COMP2121: Microprocessors and Interfacing

Serial Input/Output

http://www.cse.unsw.edu.au/~cs2121

Lecturer: Hui Wu

Term 2, 2019

1

Overview

- The Components of Asynchronous Communication Systems
- Standards for the Serial I/O Interface
- RS232-C and Other Standards
- USART (Universal Synchronous and Asynchronous serial Receiver and Transmitter) in AVR

2

Why Serial I/O?

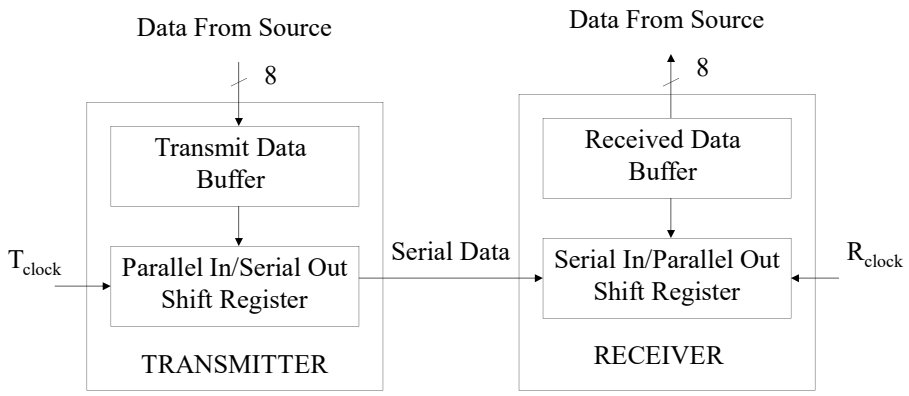
Problems with Parallel I/O:

- Needs a wire for each bit.
- When the source and destination are more than a few feet the parallel cable can be bulky and expensive.
- Susceptible to reflections and induced noise for long distance communication.

Serial I/O provides a solution to these problems.

3

The Components of a Asynchronous Communication System (1/3)



4

The Components of a Asynchronous Communication System (2/3)

At the communication source:

- The parallel interface transfers data from the source to the transmit data buffer.
- These data are shifted into the parallel in/serial out shift register and T_{clock} shifts the data out to the receiver.

5

The Components of a Asynchronous Communication System (3/3)

At the communication destination:

- R_{clock} shifts each bit received into the serial in/parallel out shift register.
- After all data bits have been shifted, they are transferred to the received data buffer.
- The data in the received data buffer are transferred to the input operation via the parallel interface.

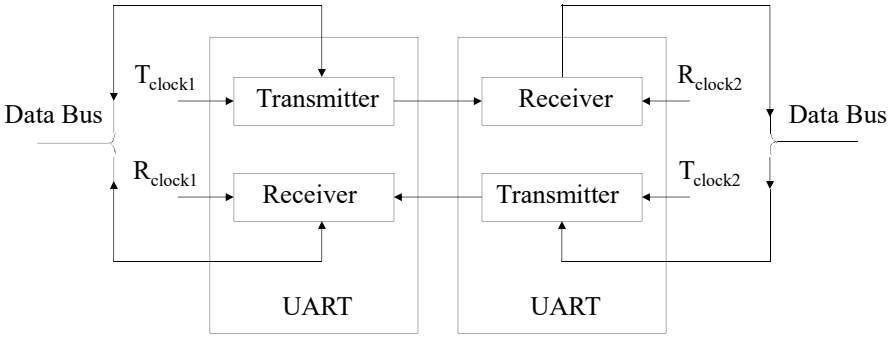
6

UART (1/2)

- The device that implements both transmitter and receiver in a single integrated circuit is called a UART (Universal Asynchronous Receiver/Transmitter).
- UART is the basis for most serial communication hardware.
- Details of UART will be covered later.

7

UART (2/2)



8

Design Considerations of the Serial Communication System

- How are data to be encoded?
- If the data are sent in serial, which bit is set first?
- How the receiver synchronised with the transmitter?
- What is the data rate?
- How are the electrical signals for logic values defined?
- How does the system provide for handshaking?

9

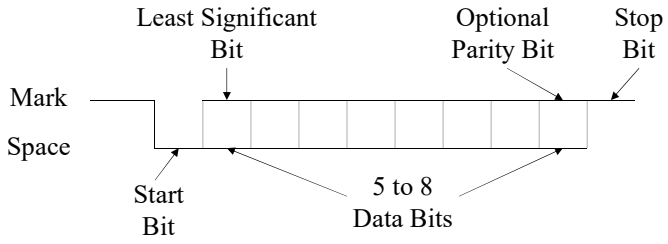
Data Encoding and Transmission (1/3)

- Several codes used for the alphanumeric information.
- The most common is ASCII (American Standard Code for Information Interchange), using 7 bits to encode 96 printable characters and 32 control characters.
- Two choices for the order of data transmission: least significant bit first or most significant bit first.
- USRT uses least significant bit first order.
- Data are transmitted asynchronously. Therefore, synchronisation between transmitter and receiver is required.
- UART provides a way to synchronise the receiver shift register with the transmitter shift register.

10

Data Encoding and Transmission (2/3)

- Data are encapsulated in two other bits called start bit and stop bit.
- Mark and space: the logic one and zero levels are called mark and space.
  - When the transmitter is not sending anything, it holds the line at mark level, also called idle level.



