

	<p>If successful an "OK" is returned, or nothing will be returned if already in data or fast data mode, and connected. A "NO CARRIER" occurs when the Bluetooth connection has been lost.</p> <p>EXAMPLE: TYPE : +++<cr> REPLY: <cr_lf>OK<cr_lf> or <cr_lf>NO CARRIER<cr_lf></p>
<p>SESC</p>	<p>,<nn> {Set Escape ASCII Character} where nn, is the ASCII decimal character <256 or non extended ascii characters.</p> <p>EXAMPLE: TYPE : ATSESC,43<cr> // 43 = 2B(hex) = "+" ASCII character REPLY: <cr_lf>OK<cr_lf> or <cr_lf>ERROR<cr_lf></p> <p>Do not need to perform reset and stores permanently.</p> <p>Need to add error constraint if not an integer value.</p>
<p>MD</p>	<p>{Put Radio into Data Mode}</p> <p>This sequence is used to force the radio into Data Mode. In order for this to have an effect the Radio has to be CONNECTED. If successful an "OK" is returned else a "NO CARRIER". Also a "NO CARRIER" occurs when the connection has been lost.</p> <p>EXAMPLE: TYPE : ATMD<cr> REPLY: <cr_lf>OK<cr_lf> or <cr_lf>NO CARRIER<cr_lf></p>
<p>MF</p>	<p>{Put Radio into Fast Data Mode}</p> <p>This sequence is used to force the radio into fast data mode. Once in Fast Data Mode all commands are treated as data, ways to get out of this mode it to reset power on the radio, or strobe PIO(4) while connected.</p> <p>If successful an "OK" is returned, or nothing will be returned if already in fast data mode and connected else "No Carrier" if the Bluetooth RF link is not established.</p> <p>EXAMPLE: TYPE : ATMF<cr> REPLY: <cr_lf>OK<cr_lf> // if connected</p> <p>or</p> <p><cr_lf>OK<cr_lf> <cr_lf>NO CARRIER<cr_lf> // not connected</p>

Note: Byte Gaps and Data Latency – The way *Bluetooth* is designed and operates random byte gaps of 5 msec to 20 msec are common. Packet size will vary from transmission to transmission. The faster the UART speed the smaller the byte gap delay.

Effective data payload throughput in fast streaming mode is approximately 200Kbps and 60Kbps in regular data mode when the AT parser looks at each character for ASCII valid command scripts in the data stream. The radio RX has very limited buffering so if you do not use hardware flow control and are transmitting further distances you will quickly overflow the buffer because of RF retransmissions, etc.