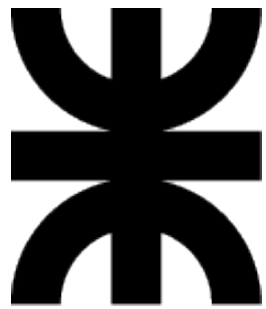
Informática II
Programación del microcontrolador ATmega328
UART

Gonzalo F. Perez Paina



Universidad Tecnológica Nacional
Facultad Regional Córdoba
UTN-FRC

– 2024 –

Puerto serie USART

El μC ATmega328 posee un periférico denominado USART (Universal Synchronous and Asynchronous Receiver and Transmitter) que permite la comunicación serie con otro dispositivo (p.e. una PC).

Puerto serie USART

El μC ATmega328 posee un periférico denominado USART (Universal Synchronous and Asynchronous Receiver and Transmitter) que permite la comunicación serie con otro dispositivo (p.e. una PC).

- La USART del ATmega328 se denomina USART0.

Puerto serie USART

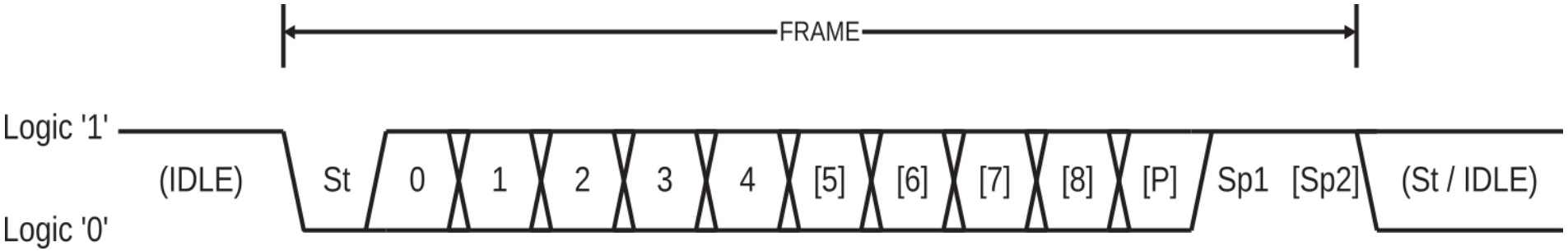
El μC ATmega328 posee un periférico denominado USART (Universal Synchronous and Asynchronous Receiver and Transmitter) que permite la comunicación serie con otro dispositivo (p.e. una PC).

- La USART del ATmega328 se denomina USART0.

Algunas características

- Permite operación Full-duplex (tiene registros independientes de transmisión y recepción)
- Permite operación síncrona (modo SPI) y asíncrona (modo UART)
- Soporta tramas seriales con 5, 6, 7, 8 y 9 bits de datos, y 1 o 2 bits de parada
- Generación de paridad impar o par y verificación de paridad por hardware

