

'Busy' Record, Put-callback Support

Kay Kasemir

Oct. 2018

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

Basic Automation Problem

1. Move motor to some position

Has the motor
reached the position?

2. Open shutter

How long does the
shutter need to open?

3. Take data for 5 seconds

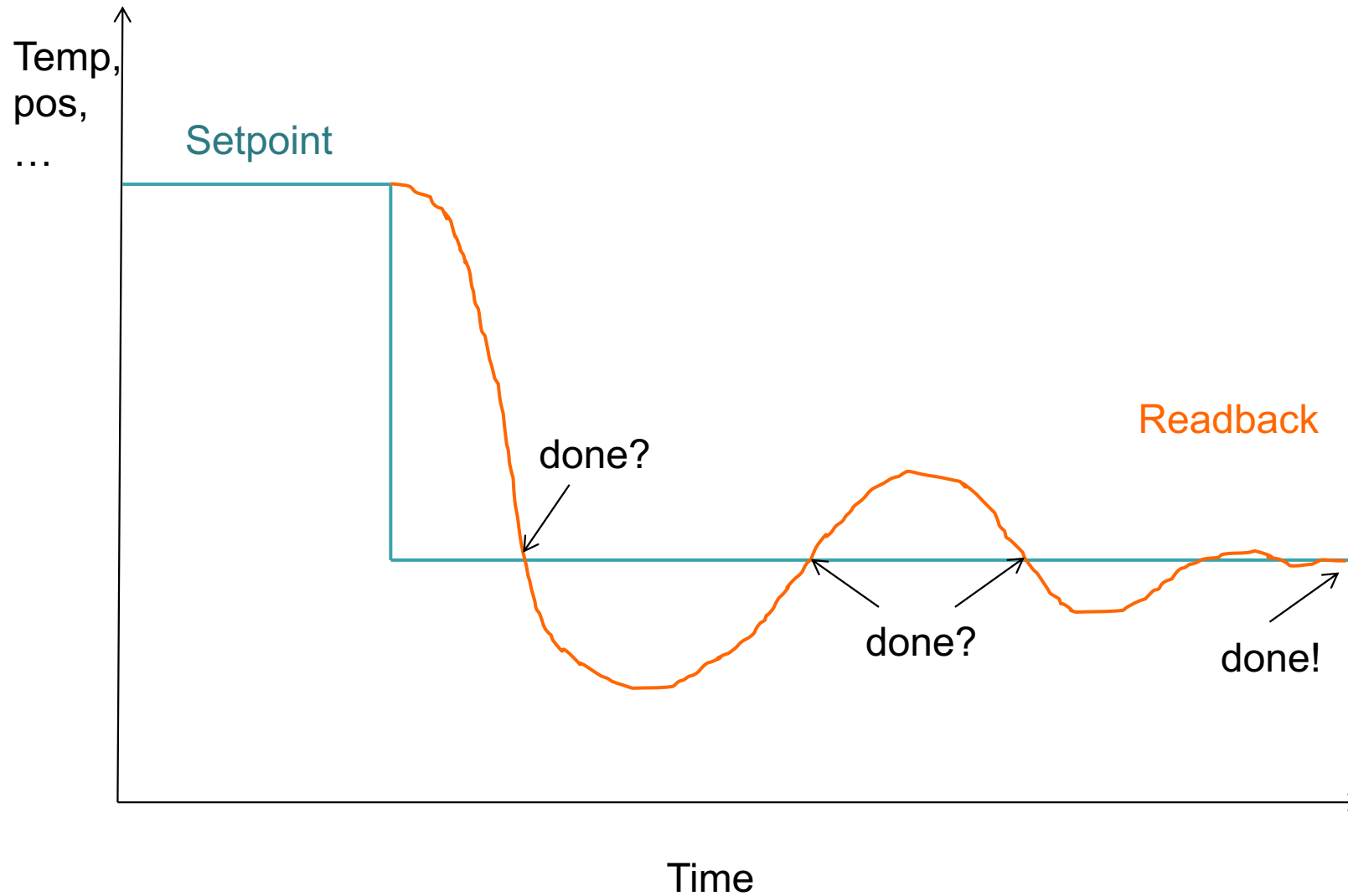
4. Close shutter

What not to do

1. Ask motor to move to position X
2. Wait until motor encoder reports $X \pm 0.1 \text{ mm}$
3. Ask shutter to open
4. Wait 4 seconds - Fred said that's how long it'll take
5. Take data for 5 seconds
6. Close shutter

Not reliable!

