RAM. ROBOTICS AND MECHATRONICS

HARDWARE DESIGN OF A MOBILE EDUCATION ROBOT, THAT USES RASPBERRY PI 4 AND FPGA-BOARD FOR REAL-TIME CONTROL

R. (Rob) Binnenmars

BSC ASSIGNMENT

Committee:
dr. ir. J.F. Broenink
dr. ir. G. van Oort
ing. M.H. Schwirtz
dr. ir. A.Q.L. Keemink

July, 2022

023RaM2022
Robotics and Mechatronics
EEMCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | TECHMED CENTRE
UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Summary

To learn how to program a mobile robot, mobile robots are needed. Because RaM and the Robot Education Lab wants a new robot, and because with the available robots the real time control can not be accessed, a mobile robot needs to be made. In this assignment, the mechanical aspects and hardware of the robot are designed and made. The control unit of the robot is done by another student.

The two main requirements are: the robot needs to be mobile and the robot needs to have space for a robotic arm. This robot is divided into 3 separate parts, the mechanics, the drive train, and the power system. The wheels and the frame are analysed and the components are chosen. The final robot uses a rectangular frame of 24 by 34 cm made from aluminium profiles. Four wheels are used, two of which are round drive wheels of 70mm in diameter, and the other two are swivel wheels to make the frame stable.

The motor specifications, torque, RPM and power are calculated. With these specifications, several motors were found. Because of a tight project schedule, the components having a short delivery time had to be chosen. This restricted the motor choice to a reasonable quality choice, instead of a high quality Macon motor.

To drive the motor, a motor driver of RaM was used, which has a VNH2SP30-E H-bridge motor driver chip on it. The inputs to these drivers and the output of the encoders are connected to an icoBoard, which contains an FPGA. The icoBoard is connected via spi to a Raspberry Pi 4. All of the components are powered by a rechargeable 12V Nickel Metal Hydride battery pack, with a 5V DC-DC converter for the correct voltage for the Raspberry pi.

The robot is constructed, everything is connected mechanically or electrically. The robot is tested, and meets most requirements, but not all. The robot is slower than wanted, the robot could reach a velocity of 0.42 m/s, the required velocity was 0.5 m/s. But the robot could accelerate to this velocity in 1.09 seconds, which is well below the required 2 seconds. The encoders also did not meet the requirements, the pulse count was 3014-3100 with one rotation of the wheel, while 3408 was expected from the data sheet. The motors of the robot were required to move 10kg, the final weight of the robot is 2.5 Kg. Implying a payload or robotic arm of 7.5 Kg can be added. Due to some software issues, a 20-sim controller could not be tested to drive a trajectory. With some adjustments and better components this robot could be used by the Robot Education Lab.

Contents

1 Introduction

1.1 Context

Robots are used for more and more applications. To make these robot, people are need, who design and program the robots. To learn programming a robot and get hands on experience education robots are needed. With the robots that are available, the real-time control can not be accessed. If a good robot is not available, a robot needs to be designed and made.

1.2 Problem statement

For the robot education lab (REL) a mobile robot is needed that is controllable by a Raspberry Pi 4 with icoBoard. For the REL the robot should be able move with some precision, and possibility to add a robotic arm. From these requirements more technical requirements are defined. To make the robot move with precision a frame, wheels, motors for the wheels and sensor to know how much the robot moved. For the possibility to add an robotic arm the robot needs to have room for the arm, have enough power to move the robot with that arm, and be stable.
From all the requirements, technical specifications can be calculated and decided, for example, what are the motors specifications, what motor driver to use and how much capacity the battery needs to have. Further, this is an Bsc assignment, which means this project needs to be done in 10 weeks. Decisions need to be made what is more important and what can be less than ideal. More important these decisions need to be documented as to why that decision was made.
Another student designed the software (Vinkenvleugel, 2022). Everything from the output of the control unit to the wheels and back need to be designed in the scope of this project, together with how everything is connected mechanically and electrically.

1.3 Project goals

The goal of this bachelor's assignment is to design and make the hardware and mechanics for a mobile education robot. This goal is split up into 3 different sub-goals. These are:

- Design and make the mechanical aspects of the robot, which consists of the frame, base and wheels.
- Design and make the drive train of the robot, which consists of the motor driver, motor, gearbox, encoder and the control of the robot.
- Design and make the power system of the robot in the scope of this project, which are the battery and the voltage regulators.

In Figure 1.1, the components of the robot are displayed with a colour for each category, with an arrow between components that have a mechanical or electrical connection.

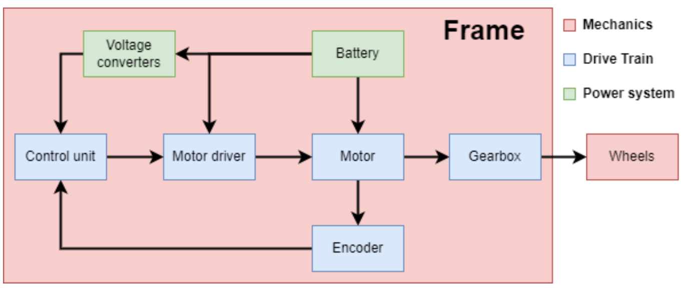


Figure 1.1: Overview of the robot's components

1.4 Requirements

To make decision about the robot, requirements are needed to substantiate the decisions. As stated in the problem statement, the robot must to move with some precision, and possible to add a robotic are. To have room for the robotic arm the size of the robot should be around the size of an A4 paper. To prevent the robot falling over if the robot moves or if the robot arm moves the robot should be stable. The motors need to be able to move the robot, for this a weight limited of 10 kg. The should drive to the robot in a timely manner, for this a nominal velocity of 0.5 m/s and accelerate to that velocity in 2 seconds. To move with some precision, feedback to a controller is needed to see how much is move. For this robot an error of 0.5 cm per meter was chosen or an accuracy of 99.5%. Not every surface the robot will drive over is level, this means the robot should also be able to drive up a slope, for this a slope of 8.5° or slope of 15%. To accommodate for a limited experience in constructing a robot, the ease of construction is also big element in choosing the components for this robot. To experiment or test the robot, a mobile rechargeable power source is needed that can power the robot for at least one hour. The robotic arm and other sensors have inputs and outputs, ports are needed to connect to. The control unit is already chosen, all of these components need to be controllable by a Raspberry Pi 4 with icoBoard. All of the requirements are restated in the following list.

