CAN/Stepper Motor Demonstration

Fujitsu Microelectronics, Inc.

Introduction

Fujitsu has very powerful 16-bit MCU products for automotive applications. Two product lines intended for these applications are the MB90590 and MB90595 series. These products have built-in CAN controllers and stepper-motor drivers, which are both required by dashboard applications.

This application note describes a demonstration to show one way an MB90F598 can be used. The requirements for the demo are that it smoothly control the stepper motor, and that it send stepper-motor-related messages by way of the CAN interface.

Control Scheme of the Demo

As shown in Figure 1, the configuration for the demo consists of two Flash/CAN 100 boards, a general type four-wire stepper motor, a PC, one CAN interface cable, and one RS232 cable.

CAN_NODE_1 (see Figure 1) is connected to the PC through the UART interface. This enables the PC to display the function menu and to poll the selected menu entry. Depending on the menu entry, CAN_NODE_1 goes to the corresponding service routine to achieve the selected function.

The service routines do the following: create the corresponding message identifier, such as the stepper motor message identifier (SM_ID) and the LED flash message identifier (LED_ID); prepare the message buffer to store the related control data; and transfer the corresponding control data to other nodes through the CAN interface.

CAN_NODE_1 runs the CAN status monitor, which is activated by reload timer. Every 2 microseconds, CAN_NODE_1 checks the CAN controller status. The possible status conditions CAN_NODE_1 will encounter are Error Active, Warning, Error Passive, and Bus Off. The result of a status check is presented by the I/O Port 4_1, which is connected to an LED. The LED flash pattern shows the current CAN node status. The flash patterns are:

```
No flash:            Error Active (everything OK)

Two short flashes:   Error Warning (still OK)

Three short flashes: Error Passive (something
                     wrong, e.g. only running node
                     in net)

One long flash:      Bus Off (heavy problem)
```

The status checking feature provides a CAN management layer that can be used for further complex applications.

CAN_NODE_1 acts as a data acquisition unit that collects data through sensors and the A/D converter. In an actual implementation, the collected data could be engine speed, engine temperature, fuel position in the tank, and open/close information for doors and windows. The data can be processed in the local MCU or sent directly to a remote MCU through the CAN interface.

CAN_NODE_2 is an execution node. Depending on the message identifier and data sent by CAN_NODE_1, CAN_NODE_2 fetches data from the appropriate message buffer and activates the corresponding service routine. For example, the service routine can drive the stepper motors in different steps (functions) or light the LEDs in different patterns. Because of limited resources, this demo uses only one stepper motor running in different modes.

The menu of activities for this demo includes Stepper Motor Random Mode, Stepper Motor Regular Mode, and LED Mode, as described on the following page.

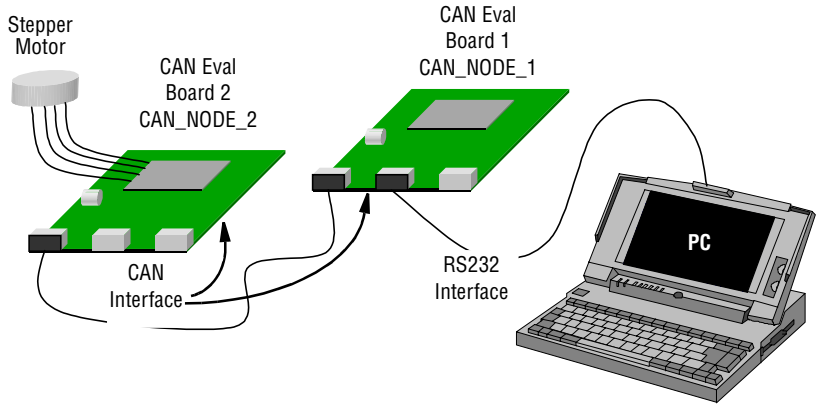


Figure 1. The Demo Configuration

CAN/Stepper Motor Demonstration

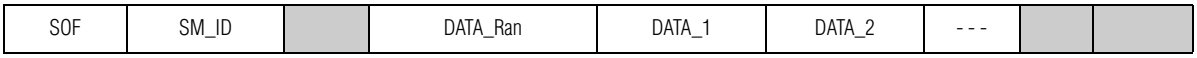
Stepper Motor Random Mode:
In this mode, CAN_NODE_1 randomly generates the stepper control data and transfers this data to CAN_NODE_2 through the CAN interface. CAN_NODE_2 drives the stepper motor according to the data it receives from CAN_NODE_1. The stepper motor turns in different directions and different steps.

