LumiTouch: Lighting Adaptation Through Interactive Sensing

Baoyu Li **Mingtong Zhang**
University of Illinois Urbana-Champaign
{baoyul2, mz62}@illinois.edu

1 Motivation

Automated lighting systems are pivotal in the evolution of smart home technologies, focusing on enhancing user comfort, convenience, and sustainability. Our project develops an advanced adaptive lighting system that integrates interactive sensing technologies, such as force sensing resistors (FSR) [1, 2, 3, 4] and ultrasonic sensors [5, 6], to create a responsive environment that dynamically adjusts to the presence and activities of individuals within a space.

Force sensors are critical to our design; they detect even minimal pressures such as those generated when a person sits down or walks across a room. This sensitivity allows the lighting system to react subtly to changes in room occupancy and activity, adjusting the lighting levels to provide optimal visibility and comfort. Meanwhile, ultrasonic sensors enhance the system by using sound waves to accurately measure distances and detect movement [6]. This enables our lighting solutions to intelligently manage light intensity and activation—lights turn on, off, or change in brightness as people approach, move within, or leave the room.

The integration of these sensors not only tailors the lighting experience to the needs of each user but also contributes to significant energy savings. By ensuring that lights are used only when necessary, and at appropriate levels, our system reduces unnecessary power consumption, which is a critical concern in today's energy-conscious world [7, 8]. This capability of our adaptive lighting system to merge convenience, efficiency, and energy conservation makes it an attractive solution for anyone looking to enhance their living or working environment.

Moreover, the implementation of such technology in smart homes and offices has the potential to revolutionize the way we interact with our spaces. It offers a practical solution to the problem of wasted electricity in unoccupied rooms—a common issue in many buildings that not only impacts the environment but also inflates energy bills [9].

Given the increasing focus on sustainability and the integration of smart technology, our adaptive lighting system represents a significant step forward. It promises to improve user experience while providing a tangible return on investment through energy savings and improved environmental stewardship. Investing in this technology means investing in the future of smart, sustainable living, making it an ideal choice for forward-thinking homeowners and businesses alike.

The video of our project can be found at https://drive.google.com/file/d/1xwM0cCyaitssnh8i8X5o6ZnUSOekoxKG/view.

2 Technical Approach

In this section, we present the technical strategy implemented for LumiTouch. Our objective is to smartly tailor the lighting system based on user behavior. This adaptation is accomplished by receiving and processing signals from interactive sensing. As depicted in Figure 1, our system framework is comprised of two main components that facilitate communication between the user and the lighting system: the **interactive sensing** and the **control and processing unit**.

The interactive sensing in our system consists of a force sensing resistor (FSR) and an ultrasonic sensor. The FSR detects touch by reducing resistance when force or pressure is applied to its sensing

