Universität Leipzig
Fakultät für Mathematik und Informatik
Institut für Informatik

Murmecha: a low-cost robot platform designed for investigating stigmergic swarm behaviour

Master Thesis

By David Schulte
Informatik
Leipzig: November, 2024

Prof. Middendorf, Fakultät für Mathematik, Institut für Informatik

Abstract

In this thesis a robotics platform designed for use in swarm experiments is developed. The focus is placed on stigmergy and its implications on swarm behaviour. The developed robots are able to interact with their environment and modify it. This modification is done by emitting UV-light at a wavelength the environment is sensitive to. The robots are also equipped with sensors designed specifically to observe these modifications. The developed robots are also adaptable for use in other experiments in a similar environment.

Selbstständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann. Ich versichere, dass das elektronische Exemplar mit den gedruckten Exemplaren übereinstimmt.

Leipzig, den 5.11.2024

Contents

1 Introduction **1**
1.1 Context . . . 1
1.2 Description . . . 1
1.3 Structure . . . 2

2 Analysis **3**
2.1 Stigmergy in swarm robotics . . . 3
2.1.1 Electronic pheromone emulation . . . 3
2.1.2 Physical methods of emulating pheromones . . . 4
2.2 Other robot platforms . . . 4
2.2.1 Vibration driven robots . . . 4
2.2.2 Wheel driven robots . . . 5
2.2.3 Legged robots . . . 6
2.3 Desired applications . . . 6
2.3.1 Line following . . . 6
2.3.2 Detection of ground type . . . 7
2.3.3 Direct interaction with other robots . . . 7
2.4 Requirements . . . 8
2.4.1 Movement . . . 8
2.4.2 Ground sensing . . . 8
2.4.3 Ground interaction . . . 9
2.4.4 Environment sensing . . . 9
2.4.5 General requirements . . . 10

3 Component evaluation **11**
3.1 Microcontroller . . . 11
3.2 Sensing . . . 12
3.2.1 Movement and orientation . . . 12
3.2.2 Distance to objects . . . 12
3.2.3 Brightness and colour . . . 13
3.2.4 Battery voltage measurement . . . 14
3.3 Moving and Interacting . . . 15
3.3.1 Modes of locomotion . . . 15
3.3.2 Motors . . . 16
3.3.3 Geared stepper motors . . . 17
3.3.4 Motor drivers . . . 18
3.3.5 LED drivers . . . 19

4 Implementation **23**
4.1 Sensors . . . 23
4.1.1 Measuring analogue values . . . 23
4.1.2 Light sensors . . . 24
4.1.3 Colour sensor . . . 26
4.1.4 Inertial Measurement Unit . . . 26
4.1.5 Infrared Sensors . . . 27
4.1.6 Battery monitor . . . 29
4.2 Motion . . . 29
4.2.1 Wheels . . . 29
4.2.2 Wheel design . . . 30
4.2.3 Wheel and motor assembly . . . 31
4.2.4 Required motor torque . . . 31
4.2.5 Stepper motor and driver . . . 33
4.3 Interacting with the ground . . . 34
4.4 Battery charging and power supply . . . 36
4.4.1 Battery . . . 36
4.4.2 Power conversion . . . 36
4.5 Internal communication . . . 37
4.6 External communication . . . 38
4.6.1 Wired communication . . . 38
4.6.2 Wireless communication . . . 38
4.7 Possible expansions . . . 39
4.7.1 Planned expansion . . . 39
4.7.2 Standard interfaces . . . 39

5 Evaluation **41**
5.1 Individual Systems . . . 41
5.1.1 Battery life . . . 41
5.1.2 Straight-line Movement . . . 41
5.1.3 Manoeuvrability . . . 43
5.2 Combined tests . . . 46
5.2.1 Line following . . . 47
5.2.2 Variance between robots . . . 51
5.3 Problems . . . 52
5.3.1 USB connector . . . 52
5.3.2 Wi-Fi connectivity . . . 52
5.4 Future improvements . . . 53
5.4.1 Motion . . . 53
5.4.2 Wheel rotation . . . 53
5.4.3 Light sensing . . . 53

5.5 Conclusion 54

A Hardware documentation **63**
A.1 Part list 63
A.2 Pin layout 64
A.3 I^2C Addresses 66

B Software **67**

List of Figures

1.1 Image of the developed robot 2
2.1 Placement of light sensors as seen from above. 7
3.1 Schematic of a phototransistor 14
3.2 Signals required to drive a stepper motor 18
3.3 LED circuit with an NPN-transistor 20
4.1 Circuit for measuring phototransistor current 24
4.2 Placement of components on the PCB of the robot 27
4.3 Placement and orientation of the IR-sensors on the robot 28
4.4 Perspective view of a wheel 30
4.5 Wheel and motor assembly 31
4.6 Circuit to change micro-stepping mode 34
4.7 Driver circuit for the UV-LED 35
4.8 Circuit for switching between power source when charger is connected 36
5.1 Battery discharge curve 42
5.2 Paths followed when command to move in a straight line 43
5.3 Straight-line velocity of the robot 44
5.4 Trajectory when moving in a circle 45
5.5 Trajectory when moving in a square with sides of 250mm 46
5.6 Trajectory when moving in a square with rounded corners 47
5.7 Example of path used for line following 49
5.8 Distribution of velocities while following line 51

List of Tables

5.1 Results of the line following experiment 50
5.2 Measured brightness for different robots 52
A.1 GPIO pin layout 65
A.2 List of I^2C addresses 66

1 Introduction

Swarm robotics present an interesting approach to understand swarm behaviour in animals and for exploration of difficult to reach places such as other planets or the deep sea. To study these swarms, experiments where robots interact with a limited set of instructions and programmed behaviours are performed.

An interesting subject to study is how agents can interact through the use of their environment. In Biology ants can produce complex behaviours as a swarm by communicating information through pheromones as described in [1]. The robots will be used to study if such emerging behaviours are also possible for man-made machinery. A method of interaction with the environment has already been chosen. The robots can leave trails of light on the ground by energizing a phosphorescent surface.

The robots used in these experiments need to be able to interact between themselves and move on the given surface. Given the nature of these experiments is to study swarm behaviours a large number of robots is required to reproduce swarms. This presents a challenge as many robotics platforms are costly. Additionally larger robots require a larger space to be able to move freely.

It is the aim of this work to study the requirements for a robot in stigmergic swarm experiments and to develop said robot.

1.1 Context

This work focuses on robots designed for stigmergic experiments. In the experiments the robots move on a photosensitive surface and can interact with their environment through this surface. The main focus of this thesis is the development of a robot platform suited for experiments involving two such photosensitive surfaces. The primary surface is a phosphorescent material that emits light after being exposed to UV-light. A secondary surface can also be used where a photoswitch molecule changes colour when hit by a specific wavelength of UV-light.

The robots developed are thus suited for the environment used locally. However they should be adaptable to any setup using light as the means of interaction with the ground. One may however need to modify some components to match the given environment.

1.2 Description

The developed robot is small to fit the arena the experiments will be carried out in. Two wheels are used to propel the robot and a metallic ball is used to keep three points of contact with the ground for stability.

The developed robot has a circular footprint with a diameter of 75mm at the widest part of its dome-shaped upper body and a height of 59mm above the ground at its highest point. Its two wheels have an outer diameter of 20mm. Figure 1.1 shows a picture of a developed robot.

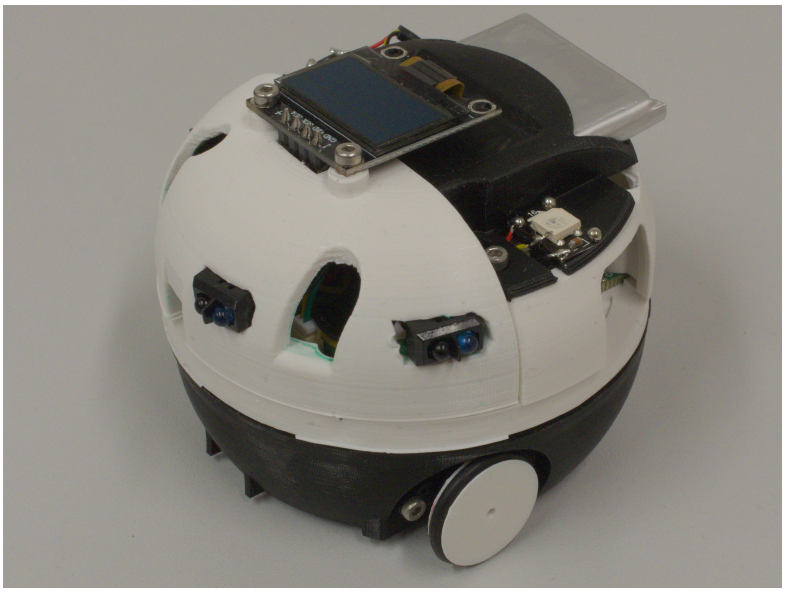


Figure 1.1: Image of one of the developed robots.

The robot also has sensors for measuring brightness, both ambient and on the ground. The distance to other robots and obstacles can also be measured by some distance sensors to complete the information obtained about the robots' environment.

1.3 Structure

In a first step an overview of existing swarm experiments is undertaken. From there the requirements for the robotics platform will be established. Once these requirements are known, further research into the required components on the robots will follow. In this part the components to be used on the robots are selected. Later, in another step, the implementation of these components on the robots will be discussed. Finally the developed robots will be tested on a set of oft-performed tasks.

2 Analysis

In order to develop a number of requirements for the robot, at first an overview of existing implementations of stigmergy in robotic swarms is presented. Then other, existing, robot platforms are studied and evaluated. From there a list of desired applications for the new platform is developed. This leads to the requirements for this new platform.

2.1 Stigmergy in swarm robotics

Stigmergy does not require any centralized method of communication between agents. In nature this process is often realized by the use of pheromones [1–3]. Over the years different methods of emulating these pheromones have been developed. In more recent works even the emulation of multiple pheromone types as been shown as in [4]. In this section a short overview of different methods of pheromone emulation is presented.

2.1.1 Electronic pheromone emulation

Many methods of emulating pheromones require specialized electronics to be present in the environment. This can be RFID-tags such as in [5,6] or computer-controlled screens or projectors [7–9]. Using specialized equipment to emulate pheromones allows for maximum control over the pheromones properties.

Screens and projectors When using screen or projectors to emulate pheromones a computer programme is used to simulate the pheromones' behaviour. This programme can be modified to achieve any required properties of the pheromone, as long as the used hardware can display it. For example in [7] the rate of evaporation can be tweaked to suit the experiments requirements. In most of these systems the robots send the information of depositing pheromone over the network to a computer modelling their environment.

RFID tags Another method for emulating pheromones is the use of RFID tags buried in the surface the robots move on [5]. These tags contain a small amount of memory that can be read and written to by the robots. The information about the pheromone is stored distributed across the tags.

Robot as pheromone A oft-used approach to emulating pheromones is the use of either dedicated or normal robots as markers for pheromone deposits [10,11]. In these methods some robots remain stationary and act as the pheromone deposits

for other robots. This method has a significant drawback as part of the robot swarm has to be dedicated to remain stationary and not participate in the task at hand.

2.1.2 Physical methods of emulating pheromones

Other methods for stigmergy are based in physical systems. These exploit markings left by other robots to communicate. An early example for such a method is leaving a heated trail behind the robot [12]. This method presents a problem for small robots as the power consumption is high. In [12] a halogen bulb with 70 W of power is used to leave the trail.

Substance deposition Some methods aim to reproduce pheromones as seen in nature be depositing substances in the environment. This yields realistic results as the substances behave similarly to the pheromones deposited by animals. A prominent example of such a substance is a mixture of alcohol and water as in [13]. The detection of such volatile compounds is done with often expensive sensors.

Other approaches use non-volatile compounds such as ferro-fluid in [14]. This fluid is magnetized by specialized hardware on the robot and then deposited on the ground [14]. Following robots measure the variance in the magnetic field with widely available magnetic sensors. The magnetisation of the deposited substance gradually decreases over time, simulating the evaporation of the pheromone [14].

Photosensitive surfaces Another method used for simulating pheromones is a surface that reacts to light. Such a material can change its colour when hit with a certain wavelength of light as in [15] where two different colours can be obtained. Besides colour-changing surfaces, phosphorescence is also a possibility [16]. Photosensitive surfaces have the advantage of not requiring any specialized hardware on the robot. An LED is sufficient for creating the deposit of pheromones.

2.2 Other robot platforms

Some similarly sized robotics platforms have already been developed. While some use vibrations [17, 18] as a means to movement, others use wheels.

2.2.1 Vibration driven robots

Some examples of robots using vibrations to move are the Dezibot [17] and the Kilobot [18]. These robots can have simpler construction as fewer parts are required than when using wheels. However this also comes with downsides, as the

movement can be chaotic if the external conditions change. This can already occur in simple systems [19].

Vibrations can also render certain sensors more difficult or impossible to use. An example for this is an inertial measurement unit (IMU). In this case the vibrations can mask any signal one measures as their amplitude is often much higher than the signal itself.

Kilobot The Kilobot is a widely-used robotics platform in swarm experiments. It uses vibration motors to move on brass legs. However the estimated cost of $14 is based on an order quantity of 1000 robots [18]. This number is not often needed thus the price is usually higher. The Kilobot also lacks significant extensions.

Dezibot Though this robot is in practice cheaper than the Kilobot, a definite lack of availability can be noted. It also does not perform and requires extensive work to get going. The Dezibot also uses vibration motors to move [17]. However its movements can be quite unpredictable.

2.2.2 Wheel driven robots

Other robots use wheels to move. These have the advantage of being easier to predict than motion induces by vibrations. Wheels however often add mechanical complexity and thus cost. Some examples of robots using wheels include the E-puck [20], Kephera [21], the MICAbot [22] and the TinyTeRP [23].

E-puck and Kephera Both robots are similar and use two side-mounted wheels to drive [20]. However the relatively high cost 550 € [20] per robot of the E-puck would limit the size of the swarm.

The Kephera is an older platform developed in 1991 [21]. With the passage of time much better microcontrollers became available and the processing power has increased by orders of magnitude. Therefore the Kephera is limited when compared to more modern options.

MICAbot The MICAbot is a wheel-driven robot where the wheels are mounted towards the front of the robot [22]. It is also an older platform already introduced in 2003 [22]. It also lacks the expandability of the E-puck where extra modules can be fitted [20].

