The SPOT* Personal Thermal Comfort System

Alimohammad Rabbani and S. Keshav
School of Computer Science, University of Waterloo
Waterloo, Ontario, Canada
{amrabban, keshav}@uwaterloo.ca

ABSTRACT

Conventional centralized HVAC systems cannot provide office workers with personalized thermal comfort because workers in a single zone share a common air handling unit and thus a single air temperature. Moreover, they heat or cool an entire zone even if a single worker is present, which can waste energy. Both drawbacks are addressed by Personal Environmental Control (PEC) systems that modify the thermal envelope around a worker's body to provide personalized comfort. However, most PEC systems are both expensive and difficult to deploy, making them unsuitable for large-scale deployment. In contrast, we present the design and implementation of the SPOT* PEC system that is carefully designed for rapid and scalable deployment. Intuitive web-based interfaces for user controls allow SPOT* to be installed in only about 15 minutes, including user training. It is also low-cost because it uses the fewest possible sensors and a lightweight compute engine that can optionally be located in the cloud. We present the detailed design of the SPOT* system and results from a cumulative 58,000 hours of operation in 15 offices. We find that in our deployment SPOT* reduced the average absolute discomfort experienced by a typical user by ~67% compared to the same HVAC system in the absence of SPOT*.

CCS Concepts

•**Computer systems organization → Sensors and actuators;**

Keywords

Personal Environmental Comfort Systems, Personal Thermal Comfort

1. INTRODUCTION

Conventional centralized HVAC systems have long struggled to provide thermal comfort to individual office workers[1]. This is for two intrinsic reasons. First, workers typically cannot express their individual comfort preferences to the HVAC system. Instead, HVAC parameters such as temperature setpoints are chosen by building managers to provide comfort to the 'average' worker. Second, a single Air Handling Unit (AHU) is usually shared by multiple workers who may feel comfortable at different temperatures. Therefore, no matter what temperature the AHU is set to, one or more workers may be uncomfortable[2][1]. Moreover, they heat or cool an entire zone even if a single worker is present, which can be wasteful.

These drawbacks motivate the design of Personal Environmental Control (PEC) systems [2, 1] that provide workers in buildings with conventional centralized HVAC, with a *personalized* system for thermal comfort. Several PEC systems have been studied over the past two decades [2, 3, 4, 5, 6, 7]. However, each of these systems has some drawbacks. Some systems are onerous to use, for example requiring workers to wear devices on each wrist [4] or special clothing [7]; some are intrusive, using video cameras to watch workers [5, 6]; some can only be used for heating, not cooling [5]; others modify a worker's desk to add vents [2, 3], which is disruptive. Moreover, all these systems are quite expensive. Thus, none of them are suitable for immediate, large-scale, practical deployment.

In this paper, we present SPOT*, an individual thermal comfort system that can be rapidly and cost-effectively deployed. To achieve this primary goal, our sub-goals are:

- **Low per-unit cost**. The bill of materials for the prototype costs $185; lower costs can be expected with volume production.
- **Low operating cost**. In cooling mode, a SPOT* unit uses only about 12-16W, and in heating mode, it uses about 700W. We find that the median consumption in our trial is 4.7KWh/month, which costs about $0.5/month.
- **Plug-and-play deployment**. Setting up a single SPOT* unit takes only about 5 minutes.
- **Legacy compatibility**. There is no need to rebuild or modify existing central HVAC systems to use SPOT*.
- **Ease of use**. Users do not have to interact with SPOT* to benefit from it. They also don't need to wear any devices or special clothing.
- **Ease of user training**. It takes less than 10 minutes to train and prepare users for using SPOT*.

[1]We use the terms 'worker' and 'user' interchangeably.

[2]Per-office re-heaters or Variable Refrigerant Flow systems do allow individualized control, but are costly, so they are rare in North America.

BuildSys '16, November 16 - 17, 2016, Palo Alto, CA, USA

ISBN 978-1-4503-4264-3/16/11...$15.00
DOI: http://dx.doi.org/10.1145/2993422.2993578

In addition, SPOT* is built around a Raspberry Pi and both the hardware and software are Free and Open Source[3].

The rest of this paper is laid out as follows. Section 2 presents a background on quantitative comfort modeling and an overview of prior work. In Section 3 we explain our design goals and architecture. Implementation details and different hardware and software components are presented in Section 4. Section 5 evaluates SPOT*'s performance. Finally, in Section 6, we conclude the paper.

2. RELATED WORK

The literature in the area of personal thermal comfort is quite sparse, with a few seminal works from the 1990's, but little work until recently. In this section, we describe prior work in the area and also provide some background technical information in the area of personal comfort.

2.1 Personal Environmental Control Systems

Personal Environmental Control (PEC) systems seek to simultaneously meet two goals. First, it is impossible for a conventional centrally-controlled HVAC system to have 100% user satisfaction in buildings where multiple people share the same temperature setpoint [1]. Thus, a PEC system modifies a personal thermal envelope to increase user comfort. Note this increase in personal comfort may use additional energy, in that a PEC may re-heat cooled air. Second, buildings' HVAC energy consumption accounts for a significant portion of CO_2 emissions in the world [1, 8, 9], with about half of the emissions coming from commercial buildings [10, 11]. Occupancy-based control of HVAC systems, allows buildings to operate outside of comfort regimes when unoccupied [12, 13]. An PEC system can potentially meet this goal by allowing common areas in the building to be heated less in winter and cooled less in summer, while keeping users comfortable in their workspaces (though, as we have observed, if comfort is to be paramount, PECs may actually increase energy costs).

The seminal work on PEC system design is by Bauman et al. [2] who perform a field study and provide a group of users with a manually controlled desktop task/ambient conditioning (TAC) system. They show significant comfort improvements when users are equipped with the TAC. However, deploying TAC requires extensive modifications to user workspaces, including drilling holes into their desk, which is both expensive and disruptive.

Zhang et al. build a system that locally heats or cools crucial parts of body (i.e. hands, feet, face), to keep users comfortable [3]. While their system reduces energy consumption, and has a fine control over how users feel, it requires extensive user engagement. Moreover, although the paper does not present the cost of the system, it appears that it can be quite expensive, due to the use of individual heaters for hands and feet, and two separate face-level pedestal fans.