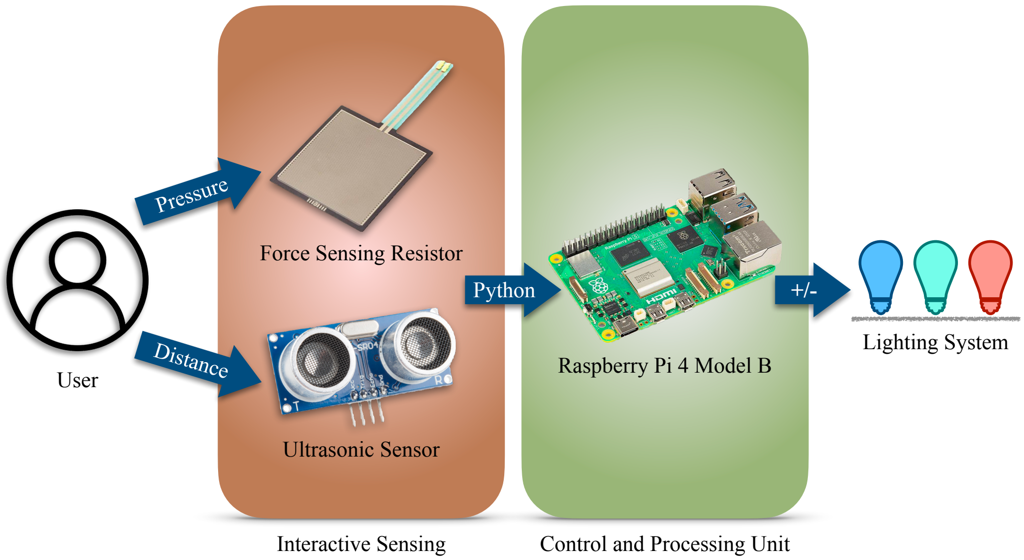


Figure 1: **Overview of LumiTouch.** Our framework consists of two primary components: (a) The **Interactive Sensing** module, which includes a force sensing resistor (FSR) and an ultrasonic sensor. These sensors are designed to gather interactive information, capturing pressure and distance data respectively from user interactions. (b) The **Control and Processing Unit** is powered by a Raspberry Pi 4 Model B, which reads and processes the data from interactive sensors. Using this processed information, the unit dynamically manages the lighting system, adjusting the on/off status and brightness level based on human behavior.

pads, making it a cost-effective and commonly used component in various tactile applications [4, 3]. When a user sits on a chair equipped with an FSR, the sensor detects the pressure and activates the light accordingly. Although the FSR can qualitatively measure changes in force or pressure and adjust the light intensity, it requires an additional ADS1015 ADC board [10] integrated into the Raspberry Pi system to read the analog data from the FSR. Unfortunately, we were unable to procure this board in time, leading us to utilize an alternative interactive sensor, the ultrasonic sensor, which measures the distance between the user and the table to control light intensity. We hypothesize that closer proximity to the table should trigger brighter lighting to create a suitable and comfortable working environment, whereas greater distances suggest the user may be taking a break or leaving the workspace, prompting the system to dim the lights to conserve energy.

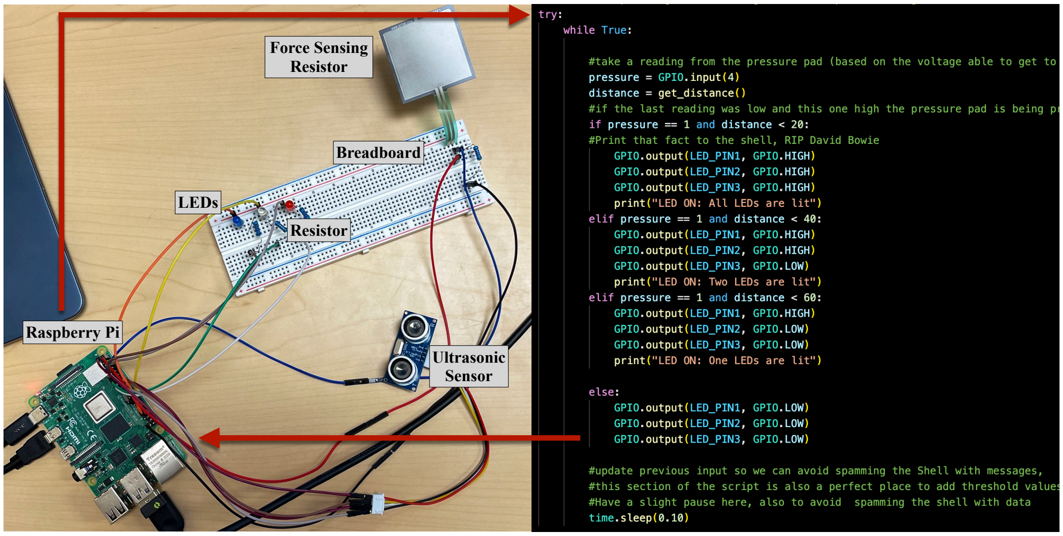
In addition to the interactive sensors, a Raspberry Pi is required to process the signals and control the lighting system. The Raspberry Pi, a compact single-board computer, is extensively used in Internet of Things (IoT) education and projects. Connecting the Rasberry Pi and a breadboard, we can use Python to process the data from the sensors, enabling actions such as switching the lights on or off and adjusting the brightness. The specifics of this implementation will be detailed in Section 3.

3 Implementation Details

In this section, we offer a thorough description of our implementation. We begin with an overview of our hardware modules in Section 3.1. Subsequently, we detail our software implementation and the associated packages in Section 3.2. Finally, we include an informal list of all external materials referenced for the project in Section 3.3.

