UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TRABAJO FIN DE GRADO

Grado en Ingeniería Industrial y Automática.

CONTROL Y GUIADO DE UN ROBOT MÓVIL.

Volumen III

Anexo

Autor: Raúl Cruzado Muñoz
Director: Juan Gámiz

Convocatoria: Mayo 2017

Índice Anexo

A. Código de programa Robot SR1.

```
1   Attribute VB_Name = "Explorer"
2   '*********************************************************************************
3   '* EXPLORER *
4   '* Programa para controlar el robot SR1 con cámara completamente de forma remota *
5   '* mediante órdenes recibidas por el puerto serie. Se controla tanto en los movimientos
    del *
6   '* robot, como los de la cámara. Además se ha implementado la función de sonar de barrido
    *
7   '* aprovechando que el sensor de distancias está montado sobre la torreta de la cámara.
    *
8   '* También se han implementado movimientos y giros basados en grados reales y la *
9   '* utilización de memorias de las posiciones de la cámara. La comunicación serie se
    realiza *
10  '* entre el robot en un lado y con un cable serie que está conectado a un PC que ejecuta
    el *
11  '* programa SR1EXPLORER en el otro. *
12  '*********************************************************************************
13  'Asignación de pines en el SR1
14  Const Txd As Byte = 1 'Pin Transmisión de datos Puerto Serie 1
15  Const Rxd As Byte = 2 'Pin recepción de datos Puerto Serie 1
16  Const Scl As Byte = 5 'Pin del clock del bus I2C SCL
17  Const Sda As Byte = 6 'Pin de datos del bus I2C SDA
18  Const Evc As Byte = 7 'Pin para conectar la alimentación auxiliar
19  Const Buz As Byte = 8 'Pin usado por el Beep
20  Const Led As Byte = 9 'Pin para activar los leds
21  Const IR1 As Byte = 10 'Pin de la entrada IR1
22  Const IR2 As Byte = 11 'Pin de la entrada IR2
23  Const IR3 As Byte = 12 'Pin de la entrada IR3
24  Const Se1 As Byte = 13 'Pin del Servo 1
25  Const Se2 As Byte = 14 'Pin del Servo 2
26  Const Se3 As Byte = 15 'Pin del Servo 3
27  Const Se4 As Byte = 16 'Pin del Servo 4
28  Const Inc As Byte = 17 'Pin usado por el sensor de inclinación
29  Const Bur As Byte = 18 'Pin usado por el Bumper Derecho
30  Const Bul As Byte = 19 'Pin usado por el Bumper Izquierdo
31  Const Tem As Byte = 20 'Pin usado por sensor de temperatura
32  Const LRo As Byte = 25 'Pin del led rojo del BasicX24
33  Const LVe As Byte = 26 'Pin del led verde del BasicX24
34  'Constantes y variables generales
35  Const ServoAce1 As Integer = 40 'Valor de aceleración servo 1
36  Const ServoAce2 As Integer = 40 'Valor de aceleración servo 2
37  Const ServoMin As Integer = 800 'Valor Mínimo de los servos 1 y 2
38  Const ServoMax As Integer = 2000 'Valor Máximo de los servos 1 y 2
39  Const Lon As Byte = 1 'Valor de pin para Led encendido
40  Const Lof As Byte = 0 'Valor de pin para Led apagado
41  Const SrfReg As Byte = 0 'Registro de comandos del SRF08
42  Const LuzReg As Byte = 1 'Registro del sensor de luz del SRF08
43  Const RangoReg As Byte = 2 'Registro de las mediciones del SRF08
44  Const RangoCmd As Byte = 81 'Registro de comandos, 81 centímetros
45  Const Sonar1 As Byte = &HE0 'Dirección del sensor SRF08
46  Const Brujula1 As Byte = &HC0 'Dirección del sensor brújula
47  Const RumboReg As Byte = 2 'La dirección en registro 2 y 3(High:Low)
48  Dim Dist As New UnsignedInteger 'Distancia medida por el sensor SRF08
49  Dim Lux As Byte 'Valor de la luminosidad del sensor SRF08
50  Dim Rumbo As New UnsignedInteger 'Valor de rumbo de la brújula digital.
51  Dim DirACT As Integer 'Valor de brújula en grados exactos
52  Dim Temperatura As Single 'Valor de la temperatura x 2 en grados
53  Dim I2cAck As Boolean 'Indicador de Ok bus I2C
```

```
54  Dim Se1Stop As Integer 'Valor de parada Servo1
55  Dim Se2Stop As Integer 'Valor de parada Servo2
56  Dim Se3Stop As Integer 'Valor de parada Servo3
57  Dim Se4Stop As Integer 'Valor de parada Servo4
58  Dim Se1Value As Integer 'Valor del Servo Servo1
59  Dim Se2Value As Integer 'Valor del Servo Servo2
60  Dim Se3Value As Integer 'Valor del Servo Servo3
61  Dim Se4Value As Integer 'Valor del Servo Servo4
62  Dim Se3Min As Integer 'Valor Mínimo del Servo3
63  Dim Se3Max As Integer 'Valor Máximo del servo3
64  Dim Se4Min As Integer 'Valor Mínimo del Servo4
65  Dim Se4Med As Integer 'Valor Medio del servo4
66  Dim Se4Max As Integer 'Valor Máximo del servo4
67  Dim Escalon As Integer 'Incremento del movimiento horizontal
68  Dim InBuff(1 To 20) As Byte 'Búfer de entrada del puerto serie
69  Dim OutBuff(1 To 20) As Byte 'Búfer de salida del puerto serie
70  Dim Se3Pos As Byte 'Guarda la altura del scan
71  Dim Se4Pos As Byte 'Guarda la posición del servo 4
72  Dim CamM3(1 To 6) As Byte 'Memoria para las posiciones de la cámaras
73  Dim CamM4(1 To 6) As Byte 'Memoria para las posiciones de la cámaras
74  Dim Acel1 As Integer 'Valor de aceleración servo1 en los giros
75  Dim Acel2 As Integer 'Valor de aceleración servo2 en los giros
76  Dim Progra(1 To 3, 1 To 40) As Byte 'Matriz que contiene los comandos del programa
77  Dim Programa As Byte 'Numero del programa
78  Dim Dato As Byte 'Valor de cada elemento
79  Dim Comando As New BoundedString_12 'Contiene el comando recibido
80  Dim Orden(1 To 10) As Byte 'Contiene los caracteres del comando
81  Dim Conta As Integer 'Numero de bytes recibidos en el puerto serie
82
83  Dim Tx As String
84  Dim Valrs As Single
85  Dim Valr As Integer
86  Dim Success As Boolean
87
88  'Programa principal
89  Public Sub Main()
