

# Usando Linux para programar microcontroladores STM32

All source code including text of this document is  
available at <https://github.com/hans-jorg/XXXXX>

# MIT License

Copyright (c) 2020 Hans Jorg Andreas Schneebeili

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Versão	<b>1.1</b>
Data	<b>22/10/2020</b>

## Introdução

Processadores STM32 são processadores fabricados pela ST, que usam arquitetura Core Cortex-M.

Para se gerar código para processadores STM32 são necessárias as seguintes ferramentas:

- Compiladores e ligadores
- Mecanismos para transferir código para a placa
- Ferramentas para depuração

Altamente desejáveis são:

- Definição de registradores e outras características de processadores
- Bibliotecas para acesso a hardware
- Bibliotecas auxiliares
- Kernels
- Sistemas operacionais

## Compiladores

O mais comum e fácil de se obter uma versão do GNU C com as modificações necessárias para gerar código para Cortex-M. No processo de desenvolvimento desta versão participa a própria ARM (embora ela seja dona da Keil, uma fornecedora de software para desenvolvimento concorrente).

Existem duas maneiras para instalar o software.

### Baixando e descompactando o pacote

1. Baixar no sítio <https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm> o pacote específico para linux, atualmente com o nome gcc-arm-none-eabi-9-2020-q2-update-x86\_64-linux.tar.bz2. Criar uma pasta temporária e fazer o download, no caso abaixo, por linha de comando, substituindo TEMP pelo caminho da pasta.

```
mkdir TEMP
cd TEMP
wget https://developer.arm.com/-/media/Files/downloads/gnu-rm/9-2020q2/gcc-arm-none-eabi-9-2020-q2-update-x86_64-linux.tar.bz2
```

2. Ir para a pasta /opt (usada para instalações de software fora dos pacotes da distribuição) e descompactar o pacote baixado.

```
cp /opt
tar xvf TEMP/gnu-rm/9-2020q2/gcc-arm-none-eabi-9-2020-q2-update-x86_64-linux.tar.bz2
```

3. Acrescentar a pasta na variável PATH. Para que isto seja feito de forma permanente, editar o arquivo ~/.bashrc e acrescentar a linha abaixo.

```
ARMGCCDIR=/opt/gnu-rm/9-2020q2/gcc-arm-none-eabi-9-2020-q2-update
PATH=$ARMGCCDIR/bin:$PATH
```

### Usando as ferramentas de gerenciamento de pacotes

A maioria das distribuições linux disponibiliza pacotes com o compilador e ferramentas para desenvolvimento. Em geral, os repositórios são lentos na atualização e os pacotes baixados diretamente do sítio acima são mais atualizados.

1. Atualizar o sistema de gerenciamento de pacotes com o comando

```
sudo apt-get update
```

2. Instalar o pacote com o comando

```
sudo apt-get install gcc-arm-none-eabi binutils-arm-none-eabi
```

Neste esquema, os pacotes são automaticamente atualizados junto com o restante do sistema.

## Software instalado

Após o procedimento acima, ficam disponíveis diversos aplicativos como os listados abaixo.

<code>arm-none-eabi-gcc</code>	Compilador C
<code>arm-none-eabi-nm</code>	Lista símbolos
<code>arm-none-eabi-ar</code>	Gerenciador de bibliotecas
<code>arm-none-eabi-objcopy</code>	Converte objeto para outros formatos
<code>arm-none-eabi-as</code>	Assemblador/Montador
<code>arm-none-eabi-objdump</code>	Lista código objeto
<code>arm-none-eabi-c++</code>	Compilador C++
<code>arm-none-eabi-size</code>	Lista tamanho do código
<code>arm-none-eabi-cpp</code>	Preprocessador
<code>arm-none-eabi-gdb</code>	Depurador
<code>arm-none-eabi-strings</code>	Lista cadeias de caracteres no código
<code>arm-none-eabi-strip</code>	Elimina informações desnecessárias no arquivo código
<code>arm-none-eabi-ld</code>	Ligador
<code>arm-none-eabi-gdb</code>	Depurador

