Udruga mehatroničara
2025 robot mechanical and electrical description with a software flow chart

Authors: Petra Horvat, Jan Hajsok, Josip Dujić

Mechanical Description

UM Robot is constructed in three levels or platforms. Each platform is made of 5 mm aluminium sheet which is cut in precise measures with laser cutting technology. Threaded and through holes on each platform define positions for walls, bridges and all mounting points including dribbling and kicking mechanism constituents. Platforms are connected with 3D printed bridges and walls with features that ease assembly. Each Bridge is connected to two out of three platforms.

Motion

Robot motion is achieved by 4 brushless DC Flysky motors which are connected to omni wheels via gear transmission with reduction of around 5. Omni Wheels are produced in house, and are derived in three rows with 8 polyurethane wheels within one row. Gear transmission is covered with 3D printed housing.

Every power module (Consisting of omni wheel, shaft, gear transmission and motor with mounts) is equipped with two dampers with springs which are connected through rods between the second and third platform of the robot providing smooth motion of the robot and robust behaviour in high acceleration rates.

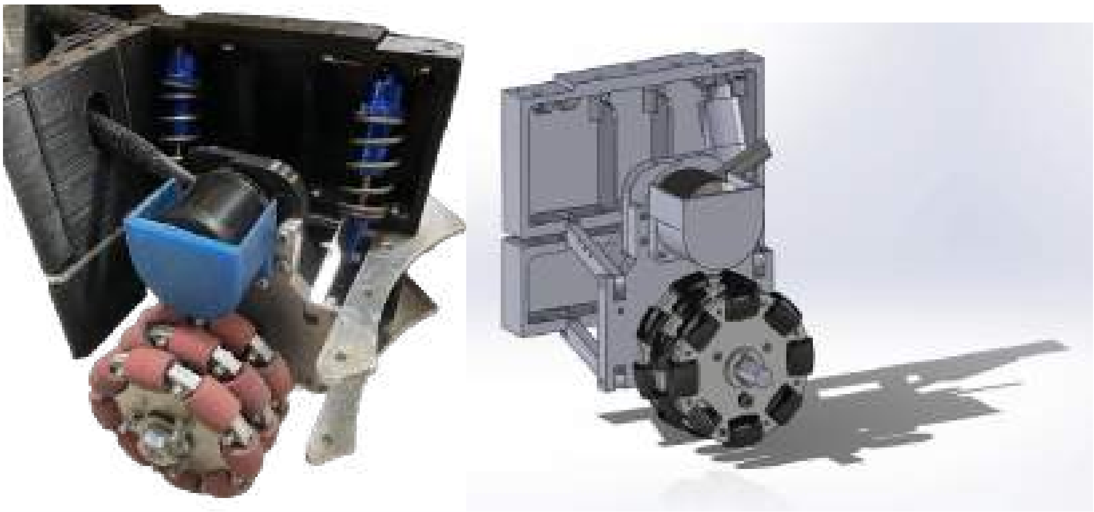


Figure 1 Power module with suspension

The design allows for parts of suspension system , which is a key subassemby of the robot, to be manufactured using only 2.5D milling. Combined with 3D printing, everything can be built with 2 simple machines, while still offering impressive stiffness and durability.

The suspension system was tested with 80 kg of weight (an average human + robot parts) and it behaved well. By changing springs, the robot can be adapted to different carrying capacity, safely to 50+ kg. The robot could easily be adapted to carry a 6 degree robotic arm and additional battery power packs.

Two 6S LiPo 5000 mAh batteries were replaced with in house battery assembly with double capacity and same voltage. Cells were ordered and the rest of the battery pack was designed and assembled in house. In comparison to the older version, the new battery pack is equipped with a battery management system (BMS), and clips for easier assemble/disassemble feature.

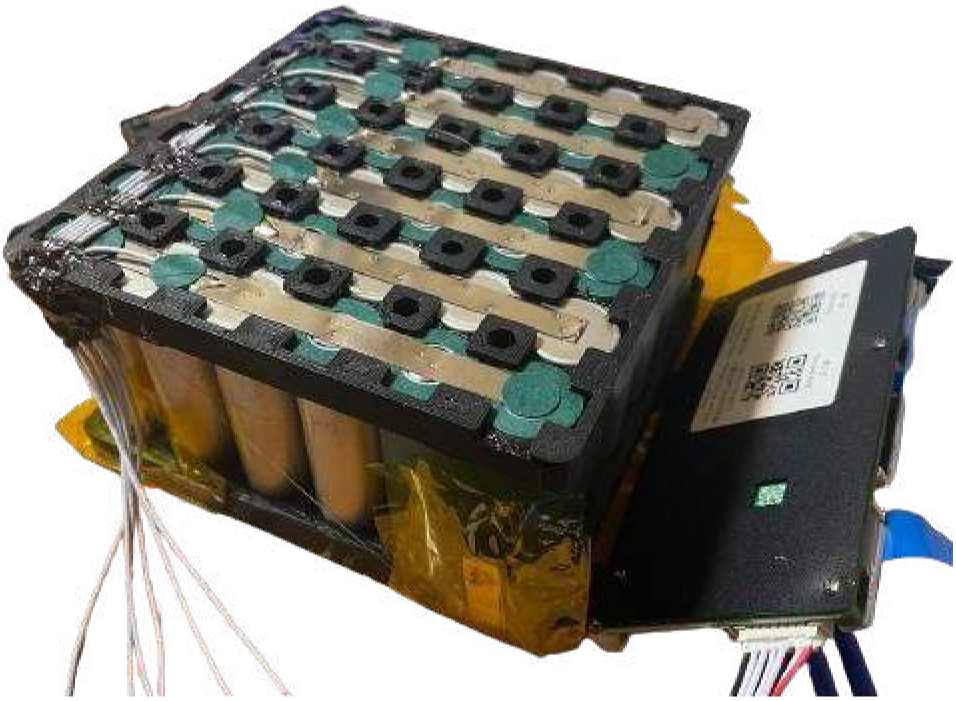


Figure 2 Inside of a 12S3P battery pack with BMS

Dribbling mechanism

The ball is maintained in robot possession with flexible rubber (high energy dense speargun latex rubber) which is equipped with non-torsive wire inside of it. The rubber is connected to the shaft with clips and is rotated with a brushless Flysky motor through belt transmission. The shaft is positioned with 2 bearings inside of the 3D printed housing. The housing is attached to the aluminium panel which is also used to mount the motor and transmission. Whole assembly is connected on top of the third platform with bolts. The whole assembly can be adjusted in height depending on rubber length, friction and diameter to achieve optimal contact with the ball. Friction between ball and rubber maintains contact and the peripheral speed assures positioning and catching.

