main.c

```
/* USER CODE BEGIN Header */
/**********************************************************************************
  * @file           : main.c
  * @brief          : Main program body
  **********************************************************************************
  * @attention
  *
  * Copyright (c) 2023 STMicroelectronics.
  * All rights reserved.
  *
  * This software is licensed under terms that can be found in the LICENSE file
  * in the root directory of this software component.
  * If no LICENSE file comes with this software, it is provided AS-IS.
  *
  **********************************************************************************/
/* USER CODE END Header */
/* Includes ------------------------------------------------------------------*/
#include "main.h"
/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
/* USER CODE END Includes */
/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN PTD */
/* USER CODE END PTD */
/* Private define ------------------------------------------------------------*/
/* USER CODE BEGIN PD */
/* USER CODE END PD */
/* Private macro -------------------------------------------------------------*/
/* USER CODE BEGIN PM */
/* USER CODE END PM */
/* Private variables ---------------------------------------------------------*/
CAN_HandleTypeDef hcan1;
/* USER CODE BEGIN PV */
/* USER CODE END PV */
```

main.c

```
/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_CAN1_Init(void);
/* USER CODE BEGIN PFP */
/* USER CODE END PFP */
/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
CAN_TxHeaderTypeDef uTxHeader;
CAN_RxHeaderTypeDef uRxHeader;

uint32_t uTxMailbox;
uint8_t uTxData[8];
uint8_t uRxData[8];
uint8_t uCount = 0;

void HAL_CAN_RxFifo0MsgPendingCallback(CAN_HandleTypeDef *hcan1)
{
	if (HAL_CAN_GetRxMessage(hcan1, CAN_RX_FIFO0, &uRxHeader, uRxData) != HAL_OK)
	 {
		Error_Handler();
	 }
	if (uCount >= 15)
		{
		uCount = 0;
		}
	else
	{
		uCount++;
	}
}
/* USER CODE END 0 */
/**
  * @brief  The application entry point.
```

main.c

```
 * @retval int
 */
int main(void)
{
 /* USER CODE BEGIN 1 */
 /* USER CODE END 1 */
 /* MCU Configuration--------------------------------------------------------*/
 /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
 HAL_Init();
 /* USER CODE BEGIN Init */
 /* USER CODE END Init */
 /* Configure the system clock */
 SystemClock_Config();
 /* USER CODE BEGIN SysInit */
 /* USER CODE END SysInit */
 /* Initialize all configured peripherals */
 MX_GPIO_Init();
 MX_CAN1_Init();
 /* USER CODE BEGIN 2 */
 HAL_CAN_Start(&hcan1);
 HAL_CAN_ActivateNotification(&hcan1, CAN_IT_RX_FIFO0_MSG_PENDING);

uTxHeader.DLC = 1;
 uTxHeader.ExtId = 0;
 uTxHeader.IDE = CAN_ID_STD;
 uTxHeader.RTR = CAN_RTR_DATA;
 uTxHeader.StdId = 0x00A;
 uTxHeader.TransmitGlobalTime = DISABLE;
 /* USER CODE END 2 */
 /* Infinite loop */
 /* USER CODE BEGIN WHILE */
 while (1)
 {
   /* USER CODE END WHILE */
```

main.c

```
  /* USER CODE BEGIN 3 */
            uTxData[0] = uCount;
            HAL_CAN_AddTxMessage(&hcan1, &uTxHeader, uTxData, &uTxMailbox);
            HAL_GPIO_WritePin(GPIOA, LED_IN_PA5_Pin, 1);
            HAL_Delay(50);
            HAL_GPIO_WritePin(GPIOA, LED_IN_PA5_Pin, 0);
            HAL_Delay(50);
 }
 /* USER CODE END 3 */
}
/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
 RCC_OscInitTypeDef RCC_OscInitStruct = {0};
 RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
 /** Configure the main internal regulator output voltage
 */
 __HAL_RCC_PWR_CLK_ENABLE();
 __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE3);
 /** Initializes the RCC Oscillators according to the specified parameters
 * in the RCC_OscInitTypeDef structure.
 */
 RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
 RCC_OscInitStruct.HSIState = RCC_HSI_ON;
 RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
 RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
 RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
 RCC_OscInitStruct.PLL.PLLM = 8;
 RCC_OscInitStruct.PLL.PLLN = 80;
 RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
 RCC_OscInitStruct.PLL.PLLQ = 2;
```

