



Instituto Politécnico Nacional



Escuela Superior de Cómputo

Servo Motor

TAREA 8

Materia:

Introducción a los microcontroladores

Grupo:

3CM16

Profesor:

Pérez Pérez José Juan

Integrantes:

Castro Cruces Jorge Eduardo

Cortes Ramírez Roberto Carlos

Domínguez Acosta José Práxedes

Fecha:

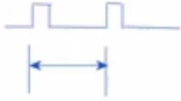
Lunes, 1 de noviembre de 2021

Descripción del problema

Descripción:

Ejercicio servomotor 8 posiciones.


Generación de una señal de tipo PWM para controlar la posición de un servomotor SG90



SG90

TH + TL = 20 ms

TH	TL	posición
1 ms	19.0 ms	0 grados
1.5 ms	18.5 ms	90 grados
2.0 ms	18.0 ms	180 grados



1.25 ms 18.75 ms 45 grados
1.125 ms 18.875 ms 22.5 grados

Unmute Start Video Participants Chat Share Screen Record Reactions Leave

darcy.rsgc.on.ca/ACES/TEI4M/AVRdelay.html

AVR Delay Loop Calculator

Developed originally by [Bret Mulvey](#). Register enhancement by T. Morland. (ACES '18)

MHz microcontroller clock frequency

cycles for `rcall/ret` or other overhead

first register to be used by delay loop

ns us ms s mins hrs days

cycles go

☒ assembler ☐ avr-gcc

```
; Assembly code auto-generated
; by utility from Bret Mulvey
; Delay 136 cycles
; 136us at 1 MHz

    ldi r18, 45
L1:  dec r18
     brne L1
     nop
```

Código del programa

```
1.      .include "m8535def.inc"
2.      .def aux = r16
3.      ldi aux, low(ramend)
4.      out spl, aux
5.      ldi aux, high(ramend)
6.      out sph, aux
7.      ser aux
8.      out ddra, aux
9.      out portb, aux
10.
11.     checa:
12.         sbis pinb, 0                ;0
13.         rcall cero
14.         sbis pinb, 1                ;25.71
15.         rcall veinte5
16.         sbis pinb, 2                ;51.42
17.         rcall cincuenta1
18.         sbis pinb, 3                ;77.14
19.         rcall setenta7
20.         sbis pinb, 4                ;102.85
21.         rcall cien2
22.         sbis pinb, 5                ;128.57
23.         rcall cien28
24.         sbis pinb, 6                ;154.28
25.         rcall cien54
26.         sbis pinb, 7                ;180
27.         rcall cien80
28.         rjmp checa
29.
30.     cero:
31.         sbi porta, 0
32.         rcall medms
33.         rcall medms
34.         rcall medms
35.         rcall medms
36.         rcall medms
37.         rcall medms
38.         rcall medms
39.         cbi porta, 0
40.         ldi aux, 43
41.     cta0:
42.         rcall medms
43.         dec aux
44.         brne cta0
45.         ret
46.
```

```

47.     veinte5:
48.         sbi porta,0
49.         rcall medms
50.         rcall medms
51.         rcall medms
52.         rcall medms
53.         rcall medms
54.         rcall medms
55.         rcall medms
56.         rcall medms
57.         cbi porta,0
58.         ldi aux,42
59.     cta1:
60.         rcall medms
61.         dec aux
62.         brne cta1
63.         ret
64.
65.     cincuenta1:
66.         sbi porta,0
67.         rcall medms
68.         rcall medms
69.         rcall medms
70.         rcall medms
71.         rcall medms
72.         rcall medms
73.         rcall medms
74.         rcall medms
75.         rcall medms
76.         cbi porta,0
77.         ldi aux,41
78.     cta2:
79.         rcall medms
80.         dec aux
81.         brne cta2
82.         ret
83.
84.     setenta7:
85.         sbi porta,0
86.         rcall medms
87.         rcall medms
88.         rcall medms
89.         rcall medms
90.         rcall medms
91.         rcall medms
92.         rcall medms
93.         rcall medms
94.         rcall medms

