University of Zurich[UZH]

Fully Automated Charging of Electric Vehicles

Areg Arakelyan
Berikon, Switzerland
Student ID: 16-732-935

Supervisor: Prof. Burkhard Stiller and Dr. Eryk Schiller
Date of Submission: January 16, 2023

University of Zurich
Department of Informatics (IFI)
Binzmuehlestrasse 14, CH-8050 Zuerich, Switzerland

BACHELOR THESIS – Communication Systems Group, Prof. Dr. Burkhard Stiller

Bachelor Thesis
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmuehlestrasse 14, CH-8050 Zuerich, Switzerland
URL: http://www.csg.uzh.ch/

Zusammenfassung

Im Zusammenhang mit ernsthaften Umweltproblemen und geopolitischen Herausforderungen hat die Elektromobilität in den letzten Jahren an Bedeutung zugenommen und wird weltweit akzeptiert und adaptiert.

Die Zukunft ist elektrisch und unser Ziel ist es, die infrastrukturellen Herausforderungen insbesondere in älteren Schweizer Mietwohnungen mit Tiefgaragen für ihre Bewohner anzugehen. Benutzerfreundliche Ansätze zum Teilen von Ladestationen zwischen Bewohnern, ein vollständig automatisiertes Ladeverfahren und Lösungen für Verbindungsprobleme in Tiefgaragen sind das Hauptaugenmerk unserer Forschung.

Einzigartig an unserer Lösung ist die Verwendung eines niedrigleistungsfähigen primitiven tragbaren EV-Ladegeräts als Alternative zu kommerziellen (hochleistungsfähigen) Geräten, die mit vorinstallierten Kommunikationsprotokollen mit einem zentralen Management-System ausgestattet sind.

Wir haben nicht nur das primitive Gerät digitalisiert, so dass es fernbedient und mit einem zentralen Management-System kommunizieren kann, sondern auch eine angepasste Umgebung implementiert, die ein vollständig automatisiertes und hands-free EV-Ladeverfahren basierend auf Fahrzeugkennzeichenerkennungsmechanismen ermöglicht. Unsere Lösung kann an jede normale Steckdose angeschlossen werden und erfordert keine Infrastruktur-Upgrades in älteren Gebäuden.

Gleichzeitig können die Benutzer immer noch von einem zentralen Management-System und einer intuitiven Web-Oberfläche zur gemeinsamen Nutzung profitieren, insbesondere im Hinblick auf Benutzerfreundlichkeit und ordnungsgemässes Feedback vor, während und nach dem Ladevorgang.

Abstract

In relation to serious environmental issues and geopolitical challenges, electric mobility has increased in relevance in the past years up to an extent of worldwide acceptance and adaptation.

The future is electric, and our work is aiming to confront the infrastructural challenges focusing especially on older Swiss rental apartments that provide underground parking options for their residents.

User-friendly approaches for the sharing of charging stations between residents, a fully automated charging procedure, and solutions regarding connectivity issues in underground garages are the main focus of our research.

Unique to our solution is the usage of a low-power primitive portable EV home charging device as an alternative to commercial (high-power) devices that come with pre-built communication protocols with a central management system.

We did not only digitize the primitive device enabling its remote usage and communication to a central management system but we also implemented a tailored environment that enhances a fully automated and hands-free EV charging approach based on vehicle plate number recognition mechanisms.

Our solution can be plugged into any domestic socket and implies no infrastructural upgrades in older buildings. At the same time, users can still take advantage of a central management system and an intuitive web frontend for the purpose of shared usage focusing especially on user-friendliness and proper feedback before, during, and after the charging process.

When it comes to the installation costs, our solution represents a fraction of the costs of commercial charging stations since low-power charging devices are much more affordable and foremost because no installations done by certified electricians are longer required.

Regarding future plans: Our work will be further developed to adapt the form of a black box that can be easily integrated into any Swiss residential building that provides an available home socket.

Acknowledgments

First and foremost, I would like to thank the Communication Systems Group and specially Prof. Burkhard Stiller and my supervisor Dr. Eryk Schiller for the great opportunity.

Your sincere and good willed contribution and support to this project have been incredibly encouraging. Since the first day I have felt tremendous support make this project doable.

Their constructive feedback along with ideas and out of the box thinking have been a key aspect during the whole project. This project would not have possible without you.

Contents

Chapter 1

Introduction

1.1 Motivation

Electric and Plug-In Hybrid Vehicles (EVs) have been acquiring worldwide relevance in the last decade due to their advantages. The potential is overwhelming - especially when combined with renewable energy sources such as photovoltaic.

EVs are a valuable option to keep in mind. In contrast, future plans show a high initiative to take advantage of photovoltaic sources by integrating panels directly into the roofs of cars. This, however, lies far in the future since the current technology standards are confronting severe challenges regarding the storage and transportation of electricity.

Nowadays, it is inevitable not to consider hybrid and electric options when it comes to the acquisition of a new vehicle. On the one hand, we have the combustible crisis worldwide, with prices oscillating up to 7.6 percent in the United States within 8 days [1] as an example of one of the most extreme cases.

On the other hand, the environmental factor is as well an important aspect to keep in mind.

Car manufacturers are drastically adapting their assortment, reaching a point where some only provide electric and hybrid options [2].

At the same time, purely electric competitors such as Tesla, Lucid Motor, or Rivian are increasingly gaining market share in the automobile industry [3].

Our solution aims to support the expansion of electric mobility in Switzerland by providing a solution adaptable for the most prominent Swiss population group, i.e., tenants leaving in apartment buildings for rent.

For this purpose, this work takes advantage of a tailored hardware and software infrastructure that relies on Automatic Number Plate Recognition (ANPR) mechanisms, allowing hands-free and fully automated EV charging while guaranteeing decent security and privacy standards for electric and EV drivers (e-drivers).

1.2 Status Quo in Switzerland

Based on surveys conducted in Switzerland, the status quo indicates an explicit dependency on a good network of charging stations for the proper adaptation of EVs. [4].

People need more comfortable and efficient charging alternatives than driving to an external charging station and waiting up to 40 minutes for a full charge, for example, if driving a plug-in hybrid electric Mitsubishi Outlander that provides a total range of 40 kilometers (when driving full electric) and charging it on a 43-50kW rapid charging station [5].

However, significant challenges [6] such as the low number of charge points or the geographical spread of the charge points prevent a proper rollout.

The dominant majority of the charging stations in Switzerland are private stations (in houses or private agglomerations) or publicly available against a charging fee.

When constructing new buildings, charging options are evaluated and considered by default.

The recommended solution for rental buildings in Switzerland is to build a charging station in each parking spot or at least provide the necessary infrastructural requirements to install them in the future if the renter requests as described by the Municipal Electric Utility Zurich in one of their official advisory whitepapers [7] regarding the installation of charging stations.

Installing a charging station in all parking slots in older buildings is challenging and expensive since these do not provide the required infrastructure. Most of the existing Swiss edifices were not built aiming to support efficient EV charging along with high-power devices [8].

In the following sections, we have categorized the factors that impede the rollout:

1.2.1 Financial Limitations

The status quo and the standards within the financing of EV charging points are [9]:

1. Subscription models for e-drivers and transaction costs (pro kWh) are the most common approaches. E-drivers pay for the usage of a central management system that is designed for remote maintenance of the charging stations, charging management, the collection of billing data, and user management. The owner of the building on the other hand takes the costs for the device and its installation.

2. Splitting the costs between both the e-driver and the property owner. The owner will cover the installation costs (since the installation is location-dependent), and the e-driver (tenant in the property) takes the costs of the charging device along with an additional monthly fee for the usage of a central server system.

There is no defined standard for the facilitation of an EV charging infrastructure and the financial costs can be regarded as one major factor. Real estate managers [10] and owners are aware that there is no clear regulation or need for the adaption of charging infrastructure. At the same time, most Swiss rental apartments are often rented out.

1.2.2 Infrastructural challenges

In Switzerland, there are strict regulations when it comes to the installation of high-power devices [11]. Most charging commercial wall boxes depend on power sources of at least 11 kW. These apply under the category of high-power devices and require the availability of a high-current power source and cable, which must be installed by a certified electrician [12].

Furthermore, a load management system must be installed if more than 50 kW is requested from all the charging stations combined [13].

The load management system is responsible for the secure, efficient, and automated loading of the electric power available in a building.

For instance, if the building's energy consumption is high, the amount of available current for the charging procedures will be decreased, aiming for an efficient balancing of the electricity in a building.

1.2.3 Connectivity challenges

A compelling shared approach for EV charging is only doable if internet connectivity is provided based on the necessity of a central management system that verifies and authenticates the e-drives and tracks their charging costs [14].

The dependency on internet connectivity to manage charging sessions is a further significant challenge in underground garages since, in most of the underground garages, there is no Global System for Mobile Communications (GSM) signal or internet connection, nor wireless fidelity (WIFI).

This barrier is currently handled by distributing physical keys, for instance in form of Radio-frequency identification (RFID) cards to the e-drivers. However, the dependency on physical devices is often translated into further administrative costs. These must be distributed and may get lost, need replacement, or be interchangeable among the e-drivers.

1.3 Description of the Work

The main goal of this thesis is to develop a hands-free and fully automated approach for charging EVs in locations where no internet connectivity is provided for e-drivers. To achieve this, this project will take advantage of current state-of-the-art automated plate recognition systems for vehicle identification and authentication, along with a user management tool. This will avoid any dependencies on physical keys or devices.

All the communication between the charging points and the central management system will be handled by our own communication mechanisms, which will allow an accurate power consumption measurement relying on a digital power meter.

For this purpose, we will use a smart composition of software and hardware components that will allow the shared, hands-free, and fully automated usage of a low-power home portable charger (3.6 kW).

A user-friendly web front end will serve for the tracking and traceability of the charging sessions and the power consumption, along with plots and diagrams.

The front end will also serve as a backup for the triggering of the charging sessions if the ANPR systems do not correctly identify the vehicle due to damages or missing license plates.

1.4 Thesis Outline

This project encapsulates a smart software and hardware solution tailored for low-power portable chargers (3.6 kW) aiming a fully automated and hands-free shared usage and user cost-accountability.

The solution will serve as an alternative to the widely adopted high-power current charging solutions that come with the standardized communication protocol Open Charge Point Protocol (OCPP).

We aim to enhance electric car adaptation by providing an accessible, adaptable, and more affordable option for shared charging stations in Swiss rental buildings.

The ease of the installation of the low-power device, the tailored central management system, and the avoidance of any app or physical keys for the charging process along with the overall reduction of the installation when compared to a high-power device due to the usage of low-power devices will enhance the adaption of charging stations in Swiss rental buildings which will potentially motivate renters to adapt to the new technology.

Chapter 2

Related Work

Due to the rapid adoption of EVs, standardized communication protocols have been appointed for charging stations and their communication with a central server.

This project takes into consideration the status quo of the communication protocols used in EV charging, their advantages, and their weak points. This will serve as a base to conceive and design our communication mechanism between the central server and our low-power charging device while only relying on limited hardware resources.

Furthermore, this project will evaluate the future of EV charging approaches to identify the direction the current development is heading. Keywords regarding the future of EV charging are Smart Charging and the ISO 15118 communication interface.