Puerto serie USART – Trama

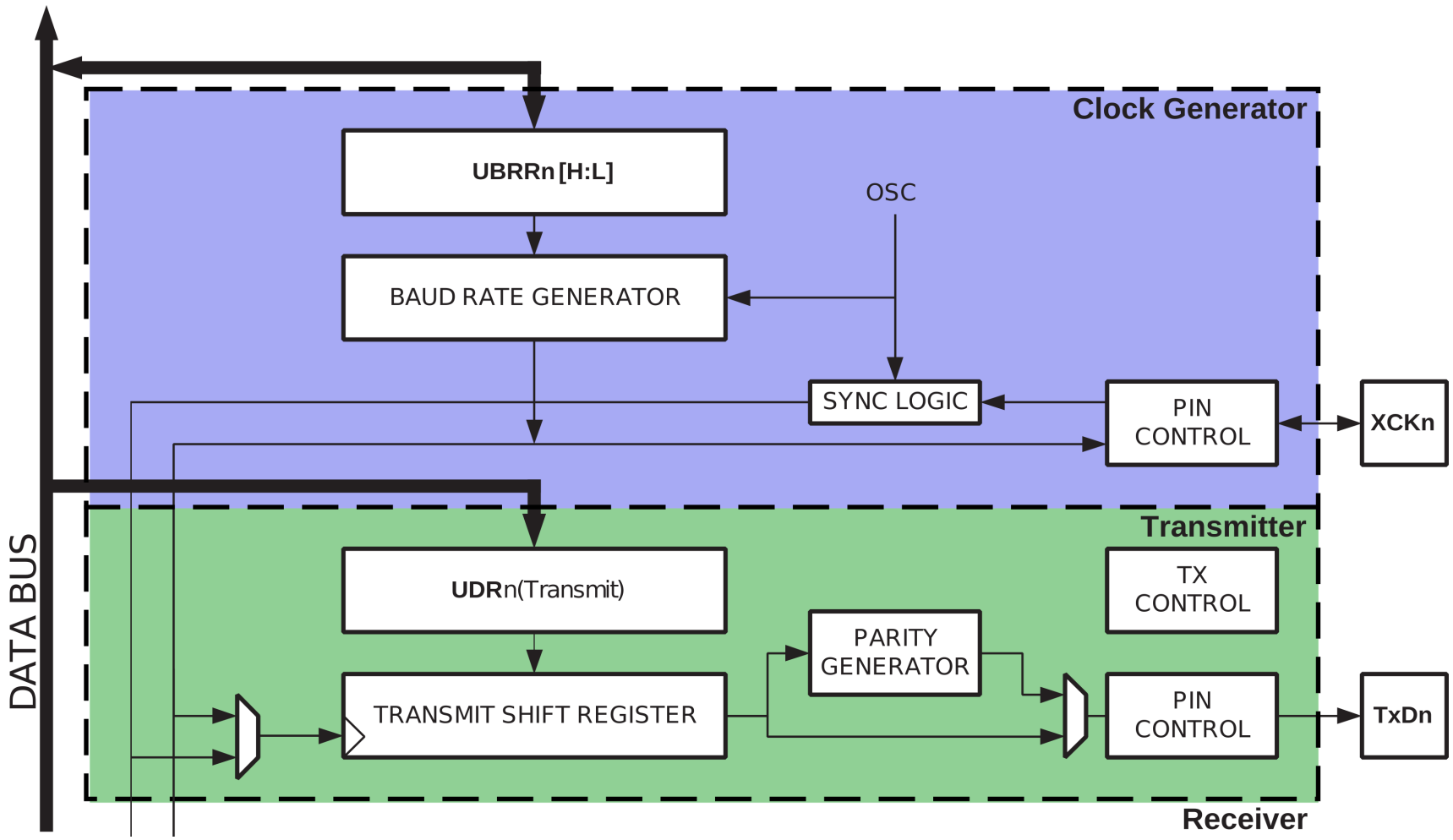


Donde:

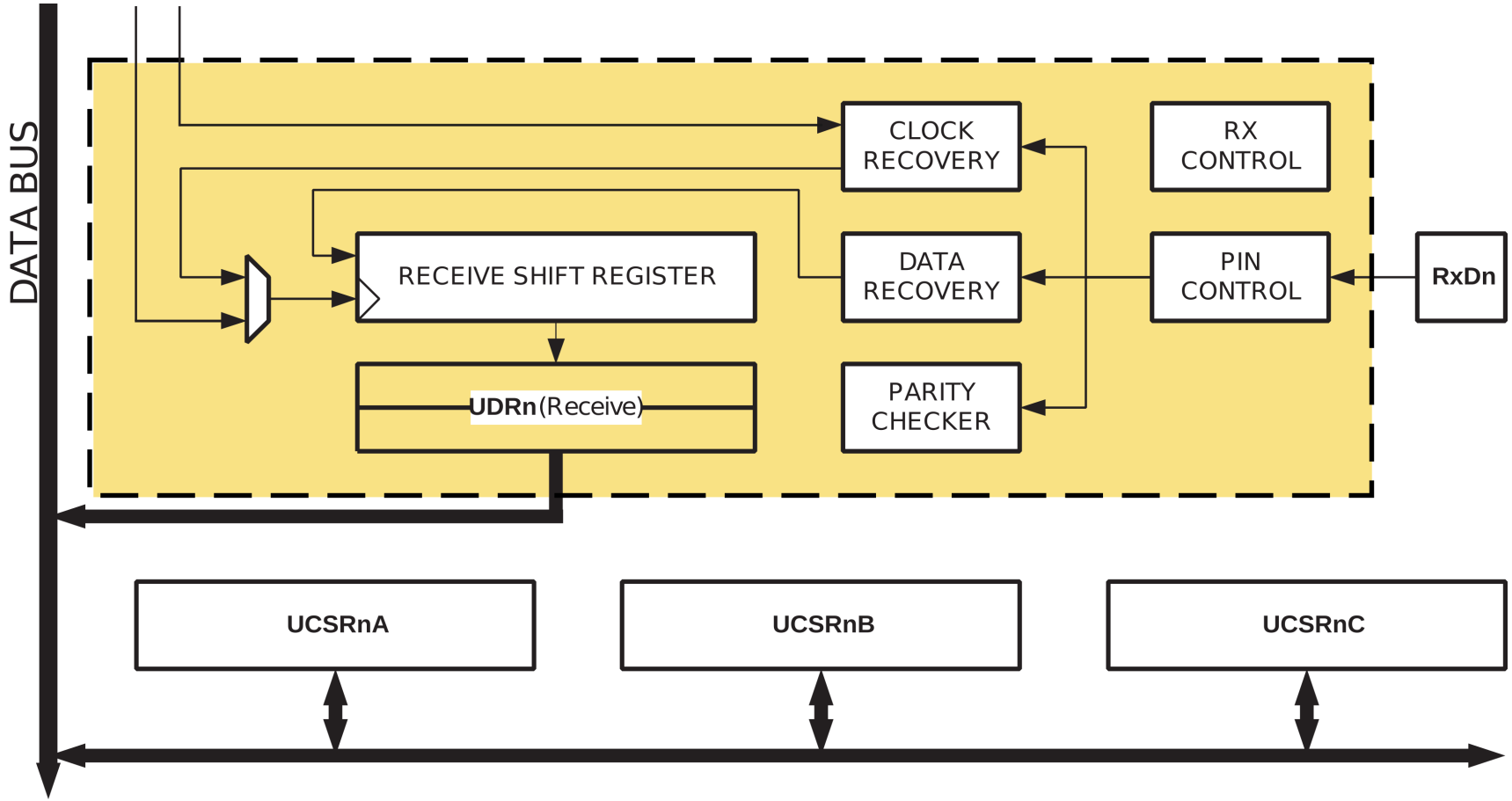
- **St:** Bit de inicio de trama (nivel bajo)
- **(n):** Bits de datos (0 a 8)
- **P:** Bit de paridad, que puede ser impar o par
- **Sp:** Bit de parada (nivel alto)
- **IDLE:** No hay transferencia de datos sobre las líneas de comunicación (RXD o TXD) (nivel alto)

Se envía primer el bit menos significativo (LSb: Least Significant bit)

Puerto serie USART – Diagrama en bloques (1/2)



Puerto serie USART – Diagrama en bloques (2/2)



Puerto serie USART – Velocidad de comunic.

