



UNIVERSIDAD DE BUENOS AIRES  
FACULTAD DE INGENIERÍA  
Año 2014 - 1<sup>er</sup> Cuatrimestre

## Comunicación Serie

Protocolo RS232, niveles lógicos, ruido, distancia de cable

FECHA: 10 de junio de 2014

INTEGRANTES:

Cánepa, Víctor Manuel <canepavictor@gmail.com>	- #92.132
Ferrari Bihurriet, Francisco <fferrari@fi.uba.ar>	- #92.275
Picard, Agustín Martín <agustin.picard@gmail.com>	- #95.476

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Comunicación Serie</b>	<b>2</b>
2.1. Codificación de Valores Digitales . . . . .	3
2.2. Método de Transmisión Asíncrona . . . . .	3
2.3. Ruido . . . . .	5
2.4. Interfaces . . . . .	5
<b>3. Protocolo RS-232 (Especificación TIA/EIA-232-F [1])</b>	<b>7</b>
3.1. Características Mecánicas (ISO/IEC 2110) . . . . .	7
3.2. Características Eléctricas (ITU-T V.28) . . . . .	7
3.3. Características Funcionales (ITU-T V.24) . . . . .	9
3.4. Características de Procedimiento (ITU-T V.24) . . . . .	9
<b>4. Apéndice: Red Conceptual</b>	<b>11</b>

# 1. Introducción

La necesidad de transmitir datos por un medio desde un dispositivo a otro es algo muy usual en la electrónica. Entre la inmensidad de ejemplos que hay sobre comunicación de datos podemos hacer mención de dispositivos comunes como lo es el mouse conectado a una computadora, impresoras, teclados, teléfonos, etc.

Al disponer de una interfaz podremos transmitir datos por un medio en el cual un elemento actúa como emisor (quien transfiere el dato) y otro cumple el papel de receptor (recibe el dato que le fue enviado). Podríamos notar que es de suma importancia que ambos dispositivos “hablen el mismo idioma”, esto quiere decir que la información que se desea transferir llegue y sea entendida por el receptor.

Los protocolos de comunicación son fundamentales para facilitar la tarea de transmisión de datos, estos disponen de una interfaz bien definida basada en estándares, mediante los cuales se logra establecer un canal de comunicación entre dos dispositivos cualesquiera, configurados bajo una misma interfaz y norma.

El protocolo que será de nuestro interés y estudiaremos en este informe, es el RS-232, uno de los más comunes y usados para las comunicaciones de tipo serie. El estándar actual se define en la norma *EIA-232-F Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange* (última modificación, hecha en 1997, sobre el original de 1962). [1]

# 2. Comunicación Serie

Se denomina comunicación serie a aquella en la que se transmite la información por un solo canal de datos, en contraposición con la comunicación paralelo, donde la transmisión es de a múltiples líneas en simultáneo.

Se puede notar la diferencia entre estos dos por la velocidad con que se transmiten los datos. Mientras en la comunicación serie se dispone de un solo cable por donde se envían los datos, en la comunicación paralelo aumenta notablemente la velocidad de transmisión al disponer de varios cables que entregan a cada instante un paquete de datos completo (por ejemplo un paquete de 1 byte - 8 bits simultáneos). A pesar de que en paralelo la comunicación es más eficiente respecto a velocidad (en bit/s = bps = baudio) a costo de más cables, si se considera el alcance y el largo del cable permitido por el paralelo, se encuentran más limitaciones que con la comunicación serie, esto se debe a que la diafonía o ‘*crosstalk*’ (una señal perturba otra señal cercana) produce interferencias entre las líneas paralelas, siendo cada vez más notoria cuanto más largo se hace el cable.

La comunicación serie es la que se utiliza en el protocolo RS-232. Bajo esta modalidad, los datos se transfieren por un único cable (más el común o “tierra”), por lo que la transmisión debe hacerse de a un bit, completando los datos conforme avanza el tiempo. Un ejemplo puede ser un dispositivo que está capacitado para transferir 16 bits por segundo, si se deseara transmitir dos palabras con tamaño de un byte (8 bits) cada una, se requeriría un tiempo de  $\frac{2 \cdot 8 \text{ bit}}{16 \text{ bit/s}} = 1 \text{ s}$ .

Para que un receptor interprete una “cadena” de datos, debe tomar muestras bajo intervalos regulares de tiempo, con la misma velocidad con que emisor ha generado dicha “cadena”, esto hace necesario contar con una fuente de reloj o ‘*clock*’.

