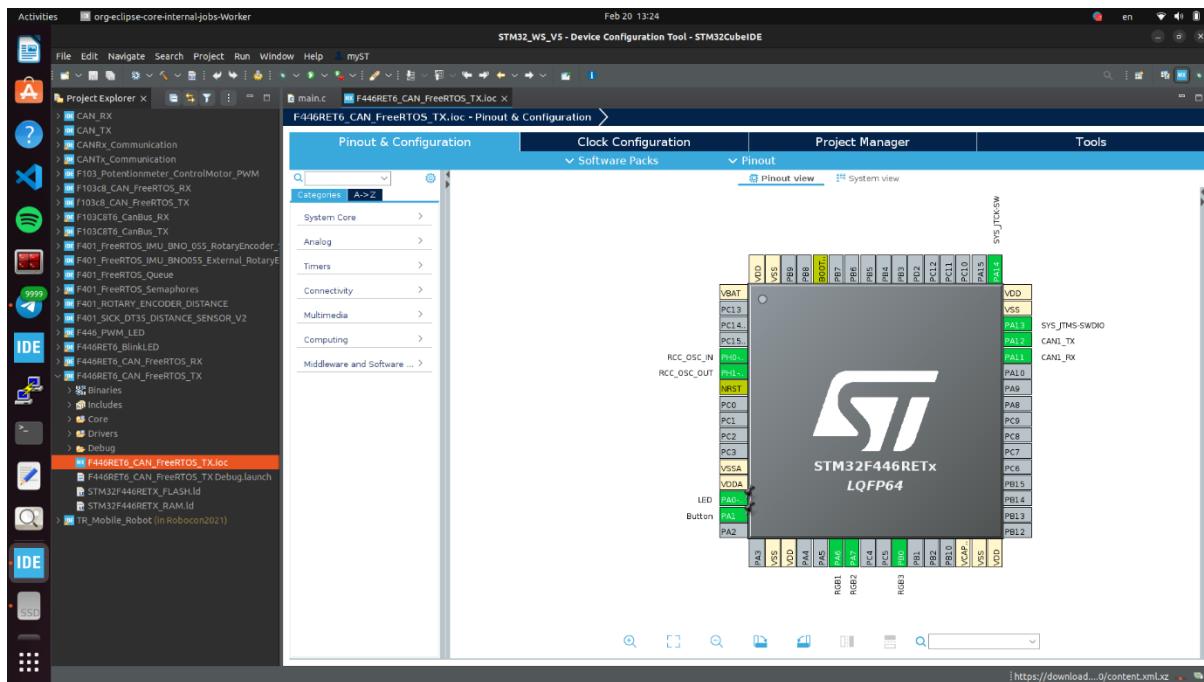
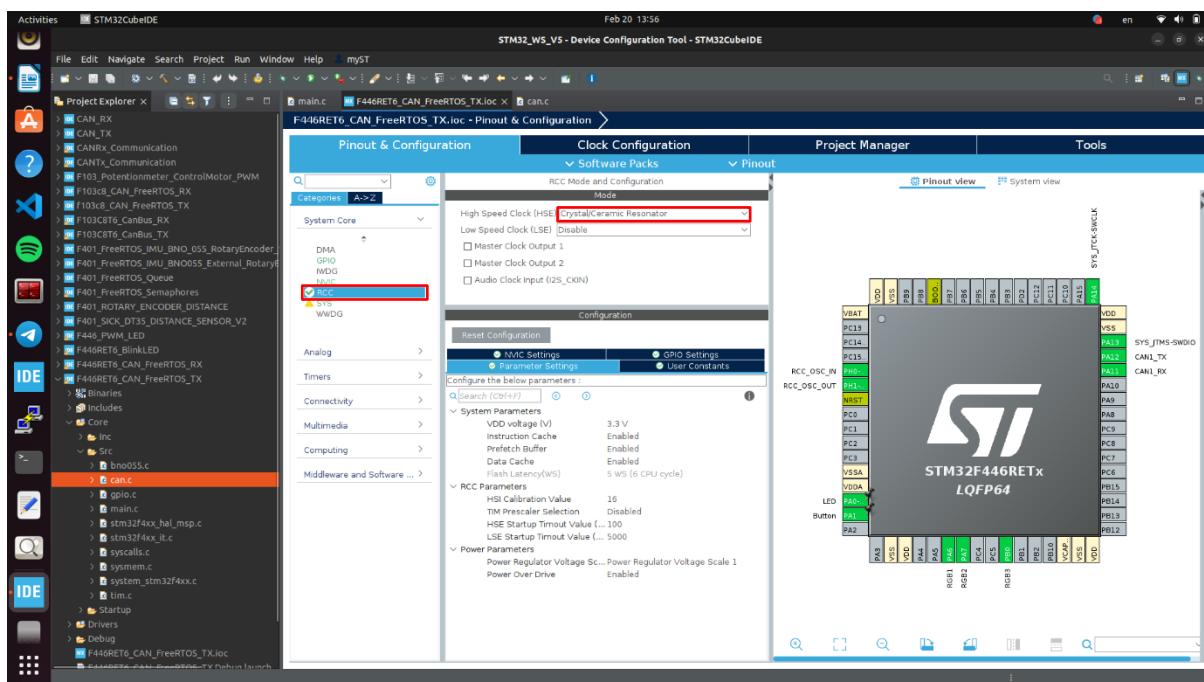


# CAN BUS STM32 Tx (F446RE Nucleo) Rx (F446RE Nucleo)

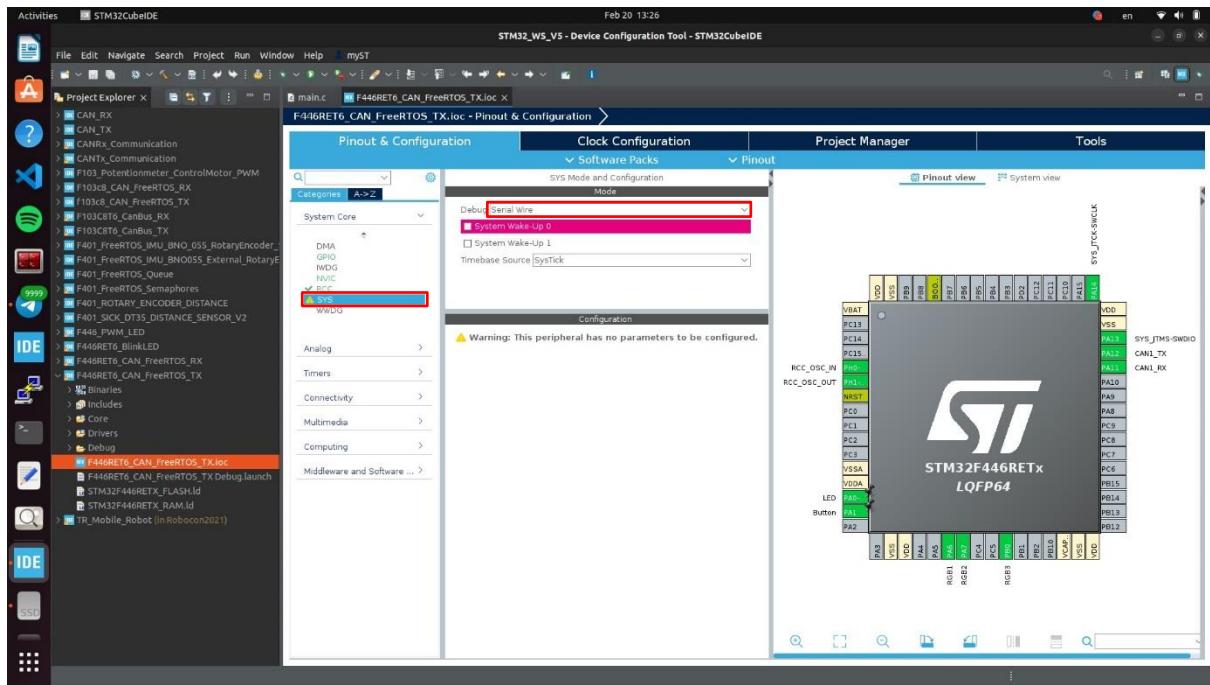
## I. Transmitter\_Tx (F446RE Nucleo)



