

```

1 #include "mbed.h"
2 #define MAXPOS 50          // en milímetros
3 #define SS_TIME 100       // en microsegundos
4 #include "Serial.h"        //Comunicacion
5 #define MEDIO_G 2440
6 #define AVANCE_1 1600
7 #define AVANCE_2 3200
8 #define AVANCE_3 4800
9 Serial pc(USBTX, USBRX);
10
11
12
13 PwmOut myServoX(D5);
14 PwmOut myServoY(D6);
15 PwmOut myServoZ(D3);
16 Serial pc(USBTX, USBRX);
17
18 void draw();
19 void nodraw();
20 void Dibujar();
21 void Guardar();
22 uint32_t read_command();
23 uint8_t mem_put(uint32_t data);
24 uint8_t mem_get(uint32_t* data);
25 void tail_reset();
26 void mem_free();
27 void vertex2d(uint8_t x, uint8_t y);
28 void traslado(int x,int y);
29 uint32_t speed = 0;
30

```

Se incluye la librería MBED

Se define el No. De pasos para cada giro (medio giro).  
**(AVANCE 1 – 2 – 3)**

Se define comunicación serial.

Se define el PIN de salida para los (3) tres servomotores

Se definen cada una de las funciones del programa.

```

33
34 uint32_t buffer[10];
35
36 int mem_head = 0;
37 int mem_tail = 0;
38 uint8_t full = 0;
39
40
41 class stepmotor
42 {
43 public:
44
45     stepmotor(PinName in1, PinName in2, PinName in3, PinName in4, PinName in5, PinName in6, PinName in7, PinName in8);
46     void step(uint32_t num_steps, bool cw);
47     void step_h(uint32_t num_steps, bool cw);
48     void step_ah(uint32_t num_steps, bool cw);
49     void set_speed(int speed);
50     uint32_t get_speed();
51     // BusOut motor_out;
52 private:
53     BusOut motor_out;
54     uint32_t motorSpeed;
55     uint32_t nstep;
56
57     void move();
58     void move_h();
59     void move_ah();
60 };
61
62
63 //endif
64
65 stepmotor::stepmotor(PinName in1, PinName in2, PinName in3, PinName in4, PinName in5, PinName in6, PinName in7, PinName in8) : motor_out(in1,in2,in3,in
66 {
67
68     motor_out=0x0;
69     nstep=0;
70     motorSpeed=1100;
71
72 }
73
74
75 void stepmotor::move()
76 {
77     switch(nstep)
78     {
79         case 0: motor_out = 0x1; break; // 0000 0001
80         case 1: motor_out = 0x10; break; // 0001 0000
81         case 2: motor_out = 0x3; break; // 0000 0011
82         case 3: motor_out = 0x30; break; // 0011 0000
83         case 4: motor_out = 0x2; break; // 0000 0010
84         case 5: motor_out = 0x20; break; // 0010 0000
85         case 6: motor_out = 0x6; break; // 0000 0110
86
87         case 6: motor_out = 0x6; break; // 0000 0110
88         case 7: motor_out = 0x60; break; // 0110 0000
89         case 8: motor_out = 0x4; break; // 0000 0100
90         case 9: motor_out = 0x40; break; // 0100 0000
91         case 10: motor_out = 0xC; break; // 0000 1100
92         case 11: motor_out = 0xC0; break; // 1100 0000
93         case 12: motor_out = 0x8; break; // 0000 1000
94         case 13: motor_out = 0x80; break; // 1000 0000
95         case 14: motor_out = 0x9; break; // 0000 1001
96         case 15: motor_out = 0x90; break; // 1001 0000
97         default: motor_out = 0x00; break; // 0000
98     }
99     wait_us(motorSpeed);
100     motor_out = 0x00;
101 }
102
103 void stepmotor::move_h()
104 {
105     switch(nstep)
106     {
107         case 0: motor_out = 0x1; break; // 0000 0001
108         case 1: motor_out = 0x90; break; // 1001 0000
109

