

```

' {$STAMP BS2}                                ' Directives Stamp
' {$PBASIC 2.5}                               ' Directives Pbasic
'
'
'
' But du programme: Après deux décomptes de 5 à 0, le robot parcourt un carré de 20cm.
'                               Le robot est équipé de capteurs Infra-rouges pour d'éventuels obstacles.
'                               Un piezzo émettra un son lorsque les capteurs détecteront un obstacle.
'                               Un bouton poussoir a été placé pour mettre fin au programme.
' Nom du programme: EAC4Mcr7segment

' ----- [ Informations ] -----
' --- Afficheur à 7 segments ---
' Segment a = 7, branché sur la sortie du µc 11
' Segment b = 6, branché sur la sortie du µc 10
' Segment c = 4, branché sur la sortie du µc 9
' Segment d = 2, branché sur la sortie du µc 8
' Segment e = 1, branché sur la sortie du µc 7
' Segment f = 9, branché sur la sortie du µc 6
' Segment g =10, branché sur la sortie du µc 5
' --- LDR ---
' Oeil droit connecté sur IN4
' Oeil gauche connecté sur IN1
' --- Reset ---
' Un bouton a été placé sur la IN 0. Si le bouton est pressé, le robot et son appareillage se
stoppent.

' Piezzo branché sur IN3

' ----- [ Déclaration des variables ] -----
I VAR Byte      ' I = variable en byte, place réservée dans la mémoire

' ----- [ Programme ] -----
Debut1:
DEBUG "Début du programme", CR                ' Affiche un message à l'écran, retour à
la ligne
DEBUG "Affichage à 7 segments - Parcours 20cm",CLS,          ' Affiche un message à l'écran,
effacage de l'écran

      "Etat du bouton poussoir",CR      ' Affiche un message à l'écran, retour à la ligne

      DEBUG BIN1 IN0                    ' Affiche l'état du bouton poussoir en binaire ( 0 ou
1)

IF ( IN0 = 0) THEN
  GOSUB Debut1      ' Effectue un reset complet du programme
ENDIF      ' Fin de L' IF

IF ( IN0 = 1) THEN
  ' Si IN0=0 ( BP non déclenché), alors
  FOR I = 1 TO 2
  ' Crée une boucle, qui se répètera deux fois
  GOSUB A50
  ' Fais appel à la macro A50
  GOSUB A_T
  ' Fais appel à la macro A_T
  NEXT
  ' Retour au "FOR" en ajoutant 1 à la valeur
ENDIF
' Fin de l' IF

IF ( IN0 = 1) THEN
  ' Crée une boucle, qui se répètera quatre fois
  FOR I = 1 TO 4
  GOSUB M_GD
  ' Appel à la macro M_GD
  NEXT
  ' Retour au "FOR" en ajoutant 1 à la valeur
ENDIF

IF ( IN4 = 0) THEN
  ' Si IN4=0 (oeil droit détectant de l'ombre), alors
  FREQOUT 3, 500, 5500      ' Une fréquence de 5500 Hz est émise pendant 500 millisecondes sur la

```

```

borne 3
  GOSUB M_OD          ' Appel à la macro M_OD
  PAUSE 10           ' Pause de 10 millisecondes
ELSEIF ( IN1 =0) THEN ' Si IN1=0 (oeil gauche détectant de l'ombre), alors
  FREQOUT 3, 500, 5500 ' Une fréquence de 5500 Hz est émise pendant 500 millisecondes sur la
borne 3
  GOSUB M_0G          ' Appel à la macro M_0G
  PAUSE 10           ' Pause de 10 millisecondes
ENDIF                ' Fin de l' IF
DEBUG CLS

DEBUG " Etat des capteurs infra-rouges", CR, ' Affiche un message à l'écran et retour à la
ligne "Gauche Droite",CR, ' Affiche un message à l'écran et retour à la
ligne "....." ' Affiche un message à l'écran
DO ' Boucle
  DEBUG CRSRXY, 0,3, ' Place le curseur aux coordonnées x=0 et Y=3 de l'écran
  "P4= ", BIN1 IN4 ' Affiche la valeur en binaire de l'entrée IN4 à l'écran
  DEBUG "P1= ", BIN1 IN1 ' Affiche la valeur en binaire de l'entrée IN1 à l'écran
  PAUSE 50 ' Pause de 50 millisecondes
LOOP ' Renvoi au "Do", boucle sans fin

DEBUG CLS

RETURN

END ' Fin du programme

```

' ----- [ Macros ] -----

A50:

```

DEBUG CR, "Affichage du chiffre 5" ' affichage du chiffre 5 à l'écran"
HIGH 11 ' allume la LED en sortie 15
PAUSE 5
LOW 10 ' éteint la LED en sortie 14
PAUSE 5
HIGH 9
PAUSE 5
HIGH 8
PAUSE 5
LOW 7
PAUSE 5
HIGH 6
PAUSE 5
HIGH 5
PAUSE 750

DEBUG CR, "Affichage du chiffre 4" ' affichage du chiffre 4 à l'écran"
LOW 11
PAUSE 5
HIGH 10
PAUSE 5
HIGH 9
PAUSE 5
LOW 8
PAUSE 5
LOW 7
PAUSE 5
HIGH 6
PAUSE 5
HIGH 5
PAUSE 750

```

```

DEBUG CR, "Affichage du chiffre 3" ' affichage du chiffre 3 à l'écran"
HIGH 11
  PAUSE 5
HIGH 10
  PAUSE 5
HIGH 9
  PAUSE 5
HIGH 8
  PAUSE 5
LOW 7
  PAUSE 5
LOW 6
  PAUSE 5
HIGH 5
  PAUSE 750

