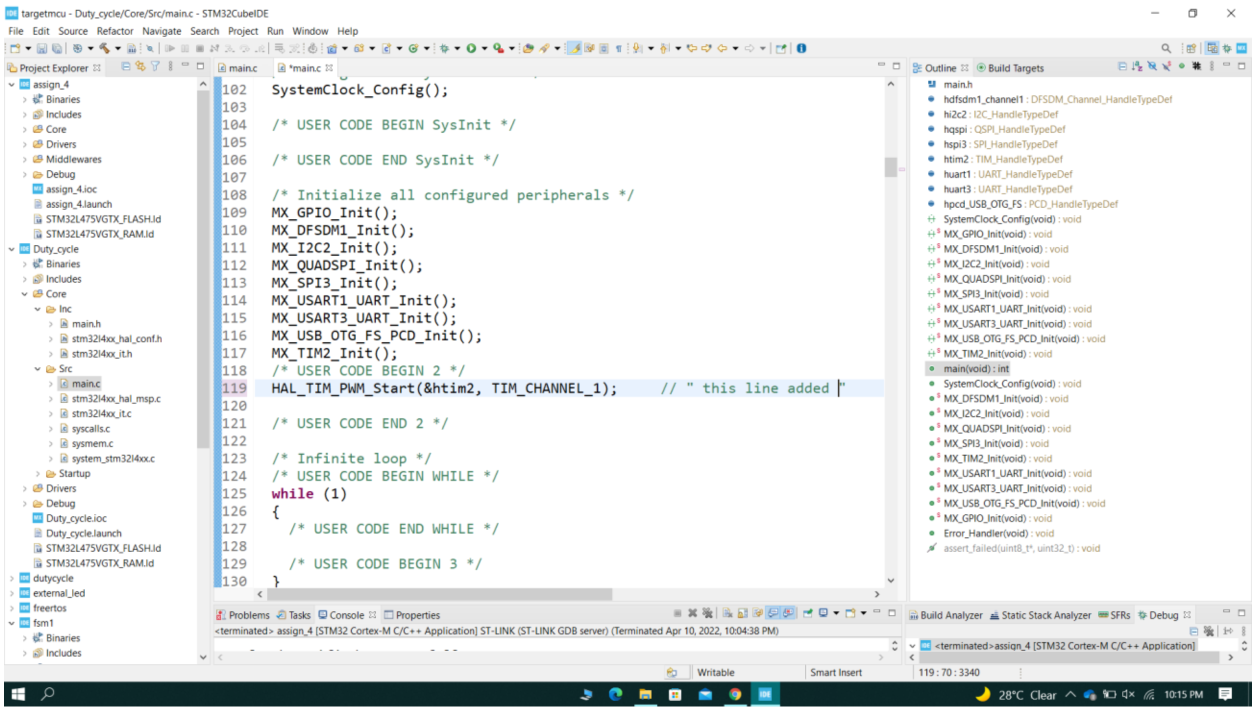
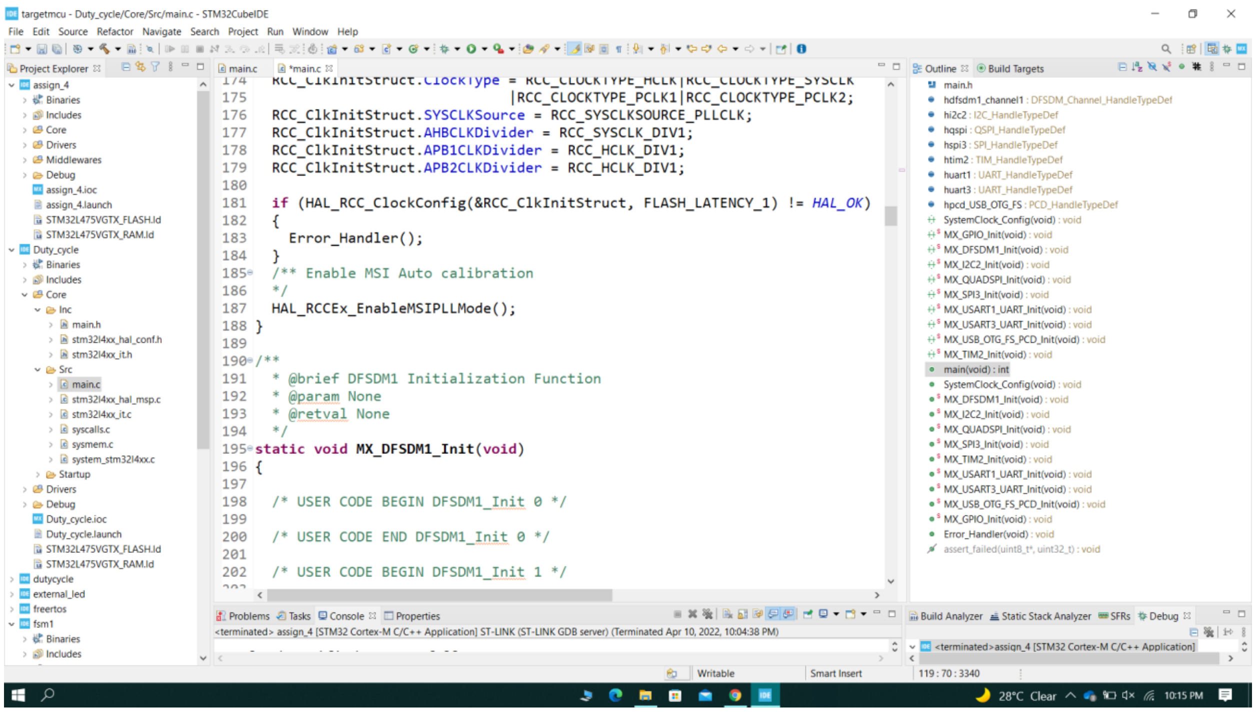
# MCU ASSINGMENT NO : - 3