Figure 3 Dribbling mechanism

When the ball is outside of the robot reach and it is not detected with a vision system, there is no signal to the motor driver and the rubber doesn't rotate. When the ball is within reach and there is a need to catch the ball, the rubber starts to spin and sucks the ball. When the ball is in possession, the speed of rubber is regulated depending on the next desired move. It can rotate in both directions, so the ball can be rotated in a neutral manner or towards the robot centre. When kicking of the ball occurs, rubber stops spinning and the kick is derived without resistance.

Kicking mechanism

The kicking mechanism is positioned on the third robot platform. Action of kicking the ball is derived mechanically and not electrically like the most MSL solutions. The ball is kicked with an aluminium leg which is connected to moving parts of the robot. Energy which kicks the leg is stored in mentioned latex rubber which is stretched to a longer than nominal length. All parts of the kicking mechanism are made of aluminium and steel with few 3D printed parts and some standard elements.

Stepper motor with brake system drives spindle through belt transmission with ratio 1. Rotational movement of the spindle provides linear movement of the nut on which is mounted the trigger mechanism (picture). Position of the trigger mechanism can be adjusted very precisely. Its position influences the strength of the kick. On the platform is a rail on which the U profile slides back and forth. At the initialization of the kicking action, the trigger mechanism attaches the U profile with 2 steel half moons (Picture) and positions both of the elements to the desired point. The rubber is mounted to the U profile from both sides with bolts. The other end of the rubber is mounted to the platform walls. When the U profile attached to the trigger mechanism reaches the desired position, the signal to the servo motor in the trigger determines when the kick will occur. Servo motor latches the two half moons and the U profile with stretched rubber bands is then free to move. It happens instantly and the energy stored in the rubbers drags the U profile towards the block that is attached to the robot leg. Bump energy is transferred to the leg, and that energy is transferred to the ball in possession.

Figure 4 A "trigger" mechanism for releasing band tension with reduced force

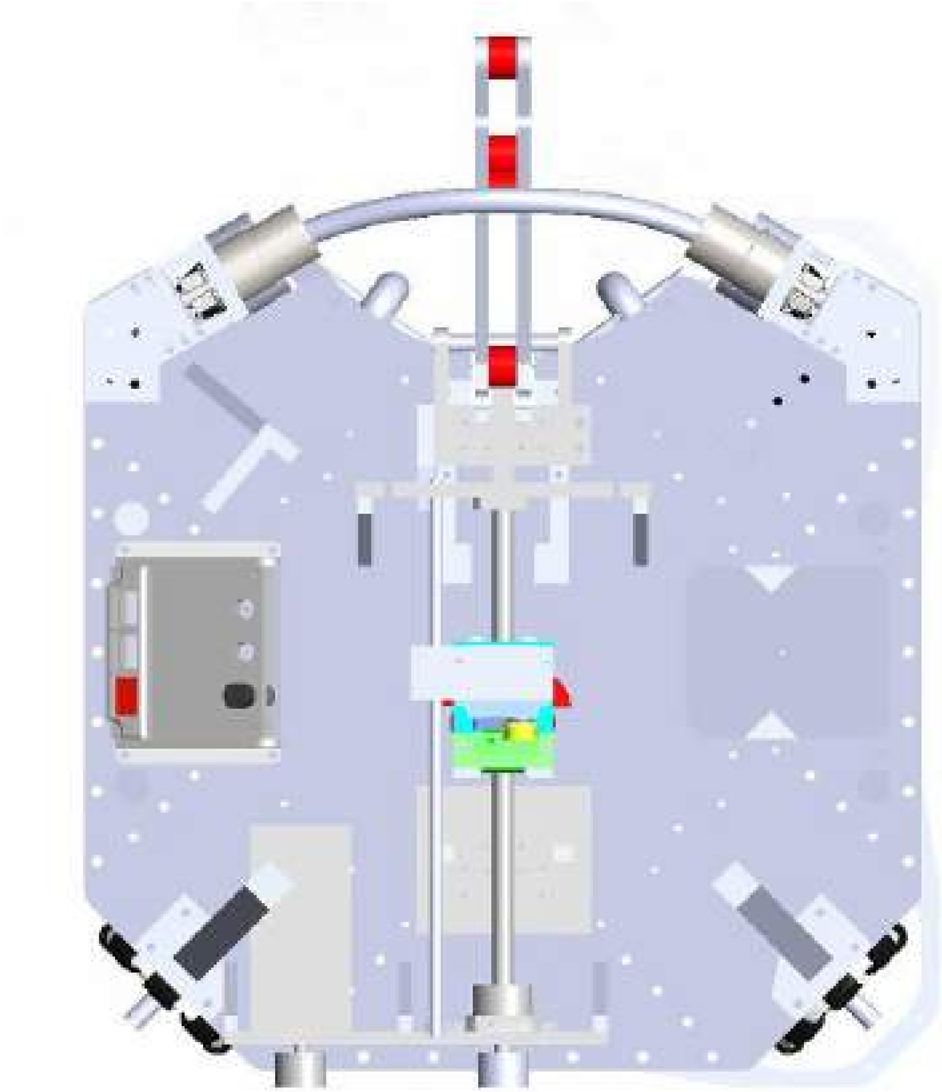


Figure 5 Top view of the robot without electronics box

Figure 6 A view of the robot that shows the underside. Everything is sealed from dust.

