Faculty of Engineering, Alex. University

CAN Module report

MCP2515 CAN Bus Transceiver Interface Module

Kerollos Emad Sobhey **ID: 146**
Faculty of Engineering, Alex. University 1/5/2022 Dr. Mohammed Salah

Abstract:

Modern cars may have up to 100+ sensor and ECUs that have to interact with each other and communicate to transmit data and exchange information to monitoring the automobile status. With more features being added to cars like Smart car systems, Autonomous driving system, V2X system, microcontrollers and ECUs increase by huge number. With sensors units delivering information like speed, position, heat, etc. So Controller Area Network (CAN) communication protocol has been found to be reliable to handle the communication with the possible lower error as the information is critical to avoid accidents and traffic issues. [1]

So this application report represents

- why using Controller Area Network (CAN)
- an introduction to CAN communication protocol
- introduction to MCP2515 and TJA1050
- interfacing CAN Module with STM32F103c8t6 and Raspberry Pi

CAN Module report

MCP2515 CAN Bus Transceiver Interface Module

CAN which stands for Controller Area Network is used to allow microcontrollers and devices to communicate with each other within a vehicle without a host computer. The CAN bus is in fact a good communication protocol for many applications that use multiple distributed MCUs and sensors which need to interact with each other such as in factory automation or robotics. [1]

Why using CAN protocol

A Controller Area Network (CAN) is ideally suitable for industrial communication protocols. Its cost, performance, and upgradeability provide high flexibility in system design. [2]

So there is many advantages of CAN protocol.

The 5 advantages of CAN Protocol:

1. Low Cost

CAN Bus only requires 2 wires to connect all ECUs, sensors and actuators, rather than using wire for each electronic module.

The first vehicle to use CAN bus wiring was the BMW 850 coupe released in 1986. Implementation of CAN bus architecture reduced the length of wiring in the BMW 850 by 1.25 miles, which in turn reduced its weight by well over 100 pounds. Based on the current cost of copper wiring, the total cost savings from the saved materials would amount to nearly $600. [3]

2. Built-in Error Detection

CAN protocol supports centralized control over each node. Nodes can communicate with other nodes on the network by microcontroller, CAN controller, and CAN transmitter.

As the number of nodes may approach to 100+, so the CAN bus protocol uses lossless bit-wise arbitration technique, by the way it will be discussed in details later, to resolve network traffic. So each node should be given “priority” to communicate its messages first.

Error handling is built into the CAN protocol, with each node checking for errors in transmission and maintaining its own error counter. Nodes transmit a special Error Flag message when errors are detected and will destroy the offending bus traffic to prevent it from spreading through the system. Even the node that is generating the fault will detect its own error in transmission, raising its error counter and eventually leading the device to "bus off" and cease participating in network traffic. In this way, CAN nodes can both detect errors and prevent faulty devices from creating useless bus traffic. [3]

3. Robustness

Durability and reliability are key areas of concern when choosing a communication protocol for deployment in your embedded engineering projects. As you deploy your products into the live environment, you'll want to choose a communication protocol that is self-sustaining, with the ability to carry on operating for long periods of time without outside maintenance or intervention.

This need makes the protocol's error detection capabilities particularly advantageous, as they enable systems to identify and recover from errors on their own without intervention from an outside actor. There are five mechanisms for detecting errors in the CAN protocol:

1. Bit monitoring
2. Bit stuffing
3. Frame check
4. Acknowledgment check
5. Cyclic redundancy check

CAN high-speed bus lines are highly resistant to electrical disturbances, and the CAN controllers and transceivers that communicate with electronic devices are available in industrial or extended temperature ranges.

A CAN bus cable is typically vulnerable to the failure modes listed in the ISO 11898 standard, such as:

1. CAN_H interrupted
2. CAN_L interrupted
3. CAN_H shorted to battery voltage
4. CAN_L shorted to ground
5. CAN_H shorted to ground
6. CAN_L shorted to battery voltage
7. CAN_L shorted to CAN_H wire
8. CAN_H and CAN_L interrupted at the same location
9. Loss of connection to the termination network

While most CAN transceivers will not survive these types of failures, some electronics manufacturers have constructed fault-resistant CAN transceivers that can handle all of them, though they may have a restricted maximum speed as a trade-off. Together, these features expand the suitability of CAN bus networks for applications in the most rugged and demanding environments. [3]

4. Speed

When the CAN protocol was first defined, it was described in three layers: the object layer, the physical layer, and the transfer layer. Later, when the CAN specification was created, specific definitions for the physical layer were excluded. This gave engineers the flexibility to design systems with transmission mediums and voltages that suited their intended applications. Later, to help drive adoption of CAN devices and networks, standards were finally released for the CAN physical later in the form of ISO 11898-2.

There are currently two defined physical layer standards, two types of CAN protocol, each with its own advantages and disadvantages.

