Industrial CAN I/O Module

Jaromir Chocholac & Zdenek Kaspar
Motorola Semiconductor Product Sector

Introduction

Manufacturers have a constant objective to send their new products to market as quickly as possible. In accordance with this fact, the trend in manufacturing plants is to operate faster, more efficiently and more flexibly. Automation is a key component for accomplishing this goal. To achieve automation, manufacturers are implementing an open distributed architecture over control devices and systems. This paper describes the Industrial CAN I/O Module, which provides inputs and outputs for distributed automation control. It also discusses data exchange between connected modules, which is performed through the Controller Area Network (CAN) bus.

Concept of the Module

The module described here is based on Motorola's HCS12 microcontroller. It can be used as a hardware platform for high-level communication protocol software development. In addition, the module enables the implementation and testing of user software. For this purpose, the board is equipped with a Background Debug Mode (BDM) interface for reprogramming and debugging. The module contains analog inputs, digital inputs, digital outputs, a controller with a CAN transceiver, and optional RS232 and RS485 transceivers. The CAN interface is compatible to ISO 11898 and allows a maximum data transfer rate of 500 kbit/s. The block diagram of the module can be seen in Figure 1.

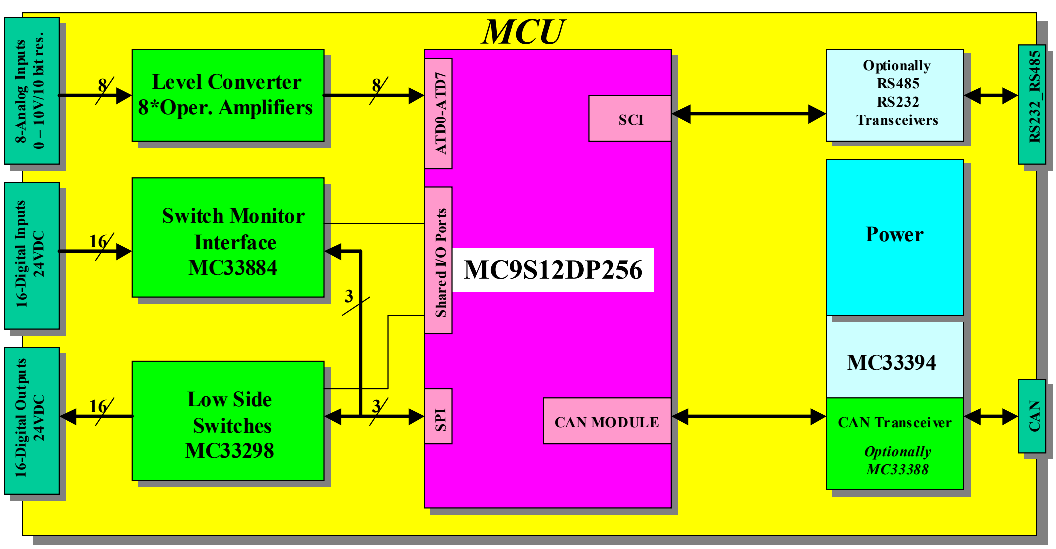

Figure 1. Industrial CAN I/O Module Block Diagram

The Industrial CAN I/O Module is a modular system. It is logically divided into the following three basic boards:

- Base Board
- Power Supply
- I/O Board

Data transfer between the boards is ensured by the Serial Peripheral Interface (SPI) protocol.

The Base Board
The main function of the base board of the Industrial CAN I/O Module is to control the module and communicate with the system control unit. This board is equipped with eight analog channels.

The Base Board block diagram can be seen in Figure 2.

