

IoT Smart Calendar Display

by Cole Lindeman and Jason Morris

Advisor: Dr. Malinowski

Date: 10/6/2016

Functional Description:

An Internet of Things Smart Calendar that organizes a collection of input connections and uses the information to display data as well as send alerts.

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. There will be a feature to leave messages from the calendar.

System Level Inputs

- Information from the Internet via a wired connection
 - Google Calendar Web page (in HTML) of Dr. Malinowski's calendar
 - GPS data tracking packets sent from an app on Dr. Malinowski's phone
 - Automatic updates by doing a git pull request to our remote master branch
 - One page of an optional urgent announcement
- Motion sensor data from multiple attached sensors to detect people walking by or stopping in front of the calendar
- Touch screen monitor inputs for interactivity for people stopping at the calendar
 - Interactivity is limited to looking at calendar, opting to leave a message, or scrolling through advertisements for courses for next semester
- Keyless entry communication (possibly from an app over the Internet) for Dr. Malinowski to open his office door
- Mini USB keyboard, software keyboard, or USB microphone for people to leave a message for Dr. Malinowski to see later
- Sensor to detect if door is open
- Conventional 120V power cable to power device and sensors

System Level Outputs

- HDMI or USB connection to visual display on monitor to display calendar or advertisements
- Information to the Internet via a wired connection
 - Send email or text alert if special cases are met
 - Smart Calendar has noticed student is waiting and office hours are concurrent or Dr. Malinowski is on campus
 - Send or store recorded messages to online database
 - Send tracking data to an online database
- Wired connection to mechanical door lock system*

*May not be implemented; permission has been denied.

Modes of Operation

- Low Power: Device is in a power saving mode, display is off, and all inputs are not being read

For power saving when nobody is around to look at the calendar overnight
- Startup: Operating system loaded, GUI loaded in kiosk mode, display is off, and all inputs are not being read

For loading features out of Low Power mode into Powered On mode, device could automatically enter Startup from Low Power based on default criteria like a given time of day
- Powered On: Actively retrieve all inputs, display calendar on monitor, and decide if to send alerts

For implementing all designated Smart Calendar features
- Shutting Down: Send stored tracking data to the internet, display is off, and all inputs are not being read

For stopping all features and going to Low Power mode from Powered On mode, device could automatically enter Shutting Down from Powered On based on default criteria like a given time of day

Subsystems:

- **Raspberry Pi 3, Power Supply, Ethernet, and Digital I/O Pins**

We will be using a Raspberry Pi 3 to interface with all sensors and devices as well as to communicate with the internet using its onboard ethernet port.

Inside the System we will also need to have the Raspberry Pi 3 power supply to power the board. This will connect the conventional 120V power input to the power inlet on the board.

The wires coming in 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 ethernet cable connected to the Internet will be directly connected to the board's ethernet.

USB connections will be made to the board's various USB ports. The monitor and keyboard will connect using these.

Inside the Raspberry Pi 3:

- **GUI Process**

This process will handle the graphics going to and from the in/out connection touch display monitor connected via USB. 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 the keyboard input. The messages will then be stored in memory. There will also be interactivity for people to interact with 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**

