



# CubeBot: FPGA-Based Rubik's Cube Solver

Mohammad Hamdan, Momen Anani  
Supervisor: Dr.Suleiman Naeem Abu Kharmeh



2025

# Table of contents

**01**

**Introduction,  
Motivation**

**02**

**System Arch &  
Key Features**

**03**

**Methodology**

**04**

**Implementation  
& Key Results**

**05**

**Challenges &  
Future Work**

**06**

**Conclusion**





**01**

**Introduction,  
Motivation**





# What is CubeBot?

- A fully autonomous robot that **senses, computes, and solves** a 3x3 Rubik's Cube.
- Built on a **heterogeneous embedded architecture** combining **FPGA acceleration, ARM processing, and microcontroller control**.
- **Elegant Proof-of-Concept:** Demonstrates complex robotics using minimal, cost-effective hardware



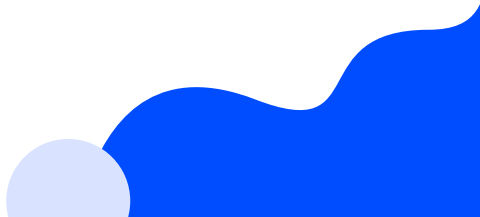
# The Challenge & Why FPGA?

## The Problem with Traditional Solvers:

### Software-Only (Raspberry Pi/PC):

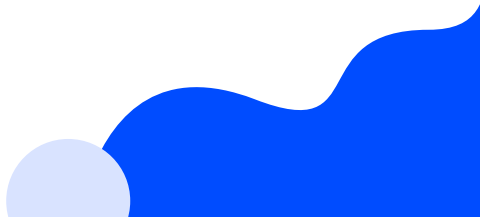
- Non-deterministic timing due to OS scheduling.
- High latency in image processing.
- Struggles with concurrent tasks (vision + control).

### Microcontroller-Only (Arduino):

- Limited computational power for complex solving algorithms.
  - Poor performance in real-time image processing.
- 

# The Challenge & Why FPGA? Cont.

## Our Solution: Strategic Heterogeneity:

- **FPGA** : For **deterministic, hardware-accelerated** color perception.
  - **ARM Cortex-A9 (HPS)**: For **high-level coordination** and complex algorithm execution.
  - **ESP32**: For **precise, real-time motor control** and wireless interfacing.
- 



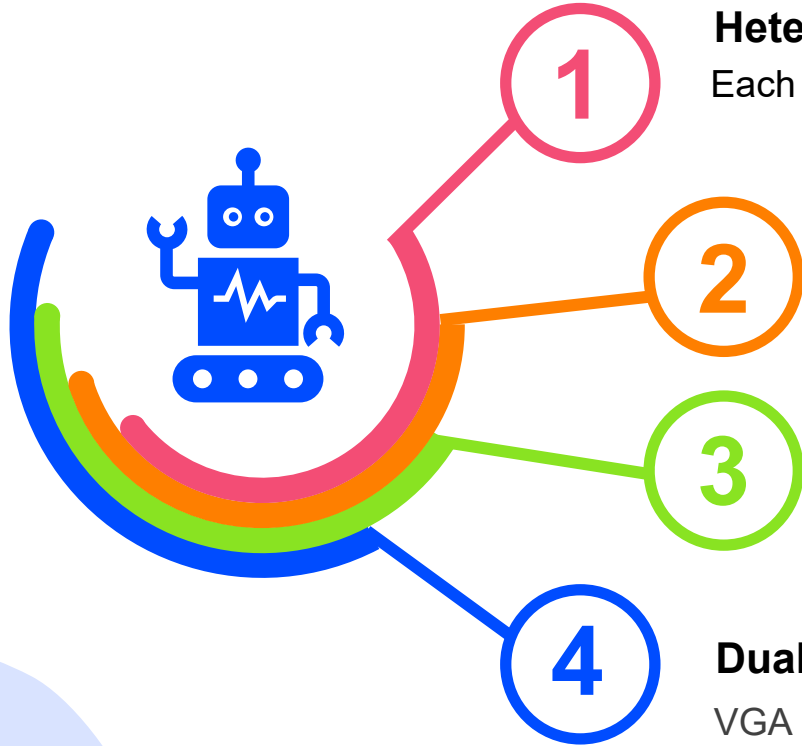
**02**

**System Architecture &  
Key Features**



# Core Differentiators

## What Makes CubeBot Unique



### 1 Heterogeneous Processing

Each platform excels at its dedicated task.

