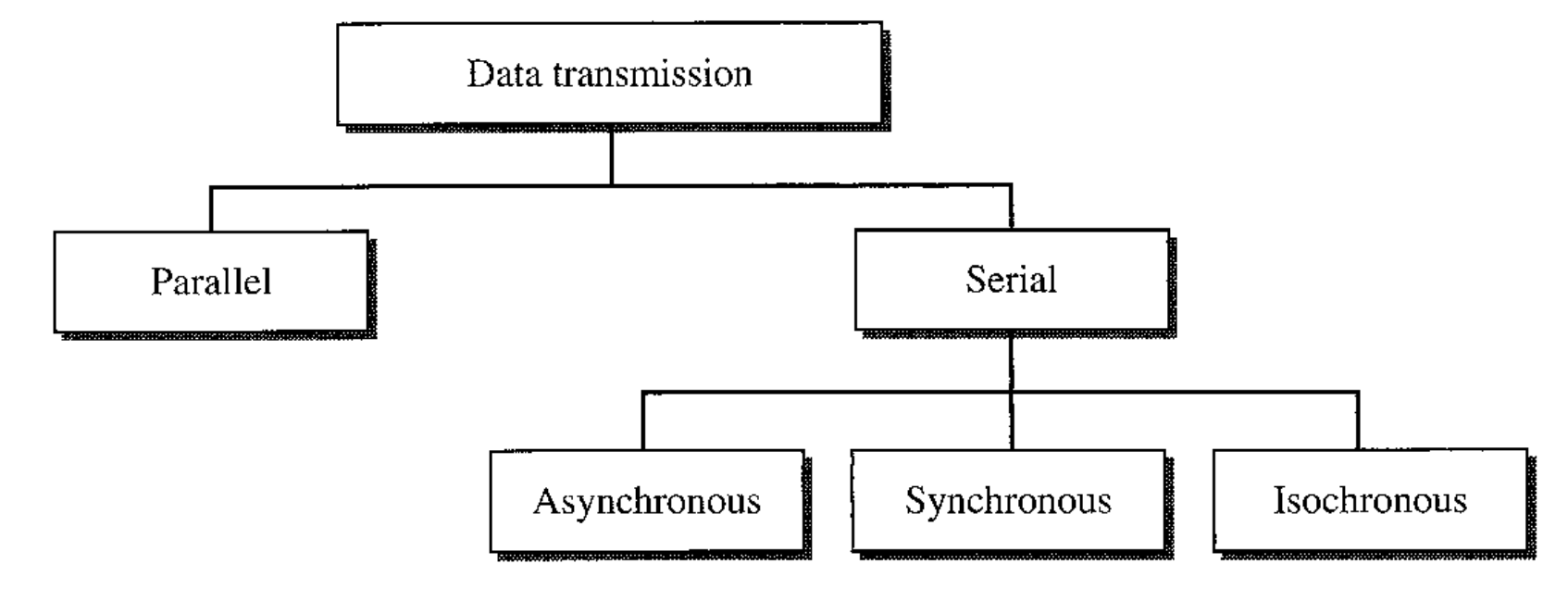
Σειριακή είσοδος/έξοδος του Arduino

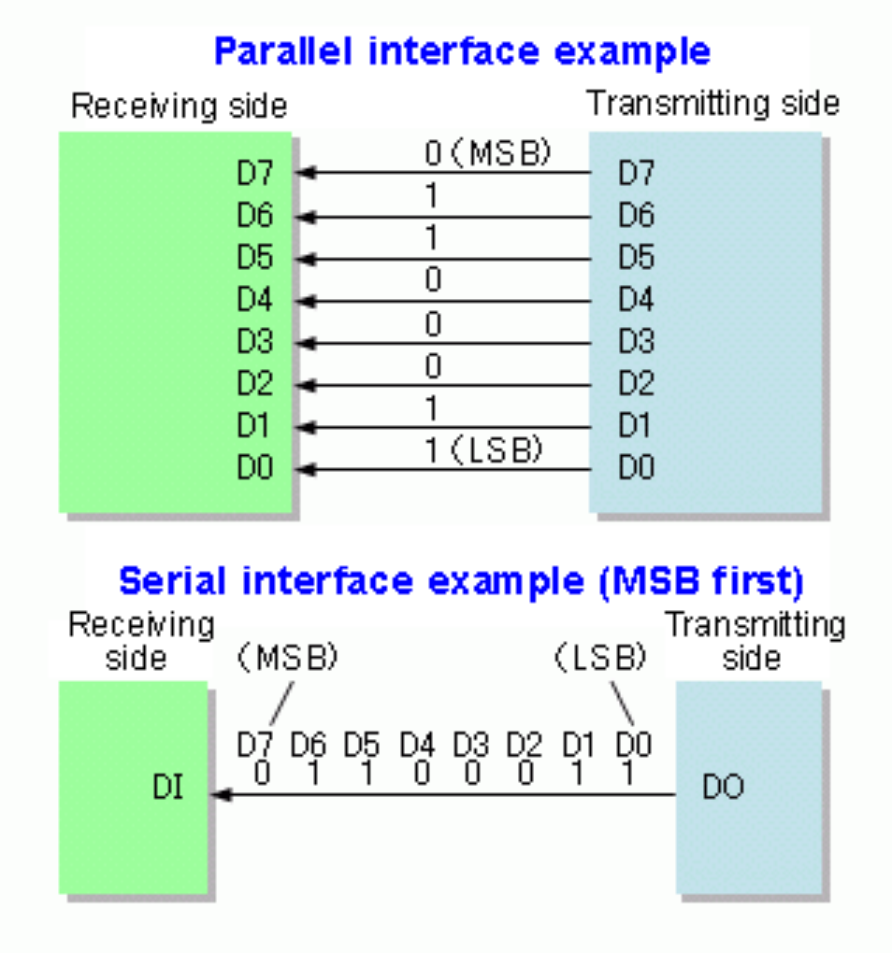
Τρόποι μετάδοσης δεδομένων



Σειριακή και παράλληλη μετάδοση

Με την *σειριακή μετάδοση* τα bit των κωδικοποιημένων χαρακτήρων αποστέλλονται το ένα κατόπιν του άλλου μέσα από ένα απλό κανάλι μετάδοσης.
Με την *παράλληλη μετάδοση* τα bit των κωδικοποιημένων χαρακτήρων αποστέλλονται ταυτόχρονα με την χρήση τόσων καναλιών όσων και τα bit των κωδικοποιημένων χαρακτήρων.

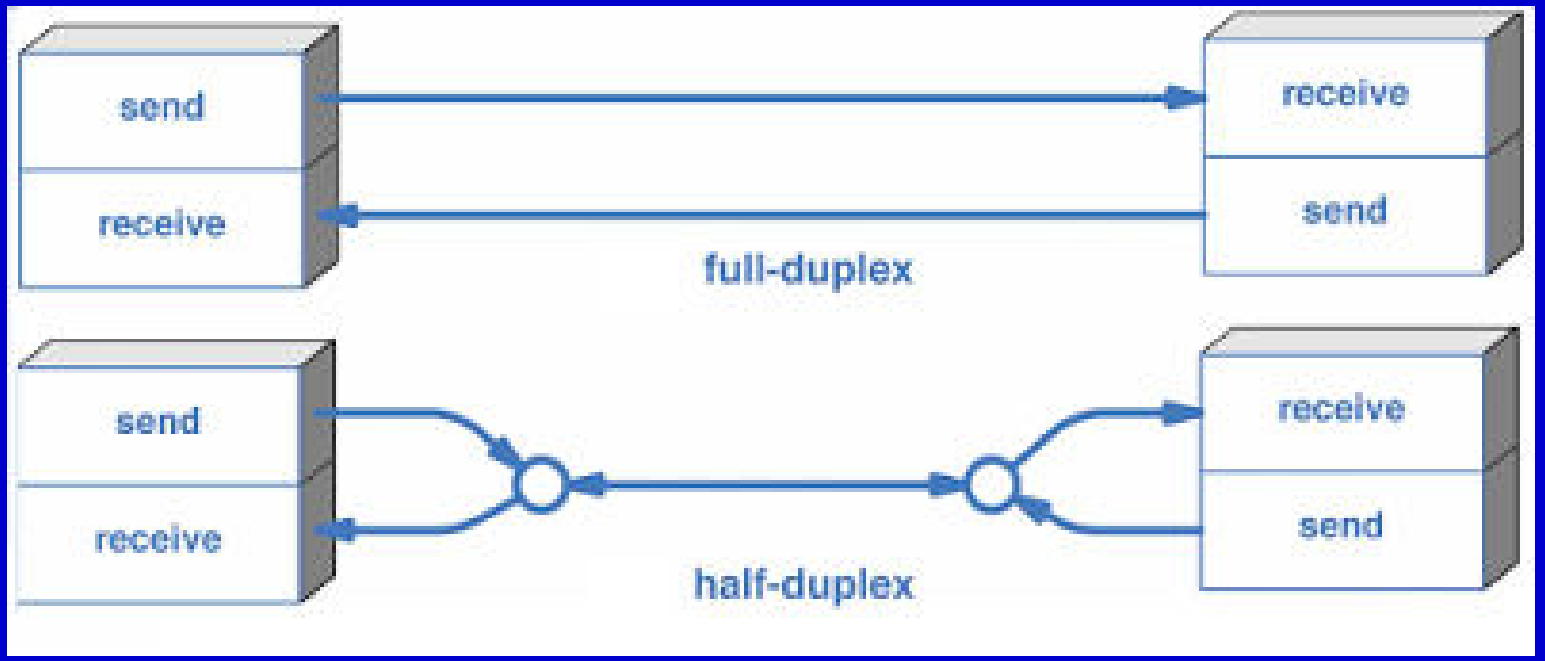
Παράλληλη και σειριακή μετάδοση



Μετάδοση Half Duplex και Full Duplex

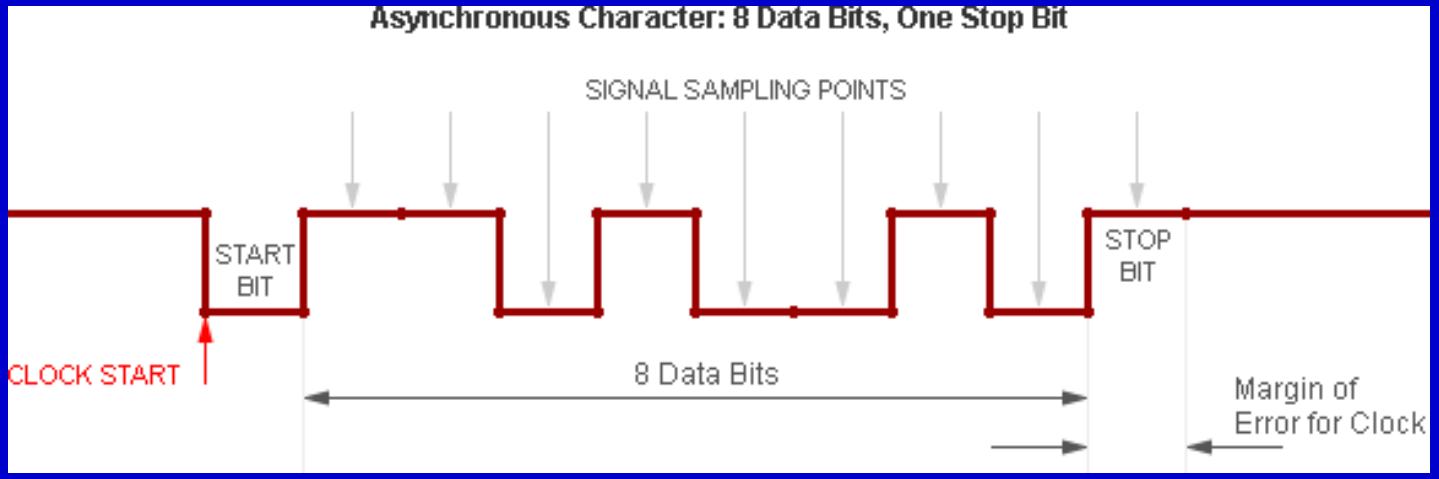
Με την μετάδοση *full duplex* και τα δύο συστήματα μπορούν να επικοινωνούν και προς τις δύο κατευθύνεις ταυτόχρονα. Παράδειγμα full-duplex μετάδοσης είναι το τηλεφωνικό δίκτυο με το οποίο οι χρήστες στα δύο άκρα μπορούν να ομιλούν και να ακούν ταυτόχρονα.
Με την μετάδοση *half-duplex* σε αντίθεση κάθε σύστημα μπορεί να επικοινωνεί με το άλλο αλλά όχι ταυτόχρονα. Η επικοινωνία γίνεται προς μία κατεύθυνση κάθε φορά. Παράδειγμα half-duplex επικοινωνίας είναι το walkie-talkie two-way radio το οποίο έχει πλήκτρο "push-to-talk".

