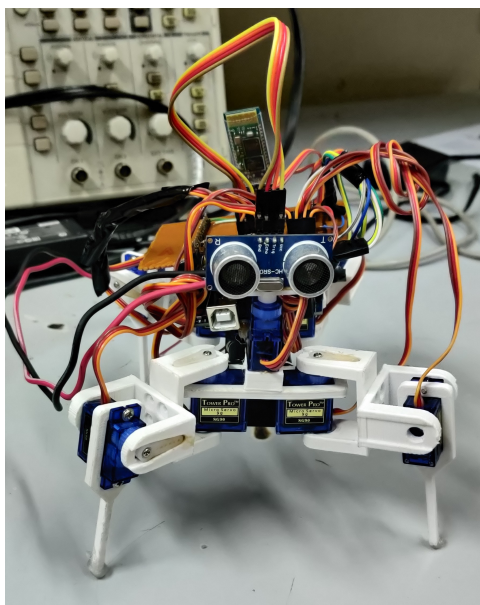




UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA
Año 2019 – 2^{do} Cuatrimestre

LABORATORIO DE MICROPROCESADORES (86.07)



Proyecto WILSON

TEMA: Cuadrúpedo controlado inalámbricamente

FECHA: 4/12/2019

INTEGRANTES:

Echegaray, Ignacio - #98868

<nachoechegaray@gmail.com>

Lowy, Ariel - #98298

<arilow@live.com.ar>

Rack, Lucas Alexis - #99425

<lucas_rack@live.com.ar>

Índice

1. Objetivos y estrategias	1
1.1. Descripción del proyecto y funcionamiento	1
1.2. Principales consideraciones	1
2. Lista de Componentes	2
3. Circuito esquemático	2
4. Diagrama de bloques	4
5. Diagrama de flujo	5
6. Resultados	6
6.1. Aplicación de I ² C	6
6.2. Movimiento	6
6.3. Cálculo de ángulos para los servo-motores	7
6.4. Interrupciones modulo bluetoooh	9
6.5. Ciclos del sensor	9
7. Conclusiones	10

1. Objetivos y estrategias

Como proyecto a llevar a cabo en la materia se optó por el desarrollo de un robot cuadrúpedo. Esto es, una pieza articulada con 4 extremidades capaces de permitirle desplazamiento en todas las direcciones de un plano. Para su programación se implementará un código escrito en *Assembler*, el cuál permite programar el microcontrolador deseado. Se hará uso de una placa Arduino Uno, la cuál posee el *micro* ATmega 328p y una "shield" para su adaptación a las distintas funcionalidades que se pretenden desarrollar.

1.1. Descripción del proyecto y funcionamiento

El presente proyecto tiene como objetivo diseñar e implementar un robot cuadrúpedo de ocho ejes capaz de desplazarse para adelante, para atrás y rotar en ambos sentidos. Se trabajará el movimiento de los ejes de cada extremidad por separado para luego integrarlas en el movimiento completo deseado. Cada eje poseerá un servomotor que puede ser posicionado según la instrucción en un ángulo entre 0 y 180 grados a la velocidad deseada.

El dispositivo será controlado con un mando inalámbrico mediante un módulo bluetooth receptor desarrollado por Arduino. Se utilizará como emisor cualquier dispositivo móvil con sistema operativo Android o IOS. Basta con descargar la aplicación "Arduino Bluetooth" de la tienda de aplicaciones y seguir unos sencillos pasos detallados en esta para establecer la comunicación necesaria.

En cuanto al sensor, siempre estará posicionado en el sentido del desplazamiento del conjunto, con el fin de evitar colisiones. En caso de desplazarse en reversa, no será contemplado su funcionamiento. De detectar una posible colisión, provocará una parada de todos los motores.

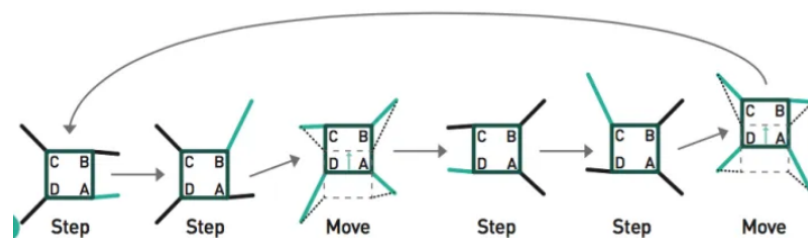


Figura 1.1: caminata.

1.2. Principales consideraciones

Una vez desarrolladas las funciones para mover los servos un ángulo específico, para optimizar al máximo el funcionamiento del cuadrúpedo, es importante la correcta interpolación de los ejes que posee, pudiendo así desplazarse la mayor distancia con el menor movimiento posible.

2. Lista de Componentes

- 9 Unidades de servomotores sg90 9g y sus respectivos cables.
- 1 placa Arduino Uno.
- 1 Sensor Ultrasónico Arduino Hc-sr04 .
- 1 Módulo Bluetooth Arduino HC05
- Fuente de entre 5 y 9 V según objetivos deseados.
- Articulaciones impresas en 3D.
- Cables Arduino.

3. Circuito esquemático

Se incluye a continuación el circuito físico creado en *Fritzing*

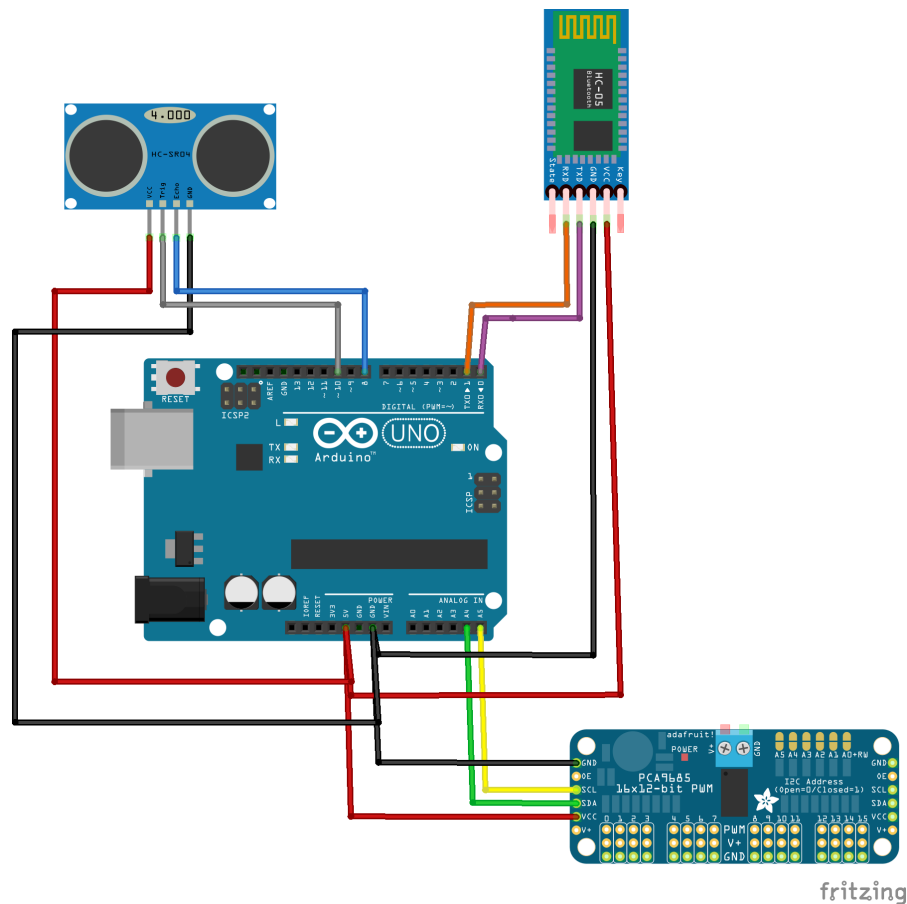
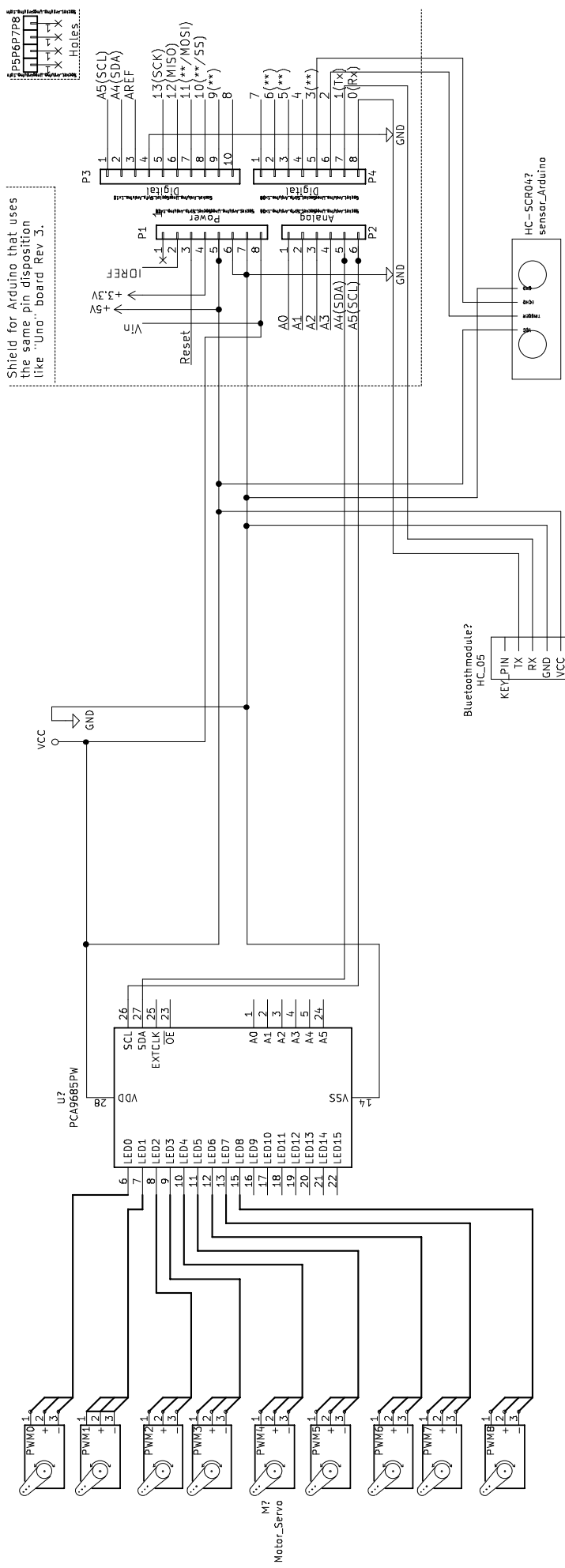


Figura 3.2: Diagrama de conexiones hecho en fritzing.



4. Diagrama de bloques

A continuación se adjunta el diagrama de bloques correspondiente al hardware y a los dispositivos que permiten el funcionamiento del proyecto.

