

```

#include "main.h"

/* Private variables -----*/
ADC_HandleTypeDef hadc1;

/* USER CODE BEGIN PV */
uint32_t adcValue = 0;
uint32_t filteredValue = 0;
uint8_t beatDetected = 0;

#define FILTER_SIZE 10
uint16_t buffer[FILTER_SIZE];
uint8_t indexBuf = 0;
/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_ADC1_Init(void);

/* USER CODE BEGIN 0 */
uint16_t moving_average(uint16_t val)
{
    buffer[indexBuf++] = val;
    if(indexBuf >= FILTER_SIZE) indexBuf = 0;

    uint32_t sum = 0;
    for(int i=0;i<FILTER_SIZE;i++) sum += buffer[i];

```

```

        return sum / FILTER_SIZE;
    }

/* USER CODE END 0 */

int main(void)
{
    HAL_Init();
    SystemClock_Config();

    MX_GPIO_Init();
    MX_ADC1_Init();

/* USER CODE BEGIN 2 */
    HAL_ADC_Start(&hadc1);
/* USER CODE END 2 */

/* Infinite loop */
    while (1)
    {
        // ==== BACA ADC ====
        HAL_ADC_Start(&hadc1);
        HAL_ADC_PollForConversion(&hadc1, 10);
        adcValue = HAL_ADC_GetValue(&hadc1);

        // ==== FILTER ====
        filteredValue = moving_average(adcValue);

        // ==== THRESHOLD ====
        uint32_t threshold = 2000;

```

```

// ===== DETEKSI DETAK =====

if(filteredValue > threshold && beatDetected == 0)
{
    beatDetected = 1;

    // LED HIJAU ON
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_SET);

    HAL_Delay(100);

    // LED OFF
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
}

if(filteredValue < threshold)
{
    beatDetected = 0;
}

HAL_Delay(5);
}

/* ===== CLOCK ===== */
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;
HAL_RCC_OscConfig(&RCC_OscInitStruct);

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;

HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_0);
}

/* ===== ADC ===== */
static void MX_ADC1_Init(void)
{
    ADC_ChannelConfTypeDef sConfig = {0};

    __HAL_RCC_ADC1_CLK_ENABLE();

    hadc1.Instance = ADC1;
    hadc1.Init.ScanConvMode = ADC_SCAN_DISABLE;
    hadc1.Init.ContinuousConvMode = DISABLE;
    hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
    hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;

```

```
hadc1.Init.NbrOfConversion = 1;
```

```
HAL_ADC_Init(&hadc1);
```

```
sConfig.Channel = ADC_CHANNEL_0;
```

```
sConfig.Rank = ADC_REGULAR_RANK_1;
```

```
sConfig.SamplingTime = ADC_SAMPLETIME_71CYCLES_5;
```

```
HAL_ADC_ConfigChannel(&hadc1, &sConfig);
```

```
}
```

```
/* ===== GPIO ===== */
```

```
static void MX_GPIO_Init(void)
```

```
{
```

```
__HAL_RCC_GPIOA_CLK_ENABLE();
```

```
__HAL_RCC_GPIOB_CLK_ENABLE();
```

```
GPIO_InitTypeDef GPIO_InitStructure = {0};
```

```
// PA0 = ADC
```

```
GPIO_InitStructure.Pin = GPIO_PIN_0;
```

```
GPIO_InitStructure.Mode = GPIO_MODE_ANALOG;
```

```
HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);
```

```
// LED PB0
```

```
GPIO_InitStructure.Pin = GPIO_PIN_0;
```

```
GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
```

```
GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
```

```
HAL_GPIO_Init(GPIOB, &GPIO_InitStructure);
```

```
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
```

}