Electrical and Computer Engineering Department
ECE 4510 Microcontroller Applications

Lab 7

Control Area Network (CAN) Interface

Isaac Bagley and Rohullah Sah

March 3, 2023

Introduction

The goal of this lab was to understand the basics of the CAN interface and to use the CAN Module of the STM3F4 microcontroller along with an external CAN transceiver module to implement a CAN communications port. Once we learnt how to develop a C program that allows us to use the CAN module to transmit and program to receive, we then combined what we learned and developed a program that can interface with a Hyundai Elantra instrument cluster to control the speedometer and the RPM. Apart from this, we had a bonus question which was to develop a C program that utilizes the TX and RX Error Counters to characterize the stability of the CAN network and we did.

Procedure

Task 1

As for Task1, we were to develop a C program which would send a prepared message of 8 bytes through the CAN bus in an infinite loop at every T ms and also blink an LED on the DISC1 board. Once we got through this, we then had to take a few screenshots from an oscilloscope.

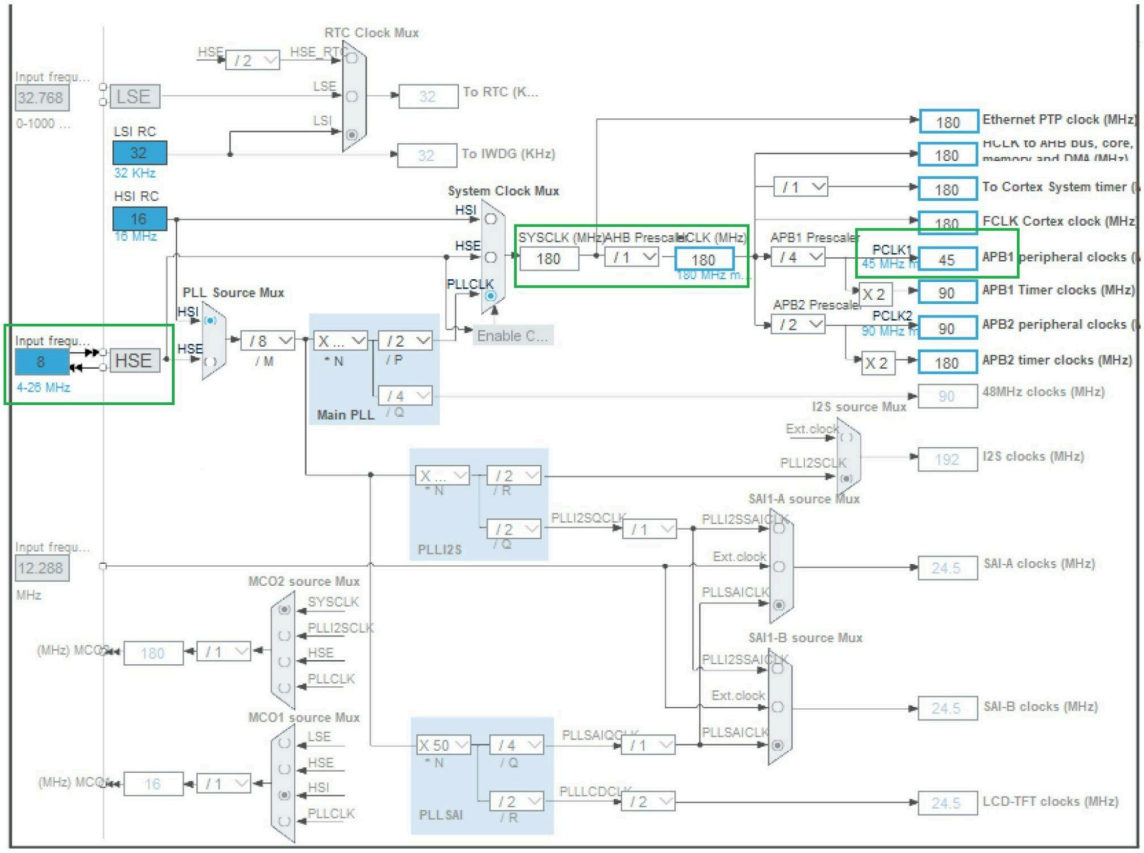


Figure 1: Clock Configuration for Task1 shown in figure.

To change Baud Rate to 500 kHz and the Nominal-Bit-Time NBT to 2us, we had set HCLK to 180 MHz and APB1 to 45MHz. As you can see in the picture below, the Baud Rate and NBT has changed to what we intended to and the rest of the settings are changed manually as per the parameters given.

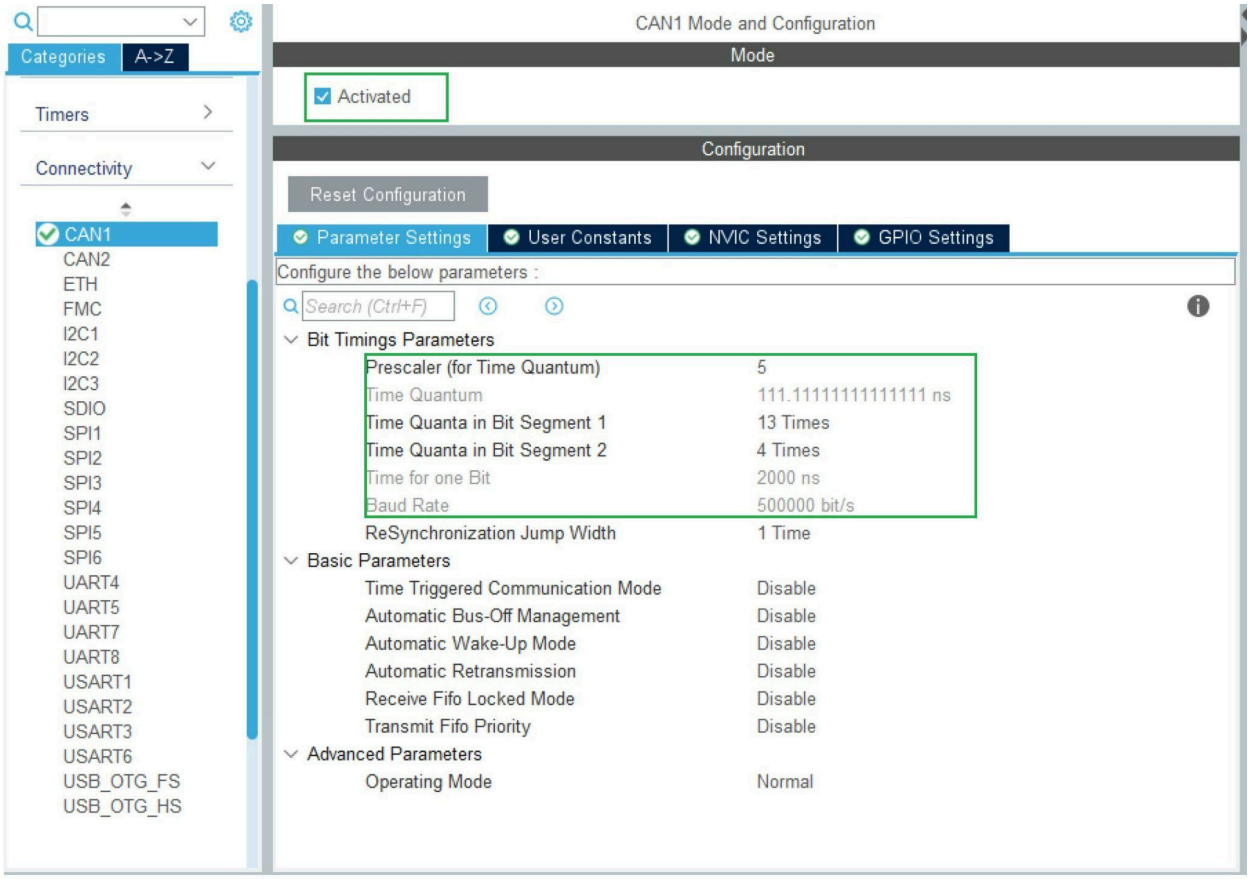


Figure 2: CAN1 Mode and Configuration

By clicking on the CAN1 option on the category list, we then had to activate the mode and set the Bit Timings Parameters just like the figure above. This way, the Pins enable too according to their functionality.

Figure 3: Pinout Configuration for Task1 which shows PA11(CAN1_RX) and PA12(CAN1_TX) enabled.

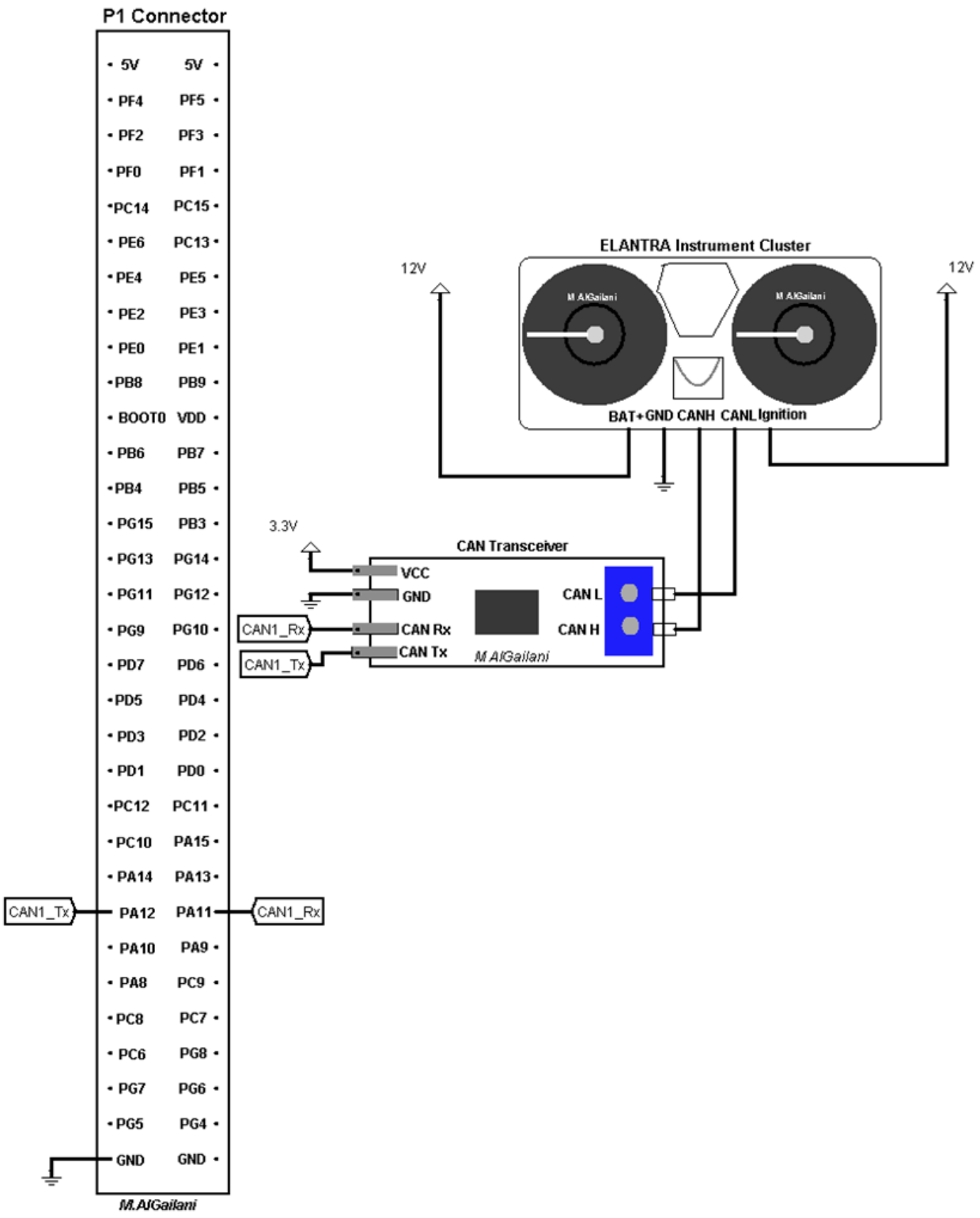


Figure 4: Schematic Diagram

Task 2

For this Task, we had to develop a C program that would set ID filters for the CAN Module to receive messages with only two predetermined ID's even though the Instrument Cluster will continuously broadcast its own messages non-stop and to toggle the Green LED each time a message is received.

Set up for ID filters are:

i. Set a 16-bit scale ID List filter to detect IDs 0x690 and 0x4F0.

ii. ID Mask filter to detect all IDs from 0x600 to 0x7FF.

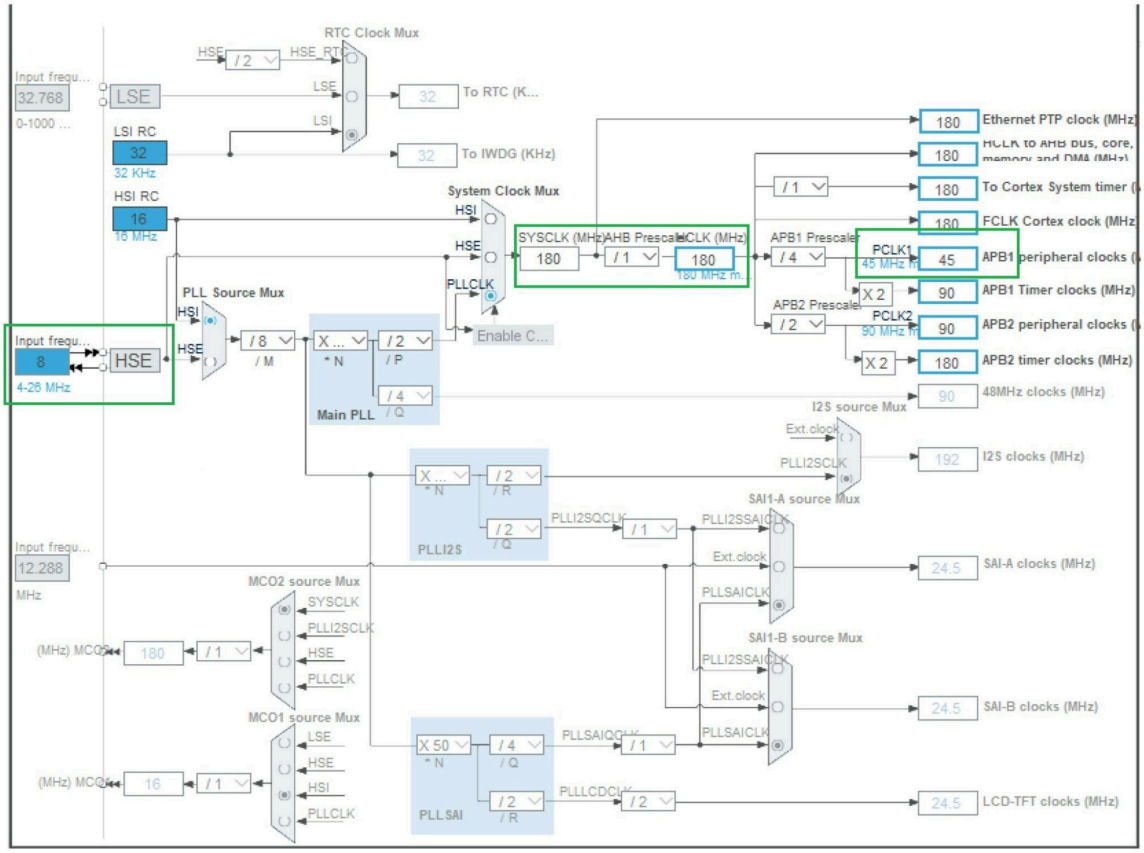


Figure 5: Clock Configuration for Task1 shown in figure.

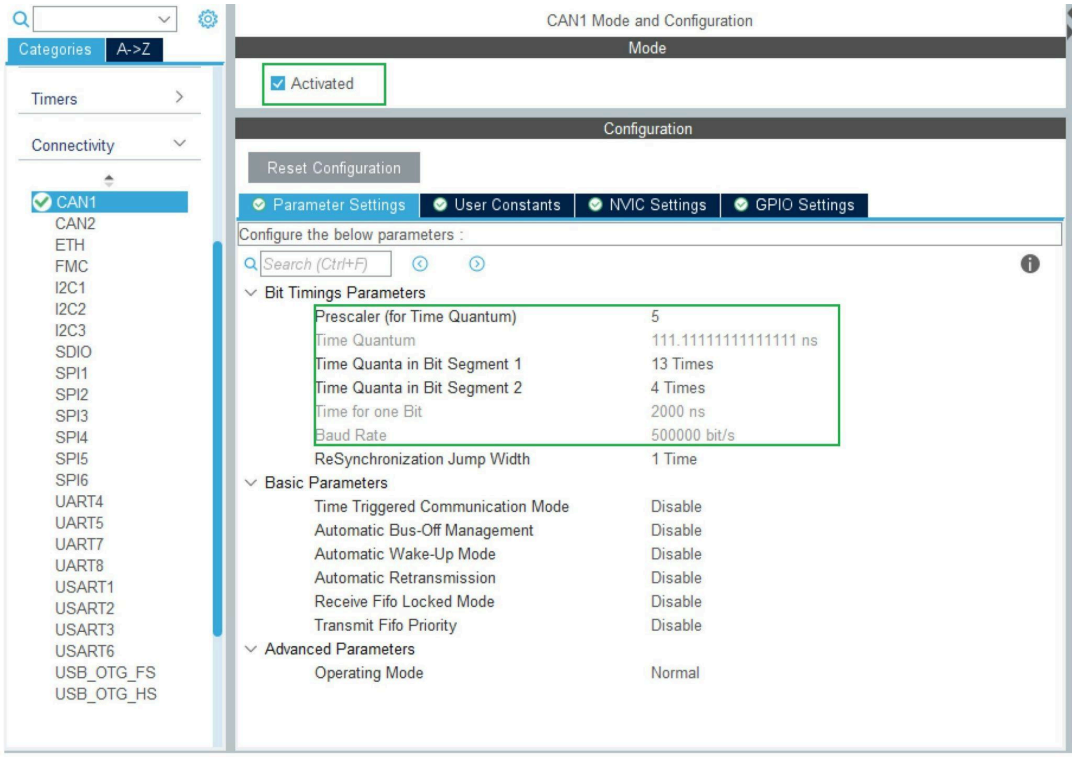


Figure 6: CAN1 Mode and Configuration

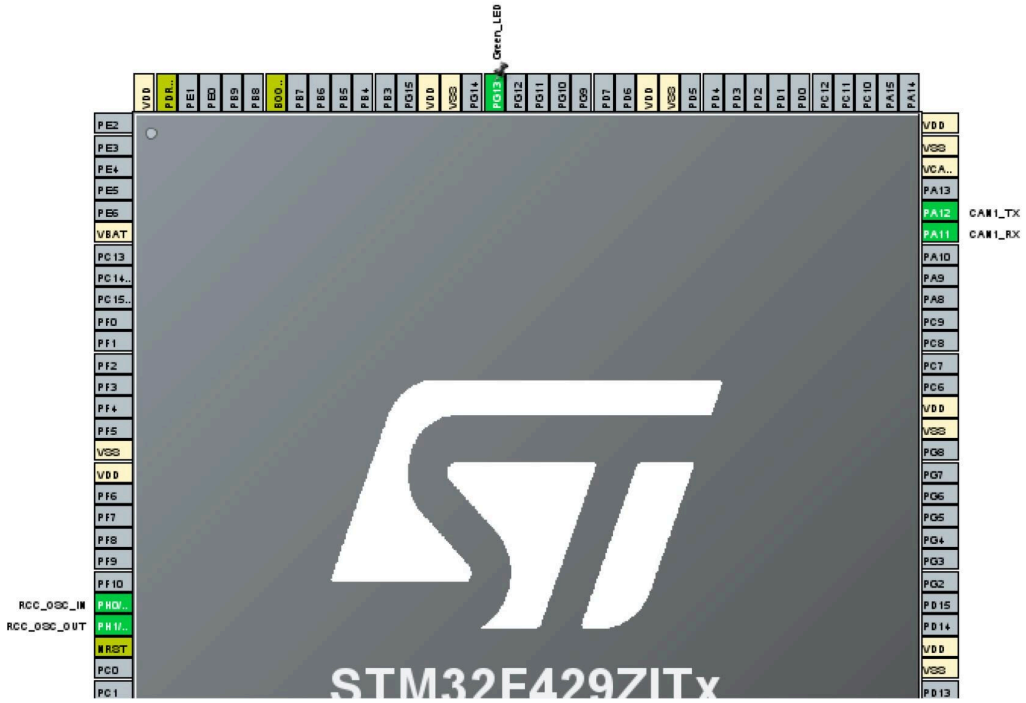


Figure 7: Pinout Configuration

As shown in the above figure, Port G Pin 13 has been enabled to toggle a Green LED on the DISC1 board and the rest are enabled through CAN1.

Task 3 with Bonus

The last part was to develop a C program that would allow us to use the CAN Module to control the Speedometer and RPM dials of the Hyundai Elantra instrument cluster. The program was to work with the blue user button(PA0) on the DISC1 Board to increment and decrement the values of the Speed and RPM dials. Also for the bonus part, we were to develop a C program that utilizes the TX and RX Error Counters to characterize the stability of the CAN network. If an error has been detected then the contents of the counters will be incremented. If the network recovers, the contents of the counter will be decreased. The program would then print the contents of the counters and turn off an LED on the DISC1 Board when the TX Error Counter value is smaller than 1. The LED should be turned on when the TX Error Counter contents reach 1.

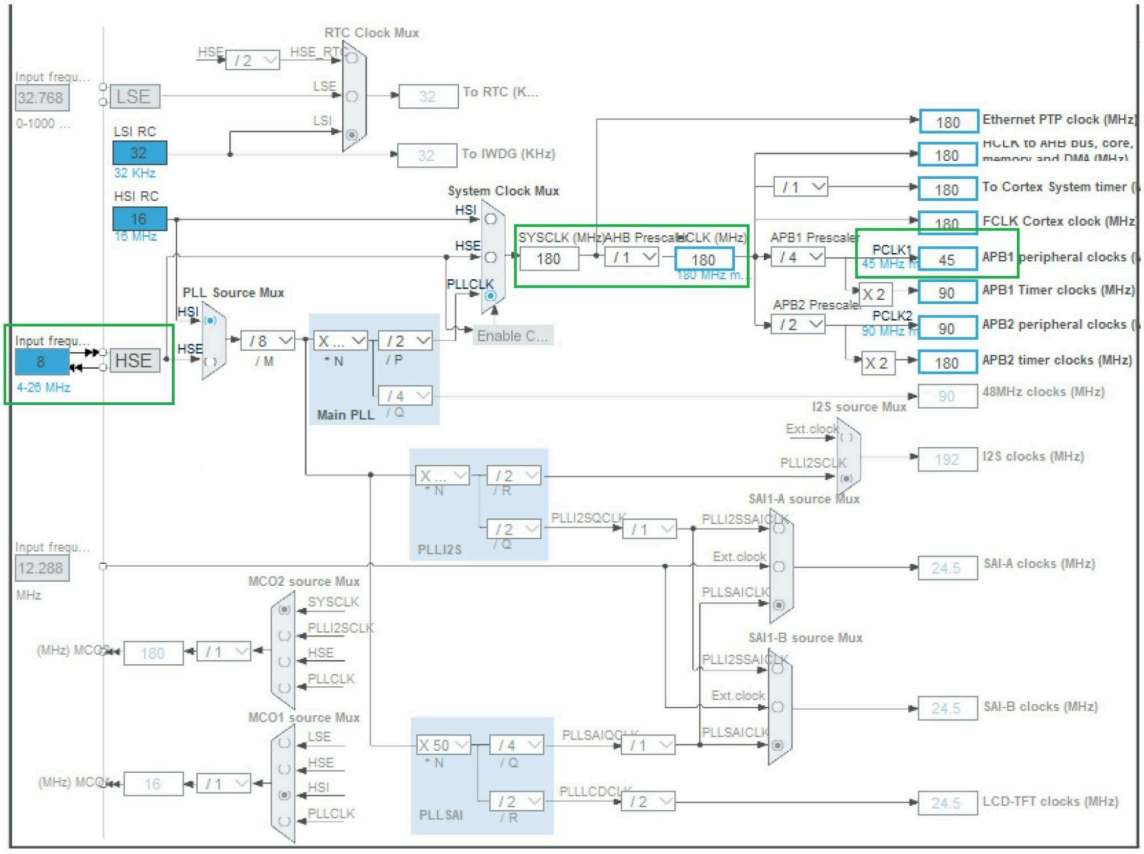


Figure 8: Clock Configuration for Task1 shown in figure.

