Peripherals and robotics PCB for Control Unit boards

Department Lippstadt 2

Project Work
for obtainment of the degree of
Bachelor of Engineering
submitted by

Shahzaib Waseem

Electronic Engineering
Mat.Nr.: 2210012
<shahzaib.waseem@stud.hshl.de>

February 13, 2025

Supervisor: Prof. Dr. Ali Hayek

Contents

1 Requirements

The requirements of this project work are described as follows:

1. A PCB must be designed onto which custom Control Unit boards can be connected with the so-called 'High Density Connectors'
2. There should be negligible wiring or manual configuration needed to use the PCB. Embrace the 'plug and play' idea
3. The PCB must be multipurpose. Should be usable for different courses with different Control Unit requirements. For example, FPGA, MCUs, etc.
4. The PCB should allow for robotics projects, and should be be designed in an optimal structured way for this purpose. Proper, visible separation for robotics components is needed.
5. The PCB is centered around the Arduino Portenta H7. This Arduino board can be directly connected to the PCB, and it interfaces with the electronic components on the PCB out of the box.
6. The High Density Connectors on the Arduino Portenta H7 are to be used as reference when desiging PCBs for other Control Unit boards
7. The PCB must allow for any and every Control Unit to interface with **all** the electronic components on it, using its High Density Connectors.
8. Control Unit boards, including the Arduino Portenta H7, are provided with the appropriate power from the PCB. All Control Unit boards simply require 'flashing' or uploading of some sort for them to be 'programmed' or configured beforehand to be used with the PCB.

2 Pin mappings

This 'Peripherals and robotics PCB for Control Unit boards' acts as an interface between various electronic components that can either be actuators or sensors for any control unit. The control unit that connects to this PCB needs its own PCB design that matches the connections or pin mappings of the connectors found on this board.

In order to develop, maintain, and use this PCB in a structured, scalable, and efficient way, a table was constructed that maps the various electronic component pins to that of the headers onto which a MCU board would be placed.

These are called 'High density connectors' and, as the name suggests, contain a high number of pins for connections. Specifically, these are in total 80 pins. Since the PCB contains a relatively large number of electronic components, 2 of these high density connectors have been used. The tables describing the pin mapping are shown below in tables 1.1, 1.2, 1.3, and 1.4.

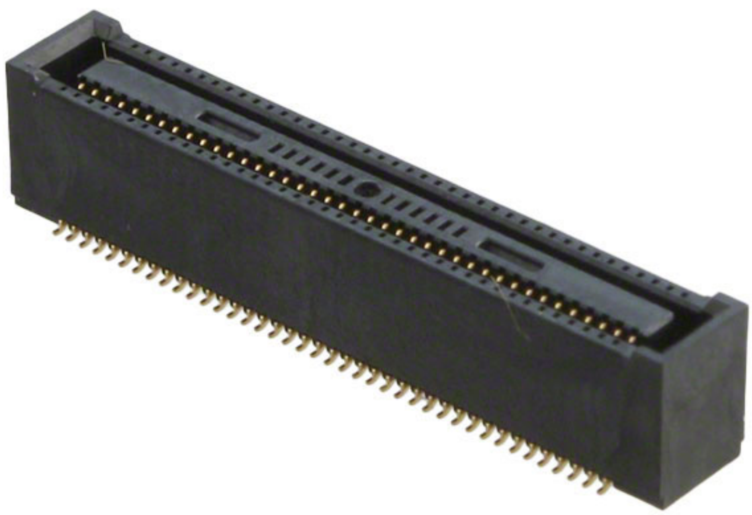

Figure 2.1: High density pin connectors [5]

Connector codes	Electronic component	Arduino Portenta H7 pin connection
X1	Keypad R1	
X2	Keypad R2	
X3	Keypad R3	
X4	Keypad R4	
X5	Keypad C1	
X6	Keypad C2	
X7	Keypad C3	
X8	Keypad C4	
X9	SPI SCK	
X10	SPI MISO	
X11	SPI MOSI	
X12	HC05 BLE RXD	
X13	HC05 BLE TXD	
X14	ESP8266 WLAN RXD	
X15	ESP8266 WLAN TXD	
X16	GPIO 1	
X17	GPIO 2	
X18	GPIO 3	
X19	GPIO 4	
X20	DNC	
X21	VIN	
X22	GND	
X23	DNC	
X24	LED 1	
X25	LED 2	
X26	LED 3	
X27	GPIO 5	
X28	GPIO 6	
X29	GPIO 7	
X30	GPIO 8	
X31	GND	
X32	VIN	
X33	LED 4	
X34	LED 5	
X35	LED 6	
X36	LED 7	
X37	LED 8	
X38	LED 9	
X39	LED 10	
X40	LED 11	

Table 2.1: X connectors

Connector codes	Electronic component	Arduino Portenta H7 pin connection
X41	VIN	
X42	GND	
X43	DNC	
X44	I2C SDA	D11
X45	DNC	
X46	I2C SCL	D12
X47	GND	
X48	VIN	
X49	LED 12	
X50	LED 13	
X51	LED 14	
X52	LED 15	
X53	VSYS	
X54	GND	
X55	DNC	
X56	Switch 1	
X57	Switch 2	
X58	Switch 3	
X59	Switch 4	
X60	Switch 5	
X61	Switch 6	
X62	Switch 7	
X63	Switch 8	
X64	VSYS	
X65	Switch 9	
X66	Switch 10	
X67	Switch 11	
X68	Switch 12	
X69	Switch 13	
X70	Switch 14	
X71	Switch 15	
X72	Switch 16	
X73	RESET	
X74	DNC	
X75	Push button 1	
X76	Push button 2	
X77	Push button 3	
X78	Push button 4	
X79	Potentiometer 1	
X80	DNC	

Table 2.2: X connectors

Connector codes	Electronic component	Arduino Portenta H7 pin connection
Y1	Color sensor S0	
Y2	Moisture sensor signal	
Y3	Color sensor S1	
Y4	Color sensor S2	
Y5	GPIO 9	
Y6	GPIO 10	
Y7	GPIO 11	
Y8	GPIO 12	
Y9	GPIO 13	
Y10	GPIO 14	
Y11	GPIO 15	
Y12	GPIO 16	
Y13	Buzzer	
Y14	Color sensor S3	
Y15	Seven segment display (segment a)	
Y16	Seven segment display (segment b)	
Y17	Seven segment display (segment c)	
Y18	Seven segment display (segment d)	
Y19	Seven segment display (segment e)	
Y20	Seven segment display (segment f)	
Y21	Seven segment display (segment g)	
Y22	Color sensor Out	
Y23	VCC / +3V3	
Y24	GND	
Y25	Seven segment display (anode 1)	
Y26	Seven segment display (anode 2)	
Y27	Seven segment display (anode 3)	
Y28	Seven segment display (anode 4)	
Y29	Seven segment display (anode 5)	
Y30	Seven segment display (anode 6)	
Y31	Seven segment display (anode 7)	
Y32	Seven segment display (anode 8)	
Y33	GND	
Y34	VCC	
Y35	DNC	
Y36	Motor driver IN2	D7
Y37	DNC	
Y38	Left IR sensor pin	D9
Y39	DNC	
Y40	Right IR sensor pin	D10

Table 2.3: Y connectors

Connector codes	Electronic component	Arduino Portenta H7 pin connection
Y41	DNC	
Y42	Motor driver IN4	D8
Y43	VCC / +3V3	
Y44	GND	
Y45	DNC	
Y46	PIR motion sensor	
Y47	DNC	
Y48	LoRa G0	
Y49	LoRa CS	
Y50	LoRa RST	
Y51	RFID CS	
Y52	RFID RST	
Y53	DNC	
Y54	DHT22 temperature sensor pin	
Y55	DNC	
Y56	DNC	
Y57	GND	
Y58	DNC	
Y59	Distance sensor trigger	D6
Y60	Motor driver IN1	D1
Y61	Left servo signal	D5
Y62	Motor driver EnA	D0
Y63	Motor driver IN3	D4
Y64	DNC	
Y65	Motor driver EnB	D3
Y66	DNC	
Y67	Right servo signal	D2
Y68	DNC	
Y69	VCC / +3V3	
Y70	GND	
Y71	ADC VREF +	
Y72	ADC VREF -	
Y73	LDR	A0
Y74	Piezo vibration sensor	A4
Y75	Potentiometer 2	A1
Y76	Distance sensor echo	D20
Y77	LED 16	A2
Y78	Push button 5	D21
Y79	Microphone analog pin	A3
Y80	DNC	

Table 2.4: Y connectors

3 Features

This PCB allows users to test and interface their desired Control Unit boards with a variety of sensors and actuators.