```

```

95.          rcall medms
96.          cbi porta,0
97.          ldi aux,40
98.  cta3:
99.          rcall medms
100.         dec aux
101.         brne cta3
102.         ret
103.
104.  cien2:
105.         sbi porta,0
106.         rcall medms
107.         rcall medms
108.         rcall medms
109.         rcall medms
110.         rcall medms
111.         rcall medms
112.         rcall medms
113.         rcall medms
114.         rcall medms
115.         rcall medms
116.         rcall medms
117.         cbi porta,0
118.         ldi aux,39
119.  cta4:
120.         rcall medms
121.         dec aux
122.         brne cta4
123.         ret
124.
125.  cien28:
126.         sbi porta,0
127.         rcall medms
128.         rcall medms
129.         rcall medms
130.         rcall medms
131.         rcall medms
132.         rcall medms
133.         rcall medms
134.         rcall medms
135.         rcall medms
136.         rcall medms
137.         rcall medms
138.         rcall medms
139.         cbi porta,0
140.         ldi aux,38
141.  cta5:
142.         rcall medms

```

```

143.          dec aux
144.          brne cta5
145.          ret
146.
147.      cien54:
148.          sbi porta,0
149.          rcall medms
150.          rcall medms
151.          rcall medms
152.          rcall medms
153.          rcall medms
154.          rcall medms
155.          rcall medms
156.          rcall medms
157.          rcall medms
158.          rcall medms
159.          rcall medms
160.          rcall medms
161.          rcall medms
162.          cbi porta,0
163.          ldi aux,37
164.
165.      cta6:
166.          rcall medms
167.          dec aux
168.          brne cta6
169.          ret
170.
171.      cien80:
172.          sbi porta,0
173.          rcall medms
174.          rcall medms
175.          rcall medms
176.          rcall medms
177.          rcall medms
178.          rcall medms
179.          rcall medms
180.          rcall medms
181.          rcall medms
182.          rcall medms
183.          rcall medms
184.          rcall medms
185.          rcall medms
186.          rcall medms
187.          cbi porta,0
188.          ldi aux,36
189.
190.      cta7:

```

```
191.          rcall medms
192.          dec aux
193.          brne cta7
194.          ret
195.
196. medms:
197.          ; Assembly code auto-generated
198.          ; by utility from Bret Mulvey
199.          ; Delay 136 cycles
200.          ; 136us at 1 MHz
201.          ldi r18, 45
202.
203. L1: dec r18
204.          brne L1
205.          nop
206.          ret
207.
```

Simulación en AVR Studio 4

AVR Studio - [C:\Users\georg\Desktop\ESCOM\8vo Semestre\Micros\Tareas\8ServoMotor\AVRstudio\ServoMotor.asm]

File Project Build Edit View Tools Debug Window Help

Trace Disabled

Processor

Name	Value
Program Counter	0x000000
Stack Pointer	0x0000
X pointer	0x0000
Y pointer	0x0000
Z pointer	0x0000
Cycle Counter	0
Frequency	4.0000 MHz
Stop Watch	0.00 us
SREG	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Registers	

```
.include "m8535def.inc"
.def aux = r16
ldi aux, low(ramend)
out spl, aux
ldi aux, high(ramend)
out sph, aux
ser aux
out ddra, aux
out portb, aux

chea:
sbis pinb.0      ;0
rcall cero
sbis pinb.1      ;25.71
rcall veinte5
sbis pinb.2      ;51.42
rcall cincuenta1
sbis pinb.3      ;77.14
rcall setenta7
sbis pinb.4      ;102.85
rcall cien2
sbis pinb.5      ;128.57
rcall cien28
sbis pinb.6      ;154.28
rcall cien54
sbis pinb.7      ;180
rcall cien80
rjmp chea
```

C:\Users\georg\Desktop\ESCOM\8vo Semestre\Micros\Tareas\8ServoMotor\AVRstudio\ServoMotor.asm

Build

ATmega8535 memory use summary [bytes]:

Segment	Begin	End	Code	Data	Used	Size	Use%
[.cseg]	0x000000	0x000152	338	0	338	8192	4.1%
[.dseg]	0x000060	0x000060	0	0	0	512	0.0%
[.eseg]	0x000000	0x000000	0	0	0	512	0.0%

Assembly complete, 0 errors, 0 warnings

I/O View

ANALOG_COMPARA...