Es así que en la comunicación serie se presentan dos modos de transmisión, el sincrónico y el asincrónico. Con estos métodos podemos mantener los dispositivos cooperando en el perfecto sincronismo requerido por el envío de datos, evitando que un desfase en la toma de muestras altere la interpretación. En

el método sincrónico el reloj es compartido por ambos dispositivos, este puede estar generado por uno de ellos o bien una fuente externa. De este modo ambos dispositivos se encuentran sincronizados en todo momento. En cambio el método asincrónico funciona con un reloj en cada dispositivo, sincronizados a la misma frecuencia, pero susceptibles a pequeñas diferencias que tarde o temprano conllevan a un desfasaje. Como veremos, este problema es resuelto mediante la transmisión de paquetes de largo fijo, re-sincronizando entre la recepción de un paquete y otro.

Haremos énfasis en el método asincrónico ya que es el más sencillo y el de interés para este curso.

Más allá del modo, junto a los datos de transmisión, se pueden utilizar cables de control encargados de detener, alterar, o modificar la comunicación. Centrados en el modo asincrónico, con una configuración mínima, se puede obviar estos cables de control y solo hacer uso de los de datos. Hoy en día esta simplificación es muy usada, ya que las líneas de control eran más útiles en dispositivos antiguos, como los viejos módem.

## 2.1. Codificación de Valores Digitales

Como denominación para los valores lógicos, en el protocolo RS-232 queda definida la implementación de un código del tipo *No Retorno a Nivel Cero* (NRZ-L o ‘*Nonreturn to Zero Level*’). Esto se verá en la sección 3.2, pero es necesario hacer aquí un paréntesis para entender de qué se trata.

Un código *No retorno a Cero* (NRZ o ‘*Nonreturn to Zero*’) es aquel en el que existe un voltaje distinto para representar cada nivel lógico. Como característica de estos códigos, el voltaje se mantiene constante durante todo el intervalo de transmisión de un bit. En otros códigos existe un estado intermedio de “espera” entre un bit y el siguiente, de ahí el nombre, ya que en los códigos NRZ esto no sucede. El típico ejemplo es una señal donde el 0 lógico (“marca” o ‘*mark*’) es indicado con un valor de 0 V, en tanto que el 1 lógico (“espacio” o ‘*space*’) se corresponde con una tensión de 5 V.

*No Retorno a Nivel Cero* (NRZ-L) es un tipo de código NRZ, en el que un estado lógico está representado por un voltaje negativo y el otro por un voltaje positivo. En este caso, la señal nunca permanecerá en 0 V, por eso se habla de no retorno a **nivel** cero.

Como ejemplo se puede citar un caso en donde el 1 corresponda con  $-12\text{ V}$  y el 0 con  $12\text{ V}$ . Notar que el valor de tensión baja se corresponde con el 1 lógico, esto es algo común en transmisiones serie (y quizás no tan intuitivo), es por eso que todos los gráficos utilizarán esta nomenclatura, a fin de familiarizar al lector y no mezclar convenciones.

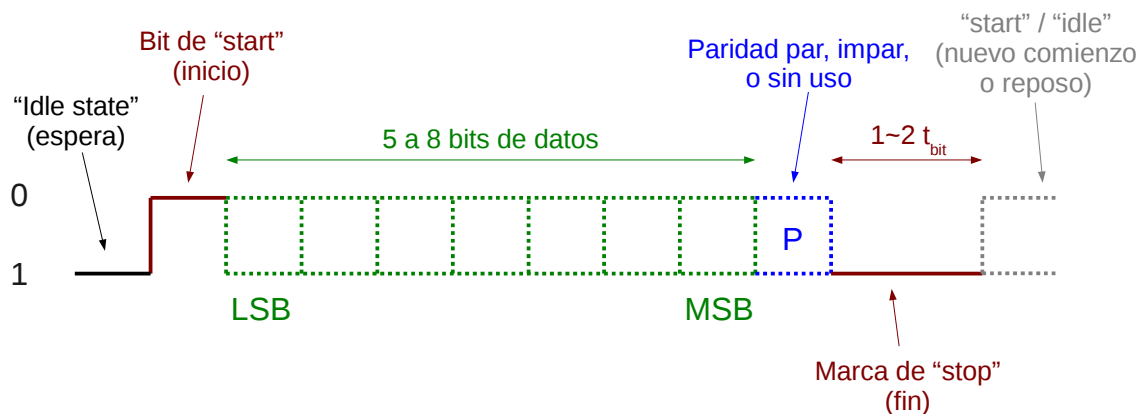
Es de interés notar que el código NRZ-L permite tener pulsos de mayor energía que si fuera un RZ (‘*Return to Zero*’).

