Open Source Building Science Sensors for Indoor Microbiology

Indoor Air 2014 | Hong Kong

Brent Stephens[1], Akram Ali[1], Deion Debose[1], Boyang Dong[2], Torkan Fazli[1]

[1]Civil, Architectural and Environmental Engineering, Illinois Institute of Technology, Chicago, IL

[2]Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL

OPEN SOURCE
BUILDING SCIENCE SENSORS

ILLINOIS INSTITUTE
OF TECHNOLOGY

The Built Environment Research Group

advancing energy, environmental, and sustainability
research within the built environment
at Illinois Institute of Technology

web www.built-envi.com **email** brent@iit.edu **twitter** @built_envi

Motivation

- Recent studies have greatly increased our knowledge of microbial ecology of the indoor environments in which we live and work

Summarized in Kelley and Gilbert **2013** *Genome Biol* 14:202; Konya and Scott **2014** *Curr Sustain Energy Rep* 1:35-42

- Limited collection of long-term building science data to date
- Insufficiently described built environment metadata (or more accurately built environment *data*) can limit our ability to compare microbial ecology results from one indoor environment to another
  - Or assess how best to control indoor microbial communities

Two related efforts

1. Review of recent indoor microbial literature through the lens of a building scientist
   – And recommendations for tools for microbial ecologists to incorporate more building science measurements in their work

2. OSBSS: Open Source Building Science Sensors
   – An effort to design and build open source building environmental sensors for microbiology studies

Literature review of recent MoBE investigations

We reviewed ~30 recent studies on the microbiology of the built environment (MoBE)

- Identified 3 general categories based on level of detail in measuring/documenting built environment metadata or data:

1. Microbial diversity in the absence of building characteristics

   e.g., Hewitt et al. **2012** *PLoS ONE* 7(5):e37849

2. Microbial diversity and *basic* building, HVAC, and/or environmental metadata

   e.g., Kembel et al. **2012** *ISME J* 6:1469-1479;
   Adams et al. **2014** *PLoS ONE* 9(3):e91283

3. Microbial diversity and *detailed* characterizations of built environment data and/or human occupancy/activities

   e.g., Qian et al. **2012** *Indoor Air* 22:339-351

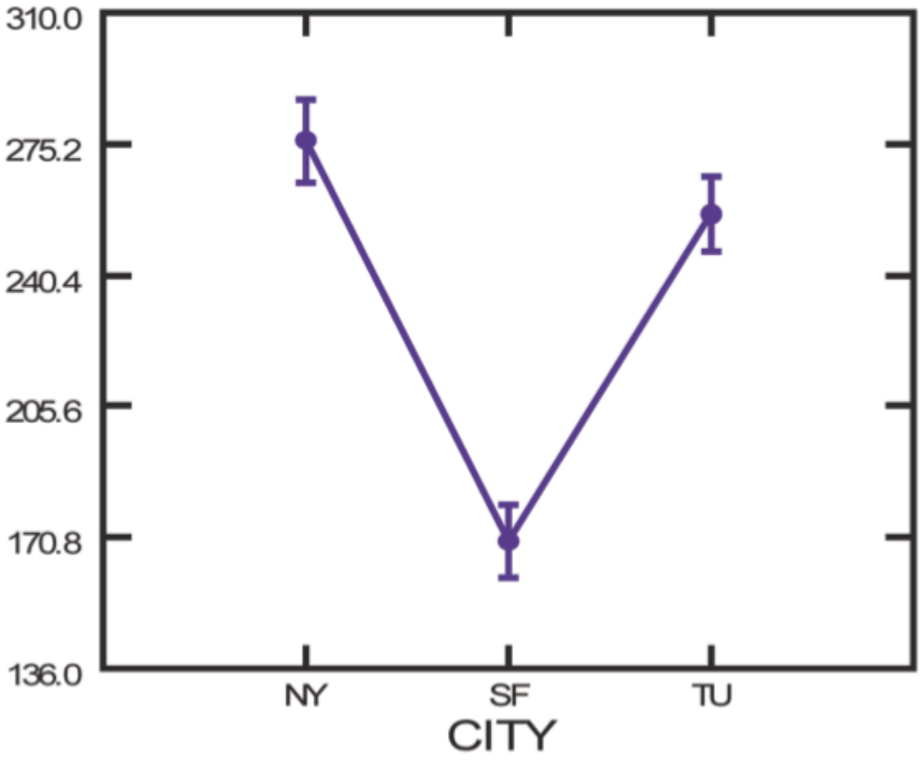
Ramos and Stephens **2014** *Building and Environment*

1. Microbial diversity in the absence of built environment data

"Bacterial community diversity of the Tucson samples was clearly distinguishable from that of New York and San Francisco, which were indistinguishable"

Interesting, but ***why***?

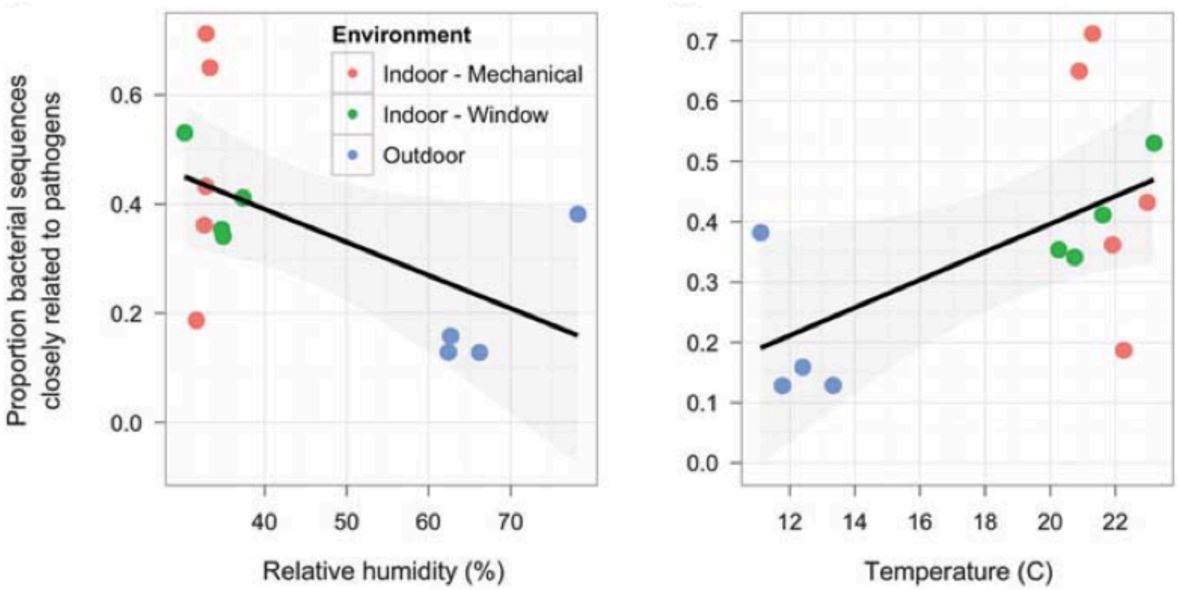
Bacterial abundance in offices in 3 US cities

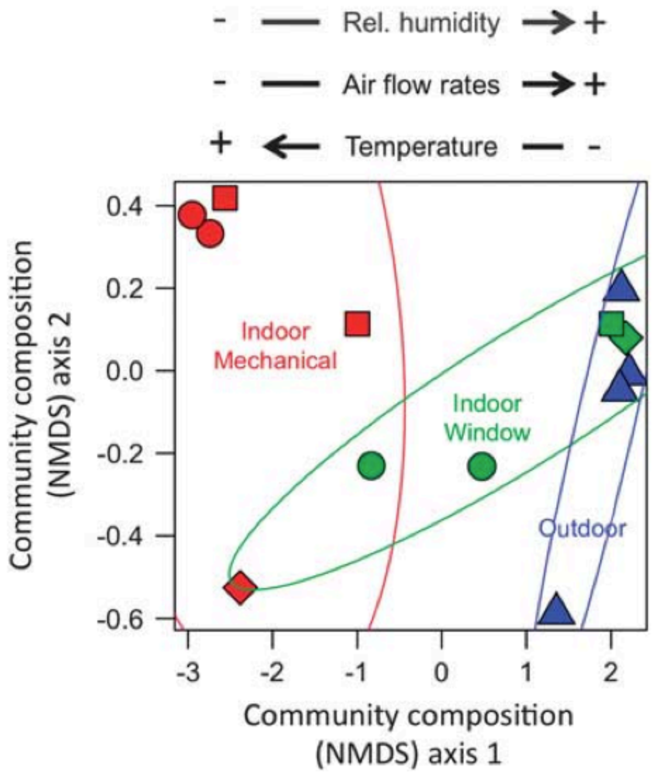


Hewitt et al. **2012** *PLoS ONE* 7(5):e37849

2. Microbial diversity and *basic* built environment data

Airborne bacteria in a healthcare facility





Kembel et al. **2012** *ISME J* 6:1469-1479

Interesting, but built environment factors are correlated

“Bacterial richness tended to be higher in those four (of 11) units that reported at least occasional humidifier use”

Adams et al. **2014** *PLoS ONE* 9(3):e91283

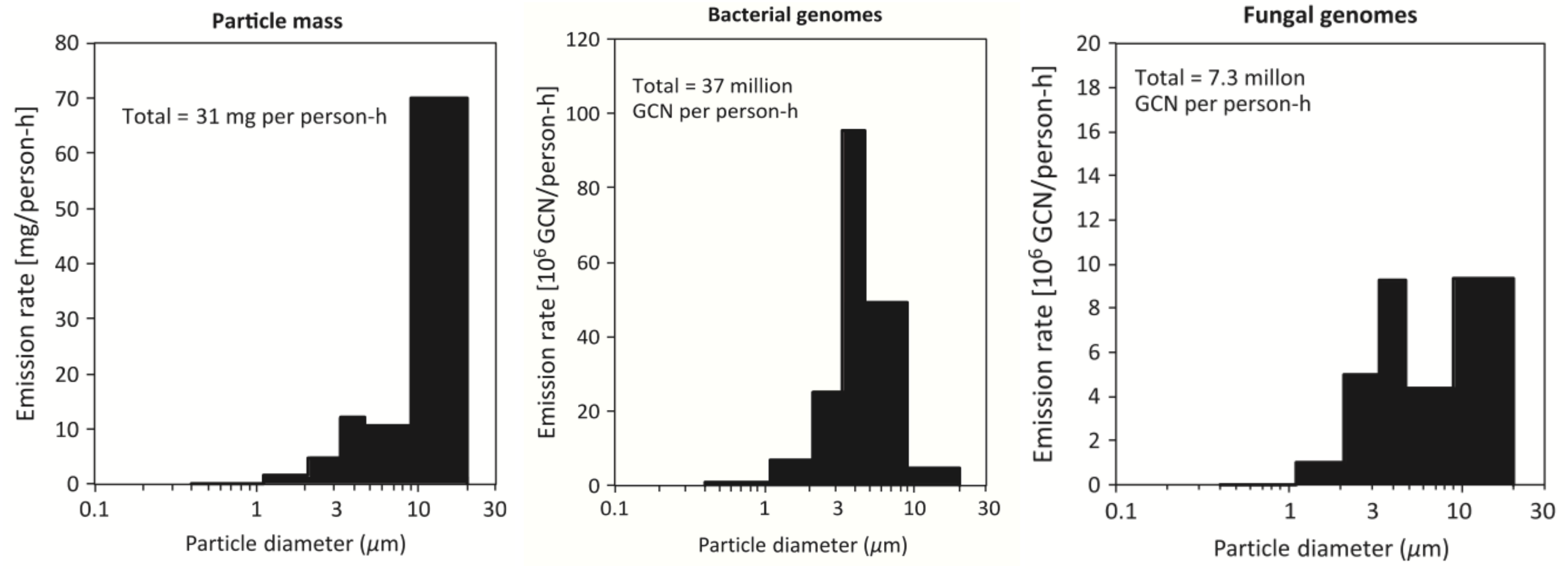
Occupant activities

3. Microbial diversity and *detailed* building data

Size-resolved emission rates of airborne bacteria and fungi in an occupied classroom