2.1 Open Charging Point Protocol

OCPP is a commonly used communication protocol between charging stations and the server (i.e., the central management system). The protocol is mainly used to support the communication between the charging points and a central management server. However, the protocol is also applied to process information from the central management server towards the devices and other systems related to the charging management process, as described by [15].

2.1.1 Versions of the Open Charging Point Protocol

Nowadays, most public charging stations use the OCPP, more specifically, the OCPP version 1.5, which relies on the messaging protocol SOAP and Hyper Text Transfer Protocol (HTTP). With the newest version of OCPP (OCPP 1.6 and above), novel approaches for message transferring are being used, for example, the JavaScript Object Notation (JSON) notation.

OCPP 1.2 and 1.5

The initial versions of OCPP were mainly focusing on the core functions to successfully establish the communication between a charging point and a central management system.

OCPP 1.2 and it's updated version where more functionalities were added, OCPP 1.5, consists of 25 operations from which 15 are initiated by the server and 10 are triggered from the charging point [16]. The versions of OCPP 1.2 and 1.5 relied purely on Simple Object Access Protocol (SOAP) as a messaging protocol [17].

To provide an overview and illustrate some of the operations available in the OCPP protocol, we summarize a few key features supported by the communication protocol.

Authorize: The charging point sends a request to the central management server aiming to establish a Web Socket connection.

Diagnostics Status Notification: The charging point sends its current status to the central management server.

Meter Values: The charging point sends measurement data on the current electrical power flowing from the charging point toward the EV.

Start/Stop Transaction: The central management system will trigger or block the charging session by sending a request to the charging point.

OCPP 1.6 and the introduction of JSON

Aiming for a more efficient and sustainable message transfer, adjustments on the OCPP 1.5 were conducted and JSON was integrated as an alternative messaging protocol approach. Next to the performance upgrades, in terms of OCPP 1.6 JSON, 35 novel use cases were introduced for more complete management and manipulation of charging points and their communication with a central management system [18].

The following listing illustrates an example OCPP 1.6 JSON boot message [15].

```
[3 , "37278218" ,
{
" status ": "Accepted" ,
"currentTime " : " 14:34:13.482Z" ,
"heartbeatInterval": 250
} ]
```
OCPP 1.6 bootNotification JSON

The size of the messages is decreased by 15 percent if compared to an equivalent XML message (OCPP 1.2) on this specific listing. Depending on the data being transferred, the size reduction may be even higher. This is especially relevant when the hardware resources are limited, and the overall performance may be negatively affected.

OCPP 2.0 and the introduction of Smart Charging

Next to the benefits of the quicker messaging and data transfer of the OCPP 2.0 JSON, the smart charging solution provided by the protocol is an extraordinary further functionality. OCPP can set limitations and rules to the amount of power being delivered during a charging session. An intelligent charging approach optimizes the charging session of one device attached to an EV and may also manipulate and control a whole group of charging points.

Smart Charging can also be applied on an infrastructural level where the power consumption of all charging points managed by the central server will be regulated and limited to match specific criteria. For instance, the current power available comes from a source of renewable energy sources, such as photovoltaic panels on the roof of the building. For this exact purpose, constant communication between all single charging points and the central server is necessary, where the difference in the messages' sizes can be critical, mainly when having limited resources.

Chapter 3

Design and Components

Based on the research findings in the related work chapter, this chapter will focus on the design of a tailored hardware infrastructure aiming at a fully automated, hands-free, and shared usage of a portable low-power EV charging device. The hardware infrastructure will rely on a central management system that will handle user requests and communication between the hardware components.

3.1 Hardware Components

All of the individual hardware components described in the following subsections be replaced by similar devices. This project is meant to be a device-independent solution where all of the single components could potentially be replaced by equivalent ones.

3.1.1 Portable Charger

This solution is considering only low-power EV charging devices. This implies that we will take only into consideration charging devices that support a maximum electric power of 3.6 kW (16 A). These low-power devices can be plugged into any domestic socket in Switzerland.

We have taken use of a low-power portable charging device of the branch Morec[19].

The device properties will be specified in the following segment.

- Plug, Socket-Outlet: IEC 62196-2
- Connector Type: Type 2 connector
- Supported Electric Current: 10/16 A
- Delivered Power: 2,2/3,6 kW

Figure 3.1: Morec portable home 3.6kW EV charger

- Input Voltage: 240 V
- Power Source: Alternating Current (AC)
- Cable Length: 7.5 meters
- Material: Acrylonitrile-Butadiene-Styrol

The portable device has security mechanisms integrated that prevent it from overheating and any damage to the attached EV.

The role and functionality of the charging device will not be altered in any way. Its basic functionality will be coupled with smartly designed components to create a hardware environment that can be managed by a central management system.

3.1.2 Switch

As for the remote triggering and stopping of the electric current within the environment, we will use a smart switch from the brand Gravity [20]. The switch will turn on and turn off and respectively allow or block the electric circuit entering the portable charger by remote manipulation from the RPI.

The device properties will be specified in the following segment.

- Operating Voltage: 3 Volt (V) to 5 V Direct current (DC)
- Logic Level Voltage: 3 V to 5 V DC
- Control Signal: Transistor-Transistor Logic (TTL) level

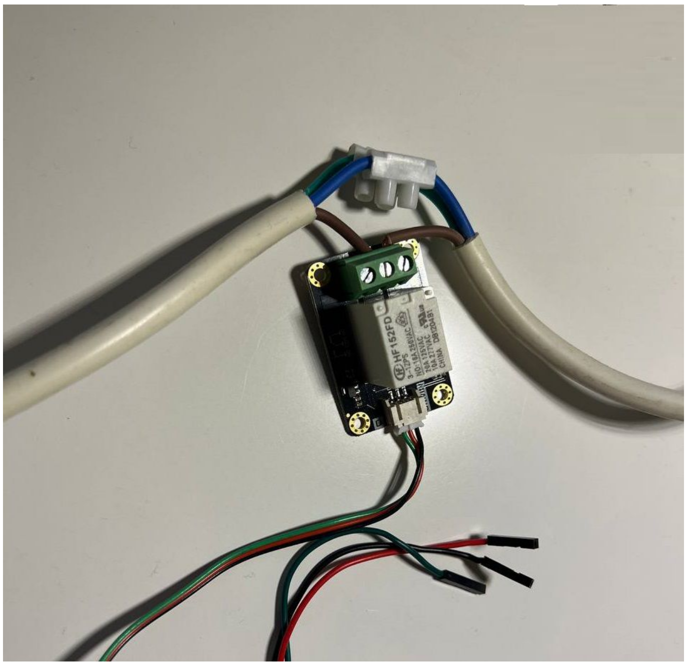

Figure 3.2: Gravity smart switch

- Maximal Voltage: 400 V AC
- Maximal Current: 16 A
- Time Switching: 10 milliseconds (ms)
- Interface: Gravity Digital
- Operation: Via 3.3-5 V IO-port

The switch is directly integrated into the circuit by coupling it to the electric current by attaching it to the external/outer conductor (L1). On the other side, the communication with the Raspberry Pi 4 (RPI) [21], which will act as the motherboard, will be directly capacitated through the RPIs pins. By turning the RPIs pins on and off, the 3.3 V signal will be directed toward the smart switch that will react to the signal by adapting its physical state (on or off).

Low-power signals are sent toward the switch by using an arbitrary general-purpose input/output (GPIO) pin. In this setup, we use GPIO pin 21. The smart switch has transistors integrated which will amplify the signal sent by the RPI (3.3 V) in order to physically change the state of the switch and thus allow the electric current to flow into the circuit.

In the following listing, we will illustrate how to manipulate and trigger the GPIO pin's state change.

```
import RPi.GPIO as GPIO
import time
```

```
# BMC Broadcomings, BOARD boards
GPIO.setmode(GPIO.BCM)

# Relay 1 initialize the pins to use
GPIO.setup(21, GPIO.OUT)

# run all code and clean up

try:
    # Turns on the pin 21 on the RPI
    GPIO.output(21, GPIO.HIGH)

    #This print statement will provide us with feedback regarding if the
        status of the pin.
    print('The GPIO pin is on')

    # This will pause the code for 5 seconds and will avoid unwanted damages
        in the switch
    time.sleep(5)

    # Turns off the pin 21 on the RPI
    GPIO.output(21, GPIO.LOW)

    #This print statement will provide us with feedback regarding if the
        status of the pin.
    print('The GPIO pin is off')

finally:
    # To clean up all the used GPIO pins/ports
    GPIO.cleanup()
```

For feedback purposes, we are printing the output in the console, which if the pin was able to run on successfully would notify with ”The GPIO pin is on”. After turning the pin on, the RPI will send a 3.3 V signal to the switch, which will allow the electric current to flow into the EV and thus activate the charging procedure.

After the pin and consequently the switch was triggered, a delay of 5 seconds was implemented. This would prevent any damage to the switch due to constant signals coming from the backend in case of incorrect implementation. After the delay, the switch will turn off again by setting the GPIO pin 21 on the RPI off. Here once again, we have printed the output due to feedback purposes.

3.1.3 Digital Power Meter

A single-phase digital power meter of the branch ORNO [22] will be used to measure the power that is flowing toward our environment and consequently flowing into the attached

EV if the charging device is plugged in. The power measurements digital power meter fulfill and are responsible for one of the most critical aspects of this project, the cost tracking of each independent e-driver.

Power measurements in kW will be further translated to energy (kWh) in the central server system.

In this project, we use a single-phase ORNO 515 digital power meter that supports RS485 Modbus RTU Protocol communication for reading data and setting parameters remotely from the central management system.

The digital power meter will contribute to our purposes by facilitating the following use cases:

- Forward and reverse energy measurement (bi-directional measurement of the active and reactive energy, as well as the voltage and current.
- Read the network parameters, analyze the power quality and load condition
- RS485 Modbus RTU Protocol communication
- Physical LED screen directly built-in into the meter that serves as the further feedback indicator

The device properties will be specified in the following segment.

- Power supply : 1x230 V AC, 50/60 Hz
- Display: LCD (number of digits 5 + 1 = 00000.0 kWh)
- Pulse frequency: 1000 imp / kWh
- Current: minimal current supported 0.25A / maximal current supported 100A
- Degree of protection: IP20
- Kind of output: RS485 Modbus RTU
- Communication protocol: Modbus RTU
- Type of counted signal: Electric power

The meter will communicate with the RPI via a communication protocol (Modbus). This will be possible by using a USB 2.0 - RS485 Converter [23], which will be inserted into the RPI and attached to the meter.

Two critical aspects are missing in this digital power meter.

- The device is not able to measure the total power consumed over a time interval. It can only measure the current power consumption.

- The device provides power measurements, not energy measurements. The power measurements must be translated into energy units as a further step in the central management system.