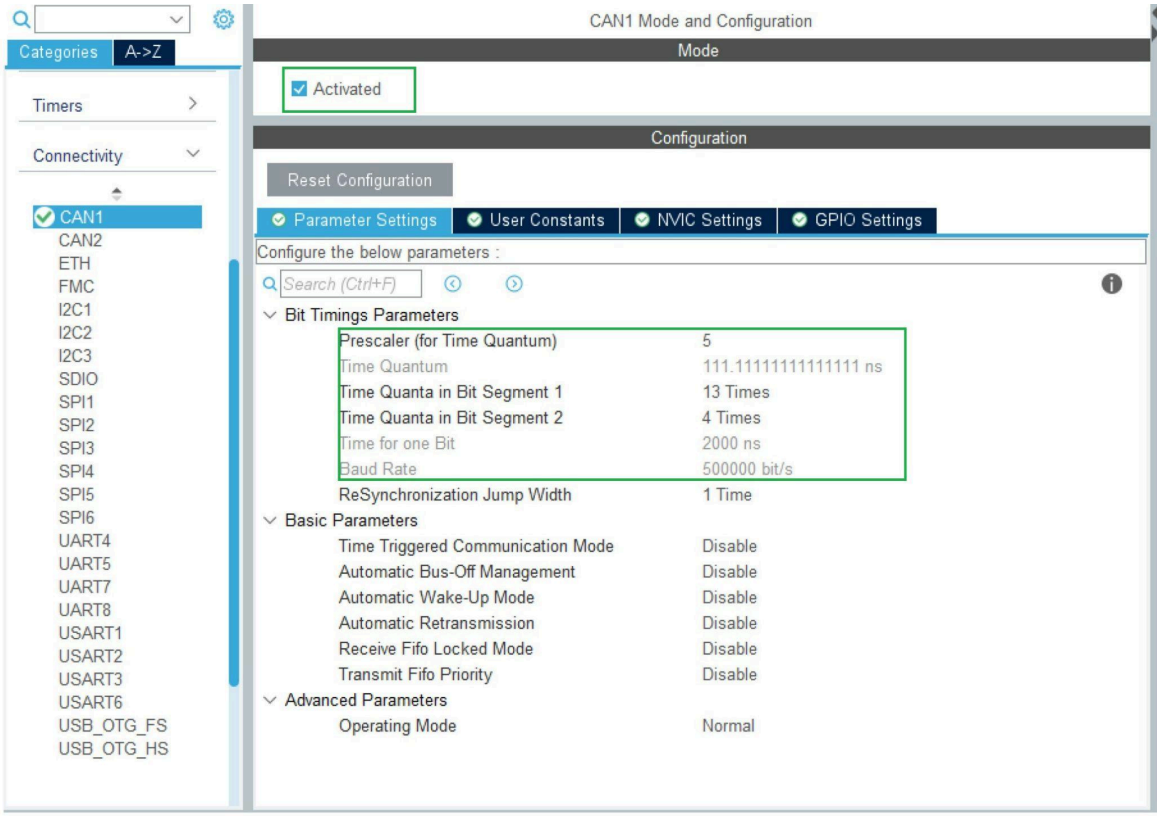


Figure 9: CAN1 Mode and Configuration

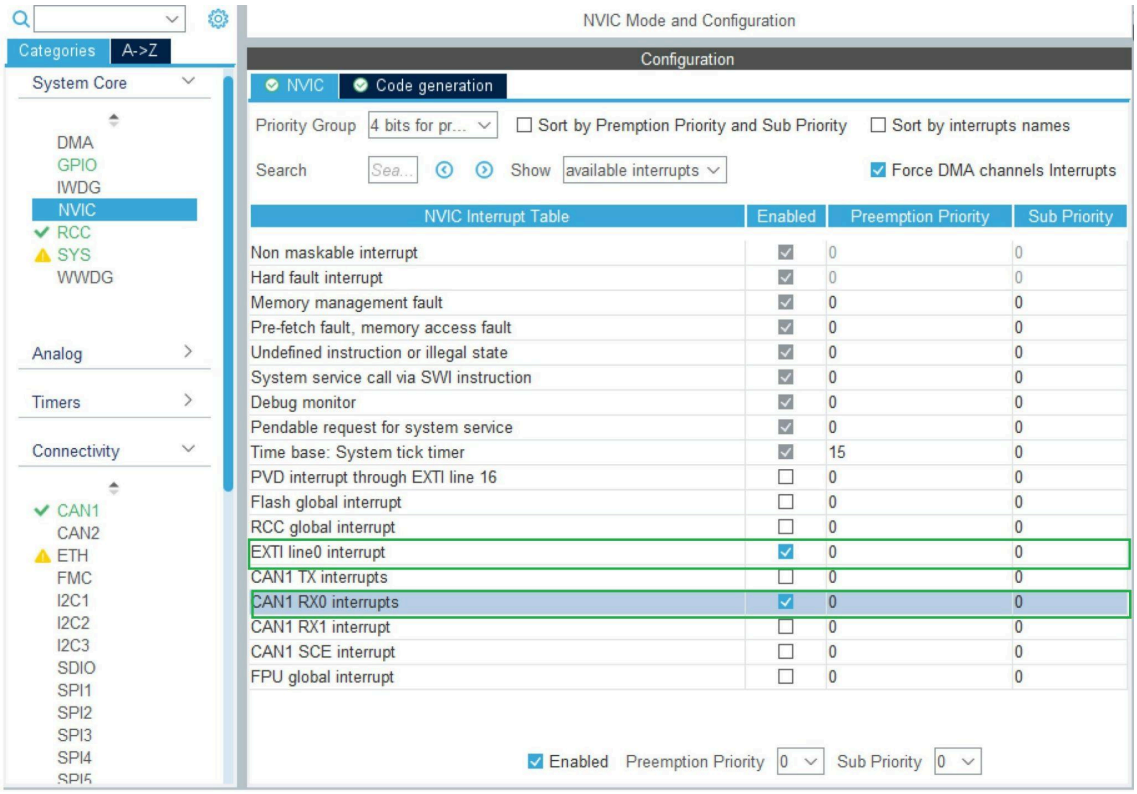


Figure 10: Enabling Interrupts.

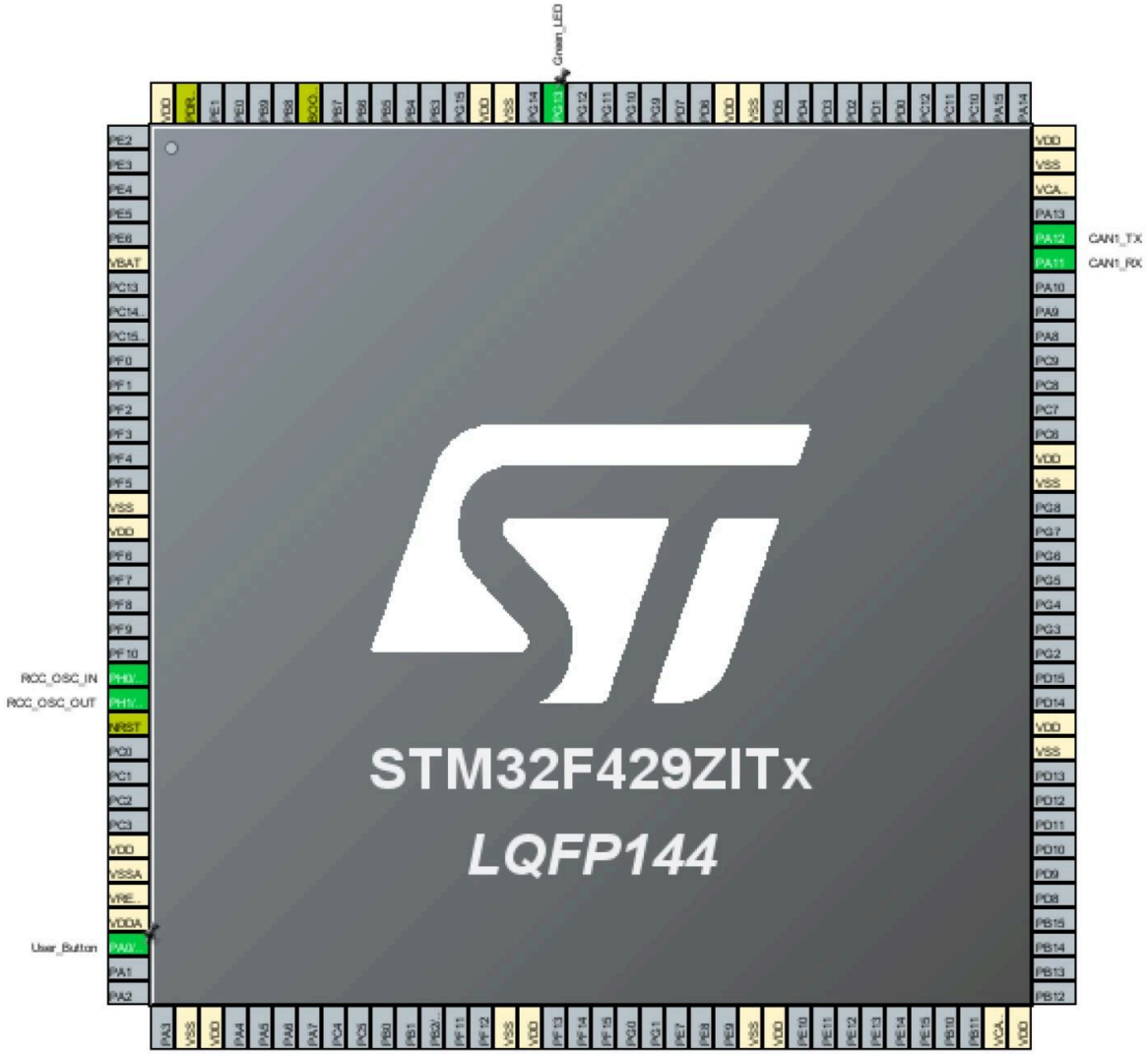


Figure 11: Port A Pin 0 acts as the user button on the MCU.

Results

Task 1

For task 1, we read the message sent from the microcontroller through the CAN Tx signal.

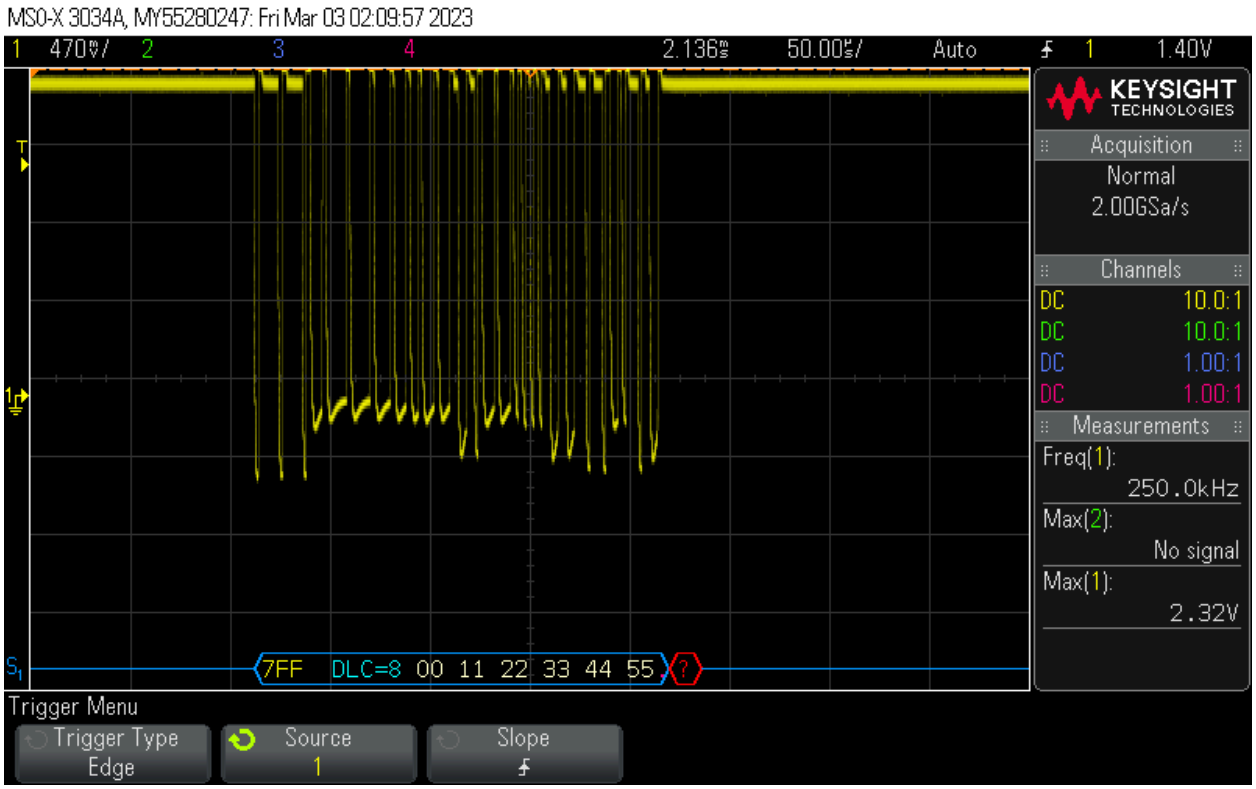
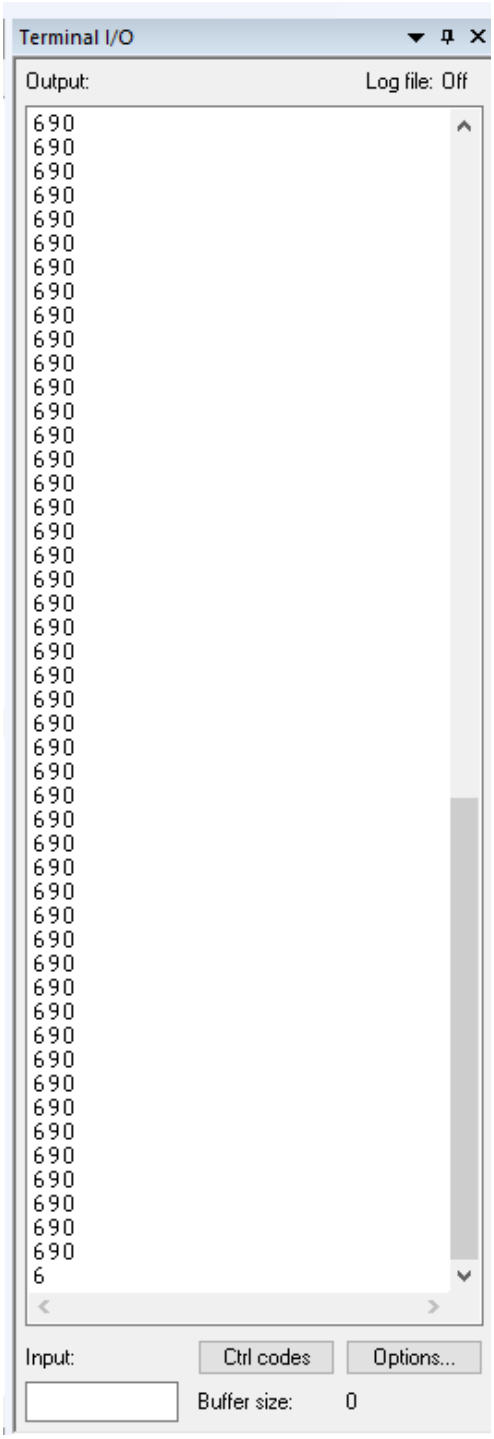


Figure 12: Values read and sent through Oscilloscope as specified by the prelab

Task 2

For task 2 we set up a message ID mask to only look for messages related to the speedometer with an ID of 690.


Terminal I/O
Output:
Log file: Off
690
690
690

Task 3 with Bonus

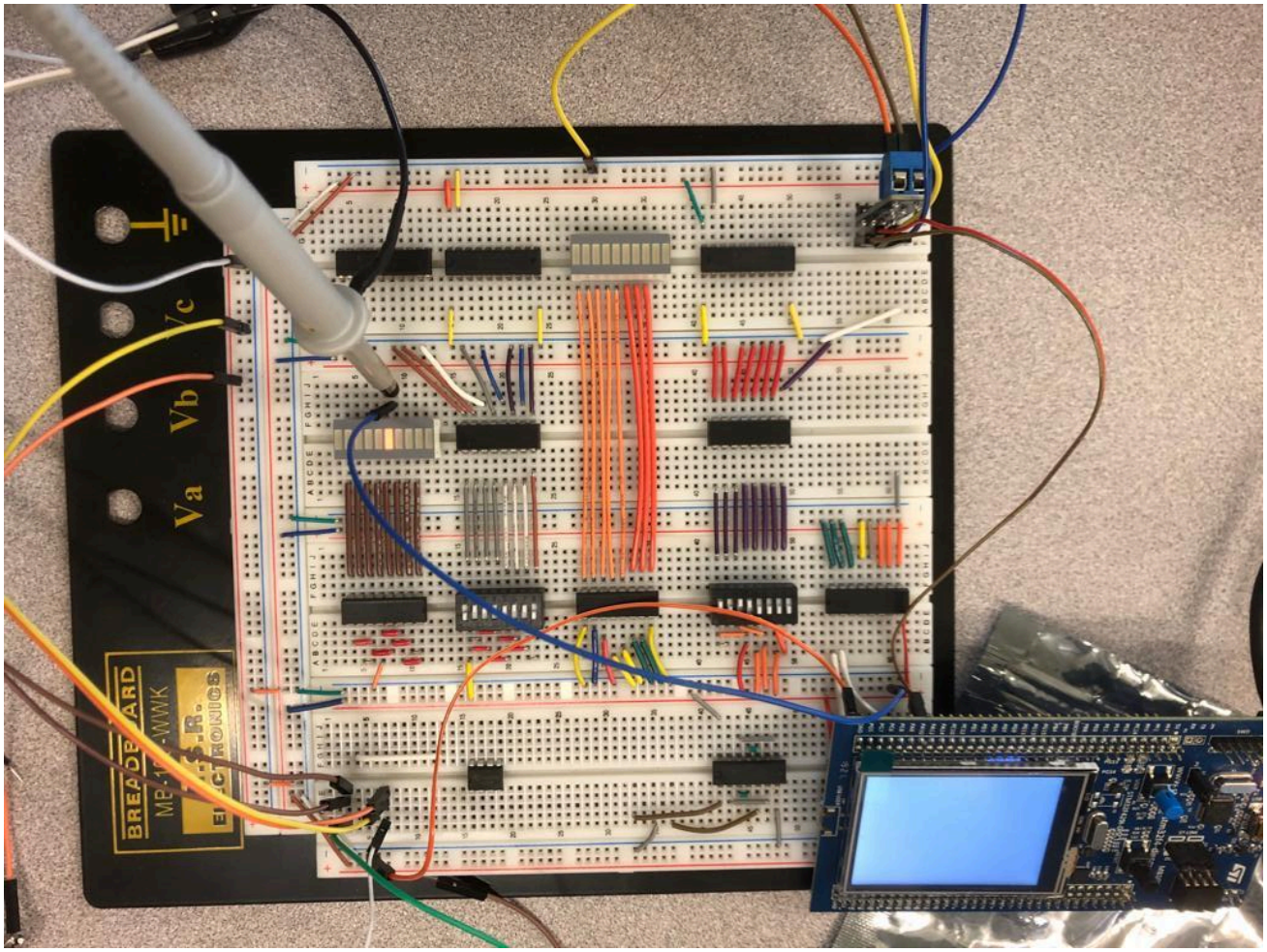


Figure 14: LED showing TX Error Counter reaches 1. The bonus.

Figure 15: The Elantra cluster working correctly showing a speed of 120 and RPM of 6000.

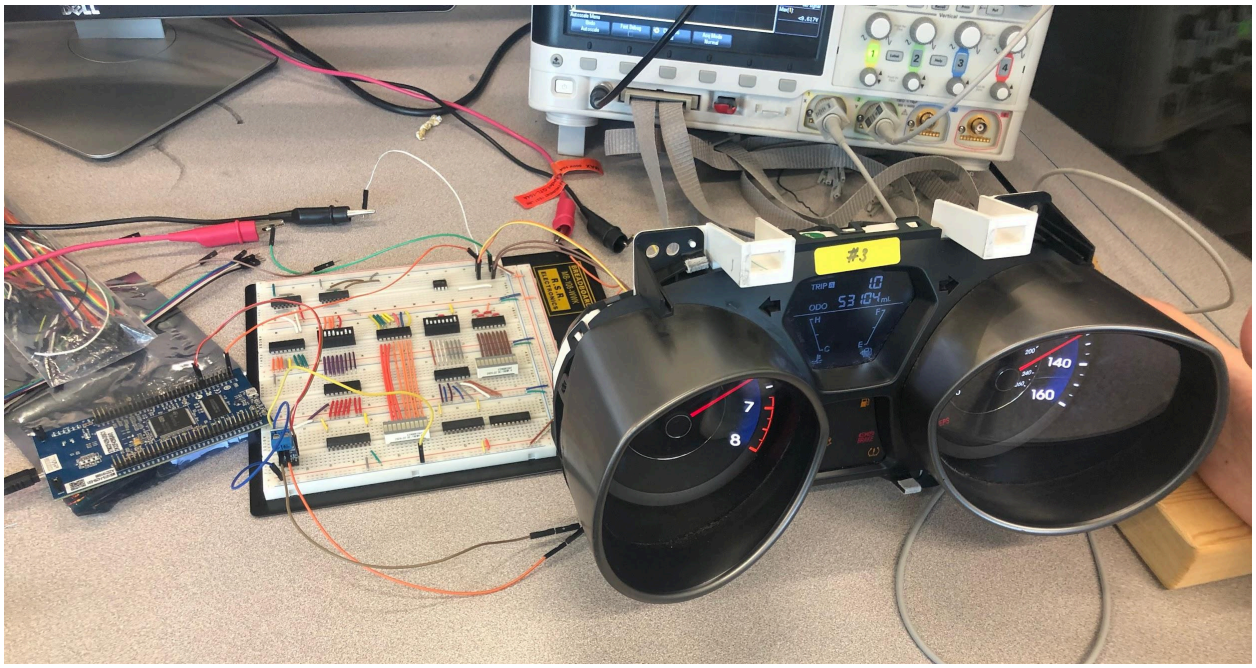


Figure 16: the full setup for task 3 before the bonus

Conclusion

This lab served as an excellent introduction to the CAN module and how to send and receive data with it. We can filter the IDs based on List or Mask options in addition to simply obtaining data, so that we only receive information that is pertinent to us. We can transmit the same message continuously using the ID of the data we are delivering as a basis for verification. Also, we were able to control the speedometer and the RPM dial using our CAN module and C program to interact with the instrument cluster of a Hyundai Elantra. Although we had a hard time doing the bonus question, we did it eventually with the help of our TA, Tony.

Appendix

C Code

Task 1

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 */
CAN_TxHeaderTypeDef TxHeader;
uint8_t data []={0x00,0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77}; // 8-byte data frame
uint32_t mailbox; // Actually sending the data

/* USER CODE END PV */

/* 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 */

/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  * @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 */
   TxHeader.StdId = 0x7FF; //maximum 11 bits wide
 TxHeader.DLC = 8; //(Data Length Code) length of the message
 TxHeader.IDE = CAN_ID_STD; //(Identifier Extension) standard
 TxHeader.RTR = CAN_RTR_DATA; //(Remote Transmission Request) type of frame is
Data
   HAL_CAN_Start(&hcan1);
   HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
  HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);  // send the data in
a loop
   HAL_Delay(1);
          /* USER CODE END WHILE */

          /* USER CODE BEGIN 3 */
  }
  /* 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_SCALE1
);

  /** Initializes the RCC Oscillators according to the specified parameters
  * in the RCC_OscInitTypeDef structure.
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
  RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
  RCC_OscInitStruct.PLL.PLLM = 4;
  RCC_OscInitStruct.PLL.PLLN = 180;
  RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
  RCC_OscInitStruct.PLL.PLLQ = 4;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
        Error_Handler();
  }

  /** Activate the Over-Drive mode
  */
  if (HAL_PWREx_EnableOverDrive() != 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_DIV4;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV4;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) != 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 = 5;
  hcan1.Init.Mode = CAN_MODE_NORMAL;
  hcan1.Init.SyncJumpWidth = CAN_SJW_1TQ;
  hcan1.Init.TimeSeg1 = CAN_BS1_13TQ;
  hcan1.Init.TimeSeg2 = CAN_BS2_4TQ;
  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 */

  /* USER CODE END CAN1_Init 2 */

}

/**
  * @brief GPIO Initialization Function
  * @param None
  * @retval None
  */
static void MX_GPIO_Init(void)
{

  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOH_CLK_ENABLE();
  __HAL_RCC_GPIOA_CLK_ENABLE();

}

/* 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
/**
  * @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 */
```

Task 2

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 */
CAN_RxHeaderTypeDef RxHeader; //Rx refers as input
uint8_t datarx [8];
CAN_FilterTypeDef filter; //Set filter for receiving specific ID message
CAN_RxHeaderTypeDef RxHeader; //Rx refers as input

/* USER CODE END PV */

/* 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 */

/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  * @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 */
  filter.FilterActivation = CAN_FILTER_ENABLE; //enabling filter
  filter.FilterBank = 0; //0-13
  filter.SlaveStartFilterBank = 0;//0-13
  filter.FilterScale = CAN_FILTERSCALE_16BIT; //16 or 32 bits
  filter.FilterFIFOAssignment = CAN_FILTER_FIFO0; //FIFO 0 or 1
  filter.FilterMode = CAN_FILTERMODE_IDMASK;
  filter.FilterIdHigh = 0x600 << 5; // shift 5 left. Because 16-bit scale filter for the 11-bit
CAN standard ID
  filter.FilterIdLow = 0x600 << 5; // repeat because using 16 bits
  filter.FilterMaskIdHigh = 0x600 << 5; // only looking at 10th and 9th bit
  filter.FilterMaskIdLow = 0x600 << 5; // only looking at 10th and 9th bit


  HAL_CAN_Start(&hcan1);
  HAL_CAN_ConfigFilter(&hcan1, &filter);
  HAL_CAN_ActivateNotification(&hcan1, CAN_IT_RX_FIFO0_MSG_PENDING);


  /* USER CODE END 2 */

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

        /* USER CODE BEGIN 3 */
  }
  /* 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_SCALE1
);

  /** Initializes the RCC Oscillators according to the specified parameters
  * in the RCC_OscInitTypeDef structure.
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
  RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
  RCC_OscInitStruct.PLL.PLLM = 4;
  RCC_OscInitStruct.PLL.PLLN = 180;
  RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
  RCC_OscInitStruct.PLL.PLLQ = 4;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
        Error_Handler();
  }

  /** Activate the Over-Drive mode
  */
  if (HAL_PWREx_EnableOverDrive() != 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_DIV4;
```

```
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV4;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) != 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 = 5;
  hcan1.Init.Mode = CAN_MODE_NORMAL;
  hcan1.Init.SyncJumpWidth = CAN_SJW_1TQ;
  hcan1.Init.TimeSeg1 = CAN_BS1_13TQ;
  hcan1.Init.TimeSeg2 = CAN_BS2_4TQ;
  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 */
```

```
  /* USER CODE END CAN1_Init 2 */

}

/**
  * @brief GPIO Initialization Function
  * @param None
  * @retval None
  */
static void MX_GPIO_Init(void)
{

  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOH_CLK_ENABLE();
  __HAL_RCC_GPIOA_CLK_ENABLE();

}

/* 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
/**
```

```
  * @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 */
```