- El registro Baud Rate (UBRRn) en conjunto con un contador descendente (down-counter) funcionan como un prescaler o generador de baud-rate programable.
- El contador descendente (corriendo a f_{osc}) se carga con el valor del registro UBRRn cada vez que alcanza el valor cero y genera un pulso de reloj.
- El reloj del generador de baud-rate tiene una frecuencia $f_{osc}/(\text{UBRRn}+1)$ la cual se divide luego por 2, 8 y 16 dependiendo del modo de la USART.

Puerto serie USART – Velocidad de comunic.

- El registro Baud Rate (UBRRn) en conjunto con un contador descendente (down-counter) funcionan como un prescaler o generador de baud-rate programable.
- El contador descendente (corriendo a f_{osc}) se carga con el valor del registro UBRRn cada vez que alcanza el valor cero y genera un pulso de reloj.
- El reloj del generador de baud-rate tiene una frecuencia $f_{osc}/(\text{UBRRn}+1)$ la cual se divide luego por 2, 8 y 16 dependiendo del modo de la USART.

En el modo asíncrono normal la tasa de transmisión de datos (BR) es

$$BR = \frac{f_{osc}}{16 \cdot (\text{UBRRn} + 1)},$$

por lo tanto

$$\text{UBRRn} = \frac{f_{osc}}{16 \cdot BR} - 1.$$

donde UBRRn es el valor del registro de velocidad.

Puerto serie USART – Velocidad de comunic.

Baud rate	UBRRn	Error (%)	Baud rate	UBRRn	Error (%)
2400	416	-0.1	28.8K	34	-0.8
4800	207	0.2	38.4K	25	0.2
9600	103	0.2	57.6K	16	2.1
14.4K	68	0.6	76.8K	12	0.2
19.2K	51	0.2	115.2K	8	-3.5

Velocidades de comunicación y valor del registro UBRRn
para $f_{osc} = 16MHz$

Puerto serie USART – Inicialización

- Para poder utilizar la USART en una comunicación es necesario su inicialización.
- En general, el proceso de inicialización consiste en configurar el baud-rate, el formato de la trama y habilitar el transmisor y/o el receptor.

Puerto serie USART – Inicialización

- Para poder utilizar la USART en una comunicación es necesario su inicialización.
- En general, el proceso de inicialización consiste en configurar el baud-rate, el formato de la trama y habilitar el transmisor y/o el receptor.

Configuración:

1. Configuración del baud-rate: se realiza escribiendo los registros UBRR0[H:L]

Puerto serie USART – Inicialización

- Para poder utilizar la USART en una comunicación es necesario su inicialización.
- En general, el proceso de inicialización consiste en configurar el baud-rate, el formato de la trama y habilitar el transmisor y/o el receptor.

Configuración:

1. Configuración del baud-rate: se realiza escribiendo los registros UBRR0[H:L]
2. El formato de la trama de comunicación se configura utilizando el registro UCSR0C

Puerto serie USART – Inicialización

- Para poder utilizar la USART en una comunicación es necesario su inicialización.
- En general, el proceso de inicialización consiste en configurar el baud-rate, el formato de la trama y habilitar el transmisor y/o el receptor.

Configuración:

1. Configuración del baud-rate: se realiza escribiendo los registros UBRR0[H:L]
2. El formato de la trama de comunicación se configura utilizando el registro UCSR0C
3. La habilitación de la transmisión y recepción se realiza a través del registro UCSR0B

Puerto serie USART – Registros

Bit	7	6	5	4	3	2	1	0	
0xC6	**RXB[7:0]**								**UDR0 (Read)**
	TXB[7:0]								**UDR0 (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	

0xC0	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0	**UCSR0A**
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial value	0	0	1	0	0	0	0	0	

0xC1	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80	**UCSR0B**
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	

0xC2	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01	UCSZ00	UCPOL0	**UCSR0C**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial value	0	0	0	0	0	1	1	0	

0xC5	-	-	-	-	**UBRR0[11:8]**				**UBRR0H**
0xC4	**UBRR0[7:0]**								**UBRR0L**
Read/Write	R	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	

USART0 – Registro de datos

Los registros búfer de transmisión y de recepción de datos de la USART comparten la misma dirección de entrada/salida como registro `UDR0` (USART Data Register).

- Cuando se realiza una escritura en `UDR0`, los datos se almacenan en el registro búfer de transmisión (`TXB`).

USART0 – Registro de datos

Los registros búfer de transmisión y de recepción de datos de la USART comparten la misma dirección de entrada/salida como registro `UDR0` (USART Data Register).