The terms sensors and actuators are general and describe all kinds of electronic components with which information or energy is either transmitted to the real world, or taken from it. Common electronic components that are used in majority of projects or labs around the world for the purpose of learning embedded systems are included in this PCB. These electronic components are listed as follows:

1. IR sensors: YL-63
2. Motor driver: L298N
3. DC Motors
4. Distance sensor: (Ultrasonic) HC-SR04
5. Microphone module: KY-038
6. Servo motor: SG90
7. Single Pole Single Throw (SPST) Switches
8. Normally Open (NO) Tactile Push buttons
9. LEDs
10. Seven segment display: LTC-4627JD
11. Potentiometer
12. Temperature and humidity sensor: DHT-22
13. PIR motion sensor: HC-SR501
14. Piezo vibration sensor
15. 4x4 matrix keypad
16. ESP8266 WLAN module
17. Light Dependent Resistor (LDR)
18. Bluetooth module: HC-05
19. Passive buzzer
20. Color sensor: TCS3200
21. Capacitve moisture sensor: HW-390

22. RFID sensor: RC522
23. LoRa module: RFM95W
24. OLED: SSD1306 I2C

The 'Applications' chapter outlines examples of suitable use cases for the PCB.

4 Robotics applications

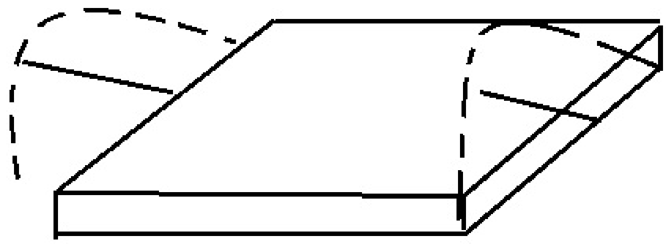

Figure 4.1: Servo motor movement for robotic arm projects

The PCB contains electronic components for robotics projects. There are servo motors that allow specifically for robotic arms projects. Figure 3.1 shows an example of locus or axis in which a robotic arm would move.

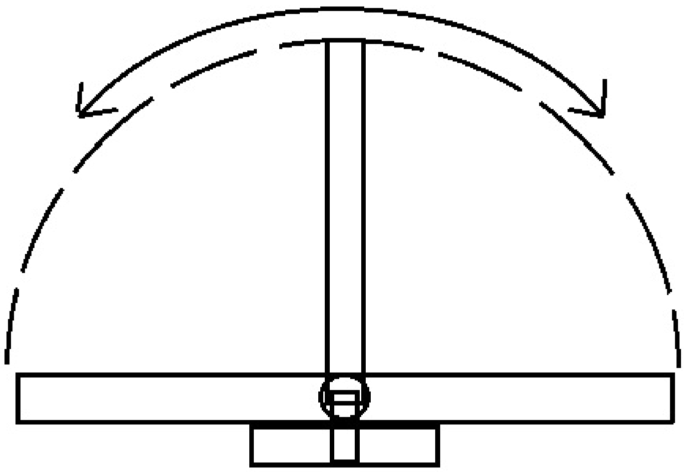

Figure 4.2: Servo motor movement: Side view of PCB

The robotic arm would be able to move in a circular motion or have a rotational axis parallel to the length of the PCB. Depending on the type of servo motor used and attached, the range of angles in which the robotic arm would move, for example, could be from 0 to 180 degrees. This is shown in the PCB's side view in figure 3.2.

Figure 3.3 showcases how a 3D-printed robotic arm could be attached to the PCB in the designated spot of the servo motors. Such a robotic arm setup could be used in an obstacle removing robot project. The robot could pick or lift up a small and light obstacle using a curved part that would allow the target obstacle to simply slide onto the space inside and

Figure 4.3: Heavy duty construction vehicle: front-loader [8]

be carried off of the ground and be displaced wherever the robot is controlled to go to for dropping off said obstacle.

Similar to an obstacle removing robot, the PCB can also be used for an obstacle avoidance robot project, as shown in figure 3.4. The ultrasonic distance sensor HC-SR04 can be used to detect obstacles in the robot's path, and then a control unit connected to the PCB can allow the robot to take decisions to maneuver around it and thereby avoid the obstacle.

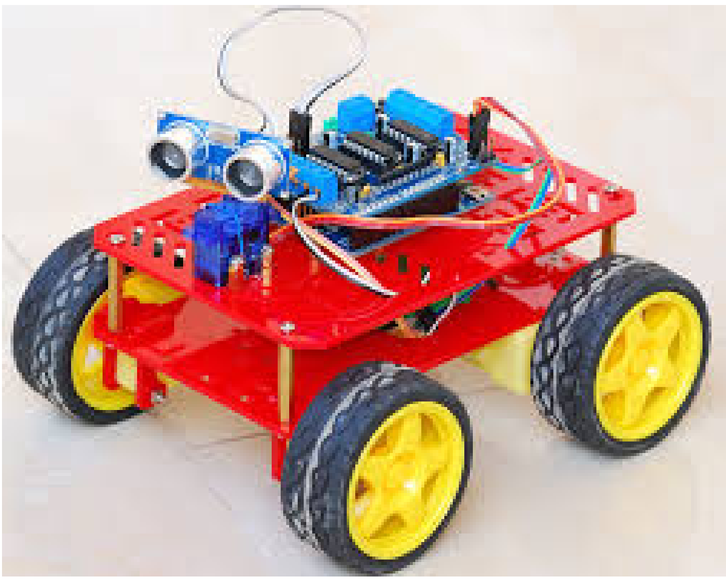

Figure 4.4: Example of an obstacle avoidance robot [2]

Finally, there is also the IR sensors in the PCB that allow for building a Line Following Robot (LFR) project. Two IR sensors are placed side-by-side with a gap within which a line would be situated that would be followed by the robot. The IR sensor would need to be soldered onto the male connectors on the PCB, found in the designated spot along the bottom-front edge of the PCB. Example of a line following robot (LFR) is show in figure 3.5.

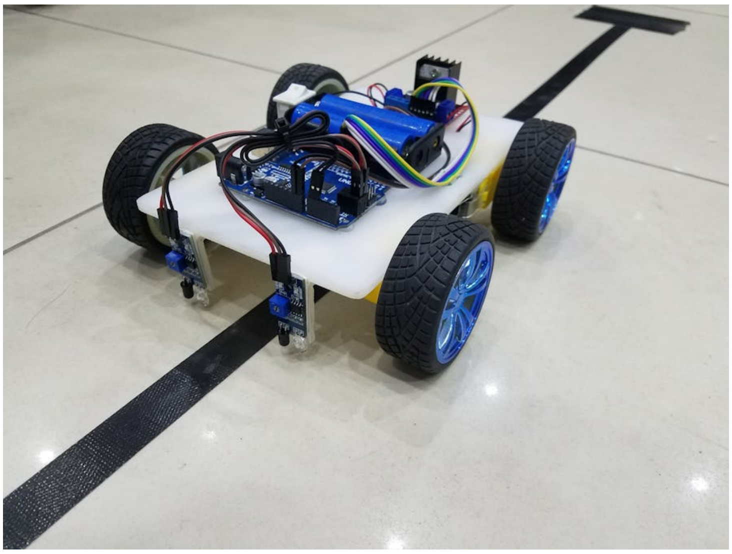

Figure 4.5: Example of a line following robot (LFR) [7]

The distance between the two IR sensors for a LFR can be freely adjusted based on the position they are attached to at the front edge of the PCB.

For any and all robotics projects that require movement through wheels, there are 2 DC motors at the back of the PCB that connect to a motor driver for easy, speed and direction controlled movement similar to that of an automotive vehicle. This is made possible by attaching a ball wheel using hot glue near the middle edge of the bottom-front of the PCB in order to balance the PCB according to its center of gravity.

