 Instituto Politécnico Nacional

Escuela Superior de Computo

Materia:

Introducción a los microcontroladores.

Profesor:

Sanchez Aguilar Fernando

Alumnos:

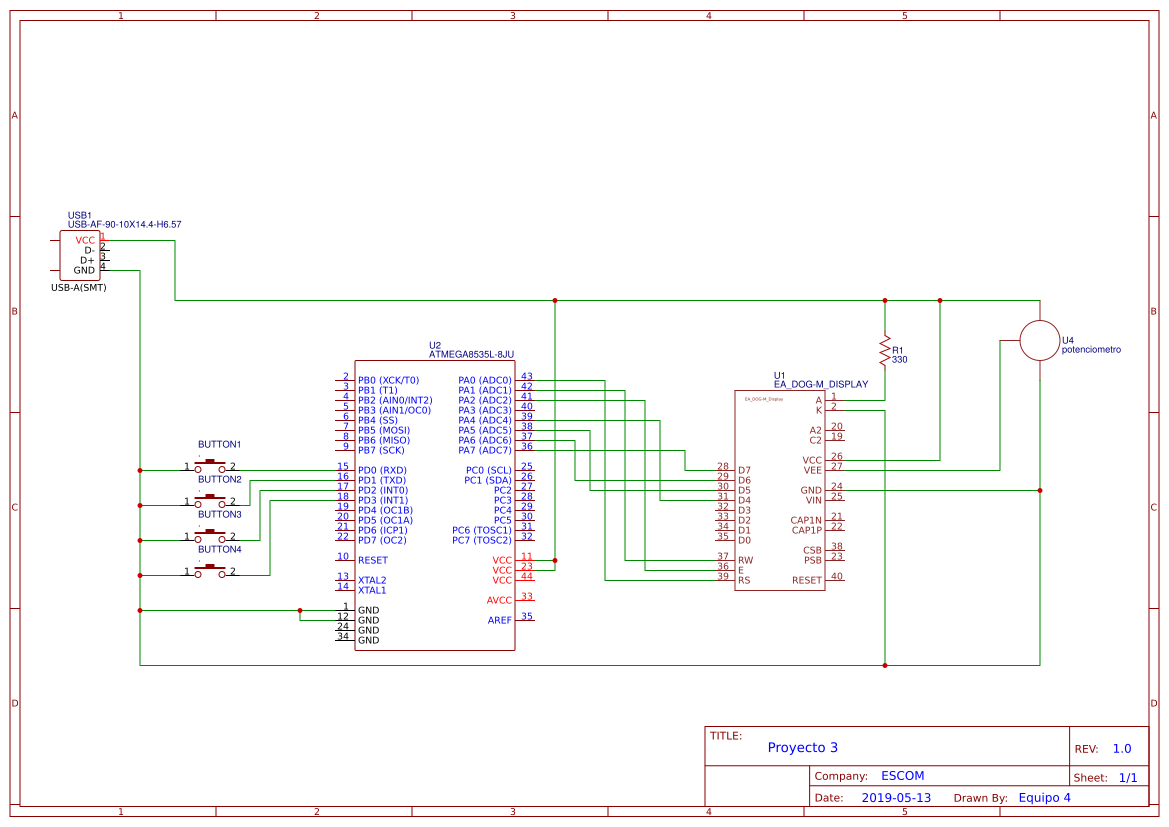
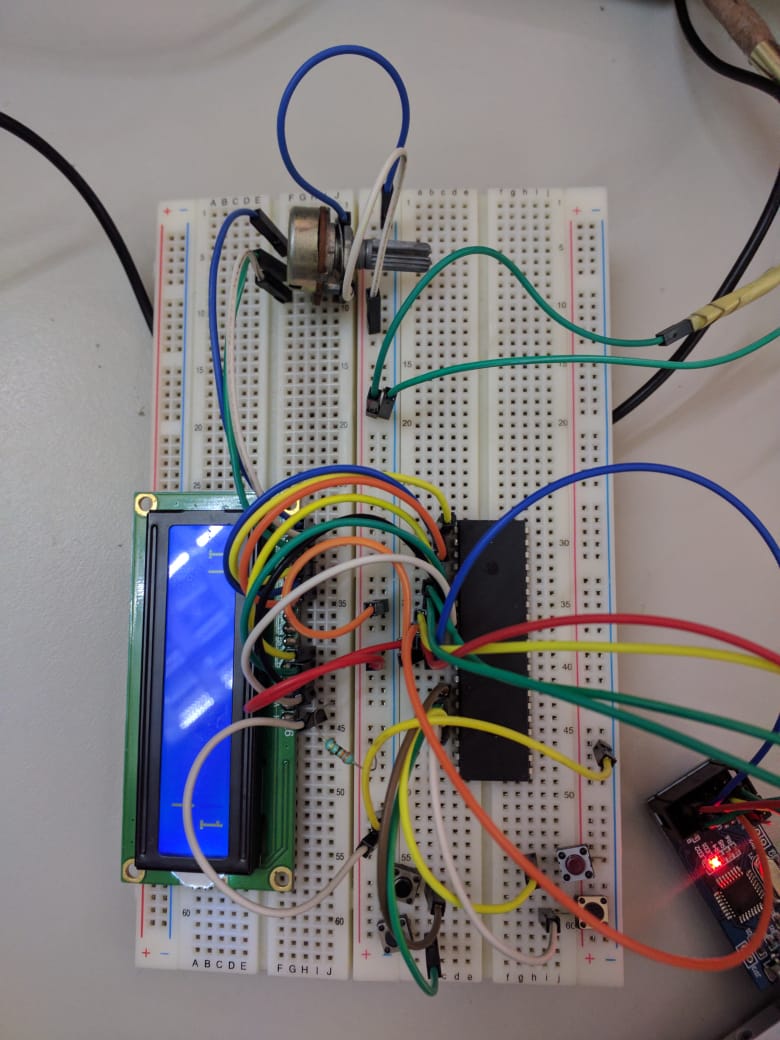
Aldavera Gallaga Iván