targetmcu - Duty_cycle/Core/Src/main.c - STM32CubeIDE

```
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DFSDM1_Init();
  MX_I2C2_Init();
  MX_QUADSPI_Init();
  MX_SPI3_Init();
  MX_USART1_UART_Init();
  MX_USART3_UART_Init();
  MX_USB_OTG_FS_PCD_Init();
  MX_TIM2_Init();
  /* USER CODE BEGIN 2 */
  HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);      // " this line added "

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
```

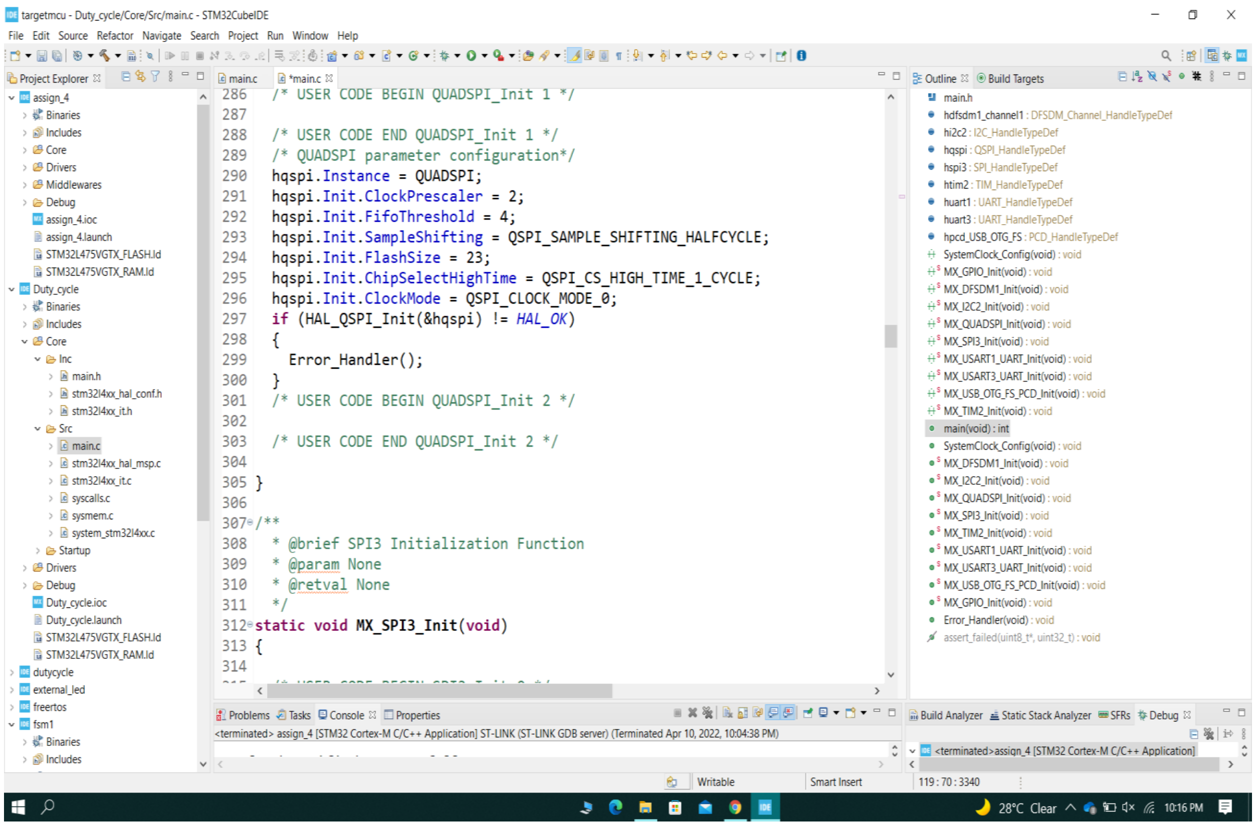
```
<terminated> assign_4 [STM32 Cortex-M C/C++ Application] ST-LINK (ST-LINK GDB server) (Terminated Apr 10, 2022, 10:04:38 PM)
```