3.1 Hardware Module

Our hardware setup is illustrated in Figure 2(a). The setup includes interactive sensors (FSR and ultrasonic sensor), a breadboard, $10k\Omega$ resistors, LEDs, a Raspberry Pi, and jumper wires. The Raspberry Pi serves as the central component, equipped with General-Purpose Input/Output (GPIO)



(a) Hardware module (b) Software module

Figure 2: **System Setup.** (a) Our **hardware module** consists of the interactive sensors (FSR and ultrasonic sensor), a breadboard, $10k\Omega$ resistors, LEDs, Raspberry Pi , and jumper wires. (b) Our **software module** relies on Python and the RPi.GPIO module to control and interact with the GPIO pins on the Rasberry Pi.

pins, which enable interaction with external devices like the FSR, ultrasonic sensor, and LEDs. Further details on the GPIO header are provided in the Figure 3.

The FSR is inserted into the breadboard alongside a $10k\Omega$ resistor. In terms of wiring, the red jumper wire is connected to a 5V Pin on the Raspberry Pi and the black jumper wire to a Ground Pin. The blue jumper wire, which we refer to as the Sniffing Wire, is connected to GPIO 4 Pin on the Raspberry Pi. This Sniffing Wire detects whether the FSR is being pressed.

Our setup for the ultrasonic sensor is the same as Lab 1, where we set GPIO pin 23 as the trigger pin and GPIO pin 24 as the echo pin. We utilize these GPIO pins to interface with the ultrasonic sensor. The `get_distance()` function triggers the sensor, measures the time it takes for the signal to return, calculates the distance using the speed of sound, and returns the distance measured. Ensure the ultrasonic sensor's VCC and GND are connected to the Raspberry Pi's 5V and ground pins, respectively, while its trigger and echo pins are connected to GPIO pins 23 and 24.

The LEDs, along with a $10k\Omega$ resistor, are connected to a breadboard and interfaced with GPIO Pins 17, 27, and 22 for the blue, white, and red LEDs, respectively. When the Raspberry Pi receives pressure and distance signals within the set threshold, the LEDs will be activated.

3.2 Software Module

Library The system relies on Python for its ease of use and extensive library support. The specific libraries used in this implementation include:

- RPi.GPIO: This library provides necessary functions to control and interact with the GPIO pins on a Raspberry Pi. It is essential for reading and writing to the pins.
- time: Utilized to manage timing and delays within the script, which is crucial for debouncing the input signal.

Configuration The GPIO pins are configured to operate in the Broadcom SOC channel (BCM) numbering scheme, which is set by invoking `GPIO.setmode(GPIO.BCM)`. This mode allows for referencing the GPIO pins by their Broadcom-specific numbers rather than their physical locations on the Raspberry Pi board. For example, for input detection, GPIO pin 4 is configured as an input pin

