Automatic Parking Meter

ECE 445 Design Document

Team 43: Elliot Salvaggio, Kishan Surti, Rutu Patel

ECE 445 Project Proposal - Spring 2020

TA: Shuai Tang

1 Introduction

1.1 Problem and Solution Overview

In recent years, we have seen rapid innovation in the car industry. The push for quality electric vehicles and self-driving capabilities for cars has been more and more apparent over the last decade. Despite this progress that has been made in simplifying vehicles, the same has not been made for paid metered parking, an archaic experience everyone with a car must deal with. Overpaying for parking costs Americans over $20 Billion a year [9] and an average of 125,000 people receive a parking citation every single day in the US [10]. We believe this can be attributed largely to the inconveniences associated with parking meters. Although in some cities one can use a simple app to pay a meter, this does not remove the root of the problem. It can be hard to estimate the length of stay one anticipates on being parked at a location, and might end up underpaying, which can lead to a ticket. People also clearly overpay for parking significantly, which adds up over time. It can be stressful knowing you have to leave your location when you're running low on parking time, and difficult to decide how long you believe you will be there for. Additionally, if you're in a rush, one might forget to pay at all. We believe we can solve all these problems with a simple solution.

We propose a solution that is an add-on unit to existing meters in the parking lot. Users would add their name and license plate information to the app associated with our system. Once they park in a parking spot, the unit would detect the car moving into the parking space and take a picture of it's license plate. Our unit will analyze this picture, find the user's license plate number, and using this information, charge them for their stay automatically as the user returns to the spot and drives away. Our solution eliminates the need for someone to worry about whether they've paid for their parking, or have to figure out the length of their stay before they've even gotten out of the car. With our solution, people can be relieved they no longer have to worry about receiving a parking citation.

1.2 Background

We decided to do something about this problem because we have cars and often complain about the issues that come from having to park in paid parking spots on campus. We've been in situations where we forget to pay for parking and get tickets, or have to leave a meeting because we don't want to have to pay for an additional twenty minutes of parking that may not be needed. We believe that with the basic technology we have today that we can come up with a solution to fix this old broken design.

The basic problem is that most parking meters just take coins to pay for parking. We know that on campus the MobileMeter app is the solution to this issue, and it does solve the problem of using coins, but all the issues that come from overpaying and underpaying for parking persist with this simple app. This is why we feel like we can still do better than this existing basic solution. In ideal conditions, someone will simply drive into a parking spot, leave for some time, then return and drive off, and behind the scenes our product will do all the work to keep track of their stay.

