



Exemples BoeBot

Joan Oliver
Setembre'04

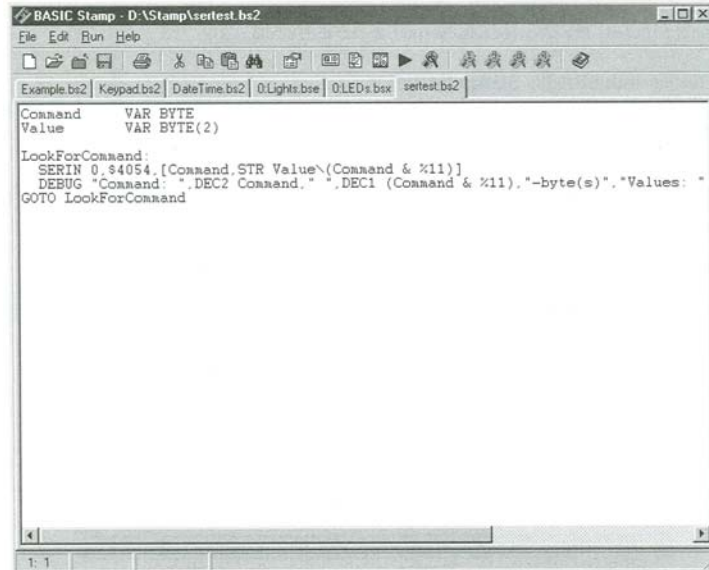


Índex

- L'entorn d'aplicació
- Placa de desenvolupament
- Actuació sobre motors
- Actuació sobre piezobuzzer
- Controlant la distància
- Navegació tàctil
- Detecció per infraroigs

Joan Oliver – Setembre 2004

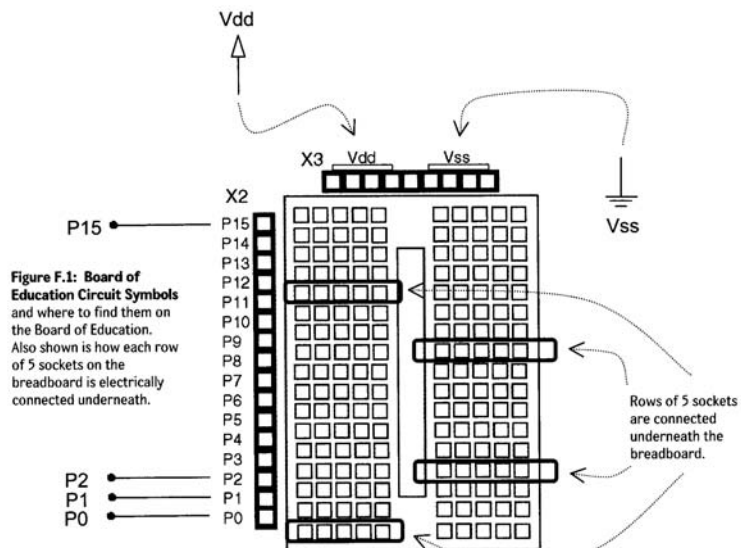
L'entorn d'aplicació



```
BASIC Stamp - D:\AStamp\seritest.bs2
File Edit Run Help
Example.bs2 | Keypad.bs2 | DateTime.bs2 | 0Lights.bs2 | 0LEDs.bs2 | seritest.bs2
Command  VAR BYTE
Value     VAR BYTE(2)
LookForCommand:
SERIN 0,$4054,[Command,STR Value\{Command & %11}]
DEBUG "Command: ",DEC2 Command," ",DEC1 (Command & %11),"-byte(s)", "Values: "
GOTO LookForCommand
```

Joan Oliver – Setembre 2004

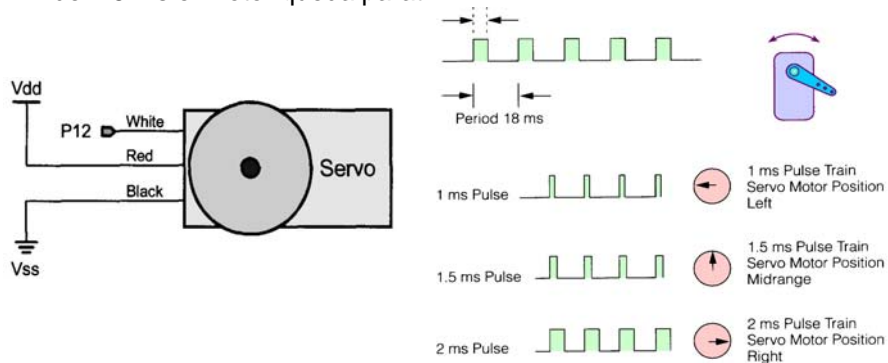
Placa de desenvolupament



Joan Oliver – Setembre 2004

Actuació sobre motors (I)