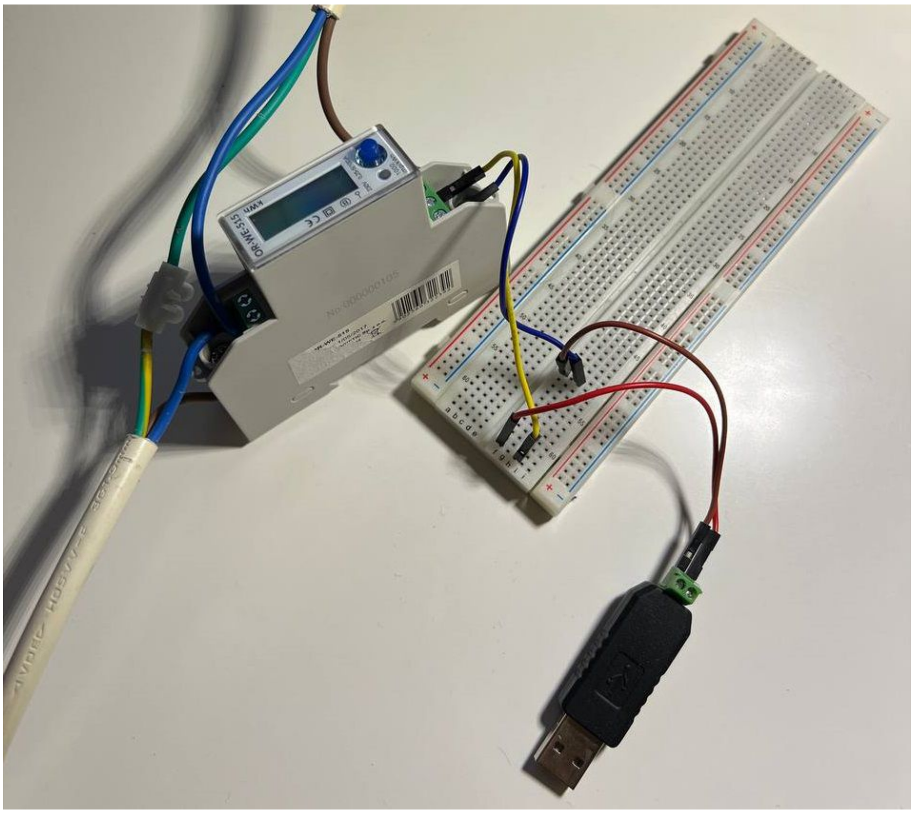

Figure 3.3: ORNO 515 digital power meter and RS485 to USB 2.0 converter communicating through jumping cables via a breadboard

3.1.4 Esp Eye

The Esp Eye [24] (ESP32) runs on Espressif Systems which is a low-cost, low-power system on SoC (System On Chip) chip that supports WIFI. Includes clock tuning with fine resolution, power modes multiple, and dynamic power scaling. The purpose of the Esp Eye is to support the ANPR modules in this project by capturing high-quality snapshots of the plates of approaching vehicles.

In addition to a 2-megapixel camera and a microphone, the Esp Eye board is equipped with an ESP32 chip, 8 MB PSRAM, and 4 MB Flash.

Esp Eye is used in conjunction with the ESP-WROVER-KIT (or other ESP32 cards) to add eyes and ears to your IoT applications. The Esp Eye can be used as a development board, ESP-WROVER-KIT, or other ESP32-based development boards. By adding just a few peripherals, such as B. cameras and screens, you can then easily create complete IoT applications.

All the configurations and setup on the Esp Eye have been done by using Arduino using the management package Arduino-esp32 development kit.

All the images taken for the purpose of the project will be configured as follows:

The configuration used for the Esp Eye in this project:

Figure 3.4: ESP32 ESP-EYE IoT Module

- Resolution of SXGA (1280x1024)
- H-Mirror effect
- V-Flip effect

3.1.5 Raspberry Pi 4

As a motherboard aiming to handle all the communication and data storage, we will use an RPI 4.

The RPI consists of a motherboard that supports different components of a computer such as an ARM processor, a graphics chip, and RAM memory of 4 GB. Thanks to its ports and inputs, it allows you to connect peripheral devices. For example, the digital power meter, the smart switch, and the Esp Eye.

The RPI is able to connect to a network through the Ethernet port, Wi-Fi and Bluetooth.

The central management server will be running on the RPI which will at the same time act as a communication interface for the smart switch (by using a GPIO pin), the digital meter (along with an RS485 USB 2.0 converter), and the Esp Eye.

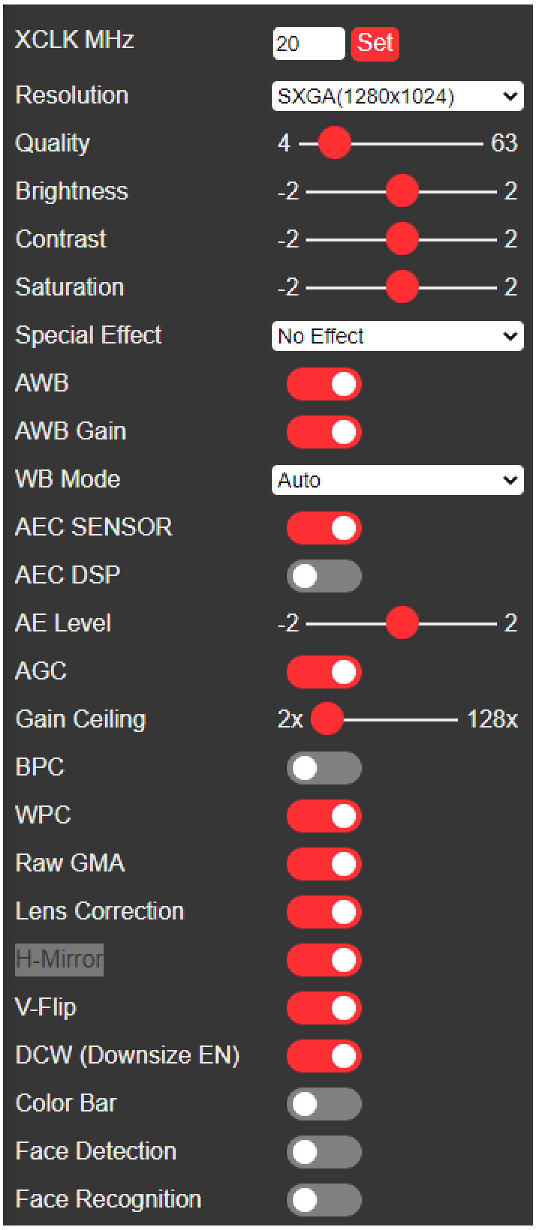

Figure 3.5: Configuration of the Esp Eye

The device properties will be specified in the following segment.

- Processor: Broadcom BCM2711, quad-core Cortex-A72 (ARM v8), 64-bit SoC @ 1.5GHz
- Memory: 4GB
- Connectivity: 2.4 GHz and 5.0 GHz IEEE 802.11, bg, n, ac wireless, LAN, Bluetooth 5.0, BLE, Gigabit Ethernet, 2 x USB 3.0 ports, 2 x USB 2.0 ports
- GPIO: Standard 40-pin GPIO header, (fully backward-compatible with previous boards)

- Video and Sound:2 x micro HDMI ports (up to 4Kp60 supported), 2-lane MIPI DSI display port, 2-lane MIPI CSI camera port, 4-pole stereo audio, and composite video port

- Multimedia: H.265 (4Kp60 decode), H.264 (1080p60 decode, 1080p30 encode), OpenGL ES, 3.0 graphics

- SD card support: Micro SD card slot for loading operating system and further data storage

- Input power: 5V DC via USB-C connector, 5V DC via GPIO header

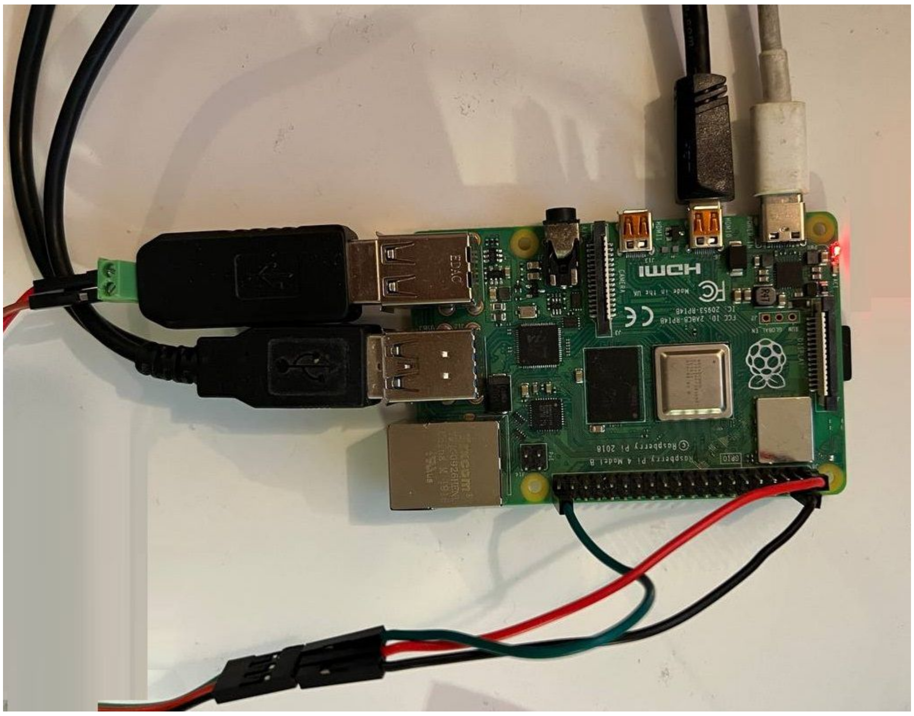

Figure 3.6: Hardware setup and communication of the Raspberry PI 4

The next figure illustrates the integration of the RPI along with the hardware components described in the sections above.

- USB 2.0 to RS485 Converter

- 3.3V Power Pin, 5V Power Pin, and the GPIO 21

- The Keyboard and Mouse USB 2.0

- The Micro HDMI Cable

- The Power Cable

Other necessary items

For the purpose of communication between the RPI and the smart switch, jumping cables have been used along with a breadboard.

For the communication between the RPI and the meter, a USB 2.0 RS485 converter has been used along with a breadboard.

Further cable extensions, plugs, and sockets have been attached to the system for the purpose of comfort.

3.2 Software

The hardware components will rely on a software solution that will:

1. Enable and handle the communication with the hardware components and the motherboard

2. Measurement of the consumption data for all users

3. Handling of the plate recognition process

4. Handling of the user requests from the web frontend

The software implementation is illustrated in the following figure.

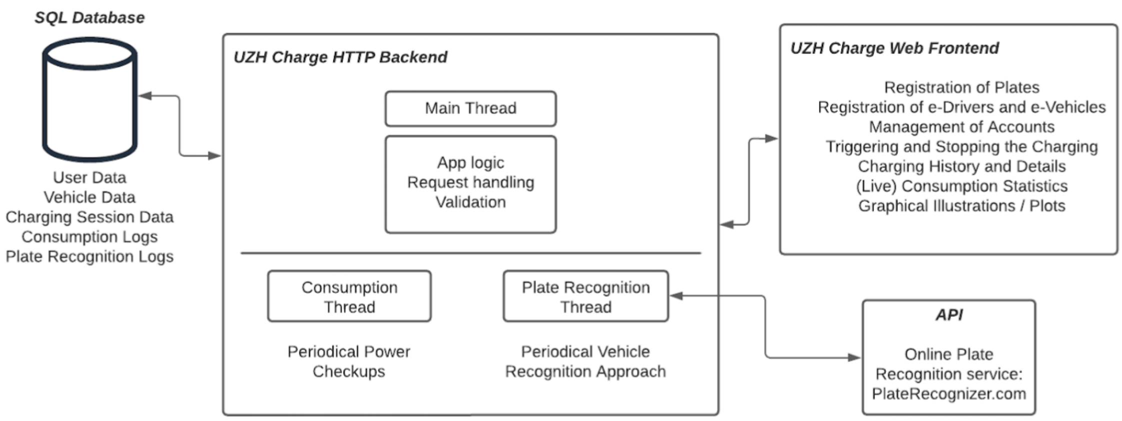

Figure 3.7: Software components

3.2.1 Central Management System

The backend comprises three major components that interact together and handle the three different essential aspects of this project. We will separately illustrate the composition of these three components.

