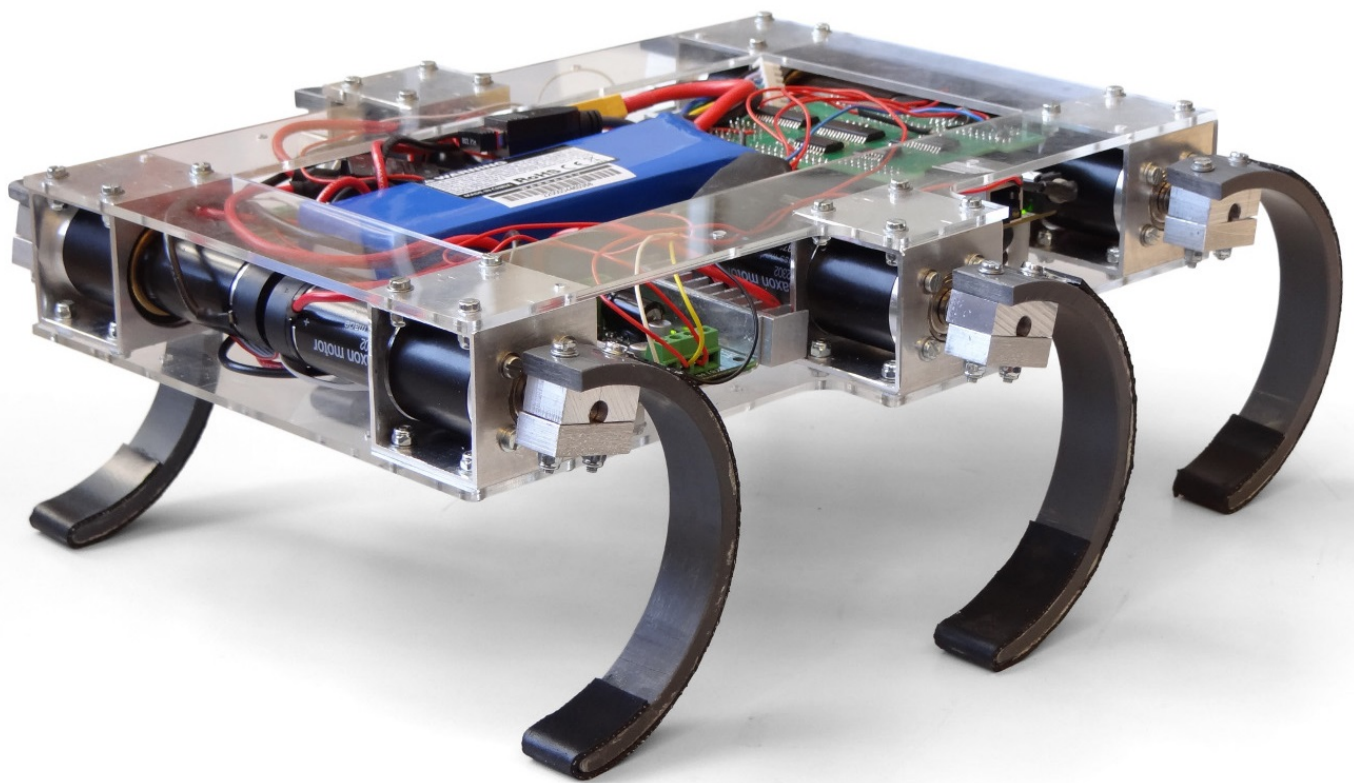


Autonomous charging station for the ZEBRO

Bachelor thesis

Niels de Jong (4280121)
Tom Heijnen(4304586)

Technische Universiteit Delft



Abstract

The ZEBRO is a fully autonomous six-legged robot designed for swarm behaviour and is the size of an A4 sheet of paper. However, there does not exist an charging station for the ZEBRO yet. At this time, the battery still has to be replaced manually, which defeats the purpose of an autonomous system. An autonomous charging station has to be designed to complete the system. The charging station can be split into 3 parts: the autonomous charging station, the wireless power transfer, and the battery management system (located in the ZEBRO). In this particular thesis, the autonomous charging station will be discussed.

Since all components in the ZEBRO are built to be modular, the design of the autonomous charging station is chosen to fit this principle. The charging station will consist of one central unit, and several connectable (modular) pads. Each pad will be able to hold and charge a single ZEBRO in 30 minutes.

The charging station contains a communication module to communicate its location to ZEBRO's in need of charging. To calculate the locations of the connected pads, a grid will be created by serial communication between the charging station and the connected pads through Raspberry Pi's. (The pads do not have a communication module and do not have their location defined)

To power the wireless power transfer unit, a power supply of 48V and 500W will be implemented in the charging station. Each pad will consume about 100W, so a maximum of 4 pads can be connected. (Due to power loss). If more pads need to be connected, an additional power supply can be put in parallel.

All separate components have been simulated or tested and do work as predicted. The integration of all components of the charging station was successful as well. At this time, the complete system with the wireless power transfer and battery management system has not yet been put together and so it is unknown if it works as a whole.

Contents

Abstract	iii
1 Introduction	1
2 Program of requirements	3
3 Design Process	5
3.1 Connection options.	5
3.1.1 Magnetic spring.	6
3.1.2 Charging pads	6
3.1.3 Wall-slide connection	7
3.1.4 Fyke	9
3.1.5 Design choice	9
3.2 Design overview.	10
3.3 Power.	12
3.3.1 AC/DC	12
3.3.2 Safety measures	13
3.4 Processing	17
3.4.1 Grid creation	19
3.4.2 Raspberry Pi management	21
3.4.3 Communication.	21
3.5 Total design	22
4 Prototype implementation	25
4.1 Pad layout	25
4.2 Grid simulating.	27
4.3 Load switch test	27
4.4 Complete prototype	29
4.5 Components and costs	29
5 Conclusion	31
A Raspberry Pi pin layout	33
B Installation manual	37
B.1 Station	37
B.2 Pads	39
B.3 Use	41
Bibliography	43

1

Introduction

The ZEBRO project is an initiative of the TU Delft to research swarm-related behaviour of simple and straight-forward robots (ZEBRO's). The purpose of a ZEBRO is to cooperate with its colleagues to perform tasks which are not suitable for bigger robots and would be time consuming for humans. An example of such a task is the mapping of a disaster area.

The ZEBRO's come in different sizes. The emphasis of the design for autonomous charging will be laid on the deci-ZEBRO, where an expansion to the kilo-ZEBRO may be possible.

The ZEBRO's are fully autonomous robots. In the current design state, the battery has to be manually disconnected, recharged, and reconnected. This defeats the purpose of the autonomous behaviour of the robot. A goal set by the ZEBRO team is to have a swarm of 100 robots by the end of June 2017. In such an expanded swarm, the manual connection method is very time consuming. For this reason, an autonomous charging station and charger module are needed.

The global charging scheme is as follows. The charger module will sense when the battery is running low, and sent a signal to the top-level that the ZEBRO needs to recharge. The ZEBRO will then check if there is a charging station nearby and if there is a charging spot available. If so, the ZEBRO can go to this charging station where the charging will be done autonomously. Being fully charged, the ZEBRO can continue with its tasks.

The design of the charging station and the charging module can be split up in three sub components. First, the charging station itself, converting power from a wall socket to the ideal voltage to transfer by means of inductance. The second component is about transferring the received power. The final sub component is the internal battery management system in the ZEBRO. Where the ideal method to charge the battery is specified. The emphasis of this particular thesis is laid on the design of the wireless charging station. A very simplified overview of the system is shown in figure [1.1](#) for clarification

The optimal charging method to charge the ZEBRO's in the most effective way is not known yet and has to be researched. However, a lot of information about charging methods, safety measures and communication methods is available. The first goal of the project is to deliver a working charging station. Additional to this, design choices have to be well-considered. Finally, the design must be created in such a way that possible adjustment for perfection is possible.

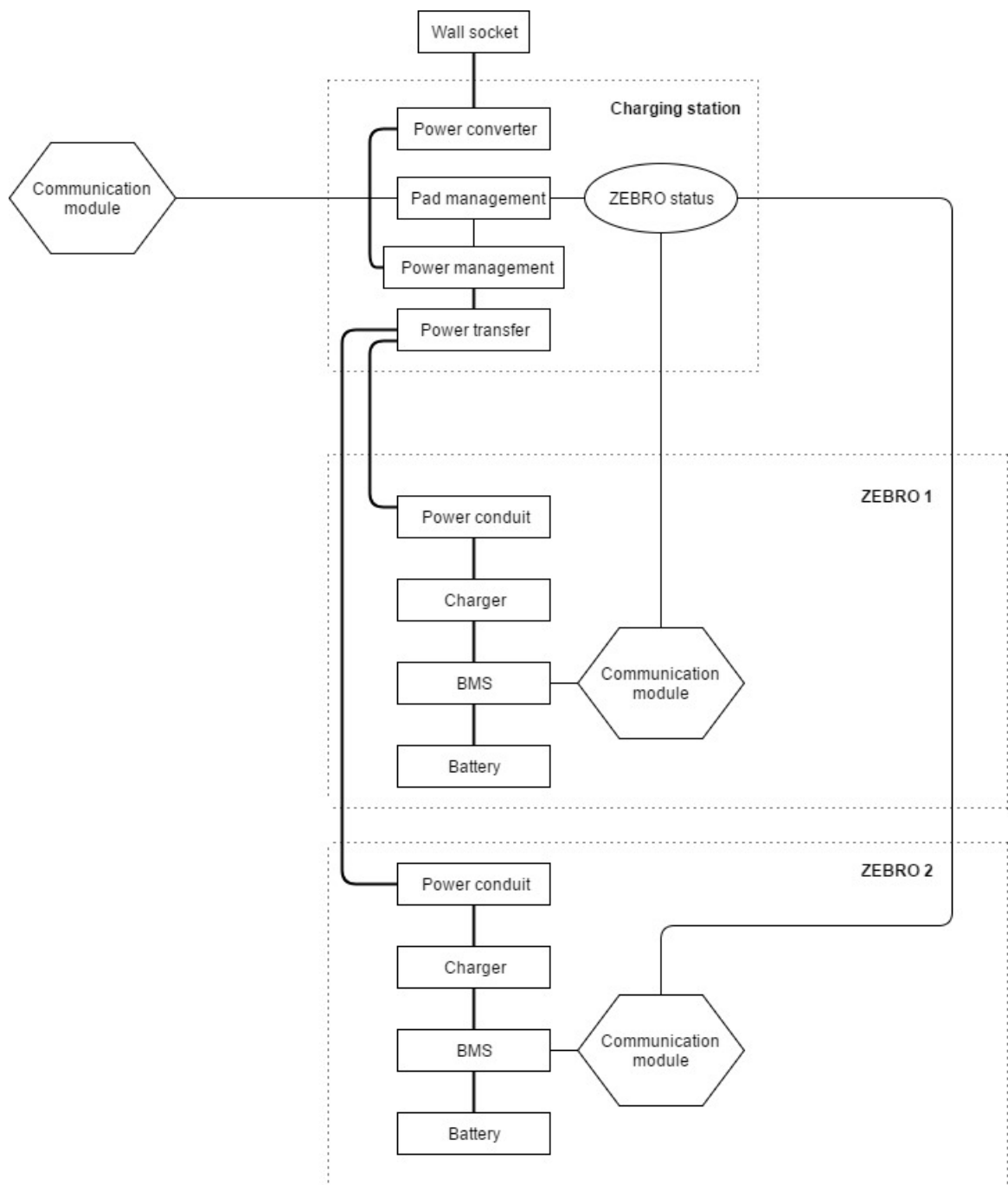


Figure 1.1: General overview of the complete system

2

Program of requirements

A PoR (Program of requirements) is set up for the autonomous charging station and charger module of the ZEBRO to give a clear overview of the goals and challenges of the design. In this PoR, the emphasis will be laid on the the charger station, which is discussed in this thesis. The charger module is developed for the specific purpose of the ZEBRO. This results in an absence of business and marketing requirements. However, in the far future, some may be added to the PoR when the ZEBRO is deployed on a bigger scale.

This program of requirements does not solely consist of requirements given by the ZEBRO team. Some of these requirements have been added in the course of the project to define what requirements are needed to achieve a system that upholds the given requirements.

1. Functional requirements

- 1.1. The charging station must be capable of charging the battery of a ZEBRO autonomous.
- 1.2. The charging station must be able to maintain a swarm of 10 ZEBRO's charged.
- 1.3. The system must be expandable for more ZEBRO's.
- 1.4. The charging station must be able to be expendable from the deci ZEBRO to the kilo ZEBRO.
- 1.5. Data about the charging state of the ZEBRO must be made available.
- 1.6. The charging module must be completely modular, independent of all other modules.