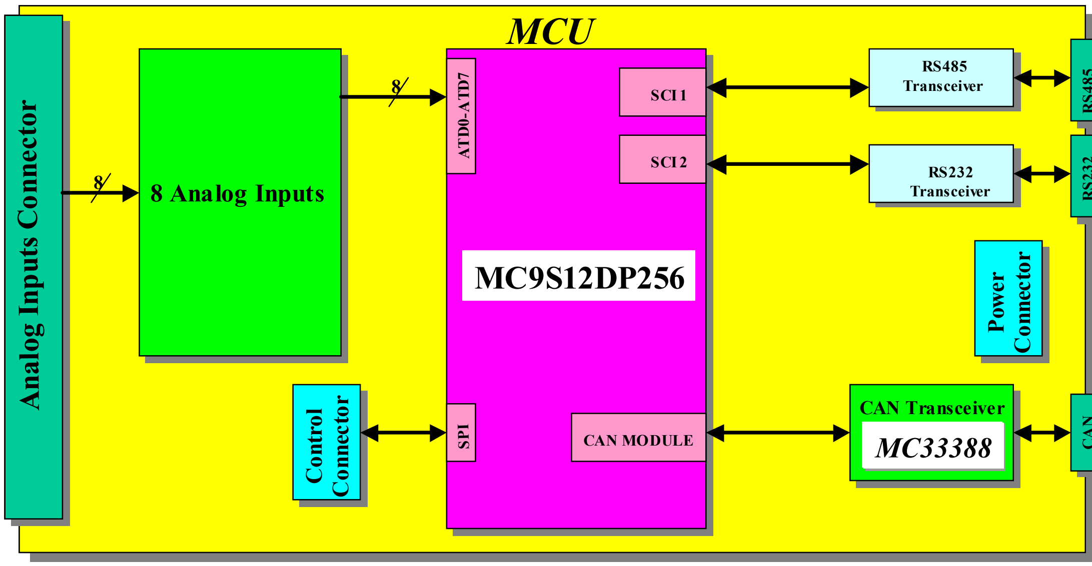

Figure 2. The Base Board Block Diagram

This board is logically divided into following four basic blocks:
- Microcontroller block
- CAN Interface
- Analog Inputs
- RS232_485 Interface

Microcontroller block
Motorola's 16-bit MC9S12DP256 microcontroller unit (MCU) is the main component of the Base Board. For further details about the microcontroller, please see [1], [9].

The Industrial CAN I/O Module uses one of five CAN peripheries (CAN0) built into the MCU. The periphery is a communication controller implementing the CAN 2.0 A/B protocol, as defined in the BOSCH specification – refer to [5]. CAN periphery is a specific implementation of the Motorola Scalable CAN (MSCAN). It uses 2 external pins, 1 input (RxCAN0) and 1 output (TxCAN0) to communicate with CAN bus through the CAN transceiver. One output signal is assigned to the CAN transceiver chip control.

As mentioned above, data transfer between the boards is ensured by the SPI protocol. The SPI module allows full-duplex, synchronous, serial communication between the MCU and peripheral devices. The Industrial CAN I/O Module uses one of three SPI peripheries (SPI0) of the MCU. The SPI can be configured to operate as a master or as a slave. The master mode must be selected for the SPI0 periphery because only a master SPI can initiate transmissions with the peripherals.

The Industrial CAN I/O Module uses one of two 8 channel Analog-to-Digital (A/D) Converters (AN0) built into the MCU. The A/D module performs analog to digital conversions. This module contains all the necessary analog and digital electronics to perform a single analog to digital conversion. The resolution of the A/D converter is program selectable at either 8 or 10 bits. It is capable of accepting 5 volts inputs without permanent damage while simultaneously operating at 5 volts.

The board provides an 8-position DIP-switch for configuring the Address of the Node (Node ID), as well as the CAN speed.

Single-wire communication with host development system is done through the Background Debug Mode (BDM) system implemented in on-chip hardware. Connection to the target system is made through the standard six pins BDM Connector.

The CAN Interface

Each CAN Node is connected physically to the CAN bus line through a transceiver chip. The transceiver is capable of driving the large current needed for the CAN bus, and has current protection against defected CAN or defected stations. There are two possible options for the CAN transceiver on the board. One of them is to use Fault Tolerant CAN Interface MC33388; the other one is to use the high speed CAN transceiver part of the MC33394 device. The Module uses high-speed CAN transceiver part of the MC33394 device, which is part of the Power Board and will be described later in this paper.