```

Función para mover los (2) dos motores pasó a paso hacia adelante

```

113 {
114
115     switch(nstep)
116     {
117         case 0: motor_out = 0x1; break; // 0000 0001
118         case 1: motor_out = 0x90; break; // 1001 0000
119
120         case 2: motor_out = 0x3; break; // 0000 0011
121         case 3: motor_out = 0x80; break; // 0011 0000
122
123         case 4: motor_out = 0x2; break; // 0000 0010
124         case 5: motor_out = 0xC0; break; // 0010 0000
125
126         case 6: motor_out = 0x6; break; // 0000 0110
127         case 7: motor_out = 0x40; break; // 0110 0000
128
129         case 8: motor_out = 0x4; break; // 0000 0100
130         case 9: motor_out = 0x60; break; // 0100 0000
131
132         case 10: motor_out = 0xC; break; // 0000 1100
133         case 11: motor_out = 0x20; break; // 1100 0000
134
135         case 12: motor_out = 0x8; break; // 0000 1000
136         case 13: motor_out = 0x30; break; // 1000 0000
137
138         case 14: motor_out = 0x9; break; // 0000 1001
139         case 15: motor_out = 0x10; break; // 1001 0000
140
141         default: motor_out = 0x00; break; // 0000
142     }
143 }
144
145 void stepmotor::move_ah()
146 {
147     switch(nstep)
148     {
149         case 0: motor_out = 0x9; break; // 0000 0001
150         case 1: motor_out = 0x10; break; // 0001 0000
151
152         case 2: motor_out = 0x8; break; // 0000 0011
153         case 3: motor_out = 0x30; break; // 0011 0000
154
155         case 4: motor_out = 0xC; break; // 0000 0010
156         case 5: motor_out = 0x20; break; // 0010 0000
157
158         case 6: motor_out = 0x4; break; // 0000 0110
159         case 7: motor_out = 0x60; break; // 0110 0000
160
161         case 8: motor_out = 0x6; break; // 0000 0100
162         case 9: motor_out = 0x40; break; // 0100 0000
163
164         case 10: motor_out = 0x2; break; // 0000 1100
165         case 11: motor_out = 0xC0; break; // 1100 0000
166
167         case 12: motor_out = 0x3; break; // 0000 1000
168         case 13: motor_out = 0x80; break; // 1000 0000
169
170         case 14: motor_out = 0x1; break; // 0000 1001
171         case 15: motor_out = 0x90; break; // 1001 0000
172
173
174

```

Función para mover los (2) dos motores pasó a paso en sentido horario

Función para mover los (2) dos motores pasó a paso en sentido anti - horario

# Funciones de Clase Stepmotor

```

181
182
183 void Stepmotor::set_speed(int speed)
184 {
185     motorSpeed=speed; //set motor speed us
186 }
187
188
189 uint32_t Stepmotor::get_speed()
190 {
191     return motorSpeed; //
192 }
193
194
195 void Stepmotor::step(uint32_t num_steps, bool cw)
196 {
197     // funcion para mover el motor N pasos CW o CCW
198     // num_steps número de paso que da el motor
199     // cw =True para dirección en sentido del reloj
200     // cw =False para dirección contraria de las manecillas del reloj
201
202     uint32_t count=num_steps ;
203     while(count)
204     {
205         if (cw) nstep++;
206         else nstep--;
207         if (nstep>15) nstep=0;
208         if (nstep<0) nstep=15;
209         move();
210         count--;

```

```

213
214 }
215 void Stepmotor::step_h(uint32_t num_steps, bool cw)
216 {
217
218     uint32_t count=num_steps ;
219     while(count)
220     {
221         if (cw) nstep++;
222         else nstep--;
223         if (nstep>15) nstep=0;
224         if (nstep<0) nstep=15;
225         move_h();
226         count--;
227
228     }
229
230 }
231 void Stepmotor::step_ah(uint32_t num_steps, bool cw)
232 {
233
234     uint32_t count=num_steps ;
235     while(count)
236     {
237         if (cw) nstep++;
238         else nstep--;
239         if (nstep>15) nstep=0;
240         if (nstep<0) nstep=15;
241         move_ah();
242         count--;