### 2 Deterministic Vision

FPGA-based pipeline guarantees low processing time.

### 3 Robust Communication

Custom UART packet protocol with 100% reliability.

### 4 Dual Monitoring

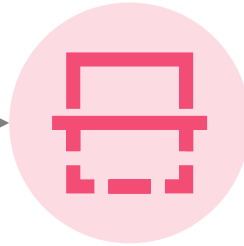
VGA hardware display + Wireless Web Dashboard for full system visibility.

# End-to-End System Flow



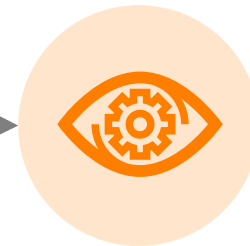
## User Command

Via Wireless Dashboard or Push Buttons



## Scanning

USB Camera captures cube faces



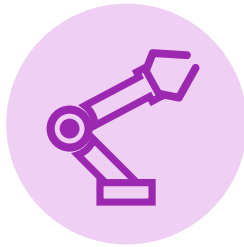
## Perception

FPGA Hardware-accelerated color extraction



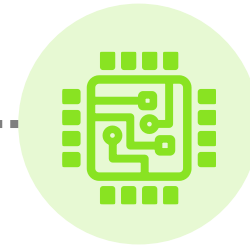
## Monitoring

Real-time status on VGA & Dashboard.



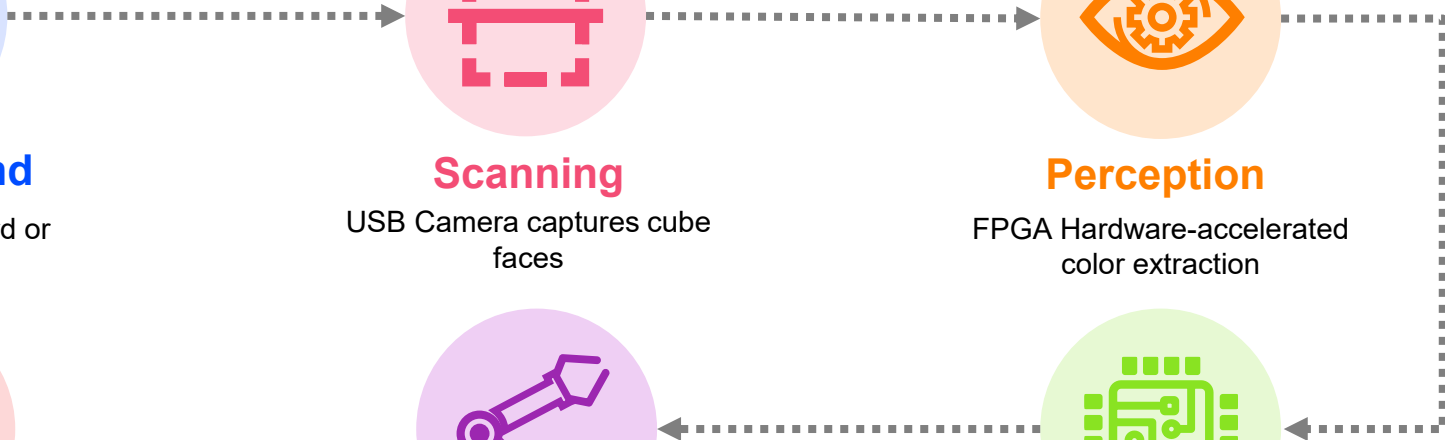
## Actuation

Motor control executes moves with sensor feedback on Esp.



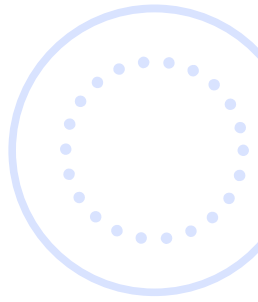
## Computation

State validation & Kociemba algorithm solving on HPS



# 03

## Methodology



# FPGA Development

## Platform Designer (Qsys)

Built some Subsystems with custom IP blocks.