2. Ecological embedding in the environment

- 2.1. All components of the charging station and the charger module have to be provided with short circuit protection.
- 2.2. The coil attached to the ZEBRO must be provided with a metallic plate to prevent interference of the electromagnetic field in the components of the ZEBRO.
- 2.3. A shutdown button must be provided for each pad and the charger station to assure safe disconnection.
- 2.4. When the charger station is operating, all electrical components must be isolated from direct contact.
- 2.5. The voltage delivered to the Raspberry Pi's can not exceed the range of 4.75 – 5.25V.^[1]

3. System requirements

3.1. Utilization feature

- 3.1.1. The charger station must be able to create a grid of all connected pads.
- 3.1.2. The charger station must indicate which pads (and their locations) are free to the communication module.
- 3.1.3. The minimum capacity of the battery is $3000mAh$.
- 3.1.4. $48V$ must be delivered to the coils.
- 3.1.5. The battery management system must deliver a voltage of $14.4V$.

3.2. Production and discarding features

- 3.2.1. The assembly of both the charger station and charger module must be easy.
- 3.2.2. All components must be easily dismantable and replaceable.
- 3.2.3. In transport, the charger station should be compact.

4. Development of manufacturing methodologies

- 4.1. The battery dimensions can not exceed $44 \times 150 \times 25mm$
- 4.2. A ZEBRO must fit on its pad independent of its orientation.

3

Design Process

The design process is the heart of the project. First of all, a decision for the charging method has to be made. Several options are shown and their positives and negatives are considered. The decision for the design that will be further elaborated is based on the program of requirements. The chosen design will be discussed in detail.

3.1. Connection options

This section starts with the absolute basics. The very first design decision that has to be made to get the project going is about the charging method. This can be done either with contact or without contact. Both of these solutions have their advantages and disadvantages. The initial preference lays with the contactless solution. To check whether this solution is the most viable, its advantages and disadvantages with respect to contact charging are displayed below.

- + There is no need for an autonomous mechanical coupling mechanism.
- + The temperature issue that occurs with coupling resistances [2],[3] does not have a role in wireless charging.
- + The ZEBRO does not need a specific orientation to enter the charging station.
- + The charging station can be build waterproof, since there are no objects that need a physical connection. This holds for the ZEBRO as well.
- The energy losses due to all the converting steps are higher than the energy losses with a connection based solution. [4]
- A lot more electrical components are needed to complete the system.

The amount of advantages of the wireless charging solution does exceed the disadvantages. Furthermore, the fact that the total energy consumption of the system is higher than the contact solution is not a big importance, neither was a maximum power usage defined in the program of requirements. So apart from extremes, the total power consumption does not really matter. And it can be concluded that contactless charging is the better solution for this particular project.

Nevertheless, for both options some solutions are elaborated. Including a magnetic connection on a spring, charging pads with a contact and with a contactless connection, contact connection against the wall and finally a connection using a fyke.

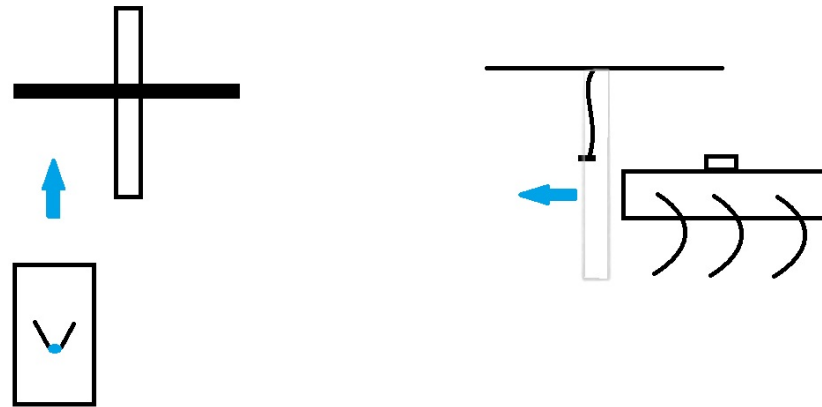


Figure 3.1: Sketches of the magnetic coupling by spring solution

3.1.1.1. Magnetic spring

The first solution is based on contact charging. The idea is to transfer power through a conducting wire that hangs down from a small roof of the charging station. The ZEBRO will have a small fyke on its roof to make sure the wire gets connected on the right point. Some simple sketches of the idea are given in figure 3.1. When the ZEBRO is done charging, it can simply walk away. The magnetic connection will disconnect automatically when the force of the spring exceeds the coupling force. The advantages and disadvantages of this design option are given below.

- + This solution can be expanded to work for the kilo ZEBRO as well, by adjusting the height of the wire.
- + It is easily expandable for a lot of ZEBRO's, by making the roof longer and adding multiple connection points.
- Filth may get stuck in the fyke on the ZEBRO, hindering the connection.
- The contact area is very small, while large currents are needed to full fill the charging speed given in the program of requirements. If the magnetic coupling force is not high enough, a lot of heat will appear due to the contact resistance.[5]
- The location detection accuracy (which is dependent on another module, the localization module) needed to connect with the wire is pretty high.
- There is little room left on top of the ZEBRO since both the communication module and localization module will be placed here as well.

3.1.1.2. Charging pads

Another design solution is charging with charging pads. This design can be implemented with either contact or contactless charging. In this solution a square central charging station retrieves power from the wall socket. It has connections on all four sides. On each side, a charging pad can be connected. Each charging pad has 4 connections as well, so the system can be expanded. A simple sketch of the setup can be seen in figure 3.2 As can be seen in this figure, it is possible to design different kind of pads which can be used to charge different kind of ZEBRO's. The pads can either be implemented with coils for contactless charging, or with some



Figure 3.2: General overview of the structure with the main charging station and additional pads

kind of mechanism that provides contact. There are several options to assure this contact, but since it is unknown how accurate the ZEBRO's can reach a location, it is best to assure that the effective charging area is as big as possible. A good solution to this problem is a cone. All pads could be implemented with a cone, and the ZEBRO will have a gap underneath where the cone can enter. This way, the ZEBRO does not need to walk to an exact location. A sketch of this solution is displayed in figure 3.3

Just like the other options, there are some advantages and disadvantages for the charging pads.

- + Modular and very easily expandable for larger ZEBRO swarms.
- + Can easily be implemented for all type's of ZEBRO.
- + The ZEBRO does not need a specific orientation to enter the charging station.
- + The solution is very compact and easy in transport.
- + Looks professional and high tech.
- Requires a lot more control than other charging options due to the modularity of the system.
- May be relative expensive due to additional control elements that have to be implemented in the pads. (The charging station has to be aware what pads are connected and where they are connected to broadcast the correct location of these pads.) These extra costs will be in the order of tens of euro's.
- For the option with cones, there isn't much room inside the ZEBRO to implement the hole. This could go at the expense of battery capacity.

3.1.3. Wall-slide connection

The accuracy of the ZEBRO is uncertain as mentioned before. A connection type where the ZEBRO simply has to walk against a wall can come in handy to avoid this problem. In this solution, the ZEBRO walks on a surface which has little grip, until it hits the wall of the charging station. Here, it lays itself down on top of a small bar with connection points that sticks out.

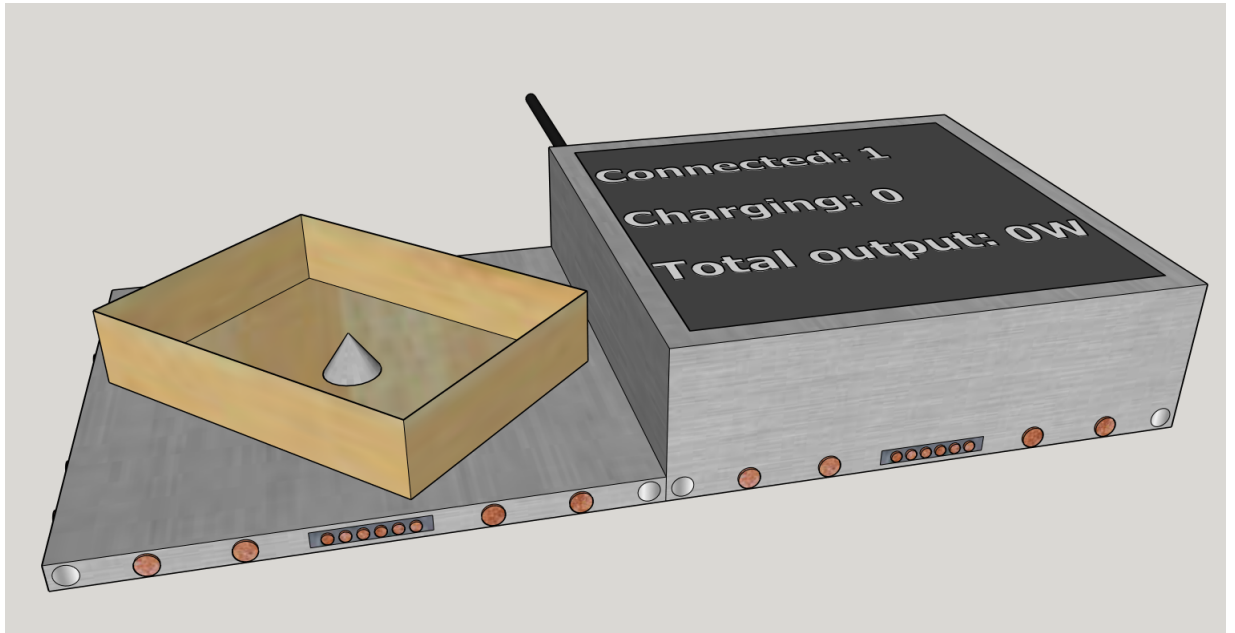


Figure 3.3: Sketch of connecting the ZEBRO with a pad using a cone

The bar has springs implemented, to assure pressure between the contact points. A sketch of the front and side view is displayed in figure 3.4

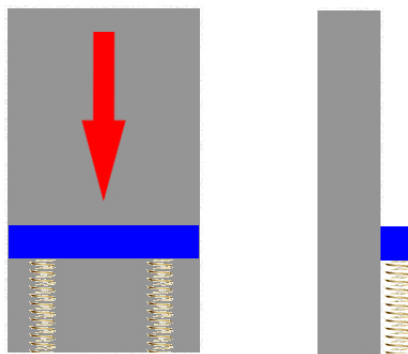


Figure 3.4: Sketch of the wall connected design solution

The advantages and disadvantages of this design solution are given below.

- + The location determination accuracy is allowed to be low.
- + Can be expanded to the kilo ZEBRO.
- It is a mechanically challenging solution.
- The only force assuring a contact connection is the weight of the ZEBRO. The contact resistance may be a problem here.

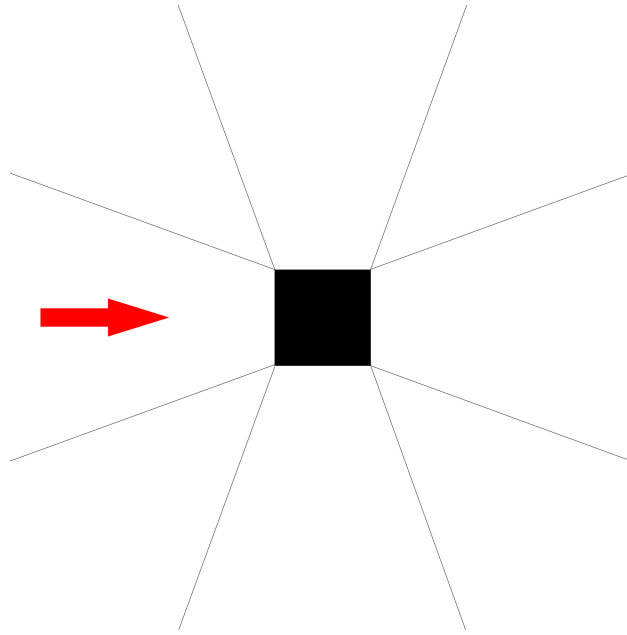


Figure 3.5: Guiding the ZEBRO by using a fyke

3.1.4. Fyke

The final design solution is implementing the charging station with a fyke. With the fyke, it does not matter if the ZEBRO walks slightly in the wrong direction, it will be forced to end up on the charging platform. A simple sketch can be seen in figure 3.5. The connection can be done by using a contactless charging method by putting a coil in front of the charging station, or combining this solution with the previous mentioned wall-slide connection. Some advantages and disadvantages of this solution are stated below.

- + Simple but efficient.
- + The positioning of the ZEBRO on the charging module is not that dependent on the accuracy of the localization.
- The charging station is very big and bulky.
- It has a limited capacity of 4 ZEBRO's per charging station.
- Can not easily be expended to the kilo ZEBRO.
- The ZEBRO needs a specific orientation en direction to enter the fyke.

3.1.5. Design choice

As given in the previous paragraphs, each design solution has its advantages and disadvantages. Based on these, a decision can be made to pick the most effective design for this particular situation. First of all, it is important that the solution can fulfill all requirements that are stated in the program of requirements. This is where the fyke gets discarded, it is not expandable for larger ZEBRO swarm and neither can it be used of the kilo ZEBRO.

