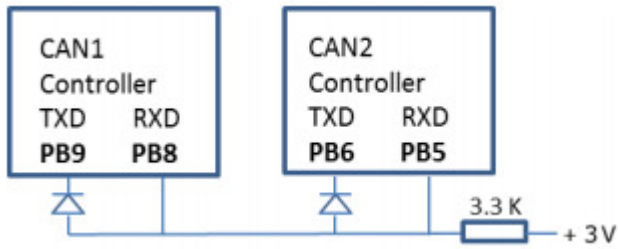


Hi.
I'm trying to make my STM32F4-Discovery talk to itself through CAN, using this configuration



and here's my code.

```

001. #include "stm32f4xx_hal.h"
002. #include "stm32f4xx_hal_can.h"
003. #include "Board_Led.h"
004.
005.
006. CAN_HandleTypeDef CAN1_Handle, CAN2_Handle;
007. CAN_FilterConfTypeDef CAN2_Filter;
008. CanRxMsgTypeDef CAN1RX, CAN2RX;
009. CanTxMsgTypeDef CAN1TX, CAN2TX;
010. GPIO_InitTypeDef GPIO_Init;
011.
012. static void SystemClock_Config(void);
013. void CAN1_Config(void);
014. void CAN2_Config(void);
015. void CAN_SendMsg(void);
016. void Button_Config(void);
017. int main (void)
018. {
019.     HAL_Init();
020.     SystemClock_Config();
021.     LED_Initialize();
022.     LED_On(0);
023.     Button_Config();
024.     CAN1_Config();
025.     CAN2_Config();
026.     while (1);
027. }
028.
029. void CAN1_Config(void)
030. {
031.     //GPIO Config PD0 = CAN1 RX, PD1 = CAN1 TX
032.     __GPIOC_CLK_ENABLE();
033.     GPIO_Init.Mode =GPIO_MODE_AF_PP;
034.     GPIO_Init.Alternate= GPIO_AF9_CAN1;
035.     GPIO_Init.Pin = GPIO_PIN_0|GPIO_PIN_1;
036.     GPIO_Init.Pull = GPIO_PULLUP;
037.     GPIO_Init.Speed = GPIO_SPEED_FAST;
038.     HAL_GPIO_Init(GPIOC,&GPIO_Init);
039.
040.     __CAN1_CLK_ENABLE();
041.
042.     CAN1_Handle.Instance= CAN1;
043.     CAN1_Handle.pRxMsg = &CAN1RX;
044.     CAN1_Handle.pTxMsg = &CAN1TX;
045.     CAN1_Handle.Init.ABOM = DISABLE;
046.     CAN1_Handle.Init.AWUM = DISABLE;
047.     CAN1_Handle.Init.BS1 = CAN_BS1_9TQ;
048.     CAN1_Handle.Init.BS2 = CAN_BS2_8TQ;
049.     CAN1_Handle.Init.Mode = CAN_MODE_NORMAL;
050.     CAN1_Handle.Init.NART = DISABLE;
051.     CAN1_Handle.Init.Prescaler = 16;
052.     CAN1_Handle.Init.RFLM = DISABLE;
053.     CAN1_Handle.Init.TTCM = DISABLE;
054.     CAN1_Handle.Init.SJW = CAN_SJW_4TQ;
055.     CAN1_Handle.Init.TYED = DISABLE;

