The following slides accompany recorded audio, and were part of a presentation at:

Texas Linux Fest 2011

Austin, TX

Hilton Hotel

Saturday April 2, 2011

2:00 – 2:45 PM CST

http://texaslinuxfest.org/

http://www.thomasstover.com/ladtools/

Defining “Automation”

More Generally – substituting repetitive, tedious, dangerous, or difficult human labor with technology

More Specifically & More Currently - interfacing computers with sensors and machines; ie input and output in the “real” physical world

Lineage of Automation

- Related to, but not really the same story as:
  - History of Harnessed Energy
  - History of Industrialization
  - History of Mechanization
  - History of Robotics
  - History of Telemetry
  - History of Computer Science
  - History of Data Recording
  - History of Electronics
  - History of Civil & Defense Engineering

An Automated World is a Hackable World

PC Based DAQ / Control

- Special PC-104/PCI/PCI-E boards
- Heavy use of “break out box” wiring
- USB / IEEE variations
- Embedded & Industrialized PCs
- High Bandwidth IO, i.e. sonar data
- Often real time OS is used
- Occasionally with integrated FPGA

Microcontrollers in Automation

- Probably the most common in hobbyist uses
- Prototype Boards vs Volume Fabrication
- Quantity, Availability, Purchasing issues
- BASIC variants / uCLinux / C w/ no os

ASIC & FPGA
in Automation

PLC Based Automation

- Component Separation
  - Development
  - Robustness
  - Hard / Soft Real Time
- Regulatory / Building Code Approval
- It's “accepted”
- The way many fields are being “trained” to think

The PLC Industry

&

AD disclaimers

Why Linux?

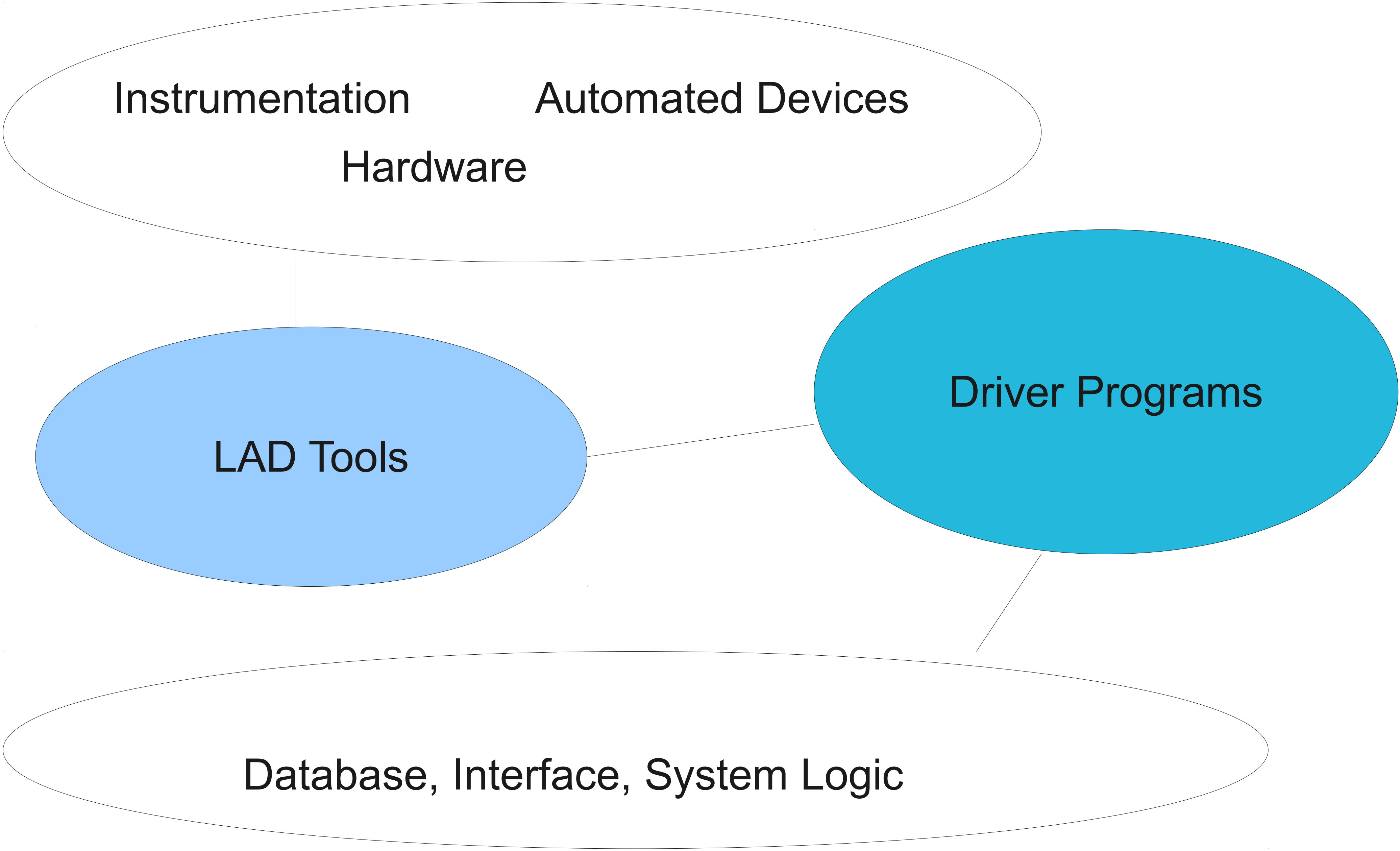
- Long Term Time Scale and Stability
- More Development Choices
- Security, Security, Security