Μετάδοση Half Duplex και Full Duplex

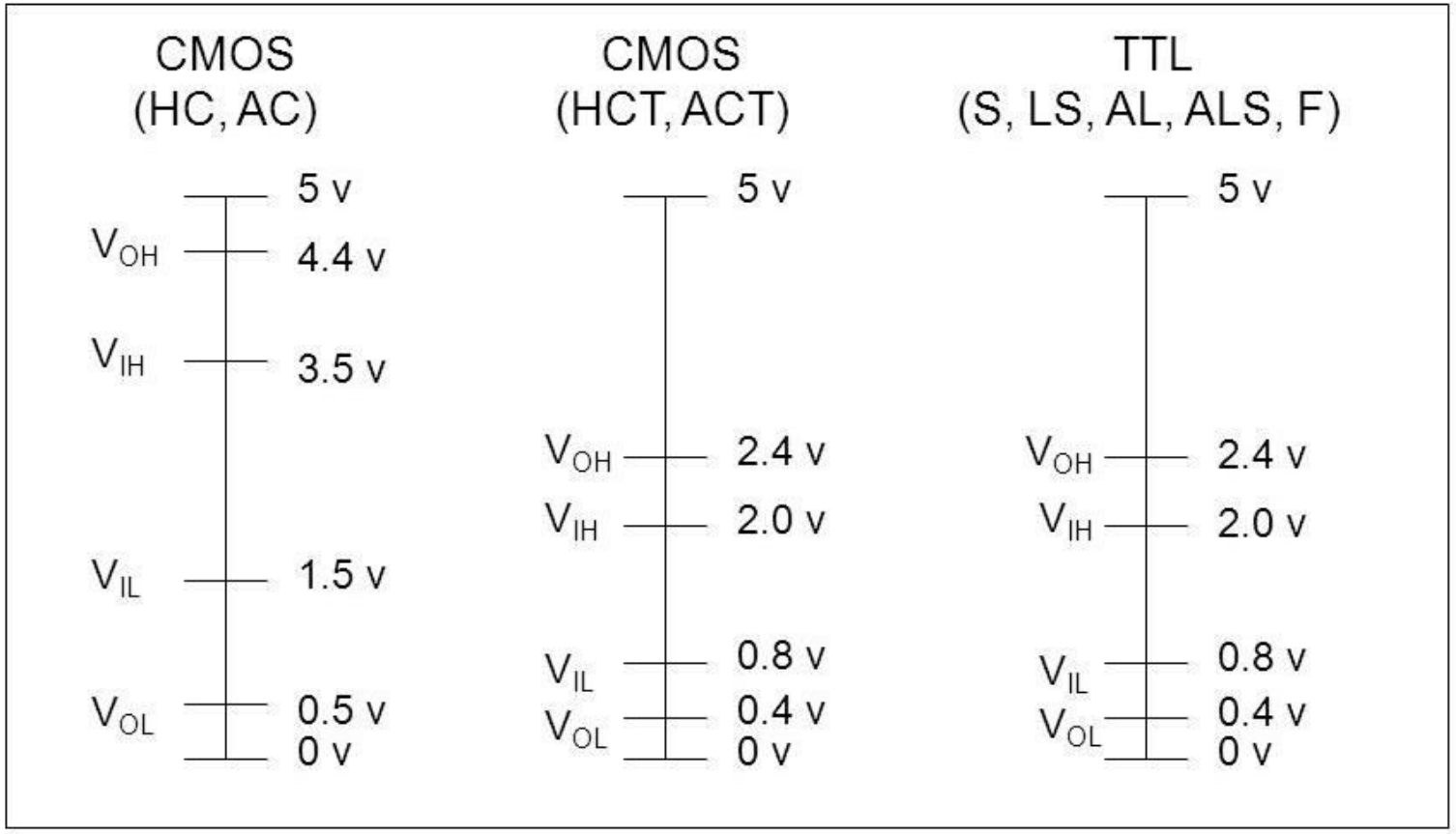


Ασύγχρονη μετάδοση χαρακτήρα

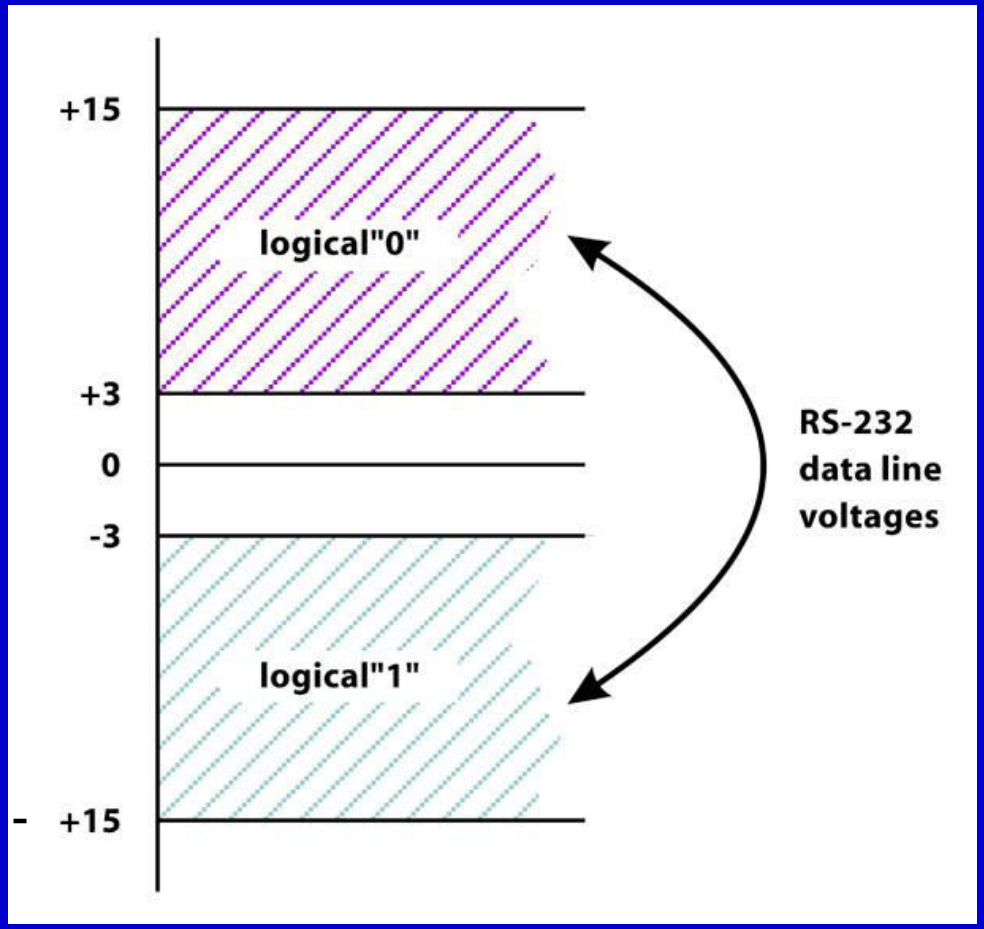
Στην *ασύγχρονη μετάδοση* (*asynchronous transmission*) μεταφέρεται ένας χαρακτήρας κάθε φορά. Ο χρονισμός πρέπει να διατηρείται μόνο κατά την διάρκεια της μετάδοσης ενός χαρακτήρα. Ο δέκτης έχει την δυνατότητα να επανασυγχρονίζεται στην αρχή κάθε νέου χαρακτήρα.



