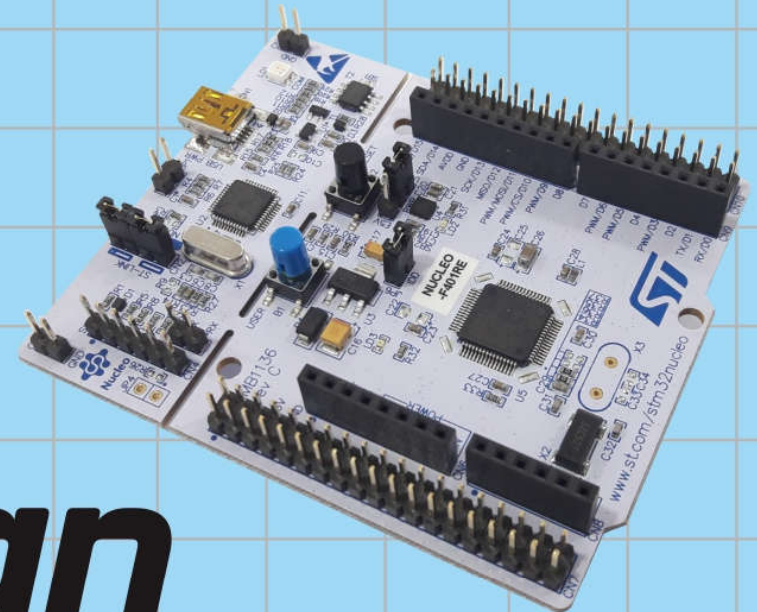


DEBUGGING

Menggunakan
STM32CubeIDE

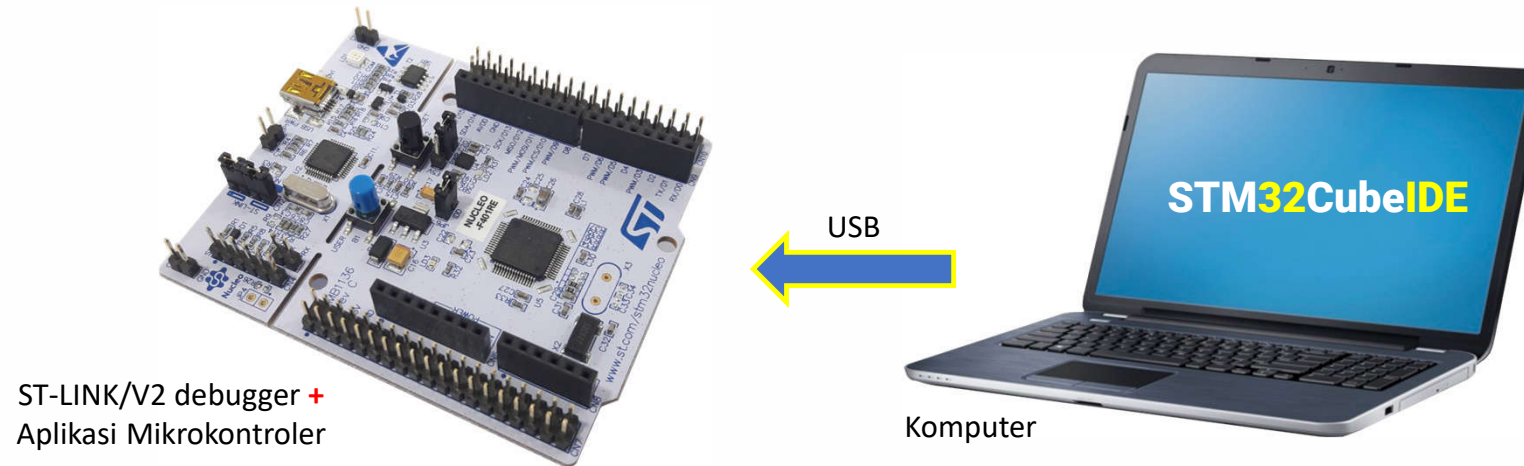


ROBONESIA.com
more than robotics learning

Apa itu **Debugging** ?