Requirements

- Area of an A4
- Be Stable
- Weight of (at most) 10 kg
- Linear velocity of 0.5 m/s
- Accelerates to the desired velocity in 2 s
- Can drive up a slope with an angle of 8.5°
- Movement with at least 99.5% accuracy.
- Ease of construction
- Powered by rechargeable batteries
- Power the whole robot for at least one hour
- Possibility to add additional sensors and actuators
- Raspberry Pi 4 as the main computing unit
- icoBoard with FPGA for I/O

1.5 Report organization

In Chapter 2 background information is given, Then Chapters 3, 4 and 5 are each a chapter on the mechanics, drive train and the power system respectively. In the Chapter 6, the robot is integrated, constructed and tested.

2 Background

2.1 Existing education robots

Several robots for education exist. Some examples,

- the R2-G2P (Maljaars, 2008)
- GoPiGo (Dexter-Industries, 2022)
- Irobot create (iRobot, 2022)
- Turtlebot (Turtlebot, 2022).

The R2-G2P is a 15-year-old robot made for a robotics masterclass at the University of Twente, it is a small rectangular robot with distance and black line sensors. This robot is too small and too old for the REL to use. The Turtlebot 3 is a rectangular robot of 2 different heights and sizes, controlled by a raspberry pi, it has a velocity of approximately 0.25 m/s, but can carry 15 or 30 kg(depends on which version). This seems a good robot for the REL, but the real time control can not be accesed. This robot is also used for the master course robotics at the KU Leuven (Amsters and Slaets, 2020). The GoPiGo robot is a more toy-like robot that is controlled by a Raspberry Pi, it has 3 wheels and a camera. Here real time control can be added to the robot, but the robot is too small, the robot and especially the motors quality are bad. The iRobot create 3 is like the Roomba vacuum cleaner, it is a round robot with several sensors. Here again the real time control can not be accessed. The robot for this project needs to make a robot that do not have these problems.

2.2 Real-time systems

Real-time systems are event-driven systems with precise timing of events (Foundation, 2020). There are 3 different types of real-time systems: hard, firm, and soft. The difference between these is how the system reacts when it misses a deadline of an event.
Hard real-time systems have a strict deadline, if it misses an event, the system fails. Examples are printers, if the printer head does not move the print is ruined and needs to be redone.
With firm real-time, if some deadlines are missed, the system will not fail but the missed events are not useful anymore. But too many missed events would be a system failure. An example of firm real-time is streaming a video, if a few frames are missed the quality is lower but too much missed events and the video is not watchable.
Soft real-time can miss several events and the result of the events can still be useful. An example of this is opening a web page, if the URL of a link is click the web page is displayed, but if this takes a longer the system has not failed.

2.3 Odometry

Odometry is the measurement of distance, by using data from motion sensors (Ben-Ari and Mondada, 2018). If the velocity is known that and the time it took, the distance can be calculated that the robot drove. Measuring the time it took to drive an unknown distance is easy with the internal clock of the robot. The speed of the robot is more difficult because this is dependent on external forces. A way to measure the distance is to measure the rotations of a motor or wheel, which can be done by an encoder. With the size of the wheel and the number of wheel rotations, the distance can be calculated. This assumes there are no errors in the measurements of the wheel, size of the wheel and that the wheel does not slip. If this happens the robot can drift away from the true value, the real position of the robot does not match with where the robot "thinks" it is.

3 Mechanical aspects

The first part of the robot that has been analyzed and designed are the mechanical aspects of the robot, these are the form of the base of the robot, what wheels to use, and where to connect the wheels to the robot. To decide which options for the mechanical aspect the requirements from the introduction are used. The relevant requirements are:

- Area of an A4
- Be Stable
- Weight of (at most) 10 kg
- Ease of construction
- Can drive up a slope with an angle of 8.5°

Besides these requirements the robot will drive indoors over a mostly even surface, some small bumps or threshold should be be able to be passed. The turning radius should be as small as possible, so it can rotate is small places.

3.1 The base of the robot

Rectangular, round or triangular: a base can have a lot of different forms. A rectangular base or a base with straight sides is easier to make and connect the motor to. A round base can rotate in place easier, because the the distance to a side form the centre of rotation is the same all around the robot. If a robotic arm is added to the robot the distance to each side is equal (if the arm is placed in the middle). Both of these reason makes it a good option for the robot.
To make the robot easier to build Boikon is used (Boikon, 2022), these are straight aluminum profiles of different sizes, for this project 2 by 2 cm thick pieces are used. These profiles can be connected with right angles. A decision table is made for 4 common shapes, shown in Table 3.1. How easy to construct the base is given a higher weight, because of the Boikon. The rectangular form has the best score and is the chosen form of the base for the robot.

Table 3.1: Decision table of the base form

Base form	Easy to constuct	Stability	Easy to add motors	Easily to rotate	Total weighted sum
Weight	2	1	1	1	
Round	N	Y	N	Y	2
Rectangle	Y	Y	Y	N	4
Hexagon/octagon	N	Y	Y	Y	3
Triangle	N	Y	N	N	1

3.2 Wheels of the robot

There are a lot of different wheels, round wheels, tracks, Omni wheels, legs, and many more. All of these have their use. Round wheels are easy to construct, easy to connect to a robot, the robot can go fast with them, and does not give many inaccuracies in the odometry. Tracks are for uneven or rough terrain, because of their good grip and big surface contact. Tracks are harder to construct and give many inaccuracies in odometry. Omni wheels can drive in all directions but are harder to construct and control. Legs are hard to construct and control.
Not all wheels have the same purpose. Drive wheels are wheels connected to a motor to move the robot. Support wheels support the robot, to stabilize the robot. The support wheels do not need to be the same form as the drive wheels, but they can be. Some often used support wheels are a ball caster (a round ball in a housing), and a swivel wheel (a wheel that can rotate so it can turn in all directions).
For the drive wheels, round wheels are chosen because they are easy to control, give enough

grip, give the least inaccuracies in the odometry, and the robot is not for rough terrain. For the support wheels, the swivel wheel is chosen since the available ball joints were too small for this robot, else these were chosen because they give less friction. To decide how many of each type of wheels is needed, first the possible ways to steer the robot are discussed and decided. After that how many wheels and the size of the wheels can be chosen.