The remaining three options all have their assets, where the charging pads appear to have the most positives. And especially because the system is very modular, it does look very appealing to continue with further investigation. So the charging pads are chosen to be further elaborated in the following paragraphs.

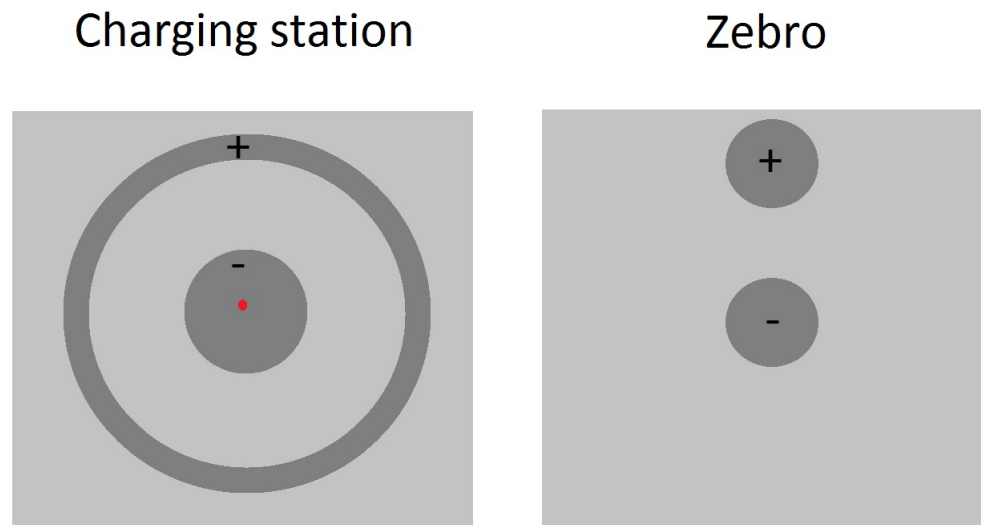


Figure 3.6: Connecting the ZEBRO with contacts shaped in a circle

But first a decision has to be made between contactless or contact charging. Before making the final decision, some more contact options that could be implemented in the pads are considered. The big plus of the charging pads, is that the ZEBRO can approach the pad from any side, without having to consider its orientation. So the contacts should be designed in such a way that this principle holds. The solution to this issue can be seen in figure 3.6.

In this design there is the possibility that the negative contact of the ZEBRO is contacted with the positive contact of the charging station. Problems can be avoided by placing a diode in the system. To make sure that the ZEBRO is on the right spot, this can be confirmed using infrared.

The design proposal given above for contact charging is legit and could work. However, it still has a lot of negatives, which contactless charging does not, as discussed in section 3.1. Neither does it have a big advantage to its favor. This is why contactless charging pads will be further elaborated.

3.2. Design overview

This section will give an elaborated overview of the chosen solution to implement charging pads with the charging station. A flow chart of the basic components that will be implemented is given in figure 3.7. The design of the autonomous charging station can be split into two main sub components: data processing and power management. Since the charging station will be connected to a wall socket, the 230V AC has to be converted to 48V DC (this is the voltage where the power transfer module works on) and distributed to all connected pads. The voltage of 48V is chosen to remain within the region of ELV (extreme low voltages) for safety reasons.[6] (The 48V DC will later be converted to 48V AC, which is just below the maximum of 50V) The power management section needs to be able to transfer 5V DC as well, for the processing section.

The processing section will be implemented with Raspberry Pi zero's (w). [7] The Raspberry Pi's are chosen because they have a clear user interface, making the design process a lot easier. They are mostly used as a proof of concept, but there may be a possibility to switch from Rasp-

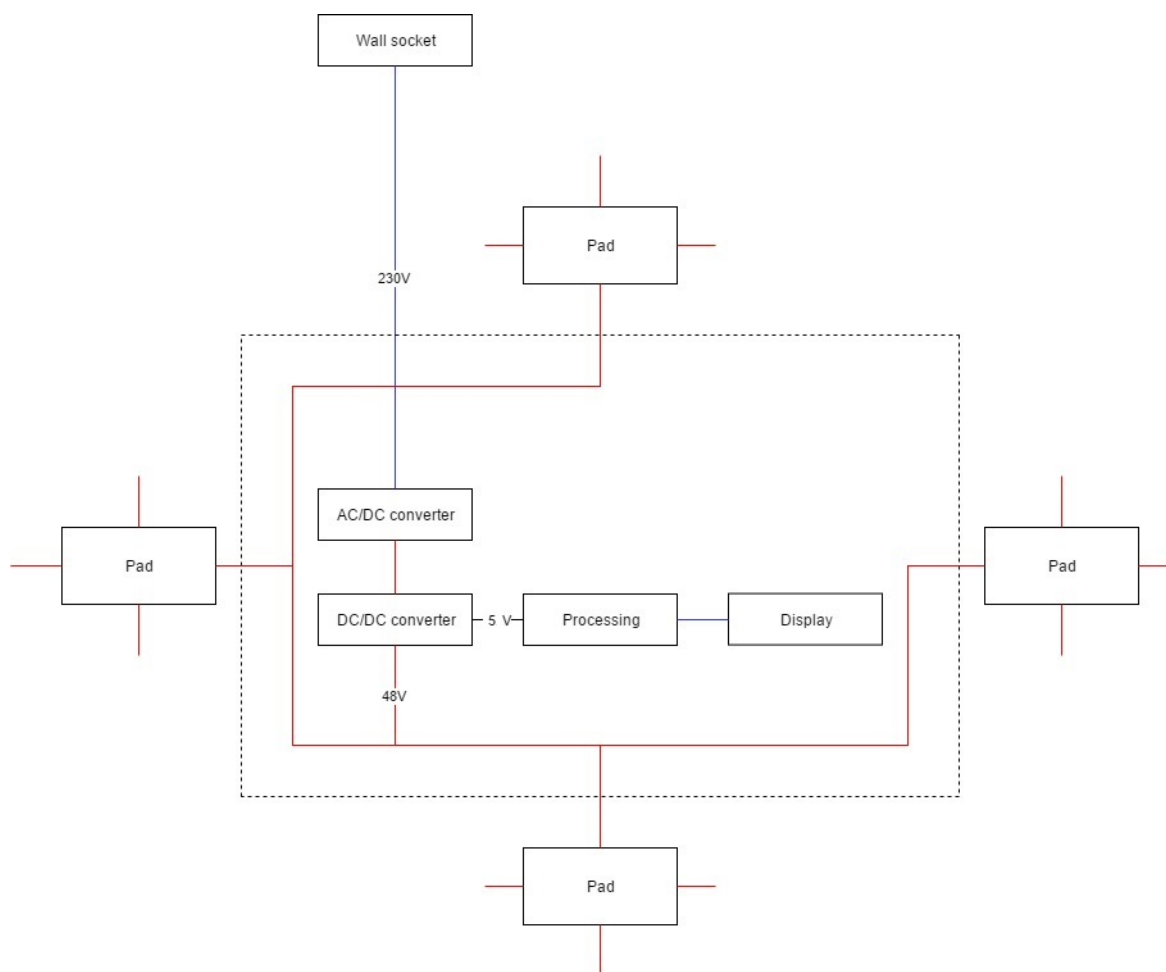


Figure 3.7: Flowdiagram of the components of the charging station

berry Pi's to loose micro controllers later. The Raspberry Pi's need a 5V DC supply. Since each pad needs control as well, they will also be implemented with Raspberry Pi zero's and need a 5V DC supply.

As final component, some user interface has to be implemented on the charging station to show what pads are connected, if there is a ZEBRO on a pad and if the ZEBRO is getting charged. There are several ways to do this, some examples are: a LED array, a display or just simply by showing the status in the terminal of a connected screen.

3.3. Power

As mentioned before, the power section has to deliver two voltages, the 48V and the 5V. With the supplied 48V, all the pads can be installed in parallel. It is important to mention that the power transfer uses a current of 2A, so a total power of minimal 100W has to be delivered to each pad.

3.3.1. AC/DC

AC/DC converters that convert 230V AC to 48V DC are available with different nominal power ratings. But using a power supply that can supply for example 15 pads with sufficient power, is quite useless if the charging station is used for a small ZEBRO swarm, where 4 pads may be enough capacity already. And so, just as in the rest of the design of the ZEBRO, modularity can avert this problem. Placing two 48V power supplies in parallel, results in the same voltage over the circuit, but then with more power available. However, two power supplies can not simply be put into parallel. Current can flow from one power supply to the other one, causing all kinds of unpredictable effects, which depend on the layout of the power supply. There are a lot of ways to optimize such a current-sharing system. [8] But in this case, both power supplies come from one central power supply and the solution is a lot more straightforward. It can be solved by simply putting a diode in series with each power supply. It is important that the diode can manage the high currents that flow through it, and does not get fried. Since currents of 10A can be reached with a power supply of 500W. However, the power supplies that are used already contain a mechanism to avoid these problems. [9]

There are several ways to deliver a voltage of 5V to the Raspberry pi's. The first option is a fairly straightforward solution. Since there is a voltage of 48V available from the AC/DC converter, a simple amplifier circuit can be created with a few resistors and an opamp. This circuit can be found in figure 3.8. However this circuit has the big negative that a lot of power is dissipated in the opamp and the resistors. The second solution is somewhat more elegant. This involves a buck converter and a LDO regulator. The buck converter is used to drop the voltage to a level in which the LDO regulator is able to operate. The LDO regulator stabilizes the voltage output to 5V. These 2 electrical elements can either be placed in the main charging station to convert the 48V one time, or they can be placed in each separate pad. The benefit of placing one in each separate pad is less connectors for each pad and the charging station. However, this option does result in a lot more installation effort, higher costs, and more complexity in the pads. Which are preferred to keep as simple as possible.

Another option is to forward the 230V AC from the wall socket to each pad, and use a simple phone adapter to drain the required 5V. This idea is rejected due to safety reasons.

The final option that is considered is the most simple and straightforward solution. Here the 5V DC is reached by simply putting a 230V AC to 5V DC converter in the main charging station. Each pad will be connected in parallel to supply the correct voltage to each Raspberry Pi.

Since the program of requirements states that the charging station should be easily constructed,

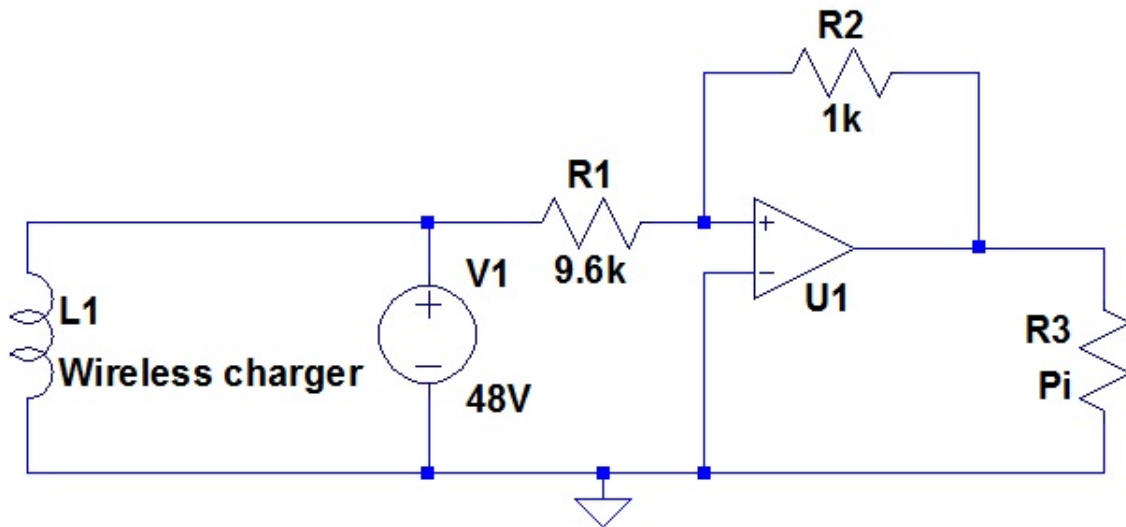


Figure 3.8: Raspberry Pi powered by an opamp

the final option is considered to be the best one for this design. A simple 40W LED driver can be placed in the charging station and supplies more than enough power for all the Raspberry Pi's that can possibly connected. [10] Since there is an excess of power, it is also possible to power the microchip that the power transformation design team may need as well.

