If you want I can help you with linux part of serial port implementation.

First you should tell me what interface you want.

My proposition:

1. Vector<String> ListPorts();
returns list of available ports. On windows you can use QueryDosDeviceW to get list of devices.
On linux udev does similar thing.

2. bool Open(const char * port_name, enum speed, enum flow_control, enum parity)

3. void Close()

4. int Write(const void * data, int len, int timeout_ms = 5000);
5. int Write(String data, int timeout_ms = 5000);

6. String ReadUntil(int amount, int stop_byte = -1, int timeout_ms = 5000);

7. void FlushInput();
to clean input queue

8. void FlushOutput();
to clean output queue

I would like to have timeouts in every Write and Read functions, because they often depend on the command you send to device.

Things like stop_byte are useful for communication with modems.

Maybe we should also have Callbacks (when new data arrive, when usb serial device is (un)plugged)?