Lara Soto Rubén Jair

Morales Castellanos Adolfo Erik

Proyecto 3

Ping pong



1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. This program was created by the
3. CodeWizardAVR V2.60 Evaluation
4. Automatic Program Generator
5. © Copyright 1998-2012 Pavel Haiduc, HP InfoTech s.r.l.
6. http://www.hpinfotech.com
8. Project :
9. Version :
10. Date    : 16/04/2019
11. Author  : Equipo 4
12. Company :
13. Comments:

16. Chip type               : ATmega8535L
17. Program type            : Application
18. AVR Core Clock frequency: 1,000000 MHz
19. Memory model            : Small
20. External RAM size       : 0
21. Data Stack size         : 128
22. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
24. #include <mega8535.h>
26. #include <delay.h>
27. #include <stdlib.h>
28. // Alphanumeric LCD functions
29. #include <alcd.h>


33. #define movi1 PIND.0
34. #define movd1 PIND.1
35. #define movi2 PIND.2
36. #define movd2 PIND.3
38. **int** num=0;
39. **int** portero;
40. unsigned **char** jug1=0,jug2=0;
41. **const** **char** car=48; //codigo ascii
43. // Declare your global variables here
45. #define ADC\_VREF\_TYPE ((0<<REFS1) | (1<<REFS0) | (1<<ADLAR))
47. // Read the 8 most significant bits
48. // of the AD conversion result