Recently (December 2014), the US ARPA-E has funded several research teams to develop Local Thermal Management Systems as part of the DELTA program [14, 7]. The primary goal is to allow buildings to be operated across a wider range of setpoints, reducing overall energy consumption. However, the projects have yet to reveal the details of their operation. We observe that most of the efforts appear to be directed towards the development of smart clothing that can perform heating or cooling functions. Forcing workers to wear special clothing appears to be both onerous and impractical.

The work closest to ours is our prior work on SPOT and SPOT+ [5, 6]. SPOT reactively controls a space heater to control room temperature in winter. In contrast, SPOT+ predicts occupancy to allow pre-heating. SPOT+ also uses an optimal control strategy to minimize energy consumption. Although interesting as prototypes, the use of multiple fine-grained sensors in SPOT and SPOT+, including a Microsoft Kinect, make them expensive ($1,000 per office). Neither system provides cooling and both are intrusive, because they need to process video images in real time.

2.2 Comfort Models

The best known model for human comfort in buildings is the Predicted Mean Vote (PMV) model [15]. The PMV model estimates an average worker's comfort level on the 7-point ASHRAE scale[4] using a function $f_{pmv}(\cdot)$:

$$pmv = f_{pmv}(t_a, \bar{t_r}, v_{ar}, p_a, M, I_{cl}) \qquad (1)$$

where *pmv* is the predicted mean vote and :

- t_a is the air temperature
- $\bar{t_r}$ is the mean background radiant temperature
- v_{ar} is the air velocity
- p_a is the humidity level
- M is the metabolic rate of a worker
- I_{cl} is the worker's clothing insulation factor

In prior work, we generalized PMV to the Predicted Personal Vote (PPV), where, during a training period, the system collects comfort votes from the user to extract two user-specific parameters a and b using least-squares regression. Then, the PPV is computed as a*PMV + b [6]. Since our focus is on *personal* comfort, this is also the metric we use in SPOT*.

A more recent alternative to the PMV model is the adaptive model for thermal comfort [16] that is commonly used in HVAC research for naturally ventilated buildings, and centrally air-conditioned buildings where occupants have adaptive controls (such as operable windows) [1]. SPOT* is targeted at legacy environments that lack such controls. Therefore, we use PPV noting that, in the absence of training by users, SPOT* defaults to using PMV instead of PPV.

2.3 Comfort and Air Movement

Many studies have found that air movement (using ceiling or desktop fans) is a energy- and cost-effective alternative to cooling air[17, 18, 19, 20]. In [2], Bauman et al. show significant improvements in comfort by providing users with a controllable desktop fan. Drawing upon these results, SPOT* uses the cooling effect of air movement in warm temperatures to improve user comfort. Note that this does not address humidity. In humid conditions, SPOT* relies on a central HVAC unit to provide comfort.

3. DESIGN

We now discuss the design of the SPOT* system. We note that our design has been approved for electrical safety

[3]https://github.com/AlimoRabbani/SPOTstar.

[4]Cold (-3), Cool (-2), Slightly Cool (-1), Neutral (0), Slightly Warm (+1), Warm (+2), and Hot (+3).

by the Canadian Electrical Safety Authority and is therefore eligible for deployment outside of our laboratory.

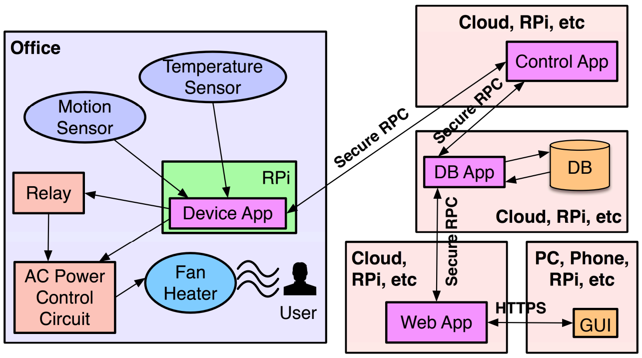

Figure 1: SPOT* has 5 main components. Actuation and sensing (left hand side of the figure), control application, data storage, web application, and graphical user interface. All software components communicate through RPC to allow the system to easily assume different configurations.

Figure 2: SPOT* hardware. The fan/heater is shown on the left side, and the actuation box is on the right side, with the small black sensor placed on top. Two power cords of the modified fan/heater are connected to the box. In a typical deployment, the fan is placed on top of the box, and the desktop footprint is smaller than a single sheet of A4 paper.

3.1 System Architecture

SPOT* includes both software and hardware components, and is fairly complex. We therefore first present an outline of the system architecture (Figure 1), then make a second pass over the design, presenting implementation details in Section 4.

We start with a description of the hardware placed in each office (the left hand side of Figure 1). This part of the system consists of a heating/cooling device (essentially a fan with a resistive heating coil), temperature and occupancy sensors, and solenoid actuators to turn the fan and heater on or off. The software entity that interfaces with the sensors and actuators is the *device manager application* (labeled 'device app' in the figure) executing on a Raspberry Pi (RPi). The device manager application reads sensor values from sensors and writes commands to the actuators. It sends the read sensor values to a *control app* and receives commands from it, that it carries out. The control app receives sensor measurements from the device app and uses this to decide to turn the fan/heater on or off and to select the fan's speed.

The past history of occupancy, temperature measurements, actuation events, and device training data are stored in a database (DB) by a *DB app*. This is useful for system administration and debugging, making control decisions based on user preferences, and providing feedback to the user in the form of historical charts. The DB can reside on the RPi for privacy, or on the cloud to provide better reliability and availability.

User interface with SPOT* is by means of a *Web app* that provides status information to SPOT* users and also allows them to communicate their comfort preferences to the control app. Finally, the *GUI* is the only software component of the system visible to users. In a networked setup, the GUI can be invoked on any device with an internet browser (e.g. PC, smart phone, RPi). For standalone installations, we have added a 7-inch touch screen to the RPi and installed it on the SPOT* box. The user can access the GUI directly from the box (see Figure 5).

4. IMPLEMENTATION DETAILS

In this section, we discuss the details of our implementation.

4.1 Actuation and Sensing

Actuation and sensing consists of a desktop fan/heater (Figure 2) to maintain user comfort, sensors to measure air temperature and occupancy[5], actuators to turn the fan/heater on or off and control its speed, an RPi that acts as both a network and a compute node, and a device application that runs on the RPi to communicate with other software elements of the system.

