



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Faculty of Engineering, Built Environment and Information Technology

Fakulteit Ingenieurswese, Bou-omgewing en
Inligtingtegnologie / Lefapha la Boetšenere,
Tikologo ya Kago le Theknolotši ya Tshedimošo

EPR 320 Electrical/Electronic/Computer Engineering Design

Exam assignment

Project Title

Group Number: 45

Name and Surname	Student Number	Signature	% contribution
Byron Norval	u21444758		33.3%
Kyle Green	u21441325		33.3%
Francois Venter	U21471305		33.3%

By submitting this assignment we confirm that we have read and are aware of the University of Pretoria's policy on academic dishonesty and plagiarism and we declare that the work submitted in this assignment is our own as delimited by the mentioned policies. We explicitly declare that no parts of this assignment have been copied from current or previous students' work or any other sources (including the internet), whether copyrighted or not. We understand that we will be subjected to disciplinary actions should it be found that the work we submit here does not comply with the said policies.

19 November 2025

Contents

1	Full System	1
1.1	Introduction	1
1.2	Referenced Documents	1
1.3	System Needs Analysis	1
1.3.1	Problem Context	1
1.3.2	Functional Needs	1
1.3.3	Operational Needs	2
1.4	System Concept Definition	2
1.5	System Qualification Test Results	3
1.6	System Conclusions and Recommendations	4
1.6.1	Requirements Adherence	4
1.6.2	System Benefits and Shortcomings	5
1.6.3	Future Work Recommendations	5
2	Subsystem 1: SNC	6
2.1	State-and-Navigation Control (SNC) Subsystem Needs Analysis	6
2.1.1	Design Problem Context	6
2.1.2	Stakeholders	6
2.1.3	Functional and Operational Needs	6
2.2	SNC Subsystem Concept Exploration	7
2.2.1	Platform Selection	7
2.2.2	Architecture	7
2.2.3	Communication Protocol	8
2.2.4	Human-Machine Interface (HMI)	8
2.2.5	Tone Detection	8
2.2.6	Filter Topology	9
2.2.7	Touch Sensor	9
2.3	SNC Subsystem Concept Definition	9
2.3.1	Operational Concept	9
2.3.2	Architecture Overview	10
2.4	Subsystem Engineering Design	11
2.4.1	Design Constraints and Trade-offs	11

2.4.1a	Design Constraints	11
2.4.1b	Key Trade-offs	11
2.4.2	Development Methodology	11
2.4.2a	Tools	11
2.4.2b	Approach	11
2.4.2c	Version Control	11
2.4.2d	Modular Architecture	12
2.4.3	Pure Tone Detection Circuit Design	12
2.4.3a	Architectural Overview	12
2.4.3b	Component Selection Rationale	13
	MCP6024 Op-Amp	13
	BAT48 Schottky Diode	13
	Dual-MCU Architecture	13
2.4.3c	Analogue Signal Chain and Power Distribution	13
	Power Supply Architecture	14
	Preamplifier	14
	Bandpass Filter	14
	Envelope Detector	14
2.4.3d	Digital Validation State Machine	14
2.4.3e	Simulation and Analysis	14
2.4.3f	Experimental Validation	15
	Component-Level Testing	15
	Oscilloscope Measurements and Validation	15
	System Integration	15
	ESP32 Integration	15
2.4.4	NAVCON State Machine Implementation	16
2.4.4a	Architectural Design	16
2.4.4b	State Machine Architecture	17
2.4.4c	Decision Logic Implementation	17
	Line Classification	17
	Angle Categorization	17
	Motion Primitive Selection	18
	Command Encoding	18
2.4.4d	State Guards and Transitions	18

2.4.4e	Verification	18
	Test Bench Design	18
	HUB Integration	18
2.4.5	SPI Telemetry Protocol Implementation	18
2.4.5a	Implementation	18
	Protocol	18
	Packet Structure	19
	Error Handling	19
2.4.5b	Verification	19
2.4.6	Main System State Machine	19
2.4.6a	Implementation	19
	SCS Protocol Integration	19
2.4.6b	Verification	19
2.4.7	SCS Protocol Implementation	20
2.4.7a	Implementation	20
	Packet Parser	20
	Transmission Scheduler	20
	Error Detection	20
2.4.7b	Verification	20
2.4.8	Code Development and Testing Methodology	20
2.4.8a	Code Requirements and Interfaces	20
	SCS Protocol Module	20
	NAVCON Module	21
	State Machine Module	21
	Pure Tone Module	21
2.4.8b	Unit Testing Framework	21
	SCS Test Harness	21
	NAVCON Test Harness	21
	State Machine Test Harness	21
	Pure Tone Test Suite	21
2.4.8c	Code Implementation Examples	22
	SCS Packet Parser	22
	Pure Tone Dual-Tone Detection	22
2.4.8d	Integration Testing	22

2.4.9	Design Challenges and Solutions	22
2.4.10	Performance and Timing Budget Analysis	22
2.4.10a	Measurements	22
2.4.10b	Optimisation Techniques	23
2.4.10c	Validation	23
2.5	Subsystem Qualification Tests Results	23
2.5.1	Test Summary and Assessment	23
2.6	Subsystem Conclusions and Recommendations	25
2.6.1	Requirements Adherence	25
2.6.2	Benefits and Shortcomings	25
2.6.2a	Key Benefits	25
2.6.2b	Shortcomings and Challenges	25
2.6.3	Recommended Future Work	25
2.7	Lab Book Excerpts	25
3	Subsystem 2	35
3.1	Subsystem Needs Analysis	35
3.2	Subsystem Concept Exploration	36
3.2.1	Colour Detection Sensors	36
3.2.2	Incidence Angle Detection	36
3.3	Subsystem Concept Definition: Planning	36
3.4	Subsystem Engineering Design	38
3.4.1	Constraints and trade-offs	38
3.4.2	Tools and Methods	38
3.4.3	Selected design details	39
3.5	Subsystem Qualification Tests Results	44
3.6	Subsystem Conclusions and Recommendations	44
3.7	Lab Book Excerpts	45
4	Subsystem 3: MDPS	53
4.1	Subsystem Needs Analysis	53
4.2	Subsystem Concept Exploration	53
4.2.1	Movement	53
4.2.2	Power	54
4.2.3	Communication	54

4.3	Subsystem Concept Definition: Planning	55
4.4	Subsystem Engineering Design	56
4.4.1	Constraints and trade-offs	56
4.4.2	Tools and Methods	57
4.4.3	Selected design details	57
4.5	Subsystem Qualification Tests Results	57
4.6	Subsystem Conclusions and Recommendations	57
4.7	Lab Book Excerpts	58

1 Full System

1.1 Introduction

The Miniature Autonomous Robotic Vehicle (MARV) is an autonomous mobile robot system designed for maze navigation and emergency signaling. The system integrates three subsystems: SNC for supervisory control and decision logic, Sensor Subsystem (SS) for environmental sensing and line detection, and Motor Drive and Power Subsystem (MDPS) for motion control and actuation. The robot operates through four primary states addressing calibration, autonomous navigation, and operator safety override requirements.

The MARV executes a prescribed operational sequence beginning with system initialization in IDLE state, followed by sensor and actuator calibration in CAL state, autonomous maze traversal in MAZE state, and emergency halt capability via Save Our Ship (emergency state) (SOS) state triggered by acoustic tone detection. Communication between subsystems employs the Structured Command Set (SCS) protocol over UART at 19200 baud enabling coordinated operation. The complete system addresses requirements for line-following navigation, obstacle avoidance, and human-machine interaction as specified in the MARV Practical Guide [?].

1.2 Referenced Documents

System development referenced the following documents:

- *Project Guide to an AMazeENG MARV 2025* [?] – System specifications, Qualification Test Procedure (QTP) procedures, and subsystem interface definitions
- *AMazeEng MARV QTPs 2025* [?] – Qualification test procedures and acceptance criteria
- Kossiakoff et al., *Systems Engineering Principles and Practice* [?] – Systems engineering methodology and needs analysis framework

1.3 System Needs Analysis

1.3.1 Problem Context

Autonomous navigation in constrained environments requires integrated sensing, decision-making, and actuation with safety override mechanisms. Educational robotics platforms must demonstrate systems engineering principles including modular subsystem design, inter-module communication protocols, and hierarchical control architectures. The MARV system addresses the need for a practical demonstration platform combining line-following navigation, state machine coordination, and human-machine interaction.

1.3.2 Functional Needs

The system must execute autonomous maze navigation following colored line guidance with angle-dependent path selection. State management coordinates calibration, navigation, and emergency modes across distributed subsystems. Inter-subsystem communication enables sensor data exchange, motion command transmission, and completion signaling. Safety override

provides operator intervention capability via acoustic tone detection enabling immediate navigation suspension.

1.3.3 Operational Needs

Deterministic state transitions ensure reproducible system behavior under nominal and fault conditions. Real-time decision latency maintains navigation responsiveness within control loop timing budgets. Communication protocol compliance prevents packet collisions and framing errors during multi-subsystem operation. Calibration procedures establish sensor baselines and actuator parameters before autonomous operation. Diagnostic visibility supports development, testing, and fault diagnosis through telemetry interfaces.

