SPRO3 report

Botond Bencsik
Casper Hvide Bjerre Simenel
Felix Leo Hoch
Henrik Maarten Bongers
Laura Barney
Arthur Kibalama

08/09/2023

Contents

1 Introduction

Throughout history a certain industrial trend has become the most pronounced it has ever been, automation. Any and all tasks that could be automated in a financially viable way have seen at least one instance of it happening. This is due to the numerous benefits that an autonomous system can provide. In a similar vein our team also aimed at contributing to this phenomenon within the scope of the third semester project. The aim of the project was to make an autonomous intelligent vehicle.

Project work started with exploring the possible fields where such a vehicle could find sensible applications. A mindmap was used to order and keep track of various ideas. Some of these include a food and beverage delivery system both in a restaurant and home settings and some were related to various

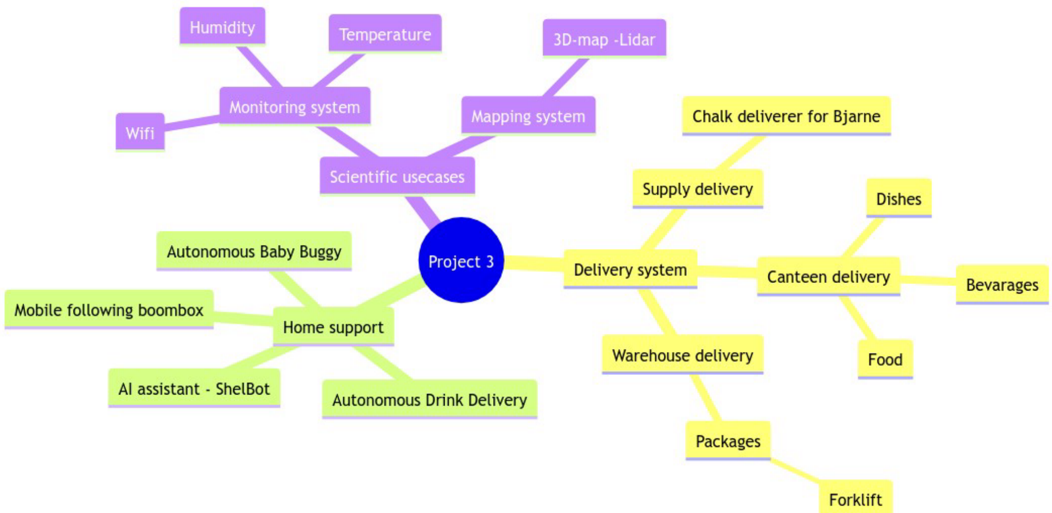


Figure 1: Mind-map of project ideas

Throughout the idea phase, the aspect of presenting the idea at the tech expo at the end of the semester was important. The audience should be able to grasp the concept of the project and understand how it could be beneficial in everyday life.

2 Problem Formulation

In the third semester our professors assigned us a project where the focus was to be the importance of sensors and actuators with autonomous vehicles being one example. Following this lead the first topic that came to mind was logistics, more precisely forklifts. To investigate the applicability of such devices in the aforementioned field our team members conducted market research starting from two separate perspectives, one being more industrial and the other more social.

2.1 more like a problem formulation???

2.1.1 Forklift related accidents and injuries

Forklift related accidents are a common occurrence, both with solid stationery objects and moving pedestrians. According to data from [1][there a around 34,000 injuries from forklifts in the US every year]. This results in a great number of lost work days and extra strain and pressure to make up the deficit.

Out of 143 incidents, where the collision happened with a solid object 75 of them are with stationery objects. And 53 of them are collisions between forklifts. Both of these types of incidents are related to human errors, like not paying attention to the surroundings when operating forklifts.

Out of 322 incidents involving pedestrians around 50% of them have been caused by a forklift striking a pedestrian by accident.

The safety of forklifts can be greatly increased with the use for sensors that enable the forklifts to avoid obstacles on their own, even with the human error factored in. This is similar to how safety in cars has been greatly improved in recent years.

2.1.2 Is there a market for self-driving forklifts?

Most companies are always competing to maximize the quality of the product while keeping the cost as low as possible. Many companies choose to outsource manufacturing to countries where, due to the lower average salary, they could further the aforementioned goal. This could achieve lower manufacturing prices but sometimes the quality suffered in turn. After learning this some companies choose to move manufacturing back to their origin. To stay competitive despite the higher costs, automating tasks can be a lucrative investment. For example having warehouses that can work 24 hours a day can greatly reduce certain costs. All without compromising the quality of the products.

2.1.3 Real-life applications

A research paper from a japanese university revealed that [3, in countries with aging demographics autonomous vehicles such as an autonomous forklift are required to lighten the burden on manpower and manual labor]. The authors of this paper detailed an autonomous pallet handling system for forklifts, which is able to handle pallets used for harvesting vegetables with no further human input in an outdoors environment. From a paper about aging populations it is stated that [2, 11% of the world is over 60 years of age and this ratio is expected to rise up to 22% in 2050].

2.1.4 Conclusion

These four papers clearly indicate that there is indeed a market and an application of highly automated processes regarding logistics and forklifts as well. This is how our group came to the conclusion that an automated forklift would solve a relevant problem all the while fitting in the frame outlined by our professors.

2.2 Similar products

Similar products are already widely available on the market and customers can decide based on their requirements without having to focus on only one ”innovative” product available.

Hyster Robotics provides autonomous forklifts up until stage 3. features are

1. Running on programmed path or manual mode
2. self locating and navigating
3. does not require laser reflectors, guide magnets or wires
4. communicates with WMS warehouse management systems

Otto Motors Lifter is one of their products next to the mobile robots featuring

1. Intelligent Pallet detection
2. autonomous lane clearing
3. in house Fleet Manager software
4. continuos remapping of area adapting to dynamic workplace

Toyota has the Automated Workhorse which can only lift pallets up for moving, but can lift up to 8tons which is much more than others in this sector.

Other companies selling autonomous forklifts are Multiway,Vecna, Cronw and more are there to find which gives a statement for the interest in this sector.

Company	MaxLoad kg	sensors	speed m/s	accuracy m	turning radius m
Hyster	cell5	cell6			
Otto Motors	1200	5x Intel RealSense Cameras 3x SICK Microscan3 (360°FOV)	1.5	0.05	1.9
Toyota	8000		1.2		0.1

Table 1: Comparison of similar products

2.3 Outlined requirements:

This semester, the emphasis is on sensor and actuator technology for mechatronic products, with a specific focus on applying these concepts in the prototype of an autonomous vehicle.The semester supervisors have outlined specific requirements that must be incorporated into the project.

1. Working prototype,tested and documented
2. Risk analysis(small part, 1-2 pages)
3. User interaction / requirement study
4. Hardware
   (a) 2 analog sensor
   (b) 1 Analog filter
   (c) Motor control / driver, can be a module
   (d) At least one small designed custom PCB
   (e) Basic Mechanics
5. Functionality
   (a) Object avoidance
   (b) Self driving
   (c) No machine learning
6. Budget of maximum 1000 DKK

2.3.1 Safety basics:

We have to implement the base of a Risk-analysis based on IEC14971 Standard for medical devices. This has to include functional risk and operator/person safety risk. Subjects can be Hazard, Likelyhood of occurrence and it´s Severity.

2.4 Technical requirements

In order to select certain parts for the project some technical requirements have to be estimated. These include, but are not limited to the mass of the payload, the mass of the vehicle and the desired operating speed of the forklift.

2.4.1 Payload

The group agreed that the payload, for modeling purposes, will be cans of soda.

1. Forklift can lift a pallet of 4 european standard sized cans. Individual can dimensions: Width: $66.1 \cdot 10^{-3}$ m; Height: $115.2 \cdot 10^{-3}$ m; Volume: 0.33 L.

