The UART protocol

By: Yehia M. Abu Eita

Outlines

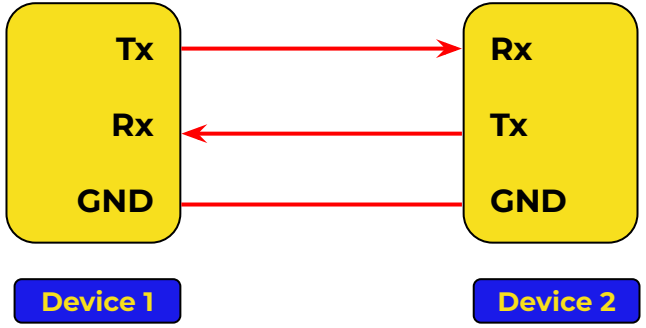
- What is UART?
- UART Flow Control Protocols
- UART frame format
- UART block diagram
- UART modes of operation
- ATmega32 UART registers
- ATmega32 UART baud rate calculation
- Steps to program ATmega32 UART

What is UART?

- It is one of the serial communications used in embedded systems.
- It is the acronym of **U**niversal **A**synchronous **R**eceive **T**ransmit.
- It is an **Asynchronous** communications system.
- It is a **full-duplex** Communications system.
- It is also **wired** communication system.
- **It can be used to interface with:**
  - **Terminal applications on PC (ex: Putty, Hercules, teraterm, etc.) using USB to TTL module**
  - **Bluetooth module**
  - **Wifi module**

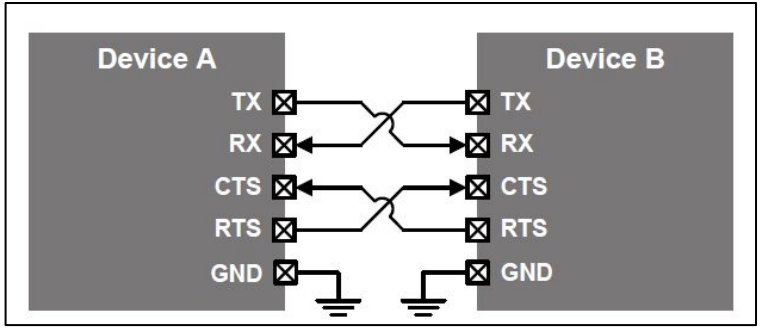
UART Flow Control Protocols

- **Software Flow Control**:
  - Only **3 wires** are required (**RX**, **TX**, and **GND**).
  - Transmission is started and stopped by **sending special flow control characters**.
  - The flow control characters are sent over the normal TX and RX line.



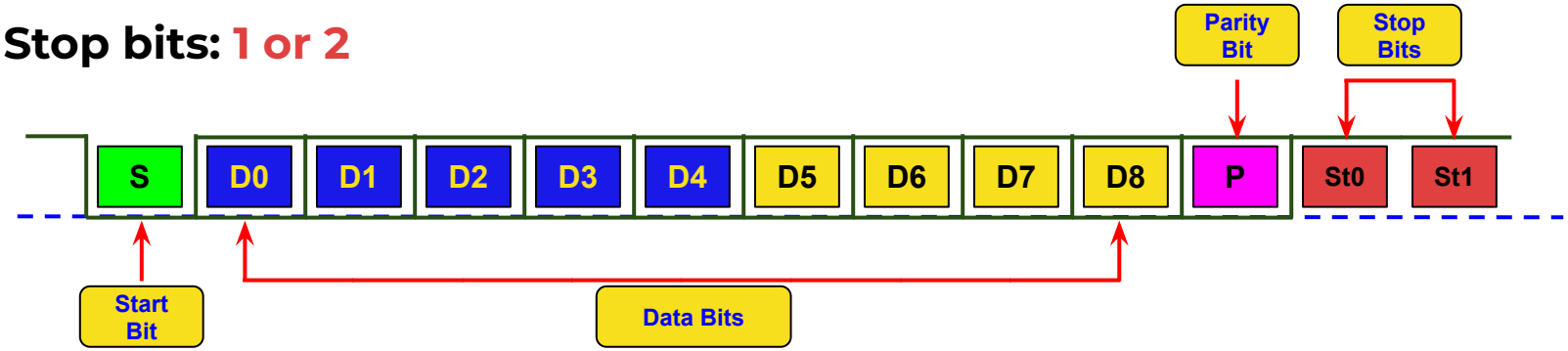
UART Flow Control Protocols

- **Hardware Flow Control**:
  - It is called **RTS/CTS** flow control.
  - **Two extra lines** needed in addition to the data lines:
    - **RTS: Request To send line.**
    - **CTS: Clear To Send line.**
  - Each device will use its **RTS to output if it is ready to accept new data** and **read CTS to see if it is allowed to send data** to the other device.



