Internet of Things Smart Calendar
Project Proposal

Advisor: Dr. Malinowski

Cole Lindeman, Jason Morris

Table of Contents

Abstract

Many university professors like to post their schedule and office hours right outside their door so that students know when they can find them. Unfortunately, sudden changes to a professor's agenda can make it hard to keep that schedule up to date. The Internet of Things Smart Calendar seeks to offer an easy solution. The Internet of Things Smart Calendar is a device that interfaces with Google Calendar and keeps an updated display of the professor's Google Calendar. It provides the service of displaying a professor's office hours, advertisements, announcements, and other relevant information that the professor would like the students to know about.

Creating an approach to this solution involved many hours of planning and preparation. Research was done on relevant projects that were similar to this one. It was decided that the project would be powered by a Raspberry Pi and a touch screen would be used as the display. The Raspberry Pi will be interfacing with motion and door sensors and make decisions as to what should be displayed. So far, a user interface has been created using HTML and CSS.

Continuing to create this solution utilizing new and up to date technology should give teachers around the world an easier way to manage their schedule and provide their students with desired information.

I. Introduction

This project is an Internet of Things Smart Calendar that organizes a collection of input connections and uses the information to display data as well as send messages.

The Smart Calendar will display the professor's calendar as well as advertisements on an interactive monitor outside his room. Sensors will be used to recognize when students are nearby for tracking purposes; the Smart Calendar will also be able to display ads for passersby as well as display the calendar for people stopping at the Smart Calendar.

The Calendar will be able to contact the professor to send messages. The Smart Calendar will send an alert if a student is outside the office door during office hours. There will also be a feature to optionally leave messages using the Calendar which will be sent directly to the professor.

The Smart Calendar display will be placed directly outside the professor's office while the main controller device will be in the ceiling. Additionally, motion sensors will be placed along the wall and a magnetic sensor will be attached to the office door.

II. Review of Literature and Prior Work

Similar projects involving a smart calendar have been created. One person has created a project that displays their google calendar on one half of the screen, and the local weather on the other half [1]. It automatically shuts down at night and has buttons on the side to toggle the calendar between monthly, weekly, and daily views of the calendar. This particular project is powered by a Raspberry Pi. However, it does not have motion sensors or touch screen interactivity.

Another similar product that has been created is called the DAKboard [2]. The DAKboard is a customizable wall display that displays relevant information to the user's life, such as pictures, weather, and upcoming calendar events. This display also lacks motion sensors and a touch screen.

III. Applicable Standards and Patents

A. Standards to Consider

Since this project will involve electrical wires, a large concern is safety. The project will be installed on a wall in a building where many people walk by and possibly interact with the device. OSHA has a myriad of safety standards for electronic equipment and

electrical wiring. These standards must absolutely be adhered to as installing the device in the building could be breaching building code and would have to be taken down. As an example, OSHA standard 1910.305(f)(1) demands any “conductors used for general wiring shall be insulated” which means that anything used has to be jacketed or covered somehow so no conductors are exposed [3].

It would also be a good plan to follow one or more sets of cyber-security standards as the device will connect to the Internet and could potentially have a multitude of possible security breaches. If the device or any database that the device uses is comprised, unwanted advertisements and announcements could appear on the Smart Calendar.

Some helpful tools applicable to solving functionality are Javascript, HTML and CSS which all fall under W3C standards [4]. Any web applications should be designed and tested with the wealth of standards, specifications and guidelines provided by this platform. These standards are intended to improve quality whenever applicable and are also free to view.

B. Patents for Similar Ideas

There are some patents for features on electronic calendars that could possibly be used to implement required functionality. In order to avoid possible patent troubles, the features should be implemented using different methods.

One method that cannot be used is using email to update the Smart Calendar. This method is patented by Xerox and is very specific on patenting the idea of updating an electronic calendar with information in an email message [5]. Luckily, there are plenty of other options on how to update the Smart Calendar. Avoiding email updates altogether could be possible using a direct update link or by having a website host retrievable updates.

There are no current plans to obtain a patent for any of the other methods used for the functionality. The project could possibly later be packaged as a product as well as reproduced and redistributed; in order to allow this to be possible, no patents can be violated.