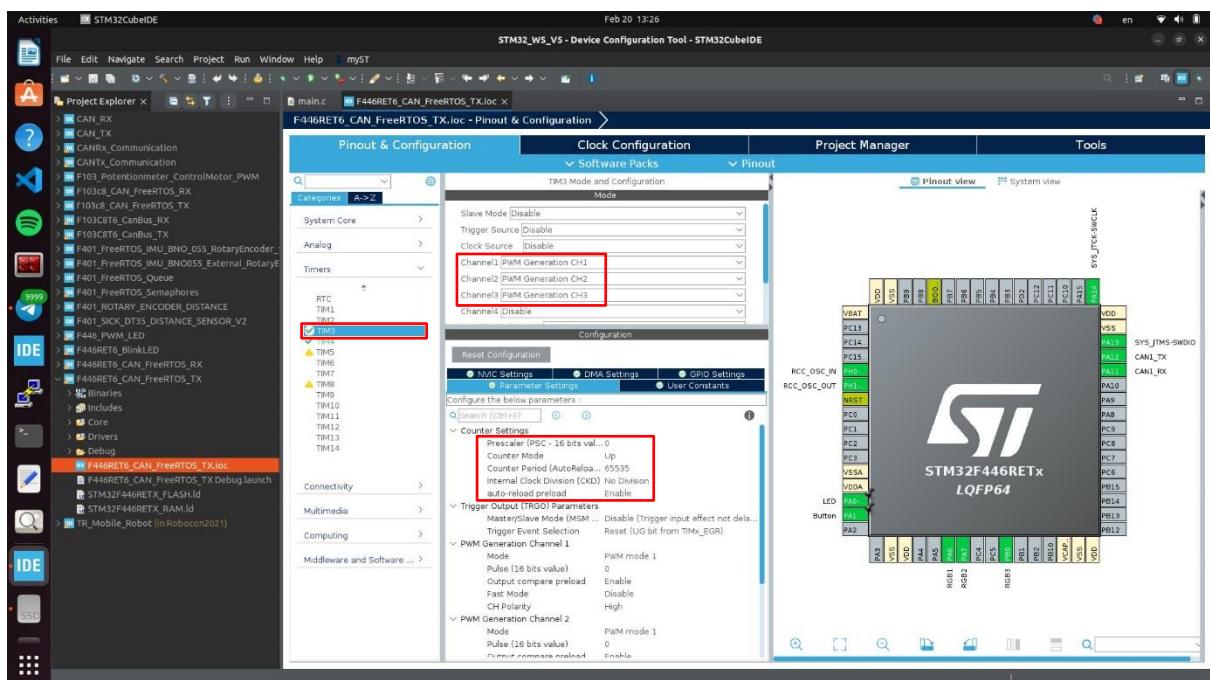
### 1. Configuration RCC



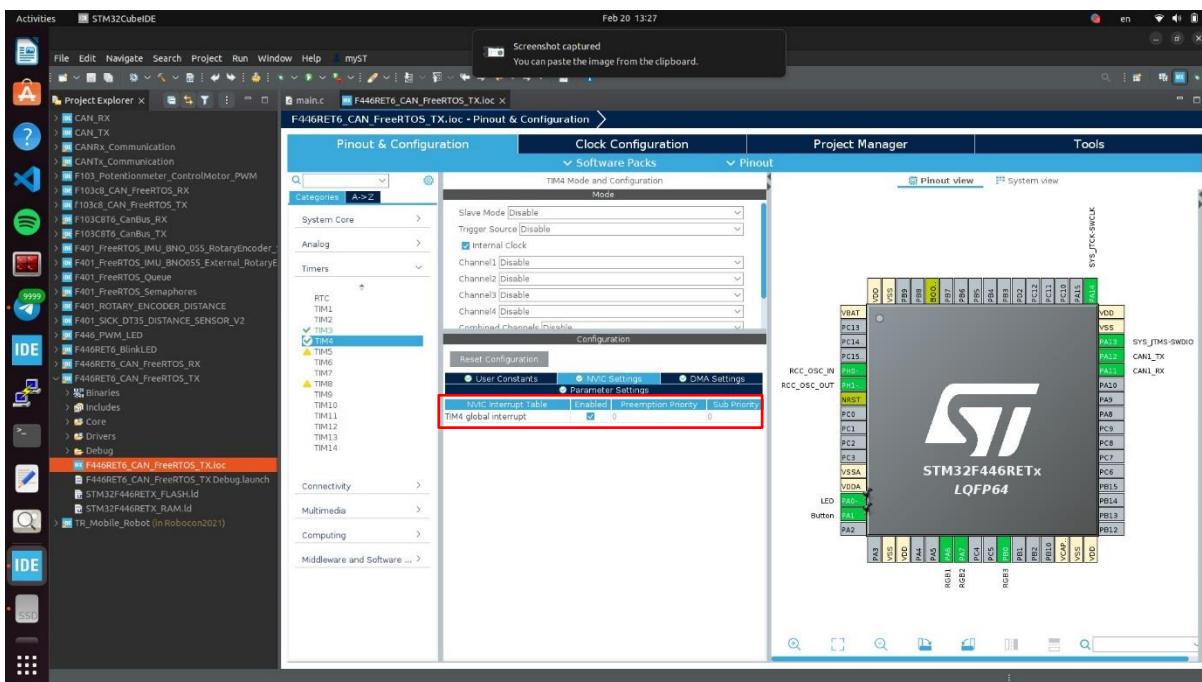
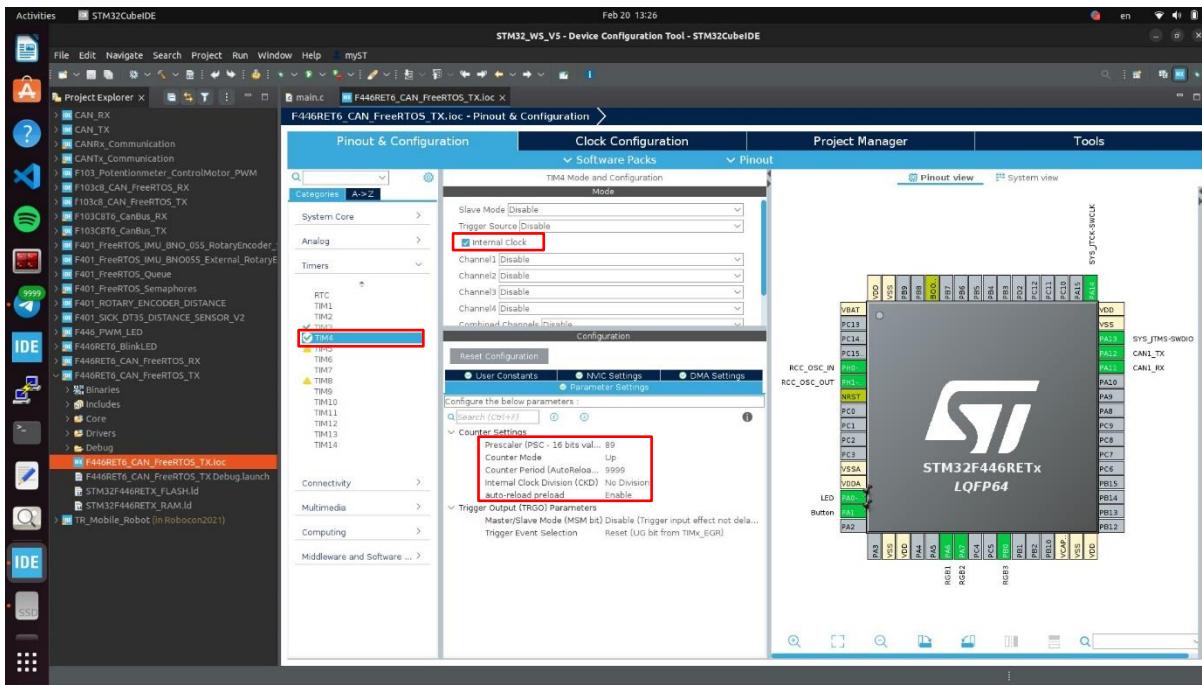
## 2. Configuration SYS



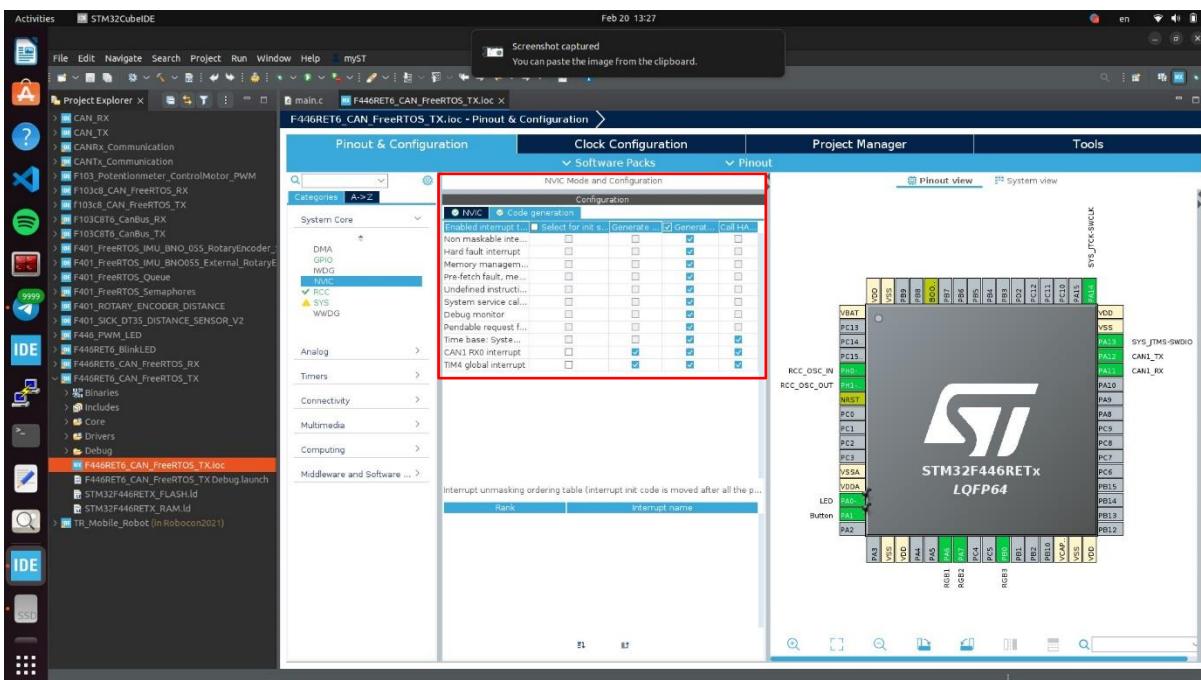
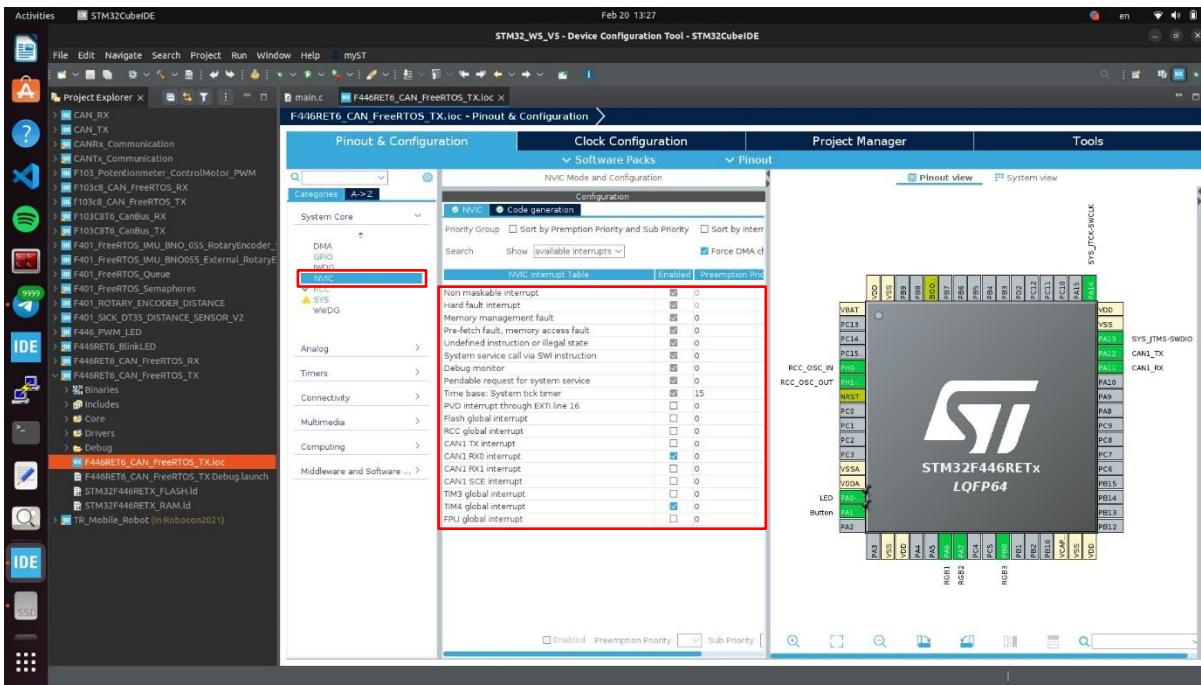
### 3. Configuration TIM3 PWM LED