Existen otros códigos que utilizan más de un estado de señal para un mismo bit (como el ‘*Manchester*’), o un único estado de señal para más de un bit (como el NRZI), pero esto es solo un dato informativo, para más detalles consultar el capítulo 4 de la referencia [2].

## 2.2. Método de Transmisión Asincrónica

Para transferir datos por la interfaz, en modo asincrónico, se envía junto con la información que se desea entregar, bits que indican el comienzo y final de un flujo de datos. Con esto, una importante característica que proporciona este método, por la cual se diferencia con el sincrónico, es que en cuanto comienza la transferencia de un paquete de datos, el receptor se sincroniza adecuándose al momento en que el transmisor comienza a enviar los datos.

La figura 2.1 muestra un ejemplo típico de transmisión asincrónica en NRZ:



**Figura 2.1:** Transmisión asincrónica típica

Si se transfieren valores digitales de a 1 bit por unidad de tiempo, se puede dividir los bits del flujo de datos en la cantidad que se aprecia en la figura, esto es un paquete de datos de 5 a 8 bits más un bit paridad y dos bits indicadores.

Si por un tiempo indefinido la línea permanece en 1, no hay transmisión de datos y el receptor se mantiene en espera (*'idle state'*). Cuando se cambia a 0, este primer bit, conocido como bit de *'start'*, indica que comenzará una nueva transmisión de datos (para un tamaño de datos y una velocidad previamente determinados). Este es el bit que permite que el receptor se sincronice cada vez que es indicado para registrar los datos, poniendo en fase su *'clock'* con el del transmisor.

Luego viene el paquete de datos (5 a 8 bits, comenzando el envío desde el bit menos significativo) seguido de un bit de paridad. Si la convención es paridad par, implica que el bloque total (incluido el bit de paridad) debe tener una cantidad par de unos. Si en cambio se habla de paridad impar, el transmisor genera el bit de paridad de manera que el bloque total tenga un número impar de unos. Esto permite detectar errores simples de transmisión, aunque el receptor bien podría ignorar este chequeo. El bit de paridad es opcional y esto se aclara previamente mediante una convención.

Finalmente, se coloca un bit en 1 de duración 1, 1,5 o 2 tiempos de bit. Con este último bit se indica el final del flujo para el dato y pasado su tiempo se habilita a un nuevo envío de datos o a permanecer nuevamente en espera.

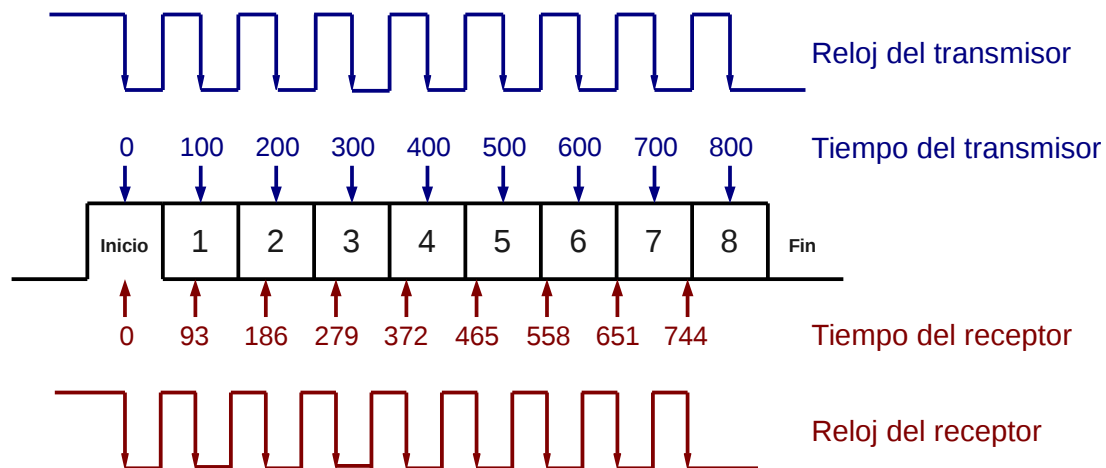
Para indicar los parámetros en la transmisión de datos, tales como la velocidad de bits, la cantidad de bits de datos, el tipo de paridad (o su ausencia) y la longitud del bit de *'stop'*, existe una nomenclatura descrita en la siguiente tabla:

Velocidad de bits	Bits de datos	Paridad	Tiempo de <i>'stop bit'</i>
4800	5	N: <i>'None'</i> (sin)	1
9600	6	E: <i>'Even'</i> (par)	1,5
19200	7	O: <i>'Odd'</i> (impar)	2
etc..	8		

**Cuadro 2.1:** Nomenclatura para los parámetros de transmisión

Así, para una transmisión a 9600 bps, con 8 bits de datos, sin paridad, con bit de stop de longitud 1 tiempo de bit (la más utilizada), la nomenclatura resultante es 9600/8N1, otros ejemplos son 9600/7E2 y 4800/7O1.