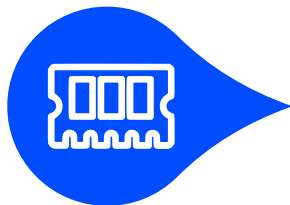


## Image Processing Pipeline

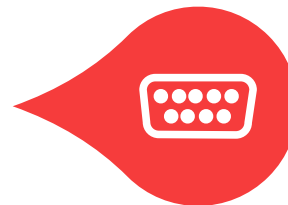
Designed for deterministic timing

## Memory-Mapped I/O

HPS accesses FPGA at 0xFF200000 via /dev/mem



## Key FPGA Design Decisions

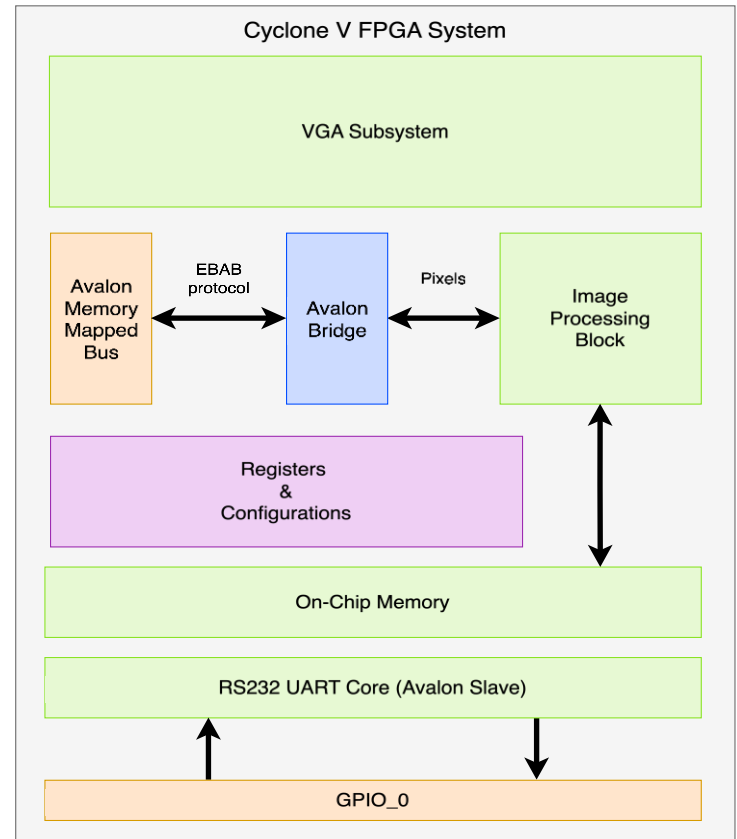


## VGA Subsystem

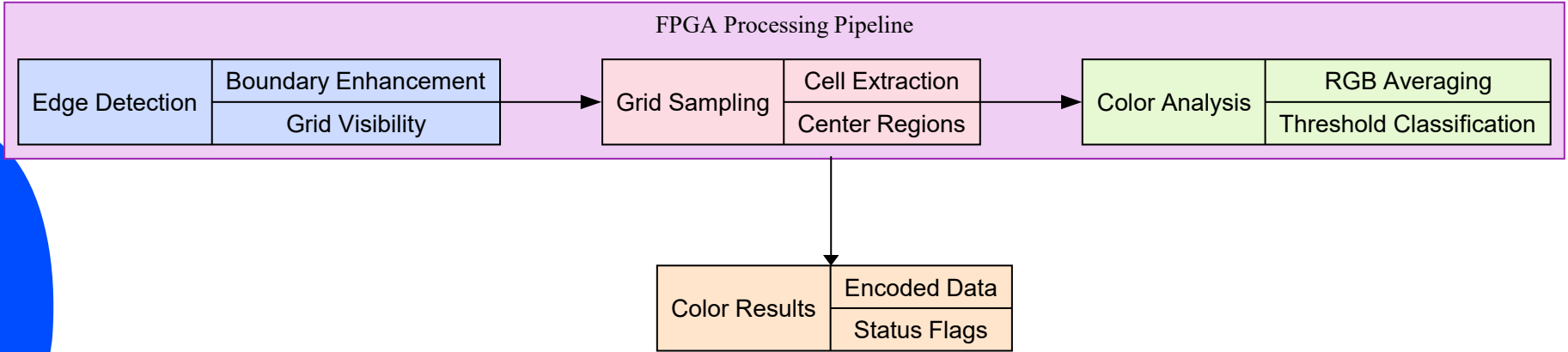
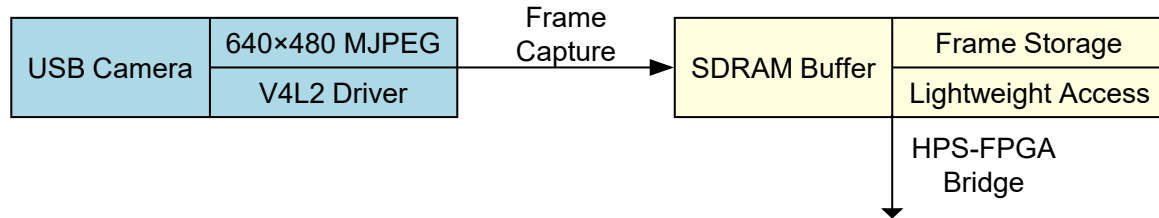
Hardware display independent of CPU load

# What We Built on FPGA

- Image processing unit.
- On-chip memory buffers for move storage.
- UART IP for external communication.
- VGA Subsystem.
- Different registers and Bridges.