It.c

```
/* USER CODE BEGIN Header */
/**
  ******************************************************************************
  * @file         stm32f4xx_it.c
  * @brief   Interrupt Service Routines.
  ******************************************************************************
  * @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"
#include "stm32f4xx_it.h"
/* Private includes ----------------------------------------------------------*/
```

```
/* USER CODE BEGIN Includes */
#include "stdio.h"
/* USER CODE END Includes */

/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN TD */

/* USER CODE END TD */

/* Private define ------------------------------------------------------------*/
/* USER CODE BEGIN PD */

/* USER CODE END PD */

/* Private macro -------------------------------------------------------------*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables ---------------------------------------------------------*/
/* USER CODE BEGIN PV */

/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/* External variables --------------------------------------------------------*/
extern CAN_HandleTypeDef hcan1;
/* USER CODE BEGIN EV */
extern uint8_t datarx[];
extern CAN_FilterTypeDef filter;
extern CAN_RxHeaderTypeDef RxHeader; //Rx refers as input
```

```
/* USER CODE END EV */

/******************************************************************************/
/*        Cortex-M4 Processor Interruption and Exception Handlers             */
/******************************************************************************/
/**
  * @brief This function handles Non maskable interrupt.
  */
void NMI_Handler(void)
{
  /* USER CODE BEGIN NonMaskableInt_IRQn 0 */

  /* USER CODE END NonMaskableInt_IRQn 0 */
  /* USER CODE BEGIN NonMaskableInt_IRQn 1 */
  while (1)
  {
  }
  /* USER CODE END NonMaskableInt_IRQn 1 */
}

/**
  * @brief This function handles Hard fault interrupt.
  */
void HardFault_Handler(void)
{
  /* USER CODE BEGIN HardFault_IRQn 0 */

  /* USER CODE END HardFault_IRQn 0 */
  while (1)
  {
        /* USER CODE BEGIN W1_HardFault_IRQn 0 */
        /* USER CODE END W1_HardFault_IRQn 0 */
  }
}

/**
  * @brief This function handles Memory management fault.
  */
void MemManage_Handler(void)
{
```

```
  /* USER CODE BEGIN MemoryManagement_IRQn 0 */

  /* USER CODE END MemoryManagement_IRQn 0 */
  while (1)
  {
        /* USER CODE BEGIN W1_MemoryManagement_IRQn 0 */
        /* USER CODE END W1_MemoryManagement_IRQn 0 */
  }
}

/**
  * @brief This function handles Pre-fetch fault, memory access fault.
  */
void BusFault_Handler(void)
{
  /* USER CODE BEGIN BusFault_IRQn 0 */

  /* USER CODE END BusFault_IRQn 0 */
  while (1)
  {
        /* USER CODE BEGIN W1_BusFault_IRQn 0 */
        /* USER CODE END W1_BusFault_IRQn 0 */
  }
}

/**
  * @brief This function handles Undefined instruction or illegal state.
  */
void UsageFault_Handler(void)
{
  /* USER CODE BEGIN UsageFault_IRQn 0 */

  /* USER CODE END UsageFault_IRQn 0 */
  while (1)
  {
        /* USER CODE BEGIN W1_UsageFault_IRQn 0 */
        /* USER CODE END W1_UsageFault_IRQn 0 */
  }
}
```

```
/**
  * @brief This function handles System service call via SWI instruction.
  */
void SVC_Handler(void)
{
  /* USER CODE BEGIN SVCall_IRQn 0 */

  /* USER CODE END SVCall_IRQn 0 */
  /* USER CODE BEGIN SVCall_IRQn 1 */

  /* USER CODE END SVCall_IRQn 1 */
}

/**
  * @brief This function handles Debug monitor.
  */
void DebugMon_Handler(void)
{
  /* USER CODE BEGIN DebugMonitor_IRQn 0 */

  /* USER CODE END DebugMonitor_IRQn 0 */
  /* USER CODE BEGIN DebugMonitor_IRQn 1 */

  /* USER CODE END DebugMonitor_IRQn 1 */
}

/**
  * @brief This function handles Pendable request for system service.
  */
void PendSV_Handler(void)
{
  /* USER CODE BEGIN PendSV_IRQn 0 */

  /* USER CODE END PendSV_IRQn 0 */
  /* USER CODE BEGIN PendSV_IRQn 1 */

  /* USER CODE END PendSV_IRQn 1 */
}

/**
```

```
  * @brief This function handles System tick timer.
  */
void SysTick_Handler(void)
{
  /* USER CODE BEGIN SysTick_IRQn 0 */

  /* USER CODE END SysTick_IRQn 0 */
  HAL_IncTick();
  /* USER CODE BEGIN SysTick_IRQn 1 */

  /* USER CODE END SysTick_IRQn 1 */
}

/******************************************************************************/
/* STM32F4xx Peripheral Interrupt Handlers                                    */
/* Add here the Interrupt Handlers for the used peripherals.                  */
/* For the available peripheral interrupt handler names,                      */
/* please refer to the startup file (startup_stm32f4xx.s).                    */
/******************************************************************************/

/**
  * @brief This function handles CAN1 TX interrupts.
  */
void CAN1_TX_IRQHandler(void)
{
  /* USER CODE BEGIN CAN1_TX_IRQn 0 */

  /* USER CODE END CAN1_TX_IRQn 0 */
  HAL_CAN_IRQHandler(&hcan1);
  /* USER CODE BEGIN CAN1_TX_IRQn 1 */

  /* USER CODE END CAN1_TX_IRQn 1 */
}

/**
  * @brief This function handles CAN1 RX0 interrupts.
  */
void CAN1_RX0_IRQHandler(void)
{
  /* USER CODE BEGIN CAN1_RX0_IRQn 0 */
```

```
  /* USER CODE END CAN1_RX0_IRQn 0 */
  HAL_CAN_IRQHandler(&hcan1);
  /* USER CODE BEGIN CAN1_RX0_IRQn 1 */
  HAL_CAN_GetRxMessage(&hcan1, CAN_FILTER_FIFO0, &RxHeader, datarx);
  printf("%x\n", RxHeader.StdId);
  /* USER CODE END CAN1_RX0_IRQn 1 */
}

/**
  * @brief This function handles CAN1 RX1 interrupt.
  */
void CAN1_RX1_IRQHandler(void)
{
  /* USER CODE BEGIN CAN1_RX1_IRQn 0 */
  printf("here");
  /* USER CODE END CAN1_RX1_IRQn 0 */
  HAL_CAN_IRQHandler(&hcan1);
  /* USER CODE BEGIN CAN1_RX1_IRQn 1 */

  /* USER CODE END CAN1_RX1_IRQn 1 */
}

/* USER CODE BEGIN 1 */

/* USER CODE END 1 */
```

Task 3 with Bonus

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 */
CAN_RxHeaderTypeDef RxHeader; //Rx refers as input
CAN_TxHeaderTypeDef TxHeader;
uint32_t mailbox;
uint8_t datarx [8];
uint8_t data [8];
```

```
CAN_FilterTypeDef filter; //Set filter for receiving specific ID message
int error;
/* USER CODE END PV */

/* 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 */

/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  * @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 */
  filter.FilterActivation = CAN_FILTER_ENABLE; //enabling filter
  filter.FilterBank = 0; //0-13
  filter.SlaveStartFilterBank = 0;//0-13
  filter.FilterScale = CAN_FILTERSCALE_16BIT; //16 or 32 bits
  filter.FilterFIFOAssignment = CAN_FILTER_FIFO0; //FIFO 0 or 1
  filter.FilterMode = CAN_FILTERMODE_IDLIST; // List mode
  filter.FilterIdHigh = 0x690 << 5; // First ID

  TxHeader.StdId = 0x7FF; //maximum 11 bits wide
 TxHeader.DLC = 8; //(Data Length Code) length of the message
 TxHeader.IDE = CAN_ID_STD; //(Identifier Extension) standard
 TxHeader.RTR = CAN_RTR_DATA; //(Remote Transmission Request) type of frame is
Data
  //filter.FilterIdLow = 0x4F0 << 5; // Second ID

  HAL_CAN_Start(&hcan1);
  HAL_CAN_ConfigFilter(&hcan1, &filter);
  HAL_CAN_ActivateNotification(&hcan1, CAN_IT_RX_FIFO0_MSG_PENDING);

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
 if (error != 0) {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET);
 }
  else{
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_RESET);
 }
        /* USER CODE END WHILE */
```

```
        /* USER CODE BEGIN 3 */
  }
  /* 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_SCALE1
);

  /** Initializes the RCC Oscillators according to the specified parameters
  * in the RCC_OscInitTypeDef structure.
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
  RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
  RCC_OscInitStruct.PLL.PLLM = 4;
  RCC_OscInitStruct.PLL.PLLN = 180;
  RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
  RCC_OscInitStruct.PLL.PLLQ = 4;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
        Error_Handler();
  }

  /** Activate the Over-Drive mode
  */
  if (HAL_PWREx_EnableOverDrive() != 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_DIV4;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV4;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) != 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 = 5;
  hcan1.Init.Mode = CAN_MODE_NORMAL;
  hcan1.Init.SyncJumpWidth = CAN_SJW_1TQ;
  hcan1.Init.TimeSeg1 = CAN_BS1_13TQ;
```

```
  hcan1.Init.TimeSeg2 = CAN_BS2_4TQ;
  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 */

  /* USER CODE END CAN1_Init 2 */

}

/**
  * @brief GPIO Initialization Function
  * @param None
  * @retval None
  */
static void MX_GPIO_Init(void)
{
  GPIO_InitTypeDef GPIO_InitStruct = {0};

  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOH_CLK_ENABLE();
  __HAL_RCC_GPIOA_CLK_ENABLE();
  __HAL_RCC_GPIOG_CLK_ENABLE();

  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_RESET);

  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOG, GPIO_PIN_13, GPIO_PIN_RESET);

  /*Configure GPIO pin : PA0 */
  GPIO_InitStruct.Pin = GPIO_PIN_0;
  GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
```

```
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

  /*Configure GPIO pin : PA10 */
  GPIO_InitStruct.Pin = GPIO_PIN_10;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

  /*Configure GPIO pin : PG13 */
  GPIO_InitStruct.Pin = GPIO_PIN_13;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOG, &GPIO_InitStruct);

  /* EXTI interrupt init*/
  HAL_NVIC_SetPriority(EXTI0_IRQn, 0, 0);
  HAL_NVIC_EnableIRQ(EXTI0_IRQn);

}

/* 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
/**
  * @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 */
```

it.c

```
/* USER CODE BEGIN Header */
/**
  ******************************************************************************
  * @file         stm32f4xx_it.c
  * @brief   Interrupt Service Routines.
  ******************************************************************************
  * @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"
#include "stm32f4xx_it.h"
/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
/* USER CODE END Includes */

/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN TD */

/* USER CODE END TD */

/* Private define ------------------------------------------------------------*/
/* USER CODE BEGIN PD */

/* USER CODE END PD */

/* Private macro -------------------------------------------------------------*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables ---------------------------------------------------------*/
/* USER CODE BEGIN PV */
uint8_t message = 0;
/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/* External variables --------------------------------------------------------*/
extern CAN_HandleTypeDef hcan1;
/* USER CODE BEGIN EV */
```

```
extern uint8_t datarx[];
extern CAN_FilterTypeDef filter;
extern CAN_RxHeaderTypeDef RxHeader; //Rx refers as input
extern CAN_TxHeaderTypeDef TxHeader;
extern uint32_t mailbox;
extern uint8_t data [8];
extern int error;
/* USER CODE END EV */

/******************************************************************************/
/*        Cortex-M4 Processor Interruption and Exception Handlers             */
/******************************************************************************/
/**
  * @brief This function handles Non maskable interrupt.
  */
void NMI_Handler(void)
{
  /* USER CODE BEGIN NonMaskableInt_IRQn 0 */

  /* USER CODE END NonMaskableInt_IRQn 0 */
  /* USER CODE BEGIN NonMaskableInt_IRQn 1 */
  while (1)
  {
  }
  /* USER CODE END NonMaskableInt_IRQn 1 */
}

/**
  * @brief This function handles Hard fault interrupt.
  */
void HardFault_Handler(void)
{
  /* USER CODE BEGIN HardFault_IRQn 0 */

  /* USER CODE END HardFault_IRQn 0 */
  while (1)
  {
        /* USER CODE BEGIN W1_HardFault_IRQn 0 */
        /* USER CODE END W1_HardFault_IRQn 0 */
  }
```

```
}

/**
  * @brief This function handles Memory management fault.
  */
void MemManage_Handler(void)
{
  /* USER CODE BEGIN MemoryManagement_IRQn 0 */

  /* USER CODE END MemoryManagement_IRQn 0 */
  while (1)
  {
        /* USER CODE BEGIN W1_MemoryManagement_IRQn 0 */
        /* USER CODE END W1_MemoryManagement_IRQn 0 */
  }
}

/**
  * @brief This function handles Pre-fetch fault, memory access fault.
  */
void BusFault_Handler(void)
{
  /* USER CODE BEGIN BusFault_IRQn 0 */

  /* USER CODE END BusFault_IRQn 0 */
  while (1)
  {
        /* USER CODE BEGIN W1_BusFault_IRQn 0 */
        /* USER CODE END W1_BusFault_IRQn 0 */
  }
}

/**
  * @brief This function handles Undefined instruction or illegal state.
  */
void UsageFault_Handler(void)
{
  /* USER CODE BEGIN UsageFault_IRQn 0 */

  /* USER CODE END UsageFault_IRQn 0 */
```

```
  while (1)
  {
        /* USER CODE BEGIN W1_UsageFault_IRQn 0 */
        /* USER CODE END W1_UsageFault_IRQn 0 */
  }
}

/**
  * @brief This function handles System service call via SWI instruction.
  */
void SVC_Handler(void)
{
  /* USER CODE BEGIN SVCall_IRQn 0 */

  /* USER CODE END SVCall_IRQn 0 */
  /* USER CODE BEGIN SVCall_IRQn 1 */

  /* USER CODE END SVCall_IRQn 1 */
}

/**
  * @brief This function handles Debug monitor.
  */
void DebugMon_Handler(void)
{
  /* USER CODE BEGIN DebugMonitor_IRQn 0 */

  /* USER CODE END DebugMonitor_IRQn 0 */
  /* USER CODE BEGIN DebugMonitor_IRQn 1 */

  /* USER CODE END DebugMonitor_IRQn 1 */
}

/**
  * @brief This function handles Pendable request for system service.
  */
void PendSV_Handler(void)
{
  /* USER CODE BEGIN PendSV_IRQn 0 */
```

```
  /* USER CODE END PendSV_IRQn 0 */
  /* USER CODE BEGIN PendSV_IRQn 1 */

  /* USER CODE END PendSV_IRQn 1 */
}

/**
  * @brief This function handles System tick timer.
  */
void SysTick_Handler(void)
{
  /* USER CODE BEGIN SysTick_IRQn 0 */

  /* USER CODE END SysTick_IRQn 0 */
  HAL_IncTick();
  /* USER CODE BEGIN SysTick_IRQn 1 */

  /* USER CODE END SysTick_IRQn 1 */
}

/******************************************************************************/
/* STM32F4xx Peripheral Interrupt Handlers                                    */
/* Add here the Interrupt Handlers for the used peripherals.                  */
/* For the available peripheral interrupt handler names,                      */
/* please refer to the startup file (startup_stm32f4xx.s).                    */
/******************************************************************************/

/**
  * @brief This function handles EXTI line0 interrupt.
  */
void EXTI0_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI0_IRQn 0 */

  /* USER CODE END EXTI0_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);
  /* USER CODE BEGIN EXTI0_IRQn 1 */
if (message < 8) {
        message ++;
  }
```

```
  else {
        message = 0;
  }
  /* USER CODE END EXTI0_IRQn 1 */
}

/**
  * @brief This function handles CAN1 TX interrupts.
  */
void CAN1_TX_IRQHandler(void)
{
  /* USER CODE BEGIN CAN1_TX_IRQn 0 */

  /* USER CODE END CAN1_TX_IRQn 0 */
  HAL_CAN_IRQHandler(&hcan1);
  /* USER CODE BEGIN CAN1_TX_IRQn 1 */

  /* USER CODE END CAN1_TX_IRQn 1 */
}

/**
  * @brief This function handles CAN1 RX0 interrupts.
  */
void CAN1_RX0_IRQHandler(void)
{
  /* USER CODE BEGIN CAN1_RX0_IRQn 0 */

  /* USER CODE END CAN1_RX0_IRQn 0 */
  HAL_CAN_IRQHandler(&hcan1);
  /* USER CODE BEGIN CAN1_RX0_IRQn 1 */
  HAL_CAN_GetRxMessage(&hcan1, CAN_FILTER_FIFO0, &RxHeader, datarx);
  if (RxHeader.StdId == 0x690){
        if (message == 0) {
        TxHeader.StdId = 0x440;
        data[2] =0x00;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        TxHeader.StdId = 0x316;
        data[2] =0x00;
        data[3] =0x00;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
```

```
        }
        else if (message == 1) {
        TxHeader.StdId = 0x440;
        data[2] =0x1D;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        TxHeader.StdId = 0x316;
        data[2] =0xa0;
        data[3] =0x0f;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        }
        else if (message == 2) {
        TxHeader.StdId = 0x440;
        data[2] =0x3c;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        TxHeader.StdId = 0x316;
        data[2] =0x40;
        data[3] =0x1f;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        }
        else if (message == 3) {
        TxHeader.StdId = 0x440;
        data[2] =0x5b;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        TxHeader.StdId = 0x316;
        data[2] =0xe0;
        data[3] =0x2e;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        }
        else if (message == 4) {
        TxHeader.StdId = 0x440;
        data[2] =0x7b;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        TxHeader.StdId = 0x316;
        data[2] =0x80;
        data[3] =0x3e;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        }
        else if (message == 5) {
        TxHeader.StdId = 0x440;
        data[2] =0x9a;
```

```
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        TxHeader.StdId = 0x316;
        data[2] =0x20;
        data[3] =0x4e;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        }
        else if (message == 6) {
        TxHeader.StdId = 0x440;
        data[2] =0xba;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        TxHeader.StdId = 0x316;
        data[2] =0xc0;
        data[3] =0x5d;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        }
        else if (message == 7) {
        TxHeader.StdId = 0x440;
        data[2] =0xd9;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        TxHeader.StdId = 0x316;
        data[2] =0x60;
        data[3] =0x6d;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        }
        else if (message == 8) {
        TxHeader.StdId = 0x440;
        data[2] =0xfa;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        TxHeader.StdId = 0x316;
        data[2] =0x00;
        data[3] =0x7d;
        error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
        }
  }
  /* USER CODE END CAN1_RX0_IRQn 1 */
}

/**
  * @brief This function handles CAN1 RX1 interrupt.
  */
```

```
void CAN1_RX1_IRQHandler(void)
{
  /* USER CODE BEGIN CAN1_RX1_IRQn 0 */

  /* USER CODE END CAN1_RX1_IRQn 0 */
  HAL_CAN_IRQHandler(&hcan1);
  /* USER CODE BEGIN CAN1_RX1_IRQn 1 */

  /* USER CODE END CAN1_RX1_IRQn 1 */
}

/* USER CODE BEGIN 1 */

/* USER CODE END 1 */
```

.lst Files

Task 1

Main.lst

```
################################################################################
#
# IAR ANSI C/C++ Compiler V9.20.4.327/W64 for ARM         02/Mar/2023  15:20:41
# Copyright 1999-2022 IAR Systems AB.
#
#       Cpu mode             =  thumb
#       Endian          =  little
#       Source file     =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\Core\Src\main.c
#       Command line         =
#       -f
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM\Prelab6\Obj\Application\User\Core\main.o.rsp
#
(S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\Core\Src\main.c
```

```
#        -D USE_HAL_DRIVER -D STM32F429xx -lC
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM\Prelab6\List\Application\User\Core
#        -o
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM\Prelab6\Obj\Application\User\Core
#        --debug --endian=little --cpu=Cortex-M4 -e --fpu=VFPv4_sp
#        --dlib_config S:\School_Work\arm\inc\c\DLib_Config_Full.h -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM/../Core/Inc\
#        -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM/../Drivers/STM32F4xx_HAL_Driver/Inc\
#        -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM/../Drivers/STM32F4xx_HAL_Driver/Inc/Legacy\
#        -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM/../Drivers/CMSIS/Device/ST/STM32F4xx/Include\
#        -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM/../Drivers/CMSIS/Include\
#        -Ohz) --dependencies=n
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM\Prelab6\Obj\Application\User\Core\main.o.d
#        Locale         =  C
#        List file      =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM\Prelab6\List\Application\User\Core\main.lst
#        Object file    =
```

```
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM\Prelab6\Obj\Application\User\Core\main.o
#        Runtime model:
#        __CPP_Runtime   =  1
#        __SystemLibrary =  DLib
#        __dlib_version  =  6
#        __size_limit    =  32768|ARM.EW.LINKER
#
#################################################################################

S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\Core\Src\main.c
        1       /* USER CODE BEGIN Header */
        2       /**
        3       ******************************************************************************
        4       * @file           : main.c
        5       * @brief          : Main program body
        6       ******************************************************************************
        7       * @attention
        8       *
        9       * Copyright (c) 2023 STMicroelectronics.
        10              * All rights reserved.
        11              *
        12              * This software is licensed under terms that can be found in the LICENSE file
        13              * in the root directory of this software component.
        14              * If no LICENSE file comes with this software, it is provided AS-IS.
        15              *
        16      ******************************************************************************
        17              */
        18      /* USER CODE END Header */
        19      /* Includes ------------------------------------------------------------------*/
        20      #include "main.h"
        21
        22      /* Private includes ----------------------------------------------------------*/
        23      /* USER CODE BEGIN Includes */
        24
```

```
        25      /* USER CODE END Includes */
        26
        27      /* Private typedef -----------------------------------------------------------*/
        28      /* USER CODE BEGIN PTD */
        29
        30      /* USER CODE END PTD */
        31
        32      /* Private define ------------------------------------------------------------*/
        33      /* USER CODE BEGIN PD */
        34      /* USER CODE END PD */
        35
        36      /* Private macro -------------------------------------------------------------*/
        37      /* USER CODE BEGIN PM */
        38
        39      /* USER CODE END PM */
        40
        41      /* Private variables ---------------------------------------------------------*/

  \                             In section .data, align 4
        42      CAN_HandleTypeDef hcan1;
  \             hcan1:
  \     0x0   0x0000'0000           DC32 0x0
  \     0x4                   DS8 36
        43
        44      /* USER CODE BEGIN PV */
        45      CAN_TxHeaderTypeDef TxHeader;
  \             TxHeader:
  \     0x28                  DS8 24
        46      uint8_t data []={0x00,0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77}; // 8-byte
data frame
  \             `data`:
  \     0x40    0x00 0x11             DC8 0, 17, 34, 51, 68, 85, 102, 119

  \             0x22 0x33

  \             0x44 0x55

  \             0x66 0x77
        47      uint32_t mailbox; // Actually sending the data
  \             mailbox:
```

```
\      0x48                    DS8 4
       48
       49      /* USER CODE END PV */
       50
       51      /* Private function prototypes -----------------------------------------------*/
       52      void SystemClock_Config(void);
       53      static void MX_GPIO_Init(void);
       54      static void MX_CAN1_Init(void);
       55      /* USER CODE BEGIN PFP */
       56
       57      /* USER CODE END PFP */
       58
       59      /* Private user code ---------------------------------------------------------*/
       60      /* USER CODE BEGIN 0 */
       61
       62      /* USER CODE END 0 */
       63
       64      /**
       65              * @brief  The application entry point.
       66              * @retval int
       67              */

\                              In section .text, align 2, keep-with-next
       68      int main(void)
       69      {
\              main: (+1)
\      0x0   0xB51C          PUSH          {R2-R4,LR}
       70              /* USER CODE BEGIN 1 */
       71
       72              /* USER CODE END 1 */
       73
       74              /* MCU Configuration--------------------------------------------------------*/
       75
       76              /* Reset of all peripherals, Initializes the Flash interface and the
Systick. */
       77              HAL_Init();
\      0x2   0x.... 0x....       BL      HAL_Init
       78
       79              /* USER CODE BEGIN Init */
       80
```

```
        81              /* USER CODE END Init */
        82
        83              /* Configure the system clock */
        84              SystemClock_Config();
\       0x6   0x.... 0x....     BL      SystemClock_Config
        85
        86              /* USER CODE BEGIN SysInit */
        87
        88              /* USER CODE END SysInit */
        89
        90              /* Initialize all configured peripherals */
        91              MX_GPIO_Init();
\       0xA   0x2000            MOVS        R0,#+0
\       0xC   0x9000            STR   R0,[SP, #+0]
        92              MX_CAN1_Init();
\       0xE   0x....            LDR.N       R4,??DataTable1
\       0x10   0x....           LDR.N       R0,??DataTable1_1
\       0x12   0x6801           LDR   R1,[R0, #+0]
\       0x14   0xF041 0x0180    ORR   R1,R1,#0x80
\       0x18   0x6001           STR   R1,[R0, #+0]
\       0x1A   0x6802           LDR   R2,[R0, #+0]
\       0x1C   0xF002 0x0280    AND   R2,R2,#0x80
\       0x20   0x9200           STR   R2,[SP, #+0]
\       0x22   0x2200           MOVS        R2,#+0
\       0x24   0x9900           LDR   R1,[SP, #+0]
\       0x26   0x....           LDR.N       R1,??DataTable1_2
\       0x28   0x9200           STR   R2,[SP, #+0]
\       0x2A   0x2205           MOVS        R2,#+5
\       0x2C   0x6803           LDR   R3,[R0, #+0]
\       0x2E   0xF043 0x0301    ORR   R3,R3,#0x1
\       0x32   0x6003           STR   R3,[R0, #+0]
\       0x34   0x6800           LDR   R0,[R0, #+0]
\       0x36   0xF000 0x0001    AND   R0,R0,#0x1
\       0x3A   0x9000           STR   R0,[SP, #+0]
\       0x3C   0x9800           LDR   R0,[SP, #+0]
\       0x3E   0x6021           STR   R1,[R4, #+0]
\       0x40   0x2000           MOVS        R0,#+0
\       0x42   0x60A0           STR   R0,[R4, #+8]
\       0x44   0x60E0           STR   R0,[R4, #+12]
\       0x46   0xF44F 0x1140    MOV   R1,#+3145728
```