High Speed CAN offers signal transfer rates of between 40 kbps and 1 Mbps, depending on the length of the cable. CAN-based bus protocols like DeviceNet and CANopen use this physical standard to support simple cable connections with high-speed data transfer.

Low Speed CAN network offers lower signal transfer rates that may start at 40kbps but are often capped at or near 125 kbps. The lower signaling rates allow communication to continue on the bus, even when a wiring failure takes place. While high-speed CAN networks terminate at either end of the bus line with a 120-ohm resistor, each device in a low-speed CAN network has its own termination. Low Speed CAN network exhibits greater fault tolerance and are vulnerable to fewer failure modes, but slower transfer speeds make them poorly suited to networks that require rapid and frequent communication. [3]

5. Flexibility

To appreciate the flexibility of the CAN bus protocol in communications, we need to differentiate between address-based and message-based protocols. In an address-based communication protocol, nodes communicate directly with each other by configuring themselves onto the same protocol address.

The CAN bus protocol is known as a message-based communication protocol. In this type of protocol, nodes on the bus have no identifying information associated with them. As a result, nodes can easily be added or removed (a process called hot-plugging) without performing any software or hardware updates on the system.

This feature makes it easy for engineers to integrate new electronic devices into the CAN bus network without significant programming overhead and supports a modular system that is easily modified to suit your specs or requirements. [3]

Introduction to CAN communication protocol

The CAN bus was developed by BOSCH (1) as a multi-master, message broadcast system that specifies a maximum signaling rate of 1 megabit per second (bps). Unlike a traditional network such as USB or Ethernet, CAN does not send large blocks of data point-to-point from node A to node B under the supervision of a central bus master. In a CAN network, many short messages like temperature or RPM are broadcast to the entire network, which provides for data consistency in every node of the system.
Once CAN basics such as message format, message identifiers, and bit-wise arbitration -- a major benefit
of the CAN signaling scheme are explained, a CAN bus implementation is examined, typical waveforms presented, and transceiver features examined. [2]

The CAN Standard

;CAN is an International Standardization Organization (ISO) defined serial communications bus originally developed for the automotive industry [4] to replace the complex wiring harness with a two-wire bus. The specification calls for high immunity to electrical interference and the ability to self-diagnose and repair data errors. These features have led to CAN's popularity in a variety of industries including building automation, medical, and manufacturing.

The CAN communications protocol, ISO-11898: 2003, describes how information is passed between devices on a network and conforms to the Open Systems Interconnection (OSI) model that is defined in terms of layers. Actual communication between devices connected by the physical medium is defined by the physical layer of the model. The ISO 11898 architecture defines the lowest two layers of the seven layer OSI/ISO model as the data-link layer and physical layer in Figure 3.

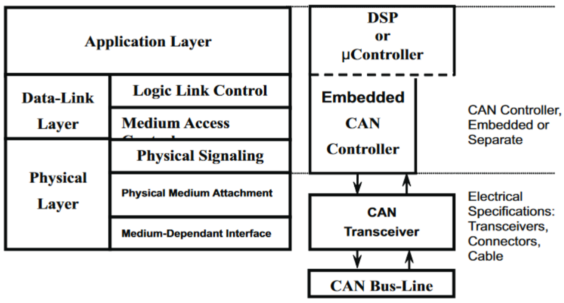

Figure 2: The Layered ISO 11898 Standard Architecture. [4]

The Bit Fields of Standard CAN

- SOF- Start of Frame
- 11-bit Identifier- establishes the priority of the message. The lower the binary, the higher its priority.
- RTR- Remote Transmission Request is dominant when information is required from another node.
- IDE-dominant single identifier extension bit means that a standard CAN identifier with no extension is being transmitted.
- R0–Reserved bit.
- DLC–The 4-bit data length code.
- Data–Up to 64 bits.
- CRC–The 16-bit cyclic redundancy check.
- ACK-Acknowledgment bit
- EOF-End of Frame.
- IFS–This 7-bit interframe space (IFS) contains the time required by the controller to move a correctly received frame to its proper position in a message buffer area.

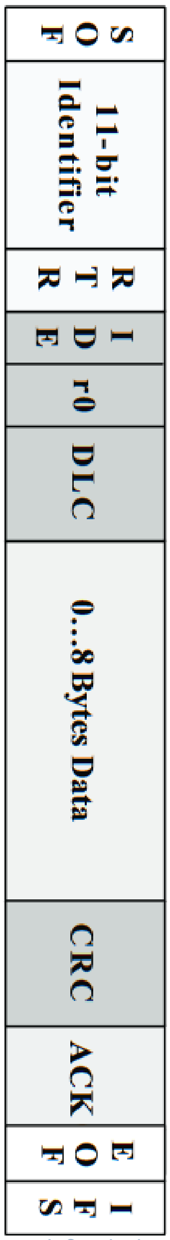

Figure 1: Standard CAN: 11-bit identifier

Introduction to MCP2515 and TJA1050

MCP2515 CAN Bus Transceiver Interface Module is mainly based on MCP2515 CAN Bus Controller, and TJA1050 CAN Transceiver.

