Raspberry Pi

- We will play a bit with the GPIO pins of the Pi to make cool things
- We will show how to light up LEDs and measure temperatures
- I'd like you to get to "tinker" – you are engineers and scientists after all…

Controlling hardware is hard

- Have you ever tried to control some actual piece of hardware with your laptop?
- It can do all kind of useful things, such as showing this PowerPoint presentation, but it doesn't have any "pins" to, say, turn some light on
- Old, 1995-vintage PCs sometimes still had a parallel port that could somehow do that, with some effort
- Throw in the Raspberry Pi with lots of "General Purpose I/O" ("GPIO") pins

Hardware control is back…

- For the first time in a long while, we have a cheap (<R500 or <R200) linux-running single-board computer with lots of physical I/O capabilities
- You can use this for all kinds of really interesting projects
- We will concentrate here on the RPi, but let me point out another device for applications where even a RPi seems overkill:

The ESP8266

This chip was made to retrofit a Arduino board with wireless capabilities

Turns out it can be used as an Arduino in its own right – same pins, same processor, etc

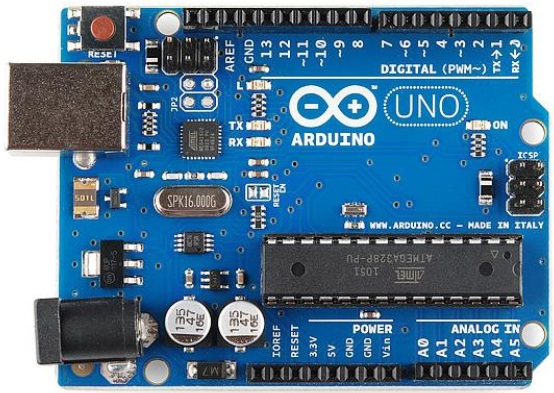
For R25 ($1.50), you can have a wireless-capable single-purpose unit to control stuff

The larger -12E version adds USB support and more pins (~R80)

Back to Raspberry Pis…

ESP8266

ESP8266-12E



Arduino

The Raspberry Pi

Conceived by a non-profit “Raspberry Pi Foundation” to bring “tinkering” back to us

(this goes against the trend set by Apple, MicroSoft, and also Google to make everything “sealed” and tinker-proof)

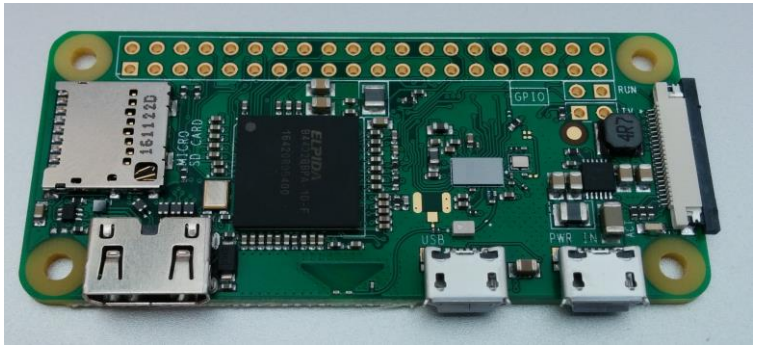
Here have a totally open Linux environment that can be used to learn how computers work on the inside

It is cheap – with a case and (cell-phone style) power supply it costs less than R500

And it has full networking, USB, sound output, HDMI video, a camera port, and lots of GPIO pins that you can use in many different ways

Standard RPi Linux distros come with built-in support for many sensor protocols out of the box (one-wire, infrared, etc etc)

And the little brother….



Raspberry Pi Zero W

½ size

½ power

But with wireless and Bluetooth on board

Even cheaper - ~ R200

My house is run by RPis

Long ago I decided to make my house (especially the heating system) computer-controlled

It has “3-zone heat” – meaning 3 circulation pumps that pump hot water upstairs, ground floor, basement when the thermostats say “heat, please!”