J. Qian[1,2], D. Hospodsky[1],
N. Yamamoto[1,3], W. W. Nazaroff[4],
J. Peccia[1]

- Detailed characterization of building operation and occupancy allowed for estimating per-occupant emission rates using a mass-balance model… the **power of building characterization!**



Qian et al. **2012** *Indoor Air* 22:339-351

"Tools to improve built environment data collection for indoor microbial ecology investigations"

Suggestions for built environment measurements

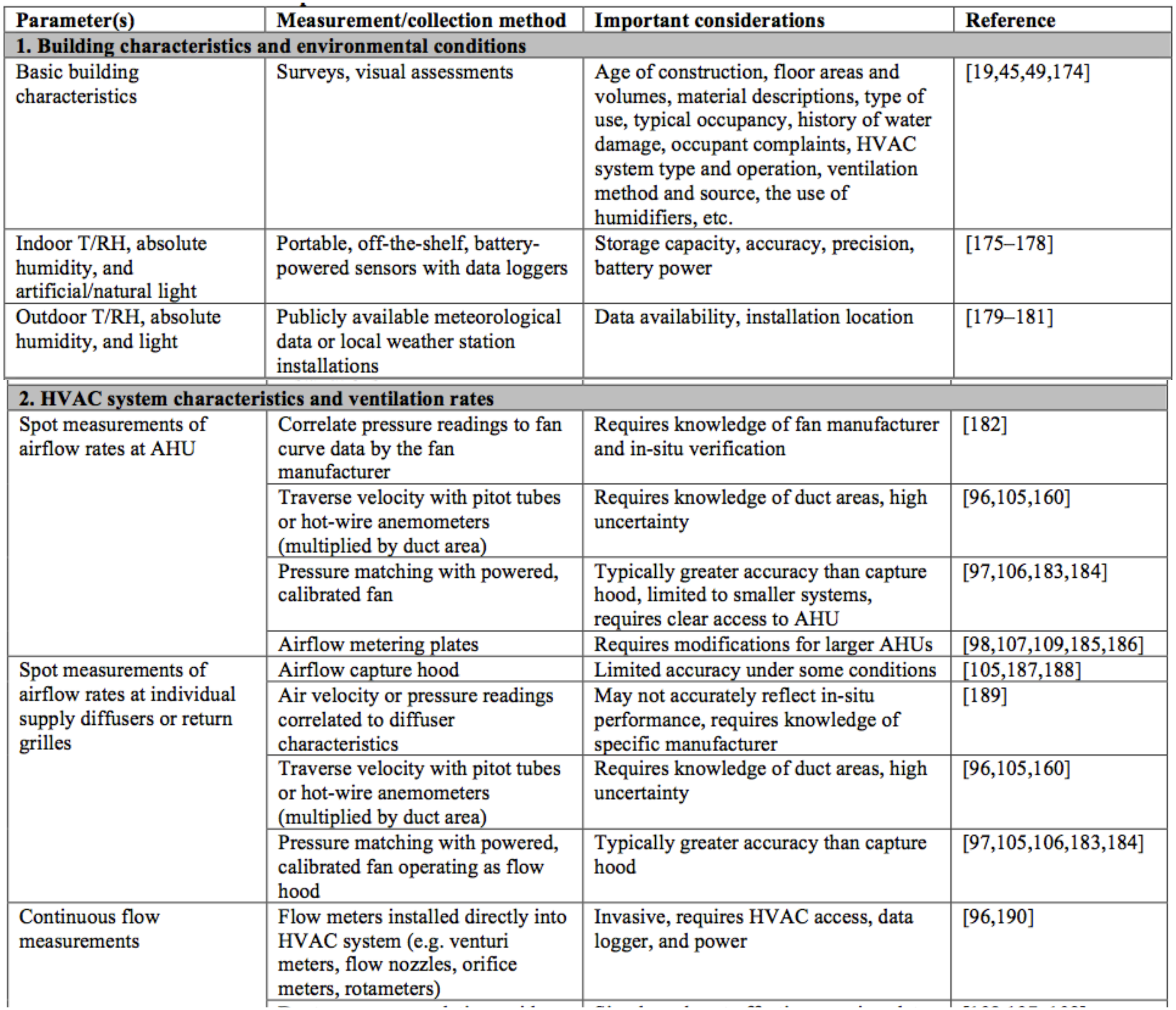
1. Measure detailed building characteristics and long-term indoor environmental conditions
   - Building surveys, T, RH, W, light, others
2. Measure human occupancy and activity
   - Proximity, trip wires, CO_2, RFID, acoustic, Bluetooth, video
3. Characterize HVAC systems and measure ventilation rates (and sources of air)
4. Characterize surfaces
   - T, a_w, pH, porosity, qualitative details, frequency of cleaning
5. Standardize air sampling and quantifying aerosol dynamics

A few references on environmental conditions and microbial communities:
Tang **2009** *J R Soc Interface;* Noyce et al **2006** *J Hosp Infect*; Mbithi et al **1991** *Appl Environ Microbiol;*
Baughman, Arens **1996** *ASHRAE Trans*; Jawad et al **1996** *J Clin Microbiol*; McEldowney, Fletcher **1988** *Lett Appl Microbiol*;
Coughenour et al **2011** *Microb Drug Resist*; Hobday, Dancer **2013** *J Hosp Infect*

Ramos and Stephens **2014** *Building and Environment* doi:10.1016/j.buildenv.2014.07.004

“Tools to improve built environment data collection for indoor microbial ecology investigations”

Long list of measurement types and techniques

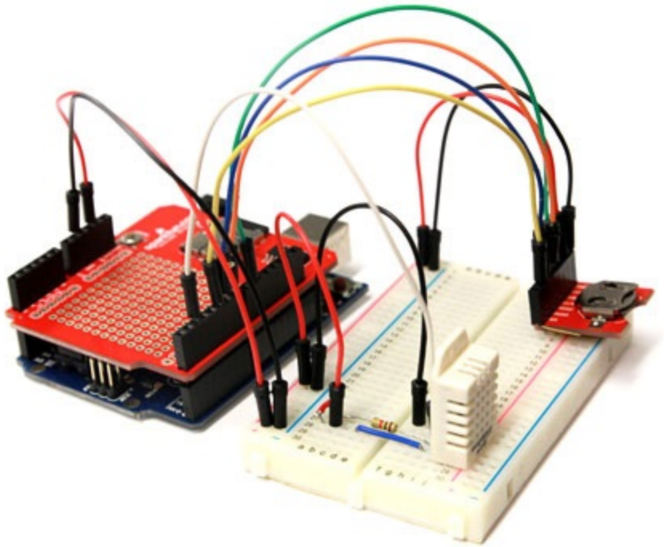


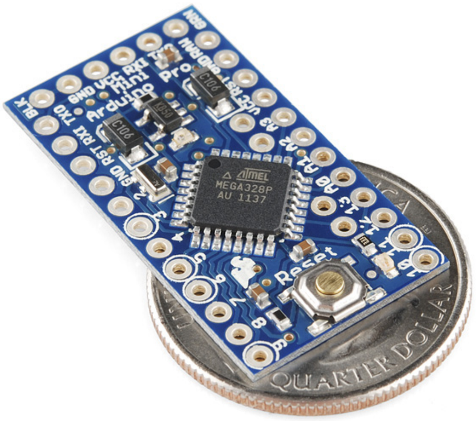
Parameter(s)	Measurement/collection method	Important considerations	Reference
1. Building characteristics and environmental conditions			
Basic building characteristics	Surveys, visual assessments	Age of construction, floor areas and volumes, material descriptions, type of use, typical occupancy, history of water damage, occupant complaints, HVAC system type and operation, ventilation method and source, the use of humidifiers, etc.	[19,45,49,174]
Indoor T/RH, absolute humidity, and artificial/natural light	Portable, off-the-shelf, battery-powered sensors with data loggers	Storage capacity, accuracy, precision, battery power	[175–178]
Outdoor T/RH, absolute humidity, and light	Publicly available meteorological data or local weather station installations	Data availability, installation location	[179–181]
2. HVAC system characteristics and ventilation rates			
Spot measurements of airflow rates at AHU	Correlate pressure readings to fan curve data by the fan manufacturer	Requires knowledge of fan manufacturer and in-situ verification	[182]
	Traverse velocity with pitot tubes or hot-wire anemometers (multiplied by duct area)	Requires knowledge of duct areas, high uncertainty	[96,105,160]
	Pressure matching with powered, calibrated fan	Typically greater accuracy than capture hood, limited to smaller systems, requires clear access to AHU	[97,106,183,184]
	Airflow metering plates	Requires modifications for larger AHUs	[98,107,109,185,186]
Spot measurements of airflow rates at individual supply diffusers or return grilles	Airflow capture hood	Limited accuracy under some conditions	[105,187,188]
	Air velocity or pressure readings correlated to diffuser characteristics	May not accurately reflect in-situ performance, requires knowledge of specific manufacturer	[189]
	Traverse velocity with pitot tubes or hot-wire anemometers (multiplied by duct area)	Requires knowledge of duct areas, high uncertainty	[96,105,160]
	Pressure matching with powered, calibrated fan operating as flow hood	Typically greater accuracy than capture hood	[97,105,106,183,184]
Continuous flow measurements	Flow meters installed directly into HVAC system (e.g. venturi meters, flow nozzles, orifice meters, rotameters)	Invasive, requires HVAC access, data logger, and power	[96,190]

Ramos and Stephens **2014** *Building and Environment*

OPEN SOURCE BUILDING SCIENCE SENSORS

*The **Open Source Building Science Sensors (OSBSS)** project is designing and demonstrating how to build inexpensive building environmental and operational sensors for long-term studies of the indoor environment using open source hardware and software*





ARDUINO

http://www.osbss.com

OPEN SOURCE BUILDING SCIENCE SENSORS

*The **Open Source Building Science Sensors (OSBSS)** project is designing and demonstrating how to build inexpensive building environmental and operational sensors for long-term studies of the indoor environment using open source hardware and software*

T/RH

Surface T

Eq. RH (a_w)

