Multi-Modal Sensor Hardware Blueprint for Real-Time Action Detection Product

AI/ML Project Documentation

August 20, 2025

1 Overview

This document provides a blueprint for building a physical product that integrates LIDAR, thermal camera, and normal RGB camera into a single device for 360-degree coverage and continuous monitoring. The product collects inputs from all three sensors, combines them electrically, and sends fused data to software for action detection (walking, jumping, clapping, or null). Designed for non-electrical engineers, it explains components, architecture, and a step-by-step build process. The goal is a compact, robust device that runs 24/7, powering our multi-modal fusion system (previous blueprint). After hardware, we transition to firmware/software.

1.1 Objectives

- **Integrate Sensors**: Combine LIDAR, thermal, and RGB cameras for 360-degree input collection.
- **Ensure Continuous Operation**: Design for non-stop monitoring with power management and heat dissipation.
- **List Components**: Detail required electrical parts for assembly.
- **Architecture and Build**: Provide a simple architecture and beginner-friendly steps to create the product.
- **Transition**: Outline next steps for firmware (sensor control) and software (data processing).

1.2 Project Context

- **System**: Multi-modal fusion for action detection, extending Three-Action Classifier (MobileNetV2, 45 videos, >85% accuracy).
- **Hardware Role**: The product houses sensors, collects raw data (point clouds from LIDAR, heat maps from thermal, images from RGB), and sends it to software.
- **Goals**: Compact device (20x20x10cm, 1-2kg), low power (10-20W), continuous operation (>24 hours without interruption).

2 General Explanation

The product is like a smart "eyeball" device that sees the world in three ways: normal colors (RGB camera), heat (thermal camera), and 3D shapes/distances (LIDAR). These three "eyes" work together in one box, covering all directions (360 degrees) without blind spots, and keep watching non-stop. The device collects what each eye sees, mixes the signals electrically inside, and sends the combined info to a computer or phone for the software to decide if there's an action (like jumping) or nothing. If no action, it stays quiet. Building it is like putting together a puzzle: choose the right parts (sensors, wires, power supply), connect them on a board, and enclose in a case. It's designed to run forever, with fans to stay cool and batteries for backup.

3 Technical Details

The product architecture is a centralized design with sensors mounted for 360-degree coverage, connected to a microcontroller board for data aggregation and transmission. It uses a single power source, shared data bus, and synchronization circuitry for continuous operation. Below are key technical aspects.

3.1 Hardware Components List

- **LIDAR Sensor**: Velodyne Puck or Ouster OS1 (360-degree scanning, 10-20m range, 1-2W power, $500-$1000).
- **Thermal Camera**: FLIR Lepton or Seek Thermal CompactPRO (320x240 resolution, 8-14m wavelength, 1W power, $200-$400).
- **Normal RGB Camera**: Raspberry Pi Camera Module v2 or Arducam OV9281 (1080p resolution, 30 FPS, 0.5W power, $20-$50). Use 4-6 for full 360-degree coverage if one isn't omnidirectional.
- **Microcontroller Board**: Raspberry Pi 5 or NVIDIA Jetson Nano (4-8 GB RAM, multi-core CPU/GPU, supports CSI/USB interfaces, 5-10W power, $35-$100).
- **Power Supply**: 12V DC adapter with LiPo battery backup (10,000mAh for 24+ hours, voltage regulator for 5V/3.3V, $20-$50).
- **Data Bus and Connectors**: I2C/SPI for sensor communication, USB-C for Jetson, Ethernet for high-bandwidth output ($10-$20).
- **Cooling and Enclosure**: Heat sinks/fans for Jetson (to prevent throttling), waterproof IP67 case (20x20x10cm, aluminum/plastic, $50-$100).
- **Additional Parts**: Wires, breadboards for prototyping, soldering kit, fuses for safety ($20).

Total estimated cost: $800-$1700 (prototype); $400-$800 (optimized).