- Cuando se realiza una escritura en `UDR0`, los datos se almacenan en el registro búfer de transmisión (`TXB`).
- Cuando se lee `UDR0` se obtiene el valor del registro búfer de recepción de datos (`RXB`).

USART0 – Registro de datos

Los registros búfer de transmisión y de recepción de datos de la USART comparten la misma dirección de entrada/salida como registro `UDR0` (USART Data Register).

- Cuando se realiza una escritura en `UDR0`, los datos se almacenan en el registro búfer de transmisión (`TXB`).
- Cuando se lee `UDR0` se obtiene el valor del registro búfer de recepción de datos (`RXB`).
- El búfer de transmisión puede escribirse cuando el bit `UDRE0` del registro `UCSR0A` este a 1.

USART0 – Registro de datos

Los registros búfer de transmisión y de recepción de datos de la USART comparten la misma dirección de entrada/salida como registro `UDR0` (USART Data Register).

- Cuando se realiza una escritura en `UDR0`, los datos se almacenan en el registro búfer de transmisión (`TXB`).
- Cuando se lee `UDR0` se obtiene el valor del registro búfer de recepción de datos (`RXB`).
- El búfer de transmisión puede escribirse cuando el bit `UDRE0` del registro `UCSR0A` este a 1.
- El búfer de recepción consta de una FIFO de dos niveles, cuyo estado cambia cada vez que se acceda al búfer de recepción.

USART0 – Registro de control y estado A

Funciones de los bits más utilizados del registro UCSR0A:

- Bit 7 – `RXC0` (USART Receive Complete): bit que actúa como bandera (flag) para indicar que hay un dato no leído en el búfer de recepción y se borra cuando el búfer de recepción está vacío.

USART0 – Registro de control y estado A

Funciones de los bits más utilizados del registro UCSR0A:

- Bit 7 – `RXC0` (USART Receive Complete): bit que actúa como bandera (flag) para indicar que hay un dato no leído en el búfer de recepción y se borra cuando el búfer de recepción está vacío.
- Bit 6 – `TXC0` (USART Transmit Complete): este bit de bandera se pone a 1 cuando la trama completa en el registro de desplazamiento de transmisión ha sido serializado y no hay datos nuevos en el búfer de transmisión.

USART0 – Registro de control y estado A

Funciones de los bits más utilizados del registro UCSR0A:

- Bit 7 – `RXC0` (USART Receive Complete): bit que actúa como bandera (flag) para indicar que hay un dato no leído en el búfer de recepción y se borra cuando el búfer de recepción está vacío.
- Bit 6 – `TXC0` (USART Transmit Complete): este bit de bandera se pone a 1 cuando la trama completa en el registro de desplazamiento de transmisión ha sido serializado y no hay datos nuevos en el búfer de transmisión.
- Bit 5 – `UDRE0` (USART Data Register Empty): este flag indica si el búfer de transmisión está listo para recibir un dato nuevo. Si está a 1 el búfer está vacío y listo para ser escrito.

USART0 – Registro de control y estado A

Funciones de los bits más utilizados del registro UCSR0A:

- Bit 7 – `RXC0` (USART Receive Complete): bit que actúa como bandera (flag) para indicar que hay un dato no leído en el búfer de recepción y se borra cuando el búfer de recepción está vacío.
- Bit 6 – `TXC0` (USART Transmit Complete): este bit de bandera se pone a 1 cuando la trama completa en el registro de desplazamiento de transmisión ha sido serializado y no hay datos nuevos en el búfer de transmisión.
- Bit 5 – `UDRE0` (USART Data Register Empty): este flag indica si el búfer de transmisión está listo para recibir un dato nuevo. Si está a 1 el búfer está vacío y listo para ser escrito.
- Bit 4 – `FE0` (Frame Error):

USART0 – Registro de control y estado A

Funciones de los bits más utilizados del registro UCSR0A:

- Bit 7 – `RXC0` (USART Receive Complete): bit que actúa como bandera (flag) para indicar que hay un dato no leído en el búfer de recepción y se borra cuando el búfer de recepción está vacío.
- Bit 6 – `TXC0` (USART Transmit Complete): este bit de bandera se pone a 1 cuando la trama completa en el registro de desplazamiento de transmisión ha sido serializado y no hay datos nuevos en el búfer de transmisión.
- Bit 5 – `UDRE0` (USART Data Register Empty): este flag indica si el búfer de transmisión está listo para recibir un dato nuevo. Si está a 1 el búfer está vacío y listo para ser escrito.
- Bit 4 – `FE0` (Frame Error):
- Bit 3 – `DOR0` (Data OverRun):

USART0 – Registro de control y estado A

Funciones de los bits más utilizados del registro UCSR0A:

- Bit 7 – `RXC0` (USART Receive Complete): bit que actúa como bandera (flag) para indicar que hay un dato no leído en el búfer de recepción y se borra cuando el búfer de recepción está vacío.
- Bit 6 – `TXC0` (USART Transmit Complete): este bit de bandera se pone a 1 cuando la trama completa en el registro de desplazamiento de transmisión ha sido serializado y no hay datos nuevos en el búfer de transmisión.
- Bit 5 – `UDRE0` (USART Data Register Empty): este flag indica si el búfer de transmisión está listo para recibir un dato nuevo. Si está a 1 el búfer está vacío y listo para ser escrito.
- Bit 4 – `FE0` (Frame Error):
- Bit 3 – `DOR0` (Data OverRun):
- Bit 2 – `UPE0` (USART Parity Error):

USART0 – Registro de control y estado B

Funciones de los bits más utilizados del registro UCSR0B:

- Bit 4 – `RXEN0` (Receiver Enable): al escribir un 1 se habilita la recepción de datos y se configura el pin `RxD0`.