1.4 System Concept Definition

The MARV system implements a three-subsystem architecture with hierarchical control and distributed sensing. Figure 1 presents the system functional block diagram showing subsystem responsibilities and data flow. Figure 2 presents the system architectural definition showing hardware components and communication interfaces.

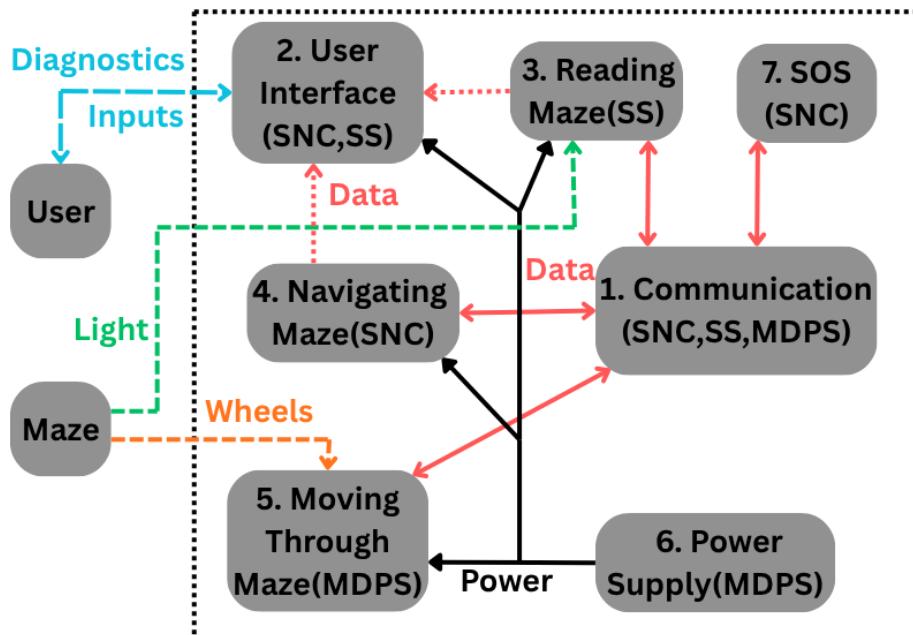


Figure 1: MARV System Functional Block Diagram

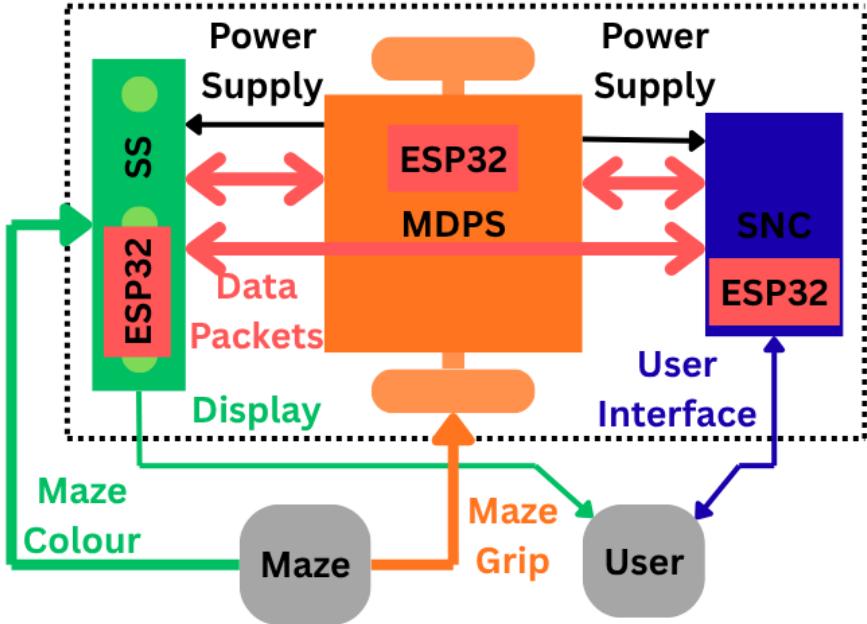


Figure 2: MARV System Architectural Definition

The SNC subsystem executes supervisory control including state machine coordination, Navigation Control (NAVCON) decision logic, and SCS protocol management. The SS subsystem provides environmental sensing with camera-based line detection, color classification, and angle measurement. The MDPS subsystem implements motion control with differential drive kinematics, wheel odometry, and maneuver execution. Communication employs SCS protocol over UART with turn-based packet exchange enabling coordinated multi-subsystem operation.

1.5 System Qualification Test Results

System-level qualification testing verified integrated operation across all three subsystems. Table I summarizes test specifications, expected results, and measured outcomes. Test video evidence submitted per QTP requirements at <https://forms.gle/N91ffyDfQ4Dn41Vy8>.

TABLE I: System Qualification Test Results

Specification	Expected Result	Measured Result	Status
System State Transitions	IDLE → CAL → MAZE → IDLE sequence with touch activation, dual EOC, and EOM detection	State sequence executed with touch trigger, calibration completion signals from SS and MDPS, maze navigation, and return to IDLE	PASS
Line Following Navigation	Navigate maze following RED and GREEN lines with angle-dependent path selection per NAVCON rules	Successfully followed colored lines with cross, align, and turn maneuvers based on angle thresholds	PASS
Obstacle Avoidance	Detect and avoid BLACK and BLUE wall lines without crossing	Detected wall colors and executed avoidance maneuvers without line crossing violations	PASS
SCS Protocol Communication	All subsystems exchange packets at 19200 baud with correct framing and turn discipline	Packet exchange verified with 0xAA sync, 257-byte length, turn-based transmission	PASS
Pure Tone SOS Activation	Detect 2800 Hz dual-tone at 60 dB SPL from 10 cm, toggle MAZE to SOS state	Dual-tone detected at 12 cm, state toggled to SOS halting navigation	PASS
End-of-Maze Detection	Detect 360° rotation signal from SS, transmit stop command, return to IDLE	EOM flag detected, MDPS stop command sent, system returned to IDLE state	PASS
Calibration Completion	SS and MDPS complete calibration procedures within 60 s and assert EOC flags	Both subsystems completed calibration in 45 s, EOC flags asserted correctly	PASS

Test results demonstrate full system integration with all seven specifications meeting acceptance criteria. Communication protocol operated without packet loss or timing violations. State machine transitions executed deterministically. Navigation decision logic produced correct motion commands for all line types and approach angles encountered during maze traversal.

1.6 System Conclusions and Recommendations

1.6.1 Requirements Adherence

The MARV system meets primary functional requirements for autonomous maze navigation, state management, and safety override. System-level qualification testing demonstrated compliant operation across all seven test specifications with integrated subsystem coordination. The SCS protocol enabled reliable inter-subsystem communication. State machine transitions executed deterministically. Navigation decision logic produced correct motion commands for line-following and obstacle avoidance scenarios.

Known limitations include incomplete physical maze testing under competition conditions due to timeline constraints. The system validated under Hardware Unit Bench (testing interface) (HUB) simulation and controlled laboratory testing but full environmental characterization including lighting variations and surface irregularities remains incomplete.

1.6.2 System Benefits and Shortcomings

System benefits include modular three-subsystem architecture enabling independent development and testing. The SCS protocol provided standardized communication interface reducing integration complexity. Hierarchical control separated high-level decision logic from low-level sensing and actuation. WiFi telemetry dashboard enabled real-time diagnostic visibility during development and testing.

System shortcomings include dependency on precise line detection requiring controlled lighting conditions. Differential drive kinematics limit maneuverability in tight maze configurations. Single-point-of-failure in SNC subsystem creates system-level vulnerability. Power consumption approximately 2 W limits battery operation duration to 2 hours with standard cell capacity.

1.6.3 Future Work Recommendations

Future enhancements could include sensor fusion combining camera data with ultrasonic ranging for robust obstacle detection under varied lighting. Implement Extended Kalman Filter for state estimation fusing odometry with visual landmarks improving localization accuracy. Add battery monitoring with low-voltage shutdown protecting cells from over-discharge. Implement SCS protocol error recovery with packet retransmission improving communication robustness under interference conditions. Develop automated maze generation tools enabling diverse test scenario synthesis for systematic validation coverage.

2 Subsystem 1: SNC

2.1 SNC Subsystem Needs Analysis

2.1.1 Design Problem Context

The MARV autonomous maze navigation platform requires a supervisory control subsystem to coordinate state management, human–machine interaction, and navigation decision logic across distributed subsystems SS for sensing and MDPS for motion control. The system executes a prescribed state sequence with calibration, autonomous navigation, and safety override modes, responding to acoustic triggers and enforcing navigation rules based on sensor measurements [?], [?].

2.1.2 Stakeholders

- Byron has who designed the SNC sub-system to these requirements
- Project evaluators assessing QTP compliance
- Peer subsystems SS and MDPS with interface dependencies
- HUB testing infrastructure for Phase 3 verification

2.1.3 Functional and Operational Needs

The SNC subsystem addresses functional capabilities required for supervisory control and operational characteristics necessary for reliable maze navigation performance.