```

```

249 {
250     myServoX.period_ms(20);
251     myServoY.period_ms(20);
252     pc.baud(9600);
253     pc.printf("inicializacion \n listo para dibujar o guardar \n");
254     uint32_t read_cc;
255     while(1)
256     {
257         read_cc=read_command();
258         switch (read_cc) {
259             case 0xFF: Dibujar(); break;
260             case 0xFE: Guardar(); break;
261             default: pc.printf("error de comando. \n se esperaba FE F0(guardar) o FF F0(dibujar) \n");break ;
262         }
263     }
264 }
265 uint32_t read_command()
266 {
267     uint32_t val=0;
268     uint8_t cnt=0;
269
270     char endc=command.getc();
271
272     while(endc != 0xF0 && cnt <4) {
273         if(endc!=0xF0)
274             val=((val<<8) +endc);
275         endc=command.getc();
276         cnt++;
277     }
278     if(endc==0xF0)
285
286     pc.printf("dibujo en proceso... \n");
287
288     uint8_t error=0;
289     uint32_t dato;
290
291     tail_reset();
292
293     while(error==0)
294     {
295         error = mem_get(&dato);
296         if (error==0)
297         {
298             switch (dato)
299             {
300                 case 0xFC:
301                     pc.printf("-> Baja Z\n");
302                     draw();
303                     break ;
304                 case 0xFB:
305                     pc.printf("-> Sube Z\n");
306                     nodraw();
307                     break ;
308
309                 default:
310                     int y      = (uint8_t) (dato);
311                     int x      = (uint8_t) (dato>>8);
312                     char ncomm  = (uint8_t) (dato>>16);
313
314                     if (ncomm == 0xFD) //vertex2d =fd

```

Función main del programa

Setea periodo de los servos en 20 mili segundos.

Setea la velocidad de comunicación serial a 9600.

Lectura de los comandos enviados mediante la comunicación

FF – DIBUJAR

FE – GUARDAR

FC – BAJAR Z

FB – SUBIR Z

F0 – FINAL PARA CADA INSTRUCCIÓN

FD – VERTEX2D

F9 – TRANSLADO DEL PICCOLO

FA – STOP DE INSTRUCCIONES

```

317         vertex2d(x,y);
318     }
319     if (ncomm == 0xF9) //Desplazamiento en cuadro
320     {
321         pc.printf("-> Trasladar piccolo fila %d y columna %d \n",x, y);
322         traslado(x,y);
323     }
324     else
325         pc.printf("-> ERROR DE COMMANDO: %d %d %d \n " ,ncomm,x,y,y);
326     break;
327 }
328 }
329 }
330
331 pc.printf("fin del dibujo..\n");
332
333 }
334 void draw()
335 {
336     myServoZ.period_ms(20);
337     myServoZ.pulsewidth_us(2000);
338     wait(1);
339 }
340 void nodraw()
341 {
342     myServoZ.period_ms(20);
343     myServoZ.pulsewidth_us(500);
344     wait(1);
345 }
346 void Guardar()

```

Función para bajar y subir Z

```

349
350     uint32_t dato=0;
351     mem_free();
352     while(dato !=0xFA)          //stop = FA
353     {
354         dato = read_command();
355         if (dato !=0xFA)
356             mem_put(dato);
357     }
358     pc.printf("fin de guardado...\n");
359 }
360 }
361 void tail_reset()
362 {
363     mem_tail=0;
364 }
365
366 void mem_free()
367 {
368     mem_head=0;
369     full=0;
370 }
371 uint8_t mem_put(uint32_t data)
372 {
373
374     if (full)
375         return 1;
376     buffer[mem_head] = data;
377     mem_head += 1;
378     if (mem_head == 10)