USART0 – Registro de control y estado B

Funciones de los bits más utilizados del registro UCSR0B:

- Bit 4 – `RXEN0` (Receiver Enable): al escribir un 1 se habilita la recepción de datos y se configura el pin `RxD0`.
- Bit 3 – `TXEN0` (Transmitter Enable): al escribir un 1 se habilita la transmisión de datos y se configura el pin `TxD0`.

USART0 – Registro de control y estado B

Funciones de los bits más utilizados del registro UCSR0B:

- Bit 4 – `RXEN0` (Receiver Enable): al escribir un 1 se habilita la recepción de datos y se configura el pin `RxD0`.
- Bit 3 – `TXEN0` (Transmitter Enable): al escribir un 1 se habilita la transmisión de datos y se configura el pin `TxD0`.
- Bit 2 – `UCSZ02` (Character Size): este bit, en combinación con los bits `UCSZ01:0`, configuran la cantidad de bits de datos (Character SiZe) de la trama de transmisión y recepción.

USART0 – Registro de control y estado C

Funciones de los bits más utilizados del registro UCSR0C:

- Bits 7:6 – `UMSEL01:0` (USART Mode Select): estos bits seleccionan el modo de operación de la USART0, donde para el modo asíncrono los valores deben ser `00`.

USART0 – Registro de control y estado C

Funciones de los bits más utilizados del registro UCSR0C:

- Bits 7:6 – `UMSEL01:0` (USART Mode Select): estos bits seleccionan el modo de operación de la USART0, donde para el modo asíncrono los valores deben ser `00`.
- Bits 5:4 – `UPM01:0` (Parity Mode): fijan el tipo de paridad utilizada. Estos son: `00`, paridad desactivada; `01`, reservado; `10`, paridad par; y `11`, paridad impar.

USART0 – Registro de control y estado C

Funciones de los bits más utilizados del registro UCSR0C:

- Bits 7:6 – `UMSEL01:0` (USART Mode Select): estos bits seleccionan el modo de operación de la USART0, donde para el modo asíncrono los valores deben ser `00`.
- Bits 5:4 – `UPM01:0` (Parity Mode): fijan el tipo de paridad utilizada. Estos son: `00`, paridad desactivada; `01`, reservado; `10`, paridad par; y `11`, paridad impar.
- Bit 3 – `USBS0` (Stop Bit Select): seleccionan la cantidad de bits de stop agregados en la transmisión (el receptor ignora esta configuración). Las opciones son `0`: 1-bit de parada, o `1`: 2-bits de parada.

USART0 – Registro de control y estado C

Funciones de los bits más utilizados del registro UCSR0C:

- Bits 7:6 – `UMSEL01:0` (USART Mode Select): estos bits seleccionan el modo de operación de la USART0, donde para el modo asíncrono los valores deben ser `00`.
- Bits 5:4 – `UPM01:0` (Parity Mode): fijan el tipo de paridad utilizada. Estos son: `00`, paridad desactivada; `01`, reservado; `10`, paridad par; y `11`, paridad impar.
- Bit 3 – `USBS0` (Stop Bit Select): seleccionan la cantidad de bits de stop agregados en la transmisión (el receptor ignora esta configuración). Las opciones son `0`: 1-bit de parada, o `1`: 2-bits de parada.
- Bit 2:1 – `UCSZ01:0` (Character Size): estos bits, en conjunto con el bit `UCSZ02` en `UCSR0B`, configuran la cantidad de bits de datos (Character SiZe) utilizada en la trama de transmisión y recepción.

USART0 – Tamaño de caracter (UCSR0B/C)

UCSZ02	UCSZ01	UCSZ00	Tamaño de caracter
0	0	0	5 bits
0	0	1	6 bits
0	1	0	7 bits
0	1	1	8 bits
1	0	0	Reservado
1	0	1	Reservado
1	1	0	Reservado
1	1	1	9 bits

Programación de la UART – Configuración

La configuración de la USART consiste en:

- seleccionar el modo de funcionamiento asíncrono (UART),
- definir el tipo de trama,
- velocidad de comunicación (bps) y
- habilitar el transmisor y/o el receptor.

Programación de la UART – Configuración

La configuración de la USART consiste en:

- seleccionar el modo de funcionamiento asíncrono (UART),
- definir el tipo de trama,
- velocidad de comunicación (bps) y
- habilitar el transmisor y/o el receptor.

- El tipo de trama queda determinada por la cantidad de bits de datos, y de parada y si se habilita o no la detección de errores mediante paridad y el tipo de paridad (par o impar).

Programación de la UART – Configuración

La configuración de la USART consiste en:

- seleccionar el modo de funcionamiento asíncrono (UART),
- definir el tipo de trama,
- velocidad de comunicación (bps) y
- habilitar el transmisor y/o el receptor.