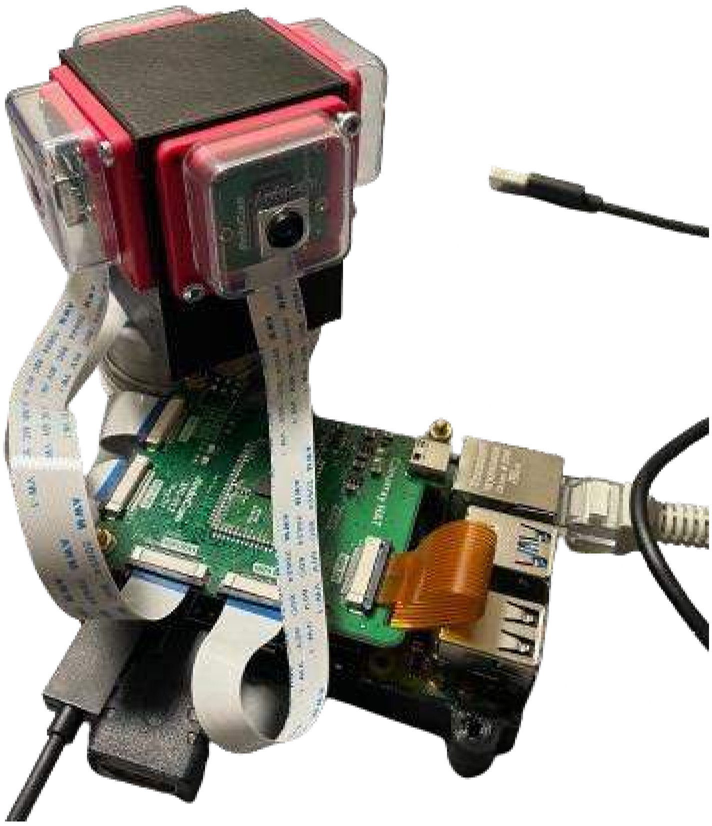


Figure 7 360 degree vision solution using Arducam and Raspberry Pi

Software Description

The robot software is built using Robot Operation System 2 (ROS2). This allows us to quickly prototype and experiment with different sensor modules and parts. Our system consists of three types of nodes: Input, Output and Control nodes. Each input (sensors, camera) has its own node that converts raw data to data formats used by other nodes.

Our sensor (input) nodes include (but are not limited to):

- IMU_I2C node - Reads compass data from our IMU
- Ros2_t265 node - Reads data from our Intel Realsense T265 tracking camera and publishes the transform to TF2
- Camera Stream Node - Reads the ethernet stream of the quad camera system running on a Raspberry PI
- Lidar_Node - Reads lidar data over USB and forwards it to other nodes ● Kinect_Node - Reads RGBD data from Kinect v2

Our output nodes include (but are not limited to):

- Motor_Controller - Calculates the inverse kinematics for the omni wheels and sends commands to the motors using CAN bus.
- Dribbler_Controller - Communicates with the dribbler/kicker microcontroller to set the dribbler speed
- Kicker_Service - Sends commands to the dribbler/kicker microcontroller to set the power of the kicker and to kick the ball

The centre of our sensor stack is the T265 tracking camera that runs VSLAM onboard and outputs 6DOF coordinates of relative movement. The transformation is then published into

TF2 where all other sensors are positioned with static transformations. Below is our TF2 transformation tree. Static transformations are shown in red and dynamic transformations are shown in black. When the T265 camera starts tracking, it tracks relative to it's start position "map". This is then statically transformed to the "world" frame which represents the centre of the field. The TF2 system is also used to define the robot's movement. The "target" frame is positioned relative to the world and is then queried from the "robot" frame to get the vector that represents the direction in which the robot should move to reach the target. Detected obstacles are also placed in the TF2 tree and the VFF algorithm is used to decide how the robot should move to reach the target while avoiding obstacles.

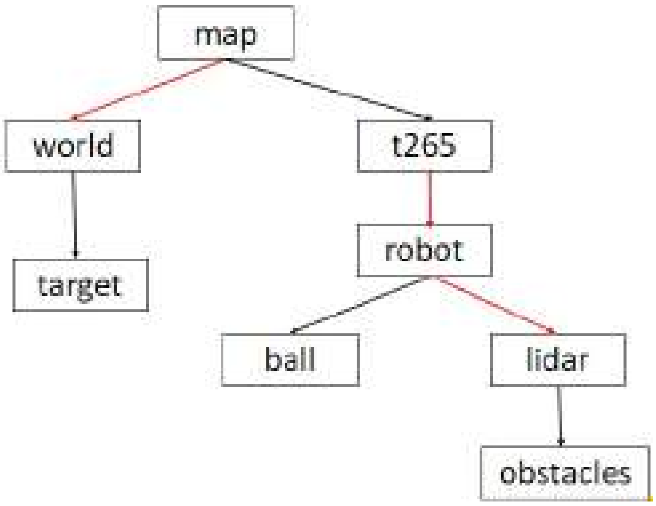


Figure 8 TF2 transformation tree

Simulation

To enable fast prototyping of our control loops and decision algorithms we developed a digital twin simulation of our robot using the Gazebo simulator. The main benefit of the simulation is that it allows us to use the same code that is in the physical robot in the simulation with minimal changes (velocities, PID control loops).