```

Lectura del comando enviado

```

381 }
382 uint8_t mem_get(uint32_t* data)
383 {
384     if (mem_head == 0)
385         return 1;
386     if (mem_head == mem_tail)
387         return 1;
388
389     *data = buffer[mem_tail];
390     mem_tail += 1;
391
392     return 0;
393 }
394
395 void vertex2d(uint8_t x, uint8_t y)
396 {
397     // UBICAR EL CODIGO QUE HACE FALTA
398     // la función como parametros tiene las coordenadas x y y en milímetros y
399     // debe calcular las variables pulseX y pulseY para dar paso a la actualización
400     // de la posición de los dos servomotores
401     int pulseX = x*30+500;
402     int pulseY = y*30+500;
403     myServoX.pulsewidth_us(pulseX);
404     wait_ms(1000);
405     myServoY.pulsewidth_us(pulseY);
406     wait_ms(1000);
407 }
408
409 stepmotor smotor1(D2,D4,D7,D8,D9,D10,D11,D12);

```

## Función VERTEX 2D

```

409 stepmotor smotor1(D2,D4,D7,D8,D9,D10,D11,D12);
410
411 void traslado(int x, int y) // int main()
412 {
413     uint32_t speed=1500;
414     bool cw=false;
415     switch(x)
416     {
417     case 1:
418         switch(y)
419         {
420             case 1:
421                 pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
422                 break;
423             case 2:
424                 for(int i=1; i <= 2; i = i + 1)
425                 {
426                     smotor1.set_speed(speed);
427                     smotor1.step_h(MEDIO_G,cw); // 2048
428
429                     cw=!cw;
430                     speed=speed+200;
431
432                     if (speed > 10000)
433                         speed =1500;
434                     // pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
435                     //wait(1);
436                 }
437             }
438     }

```

## Función traslado:

Se basa en 4 casos para (X) y 4 casos para (Y), los cuales combinados forman 16 posiciones en las cuales se va a ubicar el PICCOLO.

El PICCOLO siempre se va a ubicar en la posición superior izquierda para lo cual las posiciones 1.1, 2.1, 3.1 y 4.1 van hacer un desplazamiento lineal, para las siguientes posiciones dará medio giro, se traslada, se devuelve medio giro y avanza a la posición linealmente. ....

```

441     for(int i=1; i <= 2; i = i + 1)
442     {
443         smotor1.set_speed(speed);
444         smotor1.step(AVANCE_1,cw); // 1785
445         cw=!cw;
446         speed=speed+200;
447
448         if (speed > 10000)
449             speed =1500;
450         //pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x,
451         //wait(1);
452     }
453
454     for(int i=1; i <= 2; i = i + 1)
455     {
456         smotor1.set_speed(speed);
457         smotor1.step_ah(MEDIO_G,cw); //2048
458         cw=!cw;
459         speed=speed+200;
460
461         if (speed > 10000)
462             speed =1500;
463         pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
464         //wait(1);
465     }
466
467     break;
468
469 case 3:
470     for(int i=1; i <= 2; i = i + 1)

```

.... Para cada caso de (X) existe (4) cuatro casos de (Y).

Fue necesario incluir (FOR) para cada giro de los motores.

Una vez el PICCOLO llega a su posición se devuelve el texto donde se informa en qué posición se encuentra.

```

469 case 3:
470     for(int i=1; i <= 2; i = i + 1)
471     {
472         smotor1.set_speed(speed);
473         smotor1.step_h(MEDIO_G,cw); //2048
474         cw=!cw;
475         speed=speed+200;
476
477         if (speed > 10000)
478             speed =1500;
479         // pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
480         //wait(1);
481     }
482
483     for(int i=1; i <= 2; i = i + 1)
484     {
485         smotor1.set_speed(speed);
486         smotor1.step(AVANCE_2,cw); //3570
487         cw=!cw;
488         speed=speed+200;
489
490         if (speed > 10000)
491             speed =1500;
492         //pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
493         //wait(1);
494     }
495
496     for(int i=1; i <= 2; i = i + 1)
497     {
498         smotor1.set_speed(speed);