En la figura 2.2 se observan los pulsos de cada uno de los relojes sincronizados por el bit de *'start'*, tanto el del transmisor como el del receptor. Puede notarse que hay un desfase entre las frecuencias de pulsación, que a medida que van registrándose los datos, va arrastrando este error cada vez más y



**Figura 2.2:** Error por relojes desfasados en un 7 % de sus frecuencias

más. Se estará libre de este problema, siempre y cuando el desfase tenga un tamaño menor a un 5 % (para un paquete de 8 bits de datos).

El ejemplo de la figura 2.2 tiene un 7 % de desfase entre los relojes y presenta un error de lectura antes del bit de ‘stop’. Este problema puede ser más grave si además de interpretarse un dato erróneo, se malinterpreta un bit de datos como ‘start’ o ‘stop’, ya que toda referencia del principio y fin del paquete se perdería, cayendo en lo que se conoce como ‘*framing error*’.

## 2.3. Ruido

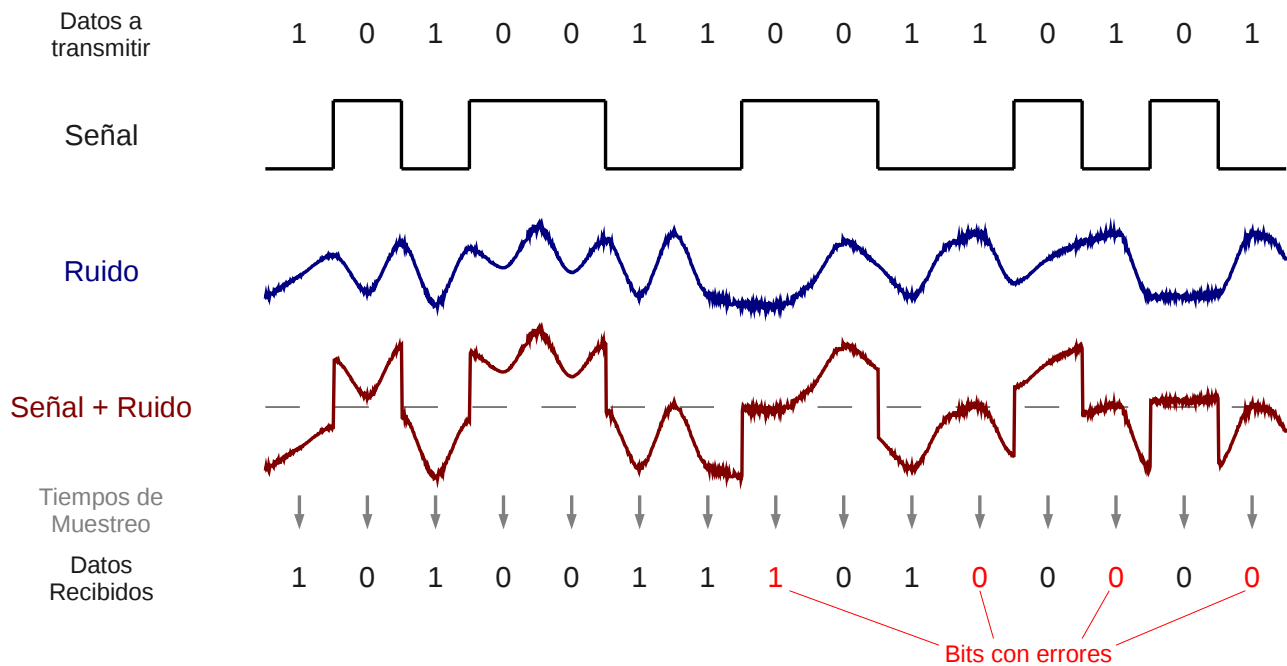
Se denomina ruido eléctrico a todas aquellas señales no deseadas, que están sumadas a la señal útil, de manera que la pueden alterar produciendo efectos perjudiciales. La principal fuente de ruido es la red que suministra la energía eléctrica, porque alrededor de los conductores se produce un campo electromagnético que oscila a 50 Hz. Además por estos conductores se propaga el ruido producido por otros dispositivos conectados al suministro.