IV. Subsystem Level Function Requirements

The Raspberry Pi 3 will interface with all sensors and devices. It will also communicate with the internet using its onboard ethernet port. It needs a power supply connected to the power inlet from a 120V conventional power line. The Pi will be resting inside a protective case in the ceiling.

The wires coming in/out from the sensors need to be attached to the Raspberry Pi's digital I/O pins. Power will be siphoned out of the Vcc and Ground pins for the sensors' usage using more wires. The voltage may have to be regulated to an appropriate rating for use with the sensors which might require simple circuitry. The sensors will be placed in necessary positions to be able to track required information. The wires will have to reach the Pi from these positions.

The monitor for display will be mounted on the wall underneath the Raspberry Pi. The specific monitor ordered needs an HDMI cable, a USB cable and its own power supply. The USB and HDMI cables will need to reach the Pi and the power supply needs to source its own power from a 120V conventional power line.

A. Inside the Pi

The Raspberry Pi will be running multiple processes to make meaningful information out of sensors, control devices and use the internet. These processes include:

- GUI Process

This process will handle the graphics going to the in/out connection with the touch display monitor connected via USB and HDMI. Some of the data displayed in the graphics (Doctor Malinowski's google calendar, class advertisements, weather, etc.) will be grabbed from the internet.

This process also handles the message interface which allows users to leave a message using software keyboard input typed via the touch screen. The messages will then be stored in memory to be sent. There will also be interactivity for people to use the GUI to control the information being displayed using the monitor's touch controls. When not being interacted with, this process will default to displaying advertisements to passers-by and switch to displaying the calendar for people who stop at the calendar.

- Tracking Process

In order for the board to make sense of all of the sensor data, it needs a process to read pin information and make decisions. The process will gather information from all motion sensors and the open door sensor by reading the board's specific drivers for the digital I/O pins that the sensors are connected to. It will then form coherent, human-readable tracking data and send it out to memory for storage. The Tracking Process will also tell the GUI Process when someone stops in front of the Smart Calendar.

- Memory

A portion of read/write memory will be allocated on the Raspberry Pi 3's SD card storage for temporary data. This data will be sent to the Internet as a data file and uploaded to Google Drive.

- Sleep/Wake Up Process

A process will be reading the onboard Real Time Clock. When it is time to sleep, the process will shut down any unnecessary processes and set the digital I/O pins to low power to unpower the sensors. When it is time to wake up, the process will start up any other necessary processes and power the digital I/O pins to power the sensors.

- Door Lock Process

This process will accept a secure message from the internet and then use the digital I/O pin to communicate with the electronic door lock to unlock the door.