```

```

501     speed=speed+200;
502
503     if (speed > 10000)
504         speed =1500;
505     pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
506     //wait(1);
507 }
508 break;
509
510 case 4:
511     for(int i=1; i <= 2; i = i + 1)
512     {
513         smotor1.set_speed(speed);
514         smotor1.step_h(MEDIO_G,cw); //2048
515         cw=!cw;
516         speed=speed+200;
517
518         if (speed > 10000)
519             speed =1500;
520         // pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
521         //wait(1);
522     }
523
524     for(int i=1; i <= 2; i = i + 1)
525     {
526         smotor1.set_speed(speed);
527         smotor1.step(AVANCE_3,cw); //5362
528         cw=!cw;
529         speed=speed+200;
530

```



```

529     speed=speed+200;
530
531     if (speed > 10000)
532     speed =1500;
533     //pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
534     // wait(1);
535     }
536
537     for(int i=1; i <= 2; i = i + 1)
538     {
539         smotor1.set_speed(speed);
540         smotor1.step_ah(MEDIO_G,cw);//2048
541         cw=!cw;
542         speed=speed+200;
543
544         if (speed > 10000)
545         speed =1500;
546         pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
547         // wait(1);
548     }
549     break;
550 }
551 break;
552
553 case 2:
554     switch(y)
555     {
556     case 1:
557         for(int i=1; i <= 2; i = i + 1)
558

```

```

553     case 2:
554         switch(y)
555         {
556         case 1:
557             for(int i=1; i <= 2; i = i + 1)
558             {
559                 smotor1.set_speed(speed);
560                 smotor1.step(AVANCE_1,cw); // 1785
561                 cw=!cw;
562                 speed=speed+200;
563
564                 if (speed > 10000)
565                 speed =1500;
566                 pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
567                 // wait(1);
568             }
569             break;
570
571         case 2:
572             for(int i=1; i <= 2; i = i + 1)
573             {
574                 smotor1.set_speed(speed);
575                 smotor1.step_h(MEDIO_G,cw);//2048
576                 cw=!cw;
577                 speed=speed+200;
578
579                 if (speed > 10000)
580                 speed =1500;
581                 // pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
582

```

```

585         for(int i=1; i <= 2; i = i + 1)
586         {
587             smotor1.set_speed(speed);
588             smotor1.step(AVANCE_1,cw);//1785
589             cw=!cw;
590             speed=speed+200;
591
592             if (speed > 10000)
593             speed =1500;
594             //pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
595             // wait(1);
596         }
597         for(int i=1; i <= 2; i = i + 1)
598         {
599             smotor1.set_speed(speed);
600             smotor1.step_ah(MEDIO_G,cw);//2048
601             cw=!cw;
602             speed=speed+200;
603
604             if (speed > 10000)
605             speed =1500;
606             //pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
607             //wait(1);
608         }
609         for(int i=1; i <= 2; i = i + 1)
610         {
611             smotor1.set_speed(speed);
612             smotor1.step(AVANCE_1,cw);//1785
613             cw=!cw;
614

