

```

001 ; slave
002 ; CONFIGURACIONES
003 #picaxe 40x2          ' PICAXE-28X2-5V (18F25K22)
004 setfreq m8
005 ;*****
006 ;*** Configuracio comandament infrarroig          ***
007 ;*****
008 symbol codi_davant=$0D
009 symbol codi_enrere=$16
010 symbol codi_dreta=$10
011 symbol codi_esquerra=$11
012 symbol codi_aturar=$15
013 symbol codi_arc_esquerra=$12
014 symbol codi_arc_dreta=$13
015 symbol detector_IR=C.0
016 ;*****
017 ;*** Configuracio sensors          40X2          ***
018 ;*****
019 symbol sensor_dreta=3
020 symbol sensor_esquerra=2
021 symbol sensor_0=A.0
022 symbol sensor_1=A.1
023 symbol sensor_2=A.2
024 symbol sensor_3=A.3
025 symbol sensor_5=A.5
026 symbol sensor_6=A.6
027 symbol sensor_7=A.7
028 symbol led_5=C.5
029 symbol led_3=B.3
030 symbol led_2=B.2
031 symbol led_1=B.1
032 symbol led_0=B.0
033 symbol led_dreta=B.3
034 symbol led_esquerra=B.2
035 symbol n_conversions_adc=4
036 ;*****
037 ;*** Configuracio botons          40X2          ***
038 ;*****
039 symbol pujar = pinC.6
040 symbol baixar = pinC.7
041 ;*****
042 ;*** Configuracio bridge motors          40x2          ***
043 ;*****
044 symbol Positiu_Mdreta = B.5
045 symbol Negatiu_Mdreta = B.4
046 symbol Enable_Mdreta = C.2
047 symbol Positiu_Mesquerra = B.7
048 symbol Negatiu_Mesquerra = B.6
049 symbol Enable_Mesquerra = C.1
050 high Enable_Mdreta
051 high Enable_Mesquerra
052 ;*****
053 ;*** Definicio de variables          ***
054 ;*****
055 symbol codi_comand=b0 ; codi premut del comandament IR
056 symbol taula_veritat=b1 ; Combinaci? decimal taula veritat de rastrejador
057 symbol left_dark=b2 ; Lectura sensor esquerra llum ambient
058 symbol left_light=b3 ; Lectura sensor esquerra reflexio
059 symbol left_sensor=b4 ; Lectura esquerra
060 symbol left_sensor_=b5 ;
061 symbol right_dark=b6 ; Lectura sensor dreta llum ambient
062 symbol right_light=b7 ; Lectura sensor dreta reflexio
063 symbol right_sensor=b8; Lectura dreta
064 symbol right_sensor_=b9 ; Lectura dreta
065 symbol i2cbus=b10 ;
066 symbol shadow=b11 ; Per interrupcions

```

```

067 symbol AA=b12 ; index bucle de test comandament
068 symbol BB=b13 ; index bucle de test comandament
069 symbol comande=20
070 symbol codi=b19
071 symbol codi_=b20
072 symbol optoA=b21
073 symbol optoB=b22
074 symbol PWMA=b25
075 symbol PWMB=b26
076 symbol aleatori=b27
077 symbol done=b28
078 symbol done_tim=b29
079 symbol graf=b30
080 symbol lim_compte=b31
081 symbol comptador=b32
082
083 symbol k=b40
084 symbol adc_acum=w23
085 ;*****
086 ;DEFINICIO DE CONSTANTS
087 ;*****
088 symbol temps_adc=10
089 symbol llindar_dreta = 4
090 symbol llindar_esquerra = 4
091 ;*****
092 ;CONFIGURACIO BUS I2C
093 ;*****
094 ;let i2cbus=$E8
095 call codificar_portd
096 hi2csetup i2cslave,i2cbus
097 pause 1000 ; imprescindible no quantificat
098 ;*****
099 ;** Scratchpad I2C **
100 ;*****
101 symbol @i2c_com=20
102 symbol @i2c_optoA=21
103 symbol @i2c_opto_left=21
104 symbol @i2c_optoB=22
105 symbol @i2c_opto_right=22
106 symbol @i2c_PWMA=25
107 symbol @i2c_PWMB=26
108 symbol @i2c_random=27
109 symbol @i2c_done=28
110 symbol @i2c_done_tim=29
111 symbol @i2c_graf=30
112 ;*****
113 ;** Codi control cinta **
114 ;*****
115 symbol cin_aturar=30 ; cinta aturada
116 symbol cin_davant=31 ; cinta marxa davant
117 symbol cin_enrere=32 ; cinta marxa enrere
118 symbol cin_A_left=40 ; cinta marxa davant fins left
119 symbol cin_A_right=41 ; cinta marxa enrere fins right
120 symbol cin_A_left_tim=42 ; cinta marxa enrere fins right + tim
121 symbol cin_A_right_tim=43 ; cinta marxa enrere fins right + tim
122 symbol cin_reset=49 ; cinta marxa enrere fins right + tim
123 ;*****
124 ;PROGRAMA PRINCIPAL INICIAL
125 ;*****
126 call aturar ; cinta aturada
127 call control_pwma ; ACTICA PWMA
128 ;setintflags %01000000,%01000000 ; Activa interrupcions I2C
129 let done=1
130 let graf=0
131 put @i2c_done,done
132 let done_tim=1

```

```

133 put @i2c_done_tim,done_tim
134 let lim_compte=30
135 let comptador=0
136 let codi=0
137 let codi_=0
138 ;*****
139 ;PROGRAMA PRINCIPAL REITERATIU
140 ;*****
141 setint %00000000,%00000001; activar aquesta líia en test comandament
142 main;; nomes activar un test. Provar-los individualment.
