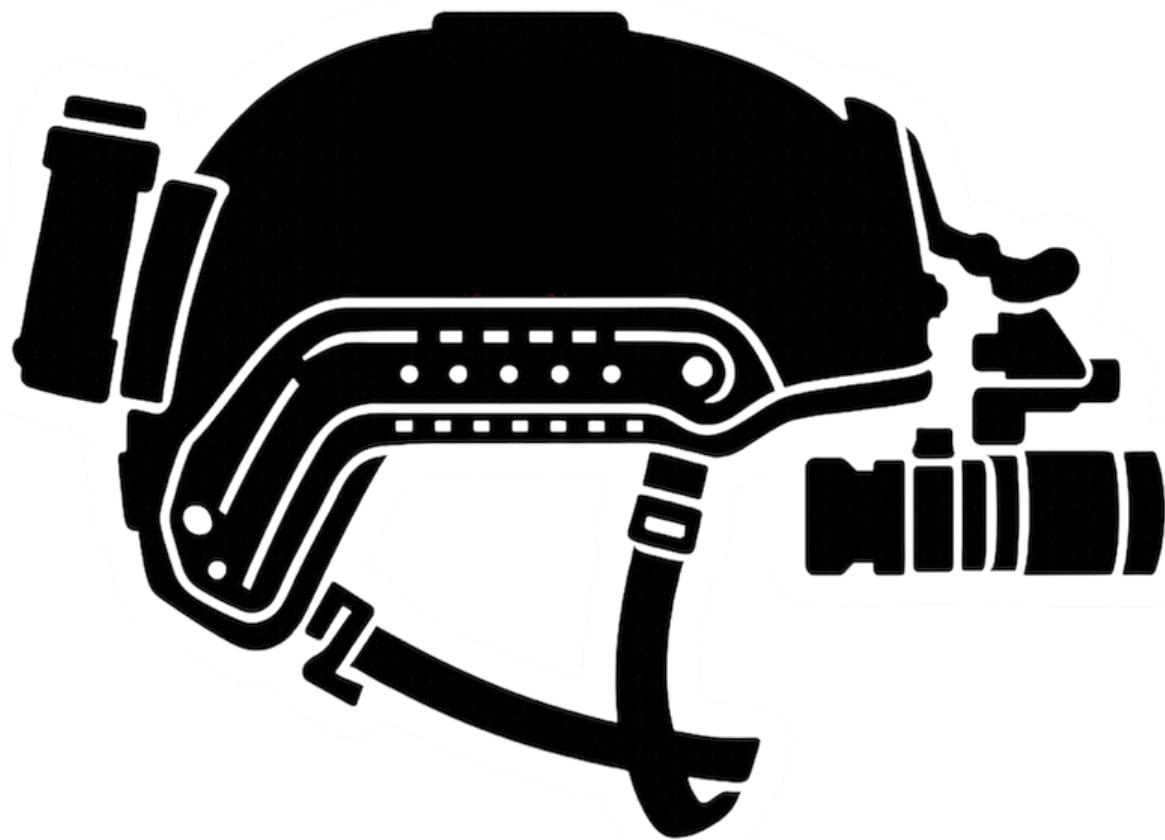


Project Documentation
Digital Night Observation Device

Autor: Odin Hasson, Benjamin Schalk
Datum: 17.6.2025



1 Introduction

1.1 Group members

Odin Hasson



As the team leader, I oversaw the allocation of team members, ensuring that each individual was assigned tasks best suited to their expertise and the project's needs. My primary technical contributions centered on the hardware development of our project, with a particular focus on designing the foundational circuitry. Additionally, I collaborated closely with Benjamin Schalk in the co-design of the PCB, ensuring both functionality and efficiency. Beyond these core responsibilities, I provided support across various aspects of the design process and took sole responsibility for all manufacturing and implementation related to 3D printing and 3D printed components.