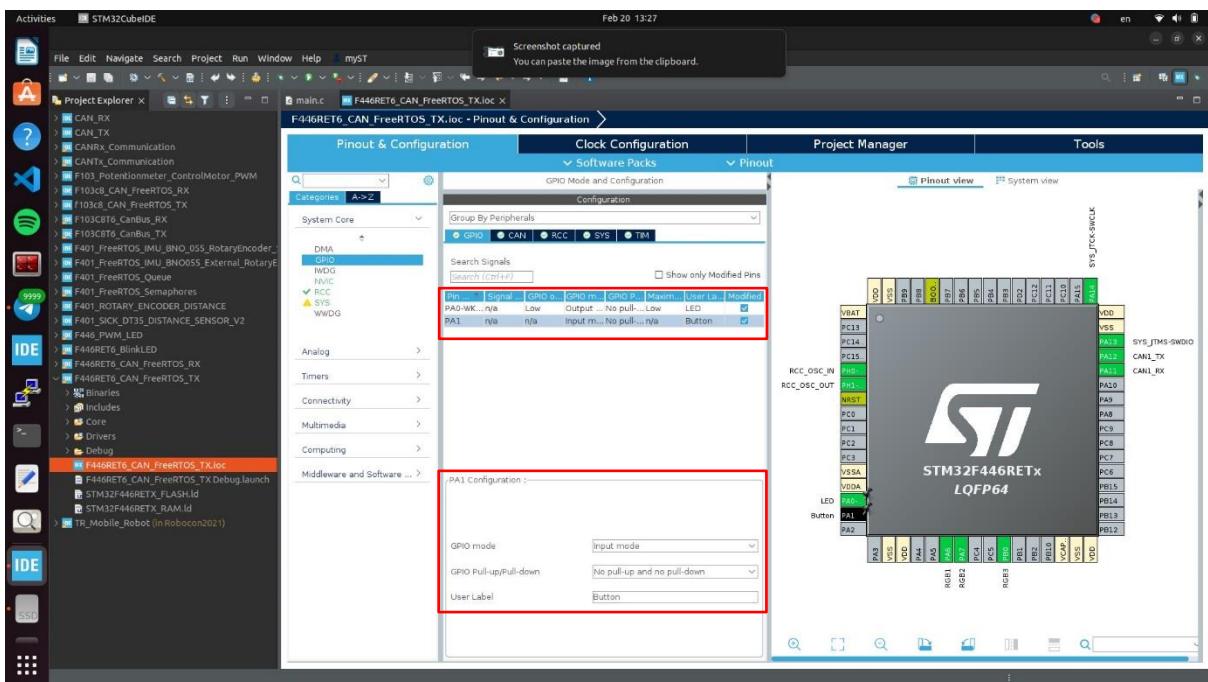
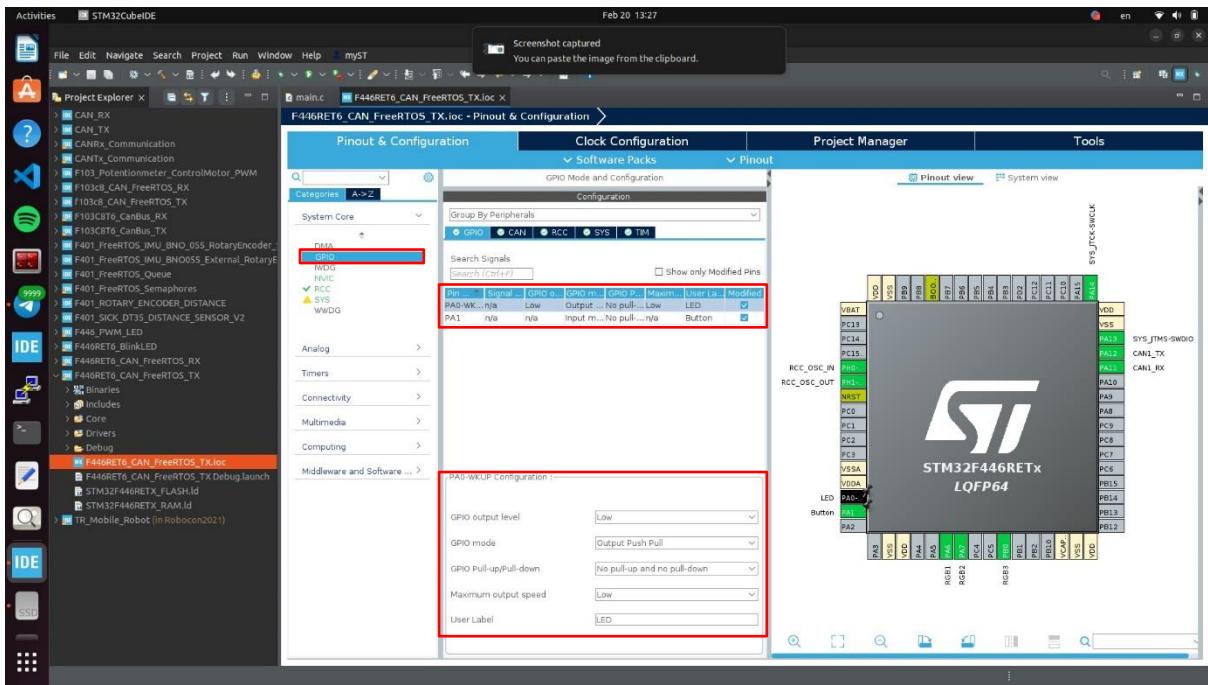
## 4. Configuration TIM4 (Time Base)



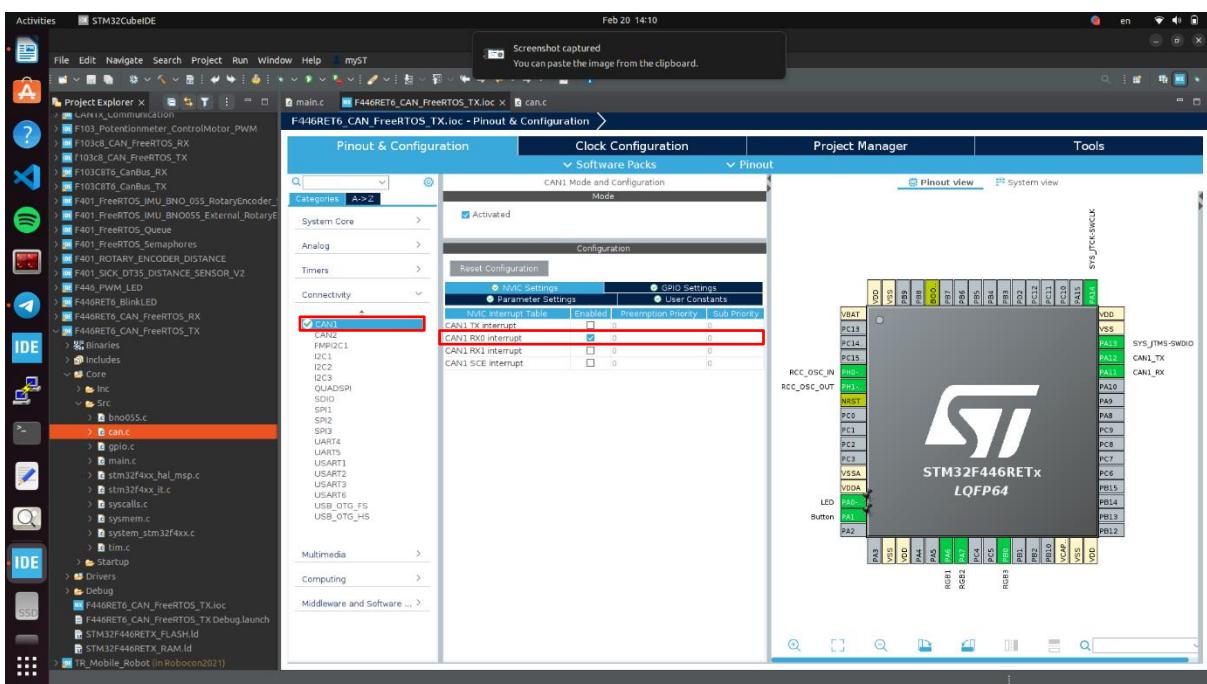
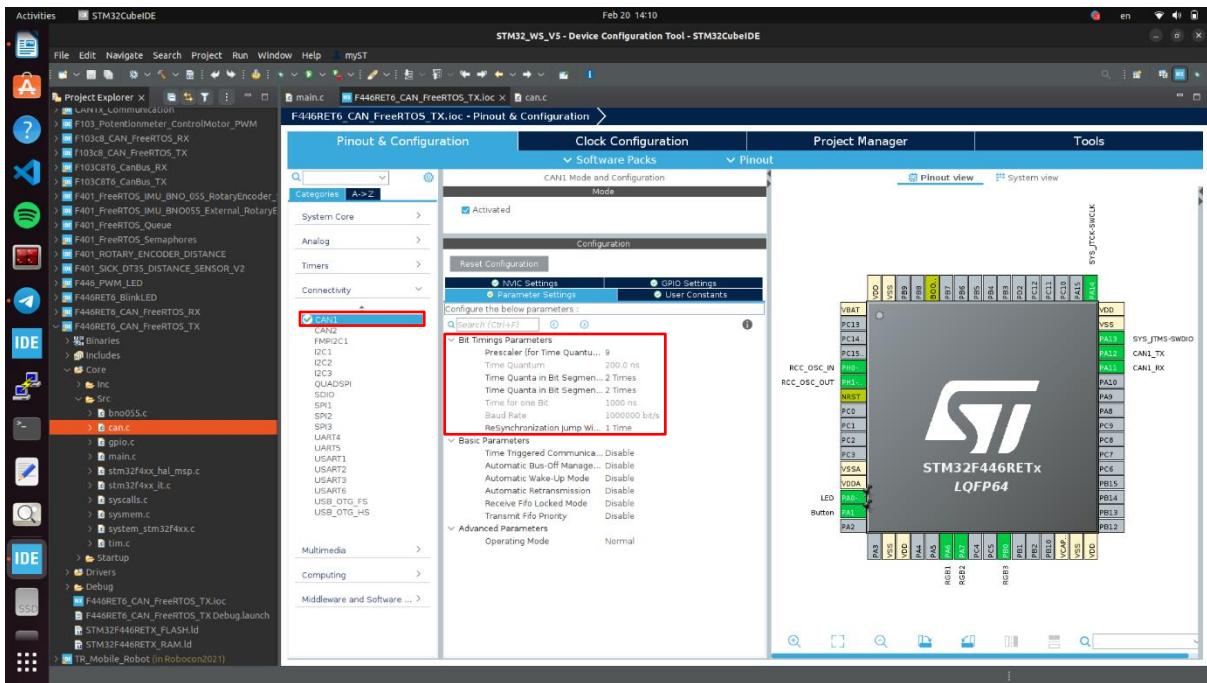
## 5. Configuration NVIC



