 Instituto Politécnico Nacional

Escuela Superior de Computo

Materia:

Introducción a los microcontroladores.

Profesor:

Sanchez Aguilar Fernando

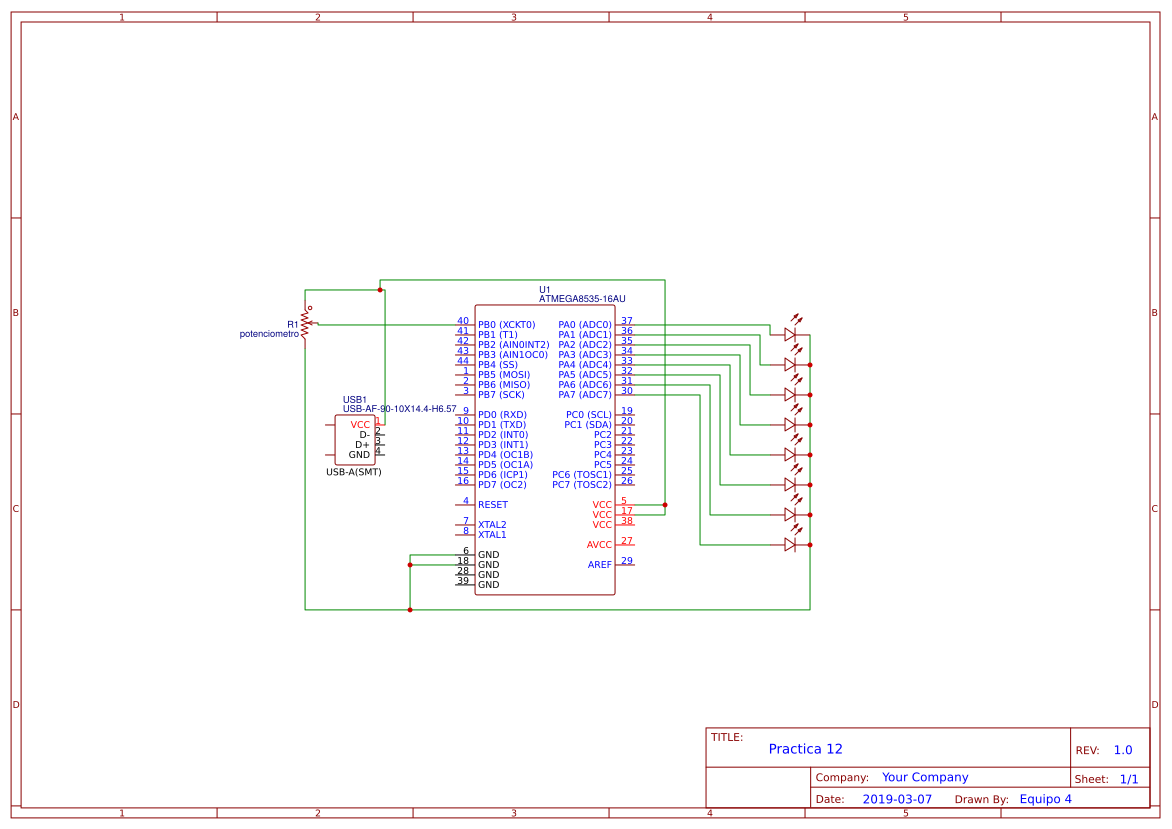
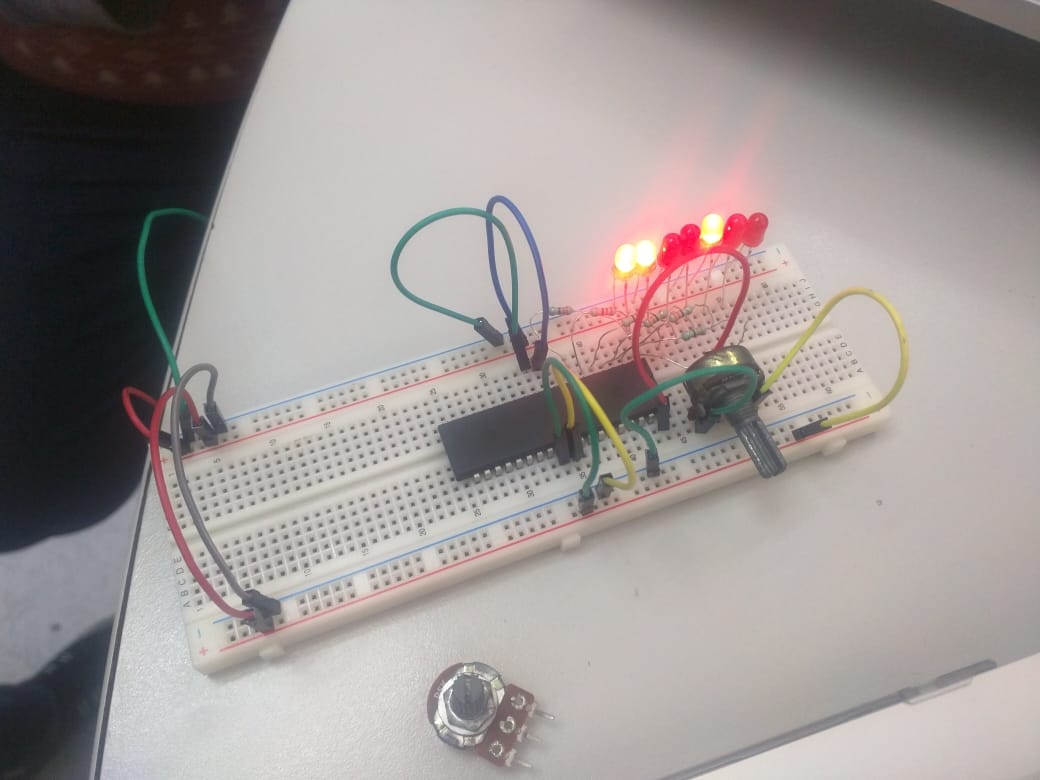
Alumnos:

Aldavera Gallaga Iván

Lara Soto Rubén Jair

Morales Castellanos Adolfo Erik

Practica N°13



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    : 26/02/2019
11. Author  : Equipo 4
12. Company : ESCOM
13. Comments:

16. Chip type               : ATmega8535L
17. Program type            : Application
18. AVR Core Clock frequency: 4,000000 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
34. unsigned **char** read\_adc(unsigned **char** adc\_input)
35. {
36. ADMUX=adc\_input | ADC\_VREF\_TYPE;
37. // Delay needed for the stabilization of the ADC input voltage
38. delay\_us(10);
39. // Start the AD conversion
40. ADCSRA|=(1<<ADSC);
41. // Wait for the AD conversion to complete
42. **while** ((ADCSRA & (1<<ADIF))==0);
43. ADCSRA|=(1<<ADIF);
44. **return** ADCH;
45. }
47. **void** main(**void**)
48. {
49. // Declare your local variables here
51. // Input/Output Ports initialization
52. // Port A initialization
53. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
54. DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
55. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
56. PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);
58. // Port B initialization
59. // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
60. DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
61. // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
62. PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);
64. // Port C initialization
65. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
66. DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);
67. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
68. PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);
70. // Port D initialization
71. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
72. DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);
73. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
74. PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);
76. // Timer/Counter 0 initialization
77. // Clock source: System Clock
78. // Clock value: Timer 0 Stopped
79. // Mode: Normal top=0xFF
80. // OC0 output: Disconnected
81. TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
82. TCNT0=0x00;
83. OCR0=0x00;
85. // Timer/Counter 1 initialization
86. // Clock source: System Clock
87. // Clock value: Timer1 Stopped
88. // Mode: Normal top=0xFFFF
89. // OC1A output: Disconnected
90. // OC1B output: Disconnected
91. // Noise Canceler: Off
92. // Input Capture on Falling Edge
93. // Timer1 Overflow Interrupt: Off
94. // Input Capture Interrupt: Off
95. // Compare A Match Interrupt: Off
96. // Compare B Match Interrupt: Off
97. TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
98. TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
99. TCNT1H=0x00;
100. TCNT1L=0x00;
101. ICR1H=0x00;
102. ICR1L=0x00;
103. OCR1AH=0x00;
104. OCR1AL=0x00;
105. OCR1BH=0x00;
106. OCR1BL=0x00;
108. // Timer/Counter 2 initialization
109. // Clock source: System Clock
110. // Clock value: Timer2 Stopped
111. // Mode: Normal top=0xFF
112. // OC2 output: Disconnected
113. ASSR=0<<AS2;
114. TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<CS21) | (0<<CS20);
115. TCNT2=0x00;
116. OCR2=0x00;
118. // Timer(s)/Counter(s) Interrupt(s) initialization
119. TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);
121. // External Interrupt(s) initialization
122. // INT0: Off
123. // INT1: Off
124. // INT2: Off
125. MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
126. MCUCSR=(0<<ISC2);
128. // USART initialization
129. // USART disabled
130. UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
132. // Analog Comparator initialization
133. // Analog Comparator: Off
134. ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
136. // ADC initialization
137. // ADC Clock frequency: 500,000 kHz
138. // ADC Voltage Reference: AVCC pin
139. // ADC High Speed Mode: Off
140. // ADC Auto Trigger Source: ADC Stopped
141. // Only the 8 most significant bits of
142. // the AD conversion result are used
143. ADMUX=ADC\_VREF\_TYPE;
144. ADCSRA=(1<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (1<<ADPS1) | (1<<ADPS0);
145. SFIOR=(1<<ADHSM) | (0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);
147. // SPI initialization
148. // SPI disabled
149. SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);
151. // TWI initialization
152. // TWI disabled
153. TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
155. **while** (1)
156. {
157. PORTB=read\_adc(0);
158. }
159. }