
Subject: Re: Some projects are ready to become open-source

Posted by [Mindtraveller](#) on Tue, 06 May 2014 21:44:12 GMT

[View Forum Message](#) <> [Reply to Message](#)

SerialPort - not just a simple serial port package. It is based on MtAlt and BNF.

Long story short, it loads the file with description of each serial port to be used. A multithreaded class is created for each port. It also loads a file with protocol descriptions (i.e. modbus,...).

Let's look how you request i/o with serial port:

```
void SendReceive_  
(  
    String      sendProto,  
    Vector<Value> sendArgs,  
    String      receiveProto,  
    dword       timeout,  
    void        *custom,  
    Ptr<OBJECT>  notify,  
    void (OBJECT::*cb)(bool success, Vector<Value> args, void *custom),  
    bool        useAcc = false,  
    bool        priority = false  
)
```

Ending underscore means this is asynchronous function which may be called from any other thread ("mt safe").

sendProto and sendArgs are send data generation protocol and it's variables (string, integer, float, etc.)

receiveProto is data parsing protocol to parse incoming data. As soon as incoming bytes are parsed successfully, incoming datagram is considered successfully received.

timeout is a receive timeout which is monitored by internal thread.

custom - is just a custom value to distinguish multiple SendReceive_() calls from each other.

notify/cb - is a CallbackQueue/CallbackThread object and it's member function which is called upon send/receive/timeout event.

useAcc - is a "caching" feature; Serial port classes keep some cache of incoming byte sequences already parsed. And if the same sequence is met, no actual parsing is made, instead returning cached result. This is extremely useful for slow processors like early celerons or arm. My tests revealed that more than 60% of incoming packages are the same and thus are cached very effectively leaving many processor resources for other tasks.

priority - is a simple true/false switch putting your call into the front of queue or into the end of it. Yes, just imagine you call SendReceive_() before previous call is still working. According to MtAlt, all subsequent calls are added to the queue (not necessarily to the tail, because we have priorities).

This package needs further discussion and examples which will follow if someone interests. Should we make new topic for it?
