

Sesión Laboratorio de Robótica – Scorbob (II)

Durante esta segunda manejamos variables locales y globales, subprogramas, entradas/salidas (sensores, señales) y aprendemos estructuras de control de flujo, y a controlar la cinta transportadora.

Ejercicio 1: subprogramas, SHIFTC

Programas: P21, COGER, SOLTA, CERCA, LEJOS

El objetivo es generar unos subprogramas a los que se pueda pasar “como parámetro” una posición, y realicen las operaciones de aproximación, alejamiento, y coger y soltar objetos con la pinza. Esto lo hemos realizado definiendo 3 variables, AUX, AUX2 y AUX3, que usamos para pasar la posición del programa principal (P21) a las subrutinas CERCA, LEJOS, COGER, SOLTA (usando SETP). El salto a las subrutinas lo hacemos con GOSUB. Hay también una variable PINZA definida, que usamos en apartados posteriores.

Ejercicio 2

Programas: P22, P21, COGER, SOLTA, CERCA, LEJOS

El segundo apartado es una ampliación del primero: realiza lo mismo, pero antes de empezar espera a que se active un interruptor para empezar a construir. Antes de empezar a construir, la salida 1 (OUTPUT[1]) está activa, y se desactiva al comenzar a construir.

En el programa principal, P22, encendemos la salida 1, esperamos al interruptor, apagamos la salida 1, y saltamos al programa P21.

Ejercicio 3

Programas: P23, P232, STOP, RECU, P21...

Ampliamos el programa anterior: al activarse la entrada 1 se lanza el programa STOP mediante el comando TRIGGER. STOP para (mediante la instrucción STOP) el programa P23, que es el principal, limpia el buffer de instrucciones y mueve el robot a una posición de reposo con velocidad reducida.

En la segunda parte, cuyo programa principal es P232, modificamos el programa anterior para que, al cambiar el interruptor, se suspende P232 (SUSPEND) dentro del programa RECU y, si tenemos la pieza, la dejamos en una posición intermedia (INT e INT2), reanudando el movimiento por donde iba al reactivar el interruptor (coge la pieza de nuevo y sigue con la rutina). Aquí no limpiamos el buffer, porque lo necesitamos (CONTINUE).

Ejercicio 4

Programas: P24, P242

En este apartado se trata de sincronizar dos programas mediante los comandos PEND y POST, y usar la variable TIME para realizar movimientos temporizados. Primero ejecutamos P242, que empieza por lanzar el programa P24, y luego se queda esperando al interruptor. Entretanto, P24 inicializa TIEM a 0 y se queda colgando de P242 hasta que en TIEM haya un valor distinto de

0 (cosa que sucede cuando el interruptor es activado y desactivado: el tiempo entre uno y otro evento es el que se guarda en la variable TIEM).

*El programa BOTON no se llega a usar.

Ejercicio 5

Programas: P25

En este apartado, usamos la cinta transportadora. Para ello primero se activa la salida (SET OUT[7] = 1) y luego se manda orden con la velocidad a la que deseamos que se desplace (SET ANOUT[7] = ...). Para hacerlo mejor, la velocidad se ajusta de manera que el primer segundo se desplaza más lentamente, 3 segundos intermedios a velocidad normal, y otro segundo antes de parar, lentamente de nuevo. Y se procede de igual forma a la inversa, para devolver la pieza a su posición original.

Hemos observado que para velocidades negativas teníamos que poner esperas más largas para que recorriese la misma distancia.

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 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 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 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)
```