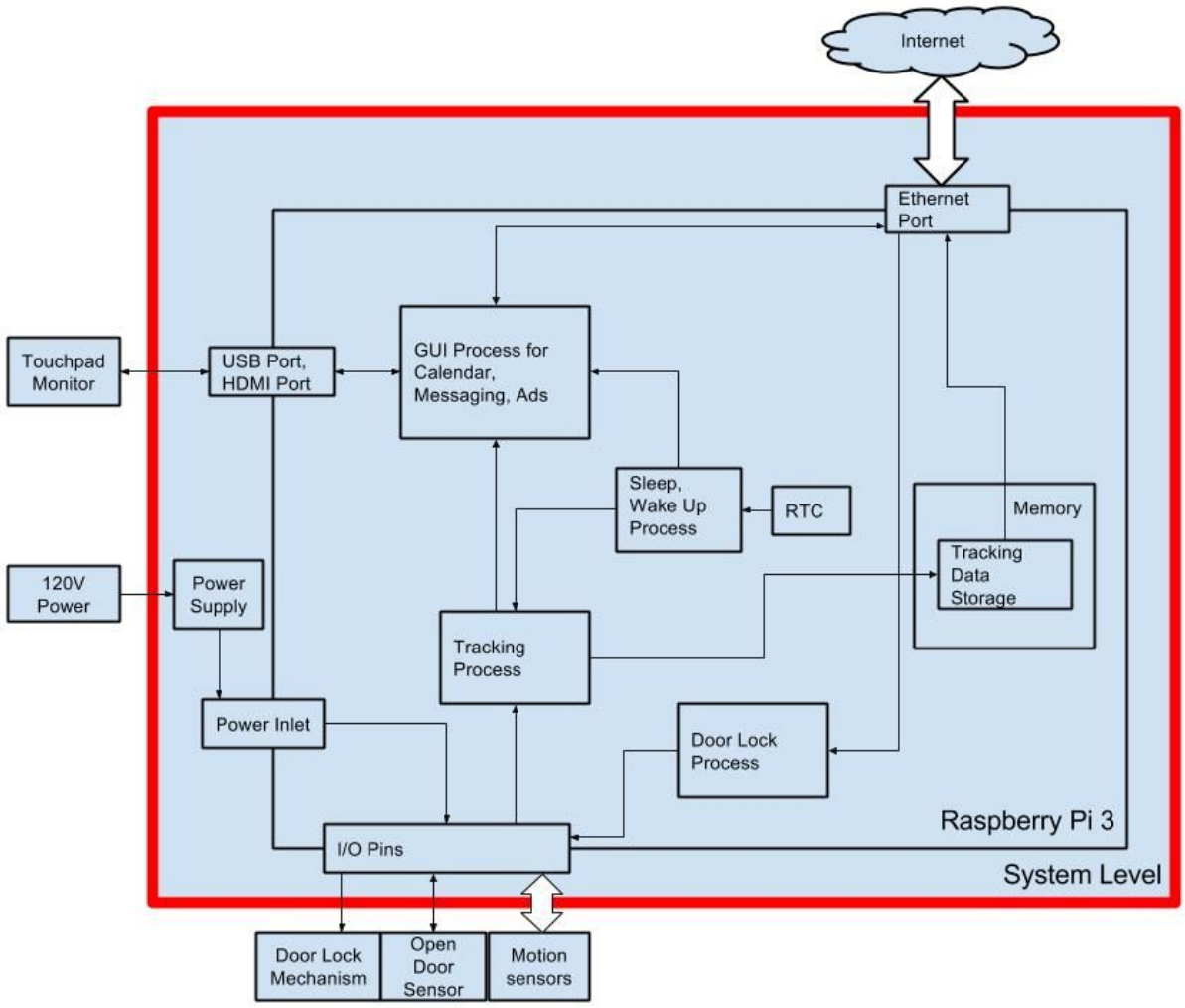

Figure 1. Subsystem Level Diagram

A diagram detailing the connections between subsystems and I/O.