Benjamin Schalk



In this project, my primary responsibility was to develop the software. In addition to the software development, I actively contributed to the hardware development, collaborating closely with Odin Hasson. I contributed heavily to the design of the PCB.

Figure 1: Benjamin Schalk

1.2 Project Overview

The Digital Night Observation Device (henceforth referred to as “DNOD” in this document) consists, at its most fundamental level, of a microcontroller, a digital display, and a NoIR camera. The microcontroller utilizes an AI chip (also known as a KPU) to process video and perform facial recognition. The camera is a simple digital unit that outputs video directly to the microcontroller board. The board processes the video signal and forwards the modified signal to the display.

2 Design

2.1 High Level Architecture (Hardware)

Figure 2 visualizes the high-level hardware architecture. Light blue boxes represent hardware components within the Sipeed M1W module, while light yellow boxes indicate input and output elements. Note that the diagram does not show the entire system: subcircuits are required to support each I/O element and the M1W module itself are omitted for clarity. Additionally, the battery used to power DNOD is not shown.

Figure 2: High level architecture (Hardware)

2.2 High Level Architecture (Software)

Figure 3 depicts the high-level software architecture. The reference config stores important const values such as image ratio, color depth, and other important information. In the depiction, blue boxes reference functions, and excluding the initial reference to the config, the rest of the functions loop. Using the config the user can easily adjust settings such as bitrate, which version of the MLM¹ should be used, aswell as graphical settings such as upscaling or edge sharpening in order to increase visual quality, or sacrifice visual quality for performance.

Figure 3: High level architecture (Software)

¹Machine Learning Model

2.3 Project Housing

2.3.1 PCB, Camera, and Display Housing

We decided to place the PCB, camera, and display within the same housing to save space. The housing effectively “stacks” each core component vertically. Each individual component is separated from the others to ensure proper isolation. The display and the camera are connected via 24 Pin flat ribbon cable connectors. Additionally, the case includes a dovetail, so that the device can be mounted to a helmet. The software Shapr3D and Fusion was used to design the housing



Figure 4: PCB, display, and camera housing when viewed from the front



Figure 5: PCB, display, and camera housing when viewed from the rear

2.3.2 Battery Housing

The battery housing is quite simple, it contains a slot for a DC jack port and, at the bottom, small dovetails onto which the battery connectors can slide.

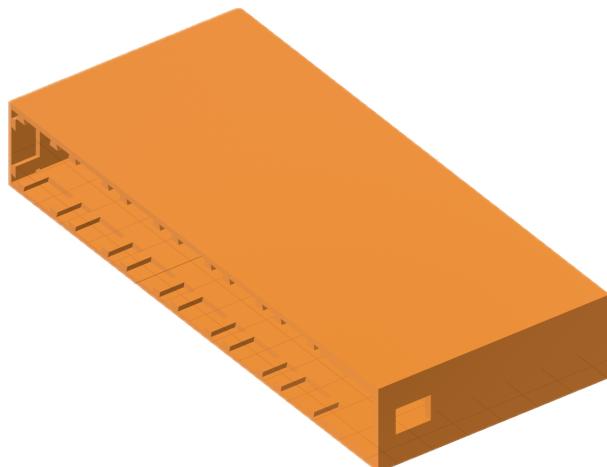


Figure 6: Battery housing

2.4 Circuit Design

Our base circuit is derived from elements of the SIPEED Maixduino, a microcontroller built around the SIPEED M1 module. The SIPEED Maixduino is well-documented, and its complete circuit schematic is readily available online. Utilizing both the official documentation and our own technical knowledge, we designed an alternative version that retains and enhances the specific features of the Maixduino relevant to our application.

2.5 Circuit Components

2.5.1 K210 Circuit

The K210 circuit forms the core of the processing system, incorporating the AI-capable microcontroller and its supporting components. See page 9 for the schematic overview.