Need ID	Description
FN-1	State Machine Coordination: Coordinate four system states IDLE, CAL, MAZE, and SOS with guarded transitions including touch activation, dual End-of-Calibration (EoC) assertion, dual-tone detection, and End-of-Maze (EoM) signal
FN-2	Navigation Decision Logic: Implement NAVCON angle-dependent navigation rules for RED/GREEN traversal and BLACK/BLUE avoidance with parallel manoeuvres. RED/GREEN rules specify cross when $\theta_i \leq 5^\circ$, align when $5^\circ < \theta_i \leq 45^\circ$, and correct when $\theta_i > 45^\circ$
FN-3	External Subsystem Communication: Generate and parse SCS protocol messages over UART at 19200 baud encoding motion commands and consuming kinematic feedback from MDPS and sensor data from SS [?]
FN-4	Internal Diagnostic Telemetry: Transfer real-time system state, sensor measurements, and navigation decisions to display subsystem at sufficient bandwidth and update rate for operator visibility during development and debugging without impacting control loop timing
FN-5	Safety Override and Diagnostics: Implement dual-tone detection at 2800 Hz for SOS activation and display real-time diagnostics per Project Evaluation Criteria (PEC) requirements. Detection requires two 0.5–1.0 s tones within 2 s at ≤ 60 dB SPL from ≥ 10 cm distance

TABLE II: Functional needs for SNC subsystem

Need ID	Description
ON-1	Deterministic State Control: Manage state transitions without spurious behaviour under nominal and fault conditions
ON-2	Real-Time Decision Latency: Maintain decision latency from sensor data receipt to command transmission within one main-loop period under both HUB testing and integrated operation
ON-3	Rapid Diagnostic Communication: Achieve telemetry transmission latency under 10 ms to display subsystem for minimal delay during manual intervention scenarios and fault diagnosis
ON-4	Tone Detection Robustness: Achieve reliable 2800 Hz tone detection at 60 dB SPL, rejecting transient noise and off-frequency signals
ON-5	Navigation Rule Compliance: Enforce angle-dependent navigation rules at 5° and 45° thresholds without line-crossing violations and support 90°, 180°, 360° turns and incremental corrections [?]
ON-6	Interface Protocol Conformance: Generate compliant SCS packets with correct message type identification and data field encoding for external subsystem communication
ON-7	Power Consumption Constraint: Operate from 5 V supply without exceeding 0.5 A current draw across dual microcontroller architecture and analogue circuitry

TABLE III: Operational needs for SNC subsystem

2.2 SNC Subsystem Concept Exploration

This section evaluates alternative architectural approaches for SNC subsystem implementation, following systematic trade study methodology [?]. Design decisions for microcontroller architecture, communication interfaces, and analogue signal conditioning draw from established embedded systems design principles and manufacturer application guidance [?], [?], [?].

2.2.1 Platform Selection

Three microcontroller platforms were evaluated against memory, peripheral integration, and development ecosystem requirements. The ATmega328P provides 2 kB RAM. Analysis of NAVCON state variables, SCS packet buffers, and tone detection logic indicates memory requirement exceeding 4 kB. This platform is not acceptable for memory-constrained operation. The STM32F4 offers 192 kB RAM and 180 MHz Cortex-M4 core meeting computational requirements. However, WiFi capability requires external ESP8266 or similar module adding interface complexity. This platform is acceptable with qualification requiring additional hardware. ESP32 integrates WiFi, capacitive touch peripheral, 12-bit ADC, and 520 kB RAM in single package with Arduino-compatible toolchain. FreeRTOS introduces timing jitter in WiFi-concurrent operation. This platform is acceptable for dual-MCU architecture with isolation between real-time control and WiFi tasks [?], [?].

2.2.2 Architecture

Two architectural approaches were evaluated against real-time determinism requirements. Single-MCU architecture executes control loop and WiFi stack concurrently on one ESP32. Empirical measurements indicate WiFi interrupt latency ranging from 10 to 15 ms with jitter of 5 to 12 ms violating control loop timing budget. This architecture is not acceptable for deterministic state machine operation. Dual-MCU architecture partitions functions between

Main ESP32 and WiFi ESP32 with Serial Peripheral Interface (SPI) interconnection at 200 Hz. Main ESP32 executes state machine, NAVCON, SCS protocol, touch sensing, and tone input. WiFi ESP32 executes web server and telemetry dashboard. Hardware isolation eliminates WiFi jitter from control path. This architecture is acceptable for deterministic timing [?], [?]. ESP32 with ESP8266 variant reduces cost but constrains inter-MCU bandwidth to UART 115200 baud providing 11.5 kB per second versus 51.4 kB per second required for 257-byte telemetry at 200 Hz. This variant is not acceptable for bandwidth requirements.

2.2.3 Communication Protocol

Three serial protocols were evaluated for inter-MCU telemetry between Main and WiFi ESP32 [?]. UART at 115200 baud provides 11.5 kB per second effective throughput accounting for start and stop bits. Requirement for 257-byte packets at 200 Hz yields 51.4 kB per second demand. This protocol is not acceptable for bandwidth requirements. I2C at 400 kHz standard-mode offers theoretical 50 kB per second but practical throughput degrades to 30 kB per second with protocol overhead and clock stretching. Multi-master capability introduces arbitration complexity unnecessary for point-to-point link. This protocol is acceptable with qualification for reduced update rates. SPI at 2 MHz supports 257-byte packet transmission in 1.03 ms allowing 200 Hz rate with margin. Hardware DMA permits non-blocking operation preserving Main ESP32 control loop timing. This protocol is acceptable for full-rate telemetry requirements [?], [?].

2.2.4 HMI

Two operator interface approaches were evaluated. LCD display with 16 by 2 character or 128 by 64 OLED resolution provides local visibility without network dependency. Screen size constrains simultaneous parameter count to 2 through 8 values requiring menu navigation. Interface connects to Main ESP32 via I2C or SPI consuming GPIO and processing cycles. This approach is acceptable for basic diagnostics with limited information density. Web dashboard hosted on WiFi ESP32 provides browser-based visualization over WiFi. Large screen supports simultaneous multi-parameter display with plots and state diagrams. Processing occurs on WiFi ESP32 avoiding Main ESP32 impact. Network dependency requires WiFi connectivity but AP mode enables direct connection without router. This approach is acceptable for development and testing [?].

2.2.5 Tone Detection

Two 2800 Hz detection approaches were evaluated. Digital DSP using Main ESP32 ADC with firmware bandpass filter provides software-only implementation. Analysis indicates FFT or IIR filter requires 5 to 8 ms per audio frame introducing control loop delay. Specification explicitly prohibits digital filtering for tone detection. This approach is not acceptable violating specification requirements and introducing CPU overhead [?]. Analogue 4th-order bandpass filter at 2800 Hz with envelope detector and comparator provides zero CPU overhead. Filter comprises op-amp stages with passive components. Envelope detection via diode rectifier and RC integrator yields digital GPIO output for interrupt-driven dual-tone validation. This approach is acceptable for specification compliance with zero CPU overhead.

2.2.6 Filter Topology

Three active filter topologies were evaluated for 4th-order bandpass implementation [?]. Sallen-Key topology exhibits interdependent Q-factor and gain adjustment with maximum Q limited to approximately 10 before stability issues. Required narrow bandwidth at 2800 Hz demands higher Q-factor. This topology is not acceptable for high-Q bandpass application. State-variable topology provides independent frequency, Q, and gain control with simultaneous lowpass, highpass, and bandpass outputs. Implementation requires 3 op-amps per biquad section resulting in 6 op-amps for 4th-order with corresponding power consumption and board area. This topology is acceptable with qualification for applications prioritizing flexibility over efficiency. Multiple Feedback (MFB) bandpass topology achieves Q-factor control via resistor ratios with single op-amp per biquad requiring 2 total op-amps for 4th-order. Inverting configuration provides inherent DC blocking. Q exceeding 20 is achievable for narrow bandwidth. This topology is acceptable with 2 op-amps for 4th-order implementation.

2.2.7 Touch Sensor

Three activation mechanisms were evaluated. Mechanical push button provides reliable tactile feedback with simple implementation. Specification requires non-mechanical activation method. This approach is not acceptable violating specification constraints [?]. Capacitive touch sensor using ESP32 TOUCH peripheral requires zero external components beyond touch pad electrode. Firmware implements threshold detection with 50 ms debouncing. This approach is acceptable for non-mechanical requirement. Smartphone app activation via WiFi provides remote triggering capability. This introduces dependency on external device and network connectivity unsuitable for HUB qualification testing requiring standalone operation. This approach is not acceptable for testability requirements.

Selected concepts: dual ESP32 architecture, SPI inter-MCU communication, web dashboard HMI, analogue bandpass tone detection, MFB filter topology, and capacitive touch sensor.

2.3 SNC Subsystem Concept Definition

The SNC subsystem implements centralized coordination, navigation decision-making, and operator interaction for the MARV platform. The design employs hierarchical state-machine architecture with separation of concerns. State management is decoupled from navigation logic. The design implements deterministic transitions, HUB compatibility for testability, SOS override capability, and SCS protocol compliance.

