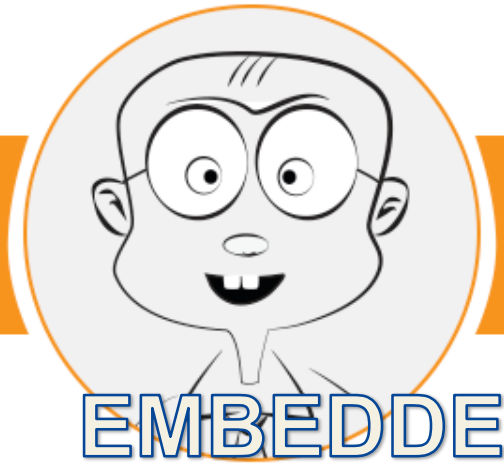


EMBEDDED SYSTEM COURSE

LECTURE 8: PERIPHERALS UART

Learning Goals

- Understanding basis concepts about UART and how to transmit/receive data via UART connection.
- Introduce about RS232 protocol.
- Understanding on how to configure the KL46 UART module.

Table of contents

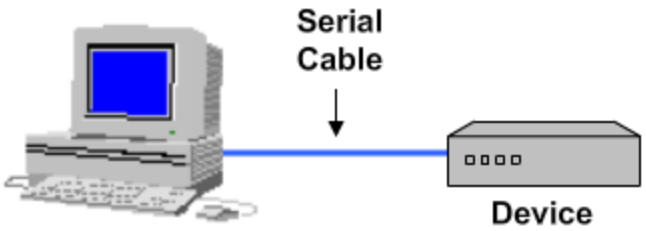
Table of contents

Introduction to UART

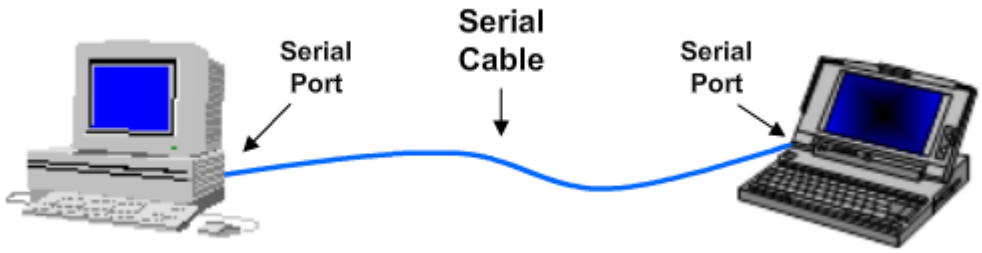
- UART: Universal Asynchronous Receiver/Transmitter
- A UART maybe used when:
  - High speed is not required
  - An inexpensive communication link between two devices is required
- UART communication is very cheap:
  - Single wire for each direction (plus ground wire)
  - Asynchronous because no clock signal is transmit.
  - Relatively simple hardware.
- Very widely used

