



Constants.H

UP = 1

DOWN = 2

OFF = 3

TEMP = 1

LIGHT = 2

GET_DATA = 1

StartProcess

1) Fork / Exec display

2) CREATE Sensor pipes.

↳ ex: /dev/s1

3) Create Sensor

↳ Pipe read end

4) Create EnvSimulator

- Give WRITE FD for all sensor pipes

* EnvSim agnostic to sensor
actuator pairs

5) Create Actuators

6) Create Managers

- Sensor server name
- actuator server name

display

- while loop for reading messages
- handle pulses in dates
- ncurses (maybe)
(otherwise update
xHz)

↳ varname ?
↳ reading ↳ payload

Sensor

- has pipe read fd
- X Hz read from pipe
- Store most recent data
- read thread / msgReceive thread

N.B. TWO THREADS



VarManager:

- Read from sensor (msgSend("GET_DATA"))
- Pulse to display data
- msgSend(activator)
 - ↳ Pulse to env and reply
 - Set thresholds in .h as Macros
- Sleep u (xHz)

Actuator

- Read loop
- Pulse to envSim
- Reply UP | DOWN | OFF

ON/OFF
VAR CONST
↳ TEMP
↳ LIGHT

EnvSim

TWO THREADS
fill in

- While loop, msgRead → Get actuator pulses

- STORE ACTUATOR STATE

- If actuator off:

modify temp by $[-2, +2]$

- If actuator UP:

modify temp by $[0, 1]$

similar for DOWN

$[-1, -0.1]$

WRITE TO PIPE NEW

DATA POINT