143 ;call test_motors ; 40x2 endavant,enrere; dreta, esquerra.
144 ;call test_detectors ; 40x2 NO DETECTA: enrere, DETECTA: davant.
145 ;call test_botons ; 40x2 BAIJAR:endavant esqu PUJAR: davant dreta
146 ;call test_leds_sensors ; 40x2 ESQUERRA: rapid, DRETA: lent
147 ;call test_encoders ; 40x2 les sortides en leds B.3, B.2, B.1, B.0
148 call test_comandament ; 40x2 activar interrupcions.
149 goto main
150 ;*****
151 control_pwm:
152 pwmout pwmdiv16, Enable_Mdreta, 124, 499 ; 1000Hz at 100% @ 8MHz
153 pwmout pwmdiv16, Enable_Mesquerra, 124, 499 ; 1000Hz at 100% @ 8MHz
154 return
155 ;*****
156 ;SECUENCIA D'INTERRUPCIO
157 ;*****
158 interrupt: ; el sensor ha rebut algo
159 LET pinsc=%00000000 ; deixa el display en blanc
160 irin [1,lloc_timeout],detector_IR,codi_comand ; llegir el sensor IR
161 call codi_comandament ; convertir el codi llegit
162 lloc_timeout:
163 let AA=0
164 setint %00000000,%00000001 ; reactivar interrupcio
165 return
166 ;*****
167 ;SUBROUTINES
168 ;*****
169 test_comandament:
170 ;setint %00000000,%00000001
171 pause 10
172 if AA>100 then ; DESPR'S DE PREMIER UN BOTO DEL COMANDAMENT
173 call restituir ; DURANT 100 VEGADES*10ms= 1SEGON fa el
174 else inc AA ; mateix moviment
175 endif
176 debug codi_comandament
177 return
178 restituir:
179 call aturar
180 let AA=0
181 return
182 codi_comandament:
183 select case codi_comand
184 case codi_davant
185 call davant
186 case codi_enrere
187 call enrere
188 case codi_dreta
189 call dreta
190 case codi_esquerra
191 call esquerra
192 case codi_aturar
193 call aturar
194 case codi_arc_esquerra
195 call arc_esquerra
196 case codi_arc_dreta
197 call arc_dreta
198 endselect

```

```

199 return
200 llegir_esquerra:
201 let k=n_conversions_adc ; defineix el bucle
202 let adc_acum=0 ; esborra acumulador
203 llegir_esquerra_:
204 low led_esquerra ; apaga LED
205 pause temps_adc
206 readadc sensor_esquerra,left_dark ; llegeix dark
207 high led_esquerra ; encen LED
208 pause temps_adc
209 readadc sensor_esquerra,left_light ; llegeix light
210 pause temps_adc
211 low led_esquerra ; apaga LED
212 if left_light>left_dark then
213     let left_sensor=left_light-left_dark
214     else
215         let left_sensor=0
216 endif
217 let left_sensor_=0 ; MSB=0
218 let adc_acum=adc_acum+left_sensor ; acumula LSB
219 dec k
220 if k<>0 then goto llegir_esquerra_
221 let left_sensor=adc_acum/n_conversions_adc ; comput final
222 end_llegir_esquerra:
223 put @i2c_optoA,left_sensor ; Desa valor a scrтчpad I2C
224 let optoA=left_sensor
225 return
226 llegir_dreta:
227 let k=n_conversions_adc ; defineix el bucle
228 let adc_acum=0 ; esborra acumulador
229 llegir_dreta_:
230 low led_dreta ; apaga LED
231 pause temps_adc
232 readadc sensor_dreta,right_dark ; llegeix foscior
233 high led_dreta ; ENCEN LED
234 pause temps_adc
235 readadc sensor_dreta,right_light ; llegeix llum
236 pause temps_adc
237 low led_dreta ; apaga LED
238 if right_light>right_dark then
239     ; llun superior a foscior
240     let right_sensor=right_light-right_dark
241     else
242         let right_sensor=0
243 endif
244 let right_sensor_=0 ; MSB=0
245 let adc_acum=adc_acum+right_sensor ; acumula LSB
