Problem in block CAN_Send and CAN_Receive from STM32-MAT/TARGET 4.4.1

This document explains, in a step by step problem in block send and receives CAN and proposes a solution for this problem

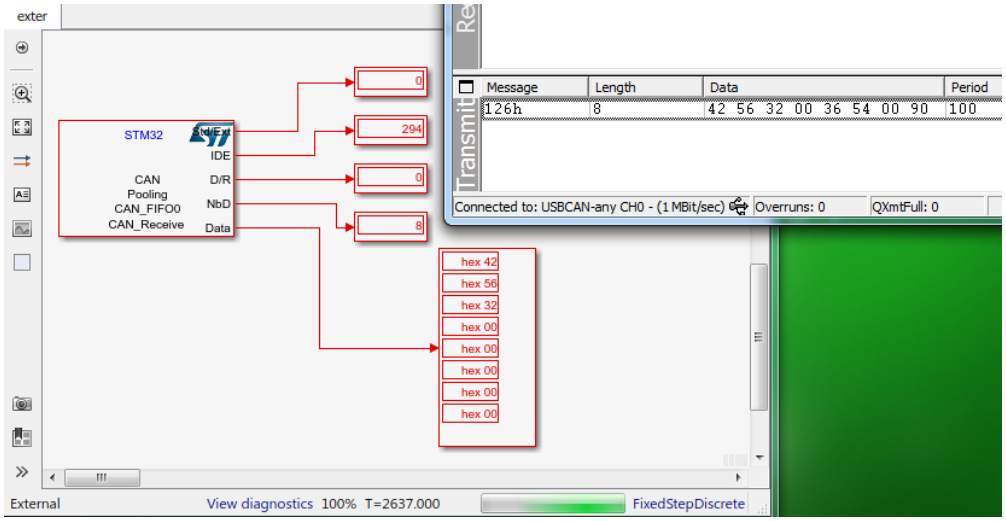
By:

Abdelghafour SID, Electrical Engineering and Master in Electronics
Courriel: sidabdelghafour1@gmail.com
Cel. : (+213) 663407809

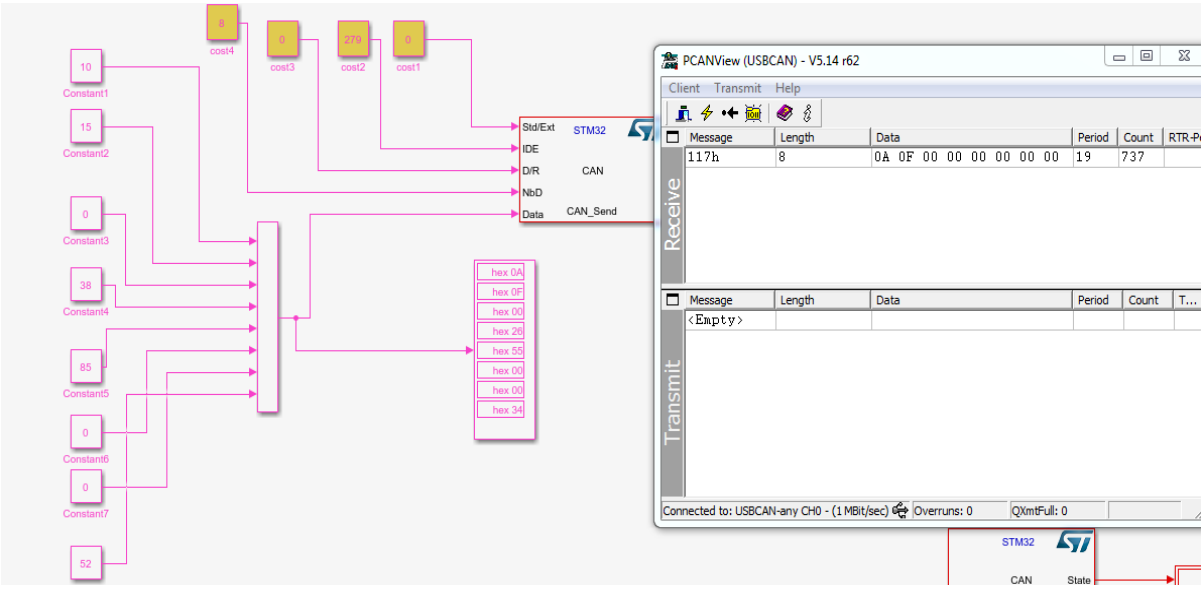
When we send or receive a frame CAN contain a zero the block Delete the frame from the first zero

