سرفصل های دوره

- **آشنایی با میکروکنترلرهای** AVR
- **مروری بر اصول زبان برنامه نویسی** C **مقدماتی**
- **کار با واحد** I/O (Input/Output)
- **آشنایی با واحد وقفه** (Interrupt)
- **آشنایی کامل با واحد تایمر و کانتر** (Timer/Counter)
- **آشنایی با مبدل آنالوگ به دیجیتال** ((Analog Digital Conversion (ADC
- **آشنایی با واحد ارتباط سریال آسنکرون** ((Universal Asynchronous Serial Receiver and Transmitter (UART
- **راه اندازی واحد** SPI (Serial Peripheral Interface)
- **انجام چندین پروژه عملی و مثال کاربردی برای هر بخش به صورت شبیه سازی و پیاده سازی سخت افزاری**

❖ ارتباط سریال آسنکرون (Universal Asynchronous Serial Receiver and Transmitter (UART))

- دو روش انتقال اطلاعات
  - انتقال به روش موازی (برای ارسال n بیت اطلاعات به n خطِ انتقال نیاز است)
  - انتقال به روش سریال (در هر لحظه فقط یک بیت انتقال می یابد)

- سرعت انتقال اطلاعات در روش موازی بیشتر است اما برای فواصل طولانی از نظر هزینه مقرون به صرفه نیست. همچنین برای انتقال اطلاعات در کانال مخابراتی بی سیم باید از روش سریال استفاده کرد.

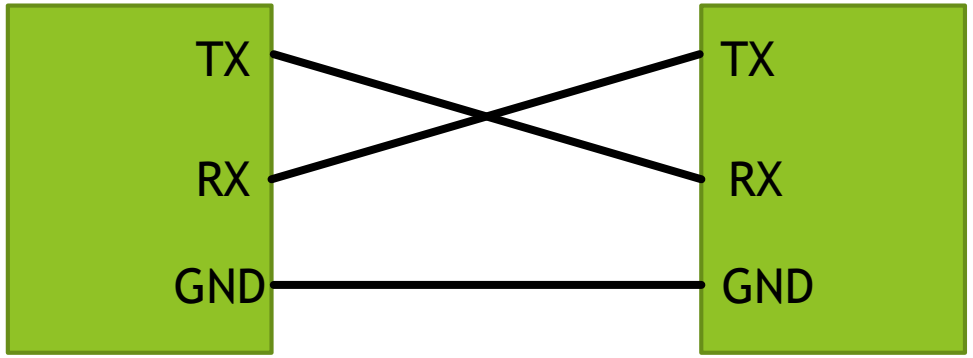
- بنابراین ارتباطات سریال یکی از بخش های بسیار مهم در ارتباط یک میکروکنترلر با دیگر قطعات و دستگاه ها می باشد.

- انواع ارتباطات سریال از نظر همزمانی
  - سنکرون یا همزمان (علاوه بر خط دیتا به کلاک هم نیاز دارد)
  - آسنکرون یا غیر همزمان(به کلاک نیاز ندارد)

- **انواع ارتباطات سریال از نظر جهت انتقال اطلاعات**
  - **یک طرفه یا ساده** (Simplex)
  - **نیمه دو طرفه** (Half Duplex)
  - **دو طرفه** (Full Duplex)
- **انواع ارتباطات سریال از نظر سطح سیگنال**
  - **معمولی**
  - **تفاضلی**
- **برخی از ارتباطات سریال**
  - USART/UART
  - RS232/RS485/RS422
  - USB
  - SPI
  - I2C
  - CAN
  - Ethernet
  - I2S
  - **و غیره**

❖ **ویژگی های ارتباط سریال** UART

- **یکی از ارتباطات سریال مهم ارتباط سریال آسنکرون** (UART) **می باشد.**
- **این ارتباط از دو مسیر** TX **برای ارسال اطلاعات و** RX **برای دریافت اطلاعات تشکیل می شود که یک ارتباط تمام دو طرفه محسوب می شود.**
- **برای یکسان بودن سطح سیگنال دو طرف، باید طرفین ارتباط،** GND **مشترک داشته باشند تا مسیر انتقال کامل شود.**
- **این نوع ارتباط می تواند سنکرون (همراه با کلاک) نیز باشد که در این صورت** USART **نامیده می شود.**

