

# Configuración megaAVR® USART

---

En esta sección, cubriremos los pasos básicos de codificación necesarios para configurar/usar el módulo USART en un MCU megaAVR®, con un enfoque en el dispositivo **ATmega328PB** .  
(<http://www.microchip.com/wwwproducts/en/ATmega328PB>)

## Visión general

El módulo USART consta de tres secciones principales, como se muestra en el siguiente diagrama: **generador de reloj** , **transmisor** y **receptor** .

The diagram illustrates the internal architecture of the USART module, organized into three main functional blocks: **Clock Generator**, **Transmitter**, and **Receiver**, all interfaced with a **DATA BUS**.

- Clock Generator:** Contains the **UBRRn[H:L]** register and the **BAUD RATE GENERATOR**. It receives an external oscillator signal (**OSC**) and provides a clock signal to the **SYNC LOGIC** and the **Transmitter**.
- Transmitter:** Contains the **UDRn(Transmit)** register, **TRANSMIT SHIFT REGISTER**, **PARITY GENERATOR**, and **PIN CONTROL**. Data from the **DATA BUS** is loaded into the **TRANSMIT SHIFT REGISTER**, which then feeds into the **PARITY GENERATOR** and the **PIN CONTROL** block. The **PIN CONTROL** block outputs the **TxDn** signal.
- Receiver:** Contains the **RECEIVE SHIFT REGISTER**, **CLOCK RECOVERY**, **DATA RECOVERY**, **PARITY CHECKER**, and **PIN CONTROL**. The **PIN CONTROL** block receives the **RxDn** signal and provides feedback to the **CLOCK RECOVERY** and **DATA RECOVERY** blocks. The **RECEIVE SHIFT REGISTER** outputs data to the **DATA BUS** and the **UDRn(Receive)** register.
- Control Registers:** **UCSRnA**, **UCSRnB**, and **UCSRnC** are shown at the bottom, each with a bidirectional connection to the **DATA BUS**.



Los registros clave (resaltados en gris) incluyen:

- Registros de control y estado ( **UCSRnA** , **UCSRnB** , **UCSRnC** ) compartidos por las tres secciones.
- Registro de datos **UDRn** compartido por las secciones Transmisor y Receptor.
- Registros de control de velocidad en baudios **UBRRn[H:L]** utilizados por el generador de reloj.

## consecuencia

consecuencia.

2. Habilite las secciones de transmisión y recepción en serie de USART.
3. Si está transmitiendo, espere hasta que el registro de desplazamiento de transmisión esté vacío (sondee en **UCSRnA.UDREN**), luego cargue su byte de datos en **UDRn**.
4. Si recibe, espere hasta que se establezca el bit de recepción de datos del receptor (sondee en **UCSRnA.RXCn**), luego lea los datos de **UDRn**. La lectura de UDRn borra automáticamente el bit y prepara el hardware para recibir el siguiente byte.

## Inicialización

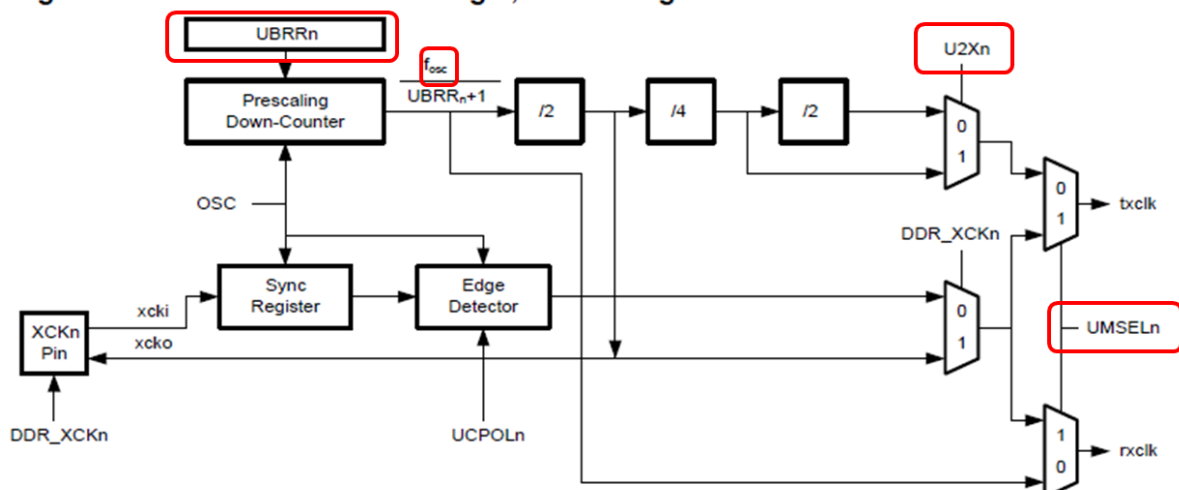
El USART debe inicializarse antes de que pueda tener lugar cualquier comunicación. El proceso de inicialización normalmente consiste en:

