

```
<a name="home">
```

```
</a>
```

```
<br />
```

```
<div style="text-align: center;">
```

```
<a
```

```
href="https://ardizal222007.blogspot.com/p/modul-ii-praktikum-mikroprosesor.html">[KEMBA  
LI KE MENU SEBELUMNYA]</a></div>
```

```
<br />
```

```
<center>
```

```
<div style="background-color: white; border: 2px dashed rgb(23, 128, 221); height: 240px;  
overflow: auto; padding: 10px; text-align: center; width: 330px;">
```

```
<b>DAFTAR ISI</b>
```

```
<br />
```

```
<div style="text-align: left;">
```

```
<a href="#prosedur">1. Prosedur </a></div>
```

```
<div style="text-align: left;">
```

```
<a href="#hard">2. Hardware dan Diagram Blok </a></div>
```

```
<div style="text-align: left;">
```

```
<a href="#rangkaian">3. Rangkaian Simulasi dan Prinsip Kerja <br /></a></div>
```

```
<div style="text-align: left;">
```

```
<a href="#flow">4. Flowchart dan Listing Program</a></div>
```

```
<div style="text-align: left;">
```

```
<a href="#video">5. Video Demo </a><br />
```

```
<div style="text-align: left;">
```

```
<a href="#analisa">6. Analisa </a></div>
```

```
<div style="text-align: left;">
```

```
<a href="#download">7. Download File </a>&nbsp;</div>
```


sPfhyphehypheni8LJX9_jzbtPCbKL7T5kblbpsqJcGtrAtTwaPdphjzpsLAsWk4GYfuzCXEAALGTfWrB
8yw4e4B9AOaKbY7eOM/s1052/STM32-Pin-Details_0.png" style="margin-left: 1em;
margin-right: 1em;"></div><div>
</div><ul style="text-align: left;"><span
style="font-family: times;">Push button<div class="separator" style="clear:
both; text-align: center;"><a
href="https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEgvFo8oY_8Xj8k98ee_q5Z
vXIQKBkQ9rtUvgSrwnRwaiPSHg64kfnocQnpon01AO8m0RdxAzSfmKrnOoSiFgkAbYyow2S7kXpVi
7BhymC18djCoxuWtBNKc5q0oXW8LMZm8ZSy9P87Qpwj4IB-T9fs3Y8Vsn5kUasXOjiBcBHTECazO
PhE6NvM1TiU5Eug/s558/Screenshot%202025-03-17%20075743.png" style="margin-left: 1em;
margin-right: 1em;"></div>
<div>
</div><ul style="text-align: left;"><span style="font-family:
times;">LED RGB<div><div class="separator" style="clear: both; text-align:
center;"><a
href="https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEihLHAebZic0CLqrpEXdWo
g3BPZStq2HNIGHSS4Qrpn8k2Yj98egVRihtsnNLdUnzDD9YzJAsyCcVSSGEHrTuZIsjwCVIN3PBzHE4K
AdjwVK7B-FkhqEx8Z8VE1dYWzDAneiBLL7jf_aTYyWu1yAYC9755wleXOkdc3Geldq5XqG5_MifW8
0inYlanJQ5M/s1920/types-of-rgb-min.jpg" style="margin-left: 1em; margin-right: 1em;"></div>
</div><div
class="separator" style="clear: both; text-align: center;"><li style="text-align:
justify;">Resistor<div class="separator"
style="clear: both; text-align: center;"><a
href="https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEhmKdMUtXfKifH8FnTVk1
wP5GfI9N67wRfreZLKqv1_ix8pNcBQTicqxT09Eoi3X-UdpKP6cFwlmweQUmJjETeM-rdEtMXuRmhl

hUv6tHzj11bEE6OKqvw0fgd4HlzS8FAFmKZOgaYQDeBVswiyLnBaa0AsztAf4TpPdIBJ6xZtDqTgMXt
yMtVdnFvMULY/s328/th%20(2).jpg" style="margin-left: 1em; margin-right: 1em;"></div><div class="separator"