The Analog Inputs

The module provides eight analog channels. There is a passive low-pass filter with an attenuation slope of -40dB/dec, and with a cut off frequency of 1kHz, in the input of each analog channel.

This input filter is used to eliminate unwanted high-frequency noise and interference, introduced prior to sampling.

The analog channel can be configured for voltage or current mode. The resistor 250 ohm provides current sensing, as well as substantial over-current protection, for 4-20mA current loops. The analog channel provides a voltage input signal in the range 0 to 10V, 0 to 5V, -5V to +5V, -2.5V to +2.5V, or a current in the range 4 to 20 mA. The input range is controlled by the micro-controller.

The RS232_485 Interface

The Base Board provides an RS-232 interface for connection to a PC, or a similar host, as well as an RS-485 interface that can be used for industrial applications.

The Power Supply

The power supply is designed to meet all the power supply needs of Industrial CAN I/O Module. The schematic of the power supply can be seen in Figure 3.

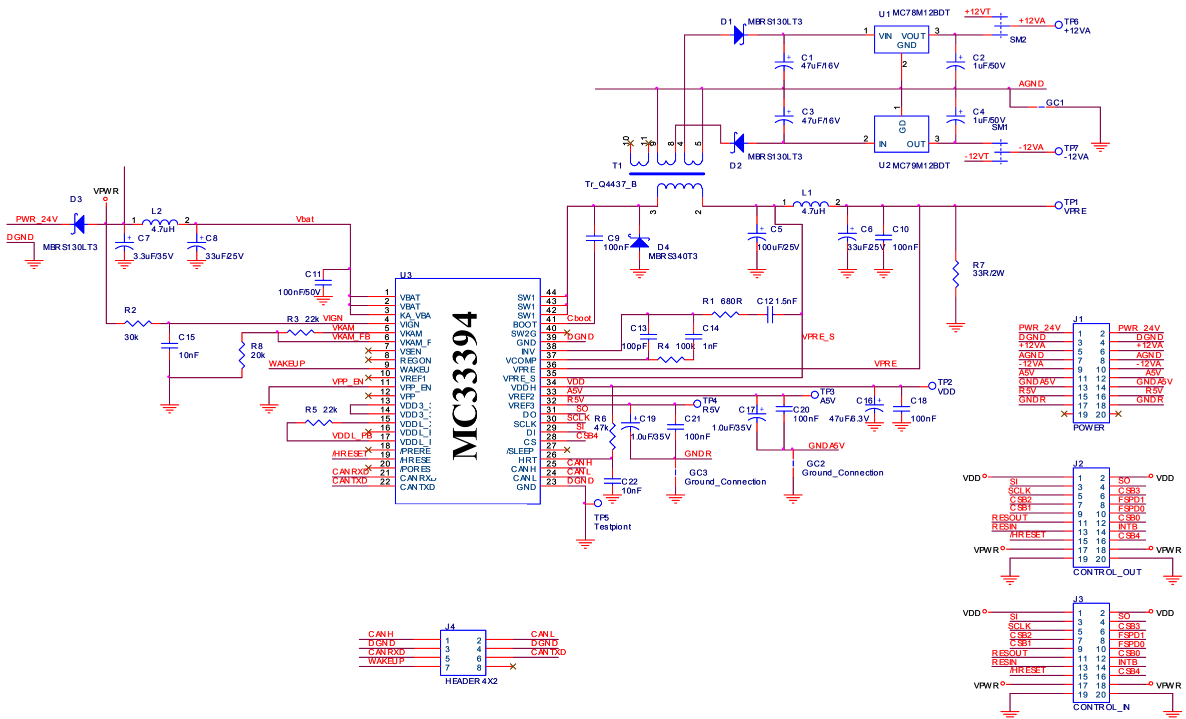

Figure 3. Power Supply Board Schematic

The main device of the board is the PC33394, a multi-output power supply integrated circuit with a high speed CAN transceiver. This device incorporates a switching regulator to supply a pre-regulated output, followed by Low Drop Out (LDO) regulators. The PC33394 device is not fully utilized for this application.