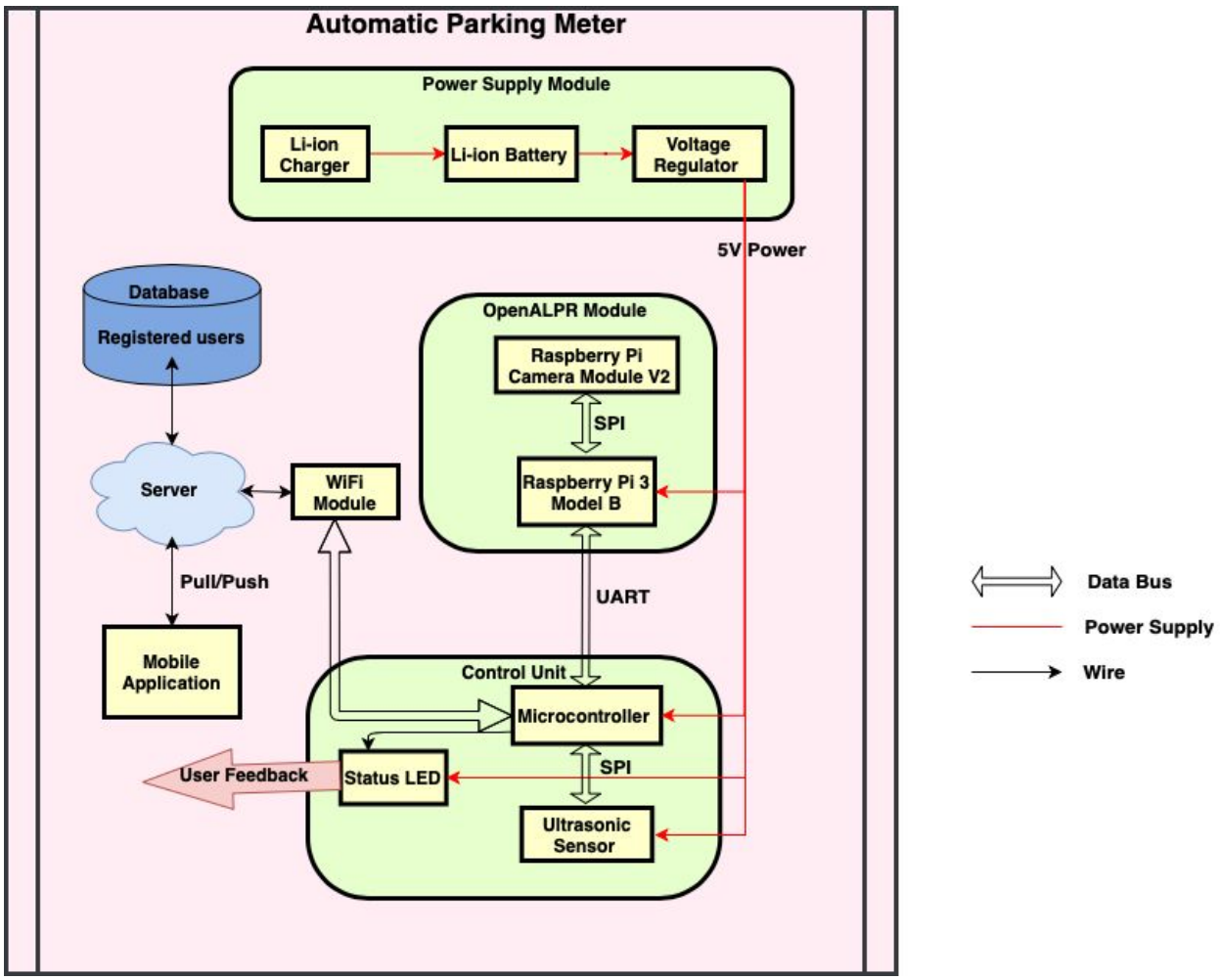
1.4 High-Level Requirements

- Able to detect a car entering and leaving a parking spot when it is within 5 feet of our system.
- Able to extract a vehicle's license plate, with exactly seven characters of precision, from a picture taken of a parked car's license plate and associate that plate number with a registered user.
- Able to give feedback to the user that our system has recognized the parked car's plate number and registered their stay through a single status LED within 30 seconds.