3.2.1 Steering

A robot can be steered several different ways to steer a robot. Ackerman steering which is used by cars, for which extra mechanics is needed to rotate some wheels and also a motor to preform steering is needed. Another way to steer is skid steering, where the robot turns by controlling the speed of each side of the robot. All the wheels on the same side are driven and rotate by the same motor, connected by gears or a chain. To turn some wheels slip, which gives inaccuracies in the odometry, which is not desirable. Another way to steer is differential steering, which turns the same way as skid steering. The difference is that only one wheel on each side is driven, given each side has more then one wheel, the other wheels on the robot are for support and need to be swivel wheels. This gives fewer inaccuracies in the odometry and has an easier mechanical construction.
For these reasons differential steering is chosen to steer the robot. This implies that two drive wheels are needed for the robot.

3.2.2 Number of wheels

A decision table is made for the number of wheels is shown in Table 3.2. Easy to construct is weighted heavier because of the limited experience in constructing a robot. From the table can be concluded that 4 wheels are the best option for the robot. For the differential steering 2 drive wheels are needed, this means that 2 swivel wheels are needed.

Table 3.2: Decision table of the number of wheels

Wheel Amount	Stable	Easy to construct	Easy steering	Total weighted sum
Weights	1	3	1	-
2	N	N	Y	1
3	N	Y	Y	4
4	Y	Y	Y	5
5+	Y	N	Y	2

3.2.3 Wheel size

The size of the wheels affects the torque and angular velocity calculations for the motor. The wheels need to be bought, implying only a few options. The wheels need to be big enough to not let the frame or motor touch the floor. Most of the possible motors are not more than 40mm in diameter. Wheels were found with a diameter of 70mm, this is enough to clear the possible motors from the ground.

3.3 Placement of the mechanical components

All of these components need to be placed on the frame or attached to the frame. The placement of the components can affect the robot's movement and stability a lot.

Stability

For an object to be stable and not fall over, the centre of gravity needs to be inside the base of support (BoS). The BoS is the area between all points of contact with the surface underneath the robot. A one-wheeled robot has one contact point, to be stable the centre of gravity needs to

be precisely above this point. With a 2 wheeled robot like a Segway, the centre of gravity needs to be above the axis between the wheels. Since it is a line it is easier to have it above this line, but if the segway is at even a small angle the center of gravity is not above this BoS anymore. Both of these needs a controller to keep the centre of gravity above the BoS. For more-wheeled robots the BoS is a plane between the contact points. For three-wheeled robots, the BoS is a triangular plane, for four-wheeled, it is a quadrilateral plane.
The height of the centre of gravity is also important. If it is higher, it is easier for the centre of gravity to be outside the BoS when the robot drives at an angle. This means the robot's centre of gravity needs to be within the BoS and as low as possible to be stable.

Component placement

To maximize the BoS the wheels need to be placed as far as possible from each other, for a rectangular frame this is at each corner of the frame. The robot should also turn easily with an as small as possible turn radius, to do this the drive wheels should be in the middle of the robot. To always have enough grip the drive wheels should always touch the floor. If the robot goes over an uneven floor of the robot should not tilt, which results in 1 or 2 of the drive wheels not touching the floor.
For 4 wheels, the best place for the drive wheels is in the middle of the robot, and a swivel wheel in the front and back side of the robot, this configuration is shown in Figure 3.1.
To support the motors, a support beam is used between the drive wheels. To further prevent the drive wheels not being able to touch the floor, is to let the swivel wheels not always touch the floor. Then the robot will drive at a slight angle, but when the robot need to cross a threshold the drive wheels still touch the ground.

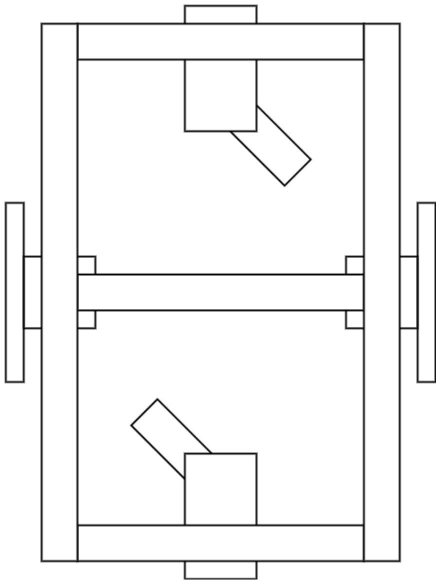


Figure 3.1: The frame of the robot

4 Drive Train

The Drive train of this robot consists of several parts that are shown in Figure 1.1. The drive train consists of 5 components:

- Motor
- Gearbox
- Encoder
- Motor driver
- Controller

First, the motor is discussed with how to calculate the motor specifications, and then the encoder is discussed. A motor, gearbox, encoder combination is chosen out of several possible options. After that, the motor driver is discussed and chosen. For the robot the Raspberry PI 4 with icoBoard was already chosen, only the controller inputs, outputs and how to handle the encoder will be discussed. The requirements for the drive train are:

- Maximum linear velocity of 0.5 m/s
- Accelerates to the desired linear velocity in 2 s
- Weight of (at most) 10 kg
- Can drive up a slope with an angle of 8.5°
- Raspberry Pi 4 as the main computing unit
- icoBoard with FPGA for I/O
- Movement with at least 99.5% accuracy.

4.1 Motor

Different types of electric motors are possible: AC motors, DC brushless motors, DC brushed motors, servo motors, and stepper motors (Prasaath, 2017). Each has its use, but for fast rotation of the wheel DC (brushed/brushless) and AC are the most useful. AC motors, like the name suggest, need alternating current to operate and need little maintenance.
DC motors work on direct current, 2 kinds DC motors are: brushed motors and brushless motors. The difference is that brushed motors transport energy over brushes from the stator (the stationary part of a motor, the housing) to the rotor (the rotating part, axial). With brushless motors, the rotor is made from permanent magnets and does not need energy, the stator is made from electromagnets, which polarity needs to be controlled to rotate the motor. Since no brushes are used that need maintenance and be replaced. Brushed motors need more maintenance to replace the brushes, but brushless motors needs a better motor driver to control the precise timing and polarity of the magnets.
For a mobile robot it is difficult to get AC current, as for this extra components are needed. AC motors are then not useful, so DC motors are then the choice for this robot. The decision between brushed and brushless DC motors depends on what is available with the right specifications, what motor driver are available, and what can be delivered in time. To get the right motor first the motor specifications need to be calculated, the torque, angular velocity and the power.

