Hello all,
i have 32F072BDISCOVERY with STM32F072RBT6.

Soldet external 8mhz crystal,

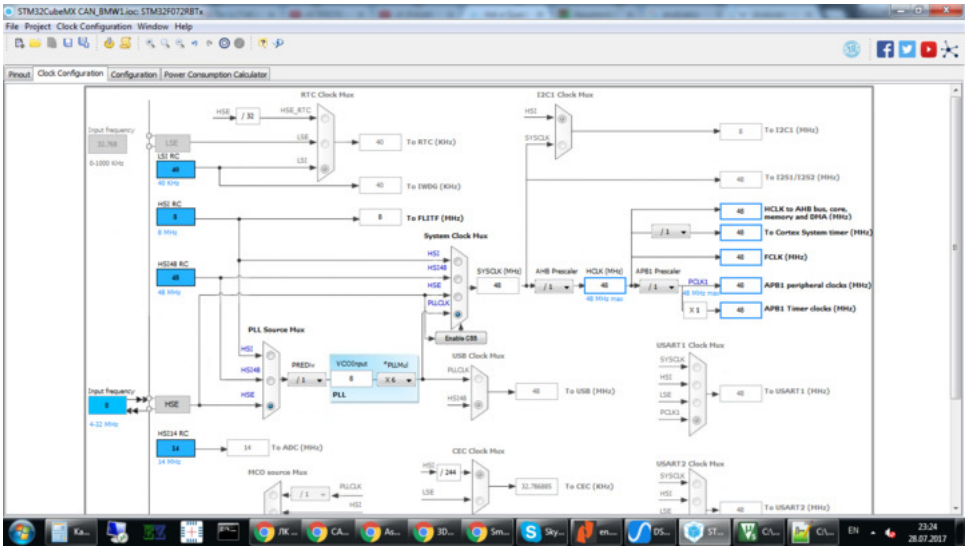
Remove SB17, SB18, SB1
X3 (8mhz crystal), R36, R37, C22, C23 soldered

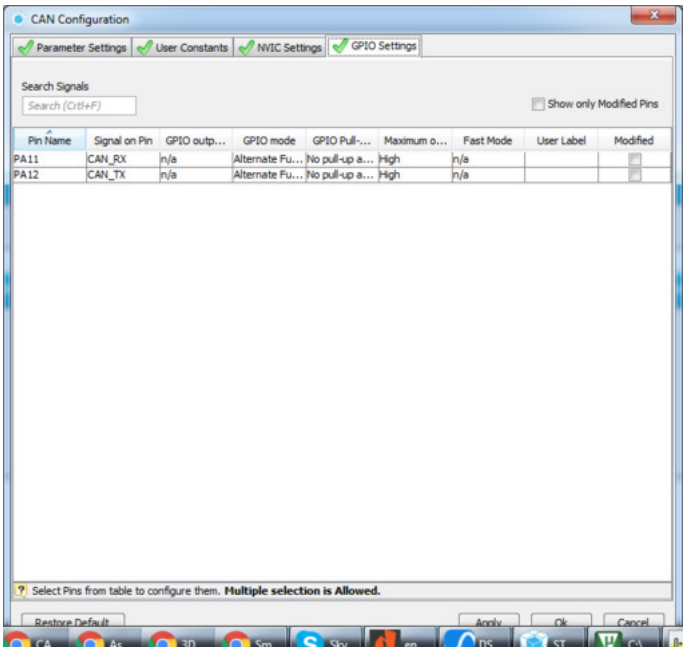
The problem is that can bus not working, i trying to check it by Analyser and have connected mp 2551, no signal.....

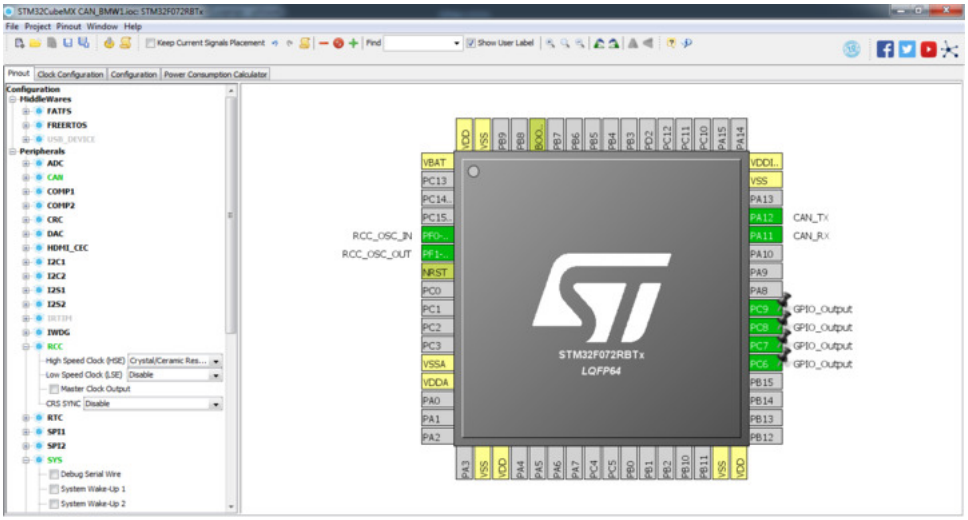
Configured all by cubemx with default pins 11 and 12pa.