A dumb thermostat is a really bad solution – even if it claims to be “programmable”

I disconnected the 3 original thermostats and replaced them with at least one temperature sensor in each room

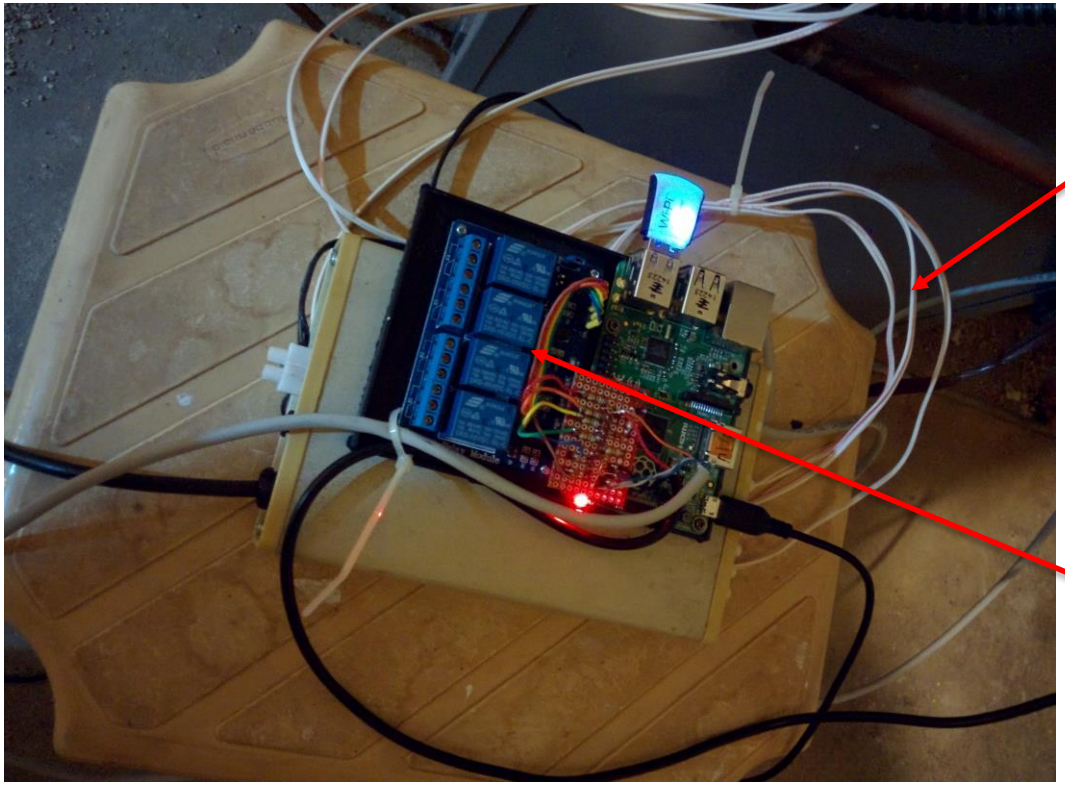
More “sensory input” means better control

Master controller is aware if someone is actually at home

Daily schedule infinitely programmable

Furnace Control

This is the “master module” that controls the circulation pumps



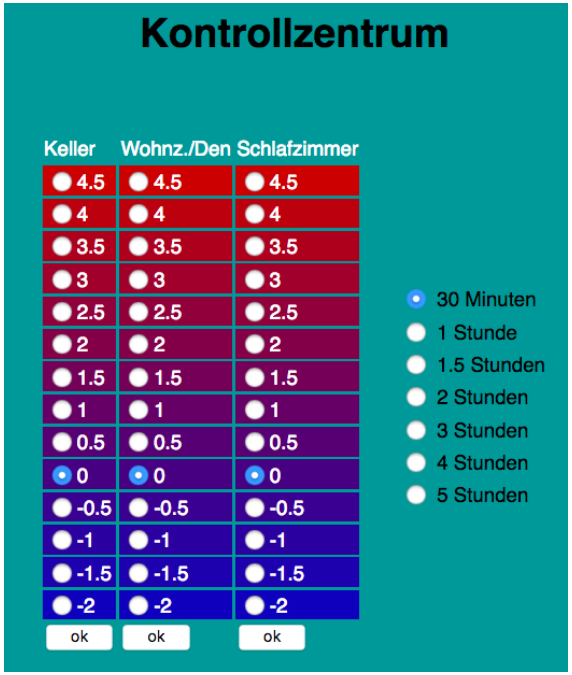
3 lines going to the pumps

(picture was taken before I switched to the new RPi unit)

Has been at work now for almost a year w/o glitch

Relais where the pump wires are now connected
Controlled by the Rpi GPIO pins

Web Controls



Runningtime: 0:6.08333
Yesterday: 0:21.4333
Before: 0:22.2167

The system is aware when the burner is actually firing –
I can calculate my heating costs every day

Compared to the “dumb” system I save about 18% in heating costs
(measured by ½ winter of running the system in “observer mode” only)