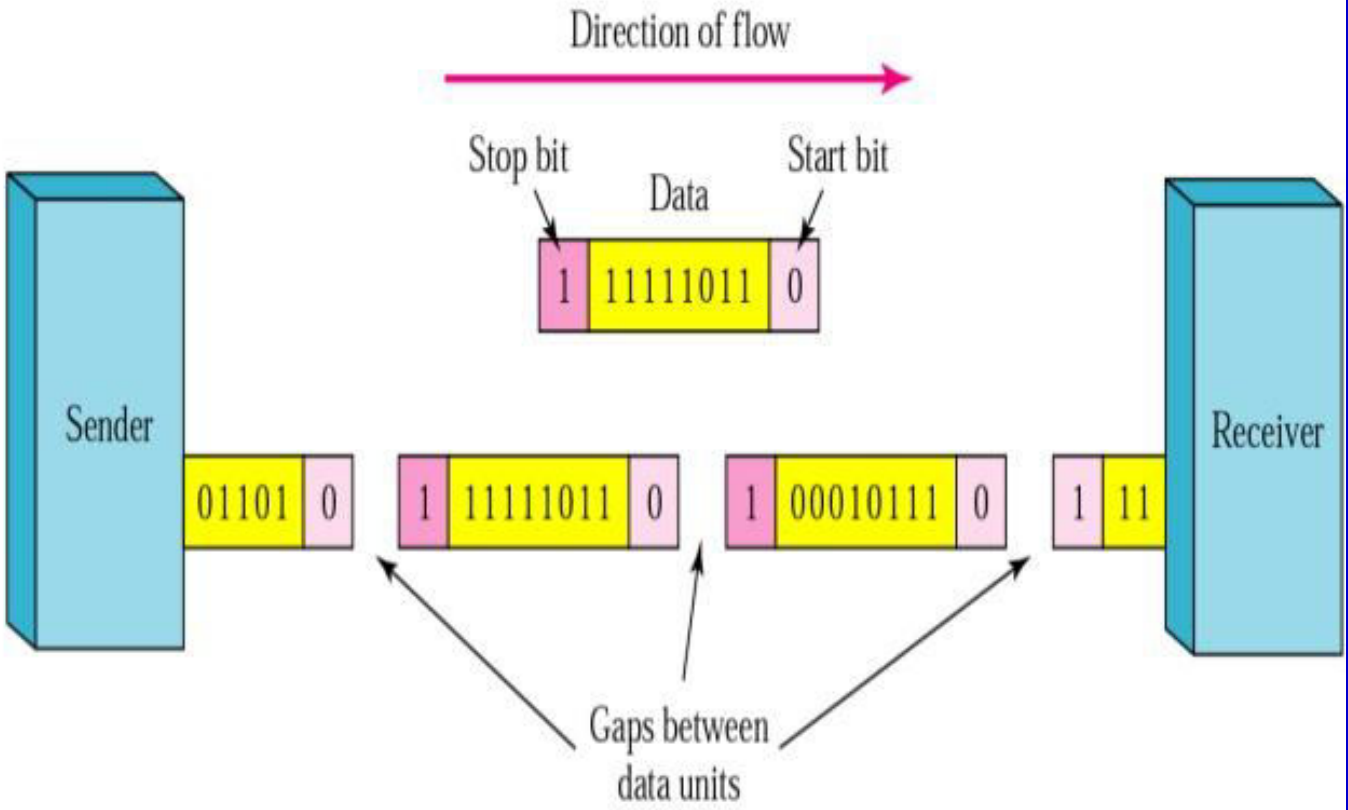
Comparison of Signal Levels



Κωδικοποίηση του 0 και του 1 σε RS-232



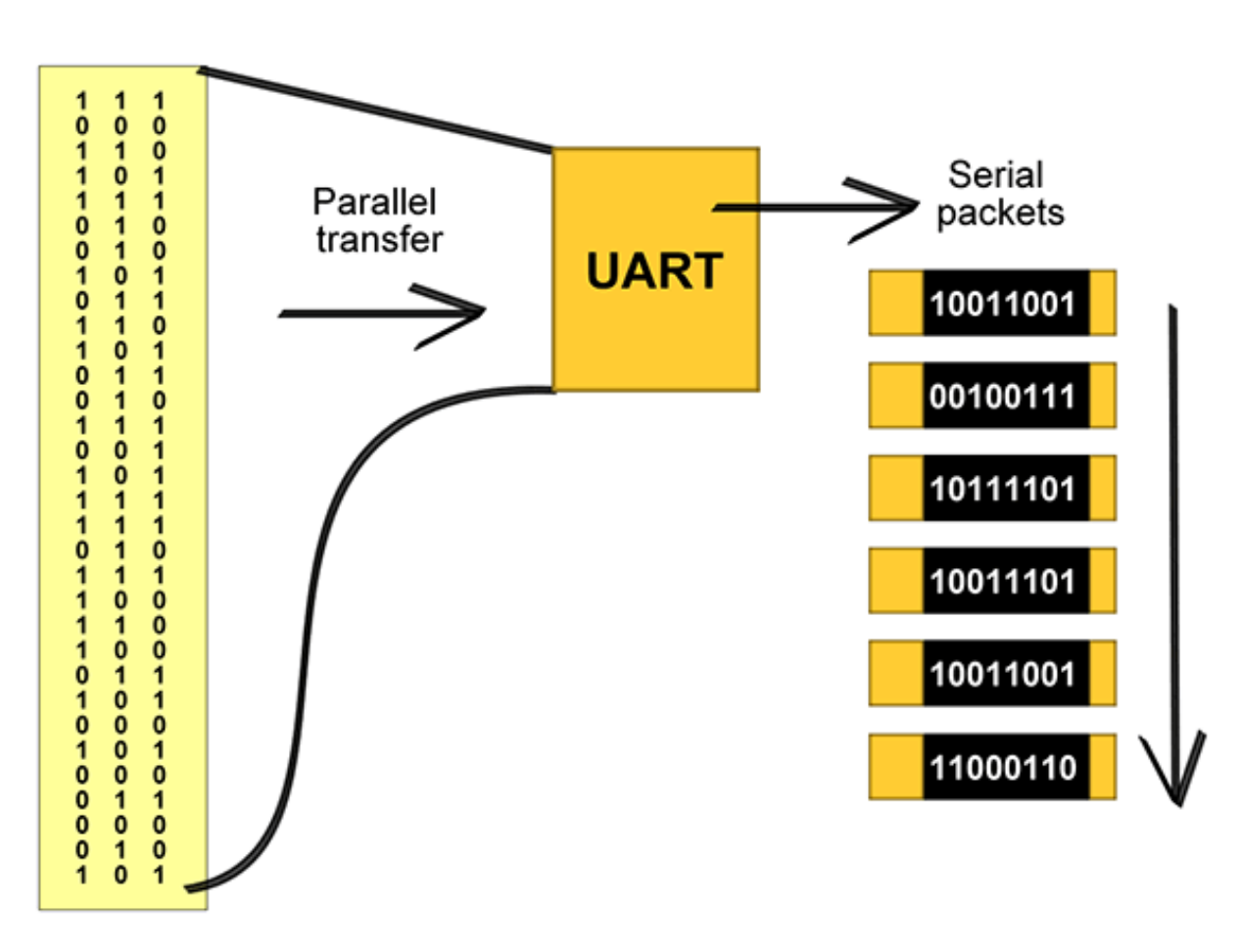
Asynchronous Transmission



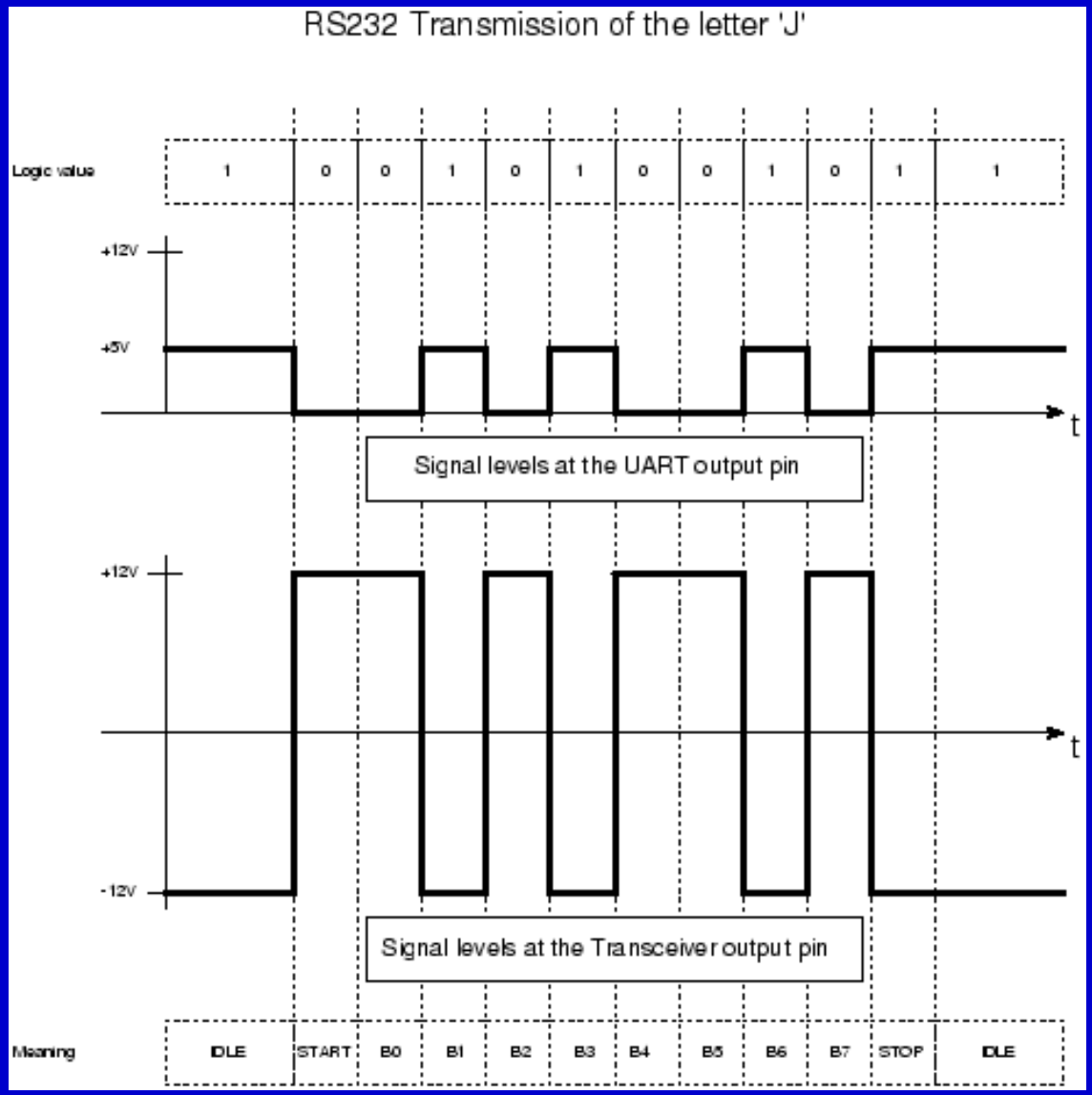
UART

A universal asynchronous receiver-transmitter (UART) is a computer hardware device for asynchronous serial communication in which the data format and transmission speeds are configurable. The electric signaling levels and methods are handled by a driver circuit external to the UART. A UART is usually an individual (or part of an) integrated circuit (IC) used for serial communications over a computer or peripheral device serial port. One or more UART peripherals are commonly integrated in microcontroller chips. A related device, the universal synchronous and asynchronous receiver-transmitter (USART) also supports synchronous operation.

Λειτουργία του UART

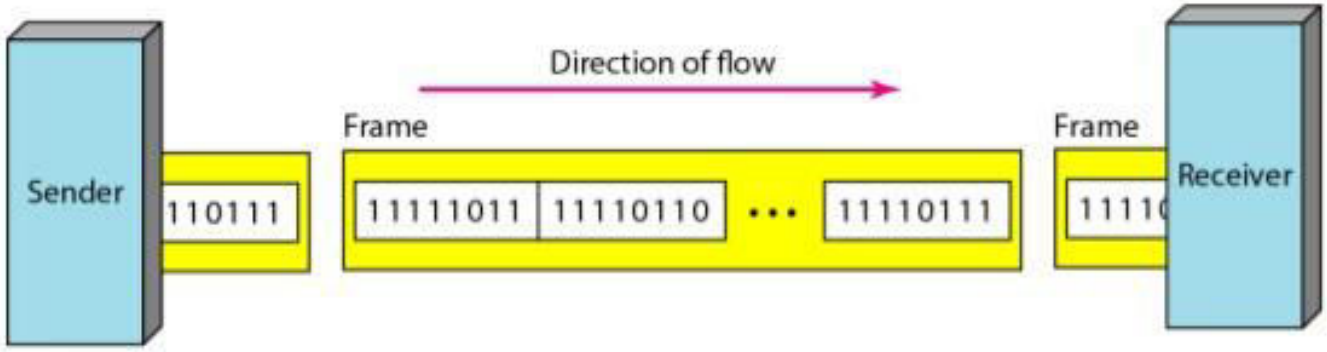


Σειριακή κωδικοποίηση χαρακτήρα σε TTL και RS-232



Synchronous transmission

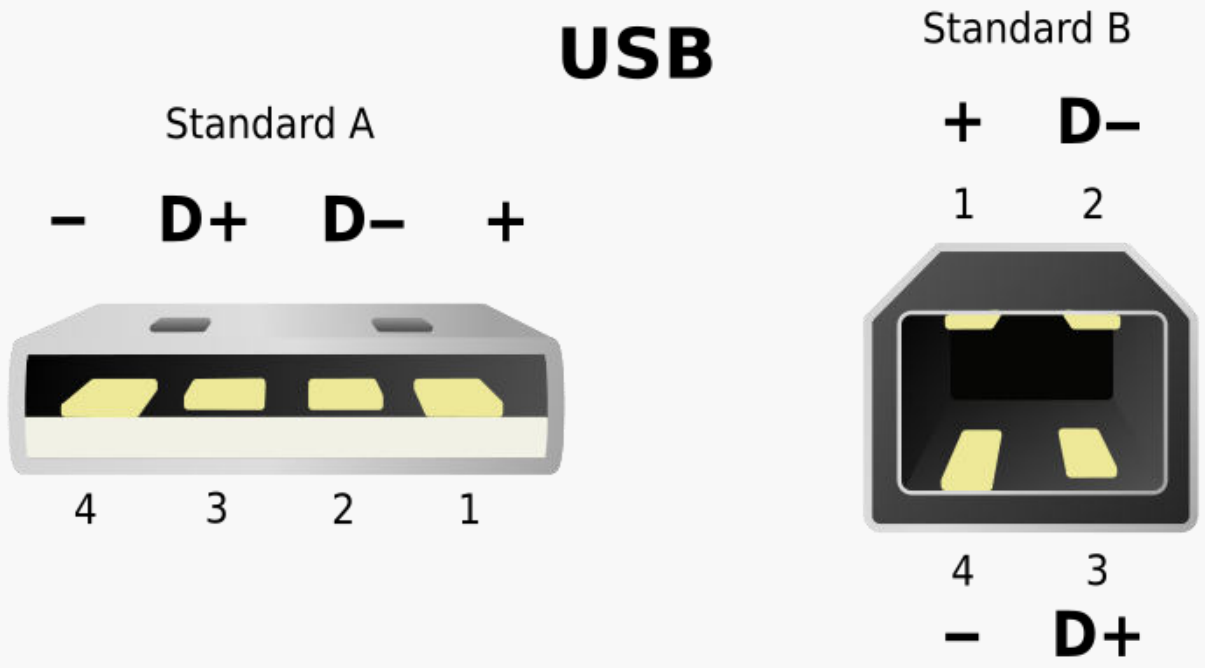
In *synchronous* transmission, we send bits one after another without start or stop bits or gaps. It is the responsibility of the receiver to group the bits.



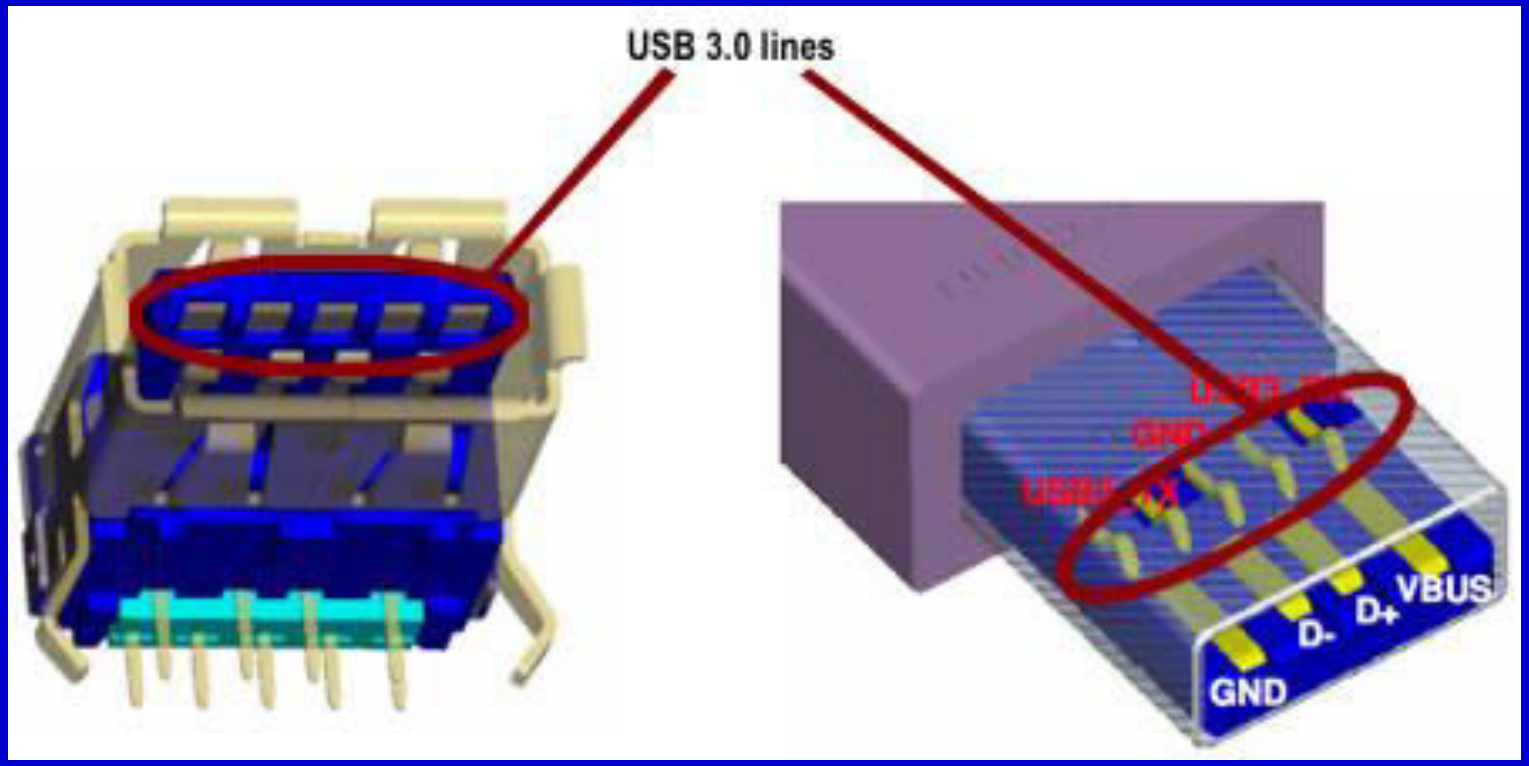
Δίαυλος USB

Ο δίαυλος *Universal Serial Bus* (USB), χρησιμοποιείται για την επικοινωνία των υπολογιστών με περιφερειακές μονάδες. Μεταφέρει επίσης και τροφοδοσία. Η μετάδοση των δεδομένων είναι **σύγχρονη**. Ο συνδετήρας του διαύλου USB απαιτεί σχετικά μικρό χώρο για την υλοποίησή του και μπορεί να παρέχει τροφοδοσία σε απλές συσκευές όπως ποντίκι, πληκτρολόγιο, κλπ. Με τη χρήση του διαύλου USB οι περιφερειακές συσκευές και τα χαρακτηριστικά τους αναγνωρίζονται αυτόματα.

