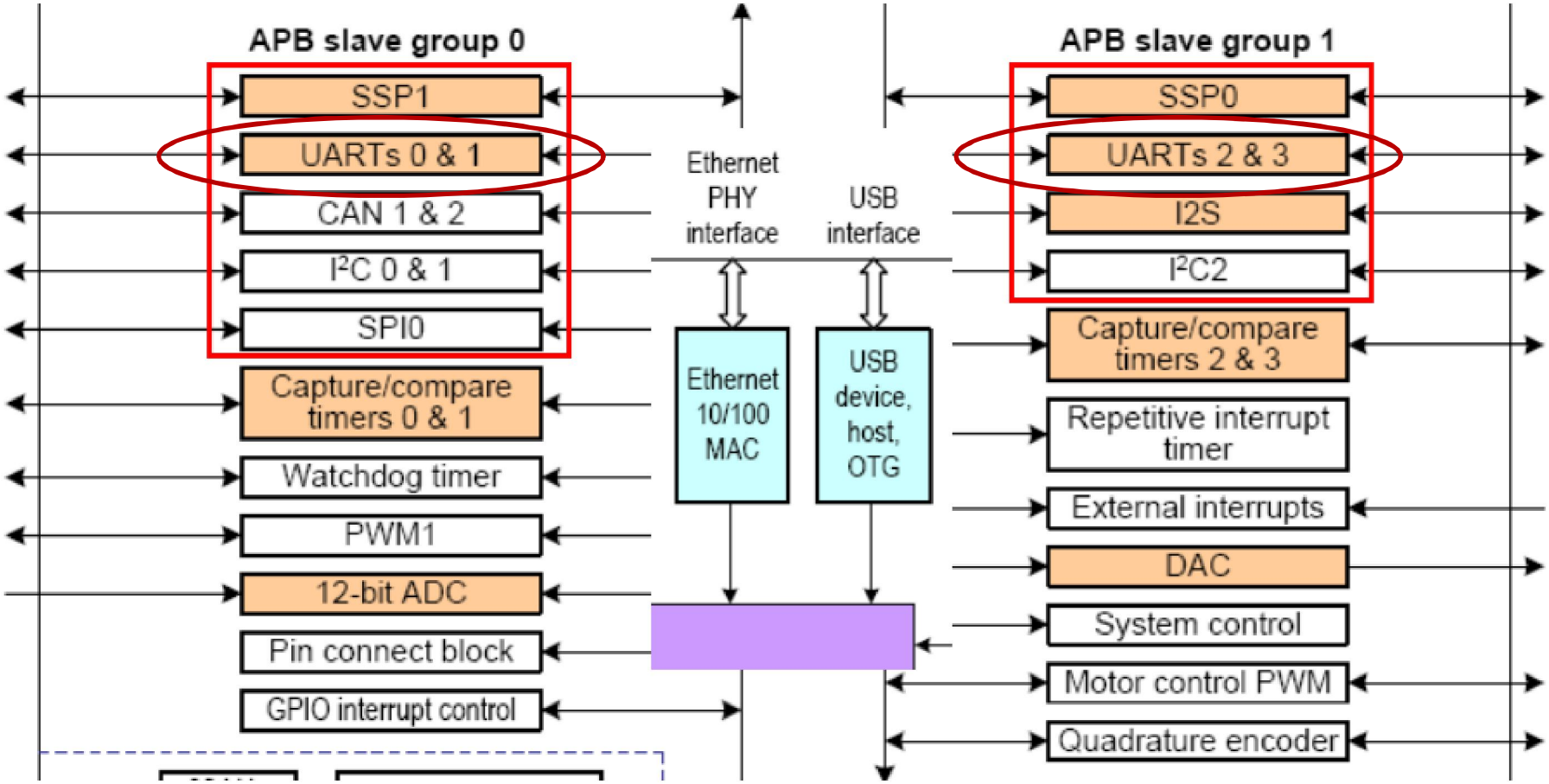
Serial Interfaces

(**UART: U**niversal **A**synchronous **R**x/**T**x)

Serial Interfaces: UARTs 0...3

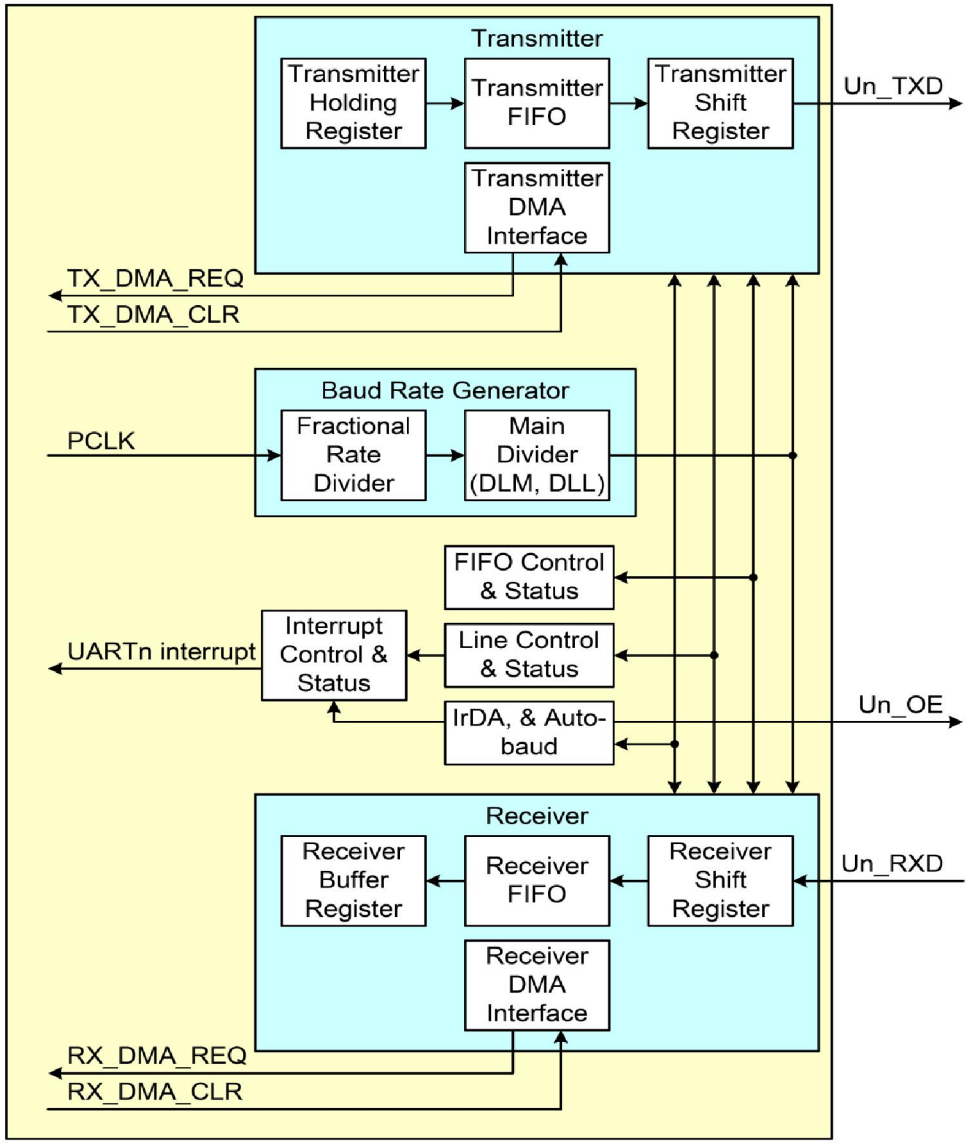


UART: Universal Asynchronous Rx/Tx

- **4 modules** (UART0, 1, 2, 3).
- 16 byte Receive and Transmit FIFOs with DMA support.
- Fractional divider for baud rate control, auto-baud capabilities, and implementation of software or hardware flow control.
- EIA-485/RS-485 and 9-bit mode support (UART1).
  - Allows both software address detection and automatic address detection using 9-bit mode.
  - Auto Direction Control.
    - Control line RTS/DTS to enable and disable the driver.
    - Address Match Register: store the station address in a RS-485 milti-drop communication.
    - Delay Value register: set the delay between the last stop-bit and the deassertion of the DIR line.
- Modem control support (UART1).
- IrDA support for infrared communication (UART3).
- Maximum possible speed of the UART ~ 6 Mbps.

