**User button for STM32F407G-DISC1**

**Polling:**

1. Open the ide and create a file name.c in the Src (your\_proj-> Core->Src->name.c) and use that code :

#include "name.h" // Include the header file for name (possibly defining the function prototypes or macros)

#include "main.h" // Include the main header (likely contains hardware configuration and peripheral setup)

void name() // Define the function name()

{

int clk = 0; // Initialize 'clk' as a counter to control which LED is turned on

while (1) { // Infinite loop that keeps running

// Check the value of clk and turn on the corresponding LED

if (clk == 0) {

HAL\_GPIO\_WritePin(GPIOD, LD3\_Pin, GPIO\_PIN\_SET); // Turn ON LED connected to LD3\_Pin

} else if (clk == 1) {

HAL\_GPIO\_WritePin(GPIOD, LD4\_Pin, GPIO\_PIN\_SET); // Turn ON LED connected to LD4\_Pin

} else if (clk == 2) {

HAL\_GPIO\_WritePin(GPIOD, LD6\_Pin, GPIO\_PIN\_SET); // Turn ON LED connected to LD6\_Pin

} else if (clk == 3) {

HAL\_GPIO\_WritePin(GPIOD, LD5\_Pin, GPIO\_PIN\_SET); // Turn ON LED connected to LD5\_Pin

}

// Read the state of the button connected to GPIOA pin PA0

if (HAL\_GPIO\_ReadPin(GPIOA, PA0)) {

clk++; // Increment the clk counter when the button is pressed

HAL\_Delay(200); // Add a delay of 200ms to avoid button bouncing and ensure proper response

// Reset all LEDs (turn off) after the button press

HAL\_GPIO\_WritePin(GPIOD, LD3\_Pin, GPIO\_PIN\_RESET); // Turn OFF LD3

HAL\_GPIO\_WritePin(GPIOD, LD4\_Pin, GPIO\_PIN\_RESET); // Turn OFF LD4

HAL\_GPIO\_WritePin(GPIOD, LD5\_Pin, GPIO\_PIN\_RESET); // Turn OFF LD5

HAL\_GPIO\_WritePin(GPIOD, LD6\_Pin, GPIO\_PIN\_RESET); // Turn OFF LD6

}

// If clk reaches 4 (i.e., after LD5 is ON), reset clk to 0 to loop back to LD3

if (clk == 4) {

clk = 0; // Reset clk to 0 to start the cycle over

}

}

HAL\_Delay(100); // Add a 100ms delay for general debouncing/stability (This line is unreachable but may be used elsewhere)

}

1. Create a file name.h in the Inc (your\_proj-> Core->Inc->name.h) and use that code :

#ifndef INC\_NAME\_H\_ // If not defined, define INC\_NAME\_H\_ to prevent multiple inclusions of this file

#define INC\_NAME\_H\_ // Define INC\_NAME\_H\_ to ensure the contents of this file are included only once

// Function prototype for name

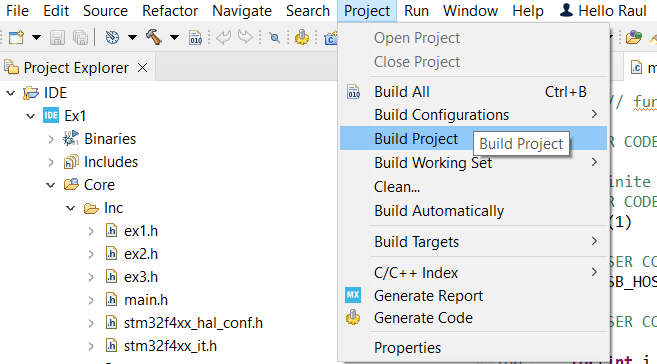
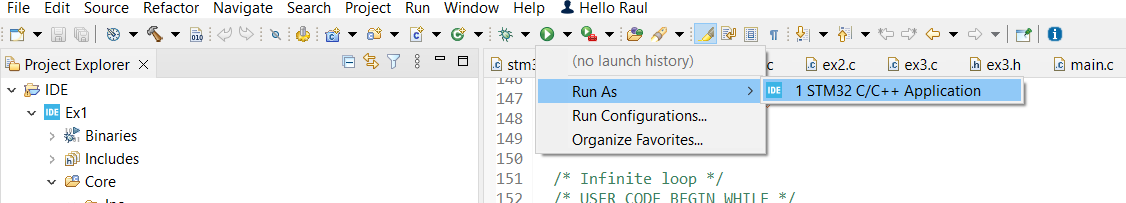
// This function is defined elsewhere, and this header provides its declaration