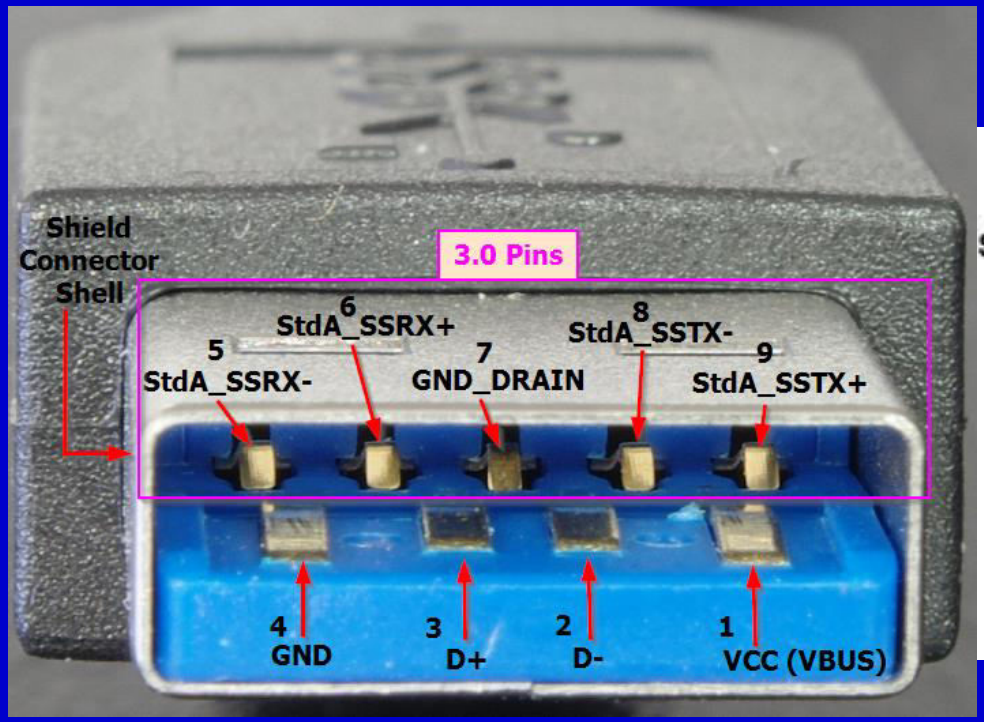
Ακροδέκτες USB

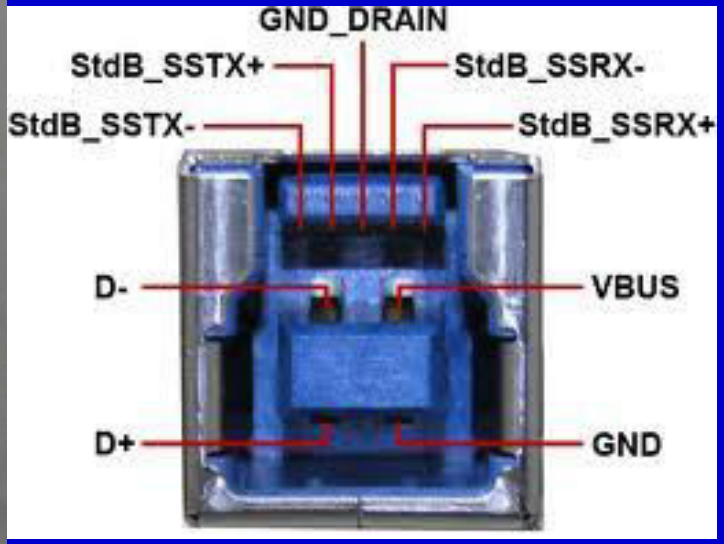


Θύρες USB 3.x

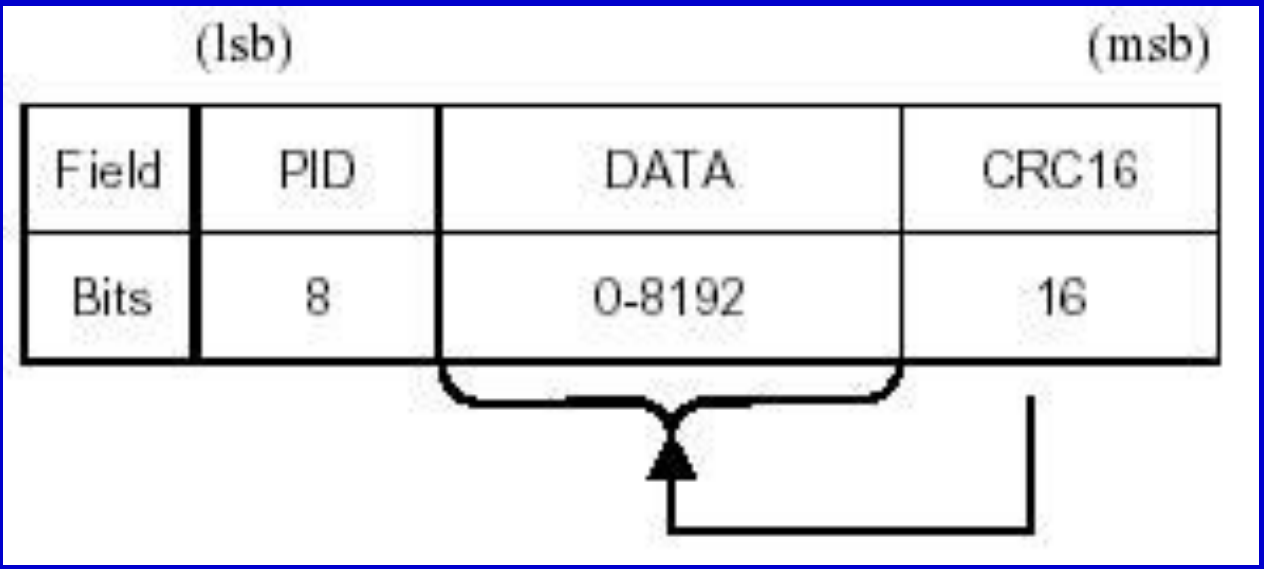


Θύρες USB 3.x

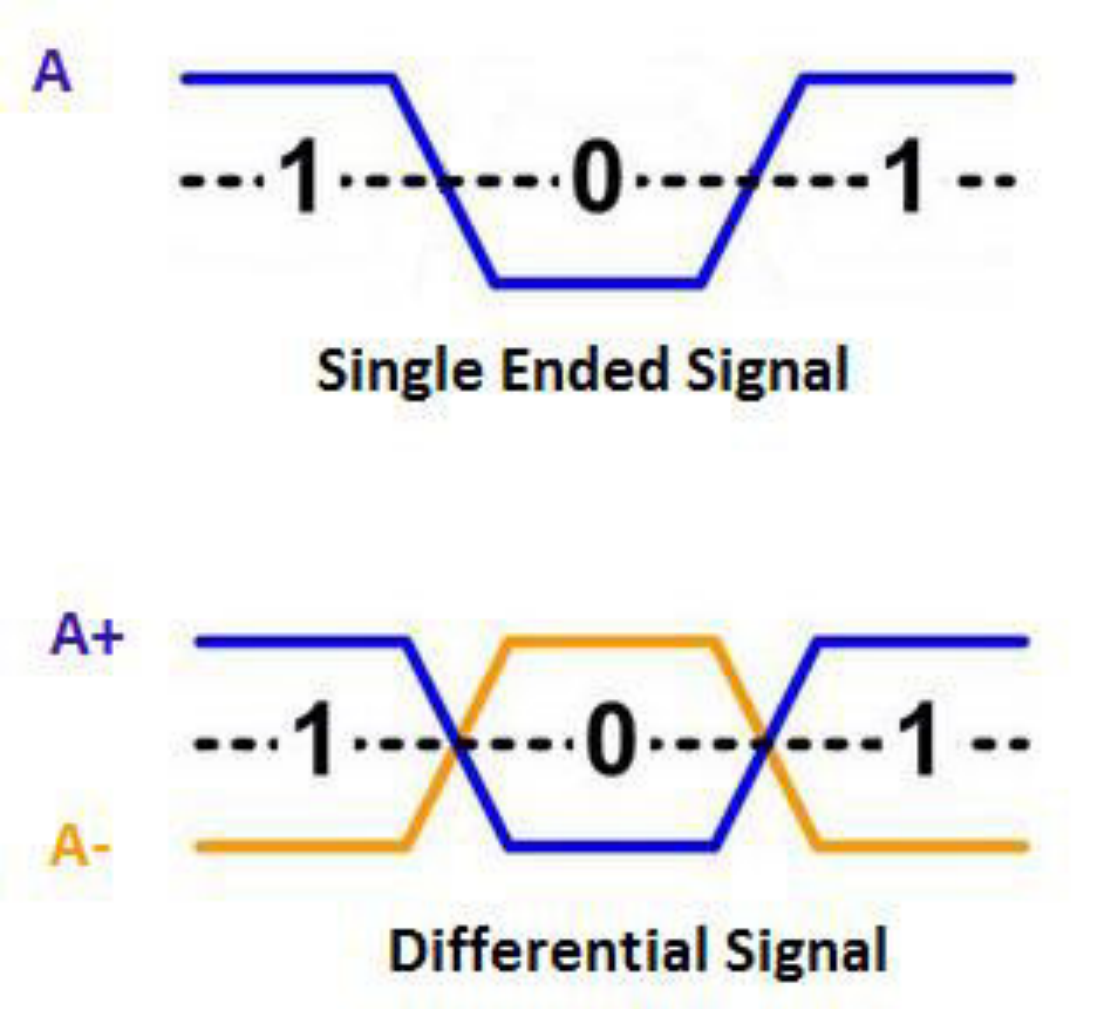

Shield
Connector
Shell
3.0 Pins
5
StdA_SSRX-
6
StdA_SSRX+
7
GND_DRAIN
8
StdA_SSTX-
9
StdA_SSTX+
4
GND
3
D+
2
D-
1
VCC (VBUS)


GND_DRAIN
StdB_SSTX+
StdB_SSRX-
StdB_SSTX-
StdB_SSRX+
D-
VBUS
D+
GND

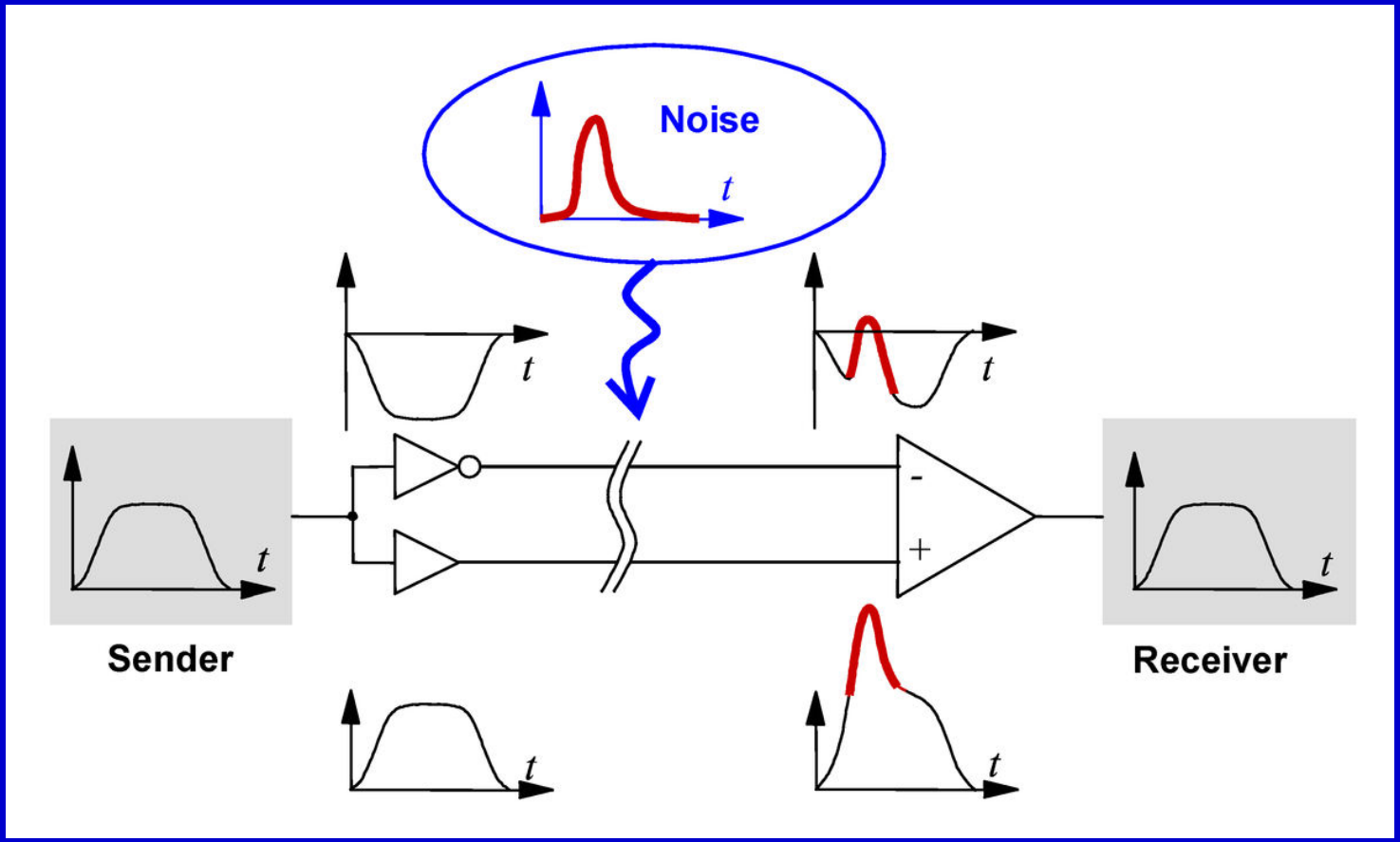
Η επικοινωνία στον δίαυλο USB είναι σύγχρονη και γίνεται με την χρήση πακέτων ελέγχου και πακέτων δεδομένων. Στην συνέχεια δίδεται η δομή των *πακέτων δεδομένων* του USB. (PID : Packet Identification)