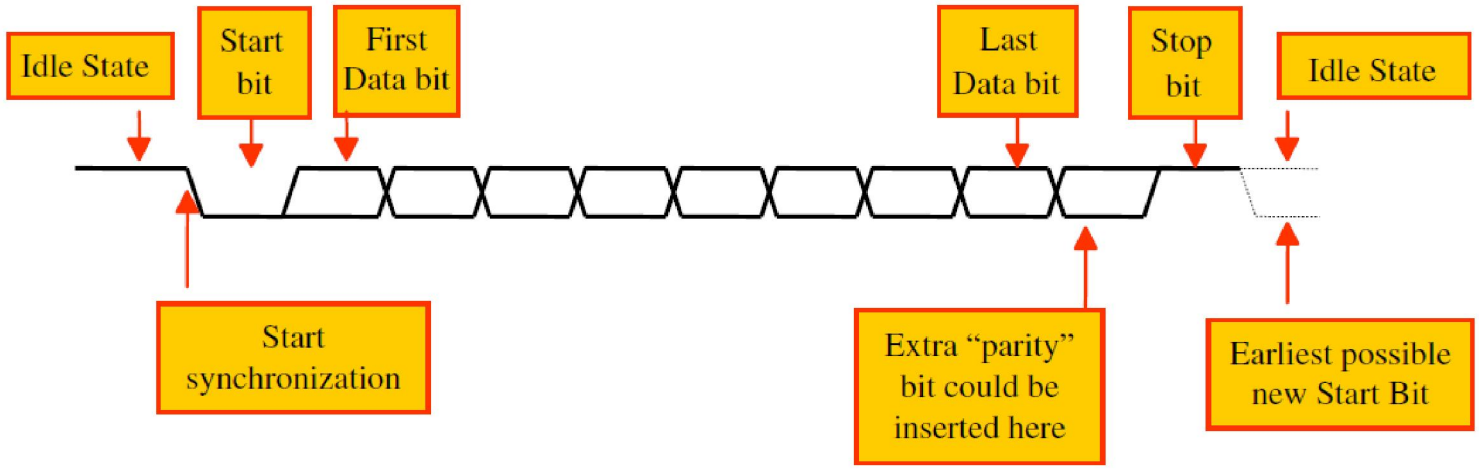
UART: Universal Asynchronous Rx / Tx

UART0,2,3

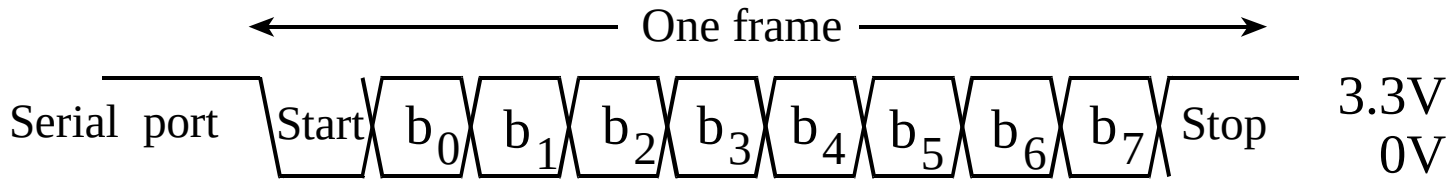


UART: TX Character Framing

- Start Bit.
- Data Bits of 5, 6, 7 or 8.
- Parity Bit.
- Stop Bit of 1, 1.5 or 2.

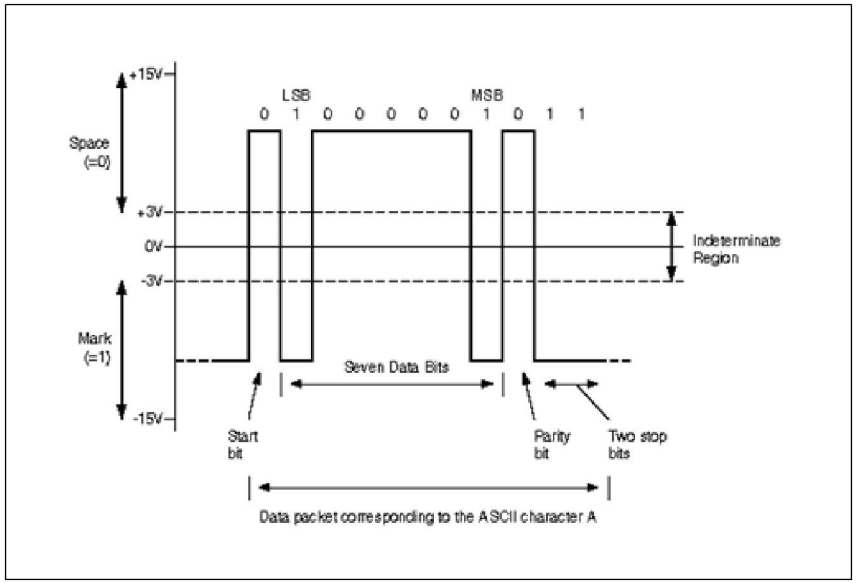


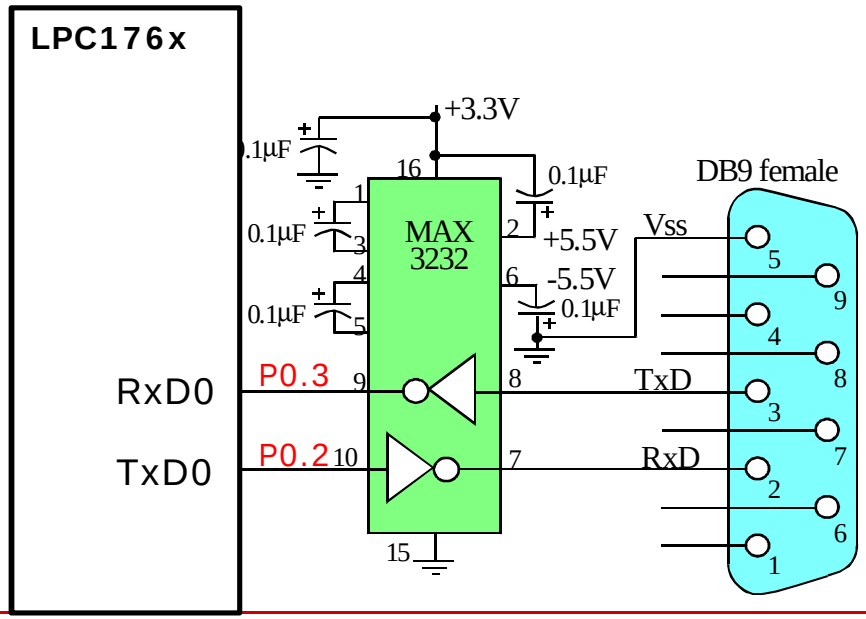
- Example: 8 bits data, No Parity, 1 stop bit



UART: RS-232

- RS-232 is a popular communications interface for connecting modems and data acquisition devices (i.e. GPS receivers, electronic balances, data loggers, ...) to computers.
- RS-232 can be plugged straight into the computer's serial port (know as COM or Comm port).
- **RS-232 Line Driver**
  - Each signal (TxD, RxD) on the interface connector with reference to a signal ground.
  - The "idle" state (MARK) has the signal level negative with respect to common whereas the active state (SPACE) has the signal level positive respect to the same reference.



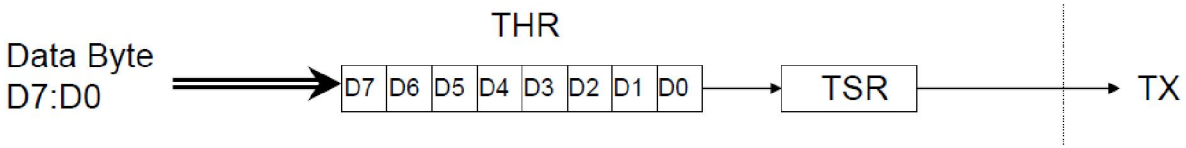


UART: Transmitter

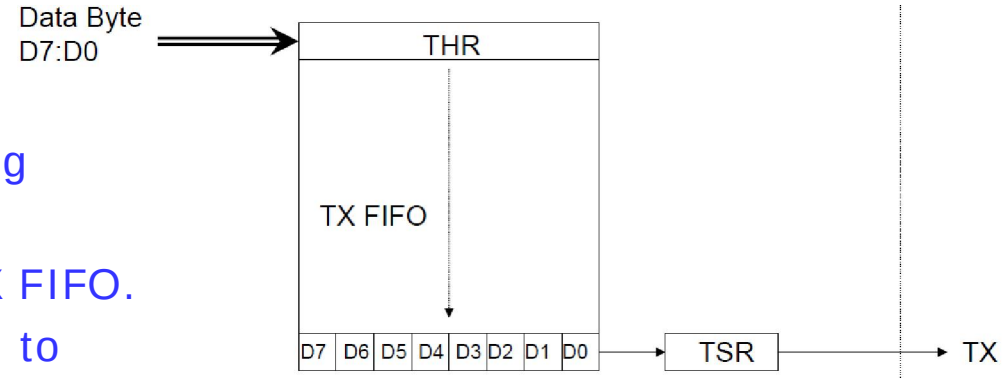
- Parallel-to-serial conversion
- Non-FIFO Mode
  - Transmit Holding Register (THR) and Transmit Shift Register (TSR)
- FIFO Mode
  - Transmit (TX) FIFO and Transmit Shift Register (TSR)
- 16x timing for bit shifting.
- Character Framing.
- Parity Insertion.
- TX FIFO interrupt and status.

UART: Transmitter

- **Non FIFO mode:**
  - Write Data to Transmit Holding Register (THR).
  - Data in THR is transferred to Transmit Shift Register (TSR) when TSR is empty.
  - TSR shifts the data out on the TX output pin.

