

# IoT Smart Calendar ☺

Jason Morris, Cole Lindeman

Advisor: Dr. Malinowski

## Introduction

**Designed to display:**

- Professor's calendar
- Advertisements
- Announcements

**What it can do:**

- Touch interactivity
- Leave messages
- Track motion
- Automatic Internet downloads and uploads
- Save power

## The Blueprints

**Project is powered by:**

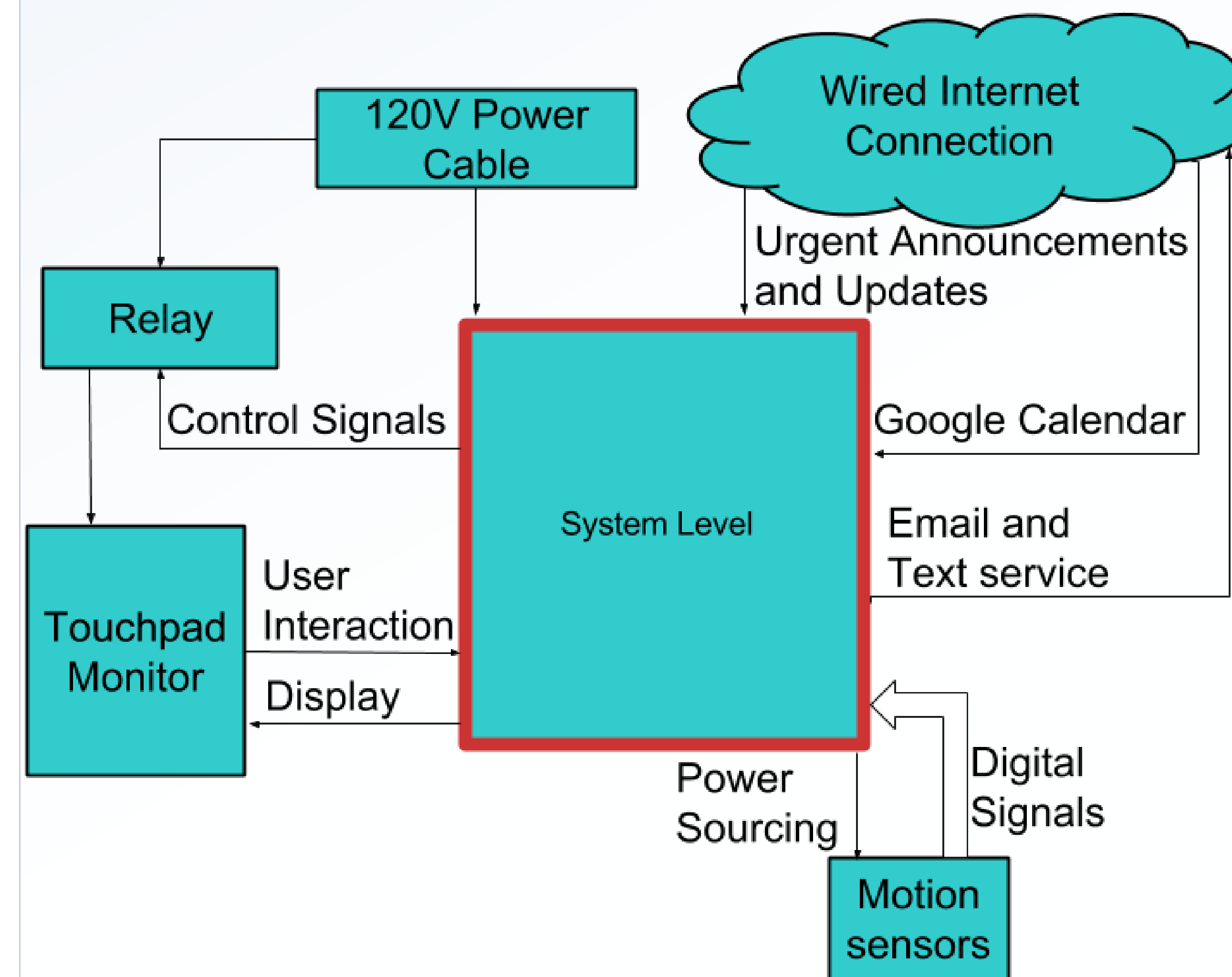
**Hardware**

- Raspberry Pi 3 Model B
- Interactive LCD Touchscreen Monitor
- Passive Infrared Motion Sensors
- USB Relay Module