11

Data Encoding and Transmission (3/3)

- Typical bits in data transmission:
  - Start bit: When the transmitter has data to send, it first changes the line from the mark to the space level for one bit time. This synchronises the receiver with transmitter. When the receiver detects the start bit, it knows to start clocking in the serial data bits.
  - Data bits: Almost any number of data bits can be sent between the start and stop bits. Typically, between 5 and 8 bits are sued.
  - Parity bit: The parity bit, used to detect errors in the data, is added to the data to make the total number of ones odd (odd parity) or even (even parity).
  - Stop bit: The stop bit is added at the end of data bits. It gives one bit-time between successive characters. Some systems require more than one stop bit.

12

Data Transmission Rate

- The rate at which bits are transmitted is called baud rate.
- It is given in bits per second.
- Standard data rates – Baud:
  110, 150, 300, 600, 900, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57800

13

Standards for the Serial I/O Interface (1/2)

Interface standards are needed to allow different manufacturers' equipment to be interconnected and must define the following elements:

- Handshaking signals.
- Direction of signal flow.
- Types of communication devices.
- Connectors and interface mechanical considerations.
- Electrical signal levels.

14

Standards for the Serial I/O Interface (2/2)

The existing standards include RS-232-C, RS-422, RS-423 and RS-485.

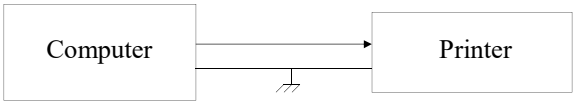
- RS-232-C standard is used in most serial interface.
- If the signals must be transmitted farther than 50 feet or greater than 20 Kbits/second, another electrical interface standard such as RS-422, RS-423 or RS-485 should be chosen.
- For RS-422, RS-423 and RS-485, handshaking, direction of signal flow, and the types of communication devices are based on the RS-232-C standard.

15

Communication System Types (1/3)

Three ways that the data can be sent in serial communication system:

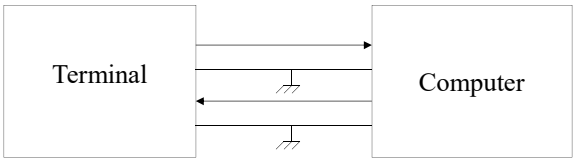
- Simplex system:
  - ❑ Data are sent in one direction only, say, to a serial printer.
  - ❑ If the computer does not send data faster than the printer can accept it, no handshaking signals are required.
  - ❑ Two signal wires are needed for this system.



16

Communication System Types (2/3)

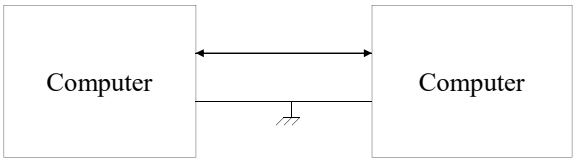
- Full-duplex (FDX) system:
  - ❑ Data are transmitted in two directions.
  - ❑ It is called four-wire system, although only two signal wires and a common ground are sufficient.



17

Communication System Types (3/3)

- Half-duplex (HDX) system:
  - ❑ Data are transmitted in two directions with only one pair of signal lines.
  - ❑ Additional hardware and handshaking signals must be added to an HDX system.



18

Half-Duplex Handshaking Signals (1/2)

Figure 1 shows a half-duplex system with additional interface circuitry and handshaking signals defined for the RS-232-C interface standard.

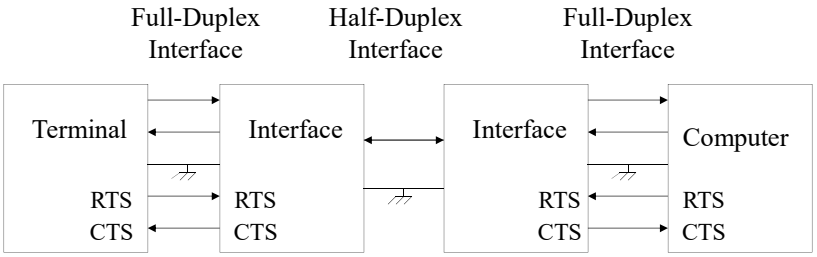


Figure 1 Half-duplex system with handshaking

19

Half-Duplex Handshaking Signals (2/2)

- The interface blocks have three roles:
  - They give a full-duplex channel between themselves and the terminal or computer.
  - They decide whether they or their opposite interface is sending or receiving data
  - They use and control the request to send (RTS) and clear to send (CTS) handshaking signals.
    - The RTS signal is asserted by the terminal or computer when data are to be sent.
    - When the interface finds that the other system is not sending data, it asserts CTS signal. The sending station must wait until it is clear to send before transmitting.
- Half duplex systems are not often used these days, although the RTS/CTS handshaking signals have been retained to control the flow of data.

20

Data Terminal Equipment and Data Communication Equipment (1/2)

- The blocks labelled "Interface" in Figure 1 are, in practice, modems, and the two-wire half-duplex line is a telephone line.
- Modems are called data communication equipment (DCE).
- The terminals or the computers to which modems are attached to are called data terminal equipment (DTE).
- A modem is a modulator/demodulator.
  - It converts logic levels into tones to be sent over a telephone line.
  - At the other end of the telephone line, a demodulator converts the tones back to logic levels.
  - In a half-duplex system, a single set of tones are defined, one for space and one for mark.

21

Data Terminal Equipment and Data Communication Equipment (2/2)

- Half-duplex modems are no longer used because modem have been developed to allow full-duplex data transmission over a telephone line.
- A full-duplex system has two types of modems, called originate and answer modems, and two sets of tones.