style="clear: both; text-align: center;">
</div><div style="text-align:
justify;">Sensor Soil Moisture<div class="separator" style="clear: both;
text-align: center;"><a
href="https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEiOLqTOL6NAWfPYyt3Vq
MFJQvZh4_SSX9X_wiih0ug6qcEqSIV6EigX-Nsk6_SUxIBxcCYqQNi7R8hFuUf9FTefZfmCzG1hLHjTM
crdlaG6C4kKqh3q86D8Sf_ZdOClhaWzK6z9nCR5YD22ClvcWGHufphkGPDGnmWGP6Olinh3PrXqf
rRePnqWqcU7BqQ/s407/Screenshot%202025-05-02%20140134.png" style="margin-left: 1em;
margin-right: 1em;"></div>
<div style="text-align: justify;">
</div><div style="text-align:
justify;">
</div><div>Buzzer<div>
</div></div><div
class="separator" style="clear: both; text-align: center;"><a
href="https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEi3e5pXXu2E0VxfcKkwOrlt
hOrdmqoqQRw9cMF4DGshw16NmUJ0BjEyiiLnXYO81Ps78g0MgkNazoEyQe3CIsoqula9hdXTJbW
ujDSKNio5pHOQ2S0V0Z7PaN19EoHfoDT5VJkK3AKZzvUKesrP9GBpYZXQcXImO9bCqBLQsTlchCGc
u_aZwnYDJzm64BU/s2357/image.jpg" style="margin-left: 1em; margin-right: 1em;"></div>
<div><span
style="font-family: times;">Jumper Cable Wire<div class="separator"
style="clear: both; text-align: center;"><a
href="https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEiAW4SotbNGcseBQV2fv5
K5FZ7azhq5_rGwAPHAKbDWYZ3b18yUC9QIKraQSIQ6h0YDbFuzlf_3Rejb69ekDyOKTAFQwkgqrU
YH_4P5QVkrGgZ6934OqA-2SkUXCnegrZymC67tuRnR02VLHfBnT1CZ3y-kholJxQ7dM2-pDhcxTC
fBtc5m9RI15Z3ebc/s392/th.jpg" style="margin-left: 1em; margin-right: 1em;"></div>Beard board</div><div class="separator" style="clear: both; text-align: center;"></div><div>
</div></div><div style="text-align: justify;">
</div><div style="text-align: justify;">
</div><div style="text-align: justify;">Diagram Blok
</div><div style="text-align: justify;"><div class="separator" style="clear: both; text-align: center;"></div>
</div><div style="text-align: justify;">
</div></div><div class="separator" style="clear: both; text-align: center;">
</div><p><p>3. Rangkaian Simulasi[Kembali]</p><p>Rangkailah seperti rangkaian percobaan 7 pada modul 2</p><p></p>
<p></p><p></p><p></p><p></p><div class="separator" style="clear: both; text-align: center;"><a

href="https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEiLGBRgRdr5JrCRXpzgluSL

[illegible]

[illegible]

```

/*
#define ADC_THRESHOLD_MID 1500
Frekuensi Buzzer - using uint32_t instead of uint16_t */
const uint32_t
pwm_periods[] = {1000, 50000, 719999}; // 72MHz/freq - 1
void
SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void
MX_ADC1_Init(void);
static void MX_TIM2_Init(void);
void
update_leds_and_buzzer(uint32_t adc_val, uint8_t btn_state); // Updated function
signature
void change_sound_pattern(void);
void Error_Handler(void); //
Explicit declaration
int main(void) {
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_ADC1_Init();
    MX_TIM2_Init();
    HAL_TIM_PWM_Start(&htim2,
TIM_CHANNEL_3);
    __HAL_TIM_SET_COMPARE(&htim2,
TIM_CHANNEL_3, 0);
    HAL_ADC_Start(&hadc1);