1. First thread: The request validations, user management, and overall interaction from the e-drivers and administrators with the front end.
2. Second thread: The measurement of the power consumption.
3. Third thread: The ANPR process will constantly run in the background.

Traditional Backend - Request handling

The first one is the typical backend aiming to validate the incoming requests from the front end. Two significant roles are implemented - the administrator and the user (e-driver). The administrator role carries responsibility for the following actions and can retrieve the following data:

- Registration of users
- Registration of vehicles for users
- Retrieve charging sessions of all e-drivers
- Retrieve consumption statistics on all charging procedures
- Retrieve output logs of the plate recognition process

On the other hand, the e-driver has access to the following actions and data:

- Account registration and account management
- Retrieve (own) last charging session statistics
- Current power consumption (while charging)
- Trigger charging procedure by button click in the web frontend
- Trigger charging procedure by approaching with a vehicle that is registered along with valid vehicle plate numbers.

3.2.2 Automatic number-plate recognition

Aiming for a fully automatized hands-free approach we will consider three ANPR solutions for this project. (1) UZH-ANPR, (2) the open source OpenALPR [25], and a paid service named (3) Plate Recognizer [26].

The purpose of using an ANPR is to avoid any physical keys or internet-dependent devices from the part of the e-drivers to trigger the charging process.

The first two of the three mentioned approaches are fully open source. The first one is a composition of different open-source components that have shown the best accuracy regarding license plate recognition. The second solution is a complete open-source solution, where an image will be provided and a string containing the license plates is returned. No chances, improvement, or alterations have been conducted to the tool.

The third solution for ANPR is a paid online service called Plate Recognizer. For our purpose, images of vehicles have been uploaded to the online tool and as a result, a string containing the license plates has been returned.

3.2.3 UZH-ANPR

UZH-ANPS is built based on the various research and solutions proposed online. [27].

UZH-ANPR relies on the following mechanism for license plate recognition:

- Image acquisition
- Image Color conversion from RGB to gray level

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

The previous listing illustrates the process of conversion of an RGB image into greyscale using OPENCV. The variable *img* represents the most recent snapshot captured by the Esp Eye.

1. Image noise removal and smoothing by using the Gaussian algorithm

   ```
   bfilter = cv2.bilateralFilter(gray, 11, 17, 17) #Noise reduction
   ```

   The previous listing illustrates the bilateral filter for smoothing images and reducing their noise. The parameter (11, 17, 17) specifies how intense the noise reduction would be. The edges, however, will preserve. The variable gray represents the output of the previous image processing mechanism.

2. Image edge detection

   The canny algorithm enables the detection of edges.

```
edged = cv2.Canny(bfilter, 30, 200) #Edge detection
```

The result of the edge detection mechanism is illustrated in the following figure whereas *bfilter* is the output of the previous noise reduction procedure.

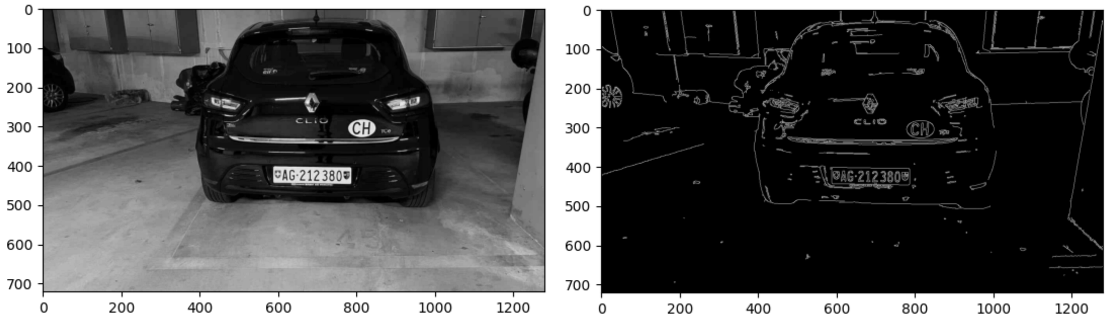

Figure 3.8: Canny method on OpenCV for the edge detection of an image.

This will help identify edges in the picture and potentially make it easier to recognize the number plate.

3. Image edge detection

   Detecting polygons within the images ideally shapes within the images. We are looking for a contour containing four edges, which will potentially be the number plate.

```
keypoints = cv2.findContours(edged(), cv2.RETR_TREE,
    cv2.CHAIN_APPROX_SIMPLE)
contours = imutils.grab_contours(keypoints)
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]
```

   The function under the variable key points will find the contours in the edged image. The two arguments, first one is how our result will be returned, in this case, as a tree which will allow us to traverse the tree to find different levels of contours. The second argument is how the results would be returned. The approx simple algorithm will approximate the results for overall contour recognition instead of detailed contour identification, which would return too many particularized contours.

   The second variable will grab the contours, which will be sorted in the last command. The top 10 contours will be then returned.

4. Finding contours in the image

   Next, we will iterate through our edged image's top 10 contour results.

```
location = None
for contour in contours:
    approx = cv2.approxPolyDP(contour, 15, True)
    if len(approx) == 4:
        location = approx
```

```
        break
```

The higher the approxpolydp argument is set, the rougher and less strict the approximation or accuracy grain will be. If the number is high, the result will be rounded. If our approximation is four key points, this could be the license plate since license plates in Switzerland always consist of 4 key points, which means that these are rectangles.

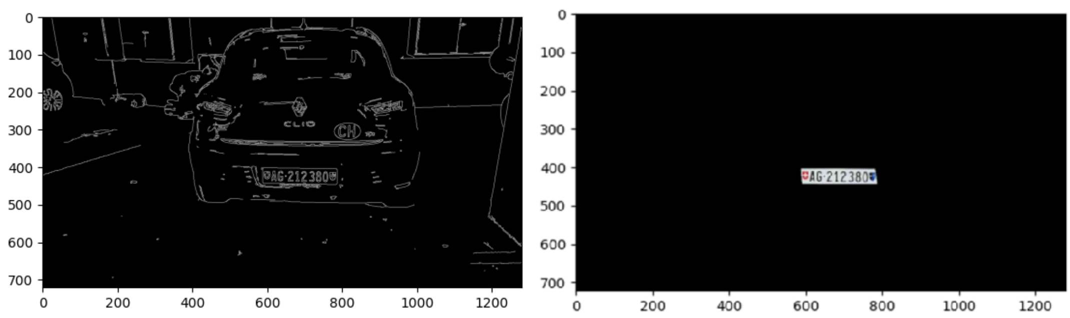

Figure 3.9: Apply the approxPolyDP for the contour approximation during the license plate recognition process.

5. Masking the image and extracting the license plate section

   After applying a successful contour search to identify where exactly the license plate is located, we will be isolating that exact section.

```
mask = np.zeros(gray.shape, np.uint8)
new_image = cv2.drawContours(mask, [location], 0,255, -1)
new_image = cv2.bitwise_and(img, img, mask=mask)
```

   np.zeros creates a blank mask with the exact size of our original gray image. Then we passed the shape of the original images and specified what we would like to fill in, i.e., blank zeros in this case. We then draw contours in the license plate location.

   Overlay that masks over our original image by the cv bitwise *and* operation allows returning the segment of the image that represents the number plate.

   For the purpose of isolating the section where the plate is located, we will be cropping the image by using further OpenCV methods.

```
(x,y) = np.where(mask==255)
(x1, y1) = (np.min(x), np.min(y))
(x2, y2) = (np.max(x), np.max(y))
cropped_image = gray[x1:x2+1, y1:y2+1]
```

   This will make it easier for optical character recognition to perform. The cropped image represents the license plate.

   Once we have the cropped image containing the license plates, the process of character segmentation can start.

Figure 3.10: Result of applying the extraction of the contour of the license plate.

6. Character Segmentation and Optical character recognition

   For the sake of reading the tag out of the cropped image containing the license plate, we have used the open-source library Tesseract.

   ```
   if isinstance(pytesseract.image_to_string(result), str):
   ```

   The reader is composed of a single method called image to string which will return a string value.

7. Transformation of the string into the format of a Swiss vehicle plate number

   A problem arose during the process of the image-to-string transformation. Often the method returned correct values, however, the formatting was wrong. To deal with this issue, we have run the algorithm 60 times by using 60 different plates and we have collected the most common formatting issues and handled them one by one. E.g. AG-397 209 which should be interpreted as AG 397209.

8. Verification of the vehicle plate number in the database

   As soon as the correct formatted string was generated, we would double-check if the license number was registered in the database under validated users to continue with the charging process.

3.2.4 OpenALPR

OpenALPR is an open-source Automatic License Plate Recognition library that is written in C++ [25].

In this project, we will be using the bindings of the library in Python. In the context of this project, OpenALPR will analyze a set of image captures of vehicles that approach the charging point.

The output provided by OpenALPR is a string representation of the license plates of the approached vehicle.

3.2.5 Plate Recognizer

As our third plate recognition approach, we will be using Plate Recognizer. Plate Recognizer is a paid online service that receives an image as input through their REST APIs and returns a string representation of the license plates of the vehicle in the image (if any vehicle and license plates are found).