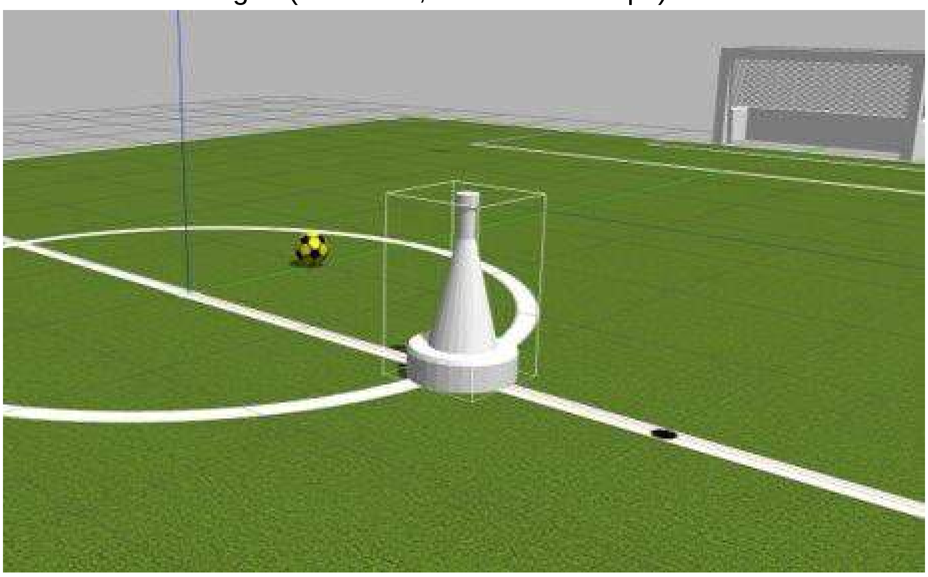


Figure 9 Simulated environment

The simulation replaces all ROS2 nodes that have contact with the physical world. All input nodes (Camera, IMU, tracking camera etc.) and all output nodes (Motor controller and kicker/dribbler controller) have twin nodes in the simulation which the rest of the software can interface with. Below is a bare-bones diagram of our ROS2 graph with annotated input and output nodes that get replaced by the simulation.

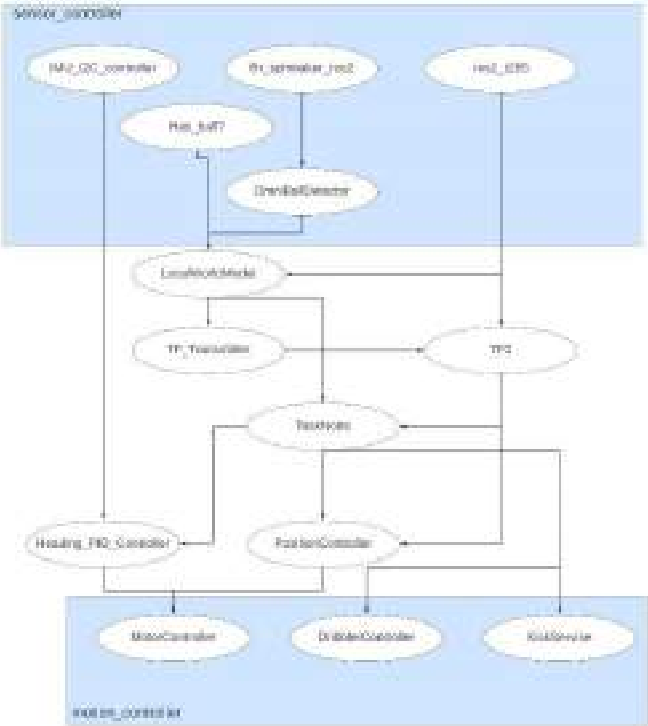


Figure 10 ROS2 graph

Hardware description

This part of the document describes hardware related systems on the robot. Relevant schematics are located at the end of the document.

Power distribution system

Robot is powered by a single Li-Ion battery pack in 3P12S configuration to achieve nominal voltage of 44,3 V with capacity of 9000 mAh and current output capability of 100A continuous.

Each of the five Vedder's Electronic Speed Controllers (VESC), four to control 1000W BLDC motors that are driving our omni wheel movement system, an additional VESC for controlling our dribbling system, and a stepper driver used to drive the kicker system tensioning system, are connected directly to battery voltage through a custom anti-spark switch with inrush current control. Rest of the robot is divided into two separate power circuits. One circuit powers our main processing unit (Jetson NANO) in addition to all cameras and peripherals connected directly to it. Another circuit is used to power the rest of the modules on the robot.

First circuit is a dual voltage system consisting of a 12V rail supplied by a switch mode power supply, used to power the Jetson computer, and a 5V rail made by stepping down the 12V using a buck converter. 5V rail is used to power USB peripherals.

Second circuit consists of three voltage rails. First is a 28 V rail, used to power the tensioning system brake and two stepper drivers used to control the kick angle adjustment system. 8V rail powers a servo motor used to trigger the shooting mechanism. A 5V rail is used to power the individual control modules (Kicker controller, button box etc.)

Communication

To enable communication between different modules on the robot, a CAN communication protocol is used. All devices are connected to a single CAN bus, so the data can be accessed by any module, separately from the main processing unit, additionally, any error that occurs on individual modules can be detected. CAN bus controllers are either separate modules or built-in, depending on the hardware used for a specific module.

Movement system

The omni wheel system of our robot uses 4 1000W BLDC motors that are controlled by Vedder's Electronic Speed Controllers (VESC). Robot's dribbler is also controlled by the same system. VESCs communicate with the Central processing unit (Jetson NANO) through a CAN bus.

Kicker control system

The kicker of our robot uses an ESP32 microcontroller with a custom shield that contains necessary logic level converters and passive components to adjust the 3.3V logic on the ESP32 to the rest of the robot which uses 5V. Kicker control system acquires commands from the central processing unit (Jetson NANO) through the CAN bus. Kicker control system controls three steppers, one to tension the kicking system, and other two to adjust the angle of the kicking system. Movement of each component on threaded spindles moved by these stepper motors have limit switches to secure the kicking system from self-destruction. Kicker control system also controls the brake for the tensioning system and a servo motor for triggering the kick. It is also connected to the emergency stop signal to distension the kick system if the robot is force stopped.

Button box

It is a separate module on the robot to allow the quick change of certain software parameters on the go. Buttons are connected to an Arduino that relays button states to the Jetson via CAN. Button box can also activate certain LED indicators if there is an error.