```

```

621     }
622     break;
623
624     case 3:
625     for(int i=1; i <= 2; i = i + 1)
626     {
627         smotor1.set_speed(speed);
628         smotor1.step_h(MEDIO_G,cw);//2048
629         cw=!cw;
630         speed=speed+200;
631
632         if (speed > 10000)
633             speed =1500;
634         // pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
635         //wait(1);
636     }
637
638     for(int i=1; i <= 2; i = i + 1)
639     {
640         smotor1.set_speed(speed);
641         smotor1.step(AVANCE_2,cw);//3570
642         cw=!cw;
643         speed=speed+200;
644
645         if (speed > 10000)
646             speed =1500;
647         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
648         //wait(1);
649     }
650     for(int i=1; i <= 2; i = i + 1)
651
652     speed =1500;
653     // pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
654     // wait(1);
655     }
656
657     for(int i=1; i <= 2; i = i + 1)
658     {
659         smotor1.set_speed(speed);
660         smotor1.step(AVANCE_3,cw);//5362
661         cw=!cw;
662         speed=speed+200;
663
664         if (speed > 10000)
665             speed =1500;
666         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
667         //wait(1);
668     }
669
670     for(int i=1; i <= 2; i = i + 1)
671     {
672         smotor1.set_speed(speed);
673         smotor1.step_ah(MEDIO_G,cw)// 2048
674         cw=!cw;
675         speed=speed+200;
676
677         if (speed > 10000)
678             speed =1500;
679         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
680         // wait(1);
681     }
682     for(int i=1; i <= 2; i = i + 1)
683
684     speed =1500;
685     // pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
686     // wait(1);
687     }
688
689     for(int i=1; i <= 2; i = i + 1)
690     {
691         smotor1.set_speed(speed);
692         smotor1.step(AVANCE_1,cw);//1785
693         cw=!cw;
694         speed=speed+200;
695
696         if (speed > 10000)
697             speed =1500;
698         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
699         //wait(1);
700     }
701
702     for(int i=1; i <= 2; i = i + 1)
703     {
704         smotor1.set_speed(speed);
705         smotor1.step_ah(MEDIO_G,cw)// 2048
706         cw=!cw;
707         speed=speed+200;
708
709         if (speed > 10000)
710             speed =1500;
711         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
712         // wait(1);
713     }
714     for(int i=1; i <= 2; i = i + 1)
715
716     speed =1500;
717     // pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
718     // wait(1);
719     }
720     break;
721
722     case 3:
723     switch(y)
724     {
725     case 1:
726     for(int i=1; i <= 2; i = i + 1)
727     {
728         smotor1.set_speed(speed);
729         smotor1.step(AVANCE_2,cw);//3570
730         cw=!cw;
731         speed=speed+200;
732
733         if (speed > 10000)
734             speed =1500;
735         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
736         // wait(1);
737     }
738     break;
739
740     case 2:
741     for(int i=1; i <= 2; i = i + 1)
742     {
743         smotor1.set_speed(speed);
744         smotor1.step_ah(MEDIO_G,cw)// 2048
745         cw=!cw;
746         speed=speed+200;
747
748         if (speed > 10000)
749             speed =1500;
750         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
751         // wait(1);
752     }
753     break;
754
755     case 3:
756     for(int i=1; i <= 2; i = i + 1)
757     {
758         smotor1.set_speed(speed);
759         smotor1.step(AVANCE_1,cw);//1785
760         cw=!cw;
761         speed=speed+200;
762
763         if (speed > 10000)
764             speed =1500;
765         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
766         //wait(1);
767     }
768     break;
769
770     case 4:
771     for(int i=1; i <= 2; i = i + 1)
772     {
773         smotor1.set_speed(speed);
774         smotor1.step_h(MEDIO_G,cw)// 2048
775         cw=!cw;
776         speed=speed+200;
777
778         if (speed > 10000)
779             speed =1500;
780         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
781         // wait(1);
782     }
783     break;
784
785     case 5:
786     for(int i=1; i <= 2; i = i + 1)
787     {
788         smotor1.set_speed(speed);
789         smotor1.step(AVANCE_2,cw);//3570
790         cw=!cw;
791         speed=speed+200;
792
793         if (speed > 10000)
794             speed =1500;
795         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
796         // wait(1);
797     }
798     break;
799
800     case 6:
801     for(int i=1; i <= 2; i = i + 1)
802     {