- El tipo de trama queda determinada por la cantidad de bits de datos, y de parada y si se habilita o no la detección de errores mediante paridad y el tipo de paridad (par o impar).
- La configuración de la USART0 del μC ATmega328 se utilizan mediante los registros de control y estado B y C, `UCSR0B` y `UCSR0C`.

Programación de la UART – Configuración

La configuración de la USART consiste en:

- seleccionar el modo de funcionamiento asíncrono (UART),
- definir el tipo de trama,
- velocidad de comunicación (bps) y
- habilitar el transmisor y/o el receptor.

- El tipo de trama queda determinada por la cantidad de bits de datos, y de parada y si se habilita o no la detección de errores mediante paridad y el tipo de paridad (par o impar).
- La configuración de la USART0 del μC ATmega328 se utilizan mediante los registros de control y estado B y C, `UCSR0B` y `UCSR0C`.
- El formato de trama suele indicarse por una combinación de dos números y una letra, p.e.: 8N1, 5E2, 7O1, etc.

Programación de la UART – Configuración

Definición de constantes simbólicas (antes de la función `main()`):

```
#define BAUD_RATE_4800 207
#define BAUD_RATE_9600 103
#define BAUD_RATE_38400 25
#define BAUD_RATE_57600 16
#define BAUD_RATE_115200 8
```

En la función main:

```
uint16_t ubrr = BAUD_RATE_115200;

/* Configuración el baud-rate */
UBRR0H = (ubrr >> 8) & 0xFF; /* Byte más significativo */
UBRR0L = ubrr & 0xFF; /* Byte menos significativo */

UCSR0C = _BV(UCSZ00) | _BV(UCSZ01); /* UART con trama 8N1 */
```

La última línea equivale a escribir el valor 6d = 00000110b = 0x06 en el registro.

Programación de la UART – Transmisión

Archivo 'hola_mundo.c'

```
1  #include <avr/io.h>
2  #include <util/delay.h>
3  #include <stdint.h>
4
5  #define BAUD_RATE_115200 8
6
7  int main()
8  {
9    int i;
10   uint16_t ubrr = BAUD_RATE_115200;
11   uint8_t dato[] = "Hola mundo\n";
12
13   /* Configuración el baud-rate */
14   UBRR0H = (ubrr >> 8) & 0xFF; /* Byte más significativo */
15   UBRR0L = ubrr & 0xFF; /* Byte menos significativo */
16
17   UCSR0C = _BV(UCSZ00) | _BV(UCSZ01); /* UART con trama 8N1 */
18   UCSR0B |= _BV(TXEN0); /* Habilita el transmisor */
19
```

Programación de la UART – Transmisión

Archivo 'hola_mundo.c' (cont.)

```
20    while(1)
21    {
22      for(i = 0; dato[i] != '\0'; i++)
23      {
24        /* Espera a que el búfer de transmisión esté vacío */
25        while( !(UCSR0A & _BV(UDRE0)) ) ;
26        UDR0 = dato[i]; /* Escribe el búfer de transmisión */
27      }
28      _delay_ms(500);
29    }
30    return 0;
31  }
```

Programación de la UART – Transmisión

Archivo 'hola_mundo.c' (cont.)