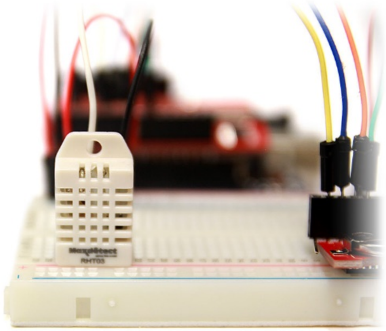
Data logger ⇆ Diff. pressure

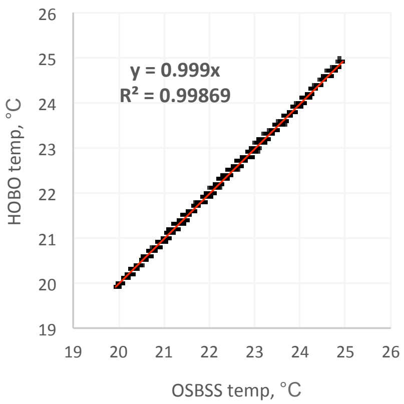
On/off

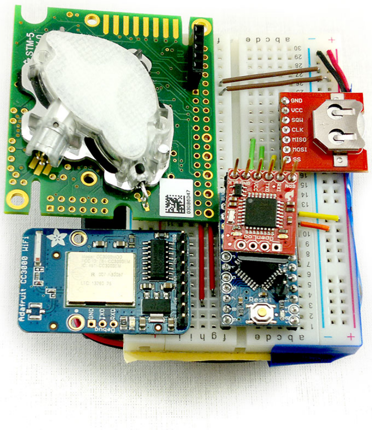
Proximity IR

Dual IR beam break

CO_2





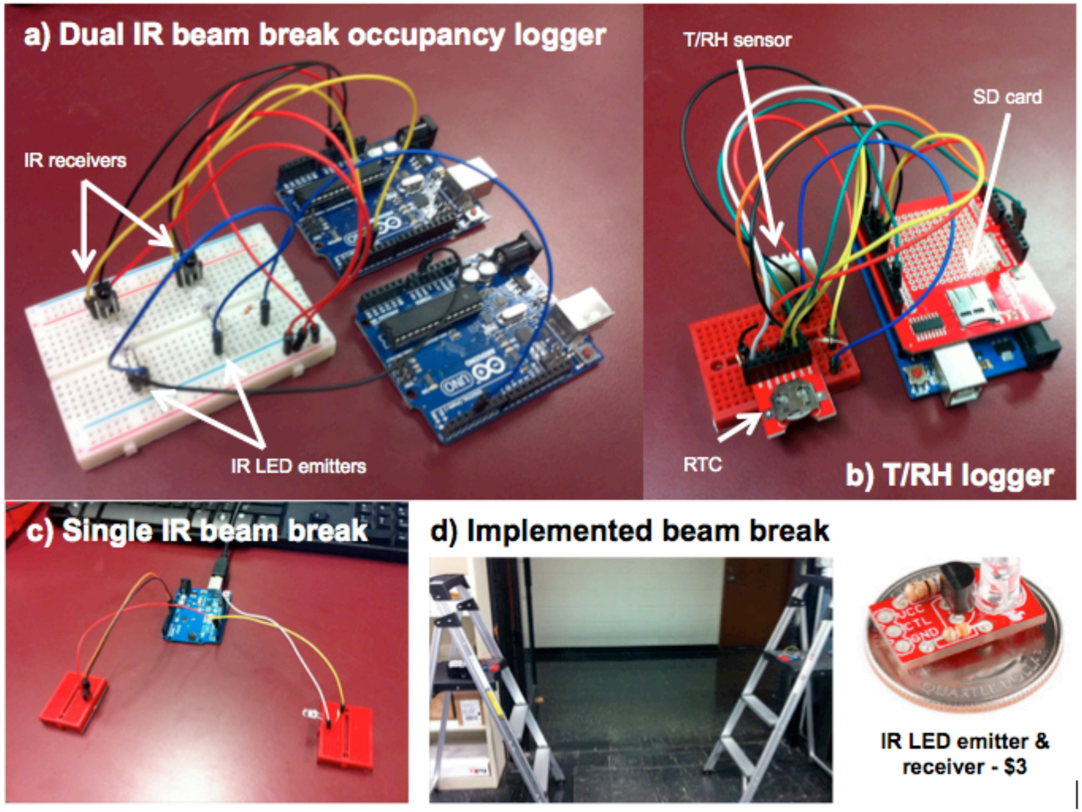


http://www.osbss.com

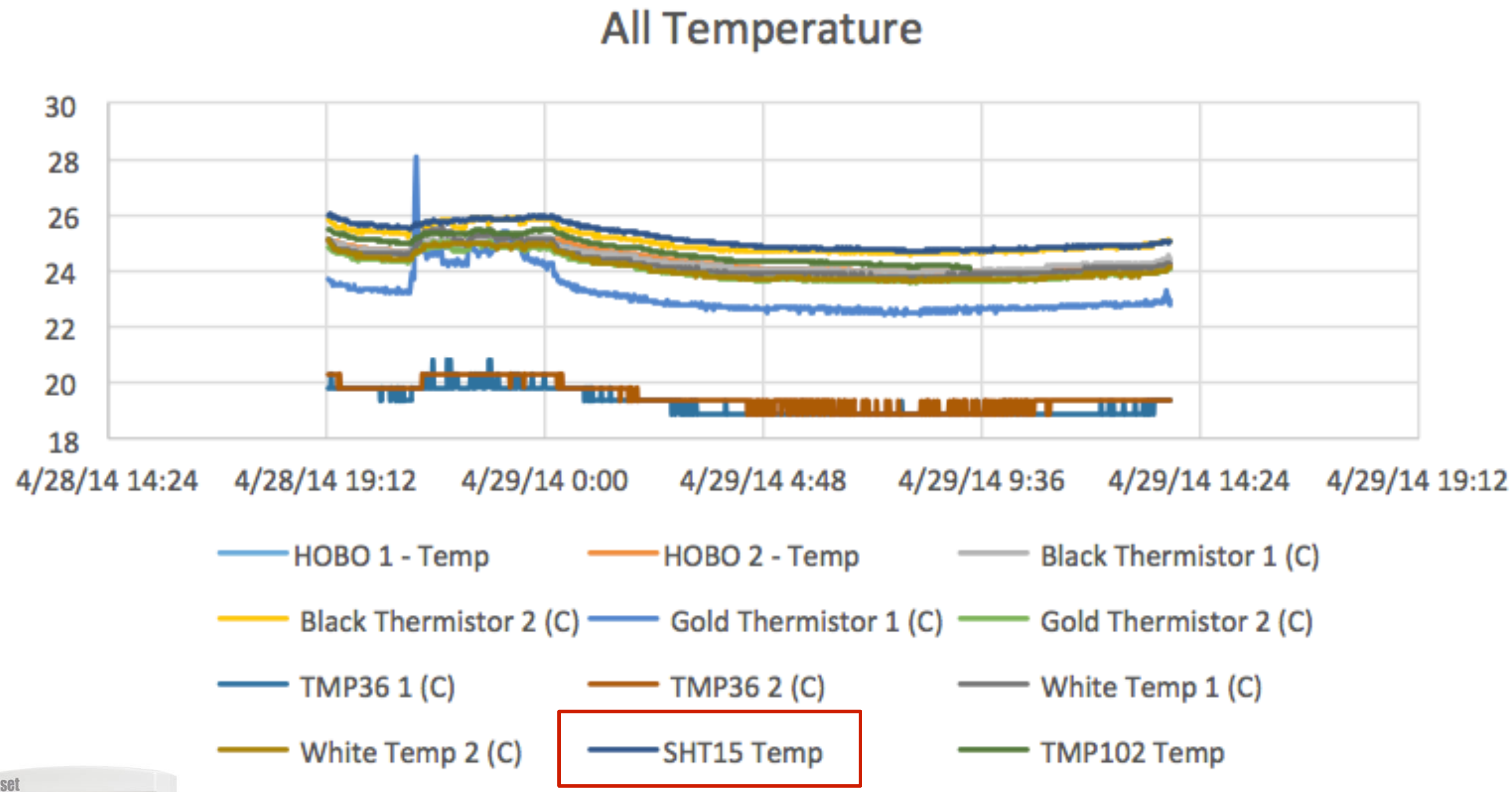
Development process: Stage 1 (Concept)

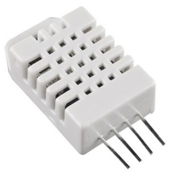
- Begin with breadboard (solder-less) concept development on Arduino Uno controllers with off-the-shelf sensors
  - Allows for testing basic functionality, accuracy, and developing code

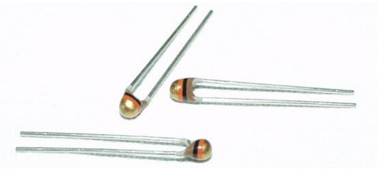
- Issues at this stage:
  - High power draw
  - Real time clock (RTC)
  - Data storage
  - Durability
  - Aesthetics


a) Dual IR beam break occupancy logger
IR receivers
IR LED emitters
T/RH sensor
SD card
RTC
b) T/RH logger
c) Single IR beam break
d) Implemented beam break
IR LED emitter & receiver - $3

Development process: Stage 1 (Concept)

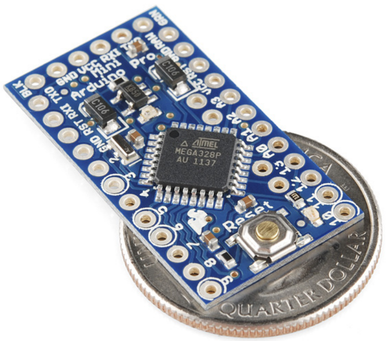






Development process: Stage 2 (Prototype)

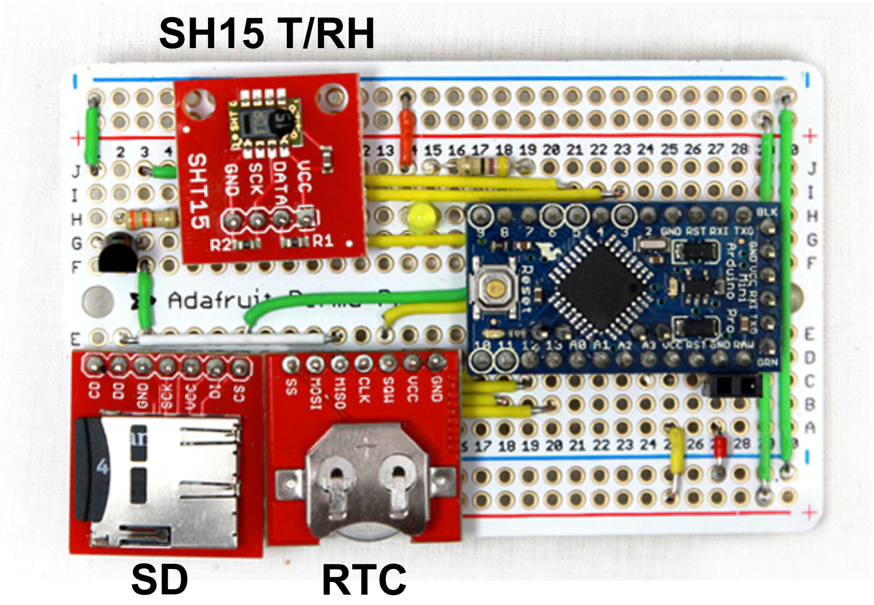
- Move from Arduino Uno to Arduino Mini Pro (or knock-off versions for $3)
  - Large reductions in power draw with custom libraries (use of sleep mode functions)