Table of contents

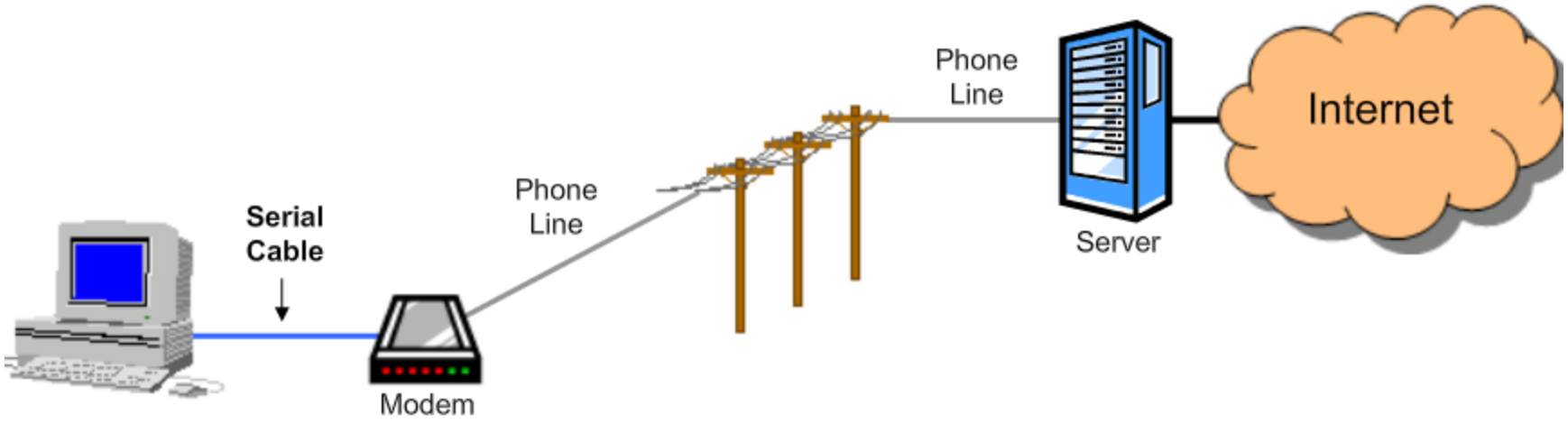
UART applications



PC to Device



PC to PC



PC to Internet

UART applications

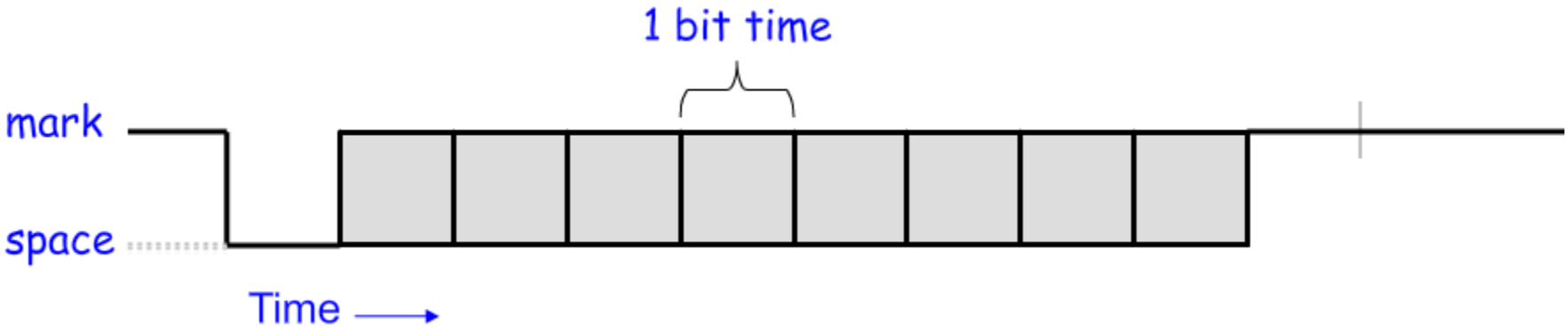
- Become much less common
- Largely been replaced by faster, more sophisticated interfaces
  - ✓ PCs: USB(peripherals), Internet(Network)
  - ✓ Chip to chip: I2C, SPI
- Still use today when simple low speed communication is needed: for debugger.

UART functions

- **Transmitter**
  - ✓Convert from parallel to serial
  - ✓Add Start and Stop bits
  - ✓Add parity bit
- **Receiver**
  - ✓Convert from serial to parallel
  - ✓Remove Start and Stop bits
  - ✓Check and remove parity bit

Table of contents

Data transmission

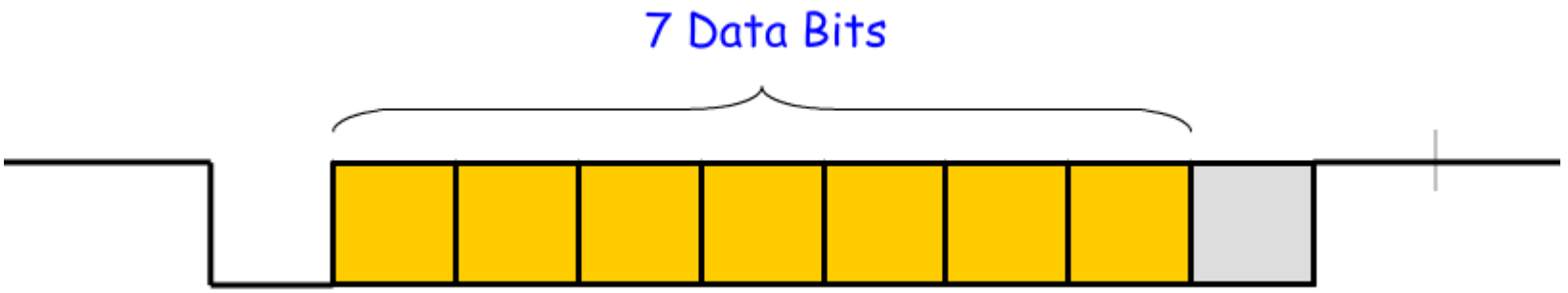


- Use a single wire for transmission
- Each bit has fixed time duration determined by the transmission rate.
  - ✓ Example: a 1200 bps (bit per second) UART will have a 1/1200 s or about 833.3 us bit duration.

Data transmission

- The Start bit marks the beginning of a new transmission.
- When detected, the receiver synchronizes with the new data stream.