HAL

Before solded crystal and after, the same problem, no signal.







Here is my configs and code.

Main

```
/**
  ******************************************************************************
  * File Name          : main.c
  * Description        : Main program body
  ******************************************************************************
  ** This notice applies to any and all portions of this file
  * that are not between comment pairs USER CODE BEGIN and
  * USER CODE END. Other portions of this file, whether
  * inserted by the user or by software development tools
  * are owned by their respective copyright owners.
  *
  * COPYRIGHT(c) 2017 STMicroelectronics
  *
  * Redistribution and use in source and binary forms, with or without modification,
  * are permitted provided that the following conditions are met:
  *   1. Redistributions of source code must retain the above copyright notice,
  *      this list of conditions and the following disclaimer.
  *   2. Redistributions in binary form must reproduce the above copyright notice,
  *      this list of conditions and the following disclaimer in the documentation
  *      and/or other materials provided with the distribution.
  *   3. Neither the name of STMicroelectronics nor the names of its contributors
  *      may be used to endorse or promote products derived from this software
  *      without specific prior written permission.
  *
  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
  * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
  * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
  * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
  * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
  * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
  * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
  * CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
  * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
  * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
  *
  ******************************************************************************
  */
/* Includes ------------------------------------------------------------------*/
#include "main.h"
#include "stm32f0xx_hal.h"

/* USER CODE BEGIN Includes */

/* USER CODE END Includes */

/* Private variables ---------------------------------------------------------*/
CAN_HandleTypeDef hcan;

/* USER CODE BEGIN PV */
/* Private variables ---------------------------------------------------------*/
static CanTxMsgTypeDef        canTxMessage;
static CanRxMsgTypeDef        canRxMessage;
/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_CAN_Init(void);

/* USER CODE BEGIN PFP */
/* Private function prototypes -----------------------------------------------*/
void CAN_Transmit(void);

/* USER CODE END PFP */

/* USER CODE BEGIN 0 */
uint16_t CAN_ID;
uint8_t CAN_DLC;
uint8_t DATA[8];

void CAN_Transmit(void)
{
    //????????? ????????? ??? ????????;
```

```
      hcan.pTxMsg->StdId = 0x222;
      hcan.pTxMsg->DLC = 8;
      hcan.pTxMsg->Data[0] = 1;
      hcan.pTxMsg->Data[1] = 2;
      hcan.pTxMsg->Data[2] = 3;
      hcan.pTxMsg->Data[3] = 4;
      hcan.pTxMsg->Data[4] = 5;
      hcan.pTxMsg->Data[5] = 6;
      hcan.pTxMsg->Data[6] = 7;
      hcan.pTxMsg->Data[7] = 8;
      HAL_CAN_Transmit(&hcan, 10);
}


void HAL_CAN_RxCpltCallback(CAN_HandleTypeDef* CanHandle)
{
      HAL_GPIO_TogglePin(GPIOC, GPIO_PIN_6);
      CAN_ID = CanHandle->pRxMsg->StdId;
      CAN_DLC = CanHandle->pRxMsg->DLC;
      DATA[0] = CanHandle->pRxMsg->Data[0];
      DATA[1] = CanHandle->pRxMsg->Data[1];
      DATA[2] = CanHandle->pRxMsg->Data[2];
      DATA[3] = CanHandle->pRxMsg->Data[3];
      DATA[4] = CanHandle->pRxMsg->Data[4];
      DATA[5] = CanHandle->pRxMsg->Data[5];
      DATA[6] = CanHandle->pRxMsg->Data[6];
      DATA[7] = CanHandle->pRxMsg->Data[7];
  HAL_CAN_Receive_IT(&hcan, CAN_FIFO0);
}


/* USER CODE END 0 */

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_CAN_Init();

  /* USER CODE BEGIN 2 */

  hcan.pTxMsg = &canTxMessage;
  hcan.pRxMsg = &canRxMessage;

     CAN_FilterConfTypeDef canFilterConfig;
  canFilterConfig.FilterNumber = 0;
  canFilterConfig.FilterMode = CAN_FILTERMODE_IDLIST;
  canFilterConfig.FilterScale = CAN_FILTERSCALE_16BIT;
  canFilterConfig.FilterIdHigh = 0x53A<<5; //????? 1