4.1.1 Torque

The maximum torque of a motor determines how much mass it can accelerate and if it even can move the robot. The total torque needed for the robot can be calcualted by

$$\tau_{robot} = F_{robot} \cdot r$$

where F_{robot} is the total force needed by the robot and r is the radius of the drive wheel. Because multiple motors are used and not each motor is 100% efficient, the force for each motor is

$$F_{robot} = \frac{F_{total}}{N \cdot \eta}$$

N the amount of motors is and η the efficiency of the motor. The total force needed is the worst-case scenario force, this is when the robot should accelerate from 0 to the desired acceleration on a slope. This force needed consists of three different forces, the acceleration force F_a, the Downhill-slope force F_H, and the friction force F_R. The total force is

$$F_{total} = F_a + F_H + F_R$$

In a Maxon manual are formulas or equations to get these forces (Maxon, 2022).

$$F_a = m \cdot a$$

$$F_H = F_G \cdot \sin\alpha$$

$$F_R = \mu \cdot F_G \cdot \cos\alpha$$

$$F_G = g \cdot m$$

Where m is the mass of the robot, α the angle of the slope, μ the friction coefficient, g the gravitational acceleration constant of $9.81\ \mathrm{m/s^2}$, and a is the acceleration of the robot.

4.1.2 Angular Velocity

Another specification of a motor is the speed at which the axial can rotate or the angular velocity. The unit the angular velocity is stated is RPM or Revolutions per minute or in radians per second. The angular velocity in rad/s is:

$$\omega = v/r$$

where V is the velocity of the robot and r the radius of the wheel. To get the angular velocity in RPM, it needs to be multiplied by $60/2\pi$.

4.1.3 Power

Power is also a useful specification of a motor since mechanical power is also the electrical power. The mechanical power is:

$$P = \omega \cdot \tau$$

with the angular velocity in radians per second.

4.2 Encoder

An encoder is a device with which the rotational speed and direction of something can be measured, for example, a motor shaft or wheel. This speed and direction can then be an input for loop control. There are different types of encoders(Goodwin, 2020), Absolute and incremental encoders. Absolute encoders know the precise location or orientation of the object, so even when the power is disconnected and reconnected the robot could be calibrated it own angle. Incremental encoders know the difference it has moved, it does not know the precise location/orientation. Then these types can also be measured in different ways (Smoot, 2019), magnetic encoders, optical encoders, and capacitive encoders. The wheels are round and the current orientation of the axial/wheel does not matter, which implies an absolute encoder is not needed and a incremental encoder is good enough. The option between the 3 methods to measure the rotation is less important, all 3 are possible.

Table 4.1: Variables to calculate the motor specifications

Requirement	Value
Velocity	0.5 m/s
Acceleration	0.25 m/s^2
Weight	10 kg
Wheel radius	0.035 m
Slope	8.5°
Efficiency	70 %

4.3 Chosen motor

4.3.1 Motor specifications

With the formulas and the derived equations and the following variables, the motor specifications are calculated. The motor specifications were calculated with these variables and shown in Table 4.2. Now several motors can be sought that meet the requirements or have more torque and angular velocity.

Table 4.2: Motor specifications per drive motor

Torque [Nm]	0.425
Angular velocity [RPM]	136.4
Power [W]	6.0

4.3.2 Chosen motor

With the specifications that are found a motor can be searched for. The torque that was calculated did not account for the friction force, implying the motor torque needs to be bigger. Also if bigger wheels are wanted, a higher torque is needed. For both of these reason the torque specification is doubled to at least 0.85 Nm. The RPM of the possible motors should be the wanted RPM, but this specification is less important, because it will still move only slower.

Since RaM often uses Maxon motors, this brand of motors are looked at first. Maxon motors are of really high quality and precision and many different options and combinations are available, but they have a long delivery time. For this reason, also other suppliers were considered, to see options that can be delivered within a week. Six motors were found that were possible motors to use, three of them are Maxon motors and the other three motors were found on Robotshop.com. In appendix A, is stated which motor has what motor,encoder and gearbox. With the Maxon motors each component (motor,gearbox,encoder) could be chosen separately. For the other motors each motor, gearbox and encoder were already combined.These 6 motors and their relevant specifications are shown in Table 4.3.

Table 4.3: Motor specification of the 6 different motors

Motor	Torque [Nm]	Angular velocity [RPM]	Power [W]	Quality	Delivery time	Cost [€]	Encoder pulses per rotation of the wheel	Axial diameter[mm]	Gear ratio XX:1	Velocity [m/s] (70mm wheel)	Rated load current [A]
RE30 Maxon motor	1.22	105.21	15	good	Long	774.19	11500	6	23	0.39	1.35
RE25 Maxon motor	0.93	155.21	10	good	Long	639.1	17500	6	35	0.57	1.24
DCX19S Maxon motor	1.27	97.29	16.1	good	Long	343.46	56832	4	111	0.36	1.35
74RPM robotzone motor	1.37	74	11	okay	1-3 days	67.2	1194	6	100	0.27	2
Cytron planetary gear motor	1.76	120	41.3	okay	1-3 days	70	245	8	49	0.44	5.5
104RPM robotzone motor	0.98	104	11	okay	1-3 days	67.2	854	6	71	0.38	2

To decide which motor to use a decision table is made, which is shown in Table 4.4. The technical specifications (torque, rpm, encoder pulses and motor quality) are essential for a good robot. The Torque of the motor is important, otherwise the robot can not drive. The RPM of the motor is less important, if this is lower than the requirements the robot will go slower but still move. The encoder criteria are needed to be controlled accurately with real-time control. The axial criteria is for the wheels, which have a 6mm axis. The technical specifications are the best for the motor modules from Maxon. The delivery time for these motors is really long and can not be delivered before the end of this assignment. For this reason, the delivery time criteria have a really high weighting of 10. This results in the Maxon motors falling off and the motors from the robot shop being the better option. For this assignment, 104RPM Robotzone motor is the best motor to use, the specifications of the motor and encoder are shown in the datasheets (City, 2022a,b).