```
\      0x4A   0xF44F 0x2040      MOV   R0,#+786432
\      0x4E   0x6120        STR   R0,[R4, #+16]
\      0x50   0x6062        STR   R2,[R4, #+4]
\      0x52   0x2000        MOVS        R0,#+0
\      0x54   0x61A0        STR   R0,[R4, #+24]
\      0x56   0x83A0        STRH        R0,[R4, #+28]
\      0x58   0x6161        STR   R1,[R4, #+20]
\      0x5A   0x4620        MOV   R0,R4
\      0x5C   0x.... 0x....     BL      HAL_CAN_Init
\      0x60   0xB108        CBZ.N       R0,??main_0
\      0x62   0x.... 0x....     BL      Error_Handler
       93               /* USER CODE BEGIN 2 */
       94               TxHeader.StdId = 0x7FF; //maximum 11 bits wide
\              ??main_0: (+1)
\      0x66   0xF240 0x70FF      MOVW        R0,#+2047
\      0x6A   0x62A0                STR   R0,[R4, #+40]
       95               TxHeader.DLC = 8; //(Data Length Code) length of the message
\      0x6C   0x2108        MOVS        R1,#+8
       96               TxHeader.IDE = CAN_ID_STD; //(Identifier Extension) standard
\      0x6E   0x2000        MOVS        R0,#+0
\      0x70   0x6320        STR   R0,[R4, #+48]
       97               TxHeader.RTR = CAN_RTR_DATA; //(Remote Transmission
Request) type of frame is Data
\      0x72   0x6360        STR   R0,[R4, #+52]
\      0x74   0x63A1        STR   R1,[R4, #+56]
       98               HAL_CAN_Start(&hcan1);
\      0x76   0x4620        MOV   R0,R4
\      0x78   0x.... 0x....     BL      HAL_CAN_Start
       99               HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
\      0x7C   0x.... 0x....     BL      ?Subroutine0
       100              /* USER CODE END 2 */
       101
       102              /* Infinite loop */
       103              /* USER CODE BEGIN WHILE */
       104              while (1)
       105              {
       106              HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data, &mailbox);
// send the data in a loop
\              ??CrossCallReturnLabel_1: (+1)
\      0x80   0x.... 0x....     BL      ?Subroutine0
```

```
      107             HAL_Delay(1);
\             ??CrossCallReturnLabel_0: (+1)
\     0x84   0x2001         MOVS         R0,#+1
\     0x86   0x.... 0x....  BL      HAL_Delay
\     0x8A   0xE7F9                B.N     ??CrossCallReturnLabel_1
      108             /* USER CODE END WHILE */
      109
      110             /* USER CODE BEGIN 3 */
      111             }
      112             /* USER CODE END 3 */
      113             }

\                             In section .text, align 2, keep-with-next
\             ?Subroutine0: (+1)
\     0x0   0xF104 0x0348         ADD   R3,R4,#+72
\     0x4   0xF104 0x0240         ADD   R2,R4,#+64
\     0x8   0xF104 0x0128         ADD   R1,R4,#+40
\     0xC   0x4620          MOV   R0,R4
\     0xE   0x.... 0x....   B.W   HAL_CAN_AddTxMessage
      114
      115             /**
      116             * @brief System Clock Configuration
      117             * @retval None
      118             */

\                             In section .text, align 2, keep-with-next
      119             void SystemClock_Config(void)
      120             {
\             SystemClock_Config: (+1)
\     0x0   0xB580          PUSH         {R7,LR}
\     0x2   0xB092          SUB   SP,SP,#+72
\     0x4   0x2230          MOVS         R2,#+48
\     0x6   0x2100          MOVS         R1,#+0
\     0x8   0xA806          ADD   R0,SP,#+24
\     0xA   0x.... 0x....   BL      memset
\     0xE   0x2214          MOVS         R2,#+20
\     0x10   0x2100         MOVS         R1,#+0
\     0x12   0xA801         ADD   R0,SP,#+4
\     0x14   0x.... 0x....  BL      memset
      121             RCC_OscInitTypeDef RCC_OscInitStruct = {0};
```

```
        122             RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
        123
        124             /** Configure the main internal regulator output voltage
        125             */
        126             __HAL_RCC_PWR_CLK_ENABLE();
  \     0x18   0x2000         MOVS         R0,#+0
  \     0x1A   0x9000         STR    R0,[SP, #+0]
        127
__HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1
);
        128
        129             /** Initializes the RCC Oscillators according to the specified
parameters
        130             * in the RCC_OscInitTypeDef structure.
        131             */
        132             RCC_OscInitStruct.OscillatorType =
RCC_OSCILLATORTYPE_HSE;
        133             RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  \     0x1C   0xF44F 0x3380       MOV   R3,#+65536
  \     0x20   0x....              LDR.N         R0,??DataTable1_3
  \     0x22   0x6801         LDR    R1,[R0, #+0]
  \     0x24   0xF041 0x5180       ORR   R1,R1,#0x10000000
  \     0x28   0x6001         STR    R1,[R0, #+0]
  \     0x2A   0x2100         MOVS         R1,#+0
  \     0x2C   0x6800         LDR    R0,[R0, #+0]
  \     0x2E   0xF000 0x5080       AND   R0,R0,#0x10000000
  \     0x32   0x9000         STR    R0,[SP, #+0]
  \     0x34   0x9800         LDR    R0,[SP, #+0]
  \     0x36   0x....              LDR.N         R0,??DataTable1_4
  \     0x38   0x9100         STR    R1,[SP, #+0]
  \     0x3A   0x6802         LDR    R2,[R0, #+0]
  \     0x3C   0xF442 0x4240       ORR   R2,R2,#0xC000
  \     0x40   0x6002         STR    R2,[R0, #+0]
  \     0x42   0x2201         MOVS         R2,#+1
  \     0x44   0x6800         LDR    R0,[R0, #+0]
  \     0x46   0xF400 0x4040       AND   R0,R0,#0xC000
  \     0x4A   0x9000         STR    R0,[SP, #+0]
        134             RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
        135             RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
        136             RCC_OscInitStruct.PLL.PLLM = 4;
```

```
    137             RCC_OscInitStruct.PLL.PLLN = 180;
    138             RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
    139             RCC_OscInitStruct.PLL.PLLQ = 4;
    140             if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
\       0x4C  0xA806            ADD   R0,SP,#+24
\       0x4E  0x9900      LDR   R1,[SP, #+0]
\       0x50  0x9206      STR   R2,[SP, #+24]
\       0x52  0x2102      MOVS        R1,#+2
\       0x54  0xF44F 0x0280     MOV   R2,#+4194304
\       0x58  0x910C      STR   R1,[SP, #+48]
\       0x5A  0x920D            STR   R2,[SP, #+52]
\       0x5C  0x2104      MOVS        R1,#+4
\       0x5E  0x22B4            MOVS        R2,#+180
\       0x60  0x910E      STR   R1,[SP, #+56]
\       0x62  0x920F      STR   R2,[SP, #+60]
\       0x64  0x2102      MOVS        R1,#+2
\       0x66  0x2204      MOVS        R2,#+4
\       0x68  0x9307      STR   R3,[SP, #+28]
\       0x6A  0x9110      STR   R1,[SP, #+64]
\       0x6C  0x9211      STR   R2,[SP, #+68]
\       0x6E  0x.... 0x....     BL    HAL_RCC_OscConfig
\       0x72  0xB108      CBZ.N       R0,??SystemClock_Config_0
    141             {
    142             Error_Handler();
\       0x74  0x.... 0x....     BL    Error_Handler
    143             }
    144
    145             /** Activate the Over-Drive mode
    146             */
    147             if (HAL_PWREx_EnableOverDrive() != HAL_OK)
\           ??SystemClock_Config_0: (+1)
\       0x78  0x.... 0x....     BL    HAL_PWREx_EnableOverDrive
\       0x7C  0xB108            CBZ.N       R0,??SystemClock_Config_1
    148             {
    149             Error_Handler();
\       0x7E  0xB672            CPSID       I
\           ??SystemClock_Config_2: (+1)
\       0x80  0xE7FE            B.N   ??SystemClock_Config_2
    150             }
    151
```

```
        152             /** Initializes the CPU, AHB and APB buses clocks
        153             */
        154             RCC_ClkInitStruct.ClockType =
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
        155
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
        156             RCC_ClkInitStruct.SYSCLKSource =
RCC_SYSCLKSOURCE_PLLCLK;
  \             ??SystemClock_Config_1: (+1)
  \     0x82  0x2102        MOVS        R1,#+2
  \     0x84  0x9102        STR   R1,[SP, #+8]
  \     0x86  0x200F        MOVS        R0,#+15
        157             RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
        158             RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
  \     0x88  0xF44F 0x51A0       MOV   R1,#+5120
  \     0x8C  0x9001        STR   R0,[SP, #+4]
  \     0x8E  0x2200        MOVS        R2,#+0
  \     0x90  0x9104        STR   R1,[SP, #+16]
        159             RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV4;
  \     0x92  0x9105        STR   R1,[SP, #+20]
  \     0x94  0x9203        STR   R2,[SP, #+12]
        160
        161             if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct,
FLASH_LATENCY_5) != HAL_OK)
  \     0x96  0x2105        MOVS        R1,#+5
  \     0x98  0xA801        ADD   R0,SP,#+4
  \     0x9A  0x.... 0x....     BL      HAL_RCC_ClockConfig
  \     0x9E  0xB108                CBZ.N         R0,??SystemClock_Config_3
        162             {
        163             Error_Handler();
  \     0xA0  0xB672                CPSID         I
  \             ??SystemClock_Config_4: (+1)
  \     0xA2  0xE7FE                B.N     ??SystemClock_Config_4
        164             }
        165             }
  \             ??SystemClock_Config_3: (+1)
  \     0xA4  0xB013                ADD   SP,SP,#+76
  \     0xA6  0xBD00                POP   {PC}
        166
        167             /**
```

```
168         * @brief CAN1 Initialization Function
169         * @param None
170         * @retval None
171         */
172         static void MX_CAN1_Init(void)
173         {
174
175         /* USER CODE BEGIN CAN1_Init 0 */
176
177         /* USER CODE END CAN1_Init 0 */
178
179         /* USER CODE BEGIN CAN1_Init 1 */
180
181         /* USER CODE END CAN1_Init 1 */
182         hcan1.Instance = CAN1;
183         hcan1.Init.Prescaler = 5;
184         hcan1.Init.Mode = CAN_MODE_NORMAL;
185         hcan1.Init.SyncJumpWidth = CAN_SJW_1TQ;
186         hcan1.Init.TimeSeg1 = CAN_BS1_13TQ;
187         hcan1.Init.TimeSeg2 = CAN_BS2_4TQ;
188         hcan1.Init.TimeTriggeredMode = DISABLE;
189         hcan1.Init.AutoBusOff = DISABLE;
190         hcan1.Init.AutoWakeUp = DISABLE;
191         hcan1.Init.AutoRetransmission = DISABLE;
192         hcan1.Init.ReceiveFifoLocked = DISABLE;
193         hcan1.Init.TransmitFifoPriority = DISABLE;
194         if (HAL_CAN_Init(&hcan1) != HAL_OK)
195         {
196         Error_Handler();
197         }
198         /* USER CODE BEGIN CAN1_Init 2 */
199
200         /* USER CODE END CAN1_Init 2 */
201
202         }
203
204         /**
205         * @brief GPIO Initialization Function
206         * @param None
207         * @retval None
```

```
        208             */
        209             static void MX_GPIO_Init(void)
        210             {
        211
        212             /* GPIO Ports Clock Enable */
        213             __HAL_RCC_GPIOH_CLK_ENABLE();
        214             __HAL_RCC_GPIOA_CLK_ENABLE();
        215
        216             }
        217
        218             /* USER CODE BEGIN 4 */
        219
        220             /* USER CODE END 4 */
        221
        222             /**
        223             * @brief  This function is executed in case of error occurrence.
        224             * @retval None
        225             */

  \                             In section .text, align 2, keep-with-next
        226             void Error_Handler(void)
        227             {
        228             /* USER CODE BEGIN Error_Handler_Debug */
        229             /* User can add his own implementation to report the HAL error
return state */
        230             __disable_irq();
  \             Error_Handler: (+1)
  \     0x0   0xB672          CPSID         I
        231             while (1)
  \             ??Error_Handler_0: (+1)
  \     0x2   0xE7FE          B.N     ??Error_Handler_0
        232             {
        233             }
        234             /* USER CODE END Error_Handler_Debug */
        235             }

  \                             In section .text, align 4, keep-with-next
  \             ??DataTable1:
  \     0x0   0x....'....     DC32  hcan1
```

```
  \                              In section .text, align 4, keep-with-next
  \              ??DataTable1_1:
  \      0x0   0x4002'3830            DC32  0x40023830

  \                              In section .text, align 4, keep-with-next
  \              ??DataTable1_2:
  \      0x0   0x4000'6400            DC32  0x40006400

  \                              In section .text, align 4, keep-with-next
  \              ??DataTable1_3:
  \      0x0   0x4002'3840            DC32  0x40023840

  \                              In section .text, align 4, keep-with-next
  \              ??DataTable1_4:
  \      0x0   0x4000'7000            DC32  0x40007000
        236
        237             #ifdef  USE_FULL_ASSERT
        238             /**
        239             * @brief  Reports the name of the source file and the source line
number
        240             *         where the assert_param error has occurred.
        241             * @param  file: pointer to the source file name
        242             * @param  line: assert_param error line source number
        243             * @retval None
        244             */
        245             void assert_failed(uint8_t *file, uint32_t line)
        246             {
        247             /* USER CODE BEGIN 6 */
        248             /* User can add his own implementation to report the file name and
line number,
        249             ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line)
*/
        250             /* USER CODE END 6 */
        251             }
        252             #endif /* USE_FULL_ASSERT */

  Maximum stack usage in bytes:

  .cstack Function
  ------- --------
```

```
      0   Error_Handler
      80   SystemClock_Config
      80   -> Error_Handler
      80   -> HAL_PWREx_EnableOverDrive
      80   -> HAL_RCC_ClockConfig
      80   -> HAL_RCC_OscConfig
      80   -> memset
      16   main
      16   -> Error_Handler
      16   -> HAL_CAN_AddTxMessage
      16   -> HAL_CAN_Init
      16   -> HAL_CAN_Start
      16   -> HAL_Delay
      16   -> HAL_Init
      16   -> SystemClock_Config


 Section sizes:

 Bytes  Function/Label
 -----  --------------
       4  ??DataTable1
       4  ??DataTable1_1
       4  ??DataTable1_2
       4  ??DataTable1_3
       4  ??DataTable1_4
       18  ?Subroutine0
       4  Error_Handler
       168  SystemClock_Config
       76  hcan1
       TxHeader
       data
       mailbox
       140  main


  76 bytes in section .data
 350 bytes in section .text

 350 bytes of CODE memory
```

76 bytes of DATA memory

Errors: none
Warnings: none

It.lst

```
#############################################################################
#########
#
# IAR ANSI C/C++ Compiler V9.20.4.327/W64 for ARM         02/Mar/2023  15:20:41
# Copyright 1999-2022 IAR Systems AB.
#
#       Cpu mode             =  thumb
#       Endian          =  little
#       Source file     =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\Core\Src\st
m32f4xx_it.c
#       Command line         =
#       -f
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM\Pr
elab6\Obj\Application\User\Core\stm32f4xx_it.o.rsp
#
(S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\Core\Src\s
tm32f4xx_it.c
#       -D USE_HAL_DRIVER -D STM32F429xx -lC
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM\Pr
elab6\List\Application\User\Core
#       -o
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM\Pr
elab6\Obj\Application\User\Core
#       --debug --endian=little --cpu=Cortex-M4 -e --fpu=VFPv4_sp
#       --dlib_config S:\School_Work\arm\inc\c\DLib_Config_Full.h -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM/../
Core/Inc\
```

```
#       -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM/../
Drivers/STM32F4xx_HAL_Driver/Inc\
#       -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM/../
Drivers/STM32F4xx_HAL_Driver/Inc/Legacy\
#       -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM/../
Drivers/CMSIS/Device/ST/STM32F4xx/Include\
#       -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM/../
Drivers/CMSIS/Include\
#       -Ohz) --dependencies=n
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM\Pr
elab6\Obj\Application\User\Core\stm32f4xx_it.o.d
#       Locale          =  C
#       List file       =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM\Pr
elab6\List\Application\User\Core\stm32f4xx_it.lst
#       Object file     =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\EWARM\Pr
elab6\Obj\Application\User\Core\stm32f4xx_it.o
#       Runtime model:
#       __CPP_Runtime   =  1
#       __SystemLibrary =  DLib
#       __dlib_version  =  6
#       __size_limit    =  32768|ARM.EW.LINKER
#
################################################################################
#########

S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Prelab6\Core\Src\st
m32f4xx_it.c
```

```
1       /* USER CODE BEGIN Header */
2       /**
3       ******************************************************************************
4       * @filestm32f4xx_it.c
5       * @brief   Interrupt Service Routines.
6       ******************************************************************************
7       * @attention
8       *
9       * Copyright (c) 2023 STMicroelectronics.
10              * All rights reserved.
11              *
12              * This software is licensed under terms that can be found in the
LICENSE file
13              * in the root directory of this software component.
14              * If no LICENSE file comes with this software, it is provided AS-IS.
15              *
16
******************************************************************************
17              */
18      /* USER CODE END Header */
19
20      /* Includes ------------------------------------------------------------*/
21      #include "main.h"
22      #include "stm32f4xx_it.h"
23      /* Private includes ----------------------------------------------------------*/
24      /* USER CODE BEGIN Includes */
25      /* USER CODE END Includes */
26
27      /* Private typedef -----------------------------------------------------------*/
28      /* USER CODE BEGIN TD */
29
30      /* USER CODE END TD */
31
32      /* Private define ------------------------------------------------------------*/
33      /* USER CODE BEGIN PD */
34
35      /* USER CODE END PD */
36
37      /* Private macro -------------------------------------------------------------*/
38      /* USER CODE BEGIN PM */
```

```
39
40      /* USER CODE END PM */
41
42      /* Private variables ---------------------------------------------------------*/
43      /* USER CODE BEGIN PV */
44
45      /* USER CODE END PV */
46
47      /* Private function prototypes -----------------------------------------------*/
48      /* USER CODE BEGIN PFP */
49
50      /* USER CODE END PFP */
51
52      /* Private user code ---------------------------------------------------------*/
53      /* USER CODE BEGIN 0 */
54
55      /* USER CODE END 0 */
56
57      /* External variables --------------------------------------------------------*/
58      extern CAN_HandleTypeDef hcan1;
59      /* USER CODE BEGIN EV */
60
61      /* USER CODE END EV */
62
63      /******************************************************************************/
64      /*       Cortex-M4 Processor Interruption and Exception Handlers
*/
65      /******************************************************************************/
66      /**
67              * @brief This function handles Non maskable interrupt.
68              */

  \                     In section .text, align 2, keep-with-next
69      void NMI_Handler(void)
70      {
71              /* USER CODE BEGIN NonMaskableInt_IRQn 0 */
72
73              /* USER CODE END NonMaskableInt_IRQn 0 */
74              /* USER CODE BEGIN NonMaskableInt_IRQn 1 */
75              while (1)
```

```
\             NMI_Handler: (+1)
\             ??NMI_Handler_0: (+1)
\     0x0  0xE7FE         B.N    ??NMI_Handler_0
      76             {
      77             }
      78             /* USER CODE END NonMaskableInt_IRQn 1 */
      79     }
      80
      81     /**
      82             * @brief This function handles Hard fault interrupt.
      83             */

\                            In section .text, align 2, keep-with-next
      84     void HardFault_Handler(void)
      85     {
      86             /* USER CODE BEGIN HardFault_IRQn 0 */
      87
      88             /* USER CODE END HardFault_IRQn 0 */
      89             while (1)
\             HardFault_Handler: (+1)
\             ??HardFault_Handler_0: (+1)
\     0x0  0xE7FE         B.N    ??HardFault_Handler_0
      90             {
      91             /* USER CODE BEGIN W1_HardFault_IRQn 0 */
      92             /* USER CODE END W1_HardFault_IRQn 0 */
      93             }
      94     }
      95
      96     /**
      97             * @brief This function handles Memory management fault.
      98             */

\                            In section .text, align 2, keep-with-next
      99     void MemManage_Handler(void)
      100            {
      101            /* USER CODE BEGIN MemoryManagement_IRQn 0 */
      102
      103            /* USER CODE END MemoryManagement_IRQn 0 */
      104            while (1)
\             MemManage_Handler: (+1)
```

```
\             ??MemManage_Handler_0: (+1)
\     0x0  0xE7FE         B.N     ??MemManage_Handler_0
      105           {
      106           /* USER CODE BEGIN W1_MemoryManagement_IRQn 0 */
      107           /* USER CODE END W1_MemoryManagement_IRQn 0 */
      108           }
      109           }
      110
      111           /**
      112           * @brief This function handles Pre-fetch fault, memory access fault.
      113           */

\                           In section .text, align 2, keep-with-next
      114           void BusFault_Handler(void)
      115           {
      116           /* USER CODE BEGIN BusFault_IRQn 0 */
      117
      118           /* USER CODE END BusFault_IRQn 0 */
      119           while (1)
\             BusFault_Handler: (+1)
\             ??BusFault_Handler_0: (+1)
\     0x0  0xE7FE         B.N     ??BusFault_Handler_0
      120           {
      121           /* USER CODE BEGIN W1_BusFault_IRQn 0 */
      122           /* USER CODE END W1_BusFault_IRQn 0 */
      123           }
      124           }
      125
      126           /**
      127           * @brief This function handles Undefined instruction or illegal state.
      128           */

\                           In section .text, align 2, keep-with-next
      129           void UsageFault_Handler(void)
      130           {
      131           /* USER CODE BEGIN UsageFault_IRQn 0 */
      132
      133           /* USER CODE END UsageFault_IRQn 0 */
      134           while (1)
\             UsageFault_Handler: (+1)
```

```
\              ??UsageFault_Handler_0: (+1)
\      0x0  0xE7FE          B.N     ??UsageFault_Handler_0
       135            {
       136            /* USER CODE BEGIN W1_UsageFault_IRQn 0 */
       137            /* USER CODE END W1_UsageFault_IRQn 0 */
       138            }
       139            }
       140
       141            /**
       142            * @brief This function handles System service call via SWI
instruction.
       143            */

\                             In section .text, align 2, keep-with-next
       144            void SVC_Handler(void)
       145            {
       146            /* USER CODE BEGIN SVCall_IRQn 0 */
       147
       148            /* USER CODE END SVCall_IRQn 0 */
       149            /* USER CODE BEGIN SVCall_IRQn 1 */
       150
       151            /* USER CODE END SVCall_IRQn 1 */
       152            }
\              SVC_Handler: (+1)
\      0x0  0x4770          BX      LR
       153
       154            /**
       155            * @brief This function handles Debug monitor.
       156            */

\                             In section .text, align 2, keep-with-next
       157            void DebugMon_Handler(void)
       158            {
       159            /* USER CODE BEGIN DebugMonitor_IRQn 0 */
       160
       161            /* USER CODE END DebugMonitor_IRQn 0 */
       162            /* USER CODE BEGIN DebugMonitor_IRQn 1 */
       163
       164            /* USER CODE END DebugMonitor_IRQn 1 */
       165            }
```

```
\              DebugMon_Handler: (+1)
\      0x0  0x4770          BX      LR
       166
       167            /**
       168            * @brief This function handles Pendable request for system
service.
       169            */

\                             In section .text, align 2, keep-with-next
       170            void PendSV_Handler(void)
       171            {
       172            /* USER CODE BEGIN PendSV_IRQn 0 */
       173
       174            /* USER CODE END PendSV_IRQn 0 */
       175            /* USER CODE BEGIN PendSV_IRQn 1 */
       176
       177            /* USER CODE END PendSV_IRQn 1 */
       178            }
\              PendSV_Handler: (+1)
\      0x0  0x4770          BX      LR
       179
       180            /**
       181            * @brief This function handles System tick timer.
       182            */

\                             In section .text, align 2, keep-with-next
       183            void SysTick_Handler(void)
       184            {
       185            /* USER CODE BEGIN SysTick_IRQn 0 */
       186
       187            /* USER CODE END SysTick_IRQn 0 */
       188            HAL_IncTick();
\              SysTick_Handler: (+1)
\      0x0  0x.... 0x....       B.W    HAL_IncTick
       189            /* USER CODE BEGIN SysTick_IRQn 1 */
       190
       191            /* USER CODE END SysTick_IRQn 1 */
       192            }
       193
```

```
        194
/******************************************************************************/
        195             /* STM32F4xx Peripheral Interrupt Handlers
*/
        196             /* Add here the Interrupt Handlers for the used peripherals.
*/
        197             /* For the available peripheral interrupt handler names,
*/
        198             /* please refer to the startup file (startup_stm32f4xx.s).
*/
        199
/******************************************************************************/
        200
        201             /**
        202             * @brief This function handles CAN1 TX interrupts.
        203             */

  \                             In section .text, align 2, keep-with-next
        204             void CAN1_TX_IRQHandler(void)
        205             {
  \             CAN1_TX_IRQHandler: (+1)
  \      0x0   0xB510           PUSH          {R4,LR}
        206             /* USER CODE BEGIN CAN1_TX_IRQn 0 */
        207
        208             /* USER CODE END CAN1_TX_IRQn 0 */
        209             HAL_CAN_IRQHandler(&hcan1);
  \      0x2   0x....           LDR.N         R4,??DataTable2
  \      0x4   0x4620           MOV   R0,R4
  \      0x6   0x.... 0x....    BL      HAL_CAN_IRQHandler
        210             /* USER CODE BEGIN CAN1_TX_IRQn 1 */
        211             HAL_CAN_TxMailbox0CompleteCallback(&hcan1);
  \      0xA   0x4620           MOV   R0,R4
  \      0xC   0x.... 0x....    BL      HAL_CAN_TxMailbox0CompleteCallback
        212             HAL_Delay(50);
  \      0x10   0xE8BD 0x4010   POP   {R4,LR}
  \      0x14   0x2032          MOVS          R0,#+50
  \      0x16   0x.... 0x....   B.W    HAL_Delay
        213             /* USER CODE END CAN1_TX_IRQn 1 */
        214             }
        215
```