Data transmission

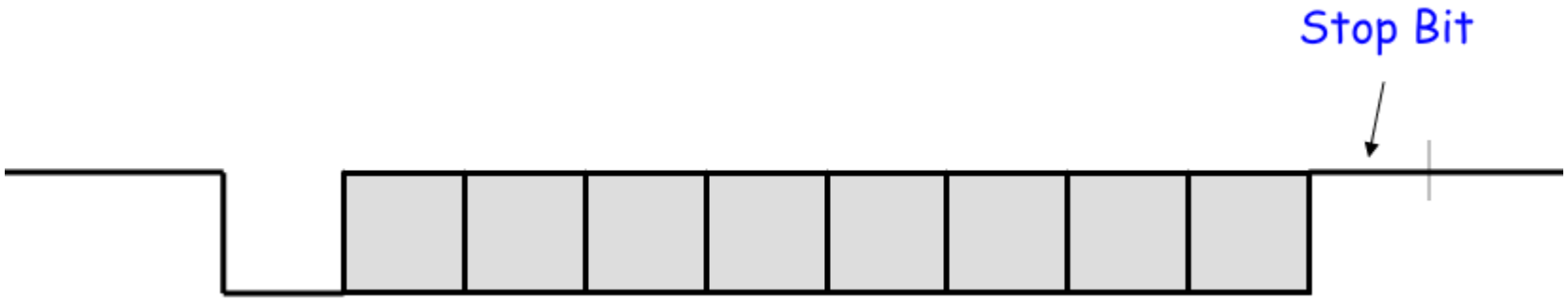


- Next follow the data bits (7 or 8).
- The least significant bit is sent first.

Data transmission

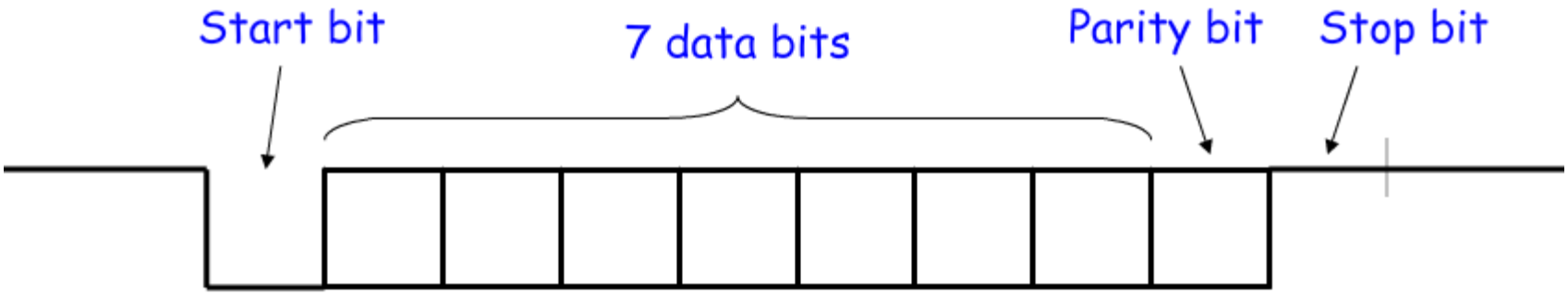
- The parity bit is added to make the number of 1's even (even parity) or odd (odd parity).
- This is can be used by the receiver to check for transmission errors.
- Use of parity bit is optional.

Data transmission



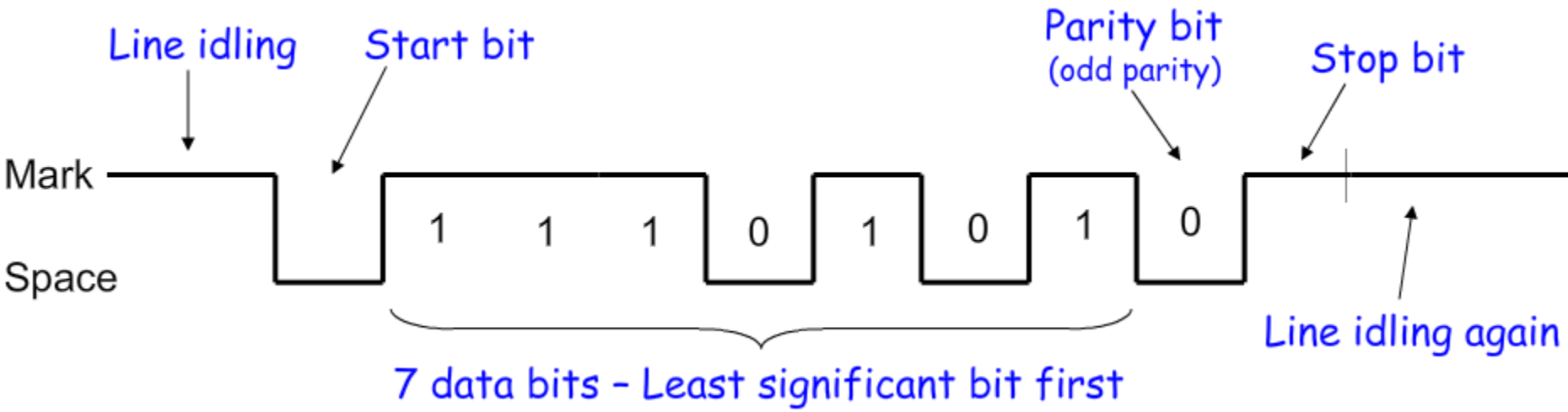
- The Stop bit marks the end of transmission.
- Receiver checks to make sure it is “1”.
- Separates one word from the start bit of the next word.

