 Instituto Politécnico Nacional

Escuela Superior de Computo

Materia:

Introducción a los microcontroladores.

Profesor:

Sanchez Aguilar Fernando

Alumnos:

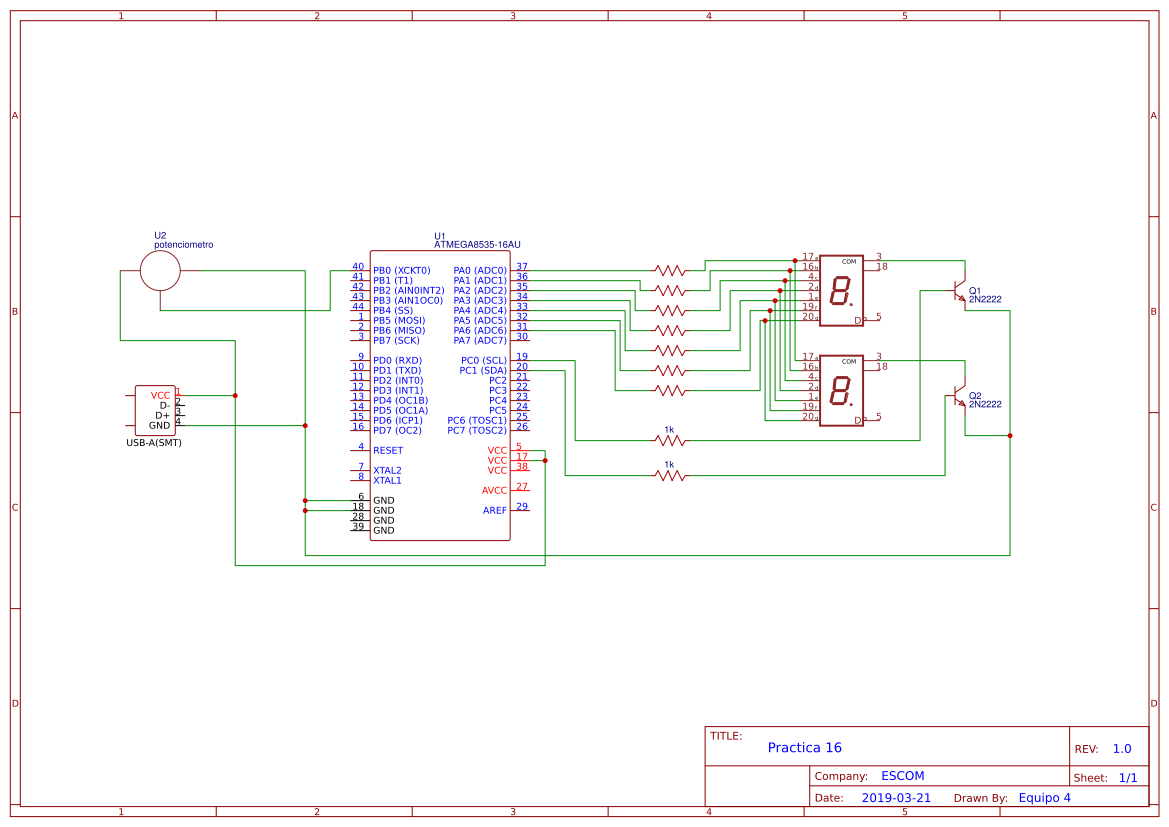
Aldavera Gallaga Iván

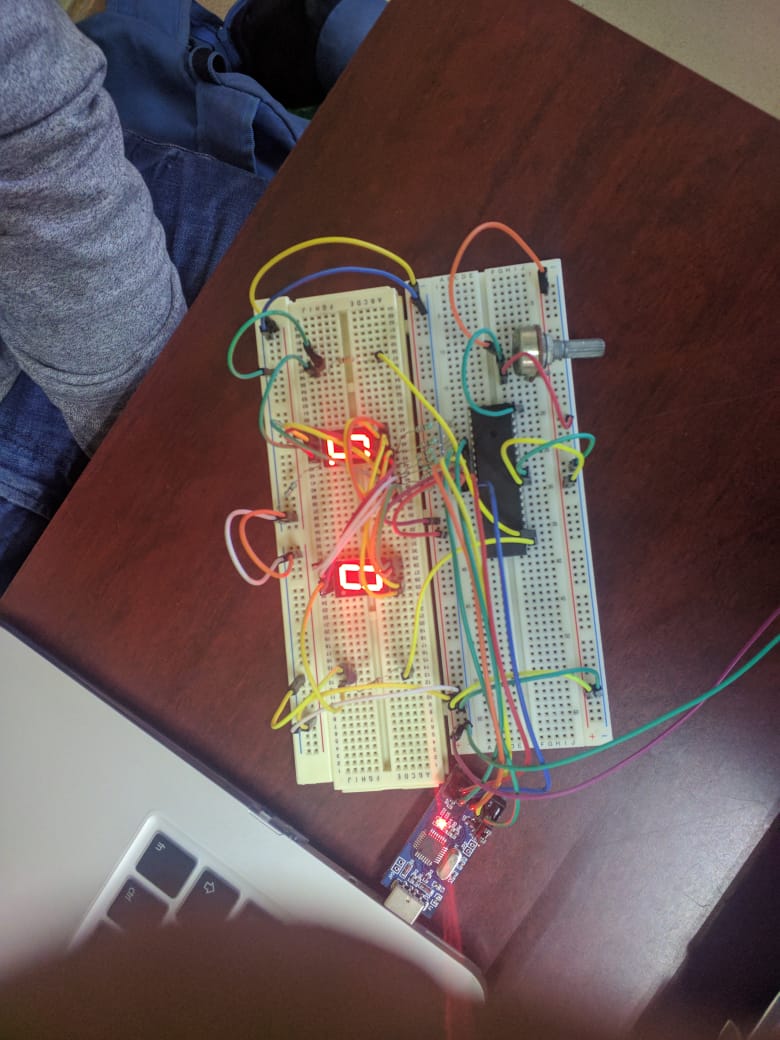
Lara Soto Rubén Jair

Morales Castellanos Adolfo Erik

Practica N°15

Volmetro





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    : 04/03/2019
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>
28. // Declare your global variables here
30. #define ADC\_VREF\_TYPE ((0<<REFS1) | (1<<REFS0) | (1<<ADLAR))
32. // Read the 8 most significant bits
33. // of the AD conversion result

36. **const** **char** tab7seg[10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7c,0x07,0x7f,0x6f};
37. #define C0 PORTC.0
38. #define C1 PORTC.1
40. unsigned **char** cn;
41. unsigned **char** decenas;
42. unsigned **char** unidades;
44. unsigned **char** read\_adc(unsigned **char** adc\_input)
45. {
46. ADMUX=adc\_input | ADC\_VREF\_TYPE;
47. // Delay needed for the stabilization of the ADC input voltage
48. delay\_us(10);
49. // Start the AD conversion
50. ADCSRA|=(1<<ADSC);
51. // Wait for the AD conversion to complete
52. **while** ((ADCSRA & (1<<ADIF))==0);
53. ADCSRA|=(1<<ADIF);
54. **return** ADCH;
55. }
57. **void** main(**void**)
58. {
59. // Declare your local variables here
61. // Input/Output Ports initialization
62. // Port A initialization
63. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
64. DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
65. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
66. PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);
68. // Port B initialization
69. // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
70. DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
71. // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
72. PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);
74. // Port C initialization
75. // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
76. DDRC=(1<<DDC7) | (1<<DDC6) | (1<<DDC5) | (1<<DDC4) | (1<<DDC3) | (1<<DDC2) | (1<<DDC1) | (1<<DDC0);
77. // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
78. PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);
80. // Port D initialization
81. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
82. DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);
83. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
84. PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);
86. // Timer/Counter 0 initialization
87. // Clock source: System Clock
88. // Clock value: Timer 0 Stopped