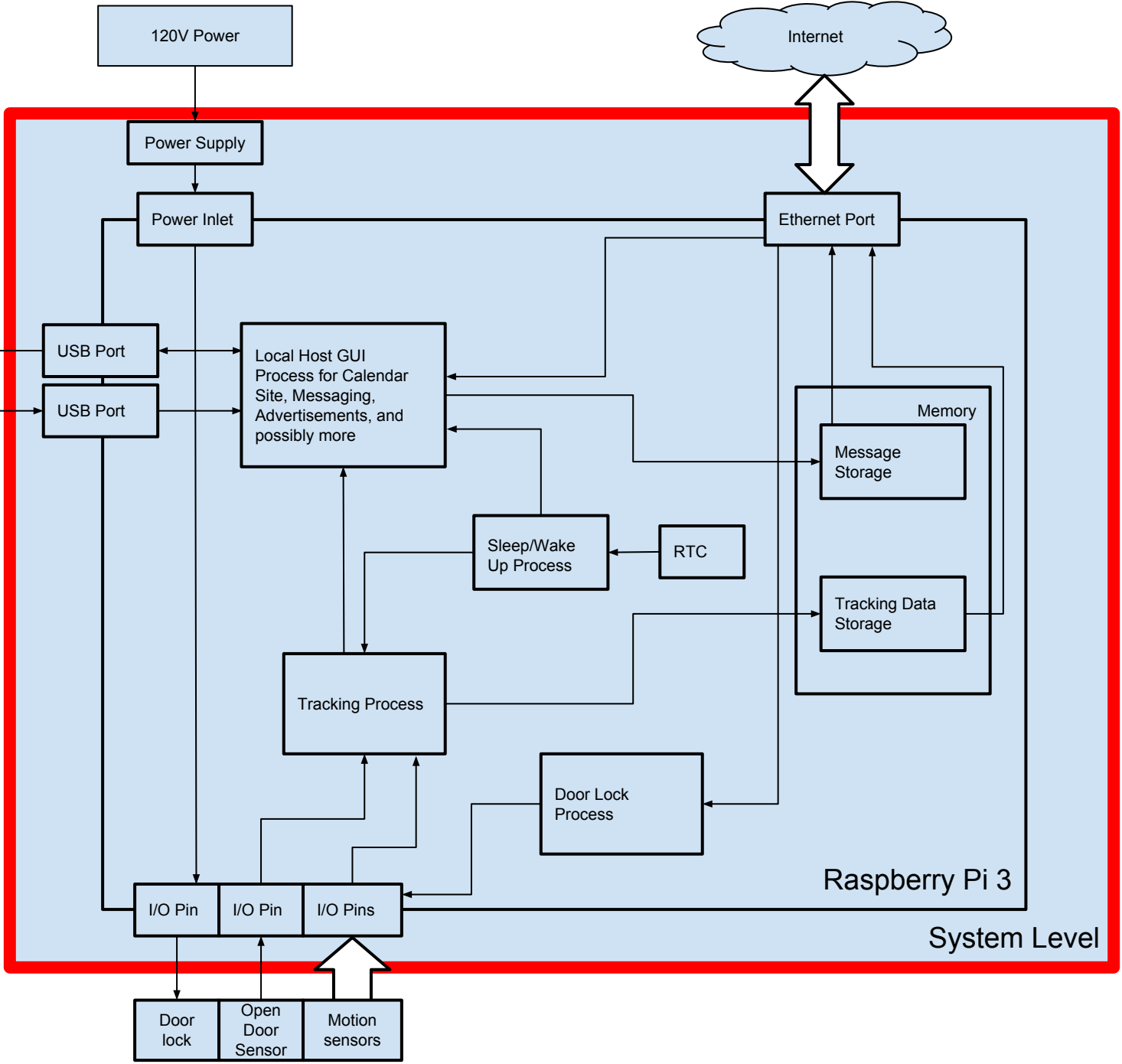
A portion of read/write memory will be allocated on the Raspberry Pi 3's SD card storage for temporary data. This data will be sent to the Internet as a data file and uploaded to Google Drive if it is tracking data or E-mailed to Doctor Malinowski if it is a message. If the data is successfully sent, then it will be cleared from memory.

