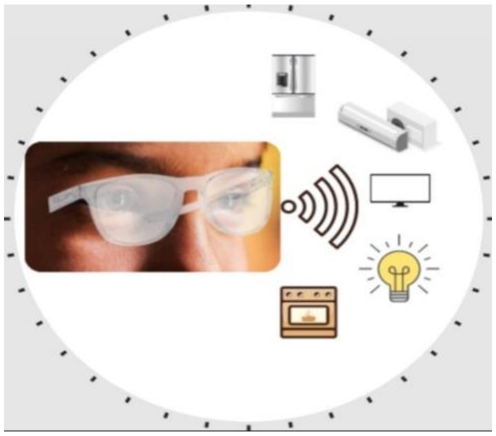
PROJECT

Abstract

We propose a Eye-Blink Controlled Smart Home Automation system that leverages an ESP8266 microcontroller and an eye-blink sensor to enable hands-free control of smart home devices. By detecting blinks and interpreting their duration, users can trigger actions like turning on lights, micrwovens or even controlling entertainment systems with a simple blink. This approach offers potential advantages such as increased accessibility for users with limited mobility, a more intuitive control method compared to traditional remotes or voice assistants, and potentially improved privacy compared to voice-activated systems. We are exploring two potential control methods: direct IR control for basic devices and Wi-Fi communication for a wider range of smart home systems.

Introduction

The concept of smart homes has become increasingly popular, offering the ability to control various aspects of our living environment through automation and remote access. However, current control methods often rely on physical remotes, voice assistants, or smartphone applications. These methods may not be ideal for everyone, particularly users with limited mobility or those seeking a more intuitive and hands-free control experience "Eye-Blink Controlled Smart Home Automation" an approach that utilizes eye blinks as the primary control method. By leveraging an ESP8266 microcontroller and an eye-blink sensor, the system aims to provide a user-friendly and accessible way to interact with smart home devices. The system detects blinks and interprets their duration, allowing users to trigger specific actions through simple blinks.

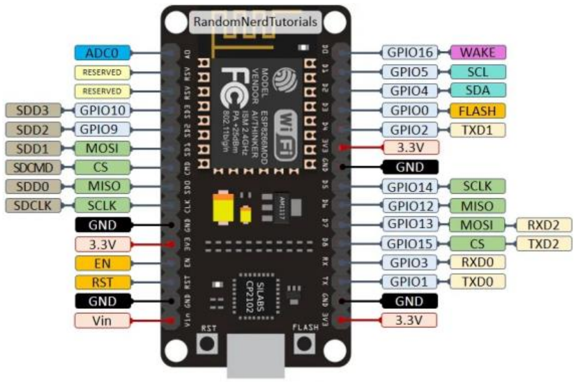


Project Requirements

Hardware Requirements

ESP8266 wifi module :

This is a microcontroller board that will be programmed to process the eye blink signals and communicate with the Python program.

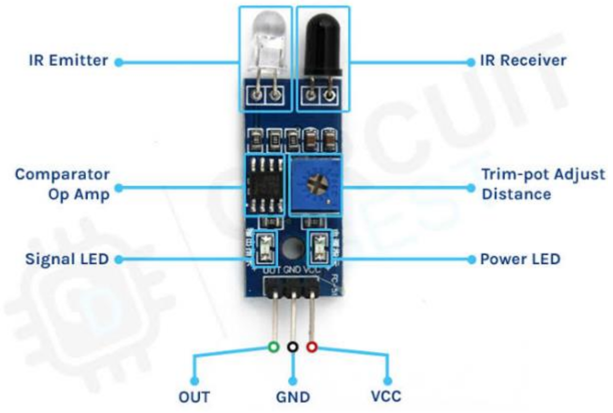


Eye Blink Sensor :

This sensor, typically detects changes in light reflecting off eye when blink which is connected to the Nodemcu.This sensor emit and detect infrared radiation, which is invisible to the human eye. When eye closes during a blink, it reflects less infrared light back to the sensor. This change in reflection is detected by the sensor and interpreted as a blink.

<u>**IR Sensor**</u> :

This sensor emits the Infrared rays mimicking the remote control and triggering the desired action on the appliance.



