RESEARCH ARTICLE

International Journal of Computer Communication and Informatics

DOI: 10.34256/ijcci2222

Implementation of Driver Software of Trailer Module Chip

C. Sundar Ganesh [a], T. Vinoth Kumar [b*], B. Gopinath [c], C. Madhan Kumar [b], B. Suresh [d], M. Vijayaragavan [e]

[a] Department of Electrical and Electronics Engineering, Karpagam College of Engineering, Coimbatore, India.

[b] Department of Electrical and Electronics Engineering, RVS College of Engineering and Technology Coimbatore, India

[c] Department of Electrical and Electronics Engineering, Christ the King Engineering College Coimbatore, India

[d] Department of Electrical and Electronics Engineering, Angel College of Engineering and Technology, Tirupur, India

[e] Department of Electrical and Electronics Engineering, Mailam Engineering College Mailam, India

* Corresponding Author: tvinoth15f@gmail.com

Received: 17-08-2022, Revised: 23-11-2022, Accepted: 16-12-2022, Published: 30-12-2022

Abstract: The aim of the project is to develop a driver software for UJA1076A SBC in embedded C using IAR Embedded Workbench and integrate the driver software with application software of Trailer module. Currently MC33903 system basis chip from Freescale is used in Trailer Module. As an initiative to reduce the material cost for the Trailer module product, a lower price SBC NXP UJA1076A has been used. Also due to the fact that the newly proposed SBC has less number of operating modes and registers to configure, it helps in making the driver software much more simpler, thus reducing the risk of hidden issues in the otherwise complex design and code of the current SBC driver software.

Keywords: System Basis Chip (SBC), Electronic Control Unit (ECU), TRailer Module(TRM), Controller Area Network (CAN), Serial Peripheral Interface (SPI).

1. Introduction

In automotive electronics, ECU controls one or more of the electrical system or subsystems in a vehicle. Currently, a commercial vehicle can have up to 40 ECUs, and a car up to 100.one of the ECU is TRailer Module (TRM) that contains System Basis Chip (SBC) [1]. The trailer module controls and monitors the current circuits of the trailer. It also separates the current circuits of the trailer from the vehicle network of the towing vehicle [2]. A SBC is an integrated circuit that includes various functions of automotive ECU on a single die. SBCs integrate an energy management module to supply the system Microcontroller [3]. Low-power modes for transceivers to link with in-vehicle networks and transceiver interfaces for battery conservation.

2. Comparison Between MC33903 and UJA1076A

Currently MC33903 system basis chip from free scale is used in Trailer Module. MC33903 has more number of operating modes and registers to configure [4]. Hence the driver software design and code became complex and it lead to the increase of hidden design risk and code issues. The cost of the SBC is high. To overcome the above disadvantages, the SBC which is used in Trailer module is changed from MC33903 to NXP UJA1076A but microcontroller remains same. This new SBC's cost is lesser than MC33903 which is also an added advantage. Hence the software driver that handles MC33903 chip also has to be changed to a software driver that can control the UJA1076A [5-7]. The difference between MC33903 and UJA1076A is shown in TABLE 1

Table 1 Difference Between MC33903 and UJA1076A

Parameters	**MC33903**	**UJA1076A**
Number of Operating modes	9	4
Number of Registers	20	4
Hidden design issue	High	Low
Code complexity	High	Low
Cost	High	Low
Maximum baud rate for CAN	1.0Mb/s	5Mb/s
Sleep current consumption	100 uA or few mA	87uA
Maximum DC capability	150mA	250mA