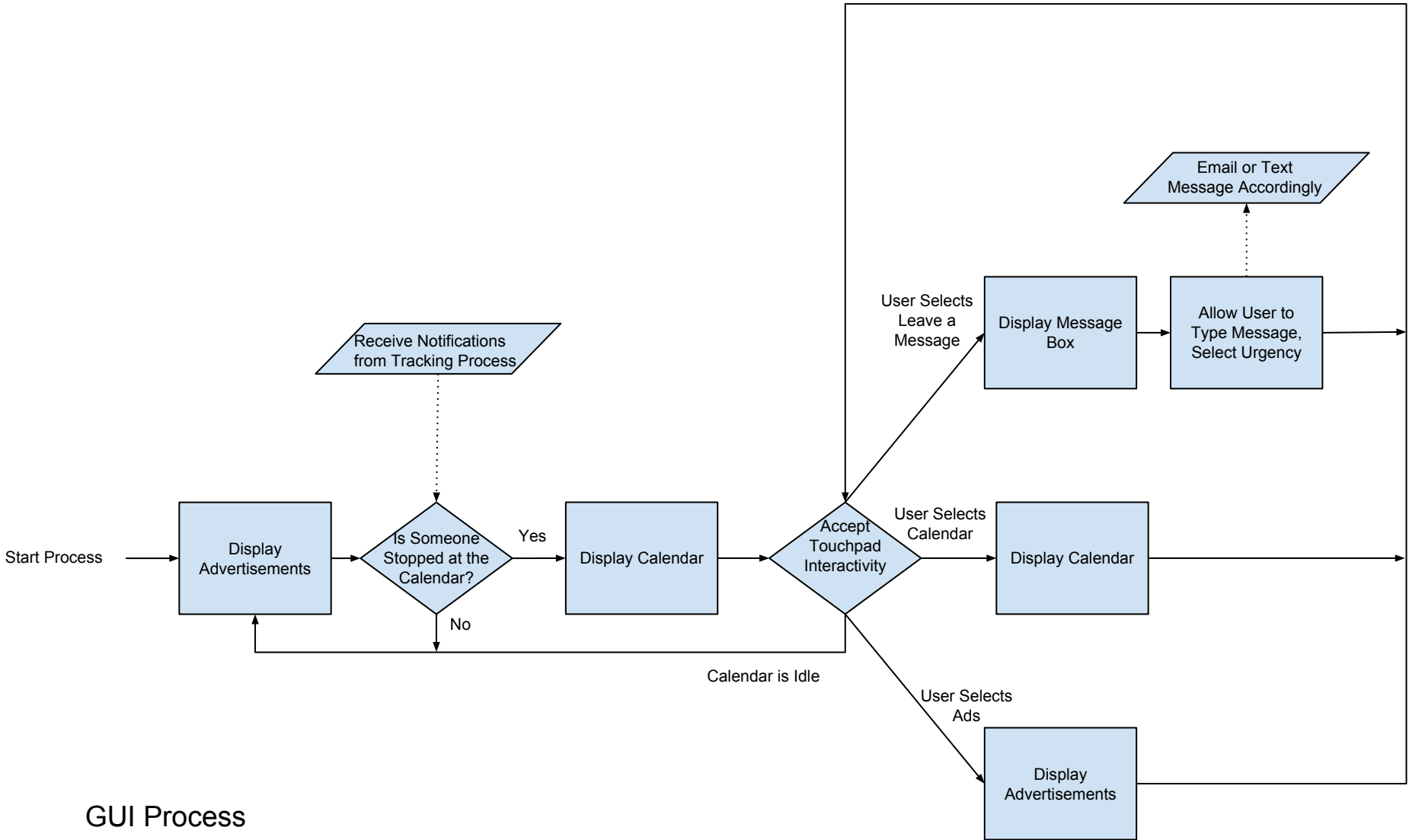
- **Sleep/Wake Up Process**

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 necessary processes and power the digital I/O pins to power the sensors.