2. Forklift can lift 3.5 kg of mass. This includes 8 cans and a pallet.
3. Dimensions of the pallet should be $150 \cdot 10^{-3}$ m by $150 \cdot 10^{-3}$ m.
4. Payload can be lifted at least $150 \cdot 10^{-3}$ m high.
5. Two pallets can be stacked on top of each other.
6. Lifting speed reaches $10 \cdot 10^{-3} \frac{m}{s}$.

2.4.2 Movement

1. Vehicle can travel at $50 * 10^{-3} \frac{m}{s}$.
2. Vehicle can turn at least $18 \frac{\deg}{s}$.
3. This would be about how accurately the vehicle can follow the path but there is no solution to navigation yet so this could be meaningless with some movement types
4. Instead of the previous one we could have something like this: The vehicle can deviate less than $100 \cdot 10^{-3}$ m left or right every meter traveled.
5. Vehicle is able pick up a pallet by inserting the forks into the pallet.

2.4.3 Mass of the vehicle

1. Vehicle should weigh less than 10 kg including the payload.

2.5 Company Engagement

In our pursuit of innovation, we engaged in meaningful conversations with potential businesses, seeking feedback on our project. Companies where forklifts are operated in warehouses with high levels of food traffic were of particular interest.

The local department of the international chain Bauhaus was contacted. What is particularly interesting about this business is that its warehouses are integrated into the shop, allowing customers to walk around the same space. Even though Bauhaus warehouse manager (Martin Hansen)? could see the potential in an autonomous forklift, he couldn't imagine them driving around in his warehouse in its current state. With the current sensor capabilities, it wouldn't live up to Bauhaus's strict requirements to ensure customer safety. He thinks that the project would be more suited for distribution warehouses, where only trained personnel would be on the ground floor. In conclusion, the interview provided insights into the project's target market.

3 Management

In response to the challenges outlined in the problem formulation, our approach to project management gained significant importance. Delving into the details of the requirements, we engaged in comprehensive discussions to formulate a robust plan. This section provides a detailed exploration of the methodologies employed to effectively manage project timelines, ensuring a strategic and well-coordinated effort towards successful project completion.

3.1 Time management

We aimed to include everyone in all areas of the project deployment. This meant we insured that everyone got the opportunity to be involved, with a task fitting to there knowledge level.

We began with creating a general time plan for the whole semester project progress.

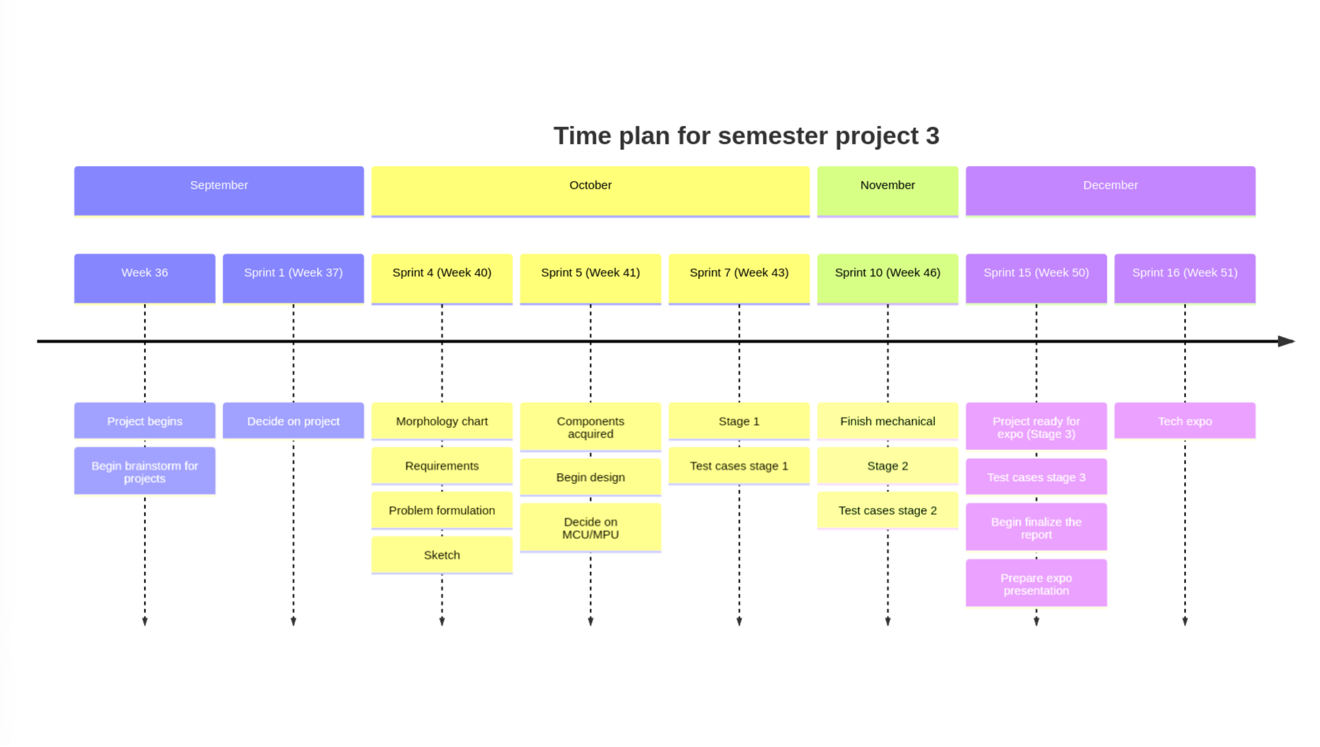


Figure 2: Initial time plan

3.2 Task management

For the semester period the Scrum agile management framework was adopted and customized some parts of it to fit our project and teams needs. We chose specifically this model due to the agile planing it introduces, and gradual learning curve. We defined a sprint to be 7 days, from friday to friday with an estimate of 8 hours of workload per team member. We categorized each task according to the time estimated to complete it using these tags:

- Sønderborg - 0.5 hours
- Keil - 1 hours
- Valencia - 2 hours
- Budapest - 4 hours
- Hamburg - 6 hours

We chose our home cities and sorted them by size to make it more simple to use in conversations and to give a better visual idea of the task size. We decided that the time limited on a task should be 6 hours. If a task would require more time then it should be split into multiple tasks. This was to ensure that it was possible to see if the task was possible to achieve before investing more time into it.

Throughout the semester period stand-up meeting according to the scrum model where conducted every Wednesday. This was to touch base and catch any potential problems. The meeting was written down

Week 49 (Sprint 10)

Name	What did I work on?	What am I working on?	What issues are blocking me?
Henrik	consulted teacher about the sensors: wheatstone bridge, setup, went to the labs	try to implement it, constant source, LT spice, test weight sensor	
Laura	model of the IR sensor, went to the lab with Henrik	develop further the amplifier	
Boti	writing arduino code for the IR sensor:analog sensor	test to differentiate different materials for the sensor	no printers available
Felix	report management task, physical assembly of the mast, cut the guide rodes	mast assembly	printers
Arthur	researching how to build the pcb of the drivers	design motor pcb	
Casper	created test cases	continue with the same task	time problems

Table 2: Sample of a weekly stand-up meeting

During the project period, we observed that the Scrum model functioned effectively when the workload was manageable. However, given the intensiveness of the last phase of the project, the workload for upcoming tasks was anticipated to be too substantial for Scrum to provide optimal benefits. As a result, we decided to retain the Kanban board for task management. Instead of having one person create and assign tasks, each individual team member was empowered to take ownership of task creation and assignment.