- Motors controlats per un tren de polsos.
- El microcontrolador emprà una sortida per generar aquest pols.
- Amb polsos de 1 ms el motor gira amb el sentit de les agulles del motor. Amb polsos de 2 ms gira en el sentit contrari. Amb polsos de 1.5 ms el motor queda parat.



Joan Oliver – Setembre 2004

Actuació sobre motors (II)

- Sabent que el període del microcontrolador és de 2 μ s, es calcula fàcilment que cal executar 750 cicles del microcontrolador per a tenir un pols de 1.5 ms (que atura el motor).
- En conseqüència, la comanda del programa que atura el motor és:

Pulsout 12, 500

- El primer camp de la comanda fa referència al port de sortida del microcontrolador que controlarà el motor. El segon és la durada del pols.
- Què fa el següent programa? Com aturarem els motors?

```
' Robotics! v1.5, Program Listing 1.5: Full Speed Ahead - both servos.
' {$Stamp bs2}                               ' Stamp Directive.

low 12                                         ' Set P12 to output-low.
low 13                                         ' Set P13 to output-low.

loop:                                          ' Label for "goto loop to return to"

  pulsout 12, 500                              ' Send 1.0 ms pulses to P12
  pulsout 13, 1000                            ' Send 2.0 ms pulses to P13
  pause 20                                    ' every 20 ms.

goto loop                                     ' Send program to "loop: " label.
```

Joan Oliver – Setembre 2004

Actuació sobre motors (III)

- ... arrencant i parant!

```
' Robotics! v1.5, Program Listing 2.4: Ramping for Start and Stop
' {$Stamp bs2}                                ' Stamp Directive.

'-----Declarations-----
pulse_count var word                          ' For...next loop counter.
right_width var word                          ' Variable stores right pulse width.
left_width var word                            ' Variable stores left pulse width.

'-----Initialization-----
output 2                                       ' Set P2 to output.
freqout 2, 2000, 3000                          ' Signal program is starting/restarting.
low 12                                         ' Set P12 and 13 to output-low.
low 13

'-----Main Routine-----
main:                                          ' Main routine.
  ramp_up_forward:                             ' Routine ramps into forward motion.
    for pulse_count = 0 to 250 step 2         ' For loop counts up in steps of 2.
      pulsout 12, 750 - pulse_count           ' Pulse sent is 1.5 ms - pulse_count value.
      pulsout 13, 750 + pulse_count           ' Pulse sent is 1.5 ms + pulse_count value.
      pause 20                                 ' Pause for 20 ms.
    next

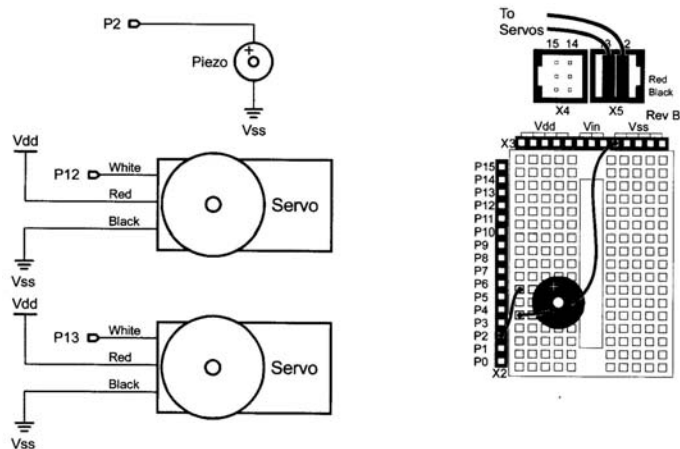
  forward:                                     ' Forward routine.
    for pulse_count = 1 to 100                ' Loop that sends 100 forward pulses.
      pulsout 12, 500                          ' 1.0 ms pulse to right servo.
      pulsout 13, 1000                          ' 2.0 ms pulse to left servo.
      pause 20                                 ' Pause for 20 ms.
    next

  ramp_down_forward:                           ' Routine ramps out of forward motion.
    for pulse_count = 250 to 0 step 2         ' For loop counts down in steps of 2.
      pulsout 12, 750 - pulse_count           ' Pulse sent is 1.5 ms - pulse_count value.
      pulsout 13, 750 + pulse_count           ' Pulse sent is 1.5 ms + pulse_count value.
      pause 20                                 ' Pause for 20 ms.
    next

  stop                                         ' Stop until reset.
```

