Sensors 2

David Andersen
Low-Power Computing
Carnegie Mellon University

Before we start...

- Some ?s from last time about the lock/ power interface - why is this useful, etc?
- Some context: Programming an ATMega mote

Sample AtMega Code

```
/*
 * ms_sleep() - delay for ms milliseconds
 */
void ms_sleep(uint16_t ms)
{
  TCNT0  = 0;
  ms_count = 0;
  while (ms_count != ms)
    ;
}
```

```
SIGNAL(SIG_OUTPUT_COMPARE0)
{
  ms_count++;
}
```

Timer

```
int main(void)
{
  init_timer();
  sei(); /* Enable interrupts */
  DDRB = 0x01;  /* enable PORTB 1 as an output */
  while (1) {
    ms_sleep(512);    /* wait 0.5 seconds */
    PORTB ^= 0x01;    /* toggle LED */
  }
}
```

```
configuration BlinkAppC{
}
implementation {
  components MainC, BlinkC, LedsC;
  components new TimerMilliC()
             as Timer0;

  BlinkC -> MainC.Boot;

  BlinkC.BlTimer -> Timer0;
  BlinkC.Leds -> LedsC;

}
```

```
module BlinkC {
  uses interface Timer<TMilli>
           as BlTimer;
  uses interface Leds;
  uses interface Boot;
}
implementation{
  event void Boot.booted() {
    call BlTimer.startPeriodic(1000);
  }
  event void BlTimer.fired() {
    call Leds.led0Toggle();
  }
}
```

Slide credit: Nicolas Burri

Profiling and Simulation

- We'll return to this issue a bit later - PowerSCOPE, etc., provide tools for monitoring power consumption and attributing it to a system component
- For now: Think about toolbox of analysis methods (useful for projects.)
  - Cycle-accurate sim (Atemu; Bochs); Native code sim (TOSSIM; VMWare); Implementation on real h/w; hardware-assisted sim (maybe later - for high-perf arch).

TOSSIM

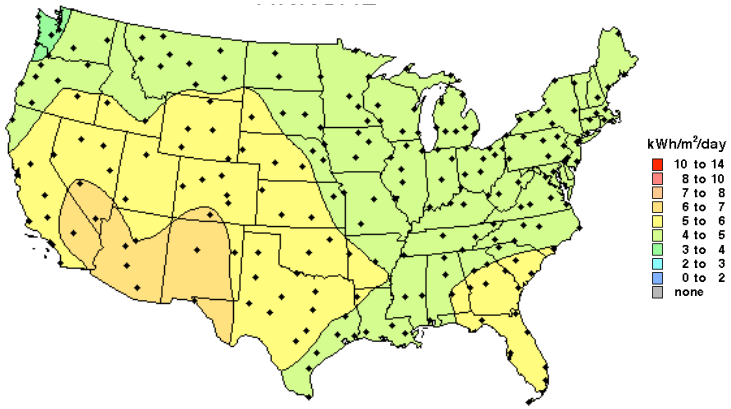
- More value from OS abstractions
  - calling Leds.redOn() easier to simulate than inferring that, for particular h/w,
  - PORTB ^= 0x01 (flip bit 1 on PORTB register, which activates output foo, iff PORTB set to output, and, and, ...)
- But for power - had to do basic-block analysis to *estimate* cycle counts.
- May not account for unanticipated h/w quirks
  - Pretty good sim, and *very* useful, but beware
  - Rod Brooks: "Simulation is doomed to succeed."

