# FTDI Pattern Generator

FTDI Pattern Generator can be used for formatting (parsing) and sending any communication protocol (like SPI) using FTDI port. It uses a pattern text file which specifies the communication protocol to be followed. FTDI pattern generator uses FTDI’s Synchronous Bit Bang mode. FTDI chips which support Synchronous bit bang can use FTDI pattern Generator for writing and reading back. FTDI chips which supports only bit bang(asynchronous bit bang) can be used only for write.

## FTDI Pattern Format File

FTDI Pattern to be followed is passed to the DLL as a text file. It contains two sections – Write and Read. Each section contains 8 rows, representing 8 pins of a FTDI port. The 1st column represents the pin index (starting from 0). The format file will contain multiple columns, each column representing the 8-bits (1 byte) that will be written to the 8 pins of FTDI port.

For example, the clock pin (In fig below, Pin 0) will be represented by alternating 1’s and 0’s representing the clock signal. For write function, the FTDI pattern generator gets the address and data, and forms a byte array (each byte representing 1 column) and writes to the port. For read function, it will get the address from and will extract the data bits from the bytes the port receives.

* “Ax” indicates “bit x of the address to be written” for both the read and write case. The address is always written for both reading and writing. Example: A5 refers to 5th address bit(starting from A0)
* “Dx” indicates “bit x of the data to be written” for writing and “bit x of the data to be read” in the read case. The data is written for writing and read for reading.
* The address length and data length for each device is extracted from the file based on the largest address bit and largest data bit (i.e. A31 indicates a 32 bit address).
* Except for the pin from where the read bits are received, all other pins are configured to be output pins.
* The write and read pin can even be same. (3-wire SPI).
* To retain the current state of a pin (before writing/reading) use “X” in the 2nd column (column next to the pin index). The current state of the pin will be obtained and the entire row of that pin will be replaced with its current state(“0” or “1”)