51. **void** main(**void**)
52. {
53. // Declare your local variables here
55. // Input/Output Ports initialization
56. // Port A initialization
57. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
58. DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
59. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
60. PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);
62. // Port B initialization
63. // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
64. DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
65. // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
66. PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);
68. // Port C initialization
69. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
70. DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);
71. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
72. PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);
74. // Port D initialization
75. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
76. DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);
77. // State: Bit7=P Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
78. PORTD=(1<<PORTD7) | (1<<PORTD6) | (1<<PORTD5) | (1<<PORTD4) | (1<<PORTD3) | (1<<PORTD2) | (1<<PORTD1) | (1<<PORTD0);
80. // Timer/Counter 0 initialization
81. // Clock source: System Clock
82. // Clock value: Timer 0 Stopped
83. // Mode: Normal top=0xFF
84. // OC0 output: Disconnected
85. TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
86. TCNT0=0x00;
87. OCR0=0x00;
89. // Timer/Counter 1 initialization
90. // Clock source: System Clock
91. // Clock value: Timer1 Stopped
92. // Mode: Normal top=0xFFFF
93. // OC1A output: Disconnected
94. // OC1B output: Disconnected
95. // Noise Canceler: Off
96. // Input Capture on Falling Edge
97. // Timer1 Overflow Interrupt: Off
98. // Input Capture Interrupt: Off
99. // Compare A Match Interrupt: Off
100. // Compare B Match Interrupt: Off
101. TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
102. TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
103. TCNT1H=0x00;
104. TCNT1L=0x00;
105. ICR1H=0x00;
106. ICR1L=0x00;
107. OCR1AH=0x00;
108. OCR1AL=0x00;
109. OCR1BH=0x00;
110. OCR1BL=0x00;
112. // Timer/Counter 2 initialization
113. // Clock source: System Clock
114. // Clock value: Timer2 Stopped
115. // Mode: Normal top=0xFF
116. // OC2 output: Disconnected
117. ASSR=0<<AS2;
118. TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<CS21) | (0<<CS20);
119. TCNT2=0x00;
120. OCR2=0x00;
122. // Timer(s)/Counter(s) Interrupt(s) initialization
123. TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);
125. // External Interrupt(s) initialization
126. // INT0: Off
127. // INT1: Off
128. // INT2: Off
129. MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
130. MCUCSR=(0<<ISC2);
132. // USART initialization
133. // USART disabled
134. UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
136. // Analog Comparator initialization
137. // Analog Comparator: Off
138. ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
140. // ADC initialization
141. // ADC Clock frequency: 500,000 kHz
142. // ADC Voltage Reference: AVCC pin
143. // ADC High Speed Mode: Off
144. // ADC Auto Trigger Source: ADC Stopped
145. // Only the 8 most significant bits of
146. // the AD conversion result are used
147. ADMUX=ADC\_VREF\_TYPE;
148. ADCSRA=(1<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) | (1<<ADPS0);
149. SFIOR=(1<<ADHSM) | (0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);
151. // SPI initialization
152. // SPI disabled
153. SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);
155. // TWI initialization
156. // TWI disabled
157. TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
159. // Alphanumeric LCD initialization
160. // Connections are specified in the
161. // Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
162. // RS - PORTB Bit 0
163. // RD - PORTB Bit 1
164. // EN - PORTB Bit 2
165. // D4 - PORTB Bit 4
166. // D5 - PORTB Bit 5
167. // D6 - PORTB Bit 6
168. // D7 - PORTB Bit 7
169. // Characters/line: 16
170. lcd\_init(16);
172. **while** (1)
173. {
174. jug1=0, jug2=0;
175. lcd\_clear();
176. lcd\_gotoxy(4,0);
177. lcd\_putsf("PING-PONG");
178. lcd\_gotoxy(6,1);
179. lcd\_putsf("FAS");
180. delay\_ms(1500);
181. lcd\_clear();
183. lcd\_gotoxy(1,0);
184. lcd\_putsf("INSTRUCCIONES");
185. delay\_ms(1000);
186. lcd\_clear();
188. lcd\_gotoxy(0,0);
189. lcd\_putsf("1.Gana quien");
190. lcd\_gotoxy(0,1);
191. lcd\_putsf("tenga 9 puntos");
192. delay\_ms(1800);
193. lcd\_clear();
195. lcd\_gotoxy(0,0);
196. lcd\_putsf("2.Diviertanse");
197. delay\_ms(1800);
198. lcd\_clear();
200. lcd\_gotoxy(2,0);
201. lcd\_putsf("Creado por:");
202. delay\_ms(1500);
203. lcd\_clear();
205. lcd\_gotoxy(0,0);
206. lcd\_putsf("Ruben Lara");
207. delay\_ms(1500);
208. lcd\_clear();
210. lcd\_gotoxy(0,0);
211. lcd\_putsf("Erik Morales");
212. delay\_ms(1500);
213. lcd\_clear();
215. lcd\_gotoxy(0,0);
216. lcd\_putsf("Ivan Aldavera");
217. delay\_ms(1500);
218. lcd\_clear();
220. lcd\_gotoxy(0,0);
221. lcd\_putsf("Comienza");
222. delay\_ms(1500);
223. lcd\_clear();
225. **do**{
226. **int** i,x=0,y=1;
227. lcd\_gotoxy(0,0);
228. lcd\_putchar(jug1+car);
229. lcd\_gotoxy(15,0);
230. lcd\_putchar(jug2+car);
232. lcd\_gotoxy(5,0);
234. num++;
235. **if**(num==4){
236. num=0;
237. }
239. **if**(num==0){
240. **for**(i=13; i>=2; i--){
242. **if**(movi1==0){
243. lcd\_gotoxy(1,1);
244. lcd\_putchar(32);
245. portero=1;
246. delay\_ms(300);
247. lcd\_gotoxy(1,portero);
248. lcd\_putchar(124);
249. }
251. **else**{
252. lcd\_gotoxy(1,0);
253. lcd\_putchar(32);
254. portero=0;
255. delay\_ms(300);
256. lcd\_gotoxy(1,portero);
257. lcd\_putchar(124);
258. }


