

Texto Complementar

Disciplina:	Introdução à Programação Estruturada
Professor (a):	Salatiel Luz Marinho

Fonte: <https://www.embarcados.com.br/zerynth/>

Autor: Traduzido e adaptado por Equipe Embarcados

Zerynth: Um Framework Python para dispositivos IoT

Uma das maiores inovações na construção civil foi a ideia de pré-fabricar blocos de construção em uma fábrica, meses antes de serem necessários no local de trabalho. Isso permitiu que o trabalho não só fosse concluído rapidamente (uma vez que o solo já estivesse sendo trabalhado no local), mas também serviu para aumentar a qualidade. O desenvolvimento de software de dispositivos embarcados de IoT pode alavancar um conceito semelhante graças às estruturas de desenvolvimento de software (também chamadas de kits de desenvolvimento de software ou SDKs). Para entender o que é um SDK, pense em apenas adquirir a madeira para construir uma casa do zero. Você teria que derrubar árvores, secar a madeira em um forno e cortá-la em madeira utilizável. Ou você pode visitar um depósito de madeira e simplesmente comprar a madeira que é praticamente perfeita para a tarefa em questão. Você tem autonomia para gastar menos tempo em tarefas sem valor agregado (embora importantes) e mais tempo criando a casa que o usuário final deseja.

Zerynth SDK

SDKs são, para o desenvolvimento de software, equivalentes a comprar madeira em um depósito de madeira em vez de derrubar árvores você mesmo. Uma opção do SDK para desenvolvedores embarcados que procuram programar dispositivos IoT em Python é a plataforma Zerynth. O Zerynth suporta muitas arquiteturas populares de microcontroladores de 32 bits, incluindo SAMD21, ESP32, ESP8266 e NRF52832, para citar alguns. O SDK está disponível para instalação em Windows, Mac OS e Linux. O Zerynth SDK é composto de dois componentes principais: o Zerynth Toolchain e o Zerynth Studio Integrated Development Environment (IDE).

O Zerynth Toolchain (ZTC) é a ferramenta de linha de comando que serve como o núcleo para o desenvolvimento de firmware, depuração e gerenciamento de dispositivos dentro do ecossistema Zerynth. Zerynth Studio é uma interface gráfica que fica no topo da interface de linha de comando do ZTC. Ele fornece um editor de código gráfico e interface de depuração e uma infinidade de exemplos de código Python para ajudar a começar a usar o Zerynth. Um recurso interessante do IDE é o suporte para os chamados dispositivos

virtuais, que permitem desenvolver e verificar o código da aplicação sem um dispositivo físico conectado ao computador host local.

O editor de código apresenta muitos recursos modernos, incluindo destaque de sintaxe, preenchimento automático e snippets inteligentes. Outros recursos que devem deixar os desenvolvedores animados incluem suporte integrado para repositórios Git e um gerenciador de pacotes integrado. Os desenvolvedores têm acesso às bibliotecas de código Python oficiais fornecidas pela Zerynth, bem como às bibliotecas enviadas pela comunidade de usuários. A capacidade de acessar repositórios de código-fonte comprovado e confiável permite que os desenvolvedores se concentrem na funcionalidade de valor agregado para seus clientes, em vez de perder tempo reinventando a roda no código Python comumente usado. O poder das bibliotecas de código-fonte é amplificado por um segundo componente importante do ecossistema Zerynth, o Sistema Operacional Zerynth (OS).

Zerynth OS

Zerynth OS é um sistema operacional em tempo real (RTOS) multithreaded ajustado para funcionar em uma variedade de arquiteturas de microcontroladores de 32 bits com memória limitada. Ele contém pilhas de software altamente personalizadas para muitos protocolos de comunicação, incluindo Wi-Fi®, Bluetooth® e LoRa. Ele também fornece funcionalidade de gerenciamento de segurança e energia. O uso de camadas de abstração para as interações de hardware e RTOS permitem que o Zerynth OS seja altamente independente das plataformas de microcontrolador subjacentes. O núcleo do Zerynth OS é a Máquina Virtual Zerynth (VM), que executa os scripts Python escritos pelo usuário sobre qualquer número de RTOSes e plataformas de hardware. Os RTOS prontos para uso incluem FreeRtos e CHIBIOS.

Por meio do uso de uma chamada VM Operative System Abstraction Layer (VOSAL), os desenvolvedores podem criar aplicações de alto desempenho em execução no Zerynth VM que aproveitam recursos de nível de sistema operacional, como semáforos e threads, independentemente do RTOS subjacente selecionado. Com relação à abstração de hardware, o Zerynth OS fornece um mecanismo (VM Hardware Abstraction Layer ou VHAL) para interagir com muitos dos periféricos encontrados em microcontroladores modernos, como interrupções, GPIO, I2C, SPI, ADC, PWM e UART. O VHAL, escrito de forma personalizada para cada microcontrolador compatível, fornece uma interface de software padrão única para todos os periféricos mencionados anteriormente. Em outras palavras, escreva o código Python da aplicação uma vez e ele funcionará em várias plataformas de hardware, desde que o dispositivo tenha os periféricos.

Zerynth Device Manager

Claro, desenvolver código Python para um dispositivo IoT é apenas metade da batalha. Eventualmente, um dispositivo deve sair para o mundo e se conectar à Internet para realizar o propósito pretendido. Para auxiliar nas várias funções associadas ao campo de um dispositivo IoT (por exemplo, provisionar remotamente, organizar, monitorar e gerenciar vários dispositivos em todo o mundo), o Zerynth Device Manager (ZDM) é fornecido aos desenvolvedores também.

Por meio do ZDM, os dispositivos podem ser integrados com segurança à nuvem. O gerenciamento de dispositivos em todo o seu ciclo de vida para incluir chamadas de procedimento remoto (RPC) e atualizações de Firmware Over-the-Air (FOTA) também é possível usando APIs REST via ZDM. O ZDM fornece ferramentas para agregar e armazenar os dados coletados pelos dispositivos IoT em campo. Ele também fornece gerenciamento de eventos e um mecanismo de alarme para responder aos dados brutos à medida que são recebidos. O ZDM pode ser usado como um serviço baseado em nuvem ou pode ser instalado no local para aqueles clientes que têm necessidades comerciais ou de segurança exclusivas.

Se, no entanto, um desenvolvedor requer o uso de um provedor de serviços de nuvem terceirizado (por exemplo, Amazon Web Services, Microsoft Azure, Google Cloud Platform, Ubidots, The Things Network), é uma simples questão de alterar uma linha de código em o aplicativo Python, não uma reescrita de código completa.

Python e Zerynth: Poder para quem codifica

Python é uma linguagem de programação poderosa que continua ganhando força em casos de uso além das aplicações de desktop tradicionais. Os desenvolvedores de eletrônicos embarcados, especialmente dispositivos IoT, podem se beneficiar significativamente com a adoção de estruturas de desenvolvimento de software como o Zerynth. As estruturas garantem a consistência e podem até contribuir para melhorar o desempenho, o gerenciamento de energia e a segurança. Quando combinados com a exclusividade e os recursos da linguagem Python, os frameworks capacitam os desenvolvedores a construir sistemas embarcados da mais alta qualidade, eliminando o trabalho pesado, fornecendo bibliotecas de código-fonte comprovadas para tarefas repetitivas e recursos gratuitos para se concentrar na funcionalidade voltada para o cliente.