

```

001 ; slave
002 ; Es tracta de controlar un motor amb un potenciometre.
003 ; De 0% a 45% del potenciometre, el motor ha de girar en sentit CW.
004 ; De 55% a 100% del potenciometre, el motor ha de girar en sentit CCW.
005 ; S'utilitza la tarja FADI_40X2 i UNA TARJA FADILEDS.
006 ; CONFIGURACIONS
007 #picaxe 40x2          ' PICAXE-28X2-5V (18F25K22)
008 setfreq m8
009 ;*****
010 ;*** Configuracio comandament infrarroig          ***
011 ;*****
012 symbol codi_davant=$0D
013 symbol codi_enrere=$16
014 symbol codi_dreta=$10
015 symbol codi_esquerra=$11
016 symbol codi_aturar=$15
017 symbol codi_arc_esquerra=$12
018 symbol codi_arc_dreta=$13
019 symbol detector_IR=C.0
020 ;*****
021 ;*** Configuracio sensors          40X2          ***
022 ;*****
023 symbol sensor_dreta=3
024 symbol sensor_esquerra=2
025 symbol potenciometre=A.0
026 symbol sensor_1=A.1
027 symbol sensor_2=A.2
028 symbol sensor_3=A.3
029 symbol sensor_5=A.5
030 symbol sensor_6=A.6
031 symbol sensor_7=A.7
032 symbol led_5=C.5
033 symbol led_3=B.3
034 symbol led_2=B.2
035 symbol led_1=B.1
036 symbol led_0=B.0
037 symbol led_dreta=B.3
038 symbol led_esquerra=B.2
039 symbol n_conversions_adc=4
040 ;*****
041 ;*** Configuracio botons          40X2          ***
042 ;*****
043 symbol pujar = pinC.6
044 symbol baixar = pinC.7
045 ;*****
046 ;*** Configuracio bridge motors          40x2          ***
047 ;*****
048 symbol Positiu_Mdreta = B.5
049 symbol Negatiu_Mdreta = B.4
050 symbol Enable_Mdreta = C.2
051 symbol Positiu_Mesquerra = B.7
052 symbol Negatiu_Mesquerra = B.6
053 symbol Enable_Mesquerra = C.1
054 high Enable_Mdreta
055 high Enable_Mesquerra
056 ;*****
057 ;*** Definicio de variables          ***
058 ;*****
059 symbol codi_comand=b0 ; codi premut del comandament IR
060 symbol taula_veritat=b1 ; Combinaci? decimal taula veritat de rastrejador
061 symbol left_dark=b2 ; Lectura sensor esquerra llum ambient
062 symbol left_light=b3 ; Lectura sensor esquerra reflexio
063 symbol left_sensor=b4 ; Lectura esquerra
064 symbol left_sensor_=b5 ;
065 symbol right_dark=b6 ; Lectura sensor dreta llum ambient
066 symbol right_light=b7 ; Lectura sensor dreta reflexio

```

```

067 symbol right_sensor=b8; Lectura dreta
068 symbol right_sensor_=b9 ; Lectura dreta
069 symbol i2cbus=b10 ;
070 symbol shadow=b11 ; Per interrupcions
071 symbol AA=b12 ; index bucle de test comandament
072 symbol BB=b13 ; index bucle de test comandament
073 symbol comande=20
074 symbol codi=b19
075 symbol codi_=b20
076 symbol optoA=b21
077 symbol optoB=b22
078 symbol PWMA=b25
079 symbol PWMB=b26
080 symbol aleatori=b27
081 symbol done=b28
082 symbol done_tim=b29
083 symbol graf=b30
084 symbol lim_compte=b31
085 symbol comptador=b32
086 symbol lectura=b34
087
088 symbol k=b40
089 symbol adc_acum=w23
090 ;*****
091 ;DEFINICIO DE CONSTANTS
092 ;*****
093 symbol temps_adc=10
094 symbol llindar_dreta = 4
095 symbol llindar_esquerra = 4
096 ;*****
097 ;CONFIGURACIO BUS I2C
098 ;*****
099 ;let i2cbus=$E8
100 call codificar_portd
101 hi2csetup i2cslave,i2cbus
102 pause 1000 ; imprescindible no quantificat
103 ;*****
104 ;** Scratchpad I2C **
105 ;*****
106 symbol @i2c_com=20
107 symbol @i2c_optoA=21