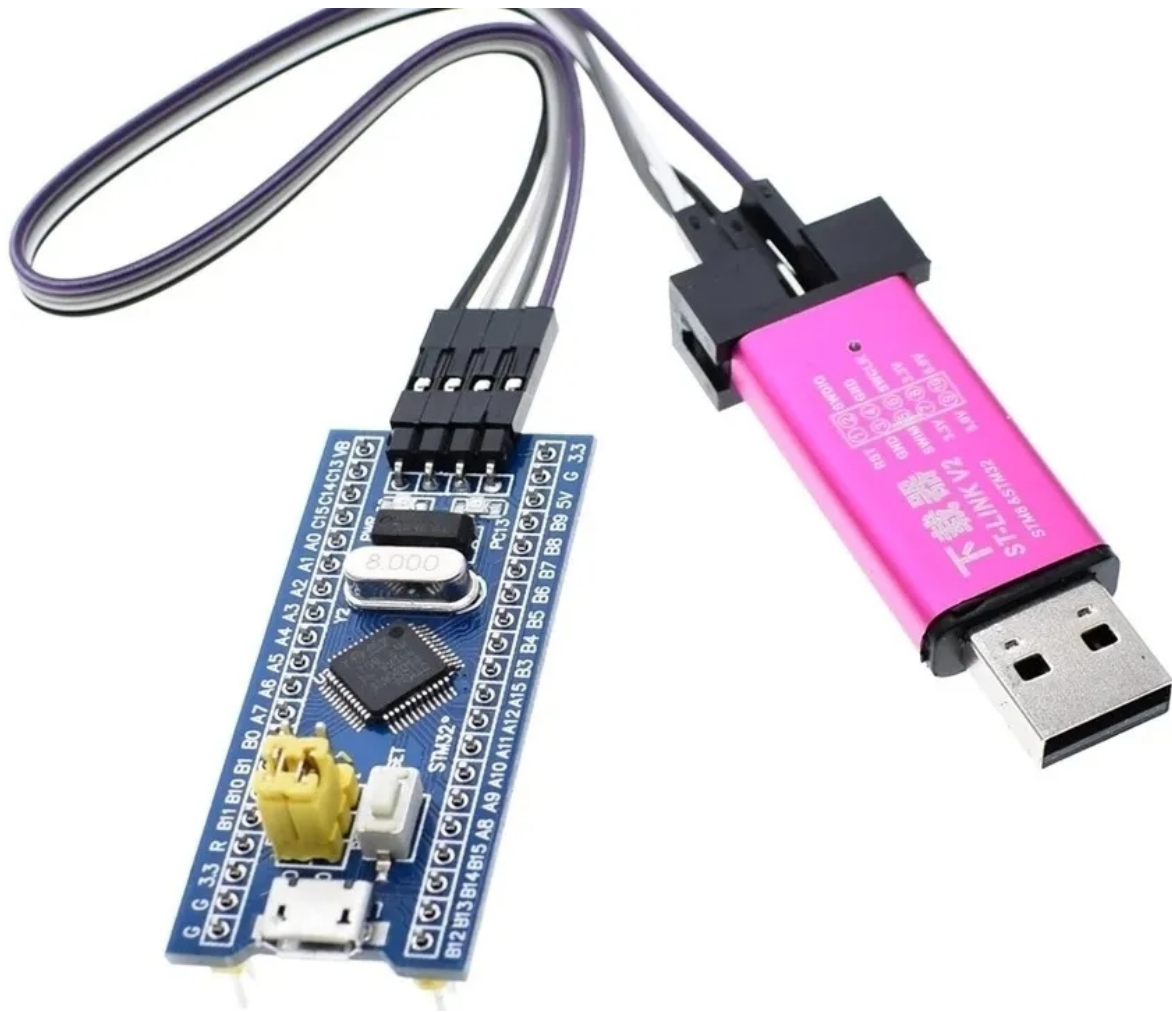
## Mecanismo de transferência de código para placa

Os mecanismos para transferência de código envolvem hardware e software. Além de transferir o código, um aplicativo no PC pode acessar o processador alvo, alterar o conteúdo de sua memória flash, reiniciar o processador, interromper seu funcionamento, alterar registradores e conteúdo da memória RAM. Para isto, é necessário um dispositivo que se comunica com o PC através de uma interface USB e com o processador através de uma interface de depuração.

