CAN BOOSTERPACK USER'S GUIDE

MAY 31, 2018

EDDEFLEX
www.eddeflex.com

Table of Contents

1 Introduction

CAN boosterpack by EddeFlex is designed to extend the abilities of TIVA C board from TI.
The boosterpack utilizes the native CAN controller of TM4C123x as CAN0 module and use an external stand-alone CAN controller MCP2515 as CAN1 module.
SSI2 module of TM4C122x is utilized to communicate with MCP2515 and to have full control on the newly added peripheral.

EddeFlex recommends using of ROM functions provided by TI as MCAL layer for the TM4C123x.
Also based on APIs provided by TI, EddeFlex provides a driver for MCP2515 CAN controller, with full access on its functionalities.

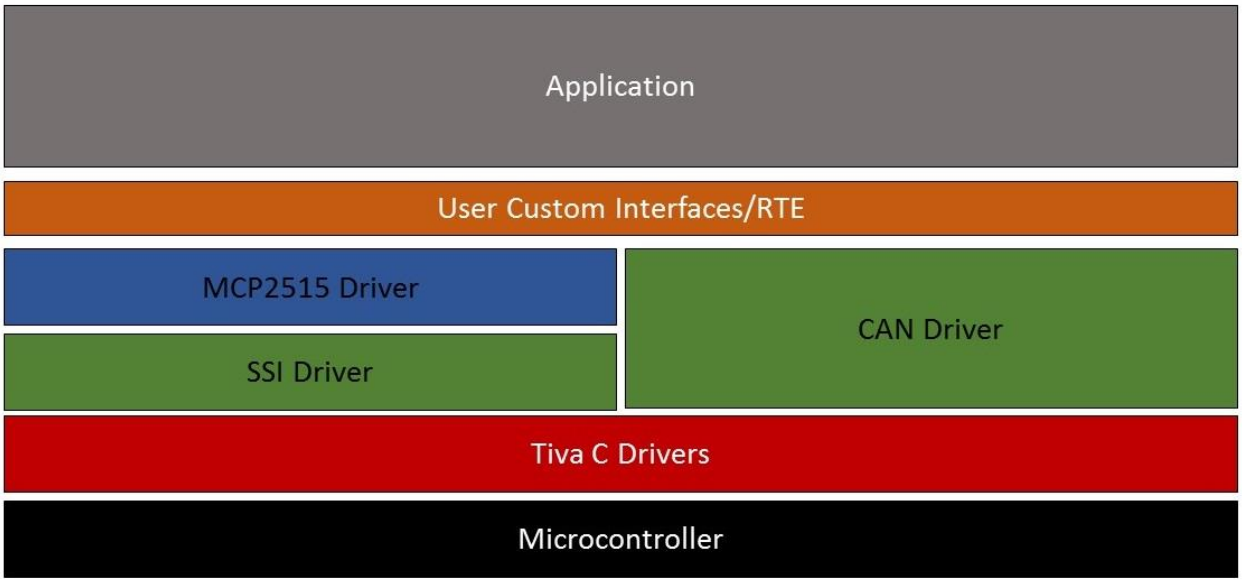
EddeFlex also provide an example project for the CAN boosterpack.
Connect CAN boosterpack to TIVA C and connect CAN0H to CAN1H and CAN0L to CAN1L then upload the example. Open the PC serial monitor and you will be able to see the messages on the bus and the controller received it (received by CAN0 or CAN1).
For more details you can check CAN boosterpack tutorials.