3.3 Stage division

Why divide the project up into multiple smaller stages? It means that there will be multiple deadlines, this helps keep the project on track. Keeping the project on track in the early stages contributes to the final stages where more issues may occur, this mens that there is always a working earlier version to showcase if it becomes necessary. It is also earlier for both the customer and the development team to see how the project will turn out and if there are any design mistakes that have to be changed before the final version.

It is impotent for this method to work, that each stage is clearly defined. This ensures that all features for each stage are implemented.

The below list contests all the requirement of functionalities for each stage. It is not a implementation list.

Stage 1: Basic movement

- The forklift should move unrestricted in any direction.
- The forklift should be powered by a battery.

Stage 2: Line following

- The forklift movement should be restricted to a pre-defined path.
- The path should be a line on the floor.
- The forklift should stop moving if there is a obstacle in its path.
- The forklift platform is solid with no risk of electronic or mechanical parts getting loses.

Stage 3: Pallet placement

- The forks on the forklift should be able to move up and down.

- The forklift should be able to pickup a pallet by itself.
- The forklift should be able to place a pallet by itself.
- The pickup and placement should be in predefined locations.

The functionalities are not restricted to a specific stage. It is fine that if a functionality can be implemented before the begin of its related stage. The only requirement is that the required functionalities for the current stage is completed before the deadline of the stage.

3.4 Individual evaluation

A short evaluation of the project from each member in their own opinion. This is used to improve upcoming projects, to prevent making the same mistakes again.

Henrik: Write here

Laura: Write here

Boti: Write here

Felix: Write here

Arthur: Write here

Casper: Write here

4 Consideration

To give a good overview over what option there a available for obtaining specified functionalities, a mind map was created.

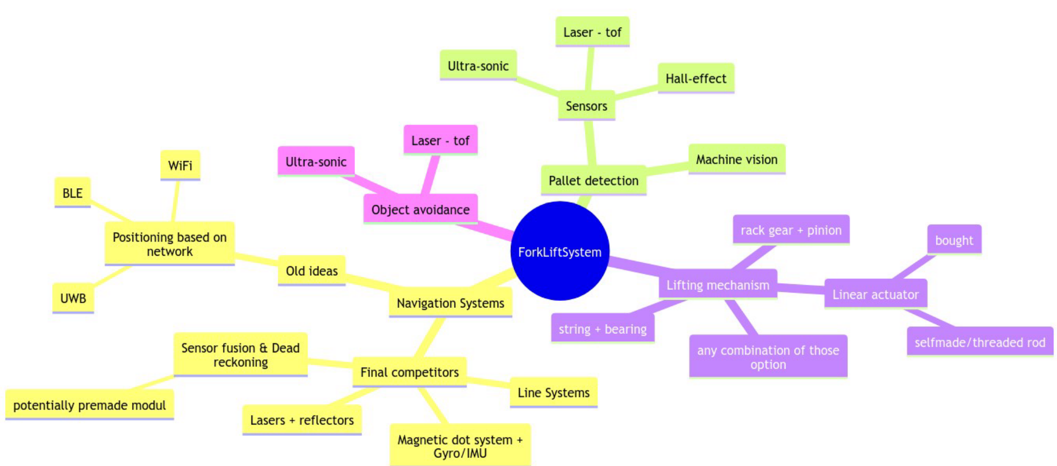


Figure 3: Options to consider

The mind map gave a good inside what topics to research further into before making any final chooses.

4.1 Forklifts

When looking into forklifts there are different types, that are task dependent. Some of those are the pallet jack or Low Lift Truck, Stacker and the most common Counterbalanced forklift.

The basic parts of the Counterbalanced forklift as seen in the picture.

1. fork
2. mast

Figure 4: Counterbalanced Forklift

3. drive and steering wheels
4. counterweight
5. motor compartment

subsection

4.1.1 Functionality

For the functionality of forklifts there are two different subjects that concern us. For one the drive, which should make it possible to rotate 360° around the center axle and the lifting of the fork it self. The amount of lifting is dependent on the type of forklift. Therefor the task of a low lift truck is lifting of the ground for simple 2 dimensional movement and other types need to be able to lift up to multiple pallets or meters, where also the name stacker derives from.

Lifting The lifting mechanism works by actuating an actuator which is and hydraulic cylinder in this case. A chain that is connected between an anchor, roller and the fork lifts the fork and by making use of the leverage of this system the forklift can lift greater weights than when connecting the fork directly to the actuator.

Driving On a forklift as seen in [4], the front wheels are connected to the drive while the back wheels are able to rotate 360° and only act for steering.

4.2 Lift solutions

When choosing our solution for the lifting of the fork it is important to specify the mast type before hand or decide on designing a modular solution, where different mast´s can be mounted depending on the request of the customers.

Those different mast types are called Simplex, Duplex, Triplex and Quadriplex which stand for the stage of the mast assembly and in essence for the maximum height it can operate at.

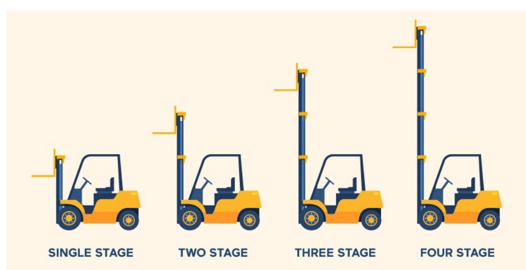


Figure 5: Stages

For each stage after the Simplex the freelift is also of interest, as states at which point the mast will extend. Having full freelift allows the forklift to lift the fork until the top off a Simplex stage without extending the mast, which makes the forklift useable in areas where not much overhead clearence is given, for example when emptying a truck.

4.2.1 Real lifting systems

Taking a closer look at the forklifts on the market, each lifting system is working with hydraulics. Reasons for this are:

1. Ability to switch forks easily(clamping, rotating, extra long and changing width)
2. Safety(Forks will never lower without permission) and no moving parts
3. Easy to maintenance
4. high power output
5. precise control

