MONITORING EVERYTHING THAT ISN'T A SERVER

With python, a RaspberryPi, and some bits

server density

Tom Wardill
github.com/tomwardill
@tomwardill

http://www.flickr.com/photos/dirkdekok/5588801438/

SERVER MONITORING IS EASY

www.serverdensity.com

EDUCATION

Software is boring

WHY PYTHON

- Ubiquitous
- Cheap
- Library Support
- External Dependencies

THINGS YOU CAN MONITOR

Work

Location

House

Activity

Health

Others that I can’t think of, but are obvious to someone else.

WORK

- Easy
- Techniques:
    - Git Post Commit Hooks
    - File Changes
    - IM Messages
    - Emails

LOCATION

- Also relatively easy
- Direct Location
  - GPS
  - Location Services
- Delayed / Manual Location
  - Foursquare
  - Latitude

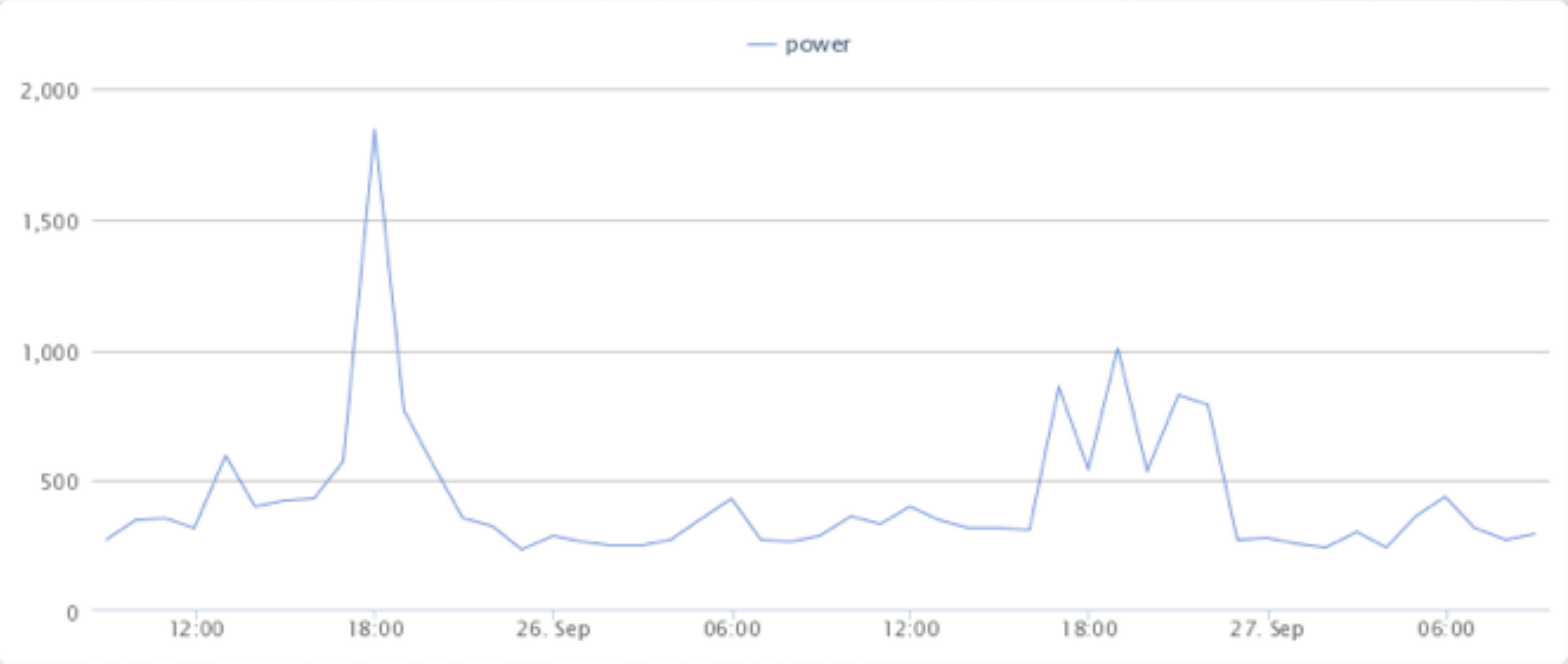
PRESENCE

HOUSE

- Power
- Gas
- Temperature
- Door Openings
- Cat activity
- Plenty of opportunities for monitoring

HOUSE

- Power
- Gas
- Temperature
- Door Openings
- Cat activity
- Plenty of opportunities for monitoring