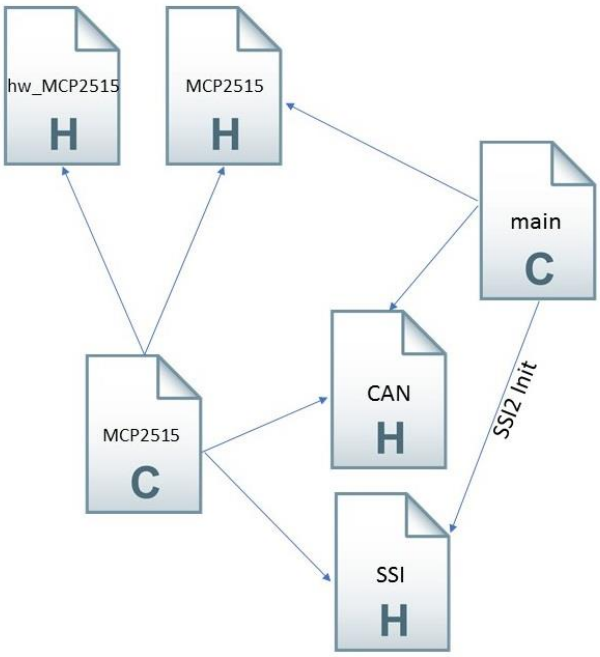
As mentioned before EddeFlex recommends the use of TI TM4C123x drivers, Below the link of the published document by TI describe the APIs for each peripheral.
http://www.ti.com/lit/ug/spmu367/spmu367.pdf

This document describes the set of APIs provided by EddeFlex for MCP2515.

2 MCP2515 Driver

2.1 Software Architecture





To use CAN boosterpack MCP2515 driver, CAN.h and MCP2515.h must be included. Please note that CAN.h must be included before MCP2515.h as the following example:

```
#include "std_types.h"
#include "hw_memmap.h"
#include "sysctl.h"
#include "rom.h"
#include "pin_map.h"
#include "EddeOS.h"
#include "GPIO.h"
#include "UART.h"
#include "SSI.h"
#include "CAN.h"
#include "MCP2515.h"
```

2.2 Datatypes

2.2.1 strCANMsgObject

<table>
<tr><th>Item</th><th colspan="3">Description</th></tr>
<tr><td>Name</td><td colspan="3">strCANMsgObject</td></tr>
<tr><td>Type</td><td colspan="3">Struct</td></tr>
<tr><td>Used by</td><td colspan="3">• MCP2515_CANMessageSet
• MCP2515_CANMessageGet</td></tr>
<tr><td rowspan="6">Elements</td><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>ui32MsgID</td><td>uint32</td><td>The CAN message identifier used for 11 or 29-bit identifiers.</td></tr>
<tr><td>ui32MsgIDMask</td><td>uint32</td><td>For RX messages, the messages identifier mask used when identifier filtering is enabled.</td></tr>
<tr><td>ui32Flags</td><td>uint32</td><td>This value holds various status flags and settings specified by tCANObjFlags, the only elements which can be used with MCP2515 are:
• MSG_OBJ_EXTENDED_ID
• MSG_OBJ_USE_ID_FILTER
• MSG_OBJ_REMOTE_FRAME
• MSG_OBJ_NO_FLAGS</td></tr>
<tr><td>ui32MsgLen</td><td>uint32</td><td>This value is the number of bytes of data in the message object.</td></tr>
<tr><td>pui8MsgData</td><td>uint8</td><td>This is a pointer to the message object's data.</td></tr>
</table>

2.3 Functions

2.3.1 MCP2515_SW_Reset

Item	Description		
Name	MCP2515_SW_Reset		
Functionality	Preform soft reset to MCP2515		
Inputs	Input Name	Type	Functionality
	u8CSPin	uint8	Determine which chip select going to use with MCP2515 chip. • GPIO_PIN_3 for CS12 • GPIO_PIN_4 for CS13 • #define MCP2515_CS12 GPIO_PIN_3 • #define MCP2515_CS13 GPIO_PIN_4 If two CAN boosterpack are connected, The two external CAN controllers can be used with the same provided APIs but with different u8CSPin input.
Outputs	None		
Return	None		

2.3.2 MCP2515_Init

Item	Description			
Name	MCP2515_Init			
Functionality	Initialize MCP2515 controller and configure CAN bit rate			
Inputs	Input Name	Type	Functionality	
	u8CSPin	uint8	Determine which chip select going to use with MCP2515 chip. • GPIO_PIN_3 for CS12 • GPIO_PIN_4 for CS13 • #define MCP2515_CS12 GPIO_PIN_3 • #define MCP2515_CS13 GPIO_PIN_4 If two CAN boosterpack are connected, the two external CAN controllers can be used with the same provided APIs but with different u8CSPin input.	
	u8BitRate	uint8	Commanded bit rate, It can be one of the following bit rates	
			Bit rate	MACRO
			1000k	MCP2515_BitRate_1000kBPS
			500k	MCP2515_BitRate_500kBPS
			250K	MCP2515_BitRate_250kBPS
			200K	MCP2515_BitRate_200kBPS
			125K	MCP2515_BitRate_125kBPS
			100K	MCP2515_BitRate_100kBPS
			80K	MCP2515_BitRate_80kBPS
			50K	MCP2515_BitRate_50kBPS
			40K	MCP2515_BitRate_40kBPS
			31.25K	MCP2515_BitRate_31k25BPS
			20K	MCP2515_BitRate_20kBPS