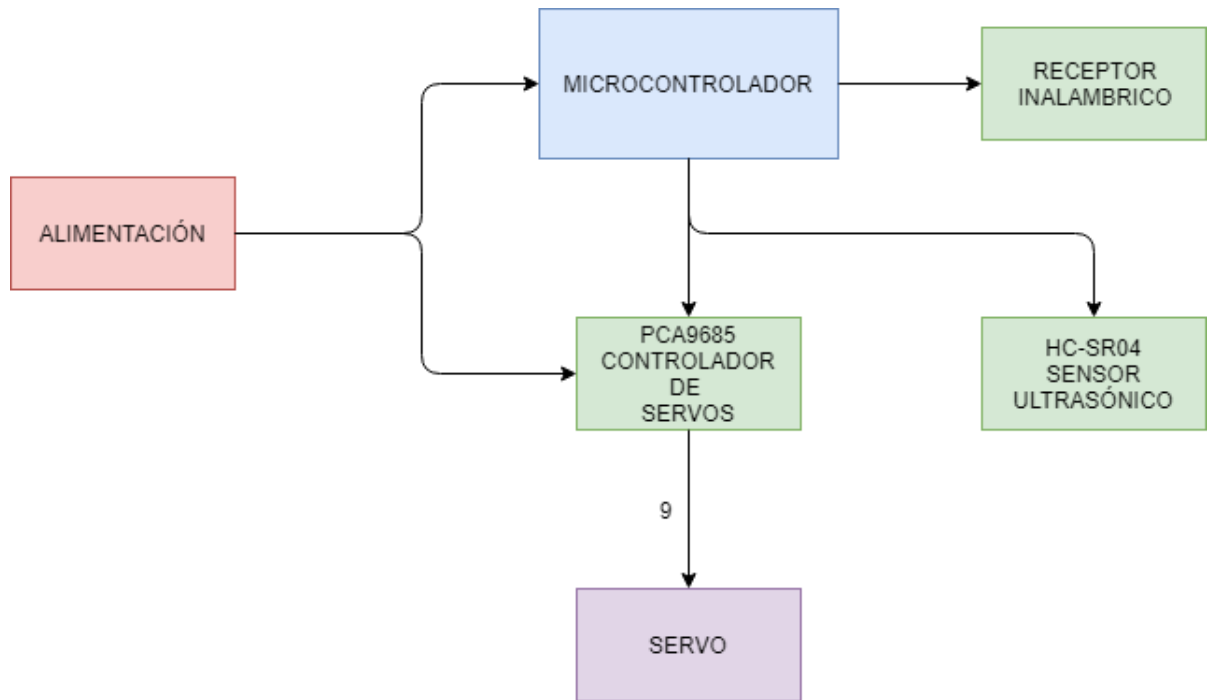


Figura 4.1: Diagrama de bloques

5. Diagrama de flujo

Se armó el siguiente diagrama de flujo basado en el código fuente desarrollado

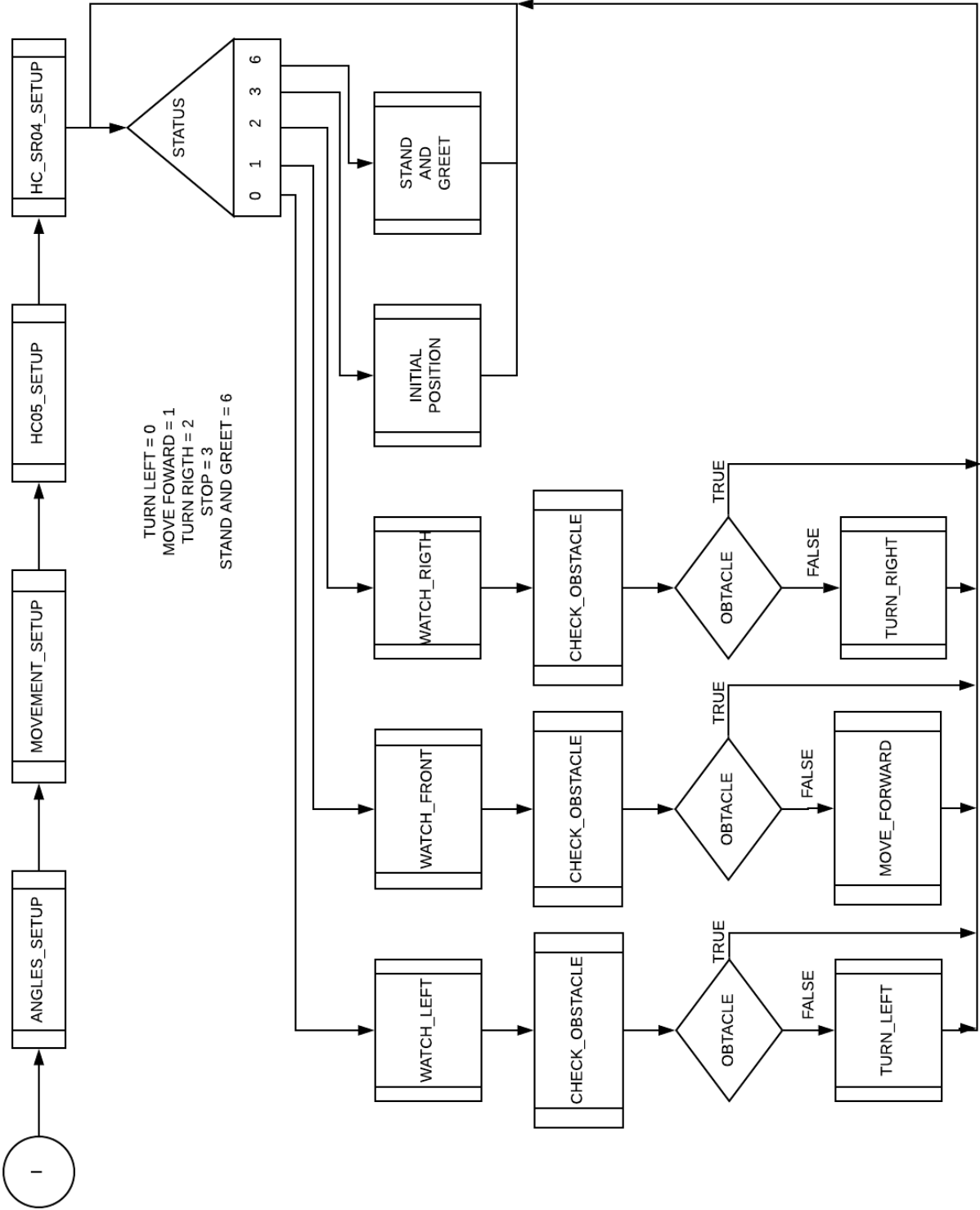


Figura 5.1: Diagrama de flujo del código fuente utilizado

6. Resultados

6.1. Aplicación de I²C

Los servos utilizados funcionan con entrada de señal PWM. Como el arduino posee solo 2 salidas con este tipo de señal y el proyecto funciona con 9 servos (2 por pata más uno del sensor de ultrasonido), se vio la necesidad de utilizar el protocolo de comunicación I²C, el cuál es utilizado por la placa PCA9685.

Esta forma de transferencia de datos está diseñada como un bus maestro-esclavo. La señal es siempre inicializada por un maestro (en este caso, el Atmega 328p) el cuál envía la señal y el esclavo reacciona.

La placa, básicamente funciona como un demultiplexor de 2 entradas y 16 salidas al que, mediante el protocolo mencionado, a una entrada se le envía la salida a utilizar y a la otra, se señal que se desea. De esta manera se pudo controlar todos los servos.

La siguiente imagen corresponde a los pulsos que utiliza la señal PWM para transmitir información a los periféricos. Por comodidad, el valor de *LED OFF* se fijó en 0, mientras que los valores de *LED ON* se determinaron de manera empírica para definir los posibles ángulos de cada servo. Estos valores se mantuvieron en un rango entre 500 y 2000

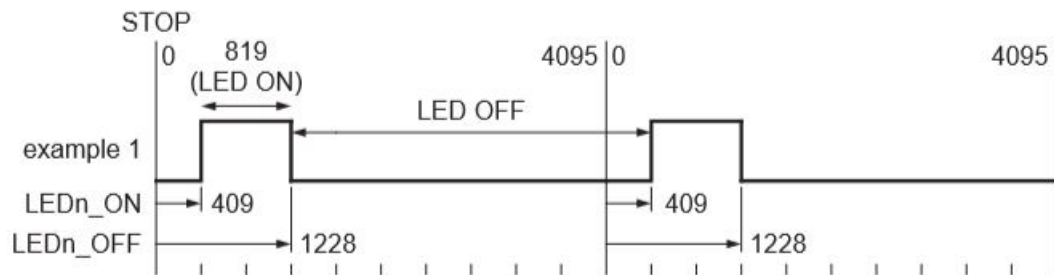


Figura 6.1: Señal de I²C

6.2. Movimiento

Para dar nombre a los movimientos de las articulaciones del cuadrúpedo, se basó en el movimiento que realizan los aviones sobre su propio eje para dar dirección a sus trayectorias. Se utilizó el término *YAW* para referirse al movimiento de los ejes número 1 de cada pata, mientras que se utilizó *ROLL* para la rotación de los servos correspondientes a los segundos ejes de las patas.

Una vez definido esto, se aplicaron estos términos en el código fuente para desarrollar todas las funciones correspondientes a movimiento, es decir, con combinaciones de *YAW-ROLL* se lograron todas las rutinas del cuadrúpedo.

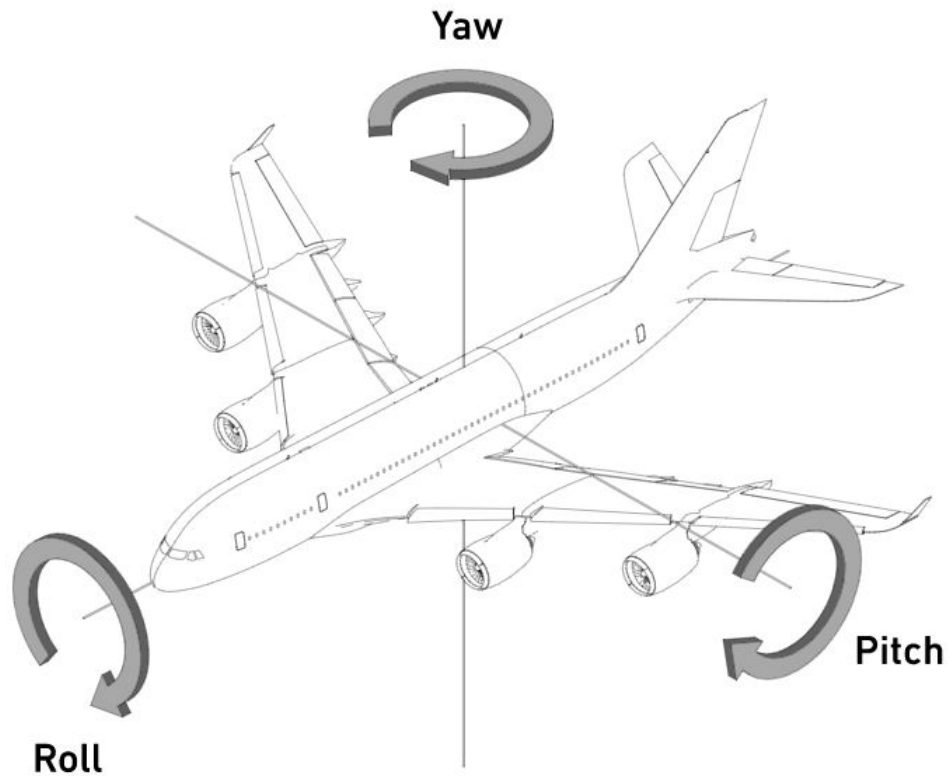


Figura 6.2: Movimiento rotacional de los ejes de un avión

6.3. Cálculo de ángulos para los servo-motores

Cómo punto de referencia del movimiento de los servos, se tomó al piso como ángulo inicial de los segundos ejes de cada pata según se observa en la figura 6.3.

