AN205447

F²MC-8FX Family MB95200 Series 8-Bit Microcontroller LIN/UART API

Associated Part Family: MB95200 Series

This document introduces LIN/UART API.

Contents

1 Introduction....1
2 LIN-UART Library Function List....2
3 LIN-UART Function Detail....3
3.1 LIN_UART_Restart Function....3
3.2 LIN_UART_Mode Function....4
3.3 LIN_UART_Init Function....4
3.4 LIN_UART_Asyn_Mode0_Init Function....6
3.5 LIN_UART_Asyn_Mode1_Init Function....6
3.6 LIN_UART_LIN_Init Function....7
3.7 LIN_UART_Syn_Init Function....8
3.8 LIN_UART_Transmit_Control_Bit Function....9
3.9 LIN_UART_Receive_Control_Bit Function....10
3.10 LIN_UART_Error_Handle Function....10
3.11 Serial_IO_Direct_Access Function....11
3.12 UART_Sendbyte Function....12
3.13 UART_Readbyte Function....13
3.14 LIN_UART_Write_Data Function....13
3.15 LIN_UART_Read_Byte Function....13
3.16 LIN_UART_Baudrate Function....14
4 Usage Demo....15
4.1 LIN UART Restart....15
4.2 LIN UART Operates in Mode 0 (Asynchronous Normal Mode)....15
4.3 LIN UART Operates in Mode 1 (Asynchronous Multiprocessor Mode)....16
4.4 LIN UART Operates in Mode 2 (Synchronous Mode)....16
4.5 LIN UART Operates in Mode 3 (LIN Mode)....17
4.6 LIN UART Receive Data....17
4.7 LIN UART Transmit Data....18
4.8 LIN UART Error Handle....18
4.9 LIN UART Serial Access....18
5 Additional Information....19
Document History....20

1 Introduction

This document introduces LIN/UART API.

We should consider some conditions, such as ‘reset’, ‘initial’, ‘mode set’, ‘transmit’, ‘receive’, ‘error handle’, ‘direction access’ and ‘baud rate set’ .The following chapters describe them in details.

2 LIN-UART Library Function List

This section introduces all functions of LIN-UART.

Table 1 lists the LIN-UART library functions.

Table 1. LIN-UART Functions

Function name	Description
void LIN_UART_Restart (LIN_UART_Reload, LIN_UART_Clear)	Restart the reload count and LIN UART software reset
void LIN_UART_Mode (Mode)	Select LIN UART work mode
void LIN_UART_Init (Bit_Length, Stop_Bit, Direction, Ex_Clk_Select, Ex_Clk_Input)	Initialize LIN_UART
void LIN_UART_Asyn_Mode0_Init (Parity_Status, Parity_Select)	Setting the parity enable and select even or odd parity
void LIN_UART_Asyn_Mode1_Init (Data_Type)	Initialize LIN_UART asynchronous mode 1(multiprocessor mode)
void LIN_UART_LIN_Init (Break_Bit, Break_Gerention)	Initialize LIN_UART LIN mode
void LIN_UART_Syn_Init (Sampling_Clock, Continue_Clk, Ss_Status, Serial_Mode, Clk_Delay)	Initialize LIN_UART synchronous mode
void LIN_UART_Transmit_Control_Bit (Status)	Enable or disable transmit data
void LIN_UART_Receive_Control_Bit (Status)	Enable or disable receive data
uchar LIN_UART_Error_Handle (Status)	Description error type and clear error flag.
void Serial_IO_Direct_Access(Serial_Access_Status, Serial_Direct_Access)	Serial I/O pin direct access
void UART_Sendbyte (Data)	Transmit a byte
uchar UART_Readbyte (void)	Receive a byte
void LIN_UART_Write_Data (Length, *Data)	Transmit the length of data
void LIN_UART_Read_Data (Length, *Data)	Receive the length of data
void LIN_UART_Baudrate (Sys_Clock, Baudrate)	Set communication baud rate

3 LIN-UART Function Detail

This section introduces the details of LIN-UART every function.

3.1 LIN_UART_Restart Function

Table 2 describes LIN_UART_Restart function.

Table 2. LIN_UART_Restart Function

Function name	LIN_UART_Restart
Function prototype	void LIN_UART_Restart (uchar LIN_UART_Reload,uchar LIN_UART_Clear)
Behavior description	reloads count restart and reset the LIN UART
Input parameter1	LIN_UART_Reload, Reload count restart
Input parameter2	LIN_UART_Clear, Reset the LIN UART
Return value	None

The function reloads count restart and reset the LIN UART. The parameter reloads count restart bit and programmable clear bit (LIN-UART software reset).