Power distribution box

A simple plastic box containing three buck converters used to supply 5, 8 and 28 V for devices as described above. Each buck converter has a separate protective fuse and a switch that enables us to turn off each buck converter individually if there is need for maintenance purposes on the go, without the need to shut down the robot.

Object tracking system

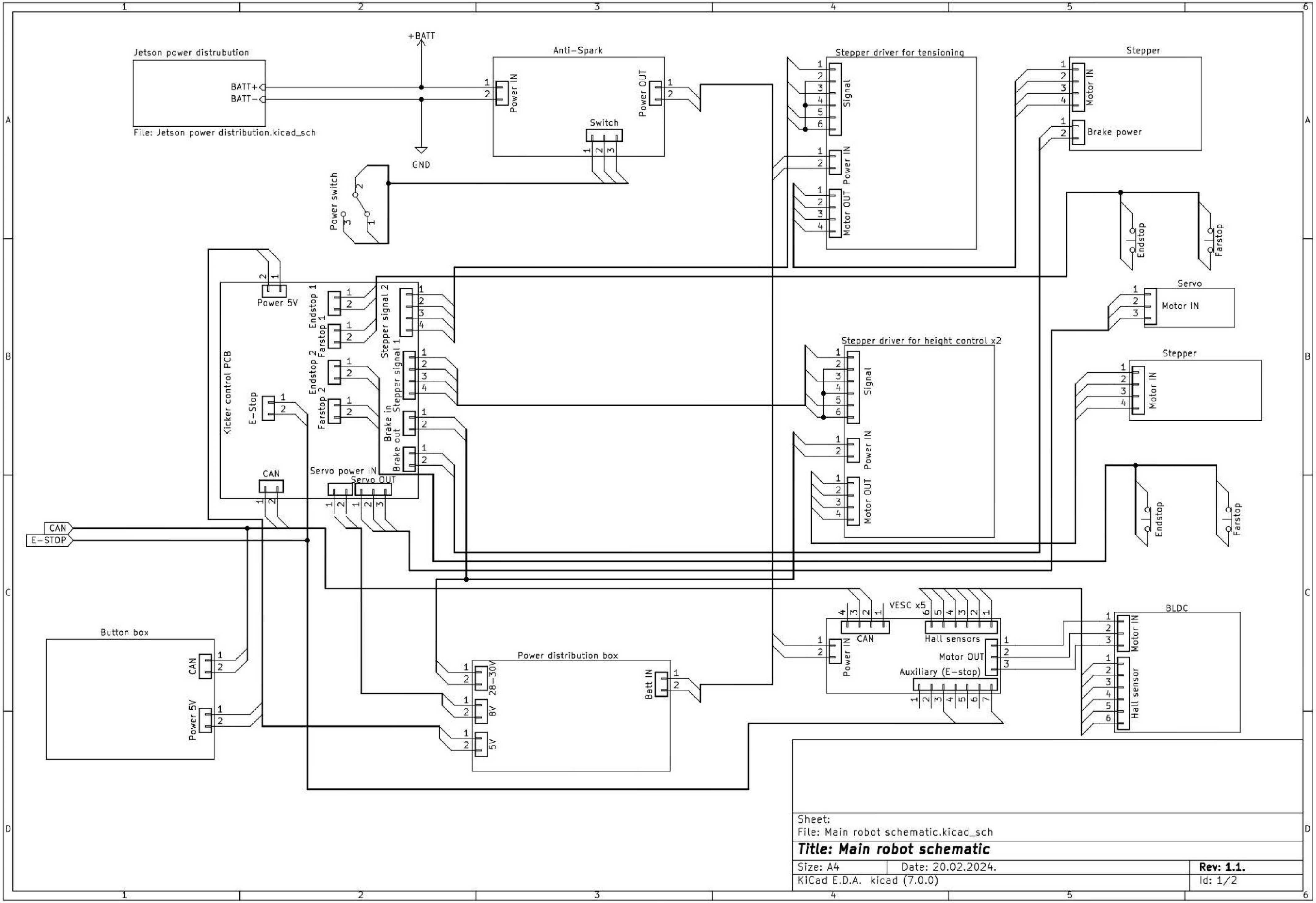
Object tracking system consists of a 360 degrees FOV camera that is mounted on the top of the robot, and a Kinect that is mounted front facing about halfway between the dribbler and the camera. The camera is used to track objects around the robot, while the Kinect is primarily used to determine the distance to the ball when the robot is facing it. Camera and Kinect are connected to the Central processing unit via USB cables. To detect obstacles, a LIDAR sensor mounted on the underside of the top plate is used.