Data transmission



- In the configuration shown, it takes 10 bits to send 7bits of data.
- Transmission efficiency is 70%.

Data transmission

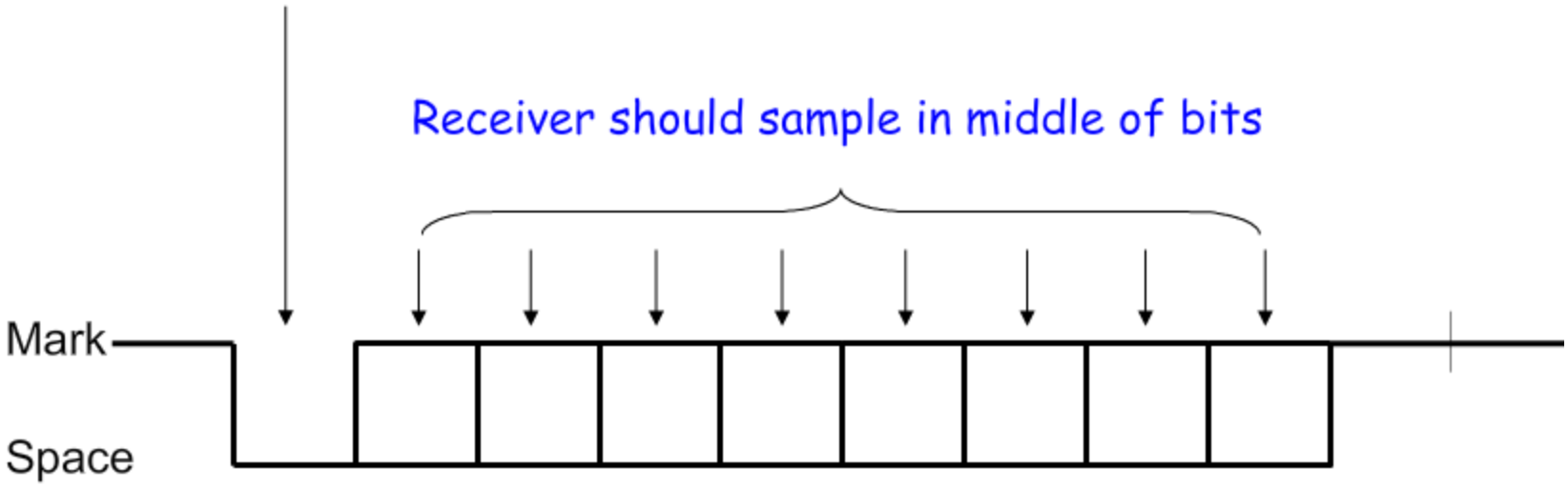


- An example of sending the ASCII letter “W”- 1010111.

