

```
1 ; slave
2 ; CONFIGURACIONES
3 #picaxe 40x2 ' PICAXE-28X2-5V (18F25K22)
4 setfreq m8
5 ;*****
6 ;*** Configuracio comandament infrarroig ***
7 ;*****
8 symbol codi_davant=$0D
9 symbol codi_enrere=$16
10 symbol codi_dreta=$10
11 symbol codi_esquerra=$11
12 symbol codi_aturar=$15
13 symbol codi_arc_esquerra=$12
14 symbol codi_arc_dreta=$13
15 symbol detector_IR=C.0
16 ;*****
17 ;*** Configuracio sensors 40X2 ***
18 ;*****
19 symbol sensor_dreta=3
20 symbol sensor_esquerra=2
21 symbol sensor_0=A.0
22 symbol sensor_1=A.1
23 symbol sensor_2=A.2
24 symbol sensor_3=A.3
25 symbol led_3=B.3
26 symbol led_2=B.2
27 symbol led_1=B.1
28 symbol led_0=B.0
29 symbol led_dreta=B.3
30 symbol led_esquerra=B.2
31 symbol n_conversions_adc=4
32 ;*****
33 ;*** Configuracio botons 40X2 ***
34 ;*****
35 symbol pujar = pinC.6
36 symbol baixar = pinC.7
37 ;*****
38 ;*** Configuracio bridge motors 40x2 ***
39 ;*****
40 symbol MA_positiu = B.5
41 symbol MA_negatiu = B.4
42 symbol MA_pwm = C.2
43 symbol MB_positiu = B.7
44 symbol MB_negatiu = B.6
45 symbol MB_pwm = C.1
46 symbol MC_positiu = A.5
47 symbol MC_negatiu = A.6
48 symbol MC_pwm = C.2
49 symbol MD_positiu = A.7
50 symbol MD_negatiu = C.5
51 symbol MD_pwm = C.1
52 high MA_pwm
53 high MB_pwm
54 high MC_pwm
55 high MD_pwm
56 ;*****
```

C:\Users\Miner1\Desktop\40x2v31_test_slave.bas

```
57 ;*** Definicio de variables ***
58 ;*****
59 symbol codi_comand=b0 ; codi premut del comandament IR
60 symbol taula_veritat=b1 ; Combinacio decimal taula veritat de P
rastrejador
61 symbol left_dark=b2 ; Lectura sensor esquerra llum ambient
62 symbol left_light=b3 ; Lectura sensor esquerra reflexio
63 symbol left_sensor=b4 ; Lectura esquerra
64 symbol left_sensor_=b5 ;
65 symbol right_dark=b6 ; Lectura sensor dreta llum ambient
66 symbol right_light=b7 ; Lectura sensor dreta reflexio
67 symbol right_sensor=b8; Lectura dreta
68 symbol right_sensor_=b9 ; Lectura dreta
69 symbol i2cbus=b10 ;
70 symbol shadow=b11 ; Per interrupcions
71 symbol AA=b12 ; index bucle de test comandament
72 symbol BB=b13 ; index bucle de test comandament
73 symbol comande=20
74 symbol codi=b19
75 symbol codi_=b20
76 symbol optoA=b21
77 symbol optoB=b22
78 symbol PWMA=b25
79 symbol PWMB=b26
80 symbol aleatori=b27
81 symbol done=b28
82 symbol done_tim=b29
83 symbol graf=b30
84 symbol lim_compte=b31
85 symbol comptador=b32
86
87 symbol k=b40
88 symbol adc_acum=w23
89 ;*****
90 ;DEFINICIO DE CONSTANTS
91 ;*****
92 symbol temps_adc=10
93 symbol llindar_dreta = 4
94 symbol llindar_esquerra = 4
95 ;*****
96 ;CONFIGURACIO BUS I2C
97 ;*****
98 ;let i2cbus=$E8
99 call codificar_portd
100 hi2csetup i2cslave,i2cbus
101 pause 1000 ; imprescindible no quantificat
102 ;*****
103 ;*** Scratchpad I2C ***
104 ;*****
105 symbol @i2c_com=20
106 symbol @i2c_optoA=21
107 symbol @i2c_opto_left=21
108 symbol @i2c_optoB=22
109 symbol @i2c_opto_right=22
110 symbol @i2c_PWMA=25
111 symbol @i2c_PWMB=26
```

C:\Users\Miner1\Desktop\40x2v31_test_slave.bas

```
112 symbol @i2c_random=27
113 symbol @i2c_done=28
114 symbol @i2c_done_tim=29
115 symbol @i2c_graf=30
