**Instituto Politécnico Nacional**

**Escuela Superior de Cómputo**

Introducción a los Microcontroladores

***Práctica 6:*** *Contador con Rebotes*

**Integrantes del equipo:**

Martínez Ortega Juan Yael

Sampayo Hernández Mauro

**Grupo:** 3CM6

**Profesor:** *Fernando Aguilar Sánchez* **Fecha de entrega:** 8 de noviembre de 2020

Práctica 1: Uso de Puertos de Entrada y Salida

3CM6

ESCOM-IPN

*1. Introducción Teórica*

*2. Desarrollo experimental de la práctica*

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This program was produced by the

CodeWizardAVR V2.05.0 Professional

Automatic Program Generator

Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.

http://www.hpinfotech.com

Project :

Version :

Date : 02/11/2020

Author : NeVaDa

Company :

Comments:

Chip type : ATmega8535

Program type : Application

AVR Core Clock frequency: 1.000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 128

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <mega8535.h>

#include <delay.h>

#define boton PIND.0 //Etiqueta el Pin 0 del Puerto D -> boton

bit botonp**;**//Artificio para hacer la detecci󮠤e los flancos

bit botona**;**//Artificio para hacer la detecci󮠤e los flancos

unsigned char var**;**

const char tabla7segmentos**[**10**]={**0x3f**,**0x06**,**0x5b**,**0x4f**,**0x66**,**0x6d**,**0x7c**,**0x07**,**0x7f**,**0x6f**};**

void main**(**void**)**

**{**

// Declare your local variables here

// Input/Output Ports initialization

// Port A 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

PORTA**=**0x00**;**

DDRA**=**0x00**;**

// Port B initialization

// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out

// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0

PORTB**=**0x00**;**

DDRB**=**0xFF**;**

// Port C 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

PORTC**=**0x00**;**

DDRC**=**0x00**;**

// 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=P

PORTD**=**0x01**;**

DDRD**=**0x00**;**

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0**=**0x00**;**

TCNT0**=**0x00**;**

OCR0**=**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

// INT2: Off

MCUCR**=**0x00**;**

MCUCSR**=**0x00**;**

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK**=**0x00**;**

// USART initialization

// USART disabled

UCSRB**=**0x00**;**

// Analog Comparator initialization

// Analog Comparator: Off

// Analog Comparator Input Capture by Timer/Counter 1: Off

ACSR**=**0x80**;**

SFIOR**=**0x00**;**

// ADC initialization

// ADC disabled

ADCSRA**=**0x00**;**

// SPI initialization

// SPI disabled

SPCR**=**0x00**;**

// TWI initialization

// TWI disabled

TWCR**=**0x00**;**

**while** **(**1**)**

**{**

**if(**boton**==**0**)**

botona**=**0**;**

**else**

botona**=**1**;**

**if((**botonp**==**1**)&&(**botona**==**0**))**//hubo cambio de flanco de 1 a 0

**{**

var**++;**//se incrementa la var1iable

**if(**var**==**10**)**

var**=**0**;**

delay\_ms**(**40**);**//se coloca retardo de 40ms para eliminar rebotes

**}**

**if((**botonp**==**0**)&&(**botona**==**1**))**//hubo cambio de flanco de 0 a 1

delay\_ms**(**40**);**

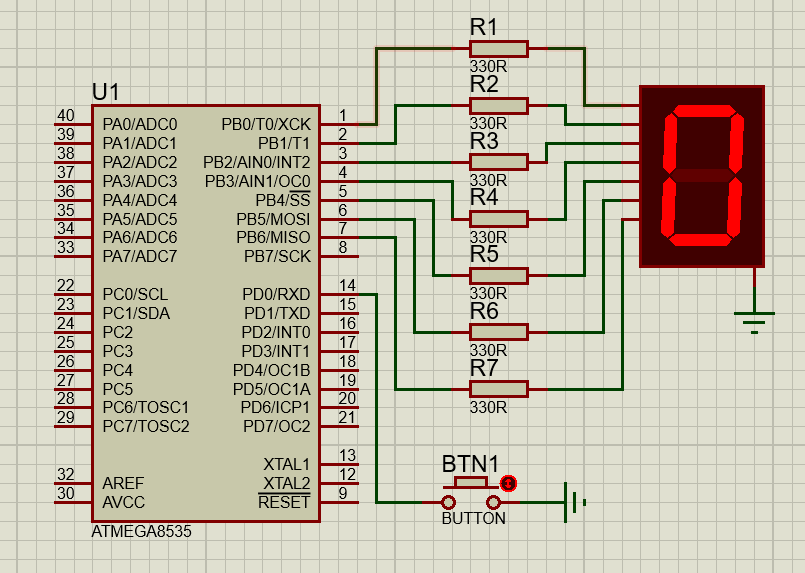
PORTB**=**tabla7segmentos**[**var**];**

botonp**=**botona**;**

**}**

**}**

**2.2 Simulación**



*3. Conclusiones Individuales*

**3.1 Martínez Ortega Juan Yael**

**3.2 Sampayo Hernández Mauro**

*4. Anexos*

1. **Hoja de especificaciones del Microcontrolador ATMega8535**