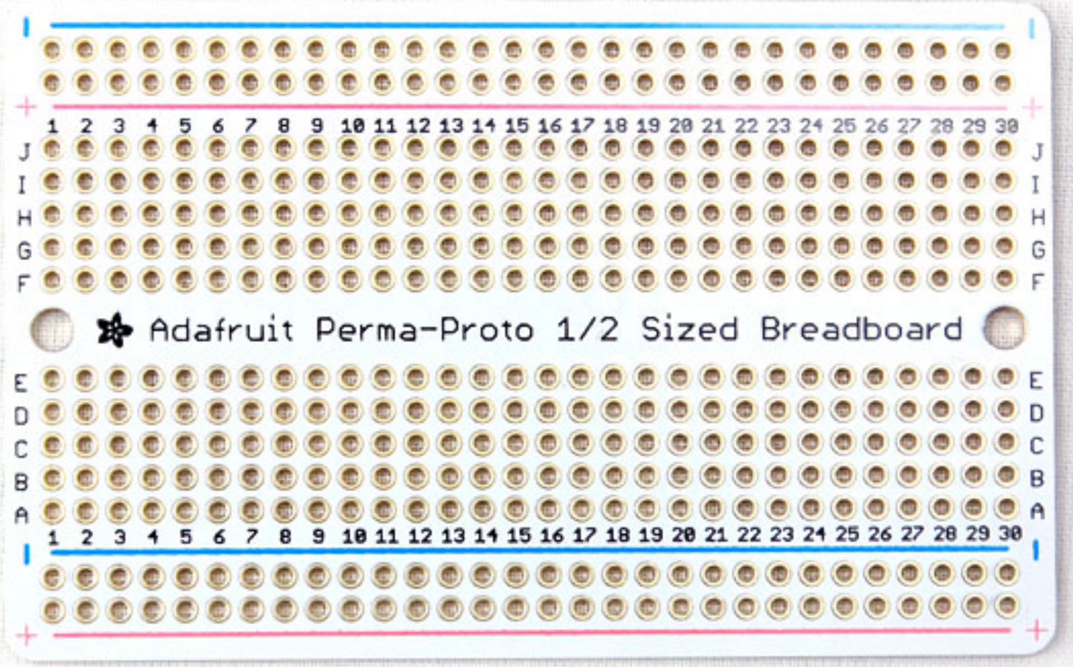
From ~20 mA resting (Uno)
To ~20 μA resting (Mini + code)
From ~4 days to ~400 days on AA

- Upgrade to solderable breadboard
  - Improves durability

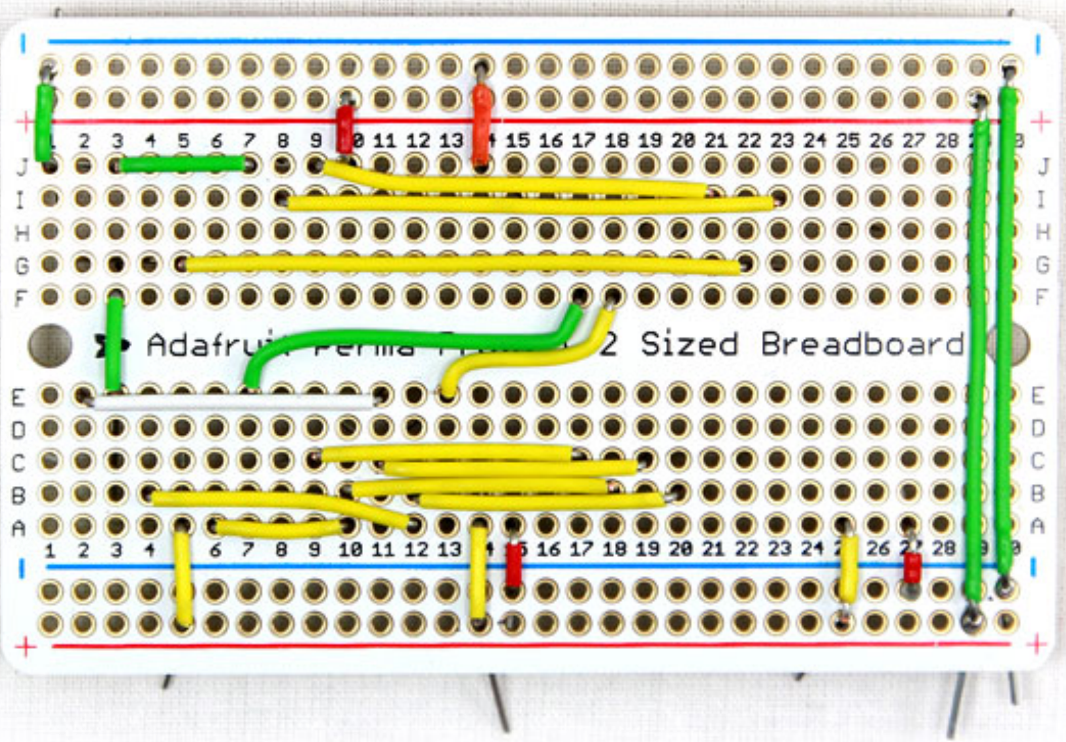
- Upgrade to custom enclosures
  - Improves aesthetics



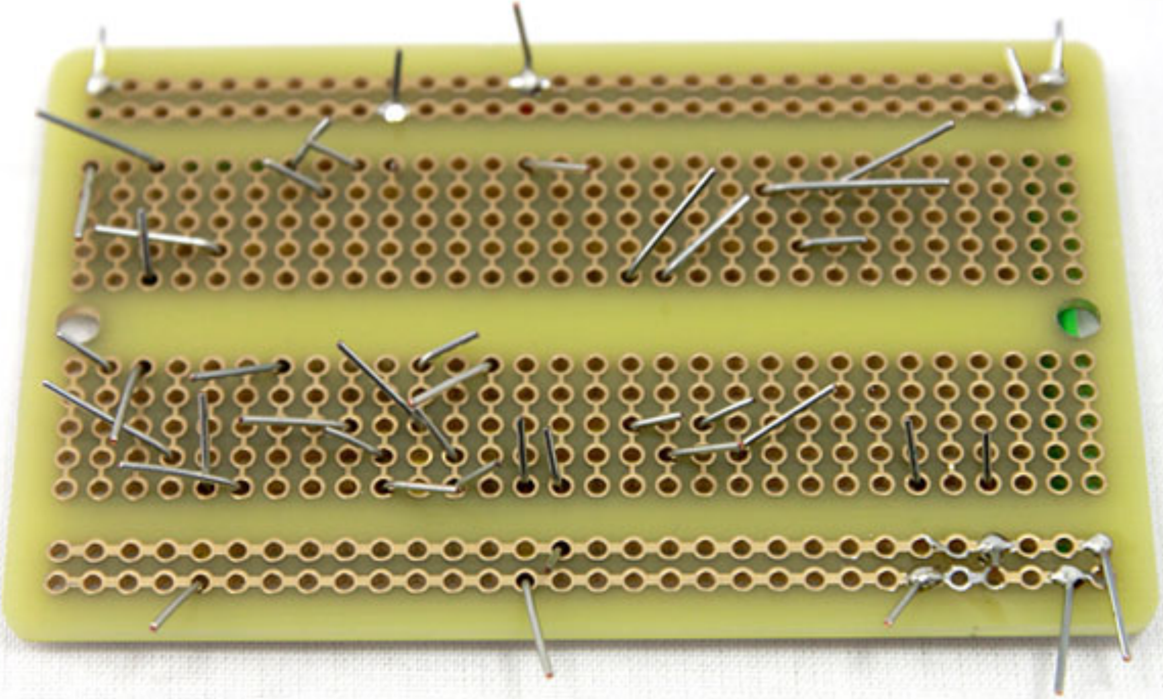
Development process: Stage 3 (Tutorials)



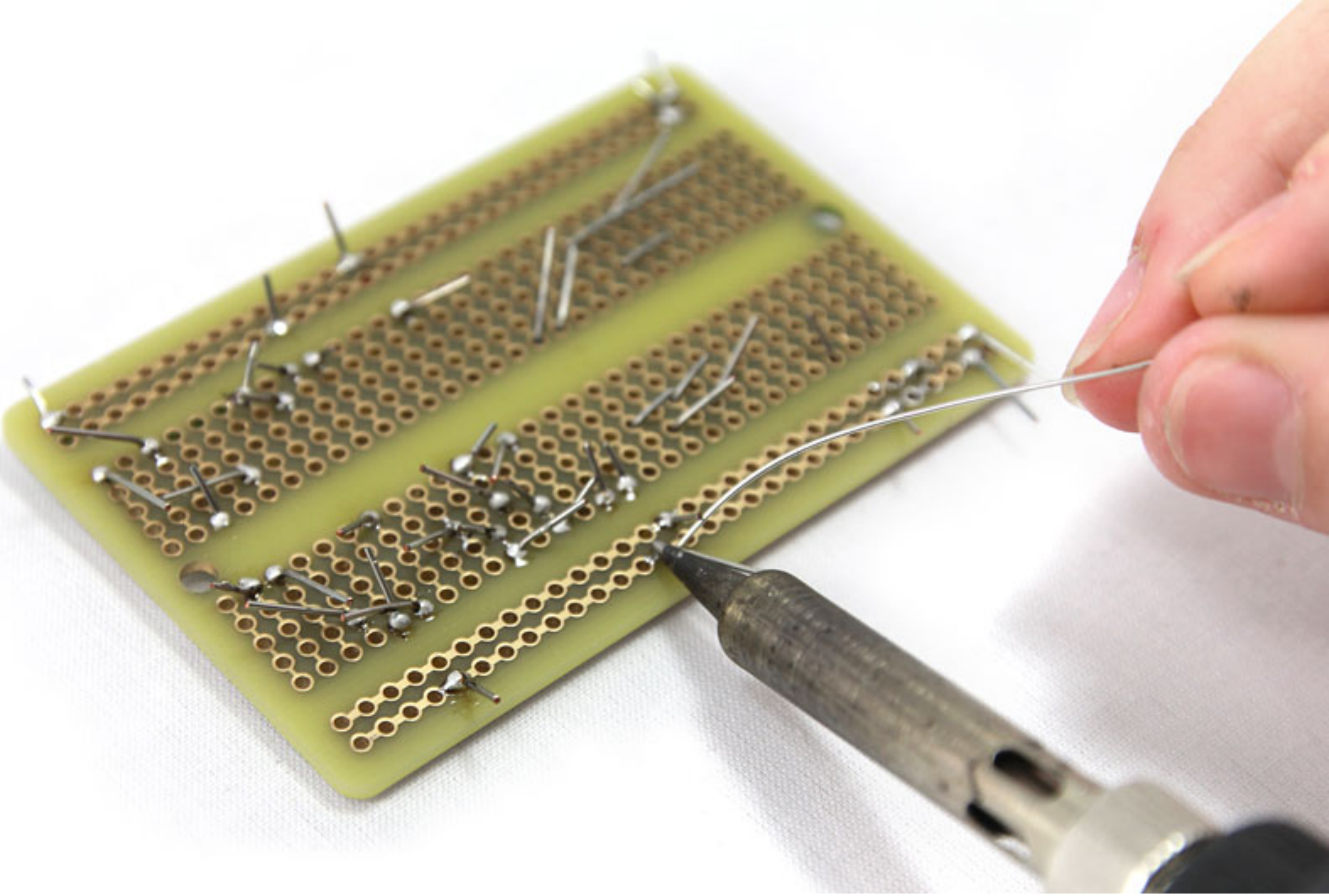
Development process: Stage 3 (Tutorials)