- *Debugging* adalah proses yang bertujuan untuk **mengidentifikasi** dan **menghapus bug** atau **kesalahan/error** yang terdapat di dalam suatu kode program.
- *Bug* pada program menyebabkan sebuah program aplikasi *embedded systems* tidak dapat *di-compile/build*.
- Proses *debugging* proyek aplikasi *embedded systems* mikrokontroler **STM32** yang dapat dilakukan menggunakan **STM32CubeIDE** adalah:
 1. Melacak dan memperbaiki *error/bugs*/kesalahan yang terdapat pada program.
 2. Memonitor penggunaan area memori (RAM & FLASH).
 3. Memonitor nilai variabel-variabel yang dibuat.
 4. Memonitor *General Purpose Register* (GPR) yang digunakan.
 5. Memonitor *Special Function Register* (SFR) yang digunakan.
 6. *Live Expression* → Monitoring kondisi GPIO (Data numerik).
 7. *Serial Wire Viewer (SWV) Data Trace* → Monitoring kondisi GPIO (*Timeline Graphic*).

Kebutuhan Hardware untuk Proses Debugging



Debugging Pada

Perspective - User Development



ROBONESIA.com
more than robotics learning

Debugging - Perspective User Development

1. Melacak dan memperbaiki *error/bugs*/kesalahan pada program.
2. Memonitor penggunaan area memori (RAM & FLASH)

```
MX led_blinkioc  *main.c x
87
88 /* Initialize all configured peripherals */
89 MX_GPIO_Init();
90 MX_USART2_UART_Init();
91 /* USER CODE BEGIN 2 */
92
93 /* USER CODE END 2 */
94
95 /* Infinite loop */
96 /* USER CODE BEGIN WHILE */
97 while (1)
98 {
99     HAL_GPIO_TogglePinn(LD2_GPIO_Port,LD2_Pin);
100     HAL_Delay(500);
101     monitor_kondisi_LED = HAL_GPIO_ReadPin(LD2_GPIO_Port,LD2_Pin);
102
103     /* USER CODE END WHILE */
104
```

Error pada program

led_blink.elf - /led_blink/Debug - Oct 6, 2022, 6:26:52 AM						
Memory Regions						
Region	Start address	End address	Size	Free	Used	Usage (%)
RAM	0x20000000	0x20018000	96 KB	94.38 KB	1.62 KB	1.68%
FLASH	0x08000000	0x08080000	512 KB	504.3 KB	7.7 KB	1.50%

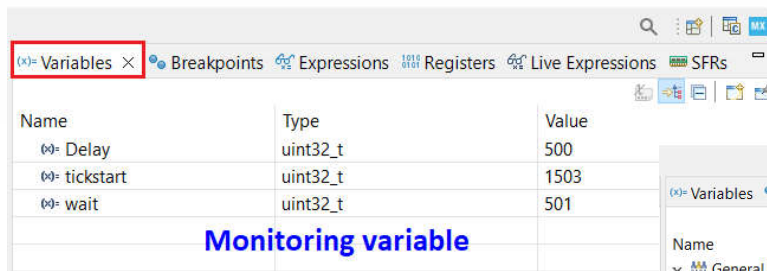
Memory use (RAM & Flash) monitoring

Debugging Pada Perspective - Debug



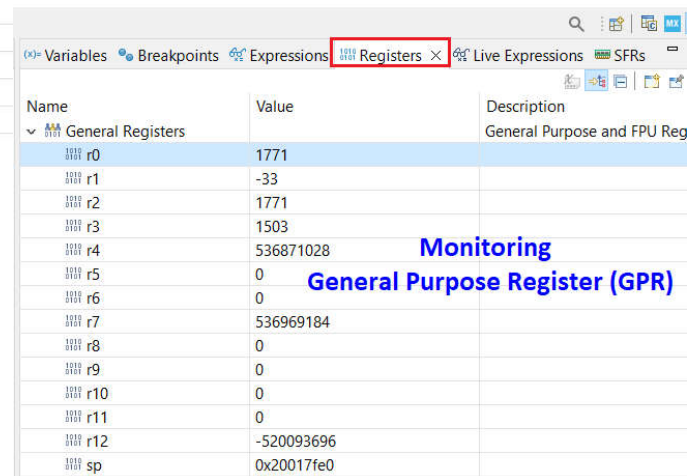
Sebelum Proses Debug Dijalankan

1. Memonitor nilai *variabel-variabel* yang dibuat.
2. Memonitor *General Purpose Register* (GPR) yang digunakan.
3. Memonitor *Special Function Register* (SFR) yang digunakan.