TinyTeRP The TinyTeRP robots are the smallest at $17 \times 18 \times 20\text{mm}^3$ [23]. Despite their small size they present the ability to add extensions [23]. However

these extension can only be added to the top of the platform as the motors and drive gears are located below the main Printed Circuit Board (PCB).

2.2.3 Legged robots

Another option for moving is legs. Some robots use legs to move. The Spiderino presented in [24] uses legs.

Legs present a challenge both in the mechanical implementation as well as in the experimental use of the robots. Walking is mechanically complex and legs thus require a great amount of parts to work efficiently. Another problem presented by legs is the high elevation of the underside of the robot from the ground. This would lead to dispersion of the light emitted by the UV-LED. Additionally legged locomotion often leads to a gait where both the distance to and angle with the ground change during each step. This would also lead to an uneven trace left by the UV-LED.

2.3 Desired applications

The robot will be used in swarm experiments on a surface that can react to UV-light. They can use this surface to communicate simple pieces of information. Additionally the robots need to display information for the camera-based tracking system used to record the experiments [25]. It should also be possible to add attachments to the robots if future experiments require further means of interaction.

In the following sections a few, more specific, tasks the robots should be able to achieve are described.

2.3.1 Line following

Many swarm experiments involve following other robots or communication via the environment. In many setups this is achieved by modifying the ground the robots are on. These modifications are usually done by changing the brightness of the ground. So the robots should be able to detect changes in the ground's brightness.

Additionally the robots often will follow lines drawn by other robots. This task should be as straight-forward as possible to implement in software. To make following lines a trivial exercise three sensors can be used, one centred one offset to the left and one offset to the right as seen in Figure 2.1. When the robot is on the line the centre sensor will measure a high value while both side-sensors will measure lower values. When the line is offset below the robot, the side-sensor to which the line is offset will measure a higher value and the robot can steer accordingly.

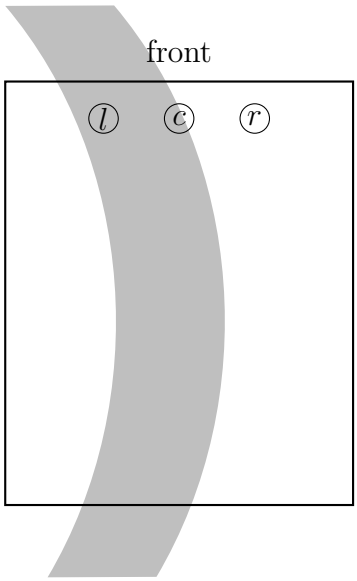


Figure 2.1: Placement of the light sensors as seen from above. This placement allows for easy line-following. The robot can follow a line by using the difference in brightness between sensors to compute the correction it needs to apply. Here the left sensor measures a high signal while the right one measures no signal.

In the example shown in Figure 2.1 the left sensor would measure a high signal while the right one would not measure any signal. To correct this the robot would need to turn to the left.

2.3.2 Detection of ground type

Different locations of interest may be marked on the ground below the robot. The robot will need to distinguish between different regions such as the nest, a source of food or an area of danger, depending on the current experimental setup. For this the robot should be able to tell apart different surfaces by some property. A simple property for this distinction is the colour of the ground. This can easily be modified by placing a piece of paper with a different colour onto the ground.

The robots should thus be able to distinguish the colours of different sheets of coloured paper on the ground.

2.3.3 Direct interaction with other robots

The robot must also be able to detect obstacles and other robots. For this the robot will use distance sensors placed on four sides. With these the robot can

sense the distance to obstacles in all directions. Some of these sensors may also allow the robots to communicate as they can also send out infrared signals other robots can detect. This allows for short-distance communication between robots.

2.4 Requirements

From the previously mentioned tasks a few requirements for the robots can be derived. In this analysis factors discovered during previous work with an other robotics platform in swarm experiments are also considered.

2.4.1 Movement

All experiments require the robots to move in the arena. This movement has to be precise and repeatable for good results. The robots must be able to move in different patterns to proceed around the arena.

Required motion patterns The robots must be able to move in a straight line. This is required when the robot has to move toward a target. Similarly the robots must be able to drive along curved lines. This allows for evasive manoeuvres and generally to turn around.

With these patterns the robots can already cover the entire arena. However some additional motions may be helpful in some experiments.

Additional motion patterns Some experiments may require the robots to turn in-place. Thus the robots should also be able to perform such turns.

If the robot ever enters into a corner and there is not enough room to manoeuvre it will need to be able to move backwards. The robots should be able to drive backwards.

Velocity control The robots should be able to control their velocity to enable movement at different speeds. It should be possible for a robot to move at a range of different velocities depending on the requirements of the experiment being performed. This should be controllable by software.

2.4.2 Ground sensing

The experiments are designed with the interaction between robots to be performed via stigmergy by changing the surface the robots drive on. These changes are either in the brightness of emitted or reflected light for the phosphorescent material and photo-switch respectively or in colour for the photo-switch only. As the experiments rely on this information, the robots must be able to detect these stimuli.

Emitted light The robots must be able to measure the emitted light from the phosphorescent surface. They need to be able to detect the presence or absence of light but also the intensity of light currently measured as this contains relevant information such as the age of the followed track.

Reflected light On the photo-switch the robots must be able to measure reflected light, as ambient light also varies with time and place. For measuring the same apparatus as for emitted light can probably be used. However the robots must also be able to emit some light to be reflected by the surface.

Colours As described above, some experiments require the marking of certain areas of interest. These areas can be marked by placing coloured pieces of paper on the surface the robots move on. The robots must be able to distinguish these areas from the usual surface as well as identifying the type of area they are currently in. Thus they need to have a method of detecting colour.

Line following As described in 2.3.1 the robots should have more than one sensor for the brightness of the ground. This will greatly aid with the following of lines.

2.4.3 Ground interaction

As previously mentioned the experiments involve the modification of the ground by either exciting the phosphorescent material or changing the colour of the photo-switch.

Exciting the phosphorescent material and the photo-switch Both the photo-switch and the phosphorescent material require UV-light to change their properties. The robots should thus be able to emit UV-light of the correct wavelength to interact with these substances.

Other materials and erasing Some other materials have been studied such as photo-switch with a different colour. In addition a method of erasing the photo-switch may also be interesting in future experiments. Thus the robots should have the capability to be expanded for such future experiments.

2.4.4 Environment sensing

In addition to the ground, the robots must also sense their environment. This is required both for interaction between robots and between a robot and its environment.

Sensing obstacles For some experiments there may be some obstacles for the robots to avoid while performing their tasks. Before the robots can avoid such obstacles they need to sense them. Thus the robots need to sense obstacles in front of them.

Additionally some obstacles may not be situated in front of the robot but to its side. The robots must also sense obstacles to the side and behind them.

Sensing other robots Experiments may require the detection of another robot's presence and adapting ones behaviour accordingly. The robots must be able to detect other robots, both active and passive.

Close quarters communication Localized communication between robots may be an element of some future experiments. Thus the robots need a way to communicate over short distances and to exchange small pieces of information with neighbouring robots.

2.4.5 General requirements

In addition to these more specialized requirements the robots also have some more generalized requirements.

Battery life The experiments may last a long time and thus the robots need to have a sufficient battery life to endure an entire experiment. Both the photoswitch and the phosphorescent surface decrease in signal intensity over about half an hour. The robots should have a battery life exceeding at least twice this time.

Displaying information The robots should also be able to display some information for debugging purposes as well as for the tracking system.

Extensibility As not all experiments the robots will be used for can already be designed, the robots should be adaptable to new requirements. The robots should allow for the addition of some new components such as a small camera. For these attachments a standardized protocol should be used to allow for maximum flexibility.

3 Component evaluation and selection

In this section the individual systems of the robot will be studied. The components, both electronic and mechanical, will be selected. This is selection is performed based on the requirements established in 2.4.

For each component a specification of its given task is first established. Then an exploration of different solutions for the problem is presented. From there the exact component is chosen.

3.1 Microcontroller

The most important component is the Micro controller unit (MCU). This will control all other components and will act as the CPU for the robot. Most available MCUs have integrated storage for programmes and some built-in components such as Analogue Digital Converter (ADC).

Requirements for MCU The microcontroller on the robots should be able to communicate with an external computer to be programmed. It should also be able to control the other components required by the robot. A dual-core controller is useful as it allows for some tasks to be performed even while the logic programme for the current experiment is running.

Choice of MCU For the robots the esp32s3 (ESP32-S3-MINI-1) was chosen for its high availability and low cost. It also provides additional features such as built in Universal serial bus (USB) to Universal asynchronous receiver-transmitter (UART) conversion for uploading programmes.

The chosen chip includes 8MB of flash memory for programme code and other persistent data. It also has two hardware Inter-integrated circuit (I^2C) controllers for communication with other components [26]. This allows for the separation of the attached components into groups for different tasks. For more connectivity it also includes a hardware Serial Peripheral Interface (SPI) controller [26].

In total the selected MCU has 36 General Purpose Input-Output (GPIO) pins. These pins can each be assigned a different set of functions. Care must be taken when choosing a pin for a function as not all pins support all functions. In particular the set of pins that can read analogue values is very limited [26].

Connectivity Next to the features mentioned above, the MCU also offers built-in capabilities for Wi-Fi and Bluetooth. They are required for the robots to send status information to a monitoring computer during the experiments. These data are later used to analyse the experiments.

3.2 Sensing

The robots need to be able to sense their environment. For this they require components to register different stimuli from the outside. In this section the required and preferred sensors to use will be discussed. Later the implementation of the chosen sensors on the robot will be detailed in section 4.1.

3.2.1 Movement and orientation

The robots may need to estimate their current orientation in three-dimensional space. This can be achieved by using an algorithm such as the one presented in [27]. These algorithms require the current acceleration and angular velocity of the tracked system as input and produce a quaternion representation of the orientation by using gravity to locate the downward facing direction. To allow for this functionality the robots should have an IMU to measure acceleration and angular velocity.

The IMU used on the robot is the LSM6DS3. This integrated circuit (IC) provides measurements for both linear acceleration as well as angular velocity in all three spatial dimensions. Additionally the wide availability and moderate cost of this chip allowed for easier implementation.

Connection Most algorithms for orientation estimation such as in [27, 28] perform best when a high sampling rate of the required inputs can be achieved. It is thus important to have a high bandwidth communication link between the IMU and the MCU.

3.2.2 Distance to objects

The robots must sense obstacles or other robots in their environment. For this sensors measuring distances to objects can be used. Different options present themselves for this type of sensor.

Ultrasonic sensors These sensors work by sending out a sound-wave and measuring the delay until its reflection returns. Due to the nature of waves only interacting with objects of their approximate size the hardware required both to emit and to receive sound waves is on the scale of about 20mm. Some newer and smaller designs exist [29], but they are not widely available and would increase the cost of the robot significantly.

These sensors usually have a larger range than other sensors in same size category [30]. This range comes at the cost of having a lower sample-rate as the sound waves travel much slower than the light used in infrared sensors for example.

Optical time of flight These sensors work in a manner similar to ultrasonic sensor by also measuring the delay between the emission and return of a reflection of a signal. However the signal now is a pulse of light. The higher speed of light compared to sound requires a much more precise timing for the sensor to be accurate. [31–33]

Active infrared sensors These sensors work by measuring the intensity of infrared light reflected by a surface close to the sensor. They have a smaller range compared to ultrasonic sensors. These sensors are much simpler than time of flight sensors, as there is no timing that must be kept aligned. As long as the emitter is active a signal can be measured at any time. Such sensors are much more susceptible to disturbances and to outside influence as the reflectivity of the material the light hits can change the value measured [34, 35]. A more absorbent surface may be seen as further away than a more reflective one. However models have been developed such as in [34] to correct for this sensitivity to the type of surface.

Chosen distance sensors The robots are equipped with active infrared sensors. The specific model used is the TCRT1050.

3.2.3 Brightness and colour

As laid out in 2.4.2, the robots must be able to sense the brightness of the ground below them. Additionally the colour of the ground should also be identifiable. As such some light sensing components are required.

Photo-resistors Photo-resistors are resistors where the resistance depends on the amount of light received on their surface. These components can be used to detect the presence or absence of light by measuring their resistance. This can be done by using a voltage-divider and measuring the voltage at the point between the resistors to ground.

Photodiode Photodiodes are components that produce a small current when hit by light [36]. This photocurrent is often small and may need amplification to be measurable by commonly available hardware.

Phototransistors Phototransistors are a combination of a photodiode and a transistor [36]. The small photocurrent generated by the photodiode is amplified by the transistor to a level much easier to measure. This allows for a much simpler

circuit to be used. A schematic of a phototransistor is shown in Figure 3.1. For more information on how the current can be measured see section 4.1.1.

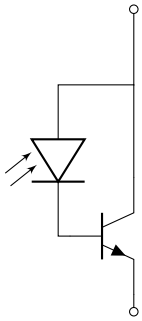


Figure 3.1: Simplified schematic of a phototransistor. A phototransistor can be conceived of as a photo-diode with an attached transistor to amplify its signal.

Colour sensors Many colour sensors are sets of phototransistors where each transistor is covered by a different colour-filter. Then the signal of each transistor can be measured and the colour is given as a set of values. These sensors are often available as an IC and can easily be integrated into the design. [37]

Choice of light sensors The robots are equipped with four phototransistors. Three of which are arranged as described in 2.3.1. The fourth phototransistor is facing forward to measure ambient light around the robot. All phototransistors are part of the SFH320 family. The front facing one has a different body as it measures at an angle to the surface it is soldered to.

Additionally the robots are fitted with a colours sensor located just behind the centre phototransistor. This sensor is a VEML6040.

3.2.4 Battery voltage measurement

The robots will need to know the current level of charge of their battery. This allows them to stop discharging it before damaging the battery. To allow this a circuit is needed. This circuit needs to lower the measurable voltage to an acceptable level for the input of an ADC.

Voltage divider This is a simple way to read voltages higher than the maximum voltage allowed for an ADC. Two resistors are used to divide the voltage drop across them in such a manner that even with the highest possible battery voltage the voltage placed on the input of the ADC is still within its bounds. This

voltage can be used in combination with a battery discharge curve to compute the Remaining State of Charge (RSOC) of the battery.

This is a simple solution to read the battery voltage. This setup requires the two resistors to be constantly connected between the terminals of the battery. If the battery is left for a long period of time, this can lead to deep-discharge as even if the current flowing through the resistors is small, it is not zero. Such a deep discharge is harmful to Li-ion batteries and can lead to battery damage [38].