Table of contents

Data Reception

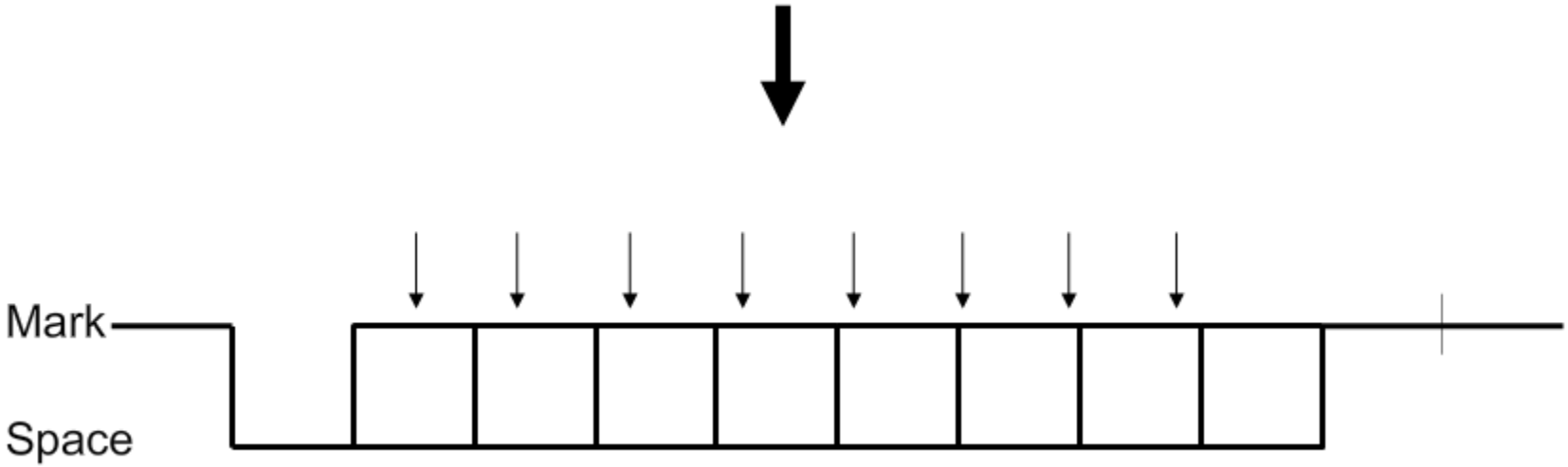
Start bit says a character is coming,
receiver resets its timers



Receiver uses a timer (counter) to time when it samples.
Transmission rate (i.e., bit duration) must be known!

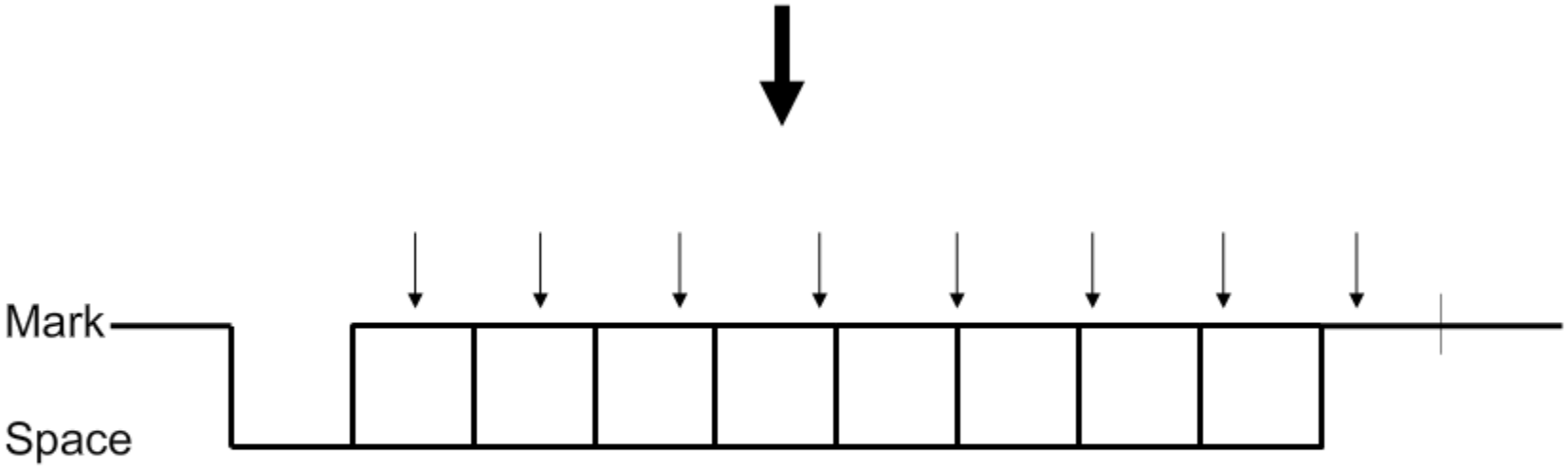
Data Reception

If receiver samples too quickly, see what happens…



Data Reception

If receiver samples too slowly, see what happens…



Receiver resynchronizes on every start bit.
Only has to be accurate enough to read 9 bits.

Data Reception

- Receiver also verifies that stop bit is “1”, if not, reports “framing error” to host system.
- New start bit can appear immediately after Stop bit.
- Receive will resynchronize for each Start bit.

