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.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

**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
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
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**
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**
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**
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**
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**
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**
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
165. **cta6:**
166. rcall medms
167. **dec** aux
168. brne cta6
169. **ret**
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
190. **cta7:**
191. rcall medms
192. **dec** aux
193. brne cta7
194. **ret**
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
203. L1: **dec** r18
204. brne L1
205. **nop**
206. **ret**

**Simulación en AVR Studio 4**

Una captura de pantalla de una computadora

Descripción generada automáticamente

**Simulación en Proteus 8 Professional**

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**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°. 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 los180 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 calculadoraAVR.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.