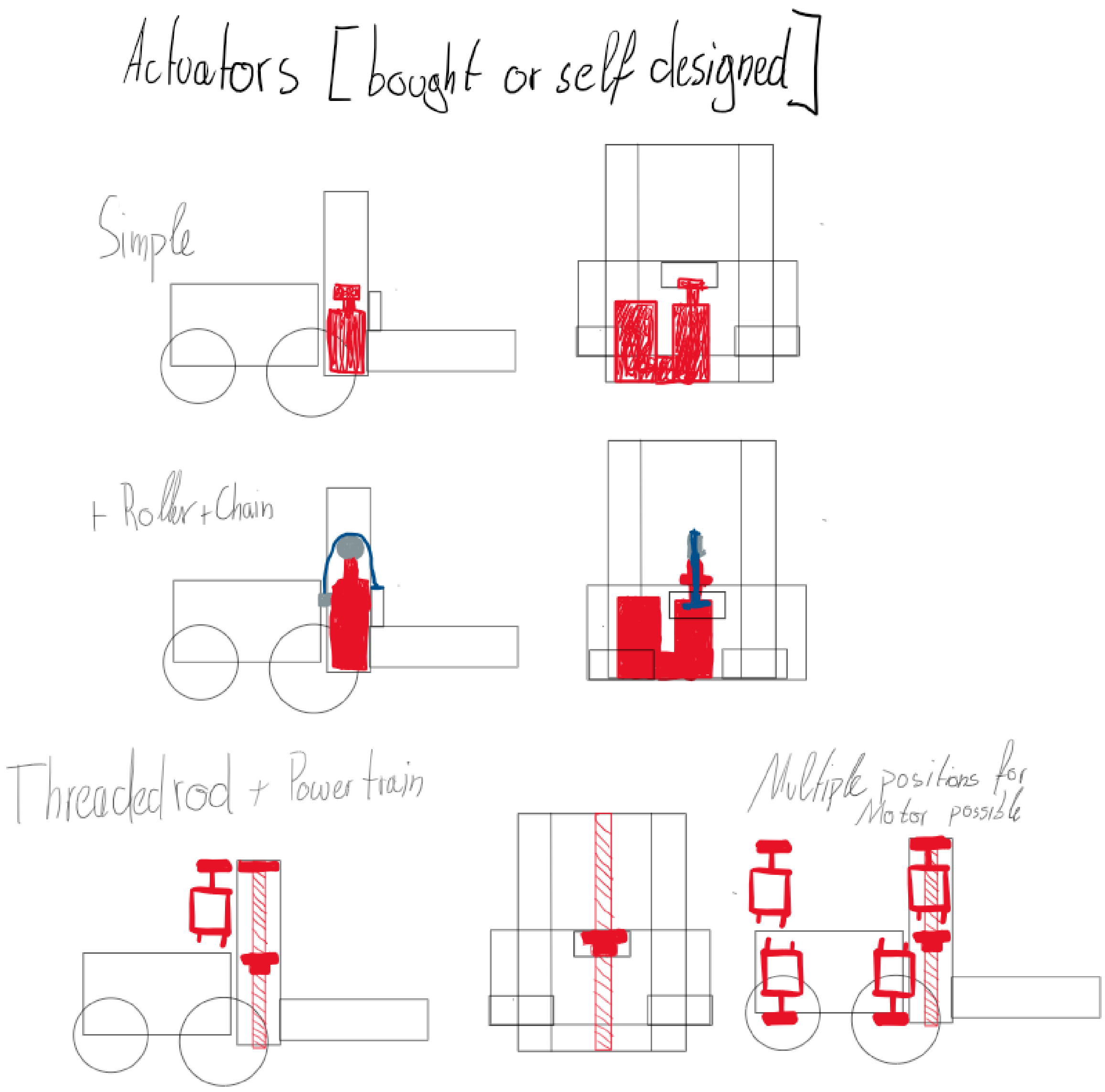
For lifting a hydraulic piston pushes a roller bearing up with a chain connected to an anchor on one side, rolling over it and connected with the fork on the other side. This creates a 2to1 lifting ratio where, 1meter lifted piston lifts the fork up 2meters.

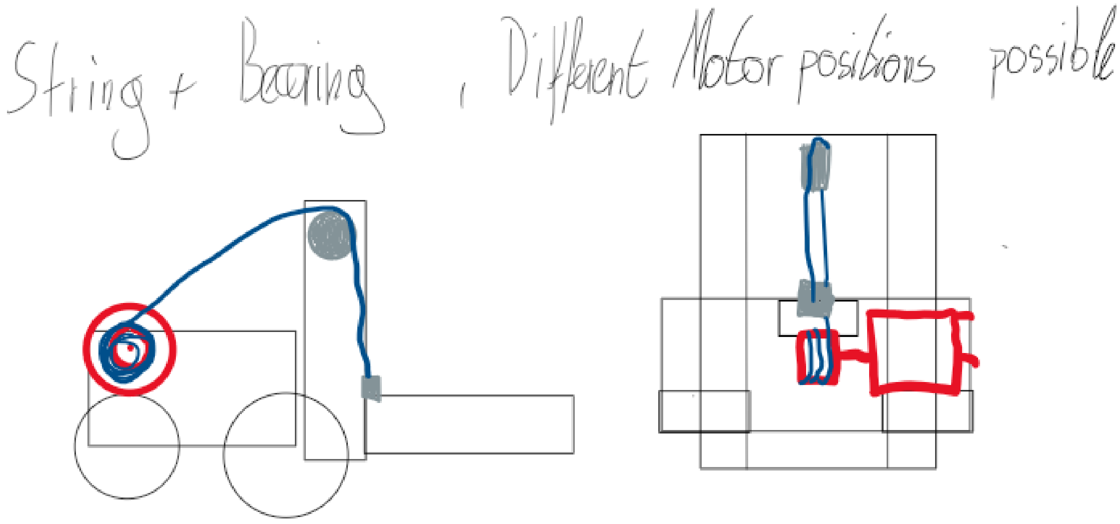
4.2.2 Lifting solutions

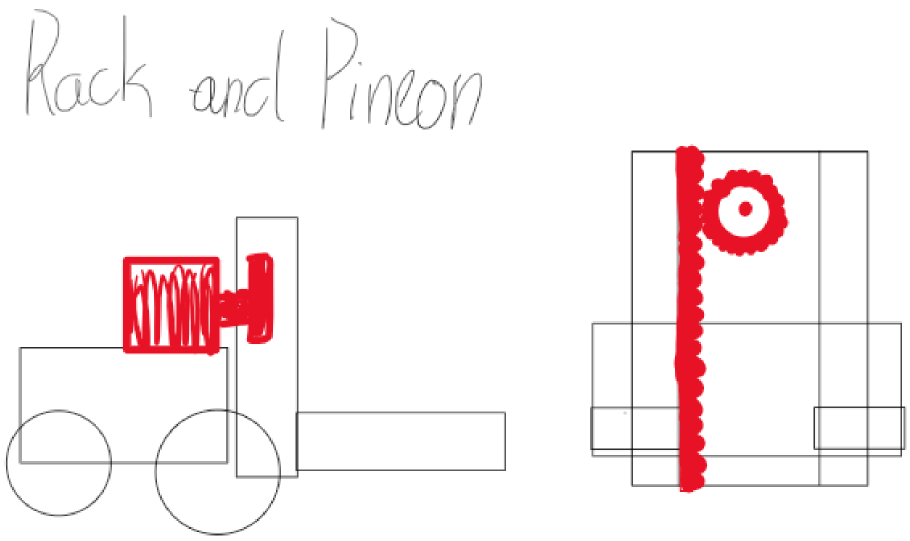
Because of the high cost, low weight of load we will carry and not having the focus on a acurate downscalled forklift we won´t be able to use a hydraulic system for lifting. Using a roller-chain should be considered in each solution for better load distribution and less movement need due to the 2to1 ratio. Options can be:

1. Linear Actuator(Bought)
2. Linear Actuator(own design)(threaded rod)
3. string+bearing
4. rack gear and pinion
5. //any combination out of those for higher stages

As our focus in this project lays on sensors and autonomous driving a Simplex stage 1 forklift will be enough of a mechanical task and could be upgraded if there is time to fill. Solutions could look like:







4.2.3 Lifting Calculations

To add a first value of safety the choosen system should be able to lift at least 30% (to be defined by a standard later) more than the maximum weight specification of our forklift. In this way we account for variances and fatique of the components we are working with. We want to lift 4 beer cans per pallet and two pallets stacked which gives:

$$4 * 0.333 * 2 * 1.3 = 3.46kg$$

Including weight of pallets and fork into the 30% and rounding up would give a plausible 3.5 kg actual lifting force or $3.5 * 9.81 = 34.5N$ which our lifting solution should be able to to provide.

4.3 Processing

When considering what processing power to use in the project where embedded system is wanted, there are to main candidates the microcontroller or the microprocessor. There are different aspect of each processing power that can be useful depending on what the project requires.

There are some key difference between MCU and MPU:

- Lower Power Consumption (MCU)
- Cheaper to Manufacture (MCU)
- Better Suited for Highly Intensive Work (MPU)
- Possibility to Utilize Cores (MPU)

Both options are well-suited for running real-time systems where the timing of actions is essential.

For this particular project, the benefits of a Microcontroller Unit (MCU) do not play a large role. When comparing the power consumption of the Microprocessor Unit (MPU) next to the motors, the MPU uses minimal power in comparison. The forklift consists of expensive parts, and most customers won't mind the small increase in price for the extra functionality, especially when they are already investing a significant amount.

The critical aspect is the workload the processor can handle when running a real-time system. This means that no compromises in safety or functionality are required for the machine to operate effectively.