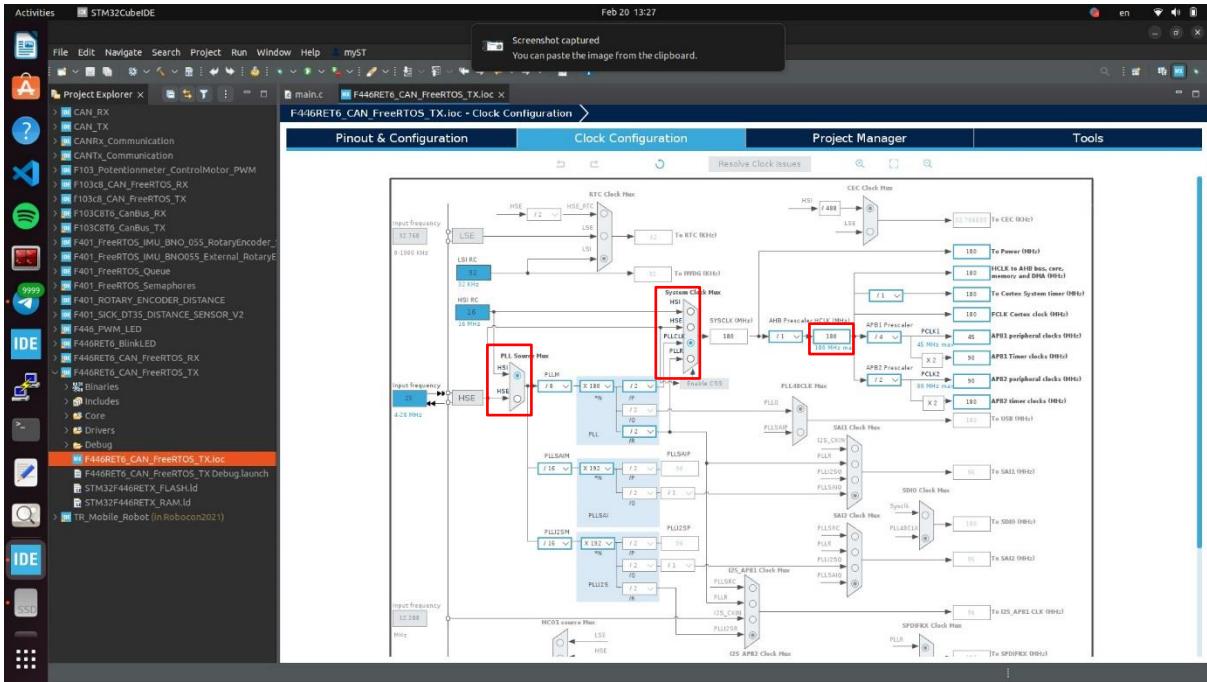
## 6. Configuration GPIO



## 7. Configuration CAN



## 8. Configuration CLOCK



## 9. Code in main.c (Tx)

The screenshot shows the STM32CubeIDE IDE with the file 'main.c' open for the project 'F446RET6\_CAN\_FreeRTOS\_TX.loc'. The code is as follows:

```
16
17 // USER CODE END Header */
18 /* Includes */
19 #include "main.h"
20 #include "can.h"
21 #include "tim.h"
22 #include "spi.h"
23 #include "usart.h"
24
25 /* Private includes */
26 /* USER CODE BEGIN Includes */
27
28 #include "bno055_stm32.h"
29 #include "bno055.h"
30
31 /* USER CODE END Includes */
32
33 /* Private typedef */
34 /* USER CODE BEGIN PTD */
35
36 /* Private macro */
37
38 /* USER CODE END PTD */
39
40 /* Private define */
41 /* USER CODE BEGIN PD */
42
43 /* USER CODE END PD */
44
45 /* Private macro */
46 /* USER CODE BEGIN PM */
47
48 /* USER CODE END PM */
49
50 /* Private variables */
51
52 /* USER CODE BEGIN PV */
53
54 /* CAN */
55
56     CAN_RxHeaderTypeDef RxHeader;
57     uint8_t RxData[8];
58     CAN_TxHeaderTypeDef TxHeader;
59     uint8_t TxData[8];
60     uint32_t TxMailbox;
61     uint16_t ReadValue;
62
```

The code is mostly empty, with comments indicating the start and end of user-defined sections for headers, includes, defines, macros, and variables.

The screenshot shows the STM32CubeIDE interface with the following details:

- Title Bar:** STM32CubeIDE - STM32\_WS\_VS - F446RET6\_CAN\_FreeRTOS\_TX.Core/Src/main.c - STM32CubeIDE
- File Explorer (Project Explorer):** Shows the project structure with files like CAN\_RX, CAN\_TX, CANRxCCommunication, CANTxCommunication, F103\_Potentiometer\_ControlMotor\_PWM, F103KB\_CAN\_FreeRTOS\_RX, F103KB\_CAN\_FreeRTOS\_TX, F103C8T6\_CanBus\_RX, F401\_FreeRTOS\_IMU\_BNO\_055\_RotationCode, F401\_FreeRTOS\_IMU\_BNO055\_External\_Rotation, F401\_FreeRTOS\_Queue, F401\_FreeRTOS\_Semaphores, F401\_ROTARY\_ENCODER\_DISTANCE, F401\_SICK\_DTS5\_DISTANCE\_SENSOR\_V2, F446\_PWM\_LED, F446RET6\_BlinkLED, F446RET6\_CAN\_FreeRTOS\_RX, F446RET6\_CAN\_FreeRTOS\_TX.
- Code Editor:** The main.c file is open, showing code related to CAN communication and sensor data processing. It includes definitions for RxHeader and TxHeader structures, and variables for IMU data (BNO055), rotation encoder distance, and Sick DTS5 distance sensor. It also handles PWM LED control and CAN message transmission.
- Toolbars:** Standard file operations (File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help) and specific STM32 tools.
- Status Bar:** Displays the build status: F446RET6\_CAN\_FreeRTOS\_TX.elf - /F446RET6\_CAN\_FreeRTOS\_TX.Debug - Feb 20, 2024, 11:28:34 AM.

```
File Edit Source Refactor Navigate Project Run Window Help myST

Project Explorer x
CAN_RX
CAN_TX
CANRx Communication
CANTx Communication
F103_Potentiometer_ControlMotor_PWM
F103K_CAN_FreeRTOS_RX
F103K_CAN_FreeRTOS_TX
F103K_CAN_Canbus_RX
F103K_CAN_Canbus_TX
F401_FrertOS_IMU_BNO_055_RotationEncoder
F401_FrertOS_IMU_BNO_055_Extrernal_Rotation
F401_FrertOS_Queue
F401_ROTARY_ENCODER_DISTANCE
F401_SICK_D35_DISTANCE_SENSOR_V2
F446_PWM_LED
F446RET6_BLINKED
F446RET6_CAN_FreeRTOS_RX
F446RET6_CAN_FreeRTOS_TX
  Binarie
  Includes
  Core
  Drivers
  Debug
  F446RET6_CAN_FreeRTOS_TX
    F446RET6_CAN_FreeRTOS_TX.launch
    STM32F446RET6_FLASH.ld
    STM32F446RET6_RAM.ld
    TR_Mobile_Robot [in Robocore2021]
Build Analyzer x Static Stack Analyzer: Cyclomatic Complexity
F446RET6_CAN_FreeRTOS_TX.elf - /F446RET6_CAN_FreeRTOS_TX/Debug - Feb 20, 2024, 11:28:34 AM
Memory Regions Memory Details
```

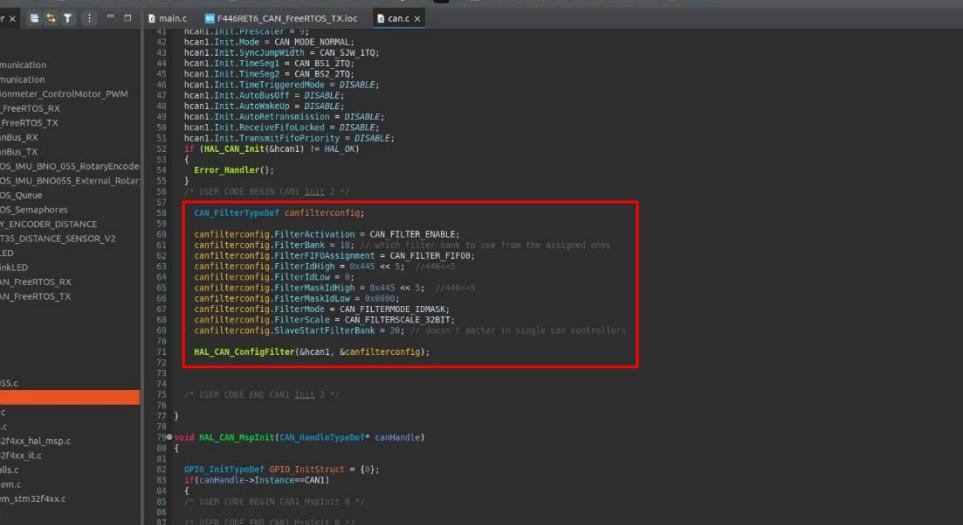
The screenshot shows the STM32CubeIDE interface with the following details:

- Title Bar:** Activities - STM32CubeIDE - STM32\_WS\_VS - F446RET6\_CAN\_FreeRTOS\_Tx/Core/Src/main.c - STM32CubeIDE
- File Explorer (Project Explorer):** Shows the project structure with files like main.c, HAL\_CAN\_Start(), HAL\_TIM\_Base\_Start\_IT(), and various sensor and motor driver functions.
- Code Editor:** Displays the main.c source code for the F446RET6\_CAN\_FreeRTOS\_Tx project. The code includes initialization of CAN, TIM, and USART peripherals, as well as handling of various sensors (IMU, BNO055, Sick, Encoder) and actuators (Motor PWM, RGB LED). It also includes a main loop with infinite loops and user-defined sections.
- Toolbars:** Standard IDE toolbars for file operations, search, and navigation.
- Status Bar:** Shows the date and time (Feb 20, 2024, 11:28:34 AM).