Originate and answer modem tone definitions for Bell 212A

Originate modem	**Direction**	**Answer modem**
Modulator tones		Demodulator
1070 Hz — Space	→	
1270 Hz — Mark		
Demodulator		Modulator tones
	←	2025 Hz — Space
		2225 Hz — Mark

22

Modem Handshaking Signals (1/3)

- Ring Indicator (RI)
  - The telephone company transmits a special tone that rings the phone. The modem can detect this and asserts the RI signal.
  - The terminal or computer can use RI to start some special process such as notifying the user that the other end is calling or to answer the telephone in an answer modem.
- Data Set Ready (DSR)
  - This signal tells the DTE that the modem (also called data set) has established a connection over the telephone line to the far end.

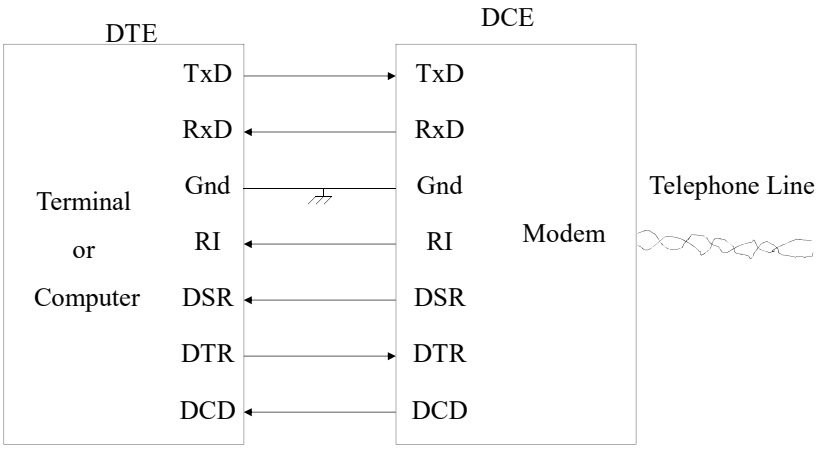
23

Modem Handshaking Signals (2/3)

- Data Terminal Ready (DTR)
  - This signal comes from the DTE and inform the modem that it is ready to operate.
  - This is usually just an indication that the power is turned on in the terminal but could be controlled by a computer.
  - An intelligent answer modem can use it to answer a call automatically only when the computer or terminal is ready.
- Data Carrier Detect (DCD)
  - DCD is asserted when the carrier, or tone defined for a mark, is being generated by the modem on the other end.
  - It was originally used in half-duplex systems.
    - When one end wanted to transmit, it first asserted the RTS line. The modem then checked the DCD bit. If it found it asserted, it knew the other end was sending. When DCD was deasserted, CTS was asserted, allowing transmission from the requesting terminal.

24

Modem Handshaking Signals (3/3)



25

RS-232-C Signal Definitions (1/3)

DE9	DB25	Signal	Purpose
	1	PG	Protective ground: this is actually the shield in a shielded cable. It is designed to be connected to the equipment frame and may be connected to external grounds.
3	2	TxD	Transmitted data: Sourced by DTE and received by DCE. Data terminal equipment cannot send unless RTS, CTS, DSR and DTR are asserted.
2	3	RxD	Received data: Received by DTE, sourced by DCE.
7	4	RTS	Request to send: Sourced by DTE, received by DCE. RTS is asserted by the DTE when it wants to send data. The DCE responds by asserting CTS.

26

RS-232-C Signal Definitions (2/3)

DE9	DB25	Signal	Purpose
8	5	CTS	Clear to send: Sourced by DCE, received by DTE. CTS must be asserted before the DTE can transmit data.
6	6	DSR	Data set ready: Sourced by DCE and received by DTE. Indicates that the DCE has made a connection on the telephone line and is ready to receive data from the terminal. The DTE must see this asserted before it can transmit data.
5	7	SG	Signal ground: Ground reference for this signal is separate from pin 1, protective ground.

27

RS-232-C Signal Definitions (3/3)

DE9	DB25	Signal	Purpose
1	8	DCD	Data carrier detect: Sourced by DCE, received by DTE. Indicates that a DCE has detected the carrier on the telephone line. Originally it was used in half-duplex systems but can be used in full-duplex systems, too.
4	20	DTR	Data terminal ready: Sourced by DTE and received by DCE. Indicates that DTE is ready to send or receive data.
9	22	RI	Ring indicator: Sourced by DCE and received by DTE. Indicates that a ringing signal is detected.

28

RS-232-C Interconnections (1/5)

- When two serial ports are connected, the data rate, the number of data bits, whether parity is used, the type of parity, and the number of stop bits must be set properly and identically on each UART.
- Proper cables must be used. There are three kinds of cables from which to choose, depending on the types of devices to be interconnected.
  - ❑ The full DTE – DCE cable.
  - ❑ The DTE – DTE null modem cable.
  - ❑ The minimal DTE – DCE cable.

