

Memoria EEPROM



Objetivo

Al término de la sesión, los integrantes del equipo contarán con la habilidad de hacer uso de la memoria EEPROM del microcontrolador.

Introducción Teórica

Realizada por los alumnos.

Materiales y Equipo empleado

- ✓ CodeVision AVR
- ✓ AVR Studio 4
- ✓ Microcontrolador ATmega 8535
- ✓ 1 Display cátodo común
- ✓ 7 Resistores de $330\ \Omega$ a $\frac{1}{4}\text{ W}$
- ✓ 2 Push Button

Desarrollo Experimental

1.- Haga un programa en el cual con un botón conectado al pin C0 incrementará el valor en el display conectado en el puerto B y cuando se presione el botón C1 lo guarde en la memoria EEPROM. Después desconecte el microcontrolador de la energía eléctrica y vuélvalo a conectar para que observe que el dato se quedó guardado.

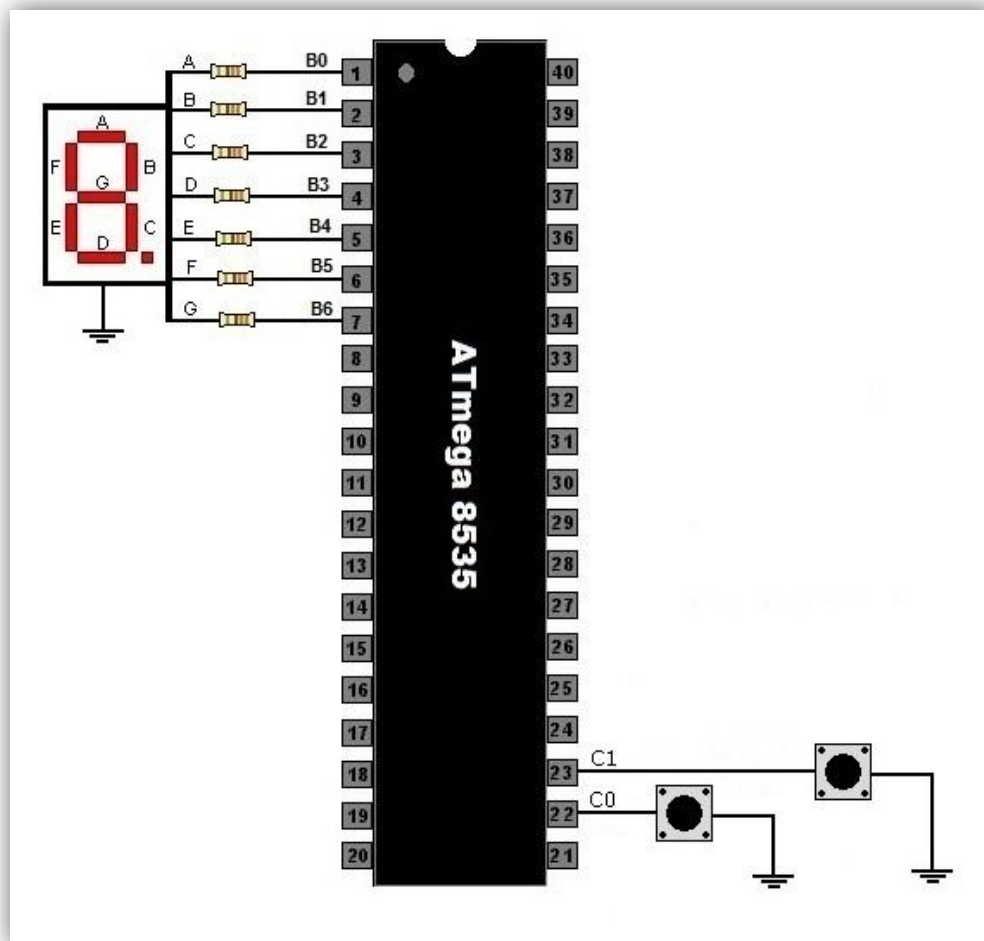


Figura 1. Circuito para el uso de la memoria EEPROM.

Estructura del programa

```
#include <mega8535.h>
#include <delay.h>
#define boton PINC.0
#define boton_guarda PINC.1
bit botonp;
bit botona;
unsigned char var; //Ahora la variable se guarda en eeprom
const char tabla7segmentos [10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7c,0x07,0x7f,0x6f};
eeprom char datoaguardar;
void checa_boton (void); // Aquí se declaran todas las funciones que se van usar
// Declare your global variables here

.
.
.
if (datoaguardar>10)
datoaguardar=0;
var=datoaguardar;
while (1)
{
checa_boton();
PORTB=tabla7segmentos [var];
if (boton_guarda==0) //Si se presiona el botón de guardar
datoaguardar=var; //se grabara la eeprom con el valor de var
// Place your code here
};
}

void checa_boton (void)
{
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 variable
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); //Se coloca retardo de 40mS para eliminar rebotes
botonp=botona;
}
```

Observaciones y Conclusiones Individuales

Bibliografía

