



Instituto Politécnico Nacional



Escuela Superior de Computo

Materia:

Introducción a los microcontroladores.

Profesor:

Sanchez Aguilar Fernando

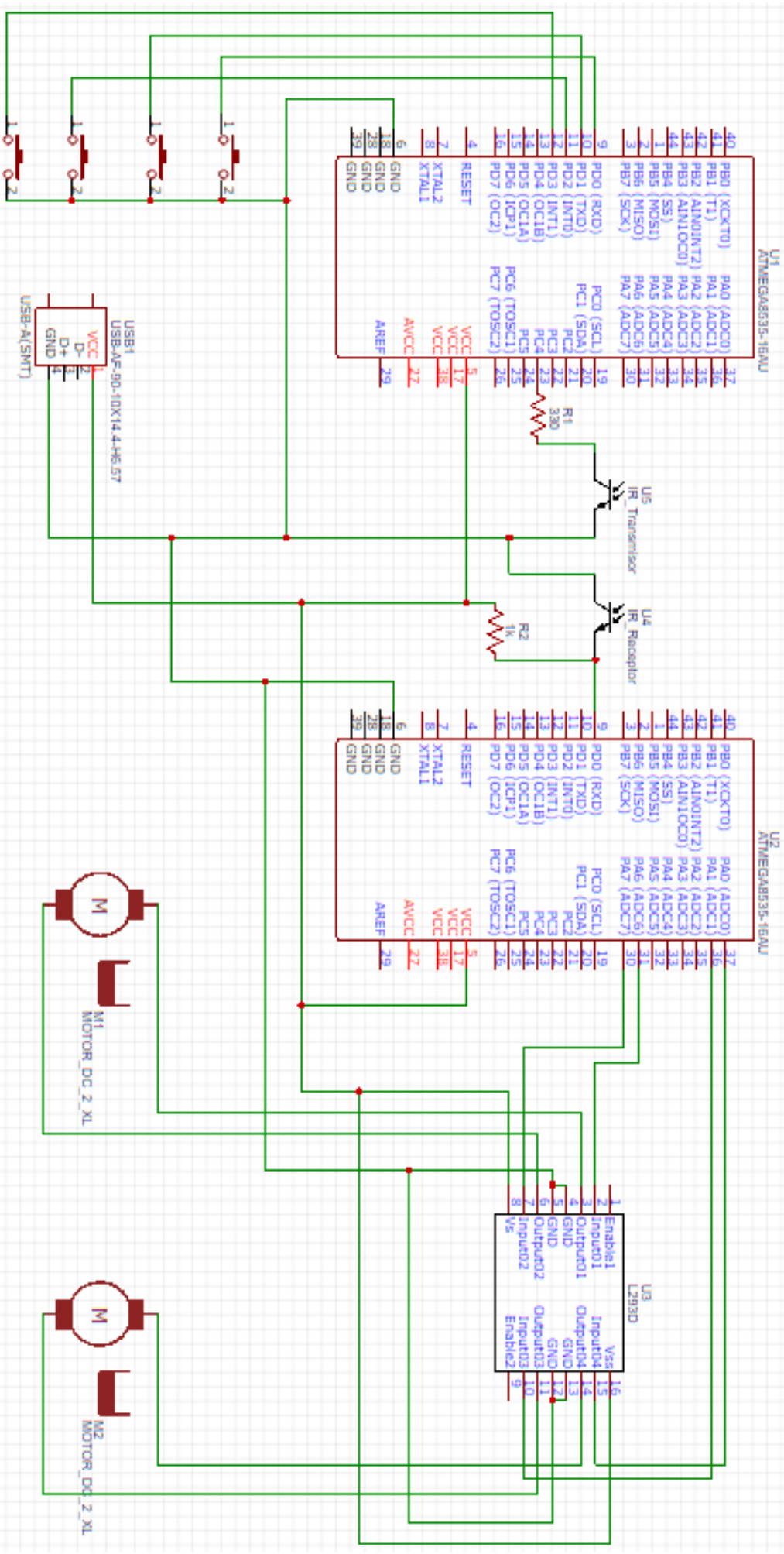
Alumnos:

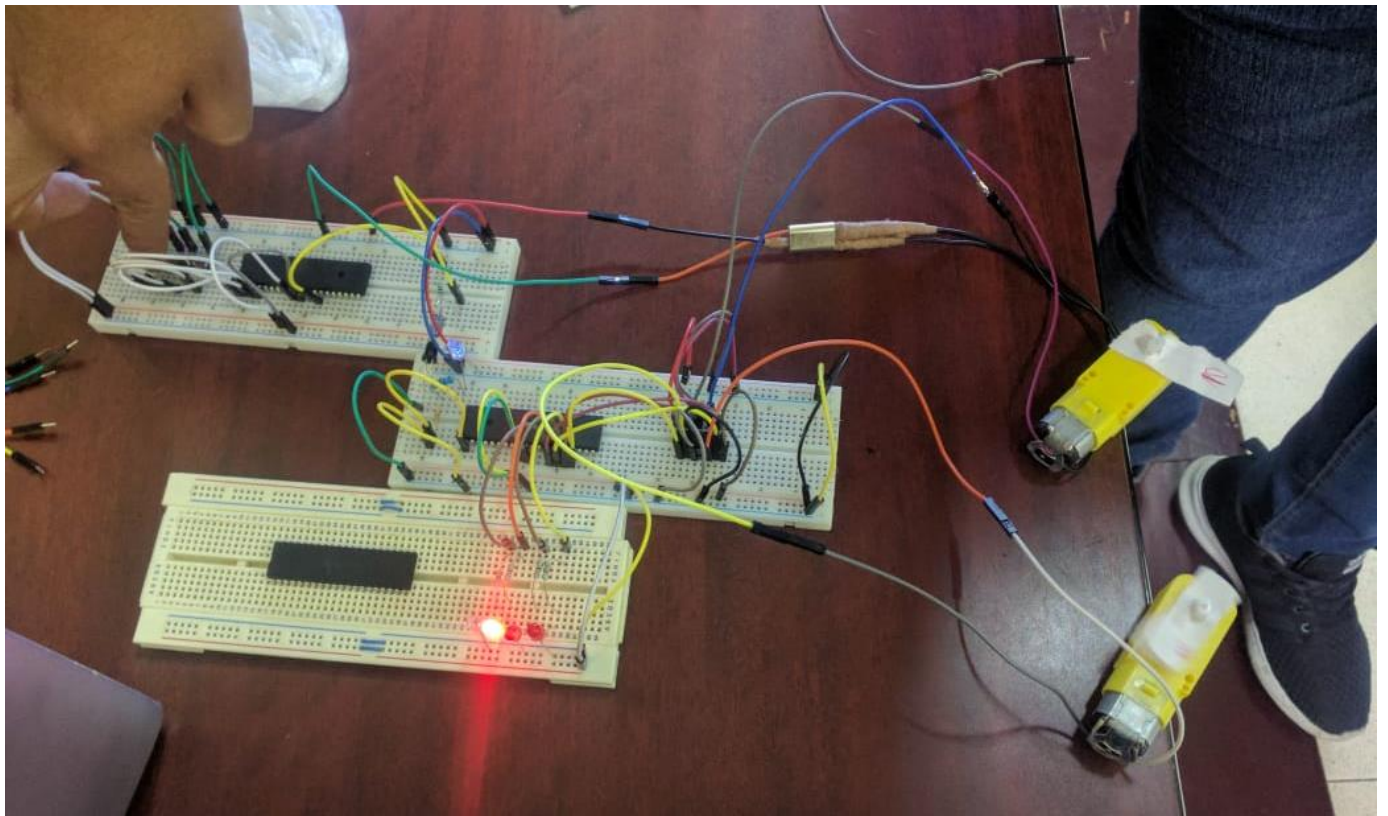
Aldavera Gallaga Iván

Lara Soto Rubén Jair

Morales Castellanos Adolfo Erik

Practica N°9(Proyecto)