main.c

```
  RCC_OscInitStruct.PLL.PLLR = 2;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    Error_Handler();
  }
  /** Initializes the CPU, AHB and APB buses clocks
  */
  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_DIV2;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;
  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
  {
    Error_Handler();
  }
}
/**
  * @brief CAN1 Initialization Function
  * @param None
  * @retval None
  */
static void MX_CAN1_Init(void)
{
  /* USER CODE BEGIN CAN1_Init 0 */
  /* USER CODE END CAN1_Init 0 */
  /* USER CODE BEGIN CAN1_Init 1 */
  /* USER CODE END CAN1_Init 1 */
  hcan1.Instance = CAN1;
  hcan1.Init.Prescaler = 20;
  hcan1.Init.Mode = CAN_MODE_LOOPBACK;
  hcan1.Init.SyncJumpWidth = CAN_SJW_1TQ;
  hcan1.Init.TimeSeg1 = CAN_BS1_13TQ;
```

main.c

```
  hcan1.Init.TimeSeg2 = CAN_BS2_2TQ;
  hcan1.Init.TimeTriggeredMode = DISABLE;
  hcan1.Init.AutoBusOff = DISABLE;
  hcan1.Init.AutoWakeUp = DISABLE;
  hcan1.Init.AutoRetransmission = DISABLE;
  hcan1.Init.ReceiveFifoLocked = DISABLE;
  hcan1.Init.TransmitFifoPriority = DISABLE;
  if (HAL_CAN_Init(&hcan1) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN CAN1_Init 2 */
  CAN_FilterTypeDef uCAN_FilterConfig;

  uCAN_FilterConfig.FilterActivation = CAN_FILTER_ENABLE;
  uCAN_FilterConfig.FilterBank = 0;
  uCAN_FilterConfig.FilterFIFOAssignment = CAN_RX_FIFO0;
  uCAN_FilterConfig.FilterIdHigh = 0x000<<5;
  uCAN_FilterConfig.FilterIdLow = 0x000;
  uCAN_FilterConfig.FilterMaskIdHigh = 0x000<<5;
  uCAN_FilterConfig.FilterMaskIdLow = 0x000;
  uCAN_FilterConfig.FilterMode = CAN_FILTERMODE_IDMASK;
  uCAN_FilterConfig.FilterScale = CAN_FILTERSCALE_32BIT;
  uCAN_FilterConfig.SlaveStartFilterBank = 14;

  HAL_CAN_ConfigFilter(&hcan1, &uCAN_FilterConfig);
  /* USER CODE END CAN1_Init 2 */
}
/**
  * @brief GPIO Initialization Function
  * @param None
  * @retval None
  */
static void MX_GPIO_Init(void)
```

main.c

```
{
  GPIO_InitTypeDef GPIO_InitStruct = {0};
/* USER CODE BEGIN MX_GPIO_Init_1 */
/* USER CODE END MX_GPIO_Init_1 */
  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOA_CLK_ENABLE();
  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(LED_IN_PA5_GPIO_Port, LED_IN_PA5_Pin, GPIO_PIN_RESET);
  /*Configure GPIO pin : LED_IN_PA5_Pin */
  GPIO_InitStruct.Pin = LED_IN_PA5_Pin;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(LED_IN_PA5_GPIO_Port, &GPIO_InitStruct);
/* USER CODE BEGIN MX_GPIO_Init_2 */
/* USER CODE END MX_GPIO_Init_2 */
}
/* USER CODE BEGIN 4 */
/* USER CODE END 4 */
/**
  * @brief  This function is executed in case of error occurrence.
  * @retval None
  */
void Error_Handler(void)
{
  /* USER CODE BEGIN Error_Handler_Debug */
  /* User can add his own implementation to report the HAL error return state */
  __disable_irq();
  while (1)
  {
  }
  /* USER CODE END Error_Handler_Debug */
}
#ifdef  USE_FULL_ASSERT
```

main.c

```
/**
  * @brief  Reports the name of the source file and the source line number
  *         where the assert_param error has occurred.
  * @param  file: pointer to the source file name
  * @param  line: assert_param error line source number
  * @retval None
  */
void assert_failed(uint8_t *file, uint32_t line)
{
  /* USER CODE BEGIN 6 */
  /* User can add his own implementation to report the file name and line number,
     ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
  /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */
```