```
    216           /**
    217           * @brief This function handles CAN1 RX0 interrupts.
    218           */

\                         In section .text, align 2, keep-with-next
    219           void CAN1_RX0_IRQHandler(void)
    220           {
    221           /* USER CODE BEGIN CAN1_RX0_IRQn 0 */
    222
    223           /* USER CODE END CAN1_RX0_IRQn 0 */
    224           HAL_CAN_IRQHandler(&hcan1);
\           CAN1_RX0_IRQHandler: (+1)
\     0x0  0x....           B.N     ?Subroutine0
    225           /* USER CODE BEGIN CAN1_RX0_IRQn 1 */
    226
    227           /* USER CODE END CAN1_RX0_IRQn 1 */
    228           }

\                         In section .text, align 2, keep-with-next
\           ?Subroutine0: (+1)
\     0x0  0x....           LDR.N          R0,??DataTable2
\     0x2  0x.... 0x....    B.W    HAL_CAN_IRQHandler
    229
    230           /**
    231           * @brief This function handles CAN1 RX1 interrupt.
    232           */

\                         In section .text, align 4, keep-with-next
    233           void CAN1_RX1_IRQHandler(void)
    234           {
    235           /* USER CODE BEGIN CAN1_RX1_IRQn 0 */
    236
    237           /* USER CODE END CAN1_RX1_IRQn 0 */
    238           HAL_CAN_IRQHandler(&hcan1);
\           CAN1_RX1_IRQHandler: (+1)
\     0x0  0xBF00           Nop
\     0x2                   REQUIRE ?Subroutine0
\     0x2                   ;; // Fall through to label ?Subroutine0
    239           /* USER CODE BEGIN CAN1_RX1_IRQn 1 */
    240
```

```
      241             /* USER CODE END CAN1_RX1_IRQn 1 */
      242             }

\                               In section .text, align 4, keep-with-next
\             ??DataTable2:
\     0x0   0x....'....         DC32  hcan1
      243
      244             /* USER CODE BEGIN 1 */
      245
      246             /* USER CODE END 1 */
```

Maximum stack usage in bytes:

```
.cstack Function
------- --------
      0   BusFault_Handler
      0   CAN1_RX0_IRQHandler
      0   -> HAL_CAN_IRQHandler
      0   CAN1_RX1_IRQHandler
      0   -> HAL_CAN_IRQHandler
      8   CAN1_TX_IRQHandler
      8   -> HAL_CAN_IRQHandler
      8   -> HAL_CAN_TxMailbox0CompleteCallback
      0   -> HAL_Delay
      0   DebugMon_Handler
      0   HardFault_Handler
      0   MemManage_Handler
      0   NMI_Handler
      0   PendSV_Handler
      0   SVC_Handler
      0   SysTick_Handler
      0   -> HAL_IncTick
      0   UsageFault_Handler
```

Section sizes:

```
Bytes  Function/Label
-----  --------------
      4  ??DataTable2
```

```
        6  ?Subroutine0
        2  BusFault_Handler
        2  CAN1_RX0_IRQHandler
        2  CAN1_RX1_IRQHandler
       26  CAN1_TX_IRQHandler
        2  DebugMon_Handler
        2  HardFault_Handler
        2  MemManage_Handler
        2  NMI_Handler
        2  PendSV_Handler
        2  SVC_Handler
        4  SysTick_Handler
        2  UsageFault_Handler


 60 bytes in section .text

 60 bytes of CODE memory

Errors: none
Warnings: none
```

Task 2

Main.lst

```
###############################################################################
#########
#
# IAR ANSI C/C++ Compiler V9.20.4.327/W64 for ARM         02/Mar/2023  15:21:44
# Copyright 1999-2022 IAR Systems AB.
#
#        Cpu mode             =  thumb
#        Endian          =  little
#        Source file     =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\Core\Src\mai
n.c
#        Command line         =
#        -f
```

```
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM\Tas
k2\Obj\Application\User\Core\main.o.rsp
#
(S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\Core\Src\ma
in.c
#       -D USE_HAL_DRIVER -D STM32F429xx -lC
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM\Tas
k2\List\Application\User\Core
#       -o
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM\Tas
k2\Obj\Application\User\Core
#       --debug --endian=little --cpu=Cortex-M4 -e --fpu=VFPv4_sp
#       --dlib_config S:\School_Work\arm\inc\c\DLib_Config_Full.h -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM/../C
ore/Inc\
#       -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM/../D
rivers/STM32F4xx_HAL_Driver/Inc\
#       -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM/../D
rivers/STM32F4xx_HAL_Driver/Inc/Legacy\
#       -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM/../D
rivers/CMSIS/Device/ST/STM32F4xx/Include\
#       -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM/../D
rivers/CMSIS/Include\
#       -Ohz) --dependencies=n
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM\Tas
k2\Obj\Application\User\Core\main.o.d
#       Locale          =  C
```

```
#       List file       =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM\Task2\List\Application\User\Core\main.lst
#       Object file     =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM\Task2\Obj\Application\User\Core\main.o
#       Runtime model:
#       __CPP_Runtime   =  1
#       __SystemLibrary =  DLib
#       __dlib_version  =  6
#       __size_limit    =  32768|ARM.EW.LINKER
#
###############################################################################

S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\Core\Src\main.c
       1        /* USER CODE BEGIN Header */
       2        /**
       3        ******************************************************************************
       4        * @file           : main.c
       5        * @brief          : Main program body
       6        ******************************************************************************
       7        * @attention
       8        *
       9        * Copyright (c) 2023 STMicroelectronics.
       10               * All rights reserved.
       11               *
       12               * This software is licensed under terms that can be found in the LICENSE file
       13               * in the root directory of this software component.
       14               * If no LICENSE file comes with this software, it is provided AS-IS.
       15               *
       16
******************************************************************************
       17               */
       18       /* USER CODE END Header */
       19       /* Includes ------------------------------------------------------------*/
```

```
      20     #include "main.h"
      21
      22     /* Private includes ----------------------------------------------------------*/
      23     /* USER CODE BEGIN Includes */
      24
      25     /* USER CODE END Includes */
      26
      27     /* Private typedef -----------------------------------------------------------*/
      28     /* USER CODE BEGIN PTD */
      29
      30     /* USER CODE END PTD */
      31
      32     /* Private define ------------------------------------------------------------*/
      33     /* USER CODE BEGIN PD */
      34     /* USER CODE END PD */
      35
      36     /* Private macro -------------------------------------------------------------*/
      37     /* USER CODE BEGIN PM */
      38
      39     /* USER CODE END PM */
      40
      41     /* Private variables ---------------------------------------------------------*/

\                          In section .bss, align 4
      42     CAN_HandleTypeDef hcan1;
\            hcan1:
\      0x0                 DS8 40
      43
      44     /* USER CODE BEGIN PV */
      45     CAN_RxHeaderTypeDef RxHeader; //Rx refers as input
      46     uint8_t datarx [8];
      47     CAN_FilterTypeDef filter; //Set filter for receiving specific ID message
\            filter:
\      0x28                DS8 40

\                          In section .bss, align 4
\            datarx:
\      0x0                 DS8 8

\                          In section .bss, align 4
```

```
        48      CAN_RxHeaderTypeDef RxHeader; //Rx refers as input
  \             RxHeader:
  \     0x0                     DS8 28
        49
        50      /* USER CODE END PV */
        51
        52      /* Private function prototypes -----------------------------------------------*/
        53      void SystemClock_Config(void);
        54      static void MX_GPIO_Init(void);
        55      static void MX_CAN1_Init(void);
        56      /* USER CODE BEGIN PFP */
        57
        58      /* USER CODE END PFP */
        59
        60      /* Private user code ---------------------------------------------------------*/
        61      /* USER CODE BEGIN 0 */
        62
        63      /* USER CODE END 0 */
        64
        65      /**
        66              * @brief  The application entry point.
        67              * @retval int
        68              */

  \                             In section .text, align 2, keep-with-next
        69      int main(void)
        70      {
  \             main: (+1)
  \     0x0   0xB51C          PUSH          {R2-R4,LR}
        71              /* USER CODE BEGIN 1 */
        72
        73              /* USER CODE END 1 */
        74
        75              /* MCU Configuration--------------------------------------------------------*/
        76
        77              /* Reset of all peripherals, Initializes the Flash interface and the
Systick. */
        78              HAL_Init();
  \     0x2   0x.... 0x....       BL      HAL_Init
        79
```

```
        80              /* USER CODE BEGIN Init */
        81
        82              /* USER CODE END Init */
        83
        84              /* Configure the system clock */
        85              SystemClock_Config();
\       0x6   0x.... 0x....     BL      SystemClock_Config
        86
        87              /* USER CODE BEGIN SysInit */
        88
        89              /* USER CODE END SysInit */
        90
        91              /* Initialize all configured peripherals */
        92              MX_GPIO_Init();
\       0xA   0x2000            MOVS        R0,#+0
\       0xC   0x9000            STR   R0,[SP, #+0]
        93              MX_CAN1_Init();
\       0xE   0x....            LDR.N       R4,??DataTable1
\       0x10   0x....           LDR.N       R0,??DataTable1_1
\       0x12   0x6801           LDR   R1,[R0, #+0]
\       0x14   0xF041 0x0180    ORR   R1,R1,#0x80
\       0x18   0x6001           STR   R1,[R0, #+0]
\       0x1A   0x6802           LDR   R2,[R0, #+0]
\       0x1C   0xF002 0x0280    AND   R2,R2,#0x80
\       0x20   0x9200           STR   R2,[SP, #+0]
\       0x22   0x2200           MOVS        R2,#+0
\       0x24   0x9900           LDR   R1,[SP, #+0]
\       0x26   0x....           LDR.N       R1,??DataTable1_2
\       0x28   0x9200           STR   R2,[SP, #+0]
\       0x2A   0x2205           MOVS        R2,#+5
\       0x2C   0x6803           LDR   R3,[R0, #+0]
\       0x2E   0xF043 0x0301    ORR   R3,R3,#0x1
\       0x32   0x6003           STR   R3,[R0, #+0]
\       0x34   0x6800           LDR   R0,[R0, #+0]
\       0x36   0xF000 0x0001    AND   R0,R0,#0x1
\       0x3A   0x9000           STR   R0,[SP, #+0]
\       0x3C   0x9800           LDR   R0,[SP, #+0]
\       0x3E   0x6021           STR   R1,[R4, #+0]
\       0x40   0x2000           MOVS        R0,#+0
\       0x42   0x60A0           STR   R0,[R4, #+8]
```

```
  \      0x44   0x60E0        STR    R0,[R4, #+12]
  \      0x46   0xF44F 0x1140        MOV   R1,#+3145728
  \      0x4A   0xF44F 0x2040        MOV   R0,#+786432
  \      0x4E   0x6120        STR    R0,[R4, #+16]
  \      0x50   0x6062        STR    R2,[R4, #+4]
  \      0x52   0x2000        MOVS          R0,#+0
  \      0x54   0x61A0        STR    R0,[R4, #+24]
  \      0x56   0x83A0        STRH          R0,[R4, #+28]
  \      0x58   0x6161        STR    R1,[R4, #+20]
  \      0x5A   0x4620        MOV   R0,R4
  \      0x5C   0x.... 0x....      BL      HAL_CAN_Init
  \      0x60   0xB108        CBZ.N         R0,??main_0
  \      0x62   0x.... 0x....      BL      Error_Handler
         94               /* USER CODE BEGIN 2 */
         95               filter.FilterActivation = CAN_FILTER_ENABLE; //enabling filter
  \              ??main_0: (+1)
  \      0x66   0x2001        MOVS          R0,#+1
  \      0x68   0x64A0        STR    R0,[R4, #+72]
         96               filter.FilterBank = 0; //0-13
  \      0x6A   0x2100        MOVS          R1,#+0
         97               filter.SlaveStartFilterBank = 0;//0-13
         98               filter.FilterScale = CAN_FILTERSCALE_16BIT; //16 or 32 bits
         99               filter.FilterFIFOAssignment = CAN_FILTER_FIFO0; //FIFO 0 or 1
         100              filter.FilterMode = CAN_FILTERMODE_IDMASK;
         101              filter.FilterIdHigh = 0x600 << 5; // shift 5 left. Because 16-bit scale
filter for the 11-bit CAN standard ID
  \      0x6C   0xF44F 0x4040        MOV   R0,#+49152
  \      0x70   0x62A0        STR    R0,[R4, #+40]
         102              filter.FilterIdLow = 0x600 << 5; // repeat because using 16 bits
  \      0x72   0x62E0        STR    R0,[R4, #+44]
         103              filter.FilterMaskIdHigh = 0x600 << 5; // only looking at 10th and 9th
bit
  \      0x74   0x6320        STR    R0,[R4, #+48]
         104              filter.FilterMaskIdLow = 0x600 << 5; // only looking at 10th and 9th
bit
  \      0x76   0x6360        STR    R0,[R4, #+52]
  \      0x78   0x63E1        STR    R1,[R4, #+60]
  \      0x7A   0x64E1                STR    R1,[R4, #+76]
  \      0x7C   0x6461        STR    R1,[R4, #+68]
  \      0x7E   0x63A1                STR    R1,[R4, #+56]
```

```
\      0x80   0x6421         STR    R1,[R4, #+64]
       105
       106
       107              HAL_CAN_Start(&hcan1);
\      0x82   0x4620         MOV    R0,R4
\      0x84   0x.... 0x....  BL     HAL_CAN_Start
       108              HAL_CAN_ConfigFilter(&hcan1, &filter);
\      0x88   0xF104 0x0128  ADD    R1,R4,#+40
\      0x8C   0x4620         MOV    R0,R4
\      0x8E   0x.... 0x....  BL     HAL_CAN_ConfigFilter
       109              HAL_CAN_ActivateNotification(&hcan1,
CAN_IT_RX_FIFO0_MSG_PENDING);
\      0x92   0x2102         MOVS         R1,#+2
\      0x94   0x4620         MOV    R0,R4
\      0x96   0x.... 0x....  BL     HAL_CAN_ActivateNotification
       110
       111
       112              /* USER CODE END 2 */
       113
       114              /* Infinite loop */
       115              /* USER CODE BEGIN WHILE */
       116              while (1)
\                  ??main_1: (+1)
\      0x9A   0xE7FE              B.N    ??main_1
       117              {
       118              /* USER CODE END WHILE */
       119
       120              /* USER CODE BEGIN 3 */
       121              }
       122              /* USER CODE END 3 */
       123              }
       124
       125              /**
       126              * @brief System Clock Configuration
       127              * @retval None
       128              */

\                               In section .text, align 2, keep-with-next
       129              void SystemClock_Config(void)
       130              {
```

```
  \                 SystemClock_Config: (+1)
  \       0x0   0xB580          PUSH         {R7,LR}
  \       0x2   0xB092          SUB    SP,SP,#+72
  \       0x4   0x2230          MOVS         R2,#+48
  \       0x6   0x2100          MOVS         R1,#+0
  \       0x8   0xA806          ADD    R0,SP,#+24
  \       0xA   0x.... 0x....   BL       memset
  \       0xE   0x2214          MOVS         R2,#+20
  \       0x10  0x2100          MOVS         R1,#+0
  \       0x12  0xA801          ADD    R0,SP,#+4
  \       0x14  0x.... 0x....   BL       memset
          131               RCC_OscInitTypeDef RCC_OscInitStruct = {0};
          132               RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
          133
          134               /** Configure the main internal regulator output voltage
          135               */
          136               __HAL_RCC_PWR_CLK_ENABLE();
  \       0x18  0x2000          MOVS         R0,#+0
  \       0x1A  0x9000          STR    R0,[SP, #+0]
          137
__HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1
);
          138
          139               /** Initializes the RCC Oscillators according to the specified
parameters
          140               * in the RCC_OscInitTypeDef structure.
          141               */
          142               RCC_OscInitStruct.OscillatorType =
RCC_OSCILLATORTYPE_HSE;
          143               RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  \       0x1C  0xF44F 0x3380       MOV    R3,#+65536
  \       0x20  0x....              LDR.N          R0,??DataTable1_3
  \       0x22  0x6801          LDR    R1,[R0, #+0]
  \       0x24  0xF041 0x5180       ORR    R1,R1,#0x10000000
  \       0x28  0x6001          STR    R1,[R0, #+0]
  \       0x2A  0x2100          MOVS         R1,#+0
  \       0x2C  0x6800          LDR    R0,[R0, #+0]
  \       0x2E  0xF000 0x5080       AND    R0,R0,#0x10000000
  \       0x32  0x9000          STR    R0,[SP, #+0]
  \       0x34  0x9800          LDR    R0,[SP, #+0]
```

```
\     0x36   0x....          LDR.N          R0,??DataTable1_4
\     0x38   0x9100          STR    R1,[SP, #+0]
\     0x3A   0x6802          LDR    R2,[R0, #+0]
\     0x3C   0xF442 0x4240   ORR    R2,R2,#0xC000
\     0x40   0x6002          STR    R2,[R0, #+0]
\     0x42   0x2201          MOVS           R2,#+1
\     0x44   0x6800          LDR    R0,[R0, #+0]
\     0x46   0xF400 0x4040   AND    R0,R0,#0xC000
\     0x4A   0x9000          STR    R0,[SP, #+0]
      144              RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
      145              RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
      146              RCC_OscInitStruct.PLL.PLLM = 4;
      147              RCC_OscInitStruct.PLL.PLLN = 180;
      148              RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
      149              RCC_OscInitStruct.PLL.PLLQ = 4;
      150              if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
\     0x4C   0xA806                  ADD    R0,SP,#+24
\     0x4E   0x9900          LDR    R1,[SP, #+0]
\     0x50   0x9206          STR    R2,[SP, #+24]
\     0x52   0x2102          MOVS           R1,#+2
\     0x54   0xF44F 0x0280   MOV    R2,#+4194304
\     0x58   0x910C          STR    R1,[SP, #+48]
\     0x5A   0x920D                  STR    R2,[SP, #+52]
\     0x5C   0x2104          MOVS           R1,#+4
\     0x5E   0x22B4                  MOVS           R2,#+180
\     0x60   0x910E          STR    R1,[SP, #+56]
\     0x62   0x920F          STR    R2,[SP, #+60]
\     0x64   0x2102          MOVS           R1,#+2
\     0x66   0x2204          MOVS           R2,#+4
\     0x68   0x9307          STR    R3,[SP, #+28]
\     0x6A   0x9110          STR    R1,[SP, #+64]
\     0x6C   0x9211          STR    R2,[SP, #+68]
\     0x6E   0x.... 0x....   BL     HAL_RCC_OscConfig
\     0x72   0xB108          CBZ.N          R0,??SystemClock_Config_0
      151              {
      152              Error_Handler();
\     0x74   0x.... 0x....   BL     Error_Handler
      153              }
      154
      155              /** Activate the Over-Drive mode
```

```
        156             */
        157             if (HAL_PWREx_EnableOverDrive() != HAL_OK)
  \             ??SystemClock_Config_0: (+1)
  \     0x78  0x.... 0x....     BL      HAL_PWREx_EnableOverDrive
  \     0x7C  0xB108            CBZ.N   R0,??SystemClock_Config_1
        158             {
        159             Error_Handler();
  \     0x7E  0xB672            CPSID   I
  \             ??SystemClock_Config_2: (+1)
  \     0x80  0xE7FE            B.N     ??SystemClock_Config_2
        160             }
        161
        162             /** Initializes the CPU, AHB and APB buses clocks
        163             */
        164             RCC_ClkInitStruct.ClockType =
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
        165
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
        166             RCC_ClkInitStruct.SYSCLKSource =
RCC_SYSCLKSOURCE_PLLCLK;
  \             ??SystemClock_Config_1: (+1)
  \     0x82  0x2102        MOVS    R1,#+2
  \     0x84  0x9102        STR     R1,[SP, #+8]
  \     0x86  0x200F        MOVS    R0,#+15
        167             RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
        168             RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
  \     0x88  0xF44F 0x51A0     MOV     R1,#+5120
  \     0x8C  0x9001        STR     R0,[SP, #+4]
  \     0x8E  0x2200        MOVS    R2,#+0
  \     0x90  0x9104        STR     R1,[SP, #+16]
        169             RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV4;
  \     0x92  0x9105        STR     R1,[SP, #+20]
  \     0x94  0x9203        STR     R2,[SP, #+12]
        170
        171             if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct,
FLASH_LATENCY_5) != HAL_OK)
  \     0x96  0x2105        MOVS    R1,#+5
  \     0x98  0xA801        ADD     R0,SP,#+4
  \     0x9A  0x.... 0x....     BL      HAL_RCC_ClockConfig
  \     0x9E  0xB108            CBZ.N   R0,??SystemClock_Config_3
```

```
      172           {
      173           Error_Handler();
\     0xA0  0xB672          CPSID         I
\           ??SystemClock_Config_4: (+1)
\     0xA2  0xE7FE          B.N    ??SystemClock_Config_4
      174           }
      175           }
\           ??SystemClock_Config_3: (+1)
\     0xA4  0xB013          ADD    SP,SP,#+76
\     0xA6  0xBD00          POP    {PC}
      176
      177           /**
      178           * @brief CAN1 Initialization Function
      179           * @param None
      180           * @retval None
      181           */
      182           static void MX_CAN1_Init(void)
      183           {
      184
      185           /* USER CODE BEGIN CAN1_Init 0 */
      186
      187           /* USER CODE END CAN1_Init 0 */
      188
      189           /* USER CODE BEGIN CAN1_Init 1 */
      190
      191           /* USER CODE END CAN1_Init 1 */
      192           hcan1.Instance = CAN1;
      193           hcan1.Init.Prescaler = 5;
      194           hcan1.Init.Mode = CAN_MODE_NORMAL;
      195           hcan1.Init.SyncJumpWidth = CAN_SJW_1TQ;
      196           hcan1.Init.TimeSeg1 = CAN_BS1_13TQ;
      197           hcan1.Init.TimeSeg2 = CAN_BS2_4TQ;
      198           hcan1.Init.TimeTriggeredMode = DISABLE;
      199           hcan1.Init.AutoBusOff = DISABLE;
      200           hcan1.Init.AutoWakeUp = DISABLE;
      201           hcan1.Init.AutoRetransmission = DISABLE;
      202           hcan1.Init.ReceiveFifoLocked = DISABLE;
      203           hcan1.Init.TransmitFifoPriority = DISABLE;
      204           if (HAL_CAN_Init(&hcan1) != HAL_OK)
      205           {
```

```
        206             Error_Handler();
        207             }
        208             /* USER CODE BEGIN CAN1_Init 2 */
        209
        210             /* USER CODE END CAN1_Init 2 */
        211
        212             }
        213
        214             /**
        215             * @brief GPIO Initialization Function
        216             * @param None
        217             * @retval None
        218             */
        219             static void MX_GPIO_Init(void)
        220             {
        221
        222             /* GPIO Ports Clock Enable */
        223             __HAL_RCC_GPIOH_CLK_ENABLE();
        224             __HAL_RCC_GPIOA_CLK_ENABLE();
        225
        226             }
        227
        228             /* USER CODE BEGIN 4 */
        229
        230             /* USER CODE END 4 */
        231
        232             /**
        233             * @brief  This function is executed in case of error occurrence.
        234             * @retval None
        235             */

  \                             In section .text, align 2, keep-with-next
        236             void Error_Handler(void)
        237             {
        238             /* USER CODE BEGIN Error_Handler_Debug */
        239             /* User can add his own implementation to report the HAL error
return state */
        240               __disable_irq();
  \               Error_Handler: (+1)
  \       0x0   0xB672          CPSID         I
```

```
        241             while (1)
  \              ??Error_Handler_0: (+1)
  \      0x2  0xE7FE         B.N    ??Error_Handler_0
        242             {
        243             }
        244             /* USER CODE END Error_Handler_Debug */
        245             }

  \                             In section .text, align 4, keep-with-next
  \              ??DataTable1:
  \      0x0  0x....'....         DC32  hcan1

  \                             In section .text, align 4, keep-with-next
  \              ??DataTable1_1:
  \      0x0  0x4002'3830             DC32  0x40023830

  \                             In section .text, align 4, keep-with-next
  \              ??DataTable1_2:
  \      0x0  0x4000'6400             DC32  0x40006400

  \                             In section .text, align 4, keep-with-next
  \              ??DataTable1_3:
  \      0x0  0x4002'3840             DC32  0x40023840

  \                             In section .text, align 4, keep-with-next
  \              ??DataTable1_4:
  \      0x0  0x4000'7000             DC32  0x40007000
        246
        247             #ifdef  USE_FULL_ASSERT
        248             /**
        249             * @brief  Reports the name of the source file and the source line
number
        250             *         where the assert_param error has occurred.
        251             * @param  file: pointer to the source file name
        252             * @param  line: assert_param error line source number
        253             * @retval None
        254             */
        255             void assert_failed(uint8_t *file, uint32_t line)
        256             {
        257             /* USER CODE BEGIN 6 */
```

```
        258             /* User can add his own implementation to report the file name and
line number,
        259             ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line)
*/
        260             /* USER CODE END 6 */
        261             }
        262             #endif /* USE_FULL_ASSERT */
```

Maximum stack usage in bytes:

```
.cstack Function
------- --------
        0   Error_Handler
        80   SystemClock_Config
        80   -> Error_Handler
        80   -> HAL_PWREx_EnableOverDrive
        80   -> HAL_RCC_ClockConfig
        80   -> HAL_RCC_OscConfig
        80   -> memset
        16   main
        16   -> Error_Handler
        16   -> HAL_CAN_ActivateNotification
        16   -> HAL_CAN_ConfigFilter
        16   -> HAL_CAN_Init
        16   -> HAL_CAN_Start
        16   -> HAL_Init
        16   -> SystemClock_Config
```

Section sizes:

```
Bytes  Function/Label
-----  --------------
        4  ??DataTable1
        4  ??DataTable1_1
        4  ??DataTable1_2
        4  ??DataTable1_3
        4  ??DataTable1_4
        4  Error_Handler
        28  RxHeader
```