What about MODBUS?

L.A.D. Tools

- C library
- Language Bindings
- Data proxy / connection sharing server
- Utilities
- Integration with other software

Multi-tiered Automation Architecture



Alternative Use Cases

Power cycle buggy hardware

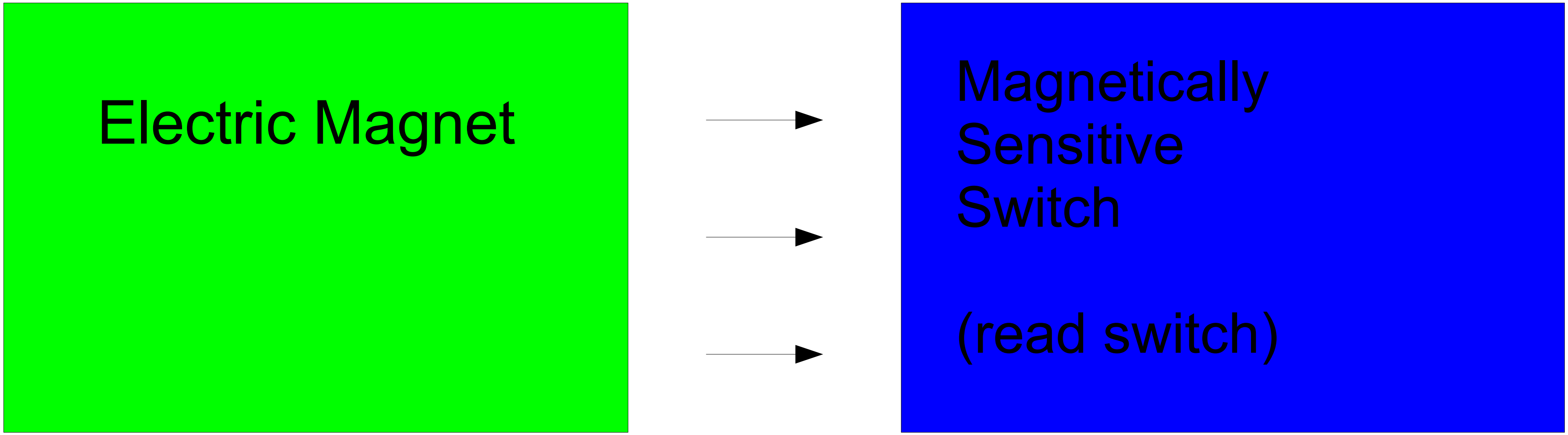
Server Closet Monitoring

Home / Building Automation

Science Projects

Relays

Logic from one circuit can be proxied to another.



Electric Magnet

Magnetically
Sensitive
Switch

(read switch)

“coil”

“contact”

Search for “Harry Porter's Relay Computer”

Relays

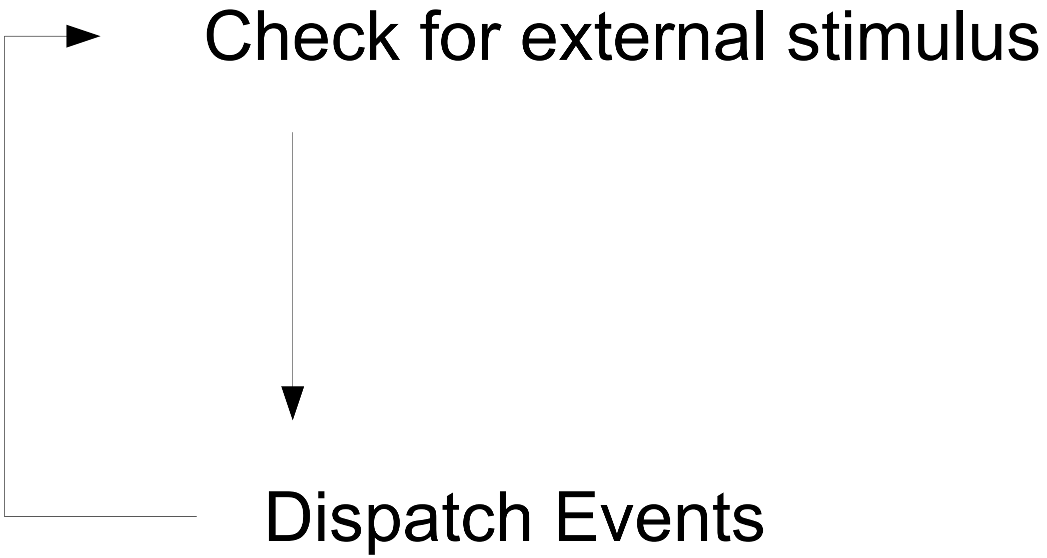
```
on_coil_signal_or_hardware_interrupt()
{

 for each normally open contact()
 {
  contact state = coil_energized_state;
 }

 for each normally closed contact()
 {
  contact state =  !(coil_energized_state);
 }

}
```