4.3.1 Choosing the right MPU

It was decide that the software should be designed with real-time system i mind, because of the important of action taking place at the right time. There are a wide variate of option available:

- STM32
- RaspberryPi
- ESP32
- BeagleBone

One of these options stands out—the ESP32. It is the most cost-effective MPU available on the list while still offering the required features for smooth real-time system operations. The ESP32 comes with a single processor with two cores, distinguishing itself from some of the more expensive options that have two processors with one core each. Although the two-processor approach allows for a smoother experience, it comes with increased programming complexity.

For the budget prototype, the ESP32 will do the job effectively. However, for a potential continuation of the prototype, it should be considered whether the STM32 might be a better option, given its potential advantages.

4.4 Morphology chart

Summarizing the considerations in a morphology chart to select the options to continue with.

Functions	**1**	**2**	**3**	**4**	**5**
Navigation system	Lines on the floor	GPS location + pallet detection	Preprogrammed map	Sensor fusion (IMU + GPS)	Bluetooth RTLS
Micro controller / processor	Arduino nano	STM32xx	RaspberryPi	E-ESP32	BeagleBone
Pallet recognition	LoadCells	Hall Effect	buttons	Ultrasonic	photoelectric
Coding standard	Option 1	Option 2	Option 3	Option 4	Option 5
Mast Stage	Simplex	Duplex	Triplex	Quadriplex	-
Lift solution	Linear Actuator	L.A + RollerChain	Threaded Rod + linear bearings	String&Bearing	Rack&Pineon
Lifting sensing	End stops Top&Down	Encoder	Stepper Motors	LoadCell	-
Motors	Stepper on Mast	DC-Motors Movement	-	-	-
Sensors Fork	LoadCell	-	-	-	-
Sensors Body	Ultrasonic Distance HC-SR04	Infrared Sensors	LiDAR	Optical Sensors	-
Body design Elements	LCD	Tank tracks	Perry the Forklift	Googly Eyes	-

Table 3: Morphology chart

* Green cells indicates chosen option.

5 Electronics

One of the main focuses of this third semester project is analog electronics. Electronics bring the software and the mechanical part together. The software needs feedback from the analog world. It needs to know how heavy the load is and whether there is an obstacle in front. Furthermore, all the smart digital electronics need to be powered. But when powering everything from a battery the current state of the battery has to be evaluated. All these tasks are necessary to guarantee safe operations.

The tasks include the following:

- Project requirements
- Interfacing the analog load cells
    - Loadcell PCB Design
- Power Supply and monitoring
-
    - VeroBoard Design
- Motor Drivers
    - Motor Driver PCB Design

5.1 Fullfilling the project requirements

The project tasks given this semester include the usage of at least two analog sensors and at least one analog filter. In this project there are two arrays of IR-sensors - each counting 8 individual sensors with analog output - and an self-designed circuit fusing the output of 4 analog loadcells and amplifying it up. The following subsection will not only elaborate on the decision making behind them, but also on the research and the working of the final product.

5.2 Research and Evaluation for Loadcell-Interfacing

5.3 Choice of loadcells

One of the functionalities of the forklift is measuring the weight lifted. This serves the purpose of ensuring safe operations not only for customers, but also preventing the forklift from tilting or damaging itself in one way or the other.

In accordance to the set requirements, the forklift should at least lift 3.5kg. Hence, the loadcells should be able to withstand this weight. Next to buying new loadcells, one possible option is to take the loadcells out of a kitchen weight rated for that weight. This is beneficial in serveral aspects. Not only does it give a insight into reverse engineering, but also provides beneficial insight in classifying unknown sensor characteristics. Furthermore, it does not stress the budget.

Insert picture of loadcells here.

A kitchen scale rated for max. 5kg was bought. The four loadcells of this scale were taken to the lab and their behavior was evaluated.

Conclusion: The loadcells suit the application. They can not only withstand the required stress-requirements, but also work reliable. Moreover, they fulfill the space requirements for the "small-scale" fork. Furthermore, they bring a considerable learning effect with them.

The next engineering challenge is interfacing these loadcells. These loadcells include two variable resistances. One of them increases with stress the other one decreases with stress.

Interfacing bears serveral challenges:

- Output of interface must be compatible with ADC range of microcontroller.
- Current through loadcells must be limited.
- Output needs to be amplified.

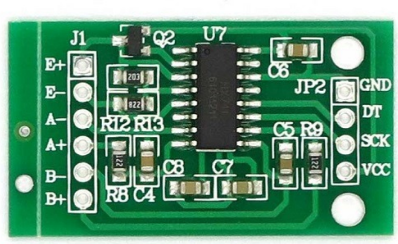
5.4 Available Options

In this section the available options to overcome these challenges are discussed.

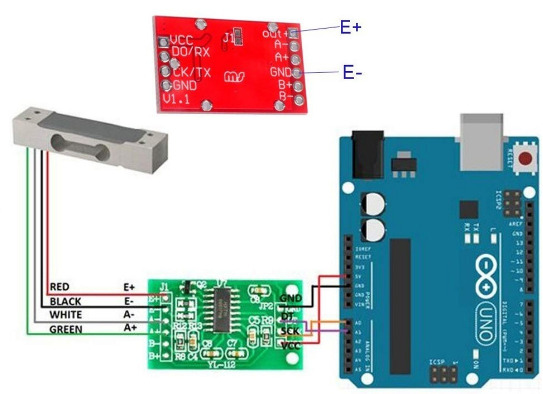
- HX711 Interface Module and load-cell
- Wheatstone bridge - in different configuration
  - With constant current source
  - Without constant current source

5.4.1 HX711 Interface Module and load-cell

A Load Cell serves as a sensor that transforms applied force, encompassing pressure, rotational force, compression, or tension, into quantifiable electrical signals. The load cell generates an output in millivolt range; therefore, it is essential to magnify this signal into a higher-level amplitude and subsequently convert it into a digital format for further processing. To accomplish this task, the HX711 interface module could be used. This was recommended by a professor. This module serves to amplify the load cell's low-voltage output and transmit it to the microcontroller for weight calculation. The illustration below depicts the HX711 interface module.



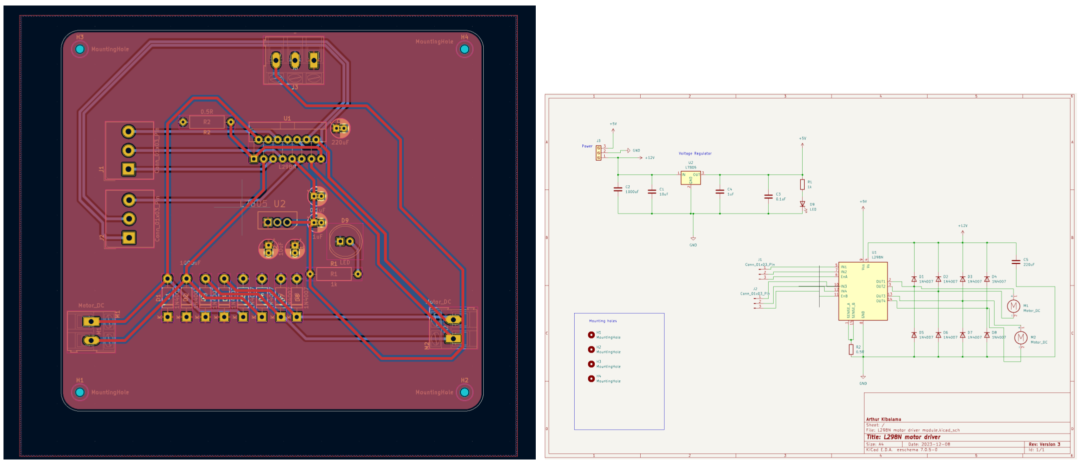
(a) HX711 Interface Module



(b) HX711 Interface Module and Load cell using Arduino

5.5 Motordriver PCB

This is a 2 layer PCB for the Motordriver.