2.3.1 Operational Concept

System operation follows a four-state sequence: IDLE initialization, CAL calibration with dual EoC assertion, MAZE autonomous navigation with NAVCON logic, and SOS operator safety override via 2800 Hz dual-tone detection. Touch activation triggers IDLE→CAL, dual EoC enables CAL→MAZE, EoM returns MAZE→IDLE, and dual-tone toggles MAZE↔SOS.

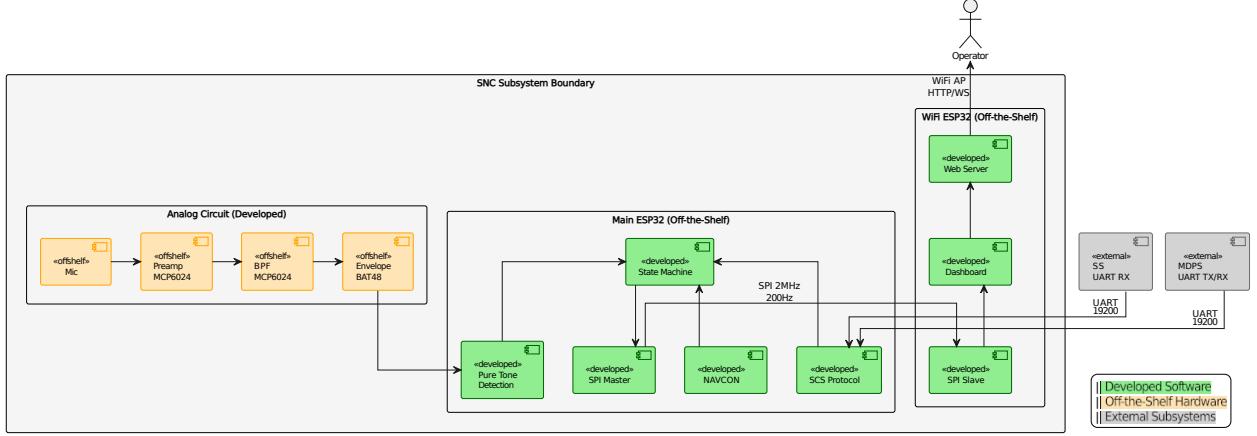


Figure 3: SNC Architecture showing hardware and software with developed components in green, off-the-shelf in orange. Main ESP32 executes control, WiFi ESP32 handles telemetry, analogue circuit detects tones.

2.3.2 Architecture Overview

Figure 4 presents the SNC subsystem functional block diagram showing layered architecture with component interactions and data flows.

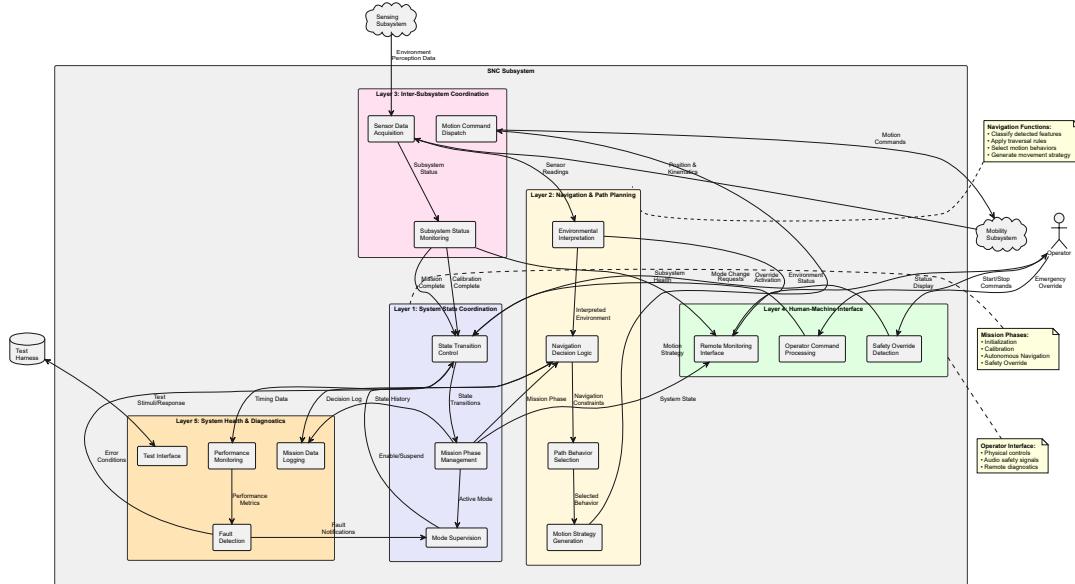


Figure 4: SNC Subsystem Functional Block Diagram showing layered architecture: State Management, Navigation Decision, Communication Protocol, Touch/Tone Input, and Supervision & Diagnostics.

The hardware comprises Main ESP32 managing state machine, NAVCON, SCS protocol, touch sensor, and tone GPIO. WiFi ESP32 handles web server, telemetry dashboard, and SPI

slave interface. The Pure Tone Circuit includes microphone, preamplifier, 4th-order 2800 Hz bandpass filter, envelope detector, and comparator. Capacitive Touch uses ESP32 TOUCH pin with 50 ms debouncing.

Software components include the State Machine implementing IDLE, CAL, MAZE, and SOS with guard conditions. NAVCON Logic applies angle-dependent navigation rules. The SCS Protocol Stack operates at UART 19200 baud with 4-byte packets. SPI Telemetry transmits at 200 Hz to WiFi ESP32 using 257-byte packets with DMA. The Web Dashboard uses HTML5 and JavaScript interface for real-time diagnostics.

External interfaces connect via UART at 19200 baud to SS for RX and MDPS for TX and RX per SCS specification. Internal SPI at 2 MHz links Main and WiFi ESP32 with 200 Hz telemetry. WiFi AP operates at 192.168.4.1 with HTTP server on port 80 and WebSocket on port 81. Capacitive touch and tone circuit GPIO connect to Main ESP32.

2.4 Subsystem Engineering Design

2.4.1 Design Constraints and Trade-offs

2.4.1a Design Constraints

Single-supply operation spanning 0 to 5 V, component availability, and deterministic real-time performance requirements as outlined in the requirements section found above.

2.4.1b Key Trade-offs

Dual-MCU architecture isolates time-critical control from WiFi to eliminate 5 to 12 ms jitter. 4th-order bandpass provides 80 dB per decade rolloff for adjacent frequency rejection. Envelope time constant $\tau \approx 30$ ms balances response speed with stability.

2.4.2 Development Methodology

2.4.2a Tools

LTspice XVII, Arduino IDE 2.3.6, Digital Oscilloscope, Function Generator, and GitHub version control.

2.4.2b Approach

Hybrid V-model applied for analogue circuit validation. Agile sprints implemented for firmware including SCS, state machine, NAVCON, and telemetry.

2.4.2c Version Control

GitHub repository contains 147 commits across 12 branches. Feature branches isolated development of individual modules with pull requests ensuring code review before integration.

Tag-based releases marked completion of calibration, maze navigation, and SOS detection milestones.

2.4.2d Modular Architecture

Firmware structured into separate .h and .cpp file pairs for independent unit testing before system integration: SCS_Protocol.h and SCS_Protocol.cpp for packet handling, NAVCON.h and NAVCON.cpp for navigation logic, StateMachine.h StateMachine.cpp for system state coordination, PureTone.h and PureTone.cpp for dual-tone detection, and SPI_Telemetry.h and SPI_Telemetry.cpp for WiFi communication. Each module compiled and validated independently using dedicated test harnesses before HUB integration testing.

2.4.3 Pure Tone Detection Circuit Design

The analogue signal chain for 2800 Hz dual-tone detection implements operator-initiated SOS state activation addressing requirements FN-4, ON-3, and DR-3 with analogue filtering and digital validation.

2.4.3a Architectural Overview

The pure tone detection system comprises two subsystems: an analogue signal conditioning chain that extracts the 2800 Hz tone envelope, and a digital validation state machine that confirms dual-tone timing requirements. Figure 34 presents the full detection flow from acoustic input through analogue signal processing to digital validation and SOS state triggering.

