OAK RIDGE National Laboratory

EPICS 'Stream' Device Support

Klemen Vodopivec
Kay Kasemir

July 2026

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

What is a “streaming” device?

- Text (ASCII) bi-directional protocol
- (Usually) one connection per device
- Command & response
  - What is the current temperature in Kelvin on channel A?

    `KRDG? A\n`
  - Reply

    `+077.350E+0\n`
- Serial (RS-232, RS485), Networked (TCP), GPIB (IEEE-488)

➔ Handled by EPICS “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, nc, HyperTerm, Putty, …
  - 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, then 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:A")
{
  field(DTYP, "stream")
  field(INP,  "@demo.proto getTempA TC1")
  field(EGU,  "Kelvin")
  field(PREC, "2")
  field(SCAN, "5 second")
}
```

IOC st.cmd

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

EPICS StreamDevice features

- Connection management through asyn
  - Automatically connect
  - Re-connect after disconnects

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

- Allow many records to communicate via one connection
  - Threading, queuing, …
- Handle timeouts, errors
  - Put records into ‘alarm’ state

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

- Debug options
  - Log every byte sent/received (as part of asyn)

```
IOC → Device: KRDG? A

IOC ← Device: 47.11
```

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
```

See /ics/examples/07_stream/demoApp/src/Makefile

EPICS Database

stream-demo.db

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

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

See /ics/examples/07_stream/demoApp/Db

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";
    }
}
```

See /ics/examples/07_stream/demoApp/Db

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;
    }
}
```

Comment

```
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 = 
ReplyTimeout = 
ReadTimeout = 

# Used with SC
# triggered fr
# Prompts, the
getB
{
    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";
}

ple with initialization,
rwise 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")

# What to log
# 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, 0xE)

# How to show the logged text
# ASYN_TRACEIO_NODATA 0x0000
# ASYN_TRACEIO_ASCII  0x0001
# ASYN_TRACEIO_ESCAPE 0x0002
# ASYN_TRACEIO_HEX    0x0004
asynSetTraceIOMask("NC", 0, 6)

dbLoadRecords("db/stream-demo.db")
```

See /ics/examples/07_stream/iocBoot/iocdemo

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?


A 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?


A 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?


A 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?


A 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?


A 3.14


B?



B?


34
```

More to try

- Note how IOC once asks for “CURRENT?” at startup
- Later, writing to “current” record will write that value to device
- Device might at any time send “A {some number}”

More in StreamDevice Manual (online)

- Formats supported by record type
  - ai, ao — %d or %i for integer, %f for floating point, %.3f for 3 digits
  - bi, bo — %d or %i for enum index, %s for ZNAM/ONAM label
  - Numeric waveform — %d or %i or %f to read elements
    char waveform %s to read until whitespace, %#s to read until `null` character
- Can handle tricky protocols
  - Value skipping (ie. ROI %*d %d)
  - Multi-line messages
  - Checksums
  - Format converters

➔ https://paulscherrerinstitute.github.io/StreamDevice/

Check asyn for connection details

- drvAsynSerialPortConfigure
  - Baud, bits, parity, stop etc.
- drvAsynIPPortConfigure
  - Hostname and port

➔https://epics.anl.gov/modules/soft/asyn/

➔https://epics.anl.gov/modules/soft/asyn/R4-38/asynDriver.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";
}
```