Table 4.4: Motor decision table

Motor	Enough Torque (70mm wheel)	Enough RPM (bigger wheels)	Enough Torque (70mm wheel)	Enough RPM (bigger wheels)	Enough Power	Good Quality	Short delivery time	Enough encoder pulses per wheel rotation	Right axial diameter	Total weighted sum
Weight	3	2	3	2	2	2	10	2	1	-
RE30 Maxon motor	Y	N	Y	Y	Y	Y	N	Y	Y	13
RE25 Maxon motor	Y	Y	Y	Y	Y	Y	N	Y	Y	15
DCX19S Maxon motor	Y	N	Y	Y	Y	Y	N	Y	N	12
74RPM robotzone motor	Y	N	Y	N	Y	N	Y	Y	Y	19
Cytron planetary gear motor	Y	N	Y	Y	Y	N	Y	N	N	18
104RPM robotzone motor	Y	N	Y	Y	Y	N	Y	Y	Y	21

4.4 Motor driver

A motor changes electrical power into rotational mechanical power, the voltage across the motor is related to the angular velocity and the motor current is related to the motor torque. The motor can be controlled in two ways, with voltage or current. The motors are controlled by the Raspberry pi4 with an icoBoard with digital IO pins. This pin can not supply enough power to rotate the motor, to make this possible a motor driver is needed. The motor driver has multiple connections, Voltage connection to power the motor with the desired voltage and that can draw the desired current. A connection to connect the pins from the icoBoard and an output connection to the motor. The motor driver is controlled by 3 pins, 2 for the direction and one for the voltage to the motor. The direction of the motor is done by turning one of the 2 pins high and the other stays low, if both are low or both high the motor does not rotate. The voltage to the motor is controlled with a PWM signal, a duty cycle of 100% means that the same voltage level of the input is at the output. A duty cycle of 50% means that half of the input voltage level is at the output. The motor driver needs to supply the motor with voltage and current.
To know what motor driver can be used, the voltages and what current the motor driver need to be able to output needs to be known. From the datasheet (City, 2022a) of the motor, the motor has a stall current of 20A. This motor current will be checked if this is correct because it seems really high. If the motor current is correct the motor driver needs to be able to supply this current, or the start voltage should not be 12V. This is because the stall current is proportional to the motor voltage. The RaM department made its own driver which it uses for many projects, this driver contains a VNH2SP30 H-bridge motor driver (VNH, 2017). The chip can output 30A continuously, which is enough for the stall current. This motor driver is used because it is available and can handle the motor stall current.

4.5 Control unit

The encoder and motor driver are connected to the icoBoard. This icoBoard contains a FPGA, that can read the two encoder channels with a high frequency, at least 4 times per pulse, this is called x4 encoding. If an event is missed this can lead to missing a pulse or changing the pulse count in the wrong direction. For the robot, this means a loss of accuracy, for some programs this can lead to failure (driving into a wall or of a certain course) because of this is a firm real time system, or if no event can be missed hard real time. The pulse count can be send over a SPI connection to the Raspberry pi, this is done on a lower frequency but still real time. On the Raspberry Pi, the pulse count is an input to a controller. The controller output a duty cycle and direction, which is send back to the FPGA over the same SPI connection. In the FPGA the PWM and enable values are changed, if needed, and the voltage to the motor is changed.

4.6 Validation

To be integrated into the whole robot each component of the drive train needs to be validated first.

4.6.1 Motor

Method

The stall current of the motor needs to be checked if this is 20A at 12V. This is done by connecting the motor in series with a shunt of 0.18 Ω and then measuring the voltage of the shunt with an oscilloscope. The motor is stalled, and the voltage over the shunt is measured. To get the current through the motor and shunt ohms law is used

$$I = V/R$$

Where V is the voltage over the shunt and R the resistance of the shunt. To calculate the stall current at 12V,

$$I_{12V} = I_{Shunt} \cdot \frac{12V}{V_{cc} - V_{shunt}}$$

where I_{Shunt} is the current through the shunt at the 3 different supply voltages, V_{cc} the supply voltage, and V_{shunt} the voltage over the shunt.

Results

The experiment was done 5 times and the average of those 5 times was taken. Since the motor is strong and the power supply only could supply 2.5 A, the experiment was only tried on 1, 2 and 3 volt. The results are shown in Table 4.5. It can be seen that the stall current of the motor, 14.64A is high, but not as high as expected from the data sheet. There are some differences between the 3 stall currents at 12 V, this can be the result of not stalling the motor fully. The supply was already at the maximum current the power supply could give while stalling the motor with a supply voltage of 3 V.

Table 4.5: Stall current of the motor results

supply [V]	shunt [V]	Loop current [A] (at supply V)	Loop current [A] (at 12V)
1.0	0.203	1.13	16.98
2.0	0.363	2.01	14.78
3.0	0.463	2.57	12.17
		Average	14.64

4.6.2 Encoder

Method

To validate the encoder the wheel should be rotated a 10 times, by the motor, slow enough to be counted by the user. Then the pulse counter should be 10 times the counts per rotations. To be more accurate this experiment is done 5 times per motor. If the datasheets (City, 2022a,b) are correct and x4 encoding is used, the pulse count should be at ±3408 counts per rotation.

Results

Code was written that measured the encoder pulses. Then one wheel was rotated at a low speed for 10 rotations, 5 times for both wheels. The pulse count per rotation of each experiment is shown in Table 4.6, with average of the 5 results. The results are lower then expected. This can be the result of a measurement error by the user, measurement error by the controller or the pulse per turn are less then stated in the datasheet. It is most likely that the encoder has less counts per rotation of the wheel is less then stated, because the quality of the motors. If the original encoder value was used, the accuracy would be between 88-90%, which very inaccurate. To make is make it accurate enough to meet the requirement of 99.5%, the true encoder pulses per rotation needs to be known.

Table 4.6: Average pulse count for one rotation of the wheel

	Pulse Count right motor	Pulse Count left motor
Test 1	3043.0	3034.0
Test 2	3126.4	3033.1
Test 3	3132.7	2966.1
Test 4	3089.5	3013.9
Test 5	3113.3	3027.2
Average	3101.0	3014.9

4.6.3 Motor driver

Method

To see if the motor driver outputs the correct voltage to the motor, the input voltage and output voltage are measured with different PWM duty cycles, while a motor was connected to the motor driver as load. If the motor driver works correctly the output voltage should be the input voltage times the duty cycle.