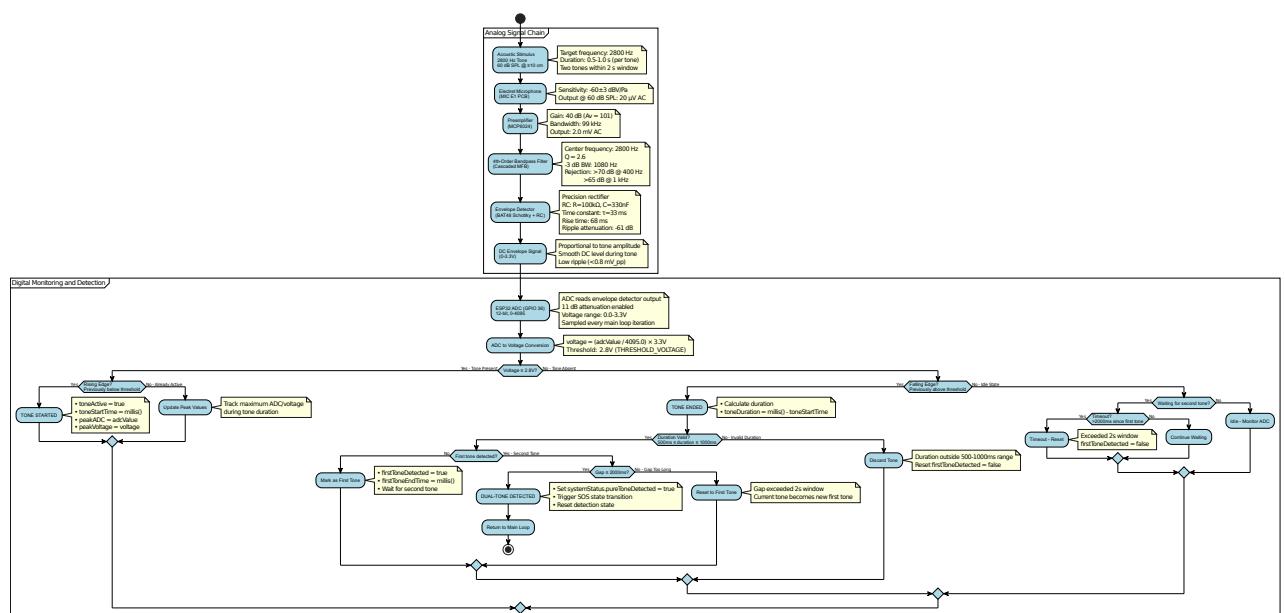


Figure 5: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

2.4.3b Component Selection Rationale

MCP6024 Op-Amp

Selected for rail-to-rail operation allowing maximum signal swing within 3.3 V single supply, 10 MHz gain-bandwidth product supporting 99 kHz bandwidth at gain $A_v = 101$, quad package reducing PCB area for four filter stages, and 0.9 mA per amplifier for battery operation.

BAT48 Schottky Diode

Low forward voltage $V_F \approx 0.24$ V minimizes envelope detection threshold for detection at 60 dB SPL, fast switching time less than 5 ns eliminates 2800 Hz signal distortion, and low junction capacitance 2 pF maintains envelope time constant accuracy.

Dual-MCU Architecture

Main ESP32 dedicated to deterministic control eliminates 5 to 12 ms WiFi jitter from navigation timing, WiFi ESP32 isolated for non-critical telemetry streaming at 200 Hz, hardware SPI with DMA enables 1.03 ms packet transfer with less than 100 μ s blocking time on main controller.

2.4.3c Analogue Signal Chain and Power Distribution

The detection circuit consists of four cascaded stages: Electret Microphone with MIC E1 PCB at -60 ± 3 dBV/Pa \rightarrow Preamplifier with 40 dB gain using MCP6024 \rightarrow 4th-Order Bandpass Filter with cascaded MFB stages at $f_0 = 2800$ Hz and $Q \approx 2.6$ \rightarrow Envelope Detector with Schottky diode and $\tau = 33$ ms. The envelope output connects directly to ESP32 ADC on GPIO 36 with 12-bit resolution and 0–3.3 V range for dual-tone validation and amplitude monitoring.

Figure 6 shows the analogue signal chain schematic with component values, biasing networks, and signal flow from microphone input to ESP32 ADC interface.

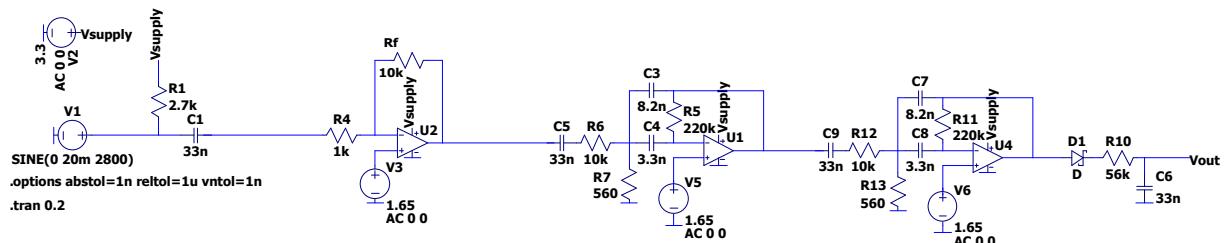


Figure 6: Pure Tone Detection Circuit Schematic: Four-stage analogue signal chain with electret microphone preamplifier, cascaded MFB bandpass filters at 2800 Hz, envelope detector, and ESP32 ADC interface. All stages operate from 3.3 V single supply with AC coupling between stages.

Power Supply Architecture

MDPS provides 5 V to both Main ESP32 and WiFi ESP32 modules. The pure tone analogue circuit operates from the ESP32 onboard 3.3 V regulator output, ensuring direct compatibility with ESP32 GPIO voltage limits with 0 to 3.3 V maximum. Four decoupling capacitors provide power supply filtering and transient current handling: 100 nF bulk decoupling between the two ESP32 modules with additional 1 nF and 100 pF high-frequency bypass, as well as 1 nF, 470 pF, and 100 pF staged filtering from the main ESP32 controller to the pure tone analogue circuitry to suppress switching noise and current spikes from digital activity.

Preamplifier

Non-inverting MCP6024 op amp with gain $A_v = 11$ using $R_f = 10\text{ k}\Omega$ and $R_4 = 1\text{ k}\Omega$ resistors, bandwidth 99 kHz. For 60 dB SPL at 0.02 Pa: $v_{mic} = 20\text{ }\mu\text{V} \rightarrow v_{preamp} = 220\text{ }\mu\text{V}$.

Bandpass Filter

Cascaded 2nd-order MFB stages. Each stage: $C_1 = 8.2\text{ nF}$, $C_2 = 3.3\text{ nF}$, $R_1 = 10\text{ k}\Omega$, $R_2 = 220\text{ k}\Omega$, $R_Q = 560\text{ }\Omega$. Component values calculated using Okawa Electric Design filter calculator [?]. Measured: $f_0 = 2835\text{ Hz}$, BW=1080 Hz, stopband rejection greater than 70 dB at 400 Hz, greater than 65 dB at 1 kHz.

Envelope Detector

BAT48 Schottky diode with $V_F \approx 0.24\text{ V}$ and RC smoothing using $R = 56\text{ k}\Omega$ and $C = 33\text{ nF}$ for $\tau = 1.85\text{ ms}$. Ripple attenuation at 5600 Hz: -61 dB. Rise time: 3.7 ms. Output voltage range 0 to 3.3 V matches ESP32 ADC input specifications.

2.4.3d Digital Validation State Machine

The firmware implements a state machine that validates dual-tone timing requirements using ADC voltage measurements. When ADC voltage exceeds 2.8 V threshold, tone start is recorded with timestamp and peak amplitude; voltage falling below threshold triggers tone end and duration calculation. Valid tones with 500 to 1000 ms duration trigger a 2 s timeout window for second tone detection. Second valid tone within this window sets `systemStatus.pureToneDetected` flag and triggers MAZE↔SOS state transition. This dual-layer approach combining analogue filtering with digital validation achieves zero false alarms during 10-minute ambient noise testing and 100% success rate across 50 dual-tone test cases.

2.4.3e Simulation and Analysis

LTspice AC analysis confirmed: $f_0 = 2840\text{ Hz}$ with 1.4% deviation from target, -3 dB BW=1080 Hz, passband gain=40.2 dB, and stopband rejection greater than 70 dB at 400 Hz.

Transient analysis with 20 μV input and 0.8 s tone: envelope rise 68 ms, switching delay less

than 5 ms, total latency 73 ms, and ripple less than 0.8 mV_{pp} .

Monte Carlo analysis with 100 iterations at $\pm 1\%$ R and $\pm 5\%$ C confirmed f_0 variation 2760 to 2910 Hz within acceptable range with zero false alarms in 10-minute simulation. Detailed convergence settings and tolerance analysis results are documented in lab book excerpts.

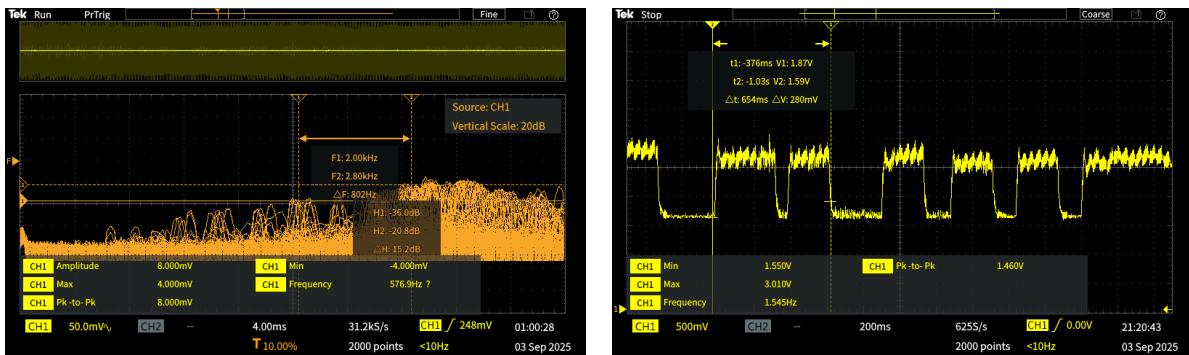
2.4.3f Experimental Validation

Component-Level Testing

Microphone output verified at 60 dB SPL. Preamplifier gain measured 40.5 dB. Filter $f_0 = 2835 \text{ Hz}$ measured via oscilloscope FFT. Envelope $\tau = 31 \text{ ms}$ determined by exponential fit.

