

```

include "main.h" void SystemClock_Config(void); static void MX_GPIO_Init(void); int main(void) {
HAL_Init(); SystemClock_Config(); MX_GPIO_Init(); while (1) { uint8_t pir_status =
HAL_GPIO_ReadPin(GPIOB, PIR_Pin); uint8_t touch_status = HAL_GPIO_ReadPin(GPIOB,
TOUCH_Pin); HAL_GPIO_WritePin(BLUE_GPIO_Port, BLUE_Pin, pir_status); if (touch_status
== GPIO_PIN_SET) { HAL_GPIO_WritePin(GPIOA, RED_Pin, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOA, GREEN_Pin, GPIO_PIN_SET); HAL_Delay(3000);
HAL_GPIO_WritePin(GPIOA, GREEN_Pin, GPIO_PIN_RESET); HAL_Delay(3000); } else {
if (pir_status == GPIO_PIN_RESET && touch_status == GPIO_PIN_RESET) {
HAL_GPIO_WritePin(GPIOA, RED_Pin, GPIO_PIN_SET); } else {
HAL_GPIO_WritePin(GPIOA, RED_Pin, GPIO_PIN_RESET); } } } }

void SystemClock_Config(void)
{
RCC_OscInitTypeDef RCC_OscInitStruct = {0};
RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
RCC_OscInitStruct.HSIState = RCC_HSI_ON;
RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_NONE;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
Error_Handler();
}
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();
}
}

```

```

static void MX_GPIO_Init(void)
{
GPIO_InitTypeDef GPIO_InitStructure = {0};
__HAL_RCC_GPIOD_CLK_ENABLE();
__HAL_RCC_GPIOA_CLK_ENABLE();
__HAL_RCC_GPIOB_CLK_ENABLE();
HAL_GPIO_WritePin(GPIOA, RED_Pin|GREEN_Pin, GPIO_PIN_RESET);
HAL_GPIO_WritePin(BLUE_GPIO_Port, BLUE_Pin, GPIO_PIN_RESET);
GPIO_InitStructure.Pin = RED_Pin|GREEN_Pin;
GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStructure.Pull = GPIO_NOPULL;
GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);
GPIO_InitStructure.Pin = BLUE_Pin;
GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStructure.Pull = GPIO_NOPULL;
GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(BLUE_GPIO_Port, &GPIO_InitStructure);
GPIO_InitStructure.Pin = PIR_Pin|TOUCH_Pin;
GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
GPIO_InitStructure.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOB, &GPIO_InitStructure);
}

void Error_Handler(void)
{
__disable_irq();
while (1)
{
}
}

```