108 symbol @i2c_opto_left=21
109 symbol @i2c_optoB=22
110 symbol @i2c_opto_right=22
111 symbol @i2c_PWMA=25
112 symbol @i2c_PWMB=26
113 symbol @i2c_random=27
114 symbol @i2c_done=28
115 symbol @i2c_done_tim=29
116 symbol @i2c_graf=30
117 ;*****
118 ;** Codi control cinta **
119 ;*****
120 symbol cin_aturar=30 ; cinta aturada
121 symbol cin_davant=31 ; cinta marxa davant
122 symbol cin_enrere=32 ; cinta marxa enrere
123 symbol cin_A_left=40 ; cinta marxa davant fins left
124 symbol cin_A_right=41 ; cinta marxa enrere fins righ
125 symbol cin_A_left_tim=42 ; cinta marxa enrere fins righ + tim
126 symbol cin_A_right_tim=43 ; cinta marxa enrere fins righ + tim
127 symbol cin_reset=49 ; cinta marxa enrere fins righ + tim
128 ;*****
129 ;PROGRAMA PRINCIPAL INICIAL
130 ;*****
131 call aturar ; cinta aturada
132 call control_pwma ; ACTICA PWMA

```

```

133 ;setintflags %01000000,%01000000 ; Activa interrupcions I2C
134 let done=1
135 let graf=0
136 put @i2c_done,done
137 let done_tim=1
138 put @i2c_done_tim,done_tim
139 let lim_compte=30
140 let comptador=0
141 let codi=0
142 let codi_=0
143 ;*****
144 ;PROGRAMA PRINCIPAL REITERATIU
145 ;*****
146 ;setint %00000000,%00000001; activar aquesta l?nia en test comandament
147 main:; nomes activar un test. Provar-los individualment.
148 ;call test_motors ; 40x2 endavant,enrere; dreta, esquerra.
149 ;call test_detectors ; 40x2 NO DETECTA: enrere, DETECTA: davant.
150 ;call test_botons ; 40x2 BAIJAR:endavant esqu PUJAR: davant dreta
151 ;call test_leds_sensors ; 40x2 ESQUERRA: rapid, DRETA: lent
152 ;call test_encoders ; 40x2 les sortides en leds B.3, B.2, B.1, B.0
153 ;call test_comandament ; 40x2 activar interrupcions.
154 call pwm_control_pot
155 goto main
156 ;*****
157 ;SUBROUTINES
158 ;*****
159 pwm_control_pot:
160 readadc a.0,lectura
161 call moviment
162 ; PWM m?xim:499 m?nim motor experimental:199
163 symbol minim=99
164 symbol maxim=499
165 symbol span=w19
166 ;let lectura=230
167 if lectura<116 then
168 let w18=maxim-minim
169 let w19=lectura*w18
170 let w20=w19/100
171 let w21=maxim-w20
172 endif
173
174 if lectura>140 then
175 let w18=maxim-minim
176 let lectura=255-lectura
177 let w19=lectura*w18
178 let w20=w19/100
179 let w21=maxim-w20
180 endif
181
182 ;pwmout pwmdiv16, Enable_Mdreta, 124, w21 ; 1000Hz at 90% @ 8MHz
183 ;pwmout pwmdiv16, Enable_Mesquerra, 124, w21 ; 1000Hz at 90% @ 8MHz
184 return
185 moviment:
186 select case lectura
187 case <116
188 high Positiu_Mdreta,Positiu_Mesquerra
189 low Negatiu_Mdreta,Negatiu_Mesquerra
190 pwmout pwmdiv16, Enable_Mdreta, 124, w21 ; 1000Hz at 90% @ 8MHz
191 pwmout pwmdiv16, Enable_Mesquerra, 124, w21 ; 1000Hz at 90% @ 8MHz
192 case >141
193 high Negatiu_Mdreta,Negatiu_Mesquerra
194 low Positiu_Mdreta,Positiu_Mesquerra
195 pwmout pwmdiv16, Enable_Mdreta, 124, w21 ; 1000Hz at 90% @ 8MHz
196 pwmout pwmdiv16, Enable_Mesquerra, 124, w21 ; 1000Hz at 90% @ 8MHz
197 else
198 call aturar_esquerra

```

```

199 call aturar_dreta
200 low Enable_Mdreta, Enable_Mesquerra
201 endselect
202 return
203
204 ;pwmout pwmdiv16, C.1, 124, 499 ; 1000Hz at 100% @ 8MHz
205 ;pwmout pwmdiv16, C.1, 124, 99 ; 1000Hz at 20% @ 8MHz