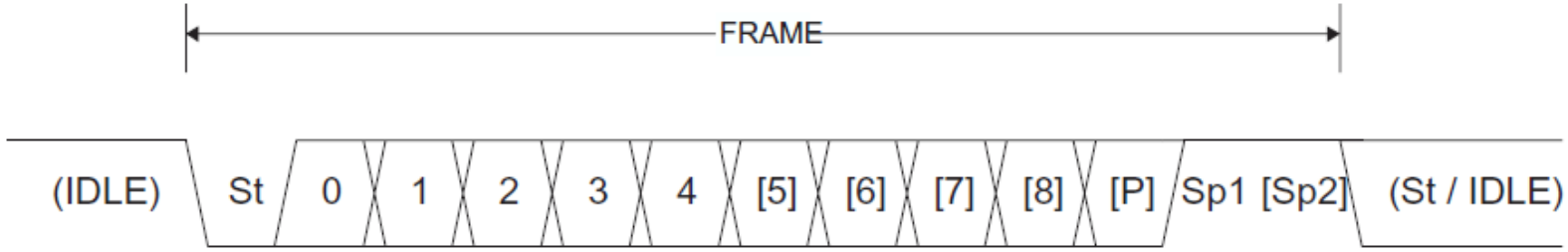


❖ **نرخ ارسال اطلاعات (Baud Rate)**

- **این ارتباط به دلیل عدم وجود کلاک، با تکنیک زمان بندی، اطلاعات را ارسال و دریافت می کند. این بازه زمانی، Baud Rate یا نرخ ارسال نام دارد که بر حسب تعداد بیت در یک ثانیه می باشد و کاربر آن را مشخص می کند.**

❖ **نحوه فریم بندی اطلاعات در UART**

Figure 72. Frame Formats



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.

❖ **بیت توازن** (Parity)