    while (1)
    {
        static uint32_t last_adc_tick = 0;
        static uint32_t last_sound_change = 0;
        uint8_t button_state =
HAL_GPIO_ReadPin(BUTTON_PORT, BUTTON_PIN);
        if (HAL_GetTick() - last_adc_tick > 200) {
            last_adc_tick =
HAL_GetTick();
            HAL_ADC_Start(&hadc1);
            if
(HAL_ADC_PollForConversion(&hadc1, 10) == HAL_OK) {
                update_leds_and_buzzer(HAL_ADC_GetValue(&hadc1),
button_state);
            }
            if (button_state == GPIO_PIN_RESET &&
(HAL_ADC_GetValue(&hadc1) < ADC_THRESHOLD_MID)) {
                if (HAL_GetTick() - last_sound_change > 1000) {
                    last_sound_change = HAL_GetTick();
                    change_sound_pattern();
                }
            }
            HAL_Delay(10);
        }
        void update_leds_and_buzzer(uint32_t adc_val,
uint8_t btn_state) {
            HAL_GPIO_WritePin(LED_PORT,
LED_RED_PIN|LED_GREEN_PIN|LED_BLUE_PIN, GPIO_PIN_RESET);
            if (adc_val >= ADC_THRESHOLD_HIGH) {
                HAL_GPIO_WritePin(LED_PORT, LED_GREEN_PIN, GPIO_PIN_SET);
                __HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_3, 0);
            }
            else if (adc_val >= ADC_THRESHOLD_MID) {
                HAL_GPIO_WritePin(LED_PORT, LED_BLUE_PIN, GPIO_PIN_SET);
                HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_3, 0);
            }
        }
    }
}

```



```

</div><div>&nbsp;};</div><div>&nbsp;HAL_GPIO_WritePin(LED_PORT,  
LED_RED_PIN, GPIO_PIN_SET);</div><div>&nbsp;&nbsp;&nbsp;if (btn_state == GPIO_PIN_RESET)  
</div><div>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&__HAL_TIM_SET_AUTORELOAD(&htim2,  
pwm_periods[sound_pattern]);</div><div>&nbsp;&nbsp;&nbsp;&nbsp;&__HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_3,  
pwm_periods[sound_pattern]/2);</div><div>&nbsp;&nbsp;&nbsp;} else {</div><div>&nbsp;&nbsp;&nbsp;&nbsp;&__HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_3, 0);</div><div>&nbsp;&nbsp;&nbsp;&nbsp;&}</div><div>&nbsp;&nbsp;&nbsp;&__HAL_TIM_SET_AUTORELOAD(&htim2,  
pwm_periods[sound_pattern]);</div><div>&nbsp;&nbsp;&nbsp;&__HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_3,  
pwm_periods[sound_pattern]/2);</div><div>&nbsp;&nbsp;&nbsp;}</div><div><br /></div><div>void  
change_sound_pattern(void) {</div><div>&nbsp;&nbsp;&sound_pattern = (sound_pattern + 1) %  
3;</div><div>&nbsp;&nbsp;&if (HAL_ADC_GetValue(&hadc1) < ADC_THRESH_MID  
&&&)</div><div>&nbsp;&nbsp;&&nbsp;&HAL_GPIO_ReadPin(BUTTON_PORT,  
BUTTON_PIN) == GPIO_PIN_SET) {</div><div>&nbsp;&nbsp;&&__HAL_TIM_SET_AUTORELOAD(&htim2,  
pwm_periods[sound_pattern]);</div><div>&nbsp;&nbsp;&&__HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_3,  
pwm_periods[sound_pattern]/2);</div><div>&nbsp;&nbsp;&}</div><div><br /></div><div>void SystemClock_Config(void)</div><div>{</div><div>&nbsp;&RCC_OscInitTypeDef RCC_OscInitStruct = {0};</div><div>&nbsp;&RCC_ClkInitTypeDef  