(a) Motordriver PCB (b) L298n Motordriver Schematic

How I designed the PCB

Decoupling Capacitors

In the strictest definition, a decoupling capacitor isn't a distinct component per se, instead it characterizes a capacitor's role within an electronic circuit. Its primary function is to enhance stability on the power supply plane by mitigating voltage fluctuations. In any design involving semiconductor ICs, the inclusion of decoupling capacitors is imperative. This is because the voltage supplied to the components deviates significantly from the ideal scenario. Unlike the theoretical perfectly steady line, real-world voltage readings exhibit fluctuations, even with a meticulously clean power supply. The decoupling capacitor operates as a reservoir, contributing to voltage stabilization through two key mechanisms. First, it absorbs excess charges when the voltage surpasses the rated value. Simultaneously, it releases stored charges when the voltage decreases, ensuring a consistently stable power supply. Typically, a combination of at least two capacitors with different values is employed to stabilize the voltage supplied to a component's VDD pin. A capacitor around 10 uF serves as a substantial buffer to smooth out low-frequency fluctuations, while a smaller capacitor of approximately 100 nF addresses high-frequency changes in voltage.

6 Mechanical

6.1 Manufacturing methods

The method by which components are manufactured has a significant influence on various aspects of the project development process. If the chosen manufacturing method for a prototype is expensive, it can lead to the project exceeding its budget. Similarly, if the manufacturing processes are too time-consuming, it can result in delays for the project.

6.1.1 3D prining

Employing 3D printing technology to manufacture components for prototype projects offers a many advantages. This cutting-edge manufacturing method allows for rapid and cost-effective iteration, enabling us to swiftly refine and test various design iterations. The flexibility of 3D printing allows the production of intricate and complex geometries that might be challenging or impossible with traditional manufacturing methods.

For manufacturing prototypes using 3D printing, the FDM method was chosen. In particular, the materials PLA and PETG were used in the manufacturing process. Both materials are non-toxic, meaning they are safe to use indoors. However, they do produce volatile organic compounds (VOCs) and microplastics in the air, so a well-ventilated workspace is recommended. Another great benefit is the low cost of the materials.PETG is particularly used where durability is important, but it is not used everywhere because it is more expensive than PLA. In most cases, PLA can do the job effectively.

The flexible material TPU was utilized to manufacture the outer layer of the wheels. TPU is known for its exceptional flexibility and high friction, similar to rubber. This makes it well-suited for the production of wheels and other flexible components.

6.1.2 Leaser cutting

3D printing is excellent for intricate designs with relatively small areas. However, as the printing area becomes larger, it can become expensive and time-consuming. In such cases, laser cutting emerges as a more favorable option. When using laser cutting, the design must be created in 2D. The thickness (the third dimension) depends on the chosen cutting material.

In the prototype, laser cutting has been employed in various areas, with one notable example being the pallet. Due to its large area and fewer intricate details, the pallet is an ideal component to manufacture using a laser cutter.

6.1.3 Other methods

In the initial phase, the base plate for the forklift was laser-cut. However, it was quickly discovered that the available materials were not durable enough to withstand the load introduced by the fully loaded pallet. The decision was made to manufacture the base plate using stainless steel, which was readily available on campus. This choice enhanced the durability and weight of the base, contributing to the counterbalance force for the forklift.

6.2 Part numbering

We introduced a part numbering system in order to keep a clean overview about which components we create, manufacture are currently working on. Applying this also allows for intuitive reopening of older versions and picking up the project from another college.

Part numbers begin with FL - indicating that they belong to our project (FL = Forklift). The forklift is split into sections. For example, base, fork and top. These are just examples and we can adapt them when the time comes and we start designing. So far, the number consists out of the following: FL-Xx, where Xx is being replaced by the starting letters of the subsection - for instance, Ba for base. This is then followed by D or M, which indicates whether it is a drawing or a model. Giving as a formula: FL-XxX. This is then followed by the version number -xx. **The numbering starts with 00**. Leading to the final and **general formula**:

- **FL-XxX-xx**

The general assembly is marked as **FL-AsX-xx**.

6.2.1 Examples

Part number:

- FL-BaM-00 - Project forklift - Base Model - version 0 (meaning original)
- FL-FoD-04 - Project forklift - Fork Drawing - version 4

6.3 Versions & Assembly System

6.4 Assembly

The assembly serves several purposes in the manufacturing process. One notable benefit is the ability to ensure that newly designed models fit into the assembly, allowing for a preliminary check before investing time in manufacturing the actual model. The time invested in the assembly ultimately translates into time saved during the redesign and re-manufacturing phases.

Another advantage is that the assembly provides a visualization of the final model. This can be used in conversations to prevent misunderstandings that can result in feature mistakes. Having a visual repre-

Figure 8: Figure of assembly

sentation facilitates clearer communication, reducing the likelihood of errors and ensuring that the design aligns with the intended specifications.

6.5 Lifting Development

7 Code

7.1 FreeRTOS

In the design of an autonomous vehicle, factors such as precise timing, responsiveness, and predictability are crucial for the software. If the software fails to process inputs from sensors quickly enough, it could lead to an accident. These critical aspects can be addressed by utilizing a real-time operating system (RTOS) like FreeRTOS. With RTOS task scheduling, essential actions such as halting movement when an obstacle blocks the way are executed within specified time constraints. This ensures that the action of stopping the vehicle is not obstructed by other tasks, thus preventing accidents.

7.1.1 The Espressif flavor

One of the reasons we chose the ESP32 MCU was to leverage its capabilities with FreeRTOS. The ESP32 uses a custom flavor of FreeRTOS made by Espressif. A key difference for this custom flavor is its support for Dual-Core processors. This means that tasks can be distributed across two cores, as opposed to the original one core support in FreeRTOS. Another advantage is the presence of a HAL, ensuring that MCU changes in the ESP ecosystem are durable without the need for significant code maneuvers.

Throughout this section some FreeRTOS native function are used. These functions include:

Tasks:

```
    // Task to be created.Pointer that will be used as the parameter for the task being
    void vTaskCode( void * pvParameters )
    {
        for( ;; )
        {
            // Task code goes here.
        }
    }

```

Listing 1: FreeRTOS task creation

There are a few things to notice that differ from a function in regular C language. Firstly, the function name has a prefix of 'v,' where 'v' stands for void, indicating that the function does not return anything. Another notable aspect is the use of a 'for' loop. In this context, a task is designed to run continuous operations, hence the inclusion of the forever loop. It's important to highlight that, unlike native C, the forever loop won't block other operations. The 'pv' prefix on the function parameter indicates that the return type is a pointer to void. This parameter is used to continuously pass values to the task throughout the program

Creating task:

```
    xTaskCreate( vTaskCode, "NAME", STACK_SIZE, &ucParameterToPass, tskIDLE_PRIORITY, &xHandle );

```

Listing 2: Creating a task

When creating a task, several parameters need to be fulfilled. These include:

- **vTestCode:** The task function to call, as seen in listings 1.
- **NAME:** A given name for the task..
- **STACK_SIZE:** The size of the task stack, specified in bytes.
- **ucParameterToPass:** A pointer to pass parameters to the task.
- **tskIDLE_PRIORITY:** The priority at which tasks are run.
- **xHandle:** The task handle by which the task can be referred to.

The 'x' prefix in the function name indicates that the function returns a value. In this case, it signifies the return of the handle, which is used to reference to the task later on.