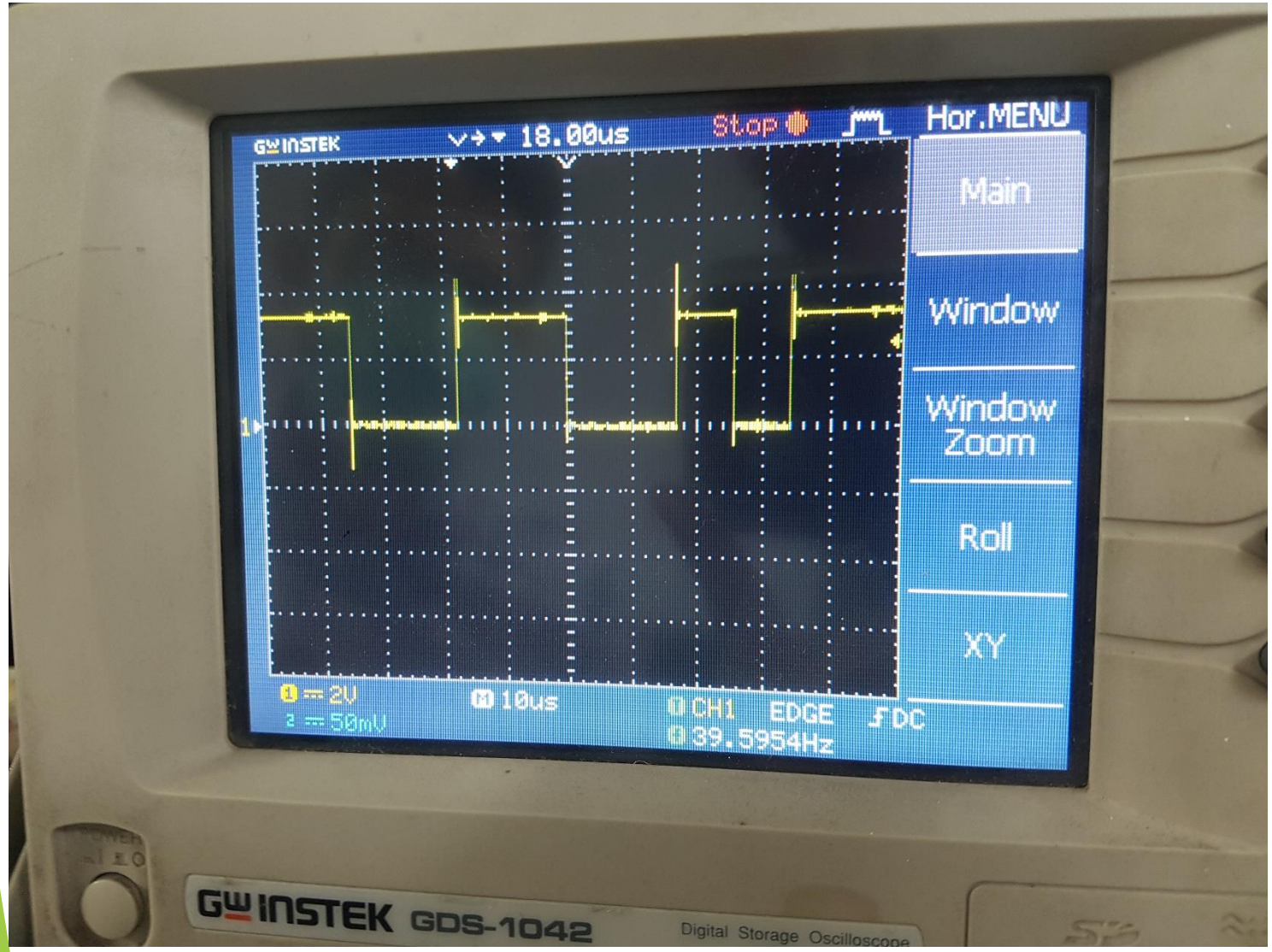
- **استفاده از بیت توازن در هنگام ارسال و دریافت اطلاعات به منظور تشخیص خطا انجام می شود که می تواند به صورت پریتی فرد یا زوج تنظیم شود و حداکثر قابلیت دارد وقوع خطا را در ۵۰ درصد مواقع تشخیص دهد.**

$$P_{even} = d_{n-1} \oplus \ldots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 0$$
$$P_{odd} = d_{n-1} \oplus \ldots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 1$$

❖ **UART در میکروکنترلرهای AVR**

- **میکروکنترلرهای AVR بسته به نوع های مختلف آن، داری یک یا چند واحد USART/UART با ویژگـــی هـــای مشترک زیرمی باشند:**

- **عملکرد کاملاً دو طرفه**
- **عملکرد سریال به صورت سنکرون و آسنکرون**
- **پشتیبانی از قالب های داده ۵ تا ۹ بیتی و یک یا دو بیت توقف**
- **مولد بیت توازن و تستر سخت افزاری آن**
- **آشکارسازی خطای فریم**
- **دارای سه وقفه ی مجزا برای ارسال و دریافت**



- برای مثـــال اگرتعـــداد بیتهـــای داده ۸ بیـــت و بیت توقف یـــک بیـــت، بدون پریتی تنظیم شود و داده ی 0xA6 را بخواهیم ارســـال کنـــیم شکل سیگنال ارسالی به صورت روبرو می شود:

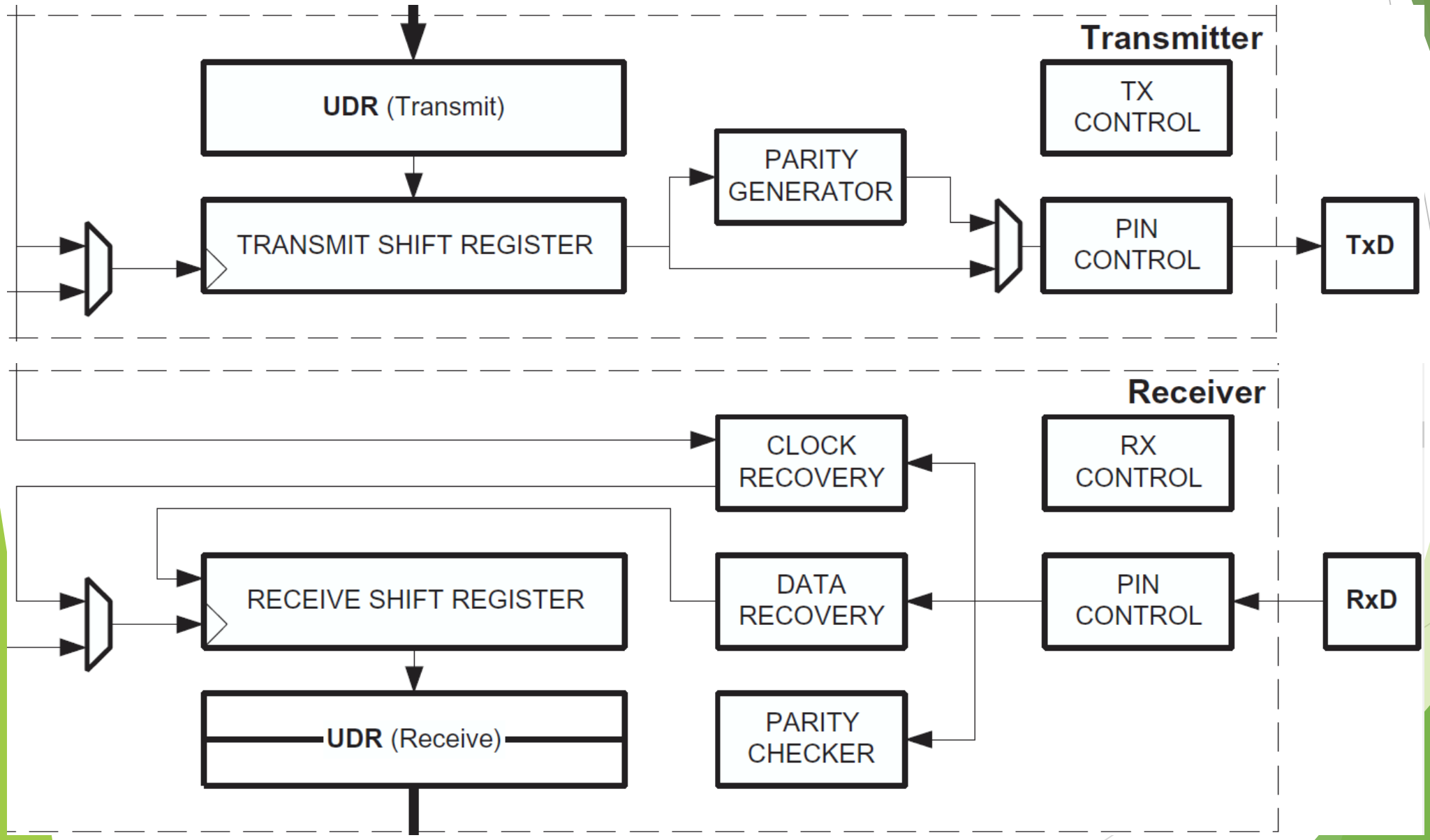
❖ آشنایی با رجیسترهای واحد UART

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	

- **بافر ارسال داده** (TXB) **و بافر دریافت داده** (RXB) **که در آدرس های جداگانه ای در حافظه تعریف شده اند با نام مشترک رجیستر** UDR **قابل دسترسی هستند.**

- **همان طور که در شکل بعدی که نحوه ی ارسال داده توسط بلوک فرستنده را نشان می دهد، مشـــخص اســـت، برای ارسال داده از یک شیفت رجیستر** PISO **استفاده شده است.**

- **همچنین برای دریافت داده از یک شیفت رجیستر** SIPO **استفاده شده است. عملکرد بلوک گیرنده نیـــز در شـــکل بعدی مشخص است.**