29

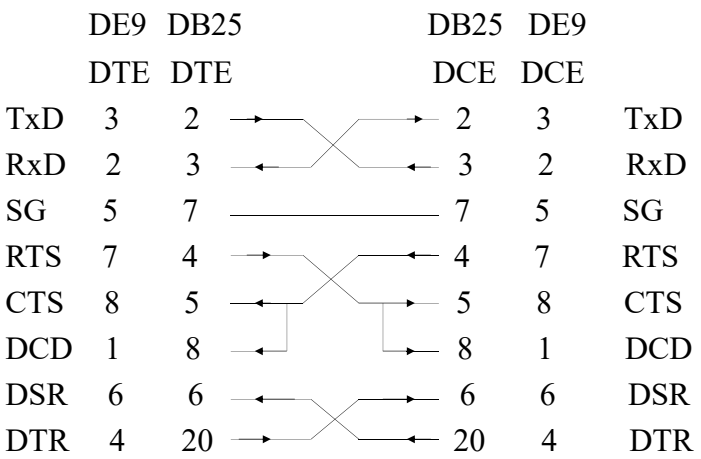
RS-232-C Interconnections (2/5)

	DE9 DTE	DB25 DTE		DB25 DCE	DE9 DCE	
TxD	3	2	→	2	3	TxD
RxD	2	3	←	3	2	RxD
SG	5	7		7	5	SG
RTS	7	4	→	4	7	RTS
CTS	8	5	←	5	8	CTS
DCD	1	8	←	8	1	DCD
DSR	6	6	←	6	6	DSR
DTR	4	20	→	20	4	DTR

Full DTE – DCE cable

30

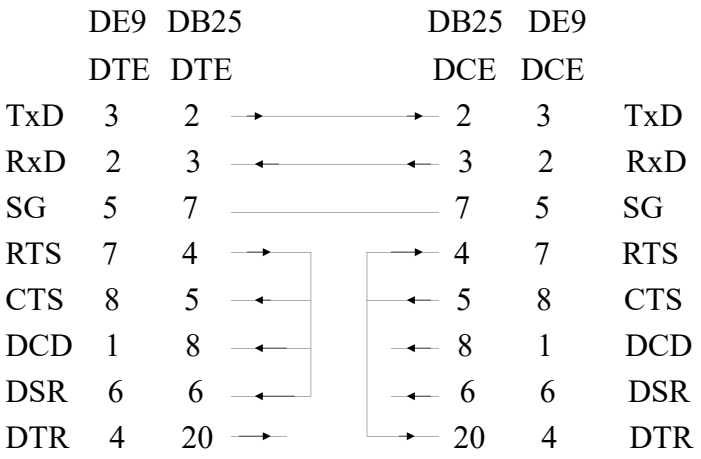
RS-232-C Interconnections (3/5)



DTE – DTE null modem cable

31

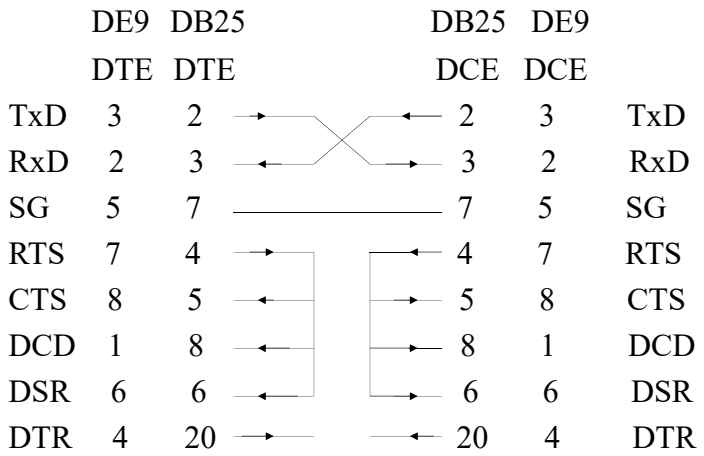
RS-232-C Interconnections (4/5)



Minimal three-wire cable

32

RS-232-C Interconnections (5/5)

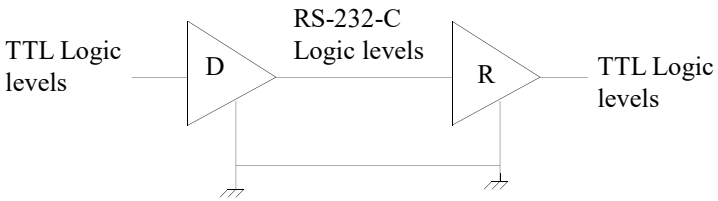


Minimal null modem cable

33

RS-232-C Interface

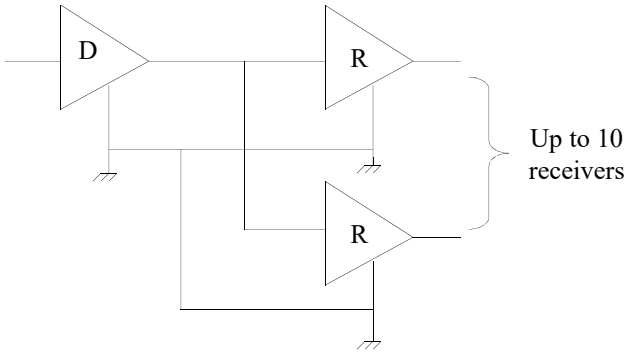
RS-232-C Logic levels:	
Mark	-25 to –3 volts
Space	+25 to +3 volts



34

RS-423 Standard

- Also a single ended system.
- Allows longer distance and higher data rates than RS-232-C.
- Allows a driver to broadcast data to 10 receivers.