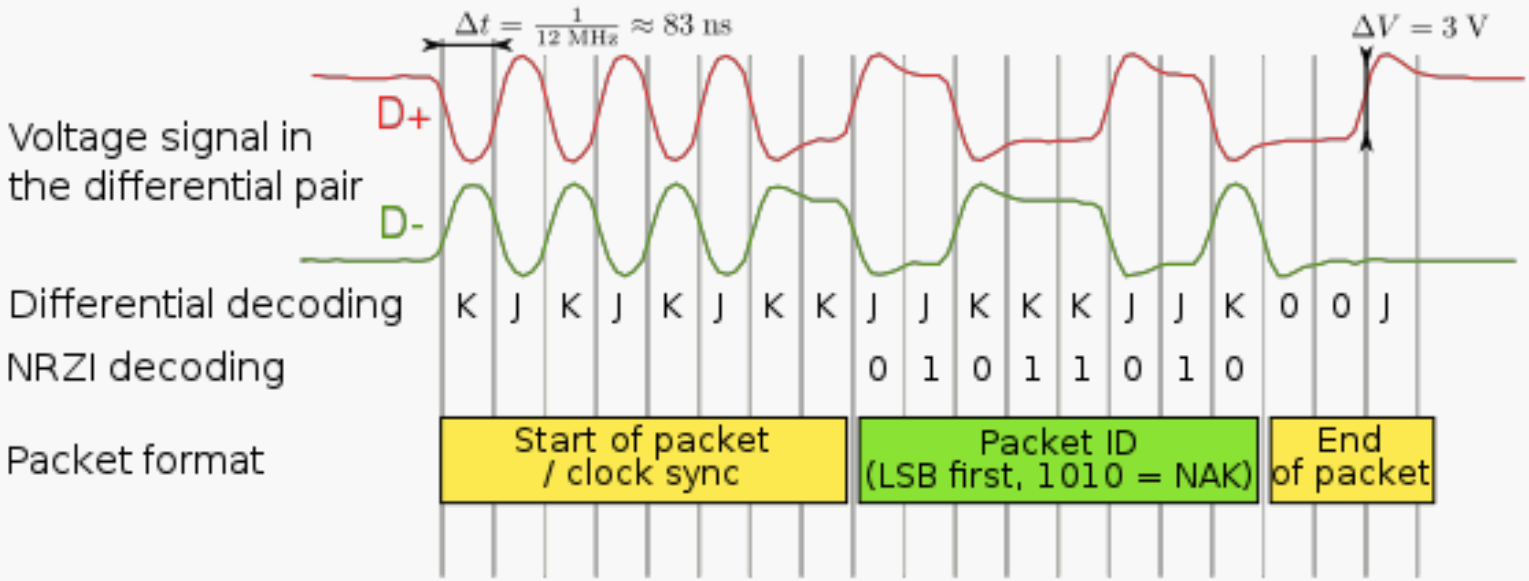
Απλή και διαφορική μετάδοση



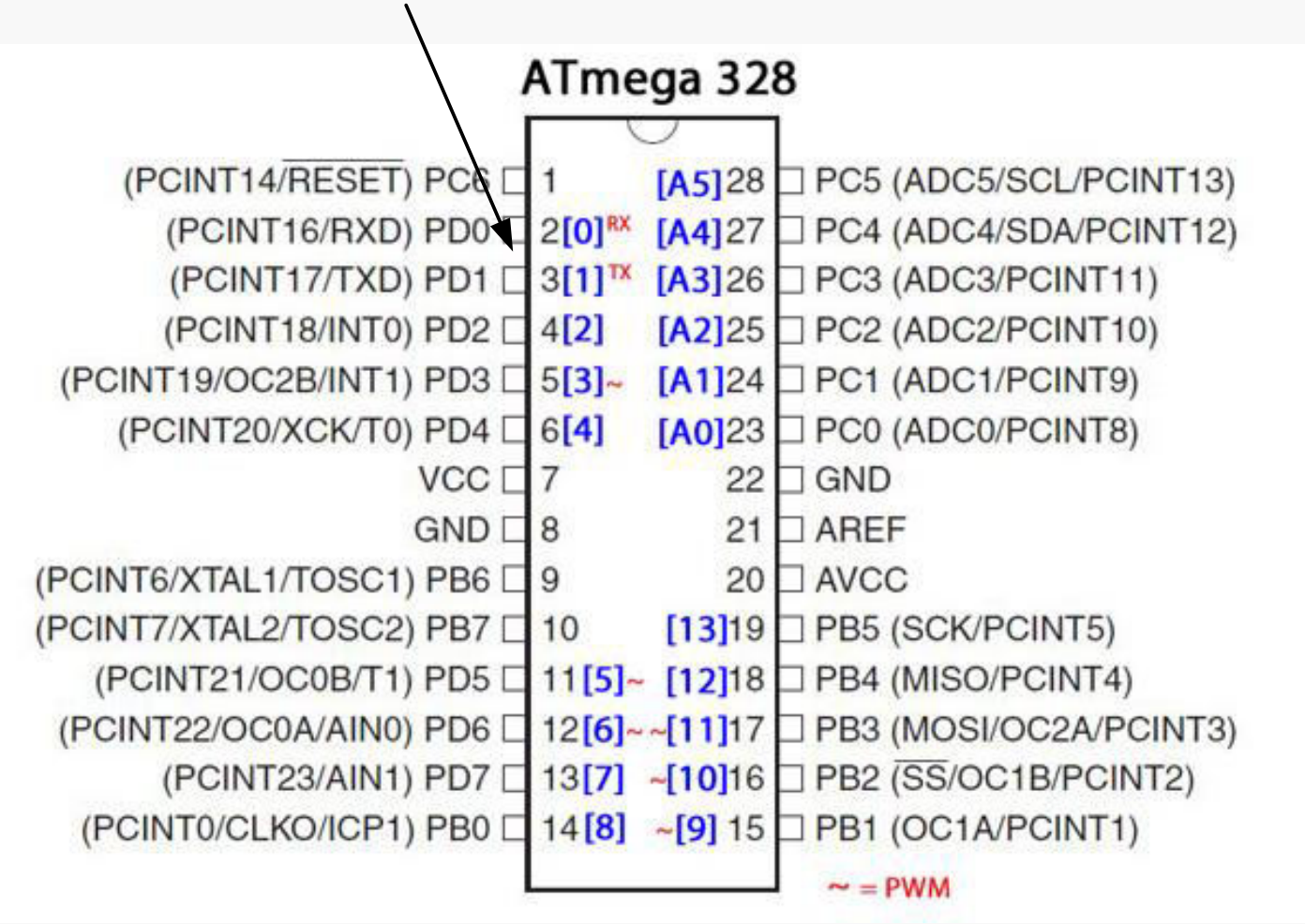
Differential transmission and noise



Negative Acknowledge packet transmitted by USB



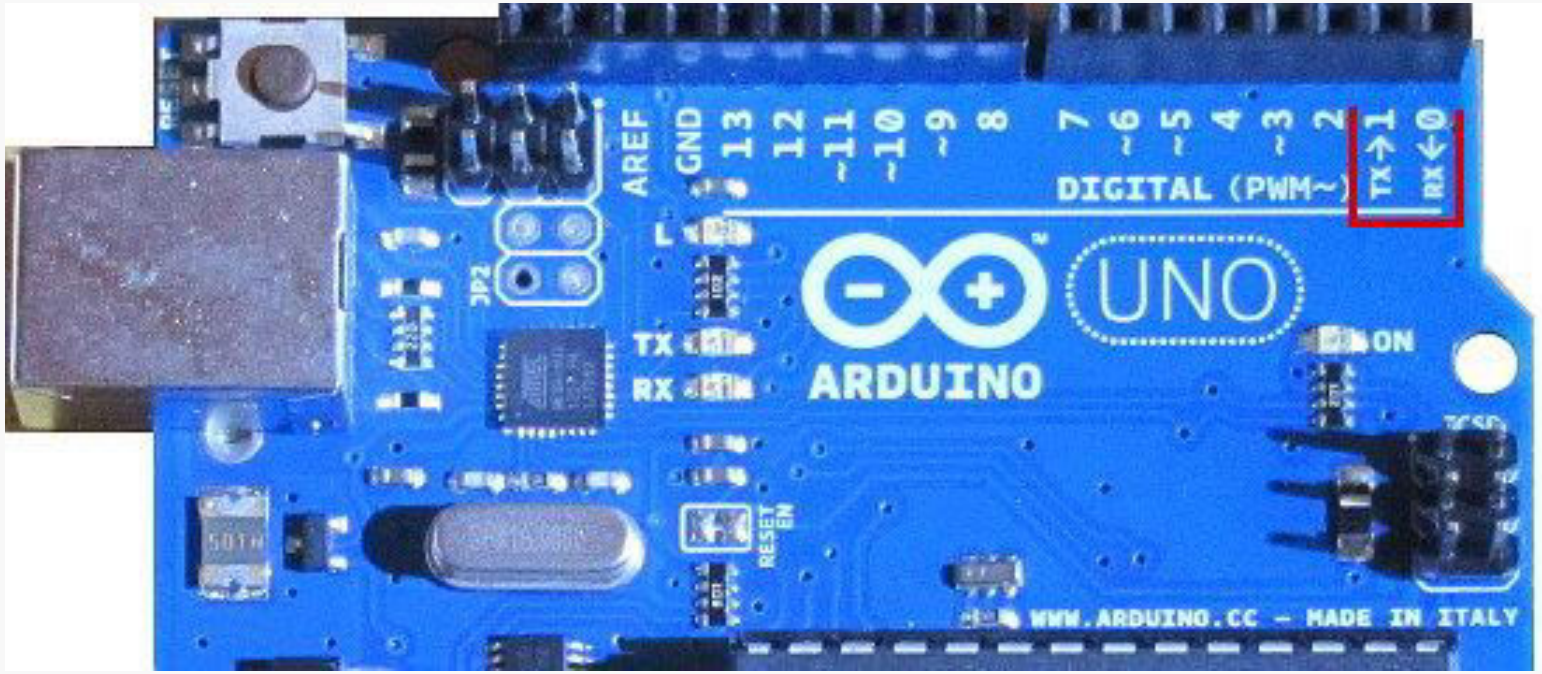
Σειριακές πόρτες του μικροελεγκτή AVR



Serial ports of Arduino

Serial Ports are used for communication between the Arduino board and a computer or other devices. All Arduino boards have at least one serial port (also known as a UART or USART)**:** Serial. It communicates on digital **pins 0 (RX) and 1 (TX)** as well as with the computer via USB.
Thus, if you use these functions, you cannot also use pins 0 and 1 for digital input or output.
Serial communication on pins TX/RX uses TTL logic levels (5V or 3.3V depending on the board).

Serial ports of Arduino UNO




serial_tx_msg | Arduino 1.0
File Edit Sketch Tools Help
Serial Monitor ----->
serial_tx_msg
/*----------
Program: serial_tx_msg
(serial transmit message)
Description: Sends a text message out of the serial (USB)
port of the Arduino every second.
serial_tx_msg | Arduino 0022
File Edit Sketch Tools Help
<----- Serial Monitor
serial_tx_msg
Program: serial_tx_msg
(serial transmit message)
Description: Sends a text message out of the serial (USB)
port of the Arduino every second.

Εντολές σειριακής εισόδου/εξόδου του Arduino

Serial.begin(speed)

serial.begin sets the data rate in bits per second (baud) for serial data transmission.
For communicating with the computer, use one of these rates: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, or 115200. You can, however, specify other rates, for example, to communicate over pins 0 and 1 with a component that requires a particular baud rate.