Transmisior

```
1. Chip type           : ATmega8535
2. Program type        : Application
3. AVR Core Clock frequency: 1.000000 MHz
4. Memory model         : Small
5. External RAM size    : 0
6. Data Stack size     : 128
7. *****/
8.
9. #include <mega8535.h>
10. #include <delay.h>
11.
12. char opcion, var;
13.
14. void main(void)
15. {
16. // Declare your local variables here
17.
18. // Input/Output Ports initialization
19. // Port A initialization
20. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
21. DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
22. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
23. PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);
24.
25. // Port B initialization
26. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
27. DDRB=(0<<ddb7) | (0<<ddb6) | (0<<ddb5) | (0<<ddb4) | (0<<ddb3) | (0<<ddb2) | (0<<ddb1) | (0<<ddb0);
28. // State: Bit7=P Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
29. PORTB=(1<<PORTB7) | (1<<PORTB6) | (1<<PORTB5) | (1<<PORTB4) | (1<<PORTB3) | (1<<PORTB2) | (1<<PORTB1) | (1<<PORTB0);
30.
31. // Port C initialization
32. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
33. DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);
34. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
35. PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);
36.
37. // Port D initialization
38. // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
39. DDRD=(1<<DDD7) | (1<<DDD6) | (1<<DDD5) | (1<<DDD4) | (1<<DDD3) | (1<<DDD2) | (1<<DDD1) | (1<<DDD0);
40. // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
41. PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);
42.
43. // Timer/Counter 0 initialization
44. // Clock source: System Clock
45. // Clock value: Timer 0 Stopped
46. // Mode: Normal top=0xFF
47. // OC0 output: Disconnected
```

```

48. TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) |
   (0<<CS00);
49. TCNT0=0x00;
50. OCR0=0x00;
51.
52. // Timer/Counter 1 initialization
53. // Clock source: System Clock
54. // Clock value: Timer1 Stopped
55. // Mode: Normal top=0xFFFF
56. // OC1A output: Disconnected
57. // OC1B output: Disconnected
58. // Noise Canceler: Off
59. // Input Capture on Falling Edge
60. // Timer1 Overflow Interrupt: Off
61. // Input Capture Interrupt: Off
62. // Compare A Match Interrupt: Off
63. // Compare B Match Interrupt: Off
64. TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
65. TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) |
   (0<<CS10);
66. TCNT1H=0x00;
67. TCNT1L=0x00;
68. ICR1H=0x00;
69. ICR1L=0x00;
70. OCR1AH=0x00;
71. OCR1AL=0x00;
72. OCR1BH=0x00;
73. OCR1BL=0x00;
74.
75. // Timer/Counter 2 initialization
76. // Clock source: System Clock
77. // Clock value: Timer2 Stopped
78. // Mode: Normal top=0xFF
79. // OC2 output: Disconnected
80. ASSR=0<<AS2;
81. TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<CS21) |
   (0<<CS20);
82. TCNT2=0x00;
83. OCR2=0x00;
84.
85. // Timer(s)/Counter(s) Interrupt(s) initialization
86. TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) |
   (0<<OCIE0) | (0<<TOIE0);
87.
88. // External Interrupt(s) initialization
89. // INT0: Off
90. // INT1: Off
91. // INT2: Off
92. MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
93. MCUCSR=(0<<ISC2);
94.
95. // USART initialization
96. // USART disabled
97. UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) |
   (0<<RXB8) | (0<<TXB8);
98.
99. // Analog Comparator initialization
100. // Analog Comparator: Off
101. // The Analog Comparator's positive input is
102. // connected to the AIN0 pin

```

```

103.     // The Analog Comparator's negative input is
104.     // connected to the AIN1 pin
105.     ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) |
(0<<ACIS1) | (0<<ACIS0);
106.     SFIOR=(0<<ACME);
107.
108.     // ADC initialization
109.     // ADC disabled
110.     ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<AD
PS2) | (0<<ADPS1) | (0<<ADPS0);
111.
112.     // SPI initialization
113.     // SPI disabled
114.     SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA)
| (0<<SPR1) | (0<<SPR0);
115.
116.     // TWI initialization
117.     // TWI disabled
118.     TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
119.
120.
121.     var = 0;
122.     opcion = 0;
123.     while (1)
124.     {
125.         opcion = PINB & 0x0f;
126.
127.         if(opcion == 0x0e)//Arriba
128.         {
129.             for(var = 0; var < 20; var++)
130.             {
131.                 PORTD = 0;
132.                 delay_ms(20);
133.                 PORTD = 1;
134.                 delay_ms(1);
135.             }
136.         }
137.         else if(opcion == 0x0d)//Abajo
138.         {
139.             for(var = 0; var < 30 ; var++)
140.             {
141.                 PORTD = 0;
142.                 delay_ms(15);
143.                 PORTD = 1;
144.                 delay_ms(1);
145.             }
146.         }
147.         else if(opcion == 0x0b)//Izquierda
148.         {
149.             for(var = 0; var < 40; var++)
150.             {
151.                 PORTD = 0;
152.                 delay_ms(10);
153.                 PORTD = 1;
154.                 delay_ms(1);
155.             }
156.         }
157.         else if(opcion == 0x07)//Derecha
158.         {
159.             for(var = 0; var < 50; var++)
160.             {

```

```

161.          PORTD = 0;
162.          delay_ms(5);
163.          PORTD = 1;
164.          delay_ms(1);
165.      }
166.  }
167.  else
168.  {
169.      PORTD = 1;
170.      delay_ms(40);
171.  }
172.
173.  }
174.  }