90  Dim SubComando As String * 3 'Contiene la subfamilia del comando
91  Dim Memo As Byte 'Numero de la memoria
92  Call OpenQueue(InBuff, 20) 'Prepara el búfer de recepción del P. serie
93  Call OpenQueue(OutBuff, 20) 'Prepara el búfer de transmisión del P. serie
94  Call OpenCom(1, 19200, InBuff, OutBuff) 'Abre el puerto Com 1
95  Call ServoStop 'Recupera los valores de parada de los servos
96  Call CamStop 'Recupera los valores de trabajo de la cámara
97  Programa = 1 'Numero de programa por defecto
98  Escalon = (Se4Max - Se4Min) \ 90 'Son 30 pasos en 180 grados
99  Se1Value = Se1Stop 'Establece el valor inicial de los servos
100 Se2Value = Se2Stop 'como parado
101 Se3Value = Se3Min 'Pone la cámara en modo reposo
102 Se4Value = Se4Med 'Y la centra en su posición
103 Call PutPin(Evc, 0) 'Apaga la cámara
104 Call PutPin(Led, 0) 'Apaga los leds
105 Call CReposo 'Mueve la cámara a su sitio de reposo
106 Call Beep_Beep 'Hace el Beep de Inicio
107 Do
108 Conta = GetQueueCount(InBuff) 'Comprueba el numero de caracteres recibidos
109 If Conta > 4 Then 'Si hay más de 4 debe de haber un comando
110 Call GetQueue(InBuff, Orden, 8) 'Lee 8 caracteres del puerto serie
111 'Asigna a Comando los 5 caracteres leídos
112 Comando = Chr(Orden(1)) & Chr(Orden(2)) & Chr(Orden(3)) & Chr(Orden(4)) & Chr(Orden(5))
113
114 'Asigna a Tx los 3 último caracteres leidos
115 Tx = Chr(Orden(6)) & Chr(Orden(7)) & Chr(Orden(8))
116
```

```
117 'Convierte los strings en numero Integer
118 Call ValueS(Tx, Valrs, Success)
119 Valr=CInt(Valrs)
120
121 'Asigna a SubComando solo los 3 primeros para identificar la familia del comando
122 SubComando = Chr(Orden(1)) & Chr(Orden(2)) & Chr(Orden(3))
123 If SubComando = "MOR" Then 'Comandos de movimiento del robot
124 If Comando = "MORAE" Then
125 Call Adelante 'Robot adelante
126 End If
127 If Comando = "MORAT" Then
128 Call Atras 'Robot atrás
129 End If
130 If Comando = "MORIZ" Then
131 Call GIzquierda(90, 0) 'Giro a la izquierda 90 grados
132 End If
133 If Comando = "MORDE" Then
134 Call GDerecha(90, 0) 'Giro a la derecha 90 grados
135 End If
136 If Comando = "MORIA" Then
137 Call GIzquierda(45, 0) 'Giro a la izquierda 45 grado
138 End If
139 If Comando = "MORDA" Then
140 Call GDerecha(45, 0) 'Giro a la derecha 45 grados
141 End If
142 If Comando = "MORAA" Then
143 Call MAdelante 'Avance rápido
144 End If
145 If Comando = "MORAI" Then
146 Call GIzquierda(5, 0) 'Pequeño giro a la izquierda
147 End If
148 If Comando = "MORAD" Then
149 Call GDerecha(5, 0) 'Pequeño giro a la derecha
150 End If
151
152 IF Comando ="MORRF" Then         'Se mueve hacia delante la distancia indicada
153 Call Adelantepar (Valr,0)
154 End If
155
156 IF Comando ="MORRR" Then         'Se mueve hacia atras la distancia indicada
157 Call Atraspar (Valr,0)
158 End If
159
160 If Comando= "MORRD" Then         'Gira  hacia  la  derecha  los  grados  indicados
    (servomotores)
161 Call GDerecha (Valr,0)
162 End If
163
164 If Comando= "MORRI" Then         'Gira  hacia  la  izquierda  los  grados  indicados
    (servomotores)
165 Call GIzquierda (Valr,0)
166 End If
167
168
169 If Comando= "MORRU" Then         'Gira al valor de grados indicado con brujula
170 Call GRum (Valr,0)
171 End If
172
173 If Comando = "MORNO" Then
174 Call Norte 'Gira hacia el norte
175 End If
176
177
```

```
178 If Comando = "MORCA" Then
179 Call Camara 'Gira a la posición de la cámara
180 End If
181 End If
182 If SubComando = "MOC" Then 'Comandos de movimiento de cámara
183 If Comando = "MOCDE" Then
184 Call CDerecha 'Mueve la cámara a la derecha
185 End If
186 If Comando = "MOCDD" Then
187 Call CDerechaD 'Mueve la cámara a la derecha rápido
188 End If
189 If Comando = "MOCCE" Then
190 Call CCentro 'Mueve la cámara al centro
191 End If
192 If Comando = "MOCIZ" Then
193 Call CIzquierda 'Mueve la cámara a la izquierda
194 End If
195 If Comando = "MOCII" Then
196 Call CIzquierdaI 'Mueve la cámara a la izquierda rápido
197 End If
198 If Comando = "MOCAR" Then
199 Call CArriba 'Mueve la cámara arriba
200 End If
201 If Comando = "MOCAB" Then
202 Call CAbajo 'Mueve la cámara abajo
203 End If
204 End If
205 If SubComando = "REC" Then 'Comandos de memorizar posición de cámara
206 Memo = Orden(5) - 48 'Calcula el numero de memoria
207 Call MemoCam(Memo) 'Guarda la memoria de la cámara
208 End If
209 If SubComando = "CAL" Then 'Comandos de recuperar posición de cámara
210 Memo = Orden(5) - 48 'Calcula el numero de memoria
211 Call MoveCam(CamM4(Memo), CamM3(Memo)) 'Mueve la cámara
212 End If
213 If SubComando = "TEL" Then 'Comandos de telemetría
214 If Comando = "TELDI" Then
215 Call Sonar 'Comprueba la distancia y la envía
216 End If
217 If Comando = "TELLU" Then
218 Call Luminosidad 'Comprueba la luz y la envía
219 End If
220 If Comando = "TELTE" Then
221 Call Tempera 'Comprueba la temperatura y la envía
222 End If
223 If Comando = "TELRU" Then
224 Call Brujula 'Comprueba el rumbo y lo envía
225 End If
226 If Comando = "TELIN" Then
227 Call Inclinacion 'Comprueba la inclinación
228 End If
229 If Comando = "TELIZ" Then
230 Call BIzquierda 'Comprueba parachoques izquierdo
231 End If
232 If Comando = "TELDE" Then
233 Call BDerecha 'Comprueba parachoques derecho
234 End If
235
236 If comando ="TELTE" Then 'Comprueba parachoques y sensor inclinación
237 Call Check
238 End If
239
240 End If
```

```
241 'If SubComando = "TXP" Then 'Comandos de recibir programa
242 'Programa = Orden(5) - 48 'Calcula el numero de programa
243 'Call RXProgra 'Recibe los datos del programa