Table 3 describes the LIN_UART_Reload parameter values.

Table 3. LIN_UART_Reload Definition

LIN_UART_Reload	Description
Dis_LIN_UART_Restart	No effect
En_LIN_UART_Restart	Reload count

Table 4 describes the LIN_UART_Clear parameter values.

Table 4. LIN_UART_Clear Definition

LIN_UART_Clear	Description
Dis_LIN_UART_Clear	No effect
En_LIN_UART_Clear	Clear bit

3.2 LIN_UART_Mode Function

Table 5 describes LIN_UART_Mode function.

Table 5. LIN_UART_Mode Function

Function name	LIN_UART_Mode
Function prototype	void LIN_UART_Mode (uchar Mode)
Behavior description	Set work mode
Input parameter	Mode
Return value	None

The function selects the LIN UART operating mode.

Table 6 describes the Mode parameter values.

Table 6. Mode Definition

Mode	Description
Mode_0	Mode 0
Mode_1	Mode 1
Mode_2	Mode 2
Mode_3	Mode 3

3.3 LIN_UART_Init Function

Table 7 describes LIN_UART_Init function.

Table 7. LIN_UART_Init Function

Function name	LIN_UART_Init
Function prototype	void LIN_UART_Init (uchar Bit_Length,uchar Stop_Bit,uchar Direction,uchar Ex_Clk_Select,uchar Ex_Clk_Input)
Behavior description	
Input parameter1	Bit_Length, Data length bit
Input parameter2	Stop_Bit, Setting stop bit length
Input parameter3	Direction, LSB first or MSB first
Input parameter4	Ex_Clk_Select, Clock source select
Input parameter5	Ex_Clk_Input, Clock input enable
Return value	None

The function set the public parameter for the LIN UART works all modes. The parameters include the communication data length bit, stop length bit, transfer direction select bit, external serial clock source select bit and one-to-one external clock input enable bit.

Table 8 describes the Bit_Length parameter values.

Table 8. Bit_Length Definition

Bit_Length	Description
Length_7_Bit	7 bit length data
Length_8_Bit	8 bit length data

Table 9 describes the Stop_Bit parameter values.

Table 9. Stop_Bit Definition

Stop_Bit	Description
Stop_1_Bit	1 bit stop bit
Stop_2_Bit	2 bit stop bit

Table 10 describes the Direction parameter values.

Table 10. Direction Definition

Direction	Description
LSB	LSB first
MSB	MSB first

Table 11 describes the Ex_Clk_Select parameter values.

Table 11. Ex_Clk_Select Definition

Ex_Clk_Select	Description
Internal_Clock	Select baud rate generator
External_Clock	Select external serial source clock

Table 12 describes the Ex_Clk_Input parameter values.

Table 12. Ex_Clk_Input Definition

Ex_Clk_Input	Description
Internal_Clk_Input	External clock input by the baud rate
External_Clk_Input	External clock input by external clock directly

Notes:

In all mode, when “SMR_EXT = 0”, OTO bit is fixed at “0”.

In mode 2, when “ESCR_MS = 1”, then “SMR_SCKE = 0, EXT = 1, OTO = 1”

3.4 LIN_UART_Asyn_Mode0_Init Function

Table 13 describes LIN_UART_Asyn_Mode0_Init function.

Table 13. LIN_UART_Asyn_Mode0_Init Function

Function name	LIN_UART_Asyn_Mode0_Init
Function prototype	void LIN_UART_Asyn_Mode0_Init (uchar Parity_Status,uchar Parity_Select)
Behavior description	set parity status bit and select even parity or odd parity for LIN UART
Input parameter1	Parity_Status, Parity enable bit
Input parameter2	Parity_Select, Parity select bit
Return value	None

The function sets parity status bit and selects even parity or odd parity for LIN UART.

Table 14 describes the Parity_Status parameter values.

Table 14. Parity_Status Definition

Parity_Status	Description
Parity_Disable	Disable parity
Parity_Enable	Enable parity

Table 15 describes the Parity_Select parameter values.

Table 15. Parity_Select Definition

Parity_Select	Description
Parity_Even	Even parity
Parity_Odd	Odd parity

3.5 LIN_UART_Asyn_Mode1_Init Function

Table 16 describes LIN_UART_Asyn_Mode1_Init function.