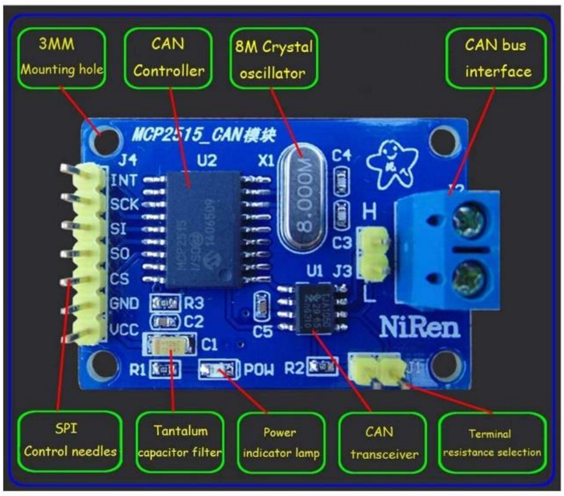

Figure 3: MCP2515 CAN BUS MODULE

MCP2515 CAN BUS Controller

Description

Microchip Technology's MCP2515 is a stand-alone Controller Area Network (CAN) controller that implements the CAN specification, version 2.0B. It is capable of transmitting and receiving both standard and extended data and remote frames. The MCP2515 has two acceptance masks and six acceptance filters that are used to filter out unwanted messages, thereby reducing the host MCUs overhead. The MCP2515 interfaces with microcontrollers (MCUs) via an industry standard Serial Peripheral Interface (SPI). [5]

Package Type

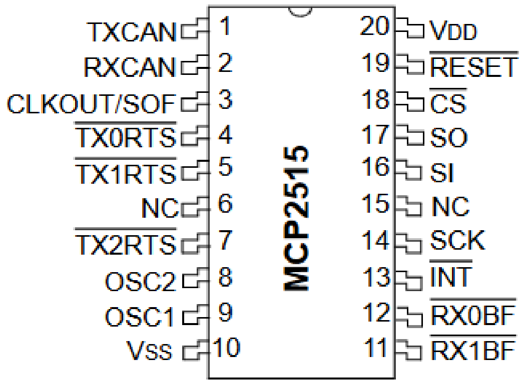

Figure 4: 20-LEAD TSSOP

TJA1050 High speed CAN transceiver

Description

The TJA1050 is the interface between the Controller Area Network (CAN) protocol controller and the physical bus. The device provides differential transmit capability to the bus and differential receive capability to the CAN controller.

The TJA1050 is the third Philips high-speed CAN transceiver after the PCA82C250 and the PCA82C251.

The most important differences are:

- Much lower electromagnetic emission due to optimal matching of the output signals CANH and CANL
- Improved behavior in case of an unpowered node
- No standby mode.

This makes the TJA1050 eminently suitable for use in nodes that are in a power-down situation in partially powered networks. [6]