			10K	MCP2515_BitRate_10kBPS
			5K	MCP2515_BitRate_5kBPS
Outputs	None			
Return	None			

2.3.3 MCP2515_Enable

Item	Description		
Name	MCP2515_Enable		
Functionality	Enable MCP2515 by select normal operation mode		
Inputs	Input Name	Type	Functionality
	u8CSPin	uint8	Determine which chip select going to use with MCP2515 chip. • GPIO_PIN_3 for CS12 • GPIO_PIN_4 for CS13 • #define MCP2515_CS12 GPIO_PIN_3 • #define MCP2515_CS13 GPIO_PIN_4 If two CAN boosterpack are connected, the two external CAN controllers can be used with the same provided APIs but with different u8CSPin input.
Outputs	None		
Return	None		

2.3.4 MCP2515_Disable

Item	Description		
Name	MCP2515_Disable		
Functionality	Disable MCP2515 by select sleep mode		
Inputs	Input Name	Type	Functionality
	u8CSPin	uint8	Determine which chip select going to use with MCP2515 chip. • GPIO_PIN_3 for CS12 • GPIO_PIN_4 for CS13 • #define MCP2515_CS12 GPIO_PIN_3 • #define MCP2515_CS13 GPIO_PIN_4 If two CAN boosterpack are connected, the two external CAN controllers can be used with the same provided APIs but with different u8CSPin input.
Outputs	None		
Return	None		

2.3.5 MCP2515_CANMessageSet

<table>
<tr><td>Item</td><td colspan="4">Description</td></tr>
<tr><td>Name</td><td colspan="4">MCP2515_CANMessageSet</td></tr>
<tr><td>Functionality</td><td colspan="4">Set CAN message to TX buffer and request send.</td></tr>
<tr><td rowspan="5">Inputs</td><td>Input Name</td><td>Type</td><td colspan="2">Functionality</td></tr>
<tr><td>u8CSPin</td><td>uint8</td><td colspan="2">Determine which chip select going to use with MCP2515 chip.
• GPIO_PIN_3 for CS12
• GPIO_PIN_4 for CS13
• #define MCP2515_CS12 GPIO_PIN_3
• #define MCP2515_CS13 GPIO_PIN_4
If two CAN boosterpack are connected, the two external CAN controllers can be used with the same provided APIs but with different u8CSPin input.</td></tr>
<tr><td>u8BuffId</td><td>uint8</td><td colspan="2">Select the TX buffer which will be used to transmit the CAN message.
MCP2515 chip contains 3 TX buffer, the buffer priority can be configured but this functionality is not support by this APIs yet, by using the default settings of MCP2515 controller we will got the following priorities
<table>
<tr><td>priority</td><td>Buffer num</td><td>Buffer MACRO ID</td></tr>
<tr><td>Highest</td><td>2</td><td>MCP2515_TX2_ID</td></tr>
<tr><td>Intermediate</td><td>1</td><td>MCP2515_TX1_ID</td></tr>
<tr><td>Lowest</td><td>0</td><td>MCP2515_TX0_ID</td></tr>
</table></td></tr>
<tr><td>psMsgObject</td><td>strCANMsgObject</td><td colspan="2">Please see strCANMsgObject table</td></tr>
<tr><td>Outputs</td><td colspan="4">None</td></tr>
<tr><td>Return</td><td colspan="4">None</td></tr>
</table>