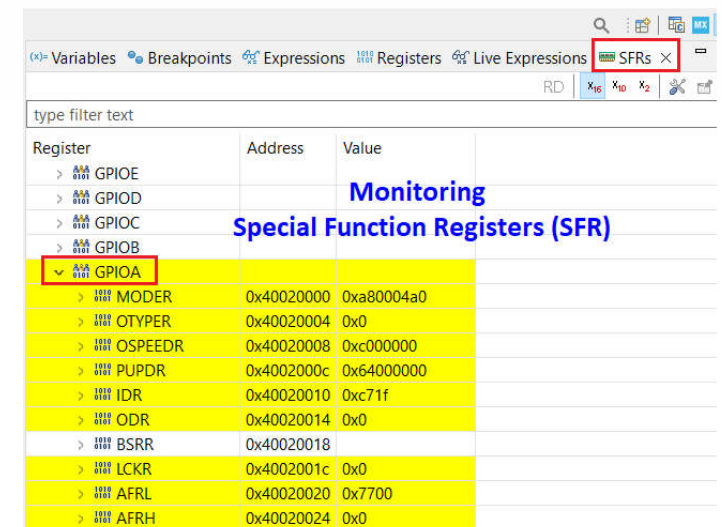
Name	Type	Value
Delay	uint32_t	500
tickstart	uint32_t	1503
wait	uint32_t	501

Monitoring variable



Name	Value	Description
General Registers		General Purpose and FPU Reg
r0	1771	
r1	-33	
r2	1771	
r3	1503	
r4	536871028	
r5	0	
r6	0	
r7	536969184	
r8	0	
r9	0	
r10	0	
r11	0	
r12	-520093696	
sp	0x20017fe0	

Monitoring General Purpose Register (GPR)



Register	Address	Value
GPIOE		
GPIOD		
GPIOC		
GPIOB		
GPIOA		
MODER	0x40020000	0xa80004a0
OTYPER	0x40020004	0x0
OSPEEDR	0x40020008	0xc0000000
PUPDR	0x4002000c	0x64000000
IDR	0x40020010	0xc71f
ODR	0x40020014	0x0
BSRR	0x40020018	
LCKR	0x4002001c	0x0
AFRL	0x40020020	0x7700
AFRH	0x40020024	0x0

Monitoring Special Function Registers (SFR)

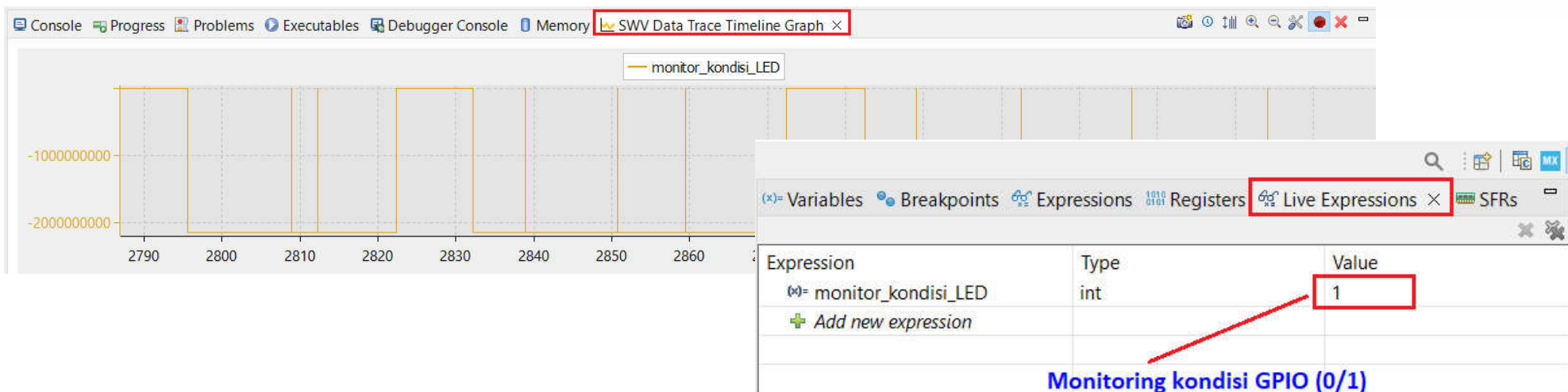
Saat Proses Debug Berjalan

1. Live Expression

Monitoring kondisi GPIO dengan visualisasi berupa **data numerik (1/0)**.

2. Serial Wire Viewer (SWV) Data Trace

Monitoring kondisi GPIO dengan visualisasi berupa **Timeline graphic**.



Untuk dapat melakukan “**Live Expression**” dan “**SWV Data Trace**”,
tambahkan kode program berikut:

```
int monitor_kondisi_LED = 0;  
monitor_kondisi_LED = HAL_GPIO_ReadPin(LD2_GPIO_Port,LD2_Pin);
```

Terima kasih



ROBONESIA.com
more than robotics learning