```

Receptor

```

1. #include <mega8535.h>
2. #include <delay.h>
3.
4. char anterior;
5.
6. unsigned char getPIN()
7. {
8.     if(PINB & 1 && !anterior)
9.     {
10.
11.         delay_ms(6);
12.         if(PINB & 0x1)
13.             delay_ms(4);
14.         else
15.             return 0x2;    //izquierda
16.
17.         if(PINB & 0x1)
18.             delay_ms(5);
19.         else
20.             return 0x1;    //derecha
21.
22.         if(PINB & 0x1)
23.             delay_ms(5);
24.         else
25.             return 0x3;    //arriba
26.
27.         if(PINB & 0x1)
28.             return 0;
29.         else
30.             return 0xc;    //abajo
31.     }
32.     return 0;
33. }
34.
35. void main(void)
36. {
37.     // Declare your local variables here
38.
39.     // Input/Output Ports initialization
40.     // Port A initialization
41.     // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

```

```

42. DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
43. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
44. PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);
45.
46. // Port B initialization
47. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
48. DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) | (0<<DDB1) | (0<<DDB0);
49. // State: Bit7=P Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
50. PORTB=(1<<PORTB7) | (1<<PORTB6) | (1<<PORTB5) | (1<<PORTB4) | (1<<PORTB3) | (1<<PORTB2) | (1<<PORTB1) | (1<<PORTB0);
51.
52. // Port C initialization
53. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
54. DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);
55. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
56. PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);
57.
58. // Port D initialization
59. // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
60. DDRD=(1<<DDD7) | (1<<DDD6) | (1<<DDD5) | (1<<DDD4) | (1<<DDD3) | (1<<DDD2) | (1<<DDD1) | (1<<DDD0);
61. // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
62. PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);
63.
64. // Timer/Counter 0 initialization
65. // Clock source: System Clock
66. // Clock value: Timer 0 Stopped
67. // Mode: Normal top=0xFF
68. // OC0 output: Disconnected
69. TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
70. TCNT0=0x00;
71. OCR0=0x00;
72.
73. // Timer/Counter 1 initialization
74. // Clock source: System Clock
75. // Clock value: Timer1 Stopped
76. // Mode: Normal top=0xFFFF
77. // OC1A output: Disconnected
78. // OC1B output: Disconnected
79. // Noise Canceler: Off
80. // Input Capture on Falling Edge
81. // Timer1 Overflow Interrupt: Off
82. // Input Capture Interrupt: Off
83. // Compare A Match Interrupt: Off
84. // Compare B Match Interrupt: Off
85. TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
86. TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
87. TCNT1H=0x00;
88. TCNT1L=0x00;
89. ICR1H=0x00;
90. ICR1L=0x00;

```



```

91. OCR1AH=0x00;
92. OCR1AL=0x00;
93. OCR1BH=0x00;
94. OCR1BL=0x00;
95.
96. // Timer/Counter 2 initialization
97. // Clock source: System Clock
98. // Clock value: Timer2 Stopped
99. // Mode: Normal top=0xFF
100. // OC2 output: Disconnected
101. ASSR=0<<AS2;
102. TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<
CS21) | (0<<CS20);
103. TCNT2=0x00;
104. OCR2=0x00;
105.
106. // Timer(s)/Counter(s) Interrupt(s) initialization
107. TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) |
(0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);
108.
109. // External Interrupt(s) initialization
110. // INT0: Off
111. // INT1: Off
112. // INT2: Off
113. MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
114. MCUCSR=(0<<ISC2);
115.
116. // USART initialization
117. // USART disabled
118. UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<U
CS22) | (0<<RXB8) | (0<<TXB8);
119.
120. // Analog Comparator initialization
121. // Analog Comparator: Off
122. // The Analog Comparator's positive input is
123. // connected to the AIN0 pin
124. // The Analog Comparator's negative input is
125. // connected to the AIN1 pin
126. ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) |
(0<<ACIS1) | (0<<ACIS0);
127. SFIOR=(0<<ACME);
128.
129. // ADC initialization
130. // ADC disabled
131. ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<AD
PS2) | (0<<ADPS1) | (0<<ADPS0);
132.
133. // SPI initialization
134. // SPI disabled
135. SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA)
| (0<<SPR1) | (0<<SPR0);
136.
137. // TWI initialization
138. // TWI disabled
139. TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
140.
141. while (1)
142. {
143.     anterior = PINB & 1;
144.     PORTD = getPIN();
145.     if(PORTD != 0) delay_ms(500);}}

```