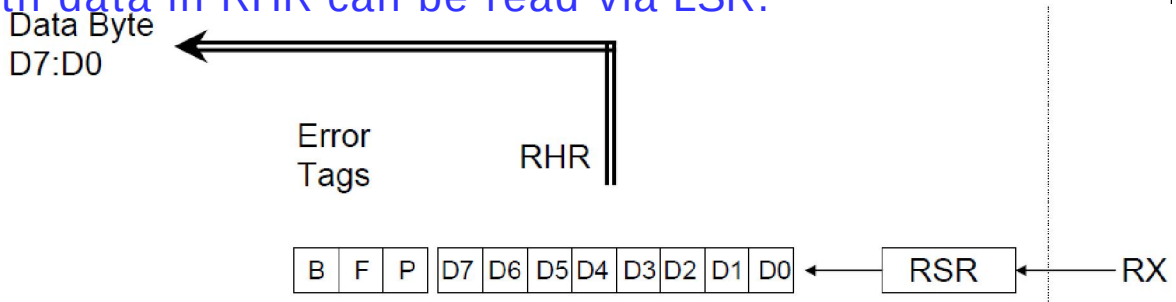


- **FIFO mode:**
  - Write Data to Transmit Holding Register (THR).
  - Transmit data is queued in TX FIFO.
  - Data in TX FIFO is transferred to Transmit Shift Register (TSR) when TSR is empty.
  - TSR shifts data out on TX output pin.

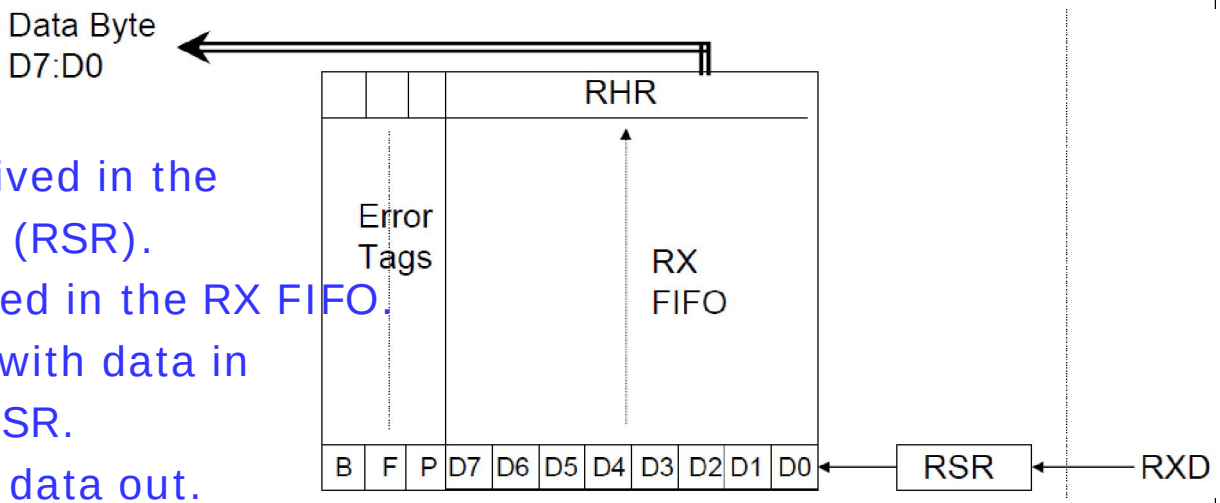


UART: Receiver

- **Non FIFO mode:**
  - Incoming data is received in the Receive Shift Register (RSR).
  - Received data is transferred to the RHR.
  - Error tags associated with data in RHR can be read via LSR.
  - Read RHR to read the data out.

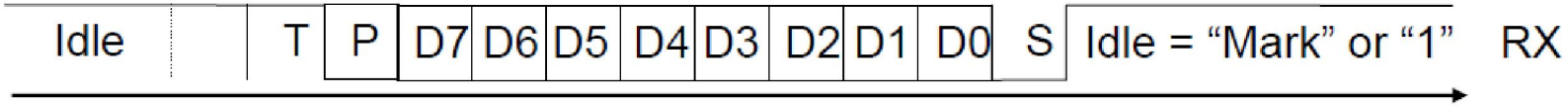


- **FIFO mode:**
  - Incoming data is received in the Receive Shift Register (RSR).
  - Received data is queued in the RX FIFO.
  - Error tags associated with data in RHR can be read via LSR.
  - Read RHR to read the data out.



UART: RX Character Validation

- Start bit detection and validation:
  - HIGH to LOW transition indicates a start bit.
  - Start bit validated if RX input is still LOW during mid bit sampling.
- Data, parity and stop bits are sampled at mid bit.
- A valid stop bit is HIGH when the stop bit is sampled.



UART: RX Error Reporting

- **Line Status errors**
  - Error tags are associated with each byte:
    - Framing error if stop bit is not detected.
    - Parity error if parity bit is incorrect.
    - Break detected if RX input is LOW for duration of one character time and stop bit is not detected.
- Overrun error if character is received in RSR when RX FIFO is full.
  - Non-FIFO mode:
    - RHR has a data byte and data received in RSR.
    - RSR data overwrites RHR data.
  - FIFO mode:
    - RX FIFO is full and data is received in RSR.
    - Data in RX FIFO is not overwritten by data in RSR.

UART: Baudrate

o Configurable baudrate:

$$UARTn_{baudrate} = \frac{PCLK}{16 \times (256 \times UnDLM + UnDLL) \times \left(1 + \frac{DivAddVal}{MulVal}\right)}$$

- n **UnDLM**, **UnDLL** are **8 bits data**.
- n 1 ≤ **MULVAL** ≤15
- n 0 ≤ **DIVADDVAL** ≤14
- n DIVADDVAL ≤ MULVAL

Reset

Table 286. Fractional Divider setting look-up table

FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal
1.000	0/1	1.250	1/4	1.500	1/2	1.750	3/4
1.067	1/15	1.267	4/15	1.533	8/15	1.769	10/13
1.071	1/14	1.273	3/11	1.538	7/13	1.778	7/9
1.077	1/13	1.286	2/7	1.545	6/11	1.786	11/14
1.083	1/12	1.300	3/10	1.556	5/9	1.800	4/5
1.091	1/11	1.308	4/13	1.571	4/7	1.818	9/11
1.100	1/10	1.333	1/3	1.583	7/12	1.833	5/6
1.111	1/9	1.357	5/14	1.600	3/5	1.846	11/13
1.125	1/8	1.364	4/11	1.615	8/13	1.857	6/7
1.133	2/15	1.375	3/8	1.625	5/8	1.867	13/15
1.143	1/7	1.385	5/13	1.636	7/11	1.875	7/8
1.154	2/13	1.400	2/5	1.643	9/14	1.889	8/9
1.167	1/6	1.417	5/12	1.667	2/3	1.900	9/10
1.182	2/11	1.429	3/7	1.692	9/13	1.909	10/11
1.200	1/5	1.444	4/9	1.700	7/10	1.917	11/12
1.214	3/14	1.455	5/11	1.714	5/7	1.923	12/13
1.222	2/9	1.462	6/13	1.727	8/11	1.929	13/14
1.231	3/13	1.467	7/15	1.733	11/15	1.933	14/15

Table 285: UARTn Fractional Divider Register (U0FDR - address 0x4000 C028, U2FDR - 0x4009 8028, U3FDR - 0x4009 C028) bit description

Bit	Function	Value	Description	Reset value
3:0	DIVADDVAL	0	Baud-rate generation pre-scaler divisor value. If this field is 0, fractional baud-rate generator will not impact the UARTn baudrate.	0
7:4	MULVAL	1	Baud-rate pre-scaler multiplier value. This field must be greater or equal 1 for UARTn to operate properly, regardless of whether the fractional baud-rate generator is used or not.	1
31:8	-	NA	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	0

- n Example -> UART0 (19200 baudios)

```
LPC_UART0->LCR |= DLAB_ENABLE; // importante poner a 1
LPC_UART0->DLM =0;
LPC_UART0->DLL =81;
LPC_UART0->LCR &= ~DLAB_ENABLE;// importante poner a 0
```

$$U0DL_{16} = \frac{PCLK[Hz]}{16 \cdot V_t[baudios]} = \frac{100^6/4}{16 \cdot 19200} = 81.38$$

UART: Register Map (I)