5 PCB design

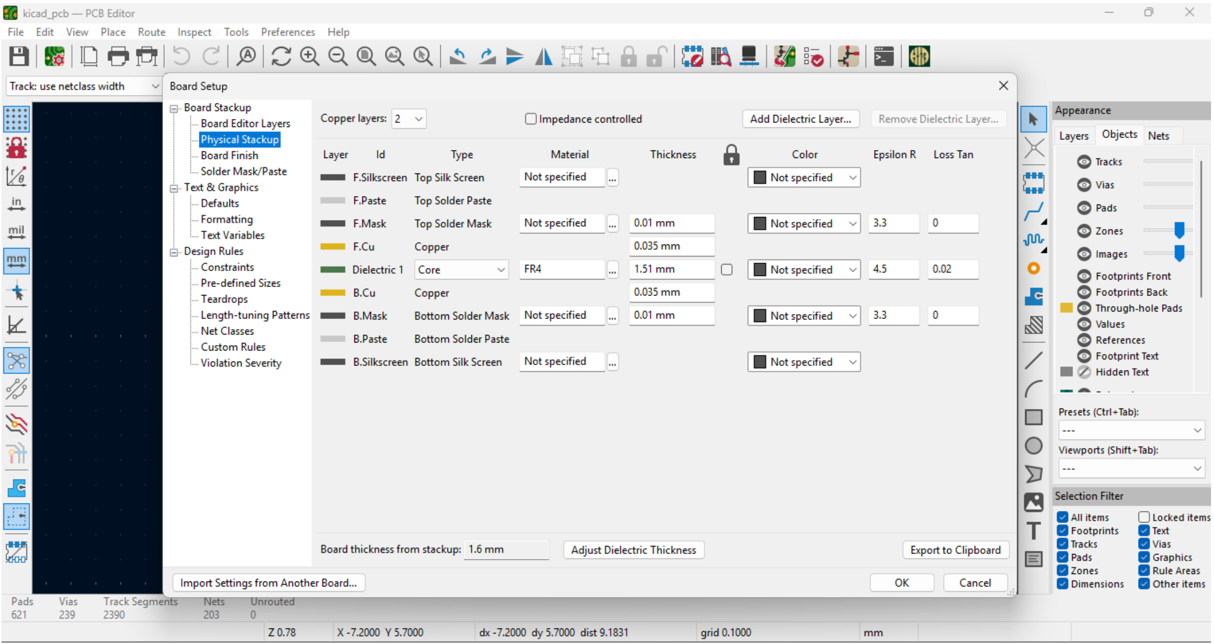

Figure 5.1: PCB layers and stackup information

Figure 4.1 showcases the information related to the layers and stackup of the PCB design in this project. Below are the key points described in figure 4.1:

1. Dimensions: 180 mm x 180 mm
2. Board thickness: 1.6 mm
3. Copper layer thickness: 0.035 mm
4. Number of copper layers: 2 (Top and bottom)
5. Core thickness: 1.51 mm
6. Core material: FR4
7. Trace width: 0.2 mm
8. Clearance: 0.1 mm
9. Via size (PTH diameter): 0.45 mm
10. Via hole (depth): 0.2 mm

Figure 4.2 shows the footprints used for all the electronic components on the PCB, as well the traces connecting them. The red colored traces and copper parts represent the Top

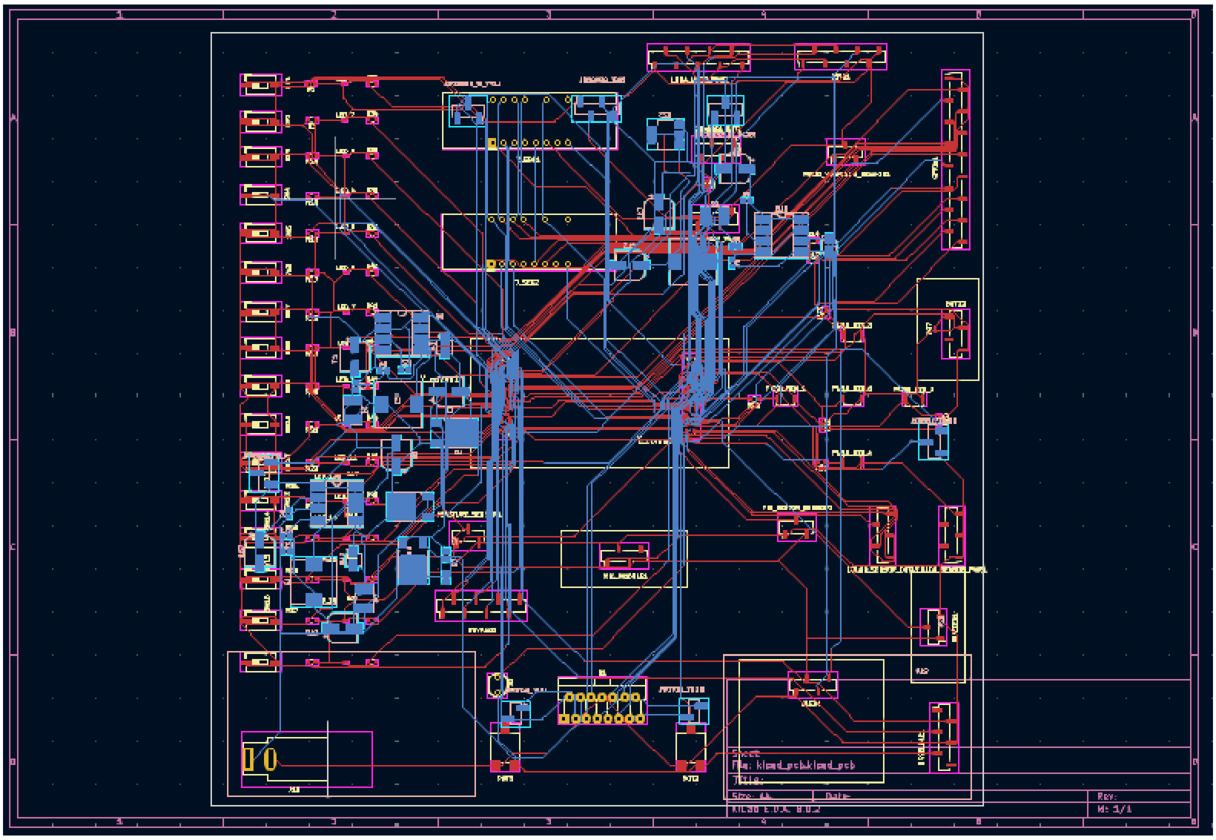

Figure 5.2: PCB design view

layer. This means that these traces and copper parts are found exclusively on the top side of the PCB. The blue colored traces and copper parts represent the Bottom layer. This means that these traces and copper parts are found exclusively on the bottom side of the PCB. Not everything on the PCB is either red or blue. There are through-hole vias and Through Hole Technology (THT) components that would require copper pads on both, top and bottom layers of the PCB to function. As such, these vias and components will be visible on both sides of the PCB.

The PCB uses a total of 2 layers: top and bottom. The top layer contains electronic components associated to 'peripherals' for normal use, for learning how to interface with common actuators and sensors. The bottom layer is exclusively dedicated to electronic components used primarily in 'robotics' projects, such as Line Following Robot (LFR), obstacle avoiding robots, etc. An exception to this rule is the placement of power regulator components on the bottom layer. The reason for this is to save space on the top layer for the large area taken up by the peripherals there, and because power regulator circuitry is almost completely irrelevant for the user. The DC power jack that serves as input for the power regulator circuits is found on the top layer so that it is convenient for the user to plug into it.

The bottom layer contains connectors for servo motors on the left and on the right middle edges of the PCB. Since the bottom layer is for robotics, the two servo motors can act as pivots or bases for 3D-printed robotic arms. The robotic arms may move in several directions for creating a variety of versions of robotic arm projects, such as for an excavator that can scoop-up or lift objects or sand-like material in front. Another set-up of robotic arm could be for grabbing objects located mid-air on top of the PCB at a height within reach of the 3D-printed robotic arm's body. There are also connectors for IR sensors at the bottom edge that are suitable for being used in Line following robot (LFR) projects. For In addition, there is also a connector for proximity or distance sensor HC-05 on the bottom edge making the PCB suitable for, in addition to other use cases, as a object avoiding robot. Finally, for all these vehicle-like robot projects, two DC motor connections are also found on the bottom layer of the PCB onto which suitable DC motors can be soldered to accordingly, and then can be secured or latched appropriately with hot glue, duct tape, 3D-printed parts, etc.

The top layer contains several peripherals, for which a separate list of components can be found along with this document. The top layer contains switches and LEDs that can be interfaced with by a control unit connected to the PCB. There are also female connectors on the top layer that allow common sensors to be directly plugged into the PCB. The connectors are mostly along the edges of the board for ease of connectivity. Due to limited area on the board, some connectors are also further away from the edges and more towards the middle of the PCB. Majority of the footprints used in the PCB are SMD, so that the top and bottom layers and their respective components are isolated from each other for easier and free placement of components. Otherwise, through-hole technology (THT) components would intrude the space on the other side or layer of the PCB, introducing regions where traces cannot be made without causing short-circuits.

For the PCB design, KiCAD 8.0 was used. While working on the schematics of the circuits to be implemented in the PCB design, a hierarchical approach was chosen. This allowed schematics within schematics in a hierarchy that allowed for smooth and easy-to-read development of the schematics in the circuit. Once the schematic was done, it was time to work on the PCB file. As described in the paragraphs above, the electronic components to be placed on the top layer of the PCB were done so while making sure that the limited area on the layer was utilized to the fullest. Empty spaces or gaps were kept to a minimum. The same principles and ideas applied to the bottom layer of the PCB.