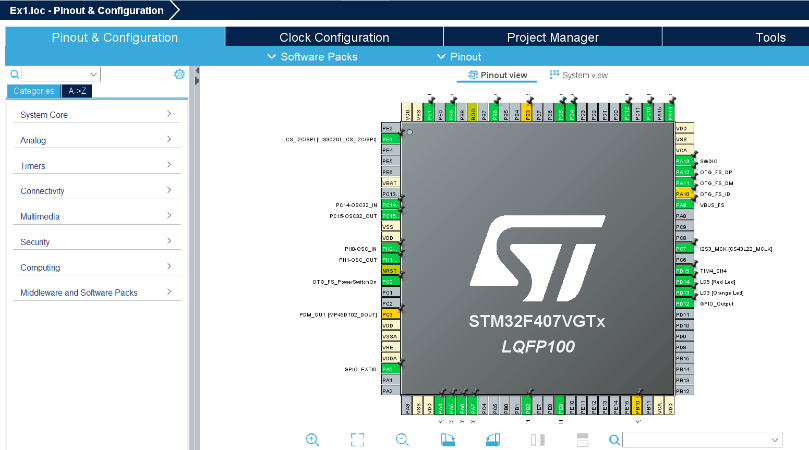
void name();

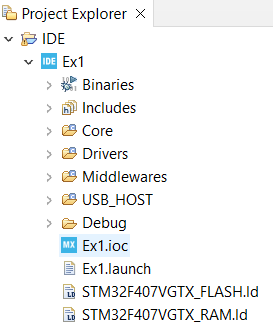
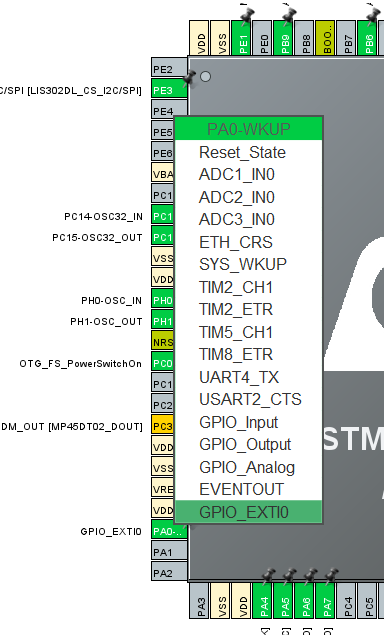
#endif /\* INC\_NAME\_H\_ \*/ // End of the include guard to prevent multiple inclusions

1. In main.c between /\* USER CODE BEGIN Includes \*//\* USER CODE END Includes \*/

you need to include your file . Also inside the while loop on int main() call your function.

1. Build your Project
2. Run as STM32 C/C++ Application

**Interrupt:**

1. Open the .ioc
2. Search for pin PA0 and configure it into GPIO\_EXTI0

1. Generate the new project and search for stm32f4x\_it.c
2. Inside stm32f4xx\_it.c search the : void EXTI0\_IRQHandler(void) function and paste that code :

void EXTI0\_IRQHandler(void) // This function handles the external interrupt for pin 0 (EXTI line 0)

{

/\* USER CODE BEGIN EXTI0\_IRQn 0 \*/

clk++; // Increment the clk counter on each interrupt (button press)

// Turn off all LEDs by resetting their pins (turn them OFF)

HAL\_GPIO\_WritePin(GPIOD, LD3\_Pin, GPIO\_PIN\_RESET); // Turn OFF LD3

HAL\_GPIO\_WritePin(GPIOD, LD4\_Pin, GPIO\_PIN\_RESET); // Turn OFF LD4

HAL\_GPIO\_WritePin(GPIOD, LD5\_Pin, GPIO\_PIN\_RESET); // Turn OFF LD5

HAL\_GPIO\_WritePin(GPIOD, LD6\_Pin, GPIO\_PIN\_RESET); // Turn OFF LD6

// Reset clk to 0 if it reaches 4 (after LD5 is ON, loop back to LD3)

if(clk == 4) {

clk = 0; // Reset clk to restart the LED sequence

}

// Depending on the value of clk, turn on the corresponding LED

if (clk == 0) {

HAL\_GPIO\_WritePin(GPIOD, LD3\_Pin, GPIO\_PIN\_SET); // Turn ON LD3 when clk is 0

} else if (clk == 1) {

HAL\_GPIO\_WritePin(GPIOD, LD4\_Pin, GPIO\_PIN\_SET); // Turn ON LD4 when clk is 1

} else if (clk == 2) {

HAL\_GPIO\_WritePin(GPIOD, LD6\_Pin, GPIO\_PIN\_SET); // Turn ON LD6 when clk is 2

} else if (clk == 3) {

HAL\_GPIO\_WritePin(GPIOD, LD5\_Pin, GPIO\_PIN\_SET); // Turn ON LD5 when clk is 3

}

/\* USER CODE END EXTI0\_IRQn 0 \*/

// Call HAL EXTI handler to process the interrupt and clear the flag

HAL\_GPIO\_EXTI\_IRQHandler(GPIO\_PIN\_0);

/\* USER CODE BEGIN EXTI0\_IRQn 1 \*/

// Custom code can be added here if needed after the interrupt is processed

/\* USER CODE END EXTI0\_IRQn 1 \*/

}

1. After you have done all that Build the project and run it and you have made a script that will turn on all the LED’s each at a time using the Polling method but also Interrupt method.