206 ;pwmout pwmdiv16, C.2, 124, 149 ; 1000Hz at 30% @ 8MHz
207
208 ;pwmout pwmdiv16, C.2, 124, 199 ; 1000Hz at 40% @ 8MHz
209 ;pwmout pwmdiv16, C.2, 124, 249 ; 1000Hz at 50% @ 8MHz
210 ;pwmout pwmdiv16, C.2, 124, 299 ; 1000Hz at 60% @ 8MHz
211 ;pwmout pwmdiv16, C.2, 124, 349 ; 1000Hz at 70% @ 8MHz
212 ;pwmout pwmdiv16, C.2, 124, 399 ; 1000Hz at 80% @ 8MHz
213 ;pwmout pwmdiv16, C.2, 124, 449 ; 1000Hz at 90% @ 8MHz
214 ;pwmout pwmdiv16, C.2, 124, 499 ; 1000Hz at 100% @ 8MHz
215
216
217
218
219
220 return
221
222
223
224
225
226 ;*****
227 control_pwm_a:
228 pwmout pwmdiv16, Enable_Mdreta, 124, 499 ; 1000Hz at 100% @ 8MHz
229 pwmout pwmdiv16, Enable_Mesquerra, 124, 499 ; 1000Hz at 100% @ 8MHz
230 return
231 ;*****
232 ;SECUENCIA D'INTERRUPCIO
233 ;*****
234 interrupt: ; el sensor ha rebut algo
235 LET pinsc=%00000000 ; deixa el display en blanc
236 irin [1,lloc_timeout],detector_IR,codi_comand ; llegir el sensor IR
237 call codi_comandament ; convertir el codi llegit
238 lloc_timeout:
239 let AA=0
240 setint %00000000,%00000001 ; reactivar interrupcio
241 return
242 ;*****
243 ;SUBROUTINES
244 ;*****
245 test_comandament:
246 ;setint %00000000,%00000001
247 pause 10
248 if AA>100 then ; DESPR?S DE PREMIER UN BOTO DEL COMANDAMENT
249 call restituir ; DURANT 100 VEGADES*10ms= 1SEGON fa el
250 else inc AA ; mateix moviment
251 endif
252 debug codi_comandament
253 return
254 restituir:
255 call aturar
256 let AA=0
257 return
258 codi_comandament:
259 select case codi_comand
260 case codi_davant
261 call davant
262 case codi_enrere
263 call enrere
264 case codi_dreta

```

```

265 call dreta
266 case codi_esquerra
267 call esquerra
268 case codi_aturar
269 call aturar
270 case codi_arc_esquerra
271 call arc_esquerra
272 case codi_arc_dreta
273 call arc_dreta
274 endselect
275 return
276 llegir_esquerra:
277 let k=n_conversions_adc ; defineix el bucle
278 let adc_acum=0 ; esborra acumulador
279 llegir_esquerra_:
280 low led_esquerra ; apaga LED
281 pause temps_adc
282 readadc sensor_esquerra,left_dark ; llegeix dark
283 high led_esquerra ; encen LED
284 pause temps_adc
285 readadc sensor_esquerra,left_light ; llegeix light
286 pause temps_adc
287 low led_esquerra ; apaga LED
288 if left_light>left_dark then
289     let left_sensor=left_light-left_dark
290 else
291     let left_sensor=0
292 endif
293 let left_sensor_=0 ; MSB=0
294 let adc_acum=adc_acum+left_sensor ; acumula LSB
295 dec k
296 if k<>0 then goto llegir_esquerra_
297 let left_sensor=adc_acum/n_conversions_adc ; comput final
298 end_llegir_esquerra:
299 put @i2c_optoA,left_sensor ; Desa valor a scrтчpad I2C
300 let optoA=left_sensor
301 return
302 llegir_dreta:
303 let k=n_conversions_adc ; defineix el bucle
304 let adc_acum=0 ; esborra acumulador
305 llegir_dreta_:
306 low led_dreta ; apaga LED
307 pause temps_adc
308 readadc sensor_dreta,right_dark ; llegeix fosc
309 high led_dreta ; ENCEN LED
310 pause temps_adc
311 readadc sensor_dreta,right_light ; llegeix llum
312 pause temps_adc
313 low led_dreta ; apaga LED
314 if right_light>right_dark then
315     ; llun superior a fosc
316     let right_sensor=right_light-right_dark
317 else
318     let right_sensor=0
319 endif
320 let right_sensor_=0 ; MSB=0