3. Hardware Architecture of UJA1076A

The UJA1076A SBC enables the use of a high-speed CAN as the primary network interface to support the networking applications used to regulate power and sensor peripherals [8]. In a single, specialised chip, it combines the capabilities of a high-speed CAN transceiver, two voltage regulators, and a watchdog. It manages the ECU's power-up and power-down capabilities and makes sure the system is highly reliable. The following integrated devices are found in the core SBC: a high-speed CAN transceiver, a 250 mA voltage regulator for powering a microcontroller, Serial Peripheral Interface (SPI) full duplex, 2 local wake-up input ports, another voltage regulator for powering the on-board CAN transceiver, and a limp-home output connection are all included. The core SBC offers an intelligent combination of system-specific functions, such as advanced low-power concept, Safe and controlled system start-up behavior, and detailed status reporting on system and sub-system levels, in addition to the benefits of combining these common ECU functions in a single package [9-11]. Fig.1. shows the block diagram of UJA1076A

4. Modules In UJA1076A

A System Controller

The system controller oversees SBC internal operations and register configuration. The microcontroller is given comprehensive device status information [12]. A state machine serves as the system controller. The SBC operational modes and how mode changes are initiated are illustrated in Fig.2.

B. Operating Modes

- *Off Mode*

Voltage regulators are disabled and the bus systems are in a high-resistance state while the system is in Off mode..

- *Standby Mode*

V1 is turned on and the CAN transceiver will be in low-power mode. The watchdog may be operating in Off mode or Timeout mode.

- *Normal Mode*

V1 is operated and the CAN physical layer will be enabled or in a low-power state with bus wake-up detection active.

- *Sleep Mode*

V1 and V2 are off in sleep mode, and the CAN transceiver will be turned off with bus wake-up detection turned on. The reset pin is LOW and the watchdog is not running.

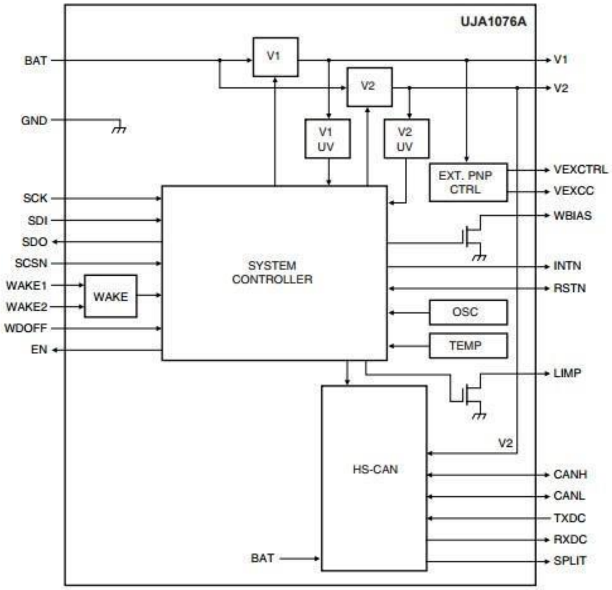

Figure.1 Block Diagram of UJA1076A

- *Overtemp Mode*

When the chip temperature reaches the over-temperature protection ativation threshold, Tth(act)otp, the SBC will enter this mode.

C. Serial Peripheral Interface

The communication channel with the microcontroller is provided via the Serial Peripheral Interface which enables multi-slave operations. Since the SPI is set up for full duplex data transfer, when fresh control data is shifted in, status information is returned [13]. On the falling clock edge, bit sampling is carried out, and on the rising clock edge, data shifts. Four interface signals are utilised by the SPI for synchronisation and data transfer:

D. Watchdog Timer

The watch dog modes has operated in three modes: Window, Timeout and Off.

Window Mode

In Window mode, the watchdog keeps running continually. In watchdog Window mode, an SBC reset is produced by a watchdog trigger event inside a closed watchdog window [3]. The timer resets right away if the watchdog is triggered while the watchdog window is still open.

Timeout Mode

In Timeout mode, the watchdog keeps running continually. A watchdog trigger allows it to be reset at any time.