2.3.6 MCP2515_CANMessageGet

<table>
<tr><td>Item</td><td colspan="3">Description</td></tr>
<tr><td>Name</td><td colspan="3">MCP2515_CANMessageGet</td></tr>
<tr><td>Functionality</td><td colspan="3">Get CAN message from RX buffer.</td></tr>
<tr><td rowspan="2">Inputs</td><td>Input Name</td><td>Type</td><td>Functionality</td></tr>
<tr><td>u8CSPin</td><td>uint8</td><td>Determine which chip select going to use with MCP2515 chip.
• GPIO_PIN_3 for CS12
• GPIO_PIN_4 for CS13
• #define MCP2515_CS12 GPIO_PIN_3
• #define MCP2515_CS13 GPIO_PIN_4
If two CAN boosterpack are connected, the two external CAN controllers can be used with the same provided APIs but with different u8CSPin input.</td></tr>
</table>

<table>
<tr><td></td><td>u8BuffId</td><td>uint8</td><td colspan="3">Select the TX buffer which will be used to transmit the CAN message.
MCP2515 chip contains 3 TX buffer, the buffer priority can be configured but this functionality is not support by this APIs yet, by using the default settings of MCP2515 controller we will got the following priorities
<table><tr><td>priority</td><td>Buffer num</td><td>Buffer MACRO ID</td></tr><tr><td>Highest</td><td>2</td><td>MCP2515_TX2_ID</td></tr><tr><td>Intermediate</td><td>1</td><td>MCP2515_TX1_ID</td></tr><tr><td>Lowest</td><td>0</td><td>MCP2515_TX0_ID</td></tr></table></td></tr>
<tr><td rowspan="2">Outputs</td><td>Input Name</td><td>Type</td><td colspan="3">Functionality</td></tr>
<tr><td>psMsgObject</td><td>strCANMsgObject</td><td colspan="3">Please see strCANMsgObject table</td></tr>
<tr><td>Return</td><td colspan="5">None</td></tr>
</table>

2.3.7 MCP2515_CANStatusGet

<table>
<tr><td>Item</td><td colspan="5">Description</td></tr>
<tr><td>Name</td><td colspan="5">MCP2515_CANStatusGet</td></tr>
<tr><td>Functionality</td><td colspan="5">Get status of MCP2515 using special SPI status command</td></tr>
<tr><td rowspan="2">Inputs</td><td>Input Name</td><td>Type</td><td colspan="3">Functionality</td></tr>
<tr><td>u8CSPin</td><td>uint8</td><td colspan="3">Determine which chip select going to use with MCP2515 chip.
• GPIO_PIN_3 for CS12
• GPIO_PIN_4 for CS13
• #define MCP2515_CS12 GPIO_PIN_3
• #define MCP2515_CS13 GPIO_PIN_4
If two CAN boosterpack are connected, the two external CAN controllers can be used with the same provided APIs but with different u8CSPin input.</td></tr>
<tr><td>Outputs</td><td colspan="5">None</td></tr>
<tr><td rowspan="4">Return</td><td>Type</td><td colspan="4">uint8</td></tr>
<tr><td rowspan="3">Discerption</td><td>Value</td><td>MACRO</td><td colspan="2">Meaning</td></tr>
<tr><td>0x01</td><td>MCP2515_Stat_RX0Buff</td><td colspan="2">Receive Buffer 0 Full Interrupt Flag bit in CANINTF register
• 1 = Interrupt is pending (must be cleared by MCU to reset the interrupt condition)
• 0 = No interrupt is pending</td></tr>
<tr><td>0x02</td><td>MCP2515_Stat_RX1Buff</td><td colspan="2">Receive Buffer 1 Full Interrupt Flag bit in CANINTF register
• 1 = Interrupt is pending (must be cleared by MCU</td></tr>
</table>

				to reset the interrupt condition) • 0 = No interrupt is pending
		0x04	MCP2515_Stat_TX0Req	Message Transmit Request bit for Buffer 0 in TXB0CTRL register • 1 = Buffer is currently pending transmission • 0 = Buffer is not currently pending transmission
		0x08	MCP2515_Stat_TX0Buff	Transmit Buffer 0 Empty Interrupt Flag bit in CANINTF register • 1 = Interrupt is pending (must be cleared by MCU to reset the interrupt condition) • 0 = No interrupt is pending
		0x10	MCP2515_Stat_TX1Req	Message Transmit Request bit for Buffer 1 in TXB1CTRL register • 1 = Buffer is currently pending transmission • 0 = Buffer is not currently pending transmission
		0x20	MCP2515_Stat_TX1Buff	Transmit Buffer 1 Empty Interrupt Flag bit in CANINTF register • 1 = Interrupt is pending (must be cleared by MCU to reset the interrupt condition) • 0 = No interrupt is pending
		0x40	MCP2515_Stat_TX2Req	Message Transmit Request bit for Buffer 2 in TXB2CTRL register • 1 = Buffer is currently pending transmission • 0 = Buffer is not currently pending transmission
		0x80	MCP2515_Stat_TX2Buff	Transmit Buffer 2 Empty Interrupt Flag bit in CANINTF register • 1 = Interrupt is pending (must be cleared by MCU to reset the interrupt condition) • 0 = No interrupt is pending
Note	To clear pending interrupts MCP2515_CANIntClear can be used			