RCC_ClkInitStruct = {0};</div><div>&nbsp;&RCC_PeriphCLKInitTypeDef PeriphClkInit =  
{0};</div><div><br /></div><div>&nbsp;&/* Initializes the RCC Oscillators according to the  
specified parameters</div><div>&nbsp;&* in the RCC_OscInitTypeDef  
structure.</div><div>&nbsp;&*/</div><div>&nbsp;&RCC_OscInitStruct.OscillatorType =  
RCC_OSCILLATORTYPE_HSE;</div><div>&nbsp;&RCC_OscInitStruct.HSEState =  
RCC_HSE_ON;</div><div>&nbsp;&RCC_OscInitStruct.HSEPredivValue =  
RCC_HSE_PREDIV_DIV1;</div><div>&nbsp;&RCC_OscInitStruct.HSIState =  
RCC_HSI_ON;</div><div>&nbsp;&RCC_OscInitStruct.PLL.PLLState =  
RCC_PLL_ON;</div><div>&nbsp;&RCC_OscInitStruct.PLL.PLLSource =  
RCC_PLLSOURCE_HSE;</div><div>&nbsp;&RCC_OscInitStruct.PLL.PLLMUL =  
RCC_PLL_MUL9;</div><div>&nbsp;&if (HAL_RCC_OscConfig(&RCC_OscInitStruct) !=  
HAL_OK)</div><div>&nbsp;&{</div><div>&nbsp;&&nbsp;&Error_Handler();</div><div>&nbsp;&&}</div><div><br /></div><div>&nbsp;&/* Initializes the CPU, AHB and APB buses  
clocks</div><div>&nbsp;&*/</div><div>&nbsp;&RCC_ClkInitStruct.ClockType =  
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK</div><div>&nbsp;&&nbsp;&&nbsp;&&~(RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;</div><div>&nbsp;&&RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;</div><div>&nbsp;&RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;</div><div>&nbsp;&RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;</div><div>&nbsp;&&}</div>

```

```

RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;</div><div><br /></div><div>&nbsp;</div> if
(HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) !=
HAL_OK)</div><div>&nbsp;</div><div>&nbsp;</div>&nbsp;</div><div>&nbsp;</div> Error_Handler();</div><div>&nbsp;</div>
}</div><div>&nbsp;</div> PeriphClkInit.PeriphClockSelection =
RCC_PERIPHCLK_ADC;</div><div>&nbsp;</div><div>&nbsp;</div> PeriphClkInit.AdcClockSelection =
RCC_ADCCLK2_DIV6;</div><div>&nbsp;</div><div>&nbsp;</div> if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit)
!= HAL_OK)</div><div>&nbsp;</div><div>&nbsp;</div><div>&nbsp;</div>&nbsp;</div><div>&nbsp;</div> Error_Handler();</div><div>&nbsp;</div>
}</div><div></div><div><br /></div><div>/**</div><div>&nbsp;</div> * @brief ADC1 Initialization
Function</div><div>&nbsp;</div> * @param None</div><div>&nbsp;</div> * @retval
None</div><div>&nbsp;</div> */</div><div>static void
MX_ADC1_Init(void)</div><div>{</div><div><br /></div><div>&nbsp;</div> /* USER CODE BEGIN
ADC1_Init 0 */</div><div><br /></div><div>&nbsp;</div> /* USER CODE END ADC1_Init 0
*/</div><div><br /></div><div>&nbsp;</div> ADC_ChannelConfTypeDef sConfig = {0};</div><div><br
/></div><div>&nbsp;</div> /* USER CODE BEGIN ADC1_Init 1 */</div><div><br /></div><div>&nbsp;</div>
/* USER CODE END ADC1_Init 1 */</div><div><br /></div><div>&nbsp;</div> /** Common
config</div><div>&nbsp;</div> */</div><div>&nbsp;</div> hadc1.Instance = ADC1;</div><div>&nbsp;</div>