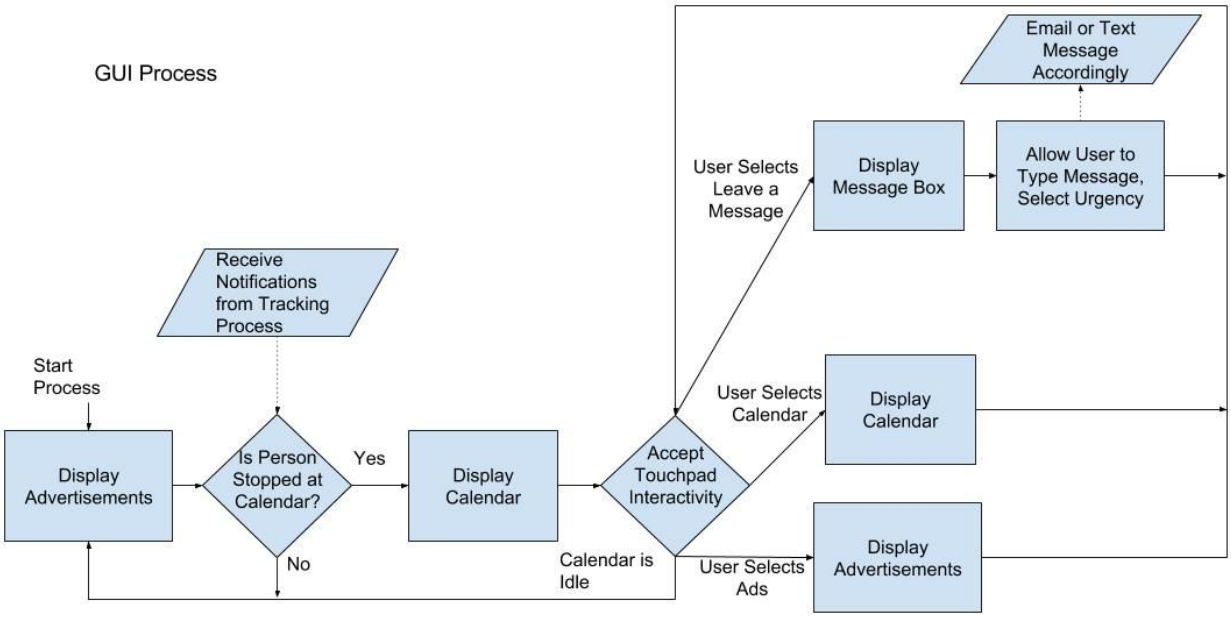

Figure 2. Gui Process Block Diagram

A block diagram detailing the design for the process that controls what is displayed. The process also handles messaging.

