

# Sesión Laboratorio de Robótica – Scorbob (III)

Durante esta tercera sesión realizamos un proyecto libre en conjunto con otro grupo, poniendo en práctica los conocimientos que hemos adquirido en las sesiones anteriores.

## Clasificación de piezas: madera o metal

El ejercicio que nos hemos propuesto realizar es un clasificador de piezas según sean de madera o de metal. Para ello nos servimos principalmente del sensor inductivo, que detecta las piezas de metal, activando su salida cuando la pieza es de metal.

El funcionamiento es el siguiente: las piezas, de madera y metal, están almacenadas a un lado de la mesa (el de nuestro grupo). En la cinta están puestos el sensor inductivo y un sensor óptico para detectar la presencia de las piezas. Nuestro robot va cogiendo piezas y las deja sobre la cinta. Cuando el robot deja la pieza en la cinta y se retira a una posición segura, la cinta se pone en marcha (el eje de la cinta lo controlamos nosotros). La cinta se ejecuta en un subprograma, CINTA, que se ejecuta hasta que se activa nuestra entrada 14. Es la correspondiente al sensor óptico. Lo único que hace el programa CINTA es activarla y moverla. Al activarse IN[14], salta un TRIGGER STOPC, que para la cinta, y si se ha detectado un objeto, activa nuestra salida OUT[6] para avisar al otro robot de que hay un objeto, y si además es de metal, activa nuestra salida OUT[5] para avisarlo también de eso.

Realizado eso, pueden suceder dos cosas:

Que la pieza no sea de metal, en cuyo caso nuestro robot, que está ya en una posición de seguridad, queda esperando a que el otro robot termine, para coger otra pieza y empezar de nuevo el proceso. El otro robot recibe nuestra señal OUT[6] y toma la pieza, y, como no ha recibido nuestra OUT[5], se la lleva a su lado de la mesa y la almacena.

Que la pieza sea de metal. Entonces el otro robot toma la pieza y se dirige a la posición de encuentro de los robots. Nuestro robot espera a que el otro esté en posición, y se aproxima (en dos pasos) para recoger la pieza en el aire. La tomamos, y enviamos una señal OUT[10] para avisar de que la hemos cogido. Entonces quedamos a la espera de la señal IN[2] del otro robot, que nos avisa de que ya ha soltado la pieza. Concluido esto, él se aleja y nosotros nos llevamos la pieza a nuestro lado de la mesa, para su almacenamiento.

Este proceso se puede repetir indefinidamente.

## Código

```
$v 1 AUX 0
$v 2 PINZA 1

$v 3 TIMEA 808
$v 4 TIEM 0
$v 5 VAR 808
$p 1 1 0 0 0 0 0 0 0
$p 2 1 POS1 -4771 10675 -10138 -1054 1033 0
$p 3 1 POS2 -4772 13444 -10855 -1057 1032 0
$p 4 1 POS3 -1812 11259 -9388 -1914 185 0
```

```

$P 5 1 POS4 -1814 14012 -9880 -1917 184 0
$P 6 1 ZERO -2366 8038 -7342 -1914 185 0
$P 7 1 POS6 -3021 12703 -13756 -4077 -1974 0
$P 8 1 POS5 -3018 9466 -12527 -4075 -1972 0
$P 9 1 POS8 2937 13141 -11726 -2364 -197 0
$P 10 1 POS7 2937 10281 -10860 -2360 -193 0
$P 11 1 POS10 5234 13284 -11420 -2815 -642 0
$P 12 1 POS9 5233 10460 -10629 -2812 -640 0
$P 13 1 POS12 4273 11940 -12219 -4272 -2481 0
$P 14 1 POS11 3952 9395 -10255 -4445 -2338 0
$P 15 1 AUX1 32767 0 0 0 0 0
$P 16 1 AUX2 4272 8960 -10784 -4270 -2480 0
$P 17 1 AUX 4273 11940 -12219 -4272 -2481 0
$P 18 1 REST -3357 4910 -3611 -1914 185 0
$P 19 1 INT -12116 11018 -5829 -4741 -2688 0
$P 20 1 INT2 -11989 8738 -4796 -4739 -2687 0
$P 21 1 AUX3 -2366 8038 -7342 -1914 185 0
$P 22 1 POS20 -3052 14204 -9395 -1914 185 0
$P 23 1 POS21 -3051 11449 -8979 -1911 187 0
$P 24 1 POS22 1534 15317 -7999 -1914 185 0
$P 25 1 POS23 1533 12274 -8094 -1911 187 0
$P 26 1 A -4037 16705 -10504 -8363 -1846 0
$P 27 1 B -14558 6894 -5642 -3913 -1966 0
$P 28 1 CINTA -14559 8587 -7977 -3916 -1968 0
$P 29 1 APRA -3200 10654 -10214 -2919 1185 0
$P 1 2 00 0 0 0 0 0
$P 2 2 AM -12679 0 -1 -1 -1 0
$pr 1 P3
$pr 2 P4
$pr 3 P5
$pr 4 CERCA
$pr 5 LEJOS
$pr 6 COGER
$pr 7 SOLTA
$pr 8 P21
$pr 9 P22
$pr 10 P23
$pr 11 STOP
$pr 12 RECU
$pr 13 P232
$pr 14 P24
$pr 15 BOTON
$pr 16 P242
$pr 17 P25
$par 1 -400 120 333 0 0 50 12 30
$par 9 0 44 0 0 0 0 0
$par 17 0 0 0 0 200 200 200 250
$par 25 250 180 700 700 700 700 700 0
$par 33 12770 -10216 10216 5022 5022 0 0 0
$par 41 250 250 250 300 300 500 500 500
$par 49 500 500 500 0 13653 -2867 1773 0
$par 57 0 0 0 0 1000 2500 2500 1200