3.2 Architecture for Combining Components

The architecture is a star topology: all sensors connect to the central microcontroller (Jetson Nano) via dedicated interfaces (CSI for cameras, USB/Ethernet for LIDAR). A shared power bus distributes voltage, with regulators for each sensor. Data flows through a common bus (I2C/SPI) for synchronization, aggregated on the board, and output via Ethernet/USB to software. For 360-degree coverage, mount sensors in a circular array (e.g., LIDAR on top, thermal/RGB around sides). Firmware (e.g., on Jetson) synchronizes timestamps and fuses low-level signals (e.g., align RGB frames with thermal heat maps and LIDAR points). Continuous operation uses battery backup and low-power modes (e.g., sleep sensors when idle). Heat dissipation via passive cooling (sinks) and active fans triggered at >60řC.

3.3 Step-by-Step Build Process

1. **Plan and Gather Components**: List parts (above), buy from suppliers (Amazon, Adafruit, Digi-Key). Draw a simple diagram: central board with sensors branching out.

2. **Prototype on Breadboard**: Connect sensors to Jetson using wires (e.g., RGB via CSI pins, thermal via I2C, LIDAR via USB). Use jumper wires for testingno soldering yet. Power the board with adapter, test connections (e.g., run basic scripts to read sensor data).

3. **Add Power Management**: Wire power supply to a distribution board (e.g., breadboard power module), connect battery for backup. Add regulators to step down voltage (12V to 5V/3.3V for sensors). Test continuous run (e.g., 1 hour without interruption).

4. **Integrate Data Bus**: Connect sensors to Jetson's GPIO pins (I2C/SPI for thermal/RGB, USB for LIDAR). Use a multiplexer if pins are limited. Test data flow: read simultaneous inputs from all sensors.

5. **Mount for 360-Degree Coverage**: Attach sensors to a circular base (e.g., 3D-printed mount): LIDAR on top for full scan, thermal/RGB in a ring (e.g., 72-degree spacing for 360 coverage). Ensure no blind spots by overlapping fields of view.

6. **Add Cooling and Enclosure**: Attach heat sinks/fans to Jetson, wire fan to GPIO for temperature-triggered control. Enclose in case with vents and mounts for sensors. Test thermal stability (e.g., run for 2 hours, monitor temperature <60řC).

7. **Test Basic Functionality**: Power on, verify all sensors output data continuously (e.g., RGB frames, thermal maps, LIDAR points). Check for interruptions (e.g., battery switchover).

8. **Final Assembly and Soldering**: Solder connections on a PCB (printed circuit board, $20 custom order from JLCPCB) for permanence. Seal enclosure for dust/weather protection. Perform full test: 24-hour run with dummy data.

9. **Safety and Certification**: Add fuses to prevent shorts, test for electrical safety. For production, certify (e.g., CE/FCC for EMI).

Total build time: 1-2 weeks for prototyping (non-engineer with online guides), 1 month for polished product. Cost: $800-$1700 (as listed). Resources: YouTube tutorials (e.g., "Jetson Nano Sensor Integration"), Adafruit guides, or hire a freelance engineer via Upwork ($200-500 for assembly).

4 Transition to Firmware and Software

Once the hardware is built, firmware (e.g., embedded Linux on Jetson) handles sensor control and data streaming, while software (e.g., PyTorch/DeepStream on a connected server) processes fused inputs for action detection. The hardware sends synchronized data via Ethernet/USB, enabling the multi-modal fusion technique (Hybrid Fusion with Transformers) explained previously.

4.1 Firmware Role

Firmware manages sensor synchronization, data acquisition, and low-level fusion (e.g., timestamp alignment), ensuring continuous operation. Use NVIDIA JetPack SDK for Jetson firmware, with custom scripts for sensor polling.

4.2 Software Role

Software performs high-level fusion, inference, and reactions. Use DeepStream for pipeline, TensorRT for optimized MobileNetV2, and ROS 2 for orchestration.

This blueprint enables a self-contained product. Next, we can dive into firmware/-software details step by step.