3.3 Main Architecture

3.3.1 Process Flow

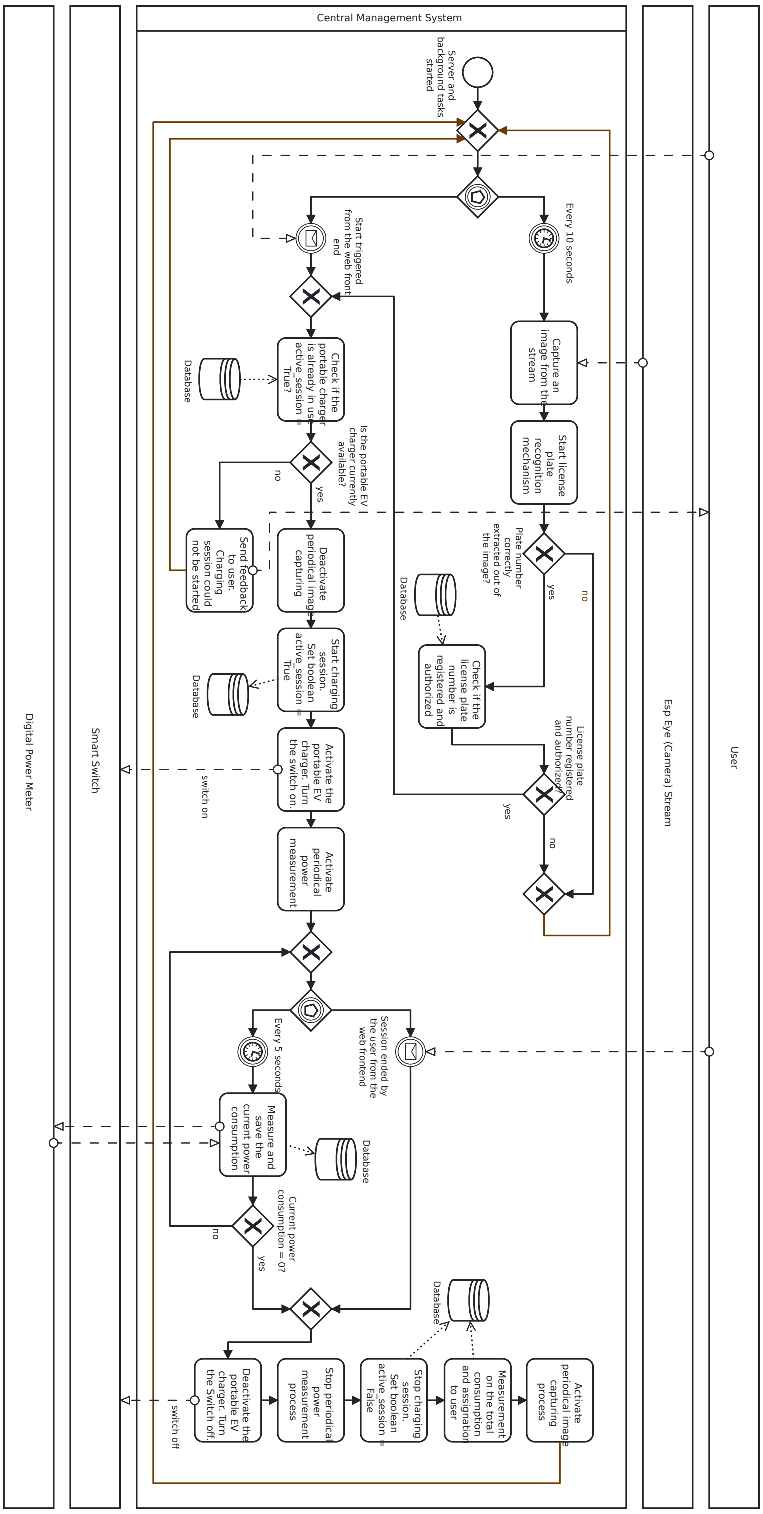

Figure 3.11: The BPMN 2.0 process flow diagram illustrates the user interaction with the solution and the management and proper manipulation of the hardware devices

Initializing a charging session

Once the server and all relevant processes are activated, one of two conditions must occur to trigger a response from the central management server and consequently, activate or deny the initialization of a new charging session.

A charging session is considered to be active once the flow reaches the task ”Deactivate periodical image capturing”. This implies that one of the previously described conditions has been fulfilled and there is no active charging session currently registered.

- Condition 1: A registered user triggers a new charging session through the web front end by a button click Only registered users can initiate a charging session. This implies that the user must have an account and be currently logged in. Once the user triggers the activation of the charging session, the central management system will check if there are active charging sessions already.
- Condition 2: Registered and authorized license plates are recognized by the license plate recognition process. The esp eye is continuously streaming on the local network. The esp eye does not directly communicate with the central management. A dedicated thread in the central management system is responsible for periodically capturing images from the stream. These images will be further analyzed and if a license plate is recognized, a string containing the license plates will be further validated, and consequently, a charging session (corresponding to the user under whom the license plate is registered) will be initialized or denied.

  If the charging session could be initialized, the periodical image-capturing process is stopped and the periodical power measurement is enhanced (Task: ”Activate periodical power measurement”). Furthermore, once a charging session is initialized, a flag will be set to represent that there is an active charging session and consequently deny any further requests for new charging sessions on that charging station.

Power Measurement

The central management system periodically triggers a response from the digital meter containing information regarding the current power consumption of the charging session. The power measurement takes place in form of a loop meant to continuously measure and store the power consumption data in the database.

The power measurement never reaches zero, since devices such as the digital power meter and the (light-emitting diode) LED display on the portable charger are continuously absorbing a small percentage of the power available.

The power measurement loop also includes mechanisms to ensure the accuracy and integrity of the data being collected. The collected data is then analyzed and transformed into energy values. Furthermore, the current power consumption can be reviewed anytime in the web front end, allowing monitoring and traceability to ensure safety.

Finalizing a charging session

The power measurement loop can be only exited, and consequently, the charging session is finalized if one of the following conditions takes place:

- Exit Condition 1: The user ends the charging session from the web front end with a button click
- Exit Condition 2: The power measurement indicates a power consumption of near 0 (under a predefined threshold).

Once one of these conditions took place, the charging session will be categorized as finalized and the following tasks will be conducted:

1. A signal is sent to the smart switch to turn off the electricity flow. This will consequently stop the charging of the vehicle since no current can be passed into the portable charger
2. The power measurement loop is stopped and periodically power consumption check-ups are deactivated
3. The total amount of power and energy consumption will be assigned to the user along with the total charging time (in minutes)
4. The license plate recognition thread is reactivated, and the central management system will initialize the periodical image capturing out of the video stream of the Esp Eye
5. The system is reset to the original state, enabling the request handling and initialization of new charging sessions

The practical challenge of the Exit Condition 2

A crucial challenge of the actual system occurs once a vehicle is fully charged. Once a vehicle is fully charged, the power measurement will indicate a power consumption of near zero.

Once the power consumption decreases under the specified threshold (< 0.1 kW), the current vehicle charging is set to fully charged. This will trigger the finalization of the charging session in order to reactivate the license plate recognition process.

The challenge is that if the vehicle remains parked in the e-vehicle charging spot, the camera will continuously identify the vehicle's license numbers and will consequently trigger a new charging session.

The new charging session however will be stopped immediately, since the vehicle is fully charged and will not absolve any energy from the source. Meaning that the power consumption will be under the specified threshold (since the vehicle is fully charged) and the charging process will terminate. This implies that numerous charging sessions have been initialized and terminated after the first power measurement. This creates a continuous loop of activating charging sessions for a vehicle that has already fully charged.

Chapter 4

Implamentation and Evaluation

Based on the research findings in the related work chapter and the architectural design proposed in the design and components chapter, this chapter will focus on the conception and implementation of the communication between the hardware components with the RPI and the implementation of the central management system that will use a tailored communication protocol to handle a low-power EV charging device.

For this purpose, the challenge described in the introduction chapter is written in form of use cases in the following section.

As an e-driver who wants to charge an EV, I want to be able to:

- Rely on a low-power domestic charging point (3.6 kW).
- Ensure a shared approach where multiple e-drivers can use the devices.
- Ensure accurate accountability and cost tracking for all of my charging sessions for all the e-drivers.
- Start the charging process in an automated way without having to use any app or physical key.
- Start the charging process as a verified e-driver without having access to the internet and consequently log in to an application or web interface.
- Check my charging statistics at any time using a web interface.

4.1 Hardware Setup

The final implementation of the hardware setup after the installation of all elements is displayed in the following figure.

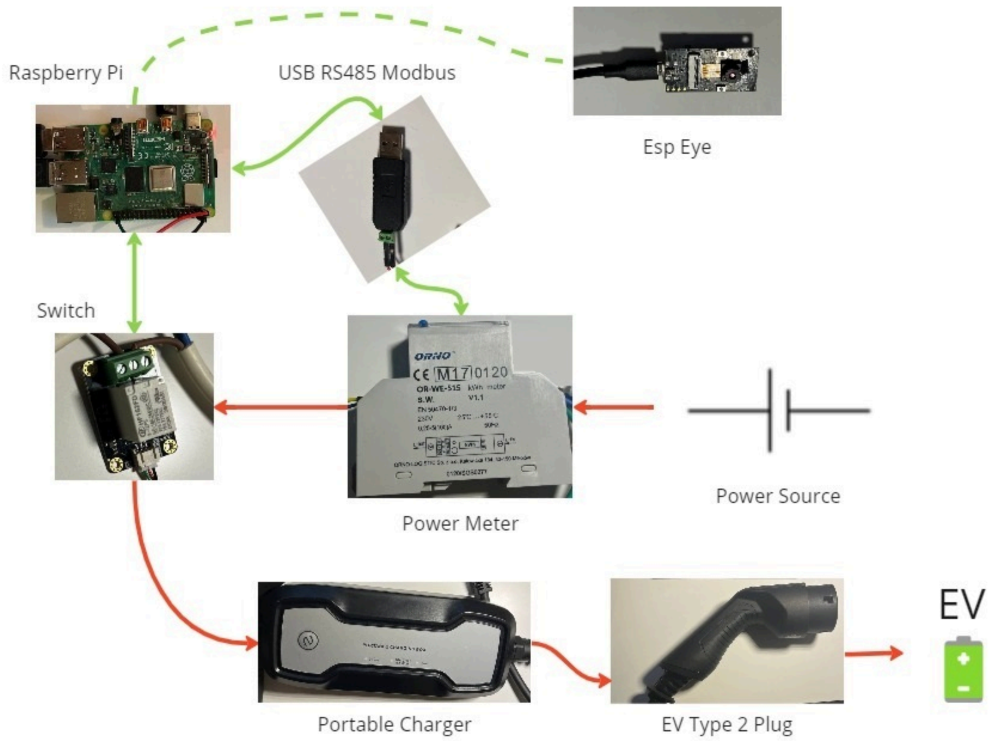

Figure 4.1: Hardware setup for an automated and hands-free low-power charging of EVs.

4.1.1 Communication and Data Flow

The communication between the devices is illustrated by the green lines. All the communication is handled by and through the RPI, where our central management system is running.

Ditigal Power Meter

The first communication component can be found between the power meter and the central server. The communication is facilitated by a USB 2.0 RS485 converter, which is directly plugged into the RPI. The converter will translate the two-wire RS485 signal-supporting digital meter to a USB 2.0 port which is supported by the RPI. The logic and the connectivity between both devices are facilitated by the Modbus data communication protocol.

Basically, the RPI sends a signal from the central management server every 30 seconds to trigger a message from the digital meter. The response from the digital meter is originally in hexadecimal form and will be translated into float values in the server. The final float value represents a power measurement in kW and will be stored in the database along with user meta information and timestamps.

Smart Switch

The second communication component occurs between the switch and the central server. The communication is directly handled by the RPIs GPIO pins. By turning the RPIs pins on and off, the 3.3 V signal will be directed toward the smart switch that will react to the signal by adapting its physical state (on or off). The RPI receives a signal from the central server system that will trigger the physical switching of the current state physical state of the switch.

ESP-EYE

The third communication component, illustrated by a dotted line, corresponds to the esp-eye. The esp-eye however will not directly communicate with the central management server nor send or receive any data to the backend, as for the previously described components. The esp-eye will be constantly streaming to a specific internet address. The RPI sends periodically a GET request to that specific address triggering a snapshot that will be saved temporarily in a specific folder locally in the RPI as shown in the following listing.

```
# The variable URL contains a defined address where the esp-eye is
    continuously streaming
url= 'http://172.20.10.5'

# The variable r sends a get request to the predefined endpoint provided by
    the esp-eye, the id of the image will be its timestamp, whereas the
    variable time, stands for the current time.
r = requests.get(f'{url}/capture?id={int(round(time.time()*1000))}',
    stream=True)

with open(output_path, 'wb') as out_file:
    shutil.copyfileobj(r.raw, out_file)
```

This is possible since both, the RPI and the esp-eye are running in the same network. Once the image processing is concluded and the string with the license plate is acquired, the image is removed.