The internal switching regulator incorporates circuitry to implement a Buck or a Buck/Boost regulator. Only the Buck regulator is implemented on this board. The fly-back converter provides symmetrical voltages to supply the analog channels of the module. The output voltage is derived through the transformer, rectified, and the linear regulators generate a symmetrical +12V, -12V output voltage.

A high speed CAN transceiver physical layer interfaces the microcontroller CMOS outputs and the differential bus lines. The CAN driver is short circuit protected and tolerant of loss of battery or ground conditions.

The I/O Board

The I/O Board provides 16 digital inputs and 16 digital outputs. The block diagram of the I/O Board can be seen in Figure 4. The I/O Board is connected through the connector J1 to the Base Board. The I/O Board is logically divided into the following two basic blocks:

- Digital Inputs
- Digital Outputs

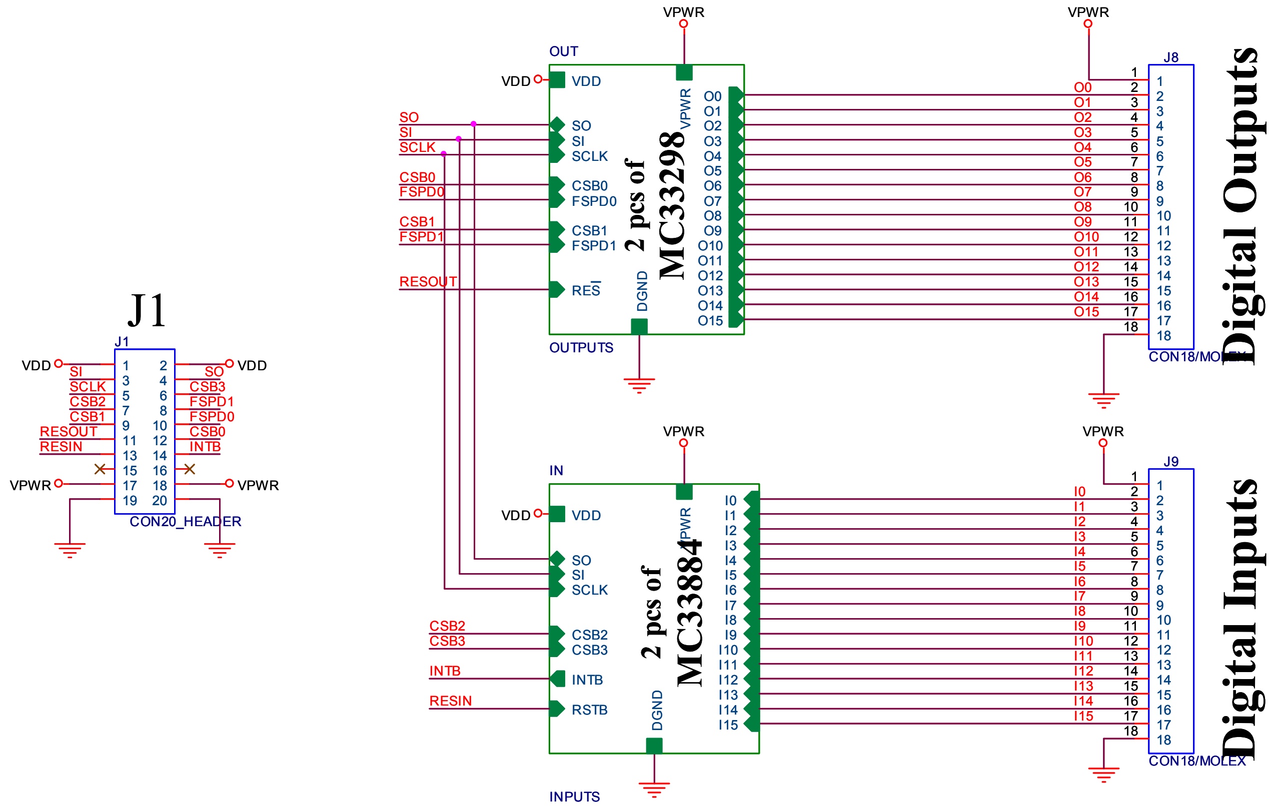

Figure 4. I/O Board Schematic

The Input Block

Two Switch Monitor Interface MC33884 devices on the board provide an interface between electrical switches and the microcontroller. The MC33884 monitors the OPEN/CLOSED status of multiple external switches used in the system. The device supplies switch contact pull-up and pull-down current, while monitoring the input voltage level. All inputs are protected for transients with a static discharge capacitor used on the inputs.