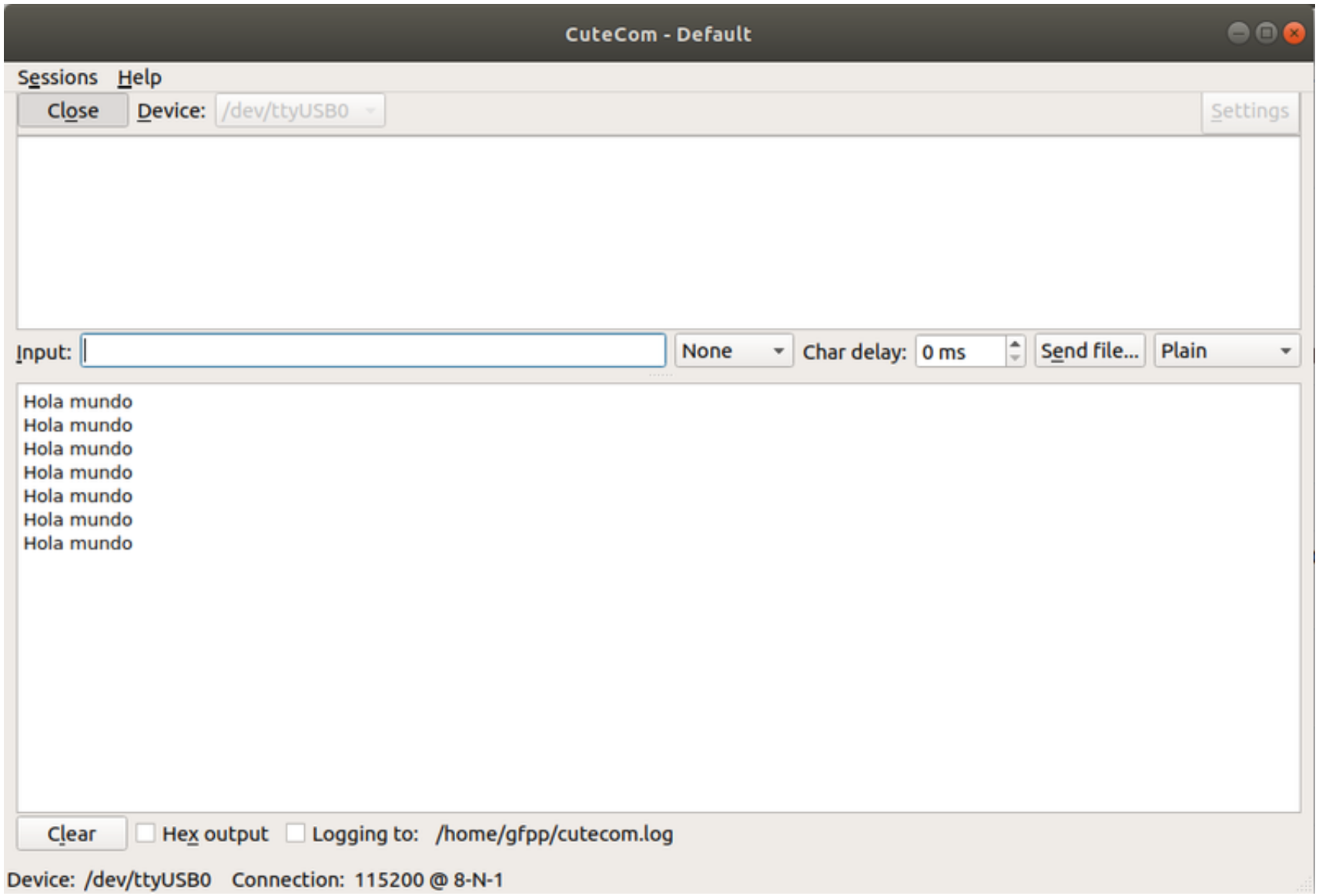
```
20   while(1)
21   {
22     for(i = 0; dato[i] != '\0'; i++)
23     {
24       /* Espera a que el búfer de transmisión esté vacío */
25       while( !(UCSR0A & _BV(UDRE0)) ) ;
26       UDR0 = dato[i]; /* Escribe el búfer de transmisión */
27     }
28     _delay_ms(500);
29   }
30   return 0;
31 }
```

Se escribe el registro de transmisión de datos, UDR0, habiendo verificado que el búfer de transmisión está vacío. La línea:

```
while( !(UCSR0A & _BV(UDRE0)) ) ;
```

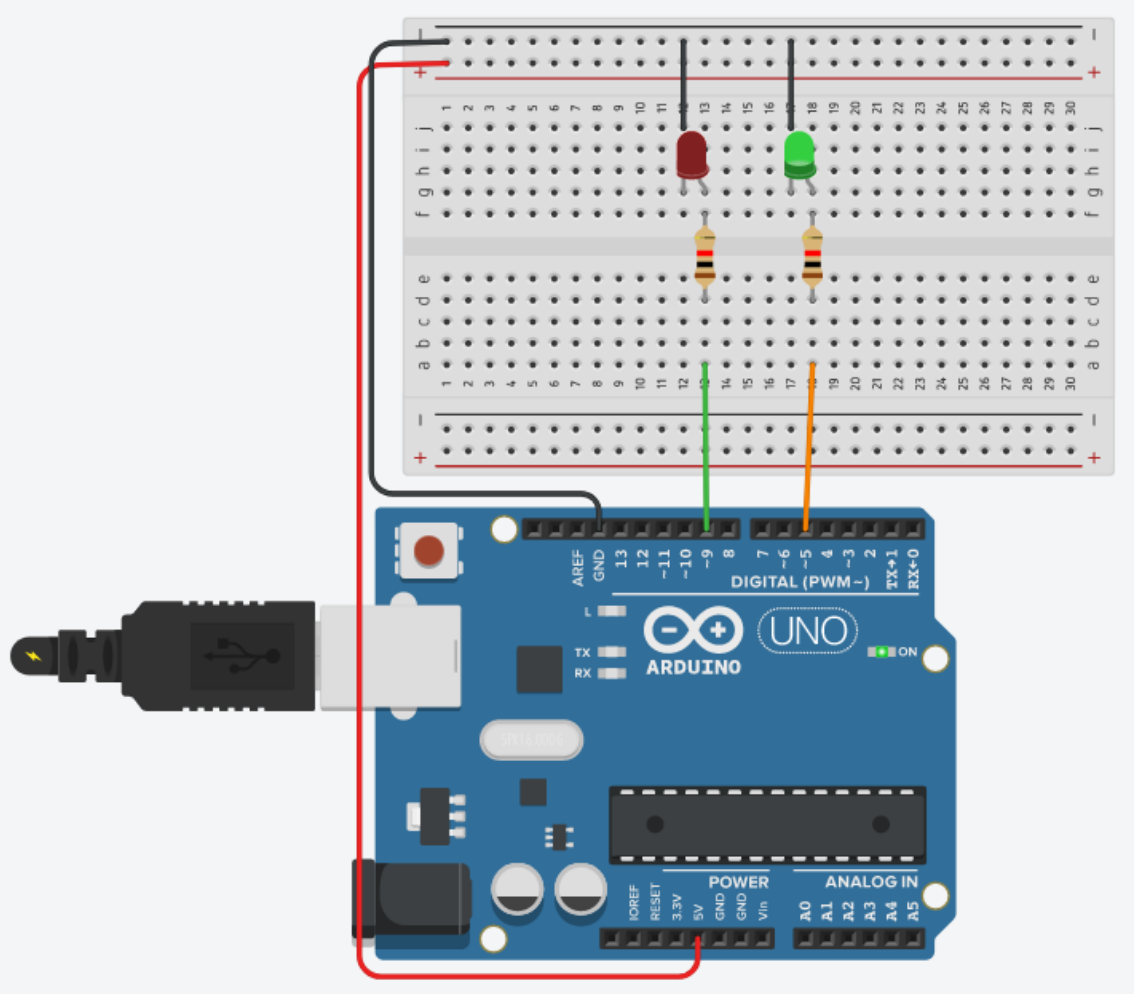
espera a que el búfer de transmisión este vacío.