4.1.2 Electric Current

The origin of the electric current, illustrated by red lines is a domestic power source of typically 3.6 kW. As a first step in the sequence, the current flows into the digital meter where the power of the current will be periodically measured and further converted into energy terms in the central server system.

As s second step, the electric current will flow towards the smart switch. The electric current will then only pass through the switch if the switch is turned on, which means the switch is physically set up in a state where the electric current is permitted to flow toward the next component, the portable charger. The physical state of the switch will be managed by the RPI.

If the switch is activated, the electric current will arrive at its final destination, the portable low-power charging device. The device will be plugged into an EV by using a type 2 connector which will consequently charge.

The portable low-power charger has built-in approaches to ensure safe charging and to prevent the EV and the charger itself from any damage. This means that the electric current will only arrive at the EV if no unexpected circumstances are given, for instance, if the device is overheating or not plugged into the EV correctly.

4.2 Key Behaviours

4.2.1 Trigger the charging procedure

The right triggering and functionality of the switch are key to the success of this project. It is not only to be able to trigger the switch in the desired way but also to trace the behavioral patterns and store them in a database aiming for proper log mechanisms.

The switch will be activated whenever a validated user triggers the charging process by approaching a vehicle with valid plate numbers or manually pressing the button "start charging" in the front end.

The switch is triggered from the RPI by using specified GPIOs that provide a 3V signal to our switch. The switch relies on built-in transistors that will amplify the received electric current into a stronger signal to change the physical state of the switch.

4.2.2 End the charging procedure

Our work proposes a mechanism to accurately identify when a charging session should be stopped. Commercial charging stations that rely on standardized communication protocols provide approaches to physically detect whenever the car is plugged in or charging. To replicate this behavior, we propose an alternative approach that could potentially involve a series of conflicts and error margins. Our approach relies on the consumed power during a specific time interval of 20 seconds.

If the periodical measurement of the power indicates that the actual power consumption of an active charging session has decreased to near zero (0 kW = no consumption), a timer will be triggered. The actual power consumption will never reach zero since the devices continuously consume a small amount of energy (for instance the LED display of the portable EV charger).

The timer will continuously check the power consumption values over a period of 20 seconds. If the consumption does not increase during the time-lapse, the charging session will be flagged as finished. The charging device will turn off, the user will be logged out of the charging session and the ANPR mechanism will be triggered.

Potential problems:

- What if the current of the building is off for more than 20 seconds?
- What if the e-driver pulls out the cable on purpose and plugs it again after 20 seconds?
- What if the e-drivers quickly move the car and another vehicle approaches within the time-lapse and charges under the previously logged-in account?

4.2.3 Measurement and cost accountability

The Single-Phase Digital Power Meter will serve the purpose of power measurement. Power will translate in the backend into terms of energy (kWh). Power values measured in kW can be translated into energy terms in kWh by providing a time specification by using the following formula:

$$E_{(kWh)} = P_{(kW)} \times T_{(hrs)}$$

Figure 4.2: Conversion of power measurements into energy

The Digital power meter will periodically measure the power that is flowing through the circuit into the EV. The data will be sent to the backend where it will be processed and translated into energy terms and saved in the database along with timestamps.

The digital meter uses Modbus RTU as a communication protocol. Our motherboard - the RPI4 - does not provide any options for direct communication with slave devices using Modbus RTU. To enable the communication we have relied on an RS485 to USB converter [23].

The following code snipped will illustrate who the digital power meter is communicating through the RS485 USB with the backend. The output expected is a single string value in kW of the current power that is being pushed through our circuit.

```
import pymodbus
from pymodbus.pdu import ModbusRequest
from pymodbus.client import ModbusSerialClient as ModbusClient
from pymodbus.transaction import ModbusRtuFramer

#Setup and initialize the Modbus client by specifying the communication
   protocol, the port where the USB485 is found, and the required exchange
   requirements.
```

```
client = ModbusClient(method = 'rtu', port='/dev/ttyUSB0', stopbits = 1,
    bytesize = 8, parity = 'E' , baudrate= 9600)

#Connect to the serial Modbus server
connection = client.connect()

#Starting add num of reg to read, slave unit.
coil = client.read_holding_registers(0x0140,2,slave=1)# address, count, slave
    address

#Capturing and saving the power values (in kW) into a variable
consumption = coil.getRegister(t)

#Closes the underlying socket connection
client.close()
```

An important trade-off comes up when it comes to measurement intervals. An interval of 30 seconds has been chosen for the sake of accuracy.

4.2.4 Hands-free Automated Charging

The backend will trigger the esp eye to capture periodically images and save them in a given direction in the RPI. The images however will be analyzed and put into the ANRP algorithm. If any vehicle was recognized the backend will delete automatically the previously captured image. If a vehicle was recognized and the charging procedure could be successfully triggered (meaning that the vehicle is registered in the system) the image will be saved in the database as proof for the future accountability of the costs related to that specific charging session. The stored image however will not contain anything other than the plate numbers due to potential data privacy issues.

4.3 Software Implementation

4.3.1 First thread Backend

Request handling, and validations.

4.3.2 Plate recognition thread

An esp eye will be positioned at the parking slot. The esp eye is responsible to take pictures periodically and send them to the backend. The backend will then:

- Recognize vehicles

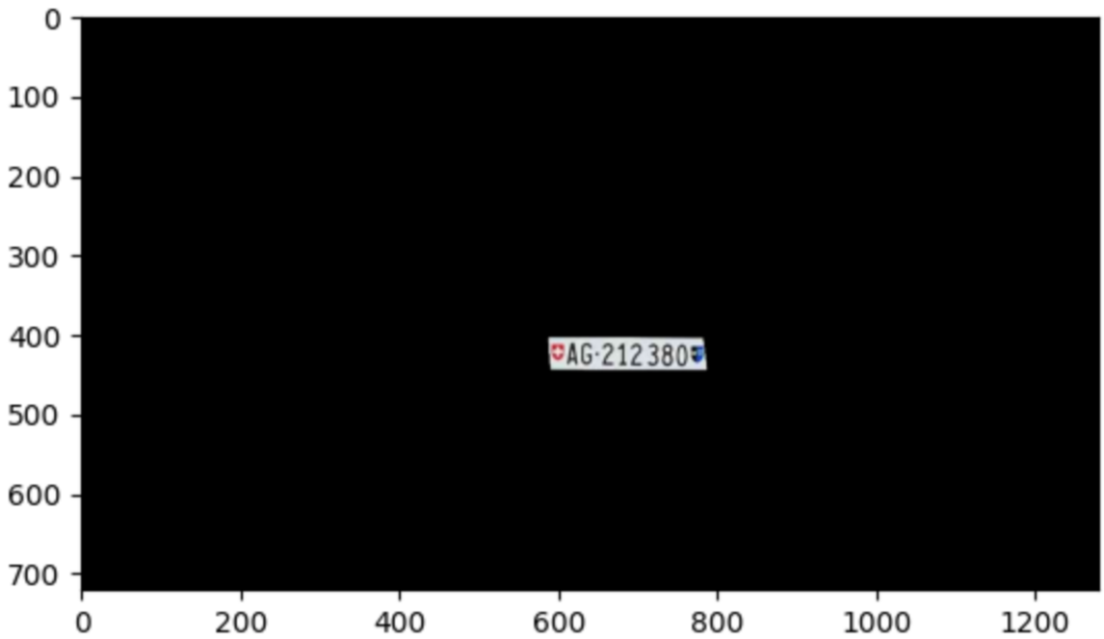

Figure 4.3: Example of an image that would be stored in the central management system's database for protocoling purposes.

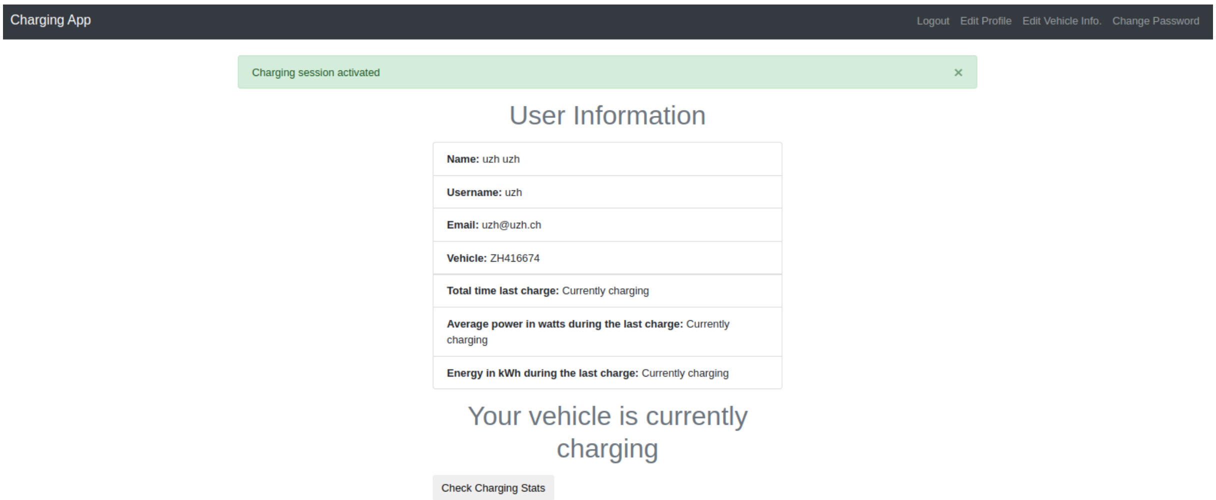

Figure 4.4: The web front displays the current status of the charging vehicle.

- Recognize plates
- Recognize string

- Check if the user is registered and validated
- Activate the switch through the GPIO

As soon as the previously recognized plate is not more perceived, the switch will turn off.

1. Own approach
2. Open source approach
3. Paid API approach

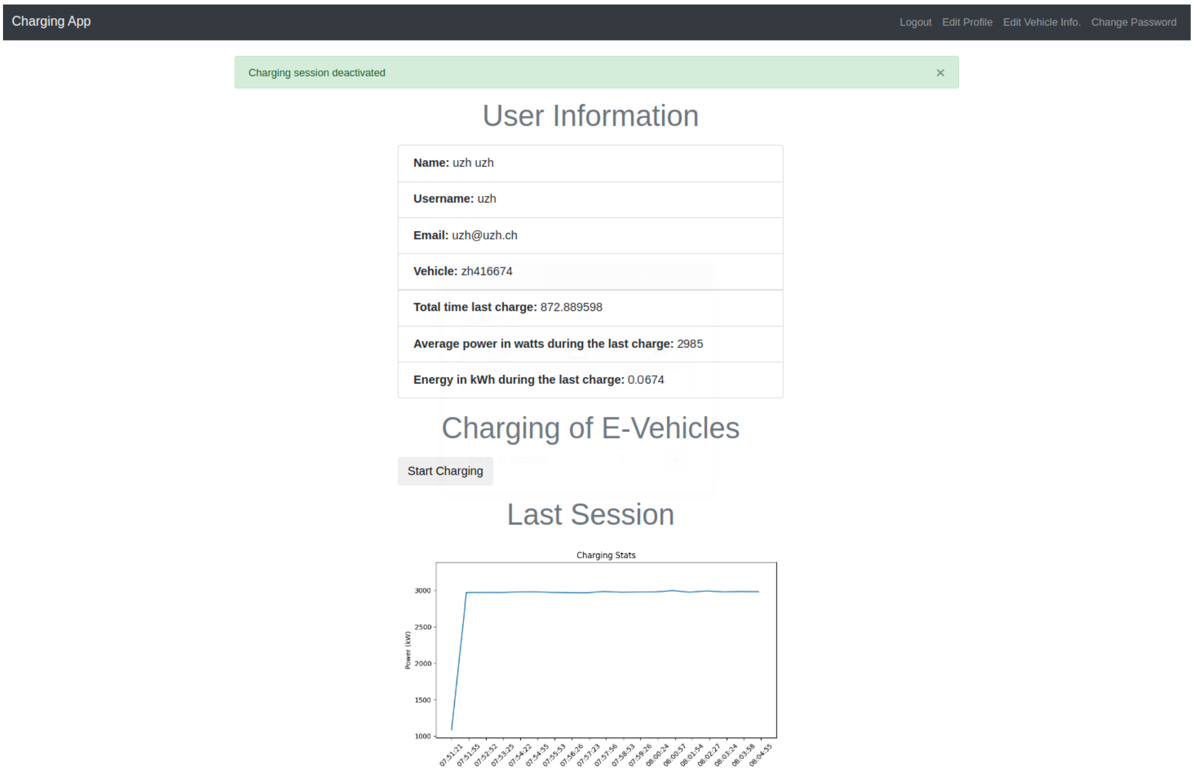

Figure 4.5: User and vehicle metadata along with measurement and a summary of the last charging session.