- **Door Lock Process**

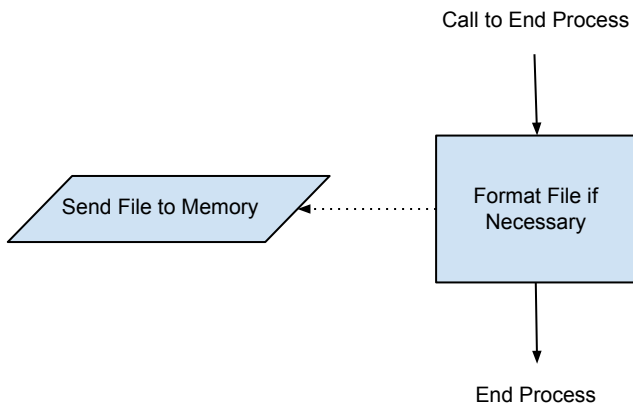
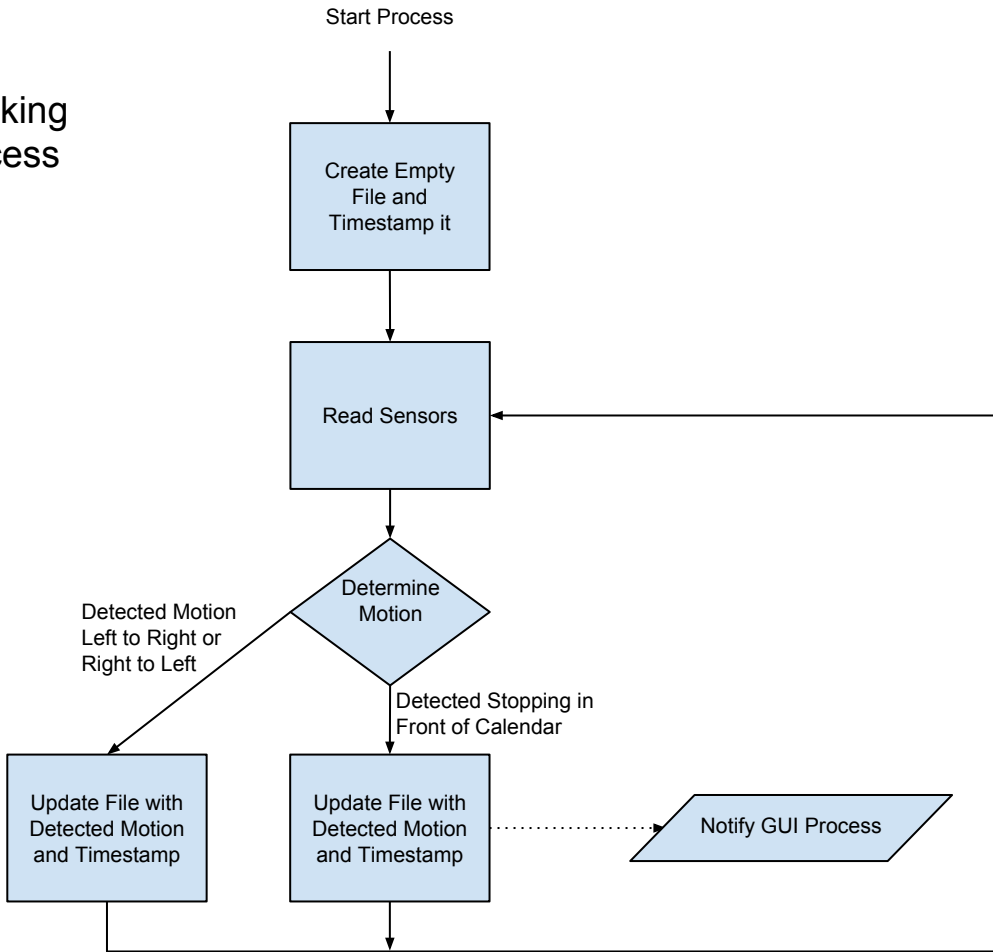
This process will accept a secure message from the internet and then use the digital I/O pin to communicate with the electronic door lock to unlock the door.





GUI Process

Tracking Process



Memory Process