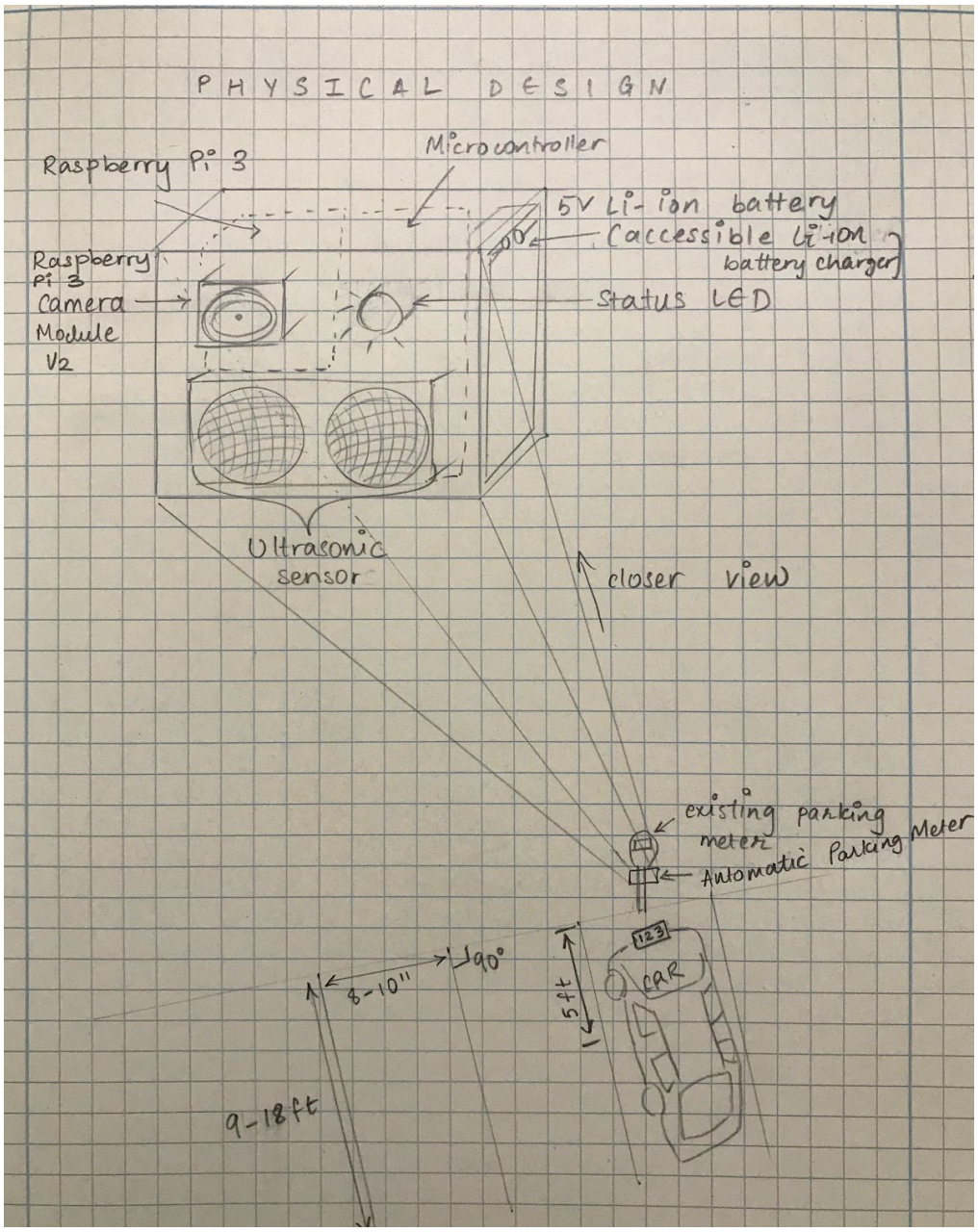
2 Design

2.1 Block Diagram

There are three main parts to the design of our system. There is the PCB microcontroller system called control unit that contains the ultrasonic sensor and LEDs, the Raspberry Pi system that has the camera attached called OpenALPR module, Power Supply module for required energy to run the overall system, and the Server-App-Database system that communicates with the Raspberry Pi to keep track of user data. *Figure 1* shows the blocked diagram of the proposed Automatic Parking Meter.



2.2 Physical Design



2.3 Power Supply

We need a power supply module which is responsible for providing power to the control unit, and OpenALPR unit. The following are components of this system.

2.3.1 Li-ion Battery: We will have one rechargeable 5V Li-ion battery that will be used to supply power to each component of the project; with power to the multicontroller and Raspberry Pi 3. We will need a 5V power supply for the Raspberry Pi 3 B [8], and multicontroller.

Requirement	Verification
The battery must be able to power the system to be able to run OpenALPR, sensor working, as well as illuminate status LED at 4.5-5.5V.	1. Use a multimeter across the battery to ensure that the voltage supplied is within 4.5V-5V

2.3.2 Li-ion Charger: Li-ion Charger is used to charge the rechargeable Li-ion battery used in the overall system, so the user is able to manually charge the battery.

Requirement	Verification
Li-ion battery recharges to 4.5V-5V from a charger supplying 5V DC input.	1. Use an AC 100-240V wall outlet charger that converts to DC 5V, and use this to charge the battery. 2. At the end of a charge cycle, use a multimeter across the charged battery to measure whether it is within the desired 4.5-5.5V.

2.3.3 Voltage Regulator: We will use a voltage regulator to regulate the voltage that is sent to different components such as Raspberry Pi 3, ultrasonic sensor, and microcontroller .

