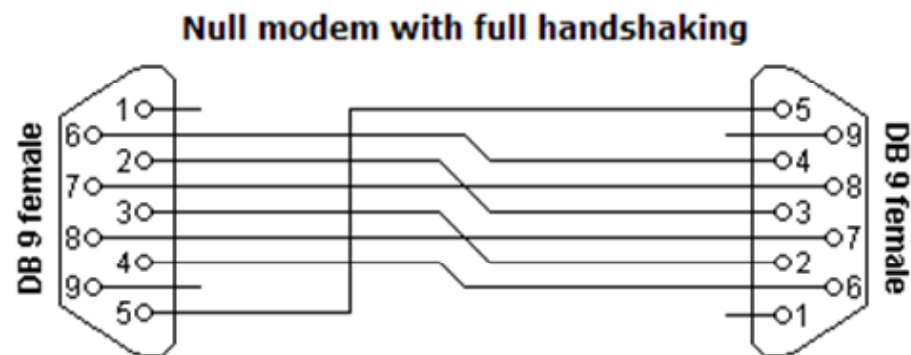


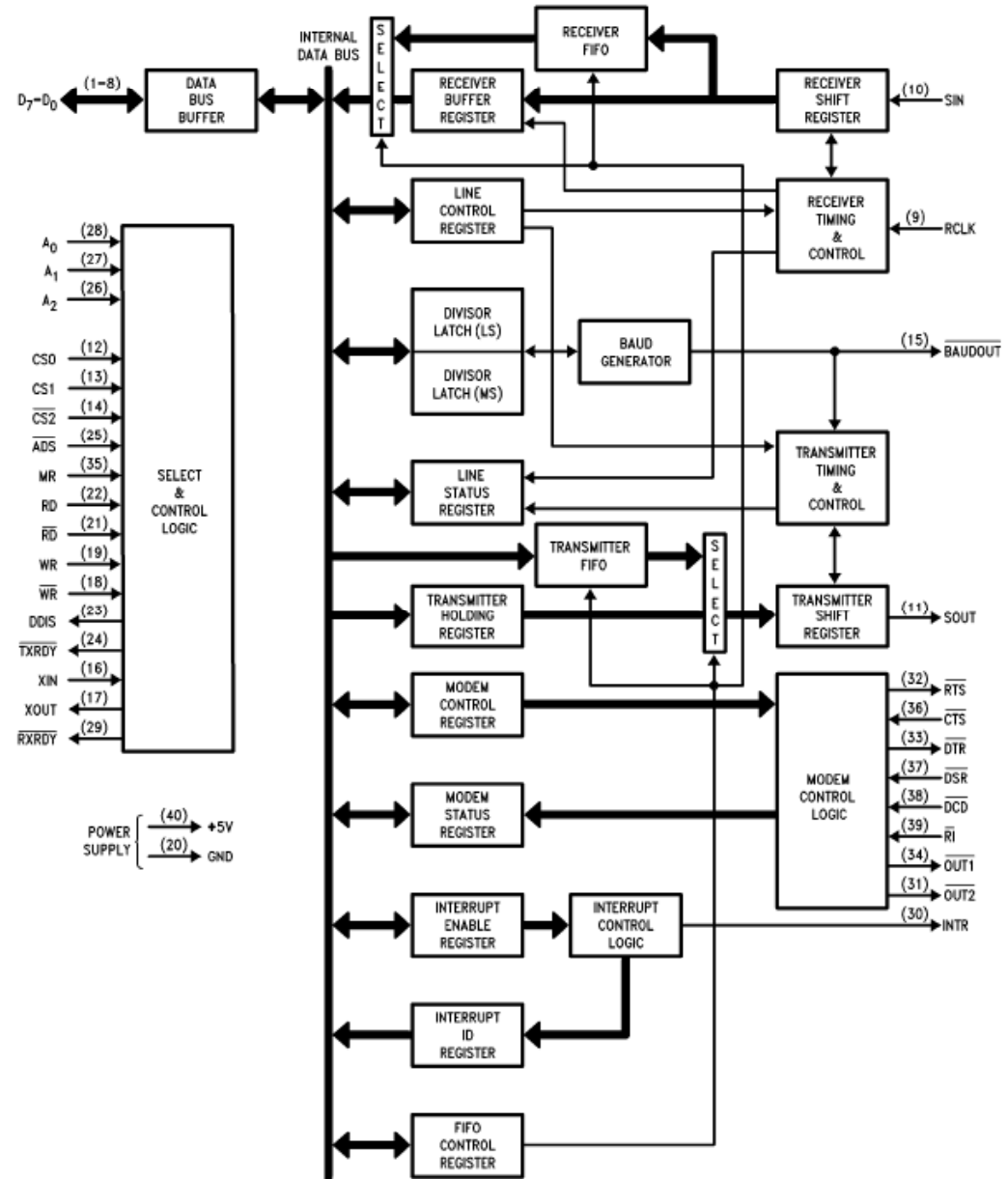
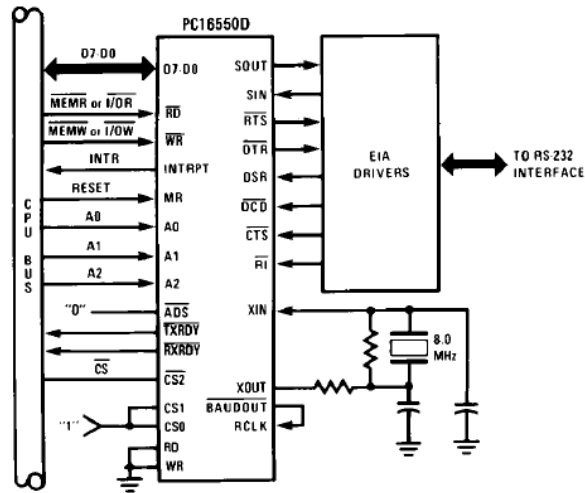
# RS232

