

Projeto Intermediário - 3 - Alarme e intensidade por controle de botões

Tutorial:

Componentes

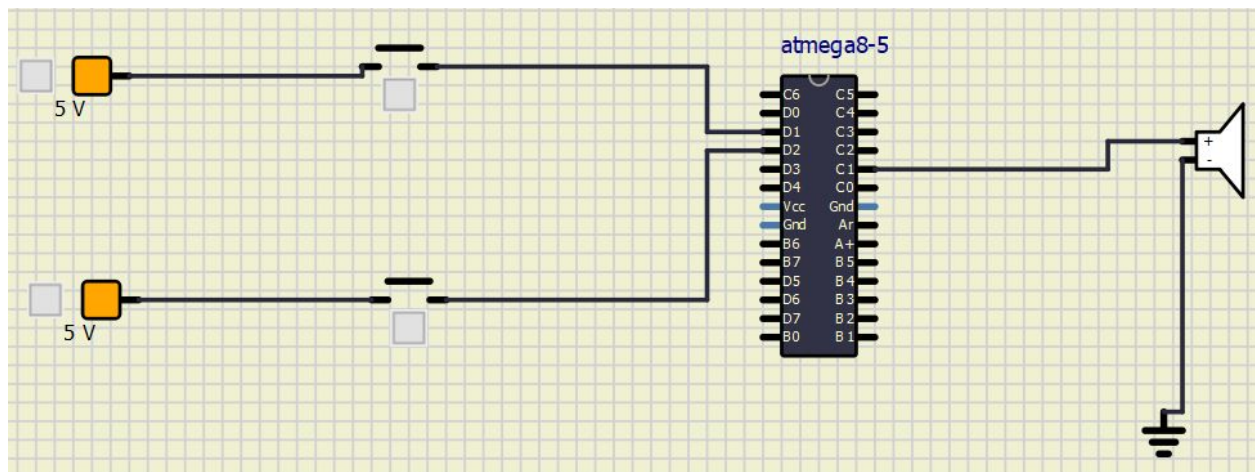
2 Tensão Fixa 5v

2 Botão

1 Saída de Áudio

1 Microcontrolador Atmega8-18

Simulação Software SimulIDE



Montagem do Projeto no SimulIDE

Conexões dos componentes:

Botão 1	Porta D1
Botão 2	Porta D2
Saída de Áudio	Porta C1

Programação em C Software CODEVision

Bibliotecas utilizadas:

<stdio.h>
<delay.h>
<mega8.h>

/*****

This program was produced by the
CodeWizardAVR V2.05.0 Advanced
Automatic Program Generator
© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project :
Version :
Date : 16/05/2013
Author : www.Eca.ir * www.Webkade.ir
Company :
Comments:

Chip type : ATmega8
Program type : Application
AVR Core Clock frequency: 14,745600 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 256
*****/

#include <mega8.h>

// Standard Input/Output functions
#include <stdio.h>

#include <delay.h> // Inclui a biblioteca de funções de temporização

// Declare your global variables here

int cont;

```
void main(void)
{
// Declare your local variables here
cont = 0;

// Input/Output Ports initialization
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x02;

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x02;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
TCCR0=0x00;
TCNT0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
```

```
TCCR1A=0x00;  
TCCR1B=0x00;  
TCNT1H=0x00;  
TCNT1L=0x00;  
ICR1H=0x00;  
ICR1L=0x00;  
OCR1AH=0x00;  
OCR1AL=0x00;  
OCR1BH=0x00;  
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization  
// Clock source: System Clock  
// Clock value: Timer2 Stopped  
// Mode: Normal top=0xFF  
// OC2 output: Disconnected  
ASSR=0x00;  
TCCR2=0x00;  
TCNT2=0x00;  
OCR2=0x00;
```

```
// External Interrupt(s) initialization  
// INT0: Off  
// INT1: Off  
MCUCR=0x00;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization  
TIMSK=0x00;
```

```
// USART initialization  
// Communication Parameters: 8 Data, 1 Stop, No Parity  
// USART Receiver: Off  
// USART Transmitter: On  
// USART Mode: Asynchronous  
// USART Baud Rate: 19200  
UCSRA=0x00;  
UCSRB=0x08;  
UCSRC=0x86;  
UBRRH=0x00;  
UBRRL=0x2F;
```

```
// Analog Comparator initialization  
// Analog Comparator: Off
```

```
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIO=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

while (1)
{
    // Place your code here
    if (PIND.1)
    {
        cont = cont + 1;
        printf("Numero de pessoas no interior: = %d.\r\n",cont);

    }

    delay_ms(500); // Aguarda 500 milisegundos

    if (PIND.2)
    {
        if(cont == 0)
        {
            printf("Ambiente vazio \r\n");
        }

        else
        {
            cont = cont - 1;
            printf("Numero de pessoas no interior: = %d.\r\n",cont);
        }

    }

};
```

```
while(cont==5)
{
    PORTC.1=~PORTC.1;
    delay_ms(500);

    break;
}

while(cont==6)
{
    PORTC.1=~PORTC.1;
    delay_ms(150);

    break;
}

while(cont >= 7)
{
    PORTC.1=1;
    delay_ms(10);

    break;
}

} //while
}
```