7.1.2 Development enviroment

A familiar development environment is crucial for ensuring quality and productivity in coding. Learning the ins and outs of a new Integrated Development Environment (IDE) can be time-consuming. Therefore, it is a significant advantage that Espressif has integrated their ESP-IDF into the VS Code IDE. VS Code is an IDE familiar to everyone participating in the project, and its extensive array of tools further facilitates development.

7.2 Planning

Before delving into programming, it's advisable to initiate the planning phase for the program sequence. This involved creating flowcharts for each stage of the development process.

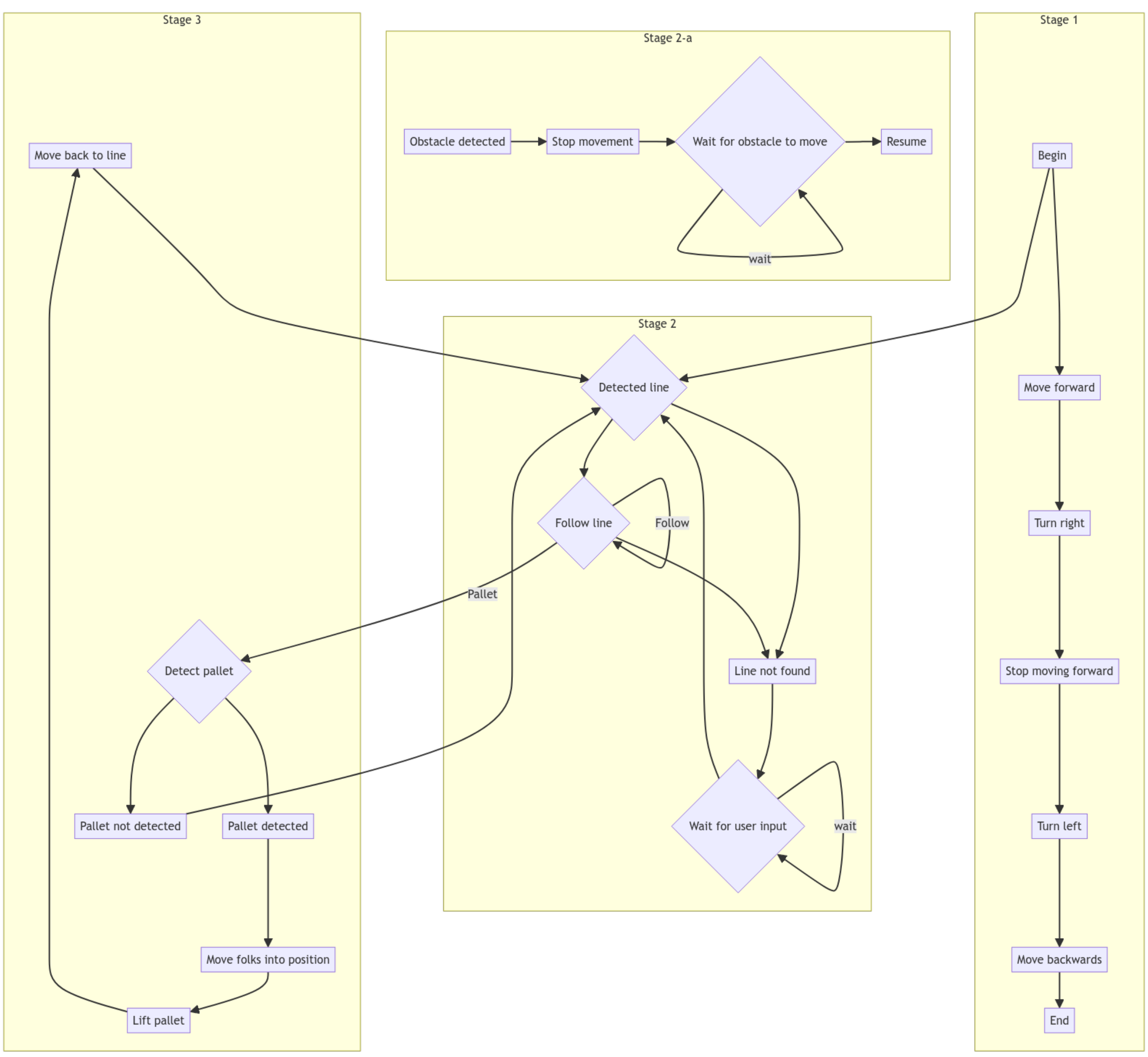


Figure 9: Three-Stage Flowchart

The flowchart assists the software developer in implementing the code that will execute the planned sequence. It is also essential for planning out test cases to ensure thorough testing of all software functionalities. Overall, the flowchart saves time in the long run by preventing the software developer from hesitating and waiting to determine how the implementation should be done.

7.2.1 Version control

Effective planning of the Git branch structure is crucial for software development projects. When employing Git as a collaboration tool, it is imperative to ensure that the team adheres to specific guidelines:

- **Protected Master Branch:** Only merge requests are permitted on the master branch.
- **Developer Branch Commits:** Commits should be made on the developer branch.
- **Independent Feature Development:** Create a new branch for independent features, branching from the developer branch.
- **Meaningful Commit Messages:** Commit messages should be both meaningful and concise.
- **Peer Review for Merge Requests:** All merge requests must undergo review by another team member.

These guidelines significantly contribute to the smooth running of the development process. Primarily, they prevent the occurrence of large and time-consuming merge requests. When followed correctly, these guidelines result in a branch structure resembling the illustration in Figure 10.

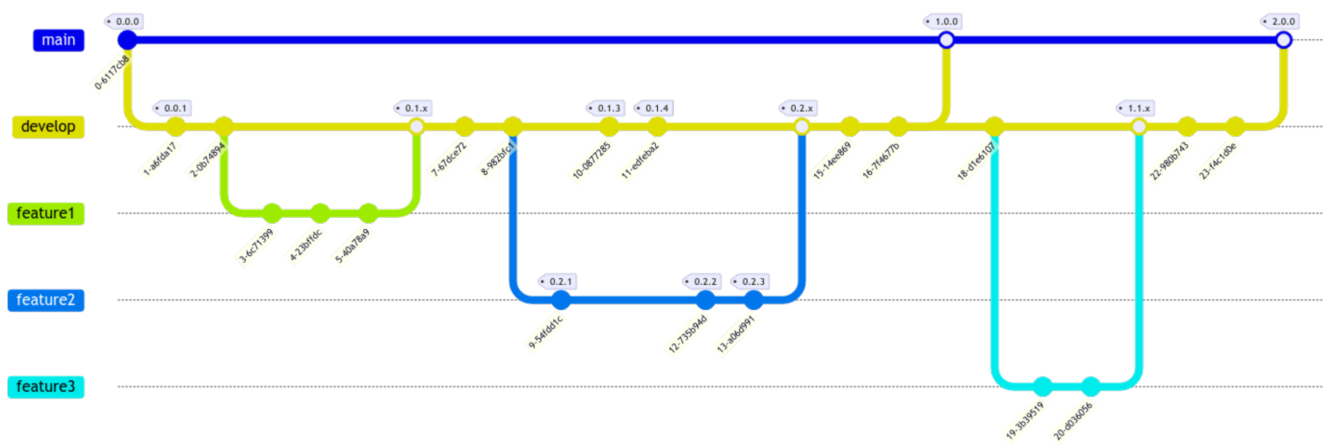


Figure 10: Git branch structure

7.3 ADC

When programming the ability to use a ADC (Analog to digital converter) to read a analog value on the microprocessor is of crucial impotents. In our project the ADC is use for reading values from our infrared and load sensors. On the ESP32 a total of 18 ADC channels are available with a config resolution options of 9,10,11 and 12-bits.

7.3.1 Configuring

It is impotent to ensure when using RTOS that functions are Thread safe. If not it can result in RTOS not being able to handle a task in the desired time frame. For this reason the libraries that are include in the ESP-IDF environment that are designed with thread safe in mind. The particular library used to facilitate ADC is called *esp_adc/adc_oneshot.h* that replaced the previous one in version 5.0.4 of ESP-IDF.