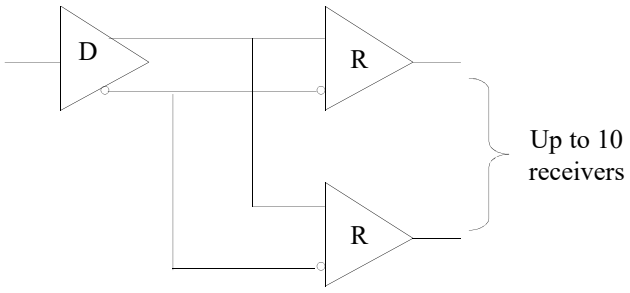


RS-423 Interface

35

RS-422 Standard

- RS-422 line drivers and receivers operates with differential amplifier.
  - ❑ These drivers eliminate much of the common-mode noise experienced with long transmission lines, thus allowing the longer distances and higher data rates.

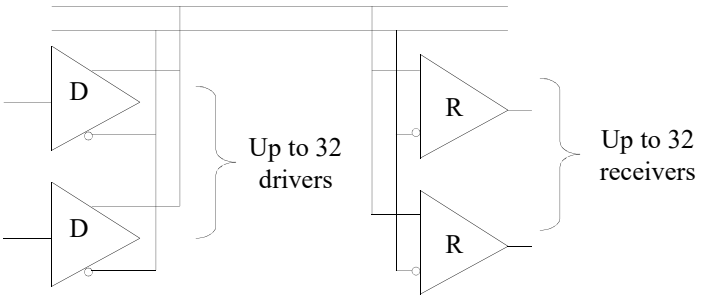


RS-422 Interface

36

RS-485 Standard

- Similar to RS-422 in that it uses differential line drivers and receivers.
- Unlike RS-422, RS-485 provides for multiple drivers and receivers in a bussed environment.
  - ❑ Up to 32 drivers/receivers pairs can be used together.



RS-485 Interface

37

Line Lengths and Data Rates

RS-423 line length and data rate

Line length (ft)	Data rate
30	100 Kbits/s
300	10 Kbits/s
4000	1 Kbits/s

RS-422 line length and data rate

Line length (ft)	Data rate
40	10 Mbits/s
400	1 Mbits/s
4000	100 Kbits/s

RS-485 line length and data rate

Line length (ft)	Data rate
40	10 Mbits/s
400	1 Mbits/
4000	100 Kbits/s

38

Summary of Standards

Specification	RS-232-C	RS-423	RS-422	RS-485
Receiver input voltage	±3 to ±15V	±200mV to ±12V	±200mV to ±7V	±200mV to -7 to +12V
Driver output signal	±5 to ±15V	±3.6 to ±6V	±2 to ±5V	±1.5 to ±5V
Maximum data rate	20 Kb/s	100 Kb/s	10 Mb/s	10 Mb/s
Maximum cable length	50 ft	4000 ft	4000 ft	4000 ft
Driver source Impedance	3-7 KΩ	450 Ω min	100 Ω	54Ω
Receiver input resistance	3 KΩ	4 KΩ min	4 K Ω min	12 KΩ minimum
Mode	Singled-ended	Singled-ended	Differential	Differential
Number of drivers and receivers allowed on one line	1 Driver 1 Receivers	1 driver 10 Receivers	1 Driver 10 Receivers	32 Driver 32 Receivers

39

Main Features of USART in AVR (1/2)

- Full duplex operation (independent serial receive and transmit registers).
- Asynchronous or synchronous operation.
- Master or slave clocked synchronous operation.
- High resolution baud rate generator.
- Supports serial frames with 5, 6, 7, 8, or 9 data bits and 1 or 2 stop bits.
- Odd or even parity generation and parity check supported by hardware.

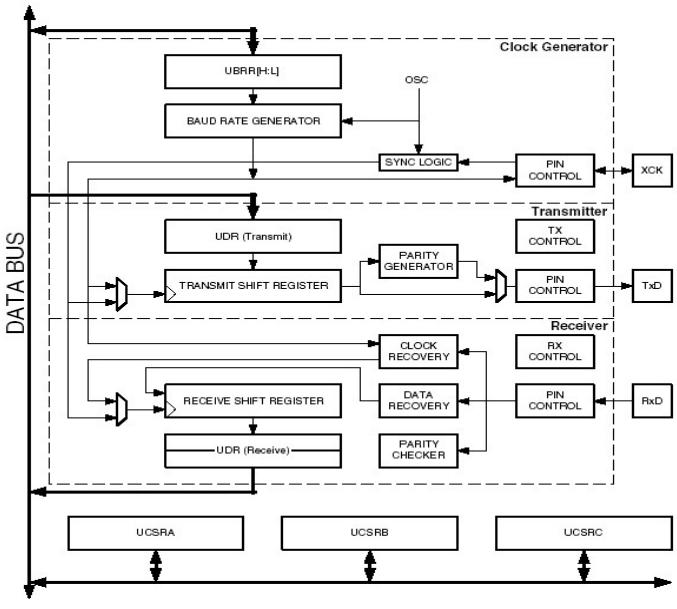
40

Main Features of USART in AVR (2/2)

- Framing error detection.
- Noise filtering includes false start bit detection and digital low pass filter.
- Three separate interrupts on TX Complete, TX Data Register Empty and RX Complete.
- Multi-processor communication mode.
- Double speed asynchronous communication mode.

41

The Block Diagram of USART



42

The Major Components of USART (1/2)

Three main components:

- Clock generator
  - The Clock Generation logic consists of synchronization logic for external clock input used by synchronous slave operation, and the baud rate generator.
- Transmitter
  - The Transmitter consists of a single write buffer, a serial Shift Register, Parity Generator and Control Logic for handling different serial frame formats.
    - The write buffer allows a continuous transfer of data without any delay between frames.

