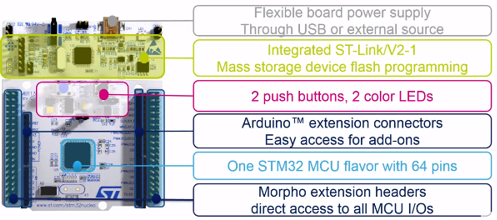
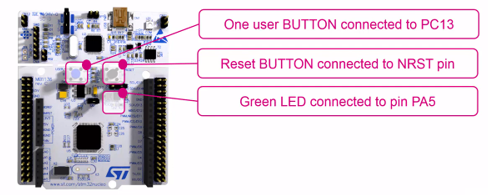
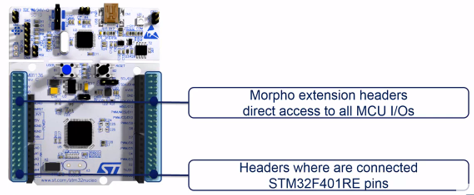
STM32F401RE NUCLEO FEATURES



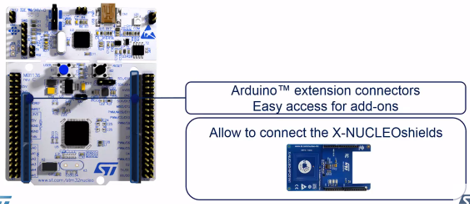
STM32F401RE NUCLEO USER IOS



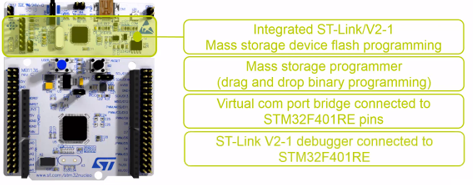
STM32F401RE NUCLEO HEADERS

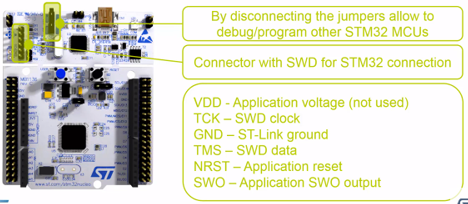


STM32F401RE NUCLEO EXTENSIONS

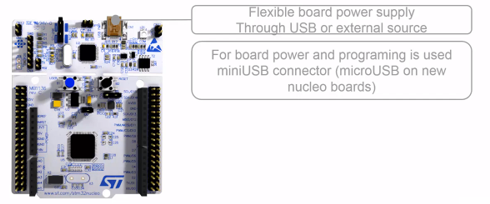


STM32F401RE NUCLEO DEBUG





STM32F401RE NUCLEO POWER



The STM32F401RE is a microcontroller belonging to the STM32F4 series, which is part of the STM32 family of 32-bit ARM Cortex-M based microcontrollers. Here are some key features and characteristics of the STM32F401RE:

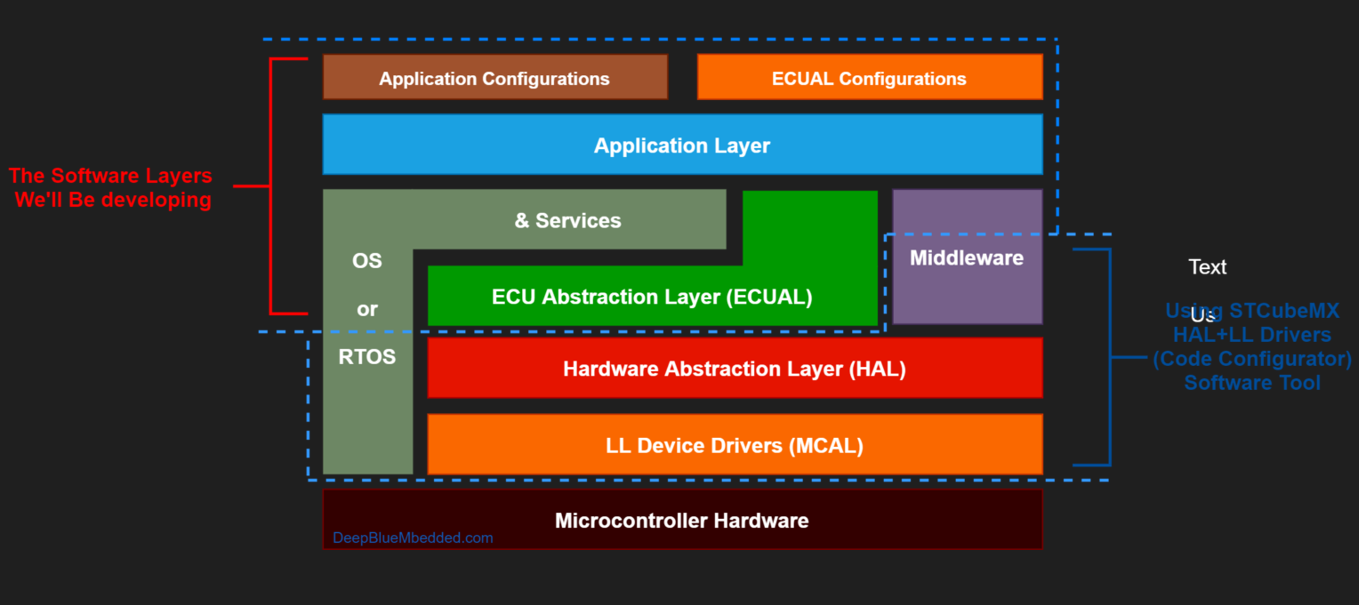
1. **Microcontroller Architecture:**
   * ARM Cortex-M4 core running at up to 84 MHz
   * 32-bit RISC architecture.
2. **Memory:**
   * Flash memory: Up to 512 KB.
   * RAM: Up to 96 KB.
3. **Peripherals:**
   * Multiple General-Purpose Timers (TIM2, TIM3, TIM4, TIM5, TIM9 to TIM14).
   * Advanced Control Timers (TIM1, TIM8).
   * Universal Synchronous/Asynchronous Receiver/Transmitter (USART) modules.
   * Serial Peripheral Interface (SPI) and Inter-Integrated Circuit (I2C) interfaces.
   * Analog-to-Digital Converter (ADC) with multiple channels.
   * Digital-to-Analog Converter (DAC).
   * USB 2.0 Full-Speed interface.
4. **GPIO (General-Purpose Input/Output):**
   * Multiple GPIO pins distributed across different ports (GPIOA, GPIOB, GPIOC, GPIOD, GPIOE, GPIOH).
5. **Clocks:**
   * Internal 16 MHz RC oscillator.
   * Internal 32 kHz oscillator for RTC.
   * PLL (Phase-Locked Loop) for clock frequency multiplication.
6. **Communication Interfaces:**
   * USART, SPI, I2C, CAN, USB, etc., for communication with other devices.
7. **Analog Components:**
   * 12-bit ADC with multiple channels.
   * 12-bit DAC.
8. **Power Management:**
   * Low-power modes to optimize energy consumption.
9. **Development Support:**
   * Development tools and IDEs such as STM32CubeMX, STM32CubeIDE, and STM32Cube HAL.
   * Debugging support with features like JTAG/SWD.
10. **Packages:**
    * Available in different package options, including LQFP64, LQFP100, and others.

The STM32Cube Hardware Abstraction Layer (HAL), an STM32 abstraction layer embedded software ensuring maximized portability across the STM32 microcontroller. The HAL is available for all the hardware peripherals.

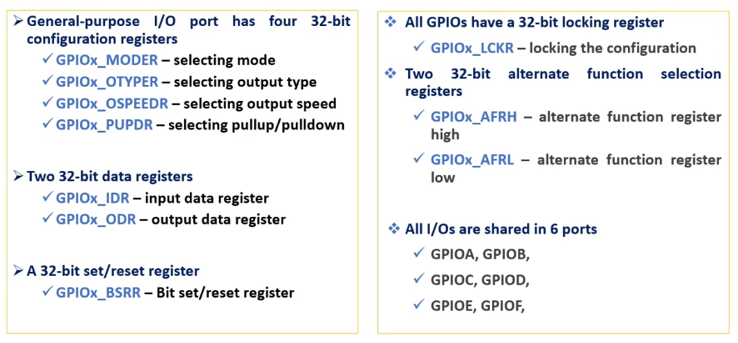
The low-layer APIs (LL) offering a fast light-weight expert-oriented layer that is closer to the hardware than the HAL. The LL APIs are available only for a set of peripherals.

The HAL and LL are complementary and cover a wide range of applications requirements:

* The HAL offers high-level and feature-oriented APIs, with a high-portability level. They hide the MCU and peripheral complexity to end-user.
* The LL offers low-level APIs at registers level, with better optimization but less portability. They require deep knowledge of the MCU and peripherals specifications.