UART options

- UARTs usually have programmable options:
  - ✓ Data: 7 or 8 bits.
  - ✓ Parity: even, odd, none, mark, space.
  - ✓ Stop bits: 1, 1.5, 2.
  - ✓ Baud rate: 300, 1200, 2400, 4800, 9600 , 19.2k, 38.4k, 57.6k. 115.2k…

UART Data throughput

- Data throughput Example
  - ✓Assume 19200 baud, 8 data bits, no parity, 1 stop bit
    - 19200 baud→19.2kbps
    - 1 start bit + 8 data bits + 1 stop bit = 10 bits → It takes 10 bits to send 8 bits of data.
    - 19.2 kbps * 8/10 = 15.36 kbps.
    - How many KB (kilobytes) per second:

=15.36 * 1KB / 8192 bits = 1.875 KB/s

Table of contents

RS232

- RS232 is the most common UART standard.
  - Used by PC serial ports.
- RS232 does not use positive logic.
  - Logic 1 is any signal from -25V to -3V
  - Logic 0 is any signal from +3V to 25V
  - The range -3V to 3V is a transition region that is not assigned to a logic level.
- RS232 connecting between a *DTE* (Data Terminal Equipment) and a *DCE* (Data Circuit-terminating Equipment).

RS232 signals

Signal			Origin		DB-25 pin
Name	Typical purpose	Abbreviation	DTE	DCE	
Data Terminal Ready	Indicates presence of DTE to DCE.	DTR	•		20
Data Carrier Detect	DCE is connected to the telephone line.	DCD		•	8
Data Set Ready	DCE is ready to receive commands or data.	DSR		•	6
Ring Indicator	DCE has detected an incoming ring signal on the telephone line.	RI		•	22
Request To Send	DTE requests the DCE prepare to receive data.	RTS	•		4
Clear To Send	Indicates DCE is ready to accept data.	CTS		•	5
Transmitted Data	Carries data from DTE to DCE.	TxD	•		2
Received Data	Carries data from DCE to DTE.	RxD		•	3
Common Ground		GND	common		7
Protective Ground		PG	common		1