2.5.2 LCD Connector Circuit

The LCD connector circuit interfaces with the display module, handling both data transmission and power delivery. See page 10 for the schematic overview.

2.5.3 Camera Connector Circuit

The camera connector circuit provides the interface between the NoIR camera module and the main processing unit. This circuit ensures proper signal integrity and power delivery to the camera module. See page 11 for the schematic overview.

2.5.4 USB-C Port Circuit

The USB-C port circuit handles both power input and data communication with external devices. See page 12 for the schematic overview.

2.5.5 GPIO Circuit

The General Purpose Input/Output (GPIO) circuit provides expandable connection points for additional sensors or peripherals. See page 13 for the schematic overview.

2.5.6 SD Card Circuit

The SD card circuit provides expandable storage capabilities for the device, allowing for data logging and firmware storage. See page 14 for the schematic overview.

2.5.7 DC Power Circuit

The DC power circuit handles the main power distribution throughout the device, ensuring all components receive appropriate voltage levels. See page 15 for the schematic overview.

2.5.8 Camera Power Circuit

The camera power circuit manages the power supply requirements for the camera module, providing stable voltage regulation and filtering to ensure optimal image quality. See page 16 for the schematic overview.

2.5.9 ISP Download

This circuit allows in-system programming of a microcontroller via UART using an ESP32. See 17 for the schematic overview.

2.5.10 DAC Circuit

The Digital-to-Analog Converter (DAC) circuit enables the conversion of digital signals from the microcontroller to analog outputs when required. See page 18 for the schematic overview.

2.5.11 ESP-32

The ESP-32 was originally included as we intended to include Bluetooth/WiFi capabilities to our projects, but couldn't achieve that goal due to time constraints. See page 19 for the schematic overview

2.5.12 Miscellaneous Circuits

Additional supporting circuits provide various auxiliary functions such as status indicators, reset functionality, and system monitoring. See page section 2.5.12 for the schematic overview.

width=!,height=!, pages=1, scale=0.90, pagecommand=, fitpaper=true

width=!,height=!, pages=2, scale=0.90, pagecommand=, fitpaper=true

width=!,height=!, pages=3, scale=0.90, pagecommand=, fitpaper=true

width=!,height=!, pages=4, scale=0.90, pagecommand=, fitpaper=true

width=!,height=!, pages=5, scale=0.90, pagecommand=, fitpaper=true

width=!,height=!, pages=6, scale=0.90, pagecommand=, fitpaper=true

width=!,height=!, pages=7, scale=0.90, pagecommand=, fitpaper=true

width=!,height=!, pages=8, scale=0.90, pagecommand=, fitpaper=true

width=!,height=!, pages=9, scale=0.90, pagecommand=, fitpaper=true

width=!,height=!, pages=10, scale=0.90, pagecommand=, fitpaper=true

width=!,height=!, pages=11, scale=0.90, pagecommand=, fitpaper=true

width=!,height=!, pages=12, scale=0.90, pagecommand=, fitpaper=true

2.6 PCB Design

To design our PCB, we used the software EasyEda Pro. We did two iterations, We made two iterations, the second iteration is depicted here.

2.6.1 Top Layer

The layout of the top layer can be found in section 2.6.2. It illustrates the primary routing and placement of components on the upper surface of the PCB.

2.6.2 Bottom Layer

The layout of the bottom layer is shown in section 2.6.2, highlighting the ground planes and any signal routing on the reverse side.

width=!,height=!, pages=1, landscape=true, scale=0.8, pagecommand=, fitpaper=false
width=!,height=!, pages=2, landscape=true, scale=0.8, pagecommand=, fitpaper=false

3 Production

3.1 Programming

The Sipeed M1 K210 module can be flashed using the software known as KFlash, and then, using Sipeed's proprietary

3.1.1 SmartFaceDisplay 1.0

Our SmartFaceDisplay 1.0 is a compact solution for real-time face recognition with a camera, AI module, and display. As soon as the device is powered on, the camera starts and continuously sends images to the onboard "KPU" accelerator. The YOLO algorithm analyzes each frame for faces and automatically overlays detected areas with colored bounding boxes. The final image, including these markings, is displayed live on the screen, while the current frame rate (FPS) is shown in the corner.