The screenshot shows the STM32CubeIDE interface with the following details:

- Title Bar:** Activities STM32CubeIDE, ST32CubeIDE, Feb 20 13:28
- File Menu:** File Edit Source Refactor Navigate Search Project Run Window Help myST
- Project Explorer:** CAN\_RX, CAN\_TX, CANRx Communication, CANTx Communication, F103\_Potentiometer\_ControlMotor\_PWM, F103K\_CAN\_FreeRTOS\_RX, F103K\_CAN\_FreeRTOS\_TX, F103C816\_Canbus\_RX, F103C816\_Canbus\_TX, F401\_FreeRTOS\_IMU\_BNO\_055\_RotationEncoder, F401\_FreeRTOS\_IMU\_BNO055\_External\_Rotation, F401\_FreeRTOS\_Queue, F401\_FreeRTOS\_Semaphores, F401\_ROTARY\_ENCODER\_DISTANCE, F401\_SICK\_D35\_DISTANCE\_SENSOR\_V2, F446\_PWM\_LED, F446RET6\_BlinkLED, F446RET6\_CAN\_FreeRTOS\_RX, F446RET6\_CAN\_FreeRTOS\_TX.
- Code Editor:** The main.c file is open, showing code for F446RET6\_CAN\_FreeRTOS\_Tx. The code includes definitions for CAN\_RX and CAN\_TX, and handles for HAL\_RCC\_ClockConfig, HAL\_GPIO\_Init, HAL\_CAN\_AddTxMessage, and HAL\_TIM\_PeriodElapsedCallback. It also includes comments for USER CODE BEGIN and USER CODE END.
- Build Analyzer:** A static stack analyzer report for F446RET6\_CAN\_FreeRTOS\_Tx.elf is displayed, showing cyclomatic complexity.

#### 10. Code can.c in (Tx)



The screenshot shows the STM32CubeIDE interface with the project **STM32\_W5\_V5 - F446RET6\_CAN\_FreeRTOS\_TX/Core/Src/can.c**. The code editor displays the **main.c** file, which includes the following code snippet:

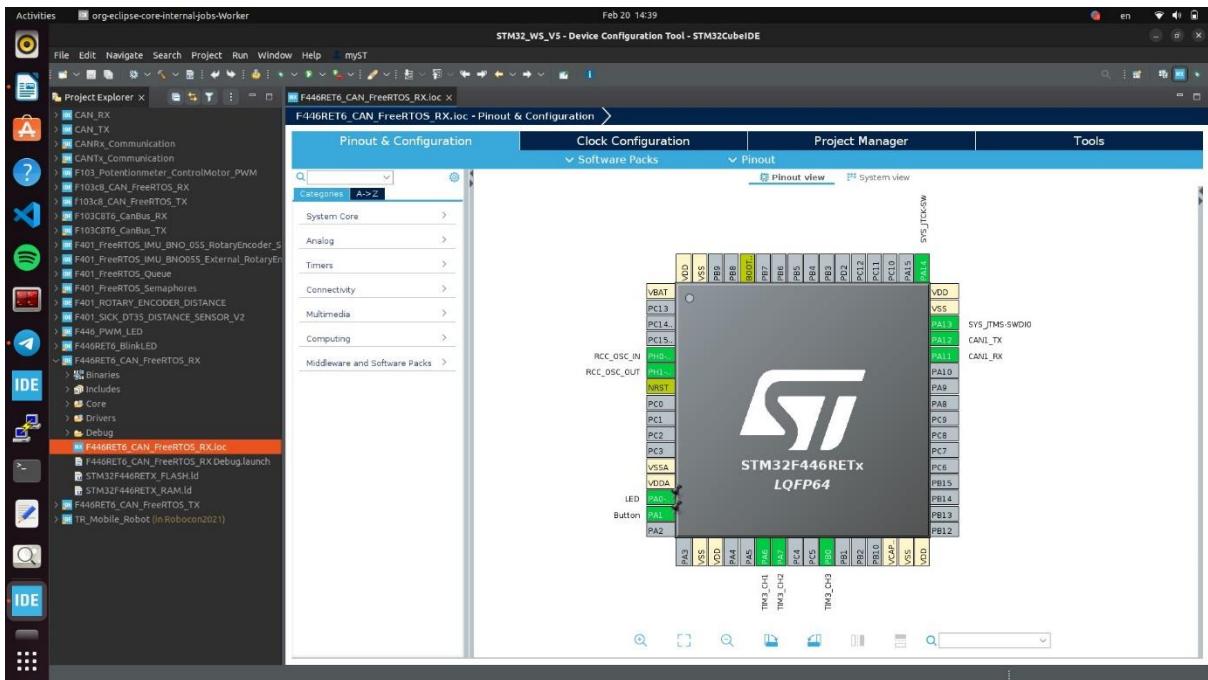
```
HAL_CAN_ConfigFilter(&hcan1, &canfilterconfig);

void HAL_CAN_MspInit(CAN_HandleTypeDef* canHandle)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};
    if(canHandle->Instance==CAN1)
    {
        /* USER CODE BEGIN CAN1_MspInit 0 */
        /* USER CODE END CAN1_MspInit 0 */
    }
}
```

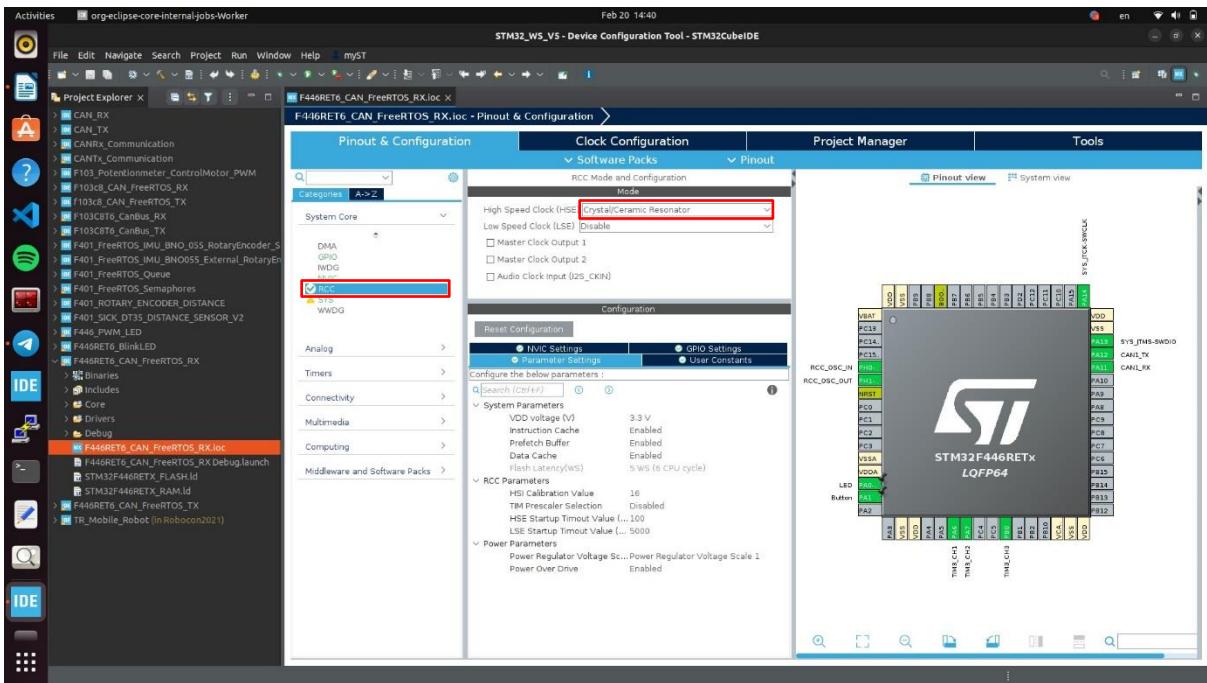
A red box highlights the **canfilterconfig** structure definition:

```
CAN_FilterTypeDef canfilterconfig;
canfilterconfig.FilterActivation = CAN_FILTER_ENABLE;
canfilterconfig.FilterBank = 18; // Which filter bank to use from the assigned ones
canfilterconfig.FilterIDAssignment = CAN_FILTER_IDFOB;
canfilterconfig.FilterMaskIdHigh = 0x445 << 5; // 446<<5
canfilterconfig.FilterMaskIdLow = 0x0000;
canfilterconfig.FilterMode = CAN_FILTERMODE_IDMASK;
canfilterconfig.FilterScale = CAN_FILTERSCALE_32BIT;
canfilterconfig.SlaveMatchFilterBank = 0; // Not matter in single can controllers
```