# FPGA Image Processing Pipeline



# ARM HPS Software Development

## Software Architecture

### Embedded Linux

Chose for flexibility with camera drivers (V4L2).

### Main State Machine

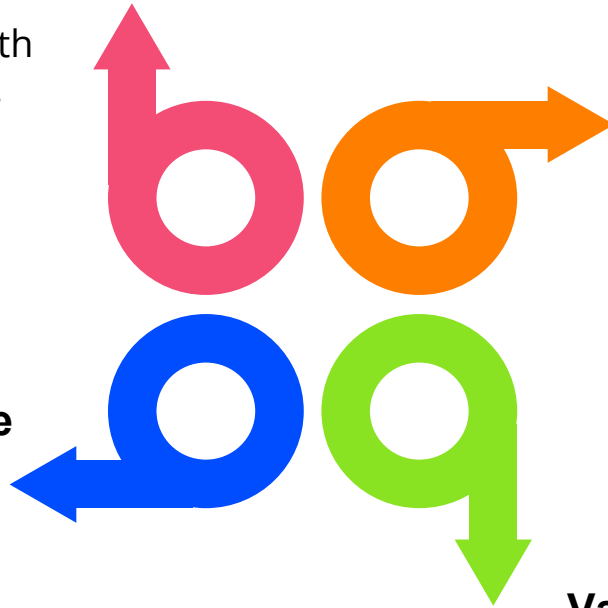
1. Controls scan
2. validate
3. solve
4. execute flow

### Cube Solving

Implemented Kociemba two-phase algorithm

### Validation Logic

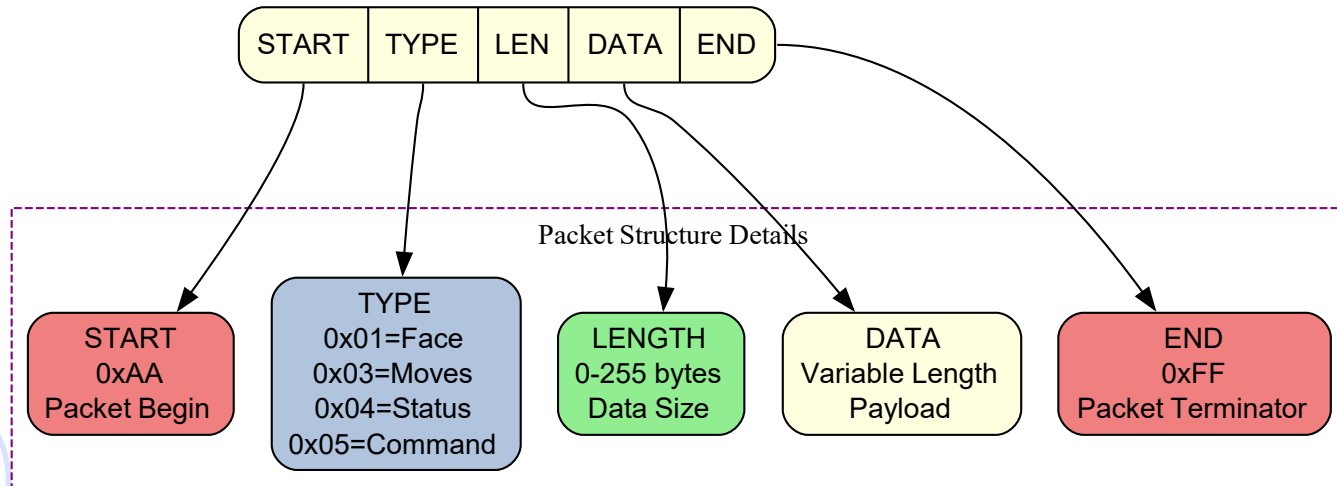
Checks color counts, parity.