Para el caso de los movimientos sobre el plano de apoyo, es decir, los *YAW*, se tomó como 0 a la posición que posiciona al servo más cercana de la parte trasera. Se puede dar una obtener una mejor perspectiva observando la figura 6.4

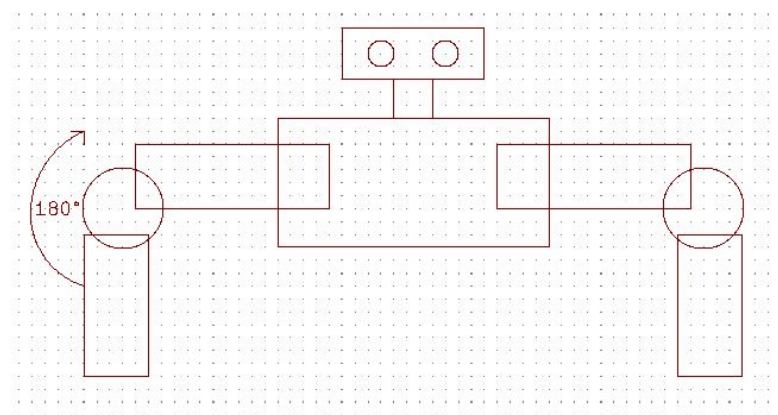


Figura 6.3: Esquemático frontal

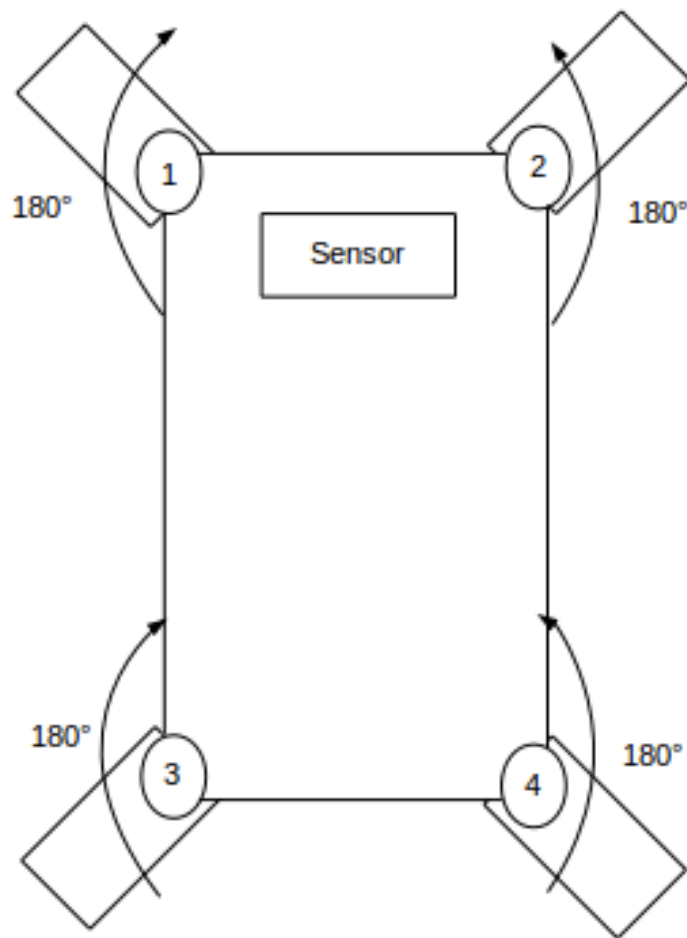


Figura 6.4: Esquemático superior

Cómo se explicó en la sección anterior, fue necesario traducir los ángulos observados en la aplicación real, a valores que el microcontrolador y el módulo PWM sean capaces de interpretar.

Cómo los servos no son idénticos entre sí, se calibraron por separado, obteniendo así la tabla a continuación extraída del código fuente, donde se le asignó un ancho de pulso a cada ángulo de cada servo que se utiliza en el proyecto.

PATA:	1	2	3	4
YAW_0	1750,	800,	1800,	385
YAW_15	1750,	800,	1700,	450
YAW_30	1750,	800,	1600,	575
YAW_45	1750,	800,	1550,	700
YAW_60	1575,	925,	1300,	825
YAW_75	1400,	1100,	1200,	925
YAW_90	1300,	1200,	1100,	1050
YAW_105	1140,	1360,	1000,	1200
YAW_120	1030,	1490,	850,	1325
YAW_135	925,	1600,	750,	1450
YAW_150	816,	1750,	750,	1450
YAW_165	708,	1870,	750,	1450
YAW_180	600,	2000,	750,	1450
ROLL_0	500,	500,	700,	500
ROLL_30	750,	750,	900,	800
ROLL_45	850,	850,	1000,	900
ROLL_60	950,	950,	1100,	1000
ROLL_90	1200,	1200,	1500,	1400
ROLL_120	1450,	1450,	1500,	1400
ROLL_150	1700,	1700,	1500,	1400
ROLL_180	2000,	2000,	1500,	1400

Figura 6.5: Tabla con los res

6.4. Interrupciones modulo bluetooth

Para modificar el valor del registro de estado de acción, se utilizo la interrupción de recepción de datos de la usart. Al tener conectado el modulo bluetooth, cada vez que se envíe un comando desde el celular se modificara esta variable y en el ciclo principal se cambiara de acción, para realizar el movimiento pertinente a este comando.

6.5. Ciclos del sensor

Para las instrucciones de traslación se verifica previamente si el sensor detecta un objeto dentro del rango de distancia estipulado. Si no es así, procede a realizar la acción. Se considera traslación como una iteración de las rutinas de movimiento "MOVE FORWARD", "TURN LEFT" y "TURN RIGHT", siendo estas las únicas que pueden provocar colisiones. A modo de analogía, se puede pensar como si una persona comprobara previo a cada paso si tiene un objeto en la dirección que se está moviendo.

Para las rutinas de doblar, fue necesario incorporar el noveno servo, capaz de rotar los 180° del frente panorámico del cuadrúpedo.

7. Conclusiones

Luego de finalizar el proyecto, se concluyó que se alcanzaron los objetivos propuestos desde un principio en la sección de objetivos. No obstante, se presentaron numerosos problemas en el proceso que se detallarán a continuación.

Al momento de hacer caminar al cuadrúpedo, se presentaron diversos problemas tales como la sinergia a aplicar entre las patas y lograr así un andar fluido y recto. Para solucionar esto se estudió con detalle los pasos que realizan otros casos de éxito encontrados en internet, llegando así a la combinación de rutinas que se puede encontrar en el código fuente y explicadas a lo largo del informe.

Con lo que respecta al código desarrollado en Assembler, el mayor problema y del cuál no se encontró solución, fue que para ciertos servos, cuando se los quería llevar a posiciones de ángulos extremos (0° o 180°) no cumplían con las instrucciones. En algunos casos, simplemente el servo no se movía hasta la posición y perdía comunicación con el Arduino, impidiendo que realice otra rutina sin reiniciar el sistema. En otros, simplemente iban a ángulos de valores alejados. La causa de esto último es que podíamos estar accediendo a otro espacio en memoria ROM inmediato a la ocupada por los ángulos. Cómo esto surgió en los días previos a la entrega, la solución fue corregir las rutinas de movimiento para prescindir de los ángulos conflictivos.

Por último, cabe destacar los problemas que surgieron con la alimentación del articulado. Se optó por prescindir de baterías inalámbricas por los siguientes motivos:

- El peso de las mismas, sumado a los componentes que tuvimos que agregarle al prototipo inicial, excedían la carga máxima del robot, haciendo que en varias ocasiones los servos cedan y se pierda estabilidad
- Se necesita una corriente inicial de aproximadamente 2 A para encender el microcontrolador y sus periféricos, algo que habría hecho ineficiente y costoso el uso de las baterías dada la cantidad de pruebas que se necesitaron. Cabe mencionar, que para soportar la corriente mencionada, se cambiaron los cables de alimentación por unos de mayor capacidad.
- para mantener encendido el Arduino 1, se requerían entre 6 y 9 V de tensión, algo que se hubiera alcanzado con, por lo menos, 4 pilas de 1,5 V cada una.

Apéndice

Código

Main

```
1  .INCLUDE          "M328PDEF.INC"
2  .INCLUDE          "DEFINES.INC"
3  .INCLUDE          "MACROS.INC"
4
5  .CSEG
6
7  .ORG              0x00
8      RJMP          MAIN
9  .ORG ICP1addr
10     RJMP ISR_ICP1
11 .ORG OVF1addr
12     RJMP RESTART_TIMER
13 .ORG              0x24
14     RJMP          HC05_RECEPTION_HANDLER
15
16
17 .ORG              INT_VECTORS_SIZE
18
19 MAIN:
20     STACK_INIT
21
22     SBI            DDRB,5
23
24     LDI            TEMP,(1<<SE)
25     OUT            MCUCR,TEMP
26     LDI            TEMP,(1<<SE)|(0<<SM2)|(0<<SM1)|(0<<SM0)
27     OUT            SMCR,TEMP
28
29
30     RCALL          INIC_ANGULOS           ;Inicializa valores de
    ↪ angulos
31     RCALL          MOVEMENT_SETUP        ;Inicializa protocolo I2C.
    ↪ Servos en pocision inicial
32     RCALL          HC05_SETUP             ;Inicializa bluetooth
33     RCALL          HCSR04_SETUP ; Sensor ultras[U+FFFD]o: Configuro el
    ↪ trigger y configuro el timer/wave generator
34
35
36     RCALL          STAND_GREET_LEFT
37
38     RCALL          INITIAL_POSITION
39
40 MAIN_LOOP:
41     RCALL          MOVEMENT_SWITCH
```

```
42         RJMP         MAIN_LOOP
43
44
45
46 ;error
47 ERRLOOP:
48         RJMP         ERRLOOP
49
50
51
52
53
54
55
56
57 .INCLUDE         "HCSR04.INC"
58 .INCLUDE         "HC05.INC"
59 .INCLUDE         "ANGULOS.INC"
60 .INCLUDE         "MOVEMENT.INC"
```

I2C

```
1 ; Equate statements
2 .equ      start          = $08      ; Start Condition
3 .equ      Rep_start      = $10      ; Repeated Start
   ↪ Condition Message
4 .equ      MT_SLA_ACK      = $18      ; Master
   ↪ Transmitter Slave Address Acknowledge
5 .equ      MT_I2C_DATA_ACK = $28      ; Master Transmitter
   ↪ I2C_DATA Acknowledge
6 .equ      MT_I2C_DATA_NACK = $30     ; Master Transmitter
   ↪ I2C_DATA Not Acknowledge
7 .equ      MR_SLA_ACK      = $40      ; Master Receiver
   ↪ Slave Address Acknowledge
8 .equ      MR_SLA_NACK     = $48      ; Master
   ↪ Receiver Slave Address Acknowledge
9 .equ      MR_I2C_DATA_ACK = $50      ; Master Receiver
   ↪ I2C_DATA Acknowledge
10 .equ      MR_I2C_DATA_NACK = $58    ; Master Receiver
   ↪ I2C_DATA Not Acknowledge
11 .equ      W              = 0        ; Write Bit
12 .equ      R              = 1        ; Read Bit
13 ;
14 ; Reset
15 ;
16
17
18 I2C_SETUP:
19 ;
20 ;setup DDR/IO
21 ;
22 ;set pullups, DDRC output
23     ldi      temp, 0xff
24     out      DDRC,temp
25     out      PORTC,temp
26
27 ;
28 ;setup speed etc. of i2c port
29 ;
30 ;400Khz=8MHz/(16+2*TWBR*CLKPRS) TWBR and clock prescaler set
31 ;i2c freq. clock prescaler default is 1, and is set in
32 ;TWSR
33     ldi      temp,72
34     sts      TWBR,temp
35
36     ldi      temp,0x00
37     sts      TWSR,temp
38     RET
39
```

```

40 ; -SEND I2C START
41 ; -This will send a I2C_DATA out as the address
42 ; -sends start condition and address-
43 I2C_START:
44     ldi        temp, (1<<TWINT) | (1<<TWSTA) | (1<<TWEN)
45     sts        TWCR, temp
46
47 ; wait for start condition to be sent.  when TWINT in TWCR is cleared, it
   ↪ is sent
48 WAIT_START:
49     lds        temp, TWCR
50     sbrs       temp, TWINT
51     rjmp       WAIT_START
52 ; check TWSR for bus status.
53 ; andi masks last three bits, which are 2=? 1:0 prescaler value
54     lds        temp, TWSR
55     andi       temp, 0b11111000
56     cpi        temp, START
57     breq       RET_I2C_START
58     jmp        errloop
59 RET_I2C_START:
60     ret
61
62 ; PUT I2C
63 ; bytes is stored in r17 aka I2C_DATA
64 ; use this by putting the address or I2C_DATA to put on the i2c line in
   ↪ I2C_DATA (r21)
65 ; then this function will disable the TW int, write I2C_DATA to TWDR, and
   ↪ wait for an ack
66 I2C_PUT:
67 NEXT1:
68     ldi        temp, (0<<TWINT) | (1<<TWEN)
69     sts        TWCR, temp
70     sts        TWDR, I2C_DATA
71     ldi        temp, (1<<TWINT) | (1<<TWEN)
72     sts        TWCR, temp
73 WAIT_DONE:
74     lds        temp, TWCR
75     sbrs       temp, TWINT
76     rjmp       WAIT_DONE
77     ret
78
79 I2C_GET:
80 WAIT_FOR_BYTE:
81     lds        temp, TWCR
82     sbrs       temp, TWINT
83     rjmp       WAIT_FOR_BYTE
84     lds        I2C_DATA, TWDR
85     ret