Table 270. UART0/2/3 Register Map

Generic Name	Description	Access	Reset value[1]	UARTn Register Name & Address
RBR (DLAB =0)	Receiver Buffer Register. Contains the next received character to be read.	RO	NA	U0RBR - 0x4000 C000 U2RBR - 0x4009 8000 U3RBR - 0x4009 C000
THR (DLAB =0)	Transmit Holding Register. The next character to be transmitted is written here.	WO	NA	U0THR - 0x4000 C000 U2THR - 0x4009 8000 U3THR - 0x4009 C000
DLL (DLAB =1)	Divisor Latch LSB. Least significant byte of the baud rate divisor value. The full divisor is used to generate a baud rate from the fractional rate divider.	R/W	0x01	U0DLL - 0x4000 C000 U2DLL - 0x4009 8000 U3DLL - 0x4009 C000
DLM (DLAB =1)	Divisor Latch MSB. Most significant byte of the baud rate divisor value. The full divisor is used to generate a baud rate from the fractional rate divider.	R/W	0x00	U0DLM - 0x4000 C004 U2DLM - 0x4009 8004 U3DLM - 0x4009 C004
IER (DLAB =0)	Interrupt Enable Register. Contains individual interrupt enable bits for the 7 potential UART interrupts.	R/W	0x00	U0IER - 0x4000 C004 U2IER - 0x4009 8004 U3IER - 0x4009 C004
IIR	Interrupt ID Register. Identifies which interrupt(s) are pending.	RO	0x01	U0IIR - 0x4000 C008 U2IIR - 0x4009 8008 U3IIR - 0x4009 C008
FCR	FIFO Control Register. Controls UART FIFO usage and modes.	WO	0x00	U0FCR - 0x4000 C008 U2FCR - 0x4009 8008 U3FCR - 0x4009 C008
LCR	Line Control Register. Contains controls for frame formatting and break generation.	R/W	0x00	U0LCR - 0x4000 C00C U2LCR - 0x4009 800C U3LCR - 0x4009 C00C
LSR	Line Status Register. Contains flags for transmit and receive status, including line errors.	RO	0x60	U0LSR - 0x4000 C014 U2LSR - 0x4009 8014 U3LSR - 0x4009 C014

UART: Register Map (II)

Table 270. UART0/2/3 Register Map

Generic Name	Description	Access	Reset value[1]	UARTn Register Name & Address
SCR	Scratch Pad Register. 8-bit temporary storage for software.	R/W	0x00	U0SCR - 0x4000 C01C U2SCR - 0x4009 801C U3SCR - 0x4009 C01C
ACR	Auto-baud Control Register. Contains controls for the auto-baud feature.	R/W	0x00	U0ACR - 0x4000 C020 U2ACR - 0x4009 8020 U3ACR - 0x4009 C020
ICR	IrDA Control Register. Enables and configures the IrDA mode.	R/W	0x00	U0ICR - 0x4000 C024 U2ICR - 0x4009 8024 U3ICR - 0x4009 C024
FDR	Fractional Divider Register. Generates a clock input for the baud rate divider.	R/W	0x10	U0FDR - 0x4000 C028 U2FDR - 0x4009 8028 U3FDR - 0x4009 C028
TER	Transmit Enable Register. Turns off UART transmitter for use with software flow control.	R/W	0x80	U0TER - 0x4000 C030 U2TER - 0x4009 8030 U3TER - 0x4009 C030

[1] Reset Value reflects the data stored in used bits only. It does not include reserved bits content.

UART: Rx Buffer Register ,Tx Holding Register

Table 271: UARTn Receiver Buffer Register (U0RBR - address 0x4000 C000, U2RBR - 0x4009 8000, U3RBR - 04009 C000 when DLAB = 0) bit description

Bit	Symbol	Description	Reset Value
7:0	RBR	The UARTn Receiver Buffer Register contains the oldest received byte in the UARTn Rx FIFO.	Undefined
31:8	-	Reserved, the value read from a reserved bit is not defined.	NA

Table 272: UARTn Transmit Holding Register (U0THR - address 0x4000 C000, U2THR - 0x4009 8000, U3THR - 0x4009 C000 when DLAB = 0) bit description

Bit	Symbol	Description	Reset Value
7:0	THR	Writing to the UARTn Transmit Holding Register causes the data to be stored in the UARTn transmit FIFO. The byte will be sent when it reaches the bottom of the FIFO and the transmitter is available.	NA
31:8	-	Reserved, user software should not write ones to reserved bits.	NA

UART: Divisor Latch LSB, MSB Registers

Table 273: UARTn Divisor Latch LSB register (U0DLL - address 0x4000 C000, U2DLL - 0x4009 8000, U3DLL - 0x4009 C000 when DLAB = 1) bit description

Bit	Symbol	Description	Reset Value
7:0	DLLSB	The UARTn Divisor Latch LSB Register, along with the UnDLM register, determines the baud rate of the UARTn.	0x01
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Table 274: UARTn Divisor Latch MSB register (U0DLM - address 0x4000 C004, U2DLM - 0x4009 8004, U3DLM - 0x4009 C004 when DLAB = 1) bit description

Bit	Symbol	Description	Reset Value
7:0	DLMSB	The UARTn Divisor Latch MSB Register, along with the U0DLL register, determines the baud rate of the UARTn.	0x00
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

- Example set baudrate (FR=1):

$$U0DL_{16} = \frac{PCLK[Hz]}{16 \cdot V_t[baudios]}$$

```
DL=F_pclk/16*baud; // Round to the nearest whole!!!!
LPC_UART0->DLL= DL%256; //LSB
LPC_UART0->DLM= DL/256;  //MSB
```

UART: Interrupt Enable Register

Table 275: UARTn Interrupt Enable Register (U0IER - address 0x4000 C004, U2IER - 0x4009 8004, U3IER - 0x4009 C004 when DLAB = 0) bit description

Bit	Symbol	Value	Description	Reset Value
0	RBR Interrupt Enable		Enables the Receive Data Available interrupt for UARTn. It also controls the Character Receive Time-out interrupt.	0
		0	Disable the RDA interrupts.	
		1	Enable the RDA interrupts.	
1	THRE Interrupt Enable		Enables the THRE interrupt for UARTn. The status of this can be read from UnLSR[5].	0
		0	Disable the THRE interrupts.	
		1	Enable the THRE interrupts.	
2	RX Line Status Interrupt Enable		Enables the UARTn RX line status interrupts. The status of this interrupt can be read from UnLSR[4:1].	0
		0	Disable the RX line status interrupts.	
		1	Enable the RX line status interrupts.	
7:3	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
8	ABEOIntEn		Enables the end of auto-baud interrupt.	0
		0	Disable end of auto-baud Interrupt.	
		1	Enable end of auto-baud Interrupt.	
9	ABTOIntEn		Enables the auto-baud time-out interrupt.	0
		0	Disable auto-baud time-out Interrupt.	
		1	Enable auto-baud time-out Interrupt.	
31:10	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

UART: Interrupt Identification Register

Table 276: UARTn Interrupt Identification Register (U0IIR - address 0x4000 C008, U2IIR - 0x4009 8008, U3IIR - 0x4009 C008) bit description

Bit	Symbol	Value	Description	Reset Value
0	IntStatus		Interrupt status. Note that UnIIR[0] is active low. The pending interrupt can be determined by evaluating UnIIR[3:1].	1
		0	At least one interrupt is pending.	
		1	No interrupt is pending.	
3:1	IntId		Interrupt identification. UnIER[3:1] identifies an interrupt corresponding to the UARTn Rx or TX FIFO. All other combinations of UnIER[3:1] not listed below are reserved (000,100,101,111).	0
		011	1 - Receive Line Status (RLS).	
		010	2a - Receive Data Available (RDA). ★	
		110	2b - Character Time-out Indicator (CTI).	
		001	3 - THRE Interrupt ★	
5:4	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
7:6	FIFO Enable		Copies of UnFCR[0].	0
8	ABEOInt		End of auto-baud interrupt. True if auto-baud has finished successfully and interrupt is enabled.	0
9	ABTOInt		Auto-baud time-out interrupt. True if auto-baud has timed out and interrupt is enabled.	0
31:10	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

UART: Interrupt Handling

Table 277: UARTn Interrupt Handling

U0IIR[3:0] value[1]	Priority	Interrupt Type	Interrupt Source	Interrupt Reset
0001	-	None	None	-
0110	Highest	RX Line Status / Error	OE[2] or PE[2] or FE[2] or BI[2]	UnLSR Read[2]
0100 ★	Second	RX Data Available	Rx data available or trigger level reached in FIFO (UnFCR0=1)	UnRBR Read[3] or UARTn FIFO drops below trigger level
1100	Second	Character Time-out indication	Minimum of one character in the Rx FIFO and no character input or removed during a time period depending on how many characters are in FIFO and what the trigger level is set at (3.5 to 4.5 character times). The exact time will be: [(word length) × 7 - 2] × 8 + [(trigger level - number of characters) × 8 + 1] RCLKs	UnRBR Read[3]
0010 ★	Third	THRE	THRE[2]	UnIIR Read (if source of interrupt) or THR write[4]

UART: FIFO Control Register

Table 278: UARTn FIFO Control Register (U0FCR - address 0x4000 C008, U2FCR - 0x4009 8008, U3FCR - 0x4007 C008) bit description

Bit	Symbol	Value	Description	Reset Value
0	FIFO Enable	0	UARTn FIFOs are disabled. Must not be used in the application.	0
		1	Active high enable for both UARTn Rx and TX FIFOs and UnFCR[7:1] access. This bit must be set for proper UART operation. Any transition on this bit will automatically clear the related UART FIFOs.	
1	RX FIFO Reset	0	No impact on either of UARTn FIFOs.	0
		1	Writing a logic 1 to UnFCR[1] will clear all bytes in UARTn Rx FIFO, reset the pointer logic. This bit is self-clearing.	
2	TX FIFO Reset	0	No impact on either of UARTn FIFOs.	0
		1	Writing a logic 1 to UnFCR[2] will clear all bytes in UARTn TX FIFO, reset the pointer logic. This bit is self-clearing.	
3	DMA Mode Select		When the FIFO enable bit (bit 0 of this register) is set, this bit selects the DMA mode. See Section 14.4.6.1.	0
5:4	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
7:6	RX Trigger Level		These two bits determine how many receiver UARTn FIFO characters must be written before an interrupt or DMA request is activated.	0
		00	Trigger level 0 (1 character or 0x01)	
		01	Trigger level 1 (4 characters or 0x04)	
		10	Trigger level 2 (8 characters or 0x08)	
		11	Trigger level 3 (14 characters or 0x0E)	
31:8	-		Reserved, user software should not write ones to reserved bits.	NA