2.3.8 MCP2515_CANIntClear

Item	Description			
Name	MCP2515_CANIntClear			
Functionality	Clear pending interrupts in CANINTF register			
Inputs	Input Name	Type	Functionality	
	u8CSPin	uint8	Determine which chip select going to use with MCP2515 chip. • GPIO_PIN_3 for CS12 • GPIO_PIN_4 for CS13 • #define MCP2515_CS12 GPIO_PIN_3 • #define MCP2515_CS13 GPIO_PIN_4 If two CAN boosterpack are connected, the two external CAN controllers can be used with the same provided APIs but with different u8CSPin input.	
	u8IntFlags	uint8	MACRO	Meaning
			MCP2515_Int_RX0Buff	Receive Buffer 0 Full Interrupt Flag bit
			MCP2515_Int_RX1Buff	Receive Buffer 1 Full Interrupt Flag bit
			MCP2515_Int_TX0Buff	Transmit Buffer 0 Empty Interrupt Flag bit
			MCP2515_Int_TX1Buff	Transmit Buffer 1 Empty Interrupt Flag bit
			MCP2515_Int_TX2Buff	Transmit Buffer 2 Empty Interrupt Flag bit
			MCP2515_Int_BusError	Error Interrupt Flag bit
			MCP2515_Int_WakeUp	Wake-up Interrupt Flag bit
			MCP2515_Int_MsgError	Message Error Interrupt Flag bit
Outputs	None			
Return	None			

2.3.9 MCP2515_CANIntEnable

Item	Description			
Name	MCP2515_CANIntEnable			
Functionality	Enable interrupts in CANINTF register which enables interrupts physical output on $\overline{INT}$ pin			
Inputs	Input Name	Type	Functionality	
	u8CSPin	uint8	Determine which chip select going to use with MCP2515 chip. • GPIO_PIN_3 for CS12 • GPIO_PIN_4 for CS13 • #define MCP2515_CS12 GPIO_PIN_3 • #define MCP2515_CS13 GPIO_PIN_4 If two CAN boosterpack are connected, the two external CAN controllers can be used with the same provided APIs but with different u8CSPin input.	
	u8IntFlags	uint8	MACRO	Meaning
			MCP2515_Int_RX0Buff	Receive Buffer 0 Full Interrupt Enable bit

			MCP2515_Int_RX1Buff	Receive Buffer 1 Full Interrupt Enable bit
			MCP2515_Int_TX0Buff	Transmit Buffer 0 Empty Interrupt Enable bit
			MCP2515_Int_TX1Buff	Transmit Buffer 1 Empty Interrupt Enable bit
			MCP2515_Int_TX2Buff	Transmit Buffer 2 Empty Interrupt Enable bit
			MCP2515_Int_BusError	Error Interrupt Enable bit
			MCP2515_Int_WakeUp	Wake-up Interrupt Enable bit
			MCP2515_Int_MsgError	Message Error Interrupt Enable bit
Outputs	None			
Return	None			

