Global Journal of Engineering and Technology Advances
eISSN: 2582-5003
Cross Ref DOI: 10.30574/gjeta
Journal homepage: https://gjeta.com/

(RESEARCH ARTICLE)

Check for updates

Design and development of a system for controlling the electrical loads of a residence thru eye blinks: Case of assistance provided to people with reduced mobility

Narcisse Meni Babakidi [1], Christian Vunda Ngulumingi [1], Jordan Kondatata Mbambu [2,*], Marcien Tangenyi Okito [3] and God'El Kinyoka Kabalumuna [4]

[1] *Laboratory of Electrical and Electronic Engineering, ISTA-Kinshasa, Kinshasa, DR Congo.*
[2] *Laboratory of Electrical and Electronic Engineering, ISTA-Kasangulu, Kongo-Central, DR Congo.*
[3] *Laboratory of Electrical and Electronic Engineering, ISTA-Gombe-Matadi, Kongo-Central, DR Congo.*
[4] *National Pedagogical University, Faculty of Sciences, Department of Physics and Applied Sciences, Kinshasa, DR Congo*

Global Journal of Engineering and Technology Advances, 2026, 26(02), 029-038

Publication history: Received on 20 December 2025; revised on 28 January 2026; accepted on 31 January 2026

Article DOI: https://doi.org/10.30574/gjeta.2026.26.2.0033

Abstract

This work is dedicated to the design and development of a device for controlling electrical loads by blinking. It was primarily intended for people with reduced mobility, this system offers a natural and non-intrusive interface to control lighting, household appliances, and other electrical devices in a home. The device is based on an infrared camera coupled with a real-time image processing algorithm capable of identifying eye blinks and translating them into commands executed by a microcontroller. This work has thus made a modest but significant contribution to the issue of assistance for people with reduced mobility. By leveraging open-source tools and locally available components, we have shown that it is possible to design a simple, intuitive technological solution suitable for daily use. The tests conducted under practical conditions reveal the reliability, responsiveness, and ease of use of the system. This article represents a significant technological breakthrough in the home care sector for individuals with mobility difficulties.

Keywords: Design; Development; Bias; Eye Blink; People with Reduced Mobility

1. Introduction

In a world where technology is evolving at a dizzying pace, where cars drive themselves, and where connected objects are invading our homes, it is paradoxical to note that millions of people with severe motor disabilities still struggle to perform what are otherwise simple tasks. Turning on a light, adjusting the temperature of a room, or even opening a door can be an insurmountable challenge for those who do not have the use of their hands or limbs. Yet, the human body holds often underutilized capabilities...

Indeed, imagine for a moment a quadriplegic person turning on their lamp with a series of eye blinks, or adjusting their wheelchair without having to call for help. This is not science fiction, it is a reality within reach. With affordable electronic components, clever programming, and a good dose of ingenuity, it is possible to create a system that works wirelessly, effortlessly, and without frustration.

The idea is not new, but most existing solutions are either too expensive, too complex, or simply unsuitable for daily use. We are going to change that, develop a reliable, intuitive, and accessible device that empowers those whom life has deprived of mobility.

* Corresponding author: Kondatata Mbambu Jordan.

Faced with the aforementioned problems, the question is: how to design and develop a system that allows people with reduced mobility to control various electrical equipment with just the movement of their eyes, in order to facilitate their interactions with their environment and increase their independence?

Based on the above, we postulate that an electrical control device using eye blinks, designed with accessible components and an intuitive interface, can offer a viable solution to improve the autonomy of people with reduced mobility. By utilizing inexpensive sensors (such as electrodes or miniature cameras) and a lightweight processing system (microcontroller, simple algorithmic logic), it would be possible to create a reliable, ergonomic, and financially affordable tool. The objective pursued in this article is to propose an adorable technological solution specially adapted to the needs of people with severe motor disabilities.

2. Materials, software, and methodology

2.1. Methodology

The work will implement a varied and exhaustive approach, integrating the following essential steps:

2.2. Study of user requirements and constraints

A functional specification document was drafted in collaboration with experts in occupational therapy and end-users. It specifies the requirement for intuitive handling, low command latency, and compatibility with various loads (LED lighting, motor, electrical outlet).

2.3. Choice of detection technologies

We opted for an infrared camera due to its performance in various lighting conditions, while remaining untrained. Computer vision algorithms based on edge detection and motion analysis methods were developed in Python, notably thru the use of the OpenCV library. A temporal filter analyzes the detected blink to remove false positives caused by involuntary movements.

