Report sketch

# Introduction

This is a little sketch of mine about this semester project’s report. I was part of the software division, tasked with finding a solution about giving the drone eyes so to speak. The main part of it is about the drone being able to detect and read the text on car licence plates to check which car is needed to be parked or taken out, and a side part of it was about checking the car wheels to so that the lifting mechanism would be in place. Some additional parts I wanted to do were about the different functions of the drone, like sensors, motor drivers and the operating system of the drone with the use of ROS2.

# Licence plate reader

## Introduction

The license plate identification is a program that uses the main camera connected to the raspberry pi 5 that is used to capture, modify, then read the licence plates of cars. This feature is made to identify the car that is needed to moved/parked, without any outside interaction other than the request itself.

## Functions

There are four main functions that is needed to be performed for the identification to happen successfully. For almost all functions a library was used that was made to carry out these functions, with as little problems as possible

## Licence plate tracking

For detecting and tracking the licence plate a library called YOLOv3 (You Only Look Once v3) is used that was created to detect and identify different objects that enter the camera’s sight. It has many pre-trained functions that can be loaded to detect almost anything and has many variations depending on what it is needed for

## Frame modification

The library that takes care of capturing images and modifying them is called OpenCV. This library is made for the application of Computer Vision which is about getting images and helping the computer in understanding them, so they can be used for different uses.

## Reading from pictures

EasyOCR in a way a simple program that was made to read text from images. EasyOCR is usually used with picture modification libraries, because even though it can read text from untouched images, these libraries can convert the frame into an another one where the text is possibly easier to read making the text reading much more reliable.

## Compare licence numbers

The last step of this code is about comparing the read licence numbers with the one that was requested. Whenever the reading was successful with high enough confidence then that set of numbers and letters gets compared to an already saved or a given numbers and letters and if the comparison was successful the lifting process begins. This step can be made without any significant libraries.

## Explanation of the code

The code starts with loading all the libraries and its dependencies and initialises the system. Then a loop starts going that captures the frames of the camera and looks for any licence plates.

When a licence plate was detected, the frame modification begins with the picture turning grey and get thresholded where if a pixel’s scale is above a give threshold, it becomes black and it its below, it becomes white, resulting in a black and white picture.

The black and white picture, if done well the text is read from it and then compared to a known licence number. When the comparison results in a failure the program starts looking for another car, but if it was correct then the position correction and lifting procedures begins.

## Sources

The base of a code was downloaded from a public source which is owned by an engineer and content creator called Computer Vision Engineer: <https://github.com/computervisioneng/yolov3-from-opencv-object-detection>

YOLOv3 is an open source library out detecting object which can be trained to notice different objects: <https://pjreddie.com/darknet/yolo/>

OpenCV is an Open Computer Vision library about reading and modifying either images or a series of images: <https://github.com/opencv/opencv>

EasyOCR is an Optical Character Reader made to read text from images: https://github.com/JaidedAI/EasyOCR

## Conclusion

This program helps with the identification of the car that is needed to be lifted and parked/moved. Right know the code is in working order, but there is a limitation with the distance, where it can detect the licence plate from. Since the drone is operated with Raspberry pi 5 paired with ROS2 the licence plate reader was not able to be implemented because of some packages needing virtual environments making them unable to be implemented in this semester.

# Sensors and motor drivers

## Introduction

The main objective of the semester program is to make a robot that can move around, possibly autonomously and the to be able to detect and evade obstacles as it is doing its appointed job. The movement was done with four motors and two motor drives to have differential drive, so that the drone can turn around its centre point. The obstacle detection of the drone is made with the combine of a lidar sensor and ultrasonic sensors because during the lifting process the lidar has blind zones, for which the best substitutions are easy to implement proximity sensors. As an addition as this project is going to be a parking robot, additional motors were implemented with their own motor drivers for the lifting systems.

## Driving Motors

The program for the driving motors is done with Arduino coding on an Arduino nano v3. The speed of the motor is determined by the raspberry pi 5 with the help of the lidar to see how far the drone can go and changing its speed values accordingly. This value would then be translated into a code that the serial port can send to the Arduino with the direction the drone wants to go so that motor will be operated as needed. The Arduino with the information it got would send direction and PWM signals to the motor drivers making the drone start moving.

## Ultrasonic sensor and lifting motor

The ultrasonic sensors and the lifting motors for the prototype a raspberry pi 3 is implemented because of the ease of use with the core of the drone, the raspberry pi 5. Installing ROS2 on the raspberry 3 meant that there is a LAN connection between the two PIs meaning easy transfer of information. With the usage of a python package GPIOzero the raspberry pi 3 can be used to operate the sensor and the lifting motor.

# Explanation of the code

This part of the code is made up of multiple codes because of the different hardware used they are either made with python or Arduino code for the Arduino nano.

## Movement motors

The motors are handled with the combination of the raspberry pi 5 and the Arduino nano. It starts with reading the distance the lidar reads in front of the drone and the implemented controller for the prototype, then turning that number into another one that can be used for the motor’s pwm. With the speed acquired, the direction is needed which can be obtained with reading the directions of the controller and with the speed and the direction connected, the command is sent to the Arduino nano. When a command is received by the Arduino it changes the speed and direction accordingly and stands by for the new order.

## Ultrasonic sensor and lifting motor

The ultrasonic sensor and lifting motor get are handled by the raspberry pi 3 with the instructions from the raspberry pi 5. Since the lifting system can only be used when the drone has completely stopped for lifting or when stopped for parking and for that the core sends a signal to the lifting PI. When the lifting system gets its green light, the ultrasonic sensor starts detecting so that the procedure stops when somebody or something is too close to it. For lifting motor, the direction is needed to where the motor should turn and when that is acquired from the raspberry pi 5 and when that has happened, the motor starts to rotate to the needed direction with a given speed lifting the car from its wheels and parking it.

## Sources

For the operating system of the robot, so that is runs as an automatic web system ROS2 was used: <https://docs.ros.org/en/jazzy/Tutorials.html>

To get access to the GPIO pins the GPIOzero package was used which has many tools and easy implementation: <https://gpiozero.readthedocs.io/en/latest/>

## Conclusion

The program for the movements is something that has a good shape, but it is one 100% operational because of 1 or 2 thing that was not resolved yet, which will be in the future. The lifting motor driver and the ultrasonic sensor codes are done and tested, but because of some limitations the lifting system was not produced and these programs we not used.