To conclude the PCB design, the Design Rule Checker (DRC) was used extensively provided in the PCB design part of KiCAD. The DRC tool displays 2 types of 'negatives' that should be looked out for before creating and sending the Gerber file of the PCB for manufacturing. The first are the warnings. These are simply suboptimal implementations or bad practices in the PCB design that, according to KiCAD, you might need to consider and correct before manufacturing the PCB. The second are the errors. These are actual intolerable design decisions or mistakes that are definitely bound to stop the PCB from functioning properly. Beyond the functionality of the PCB is the safety of people nearby the PCB when it is supplied with power, as the DRC errors also include short-circuits, which rapidly heat-up the PCB and its components, causing the board to catch fire or even explode - casualties guaranteed. All this being said, a screenshot of a successful DRC run with zero errors in the PCB is shown in figure 4.3.

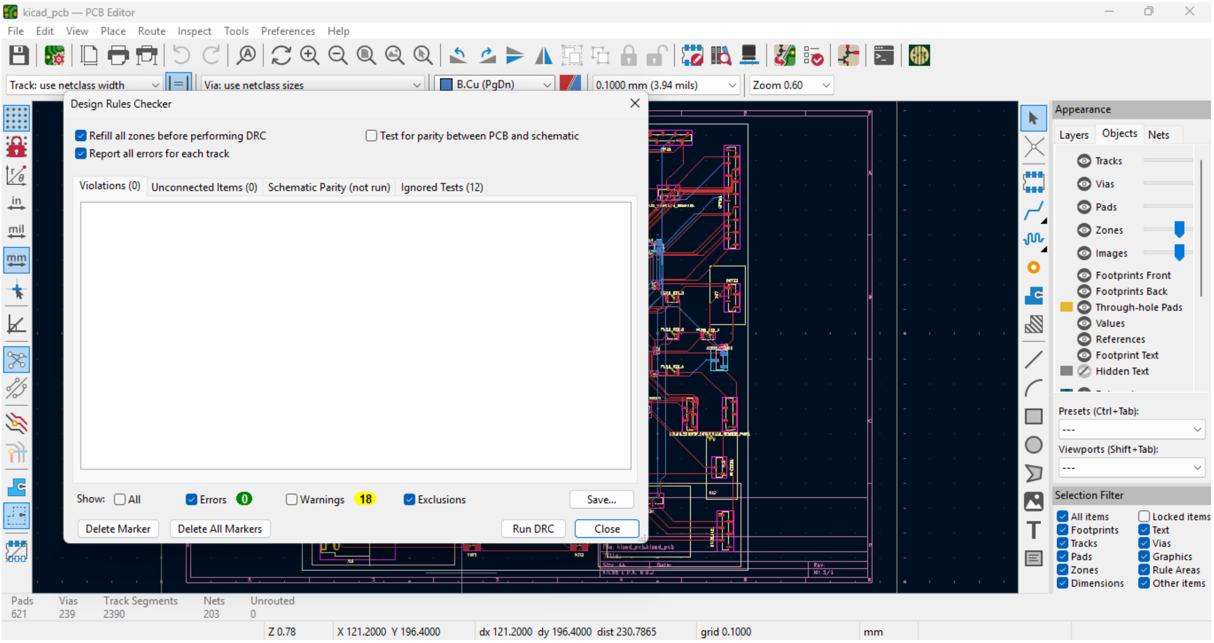

Figure 5.3: PCB successful DRC run

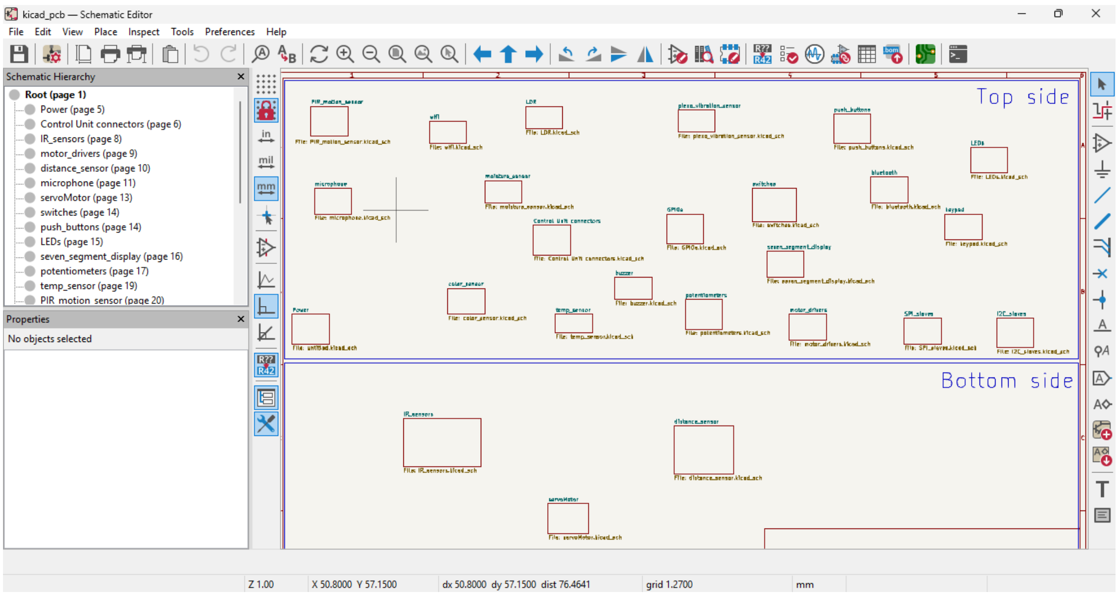

Figure 5.4: Hierarchical schematics: root

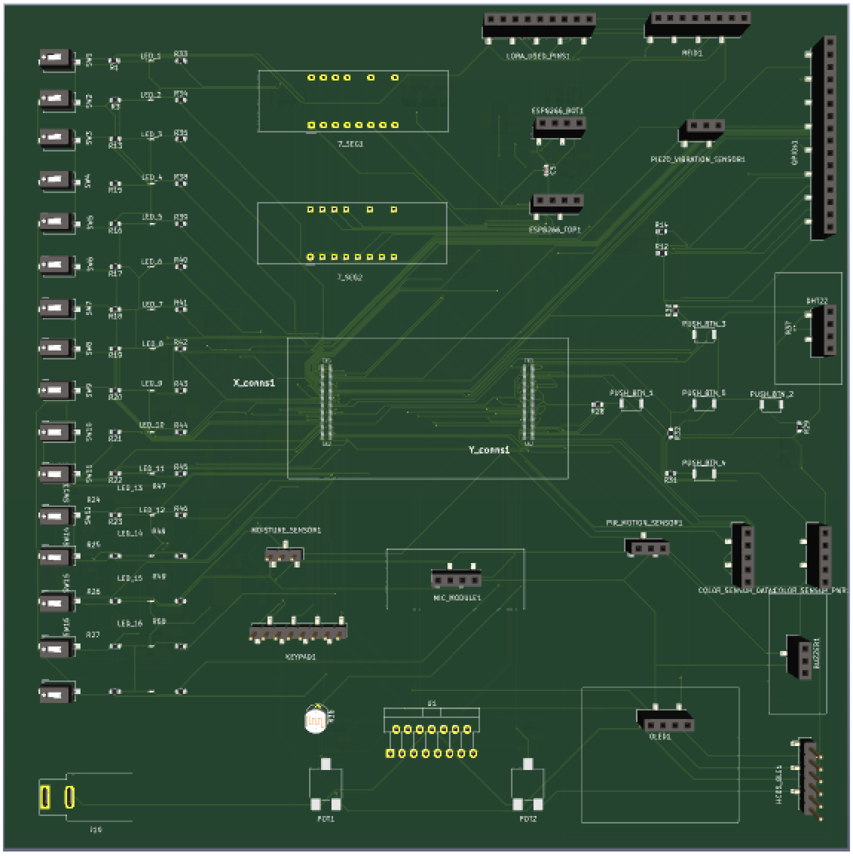

Figure 5.5: PCB 3D top view

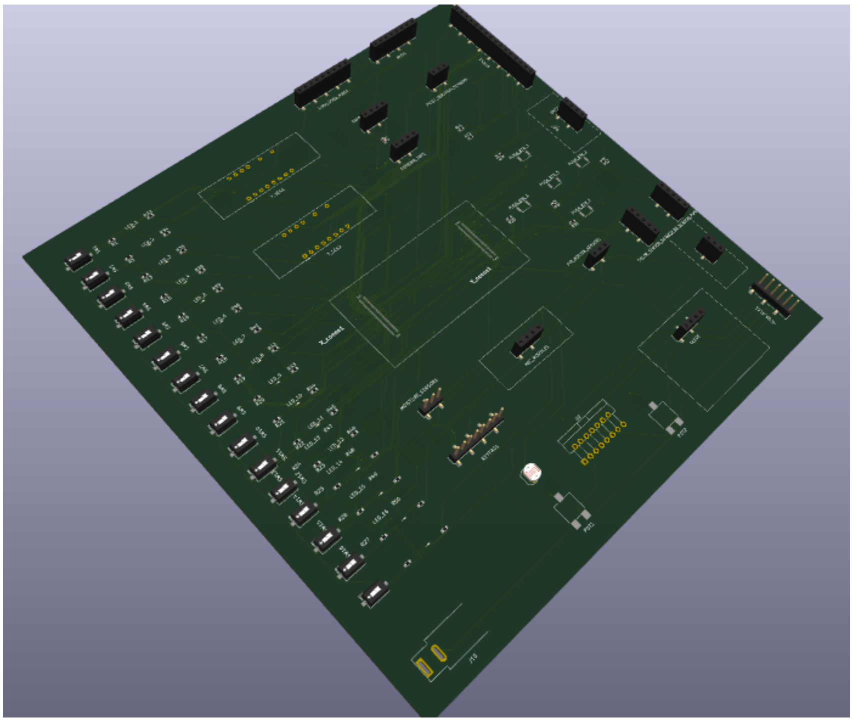

Figure 5.6: PCB 3D isometric view

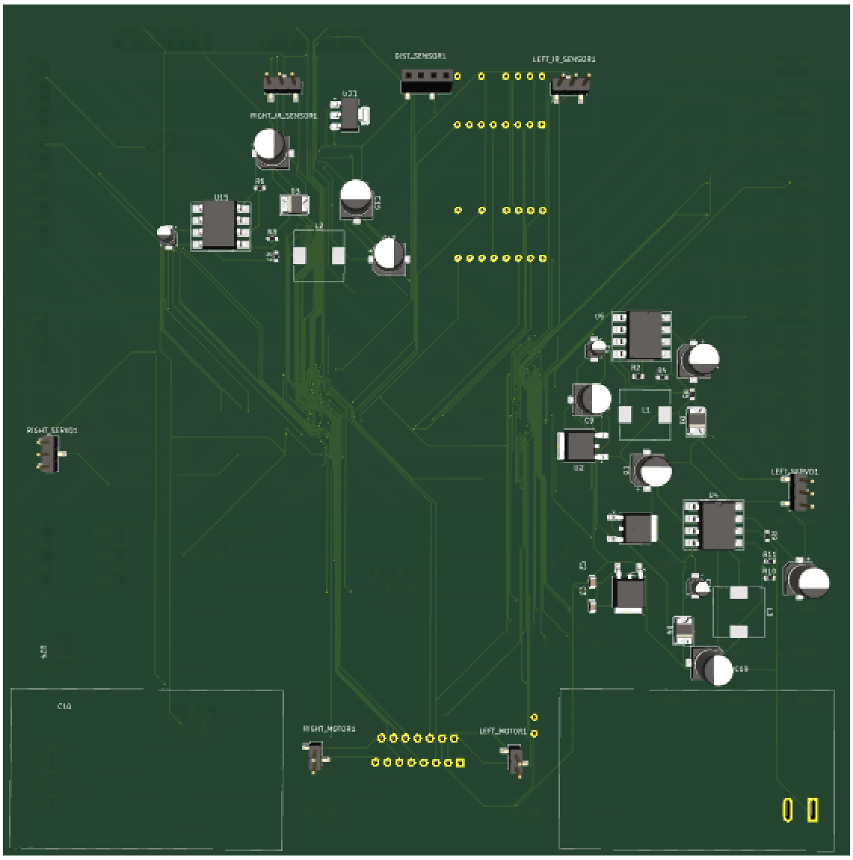

Figure 5.7: PCB 3D back view

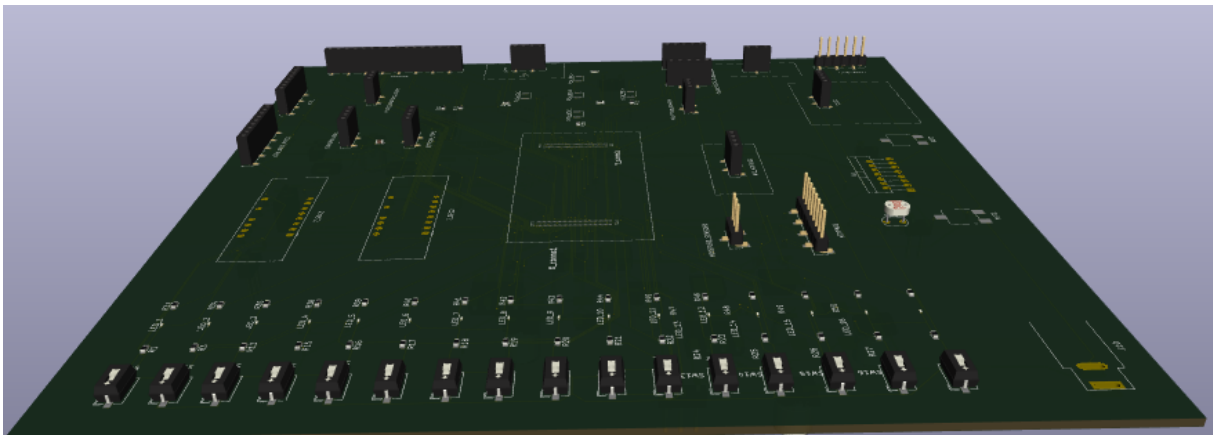

Figure 5.8: PCB 3D side view

6 Peripherals

This chapter includes descriptions about some of the peripherals for which there are connectors on the PCB.

6.1 Seven segment display: LTC-4627JD

Figure 6.1: Seven segment display: LTC-4627JD [10]

A common anode seven-segment display module for limited alphanumeric and numerical character representation is the LTC-4627JD. This four-digit display, which is produced by Lite-On, is helpful in applications where it is necessary to clearly display numerical data. Digital clocks, counters, measuring devices, and embedded systems that need a straightforward but efficient visual output frequently use it. It can be used in a range of settings since the red LED segments offer good visibility [4].