2.4. Development of the control module

A microcontroller (Arduino board) manages the electrical commands using appropriate solid-state relays for domestic loads. The communication interface between the detection device and the Arduino board is established via UART to ensure optimal synchronization and robustness.

Figure 1 represents the synoptic diagram that illustrates the operating principle of our system.

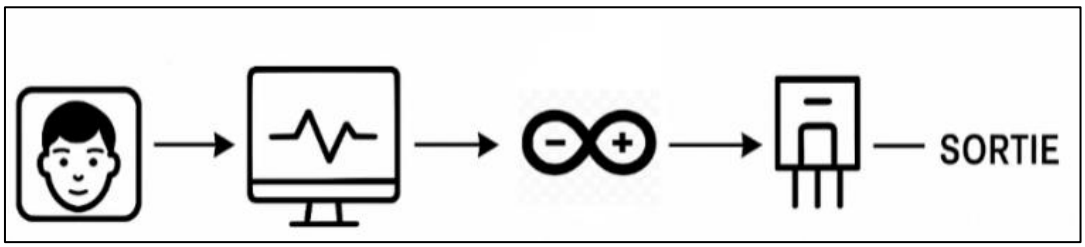

Figure 1 Synoptic diagram

In what follows (Figure 2), we present the wiring diagram of our system under Proteus ISIS.

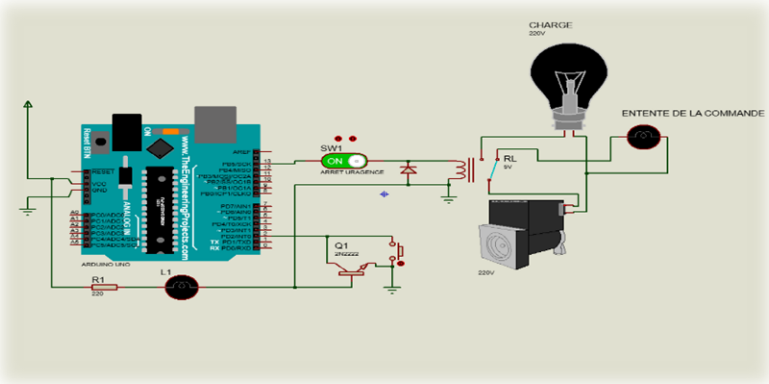

Figure 2 Electronic diagram

2.5. Materials

2.6. Webcam

The first element, and not the least, is the vision sensor: a standard webcam with a minimum resolution of 720p, as illustrated in figure 3.

Figure 3 Webcam Camera [3]

The advantage of this choice is that it is affordable and easily available equipment, which can be replaced or upgraded without difficulty according to the user's needs.

2.7. Computer

It is the one that hosts the Python program responsible for capturing the video stream, analyzing the images in real-time, and detecting eye blinks using the OpenCV, Dlib, or MediaPipe libraries [2]. The computer is connected to the webcam via a USB port for video capture, and to the Arduino for transmitting control commands for electrical loads, figure 4.

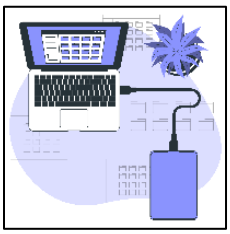

Figure 4 Arduino-PC connection

2.8. Microcontroller

The heart of the control system is entrusted to an Arduino Uno board, based on the ATmega328P microcontroller, as shown in Figure 5 below:

Figure 5 Arduino Uno board [5]

This board acts as an intelligent bridge between the detection software and the physical world. It receives orders via a USB connection and acts concretely on the electrical circuit thru its digital input/output pins.

In the context of this project, the Arduino is configured to attentively listen to the signals sent by the Python script. As soon as a valid order is received (a voluntary blink correctly detected), it triggers the relay to power on or cut off the electrical load, instantly and securely.

2.9. Opto-coupler relay

The opto-coupler relay is a gateway: when the Arduino sends an electrical signal, the relay "clicks" mechanically to open or close the power circuit [9].

Figure 6 below represents the opto-coupler relay.