Other Controls & Sensors @Home

System also controls the lig

They turn off when no one

No more lights left on unles

Also runs a water alarm for

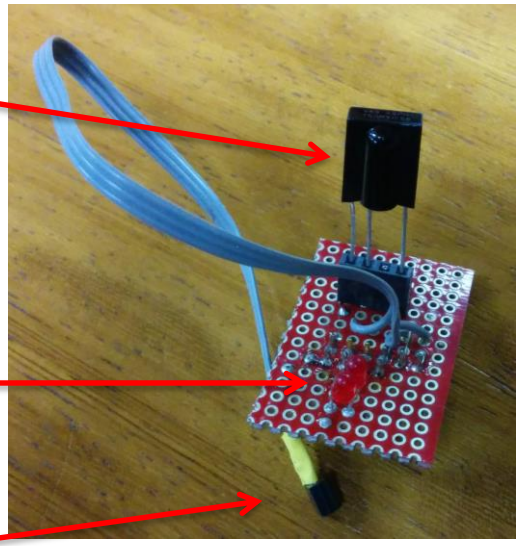
Water sensor alarm sendin

Turns a light on and takes a
can recognize false alarms

Saved the day (and a costly

Raspberry Pi “Sensor” package

- This is a small 20x30mm^2 hand-made board (I made it myself in an hour)
- It has a header for an infrared receiver that allows to control things with a remote
- It has two LEDs that can be turned on and off through GPIO pins
- it has a temperature readout



Sensor Input

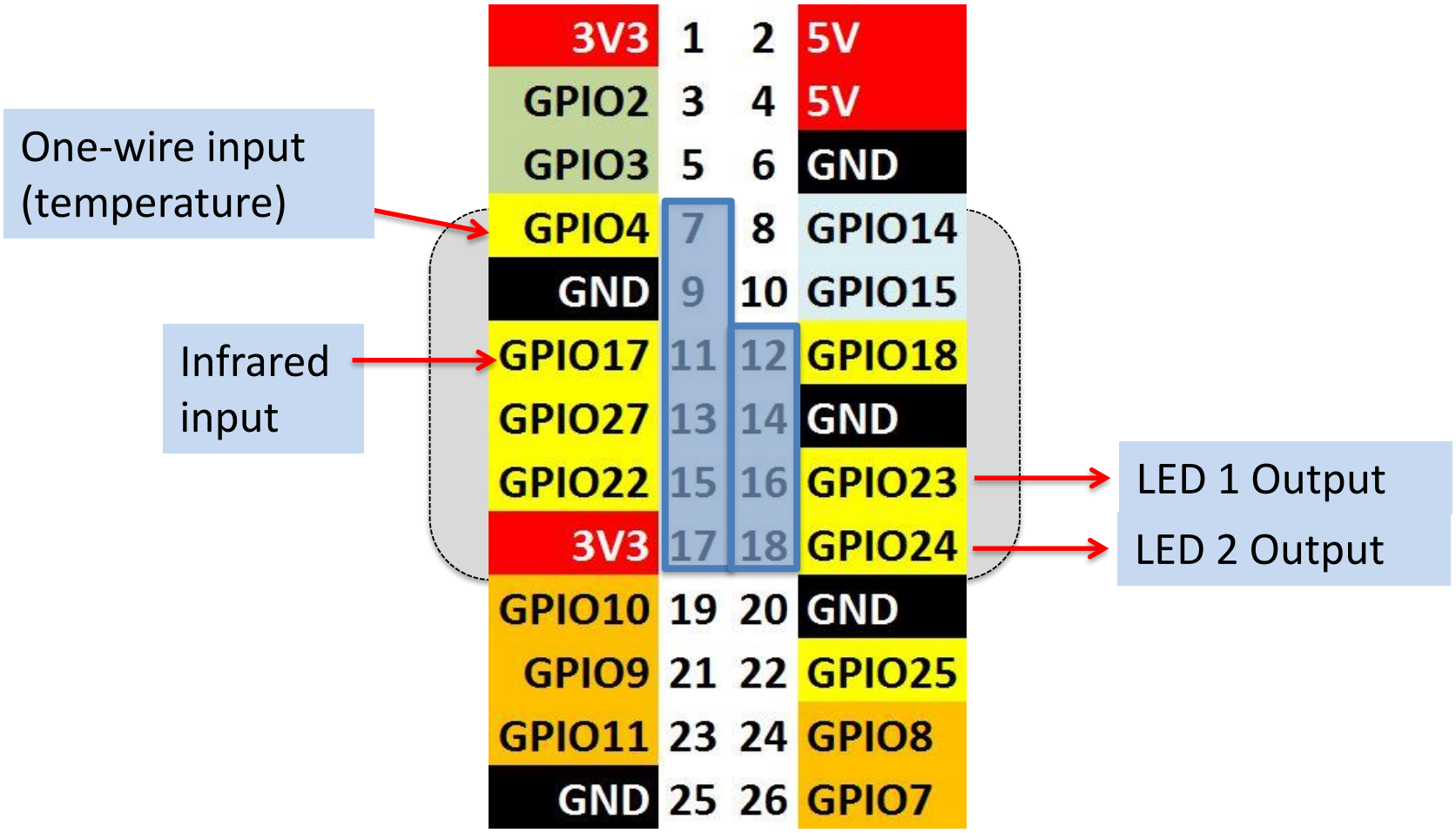
- The Pi has software out of the box to read out one-wire “Dallas” protocol sensors
- We use it to read DS18B20 digital thermometers (by default on GPIO pin 4)
- We read a TSOP 312 infrared receiver (on GPIO pin 17 – default input)
- There two LEDs using GPIO pins 23 and 24