```

```

86
87 ; -SEND I2C STop
88 I2C_STOP:
89     ldi     temp, (1<<TWINT) || (1<<TWSTO) || (1<<TWEN)
90     sts     TWCR, temp
91 ; check TWCR to see if there is still a transmission- if not, stop bit
   ↳ has been sent
92 Check1:
93     lds     temp, TWCR
94     andi    temp, 0b00010000      ; Check to see that no
   ↳ transmission is going on
95     brne    Check1
96     ret

```

PCA9685

```
1  .INCLUDE "I2C.INC"
2
3  .EQU      PCA9685_DIR      = 0x40
4  .EQU      PCA9685_MODE1    = 0x00
5  .EQU      PCA9685_LED0     = 0x06
6  .EQU      PCA9685_PRE_SCALE = 0xFE
7
8
9  PCA9685_SETUP:
10      PUSH        I2C_DATA
11
12      RCALL        I2C_SETUP
13
14      RCALL        I2C_START
15      LDI          I2C_DATA,(PCA9685_DIR << 1)
16      RCALL        I2C_PUT
17      LDI          I2C_DATA,PCA9685_MODE1
18      RCALL        I2C_PUT
19      LDI          I2C_DATA,0x20 ;SET AUTO INCREMENT
20      RCALL        I2C_PUT
21      RCALL        I2C_STOP
22
23      RCALL        I2C_START
24      LDI          I2C_DATA,(PCA9685_DIR << 1)
25      RCALL        I2C_PUT
26      LDI          I2C_DATA,PCA9685_PRE_SCALE
27      RCALL        I2C_PUT
28      LDI          I2C_DATA,0x65
29      RCALL        I2C_PUT
30      RCALL        I2C_STOP
31
32      POP          I2C_DATA
33      RET
34
35  ;PRE
36  ;R23 -> SERVO NUMBER
37  ;R24 -> POS LOW
38  ;R25 -> POS HIGH
39  PCA9685_MOVE:
40      PUSH        I2C_DATA
41      ;MULTIPLY BY 4
42      LSL          SERVO_NUMBER
43      LSL          SERVO_NUMBER
44      ;ADD        LED0 DIR
45      ;CLR        I2C_DATA
46      LDI          I2C_DATA,PCA9685_LED0
47      ADD          SERVO_NUMBER,I2C_DATA
```

```

48
49      RCALL      I2C_START
50      LDI        I2C_DATA,(PCA9685_DIR << 1)
51      RCALL      I2C_PUT
52      MOV        I2C_DATA,SERVO_NUMBER
53      RCALL      I2C_PUT
54      LDI        I2C_DATA,0x00                ;LEDN_ON_LOW
55      RCALL      I2C_PUT
56      LDI        I2C_DATA,0x00                ;LEDN_ON_HIGH
57      RCALL      I2C_PUT
58      MOV        I2C_DATA,POSITION_LOW        ;LEDN_OFF_LOW
59      RCALL      I2C_PUT
60      MOV        I2C_DATA,POSITION_HIGH       ;LEDN_OFF_HIGH
61      RCALL      I2C_PUT
62      RCALL      I2C_STOP
63
64      POP        I2C_DATA
65      RET

```

MOVEMENT

```
1  .EQU          SERVO_MAX = 2600
2  .EQU          SERVO_MID = 1000
3  .EQU          SERVO_MIN = 500
4
5  .EQU          LEG_1 = 0X0102
6  .EQU          LEG_2 = 0X0304
7  .EQU          LEG_3 = 0X0506
8  .EQU          LEG_4 = 0X0708
9
10 .EQU          SERVO_TESTER = 1
11
12 .INCLUDE      "PCA9685.INC"
13
14 MOVEMENT_SETUP:
15     RCALL      PCA9685_SETUP
16     RCALL      INITIAL_POSITION
17     RET
18
19 INITIAL_POSITION:
20
21     LDI        LEG_L,LOW(LEG_1)
22     LDI        LEG_H,HIGH(LEG_1)
23     LDI        XL,LOW(ANG_YAW_135)
24     LDI        XH,HIGH(ANG_YAW_135)
25     RCALL      YAW_LEG
26     ;PATA 1 A 135
27     ;RCALL     MOVEMENT_DELAY
28
29     LDI        LEG_L,LOW(LEG_2)
30     LDI        LEG_H,HIGH(LEG_2)
31     LDI        XL,LOW(ANG_YAW_135)
32     LDI        XH,HIGH(ANG_YAW_135)
33     RCALL      YAW_LEG
34     ;PATA 2 A 135
35     ;RCALL     MOVEMENT_DELAY
36
37     LDI        LEG_L,LOW(LEG_3)
38     LDI        LEG_H,HIGH(LEG_3)
39     LDI        XL,LOW(ANG_YAW_45)
40     LDI        XH,HIGH(ANG_YAW_45)
41     RCALL      YAW_LEG
42     ;PATA 3 A 45
43     ;RCALL     MOVEMENT_DELAY
44
45     LDI        LEG_L,LOW(LEG_4)
46     LDI        LEG_H,HIGH(LEG_4)
47     LDI        XL,LOW(ANG_YAW_45)
```

```

48         LDI                XH,HIGH(ANG_YAW_45)
49         RCALL              YAW_LEG
50         ;PATA 4 A 45
51
52         LDI                LEG_L,LOW(LEG_1)
53         LDI                LEG_H,HIGH(LEG_1)
54         LDI                XL,LOW(ANG_ROLL_30)
55         LDI                XH,HIGH(ANG_ROLL_30)
56         RCALL              ROLL_LEG
57
58         LDI                LEG_L,LOW(LEG_2)
59         LDI                LEG_H,HIGH(LEG_2)
60         LDI                XL,LOW(ANG_ROLL_30)
61         LDI                XH,HIGH(ANG_ROLL_30)
62         RCALL              ROLL_LEG
63
64         LDI                LEG_L,LOW(LEG_3)
65         LDI                LEG_H,HIGH(LEG_3)
66         LDI                XL,LOW(ANG_ROLL_30)
67         LDI                XH,HIGH(ANG_ROLL_30)
68         RCALL              ROLL_LEG
69
70         LDI                LEG_L,LOW(LEG_4)
71         LDI                LEG_H,HIGH(LEG_4)
72         LDI                XL,LOW(ANG_ROLL_30)
73         LDI                XH,HIGH(ANG_ROLL_30)
74         RCALL              ROLL_LEG
75
76         RCALL              WATCH_FRONT
77
78         RET
79
80 MOVEMENT_SWITCH:           ;ACTUA COMO MQUINA DE ESTADOS
81         LDS                TEMP, GPIORO
82         CPI                TEMP, 0
83         BREQ               TRY_TURN_LEFT
84         CPI                TEMP, 1
85         BREQ               TRY_MOVE_FORWARD
86         CPI                TEMP, 2
87         BREQ               TRY_TURN_RIGHT
88         CPI                TEMP, 3
89         BREQ               TRY_INITIAL_POSITION
90         CPI                TEMP, 6
91         BREQ               TRY_STAND_GREET_LEFT
92         CPI                TEMP, 7
93         BREQ               TRY_PLAY_DEAD
94
95
96         RET

```

```

97
98
99
100
101 TRY_PLAY_DEAD:
102     RCALL     WATCH_RIGHT
103     RCALL     DELAY_100_MS
104     RCALL     WATCH_LEFT
105     RCALL     DELAY_100_MS
106
107     RCALL     PLAY_DEAD
108
109     RET
110
111
112
113 TRY_TURN_RIGHT:
114     RCALL     WATCH_RIGHT
115     RCALL     HCSR04_CHECK
116     LDS       TEMP, GPIOR1
117     CPI       TEMP, OBSTACLE_NOT_FOUND
118     BRNE     INITIAL_POSITION_LABEL
119     RCALL     TURN_RIGHT
120     RET
121
122 TRY_TURN_LEFT:
123     RCALL     WATCH_LEFT
124     RCALL     HCSR04_CHECK
125     LDS       TEMP, GPIOR1
126     CPI       TEMP, OBSTACLE_NOT_FOUND
127     BRNE     INITIAL_POSITION_LABEL
128     RCALL     TURN_LEFT
129     RET
130
131 TRY_MOVE_FORWARD:
132     RCALL     WATCH_FRONT
133     RCALL     HCSR04_CHECK
134     LDS       TEMP, GPIOR1
135     CPI       TEMP, OBSTACLE_NOT_FOUND
136     BRNE     INITIAL_POSITION_LABEL
137     RCALL     MOVE_FORWARD
138     RET
139
140 TRY_INITIAL_POSITION:
141     RCALL     INITIAL_POSITION
142     SLEEP
143     RET
144
145 TRY_STAND_GREET_LEFT:

```

```

146         RCALL          STAND_GREET_LEFT
147         RET
148
149
150 INITIAL_POSITION_LABEL:
151         RJMP          INITIAL_POSITION
152
153 WATCH_FRONT:
154         LDI           SERVO_NUMBER, 0
155         LDI           POSITION_LOW ,LOW(1000)
156         LDI           POSITION_HIGH, HIGH(1000)
157         RCALL          PCA9685_MOVE
158         RET
159
160 WATCH_RIGHT:
161         LDI           SERVO_NUMBER, 0
162         LDI           POSITION_LOW ,LOW(500)
163         LDI           POSITION_HIGH, HIGH(500)
164         RCALL          PCA9685_MOVE
165         RET
166
167 WATCH_LEFT:
168         LDI           SERVO_NUMBER, 0
169         LDI           POSITION_LOW ,LOW(1500)
170         LDI           POSITION_HIGH, HIGH(1500)
171         RCALL          PCA9685_MOVE
172         RET
173
174
175 ROLL_LEG:          ;MUEVE PATA EN PLANO VERTICAL
176         RCALL          FIND_ANG_LEG
177         MOV            R23,LEG_L
178         RCALL          PCA9685_MOVE
179
180         RET
181
182
183 YAW_LEG:           ;MUEVE PATA EN PLANO HORIZONTAL
184         RCALL          FIND_ANG_LEG
185         MOV            R23,LEG_H
186         RCALL          PCA9685_MOVE
187
188         RET
189
190 MOVE_FORWARD:      ;CAMINA HACIA ADELANTE
191         RCALL          MOVE_RIGHT_FORWARD
192         RCALL          MOVE_LEFT_FORWARD
193         RET
194