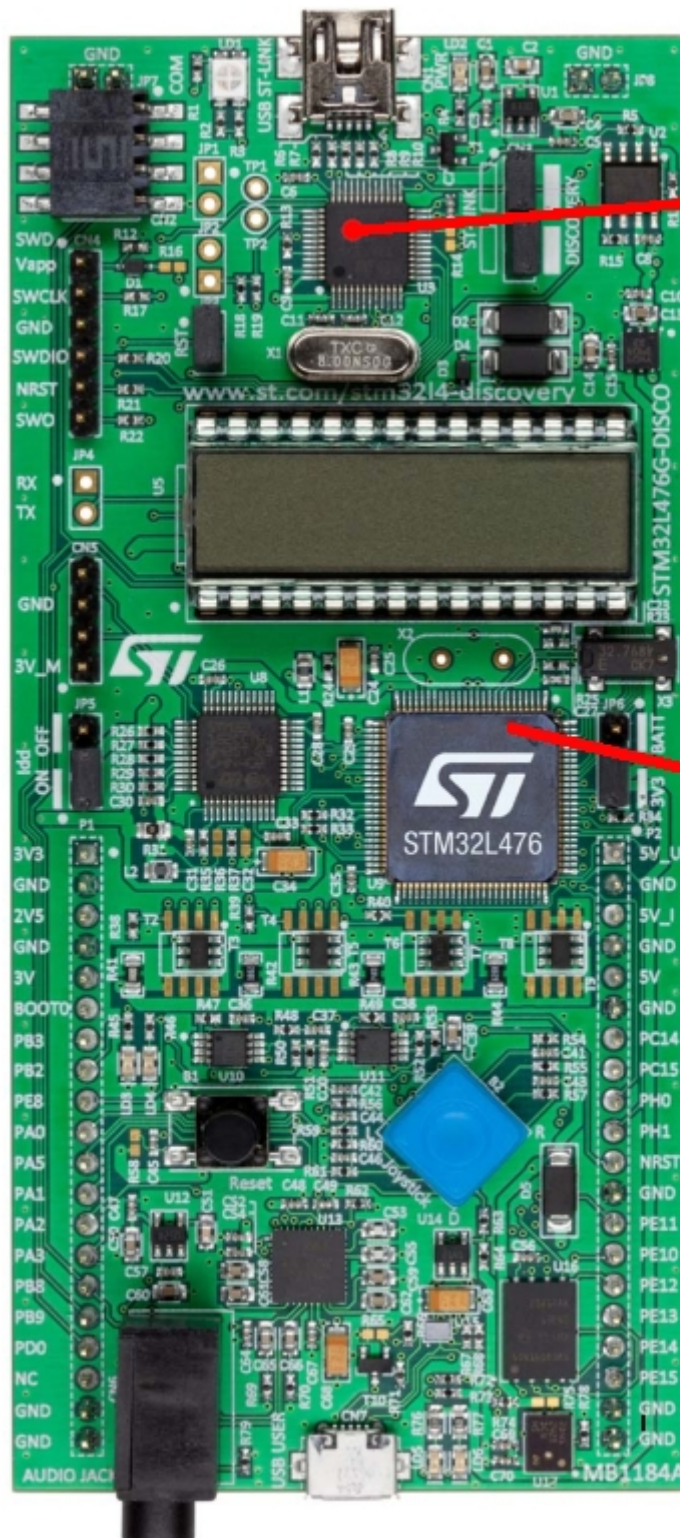
Neste dispositivo há um processador hospedeiro com um software específico. Em outras placas de desenvolvimento, é necessário um periférico USB, que conectado ao PC e a placa de desenvolvimento, permite as operações acima. No caso ST, eles são chamadas ST-LINK<sup>1</sup>.