- Configuración de la velocidad en baudios,
- Configuración del formato de cuadro y
- Habilitación del Transmisor o del Receptor según el uso.

## Configuración de la tasa de baudios

La generación de reloj interno se utiliza para el modo de funcionamiento asíncrono. La lógica de generación de reloj genera el reloj base para el transmisor y el receptor (los registros clave y los bits de control están resaltados):

Figure 24-2. Clock Generation Logic, Block Diagram



Signal description:

- txclk: Transmitter clock (internal signal).
- rxclk: Receiver base clock (internal signal).
- xcki: Input from XCKn pin (internal signal). Used for synchronous slave operation.
- xcko: Clock output to XCKn pin (internal signal). Used for synchronous master operation.
- **f<sub>osc</sub>: System clock frequency.**

(/local--files/8avr:usart-mega-configuration/usart-clock-generator-diagram.png)

## Selección de modo USART (UMSELn)

La ecuación de velocidad en baudios utilizada por el módulo se establece en función del modo de funcionamiento. Para la operación en modo asíncrono, los bits de selección de modo USART en el registro C de control y estado de USART ( **UCSRnC.UMSELn[1:0]** ) se utilizan para seleccionar **la operación asíncrona** ( **UMSEL[1:0] = 00** ) como se muestra:

**Name:** UCSR0C, UCSR1C  
**Offset:** 0xC2 + n\*0x08 [n=0..1]  
**Reset:** 0x06  
**Property:** -

Bit	7	6	5	4	3	2	1	0
	UMSEL[1:0]		UPM[1:0]		USBS	UCSZ1 / UDORD	UCSZ0 / UCPHA	UCPOL
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	1	0

(/local--files/8avr:usart-mega-configuration/usart-mode-select-bits.png)

## Modo de doble velocidad (U2Xn)

Para el modo asíncrono, la tasa de TX de USART se puede duplicar configurando el bit U2Xn en el registro UCSRnA ( **UCSRnA.U2Xn = 1** ).



Con el modo de velocidad doble configurado, el receptor solo usará la mitad del número de muestras (reducidas de 16 a 8) para el muestreo de datos y la recuperación del reloj y, por lo tanto, se requiere una configuración de velocidad en baudios y un reloj del sistema más precisos cuando se usa este modo.

## Registro de velocidad en baudios (UBRRn)

El registro de tasa de baudios USART ( **UBRRn** ) y el contador descendente conectado a él funcionan como un preescalador programable o generador de tasa de baudios. El contador regresivo, que se ejecuta en el reloj del sistema ( $f_{osc}$ ), se carga con el valor UBRRn cada vez que el contador llega a cero o cuando se escribe el registro UBRRnL. Se genera un reloj cada vez que el contador llega a cero. Este reloj es la salida del reloj del generador de velocidad en baudios ( $= f_{osc}/(UBRRn+1)$ ). El transmisor divide la salida del reloj del generador de velocidad en baudios por 2, 8 o 16 según el modo. La salida del generador de velocidad en baudios es utilizada directamente por el reloj del receptor y las unidades de recuperación de datos. Sin embargo, las unidades de recuperación usan una máquina de estado que usa 2, 8 o 16 estados según el modo establecido por el estado de los bits UMSEL, U2Xn y DDR\_XCK. La siguiente tabla contiene ecuaciones para calcular la tasa de baudios (en bits por segundo) y para calcular el valor UBRRn para cada modo de operación utilizando una fuente de reloj generada internamente.

