

Configuração do SDK Raspberry Pi Pico no macOS

Introdução

A instalação do ambiente de desenvolvimento no macOS é bastante semelhante ao processo para sistemas Linux. Este guia foi testado em computadores Mac com processadores Intel e Apple Silicon.

Instalando as ferramentas necessárias

Para começar, você precisará do Homebrew, um gerenciador de pacotes popular no macOS. Caso ainda não o tenha instalado, execute o seguinte comando no terminal:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Com o Homebrew instalado, adicione as ferramentas necessárias usando os comandos abaixo:

```
brew install cmake
```

```
brew tap ArmMbed/homebrew-formulae
```

```
brew install arm-none-eabi-gcc
```

Estruturando as pastas do projeto

Para facilitar a organização, recomendamos criar uma estrutura de pastas específica para o desenvolvimento com o Raspberry Pi Pico. Por exemplo, você pode criar uma pasta chamada `~/Developer/pico`. Dentro dela, organize subpastas como `_projects` (para seus projetos) e `lib` (para bibliotecas externas).

Configuração do SDK Raspberry Pi Pico no macOS

Baixando o SDK e projetos de exemplo

Com as pastas criadas, baixe o SDK oficial e os projetos de exemplo diretamente do repositório GitHub da Raspberry Pi Foundation. No terminal, execute os comandos:

```
cd ~/Developer/pico
git clone -b master https://github.com/raspberrypi/pico-sdk.git
cd pico-sdk
git submodule update --init
cd ..
git clone -b master https://github.com/raspberrypi/pico-examples.git
```

Além disso, você pode optar por baixar repositórios adicionais, como o de bibliotecas extras e o playground, que incluem mais exemplos de uso.

Configurando o SDK para compilar exemplos

Antes de compilar os projetos, você precisa definir a variável de ambiente PICO_SDK_PATH para apontar para o diretório do SDK. Faça isso com o seguinte comando:

```
export PICO_SDK_PATH=$HOME/Developer/pico/pico-sdk
```

Para evitar repetir este comando sempre que abrir o terminal, adicione-o ao arquivo de configuração do shell (~/.bashrc ou ~/.zshrc).

Agora, navegue até a pasta de exemplos e configure o ambiente de build com cmake:

Configuração do SDK Raspberry Pi Pico no macOS

```
cd ~/Developer/pico/pico-examples
```

```
mkdir build && cd build
```

```
cmake ..
```

Se tudo estiver correto, você verá mensagens indicando que a configuração foi concluída. Para compilar o exemplo básico blink, use:

```
make blink
```

Instalando o OpenOCD

O OpenOCD é uma ferramenta essencial para depuração via SWD. Instale as dependências com o Homebrew:

```
brew install libtool automake libusb wget pkg-config gcc texinfo
```

Baixe e compile o OpenOCD com os seguintes comandos:

```
cd ~/Developer/pico
```

```
git clone https://github.com/raspberrypi/openocd.git --branch picoprobe --depth=1
```

```
cd openocd
```

```
export PATH="/usr/local/opt/texinfo/bin:$PATH"
```

```
./bootstrap
```

```
./configure --enable-picoprobe --disable-werror
```

Configuração do SDK Raspberry Pi Pico no macOS

```
make -j4
```

Após compilar, execute o comando `src/openocd` para verificar a instalação.

Comunicação serial com o Raspberry Pi Pico

Para acessar a interface serial do Pico no macOS, utilize a ferramenta minicom. Instale-a com:

```
brew install minicom
```

Descubra o dispositivo serial correspondente ao Pico (geralmente localizado em `/dev/tty.usbmodemXXXX`) usando:

```
ls /dev/tty.usbmodem*
```

Conecte-se ao dispositivo com o comando:

```
minicom -D /dev/tty.usbmodemXXXX 115200
```

Para encerrar o minicom, pressione CTRL+A, depois Z, e em seguida X.