262. lcd\_gotoxy(i,x);
263. lcd\_putchar(46);
264. delay\_ms(200);
265. lcd\_gotoxy(i,x);
266. lcd\_putchar(32);
267. i--;
268. lcd\_gotoxy(i,y);
269. lcd\_putchar(46);
270. delay\_ms(200);
271. lcd\_gotoxy(i,y);
272. lcd\_putchar(32);
273. **if**(i==2 && portero==0){
274. jug2++;
275. }
276. }
277. }


281. **else** **if**(num==1){
282. **for**(i=2; i<14; i++){
284. **if**(movi2==0){
285. lcd\_gotoxy(14,1);
286. lcd\_putchar(32);
287. portero=1;
288. delay\_ms(300);
289. lcd\_gotoxy(14,portero);
290. lcd\_putchar(124);
291. }
293. **else**{
294. lcd\_gotoxy(14,0);
295. lcd\_putchar(32);
296. portero=0;
297. delay\_ms(300);
298. lcd\_gotoxy(14,portero);
299. lcd\_putchar(124);
300. }
302. lcd\_gotoxy(i,x);
303. lcd\_putchar(46);
304. delay\_ms(200);
305. lcd\_gotoxy(i,x);
306. lcd\_putchar(32);
307. i++;
308. lcd\_gotoxy(i,y);
309. lcd\_putchar(46);
310. delay\_ms(200);
311. lcd\_gotoxy(i,y);
312. lcd\_putchar(32);
314. **if**(i==13 && portero==0){
315. jug1++;
316. }
317. }
318. }


322. **else** **if**(num==2){
324. **for**(i=13; i>=2; i--){
325. **if**(movd1==0){
326. lcd\_gotoxy(1,0);
327. lcd\_putchar(32);
328. portero=0;
329. delay\_ms(300);
330. lcd\_gotoxy(1,portero);
331. lcd\_putchar(124);
332. }
334. **else**{
335. lcd\_gotoxy(1,1);
336. lcd\_putchar(32);
337. portero=1;
338. delay\_ms(300);
339. lcd\_gotoxy(1,portero);
340. lcd\_putchar(124);
341. }
343. lcd\_gotoxy(i,y);
344. lcd\_putchar(46);
345. delay\_ms(200);
346. lcd\_gotoxy(i,y);
347. lcd\_putchar(32);
348. i--;
349. lcd\_gotoxy(i,x);
350. lcd\_putchar(46);
351. delay\_ms(200);
352. lcd\_gotoxy(i,x);
353. lcd\_putchar(32);
355. **if**(i==2 && portero==1){
356. jug2++;
357. }
358. }
359. }


363. **else** **if**(num==3){
364. **for**(i=2; i<=13; i++){
365. **if**(movd2==0){
366. lcd\_gotoxy(14,0);
367. lcd\_putchar(32);
368. portero=0;
369. delay\_ms(300);
370. lcd\_gotoxy(14,portero);
371. lcd\_putchar(124);
372. }
374. **else**{
375. lcd\_gotoxy(14,1);
376. lcd\_putchar(32);
377. portero=1;
378. delay\_ms(300);
379. lcd\_gotoxy(14,portero);
380. lcd\_putchar(124);
381. }
383. lcd\_gotoxy(i,y);
384. lcd\_putchar(46);
385. delay\_ms(200);
386. lcd\_gotoxy(i,y);
387. lcd\_putchar(32);
388. i++;
389. lcd\_gotoxy(i,x);
390. lcd\_putchar(46);
391. delay\_ms(200);
392. lcd\_gotoxy(i,x);
393. lcd\_putchar(32);
394. **if**(i==13 && portero==1){
395. jug1++;
396. }
397. }
398. }
400. }**while**(jug2<=9 && jug1<=9);
402. }
403. }