General-purpose I/Os (GPIO) Each general-purpose I/O port has four 32-bit configuration registers (GPIOx\_MODER, GPIOx\_OTYPER, GPIOx\_OSPEEDR and GPIOx\_PUPDR), two 32-bit data registers (GPIOx\_IDR and GPIOx\_ODR) and a 32-bit set/reset register (GPIOx\_BSRR). In addition all GPIOs have a 32-bit locking register (GPIOx\_LCKR) and two 32-bit alternate function selection registers (GPIOx\_AFRH and GPIOx\_AFRL).



To use MicroPython on an STM32 microcontroller, you can follow these general steps. Keep in mind that the exact process may vary based on your specific STM32 board model and MicroPython version.

**Prerequisites:**

1. **STM32 Board:**
   * Ensure you have an STM32 development board. Popular models include STM32F103, STM32F407, STM32F401, etc.
2. **ST-Link Programmer:**
   * You may need an ST-Link programmer to flash MicroPython firmware onto the STM32 board.
3. **MicroPython Firmware:**
   * Download the MicroPython firmware binary for your specific STM32 board from the MicroPython website or GitHub releases.

**Steps:**

1. **Connect ST-Link Programmer:**
   * Connect the ST-Link programmer to your STM32 board. Make sure to connect the appropriate pins (SWD pins) for programming.
2. **Install ST-Link Utility:**
   * Install the ST-Link utility on your computer. This utility is used to program the firmware onto the STM32 board. You can download it from the STMicroelectronics website.
3. **Flash MicroPython Firmware:**
   * Use the ST-Link utility or other flashing tools to program the MicroPython firmware binary onto the STM32 board. This involves selecting the firmware binary file and initiating the flashing process.
4. **Connect to Serial Console:**
   * Connect to the serial port of your STM32 board. This may involve using a USB-to-Serial converter or connecting to a built-in USB-to-serial converter on your board.
5. **Open Serial Terminal:**
   * Use a terminal emulator (e.g., PuTTY, Minicom, or **screen** on Unix-like systems) to open a serial connection to the STM32 board. Configure the terminal with the appropriate settings (baud rate, data bits, stop bits, parity).
6. **Interact with MicroPython:**
   * Once connected to the serial port, you should see the MicroPython REPL (Read-Eval-Print Loop) prompt. Here, you can enter Python commands and scripts.
7. **Explore MicroPython on STM32:**
   * Experiment with MicroPython on your STM32 board. You can interact with GPIO pins, sensors, and other peripherals using MicroPython.

**Example Commands (MicroPython REPL):**

Here are some basic commands you can try in the MicroPython REPL:

pythonCopy code

# Blink an LED connected to Pin PA5 import machine import time led = machine.Pin('PA5', machine.Pin.OUT) while True: led.toggle() time.sleep\_ms(500)

These steps provide a general overview of the process. Always refer to the specific documentation for your STM32 board and MicroPython version for detailed and accurate instructions.

When connecting the ST-Link programmer to your STM32 board, you need to identify and connect the appropriate pins for SWD (Serial Wire Debug) programming. SWD is a standard protocol used for programming and debugging ARM Cortex microcontrollers, including STM32 devices.

Here are the typical connections for SWD:

1. **SWDIO (Serial Wire Debug Input/Output):**
   * Connect the SWDIO pin on the ST-Link programmer to the SWDIO pin on the STM32 board.
2. **SWCLK (Serial Wire Clock):**
   * Connect the SWCLK pin on the ST-Link programmer to the SWCLK pin on the STM32 board.
3. **GND (Ground):**
   * Connect the GND pin on the ST-Link programmer to the ground (GND) on the STM32 board.
4. **VCC (Power):**
   * Connect the VCC pin on the ST-Link programmer to the power source on the STM32 board. Ensure that the voltage levels are compatible.
5. **NRST (Reset):**
   * Connect the NRST pin on the ST-Link programmer to the NRST (Reset) pin on the STM32 board. This pin is used to reset the microcontroller during programming.

**Steps:**

1. **Identify the SWD Pins on Your STM32 Board:**
   * Refer to the datasheet or reference manual of your specific STM32 board to identify the SWD pins. Commonly, they are labeled as SWDIO, SWCLK, NRST, GND.
2. **Connect the ST-Link Programmer:**
   * Connect the corresponding pins on the ST-Link programmer to the identified SWD pins on your STM32 board.
3. **Power Up the STM32 Board:**
   * Ensure that your STM32 board is powered up. If needed, connect the VCC pin on the ST-Link programmer to the power source on the STM32 board.
4. **Check Voltage Levels:**
   * Verify that the voltage levels of the ST-Link programmer and the STM32 board are compatible. Mismatched voltage levels can damage the components.
5. **Connect the ST-Link Programmer to Your Computer:**
   * Connect the ST-Link programmer to your computer using USB. Install any necessary drivers for the programmer.
6. **Use ST-Link Utility or Other Flashing Tool:**
   * Use the ST-Link Utility or another flashing tool to load the MicroPython firmware onto the STM32 board. Follow the tool-specific instructions for flashing.
7. **Verify Connections:**
   * Double-check all connections to ensure that they are secure and correctly aligned.

