UART 接口函数

*1. int FUartPs_init(FUartPs_T *dev, u32 id, u32 addr, u32 clk);*

描述	* This function initializes a uart. It disables all interrupts and * resets the Tx and Rx FIFOs. It also initializes the driver's * internal data structures.
参数	* @param dev is uart handle. * @param id is uart id 0 or 1. * @param addr is the base address of uart. * @param clk is the clock of uart.
返回值	* @return * 0 -- if successful * -FMSH_ENOSYS -- hardware parameters for the device could not be * automatically determined

*2. int32_t FUartPs_isBusy(FUartPs_T *dev);*

描述	* This function returns whether the UART is busy (transmitting and/or * receiving) or not. If the UART busy bit is unsupported, an error * code is returned.
参数	* @param dev is uart handle.
返回值	* @return * TRUE -- UART is busy * FALSE -- UART is not busy

*3. int FUartPs_reset(FUartPs_T *dev)*

描述	* This function performs a hardware reset on a FMSH_apb_uart device.
参数	* @param dev is uart handle.
返回值	* @return * 0 -- if successful * -FMSH_ENOSYS -- function not supported

*4. void FUartPs_resetTxFifo(FUartPs_T *dev)*

描述	* This funciton resets the transmitter FIFO.
参数	* @param dev is uart handle.
返回值	* @return None.

*5. void FUartPs_resetRxFifo(FUartPs_T *dev)*

描述	* This function resets the receiver FIFO.
参数	* @param dev is uart handle.
返回值	* @return None.

*6. u8 FUartPs_setBaudRate(FUartPs_T *dev, u32 baudRate)*

描述	** This function set buad rate of uart device*
参数	** @param dev is a pointer to the uart instance.* ** @param baud_rate is the buad rate*
返回值	** @return 0 if successful, otherwise 1.*

*7. u8 FUartPs_setBaudRate(FUartPs_T *dev, u32 baudRate)*

描述	** This function set buad rate of uart device*
参数	** @param dev is a pointer to the uart instance.* ** @param baud_rate is the buad rate*
返回值	** @return 0 if successful, otherwise 1.*

*8. u32 FUartPs_getBaudRate(FUartPs_T *dev)*

描述	** This function returns buad rate of uart device*
参数	** @param dev is a pointer to the uart instance.*
返回值	** @return uart buad rate.*

*9. int32_t FUartPs_setLineControl(FUartPs_T *dev, enum FUartPs_line_control setting)*

描述	* This function is used to set the parity and the number of data and * stop bits. The FUartPs_line_control defintions are used to specify * this mode. The line control settings should not be changed when the * UART is busy.
参数	* @param * dev -- FMSH_apb_uart handle * mode -- line control settings
返回值	* @return * 0 -- if successful * -FMSH_EBUSY -- UART is busy

*10. enum FUartPs_line_control FUartPs_getLineControl(FUartPs_T *dev)*

描述	* Returns the line control settings.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * The current line control settings.

*11. void FUartPs_setDataBits(FUartPs_T *dev, enum FUartPs_cls cls)*

描述	* Sets the number of bits per character (5/6/7/8).
参数	* @param * dev -- FMSH_apb_uart handle * cls -- number of data bits per character
返回值	* @return None.

*12. enum FUartPs_cls FUartPs_getDataBits(FUartPs_T *dev)*

描述	* Returns the number of bits per character setting for data transfers.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* The current number of data bits setting.

*13. void FUartPs_setStopBits(FUartPs_T *dev, enum FUartPs_stop_bits stop)*

描述	* Sets the number of stop bits (1/1.5/2).
参数	* @param * dev -- FMSH_apb_uart handle

	* stop -- number of stop bits
返回值	* @return None.

*14. enum FUartPs_stop_bits FUartPs_getStopBits(FUartPs_T *dev)*

描述	* Returns the number of stop bits setting for data transfers.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * The current number of stop bits setting.

*15. void FUartPs_setParity(FUartPs_T *dev, enum FUartPs_parity*

parity)

描述	* Sets the parity mode (none/odd/even).
参数	* @param * dev -- FMSH_apb_uart handle * parity -- parity to set
返回值	* @return None。

*16. void FUartPs_setStick(FUartPs_T *dev, u8 stick)*

描述	
参数	
返回值	

*17. enum FUartPs_parity FUartPs_getParity(FUartPs_T *dev)*

描述	* Returns the parity setting for data transfers.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * The current parity setting.