Table 16. LIN_UART_Asyn_Mode1_Init Function

Function name	LIN_UART_Asyn_Mode1_Init
Function prototype	void LIN_UART_Asyn_Mode1_Init (uchar Data_Type)
Behavior description	set transfer data type in the asynchronous multiprocessor communication mode
Input parameter	Data_Type, Setting transmit address or data
Return value	None

The function set transfer data type in the asynchronous multiprocessor communication mode.

Table 17 describes the Data_Type parameter values.

Table 17. Data_Type Definition

Data_Type	Description
Data_Format	Data frame
Address_Format	Address frame

3.6 LIN_UART_LIN_Init Function

Table 18 describes LIN_UART_LIN_Init function.

Table 18. LIN_UART_LIN_Init Function

Function name	LIN_UART_LIN_Init
Function prototype	void LIN_UART_LIN_Init (uchar Break_bit,uchar Break_Gerention)
Behavior description	set parameter about the LIN mode
Input parameter1	Break_bit, Synch break length bit
Input parameter2	Break_Gerention, Lin synch break generation bit
Return value	None

The function set parameter about the LIN mode. The parameter have the synch break length selection bit, LIN synch break generation bit.

Table 19 describes the Break_bit parameter values.

Table 19. Break_bit Definition

Break_bit	Description
Break_13_Bit	13 bit synch break
Break_14_Bit	14 bit synch break
Break_15_Bit	15 bit synch break
Break_16_Bit	16 bit synch break

Table 20 describes the Break_Gerention parameter values.

Table 20. Break_Gerention Definition

Break_Gerention	Description
Break_Close	No effect
Break_Open	LIN synch break generation

3.7 LIN_UART_Syn_Init Function

Table 21 describes LIN_UART_Syn_Init function.

Table 21. LIN_UART_Syn_Init Function

Function name	LIN_UART_Syn_Init
Function prototype	void LIN_UART_Syn_Init (uchar Sampling_Clock,uchar Continue_Clk, uchar Ss_Status,uchar Serial_Mode,uchar Clk_Delay);
Behavior description	set parameter about the LIN UART operates on the synchronous mode
Input parameter1	Sampling_Clock, Synch break rising or falling sampling
Input parameter2	Continue_Clk, Clock output enable bit
Input parameter3	Ss_Status, Start/stop enable bit
Input parameter4	Serial_Mode, Serial clock select bit
Input parameter5	Clk_Delay, Delay serial clock
Return value	None

The function sets parameter about the LIN UART operates on the synchronous mode. The parameter have sampling clock edge select bit, continue clock output enable bit, start stop bit, serial clock transmit or receive side select bit, serial clock delay enable bit.

Table 22 describes the Sampling_Clock parameter values.

Table 22. Sampling_Clock Definition

Sampling_Clock	Description
Rising_Edge	Rising sampling
Falling_Edge	Falling sampling

Table 23 describes the Continue_Clk parameter values.

Table 23. Continue_Clk Definition

Continue_Clk	Description
Clk_Out_Disable	Disable continue clock out
Clk_Out_Enable	Enable continue clock out

Table 24 describes the Ss_Status parameter values.

Table 24. Ss_Status Definition

Ss_Status	Description
Ss_Close	No start/stop bit
Ss_Open	Start/stop bit available

Table 25 describes the Serial_Mode parameter values.

Table 25. Serial_Mode Definition

Serial_Mode	Description
Transmit_Mode	Transmission side (serial clock generation)
Receive_Mode	Reception side (external serial clock reception)

Table 26 describes the Clk_Delay parameter values.

Table 26. Clk_Delay Definition

Clk_Delay	Description
Delay_Clk_Disable	Disable delay serial clock
Delay_Clk_Enable	Enable delay serial clock

3.8 LIN_UART_Transmit_Control_Bit Function

Table 27 describes LIN_UART_Transmit_Control_Bit function.

Table 27. LIN_UART_Transmit_Control_Bit Function

Function name	LIN_UART_Transmit_Control_Bit
Function prototype	void LIN_UART_Transmit_Control_Bit (uchar Status)
Behavior description	set transmit status
Input parameter	Status, Enable or disable transmit data
Return value	None

The function set transmit status. The parameter status enables or disable transmit data, SOT pin set as serial data output pin or general-purpose I/O by parameter “Status, SCK pin as general-purpose I/O in mode 0, 1, 3. SCK pin as LIN UART clock output pin or input pin by SCR_MS state in mode 2.

Table 28 describes the Status parameter values.