246 dec k
247 if k<>0 then goto llegir_dreta_
248 let right_sensor=adc_acum/n_conversions_adc ; comput final
249 ;let @i2c_opto_left=0
250 let optoB=right_sensor
251 put @i2c_optoB,right_sensor
252 ;poke right_sensor,optoB
253
254 ;debug
255 ;pause 100
256 return
257 ;*****
258 ;TEST DISPOSITIUS
259 ;*****
260 test_detectors:
261 ;setint %00000000,%00000000
262 call test_esquerra
263 call test_dreta
264 return

```

```

265 test_dreta:
266 call llegir_dreta
267 if right_sensor>llindar_dreta then call davant_dreta
268 if right_sensor=<llindar_dreta then call aturar_dreta
269 return
270 test_esquerra:
271 call llegir_esquerra
272 if left_sensor>llindar_esquerra then call davant_esquerra
273 if left_sensor=<llindar_esquerra then call aturar_esquerra
274 return
275 test_leds_sensors:
276 high led_3
277 pause 200
278 low led_3
279 high led_2
280 pause 200
281 low led_2
282 high led_1
283 pause 200
284 low led_1
285 high led_0
286 pause 200
287 low led_0
288 high led_5
289 pause 200
290 low led_5
291 return
292 test_motors:
293 if baixar=1 then goto test_motors_
294 RETURN
295 test_motors_:
296 call davant
297 pause 3000
298 call enrere
299 pause 3000
300 call dreta
301 pause 3000
302 call esquerra
303 pause 1000
304 call aturar
305 pause 5000
306 return
307 test_botons:
308 if pujar=1 then call davant_dreta
309 if pujar=0 then call aturar_dreta
310 if baixar=1 then call davant_esquerra
311 if baixar=0 then call aturar_esquerra
312 if pujar=1 then high led_dreta endif
313 if pujar=0 then low led_dreta endif
314 if baixar=1 then high led_esquerra endif
315 if baixar=0 then low led_esquerra endif
316 return
317 return
318 test_encoders:
319 ;setintflags %00000000,%00000000 ; Activa interrupcions I2C
320 call codificar_portd
321
322 if pinC.6=1 then
323     let i2cbus = i2cbus and %11110000
324     if i2cbus>=128 then
325         let i2cbus=i2cbus-128
326         high B.3
327     else low B.3
328     endif
329     if i2cbus>=64 then
330         let i2cbus=i2cbus-64

```

```

331         high B.2
332         else low B.2
333     endif
334     if i2cbus>=32 then
335         let i2cbus=i2cbus-32
336         high B.1
337         else low B.1
338     endif
339     if i2cbus>=16 then
340         let i2cbus=i2cbus-16
341         high B.0
342         else low B.0
343     endif
344 endif
345 if pinC.6=0 then
346     let i2cbus = i2cbus and %00001111
347     if i2cbus>=8 then
348         let i2cbus=i2cbus-8
349         high B.3
350         else low B.3
351     endif
352     if i2cbus>=4 then
353         let i2cbus=i2cbus-4
354         high B.2
355         else low B.2
356     endif
357     if i2cbus>=2 then
358         let i2cbus=i2cbus-2
359         high B.1
360         else low B.1
361     endif
362     if i2cbus>=1 then
363         let i2cbus=i2cbus-1
364         high B.0
365         else low B.0
366     endif
367 endif
368 return
369 codificar_portd:
370 let BB=0
371 let dirsd=%00000000; 0 entrada 1 sortida
372     if pind.0=1 then let BB=BB+2 endif
373     if pind.1=1 then let BB=BB+1 endif
374     if pind.2=1 then let BB=BB+8 endif
375     if pind.3=1 then let BB=BB+4 endif
376     if pind.4=1 then let BB=BB+32 endif