43

The Major Components of USART (2/2)

- Receiver
  - The Receiver is the most complex part of the USART module due to its clock and data recovery units.
  - The recovery units are used for asynchronous data reception.
  - In addition to the recovery units, the Receiver includes a Parity Checker, Control Logic, a Shift Register and a Two Level Receive Buffer (UDR).
  - The Receiver supports the same frame formats as the Transmitter, and can detect Frame Error, Data OverRun and Parity Errors.

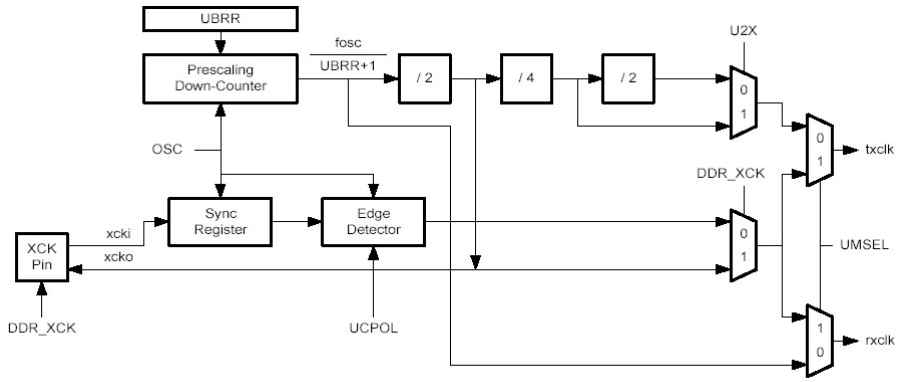
44

Clock Generation (1/3)

- The Clock Generation logic generates the base clock for the Transmitter and Receiver.
- The USART supports four modes of clock operation: Normal asynchronous, Double Speed asynchronous, Master synchronous and Slave synchronous mode.
  - The UMSEL bit in USART Control and Status Register C (UCSRC) selects between asynchronous and synchronous operation.
  - Double Speed (asynchronous mode only) is controlled by the U2X found in the UCSRB Register.
- When using synchronous mode (UMSEL = 1), the Data Direction Register for the XCK pin (DDR_XCK) controls whether the clock source is internal (Master mode) or external (Slave mode).
- The XCK pin is only active when using synchronous mode.

45

Clock Generation (2/3)



46

Clock Generation (3/3)

Signal description:

txclk: Transmitter clock (internal signal).

rxclk: Receiver base clock (internal signal).

xcki: Input from XCK pin (internal signal). Used for synchronous slave operation.

xcko: Clock output to XCK pin (internal signal). Used for synchronous master operation.

fosc: XTAL pin frequency (system clock).

47

The Baud Rate Generator

- The USART Baud Rate Register (UBRR) and the down-counter connected to it function as a programmable prescaler or baud rate generator.
- The down-counter, running at system clock (fOSC), is loaded with the UBRR value each time the counter has counted down to zero or when the UBRRL Register is written.
- A clock is generated each time the counter reaches zero.
  - This clock is the baud rate generator clock output (=fOSC/(UBRR+1)).
- The transmitter divides the baud rate generator clock output by 2, 8, or 16 depending on mode.
- The baud rate generator output is used directly by the receiver's clock and data recovery units. However, the recovery units use a state machine that uses 2, 8 or 16 states depending on mode set by the state of the UMSEL, U2X and DDR_XCK bits.

48

Frame Formats (1/3)

- A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error checking.
- The USART accepts all 30 combinations of the following as valid frame formats:
  - 1 start bit
  - 5, 6, 7, 8, or 9 data bits
  - no, even or odd parity bit
  - 1 or 2 stop bits

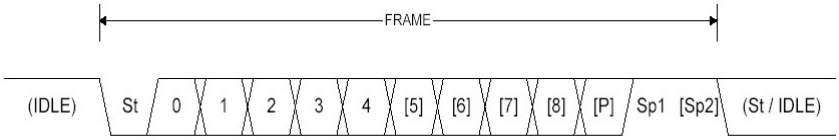
49

Frame Formats (2/3)

- A frame starts with the start bit followed by the least significant data bit. Then the next data bits, up to a total of nine, are succeeding, ending with the most significant bit.
- If enabled, the parity bit is inserted after the data bits, before the stop bits. When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle (high) state.

50

Frame Formats (3/3)



St Start bit, always low.

(n) Data bits (0 to 8).

P Parity bit. Can be odd or even.

Sp Stop bit, always high.

IDLE No transfers on the communication line (RxD or TxD). An IDLE line must be high.

51

Parity Bit Calculation

- The parity bit is calculated by doing an exclusive-or of all the data bits. If odd parity is used, the result of the exclusive or is inverted. The relation between the parity bit and data bits is as follows:

$$P_{even} = d_n \oplus d_{n-1} \oplus \ldots \oplus d1 \oplus d0 \oplus 0$$
$$P_{odd} = d_n \oplus d_{n-1} \oplus \ldots \oplus d1 \oplus d0 \oplus 1$$

Where P_{even} Parity bit using even parity

P_{odd} Parity bit using odd parity

d_n Data bit n of the character

- If used, the parity bit is located between the last data bit and first stop bit of a serial frame.