Table 28. Status Definition

Status	Description
Transmit_Enable	Enable transmit
Transmit_Disable	Disable transmit

3.9 LIN_UART_Receive_Control_Bit Function

Table 29 describes LIN_UART_Receive_Control_Bit function.

Table 29. LIN_UART_Receive_Control_Bit Function

Function name	LIN_UART_Receive_Control_Bit
Function prototype	void LIN_UART_Receive_Control_Bit (uchar Status)
Behavior description	set receive status
Input parameter	Status, Enable or disable receive data
Return value	None

The function set receive status. The parameter status enables or disable receive data, SOT pin set as serial data input pin or general-purpose I/O by parameter “Status, SCK pin as general-purpose I/O in mode 0, 1, 3. SCK pin as LIN UART clock output pin or input pin by SCR_MS state in mode 2.

Table 30 describes the Status parameter values.

Table 30. Status Definition

Status	Description
Receive_Enable	Enable receive
Receive_Disable	Disable receive

3.10 LIN_UART_Error_Handle Function

Table 31 describes LIN_UART_Error_Handle function.

Table 31. LIN_UART_Error_Handle Function

Function name	LIN_UART_Error_Handle
Function prototype	uchar LIN_UART_Error_Handle (uchar Mode)
Behavior description	checks error in communication
Input parameter	Mode, Check error and clear error flag bit
Return value	Refer to table 3-32 for more details

The function checks error in communication. And validation the error comes from the parity, overrun or format. It can set enable or disable the clear the error.

Table 32 describes the Mode parameter values.

Table 32. Mode Definition

Mode	Description
Check_Error	Check error
Clear_Error	Check and clear error flag bit

Table 33 describes the Return values.

Table 33. Return Value Definition

Return value	Description
0x00	No error, operate successful
0x20	Framing error
0x40	Overrun error
0x80	Parity error
0x60	Overrun and framing error
0xA0	Parity and framing error
0xC0	Parity and overrun error
0xE0	Parity, overrun and framing error

3.11 Serial_IO_Direct_Access Function

Table 34 describes Serial_IO_Direct_Access function.

Table 34. Serial_IO_Direct_Access Function

Function name	Serial_IO_Direct_Access
Function prototype	void Serial_IO_Direct_Access(uchar Serial_Access_Status,uchar Serial_Direct_Access)
Behavior description	checks error in communication
Input parameter1	Serial_Access_Status, Access enable bit
Input parameter2	Serial_Direct_Access, Direct access bit
Return value	None

The function set serial output pin direct access state and serial I/O pin direct access.

Table 35 describes the Serial_Access_Status parameter values.

Table 35. Serial_Access_Status Definition

Serial_Access_Status	Description
Dis_Serial_Access	Disable direct access
En_Serial_Access	Enable direct access

Table 36 describes the Serial_Direct_Access parameter values.

Table 36. Serial_Direct_Access Definition

Serial_Direct_Access	Description
Direct_Access_0	Fix SOT at ”0”
Direct_Access_1	Fix SOT at “1”

Notes: Interaction between SOPE (Serial_Access_Status) and SIOP (Serial_Direct_Access)

SOPE	SIOP	Write to SIOP	Read from SIOP
0	R/W	No effect(however, the write value is retained)	Return the SIN value
1	R/W	Write “0” or “1” to SOT	Return the SIN value
1	RMW	Read the SOT value, write “0” or “1”	

3.12 UART_Sendbyte Function

Table 37 describes UART_Sendbyte function.

Table 37. UART_Sendbyte Function

Function name	UART_Sendbyte
Function prototype	void UART_Sendbyte (uchar Data)
Behavior description	transmits a byte data
Input parameter	Data, Send a byte data
Return value	None

The function transmits a byte data. The parameter data means transmit data.

3.13 UART_Readbyte Function

Table 38 describes UART_Readbyte function.

Table 38. UART_Readbyte Function

Function name	UART_Readbyte
Function prototype	uchar UART_Readbyte (void)
Behavior description	reception a byte data and return this data
Input parameter	None
Return value	return reception a byte data

The function reception a byte data and return this data.

3.14 LIN_UART_Write_Data Function

Table 39 describes LIN_UART_Write_Data function.

Table 39. LIN_UART_Write_Data Function

Function name	LIN_UART_Write_Data
Function prototype	void LIN_UART_Write_Data (uchar Length,char *Data)
Behavior description	transmits length of the data
Input parameter1	Length, Communication data length, Transmit data length
Input parameter2	*Data, Transmit data, Transmit pointer type data
Return value	None

Note: The *Data is pointer type variable. The transmit data may be a byte, an array or a section character.