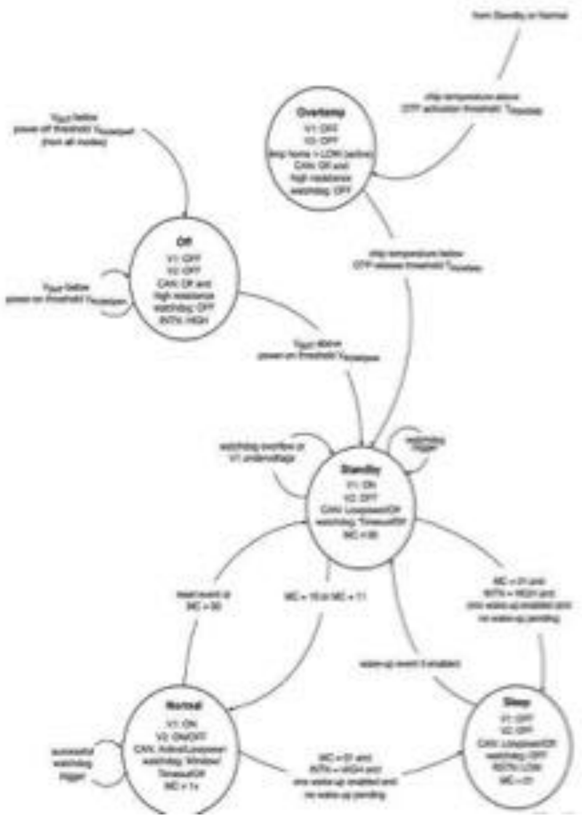

Figure. 2 State diagram of Operating Modes

Off Mode

In this mode, watchdog is disabled. When the SBC is in Off, Over temp, or Sleep modes, the watchdog is in Off mode.

E. Voltage Regulator V1

The voltage regulator V1 is used to power the microcontroller, as well as extra transceivers and its peripherals. Up to 250 mA at 3.3 V or 5 V can be delivered by V1, which is powered by pin BAT.

F. Voltage Regulator V2

Voltage regulator V2, which supplies a 5 V supply, is set aside for the high-speed CAN transceiver [14]. The MC bits in the Mode Control register allow for the activation and deactivation of V2.

G. CAN transceiver

An interface between a Controller Area Network (CAN) protocol controller and the actual two-wire CAN bus is provided by the TJA1042 high-speed CAN transceiver [4]. It operates in two different ways: Active mode and Low power/off mode. The transceiver send and receive data using the CANH and CANL pins during CAN Active mode. Digital data is output on pin RXDC by the differential receiver, which transforms the analog data on the bus lines. The transmitter transforms digital data input on pin TXDC from a CAN controller into signals appropriate for transmission over bus lines. In Off mode, the CAN transceiver is totally turned off to reduce current consumption [12, 15].

H. Limp Output

In the case of an ECU failure, the LIMP pin can be utilized to activate the so-called "limp home" hardware.

The limp home warning control bit (LHWC) will be set after a reset. After each reset event, the programme needs to delete LHWC to make sure the LIMP output isn't active during regular operation.

I. Registers for configuration

UJA1076A has four registers to configure the components with various modes [14]

Watchdog and status register

Mode control register

Interrupt control register

Interrupt status register

5. Proposed Algorithm for Driver Software Implementation

The algorithm and flowchart of driver software has been explained for different operating modes of SBC. Figures 3, 4,5 and 6 shows for the flowchart for different modes.

A. Init_On_Reset Algorithm

Step 1: Clear TX and RX Buffer

Step 2: Set SBC_state=SBC_init
Step 3: Jump to Init_Config

Init_ Config (Register Settings for Normal Mode)

Step 1: Set Watch Dog Period and Mode in Wd_And_Status Register
Step 2: Set Normal Mode in Mode_Control Register
Step 3: Enable Can and Under_Vltg Warnings in Int_Control Register
Step 4: Set Required Interrupt Warnings in Int_Status Register

B. Power Down Algorithm

Step 1: Set SBC_state=SBC_standby
Step 2: **if** Supply Voltage is under voltage **then**
/*Disable the CAN and Wake up detection
Enable Standby Mode*/
else
/* Enable CAN Wake Up Detection and Standby Mode*/