**Software**

- Ubuntu MATE Operating System
- Apache 2 Web Server
- HTML, CSS, JavaScript GUI
- PHP Mail Service
- Python Background Service
- Git Source Control and Updating
- Firefox Web Browser

## System Level Diagram



## Git Updates

**How it works:**

1. Project Repository uploaded on GitHub
2. Changes pushed to repository
3. Raspberry Pi runs periodic Git pulls

**What is updated:**

- Advertisement Pictures
- Class Announcements
- Graphical Interface Improvements

## Infrared Tracking

1. Raspberry Pi reads sensors through GPIO
2. Detected motion is tracked and stored
3. Formats text file about activity
4. Tells other processes about activity
5. Uploads text file at the end of the day

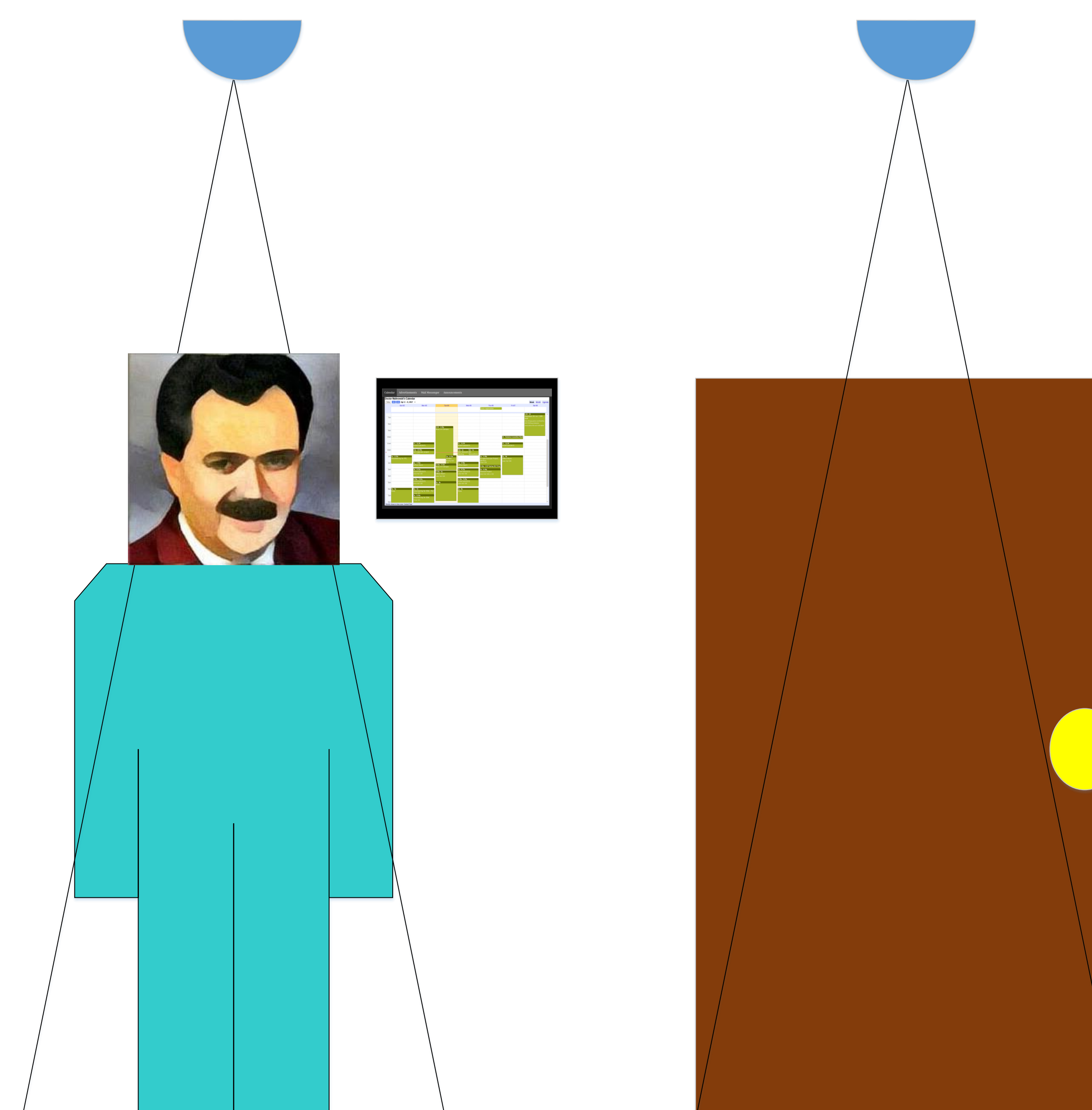
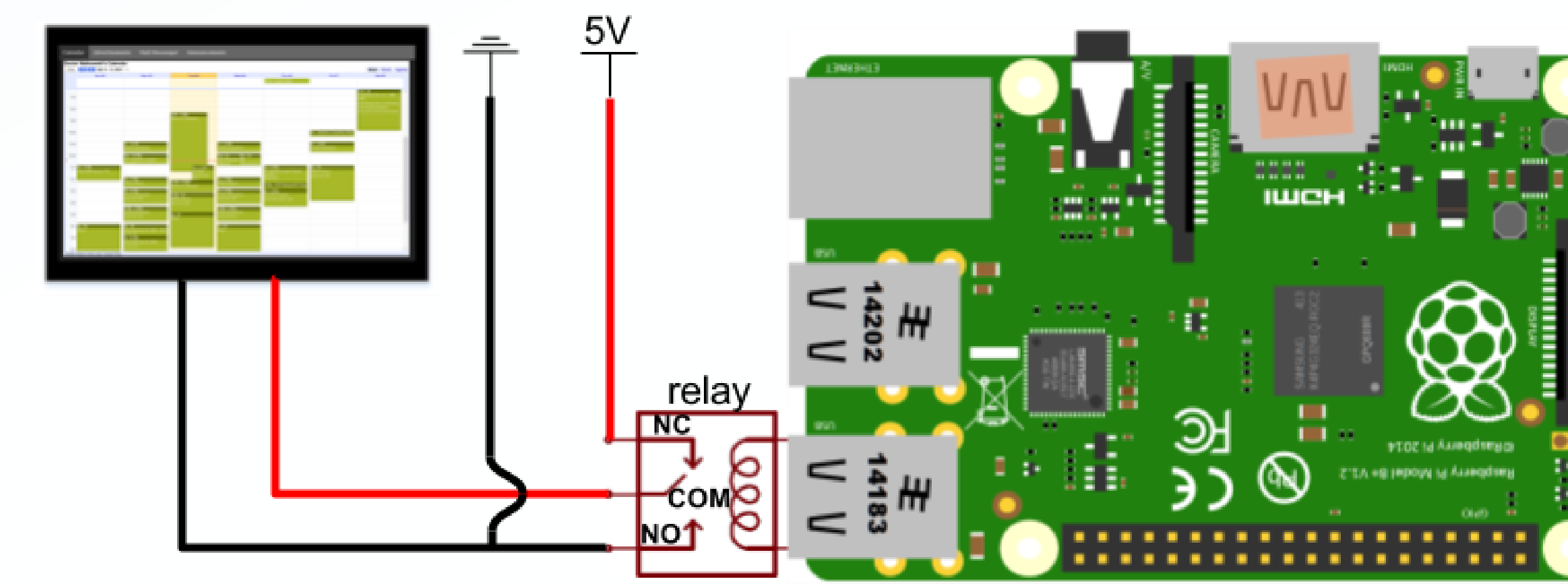
## Monitor Power Control