244 'End If
245 'If SubComando = "EJP" Then 'Comandos de ejecutar programa
246 'Programa = Orden(5) - 48 'Calcula el numero de programa
247 'Call EJProgra 'Ejecuta el programa
248 'End If
249 'Comandos varios
250 If Comando = "CAMON" Then
251 Call PutPin(Evc, 1) 'Enciende la cámara
252 End If
253 If Comando = "CAMOF" Then
254 Call PutPin(Evc, 0) 'Apaga la cámara
255 End If
256 If Comando = "LEDON" Then
257 Call PutPin(Led, 1) 'Enciende los leds
258 End If
259 If Comando = "LEDOF" Then
260 Call PutPin(Led, 0) 'Apaga los leds
261 End If
262 If Comando = "ALARM" Then
263 Call Beep_Beep 'Suena beep-beep con leds
264 End If
265 If Comando = "BIPON" Then
266 Call Beep 'Hace un beep
267 End If
268 If Comando = "SCAN0" Then
269 Se3Pos = 0 'Posición baja
270 Call Scan 'hace un barrido del sonar
271 End If
272 If Comando = "SCAN1" Then
273 Se3Pos = 16 'Posición media
274 Call Scan 'hace un barrido del sonar
275 End If
276 If Comando = "SCAN2" Then
277 Se3Pos = 32 'Posición alta
278 Call Scan 'hace un barrido del sonar
279 End If
280 Call ClearQueue(InBuff)
281 End If
282 Loop
283 End Sub
284 'Subrutina que ejecuta los comandos del programa actual
285 Sub EJProgra()
286 Dim Valor As Byte 'Índice de los comandos leídos
287 Valor = 1 'Comienza por el primero
288 Do Until Valor > 40 'Repite la secuencia hasta el final del programa
289 Dato = Progra(Programa, Valor) 'Lee un dato del programa
290 Valor = Valor + 1 'Aumenta el puntero del programa
291 If Dato < 65 Then 'Si es menor de 65 ha terminado el programa
292 Exit Do 'y sale
293 End If
294 Call Procesa(Dato) 'Llama a la rutina que ejecuta las órdenes
295 Call Sleep(200) 'Pausa entre instrucciones para evitar reinicios
296 Loop
297 End Sub
298 'Subrutina que ejecuta el comando pasado
299 Sub Procesa(ByVal Coman As Byte)
300 Select Case Coman
301 Case 65
302 Call Adelante 'Mueve el robot hacia adelante
303 Case 66
```

```
304 Call MAdelante 'Mueve el robot hacia adelante rápido
305 Case 67
306 Call Atras 'Mueve el robot hacia atrás
307 Case 68
308 Call GIzquierda(5, 0) 'Pequeño giro a la izquierda
309 Case 69
310 Call GIzquierda(45, 0) 'Giro a la izquierda 45 grado
311 Case 70
312 Call GIzquierda(90, 0) 'Giro a la izquierda 90 grados
313 Case 71
314 Call GDerecha(5, 0) 'Pequeño giro a la derecha
315 Case 72
316 Call GDerecha(45, 0) 'Giro a la derecha 45 grados
317 Case 73
318 Call GDerecha(90, 0) 'Giro a la derecha 90 grados
319 Case 74
320 Call Camara 'Gira hacia la cámara
321 Case 75
322 Call Norte 'Gira al norte
323 Case 76
324 Call CArriba 'Mueve la cámara arriba
325 Case 77
326 Call CAbajo 'Mueve la cámara abajo
327 Case 78
328 Call CCentro 'Mueve la cámara al centro
329 Case 79
330 Call CDerecha 'Mueve la cámara a la derecha
331 Case 80
332 Call CDerechaD 'Mueve la cámara a la derecha rápido
333 Case 81
334 Call CIzquierda 'Mueve la cámara a la izquierda
335 Case 82
336 Call CIzquierdaI 'Mueve la cámara a la izquierda rápido
337 Case 83
338 Call PutPin(Evc, 1) 'Enciende la cámara
339 Case 84
340 Call PutPin(Evc, 0) 'Apaga la cámara
341 Case 85
342 Call PutPin(Led, 1) 'Enciende los leds
343 Case 86
344 Call PutPin(Led, 0) 'Apaga los leds
345 Case 87
346 Call Beep 'Hace un beep
347 Case 88
348 Call Beep_Beep 'Suena beep-beep con leds
349 End Select
350 End Sub
351 'Subrutina que recibe los 40 caracteres del programa y los guarda en la variable,
    recibiéndolos de 10 en 10
352 'Sub RXProgra()
353 'Dim Valor As Byte 'Contador
354 'Valor = 1
355 'Dim Repite As Byte 'Contador
356 'Call ClearQueue(InBuff) 'Limpia el búfer inicialmente
357 'For Repite = 1 To 4 'Se reciben 10 caracteres 4 veces
358 'Conta = 0
359 'Do Until Conta > 9
360 'Conta = GetQueueCount(InBuff) 'Comprueba el numero de caracteres recibidos
361 'Loop
362 'Call GetQueue(InBuff, Orden, 10) 'Lee 10 caracteres del puerto serie
363 'For Dato = 1 To 10 'Los guarda en la memoria
364 'Progra(Programa, Valor) = Orden(Dato)
365 'Valor = Valor + 1 'Incrementa el puntero de memoria
```

```
366 'Next
367 'Call Beep
368 'Next
369 'Call Beep 'Indica que ha recibido el programa
370 'End Sub
371 'Subrutina que hace el barrido de mediciones y lo envía por el puerto serie. Se hacen 32
    mediciones entre el límite máximo y el mínimo del servo 4
372 Sub Scan()
373 Dim N As Byte 'Contador auxiliar
374 Dim DLuz As String * 3 'Valor de la luminosidad como cadena
375 Dim DRango As String * 3 'Valor de la distancia como cadena
376 Dim Se3Ant As Integer 'Valor del servo 3 antes del scan
377 Dim Se4Ant As Integer 'Valor del servo 4 antes del scan
378 Se3Ant = Se3Value 'Los guarda para recuperarlos
379 Se4Ant = Se4Value 'después del escáner
380 Se4Pos = 0 'Primer valor es el centro
381 Call MoveCam(Se4Pos, Se3Pos) 'Mueve la cámara al inicio
382 Se4Value = Se4Min 'Primer valor es el valor mínimo
383 For N = 1 To 30 'Son 30 medidas
384 Call I2cByteWrite(Sonar1, SrfReg, RangoCmd) 'Comando de hacer la medición en Cm
385 Call CPulsos 'Mueve el sensor horizontalmente
386 Se4Value = Se4Value + Escalon 'Mientras se hace la medición que
387 Call CPulsos 'tarda unos 65 ms se mueve el servo
388 Se4Value = Se4Value + Escalon 'tres veces
389 Call CPulsos 'Ahora la medida ya esta lista
390 Se4Value = Se4Value + Escalon 'Sigue con el movimiento
391 Call Sleep(5) 'Una pausa para que termine la medición
392 Lux = I2CByteRead(Sonar1, LuzReg) 'Lee el sensor de luz
393 Dist = I2CWordRead(Sonar1, RangoReg) 'Lee la distancia medida
394 If Dist > 999 Then 'Limita el valor a 3 caracteres