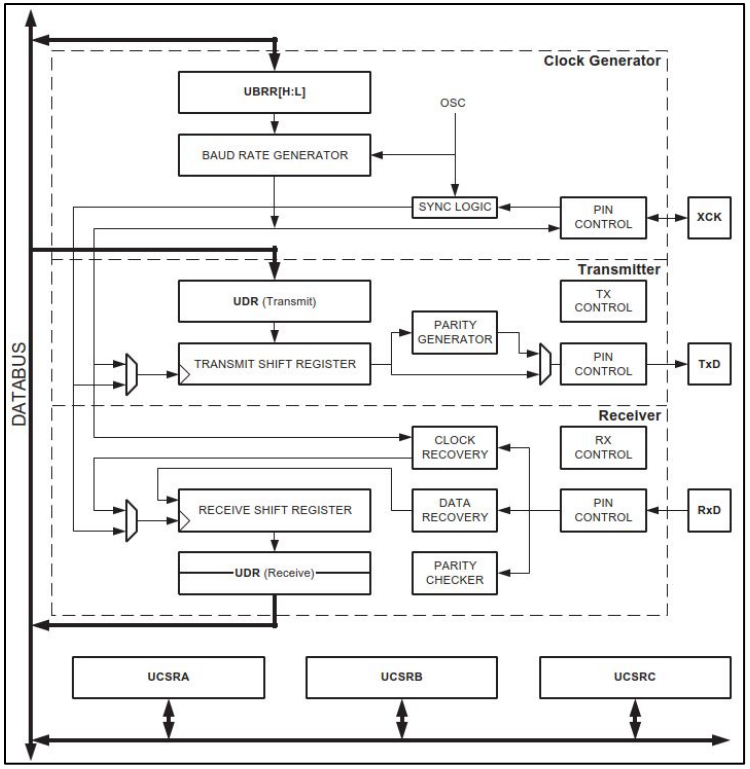
UART frame format

- **Start bits:** 1
- **Data bits:** from 5 to 9 data bits
- **Parity:** no parity, Even or Odd
- **Stop bits:** 1 or 2



UART block diagram

- Clock generator
- Transmitter
- Receiver



UART modes of operation

- **Asynchronous mode**
  - Using only Tx and Rx pins for operation.
  - It has normal frame format.
  - It has programmed standard baud rates.
- **Synchronous mode (USART)**
  - Using the clock pin in addition to Tx and Rx pins.
  - It has also normal frame formats.
  - It has programmed standard baud rates.
- **Multi-processor communications mode**
  - There will be UART master and Multi-slaves.
  - Each slave has address.
  - The frame is doubled in size, the first frame determines the address of the slave, and the second frame is the data.
  - Slaves can differentiate between address and data frame according to the first stop or to the ninth bit.

ATmega32 UART registers

UDR – USART I/O Data Register

Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDR (Read)
	TXB[7:0]								UDR (Write)
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

UCSRA – USART Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

UCSRB – USART Control and Status Register B

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

UCSRC – USART Control and Status Register C

Bit	7	6	5	4	3	2	1	0	
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	0	0	0	0	1	1	0	

UBRRL and UBRRH – USART Baud Rate Registers

Bit	15	14	13	12	11	10	9	8	
	URSEL	–	–	–	UBRR[11:8]				UBRRH
	UBRR[7:0]								UBRRL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

ATmega32 UART baud rate calculation

Operating Mode	Equation for Calculating Baud Rate[1]	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{OSC}}{16(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{OSC}}{8(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{OSC}}{2(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

Baud Rate (bps)	f_{osc} = 1.0000MHz				f_{osc} = 1.8432MHz				f_{osc} = 2.0000MHz			
	U2X = 0		U2X = 1		U2X = 0		U2X = 1		U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error
2400	25	0.2%	51	0.2%	47	0.0%	95	0.0%	51	0.2%	103	0.2%
4800	12	0.2%	25	0.2%	23	0.0%	47	0.0%	25	0.2%	51	0.2%
9600	6	-7.0%	12	0.2%	11	0.0%	23	0.0%	12	0.2%	25	0.2%
14.4k	3	8.5%	8	-3.5%	7	0.0%	15	0.0%	8	-3.5%	16	2.1%

Steps to program ATmega32 UART

- **Initializing:**
  - Selecting the **baud rate**, **UBRRH** and **UBRRL**
  - **Enable transmitting** or **receiving** or **both**, enable or disable **interrupts** , **UCSRB**
  - **Set the frame format**, number of data bits, choose parity, and number of stop bits, **UCSRB** and **UCSRBC**
  - Choose **normal or double speed**, **UCSRA**
- **Transmit data:**
  - **Wait** for **data buffer to be empty,** or use an ISR if interrupts are enabled, **UCSRA**
  - **Write** data into the **data register**, **UDR**
- **Receive data:**
  - **Wait** for **reception complete**, or use an ISR if interrupts are enabled, **UCSRA**
  - **Read** the data from the **data register**, **UDR**

Summary

- The UART protocol is the simplest protocol you can use
- It is asynchronous and full-duplex protocol
- Transmitter and receiver must have the same baud rates
- UART can be connected to a USB to TTL module to connect the microcontroller to a computer and sending information about the application during run-time