Serial.Read()

Serial.Read function reads incoming serial data.

serial.print()

serial.print prints data to the serial port as human-readable ASCII text. This command can take many forms. Numbers are printed using an ASCII character for each digit.
Floats are similarly printed as ASCII digits, defaulting to two decimal places.
Bytes are sent as a single character.

Serial.print(val)
Serial.print(val, format)

Serial.println()

Prints data to the serial port as human-readable ASCII text followed by a carriage return character (ASCII 13, or '\r') and a newline character (ASCII 10, or '\n'). This command takes the same forms as Serial.print().

Serial.println(val)
Serial.println(val, format)

val : the value to print - any data type
format: specifies the number base (for integral data types)
or number of decimal places (for floating point types)

Serial.print(78) gives "78"
Serial.print(1.23456) gives "1.23"
Serial.print('N') gives "N"
Serial.print("Hello world") gives "Hello world"
Serial.print(78, BIN) gives "1001110"
Serial.print(78, OCT) gives "116"
Serial.print(78, DEC) gives "78"
Serial.print(78, HEX) gives "4E"
Serial.println(1.23456, 0) gives "1"
Serial.println(1.23456, 2) gives "1.23"
Serial.println(1.23456, 4) gives "1.2346"

Serial.write(val)
Serial.write(str)
Serial.write(buf, len)

Writes binary data to the serial port. This data is sent as a byte or series of bytes; to send the characters representing the digits of a number use the print() function instead.

val: a value to send as a single byte
str: a string to send as a series of bytes

buf: an array to send as a series of bytes
len: the length of the buffer

Serial.available()

serial.available get the number of bytes (characters) available for reading from the serial port. This is data that's already arrived and stored in the serial receive buffer (which holds 64 bytes).

Serial.parseInt()

Looks for the next valid integer in the incoming serial stream.

In particular:
Initial characters that are not digits or a minus sign, are skipped;
Parsing stops when no characters have been read for a configurable time-out value, or a non-digit is read;
If no valid digits were read when the time-out (see Serial.setTimeout()) occurs, 0 is returned;

Πρόγραμμα που διαβάζει τα δεδομένα από την σειριακή είσοδο και τα στέλνει στην σειριακή έξοδο.

```
int val
void setup() {
        Serial.begin(9600); // opens serial port,
                                    //sets data rate to 9600 bps
}
void loop() {
       // send data only when you receive data:
        if (Serial.available() > 0)
        {
              // read the incoming byte:
                incomingByte = Serial.read();
              // say what you got:
                Serial.print("I received: ");
                Serial.println(incomingByte, DEC);
        }
}
```

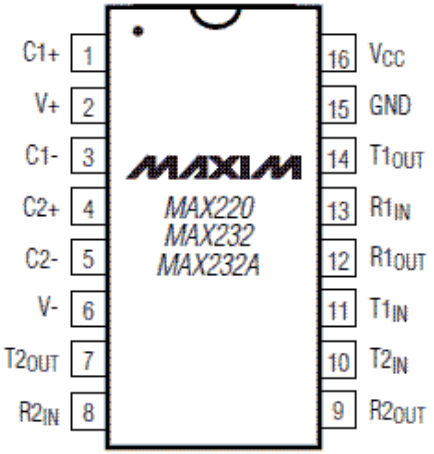
Arduino Zero USB ports

The Arduino Zero has two USB ports available. The *Native* USB port (which supports CDC serial communication using the *SerialUSB* object) is connected directly to the SAMD21 MCU. The other USB port is the *Programming* port. It is connected to the ATMEL embedded debugger (EDBG), the onboard programmer and debugger which can also acts as a USB-to-Serial converter. This Programming port is the default for uploading sketches and communicating with the board.

The USB-to-serial converter of the Programming port is connected to the first UART of the SAMD21. It's possible to communicate over this port using the "Serial" object in the Arduino programming language.

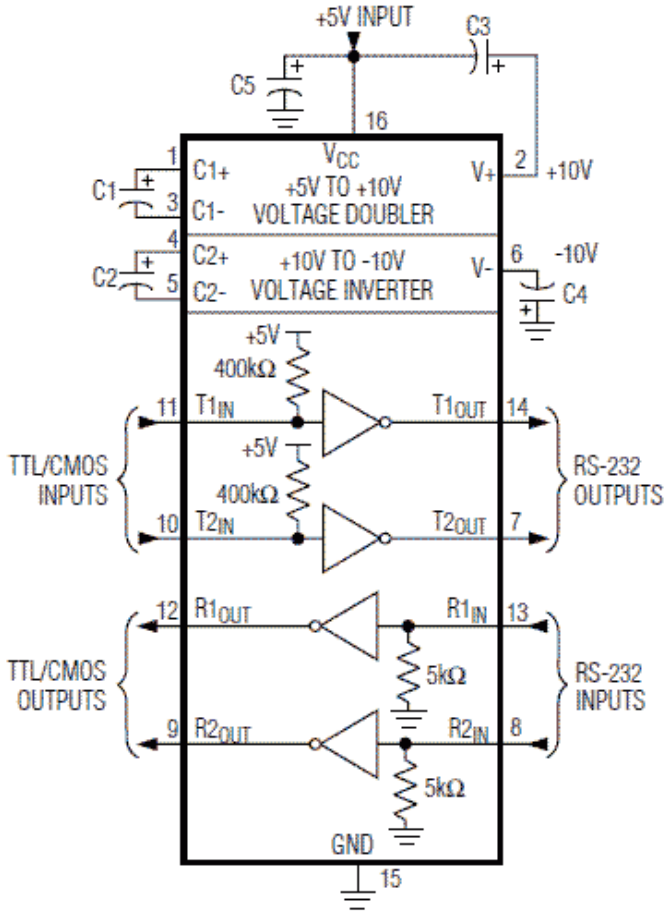
Ολοκληρωμένο μετατροπής σήματος TTL σε RS232 και αντίστροφα

TOP VIEW



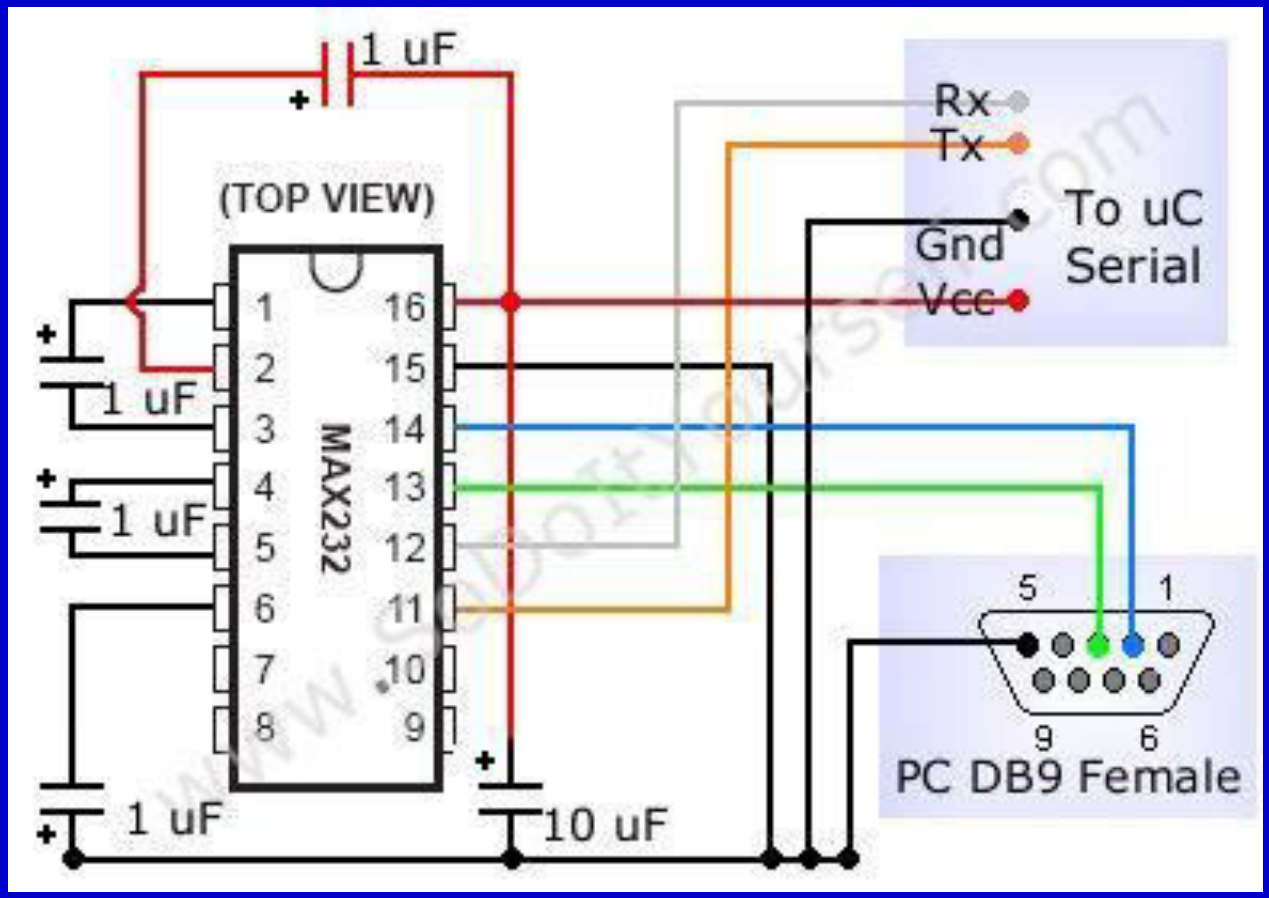
DIP/SO

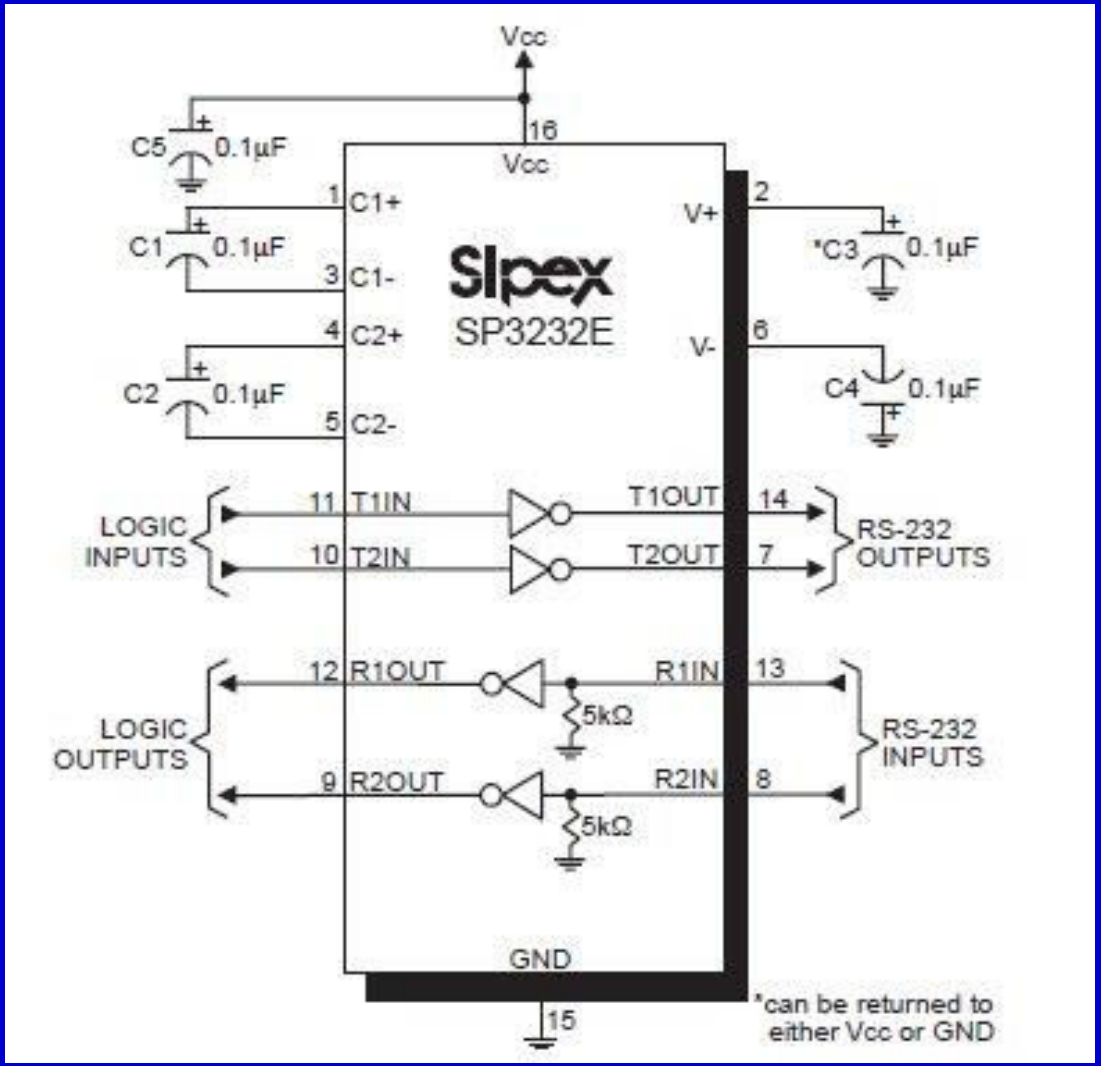
CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	0.047	0.33	0.33	0.33	0.33
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1