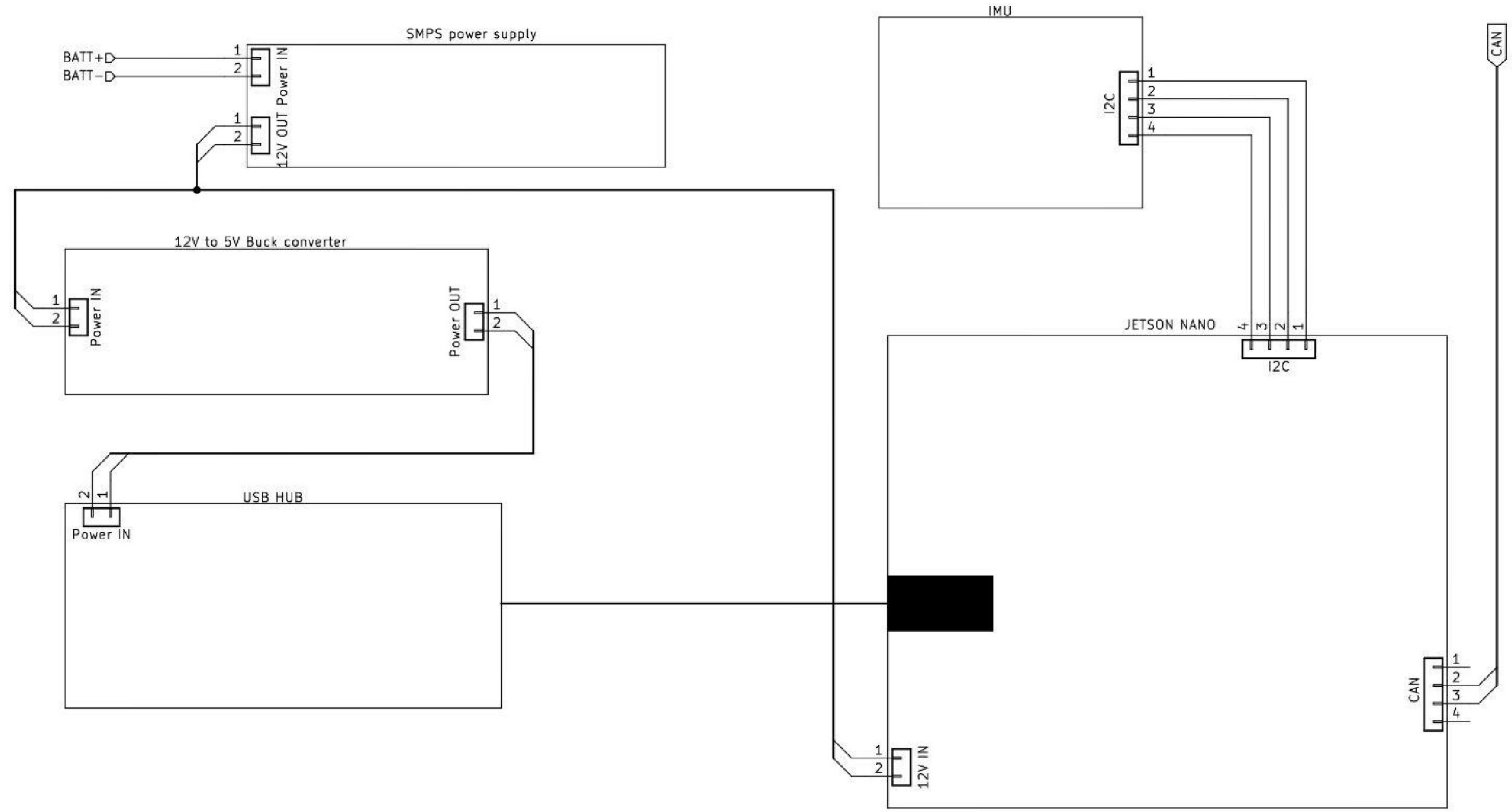
Inertial measurement unit and position tracking

Robot has two positioning systems, one using an IMU connected through I2C directly to the Jetson computer, the other is an Intel RealSense camera that relays relative spatial coordinates to the Jetson via USB.

Anti-spark switch

Anti-spark switch is a MOSFET based switching circuit, whose advantage over traditional ways of closing circuits in high current circuits is protection from sparking. Main components of our anti spark switch are: a voltage regulator for reducing voltage from batteries to a microcontroller, an optocoupler, and seven N-channel high current MOSFETs. Microcontroller is used for controlling MOSFETs through a two channel optocoupler. One channel is for controlling a single MOSFET used for precharging the load capacitors, while the other channel is for controlling other six MOSFETS whose purpose is to connect or cut off power on user demand. The high current going through the anti-spark switch creates nonnegligible heating of the MOSFETS and therefore it requires substantial cooling. The switch dissipates its heat through two passive coolers.