Dedicated battery management chip Some ICs exist that can monitor the charge state of a Li-ion battery. These can measure the voltage and often even provide an estimate of the RSOC. They can also provide an interrupt signal to the MCU when the voltage drops below a threshold that can be set by software. Additionally they offer protection mechanisms for the battery, such a disconnecting the pins used for measuring when the voltage drops below a safe threshold. The robots are equipped with a battery management chip, the LC709203F.

3.3 Moving and Interacting

Having discussed the possibilities for the sensors and other measuring devices of the robots, the focus must now be shifted to the locomotion and other actuators. In this section an overview of possible techniques for locomotion, motors and other actuators will be established.

In a first step options for the movement of the robots will be explored. Once movement has been achieved the focus is shifted onto the interactions of the robots with the photo-sensitive surfaces.

3.3.1 Modes of locomotion

Different modes of locomotion were taken into consideration when designing the robot. In this section these modes are compared.

Vibrations Vibration driven locomotion is easy to implement with inexpensive hardware as only a non-aligned mass is required to generate the vibrations when attached to motor. However this form of locomotion is often unpredictable [19].

Robots driven by this form of locomotion may require extensive calibration and work to move in a predictable and repeatable manner. Additionally the functionality of such a robot may depend on the surface it is moving on.

Legs Legs present a nature-inspired form of locomotion. They often allow for a much greater range of terrain to be traversed [39]. They present a significant

challenge to implement in hardware as the mechanical complexity of such a system is often large [40].

Wheels Wheels are a simple way of transmitting the motion from the motor to the ground [41]. For low speeds and high friction one can assume rolling without sliding. This simplifies the equations governing the movement of the robot allowing for control logic with fewer parameters.

This led to the choice of wheels as the mode of locomotion for the robots. Their simplicity when combined with the right choice of motor leads to a straight forward design.

3.3.2 Motors

With wheels being chosen as the mode of locomotion, different motors can be considered for the task of making them turn. Each type of motor has some advantages and some disadvantages. The following presents an overview over different possible motors.

Brushed DC Motors These motors have a very simple construction. An inbuild alternator converts the applied Direct Current (DC) voltage to an Alternating Current (AC) voltage required by any motor to be able to move. This conversion is done by having two brushes that alternately rub along the positive and negative terminals [42].

However this construction is prone to wear as the brushes rub against some contact area. Additionally the control of such a motor is not very precise as only the current voltage applied can be controlled. This is not directly related to any measure of the motors output angular velocity.

Brushless DC Motor The name of brushless DC motor is a misnomer. Brushless DC motors are not DC motors. They require multiphase AC voltage to operate [42]. This is however most often generated by a motor control circuit in close proximity to the motor.

Many designs require three phases with a phase shift of $\frac{2\pi}{3}$ to operate. This allows for much finer control over the rotational frequency of the motor, as the motor turns once for each cycle of the applied voltage. So the angular velocity of the motor is the same as the angular frequency of the driving signal which can be precisely controlled.

Stepper Motor A stepper motor is an evolution of a brushless DC motor designed to be moved in discreet steps. Such a motor has a set number of steps

per rotation that can be advanced one at a time. This allows for precise control over the motor's position. If these steps are taken in regular intervals the rotation speed of such a motor can be controlled by the frequency with which steps are taken.

Produced torque The torque produced by a stepper motor at any one moment in time can vary by how many phases the motor in question has. It can be described by the following equation from [43]

$$T_A = \sum_{j=1}^{m} T_{Mj} - \eta\omega - T_F \tag{3.1}$$

where η is the viscous damping factor of the medium the motor is running in, ω is the current angular velocity of the motor, T_F is the torque due to internal friction and the T_{Mj} are the torques produced by each phase. The T_{Mj} can be computed by [43]

$$T_{Mj} = k_m \sin\left(n\phi(t) + \phi_{0j}\right) \cdot i_j(t) \tag{3.2}$$

where k_m is a constant depending on the construction of the motor, n is the number of steps taken in the current full rotation, $\phi(t)$ is the rotor position [1] , ϕ_{0j} is the phase-angle of the motor phase j and $i_j(t)$ is the current flowing through phase j.

As one can see from equation 3.2, if the motor is blocked and cannot advance and the number of steps taken increases with time there is a point where the direction of the produced torque will flip as the sign of $\sin\left(n\phi(t) + \phi_{0j}\right)$ changes. This can cause a motor to vibrate when blocked.

These advantages in the controllability of stepper motors led to the choice to use them as motors on the robot. However the torque produced by small motors may not be enough for the robots to be able to move freely.

3.3.3 Geared stepper motors

The torque produced by the motors can be amplified by the use of gears. This allows for a higher torque to be created at the cost of reduced angular momentum. This trade-off can be useful as electric motors often produce little torque but a high angular velocity.

One drawback of using gears is increased audible noise during operation as the motors mechanical parts are in motion against one another. This also leads to some losses due to friction. However these losses are often small and the increased output torque is still sufficient.

[1] This will not always correspond to the current step. The motor will only move when the torque produced is higher than the required amount for it to move, even if the number of steps taken continues to increase. This can lead the motor to miss steps.

Gearboxes Stepper motors can be attached to gearboxes containing the gears for easier implementation. A choice has to be made with regard to what reduction ratio to use. If a to high reduction ratio is used the robot will not be able to reach high speeds while with a too low ratio the robot may not be able to move as the resulting torque is not sufficient.

Increased wear Using gears will increase the wear on the mechanical components when compared to directly attaching the wheel to the motor. The friction between the gears will lead to significant wear over time. This will lead to a decreased accuracy in motion over time as well as less efficient driving. This wear must be accounted for in the design of the concerned parts

Chosen motors Considering all the above factors the motors chosen for the robot are geared stepper motors. The increase in cost for these parts is justified by the higher torque and reduced mechanical complexity outside the motors.

3.3.4 Motor drivers

The chosen motors are stepper motors and require two phases of current to work. These phases must be synchronised and offset in time for the motor to turn correctly. Figure 3.2 shows the required signals to drive a stepper motor forward by eight steps. This signal can either be generated by the MCU in software or a hardware stepper driver can be used.

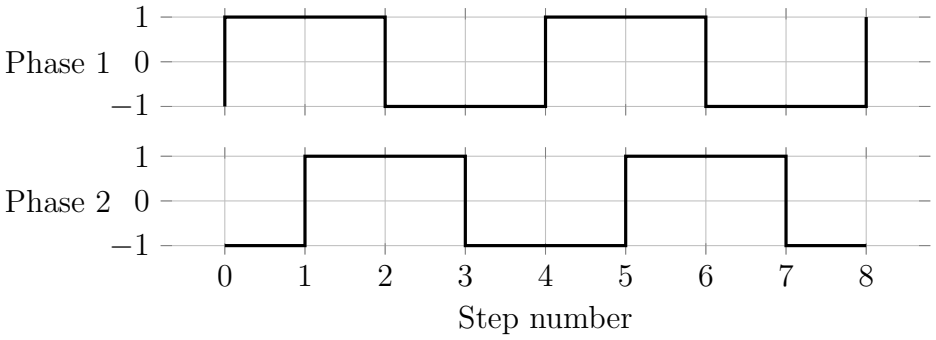


Figure 3.2: Signals required to drive a two-phase stepper motor. At each marked step the motor will advance by one of its steps. This signal can be generated by a MCU or by dedicated driver-IC.

Because of the limited power output of the MCU pins, a hardware driver is required on the robot as the motors draw more current than can be provided by such a pin.

H-bridge driver H-bridge drivers allow for a higher current and/or voltage to be controlled by a lower power circuit. Such a driver would allow the MCU to control the motor by connecting some pins to the supply voltage and others to ground. With this the signal shown in Figure 3.2 can be applied to the motor through a driver IC that handles the high current connections.

Stepper motor drivers Stepper motor drivers are motor drivers with added circuitry to generate the signals shown in Figure 3.2. This allows for easier control of the motor by the MCU as only two signals are required to drive the motor. One to set the direction of rotation and one to advance the motor by one step in the selected direction.

Additionally some stepper drivers feature methods for more precise control over the motors angle.

Micro-stepping As described in 3.3.2 a stepper-motor turns in discreet steps. Each motor has a set number of steps for a complete turn. Micro-stepping is a technique of exciting both phases in such a way as to simulate steps between the steps given by hardware. Depending on the motor this can lead to finer control and smoother motion when turning a stepper motor.

Micro-stepping can often be configured by changing the voltage applied to a pin of the driver IC. This allows this feature to be enabled and disabled by the MCU.

Chosen motor drivers As the motors chosen for the robot are geared-stepper motors, a stepper driver was chosen as the driver for the motors. Each motor is controlled by a single DRV8834 stepper-driver.

3.3.5 LED drivers

The robot requires different LEDs for different purposes. An UV-LED is needed to excite both the phosphorescent surface and the photo-switch. This component can draw a high current when in use because of the higher energy required for shorter wavelengths. Other LEDs are present to help sensors measure the reflected brightness and colour. All these LEDs require some control-circuits to be driven correctly. In this section different approaches to driving these LEDs are studied.

Direct attach The LEDs could be directly attached to a pin of the MCU. This is a very simple method of controlling the LED as the MCU can emit a Pulse Width Modulated (PWM) signal to control the brightness of the LED. However the MCU has a current limit it can output on any one pin. Thus the LED must

require less than this limit to reach the desired brightness. This method cannot be used for high-power components.

For Components with more than one colour or channel one pin is required for each channel of the component. This can lead to a high number of pins required.

NPN-transistor Another option for driving the LEDs would be an NPN-transistor. This component can amplify the current produced by the MCU [36]. However these components still produce a voltage drop when connected in series with the LED.

This solution only requires one additional component and can provide a higher current to the LED. It is still simple to implement as shown in Figure 3.3

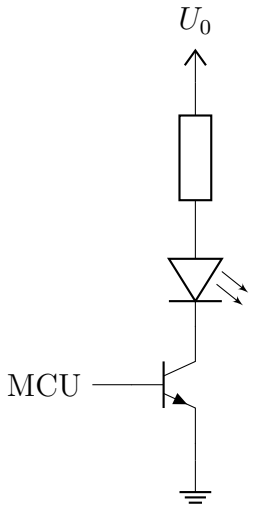


Figure 3.3: LED circuit with an NPN-transistor. The transistor amplifies the current coming from the MCU. This allows for higher powered LEDs to be driven. The transistor also introduces a voltage drop after the LED.

CMOS-transistor These transistors have a lower voltage drop when conducting [36]. This allows the control of LEDs whose forward voltage drop is close to the voltage of the power supply. The chosen UV-LED has a forward voltage drop of 3.2V which is close to the 3.3V of the supply.

The circuit needed for this type of control is similar to the one show in Figure 3.3 where the NPN-transistor has been replaced by a N-channel MOSFET.

Neopixel A communication protocol that allows RGB-LEDs with integrated logic to be controlled with a single data line. With this protocol many LEDs can be controlled with a single data-pin on the MCU. This allows for higher flexibility

and further expansion. However each LED must contain a small controller IC which can lead to higher costs per LED.

Chosen solution For the UV-LED the chosen solution was a CMOS-transistor as this allowed the high currents needed for this component. For more detail see 4.3. The socket for the LED that may be installed to erase the photo-switch in future experiments is also driven by a CMOS transistor, as this LED will also have a high power draw.

For the RGB-LEDs on the robot the Neopixel system was chosen. This allows to chain LEDs of the PCB and mount them on the chassis.

4 Implementation

Having chosen the components to be used on the robot, they now need to be implemented. For this each subsystem will be explored individually and designed to meet the required criteria. In a first step the sensors and their control circuits of the robot will be implemented. Then the system for the motion of the robot will be described. Finally the components required for the interaction with the photo-sensitive surface are detailed.

4.1 Sensors

Choosing the correct sensors is necessary to accomplish the tasks set out in the previous section. Here the choices made with regards to these components will be discussed. Additionally the configuration of the components on the robot will also be elaborated on.

4.1.1 Measuring analogue values

Many components are analogue devices and will thus produce currents or voltages corresponding to the input signal to be measured. Voltages can directly be measured by using a ADC. These components convert the applied voltage to a digital number with a given resolution. Most MCU provide such components as part of the built-in hardware. However for certain applications dedicated chips may be more appropriate. This is most often the case when higher resolution than the built-in components can deliver is required to measure the signal one wants to capture. Additionally dedicated components may offer lower noise-levels than the integrated ADCs.

Measuring currents Some components such as phototransistors do not produce a voltage as output [36]. Therefore they cannot directly be attached to the input of an ADC and yield usable results. One needs to convert the generated current into a measurable voltage. This can be done by using a resistor and the connections shown in Figure 4.1. In this circuit the current i_R provided by the phototransistor is passed through a resistor R_M. One can then measure the voltage across this resistor and calculate the current generated using the following equation.

$$i_R = \frac{u_R}{R_M} \tag{4.1}$$

With such a circuit it is thus possible to measure currents. Another effect of the resistor, is that one can use its value to determine the range of currents that can be measured. By choosing the value of the resistor such that the resulting voltages

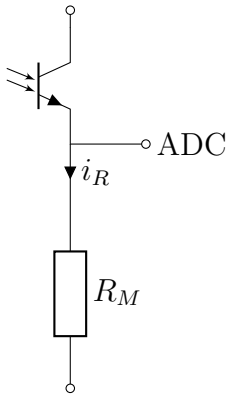


Figure 4.1: Circuit for converting the output current of a phototransistor to a measurable voltage for a ADC. The current flowing through the resistor will be measurable as a voltage on the ADC

closely match the accepted input range of the corresponding ADC on can tune this circuit for the given current to measure.

Power supply for analogue components Analogue components can only function properly if the supplied voltage is constant. A drop in supply voltage will also be seen as a drop in output signal. To prevent this the voltage supplied to these analogue components must be filtered to minimize such drops.

On the robot some capacitors are placed across the PCB to minimize such voltage drops. Along with two bigger 100µF electrolytic capacitors specifically for the motors. These should provide enough bulk capacitance to eliminate most drops in supply voltage.

4.1.2 Light sensors

The robots need to be able to measure the light emitted by or reflected off the surface they are located upon. This implies the need to have some sensors that can measure light facing downwards. Additionally an LED will be used to emit some light when the reflectivity is measured.

Phototransistors The robot's design incorporates four phototransistors. Three are used to sense light emitted or reflected by the ground below the robot. This is intended to be used to follow markings on the ground.