The MC33884 device can run in one of four modes of operation - Sleep, Normal, Polling, and Polling + INT Timer. All modes of operation are programmed via the SPI control. The response to an SPI command returns Switch Status, and Mode settings. More details can be found in the datasheet for MC33884/D [7].

The Module uses 4 programmable Switch-to-Ground or Battery sense inputs, and 4 Switch-to-Ground sense inputs of each MC33884 device. The devices on the board are used in parallel configuration.

The Output Block

Two low side power switch MC33298 devices on the board enable MCU to direct control of various inductive or incandescent loads. The devices on the board are programmed via the SPI control, and are used in parallel configuration. The response to an SPI command returns the status of the device's output switches. LEDs D1 to D16 also signal the status of the device's output switches. Flashing LED indicates ON status.

Software introduction

Software content of the system can be divided into two basic groups. The first group consists of all initialization routines necessary to configure both MCU peripherals and all sub-modules of the system. The second part consists of the routines of the Industrial CAN I/O board demo application.

Initialization and Configuration Introduction

The routines of the first group (responsible for initialization and configuration) include:

- Motorola Scalable CAN (msCAN) periphery module initialization (including module addressing, baud-rate selection, CAN identifiers settings, msCAN message objects definition)
- Configuration of the CAN Interface block of the Base Board Configuration of the I/O Board
- Analog to Digital converter (ADC) periphery module initialization (including accuracy)
- Configuration of the Analog Inputs block of the Base Board (voltage / current loop mode setting and range selection of voltage mode)
- Serial Peripheral Interface (SPI) periphery module initialization for communication with devices of I/O Board
- Configuration of the I/O Board (digital inputs MC33884 device and digital output MC33298 device) via the SPI channel
- Configuration of the PC33394 multi-output Power Supply with integrated high speed CAN transceiver via the SPI channel
- Serial Communication Interface (SCI) periphery module initialization for RS232_485 Interface block of the Base Board
- Real Time Interrupt (RTI) module initialization for demo application

Application Introduction

The aim of the demo application is to show main features of the module (16 digital and 8 analog inputs, 16 digital outputs) together with the utilization of the CAN connectivity. The module receives the configuration messages from the superior device and informs it back about the status of inputs/outputs. A list of all events of the module is in the following Table 1.

Event description	Initiator	Recipient
Change of Analog Inputs block parameters (range, accuracy)	Superior device	CAN I / O
Status of Analog Inputs block values	CAN I / O	Superior device
Set new Digital Output values of I/O Board	Superior device	CAN I / O
Status of Digital Inputs values of I/O Board	CAN I / O	Superior device
Status of Digital Inputs values of I/O Board due to a change-of-state	CAN I / O	Superior device

Table 1: List of application events

A straightforward structure of the messages based on the CAN 2.0A 11-bit long identifiers [3] was created for all events listed in Table 1. Every message type has a definition of its identifiers as can be seen in Table 2. Note that the definition of the Group number and used identifiers structure are chosen in correspondence with the DeviceNet specification [4]. The sign "*a*" in CAN identifier definition stands for one bit of Node Address (NodeID) as mentioned in Micro-controller section.