- Ασύγχρονη σειριακή μετάδοση
- Start, stop και parity bits
- Περιορισμένη ταχύτητα μετάδοσης



Connector 1	Connector 2	Function
2	3	Rx ← Tx
3	2	Tx → Rx
4	6	DTR → DSR
5	5	Signal ground
6	4	DSR ← DTR
7	8	RTS → CTS
8	7	CTS ← RTS

# PC16550D Universal Asynchronous Receiver/Transmitter with FIFOs†



## Επικοινωνία Matlab μοντέλων μέσω RS-232

**% Serial Port Initialization**

```
s_port = serial('COM14', 'BaudRate', 57600, 'Parity','none');  
fopen(s_port);
```

**% Read/Set properties**

```
get(s_port,'OutputBufferSize');
```

```
set(s_port,'BaudRate',4800);
```

**% Read/Write Data**

```
len_in = s_port.BytesAvailable;  
data_in = fread(s_port,1);
```

Ενημέρωση για το πόσοι χαρακτήρες είναι διαθέσιμοι στον *buffer* εισόδου

Ανάγνωση ενός χαρακτήρα από τον *buffer* εισόδου

```
len_out = s_port.BytesToOutput;  
fwrite(s_port, data_out, 'uchar');
```

Ενημέρωση για το πόσοι χαρακτήρες είναι προς μετάδοση στον *buffer* εξόδου

Εγγραφή ενός χαρακτήρα στον *buffer* εξόδου

**% End of Session**

```
fclose(s_port);  
clear s_port;
```

```

get(s)
    ByteOrder = littleEndian
    BytesAvailable = 0
    BytesAvailableFcn =
    BytesAvailableFcnCount = 48
    BytesAvailableFcnMode = terminator
    BytesToOutput = 0
    ErrorFcn =
    InputBufferSize = 512
    Name = Serial-COM1
    OutputBufferSize = 512
    OutputEmptyFcn =
    RecordDetail = compact
    RecordMode = overwrite
    RecordName = record.txt
    RecordStatus = off
    Status = closed
    Tag =
    Timeout = 10
    TimerFcn =
    TimerPeriod = 1
    TransferStatus = idle
    Type = serial
    UserData = []
    ValuesReceived = 0
    ValuesSent = 0

```

## Matlab και RS-232

```

SERIAL specific properties:
BaudRate = 9600
BreakInterruptFcn =
DataBits = 8
DataTerminalReady = on
FlowControl = none
Parity = none
PinStatus = [1x1 struct]
PinStatusFcn =
Port = COM1
ReadAsyncMode = continuous
RequestToSend = on
StopBits = 1
Terminator = LF

```

## Επικοινωνία Stateflow μοντέλων μέσω RS-232

Αρχικοποίηση και ενεργοποίηση RS232 port στο StartFcn callback

Παραμετροποίηση: InputBufferSize και OutputBufferSize

Απενεργοποίηση RS232 port στο StopFcn callback

Χρήση εντολών διαχείρισης ενός RS232 port μέσω κώδικα Matlab και χρήση μεταβλητών στο workspace.