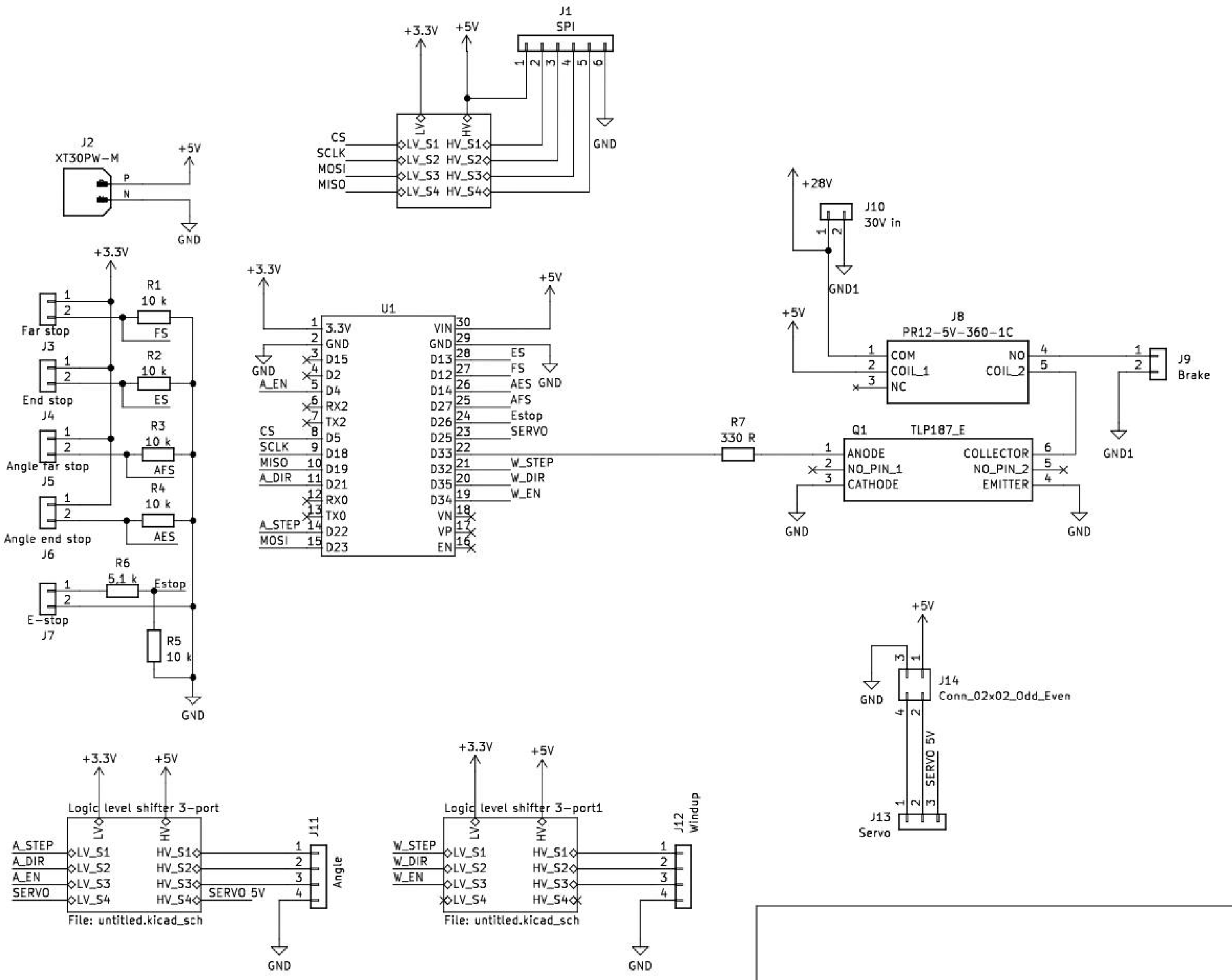

Jetson power distrubution
BATT+
BATT−
File: Jetson power distribution.kicad_sch
+BATT
GND
Anti−Spark
Power IN
Power OUT
Switch
Power switch
Stepper driver for tensioning
Signal
Power IN
Motor OUT
Stepper
Motor IN
Brake power
Endstop
Farstop
Servo
Motor IN
Stepper driver for height control x2
Signal
Power IN
Motor OUT
Stepper
Motor IN
Endstop
Farstop
Kicker control PCB
Power 5V
Endstop 1
Farstop 1
Endstop 2
Farstop 2
Stepper signal 2
Stepper signal 1
Brake in
Brake out
E−Stop
CAN
Servo power IN
Servo OUT
CAN
E−STOP
Button box
CAN
Power 5V
Power distribution box
28−30V
8V
5V
Batt IN
VESC x5
CAN
Hall sensors
Power IN
Motor OUT
Auxiliary (E−stop)
BLDC
Motor IN
Hall sensor
Sheet:
File: Main robot schematic.kicad_sch
Title: Main robot schematic
Size: A4
Date: 20.02.2024.
Rev: 1.1.
KiCad E.D.A. kicad (7.0.0)
Id: 1/2