  canFilterConfig.FilterIdLow = 0x54C<<5;     //????? 2
  canFilterConfig.FilterMaskIdHigh = 0x53D<<5;     //????? 3
  canFilterConfig.FilterMaskIdLow = 0x54F<<5;     //????? 4
  canFilterConfig.FilterFIFOAssignment = 0;
  canFilterConfig.FilterActivation = ENABLE;
  canFilterConfig.BankNumber = 1;
     HAL_CAN_ConfigFilter(&hcan, &canFilterConfig);

     canFilterConfig.FilterNumber = 1;
     canFilterConfig.FilterIdHigh = 0xA<<5; //????? 5
  canFilterConfig.FilterIdLow = 0x33<<5;     //????? 6
  canFilterConfig.FilterMaskIdHigh = 0xA<<5;     //????? 7
  canFilterConfig.FilterMaskIdLow = 0x33<<5;     //????? 8
     HAL_CAN_ConfigFilter(&hcan, &canFilterConfig);

     HAL_CAN_Receive_IT(&hcan, CAN_FIFO0) ;

  /* USER CODE END 2 */

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

  /* USER CODE BEGIN 3 */
          CAN_Transmit();
          HAL_Delay(200);
          HAL_GPIO_TogglePin (GPIOC, GPIO_PIN_8);

  }
  /* USER CODE END 3 */

}
```

```
/** System Clock Configuration
*/
void SystemClock_Config(void)
{

  RCC_OscInitTypeDef RCC_OscInitStruct;
  RCC_ClkInitTypeDef RCC_ClkInitStruct;

    /**Initializes the CPU, AHB and APB busses clocks
    */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
  RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  RCC_OscInitStruct.HSIState = RCC_HSI_ON;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
  RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL6;
  RCC_OscInitStruct.PLL.PREDIV = RCC_PREDIV_DIV1;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

    /**Initializes the CPU, AHB and APB busses clocks
    */
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                              |RCC_CLOCKTYPE_PCLK1;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_1) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

    /**Configure the Systick interrupt time
    */
  HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);

    /**Configure the Systick
    */
  HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);

  /* SysTick_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
}

/* CAN init function */
static void MX_CAN_Init(void)
{

  hcan.Instance = CAN;
  hcan.Init.Prescaler = 20;
  hcan.Init.Mode = CAN_MODE_NORMAL;
  hcan.Init.SJW = CAN_SJW_1TQ;
  hcan.Init.BS1 = CAN_BS1_15TQ;
  hcan.Init.BS2 = CAN_BS2_8TQ;
  hcan.Init.TTCM = DISABLE;
  hcan.Init.ABOM = DISABLE;
  hcan.Init.AWUM = DISABLE;
  hcan.Init.NART = DISABLE;
  hcan.Init.RFLM = DISABLE;
  hcan.Init.TXFP = DISABLE;
  if (HAL_CAN_Init(&hcan) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

}

/** Configure pins as
        * Analog
        * Input
        * Output
        * EVENT_OUT
        * EXTI
*/
static void MX_GPIO_Init(void)
{

  GPIO_InitTypeDef GPIO_InitStruct;

  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOF_CLK_ENABLE();
  __HAL_RCC_GPIOC_CLK_ENABLE();
  __HAL_RCC_GPIOA_CLK_ENABLE();

  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOC, GPIO_PIN_6|GPIO_PIN_7|GPIO_PIN_8|GPIO_PIN_9, GPIO_PIN_RESET);

  /*Configure GPIO pins : PC6 PC7 PC8 PC9 */
  GPIO_InitStruct.Pin = GPIO_PIN_6|GPIO_PIN_7|GPIO_PIN_8|GPIO_PIN_9;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

}

/* USER CODE BEGIN 4 */

/* USER CODE END 4 */
```

```
/**
  * @brief  This function is executed in case of error occurrence.
  * @param  None
  * @retval None
  */
void _Error_Handler(char * file, int line)
{
  /* USER CODE BEGIN Error_Handler_Debug */
  /* User can add his own implementation to report the HAL error return state */
  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

/**
  * @}
  */

/**
  * @}
*/

/************************ (C) COPYRIGHT STMicroelectronics *****END OF FILE****/
?????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????

void HAL_CAN_MspDeInit(CAN_HandleTypeDef* hcan)
{

  if(hcan->Instance==CAN)
  {
  /* USER CODE BEGIN CAN_MspDeInit 0 */

  /* USER CODE END CAN_MspDeInit 0 */
    /* Peripheral clock disable */
    __HAL_RCC_CAN1_CLK_DISABLE();

    /**CAN GPIO Configuration
    PA11     ------> CAN_RX
    PA12     ------> CAN_TX
    */
    HAL_GPIO_DeInit(GPIOA, GPIO_PIN_11|GPIO_PIN_12);

    /* CAN interrupt DeInit */
    HAL_NVIC_DisableIRQ(CEC_CAN_IRQn);
  /* USER CODE BEGIN CAN_MspDeInit 1 */

  /* USER CODE END CAN_MspDeInit 1 */
  }

}?????????????????????????
```