DS18B20

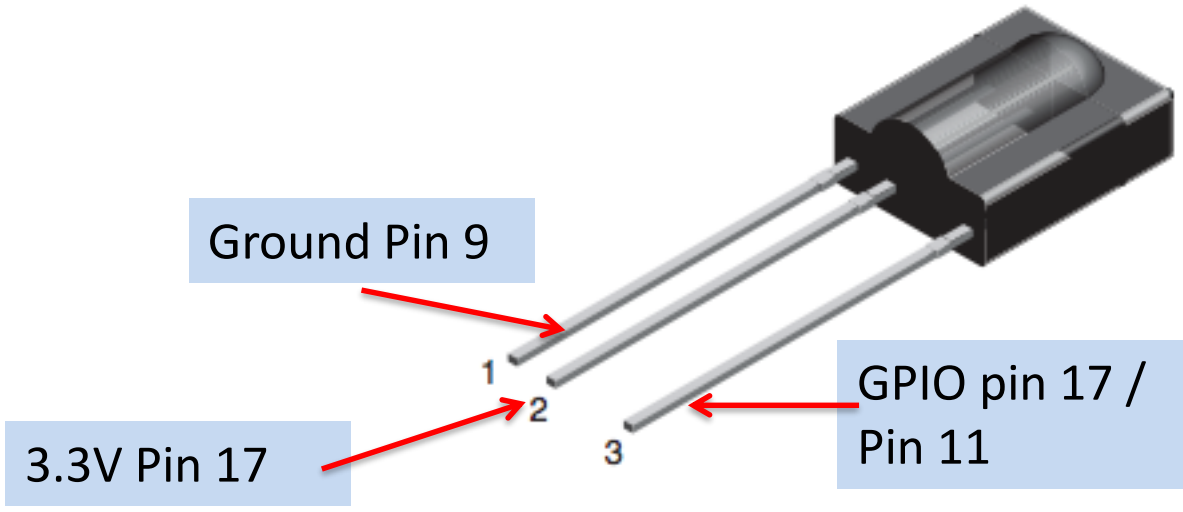
TSOP 312

Please distinguish between GPIO pin numbers (on the chip -> function) and the pin number on the header. E.g. GPIO pin 4 is pin 7 on the header.

Connections



Infrared Circuit



MECHANICAL DATA

Pinning:

1 = GND, 2 = V_S, 3 = OUT

Temperature Circuit

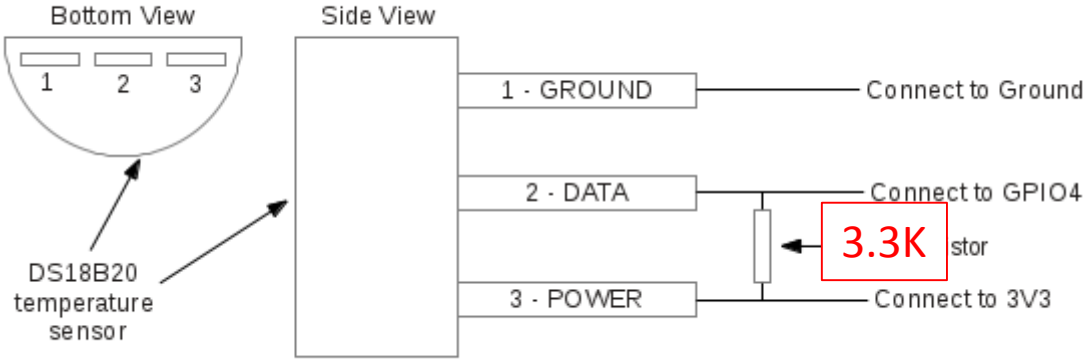
DS18B20

3.3K Resistor between 3.3V and output

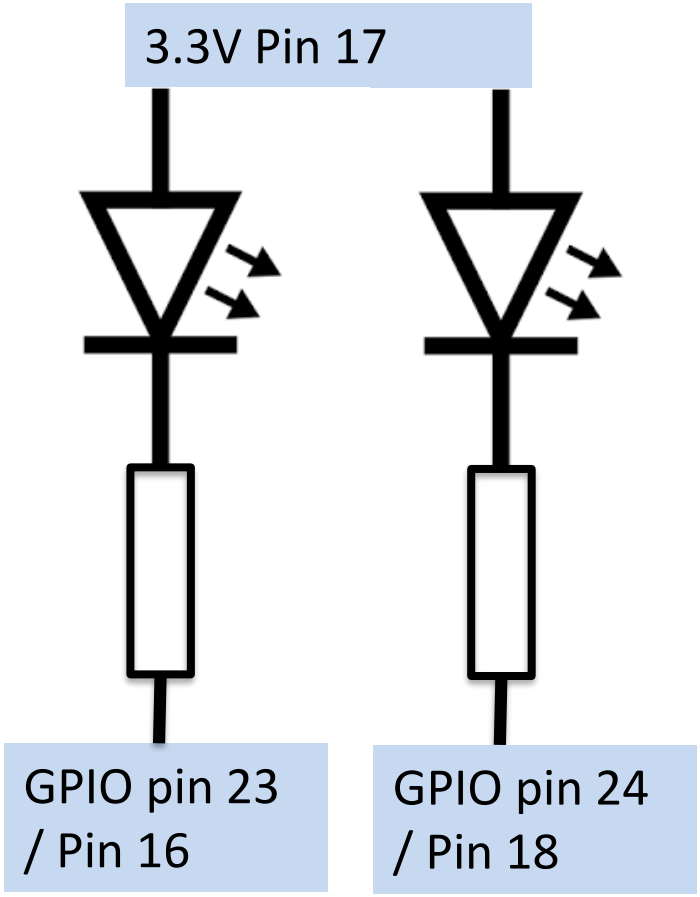
Ground Pin 9

3.3V Pin 17

GPIO pin 4 / Pin 7



LED 1 and 2




```
echo 23 > /sys/class/gpio/export
echo  out > /sys/class/gpio/gpio23/direction

echo 1  > /sys/class/gpio/gpio23/value  #turn on
echo 0  > /sys/class/gpio/gpio23/value  # turn 0ff
```

Enabling one-wire protocol

This enables our temperature reading with the DS18B20, which talks one-wire protocol

Edit /boot/config.txt and add

```
dtoverlay=w1-gpio
```

Then reboot

Software (DS18B20)

Each DS18B20 sensor has a unique and unchangeable 64bit serial number assigned by the manufacturer.

It shows up as
/sys/devices/w1_bus_master1/28-xxxxxxxxxxxxx/
where 28- is the device type and xxxxxxxxxx is the serial number

```
# cat /sys/devices/w1_bus_master1/28-0000065e5516/w1_slave
31 01 4b 46 7f ff 0f 10 1f : crc=1f YES
31 01 4b 46 7f ff 0f 10 1f t=19062
```

This is the temperature in “milli-Centigrade”
(e.g. 19062 deg. C = 66.31 F)

```
# grep t= /sys/devices/w1_bus_master1/28-0000065e5516/w1_slave | awk -F= '{print $NF}’
19062
```

One more shell command

```
# cat /sys/devices/w1_bus_master1/28-0000065e5516/w1_slave
31 01 4b 46 7f ff 0f 10 1f : crc=1f YES
31 01 4b 46 7f ff 0f 10 1f t=19062
```

```
grep t= /sys/devices/w1_bus_master1/28-0000065e5516/w1_slave |
awk -F= '{print $NF}’
```

Enabling Infrared

Edit /boot/config.txt and add

```
dtoverlay=lirc-rpi,gpio_in_pin=17,gpio_in_pull=high,gpio_out_pin=18
```

Then reboot

Software (Infrared)