The LTC-4627JD has seven distinct LED segments for each digit, numbered A through G, and an optional decimal point. The arrangement of these LEDs enables the display of the digits 0 through 9 as well as a few letters as necessary. All of the LEDs' anode terminals inside a digit are internally connected in a common anode display, which means that a positive voltage must be applied to the display while the individual cathodes are controlled to light up the necessary segments. To stop excessive current flow that can harm the LEDs, the right current-limiting resistors must be employed. Each segment's forward voltage is normally around 2.0V, and for best brightness and lifespan, a current of roughly 10–20mA per segment is advised [4].

Depending on the particular version, the display includes either 12 or 16 pins, which include connections for the various segment cathodes and the common anode. Multiplexing is a widely used technique to show numerals over all four digits. Multiplexing turns on the necessary segments one digit at a time rather than regulating each segment separately. Rapid repetition of this operation gives the appearance that every digit is lighted at once. This technique guarantees effective operation while drastically lowering the number of control pins required [4].

Using general-purpose input/output pins, microcontrollers like Arduino, Raspberry Pi, or PIC can communicate with the LTC-4627JD. Driver integrated circuits (ICs), such as the 74HC595 shift register or the MAX7219 LED driver, are frequently used to simplify wiring and programming because manually manipulating numerous digits can be complicated. These integrated circuits (ICs) offer extra capabilities like brightness adjustment and enable the display to be operated with fewer pins. Developers can more easily include the display into their designs by using software libraries that manage segment control and multiplexing automatically, like the SevSeg library for Arduino [4].

6.2 Servo motor: SG90

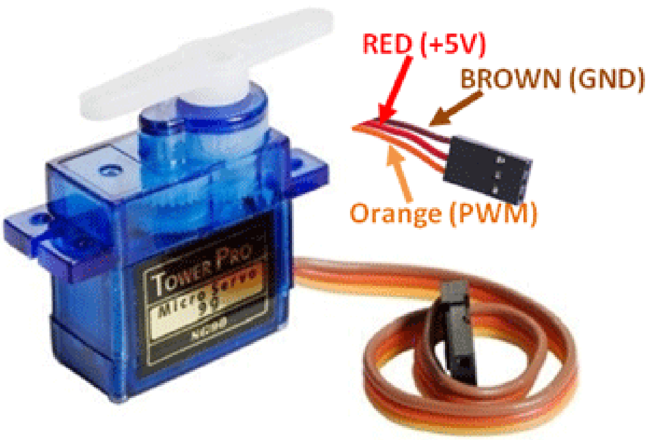

Figure 6.2: Servo motor: SG90 [1]

Robotics, automation, and remote-controlled applications frequently use the SG90, a compact and lightweight servo motor. Because to its low cost, simplicity of use, and compatibility with microcontrollers such as Arduino, Raspberry Pi, and PIC, it is one of the most often used servo motors. Projects requiring precise angular movement, such automated mechanisms, pan-tilt camera systems, and robotic arms, can benefit from the motor's straightforward control mechanism [4].