```

```

$par 65 1200 1500 1200 1200 1200 1200 1200 1200 1
$par 73 5400 0 3500 5 0 1 2 0
$par 81 0 0 0 0 0 0 0 0
$par 89 0 0 0 0 160 3490 2210 2210
$par 97 1450 0 0 0 24330 17160 21000 16000
$par 105 16000 32767 32767 32767 32767 32767 32767 0
$par 113 0 0 0 0 0 0 0 0
$par 121 -18830 -400 -32000 -16000 -16000 -32768 -32768 -32768
$par 129 -32768 -32768 -32768 0 0 0 0 0
$par 137 0 0 0 0 3000 3000 3000 3000
$par 145 3000 3000 16000 16000 16000 16000 16000 0
$par 153 0 0 0 0 0 0 0 0
$par 161 4500 4500 4500 4500 4500 4900 4500 4500
$par 169 4500 4500 4500 0 0 0 0 0
$par 177 0 0 0 0 5 5 3 5
$par 185 5 25 10 10 10 10 10 0
$par 193 0 0 0 0 0 0 0 0
$par 201 4500 4000 4000 4000 4000 4000 3000 3000
$par 209 3000 3000 3000 0 0 0 0 0
$par 217 0 0 0 0 4500 6000 6000 3000
$par 225 3000 6000 3000 3000 3000 3000 3000 0
$par 233 0 0 0 0 0 0 0 0
$par 241 24 24 30 24 24 12 5 5
$par 249 5 5 5 0 0 0 0 0
$par 257 0 0 0 0 20 20 20 20
$par 265 20 10 10 10 10 10 10 1
$par 273 0 0 0 0 0 0 0 0
$par 281 4600 4600 4000 4600 4600 4900 4000 4000
$par 289 4000 4000 4000 0 0 8000 50 40
$par 297 50 30 660 15 0 0 0 0
$par 305 0 0 0 0 0 0 0 0