This component consists of two hardware devices that we designed and implemented: an actuation box, and a sensing box. The **actuation box** contains the RPi and actuators. The sensors are placed in a separate **sensing box** that is closer to the user. Here, we describe how each of these components are implemented and how they work together.

4.1.1 Heating and Cooling Device

To provide a worker with both heating and cooling, we modified the inexpensive and commercially available Royal Sovereign HFN-20 [21] personal fan/heater to control its heating coil and cooling fan independently. This is a simple change, requiring only that the second power cord be attached to the power connector terminals of the heating coil. The two power cords in the modified version are connected to the actuation box (See Figure 2). This allows relays inside the actuation box to turn the fan and the heating coil on or off, and the AC power control circuit to control the speed of the fan, up to its maximum air velocity of $2.1ms^{-1}$. We note that even at full speed, the fan noise is not very noticeable and we have received only one complaint about fan noise thus far.

[5]To save energy when the office is unoccupied.

4.1.2 Sensing

To reduce sensor costs, SPOT* uses default values for humidity and clothing level in the PPV equation, using standard values in the literature. Specifically, given that SPOT* is meant to be deployed in an HVAC-controlled office space, it assumes that the humidity is controlled to 50% [22]. Also, it assumes that an office worker's metabolic rate is a constant $1.2met$ [23] and wearing $0.6clo$ (corresponding to wearing trousers and a long-sleeved shirt, which is typical of an office) [24]. Thus, it requires only two sensors: a temperature sensor and a motion sensor. Data from these sensors are used to compute the PPV value as discussed in Section 2.2. Note that if these default values are incorrect, then users can easily correct them using the GUI described in Section 4.5.

The temperature and motion sensors are thermally separated and are placed in the sensing box along with an analog-to-digital converter (ADC) so that only digital values travel on the sensing communication link, reducing the effect of noise.

Temperature Sensor. To obtain temperature readings, we use the AD22100 surface-mount temperature sensor with $0.1\,^{\circ}$C resolution [25]. The temperature sensor has an analog output and is connected to the RPi through the ADC. This sensor's temperature values are later used in PPV calculations.

Occupancy Detection. We use the AMN22111 passive infrared human detection sensor [26]. It outputs analog values that are converted to values between 0 and 1000 on the RPi. When there is no movement, the sensor output values are approximately 500. Each movement causes the sensor to first generate one value close to 1000 and then another close to 0. The closer these values are to 1000 and 0, the greater the intensity of movement. Over a 30-second window, a standard deviation close to 0 indicates almost no movement, and thus no occupancy, while higher standard deviations correspond to more movement (See Figure 3). Note that there is an approximately 30s delay in detecting occupancy with this approach.

Analog-to-Digital Converter. Both the temperature and occupancy sensors generate analog outputs, and, since the RPi does not have analog inputs, we connect a MAX11612 analog-to-digital converter [27] to the RPi through the I2C serial pins. The ADC converts sensor outputs to 12bit digital signals and sends them to the RPi upon request.

4.1.3 Actuation

The actuation box physically controls the fan/heater using relays, a custom made AC power control circuit, and an RPi. It has two power controllers, one for the fan and one for the heater (Figure 4).

Relays. Two RPi GPIO output pins are connected to two electromechanical relays[6] to close and open the AC circuit of the fan and the heater independently. Upon receiving a command from the control app (see Section 4.1.4), the device app on the RPi sets the two GPIO outputs to 0V or 3V respectively to execute the command.

AC Power Control Circuit. We implemented a standard AC-control circuit [28] to control the fan speed. It limits the current going through the fan using a TRIAC that modulates the current based on a control signal from a 12-bit MAX5805 digital-to-analog converter (DAC). The DAC's output voltage is controlled by the RPi using the I2C serial protocol. The left hand side of Figure 4 shows the PCB that we manufactured.

Raspberry Pi. The RPi runs the Raspbian operating system, and is connected to and powered through our custom-built AC control circuit with a 40-pin ribbon cable. It runs the device app and controls status lights on the box by toggling output signals on GPIO pins. In a networked configuration, we use an Edimax EW-7811Un USB dongle to connect the RPi to the building's WiFi network.

4.1.4 Device App

The device app has several tasks. It collects data from sensors and transmits them to the control app locally or over the network. It also executes commands received from the control app. Using the I2C protocol, it reads sensor measurements from the ADC connected to the RPi.

The device app collects motion data twice every second, and sends it to the control app. Hailemariam et al. report that occupancy can be reliably detected by finding the standard deviation of the AMN23111 motion sensor [26] data every two minutes [29]. However, we found that the more sensitive AMN22111 motion sensor [26] allows us to lower the occupancy detection interval from 2 minutes to 30 seconds.

To reduce the amount of inter-process and network communication, the device app computes the standard deviation of raw motion values every 30 seconds, and sends only this value to the control app. It also reads and transmits the measured temperature every 10 seconds.

Upon receiving a command from the control app, the device app toggles GPIO outputs connected to relays to execute the command. In addition, it communicates with the DAC using I2C protocol to alter its output and set the speed of the fan. Due to the design of our selected fan/heater, to guarantee safe operation, we must make sure that the fan always spins with its maximum speed when the heating coil is powered.

4.2 Control Application

The control app listens for RPC connections from the device app. Each call from the device app updates either the temperature or the standard deviation of motion values. The control app passes these values to the DB app to be stored. It also makes a control decision based on occupancy and PPV, and invokes the appropriate procedures on the device app, as discussed next.

4.2.1 Occupancy Inference

The control app receives a standard deviation value O that represents movement intensity in every 30-second period from the device app. It determines if the station is occupied if this value exceeds a threshold $T_o = 17.25$ [29]. Note that in a shared office, background movements may cause false positives. To avoid this situation, we employ a low-pass filter in the form of a leaky bucket with water level L and capacity C as follows:

- L is 0 when the application starts.
- On update, if $O \geq T_o : L = min(L + 1, C)$
- On update, if $O < T_o : L = max(L - 1, 0)$

The control app infers that the station is occupied if $L =$

[6]We use electromechanical relays, rather than solid state relays, to reduce cost.