```

```

801     for(int i=1; i <= 2; i = i + 1)
802     {
803         smotor1.set_speed(speed);
804         smotor1.step_h(MEDIO_G,cw);//2048
805         cw=!cw;
806         speed=speed+200;
807
808         if (speed > 10000)
809             speed =1500;
810         // pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
811         // wait(1);
812     }
813
814     for(int i=1; i <= 2; i = i + 1)
815     {
816         smotor1.set_speed(speed);
817         smotor1.step(AVANCE_2,cw);//3570
818         cw=!cw;
819         speed=speed+200;
820
821         if (speed > 10000)
822             speed =1500;
823         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
824         // wait(1);
825     }
826     for(int i=1; i <= 2; i = i + 1)
827     {
828         smotor1.set_speed(speed);
829         smotor1.step_ah(MEDIO_G,cw);//2048
830         cw=!cw;

```

```

837     }
838     for(int i=1; i <= 2; i = i + 1)
839     {
840         smotor1.set_speed(speed);
841         smotor1.step(AVANCE_2,cw);//3570
842         cw=!cw;
843         speed=speed+200;
844
845         if (speed > 10000)
846             speed =1500;
847         pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
848         // wait(1);
849     }
850     break;
851
852 case 4:
853     for(int i=1; i <= 2; i = i + 1)
854     {
855         smotor1.set_speed(speed);
856         smotor1.step_h(MEDIO_G,cw);//2048
857         cw=!cw;
858         speed=speed+200;
859
860         if (speed > 10000)
861             speed =1500;
862         // pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
863         // wait(1);
864     }
865     for(int i=1; i <= 2; i = i + 1)

```

```

865     for(int i=1; i <= 2; i = i + 1)
866     {
867         smotor1.set_speed(speed);
868         smotor1.step(AVANCE_3,cw);//5362
869         cw=!cw;
870         speed=speed+200;
871
872         if (speed > 10000)
873             speed =1500;
874         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
875         // wait(1);
876     }
877     for(int i=1; i <= 2; i = i + 1)
878     {
879         smotor1.set_speed(speed);
880         smotor1.step_ah(MEDIO_G,cw);//2048
881         cw=!cw;
882         speed=speed+200;
883
884         if (speed > 10000)
885             speed =1500;
886         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
887         // wait(1);
888     }
889     for(int i=1; i <= 2; i = i + 1)
890     {
891         smotor1.set_speed(speed);
892         smotor1.step(AVANCE_2,cw);//3570
893         cw=!cw;
894

```

```

897         if (speed > 10000)
898             speed =1500;
899         pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
900         // wait(1);
901         }
902         break;
903     }
904     break;
905 case 4:
906     switch(y)
907     {
908     case 1:
909
910         for(int i=1; i <= 2; i = i + 1)
911         {
912             smotor1.set_speed(speed);
913             smotor1.step(AVANCE_3,cw);//5362
914             cw=!cw;
915             speed=speed+200;
916
917             if (speed > 10000)
918                 speed =1500;
919             pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
920             // wait(1);
921             }
922             break;
923
924 case 2:
925         for(int i=1; i <= 2; i = i + 1)
926         {

```

```

921         // wait(1);
922         }
923         break;
924 case 2:
925         for(int i=1; i <= 2; i = i + 1)
926         {
927             smotor1.set_speed(speed);
928             smotor1.step_h(MEDIO_G,cw);//2048
929             cw=!cw;
930             speed=speed+200;
931
932             if (speed > 10000)
933                 speed =1500;
934             // pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
935             //wait(1);
936             }
937
938         for(int i=1; i <= 2; i = i + 1)
939         {
940             smotor1.set_speed(speed);
941             smotor1.step(AVANCE_1,cw);// 1785
942             cw=!cw;
943             speed=speed+200;
944
945             if (speed > 10000)
946                 speed =1500;
947             //pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
948             // wait(1);
949             }
950         for(int i=1; i <= 2; i = i + 1)