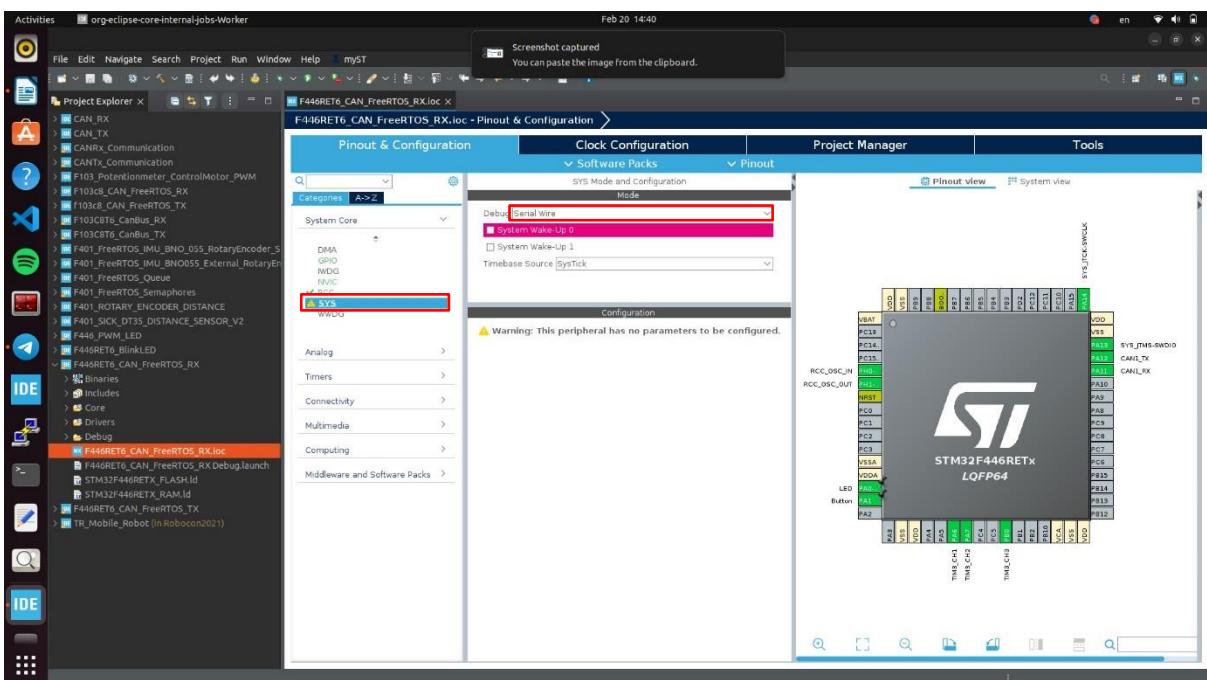
## **II. Receiver\_Rx (F446RE Nucleo)**



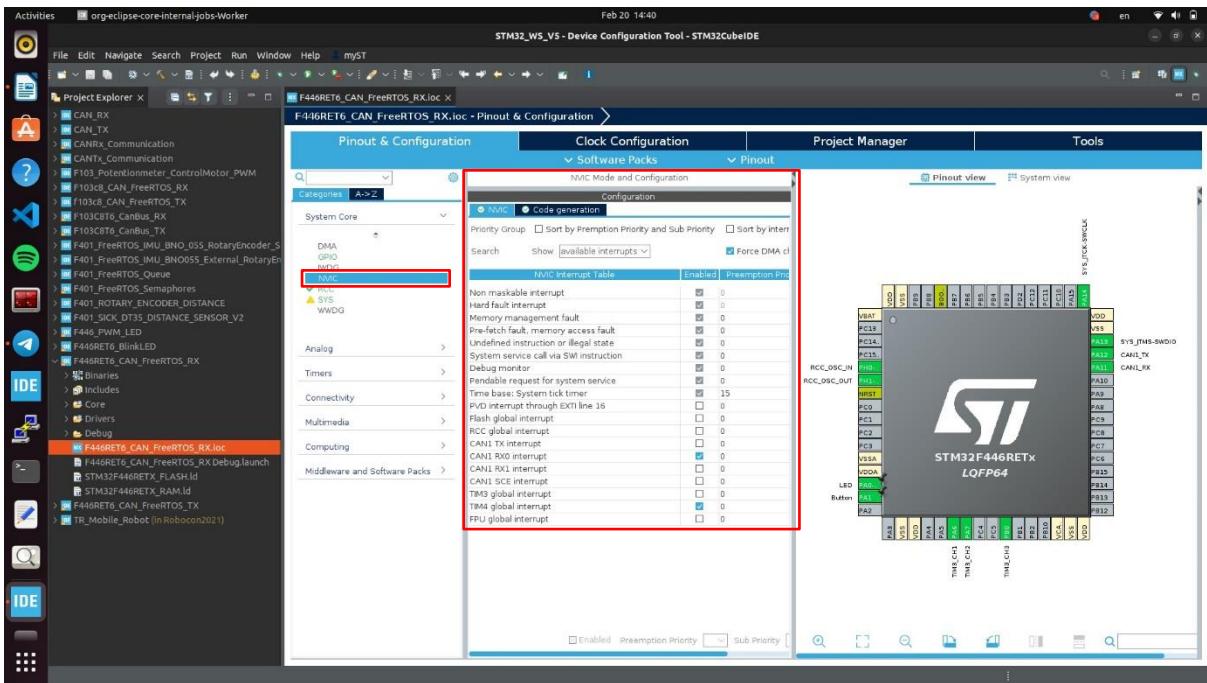
## 1. Configuration RCC



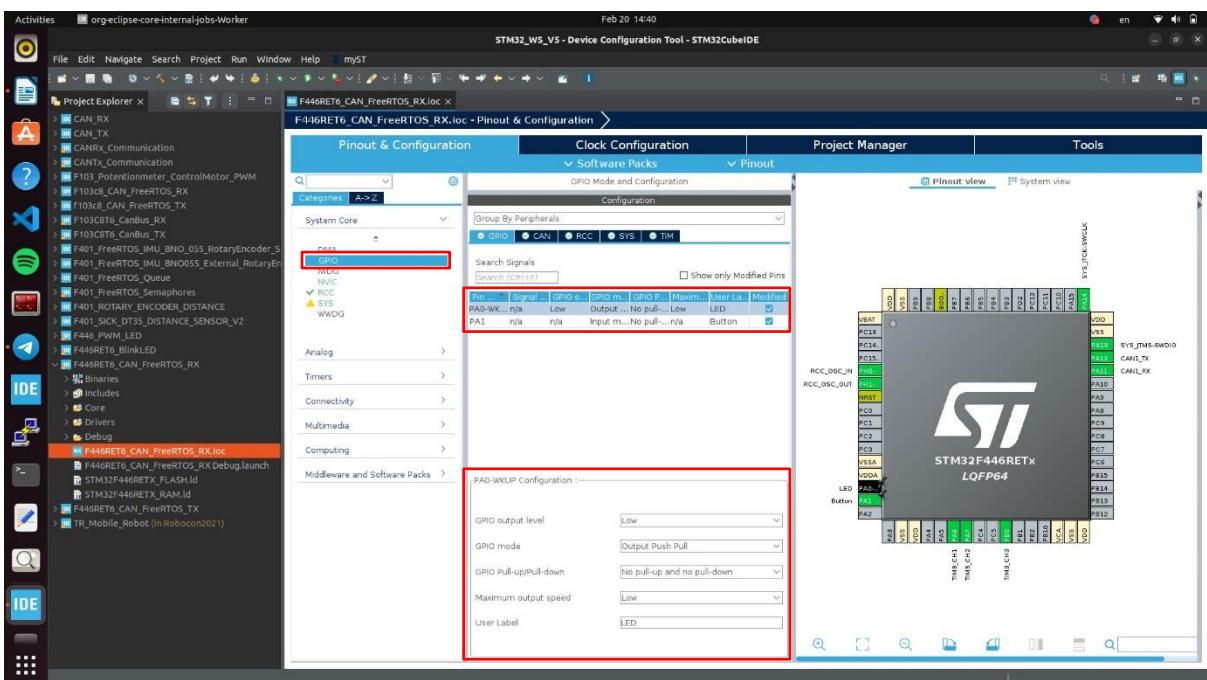
## 2. Configuration SYS

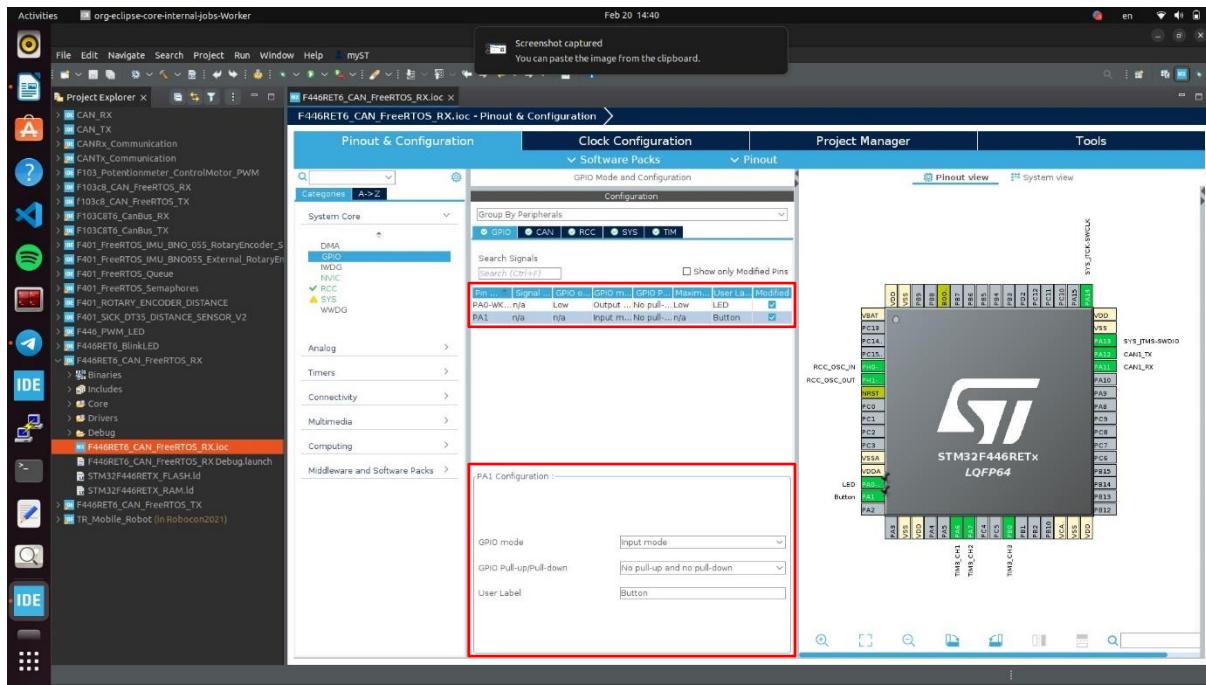