<u>**Camera Module**</u> :

The camera captures a visual image of the scene infront, the captured image is used for object detection based on image features using the trained machine learning model

<u>**Software Requirements**</u>

<u>**Arduino IDE**</u> :

It is a powerful tool for programming Arduino microcontrollers, It communicates with the Arduino board through a serial connection, It provides vast number of libraries for our project might include IRremote for controlling IR appliances.

Python IDE :

It is needed to identify the object, Necessary libraries like serial and opencv-python are needed for communication and image processing.

Project Code

Arduino Code

```
#include <IRremote.h>

#include <dht.h>


Const int eyeBlinkPin = A0;

Const int blinkThreshold = 500;

Const int minBlinkDuration = 100;

Const int maxBlinkDuration = 500;

Const int serialBaudRate = 9600;


Int previousEyeBlinkValue = 0;

Unsigned long lastBlinkTime = 0;

Int continuousBlinks = 0;


Const int irEmitterPin = 8;


IRsend irsend(irEmitterPin);


Void setup() {

  Serial.begin(serialBaudRate);
  pinMode(irEmitterPin, OUTPUT);
  irsend.enableIR();

}

Void loop() {
  Int eyeBlinkValue = analogRead(eyeBlinkPin);

  Int smoothedValue = (previousEyeBlinkValue + eyeBlinkValue) / 2;
  previousEyeBlinkValue = eyeBlinkValue;
  if (abs(smoothedValue – previousEyeBlinkValue) > 100) {
    unsigned long currentTime = millis();
    if (currentTime – lastBlinkTime > 500) {
      if (smoothedValue < blinkThreshold) {
        lastBlinkTime = currentTime;
      } else if (smoothedValue > blinkThreshold && currentTime – lastBlinkTime >
minBlinkDuration) {
        Int blinkDuration = currentTime – lastBlinkTime;
        If (blinkDuration >= minBlinkDuration && blinkDuration <= maxBlinkDuration) {
          continuousBlinks++;

          if (continuousBlinks > 10) {
            Serial.println("Camera Activated!");
            continuousBlinks = 0;
          }
        } else {
```

```
        continuousBlinks = 0;
      }
    }
  }
}
If (Serial.available() > 0) {
  String objectName = Serial.readStringUntil('\n');
  objectName.trim();
  if (objectName == "microwave") {
    if (continuousBlinks % 2 == 1) {
      irsend.sendSamsung(0xABCD);
      delay(1000);
      Serial.println("Microwave turned on!");
    } else if (continuousBlinks % 2 == 0) {
      Irsend.sendSamsung(0xDCBA);
      Delay(1000);
      Serial.println("Microwave turned off!");
    }
    continuousBlinks = 0;
  } else if (objectName == "television") {
    If (continuousBlinks % 2 == 1) {
      Irsend.sendSamsung(0xA847);
      Delay(1000);
      Serial.println("Television turned on!");
    } else if (continuousBlinks % 2 == 0) {
      Irsend.sendSamsung(0XCODE);
```

```
        Delay(1000);
        Serial.println("Television turned off!");
      }
      continuousBlinks = 0;
    }
  }
}
```

Python Code

```
Import serial
import cv2
Object_labels = {
  "light": (180, 200, 150, 220, 100, 150),
  "television": (100, 150, 50, 100, 150, 200)
}
Arduino = serial.Serial('COM3', 9600)  # Replace with your port and baud rate
While True:
 Data = arduino.readline().decode().strip()
 If data == "Camera activated!":
  Print("Camera activated by Arduino.")
  Image_data = arduino.read(image_size)
   Image_bytes = bytearray(image_data)
   Image = cv2.imdecode(np.frombuffer(image_bytes, np.uint8), cv2.IMREAD_COLOR)
  Average_color = cv2.mean(image)
```

```
   for object_name, color_range in object_labels.items():
    If (color_range[0] <= average_color[0] <= color_range[1] and
      Color_range[2] <= average_color[1] <= color_range[3] and
      Color_range[4] <= average_color[2] <= color_range[5]):
     Control_action(object_name)
     arduino.write(object_name.encode())
     break
else:
  Pass
```

