Subject: Re: About RS232
Posted by jeremy_c on Fri, 20 Aug 2010 11:58:23 GMT
View Forum Message <> Reply to Message

Last night I grabbed a copy of libctb (https://iftools.com/opensource/ctb.en.php). I modified the source a bit and made a U++ package out of it. So now I just say Add Package, Ctb and my app has serial support.

I have not tested it yet under linux but will be doing so shortly. The only problem with what I did was I had to change several headers for the package. For example, the ctb package uses:

```
#include "ctb-0.15/abcdef.h"
```

I changed these to read:

```
#include <Ctb/abcdef.h>
```

So, with each update to libctb, that would have to be done. Is there a better way? I wanted to let U++ handle the building of the package.

I started with Serial.h/.cpp that I found here and it worked fine for some devices I was controlling but not for others. I needed the ability to set 2 stop bits, to raise the DTR and RTS states, etc...

I am not sure what is the best idea here. To keep my Ctb package or to browse it's source, learn how the Data Bits, Parity, Stop Bits and Line States are set/queried and port them to the Serial.h/Serial.cpp. I kind of liked the Serial.h/.cpp's function ReadDataWaiting(). w/Ctb you can read X number of bytes with a timeout but say you want 30 bytes, Readv(buf, 30); may result in a partial read, i.e. maybe 15 were available before the timeout occurs. You'll get the 15. With Serial.h/.cpp I could enter a loop and say:

```
tries = 0;
while (ReadDataWaiting() < 30) {
  tries++;
  if (tries > 3) do_fail_code();
  Sleep(50);
}
```

Jeremy