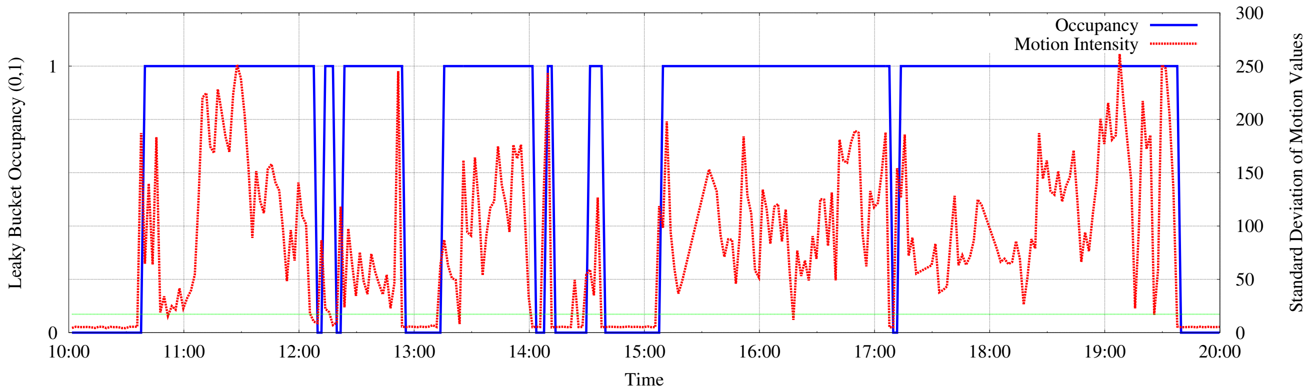

Figure 3: An example of occupancy inference, based on the standard deviation of motion values (i.e. motion intensity) during 30-second time windows over a 10-hour period. Ground truth occupancy is shown in blue and the motion threshold is shown in green. A threshold of $T_o = 17.25$ determines occupancy in each time interval. The results show 96% accuracy in detecting occupancy in our deployment.

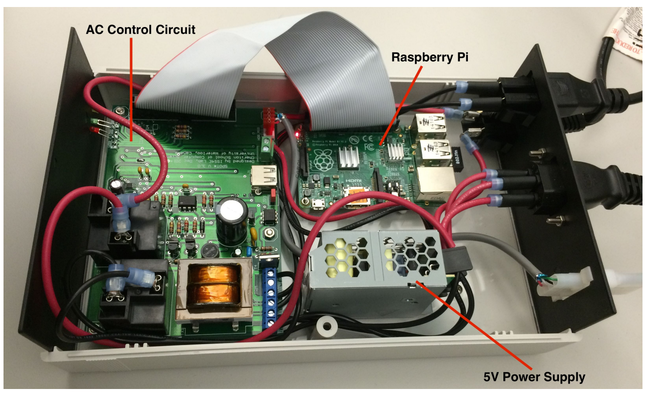

Figure 4: Inside of SPOT*'s actuation box. The AC control circuit on the left side communicates with the RPi to turn the fan/heater on or off, and set the fan's speed. The RPi, can connect to the building's WiFi network using a WiFi USB dongle, and a 5V 3A stable power supply keeps the box running.

Figure 5: For isolated installations of SPOT*, a 7-inch touch screen LCD is mounted on the actuation box. This LCD is connected to the RPi inside the box and provides a local GUI to the user. In this configuration, which has the best privacy protection, occupancy data remains physically in the office.

C and unoccupied if $L = 0$. Intermediate values for L imply that the user has recently left, recently arrived, or there was background movement close to the station. Therefore, it makes decisions oblivious to user presence when $0 < L < C$ (see Algorithm 1, described below).

4.2.2 *Control Decisions*

The control app makes reactive control decisions using the inferred occupancy and measured temperature data as shown in Algorithm 1. On receiving a motion sensor update (i.e. every 30 seconds), it updates the occupancy leaky bucket, then calculates the PPV value assuming a fan speed of zero. If the station is definitely occupied and the PPV value exceeds the comfort range T_c (i.e., the worker is too hot), the achievable PPV is re-calculated for each fan speed value in increments of $0.1ms^{-1}$, starting from zero, until the PPV (at that air speed) is within the comfort range or we reach the maximum possible fan velocity of $2.1ms^{-1}$. This determines the minimum fan speed needed to achieve a desired personal comfort level. On the other hand, if the station is occupied and the PPV value lies below the comfort range (i.e., the worker is too cold), then the heating coil is turned on, and, for safety, the fan speed is set to its maximum level. We use this approach because there is no direct relationship between fan speed and PPV - the PPV corresponding to a particular fan speed must be determined as the fixed point of a recursive relationship.

4.3 Data Store and DB Application

We use MongoDB to store data. We also implement a DB app that restricts database access to limited functions (e.g. inserting and querying occupancy and temperature, querying users, and inserting device state) for better security. The DB app communicates with the control app to log events and updates, and provide it with worker preferences. It also communicates with the web app to store worker pref-

Algorithm 1 Control app's MakeDecision procedure

if $L = 0$ **and** $Heat = true$ **then**
  $StopHeating()$
else if $L = 0$ **and** $Cool = true$ **then**
  $StopCooling()$
else if $L > 0$ **and** $Heat = true$ **and** $PPV > 0 - T_c$ **then**
  $StopHeating()$
else if $L > 0$ **and** $Cool = true$ **and** $PPV < T_c$ **then**
  $StopCooling()$
else if $L = C$ **and** $PPV > T_c$ **then**
  $S \leftarrow CalculateSpeed()$
  $StartCooling(S)$
else if $L = C$ **and** $Heat = false$ **and** $PPV < 0 - T_c$ **then**
  $StartHeating()$

erences and provide data to the web app for visualization. Because of SPOT*'s flexible architecture, we can run the DB app and the MongoDB storage locally on the RPi, or in the cloud.

4.4 Web Application

We design and implement the web application using the Flask microframework to:

- Collect votes from the user during training periods.
- Provide a manual override to the user.
- Visualize temperature, occupancy, and comfort data for users and administrators.
- Debug, monitor, and administer.

The web application runs on a WSGI Apache instance which is proxied through a publicly available Apache web server[7]. To ensure privacy and security, we use HTTPS [30] and require users to login with their credentials.

4.4.1 Training Period

SPOT* has an optional device training period to estimate the translation parameters a and b to obtain the PPV given the PMV [6] (the PMV is used by default if they never train the system). The controller uses the web interface to collect votes from the users based on the 7-point ASHRAE scale and matches them with the computed PMV value at the time of voting. Once the user ends the training it uses least squares linear regression on the collected points to determine the two parameters. The training typically takes about a day, and we advise users to train their devices once at the beginning of the summer and winter seasons.