Programación de la UART – Transmisión



Programación de la UART – Recepción

Circuito para la recepción de la UART



- El LED rojo, en `PB1`, cambia de estado al recibir el caracter `'r'`
- El LED verde, en `PD5`, cambia de estado al recibir el caracter `'v'`

Programación de la UART – Recepción

Archivo 'uart_read.c'

```
1  #include <avr/io.h>
2  #include <stdint.h>
3
4  #define BAUD_RATE_115200 8
5
6  int main()
7  {
8    uint16_t ubrr = BAUD_RATE_115200;
9
10   DDRB |= _BV(DDB1); /* Led rojo, salida */
11   DDRD |= _BV(DDD5); /* Led verde, salida */
12
13   /* Configuración el baud-rate */
14   UBRR0H = (ubrr >> 8) & 0xFF; /* Byte más significativo */
15   UBRR0L = ubrr & 0xFF; /* Byte menos significativo */
16
17   UCSR0C = _BV(UCSZ00) | _BV(UCSZ01); /* UART con trama 8N1 */
18   UCSR0B |= _BV(RXEN0); /* Habilita el receptor */
19
```

Programación de la UART – Recepción

Archivo 'uart_read.c' (cont.)

```
20   while(1)
21   {
22     /* Espera a recibir un dato */
23     while( !(UCSR0A & _BV(RXC0)) ) ;
24
25     switch(UDR0) {
26       case 'r':
27       case 'R':
28         PORTB ^= _BV(PORTB1); /* Cambia estado de LED rojo */
29         break;
30
31       case 'v':
32       case 'V':
33         PORTD ^= _BV(PORTD5); /* Cambia estado de LED verde */
34         break;
35     }
36   }
37   return 0;
38 }
```

Programación de la UART – Recepción

En el bucle principal la línea:

```
while( !(UCSR0A & _BV(RXC0)) ) ;
```

espera hasta recibir un caracter.

Programación de la UART – Recepción

En el bucle principal la línea:

```
while( !(UCSR0A & _BV(RXC0)) ) ;
```

espera hasta recibir un caracter.

Luego se utiliza la estructura de selección `switch-case` para actuar sobre el LED adecuado dependiendo del caracter recibido.

Programación de la UART – Trans. y recep.

Archivo 'to_upper.c'

```
1  #include <avr/io.h>
2  #include <stdint.h>
3  #define BAUD_RATE_57600 16
4
5  int main()
6  {
7    uint16_t ubrr = BAUD_RATE_57600;
8    uint8_t car;
9
10   /* Configuración el baud-rate */
11   UBRR0H = (ubrr >> 8) & 0xFF; /* Byte más significativo */
12   UBRR0L = ubrr & 0xFF; /* Byte menos significativo */
13
14   UCSR0C = _BV(UCSZ00) | _BV(UCSZ01); /* UART con trama 8N1 */
15   UCSR0B |= _BV(TXEN0); /* Habilita el transmisor */
16   UCSR0B |= _BV(RXEN0); /* Habilita el receptor */
17
```

Programación de la UART – Trans. y recep.

Archivo 'to_upper.c' (cont.)

```
18   while(1)
19   {
20     /* Espera a recibir un dato */
21     while( !(UCSR0A & _BV(RXC0)) ) ;
22     car = UDR0; /* Lee caracter */
23
24     if( (car >= 'a') && (car <= 'z') )
25       car -= 32;
26
27     /* Espera a que el búfer de transmisión esté vacío */
28     while( !(UCSR0A & _BV(UDRE0)) ) ;
29     UDR0 = car; /* Escribe el búfer de transmisión */
30   }
31   return 0;
32 }
```

Programación de la UART – Trans. y recep.

Archivo 'to_upper.c' (cont.)

```
18   while(1)
19   {
20     /* Espera a recibir un dato */
21     while( !(UCSR0A & _BV(RXC0)) ) ;
22     car = UDR0; /* Lee caracter */
23
24     if( (car >= 'a') && (car <= 'z') )
25       car -= 32;
26
27     /* Espera a que el búfer de transmisión esté vacío */
28     while( !(UCSR0A & _BV(UDRE0)) ) ;
29     UDR0 = car; /* Escribe el búfer de transmisión */
30   }
31   return 0;
32 }
```

O bien utilizando funciones de la biblioteca estándar de C, como:

```
if( isalpha(car) )
  car = toupper(car);
```

para lo cual es necesario incluir el archivo de cabecera `ctype.h`.