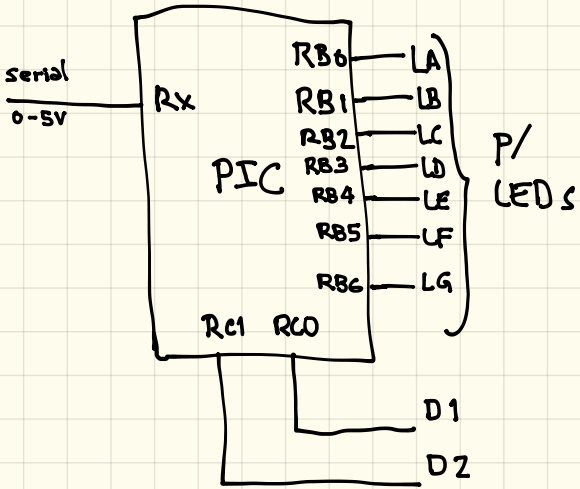
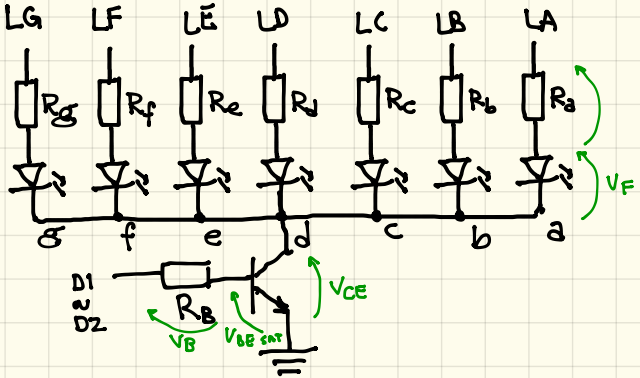
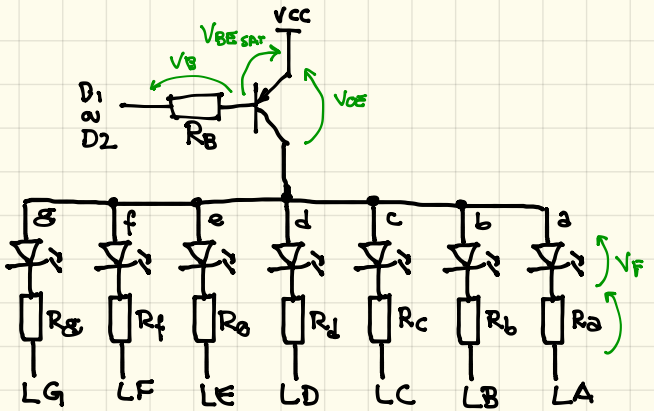
PROJETO: Display de sete segmentos duplo com serial







```c
/*
 * File:    main.c
 * Author:  Jun Okamoto Jr.
 * Project: Display de sete segmentos duplo com serial
 *          Solução para catodo comum
 *
 * Created on June 14, 2018, 10:38 AM
 */

#pragma config FOSC = INTRC_NOCLKOUT // clock interno de 8MHz

#include <xc.h>
#include "delay.h"
#include "serial.h"

#define D1 RC0 // display menos significativo
#define D2 RC1 // display mais significativo

__EEPROM_DATA(0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7c, 0x07); // carrega na EEPROM os bytes de 0 a 7
__EEPROM_DATA(0x7f, 0x6f, 255, 255, 255, 255, 255, 255);       // carrega na EEPROM os bytes de 8 a 15

// Variável Global
volatile char rcBuffer[2] = {'0','0'};  // meu buffer de recepção

void interrupt isr (void) { // rotina de tratamento de interrupções
  static char rcIndex = 0; // indice do buffer de recepção
  if (RCIE && RCIF && PEIE) { // interrupção de recepção serial
    if (rcIndex >= 2) rcIndex = 0; // se indice passar do final do buffer, volta para o início
      rcBuffer[rcIndex++] = RCREG; // coloca dado recebido no buffer e incrementa indice
  } // fim - interrupção de recepção
} // fim - rotina de tratamento de interrupções

void main(void) {
  TRISB = 0x80;   // os 7 primeiros bits são saída
  TRISC0 = 0;     // o bit 0 é saída para o display menos significativo
  TRISC1 = 0;     // o bit 1 é saída para o display mais significativo
  D1 = 0;         // inicialmente o display está desligado
  D2 = 0;         // inicialmente o display está desligado

  serial_init();   // inicializa o canal serial (dado)

  RCIE = 1;       // habilita a interrupção de recepeção serial
  PEIE = 1;       // habilita interrupção de periféricos
  GIE = 1;        // habilita interrupções (geral))

  while(1) {      // loop infinito
    PORTB = eeprom_read(rcBuffer[0]-0x30); // coloca valor do buffer na Porta B
    D1 = 1;       // liga o display menos significativo
    D2 = 0;       // desliga o display mais siginificativo
    delay_ms(10); // mantém aceso por 10 ms
    PORTB = eeprom_read(rcBuffer[1]-0x30); // coloca valor do buffer na Porta B
    D1 = 0;       // desliga o menos significativo
    D2 = 1;       // liga o mais significatibo
    delay_ms(10); // mantém aceso por 10 ms
  } // fim - loop infinito
} // fim - main
```