Each phototransistor emits a current i_{pt} depending on the intensity of luminous flux received [36]. As the ADC can only measure voltages, this current need to be

converted to a voltage as shown in 4.1.1. Then the measured voltage is given by

$$U_{measured} = R_M i_{pt} \tag{4.2}$$

Three phototransistors are arranged in the manner described in section 2.3.1. The remaining phototransistor is mounted facing forward to allow the robot to measure the brightness in-front of it.

Resistor value The phototransistors use the circuit described in 4.1.1 to measure the output current. The value of the resistor R_M in Figure 4.1 must be chosen such that the voltage present at the ADC is in its input range.

Let ϕ_{max} be the maximum luminous flux the phototransistors will receive when in use. For this the data-sheet of the phototransistors shows a typical current of $i_{max} = 650\mu A$ for a luminous flux of 1000lx. This brightness will never be reached when the robot is in use as the sensors point downwards and are surrounded by a mantle.

Using equation 4.2 the required resistor value can be determined by

$$\begin{aligned} R &= \frac{U_{max}}{i_{max}} \\ &= 5076\Omega \end{aligned} \tag{4.3}$$

Experimental value When using this value of resistor during a preliminary experiment to check its correctness the ADC could barely measure the light emitted by the phosphorescent surface. The sensitivity of the system had to be increased. The light emitted by the phosphorescent surface is usually located in the range of a maximum of $\phi_{max} \approx 10$lx.

These experiments determined a resistor of between $2M\Omega$ and $4M\Omega$ to be ideal for measuring the full range of the phosphorescent material. Thus a resistor of $3M\Omega$ has been chosen for the measurement circuit of the three phototransistors facing down.

Digitizing the signal The signal at the output of the circuit shown in Figure 4.1 is an analogue signal. This signal must be converted to a digital value for use in programmes running on the robots' MCUs. This is done by means of an external IC. The selected chip is the ADS1115. This chip was chosen as the ADC integrated into the MCU has a high noise level and insufficient sensitivity for the low light conditions.

4.1.3 Colour sensor

Each robot is equipped with a single colour sensor facing downward to detect the colour of the ground beneath it. This sensor is placed directly behind the centre phototransistor and close to an LED that can be used to emit light while measuring colour in the absence of ambient light. This is required as the ground below the robot can be darkened by the protective skirt around the robot's perimeter and experiments involving the phosphorescent surface have to be performed in the dark.

Additionally the colour sensor can be used to measure the brightness more reliably than with the phototransistors. The chosen VELM6040 provides a conversion from sensor counts to lux in [44].

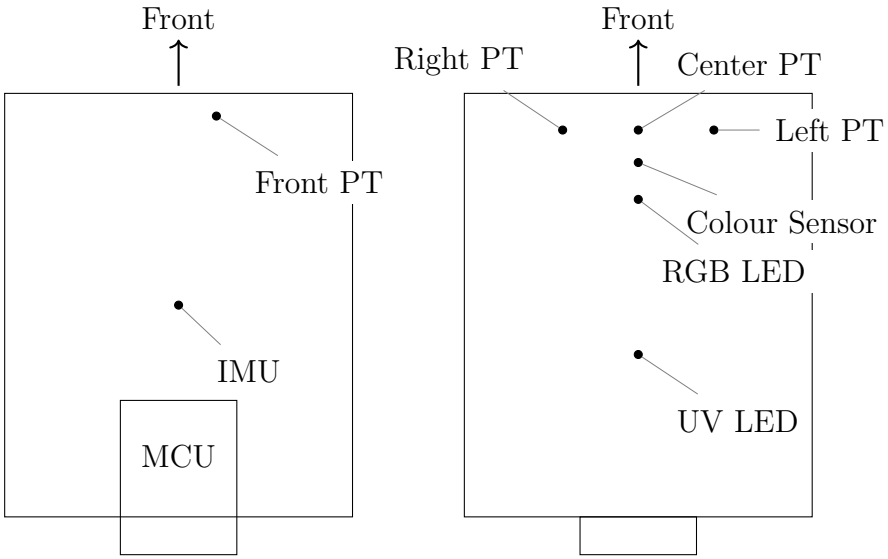
Exciting the phosphor layer When using the LED to measure the colour great care must be given not to also excite the phosphorescent surface. This can be done by not using any blue light from the LED. This will however distort the colour measured by the colour sensor and this must be accounted for in software.

4.1.4 Inertial Measurement Unit

As described above, the robots should have an IMU to measure their movement, both self-initiated and induced by others. The IMU is located in the centre of the robot on both the sagittal and axial axis. Even if this is not exactly in the centre of mass of the robot this placement should nonetheless allow for accurate and useful measurements. Figure 4.2 show the location of the IMU and other sensors that are mounted directly to the PCB.

Connection with MCU As mentioned in 3.2.1 the algorithms used for estimating the orientation of the robot require a high sampling rate of the IMU. This in turn requires a high bandwidth link between the IMU and the MCU. This link is achieved by having a dedicated I^2C bus for the IMU. This bus can run at a higher clock rate than the bus used by the other components. During testing a sampling rate of about 1kHz could be achieved.

Correcting movement As stated in 3.2.1 the IMU can be used to obtain the robot's current orientation relative to its original state. This information can be used to correct the movement of the robot in real time. The correction of the movement can be achieved by using a Proportional Integral Differential (PID) controller with a set orientation. When using such a controller, the orientation must be kept in quaternion form as to prevent artefacts when interpolating angles.



(a) Top view (b) Bottom view

Figure 4.2: Locations of the components on the PCB. The three phototransistors on the bottom side allow for easier path following. The IMU is located at the centre of the robot. The UV-LED is placed at the back to allow the robot to retrace a followed line without disturbing its measured intensity.

Other applications The IMU can also be used for other applications such as detecting collisions. However a particularly useful case is the detecting when the robot is turned over to switch of the UV-LED. This mechanism can prevent the LED from shining into the eyes of the operator.

4.1.5 Infrared Sensors

Each robot also has four infrared sensors. One facing in each direction: front, left, right and back. These infrared sensors are phototransistors with a filter to only sense infrared light.

Placement The infrared sensors are attached to the PCB with cables. This allowed for greater freedom in placing them inside the chassis. In the chassis they are positioned such that one faces forward, one faces to each side and one faces backwards. The sideways facing sensors are mounted at an angle to allow for some forward detection. The exact placement of the IR sensors in the chassis is shown

in Figure 4.3. This shows a cross section at a height of 45mm above the ground.

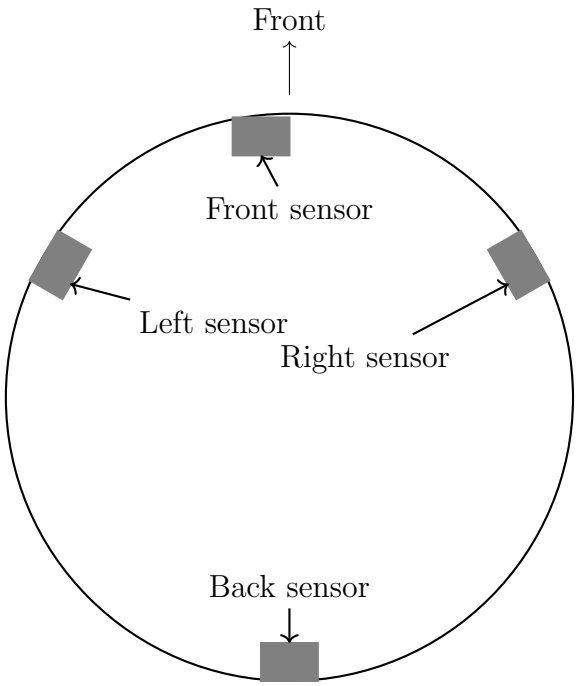


Figure 4.3: Placement and orientation of the IR-sensors on the robot as seen from above. This shows a cross section at 45mm above the ground. All sensors are facing outwards perpendicular to their sides.

Measurements As the sensors are phototransistors the circuit shown in Figure 4.1 is used to convert the output current of the IR sensors to a measurable voltage. A resistor value of $R_M = 200\text{k}\Omega$ is used. The voltages produced by this circuit are measured using a separate ADC the ADS1115. This chip is mounted centrally on the PCB.

Operation modes As the sensors are also equipped with an LED to emit the correct wavelength for the sensor they can operate in two modes one active and one passive. In the active mode the LED is illuminated and the reflected light can be measured. In the passive mode only light from the environment or other robots can be seen.

The active mode can, for example, be used when trying to detect obstacles as the amount of reflected light will increase when the distance decreases. The passive mode may be useful when trying to detect other, actively measuring, robots close-by. In this case the light emitted by other robots is detected.

To allow for active measuring a pin on the MCU can control the LEDs of the sensors. This pin controls all LEDs centrally.

Other uses The LEDs could also be used to transmit some information to other, close-by, robots. This could simulate a local mode of communication. The data-rate and reliability of this mode of communication is still to be examined.

4.1.6 Battery monitor

The robots use a dedicated IC to monitor their batteries' voltages. The LC709203F[2] chip was chosen for its high flexibility at relatively low cost. In addition to the current battery-voltage it can provide an estimate of the remaining charge as a percentage.

Low-battery warning Some interrupts can be programmed for low battery level based both on RSOC and the battery-voltage. These interrupts are generated on GPIO pin 10 of the MCU. The software library provides two methods of connecting handlers to such a low battery warning. One where a voltage can be set as the trigger and one where the RSOC is set.

These interrupts allow for the current programme to gracefully save data if this is required and to warn the operator of the low battery status. If the experiment requires or allows it this could also initiate the robot to seek recharging if such a mechanism is provided.

4.2 Motion

In this section the focus is placed on the implementation of the movement of the robot.

4.2.1 Wheels

As the robot will perform experiments indoors on a level even surface no capability to navigate through rough terrain is required. Also as the torque required from the motors is small (see 4.2.4).

This in combination with the simple and well understood design of wheels leads to the choice of wheels as a means of locomotion for the robot.

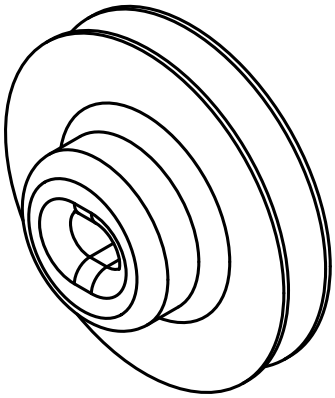


Figure 4.4: Perspective view of a wheel used on the robot. The rubber O-ring fits into the groove along the outer edge. The inner part surrounding the oval slit for mounting to the gearbox can be printed in TPU for better adhesion.

4.2.2 Wheel design

The wheels are designed to have rubber o-rings as tyres. This increases the friction between the wheel an the ground and thus allows for more controlled motion of the robot. The o-rings are fitted into a groove wrapping around the outer edge of the wheel.

The wheels are attached to the motor by using the shaft of the gearbox from the motor. This shaft has an oval shape that interfaces with a slot in the wheels hub. For a secure mounting the volume surrounding the slot is printed in Thermoplastic Polyurethane (TPU) a flexible plastic. This allows for the dimensions of the slot to be set tighter and yields a better connection between the wheel and the shaft. If no TPU is available, this part can also be printed in polylactide (PLA) but one may need to adjust the dimensions.

Wear With gears comes increased wear on the mechanical components due to the contact point being in motion. However this wear is limited to the gears inside the gearbox in the assembly shown in Figure 4.5. In case of a failure either the gearbox or the gearbox and the motor can be replaced.

[2] At the time of writing this IC is no longer being manufactured and has been replaced by the LC709204FXE.

4.2.3 Wheel and motor assembly

The motors have two mounting holes for M2 screws. These are used to mount the motors directly to the bottom chassis of the robot. A drawing of this assembly can be seen in Figure 4.5. The wheels are mounted to the motors' axles directly and are held by friction. This friction can be increased by printing the inner part of the wheel in TPU. O-rings are fitted around the wheel before mounting it.

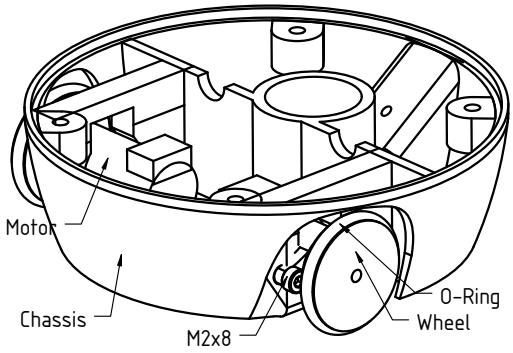


Figure 4.5: Assembly of wheel and motor for the left hand side of the robot. This assembly is mirrored on the other side. The motor is mounted to the bottom part of the chassis.

4.2.4 Required motor torque

For the robot to move the motors must produce a torque high enough to overcome the friction from the ball support and the inertia of the robot itself.

Gear ratio The selected motors are geared stepper motors. The attached gearbox has a gear ration η_g of about $1:19$. Thus the torque at the output is about 19 times the torque produced by the motor. The following equation gives the torque at the wheel of the robot as a function of the torque produced by the motor.

$$M_{wheel} = \eta_g M_{motor} \tag{4.4}$$

Conversely the angular velocity of the wheel ω_{wheel} is given by

$$\omega_{wheel} = \frac{1}{\eta_g}\omega_{motor} \tag{4.5}$$

where ω_{motor} is the angular velocity of the motor.

Rolling without slipping An assumption for these calculations is that the robots' wheels roll without slipping. This is assumed to be the case as the coefficient of static friction between the rubber on the wheel and the acrylic surface is high. If the wheels are observed slipping, the acceleration can be reduced to mitigate this.

Friction with the ground The robot has three points of contact with the ground. Two wheels and a steel ball behind. Both wheels are driven by a motor and thus the friction produced by them can be ignored if the above assumption of rolling without slipping holds. Friction still occurs between the ball and the ground or between the ball and its enclosure or both. The coefficients of dynamic friction between steel and the PLA used for printing the chassis is $\mu_{PLA} \approx 0.3$ [45]. The coefficient of friction between dry steel and acrylic if $\mu_{Acrylic} \approx 0.15$ [46]. To perform this calculation the worst case is assumed and thus a coefficient of friction of $\mu = \mu_{PLA} \approx 0.3$ is used.

The force of friction between the steel ball and its holder is given by

$$F_f = \mu N \tag{4.6}$$

where N is the normal force on the contact point. This is equal to the force due to gravity when the robot is in contact with a surface. One can assume this to be third of the total weight of the robot. Thus

$$F_f = \mu \frac{1}{3} m_{robot} g \tag{4.7}$$

where m_{robot} is the mass of the robot and g is the acceleration due to gravity.

