Stove Reminder Device

Zulmei Eshaqzaie

Abstract

A simple mistake such as leaving on an electric stove can have serious consequences such as fires, high energy consumption, and even burning down the house. Our Stove Reminder Device is designed to address this concern by keeping tabs on environmental conditions around a stove and alerting the user when there is a potentially dangerous situation. The system combines both a temperature sensor and a light sensor to detect a situation in which the stove is on while the kitchen lights are off, which is a indication that the user left the area. When the system detects unsafe conditions, it sets off a loud buzzer and provides feedback using an RGB LED. An extension feature that expands on the system's usefulness is notifying the user of the situation remotely. This project shows how simple sensors and software on a Raspberry Pi can create a practical safety device.

Introduction

People use electric stoves all the time in their homes, however they are a safety risk when left on unintentionally. Very often, people turn on a stove, leave the kitchen briefly, then forget that their stove is still active. This risk is especially dangerous at night or when then kitchen lights are off, since there are fewer visual cues that the stove is still on and generating heat.

The motivation for this this project came from realizing how easily this situation can occur and how dangerous the repercussions can be. Normally, smoke alarms only go off when smoke or fire is already present, which often times is way too late. Our goal was to build a system that can identify an unsafe condition before it escalates into a potentially deadly one.

The main goals of our Stove Reminder Device are to:

- Detect when is a stove is hot by using a temperature sensor.
- Determine whether the environment suggests a user is present by using a light sensor.

- To alert the user locally using a buzzer and LED.
- To implement a remote alerting feature.

System Design and Implementation

Hardware Components

The Stove Reminder Device is built around a Raspberry Pi as the main processing unit. An Arduino Uno was also used to handle analog sensors and data acquisition. We used these hardware components:

- **Raspberry Pi:** The main controller, ran the high level control logic, coordinated inputs and outputs, handled alerts, used its GPIO pins to interface the sensors.
- **Arduino Uno:** Used to handle the analog sensors and low level sensor readings.
- **TMP36 Temperature Sensor:** Measures the temperature of the stove. Stove is determined to be on or hot if the temperature goes beyond a set threshold.
- **LDR Light Sensor:** Detects light levels in the kitchen, low light readings below the set threshold determines if lights are on or off.
- **RGB LED:** Gives visual feedback about the state of the system. Red, yellow, and green colors represent safe, caution, and unsafe.
- **Passive buzzer:** Makes a loud buzzing sound when the conditions of what is defined as unsafe is met.
- **Resistors and Breadboard:** Used to safely connect all the hardwired components together.

Wiring Overview

The Raspberry Pi and Arduino Uno share the responsibility of the sensors and outputs. The Arduino's role was to handle the direct connections to the analog sensors while the Raspberry Pi's role was to handle decision making and control output. The Pi's 3.3V pin powers the temperature sensor and its output is converted from analog to digital. The light sensor is wired so that the changes in light intensity can be seen in measurable voltage changes. The RGB LED is connected to GPIO pins via resistors. The buzzer is also connect to a GPIO pin to generate sound.

Software Design

The software for the device is split between two different platforms. The Arduino code was ran on Arduino IDE to read sensor data. The Raspberry Pi uses Python software to process inputs and control outputs. The program can be organized into different sections.

- **Initialization:** Configuring the GPIO pins, setting thresholds for the temperature and light sensors.

- **Reading Sensors:** Periodically reading values from the temperature and light sensors and converting them into units such as Fahrenheit.

- **Decision Making:** Comparing the sensor readings to the set thresholds. A temperature is considered unsafe if it is above its set threshold and the light is considered to be off if it is below its set threshold.

- **Output Control:** Activating the buzzer and setting the color of the RGB LED based on the state of the system.

Testing and Results

Methodology

These are the conditions that must be met for each level of safety.

- **Normal/Safe:** Temperature readings are below temperature threshold and light readings are above light threshold, indicating both room temperature and normal lighting. Green LED indicates a safe state and there is no buzzer activity.

- **Low Light, No Heat:** Both temperature and light readings are below their respective thresholds, indicating kitchen lights are off but there's no active heat source.

- **High Light, High Temperature:** Both temperature and light readings are above their respective thresholds, indicating kitchen lights are on and there's an active heat source.

- **Low Light, High Temperature (Unsafe):** Temperature readings are above threshold and light readings are below threshold, indicating the stove is producing heat and and the lights are off. The buzzer activates the RGB LED turns red.

Results

Overall our system worked as intended and was reliable at detecting unsafe conditions. Using both the temperature and light sensors together showed to be an effective pairing. However, there was still a limitation. We were not able to properly configure the remote alert feature.

Challenges

One of the challenges we faced during this project was trying to balance making the system reliable and making it sensitive. Setting the thresholds too low lead to the system being too easy to activate which would lead to false alarms in practicality. Setting them too high made it hard to activate, which practically could lead to missing unsafe conditions.

The biggest challenge we faced was trying to implement the remote alert feature. We attempted to set up a feature where a user can enter their phone number and their carrier so they can have emails sent to their SMS. However, we failed in being able to do so.

Future Work

If more time was available, we could have made multiple improvements such as:

- Incorporating more sensors such as a motion sensor to detect user presence.

- Get the remote alert system to function properly.

- Design an enclosure for the system so it can be safer and more aesthetic for practical use in a kitchen.

References

- Raspberry Pi GPIO Documentation
- TMP36 Temperature Sensor Datasheet
- Python GPIO and PWM Tutorials