A random function simulates CAN_NODE_1, acquiring data from the sensors and through the A/D channels. The data is processed and transferred in terms of a stepper motor rotation position or degrees. CAN_NODE_2 drives and controls the stepper motor according to the received information.

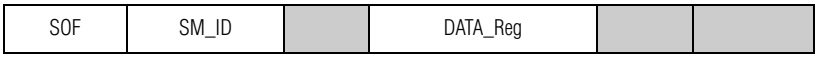
Stepper Motor Regular Mode:
In this mode, CAN_NODE_1 requests CAN_NODE_2 to drive the stepper motor smoothly by sending a specific data frame by way of the CAN interface. The stepper motor turns in a fixed direction and in fixed steps and degrees. This mode shows the smooth and accurate control of the built-in stepper motor function.

LED Mode:
The LED mode is used to demonstrate how, depending on the received message identifier, CAN_NODE_2 filters the received message and provides data to the corresponding service routine. For example, if a remote sensor detects one of the car doors open, it lights the related indicator on the dashboard. The LED information is identified by its unique ID (LED_ID). For this demo, CAN_NODE_1 sends the LED message identifier and related information to CAN_NODE_2, which switches on the LED pattern in response.

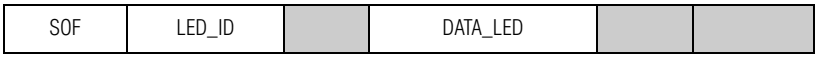
The data frame format for Stepper Motor Random Mode:



The data frame format for Stepper Motor Regular Mode:



The data frame format for LED Mode:



Software Control Flow

Figures 2 and 3 show the software control flow for CAN_NODE_1 and CAN_NODE_2, respectively.

Message_Buffer_13, Message_Buffer_14, and Message_Buffer_15 store the data of the corresponding modes, respectively. The message acceptance filter is set as Full Bit Comparison. This demo uses an 11-bit standard identifier. Depending on the user application, message transmission and reception can be configured as polling or achieved by interrupt service routine.

Future Work

FMI plans to release more application notes describing CAN demonstrations. Below is a list of topics that will be covered in future application notes:

- Add additional CAN nodes in order to show the different types of messages transferred in a distributed CAN network. Be able to clearly show features such as message filtering and message priority.
- Add remote data frame and multilevel buffer group features in order to demonstrate the Fujitsu CAN controller's complete message control scheme.
- Modularize the software functions for the CAN controller and stepper-motor driver.
- Design a PC application that can show the status of CAN nodes in real time. This would be a high-level application for CAN management and CAN network configuration.

Conclusion

The CAN/stepper motor demo in this application note shows how the MB90F598 can smoothly control the stepper motor and send messages with the CAN interface.