4.3.3 Consumption thread

The consumption thread is designed with the purpose of saving charging data. Users will be able to check their personal statistics based on this approach.

The backend task will check for the current power measurement every 30 seconds. The data captured during these periodical intervals will be saved into the database along with a timestamp.

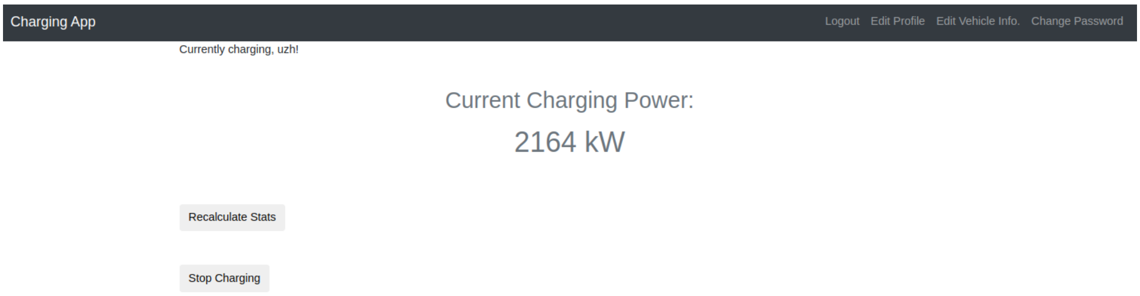

Figure 4.6: Current charging power from the charging point towards the EV.

4.3.4 Evaluation of the Automatic Number Plate Recognition

For the sake of testing the three different ANPR solutions proposed in this project, we have collected a total of 165 images. The images have been taken from the esp-eye taken from the esp eye by using an SXGA 1280 1024 resolution, V-flip, and H mirror effects.

UZH-ANPR is the solution that we have composed by analyzing the performance of different open-source tools and bringing the best-performing elements together as an own designed constellation of co-working sub-parts.

Open ALPR is an open-source tool where no modifications from our side have taken place.

Plate Recognizer is a paid online service that provides a REST API for uploading images.

The plate recognition process has been separated into two main subcategories - long and zoom snapshots.

1. Long Snapshots: The images have been taken from a distance of at least 2.5 meters from the vehicle.

   We have been analyzing if the context in the image could affect the efficiency of the ANPR solutions. For instance, if other vehicles were close to the target vehicle, contours could irritate the edge recognition algorithm.

2. Zoom Snapshots: The images have been taken from a distance of a maximum of 1.5 meters from the vehicle.

Figure 4.7: Long vehicle snapshot

Figure 4.8: Zoom vehicle snapshot

We have run all three algorithms based on a set of images captured by the esp eye.

- Image Size Reference / Resolution: Super Extended Graphics Array (SXGA), 1280x1024 pixels

The results have been illustrated in the following figure:

The y-axis represents the success rate based on the total number of vehicle snapshots provided to each of the three ANPR solutions. The highest value on the y-axis represents a 100 percent success rate. This occurs if all of the given vehicle snapshots could be successfully processed and translated into a string that contains the plate numbers.

The x-axis represents the three ANPR solutions which have been further subcategorized into two categories; 1) long snapshots and 2) short snapshots as described above. This means that the analysis, understanding, and evaluation of the three proposed ANPR solutions will be divided into two parts. This subcategorization adds an additional dimension for evaluation and analysis purposes aiming for more apparent awareness and understanding of the behavior and outcome of the ANPR solutions.

We have added an additional dimension to the results and the accuracy indicators as well and provided three distinct kinds of results from the ANPR. The three kinds of results will be referred to as result subcategories and can be categorized as follows:

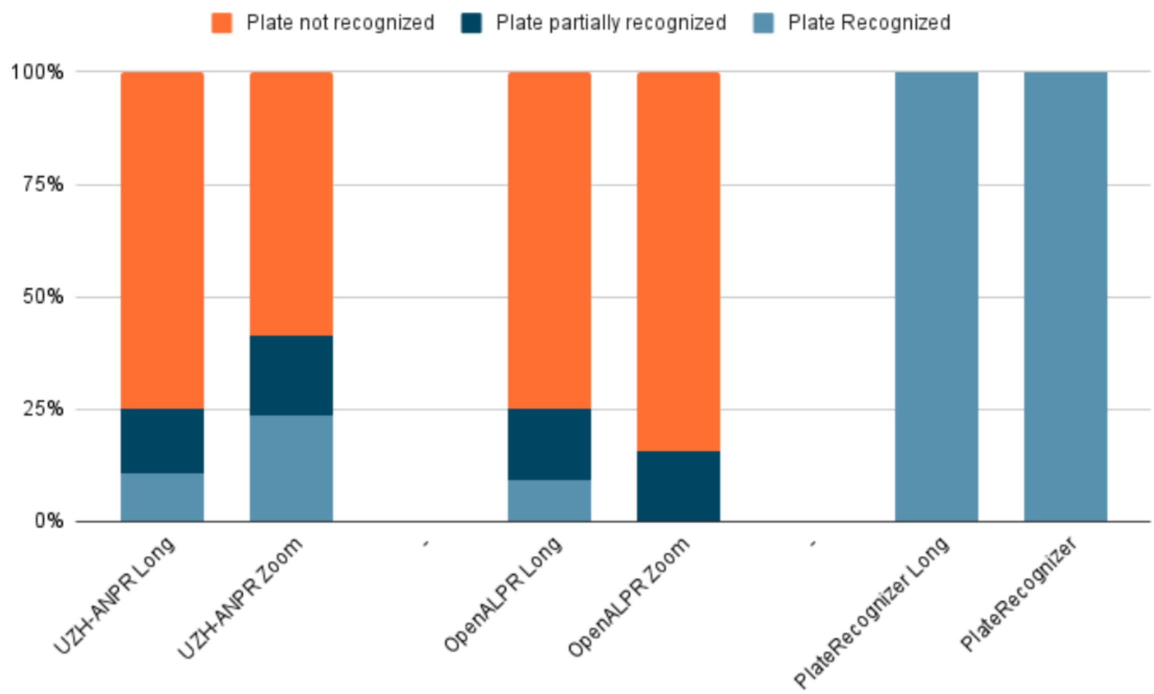

Figure 4.9: Performance indicators and results of the ANPR solutions

1. Plate not recognized: This Means that the plate was not found in the picture or the plate was found but the string could not be correctly processed. For instance, the edge recognition process triggered an object that also has four edges however it is not the plate.

2. Plate partially recognized: This means that the location of the plate was successfully detected however the string could not be processed accurately and resulted in semi-correct plate numbers. The metric used was to count the number of wrongly recognized characters. If the number of mistaken characters was less than 3 (maximum 2), the process was categorized as partially recognized. This includes the false recognition of the flags in the license plates as strings.

   For instance, AG 123 321 would be processed to 8 123 321. In this specific scenario, the number of mistaken characters would be 1.

3. Plate Recognized: This means that the location of the plate was successfully detected and the string of the plate could have been processed accurately.

4.3.5 Outcome ANPR

The first solution showed an important variance in accuracy between the two result subcategories.

During the first result subcategory phase, the long images, UZH-ANPR 25 percent of the plates were partially recognized or accurately recognized. The second result subcategory, however, provided a noticeable performance improvement, where approximately 40 percent of the vehicle snapshots could be partially or fully recognized.

OpenALPR, however, showed the opposite accuracy indications, where the long snapshots were processed at a better rate than the zoom snapshots.

The results however indicate that overall, the process of string recognition used in UZH-ANPR was more precise, since OpenALPR could only successfully translate the plate numbers in 5 percent of all the test scenarios (combining the long and zoom snapshots).

The winner with a quote of 100 percent success rate was the online paid service Plate Recognizer. The solution has shown implacable accuracy and great performance speed.

All of the images provided could be processed and 64/64 times the algorithm could identify the license plate.

An interesting example that again proves the accurate performance of the solution, is the following image. The plate is not aligned perfectly, since some degree of rotation can be perceived. The solution success identified the plates.

Figure 4.10: Winding zoom snapshot of a vehicle license plate

For the sake of testing the performance further, we have provided the algorithm with a damaged license plate. The online tool performed and processed the images without issues. This was the only case where the solution did not return the license plate on the following image:

Figure 4.11: Damaged vehicle license plate

We would count this as a successful test since indeed, there was no license number to recognize.

As a further step towards correct understanding and evaluation of UZH-ANPR, we have divided the results into a further layer. This additional layer is meant to provide added value and deeper comprehension of the results provided by UZH-ANPR and to find potential improvement factors.

The extra layer will solve the inaccurate state of the first result category, plates that were not recognized. There are several, very diverse factors that could be regarded as the reason for the failed recognition of the plate.

By being able to successfully identify the background of a failed ANPR process, we could potentially focus on that specific part of the solution. The purpose is to understand and find patterns or improvement potential.

This extra layer is categorized by the following metrics:

- Failed: The plate location could not be located in the vehicle snapshot.
- Plate: The plate location could be successfully located in the vehicle snapshot.
- Plate and String: The plate location could be successfully located and the string could be successfully processed.

Chapter 5

Summary, Conclusions, and Future Work

5.1 Summary and Conclusions

To reiterate, the goal of this thesis was to create an environment that would allow the hands-free, fully automated and shared usage of a low-power portable EV home charger. Special to our solution is the customization and enrichment of a low-power charging device by installing and encapsulating the device along with further hardware and software components. As a result, we achieved the proper digitalization and consequently the successful remote manipulation of what once was a primitive charging device that could not support any kind of communication or data transfer with a central management system.

Furthermore, we have built a tailored central management system that supports the communication of portable charging devices and an EV by simulating some of the behavior and functionality of commercial and standardized charging station communication protocols (for instance OCPP).

We are the first solution in Switzerland that has achieved a successful remote manipulation of a portable low-power EV charging device and has created a tailored hardware and software solution for hands-free, fully automated, internet-independent, and shared usage of the device.