**Table 24-1. Equations for Calculating Baud Rate Register Setting**

Operating Mode	Equation for Calculating Baud Rate <sup>(1)</sup>	Equation for Calculating UBRRn Value
Asynchronous Normal mode (U2Xn = 0)	$\text{BAUD} = \frac{f_{\text{osc}}}{16(\text{UBRRn} + 1)}$	$\text{UBRRn} = \frac{f_{\text{osc}}}{16\text{BAUD}} - 1$
Asynchronous Double Speed mode (U2Xn = 1)	$\text{BAUD} = \frac{f_{\text{osc}}}{8(\text{UBRRn} + 1)}$	$\text{UBRRn} = \frac{f_{\text{osc}}}{8\text{BAUD}} - 1$

(/local--files/8avr:usart-mega-configuration/usart-baud-equations.png)

- **BAUDIOS** : Tasa de baudios (en bits por segundo, bps)
- **f<sub>osc</sub>** : Frecuencia de reloj del oscilador del sistema
- **UBRRn** : Contenido de los Registros UBRRnH y UBRRnL, (0-4095).



La biblioteca **AVR-LIBC Setbaud** ([http://www.nongnu.org/avr-libc/user-manual/group\\_\\_util\\_\\_setbaud.html](http://www.nongnu.org/avr-libc/user-manual/group__util__setbaud.html)) contiene macros útiles para calcular los valores correctos para escribir en los registros UBRRnH y UBRRnL. Consulte el ejemplo de código de inicialización a continuación.

También se proporcionan tablas en la hoja de datos del dispositivo que contienen valores UBRRn para tasas de baudios comunes, dadas varias frecuencias de oscilador:

**Table 24-9. Examples of UBRRn Settings for Commonly Used Oscillator Frequencies**

Baud Rate [bps]	f <sub>osc</sub> = 16.0000MHz				f <sub>osc</sub> = 18.4320MHz				f <sub>osc</sub> = 20.0000MHz			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error
2400	416	-0.1%	832	0.0%	479	0.0%	959	0.0%	520	0.0%	1041	0.0%
4800	207	0.2%	416	-0.1%	239	0.0%	479	0.0%	259	0.2%	520	0.0%
9600	103	0.2%	207	0.2%	119	0.0%	239	0.0%	129	0.2%	259	0.2%
14.4k	68	0.6%	138	-0.1%	79	0.0%	159	0.0%	86	-0.2%	173	-0.2%
19.2k	51	0.2%	103	0.2%	59	0.0%	119	0.0%	64	0.2%	129	0.2%
28.8k	34	-0.8%	68	0.6%	39	0.0%	79	0.0%	42	0.9%	86	-0.2%
38.4k	25	0.2%	51	0.2%	29	0.0%	59	0.0%	32	-1.4%	64	0.2%
57.6k	16	2.1%	34	-0.8%	19	0.0%	39	0.0%	21	-1.4%	42	0.9%
76.8k	12	0.2%	25	0.2%	14	0.0%	29	0.0%	15	1.7%	32	-1.4%
115.2k	8	-3.5%	16	2.1%	9	0.0%	19	0.0%	10	-1.4%	21	-1.4%

(/local--files/8avr:usart-mega-configuration/usart-common-ubrrn-values.png)



Para los cálculos de frecuencia en baudios, generalmente se acepta que los porcentajes de error de menos de ± 2% son aceptables.