Core Features

- **Automatic Face Detection:** The AI scans each camera image for faces and draws rectangular frames around them.
- **Live Feedback:** Detection results and performance data (frames per second) are shown directly on the LCD.
- **Easy Startup:** Using the MaixPy IDE, the Python script can be quickly run on the K210 board; the model is loaded directly from flash memory at startup.

Optional Features

- **Image Filters (Convolution):** A sharpening or edge detection filter can optionally be activated (e.g., `img.conv3(edge)` or `img.conv3(sharp)`).
- **Upscaling Mode:** For more detailed displays, the image can be enlarged before rendering (`img.resize(384, 288)`), but this requires more RAM.
- **Configurable Thresholds:** Confidence parameters of the YOLO model can be adjusted during initialization to modify detection accuracy and sensitivity.
- **Alternative Models:** Instead of the preinstalled face model, other YOLO models (e.g., for detecting license plates, objects, or people in specific contexts) can also be loaded and ran.

All of these extras can be flexibly enabled or disabled by commenting or uncommenting a few lines in the script. This makes the SmartFaceDisplay 1.0 not only ready to use out of the box but also ideal for experimentation and adaptation to specific use cases — whether for visitor counting, access control, or as a demonstrator for AI-based image processing.

3.2 PCB Production

3.2.1 JLCPCB Manufacturing

The printed circuit board was manufactured in Hong Kong by **JLCPCBjlcpb2025**, components were sourced from **mouser.atmouser2025** and **LCSClsc2025**

3.2.2 Use of soldering oven

To solder the chips and smaller ICs, we used the soldering oven found in our workshop. To solder using the soldering oven, soldering paste must be applied to all contact points, before the chip is placed.