### 3. Configuration NVIC

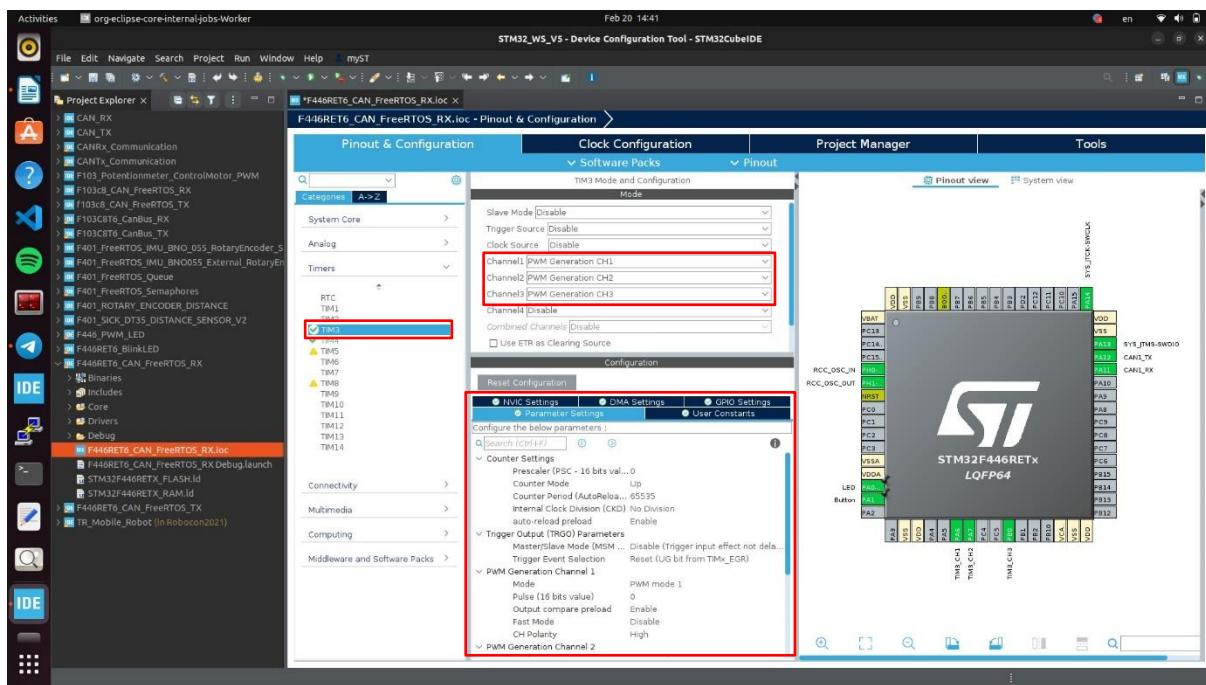


### 4. Configuration GPIO

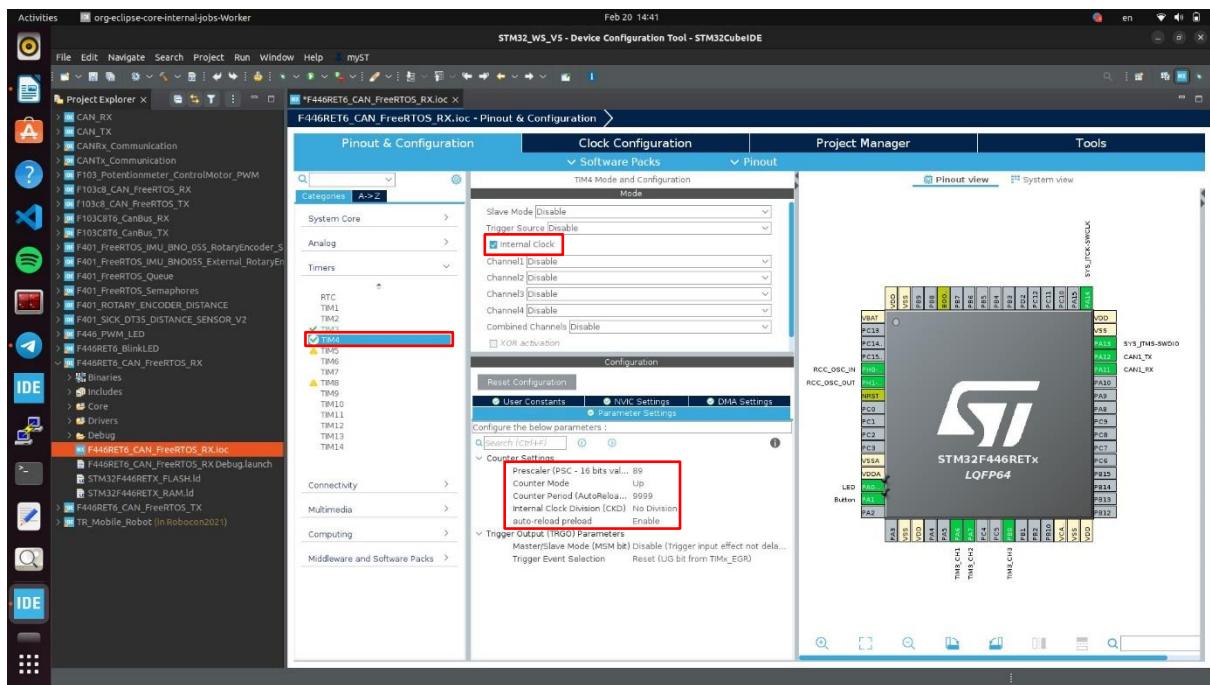




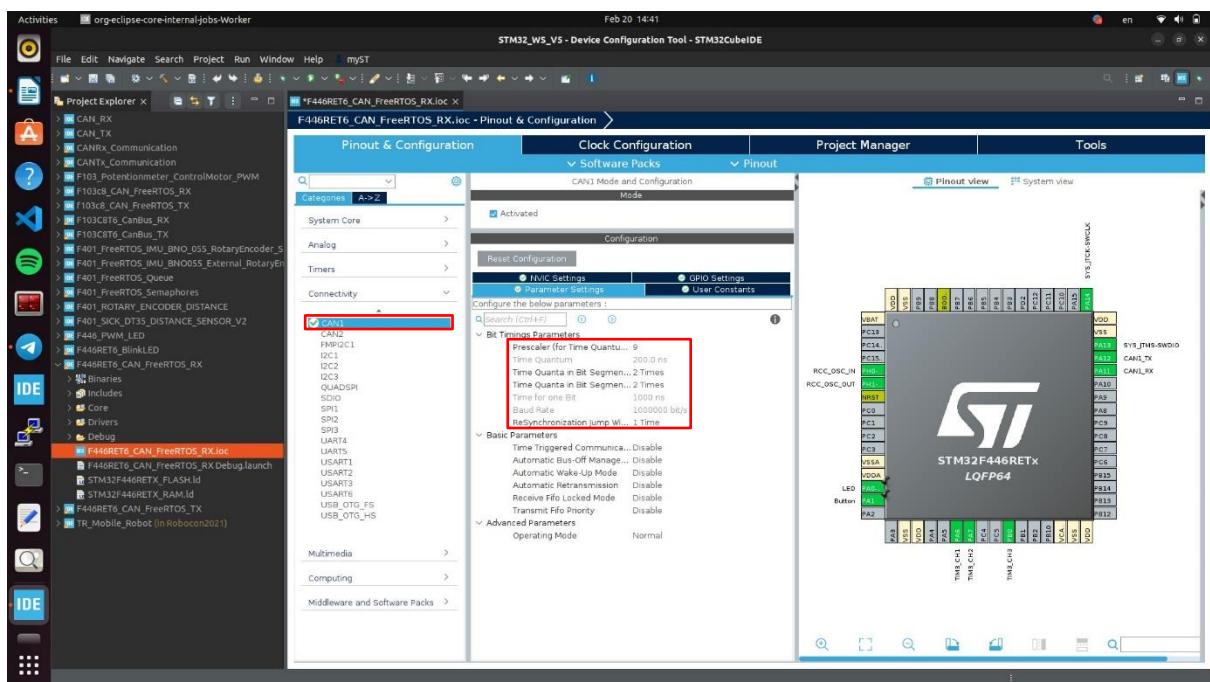
## 5. Configuration TIM3 PWM LED



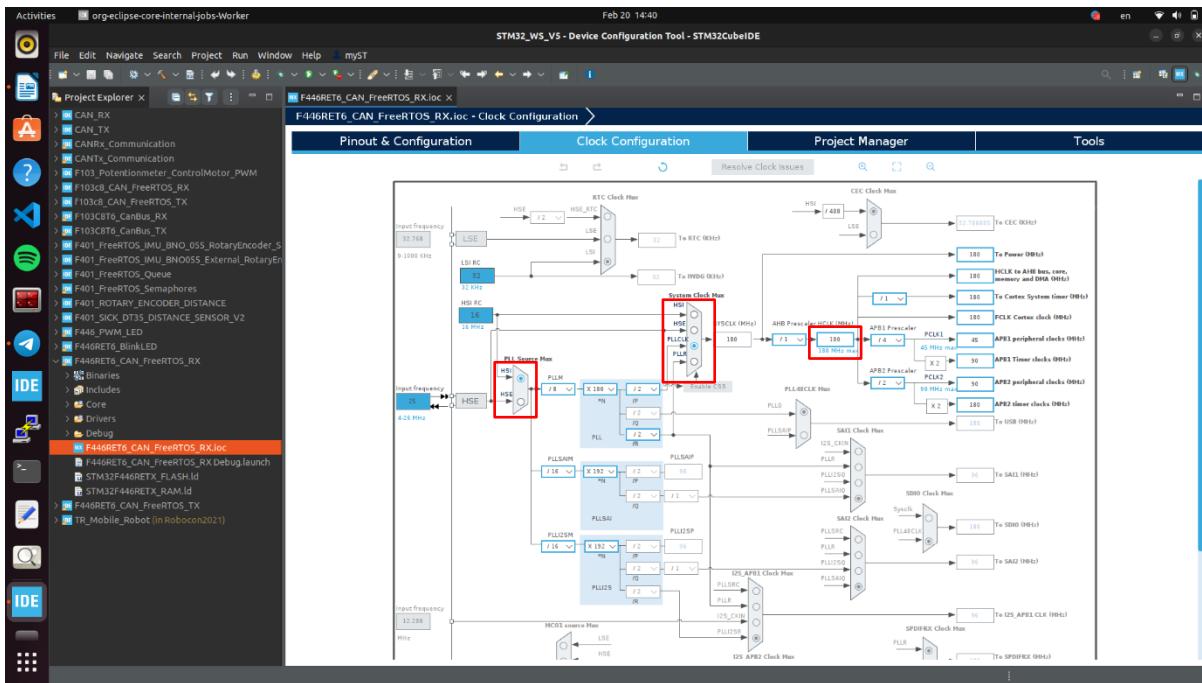
## 6. Configuration TIM4 (Time base)