Event Loops



Check for external stimulus

Dispatch Events

Interrupts, signals, sockets, files, keyboard, mouse, joystick, serial ports, messages, “contacts”

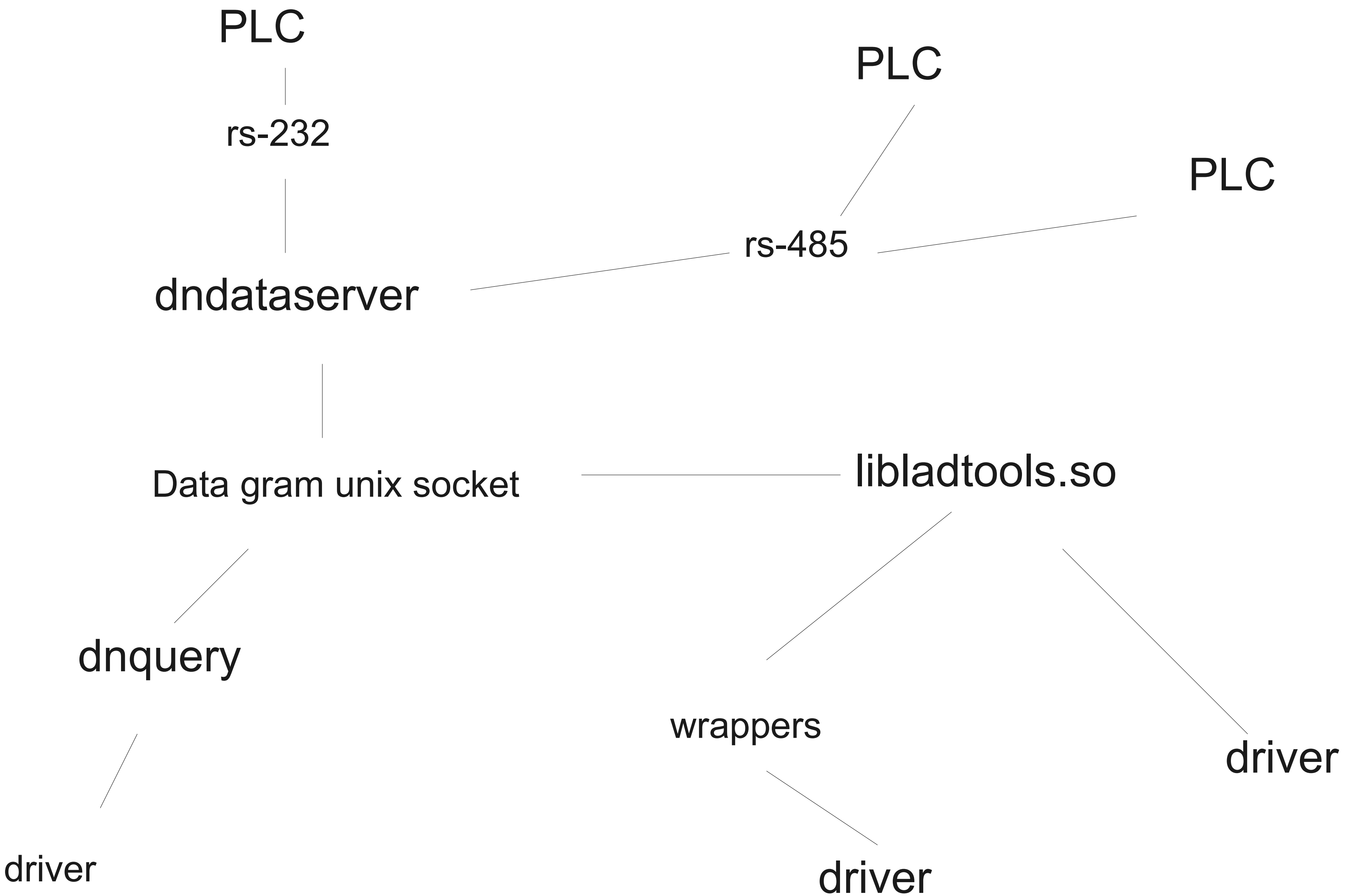
switch case, callbacks / delegates, Inheritance, thread synchronization

Hardware Logic PLC Logic IO Logic
PLC Driver Logic System Logic

Why serial?