UART: Line Control Register

Table 279: UARTn Line Control Register (U0LCR - address 0x4000 C00C, U2LCR - 0x4009 800C, U3LCR - 0x4009 C00C) bit description

Bit	Symbol	Value	Description	Reset Value
1:0	Word Length Select	00	5-bit character length	0
		01	6-bit character length	
		10	7-bit character length	
		11	8-bit character length	
2	Stop Bit Select	0	1 stop bit.	0
		1	2 stop bits (1.5 if UnLCR[1:0]=00).	
3	Parity Enable	0	Disable parity generation and checking.	0
		1	Enable parity generation and checking.	
5:4	Parity Select	00	Odd parity. Number of 1s in the transmitted character and the attached parity bit will be odd.	0
		01	Even Parity. Number of 1s in the transmitted character and the attached parity bit will be even.	
		10	Forced "1" stick parity.	
		11	Forced "0" stick parity.	
6	Break Control	0	Disable break transmission.	0
		1	Enable break transmission. Output pin UARTn TXD is forced to logic 0 when UnLCR[6] is active high.	
7	Divisor Latch Access Bit (DLAB)	0	Disable access to Divisor Latches.	0
		1	Enable access to Divisor Latches.	
31:8	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

UART: Line Status Register

Table 280: UARTn Line Status Register (U0LSR - address 0x4000 C014, U2LSR - 0x4009 8014, U3LSR - 0x4009 C014) bit description

Bit	Symbol	Value	Description	Reset Value
0	Receiver Data Ready (RDR)		UnLSR0 is set when the UnRBR holds an unread character and is cleared when the UARTn RBR FIFO is empty.	0
		0	The UARTn receiver FIFO is empty.	
		1	The UARTn receiver FIFO is not empty.	
1	Overrun Error (OE)		The overrun error condition is set as soon as it occurs. An UnLSR read clears UnLSR1. UnLSR1 is set when UARTn RSR has a new character assembled and the UARTn RBR FIFO is full. In this case, the UARTn RBR FIFO will not be overwritten and the character in the UARTn RSR will be lost.	0
		0	Overrun error status is inactive.	
		1	Overrun error status is active.	
2	Parity Error (PE)		When the parity bit of a received character is in the wrong state, a parity error occurs. An UnLSR read clears UnLSR[2]. Time of parity error detection is dependent on UnFCR[0]. **Note:** A parity error is associated with the character at the top of the UARTn RBR FIFO.	0
		0	Parity error status is inactive.	
		1	Parity error status is active.	
3	Framing Error (FE)		When the stop bit of a received character is a logic 0, a framing error occurs. An UnLSR read clears UnLSR[3]. The time of the framing error detection is dependent on UnFCR0. Upon detection of a framing error, the Rx will attempt to resynchronize to the data and assume that the bad stop bit is actually an early start bit. However, it cannot be assumed that the next received byte will be correct even if there is no Framing Error. **Note:** A framing error is associated with the character at the top of the UARTn RBR FIFO.	0
		0	Framing error status is inactive.	
		1	Framing error status is active.	

UART: Line Status Register (continued)

Table 280: UARTn Line Status Register (U0LSR - address 0x4000 C014, U2LSR - 0x4009 8014, U3LSR - 0x4009 C014) bit description *...continued*

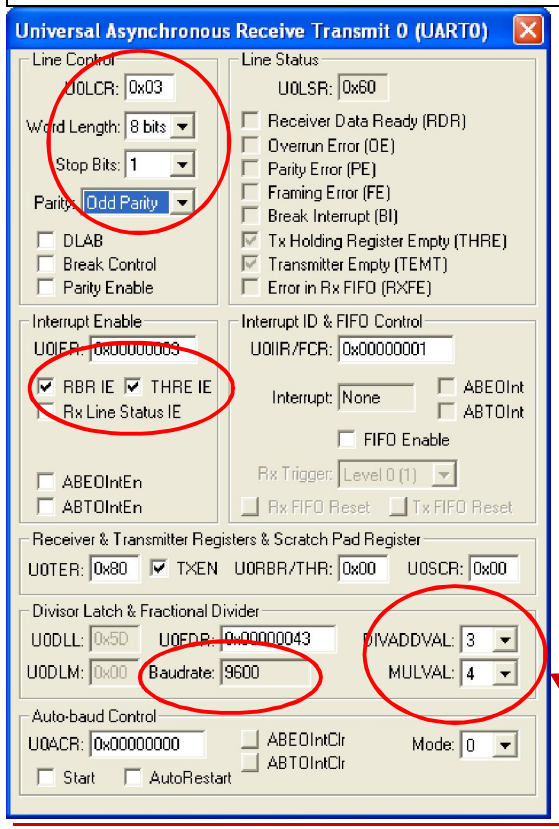
Bit	Symbol	Value	Description	Reset Value
4	Break Interrupt (BI)		When RXDn is held in the spacing state (all zeroes) for one full character transmission (start, data, parity, stop), a break interrupt occurs. Once the break condition has been detected, the receiver goes idle until RXDn goes to marking state (all ones). An UnLSR read clears this status bit. The time of break detection is dependent on UnFCR[0]. **Note:** The break interrupt is associated with the character at the top of the UARTn RBR FIFO.	0
		0	Break interrupt status is inactive.	
		1	Break interrupt status is active.	
5	Transmitter Holding Register Empty (THRE))		THRE is set immediately upon detection of an empty UARTn THR and is cleared on a UnTHR write.	1
		0	UnTHR contains valid data.	
		1	UnTHR is empty.	
6	Transmitter Empty (TEMT)		TEMT is set when both UnTHR and UnTSR are empty; TEMT is cleared when either the UnTSR or the UnTHR contain valid data.	1
		0	UnTHR and/or the UnTSR contains valid data.	
		1	UnTHR and the UnTSR are empty.	
7	Error in RX FIFO (RXFE)		UnLSR[7] is set when a character with a Rx error such as framing error, parity error or break interrupt, is loaded into the UnRBR. This bit is cleared when the UnLSR register is read and there are no subsequent errors in the UARTn FIFO.	0
		0	UnRBR contains no UARTn RX errors or UnFCR[0]=0.	
		1	UARTn RBR contains at least one UARTn RX error.	
31:8	-		Reserved, the value read from a reserved bit is not defined.	NA

UART: Software example (I)

```
void uart0_init(int baudrate) {

    LPC_PINCON->PINSEL0 = (1 << 4) | (1 << 6);     // Change P0.2 and P0.3 mode to TXD0 and RXD0

    // Set 8N1 mode (8 bits/dato, sin paridad, y 1 bit de stop)
    LPC_UART0->LCR |= CHAR_8_BIT | STOP_1_BIT | PARITY_NONE;

    uart0_set_baudrate(baudrate);                  // Set the baud rate

    LPC_UART0->IER = THRE_IRQ_ENABLE | RBR_IRQ_ENABLE; // Enable UART TX and RX interrupt (for LPC17xx UART)
    NVIC_EnableIRQ(UART0_IRQn);                    // Enable the UART interrupt (for Cortex-CM3 NVIC)

}
```

```
LPC_UART0->LCR |= DLAB_ENABLE; // importante poner a 1
LPC_UART0->DLM = DL / 256; // parte alta
LPC_UART0->DLL = DL % 256; // parte baja
LPC_UART0->LCR &= ~DLAB_ENABLE; // importante poner a 0
```

