```
/*-------------------------------------------------------
  Programa: EEPROM18F.c
  Data    : 17/05/2017
  Autor   : Prof. Vargas

  Descrição:
  Exemplo de operações com dados manipulados em EEPROM
  Bibliotecas baseadas no curso de formação de especialistas
  em microcontroladores da Microgenios
-------------------------------------------------------*/
void EEPROM_Write_byte(unsigned char address, unsigned char dta);
unsigned char EEPROM_Read_byte(unsigned char address);
void EEPROM_Write_int (unsigned char address, unsigned int dta);
unsigned int EEPROM_Read_int (unsigned int address);
//array unidimensional de byte
void EEPROM_Write_array (unsigned char address, unsigned char qtd_byte,unsigned char *dta);
void EEPROM_Read_array  (unsigned char address, unsigned char qtd_byte,unsigned char *buffer
r);
void EEPROM_Write_float(unsigned char address,unsigned char *dta);
float EEPROM_Read_float(unsigned char address);
void Resgata(void);

unsigned char j,k;
unsigned char memoria[10];

void main()
```

```
{
   LATA = 0;
   TRISA = 0b11000000;
   TRISB = 0b11000000;

   for (k=0;k<10;k=k+2)       // Prepara os dados
   {
      memoria[k] = k+1;
   }
   for (k=1;k<10;k=k+2)
   {
     memoria[k] = k-1;
   }
   for(k=0; k<10; k++)      // Preenche dados na EEPROM
   {
      EEPROM_Write_byte(k,9-k);
   }
   k=0;
   while(1)
   {
     if (Button(&PORTB,6,1,0))     //Ativa botão "Lê"
     {
        do
        {
        }while(Button(&PORTB,6,1,0)); //Aguarda soltar
        Resgata();
     }
```

```
     if (Button(&PORTB,7,1,0))     //Ativa botão "Grava"
     {
        do
        {
        }while(Button(&PORTB,7,1,0)); //Aguarda soltar
        for(j=0; j<10; j++)     // Preenche dados na EEPROM
        {
           EEPROM_Write_byte(j,j);
        }
     }
     LATA = memoria[k];
     k++;
     if(k==10)
     {
       k=0;
     }
     LATB.RB5 = ~LATB.RB5;
     Delay_ms(500);

   }

}

void EEPROM_Write_byte(unsigned char address, unsigned char dta)
{
   char  STATUS_GIE  = 0;
   EEADR        = address; //Endereço a ser gravado
```

```
   EEDATA       = dta;     //Byte a ser gravado
   EECON1.EEPGD = 0;    //Aponta para a memória EEPROM
   EECON1.CFGS  = 0;    //Acesso ao controle da EEPROM
   EECON1.WREN  = 1;    //Habilita modo de gravação da EEPROM
   STATUS_GIE = INTCON.GIEH;
   INTCON.GIEH = 0;
   EECON2 = 0x55;       //procedimento obrigatório p/ escrita
   EECON2 = 0xAA;       //procedimento obrigatório p/ escrita
   EECON1.WR = 1;       //Realiza a gravação da EEPROM
   while(EECON1.WR == 1); //Aguarda o término da gravação
   INTCON.GIEH = STATUS_GIE;
   EECON1.WREN = 0;     //Desabilita o modo de gravação
}

unsigned char EEPROM_Read_byte(unsigned char address)
{
  EEADR        = address;   //Endereço da EEPROM a ser Lido
  EECON1.CFGS  = 0;     //Acesso ao controle da memória EEPROM
  EECON1.EEPGD = 0;     //Aponta para a memória EEPROM
  EECON1.RD    = 1;     //Inicia a Leitura da EEPROM
  asm nop;              //Recomenda-se aguardar 2 ciclos
  asm nop;              // de máquina antes de ler EEDATA
  return EEDATA;        //O valor da leitura está em EEDATA
}

void EEPROM_Write_int (unsigned char address, unsigned int dta)
{
```

```
  EEPROM_Write_byte(address + 0, (char)dta);
  EEPROM_Write_byte(address + 1, (char)(dta>>8));
}

unsigned int EEPROM_Read_int (unsigned int address)
{
  unsigned char _val1, _val2;
  _val1 = EEPROM_Read_byte(address + 0);
  _val2 = EEPROM_Read_byte(address + 1);
  return ((_val1 << 8)| (int)_val2); //concatenando os bytes para formar 1 inteiro
}

//array unidimensional de bytes
void EEPROM_Write_array (unsigned char address, unsigned char qtd_byte,unsigned char *dta)
{
  char _i;
  for(_i = 0; _i < qtd_byte; ++_i)
  {
    EEPROM_Write_byte(address+_i, *dta);
    ++dta;
  }
}

void EEPROM_Read_array (unsigned char address, unsigned char qtd_byte,unsigned char *buffer
)
{
  char _i;
```

```
  for(_i = 0; _i < qtd_byte; ++_i)
  {
    *buffer = EEPROM_Read_byte(address+_i);
    ++buffer;
  }
}

void EEPROM_Write_float(unsigned char address,unsigned char *dta)
{
   EEPROM_Write_array(address, 4, &dta);
}

float EEPROM_Read_float(unsigned char address)
{
  union Result
  {
    unsigned char _res[4];
    float _t;
  } result;

   result._res[0] = EEPROM_Read_byte(address+0);
   result._res[1] = EEPROM_Read_byte(address+1);
   result._res[2] = EEPROM_Read_byte(address+2);
   result._res[3] = EEPROM_Read_byte(address+3);
   return result._t;
}
```

```
void Resgata(void)
{
   for (j=0;j<10;j++)
   {
      memoria[j]=EEPROM_Read_byte(j);
   }
}
```