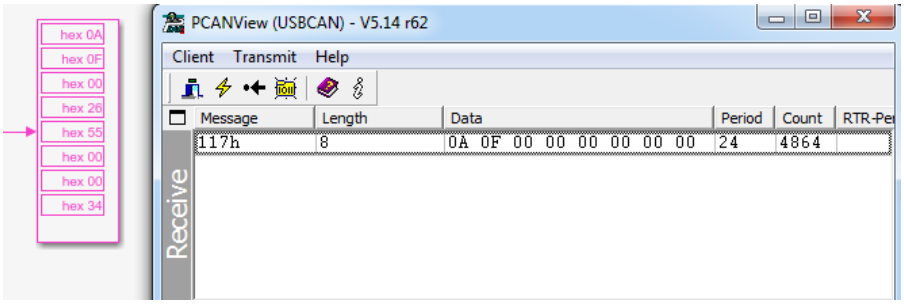
Demonstration

CAN Receive

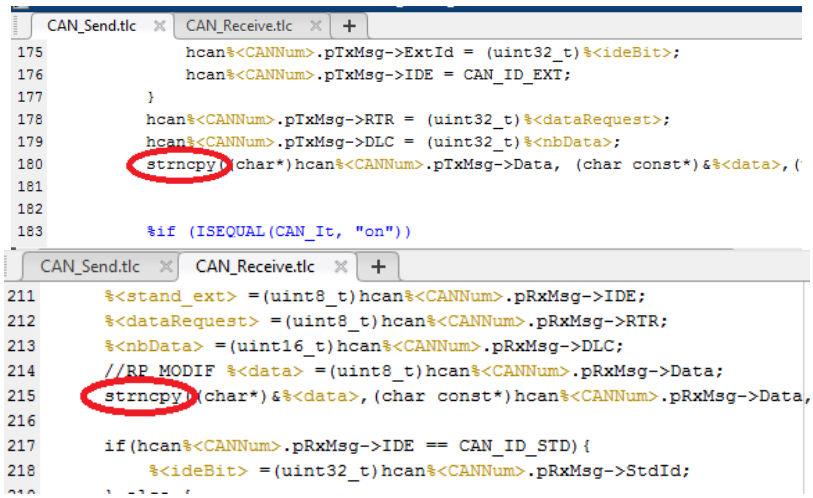


CAN Send





Source of block



The problem came from using the function 'strncpy' this function treat the frame like string chain when it found a zero it the end of the string chain

To solve this problem we should use another function to to copy the string chain, this my function:

In the code generated by STM32CubMX we change all function 'strncpy' by 'mystrncpy'

```
/* S-Function Block: <Root>/CAN_Send2 */

/* Build frame to send */
if (test_send_receive_P.cost1_Value == CAN_ID_STD) {
  hcan.pTxMsg->StdId = (uint32_t)test_send_receive_P.cost2_Value;
  hcan.pTxMsg->ExtId = (uint32_t)0;
  hcan.pTxMsg->IDE = CAN_ID_STD;
} else {
  hcan.pTxMsg->StdId = (uint32_t)0;
  hcan.pTxMsg->ExtId = (uint32_t)test_send_receive_P.cost2_Value;
  hcan.pTxMsg->IDE = CAN_ID_EXT;
}

hcan.pTxMsg->RTR = (uint32_t)test_send_receive_P.cost3_Value;
hcan.pTxMsg->DLC = (uint32_t)test_send_receive_P.cost4_Value;
strncpy((char*)hcan.pTxMsg->Data, (char const*)&test_send_receive_B.Constant1,
        (uint32_t)test_send_receive_P.cost4_Value);

/* Send pooling mode frame */
HAL_CAN_Transmit(&hcan, 10);
```

mystrncpy

```
void mystrncpy(char * reciver , char const* source,int len)
{
  for(int i =0;i<len;i++)
   *(reciver +i) =*(source+i);
}
```

```
void mystrncpy(char* A, char const* B,int len)
{
      for(int i =0;i<8;i++)
      A[i] =B[i];
}
```

Changing

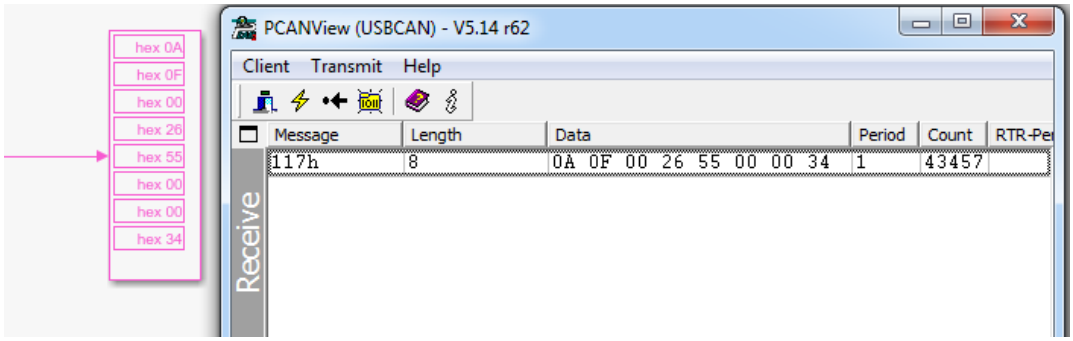
```
/* S-Function Block: <Root>/CAN_Send2 */

/* Build frame to send */
if (test_send_receive_P.cost1_Value == CAN_ID_STD) {
  hcan.pTxMsg->StdId = (uint32_t)test_send_receive_P.cost2_Value;
  hcan.pTxMsg->ExtId = (uint32_t)0;
  hcan.pTxMsg->IDE = CAN_ID_STD;
} else {
  hcan.pTxMsg->StdId = (uint32_t)0;
  hcan.pTxMsg->ExtId = (uint32_t)test_send_receive_P.cost2_Value;
  hcan.pTxMsg->IDE = CAN_ID_EXT;
}

hcan.pTxMsg->RTR = (uint32_t)test_send_receive_P.cost3_Value;
hcan.pTxMsg->DLC = (uint32_t)test_send_receive_P.cost4_Value;
mystrncpy((char*)hcan.pTxMsg->Data, (char const*)&test_send_receive_B.Constant1,
        (uint32_t)test_send_receive_P.cost4_Value);

/* Send pooling mode frame */
HAL_CAN_Transmit(&hcan, 10);
```

Result


hex 0A
hex 0F
hex 00
hex 26
hex 55
hex 00
hex 00
hex 34
PCANView (USBCAN) - V5.14 r62
Client
Transmit
Help
Message
Length
Data
Period
Count
RTR-Per
117h
8
0A 0F 00 26 55 00 00 34
1
43457
Receive