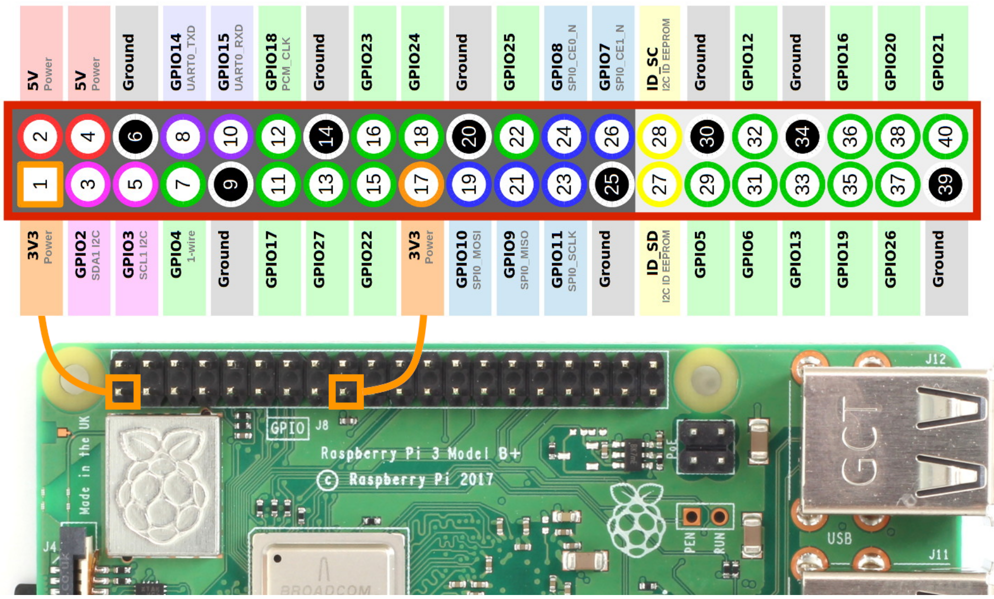


Figure 3: **Raspberry Pi 40-pin GPIO Header.** The GPIO header on the Raspberry Pi is essential for interfacing with the real world. We connect devices like the FSR, ultrasonic sensor, and LEDs to the Raspberry Pi using this header, primarily employing the 5V pins, Ground pins, and various GPIO pins. Refer to `https://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/`.

using the command `GPIO.setup(4, GPIO.IN)`. This setup enables the pin to read the state of the connected pressure pad.

Operational Logic We utilize a function `get_distance()` to measure the distance using the ultrasonic sensor. The distance is calculated based on the time interval between sending and receiving the pulse. This function outputs the measured distance and returns it for use in conditional logic.

The core functionality is contained within an infinite loop, designed to continuously monitor both the pressure pad and the distance sensor:

- Pressure Detection: The script reads the state of the pressure pad. A change from no pressure to pressure triggers subsequent checks and actions.
- Distance Measurement: The `get_distance()` function is called to retrieve the current distance from the ultrasonic sensor.
- Conditional LED Control:
  - Distance < 20 cm: All three LEDs are turned on if the pressure is detected, indicating close proximity.
  - 20 cm $<$ Distance < 40 cm: Two LEDs are turned on, suggesting moderate proximity.
  - 40 cm $<$ Distance < 60 cm: One LED is turned on, indicating further distance.
  - Otherwise: All LEDs are turned off if the conditions are not met.
- We print messages to indicate which LEDs are lit based on the detected conditions, providing real-time feedback about the environment's status.
- Debouncing and Delays: Short delays are included to stabilize the sensor readings and prevent rapid toggling of output states, which enhances the system's reliability.

Algorithm 1 Interactive-Sensor-Based LED Control

```
Import necessary libraries (GPIO, time)
Configure GPIO mode and pin setup
function GET_DISTANCE
    Emit ultrasonic pulse
    Measure echo time
    Calculate and return distance
end function
Initialize sensor pins for pressure and distance
Setup LED output pins
while True do
    Read pressure sensor
    distance ← GET_DISTANCE
    if pressure is detected then
        if distance < 20 then
            Turn on all LEDs
        else if distance < 40 then
            Turn on two LEDs
        else if distance < 60 then
            Turn on one LED
        else
            Turn off all LEDs
        end if
    else
        Turn off all LEDs
    end if
    Delay briefly to stabilize readings
end while
```

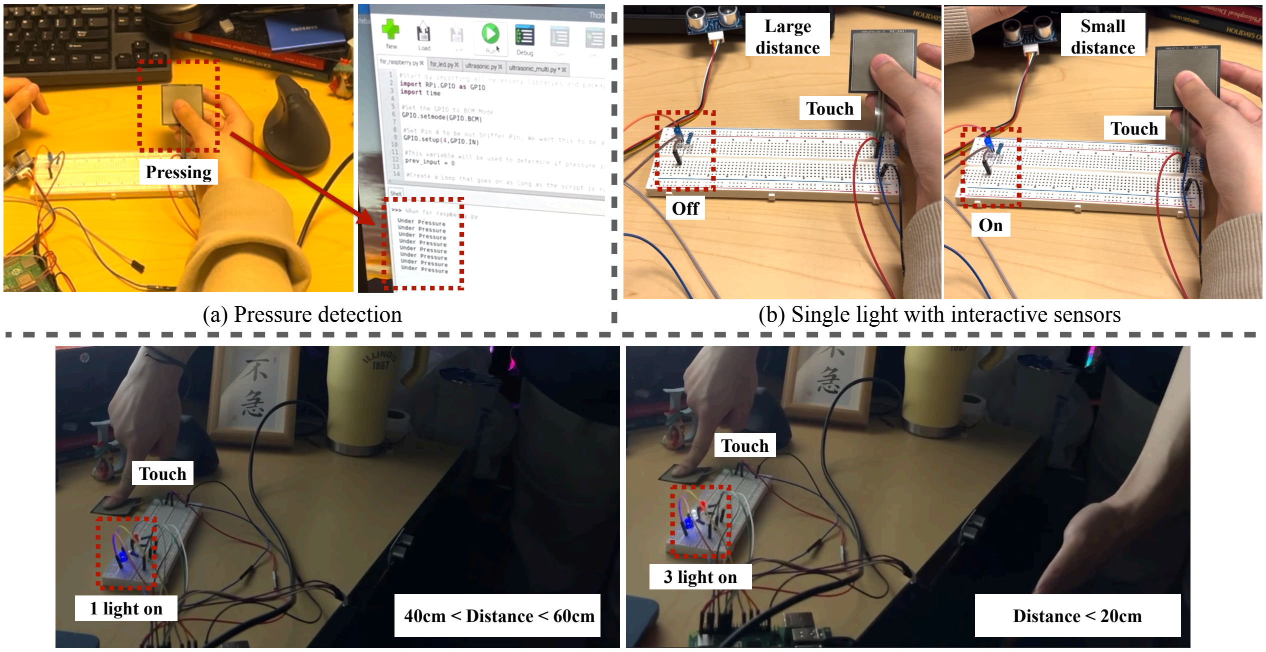
Algorithm 1 presents a concise depiction of the sequence of operations initiated by sensor inputs. Our implementation code can be found at: https://github.com/Boey-li/CS437_LumiTouch.