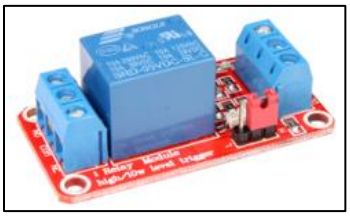

Figure 6 Opto-coupler relay [9]

In this project, it is thanks to the relay that we can turn on or off a lamp, a fan, a power outlet, without ever manually manipulating the circuit. This contactless control is at the heart of the idea: replacing a classic switch with a signal triggered by an eye blink.

2.10. Push button

This is an essential precaution to avoid any accidental triggering, especially in a context of use by a vulnerable person, figure 7.

Figure 7 BPNC

2.11. Software

2.11.1. Arduino ide

The Arduino IDE is an open-source software used to write and upload code to Arduino boards. The IDE application is compatible with various operating systems, including Windows, Mac OS X, and Linux. This software supports the C and C++ programming languages. Due to its comprehensibility, we integrated it into our project for coding purposes [5].

2.11.2. PYCHARM

To develop, test, and maintain the Python script, the PyCharm environment was chosen. PyCharm is an IDE (Integrated Development Environment) specifically designed for Python [8], developed by JetBrains.

Code Arduino

```
const int LED_PIN = 13;

const int RELAY_PIN = 12;

const int BUTTON_PIN = 2;

bool systemActive = true;

void setup()

pinMode(LED_PIN, OUTPUT);

pinMode(RELAY_PIN, OUTPUT);

pinMode(BUTTON_PIN, INPUT_PULLUP);

Serial.begin(9600);

while (!Serial);

Serial.println("ARDUINO READY");

}

void loop()

{

if (digitalRead(BUTTON_PIN) == LOW) {

systemActive = !systemActive;

digitalWrite(LED_PIN, LOW);

digitalWrite(RELAY_PIN, LOW);

Serial.print("SYSTEM:");

Serial.println(systemActive ? "ON" : "OFF");

delay(300);

}

if (systemActive && Serial.available() > 0) {
```

```
char command = Serial.read();
if (command == '1') {
digitalWrite(LED_PIN, !digitalRead(LED_PIN));
digitalWrite(RELAY_PIN, !digitalRead(RELAY_PIN));
Serial.print("STATE:");
Serial.println(digitalRead(LED_PIN) ? "ON" : "OFF");
} }}
```

Code Python

```
import cv2
import serial
import time
from serial.tools import list_ports
class EffectiveBlinkDetector:
def __init__(self):
# Configuration caméra
self.cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
self.cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
self.cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
if not self.cap.isOpened():
print("Erreur : caméra non détectée")
exit()
# Modèles de détection
self.face_cascade = cv2.CascadeClassifier(
cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
self.eye_cascade = cv2.CascadeClassifier(
cv2.data.haarcascades + 'haarcascade_eye.xml')
# Connexion Arduino
self.arduino = self.connect_arduino()
self.blink_count = 0
```

```
self.eyes_detected = False
self.last_blink_time = time.time()
def connect_arduino(self):
"""Connexion série automatique"""
ports = list_ports.comports()
for port in ports:
try:
arduino = serial.Serial(port.device, 9600, timeout=1)
time.sleep(2)
print(f"Connecté à Arduino sur {port.device}")
return arduino
except:
continue
print("Arduino non détecté")
return None
def detect_blink(self, frame):
"""Détection basique de clignement"""
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
faces = self.face_cascade.detectMultiScale(gray, 1.3, 5)
current_eyes_detected = False
for (x, y, w, h) in faces:
roi_gray = gray[y:y + h, x:x + w]
eyes = self.eye_cascade.detectMultiScale(roi_gray, 1.1, 3)
if len(eyes) >= 2:
current_eyes_detected = True
for (ex, ey, ew, eh) in eyes[:2]:
cv2.rectangle(frame, (x + ex, y + ey),
(x + ex + ew, y + ey + eh), (0, 255, 0), 2)
if self.eyes_detected and not current_eyes_detected:
```

```
if time.time() - self.last_blink_time > 0.4:
self.blink_count += 1
self.last_blink_time = time.time()
print(f"Clignement détecté ! ({self.blink_count}/3)")
if self.blink_count >= 3 and self.arduino:
self.arduino.write(b'1')
print("Commande envoyée à Arduino")
self.blink_count = 0
self.eyes_detected = current_eyes_detected
return frame def run(self):
"""Boucle principale"""
try:
while True:
ret, frame = self.cap.read()
if not ret:
print("Erreur : impossible de lire la caméra")
break
frame = self.detect_blink(frame)
cv2.putText(frame, f"Clignements: {self.blink_count}/3",
(10, 30), cv2.FONT_HERSHEY_SIMPLEX,
0.7, (0, 0, 255), 2)
cv2.imshow("Détection des Clignements", frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
print("Fermeture demandée par l'utilisateur")
break
finally: self.cap.release()
cv2.destroyAllWindows()
if self.arduino: self.arduino.close()
print("Connexion Arduino fermée")
```

```
if __name__ == "__main__":

detector = EffectiveBlinkDetector()

detector.run()
```