116 ;*****
117 ;*** Codi control cinta ***
118 ;*****
119 symbol cin_aturar=30 ; cinta aturada
120 symbol cin_davant=31 ; cinta marxa davant
121 symbol cin_enrere=32 ; cinta marxa enrere
122 symbol cin_A_left=40 ; cinta marxa davant fins left
123 symbol cin_A_right=41 ; cinta marxa enrere fins right
124 symbol cin_A_left_tim=42 ; cinta marxa enrere fins right + tim
125 symbol cin_A_right_tim=43 ; cinta marxa enrere fins right + tim
126 symbol cin_reset=49 ; cinta marxa enrere fins right + tim
127 ;*****
128 ;PROGRAMA PRINCIPAL INICIAL
129 ;*****
130 call aturar ; cinta aturada
131 call control_pwm: ; ACTICA PWMA
132 ;setintflags %01000000,%01000000 ; Activa interrupcions I2C
133 let done=1
134 let graf=0
135 put @i2c_done,done
136 let done_tim=1
137 put @i2c_done_tim,done_tim
138 let lim_compte=30
139 let comptador=0
140 let codi=0
141 let codi_=0
142 ;*****
143 ;PROGRAMA PRINCIPAL REITERATIU
144 ;*****
145 ;setint %00000000,%00000001; activar aquesta l?nia en test comandament
146 main;; nomes activar un test. Provar-los individualment.
147 ;call test_motors ; 40x2 endavant,enrere; dreta, esquerra.
148 call test_detectors ; 40x2 NO DETECTA: enrere, DETECTA: davant.
149 ;call test_botons ; 40x2 BAIXAR:endavant esqu PUIJAR: davant ↻
dreta
150 ;call test_leds_sensors ; 40x2 ESQUERRA: rapid, DRETA: lent
151 ;call test_encoders ; 40x2 les sortides en leds B.3, B.2, B.1, B.0
152 ;call test_comandament ; 40x2 activar interrupcions.
153 goto main
154 ;*****
155 control_pwm:
156 pwmout pwmdiv16, MA_pwm, 124, 499 ; 1000Hz at 100% @ 8MHz
157 pwmout pwmdiv16, MB_pwm, 124, 499 ; 1000Hz at 100% @ 8MHz
158 pwmout pwmdiv16, MC_pwm, 124, 499 ; 1000Hz at 100% @ 8MHz
159 pwmout pwmdiv16, MD_pwm, 124, 499 ; 1000Hz at 100% @ 8MHz
160 return
161 ;*****
162 ;SECUENCIA D'INTERRUPCIO
163 ;*****
164 interrupt: ; el sensor ha rebut ↻
algo
165 LET pinsc=%00000000 ; deixa el display en ↻
```

C:\Users\Miner1\Desktop\40x2v31_test_slave.bas

```
blanc
166 irin [1,lloc_timeout],detector_IR,codi_comand ; llegir el sensor IR
167 call codi_comandament ; convertir el codi ↻
    llegit
168 lloc_timeout:
169 let AA=0
170 setint %00000000,%0000001 ; reactivar interrupcio
171 return
-----
172 ;*****
173 ;SUBROUTINES
174 ;*****
175 test_comandament:
176 ;setint %00000000,%00000001
177 pause 10
178 if AA>100 then ; DESPR?S DE PREMIER UN BOTO DEL ↻
    COMANDAMENT
179     call restituir ; DURANT 100 VEGADES*10ms= 1SEGON ↻
    fa el
180     else inc AA ; mateix moviment
181 endif
182 debug codi_comandament
183 return
-----
184 restituir:
185 call aturar
186 let AA=0
187 return
-----
188 codi_comandament:
189 select case codi_comand
190 case codi_davant
191 call davant
192 case codi_enrere
193 call enrere
194 case codi_dreta
195 call dreta
196 case codi_esquerra
197 call esquerra
198 case codi_aturar
199 call aturar
200 case codi_arc_esquerra
201 call arc_esquerra
202 case codi_arc_dreta
203 call arc_dreta
204 endselect
205 return
-----
206 llegir_esquerra:
207 let k=n_conversions_adc ; defineix el bucle
208 let adc_acum=0 ; esborra acumulador
209 llegir_esquerra_:
210 low led_esquerra ; apaga LED
211 pause temps_adc
212 readadc sensor_esquerra,left_dark ; llegeix dark
213 high led_esquerra ; encen LED
214 pause temps_adc
215 readadc sensor_esquerra,left_light ; llegeix light
216 pause temps_adc
217 low led_esquerra ; apaga LED
```

```

C:\Users\Miner1\Desktop\40x2v31_test_slave.bas
218 if left_light>left_dark then
219     |         let left_sensor=left_light-left_dark
220     |         else
221     |         let left_sensor=0
222 endif
223 let left_sensor_=0           ; MSB=0
224 let adc_acum=adc_acum+left_sensor ; acumula LSB
225 dec k
226 if k<>0 then goto llegir_esquerra_
227 let left_sensor=adc_acum/n_conversions_adc ; comput final
228 end_llegir_esquerra:
229 put @i2c_optoA,left_sensor           ; Desa valor a P
scrtpad I2C
230 let optoA=left_sensor
231 return
-----
232 llegir_dreta:
233 let k=n_conversions_adc           ; defineix el bucle
234 let adc_acum=0                   ; esborra acumulador
235 llegir_dreta_:
236 low led_dreta                   ; apaga LED
237 pause temps_adc
238 readadc sensor_dreta,right_dark   ; llegeix foscor
239 high led_dreta                   ; ENCEN LED
240 pause temps_adc
241 readadc sensor_dreta,right_light   ; llegeix llum
242 pause temps_adc
243 low led_dreta                   ; apaga LED
244 if right_light>right_dark then
245     |         ; llun superior a foscor
246     |         let right_sensor=right_light-right_dark
247     |         else
248     |         let right_sensor=0
249 endif
250 let right_sensor_=0           ; MSB=0
251 let adc_acum=adc_acum+right_sensor ; acumula LSB
252 dec k
253 if k<>0 then goto llegir_dreta_
254 let right_sensor=adc_acum/n_conversions_adc ; comput final
255 ;let @i2c_opto_left=0
256 let optoB=right_sensor
257 put @i2c_optoB,right_sensor
258 ;poke right_sensor,optoB
259
260 ;debug
261 ;pause 100
262 return
-----
263 ;*****
264 ;TEST DISPOSITIUS
265 ;*****
266 test_detectors:
267 ;setint %00000000,%00000000
268 call test_esquerra
269 call test_dreta
270 return
-----
271 test_dreta:
272 call llegir_dreta

```

```

C:\Users\Miner1\Desktop\40x2v31_test_slave.bas
273  if right_sensor>llindar_dreta then call davant_dreta
274  if right_sensor=<llindar_dreta then call aturar_dreta
275  return
-----
276  test_esquerra:
277  call llegir_esquerra
278  if left_sensor>llindar_esquerra then call davant_esquerra
279  if left_sensor=<llindar_esquerra then call aturar_esquerra
280  return
-----
281  test_leds_sensors:
282  high led_3
283  pause 200
284  low led_3
285  high led_2
286  pause 200
287  low led_2
288  high led_1
289  pause 200
290  low led_1
291  high led_0
292  pause 200
293  low led_0
294  return
-----
295  test_motors:
296  if baixar=1 then goto test_motors_
297  RETURN
-----
298
299  test_motors_:
300  TESTA:
301  call on_A
302  pause 3000
303  call off_A
304  pause 1000
305  call reon_A
306  pause 3000
307  call off_A
308  pause 1000
309
310  TESTB:
311  call on_B
312  pause 3000
313  call off_B
314  pause 1000
315  call reon_B
316  pause 3000
317  call off_B
318  pause 1000
319
320  TESTC:
321  call on_C
322  pause 3000