```

```

PROGRAM CERCA
*****

```

```

SPEED 70
SETP AUX2=AUX
SHIFTC AUX2 BY Z 1000
SETP AUX3=AUX2
MOVED AUX2
END

```

```

PROGRAM LEJOS
*****

```

```

SPEED 70
SETP AUX3=ZERO
MOVED ZERO
END

```

```

PROGRAM COGER
*****

```

```
SPEED      25
OPEN
SET        PINZA = 1
SETP       AUX3=AUX
MOVELD     AUX
CLOSE
SET        PINZA = 0
SETP       AUX3=AUX2
MOVELD     AUX2
END
```

```
PROGRAM    SOLTA
*****
```

```
SPEED      25
SETP       AUX3=AUX
MOVELD     AUX
OPEN
SET        PINZA = 1
SETP       AUX3=AUX2
MOVELD     AUX2
END
```

```
PROGRAM    P21
*****
```

```
MOVED      ZERO
SETP       AUX=POS2
GOSUB      CERCA
GOSUB      COGER
GOSUB      LEJOS
SETP       AUX=POS8
GOSUB      CERCA
GOSUB      SOLTA
GOSUB      LEJOS
SETP       AUX=POS4
GOSUB      CERCA
GOSUB      COGER
GOSUB      LEJOS
SETP       AUX=POS10
GOSUB      CERCA
GOSUB      SOLTA
GOSUB      LEJOS
SETP       AUX=POS6
GOSUB      CERCA
GOSUB      COGER
GOSUB      LEJOS
SETP       AUX=POS12
GOSUB      CERCA
GOSUB      SOLTA
GOSUB      LEJOS
```

```
MOVED    ZERO
END
```

```
PROGRAM   P22
*****
```

```
SET       OUT[1] = 1
WAIT      IN[1] = 1
SET       OUT[1] = 0
GOSUB     P21
END
```

```
PROGRAM   P23
*****
```

```
SET       OUT[1] = 1
WAIT      IN[1] = 1
SET       OUT[1] = 0
TRIGGER   STOP BY IN 1 0
GOSUB     P21
END
```

```
PROGRAM   STOP
*****
```

```
STOP      P23
CLRBUF
SPEED     25
MOVED     REST
END
```

```
PROGRAM   RECU
*****
```

```
IF        PINZA = 0
  SUSPEND  P232
  SPEED    25
  MOVED    INT2
  MOVELD   INT
  OPEN
  MOVELD   INT2
  SPEED    70
  MOVED    REST
  WAIT     IN[1] = 1
  MOVED    INT2
  SPEED    25
  MOVELD   INT
  CLOSE
  MOVELD   INT2
  SPEED    70
  MOVED    AUX3
ENDIF
```

```

IF          PINZA = 1
  SUSPEND   P232
  SPEED     25
  MOVED     REST
  WAIT      IN[1] = 1
  MOVED     AUX3
ENDIF
CONTINUE   P232
END

```

```

PROGRAM     P232
*****

```

```

SET         OUT[1] = 1
WAIT        IN[1] = 1
SET         OUT[1] = 0
TRIGGER     RECU BY IN 1 0
GOSUB       P21
END

```

```

PROGRAM     P24
*****

```

```

SET         TIEM = 0
PEND        VAR FROM TIEM
PRINTLN     VAR
IF          VAR > 300
  MOVED     ZERO
  SPEED     70
  MOVED     POS1
  OPEN
  SPEED     25
  MOVED     POS2
  CLOSE
  MOVED     POS4 TIMEA
  MOVED     POS2 TIMEA
  OPEN
  SPEED     25
  MOVED     POS1
  SPEED     70
  MOVED     ZERO
ELSE
  PRINTLN   "Tiempo insuficiente"
  SPEED     70
  MOVED     ZERO
ENDIF
END

```

```

PROGRAM     BOTON
*****

```

```

WAIT        IN[4] = 1

```

```
SET      TIMEA = TIME
WAIT     IN[1] = 0
SET      TIMEA=TIME - TIMEA
END
```

```
PROGRAM   P242
*****
```

```
RUN      P24
WAIT     IN[1] = 1
SET      TIMEA = TIME
WAIT     IN[1] = 0
SET      TIMEA=TIME - TIMEA
POST     TIMEA TO TIEM
END
```

```
PROGRAM   P25
*****
```

```
MOVED    ZERO
SPEED    70
MOVED    APRA
SPEED    25
OPEN
MOVED    A
CLOSE
MOVED    APRA
SPEED    70
MOVED    B
SPEED    25
MOVELD   CINTA
OPEN
MOVELD   B
SET      OUT[7] = 1
SET      ANOUT[7] = 800
DELAY    100
SET      ANOUT[7] = 2000
DELAY    300
SET      ANOUT[7] = 800
DELAY    100
SET      ANOUT[7] = -1000
DELAY    200
SET      ANOUT[7] = -2000
DELAY    400
SET      ANOUT[7] = -1000
DELAY    200
SET      OUT[7] = 0
SPEED    25
MOVELD   CINTA
CLOSE
MOVELD   B
SPEED    70
```

```

MOVED      APRA
SPEED      25
MOVED      A
OPEN
MOVED      APRA
SPEED      70
MOVED      ZERO
END
(END)