if (FR = 1)
DL = 162.73
Baudrate = 9585 bps



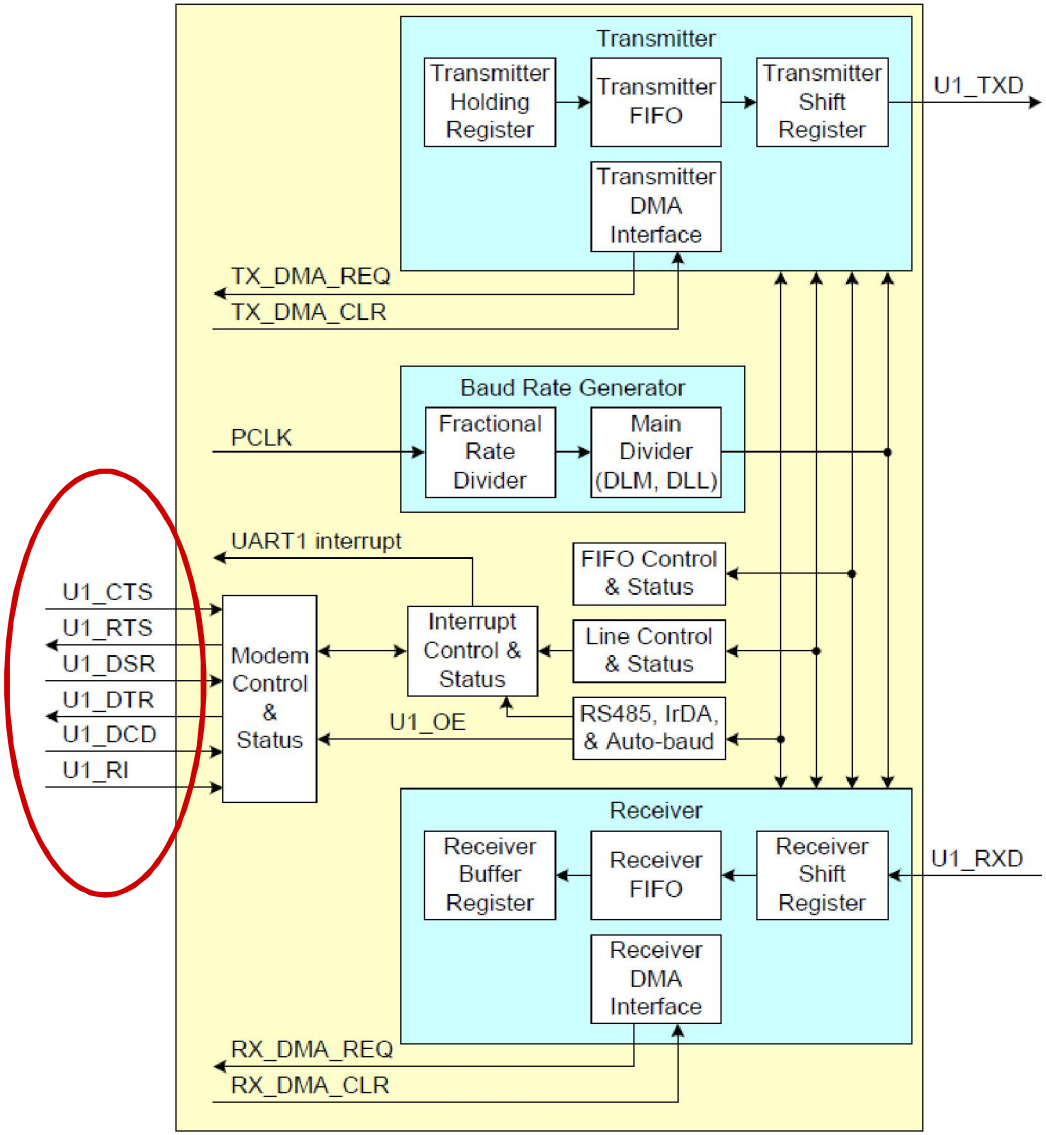

```
static int uart0_set_baudrate(unsigned int baudrate) {
    int errorStatus = -1; //< Failure

    // UART clock (FCCO / PCLK_UART0)
    // unsigned int uClk = SystemFrequency / 4;
    unsigned int uClk =SystemCoreClock/4;
    unsigned int calcBaudrate = 0;
    unsigned int temp = 0;

    unsigned int mulFracDiv, dividerAddFracDiv;
    unsigned int divider = 0;
    unsigned int mulFracDivOptimal = 1;
    unsigned int dividerAddOptimal = 0;
    unsigned int dividerOptimal = 0;

    unsigned int relativeError = 0;
    unsigned int relativeOptimalError = 100000;

    uClk = uClk >> 4; /* div by 16 */

    /*
     * The formula is :
     * BaudRate= uClk * (mulFracDiv/(mulFracDiv+dividerAddFracDiv) / (16 * DLL)
     *
     * The value of mulFracDiv and dividerAddFracDiv should comply to the following expressions:
     * 0 < mulFracDiv <= 15, 0 <= dividerAddFracDiv <= 15
     */
    for (mulFracDiv = 1; mulFracDiv <= 15; mulFracDiv++) {
        for (dividerAddFracDiv = 0; dividerAddFracDiv <= 15; dividerAddFracDiv++) {
            temp = (mulFracDiv * uClk) / (mulFracDiv + dividerAddFracDiv);

            divider = temp / baudrate;
            if ((temp % baudrate) > (baudrate / 2))
```

UART: Software example (I)

```
void tx_cadena_UART0(char *cadena)
{
   ptr_tx=cadena;
   tx_completa=0;
   LPC_UART0->THR = *ptr_tx;   // IMPORTANTE: Introducir un carácter al comienzo para iniciar TX o
}                              // activar flag interrupción por registro transmisor vacio
```

```
void UART0_IRQHandler(void) {

   switch(LPC_UART0->IIR&0x0E) {

   case 0x04:                  /* RBR, Receiver Buffer Ready */
     *ptr_rx=LPC_UART0->RBR;   /* lee el dato recibido y lo almacena */
     if(*ptr_rx++ ==13){       /* Caracter return --> Cadena completa */
       *ptr_rx=0;              /* Añadimos el caracter null para tratar los datos recibidos como una cadena*/
       rx_completa = 1;        /* rx completa */
       ptr_rx=buffer;          /* puntero al inicio del buffer para nueva recepción */
     }
     break;

   case 0x02:                      /* THRE, Transmit Holding Register empty */
     if(*ptr_tx!=0)
       LPC_UART0->THR = *ptr_tx++; /* carga un nuevo dato para ser transmitido */
     else
       tx_completa=1;
     break;
   }
}
```

UART1: Modem

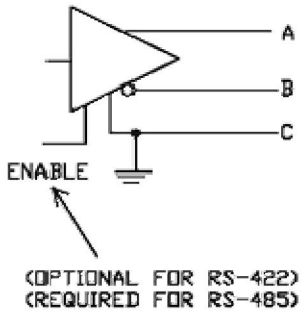

Transmitter
Transmitter Holding Register
Transmitter FIFO
Transmitter Shift Register
U1_TXD
Transmitter DMA Interface
TX_DMA_REQ
TX_DMA_CLR
Baud Rate Generator
PCLK
Fractional Rate Divider
Main Divider (DLM, DLL)
UART1 interrupt
FIFO Control & Status
U1_CTS
U1_RTS
U1_DSR
U1_DTR
U1_DCD
U1_RI
Modem Control & Status
Interrupt Control & Status
Line Control & Status
U1_OE
RS485, IrDA, & Auto-baud
Receiver
Receiver Buffer Register
Receiver FIFO
Receiver Shift Register
U1_RXD
Receiver DMA Interface
RX_DMA_REQ
RX_DMA_CLR

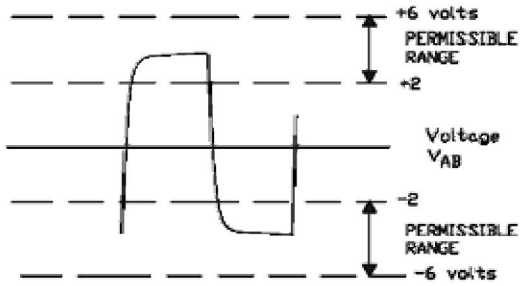
UART1: RS-485

- **What is RS-485?**
  - RS-485 is a EIA standard interface which is very common in the data acquisition world.
  - RS-485 provides balanced transmission line which also can be shared in Multi-drop mode.
  - It allows high data rates communications over long distances in real world environments.
- **How fast can RS-485 be?**
  - RS-485 was designed for greater distance and higher baudrates than RS-232.
  - According to the standard, 100kbit/s is the maximum speed and distance up to 4000 feet (**1200 meters**) can be achieved.

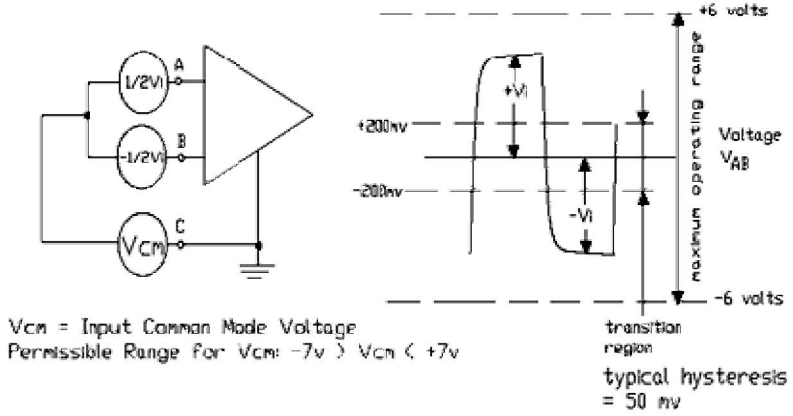
UART1: RS-485 Line Driver

- **Balanced Line Drivers**
  - Voltage produced by the driver appears across **a pair of signal wires** that transmit only one signal. Both wires are driven opposite.
  - RS-485 driver has always the “Enable” direction control signal.
  - **Differential system provides noise immunity**, because much of the common mode signal can be rejected by the receiver. So ground shifts and induced noise signals can be nullified.