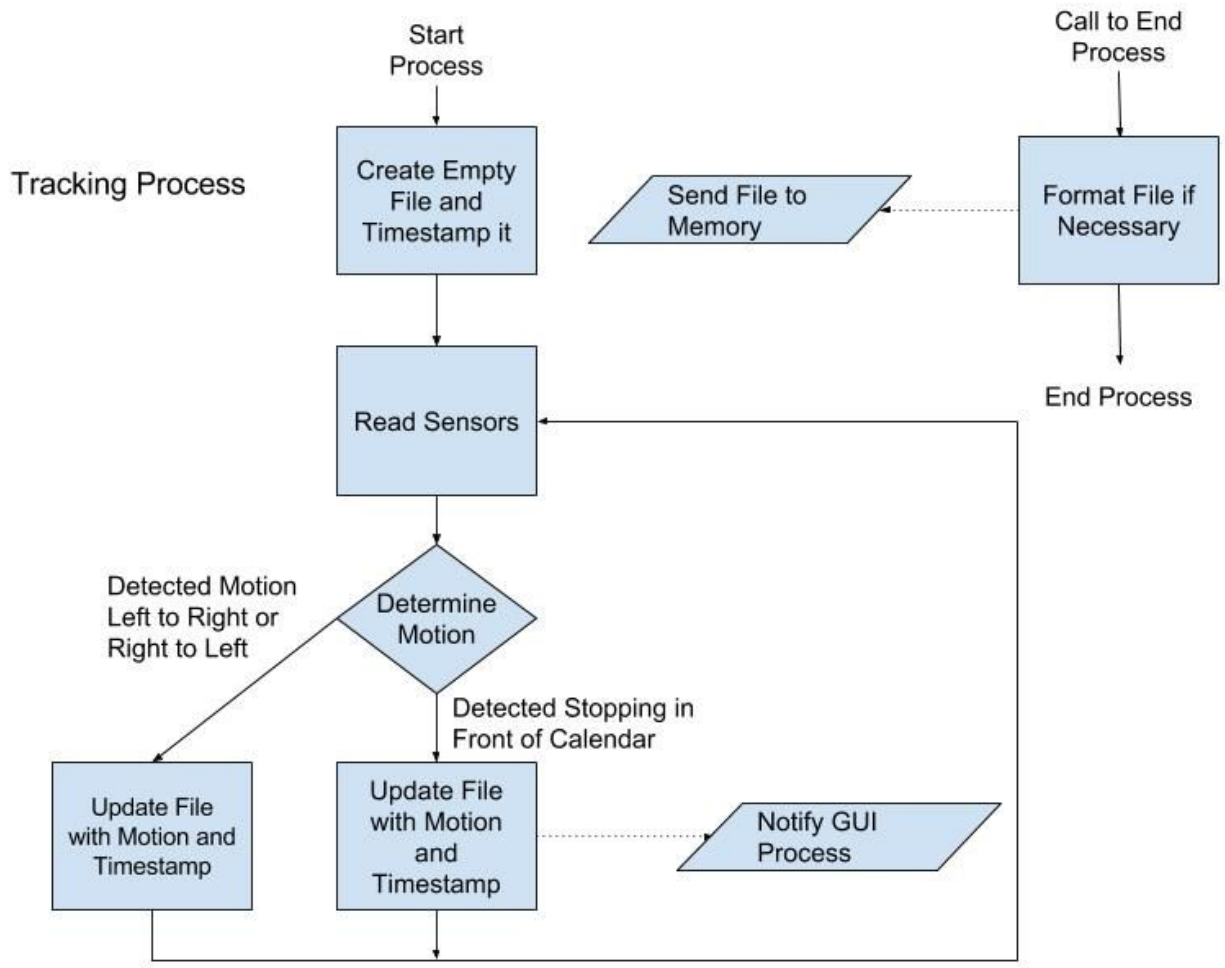

Figure 3. Tracking Process Block Diagram

A block diagram detailing the design for the process that reads the infrared sensors and sends notifications to other processes. The process also creates tracking data and uploads the data to an online database when it is told to end.

B. Modes of Operation

In order to conserve power, the Smart Calendar will also enter modes of operation where the system sleeps.

- Off

The device is completely powered off. This could happen if the device is manually unpowered or if the power goes out. From this mode, the device needs to receive power again and turn on. This might require manual interaction with the device. Once it receives power again and is turned back on, it will enter Startup.

- Low Power

The device is in a power saving mode, the display is off, and all inputs are not being read; this mode is for power saving when nobody is around to look at the calendar overnight. The Raspberry Pi cannot shut off completely because it cannot wake itself up, but it can enter sleep mode.

- Startup (transition)

The GUI is loaded in kiosk mode, the display is off, and all inputs are not being read; this transition is for loading features out of Low Power mode into Powered On mode. The Pi can automatically enter Startup from Low Power based on default criteria like a given time of day.

- Powered On

The device is actively retrieving all inputs, displaying functionality on the monitor, and deciding if to send alerts; this mode is for implementing all designated Smart Calendar features.

- Shutting Down (transition)

The device sends stored tracking data to the internet, the display is off, and all inputs are not being read; this transition is for stopping all features and going to Low Power mode from Powered On mode. The Pi can automatically enter Shutting Down from Powered On based on default criteria like a given time of day.

V. Engineering Efforts Completed to Date

Most of the required functionality of the Smart Calendar has already been considered. All of the required techniques to perform the functionality have a generic design as included in the Subsystem Level Functional Requirements section.

Many useful programming languages for this project have already been studied and researched. This includes Python, HTML paired with Javascript/Ajax/PHP, Bash, and various other Linux commands.

Some programming is done or is partially complete. For the display, there already is a working framework with a section for calendar, messages, and advertisements. In the background, example work has been done for interprocess communication.

Simulation has been done on a Raspberry Pi 2 running Ubuntu MATE and a virtual machine running Xubuntu in order to test if the programming works correctly. Since any version of Unix should work similarly, this simulation should provide a good basis for what will happen when the same program is ran on the Smart Calendar.

VI. Parts List

- Waveshare 10.1 inch 1280x800 IPS LCD Capacitive Touchscreen with case
    - $118.99
- Raspberry Pi 3 with power supply, case and heatsinks
    - $51.94
- Sandisk 32GB microSDHC card with normal SD card adapter
    - $10.59
- Aleko magnetic reed switches
    - $9.99
- Emy passive infrared motion sensor detector modules
    - $5.49
- Ethernet*, HDMI, USB and digital I/O cables
    - $14.89

Total cost: $211.89

*Ethernet cable is in house and is not included in price.

VII. Deliverables and Schedule

Tasks will mostly be done by one individual given their specific focus and skills. Some tasks will involve collaboration. The deliverables for Spring semester haven't been assigned or announced yet and are not included.