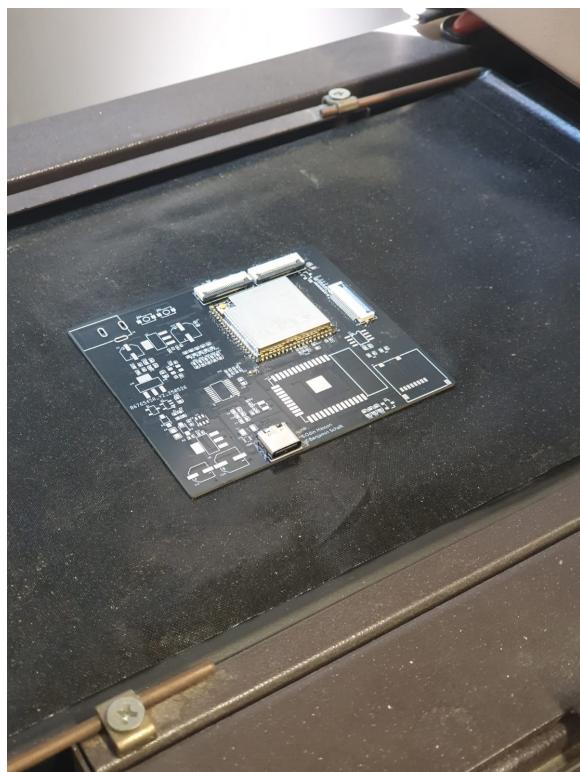


Figure 7: PCB being prepared for the soldering oven

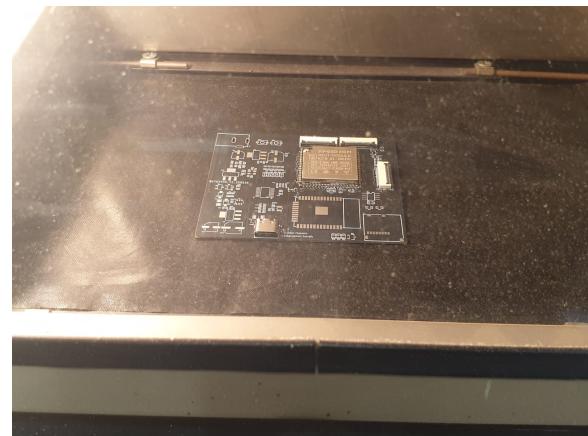


Figure 8: PCB in oven being soldered

3.2.3 Hand soldering

To solder the resistors and capacitors, we soldered by hand.

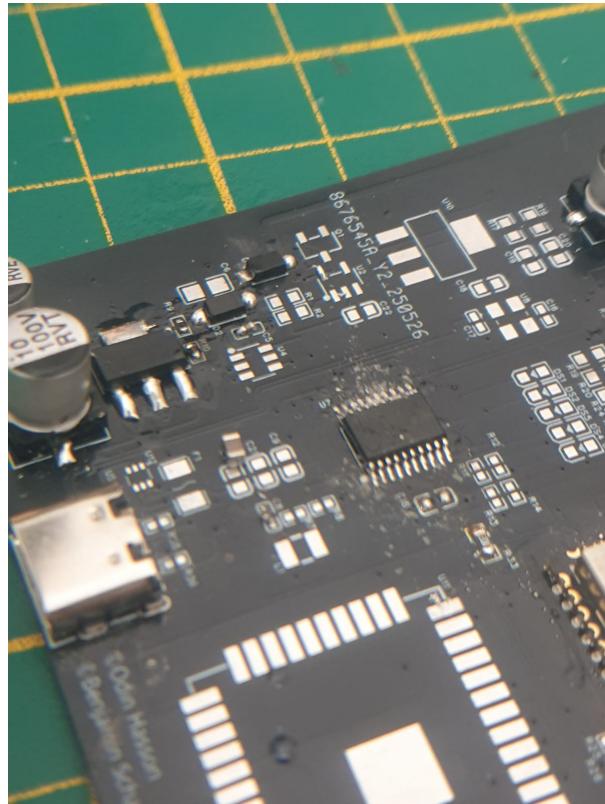


Figure 9: An example of some of the hand-done soldering.



Figure 10: Digital night observation device mounted to a helmet

4 Feedback

4.1 What went well

The design of all digital aspects such as circuits, software, and the PCB went well.

4.2 What went bad

1. Soldering the PCB was very difficult due to the small component size.
2. Certain things that we had ordered took forever to arrive and caused delays.

5 Final Product

Figure 10 shows our application of the digital night observation device. We mounted the battery to the back of the helmet, and the main board to the mounting point at the front of the helmet.

A Addendum: Daniel Liew

Originally, our group consisted of three members. Daniel Liew disappeared after the initial week of the summer semester and never returned. Since then, he has been officially expelled. As he only attended workshop once after the begin of the summer semester and was never seen again, we removed him from the development cycle. This was not done immediately; however, after being informed of his expulsion and receiving no response to our attempts to contact him, we decided to remove him from the project.

Used Software and Services

The following software tools and services were used during this project:

1. **EasyEDA Pro.** Available at: <https://easyeda.com/>. Accessed: 2025-06-17.
2. **Shapr3D.** Available at: <https://www.shapr3d.com/>. Accessed: 2025-06-17.
3. **Fusion 360.** Available at: <https://www.autodesk.com/products/fusion-360/>. Accessed: 2025-06-17.
4. **JLCPCB.** Available at: <https://jlcpcb.com/>. Accessed: 2025-06-17.
5. **LCSC Electronic Components Online.** Available at: <https://www.lcsc.com/>. Accessed: 2025-06-17.
6. **Mouser Electronics Austria.** Available at: <https://www.mouser.at/>. Accessed: 2025-06-17.