Instituto

Politécnico

Nacional

Escuela Superior de Cómputo

TecDig\_fin

Proyecto Final

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:

Miércoles, 22 de diciembre de 2021

**Descripción del problema**

Escribe un programa para el circuito dado en proteus para que relice lo siguiente:

Al inicio deberá mostrarse “ - - - - - - - -“ en los 8 displays, al pulsar algún botón, deberá mostrarse la información correspondiente en el display más a la derecha, de la siguiente forma en el caso de presionar los botones en el siguiente orden: “2,6,8,9,-“.

“ - - - - - - - -“

“ - - - - - - - 2“

“ - - - - - - 2 6“

“ - - - - - 2 6 8“

“ - - - - 2 6 8 9“

“ - - - 2 6 8 9 -“

El programa puede ser .asm ó .C

Entregar a más tardar el 22 de dic. Por equipo o individual.

[**https://www.dropbox.com/request/CcPIG1EtxRUvorgfOiLt**](https://www.dropbox.com/request/CcPIG1EtxRUvorgfOiLt)

Esquemático

Descripción generada automáticamente

**Código del programa**

1. .include"m8535def.inc"
2. .def aux =r16
3. .def col=r17
4. .def tec = r19
5. **.def tecf = r20**
6. .**equ** G = **$**40*; gui?n =$40 G=$7d*
7. .**equ** A0 = **$**3F
8. .**equ** A1 = **$**06
9. .**equ** A2 = **$**5B
10. **.equ A3 = $4F**
11. .**equ** A4 = **$**66
12. .**equ** A5 = **$**6D
13. .**equ** A6 = **$**7D
14. .**equ** A7 = **$**07
15. **.equ A8 = $7F**
16. .**equ** A9 = **$**67
17. .**equ** SUM = **$**70
18. .**equ** RES = **$**40
19. .**equ** MULT = **$**76
20. **.equ DIV = $64**
21. .**equ** ONC = **$**77
22. .**equ** **EQU** = **$**49