3.3 Citations

- UIUC CS 437 IoT Lab 1 Self-Driving Car SP24: https://docs.google.com/document/d/1UJoqfUWN6NVATq3yZ2hqSayEcFa0WyAcXU4i42Nubuc/edit#heading=h.3h12sn913lgh.
- How to use Force Sensitive Resistors with a Raspberry Pi and a ADS1015 ADC: https://core-electronics.com.au/guides/force-sensitive-pads-raspberry-pi/.
- Force Sensing Resistor (FSR) with Arduino Tutorial: https://www.makerguides.com/fsr-arduino-tutorial/.
- Simple Guide to the Raspberry Pi GPIO Header: https://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/.

4 Results

In this section, we assess the effectiveness of our interactive sensor-based lighting control framework. Our evaluation specifically seeks to explore the following questions through practical implementations: (1) Is the Raspberry Pi able to detect the pressure applied to the FSR pad? (2) How do the FSR and the ultrasonic sensor collaborate to control a single LED based on touch and distance? (3) How effectively do various interactive sensors coordinate to manage the outputs of multiple LEDs?



(a) Pressure detection

(b) Single light with interactive sensors

(c) Multiple lights with interactive sensors

Figure 4: **Results.** (a) When the FSR pad is pressed, the Raspberry Pi receives the pressure signal and displays "under pressure" on the interface. (b) If the FSR pad is touched without being near the ultrasonic sensor, the light remains off; however, if the FSR pad is touched while close to the ultrasonic sensor, the blue LED will light up. (c) Additionally, we use multiple LEDs within our system, activating a varying number of LEDs based on the distance signal to adjust the brightness of the lighting system.

4.1 Pressure Detection

First, we evaluate the effectiveness of pressure detection from the FSR pad. As shown in the Figure 4(a), when pressure is applied to the FSR pad, the Raspberry Pi successfully receives the signal and displays "under pressure" on the interface.

4.2 Single Light with Interactive Sensing

In Figure 4(b), the outcome of controlling a single LED's on/off status based on touch and distance signals from the interactive sensors is displayed. The LED will only illuminate when both the FSR pad is pressed and someone is near the ultrasonic sensor simultaneously. If either condition is not met, the light will remain off.

This design is to ensure that the lighting system only activates when there is a clear indication of a user's active engagement at a specific workstation. This dual-condition setup reduces false positives—where the light might turn on due to incidental or irrelevant movements—and helps conserve energy by only lighting the area when someone is both physically present and actively interacting with the workspace.

4.3 Multi Lights with Interactive Sensing

We further expand the system from controlling a single light to managing multiple lights, adjusting their brightness based on distance signals, as depicted in Figure 4(c). Similar to Section 4.2, the LEDs are triggered by both pressure and proximity. Additionally, the number of activated LEDs varies with the distance signal; for instance, only one LED lights up when the distance is between 40cm and 60cm, while three LEDs activate when the distance is less than 20cm. This design allows for variable brightness control, making the lighting system more adaptable to different situations and environments.