Sheet:
File: Jetson power distribution.kicad_sch

Title: Main robot schematic

Size: A4	Date: 20.02.2024.	**Rev: 1.1.**
KiCad E.D.A. kicad (7.0.0)		Id: 2/2

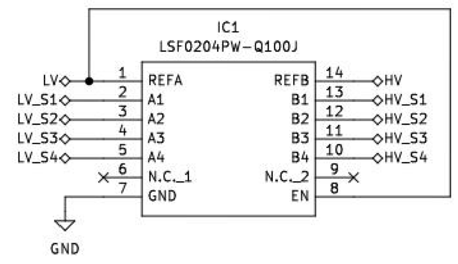


Sheet:
File: Kicker V2.1.kicad_sch

Title: Kicker control PCB

Size: A4 | Date: 2024-02-20 | **Rev: 1.0.**

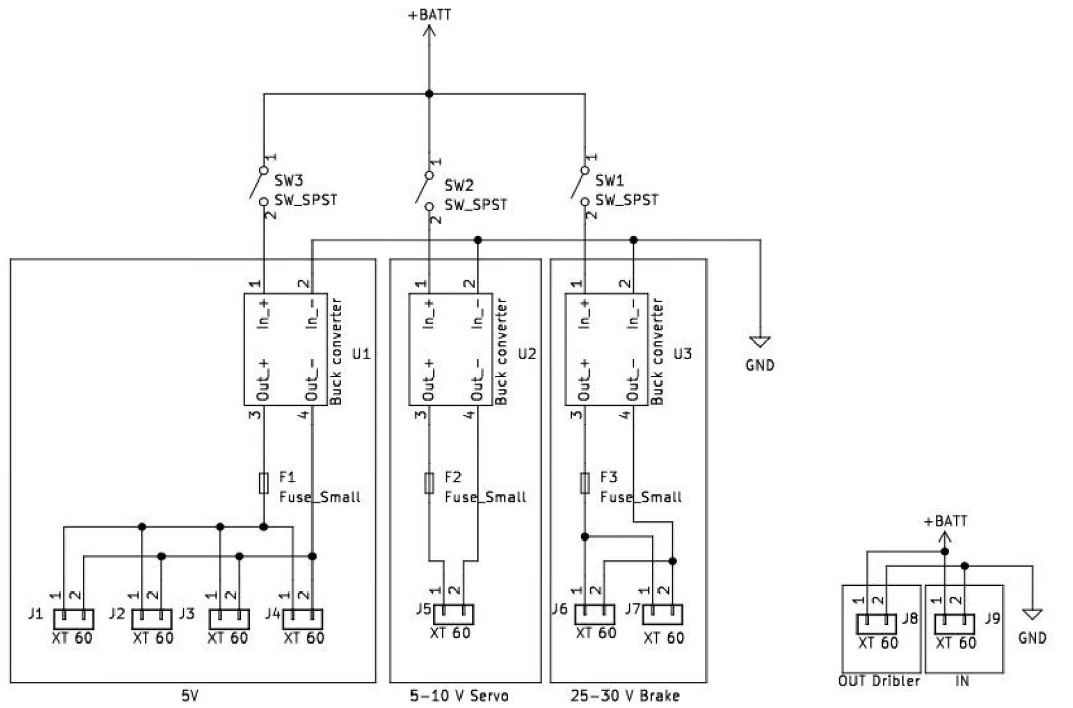
KiCad E.D.A. kicad (7.0.0) | Id: 1/4



Sheet:
File: untitled.kicad_sch

Title: Kicker control PCB

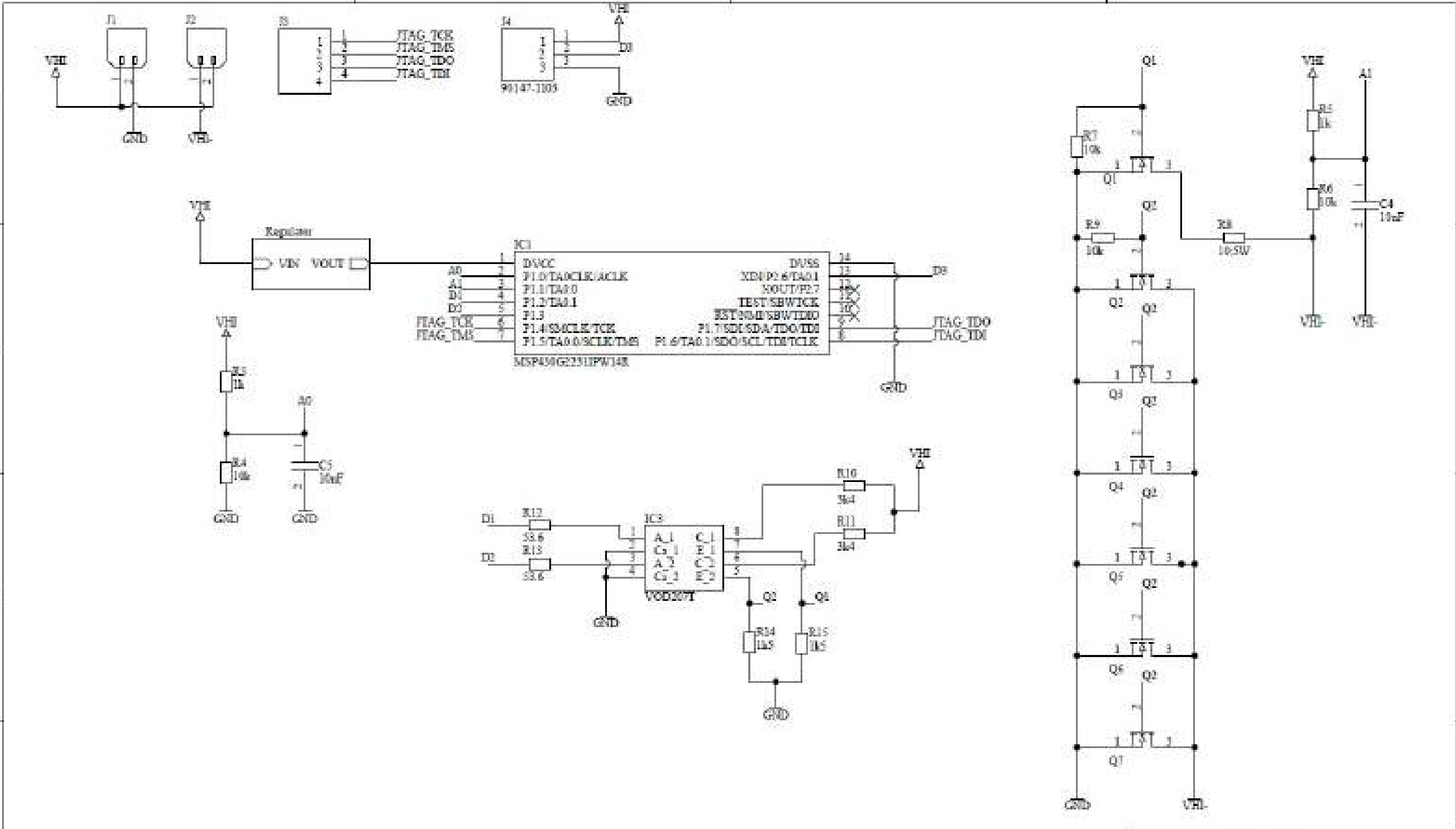
Size: A4	Date: 2024-02-20	**Rev: 1.0.**
KiCad E.D.A. kicad (7.0.0)		Id: 2/4



Sheet:
File: Kutija za napajanje.kicad_sch

Title: Power distribution box schematic

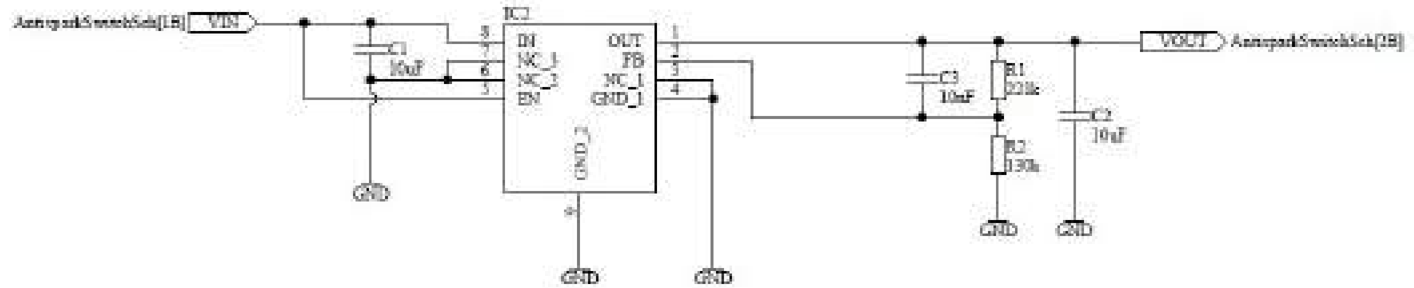
Size: A4	Date: 20.02.2024.	**Rev: 1.0.**
KiCad E.D.A. kicad (7.0.0)		Id: 1/1



Title Antispark Switch

Drawn By: Josip Vindiš

Size: A4	Number: v1.0
Date: 01/03/2024	Sheet 1 of 2



Title: **Antispark Switch**

Drawn By: Josip Vindiš

Size: A4	Number: v1.0
Date: 01/03/2024	Sheet 2 of 2