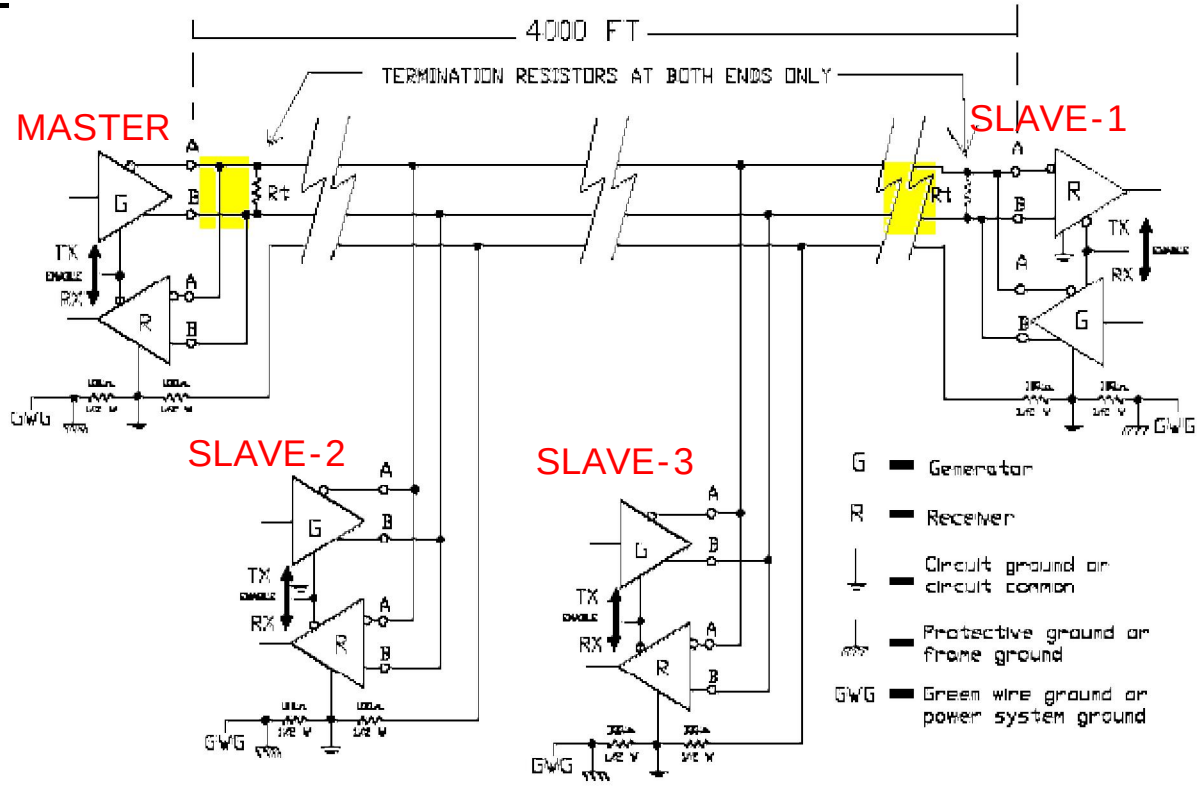
BALANCED DIFFERENTIAL OUTPUT
LINE DRIVER



BALANCED DIFFERENTIAL INPUT
LINE RECEIVER

UART1: RS-485 Network

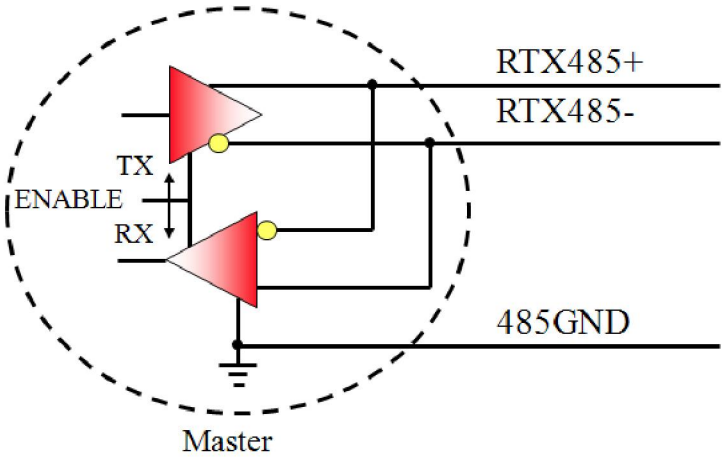
- RS-485 provides Half-Duplex, Multi-drop communications over a single twisted pair cable.
  - The standard specifies up to 32 drivers and 32 receivers can share a multidrop network.
  - Terminator resistors avoid reflected signal

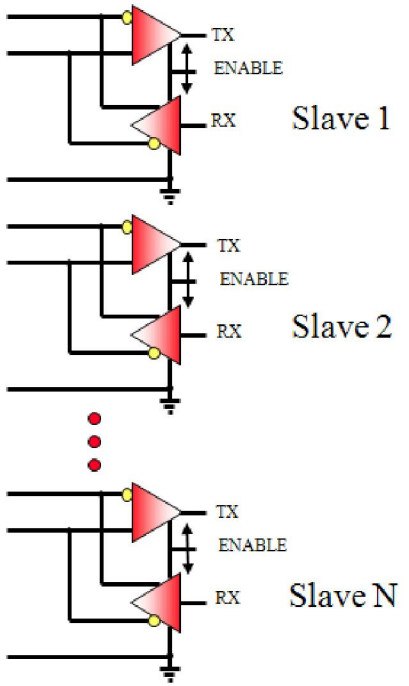


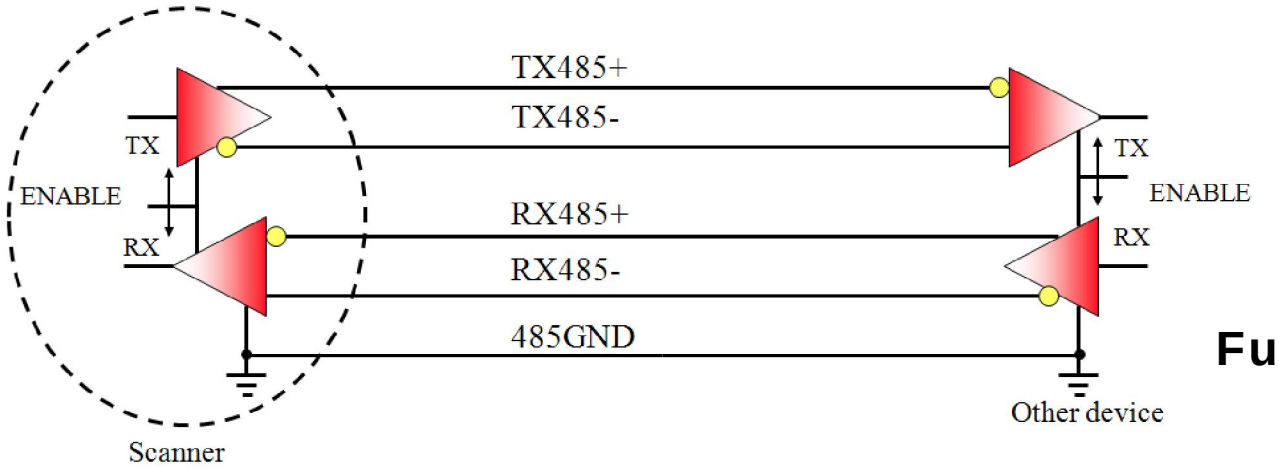
TYPCIAL RS-485 TWO WIRE MULTIDROP NETWORK

UART1: RS-485 Half and Full Duplex

Half-Duplex







Full-Duplex

UART1: RS-485 vs RS-232

	RS-232	RS-485
Mode of Operation	SINGLE-ENDED	DIFFERENTIAL
Total Number of Drivers and Receivers on One Line	1 DRIVER 1 RECEIVER	32 DRIVER 32 RECEIVER
Maximum Cable Length	50 FEET	4000 FEET
Maximum Data Rate @Max length	20kb/s	100kb/s
Driver Output Signal Level (Loaded Min.) Loaded	+/-5V to +/-15V	+/-1.5V
Driver Output Signal Level (Unloaded Max) Unloaded	+/-25V	+/-6V
Driver Load Impedance	3kΩ to 7kΩ	54Ω
Max. Driver Current in High Z State Power On	N/A	N/A
Max. Driver Current in High Z State Power Off	+/-6mA @ +/-2v	+/-100uA
Slew Rate (Max.)	30V/μS	N/A
Receiver Input Voltage Range	+/-15V	-7V to +12V
Receiver Input Sensitivity	+/-3V	+/-200mV
Receiver Input Resistance	3kΩ to 7kΩ	≥ 12kΩ

UART1: RS-485 modes of operation

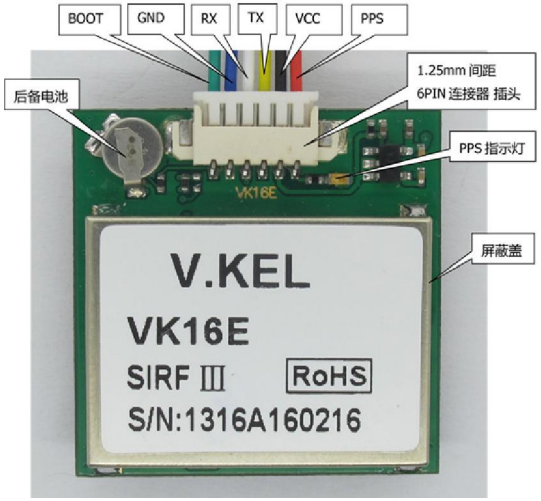
- Normal Multi-drop Mode (NMM)
- Auto Address Detection (AAD)
- Auto Direction Control
- Output inversion