377     if pind.5=1 then let BB=BB+16 endif
378     if pind.6=1 then let BB=BB+128 endif
379     if pind.7=1 then let BB=BB+64 endif
380 LET i2cbus=BB
381 return
382 ;*****
383 ;SUBROUTINES DE MOVIMENT
384 ;*****
385 soft_davant:
386 call davant
387 pause 100
388 call aturar_de_davant
389 return
390 soft_arc_esquerda:
391 call esquerda
392 call arc_esquerda
393 pause 50
394 call aturar_de_dreta
395 return
396 soft_arc_dreta:

```

```
397 call dreta
398 call arc_dreta
399 pause 50
400 call aturar_de_esquerra
401 return
402 davant:
403 low Enable_Mesquerra,Enable_Mdreta
404 high Positiu_Mdreta,Positiu_Mesquerra
405 low Negatiu_Mdreta,Negatiu_Mesquerra
406 high Enable_Mesquerra,Enable_Mdreta
407 return
408 enrere:
409 low Enable_Mesquerra,Enable_Mdreta
410 low Positiu_Mdreta,Positiu_Mesquerra
411 high Negatiu_Mdreta,Negatiu_Mesquerra
412 high Enable_Mesquerra,Enable_Mdreta
413 return
414 dreta:
415 low Enable_Mesquerra,Enable_Mdreta
416 low Positiu_Mdreta,Negatiu_Mesquerra
417 high Negatiu_Mdreta,Positiu_Mesquerra
418 high Enable_Mesquerra,Enable_Mdreta
419 return
420 arc_dreta:
421 low Enable_Mesquerra,Enable_Mdreta
422 low Positiu_Mdreta,Negatiu_Mesquerra,Negatiu_Mdreta,Positiu_Mesquerra
423 high Positiu_Mesquerra
424 high Enable_Mesquerra,Enable_Mdreta
425 return
426 esquerra:
427 low Enable_Mesquerra,Enable_Mdreta
428 low Negatiu_Mdreta,Positiu_Mesquerra
429 high Positiu_Mdreta,Negatiu_Mesquerra
430 high Enable_Mesquerra,Enable_Mdreta
431 return
432 arc_esquerra:
433 low Enable_Mesquerra,Enable_Mdreta
434 low Negatiu_Mdreta,Positiu_Mesquerra,Positiu_Mdreta,Negatiu_Mesquerra
435 high Positiu_Mdreta
436 high Enable_Mesquerra,Enable_Mdreta
437 return
438 davant_dreta:
439 low Enable_Mesquerra,Enable_Mdreta
440 high Positiu_Mdreta
441 low Negatiu_Mdreta
442 high Enable_Mesquerra,Enable_Mdreta
443 return
444 davant_esquerra:
445 low Enable_Mesquerra,Enable_Mdreta
446 high Positiu_Mesquerra
447 low Negatiu_Mesquerra
448 high Enable_Mesquerra,Enable_Mdreta
449 return
450 enrere_dreta:
451 low Enable_Mesquerra,Enable_Mdreta
452 low Positiu_Mdreta
453 high Negatiu_Mdreta
454 high Enable_Mesquerra,Enable_Mdreta
455 return
456 enrere_esquerra:
457 low Enable_Mesquerra,Enable_Mdreta
458 low Positiu_Mesquerra
459 high Negatiu_Mesquerra
460 high Enable_Mesquerra,Enable_Mdreta
461 return
462 aturar:
```

```
463 low Enable_Mesquerra, Enable_Mdreta
464 low Positiu_Mdreta, Negatiu_Mdreta
465 high Negatiu_Mesquerra, Positiu_Mesquerra
466 return
467 aturar_dreta:
468 low Enable_Mesquerra, Enable_Mdreta
469 high Positiu_Mdreta, Negatiu_Mdreta
470 return
471 aturar_esquerra:
472 low Enable_Mesquerra, Enable_Mdreta
473 high Positiu_Mesquerra, Negatiu_Mesquerra
474 return
475 aturar_de_esquerra:
476 low Enable_Mesquerra, Enable_Mdreta
477 low Negatiu_Mesquerra
478 return
479 aturar_de_dreta:
480 low Enable_Mesquerra, Enable_Mdreta
481 low Negatiu_Mdreta
482 return
483 aturar_de_davant:
484 low Enable_Mesquerra, Enable_Mdreta
485 low Negatiu_Mdreta, Negatiu_Mesquerra
486 return
487 aturar_de_enrere:
488 low Enable_Mesquerra, Enable_Mdreta
489 low Positiu_Mdreta, Positiu_Mesquerra
490 return
491
```