321 let adc_acum=adc_acum+right_sensor ; acumula LSB
322 dec k
323 if k<>0 then goto llegir_dreta_
324 let right_sensor=adc_acum/n_conversions_adc ; comput final
325 ;let @i2c_opto_left=0
326 let optoB=right_sensor
327 put @i2c_optoB,right_sensor
328 ;poke right_sensor,optoB
329
330 ;debug

```

```

331 ;pause 100
332 return
333 ;*****
334 ;TEST DISPOSITIUS
335 ;*****
336 test_detectors:
337 ;setint %00000000,%00000000
338 call test_esquerra
339 call test_dreta
340 return
341 test_dreta:
342 call llegir_dreta
343 if right_sensor>llindar_dreta then call davant_dreta
344 if right_sensor=<llindar_dreta then call aturar_dreta
345 return
346 test_esquerra:
347 call llegir_esquerra
348 if left_sensor>llindar_esquerra then call davant_esquerra
349 if left_sensor=<llindar_esquerra then call aturar_esquerra
350 return
351 test_leds_sensors:
352 high led_3
353 pause 200
354 low led_3
355 high led_2
356 pause 200
357 low led_2
358 high led_1
359 pause 200
360 low led_1
361 high led_0
362 pause 200
363 low led_0
364 high led_5
365 pause 200
366 low led_5
367 return
368 test_motors:
369 if baixar=1 then goto test_motors_
370 RETURN
371 test_motors_:
372 call davant
373 pause 3000
374 call enrere
375 pause 3000
376 call dreta
377 pause 3000
378 call esquerra
379 pause 1000
380 call aturar
381 pause 5000
382 return
383 test_botons:
384 if pujar=1 then call davant_dreta
385 if pujar=0 then call aturar_dreta
386 if baixar=1 then call davant_esquerra
387 if baixar=0 then call aturar_esquerra
388 if pujar=1 then high led_dreta endif
389 if pujar=0 then low led_dreta endif
390 if baixar=1 then high led_esquerra endif
391 if baixar=0 then low led_esquerra endif
392 return
393 return
394 test_encoders:
395 ;setintflags %00000000,%00000000 ; Activa interrupcions I2C
396 call codificar_portd

```

```

397
398 if pinC.6=1 then
399     let i2cbus = i2cbus and %11110000
400     if i2cbus>=128 then
401         let i2cbus=i2cbus-128
402         high B.3
403         else low B.3
404     endif
405     if i2cbus>=64 then
406         let i2cbus=i2cbus-64
407         high B.2
408         else low B.2
409     endif
410     if i2cbus>=32 then
411         let i2cbus=i2cbus-32
412         high B.1
413         else low B.1
414     endif
415     if i2cbus>=16 then
416         let i2cbus=i2cbus-16
417         high B.0
418         else low B.0
419     endif
420 endif
421 if pinC.6=0 then
422     let i2cbus = i2cbus and %00001111
423     if i2cbus>=8 then
424         let i2cbus=i2cbus-8
425         high B.3
426         else low B.3
427     endif
428     if i2cbus>=4 then
429         let i2cbus=i2cbus-4
430         high B.2
431         else low B.2
432     endif
433     if i2cbus>=2 then
434         let i2cbus=i2cbus-2
435         high B.1
436         else low B.1
437     endif
438     if i2cbus>=1 then
439         let i2cbus=i2cbus-1
440         high B.0
441         else low B.0
442     endif
443 endif
444 return
445 codificar_portd:
446 let BB=0
447 let dirsd=%00000000; 0 entrada 1 sortida
448     if pind.0=1 then let BB=BB+2 endif
449     if pind.1=1 then let BB=BB+1 endif
450     if pind.2=1 then let BB=BB+8 endif
451     if pind.3=1 then let BB=BB+4 endif