Acceleration The robots should be able to accelerate to their target velocity of $30\text{mm}\,\text{s}^{-1}$ in one second. Thus an acceleration of $a = 30\text{mm}\,\text{s}^{-2}$ is required. The force F required to produce such an acceleration is given by

$$F = m_{robot} a + F_f \tag{4.8}$$

where m_{robot} is the mass of the robot and F_f is the opposing force due to friction of the ball on the surface.

Using the assumption that the wheels roll without slipping, the torque required to produce the force F is given by

$$M_{wheel} = F\, R_{wheel} \tag{4.9}$$

where R_{wheel} is the radius of the wheel.

Torque at motor By combining equations 4.4 to 4.9 the torque the motor must produce can be expressed as

$$M_{motor} = \eta_g \, R_{wheel} \left(m_{robot} a + \mu \frac{1}{3} m_{robot} g \right) \tag{4.10}$$

Using a mass $m_{robot} = 100\text{g}$, $g = 9.81\text{ms}^{-2}$, $\mu = 0.3$, $N_{motor} = 9$, $\eta_g = \frac{1}{19}$ a torque of $M_{motor} = 5.3 \times 10^{-5}\text{Nm}$ must be produced by the motor. This is well within the capabilities of the chosen motor as shown by the motion of the robot is section 5.1.2.

4.2.5 Stepper motor and driver

As mentioned in 3.3.2 stepper motors allow for controlling the angle a motor is turning precisely. This in turn allows the robot to precisely control its motion as each motor can be commanded to turn a certain number of steps to achieve the desired overall motion. This leads to a high amount of control over the robot's driving.

However as mentioned in 3.3.4 stepper motors require a specific signal to be sent to each phase for them to turn properly. This can be achieved by using a stepper driver. This IC can control the signals sent to each phase of the motor. Additionally the current the motor can pull is also high and such a driver can provide such a high current.

Drivers The chosen stepper motor drivers (DRV8834) allow for different modes of operation. The chosen mode is STEP/DIR. This allows for one pin to set the direction the motor should turn and another to advance the motor by one step on each rising edge [47]. Additionally the number of micro-steps per full-step can also be configured.

Micro-stepping On the robot a circuit allows for the configuration of two different micro-stepping settings during operation. The circuit shown in Figure 4.6 toggles the states of the pins M0 and M1 on each motor driver. Pin M1 is toggled between 3.3V and ground while Pin M0 is toggled between a high-impedance state and ground. This allows to select either 32 micro-steps per steps or full steps respectively [47]. This choice can be made in software depending on the requirements of the current experiment.

Holding position Stepper motors also pull current continuously when holding a position which can lead to a high energy consumption and overheating. This can be mitigated by disabling the driver and thus cutting the supply of the motor.

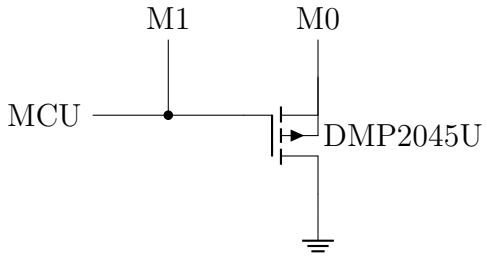


Figure 4.6: Circuit to change micro-stepping mode. This toggles pin M1 between Low and High and pin M0 between Low and High-impedance.

Doing this is only possible because the amount of holding torque required for the robot to be immobile is very small.

In the default implementation of the motor control logic, the software checks whether a motor should turn or not and subsequently disables the motor if no rotation is commanded.

4.3 Interacting with the ground

The robots require a UV-LED to interact with the photo-sensitive surfaces. This UV-LED requires a circuit for it to function.

Driving the UV-LED The UV-LED required for the interactions can pull a high current of up to 1000mA continuously. Such a high current can lead to overheating of components if not chosen carefully. This overheating will cause the brightness of the UV-LED to be unpredictable.

Component selection For optimal results, the UV-LED should have a wavelength of 395nm. This wavelength yields good results both on the phosphorescent material and the photo-switch. The selected LED that produces this wavelength can support a continuous current of up to 1000mA. However at this current the intensity of light emitted is so high both materials saturate an cannot accept more photons.

To preserve a higher degree of control the maximum current through the LED has been chosen as $i_{UV} = 330\text{mA}$. With this current the LED still has enough power to excite the phosphor to saturation. However a smaller range of brightnesses with the same number of control-steps leads to a more precise control of the brightness in experiments.

The chosen LED has a forward voltage drop of 3.2V. Thus the voltage drop across the resistor is $U_R = U_0 - U_{LED} = 3.3 - 3.2 = 0.1\text{V}$. From this the resistance

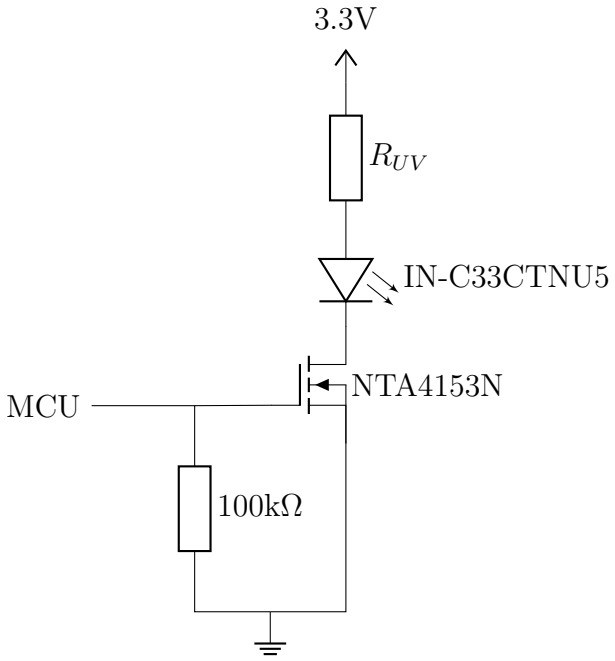


Figure 4.7: Driver circuit for the UV-LED with current limiting resistor R_{UV} and pull-down resistor to prevent flashing when MCU-pin is floating during boot

R_{UV} is given by

$$R_{UV} = \frac{U_R}{i_{UV}} \approx 300\text{m}\Omega \tag{4.11}$$

Thus a resistor of 300mΩ was chosen for R_{UV}.

Pull-down resistor As shown in Figure 4.7 a pull-down resistor is integrated into the circuit. This resistor prevents any accidental illumination of the UV-LED when the MCU-pin is in an undefined state. This can happen during the boot process of the MCU and the resulting flashes of UV-light can destroy the experiment if the robot is booted on the UV-reactive surface or blind people when the LED is facing up.

Keeping the light focused No UV-LED at the required wavelength has an angle of dispersion tight enough. Thus the light has to be kept focused on a small section of ground by some other means. This is accomplished by shining the LED into a tube integrated into the chassis. To minimize losses of light, this tube is lined with aluminium that can reflect the light back into the tunnel. With this

the LED can achieve a sufficient brightness to modify both surfaces the robot will be used on.

4.4 Battery charging and power supply

The robots are powered by a battery. This battery has a source voltage of $3.7\text{V} \leq U_{bat} \leq 4.2\text{V}$ depending on the current charge level. This voltage must be stepped down to a usable level for the MCU and other components.

4.4.1 Battery

The robots need to be able to charge their battery. For this dedicated IC the MCP73831 is used to provide an adequate charge current and voltage depending on the charging phase of the battery. Additionally a circuit also switches the source for the robots' systems to the connected charger when plugged in. The circuit used for switching the source while charging is shown in Figure 4.8.

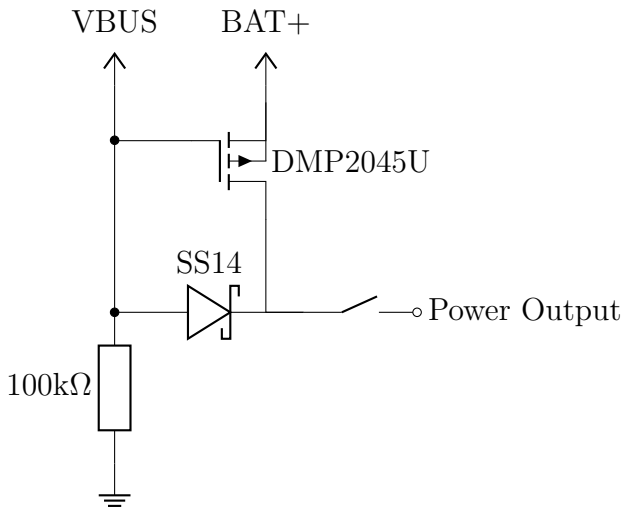


Figure 4.8: Circuit for switching between power source when charger is connected. When external power is connected to VBUS, the transistor switches and the battery is no longer connected to the power output. This also handles the switching on or off of the robot.

4.4.2 Power conversion

The MCU and most ICs on the robot require a stable supply voltage of 3.3V to function. For the best results this voltage should drop as little as possible even when a high current is drawn.

Stepping down the voltage As both the battery voltage and the voltage provided by USB are above the required 3.3V these voltages need to be stepped down. An IC is used to convert from the higher voltage. The AP2112K is a commonly used chip for this conversion. It accepts an input voltage of up to 6.0V and outputs a voltage of 3.3V at a current of up to 600mA [48].

As total consumption of over 600mA for the entire robot is to be expected if both motors are holding position and the UV-LED is at full power, two such ICs are built into the design. This allows for a total current of 1.2A to be drawn.

Stabilizing the supply Many components, especially the MCU, require a stable voltage to function properly. If the voltage drops below a threshold for even a few microseconds the MCU will shut off. Two main measures are taken to prevent the drop of the supply voltage. First a large enough power budget has been built into the supply. This makes sure the voltage will not drop for long periods of time at once.

However short drops may still occur occasionally, especially when switching on motors or other components. To prevent these spikes to drop the voltage below an acceptable level, some filtering capacitors are placed near critical components as well as throughout the PCB. These capacitors will deliver the high current required when switching on certain components.

The motors require especially high currents when switched on. To buffer this inrush-current some extra 100µF electrolytic capacitors have been placed close to the motor drivers as recommended in their data-sheet [47].

4.5 Internal communication

Most ICs on the PCB require some communication with the MCU to function. Others such as the infrared sensors and the phototransistors are analogue components and thus require an ADC to be read. The chosen MCU has an integrated ADC but for higher resolution and because of a limitation on the number of pins used two external ADCs are used to read these sensors.

ADCs and battery monitor For the ADCs used to read the values of the analogue sensors and the battery monitor chip I^2C is used to communicate. This allows for many components to be connected using a single bus. The relatively low bus speed of 400kHz is not a problem as the volume of data is not large. The only component requiring a large amount of data to be sent over this shared bus is the display. However this is only required when frequently refreshing its contents.

IMU The IMU supports a higher frequency for communication than the other components and its sample-rate is high enough to saturate the lower-speed bus. Thus it is attached to a separate I^2C bus running at 800kHz. This is possible as the MCU has two hardware I^2C units built in.

This connection to a separate bus also allows the IMU to be read by a different thread running on a different core. Using the same bus across multiple cores, the programme must ensure only one core uses the hardware at a time. Failing to do so leads to a crash and restart of the MCU. This complication is avoided by having a separate bus for the IMU.

Other elements Some ICs such as the battery monitor have the ability to send interrupts to the MCU when certain conditions are met. These are connected directly into the MCU to alert the running programme of any detected fault or unusual state.

4.6 External communication

The robots should be able to communicate with external hardware. This can be a computer recording the experiment receiving logging data from the robot or another robot if the experiment allows it.

4.6.1 Wired communication

The robots have the ability to communicate with an external computer over USB. This is most often used for uploading the firmware to the robot. It can also be used to send debugging information to the connected computer.

Protocol The robots use the UART protocol at a baud-rate of 115200 baud to communicate over USB. The MCU has integrated hardware for this protocol. The use of this hardware must be signalled in the uploaded programme [3].

4.6.2 Wireless communication

When the robots are in motion the use of wired communication becomes impractical. Thus the robots also have the ability to communicate wirelessly.

[3] The correct compiler options must be provided for the programme to send the debug output to the integrated UART to USB converter. If the default library is used these options are already set.

Wi-Fi The robots can communicate over Wi-Fi. They can both connect to a Wi-Fi-network or simulate their own. Once connected they can send and receive data over UDP or TCP. Wi-Fi is the preferred mode for wireless communication when robots may send data to one another.

The chosen MCU has a Wi-Fi-module built in allowing it to communicate over Wi-Fi.

Bluetooth Another possibility is the use of Bluetooth to send data to and from the robots. The MCU has the capabilities to use Bluetooth but this was not tested during the development of the robots. Bluetooth also has a reduced data-rate compared to Wi-Fi when using the MCU as only a low-power mode is available.

However some use-cases exist for Bluetooth such as remote-controlling the robot from another device for demonstration purposes. This is better achieved via Bluetooth as the latency is lower than Wi-Fi.

4.7 Possible expansions

As the robots are intended to be used in a variety of experiments they need to be flexible in the capabilities they provide. It may be needed for a future experiment to add some specialized equipment to one or multiple robots.

4.7.1 Planned expansion

A future expansion is planned where the robots will be able to erase traces left by other by shining a different wavelength of light onto the trace. To accommodate such an expansion a place for another LED is already present on the robots.

4.7.2 Standard interfaces

For further not yet planned expansions, the robots are equipped with two standard interfaces for submodules.

I^2C The main I^2C bus is exposed with a connector on the top side of the PCB. Submodules can be connected to this connector to extend the capabilities of the robot beyond what is already present. However one must keep in mind the addresses already in use on this bus when designing a submodule as to not have any collisions. For more details see Table A.2.

SPI Another connection is provided via SPI. This allows for a higher-bandwidth dedicated bus of communication directly with the MCU.

5 Evaluation

Having designed the robots, in this coming section some tests are run to determine if everything is working as expected. For a list of the required parts, please refer to A.1. In a first step the individual systems are tested for functionality. Then the entire robot will be subjected to tests to evaluate all systems when interacting.

5.1 Individual Systems

Some systems were first tested individually before the entire robot was tested as a whole.

