|  |
| --- |
| #include"main.h" |
|  | #include "stm32f1xx\_hal.h" //This header contains all the functions prototypes |
|  | //for the HAL module driver. |
|  |  |
|  | void SystemClock\_Config(void); //Initizing System Clock 8Mhz |
|  | void Error\_Handler(void); //Error Handler- We are not going to use it. |
|  | static void MX\_GPIO\_Init(void); //GPIO Initialization function |
|  |  |
|  | int main(void) |
|  | { |
|  |  |
|  | HAL\_Init(); //Initialize the Hal Libraries |
|  |  |
|  | /\* Configure the system clock \*/ |
|  | SystemClock\_Config(); |
|  |  |
|  | /\* Initialize all configured peripherals \*/ |
|  | MX\_GPIO\_Init(); //Calling GPIO initializing functions |
|  |  |
|  | /\* USER CODE BEGIN WHILE \*/ |
|  | while (1) |
|  | { |
|  | HAL\_GPIO\_TogglePin(Led\_GPIO\_Port,Led\_Pin); //Toggle Gpio |
|  | HAL\_Delay(1000); //Delay 1 second |
|  | } |
|  |  |
|  | } |
|  |  |
|  | /\*\* System Clock Configuration |
|  | \*/ |
|  | void SystemClock\_Config(void) |
|  | { |
|  |  |
|  | RCC\_OscInitTypeDef RCC\_OscInitStruct; |
|  | RCC\_ClkInitTypeDef RCC\_ClkInitStruct; |
|  |  |
|  | /\*\*Initializes the CPU, AHB and APB busses clocks |
|  | \*/ |
|  | RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_HSI; //Using internal 8Mhz oscillator |
|  | RCC\_OscInitStruct.HSIState = RCC\_HSI\_ON; |
|  | RCC\_OscInitStruct.HSICalibrationValue = 16; |
|  | RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_NONE; |
|  | if (HAL\_RCC\_OscConfig(&RCC\_OscInitStruct) != HAL\_OK) |
|  | { |
|  | Error\_Handler(); |
|  | } |
|  |  |
|  | /\*\*Initializes the CPU, AHB and APB busses clocks |
|  | \*/ |
|  | RCC\_ClkInitStruct.ClockType = RCC\_CLOCKTYPE\_HCLK|RCC\_CLOCKTYPE\_SYSCLK |
|  | |RCC\_CLOCKTYPE\_PCLK1|RCC\_CLOCKTYPE\_PCLK2; |
|  | RCC\_ClkInitStruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_HSI; |
|  | RCC\_ClkInitStruct.AHBCLKDivider = RCC\_SYSCLK\_DIV1; |
|  | RCC\_ClkInitStruct.APB1CLKDivider = RCC\_HCLK\_DIV1; |
|  | RCC\_ClkInitStruct.APB2CLKDivider = RCC\_HCLK\_DIV1; |
|  |  |
|  | if (HAL\_RCC\_ClockConfig(&RCC\_ClkInitStruct, FLASH\_LATENCY\_0) != HAL\_OK) |
|  | { |
|  | Error\_Handler(); |
|  | } |
|  |  |
|  | /\*\*Configure the Systick interrupt time |
|  | \*/ |
|  | HAL\_SYSTICK\_Config(HAL\_RCC\_GetHCLKFreq()/1000); |
|  |  |
|  | /\*\*Configure the Systick |
|  | \*/ |
|  | HAL\_SYSTICK\_CLKSourceConfig(SYSTICK\_CLKSOURCE\_HCLK); |
|  |  |
|  | /\* SysTick\_IRQn interrupt configuration \*/ |
|  | HAL\_NVIC\_SetPriority(SysTick\_IRQn, 0, 0); |
|  | } |
|  |  |
|  | static void MX\_GPIO\_Init(void) |
|  | { |
|  |  |
|  | GPIO\_InitTypeDef GPIO\_InitStruct; |
|  |  |
|  | /\* GPIO Ports Clock Enable \*/ |
|  | \_\_HAL\_RCC\_GPIOC\_CLK\_ENABLE(); //Enable GPIO Clock |
|  |  |
|  | /\*Configure GPIO pin Output Level \*/ |
|  | HAL\_GPIO\_WritePin(Led\_GPIO\_Port, Led\_Pin, GPIO\_PIN\_RESET); //Initialize the GPIO Pin Output Level |
|  |  |
|  | /\*Configure GPIO pin : Led\_Pin \*/ |
|  | GPIO\_InitStruct.Pin = Led\_Pin; //Pin name we give in StmCubeMx Configuration |
|  | GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP; //Pin Mode Push Pull |
|  | GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW; //GPIO speed frequency Low |
|  | HAL\_GPIO\_Init(Led\_GPIO\_Port, &GPIO\_InitStruct); //Pass the setting perimeters for initialization |
|  |  |
|  | } |