

# NERO SENSE - MASTER PLAN

## System Architecture & Future Roadmap



### Variante 1: Das Aktuelle System (Scanning Lidar)

#### Status: Operational

Ein Sensor auf einem Servo, der die Umgebung abtastet und ein detailliertes Bild auf dem Display zeichnet.

#### Verdrahtungsplan (Wemos D1 Mini)

Dies ist die finale Belegung für SPI-Display + I2C-Sensor + Servo.

Komponente	Pin	Wemos D1 Mini Pin	Funktion
<b>Display</b>	SCLK	D5 (GPIO 14)	SPI Clock
	MOSI	D7 (GPIO 13)	SPI Data
	CS	D8 (GPIO 15)	Chip Select
	DC	D3 (GPIO 0)	Data/Command
	RST	D0 (GPIO 16)	Reset
<b>Sensor</b>	SCL	D1 (GPIO 5)	I2C Clock
	SDA	D2 (GPIO 4)	I2C Data
<b>Servo</b>	Signal	D6 (GPIO 12)	PWM Signal
<b>Power</b>	VCC/GND	3.3V / GND	(Servo an 5V!)

#### Firmware Features (Final)

- **Cinematic Boot:** "Nero Robotics" Intro mit Glitch-Effekt.

- **Tron Visualization:** Verbundene Cyan-Linien (Wände) + Navy Schatten (Verdeckter Bereich).
- **Smoothing:** Low-Pass-Filter für stabile Linien.
- **Hybrid Network:** Statische IP `192.168.178.71` + Fallback auf Standalone.

## Der Prompt für Variante 1 (Backup)

Dieser Prompt erstellt die finale Firmware mit "Nero" Intro, verbundenen Linien und Schatten-Effekt.

```

Role: Senior Firmware Architect for ESP8266.
Project: NERO SENSE - Single Node Scanning Lidar.
Hardware: Wemos D1 Mini, GC9A01 Display (Hardware SPI), VL53L1X (I2C), Servo.

Pin Config (TFT_eSPI compatible):
- Display: SCLK=D5, MOSI=D7, CS=D8, DC=D3, RST=D0.
- Sensor: SCL=D1, SDA=D2.
- Servo: D6.

Task: Write the complete Arduino Firmware.

Features:
1. Boot Animation:
   - Procedural "Apple-style" intro. Center dot expands to a cyan ring. Text "NERO" (White) and
2. Radar Logic (The "Tesla View"):
   - Servo sweeps 0-180 degrees using millis() (non-blocking). Step size: 1 degree.
   - Store readings in an array `distHistory[181]`.
   - Drawing Logic (Artifact Free & Correct Orientation):
      - **Orientation**: Use `angle` directly for radians (not `180-angle`) to match physical servos.
      - **Sector Clearing**: Before drawing, draw a Black Triangle from Center to (CurrentAngle, 0).
      - **Explicit Erasure**: Draw a Black Line at the *previous* angle to ensure the old red scan is removed.
      - **Map Restoration**: Restore the "Shadow Zone" (Navy) and "Wall" (Cyan) at the previous angle.
      - **Jump Filter**: Only connect wall points if distance difference < 150mm.
      - Draw Red Scanner Line on top.
   - Split Screen: Top half (0-120px) is Radar, Bottom half is large Digital Text (Distance/Angle).
3. Connectivity:
   - Static IP: 192.168.178.71.
   - Websocket Server broadcasting JSON {angle, distance}.

Output: Complete .ino code including User_Setup.h configuration comment.

```

## Variante 2: Das Zukunfts-System (Quad-Core Static)

### Status: Concept / Next Step

4 Sensoren, 90° versetzt (Vorne, Hinten, Links, Rechts). Keine beweglichen Teile (Servo).

### Die Herausforderung & Lösung

Du kannst am ESP8266 nicht einfach 4 TOF-Sensoren parallel anschließen, da alle die gleiche I2C-Adresse (0x29) haben.

#### Die Hardware-Erweiterung:

- \* Du benötigst einen **I2C Multiplexer (TCA9548A)**.
- \* Der ESP redet nur mit dem Multiplexer.
- \* Der Multiplexer hat 8 Kanäle (SD0-SD7 / SC0-SC7).
- \* Du schließt jeden Sensor an einen eigenen Kanal an.

### Verdrahtungsplan (Zukunft)

Komponente	Pin	Geht an
<b>Multiplexer</b>	SDA / SCL	An Wemos D2 / D1
<b>Display</b>	(Alle Pins)	Wie oben (D5, D7, D8, D3, D0)
<b>Sensor Vorne</b>	SDA / SCL	An Multiplexer Kanal 0
<b>Sensor Rechts</b>	SDA / SCL	An Multiplexer Kanal 1
<b>Sensor Hinten</b>	SDA / SCL	An Multiplexer Kanal 2
<b>Sensor Links</b>	SDA / SCL	An Multiplexer Kanal 3

*(Servo entfällt, dafür hast du jetzt 360° Abdeckung ohne Mechanik)*

### Der Prompt für Variante 2 (Zukunft)

*Speichere diesen Prompt für später, wenn du den Multiplexer hast.*

Role: Senior Firmware Architect for ESP8266.

Project: NERO SENSE - Quad Static Lidar (360°).

Hardware: Wemos D1 Mini, GC9A01 Display, TCA9548A Multiplexer, 4x VL53L1X Sensors.

Hardware Concept:

- No moving parts (Solid State).
- 4 Sensors mounted at 0° (Front), 90° (Right), 180° (Back), 270° (Left).
- Sensors connected via TCA9548A Multiplexer on Channels 0, 1, 2, 3.

Task: Write the complete Arduino Firmware.

Features:

1. Visualization (The "Shield View"):

- Divide the round display into 4 quadrants.
- Center of display represents the robot.
- Visualize the distance of each sensor as a "Shield Bar" moving from the center outwards.
  - Close object = Bar close to center (Red).
  - Far object = Bar far from center (Cyan).
- Display the exact distance in mm in each quadrant corner.

2. Multiplexer Logic:

- Implement a function `tcaselect(uint8\_t i)` to switch I2C channels before reading each sensor.

3. Boot Sequence:

- "NERO SENSE 360" text intro.
- Check all 4 sensors. If one fails, show error on screen.

4. Connectivity:

- Static IP: 192.168.178.42.
- Websocket JSON: {front, right, back, left}.

Output: Complete .ino code. Use "Wire.h" and "VL53L1X.h".

Generated by Antigravity for Nero Robotics