2.3.10 MCP2515_CANIntDisable

Item	Description			
Name	MCP2515_CANIntDisable			
Functionality	Disable interrupts in CANINTF register which disables interrupts physical output on $\overline{INT}$ pin			
Inputs	Input Name	Type	Functionality	
	u8CSPin	uint8	Determine which chip select going to use with MCP2515 chip. • GPIO_PIN_3 for CS12 • GPIO_PIN_4 for CS13 • #define MCP2515_CS12 GPIO_PIN_3 • #define MCP2515_CS13 GPIO_PIN_4 If two CAN boosterpack are connected, the two external CAN controllers can be used with the same provided APIs but with different u8CSPin input.	
	u8IntFlags	uint8	MACRO	Meaning
			MCP2515_Int_RX0Buff	Receive Buffer 0 Full Interrupt Enable bit
			MCP2515_Int_RX1Buff	Receive Buffer 1 Full Interrupt Enable bit
			MCP2515_Int_TX0Buff	Transmit Buffer 0 Empty Interrupt Enable bit
			MCP2515_Int_TX1Buff	Transmit Buffer 1 Empty Interrupt Enable bit
			MCP2515_Int_TX2Buff	Transmit Buffer 2 Empty Interrupt Enable bit
			MCP2515_Int_BusError	Error Interrupt Enable bit
			MCP2515_Int_WakeUp	Wake-up Interrupt Enable bit
			MCP2515_Int_MsgError	Message Error Interrupt Enable bit
Outputs	None			
Return	None			

2.3.11 MCP2515_CANIntStatus

<table>
<tr><td>Item</td><td colspan="4">Description</td></tr>
<tr><td>Name</td><td colspan="4">MCP2515_CANIntStatus</td></tr>
<tr><td>Functionality</td><td colspan="4">Get value of CANINTF register</td></tr>
<tr><td rowspan="2">Inputs</td><td>Input Name</td><td>Type</td><td colspan="2">Functionality</td></tr>
<tr><td>u8CSPin</td><td>uint8</td><td colspan="2">Determine which chip select going to use with MCP2515 chip.
• GPIO_PIN_3 for CS12
• GPIO_PIN_4 for CS13
• #define MCP2515_CS12 GPIO_PIN_3
• #define MCP2515_CS13 GPIO_PIN_4
If two CAN boosterpack are connected, the two external CAN controllers can be used with the same provided APIs but with different u8CSPin input.</td></tr>
<tr><td>Outputs</td><td colspan="4">None</td></tr>
<tr><td rowspan="6">Return</td><td>Type</td><td colspan="3">uint8</td></tr>
<tr><td rowspan="5">Discerption</td><td>Value</td><td>MACRO</td><td>Meaning</td></tr>
<tr><td>0x01</td><td>MCP2515_Int_RX0Buff</td><td>Receive Buffer 0 Full Interrupt Flag
• 1 = Interrupt is pending (must be cleared by MCU to reset the interrupt condition)
• 0 = No interrupt is pending</td></tr>
<tr><td>0x02</td><td>MCP2515_Int_RX1Buff</td><td>Receive Buffer 1 Full Interrupt Flag
• 1 = Interrupt is pending (must be cleared by MCU to reset the interrupt condition)
• 0 = No interrupt is pending</td></tr>
<tr><td>0x04</td><td>MCP2515_Int_TX0Buff</td><td>Transmit Buffer 0 Empty Interrupt Flag
• 1 = Interrupt is pending (must be cleared by MCU to reset the interrupt condition)
• 0 = No interrupt is pending</td></tr>
<tr><td>0x08</td><td>MCP2515_Int_TX1Buff</td><td>Transmit Buffer 1 Empty Interrupt Flag
• 1 = Interrupt is pending (must be cleared by MCU to reset the interrupt condition)
• 0 = No interrupt is pending</td></tr>
</table>

		0x10	MCP2515_Int_TX2Buff	Transmit Buffer 2 Empty Interrupt Flag • 1 = Interrupt is pending (must be cleared by MCU to reset the interrupt condition) • 0 = No interrupt is pending
		0x20	MCP2515_Int_BusError	Error Interrupt Flag • 1 = Interrupt is pending (must be cleared by MCU to reset the interrupt condition) • 0 = No interrupt is pending
		0x40	MCP2515_Int_WakeUp	Wake-up Interrupt Flag • 1 = Interrupt is pending (must be cleared by MCU to reset the interrupt condition) • 0 = No interrupt is pending
		0x80	MCP2515_Int_MsgError	Message Error Interrupt Flag • 1 = Interrupt is pending (must be cleared by MCU to reset the interrupt condition) • 0 = No interrupt is pending
Note	To clear pending interrupts MCP2515_CANIntClear can be used			

