Setting up, starting, and stopping the objects:

```
1 port:   PUB start(rxpin, txpin, mode, baudrate) : okay          ' must be done for each copy

4 port:   PUB Init
          PUB AddPort(port0,rxpin,txpin,ctspin,rtspin,rtsthreshold,mode,baudrate) ' up to four ports
          PUB AddPort(port1,rxpin,txpin,ctspin,rtspin,rtsthreshold,mode,baudrate) ' can be  specified
          PUB AddPort(port2,rxpin,txpin,ctspin,rtspin,rtsthreshold,mode,baudrate)
          PUB AddPort(port3,rxpin,txpin,ctspin,rtspin,rtsthreshold,mode,baudrate)
          PUB Start : okay
```

Starts the object, specifies pin, mode, and baud rate. The four port objects also have the option of using RTS/CTS pins and buffer threshold settings to control transfers.

```
1 port:   PUB stop       ' stops the cog and frees it up for other uses
4 port:   PUB Stop
```

Sending and receiving data with the objects:

These are the functions that come with the Full Duplex Serial Object. All of the Four Port objects have the same functions, and the only added requirement is to specify the port number when calling that function. There are also several additional useful functions in the Four port objects.

```
1 port:   PUB rxflush             ' flush the receive buffer
4 port:   PUB rxflush(port)

1 port:   PUB rxcheck             ' check if byte received (never waits)
4 port:   PUB rxcheck(port)       ' returns -1 if no byte received, $00..$FF if byte

1 port:   PUB rxtime(ms)          ' Wait ms milliseconds for a byte to be received
4 port:   PUB rxtime(port,ms)     ' returns -1 if no byte received, $00..$FF if byte

1 port:   PUB rx                  ' Receive byte (may wait for byte)
4 port:   PUB rx(port)            ' returns $00..$FF

1 port:   PUB tx(txbyte)          ' Send byte (may wait for room in buffer)
4 port:   PUB tx(port,txbyte)

1 port:   PUB str(stringptr)      ' Send a zero terminated string
4 port:   PUB str(port,stringptr)

1 port:   PUB dec(value)          ' Print a decimal number
4 port:   PUB dec(port,value)

1 port:   PUB hex(value, digits)          ' Print a hexadecimal number
4 port:   PUB hex(port,value, digits)

1 port:   PUB bin(value, digits)          ' Print a binary number
4 port:   PUB bin(port,value, digits)
```