The function transmits length of the data. The parameter length means length of transmit data and parameter data means pointer variable of transmit data.

3.15 LIN_UART_Read_Byte Function

Table 40 describes LIN_UART_Read_Data function.

Table 40. LIN_UART_Read_Data Function

Function name	LIN_UART_Read_Data
Function prototype	void LIN_UART_Read_Data (uchar Length, char *Data)
Behavior description	read length of the data
Input parameter1	Length, Communication data length, Reception data length
Input parameter2	*Data, Transmit data, Reception pointer type data For example, define *Data global variable array or pointer to receive data.
Return value	None

Note: Input parameter *Data, define *Data global variable array or pointer to receive data.

The function read length of the data. The parameter length means length of receive data and parameter data means pointer variable of receive data.

3.16 LIN_UART_Baudrate Function

Table 41 describes LIN_UART_Baudrate function.

Table 41. LIN_UART_Baudrate Function

Function name	LIN_UART_Baudrate
Function prototype	void LIN_UART_Baudrate (long Sys_Clock,uint Baudrate)
Behavior description	set parameter to the Baud Rate Generator Register (BGR) of the LIN UART
Input parameter1	Sys_Clock, System clock, System work clock(main clock)
Input parameter2	Baudrate, Communication baud rate
Return value	None

Note:

1. System clock should select external main clock, main-internal CR clock, external sub clock and sub-internal CR clock. The all clock value bound please conference to in the chapter 3.
2. Baud rate set value. Any baud rate can calculate correct from 300 bps to 65535 bps. And then it recommend customer to input typical value like the follow table.

Table 42 Typical Baud Rate.

Table 42. Typical Baud Rate

Typical Baud Rate (bps)					
300	600	1200	2400	4800	7200
9600	10417	19200	28800	38400	57600

The function set parameter to the Baud Rate Generator Register (BGR) of the LIN UART.

4 Usage Demo

This section introduces some sample when use.

4.1 LIN UART Restart

The LIN UART restart reload count and clear flag bit when consumer use the following function.

LIN_UART_Restart ();

Sample code

```
void LIN_UART_REST (void)
{
    LIN_UART_Restart (En_LIN_UART_Restart, En_LIN_UART_Clear);
}
```

4.2 LIN UART Operates in Mode 0 (Asynchronous Normal Mode)

User may be using the following functions when LIN UART operates in the mode 0 (asynchronous normal mode).

LIN_UART_Mode ();

LIN_UART_Init ();

LIN_UART_Baudrate ();

LIN_UART_Asyn_Mode0_Init ();

LIN_UART_Receive_Control_Bit ();

LIN_UART_Transmit_Control_Bit ();

Sample code

```
void LIN_UART_Initailize (void)
{
LIN_UART_Mode (Mode_0);
LIN_UART_Init (Length_8_Bit, Stop_1_Bit, LSB, Internal_Clock, Internal_Clk_Input);
LIN_UART_Asyn_Mode0_Init (Parity_Enable, Parity_Even);
LIN_UART_Baudrate (10000000, 9600);          //MCLK = 10M, Baud = 9600bps
LIN_UART_Transmit_Control_Bit (Transmit_Enable);
LIN_UART_Receive_Control_Bit (Receive_Enable);
}
```

4.3 LIN UART Operates in Mode 1 (Asynchronous Multiprocessor Mode)

The LIN UART operates in mode 1 may be using the following functions.

LIN_UART_Mode ();

LIN_UART_Init ();

LIN_UART_Baudrate ();

LIN_UART_Asyn_Mode1_Init ();

LIN_UART_Receive_Control_Bit ();

LIN_UART_Transmit_Control_Bit ();

Sample code

```
void initialize (void)
{
LIN_UART_Mode (Mode_1);
LIN_UART_Init (Length_8_Bit, Stop_1_Bit, LSB, Internal_Clock, Internal_Clk_Input);
LIN_UART_Asyn_Mode1_Init (Data_Format);
LIN_UART_Baudrate (10000000, 9600);          //MCLK = 10M, Baud = 9600bps
LIN_UART_Transmit_Control_Bit (Transmit_Enable);
LIN_UART_Receive_Control_Bit (Receive_Enable);
}
```

4.4 LIN UART Operates in Mode 2 (Synchronous Mode)

The LIN UART operates in mode 2 may be using the following functions.