3.3.2. Safety measures

When transferring large sums of power, safety measures have to be taken. This system will be build to handle 8 pads, this power can be reached by placing 2 of the power supplies that were mentioned in the previous subsection in parallel. An expended flow diagram of the charging station can be seen in figure 3.9. Since there is a possibility, that 8 pads are connected to one single side of the charging station, it is very important to take into account that large currents will flow through the first wires and connectors. These currents can increase up to a current of $I = \frac{P}{U} = \frac{1000}{48} = 20.83A$. All components in the charging station and pads will be designed to be able to handle such currents. The amount of ampere that wires are able to conduct before melting is dependent on factors such as length, resistance and wire thickness. The wires that are used in the charging station and charging pads can manage a current of 32A. For the connectors, the XT60 connectors are used. These connectors can manage a current of 65A. The reason that these connectors are used, is that they are already in use with the ZEBRO team, and for them, using similar materials for different applications is a big plus for organizing.

The wires and connectors may be able to manage a situation where more than 8 pads are connected. But the power supply can not. A fail safe has to implemented to make sure that the ZEBRO's will always be charged with 100W. This can either be implemented by placing a big switch in the charging station, or placing a switch in each pad, to disconnect the coil from the power flow. Since it is rather unnecessary to disconnect all the pads when there is only 1 pad to many connected, the system where a switch is implemented in each pad will be elaborated.

A commonly used technique is to use MOSFET's for load switching.[11] The first approach that is considered to use in this scenario can be seen in figure 3.10. Here, the second voltage source

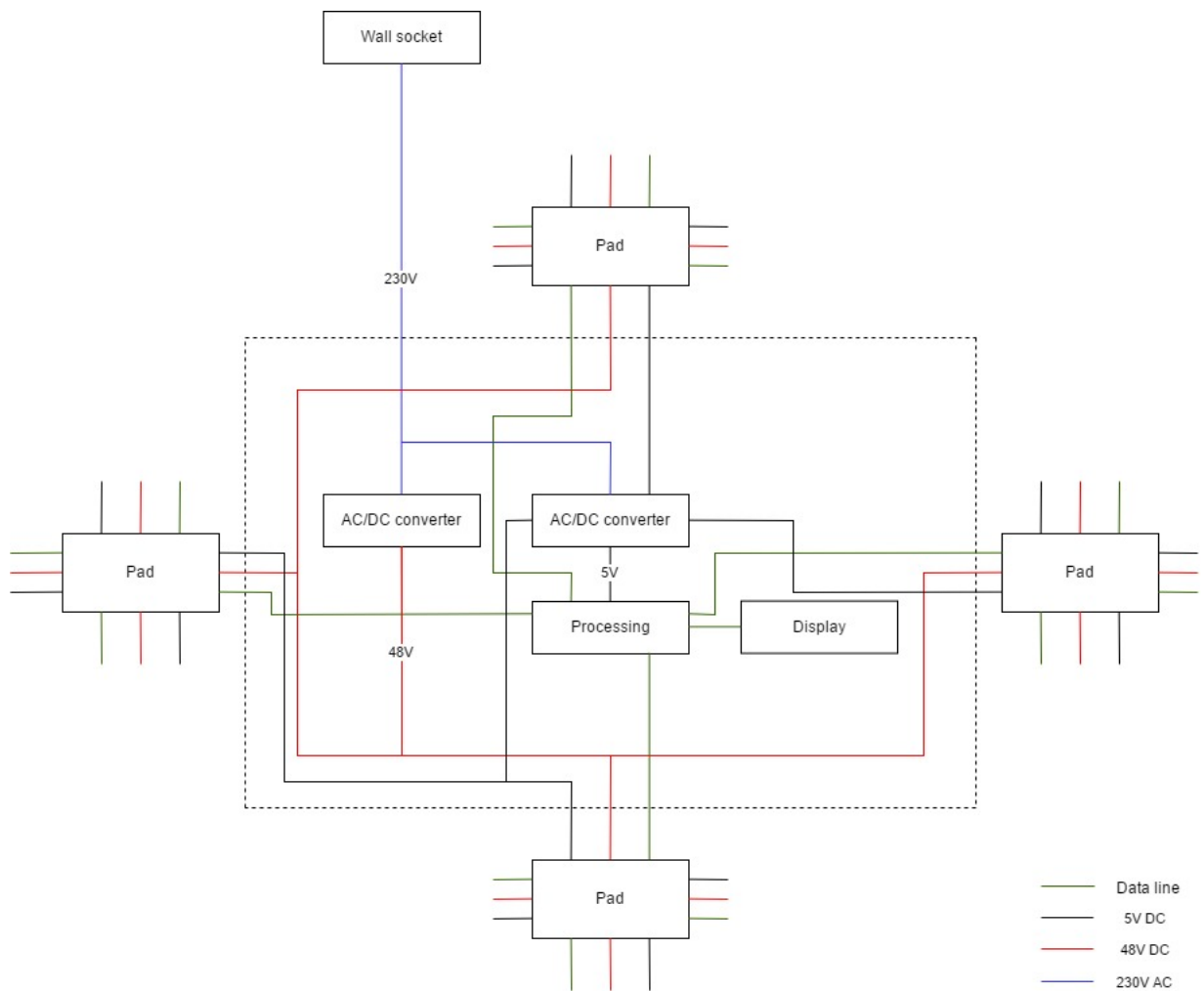


Figure 3.9: Expanded flow diagram of the main charging station

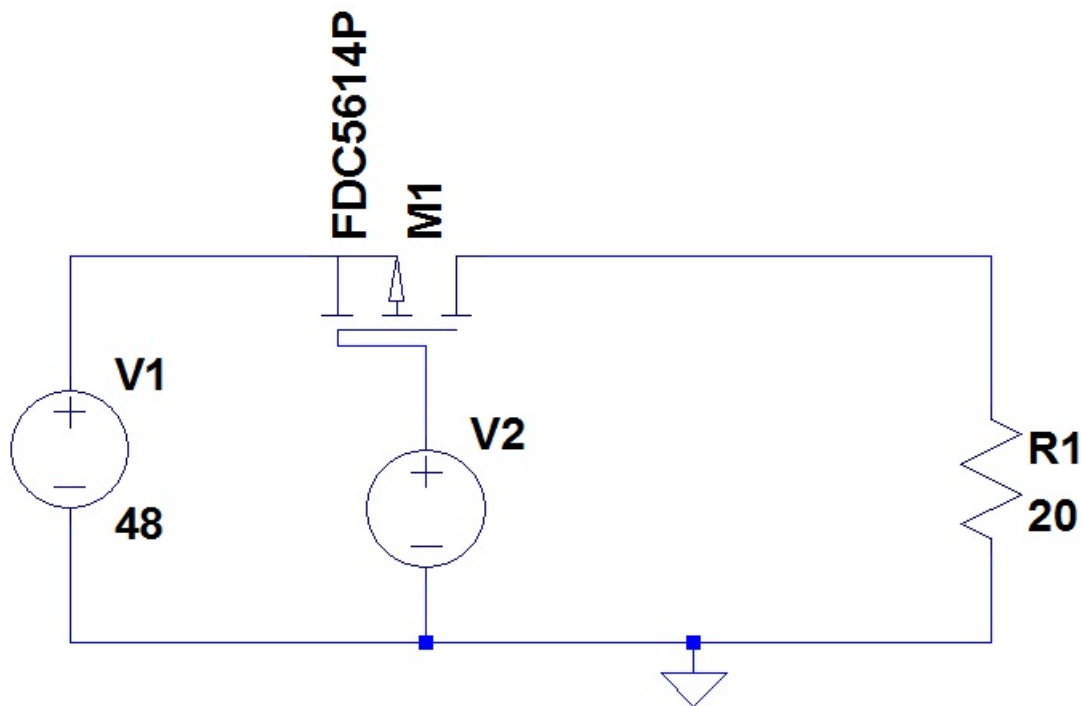


Figure 3.10: Load switching with a single MOSFET

resembles the signal that is needed to decouple the load. A result from the simulation can be seen in figure 3.11. A big problem with this solution is that the threshold voltage of the FDC5614P transistor is $-1.6V$ [12], meaning that the system won't work if the gate voltage is below $46.4V$. (This theory holds for all PMOSFET only with different threshold voltages.) This system switches inverse to the input signal as well. (No current if the input signal is high and vice versa) Which is inconvenient, because then current will flow when they Raspberry Pi's are starting up since they do not sent a signal yet. Since the switching is done with $3.3V$ or $5V$ from the Raspberry Pi's, an alternate solution has to be found.

The solution that is found to switch the load while using a low voltage signal from the Raspberry Pi is displayed in figure 3.12. In this circuit, the input signal from the Raspberry Pi (V2) is placed on the gate of an n type MOSFET (M2). If the signal is high, the NMOS will conduct and the potential on the gate of the p type MOSFET will become equal to the ground. Now the PMOS will conduct as well and the $48V$ will be put across the load resistance. If the signal from the Raspberry Pi is low, the NMOS does not conduct, and neither does the PMOS. The potential on the gate of the PMOS will be set on $48V$ through resistor R3. This resistance needs to be a high value. When the gate is set to $0V$, or ground, $48V$ will be set across R3. Here this resistor will start dissipating power. If this resistor has a low value, more power will be dissipated. In the case of $10k\Omega$, the resistor will dissipate $\frac{48}{10k} \cdot 48 = 230mW$. Which is a considerable amount. So this resistor will be replaced by a resistor of $100k\Omega$. This results in $23mW$ of dissipated power.

The other resistor (R2) is used to keep the gate of the NMOS from floating. If the Raspberry Pi is not connected, the gate will still be defined as $0V$. The power dissipated through this resistor is a lot less, since there is only a voltage of $3.3V$ across it. The power dissipated in this resistor is $\frac{3.3}{10k} \cdot 48 = 16mW$. A simulation of the result is displayed in figure 3.13. The transistors that are chosen for this particular scenario are capable of handling a V_{DS} voltage of $60V$, since they

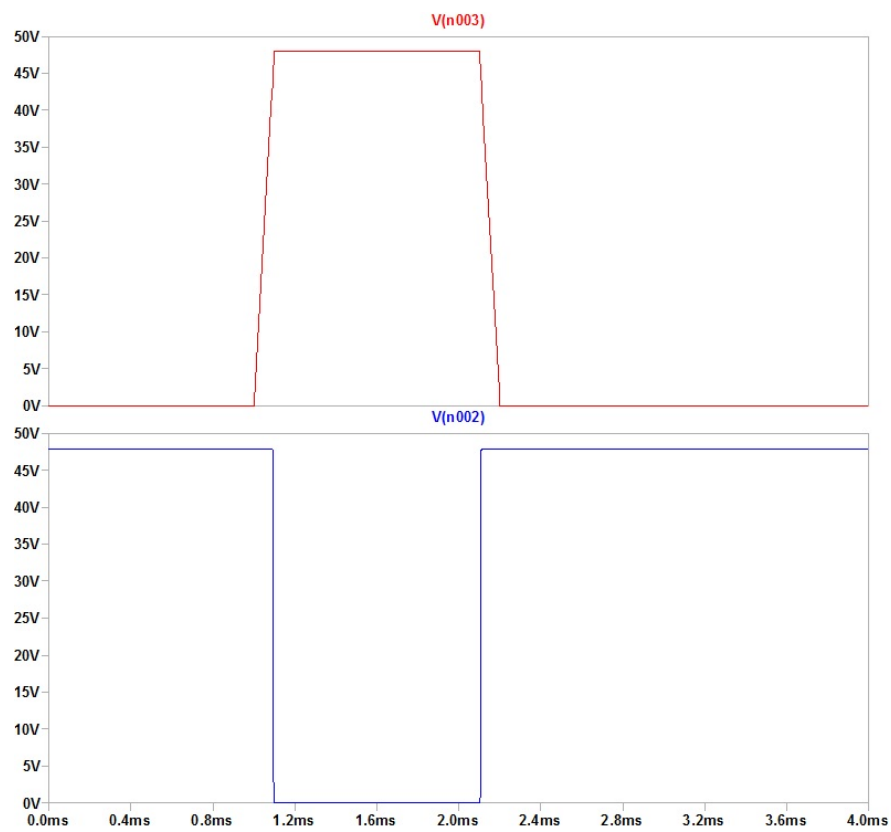


Figure 3.11: Simulation of load switching with a single MOSFET, the upper graph is V_G and the lower graph is V_{load}