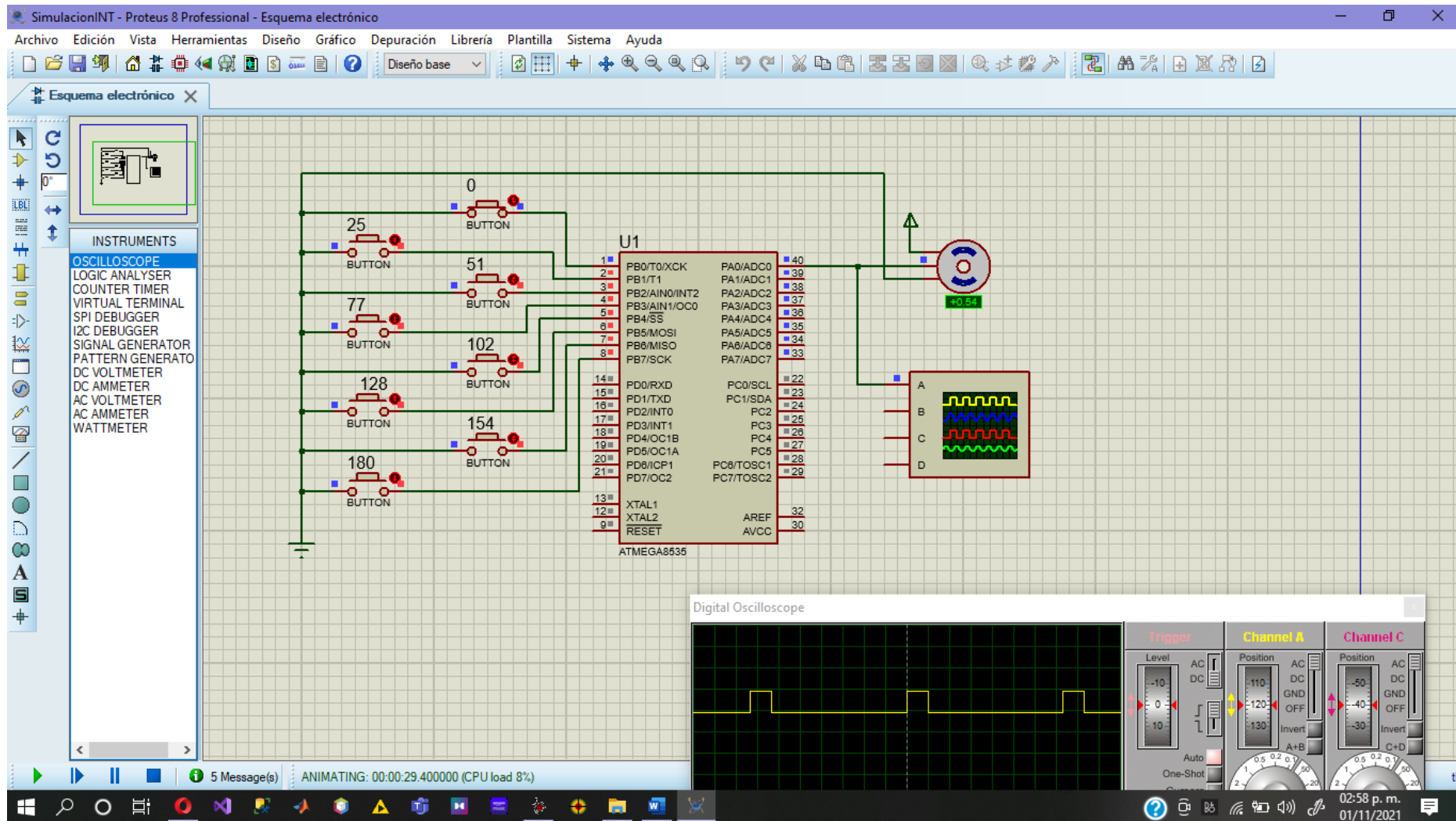
Name	Value
ANALOG_COMPARA...	
CPU	
EEPROM	
EXTERNAL_INTERR...	
PORTA	
PORTB	
PORTC	
PORTD	
SPI	
TIMER_COUNTER_0	
TIMER_COUNTER_1	
TIMER_COUNTER_2	
TWI	
USART	
WATCHDOG	