## Configuración del formato de marco

El registro C de control y estado de USART ( **UCSRnC** ) se utiliza para configurar el formato de la trama de comunicación UART: paridad, número de bits de parada y número de bits de datos. Los ajustes para el formato de cuadro típico “8N1” son los siguientes:

- **UPM[1:0] = 00** para Sin paridad
- **USBS = 0** para 1 bit de parada
- **UCSZ1[1:0] = 11** para 8 Bits

**Name:** UCSR0C, UCSR1C  
**Offset:** 0xC2 + n\*0x08 [n=0..1]  
**Reset:** 0x06  
**Property:** -

Bit	7	6	5	4	3	2	1	0
	UMSEL[1:0]		UPM[1:0]		USBS	UCSZ1 / UDORD	UCSZ0 / UCPHA	UCPOL
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	1	0

(/local--files/8avr:usart-mega-configuration/usart-frame-format-settings.png)

## Habilitación del transmisor

El transmisor USART se habilita configurando el bit de **habilitación de transmisión (TXEN)** en el registro **UCSRnB** :

**Name:** UCSR0B, UCSR1B  
**Offset:** 0xC1 + n\*0x08 [n=0..1]  
**Reset:** 0x00  
**Property:** -

Bit	7	6	5	4	3	2	1	0
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

(/local--files/8avr:usart-mega-configuration/usart-ucsrnb-txen.png)

Cuando el transmisor está habilitado, la operación normal del puerto del pin TxDn es anulada por el USART y se le asigna la función de salida en serie del transmisor.



La velocidad en baudios, el modo de operación y el formato de trama deben configurarse una vez antes de realizar cualquier transmisión.

## Habilitación del receptor

El receptor USART se habilita escribiendo el bit de **habilitación de recepción (RXEN)** en el registro **UCSRnB** a '1':

**Name:** UCSR0B, UCSR1B  
**Offset:** 0xC1 + n\*0x08 [n=0..1]  
**Reset:** 0x00  
**Property:** -

Bit	7	6	5	4	3	2	1	0
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

(/local--files/8avr:usart-mega-configuration/usart-ucsrnb-rxen.png)

Cuando el Receptor está habilitado, el USART anula la operación normal del puerto del pin RxDn y se le asigna la función como entrada en serie del Receptor.



La velocidad en baudios, el modo de operación y el formato de trama deben configurarse una vez antes de realizar cualquier transmisión.

## Ejemplo de código

El siguiente ejemplo de código de inicialización de USART utiliza la biblioteca de **utilidades** **setbaud** ([http://www.nongnu.org/avr-libc/user-manual/group\\_\\_util\\_\\_setbaud.html](http://www.nongnu.org/avr-libc/user-manual/group__util__setbaud.html)) en AVR-LIBC. Esta biblioteca proporciona macros que usan el preprocesador c para calcular los valores apropiados para **UBBRn**. **Entradas** Este archivo de encabezado requiere que los valores de entrada ya estén definidos para **F\_CPU** y **BAUD**. Además, la macro **BAUD\_TOL** definirá la tolerancia de velocidad en baudios (en porcentaje) que es aceptable durante los cálculos. El valor de **BAUD\_TOL** por defecto será +/- 2%. **Salidas** Suponiendo que los **BAUD** solicitados son válidos para la **F\_CPU** dada, entonces la macro **UBRR\_VALUE**

se establece en el valor del preescalador requerido. Se proporcionan dos macros adicionales para los bytes alto y bajo del preescalador, respectivamente: **UBRRH\_VALUE** se establece en el byte superior de **UBRR\_VALUE** y **UBRRH\_VALUE** se establece en el byte inferior de **UBRR\_VALUE**. Se definirá una macro adicional **USE\_2X**. Su valor se establece en 1 si la tasa de **BAUDIOS** deseada dentro de la tolerancia dada solo se puede lograr al establecer el bit **U2Xn** en la configuración de UART. Se definirá a 0 si no se necesita **U2Xn**.



```

1  #define F_CPU 16000000UL
2  #define BAUD 38400UL
3  #define BAUD_TOL 2
4
5  #include <avr/io.h>
6  #include <util/setbaud.h>
7
8  void USART0_Init(void){
9
10     // Set the BAUD rate
11
12     UBRR0H = UBRRH_VALUE;
13     UBRR0L = UBRL_VALUE;
14     #if USE_2X
15     UCSRA |= (1 << U2X0);
16     #else
17     UCSRA &= ~(1 << U2X0);
18     #endif
19
20     // Set the Mode & Frame
21
22     UCSRC = 0x06;
23
24     // Enable USART0 Transmi
25
26     UCSRB = (1 << TXEN0) |
27
28 }

