

EPICS 'Stream' Device Support

Klemen Vodopivec based on slides from
Kay Kasemir

Dec 2018

What is streaming device?

- Text (ASCII) bi-directional protocol
- (Usually) one connection per device
- Command & response
 - What is the current temperature on channel A?
KRDG A?
 - Reply
+077.350E+0\n
- Serial (RS-232, RS485), Networked (TCP), GPIB (IEEE-488)
- EPICS device called StreamDevice

Examples

- Temperature controllers
 - Lakeshore
 - Omega
 - ...
- Vacuum pumps
 - Varian
 - ...
- Power supplies
 - Agilent
 - Spellman
 - ...
- Moxa (modbus, profinet, serial, etc.)

Pro/Cons

- Seems easy
 - Human-readable
 - Testable with minicom, telnet
 - Create Visual Basic, LabVIEW, Python ... demo in no time
- But beware!
 - Speed and Terminators (CR, LF, CR/LF, ...) can magically change
 - Was set to something, than device power-cycled ...
 - Must handle timeouts
 - Don't just hang!
 - Must handle errors
 - Don't read "MODEL 340" as 340 Kelvin

EPICS StreamDevice Idea

Protocol File "demo.proto"

```
Terminator = CR;
```

```
getTempA
```

```
{  
  out "KRDG? A";  
  in "%f";  
}
```

Record

```
record(ai, "Temp:B")  
{  
  field(DTYP, "stream")  
  field(INP, "@demo.proto getTempA TC1")  
  field(SCAN, "5 second")  
}
```

IOC st.cmd

```
drvAsynIPPortConfigure ("TC1", "192.168.164.10:23")
```

EPICS StreamDevice features

- EPICS integration
- Connection management through asyn
 - Automatically connect
 - Re-connect after disconnects
- Allow many records to communicate via one connection
 - Threading, queuing, ...
- Handle timeouts, errors
 - Put records into ‘alarm’ state
- Debug options
 - Log every byte sent/received

Adding StreamDevice support to IOC

- configure/RELEASE or RELEASE.local:

```
ASYN=/path/to/asyn
```

```
STREAM=/path/to/StreamDevice
```

- demoApp/src/Makefile

```
demo_DBD += stream.dbd
```

```
demo_DBD += asyn.dbd
```

```
demo_DBD += drvAsynIPPort.dbd
```

```
demo_LIBS += asyn
```

```
demo_LIBS += stream
```

EPICS Database

stream-demo.db

```
record(ai, "B")
{
  field (DTYP, "stream")
  field (INP, "@demo.proto getB NC")
  field (SCAN, "5 second")
}
record(ai, "A")
{
  field (DTYP, "stream")
  field (INP, "@demo.proto getA NC")
  field (SCAN, "I/O Intr")
}

record(ao, "current")
{
  field (DTYP, "stream")
  field (OUT, "@demo.proto setCurrent NC")
  field (EGU, "A")
  field (PREC, "2")
  field (DRVL, "0")
  field (DRVH, "60")
  field (LOPR, "0")
  field (HOPR, "60")
}
```


Protocol File

stream-demo.proto

System variables

```
Terminator = CR LF;  
InTerminator = LF;  
ReplyTimeout = 10000;  
ReadTimeout = 10000;
```

```
# Used with SCAN "... second" -  
# triggered from DB record.  
# Prompts, then expects "B 5"  
getB  
{  
    out "B?";  
    in "B %f";  
    @mismatch  
    {  
        disconnect;  
    }  
}
```

```
ed with SCAN, "I/O Intr".  
# Reacts to "A 5" at any time  
getA  
{  
    PollPeriod = 50;  
    in "A %f";  
}  
  
# Example with initialization,  
# otherwise only writes when processed  
setCurrent  
{  
    out "CURRENT %.2f";  
    @init  
    {  
        out "CURRENT?";  
        in "CURRENT %f A";  
    }  
}
```

Protocol File

stream-demo.proto

Comment

```
Terminator = CR LF;  
InTerminator = LF;  
ReplyTimeout = 10000;  
ReadTimeout = 10000;
```

```
# Used with SCAN "... second" -  
# triggered from DB record.  
# Prompts, then expects "B 5"
```

```
getB  
{  
    out "B?";  
    in "B %f";  
    @mismatch  
    {  
        disconnect;  
    }  
}
```

```
    ed with SCAN, "I/O Intr".  
# Reacts to "A 5" at any time  
getA  
{  
    PollPeriod = 50;  
    in "A %f";  
}  
  
# Example with initialization,  
# otherwise only writes when processed  
setCurrent  
{  
    out "CURRENT %.2f";  
    @init  
    {  
        out "CURRENT?";  
        in "CURRENT %f A";  
    }  
}
```