```

```

949         }
950         for(int i=1; i <= 2; i = i + 1)
951         {
952             smotor1.set_speed(speed);
953             smotor1.step_ah(MEDIO_G,cw);//2048
954             cw=!cw;
955             speed=speed+200;
956
957             if (speed > 10000)
958                 speed =1500;
959             //pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
960             //wait(1);
961             }
962         for(int i=1; i <= 2; i = i + 1)
963         {
964             smotor1.set_speed(speed);
965             smotor1.step(AVANCE_3,cw);//5362
966             cw=!cw;
967             speed=speed+200;
968
969             if (speed > 10000)
970                 speed =1500;
971             pc.printf("->fin de traslado piccolo fila %d y columna %d \n",x, y);
972             //wait(1);
973             }
974             break;
975 case 3:
976         for(int i=1; i <= 2; i = i + 1)
977         {
978             smotor1.set_speed(speed);

```

```

973     }
974     break;
975 case 3:
976     for(int i=1; i <= 2; i = i + 1)
977     {
978         smotor1.set_speed(speed);
979         smotor1.step_h(MEDIO_G,cw);//2048
980         cw=!cw;
981         speed=speed+200;
982
983         if (speed > 10000)
984             speed =1500;
985         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
986         // wait(1);
987     }
988
989     for(int i=1; i <= 2; i = i + 1)
990     {
991         smotor1.set_speed(speed);
992         smotor1.step(AVANCE_2,cw);//3570
993         cw=!cw;
994         speed=speed+200;
995
996         if (speed > 10000)
997             speed =1500;
998         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
999         // wait(1);
1000     }
1001     for(int i=1; i <= 2; i = i + 1)
1002     {

```

```

997         speed =1500;
998         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
999         // wait(1);
1000     }
1001     for(int i=1; i <= 2; i = i + 1)
1002     {
1003         smotor1.set_speed(speed);
1004         smotor1.step_ah(MEDIO_G,cw);//2048
1005         cw=!cw;
1006         speed=speed+200;
1007
1008         if (speed > 10000)
1009             speed =1500;
1010         //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
1011         // wait(1);
1012     }
1013     for(int i=1; i <= 2; i = i + 1)
1014     {
1015         smotor1.set_speed(speed);
1016         smotor1.step(AVANCE_3,cw);//5362
1017         cw=!cw;
1018         speed=speed+200;
1019
1020         if (speed > 10000)
1021             speed =1500;
1022         pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
1023         // wait(1);
1024     }
1025     break;
1026 case 4:

```

```

1025         break;
1026     case 4:
1027         for(int i=1; i <= 2; i = i + 1)
1028         {
1029             smotor1.set_speed(speed);
1030             smotor1.step_h(MEDIO_G,cw);//2048
1031             cw=!cw;
1032             speed=speed+200;
1033
1034             if (speed > 10000)
1035                 speed =1500;
1036             // pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
1037             // wait(1);
1038         }
1039
1040         for(int i=1; i <= 2; i = i + 1)
1041         {
1042             smotor1.set_speed(speed);
1043             smotor1.step(AVANCE_3,cw);//5362
1044             cw=!cw;
1045             speed=speed+200;
1046
1047             if (speed > 10000)
1048                 speed =1500;
1049             //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
1050             // wait(1);
1051         }
1052         for(int i=1; i <= 2; i = i + 1)
1053         {
1054             smotor1.set_speed(speed);

```

```

1054             smotor1.set_speed(speed);
1055             smotor1.step_ah(MEDIO_G,cw);//2048
1056             cw=!cw;
1057             speed=speed+200;
1058
1059             if (speed > 10000)
1060                 speed =1500;
1061             //pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
1062             // wait(1);
1063         }
1064         for(int i=1; i <= 2; i = i + 1)
1065         {
1066             smotor1.set_speed(speed);
1067             smotor1.step(AVANCE_3,cw);//5362
1068             cw=!cw;
1069             speed=speed+200;
1070
1071             if (speed > 10000)
1072                 speed =1500;
1073             pc.printf("-->fin de traslado piccolo fila %d y columna %d \n",x, y);
1074             //wait(1);
1075         }
1076         break;
1077     }
1078 }
1079 break;
1080 }
1081 }
1082 }
1083 }

```