89. // Mode: Normal top=0xFF
90. // OC0 output: Disconnected
91. TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
92. TCNT0=0x00;
93. OCR0=0x00;
95. // Timer/Counter 1 initialization
96. // Clock source: System Clock
97. // Clock value: Timer1 Stopped
98. // Mode: Normal top=0xFFFF
99. // OC1A output: Disconnected
100. // OC1B output: Disconnected
101. // Noise Canceler: Off
102. // Input Capture on Falling Edge
103. // Timer1 Overflow Interrupt: Off
104. // Input Capture Interrupt: Off
105. // Compare A Match Interrupt: Off
106. // Compare B Match Interrupt: Off
107. TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
108. TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
109. TCNT1H=0x00;
110. TCNT1L=0x00;
111. ICR1H=0x00;
112. ICR1L=0x00;
113. OCR1AH=0x00;
114. OCR1AL=0x00;
115. OCR1BH=0x00;
116. OCR1BL=0x00;
118. // Timer/Counter 2 initialization
119. // Clock source: System Clock
120. // Clock value: Timer2 Stopped
121. // Mode: Normal top=0xFF
122. // OC2 output: Disconnected
123. ASSR=0<<AS2;
124. TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<CS21) | (0<<CS20);
125. TCNT2=0x00;
126. OCR2=0x00;
128. // Timer(s)/Counter(s) Interrupt(s) initialization
129. TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);
131. // External Interrupt(s) initialization
132. // INT0: Off
133. // INT1: Off
134. // INT2: Off
135. MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
136. MCUCSR=(0<<ISC2);
138. // USART initialization
139. // USART disabled
140. UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
142. // Analog Comparator initialization
143. // Analog Comparator: Off
144. ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
146. // ADC initialization
147. // ADC Clock frequency: 500,000 kHz
148. // ADC Voltage Reference: AVCC pin
149. // ADC High Speed Mode: Off
150. // ADC Auto Trigger Source: ADC Stopped
151. // Only the 8 most significant bits of
152. // the AD conversion result are used
153. ADMUX=ADC\_VREF\_TYPE;
154. ADCSRA=(1<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) | (1<<ADPS0);
155. SFIOR=(1<<ADHSM) | (0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);
157. // SPI initialization
158. // SPI disabled
159. SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);
161. // TWI initialization
162. // TWI disabled
163. TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
165. **while** (1)
166. {
167. cn=(read\_adc(0)\*50)/255;
168. decenas=cn/10;
169. unidades=cn%10;
171. PORTB=tab7seg[decenas];
172. C0=0;
173. C1=1;
174. delay\_ms(1);

177. PORTB=tab7seg[unidades];
178. C0=1;
179. C1=0;
180. delay\_ms(1);
182. }
183. }