Name	Address	Value	Bits
ICR1	0x26 (0x46)	0x0000	
OCR1A	0x2A (0x4A)	0x0000	
OCR1B	0x28 (0x48)	0x0000	
TCCR1A	0x2F (0x4F)	0x00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TCCR1B	0x2E (0x4E)	0x00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TCNT1	0x2C (0x4C)	0x0000	
TIFR	0x38 (0x58)	0x00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TIMSK	0x39 (0x59)	0x00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

ATmega8535 AVR Simulator Auto Stopped Ln 10, Col 1 CAP NUM OVR

02:50 p.m. 01/11/2021

Simulación en Proteus 8 Professional



SimulacionINT - Proteus 8 Professional - Esquema electrónico

Archivo Edición Vista Herramientas Diseño Gráfico Depuración Librería Plantilla Sistema Ayuda

Esquema electrónico

INSTRUMENTS

- OSCILLOSCOPE
- LOGIC ANALYSER
- COUNTER TIMER
- VIRTUAL TERMINAL
- SPI DEBUGGER
- I2C DEBUGGER
- SIGNAL GENERATOR
- PATTERN GENERATOR
- DC VOLTMETER
- DC AMMETER
- AC VOLTMETER
- AC AMMETER
- WATTMETER

0 BUTTON

25 BUTTON

51 BUTTON

77 BUTTON

102 BUTTON

128 BUTTON

154 BUTTON

180 BUTTON

U1

1 PB0/T0/XCK	PA0/ADC0	40
2 PB1/T1	PA1/ADC1	39
3 PB2/AIN0/INT2	PA2/ADC2	38
4 PB3/AIN1/OC0	PA3/ADC3	37
5 PB4/SS	PA4/ADC4	36
6 PB5/MOSI	PA5/ADC5	35
7 PB6/MISO	PA6/ADC6	34
8 PB7/SCK	PA7/ADC7	33
14 PD0/RXD	PC0/SCL	22
15 PD1/TXD	PC1/SDA	23
16 PD2/INT0	PC2	24
17 PD3/INT1	PC3	25
18 PD4/OC1B	PC4	26
19 PD5/OC1A	PC5	27
20 PD6/ICP1	PC6/TOSC1	28
21 PD7/OC2	PC7/TOSC2	29
13 XTAL1		
12 XTAL2		
9 RESET	AREF	32
	AVCC	30

ATMEGA8535

20.0

A

B

C

D

Digital Oscilloscope

Trigger

Level

AG DC

Position

AG DC

GND OFF

Invert

A+B

Auto

One-Shot

Channel A

Channel C

Level

AG DC

Position

AG DC

GND OFF

Invert

C+D

Auto

One-Shot

5 Message(s)

ANIMATING: 00:00:42.400000 (CPU load 6%)

02:58 p. m.

01/11/2021

SimulacionINT - Proteus 8 Professional - Esquema electrónico

Archivo Edición Vista Herramientas Diseño Gráfico Depuración Librería Plantilla Sistema Ayuda

Esquema electrónico

INSTRUMENTS

- OSCILLOSCOPE
- LOGIC ANALYSER
- COUNTER TIMER
- VIRTUAL TERMINAL
- SPI DEBUGGER
- I2C DEBUGGER
- SIGNAL GENERATOR
- PATTERN GENERATOR
- DC VOLTMETER
- DC AMMETER
- AC VOLTMETER
- AC AMMETER
- WATTMETER