# ARM HPS Software Development

## Communication Implementation

- Memory-mapped I/O to FPGA peripherals
- UART communication to ESP32 with custom packet protocol
- Move encoding from symbolic (R, U', F2) to robot commands



# ESP32 Motor Control Development

## Stepper Motor

Configured for 3200 steps/revolution with micro stepping

## Wireless Dashboard

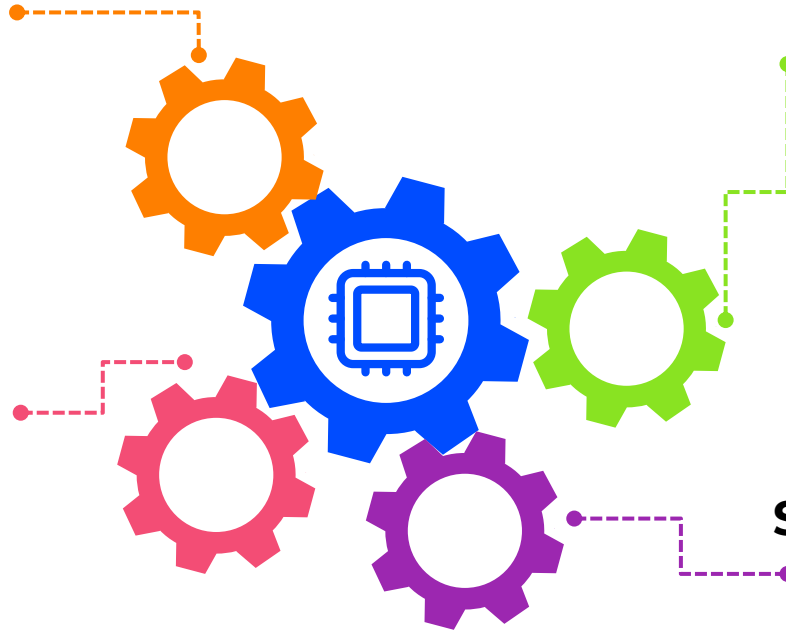
HTTP server for control and monitoring

