Dual CAN application

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	The code is a demonstration of how to use dual CAN
BSP Version	NUC230_240 Series BSP CMSIS v3.01.001
Hardware	NuEdu-EVB-NUC240 v2.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 Function Description

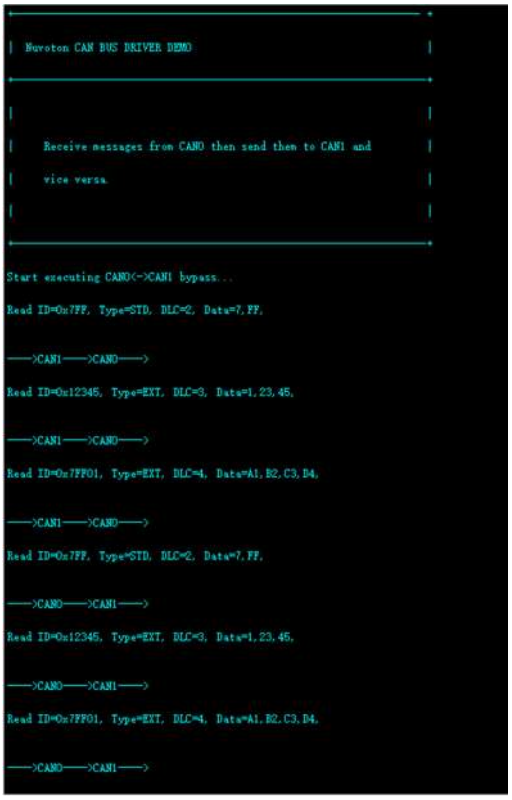
1.1 Introduction

The peripherals of Nuvoton's CAN are directly designed by Bosch, so there is no problem with CAN compatibility. Before using it, everyone needs to have a basic idea. It is not like RS485, one byte per byte is slowly transmitted. Instead, the complete packet data is stored in the register and then moved to the internal cache. The data is automatically transmitted or received through the hardware. Avoid interrupting the modification of the packet data. In the main, the program must disable the CAN interrupt when reading and writing the packet data.

1.2 Principle

The CAN example is very simple. It is a bridge that forwards the data received by CAN0 through CAN1, or forwards the data received by CAN1 through CAN0. In the application of the car, it can be used in the aftermarket, the host transmits the data to CAN0, through the NUC230, the commands of some hosts are modified, and then transmitted to each controller through CAN1. At the same time, each controller will periodically return the data back to CAN1, and then transmit the same data to the host through CAN0.

1.3 Demo Result



In this example, the test result is output with NUC240 UART0 (TX is PB1), and the serial port baud rate is 115200.

2 Code Description

CAN0 receives the data and forwards the data through CAN1

```
if (CountsInFifo(&g_srrMsg0))
{
   /* There are some messages in CAN0 FIFO,send them to CAN1 */
   GetFromFifo(&g_srrMsg0, &rrMsg);
   CAN_ShowMsg(&rrMsg);
   /* You can add code here to process CAN messages before send it out */
   printf("---->CAN0---->CAN1---->\n");
   SendMsgObj(tCAN1, TX_MSG_OBJ_ID, &rrMsg);
}
```

CAN1 receives the data and forwards the data through CAN0

```
if (CountsInFifo(&g_srrMsg1))
{
    /* There are some messages in CAN1 FIFO,send them to CAN0 */
    GetFromFifo(&g_srrMsg1, &rrMsg);
    CAN_ShowMsg(&rrMsg);
    /* You can add code here to process CAN messages before send it out */
    printf("---->CAN1---->CAN0---->\n");
    SendMsgObj(tCAN0, TX_MSG_OBJ_ID, &rrMsg);
}
```

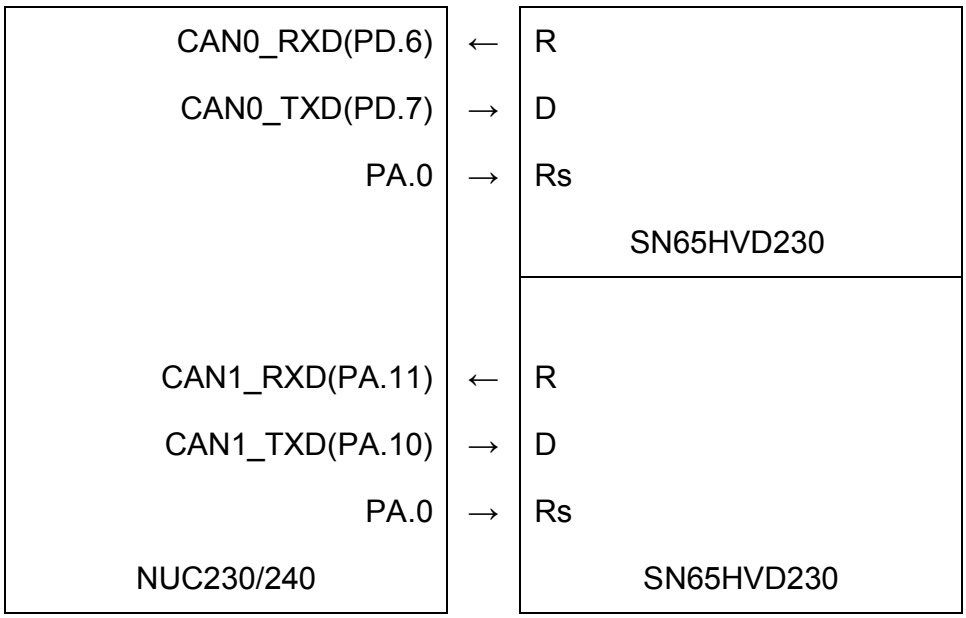
3 Software and Hardware Environment

Software Environment

- BSP version
  - NUC230_240 Series BSP CMSIS v3.01.001
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuEdu-EVB-NUC240
  - SN65HVD230(CAN Transceiver)
- Diagram



4 Directory Information

- EC_NUC230_Dual_CAN_Bypass_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code

5 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click NUC230_Dual_CAN_Bypass.uvproj.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Oct. 25, 2019	1.00	1. Initially issued.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.