4.4.2 Comfort Offset

In addition to infrequent training, we provide a manual offset override to allow users to adjust their comfort level as needed, such as when they are unwell, or when they are wearing more or fewer clothes than usual. The offset B_o is 0 by default and adjusts the PPV equation in the following way:

$$PPV = a * PMV + b - B_o \quad (2)$$

When $B_o < 0$, the user prefers cooler conditions, and when $B_o > 0$ warmer conditions are preferred.

[7]https://blizzard.cs.uwaterloo.ca/spotstar

4.5 Graphical User Interface

Standard browser-viewable content based on jQuery and Bootstrap is generated by the web app. In a networked setup, this GUI is accessible on users' desktop computers and smart phones.

In an isolated local SPOT* setup, we equip the actuation box with a 7-inch resistive touch screen LCD. The LCD is connected to the RPi using an HDMI cable through Adafruit's touch screen controller board [31]. In this setup, the data remains physically on-site and the user can access the GUI only on the box from a web browser user interface (see Figure 5).

Item	Prototype Price	Est. Volume Price
Raspberry Pi	\$40	\$5
WiFi dongle	\$10	\$5
Sensors	\$20	\$10
AC circuit components	\$50	\$20
Fan/heater	\$25	\$20
PCB manufacturing	\$20	\$10
Enclosures	\$10	\$5
Wires, connectors, etc	\$10	\$5
Total	**\$185**	**\$80**

Table 1: Bill of materials cost breakdown of hardware elements used in SPOT*. The table shows our approximate prototype cost, and the estimated mass-production price for each element. A \$17 touchscreen panel is used for the standalone version of SPOT*. However, our deployment currently consists only of network-enabled devices, which do not include the touch screen panel.

5. EVALUATION

We have deployed **45** SPOT* devices in both offices and cubicles in our campus. We sent an invitation to approximately 1500 building residents of four selected campus buildings; participants self-selected primarily because of their dissatisfaction with the existing HVAC systems comfort level. We deployed our systems in first-come-first-served order, until we ran out of devices. Thus far, only one person has withdrawn from the trial, because they left the university. On average, each device has been working for about 5.5 months, a total of ~58,000 hours of operation. We have collected data from our earliest deployed device for about 11 consecutive months, and for almost a month from our latest deployment.

Over the last 4 months, only two failures have happened, and re-plugging the device to the power outlet fixed the problem in both occurrences[8]. In this section, we compute the cost of SPOT*, measure its accuracy in detecting occupancy, evaluate users' comfort when using SPOT* with both subjective and objective measures, and explore its energy consumption.

5.1 Hardware Cost

Table 1 shows the bill-of-materials cost of hardware components used in SPOT* with approximate prices we paid,

[8]Users are provided with a 1-page user manual to troubleshoot the device on their own.

and estimated cost in mass production. In general, we assume that mass-production reduces prototype costs by 50%. We also assume that we can use the $5 Raspberry Pi Zero rather than the $40 Pi B+ that we used in our prototype. Note that the per-user cost of software and cloud servers is negligible for large deployments. Therefore, we do not include it in Table 1. Even for a single prototype, the bill of materials cost is only $185, dropping to and estimated $80 in volume production. Of course, additional costs would be incurred by a manufacturer for stocking, shipping, retail and so on, which could double or triple this estimate.

5.2 Occupancy Detection

Figure 3 shows how standard deviations of motion data translate into occupancy inferences during a typical 10-hour period. To estimate the accuracy of occupancy detection, we measured occupancy using both a video camera (with manual tagging of occupied periods) and the passive infrared sensor used in SPOT*. Over a 3-day period, SPOT* had a 96% accuracy.

5.3 Effectiveness In Maintaining Worker Comfort

We measured the effectiveness of SPOT* in maintaining user comfort in two ways. First, we compute the average absolute discomfort in the presence and absence of the SPOT* during the deployment period. Second, we measure how frequently users needed to manually override the system, as an indicator of how many times the users felt uncomfortable.

5.3.1 Average Absolute Discomfort

Gao et al. introduced the **average absolute discomfort** metric to quantify how uncomfortable a user feels [6]. Let $d(t)$ be the absolute discomfort at time t defined as

$$d(t) = max(|ppv(t)| - T_c, 0) \tag{3}$$

where threshold T_c determines a PPV range in which the user is comfortable. A T_c of 0.5 means the user is comfortable at time t if $-0.5 < ppv(t) < 0.5$. To be consistent with this work, we set T_c to 0.5 in our evaluation. Then, $\bar{d}$ or average absolute discomfort, is defined to be the average of $d(t)$ conditional on the user being present. Specifically, let $m(t) = 0$ when the workspace is not occupied and be equal to 1 when occupancy is detected. Then,

$$\bar{d} = \frac{\Sigma_t d(t)m(t)}{\Sigma_t m(t)} \tag{4}$$

To quantify SPOT*'s performance, we need to calculate the average absolute discomfort of users in as near identical environmental conditions as possible with and without SPOT*. Ideally, this would require A-B testing, with the same set of users monitored for a suitably long period of time without a SPOT* device, and then monitored again, this time with SPOT* present. Given our resource limitations, we opted for a shortcut that approximates this protocol. Observe that SPOT* performs no control actions when the workspace is unoccupied. So, the average absolute discomfort in the *absence* of SPOT* is simply the computed comfort value when the workspace is unoccupied. Thus, to measure average discomfort in absence of SPOT*, we define:

$$\bar{d^*} = \frac{\Sigma_t d(t)m'(t)}{\Sigma_t m'(t)} \tag{5}$$

where $m'(t)$ is 0 when the user is present and 1 when the the workspace is not occupied (i.e., $m' = 1 - m$). To be conservative, we further set $m'(t) = 0$ between 7pm and 7am regardless of occupancy. Therefore, we measure average discomfort in absence of SPOT*, only for the normal working hours, when our subjects are likely to be at their workstations. We recognize that this is not as good an experimental approach as standard A-B testing, but given that peak discomfort in summer happens during working hours, this is a reasonable–and much less resource-intensive–approximation.