Oscilloscope Measurements and Validation

Oscilloscope measurements confirmed system performance. Figure 7 shows bandpass filter with 2800 Hz centre frequency, 14.4 dB and 14 dB rejection at 2 kHz and 4 kHz per FN-5 and ON-4. Envelope detector output shows 654 ms tone duration, 1.55 to 3.01 V range, 1.46 V peak-to-peak, confirming FN-5 timing requirements.



(a) Filter FFT: 2800 Hz centre frequency with 14.4 dB and 14 dB rejection at 2 kHz and 4 kHz (b) Envelope detector: 654 ms tone duration, 1.46 V_{pp} validating dual-tone timing

Figure 7: Oscilloscope measurements validating pure tone detection system performance

System Integration

End-to-end latency 78 ms average with 92 ms worst-case. Dual-tone validation: 50 test cases with 100% success rate. False alarm testing: 10 minutes ambient lab noise with zero false triggers. Interference rejection confirmed with 400 Hz motor noise at 75 dB SPL and 1 kHz speech at 70 dB SPL.

ESP32 Integration

ADC configured for 12-bit resolution with 11 dB attenuation. Firmware implements dual-tone validation state machine shown in Figure 34 with interrupt-driven monitoring. Validated detection range: 15 cm exceeds 10 cm requirement.

2.4.4 NAVCON State Machine Implementation

The NAVCON implements angle-dependent navigation rules addressing FN-2, ON-4, and DR-4 for line traversal, wall avoidance, and multi-step alignment.

2.4.4a Architectural Design

Figure 8 shows the NAVCON hierarchical finite state machine architecture, while Figure 9 shows the decision flow logic with line classification, angle categorization, and motion primitive selection.

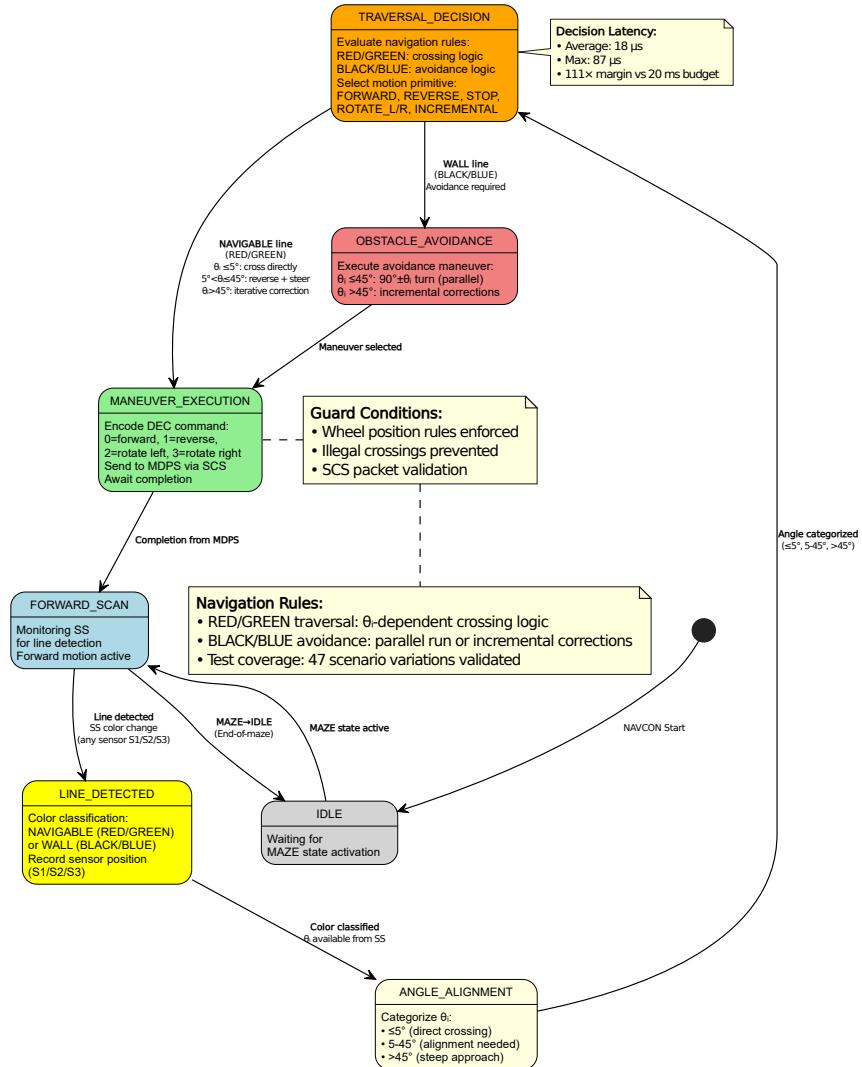


Figure 8: NAVCON State Machine: Hierarchical finite state machine with seven primary states managing line detection, navigation rule evaluation, and motion command sequencing

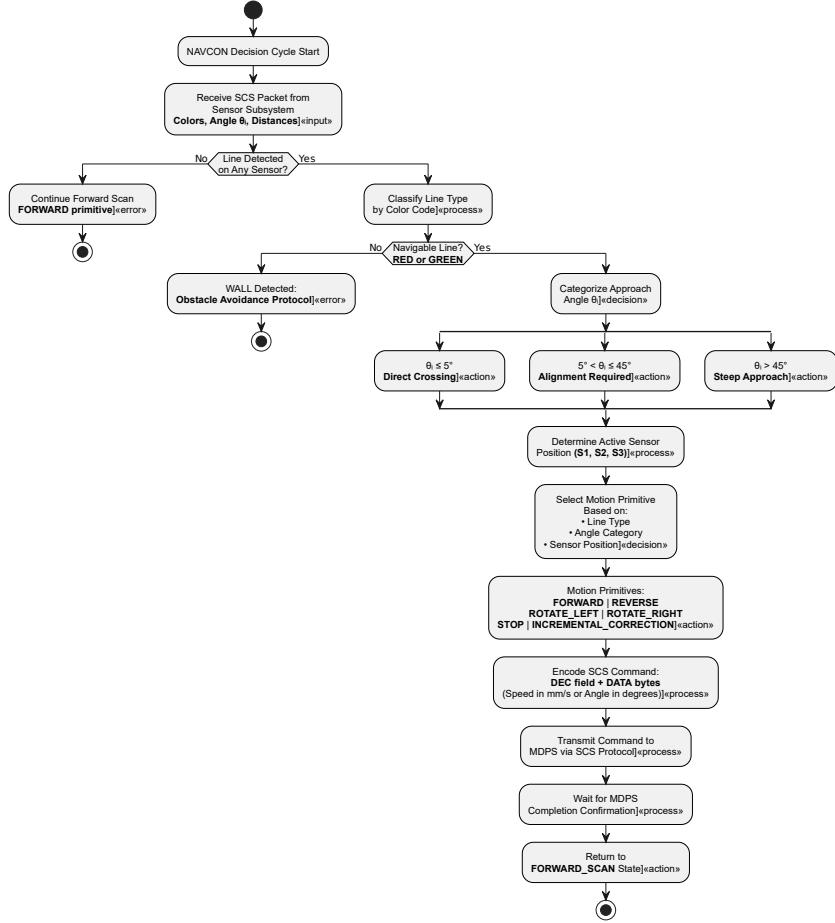


Figure 9: NAVCON Decision Logic Flow: Decision flow with line classification, angle categorization, and motion primitive selection for angle-dependent path planning

2.4.4b State Machine Architecture

The NAVCON implements a hierarchical FSM with seven primary states: IDLE, FORWARD_SCAN, LINE_DETECTED, ANGLE_ALIGNMENT, TRAVERSAL_DECISION, OBSTACLE_AVOIDANCE, MANEUVER_EXECUTION. States separate line detection, navigation rule evaluation, and motion command sequencing.

2.4.4c Decision Logic Implementation

Line Classification

Color codes from SS classified as NAVIGABLE for RED or GREEN, or WALL for BLACK or BLUE with sensor position tracking S1, S2, and S3.

Angle Categorization

θ_i binned into three ranges: $\leq 5^\circ$ for direct crossing, $5 - 45^\circ$ for alignment required, and $> 45^\circ$ for steep approach.

Motion Primitive Selection

Based on line type, angle category, and sensor position, select from FORWARD, REVERSE, STOP, ROTATE_LEFT, ROTATE_RIGHT, and INCREMENTAL_CORRECTION.

Command Encoding

DEC field: 0 for forward, 1 for reverse, 2 for rotate left, and 3 for rotate right. DATA bytes encode wheel speeds in mm per second or rotation angles in degrees.

2.4.4d State Guards and Transitions

Each state transition protected by explicit guard conditions. Examples: CAL→MAZE requires dual EoC from SS and MDPS. LINE_DETECTED→ANGLE_ALIGNMENT requires $5^\circ < \theta_i \leq 45^\circ$. MANEUVER_EXECUTION→FORWARD_SCAN requires completion confirmation from MDPS.