targetmcu - Duty_cycle/Core/Src/main.c - STM32CubeIDE

```
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                              |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  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_1) != HAL_OK)
  {
    Error_Handler();
  }
  /** Enable MSI Auto calibration
  */
  HAL_RCCEx_EnableMSIPLLMode();
}

/**
  * @brief DFSDM1 Initialization Function
  * @param None
  * @retval None
  */
static void MX_DFSDM1_Init(void)
{

  /* USER CODE BEGIN DFSDM1_Init 0 */

  /* USER CODE END DFSDM1_Init 0 */

  /* USER CODE BEGIN DFSDM1_Init 1 */
```

```
<terminated> assign_4 [STM32 Cortex-M C/C++ Application] ST-LINK (ST-LINK GDB server) (Terminated Apr 10, 2022, 10:04:38 PM)
```


targetmcu - Duty_cycle/Core/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
main.c
*main.c
Outline
Build Targets
Problems Tasks Console Properties
Build Analyzer Static Stack Analyzer SFRs Debug
Writable
Smart Insert
119 : 70 : 3340

```
  /* USER CODE BEGIN QUADSPI_Init 1 */

  /* USER CODE END QUADSPI_Init 1 */
  /* QUADSPI parameter configuration*/
  hqspi.Instance = QUADSPI;
  hqspi.Init.ClockPrescaler = 2;
  hqspi.Init.FifoThreshold = 4;
  hqspi.Init.SampleShifting = QSPI_SAMPLE_SHIFTING_HALFCYCLE;
  hqspi.Init.FlashSize = 23;
  hqspi.Init.ChipSelectHighTime = QSPI_CS_HIGH_TIME_1_CYCLE;
  hqspi.Init.ClockMode = QSPI_CLOCK_MODE_0;
  if (HAL_QSPI_Init(&hqspi) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN QUADSPI_Init 2 */

  /* USER CODE END QUADSPI_Init 2 */
}

/**
  * @brief SPI3 Initialization Function
  * @param None
  * @retval None
  */
static void MX_SPI3_Init(void)
{

```

```
<terminated> assign_4 [STM32 Cortex-M C/C++ Application] ST-LINK (ST-LINK GDB server) (Terminated Apr 10, 2022, 10:04:38 PM)
```

## ***CODE: -***

```
TIM_ClockConfigTypeDef sClockSourceConfig = {0};
  TIM_MasterConfigTypeDef sMasterConfig = {0};
  TIM_OC_InitTypeDef sConfigOC = {0};
  /* USER CODE BEGIN TIM2_Init 1 */

  /* USER CODE END TIM2_Init 1 */
  htim2.Instance = TIM2;
  htim2.Init.Prescaler = 32000;
  htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
  htim2.Init.Period = 5000;
  htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
  htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_ENABLE;
  if (HAL_TIM_Base_Init(&htim2) != HAL_OK)
  {
    Error_Handler();
  }
  sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
  if (HAL_TIM_ConfigClockSource(&htim2, &sClockSourceConfig) != HAL_OK)
  {
    Error_Handler();
  }
  if (HAL_TIM_PWM_Init(&htim2) != HAL_OK)
  {
    Error_Handler();
  }
  sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
  sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
  if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) !=
HAL_OK)
  {
    Error_Handler();
  }
  sConfigOC.OCMode = TIM_OCMODE_PWM1;
  sConfigOC.Pulse = 2000;
  sConfigOC.OCPolarity = TIM_OCPOLARITY_HIGH;
  sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
  if (HAL_TIM_PWM_ConfigChannel(&htim2, &sConfigOC, TIM_CHANNEL_1) !=
HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN TIM2_Init 2 */

  /* USER CODE END TIM2_Init 2 */
  HAL_TIM_MspPostInit(&htim2);

}

/**
  * @brief USART1 Initialization Function
  * @param None
  * @retval None
  */
static void MX_USART1_UART_Init(void)
{

  /* USER CODE BEGIN USART1_Init 0 */

  /* USER CODE END USART1_Init 0 */

  /* USER CODE BEGIN USART1_Init 1 */

  /* USER CODE END USART1_Init 1 */
  huart1.Instance = USART1;
  huart1.Init.BaudRate = 115200;
  huart1.Init.WordLength = UART_WORDLENGTH_8B;
  huart1.Init.StopBits = UART_STOPBITS_1;
  huart1.Init.Parity = UART_PARITY_NONE;
  huart1.Init.Mode = UART_MODE_TX_RX;
  huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
  huart1.Init.OverSampling = UART_OVERSAMPLING_16;
  huart1.Init.OneBitSampling = UART_ONE_BIT_SAMPLE_DISABLE;
  huart1.AdvancedInit.AdvFeatureInit = UART_ADVFEATURE_NO_INIT;
  if (HAL_UART_Init(&huart1) != HAL_OK)
  {
    Error_Handler();
  }
```

```
/* USER CODE BEGIN USART1_Init 2 */

/* USER CODE END USART1_Init 2 */

}
```