```

```

195 MOVE_LEFT_FORWARD:
196     LDI             LEG_L,LOW(LEG_1)
197     LDI             LEG_H,HIGH(LEG_1)
198     LDI             XL,LOW(ANG_ROLL_30)
199     LDI             XH,HIGH(ANG_ROLL_30)
200     RCALL           ROLL_LEG
201     ;PATA 1 ABAJO
202
203     LDI             LEG_L,LOW(LEG_4)
204     LDI             LEG_H,HIGH(LEG_4)
205     LDI             XL,LOW(ANG_ROLL_30)
206     LDI             XH,HIGH(ANG_ROLL_30)
207     RCALL           ROLL_LEG
208     ;PATA 4 ABAJO
209
210     LDI             LEG_L,LOW(LEG_2)
211     LDI             LEG_H,HIGH(LEG_2)
212     LDI             XL,LOW(ANG_ROLL_60)
213     LDI             XH,HIGH(ANG_ROLL_60)
214     RCALL           ROLL_LEG
215     ;PARA 2 ARRIBA
216
217     LDI             LEG_L,LOW(LEG_3)
218     LDI             LEG_H,HIGH(LEG_3)
219     LDI             XL,LOW(ANG_ROLL_60)
220     LDI             XH,HIGH(ANG_ROLL_60)
221     RCALL           ROLL_LEG
222     ;PATA 3 ARRIBA
223
224     RCALL           DELAY_100_MS      ;ACHICAR DELAY
225
226     LDI             LEG_L,LOW(LEG_1)
227     LDI             LEG_H,HIGH(LEG_1)
228     LDI             XL,LOW(ANG_YAW_90)
229     LDI             XH,HIGH(ANG_YAW_90)
230     RCALL           YAW_LEG
231     ;PATA 1 A 90
232
233     LDI             LEG_L,LOW(LEG_4)
234     LDI             LEG_H,HIGH(LEG_4)
235     LDI             XL,LOW(ANG_YAW_30)
236     LDI             XH,HIGH(ANG_YAW_30)
237     RCALL           YAW_LEG
238     ;PATA 4 A 30
239
240     LDI             LEG_L,LOW(LEG_2)
241     LDI             LEG_H,HIGH(LEG_2)
242     LDI             XL,LOW(ANG_YAW_135)
243     LDI             XH,HIGH(ANG_YAW_135)

```

```

244         RCALL          YAW_LEG
245         ;PATA 2 A 150
246
247         LDI             LEG_L,LOW(LEG_3)
248         LDI             LEG_H,HIGH(LEG_3)
249         LDI             XL,LOW(ANG_YAW_90)
250         LDI             XH,HIGH(ANG_YAW_90)
251         RCALL          YAW_LEG
252         ;PATA 3 A 90
253
254         RCALL DELAY_100_MS
255         RET
256
257
258     MOVE_RIGHT_FORWARD:
259
260         LDI             LEG_L,LOW(LEG_2)
261         LDI             LEG_H,HIGH(LEG_2)
262         LDI             XL,LOW(ANG_ROLL_30)
263         LDI             XH,HIGH(ANG_ROLL_30)
264         RCALL          ROLL_LEG
265         ;PATA 2 ARRIBA
266         LDI             LEG_L,LOW(LEG_3)
267         LDI             LEG_H,HIGH(LEG_3)
268         LDI             XL,LOW(ANG_ROLL_30)
269         LDI             XH,HIGH(ANG_ROLL_30)
270         RCALL          ROLL_LEG
271         ;PATA 3 ARRIBA
272         LDI             LEG_L,LOW(LEG_1)
273         LDI             LEG_H,HIGH(LEG_1)
274         LDI             XL,LOW(ANG_ROLL_60)
275         LDI             XH,HIGH(ANG_ROLL_60)
276         RCALL          ROLL_LEG
277         ;PARA 1 ABAJO
278         LDI             LEG_L,LOW(LEG_4)
279         LDI             LEG_H,HIGH(LEG_4)
280         LDI             XL,LOW(ANG_ROLL_60)
281         LDI             XH,HIGH(ANG_ROLL_60)
282         RCALL          ROLL_LEG
283         ;PATA 4 ABAJO
284
285         RCALL          DELAY_100_MS
286
287         LDI             LEG_L,LOW(LEG_2)
288         LDI             LEG_H,HIGH(LEG_2)
289         LDI             XL,LOW(ANG_YAW_90)
290         LDI             XH,HIGH(ANG_YAW_90)
291         RCALL          YAW_LEG
292         ;PATA 2 A 90

```

```

293
294         LDI             LEG_L,LOW(LEG_3)
295         LDI             LEG_H,HIGH(LEG_3)
296         LDI             XL,LOW(ANG_YAW_45)
297         LDI             XH,HIGH(ANG_YAW_45)
298         RCALL           YAW_LEG
299 ;         RCALL         DELAY_10_S
300         ;PATA 3 A 45
301
302         LDI             LEG_L,LOW(LEG_1)
303         LDI             LEG_H,HIGH(LEG_1)
304         LDI             XL,LOW(ANG_YAW_150)
305         LDI             XH,HIGH(ANG_YAW_150)
306         RCALL           YAW_LEG
307         ;PATA 1 A 150
308
309         LDI             LEG_L,LOW(LEG_4)
310         LDI             LEG_H,HIGH(LEG_4)
311         LDI             XL,LOW(ANG_YAW_90)
312         LDI             XH,HIGH(ANG_YAW_90)
313         RCALL           YAW_LEG
314         ;PATA 4 A 90
315
316
317         RCALL DELAY_100_MS
318         RET
319
320 TURN_RIGHT:           ;DOBLA A LA DERECHA
321     TURN_RIGHT_MOVE_1:
322         LDI             LEG_L,LOW(LEG_2)
323         LDI             LEG_H,HIGH(LEG_2)
324         LDI             XL,LOW(ANG_ROLL_30)
325         LDI             XH,HIGH(ANG_ROLL_30)
326         RCALL           ROLL_LEG
327         ;PATA 2 ARRIBA
328         LDI             LEG_L,LOW(LEG_1)
329         LDI             LEG_H,HIGH(LEG_1)
330         LDI             XL,LOW(ANG_ROLL_0)
331         LDI             XH,HIGH(ANG_ROLL_0)
332         RCALL           ROLL_LEG
333         ;PATA 1 ABAJO
334         LDI             LEG_L,LOW(LEG_4)
335         LDI             LEG_H,HIGH(LEG_4)
336         LDI             XL,LOW(ANG_ROLL_30)
337         LDI             XH,HIGH(ANG_ROLL_30)
338         RCALL           ROLL_LEG
339         ;PATA 4 ABAJO
340         LDI             LEG_L,LOW(LEG_3)
341         LDI             LEG_H,HIGH(LEG_3)

```

```

342         LDH          XL,LOW(ANG_ROLL_30)
343         LDH          XH,HIGH(ANG_ROLL_30)
344         RCALL        ROLL_LEG
345         ;PATA 3 ARRIBA
346
347         LDH          LEG_L,LOW(LEG_2)
348         LDH          LEG_H,HIGH(LEG_2)
349         LDH          XL,LOW(ANG_YAW_90)
350         LDH          XH,HIGH(ANG_YAW_90)
351         RCALL        YAW_LEG
352         ;PATA 2 A 90
353
354         LDH          LEG_L,LOW(LEG_1)
355         LDH          LEG_H,HIGH(LEG_1)
356         LDH          XL,LOW(ANG_YAW_135)
357         LDH          XH,HIGH(ANG_YAW_135)
358         RCALL        YAW_LEG
359         ;PATA 1 A 135
360
361         LDH          LEG_L,LOW(LEG_4)
362         LDH          LEG_H,HIGH(LEG_4)
363         LDH          XL,LOW(ANG_YAW_45)
364         LDH          XH,HIGH(ANG_YAW_45)
365         RCALL        YAW_LEG
366         ;PATA 4 A 45
367
368         LDH          LEG_L,LOW(LEG_3)
369         LDH          LEG_H,HIGH(LEG_3)
370         LDH          XL,LOW(ANG_YAW_90)
371         LDH          XH,HIGH(ANG_YAW_90)
372         RCALL        YAW_LEG
373         ;PATA 3 A 90
374
375         RCALL DELAY_100_MS
376
377     TURN_RIGHT_MOVE_2:
378         LDH          LEG_L,LOW(LEG_2)
379         LDH          LEG_H,HIGH(LEG_2)
380         LDH          XL,LOW(ANG_ROLL_0)
381         LDH          XH,HIGH(ANG_ROLL_0)
382         RCALL        ROLL_LEG
383         ;PATA 2 ABAJO
384         LDH          LEG_L,LOW(LEG_1)
385         LDH          LEG_H,HIGH(LEG_1)
386         LDH          XL,LOW(ANG_ROLL_30)
387         LDH          XH,HIGH(ANG_ROLL_30)
388         RCALL        ROLL_LEG
389         ;PATA 1 ARRIBA
390         LDH          LEG_L,LOW(LEG_4)

```

```

391     LDI             LEG_H,HIGH(LEG_4)
392     LDI             XL,LOW(ANG_ROLL_60)
393     LDI             XH,HIGH(ANG_ROLL_60)
394     RCALL           ROLL_LEG
395     ;PATA 4 ARRIBA
396     LDI             LEG_L,LOW(LEG_3)
397     LDI             LEG_H,HIGH(LEG_3)
398     LDI             XL,LOW(ANG_ROLL_30)
399     LDI             XH,HIGH(ANG_ROLL_30)
400     RCALL           ROLL_LEG
401     ;PATA 3 ABAJO
402
403     LDI             LEG_L,LOW(LEG_2)
404     LDI             LEG_H,HIGH(LEG_2)
405     LDI             XL,LOW(ANG_YAW_135)
406     LDI             XH,HIGH(ANG_YAW_135)
407     RCALL           YAW_LEG
408     ;PATA 2 A 135
409
410     LDI             LEG_L,LOW(LEG_1)
411     LDI             LEG_H,HIGH(LEG_1)
412     LDI             XL,LOW(ANG_YAW_180)
413     LDI             XH,HIGH(ANG_YAW_180)
414     RCALL           YAW_LEG
415     ;PATA 1 A 180
416
417     LDI             LEG_L,LOW(LEG_4)
418     LDI             LEG_H,HIGH(LEG_4)
419     LDI             XL,LOW(ANG_YAW_30)
420     LDI             XH,HIGH(ANG_YAW_30)
421     RCALL           YAW_LEG
422     ;PATA 4 A 0
423
424     LDI             LEG_L,LOW(LEG_3)
425     LDI             LEG_H,HIGH(LEG_3)
426     LDI             XL,LOW(ANG_YAW_45)
427     LDI             XH,HIGH(ANG_YAW_45)
428     RCALL           YAW_LEG
429     ;PATA 3 A 45
430
431     RCALL DELAY_100_MS
432
433     TURN_RIGHT_MOVE_3:
434     LDI             LEG_L,LOW(LEG_2)
435     LDI             LEG_H,HIGH(LEG_2)
436     LDI             XL,LOW(ANG_ROLL_30)
437     LDI             XH,HIGH(ANG_ROLL_30)
438     RCALL           ROLL_LEG
439     ;PATA 2 ARRIBA

```

```

440         LDI             LEG_L,LOW(LEG_3)
441         LDI             LEG_H,HIGH(LEG_3)
442         LDI             XL,LOW(ANG_ROLL_30)
443         LDI             XH,HIGH(ANG_ROLL_30)
444         RCALL           ROLL_LEG
445         ;PATA 3 ARRIBA
446
447         LDI             LEG_L,LOW(LEG_2)
448         LDI             LEG_H,HIGH(LEG_2)
449         LDI             XL,LOW(ANG_YAW_135)
450         LDI             XH,HIGH(ANG_YAW_135)
451         RCALL           YAW_LEG
452         ;PATA 2 A 90
453
454         LDI             LEG_L,LOW(LEG_3)
455         LDI             LEG_H,HIGH(LEG_3)
456         LDI             XL,LOW(ANG_YAW_45)
457         LDI             XH,HIGH(ANG_YAW_45)
458         RCALL           YAW_LEG
459         ;PATA 3 A 90
460
461         RCALL DELAY_100_MS
462
463         RCALL           INITIAL_POSITION
464
465         RET
466
467
468     TURN_LEFT:           ;DOBLA A LA IZQUIERDA
469         TURN_LEFT_MOVE_1:
470             LDI             LEG_L,LOW(LEG_1)
471             LDI             LEG_H,HIGH(LEG_1)
472             LDI             XL,LOW(ANG_ROLL_30)
473             LDI             XH,HIGH(ANG_ROLL_30)
474             RCALL           ROLL_LEG
475             ;PATA 1 ARRIBA
476             LDI             LEG_L,LOW(LEG_2)
477             LDI             LEG_H,HIGH(LEG_2)
478             LDI             XL,LOW(ANG_ROLL_0)
479             LDI             XH,HIGH(ANG_ROLL_0)
480             RCALL           ROLL_LEG
481             ;PATA 2 ABAJO
482             LDI             LEG_L,LOW(LEG_3)
483             LDI             LEG_H,HIGH(LEG_3)
484             LDI             XL,LOW(ANG_ROLL_0)
485             LDI             XH,HIGH(ANG_ROLL_0)
486             RCALL           ROLL_LEG
487             ;PATA 3 ABAJO
488             LDI             LEG_L,LOW(LEG_4)