Transmitter
UDR (Transmit)
TX CONTROL
PARITY GENERATOR
TRANSMIT SHIFT REGISTER
PIN CONTROL
TxD
Receiver
CLOCK RECOVERY
RX CONTROL
RECEIVE SHIFT REGISTER
DATA RECOVERY
PIN CONTROL
RxD
UDR (Receive)
PARITY CHECKER

❖ رجیسترهای کنترل و وضعیت

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	

- **Bit 7 - RXC: USART Receive Complete**
- **Bit 6 – TXC: USART Transmit Complete**
- **Bit 5 – UDRE: USART Data Register Empty**
- **Bit 4 – FE: Frame Error**
- **Bit 3 – DOR: Data OverRun**
- **Bit 2 – PE: Parity Error**
- **Bit 1 – U2X: Double the USART Transmission Speed**

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	

- **Bit 7 – RXCIE: RX Complete Interrupt Enable**
- **Bit 6 – TXCIE: TX Complete Interrupt Enable**
- **Bit 5 – UDRIE: USART Data Register Empty Interrupt Enable**
- **Bit 4 – RXEN: Receiver Enable**
- **Bit 3 – TXEN: Transmitter Enable**
- **Bit 2 – UCSZ2: Character Size**
- **Bit 1 – RXB8: Receive Data Bit 8**
- **Bit 0 – TXB8: Transmit Data Bit 8**

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	

- **Bit 7 – URSEL: Register Select**
- **Bit 6 – UMSEL: USART Mode Select**
- **Bit 5:4 – UPM1:0: Parity Mode**

Table 63. UMSEL Bit Settings

UMSEL	Mode
0	Asynchronous Operation
1	Synchronous Operation

Table 64. UPM Bits Settings

UPM1	UPM0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

- **Bit 3 – USBS: Stop Bit Select**

Table 65. USBS Bit Settings

USBS	Stop Bit(s)
0	1-bit
1	2-bit

- **Bit 2:1 – UCSZ1:0: Character Size**

Table 66. UCSZ Bits Settings

UCSZ2	UCSZ1	UCSZ0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-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	

Table 60. Equations for Calculating Baud Rate Register Setting

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	$BAUD = \frac{f_{OSC}}{2(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

Note: 1. The baud rate is defined to be the transfer rate in bit per second (bps).

BAUD Baud rate (in bits per second, bps)

f_{OSC} System Oscillator clock frequency

UBRR Contents of the UBRRH and UBRRL Registers, (0 - 4095)

- **مثال۱: برنامه بنویسید که واحد** UART **را در حالت** Polling **(بدون وقفه) راه اندازی کند و یک کـــاراکتر از حـــروف کوچک را دریافت کند و حرف بزرگ متناظر با آن را نمایش دهد.**

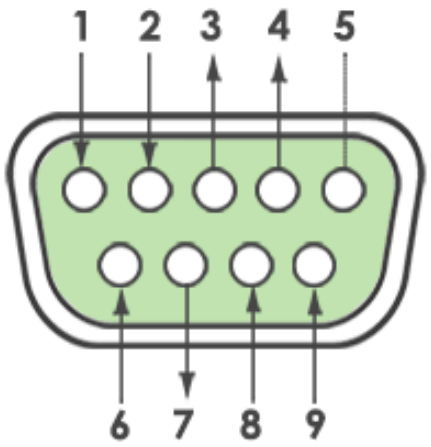
❖ **استاندارد RS232 (Recommended Standard 232)**

- **استانداردهای پیشنهادی بر پایه ی پروتکل** UART **هستند که برای ارتباط میکروکنترلرها با سایر دستگاه ها، تجهیزات و کامپیوترها تدوین شده است.**

- **استاندارد** RS232
  - **کانکتور** DB9
  - **سطح سیگنال** 12- و 12+
  - **پروتکل** UART

- **برای ارتباط** RS232 **از پایه های شماره ۲، ۳ و ۵ کانکتور** DB9 **مطابق اسلاید بعدی استفاده می شود.**