The flow of power down mode is shown in Fig.3

C. Wakeup Algorithm

Step 1: Clear TX and RX Buffer

Step 2: Set SBC_state=SBC_init

Step 3: Jump to Wakeup_Config

Wakeup_ Config (Register Settings for Normal Mode)

Step 1: Set Watch Dog Period and Mode in

Wd_And_Status Register

Step 2: Set Normal Mode in Mode_Control Register

Step 3: Enable Can and Under_Vltg Warnings in Int_Control Register

Step 4: Set Required Interrupt Warnings in Int_Status Register

The flow of Wake up mode is shown in Fig.4

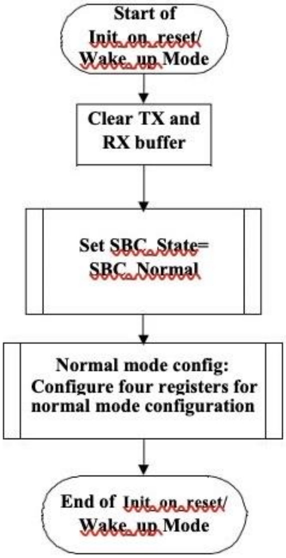

Figure. 4 Flow chart for Wakeup Mode.

D.SBC_Run Algorithm

STEP 1: Check for SBC_State

STEP 2: if SBC_state==SBC_Normal then

STEP 3: Refresh the Watchdog status

STEP 4: if CAN_Disable && Standby == False **then** /* Enable CAN RX TX */

else

/* Read CAN RX_TX Reg */

STEP 5: **if** SBC not in Normal && Standby == False **then** /*Enable Normal Mode*/

