----------------------------------------------------------------
  Declare Sub EnvoieBytes(ByVal NbrByte as Byte)
'================================================================

' Demarre les peripheriques
'----------------------------------------------------------------
  Enable Interrupts
  Enable Urxc

    Do
        If bByteRcv = 1 Then
          bByteRcv = 0
          bLenStr = bLenStr + 1
          sStrData = sStrData + Chr(bByteData)
          Call EnvoieBytes(5)
        End If
    Loop

  End
```

```
'////////////////////////////////////////////////////////////////
'////////////////////////////////////////////////////////////////
'////////////////////////////////////////////////////////////////

'----------------------------------
RxData:
  bByteRcv = 1
  bByteData = UDR
Return



'////////////////////////////////////////////////////////////////
'////////////////////////////////////////////////////////////////
'////////////////////////////////////////////////////////////////
'----------------------------------
Sub EnvoieBytes(ByVal NbrByte as Byte)

    If bLenStr = NbrByte Then
      bLenStr = 0
      Print "Voici les donnees : " ; sStrData
      sStrData = ""
    End If

End Sub
```

Exercise #10

Utiliser une **FUNCTION** afin d'envoyer les caractères reçus sur le port série. Fournir le nombre de caractères à la SUB afin de déclancher l'envoie des **BYTES** et recevoir le résultat dans une **STRING**.
Raison : Simplifier le **CODE** et améliorer le déverminage (débugage).



Code #10

```
$regfile = "m328pdef.dat"                    'Assigne le DEF du Micro

  $HwStack = 48                              'Valeur par default
  $SwStack = 32                              'Valeur par default
  $FrameSize = 32                            'Valeur par default
  $Crystal = 16000000                        'Vitesse du crystal
  $Baud = 9600                               'Vitesse du port serie

' Configuration le PORT SERIE
'-----------------------------------------
  On Urxc RxData                             'RX data du Port Serie

' Declare les VARIABLES
'-----------------------------------------
  Dim bByteRcv as Byte
  Dim bByteData as Byte
  Dim bLenStr as Byte
  Dim sStrData as string * 20                '

' Declare les Function
'-----------------------------------------
  Declare Function EnvoieBytes(ByVal NbrByte as Byte) as String

' Demarre les peripheriques
'-----------------------------------------
  Enable Interrupts
  Enable Urxc

'Fin de la configuration du Micro
'-----------------------------------------
    Do
        If bByteRcv = 1 Then
          bByteRcv = 0
          bLenStr = bLenStr + 1
          sStrData = sStrData + Chr(bByteData)
            If EnvoieBytes(5) <> "" Then
              Print EnvoieBytes (5)
              bLenStr = 0
              sStrData = ""
            End If
        End If
    Loop
  End
'*****************************************************
*********

'/////////////////////////////////////////////////////////////
'/////////////////////////////////////////////////////////////
'/////////////////////////////////////////////////////////////
'-----------------------------------------
RxData:
  bByteRcv = 1
  bByteData = UDR
Return



'/////////////////////////////////////////////////////////////
'/////////////////////////////////////////////////////////////
'/////////////////////////////////////////////////////////////
'-----------------------------------------
Function EnvoieBytes(ByVal NbrByte as Byte) as String

    If bLenStr = NbrByte Then
      EnvoieBytes = "Voici les donnees : " + sStrData
    Else
      EnvoieBytes = ""
    End If

End Function
```

Fin