0 BUTTON

25 BUTTON

51 BUTTON

77 BUTTON

102 BUTTON

128 BUTTON

154 BUTTON

180 BUTTON

U1

1 PB0/T0/XCK	PA0/ADC0	40
2 PB1/T1	PA1/ADC1	39
3 PB2/AIN0/INT2	PA2/ADC2	38
4 PB3/AIN1/OC0	PA3/ADC3	37
5 PB4/SS	PA4/ADC4	36
6 PB5/MOSI	PA5/ADC5	35
7 PB6/MISO	PA6/ADC6	34
8 PB7/SCK	PA7/ADC7	33
14 PD0/RXD	PC0/SCL	22
15 PD1/TXD	PC1/SDA	23
16 PD2/INT0	PC2	24
17 PD3/INT1	PC3	25
18 PD4/OC1B	PC4	26
19 PD5/OC1A	PC5	27
20 PD6/ICP1	PC6/TOSC1	28
21 PD7/OC2	PC7/TOSC2	29
13 XTAL1		
12 XTAL2		
9 RESET		
	AREF	32
	AVCC	30

ATMEGA8535

62.0

0 10 20 30 40 50 60 70 80 90 100

110 120 130

50 40 30

Level AG DC

Position AG DC

Trigger

Channel A

Channel C

5 Message(s) ANIMATING: 00:00:54.100000 (CPU load 6%)

02:59 p. m. 01/11/2021

SimulacionINT - Proteus 8 Professional - Esquema electrónico

Archivo Edición Vista Herramientas Diseño Gráfico Depuración Librería Plantilla Sistema Ayuda

Esquema electrónico

INSTRUMENTS

- OSCILLOSCOPE
- LOGIC ANALYSER
- COUNTER TIMER
- VIRTUAL TERMINAL
- SPI DEBUGGER
- I2C DEBUGGER
- SIGNAL GENERATOR
- PATTERN GENERATOR
- DC VOLTMETER
- DC AMMETER
- AC VOLTMETER
- AC AMMETER
- WATTMETER

0 BUTTON

25 BUTTON

51 BUTTON

77 BUTTON

102 BUTTON

128 BUTTON

154 BUTTON

180 BUTTON

U1

1	PB0/T0/XCK	40	PA0/ADC0
2	PB1/T1	39	PA1/ADC1
3	PB2/AIN0/INT2	38	PA2/ADC2
4	PB3/AIN1/OC0	37	PA3/ADC3
5	PB4/SS	36	PA4/ADC4
6	PB5/MOSI	35	PA5/ADC5
7	PB6/MISO	34	PA6/ADC6
8	PB7/SCK	33	PA7/ADC7
14	PD0/RXD	22	PC0/SCL
15	PD1/TXD	23	PC1/SDA
16	PD2/INT0	24	PC2
17	PD3/INT1	25	PC3
18	PD4/OC1B	26	PC4
19	PD5/OC1A	27	PC5
20	PD6/ICP1	28	PC6/TOSC1
21	PD7/OC2	29	PC7/TOSC2
13	XTAL1	32	AREF
12	XTAL2	30	AVCC
9	RESET		

ATMEGA8535

77.8

A

B

C

D

Digital Oscilloscope

Trigger

Level

AG

DC

Position

AG

DC

GND

OFF

Invert

A+B

Auto

One-Shot

Channel A

Position

AG

DC

GND

OFF

Invert

C+D

Channel C

Position

AG

DC

GND

OFF

Invert

C+D

5 Message(s)

ANIMATING: 00:01:01.750000 (CPU load 7%)

02:59 p. m.