- Legacy, Historical, “Standard”
- Proven, Robust, an enormous industry that keeps growing
- Less ambiguous security model
- Can leverage modern hardware: fiber, wireless, etc
- Works well in *nix
- API portability problems in general

Dataserver Model



PLC

rs-232

PLC

PLC

rs-485

dndataserver

Data gram unix socket

libladtools.so

dnquery

wrappers

driver

driver

driver

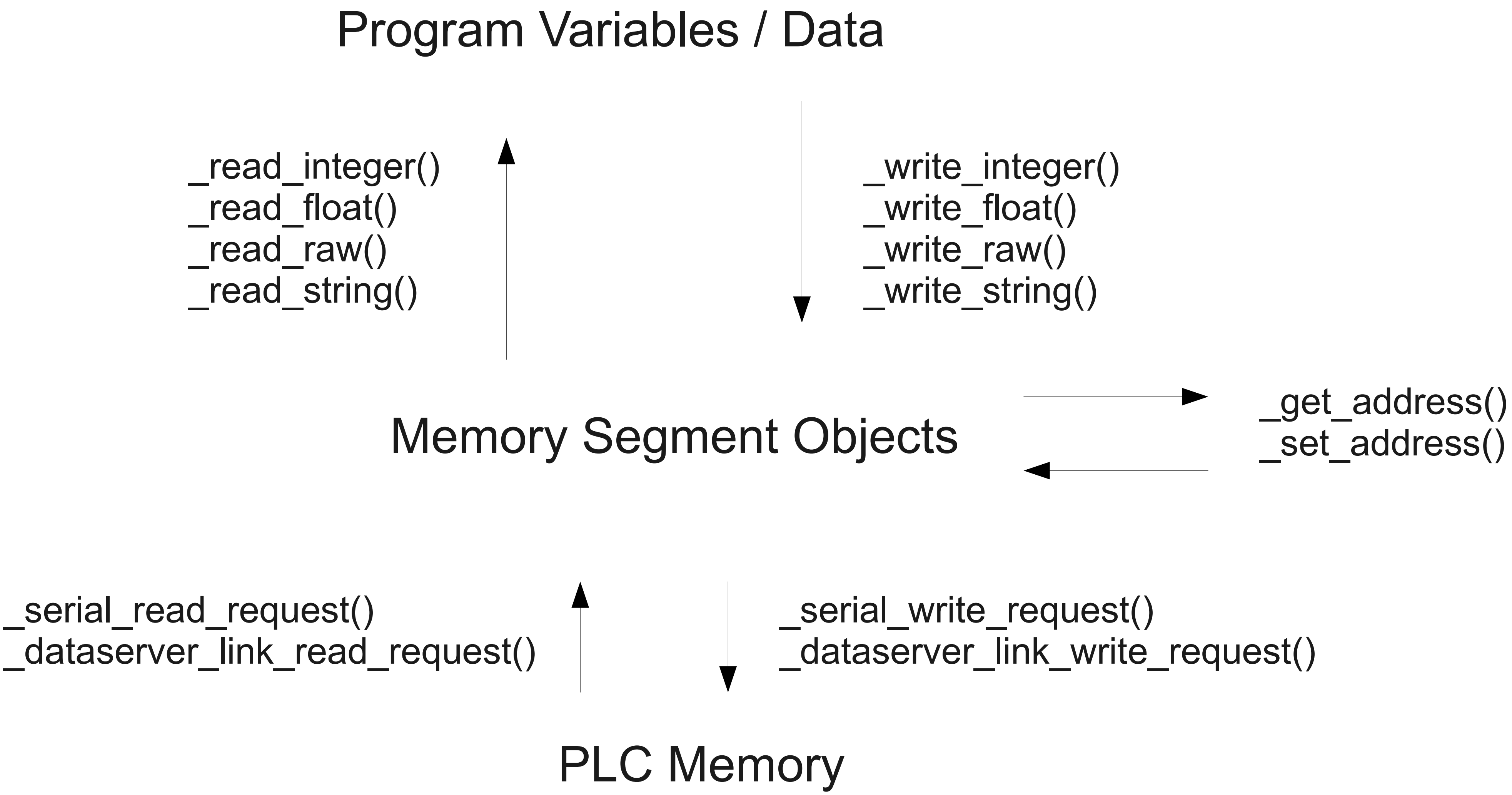
PLC Memory Model

- Flat
- Word addressable, not *byte* addressable
- Octal Notation
- Data in many formats, ie BCD
- “Special” facilities mapped to word memory

Why C?