1) apt-get install lirc (Linux Infrared Controller)
2) Edit /etc/lirc/hardware.conf to read

```
#Try to load appropriate kernel modules
LOAD_MODULES=false
# Run "lircd --driver=help" for a list of supported drivers.
DRIVER=""
# usually /dev/lirc0 is the correct setting for systems using udev
DEVICE="/dev/lirc0"
```

“mode2” refers to the infrared protocol which is virtually always used

mode2 –d /dev/lirc0
Will show you raw pulses from the transmitter

The “Lirc Daemon” lircd interprets the pulses and can issue commands and trigger actions

Remote-controlling cool stuff

I use a “leftover” remote control to do some interesting things

At home I control various lights with a n Infrared Remote that once belonged to a VCR (long gone)

I really don’t know wat this one belonged to

But pretty much any Remote will do

You can hook up arbitrary commands to any button

I made it so that Button “1” is switching on and off the one, and “2” the other LED…

But I can take it further…

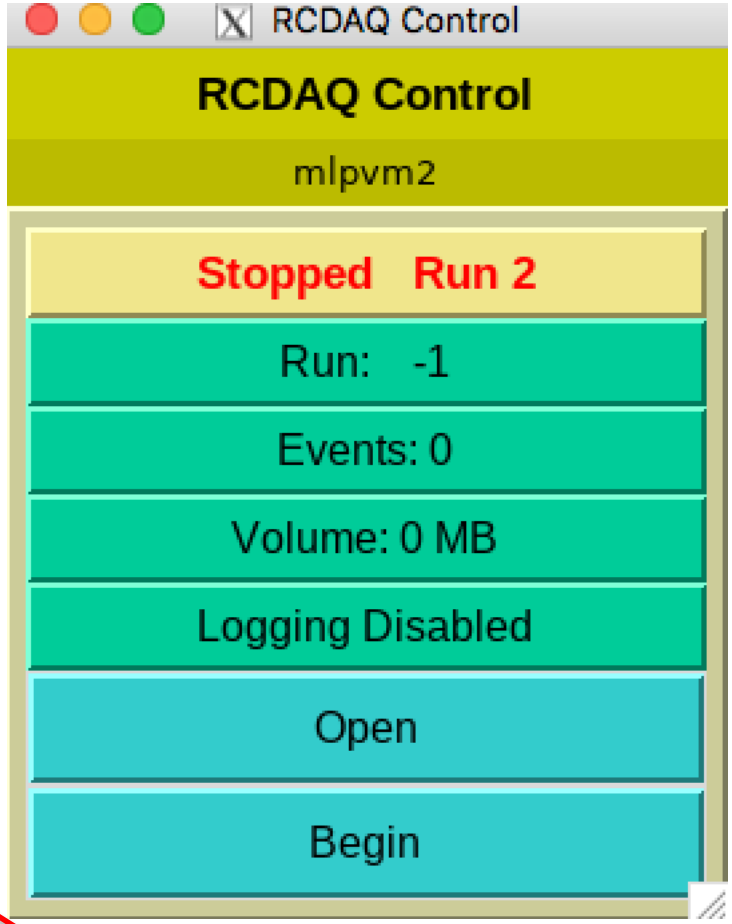
Did I mention that the rcdaq control commands work through the network? (I know I did ☺)

I make it so that the Rpi's rcdaq commands operate on my Mac (rcdaq is running in a virtual machine there)

I hooked the "End run" to the "power off" button

And the "Begin" command to the "Record" button

Daq_open and close to CH+ and CH-…



It’s really easy…

This is part of the rcdaq setup

```
begin
prog = irexec
remote = sradio
button = KEY_RECORD
repeat = 2
config = /root/rcdaq.sh daq_begin
end

begin
prog = irexec
remote = sradio
button = KEY_POWER
repeat = 2
config = /root/rcdaq.sh daq_end
end
```

And this deals with the LEDs…

```
begin
prog = irexec
remote = sradio
button = KEY_1
repeat = 2
config = /root/toggle.sh 0
end

begin
prog = irexec
remote = sradio
button = KEY_2
repeat = 2
config = /root/toggle.sh 1
end
```

Controlling stuff is cool

You are engineers, scientists…

Please – tinker! If you are only learning from books, and only do exercises prescribed by your professor, you are missing things

Think of a musician who has never learned and played anything that wasn't part of the curriculum – probably not someone you want in your orchestra!

I hope I have made you curious

There are plenty of commercial, cheap sensor boards out there

I have built **all** of mine myself - more fun.