**b2o\_1**: Το πλήθος των bytes προς μετάδοση από το device s1. Ενημερώνεται με κλήση του κώδικα **B2O\_1.m**

**d2o\_1**: Το byte που βρίσκεται στη θέση αυτή μεταδίδεται από το device s1 όταν κληθεί ο κώδικας **DOut\_1.m**

**b2r\_1**: Το πλήθος των bytes προς ανάγνωση από το device s1. Ενημερώνεται με κλήση του κώδικα **B2R\_1.m**

**d2r\_1**: Το byte που διαβάζεται από το device s1 γράφεται στη θέση αυτή όταν κληθεί ο κώδικας **DIn\_1.m**

Αντίστοιχα ισχύουν και για το 2<sup>ο</sup> RS232 port .

## RS232 και έλεγχος ροής με πρόσθετα σήματα (Hardware Flow Control)

- Υπάρχουν δύο ομάδες σημάτων για έλεγχο ροής, τα σήματα RTS (request to send) και CTS (clear to send) καθώς και τα σήματα DSR (data set ready) και DTR (data terminal ready). Τα DSR/DTR δεν χρησιμοποιούνται.
- Το σήμα RTS της μιας πλευράς συνδέεται με το σήμα CTS της άλλης πλευράς.
- Η λειτουργικότητα των σημάτων CTS/RTS μπορεί/πρέπει να διαφοροποιηθεί ανάλογα με τον τύπο του καλωδίου που χρησιμοποιείται.
- Στην περίπτωση του καλωδίου “Null Modem with Full Handshaking” το σήμα εξόδου RTS (τιμή 1) στο PC χρησιμοποιείται για να δηλώσει στο modem ότι το PC μπορεί να δεχθεί δεδομένα. Η ανάγνωση του σήματος CTS προσδιορίζει τη δυνατότητα μετάδοσης από το PC στο modem.
- Για την άλλη κατεύθυνση (από το modem στο PC) χρησιμοποιούνται τα αντίστοιχα σήματα (το RTS στο modem και το CTS στο PC).
- Η τιμή του CTS pin διαβάζεται αξιοποιώντας την PinStatus παράμετρο.
- Η τιμή του RTS pin ελέγχεται με την RequestToSend παράμετρο.

## Τυπική αρχικοποίηση RS-232

```
PORTNAME = 'COM51';
BAUDRATE = 9600;           % bps
IBL = 64;                  % Input buffer (B)
OBL = 1;                   % Output buffer (B)
fprintf('\nOpening RS232 port %s ..... ', PORTNAME);
Sport=serial(PORTNAME,'BaudRate',BAUDRATE,'Parity','none');
set(sport, 'FlowControl', 'none');
set(sport, 'InputBufferSize', 64);
set(sport, 'OutputBufferSize', 1);
fopen(sport);
sport.RecordDetail = 'verbose';
sport.RecordName = 'uart_sport.txt';
record(sport,'on')         % Activate only for debugging
```

## Hardware Flow Control από το Stateflow

Χρήση εντολών διαχείρισης ενός RS232 port μέσω κώδικα Matlab και χρήση μεταβλητών στο workspace.

**rts\_s1**: Η τιμή που θα λάβει το σήμα RTS. Ενημερώνεται με κλήση του κώδικα [RTS\\_1.m](#)

**cts\_s1**: Η τιμή που έχει το σήμα CTS. Ενημερώνεται μετά από κλήση του κώδικα [CTS\\_1.m](#)

Αντίστοιχα ισχύουν και για το 2<sup>ο</sup> RS232 port (s2).

Σε όλα τα μοντέλα να τεθεί: 'FlowControl' = 'none'

## Hardware Flow Control από τη Matlab

Η τιμή του RTS τίθεται με την εντολή ή `sport.RequestToSend = 'off';` ή `'on';`

Η τιμή που έχει το σήμα CTS αναγνωρίζεται με την εντολή `sport.PinStatus.ClearToSend`

```
if length(sport.PinStatus.ClearToSend) == 2
    cts = 1; else cts = 0; end
```

-- sport είναι το όνομα του RS232 device

Στην ενεργοποίηση του RS232 να τεθεί: 'FlowControl' = 'none'