---

<sup>1</sup> [http://www.st.com/content/st\\_com/en/products/development-tools/hardware-development-tools/development-tool-hardware-for-mcus/debug-hardware-for-mcus/debug-hardware-for-stm32-mcus/st-link-v2.html](http://www.st.com/content/st_com/en/products/development-tools/hardware-development-tools/development-tool-hardware-for-mcus/debug-hardware-for-mcus/debug-hardware-for-stm32-mcus/st-link-v2.html)



No caso das placas Discovery este dispositivo está incorporado a placa, que tem dois processadores, um hospedeiro e outro, o alvo. O processador hospedeiro implementa um protocolo compatível com as placas ST-LINK.



**STLINK**  
STM32F103CBT6

**TARGET**  
STM32L476

### Software OpenOCD

Pode ser instalado através de uma ferramenta de gerenciamento de pacotes usando o comando abaixo  
`sudo apt-get install openocd`

Deve ser, no entanto, verificada qual a versão instalada, pois pode haver atrasos grandes entre a versão no repositório e última versão disponível no site [www.openocd.org](http://www.openocd.org). No caso da versão disponível no repositório ser muito antiga, deve ser baixado o arquivo fonte no site [www.openocd.org](http://www.openocd.org) (deve ser

escolhida uma versão compactada com zip ou bzip2).

Para descompactar o arquivo baixado deve ser usado o comando correspondente

```
unzip ~/Downloads/openocd-0.10.0.zip
```

ou

```
tar -xjvf ~/Downloads/openocd-0.10.0.tar.bz2
```

Para compilar, deve ser usada a sequência de comandos

```
cd openocd-0.9.0
./configure --prefix=/opt
make
sudo make install
```

Pode acontecer que seja necessário instalar algumas dependências.

## Software STLINK

Este software permite gravação da flash e depuração.

Ele pode ser obtido no sítio <https://github.com/stlink-org/stlink>. Para instalá-lo seguir os passos abaixo.

1 – Instalar os pacotes necessários,

```
sudo apt-get install git libusb-1.0-0 libusb-1.0-0-dev pkg-config
```

2 – Clonar o repositório usando git

```
git clone https://github.com/texane/stlink.git
```

2 – Compilar

```
cd stlink
./autogen.sh
./configure
make
```

4 – Instalar usando make (ou checkinstall, se desejar mais tarde usar ferramenta de gerenciamento de pacotes)

```
make install
```

## Software STM32Flash

Este software somente permite a gravação da flash usando-se uma interface serial entre o PC e a placa de desenvolvimento e um procedimento especial de inicialização (boot). Todos os STM32 tem um software em uma memória apenas de leitura, que grava a flash a partir de uma interface serial. Este software é ativado mantendo-se em nível alto um determinado pino.

1. Baixar o software acessando o sítio <http://sourceforge.net/projects/stm32flash/> numa pasta DOWNLOAD. (Substitua DOWNLOAD pelo caminho de uma pasta).

2. Descompactá-lo em uma pasta de trabalho.

```
mkdir stm32flashwork
cd stm32flashwork
tar xzvf DOWNLOAD/stm32flash-x.y.tar.gz
```

3. Compilar

```
make
```

1. Instalar usando make (ou checkinstall, se desejar usar a ferramenta de gerenciamento de

pacotes).

```
make install
```

### ***Permissões para STLINK***

Para evitar que se tenha que usar a senha de root para se usar o software, pode-se criar uma regra para permitir o acesso à interface USB. Para tal, crie um arquivo `/etc/udev/rules/45-usb-stlink-v2.rules` com o conteúdo abaixo.

```
SUBSYSTEM=="usb", ATTR{idVendor}=="0483", ATTR{idProduct}=="3748", MODE="0666"
```

Para isto ter efeito, em geral é necessário se reiniciar o sistema. Em alguns casos, bastará desplugar o cabo USB, reiniciar o serviço com o comando abaixo e reconectar então o cabo.

```
sudo service udev restart
```