El comportamiento general del ruido es aleatorio y con forma de onda indefinida o impredecible, es posible medir su amplitud para saber si es comparable con la señal útil. Si esto ocurre, en los sistemas digitales el ruido puede provocar errores en las transmisiones de datos (figura 2.3). Pero a diferencia de los sistemas analógicos, si el mismo no es comparable con la señal, no altera la información en absoluto.

Como ya se mencionó, la diafonía o ‘*crosstalk*’ es otro tipo de “ruido”, que se da en conductores cercanos, donde una señal interfiere con la otra. Esto es tenido en cuenta por la norma EIA-232 a la hora de imponer limitaciones eléctricas.

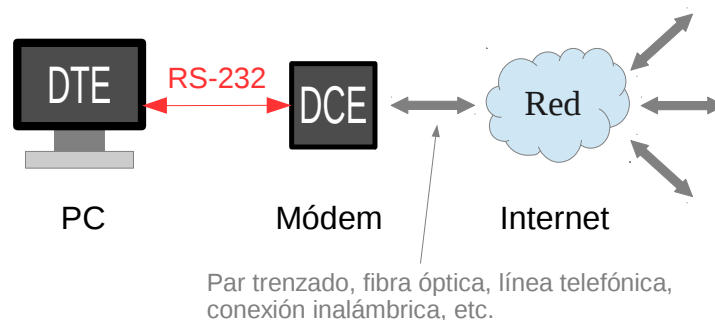
## 2.4. Interfaces

Como gran parte de los dispositivos de análisis de señales digitales poseen una capacidad de transmisión limitada, no es común su conexión directa a una red. Por esta razón necesitan de un DCE (‘*Data Circuit-Terminating Equipment*’ o actualmente aceptado como ‘*Data Communication Equipment*’), como un módem, para transmitir la información. Estos dispositivos (de análisis de señales digitales) de los que se hace mención se denominan comúnmente DTE (‘*Data Terminating Equipment*’). Un ejemplo típico podría ser una computadora o una impresora. El DCE utiliza lo que se denominan circuitos de



**Figura 2.3:** Error de lectura por un ruido cuya amplitud es comparable a la de la señal

intercambio (*'interchange circuits'*) para lograr la correcta transmisión de tanto los bits de información como los de control. La siguiente figura esquematiza la situación:



**Figura 2.4:** Conexión típica, idea original durante el desarrollo de la norma RS-232

Por un lado, el DTE es el responsable de la transmisión y recepción de la información, de a un bit; mientras que por el otro, el DCE debe interactuar con el DTE. Este proceso de transmisión requiere de mucha cooperación por parte de ambos equipos; es decir que ambos deben utilizar la misma codificación para los datos, la misma tasa de transferencia, etc. Por esta razón, cada par DCE/DTE debe estar diseñado para la cooperación mutua. Para facilitar esta tarea, tanto al fabricante como al usuario, se desarrollaron estándares, que definen las 4 características importantes de la interfaz: [2]

- **Mecánicas:** Se refiere a las características físicas de la conexión entre DTE y DCE. Normalmente, los circuitos de intercambio para la señal y el control están integrados en el conector del cable. Está normalizado que el DTE tenga el conector *macho*, reservando el conector *hembra* para el DCE.
- **Eléctricas:** Tienen que ver con el código utilizado, los niveles de tensión y el tiempo relacionado con los cambios en la tensión. Es decir, los mismos niveles de tensión deben significar lo mismo, la duración de distintos elementos de la señal deben ser iguales, etc. Estas especificaciones definen los límites de la tasa de transferencia y la longitud del cable.

- **Funcionales:** Define las funciones de cada circuito de intercambio. Estas funciones pueden estar relacionadas con la información, control, tiempos o la *tierra*.
- **De Procedimiento:** Especifica la secuencia de eventos para la transmisión de información, basado en las características funcionales de la interfaz.

Estas son las características generales de cualquier protocolo de comunicación. A continuación se focalizará en las características del protocolo RS-232.

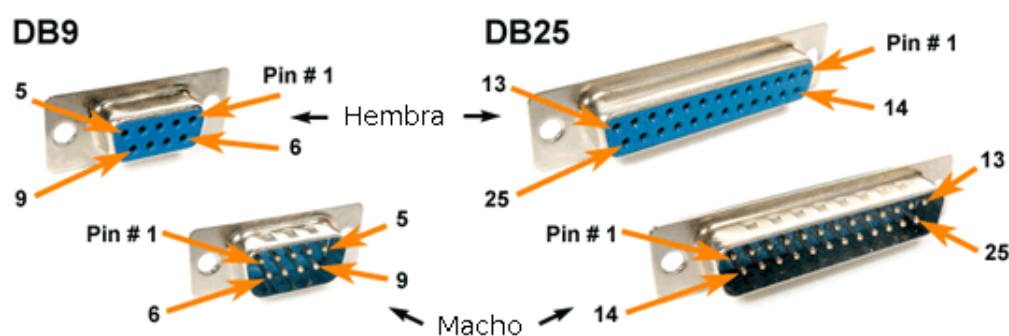
### 3. Protocolo RS-232 (Especificación TIA/EIA-232-F [1])

El estándar asegura:

- Voltajes y niveles de señal compatibles.
- Configuraciones de cableado y pines comunes.
- La mínima cantidad de información de control entre el DTE y el DCE.

