

## Laboratório 05

### Objetivos:

- Usar a entrada padrão PS/2 para conversar com um teclado PS/2
- Conhecer um novo padrão:
  - No PS/2, a comunicação é uni ou bidirecional?
  - Ao se pressionar o shift, o código enviado é outro (dado que o caractere maiúsculo tem outro código? Ou é o código do shift que também é enviado e o host (computador) precisa tratar desta informação?
- Se familiarizar com este tipo de comunicação, como uma prévia para comunicação serial padrão RS-232C – uma vez que a implementação deste último padrão é pouco mais complexa.

### Descrição:

Você fará com que ao se pressionar uma tecla qualquer de um teclado padrão PS/2, o código hexadecimal da tecla fique nos LEDs discretos (ou nos displays de 7 segmentos, caso prefira dado que seu componente já está pronto).

### Implementação:

Basicamente a comunicação serial do teclado para o host (kit) possui 11 bits, a saber:

- O sistema fica em *idle*, com nível lógico '1'. Assim, o primeiro bit chama-se *start bit* e é obviamente em '0'. Ele sinaliza o início da comunicação;
- Os próximos 8 bits são de dados, sendo que o menos significativo é o primeiro a ser enviado, na sequência do *start bit*;
- O 10º bit é o de paridade. A paridade é sempre ímpar (*odd parity*), o que significa que este bit será:
  - '0' se o número de '1's for ímpar
  - '1' se o número de '1's for par;
- O 11º bit é chamado de *stop bit* e será sempre '1'.

Observe a descrição do padrão no Capítulo 8 do User Guide do nosso kit. Este laboratório tem três facilidades:

- A comunicação entre host (no caso, nosso kit) e teclado, por padrão, é bidirecional. Isso permite, por exemplo, que o host comande os LEDs do teclado *Caps Lock*, *Num Lock* e *Scroll Lock*, ou mesmo informar ao teclado se houve erro na transmissão. A nossa primeira simplificação é que basta o host (nosso kit) receber os dados (comunicação unidirecional);
- A segunda simplificação se dá pelo código de controle. Você verá que a cada tecla pressionada, dois dados serão enviados em sequência. Como o dado referente à tecla pressionada é o segundo (último), esta questão dos dois dados não precisa ser tratada. Seu código mostrará sempre o dado que chega, e, portanto, mostrará o código de controle por um curtíssimo espaço de tempo, antes de mostrar o dado referente ao código da tecla pressionada. Tudo certo, a ideia é não gastar tempo demais com este laboratório;
- A terceira simplificação é que, ao contrário do que o padrão (*standard PS/2*) requer, seu código não precisará ter tratamento de erro. No padrão, o certo seria você contar os bits '1's dos dados e confrontar com o bit de paridade. Aqui você poderá ignorar o conteúdo do bit de paridade.

### Dica

Esta implementação pode ser muito simples através de uma FSM de poucos estados, sendo que em um deles, o dado que chega vai enchendo um registrador de deslocamento. Mãos à obra!