LIN_UART_Mode ();

LIN_UART_Init ();

LIN_UART_Baudrate ();

LIN_UART_Syn_Init ();

LIN_UART_Receive_Control_Bit ();

LIN_UART_Transmit_Control_Bit ();

Sample code

```
void initialize (void)
{
LIN_UART_Mode (Mode_2);
LIN_UART_Init (Length_8_Bit, Stop_1_Bit, LSB, External_Clock, External_Clk_Input);
LIN_UART_Syn_Init (Rising_Edge, Clk_Out_Enable, Ss_Open, Transmit_Mode, Delay_Clk_Disable,
Internal_Clock, Internal_Clk_Input);
LIN_UART_Baudrate (10000000, 9600);          //MCLK = 10M, Baud = 9600bps
LIN_UART_Transmit_Control_Bit (Transmit_Enable);
LIN_UART_Receive_Control_Bit (Receive_Enable);
}
```

4.5 LIN UART Operates in Mode 3 (LIN Mode)

The LIN UART operates in mode 3 may be using the following functions.

```
LIN_UART_Init ();
LIN_UART_Baudrate ();
LIN_UART_LIN_Init ();
LIN_UART_Receive_Control_Bit ();
LIN_UART_Transmit_Control_Bit ();
Sample code
void initialize (void)
{
LIN_UART_Init (Length_8_Bit, Stop_1_Bit, LSB, Internal_Clock, Internal_Clk_Input);
LIN_UART_Baudrate (10000000, 9600);
// system clock 10MHZ, baud rate 9600
LIN_UART_Asyn_LIN_Init (Break_13_Bit, Break_Open);
LIN_UART_Transmit_Control_Bit (Transmit_Enable);
LIN_UART_Receive_Control_Bit (Receive_Enable);
}
```

4.6 LIN UART Receive Data

The LIN UART receives data use the following functions.

```
LIN_UART_Receive_Control_Bit ();
LIN_UART_Read_Data ();
Sample code
void receive_data (void)
{
uchar *read_data = 0;
LIN_UART_Receive_Control_Bit (Receive_Enable);
                                        // enable receive data
LIN_UART_Read_Data (2, read_data);
                                        // read 2 bytes
LIN_UART_Receive_Control_Bit (Receive_Disable);
                                        // disable receive data
}
```

4.7 LIN UART Transmit Data

The LIN UART transmits data use the following functions.

LIN_UART_Transmit_Control_Bit ();

LIN_UART_Write_Data ();

Sample code

```
void receive_data (void)
{
uchar transmit [] = {0, 1, 2, 3, 4, 5};
LIN_UART_Transmit_Control_Bit (Transmit_Enable);
                                        // enable transmit data
LIN_UART_Write_Data (6, transmit);
                                        // transmit 6 bytes data
LIN_UART_Transmit_Control_Bit (Transmit_Disable);
                                        // disable transmit data
}
```

4.8 LIN UART Error Handle

The LIN UART checks error and reads the types of error when user uses the following functions.

LIN_UART_Error_Handle ();

Sample code

```
void receive_data (void)
{
uchar error_value = 0;
error_value = LIN_UART_Error_Handle (Check_Error);
                                        // check error and return error value, keep the flag bit
error_value = LIN_UART_Error_Handle (Clear_Error);
                                        // check error and return error value, clear the flag bit
}
```

4.9 LIN UART Serial Access

The LIN UART enables or disables serial output pin direct access and serial IO pin direct access when user uses the following function.

Serial_IO_Direct_Access ();

Sample code

```
void serial_access (void)
{
    Serial_IO_Direct_Access (En_Serial_Access, Direct_Access_1);
}
```

5 Additional Information

For more information about how to use MB95200H EV-board, BGM Adaptor and SOFTUNE, please refer to SKT MB2146-410-01-E User Manual, or visit website:

http://www.cypress.com/documentation/software-and-drivers/f2mc-8fx-mb95200h210h-series-starter-kit-mb2146-410a-01-e-setup

Document History

Document Title: AN205447 – F²MC-8FX Family MB95200 Series 8-Bit Microcontroller LIN/UART API

Document Number: 002-05447

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	HUAL	03/03/2009	Initial release
*A	5279152	HUAL	05/20/2016	Migrated Spansion Application Note MCU-AN- 500026-E-10 to Cypress format.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Lighting & Power Control	cypress.com/powerpsoc
Memory	cypress.com/memory
PSoC	cypress.com/psoc
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless/RF	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corporation. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2009-2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.
TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.
Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.