Protocol File

stream-demo.proto

```
Terminator = CR LF;  
InTerminator = LF;  
ReplyTimeout = 10000;  
ReadTimeout = 10000;  
  
# Used with SCAN "... second" -  
# triggered from DB record.  
# Prompts, then expects "B 5"
```

```
getB  
{  
    out "B?";  
    in "B %f";  
    @mismatch  
    {  
        disconnect;  
    }  
}
```

Function
definition and
implementation
in curly brackets

```
ed with SCAN, "I/O Intr".  
acts to "A 5" at any time  
  
PollPeriod = 50;  
in "A %f";  
}  
  
# Example with initialization,  
# otherwise only writes when processed  
setCurrent  
{  
    out "CURRENT %.2f";  
    @init  
    {  
        out "CURRENT?";  
        in "CURRENT %f A";  
    }  
}
```

Protocol File

stream-demo.proto

```
Terminator = CR LF;  
InTerminator = I;  
ReplyTimeout = 1000;  
ReadTimeout = 1000;
```

```
# Used with SCAN, "I/O Intr".  
# Reacts to "A 5" at any time  
getA
```

```
{
```

```
  out
```

```
  in "B %f";
```

```
  @mismatch
```

```
  {
```

```
    disconnect;
```

```
  }
```

```
}
```

Commands

- **out** for send
- **in** for receive

Other commands:

- wait
- event
- connect
- disconnect
- exec

```
# Used with SCAN, "I/O Intr".  
# Reacts to "A 5" at any time  
getA
```

```
{
```

```
  PollPeriod = 50;
```

```
  in "A %f";
```

```
}
```

```
# Example with initialization,  
# otherwise only writes when processed  
setCurrent
```

```
{
```

```
  out "CURRENT %.2f";
```

```
  @init
```

```
  {
```

```
    out "CURRENT?";
```

```
    in "CURRENT %f A";
```

```
  }
```

```
}
```

Protocol File

stream-demo.proto

```
Terminator = CR LF;
InTerminator = LF;
ReplyTimeout = 10000;
ReadTimeout = 10000;

# Used with SCAN " ...
# triggered from DB
# Prompts, then exp
getB
{
    out "B?";
    in "B %f";
    @mismatch
    {
        disconnect;
    }
}
```

Format converters start with % - similar to printf() syntax

%f for floating point number

```
# Used with SCAN, "I/O Intr".
# Reacts to "A 5" at any time
getA
{
    PollPeriod = 50;
    in "A %f";
}

# Example with initialization,
# otherwise only writes when processed
setCurrent
{
    out "CURRENT %.2f";
    @init
    {
        out "CURRENT?";
        in "CURRENT %f A";
    }
}
```

Protocol File

stream-demo.proto

```
Terminator = CR LF;  
InTerminator = LF;  
ReplyTimeout = 10000;  
ReadTimeout = 10000;
```

```
# Used with SCAN "... second" -  
# triggered from DB record.  
# Prompts, then expects "B 5"
```

```
getB  
{  
    out "B?";  
    in "B %f";
```

```
@mismatch  
{  
    disconnect;  
}
```

```
}
```

Exception handling

- mismatch
- writetimeout
- replytimeout
- readtimeout

```
# Used with SCAN, "I/O Intr".  
# Reacts to "A 5" at any time
```

```
getA  
{  
    PollPeriod = 50;  
    in "A %f";  
}
```

```
... with initialization,  
... otherwise only writes when processed  
... rent
```

```
... t "CURRENT %.2f";  
... nit
```

```
... out "CURRENT?";  
... in "CURRENT %f A";
```

```
}
```

```
}
```

Protocol File

stream-demo.proto

```
Terminator = CR LF;
InTerminator = LF;
ReplyTimeout = 10000;
ReadTimeout = 10000;

# Used with SCAN "... second" -
# triggered from DB record.
# Prompts, then expects "B 5"
getB
{
    out "B?";
    in "B %f";
    @mismatch
    {
        # Commented out for demo
        #disconnect;
    }
}
```

```
# Used with SCAN, "I/O Intr".
# Reacts to "A 5" at any time
getA
{
    PollPeriod = 50;
    in "A %f";
}

# Example with initialization,
# otherwise only writes when processed
setCurrent
{
    out "CURRENT %.2f";
    @init
    {
        out "CURRENT?";
        in "CURRENT %f A";
    }
}
```

IOC Startup File

iocBoot/iocdemo/st.cmd

```
epicsEnvSet ("STREAM_PROTOCOL_PATH", "path/to/proto/files")

drvAsynIPPortConfigure ("NC", "127.0.0.1:6543")

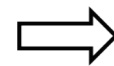
# Log some asyn info and in/out texts
# ASYN_TRACE_ERROR      0x0001
# ASYN_TRACEIO_DEVICE  0x0002
# ASYN_TRACEIO_FILTER  0x0004
# ASYN_TRACEIO_DRIVER  0x0008
# ASYN_TRACE_FLOW      0x0010
# ASYN_TRACE_WARNING   0x0020
asynSetTraceMask("NC", 0, 4)
asynSetTraceIOMask("NC", 0, 6)

dbLoadRecords ("db/stream-demo.db", "user=fred")
```