```

```
054.     CAN1_Handle.Init.ABOM = DISABLE;
055.     if(HAL_CAN_Init(&CAN1_Handle)== HAL_OK)
056.     {
057.         printf("CAN1 Initialized.\n");
058.     }
059.
060.     HAL_NVIC_EnableIRQ(CAN1_TX_IRQn);
061. }
062.
063. void CAN2_Config(void)
064. {
065.     //GPIO PB12 = CAN2RX, PB13 = CAN2TX
066.     __GPIOB_CLK_ENABLE();
067.     GPIO_Init.Mode =GPIO_MODE_AF_PP;
068.     GPIO_Init.Alternate= GPIO_AF9_CAN2;
069.     GPIO_Init.Pin = GPIO_PIN_12|GPIO_PIN_13;
070.     GPIO_Init.Pull = GPIO_PULLUP;
071.     GPIO_Init.Speed = GPIO_SPEED_FAST;
072.     HAL_GPIO_Init(GPIOB,&GPIO_Init);
073.
074.
075.     __CAN2_CLK_ENABLE();
076.     CAN2_Handle.Instance= CAN2;
077.     CAN2_Handle.pRxMsg = &CAN2RX;
078.     CAN2_Handle.pTxMsg = &CAN2TX;
079.     CAN2_Handle.Init.ABOM = DISABLE;
080.     CAN2_Handle.Init.AWUM = DISABLE;
081.     CAN2_Handle.Init.BS1 = CAN_BS1_9TQ;
082.     CAN2_Handle.Init.BS2 = CAN_BS2_8TQ;
083.     CAN2_Handle.Init.Mode = CAN_MODE_NORMAL;
084.     CAN2_Handle.Init.NART = DISABLE;
085.     CAN2_Handle.Init.Prescaler = 16;
086.     CAN2_Handle.Init.RFLM = DISABLE;
087.     CAN2_Handle.Init.TTCM = DISABLE;
088.     CAN2_Handle.Init.SJW = CAN_SJW_4TQ;
089.     CAN2_Handle.Init.TXFP = DISABLE;
090.     if(HAL_CAN_Init(&CAN2_Handle)== HAL_OK)
091.     {
092.         printf("CAN2 Initialized.\n");
093.     }
094.     CAN2_Filter.FilterNumber = 14;
095.     CAN2_Filter.FilterActivation = ENABLE;
096.     CAN2_Filter.BankNumber = 14;
097.     CAN2_Filter.FilterFIFOAssignment = CAN_FilterFIFO0;
098.     CAN2_Filter.FilterIdHigh =0x0;
099.     CAN2_Filter.FilterIdLow = 0x0;
100.     CAN2_Filter.FilterMaskIdHigh =0x0;
101.     CAN2_Filter.FilterMaskIdLow =0x0;
102.     CAN2_Filter.FilterMode = CAN_FILTERMODE_IDMASK;
103.     CAN2_Filter.FilterScale = CAN_FILTERSCALE_32BIT;
104.     if(HAL_CAN_ConfigFilter(&CAN2_Handle,&CAN2_Filter)== HAL_OK)
105.     {
106.         printf("Filter configd.\n");
107.     }
108.
109.
110.     HAL_NVIC_SetPriority(CAN2_RX0_IRQn,0,0);
111.     HAL_NVIC_EnableIRQ(CAN2_RX0_IRQn);
112.     HAL_CAN_Receive_IT(&CAN2_Handle,CAN_FIFO0);
113. }
114.
115.
116.
117. void Button_Config(void)
118. {
119.     __GPIOA_CLK_ENABLE();
120.     GPIO_Init.Mode = GPIO_MODE_IT_RISING;
121.     GPIO_Init.Pull = GPIO_PULLDOWN;
```

```
122.     GPIO_Init.Speed = GPIO_SPEED_FAST;
123.     GPIO_Init.Pin = GPIO_PIN_0;
124.     HAL_GPIO_Init(GPIOA,&GPIO_Init);
125.     HAL_NVIC_EnableIRQ(EXTI0_IRQn);
126. }
127.
128. void EXTI0_IRQHandler(void)
129. {
130.     HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);
131. }
132. void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
133. {
134.     CAN_SendMsg();
135. }
136.
137. void CAN_SendMsg(void)
138. {
139.     CAN1TX.IDE = CAN_ID_STD;
140.     CAN1TX.StdId = 10;
141.     CAN1TX.ExtId = 0x01;
142.     CAN1TX.RTR = CAN_RTR_DATA;
143.     CAN1TX.DLC = 1;
144.     CAN1TX.Data[0]=3;
145.     if(HAL_CAN_Transmit_IT(&CAN1_Handle)== HAL_OK)
146.     {
147.         printf("Msg sent.\n");
148.     }
149. }
150.
151.
152.
153. void CAN2_RX0_IRQHandler(void)
154. {
155.     HAL_CAN_IRQHandler(&CAN2_Handle);
156. }
157. void HAL_CAN_RxCpltCallback(CAN_HandleTypeDef *hcan)
158. {
159.     printf("rceived %d",CAN2RX.Data[0]);
160. }
```

The first time I push the button, the program indicates that the message is successfully sent. CAN2 receives nothing however.

HSE_Value is set to 8 MHz.

Any insights would be appreciated.

Thanks in advance.