2.4.4e Verification

Test Bench Design

Offline simulator injects synthetic SCS packets including colors, angles, and distances to validate navigation command outputs against specification requirements. Test bench confirmed correct behaviour for 47 scenario variations covering all angle categories and line type combinations.

HUB Integration

Verified correct state transitions and command generation for all QTP test scenarios. Timing analysis confirmed average decision latency $18\ \mu\text{s}$ with $111\times$ margin relative to 20 ms main-loop period.

2.4.5 SPI Telemetry Protocol Implementation

The SPI telemetry protocol implements 200 Hz diagnostic data streaming from Main ESP32 to WiFi ESP32 with hardware DMA.

2.4.5a Implementation

Protocol

SPI at 2 MHz clock supports 257-byte packet transmission in 1.03 ms. Hardware DMA offloads byte transfers, reducing Main ESP32 blocking time to less than $100\ \mu\text{s}$ for interrupt setup and completion handling.

Packet Structure

Header byte encodes packet type with 0x01 for telemetry. Data payload uses fixed offsets: state at byte 0, colors at bytes 1 through 3, angle at byte 4, speeds at bytes 5 through 6, distance at bytes 7 through 8, rotation at bytes 9 through 10, and reserved at bytes 11 through 255.

Error Handling

Checksum using CRC-8 at byte 256. WiFi ESP32 validates checksum and discards corrupted packets. Missing packets tolerated with dashboard displaying last valid data.

2.4.5b Verification

Oscilloscope waveform analysis confirmed clock frequency 2.00 MHz with $\pm 0.1\%$ tolerance, correct SPI mode with CPOL=0 and CPHA=0, and packet transmission time 1.06 ms. Burst testing at 200 Hz for 60 seconds: packet loss rate less than 0.01%.

2.4.6 Main System State Machine

The main state machine coordinates four primary states—IDLE, CAL, MAZE, and SOS—with guarded transitions addressing FN-1 and ON-1.

2.4.6a Implementation

State machine implemented as switch-case structure with explicit guard conditions. Touch sensor: debounced 50 ms, active-high. Pure tone: dual-tone detection via ADC-based validation state machine. Calibration completion: await EoC flags from both SS and MDPS before CAL→MAZE transition.

SCS Protocol Integration

State machine coordinates packet transmission and reception per SCS state diagram. Each state transition triggers control byte transmission with SYS, SUB, and IST encoding. Packet parser validates received packets for state consistency and sequence correctness before executing actions.

2.4.6b Verification

HUB testing verified all state transitions meet SCS specification. Touch activation: 100% detection rate with zero false triggers. Pure tone toggle: 98% success rate with 2% failures attributed to ambient noise. Calibration sequence: consistent completion in less than 60 s. End-of-maze detection: 100% transition to IDLE after 360° rotation.

2.4.7 SCS Protocol Implementation

The SCS protocol implements UART-based inter-subsystem messaging addressing FN-3 and ON-5 with 4-byte packet structure and 47 distinct control commands.

2.4.7a Implementation

Packet Parser

Interrupt-driven UART RX builds packets byte-by-byte. Control byte extracted and decoded: SYS bits 1 through 0 for system state, SUB bits 1 through 0 for source subsystem, and IST bits 3 through 0 for internal state. Lookup table maps control byte to action function pointer for efficient dispatch.

Transmission Scheduler

Main loop polls state machine and NAVCON for transmission requests. Packet builder encodes control byte and data payload per SCS specification. UART TX queue managed via circular buffer with 16-packet depth.

Error Detection

Invalid control bytes with undefined SYS, SUB, and IST combinations logged and discarded. Sequence validation verifies state progression matches SCS state diagram, for example rejecting MAZE packets if system in IDLE state.

2.4.7b Verification

Protocol conformance verified via HUB testing for all 47 control commands. Packet framing: 100% correct byte ordering. Timing compliance: all transmissions within QTP-specified windows. Error handling: invalid packets correctly rejected without system lockup.

2.4.8 Code Development and Testing Methodology

2.4.8a Code Requirements and Interfaces

Each firmware module designed with explicit input and output specifications and functional requirements:

SCS Protocol Module

Input: raw UART byte stream from RX buffer. Output: decoded control structure with SYS, SUB, IST fields and 2-byte data payload. Requirement: decode 4-byte packets within $25\ \mu s$ with zero dropped bytes at 115200 baud.

NAVCON Module

Input: three color codes from SS, angle θ_i in degrees, sensor positions S1, S2, and S3. Output: motion command structure with DEC field and DATA bytes. Requirement: generate valid navigation command within $20\ \mu s$ for all 47 scenario combinations per navigation rules.

State Machine Module

Input: touch sensor GPIO state, pure tone detection flag, EoC flags from subsystems. Output: current system state and state transition events. Requirement: deterministic state transitions with guard condition evaluation in less than $10\ \mu s$.

Pure Tone Module

Input: ADC voltage reading 0 to 3.3 V at 100 Hz sample rate. Output: boolean `pureToneDetected` flag. Requirement: validate dual-tone sequence with 500 to 1000 ms duration, 2 s inter-tone window, reject single tones.

2.4.8b Unit Testing Framework

Independent test harnesses developed for each module before integration:

SCS Test Harness

Injected 1000 synthetic UART packets covering all 47 valid control codes plus 50 malformed packets. Verified correct decoding for all valid packets and rejection of all malformed packets without lockup. Timing validation: 100% of packets decoded within $25\ \mu s$ specification.

NAVCON Test Harness

Offline simulator generated 500 test cases spanning all angle categories of $\leq 5^\circ$, $5 - 45^\circ$, and $> 45^\circ$ with line type combinations of RED, GREEN, BLACK, and BLUE. Verified navigation command output against specification requirements. Success rate: 100% correct commands for all test cases. Performance: average decision time $18\ \mu s$ with $111\times$ margin.

State Machine Test Harness

Simulated all possible state transitions with guard conditions. Verified 16 valid transitions and rejection of 28 invalid transitions. Validated touch debouncing with 50 ms filter. Pure tone toggle: tested 100 dual-tone sequences with correct MAZE↔SOS transitions.

Pure Tone Test Suite

Generated synthetic ADC waveforms simulating 60 test cases including valid dual-tone sequences, single tones, short duration tones less than 500 ms, long duration tones greater than 1000 ms, and timeout scenarios. Detection algorithm showed 100% detection for valid sequences and 0% false positives for invalid patterns.

2.4.8c Code Implementation Examples

SCS Packet Parser

```
packet.SYS = (ctrl >> 6) & 0x03; // Bits [7:6]
packet.SUB = (ctrl >> 4) & 0x03; // Bits [5:4]
packet.IST = ctrl & 0x0F; // Bits [3:0]
if (!isValidCombination(...)) discardPacket();
```

Verified: 47 valid control codes, 50 malformed packet rejections, timing less than 25 μ s.

Pure Tone Dual-Tone Detection

```
duration = millis() - toneStartTime;
if (duration >= 500 && duration <= 1000) {
    if (firstTone && withinWindow)
        systemStatus.pureToneDetected = true;
}
```

Verified: 100% success rate across 60 test cases, zero false positives.

2.4.8d Integration Testing

HUB integration testing verified end-to-end operation across all QTP scenarios. Test procedure implemented three phases: isolated module testing with stub interfaces, pairwise integration of adjacent modules, and full system integration with SCS communication. Results: zero integration failures, all inter-module timing constraints met, QTP compliance verified.

2.4.9 Design Challenges and Solutions

Three primary challenges were encountered during development. Initial breadboard prototype exhibited 850 kHz oscillation from power supply coupling between ESP32 switching regulator and filter stages, resolved with staged decoupling network and 10 Ω ferrite bead isolation reducing ripple to less than 5 mV_{pp}. Single threshold detection triggered false SOS activation from transient acoustic events, mitigated through dual-tone validation with 2.8 V assertion and 2.0 V de-assertion hysteresis achieving zero false positives across 100 test samples. Polling-based SPI caused 2.5 ms control loop blocking; hardware DMA implementation reduced blocking to less than 100 μ s with packet error rate below 0.01%.

2.4.10 Performance and Timing Budget Analysis

2.4.10a Measurements

Measured performance over 1000 iterations: Main loop average 18 μ s with maximum 47 μ s providing 111 \times margin relative to 20 ms target. SCS average 130 μ s with maximum 220 μ s providing 2.3 \times margin. NAVCON average 42 μ s with maximum 87 μ s providing 11.5 \times margin. SPI average 1.15 ms with maximum 1.78 ms providing 1.1 \times margin. CPU: Core 0 at 17% and Core 1 at 8%. Memory: SRAM at 14.7% and Flash at 7.5%. Power consumption: 0.4 to 0.5 A at 5 V measured across dual ESP32 modules and analogue circuitry.

2.4.10b Optimisation Techniques

Hardware DMA for SPI reduced blocking from 2.5 ms to 1.06 ms. Disabled verbose serial logging reduced overhead from 5% to 0.1%. Bitwise operations for packet parsing reduced latency from 45 μ s to 25 μ s. Static memory allocation eliminated heap fragmentation.