To configure a ADC unit:

```
        adc_oneshot_unit_handle_t adc1_handle;
        adc_oneshot_unit_init_cfg_t init_config1 = {
            .unit_id = ADC_UNIT_1,
            .ulp_mode = ADC_ULP_MODE_DISABLE,
        };
        ESP_ERROR_CHECK(adc_oneshot_new_unit(&init_config1, &adc1_handle));

```

Listing 3: Configuring ADC unit 1

The program utilize handles to reference the objected throughout the program. When *calling adc_oneshot_new_unit* a new instance is created with the specified configuration. In a similar way the channels are configured and created to the handle.

7.3.2 Reading

To read the raw value of the ADC unit 1 channel 0:

```
        adc_oneshot_read(adc1_handle, ADC1_0, &adc_value);

```

Listing 4: Readning ADC unit 1 channel 0

This function takes 3 augments: the handle itself, the desired channel to read from and where the output should be stored. This function can acquire the raw value from any ADC on the given unit as long as they are configured.

To calculate the voltage level:

$$V_{out} = D_{out} * \frac{V_{max}}{D_{max}} \tag{1}$$

Where:

D_{out}: ADC raw digital reading result

V_{max}: Max measurable analog input voltage

D_{max}:$2^{bitwidth}$

7.3.3 Implementation

The Analog-to-Digital Converter (ADC) is employed to convert analog signals from both the infrared and load cell sensors. The two infrared sensors collectively generate 16 analog outputs, while the load cell sensor contributes one. However, the specific ESP32 utilized in this project features only 12 available ADC pins on the breakout board. This poses a constraint on the number of outputs from the infrared sensors that can be accommodated.

In the context of the infrared sensors, the ADC value varies based on the reflective properties of the surface. To facilitate line-following, the program assesses the ADC values and executes movement actions based on the reflectivity of the surface for each sensor in the array.

7.4 PWM

8 Testing

In the dynamic landscape of project development, the validation of system functionalities stands as a critical phase, ensuring that each component operates seamlessly and meets the defined criteria. In the evaluation of system functionalities, a crucial aspect lies in clearly defining the anticipated behavior prior to the actual testing process. This foundational step serves as a guide, directing the systematic examination of each aspect. In this section, we delve into the methodologies employed during testing, underscoring the significance of establishing expected behavior as a fundamental precursor. Additionally, a comprehensive table of test cases is presented, providing a detailed overview of our testing approach and outcomes.

8.1 Testing Methodologies and Execution

Before initiating any test, a conscientious effort is invested in clearly defining the expected behavior of the system based on the specified requirements in the problem formulation. This not only serves as a guide for testers but also establishes a benchmark against which the actual outcomes can be measured.

The clarity of anticipated behavior lays the foundation for a comprehensive testing strategy, ensuring that each test case aligns with the project's objectives.

Our testing process follows carefully designed protocols, encompassing each test case with pre-defined pre-requirements, procedural steps, and expected behavior. The systematic execution hinges on fulfilling these requirements and adhering to outlined procedures. Success is contingent upon achieving the expected behavior, validating that the system operates according to established criteria. Conversely, deviations or procedural missteps result in an unsuccessful test, prompting a reevaluation of the system's functionality. This thorough approach ensures a comprehensive examination of the system's integrity and functionality.

8.2 Test cases

Stage 1

#	**Objective**	**Steps**	**Expected result**	**P/F**
1.1	Forward movement	1.Flash stage 1 code 2.Place forklift down on the floor 3.Press start	The forklift moves forward a arbitrary distance	
1.2	Backward movement	—”—	The forklift moves backward a arbitrary distance	
1.3	Forward right movement	—”—	The forklift moves forward and right a arbitrary distance and angle	
1.4	Forward left movement	—”—	The forklift moves forward and left a arbitrary distance and angle	
1.5	Stationary right turn	—”—	The forklift turns right a arbitrary angle	
1.6	Stationary left turn	—”—	The forklift turns left a arbitrary angle	
1.7	Read battery voltage	1.Measure the battery voltage with a multimeter 2.Flash stage 1 code 3. Read displayed voltage in display	The multimeter measurement and display value should be the same with a tolerance of 2%	
1.8	Battery voltage low	—”—	The forklift should not start when the start button is pressed. Instead it should prompt the user that the battery needs to be charged.	

Stage 2

#	**Objective**	**Steps**	**Expected result**	**P/F**
2.1	Movement following the line path	1.Flash stage 2 code 2.Place the forklift on line 3.Press start	The forklift should follow the line path when moving at a arbitrary speed.	
2.2	Detected obstacle stop	—”—4.Place obstacle on path	The forklift should stop before hitting the obstacle	
2.3	Detected obstacle start	1.Run test case 2.2 first. 2.Remove obstacle for the path	The forklift resumes it movement along the path	

Stage 3

#	**Objective**	**Steps**	**Expected result**	**P/F**
3.1	Move fork up	1.Flash stage 3 code 2.Place the forklift on line 3.Press start		
3.2	Move fork down	—”—		
3.3	Move fork to minimum	—”—		
3.4	Move fork to maximum	—”—		
3.5	Detected pallet	—”—	The forklift should detected and move into position of the pallet	
3.6	Move pallet up	—”—	The forklift should move the forks under the pallet and move it up a arbitrary distance	
3.7	Move with pallet	—”—	The forklift lift should move with the pallet to the destination point along the path	
3.7	Move pallet down		The forklift lift should move the pallet down to ground level	

9 Conclusion

10 Appendix

10.1 Acronyms

OS	Operating system
RTOS	Real-time operating system
MCU	Microcontroller
MPU	Microprocessor
ADC	Analog to digital converter
IDF	IoT Development Framework
PDB	Printed circuit board
IDE	Integrated development environment
PWM	Puls-width modulation
HAL	Hardware abstraction layer
VS code	Visual studio code
FL	Forklift
GPS	Global Positioning System
IMU	Inertial measurement unit
RTLS	Real-time locating system
STM	STMicroelectronics
L.A	Linear Actuator
LCD	Liquid-crystal display
IR	Infrared
FOV	Field of view
WMS	Warehouse management systems
BOM	Bill of materials

10.2 Glossary

FreeRTOS	A open source real-time operating system
base10	Decimal numeral system
base16	Hexadecimal numeral system
base2	Binary numeral system
c language	General purpose programming language
DevOps	A means for improving and shortening the systems development life cycle.
Flash memory	An electronic non-volatile computer memory storage medium
Gantt chart	A bar chart that illustrates a project schedule.
Github	A DevOps software package
git	Distributed version control system
Microcontroller	A small computer on a single VLSI IC chip.
Scrum	An agile project management framework
Kanban board	Visually depict work at various stages of a process
VS code	A open source code editor
LT spice	Analog electronic circuit simulator computer software
Kicat	Free software suite for electronic design automation
IEC1497	Application of risk management to medical devices is a voluntary standard for the application of risk management to medical devices.
STM32	A family of 32-bit microcontrolla family of 32-bit microcontroller integrated circuits by STMicroelectronics. er integrated circuits by STMicroelectronics.

References

[1] James W. Collins. "Injuries related to forklifts and other powered industrial vehicles in automobile manufacturing". In: *American journal of industrial medicine* (5 1999).

[2] Eleni Kanasi. "The aging population: demographics and the biology of aging". In: *Periodontology 2000* 1 (2000).

[3] Hiroyuki Onoyama Ryosuke Iinuma Yusuke Kojima. "Pallet Handling System with an Autonomous Forklift for Outdoor Fields". In: *Journal of Robotics and Mechatronics* 32 (5 2020).