- Best use of *nix shared library facilities
  - lib, bin, dev, doc package separation & versioning
- Low resource overhead
- “Wrapability”

LAD Tools API Model



Program Variables / Data

_read_integer()
_read_float()
_read_raw()
_read_string()

_write_integer()
_write_float()
_write_raw()
_write_string()

Memory Segment Objects

_get_address()
_set_address()

_serial_read_request()
_dataserver_link_read_request()

_serial_write_request()
_dataserver_link_write_request()

PLC Memory

LAD Tools API Events

- Read Completed
- Write Completed
- Progress Notify
- Error Notify (data server connections)
- Error Handler (serial connections)

- Realized inside Glib Main Loop

Why Glib?

OPC

- V1 = Windows Commercial
- V2? We'll see

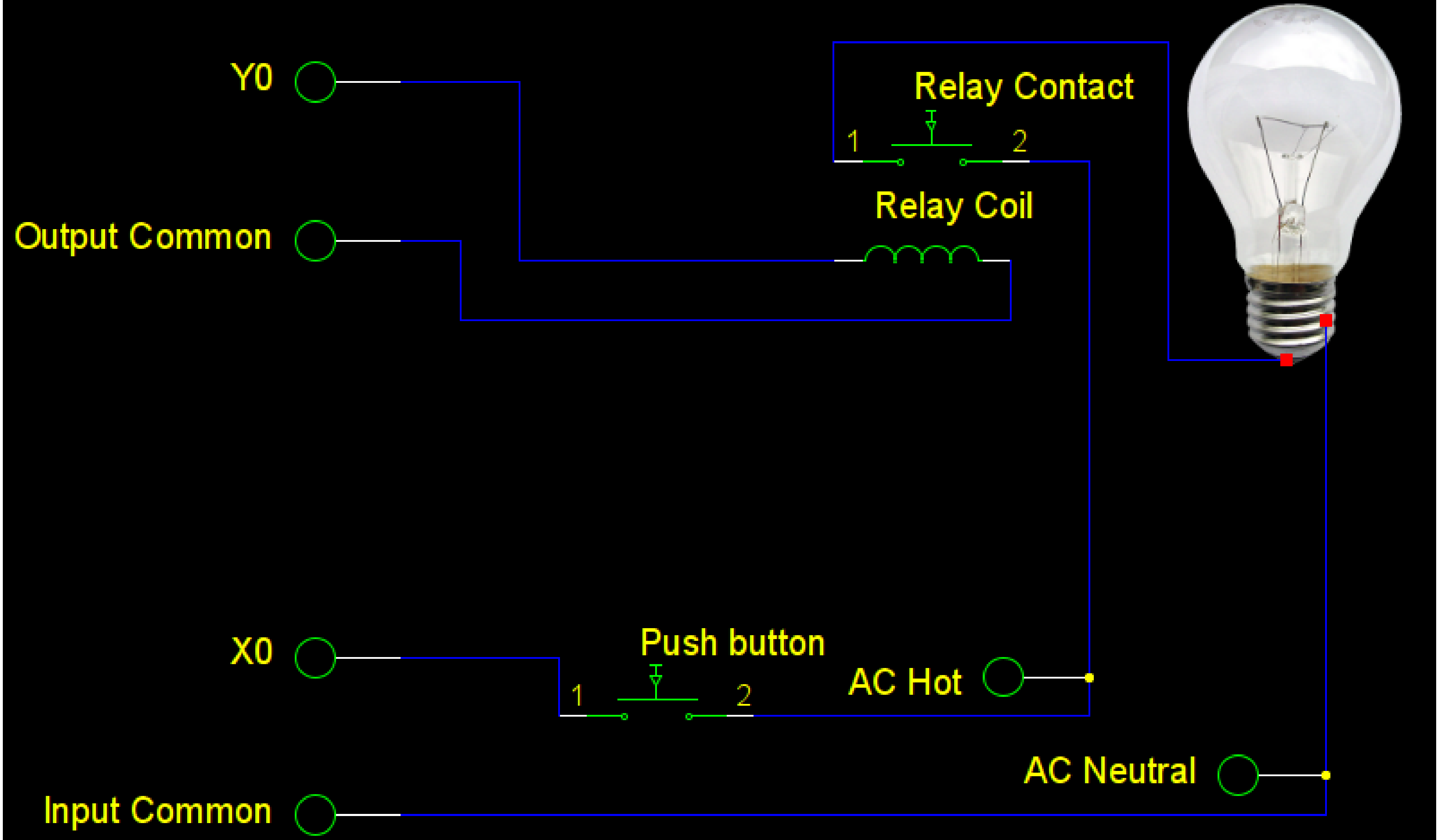
Some Integration Examples

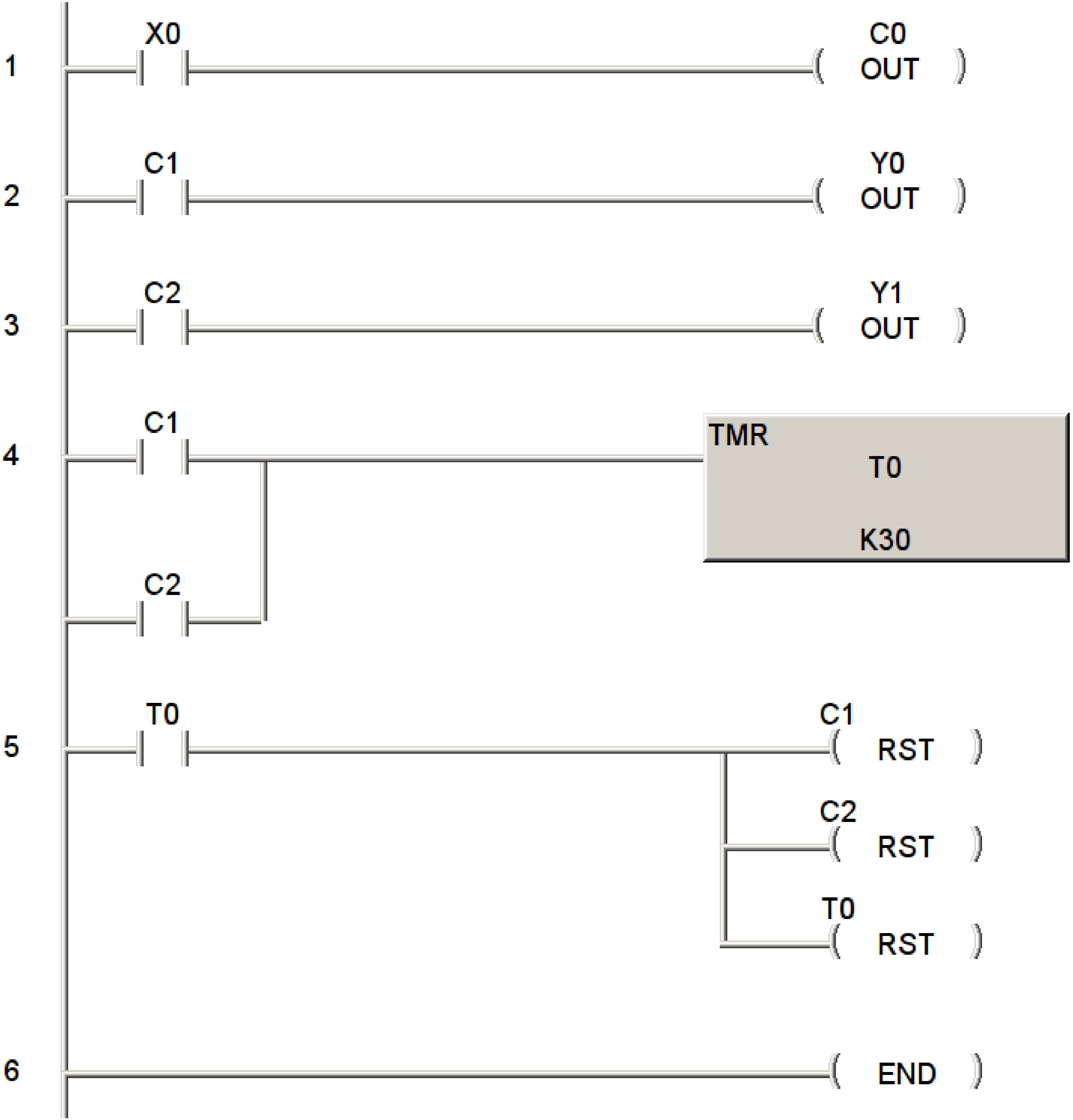
- Zabbix
- MRTG
- RRDB
- Mom & Pop Bash & Cron
- DomotiGa

Sensory Presence

- High Level Point Variable Abstraction
- Database Abstraction
- User Interface Abstraction
- Rendering & Parsing Abstraction
- Network IO Abstraction
- API & Tools

AC Output & AC Input PLC (or modules)