Results

The motor driver was connected to a icoBoard and a 12V power supply as input. Then the voltage over the motor was measured. The measurement results are shown in Figure 4.1. The wanted voltage (duty cycle times input voltage) is also plotted. The biggest difference is at a duty cycle of 0.5 or 50%. This can be accounted for in the controller, but if a good controller is used, this is not a problem.

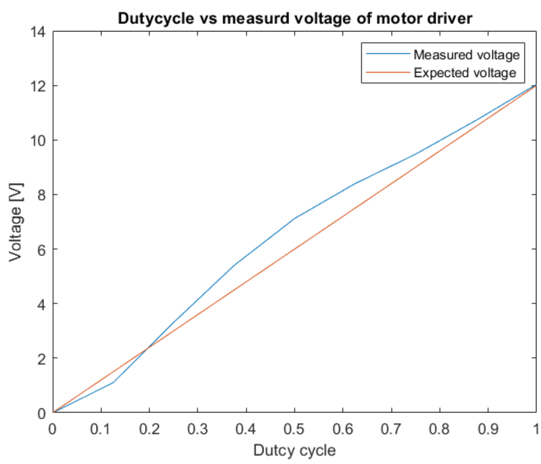


Figure 4.1: Measured voltage and expected voltage for different duty cycles.

5 Power System

5.1 Batteries

To power the robot batteries are used. For the power source, the following power requirements are set:

- Powered by rechargeable batteries
- Power the whole robot for at least one hour

It is the best if the battery voltage is the same as the voltage needed for the motors or higher, otherwise voltage converters are needed to get the voltage higher. If this is not done, then the motor can not reach the required speed.
To calculate the capacity needed the power usage(in watts) of each component needs to be determined and summed. Then multiply the total power usage by the desired operating time to get the watt-hours needed. To get the amp-hours, the watt-hour needs to be divided by the battery's voltage. There are different battery chemistry, Nickel Metal Hydride (NiMH), Lead-acid, Lithium-ion (Li-Ion), Lithium Polymer (Li-Po) and much more chemistry's. Some have a great power density, some last longer, and others cheap or are safer to use. A battery should not be discharged fully, if this is done the battery will be damaged. How far a battery can be discharged is different for each battery chemistry. The total capacity needs to be larger than the wanted capacity to not fully drain the battery.
To power the raspberry pi and to pull up some pins on the motor driver, 5 volts is needed. The 12 volts of the battery need to be reduced to 5 volts. This is done by a DC-DC converter.
For most devices or robots that run on a battery, a way exist to see the percentage of full capacity the battery is at or the state of charge (SoC) of the battery. For this, the current going in and out of the robot needs to be measured all the time. This has been done by measuring the battery's current and then integrating it. This can have some errors, because of integration errors. The control unit needs to keep track of it all the time, even when it was not powered for some time. In this robot, there is no system that checks the battery's state of charge.

5.2 Battery capacity

On the robot are, 2 motors that consume 11W at max efficiency (City, 2022a), and one RPI4 that consumes between 2.7 (when idle) and 6.4W (4 cores active) (Greeling, 2021). The components is shown in Table 5.1, Because 2 motors are used the power consumption of this is doubled. The power consumption of the motor driver, DC-DC converter and encoder are negligible.

If The robot needs to be powered for 1 hour then 28.4 watt-hours are needed. To get to ampere-hour the battery voltage is needed. Because the motors need 12V this is the battery voltage that is chosen. So the battery needs 2.37Ah ($28.4Wh/12V$). This is all theoretical battery capacity needed, but the motors will not turn at max efficiency for the whole hour, and the RPI4 will not use all 4 cores the whole hour.
A NiMH battery was found with a capacity of 3.2 Ah. The battery has 10 cells connected in series, NiMH cells have a nominal voltage of 1.2V. resulting in a battery voltage of 12V. When

Table 5.1: Theoretical power consumption per robot

	Power consumption [W]	Total [W]
Raspberry Pi 4	6.4	6.4
Motor	11	22
	Total	28.4

100% charged the cells are at a 1.5V, and when 0.9V when discharged, but for most of their discharge life the cells is at 1.2V. A fully charged battery also supplies more voltage, and an almost discharged battery less voltage. Because there is no regulator between the battery and input of the motor drivers and converter, this voltage is also supplied to the motors (at a duty cycle of 1). Nimh batteries were chosen because they are simpler to charge. To charge the batteries a charger was also bought, that can safely charge the battery.
To get the 5 volts a DC-DC converter is used. The DC-DC converter needs to be able handle the current drawn by the Raspberry pi, this is at most 1.28A. A DC-DC that could handle 5A at 5V was bought, while this is more then enough for the Raspberry Pi, if other sensors or components need 5V this DC-DC converter can still be used.

5.3 Validation

To validate the power system the battery should be checked and see what voltage it outputs, this was measured to be 12.4V. The conversion in the DC-DC converter from the batteries 12V to 5V, also worked and the output voltage was measured to be 5.1V.

6 Integration

6.1 Mechanical construction

The frame is made as in Figure 3.1. The only difference is that the boikon profiles where the swivel wheels are connected higher than the middle profile. This is because the swivel wheels are too big for the current drive wheels. this can be resolved by using bigger wheels. An acrylic plate is laser-cut with holes to fasten it to the frame and to fasten the electrical components. Two pictures of the robot are shown in Figure 6.1.

6.2 Electrical Connections

The battery, DC-DC converter, the motor driver, motor module and the icoBoard all have input and outputs that need to be connected. The connections between components are shown in Figure 6.2.
All electrical components are on one side with the battery (almost) in the middle, as is shown in 6.1. The battery is in the middle because this is the heaviest component, and to have the most grip the drive wheels need the most downforce and the swivel wheels as little as possible. The battery is bit offset to one side, to have space for the electrical components and to offset the weight of the other components.