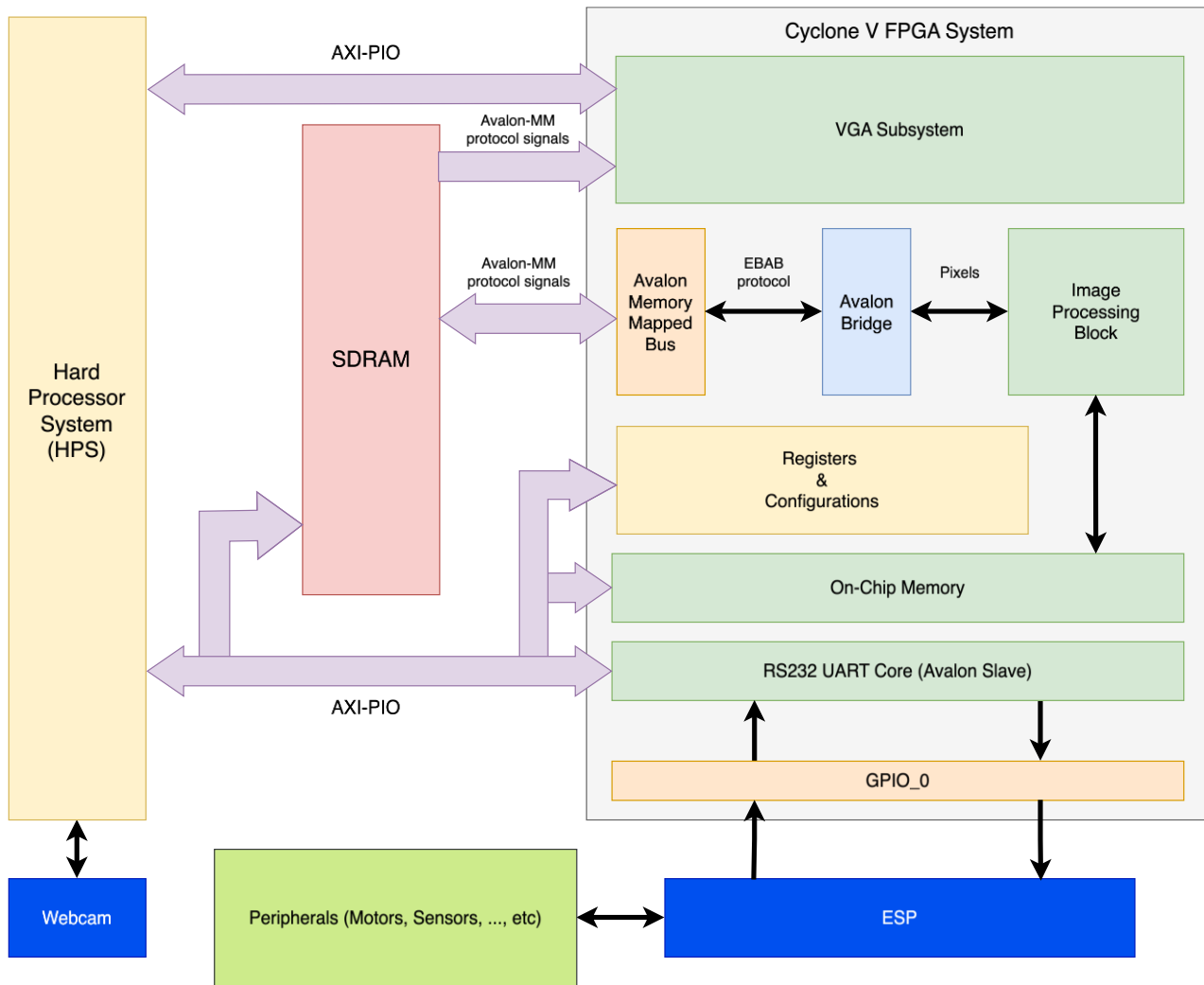
## Servo Control

Dual servos via PCA9685 PWM driver for gripping

## Sensor Integration

Optical slot sensor for mechanical alignment

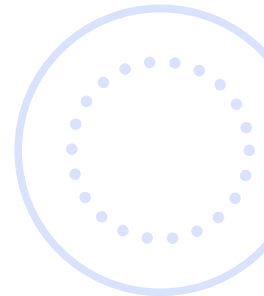






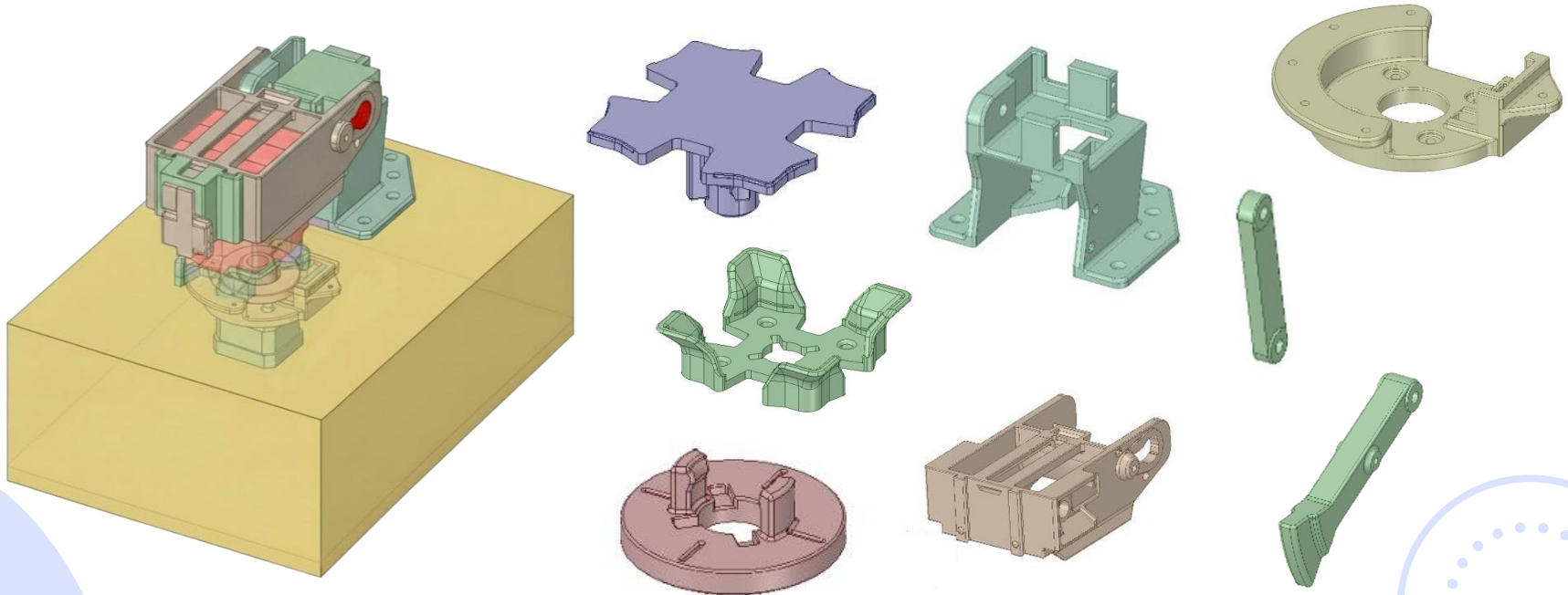
# 04

## Implementation & Key Results

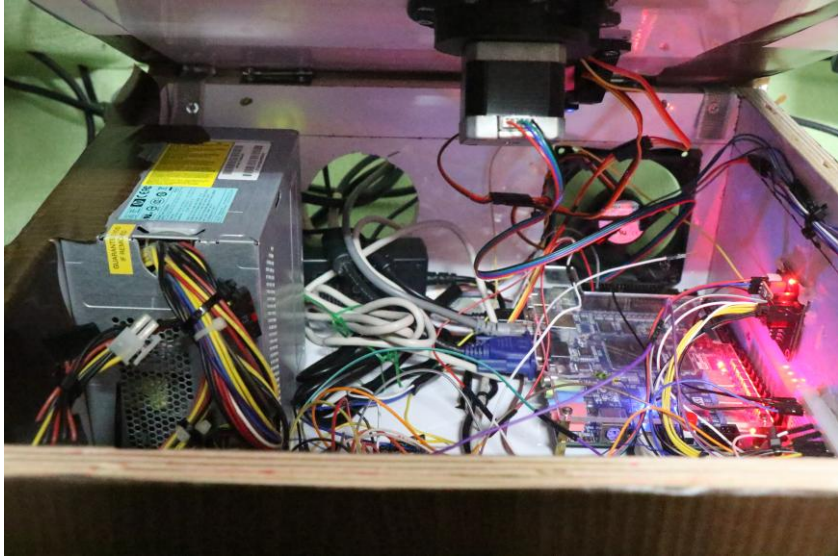


# Physical System & Interfaces

- **Fully Integrated Prototype:** Mechanical assembly with 3D-printed parts, integrated electronics