#### 3.1. Características Mecánicas (ISO/IEC 2110)

Hay 2 tipos de conectores compatibles con el protocolo RS-232: DB-25 y DB-9 (aunque en realidad DB-9 no está en la norma, es un *estándar de facto*, impuesto muy generalizado en el mercado). DB-25 es un conector con 25 pines que permite comunicación tanto sincrónica como asincrónica, y recepción y transmisión en simultáneo (a lo que se denomina '*full-duplex*'). El DB-9, en cambio, posee 9 pines y es puramente asincrónico, ya que no tiene pines dedicados a la señal del '*clock*'. En la siguiente imagen se pueden visualizar sus diferencias:



**Figura 3.1:** Conectores DB-9 y DB-25 [4]

Como se puede apreciar, las funcionalidades del conector DB-9 son más reducidas y simples. A su vez, se hará énfasis en un subconjunto aún menor.

#### 3.2. Características Eléctricas (ITU-T V.28)

En todos los circuitos de intercambio se utilizan señales digitales, y dependiendo del circuito, serán señales en binario o señales de control.

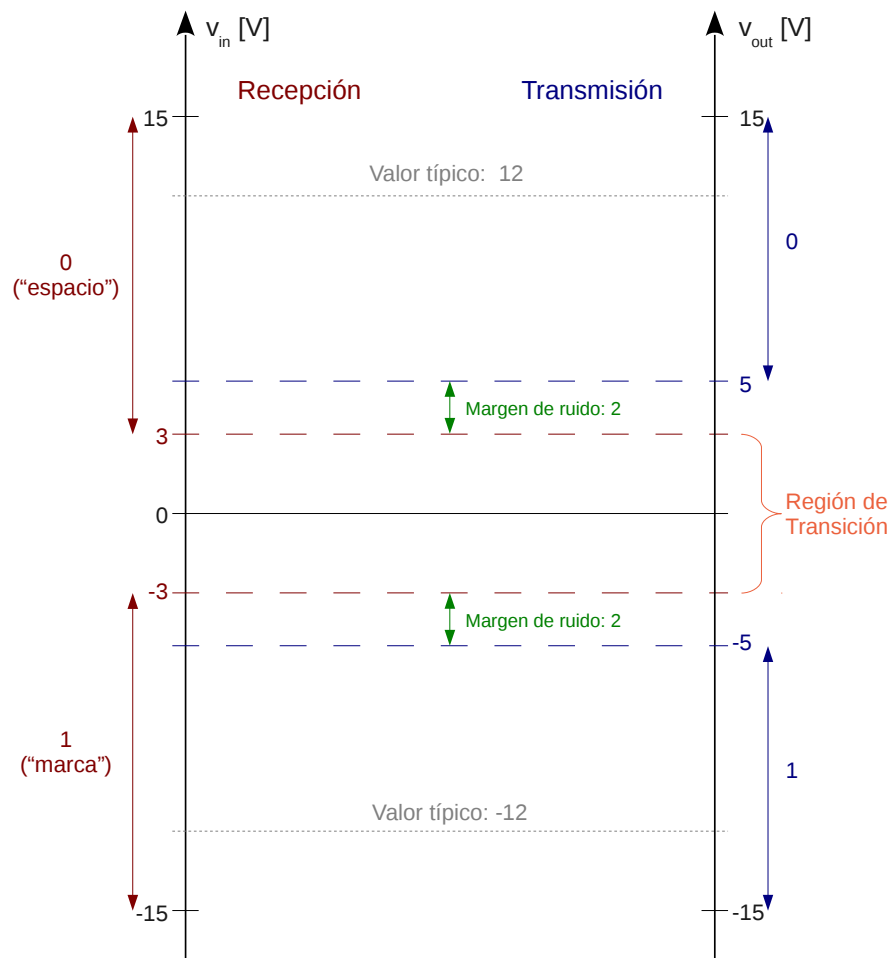
Para la recepción, la convención indica que, con respecto a la tierra, entre  $-15\text{ V}$  y  $-3\text{ V}$  significa un 1 lógico, mientras entre  $3\text{ V}$  y  $15\text{ V}$  será tomando como un 0 lógico. Lo mismo sucede con las señales



de control: tensiones mayores de 3 V son consideradas ON (encendido), mientras que menores de  $-3$  V significan OFF (apagado). La región entre  $-3$  V y  $3$  V se denomina *Región de Transición*.