452     if pind.4=1 then let BB=BB+32 endif
453     if pind.5=1 then let BB=BB+16 endif
454     if pind.6=1 then let BB=BB+128 endif
455     if pind.7=1 then let BB=BB+64 endif
456 LET i2cbus=BB
457 return
458 ;*****
459 ;SUBROUTINES DE MOVIMENT
460 ;*****
461 soft_davant:
462 call davant

```

```

463 pause 100
464 call aturar_de_davant
465 return
466 soft_arc_esquerra:
467 call esquerra
468 call arc_esquerra
469 pause 50
470 call aturar_de_dreta
471 return
472 soft_arc_dreta:
473 call dreta
474 call arc_dreta
475 pause 50
476 call aturar_de_esquerra
477 return
478 davant:
479 low Enable_Mesquerra,Enable_Mdreta
480 high Positiu_Mdreta,Positiu_Mesquerra
481 low Negatiu_Mdreta,Negatiu_Mesquerra
482 high Enable_Mesquerra,Enable_Mdreta
483 return
484 enrere:
485 low Enable_Mesquerra,Enable_Mdreta
486 low Positiu_Mdreta,Positiu_Mesquerra
487 high Negatiu_Mdreta,Negatiu_Mesquerra
488 high Enable_Mesquerra,Enable_Mdreta
489 return
490 dreta:
491 low Enable_Mesquerra,Enable_Mdreta
492 low Positiu_Mdreta,Negatiu_Mesquerra
493 high Negatiu_Mdreta,Positiu_Mesquerra
494 high Enable_Mesquerra,Enable_Mdreta
495 return
496 arc_dreta:
497 low Enable_Mesquerra,Enable_Mdreta
498 low Positiu_Mdreta,Negatiu_Mesquerra,Negatiu_Mdreta,Positiu_Mesquerra
499 high Positiu_Mesquerra
500 high Enable_Mesquerra,Enable_Mdreta
501 return
502 esquerra:
503 low Enable_Mesquerra,Enable_Mdreta
504 low Negatiu_Mdreta,Positiu_Mesquerra
505 high Positiu_Mdreta,Negatiu_Mesquerra
506 high Enable_Mesquerra,Enable_Mdreta
507 return
508 arc_esquerra:
509 low Enable_Mesquerra,Enable_Mdreta
510 low Negatiu_Mdreta,Positiu_Mesquerra,Positiu_Mdreta,Negatiu_Mesquerra
511 high Positiu_Mdreta
512 high Enable_Mesquerra,Enable_Mdreta
513 return
514 davant_dreta:
515 low Enable_Mesquerra,Enable_Mdreta
516 high Positiu_Mdreta
517 low Negatiu_Mdreta
518 high Enable_Mesquerra,Enable_Mdreta
519 return
520 davant_esquerra:
521 low Enable_Mesquerra,Enable_Mdreta
522 high Positiu_Mesquerra
523 low Negatiu_Mesquerra
524 high Enable_Mesquerra,Enable_Mdreta
525 return
526 enrere_dreta:
527 low Enable_Mesquerra,Enable_Mdreta
528 low Positiu_Mdreta

```



```
529 high Negatiu_Mdreta
530 high Enable_Mesquerra, Enable_Mdreta
531 return
532 enrere_esquerra:
533 low Enable_Mesquerra, Enable_Mdreta
534 low Positiu_Mesquerra
535 high Negatiu_Mesquerra
536 high Enable_Mesquerra, Enable_Mdreta
537 return
538 aturar:
539 low Enable_Mesquerra, Enable_Mdreta
540 low Positiu_Mdreta, Negatiu_Mdreta
541 high Negatiu_Mesquerra, Positiu_Mesquerra
542 return
543 aturar_dreta:
544 low Enable_Mesquerra, Enable_Mdreta
545 high Positiu_Mdreta, Negatiu_Mdreta
546 return
547 aturar_esquerra:
548 low Enable_Mesquerra, Enable_Mdreta
549 high Positiu_Mesquerra, Negatiu_Mesquerra
550 return
551 aturar_de_esquerra:
552 low Enable_Mesquerra, Enable_Mdreta
553 low Negatiu_Mesquerra
554 return
555 aturar_de_dreta:
556 low Enable_Mesquerra, Enable_Mdreta
557 low Negatiu_Mdreta
558 return
559 aturar_de_davant:
560 low Enable_Mesquerra, Enable_Mdreta
561 low Negatiu_Mdreta, Negatiu_Mesquerra
562 return
563 aturar_de_enrere:
564 low Enable_Mesquerra, Enable_Mdreta
565 low Positiu_Mdreta, Positiu_Mesquerra
566 return
567
```