6.3 Additional sensors

With the sensors that the robot now has, the robot can drive a certain distance and can turn a certain amount (if a good controller is used). But the only way to let the robot do these operations is to give it these commands by user input or by hard coding the distances and turn angles. To do more "fun" operations with the robot other sensors or actuators should be added. To let the robot drive towards a target like a bottle, the robot needs the location of the bottle. While this can be given to the robot by the user, other options are possible. A camera could be used to "see" the bottle, and then with some algorithms, the bottle can be detected and driven towards. To prevent the robot drive into something distance sensors can be used. These sensors can use, for example, ultrasonic waves and a sensor that picks the signals back up to determine if something is in front of the robot. These are only 2 of many sensors that could be attached to the control unit.

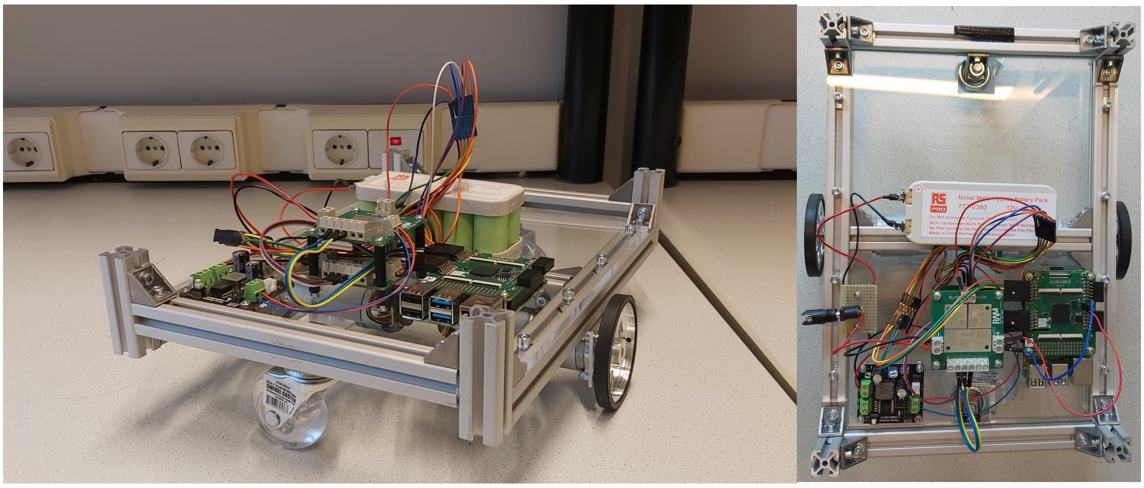
6.4 Validation of the whole robot

6.4.1 Method

To validate the whole robot the requirements are checked to see if they are met. Some are yes-no answers (for example: does it have rechargeable batteries) and some are validated with a measurement(for example: what is the velocity).

The following requirements do not need a measurement or experiment and are answered with a simple answer.

- What form does it have
- Stability: Can the drive wheels lose contact with the ground, with little pressure on a certain spot of the frame.
- Can the robot be powered by rechargable batteries
- Possibility to add other sensors
- Raspberry Pi 4 as the main computing unit
- icoBoard with FPGA for I/O



(a) Side view **(b)** Top view

Figure 6.1: Pictures of the robot

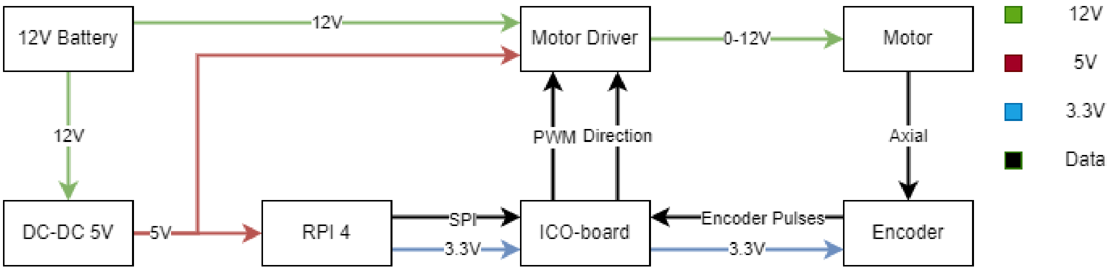


Figure 6.2: Connections between components

The rest of the requirements need to be measured or an experiment needs to be done, and the value needs to be calculated. The way to measure it or the experiment is in the following list.

- **Size**: measure with a ruler/measuring tape
- **Weight**: measure with a scale
- **Velocity**: let the robot drive for a specific distance and measure the time, this with the robot already driving at a constant speed
- **Acceleration**: letting the robot drive from stand still and measure the time, before it reaches the desired velocity.
- **Max slope**: the angle of a surface till the robot can not drive up the surface anymore, then measure the angle.
- **Power**: Use the robot for one hour

6.4.2 Results

The results of the validation are shown in Table6.1, most of the requirements are met, only the following requirements failed. The desired velocity of 0.5 m/s was not reached. The encoder accuracy result from Chapter 4 is to low. The duration the robot could be used on the batteries is unknown due not enough time to do this experiment. The velocity can be increased by using larger wheels are needed. Since the acceleration is well below 2 seconds, enough torque is left to use bigger wheels. The encoder accuracy can be increased by finding the true encoder pulses per rotation. The use time of the robot needs to be done. The others met the requirement.

Table 6.1: Validation result of the requirements

Requirement	Actual	Requirement met
Form	Rectangular	Yes
Rechargeable batteries	Yes	Yes
Possibility to add other sensors	Yes	Yes
Raspberry Pi 4 as the main computing unit	Yes	Yes
icoBoard with FPGA for I/O	Yes	Yes
Size	$27.5 \cdot 37$ cm	Yes
Weight	2.5 kg	Yes
Velocity	0.42 m/s	No
Time to reach desired velocity (acceleration)	1.09 s	Yes
Max slope	10°	Yes
Encoder precision	88-90%	No
Power	unknown	No

7 Conclusion and recommendations

7.1 Conclusions

The goal of this project was to design and make the mechanics and hardware of a mobile education robot. A 24 by 34 cm frame was designed with a rectangular form made with Boikon, with 2 drive wheels and 2 support wheels. The robot has 2 motors with encoders and a gearbox, each motor is powered by its own motor driver. The robot is controlled by a Raspberry Pi 4 and icoBoard. All of these components are powered by a 3.2mAh, 12V NiMH battery pack. For the robot, several requirements were set, and all of the requirements, except for the velocity, precision and operating time, were met. The weight of 2.5 kg is well below the max weight of 10 kg, a robotic arm or another payload can weigh 7.5 kg. On the robot there is space for a robot arm. The robot velocity, while the wheels are of the ground, is only 0.42 m/s, this is slower than wanted. The robot could accelerate to this speed in 1.09 s,this is well within the wanted 2 s. This was done with the wheels touching the ground, and the motors needing to accelerate the whole robot. The robot has rechargeable batteries, but the operating time was not tested. The robot was tried to work with a Raspberry Pi 4, the encoders and motor driver both work with the 3.3 V IO ports, but due to software issues, a control program could not be tested on the Raspberry Pi. In the end, a robot is made that can drive and has space for a robotic arm.