323  call off_C
324  pause 1000
325  call reon_C
326  pause 3000
327  call off_C
328  pause 1000

```

```
329
330 TESTD:
331 call on_D
332 pause 3000
333 call off_D
334 pause 1000
335 call reon_D
336 pause 3000
337 call off_D
338 pause 1000
339 return


---


340 test_botons:
341 if pujar=1 then call davant_dreta
342 if pujar=0 then call aturar_dreta
343 if baixar=1 then call davant_esquerra
344 if baixar=0 then call aturar_esquerra
345 if pujar=1 then high led_dreta endif
346 if pujar=0 then low led_dreta endif
347 if baixar=1 then high led_esquerra endif
348 if baixar=0 then low led_esquerra endif
349 return


---


350 return


---


351 test_encoders:
352 ;setintflags %00000000,%00000000 ; Activa interrupcions I2C
353 call codificar_portd
354
355 if pinC.6=1 then
356     let i2cbus = i2cbus and %11110000
357     if i2cbus>=128 then
358         let i2cbus=i2cbus-128
359         high B.3
360         else low B.3
361     endif
362     if i2cbus>=64 then
363         let i2cbus=i2cbus-64
364         high B.2
365         else low B.2
366     endif
367     if i2cbus>=32 then
368         let i2cbus=i2cbus-32
369         high B.1
370         else low B.1
371     endif
372     if i2cbus>=16 then
373         let i2cbus=i2cbus-16
374         high B.0
375         else low B.0
376     endif
377 endif
378 if pinC.6=0 then
379     let i2cbus = i2cbus and %00001111
380     if i2cbus>=8 then
381         let i2cbus=i2cbus-8
382         high B.3
383         else low B.3
384     endif
```

C:\Users\Miner1\Desktop\40x2v31_test_slave.bas

```
385         if i2cbus>=4 then
386             let i2cbus=i2cbus-4
387             high B.2
388             else low B.2
389         endif
390         if i2cbus>=2 then
391             let i2cbus=i2cbus-2
392             high B.1
393             else low B.1
394         endif
395         if i2cbus>=1 then
396             let i2cbus=i2cbus-1
397             high B.0
398             else low B.0
399         endif
400     endif
401     return


---


402     codificar_portd:
403     let BB=0
404     let dirsd=%00000000; 0 entrada 1 sortida
405         if pind.0=1 then let BB=BB+2 endif
406         if pind.1=1 then let BB=BB+1 endif
407         if pind.2=1 then let BB=BB+8 endif
408         if pind.3=1 then let BB=BB+4 endif
409         if pind.4=1 then let BB=BB+32 endif
410         if pind.5=1 then let BB=BB+16 endif
411         if pind.6=1 then let BB=BB+128 endif
412         if pind.7=1 then let BB=BB+64 endif
413     LET i2cbus=BB
414     return


---


415     ;*****
416     ;SUBROUTINES DE MOVIMENT
417     ;*****
418     soft_davant:
419     call davant
420     pause 100
421     call aturar_de_davant
422     return


---


423     soft_arc_esquerra:
424     call esquerra
425     call arc_esquerra
426     pause 50
427     call aturar_de_dreta
428     return


---


429     soft_arc_dreta:
430     call dreta
431     call arc_dreta
432     pause 50
433     call aturar_de_esquerra
434     return


---


435     davant:
436     low MB_pwm,MA_pwm
437     high MA_positiu,MB_positiu
438     low MA_negatiu,MB_negatiu