395 Dist = 999 'para evitar un error en la
396 End If 'recepción
397 DLuz = CStr(Lux) 'Convierte el valor de la luz en
398 If Len(DLuz) < 3 Then 'una cadena de 3 caracteres de
399 DLuz = "0" & DLuz 'longitud fija
400 End If
401 If Len(DLuz) < 3 Then 'Convierte el valor de la luz en
402 DLuz = "0" & DLuz 'una cadena de 3 caracteres de
403 End If 'longitud fija
404 DRango = CStr(Dist) 'Convierte el valor de la distancia
405 If Len(DRango) < 3 Then 'en una cadena de 3 caracteres de
406 DRango = "0" & DRango 'longitud fija, para enviarlos
407 End If 'por el puerto serie sin problemas
408 If Len(DRango) < 3 Then 'en la recepción
409 DRango = "0" & DRango
410 End If
411 Call PutQueueStr(OutBuff, DLuz & DRango) 'Manda los 6 bytes por el p.serie
412 Next
413 'Manda los 2 últimos bytes con un retardo para evitar problemas con el radio módem
414 Call Sleep(100)
415 Call PutQueueStr(OutBuff, Chr(10) & Chr(13))
416 Call MoveCam(16, Se3Pos) 'Mueve la cámara a la posición central
417 End Sub
418 'Subrutina que mueve la cámara hasta la posición pasada en XCam y YCam de forma suave
    desde la posición actual que es Se3Value y Se4Value
419 Sub MoveCam(ByVal XCam As Byte, ByVal YCam As Byte)
420 Dim EscX As Integer
421 Dim EscY As Integer
422 Dim PosFinalX As Integer
423 Dim PosFinalY As Integer
424 Dim N As Byte
425 EscX = (Se4Max - Se4Min) \ 32 'Calcula el escalón relativo a 32
426 EscY = (1000 - 490) \ 32 'para cada servo
```

```
427 PosFinalX = Se4Min + EscX * CInt(XCam) 'Calcula las posiciones finales
428 PosFinalY = 1000 - EscY * CInt(YCam) 'en función de los valores pasados
429 'Si es el valor central mueve la cámara hasta el centro exacto, no el teórico
430 If XCam = 16 Then
431 PosFinalX = Se4Med
432 End If
433 If PosFinalX < Se4Value Then 'Si destino mayor que actual, resta
434 EscX = -EscX
435 End If
436 If PosFinalY < Se3Value Then
437 EscY = -EscY
438 End If
439 For N = 1 To 32
440 Se3Value = Se3Value + EscY
441 Se4Value = Se4Value + EscX
442 If EscY > 0 Then 'Si el desplazamiento es positivo
443 If Se3Value > PosFinalY Then
444 Se3Value = PosFinalY
445 End If
446 Else
447 If Se3Value < PosFinalY Then
448 Se3Value = PosFinalY
449 End If
450 End If
451 If EscX > 0 Then 'Si el desplazamiento es positivo
452 If Se4Value > PosFinalX Then
453 Se4Value = PosFinalX
454 End If
455 Else
456 If Se4Value < PosFinalX Then
457 Se4Value = PosFinalX
458 End If
459 End If
460 Call CPulsos
461 Next
462 End Sub
463 'Subrutina que mueve la cámara a la izquierda
464 Sub CIzquierda()
465 Dim N As Integer
466 For N = 1 To 8
467 Se4Value = Se4Value - 4 'Incrementa el valor del servo
468 Call CPulsos 'Manda los pulsos a la cámara
469 Next
470 End Sub
471 'Subrutina que mueve la cámara a la izquierda rápido
472 Sub CIzquierdaI()
473 Dim N As Integer
474 For N = 1 To 20
475 Se4Value = Se4Value - 8 'Incrementa el valor del servo
476 Call CPulsos 'Manda los pulsos a la cámara
477 Next
478 End Sub
479 'Subrutina que mueve la cámara ala derecha
480 Sub CDerecha()
481 Dim N As Integer
482 For N = 1 To 8
483 Se4Value = Se4Value + 4 'Decrementa el valor del servo
484 Call CPulsos 'Manda los pulsos a la cámara
485 Next
486 End Sub
487 'Subrutina que mueve la cámara ala derecha rápido
488 Sub CDerechaD()
489 Dim N As Integer
```

```
490 For N = 1 To 20
491 Se4Value = Se4Value + 8 'Decrementa el valor del servo
492 Call CPulsos 'Manda los pulsos a la cámara
493 Next
494 End Sub
495 'Subrutina que mueve la cámara al centro
496 Sub CCentro()
497 Dim N As Integer
498 For N = 1 To 20
499 Se4Value = Se4Med 'Establece el valor del servo
500 Call CPulsos 'Manda los pulsos a la cámara
501 Next
502 End Sub
503 'Subrutina que mueve la cámara arriba
504 Sub CArriba()
505 Dim N As Integer
506 For N = 1 To 8
507 Se3Value = Se3Value -8 'Incrementa el valor del servo
508 Call CPulsos 'Manda los pulsos a la cámara
509 Next
510 End Sub
511 'Subrutina que mueve la cámara abajo
512 Sub CAbajo()
513 Dim N As Integer
514 For N = 1 To 8
515 Se3Value = Se3Value +8 'Decrementa el valor del servo
516 Call CPulsos 'Manda los pulsos a la cámara
517 Next
518 End Sub
519 'Subrutina que sitúa la cámara en la posición de reposo
520 Sub CReposo()
521 Dim N As Integer
522 For N = 1 To 40
523 Se4Value = Se4Med 'Valor de reposo del servo 4
524 Se3Value = 1000 'Valor de reposo de servo 3
525 Call CPulsos 'Manda los pulsos a la cámara
526 Next
527 End Sub
528 'Subrutina que guarda las coordenadas de la posición de la cámara como un par de
    coordenadas
529 'cuyo valor está entre 1 y 32
530 Sub MemoCam(ByVal Mpos As Byte)
531 Dim EscX As Integer
532 Dim EscY As Integer
533 Dim PosX As Integer
534 Dim PosY As Integer
535 EscX = (Se4Max - Se4Min) \ 32 'Calcula el escalón relativo a 32
536 EscY = (Se3Max - Se3Min) \ 32 'para cada servo
537 PosX = (Se4Value - Se4Min) \ EscX 'Calcula el valor relativo a 32 posiciones
538 PosY = (Se3Value - Se3Min) \ EscY 'para guardarlos en las memoria
539 CamM3(Mpos) = CByte(PosY) 'Guarda las coordenadas en la posición
540 CamM4(Mpos) = CByte(PosX) 'de memoria correspondiente.
541 End Sub
542 'Subrutina que manda los pulsos al servo horizontal y vertical de la cámara
543 Sub CPulsos()
544 If Se4Value > Se4Max Then 'Comprueba si los valores
545 Se4Value = Se4Max 'están fuera de los limites
546 End If 'máximos y mínimos y si
547 If Se4Value < Se4Min Then 'es así limita el valor para
548 Se4Value = Se4Min 'no sobrepasar los valores y
549 End If 'evitar dañar los servos
550 If Se3Value > 1000 Then 'de la cámara
551 Se3Value = 1000
```

