Universidade de São Paulo
Escola de Engenharia de São Carlos
Departamento de Engenharia Elétrica e de Computação

Prática 3: Introdução a programação em sistemas embarcados

NOME: Nº USP:
NOME: Nº USP:

Resumo

Introdução a programação em Python, com uso de condicionais, laços de repetição, funções, conversão de tipo, importação de bibliotecas, manipulação de erros, e uso de gpio no Python da Raspberry.

Conceitos importantes:

Type casting, Timer, time, Import, Try Except, if, for, while, GPIO, Virtual environment.

Objetivo

O objetivo desta prática é a familiarização dos conceitos básicos que tange a linguagem de programação Python, abrangendo os conceitos de importação de bibliotecas, uso de estruturas condicionais e laços de repetição, funções definidas pelo usuário, métodos de conversão de tipo e métodos de manipulação de erros, além disso, o uso de GPIO da rasp a fim de demonstrar uma simples aplicação de acesso aos componentes de hardware da placa a partir do Python.

Aplicação

Portanto, a partir dos conceitos apresentados acima, a prática consiste em realizar um script em Python, responsável por receber um dado de entrada via terminal, e partir dele, realizar uma contagem regressiva, o convertendo para o formato minutos e segundos, e ao final da contagem, acender um led na placa e escrever uma mensagem no terminal indicando o fim da contagem.

Antes de executar qualquer script em python, e realizar a instalação de bibliotecas python, é importante que se crie um ambiente virtual, desta forma as bibliotecas estarão instaladas somente dentro deste ambiente e não em toda a máquina.

Um ambiente virtual python é responsável por isolar um diretório do computador do restante da máquina, desta forma, todos os comandos de instalação python executados dentro dele não afetarão o restante dos projetos, evitando conflitos em projetos que usem bibliotecas diferentes.

Além disso, com o uso de um ambiente virtual pode-se avaliar quais os requisitos de um projeto de forma simplificada, avaliando-se as instalações dentro somente deste ambiente, e não a partir de uma busca em código de todas as importações feitas.

Portanto, como primeira atividade a ser executada, deve-se criar um ambiente virtual em um diretório para o projeto, e acessá-lo, a partir disso pode-se iniciar os scripts.

Como está sendo utilizado a entrada de dados, no caso a variável de tempo a ser contado, é importante que se utilize métodos de manipulação de erros fornecidos pela linguagem, e métodos de estruturas condicionais, para verificar se os valores inseridos são realmente números ou se são valores de tempo possíveis.

Em caso de inserção de um valor inválido, o script não pode retornar um erro que gere uma saída forçada, e sim deve somente apresentar na tela qual tipo de erro e pedir novamente ao usuário uma entrada adequada.
A execução da contagem regressiva, a fim de manter a organização e modularização do código, deve ser feita em uma função, que recebe o parâmetro de tempo já testado do usuário.

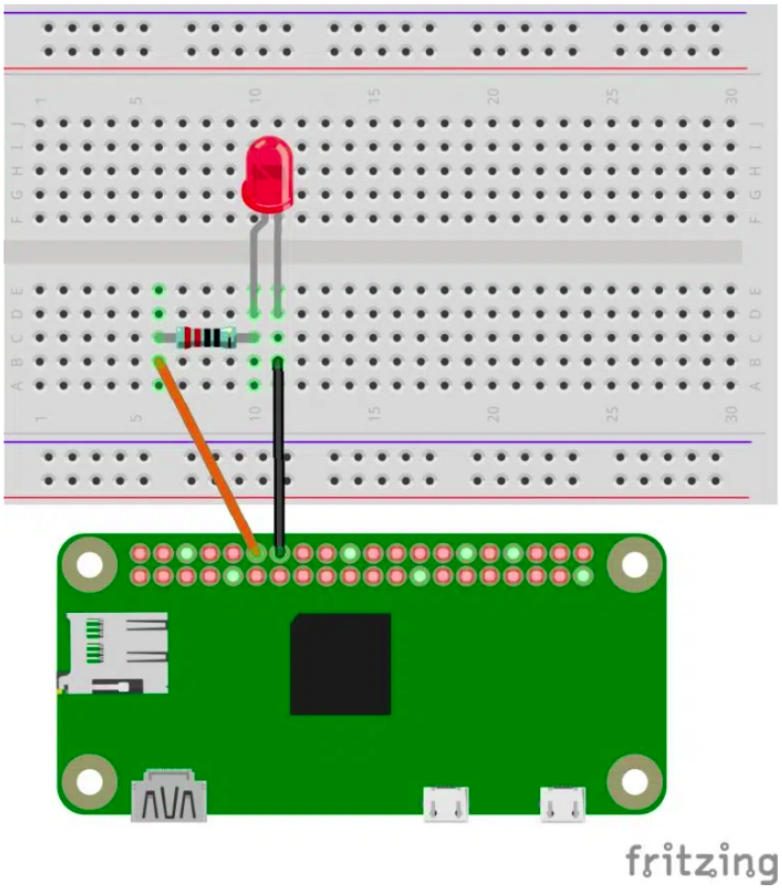
Atente-se a identação do código, uma vez que em Python, a indentação não somente serve para auxiliar na legibilidade do texto, mas também para definir se uma linha está no escopo de uma função, parâmetro ou laço.