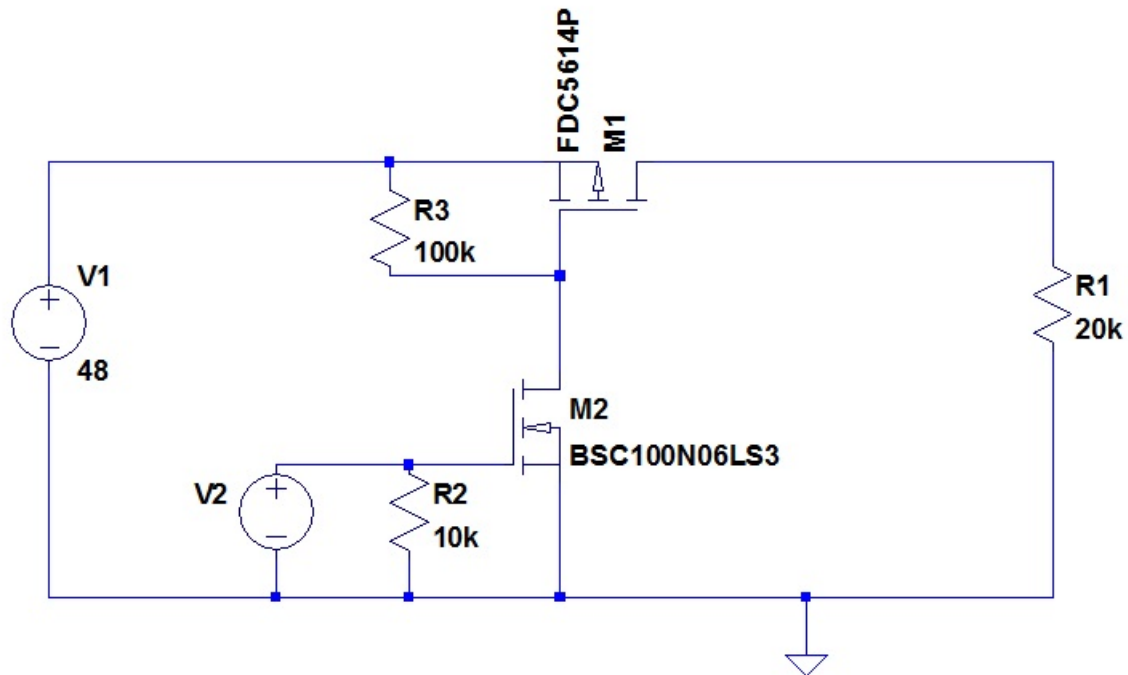


Figure 3.12: Load switching with multiple MOSFET's with a low voltage signal

will operate under a voltage of 48V. They are able to handle the current of 2A as well.[13]

The final component that needs to be implemented is the communication with the wireless power transfer team. They will send a signal which indicates if a ZEBRO is present or not. Because it is uncertain what kind of voltage this signal is, the signal can not directly be put on the Raspberry Pi. Since the GPIO pins do not have an over voltage protection. This problem is resolved by using a pull up network. This network is displayed in figure 3.14. In this figure, V2 is the unknown voltage signal delivered and V1 is the VCC of the Raspberry Pi. A PMOS is used to efficiently raise the voltage at the GPIO pin. The GPIO pin is connected with ground as well via R1 to keep the pin from floating.

3.4. Processing

Processing is an important part of the charging station. Creating an autonomous charging station without using software is possible, but this is way more complex and using software simplifies the design significantly. This is why Raspberry Pi's are used to simplify the challenge. The main task of the system is to identify and broadcast the locations of pads that are available for charging to the ZEBRO's. This way, the ZEBRO's can move towards that location and be charged. However, only the charging station has a communication module and has a defined location. The locations of the charging pads have to be determined on the basis of the location and orientation of the charging station. The locations of the pads are not predefined since the system is modular and they can be put into arbitrary positions.

In this subsection all parts implemented in the Raspberry Pi's will be discussed.

- Creating a grid to identify where pads are connected, and which pads are free for charging.

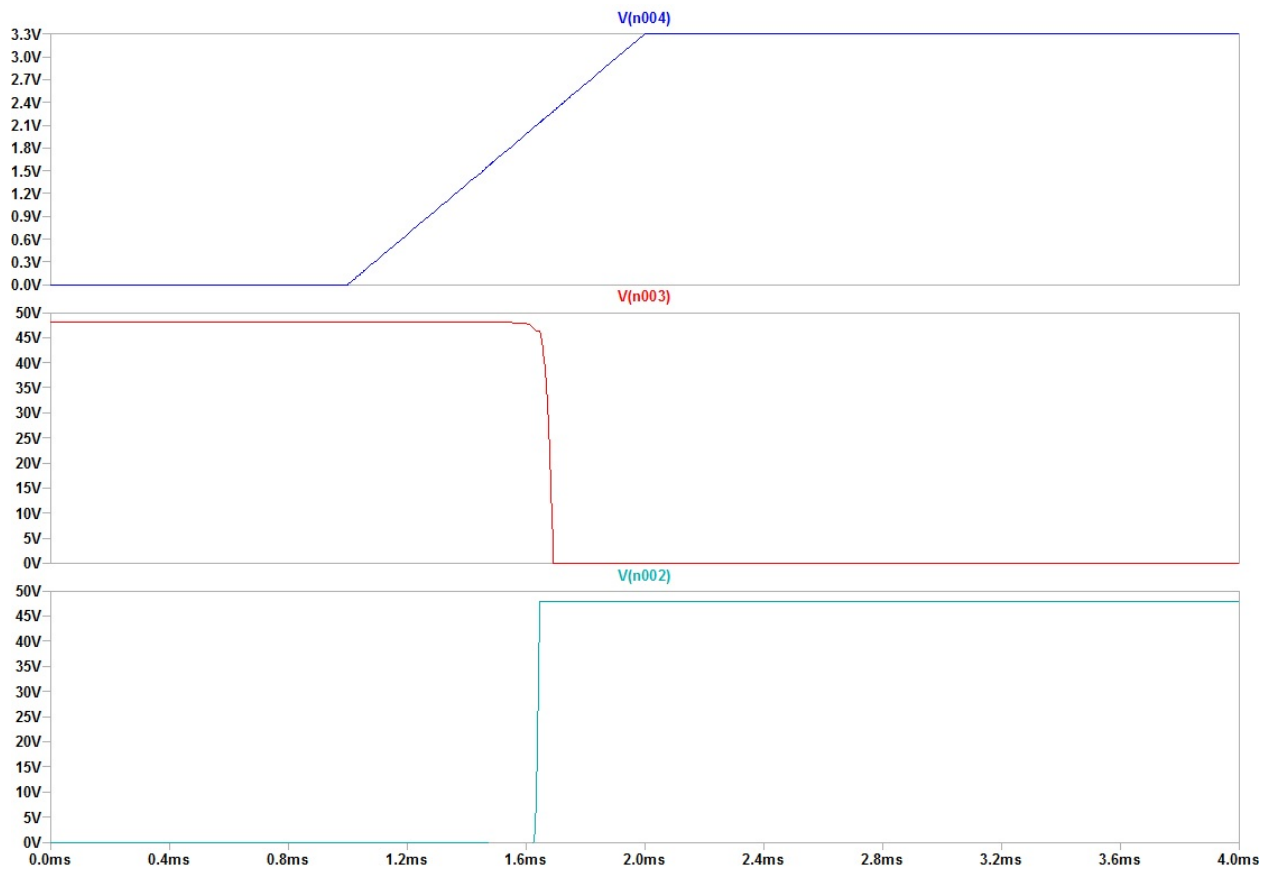


Figure 3.13: Simulation of the load switching with multiple MOSFET's. The displayed signals are the V_{GS} of the NMOS, V_G of the PMOS and V_{load}

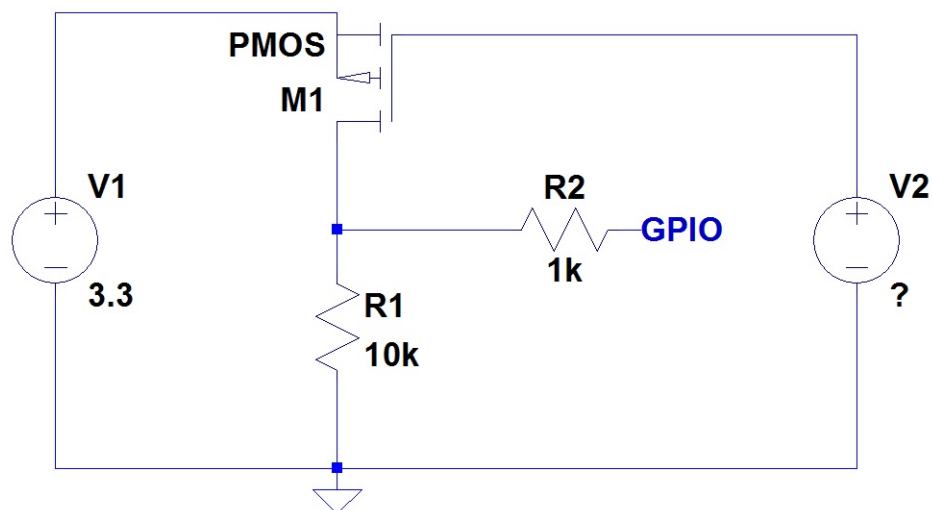


Figure 3.14: Pull up network to define the voltage on the GPIO pin

- A boot up program and an off switch for each individual Raspberry Pi.
- The display on the charging station to check the status of the pads.
- Communication management with the wireless charger.
- Communication with the communication module.

3.4.1. Grid creation

Due to the modularity of the system, it is important to keep the information in the charging station up to date. At any time, a pad can be connected and disconnected, and this needs to be processed. The communication between the charging station and its neighbours can be implemented in several ways. The most obvious communication method is I2C. In this method, each pad can put its status about ZEBRO occupants on the bus, with its own id attached with it. However, determining the location of a pad that is arbitrarily connected is impossible when purely using I2C. Getting an idea of where the pad is connected is possible by adding a direct connection between the GPIO pins of the pads and charging station. But since a direct connection between GPIO pins is needed anyway, it is more logical to implement the grid creation by implementing a serial communication system and lose the I2C.

To achieve serial communication with all neighbours, each pad (and the station as well) needs to have 3 connections with each neighbour; a connection for sending (Rx), a connection for receiving (Tx), and a connection to create a common ground. For these connections, the GPIO pins of the Raspberry Pi's will be used. A general overview of the pin layout of the Raspberry Pi can be found in [Appendix A](#).

The processing behaviour of the charging station and the charging pads can be described in a few states. A drawing for clarification can be found in [figure 3.15](#).

- State 1 The charging station assumes that no pads are connected at the start of this stage. It sends an ask for life signal to all its four sides to find out if anything is connected at all. Before entering state 2, a delay is implemented to make sure that the possible connected pads have had enough time to process the signal and send a sign of life signal back.
- State 2 In the second stage, the charging station received information about whether something is connected on each side. The connected sides are noted and state 3 is initiated.
- State 3 This stage consists of multiple parts, one for each side that is connected. Since the processing of each side can not be done at the same time, it has to be done one after the other. For each side, information is requested from the connected pad. The pad returns information about its status and about the status of its neighbours (The pad follows a similar protocol as the charging station, requesting information from all its neighbours and return the information to the charging station when the information of all its sides is known.). Then, the charging station creates a map of all connected pads, and whether they have room for a ZEBRO.
- State 4 In this final stage, the previously known map is overwritten by the new one. A pad-counter is held to find out if the maximum number of connected pads is exceeded. If it is, this is made clear to the user, that the final pads can not be used. The actual locations of the pads can be calculated and these can be put on the I2C bus to be communicated with the ZEBRO's. Finally, the data is reset, and the loop restarts from state 1 again.

The counter of the number of pads that are connected is send through every pad. So that each pad has information about how many pads are already connected. The fundamentals of the pad protocol are similar to that of the charging station, only state 4 is not implemented in the pads, and several extensions are needed as well to create a working system.