```
552 End If
553 If Se3Value < 490 Then
554 Se3Value = 490
555 End If
556 Call PulseOut(Se4, Se4Value, 1) 'Pulso servo horizontal
557 Call PulseOut(Se3, Se3Value, 1) 'Pulso servo vertical
558 Call Sleep(10) 'Retardo entre pulsos 10 x 1,95 ms
559 End Sub
560 'Subrutina que manda los pulsos a los servos
561 Sub Pulsos()
562 If Se1Value > ServoMax Then 'Verifica que los valores de los
563 Se1Value = ServoMax 'servos están dentro de los limites
564 End If 'máximo y mínimo y si los sobrepasa
565 If Se2Value > ServoMax Then 'se limitan a estos.
566 Se2Value = ServoMax
567 End If
568 If Se1Value < ServoMin Then 'Hace lo mismo con el limite
569 Se1Value = ServoMin 'inferior de ambos servos
570 End If
571 If Se2Value < ServoMin Then
572 Se2Value = ServoMin
573 End If
574 Call PulseOut(Se1, Se1Value, 1) 'Pulso servo izquierdo
575 Call PulseOut(Se2, Se2Value, 1) 'Pulso servo derecho
576 Call Sleep(10) 'Retardo entre pulsos 10 x 1,95 ms
577 End Sub
578 'Subrutina que hace que el robot gire hacia la posición de la cámara
579 Sub Camara()
580 Dim Giro As Integer
581 Dim Paso As Integer
582 Paso = (Se4Max - Se4Min) \ 180
583 If Se4Value = Se4Med Then 'Si está en posición sale
584 Exit Sub
585 End If
586 If Se4Value > Se4Med Then 'Giro hacia la izquierda
587 Giro = (Se4Value - Se4Med) \ Paso 'Calcula el ángulo
588 Paso = Paso * 3
589 Call Izquierda(Giro, Paso) 'Hace el giro
590 Else
591 Giro = (Se4Med - Se4Value) \ Paso 'Giro a la derecha
592 Paso = Paso * 3
593 Call Derecha(Giro, Paso) 'Hace el giro
594 End If
595 Call CCentro 'Aseguramos que la cámara ha vuelto a su posición central
596 End Sub
597 'Subrutina que hace que el robot gire hacia el norte. Hace el giro hacia el norte por el
    lado más corto
598 Sub Norte()
599 Acel1 = 80 'Establece un valor de aceleración
600 Acel2 = 80 'rápido para los dos servos
601 Call Dir 'Comprueba el rumbo actual
602 If DirACT < 180 Then 'Si es más pequeño de 180 gira a la derecha
603 Do
604 Call Dir 'Comprueba la dirección
605 If DirACT > 345 Then 'Cuando esta próximo al
606 Acel1 = 40 'Norte frena reduciendo la
607 Acel2 = 40 'aceleración
608 End If
609 If DirACT < 10 Then 'Evita que se pase el 359
610 Exit Do
611 End If
612 Se1Value = Se1Stop + Acel1 'Frena la rueda derecha y
613 Se2Value = Se2Stop + Acel2 'acelera la rueda izquierda
```

```
614 Call Pulsos 'mientras gira
615 Loop
616 Else 'Si es menos de 180
617 Do 'Giro por la izquierda
618 Call Dir
619 If DirACT < 15 Then 'Cuando esta próximo al
620 Acel1 = 40 'Norte frena reduciendo la
621 Acel2 = 40 'aceleración
622 End If
623 If DirACT > 350 Then 'Evita que se pase el 0
624 Exit Do 'saliendo del bucle
625 End If
626 Se1Value = Se1Stop - Acel1 'Adelante la rueda derecha
627 Se2Value = Se2Stop - Acel2 'atrás la rueda izquierda
628 Call Pulsos 'mientras gira
629 Loop
630 End If
631 End Sub
632 'Subrutina que hace que el robot realice un giro hacia la izquierda con el valor pasado
    en grados aproximadamente (no brújula)
633 Sub GDerecha(ByVal GiroI As Integer, ByVal Salto As Integer)
634 Dim Grados As Integer
635 Dim Veces As Integer
636 Acel1 = 85 'Estos valoren varían dependiendo
637 Acel2 = 85 'del valor de parada de cada servo
638 Grados = GiroI
639 For Veces = 1 To Grados \ 2
640 Se1Value = Se1Stop - Acel1 'Frena la rueda derecha
641 Se2Value = Se2Stop - Acel2 'Acelera la rueda izquierda
642 Call Pulsos 'Manda los pulsos a los servos
643 Next
644 End Sub
645 'Subrutina que hace que el robot realice un giro hacia la derecha con el valor pasado en
    grados aproximadamente (no brújula)
646 Sub GIzquierda(ByVal GiroD As Integer, ByVal Salto As Integer)
647 Dim Grados As Integer
648 Dim Veces As Integer
649 Acel1 = 78 'Estos valoren varían dependiendo
650 Acel2 = 78 'del valor de parada de cada servo
651 Grados = GiroD
652 For Veces = 1 To Grados \ 2
653 Se1Value = Se1Stop + Acel1 'Frena la rueda derecha
654 Se2Value = Se2Stop + Acel2 'Acelera la rueda izquierda
655 Call Pulsos 'Manda los pulsos a los servos
656 Next
657 End Sub
658
659 'Subrutina que hace que el robot realice un giro hacia la izquierda con el valor pasado
    en grados
660 Sub Izquierda(ByVal GiroI As Integer, ByVal Salto As Integer)
661 Dim Destino As Integer
662 Dim Ofset As Integer 'Sirve como valor diferencial
663 Dim NuevaPos As Integer 'Posición que se alcanza cada vez
664 Acel1 = 82
665 Acel2 = 82
666 Ofset = 0
667 Call Dir 'Comprueba el rumbo actual
668 If DirACT <= GiroI Then 'Si el rumbo actual es menor que el
669 DirACT = DirACT + 360 'giro que hay que dar, se suma una
670 Ofset = 360 'vuelta completa para hacer los cálculos
671 End If
672 Destino = DirACT - GiroI - 1 'Calcula el ángulo de destino
673 Do
```

```
674 Se1Value = Se1Stop - Acel1 'Frena la rueda derecha
675 Se2Value = Se2Stop - Acel2 'Acelera la rueda izquierda
676 Call Pulsos 'Manda los pulsos a los servos
677 Call Dir 'Lee el rumbo actual
678 If Salto > 0 Then 'Gira también la cámara
679 If NuevaPos <> DirACT Then
680 NuevaPos = DirACT 'Calcula desde la posición actual
681 Se4Value = Se4Value - Salto 'y le quita el salto
682 If Se4Value < Se4Med Then 'Hasta que esté en la posición
683 Se4Value = Se4Med 'central
684 End If
685 End If
686 Call PulseOut(Se4, Se4Value, 1) 'Pulso servo horizontal
687 End If
688 If DirACT <= GiroI Then 'Si todavía no ha llegado
689 DirACT = DirACT + Ofset 'actualiza la dirección actual
690 End If
691 If DirACT - Destino < 20 Then 'Cuando falta poco para llegar
692 Acel1 = 40 'gira más lentamente para no pasarse
693 Acel2 = 40
694 End If
695 If DirACT <= Destino Then 'Si se alcanza el destino se sale
696 Exit Do
697 End If
698 'Evita que se salte por debajo del 0
699 If Destino < 10 Then 'Destino entre 0 y 10
700 If DirACT > 350 Then 'Si se pasa del 0 sale
701 Exit Do
702 End If
703 End If
704 Loop
705 End Sub
706 'Subrutina que hace que el robot realice un giro hacia la derecha con el valor pasado en
    grados
707 Sub Derecha(ByVal GiroD As Integer, ByVal Salto As Integer)
708 Dim Destino As Integer
709 Dim Ofset As Integer 'Sirve como valor diferencial
710 Dim NuevaPos As Integer 'Posición que se alcanza cada vez
711 Acel1 = 75
712 Acel2 = 75
713 Se1Value = Se1Stop
714 Se2Value = Se2Stop
715 Call Dir 'Comprueba el rumbo actual
716 Destino = DirACT + GiroD - 1 'Calcula el ángulo de destino
717 Ofset = 0
718 If Destino > 359 Then 'Si el rumbo actual es menor que el
719 Destino = Destino - 360 'giro que hay que dar, se suma una
720 Ofset = 360 'vuelta completa para hacer los cálculos
721 End If
722 Do
723 Se1Value = Se1Stop + Acel1 'Frena la rueda derecha
724 Se2Value = Se2Stop + Acel2 'Acelera la rueda izquierda
725 Call Pulsos 'Manda los pulsos a los servos
726 Call Dir
727 If Salto > 0 Then 'Gira también la cámara
728 If NuevaPos <> DirACT Then
729 NuevaPos = DirACT 'Calcula desde la posición actual
730 Se4Value = Se4Value + Salto 'y le suma el salto
731 If Se4Value > Se4Med Then 'Hasta que esté en la posición
732 Se4Value = Se4Med 'central
733 End If
734 End If
735 Call PulseOut(Se4, Se4Value, 1) 'Pulso servo horizontal