```

### Código del otro grupo:

```

$v 1 POSR2 0
$p 1 1 0 0 0 0 0 0 0
$p 2 1 POUT 2021 11719 -9133 -659 655 0
$p 3 1 POS1 -13335 9863 -5942 1114 2278 0
$p 4 1 POSR -11690 6545 363 -2943 -6072 0
$p 5 1 POSH 1967 5490 -5785 -713 41 0
$p 6 1 POUT1 32767 0 0 0 0 0
$p 7 1 POS2 2025 13635 -9464 -668 650 0
$p 8 1 POSR2 -11690 6545 363 -2415 -6960 0
$p 9 1 POUT2 -13330 8977 -4476 1105 2269 0
$p 10 1 POFF -13330 8977 -4476 1105 2269 0
$p 1 2 00 0 0 0 0 0 0
$pr 1 CARGA
$pr 2 DESCA
$pr 3 METAL
$pr 4 PROG_
$pr 5 CMET
$par 1 -400 120 333 0 0 50 12 30
$par 9 0 44 0 0 0 0 0 0
$par 17 0 0 0 0 200 200 200 250
$par 25 250 180 700 700 700 700 700 0
$par 33 12770 -10216 10216 5022 5022 0 0 0
$par 41 250 250 250 300 300 500 500 500
$par 49 500 500 500 0 13653 -2867 1773 0
$par 57 0 0 0 0 1000 2500 2500 1200
$par 65 1200 1500 1200 1200 1200 1200 1200 1
$par 73 5400 0 3500 5 0 1 2 0
$par 81 0 0 0 0 0 0 0 0
$par 89 0 0 0 0 160 3490 2210 2210
$par 97 1450 0 0 0 24330 17160 21000 16000
$par 105 16000 32767 32767 32767 32767 32767 32767 0
$par 113 0 0 0 0 0 0 0 0
$par 121 -18830 -400 -32000 -16000 -16000 -32768 -32768 -32768
$par 129 -32768 -32768 -32768 0 0 0 0 0
$par 137 0 0 0 0 3000 3000 3000 3000
$par 145 3000 3000 16000 16000 16000 16000 16000 0
$par 153 0 0 0 0 0 0 0 0
$par 161 4500 4500 4500 4500 4500 4900 4500 4500
$par 169 4500 4500 4500 0 0 0 0 0
$par 177 0 0 0 0 5 5 3 5

```



```

$par 185 5 25 10 10 10 10 10 0
$par 193 0 0 0 0 0 0 0 0
$par 201 4500 4000 4000 4000 4000 4000 3000 3000
$par 209 3000 3000 3000 0 0 0 0 0
$par 217 0 0 0 0 4500 6000 6000 3000
$par 225 3000 6000 3000 3000 3000 3000 3000 0
$par 233 0 0 0 0 0 0 0 0
$par 241 24 24 30 24 24 12 5 5
$par 249 5 5 5 0 0 0 0 0
$par 257 0 0 0 0 20 20 20 20
$par 265 20 10 10 10 10 10 10 1
$par 273 0 0 0 0 0 0 0 0
$par 281 4600 4600 4000 4600 4600 4900 4000 4000
$par 289 4000 4000 4000 0 0 8000 50 40
$par 297 50 30 660 15 0 0 0 0
$par 305 0 0 0 0 0 0 0 0

```

```

PROGRAM CARGA
*****

```

```

OPEN
SHIFTC POUT BY Z 700
SPEED 50
MOVED POUT
SPEED 20
SHIFTC POUT BY Z -700
MOVED POUT
CLOSE
SHIFTC POUT BY Z 700
MOVED POUT
END

```

```

PROGRAM DESCA
*****

```

```

SET OUT[7] = 0
SHIFTC POUT BY Z 700
SPEED 40
MOVED POUT
SPEED 20
SHIFTC POUT BY Z -700
MOVED POUT
OPEN
SET OUT[11] = 1
SHIFTC POUT BY Z 700
MOVED POUT
END

```

```

PROGRAM METAL
*****

```

```

STOP PROG_

```

```

OPEN
WAIT      IN[2] = 1
SETP      POUT=POS1
GOSUB     CARGA
SPEED     40
MOVED     POSR
SET       OUT[5] = 1
WAIT      IN[4] = 1
OPEN
MOVED     POSR2
SET       OUT[6] = 1
MOVED     POSH
SET       OUT[5] = 0
SET       OUT[6] = 0
GOSUB     PROG_
END

```

```

      PROGRAM    PROG_
      *****

```

```

LABEL     1
SET       OUT[5] = 0
SET       OUT[6] = 0
SET       OUT[11] = 0
MOVED     POSH
TRIGGER   METAL BY IN 1 1
WAIT      IN[2] = 1
SETP      POUT=POS1
GOSUB     CARGA
SETP      POUT=POS2
GOSUB     DESCA
GOTO      1
END

```

```

      PROGRAM    CMET
      *****

```

```

OPEN
SPEED     60
MOVED     POUT2
SPEED     30
MOVED     POUT
CLOSE
MOVED     POUT2
END
(END)

```