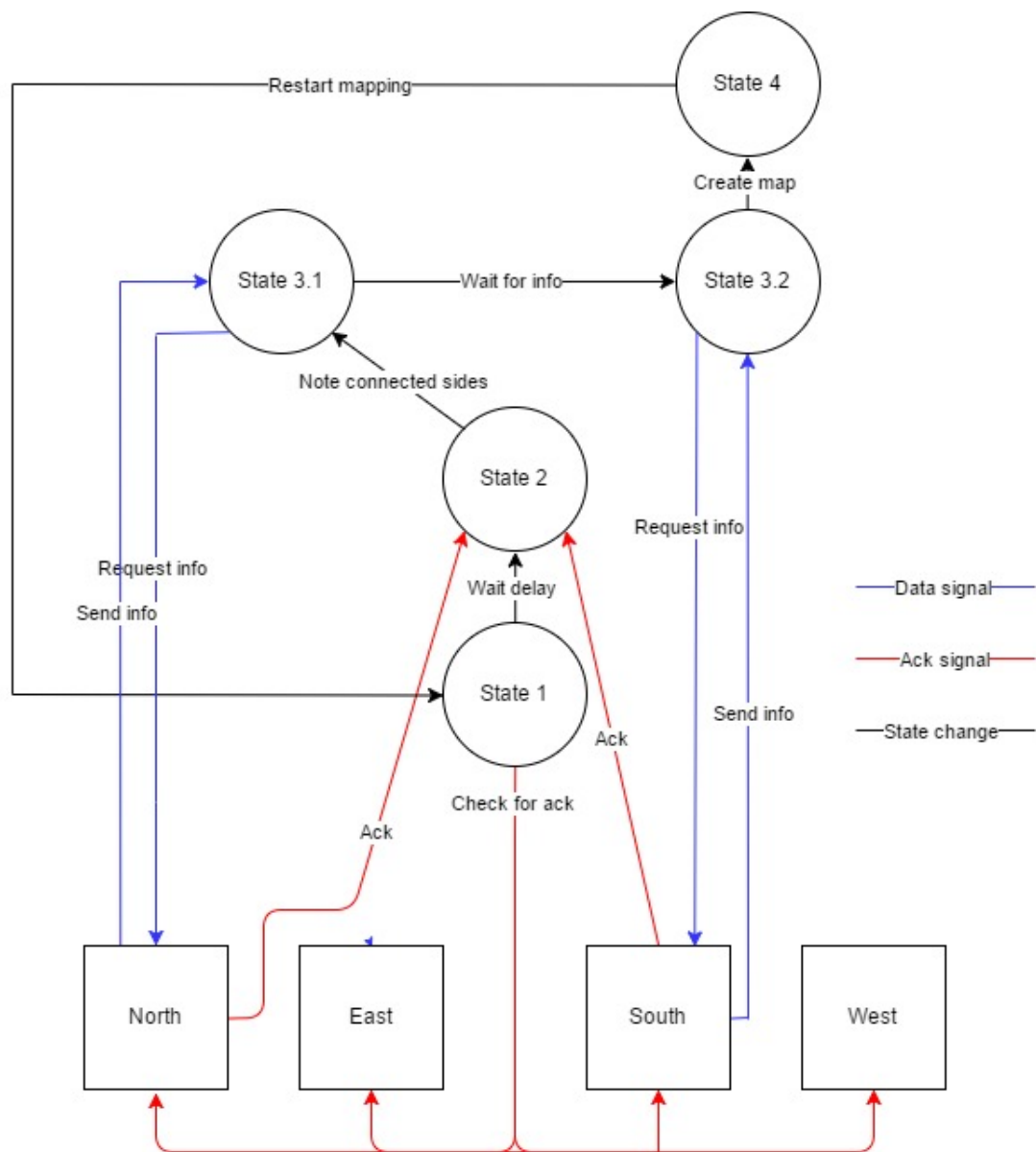


Figure 3.15: An overview of the processing behaviour of the charging station

- The orientation of the pad has to be determined. If the pad is connected to the north side of the charging station, with pins that are defined as the east side of the pad, the orientation in the pad is rotated, such that the previously east defined pins are now defined as pins located on the south side of the pad. With this information, the pad can communicate the orientation with the following pads in a similar way. Knowing this information is crucial, else the grid would contain errors about the whereabouts of the connected pads.
- The pad compares the information of how many pads are connected before it with the maximum number of connected pads that are allowed. If this pad transcends this value, it turns of the switch to the wireless power transfer, blocking power consumption.
- If the pad is within the maximum number of pads, the switch remains turned of, until a ZEBRO appears on top of the pad. Unlike the grid, this information is held and not reset after each grid creation loop. This is crucial since else the switch would be off during the time that the count of the pad is unknown. If the pad receives new information about its count, the old information is overwritten.
- The system needs to take into account that it is possible to create a loop in the grid. (From the charging stations perspective, a pad connected east of the north pad is the same as a pad connected north of the east pad.) The pad checks the received grid and checks if his neighbours are defined as 0's (empty), or something else. If there is something else than a 0, the pad will not communicate in that direction.

3.4.2. Raspberry Pi management

Since there is no visual interface of each Raspberry Pi when the system operates, the Pi's should run the written communication code whenever they receive power.(When they are connected) Since the Pi's have to start up first when receiving power, and the grid creation process needs to run a whole loop as well, it will take a while before the pad is visible for ZEBRO's. (This is a matter of seconds, and is also dependent on the number of pads connected.)

Disconnecting a Pi from its power supply without properly shutting it down first, may cause complications in the Pi. That is why a shutdown button has to be implemented, to make sure the system still works when reconnected. The shutdown button can be implemented between the VCC of the Pi and a GPIO pin.

3.4.3. Communication

As mentioned before, there are several communications that have to be applied. These are; communication with the user, communication with the communication module, and communication with the wireless charger.

Communication with the user can be implemented in several ways. But the most clear way is to use a display to notify the user about what is in the grid. For the first prototype the display used will just be the terminal. A grid will be displayed with numbers with the following meaning;

- 0 There is no pad
- 1 There is a pad, and it is available
- 2 There is a pad, and it is occupied with a ZEBRO
- 3 This is the charging station

Since almost all GPIO pins are used, it is impossible to show this information using a LED array or other type of screen that needs a GPIO connection. I2C is also an option for some displays, but these pins are needed for the communication with the communication module. However,

the Raspberry Pi has an mini-HDMI port to easily connect a regular display. The interface can always be adjusted if necessary.

Communicating with the communication module is done by I2C. The charging station only needs to send information, and does not need any information back. The grid of the charging station is send to the top level in the I2C bus. Here, the location and orientation of the charging station is stored, and the locations of the free pads can be calculated. These are put on the bus, where the communication module transmits this information to ZEBRO's looking for a charging spot.

The final communication is done with the wireless charger. As mentioned before, the electrical part of this communication is done with a pull up network. If a ZEBRO is present, the wireless charger will send a high signal as indication. This is processed in the Raspberry Pi, and the load switch is turned on, allowing current to flow through the wireless charger.

3.5. Total design

When all components of the design phase are put together, it results in figures 3.16 and 3.17.

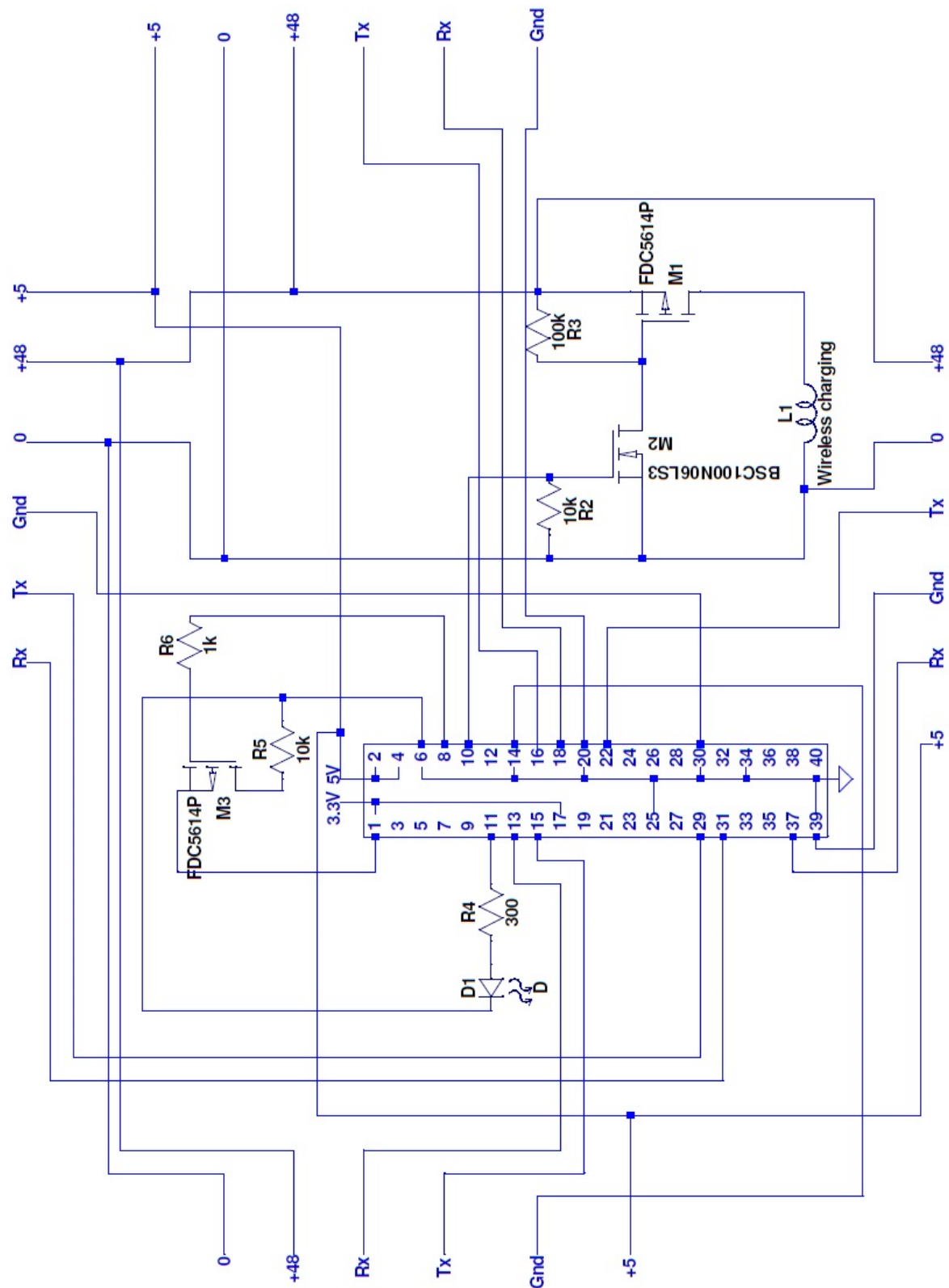


Figure 3.16: Internal structure of a pad

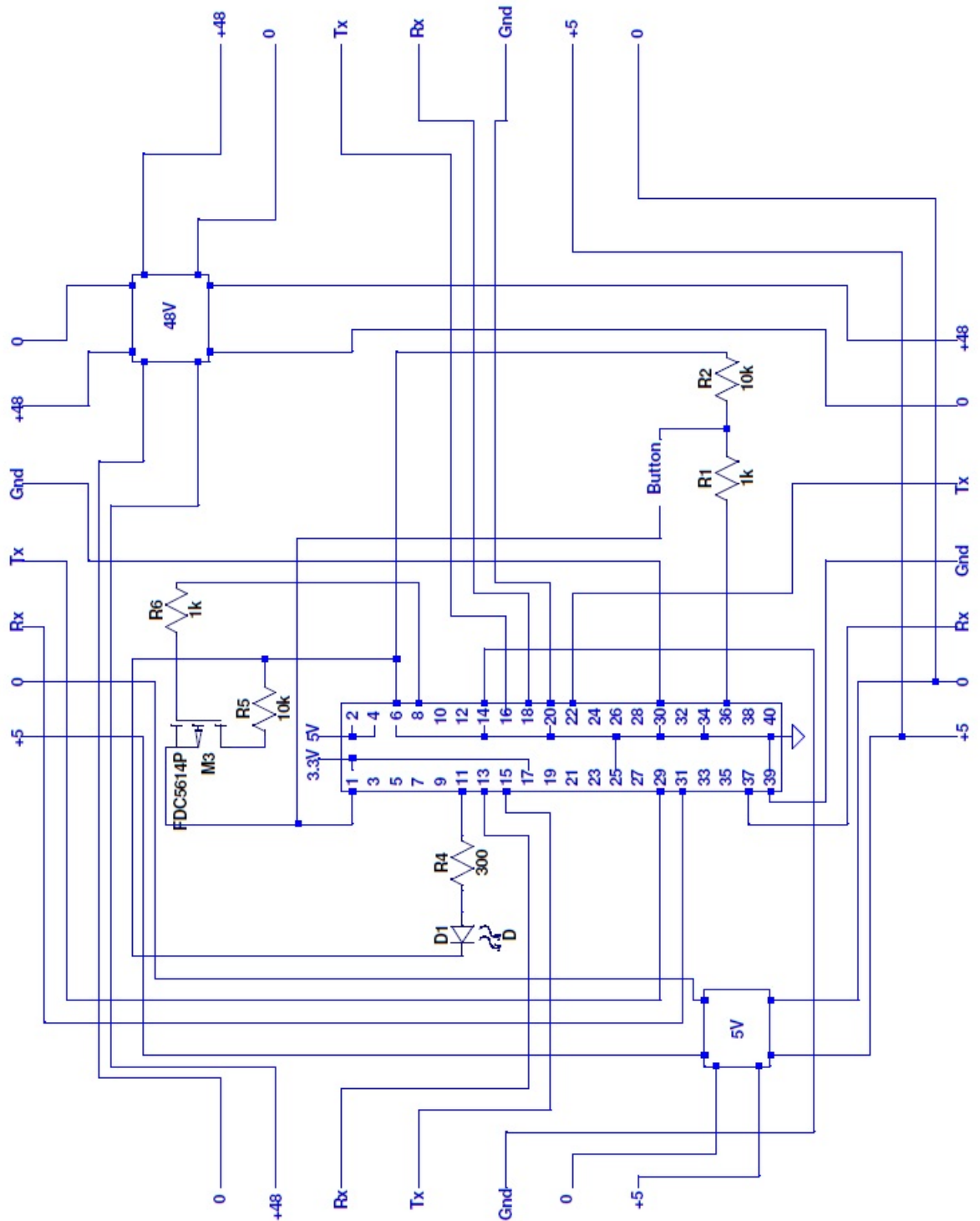


Figure 3.17: Internal structure of the station

4

Prototype implementation

Creating a system that theoretically works is a different thing than actually implementing the hardware. If a system works in theory, building the corresponding hardware implementation is not a simple task as it may seem. In this chapter the prototype implementation will be discussed on the basis of the following components;

- Design of the pads
- Test the grid creation, measuring the speed of the system
- Test whether the load switch works as predicted
- Testing of all components connected together
- A list of all the used components and costs

4.1. Pad layout

The pads need to contain only a few components; a Raspberry Pi, wires, the load switch, an inverter, the coil and, the control of the coil.(the inverter, coil and coil control are not mentioned in this thesis since they are part of the wireless power transfer subgroup.) A diagram of the inner layout of a pad is displayed in figure 4.1. To keep some of an overview, not all the wires are drawn in the diagram. The wires are not laid in circles, else they would act as an antenna that could cause complications.[14]

When connecting the Rx of one pad to the Tx of its neighbour and vice versa, this goes smoothly. However, the 48V of one pad can not be connected to the 48V of another pad, since there are no unisex connectors that are capable of handling high currents. There are a few ways to solve this issue; build male pads with only male connectors and female pads with only female connectors, build pads which are male on 2 sides and female on the other 2 sides, or use 2 connectors for each connection (this will double the amount of connectors). While the last option seems far fetched, it is the only option where the pads can still be connected in any way possible. The total system will not be build for the prototype, so these options are still open.