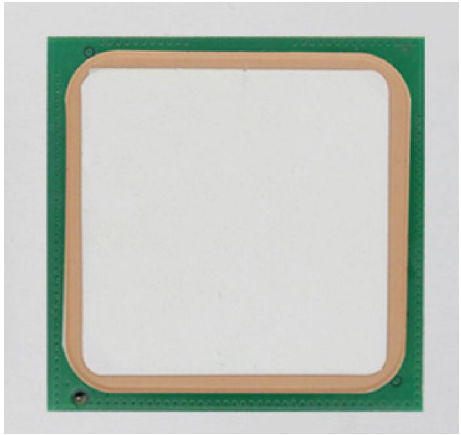
Table 308: UART1 RS485 Control register (U1RS485CTRL - address 0x4001 004C) bit description

Bit	Symbol	Value	Description	Reset value
0	NMMEN	0	RS-485/EIA-485 Normal Multidrop Mode (NMM) is disabled.	0
		1	RS-485/EIA-485 Normal Multidrop Mode (NMM) is enabled. In this mode, an address is detected when a received byte causes the UART to set the parity error and generate an interrupt.	
1	RXDIS	0	The receiver is enabled.	0
		1	The receiver is disabled.	
2	AADEN	0	Auto Address Detect (AAD) is disabled.	0
		1	Auto Address Detect (AAD) is enabled.	
3	SEL	0	If direction control is enabled (bit DCTRL = 1), pin $\overline{\text{RTS}}$ is used for direction control.	0
		1	If direction control is enabled (bit DCTRL = 1), pin DTR is used for direction control.	
4	DCTRL	0	Disable Auto Direction Control.	0
		1	Enable Auto Direction Control.	

UART: Devices with RS-232 interface

- **GPS Module** (VK16E GMOUSE GPS Module SIRF III)





- Built with fast positioning and the ability to track 20 satellites SIRF III generation chip.
- Built-in backup battery.
- Built-in high gain LNA.
- With selectable baud rate: 4800,**9600**,19200,38400
- NMEA format (ASCII):
  - RMC, GSV, GSA, GGA, VTG, GLL.

  Data example: in Polytechnic School !!!

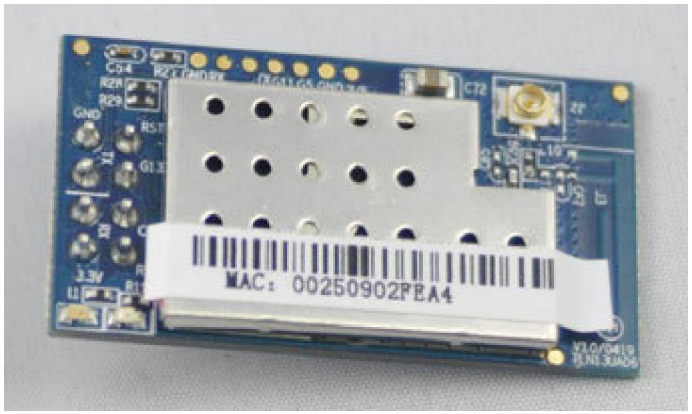
```
$GPGGA,104951.000,4030.8105,N,00321.0170,W,1,04,3.9,584.8,M,51.7,M,,0000*47
$GPRMC,104951.000,A,4030.8105,N,00321.0170,W,0.00,182.93,190214,,,A*77
$GPVTG,182.93,T,,M,0.00,N,0.0,K,A*0C
$GPGGA,104952.000,4030.8105,N,00321.0170,W,1,04,3.9,584.8,M,51.7,M,,0000*44
$GPRMC,104952.000,A,4030.8105,N,00321.0170,W,0.00,182.93,190214,,,A*74
$GPVTG,182.93,T,,M,0.00,N,0.0,K,A*0C
```

- The default output for the 9600 baud standard.

UART: Devices with RS-232 interface

o 2.4G Wireless WIFI Module (TLG10UA03)

- Full support for serial transparent data transfer mode, really Positive Serial Plug and Play.
- The new AT command set, all based on ASCII.
- Complete TCP / IP protocol stack to support DHCP protocol and DNS dynamic IP address assignment domain name resolution function.
- Built-in WEB server, implemented using IE browser.
- Remote configuration via the wireless network module parameters.
- Supports Frequency range: 2.412 ~ 2.484 GHz.
- Supports two types of wireless networks:
  Infrastructure Network (Infra) and ad hoc networks (Adhoc).
- Support multiple security authentication mechanisms:
  WEP64/WEP128 / TKIP / CCMP (AES)
  WEP/WPA-PSK/WPA2-PSK.
- Support for fast networking.
- Supports wireless roaming.



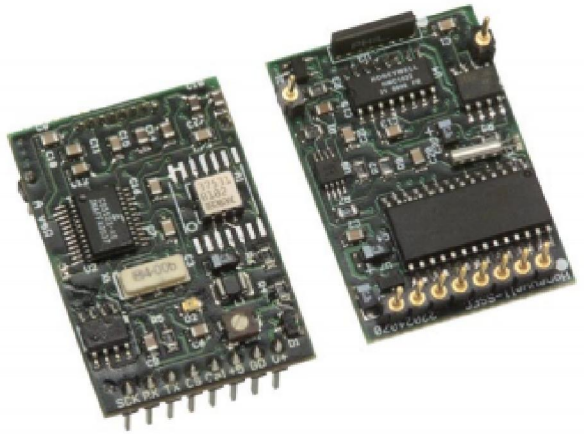
- Support for multiple network protocols: TCP / UDP / ICMP / DHCP / DNS / HTTP
- Supports automatic two operating mode and command support transparent transmission mode.
- Support AT command set controls.
- Support for a variety of parameter configuration mode: Serial / WEB server / Wireless connection.

UART: Devices with RS-232 interface

o Digital Compass (HMR3200/HMR3300)

• The HMR3200 is a two-axis precision compass with three orthogonal magnetoresistive sensors, and can be used in either vertical or horizontal orientations.

• The HMR3300 includes a MEMS accelerometer for a horizontal three-axis, tilt compensated precision compass for performance up to a ±60° tilt range.



o Optical Fingerprint Sensor (EM404)

• EM404 integrated fingerprint verification module is the latest release of HF&CCTV. It consists of optical fingerprint sensor, high performance DSP processor and Flash.

• It boasts of functions such as fingerprint enrollment, fingerprint deletion, fingerprint verification, fingerprint upload, fingerprint download, etc.

UART: Devices with RS-232 interface

- **Camera module** (CMOS 1/4 inches Image Sensor JPG)

• Command instruction:

•1 Reset command: 56002600

•2 Photographing command: 5600360100

•3 Read the data length of the captured image: 5600340100
Return: 76 00 34 00 04 00 00 XX YY
XX YY ------- picture data length, XX is the high byte, YY is the low byte.

•4 Read the picture data: 56 00 32 0C 00 0A 00 00 XX XX 00 00 YY YY ZZ ZZ
Return: 7600320000 (interval) FF D8. . . . . FF D9 (interval) 7600320000
00 00 XX XX ------ starting address (starting address must be a multiple of 8, generally 00 00)
00 00 YY YY ------ picture data length (high byte first, then low byte) ZZ ZZ -------- interval (= XX XX * 0.01 milliseconds, preferably a small number, such as 00 0A)
Note: JPEG picture file data must begin with FF D8, and end with FF D9.

•5 Stop shooting: 5600360103

•6 Set camera image compression command: 56 00 31 05 01 01 12 04 XX
XX generally is 36 (range: 00 ---- FF)

•7 Set the camera image size: (default size: 320 * 240)
320 * 240: 56 00 31 05 04 01 00 19 11
640 * 480: 56 00 31 05 04 01 00 19 00

•8 Get into sleep mode: 56 00 3E 03 00 01 01

•9 Modify baud rate: 56 00 24 03 01 XX XX

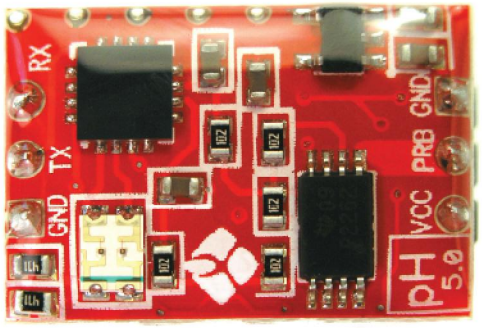
UART: Devices with RS-232 interface

MLX90614 Infrared Thermometer Module

Bluetooth to Serial Slave

Fingerprint Module Capacitive Sensor Em401

TC35 SMS Module Board



pH sensor Module (AtlasScientific)

NFC Reader Module (pn532)