Requirement	Verification
Must be able to provide 2.6V-3.8V for LED	1. Use a multimeter to measure the open-circuit voltage where we would place the LED. 2. Ensure that the voltage gets in the correct range by inserting a resistor with the proper resistance value
Must be able to provide 4.5V-5.5V for the rest of the system.	1. Use a multimeter to measure the voltage to be 4.5-5.5 V across the Raspberry Pi and microcontroller ATMega328 chip.

2.4 Control Unit

Control Unit is responsible for managing the inputs from sensors, communicating with OpenALPR unit and at the end giving feedback to users via Status LED.

2.4.1 Microcontroller: Microcontroller will be responsible for taking inputs from the ultrasonic sensor, trigger the OpenALPR unit; specifically Raspberry Pi Camera Module to take inputs, and wait for the OpenALPR unit output through SPI (Serial Peripheral Interface). It will then output the user feedback with Status LED. We plan to use a ATMega328P microcontroller. It is simple, low powered, and has low costs [11]. We believe it should be useful in a project of our size because it is known to be used in the Arduino Uno, which is a similar size to the multicontroller we plan to build.

Requirement	Verification
• Must be able to take ultrasonic sensor input within 5 meters.	1. Program the microcontroller with Arduino code. 2. Leave the arduino connected to the laptop, and have the code check if the sensor detects an object within 5 meters. 3. Confirm by printing some output if the sensor was able to detect something successfully.
• Must be able to receive feedback from the WiFi module and output green or red status LED.	1. Send any dummy data input to the server, via WiFi module to store into the database. 2. Check the data in the database to see if the data write occurred. 3. Program the microcontroller about parsing the data base for the desired user, and take the output as a yes or no. 4. Provide specific voltage across LED to illuminate it based on red or green color voltage specification.

2.4.2 Ultrasonic Sensor: For the purpose of triggering the system, we need to detect the distance between our system and the car entering or leaving the parking spot, for which we will use an Ultrasonic sensor [5], since they are more reliable than IR sensors during day time due to the influence of sunlight. The Ultrasonic sensor will measure the distance of the car to our system,

which would be set for 5 meters, thereby communicating with the microcontroller to initiate the camera input and begin processing.

Requirements	Verification
● Must be able to accurately detect the car entering the parking spot at a distance of 5 meters from our system.	1. Program the microcontroller in such a way that it holds the threshold for the ultrasonic sensor of about 5 meters to detect the car approaching. 2. Add print statements as output to confirm the sensor input is accurate after 5 meter threshold.

2.4.3 Status LED: For the purpose of displaying to the user if our system recognizes the car parked or not, we will use a bi-color standard Green and Red LED, that consumes 20mA current and 2.6-3.8 V Voltage [4]. The input to the LED is from the microcontroller regarding the OpenALPR unit.

Requirements	Verification
● Must be properly illuminated when bias with the specified voltages.	1. Confirm using a voltmeter across LED, that it receives 2.6-3.8V.

2.5 OpenALPR Module

OpenALPR (Automatic License Plate Recognition) module is considered as the main brain of our system as we use OpenALPR open source image recognition library for extracting license number from license plate image. This module communicates with the microcontroller for its input and output. When the ultrasonic sensor on the multicontroller finds there is a car moving into the spot within 5 feet of the system, we will send a signal to a Raspberry Pi 3 Model B via

microcontroller for the Pi to trigger its camera module and start computing the license plate number.

2.5.1 Raspberry Pi 3 Model B: We will use a Raspberry Pi 3 [6] to run computer vision algorithms on the picture captured by its attached camera. The attached Raspberry Pi Camera Module V2 will be triggered to take a picture by input from the control unit i.e microcontroller. Once the picture is stored on the RP3, we will run OpenALPR (Automatic License Plate Recognition) algorithms on the image to extract the car's license numbers. We will use the built in WiFi module of the Raspberry Pi 3 to send the extracted plate number to a database/server, where it will check if a user with such license number has registered through the mobile app. The Raspberry Pi would further communicate the information received from the database to the microcontroller.