5.1.1 Battery life

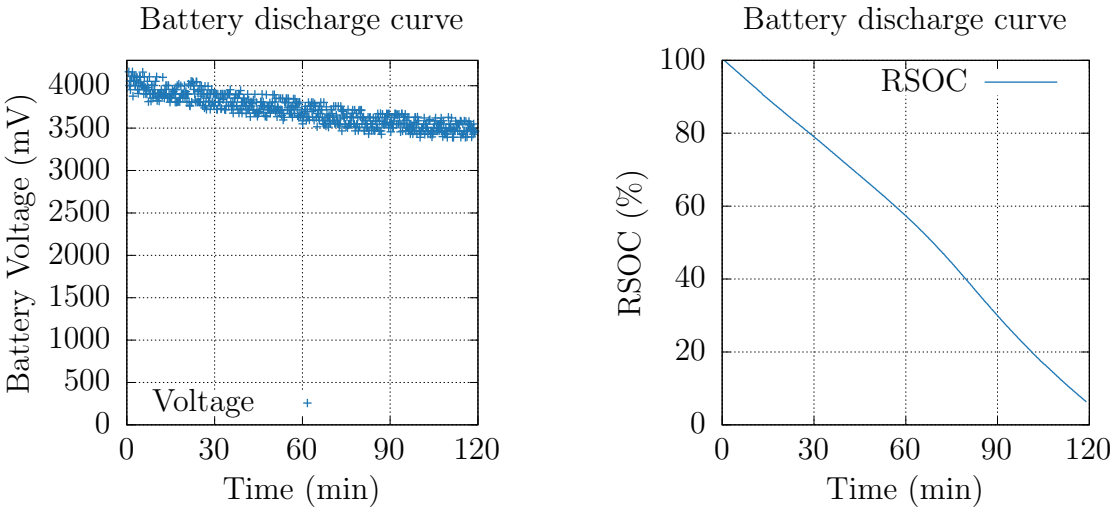
The battery life of the robot was tested by having it perform a medium workload and periodically measuring the battery voltage and the RSOC. For the results shown in Figure 5.1 the robot's motors where running at a normal driving speed of 0.3rads^{-1}. Additionally the UV-LED was set to half the maximum output power, a setting high enough to saturate the phosphorescent surface. No Wi-Fi or Bluetooth connection was active during this test.

Battery life From the measurements shown in Figure 5.1 the robots have a battery life of about 2 hours under a medium workload with a capacity of 1000mAh. This endurance is enough to perform multiple experiments without recharging.

Battery longevity As the currently existing Li-ion batteries show degradation over time [49] this battery life will degrade with the age of the battery. However if this degradation reaches a point where experiments can no longer be performed, the battery can be replaced. Thus this degradation is not problematic for the longevity of the robot itself.

5.1.2 Straight-line Movement

To evaluate the performance of the robot regarding its ability to move on the experiment's surface a series of small trials were conducted where the robot was instructed to move in a straight line at a given velocity. The movement of the robot was recorded using the tracking system for the experiments [25]. From the position data obtained the velocity was computed to verify the robot's ability to drive with a set velocity. Additionally the position data were also directly used to analyse how straight the recorded motion was.



(a) Measured battery voltage

(b) Battery RSOC computed by battery monitor IC

Figure 5.1: Battery discharge curve recorded with running motors, UV-LED at half power and no Wi-Fi enabled. The battery had a nominal capacity of 1000mAh. (a) shows the remaining voltage of the battery. (b) shows the RSOC computed by the battery management IC. With these settings a runtime of two hours seems realistic

Uncalibrated movements To test the uncalibrated movements of the robots, a robot has been chosen and programmed to drive a two different velocities. First the robot was programmed to move at a linear velocity of $30\,\mathrm{mm\,s^{-1}}$ and then later at $60\,\mathrm{mm\,s^{-1}}$. The resulting trajectories can be seen in Figure 5.2.

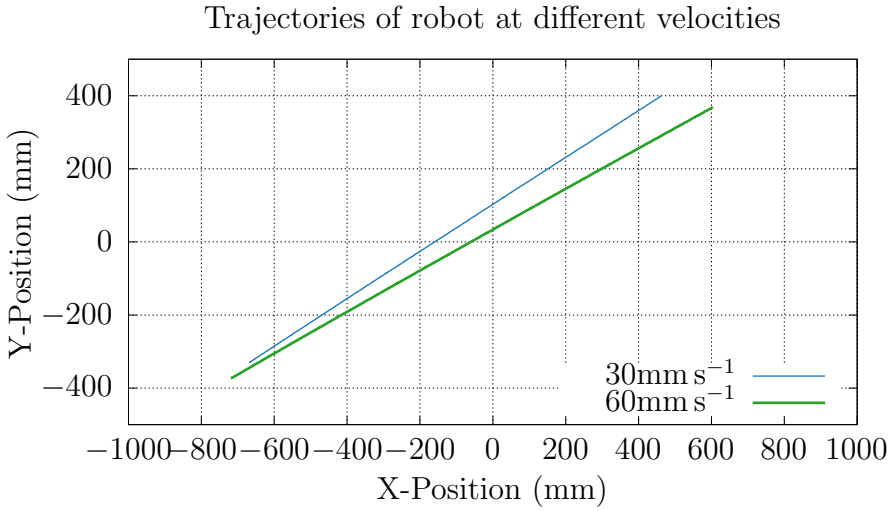


Figure 5.2: Paths followed by the tested robot when instructed to move in a straight line. The robot was tasked with moving in a straight line at two different speeds. Once at $30\,\mathrm{mm\,s^{-1}}$ and once at $60\,\mathrm{mm\,s^{-1}}$.

As one can see the paths followed by the tested robot do not appear to curve. The robots should thus be able to move in straight lines without needing to correct the motion using the IMU.

Velocity During the runs from above, the velocity of the robot can also be calculated. This was done by computing the numerical derivative of the positions over time. Then the norm of the resulting vector was used as the linear velocity of the robot. In Figure 5.3 the linear velocity of a robot commanded to move at $30\,\mathrm{mm\,s^{-1}}$ and at $60\,\mathrm{mm\,s^{-1}}$ can be seen.

The robot can reach the target velocity in both cases. When programmed to drive at $30\,\mathrm{mm\,s^{-1}}$ the robot drove an average of $29.4\,\mathrm{mm\,s^{-1}}$. At a commanded speed of $60\,\mathrm{mm\,s^{-1}}$ the robot reached $58.8\,\mathrm{mm\,s^{-1}}$ on average. For both speeds the robot can maintain the velocity with only some slight disturbances.

5.1.3 Manoeuvrability

After the robots were tested while moving in a straight line the next step was to evaluate their range of motion. The robots should be able to move along curved

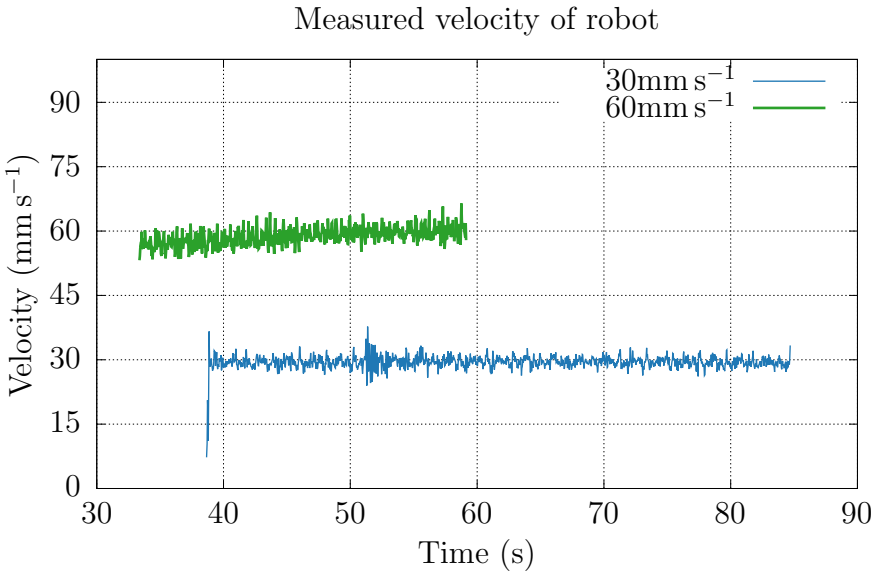


Figure 5.3: Measured velocities of robot when commanded to move in a straight line at 30mm s^{-1} and 60mm s^{-1}. The robot was started after the beginning of the recording. It then accelerates to the target velocity and moves until the operator stops it or it collides with a wall.

lines as well as straight ahead. To test this in a first step a robot was programmed to move in a circle.

Circle The tested robot was programmed to move in a circle with a diameter of 300mm. The recorded trajectory of this robot is shown in Figure 5.4. The robot was programmed to drive in a circle by setting the ratio between the velocities of the left and right wheel. The required velocities can be computed using the following equations:

$$v_r = \frac{2v_m}{1 + \frac{R-R_\perp}{R+R_\perp}} \tag{5.1}$$

$$v_l = v_r \frac{R - R_\perp}{R + R_\perp} \tag{5.2}$$

Where v_r is the velocity of the right wheel, v_l is the velocity of the left wheel, v_m is the velocity of the centre of the robot, R is the radius of the circle to drive and $R_\perp$ is half the distance between the wheels. These equations give the linear

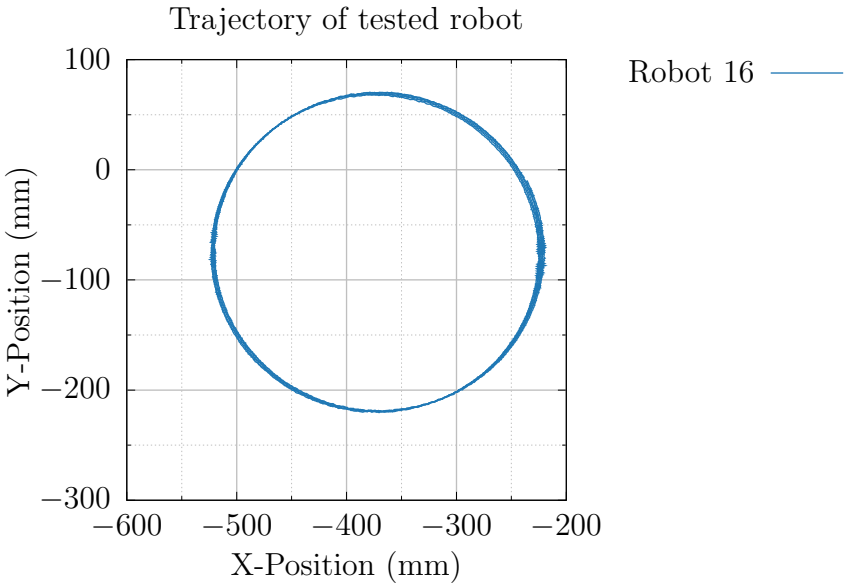


Figure 5.4: Trajectory of tested robot when programmed to move in a circle with a diameter of 300mm. A total of about 10 circles were completed to record the shown trajectory.

velocities for the wheels for a counter-clockwise circle. To drive the circle in the other direction one needs to swap v_r and v_l.

Square The circle the robot had driven until this point only required a constant ratio of velocities to be held by the motors. For a the robot to be able to draw a square, it must control the distance it drives and then be able to turn. These tests were carried out without the use of the IMU. They were performed to purely evaluate the performance of the drive-system. Figure 5.5 shows the trajectory of a robot driving a square and turning on the spot at each corner. The observed trajectory reveals a drift as the robot undershoots the turns in the corners. This can stem from an inaccurate measurement of the distance between the wheels as this is needed to compute the time required for the turn.

During experiments where the robots use their UV-LEDs to draw on the surface the robots cannot turn on the spot as the UV-LEDs is not located on the resulting rotational axis. The trail left would thus be a small arc. To eliminate this the robots can move in a square with rounded corners. Such a motion can be seen in Figure 5.6. This also shows a drift but the parameter defining the distance between the wheels was tweaked and the drift is now in the opposite direction as the robot

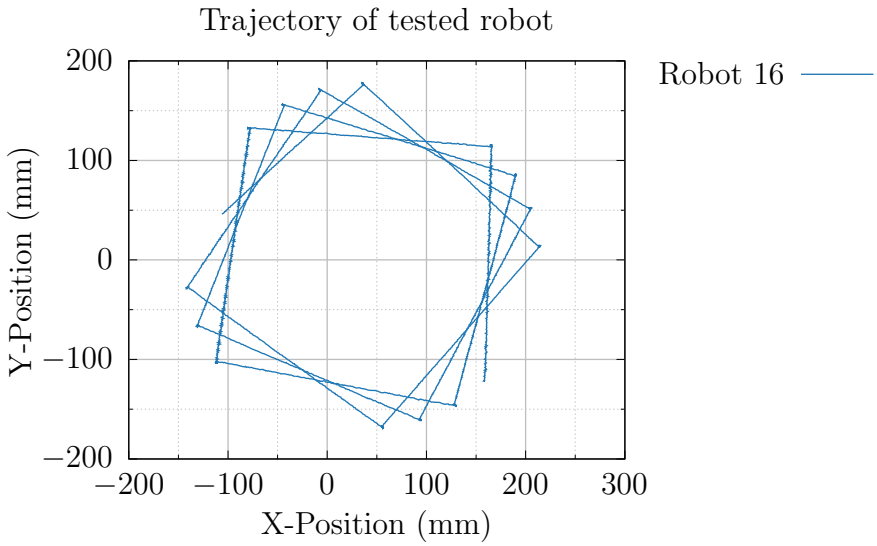


Figure 5.5: Trajectory of tested robot when programmed to move in a square with edges of length 250mm. A total of four and a half squares were driven for the shown trajectory. A drift can be observed as the robot always undershoots the angle at each corner.

overshoots the angle in each corner. This hints at the possibility for a calibration to eliminate this drift. However this would have to be done on a per-robot basis and would thus be a time-consuming endeavour.

The overall manoeuvrability of the robot was satisfactory as all the desired types of movement and manoeuvres could be achieved. However a slight drift can be observed when the robot is programmed to drive even simple shapes. This may be corrected by using the IMU to guide the robot.

5.2 Combined tests

After having tested the individual systems their interplay now needs to be verified as well. For this a few simple scenarios were designed for the tested robot to complete. The first such scenario is following a line.