25. **.macro ldb**
26. ldi aux,@1
27. **mov** @0, aux
28. .endm
29. .macro mensaje
30. ***; ldb r9,@0***
31. *; ldb r8,@1*
32. ldb r7,@0
33. ldb r6,@1
34. ldb r5,@2
35. **ldb r4,@3**
36. ldb r3,@4
37. ldb r2,@5
38. ldb r1,@6
39. ldb r0,@7
40. **.endm**
42. reset:
43. rjmp main *; vector de reset*
44. rjmp mueve*;verctor INT0*
45. **rjmp borra*; vector INT1***
46. .org **$**009
47. rjmp barre*;vector timer0*
48. main:
49. ldi aux,low**(**ramend**)**
50. **out spl,aux**
51. ldi aux,high**(**ramend**)**
52. **out** sph,aux
53. rcall config\_io
54. rcall texto0
55. **clr zh**
56. clr zl
57. ldi col,1
58. **out** portc,col
59. ld aux,z
60. **out porta,aux**
61. uno:**nop**
62. **nop**
63. **nop**
64. rjmp uno
66. config\_io:
67. ser aux
68. **out** ddra,aux
69. **out** portb,aux
70. **out ddrc,aux**
71. **out** portd,aux
72. ldi aux,3
73. **out** tccr0,aux*; preescala ck/64*
74. *; ldi aux,3*
75. ***; out tccr1b,aux***
76. ldi aux,**$**01*; 0000 0001*
77. **out** timsk,aux*; toie0*
78. ldi r18,193*; para contar 63 4ms*
79. ldi aux,**$**03*; 0000 0011*
80. **out mcucr,aux**
81. ldi aux,**$**C0*; 1100 0000*
82. **out** gicr,aux*; habilito INT0 e INT1*
83. sei
84. **ret**
86. texto0:
87. mensaje G,G,G,G,G,G,G,G
88. **ret**
89. barre:
90. **out tcnt0,r18**
91. **out** porta,zh
92. **inc** zl
93. **lsl** col
94. brne dos*; si z = 0*
95. **ldi col,1**
96. clr zl
97. dos:
98. **out** portc,col
99. ld aux,z
100. **out porta,aux**
101. reti
103. mueve:
104. **mov** **r8**,r7
105. **mov r7,r6**
106. **mov** r6,r5
107. **mov** r5,r4
108. **mov** r4,r3
109. **mov** r3,r2
110. **mov r2,r1**
111. **mov** r1,r0
112. **in** tec,pinb
113. cpi tec,**$**77
114. breq K7
115. **cpi tec,$7B**
116. breq K4
117. cpi tec,**$**7D
118. breq K1
119. cpi tec,**$**7E
120. **breq KB**
121. cpi tec,**$**B7
122. breq K8
123. cpi tec,**$**BB
124. breq K5
125. **cpi tec,$BD**
126. breq K2
127. cpi tec,**$**BE
128. breq K0
129. cpi tec,**$**D7
130. **breq K9**
131. cpi tec,**$**DB
132. breq K6
133. cpi tec,**$**DD
134. breq K3
135. **cpi tec,$DE**
136. breq KEQU
137. cpi tec,**$**E7
138. breq KA
139. cpi tec,**$**ED
140. **breq KRES**
141. cpi tec,**$**EE
142. breq KSUM
143. cpi tec, **$**EB
144. breq KMUL
145. **suelta:**
146. **in** tecf,pinb
147. cp tecf,tec
148. breq suelta
149. reti
150. **K0: ldb r0,A0**
151. rjmp suelta
152. K1: ldb r0,A1
153. rjmp suelta
154. K2: ldb r0,A2
155. **rjmp suelta**
156. K3: ldb r0,A3
157. rjmp suelta
158. K4: ldb r0,A4
159. rjmp suelta
160. **K5: ldb r0,A5**
161. rjmp suelta
162. K6: ldb r0,A6
163. rjmp suelta
164. K7: ldb r0,A7
165. **rjmp suelta**
166. K8: ldb r0,A8
167. rjmp suelta
168. K9: ldb r0,A9
169. rjmp suelta
170. **KA: ldb r0,DIV**
171. rjmp suelta
172. KB: ldb r0,ONC
173. rjmp suelta
174. KSUM:ldb r0,SUM
175. **rjmp suelta**
176. KRES:ldb r0,RES
177. rjmp suelta
178. KMUL:ldb r0,MULT
179. rjmp suelta
180. **KEQU:ldb r0,EQU**
181. rjmp suelta
183. borra:
184. *;rutina que borra ultima tecla*
185. **ldb r9,G**
186. **mov** r0,r1
187. **mov** r1,r2
188. **mov** r2,r3
189. **mov** r3,r4
190. **mov r4,r5**
191. **mov** r5,r6
192. **mov** r6,r7
193. **mov** r7,**r8**
194. **mov** **r8**,**r9**
195. **reti**

**Simulación en AVR Studio 4**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Simulación en Proteus 8 Professional**

Diagrama, Esquemático

Descripción generada automáticamente

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja

**Conclusiones**

* **Castro Cruces Jorge Eduardo**

En este proyecto final, llevamos a cabo la codificación del teclado matricial de 4x4 tipo calculadora, con el fin de poder visualizar los valores establecidos en el display. Esperamos que con este proyecto final logremos reflejar los conocimientos adquiridos durante todo el semestre, y querer agradecer por todo lo enseñado.

* **Cortes Ramírez Roberto Carlos**

En este proyecto se logró el objetivo principal, el cual era diseñar y simular un programa e lenguaje ensamblador que permita capturar la pulsación de un botón del teclado matricial 4x4, para poder visualizarlo en el arreglo de displays de 7 segmentos.

* **Domínguez Acosta José Práxedes**

En este proyecto se utilizó un código en ensamblador para poder usar el teclado matricial e ingresar datos con él y después mostrando sus valores con una pantalla LCD en el orden y forma en que se propuso para comprender su funcionamiento y saber en qué otras formas podremos usarlas en un futuro.