Requirement	Verification
● Must be able to take microcontroller input to trigger the Raspberry Pi Camera Module V2.	1. Connect the Raspberry Pi 3 along with the Camera Module V2 to any laptop. 2. Confirm that Raspberry Pi 3 triggers the Camera Module V2, after keypress. [13] 3. Confirm the camera takes a picture in any picture formats stored on the laptop.
● Must be able to perform OpenALPR accurately extracting the license number from the license image, character by character.	1. Visually confirm the output from the OpenALPR algorithm returns the same number as you can visually see on the license plate.

2.5.2 Raspberry Pi Camera Module V2: We will attach a 8 MP Raspberry Pi Camera Module V2 [7] to our Raspberry Pi 3 B that will take a picture of the car's license plate when it is notified by the microcontroller in the control unit. The camera will be attached on the front side of our system [2].

Requirements	Verification
● Be able to take a clear picture when notified by the Raspberry Pi 3 that is good enough quality of an image that it can be properly analyzed by the OpenALPR system.	1. Connect the Raspberry Pi 3 along with the Camera Module V2 to any laptop. 2. Run the script for taking a picture via camera module. 3. Confirm the camera takes a picture of good quality stored on the laptop.

2.6 Mobile Application: We will utilize a mobile application associated with our parking meter. This will give a more detailed feedback to the user, and they will be able to see the duration they have been parked in a spot, how much they would get charged accordingly, other car information details, and also the user's QR code that can be used for scanning purposes. The QR code will provide as an alternative if the Camera Module cannot detect the vehicle's license plate for some reason, and the user will be able to scan this to secure their parking spot and begin their time parked. In addition, we will have the user register their information within the application prior to parking, this is necessary for license plate recognition to be associated with an existing account.

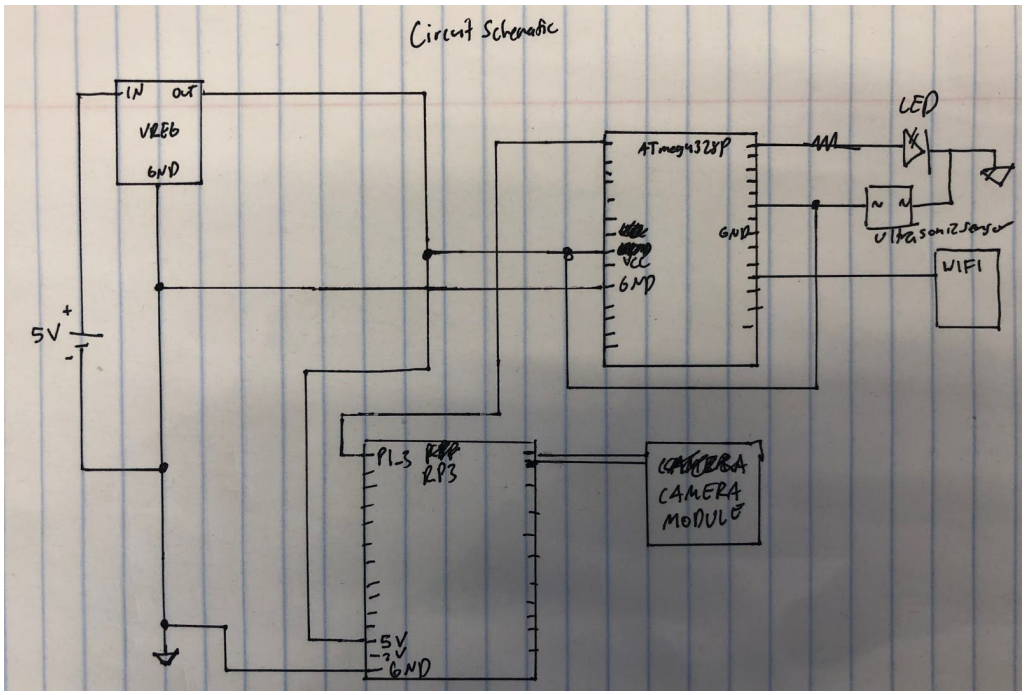
Requirement	Verification
● Users can register their information in the application and can be updated in the database.	1. Using the log-in page of the application, create a new user and refresh the database immediately after.

	2. Check if the database has now changed to reflect the addition of a new user.
● QR code can be accessed from the display page, and is scannable by the camera.	1. Connect a test LED to the Raspberry Pi, and program the camera to trigger it when a user scans a QR code. Then, take a basic QR code and provide it to the camera, and if it correctly recognizes it, the LED should light up.