Based on readback, are we done?



IOC has to tell us when it's done

1. Use CA “put-callback”
2. Record completes callback when its “done”

Requires record with completion support

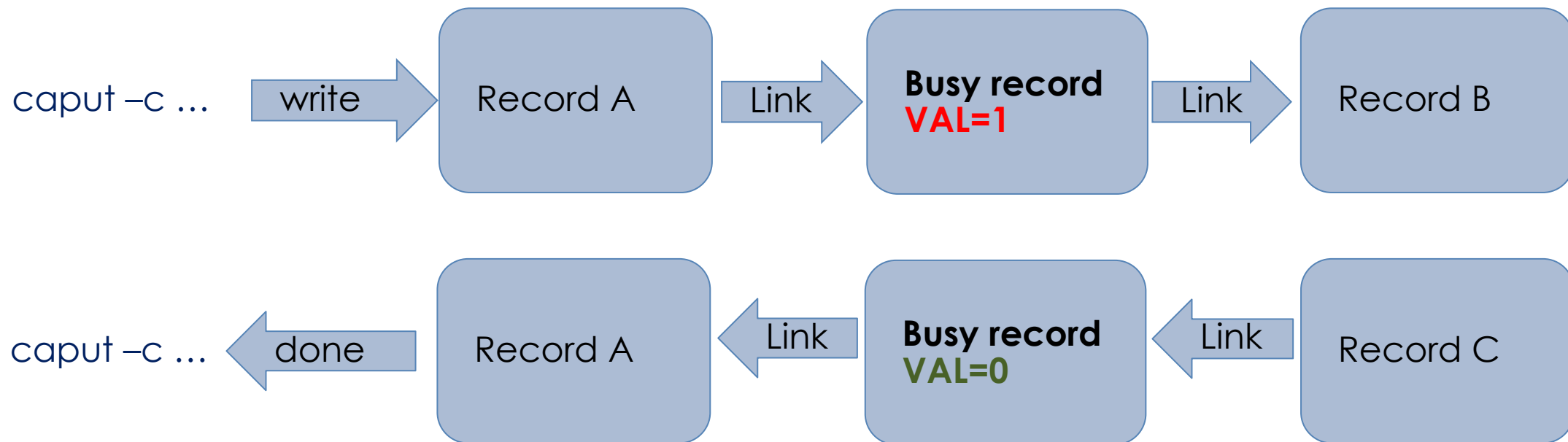
- Motor record: When reaching desired position
- General record: Check device support manual
- Plain database: Use BUSY record

EPICS record PACT field

- PACT is binary flag that is 1 (ON) when record is processing
- put-callback is mechanism to use PACT information...
- ...and not return until PACT is 0 (OFF)
- But it only works with CA
- `caput -c -w <timeout> <PV>`

BUSY record

- VAL = 0 "Done" or 1 "Busy"
- Blocks processing (PACT=1) while busy



Typical Use

- Writing to some record, which has FLNK or OUT links to a busy record, sets the busy.VAL = 1
- This causes writer to that first record to block in put-callback
- Something else resets busy.VAL = 0
 - a) Other database logic
 - b) Sequencer code
 - c) asyn device support
 - d) ...
- Put-callback completes