```
        168  SystemClock_Config
        8  datarx
        80  hcan1
        filter
        156  main


 116 bytes in section .bss
 348 bytes in section .text

 348 bytes of CODE memory
 116 bytes of DATA memory

Errors: none
Warnings: none


lt.lst
#############################################################################
##########
#
# IAR ANSI C/C++ Compiler V9.20.4.327/W64 for ARM        02/Mar/2023  15:21:44
# Copyright 1999-2022 IAR Systems AB.
#
#       Cpu mode             =  thumb
#       Endian               =  little
#       Source file          =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\Core\Src\stm
32f4xx_it.c
#       Command line         =
#       -f
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM\Tas
k2\Obj\Application\User\Core\stm32f4xx_it.o.rsp
#
(S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\Core\Src\st
m32f4xx_it.c
#       -D USE_HAL_DRIVER -D STM32F429xx -lC
```

```
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM\Tas
k2\List\Application\User\Core
#        -o
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM\Tas
k2\Obj\Application\User\Core
#        --debug --endian=little --cpu=Cortex-M4 -e --fpu=VFPv4_sp
#        --dlib_config S:\School_Work\arm\inc\c\DLib_Config_Full.h -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM/../C
ore/Inc\
#        -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM/../D
rivers/STM32F4xx_HAL_Driver/Inc\
#        -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM/../D
rivers/STM32F4xx_HAL_Driver/Inc/Legacy\
#        -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM/../D
rivers/CMSIS/Device/ST/STM32F4xx/Include\
#        -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM/../D
rivers/CMSIS/Include\
#        -Ohz) --dependencies=n
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM\Tas
k2\Obj\Application\User\Core\stm32f4xx_it.o.d
#        Locale                =  C
#        List file             =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM\Tas
k2\List\Application\User\Core\stm32f4xx_it.lst
#        Object file           =
```

```
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\EWARM\Tas
k2\Obj\Application\User\Core\stm32f4xx_it.o
#       Runtime model:
#       __CPP_Runtime              =  1
#       __SystemLibrary     =  DLib
#       __dlib_file_descriptor =  1
#       __dlib_version        =  6
#       __iar_require _Printf
#       __size_limit           =  32768|ARM.EW.LINKER
#
################################################################################
#########

S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task2\Core\Src\stm
32f4xx_it.c
        1       /* USER CODE BEGIN Header */
        2       /**
        3       ******************************************************************************
        4       * @filestm32f4xx_it.c
        5       * @brief   Interrupt Service Routines.
        6       ******************************************************************************
        7       * @attention
        8       *
        9       * Copyright (c) 2023 STMicroelectronics.
        10              * All rights reserved.
        11              *
        12              * This software is licensed under terms that can be found in the
LICENSE file
        13              * in the root directory of this software component.
        14              * If no LICENSE file comes with this software, it is provided AS-IS.
        15              *
        16
******************************************************************************
        17              */
        18      /* USER CODE END Header */
        19
        20      /* Includes ------------------------------------------------------------*/
        21      #include "main.h"
        22      #include "stm32f4xx_it.h"
```

```
23  /* Private includes ----------------------------------------------------------*/
24  /* USER CODE BEGIN Includes */
25  #include "stdio.h"
26  /* USER CODE END Includes */
27
28  /* Private typedef -----------------------------------------------------------*/
29  /* USER CODE BEGIN TD */
30
31  /* USER CODE END TD */
32
33  /* Private define ------------------------------------------------------------*/
34  /* USER CODE BEGIN PD */
35
36  /* USER CODE END PD */
37
38  /* Private macro -------------------------------------------------------------*/
39  /* USER CODE BEGIN PM */
40
41  /* USER CODE END PM */
42
43  /* Private variables ---------------------------------------------------------*/
44  /* USER CODE BEGIN PV */
45
46  /* USER CODE END PV */
47
48  /* Private function prototypes -----------------------------------------------*/
49  /* USER CODE BEGIN PFP */
50
51  /* USER CODE END PFP */
52
53  /* Private user code ---------------------------------------------------------*/
54  /* USER CODE BEGIN 0 */
55
56  /* USER CODE END 0 */
57
58  /* External variables --------------------------------------------------------*/
59  extern CAN_HandleTypeDef hcan1;
60  /* USER CODE BEGIN EV */
61  extern uint8_t datarx[];
62  extern CAN_FilterTypeDef filter;
```

```
        63      extern CAN_RxHeaderTypeDef RxHeader; //Rx refers as input
        64      /* USER CODE END EV */
        65
        66      /******************************************************************************/
        67      /*       Cortex-M4 Processor Interruption and Exception Handlers
*/
        68      /******************************************************************************/
        69      /**
        70            * @brief This function handles Non maskable interrupt.
        71            */

  \                         In section .text, align 2, keep-with-next
        72      void NMI_Handler(void)
        73      {
        74            /* USER CODE BEGIN NonMaskableInt_IRQn 0 */
        75
        76            /* USER CODE END NonMaskableInt_IRQn 0 */
        77            /* USER CODE BEGIN NonMaskableInt_IRQn 1 */
        78            while (1)
  \           NMI_Handler: (+1)
  \           ??NMI_Handler_0: (+1)
  \     0x0   0xE7FE          B.N    ??NMI_Handler_0
        79            {
        80            }
        81            /* USER CODE END NonMaskableInt_IRQn 1 */
        82      }
        83
        84      /**
        85            * @brief This function handles Hard fault interrupt.
        86            */

  \                         In section .text, align 2, keep-with-next
        87      void HardFault_Handler(void)
        88      {
        89            /* USER CODE BEGIN HardFault_IRQn 0 */
        90
        91            /* USER CODE END HardFault_IRQn 0 */
        92            while (1)
  \           HardFault_Handler: (+1)
  \           ??HardFault_Handler_0: (+1)
```

```
\     0x0  0xE7FE        B.N     ??HardFault_Handler_0
      93            {
      94            /* USER CODE BEGIN W1_HardFault_IRQn 0 */
      95            /* USER CODE END W1_HardFault_IRQn 0 */
      96            }
      97      }
      98
      99      /**
      100           * @brief This function handles Memory management fault.
      101           */

\                           In section .text, align 2, keep-with-next
      102           void MemManage_Handler(void)
      103           {
      104           /* USER CODE BEGIN MemoryManagement_IRQn 0 */
      105
      106           /* USER CODE END MemoryManagement_IRQn 0 */
      107           while (1)
\           MemManage_Handler: (+1)
\           ??MemManage_Handler_0: (+1)
\     0x0  0xE7FE        B.N     ??MemManage_Handler_0
      108           {
      109           /* USER CODE BEGIN W1_MemoryManagement_IRQn 0 */
      110           /* USER CODE END W1_MemoryManagement_IRQn 0 */
      111           }
      112           }
      113
      114           /**
      115           * @brief This function handles Pre-fetch fault, memory access fault.
      116           */

\                           In section .text, align 2, keep-with-next
      117           void BusFault_Handler(void)
      118           {
      119           /* USER CODE BEGIN BusFault_IRQn 0 */
      120
      121           /* USER CODE END BusFault_IRQn 0 */
      122           while (1)
\           BusFault_Handler: (+1)
\           ??BusFault_Handler_0: (+1)
```

```
\     0x0  0xE7FE        B.N     ??BusFault_Handler_0
      123          {
      124          /* USER CODE BEGIN W1_BusFault_IRQn 0 */
      125          /* USER CODE END W1_BusFault_IRQn 0 */
      126          }
      127          }
      128
      129          /**
      130          * @brief This function handles Undefined instruction or illegal state.
      131          */

\                          In section .text, align 2, keep-with-next
      132          void UsageFault_Handler(void)
      133          {
      134          /* USER CODE BEGIN UsageFault_IRQn 0 */
      135
      136          /* USER CODE END UsageFault_IRQn 0 */
      137          while (1)
\            UsageFault_Handler: (+1)
\            ??UsageFault_Handler_0: (+1)
\     0x0  0xE7FE        B.N     ??UsageFault_Handler_0
      138          {
      139          /* USER CODE BEGIN W1_UsageFault_IRQn 0 */
      140          /* USER CODE END W1_UsageFault_IRQn 0 */
      141          }
      142          }
      143
      144          /**
      145          * @brief This function handles System service call via SWI
instruction.
      146          */

\                          In section .text, align 2, keep-with-next
      147          void SVC_Handler(void)
      148          {
      149          /* USER CODE BEGIN SVCall_IRQn 0 */
      150
      151          /* USER CODE END SVCall_IRQn 0 */
      152          /* USER CODE BEGIN SVCall_IRQn 1 */
      153
```

```
      154           /* USER CODE END SVCall_IRQn 1 */
      155           }
\           SVC_Handler: (+1)
\     0x0  0x4770         BX      LR
      156
      157           /**
      158           * @brief This function handles Debug monitor.
      159           */

\                         In section .text, align 2, keep-with-next
      160           void DebugMon_Handler(void)
      161           {
      162           /* USER CODE BEGIN DebugMonitor_IRQn 0 */
      163
      164           /* USER CODE END DebugMonitor_IRQn 0 */
      165           /* USER CODE BEGIN DebugMonitor_IRQn 1 */
      166
      167           /* USER CODE END DebugMonitor_IRQn 1 */
      168           }
\           DebugMon_Handler: (+1)
\     0x0  0x4770         BX      LR
      169
      170           /**
      171           * @brief This function handles Pendable request for system
service.
      172           */

\                         In section .text, align 2, keep-with-next
      173           void PendSV_Handler(void)
      174           {
      175           /* USER CODE BEGIN PendSV_IRQn 0 */
      176
      177           /* USER CODE END PendSV_IRQn 0 */
      178           /* USER CODE BEGIN PendSV_IRQn 1 */
      179
      180           /* USER CODE END PendSV_IRQn 1 */
      181           }
\           PendSV_Handler: (+1)
\     0x0  0x4770         BX      LR
      182
```

```
        183             /**
        184             * @brief This function handles System tick timer.
        185             */

  \                             In section .text, align 2, keep-with-next
        186             void SysTick_Handler(void)
        187             {
        188             /* USER CODE BEGIN SysTick_IRQn 0 */
        189
        190             /* USER CODE END SysTick_IRQn 0 */
        191             HAL_IncTick();
  \               SysTick_Handler: (+1)
  \       0x0   0x.... 0x....       B.W    HAL_IncTick
        192             /* USER CODE BEGIN SysTick_IRQn 1 */
        193
        194             /* USER CODE END SysTick_IRQn 1 */
        195             }
        196
        197
/******************************************************************************/
        198             /* STM32F4xx Peripheral Interrupt Handlers
*/
        199             /* Add here the Interrupt Handlers for the used peripherals.
*/
        200             /* For the available peripheral interrupt handler names,
*/
        201             /* please refer to the startup file (startup_stm32f4xx.s).
*/
        202
/******************************************************************************/
        203
        204             /**
        205             * @brief This function handles CAN1 TX interrupts.
        206             */

  \                             In section .text, align 2, keep-with-next
        207             void CAN1_TX_IRQHandler(void)
        208             {
        209             /* USER CODE BEGIN CAN1_TX_IRQn 0 */
        210
```

```
      211           /* USER CODE END CAN1_TX_IRQn 0 */
      212           HAL_CAN_IRQHandler(&hcan1);
\           CAN1_TX_IRQHandler: (+1)
\     0x0  0x....           B.N     ?Subroutine0
      213           /* USER CODE BEGIN CAN1_TX_IRQn 1 */
      214
      215           /* USER CODE END CAN1_TX_IRQn 1 */
      216           }

\                           In section .text, align 2, keep-with-next
\           ?Subroutine0: (+1)
\     0x0  0x....           LDR.N         R0,??DataTable4_1
\     0x2  0x.... 0x....    B.W    HAL_CAN_IRQHandler
      217
      218           /**
      219           * @brief This function handles CAN1 RX0 interrupts.
      220           */

\                           In section .text, align 2, keep-with-next
      221           void CAN1_RX0_IRQHandler(void)
      222           {
\           CAN1_RX0_IRQHandler: (+1)
\     0x0  0xB538           PUSH          {R3-R5,LR}
      223           /* USER CODE BEGIN CAN1_RX0_IRQn 0 */
      224
      225           /* USER CODE END CAN1_RX0_IRQn 0 */
      226           HAL_CAN_IRQHandler(&hcan1);
\     0x2  0x....           LDR.N         R4,??DataTable4_1
\     0x4  0x....           LDR.N         R5,??DataTable4_2
\     0x6  0x4620           MOV   R0,R4
\     0x8  0x.... 0x....    BL      HAL_CAN_IRQHandler
      227           /* USER CODE BEGIN CAN1_RX0_IRQn 1 */
      228           HAL_CAN_GetRxMessage(&hcan1, CAN_FILTER_FIFO0,
&RxHeader, datarx);
\     0xC  0x....           LDR.N         R3,??DataTable4_3
\     0xE  0x462A           MOV   R2,R5
\     0x10  0x2100          MOVS          R1,#+0
\     0x12  0x4620          MOV   R0,R4
\     0x14  0x.... 0x....   BL      HAL_CAN_GetRxMessage
      229           printf("%x\n", RxHeader.StdId);
```

```
\     0x18   0x6829         LDR    R1,[R5, #+0]
\     0x1A   0xE8BD 0x4034  POP    {R2,R4,R5,LR}
\     0x1E   0x....         ADR.N  R0,??DataTable4
\     0x20   0x.... 0x....  B.W    printf
      230            /* USER CODE END CAN1_RX0_IRQn 1 */
      231            }
      232
      233            /**
      234            * @brief This function handles CAN1 RX1 interrupt.
      235            */

\                            In section .text, align 4, keep-with-next
      236            void CAN1_RX1_IRQHandler(void)
      237            {
\           CAN1_RX1_IRQHandler: (+1)
\     0x0   0xB580         PUSH   {R7,LR}
      238            /* USER CODE BEGIN CAN1_RX1_IRQn 0 */
      239            printf("here");
\     0x2   0xBF00         Nop
\     0x4   0x....         ADR.N  R0,?_1
\     0x6   0x.... 0x....  BL     printf
      240            /* USER CODE END CAN1_RX1_IRQn 0 */
      241            HAL_CAN_IRQHandler(&hcan1);
\     0xA   0xE8BD 0x4002  POP    {R1,LR}
\     0xE                  REQUIRE ?Subroutine0
\     0xE                  ;; // Fall through to label ?Subroutine0
      242            /* USER CODE BEGIN CAN1_RX1_IRQn 1 */
      243
      244            /* USER CODE END CAN1_RX1_IRQn 1 */
      245            }

\                            In section .text, align 4, keep-with-next
\           ??DataTable4:
\     0x0   0x25 0x78      DC8    "%x\n"

\           0x0A 0x00

\                            In section .text, align 4, keep-with-next
\           ??DataTable4_1:
\     0x0   0x....'....    DC32   hcan1
```

```
\                               In section .text, align 4, keep-with-next
\               ??DataTable4_2:
\       0x0   0x....'....       DC32  RxHeader

\                               In section .text, align 4, keep-with-next
\               ??DataTable4_3:
\       0x0   0x....'....       DC32  datarx

\                               In section .rodata, align 4, keep-with-next
\               ?_0:
\       0x0   0x25 0x78         DC8 "%x\012"

\               0x0A 0x00

\                               In section .text, align 4, keep-with-next
\               ?_1:
\       0x0   0x68 0x65         DC8 "here"

\               0x72 0x65

\               0x00
\       0x5                     DS8 3
        246
        247             /* USER CODE BEGIN 1 */
        248
        249             /* USER CODE END 1 */
```

Maximum stack usage in bytes:

```
.cstack Function
------- --------
        0   BusFault_Handler
        16   CAN1_RX0_IRQHandler
        16   -> HAL_CAN_GetRxMessage
        16   -> HAL_CAN_IRQHandler
        0   -> printf
        8   CAN1_RX1_IRQHandler
        0   -> HAL_CAN_IRQHandler
        8   -> printf
```

```
        0   CAN1_TX_IRQHandler
        0   -> HAL_CAN_IRQHandler
        0   DebugMon_Handler
        0   HardFault_Handler
        0   MemManage_Handler
        0   NMI_Handler
        0   PendSV_Handler
        0   SVC_Handler
        0   SysTick_Handler
        0   -> HAL_IncTick
        0   UsageFault_Handler


  Section sizes:

  Bytes  Function/Label
  -----  --------------
        4  ??DataTable4
        4  ??DataTable4_1
        4  ??DataTable4_2
        4  ??DataTable4_3
        6  ?Subroutine0
        4  ?_0
        8  ?_1
        2  BusFault_Handler
       36  CAN1_RX0_IRQHandler
       14  CAN1_RX1_IRQHandler
        2  CAN1_TX_IRQHandler
        2  DebugMon_Handler
        2  HardFault_Handler
        2  MemManage_Handler
        2  NMI_Handler
        2  PendSV_Handler
        2  SVC_Handler
        4  SysTick_Handler
        2  UsageFault_Handler


   4 bytes in section .rodata
 102 bytes in section .text
```

102 bytes of CODE memory
  4 bytes of CONST memory

Errors: none
Warnings: none

Task 3

Main.lst

```
###############################################################################
#########
#
# IAR ANSI C/C++ Compiler V9.20.4.327/W64 for ARM         02/Mar/2023  15:31:34
# Copyright 1999-2022 IAR Systems AB.
#
#       Cpu mode             =  thumb
#       Endian          =  little
#       Source file     =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\Core\Src\mai
n.c
#       Command line         =
#       -f
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM\Tas
k3\Obj\Application\User\Core\main.o.rsp
#
(S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\Core\Src\ma
in.c
#       -D USE_HAL_DRIVER -D STM32F429xx -lC
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM\Tas
k3\List\Application\User\Core
#       -o
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM\Tas
k3\Obj\Application\User\Core
#       --debug --endian=little --cpu=Cortex-M4 -e --fpu=VFPv4_sp
#       --dlib_config S:\School_Work\arm\inc\c\DLib_Config_Full.h -I
```

```
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM/../Core/Inc\
#        -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM/../Drivers/STM32F4xx_HAL_Driver/Inc\
#        -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM/../Drivers/STM32F4xx_HAL_Driver/Inc/Legacy\
#        -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM/../Drivers/CMSIS/Device/ST/STM32F4xx/Include\
#        -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM/../Drivers/CMSIS/Include\
#        -Ohz) --dependencies=n
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM\Task3\Obj\Application\User\Core\main.o.d
#        Locale         =  C
#        List file      =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM\Task3\List\Application\User\Core\main.lst
#        Object file    =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM\Task3\Obj\Application\User\Core\main.o
#        Runtime model:
#        __CPP_Runtime   =  1
#        __SystemLibrary =  DLib
#        __dlib_version  =  6
#        __size_limit    =  32768|ARM.EW.LINKER
#
#################################################################################
```

S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\Core\Src\main.c

```
1   /* USER CODE BEGIN Header */
2   /**
3   ******************************************************************************
4   * @file           : main.c
5   * @brief          : Main program body
6   ******************************************************************************
7   * @attention
8   *
9   * Copyright (c) 2023 STMicroelectronics.
10      * All rights reserved.
11      *
12      * This software is licensed under terms that can be found in the LICENSE file
13      * in the root directory of this software component.
14      * If no LICENSE file comes with this software, it is provided AS-IS.
15      *
16  ******************************************************************************
17      */
18  /* USER CODE END Header */
19  /* Includes ------------------------------------------------------------------*/
20  #include "main.h"
21
22  /* Private includes ----------------------------------------------------------*/
23  /* USER CODE BEGIN Includes */
24
25  /* USER CODE END Includes */
26
27  /* Private typedef -----------------------------------------------------------*/
28  /* USER CODE BEGIN PTD */
29
30  /* USER CODE END PTD */
31
32  /* Private define ------------------------------------------------------------*/
33  /* USER CODE BEGIN PD */
34  /* USER CODE END PD */
35
```

```
      36     /* Private macro -------------------------------------------------------*/
      37     /* USER CODE BEGIN PM */
      38
      39     /* USER CODE END PM */
      40
      41     /* Private variables ---------------------------------------------------*/

\                            In section .bss, align 4
      42     CAN_HandleTypeDef hcan1;
\            hcan1:
\     0x0                    DS8 40
      43
      44     /* USER CODE BEGIN PV */
      45     CAN_RxHeaderTypeDef RxHeader; //Rx refers as input
      46     CAN_TxHeaderTypeDef TxHeader;
      47     uint32_t mailbox;
      48     uint8_t datarx [8];
      49     uint8_t data [8];
      50     CAN_FilterTypeDef filter; //Set filter for receiving specific ID message
\            filter:
\     0x28                   DS8 40
\            TxHeader:
\     0x50                   DS8 24
      51     uint32_t error;
\            error:
\     0x68                   DS8 4
\            mailbox:
\     0x6C                   DS8 4
\            `data`:
\     0x70                   DS8 8

\                            In section .bss, align 4
\            RxHeader:
\     0x0                    DS8 28

\                            In section .bss, align 4
\            datarx:
\     0x0                    DS8 8
      52     /* USER CODE END PV */
      53
```

```
        54      /* Private function prototypes -----------------------------------------------*/
        55      void SystemClock_Config(void);
        56      static void MX_GPIO_Init(void);
        57      static void MX_CAN1_Init(void);
        58      /* USER CODE BEGIN PFP */
        59
        60      /* USER CODE END PFP */
        61
        62      /* Private user code ---------------------------------------------------------*/
        63      /* USER CODE BEGIN 0 */
        64
        65      /* USER CODE END 0 */
        66
        67      /**
        68              * @brief  The application entry point.
        69              * @retval int
        70              */

  \                             In section .text, align 2, keep-with-next
        71      int main(void)
        72      {
  \             main: (+1)
  \     0x0   0xB538          PUSH          {R3-R5,LR}
  \     0x2   0xB086          SUB    SP,SP,#+24
        73              /* USER CODE BEGIN 1 */
        74
        75              /* USER CODE END 1 */
        76
        77              /* MCU Configuration--------------------------------------------------------*/
        78
        79              /* Reset of all peripherals, Initializes the Flash interface and the
Systick. */
        80              HAL_Init();
  \     0x4   0x.... 0x....       BL      HAL_Init
        81
        82              /* USER CODE BEGIN Init */
        83
        84              /* USER CODE END Init */
        85
        86              /* Configure the system clock */
```

```
        87              SystemClock_Config();
\       0x8   0x.... 0x....     BL      SystemClock_Config
        88
        89              /* USER CODE BEGIN SysInit */
        90
        91              /* USER CODE END SysInit */
        92
        93              /* Initialize all configured peripherals */
        94              MX_GPIO_Init();
\       0xC   0x.... 0x....     BL      ?Subroutine1
\             ??CrossCallReturnLabel_2: (+1)
\       0x10  0x2000        MOVS        R0,#+0
\       0x12  0x9000        STR   R0,[SP, #+0]
\       0x14  0x....        LDR.N       R4,??DataTable1
\       0x16  0x....        LDR.N       R0,??DataTable1_1
\       0x18  0x....        LDR.N       R5,??DataTable1_2
\       0x1A  0x6801        LDR   R1,[R0, #+0]
\       0x1C  0xF041 0x0180     ORR   R1,R1,#0x80
\       0x20  0x6001        STR   R1,[R0, #+0]
\       0x22  0x6802        LDR   R2,[R0, #+0]
\       0x24  0xF002 0x0280     AND   R2,R2,#0x80
\       0x28  0x9200        STR   R2,[SP, #+0]
\       0x2A  0x2200        MOVS        R2,#+0
\       0x2C  0x9900        LDR   R1,[SP, #+0]
\       0x2E  0x9200        STR   R2,[SP, #+0]
\       0x30  0x6803        LDR   R3,[R0, #+0]
\       0x32  0xF043 0x0301     ORR   R3,R3,#0x1
\       0x36  0x6003        STR   R3,[R0, #+0]
\       0x38  0x6801        LDR   R1,[R0, #+0]
\       0x3A  0xF001 0x0101     AND   R1,R1,#0x1
\       0x3E  0x9100        STR   R1,[SP, #+0]
\       0x40  0x9900        LDR   R1,[SP, #+0]
\       0x42  0x9200        STR   R2,[SP, #+0]
\       0x44  0xF44F 0x6180     MOV   R1,#+1024
\       0x48  0x6803        LDR   R3,[R0, #+0]
\       0x4A  0xF043 0x0340     ORR   R3,R3,#0x40
\       0x4E  0x6003        STR   R3,[R0, #+0]
\       0x50  0x6800        LDR   R0,[R0, #+0]
\       0x52  0xF000 0x0040     AND   R0,R0,#0x40
\       0x56  0x9000        STR   R0,[SP, #+0]
```

```
\      0x58   0x9800         LDR    R0,[SP, #+0]
\      0x5A   0x4620         MOV    R0,R4
\      0x5C   0x.... 0x....  BL     HAL_GPIO_WritePin
\      0x60   0x2200         MOVS           R2,#+0
\      0x62   0xF44F 0x5100  MOV    R1,#+8192
\      0x66   0x4628         MOV    R0,R5
\      0x68   0x.... 0x....  BL     HAL_GPIO_WritePin
\      0x6C   0x2001         MOVS           R0,#+1
\      0x6E   0xF44F 0x1188  MOV    R1,#+1114112
\      0x72   0x9001         STR    R0,[SP, #+4]
\      0x74   0x9102         STR    R1,[SP, #+8]
\      0x76   0x2200         MOVS           R2,#+0
\      0x78   0x9203         STR    R2,[SP, #+12]
\      0x7A   0xA901                ADD    R1,SP,#+4
\      0x7C   0x4620         MOV    R0,R4
\      0x7E   0x.... 0x....  BL     HAL_GPIO_Init
\      0x82   0xF44F 0x6180  MOV    R1,#+1024
\      0x86   0x.... 0x....  BL     ?Subroutine0
\             ??CrossCallReturnLabel_0: (+1)
\      0x8A   0x4620         MOV    R0,R4
\      0x8C   0x.... 0x....  BL     HAL_GPIO_Init
\      0x90   0xF44F 0x5100  MOV    R1,#+8192
\      0x94   0x.... 0x....  BL     ?Subroutine0
\             ??CrossCallReturnLabel_1: (+1)
\      0x98   0x4628         MOV    R0,R5
\      0x9A   0x.... 0x....  BL     HAL_GPIO_Init
\      0x9E   0x2200         MOVS           R2,#+0
\      0xA0   0x2100         MOVS           R1,#+0
\      0xA2   0x2006         MOVS           R0,#+6
\      0xA4   0x.... 0x....  BL     HAL_NVIC_SetPriority
\      0xA8   0x2006         MOVS           R0,#+6
\      0xAA   0x.... 0x....  BL     HAL_NVIC_EnableIRQ
       95               MX_CAN1_Init();
\      0xAE   0x....         LDR.N          R5,??DataTable1_3
\      0xB0   0x....         LDR.N          R0,??DataTable1_4
\      0xB2   0x6028         STR    R0,[R5, #+0]
\      0xB4   0x2105         MOVS           R1,#+5
\      0xB6   0x6069         STR    R1,[R5, #+4]
\      0xB8   0xF44F 0x1040  MOV    R0,#+3145728
\      0xBC   0x2200                MOVS           R2,#+0
```