The size of the pads is very important because this is needed to calculate the exact location of each particular pad. The ZEBRO should be able to stand on a pad with all possible orientations without sticking out. So the width of the pad should be equal to the diagonal of the ZEBRO. Which is 317mm. Some tolerance is taken, the pads will be the size of 350mmx350mm. This is also the size that the charging station will be for continuity.

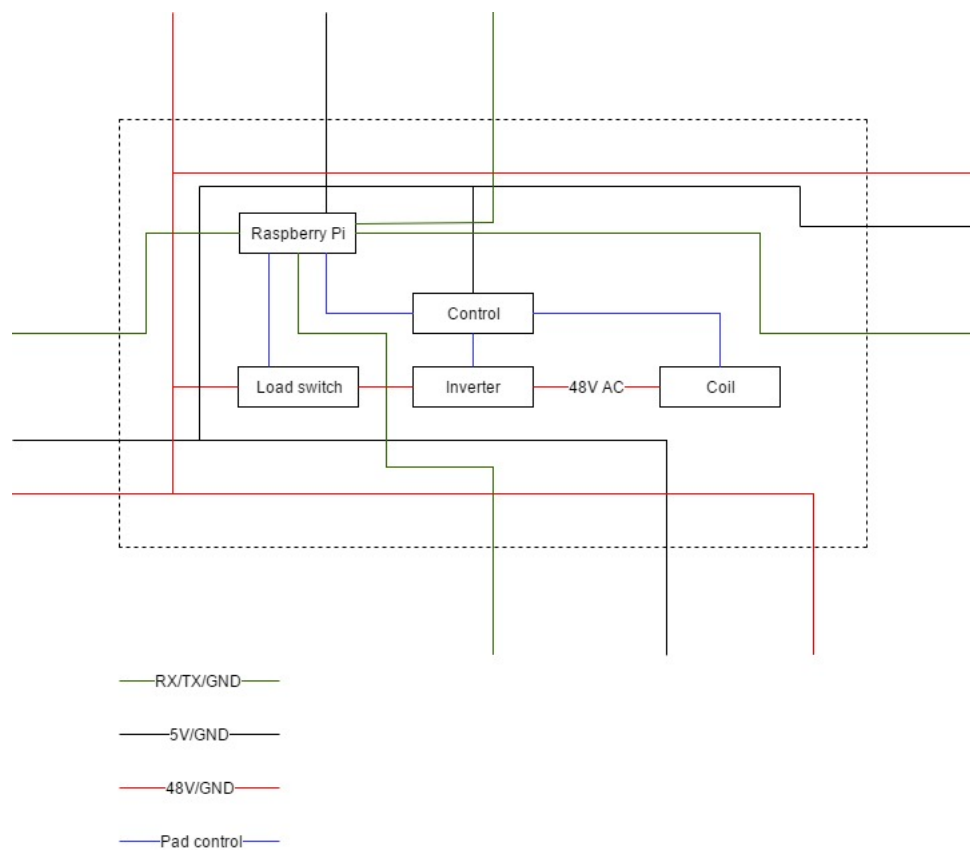


Figure 4.1: A diagram of the inner layout of a pad



Figure 4.2: Schematic test setup for the grid simulation

4.2. Grid simulating

While testing the grid creation, several things have to be tested and measured;

- Test if the charging station is able to create a correct grid
- Test if the state of a connected pad is shown correctly (ZEBRO or no ZEBRO)
- Test if the system notices if too many pads are connected and if measures are taken
- Test if the grid creation still operates when there is the possibility of a loop
- Measure the boot up speed of a Raspberry Pi
- Measure the speed of grid creation

To test all possible scenario's charging station is needed and 4 pads. The test setup can be seen in figure 4.3. The schematic test setup can be seen in figure 4.2. (A set up where all 4 sides of the charging station are connected is tested as well) The test is set up in such a way, that all aspects can be checked. During the test, a switch will be switched to simulate the departure of the ZEBRO. It is expected that this will be shown in the terminal of the charging station. The maximum allowed number of pads is set on 3 in a second test to test the behaviour of the system. The results of the simulation can be seen in figure 4.4. The results of the simulation show that all parts of the system work. the 2, indicating a ZEBRO, changes to a 1 when the switch is flicked. The loop prevention works as well, the correct number of connected pads is displayed.

When too many pads are connected, the system behaves as expected and disconnects the pads that exceed the maximum number of allowed connected pads.

The boot up speed of the charging station with 4 pads connected is measured and is approximately 25s. This includes the first loop as well, so the grid is defined. Each grid creation loop is measured as well and is approximately 3.4s. This may seem long, but speed is not the essence in this process.

4.3. Load switch test

The circuit of the load switch is tested by building the circuit on a small breadboard. For the information signal that is connected on the gate of the NMOS, a voltage supply is used and set on 3.3V. For the powering of the coil, a voltage supply is used and set on 30V. (Higher supply voltages were not available)

The circuit works just as expected. Since the 48V that is used does not approach the maximum allowed 60V given in the datasheet, the system will operate the same under operation circumstances.

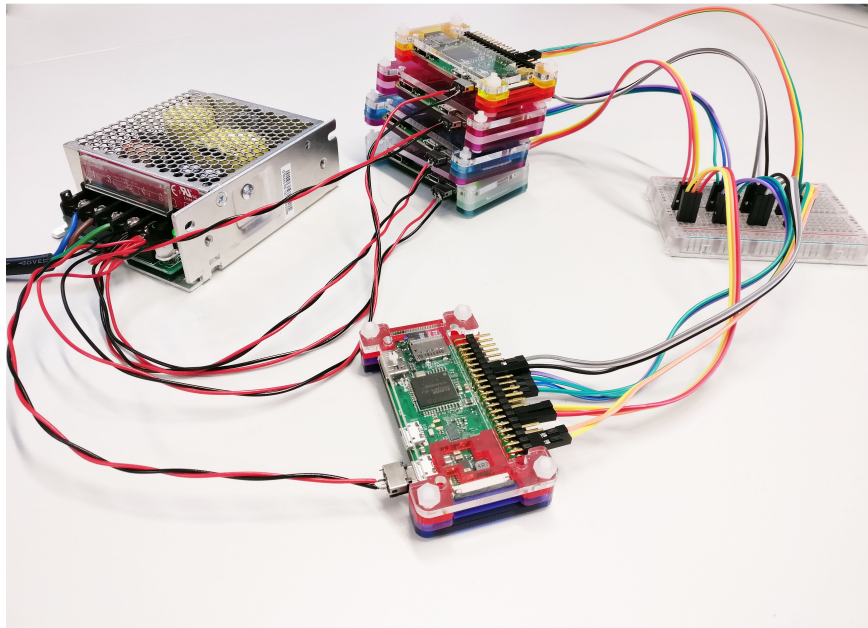


Figure 4.3: Test setup for the grid simulation

```

loopnr: 123
pads connected: 4
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 2 1 0 0 0
0 0 0 1 3 1 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

loopnr: 124
pads connected: 4
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 0 0 0
0 0 0 1 3 1 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

```

Figure 4.4: Result of the grid simulation

Table 4.1: Costs of electrical components of the charging station and pad

	Amount	Price	Total price
Charging station			
Raspberry Pi + SD card	1	19.-	19.-
XT60 connectors	8	0.75.-	6.-
Bullet connectors	16	0.15.-	2.4.-
USB connector	1	1.05.-	1.05
AC/DC 48V	1	113.-	113.-
AC/DC 12V	1	22.7.-	22.7.-
			164.15.-
Pad			
Raspberry Pi + SD card	1	19.-	19.-
XT60 connectors	8	0.75.-	6.-
Bullet connectors	16	0.15.-	2.4.-
NMOS	1	0.53.-	0.53.-
PMOS	2	0.54.-	1.08.-
			29.01.-

4.4. Complete prototype

In the final test, all components will be put together to create a complete prototype of the charging station. Since the wireless charging module is not up and running yet, its behaviour needs to be imitated. As can be seen in figure 4.1, the components of the charging station 'meet' the components on two different occasions. First there is the connection between the load switch and the inverter. And second there is the input signal of the control to the Raspberry Pi. If these signals can be properly received and transmitted, the whole autonomous charging station and charger module will work when the other sub parts are completed.

The signal coming from the control of the wireless power transfer module will be simulated with a switch. The load switch will be connected between the Raspberry Pi and a LED to simulate the connection to the inverter. A complete setup of the prototype implementation can be seen in figure 4.5.

In this figure, the 4 pads and the charging station in the middle represent the system together with the wall socket. The larger 2 squares are purely to indicate how the pads and charging station are connected to each other. Since using a power supply of 48V is not allowed, a power supply of 30V is used for a proof of concept. As mentioned earlier, this does not influence the results from the prototype test. The pads are formed in such a figure, that the loop detection can be shown to work as well.

A detailed description on how to install the Raspberry Pi's and set up the whole system to work can be found in Appendix B.

4.5. Components and costs

All electrical components and their costs used in building the autonomous charging station are displayed in table 4.1, to give an idea of what the total price of the charging station will be. There will be some additional cost for components where the prices are not known yet and miss in the table such as: casing for the charging station, casings for the charging pads, wiring and LED's.

