# TCP Multi Accept Server

Notes:
Sequence for using a TCP Multi-accept server.
TCP is a peer-to-peer connection.
This means that only one connection can be handled at a time
and once the communication is complete the connection is closed.

Join a network first.
 Use P0=0-4 to select socket.
 If you only have 1 socket you can ignore this command.
 >PK=1,3000 // Starts keep alive
 >P1=0 //Sets TCP
 OK
 >P2=5024 //selects port
 OK
 >P8=6 //Set the number backlog requests (0-6).
 OK
1. >P5=11 //Start Multi-Accept server
 [TCP TSK] Multi-Accept Setup
 [TCP SVR] Waiting on connection …
OK
2. [TCP SVR] Accepted 192.168.10.100:34489
 //Connection from TCP Client
3. > R0
 Data from 192.168.10.100
 OK
 > S3=1\r1
 1
 OK
4.  > P5=10 //Close current and wait for next connection
 [TCP SVR] Waiting on connection …
OK
 [TCP SVR] Accepted 192.168.10.100:34504 //Connection from TCP Client
2. > R0
 Data from 192.168.10.100
 OK
 > S3=1\r2
 1
 OK
 > P5=0 //Shutdown server
 [TCP TSK] Killed
 OK

When comparing to a Windows server
and a internet browser the process is the same.
The browser sends multiple http request and the server handles them sequentially

**P5=11  Function**
Creates Socket Descriptor
Set Sockets Options
Binds Socket to Ports
Listens and Queues Client Connections
Accepts & Returns Client Connection Socket Descriptors

Start TCP Multi-Accept

OK

1. Start TCP Server
P5=11

2. Waiting for client to Accept Received IP & port

Get

Yes - Client Connects

3. Process Request
Send/ Receive Data
R0/ S3

Yes

4. Request Processed
P5=10

No

Keep Server Running

Shutdown Server Command P5=0 received ?

Yes
P5=0 command issued Stop Server

End