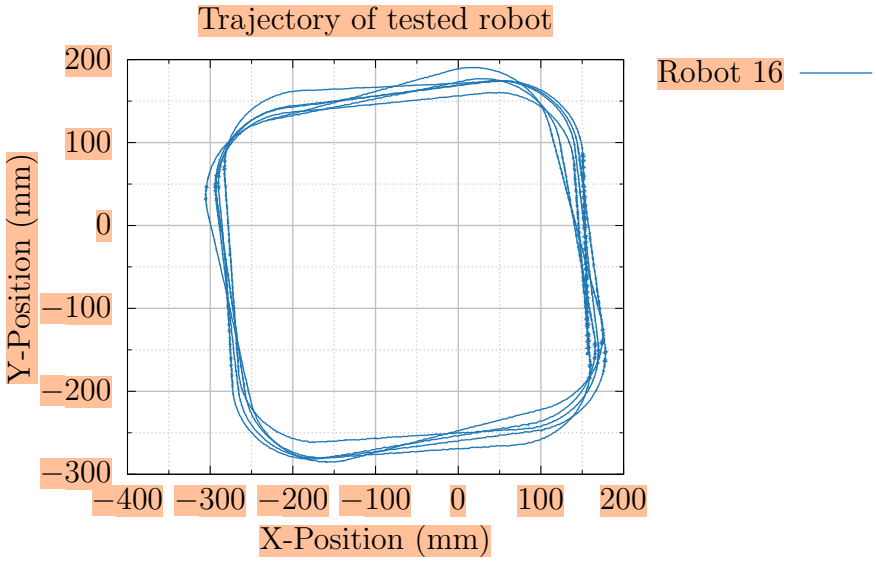


Figure 5.6: Trajectory of tested robot when programmed to move in a square with rounded corners. The length between the arcs was programmed at 250mm. A total of about 10 squares were driven during the recorded data.

5.2.1 Line following

To test whether the robot can follow a line using the phototransistors at the front the following scenario was implemented. In a first step the robot draws a line onto the phosphorescent surface. It then stops and is picked up by the operator and placed somewhere along the just drawn line. From there it will attempt to follow the line for as long as possible.

Used algorithm The algorithm used to follow a line is very simple. If the robot is on a line, the centre sensor should measure the most signal. The left and right sensor should both measure the same amount of signal, if the robot is centred on the line. As the absolute value of the signal will decay over time the algorithm uses the differences between signal values to adjust the robots motion. The adjustments made to the robots motion can be seen in Algorithm 1. The parameter v_t is the targeted velocity of the robot in $\mathrm{mm\,s^{-1}}$. The computed values v_l and v_r are the linear velocities the robots wheels should have for it to follow the line.

Algorithm 1: Algorithm used for following the path drawn on the phosphorescent surface. This is executed periodically and corrects the robots motion such that the difference between the centre sensor and the left (respectively right) sensor matches the difference between the centre and the right (respectively left) sensor.

```
1: procedure FOLLOW-LINE(v_t)
2:     (i_l, i_c, i_r) ← measured values of phototransistors
3:     d_l ← i_c − i_l
4:     d_r ← i_c − i_r
5:     v_l ← 2v_t(d_l/|d_l + d_r|)
6:     v_r ← 2v_t(d_r/|d_l + d_r|)
7:     set_linear_velocities(v_l, v_r)
```

This algorithm is called repeatedly to update the values of v_l and v_r at a given frequency f_u. Different values can be used for f_u.

Drawn line In order to test the functionality of line following a more complex pattern than a simple straight line is needed. The robots must also be able to follow the drawn line through corners. Thus a pattern using a sinusoidal motion was used. The robot drawing the line for others to follow moved in a sinusoidal manner while drawing to create a complex shape to follow. Figure 5.7 shows an example of a path used during testing. In the following paragraphs, the line L is defined as all positions where the drawing robot drove while the UV-LED was turned on.

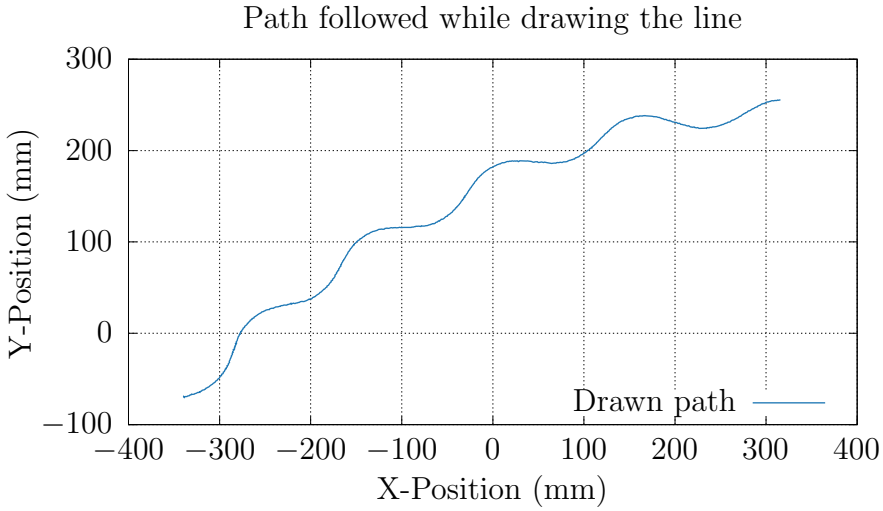


Figure 5.7: Example of a path used for testing the line following algorithm. This path was drawn using a sinusoidal motion to test the performance of the line following algorithm in more challenging conditions.

Criteria The criteria chosen to evaluate the performance of the line following were the velocity the robot could follow the line at as well as the distance the robot was away from the line. For the velocity the median was chosen, as the robot will come to a stop at the end of the line and turn around at a velocity near zero. The distance the robot is from the track at each time $d(t)$ is calculated as the minimum of all distances between the robot's current position $p_r(t)$ and any point belonging to the line.

$$d(t) := \min\{|p_r(t) - x| : x \in L\} \tag{5.3}$$

Measurements To measure the performance of the line following algorithm the distance to the drawn line is measured. For this the robot drawing the line is tracked and its positions are considered to form the drawn line to follow. The positions of the following robot are then recorded. From these positions the distance to the line can be computed at each recorded time point. Additionally the velocity of the robot is also computed for each time point.

To evaluate the algorithm under different condition two parameters were changed. The first parameter to be modified is the target velocity of the robot v_t. A faster

moving robot may deviate further from the drawn path as it can more easily overshoot the point at which it should turn. In addition to the target velocity the frequency with which adjustments are made was also modified. Both a frequency f_u of 100Hz and 10Hz were used in combination with a target velocity of $30\text{mm}\,\text{s}^{-1}$.

Frequency	**Target velocity**	**Median velocity**	**Average distance**
10Hz	$30\text{mm}\,\text{s}^{-1}$	$26.7\text{mm}\,\text{s}^{-1}$	4.8mm
100Hz	$30\text{mm}\,\text{s}^{-1}$	$27.1\text{mm}\,\text{s}^{-1}$	7.2mm
100Hz	$60\text{mm}\,\text{s}^{-1}$	$46.8\text{mm}\,\text{s}^{-1}$	7.7mm
100Hz	$15\text{mm}\,\text{s}^{-1}$	$13.6\text{mm}\,\text{s}^{-1}$	5.5mm

Table 5.1: Results of line following for different combinations of parameters. The median velocity is given as the velocity when turning at the end of the line is close to zero. The listed frequency is the frequency of updates to the commands sent to the motors, the target velocity is the velocity targeted by the robot while following the line. The distance given is the average distance the following robot was from the line.

Results Table 5.1 shows the results from running the line following algorithm. In general the measured velocity is lower than the desired target velocity v_t. This effect is to be expected as the robot has to make corrections to its trajectory that cause it to slow down. However the combination of a frequency of 100Hz and a target velocity of $60\text{mm}\,\text{s}^{-1}$ yielded a higher discrepancy than all other combinations. This can be attributed to the observed stuttering and getting stuck of the robot during this run. The robot would often stop on the line for short periods of time and had difficulty when turning at the ends of the line. The distribution of the velocity of the robot following at $60\text{mm}\,\text{s}^{-1}$ can be seen in Figure 5.8. As one can see, the robot is stopped more often than moving in the case of 100Hz and $60\text{mm}\,\text{s}^{-1}$. This explains the lower that expected average velocity.

In all measured runs the average distance from the line was acceptably small. It never exceeded 10mm. The offset can further be explained by the different locations of the involved components on the robot. The UV-LED is located at the back of the robot and follows a different path from the centre of the screen that is used for tracking. Thus the path of the line does not perfectly match the path recorded by the tracking system. However the distances were deemed small enough to not need to apply any corrections.

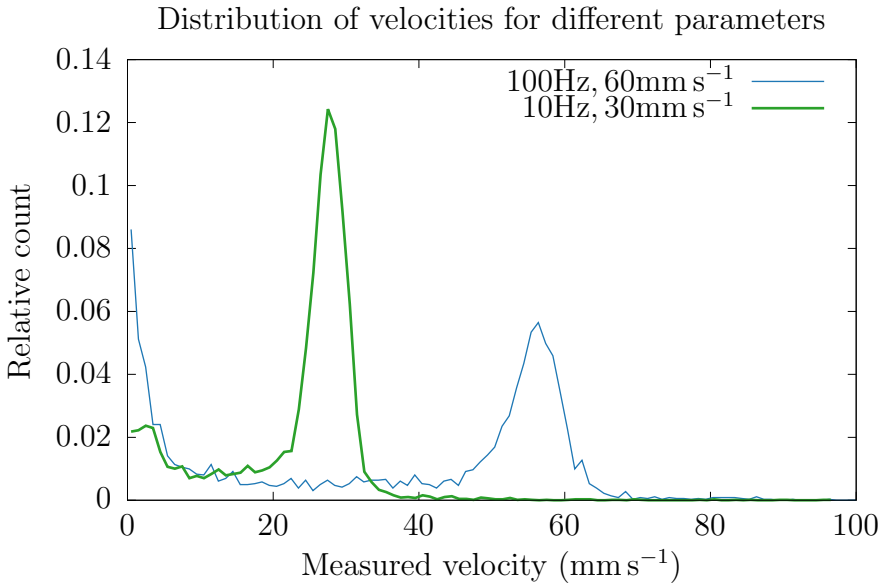


Figure 5.8: Distribution of measured velocities for two configurations. For 100Hz and 60mm s^{-1} the distribution is much wider and presents a distinct peak around 0mm s^{-1} while at 10Hz and 30mm s^{-1} the distribution is closer to the expected normal one.

5.2.2 Variance between robots

Some experiments require the robots to perform measurements of the brightness of the followed line. These measurements should be reproducible from one robot to another. It is thus important, that the robots measure similar values when exposed to the same stimulus. To test this the following experiment was conducted.

Each robot was placed on an LED with constant brightness at a fixed position. It then sent the measured brightness of the centre phototransistor to a computer over the network.

Results The results of this experiment are shown in Table 5.2. As one can see, most robots measured similar values. Only the robot with number 19 measured a significantly different level to the other robots. However even this difference could be eliminated by calibrating the robots. The values shown are raw sensor values without any corrections applied.

Robot	Mean	Standard deviation
10	12503.3	26.1
11	13616.0	137.4
16	14990.8	83.7
19	21343.1	146.7
22	15048.8	28.2
23	12579.8	49.9
25	17510.5	98.4

Table 5.2: Measured brightness values for different robots over an LED with constant brightness. The robots were fixed in place and the LED was located 22mm below the centre phototransistor. The values given are raw sensor counts.

5.3 Problems

During testing some problems with the design of the PCB have surfaced. In this section these problems will be examined and possible solutions for future version of the robot will be explored.

5.3.1 USB connector

The USB connector on the PCB is not mounted in a solid fashion. It should be mounted via four mounting pads on the bottom, soldered to the PCB. However this is not strong enough to hold the connector down when some force is applied. On some PCBs the USB connector has already been removed when plugging in.

Proposed solutions In future iterations of the design one could chose a connector mounted with through holes. This will greatly increase the stability of the mounting but may also require more space on the PCB.

5.3.2 Wi-Fi connectivity

The delivered PCBs have a problem when connecting to a Wi-Fi-network. As soon as the Wi-Fi-module is enabled the MCU restarts. No debugging information as to why the MCU restarted could be viewed.

Possible reason The most plausible reason for the MCU restarting when enabling Wi-Fi connectivity is insufficient grounding. For a stable functionality the MCU requires a stable ground plane when using the Wi-Fi module. This is not present on the PCB.

Proposed solutions The solution to the problem is to add a larger ground plane to the PCB. This can be an internal layer mostly made of copper connected to the ground of the circuits. An interim solution for the already existing robots is the create a grounding surface by connecting the conductive body of the MCU to a metallic strip. The same aluminium strip as for the UV-LED and phototransistors can be used.

5.4 Future improvements

The use of the new robots has revealed some points where improvements can be achieved. These can be added into possible future designs or, if possible, retrofitted to the robots by using the expansion connectors.

5.4.1 Motion

During the testing the wheels of the robots often had problems while turning. The mechanism with the gears was not as sturdy as hoped and the wheels could easily get jammed. A solution to this could be the use of different motors where the motor has sufficient torque not to require gearing. Alternatively the motor could also have a gearbox where the risk of such jamming is reduced as the movement of the gears is limited.

5.4.2 Wheel rotation

A brushed DC motor with a gearbox could be a good solution to prevent this problem from appearing. However this would require a new method for tracking the wheels rotation. This would also be useful for detecting the jamming of the wheels. A possible solution would be the use of some infrared beam break sensors around a spoked wheel to detect if the wheel is turning. This can also be used to count how far the wheel has turned compared to what was commanded.

5.4.3 Light sensing

The phosphorescent surface presents a challenge for the current equipment on the robot. Its maximum measured brightness was on the order of 10^2lx while some signal can still be detected at as low as 10^{-6}lx. This large difference poses some problems for the linear measuring apparatus on the robots. Currently the software has to switch the sensitivity of the ADC when the exact brightness needs to be measured. One can also make the choice to accept the clipping of the signal at the maximum measured voltage.

A possible solution for this is the use of a logarithmic amplifier before the ADC. This allows for the measurement of very different brightnesses with a single circuit and a single configuration.

5.5 Conclusion

The developed robots can accomplish the required tasks. The specific task of following a line on the phosphorescent surface is simplified by the design of the hardware and placement of the phototransistors. Some problems with the hardware remain. These appear to be solvable by tweaking the design to address the found issues. Overall the development of the robot was successful as early observations of their use in experiments show a much higher reliability and predictability when compared to the previously used ones.

References

[1] V. Witte, A. B. Attygalle, and J. Meinwald, "Complex chemical communication in the crazy ant Paratrechina longicornis Latreille (Hymenoptera: Formicidae)," *Chemoecology*, vol. 17, no. 1, pp. 57–62, Mar. 2007.