```

```

489         LDI             LEG_H,HIGH(LEG_4)
490         LDI             XL,LOW(ANG_ROLL_30)
491         LDI             XH,HIGH(ANG_ROLL_30)
492         RCALL           ROLL_LEG
493         ;PATA 4 ARRIBA
494
495         LDI             LEG_L,LOW(LEG_1)
496         LDI             LEG_H,HIGH(LEG_1)
497         LDI             XL,LOW(ANG_YAW_90)
498         LDI             XH,HIGH(ANG_YAW_90)
499         RCALL           YAW_LEG
500         ;PATA 1 A 90
501
502         LDI             LEG_L,LOW(LEG_2)
503         LDI             LEG_H,HIGH(LEG_2)
504         LDI             XL,LOW(ANG_YAW_135)
505         LDI             XH,HIGH(ANG_YAW_135)
506         RCALL           YAW_LEG
507         ;PATA 2 A 135
508
509         LDI             LEG_L,LOW(LEG_3)
510         LDI             LEG_H,HIGH(LEG_3)
511         LDI             XL,LOW(ANG_YAW_45)
512         LDI             XH,HIGH(ANG_YAW_45)
513         RCALL           YAW_LEG
514         ;PATA 3 A 45
515
516         LDI             LEG_L,LOW(LEG_4)
517         LDI             LEG_H,HIGH(LEG_4)
518         LDI             XL,LOW(ANG_YAW_90)
519         LDI             XH,HIGH(ANG_YAW_90)
520         RCALL           YAW_LEG
521         ;PATA 4 A 90
522
523         RCALL DELAY_100_MS
524
525     TURN_LEFT_MOVE_2:
526         LDI             LEG_L,LOW(LEG_1)
527         LDI             LEG_H,HIGH(LEG_1)
528         LDI             XL,LOW(ANG_ROLL_0)
529         LDI             XH,HIGH(ANG_ROLL_0)
530         RCALL           ROLL_LEG
531         ;PATA 1 ABAJO
532         LDI             LEG_L,LOW(LEG_2)
533         LDI             LEG_H,HIGH(LEG_2)
534         LDI             XL,LOW(ANG_ROLL_30)
535         LDI             XH,HIGH(ANG_ROLL_30)
536         RCALL           ROLL_LEG
537         ;PATA 2 ARRIBA

```

```

538     LDI             LEG_L,LOW(LEG_3)
539     LDI             LEG_H,HIGH(LEG_3)
540     LDI             XL,LOW(ANG_ROLL_30)
541     LDI             XH,HIGH(ANG_ROLL_30)
542     RCALL           ROLL_LEG
543     ;PATA 3 ARRIBA
544     LDI             LEG_L,LOW(LEG_4)
545     LDI             LEG_H,HIGH(LEG_4)
546     LDI             XL,LOW(ANG_ROLL_30)
547     LDI             XH,HIGH(ANG_ROLL_30)
548     RCALL           ROLL_LEG
549     ;PATA 4 ABAJO
550
551     LDI             LEG_L,LOW(LEG_1)
552     LDI             LEG_H,HIGH(LEG_1)
553     LDI             XL,LOW(ANG_YAW_135)
554     LDI             XH,HIGH(ANG_YAW_135)
555     RCALL           YAW_LEG
556     ;PATA 1 A 135
557
558     LDI             LEG_L,LOW(LEG_2)
559     LDI             LEG_H,HIGH(LEG_2)
560     LDI             XL,LOW(ANG_YAW_180)
561     LDI             XH,HIGH(ANG_YAW_180)
562     RCALL           YAW_LEG
563     ;PATA 2 A 180
564
565     LDI             LEG_L,LOW(LEG_3)
566     LDI             LEG_H,HIGH(LEG_3)
567     LDI             XL,LOW(ANG_YAW_0)
568     LDI             XH,HIGH(ANG_YAW_0)
569     RCALL           YAW_LEG
570     ;PATA 3 A 0
571
572     LDI             LEG_L,LOW(LEG_4)
573     LDI             LEG_H,HIGH(LEG_4)
574     LDI             XL,LOW(ANG_YAW_45)
575     LDI             XH,HIGH(ANG_YAW_45)
576     RCALL           YAW_LEG
577     ;PATA 4 A 45
578
579     RCALL DELAY_100_MS
580
581     TURN_LEFT_MOVE_3:
582     LDI             LEG_L,LOW(LEG_1)
583     LDI             LEG_H,HIGH(LEG_1)
584     LDI             XL,LOW(ANG_ROLL_30)
585     LDI             XH,HIGH(ANG_ROLL_30)
586     RCALL           ROLL_LEG

```

```

587      ;PATA 1 ARRIBA
588      LDI          LEG_L,LOW(LEG_4)
589      LDI          LEG_H,HIGH(LEG_4)
590      LDI          XL,LOW(ANG_ROLL_30)
591      LDI          XH,HIGH(ANG_ROLL_30)
592      RCALL        ROLL_LEG
593      ;PATA 4 ARRIBA
594
595      LDI          LEG_L,LOW(LEG_1)
596      LDI          LEG_H,HIGH(LEG_1)
597      LDI          XL,LOW(ANG_YAW_135)
598      LDI          XH,HIGH(ANG_YAW_135)
599      RCALL        YAW_LEG
600      ;PATA 1 A 90
601
602      LDI          LEG_L,LOW(LEG_4)
603      LDI          LEG_H,HIGH(LEG_4)
604      LDI          XL,LOW(ANG_YAW_45)
605      LDI          XH,HIGH(ANG_YAW_45)
606      RCALL        YAW_LEG
607      ;PATA 4 A 90
608
609      RCALL DELAY_100_MS
610
611      RCALL        INITIAL_POSITION
612
613      RET
614
615
616      PLAY_DEAD:  ;SE HACE EL MUERTO
617      LDI          LEG_L,LOW(LEG_3)
618      LDI          LEG_H,HIGH(LEG_3)
619      LDI          XL,LOW(ANG_ROLL_30)
620      LDI          XH,HIGH(ANG_ROLL_30)
621      RCALL        ROLL_LEG
622      LDI          XL,LOW(ANG_YAW_135)
623      LDI          XH,HIGH(ANG_YAW_135)
624      RCALL        YAW_LEG
625
626      LDI          LEG_L,LOW(LEG_4)
627      LDI          LEG_H,HIGH(LEG_4)
628      LDI          XL,LOW(ANG_ROLL_30)
629      LDI          XH,HIGH(ANG_ROLL_30)
630      RCALL        ROLL_LEG
631      LDI          XL,LOW(ANG_YAW_135)
632      LDI          XH,HIGH(ANG_YAW_135)
633      RCALL        YAW_LEG
634
635      RCALL        MOVEMENT_DELAY

```

```

636
637         RCALL          INITIAL_POSITION
638
639         RET
640
641
642     STAND_GREET_LEFT:  ;SALUDA
643
644         LDI             LEG_L,LOW(LEG_1)
645         LDI             LEG_H,HIGH(LEG_1)
646         LDI             XL,LOW(ANG_ROLL_30)
647         LDI             XH,HIGH(ANG_ROLL_30)
648         RCALL          ROLL_LEG
649         ;PATA 2 ARRIBA
650         LDI             XL,LOW(ANG_YAW_150)
651         LDI             XH,HIGH(ANG_YAW_150)
652         RCALL          YAW_LEG
653         ;PATA 2 A 90
654
655         LDI             LEG_L,LOW(LEG_2)
656         LDI             LEG_H,HIGH(LEG_2)
657         LDI             XL,LOW(ANG_ROLL_30)
658         LDI             XH,HIGH(ANG_ROLL_30)
659         RCALL          ROLL_LEG
660         ;PATA 2 ARRIBA
661         LDI             XL,LOW(ANG_YAW_150)
662         LDI             XH,HIGH(ANG_YAW_150)
663         RCALL          YAW_LEG
664         ;PATA 2 A 90
665
666         LDI             LEG_L,LOW(LEG_3)
667         LDI             LEG_H,HIGH(LEG_3)
668         LDI             XL,LOW(ANG_ROLL_30)
669         LDI             XH,HIGH(ANG_ROLL_30)
670         RCALL          ROLL_LEG
671         ;PATA 2 ARRIBA
672         LDI             XL,LOW(ANG_YAW_60)
673         LDI             XH,HIGH(ANG_YAW_60)
674         RCALL          YAW_LEG
675         ;PATA 2 A 90
676
677         LDI             LEG_L,LOW(LEG_4)
678         LDI             LEG_H,HIGH(LEG_4)
679         LDI             XL,LOW(ANG_ROLL_30)
680         LDI             XH,HIGH(ANG_ROLL_30)
681         RCALL          ROLL_LEG
682         ;PATA 2 ARRIBA
683         LDI             XL,LOW(ANG_YAW_60)
684         LDI             XH,HIGH(ANG_YAW_60)

```

```

685         RCALL          YAW_LEG
686         ;PATA 2 A 90
687
688         RCALL MOVEMENT_DELAY
689
690 GREET_LEFT:
691         LDI              LEG_L,LOW(LEG_1)
692         LDI              LEG_H,HIGH(LEG_1)
693         LDI              XL,LOW(ANG_ROLL_180)
694         LDI              XH,HIGH(ANG_ROLL_180)
695         RCALL            ROLL_LEG
696
697         RCALL MOVEMENT_DELAY
698
699         LDI              TEMP, 3
700
701 LOOP_GREET_LEFT:
702         PUSH            TEMP
703         LDI              XL,LOW(ANG_YAW_180)
704         LDI              XH,HIGH(ANG_YAW_180)
705         RCALL            YAW_LEG
706
707         RCALL DELAY_100_MS
708
709         LDI              XL,LOW(ANG_YAW_90)
710         LDI              XH,HIGH(ANG_YAW_90)
711         RCALL            YAW_LEG
712
713         RCALL            DELAY_100_MS
714
715         POP              TEMP
716         DEC              TEMP
717
718         BRNE LOOP_GREET_LEFT
719         RCALL INITIAL_POSITION
720         RET
721
722
723
724 MOVEMENT_DELAY:          ;DELAY DE 1 SEGUNDO PARA UN CLOCK DE 16MHz
725         PUSH            R26
726         LDI              R26,4
727 LOOP_MOVEMENT_DELAY:
728         RCALL            DELAY_100_MS
729         DEC              R26
730         brne LOOP_MOVEMENT_DELAY
731
732         POP              R26
733         RET