2.7 Server/Database: In order to have users register their information within our mobile application, we must have some sort of database that can retain this information, and can be easily accessible with our app. We have chosen to use Firebase for our backend, this will integrate nicely and we can access it quickly and efficiently.

Requirement	Verification
● We are able to retain information in the database, and make efficient access to it when checking if an account exists under a license plate.	1. Program the code to perform a read and write to the database 2. Measure the latency of both the read and write using a network capture and ensure that it is under one second.
● We are able to create new users in the database from the mobile application.	1. Parse the database to see if its been updated with new user data when registered from the app.

2.8 Schematics



3 Cost and Schedule

3.1 Cost Analysis

3.1.1 Labor Cost

We are a team of three, all Computer Engineering students. The average starting salary for an Illinois Undergraduate Computer Engineering graduate was $96,518 in 2016. This comes out $96,518/52 weeks = $1,856.11 per week. $1,856.11/40 hours = $46.40/hour. The semester is 16 weeks long. If we all plan to work 20 hours a week on the project, that puts the total labor cost to (46.40*20*16*3) = $44,544.

4 Discussion of Ethics and Safety

One of the few concerns regarding our project will be the power supply. The parking meter should theoretically be placed outside and should be able to endure various weather conditions, including rain. We must ensure that our power supply is covered properly so that rain or snow cannot enter. This could potentially cause a hazard to the user if they are touching the parking meter and it has been exposed to water leakage in some manner. This applies to the IEEE Code of Ethics #1: "to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment" [1].

Another ethical issue that can arise is keeping a user's personal information safe and secure. We will only ask for their license plate, phone number, and full name. We will not disclose any of this information, and our system will not retain any of this information for our benefit, nor will we keep track of a user's location. We will only display accurate information on our parking meter and within the mobile application. This aligns with the IEEE Code of Ethics #3: "to be honest and realistic in stating claims or estimates based on available data"[1].

5 Citations:

[1] "IEEE Code of Ethics." IEEE, www.ieee.org/about/corporate/governance/p7-8.html.
[2] "Camera Configuration." Camera Configuration - Openalpr 2.7.102 Documentation, doc.openalpr.com/camera_placement.html.
[3] 9-4.117 Parking Design Standards., qcode.us/codes/atascadero/view.php?topic=9-4-9_4_117.
[4] "5mm Bicolor Green & Red LED Specs, Data Sheet N500TGR4D from Lc." Led.com, www.lc-led.com/products/n500tgr4d.html.

[5] Pendergast, Robert L., et al. “Complete Guide for Ultrasonic Sensor HC-SR04 with Arduino.” Random Nerd Tutorials, 2 Apr. 2019, randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/.

[6] “Buy a Raspberry Pi 3 Model B – Raspberry Pi.” Buy a Raspberry Pi 3 Model B – Raspberry Pi, www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[7] “Camera Module.” Camera Module - Raspberry Pi Documentation, www.raspberrypi.org/documentation/hardware/camera/.

[8] “Power Supply.” Power Supply - Raspberry Pi Documentation, www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md.

[9] Inrix. “Searching for Parking Costs Americans $73 Billion a Year.” PR Newswire: Press Release Distribution, Targeting, Monitoring and Marketing, 26 June 2018, www.prnewswire.com/news-releases/searching-for-parking-costs-americans-73-billion-a-year-300486543.html.

[10] “Traffic Ticket Statistics.” Davis Law Firm, 1 Oct. 2019, jeffdavislawfirm.com/traffic-ticket-statistics/.

[11] Nasir, Syed Zain, et al. “Introduction to ATmega328.” The Engineering Projects, 6 Feb. 2019, www.theengineeringprojects.com/2017/08/introduction-to-atmega328.html.

[12]https://www.alliedelec.com/m/d/4252b1ecd92888dbb9d8a39b536e7bf2.pdf

[13] https://www.raspberrypi.org/documentation/hardware/camera/

[14] https://projects.raspberrypi.org/en/projects/getting-started-with-picamera

[15] https://www.arduino.cc/en/Tutorial/ArduinoToBreadboard