In our deployment, we found that the average absolute discomfort for fifteen users, selected because they had been using SPOT* for the longest period of time, in the presence of SPOT* is 0.07 compared to 0.21 in its absence. Thus, SPOT* improves user comfort by 67% in this trial. Moreover, this value is substantially lower than the comparable value for SPOT [5], a similar reactive system, whose reported average absolute discomfort is 0.20, and roughly comparable to that of SPOT+ [6], a more complex predictive system, that reduces average absolute discomfort to 0.02. As we will see below, this is despite a nearly one order of magnitude reduction in energy consumption.

Figure 6 shows a comparison between average discomfort when SPOT* is in use and when it is not in use for this subset of the users in our deployment. We find that all selected users experienced increase comfort with SPOT*, with some users (at stations 5, 7, and 8) experiencing significantly greater comfort, which we corroborated by direct interviews. Although illustrative, we plan to report on a detailed study of user experiences with SPOT* in future work.

5.3.2 Manual overrides

The offset slider in our GUI is an indicator of how many times the users felt uncomfortable during the course of our deployment. Therefore we define E_s as the measure of user discomfort as:

$$E_s = \frac{\text{number of slider events}}{\text{total occupied hours}} \tag{6}$$

The average E_s for all selected users is 0.08, which considering weekly occupied hours, means that the average user changed the offset about three times every two weeks. Of course, this could be due to excellent thermal regulation by the building HVAC system. However, participants were self-selected to be those who were most dissatisfied with their existing comfort. Thus, we draw the conclusion that SPOT* was able to correctly set the participants' thermal envelope.

5.4 Energy Cost

Recall that SPOT*'s goal is to provide personal comfort. This may come at the expense of wasted energy, for example if the AHU is chilling air, and SPOT* is re-heating it. Unfortunately, we are unable to measure the energy wasted due this potential conflict. Here, we focus on the energy cost of running a SPOT* device, ignoring any AHU costs.

SPOT* has three sources of energy consumption:

- The actuation box, including the RPi and the AC control circuit (with a constant load of 3.5 watts),
- The fan, consuming linearly between 8 and 15 watts depending on speed,
- The heater coil, consuming 700 watts when turned on.

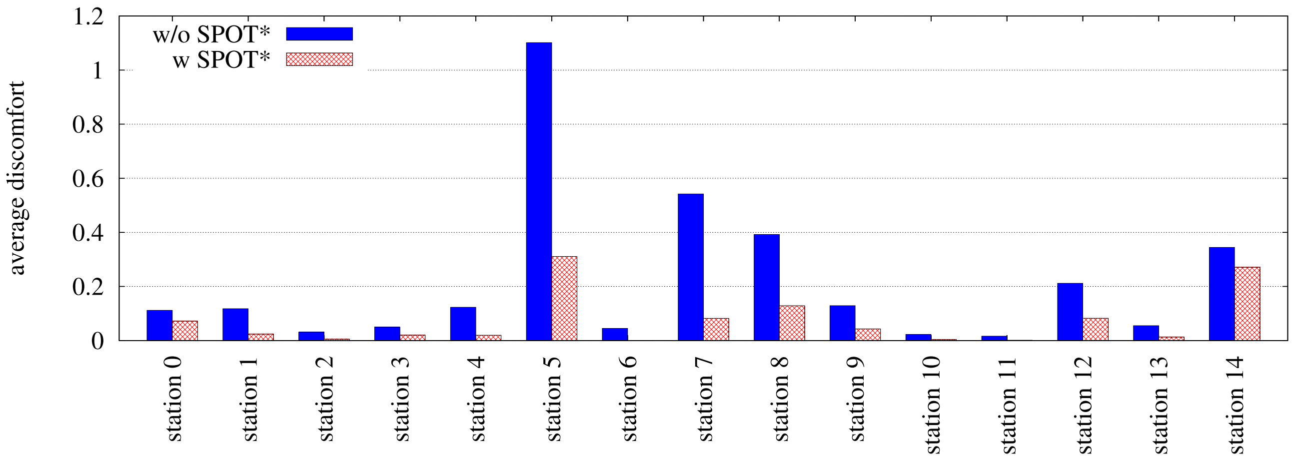

Figure 6: Average discomfort of users when SPOT* is being used, and when it is not being used. For each user, the average discomfort decreases significantly when SPOT* is maintaining comfort.

We estimate the total energy consumption of each deployed SPOT* instance by summing up the energy consumption of the three sources above.

We find that, despite being provisioned with a 700W heating coil, the average monthly energy consumption of each SPOT* device during the deployment is approximately 10 kWh, and the median consumption is only 4.7 kWh or less than 160 Wh/day. This median consumption is roughly equal to a single 40W light bulb left on for four hours each day. Compared to SPOT and SPOT+, which consume an average of roughly 150kWh and 230kWh per month, respectively [5, 6], SPOT* reduces energy consumption by an order of magnitude, while providing better or comparable user comfort. Note that the same users in the prior studies replaced SPOT/SPOT+ devices with SPOT* devices, so it is reasonable to compare these quantities. Figure 7 shows average monthly energy consumption for a subset of the deployed SPOT* devices.

6. DISCUSSION AND CONCLUSION

Motivated by the need for a practical system for personal thermal comfort, we presented the design and evaluation of the SPOT* system.

6.1 Meeting Design Goals

We now discuss how we met our goals of:

- **G1** Low per-unit cost
- **G2** Low operating cost
- **G3** Plug-and-play deployment
- **G4** Legacy compatibility
- **G5** Ease of use
- **G6** Ease of training

To meet goal **G1** (reducing system cost) SPOT* is built around the Raspberry Pi B+ single-board computer and a commodity heater/fan that we purchased from a hardware store. The RPi costs about $40 (newer versions cost $5!) and the heater/fan costs about $35. The remainder of the costs, for sensors and actuators, adds to this base cost, but we have tried our best to keep costs low. With mass production, we estimate that the bill of materials cost would be around $80, which is within striking range of the ARPA-E goal of $60/user [14].

Given that the closest related work is the two SPOT systems [5, 6], we now discuss how we achieve a similar goal as them, but at lower cost.

- SPOT uses a $200 Kinect for occupancy sensing. Instead, SPOT* uses a $20 passive IR motion sensor.
- SPOT also uses a Kinect to sense a worker's clothing level. Instead, we assume that the default clothing insulation factor is 0.6*clo* and provide a simple web-based user interface for workers to indicate that their clothing level is lower or higher than this default.
- SPOT* measures air temperature using an inexpensive temperature sensor and assumes that this temperature is identical to the background radiant temperature, so does not measure background temperature separately.