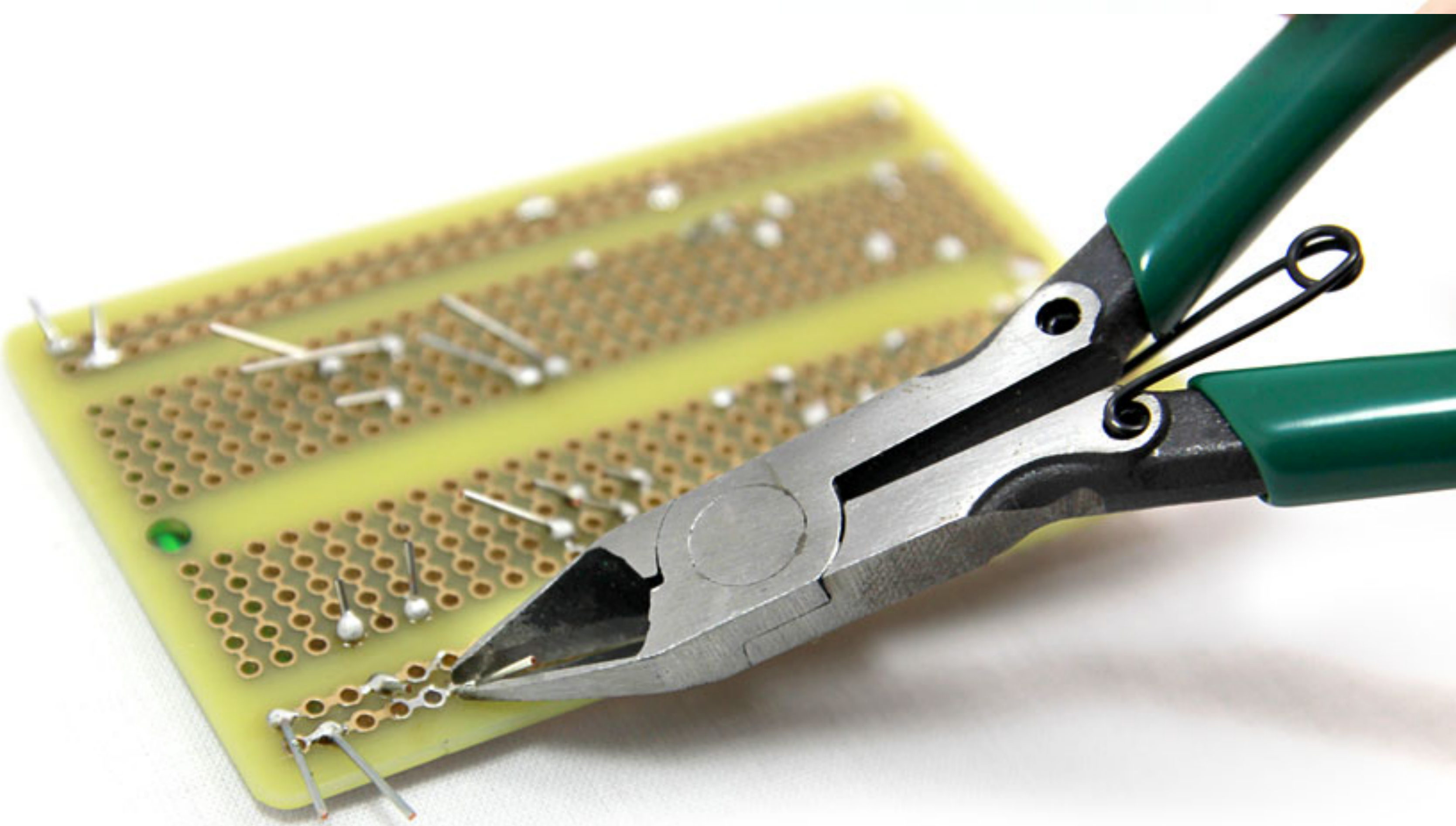
Development process: Stage 3 (Tutorials)



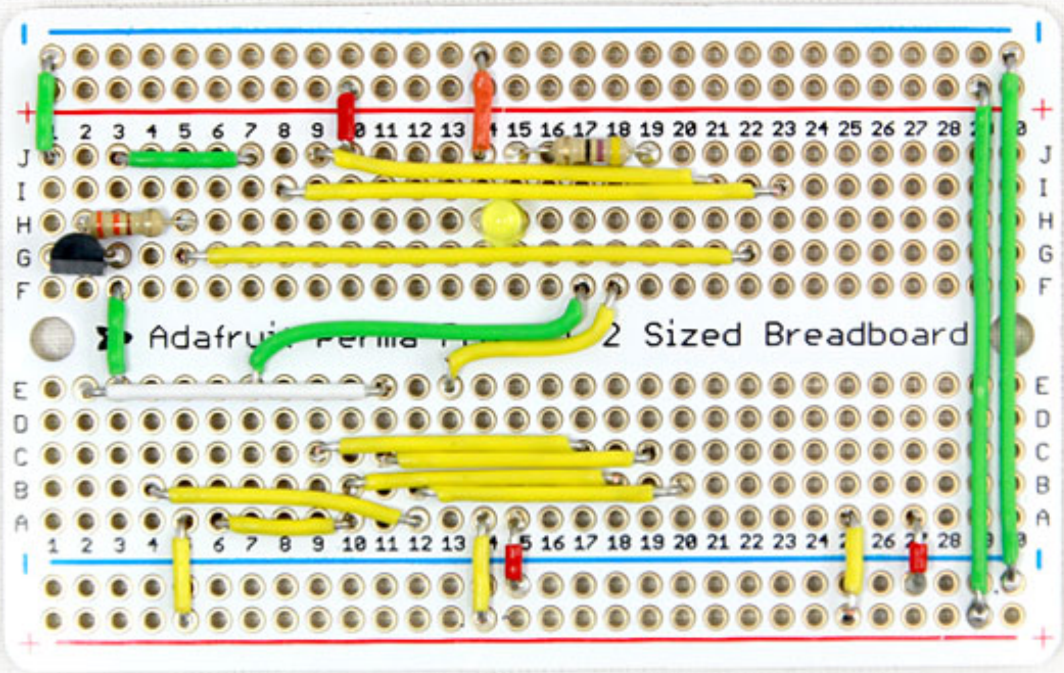
Development process: Stage 3 (Tutorials)



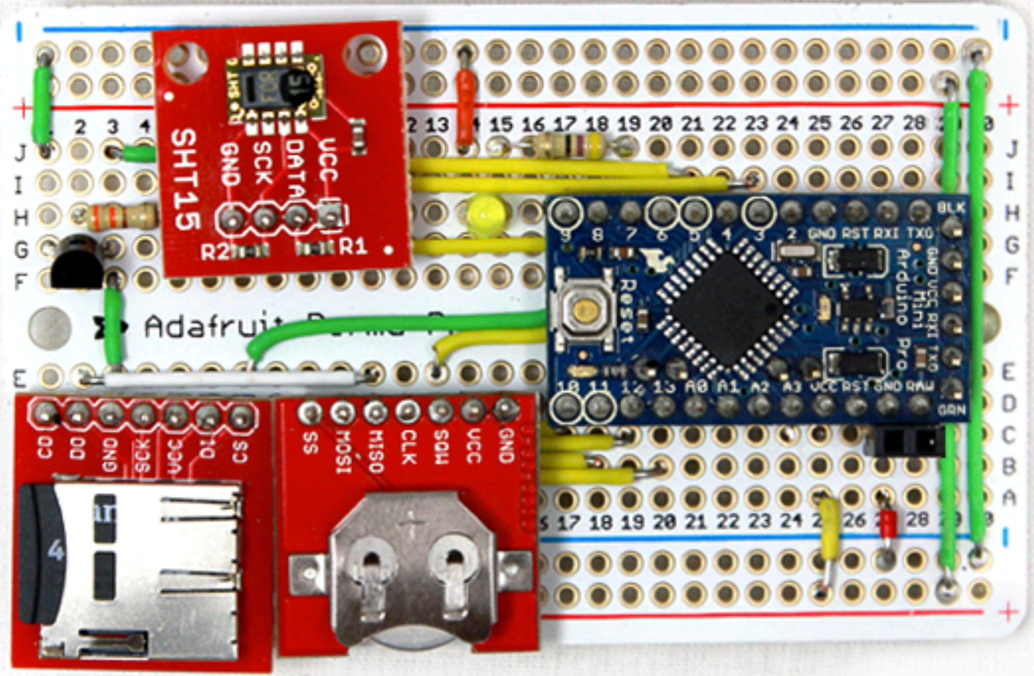
Development process: Stage 3 (Tutorials)



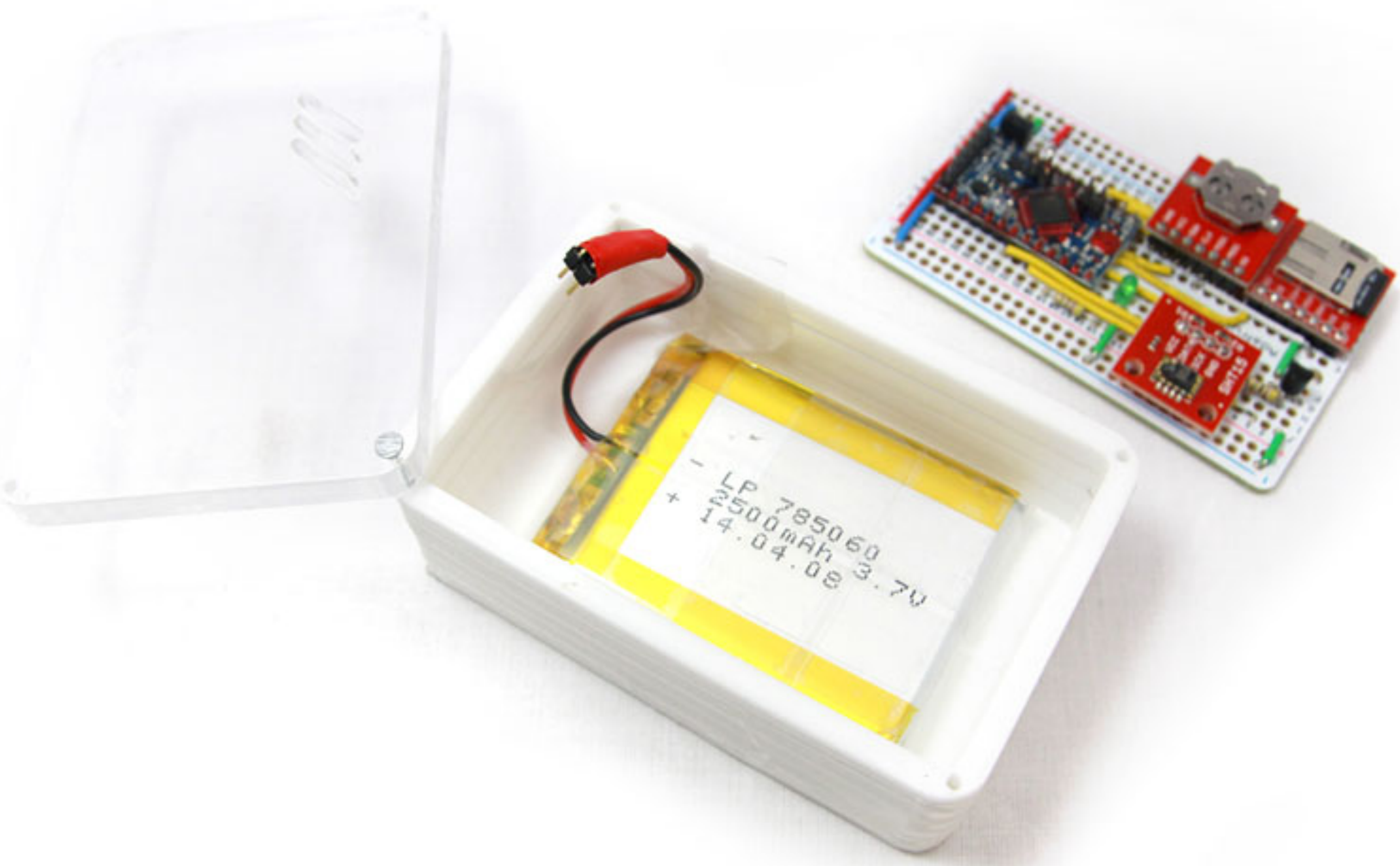
Development process: Stage 3 (Tutorials)