Para la emisión de señales, sucede algo similar: entre  $-15$  V y  $-5$  V significa un 1 y entre  $5$  V y  $15$  V significa un 0 lógico, lo que otorga un margen de ruido de 2 V.

Originalmente, la tasa de transferencia estaba limitada a 20 kbps, y la máxima longitud del cable a 15 metros. La figura 3.2 muestra un esquema de estas características previamente discutidas.



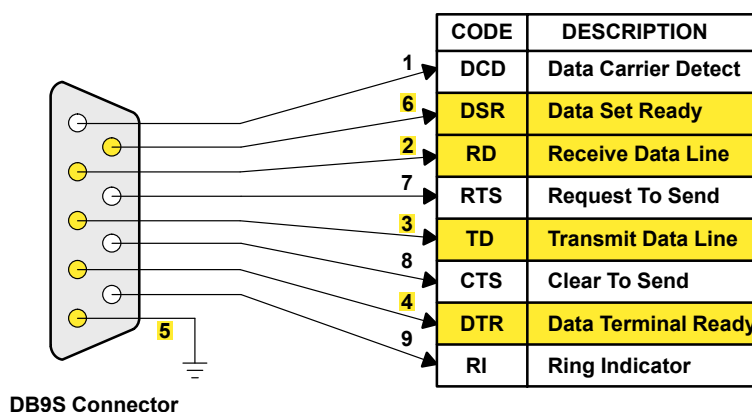
**Figura 3.2:** Tensiones y definiciones de niveles lógicos

El sistema debe ser diseñado de modo tal que cada pin sea capaz de soportar un cortocircuito con cualquier otro sin recibir un daño permanente. Además, la tensión en un pin no podrá exceder los  $\pm 25$  V, y cada línea debe tener una resistencia mínima de  $3\text{ k}\Omega$ , y una máxima de  $7\text{ k}\Omega$ . La capacitancia máxima de carga admitida es  $2500\text{ pF}$ . Esto limita los cables “comunes” a un largo de aproximadamente 15 a 20 metros.

La salida es una onda del tipo “cuadrada”, con una pendiente máxima cuyo valor está establecido en  $30\text{ V}/\mu\text{s}$ , esto es para evitar el ‘*crosstalk*’ entre los conductores. El tiempo en atravesar la región de transición ( $t_t$ ), debe ser de 1 ms en una transmisión por debajo de los 40 bps, o del 4 % de un tiempo de bit ( $t_b$ ), para velocidades mayores. Esto fija un límite teórico de velocidad de 200 kbps, donde  $t_t = 0.2\text{ }\mu\text{s}$ , alcanzando así la pendiente de  $30\text{ V}/\mu\text{s}$ . Sin embargo en la práctica, por limitaciones de corriente que dependen de la capacitancia de carga, la velocidad de transmisión está limitada a 20 kbps en cables “comunes”, pudiendo alcanzar los 120 kbps en cables más cortos, de baja capacidad.

### 3.3. Características Funcionales (ITU-T V.24)

La figura 3.3 muestra los circuitos de intercambio disponibles en el conector DB-9, aquí se resaltan los circuitos a utilizar en la placa experimental del curso.



**Figura 3.3:** Pines y funciones del conector DB-9. [3]

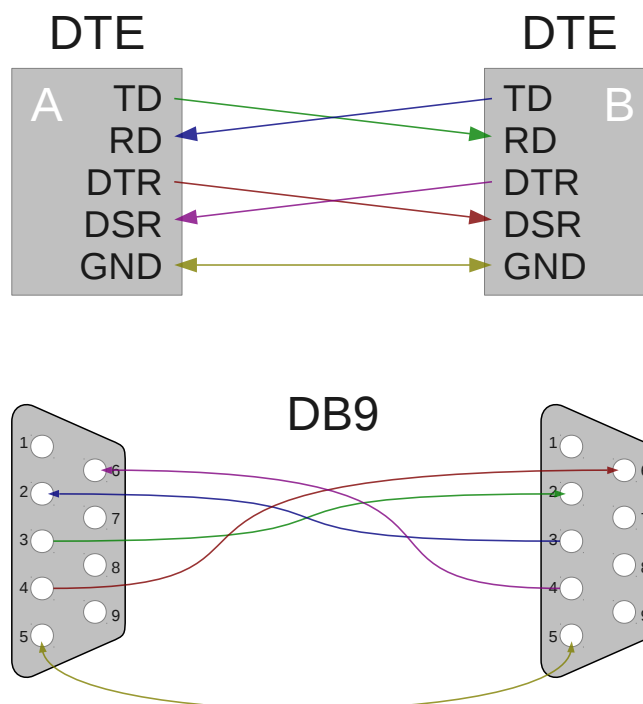
Se hará hincapié en las funciones RD (pin 2), TD (pin 3) y GND (pin 5), que corresponden a la recepción de datos, la transmisión de datos y la tierra respectivamente. Además se operará en el modo ‘*full-duplex*’ es decir, que es posible transmitir y recibir información simultáneamente, todo gracias a que hay una línea para la recepción y otra para la emisión.