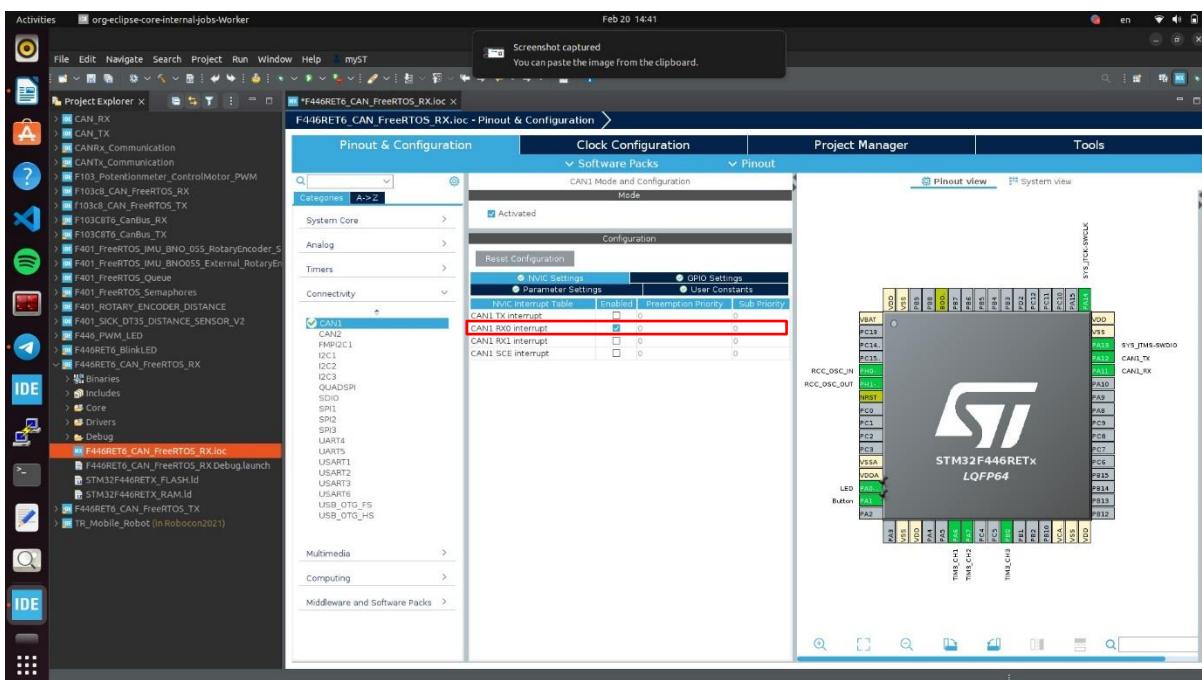
## 7. Configuration CAN



## 8. Configuration CLOCK



## 9. Configuration CAN



## 10. Code Rx main.c

The screenshot shows the STM32CubeIDE interface. The Project Explorer on the left lists various STM32 peripheral drivers and application components. The main workspace displays the source code for `F446RET6_CAN_FreeRTOS_RX.c`. The code includes definitions for CAN\_RX, CAN\_TX, and CAN\_Rx, along with various sensor and motor driver functions. The Build Analyzer at the bottom provides static analysis metrics like stack usage and cyclomatic complexity.

```
File Edit Source Refactor Search Project Run Window Help myST
```

STM32\_WS\_V5 - F446RET6\_CAN\_FreeRTOS\_RX/Core/Src/main.c - STM32CubeIDE

Project Explorer

main.c x

```
16
17
18 /* USER CODE END Header */
19
20 #include "main.h"
21 #include "can.h"
22 #include "tim.h"
23 #include "gpio.h"
24
25 /* Private includes */
26 /* USER CODE BEGIN Includes */
27
28 /* USER CODE END Includes */
29
30 /* Private typedef */
31 /* USER CODE BEGIN PTD */
32
33 /* USER CODE END PTD */
34
35 /* Private define */
36 /* USER CODE BEGIN PD */
37 /* USER CODE END PD */
38
39 /* Private macro */
40 /* USER CODE BEGIN PM */
41
42 /* USER CODE END PM */
43
44 /* Private variables */
45
46 /* USER CODE BEGIN PV */
47
48 /* CAN */
49
50 #include "CanHeaderTypeDef.h"
51
52 CanHeaderTypeDef RxHeader;
53 CanHeaderTypeDef TxHeader;
54 uint8_t TxData[8];
55 uint32_t TxMailbox;
56 uint32_t ReadValue;
57
58 int datacheck = 0;
59 float pum_M1 = 0;
60 float pum_M2 = 0;
61
62 float RxData1 = 0;
```

Build Analyzer x Static Stack Analyzer x Cyclomatic Complexity

F446RET6\_CAN\_FreeRTOS\_RX.eif - /F446RET6\_CAN\_FreeRTOS\_RX/Debug - Feb 20, 2024, 11:05:56 AM

MemoryRegions MemoryDetails

Activities org.eclipse.core.internal.jobs.Worker

File Edit Source Refactor Navigate Search Project Run Window Help myst

STM32\_WS\_V5 - F446RET6\_CAN\_FreeRTOS\_RX/Core/Src/main.c - STM32CubeIDE

Project Explorer F446RET6\_CAN\_FreeRTOS\_RX.loc main.c

```
85  /* Private function prototypes */
86  void SystemClock_Config(void);
87  /* USER CODE BEGIN PFP */
88  /* MAP */
89  float map(Float Input, Float Min_Input, Float Max_Input, float Min_Output, float Max_Output) {
90      return (float) ((Input - Min_Input) * (Max_Output - Min_Output)) / (Max_Input - Min_Input) + Min_Output;
91  }
92
93  /* STM32 INTERRUPT RECEIVER FROM USB CAN */
94  void HAL_CAN_ReceiveMsgPendingCallback(CAN_HandleTypeDef* hcan) {
95      HAL_CAN_GetRxMessage(hcan, CAN_RX_FIFO0, &RxHeader, RxData);
96      cntt++;
97      while (cntt - 100 > 0) {
98          HAL_GPIO_ToggleIn(GPIOA, GPIO_PIN_0);
99          CNTT = 0;
100     }
101
102     /* DATA RECEIVER FROM USB CAN */
103     if (RxHeader.DLC == 8) {
104         RxData1 = (RxData[0] << 8) | RxData[1];
105         RxData2 = (RxData[2] << 8) | RxData[3];
106         RxData3 = (RxData[4] << 8) | RxData[5];
107         RxData4 = (RxData[6] << 8) | RxData[7];
108
109         /* TRANSFER TO SPEED */
110         /* V1 RGB */
111         V1 = RxData1;
112         V2 = RxData2;
113         V3 = RxData3;
114         V4 = RxData4;
115
116         flag = 1;
117     }
118
119 }
120
121 /* USER CODE END PFP */
122 /* Infinite loop */
123 /* USER CODE BEGIN WHILE */
124 while (1)
125
126 /* Infinite loop */
127 /* USER CODE END WHILE */
128 /* Infinite loop */
129 /* USER CODE BEGIN END WHILE */
130 /* Infinite loop */
131
132 /* Static Stack Analyzer @ Cyclomatic Complexity
F446RET6_CAN_FreeRTOS_RX.eif - F446RET6_CAN_FreeRTOS_RX/Debug - Feb 20, 2024, 11:05:56 AM
Memory Regions Memory Details
```

Activities org.eclipse.core.internal.jobs.Worker

File Edit Source Refactor Navigate Search Project Run Window Help myst

STM32\_WS\_V5 - F446RET6\_CAN\_FreeRTOS\_RX/Core/Src/main.c - STM32CubeIDE

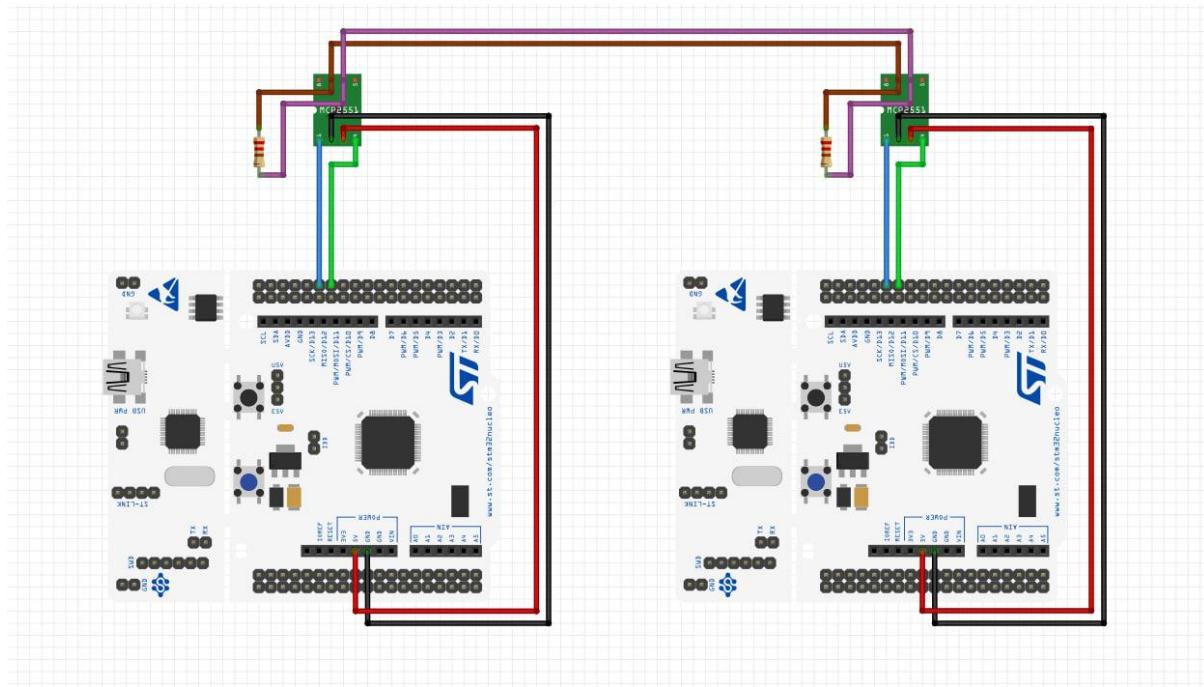
Project Explorer F446RET6\_CAN\_FreeRTOS\_RX.loc main.c

```
163  /* USER CODE END SysInit */
164
165  /* INITIALIZATION OF ALL configured peripherals */
166  MX_GPIO_Init();
167  MX_CAN1_Init();
168  MX_TIM3_Init();
169  MX_TIM4_Init();
170
171  /* CAN */
172  HAL_CAN_Start(&hcan);
173
174  /* STRUCTURE TRANSMITTER DATA */
175  HAL_CAN_ActivateNotification(&hcan, CAN_IT_RX_FIFO0_MSG_PENDING);
176  TxHeader.IDC = CAN_ID_STD;
177  TxHeader.IDE = CAN_ID_STD;
178  TxHeader.RTR = CAN_RTR_DATA;
179  TxHeader.StdId = 0x445; //0b110010100001
180
181  /* TIMER RGB */
182
183  HAL_TIM_Base_Start_IT(&htim4);
184
185  HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_1);
186  HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_2);
187  HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_3);
188
189
190
191  /* MAP TX DATA FROM 8BIT TO 16BIT (RGB) */
192
193  V1_out = 65535;
194  V2_out = 65535;
195  V3_out = 65535;
196  V4_out = 65535;
197
198
199
200  /* USER CODE END 2 */
201
202
203  /* Infinite loop */
204  /* USER CODE BEGIN WHILE */
205  while (1)
206
207 /* Static Stack Analyzer @ Cyclomatic Complexity
F446RET6_CAN_FreeRTOS_RX.eif - F446RET6_CAN_FreeRTOS_RX/Debug - Feb 20, 2024, 11:05:56 AM
Memory Regions Memory Details
```

## 11. Code can.c in (Rx)

### III. Connection and Diagram

The connection between F446 Nucleo TX and F446 Nucleo RX is shows below.



- Here the Tx and Rx from the Transceivers are connected to PA12 and PA11 of the Respective controllers
- CANH and CANL are connected to each other
- Also there is **120 ohms** Resistance at each node. This is very important, or else you will not get the data.