Actuació sobre el piezobuzzer

- El piezobuzzer oscil·la amb un senyal freqüencial. La comanda és:
output 2
freqout 2, 2000, 3000 'Senyal a 3 KHz durant 2 segons



Joan Oliver – Setembre 2004

Controlant la distància

```
' Robotics! v1.5, Program Listing 2.2: Controlling Distance
' {$Stamp bs2}                               ' Stamp Directive.

'-----Declarations-----

pulse_count var word                          ' Declare a variable for counting.

'-----Initialization-----

output 2                                       ' Set P2 to output.
freqout 2, 2000, 3000                         ' Signal program is starting/restarting.
low 12                                         ' Set P12 and 13 to output-low.
low 13

'-----Main Routine-----

main:

  forward:                                    ' Forward routine.
    for pulse_count = 1 to 100                ' Loop that sends 100 forward pulses.
      pulsout 12, 500                          ' 1.0 ms pulse to right servo.
      pulsout 13, 1000                        ' 2.0 ms pulse to left servo.
      pause 20                                 ' Pause for 20 ms.
    next

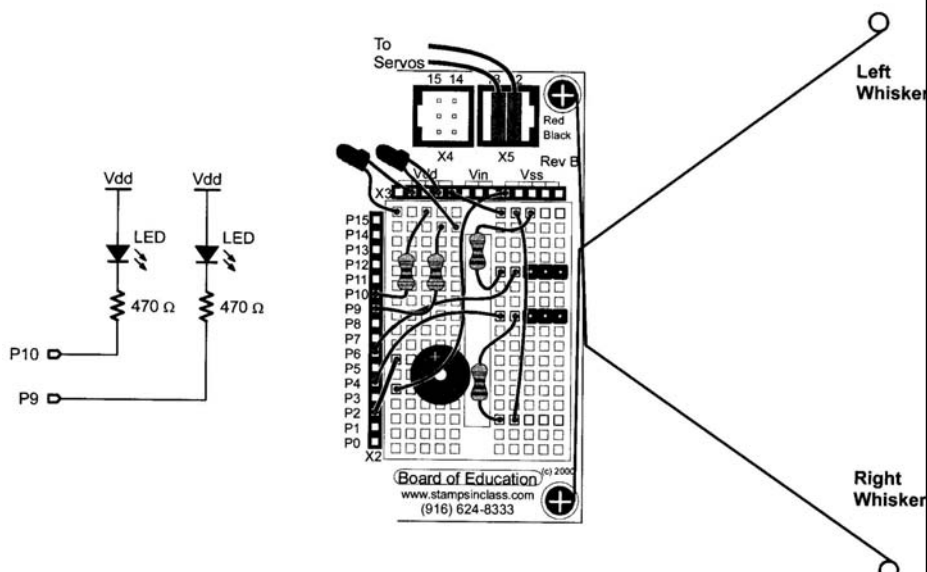
  stop                                         ' Stop until reset.
```

- Com anar enrera? Com girar a la dreta? Com girar a l'esquerra? Com accelerar a poc a poc?