```

```

DEBUG CR, "Affichage du chiffre 2" ' affichage du chiffre 2 à l'écran"
HIGH 11
  PAUSE 5
HIGH 10
  PAUSE 5
LOW 9
  PAUSE 5
HIGH 8
  PAUSE 5
HIGH 7
  PAUSE 5
LOW 6
  PAUSE 5
HIGH 5
  PAUSE 750

```

```

DEBUG CR, "Affichage du chiffre 1" ' affichage du chiffre 1 à l'écran"
LOW 11
  PAUSE 5 ' Pause de 5 millisecondes
HIGH 10
  PAUSE 5 ' Pause de 5 millisecondes
HIGH 9
  PAUSE 5 ' Pause de 5 millisecondes
LOW 8
  PAUSE 5 ' Pause de 5 millisecondes
LOW 7
  PAUSE 5 ' Pause de 5 millisecondes
LOW 6
  PAUSE 5 ' Pause de 5 millisecondes
LOW 5
  PAUSE 750 ' Pause de 750 millisecondes

```

```

DEBUG CR, "Affichage du chiffre 0" ' affichage du chiffre 0 à l'écran"
HIGH 11
  PAUSE 5 ' Pause de 5 millisecondes
HIGH 10
  PAUSE 5 ' Pause de 5 millisecondes
HIGH 9
  PAUSE 5 ' Pause de 5 millisecondes
HIGH 8
  PAUSE 5 ' Pause de 5 millisecondes
HIGH 7
  PAUSE 5 ' Pause de 5 millisecondes
HIGH 6
  PAUSE 5 ' Pause de 5 millisecondes
LOW 5
  PAUSE 750 ' Pause de 750 millisecondes

```

```

RETURN ' retour de sous-programme, le programme reprends après le GOSUB

```

```

A_T:
DEBUG CR, "Arrêt total"      ' Arrêt complet. L'afficheur s'éteint, les servos se stoppent...
  LOW 11                      ' Envoi d'un signal négatif à la borne 1, le segment s'éteint
    PAUSE 5                    ' Pause de 5 millisecondes
  LOW 10                      ' Envoi d'un signal négatif à la borne 2, le segment s'éteint
    PAUSE 5                    ' Pause de 5 millisecondes
  LOW 9                       ' Envoi d'un signal négatif à la borne 3, le segment s'éteint
    PAUSE 5                    ' Pause de 5 millisecondes
  LOW 8                       ' Envoi d'un signal négatif à la borne 4, le segment s'éteint
    PAUSE 5                    ' Pause de 5 millisecondes
  LOW 7                       ' Envoi d'un signal négatif à la borne 5, le segment s'éteint
    PAUSE 5                    ' Pause de 5 millisecondes
  LOW 6                       ' Envoi d'un signal négatif à la borne 7, le segment s'éteint
    PAUSE 5                    ' Pause de 5 millisecondes
  LOW 5                       ' Envoi d'un signal négatif à la borne 6, le segment s'éteint
    PAUSE 5                    ' Pause de 5 millisecondes
RETURN

M_GD:
                                ' Allumage des deux moteurs, le robot avance de 20 cm
FOR I = 1 TO 30                 ' crée une boucle qui se répétera 30 fois
  PULSOUT 12, 500              ' impulsion port-moteur 12, 500x 2µs, moteur gauche / Sens horlogique
  PAUSE 10                     ' Pause 10 millisecondes
  PULSOUT 13, 1000            ' impulsion port moteur 13, 1000x 2µs, moteur droit / Sens anti
horlogique
  PAUSE 10                     ' Pause 10 millisecondes
NEXT                             ' renvoi la boucle

                                ' le robot tourne à droite de 90°
FOR I = 1 TO 15                 ' crée une boucle qui se répétera 15 fois ( 15 -> 90° )
  PULSOUT 12, 1000            ' impulsion port-moteur 12, 1000x 2µs, moteur gauche / Sens horlogique
  PAUSE 20                     ' Pause 20 millisecondes
  PULSOUT 13, 1000            ' impulsion port moteur 13, 1000x 2µs, moteur droit / Sens horlogique
  PAUSE 20                     ' Pause 20 millisecondes
NEXT                             ' renvoi la boucle
RETURN                           ' retour de sous-programme, le programme reprends après le GOSUB

M_OD:
                                ' Si la IR droite détecte un obstacle, le robot tournera
                                ' de quelques degré à gauche
FOR I = 1 TO 15                 ' Crée la boucle qui se répétera 15 fois
  PULSOUT 12, 500             ' Envoi d'une impulsion de 500x 2µs à la borne 12
  PAUSE 10                    ' Pause de 10 millisecondes
  PULSOUT 13, 500             ' Envoi d'une impulsion de 500x 2µs à la borne 13
  PAUSE 10                    ' Pause de 10 millisecondes
NEXT

M_OG:
                                ' Si la IR gauche détecte un obstacle, le robot tournera
                                ' de quelques degré à droite
FOR I = 1 TO 15                 ' Crée une boucle qui se répétera 15 fois
  PULSOUT 12, 1000            ' Envoi d'une impulsion de 1000x 2µs à la borne 12
  PAUSE 20                    ' Pause 20 millisecondes
  PULSOUT 13, 1000            ' Envoi d'une impulsion de 1000x 2µs à la borne 13
  PAUSE 20                    ' Pause 20 millisecondes
NEXT                             ' Retour au " FOR" en ajoutant 1 à la valeur

```