

TÓPICOS ESPECIAIS EM SISTEMAS EMBARCADOS

TRABALHO FINAL

O que fazer?

Faça uma biblioteca para um módulo de quatro displays de 7-segmentos que use um shift-register. No módulo que deixei na secretaria o shift-register é o CI 74HC595.

Descobri que o **Wokwi** tem o módulo de 4 displays de 7-segmentos. Pode ser feito nele ou no componente físico (disponível na secretaria).

De acordo como o site <https://www.arduinoecia.com.br/contador-com-modulo-display-4-digito-74HC595/> tem um código que usa o mesmo módulo que disponibilizei na secretaria, isto é, CI 74HC595 com 4 Displays de 7-segmentos. No código o autor usa uma biblioteca chamada de "TM74HC595Display.h". O que vocês teriam que fazer seria, basicamente substituir essa biblioteca pela biblioteca de vocês.

O código fonte da biblioteca "TM74HC595Display" está aqui:

https://github.com/AlexGyver/Arduino_speedometer/tree/master/display%20library

Creio que não é complicado entender o código acima. Mas talvez seja útil entender um pouco mais como funciona o 74HC595 juntamente com o display de 7-segmentos. Para tanto sugiro dar uma olhada no site <https://www.dobitaobite.com.br/construindo-um-pinball-parte-i/>. Neste post, o autor explica como integrar os quatro displays de 7-segmentos via o CI 74HC595.

Dentre as modificações que vocês devem fazer na biblioteca, sugiro que seja pelo menos duas:

1) No caso da função "clear", minha sugestão é que esta poderia conter como parâmetro qual dígito você quer apagar. Pelo o que eu vi no GitHub isso é trivial. Outra sugestão seria definir uma função "clear_all".

2) No caso da função "digit4(contador)" eu sugiro que seja "digit(display, valor)", onde "display" pode ser de 0 a 3, e "valor" seria o que seria impresso no display. Assumo que "valor" pode ser de 0 a F (hexadecimal). No GitHub eles definem algumas letras. Mas se tiver somente os 16 dígitos hexadecimais eu já estarei satisfeito.

O código para mostrar o uso da biblioteca de vocês poderia ser mostrar os dígitos (hexadecimais) em sequência. Ou seja, inicia com o 0123, depois 1234, depois 2345, ... depois CDEF, depois DEF0, depois EF01, e assim por diante, indefinidamente.

Ideal que vocês implementem no GitHub. Sugiro que vocês implementem usando classes.

Cronograma:

- 1) Entregue um documento PDF descrevendo as decisões de projeto para a construção da biblioteca até o dia 04 de julho de 2023 (terça) via Colabweb;
- 2) A apresentação final será no dia 05 de julho de 2023 (quarta).

Sites úteis:

- 1) **Criando uma Biblioteca para Arduino.** <https://blog.eletrogate.com/criando-uma-biblioteca-para-arduino/>
- 2) **Criando uma Biblioteca para Arduino – Parte 2.**
<https://blog.eletrogate.com/criando-uma-biblioteca-para-arduino-parte-2/>
- 3) **Módulo 74HC595 com Display 4 Dígitos.**
<https://www.makerhero.com/produto/modulo-74hc595-com-display-4-digitos/>
- 4) **Contador com módulo display 4 dígitos 74HC595.**
<https://www.arduinoecia.com.br/contador-com-modulo-display-4-digitos-74HC595/>
- 5) **Código fonte da biblioteca "TM74HC595Display"**
https://github.com/AlexGyver/Arduino_speedometer/tree/master/display%20library
- 6) **Explicação do 74HC595 com quatro displays de 7-segmentos**
<https://www.dobitaobYTE.com.br/construindo-um-pinball-parte-i/>