Development process: Stage 3 (Tutorials)

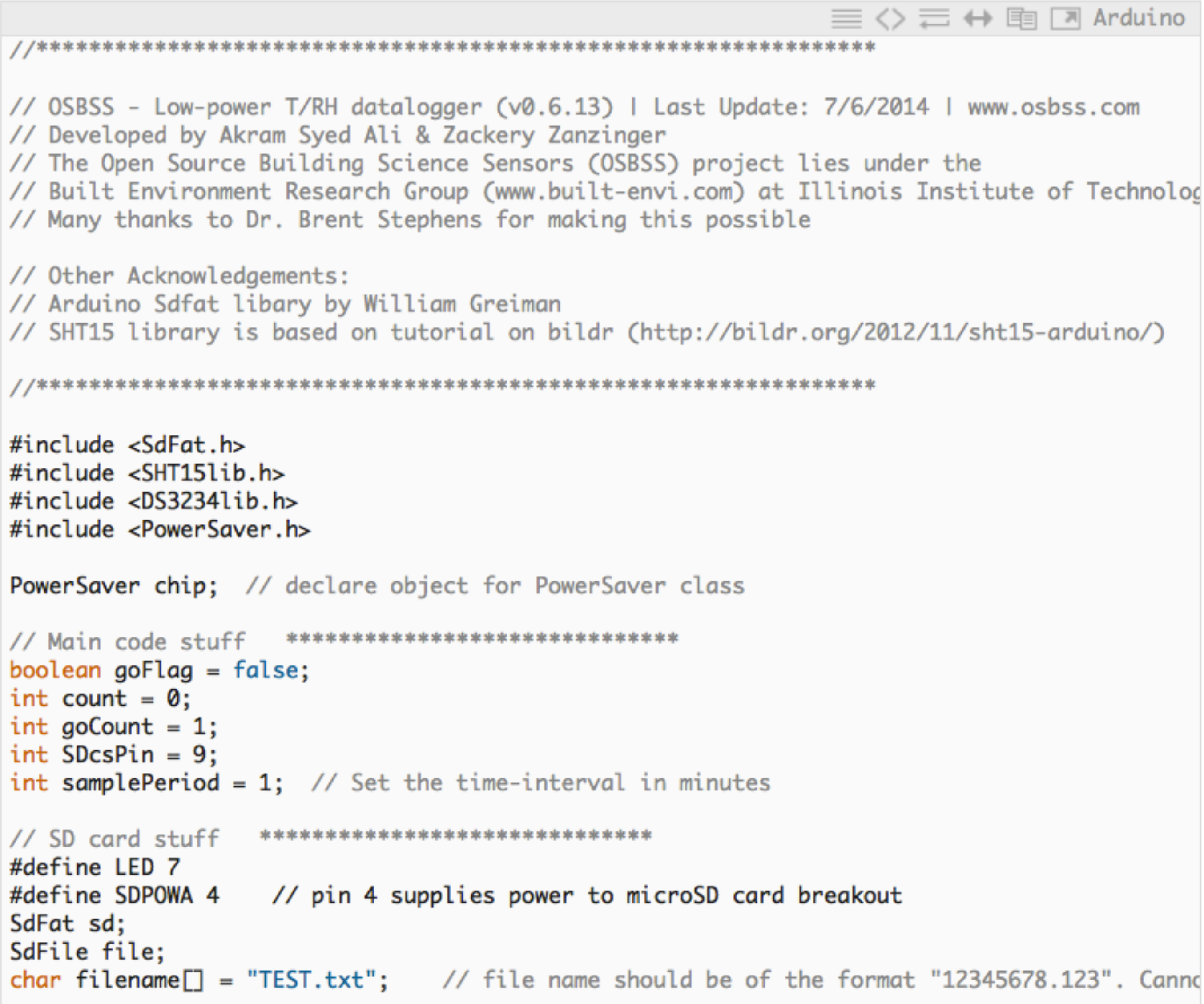


Development process: Stage 3 (Tutorials)



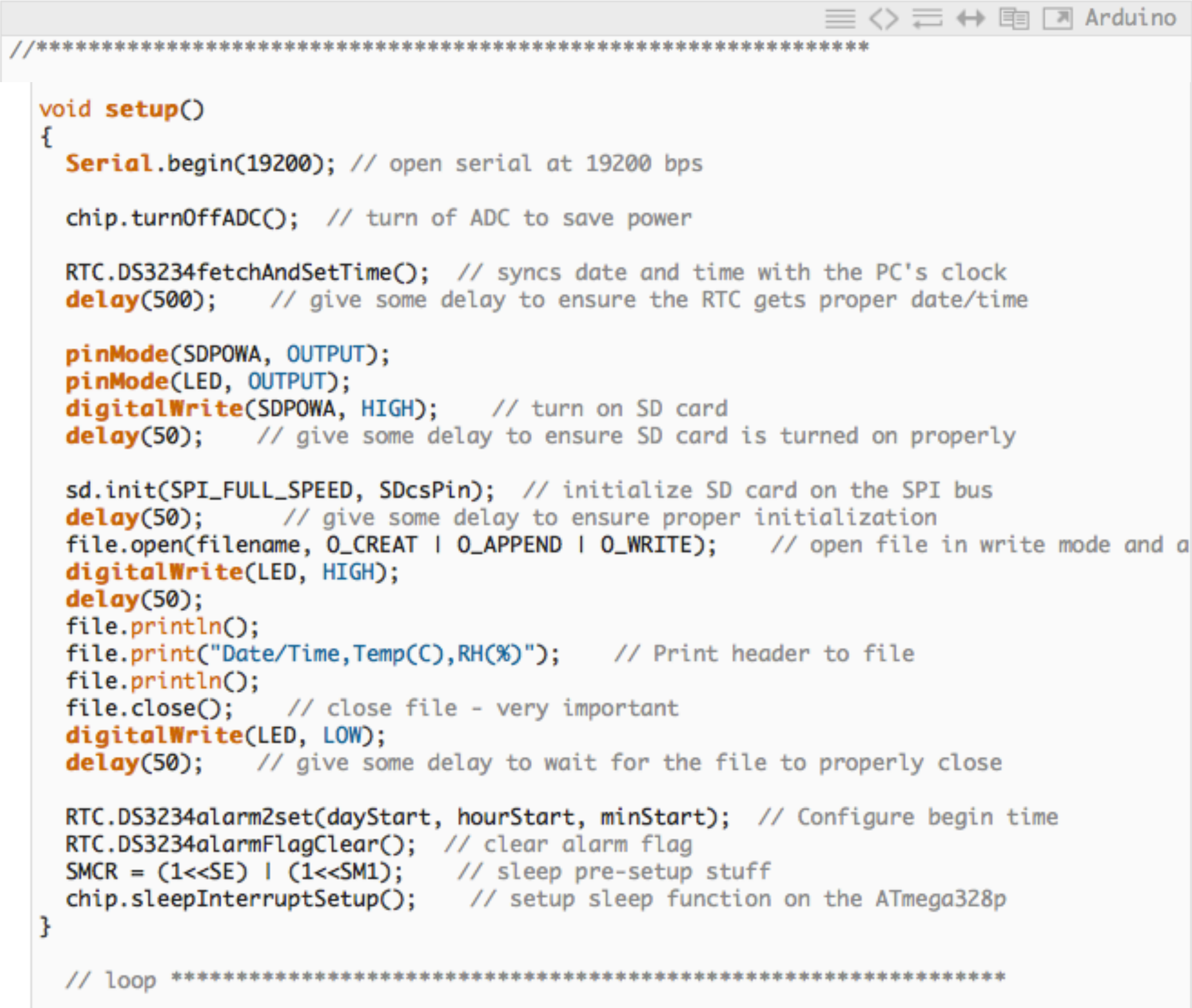
Development process: Stage 3 (Tutorials)

Development process: Stage 3 (Tutorials)