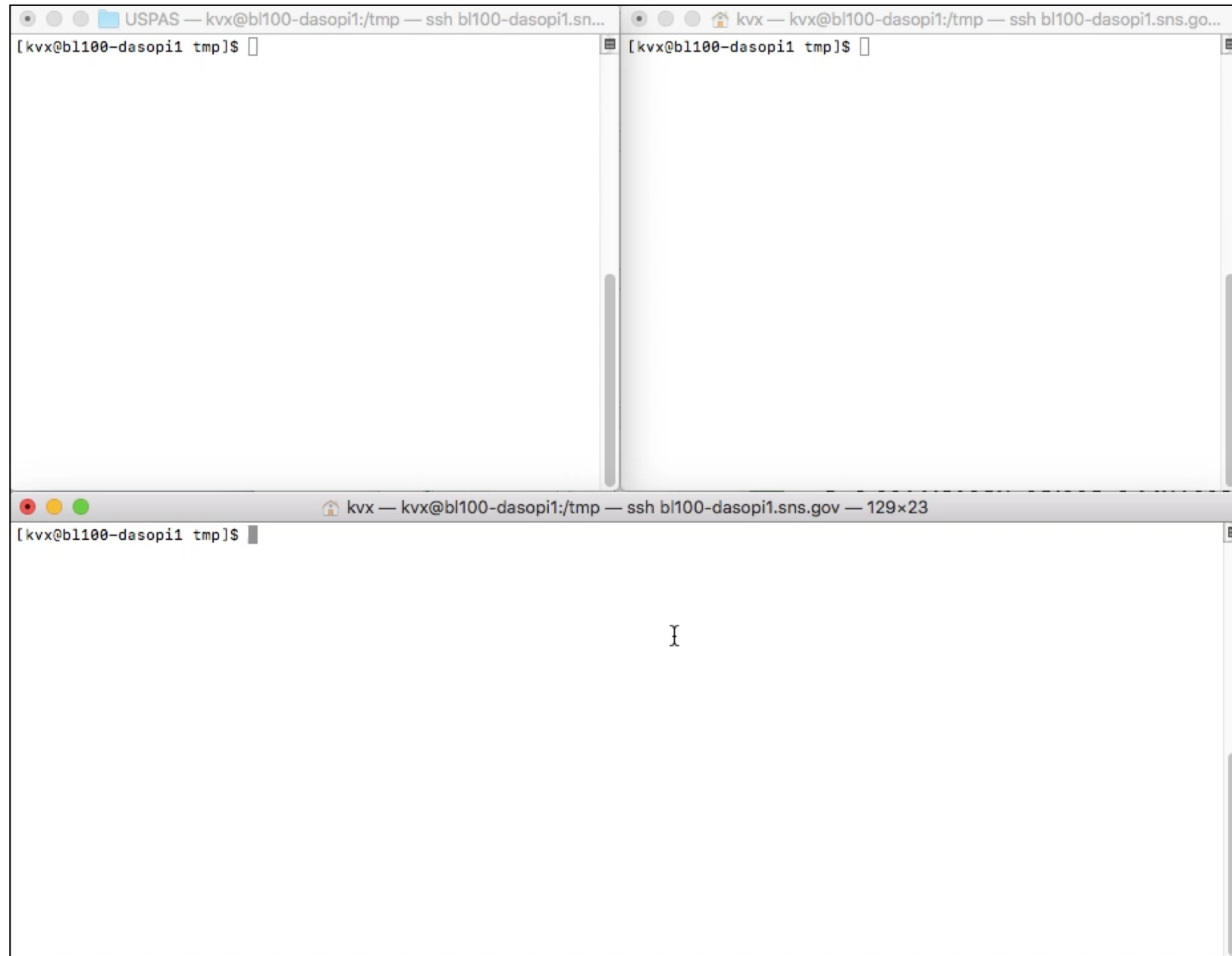
Example

```
record(ao, "SetPoint") {  
  field(FLNK, "BusySet")  
}
```

```
record(ai, "ReadBack") {  
  field(FLNK, "BusyClear")  
}
```

```
record(longout, "BusySet") {  
  field(VAL, "1")  
  field(OUT, "Busy PP")  
}  
record(longout, "BusyClear") {  
  field(VAL, "0")  
  field(OUT, "Busy PP")  
}  
record(busy, "Busy") {  
}
```

Example



Put-Callback

.. Is essential for robust automation

Use BUSY record to create database logic that supports put-callback