439     high MB_pwm,MA_pwm
440     return


---


```



```
441 enrere:
442 low MB_pwm,MA_pwm
443 low MA_positiu,MB_positiu
444 high MA_negatiu,MB_negatiu
445 high MB_pwm,MA_pwm
446 return
-----
447 dreta:
448 low MB_pwm,MA_pwm
449 low MA_positiu,MB_negatiu
450 high MA_negatiu,MB_positiu
451 high MB_pwm,MA_pwm
452 return
-----
453 arc_dreta:
454 low MB_pwm,MA_pwm
455 low MA_positiu,MB_negatiu,MA_negatiu,MB_positiu
456 high MB_positiu
457 high MB_pwm,MA_pwm
458 return
-----
459 esquerra:
460 low MB_pwm,MA_pwm
461 low MA_negatiu,MB_positiu
462 high MA_positiu,MB_negatiu
463 high MB_pwm,MA_pwm
464 return
-----
465 arc_esquerra:
466 low MB_pwm,MA_pwm
467 low MA_negatiu,MB_positiu,MA_positiu,MB_negatiu
468 high MA_positiu
469 high MB_pwm,MA_pwm
470 return
-----
471 davant_dreta:
472 low MB_pwm,MA_pwm
473 high MA_positiu
474 low MA_negatiu
475 high MB_pwm,MA_pwm
476 return
-----
477 davant_esquerra:
478 low MB_pwm,MA_pwm
479 high MB_positiu
480 low MB_negatiu
481 high MB_pwm,MA_pwm
482 return
-----
483 enrere_dreta:
484 low MB_pwm,MA_pwm
485 low MA_positiu
486 high MA_negatiu
487 high MB_pwm,MA_pwm
488 return
-----
489 enrere_esquerra:
490 low MB_pwm,MA_pwm
491 low MB_positiu
492 high MB_negatiu
493 high MB_pwm,MA_pwm
494 return
-----
495 aturar:
496 low MB_pwm,MA_pwm
```

```
497 low MA_positiu,MA_negatiu
498 high MB_negatiu,MB_positiu
499 return
-----
500 aturar_dreta:
501 low MB_pwm,MA_pwm
502 high MA_positiu,MA_negatiu
503 return
-----
504 aturar_esquerra:
505 low MB_pwm,MA_pwm
506 high MB_positiu,MB_negatiu
507 return
-----
508 aturar_de_esquerra:
509 low MB_pwm,MA_pwm
510 low MB_negatiu
511 return
-----
512 aturar_de_dreta:
513 low MB_pwm,MA_pwm
514 low MA_negatiu
515 return
-----
516 aturar_de_davant:
517 low MB_pwm,MA_pwm
518 low MA_negatiu, MB_negatiu
519 return
-----
520 aturar_de_enrere:
521 low MB_pwm,MA_pwm
522 low MA_positiu, MB_positiu
523 return
-----
524 on_A:
525 low MA_pwm
526 high MA_positiu
527 low MA_negatiu
528 high MA_pwm
529 return
-----
530 reon_A:
531 low MA_pwm
532 low MA_positiu
533 high MA_negatiu
534 high MA_pwm
535 return
-----
536 off_A:
537 low MA_pwm
538 low MA_positiu
539 low MA_negatiu
540 high MA_pwm
541 return
-----
542 on_B:
543 low MB_pwm
544 high MB_positiu
545 low MB_negatiu
546 high MB_pwm
547 return
-----
548 reon_B:
549 low MB_pwm
550 low MB_positiu
551 high MB_negatiu
552 high MB_pwm
```

```
553 return
554 off_B:
555 low MB_pwm
556 low MB_positiu
557 low MB_negatiu
558 high MB_pwm
559 return
560 on_C:
561 low MC_pwm
562 high MC_positiu
563 low MC_negatiu
564 high MC_pwm
565 return
566 reon_C:
567 low MC_pwm
568 low MC_positiu
569 high MC_negatiu
570 high MC_pwm
571 return
572 off_C:
573 low MC_pwm
574 low MC_positiu
575 low MC_negatiu
576 high MC_pwm
577 return
578 on_D:
579 low MD_pwm
580 high MD_positiu
581 low MD_negatiu
582 high MD_pwm
583 return
584 reon_D:
585 low MD_pwm
586 low MD_positiu
587 high MD_negatiu
588 high MD_pwm
589 return
590 off_D:
591 low MD_pwm
592 low MD_positiu
593 low MD_negatiu
594 high MD_pwm
595 return
596
597
598
599
600
601
602
603
604
605
```