01/11/2021

SimulacionINT - Proteus 8 Professional - Esquema electrónico

Archivo Edición Vista Herramientas Diseño Gráfico Depuración Librería Plantilla Sistema Ayuda

Esquema electrónico

INSTRUMENTS

- OSCILLOSCOPE
- LOGIC ANALYSER
- COUNTER TIMER
- VIRTUAL TERMINAL
- SPI DEBUGGER
- I2C DEBUGGER
- SIGNAL GENERATOR
- PATTERN GENERATOR
- DC VOLTMETER
- DC AMMETER
- AC VOLTMETER
- AC AMMETER
- WATTMETER

0 BUTTON

25 BUTTON

51 BUTTON

77 BUTTON

102 BUTTON

128 BUTTON

154 BUTTON

180 BUTTON

U1

1	PB0/T0/XCK	40	PA0/ADC0
2	PB1/T1	39	PA1/ADC1
3	PB2/AIN0/INT2	38	PA2/ADC2
4	PB3/AIN1/OC0	37	PA3/ADC3
5	PB4/SS	36	PA4/ADC4
6	PB5/MOSI	35	PA5/ADC5
7	PB6/MISO	34	PA6/ADC6
8	PB7/SCK	33	PA7/ADC7
14	PD0/RXD	22	PC0/SCL
15	PD1/TXD	23	PC1/SDA
16	PD2/INT0	24	PC2
17	PD3/INT1	25	PC3
18	PD4/OC1B	26	PC4
19	PD5/OC1A	27	PC5
20	PD6/ICP1	28	PC6/TOSC1
21	PD7/OC2	29	PC7/TOSC2
13	XTAL1	32	AREF
12	XTAL2	30	AVCC
9	RESET		

ATMEGA8535

104

A

B

C

D

Digital Oscilloscope

Trigger

Level

AG

DC

Position

110

120

130

Invert

A+B

Channel A

Channel C

Position

50

40

30

Invert

C+D

5 Message(s)

ANIMATING: 00:01:11.000000 (CPU load 10%)

02:59 p. m.

01/11/2021

SimulacionINT - Proteus 8 Professional - Esquema electrónico

Archivo Edición Vista Herramientas Diseño Gráfico Depuración Librería Plantilla Sistema Ayuda

Esquema electrónico

INSTRUMENTS

- OSCILLOSCOPE
- LOGIC ANALYSER
- COUNTER TIMER
- VIRTUAL TERMINAL
- SPI DEBUGGER
- I2C DEBUGGER
- SIGNAL GENERATOR
- PATTERN GENERATOR
- DC VOLTMETER
- DC AMMETER
- AC VOLTMETER
- AC AMMETER
- WATTMETER

0 BUTTON

25 BUTTON

51 BUTTON

77 BUTTON

102 BUTTON

128 BUTTON

154 BUTTON

180 BUTTON

U1

1	PB0/T0/XCK	40	PA0/ADC0
2	PB1/T1	39	PA1/ADC1
3	PB2/AIN0/INT2	38	PA2/ADC2
4	PB3/AIN1/OC0	37	PA3/ADC3
5	PB4/SS	36	PA4/ADC4
6	PB5/MOSI	35	PA5/ADC5
7	PB6/MISO	34	PA6/ADC6
8	PB7/SCK	33	PA7/ADC7
14	PD0/RXD	22	PC0/SCL
15	PD1/TXD	23	PC1/SDA
16	PD2/INT0	24	PC2
17	PD3/INT1	25	PC3
18	PD4/OC1B	26	PC4
19	PD5/OC1A	27	PC5
20	PD6/ICP1	28	PC6/TOSC1
21	PD7/OC2	29	PC7/TOSC2
13	XTAL1	32	AREF
12	XTAL2	30	AVCC
9	RESET		

ATMEGA8535

+

A

B

C

D

Digital Oscilloscope

Trigger

Level

AG

DC

Position

AG

DC

GND

OFF

Invert

A+B

Auto

One-Shot

Channel A

Channel C

Level

AG

DC

Position

AG

DC

GND

OFF

Invert

C+D

Auto

One-Shot

5 Message(s)

ANIMATING: 00:01:19.800000 (CPU load 8%)

02:59 p. m.