Because it is a micro servo, the SG90 is small and made for low-power uses. Under usual operating conditions, it draws a current of approximately 100–250mA and runs on a conventional voltage range of 4.8V to 6V. A small DC motor, a gear reduction system, and a position feedback system are all part of the motor's internal circuit. Based on the input control signal, the motor may precisely alter its position thanks to the feedback system, which makes use of a potentiometer [4].

A pulse width modulation (PWM) signal serves as the basis for controlling the SG90 servo. The width of the PWM signal transmitted to the servo shaft's control wire determines its position. An average SG90 servo reacts to a signal with a frequency of about 50 Hz and a pulse width of between one and two milliseconds. The shaft is moved to one extreme (0 degrees) by a 1 ms pulse and to the opposite extreme (180 degrees) by a 2 ms pulse. The servo is positioned at the middle, or 90 degrees, with a pulse width of around 1.5 ms. The motor may be rotated to any angle within its operational range by continually varying the pulse width [4].

A series of plastic gears that enhance torque and decrease rotation speed are attached to the motor's output shaft. With a torque of about 1.8 kg.cm at 5V, the SG90 offers enough force for lightweight applications even if its torque is less than that of bigger servos. Although they are less resilient to large loads, the lightweight plastic gears make it quieter than metal gear servos [4].

6.3 Distance sensor: HC-SR04

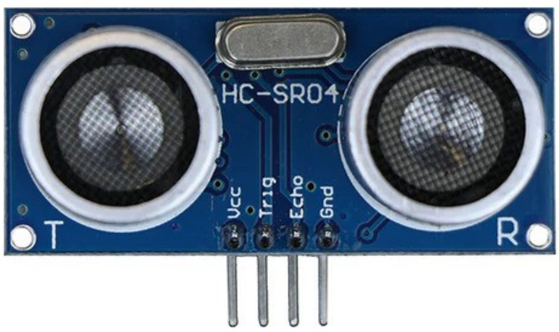

Figure 6.3: Ultrasonic distance sensor: HC-SR04 [9]

An ultrasonic proximity sensor called the HC-SR04 is frequently used in robotics, automation, and obstacle detection applications to measure distance. In order to calculate the distance to an item, it sends out ultrasonic waves and measures how long it takes for the echo to return. This sensor is a popular option for projects involving environmental sensing, navigation, and object detection because of its dependability, cost, and ease of use [4].

The transmitter and the receiver are the two primary parts of the sensor. At a frequency of 40 kHz, the transmitter sends out ultrasonic pulses that travel through the atmosphere until they strike an item and bounce back toward the receiver. Based on the speed of sound in air, which is roughly 343 meters per second, the distance is determined by measuring the time it takes for the pulse to reach the target and return. Distance = (Time $\times$ Speed of Sound) / 2 is the formula used to calculate the distance; the division by 2 takes into consideration the ultrasonic wave's round-trip travel [4].

The HC-SR04 uses relatively little current—usually about 15mA—and runs on a 5V power source. It is appropriate for applications requiring accurate distance measurements within a short to medium range because of its detection range of 2 cm to 400 cm and accuracy of roughly 3 mm. A microcontroller like an Arduino, Raspberry Pi, or ESP32 can readily read the sensor's output, which is a pulse width that reflects the distance measured [4].

Only four pins are needed for easy interface with the HC-SR04: VCC, GND, Trigger, and Echo. A microcontroller first sends a brief 10-microsecond pulse to the Trigger pin in order to detect distance. The sensor emits an ultrasonic burst as a result. The Echo pin generates a strong signal for a duration proportional to the distance after the echo is received. This duration is measured by the microcontroller, which then computes the distance. With the help of libraries offered by numerous microcontroller systems, developers may read distance information with little coding work [4].

Even with all of its benefits, the HC-SR04 has certain drawbacks. It may have trouble detecting items with soft or very absorbent surfaces that do not reflect ultrasonic signals well, and it needs a clear line of sight to do so. Environmental elements like humidity and temperature might also have a little impact on measurement accuracy. It does, however, offer a dependable and affordable distance sensing solution for the majority of real-world applications [4].

6.4 OLED: SSD1306

Popular OLED display drivers like the SSD1306 are frequently found in small monochrome OLED screens. It has several communication interfaces, including as SPI and I2C, and the I2C version is more popular because of its effective communication and straightforward wiring. Wearables, embedded devices, and Internet of Things applications that demand a small, low-power graphical output frequently use the display. With its clear images and great contrast, it's perfect for showing text, icons, and basic graphics [4].

Although there are smaller versions like 128x32 pixels, the SSD1306-based OLED display often has a resolution of 128x64 pixels. Because each pixel in an OLED emits its own light, they do not require a backlight like standard LCDs do. Particularly when displaying mostly dark information, this leads to deep blacks, outstanding contrast, and reduced power usage. With a voltage range of 3.3V to 5V, the display can be used with a variety of microcontrollers, such as the Arduino, ESP32, and Raspberry Pi [4].

Figure 6.4: OLED: SSD1306 [11]

In addition to power (VCC) and ground (GND), the I2C version of the SSD1306 display only needs two data lines: SDA (data) and SCL (clock). As a result, interacting with microcontrollers is simple and requires little wire. Libraries like the Adafruit SSD1306 library for Arduino or the MicroPython SSD1306 module for ESP8266 and ESP32 are commonly used to operate the display. These libraries make it easier to convey graphical components, text, and images to the screen [4].

The SSD1306 can draw text, lines, rectangles, circles, and bitmaps, among other display operations. Pixels can be independently manipulated for unique graphical outputs, and text can be shown in a variety of font sizes. Additionally, certain models have partial screen scrolling capabilities, and the display may be reversed to switch between black and white. The display's low refresh rate makes it unsuitable for fluid animations, but it performs admirably for static or slowly changing material [4].

6.5 IR sensor: YL-63

An infrared (IR) sensor called the YL-63 is frequently used in robotics, automation, and security systems for proximity and object detection. Infrared light is emitted, and the reflected signal from surrounding objects is detected. Because it is inexpensive, simple to use, and can detect objects without physical contact, this sensor is extensively utilized. It is perfect for line-following robots, obstacle identification, and motion sensing [4].

An infrared transmitter (IR LED) and an infrared receiver (phototransistor or photodiode) make up the sensor's two primary parts. The receiver picks up the infrared light that the transmitter releases after it bounces off an item. The presence and approximate distance

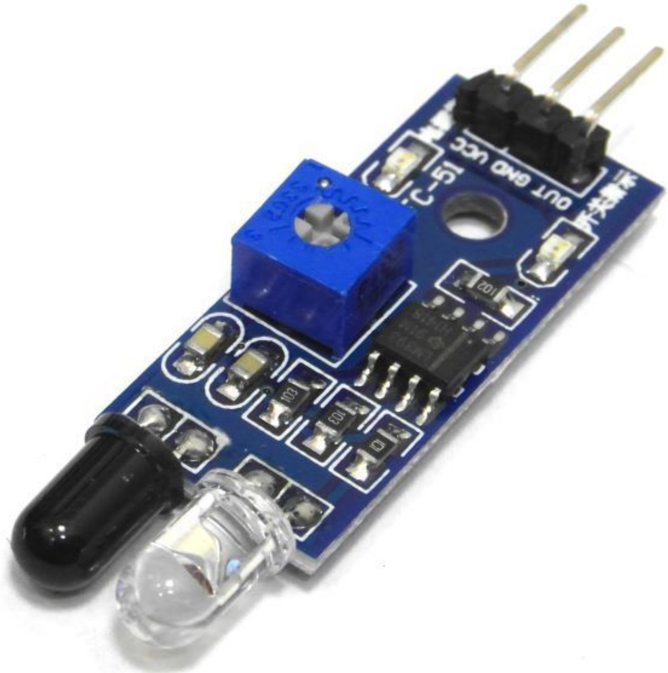

Figure 6.5: IR sensor: YL-63 [6]

of an item are determined by the intensity of the reflected signal. Dark or extremely absorbent materials that do not reflect infrared light well may be difficult for the sensor to detect, but it is especially good at identifying items with reflecting surfaces [4].

With a voltage range of 3.3V to 5V, the YL-63 can be used with the majority of microcontrollers, such as the Arduino, Raspberry Pi, and ESP32. Because it includes both analog and digital outputs, it can be used in a variety of ways. For distance estimation, the analog output offers a variable voltage that varies with the intensity of the reflected infrared light. Conversely, an onboard potentiometer can be used to modify the threshold at which the detected reflection surpasses, which triggers the digital output [4].

It is easy to interface the YL-63 with a microcontroller. Power (VCC), ground (GND), analog output (AO), and digital output (DO) connections are necessary. Depending on whether an object is detected, the sensor's digital output can be either HIGH or LOW. For applications needing incremental distance measurement, a microcontroller's ADC (Analog-to-Digital Converter) can read the analog output to provide more accurate readings [4].