Message type	Group number	Message identifier	CAN Identifier	Message object number
Analog Inputs configuration	2	101	10aaaaaa101	1
Analog Inputs status - part 1	1	1000	01000aaaaaa	3
Analog Inputs status - part 2	1	1001	01001aaaaaa	4
Digital Output configuration	2	010	10aaaaaa010	0
Digital Inputs status	1	0100	00100aaaaaa	2
Digital Inputs change-of-state	1	0001	00001aaaaaa	5

Table 2: List of message types and its parameters

Description of Software Modules

While some of the initialization/configuration routines listed in previous section are elementary, some of them need more characterization. First, it is worth mentioning that except for the msCAN periphery module all other peripheries are served directly without the use of any low level drivers. For the msCAN module, the msCAN Driver Software [2] was successfully used to create both initialization as well as the application routines more readable while total cycle time was rapidly reduced. Typical settings of the msCAN module based on the msCAN Driver Software are as follows:

1. Set *CANBTR0_Def* variable for desired CAN baud-rate according to the value of the DIP-switch set by the user.
2. Read *NodeID* (node address) from the DIP-switch of the device.
3. Set array variable of CAN identifiers *M_Identifier_CAN0[]* according to the *NodeID* information and desired structure of used message identifies as shown in Table 2 for all six used message types.
4. Configure six message objects (entities for handling CAN messages) using *CAN_ConfigMB* function, one for each message type. This configuration consists of assigning the message type and a direction of communication (for reception or transmission) for message objects.

As mentioned above, the configuration of MC33394 device [6] as well as the configuration and control of MC33884 [7] and MC33298 [8] devices are done through the SPI channel. Because of the fact that each device was originally developed for different customers, there are a couple of dissimilar SPI format related parameters, which are essential to set properly according to their documentation.

At first, it is important to set the *Master mode* of the SPI device to be equal to 1 as mentioned in the Micro-controller chapter. The *SPI Baud rate* setting (value of the *SPI Serial clock* called *SCLK*) is set to the value, which is suitable for each of three devices. And finally, for all of serviced devices the *Clock polarity value* of the *SCLK* signal has to be set to be active in high while *SCLK* idles in low state.

First of the different settings is the format of SPI; while for MC33298 (Digital Outputs) and MC33884 (Digital Inputs) devices the most significant bit is transferred first, for the MC33394 the first transferred bit it the least significant one. The second divergence is in the *SPI Clock phase* *shift* bit settings; for MC33394 and MC33884 devices this value has to be equal to 0 (the first *SCLK* edge is issued one-half cycle into the 8-cycle transfer operation) while MC33298 device needs that value to be 1 (the first *SCLK* edge issued at the beginning of the 8-cycle transfer operation). And the last difference is the fact that the Digital Outputs MC33298 device uses 8-bit long SPI format of communication while the resting two devices require 16-bit long.

The application itself is separated into the following tasks:

- Detection of the change of state of the Digital Inputs module and proper handling of that state via the dedicated CAN message (linked with message object number 5)
- Periodical signaling of the status of the Digital and Analog Inputs via CAN messages (tied with message objects number 2, 3 and 4); timing based on the *Time Base module* (*TBM*) interrupt of MCU
- CAN reception and proper handling of all types of configuration messages (message object number 0 and 1)

Literature:

[1] MC9S12DP256, Advanced information, Revision 1.1, Motorola 2000
[2] msCAN Driver Software 1.0, User Manual, Metrowerks
[3] CAN specification 2.0, Can in Automation
[4] DeviceNet specification, Release 2.0, Open DeviceNet Vendor Association, Inc.
[5] CAN Specification, Version 2.0, Robert Bosch GmbH, 1991
[6] MC33394 Switch Mode Power Supply with Multiple Linear Regulators and High Speed CAN Transceiver – Motorola Datasheet
[7] MC33884 Switch Monitor Interface – Motorola Datasheet MC33884/D
[8] MC33298 Octal Serial Switch with Serial Peripheral Interface I/O – Motorola Datasheet MC33298/D
[9] www.mot.com/semiconductor, Motorola 2002

Contact address:

Motorola Czech Systems Laboratories
Jaromir Chocholac, Zdenek Kaspar
1. maje 1009
756 61 Roznov p. R. / Czech republic
jaromir.chocholac@motorola.com
zdenek.kaspar@motorola.com