```
\     0xBE   0xF44F 0x2140      MOV   R1,#+786432
\     0xC2   0x6168       STR   R0,[R5, #+20]
\     0xC4   0x60AA             STR   R2,[R5, #+8]
\     0xC6   0x60EA             STR   R2,[R5, #+12]
\     0xC8   0x6129       STR   R1,[R5, #+16]
\     0xCA   0x61AA             STR   R2,[R5, #+24]
\     0xCC   0x83AA             STRH        R2,[R5, #+28]
\     0xCE   0x4628             MOV   R0,R5
\     0xD0   0x.... 0x....   BL    HAL_CAN_Init
\     0xD4   0xB108             CBZ.N       R0,??main_0
\     0xD6   0x.... 0x....   BL    Error_Handler
      96              /* USER CODE BEGIN 2 */
      97              filter.FilterActivation = CAN_FILTER_ENABLE; //enabling filter
      98              filter.FilterBank = 0; //0-13
\           ??main_0: (+1)
\     0xDA   0x2100             MOVS        R1,#+0
\     0xDC   0x63E9             STR   R1,[R5, #+60]
      99              filter.SlaveStartFilterBank = 0;//0-13
\     0xDE   0x64E9             STR   R1,[R5, #+76]
      100             filter.FilterScale = CAN_FILTERSCALE_16BIT; //16 or 32 bits
\     0xE0   0x6469       STR   R1,[R5, #+68]
      101             filter.FilterFIFOAssignment = CAN_FILTER_FIFO0; //FIFO 0 or 1
\     0xE2   0x63A9             STR   R1,[R5, #+56]
\     0xE4   0x2001       MOVS        R0,#+1
\     0xE6   0x64A8             STR   R0,[R5, #+72]
      102             filter.FilterMode = CAN_FILTERMODE_IDLIST; // List mode
\     0xE8   0x2101       MOVS        R1,#+1
\     0xEA   0x6429             STR   R1,[R5, #+64]
      103             filter.FilterIdHigh = 0x690 << 5; // First ID
\     0xEC   0xF44F 0x4052      MOV   R0,#+53760
\     0xF0   0x62A8       STR   R0,[R5, #+40]
      104
      105             TxHeader.StdId = 0x440; //maximum 11 bits wide
\     0xF2   0xF44F 0x6188      MOV   R1,#+1088
\     0xF6   0x6529       STR   R1,[R5, #+80]
      106             TxHeader.DLC = 8; //(Data Length Code) length of the message
\     0xF8   0x2008       MOVS        R0,#+8
\     0xFA   0x6628       STR   R0,[R5, #+96]
      107             TxHeader.IDE = CAN_ID_STD; //(Identifier Extension) standard
\     0xFC   0x2100             MOVS        R1,#+0
```

```
  \      0xFE   0x65A9                STR    R1,[R5, #+88]
         108                 TxHeader.RTR = CAN_RTR_DATA; //(Remote Transmission
Request) type of frame is Data
  \      0x100  0x65E9                STR    R1,[R5, #+92]
         109                 //filter.FilterIdLow = 0x4F0 << 5; // Second ID
         110
         111                 HAL_CAN_Start(&hcan1);
  \      0x102  0x4628                MOV    R0,R5
  \      0x104  0x.... 0x....   BL     HAL_CAN_Start
         112                 HAL_CAN_ConfigFilter(&hcan1, &filter);
  \      0x108  0xF105 0x0128   ADD    R1,R5,#+40
  \      0x10C  0x4628                MOV    R0,R5
  \      0x10E  0x.... 0x....   BL     HAL_CAN_ConfigFilter
         113                 HAL_CAN_ActivateNotification(&hcan1,
CAN_IT_RX_FIFO0_MSG_PENDING);
  \      0x112  0x2102                MOVS           R1,#+2
  \      0x114  0x4628                MOV    R0,R5
  \      0x116  0x.... 0x....   BL     HAL_CAN_ActivateNotification
  \      0x11A  0xE004                B.N    ??main_1
         114
         115                 /* USER CODE END 2 */
         116
         117                 /* Infinite loop */
         118                 /* USER CODE BEGIN WHILE */
         119                 while (1)
         120                 {
         121                 error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
         122                 HAL_Delay(5);
         123                 if (error != 0) {
         124                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET);
  \              ??main_2: (+1)
  \      0x11C  0xF44F 0x6180   MOV    R1,#+1024
  \      0x120  0x4620                MOV    R0,R4
  \      0x122  0x.... 0x....   BL     HAL_GPIO_WritePin
  \              ??main_1: (+1)
  \      0x126  0xF105 0x036C   ADD    R3,R5,#+108
  \      0x12A  0xF105 0x0270   ADD    R2,R5,#+112
  \      0x12E  0xF105 0x0150   ADD    R1,R5,#+80
  \      0x132  0x4628                MOV    R0,R5
```

```
\     0x134   0x.... 0x....   BL      HAL_CAN_AddTxMessage
\     0x138   0x66A8          STR     R0,[R5, #+104]
\     0x13A   0x2005          MOVS    R0,#+5
\     0x13C   0x.... 0x....   BL      HAL_Delay
\     0x140   0x6EA8          LDR     R0,[R5, #+104]
\     0x142   0x2800          CMP     R0,#+0
\     0x144   0xBF0C          ITE     EQ
\     0x146   0x2200          MOVEQ   R2,#+0
\     0x148   0x2201          MOVNE   R2,#+1
      125             }
      126             else{
      127             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_RESET);
\     0x14A   0xE7E7          B.N     ??main_2
      128             }
      129             /* USER CODE END WHILE */
      130
      131             /* USER CODE BEGIN 3 */
      132             }
      133             /* USER CODE END 3 */
      134             }
      135
      136             /**
      137             * @brief System Clock Configuration
      138             * @retval None
      139             */

\                               In section .text, align 2, keep-with-next
      140             void SystemClock_Config(void)
      141             {
\               SystemClock_Config: (+1)
\     0x0   0xB580            PUSH    {R7,LR}
\     0x2   0xB092            SUB     SP,SP,#+72
\     0x4   0x2230            MOVS    R2,#+48
\     0x6   0x2100            MOVS    R1,#+0
\     0x8   0xA806            ADD     R0,SP,#+24
\     0xA   0x.... 0x....     BL      memset
\     0xE   0x.... 0x....     BL      ?Subroutine1
      142             RCC_OscInitTypeDef RCC_OscInitStruct = {0};
      143             RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
      144
```

```
        145             /** Configure the main internal regulator output voltage
        146             */
        147             __HAL_RCC_PWR_CLK_ENABLE();
  \             ??CrossCallReturnLabel_3: (+1)
  \     0x12   0x2000         MOVS         R0,#+0
  \     0x14   0x9000         STR    R0,[SP, #+0]
        148
__HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1
);
        149
        150             /** Initializes the RCC Oscillators according to the specified
parameters
        151             * in the RCC_OscInitTypeDef structure.
        152             */
        153             RCC_OscInitStruct.OscillatorType =
RCC_OSCILLATORTYPE_HSE;
        154             RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  \     0x16   0xF44F 0x3380       MOV   R3,#+65536
  \     0x1A   0x....              LDR.N         R0,??DataTable1_5
  \     0x1C   0x6801         LDR    R1,[R0, #+0]
  \     0x1E   0xF041 0x5180       ORR   R1,R1,#0x10000000
  \     0x22   0x6001         STR    R1,[R0, #+0]
  \     0x24   0x2100         MOVS         R1,#+0
  \     0x26   0x6800         LDR    R0,[R0, #+0]
  \     0x28   0xF000 0x5080       AND   R0,R0,#0x10000000
  \     0x2C   0x9000         STR    R0,[SP, #+0]
  \     0x2E   0x9800         LDR    R0,[SP, #+0]
  \     0x30   0x....              LDR.N         R0,??DataTable1_6
  \     0x32   0x9100         STR    R1,[SP, #+0]
  \     0x34   0x6802         LDR    R2,[R0, #+0]
  \     0x36   0xF442 0x4240       ORR   R2,R2,#0xC000
  \     0x3A   0x6002         STR    R2,[R0, #+0]
  \     0x3C   0x2201         MOVS         R2,#+1
  \     0x3E   0x6800         LDR    R0,[R0, #+0]
  \     0x40   0xF400 0x4040       AND   R0,R0,#0xC000
  \     0x44   0x9000         STR    R0,[SP, #+0]
        155             RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
        156             RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
        157             RCC_OscInitStruct.PLL.PLLM = 4;
        158             RCC_OscInitStruct.PLL.PLLN = 180;
```

```
    159             RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
    160             RCC_OscInitStruct.PLL.PLLQ = 4;
    161             if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
\   0x46  0xA806        ADD   R0,SP,#+24
\   0x48  0x9900        LDR   R1,[SP, #+0]
\   0x4A  0x9206        STR   R2,[SP, #+24]
\   0x4C  0x2102        MOVS        R1,#+2
\   0x4E  0xF44F 0x0280     MOV   R2,#+4194304
\   0x52  0x910C        STR   R1,[SP, #+48]
\   0x54  0x920D        STR   R2,[SP, #+52]
\   0x56  0x2104        MOVS        R1,#+4
\   0x58  0x22B4        MOVS        R2,#+180
\   0x5A  0x910E              STR   R1,[SP, #+56]
\   0x5C  0x920F              STR   R2,[SP, #+60]
\   0x5E  0x2102        MOVS        R1,#+2
\   0x60  0x2204        MOVS        R2,#+4
\   0x62  0x9307        STR   R3,[SP, #+28]
\   0x64  0x9110        STR   R1,[SP, #+64]
\   0x66  0x9211        STR   R2,[SP, #+68]
\   0x68  0x.... 0x....     BL    HAL_RCC_OscConfig
\   0x6C  0xB108              CBZ.N       R0,??SystemClock_Config_0
    162             {
    163             Error_Handler();
\   0x6E  0x.... 0x....     BL    Error_Handler
    164             }
    165
    166             /** Activate the Over-Drive mode
    167             */
    168             if (HAL_PWREx_EnableOverDrive() != HAL_OK)
\           ??SystemClock_Config_0: (+1)
\   0x72  0x.... 0x....     BL    HAL_PWREx_EnableOverDrive
\   0x76  0xB108        CBZ.N       R0,??SystemClock_Config_1
    169             {
    170             Error_Handler();
\   0x78  0xB672        CPSID       I
\           ??SystemClock_Config_2: (+1)
\   0x7A  0xE7FE              B.N   ??SystemClock_Config_2
    171             }
    172
    173             /** Initializes the CPU, AHB and APB buses clocks
```

```
        174             */
        175             RCC_ClkInitStruct.ClockType =
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
        176
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
        177             RCC_ClkInitStruct.SYSCLKSource =
RCC_SYSCLKSOURCE_PLLCLK;
  \             ??SystemClock_Config_1: (+1)
  \     0x7C  0x2102        MOVS        R1,#+2
  \     0x7E  0x9102        STR   R1,[SP, #+8]
  \     0x80  0x200F        MOVS        R0,#+15
        178             RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
        179             RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
  \     0x82  0xF44F 0x51A0       MOV   R1,#+5120
  \     0x86  0x9001        STR   R0,[SP, #+4]
  \     0x88  0x2200        MOVS        R2,#+0
  \     0x8A  0x9104        STR   R1,[SP, #+16]
        180             RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV4;
  \     0x8C  0x9105        STR   R1,[SP, #+20]
  \     0x8E  0x9203        STR   R2,[SP, #+12]
        181
        182             if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct,
FLASH_LATENCY_5) != HAL_OK)
  \     0x90  0x2105        MOVS        R1,#+5
  \     0x92  0xA801        ADD   R0,SP,#+4
  \     0x94  0x.... 0x....     BL      HAL_RCC_ClockConfig
  \     0x98  0xB108        CBZ.N       R0,??SystemClock_Config_3
        183             {
        184             Error_Handler();
  \     0x9A  0xB672                CPSID       I
  \             ??SystemClock_Config_4: (+1)
  \     0x9C  0xE7FE                B.N   ??SystemClock_Config_4
        185             }
        186             }
  \             ??SystemClock_Config_3: (+1)
  \     0x9E  0xB013                ADD   SP,SP,#+76
  \     0xA0  0xBD00                POP   {PC}

  \                             In section .text, align 2, keep-with-next
  \             ?Subroutine1: (+1)
```

```
\     0x0  0x2214          MOVS        R2,#+20
\     0x2  0x2100          MOVS        R1,#+0
\     0x4  0xA801          ADD   R0,SP,#+4
\     0x6  0x.... 0x....   B.W   memset
     187
     188            /**
     189            * @brief CAN1 Initialization Function
     190            * @param None
     191            * @retval None
     192            */
     193            static void MX_CAN1_Init(void)
     194            {
     195
     196            /* USER CODE BEGIN CAN1_Init 0 */
     197
     198            /* USER CODE END CAN1_Init 0 */
     199
     200            /* USER CODE BEGIN CAN1_Init 1 */
     201
     202            /* USER CODE END CAN1_Init 1 */
     203            hcan1.Instance = CAN1;
     204            hcan1.Init.Prescaler = 5;
     205            hcan1.Init.Mode = CAN_MODE_NORMAL;
     206            hcan1.Init.SyncJumpWidth = CAN_SJW_1TQ;
     207            hcan1.Init.TimeSeg1 = CAN_BS1_13TQ;
     208            hcan1.Init.TimeSeg2 = CAN_BS2_4TQ;
     209            hcan1.Init.TimeTriggeredMode = DISABLE;
     210            hcan1.Init.AutoBusOff = DISABLE;
     211            hcan1.Init.AutoWakeUp = DISABLE;
     212            hcan1.Init.AutoRetransmission = DISABLE;
     213            hcan1.Init.ReceiveFifoLocked = DISABLE;
     214            hcan1.Init.TransmitFifoPriority = DISABLE;
     215            if (HAL_CAN_Init(&hcan1) != HAL_OK)
     216            {
     217            Error_Handler();
     218            }
     219            /* USER CODE BEGIN CAN1_Init 2 */
     220
     221            /* USER CODE END CAN1_Init 2 */
     222
```

```
223         }
224
225         /**
226         * @brief GPIO Initialization Function
227         * @param None
228         * @retval None
229         */
230         static void MX_GPIO_Init(void)
231         {
232         GPIO_InitTypeDef GPIO_InitStruct = {0};
233
234         /* GPIO Ports Clock Enable */
235         __HAL_RCC_GPIOH_CLK_ENABLE();
236         __HAL_RCC_GPIOA_CLK_ENABLE();
237         __HAL_RCC_GPIOG_CLK_ENABLE();
238
239         /*Configure GPIO pin Output Level */
240         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_RESET);
241
242         /*Configure GPIO pin Output Level */
243         HAL_GPIO_WritePin(GPIOG, GPIO_PIN_13, GPIO_PIN_RESET);
244
245         /*Configure GPIO pin : PA0 */
246         GPIO_InitStruct.Pin = GPIO_PIN_0;
247         GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
248         GPIO_InitStruct.Pull = GPIO_NOPULL;
249         HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
250
251         /*Configure GPIO pin : PA10 */
252         GPIO_InitStruct.Pin = GPIO_PIN_10;
253         GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
254         GPIO_InitStruct.Pull = GPIO_NOPULL;
255         GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
256         HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
257
258         /*Configure GPIO pin : PG13 */
259         GPIO_InitStruct.Pin = GPIO_PIN_13;
260         GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
261         GPIO_InitStruct.Pull = GPIO_NOPULL;
262         GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
```

```
        263             HAL_GPIO_Init(GPIOG, &GPIO_InitStruct);
        264
        265             /* EXTI interrupt init*/
        266             HAL_NVIC_SetPriority(EXTI0_IRQn, 0, 0);
        267             HAL_NVIC_EnableIRQ(EXTI0_IRQn);
        268
        269             }
        270
        271             /* USER CODE BEGIN 4 */
        272
        273             /* USER CODE END 4 */
        274
        275             /**
        276             * @brief  This function is executed in case of error occurrence.
        277             * @retval None
        278             */

  \                             In section .text, align 2, keep-with-next
        279             void Error_Handler(void)
        280             {
        281             /* USER CODE BEGIN Error_Handler_Debug */
        282             /* User can add his own implementation to report the HAL error
return state */
        283               __disable_irq();
  \               Error_Handler: (+1)
  \       0x0  0xB672           CPSID         I
        284               while (1)
  \               ??Error_Handler_0: (+1)
  \       0x2  0xE7FE           B.N     ??Error_Handler_0
        285             {
        286             }
        287             /* USER CODE END Error_Handler_Debug */
        288             }

  \                             In section .text, align 2, keep-with-next
  \               ?Subroutine0: (+1)
  \       0x0  0x9101           STR    R1,[SP, #+4]
  \       0x2  0x2201           MOVS          R2,#+1
  \       0x4  0x2100           MOVS          R1,#+0
  \       0x6  0x9103           STR    R1,[SP, #+12]
```

```
\      0x8   0x9104          STR    R1,[SP, #+16]
\      0xA   0x9202          STR    R2,[SP, #+8]
\      0xC   0xA901          ADD    R1,SP,#+4
\      0xE   0x4770          BX     LR

\                            In section .text, align 4, keep-with-next
\             ??DataTable1:
\      0x0   0x4002'0000            DC32  0x40020000

\                            In section .text, align 4, keep-with-next
\             ??DataTable1_1:
\      0x0   0x4002'3830            DC32  0x40023830

\                            In section .text, align 4, keep-with-next
\             ??DataTable1_2:
\      0x0   0x4002'1800            DC32  0x40021800

\                            In section .text, align 4, keep-with-next
\             ??DataTable1_3:
\      0x0   0x....'....            DC32  hcan1

\                            In section .text, align 4, keep-with-next
\             ??DataTable1_4:
\      0x0   0x4000'6400            DC32  0x40006400

\                            In section .text, align 4, keep-with-next
\             ??DataTable1_5:
\      0x0   0x4002'3840            DC32  0x40023840

\                            In section .text, align 4, keep-with-next
\             ??DataTable1_6:
\      0x0   0x4000'7000            DC32  0x40007000
       289
       290            #ifdef  USE_FULL_ASSERT
       291            /**
       292            * @brief  Reports the name of the source file and the source line
number
       293            *        where the assert_param error has occurred.
       294            * @param  file: pointer to the source file name
       295            * @param  line: assert_param error line source number
```

```
        296             * @retval None
        297             */
        298             void assert_failed(uint8_t *file, uint32_t line)
        299             {
        300             /* USER CODE BEGIN 6 */
        301             /* User can add his own implementation to report the file name and
line number,
        302             ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line)
*/
        303             /* USER CODE END 6 */
        304             }
        305             #endif /* USE_FULL_ASSERT */

  Maximum stack usage in bytes:

  .cstack Function
  ------- --------
        0   Error_Handler
        80   SystemClock_Config
        80   -> Error_Handler
        80   -> HAL_PWREx_EnableOverDrive
        80   -> HAL_RCC_ClockConfig
        80   -> HAL_RCC_OscConfig
        80   -> memset
        40   main
        40   -> Error_Handler
        40   -> HAL_CAN_ActivateNotification
        40   -> HAL_CAN_AddTxMessage
        40   -> HAL_CAN_ConfigFilter
        40   -> HAL_CAN_Init
        40   -> HAL_CAN_Start
        40   -> HAL_Delay
        40   -> HAL_GPIO_Init
        40   -> HAL_GPIO_WritePin
        40   -> HAL_Init
        40   -> HAL_NVIC_EnableIRQ
        40   -> HAL_NVIC_SetPriority
        40   -> SystemClock_Config
        40   -> memset
```

Section sizes:

Bytes Function/Label
----- --------------
4 ??DataTable1
4 ??DataTable1_1
4 ??DataTable1_2
4 ??DataTable1_3
4 ??DataTable1_4
4 ??DataTable1_5
4 ??DataTable1_6
16 ?Subroutine0
10 ?Subroutine1
4 Error_Handler
28 RxHeader
162 SystemClock_Config
8 datarx
120 hcan1
filter
TxHeader
error
mailbox
data
332 main

156 bytes in section .bss
552 bytes in section .text

552 bytes of CODE memory
156 bytes of DATA memory

Errors: none
Warnings: none

It.lst
##
#########

```
#
# IAR ANSI C/C++ Compiler V9.20.4.327/W64 for ARM         02/Mar/2023  15:31:34
# Copyright 1999-2022 IAR Systems AB.
#
#       Cpu mode              =  thumb
#       Endian          =  little
#       Source file     =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\Core\Src\stm
32f4xx_it.c
#       Command line        =
#       -f
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM\Tas
k3\Obj\Application\User\Core\stm32f4xx_it.o.rsp
#
(S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\Core\Src\st
m32f4xx_it.c
#       -D USE_HAL_DRIVER -D STM32F429xx -lC
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM\Tas
k3\List\Application\User\Core
#       -o
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM\Tas
k3\Obj\Application\User\Core
#       --debug --endian=little --cpu=Cortex-M4 -e --fpu=VFPv4_sp
#       --dlib_config S:\School_Work\arm\inc\c\DLib_Config_Full.h -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM/../C
ore/Inc\
#       -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM/../D
rivers/STM32F4xx_HAL_Driver/Inc\
#       -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM/../D
rivers/STM32F4xx_HAL_Driver/Inc/Legacy\
#       -I
```

```
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM/../D
rivers/CMSIS/Device/ST/STM32F4xx/Include\
#       -I
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM/../D
rivers/CMSIS/Include\
#       -Ohz) --dependencies=n
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM\Tas
k3\Obj\Application\User\Core\stm32f4xx_it.o.d
#       Locale          =  C
#       List file       =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM\Tas
k3\List\Application\User\Core\stm32f4xx_it.lst
#       Object file     =
#
S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\EWARM\Tas
k3\Obj\Application\User\Core\stm32f4xx_it.o
#       Runtime model:
#       __CPP_Runtime   =  1
#       __SystemLibrary =  DLib
#       __dlib_version  =  6
#       __size_limit    =  32768|ARM.EW.LINKER
#
###############################################################################
#########

S:\School_Work\Spring_2023\MicroconrollerApps\Prelabs\Prelab_7\Task3\Core\Src\stm
32f4xx_it.c
       1        /* USER CODE BEGIN Header */
       2        /**
       3        ******************************************************************************
       4        * @filestm32f4xx_it.c
       5        * @brief   Interrupt Service Routines.
       6        ******************************************************************************
       7        * @attention
       8        *
       9        * Copyright (c) 2023 STMicroelectronics.
```

```
        10              * All rights reserved.
        11              *
        12              * This software is licensed under terms that can be found in the
LICENSE file
        13              * in the root directory of this software component.
        14              * If no LICENSE file comes with this software, it is provided AS-IS.
        15              *
        16
******************************************************************************
        17              */
        18      /* USER CODE END Header */
        19
        20      /* Includes ------------------------------------------------------------*/
        21      #include "main.h"
        22      #include "stm32f4xx_it.h"
        23      /* Private includes ----------------------------------------------------------*/
        24      /* USER CODE BEGIN Includes */
        25      /* USER CODE END Includes */
        26
        27      /* Private typedef -----------------------------------------------------------*/
        28      /* USER CODE BEGIN TD */
        29
        30      /* USER CODE END TD */
        31
        32      /* Private define ------------------------------------------------------------*/
        33      /* USER CODE BEGIN PD */
        34
        35      /* USER CODE END PD */
        36
        37      /* Private macro -------------------------------------------------------------*/
        38      /* USER CODE BEGIN PM */
        39
        40      /* USER CODE END PM */
        41
        42      /* Private variables ---------------------------------------------------------*/
        43      /* USER CODE BEGIN PV */

  \                             In section .bss, align 1
        44      uint8_t message = 0;
  \             message:
```

```
\      0x0                    DS8 1
       45      /* USER CODE END PV */
       46
       47      /* Private function prototypes -----------------------------------------------*/
       48      /* USER CODE BEGIN PFP */
       49
       50      /* USER CODE END PFP */
       51
       52      /* Private user code ---------------------------------------------------------*/
       53      /* USER CODE BEGIN 0 */
       54
       55      /* USER CODE END 0 */
       56
       57      /* External variables --------------------------------------------------------*/
       58      extern CAN_HandleTypeDef hcan1;
       59      /* USER CODE BEGIN EV */
       60      extern uint8_t datarx[];
       61      extern CAN_FilterTypeDef filter;
       62      extern CAN_RxHeaderTypeDef RxHeader; //Rx refers as input
       63      extern CAN_TxHeaderTypeDef TxHeader;
       64      extern uint32_t mailbox;
       65      extern uint8_t data [8];
       66      extern uint32_t error;
       67      /* USER CODE END EV */
       68
       69      /******************************************************************************/
       70      /*       Cortex-M4 Processor Interruption and Exception Handlers
*/
       71      /******************************************************************************/
       72      /**
       73            * @brief This function handles Non maskable interrupt.
       74            */