ACTIVITY

- Sports
  - Runkeeper
  - Strava
- Cups of tea

HEALTH

- Weight
- Heart Rate
- Blood Pressure
- FitBit

TYPES OF DATA

- Float
- Location
- Boolean
- Instantaneous Time
- Time Series

COMPARING DATA

- Make sure you have accurate time

- Time Scale

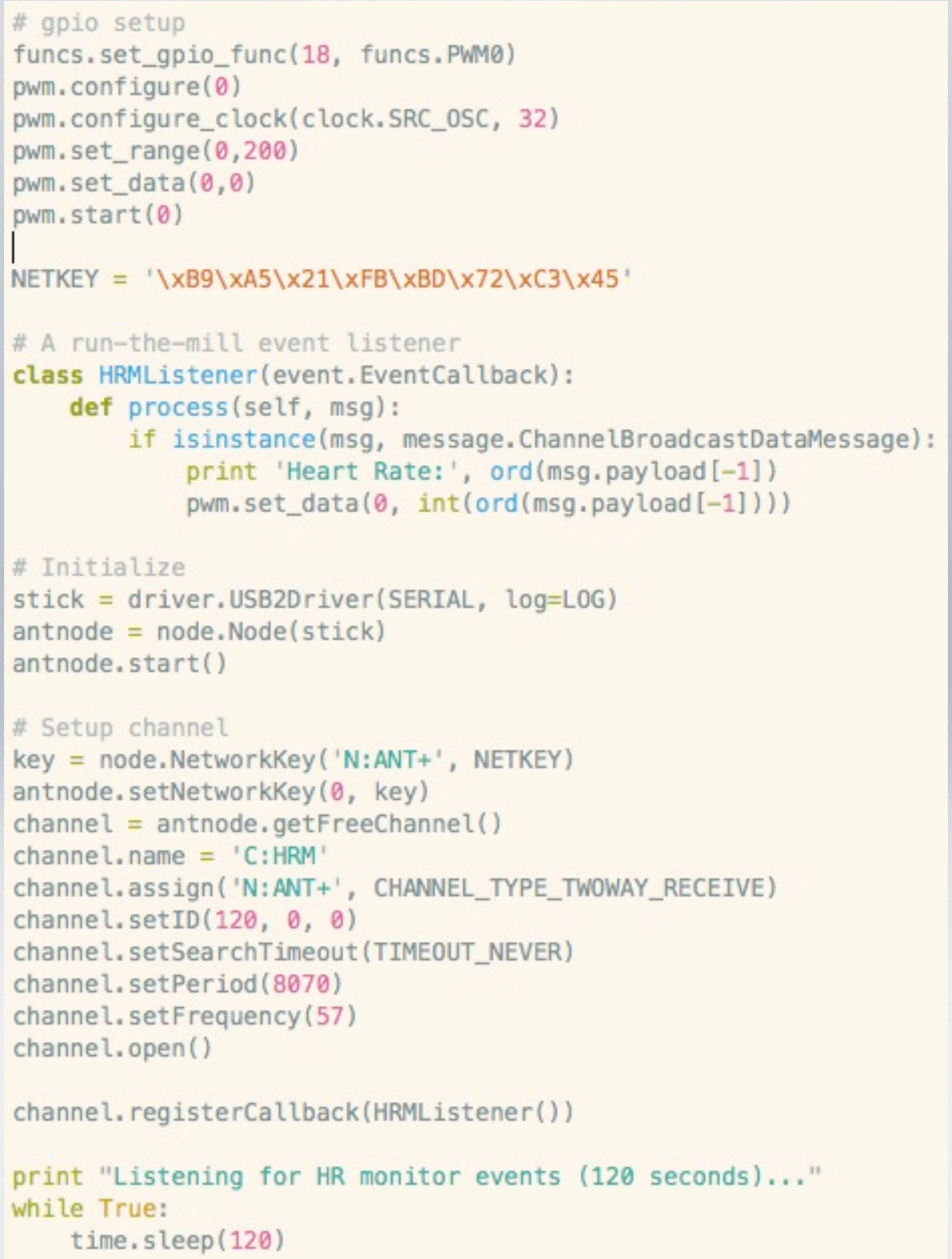
- Any time series is instantaneous if the scale is small enough.