*18. int FUartPs_enableFifos(FUartPs_T *dev)*

描述	* This function enables receive and transmit FIFOs, if they are * available.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * 0 -- if successful * -FMSH_ENOSYS -- function not supported

*19. int FUartPs_disableFifos(FUartPs_T *dev)*

描述	* This function disables receive and transmit FIFOs.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * 0 -- if successful * -FMSH_ENOSYS -- function not supported

*20. BOOL FUartPs_areFifosEnabled(FUartPs_T *dev)*

描述	* Returns whether the FIFOs and enabled or not.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * TRUE -- FIFOs are enabled * FALSE -- FIFOs are disabled or not available

*21. int FUartPs_isTxFifoFull(FUartPs_T *dev)*

描述	* Returns whether the transmitter FIFO is full or not.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * TRUE -- Tx FIFO is full * FALSE -- Tx FIFO is not full * -FMSH_ENOSYS -- function not supported

int FUartPs_*isTxFifoEmpty*(FUartPs_T *dev)

描述	* Returns whether the transmitter FIFO is empty or not.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * TRUE -- Tx FIFO is empty * FALSE -- Tx FIFO is not empty * -FMSH_ENOSYS -- function not supported

*22. int FUartPs_isRxFifoFull(FUartPs_T *dev)*

描述	* Returns whether the receiver FIFO is full or not.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * TRUE -- Rx FIFO is full * FALSE -- Rx FIFO is not full * -FMSH_ENOSYS -- function not supported

*23. int FUartPs_isRxFifoEmpty(FUartPs_T *dev)*

描述	* This function returns whether the receiver FIFO is empty or not.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * TRUE -- Rx FIFO is empty * FALSE -- Rx FIFO is not empty * -FMSH_ENOSYS -- function not supported

*24. int FUartPs_getTxFifoLevel(FUartPs_T *dev)*

描述	* This function returns the number of characters currently present in * the Tx FIFO.
参数	* @param * dev -- FMSH_apb_uart handle

返回值	* @return * -FMSH_ENOSYS -- function not supported * Otherwise number of characters currently in the Tx FIFO is returned.

*25. int FUartPs_getRxFifoLevel(FUartPs_T *dev)*

描述	* This function returns the number of characters currently present in * the Rx FIFO.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * -FMSH_ENOSYS -- function not supported * Otherwise number of characters currently in the Rx FIFO is returned.

*26. unsigned FUartPs_getFifoDepth(FUartPs_T *dev)*

描述	* Returns how many bytes deep the transmitter and receiver FIFOs are.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * FIFO depth in bytes (64)

*27. int FUartPs_enablePtime(FUartPs_T *dev)*

描述	* This function enables Programmable Threshold (THRE) Interrupt Mode * (PTIME). This mode enables triggering of interrupts for different * levels of the Tx/Rx FIFOs. Enabling PTIME also changes the

	* functionality of the lsr/thre bit (LSR[5]) to indicate that the Tx * FIFO is full. See the FMSH_apb_uart databook for more information.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * 0 -- if successful * -FMSH_ENOSYS -- function not supported

*28. int FUartPs_disablePtime(FUartPs_T *dev)*

描述	* This function disables Programmable Threshold (THRE) Interrupt Mode * (PTIME). When PTIME is disabled, the functionality of the lsr/thre * bit (LSR[5]) is normal, indicating that the Tx FIFO/THR is empty. * See the FMSH_apb_uart databook for more information.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * 0 -- if successful * -FMSH_ENOSYS -- function not supported

*29. BOOL FUartPs_isPtimeEnabled(FUartPs_T *dev)*

描述	* This function returns whether Programmable Threshold (THRE) * Interrupt Mode (PTIME) is enabled or not.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return

	* TRUE -- PTIME is enabled * FALSE -- PTIME is disabled

*30. void FUartPs_setBreak(FUartPs_T *dev, enum FMSH_state state)*

描述	* Sets the break control bit to 'state'. When enabled, it causes a * break signal to be generated, by holding the sout line low, until * the break bit is subsequently cleared (with this function).
参数	* @param * dev -- FMSH_apb_uart handle * state -- Set or Clear
返回值	* @return None.

*31. enum FMSH_state FUartPs_getBreak(FUartPs_T *dev)*

描述	* Returns the state of the break control bit.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * The current break bit state.

*32. void FUartPs_setModemLine(FUartPs_T *dev, enum FUartPs_modem_line lines)*

描述	* This function is used to Set specific modem lines. The lines * argument comprises of one or more bitwise OR'ed FUartPs_modem_line * enumerated values.
参数	* @param

	* dev -- FMSH_apb_uart handle * lines -- modem line(s) to Set
返回值	* @return None.