Connection between CAN CONTROLLER and TRANSCEIVER

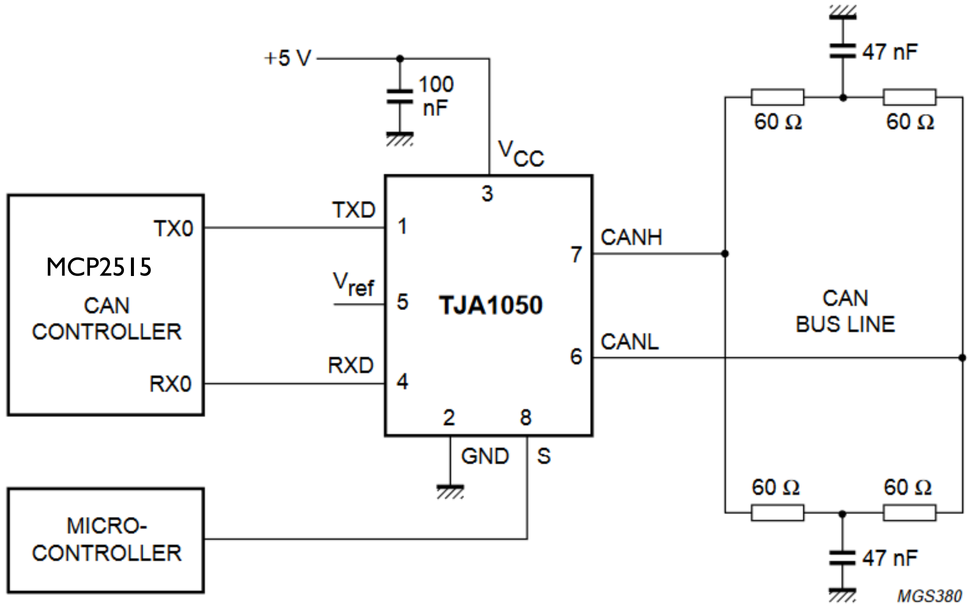

Figure 5: Schematic of interfacing the MCP2515 with TJA1050

Interfacing CAN Module with STM32F103c8t6 and Raspberry Pi

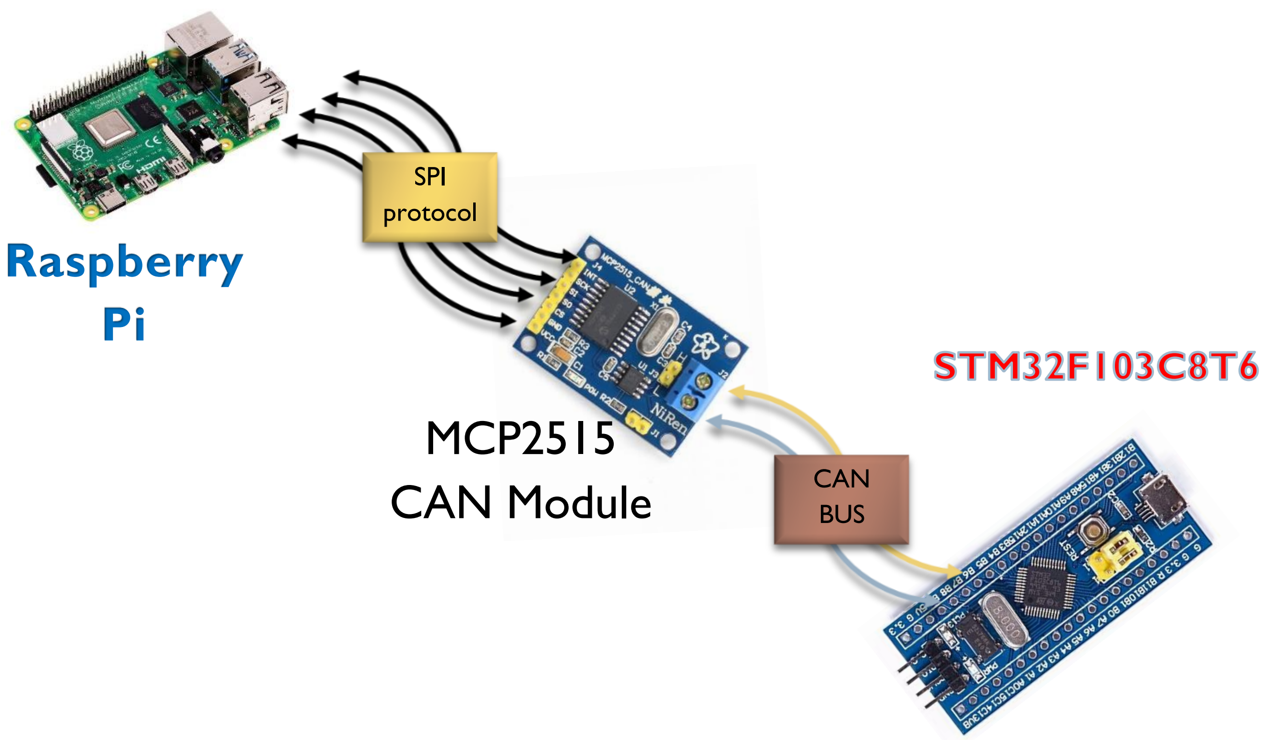

Figure 6: Interfacing RP, STM32F103, and MCP2515 CAN Bus Module

References

[1] F. Electronics, "Future Electronics Egypt," Electronic Components , 18 7 2015. [Online]. Available: https://store.fut-electronics.com/products/mcp2515-can-bus-interface-module. [Accessed 5 1 2022].

[2] S. Corrigan, "Introduction to the Controller Area Network (CAN)," Texas Instruments, Texas, 2002.

[3] StaffWriter, "TotalPhase," 20 August 2019. [Online]. Available: https://www.totalphase.com/blog/2019/08/5-advantages-of-can-bus-protocol/. [Accessed 5 1 2022].

[4] I. 2. 31, " ISO 11898-1:2003," 12 2003. [Online]. Available: https://www.iso.org/standard/33422.html. [Accessed 5 1 2022].

[5] Microchip, *Stand-Alone CAN Controller With SPI™ Interface,* Chandler,Arizona: Microchip, 2005.

[6] P. Semiconductors, *TJA1050,* Eindhoven, Netherlands: Philips Semiconductors, 2003.

TABLE OF FIGURES

Contents