```

```

734
735 DELAY_100_MS:           ;DELAY DE 100 MILISEGUNDOS PARA UN CLOCK DE 16MHz
736     PUSH                R25
737     LDI                  R25,100
738 LOOP_DELAY_100MS:
739     RCALL                DELAY_1_MS
740     DEC                  R25
741     brne LOOP_DELAY_100MS
742
743     POP                  R25
744     RET
745
746 DELAY_1_MS:             ;DELAY DE 1 MILISEGUNDO PARA UN CLOCK DE 16MHz
747     push r22
748     push r24
749     ldi r22, 100 ;
750 LOOP_DELAY_1:
751     ldi R24, 160
752 LOOP_DELAY_2:
753     DEC                  R24
754     brne LOOP_DELAY_2
755     dec r22
756     brne LOOP_DELAY_1
757
758     pop r24
759     pop r22
760     ret

```

ANGULOS

```

1  .EQU          SERVO_TESTER_1 = 7
2  .EQU          SERVO_TESTER_2 = 5
3  .EQU          ANG_TESTER_1 = 1325
4  .EQU          ANG_TESTER_2 = 900
5
6  .EQU          CANT_ANGS = 4
7
8
9  .DSEG
10
11 .ORG          0X500 ; VER DIRECCION
12 ANG_YAW_0:    .BYTE          (CANT_ANGS<<1)
13 .ORG          0X508
14 ANG_YAW_15:   .BYTE          (CANT_ANGS<<1)
15 .ORG          0X516
16 ANG_YAW_30:   .BYTE          (CANT_ANGS<<1)
17 .ORG          0X524
18 ANG_YAW_45:   .BYTE          (CANT_ANGS<<1)
19 .ORG          0X532
20 ANG_YAW_60:   .BYTE          (CANT_ANGS<<1)
21 .ORG          0X540
22 ANG_YAW_75:   .BYTE          (CANT_ANGS<<1)
23 .ORG          0X548
24 ANG_YAW_90:   .BYTE          (CANT_ANGS<<1)
25 .ORG          0X556
26 ANG_YAW_105:  .BYTE          (CANT_ANGS<<1)
27 .ORG          0X564
28 ANG_YAW_120:  .BYTE          (CANT_ANGS<<1)
29 .ORG          0X572
30 ANG_YAW_135:  .BYTE          (CANT_ANGS<<1)
31 .ORG          0X580
32 ANG_YAW_150:  .BYTE          (CANT_ANGS<<1)
33 .ORG          0X588
34 ANG_YAW_165:  .BYTE          (CANT_ANGS<<1)
35 .ORG          0X596
36 ANG_YAW_180:  .BYTE          (CANT_ANGS<<1) ; ANGULOS DE SERVO
    ↪ SUPERIOR DE CADA PATA
37
38 .ORG          0X604
39 ANG_ROLL_0:   .BYTE          (CANT_ANGS<<1)
40 .ORG          0X612
41 ANG_ROLL_30:  .BYTE          (CANT_ANGS<<1)
42 .ORG          0X620
43 ANG_ROLL_45:  .BYTE          (CANT_ANGS<<1)
44 .ORG          0X628
45 ANG_ROLL_60:  .BYTE          (CANT_ANGS<<1)

```

```

46 .ORG          0X636
47 ANG_ROLL_90:      .BYTE          (CANT_ANGS<<1)      ;ANGULOS DE SERVO
    ↪ INFERIOR DE CADA PATA
48 .ORG          0X644
49 ANG_ROLL_120:     .BYTE          (CANT_ANGS<<1)
50 .ORG          0X652
51 ANG_ROLL_150:     .BYTE          (CANT_ANGS<<1)
52 .ORG          0X660
53 ANG_ROLL_180:     .BYTE          (CANT_ANGS<<1)
54
55
56 .CSEG
57
58 INIC_ANGULOS:
59     RCALL INIC_ARR_ANG_YAW_0
60     RCALL INIC_ARR_ANG_YAW_15
61     RCALL INIC_ARR_ANG_YAW_30
62     RCALL INIC_ARR_ANG_YAW_45
63     RCALL INIC_ARR_ANG_YAW_60
64     RCALL INIC_ARR_ANG_YAW_75
65     RCALL INIC_ARR_ANG_YAW_90
66     RCALL INIC_ARR_ANG_YAW_105
67     RCALL INIC_ARR_ANG_YAW_120
68     RCALL INIC_ARR_ANG_YAW_135
69     RCALL INIC_ARR_ANG_YAW_150
70     RCALL INIC_ARR_ANG_YAW_165
71     RCALL INIC_ARR_ANG_YAW_180
72
73     RCALL INIC_ARR_ANG_ROLL_0
74     RCALL INIC_ARR_ANG_ROLL_30
75     RCALL INIC_ARR_ANG_ROLL_45
76     RCALL INIC_ARR_ANG_ROLL_60
77     RCALL INIC_ARR_ANG_ROLL_90
78     RCALL INIC_ARR_ANG_ROLL_120
79     RCALL INIC_ARR_ANG_ROLL_150
80     RCALL INIC_ARR_ANG_ROLL_180
81
82     RET
83
84
85 ;SE DEBE HACER APUNTAR A X AL ARREGLO DEL ANGULO QUE SE BUSCA
86 FIND_ANG_LEG:
87     MOV TEMP,LEG_H
88     LSR TEMP
89     INC      TEMP
90
91     LOOP_FIND_ANG_LEG:
92     LD      ANG_L,X+
93     LD      ANG_H,X+

```

```

94             DEC TEMP
95             BREQ          SEGUIR
96             RJMP LOOP_FIND_ANG_LEG
97
98     SEGUIR:          RET
99
100
101
102
103
104 INIC_ARR_ANG_YAW_0:
105     LDI ZL,LOW(ROM_ANG_YAW_0<<1)
106     LDI ZH,HIGH(ROM_ANG_YAW_0<<1)
107     LDI XL,LOW(ANG_YAW_0)
108     LDI XH,HIGH(ANG_YAW_0)
109     LDI TEMP,(CANT_ANGS<<1)
110
111     RCALL LOOP_ANG
112     RET
113
114 INIC_ARR_ANG_YAW_15:
115     LDI ZL,LOW(ROM_ANG_YAW_15<<1)
116     LDI ZH,HIGH(ROM_ANG_YAW_15<<1)
117     LDI XL,LOW(ANG_YAW_15)
118     LDI XH,HIGH(ANG_YAW_15)
119     LDI TEMP,(CANT_ANGS<<1)
120
121     RCALL LOOP_ANG
122     RET
123
124 INIC_ARR_ANG_YAW_30:
125     LDI ZL,LOW(ROM_ANG_YAW_30<<1)
126     LDI ZH,HIGH(ROM_ANG_YAW_30<<1)
127     LDI XL,LOW(ANG_YAW_30)
128     LDI XH,HIGH(ANG_YAW_30)
129     LDI TEMP,(CANT_ANGS<<1)
130
131     RCALL LOOP_ANG
132     RET
133
134 INIC_ARR_ANG_YAW_45:
135     LDI ZL,LOW(ROM_ANG_YAW_45<<1)
136     LDI ZH,HIGH(ROM_ANG_YAW_45<<1)
137     LDI XL,LOW(ANG_YAW_45)
138     LDI XH,HIGH(ANG_YAW_45)
139     LDI TEMP,(CANT_ANGS<<1)
140
141     RCALL LOOP_ANG
142     RET

```

```

143
144 INIC_ARR_ANG_YAW_60:
145     LDI ZL,LOW(ROM_ANG_YAW_60<<1)
146     LDI ZH,HIGH(ROM_ANG_YAW_60<<1)
147     LDI XL,LOW(ANG_YAW_60)
148     LDI XH,HIGH(ANG_YAW_60)
149     LDI TEMP,(CANT_ANGS<<1)
150
151     RCALL LOOP_ANG
152     RET
153
154 INIC_ARR_ANG_YAW_75:
155     LDI ZL,LOW(ROM_ANG_YAW_75<<1)
156     LDI ZH,HIGH(ROM_ANG_YAW_75<<1)
157     LDI XL,LOW(ANG_YAW_75)
158     LDI XH,HIGH(ANG_YAW_75)
159     LDI TEMP,(CANT_ANGS<<1)
160
161     RCALL LOOP_ANG
162     RET
163
164 INIC_ARR_ANG_YAW_90:
165     LDI ZL,LOW(ROM_ANG_YAW_90<<1)
166     LDI ZH,HIGH(ROM_ANG_YAW_90<<1)
167     LDI XL,LOW(ANG_YAW_90)
168     LDI XH,HIGH(ANG_YAW_90)
169     LDI TEMP,(CANT_ANGS<<1)
170
171     RCALL LOOP_ANG
172     RET
173
174 INIC_ARR_ANG_YAW_105:
175     LDI ZL,LOW(ROM_ANG_YAW_105<<1)
176     LDI ZH,HIGH(ROM_ANG_YAW_105<<1)
177     LDI XL,LOW(ANG_YAW_105)
178     LDI XH,HIGH(ANG_YAW_105)
179     LDI TEMP,(CANT_ANGS<<1)
180
181     RCALL LOOP_ANG
182     RET
183
184 INIC_ARR_ANG_YAW_120:
185     LDI ZL,LOW(ROM_ANG_YAW_120<<1)
186     LDI ZH,HIGH(ROM_ANG_YAW_120<<1)
187     LDI XL,LOW(ANG_YAW_120)
188     LDI XH,HIGH(ANG_YAW_120)
189     LDI TEMP,(CANT_ANGS<<1)
190

```

```

191         RCALL LOOP_ANG
192         RET
193
194 INIC_ARR_ANG_YAW_135:
195         LDI ZL,LOW(ROM_ANG_YAW_135<<1)
196         LDI ZH,HIGH(ROM_ANG_YAW_135<<1)
197         LDI XL,LOW(ANG_YAW_135)
198         LDI XH,HIGH(ANG_YAW_135)
199         LDI TEMP,(CANT_ANGS<<1)
200
201         RCALL LOOP_ANG
202         RET
203
204 INIC_ARR_ANG_YAW_150:
205         LDI ZL,LOW(ROM_ANG_YAW_150<<1)
206         LDI ZH,HIGH(ROM_ANG_YAW_150<<1)
207         LDI XL,LOW(ANG_YAW_150)
208         LDI XH,HIGH(ANG_YAW_150)
209         LDI TEMP,(CANT_ANGS<<1)
210
211         RCALL LOOP_ANG
212         RET
213
214 INIC_ARR_ANG_YAW_165:
215         LDI ZL,LOW(ROM_ANG_YAW_165<<1)
216         LDI ZH,HIGH(ROM_ANG_YAW_165<<1)
217         LDI XL,LOW(ANG_YAW_165)
218         LDI XH,HIGH(ANG_YAW_165)
219         LDI TEMP,(CANT_ANGS<<1)
220
221         RCALL LOOP_ANG
222         RET
223
224 INIC_ARR_ANG_YAW_180:
225         LDI ZL,LOW(ROM_ANG_YAW_180<<1)
226         LDI ZH,HIGH(ROM_ANG_YAW_180<<1)
227         LDI XL,LOW(ANG_YAW_180)
228         LDI XH,HIGH(ANG_YAW_180)
229         LDI TEMP,(CANT_ANGS<<1)
230
231         RCALL LOOP_ANG
232         RET
233
234 INIC_ARR_ANG_ROLL_0:
235         LDI ZL,LOW(ROM_ANG_ROLL_0<<1)
236         LDI ZH,HIGH(ROM_ANG_ROLL_0<<1)
237         LDI XL,LOW(ANG_ROLL_0)
238         LDI XH,HIGH(ANG_ROLL_0)