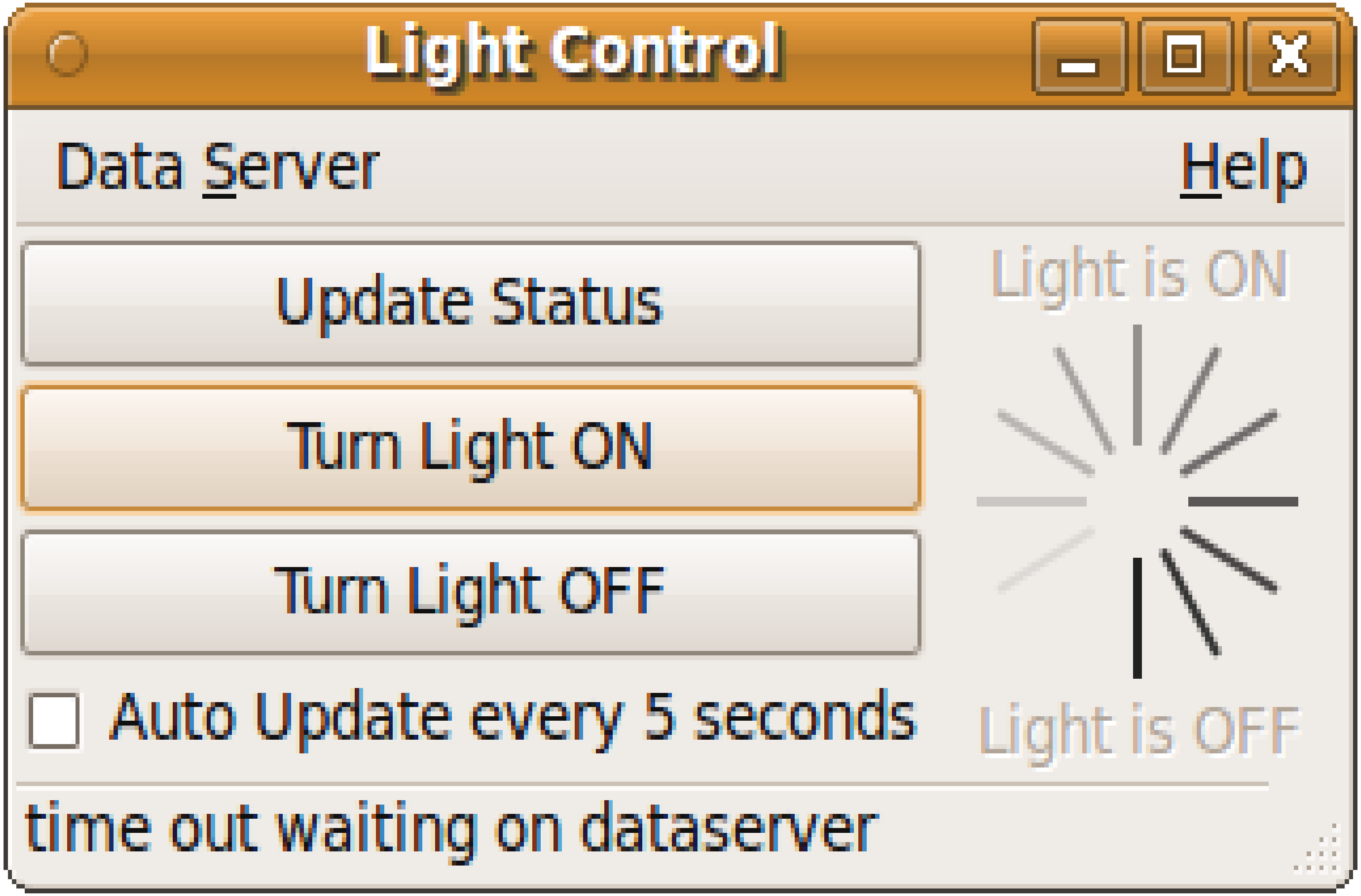

1
X0
C0
OUT
2
C1
Y0
OUT
3
C2
Y1
OUT
4
C1
TMR
T0
K30
C2
5
T0
C1
RST
C2
RST
T0
RST
6
END

Next, a quick demo was performed. A DL-06 with the above* program loaded, was connected to a USB serial dongle on /dev/ttyUSB0. The dataserver program was run in one window in foreground mode with debugging output turned on. Four example clients were shown: 1) the general purpose dnquerry utility, 2) a task specific command line program, 3) a task specific Gtk+ program, and 4) a task specific web program using libsoup. The next slides are terminal output samples & screen shots of those 5 items. The full source code will be available at http://www.thomasstover.com/ladtools/

*That was actually the wrong slide. The real program will be available with the project files on the web site.