3. results

The system achieves a blink detection rate of 98.5% and presents an average response time of less than 300 ms, thus ensuring adequate reactive performance. The algorithmic filter helped reduce false triggers to less than 2%. The UART interface has proven to ensure reliable transmission without any data loss. Users approved of the ease of use and found the device comfortable even during extended sessions. The incorporation into the prototype also proved compatibility with various currents (up to 10A) without any alteration in performance.

Figure 8 below shows the result and the test of our electrical load control system by blinking before and after blink detection.

Figure 8 Prototype not activated (No detection)

The interface plays an essential role in the functioning of the prototype. It allows the user to easily interact with the system without needing to directly access the source code.

Indeed, allowing the code to be visible or manipulable by the user could not only create confusion but also lead to handling errors or program malfunctions, figure 9.

Figure 9 Interface Framework PyQt5

4. Discussion

The results attest that using eye blinking as a means of control constitutes a reliable option for classic home control interfaces, intended for individuals with mobility limitations. Although the performance is encouraging, some constraints persist: the requirement for precise camera positioning relative to the user's face and the responsiveness to exceptional extreme lighting conditions. Future improvements aim to integrate automatic position correction systems

as well as the use of deep learning algorithms to enhance robustness. Moreover, the widespread adoption of this type of equipment raises concerns about privacy and security, particularly regarding the capture and processing of facial images.

5. Conclusion

This chapter has provided a detailed presentation of the design and implementation of the electrical load control device using eye blinks. The design phase was structured around two main axes: hardware design, including the selection and wiring of electronic components, and software design, which includes the development of the blink detection module, simulation, and the creation of the user interface.

The implementation of the interface with PyQt5 has allowed for an intuitive and ergonomic user experience, while ensuring effective communication between the detection software and the Arduino. The use of threads for real-time detection ensures the application's smoothness and the system's responsiveness.

The method used, based on sophisticated optical detection combined with an embedded control system, paves the way for a new generation of physical assistance based on biometric communication. The next step will aim to miniaturize the device, perfect the user interface, and extend the control possibilities to other domestic functionalities and other disabilities.

Compliance with ethical standards

Acknowledgments

We have the obligation to fulfilll a pleasant duty, that of thanking all the people who have contributed directly or indirectly to the writing of this article.

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Arduino.cc (2022). Serial Communication Protocol for Microcontrollers. Disponible sur: https://www.arduino.cc/en/Reference/Serial.

[2] Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library.* O'Reilly Media.

[3] Chau, M., & Betke, M. (2005). *Real-time eye tracking and blink detection with USB cameras. Boston University Computer Science Technical Report.*C. Tichi, *Electronic Hearth: Creating an American Television Culture*, Oxford University Press, 1991.

[4] Delorme, A., & Makeig, S. (2004). EEGLAB: an open-source toolbox for analysis of single-trial EEG dynamics including independent component analysis. Journal of Neuroscience Methods, 134(1), 9–21.

[5] Gayatri Mohanta. (2023). GSM-GPS-Based Animal Tracking System: Improving Wildlife Monitoring Efforts. Research and Applications: Embedded System Volume 6 Issue 3. Page 19-26 2023. All Rights Reserved.

[6] Laganière, R. (2019). *OpenCV 4 Computer Vision Application Programming Cookbook* (4th ed.). Packt Publishing.

[7] OpenCV Documentation. (2023). Real-time eye blink detection. Disponible sur : https://docs.opencv.org/

[8] Real Python. (2023). *Python Tutorials for Developers*. Disponible sur : https://realpython.com.

[9] Riyadh, H., & Sharma, N. (2019). Gaze-controlled interfaces for people with disabilities: A comprehensive review. Assistive Technology, 31(3), 136-150.

[10] Wang, J., & Luo, X. (2020). Eye-blink detection using deep learning for assistive technologies. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 28(12), 2625-2634.