\                           In section .text, align 2, keep-with-next
       75      void NMI_Handler(void)
       76      {
       77            /* USER CODE BEGIN NonMaskableInt_IRQn 0 */
       78
       79            /* USER CODE END NonMaskableInt_IRQn 0 */
       80            /* USER CODE BEGIN NonMaskableInt_IRQn 1 */
```

```
     81              while (1)
\              NMI_Handler: (+1)
\              ??NMI_Handler_0: (+1)
\        0x0   0xE7FE          B.N     ??NMI_Handler_0
     82              {
     83              }
     84              /* USER CODE END NonMaskableInt_IRQn 1 */
     85      }
     86
     87      /**
     88              * @brief This function handles Hard fault interrupt.
     89              */

\                               In section .text, align 2, keep-with-next
     90      void HardFault_Handler(void)
     91      {
     92              /* USER CODE BEGIN HardFault_IRQn 0 */
     93
     94              /* USER CODE END HardFault_IRQn 0 */
     95              while (1)
\              HardFault_Handler: (+1)
\              ??HardFault_Handler_0: (+1)
\        0x0   0xE7FE          B.N     ??HardFault_Handler_0
     96              {
     97              /* USER CODE BEGIN W1_HardFault_IRQn 0 */
     98              /* USER CODE END W1_HardFault_IRQn 0 */
     99              }
    100             }
    101
    102             /**
    103             * @brief This function handles Memory management fault.
    104             */

\                               In section .text, align 2, keep-with-next
    105             void MemManage_Handler(void)
    106             {
    107             /* USER CODE BEGIN MemoryManagement_IRQn 0 */
    108
    109             /* USER CODE END MemoryManagement_IRQn 0 */
    110             while (1)
```

```
\             MemManage_Handler: (+1)
\             ??MemManage_Handler_0: (+1)
\     0x0  0xE7FE         B.N     ??MemManage_Handler_0
      111           {
      112           /* USER CODE BEGIN W1_MemoryManagement_IRQn 0 */
      113           /* USER CODE END W1_MemoryManagement_IRQn 0 */
      114           }
      115           }
      116
      117           /**
      118           * @brief This function handles Pre-fetch fault, memory access fault.
      119           */

\                           In section .text, align 2, keep-with-next
      120           void BusFault_Handler(void)
      121           {
      122           /* USER CODE BEGIN BusFault_IRQn 0 */
      123
      124           /* USER CODE END BusFault_IRQn 0 */
      125           while (1)
\             BusFault_Handler: (+1)
\             ??BusFault_Handler_0: (+1)
\     0x0  0xE7FE         B.N     ??BusFault_Handler_0
      126           {
      127           /* USER CODE BEGIN W1_BusFault_IRQn 0 */
      128           /* USER CODE END W1_BusFault_IRQn 0 */
      129           }
      130           }
      131
      132           /**
      133           * @brief This function handles Undefined instruction or illegal state.
      134           */

\                           In section .text, align 2, keep-with-next
      135           void UsageFault_Handler(void)
      136           {
      137           /* USER CODE BEGIN UsageFault_IRQn 0 */
      138
      139           /* USER CODE END UsageFault_IRQn 0 */
      140           while (1)
```

```
\              UsageFault_Handler: (+1)
\              ??UsageFault_Handler_0: (+1)
\      0x0  0xE7FE          B.N     ??UsageFault_Handler_0
       141            {
       142            /* USER CODE BEGIN W1_UsageFault_IRQn 0 */
       143            /* USER CODE END W1_UsageFault_IRQn 0 */
       144            }
       145            }
       146
       147            /**
       148            * @brief This function handles System service call via SWI
instruction.
       149            */

\                             In section .text, align 2, keep-with-next
       150            void SVC_Handler(void)
       151            {
       152            /* USER CODE BEGIN SVCall_IRQn 0 */
       153
       154            /* USER CODE END SVCall_IRQn 0 */
       155            /* USER CODE BEGIN SVCall_IRQn 1 */
       156
       157            /* USER CODE END SVCall_IRQn 1 */
       158            }
\              SVC_Handler: (+1)
\      0x0  0x4770          BX      LR
       159
       160            /**
       161            * @brief This function handles Debug monitor.
       162            */

\                             In section .text, align 2, keep-with-next
       163            void DebugMon_Handler(void)
       164            {
       165            /* USER CODE BEGIN DebugMonitor_IRQn 0 */
       166
       167            /* USER CODE END DebugMonitor_IRQn 0 */
       168            /* USER CODE BEGIN DebugMonitor_IRQn 1 */
       169
       170            /* USER CODE END DebugMonitor_IRQn 1 */
```

```
      171           }
\           DebugMon_Handler: (+1)
\     0x0  0x4770         BX      LR
      172
      173           /**
      174           * @brief This function handles Pendable request for system
service.
      175           */

\                         In section .text, align 2, keep-with-next
      176           void PendSV_Handler(void)
      177           {
      178           /* USER CODE BEGIN PendSV_IRQn 0 */
      179
      180           /* USER CODE END PendSV_IRQn 0 */
      181           /* USER CODE BEGIN PendSV_IRQn 1 */
      182
      183           /* USER CODE END PendSV_IRQn 1 */
      184           }
\           PendSV_Handler: (+1)
\     0x0  0x4770         BX      LR
      185
      186           /**
      187           * @brief This function handles System tick timer.
      188           */

\                         In section .text, align 2, keep-with-next
      189           void SysTick_Handler(void)
      190           {
      191           /* USER CODE BEGIN SysTick_IRQn 0 */
      192
      193           /* USER CODE END SysTick_IRQn 0 */
      194           HAL_IncTick();
\           SysTick_Handler: (+1)
\     0x0  0x.... 0x....      B.W    HAL_IncTick
      195           /* USER CODE BEGIN SysTick_IRQn 1 */
      196
      197           /* USER CODE END SysTick_IRQn 1 */
      198           }
      199
```

```
        200
/******************************************************************************/
        201             /* STM32F4xx Peripheral Interrupt Handlers
*/
        202             /* Add here the Interrupt Handlers for the used peripherals.
*/
        203             /* For the available peripheral interrupt handler names,
*/
        204             /* please refer to the startup file (startup_stm32f4xx.s).
*/
        205
/******************************************************************************/
        206
        207             /**
        208             * @brief This function handles EXTI line0 interrupt.
        209             */

  \                             In section .text, align 2, keep-with-next
        210             void EXTI0_IRQHandler(void)
        211             {
  \               EXTI0_IRQHandler: (+1)
  \       0x0   0xB580          PUSH         {R7,LR}
        212             /* USER CODE BEGIN EXTI0_IRQn 0 */
        213
        214             /* USER CODE END EXTI0_IRQn 0 */
        215             HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);
  \       0x2   0x2001          MOVS         R0,#+1
  \       0x4   0x.... 0x....   BL      HAL_GPIO_EXTI_IRQHandler
        216             /* USER CODE BEGIN EXTI0_IRQn 1 */
        217             if (message < 8) {
  \       0x8   0x....          LDR.N        R0,??DataTable3
  \       0xA   0x7801          LDRB R1,[R0, #+0]
  \       0xC   0x2908          CMP   R1,#+8
  \       0xE   0xBFB4          ITE     LT
  \       0x10   0x1C49         ADDLT        R1,R1,#+1
  \       0x12   0x2100         MOVGE        R1,#+0
        218             message ++;
        219             }
        220             else {
        221             message = 0;
```

```
    222           }
    223           /* USER CODE END EXTI0_IRQn 1 */
    224           }
\     0x14  0x7001        STRB R1,[R0, #+0]
\     0x16  0xBD01              POP   {R0,PC}
    225
    226           /**
    227           * @brief This function handles CAN1 TX interrupts.
    228           */

\                             In section .text, align 2, keep-with-next
    229           void CAN1_TX_IRQHandler(void)
    230           {
    231           /* USER CODE BEGIN CAN1_TX_IRQn 0 */
    232
    233           /* USER CODE END CAN1_TX_IRQn 0 */
    234           HAL_CAN_IRQHandler(&hcan1);
\           CAN1_TX_IRQHandler: (+1)
\     0x0   0x....              B.N    ?Subroutine0
    235           /* USER CODE BEGIN CAN1_TX_IRQn 1 */
    236
    237           /* USER CODE END CAN1_TX_IRQn 1 */
    238           }

\                             In section .text, align 2, keep-with-next
\           ?Subroutine0: (+1)
\     0x0   0x....              LDR.N          R0,??DataTable3_1
\     0x2   0x.... 0x....       B.W    HAL_CAN_IRQHandler
    239
    240           /**
    241           * @brief This function handles CAN1 RX0 interrupts.
    242           */

\                             In section .text, align 2, keep-with-next
    243           void CAN1_RX0_IRQHandler(void)
    244           {
\           CAN1_RX0_IRQHandler: (+1)
\     0x0   0xE92D 0x43F8       PUSH           {R3-R9,LR}
    245           /* USER CODE BEGIN CAN1_RX0_IRQn 0 */
    246
```

```
        247             /* USER CODE END CAN1_RX0_IRQn 0 */
        248             HAL_CAN_IRQHandler(&hcan1);
  \     0x4   0x.... 0x....     LDR.W       R8,??DataTable3_1
  \     0x8   0x....            LDR.N       R5,??DataTable3_2
  \     0xA   0x4640            MOV   R0,R8
  \     0xC   0x.... 0x....     BL      HAL_CAN_IRQHandler
        249             /* USER CODE BEGIN CAN1_RX0_IRQn 1 */
        250             HAL_CAN_GetRxMessage(&hcan1, CAN_FILTER_FIFO0,
&RxHeader, datarx);
  \     0x10  0x....            LDR.N       R3,??DataTable3_3
  \     0x12  0x462A            MOV   R2,R5
  \     0x14  0x2100            MOVS        R1,#+0
  \     0x16  0x4640            MOV   R0,R8
  \     0x18  0x.... 0x....     BL      HAL_CAN_GetRxMessage
        251             if (RxHeader.StdId == 0x690){
  \     0x1C  0x6828            LDR   R0,[R5, #+0]
  \     0x1E  0xF5B0 0x6FD2     CMP   R0,#+1680
  \     0x22  0xD17D                    BNE.N       ??CAN1_RX0_IRQHandler_0
  \     0x24  0x....            LDR.N       R5,??DataTable3_4
        252             if (message == 0) {
  \     0x26  0x....            LDR.N       R1,??DataTable3
  \     0x28  0x6828            LDR   R0,[R5, #+0]
  \     0x2A  0x7809            LDRB  R1,[R1, #+0]
  \     0x2C  0x.... 0x....     LDR.W       R9,??DataTable3_5
  \     0x30  0x....            LDR.N       R4,??DataTable3_6
  \     0x32  0x....            LDR.N       R7,??DataTable3_7
  \     0x34  0xF240 0x3616     MOVW        R6,#+790
  \     0x38  0xF44F 0x6288     MOV   R2,#+1088
  \     0x3C  0xB949                    CBNZ.N  R1,??CAN1_RX0_IRQHandler_1
        253             TxHeader.StdId = 0x440;
  \     0x3E  0x603A                    STR   R2,[R7, #+0]
        254             data[2] =0x00;
  \     0x40  0x2000            MOVS        R0,#+0
  \     0x42  0x.... 0x....     BL      ?Subroutine1
        255             error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
  \             ??CrossCallReturnLabel_8: (+1)
  \     0x46  0x6028            STR   R0,[R5, #+0]
        256             TxHeader.StdId = 0x316;
  \     0x48  0x603E            STR   R6,[R7, #+0]
```

```
        257             data[2] =0x00;
  \     0x4A  0x2000        MOVS        R0,#+0
  \     0x4C  0x70A0                STRB R0,[R4, #+2]
        258             data[3] =0x00;
  \     0x4E  0x70E0                STRB R0,[R4, #+3]
        259             error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
  \     0x50  0xE05F        B.N     ??CAN1_RX0_IRQHandler_2
        260             }
        261             else if (message == 1) {
  \             ??CAN1_RX0_IRQHandler_1: (+1)
  \     0x52  0x2901        CMP   R1,#+1
  \     0x54  0xD109        BNE.N         ??CAN1_RX0_IRQHandler_3
        262             TxHeader.StdId = 0x440;
  \     0x56  0x603A        STR   R2,[R7, #+0]
        263             data[2] =0x1D;
  \     0x58  0x201D        MOVS        R0,#+29
  \     0x5A  0x.... 0x....     BL      ?Subroutine1
        264             error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
  \             ??CrossCallReturnLabel_7: (+1)
  \     0x5E  0x6028        STR   R0,[R5, #+0]
        265             TxHeader.StdId = 0x316;
  \     0x60  0x603E        STR   R6,[R7, #+0]
        266             data[2] =0xa0;
  \     0x62  0x20A0        MOVS        R0,#+160
  \     0x64  0x70A0        STRB R0,[R4, #+2]
        267             data[3] =0x0f;
  \     0x66  0x210F        MOVS        R1,#+15
  \     0x68  0xE052        B.N     ??CAN1_RX0_IRQHandler_4
        268             error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
        269             }
        270             else if (message == 2) {
  \             ??CAN1_RX0_IRQHandler_3: (+1)
  \     0x6A  0x2902        CMP   R1,#+2
  \     0x6C  0xD109                BNE.N         ??CAN1_RX0_IRQHandler_5
        271             TxHeader.StdId = 0x440;
  \     0x6E  0x603A                STR   R2,[R7, #+0]
        272             data[2] =0x3c;
```

```
  \      0x70   0x203C         MOVS          R0,#+60
  \      0x72   0x.... 0x....  BL      ?Subroutine1
         273              error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
  \              ??CrossCallReturnLabel_6: (+1)
  \      0x76   0x6028         STR    R0,[R5, #+0]
         274              TxHeader.StdId = 0x316;
  \      0x78   0x603E         STR    R6,[R7, #+0]
         275              data[2] =0x40;
  \      0x7A   0x2040         MOVS          R0,#+64
  \      0x7C   0x70A0                 STRB R0,[R4, #+2]
         276              data[3] =0x1f;
  \      0x7E   0x211F         MOVS          R1,#+31
  \      0x80   0xE046         B.N     ??CAN1_RX0_IRQHandler_4
         277              error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
         278              }
         279              else if (message == 3) {
  \              ??CAN1_RX0_IRQHandler_5: (+1)
  \      0x82   0x2903         CMP    R1,#+3
  \      0x84   0xD109         BNE.N          ??CAN1_RX0_IRQHandler_6
         280              TxHeader.StdId = 0x440;
  \      0x86   0x603A         STR    R2,[R7, #+0]
         281              data[2] =0x5b;
  \      0x88   0x205B         MOVS          R0,#+91
  \      0x8A   0x.... 0x....  BL      ?Subroutine1
         282              error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
  \              ??CrossCallReturnLabel_5: (+1)
  \      0x8E   0x6028         STR    R0,[R5, #+0]
         283              TxHeader.StdId = 0x316;
  \      0x90   0x603E         STR    R6,[R7, #+0]
         284              data[2] =0xe0;
  \      0x92   0x20E0         MOVS          R0,#+224
  \      0x94   0x70A0         STRB R0,[R4, #+2]
         285              data[3] =0x2e;
  \      0x96   0x212E         MOVS          R1,#+46
  \      0x98   0xE03A                 B.N     ??CAN1_RX0_IRQHandler_4
         286              error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
```

```
        287             }
        288             else if (message == 4) {
  \             ??CAN1_RX0_IRQHandler_6: (+1)
  \     0x9A  0x2904        CMP   R1,#+4
  \     0x9C  0xD109              BNE.N         ??CAN1_RX0_IRQHandler_7
        289             TxHeader.StdId = 0x440;
  \     0x9E  0x603A              STR   R2,[R7, #+0]
        290             data[2] =0x7b;
  \     0xA0  0x207B              MOVS          R0,#+123
  \     0xA2  0x.... 0x....   BL      ?Subroutine1
        291             error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
  \             ??CrossCallReturnLabel_4: (+1)
  \     0xA6  0x6028        STR   R0,[R5, #+0]
        292             TxHeader.StdId = 0x316;
  \     0xA8  0x603E              STR   R6,[R7, #+0]
        293             data[2] =0x80;
  \     0xAA  0x2080              MOVS          R0,#+128
  \     0xAC  0x70A0              STRB  R0,[R4, #+2]
        294             data[3] =0x3e;
  \     0xAE  0x213E              MOVS          R1,#+62
  \     0xB0  0xE02E              B.N     ??CAN1_RX0_IRQHandler_4
        295             error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
        296             }
        297             else if (message == 5) {
  \             ??CAN1_RX0_IRQHandler_7: (+1)
  \     0xB2  0x2905        CMP   R1,#+5
  \     0xB4  0xD109              BNE.N         ??CAN1_RX0_IRQHandler_8
        298             TxHeader.StdId = 0x440;
  \     0xB6  0x603A              STR   R2,[R7, #+0]
        299             data[2] =0x9a;
  \     0xB8  0x209A              MOVS          R0,#+154
  \     0xBA  0x.... 0x....   BL      ?Subroutine1
        300             error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
  \             ??CrossCallReturnLabel_3: (+1)
  \     0xBE  0x6028              STR   R0,[R5, #+0]
        301             TxHeader.StdId = 0x316;
  \     0xC0  0x603E              STR   R6,[R7, #+0]
```

```
        302             data[2] =0x20;
  \     0xC2  0x2020        MOVS        R0,#+32
  \     0xC4  0x70A0                STRB R0,[R4, #+2]
        303             data[3] =0x4e;
  \     0xC6  0x214E                MOVS        R1,#+78
  \     0xC8  0xE022                B.N    ??CAN1_RX0_IRQHandler_4
        304             error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
        305             }
        306             else if (message == 6) {
  \             ??CAN1_RX0_IRQHandler_8: (+1)
  \     0xCA  0x2906                CMP   R1,#+6
  \     0xCC  0xD109                BNE.N       ??CAN1_RX0_IRQHandler_9
        307             TxHeader.StdId = 0x440;
  \     0xCE  0x603A                STR   R2,[R7, #+0]
        308             data[2] =0xba;
  \     0xD0  0x20BA                MOVS        R0,#+186
  \     0xD2  0x.... 0x....   BL      ?Subroutine1
        309             error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
  \             ??CrossCallReturnLabel_2: (+1)
  \     0xD6  0x6028        STR   R0,[R5, #+0]
        310             TxHeader.StdId = 0x316;
  \     0xD8  0x603E                STR   R6,[R7, #+0]
        311             data[2] =0xc0;
  \     0xDA  0x20C0                MOVS        R0,#+192
  \     0xDC  0x70A0                STRB R0,[R4, #+2]
        312             data[3] =0x5d;
  \     0xDE  0x215D                MOVS        R1,#+93
  \     0xE0  0xE016                B.N    ??CAN1_RX0_IRQHandler_4
        313             error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
        314             }
        315             else if (message == 7) {
  \             ??CAN1_RX0_IRQHandler_9: (+1)
  \     0xE2  0x2907        CMP   R1,#+7
  \     0xE4  0xD109                BNE.N       ??CAN1_RX0_IRQHandler_10
        316             TxHeader.StdId = 0x440;
  \     0xE6  0x603A                STR   R2,[R7, #+0]
        317             data[2] =0xd9;
```

```
  \      0xE8   0x20D9              MOVS     R0,#+217
  \      0xEA   0x.... 0x....       BL       ?Subroutine1
         318              error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
  \              ??CrossCallReturnLabel_1: (+1)
  \      0xEE   0x6028              STR      R0,[R5, #+0]
         319              TxHeader.StdId = 0x316;
  \      0xF0   0x603E              STR      R6,[R7, #+0]
         320              data[2] =0x60;
  \      0xF2   0x2060              MOVS     R0,#+96
  \      0xF4   0x70A0              STRB     R0,[R4, #+2]
         321              data[3] =0x6d;
  \      0xF6   0x216D              MOVS     R1,#+109
  \      0xF8   0xE00A              B.N      ??CAN1_RX0_IRQHandler_4
         322              error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
         323              }
         324              else if (message == 8) {
  \              ??CAN1_RX0_IRQHandler_10: (+1)
  \      0xFA   0x2908              CMP      R1,#+8
  \      0xFC   0xD10F              BNE.N    ??CAN1_RX0_IRQHandler_11
         325              TxHeader.StdId = 0x440;
  \      0xFE   0x603A              STR      R2,[R7, #+0]
         326              data[2] =0xfa;
  \      0x100  0x20FA              MOVS     R0,#+250
  \      0x102  0x.... 0x....       BL       ?Subroutine1
         327              error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
  \              ??CrossCallReturnLabel_0: (+1)
  \      0x106  0x6028              STR      R0,[R5, #+0]
         328              TxHeader.StdId = 0x316;
  \      0x108  0x603E              STR      R6,[R7, #+0]
         329              data[2] =0x00;
  \      0x10A  0x2000              MOVS     R0,#+0
  \      0x10C  0x70A0              STRB     R0,[R4, #+2]
         330              data[3] =0x7d;
  \      0x10E  0x217D              MOVS     R1,#+125
  \              ??CAN1_RX0_IRQHandler_4: (+1)
  \      0x110  0x70E1              STRB     R1,[R4, #+3]
```

```
      331             error = HAL_CAN_AddTxMessage(&hcan1, &TxHeader, data,
&mailbox);
  \             ??CAN1_RX0_IRQHandler_2: (+1)
  \     0x112  0x464B             MOV   R3,R9
  \     0x114  0x4622             MOV   R2,R4
  \     0x116  0x4639             MOV   R1,R7
  \     0x118  0x4640             MOV   R0,R8
  \     0x11A  0x.... 0x....  BL      HAL_CAN_AddTxMessage
  \             ??CAN1_RX0_IRQHandler_11: (+1)
  \     0x11E  0x6028             STR   R0,[R5, #+0]
      332             }
      333             }
      334             /* USER CODE END CAN1_RX0_IRQn 1 */
      335             }
  \             ??CAN1_RX0_IRQHandler_0: (+1)
  \     0x120  0xE8BD 0x83F1    POP   {R0,R4-R9,PC}

  \                             In section .text, align 2, keep-with-next
  \             ?Subroutine1: (+1)
  \     0x0  0x70A0         STRB R0,[R4, #+2]
  \     0x2  0x464B         MOV  R3,R9
  \     0x4  0x4622         MOV  R2,R4
  \     0x6  0x4639         MOV  R1,R7
  \     0x8  0x4640         MOV  R0,R8
  \     0xA  0x.... 0x....      B.W   HAL_CAN_AddTxMessage
      336
      337             /**
      338             * @brief This function handles CAN1 RX1 interrupt.
      339             */

  \                             In section .text, align 4, keep-with-next
      340             void CAN1_RX1_IRQHandler(void)
      341             {
      342             /* USER CODE BEGIN CAN1_RX1_IRQn 0 */
      343
      344             /* USER CODE END CAN1_RX1_IRQn 0 */
      345             HAL_CAN_IRQHandler(&hcan1);
  \             CAN1_RX1_IRQHandler: (+1)
  \     0x0  0xBF00         Nop
  \     0x2                 REQUIRE ?Subroutine0
```

```
\     0x2                   ;; // Fall through to label ?Subroutine0
      346           /* USER CODE BEGIN CAN1_RX1_IRQn 1 */
      347
      348           /* USER CODE END CAN1_RX1_IRQn 1 */
      349           }

\                           In section .text, align 4, keep-with-next
\           ??DataTable3:
\     0x0   0x....'....     DC32 message

\                           In section .text, align 4, keep-with-next
\           ??DataTable3_1:
\     0x0   0x....'....     DC32 hcan1

\                           In section .text, align 4, keep-with-next
\           ??DataTable3_2:
\     0x0   0x....'....     DC32 RxHeader

\                           In section .text, align 4, keep-with-next
\           ??DataTable3_3:
\     0x0   0x....'....     DC32 datarx

\                           In section .text, align 4, keep-with-next
\           ??DataTable3_4:
\     0x0   0x....'....     DC32 error

\                           In section .text, align 4, keep-with-next
\           ??DataTable3_5:
\     0x0   0x....'....     DC32 mailbox

\                           In section .text, align 4, keep-with-next
\           ??DataTable3_6:
\     0x0   0x....'....     DC32 `data`

\                           In section .text, align 4, keep-with-next
\           ??DataTable3_7:
\     0x0   0x....'....     DC32 TxHeader
      350
      351           /* USER CODE BEGIN 1 */
      352
```

```
      353             /* USER CODE END 1 */
```

Maximum stack usage in bytes:

```
.cstack Function
------- --------
      0   BusFault_Handler
     32   CAN1_RX0_IRQHandler
     32   -> HAL_CAN_AddTxMessage
     32   -> HAL_CAN_GetRxMessage
     32   -> HAL_CAN_IRQHandler
      0   CAN1_RX1_IRQHandler
      0   -> HAL_CAN_IRQHandler
      0   CAN1_TX_IRQHandler
      0   -> HAL_CAN_IRQHandler
      0   DebugMon_Handler
      8   EXTI0_IRQHandler
      8   -> HAL_GPIO_EXTI_IRQHandler
      0   HardFault_Handler
      0   MemManage_Handler
      0   NMI_Handler
      0   PendSV_Handler
      0   SVC_Handler
      0   SysTick_Handler
      0   -> HAL_IncTick
      0   UsageFault_Handler
```

Section sizes:

```
Bytes  Function/Label
-----  --------------
      4  ??DataTable3
      4  ??DataTable3_1
      4  ??DataTable3_2
      4  ??DataTable3_3
      4  ??DataTable3_4
      4  ??DataTable3_5
      4  ??DataTable3_6
      4  ??DataTable3_7
```

```
   6  ?Subroutine0
  14  ?Subroutine1
   2  BusFault_Handler
 292  CAN1_RX0_IRQHandler
   2  CAN1_RX1_IRQHandler
   2  CAN1_TX_IRQHandler
   2  DebugMon_Handler
  24  EXTI0_IRQHandler
   2  HardFault_Handler
   2  MemManage_Handler
   2  NMI_Handler
   2  PendSV_Handler
   2  SVC_Handler
   4  SysTick_Handler
   2  UsageFault_Handler
   1  message


   1 byte  in section .bss
 392 bytes in section .text

 392 bytes of CODE memory
   1 byte  of DATA memory

Errors: none
Warnings: none
```