7.2 Recommendations

To improve this robot multiple components could be improved or added to the robot, some of these are:

- Other options for the frame should be considered
- The motors are noisy and not their datasheet are not correct or precise enough.
- The wheels need to be bigger or the motor needs to rotate faster
- The encoder should be tested more to see what the real pulse per rotation is.
- More sensors should be added to the robot.
- A circuit that monitors the state of charge of the battery should be added.
- The software issues need to be solved, and a controller should be made and tested.

Most of the components were chosen because they were available in time, not hard to construct or both. New decisions need to be made where these are not the biggest factor or maybe not a factor. With Boikon a rectangular frame was the only feasibly option, more forms should be considered for the frame. Maxon motors could be used as they are of better quality and have better datasheets than the ounce used currently. The motor driver might be updated to fir to these new motors The encoder pulse per rotation was not what was specified in the data sheet, this should be looked into further what the real pulse count is, if these motors and encoders are used. Now the robot can preform reprogrammed trajectories, with more sensors, the robot could decided it own trajectories to reach a destination. Now the battery state of charge is not monitored, a circuit should be added to see if the battery is almost low, otherwise the battery can be over discharged and damaged.
Due to software issues, the precision of the robot could not be tested, and no controller could be tested.

A Possible motors

RE30 Maxon motor

- Motor: RE 30 Ø30 mm, Precious Metal Brushes, 15 Watt
- Gear: Planetary Gearhead GP 32 A Ø32 mm, 0.75 - 4.5 Nm, Metal Version,
  Gear ratio: 33 : 1
- Encoder: Encoder HEDS 5540, 500 CPT, 3 Channels before gearbox or 16500 after gearbox

RE25 Maxon motor

- Motor: RE 25 Ø25 mm, Precious Metal Brushes CLL, 10 Watt
- Gear: Planetary Gearhead GP 32 A Ø32 mm, 0.75 - 4.5 Nm, Metal Version,
  Gear ratio: (23 : 1)
- Encoder: Encoder HEDS 5540, 500 CPT, 3 Channels before gearbox or 11500 after gearbox

DCX19S Maxon motor

- Motor: DCX19S GB KL 12V
- Gear: GPX22 A 111:1
  Gear ratio: 111;1
- Sensor: ENX16 EASY 512IMP before gearbox or 56832 after gearbox

74RPM Robotzone motor

- Motor: 12V, 84RPM 1347.1oz-in HD Premium Planetary Gearmotor
- Gear ratio: 100:1
- Encoder pulses: 12 counts per turn before gearbox or 1200 after gearbox

Cytron planetary gear motor

- Motor: 49:1 Planetary DC Geared Motor 42mm
- Gear ratio: 49:1
- Encoder pulses: 5 counts per turn before gearbox or 245 after gearbox

104RPM Robotzone motor

- Motor: 12V, 118RPM 958.2oz-in HD Premium Planetary Gearmotor
- Gear ratio: 71:1
- Encoder pulses: 12 counts per turn before gearbox or 852 after gearbox

Bibliography

(2017), *Automotive fully integrated H-bridge motor driver*, st.com.

Amsters, R. and P. Slaets (2020), Turtlebot 3 as a Robotics Education Platform, in *Robotics in Education*, Eds. M. Merdan, W. Lepuschitz, G. Koppensteiner, R. Balogh and D. Obdržálek, Springer International Publishing, Cham, pp. 170–181, ISBN 978-3-030-26945-6.

Ben-Ari, M. and F. Mondada (2018), *Robotic Motion and Odometry*, pp. 63–93, ISBN 978-3-319-62532-4, doi:10.1007/978-3-319-62533-1_5.

Boikon (2022), Boikon – Shaping Technological Innovations.
https://www.boikon.com/

City, S. (2022a), *118 RPM HD Premium Planetary Gear Motor*, Servo city.
https://www.servocity.com/content/downloads/118_rpm_hd_precision_planetary_gearmotor_-_specifications_1.pdf

City, S. (2022b), *NE12 Magnetic Encoder Use Parameter*, Servocity.
https://www.servocity.com/content/downloads/ne12_-_use_parameter.pdf

Dexter-Industries (2022), GOPIGO3 is a Raspberry Pi robot car for learning coding.
https://www.dexterindustries.com/gopigo3/

Foundation, R. P. (2020), What is real-time computing?
https://www.futurelearn.com/info/courses/embedded-systems/0/steps/64783

Goodwin, D. (2020), Different types of motor encoders commonly used in automation - technical articles.
https://control.com/technical-articles/different-types-of-motor-encoders-Commonly-used-in-automation/

Greeling, J. (2021), Power consumption benchmarks.
https://www.pidramble.com/wiki/benchmarks/power-consumption

iRobot (2022), Create 3 robot.
https://edu.irobot.com/what-we-offer/create3

Maljaars, P. (2008), R2-G2P.
https://wiki.ram.eemcs.utwente.nl/index.php/R2-G2P

Maxon (2022), Maxon Formula handbook.
https://online.flippingbook.com/view/72734/8/

Prasaath, H. K. (2017), Choosing Motors for Robots.
https://www.engineersgarage.com/choosing-motor-for-robots/

Smoot, J. (2019), Capacitive, magnetic, and optical encoders – comparing the technologies.
https://www.cuidevices.com/blog/capacitive-magnetic-and-optical-encoders-comparing-the-technologies

Turtlebot (2022), Turtlebot 3.
https://www.turtlebot.com/turtlebot3/

Vinkenvleugel, J. (2022), Designing an embedded software architecture for a mobile education robot with real-time control on a Raspberry Pi 4 with FPGA-based I/O.