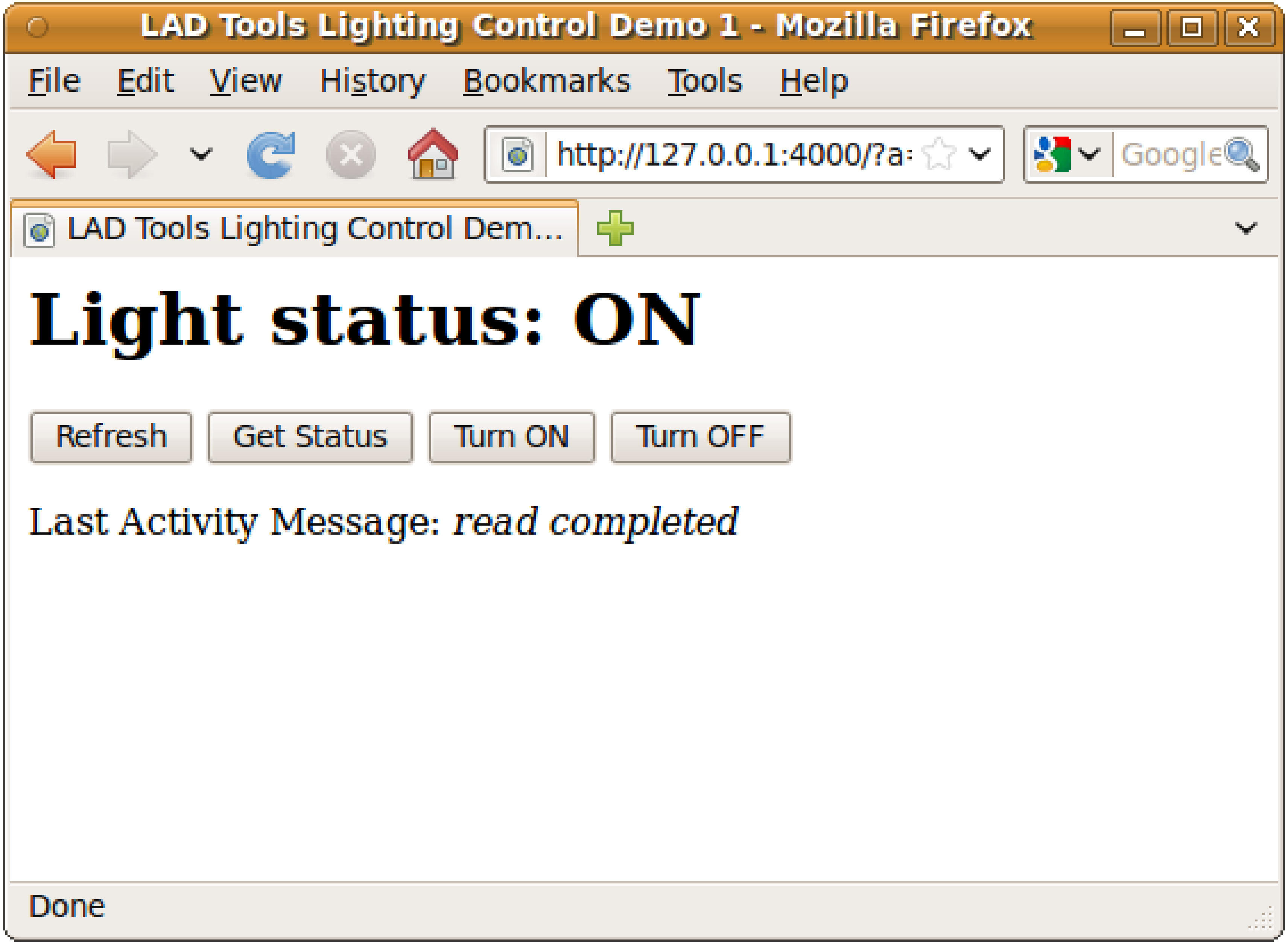
```
thomas@softtaco:~/ladtools$ LD_LIBRARY_PATH=./ ./dndataserver -v -N dev=/dev/ttyUSB0
Serial Network 0 parsed as: Device="/dev/ttyUSB0", BPS=9600, Retries=0, Timeout=0, Master
ID=0
socket_input_event()
read request: net=0, slave=1, priority=0, address="40600", length=2, context code=0
que_operation()
schedule_network_activity()
start_operation()
plc_read_completed()
issue_operation_success()
schedule_network_activity()
socket_input_event()
write request: net=0, slave=1, priority=0, address="40600", length=1
que_operation()
schedule_network_activity()
start_operation()
plc_write_completed()
issue_operation_success()
schedule_network_activity()
socket_input_event()
read request: net=0, slave=1, priority=0, address="40600", length=1, context code=0
que_operation()
schedule_network_activity()
start_operation()
plc_read_completed()
issue_operation_success()
```

```
thomas@softtaco:~/ladtools$ LD_LIBRARY_PATH=./ ./lightingdemo1cl -f
thomas@softtaco:~/ladtools$ LD_LIBRARY_PATH=./ ./dnquery -u ./LADTOOLSDATASERVER -a
40600 -r 1 --RAW --bin
00000000
thomas@softtaco:~/ladtools$ LD_LIBRARY_PATH=./ ./lightingdemo1cl -o
thomas@softtaco:~/ladtools$ LD_LIBRARY_PATH=./ ./dnquery -u ./LADTOOLSDATASERVER -a
40600 -r 1 --RAW --bin
00000001
```



□ Auto Update every 5 seconds

time out waiting on dataserver



LAD Tools Lighting Control Demo 1 - Mozilla Firefox

Light status: ON

Last Activity Message: *read completed*