Utilize a biblioteca time para gerar a base de tempo da contagem regressiva, de forma a garantir que o intervalo de tempo de contagem seja preciso.

Utilize Type casting a fim de garantir que o valor a ser contado esteja no formato desejado, e teste sobre ele as condições de erro.

Para impressão em terminal da contagem, utilize formatação de strings, de forma a demonstrar o valor de minutos e segundos no padrão de tempo MM : SS.

Para a conexão do LED a placa, utilize a conexão demonstrada abaixo:



O pinout com descrição e numeração de cada um dos pinos pode ser consultado a partir da internet a partir do site de pinout da rasp (ou também por meio do comando “pinout” no terminal).

Motivação

A manipulação de scripts em Python é de extrema importância para sistemas operacionais embarcados, principalmente os baseados em sistemas linux, uma vez que, na maioria das distribuições linux, o Python já vem pré-instalado, como é o caso do raspbian, que pôde ser consultado na prática anterior a sua versão instalada.

Além disso, em sistemas operacionais embarcados, devido a ampla utilização do Python em diversas áreas, criou-se uma comunidade grande em torno do desenvolvimento e otimização de bibliotecas e pacotes em Python, a fim de garantir que os projetos executados em Python possam ser competitivos com outras linguagens de programação, mantendo ainda a sua facilidade de aprendizado, com isto, a aplicação deste conhecimento em sistemas embarcados garante que o processo de aprendizado seja facilitado, e possua otimizações feitas pela comunidade para que possa ser executado tão eficientemente quanto outras linguagens de programação compiladas.

Outra importância da programação em Python está no uso de servidores, onde o Python pode gerenciar várias etapas do processo de uso e criação de servidores WEB, além da área de processamento de imagens, onde muito dos algoritmos modernos são feitos pensados em Python a fim de garantir que o acesso a estes algoritmos seja facilitado, dada a grande base de usuários, e ambas as aplicações podem ser aproveitadas em sistemas embarcados, principalmente os baseados em sistemas operacionais e SoC, onde geralmente se utiliza dos conceitos de redes e servidores, além de visão computacional.

Roteiro

- Antes de se utilizar o Python no computador ou na Raspberry, deve-se atentar a utilização de um ambiente virtual.
- Instale o pip a partir do terminal, com o comando apt e python3-pip
- Instale o venv a partir do terminal via apt e python3-venv
- Com isto, tem-se instalado em sua máquina o venv, a partir disto, pode-se iniciar a sua ativação:
    - Mova-se até o diretório desejado
    - Crie um venv nesse diretório com o nome sendo o os dois últimos dígitos do seu NUSP (em caso de dupla, utilize os dois últimos dígitos de ambos os integrantes) *python3 -m venv XXYY*
    - ative o venv no diretório, neste caso como está sendo utilizado o linux nos computadores e Rasps, o comando para ativação deve ser: source XXYY/bin/activate, lembre-se de alterar o valor XXYY pelo nome criado para o venv.
    - Uma vez ativado pode-se testar se o Python está funcional com o comando Python
    - Isto abrirá um terminal Python com a versão Python instalada no dispositivo, para sair do terminal Python utilize quit()
    - Com isto pode-se iniciar o projeto com os arquivos em Python e iniciar a atividade
    - Para que possa sair do venv, deve-se utilizar o comando: *deactivate*, no terminal.
- Faça a montagem do circuito para piscar o LED, conforme imagem acima, utilizando o Fritzing (necessário instalar na Rasp) e depois com os componentes usando a Protoboard.
- Escreva o programa em Python, conforme exemplo a ser demonstrado na aula, importando bibliotecas Python a serem usadas, configurando pinos GPIO,e programando o LED para piscar com intervalo inicial de 0,5 segundos. Note que o programa de exemplo utiliza “laço condicional infinito”.
- Faça o aprimoramento do programa de exemplo, de forma que um LED acenda somente após a contagem regressiva de um valor de tempo de entrada, em segundos, digitado no terminal. Implementando as funcionalidades descritas anteriormente (vide trecho destacado em amarelo acima: “time casting”; uso do

comando “print” para imprimir mensagem de erro quando de um valor de entrada errado; comando “input” para receber o valor de entrada que deve ser inteiro (int)).

- Enviar na tarefa o arquivo “.txt” contendo salvo o histórico de comandos usado no ambiente virtual criado para execução da atividade.