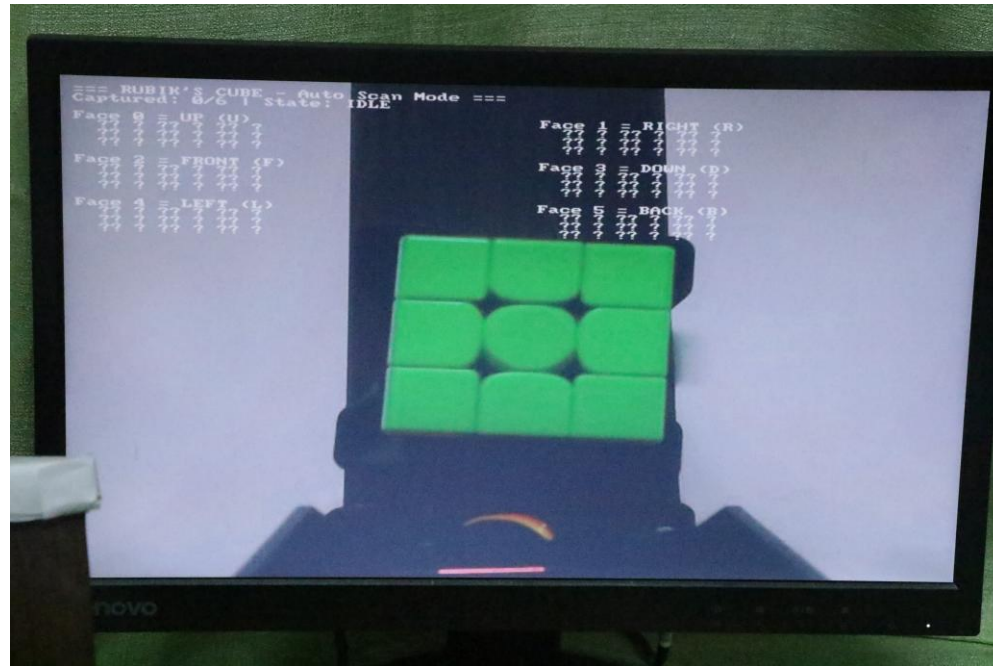


# Physical System & Interfaces Cont.



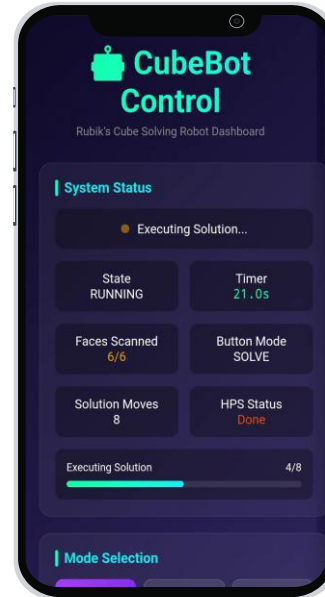
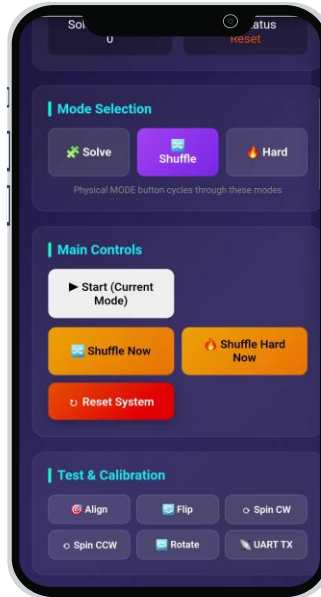
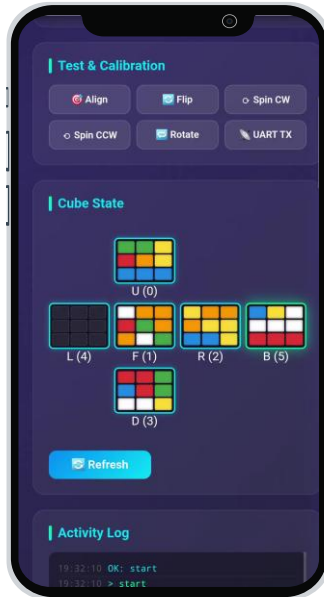
# Physical System & Interfaces Cont.

- **VGA Hardware Display:** Shows real-time state, scanning progress, move count.



# Physical System & Interfaces Cont.

- **Wireless Dashboard:** Provides control, cube state visualization, and live progress.



# Experimental Performance Validation

- **Color Detection Accuracy: 98.7%** (296/300 facelets correct).
- **Solve Success Rate: 93.3%** (28/30 solves).
- **Mean Total Solve Time: 46.1 seconds.**
  - Scanning: 12.8s | Solving: 1.6s | Execution: 28.4s
- **FPGA Efficiency:**
  - **ALMs: 27% | Block Memory: 2% | DSP Blocks: 20%**
  - **Deterministic Timing: 14.8 ms/face guaranteed.**



**05**

**Challenges &  
Future Work**



# Lessons Learned & Limitations

- **Challenge:** Camera/FPGA interface synchronization, memory coherency and debugging .
- **Challenge:** Achieving timing closure for the image pipeline in FPGA.
- **Limitation:** Color detection sensitive to ambient lighting (fixed thresholds).
- **Limitation:** Mechanical slippage caused 6.7% of failures.
- **Limitation:** Lack of FPGA learning resources.