- Hardware flow control = RTS/CTS handshaking

Table of contents

Freedom KL46 UART Block Diagram

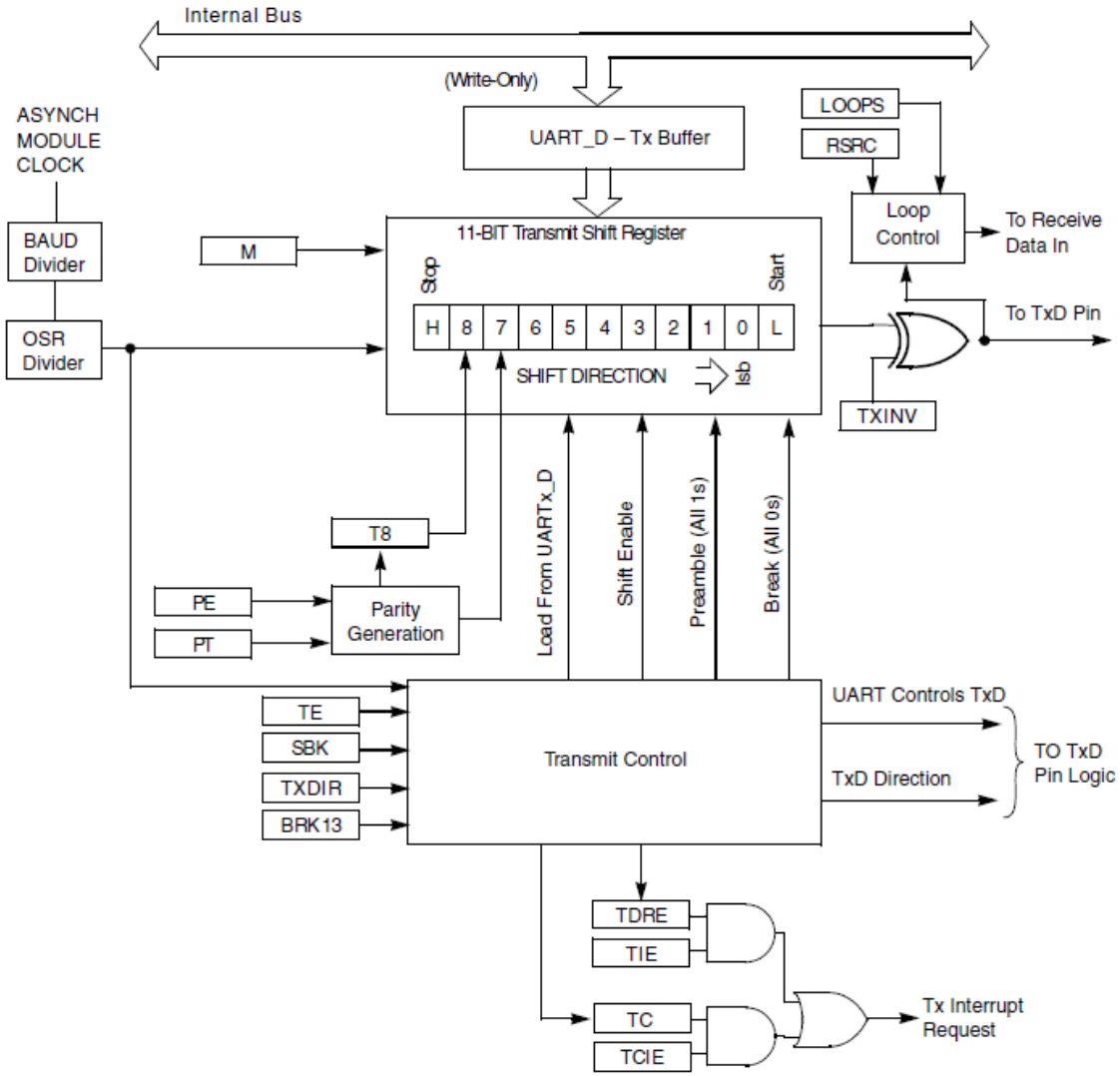


Figure 39-1. UART transmitter block diagram

Freedom KL46 UART Block Diagram

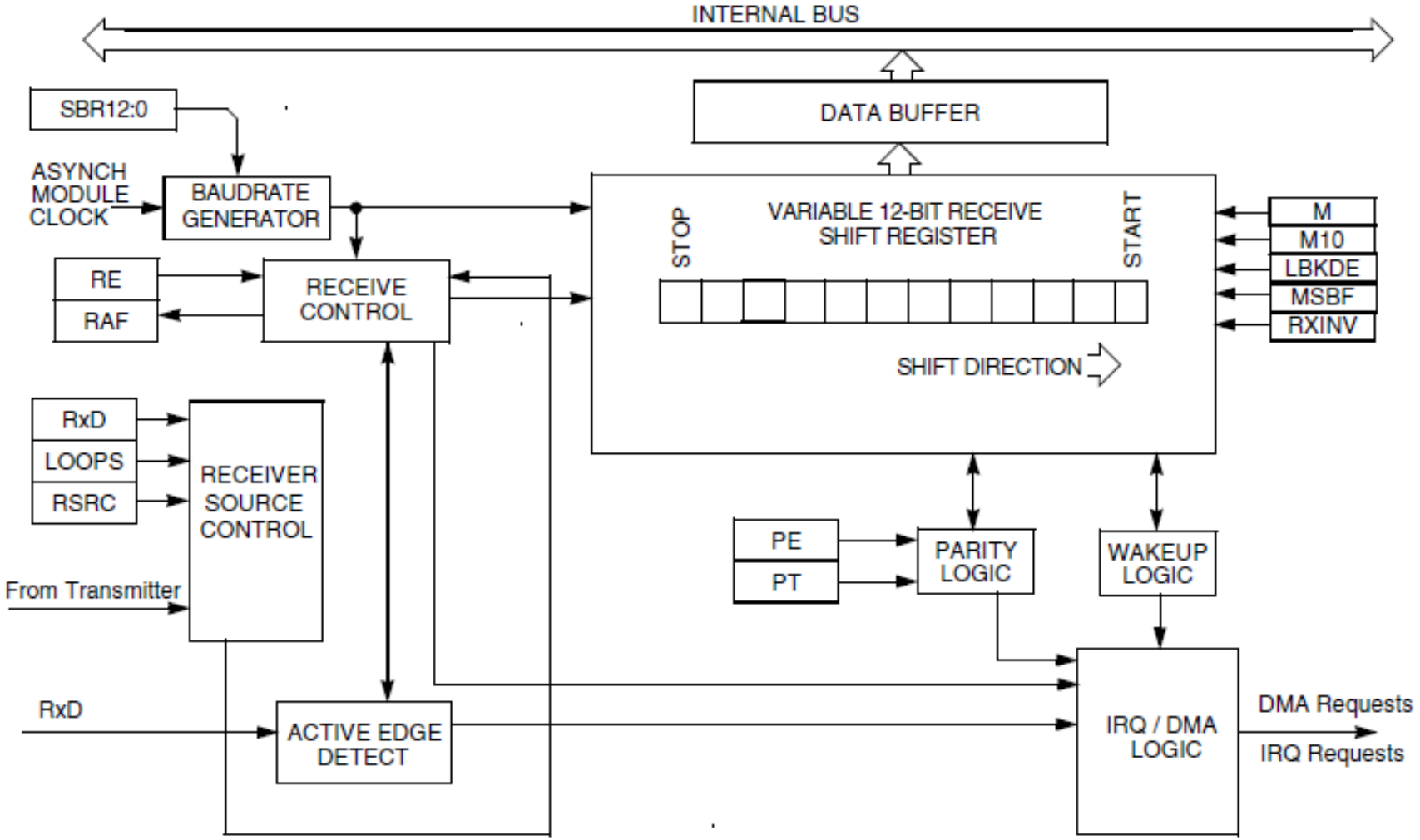


Figure 39-2. UART receiver block diagram

Features

- Full-duplex operation
- 13 bit baud rate selection.
- Programmable 8-bit, 9-bit or 10-bit data format.
- Double-buffered transmitter and receiver with separate enables
- Independent FIFO structure for transmit and receive.
- DMA interface.
- Three operation mode: Wait, Stop and Debug mode.