*33. void FUartPs_clearModemLine(FUartPs_T *dev,*

enum FUartPs_modem_line lines)

描述	* This function is used to Clear specific modem lines. The lines * argument comprises of one or more bitwise OR'ed FUartPs_modem_line * enumerated values.
参数	* @param * dev -- FMSH_apb_uart handle * lines -- modem line(s) to Clear
返回值	* @return None.

34. enum FUartPs_modem_line FUartPs_getModemLine(FUartPs_T

*dev)

描述	* This function returns the state of the modem control lines. The * FUartPs_modem_line enumerated values are used with this function's * return value to determine the current state of the modem lines.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * The current value on the modem line control settings.

*35. void FUartPs_enableLoopback(FUartPs_T *dev)*

描述	* Enables loopback mode.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * The current value on the modem line control settings.

*36. void FUartPs_disableLoopback(FUartPs_T *dev)*

描述	* Disables loopback mode.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * None.

*37. BOOL FUartPs_isLoopbackEnabled(FUartPs_T *dev)*

描述	* Returns whether loopback mode is enabled or not.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * TRUE -- loopback mode is enabled * FALSE -- loopback mode is disabled

*38. int FUartPs_enableAfc(FUartPs_T *dev)*

描述	* Enables Automatic Flow Control mode.

参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * 0 -- if successful * -FMSH_ENOSYS -- function not supported

*39. int FUartPs_disableAfc(FUartPs_T *dev)*

描述	* Disables Automatic Flow Control mode.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * 0 -- if successful * -FMSH_ENOSYS -- function not supported

*40. BOOL FUartPs_isAfcEnabled(FUartPs_T *dev)*

描述	* Returns whether Automatic Flow Control mode is enabled or not.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * TRUE -- Automatic Flow Control is enabled * FALSE -- Automatic Flow Control is disabled

41. enum FUartPs_line_status FUartPs_getLineStatus(FUartPs_T

*dev)

描述	* This function returns the current line status register value. This * value is used in conjunction with the FUartPs_line_status enumerated * values to determine the current line status. See the FMSH_apb_uart * databook for more information about the line status register.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * The current line status register value.

*42. enum FUartPs_modem_status FUartPs_getModemStatus(FUartPs_T*dev)*

描述	* This function returns the current modem status register value. This * value is used in conjunction with the FUartPs_modem_status * enumerated values to determine the current modem status. See the * FMSH_apb_uart databook for more information about the modem status * register.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * The current modem status register value.

*43. void FUartPs_setScratchpad(FUartPs_T *dev, uint8_t value)*

描述	* Sets the value of the scratchpad register. This register has no * functional use and is available to a programmer to use at their own * discretion.

参数	* @param * dev -- FMSH_apb_uart handle * byte -- value to set
返回值	* @return * None.

*44. uint8_t FUartPs_getScratchpad(FUartPs_T *dev)*

描述	* Returns the value of the scratchpad register.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * The current scratchpad register value.

*45. uint8_t FUartPs_read(FUartPs_T *dev)*

描述	* Reads a single character from the receiver FIFO and returns it. * This function does not check if there is data in the Rx FIFO * beforehand, which is a user responsibility.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * The character read from the Rx FIFO.

*46. void FUartPs_write(FUartPs_T *dev, uint8_t character)*

描述	* Writes a single character to the transmitter FIFO. This function

	* does not check if there is space in the Tx FIFO beforehand, which is * a user responsibility.
参数	* @param * dev -- FMSH_apb_uart handle * character -- character to write
返回值	* @return * None.

*47. int FUartPs_burstRead(FUartPs_T *dev, uint8_t *buffer,*

unsigned length)

描述	* This function reads characters from the Rx FIFO, using burst * transactions on the AHB bus. This function does not check if there * is enough valid data in the Rx FIFO beforehand, which is a user * responsibility. The length argument should never exceed the FIFO * depth.
参数	* @param * dev -- FMSH_apb_uart handle * buffer -- buffer to which data is stored * length -- number of characters to read
返回值	* @return * 0 -- if successful * -FMSH_ENOSYS -- function not supported

*48. int FUartPs_burstWrite(FUartPs_T *dev, uint8_t *buffer,*

unsigned length)

描述	* This function writes a characters to the Tx FIFO, using burst * transfers on the AHB bus. This function does not check if there is * sufficient spave available in the Tx FIFO beforehand, which is a * user responsibility. The length argument should never exceed the * FIFO depth.
参数	* @param * dev -- FMSH_apb_uart handle * buffer -- buffer from which to send data * length -- number of characters to write
返回值	* @return * 0 -- if successful * -FMSH_ENOSYS -- function not supported

*49. void FUartPs_enableIrq(FUartPs_T *dev, enum FUartPs_irq*

interrupts)

描述	* Enables specified interrupt(s).
参数	* @param * dev -- FMSH_apb_uart handle * interrupts -- interrupt(s) to enable
返回值	* @return * None.

