

```
package nmc.sidekick;

import java.io.*;
import javax.comm.SerialPort;

public class MicroCat implements Instrument, IoResource {

    SerialPort microcatLink;
    OutputStream ostream;
    InputStream istream;
    DataPort dataPort;

    public void setSerialPort(SerialPort aPort){
        this.microcatLink = aPort;
        try {
            ostream = aPort.getOutputStream();
            istream = aPort.getInputStream();
        } catch (IOException e) {
            System.out.println("error in getting io streams from serial port:"
                               + e);
        }
    }

    public void setDataPort(DataPort aDataPort){
        this.dataPort = aDataPort;
    }

    public byte[] poll(){

        byte[] cr = new byte[1];
        cr[0] = 0x0d;
        byte[] gt = ">".getBytes();
        System.out.println("gt is " + (char)gt[0]);
        int c = 0;
        int pos = 0;
        byte[] retbuf = new byte[1024];
        try {
            byte[] mchi = "\r".getBytes();
            ostream.write(mchi, 0, mchi.length);
            while (true) {
                int a = istream.available();
                c = istream.read();
                System.out.println("uC(" + a + "): " + (char)c);
                if ((char)c == '>')
                    break;
            }
            byte[] cmdstr = "ts\r".getBytes();
            ostream.write(cmdstr, 0, cmdstr.length);
            while (true) {
                c = istream.read();
                System.out.println("uC: " + (char)c);
                if (((char)c != '\r')
                    && ((char)c != '\n')
                    && ((char)c != '$')) {

                    if ((char)c == '>') {
                        if (pos > 7) {
                            break;
                        } else {
                            pos = 0;
                            continue;
                        }
                    }
                }
            }
        }
    }
}
```

```
        }
        retbuf[pos++] = (byte)c;
    }
} catch (IOException e) {
    System.out.println("error in talking to ucat: " + e);
}

byte[] realretval = new byte[pos];
for (int i=0; i<pos; i++) {
    realretval[i] = retbuf[i];
}

return realretval;
}

public Object getResourceType(IoResourceType aType) {
    if (aType == IoResourceType.DPA_BOARD)
        return null;
    else if (aType == IoResourceType.INSTRUMENT)
        return this;
    else
        return null; // probably should throw exception
}

public void startStreaming(){
    // launch a thread that gathers wool until it sees the
    // "send the wool to the woolmakers" tokens, and then goes
    // back to woolgathering...
    // It should probably be interrupted forcefully when
    // stopstreaming comes along.
}

public void stopStreaming(){
    // Interrupts the thread above, unless it is in the
    // phase of sending info... and even then, it should only abend
    // if it is partway through getting something sent, and that's
    // only because it might leave the Ethernet universe in an
    // uncertain state.
}
}
```