Prova Prática Studio Sol

29/05/2019

Documentação

Este documento foi criado com o propósito de tentar elucidar duvidas em relação o projeto.

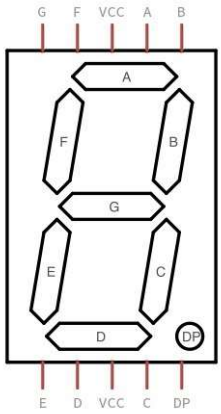
William Louzada Waillant

Bacharelando em Engenharia da Computação

01 – Funcionalidade e lógica da implementação

Foi proposto para a prova prática um relógio baseado em display de 7 segmentos, para implementar minha solução para o problema usei a linguagem de programação Python versão 3.7.

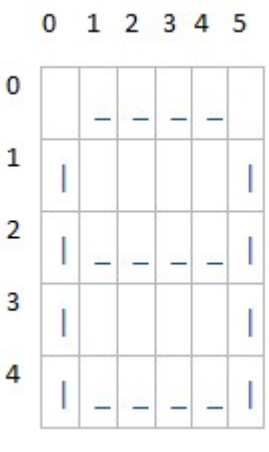
Para resolver o esse problema me baseie em um display de 7 segmentos real, usado na eletrônica, onde cada segmento do display acender se seu pino pré-estabelecido é energizado ou não, e com isso forma o número desejado, como podemos ver na imagem abaixo:



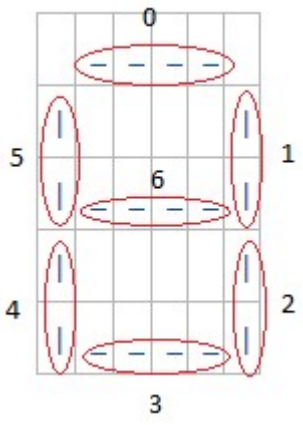
Com base nessas informações desenvolvi meu projeto de display como sendo uma matriz de caracteres (Img.1) formada de espaços vazios aonde posso apenas completar as linhas e colunas que desejar simulando o segmento acesso do display, o tamanho da matriz e a quantidade de caracteres a ser desenhado sempre é estabelicida pelo tamanho de entrada disponibilizada pelo usuario seguindo as regras de produção proposta pela prova.

Cada segmento é desenhado separadamente para que o numero possa ser formado, essa separação e demonstrada logo abaixo pelas imagem Img.2. Para desenhar o numero eu criei uma função que simula um multiplexador e me retorna um vetor de booleanos aonde cada posição do vetor corresponde a mesma posição do segmentos do meu display onde True significa que o segmente tem q ser desenhado e False que não.

Exemplo ao passar o número 4 para 0 multiplexador ele retorna um vetor com : (False,True,True,False,False,True,True).



Img.1



Img.2

Com a estrutura toda do display criada, só precisa passar os números que queira gerar, um por um, e depois concatenar suas matrizes para formar a estrutura de hora proposta na prova que é hh:mm:ss

Os pontos de separação foram gerados com duas letras “o” centralizada no meio do número com um espaço de acordo com o tamanho solicitado antes e depois.

Depois de arquitetar toda a estrutura do display temos que tratar da hora a ser exibida para o usuário e as configurações do display. O programa começa dando a hora atual do computador do usuário através da função datetime.now() do módulo datetime do python, no programa você pode alterar o valor do tamanho do display ao apertar as seta do teclado para o lado direito -> e para o lado esquerdo <- , também pode acessar as configurações do display apertando F1. Ao apertar F1 abre um menu onde você tem a opção de alterar o tamanho do display para qual deseja, além de pode alterar a timezone do seu relógio.

As timezones disponível para alteração do relógio são disponibilizadas no World Time Api, consumida pelo programa através do módulo requests. Pelo fato de haver um número muito alto de timezones no World Time Api eu dividi a escolha do usuário em região e cidade, primeiro mostro as regiões e o deixo escolher, depois mostro as cidades somente daquela região. Quando o usuário escolhe uma timezone a api me retorna várias características da timezone escolhida.

Para minha aplicação eu pego o tempo universal coordenado (UTC) daquela timezone, com isso eu atraso ou adianto a hora a ser exibida com base na hora atual dele. Como primeiro experimento do projeto eu tentei atualizar a hora da nova timezone de acordo com a hora que a api me dava, mas como a api era muito instável e a cada segundo novo precisava solicitar a nova hora vários solicitações não eram concluídas e com isso comprometendo o programa.

As ações das teclas no programa e a atualização do relógio em tempo real foi implementada através do módulo curses que permite alterações na linha do terminal e eventos sobre ele.

02 – Considerações do programa

1. Para executar o programa basta dar o comando “python relogio.py”
2. Deve-se usar o programa com o terminal em tela cheia. O modulo curses trata o terminal como uma matriz de tamanho relativo ao tamanho da janela do terminal, devido a alterações de tamanho do display durante a execução se a janela do terminal tiver menor não dará espaço.
3. No caso de execução do programa no Windows é necessário instalar o modulo curses antes, você pode baixar ele em https://www.lfd.uci.edu/~gohlke/pythonlibs/#curses , já em Linux ou mac o curses já vem nativo