Example Session

IOC console output

```
2018/12/04 16:18:11.519 NC wrote  
B?\r\n  
42 3f 0d 0a
```



NetCat interactive demo
'nc -l 127.0.0.1 6543

```
B?
```

Example Session

IOC console output

```
2018/12/04 16:18:11.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:11.902 NC read
B 12\n
42 20 31 32 0a
```

NetCat interactive demo
'nc -l 127.0.0.1 6543

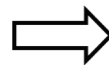
```
B?
B 12
```



Example Session

IOC console output

```
2018/12/04 16:18:11.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:11.902 NC read
B 12\n
42 20 31 32 0a
2018/12/04 16:18:16.519 NC wrote
B?\r\n
42 3f 0d 0a
```



NetCat interactive demo 'nc -l 127.0.0.1 6543

```
B?  
B 12  
B?
```

Example Session

IOC console output

```
2018/12/04 16:18:11.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:11.902 NC read
B 12\n
42 20 31 32 0a
2018/12/04 16:18:16.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:16.875 NC read
A 3.14\n
41 20 33 2e 31 34 0a
```

NetCat interactive demo
'nc -l 127.0.0.1 6543

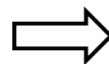
```
B?
B 12
B?
B 3.14
```



Example Session

IOC console output

```
2018/12/04 16:18:11.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:11.902 NC read
B 12\n
42 20 31 32 0a
2018/12/04 16:18:16.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:16.875 NC read
A 3.14\n
41 20 33 2e 31 34 0a
2018/12/04 16:18:21.519 NC wrote
B?\r\n
42 3f 0d 0a
```



NetCat interactive demo 'nc -l 127.0.0.1 6543

```
B?
B 12
B?
B 3.14
B?
```

Example Session

IOC console output

```
2018/12/04 16:18:11.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:11.902 NC read
B 12\n
42 20 31 32 0a
2018/12/04 16:18:16.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:16.875 NC read
A 3.14\n
41 20 33 2e 31 34 0a
2018/12/04 16:18:21.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:31.529 NC B: No reply from device within 10000 ms
```

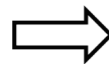
NetCat interactive demo
'nc -l 127.0.0.1 6543

```
B?
B 12
B?
B 3.14
B?
```

Example Session

IOC console output

```
2018/12/04 16:18:11.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:11.902 NC read
B 12\n
42 20 31 32 0a
2018/12/04 16:18:16.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:16.875 NC read
A 3.14\n
41 20 33 2e 31 34 0a
2018/12/04 16:18:21.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:31.529 NC B: No reply from device within 10000 ms
2018/12/04 16:18:36.519 NC wrote
B?\r\n
42 3f 0d 0a
```



NetCat interactive demo 'nc -l 127.0.0.1 6543

```
B?
B 12
B?
B 3.14
B?
B?
```

Example Session

IOC console output

```
2018/12/04 16:18:11.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:11.902 NC read
B 12\n
42 20 31 32 0a
2018/12/04 16:18:16.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:16.875 NC read
A 3.14\n
41 20 33 2e 31 34 0a
2018/12/04 16:18:21.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:31.529 NC B: No reply from device within 10000 ms
2018/12/04 16:18:36.519 NC wrote
B?\r\n
42 3f 0d 0a
2018/12/04 16:18:36.893 NC read
34\n
33 34 0a
2018/12/04 16:18:36.893 NC B: Input "34" mismatch after 0 bytes
2018/12/04 16:18:36.893 NC B: got "34" where "B " was expected
```

NetCat interactive demo
'nc -l 127.0.0.1 6543

```
B?
B 12
B?
B 3.14
B?
B?
34
```



More in StreamDevice Manual (online)

- 21 supported EPICS records
 - Including waveform
- Value skipping (ie. ROI %*d %d)
- Multi-line messages
- Format converters
- Settings aka System variables
- <http://epics.web.psi.ch/software/streamdevice/doc/index.html>

And asyn for connection details

- drvAsynSerialPortConfigure
 - Baud, bits, parity, stop etc.
- drvAsynIPPortConfigure
 - Hostname and port
- asynGpib
- <https://epics.anl.gov/modules/soft/asyn/R4-34/asynDriver.html>
- https://epics.anl.gov/modules/soft/asyn/R4-34/HowToDoSerial/HowToDoSerial_StreamDevice.html

StreamDevice...

- Allows you to concentrate on the protocol
- Handles the rest
 - Connection
 - Threads
 - Parse results
 - Update record's value and alarm state

“demo.proto”

```
Terminator = CR;
```

```
getTempA
```

```
{
```

```
    out "KRDG? A";
```

```
    in "%f";
```

```
}
```