52

USART Initialisation (1/3)

- USART has to be initialised before any communication can take place.
- The initialisation process normally consists of setting the baud rate, setting frame format and enabling the Transmitter or the Receiver depending on the usage.
- For interrupt driven USART operation, the Global Interrupt Flag should be cleared when doing the initialisation.

53

USART Initialisation (2/3)

- Before doing a re-initialisation with changed baud rate or frame format, be sure that there are no ongoing transmissions during the period the registers are changed.
  - The TXC flag can be used to check that the Transmitter has completed all transfers, and the RXC flag can be used to check that there are no unread data in the receive buffer. Note that the TXC flag must be cleared before each transmission (before UDR is written) if it is used for this purpose.

54

USART Initialisation (3/3)

Assembly Code Example:

```
USART_Init:
; Set baud rate
    out UBRRH, r17
    out UBRRL, r16
; Enable receiver and transmitter
    ldi r16, (1<<RXEN)|(1<<TXEN)
    out UCSRB,r16
; Set frame format: 8data, 2stop bit
    ldi r16, (1<<USBS)|(3<<UCSZ0)
    out UCSRC,r16
    ret
```

55

The USART Transmitter (1/3)

- The USART Transmitter is enabled by setting the Transmit Enable (TXEN) bit in the UCSRB Register.
- When the Transmitter is enabled, the normal port operation of the TxD pin is overridden by the USART and given the function as the transmitter's serial output.
- The baud rate, mode of operation and frame format must be set up once before doing any transmissions. If synchronous operation is used, the clock on the XCK pin will be overridden and used as transmission clock.

56

The USART Transmitter (2/3)

- A data transmission is initiated by loading the transmit buffer with the data to be transmitted.
- The CPU can load the transmit buffer by writing to the UDR I/O location.
- The buffered data in the transmit buffer will be moved to the Shift Register when the Shift Register is ready to send a new frame.
- The Shift Register is loaded with new data if it is in idle state (no ongoing transmission) or immediately after the last stop bit of the previous frame is transmitted.
- When the Shift Register is loaded with new data, it will transfer one complete frame at the rate given by the baud register, U2X bit or by XCK depending on mode of operation.

57

The USART Transmitter (3/3)

Assembly Code Example:

```
USART_Transmit:
; Wait for empty transmit buffer
     sbis UCSRA,UDRE
     rjmp USART_Transmit
; Put data (r16) into buffer, sends the data
     out UDR,r16
     ret
```

58

Transmitter Flags and Interrupts (1/3)

- The USART Transmitter has two flags that indicate its state: USART Data Register Empty (UDRE) and Transmit Complete (TXC). Both flags can be used for generating interrupts.
- The Data Register Empty (UDRE) flag indicates whether the transmit buffer is ready to receive new data.
  - This bit is set when the transmit buffer is empty, and cleared when the transmit buffer contains data to be transmitted that has not yet been moved into the Shift Register. For compatibility with future devices, always write this bit to zero when writing the UCSRA Register.

59

Transmitter Flags and Interrupts (2/3)

- When the Data Register empty Interrupt Enable (UDRIE) bit in UCSRB is written to one, the USART Data Register Empty Interrupt will be executed as long as UDRE is set (provided that global interrupts are enabled). UDRE is cleared by writing UDR.
- When interrupt-driven data transmission is used, the Data Register Empty Interrupt routine must either write new data to UDR in order to clear UDRE or disable the Data Register Empty Interrupt, otherwise a new interrupt will occur once the interrupt routine terminates.
- The Data Register Empty (UDRE) flag indicates whether the transmit buffer is ready to receive new data.

60

Transmitter Flags and Interrupts (3/3)

- The Transmit Complete (TXC) flag bit is set to one when the entire frame in the Transmit Shift Register has been shifted out and there are no new data currently present in the transmit buffer.
- The TXC flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location.
  - The TXC flag is useful in half-duplex communication interfaces (like the RS-485 standard), where a transmitting application must enter Receive mode and free the communication bus immediately after completing the transmission.

61

The USART Receiver (1/3)

- The USART Receiver is enabled by writing the Receive Enable (RXEN) bit in the UCSRB Register to one.
- When the Receiver is enabled, the normal pin operation of the RxD pin is overridden by the USART and given the function as the receiver's serial input.
- The baud rate, mode of operation and frame format must be set up once before any serial reception can be done.
- If synchronous operation is used, the clock on the XCK pin will be used as transfer clock.

62

The USART Receiver (2/3)

- The Receiver starts data reception when it detects a valid start bit.
- Each bit that follows the start bit will be sampled at the baud rate or XCK clock, and shifted into the Receive Shift Register until the first stop bit of a frame is received.
- A second stop bit will be ignored by the Receiver.
- When the first stop bit is received, i.e., a complete serial frame is present in the Receive Shift Register, the contents of the Shift Register will be moved into the receive buffer. The receive buffer can then be read by reading the UDR I/O location.

63

The USART Receiver (3/3)

Assembly Code Example:

```
USART_Receive:
; Wait for data to be received
sbis UCSRA, RXC
rjmp USART_Receive
; Get and return received data from buffer
in r16, UDR
ret
```

64

Receive Compete Flag and Interrupt

- The Receive Complete (RXC) flag indicates if there are unread data present in the receive buffer.
  - ❑ This flag is one when unread data exist in the receive buffer, and zero when the receive buffer is empty (i.e. does not contain any unread data).
  - ❑ If the receiver is disabled (RXEN = 0), the receive buffer will be flushed and consequently the RXC bit will become zero.
- When the Receive Complete Interrupt Enable (RXCIE) in UCSRB is set, the USART Receive Complete Interrupt will be executed as long as the RXC flag is set (provided that global interrupts are enabled).
- When interrupt-driven data reception is used, the receive complete routine must read the received data from UDR in order to clear the RXC flag; otherwise a new interrupt will occur once the interrupt routine terminates.