hadc1.Init.ScanConvMode = ADC_SCAN_DISABLE;</div><div>&nbsp;</div>
hadc1.Init.ContinuousConvMode = DISABLE;</div><div>&nbsp;</div>
hadc1.Init.DiscontinuousConvMode = DISABLE;</div><div>&nbsp;</div><div>&nbsp;</div> hadc1.Init.ExternalTrigConv =
ADC_SOFTWARE_START;</div><div>&nbsp;</div><div>&nbsp;</div> hadc1.Init.DataAlign =
ADC_DATAALIGN_RIGHT;</div><div>&nbsp;</div><div>&nbsp;</div> hadc1.Init.NbrOfConversion = 1;</div><div>&nbsp;</div>
if (HAL_ADC_Init(&hadc1) != HAL_OK)</div><div>&nbsp;</div><div></div><div>&nbsp;</div>&nbsp;</div><div>&nbsp;</div>
Error_Handler();</div><div>&nbsp;</div><div></div><div><br /></div><div>&nbsp;</div> /** Configure Regular
Channel</div><div>&nbsp;</div> */</div><div>&nbsp;</div><div>&nbsp;</div> sConfig.Channel =
ADC_CHANNEL_0;</div><div>&nbsp;</div><div>&nbsp;</div> sConfig.Rank =
ADC_REGULAR_RANK_1;</div><div>&nbsp;</div><div>&nbsp;</div> sConfig.SamplingTime =
ADC_SAMPLETIME_1CYCLE_5;</div><div>&nbsp;</div><div>&nbsp;</div> if (HAL_ADC_ConfigChannel(&hadc1,
&sConfig) != HAL_OK)</div><div>&nbsp;</div><div></div><div>&nbsp;</div>&nbsp;</div><div>&nbsp;</div>
Error_Handler();</div><div>&nbsp;</div><div></div><div>&nbsp;</div><div>&nbsp;</div> /* USER CODE BEGIN ADC1_Init 2
*/</div><div><br /></div><div>&nbsp;</div> /* USER CODE END ADC1_Init 2 */</div><div><br
/></div><div></div><div><br /></div><div>/**</div><div>&nbsp;</div> * @brief TIM2 Initialization
Function</div><div>&nbsp;</div> * @param None</div><div>&nbsp;</div> * @retval
None</div><div>&nbsp;</div> */</div><div>static void
MX_TIM2_Init(void)</div><div>{</div><div><br /></div><div>&nbsp;</div> /* USER CODE BEGIN
TIM2_Init 0 */</div><div><br /></div><div>&nbsp;</div> /* USER CODE END TIM2_Init 0
*/</div><div><br /></div><div>&nbsp;</div> TIM_MasterConfigTypeDef sMasterConfig =
{0};</div><div>&nbsp;</div><div>&nbsp;</div> TIM_OC_InitTypeDef sConfigOC = {0};</div><div><br
/></div><div>&nbsp;</div> /* USER CODE BEGIN TIM2_Init 1 */</div><div><br /></div><div>&nbsp;</div>

```

```

/* USER CODE END TIM2_Init 1 */</div><div>&nbsp; htim2.Instance = TIM2;</div><div>&nbsp;
htim2.Init.Prescaler = 0;</div><div>&nbsp; htim2.Init.CounterMode =
TIM_COUNTERMODE_UP;</div><div>&nbsp; htim2.Init.Period = 65535;</div><div>&nbsp;
htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;</div><div>&nbsp;
htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;</div><div>&nbsp; if
(HAL_TIM_PWM_Init(&htim2) != HAL_OK)</div><div>&nbsp; {</div><div>&nbsp; &nbsp; &nbsp;
Error_Handler();</div><div>&nbsp; }</div><div>&nbsp; sMasterConfig.MasterOutputTrigger =
TIM_TRGO_RESET;</div><div>&nbsp; sMasterConfig.MasterSlaveMode =
TIM_MASTERSLAVEMODE_DISABLE;</div><div>&nbsp; if
(HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) !=
HAL_OK)</div><div>&nbsp; {</div><div>&nbsp; &nbsp; &nbsp; Error_Handler();</div><div>&nbsp;
}</div><div>&nbsp; sConfigOC.OCMode = TIM_OCMODE_PWM1;</div><div>&nbsp;