Capturing Solar

- 1) Amount of solar radiation
- 2) Efficiency of solar cell
- 3) Angle of solar cell
- 4) Area of solar cell

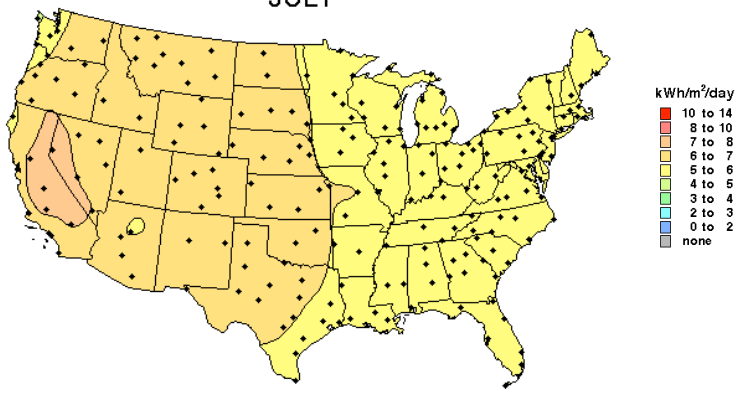
How much solar?

- Solar cell efficiency: ~16-28%, varies by cost/ weight. (Thin-film - ~16, polysilicon ~25).
- Sun-tracking? Nah - too much work for little sensors... so we lose angle of incidence. Lose about 30-40% of efficiency -> 10-18% *ish*

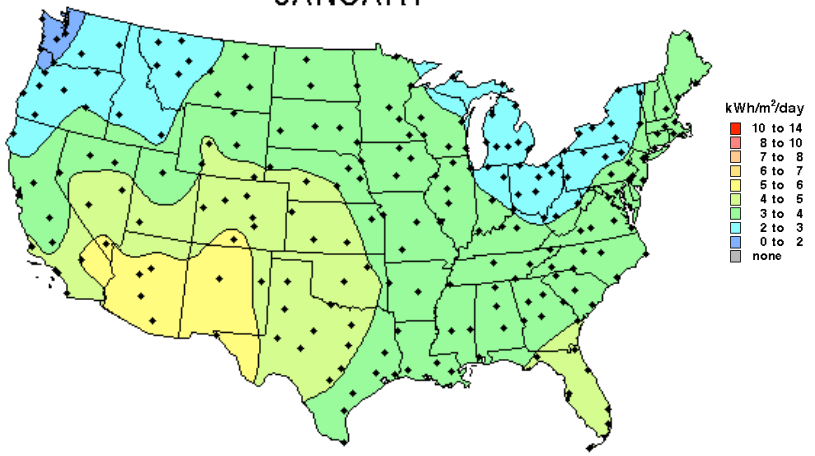


Average Solar radiation - flat plate tilted south

JULY



JANUARY



Three Scenarios
Three Groups. :)

- Meraki Mini wireless repeater
  - 200mW radio, ~2W average draw, ~7W max (e.g.,, when downloading & writing new OS to flash and SHA-1'ing it, etc.)
- Wireless temperature monitoring - taking measurements 2x per second and reporting them every 5.
- Wireless web cam takes photo every 2 minutes and uploads via 802.11.

Process

- Those workloads are vague.
  Ask for clarifications and/or state assumptions.
  Cost? Where deployed? Lifetime? Achievable duty cycle (what factors in?); etc.
- Design & analyze - solar feasible? y/n; size of cells? size of battery? capacitor? changes to workload?
- Present! Describe scenario w/assumptions, results
- Critique - the other groups' solutions. :)

How Much Solar?

- Solar radiation received, avg, Pittsburgh *in January*:
  - 2-3 kWh / m^2 / day
- At 10% efficiency... 250 Wh per square meter per day.
- 50mW sensor: 1 Wh per day.
  - 2 foot x 2 foot panel. (!)
- Rats. Still need to sleep...
  - Ex from paper 1: 20% duty cycle (200 mWh / day)
  - 30cm x 30cm would do it (paper used 16cm x 10cm)

Example: Meraki Solar

- Meraki Mini wireless mesh node
- Really handy to be able to deploy with no installation effort -- it's already wireless...

Meraki Solar

- 200mW radio
- 7W max / 2W typical power draw
- 24 hours / day: ~50Wh
- Cloudy days in pgh...
  - 40W panel: 25” x 22” x 1.4”
  - $1497. (!)