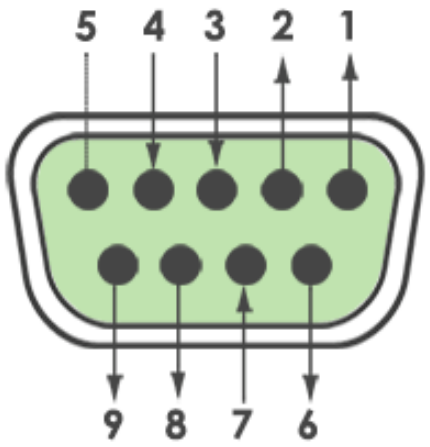
DB9 Male



Pin	Signal Direction	Signal Name	Signal Function
1	→	CD	Carrier Detect
2	←	RxD	Receive Data
3	→	TxD	Transmit Data
4	→	DTR	Data Terminal Ready
5	–	GND	Ground
6	←	DSR	Data Set Ready
7	←	RTS	Request To Send
8	→	CTS	Clear To Send
9	→	RI	Ring Indicator

→ Transmitted from DTE Device
← Received by DTE Device

DB9 Female

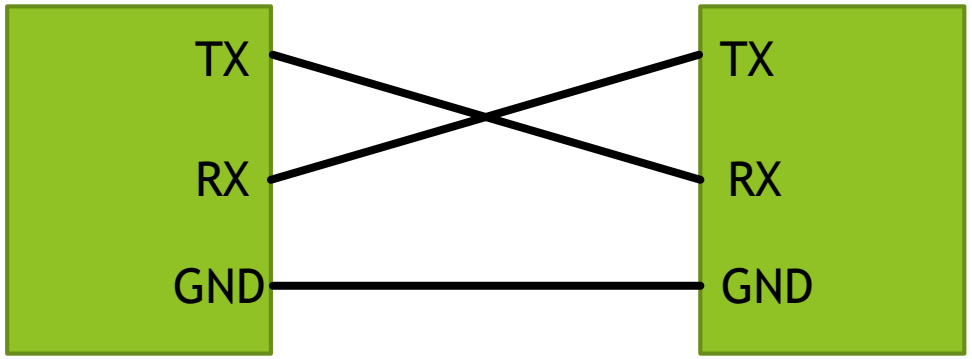


Pin	Signal Direction	Signal Name	Signal Function
1	←	CD	Carrier Detect
2	→	TxD	Transmit Data
3	←	RxD	Receive Data
4	→	DTR	Data Terminal Ready
5	–	GND	Ground
6	←	DSR	Data Set Ready
7	→	CTS	Clear To Send
8	←	RTS	Request To Send
9	←	RI	Ring Indicator

→ Transmitted from DCE Device
← Received by DCE Device

Codrey Electronics

- **سطوح سیگنال در ارتباط** RS232
- **در ارتباط سریال** UART **در صورتی دو طرف ارتباط به صورت مستقیم به هم متصل می شوند که هر دو، ســطوح منطقــی یکسانی داشته باشند.**



میکرو۲ (سطح TTL) **میکرو۱ (سطح** TTL)

- **اگر طرفین ارتباط سریال** UART **سطوح منطقی یکسانی نداشته باشند مثلاً یک طرف میکرو با سطح** TTL **و طــرف دیگــر کامپیوتر با سطح** RS232 **باشد، باید از مبدل های سطح ولتاژ استفاده کنیم. برای مثال می توان به دو نمونه از این مبدل ها اشاره کرد.** MAX232 **و** MAX3232 0 → +12 1 → -12
- **برای لپ تاپ یا سایر دستگاه هایی که دارای درگاه سریال نمی باشند تراشه هایی برای تبدیل درگاه** USB **بــه ســریال در سطح** TTL **وجود دارد که از جمله می توان به** FT232 **و** PL2303 **اشاره کرد و یا می توان از مبدل های** USB **به ســریال استفاده کرد.**

❖ **توابع موجود در** Codevision **برای کار با واحد** USART

- getchar()
- putchar()
- putsf()
- puts()
- printf()

- **مثال۲: مثال قبل را به صورت سخت افزاری با کامپیوتر و با ارتباط** RS232 **راه اندازی کنید.**