UART signals

Signal	Description	I/O
$\overline{\text{CTS}}$	Clear to send	I
$\overline{\text{RTS}}$	Request to send	O
RXD	Receive data	I
TXD	Transmit data	O
$\overline{\text{Collision}}$	Collision Detect	I

Baud Rate Registers

- **UARTx_BDH**
  - ✓ SBR[4-0]: 5 higher bits of the 13 bits UART baud rate bits
- **UARTx_BDL**
  - ✓ SBR[7-0]: 8 lower bits of the 13 bits UART baud rate bits

UART baud rate

$$= \frac{UART\ module\ clock}{16\ *\ (SBR[12:0] + BRFD)}$$

Where BRFD = the BRFA[4:0] field divided by 32

Table 57-351. Baud rates (example: module clock = 10.2 MHz)

Bits SBR (decimal)	Bits BRFA	BRFD value	Receiver clock (Hz)	Transmitter clock (Hz)	Target Baud rate	Error (%)
17	00000	0	600,000.0	37,500.0	38,400	2.3
16	10011	19/32=0.59375	614,689.3	38,418.08	38,400	0.047
33	00000	0	309,090.9	19,318.2	19,200	0.62
33	00110	6/32=0.1875	307,344.6	19,209.04	19,200	0.047
66	00000	0	154,545.5	9659.1	9600	0.62
133	00000	0	76,691.7	4793.2	4800	0.14
266	00000	0	38,345.9	2396.6	2400	0.14
531	00000	0	19,209.0	1200.6	1200	0.11
1062	00000	0	9604.5	600.3	600	0.05
2125	00000	0	4800.0	300.0	300	0.00
4250	00000	0	2400.0	150.0	150	0.00
5795	00000	0	1760.1	110.0	110	0.00

Control Registers

- UARTx_C1:
  - ✓ M: 9-bit or 8-bit select mode
  - ✓ PE: Parity Enable
  - ✓ PT: Parity Type (Even, Odd)
- UARTx_C2:
  - ✓ TIE: Transmit Interrupt or DMA Transfer Enable
  - ✓ TCIE: Transmission Complete Interrupt Enable
  - ✓ RIE: Receiver Full Interrupt or DMA Transfer Enable
  - ✓ TE: Transmitter Enable
  - ✓ RE: Receiver Enable

Status Register

- UARTx_S1
  - ✓TDRE: Transmit Data Register Empty Flag
  - ✓TC: Transmit Complete Flag.
  - ✓RDRF: Receiver Data Register Full Flag.
  - ✓FE: Framing Error Flag.
  - ✓PF: Parity Error Flag.
- UARTx_S2
  - ✓MSBF: Most Significant Bit First.
  - ✓RAF: Receiver Active Flag.

Data Register

- UARTx_D
  - RxTx[7:0] :
    - Reads return the contents of the read-only receive data register.
    - Writes go to the write-only transmit data register.

Example UART Init

1. Enable the clock for the port associated with the UART pins you want to use (SIM_SCGCx)
2. Enable UART pins (PORTx_PCRn)
3. Enable the UART module clock (SIM_SCGCn)
4. Configure the UART control registers for the desired data format
   - Number of data bits (UARTn_C1[M] and UARTn_C4[M10])
   - Parity and parity type (UARTn_C1[PE,PT])
   - MSB or LSB first (UARTn_S2[MSBF])
   - Data polarity (UARTn_S2[RXINV] and UARTn_C3[TXINV])
5. Configure the baud rate (UARTn_BDH and UARTn_BDL)
6. Enable the receiver and/or transmitter (UARTn_C2[RE, TE])

Reference

- http://www.powershow.com/view/af8e1-ZTk2Z/20_UART_flash_ppt_presentation
- http://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter
- KL46 Reference Manual

Summary

- Understanding about the basic concepts regarding UART such as its applications, how the data has been transmitted as well as received.
- Understanding about RS232 standard.
- Understand on how to configure the Freedom KL46 UART module.

Question & Answer

Thanks for your attention !

Copyright

- This course including **Lecture Presentations**, **Quiz**, **Mock Project**, **Syllabus**, **Assignments**, **Answers** are copyright by FPT Software Corporation.
- This course also uses some information from external sources and non-confidential training document from Freescale, those materials comply with the original source licenses.