else

/* Read MCR Register */

STEP 6: Check for LimpHomeMode

E. SBC Call_Back Algorithm

STEP 1: Check TX_buff if any more data still to be transferred

STEP 2: Check if SBC_state==SBC_init **then** /* Move the SBC state to normal */

STEP 3: Check if SBC_state ==SBC_standby **then /*** SBC _standby_done=true*/

STEP 4: Check if SBC_state=SBC_wakeup /*move SBC to normal mode*/

STEP 5: Check if SBC_state=SBC_normal **if** CAN_RX_TX _read **then**

check STBCC_mask==CAN_disable **if** MCR_read **then**

check normal_mode_mask! =normal mode

STEP 6: Check for Limp Home mode

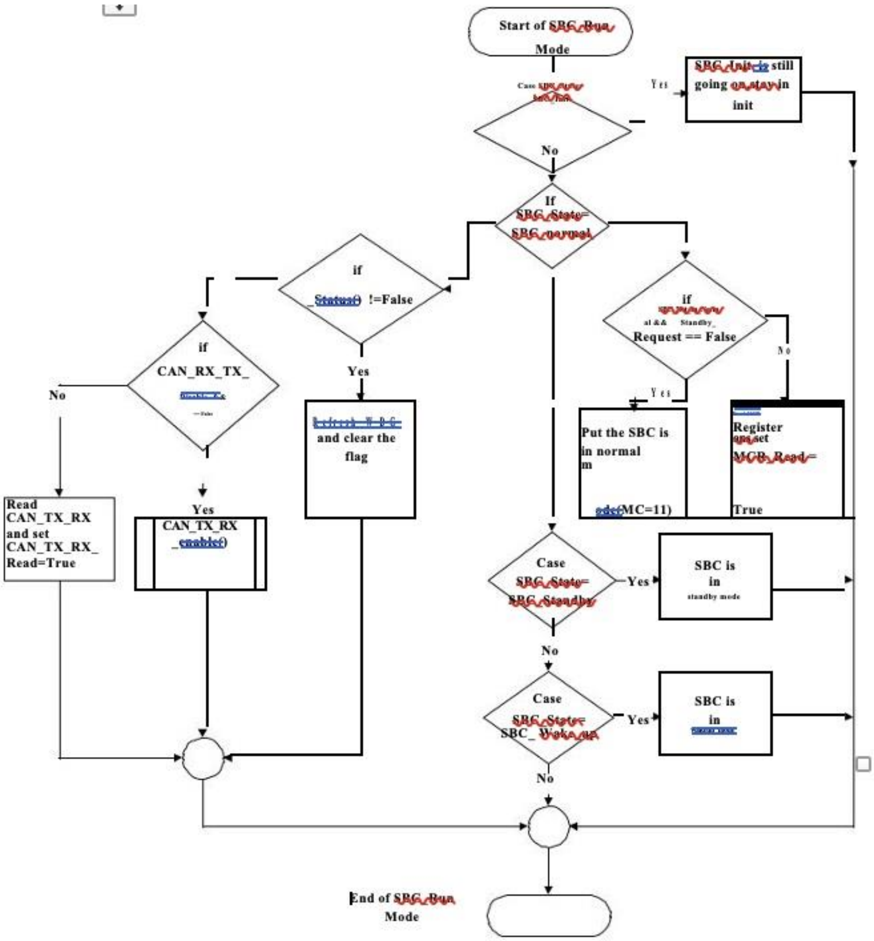

Figure. 5 Flow chart for SBC_Run mode.