The use of a fan greatly reduces the cost of cooling, compared to a standard HVAC system. We found the median energy usage of the SPOT* system to be only 4.7 kWh/month, including both heating and cooling, for a dollar cost of approximately $0.5/month, which meets goal **G2**, low operating cost.

SPOT* is designed to be rapidly deployed (goal **G3**): all that needs to be done during a deployment is for the system to be placed on the user's desk and a sensor box placed near the user. A single power cord is plugged in and a laptop is used to enter configuration parameters in a central database using a GUI. The entire process takes only a few minutes.

Legacy compatibility (goal **G4**) is achieved trivially: if the central system provides adequate user comfort, then SPOT* does nothing, becoming active only when the personal thermal envelope of the user needs modification.

Ease of use and training (goals **G5** and **G6**) are met by means of the web-based GUI for training and comfort setpoint selection. There are no controls on the desktop system itself, only a flashing yellow light indicating normal operation. If the light is not flashing, the user simply unplugs and re-plugs the device and normal operation continues. Thus, the user manual is a single page of text.

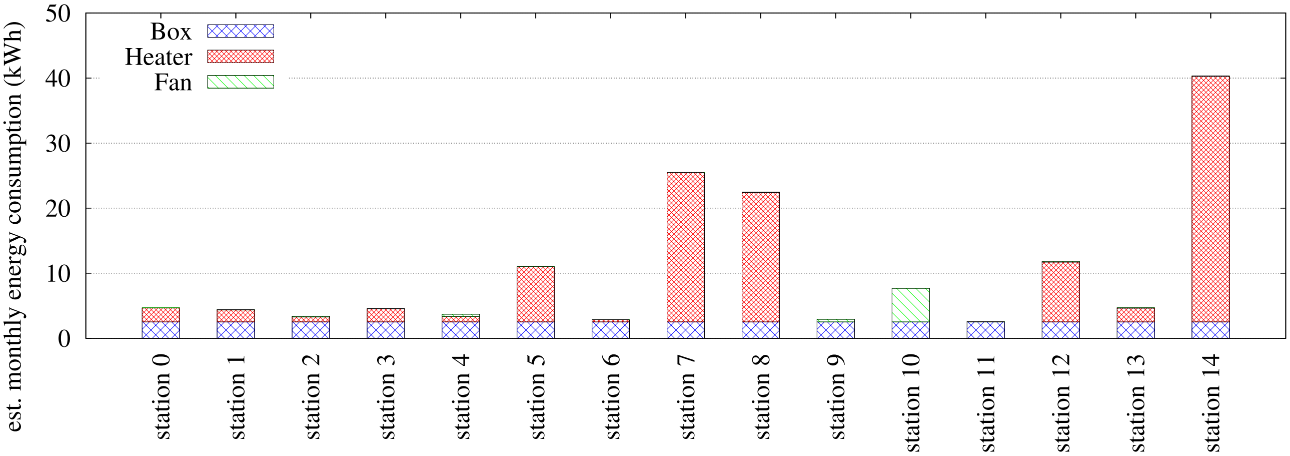

Figure 7: Estimated monthly energy consumption for each SPOT* device in kWh. A significant portion of consumption belongs to the heating coil and operating the system, and a small amount is consumed to power the fan. Station 10 is surprisingly warm, even in winter, hence the significant fan consumption.

6.2 Insights

We gained several insights from our work. First, in an air-conditioned building, most comfort parameters except temperature are kept stable enough not to affect the PMV equation significantly. Therefore it is not necessary to sense every component of the PMV equation. SPOT* estimates worker comfort with only temperature measurements and judicious choice of default parameters for the PMV equation. Errors in this estimate are corrected by a user interface that allows manual overrides. This also returns a certain degree of control over their own comfort to users, which they appreciate.

The design choice of using flexible, relocatable software components has also proven to be a good one. With little effort, we can configure a system to be standalone, suitable for privacy-sensitive workers, or to be networked, which opens up the possibility of coordinating SPOT* actions with that of a central HVAC, something we would like to pursue in future work. We hope that heater/fan manufacturers will, some day, build in a mote-like lightweight embedded compute platform into their devices, allowing us to deploy SPOT* on them, by moving the control logic, storage, and web app to the Internet cloud. This would be a fascinating use case for the Internet of Things.

Our choice of using a Raspberry Pi as the compute platform was not straightforward. We initially considered the Arduino, a smartphone running Android, and a Zolertia Z1 mote as alternatives. Indeed, our first deployment was using the Z1, which has a MSP430 controller, a minimal operating system, and a few 10s of KB of RAM. After struggling for some months with this platform, we were pleasantly surprised to find that the Raspberry Pi was not only much more powerful than the Z1, but was two-and-a-half times cheaper. We did not really need the smartphone screen, and the Arduino has no operating system and is not much cheaper than the RPi, hence our final choice. Through GUI-enabled interaction and information access, users can potentially train and personalize equipment such as office lighting. Thus, SPOT* can serve as the basis for a per-office Internet-of-Things deployment.

6.3 Limitations and Future Work

Our work suffers from one primary limitation. We were motivated, in part, by the goal of reducing building energy use by choosing a higher setpoint in summer and a lower setpoint in winter compared to standard operating procedure. Unfortunately, we were unable to persuade our building managers to actually let us modify the building temperature setpoint. Thus, we are unable to determine whether SPOT* will, indeed, reduce building energy consumption, or actually increase it, by re-heating chilled air to increase personal comfort. We are building a comprehensive building thermal simulator in current work that will allow us to partly address this deficiency. We are also keen to explore the joint optimal control of centralized HVAC systems when augmented with desktop devices.

Second, since the system was used by actual building residents, we could not estimate an energy-comfort curve: we could only measure the energy use that corresponded to their actual desired comfort.

Finally, our testing methodology does not use A-B testing, but estimates the gain in comfort when using SPOT* by measuring the PPV during unoccupied periods. Before doing a larger-scale deployment, such A-B testing should be carried out, with a stratified random sampling of the user population.

7. ACKNOWLEDGMENTS

This work would not have been possible without considerable support from Costin Ograda-Bratu and Milad Khaki who helped to design, implement, and deploy the system. Our grateful thanks to them both. We also would like to thank the anonymous reviewers of prior versions of this paper from the ACM eEnergy '15, ACM Buildys '15, ACM eEnergy '16, and ACM Buildsys '16 conferences, whose combined criticism helped us in greatly improving the quality of the paper.