```

```
736 End If
737 If DirACT > 360 - GiroD Then 'Si la dirección actual es mayor de 360
738 DirACT = DirACT - Ofset 'Hay que quitarle el offset para obtener el
739 End If 'valor real
740 If Destino - DirACT < 20 Then 'Cuando esta cerca del destino
741 Acel1 = 40 'frena para girar más despacio
742 Acel2 = 40 'y no pasarse
743 End If
744 If DirACT >= Destino Then 'Si se llega al destino se para
745 Exit Do
746 End If
747 'Evita que se pase del 359
748 If Destino > 350 Then
749 If DirACT < 10 Then
750 Exit Do
751 End If
752 End If
753 Loop
754 End Sub
755 'Subrutina que hace que el robot avance recto
756 Sub Adelante()
757 Dim N As Integer
758 Se1Value = Se1Stop 'Parte de la posición de parados
759 Se2Value = Se2Stop
760 For N = 1 To 50 'Este número controla la cantidad
761 Se1Value = Se1Value + ServoAce1 'Si no hay obstáculos rueda recto
762 Se2Value = Se2Value - ServoAce2 'Rueda derecha e izquierda adelante
763 Call Pulsos 'Envía los impulsos a los servos
764 Next
765 End Sub
766 'Subrutina que hace que el robot avance recto más tiempo
767 Sub MAdelante()
768 Dim N As Integer
769 Se1Value = Se1Stop 'Parte de la posición de parados
770 Se2Value = Se2Stop
771 For N = 1 To 100 'Esta variable controla la cantidad
772 Se1Value = Se1Value + ServoAce1 'Si no hay obstáculos rueda recto
773 Se2Value = Se2Value - ServoAce2 'Rueda derecha e izquierda adelante
774 Call Pulsos 'Envía los impulsos a los servos
775 Next
776 End Sub
777 'Subrutina que hace que el robot retroceda hacia atrás
778 Sub Atras()
779 Dim N As Integer
780 Se1Value = Se1Stop 'Parte de la posición de parados
781 Se2Value = Se2Stop
782 For N = 1 To 50 'Este número controla la cantidad
783 Se1Value = Se1Value - ServoAce2 'Si no hay obstáculos rueda recto
784 Se2Value = Se2Value + ServoAce1 'Rueda derecha e izquierda adelante
785 Call Pulsos 'Envía los impulsos a los servos
786 Next
787 End Sub
788
789 'Subrutina que hace que el robot retroceda la distancia indicada
790 Sub Atraspar(ByVal Pasos As Integer,ByVal Salto As Integer)
791
792   Dim N As Integer
793   Se1Value = Se1Stop                    'Parte de la posición de parados
794   Se2Value = Se2Stop
795   For N = 1 To Pasos                              'Este número controla la cantidad
796     Se1Value = Se1Value - ServoAce2     'Si no hay obstáculos rueda recto
797     Se2Value = Se2Value + ServoAce1    'Rueda derecha e izquierda adelante
798     Call Pulsos                                   'Envía los impulsos a los servos
```

```
799   Next
800
801 End Sub
802
803 'Subrutina que hace que el robot avance la distancia indicada
804 Sub Adelantepar(ByVal Pasos As Integer,ByVal Salto As Integer)
805
806  Dim N As Integer
807   Se1Value = Se1Stop                        'Parte de la posición de parados
808   Se2Value = Se2Stop
809   For N = 1 To Pasos                    'Este número controla la cantidad
810     Se1Value = Se1Value + ServoAce1    'Si no hay obstáculos rueda recto
811     Se2Value = Se2Value - ServoAce2     'Rueda derecha e izquierda adelante
812     Call Pulsos                                  'Envía los impulsos a los servos
813   Next
814 End Sub
815
816 'Subrutina que comprueba estado de paragolpes delateros y sensor de inclinación y lo
    envia por el puerto serie
817 Sub Check ()
818   Dim DInclinacion As String * 1
819   DInclinacion = "1"                            'Lo pone en formato de 3 bytes
820   If Getpin(Inc) = 0 Then                        'Comprueba el sensor de inclinación
821     DInclinacion = "0"                          'Lo pone en formato de 3 bytes
822   End If
823   Dim DIzquierda As String * 1
824   DIzquierda = "0"                        'Lo pone en formato de 3 bytes
825   If Getpin(Bul) = 0 Then                        'Comprueba el paragolpes izquierdo
826     DIzquierda = "1"                            'Lo pone en formato de 3 bytes
827   End If
828    Dim DDerecha As String * 1
829   DDerecha = "0"                                'Lo pone en formato de 3 bytes
830   If Getpin(Bur) = 0 Then                        'Comprueba el paragolpes derecho
831     DDerecha = "1"                              'Lo pone en formato de 3 bytes
832   End If
833 Call PutQueueStr(OutBuff, "CHECK" & DInclinacion & DIzquierda & DDerecha) 'Manda el
    comando y  1 byte por el p.serie
834 End Sub
835
836
837 'Subrutina que hace que el robot gire hacia el angulo indicado por el lado más corto.

838 Sub GRum(ByVal Grados As Integer, ByVal Salto As Integer)
839
840 Acel1 = 80 'Establece un valor de aceleración
841 Acel2 = 80 'rápido para los dos servos
842 Call Dir 'Comprueba el rumbo actual
843 If DirACT < Grados\2 Then
844 Do
845 Call Dir 'Comprueba la dirección
846 If DirACT > Grados-15 Then 'Cuando esta próximo
847 Acel1 = 40 'frena reduciendo la
848 Acel2 = 40 'aceleración
849 End If
850 If DirACT < Grados+10 Then 'Evita que se pase
851 Exit Do
852 End If
853 Se1Value = Se1Stop + Acel1 'Frena la rueda derecha y
854 Se2Value = Se2Stop + Acel2 'acelera la rueda izquierda
855 Call Pulsos 'mientras gira
856 Loop
857 Else 'Si es menos de 180
858 Do 'Giro por la izquierda
```

```
Call Dir
If DirACT < Grados + 15 Then 'Cuando esta próximo
Acel1 = 40 'frena reduciendo la
Acel2 = 40 'aceleración
End If
If DirACT > Grados -10 Then 'Evita que se pase
Exit Do 'saliendo del bucle
End If
Se1Value = Se1Stop - Acel1 'Adelante la rueda derecha
Se2Value = Se2Stop - Acel2 'atrás la rueda izquierda
Call Pulsos 'mientras gira
Loop
End If
End Sub


'Subrutina que recupera los valores de parada de los servos desde las posiciones 100 y 101 de la eeprom
Sub ServoStop()
Dim Data As Byte
Data = PersistentPeek(100) 'Recupera el primer valor
Se1Stop = CInt(Data) + 1300 'Lo guarda como un integer
Data = PersistentPeek(101) 'Recupera el segundo valor
Se2Stop = CInt(Data) + 1300 'Lo guarda como un integer
Debug.Print "memoria"
Debug.Print CStr(Se1Stop)
Debug.Print CStr(Se2Stop)
End Sub
'Subrutina que recupera los valores de trabajo de los servos de la cámara
Sub CamStop()
Dim Data As Byte
Data = PersistentPeek(102) 'Recupera el primer valor
Se3Min = 1300 + CInt(Data) * 10 'Lo guarda como un integer
Data = PersistentPeek(103) 'Recupera el segundo valor
Se3Max = Se3Min + CInt(Data) * 10 'Lo guarda como un integer
Data = PersistentPeek(104) 'Recupera el primer valor
Se4Min = 800 - CInt(Data) * 10 'Lo guarda como un integer
Data = PersistentPeek(105) 'Recupera el segundo valor
Se4Med = 1200 + CInt(Data) * 10 'Lo guarda como un integer
Data = PersistentPeek(106) 'Recupera el tercer valor
Se4Max = 1800 + CInt(Data) * 10 'Lo guarda como un integer
End Sub
'Subrutina que hace un Beep Beep por el altavoz y enciende los leds
Public Sub Beep_Beep()
Dim FBase As Integer
FBase = 2000 '2000 es la frecuencia de resonancia
Call PutPin(Led, Lon) 'Enciende los leds
Call FreqOut(Buz, FBase, FBase, 200) 'del altavoz y por lo tanto es la que
Call FreqOut(Buz, 0, 0, 40) 'produce un mayor ruido
Call FreqOut(Buz, FBase, FBase, 200) 'Puede cambiarse para obtener otros tonos.