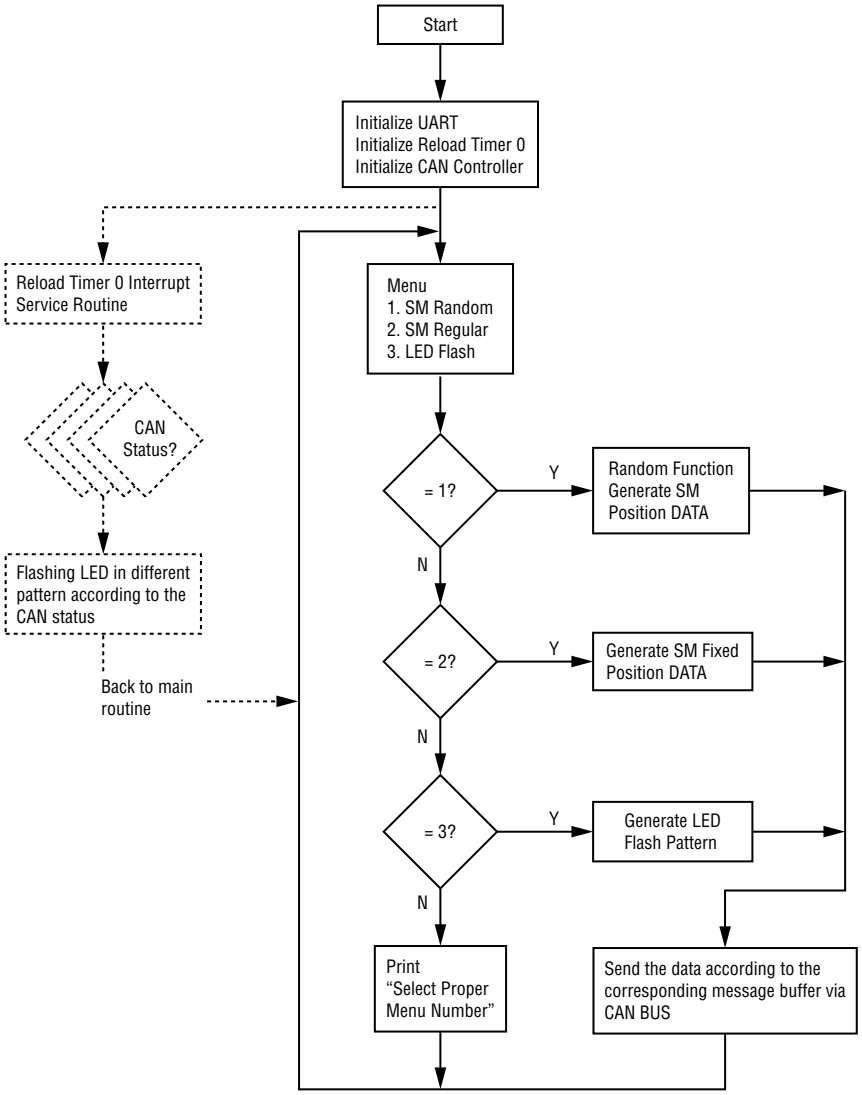


Figure 2. Software Flow Chart for CAN_NODE_1

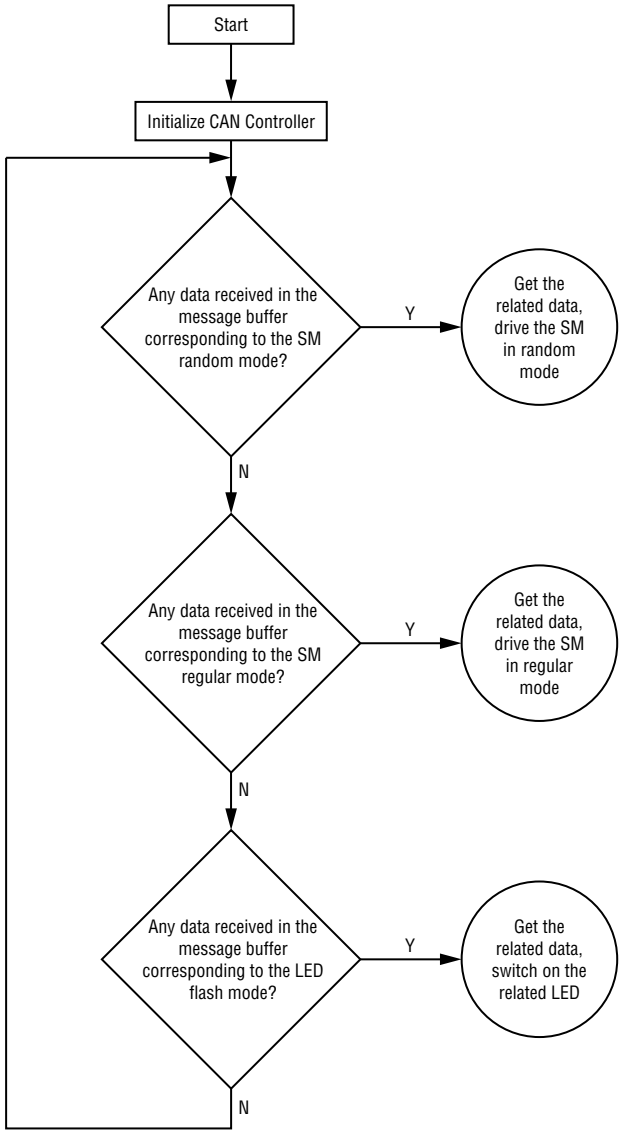


Figure 3. Software Flow Chart for CAN_NODE_2

Printed in U.S.A. EC-AN-20838-02/2000

FUJITSU MICROELECTRONICS AMERICA, INC.

Corporate Headquarters
1250 East Arques Avenue, Sunnyvale, California 94088-3470
Tel: (800) 866-8608 Fax: (408) 737-5999
E-mail: inquiry@fma.fujitsu.com Internet: http://www.fma.fujitsu.com