*50. void FUartPs_disableIrq(FUartPs_T *dev, enum FUartPs_irq*

interrupts)

描述	* Disables specified interrupt(s).

参数	* @param * dev -- FMSH_apb_uart handle * interrupts -- interrupt(s) to disable
返回值	* @return * None.

*51. BOOL FUartPs_isIrqEnabled(FUartPs_T *dev, enum FUartPs_irq*

interrupt)

描述	* Returns whether the specified interrupt is enabled or not. Only one * interrupt may be specifed per invocation of this function.
参数	* @param * dev -- FMSH_apb_uart handle * interrupt -- interrupt to check
返回值	* @return * TRUE -- interrupt is enabled * FALSE -- interrupt is disabled

*52. uint8_t FUartPs_getIrqMask(FUartPs_T *dev)*

描述	* Returns the current interrupt mask. For each bitfield, a value of * '0' indicates that an interrupt is masked while a value of '1' * indicates that an interrupt is enabled.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * TRUE -- interrupt is enabled

	* FALSE -- interrupt is disabled

*53. enum FUartPs_event FUartPs_getActiveIrq(FUartPs_T *dev)*

描述	* Returns the event identification number of the highest priority * interrupt that is active.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return * The current highest priority active interrupt.

*54. void FUartPs_setTxTrigger(FUartPs_T *dev, enum*

FUartPs_tx_trigger trigger)

描述	* Sets the trigger level of the transmitter FIFO empty interrupt.
参数	* dev -- FMSH_apb_uart handle * trigger -- level at which to set trigger
返回值	* @return * None.

55. enum FUartPs_tx_trigger FUartPs_getTxTrigger(FUartPs_T

*dev)

描述	* Gets the trigger level of the transmitter FIFO empty interrupt.
参数	* @param * dev -- FMSH_apb_uart handle
返回值	* @return

	* transmitter trigger level

*56. void FUartPs_setRxTrigger(FUartPs_T *dev, enum*

FUartPs_rx_trigger trigger)

描述	* Sets the trigger level for the receiver FIFO full interrupt.
参数	* @param * dev -- FMSH_apb_uart handle * trigger -- level at which to set trigger
返回值	* @return * None

57. enum FUartPs_rx_trigger FUartPs_getRxTrigger(FUartPs_T

*dev)

描述	* Gets the trigger level of the receiver FIFO full interrupt.
参数	* @param * dev -- FMSH_apb_uart handle * level -- level at which to set trigger
返回值	* @return * The receiver empty trigger level.

*58. void FUartPs_setListener(FUartPs_T *dev, FUartPs_callback*

userFunction)

描述	* This function is used to set a user listener callback. The listener * function is responsible for handling all events/interrupts that are

	* not handled internally by the Driver Kit. This encompasses events * such as errors or receiving data when there is no user Rx buffer set * up. In this respect, it can be considered as a way of extending the * default interrupt handler.
参数	* @param * dev -- FMSH_apb_uart handle * listener -- user listener function
返回值	* @return * None.

*59. int FUartPs_userIrqHandler(FUartPs_T *dev)*

描述	* This function identifies the current highest priority active * interrupt, if any, and forwards it to the user-specified listener * function for processing. This allows a user absolute control over * how each UART interrupt is processed. * * None of the other Interrupt API functions can be used with this * interrupt handler. This is because they are symbiotic with the * FUartPs_irqHandler() interrupt handler. All Command and Status API * functions, however, can be used within the user listener function. * This is in contrast to FUartPs_irqHandler(), where FUartPs_read(), * FUartPs_write(), FUartPs_burstRead() and FUartPs_burstWrite() cannot * be used within the user listener function.
参数	* @param * dev -- FMSH_apb_uart handle

返回值	* @return * TRUE -- an interrupt was processed * FALSE -- no interrupt was processed * -FMSH_EIO -- unrecognized interrupt ID was read