 Instituto Politécnico Nacional

Escuela Superior de Computo

Materia:

Introducción a los microcontroladores.

Profesor:

Sanchez Aguilar Fernando

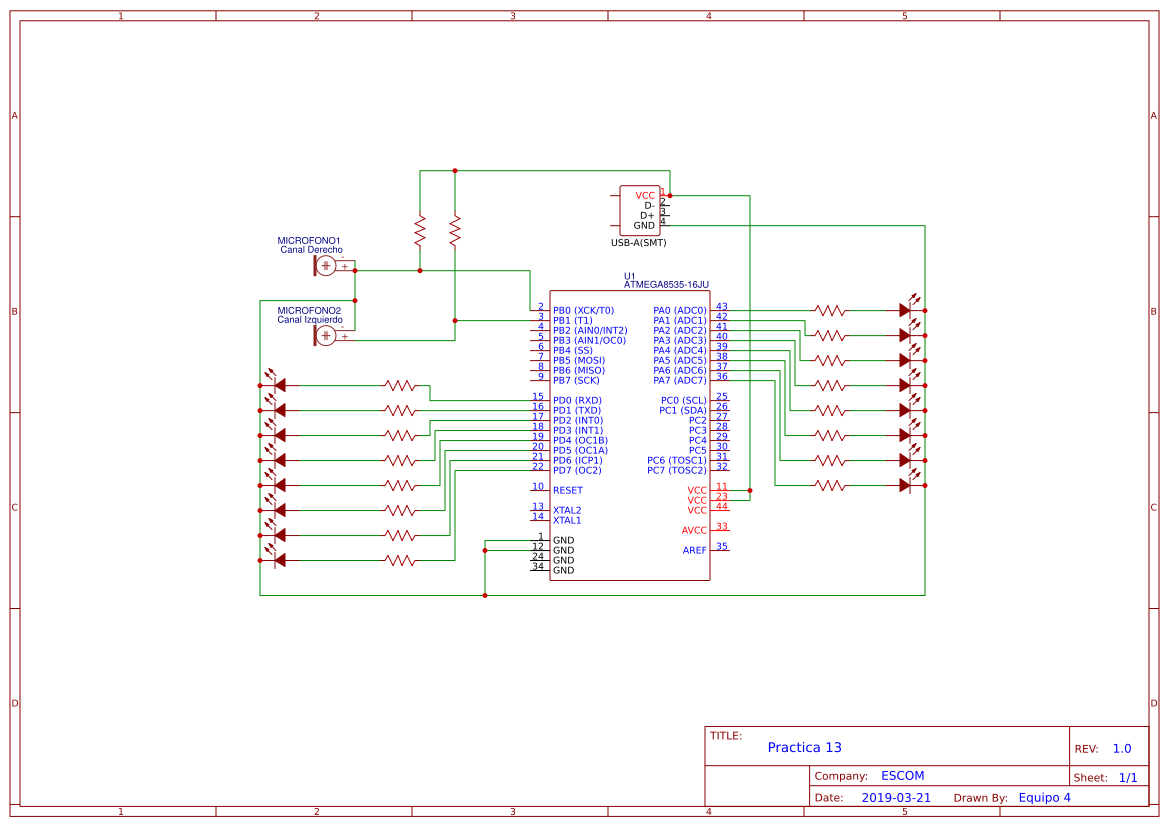
Alumnos:

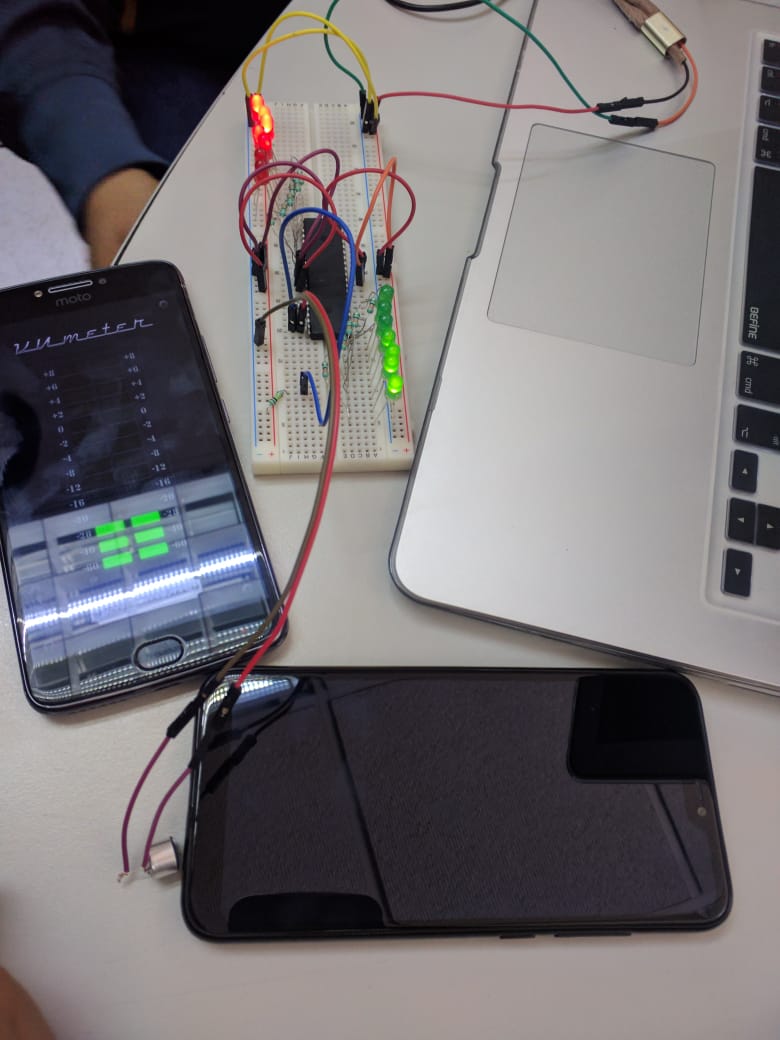
Aldavera Gallaga Iván

Lara Soto Rubén Jair

Morales Castellanos Adolfo Erik

Practica N°14





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>
25. #include <delay.h>
26. #include <io.h>
28. **int** var1,var2;
29. **const** **char** tabla[9]={0x00,0x01,0x03,0x07,0x0f,0x1f,0x3f,0x7f,0xff};
30. // Declare your global variables here
32. #define ADC\_VREF\_TYPE ((0<<REFS1) | (1<<REFS0) | (1<<ADLAR))
34. // Read the 8 most significant bits
35. // of the AD conversion result
36. unsigned **char** read\_adc(unsigned **char** adc\_input)
37. {
38. ADMUX=adc\_input | ADC\_VREF\_TYPE;
39. // Delay needed for the stabilization of the ADC input voltage
40. delay\_us(10);
41. // Start the AD conversion
42. ADCSRA|=(1<<ADSC);
43. // Wait for the AD conversion to complete
44. **while** ((ADCSRA & (1<<ADIF))==0);
45. ADCSRA|=(1<<ADIF);
46. **return** ADCH;
47. }
49. **void** main(**void**)
50. {
51. // Declare your local variables here
53. // Input/Output Ports initialization
54. // Port A initialization
55. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
56. DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
57. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
58. PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);
60. // Port B initialization
61. // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
62. DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
63. // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
64. PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);
66. // Port C initialization
67. // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
68. DDRC=(1<<DDC7) | (1<<DDC6) | (1<<DDC5) | (1<<DDC4) | (1<<DDC3) | (1<<DDC2) | (1<<DDC1) | (1<<DDC0);
69. // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
70. PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);
72. // Port D initialization
73. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
74. DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);
75. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
76. PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);
78. // Timer/Counter 0 initialization
79. // Clock source: System Clock
80. // Clock value: Timer 0 Stopped
81. // Mode: Normal top=0xFF
82. // OC0 output: Disconnected
83. TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
84. TCNT0=0x00;
85. OCR0=0x00;
87. // Timer/Counter 1 initialization
88. // Clock source: System Clock
89. // Clock value: Timer1 Stopped
90. // Mode: Normal top=0xFFFF
91. // OC1A output: Disconnected
92. // OC1B output: Disconnected
93. // Noise Canceler: Off
94. // Input Capture on Falling Edge
95. // Timer1 Overflow Interrupt: Off
96. // Input Capture Interrupt: Off
97. // Compare A Match Interrupt: Off
98. // Compare B Match Interrupt: Off
99. TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
100. TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
101. TCNT1H=0x00;
102. TCNT1L=0x00;
103. ICR1H=0x00;
104. ICR1L=0x00;
105. OCR1AH=0x00;
106. OCR1AL=0x00;
107. OCR1BH=0x00;
108. OCR1BL=0x00;
110. // Timer/Counter 2 initialization
111. // Clock source: System Clock
112. // Clock value: Timer2 Stopped
113. // Mode: Normal top=0xFF
114. // OC2 output: Disconnected
115. ASSR=0<<AS2;
116. TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<CS21) | (0<<CS20);
117. TCNT2=0x00;
118. OCR2=0x00;
120. // Timer(s)/Counter(s) Interrupt(s) initialization
121. TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);
123. // External Interrupt(s) initialization
124. // INT0: Off
125. // INT1: Off
126. // INT2: Off
127. MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
128. MCUCSR=(0<<ISC2);
130. // USART initialization
131. // USART disabled
132. UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
134. // Analog Comparator initialization
135. // Analog Comparator: Off
136. ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
138. // ADC initialization
139. // ADC Clock frequency: 500,000 kHz
140. // ADC Voltage Reference: AVCC pin
141. // ADC High Speed Mode: Off
142. // ADC Auto Trigger Source: ADC Stopped
143. // Only the 8 most significant bits of
144. // the AD conversion result are used
145. ADMUX=ADC\_VREF\_TYPE;
146. ADCSRA=(1<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) | (1<<ADPS0);
147. SFIOR=(1<<ADHSM) | (0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);
149. // SPI initialization
150. // SPI disabled
151. SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);
153. // TWI initialization
154. // TWI disabled
155. TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
157. **while** (1)
158. {
159. var1=read\_adc(0);
160. var2=read\_adc(1);
162. PORTB=tabla[var1/32];
163. delay\_ms(5);
164. PORTC=tabla[var2/32];
165. delay\_ms(5);
166. }
167. }