The motivation for this project is connected to one critical aspect of the adaptation of electric mobility in Switzerland, proper infrastructure and accessibility of charging spots.

Most of the charging stations in Switzerland are either installed in outdoor locations, in commercial (or industrial) areas, or owned by private households for private use. This situates the majority of the Swiss population, which is living in rented apartments in an unsatisfactory position. Swiss regulations do not allow the installation of high-power EV charging devices (from 11kW) without the previous adaptation of the building's physical infrastructure. The adaption of the infrastructure must be conducted by a certified electrician, who will previously analyze and determine the availability of strong current and furthermore whether the building is eligible to support a high-power device. Regulations and infrastructural challenges exclude a significant percentage of Swiss buildings from being ineligible for the adaptation of high-power devices (charging stations for instance) - or put an immense economical barrier.

Our purpose of creating a shared approach for the usage of low-power portable EV home chargers is to be directly associated with the lack of infrastructural requirements for the installation of high-power devices in most of the older Swiss buildings.

Our low-power device EV charging approach is meant to be directly attached to any socket near the parking lots. In other words, no strong current cable, certified electricians, or installation is required and every building containing an available domestic socket (3.6kW) could potentially integrate our solution.

Another key achievement of our solution is the implementation of a communication protocol that will accurately take care of the accountability, supervision, and fully automated and shared usage of the domestic charging device. To clarify, the device itself (the 3.6kW home portable EV charger) still does not have the ability to communicate with third parties. The communication and all the resulting features and functionality is to be accredited to the installation of further hardware devices (which are capable of communicating with other devices) and the implementation of a central management server that coordinates the interaction of the hardware components in the background.

Our last key achievement is directly related to the lack of internet connectivity in Swiss underground garages. Aiming for a shared environment, identification and authentication mechanisms must be built into the solution. Not only for the activation of the charging procedure itself but also for accurate cost accountability along all users.

Users would first have to navigate to the web frontend, then they would have to log in to the app and hit the activation button to consequently start the charging session. This approach was challenged by the lack of connectivity in most of the underground garages. As a solution and alternative approach to the traditional web interface, a fully automated solution was conceived relying on ANPR (automated number plate recognition) technology. This would imply the exigency of a further hardware component (a camera) and a further module in our central management system. However, the value provided by this approach significantly surpasses the efforts and costs since no internet connection (from the end user side) would be required. Our solution empowers now a fully automated and hands-free triggering and authentication process without any human intervention.

In conclusion, we have been able to implement a low-power, hands-free, fully automated, internet-independent, and share solution that runs on our own communication protocols and that can be easily installed in any Swiss rental apartment without the need for any infrastructural upgrades or certified electricians. At the same time, the economical cost of our solution represents a fraction of the price involved in the purchase and installation of a commercial high-power charging point.

5.2 Future Work

This project serves as a fundament that will evolve and translate into a standardized product (we will refer to the product described in this thesis as UZH-Charge).

UZH-Charge will encapsulate the following (standard) hardware equipment which will be fastened into a box:

1. Portable EV home charger (3.6 kW)

   Type 2 cable that will be plugged into the EVs

2. Digital power meter

   Measurement of the power consumption and transfer of the consumption information to the central management system.

3. Smart Switch

   Remotely turning on and off the device.

4. Motherboard (RPI)

   Handles the communication between all hardware and software components.

5. RFID Reader

   As an alternative authentication option.

6. Camera

   Serves for ANPR purposes.

7. (Touch)screen monitor

   Serves for the initial installation of the device and later on as feedback on the current consumption while charging an EV.

In the external body of UZH-Charge, the following components will be visible and accessible to the owner:

1. Power cable

   Attaches the devices to the domestic socket.

2. Type 2 Cable (to be attached to the EV)

   Enables the attachment and charging of any EV containing a type 2 socket.

3. LAN/Ethernet socket (if no wireless internet is provided)

   Allows internet access through cable.

4. (Touch)screen monitor

   1. Initial setup and WIFI connection (if provided) as an alternative for internet access through cable.

   2. Display of live consumption statistics during charging sessions

5. RFID Reader

   Allows the reading of a physical key as a backup for the web solution and the ANPR approach.

6. Camera (for a fully automated, hands-free process relying on ANPR)

   Serves the purpose of a fully automated, hands-free ANPR-based approach.

UZH-Charge is meant to be managed by a central management system that is composed of by following elements/modules:

1. API endpoint for the communication between the device and the central management system.

2. Thread 1: communication protocol with UZH-Charge for the basic manipulation of the device. Request handling, registration, authentication, and lookup of historical charging data.

3. Thread 2: Measurement of the consumption

4. Thread 4: RFID reader

5. Thread 3: ANPR mechanisms

Every individual UZH-Charge device is meant to have its own front-end environment. The owner will have administrator rights through an administrator account.

UZH-Charge is meant to be easily installed by the owner of the device at the desired location without the need for any prior knowledge or a certified electrician.

The installation of UZH-Charge implies the following steps:

1. Installation of the device (max. 3.5kg) on a wall.

2. Attachment of the power cable into a domestic socket.

3. Initial connectivity setup

   Option 1: LAN cable that can be directly attached to UZH-Charge's internally installed RPI.

   Option 2: (Touch)-Screen monitor where the owner can select the desired wireless connection by using the PRIs integrated WIFI.

4. Initial communication setup

   Option 1: UZH-Charge will automatically send a request to an API endpoint once connected to the internet without the intervention of the owner.

   Option 2: The owner must insert a given secret code in the web front end in order to allow the device to communicate with the central management system.

The combination of these software elements along with the hardware components will empower a sharing approach, hands-free and automated usage, and cost accountability.

Bibliography

[1] Oscillation Gas Prices, Source: https://edition.cnn.com/2022/03/04/energy/gas-prices/index.html

[2] Jaguar brand will be all-electric by 2025, Source: https://www.bbc.com/news/business-56072019

[3] Market Share Electric Vehicle Manufacturers, Source: https://www.fool.com/investing/2022/12/29/why-tesla-rivian-and-lucid-motors-are-rocketing-hi/

[4] Making electric vehicles fair and affordable for all, Source: https://www.openaccessgovernment.org/ev-infrastructure-strategy-making-electric-vehicles-fair-affordable-for-all/149173/

[5] How long to charge an electric car, Source: https://pod-point.com/guides/driver/how-long-to-charge-an-electric-car

[6] Five challenges facing the public charging network, Source: https://www.fleetnews.co.uk/electric-fleet/charging-and-infrastructure/five-challenges-facing-the-public-charging-network

[7] EWZ electromobility, Source: https://www.ewz.ch/dam/ewz/Geschaeftskunden/Elektromobilitaet/Ladel%C3%B6sungen/ewz_Ladeloesung_Flyer_A4_D_web.pdf

[8] Swiss emobility, Source: https://www.swiss-emobility.ch/de/Laden/Baubewilligungsverfahren-von-Ladestationen.php

[9] Financing models for charging points, Source: https://www.virta.global/de/blog/die-verschiedenen-finanzierungsmodelle-fuer-das-geschaeft-mit-ladestation

[10] Building owners reject the idea of charging at rented buildings, Source: https://www.srf.ch/news/panorama/frust-fehlende-ladestationen-muehsame-vermieter-bremsen-e-auto-fahrerinnen-aus

[11] What is a hight power devices, Source: https://abc-haustech.ch/schlieren-oengg-altstetten-oberengstringen-dietikon/info/starkstrom/

[12] Installation of a wall box, Source:https://blog.wallbox.com/de/ladestation-installation/

[13] Load Balancing in Buildings, Source: https://www.elektro-material.ch/de/cms/blog/lastmanagement-fuer-ladestationen

[14] Requirements of a smart charging system ,Source: https://www.ekz.ch/de/geschaeftskunden/angebote/elektromobilitaet/ladeloesungen-fuer-ihre-immobilien.html

[15] openchargealliance.org: Overview: OCPP versions, Source: https://www.openchargealliance.org, 2014 - 2018

[16] T. Venkata Pruthvi, N. Dutta, P. Babu Bobba, B. Sai Vasudeva: Implementation of OCPP Protocol for Electric Vehicle Applications, Sep. 2019.

[17] Open Charge Alliance, Source: https://www.openchargealliance.org/protocols/archive

[18] R. de Leeuw, R. Lamers, B. McMahon, K. Muhlenberg, P. Rademakers, S. Tcaciuc, K. van Zuuren: Open Charge Point Protocol 1.6, Oct. 2015.

[19] MorecTech, Source: https://morectech.eu/de/collections/type-1-charger/products/morec-type-1-switchable-10-16a-ev-charging-box-2-2-3-6kw-6m

[20] Gravity Smart Switch, Source: https://www.bastelgarage.ch/gravity-16a-relais-modul-3-3-5v-logik-kompatibel

[21] Raspberry Pi 4 Computer Model B specification, Source: https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf

[22] ORNO 515 modular digital meter, Source: https://orno.pl/de/produkt/1079/1-phasen-stromzaehler-100a-mehrtarif-rs-485-anschluss-mid-1-modul-din-th-35mm

[23] RPI USB RS485 converter, Source: https://www.reichelt.com/ch/de/raspberry-pi-usb-rs485-schnittstelle-ch340c-rpi-usb-rs485-p242783.html

[24] Esp Eye, Source: https://www.espressif.com/en/products/devkits/esp-eye/overview

[25] OPENALPR, Source: https://github.com/openalpr/openalpr

[26] Platerecognizer.com, Source: https://platerecognizer.com/

[27] ANPR with Python, Source: https://github.com/nicknochnack/ANPRwithPython, 2020

[28] A. Rio-Alvarez, J. de Andres-Suarez, M. Gonzalez-Rodriguez, D. Fernandez-Lanvin, and B. L. Perez: Effects of challenging weather and illumination on learning-based license plate detection in noncontrolled environments, Sci. Program., vol. 2019, pp. 1-16, Jun. 2019.

Abbreviations

EV	Electric and Plug-in Hybrid Vehicles
OCPP	Open Charge Point Protocol
E-Drivers	Electric or Plug-in Hybrid Vehicle Drivers
RPI	Raspberry Pi
WIFI	Wireless Fidelity
RFID	Radio-frequency identification
ANPR	Automatic Number Plate Recognition
DC	Direct current
AC	Alternating current
GPIO	General-Purpose Input/Output
API	Application Programming Interface
LED	light-emitting diode
ms	Milliseconds
V	Volt
kW	Kilowatt
kWh	Kilowatt-Hour
FACEV	Fully Automated Charging of Electric Vehicles
IoT	Internet of things
SDPM	Single-Phase Digital Power Meter
V2G	Vehicle To Grid

List of Figures

Appendix A

Installation Guidelines

This chapter provides the necessary information to run the prototype which is built in Python by using the Framework Django.

A.1 Source code

Git Repository: `https://github.com/xicoarm/ba3.git`

A.2 Envirnoment Setup

Enter the directory ”UZH-Charge”.

Installation through pip of the required libraries:

1. pip install django
2. pip install openai
3. pip install matplotlib
4. pip install opencv-python
5. pip install imutils
6. pip install pytesseract
7. pip install RPi.GPIO
8. pip install pymodbus

Once the libraries were installed, the server can be started through this command:

- python3 manage.py runserver