01/11/2021

SimulacionINT - Proteus 8 Professional - Esquema electrónico

Archivo Edición Vista Herramientas Diseño Gráfico Depuración Librería Plantilla Sistema Ayuda

Esquema electrónico

INSTRUMENTS

- OSCILLOSCOPE
- LOGIC ANALYSER
- COUNTER TIMER
- VIRTUAL TERMINAL
- SPI DEBUGGER
- I2C DEBUGGER
- SIGNAL GENERATOR
- PATTERN GENERATOR
- DC VOLTMETER
- DC AMMETER
- AC VOLTMETER
- AC AMMETER
- WATTMETER

0 BUTTON

25 BUTTON

51 BUTTON

77 BUTTON

102 BUTTON

128 BUTTON

154 BUTTON

180 BUTTON

U1

1	PB0/T0/XCK	40	PA0/ADC0
2	PB1/T1	39	PA1/ADC1
3	PB2/AIN0/INT2	38	PA2/ADC2
4	PB3/AIN1/OC0	37	PA3/ADC3
5	PB4/SS	36	PA4/ADC4
6	PB5/MOSI	35	PA5/ADC5
7	PB6/MISO	34	PA6/ADC6
8	PB7/SCK	33	PA7/ADC7
14	PD0/RXD	22	PC0/SCL
15	PD1/TXD	23	PC1/SDA
16	PD2/INT0	24	PC2
17	PD3/INT1	25	PC3
18	PD4/OC1B	26	PC4
19	PD5/OC1A	27	PC5
20	PD6/ICP1	28	PC6/TOSC1
21	PD7/OC2	29	PC7/TOSC2
13	XTAL1	32	AREF
12	XTAL2	30	AVCC
9	RESET		

ATMEGA8535

15V

A

B

C

D

Digital Oscilloscope

Trigger

Level

AG

DC

Position

AG

DC

GND

OFF

Invert

A+B

Channel A

Channel C

Level

AG

DC

Position

AG

DC

GND

OFF

Invert

C+D

5 Message(s)

ANIMATING: 00:01:27.800000 (CPU load 7%)

02:59 p. m.

01/11/2021

Conclusiones

- **Castro Cruces Jorge Eduardo**

En esta práctica se logró el objetivo principal, que era diseñar e implementar el código para controlar un servomotor de 180° con 8 botones para seleccionar 8 posiciones. primero lo que hicimos fue dividir $180/7=25.71^\circ$. Eres después estos grados equivalen a 0.1428 ms, y nos apoyamos de una calculadora AVR, para poder generar el código correspondiente del ciclo.

Para la primera posición se debía llamar 7 veces a la rutina de retraso, para las demás posiciones siguientes se fue sumando uno a las invocaciones de la rutina de retraso.

- **Cortes Ramírez Roberto Carlos**

En esta práctica pudimos hacer un código el cual podemos controlar un servomotor de 180 grados haciendo el uso de 8 botones para escoger 8 posiciones del servomotor establecidas. Para esto tuvimos que hacer la división de los 180 grados entre los 7 espacios que hay entre las 8 posiciones y como resultado nos dio 25.71° (0.1428 ms) que son los grados que hay entre cada posición. Para poder generar este código del ciclo nos apoyamos de una calculadora AVR y aparte más el código que hicimos para poder tener más idea de lo que estábamos haciendo.

Para poder ver cómo funciona usamos proteus y lo colocamos en la primera posición se tiene que llamar siete veces a la rutina de retraso y ya para las demás posiciones siguientes se le tiene sumar 1 a las llamadas de la rutina de retraso.

- **Domínguez Acosta José Práxedes**

Logramos implementar un código que nos ayudara a controlar un servomotor de 180 grados haciendo el uso de 8 botones para escoger 8 posiciones del servomotor establecidas. Para esto tuvimos que hacer la división de los

180 grados entre los 7 espacios que hay entre las 8 posiciones y como resultado nos dio 25.71° (0.1428 ms) que son los grados que hay entre cada posición. Para poder generar este código del ciclo nos apoyamos de una calculadora

AVR.

Para colocarlo en la primera posición se tiene que llamar siete veces a la rutina de retraso y ya para las demás posiciones siguientes se le tiene sumar 1 a las invocaciones de la rutina de retraso.