**What it does:**

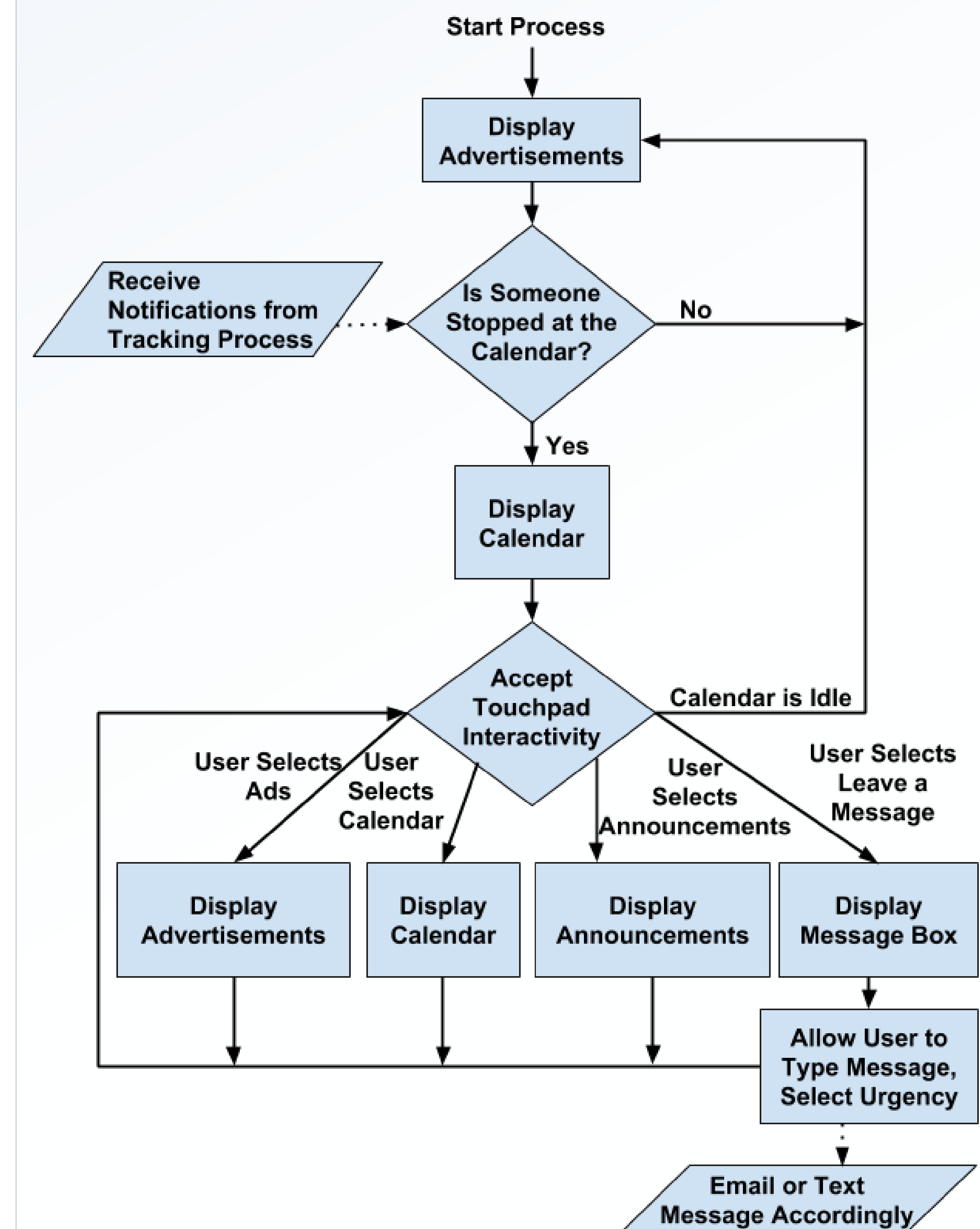
Gives us the capability to turn the monitor on and off

**How it works:**

1. Serial signal sent to relay to control switch
2. Relay switches to VCC or ground
3. The monitor is now powered on or off



## GUI Process



## Messenger

**How it works:**

1. HTML page allows users to type message
2. User selects sending type (Regular/Urgent)
  - a. Regular messages are email messages
  - b. Urgent messages are text messages
3. User presses send button
2. HTML page sends message to PHP Script
3. PHP Script sends message to Dr. Malinowski
4. User is notified that message has been sent