6.6 PIR motion sensor: HC-SR501

Passive infrared (PIR) sensors like the HC-SR501 are utilized for energy-saving, automation, and security system motion detection. It is perfect for applications like motion-triggered devices, intruder alarms, and automatic lighting since it senses variations in infrared radiation to detect movement. Because of its inexpensive cost, simplicity of integration, and consistent ability to detect movement in both humans and animals, this sensor is

Figure 6.6: PIR motion sensor: HC-SR501 [12]

frequently utilized [4].

The HC-SR501 works by sensing infrared radiation that warm objects, like people and animals, release. It is made up of a Fresnel lens that directs infrared light onto a pyroelectric sensor. An output signal is produced when an object with a different infrared signature travels inside the sensor's detecting range, changing the infrared levels that are observed. It is an energy-efficient passive sensor because it simply detects infrared radiation that is already there and doesn't emit any radiation [4].

The sensor is compatible with the majority of microcontrollers, such as Arduino, Raspberry Pi, and ESP32, because it operates in the voltage range of 4.5V to 20V. It is simple to interface with controllers because it provides a digital output (HIGH or LOW). The output (usually 3.3V or 5V) swings HIGH when motion is detected and returns to LOW after a certain amount of time. Because of its built-in voltage regulator, the sensor can operate on a variety of power sources [4].

The HC-SR501 has two potentiometers that can be adjusted to customize its behavior. The detection range, which can be set between 3 and 7 meters, is controlled by the sensitivity adjustment. The time delay adjustment, which ranges from 0.3 seconds to 5 minutes, determines how long the output stays HIGH after motion is detected. Because of its adaptability, the sensor may be used for a range of tasks, including long-duration warnings and quick-triggering systems [4].

Additionally, the sensor has two trigger modes: retriggerable and non-retriggerable. In non-retriggerable mode, even if motion persists, the sensor only produces a HIGH signal once and stays LOW after the delay period. When in retriggerable mode, the timer is reset with each movement and the output stays HIGH as long as motion is detected. Applications where a system should remain active through constant mobility will benefit from this mode [4].

6.7 Temperature and humidity sensor: DHT22

A common digital temperature and humidity sensor in HVAC systems, environmental sensing, and weather monitoring is the DHT22. It uses a thermistor and an integrated capacitive humidity sensor to offer precise temperature and humidity readings. The DHT22

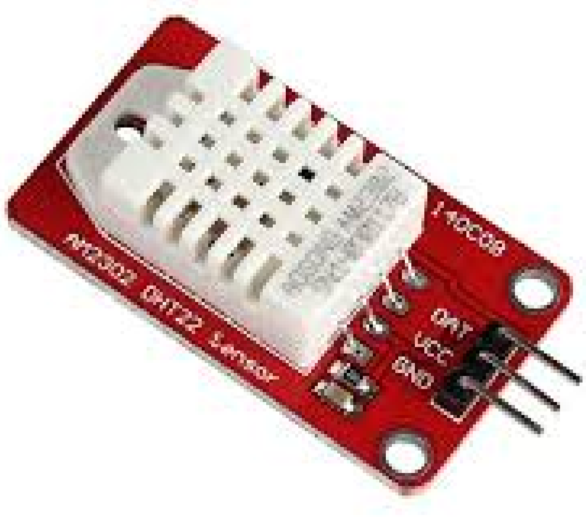

Figure 6.7: Temperature and humidity sensor: DHT22 [13]

is appropriate for applications requiring precise environmental data since it has more accuracy, a larger measurement range, and improved stability than its predecessor, the DHT11 [4].

The sensor is compatible with the majority of microcontrollers, such as Arduino, Raspberry Pi, and ESP32, because it runs on a 3.3V to 5.5V power supply. It uses a single-wire digital interface for communication, which makes wiring easier and requires fewer connections. The sensor transmits 40 bits of data, which indicate temperature and humidity levels, coupled with a checksum for data validation, according to a predetermined time protocol [4].

The DHT22 has an accuracy of ±2 % Relative Humidity (RH) and measures humidity between 0% and 100% RH. It makes use of a capacitive humidity detecting element, which uses fluctuations in capacitance to detect changes in moisture levels. Because of this, it can give accurate and consistent humidity measurements even in high-humidity settings. It may be used both indoors and outdoors because of its -40°C to 80°C temperature measurement range and ±0.5°C accuracy. With a sampling rate of 0.5 Hz, the sensor updates its data every two seconds [4].

7 Evaluation

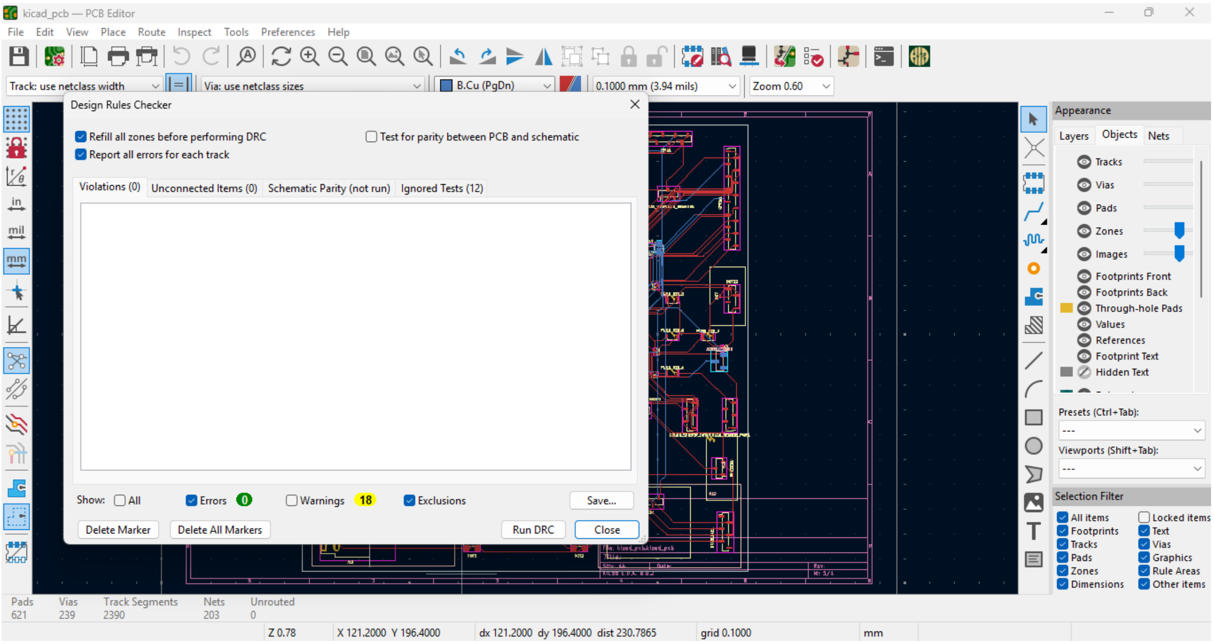

Figure 7.1: PCB DRC successful run

After the PCB design was done, multiple evaluation steps were carried out. The first step was to run the DRC tool in the KiCAD PCB design window. As seen in figure 7.1, there are 0 errors detected in the PCB design evaluation. There were, however, some warnings. Included in these warnings, for example, were warnings related to the silkscreen layer, where some labels either overlapped with each other, or got too close to each other or to the edge of the board. These warnings were ignored currently due to shortage of time to finish this project work.

Another step in the evaluation of the PCB was to verify whether the high density connectors would actually fit when an Arduino Portenta H7 is placed on it. This was done by taking the documentation of the Arduino Portenta H7 as reference, where they illustrate and label the dimensions of the board, including the horizontal margin or gap between the two high density connectors on the Arduino Portenta H7 that would connect with those on the PCB of this project work.

As shown in figure 7.2, the exact dimensions of the Arduino Portenta H7 are described, including for the high density connectors onboard: J1 and J2 [3]. The required information here for evaluation in the project work is the horizontal gap or distance between J1 and J2. As we can see from figure 7.2, J1 is 17.14 mm away from the left edge of the board, and J2 is 60.32 mm away from the left edge of the board. In order to find the distance between J1 and J2, we perform the calculation:

$$60.32 - 17.14 = 43.18$$

Therefore, we find that the horizontal gap or distance between the two high density connectors in my PCB design for my project work must be 43.18 mm. This is then verified and shown as evidence in figure 7.3, where a line in drawn in the User Drawings layer and its width is shown to be exactly 43.18 mm, as calculated above. In addition, the vertical positioning or the Y-axis value for both the connectors should also be the same, in order for the connectors to line up in an accurate, absolute straight line. This is also proven to be the case as seen in figure 7.3.