Existen otros modos en los que esto no es posible. El modo ‘*simplex*’, implementa un canal de comunicación unidireccional, en el que los datos son enviados por un dispositivo y recibidos por el otro. Por otro lado, el modo ‘*half-duplex*’ sólo permite transmitir en una dirección a la vez, haciendo uso de los circuitos de intercambio para controlar qué terminal envía los datos y cuál los recibe. De esta manera el procedimiento de envío desde un DTE (conexión ya establecida) comienza con la activación del pin RTS (requerimiento de envío) y la posterior espera de una “respuesta”. Una vez que el DCE está listo, lo indica activando el pin CTS (listo para enviar). Es entonces cuando el DTE comienza a transmitir los datos.

Los circuitos de intercambio se pueden clasificar en datos, control y tierra; siendo RD y TD circuitos de datos, DSR y DTR circuitos de control, y GND el de tierra.

### 3.4. Características de Procedimiento (ITU-T V.24)

El estándar define características de procedimiento para unos 25 circuitos diferentes, incluso parte de un procedimiento se acaba de describir en la sección anterior, pero en el marco de la placa de desarrollo interesa sólo uno. El modelo de interfaces que se va a utilizar no dispone de un DCE, por lo que la conexión a realizar va a ser DTE a DTE (una placa experimental con otra). Este caso se conoce como ‘*Null-Módem*’ y se utiliza una conexión de *cable cruzado* para “engañar” a los DTE, donde cada uno “cree” que está interactuando con un DCE. El *cable cruzado* tiene dos conectores del mismo género, donde el TD de uno se conecta al RD del otro y viceversa. Las líneas DSR y DTR serán usadas para indicar el estado de los terminales. Cuando una placa experimental esté funcionando y lista para comunicarse, va a asegurar un 1 lógico en su salida DTR (terminal de datos listo). A su vez, una placa va a verificar que el otro extremo está en línea mediante la entrada DSR (equipo de datos listo). Al tratarse de ‘*Null-Módem*’, aquí también hay un “cruce” de cables ya que el DTR de una placa se conecta al DSR de la otra. Esto se esclarece en la figura 3.4, para más información consultar la referencia [2].



**Figura 3.4:** Conexión ‘Null-Módem’ (simplificación a utilizar)

En el apéndice se proporciona una red conceptual que hace énfasis en los temas discutidos, los recortes de contenidos y los alcances de la presente investigación.

Como comentario final, el microcontrolador utilizado es capaz de generar datos serie mediante hardware a través de su UART (Transmisor-Receptor Asíncrono Universal o ‘*Universal Asynchronous Receiver-Transmitter*’) y lo hace con señales de 0 V y 5 V, para asegurar las características eléctricas de la norma se utiliza el circuito integrado MAX-232, que permite la conexión con el puerto serie de una PC. Sin embargo en el proyecto se va a trabajar con los valores proporcionados por el microcontrolador, es decir que el 0 lógico estará representado por 0 V y el 1 lógico por 5 V.

## Referencias

- [1] Texas Instruments Incorporated (2002). *Interface Circuits For TIA/EIA-232-F* (SLLA037A) [en línea]. Disponible en: <http://www.ti.com/lit/an/slla037a/slla037a.pdf> (accedido el martes 10 de Junio de 2014).
- [2] W. Stallings, “The Data Communications Interface,” en *Data and Computer Communications*, 5th ed. College Park, MD: Prentice Hall, 1996, cap. 5, pág. 139-156.
- [3] Extraída de [1], modificada para corregir un error de tipeo y resaltar los pines que utilizaremos.
- [4] <http://logmett.com/images/stories/db9db25.gif> (accedida el martes 10 de Junio de 2014, sutilmente modificada para traducir términos al castellano).

## 4. Apéndice: Red Conceptual