Once the connections are established and the firmware is flashed, you should be able to interact with MicroPython on your STM32 board. Remember to consult the documentation for your specific STM32 board and MicroPython version for any board-specific considerations.

The ST-Link programmer is a hardware tool designed and manufactured by STMicroelectronics for programming and debugging STM32 microcontrollers. It is widely used in the embedded systems development community, especially when working with STM32 microcontrollers. The ST-Link programmer provides a convenient way to program the flash memory of STM32 devices and perform debugging operations.

Key features of the ST-Link programmer include:

1. **Programming Interface:**
   * The ST-Link programmer typically supports the SWD (Serial Wire Debug) interface, a two-wire protocol used for programming and debugging ARM Cortex microcontrollers.
2. **Debugging Capabilities:**
   * In addition to programming flash memory, the ST-Link programmer enables real-time debugging of STM32 applications. Developers can set breakpoints, step through code, and inspect variables during the debugging process.
3. **Compatibility:**
   * The ST-Link programmer is compatible with a wide range of STM32 microcontrollers. Different STM32 series and models may use the same ST-Link programmer for programming and debugging.
4. **ST-Link Utility:**
   * STMicroelectronics provides the ST-Link Utility, a software tool that works in conjunction with the ST-Link programmer. This utility allows users to program and erase the flash memory of STM32 devices, perform memory inspections, and more.
5. **Firmware Updates:**
   * The ST-Link programmer's firmware can be updated to support new features, improvements, and compatibility with the latest STM32 microcontrollers.
6. **Versatility:**
   * The ST-Link programmer can be used with various integrated development environments (IDEs) and toolchains, making it versatile for different software development workflows.

When working with STM32 microcontrollers and MicroPython, the ST-Link programmer is often used to flash the MicroPython firmware onto the STM32 board, enabling developers to run Python scripts on the embedded system.

Developers can connect the ST-Link programmer to their STM32 board using the SWD interface, and the programmer is typically connected to a computer via USB. The ST-Link Utility or other compatible tools can then be used to transfer firmware and perform debugging operations.



**You**

what is st link utility

ChatGPT

**ChatGPT**

The ST-Link Utility is a software tool provided by STMicroelectronics for programming and debugging STM32 microcontrollers. It is designed to work in conjunction with the ST-Link programmer, allowing developers to perform various tasks related to the STM32 microcontroller, including programming the flash memory, erasing memory sectors, and debugging applications in real-time.

Key features of the ST-Link Utility include:

1. **Programming Flash Memory:**
   * The utility allows users to program the flash memory of STM32 microcontrollers. This is essential for loading firmware and application code onto the microcontroller.
2. **Erasing Memory Sectors:**
   * Users can erase specific sectors or the entire flash memory of the STM32 device. This is useful when reprogramming the microcontroller with updated firmware.
3. **Memory Inspection and Modification:**
   * The ST-Link Utility provides tools for inspecting and modifying memory contents. Users can view and edit memory registers, RAM, and other memory regions.
4. **Debugging Operations:**
   * Real-time debugging features are available, allowing developers to set breakpoints, step through code, and inspect variables during the execution of an application.
5. **Option Bytes Configuration:**
   * The utility allows configuration of option bytes, which control various aspects of the microcontroller's behavior, such as read protection and boot configuration.
6. **Firmware Update for ST-Link Programmer:**
   * Users can update the firmware of the connected ST-Link programmer through the utility. This ensures compatibility with the latest STM32 microcontrollers and provides access to new features.
7. **Log and Console Output:**
   * The ST-Link Utility provides a log window that displays detailed information about the programming and debugging operations. This can be helpful for troubleshooting and diagnostics.
8. **Support for Different STM32 Series:**
   * The utility supports a wide range of STM32 microcontrollers across different series and models.

Developers often use the ST-Link Utility in their development workflow, integrating it with their chosen integrated development environment (IDE) or toolchain. It provides a graphical user interface (GUI) for performing essential tasks related to STM32 microcontroller development.

To use the ST-Link Utility, developers typically connect the ST-Link programmer to the STM32 board, launch the utility on their computer, and then perform the desired programming or debugging operations.