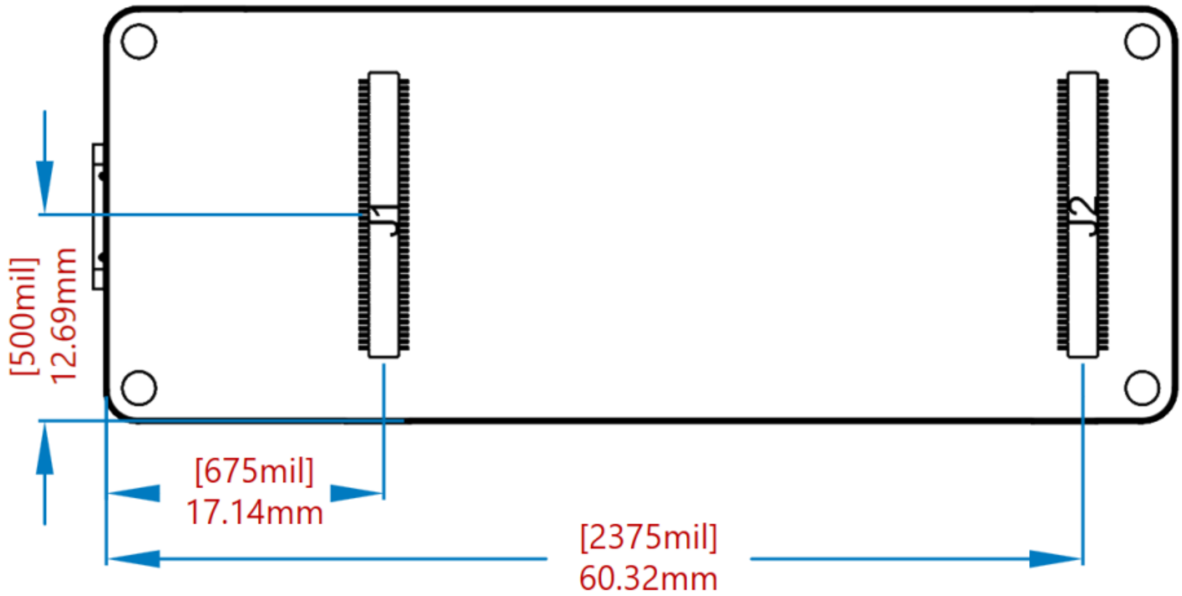

Figure 7.2: Screenshot from the datasheet of Arduino Portenta H7 showcasing the positioning of its high density connectors [3]

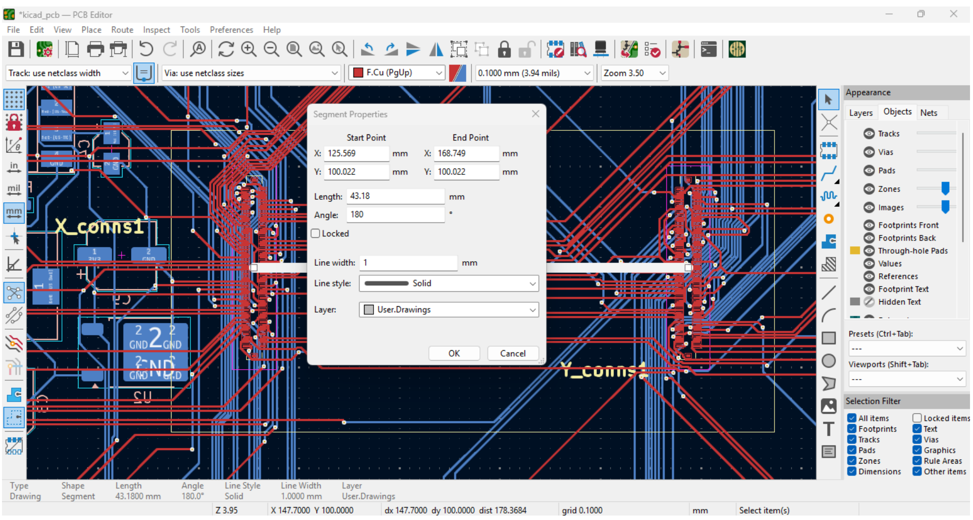

Figure 7.3: Verified the gap between high density connectors with user drawn line properties

Finally, the electrical connections implemented in the schematic and PCB design were done so according to the specifications of the electronic components used. For example, using the HC-SR04 proximity sensor required making the exact connections as stated in its datasheet. Going into more detail and being more specific, the distance sensor is actually intended to be place upside down, facing forwards. The exact electrical connections that would be required in this kind of setup were then used in the PCB. Another example is the use pull-down resistors for the push buttons and switches used in the PCB. The pull-down resistors reliably maintain a constant 0 or LOW signal when a push button or switch is not pushed or turned on, meaning that there would be no floating values or uncertain signals.

8 Future work

While the PCB for this project work was implemented keeping in mind the requirements defined at the beginning, there were still some shortcomings - room for improvement is definitely found. For example, the warnings found when running the DRC tool could be fixed after investing some more time for this project work. Another way this project work could be improved is by using more than just 2 layers in the PCB design. While the Top and Bottom layers are used in an acceptable way that allows the PCB to function, the traces or tracks in the PCB could definitely be less dense or tightly placed with more layers. Assuming a 6 layer design, there could also be a dedicated ground layer, two layers for VIN and 3.3V, and even for the SCL and SDA signals used in I2C communication. Overall, the PCB works and looks fine, but with these improvements, it would undoubtedly be rated higher than it is right now.

9 Overview of tools used

1. **ChatGPT**:

   The web link to the exact prompts and the entire conversation in the chat can be found within the references section. The prompts used were of varying nature, such as for writing the text in the description of the different peripherals in the PCB. The text provided by ChatGPT was paraphrased accordingly in order to not do plagiarism. While the author of this paper will attempt to include as many explicit citations of the chat link in paragraphs as possible, the reader of this document should note that the entire document has some level of direct or indirect help from ChatGPT. Therefore, this document is the result of a combined effort from the author and AI tools for assistance and saving time.

2. **Gemini Pro**:

   The web link to the exact prompts and the entire conversation in the chat can be found within the references section. The prompts used were of varying nature, such as for writing the text in the introduction, related work, explaining the code the author has written, etc. The text provided by Gemini Pro was paraphrased accordingly in order to not do plagiarism. While the author of this paper will attempt to include as many explicit citations of the chat link in paragraphs as possible, the reader of this document should note that the entire document has some level of direct or indirect help from Gemini Pro. Therefore, this document is the result of a combined effort from the author and AI tools for assistance and saving time.

3. **ZeroGPT**:

   This is a paraphrasing tool used to rewrite AI-generated content from platforms like ChatGPT and Gemini Pro. The use of such a paraphrasing tool is to avoid doing plagiarism.

Bibliography

[1] C. 101. Servo motor sg-90.

[2] E. . All. Obstacle avoidance robot.

[3] Arduino. Arduino portenta h7.

[4] O. ChatGPT. Project work.

[5] Digikey. Df40c-80ds-0.4v(51).

[6] A. Geek. Yl-63 ir sensor.

[7] Hackster. Line follower robot.

[8] M. in China. New wa380 wheel loader front end loader payloader construction mining equipment price.

[9] Kitronik. Ultrasonic distance sensor hc-sr04 3-5v version.

[10] R. E. LITEON. 10x4 pa-l 10mm anód piros 4-digit ltc4627jd - liteon - ltc-4627jd.

[11] Majju. 0.96 inch oled display ssd1306 oled i2c 128x64.

[12] A. T. Notch. Hc-sr501 pir motion sensor detector module.

[13] Voltaat. Dht22 temperature & humidity sensor.

Affidavit

I, Shahzaib Waseem, hereby declare that I am the author of the Project Work titled "Peripherals and robotics PCB for control unit boards", submitted to Hochschule Hamm-Lippstadt, as part of the requirements for the degree of B.Eng. Electronics Engineering.

I affirm that this project work represents my original work and has been completed in compliance with the academic regulations and ethical standards of Hochschule Hamm-Lippstadt. All sources of information, data, and references used in this thesis have been appropriately acknowledged and cited in the bibliography.

I also declare that this project work has not been previously submitted in full or in part for the award of any other degree, diploma, or qualification at any other university or institution.

I further declare that, during the preparation of this project work, I made use of advanced digital tools, including artificial intelligence (AI), to assist with certain aspects such as language refinement, grammar checks, and technical explanations. These tools were utilized solely to enhance the quality and clarity of my work and to save time, and not to replace the originality or authenticity of the research presented. I retain full responsibility for all content, ideas, and analyses provided in this document.

By signing below, I confirm that I have adhered to the rules and guidelines set forth by Hochschule Hamm-Lippstadt and accept full accountability for the content and conclusions of this project work.

Shahzaib Waseem
2210012
Date and signature: 04.02.2025