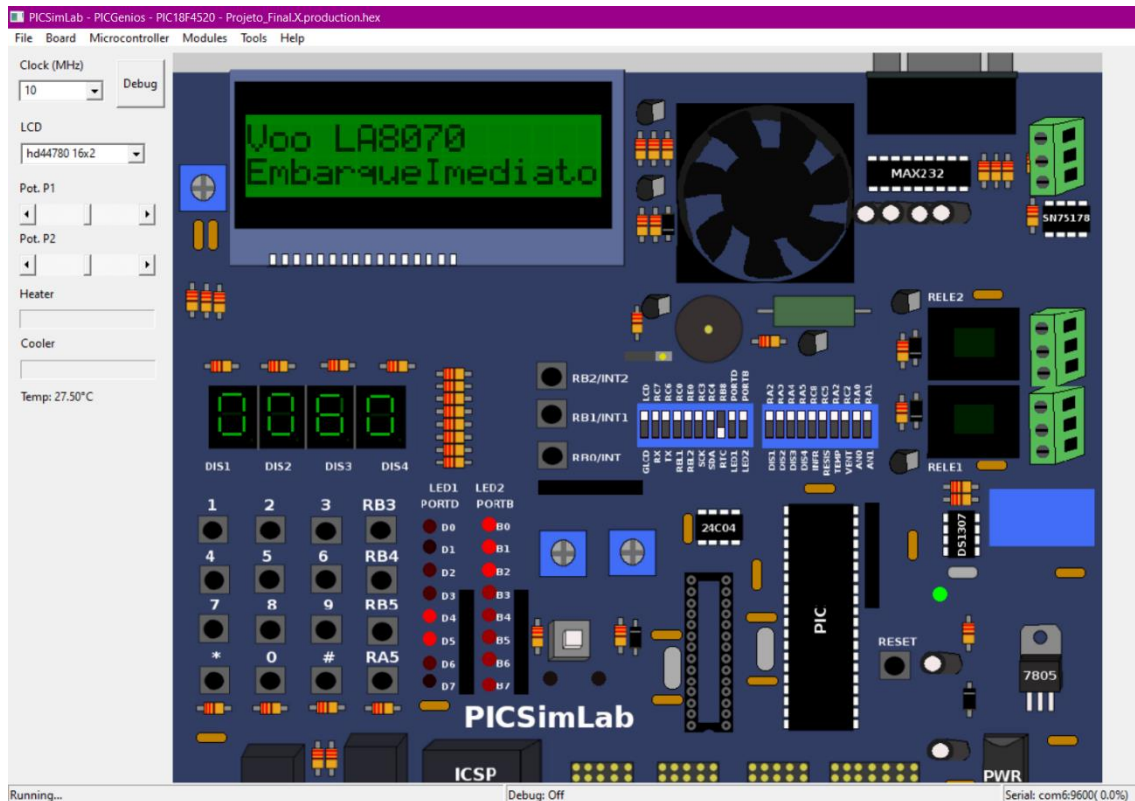


## 1. Descrição Do Funcionamento

A ideia do programa é simular um painel de informações sobre os voos que existem em grande parte dos aeroportos comerciais pelo mundo, utilizando o simulador do microcontrolador PIC18F4520, e programando-o utilizando o software MPLAB-X, ambos desenvolvidos pela *Microchip Technology Inc.* Em um primeiro momento, o projeto tem a seguinte aparência:

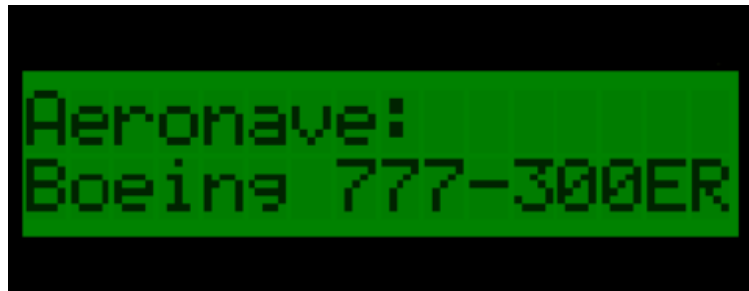


Observa-se, primeiramente, a presença de um contador do display de 7-segmentos, que vai de 60 ao 0, representando a passagem de 1 minuto. Durante esse tempo, o passageiro, ao pressionar alguns botões do teclado matricial, pode analisar diferentes informações de seu voo. São elas:

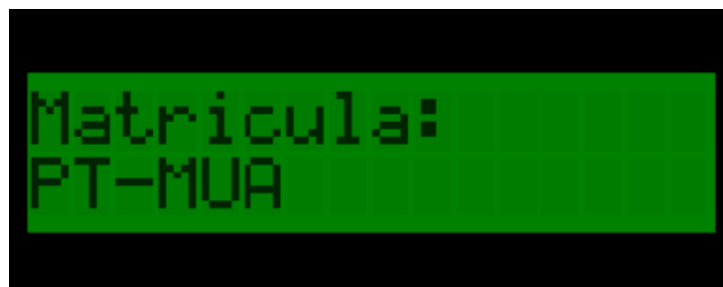
- 1) O destino do voo, no botão 4.



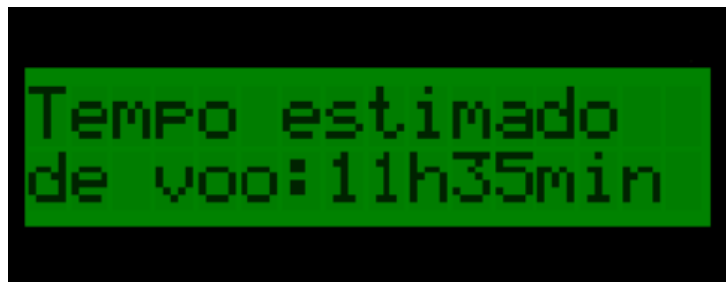
2) A aeronave utilizada, no botão 0.



3) A matrícula da aeronave, no botão 8.



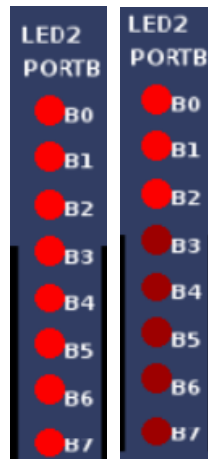
4) O tempo estimado de voo até o destino, no botão 5.



Quando se atinge 0 no *timer*, ele fica pausado por cerca de 5 segundos, enquanto o voo decola. A informação fica no LCD para o usuário, antes de iniciar-se um novo embarque.



Ademais, os leds da porta B foram configurados para piscar a cada outro segundo para criar um efeito de alerta devido ao fato de o embarque ser necessário.



## 2. Criação, dificuldades e soluções

O código foi criado com auxílio principalmente das bibliotecas *lcd*, *ssd*, *keypad*, *bits* e *config*. A estrutura a seguir foi utilizada diversas vezes para poder exibir as diferentes informações no lcd:

```
kpDebounce();  
if (kpRead() != tecla)  
{  
    tecla = kpRead();  
    if (bitTst(tecla, 2))  
    {  
        lcdCommand(CLR);  
    }  
}
```

Com `#define CLR 0x01`, e após isso o comando `lcdData()` para cada caractere da informação que viria a seguir.

O contador foi criado usando a biblioteca *ssd* entrelaçado com uma variável denominada de *valor*, que decresce conforme o tempo inserido em um laço *for*. Para mostrar a dezena e a unidade, usou-se os operadores de resto de divisão e divisão da seguinte forma:

```
value--;  
ssdDigit((value % 10), 3);  
ssdDigit((value / 10) % 10, 2);
```

Sendo que *value* tem valor inicial de 60.

Adiante, a pausa da decolagem ocorre usando uma condicional de tal forma que quando *value* = 0, entra-se em outro *for* que configura a pausa e nele também atualiza-se o lcd constantemente com a informação da decolagem.

Por fim, o efeito de piscagem do led foi atribuído de uma maneira em que o programa identificasse quando *value* estivesse em um valor par e dessa maneira apagasse-os.

A principal dificuldade de balancear no código foi a manipulação de tempo, tendo em vista que a biblioteca lcd apresenta alguns comandos para criar um pequeno *delay*, mas que pode ser agravado caso utilize-se muito texto como ocorre no projeto. Deixar os segundos precisos foi possível apenas sem evitar o efeito do *flicker* no *ssd*, e evitar o *flicker* foi possível deixando a passagem dos segundos imprecisa. A melhor solução até o momento foi evitar o flicker utilizando um clock mais elevado, de 10Mhz, devido ao fato de já estarmos trabalhando com escalas de tempo irreais.