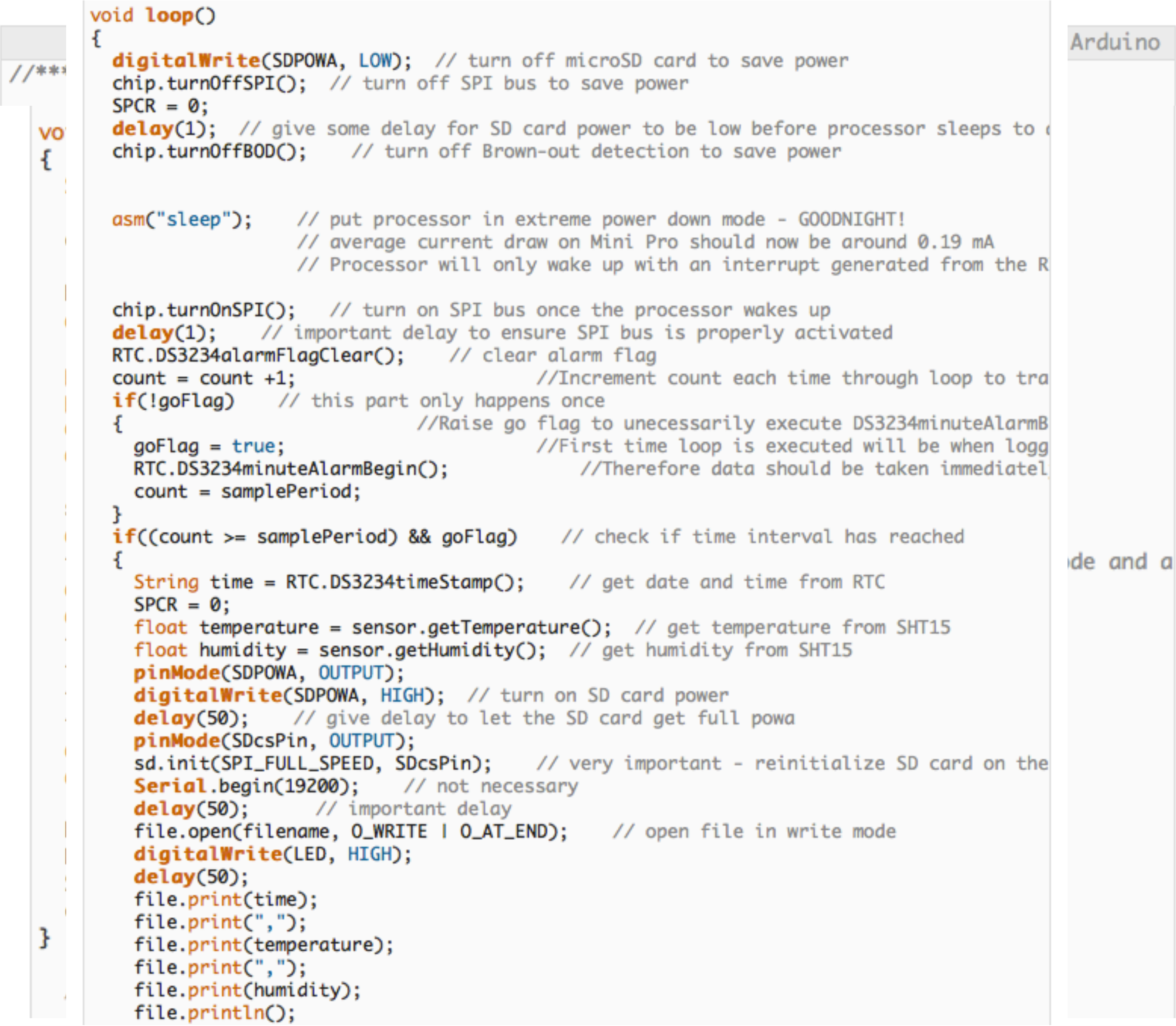
```
//*****************************************************************

// OSBSS - Low-power T/RH datalogger (v0.6.13) | Last Update: 7/6/2014 | www.osbss.com
// Developed by Akram Syed Ali & Zackery Zanzinger
// The Open Source Building Science Sensors (OSBSS) project lies under the
// Built Environment Research Group (www.built-envi.com) at Illinois Institute of Technolog
// Many thanks to Dr. Brent Stephens for making this possible

// Other Acknowledgements:
// Arduino Sdfat libary by William Greiman
// SHT15 library is based on tutorial on bildr (http://bildr.org/2012/11/sht15-arduino/)

//*****************************************************************

#include <SdFat.h>
#include <SHT15lib.h>
#include <DS3234lib.h>
#include <PowerSaver.h>

PowerSaver chip;  // declare object for PowerSaver class

// Main code stuff   ******************************
boolean goFlag = false;
int count = 0;
int goCount = 1;
int SDcsPin = 9;
int samplePeriod = 1;  // Set the time-interval in minutes

// SD card stuff   ******************************
#define LED 7
#define SDPOWA 4    // pin 4 supplies power to microSD card breakout
SdFat sd;
SdFile file;
char filename[] = "TEST.txt";    // file name should be of the format "12345678.123". Cannc
```

Development process: Stage 3 (Tutorials)



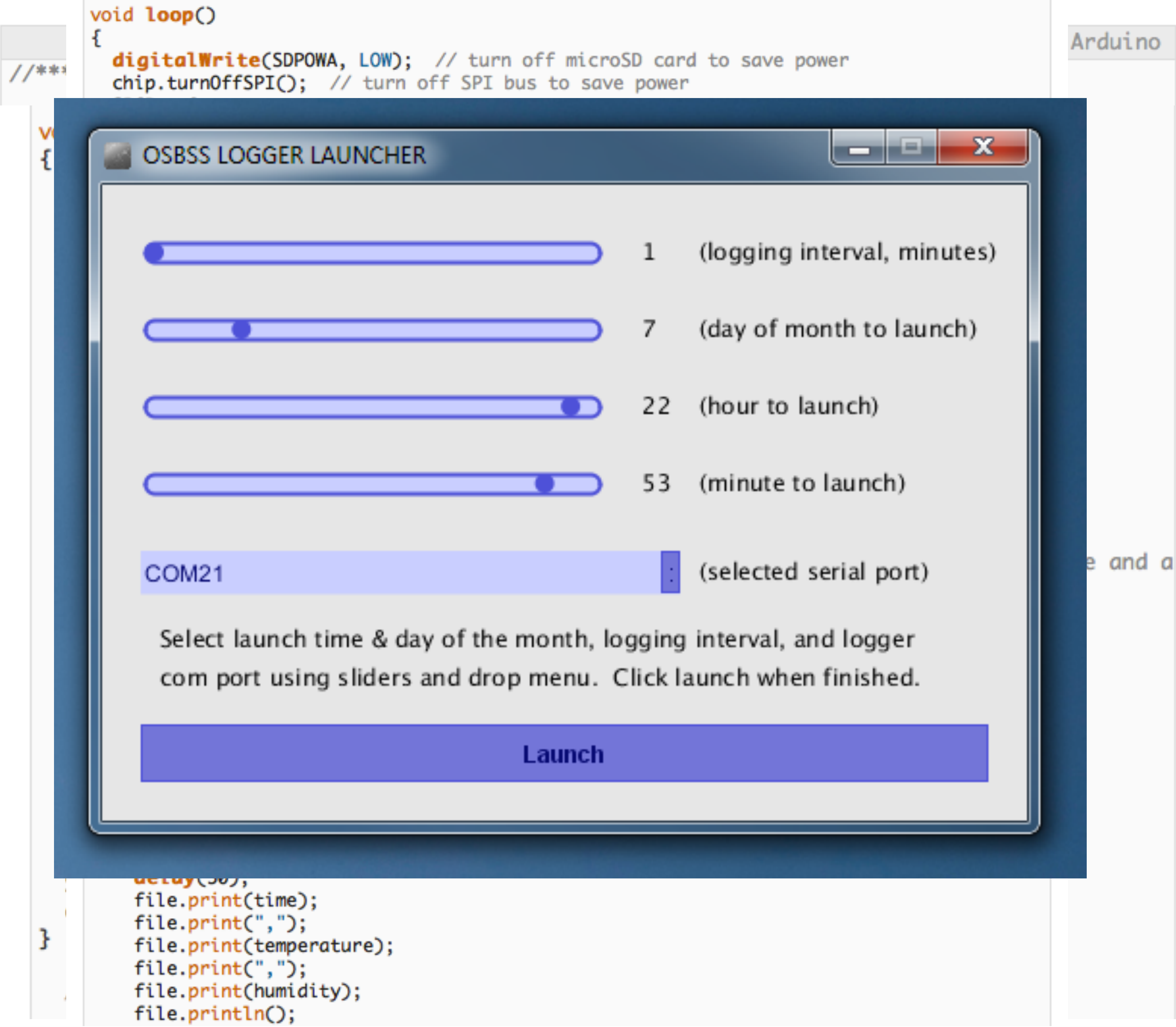
```
//**************************************************************

void setup()
{
  Serial.begin(19200); // open serial at 19200 bps

  chip.turnOffADC();  // turn of ADC to save power

  RTC.DS3234fetchAndSetTime();  // syncs date and time with the PC's clock
  delay(500);    // give some delay to ensure the RTC gets proper date/time

  pinMode(SDPOWA, OUTPUT);
  pinMode(LED, OUTPUT);
  digitalWrite(SDPOWA, HIGH);    // turn on SD card
  delay(50);    // give some delay to ensure SD card is turned on properly

  sd.init(SPI_FULL_SPEED, SDcsPin);  // initialize SD card on the SPI bus
  delay(50);     // give some delay to ensure proper initialization
  file.open(filename, O_CREAT | O_APPEND | O_WRITE);    // open file in write mode and a
  digitalWrite(LED, HIGH);
  delay(50);
  file.println();
  file.print("Date/Time,Temp(C),RH(%)");    // Print header to file
  file.println();
  file.close();    // close file - very important
  digitalWrite(LED, LOW);
  delay(50);    // give some delay to wait for the file to properly close

  RTC.DS3234alarm2set(dayStart, hourStart, minStart);  // Configure begin time
  RTC.DS3234alarmFlagClear();  // clear alarm flag
  SMCR = (1<<SE) | (1<<SM1);    // sleep pre-setup stuff
  chip.sleepInterruptSetup();    // setup sleep function on the ATmega328p
}

  // loop **************************************************************
```

Development process: Stage 3 (Tutorials)



```
void loop()
{
  digitalWrite(SDPOWA, LOW);  // turn off microSD card to save power
  chip.turnOffSPI();  // turn off SPI bus to save power
  SPCR = 0;
  delay(1);  // give some delay for SD card power to be low before processor sleeps to 
  chip.turnOffBOD();    // turn off Brown-out detection to save power


  asm("sleep");     // put processor in extreme power down mode - GOODNIGHT!
                    // average current draw on Mini Pro should now be around 0.19 mA
                    // Processor will only wake up with an interrupt generated from the R

  chip.turnOnSPI();   // turn on SPI bus once the processor wakes up
  delay(1);    // important delay to ensure SPI bus is properly activated
  RTC.DS3234alarmFlagClear();    // clear alarm flag
  count = count +1;                       //Increment count each time through loop to tra
  if(!goFlag)    // this part only happens once
  {                          //Raise go flag to unecessarily execute DS3234minuteAlarmB
    goFlag = true;                        //First time loop is executed will be when logg
    RTC.DS3234minuteAlarmBegin();            //Therefore data should be taken immediatel
    count = samplePeriod;
  }
  if((count >= samplePeriod) && goFlag)    // check if time interval has reached
  {
    String time = RTC.DS3234timeStamp();    // get date and time from RTC
    SPCR = 0;
    float temperature = sensor.getTemperature();  // get temperature from SHT15
    float humidity = sensor.getHumidity();  // get humidity from SHT15
    pinMode(SDPOWA, OUTPUT);
    digitalWrite(SDPOWA, HIGH);  // turn on SD card power
    delay(50);    // give delay to let the SD card get full powa
    pinMode(SDcsPin, OUTPUT);
    sd.init(SPI_FULL_SPEED, SDcsPin);    // very important - reinitialize SD card on the
    Serial.begin(19200);    // not necessary
    delay(50);    // important delay
    file.open(filename, O_WRITE | O_AT_END);    // open file in write mode
    digitalWrite(LED, HIGH);
    delay(50);
    file.print(time);
    file.print(",");
    file.print(temperature);
    file.print(",");
    file.print(humidity);
    file.println();
```

Development process: Stage 3 (Tutorials)



```
void loop()
{
  digitalWrite(SDPOWA, LOW);  // turn off microSD card to save power
  chip.turnOffSPI();  // turn off SPI bus to save power
```

OSBSS LOGGER LAUNCHER

1 (logging interval, minutes)

7 (day of month to launch)

22 (hour to launch)