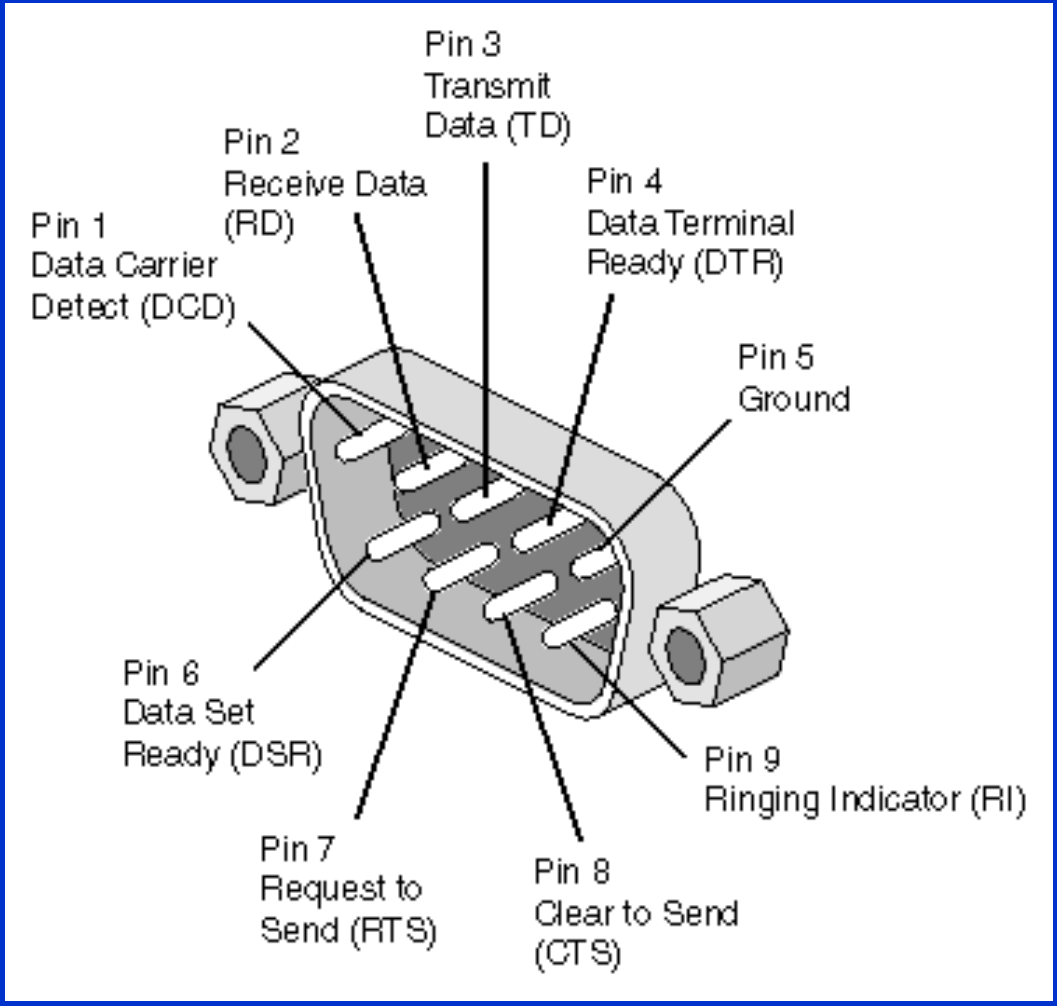
Diagrams continued in the full data sheet.

Μετατροπή σημάτων TTL σε RS232



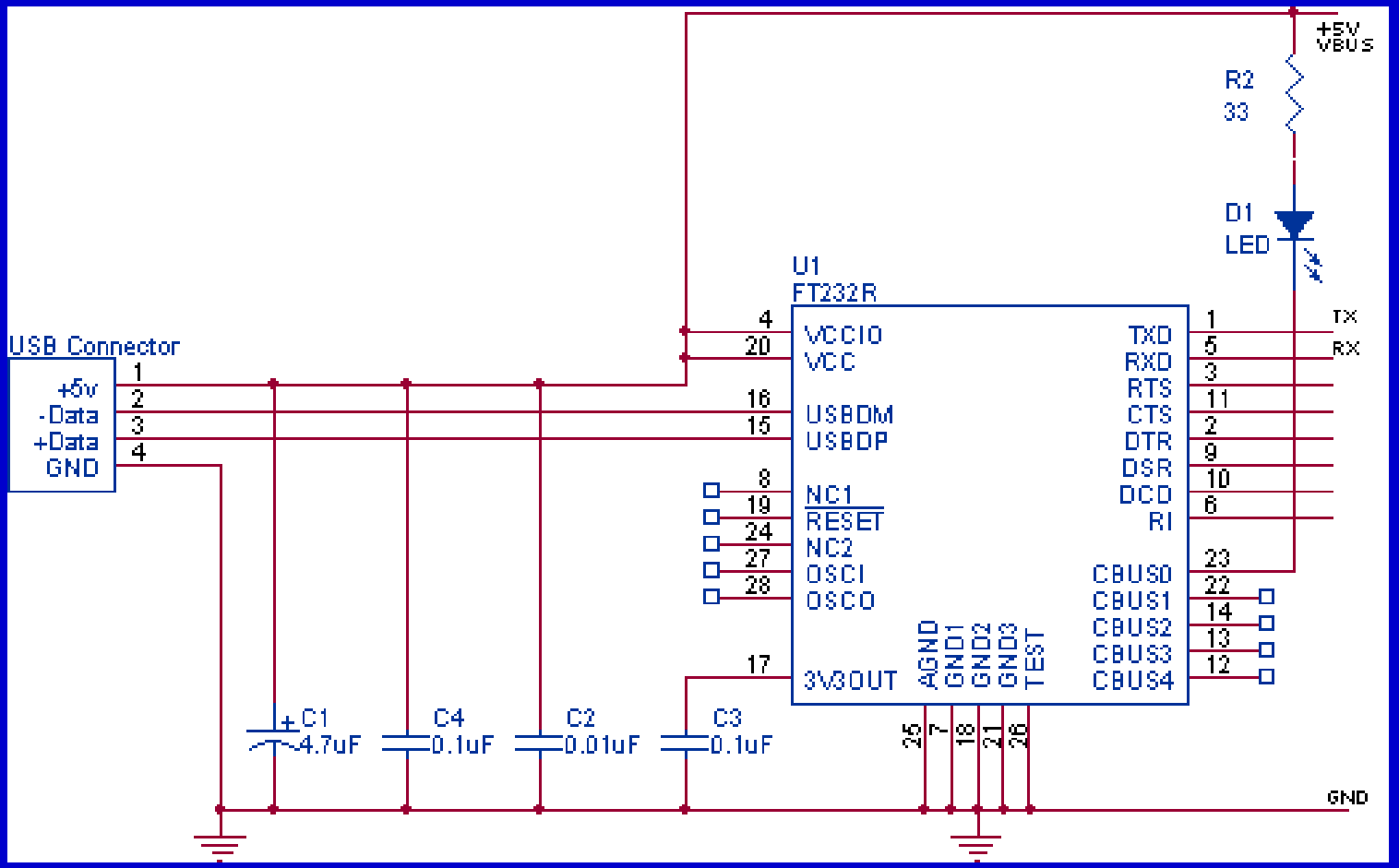

Vcc
16
Vcc
C5
0.1µF
1
C1+
C1
0.1µF
3
C1-
4
C2+
C2
0.1µF
5
C2-
Sipex
SP3232E
V+
2
*C3
0.1µF
V-
6
C4
0.1µF
LOGIC
INPUTS
11
T1IN
10
T2IN
T1OUT
14
T2OUT
7
RS-232
OUTPUTS
LOGIC
OUTPUTS
12
R1OUT
9
R2OUT
R1IN
13
R2IN
8
5kΩ
5kΩ
RS-232
INPUTS
GND
15
*can be returned to
either Vcc or GND

COM port

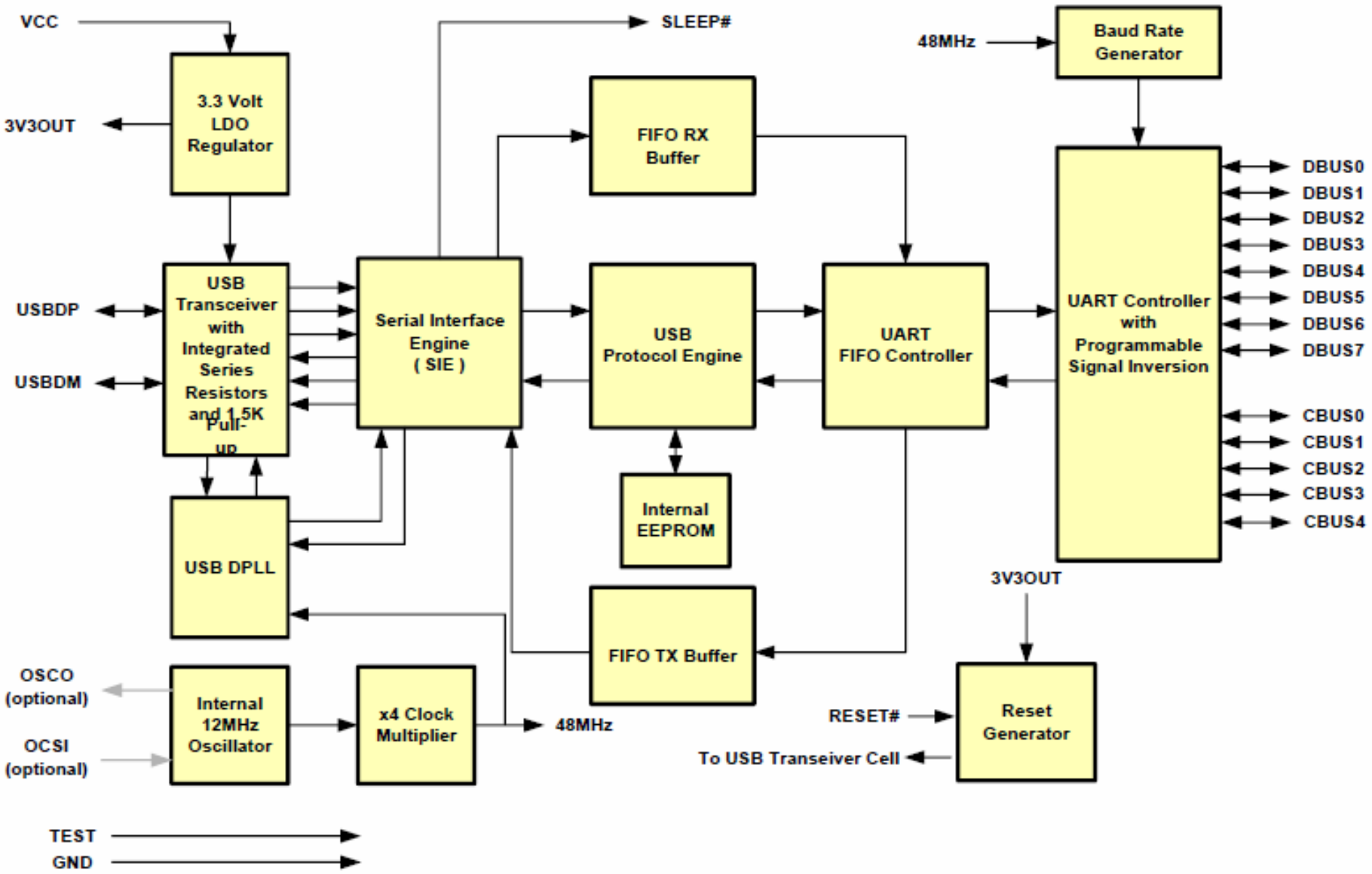

Pin 3
Transmit
Data (TD)
Pin 2
Receive Data
(RD)
Pin 4
Data Terminal
Ready (DTR)
Pin 1
Data Carrier
Detect (DCD)
Pin 5
Ground
Pin 6
Data Set
Ready (DSR)
Pin 9
Ringing Indicator (RI)
Pin 7
Request to
Send (RTS)
Pin 8
Clear to Send
(CTS)