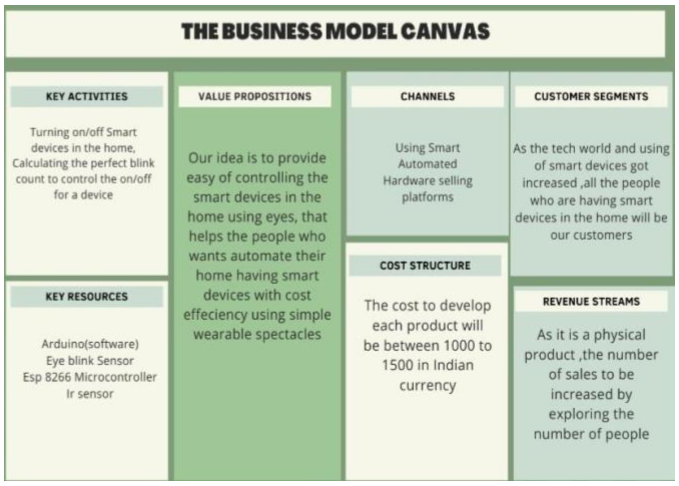
<u>Arduino code description</u> :

The Arduino code focuses on detecting eye blinks using an analog sensor connected to pin A0. It implements a mechanism to avoid double-counting blinks due to sensor . It counts consecutive blinks and resets the counter if the blink duration falls outside a specific range. Once a threshold number of continuous blinks is reached, it sends signals to the Python program via serial communication that it's ready to receive an object name, After successful identification of object by the python program, Arduino code will capture the object name in the serial monitor, Based on the received object name the Arduino will activate the IR sensor to change the status of the object(light, television, microwoven and other).

<u>Python Code description</u> :

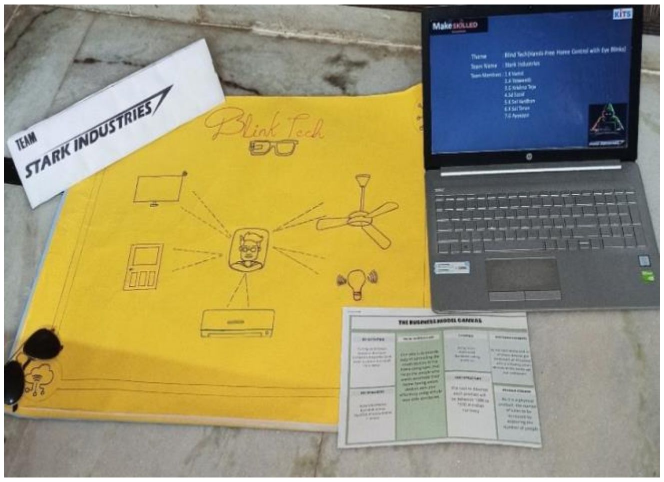
The Python code utilizes OpenCV for object detection. It establishes serial communication with the Arduino. Upon receiving a signal indicating camera activation, it retrieves image data from the Arduino. It converts the byte array to an OpenCV image and calculates the average color. By comparing the average color to predefined ranges for objects like "light" and "television," it attempts to identify the object in view. If a match is found, it triggers a message and sends the object name back to the Arduino.

Business Model Canvas



THE BUSINESS MODEL CANVAS

KEY ACTIVITIES	VALUE PROPOSITIONS	CHANNELS	CUSTOMER SEGMENTS
Turning on/off Smart devices in the home, Calculating the perfect blink count to control the on/off for a device	Our idea is to provide easy of controlling the smart devices in the home using eyes, that helps the people who wants automate their home having smart devices with cost effeciency using simple wearable spectacles	Using Smart Automated Hardware selling platforms	As the tech world and using of smart devices got increased ,all the people who are having smart devices in the home will be our customers
KEY RESOURCES		**COST STRUCTURE**	**REVENUE STREAMS**
Arduino(software) Eye blink Sensor Esp 8266 Microcontroller Ir sensor		The cost to develop each product will be between 1000 to 1500 in Indian currency	As it is a physical product ,the number of sales to be increased by exploring the number of people



Conclusion

In Conclusion, Our project uses ESP8266 microcontroller and eye blink sensor for hands-free interaction with smart home devices. We are successfully integrates an eye blink sensor with Arduino to detect blinks and trigger the actions, we developed an Arduino program to detect and count consecutive eye blinks and established serial communication between Arduino and Python for data exchange. Our project provides a valuable tool for people with limited mobility or mute by offering hands-free control over smart home devices and also serves as a foundation for more eye-blink interaction methods for home automation products.