The flow of SBC_run mode is shown in Fig.6

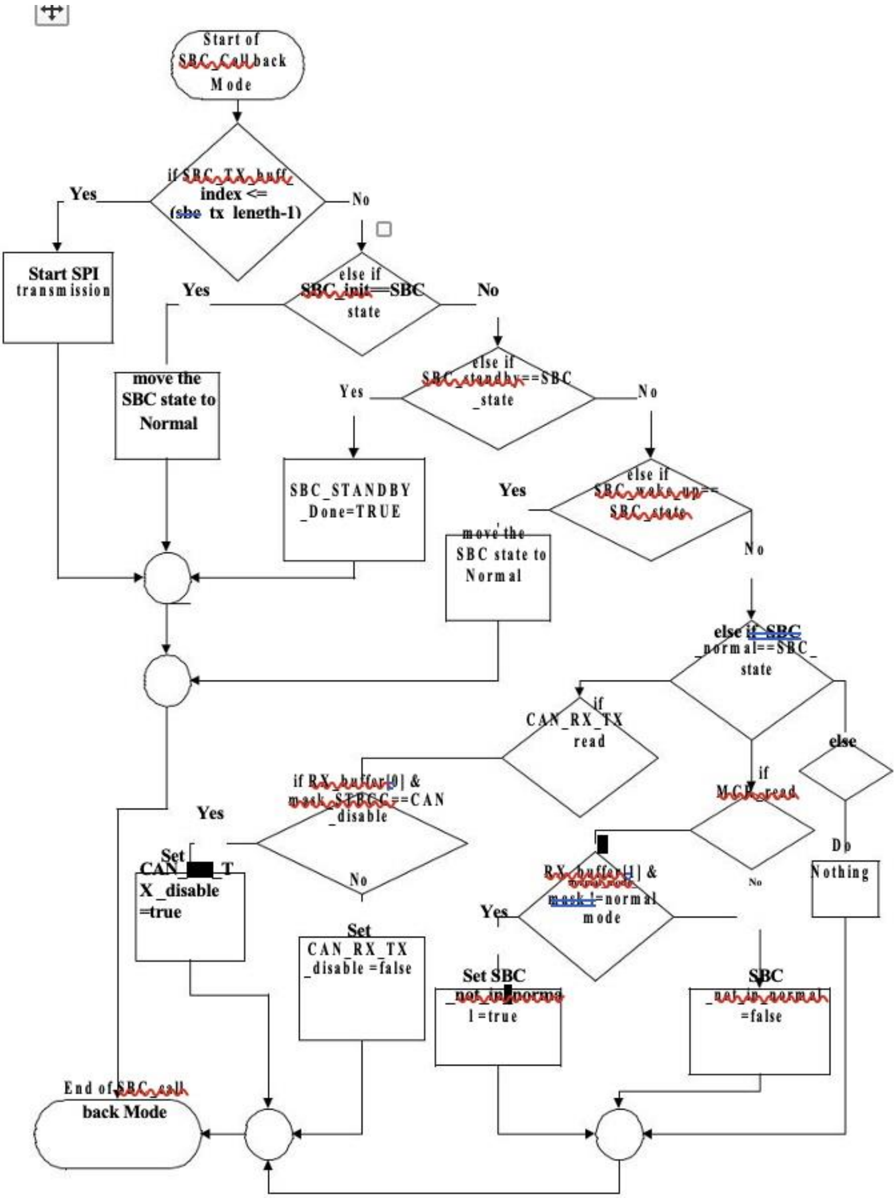

Figure .6. Flow chart for SBC_call back Mode.

6. IAR Embedded Workbench Tool

Using assembly, C, and C++, the IAR Embedded Workbench is a set of development tools for creating and debugging embedded applications [12]. A project manager, editor, build tools, and debugger are all included in the fully integrated development environment offered by IAR Embedded Workbench. One may create source files and projects, develop programmes, and debug them in a simulator or on hardware all in one continuous workflow.

7. Result Analysis and Simulation

After developing driver software for new SBC, each of the SBC pins have to be tapped and monitored using oscilloscope when the Trailer module is awake and asleep to verify the correct working of the SBC. The pins to be monitored are V1, V2, TXDC, RXDC, INTN, CANL, CANH and RSTN. Since Microcontroller communicates through Serial Peripheral Interface(SPI),the MOSI,MISO,SCK and CS pins also needs to be monitored in order to check whether proper communication has been established. SBC communicates with other ECUs in the network through CAN. Hence to verify the correct working of the CAN transceiver in the SBC, this communication channel needs to be verified using Canoe software. In normal mode, TRM management signal value is 4 as shown in Fig.7 which depicts the CANoe CAN trace VDD and CAN Transceiver's supply should be 5v in normal mode. Fig 8 shows the value of VDD and CAN Transceiver as viewed in the oscilloscope. In normal mode,if the communication between SBC and micrcontroller is proper then MOSI,MISO and SCLK pins will have a valid signals . Fig.9 shows the MOSI and MISO signal as viewed in the osilloscope. Fig.10 shows the SCLK signal as viewed in the osilloscope. Fig.11 shows the TXD and RXD signal as viewed in the osilloscope. CAN transceiver is active in normal mode hence CAN H and CAN L pins will have the correct output voltage. Fig.12 shows the CAN H and CAN L signal as viewed in the osilloscope. In sleep mode, TRM management signal value is 32 as shown in Fig.13 which depicts the CANoe CAN trace. In sleep mode, V1=5V and V2 =0V as shown in Fig.14.

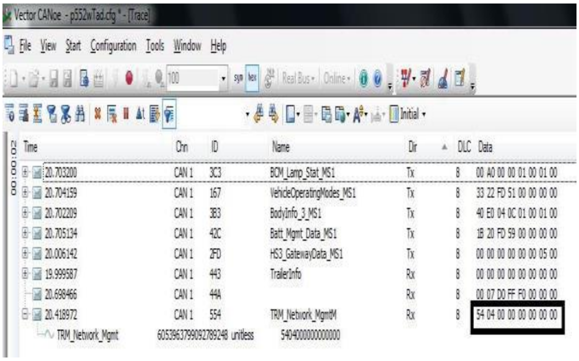

Figure.7 CANoe Trace Window in Normal Mode.

