Aula 3 - Gerenciamento do buffer

Revisão da aula 2

- Criada uma estrutura de dados para armazenar as configurações do display
    - armazena pinos e portas dos terminais do display
- feita uma função para enviar os dados em blocos
- modificadas algumas funções de acordo com o método de blocos

Alterações não feitas na aula 2

- Declarar as variáveis locais do LCD.c como static, para garantir que elas não mudaram de valor
- Funções com prefixo LCD são privadas do LCD.c, já as com LCD5110 são visíveis ao usuário

Rotina de escrever uma string

```
* Rotina de desenhar uma string no display é feita com 3 funções;
*
*      # void LCD5110_write_str(char *s)
*          - recebe a string a ser escrita
*          - chama a draw string para construir a string
*          - chama a LCD_write para enviar os dados ao display
*
*      # uint16_t LCD_draw_string(char *s)
*          - recebe a mesma string da anterior
*          - chama a draw char para montar os dados dos caracteres
*          - organiza os dados no Buffer
*
*      # void LCD_draw_char(char letra, uint8_t *buf)
*          - constroi os dados de cada caractere
*
```

Gerenciamento do buffer

A ideia agora é preparar o código para usar transferências não bloqueantes, como DMA e interrupções.

O ponto negativo dessas transmissões está no fato de que o processador não sabe como está o andamento da transmissão, o que pode acarretar em conflito nos dados. Por exemplo, o processador pode tentar escrever no Buffer enquanto o DMA está carregando os dados dele, o que vai sobreescrever os dados.

Pensando nesse problema, vamos criar um gerenciamento ao buffer para que o processador saiba o seu status antes de tentar escrever nele

declarar status

declarar buffercompartilhado

inicializar buffer

trocar o Buffer pelo buffer struct