[2] V. Di Pietro, P. Govoni, K. H. Chan, R. C. Oliveira, T. Wenseleers, and P. van den Berg, "Evolution of self-organised division of labour driven by stigmergy in leaf-cutter ants," *Scientific Reports*, vol. 12, no. 1, p. 21971, Dec. 2022.

[3] G. Theraulaz and E. Bonabeau, "A Brief History of Stigmergy," *Artificial Life*, vol. 5, no. 2, pp. 97–116, Apr. 1999.

[4] M. Salman, D. Garzón Ramos, K. Hasselmann, and M. Birattari, "Phormica: Photochromic Pheromone Release and Detection System for Stigmergic Coordination in Robot Swarms," *Frontiers in robotics and AI*, vol. 7, p. 591402, 2020.

[5] G. Valentini, A. Antoun, M. Trabattoni, B. Wiandt, Y. Tamura, E. Hocquard, V. Trianni, and M. Dorigo, "Kilogrid: A novel experimental environment for the Kilobot robot," *Swarm Intelligence*, vol. 12, no. 3, pp. 245–266, 2018.

[6] A. A. Khaliq, M. Di Rocco, and A. Saffiotti, "Stigmergic algorithms for multiple minimalistic robots on an RFID floor," *Swarm Intelligence*, vol. 8, no. 3, pp. 199–225, 2014.

[7] F. Arvin, T. Krajník, A. E. Turgut, and S. Yue, "COSΦ: Artificial pheromone system for robotic swarms research," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 407–412.

[8] X. Sun, T. Liu, C. Hu, Q. Fu, and S. Yue, "ColCOS Φ: A multiple pheromone communication system for swarm robotics and social insects research," in *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2019, pp. 59–66.

[9] A. Font Llenas, M. S. Talamali, X. Xu, J. A. Marshall, and A. Reina, "Quality-sensitive foraging by a robot swarm through virtual pheromone trails," in *Swarm Intelligence: 11th International Conference, ANTS 2018, Rome, Italy, October 29–31, 2018, Proceedings 11*. Springer, 2018, pp. 135–149.

[10] S. Adams, D. Jarne Ornia, and M. Mazo, "A self-guided approach for navigation in a minimalistic foraging robotic swarm," *Autonomous Robots*, vol. 47, no. 7, pp. 905–920, Oct. 2023.

[11] N. R. Hoff, A. Sagoff, R. J. Wood, and R. Nagpal, "Two foraging algorithms for robot swarms using only local communication," in *2010 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2010, pp. 123–130.

[12] R. Russell, "Heat trails as short-lived navigational markers for mobile robots," in *Proceedings of International Conference on Robotics and Automation*, vol. 4, Apr. 1997, pp. 3534–3539 vol.4.

[13] R. Fujisawa, H. Imamura, T. Hashimoto, and F. Matsuno, "Communication Using Pheromone Field for Multiple Robots," Oct. 2008, pp. 1391–1396.

[14] J. C. Brenes-Torres, F. Blanes, and J. Simo, "Magnetic trails: A novel artificial pheromone for swarm robotics in outdoor environments," *Computation*, vol. 10, no. 6, p. 98, 2022.

[15] M. Salman, D. Garzón Ramos, K. Hasselmann, and M. Birattari, "Phormica: Photochromic pheromone release and detection system for stigmergic coordination in robot swarms," *Frontiers in Robotics and AI*, vol. 7, p. 591402, 2020.

[16] M. L. Kronemann and V. V. Hafner, "Lumibots: Making emergence graspable in a swarm of robots," in *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, 2010, pp. 408–411.

[17] S. U. Dübener, A. Jacker, A. A. Morgner, H. F. Haupt, and J. Wagner, "Dezibot2 - The Wi-Fi based, inexpensive, and small educational mobile robot," in *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS)*. The Steering Committee of The World Congress in Computer Science, Computer ..., 2018, pp. 146–150.

[18] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 3293–3298.

[19] K. Popp and P. Stelter, "Stick-Slip Vibrations and Chaos," *Philosophical Transactions: Physical Sciences and Engineering*, vol. 332, no. 1624, pp. 89–105, 1990.

[20] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a Robot Designed for Education in Engineering," in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, vol. 1. Portugal: IPCB: Instituto Politécnico de Castelo Branco, 2009, pp. 59–65.

[21] F. Mondada, E. Franzi, and A. Guignard, "The Development of Khepera," in *Experiments with the Mini-Robot Khepera, Proceedings of the First International Khepera Workshop*, ser. HNI-Verlagsschriftenreihe, Heinz Nixdorf Institut. 64, 1999, pp. 7–14.

[22] M. McMickell, B. Goodwine, and L. Montestruque, "MICAbot: A robotic platform for large-scale distributed robotics," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2, 2003, pp. 1600–1605 vol.2.

[23] A. P. Sabelhaus, D. Mirsky, L. M. Hill, N. C. Martins, and S. Bergbreiter, "TinyTeRP: A Tiny Terrestrial Robotic Platform with modular sensing," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 2600–2605.

[24] M. Jdeed, S. Zhevzhyk, F. Steinkellner, and W. Elmenreich, "Spiderino - A low-cost robot for swarm research and educational purposes," in *2017 13th Workshop on Intelligent Solutions in Embedded Systems (WISES)*, Jun. 2017, pp. 35–39.

[25] M. L. Carnot, "Entwicklung eines kamerabasierten Tracking-Systems für die Roboterplatform Dezibot mithilfe von OLED-Bildschirmen," 2023.

[26] "ESP32-S3 Series Datasheet." [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf

[27] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*. IEEE, 2011, pp. 1–7.

[28] S. Guo, J. Wu, Z. Wang, and J. Qian, "Novel MARG-Sensor Orientation Estimation Algorithm Using Fast Kalman Filter," *Journal of Sensors*, vol. 2017, p. e8542153, Sep. 2017.

[29] W. J. Westerveld, Md. Mahmud-Ul-Hasan, R. Shnaiderman, V. Ntziachristos, X. Rottenberg, S. Severi, and V. Rochus, "Sensitive, small, broadband and scalable optomechanical ultrasound sensor in silicon photonics," *Nature Photonics*, vol. 15, no. 5, pp. 341–345, May 2021.

[30] M. Toa and A. Whitehead, "Ultrasonic sensing basics," *Dallas: Texas Instruments*, pp. 53–75, 2020.

[31] Y. Li, H. Yu, S. Liu, X. Huang, and L. Jiang, "A CMOS Time-to-Digital Converter for Real-Time Optical Time-of-Flight Sensing System," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 113–119, Aug. 2018.

[32] R. Kaufmann, M. Lehmann, M. Schweizer, M. Richter, P. Metzler, G. Lang, T. Oggier, N. Blanc, P. Seitz, G. Gruener, and U. Zbinden, "A time-of-flight line sensor: Development and application," in *Optical Sensing*, vol. 5459. SPIE, Sep. 2004, pp. 192–199.

[33] R. Lange, S. Böhmer, and B. Buxbaum, "11 - CMOS-based optical time-of-flight 3D imaging and ranging," in *High Performance Silicon Imaging (Second Edition)*, ser. Woodhead Publishing Series in Electronic and Optical Materials, D. Durini, Ed. Woodhead Publishing, Jan. 2020, pp. 319–375.

[34] G. Benet, F. Blanes, J. Simó, and P. Pérez, "Using infrared sensors for distance measurement in mobile robots," *Robotics and Autonomous Systems*, vol. 40, no. 4, pp. 255–266, Sep. 2002.

[35] T. Mohammad, "Using ultrasonic and infrared sensors for distance measurement," *World academy of science, engineering and technology*, vol. 51, pp. 293–299, 2009.

[36] P. Horowitz and W. Hill, *The Art of Electronics*, 3rd ed. Cambridge University Press.

[37] G. de Graaf and R. F. Wolffenbuttel, "Smart optical sensor systems in CMOS for measuring light intensity and colour," *Sensors and Actuators A: Physical*, vol. 67, no. 1, pp. 115–119, May 1998.

[38] P. Prochazka, D. Cervinka, J. Martis, R. Cipin, and P. Vorel, "Li-Ion Battery Deep Discharge Degradation," *ECS Transactions*, vol. 74, no. 1, p. 31, Dec. 2016.

[39] K. C. Galloway, J. E. Clark, M. Yim, and D. E. Koditschek, "Experimental investigations into the role of passive variable compliant legs for dynamic robotic locomotion," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 1243–1249.

[40] N. Kashiri, A. Abate, S. J. Abram, A. Albu-Schaffer, P. J. Clary, M. Daley, S. Faraji, R. Furnemont, M. Garabini, H. Geyer, A. M. Grabowski, J. Hurst, J. Malzahn, G. Mathijssen, D. Remy, W. Roozing, M. Shahbazi, S. N. Simha, J.-B. Song, N. Smit-Anseeuw, S. Stramigioli, B. Vanderborght, Y. Yesilevskiy, and N. Tsagarakis, "An Overview on Principles for Energy Efficient Robot Locomotion," *Frontiers in Robotics and AI*, vol. 5, 2018.

[41] R. Siegwart, P. Lamon, T. Estier, M. Lauria, and R. Piguet, "Innovative design for wheeled locomotion in rough terrain," *Robotics and Autonomous Systems*, vol. 40, no. 2, pp. 151–162, Aug. 2002.

[42] E. Bolte, *Elektrische Maschinen: Grundlagen · Magnetfelder · Erwärmung · Funktionsprinzipien · Betriebsarten · Einsatz · Entwurf · Wirtschaftlichkeit*, 2nd ed., ser. SpringerLink ; Bücher. Springer Vieweg, 2018.

[43] A. Morar, "Stepper motor model for dynamic simulation," *Acta Electrotehnica*, vol. 44, no. 2, pp. 117–122, 2003.

[44] V. Semiconductors, "VELM6040." [Online]. Available: https://www.vishay.com/docs/84276/veml6040.pdf

[45] G. Chisiu, N.-A. Stoica, and A.-M. Stoica, "Friction Behavior of 3D-printed Polymeric Materials Used in Sliding Systems," *Materiale Plastice*, vol. 58, no. 1, pp. 176–185, Apr. 2021.

[46] E. Rabinowicz, *Friction and Wear of Materials*, 2nd ed. John Wiley & Sons, 1995.

[47] "DRV8834 Dual-Bridge Stepper or DC Motor Driver." [Online]. Available: https://www.ti.com/lit/ds/symlink/drv8834.pdf

[48] "AP2112 CMOS LDO regulator." [Online]. Available: https://www.diodes.com/assets/Datasheets/AP2112.pdf

[49] M. M. Kabir and D. E. Demirocak, "Degradation mechanisms in Li-ion batteries: A state-of-the-art review," *International Journal of Energy Research*, vol. 41, no. 14, pp. 1963–1986, 2017.

Acknowledgements

I would like to thank Prof. Middendorf for the opportunity to write this thesis at his research group. Especially as the topic of this thesis does not lie within the usual topics of his department. I would also like to thank Tobias Jagla for their continued support throughout the development of the robot. Their involvement was often crucial for the success of the work. Without them these robots would still remain an undeveloped idea.

Additionally I also recognise Oliver Welz for their contributions and ideas for the mechanical design of the robots. Their input was always insightful and contributed to a well-working final result.

I would also like to thank Ivan Cygankov and Johannes Walter as both contributed many hours and ideas to the development of the robots.

A Hardware documentation

A.1 Part list

Non-soldered parts

1× 3D printed PCB-holder

1× 3D printed bottom chassis

1× 3D printed front chassis

1× 3D printed back chassis

1× 3D printed battery-holder module

2× 3D printed wheel

2× O-ring 1 × 18mm

6× Screw ISO4762 M2x8

4× Screw ISO4762 M2x12

4× Washer ISO7089 M2

2× Screw ISO4762 M2x6

2× Stepper motor

4× TCRT1050 IR-sensors

1× SSD1306 OLED display

1× 1000mAh battery

4× 20mm × 48mm rectangle of aluminium at 0.1mm thickness for the reflectors around the UV-LED and the front phototransistors

Soldered parts A detailed list of the soldered parts can be obtained from the PCB-design included in the additional data and on the disc. The schematics of the PCB can be found here: `https://git.informatik.uni-leipzig.de/ds62myry/murmecha` at tag `rev1.hardware`.

A.2 Pin layout

The following table shows a list of the used GPIO pins on the MCU. It also describes their function. All pins that are not listed are not connected and have no purpose.

GPIO	Name	Description
1	MICROSTEP_ENABLE	Enable/Disable micro-stepping in the motor drivers
2	IR_SENS_ACTIVE	Enable/Disable active measuring for IR-Sensors
3	SDA	I^2C data line
4	SCL	I^2C clock line
6	ERASE_LED	Reserved for future use with an LED capable of erasing the photo-switch. Pads are present on the PCB.
7	RIGHT_MOTOR_FAULT	Pulled low when right motor-driver has a fault.
8	LEFT_MOTOR_FAULT	Pulled low when left motor-driver has a fault.
9	RGB_LED	Controls the RGB-LEDs.
10	LEFT_MOTOR_DIR	Controls the direction of the left motor.
11	RIGHT_MOTOR_DIR	Controls the direction of the right motor.
12	TOUCH_BTN	Connects to the button mounted at the front of the PCB.
13	LEFT_MOTOR_STEP	Sends step command to the left motor.
14	RIGHT_MOTOR_STEP	Sends step command to the right motor.
15	LEFT_MOTOR_ENABLE	Enables the left motor.
16	RIGHT_MOTOR_ENABLE	Enables the right motor.
17	IMU_SDA	I^2C data line for IMU.
18	IMU_SCL	I^2C clock line for IMU.
21	UV_LED	Controls the UV-LED.
34	CS	Chip-select line for SPI
35	MOSI	Output line for SPI
36	CLK	Clock line for SPI
37	MISO	Input line for SPI

Table A.1: Layout of GPIO pins on the rev 1 board.

A.3 I²C Addresses

The following table shows a list of the I²C addresses of the mounted components. When adding components to the I²C bus please ensure there is no conflict.

Table A.2: List of used I²C addresses on the main bus

Address	**Chip**	**Description**
0x0B	LC709203F	Battery monitor
0x10	VEML6040	RGB sensor
0x3C	SSD1306	Display
0x48	ADS1115	ADC for phototransistors
0x49	ADS1115	ADC for IR sensors.

B Software

The robots also have a dedicated software library to enable easier development for the experiments. The library can be found here: `https://git.informatik.uni-leipzig.de/ds62myry/murmecha-core`. The documentation for the library is located here: `https://git.informatik.uni-leipzig.de/ds62myry/murmecha-core/-/wikis/home`.