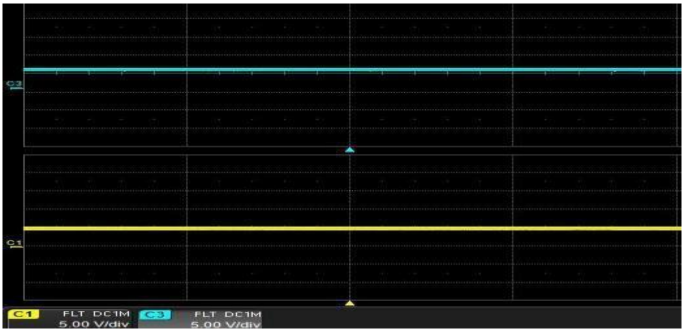

Fig.8 VDD and CAN Transceiver's Supply

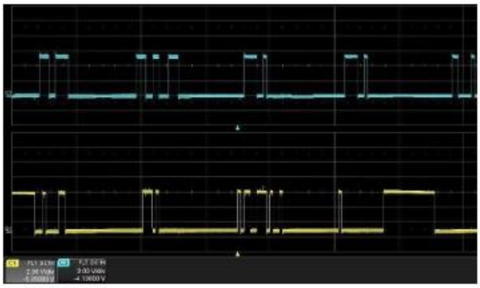

Figure 9 MOSI and MISO Signals

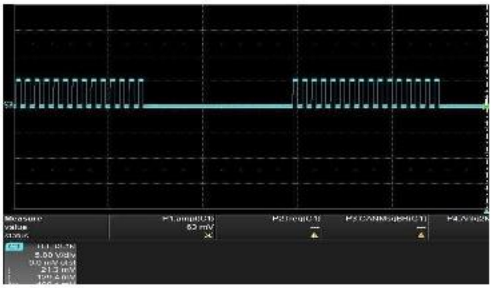

Figure 10 Serial Clock(SCLK) signal

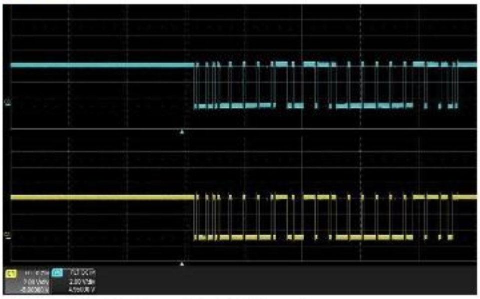

Figure 11 TXD and RXD Signals

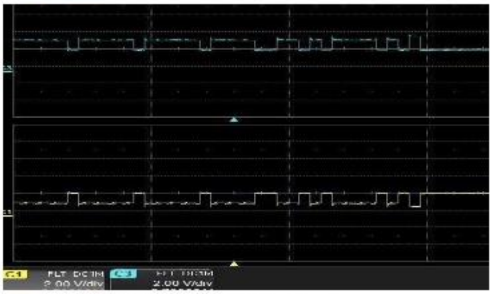

Figure 12 CANH and CANL Signals

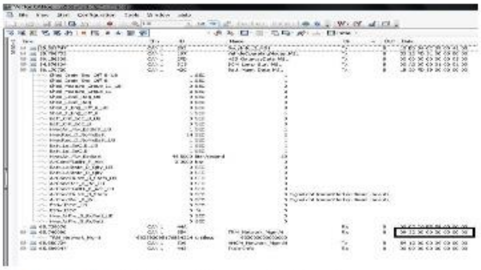

Figure 13 CANoe Trace Window in Sleep Mode

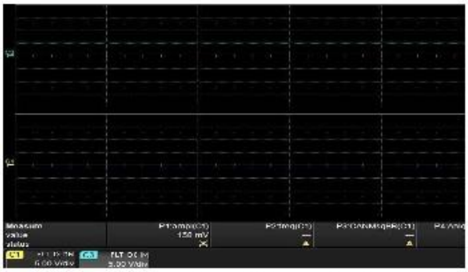

Figure 14 VDD and CAN Transceiver' Supply Voltage

5. Conclusion

Thus the driver software of SBC for trailer module has been implemented and each pins have been tapped and monitored using oscilloscope for proper output voltage at different modes. Before using the SBC in Trailer Module, all the above pins are checked to ensure that the SBC is working properly and the communication between SBC and Microcontroller is proper. After ensuring the proper functioning of SBC, driver software will be integrated to the Application software which controls the loads such as rear fog light, right turn lamp, left turn lamp, battery charge, trailer connected, battery charge, reverse lamp and park position lamp of TRM.