AUTOMATING IT

http://www.flickr.com/photos/gijsbertpeijs/7988262046/

↔

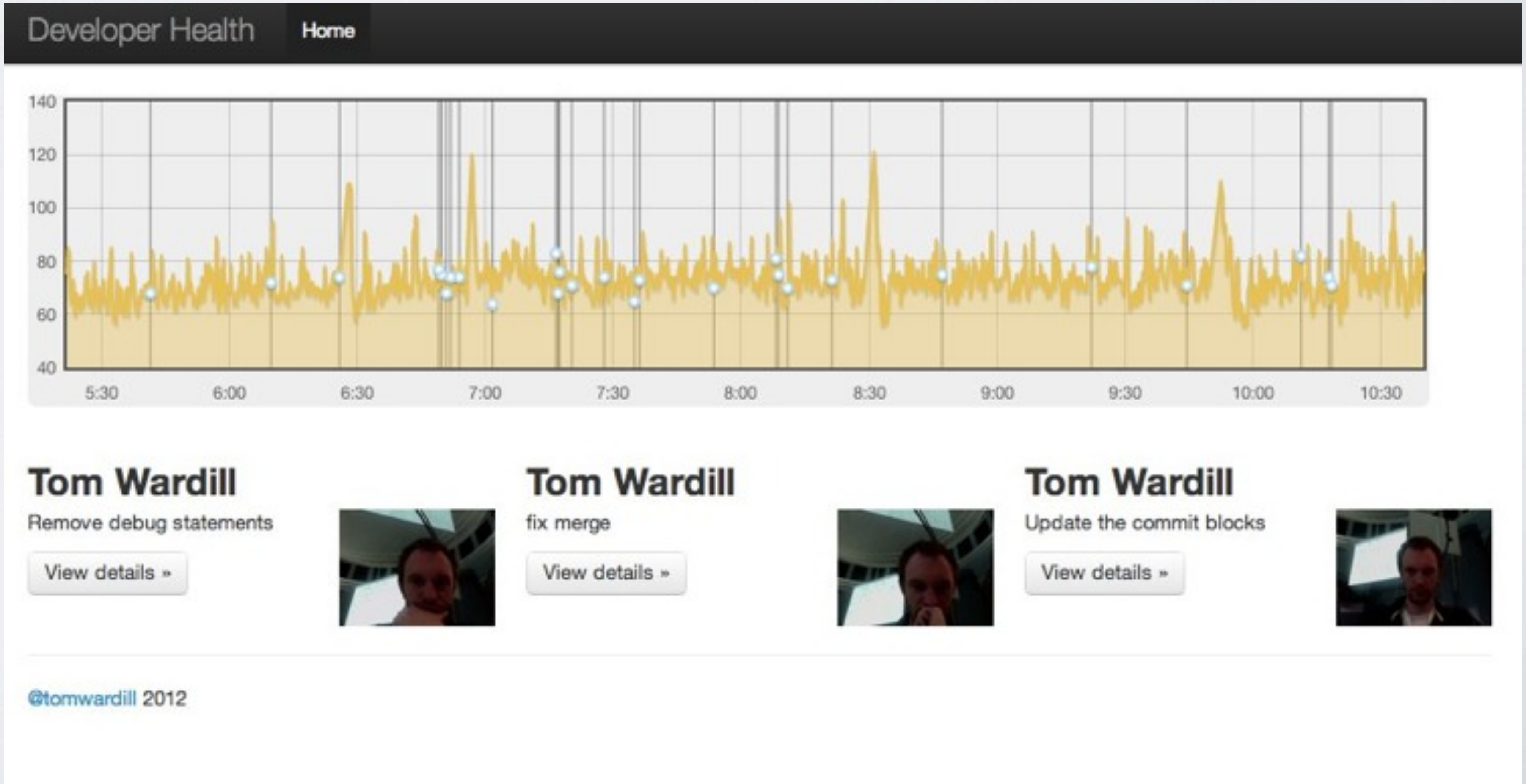
http://www.flickr.com/photos/snootlab/6052455554/



```python
# gpio setup
funcs.set_gpio_func(18, funcs.PWM0)
pwm.configure(0)
pwm.configure_clock(clock.SRC_OSC, 32)
pwm.set_range(0,200)
pwm.set_data(0,0)
pwm.start(0)

NETKEY = '\xB9\xA5\x21\xFB\xBD\x72\xC3\x45'

# A run-the-mill event listener
class HRMListener(event.EventCallback):
    def process(self, msg):
        if isinstance(msg, message.ChannelBroadcastDataMessage):
            print 'Heart Rate:', ord(msg.payload[-1])
            pwm.set_data(0, int(ord(msg.payload[-1])))

# Initialize
stick = driver.USB2Driver(SERIAL, log=LOG)
antnode = node.Node(stick)
antnode.start()

# Setup channel
key = node.NetworkKey('N:ANT+', NETKEY)
antnode.setNetworkKey(0, key)
channel = antnode.getFreeChannel()
channel.name = 'C:HRM'
channel.assign('N:ANT+', CHANNEL_TYPE_TWOWAY_RECEIVE)
channel.setID(120, 0, 0)
channel.setSearchTimeout(TIMEOUT_NEVER)
channel.setPeriod(8070)
channel.setFrequency(57)
channel.open()

channel.registerCallback(HRMListener())

print "Listening for HR monitor events (120 seconds)..."
while True:
    time.sleep(120)
```

HOST SOFTWARE

Take Data
Make Graph



Tom Wardill
Remove debug statements

Tom Wardill
fix merge

Tom Wardill
Update the commit blocks

@tomwardill 2012

BUTTONS

- Easy
- Simple to monitor
- Cheap

RFID

- Oyster Card
- RFID Stickers
- Measure Presence

http://www.flickr.com/photos/raver_mikey/7171051278/

POWER / TEMPERATURE

- Current Cost
- Google Power
- Other Monitors

www.currentcost.com

FEEDBACK

- Graphs
- Charts
- Lights
- Sounds
- Email
- SMS
- Dials
- Images
- Twitter
- Facebook
- THE INTERNETS

MONITOR YOUR TENT IN A FIELD

TOTAL COSTS

Raspberry Pi	£30
TouchaTag	€22.50
Dials	£4
Buttons	£7
HRM + ANT+	£22 + £18
Bits (Wires, etc)	£10