Call PutPin(Led, Lof) 'Apaga los leds
End Sub
'Subrutina que hace un Beep por el altavoz
Public Sub Beep()
Dim FBase As Integer
FBase = 2000
Call FreqOut(Buz, FBase, FBase, 100) 'Hace el beep simple
End Sub
'Subrutina que mide la distancia con el sensor SRF08 y lo envía por el p. serie
Sub Sonar()
Dim DRango As String * 3 'Valor de la distancia como cadena
Call I2cByteWrite(Sonar1, SrfReg, RangoCmd) 'Comando de hacer la medición en Cm
Call Sleep(38) 'Espera 70 ms para que termine
```

```
921 Dist = I2CWordRead(Sonar1, RangoReg) 'Lee la distancia medida
922 If Dist > 999 Then 'Limita el valor a 3 carácteres
923 Dist = 999 'para evitar un error en la
924 End If 'recepción
925 DRango = CStr(Dist) 'Convierte el valor de la distancia
926 If Len(DRango) < 3 Then 'en una cadena de 3 caracteres de
927 DRango = "0" & DRango 'longitud fija, para enviarlos
928 End If 'por el puerto serie sin problemas
929 If Len(DRango) < 3 Then 'en la recepción
930 DRango = "0" & DRango
931 End If
932 Call PutQueueStr(OutBuff, "TELDI" & DRango) 'Manda el comando y 3 bytes por el p.serie
933 End Sub
934 'Subrutina que mide la luz con el sensor SRF08 y lo envía por el p. serie
935 Sub Luminosidad()
936 Dim DLuz As String * 3 'Valor de la luminosidad como cadena
937 Call I2cByteWrite(Sonar1, SrfReg, RangoCmd) 'Comando de hacer la medición en Cm
938 Call Sleep(38) 'Espera 70 ms para que termine
939 Lux = I2CByteRead(Sonar1, LuzReg) 'Lee el sensor de luz
940 DLuz = CStr(Lux) 'Convierte el valor de la luz en
941 If Len(DLuz) < 3 Then 'una cadena de 3 caracteres de
942 DLuz = "0" & DLuz 'longitud fija
943 End If
944 If Len(DLuz) < 3 Then
945 DLuz = "0" & DLuz
946 End If
947 Call PutQueueStr(OutBuff, "TELLU" & DLuz) 'Manda el comando y 3 bytes por el p.serie
948 End Sub
949 'Subrutina que mide la dirección con el sensor brújula y lo asigna a DirACT para los
    giros
950 Sub Dir()
951 Rumbo = I2CWordRead(Brujula1, RumboReg) 'Lee el registro de la dirección en grados
952 Rumbo = Rumbo \ 10 'Le quita los decimales
953 DirACT = CInt(Rumbo)
954 End Sub
955 'Subrutina que mide la dirección con el sensor brújula digital y lo envía por el p. serie
956 Sub Brujula()
957 Dim DRumbo As String * 3 'Valor del rumbo en grados como cadena
958 Dim GRumbo As New UnsignedInteger 'Valor del rumbo sin decimas
959 Rumbo = I2CWordRead(Brujula1, RumboReg) 'Lee el registro de la dirección en grados
960 GRumbo = Rumbo \ 10 'Le quita los decimales
961 DRumbo = CStr(GRumbo) 'Convierte el valor del rumbo en
962 If Len(DRumbo) < 3 Then 'una cadena de 3 caracteres de
963 DRumbo = "0" & DRumbo 'longitud fija
964 End If
965 If Len(DRumbo) < 3 Then
966 DRumbo = "0" & DRumbo
967 End If
968 Call PutQueueStr(OutBuff, "TELRU" & DRumbo) 'Manda el comando y 3 bytes por el p.serie
969 End Sub
970 'Subrutina que mide la temperatura con el sensor Ds1820 mediante bus de un solo hilo. La
    temperatura se devuelve como un valor de medios grados por lo que para pasarla a grados
    hay q dividirla por 2
971 Public Sub Tempera()
972 Dim A As Byte
973 Dim DTemperatura As String
974 Dim Signo As Byte
975 Call Reset1W 'Resetea el bus de 1 hilo
976 Call BWrite1W(&HCC) 'Manda comando de salto de rom
977 Call BWrite1W(&H44) 'Comando de medir la temperatura
978 Call Sleep(375) 'Retardo de 750 ms
979 Call Reset1W 'Resetea el bus de 1 hilo
980 Call BWrite1W(&HCC) 'Manda comando de salto de rom
```

```
981  Call BWrite1W(&HBE) 'Comando de lectura del registro
982  Temperatura = CSng(BRead1W()) 'lee el byte de la temperatura
983  Signo = BRead1W() 'y el signo
984  If Signo > 0 Then 'Si es negativo le hace el complemento
985  Temperatura = Temperatura - 256# 'a dos para convertirlo
986  End If
987  DTemperatura = CStr(Temperatura) 'Convierte el valor de la temperatura en
988  If Len(DTemperatura) < 3 Then 'una cadena de 3 caracteres de
989  DTemperatura = "0" & DTemperatura 'longitud fija
990  End If
991  If Len(DTemperatura) < 3 Then
992  DTemperatura = "0" & DTemperatura
993  End If
994  Call PutQueueStr(OutBuff, "TELTE" & DTemperatura) 'Manda el comando y 3 bytes por el
     p.serie
995  End Sub
996  'Subrutina que comprueba el sensor de inclinación y manda su estado 0 o 1 por el p.
     serie. 1 significa que esta normal. 0 que está inclinado
997  Public Sub Inclinacion()
998  Dim DInclinacion As String * 3
999  DInclinacion = "001" 'Lo pone en formato de 3 bytes
1000 If Getpin(Inc) = 0 Then 'Comprueba el sensor de inclinación
1001 DInclinacion = "000" 'Lo pone en formato de 3 bytes
1002 End If
1003 Call PutQueueStr(OutBuff, "TELIN" & DInclinacion) 'Manda el comando y 1 byte por el
     p.serie
1004 End Sub
1005 'Subrutina que comprueba el paragolpes izquierdo y manda su estado 0 o 1 por el p. serie.