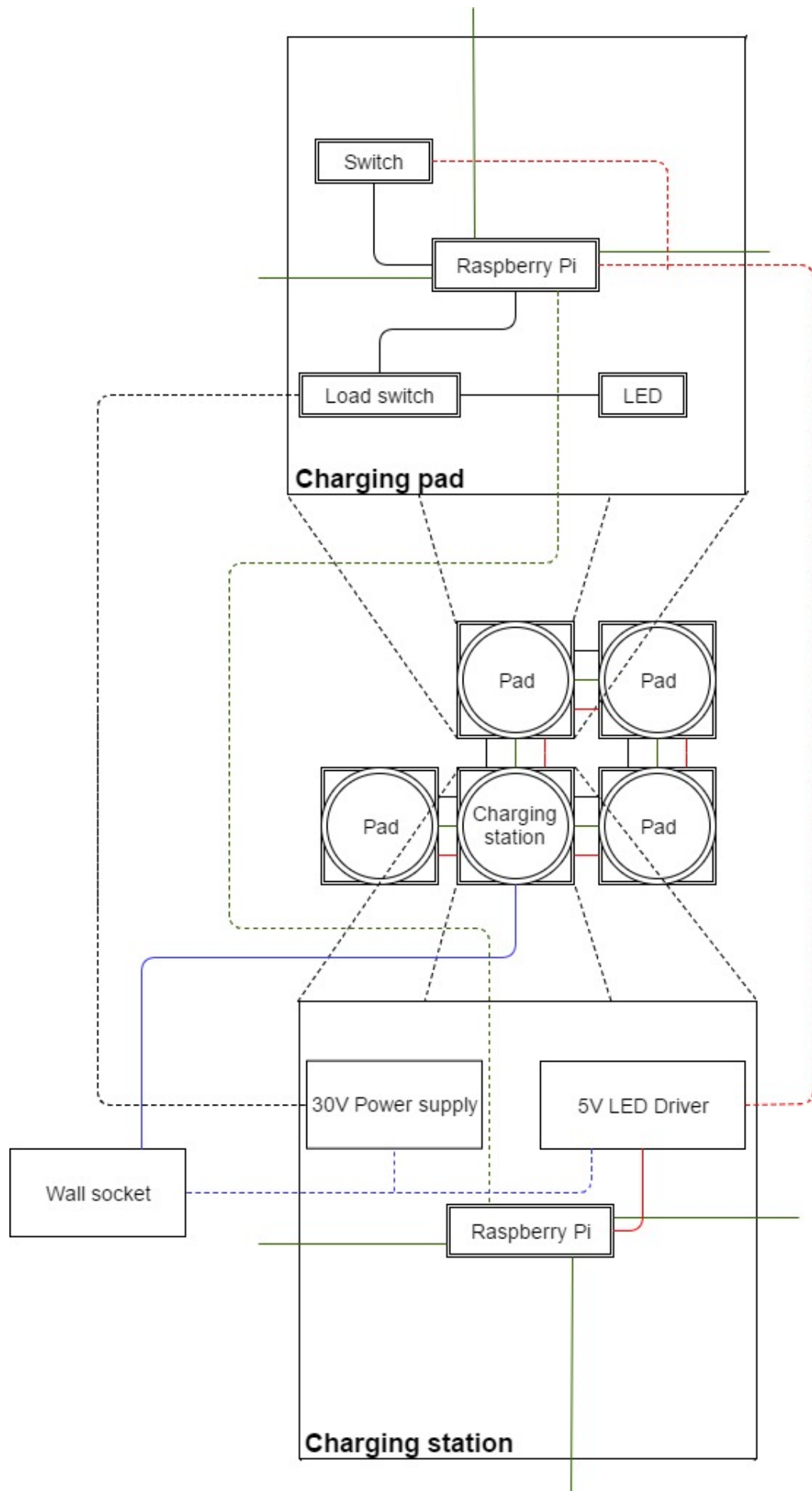


Figure 4.5: Flowdiagram of the complete prototype implementation

5

Conclusion

The autonomous charging station is designed to meet the program of requirements. Which succeeded on most fronts. Some important plus points of the design are the modularity and the possibility to expand the charging station to the kilo ZEBRO. However, the design can still be optimized, the assembly will take a considerable amount of time, and there are still some things lacking or requiring improvement.

First of all the physical exterior of the charging station and the pads is still missing. Since the diameter of the ZEBRO is $317mm$, the dimensions of the charging station and pads need to be minimal $317x317mm$. It is recommended to take a slight larger size for more tolerance so the ZEBRO's won't overlap on the next pad.

Second, swapping the Raspberry Pi's with pre-programmed micro-controllers is a possibility. The Pi's are used to test the program and research all possibilities for grid creation. But with the system complete, they are not essential.

Next, the code used could do with some improvements. The whole code consists of a single file, making it very unclear, it could be divided into several sub files for more clarity. And although it is perhaps not necessary, there may be a possibility to optimize the speed of the grid creation. Due to some unsolved complications with the data transfer, information is send per 2 bits instead of per byte. Which slows down the speed of the total data transfer considerably. There may also be a possibility to create a simultaneous process for the grid creation to each side of the charging station. (At this moment the charging station waits on the information of one side before starting communication with the next) However, this will cause for some problems with the pad count, since they won't receive information one after the other anymore, but it is something that could be looked at.

Finally, there still is no option to adjust the maximal number of pads from the outside. While the current limit is set to 4, if an additional power supply were to be added to the charging station, the limit would have to be extended to 8. This could be done either by creating a hardware system that can somehow measure the amount of power supply's connected and send this information to the control of the charging station. Or, it can be implemented in the software, where a signal of the user is needed to adjust the maximum number of allowed pads.



Raspberry Pi pin layout

Figures [A.1](#) and [A.2](#) display the pinlayout of the Raspberry Pi's. It is important that sides such as north and east are not switch, since this will result in a malfunctioning system. PIN displays the physical pin number and BCM displays the pin number that needs to be used in the code to address the pin.

Application	#BCM	#PIN	#BCM	Application
	3.3V	1	2	5.0V
	2	3	4	5.0V
	3	5	6	Gnd
	4	7	8	14 ZEBRO Status
	Gnd	9	10	15 Enable coil
Status LED	17	11	12	18
RX West	27	13	14	Gnd Gnd West
TX West	22	15	16	23 TX East
	3.3V	17	18	24 RX East
	10	19	20	Gnd Gnd East
	9	21	22	25 TX South
	11	23	24	8
	Gnd	25	26	7
	0	27	28	1
TX North	5	29	30	Gnd Gnd North
RX North	6	31	32	12
	13	33	34	Gnd
	19	35	36	16
RX South	26	37	38	20
Gnd South	Gnd	39	40	21

Figure A.2: Pinlayout of a charging pad

B

Installation manual

In this appendix is described how the charging station and pads can be put together and how they should be operated. The description is written in such a way that anyone without background knowledge is able to install the charging station.

Tools:

- Micro USB charger
- USB keyboard and mouse
- USB to micro USB cable
- USB hub
- Mini HDMI to HDMI adapter
- HDMI cable
- Monitor
- Soldering iron
- Tin
- Micro SD card reader
- Wire strippers

B.1. Station

Components

- Raspberry Pi zero w
- Micro SD card of at least 8GB
- 5V DC converter, 40W
- (1 or 2) 48V DC converter, 500W
- Micro USB connector

- Mini HDMI to HDMI adapter
- Monitor
- Wooden platform $350mm \times 350mm$
- A 2×20 header with a 90° bending
- Wire red
- Wire black
- Wire red that can conduct at least $21A$
- Wire black that can conduct at least $21A$
- 8 XT60 connectors
- 4 data connectors
- Green LED
- 300Ω resistor
- Push button
- $10k\Omega$ resistor
- $1k\Omega$ resistor

Pi installation:

1. Install raspbian with pixel on the SD card (8GB)
2. Insert SD card in the Raspberry Pi
3. Connect a screen, mouse and keyboard and the power adapter
4. Connect to the internet
5. Install PiGPIO: open the terminal and type:
 - 5.1. `rm pigpio.zip`
 - 5.2. `sudo rm -rf PiGPIO`
 - 5.3. `wget abyz.co.uk/rpi/pigpio/pigpio.zip`
 - 5.4. `unzip pigpio.zip`
 - 5.5. `cd PiGPIO`
 - 5.6. `make`
 - 5.7. `sudo make install`
6. Install WiringPi by following the following instructions:
<http://wiringpi.com/download-and-install/>
7. Download the main executable for the station from the ZEBRO git and put it in a location of your choosing. The path to the folder containing the file will be called 'location' from now on
8. Edit `/etc/rc.local` by typing the following in the terminal: `sudo nano /etc/rc.local`
9. Scroll down and type the following above 'exit 0'

- 9.1. `sudo pigpiod`
- 9.2. `cd 'location'`
- 9.3. `./main`
- 9.4. save and exit
10. Go to preferences and turn on boot to CLI
11. Shutdown the Raspberry Pi and remove the keyboard, mouse and screen

Hardware installation

- Secure the 48V DC converter to the station
- Secure the 5V DC converter to the station
- Secure the Pi on the station
- Secure the screen to the station
- Place circuits for the shutdown and status LED (green LED and a resistor, and button and resistors)
- Connect wires to the Pi according to figure [A.1](#)

B.2. Pads

Components

- Raspberry Pi Zero w
- Micro SD card of at least 8GB
- Micro USB pinnen
- Micro USB connector
- Load switch
- Wooden platform (340mm x 340mm)
- A 2x20 header with a 90°bending
- Wire red
- Wire black
- Wire red that can conduct at least 21A
- Wire black that can conduct at least 21A
- 8 XT60 connectors
- 4 data connectors
- Green LED
- Red LED
- 2x300Ω resistors

Pi installation

1. Install raspbian with pixel on the SD card (8GB)
2. Insert SD card in the Raspberry Pi
3. Connect a screen, mouse and keyboard and the power adapter
4. Connect to the internet
5. Install PiGPIO: open the terminal and type:
 - 5.1. `rm pigpio.zip`
 - 5.2. `sudo rm -rf PiGPIO`
 - 5.3. `wget abyz.co.uk/rpi/pigpio/pigpio.zip`
 - 5.4. `unzip pigpio.zip`
 - 5.5. `cd PiGPIO`
 - 5.6. `make`
 - 5.7. `sudo make install`
6. Install WiringPi by following the following instructions:
<http://wiringpi.com/download-and-install/>
7. Download the main executable for the pad from the ZEBRO git and put it in a location of your choosing. The path to the folder containing the file will be called 'location' from now on
8. Edit /etc/rc.local by typing the following in the terminal: `sudo nano /etc/rc.local`
9. Scroll down and type the following above 'exit 0'
 - 9.1. `sudo pigpiod`
 - 9.2. `cd 'location'`
 - 9.3. `./main`
 - 9.4. save and exit
10. Go to preferences and turn on boot to CLI
11. Shutdown the Raspberry Pi and remove the keyboard and the mouse

Hardware installation

- Secure the Pi on the pad
- Place circuits for the coils and the status LED (red LED and a resistor, switch and resistors, green led and a resistor)
- Place the coil
- Place the load switch
- Connect wires to the Pi according to figure [A.2](#)

B.3. Use

Place a maximum of 4 pads to the charging station, this can be done in anyway you like, for instance 4 pads to the station, one pad to the station with 3 other pads connected to it, one row of pads, or even a loop.

Connect the following wires on each side where a pad neighbours a pad or the station neighbours a pad:

- Rx -> Tx
- Signal ground->Signal ground
- Tx-> Xx
- +5V -> +5V
- 5V ground -> 5V ground
- +48V -> +48V
- 48V ground -> 48V ground

Plug in the charging station and turn on the screen When the status LEDs turn on and the screen shows a grid and information, the station is functioning.

The screen shows a grid, on this grid a few numbers are shown. they have the following meaning:

- 0 No pad
- 1 Available pad
- 2 Occupied pad
- 3 Main station

If you want to add or remove pads, it is recommended to turn off the charging station. Unplug it and wait for 10 seconds to let the DC converters discharge.

Turning it off: Do not unplug instantly! this can corrupt the files on the raspberry pi's Click on the off button and wait for all the status LED's to turn off. The device can now be unplugged

Problems: When the grid on the screen doesn't show the pads that are connected correctly, use the proper shutdown procedure, make sure Tx is connected to Rx, and restart.

Bibliography

- [1] R. Kollman and J. Betten, *Powering electronics from the usb port*, Texas Instruments Incorporated , 28 (2002).
- [2] F. Capelli, C. Abomailek, J. R. Riba, and J. Sanllehí, *Analysis of electrical contact resistance models for substation connectors*, in [2016 IEEE International Conference on Power System Technology \(POWERCON\)](#) (2016) pp. 1–6.
- [3] F. Capelli, C. Abomailek, J. R. Riba, and J. Sanllehí, *Analysis of electrical contact resistance models for substation connectors*, in [2016 IEEE International Conference on Power System Technology \(POWERCON\)](#) (2016) pp. 1–6.
- [4] F. Zhang, S. Dong, and X. Yin, *Analysis of the wireless charging's multiple coils structure and efficiency based on magnetic field*, in *2016 19th International Conference on Electrical Machines and Systems (ICEMS)* (2016) pp. 1–4.
- [5] W. Ren, H. Zhi, S. Xue, G. Zhai, and J. Song, *Numerical simulation and experimental verification for contact spot temperature and electrical contact resistance of rivet contacts*, in *ICEC 2014; The 27th International Conference on Electrical Contacts* (2014) pp. 1–5.
- [6] *IEC 62109-1*, International Electrotechnical Commission (2010).
- [7] R. P. FOUNDATION, [Raspberry pi hardware](#), .
- [8] S. Luo, Z. Ye, R.-L. Lin, and F. C. Lee, *A classification and evaluation of paralleling methods for power supply modules*, in [30th Annual IEEE Power Electronics Specialists Conference. Record. \(Cat. No.99CH36321\)](#), Vol. 2 (1999) pp. 901–908 vol.2.
- [9] *600W with PFC and Parallel Function*, MEANWELL (2010).
- [10] *Enclosed Power Supplies TXM Series, 15–150 Watt*, Traco Power (2015).
- [11] *Using MOSFETs in Load Switch Applications*, ON Semiconductor (2014), rev. 1.
- [12] *FDD5614P 60V P-Channel PowerTrench MOSFET*, FAIRCHILD (2015), rev. 1.4.
- [13] *OptiMOSTM3 Power-Transistor*, infineon (2013), rev. 2.3.
- [14] A. Emleh, H. C. Ferreira, A. J. H. Vinck, and A. Snyders, *Received noise on powerline communications where the in-building wiring acts as an antenna*, in [AFRICON, 2011](#) (2011) pp. 1–6.