Table 1
Schedule for Completion

Week		**Jason's work**	**Cole's work**
11/27/16	12/3/16	Product Proposal presentation Finalize Product Proposal	
12/4/16	12/10/16	**Final Exam**	
12/11/16	12/17/16	**Final Exam**	
12/18/16	12/24/16	**Winter Break** begins	
12/25/16	12/31/16	**Winter Break**	
1/1/17	1/7/17	**Winter Break**	

1/8/17	1/14/17	**Winter Break** ends	
1/15/17	1/21/17	**Spring Semester** begins Write Python code to host HTTP web server for Ajax to communicate with	
1/22/17	1/28/17	Write XML code using Ajax to direct browser	Continue writing Python code to communicate with Ajax
1/29/17	2/4/17	Write HTML code to direct browser back to ads when idle for long enough	Setup Raspberry Pi Setup monitor for Pi
2/5/17	2/11/17	Write javascript for ads that "follow" passersby	Figure out reading, writing, and permissions for I/O pins Connect sensors to Pi
2/12/17	2/18/17		Write Python script to poll I/O pins Write Python script to enable and disable I/O pins
2/19/17	2/25/17	Find method to upload text files Write script to use method to upload tracking text file	Write Python script to track movement with IR sensors
2/26/17	3/4/17		Write Python script to compile movement information into a text file
3/5/17	3/11/17	Write Python script to send commands to Ajax using movement information	Write Python script to communicate with door lock
3/12/17	3/18/17	**Spring Break**	
3/19/17	3/25/17	Test Internet communication	Write script for sleep/wakeup process
3/26/17	4/1/17	Test mount setup for project	
4/2/17	4/8/17	Mount project	
4/9/17	4/15/17	Spare time in case of changes	
4/16/17	4/22/17	Spare time in case of changes	
4/23/17	4/29/17	Spare time in case of changes	

Some work has been divided up such that one person can focus on one task while the other on another; this is due to specific focuses and skillsets. Sometimes tasks call for

collaboration and require both partners to work at simultaneously in which case both are responsible.

VIII. Discussion and Future Directions

For the moment, work needs to be done with Ajax to control the web browser from outside processes. This is the major boundary from continuing to program for the project.

Optionally, there are plans to implement many other features. The first one likely to be implemented would be an automatic service to pull updates from Github. This feature would likely be turned off in the final release as this could introduce unforeseen issues and possible breaches in security.

There is also talk on adding new tabs to the Smart Calendar for added interactivity. If all work is completed early, work on these additional features will start. These features could be local weather information, local weather alerts, or a simple game.

Additionally, as a side project, Jason has shown interest in making a companion phone application to the Smart Calendar. The application would have many of the same features of the Smart Calendar, but for a mobile platform.

IX. References

Citations

[1] Kmccb (2016, Apr. 7). *Raspberry Pi Framed Informational Display - Google Calendar, Weather, and More..* [Online]. Available: http://imgur.com/gallery/z94Vr

[2] M. Archambault. *DAKboard Is a Customizable Wall Display for Photos, Calendar Events, and Weather* [Online]. Available: http://petapixel.com/2015/08/19/dakboard-is-a-Customizable-wall-display-for-photos-calendar-events-and-weather/

[3] (2007, Feb. 14). *Wiring Methods, Components, and Equipment for General Use* [Online]. Available: https://www.osha.gov/pls/oshaweb/owadisp.show_document?p_table=STANDARDS&p_id=9882

[4] *Standards - W3C* [Online]. Available: https://www.w3.org/standards/

[5] JL. Meunier, C. Hagage, S. Castellani, D. Proux, E. Cheminot, F. Segond, “System and method for updating an electronic calendar” U.S. Patent 9 436 649, Sept., 6, 2016.

See Also

[6] D. J. Barrett, *Linux Pocket Guide,* 2nd ed. Sebastopol, CA: O'Reilly, 2004.

[7] *PHP 5 Tutorial* [Online]. Available: http://www.w3schools.com/php/default.asp

[8] (2016). *Python 2.7.12 Documentation* [Online]. Available: https://docs.python.org/2.7/

[9] *Linux Documentation* [Online]. Available: https://linux.die.net/

[10] *Ajax jQuery API Documentation* [Online]. Available: http://api.jquery.com/jquery.ajax/