Joan Oliver – Setembre 2004

Navegació tàctil (I)

- Llums (pins 9 i 10) i bigotis (pins 4 i 6)



Navegació tàctil (II)

```
' Robotics! v1.5, Program Listing 3.2: Roaming with Whiskers.
' ($Stamp bs2) ' Stamp Directive.

'----- Declarations -----
pulse_count var byte ' For...next loop counter.

'----- Initialization -----
output 2 ' Set P2 to output.
freqout 2, 2000, 3000 ' Start/restart signal.
low 12 ' Set P12 and 13 to output-low.
low 13

'----- Main Routine -----
main:
check_whiskers: ' Check each whisker.
  if in6 = 0 and in4 = 0 then u_turn ' Backwards if both switches close.
  if in6 = 0 then right_turn ' Right if left switch closes.
  if in4 = 0 then left_turn ' Left if right switch closes.

forward: ' If no detect, one forward pulse.
  pulsout 12,500
  pulsout 13,1000
  pause 20

goto main ' Check again.

'----- Navigation Routines -----
left_turn: ' Left turn routine.
  gosub backward ' Call Backward: before turning.
  for pulse_count = 0 to 35
    pulsout 12, 500
    pulsout 13, 500
    pause 20
  next
  goto main

right_turn: ' Right turn routine.
  gosub backward ' Call Backward: before turning.
  for pulse_count = 0 to 35
    pulsout 12, 1000
    pulsout 13, 1000
    pause 20
  next
  goto main

u_turn: ' U-turn routine.
  gosub backward ' Call Backward: before turning.
  for pulse_count = 0 to 70
    pulsout 12, 500
    pulsout 13, 500
    pause 20
  next
  goto main

'----- Navigation Subroutine -----
backward: ' Used by each navigation routine.
  for pulse_count = 0 to 70
    pulsout 12, 1000
    pulsout 13, 500
    pause 20
  next
  return
```

Navegació tàctil (III)

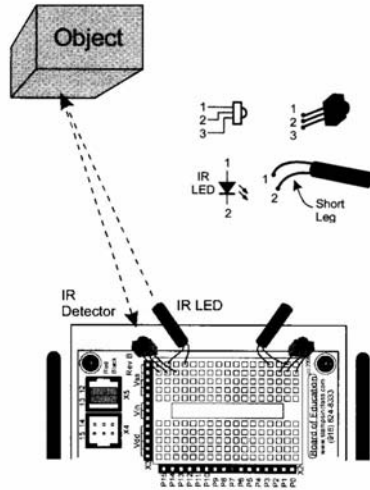
```
right_turn: ' Right turn routine.
  gosub backward ' Call Backward: before turning.
  for pulse_count = 0 to 35
    pulsout 12, 1000
    pulsout 13, 1000
    pause 20
  next
  goto main

u_turn: ' U-turn routine.
  gosub backward ' Call Backward: before turning.
  for pulse_count = 0 to 70
    pulsout 12, 500
    pulsout 13, 500
    pause 20
  next
  goto main

'----- Navigation Subroutine -----
backward: ' Used by each navigation routine.
  for pulse_count = 0 to 70
    pulsout 12, 1000
    pulsout 13, 500
    pause 20
  next
  return
```

Detecció per infraroigs (I)

- Mitjançant leds emissors i fototransistors detectors.



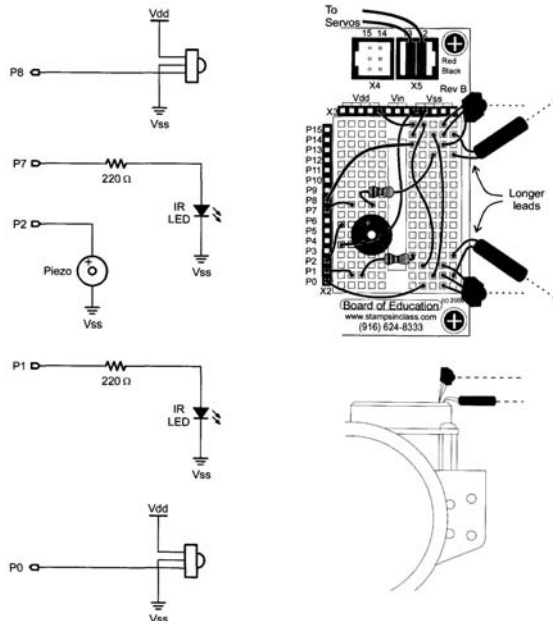
■ Funcionament:

- Sensors infraroigs sensibles a llum pròxima a 980 nm (l'ull humà capta llum des de 400 nm -violeta- a 780 nm -vermell-).
- El sensor també té una lent que només admet la llum infraroig que li arriba a través d'un filtre a 38 KHz. Això limita les interferències.
- Per tant, la instrucció a enviar serà

freqüent pin, període, 37500
període, per exemple, igual a 1 ms

Joan Oliver – Setembre 2004

Detecció per infraroigs (II)



Detecció per infraroigs (III)

```
' Robotics! v1.5, Program Listing 5.2: Roaming with Whiskers Adjusted for IR Pairs.
' {$Stamp bs2}                                ' Stamp Directive.

'----- Declarations -----
pulse_count var byte
left_IR_det var bit
right_IR_det var bit

' For...next loop counter.
' Two bit variables for saving IR
' detector output values.

'----- Initialization -----
output 2
output 7
output 1
freqout 2, 2000, 3000
low 12
low 13

' Set all I/O lines sending freqout
' signals to function as outputs
' Program start/restart signal.
' Set P12 and 13 to output-low.

'----- Main Routine -----
main:

freqout 7, 1, 38500
left_IR_det = in8

freqout 1, 1, 38500
right_IR_det = in0

' Detect object on the left.
' Send freqout signal - left IRLED.
' Store IR detector output in RAM.
' Detect object on the right.
' Repeat for the right IR pair.

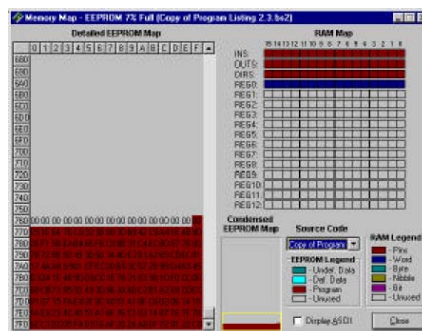
' With the exception that values stored in RAM are used instead of
' input register values, the decision making process is the same as
' the one used in Program Listing 3.2.

if left_IR_det = 0 and right_IR_det = 0 then u_turn
if left_IR_det = 0 then right_turn
if right_IR_det = 0 then left_turn
```

Joan Oliver – Setembre 2004

Ús de l'EEPROM per navegació (I)

- Tot i que l'EEPROM s'utilitza per emmagatzemar programes, es pot fer servir per guardar dades (per exemple, per navegar).
- El programa s'emmagatzema de les posicions 2047 a la 0. Les dades es posen de les posicions 0 a la 2047.
- Però compta per què l'EEPROM
 - Pot trigar varis ms en emmagatzemar dades
 - Té nombre finit de cicles d'escriptura (prop de 10M)
 - La seva missió principal és per guardar programes.



Joan Oliver – Setembre 2004

Ús de l'EEPROM per navegació (II)

- Ús de la directiva **data** per carregar el moviment a efectuar pel robot.

```
' Robotics! v1.5, Program Listing 2.5: EEPROM Navigation
' {$Stamp bs2}                               ' Stamp Directive.

'----- Declarations -----

' Label for the declarations routine.
pulse_count var word           ' Declare a variable named pulse_count.
EE_address  var byte           ' Stores & increments EEPROM address.
instruction  var byte           ' Stores instruction retrieved from EEPROM.

data        "FFFBBLFFRFQ"      ' List of Boe-Bot navigation instructions.

'----- Main Routine -----

main:                               ' Main routine label.
read EE_address, instruction        ' Read at EE_address & store in instruction
EE_address = EE_address + 1        ' Increment EE address for next read

if instruction = "F" then forward   ' Check for forward command.
if instruction = "B" then backward  ' Check for backward command.
if instruction = "R" then right_turn ' Check for right turn command.
if instruction = "L" then left_turn  ' Check for left turn command.

stop                                ' Stop executing commands until reset

'----- Navigation Routines -----

forward:                               ' Forward routine.
for pulse_count = 1 to 75            ' Send 75 forward pulses.
  pulsout 12, 500                    ' 1.0 ms pulse to right servo.
  pulsout 13, 1000                  ' 2.0 ms pulse to left servo.
  pause 20                          ' Pause for 20 ms.
next
goto main                            ' Send program back to the main.
```

Informació

- Manuals d'ús:

- Manual del Basic Stamp
- Llibre de l'estudiant



<http://www.cannic.uab.es/aplicacions/ProgramaArgo/Argo.htm>