Μετατροπέας από TTL σε RS232

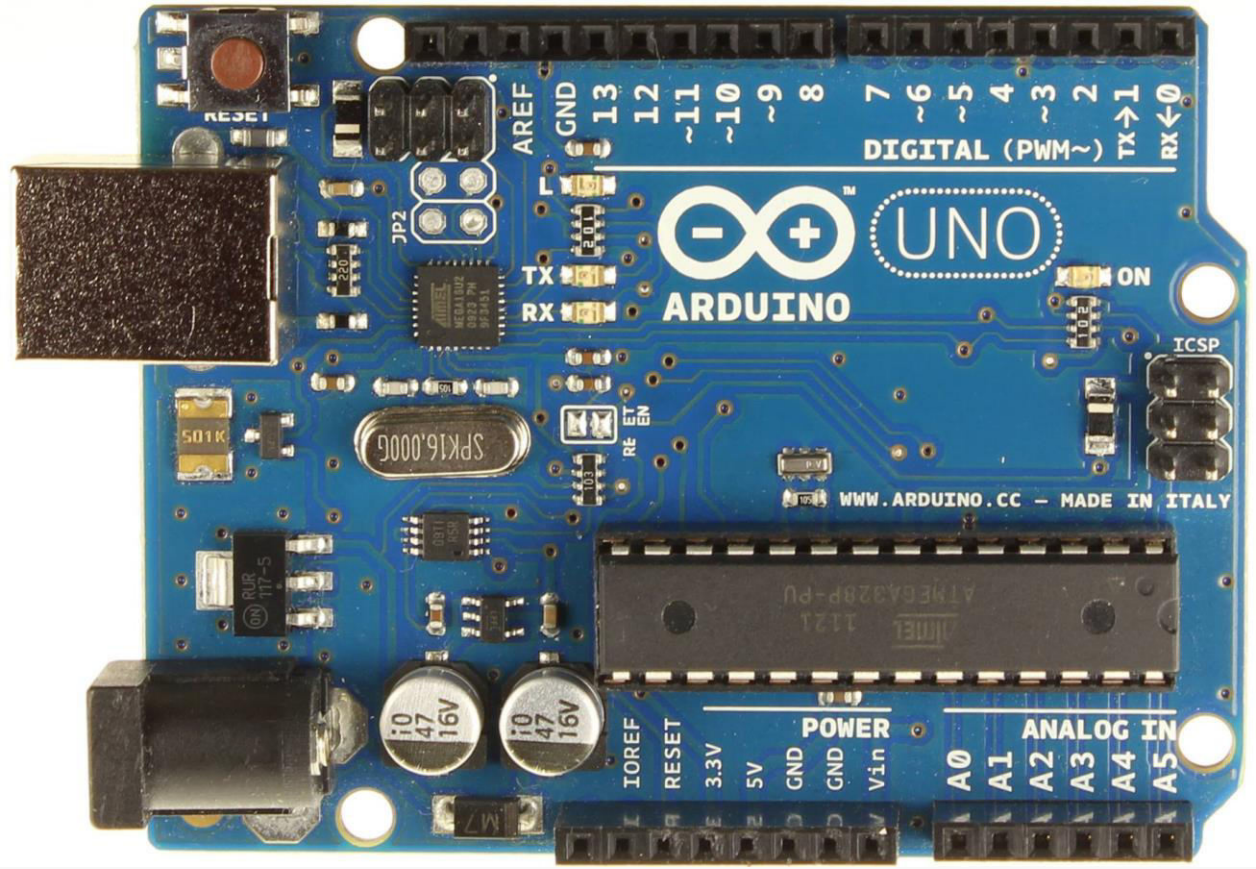
Μετατροπή σειριακών σημάτων TTL σε USB



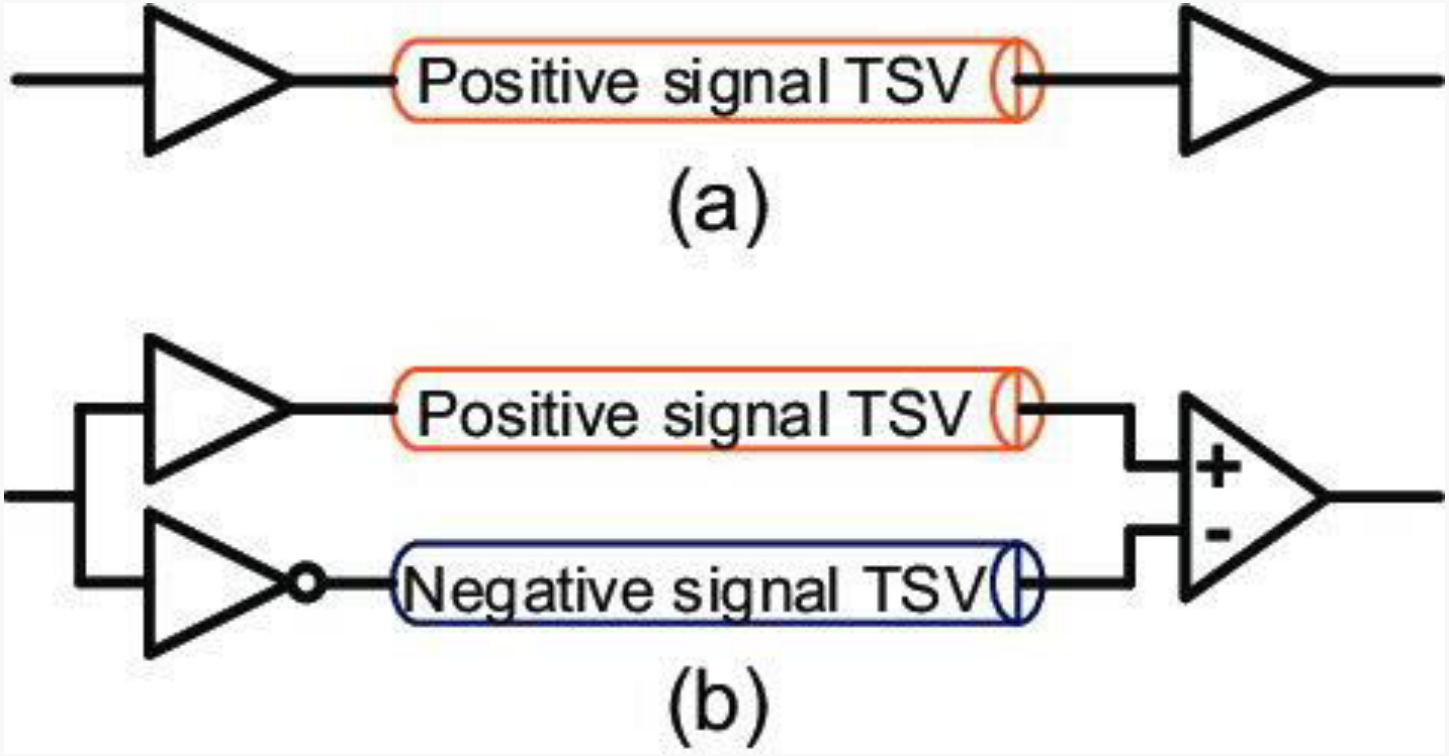
FT232R block diagram



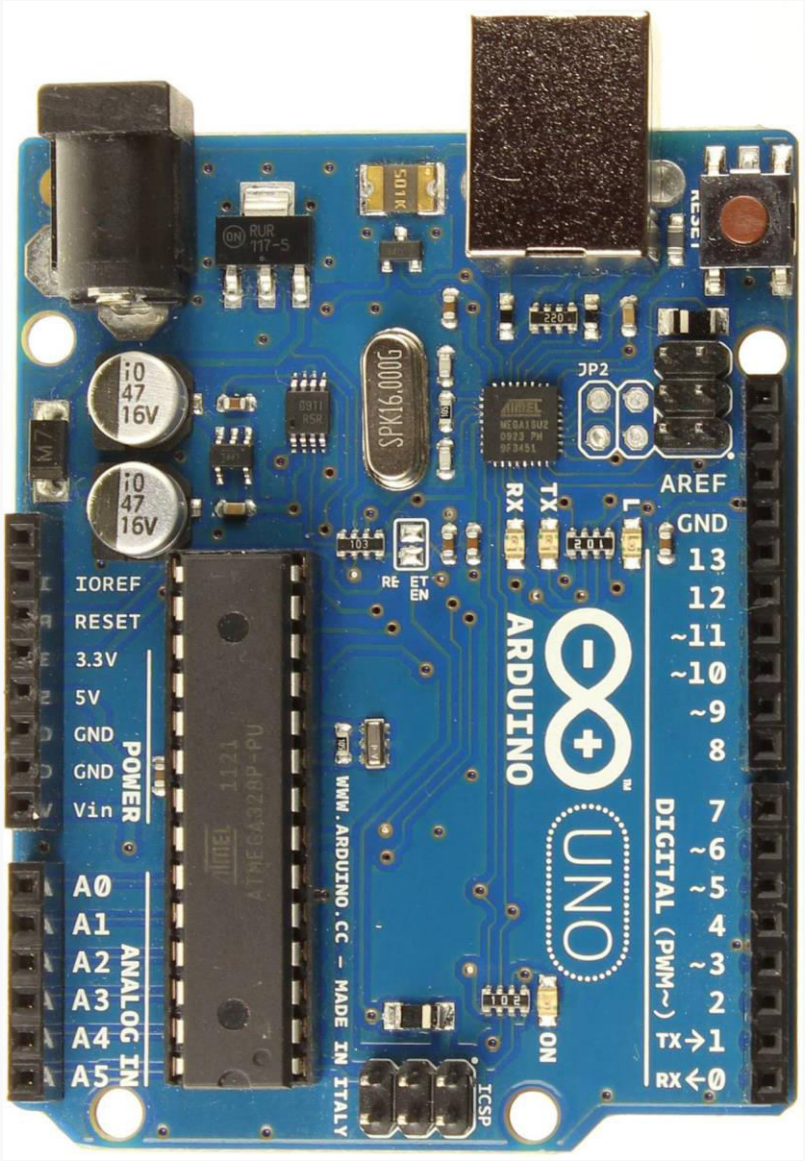
1. Στο σύστημα Arduino που δίδεται στην συνέχεια εντοπίστε τις σειριακές εισόδους/εξόδους, τα ενδεικτικά LED και το ολοκληρωμένο κύκλωμα που κάνει την μετατροπή των ασύγχρονων σειριακών εισόδων/εξόδων σε USB.

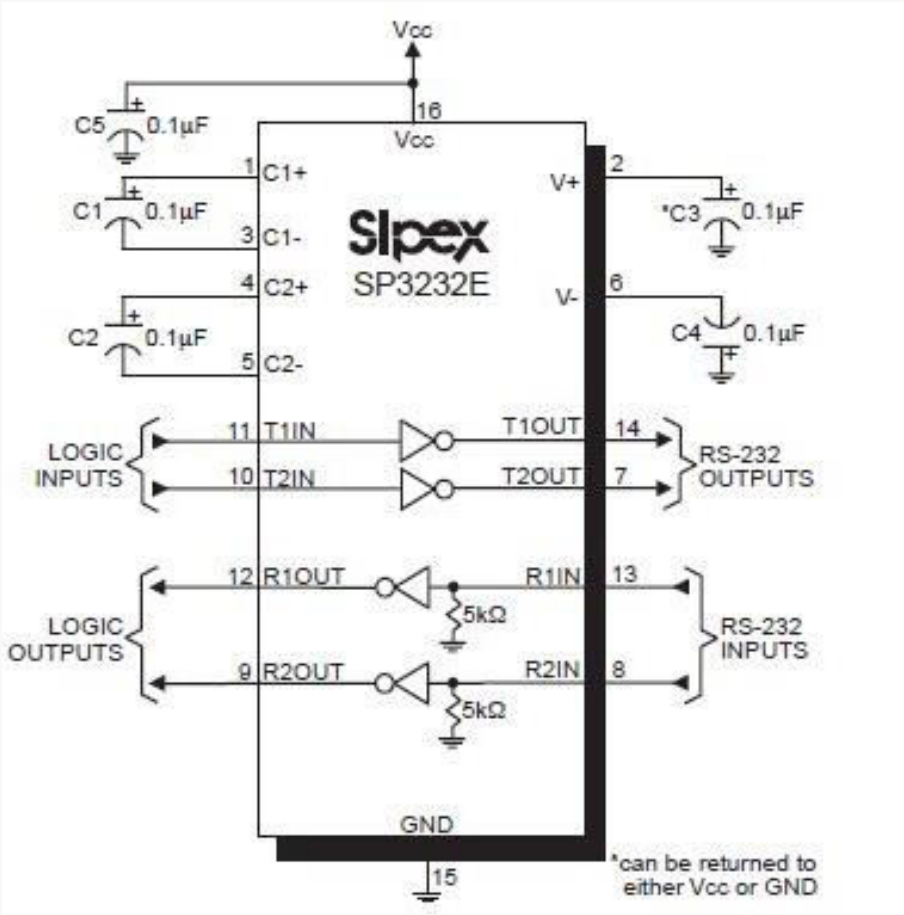


2. Από τις μεταδόσεις που περιγράφονται στην συνέχεια ποια είναι single ended και ποια differential.



3. Κάντε τις κατάλληλες συνδέσεις μεταξύ του Arduino Uno και του ολοκληρωμένου SP3232E για να δημιουργήσετε μία σειριακή πόρτα RS232


ARDUINO
UNO
POWER
ANALOG IN
DIGITAL (PWM~)
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
ICSP
ON
WWW.ARDUINO.CC – MADE IN ITALY


Vcc
16
Vcc
C5
0.1µF
1
C1+
C1
0.1µF
3
C1-
4
C2+
C2
0.1µF
5
C2-
Sipex
SP3232E
V+
2
*C3
0.1µF
V-
6
C4
0.1µF
LOGIC
INPUTS
11
T1IN
10
T2IN
T1OUT
14
T2OUT
7
RS-232
OUTPUTS
LOGIC
OUTPUTS
12
R1OUT
9
R2OUT
R1IN
13
R2IN
8
RS-232
INPUTS
5kΩ
5kΩ
GND
15
*can be returned to
either Vcc or GND