8. REFERENCES

[1] RJ Dear, T Akimoto, EA Arens, G Brager, C Candido, KWD Cheong, B Li, N Nishihara, SC Sekhar, S Tanabe, et al. Progress in thermal comfort research over the last twenty years. *Indoor air*, 23(6):442–461, 2013.

[2] Fred S Bauman, Thomas G Carter, Anne V Baughman, and Edward A Arens. Field study of the impact of a desktop task/ambient conditioning system in office buildings. *ASHRAE Transactions*, 104:1153, 1998.

[3] H. Zhang, E. Arens, D. Kim, E. Buchberger, F. Bauman, and C. Huizenga. Comfort, perceived air quality, and work performance in a low-power task–ambient conditioning system. *Building and Environment*, 45(1):29–39, 2010.

[4] Mark Feldmeier and Joseph A Paradiso. Personalized hvac control system. In *Internet of Things (IOT), 2010*, pages 1–8. IEEE, 2010.

[5] P.X. Gao and S. Keshav. SPOT: a smart personalized office thermal control system. In *Proceedings of the fourth international conference on Future energy systems*, pages 237–246. ACM, 2013.

[6] P.X. Gao and S Keshav. Optimal Personal Comfort Management Using SPOT+. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pages 1–8. ACM, 2013.

[7] Hal Hodson. Smart clothes adapt so you are always the right temperature. *New Scientist*, January 30 2016.

[8] Diana Ürge-Vorsatz, LD Danny Harvey, Sevastianos Mirasgedis, and Mark D Levine. Mitigating co2 emissions from energy use in the world's buildings. *Building Research & Information*, 35(4):379–398, 2007.

[9] Mark Levine, Diana Ürge-Vorsatz, Kornelis Blok, Luis Geng, Danny Harvey, Siwei Lang, Geoffrey Levermore, A Mongameli Mehlwana, Sevastian Mirasgedis, Aleksandra Novikova, et al. Residential and commercial buildings. *Climate change*, 20:17, 2007.

[10] Luis Pérez-Lombard, José Ortiz, and Christine Pout. A review on buildings energy consumption information. *Energy and buildings*, 40(3):394–398, 2008.

[11] A Chen. Closing in on zero-energy buildings. *EETD Fall Newsletter*, 8(2), 2009.

[12] V. L. Erickson, Y. Lin, A. Kamthe, R. Brahme, A. Surana, A. E Cerpa, M. D Sohn, and S. Narayanan. Energy efficient building environment control strategies using real-time occupancy measurements. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 19–24. ACM, 2009.

[13] V. L. Erickson, A. Á M. Carreira-Perpiñán, and A. E. Cerpa. OBSERVE: Occupancy-based system for efficient reduction of HVAC energy. In *IPSN 2011*, pages 258–269. IEEE, 2011.

[14] ARPA-E. Delivering efficient local thermal amenities. http://arpa-e.energy.gov/?q=arpa-e-site-page/view-programs. Accessed: Jan 2016.

[15] Poul O Fanger et al. Thermal comfort. analysis and applications in environmental engineering. *Thermal comfort. Analysis and applications in environmental engineering.*, 1970.

[16] Richard De Dear and Gail Schiller Brager. Developing an adaptive model of thermal comfort and preference. *ASHRAE Transactions*, 104(1):145–167, 1998.

[17] H. Zhang, E. Arens, S. Fard, C. Huizenga, G. Paliaga, G. Brager, and L. Zagreus. Air movement preferences observed in office buildings. *International Journal of Biometeorology*, 51(5):349–360, 2007.

[18] Jørn Toftum. Air movement–good or bad? *Indoor Air*, 14(s7):40–45, 2004.

[19] Tyler Hoyt, Hui Zhang, and Edward Arens. Draft or breeze? preferences for air movement in office buildings and schools from the ashrae database. 2009.

[20] Tyler Hoyt, Kwang Ho Lee, Hui Zhang, Edward Arens, and Tom Webster. Energy savings from extended air temperature setpoints and reductions in room air mixing. In *International Conference on Environmental Ergonomics 2009*, 2009.

[21] Royal Sovereign International. Oscillating Fan Heater - HFN-20. http://www.royalsovereign.ca/appliances/heaters/heater-hfn-20-785.html. Accessed: Jan 2015.

[22] Canadian Center for Occupational Health and Safety. Thermal comfort for office work. http://www.ccohs.ca/oshanswers/phys_agents/thermal_comfort.html. Accessed: June 2016.

[23] The Engineering Toolbox. Metabolic rate. http://www.engineeringtoolbox.com/met-metabolic-rate-d_733.html. Accessed: June 2016.

[24] Clothing insulation. Wikipedia. https://en.wikipedia.org/wiki/Clothing_insulation. Accessed: June 2016.

[25] Analog Devices. Voltage Output Temperature Sensor with Signal Conditioning. http://www.analog.com/static/imported-files/data_sheets/AD22100.pdf. Accessed: Jan 2015.

[26] Panasonic. MP Motion Sensor NaPiOn. http://www3.panasonic.biz/ac/e/control/sensor/human/napion/index.jsp. Accessed: Jan 2015.

[27] Maxim Integrated. Low-Power, 4-/8-/12-Channel, I2C, 12-Bit ADCs in Ultra-Small Packages. http://www.maximintegrated.com/en/products/analog/data-converters/analog-to-digital-converters/MAX11612.html. Accessed: Jan 2015.

[28] Giorgos Lazaridis. Voltage Controlled AC Light Dimmer. http://pcbheaven.com/circuitpages/Voltage_Controlled_AC_Light_Dimmer/, January 2010. Accessed: Jan 2015.

[29] Ebenezer Hailemariam, Rhys Goldstein, Ramtin Attar, and Azam Khan. Real-time occupancy detection using decision trees with multiple sensor types. In *Proceedings of the 2011 Symposium on Simulation for Architecture and Urban Design*, pages 141–148. Society for Computer Simulation International, 2011.

[30] The Internet Engineering Task Force. HTTP Over TLS. http://tools.ietf.org/html/rfc2818, May 2000. Accessed: Jan 2015.

[31] Adafruit. Tfp401 hdmi/dvi decoder to 40-pin ttl breakout - with touch. http://www.adafruit.com/products/2219. Accessed: Jan 2015.