sConfigOC.Pulse = 0;</div><div>&nbsp; sConfigOC.OCpolarity =
TIM_OCPOLARITY_HIGH;</div><div>&nbsp; sConfigOC.OCFastMode =
TIM_OCFAST_DISABLE;</div><div>&nbsp; if (HAL_TIM_PWM_ConfigChannel(&htim2,
&sConfigOC, TIM_CHANNEL_3) != HAL_OK)</div><div>&nbsp; {</div><div>&nbsp; &nbsp; &nbsp;
Error_Handler();</div><div>&nbsp; }</div><div>&nbsp; /* USER CODE BEGIN TIM2_Init 2
*/</div><div><br /></div><div>&nbsp; /* USER CODE END TIM2_Init 2 */</div><div>&nbsp;
HAL_TIM_MspPostInit(&htim2);</div><div><br /></div><div>&nbsp; }</div><div><br
/></div><div>/**</div><div>&nbsp; * @brief GPIO Initialization Function</div><div>&nbsp; *
@param None</div><div>&nbsp; * @retval None</div><div>&nbsp; */</div><div>/* GPIO
Initialization */</div><div>static void MX_GPIO_Init(void) {</div><div>&nbsp;
GPIO_InitTypeDef GPIO_InitStruct = {0};</div><div><br /></div><div>&nbsp;
__HAL_RCC_GPIOA_CLK_ENABLE();</div><div>&nbsp;
__HAL_RCC_GPIOB_CLK_ENABLE();</div><div><br /></div><div>&nbsp; /* LED Outputs
*/</div><div>&nbsp; GPIO_InitStruct.Pin =
LED_RED_PIN|LED_GREEN_PIN|LED_BLUE_PIN;</div><div>&nbsp; GPIO_InitStruct.Mode =
GPIO_MODE_OUTPUT_PP;</div><div>&nbsp; GPIO_InitStruct.Pull =
GPIO_NOPULL;</div><div>&nbsp; GPIO_InitStruct.Speed =
GPIO_SPEED_FREQ_LOW;</div><div>&nbsp; HAL_GPIO_Init(LED_PORT,
&GPIO_InitStruct);</div><div><br /></div><div>&nbsp; /* Button Input
*/</div><div>&nbsp; GPIO_InitStruct.Pin = BUTTON_PIN;</div><div>&nbsp;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;</div><div>&nbsp; GPIO_InitStruct.Pull =
GPIO_PULLUP;</div><div>&nbsp; HAL_GPIO_Init(BUTTON_PORT,
&GPIO_InitStruct);</div><div>&nbsp; }</div><div><br /></div><div>/* USER CODE BEGIN 4
*/</div><div><br /></div><div>/* USER CODE END 4 */</div><div><br
/></div><div>/**</div><div>&nbsp; * @brief&nbsp; This function is executed in case of error
occurrence.</div><div>&nbsp; * @retval None</div><div>&nbsp; */</div><div>void

```

```

Error_Handler(void)</div><div>{</div>&nbsp; /* USER CODE BEGIN Error_Handler_Debug
*/</div><div>&nbsp; /* User can add his own implementation to report the HAL error return
state */</div><div>&nbsp; __disable_irq();</div><div>&nbsp; while (1)</div><div>&nbsp;
{</div><div>&nbsp; }</div><div>&nbsp; /* USER CODE END Error_Handler_Debug
*/</div><div>}</div><div><br /></div><div>#ifdef&nbsp;
USE_FULL_ASSERT</div><div>/**</div><div>&nbsp; * @brief&nbsp; Reports the name of the
source file and the source line number</div><div>&nbsp; *&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
&nbsp;where the assert_param error has occurred.</div><div>&nbsp; * @param&nbsp; file:
pointer to the source file name</div><div>&nbsp; * @param&nbsp; line: assert_param error
line source number</div><div>&nbsp; * @retval None</div><div>&nbsp; */</div><div>void
assert_failed(uint8_t *file, uint32_t line)</div><div>{</div><div>&nbsp; /* USER CODE BEGIN 6
*/</div><div>&nbsp; /* User can add his own implementation to report the file name and line