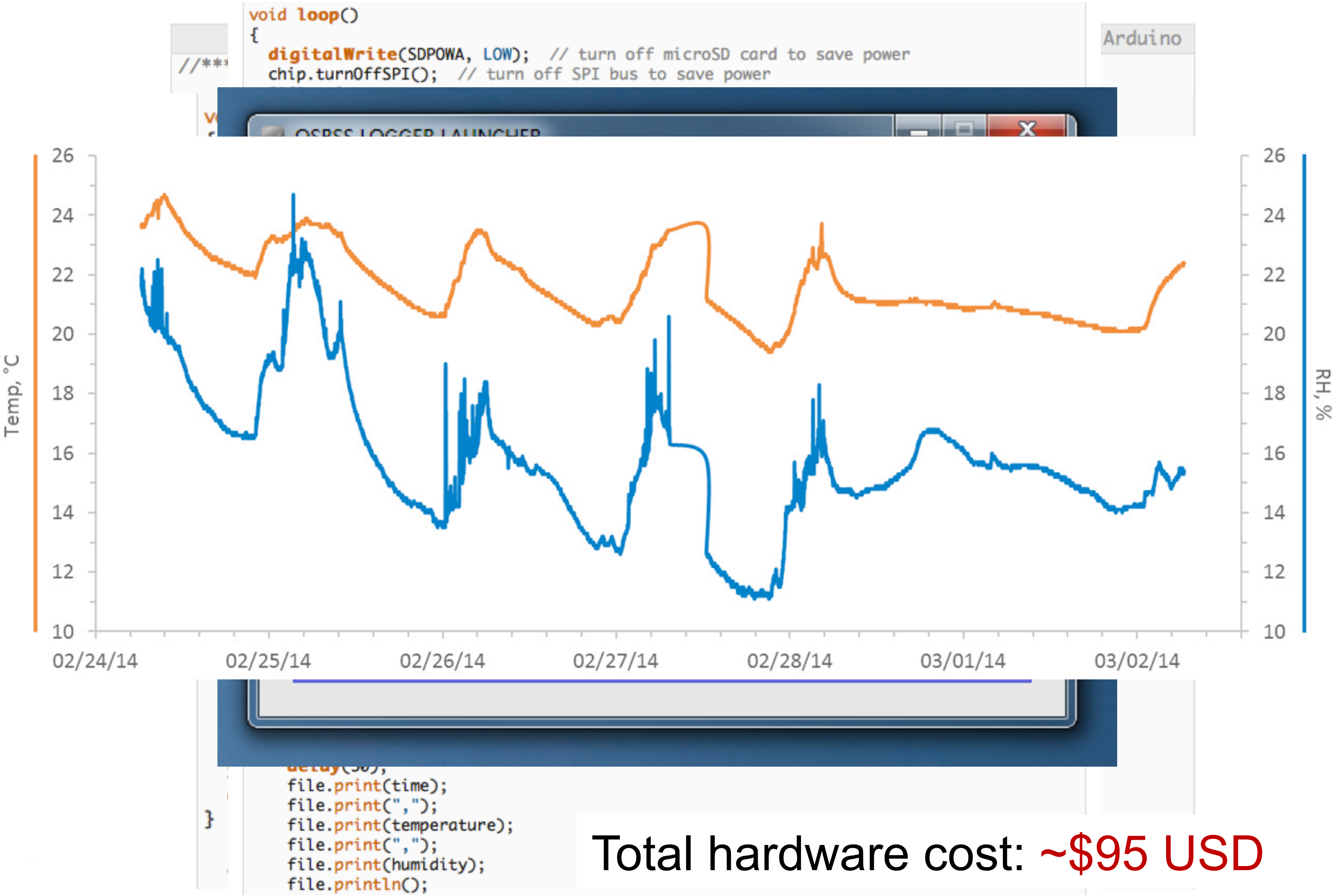
53 (minute to launch)

COM21 (selected serial port)

Select launch time & day of the month, logging interval, and logger com port using sliders and drop menu. Click launch when finished.

```
    delay(30);
    file.print(time);
    file.print(",");
    file.print(temperature);
    file.print(",");
    file.print(humidity);
    file.println();
}
```

Development process: Stage 3 (Tutorials)



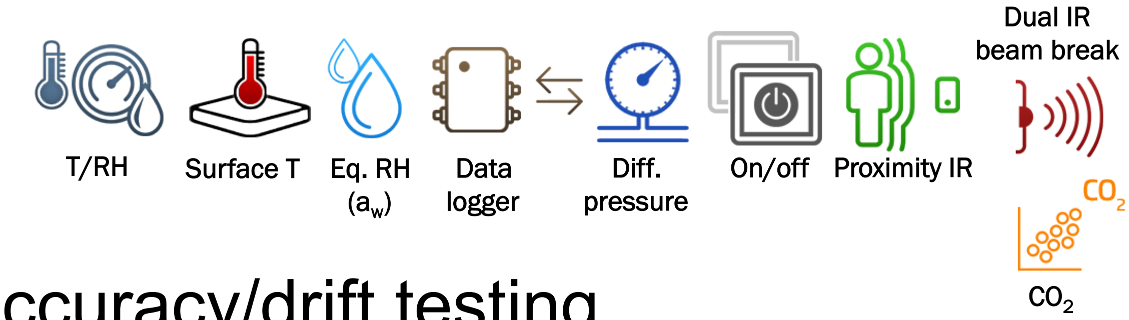
```
void loop()
{
  digitalWrite(SDPOWA, LOW);  // turn off microSD card to save power
  chip.turnOffSPI();  // turn off SPI bus to save power
```

```
    file.print(time);
    file.print(",");
    file.print(temperature);
    file.print(",");
    file.print(humidity);
    file.println();
```

Total hardware cost: ~$95 USD

Ongoing and future work

- Continued sensor development and calibration



- Stability/accuracy/drift testing
- Tutorial writing
- GUI development
- Wireless networking
  - Wi-Fi and Zigbee

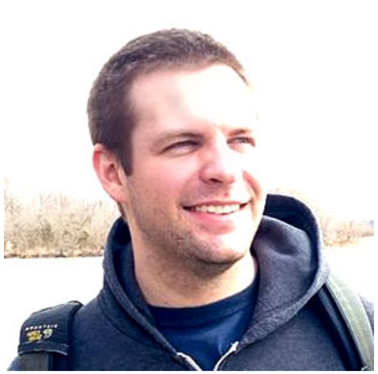
- Advisory board consultations
  - Microbiology, sensor/hardware dev, and building science

Acknowledgments

- Funding: **Alfred P. Sloan Foundation**
  - Paula Olsiewski

OPEN SOURCE
BUILDING SCIENCE SENSORS
http://www.osbss.com

Akram Ali


Zack Zanzinger

Torkan Fazli

Deion Debose

"Bobo" Dong

Questions/Comments

email: brent@iit.edu
web: www.built-envi.com

Built
Environment
Research
@ IIT

Advancing energy, environmental, and sustainability research within the built environment

ILLINOIS INSTITUTE
OF TECHNOLOGY

A quick plug:

Building science to advance research in the microbiology of the built environment (MoBE)

Supported by the Alfred P. Sloan Foundation

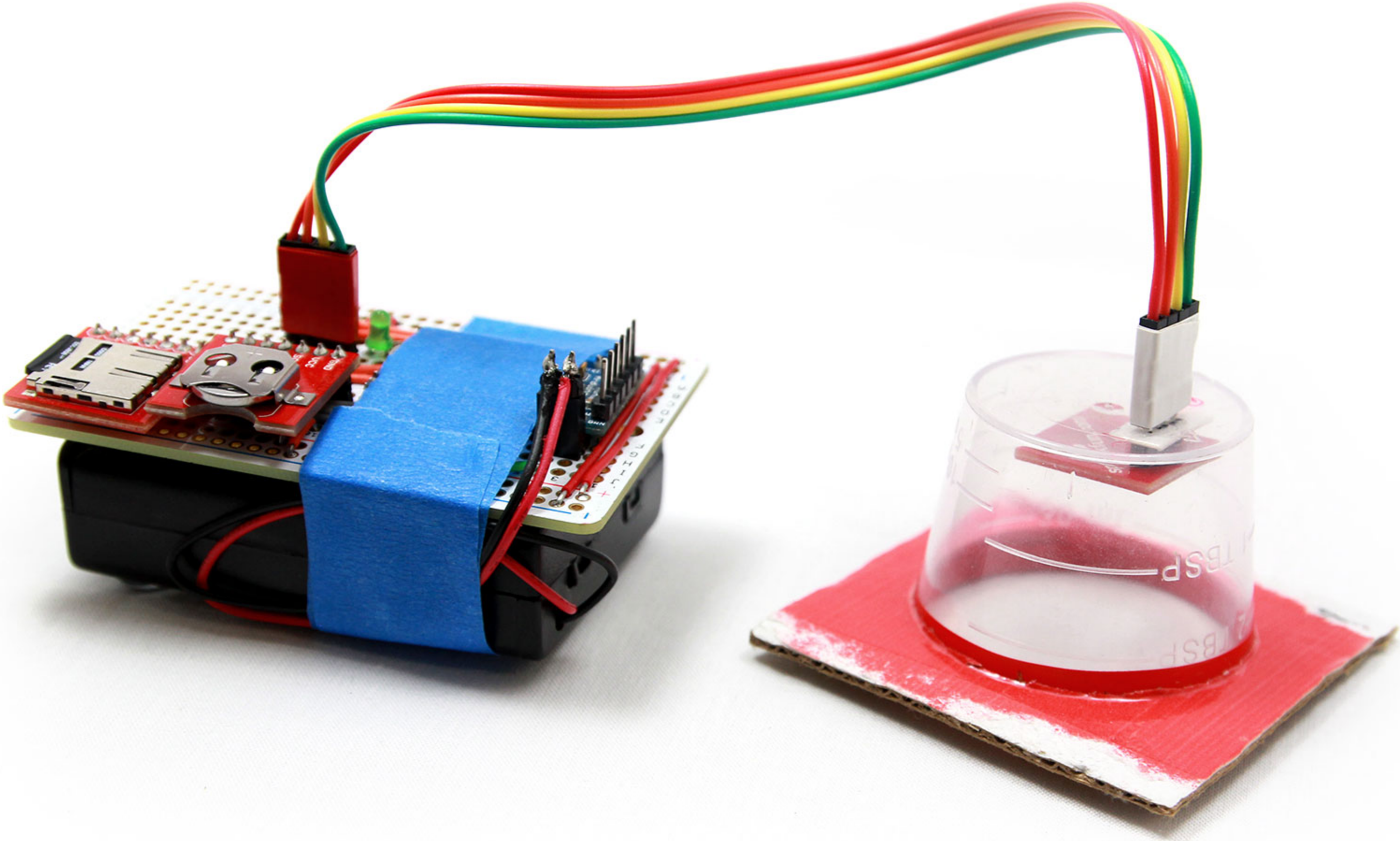
May 22-23, 2014 | Illinois Institute of Technology | Chicago, IL

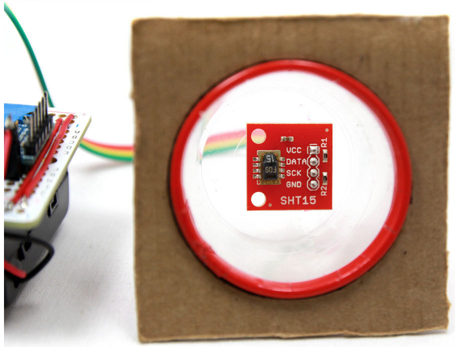
ALFRED P. SLOAN FOUNDATION

ILLINOIS INSTITUTE OF TECHNOLOGY

Full meeting report and transcript available for download:

http://built-envi.com/portfolio/mobe-building-science-workshop/





Custom equilibrium relative humidity

Air velocity (HVAC on/off)