```



La biblioteca setbaud genera mensajes de advertencia durante la compilación si los parámetros de entrada generan una configuración de tasa BAUD que producirá una tasa de baudios fuera del BAUD\_TOL deseado.

## Transmisión de datos

### Transmitir

Una transmisión de datos se inicia cargando el búfer de transmisión con los datos a transmitir. La CPU puede cargar el búfer de transmisión escribiendo en el registro **UDRn**. Para la operación de sondeo, el firmware debe monitorear el indicador de registro de datos vacío ( **UCSRnA.UDREN** ) antes de cargar **UDRn**. Los datos almacenados en el búfer de transmisión se moverán al registro de desplazamiento cuando el registro de desplazamiento esté listo para enviar una nueva trama. El registro de desplazamiento se carga con nuevos datos si está en estado inactivo (sin transmisión en curso) o inmediatamente después de que se transmita el último bit de parada de la trama anterior. Cuando el registro de desplazamiento se carga con nuevos datos, transferirá un cuadro completo a la velocidad dada por el registro de baudios.



**Name:** UDR  
**Offset:**  $0xC6 + n \cdot 0x08$  [ $n=0..1$ ]  
**Reset:** 0x00  
**Property:** -

Bit	7	6	5	4	3	2	1	0
	TXB / RXB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

(/local--files/8avr:usart-mega-configuration/usart-udrn.png)



El indicador de interrupción de transmisión completa ( **USCRnA.TXCn** ) se establece y se puede generar una interrupción de TX opcional (si está habilitada) cuando se ha desplazado todo el marco en el registro de desplazamiento. El bit indicador **USCRnA.TXCn** se borra automáticamente cuando se ejecuta una interrupción de transmisión completa, o se puede borrar escribiendo un uno en su ubicación de bit.

## Recibir

El receptor inicia la recepción de datos cuando detecta un bit de inicio válido. Cada bit que sigue al bit de inicio se muestrearán a la velocidad en baudios o al reloj XCKn, y se desplazará al registro de desplazamiento de recepción hasta que se reciba el primer bit de parada de una trama. El búfer de recepción se puede leer leyendo el registro **UDRn** . La recepción completa de un byte se puede verificar sondeando el bit RXCn en el **registro USCRnA** .



El indicador de interrupción de recepción completa ( **RXCn** ) se establece y se puede generar una interrupción de RX opcional (si está habilitada) cuando el cuadro completo en el registro de desplazamiento se ha copiado en el registro **UDRn** . Esta es una interrupción *persistente* , es decir, el firmware debe leer los datos recibidos de **UDRn** para borrar el indicador **RXCn**

## Ejemplo de código

Las siguientes API de bloqueo simple envían y reciben un byte de datos a través de USART0.



```

// Example code for USART0
// ...

```

```
1 void USART0_Transmit(unsigned char data)
2
3     // Wait for empty transmit buffer
4     while(!(UCSR0A & (1 << UCR0A)));
5
6     // Put data into buffer,
7     UDR0 = data;
8
9 }
10
11 unsigned char USART0_Receive()
12
13     // Wait for data to be received
14     while(!(UCSR0A & (1 << FIFR0)));
15
16     // Get and return received data
17     return UDR0;
18
19 }
```

## Aprende más



### Configuración de interrupciones megaAVR®

Más información > (/8avr:interrupts-mega-configuration)



### MegaAVR® USART Ejemplo (sondeo)

Más información > (/8avr:usart-mega-example-pollled)