```

```

239         LDI TEMP, (CANT_ANGS<<1)
240
241     RCALL LOOP_ANG
242     RET
243
244 INIC_ARR_ANG_ROLL_30:
245     LDI ZL, LOW(ROM_ANG_ROLL_30<<1)
246     LDI ZH, HIGH(ROM_ANG_ROLL_30<<1)
247     LDI XL, LOW(ANG_ROLL_30)
248     LDI XH, HIGH(ANG_ROLL_30)
249     LDI TEMP, (CANT_ANGS<<1)
250
251     RCALL LOOP_ANG
252     RET
253
254 INIC_ARR_ANG_ROLL_45:
255     LDI ZL, LOW(ROM_ANG_ROLL_45<<1)
256     LDI ZH, HIGH(ROM_ANG_ROLL_45<<1)
257     LDI XL, LOW(ANG_ROLL_45)
258     LDI XH, HIGH(ANG_ROLL_45)
259     LDI TEMP, (CANT_ANGS<<1)
260
261     RCALL LOOP_ANG
262     RET
263
264 INIC_ARR_ANG_ROLL_60:
265     LDI ZL, LOW(ROM_ANG_ROLL_60<<1)
266     LDI ZH, HIGH(ROM_ANG_ROLL_60<<1)
267     LDI XL, LOW(ANG_ROLL_60)
268     LDI XH, HIGH(ANG_ROLL_60)
269     LDI TEMP, (CANT_ANGS<<1)
270
271     RCALL LOOP_ANG
272     RET
273
274
275 INIC_ARR_ANG_ROLL_90:
276     LDI ZL, LOW(ROM_ANG_ROLL_90<<1)
277     LDI ZH, HIGH(ROM_ANG_ROLL_90<<1)
278     LDI XL, LOW(ANG_ROLL_90)
279     LDI XH, HIGH(ANG_ROLL_90)
280     LDI TEMP, (CANT_ANGS<<1)
281
282     RCALL LOOP_ANG
283     RET
284
285
286 INIC_ARR_ANG_ROLL_120:
287     LDI ZL, LOW(ROM_ANG_ROLL_120<<1)

```

```

288         LDI ZH,HIGH(ROM_ANG_ROLL_120<<1)
289         LDI XL,LOW(ANG_ROLL_120)
290         LDI XH,HIGH(ANG_ROLL_120)
291         LDI TEMP,(CANT_ANGS<<1)
292
293         RCALL LOOP_ANG
294         RET
295
296 INIC_ARR_ANG_ROLL_150:
297         LDI ZL,LOW(ROM_ANG_ROLL_150<<1)
298         LDI ZH,HIGH(ROM_ANG_ROLL_150<<1)
299         LDI XL,LOW(ANG_ROLL_150)
300         LDI XH,HIGH(ANG_ROLL_150)
301         LDI TEMP,(CANT_ANGS<<1)
302
303         RCALL LOOP_ANG
304         RET
305
306 INIC_ARR_ANG_ROLL_180:
307         LDI ZL,LOW(ROM_ANG_ROLL_180<<1)
308         LDI ZH,HIGH(ROM_ANG_ROLL_180<<1)
309         LDI XL,LOW(ANG_ROLL_180)
310         LDI XH,HIGH(ANG_ROLL_180)
311         LDI TEMP,(CANT_ANGS<<1)
312
313         RCALL LOOP_ANG
314         RET
315
316 LOOP_ANG:
317         LPM TEMP,Z+
318         ST X+,TEMP
319         DEC TEMP
320
321         BRNE LOOP_ANG
322
323         RET
324
325
326 SERVO_TEST_ANG:
327         ;LDI          R23,SERVO_TESTER_1
328         LDI          R24,LOW(1650)
329         LDI          R25,HIGH(1650)
330         RCALL        PCA9685_MOVE
331
332
333         LDI          R23,SERVO_TESTER_2
334         LDI          R24,LOW(ANG_TESTER_2)
335         LDI          R25,HIGH(ANG_TESTER_2)
336

```

```

337 ;          RCALL          PCA9685_MOVE
338          RCALL          MOVEMENT_DELAY
339
340          RET
341
342
343 SERVO_TEST_ANG_TOTAL: ;Prueba el servo en 3 pocisiones
344          LDI          R23,SERVO_TESTER
345          LDI          R24,LOW(1800)
346          LDI          R25,HIGH(1800)
347          RCALL          PCA9685_MOVE
348          RCALL          MOVEMENT_DELAY
349
350          LDI          R23,SERVO_TESTER
351          LDI          R24,LOW(1750)
352          LDI          R25,HIGH(1750)
353          RCALL          PCA9685_MOVE
354          RCALL          MOVEMENT_DELAY
355
356          LDI          R23,SERVO_TESTER
357          LDI          R24,LOW(1575)
358          LDI          R25,HIGH(1575)
359          RCALL          PCA9685_MOVE
360          RCALL          MOVEMENT_DELAY
361
362          LDI          R23,SERVO_TESTER
363          LDI          R24,LOW(1400)
364          LDI          R25,HIGH(1400)
365          RCALL          PCA9685_MOVE
366          RCALL          MOVEMENT_DELAY
367
368          LDI          R23,SERVO_TESTER
369          LDI          R24,LOW(1250)
370          LDI          R25,HIGH(1250)
371          RCALL          PCA9685_MOVE
372          RCALL          MOVEMENT_DELAY
373
374          LDI          R23,SERVO_TESTER
375          LDI          R24,LOW(1142)
376          LDI          R25,HIGH(1142)
377          RCALL          PCA9685_MOVE
378          RCALL          MOVEMENT_DELAY
379
380          LDI          R23,SERVO_TESTER
381          LDI          R24,LOW(1034)
382          LDI          R25,HIGH(1034)
383          RCALL          PCA9685_MOVE
384          RCALL          MOVEMENT_DELAY
385

```

```

386         LDI          R23,SERVO_TESTER
387         LDI          R24,LOW(925)
388         LDI          R25,HIGH(925)
389         RCALL         PCA9685_MOVE
390         RCALL         MOVEMENT_DELAY
391
392         LDI          R23,SERVO_TESTER
393         LDI          R24,LOW(816)
394         LDI          R25,HIGH(816)
395         RCALL         PCA9685_MOVE
396         RCALL         MOVEMENT_DELAY
397
398         LDI          R23,SERVO_TESTER
399         LDI          R24,LOW(708)
400         LDI          R25,HIGH(708)
401         RCALL         PCA9685_MOVE
402         RCALL         MOVEMENT_DELAY
403
404         LDI          R23,SERVO_TESTER
405         LDI          R24,LOW(600)
406         LDI          R25,HIGH(600)
407         RCALL         PCA9685_MOVE
408         RCALL         MOVEMENT_DELAY
409
410         RET
411
412
413         ;ARREGLOS DE ANGULOS(EN GRADOS): ANG_X: .DB PATA_1,PATA_2,PATA_3,PATA_4
414         ;PATA:                                     1                2                3
415         .ORG          0X500
416         ROM_ANG_YAW_0: .DW          1750,          800,          1800,          385
417         ROM_ANG_YAW_15: .DW         1750,          800,          1700,          450
418         ROM_ANG_YAW_30: .DW         1750,          800,          1600,          575
419         ROM_ANG_YAW_45: .DW         1750,          800,          1550,          700
420         ROM_ANG_YAW_60: .DW         1575,          925,          1300,          825
421         ROM_ANG_YAW_75: .DW         1400,          1100,          1200,          925
422         ROM_ANG_YAW_90: .DW         1300,          1200,          1100,          1050
423         ROM_ANG_YAW_105: .DW        1140,          1360,          1000,          1200
424         ROM_ANG_YAW_120: .DW        1030,          1490,          850,          1325
425         ROM_ANG_YAW_135: .DW 925,          1600,          750,          1450
426         ROM_ANG_YAW_150: .DW 816,          1750,          750,          1450
427         ROM_ANG_YAW_165: .DW         708,          1870,          750,          1450
428         ROM_ANG_YAW_180: .DW         600,          2000,          750,          1450
429         ⇨ DE SERVO SUPERIOR DE CADA PATA
430         ROM_ANG_ROLL_0: .DW         500,          500,          700,          500
431         ROM_ANG_ROLL_30: .DW         750,          750,          900,          800
432         ROM_ANG_ROLL_45: .DW         850,          850,          1000,          900
433         ROM_ANG_ROLL_60: .DW         950,          950,          1100,          1000

```

```

434 ROM_ANG_ROLL_90:.DW      1200,      1200,      1500,      1400
435 ROM_ANG_ROLL_120:.DW 1450,      1450,      1500,      1400
436 ROM_ANG_ROLL_150:.DW 1700,      1700,      1500,      1400
437 ROM_ANG_ROLL_180:.DW
    ⇨ 2000,      2000,      1500,      1400      ;ANGULOS DE SERVO
    ⇨ INFERIOR DE CADA PATA

```

HC05

```
1  HC05_SETUP:
2      ;SET TRANSMITION SPEED 9600 BAUDS
3      LDI      TEMP, HIGH(16000000/(8*9600) - 1)
4      STS      UBRROH, TEMP
5      LDI      TEMP, LOW(16000000/(8*9600) - 1)
6      STS      UBRROL, TEMP
7
8      ;SET DOUBLE SPEED
9      LDI      TEMP, (1<<U2X0)
10     STS      UCSROA, TEMP
11
12     ;ENABLE RECEPTION AND INTERRUPT
13     LDI      TEMP, (1<<RXCIE0)|(1<<RXEN0)
14     STS      UCSROB, TEMP
15
16     ;SET 8 BIT FORMAT + PAIR BIT
17     LDI      TEMP, (1<<UCSZ01)|(1<<UCSZ00)
18     STS      UCSROC, TEMP
19
20     SEI
21
22     RET
23
24  HC05_RECEPTION_HANDLER:
25      CLI
26      LDS      TEMP, UDRO
27      SUBI     TEMP, '0'
28      STS      GPIORO, TEMP
29      RETI
```

HCSR04

DEFINES

```
1  .DEF                TEMP                = R16
2
3  .def                TIMEL                = R17
4  .def                TIMEH                = R18
5
6  .DEF                LEG_L                = R19
7  .DEF                LEG_H                = R20
8
9  .def                I2C_DATA              = R21
10
11 .DEF                ANG_L                = R24
12 .DEF                ANG_H                = R25
13
14 .DEF                SERVO_NUMBER          = R23
15 .DEF                POSITION_LOW           = R24
16 .DEF                POSITION_HIGH         = R25
```

MACROS

```
1  .macro STACK_INIT          ; setup stack
2      ldi TEMP, HIGH(RAMEND)
3      out SPH, TEMP
4      ldi TEMP, LOW(RAMEND)
5      out SPL, TEMP
6  .endmacro
7
8  .macro stsi                  ;CARGA UN
9      ↪ NUMERO EN UN REGISTRO
10     ldi TEMP, @1
11     sts @0, TEMP
12 .endmacro
13
14 .macro stsi16                ;CARGA UN NUMERO DE
15     ↪ 16 BIT EN 2 REGISTROS
16     ldi TEMP, HIGH(@1)
17     sts @0H, TEMP
18     ldi TEMP, LOW(@1)
19     sts @0L, TEMP
20 .endmacro
21
22 .macro PORTB_AS_OUTPUT ; @0 es la/s patita/s que quiero configurar
23     ↪ (mandame Obxxxxxxxx y x=1 es patita para salida)
24     ldi TEMP, @0
25     out DDRB, TEMP ; seteo la/s patita/s que me pidieron
26     clr TEMP
27     out PORTB, TEMP ; inicializo en 0 el puerto B
28 .endmacro
29
30 .macro PORTC_AS_OUTPUT ; @0 es la/s patita/s que quiero configurar
31     ↪ (mandame Obxxxxxxxx y x=1 es patita para salida)
32     ldi TEMP, @0
33     out DDRC, TEMP ; seteo la/s patita/s que me pidieron
34     clr TEMP
35     out PORTC, TEMP ; inicializo en 0 el puerto C
36 .endmacro
37
38 .macro PORTD_AS_OUTPUT ; @0 es la/s patita/s que quiero configurar
39     ↪ (mandame Obxxxxxxxx y x=1 es patita para salida)
40     ldi TEMP, @0
41     out DDRD, TEMP ; seteo la/s patita/s que me pidieron
42     clr TEMP
43     out PORTD, TEMP ; inicializo en 0 el puerto D
44 .endmacro
```