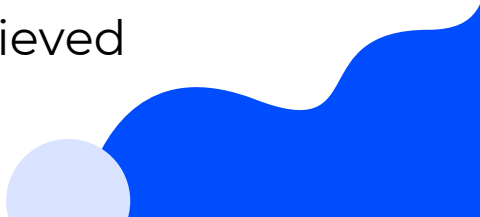


# Future Recommendations

- **Enhanced Vision:** Implement **ML-based color classification** on FPGA for lighting robustness.
- **Faster Execution:** Optimize motor profiles.
- **Power Optimization:** Reduce consumption for battery-powered operation.
- **Advanced Algorithms:** Explore reinforcement learning for novel solving strategies.



# Conclusion

- **Functional End-to-End Robot** that senses, plans, and acts autonomously.
  - **Validated Heterogeneous Architecture** proving the benefits of hardware-software co-design.
  - **High Performance & Reliability:** 98.7% accuracy, 100% communication reliability, deterministic timing.
  - **Professional-Grade Implementation** integrating FPGA design, embedded Linux, real-time control, and mechanical assembly.
  - **Elegant Proof-of-Concept:** Complex robotics achieved with minimal, cost-effective hardware.
- 

The slide features a white background with decorative blue elements. In the top right corner, there is a blue circle partially overlapping a larger blue circle, with a white dotted circle inside. In the bottom left corner, there are two overlapping blue circles, one larger and more vibrant than the other. The word "Thanks" is centered in a large, bold, blue font.

# Thanks

Do you have any questions?