number,</div><div>&nbsp; &nbsp; &nbsp; ex: printf("Wrong parameters value: file %s on line
%d\r\n", file, line) */</div><div>&nbsp; /* USER CODE END 6 */</div><div>}</div><div>#endif
/* USE_FULL_ASSERT */</div></div></div><div style="text-align: justify;"><br /></div><div
style="font-family: Consolas, "Courier New", monospace; font-size: 14px; text-align:
justify; white-space: pre;"><span style="color: green;"><br
/></span></div></div></div><p></p><p><b style="text-align: left;"><span style="font-family:
Josefin Sans; font-size: medium;"><a href="https://www.blogger.com/null" name="video">5.
Video Demo</a><a href="#home">[Kembali]</a></span></b></p><div><div class="separator"
style="clear: both; text-align: center;"><div class="separator" style="clear: both; text-align:
center;"><iframe allowfullscreen="" class="BLOG_video_class" height="266"
src="https://www.youtube.com/embed/f1ChswqaTks" width="320"
youtube-src-id="f1ChswqaTks"></iframe></div><div class="separator" style="clear: both;
text-align: center;"><br /></div></div></div><div><p></p><p style="text-align: left;"></p><p
style="text-align: justify;"><b style="text-align: left;"><b><span style="font-family: Josefin Sans;
font-size: medium;"><a href="https://www.blogger.com/null" name="analisa">6. Analisa</a><a
href="#home">[Kembali]</a></span></b></b></p><p class="MsoNormal" style="text-align:
justify;"><span>&nbsp; Tidak menggunakan amnalisa dan juga Kondisi tidak sesuai karena stlink
rusak</span></p><p style="text-align: justify;"><b style="text-align: left;"><span
style="font-family: Josefin Sans; font-size: medium;"><a href="https://www.blogger.com/null"
name="download">7. Download File</a><a href="#home">[Kembali]</a></span></b></p><p
style="text-align: justify;"><b style="text-align: left;"></b></p><center style="background-color:
white; font-family: Arial, Tahoma, Helvetica, FreeSans, sans-serif; font-size: 14.85px;"><div
style="text-align: left;"><b style="text-align: justify;"><span style="font-family: times;"><span
style="font-weight: 400;">HTML <a
href="#">[Download]</a></span></span></b></div></center><center
style="background-color: white; font-family: Arial, Tahoma, Helvetica, FreeSans, sans-serif;

```

font-size: 14.85px;"><div style="text-align: left;">Listing Program [Download]</div></center><center style="background-color: white; font-family: Arial, Tahoma, Helvetica, FreeSans, sans-serif; font-size: 14.85px;"><div style="text-align: left;">Video Demo [Download] </div><div style="text-align: left;"><div>Datasheet Mikrokontroler STM32F103C8 [Download]</div><div>Datasheet Sensor Soil Moisture [Download]</div></div><div>Datasheet Push Button [Download]</div><div>Datasheet RGB LED [Download]</div><div>Datasheet Buzzer [Download]</div></div></center><center style="background-color: white; font-size: 14.85px;"></center><p style="text-align: justify;"><b style="text-align: left;"><br

[illegible]