     1 significa que esta normal. 0 que está presionado
1006 Public Sub BIzquierda()
1007 Dim DIzquierda As String * 3
1008 DIzquierda = "001" 'Lo pone en formato de 3 bytes
1009 If Getpin(Bul) = 0 Then 'Comprueba el paragolpes izquierdo
1010 DIzquierda = "000" 'Lo pone en formato de 3 bytes
1011 End If
1012 Call PutQueueStr(OutBuff, "TELIZ" & DIzquierda) 'Manda el comando y 1 byte por el p.serie
1013 End Sub
1014 'Subrutina que comprueba el paragolpes derecho y manda su estado 0 o 1 por el p. serie.
     1 significa que esta normal. 0 que está presionado
1015 Public Sub BDerecha()
1016 Dim DDerecha As String * 3
1017 DDerecha = "001" 'Lo pone en formato de 3 bytes
1018 If Getpin(Bur) = 0 Then 'Comprueba el paragolpes derecho
1019 DDerecha = "000" 'Lo pone en formato de 3 bytes
1020 End If
1021 Call PutQueueStr(OutBuff, "TELDE" & DDerecha) 'Manda el comando y 1 byte por el p.serie
1022 End Sub
1023 'RUTINAS DE COMUNICACIÓN I2C
1024 'Subrutina que escribe un Byte en el Bus I2C
1025 Sub I2cByteWrite(ByVal I2cAddr As Byte, ByVal I2cReg As Byte, ByVal I2cData As Byte)
1026 Call I2cStart 'Manda el STAR
1027 Call I2cOutByte(I2cAddr) 'Envía la dirección
1028 Call I2cOutByte(I2cReg) 'Envía número de registro
1029 Call I2cOutByte(I2cData) 'Envía el dato
1030 Call I2cStop 'Manda el STOP
1031 End Sub
1032 'Subrutina que lee un byte por el bus I2C
1033 Function I2CByteRead(ByVal I2cAddr As Byte, ByVal I2cReg As Byte) As Byte
1034 Call I2cStart 'Manda el STAR
1035 Call I2cOutByte(I2cAddr) 'Envía la dirección
1036 Call I2cOutByte(I2cReg) 'Envía número de registro
1037 Call I2cStart 'Envía start
1038 I2cAddr = I2cAddr + 1 'Pone modo de lectura
```

```
1039 Call I2cOutByte(I2cAddr) 'Envía la dirección de lectura
1040 I2cAck = False 'Envía Nak
1041 I2CByteRead = I2cInByte() 'Lee el byte con Nak
1042 Call I2cStop 'Manda el STOP
1043 End Function
1044 'Subrutina que lee una palabra (2 Bytes) desde el Bus I2C
1045 Function I2CWordRead(ByVal I2cAddr As Byte, ByVal I2cReg As Byte) As UnsignedInteger
1046 Set I2CWordRead = New UnsignedInteger
1047 Call I2cStart 'Manda el STAR
1048 Call I2cOutByte(I2cAddr) 'Envía la dirección
1049 Call I2cOutByte(I2cReg) 'Envía número de registro
1050 Call I2cStart 'Envía start
1051 I2cAddr = I2cAddr + 1 'Pone modo de lectura
1052 Call I2cOutByte(I2cAddr) 'Envía la dirección de lectura
1053 I2cAck = True 'Envía Ack
1054 I2CWordRead = CuInt(I2cInByte() * 256) 'Lee el primer byte
1055 I2cAck = False 'Envía Nak
1056 I2CWordRead = I2CWordRead + CuInt(I2cInByte()) 'Lee el siguiente byte y lo suma
1057 Call I2cStop 'Manda el STOP
1058 End Function
1059 'Subrutina que envía un Byte por el bus I2C
1060 Sub I2cOutByte(I2cData As Byte)
1061 Call ShiftOut(Sda, Scl, 8, I2cData) 'Saca el dato secuencialmente
1062 Call PutPin(Sda, bxInputTristate) 'Pin de datos en alta impedancia
1063 Call PutPin(Scl, bxOutputHigh) 'Pone SCL a 1
1064 Call PutPin(Scl, bxOutputLow) 'Y luego a 0
1065 End Sub
1066 'Subrutina que recibe un Byte por el Bus I2C
1067 Function I2cInByte() As Byte
1068 I2cInByte = ShiftIn(Sda, Scl, 8) 'Lee el dato de forma secuencial
1069 If I2cAck = True Then 'Indica que se va a recibir más
1070 Call PutPin(Sda, bxOutputLow)
1071 Else
1072 Call PutPin(Sda, bxOutputHigh) 'Si no se va a recibir más
1073 End If
1074 Call PutPin(Scl, bxOutputHigh) 'Pone a SCL a 1
1075 Call PutPin(Scl, bxOutputLow) 'y luego a 0
1076 End Function
1077 'Subrutina del Comando Start de I2C combinando las líneas SDA y SCL
1078 Sub I2cStart()
1079 Call PutPin(Sda, bxOutputHigh)
1080 Call PutPin(Scl, bxOutputHigh)
1081 Call PutPin(Sda, bxOutputLow)
1082 Call PutPin(Scl, bxOutputLow)
1083 End Sub
1084 'Subrutina del Comando Stop del I2C combinando las líneas SDA y SCL
1085 Sub I2cStop()
1086 Call PutPin(Sda, bxOutputLow)
1087 Call PutPin(Scl, bxOutputHigh)
1088 Call PutPin(Sda, bxOutputHigh)
1089 End Sub
1090 'RUTINAS PARA LA COMUNICACIÓN POR EL BUS DE UN SOLO HILO
1091 'Subrutina que manda el Byte Dato por el bus 1W del pin1W
1092 Sub BWrite1W(ByVal Dato As Byte)
1093 Dim I As Integer
1094 Dim B As Byte
1095 For I = 0 To 7 'Son 8 bits del Byte a mandar
1096 B = Dato And 1 'Determina el valor del primer bit
1097 If B = 1 Then
1098 Call Put1Wire(Tem, 1) 'Si es uno manda un 1 por bus 1W
1099 Else
1100 Call Put1Wire(Tem, 0) 'Manda un 0 por el bus 1W
1101 End If
```

```
1102 Dato = Dato \ 2 'Rota a la derecha dividiendo por 2
1103 Next
1104 End Sub
1105 'Función que devuelve el byte leído del bus 1W en el pin Pin1W
1106 Function BRead1W() As Byte
1107 Dim I As Integer
1108 Dim B As Byte
1109 For I = 0 To 7 'Son 8 bits del Byte a mandar
1110 B = Get1Wire(Tem) 'Lee el valor del bit
1111 BRead1W = BRead1W \ 2 'Rota a la derecha dividiendo por 2
1112 If B = 1 Then
1113 BRead1W = BRead1W Or 128 'Si es uno pone un 1 en el Byte
1114 End If
1115 Next
1116 End Function
1117 'Subrutina que resetea el bus 1W que está en el pin Pin1W. Antes de enviar cualquier
     dato por el bus hay que hacer un reset, consiste en un impulso a nivel bajo de al menos
     500µs, luego se deja el pin en el modo de entrada y se espera otros 300µs para que el
     DS1820 mande su impulso bajo indicando que está listo
1118 Sub Reset1W()
1119 Dim I As Integer
1120 Call PutPin(Tem, 2) 'Pone el pin en entrada
1121 Call PutPin(Tem, 0) 'Pone un pulso a nivel bajo
1122 For I = 1 To 3 'Retardo de 500 µS
1123 Next
1124 Call PutPin(Tem, 2) 'Pone el pin en entrada
1125 For I = 1 To 3 'Retardo de 500 µS
1126 Next
1127 End S
```