Introduction

In this project, STM32F4 Development Board is used to set up a blinky application, which indicates a running light LED. This is a simple application for beginner to familiarize with the usage of the board and is also the first step to get the board ready with correct setup for further developments.

The software used is STM32CubeIDE. IDE stands for Integrated Development Environment, which is used as an interface for user to program the microcontroller. This software is user-friendly as it includes most of the features and common tools needed for various application development.

Before executing the program, setting up the board is essential. It includes determining the project type, board selection and project firmware configuration. This step is needed so that the software knows which configuration and board is used by the user.

Comes to the coding part, there is a user-friendly C code template in the IDE, which can be used as the base code of this application. First, function "HAL_int" is called to reset all the peripherals and initialize the Flash interface and the Systick so that the previous program would not be taking effect. Next, the system clock is configured with "SystemClock_Config" function to set the operating frequency. Then, initialization on "MX_GPIO_Init" and "MX_USART2_Init" variables is carried out.

The concept of this blinky application is that the output pin is first set as high (logic '1') to turn on the LED. After some delay, the pin is set to be low (logic '0') to toggle the LED. In the case, "HAL_Delay" function is used to generate delay in milliseconds. A 500ms delay is used in this LED blinking application. Besides, "HAL_GPIO_TogglePin(port, pin)" function is used to toggle the LED. All the functions above are carried out in the "while (1)" loop to keep the LED toggling.

In order to check which output pin is used for LED on the board, the declaration of library "main.h" is opened while "LD2_GPIO_Port" and "LD2_Pin" are declared. The designated port and pin are defined during the usage of "HAL_GPIO_TogglePin(port, pin)" function.

In the end, a debug session is built and being executed after the program code is completed. The LED perform expected toggling after executing the program to the microcontroller board.