4.4 Discussion of Results

The results underscore the practicality and innovation of our interactive sensor-based lighting control system, showcasing its broad applicability in various fields. The ability of the Raspberry Pi to detect pressure on the FSR pad illustrates the system's capability to sense user interactions, which is crucial for settings that demand active user participation. By incorporating both pressure and proximity sensors, the system provides targeted lighting triggered by specific actions, improving the user experience and fostering energy conservation. The system's extension to manage multiple lighting units illustrates its flexibility to adapt to different environmental conditions by dynamically adjusting the light intensity. Collectively, these features highlight the system's potential to boost operational efficiency and sustainability in environments ranging from professional workplaces to smart home systems.

5 Conclusion

In this project, we introduce LumiTouch, an adaptive lighting system equipped with interactive sensors. Our system dynamically adjusts the lighting based on user interactions detected by a force sensor resistor and an ultrasonic sensor, all processed through a Raspberry Pi. It effectively manages lighting, from a single LED to multiple LEDs, and is versatile enough to adapt to different environments and situations, enhancing user experiences and promoting energy savings.

This project represents a fascinating new venture for our team, as it expands our expertise from purely software to incorporating hardware. This is our first foray into using tactile sensors, which we have employed to develop an adaptive lighting system with the Raspberry Pi. We are thrilled with its functionality and are eager to explore the potential of tactile sensors in our future research or engineering projects. We believe tactile sensors hold great promise for developing more innovative and beneficial products for people and can also enhance the ability of robots to learn about and interact with the physical world.

6 Group

This project was equally contributed to by a team of two MSCS students: Baoyu Li (baoyul2) and Mingtong Zhang (mz62).

References

[1] S. Yaniger. Force sensing resistors: A review of the technology. In *Electro International, 1991*, pages 666–668, 1991. doi:10.1109/ELECTR.1991.718294.

[2] M. Y. Saadeh and M. B. Trabia. Identification of a force-sensing resistor for tactile applications. *Journal of intelligent material systems and structures*, 24(7):813–827, 2013.

[3] A. Sadun, J. Jalani, and J. Sukor. Force sensing resistor (fsr): a brief overview and the low-cost sensor for active compliance control. In *First international workshop on pattern recognition*, volume 10011, pages 222–226. SPIE, 2016.

[4] D. Giovanelli, E. Farella, et al. Force sensing resistor and evaluation of technology for wearable body pressure sensing. *Journal of Sensors*, 2016, 2016.

[5] B. Hoyle and L. Xu. *Ultrasonic sensors*. Oxford, UK: Butterworth-Heinemann, 1995.

[6] C. Khampachua, C. Wongrajit, R. Waranusast, and P. Pattanathaburt. Wrist-mounted smartphone-based navigation device for visually impaired people using ultrasonic sensing.

In *2016 Fifth ICT International Student Project Conference (ICT-ISPC)*, pages 93–96, 2016. doi:10.1109/ICT-ISPC.2016.7519244.

[7] I.-T. Kim, Y.-S. Kim, M. Cho, H. Nam, A. Choi, and T. Hwang. High-performance accuracy of daylight-responsive dimming systems with illuminance by distant luminaires for energy-saving buildings. *Energies*, 12(4), 2019. ISSN 1996-1073. doi:10.3390/en12040731. URL https://www.mdpi.com/1996-1073/12/4/731.

[8] H. E. Degha, F. Z. Laallam, and B. Said. Intelligent context-awareness system for energy efficiency in smart building based on ontology. *Sustainable Computing: Informatics and Systems*, 21:212–233, 2019. ISSN 2210-5379. doi:https://doi.org/10.1016/j.suscom.2019.01.013. URL https://www.sciencedirect.com/science/article/pii/S2210537918303457.

[9] M. Soheilian, G. Fischl, and M. Aries. Smart lighting application for energy saving and user well-being in the residential environment. *Sustainability*, 13(11), 2021. ISSN 2071-1050. doi:10.3390/su13116198. URL https://www.mdpi.com/2071-1050/13/11/6198.

[10] ADS1015 12-Bit ADC - 4 Channel with Programmable Gain Amplifier. https://www.adafruit.com/product/1083, 2024.