65

Receiver Error Flags (1/2)

The USART Receiver has three error flags: Frame Error (FE), Data OverRun (DOR) and USART Parity Error (UPE).

- The Frame Error (FE) flag indicates the state of the first stop bit of the next readable frame stored in the receive buffer.
  - ❑ The FE flag is zero when the stop bit was correctly read (as one), and the FE flag will be one when the stop bit was incorrect (zero).
  - ❑ This flag can be used for detecting out-of-sync conditions, detecting break conditions and protocol handling.

66

Receiver Error Flags (2/2)

- The Data OverRun (DOR) flag indicates data loss due to a receiver buffer full condition.
  - A Data OverRun occurs when the receive buffer is full (two characters), it is a new character waiting in the Receive Shift Register, and a new start bit is detected.
  - If the DOR flag is set there was one or more serial frame lost between the frame last read from UDR, and the next frame read from UDR.
- The USART Parity Error (UPE) flag indicates that the next frame in the receive buffer had a Parity Error when received.

67

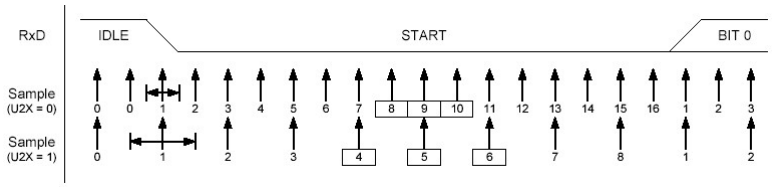
Asynchronous Data Reception

- The USART includes a clock recovery and a data recovery unit for handling asynchronous data reception.
- The clock recovery logic is used for synchronizing the internally generated baud rate clock to the incoming asynchronous serial frames at the RxD pin.
- The data recovery logic samples and low pass filters each incoming bit, thereby improving the noise immunity of the Receiver.
- The asynchronous reception operational range depends on the accuracy of the internal baud rate clock, the rate of the incoming frames, and the frame size in number of bits.

68

Asynchronous Clock Recovery (1/3)

- The Clock Recovery logic synchronizes internal clock to the incoming serial frames.
- The following figure illustrates the sampling process of the start bit of an incoming frame.
- The sample rate is 16 times the baud rate for Normal mode, and eight times the baud rate for Double Speed mode.



69

Asynchronous Clock Recovery (2/3)

- The horizontal arrows illustrate the synchronization variation due to the sampling process. Note the larger time variation when using the Double Speed mode (U2X = 1) of operation.
- Samples denoted by zero are samples done when the RxD line is idle (i.e., no communication activity).
- When the Clock Recovery logic detects a high (idle) to low (start) transition on the RxD line, the start bit detection sequence is initiated.

70

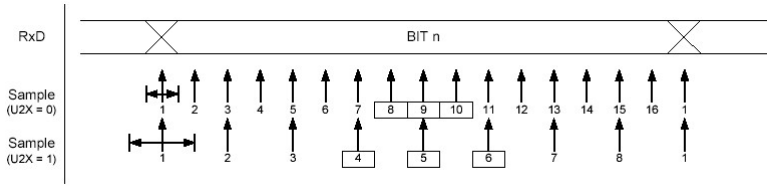
Asynchronous Clock Recovery (3/3)

- Let sample 1 denote the first zero-sample as shown in the figure. The Clock Recovery logic then uses samples 8, 9 and 10 for Normal mode, and samples 4, 5 and 6 for Double Speed mode (indicated with sample numbers inside boxes on the figure), to decide if a valid start bit is received.
- If two or more of these three samples have logical high levels (the majority wins), the start bit is rejected as a noise spike and the Receiver starts looking for the next high to low-transition.
- If however, a valid start bit is detected, the clock recovery logic is synchronized and the data recovery can begin.
- The synchronization process is repeated for each start bit.

71

Asynchronous Data Recovery (1/2)

- When the receiver clock is synchronized to the start bit, the data recovery can begin.
- The data recovery unit uses a state machine that has 16 states for each bit in Normal mode and eight states for each bit in Double Speed mode. The following figure shows the sampling of the data bits and the parity bit. Each of the samples is given a number that is equal to the state of the recovery unit.



72

Asynchronous Data Recovery (2/2)

- The decision of the logic level of the received bit is taken by doing a majority voting of the logic value to the three samples in the centre of the received bit.
  - The centre samples are emphasized on the figure by having the sample number inside boxes.
- The majority voting process is done as follows:
  - If two or all three samples have high levels, the received bit is registered to be a logic 1.
  - If two or all three samples have low levels, the received bit is registered to be a logic 0.
- This majority voting process acts as a low pass filter for the incoming signal on the RxD pin.
- The recovery process is then repeated until a complete frame is received.

73

Reading

1. Chapter 10: Serial Input/Output. Microcontrollers and Microcomputers by Fredrick M. Cady.
2. http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html.
3. http://www.beyondlogic.org/serial/serial.htm.
4. USART. Mega64 Data Sheet.

74