References

[1] UJA1076A (High-speed CAN core system basis chip) Product Data sheet, Technical Data Rev. 2, NXP Semiconductors ,2 – 31 January 2011.

[2] M. Garcia Valls, I. R. Lopez and L. F. Villar, "iLAND: An Enhanced Middleware for Real-Time Reconfiguration of Service Oriented Distributed Real-Time Systems," in IEEE Transactions on Industrial Informatics, 19(1) (2013) 228-236. https://doi.org/10.1109/TII.2012.2198662

[3] EM6151/52 Fact sheet of Windowed Watchdog IC with 5V LDO,EM MICROELECTRONIC, Copyright on 2005.

[4] MC33903 (High Speed CAN Inteface), The Product data sheet , Technical Data Rev. 7, Freescale Semiconductor , September 2011.

[5] Salvatore T. March and Gerald F. Smith, Design and natural science research on information technology, In: Decision Support Systems 15(4) (1995) 251– 266. https://doi.org/10.1016/0167-9236(94)00041-2

[6] Kimoon Kim, Gwangil Geon, Seongsoo Hong, Sunil Kim and Taehyung Kim, Resource-conscious customization of CORBA for CAN-based distributed embedded systems," Proceedings Third IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2000) (Cat. No. PR00607), 2000, pp. 34-41.https://doi.org/10.1109/ISORC.2000.839509

[7] Isabell Jahnich, Ina Podolski, and Achim Rettberg, Towards a Middleware Approach for a Self-configurable Automotive Embedded System. In: Brinkschulte, U., Givargis, T., Russo, S. (eds) Software Technologies for Embedded and Ubiquitous Systems. SEUS 2008. Lecture Notes in Computer Science, vol 5287. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-87785-1_6

[8] Key Stone Architecture Serial Peripheral Interface (SPI) user guide from Texas Instruments, March 2012.

[9] TJA1042 High-speed CAN transceiver with Standby mode Product data sheet Technical Data Rev. 9, NXP Semiconductors , 23 May 2016.

[10] Klaus Pohl et al., eds. Model-Based Engineering of Embedded Systems. Springer Berlin Heidelberg, 2012, p. 229.

[11] Howard Foster, Arun Mukhija, David S. Rosenblum & Sebastian Uchitel, A Model-Driven Approach to Dynamic and Adaptive Service Brokering Using Modes. In:

Bouguettaya, A., Krueger, I., Margaria, T. (eds) Service-Oriented Computing – ICSOC 2008. ICSOC 2008. Lecture Notes in Computer Science, vol 5364. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-89652-4_46 .

[12] IAR Embedded Workbench IDE User Guide for Advanced RISC Machines Limited's ARM Cores from IAR Systems, Copyright on 1999-2009.

[13] S. Lankes, A. Jabs and T. Bernmerl, Integration of a CAN-based connection-oriented communication model into Real-Time CORBA, Proceedings International Parallel and Distributed Processing Symposium, 2003, pp. 8.https://doi.org/10.1109/IPDPS.2003.1213239

[14] M. Panahi, W. Nie and K. -J. Lin, The Design of Middleware Support for Real-Time SOA, 2011 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, 2011, pp. 117-124.https://doi.org/10.1109/ISORC.2011.24

[15] D. Ardagna and B. Pernici, "Adaptive Service Composition in Flexible Processes," in IEEE Transactions on Software Engineering, 33(6) (2007) 369-384. https://doi.org/10.1109/TSE.2007.1011

Funding

No funding was received for conducting this study.

Conflict of interest

The Authors have no conflicts of interest to declare that they are relevant to the content of this article.

About The License