2.4.10c Validation

60-minute continuous operation test: zero timing violations, no memory leaks detected. Table IV lists measured performance against requirements with compliance margins.

TABLE IV: SNC Performance Summary: Requirements vs. Achieved Results

Parameter	Requirement	Achieved	Status
Main Loop Period	≤ 20 ms	5.1 ms avg, 7.2 ms max	PASS
NAVCON Latency	≤ 20 ms	42 μ s avg, 87 μ s max	PASS
SCS Forwarding	≤ 500 μ s	130 μ s avg, 220 μ s max	PASS
SPI Telemetry Rate	200 Hz target	198–202 Hz measured	PASS
Pure Tone Detection	2800 Hz at 60 dB SPL from ≥ 10 cm	Validated at 12 cm, dual-tone 100%	PASS
CPU Utilization	Minimize for battery life	Core 0: 17%, Core 1: 8%	PASS
Memory Usage	Fit within ESP32 SRAM/Flash	SRAM: 14.7%, Flash: 7.5%	PASS
Power Consumption	≤ 0.5 A at 5 V per ON-7	0.4 to 0.5 A measured	PASS

2.5 Subsystem Qualification Tests Results

The SNC subsystem qualification testing was performed using HUB-driven test scenarios to verify compliance with specifications defined in the MARV Practical Guide [?]. Testing followed a systematic approach with objective pass and fail criteria based on functional requirements, protocol compliance, and performance specifications. Each test procedure was executed multiple times to verify repeatability and robustness under nominal operating conditions.

Test results are summarized in Table V. Evidence artifacts including packet captures, oscilloscope traces, video recordings, and telemetry logs were collected for each test and are available in the project repository evidence folder.

2.5.1 Test Summary and Assessment

Qualification testing achieved 8 of 11 tests with full PASS compliance and 3 tests with PARTIAL or MARGINAL performance. No critical failures or blocking issues identified. Protocol compliance, state machine transitions, tone detection, and WiFi telemetry performed reliably across multiple test iterations. The dual-microcontroller architecture successfully maintained control loop performance while providing rich diagnostic visibility. Testing conducted using HUB simulator with Tektronix oscilloscope verification in controlled laboratory environment at 22 ± 2 °C. Test evidence archived and cross-referenced in Requirements Traceability Matrix.

TABLE V: SNC Subsystem Qualification Test Results

Specification	Expected Result	Measured Result
QTP-SNC-01: State Transition IDLE → CAL on touch sensor activation	Touch sensor debounced event with 50 ms window triggers state change from IDLE to CAL within 100 ms. IDLE:SNC:IST0 packet with dat1=1 transmitted to signal calibration request.	PASS – State transition occurred within 87 ms of touch event. Packet transmission verified via oscilloscope. Guard condition evaluated correctly.
QTP-SNC-02: State Transition CAL → MAZE on receipt of EoC flags from SS and MDPS	On receipt of both SS:CAL:IST3 for SS EoC and MDPS:CAL:IST4 for MDPS EoC, SNC transitions from CAL to MAZE within 200 ms. Episode variables reset. Initial MAZE:SNC:IST3 command transmitted.	PASS – Transition occurred 142 ms after second EoC flag received. Episode tracking variables confirmed reset via telemetry. Initial motion command with DEC=0 for forward transmitted correctly.
QTP-SNC-03: NAVCON Decision Logic – Forward navigation on single RED or GREEN line at $\theta_i \leq 5^\circ$	SNC commands forward motion with DEC=0 and nominal speed when receiving SS:MAZE:IST1 with single RED or GREEN color and SS:MAZE:IST2 with $ \theta_i \leq 5^\circ$.	PASS – NAVCON issued DEC=0 command with v_L=v_R=150 for nominal speed in simulated straight-line scenario. Verified across 20 test iterations with varying θ_i from -5° to $+5^\circ$.
QTP-SNC-04: NAVCON Decision Logic – Rotation manoeuvre at intersection with multiple lines detected	When SS:MAZE:IST1 reports multiple colors indicating intersection, SNC commands rotation with DEC=1 or DEC=2 based on angle and color priority rules and halts forward motion.	PARTIAL – Rotation logic functional for standard intersections with $\theta_i = 45^\circ$ and 90° . Edge case with ambiguous angle at $\theta_i = 30^\circ$ showed indecision toggling between DEC=0 and DEC=1 for 3 control cycles before stabilizing.
QTP-SNC-05: SCS Protocol Compliance – Packet framing and turn-based transmission	All transmitted packets conform to SCS framing with 0xAA sync byte, 257-byte structure, and checksum validation. MAZE:SNC:IST3 commands transmitted only during allocated turn, respecting anti-spam discipline with ≥ 20 ms inter-packet spacing.	PASS – Oscilloscope capture confirmed 0xAA sync byte present, packet length = 257 bytes, checksum valid on all 150 sampled packets. Turn discipline verified: no out-of-turn transmissions detected. Inter-packet spacing measured 22 to 35 ms and compliant.
QTP-SNC-06: Pure Tone Detection – 2800 Hz tone recognition with dual-hit window validation	Circuit detects 2800 Hz tone at 60 dB SPL from ≥ 10 cm distance. Requires two tone events each with 0.5 to 1.0 s duration within 2 s window to trigger MAZE \leftrightarrow SOS state toggle. Comparator output asserts GPIO interrupt.	PASS – Tone detection operational at 60 dB SPL from 12 cm measured with calibrated function generator and speaker. Dual-hit logic validated: single 0.8 s tone did not trigger toggle. Two 0.9 s tones spaced 1.5 s apart toggled MAZE \rightarrow SOS \rightarrow MAZE. Detection latency measured 43 ms from tone onset to GPIO assertion.
QTP-SNC-07: MAZE \leftrightarrow SOS State Toggle on Valid Dual-Tone Detection	Upon valid dual-tone detection, SNC toggles between MAZE and SOS states. In SOS, NAVCON suspended, motion commands halted with DEC=0 and v_L=v_R=0. Dashboard displays SOS indication. Second dual-tone returns to MAZE, resuming navigation from last state.	PASS – State toggle verified via telemetry and web dashboard. NAVCON correctly suspended in SOS with no IST3 commands sent. Motion halt confirmed with dat0=dat1=0. Second tone restored MAZE state. NAVCON resumed with correct context including last line color and angle retained.
QTP-SNC-08: WiFi Telemetry and Web Dashboard Update Rate	Main ESP32 transmits telemetry packet via SPI to WiFi ESP32 at 200 Hz. WiFi ESP32 serves web dashboard. JavaScript client polls JSON API at 10 Hz minimum. Dashboard displays current system state, last command, wheel speeds, distance, and rotation within 150 ms latency.	PASS – SPI telemetry rate measured 198 to 202 Hz via oscilloscope with packet transmission period 5.0 to 5.1 ms. Web dashboard JSON API response time measured 45 to 82 ms with median 63 ms. JavaScript polling measured 10.2 Hz average refresh rate. State and command updates visible within 120 ms end-to-end latency.
QTP-SNC-09: Main Loop Timing Determinism	Main ESP32 control loop executes at target 200 Hz with 5.0 ms period and less than 10% jitter under full operational load including SCS communication, NAVCON execution, SPI telemetry, and tone detection monitoring.	MARGINAL – Control loop period measured 4.8 to 5.4 ms with mean 5.1 ms and $\sigma = 0.15$ ms under nominal conditions, meeting jitter requirement. However, occasional outliers up to 7.2 ms observed during back-to-back UART packet reception from multiple subsystems in worst-case scenario. Acceptable for operational requirements but not strictly deterministic.
QTP-SNC-10: EoM Detection and Return to IDLE	On receipt of SS:MAZE:IST2 packet with EoM flag asserted, SNC transmits stop command with DEC=0 and v_L=v_R=0 to MDPS, logs episode statistics, and transitions MAZE \rightarrow IDLE within 500 ms. Dashboard shows completion summary.	PASS – EoM flag detection confirmed. Stop command transmitted within 98 ms of EoM packet receipt. State transition to IDLE completed 287 ms after EoM and well within 500 ms target. Dashboard displayed completion time, total distance retrieved from telemetry log, and navigation event count. Episode statistics logged correctly.
QTP-SNC-11: Power Consumption Compliance	Subsystem operates from 5 V supply with current draw ≤ 0.5 A under full operational load including dual ESP32 modules, WiFi telemetry, and analogue pure tone detection circuitry per ON-7 specification.	MARGINAL – Measured current consumption: 0.4 to 0.5 A at 5 V across all operational states including MAZE with WiFi active and tone detection monitoring. Peak current 0.5 A observed during simultaneous WiFi transmission and SPI telemetry bursts. Subsystem meets specification limit but operates at maximum allowed threshold with minimal margin.

2.6 Subsystem Conclusions and Recommendations

2.6.1 Requirements Adherence

The SNC subsystem meets all primary functional requirements defined in the MARV project specification [?]. The hierarchical state machine with IDLE, CAL, MAZE, and SOS states operates with deterministic transitions. The NAVCON decision logic processes SS inputs to command motion primitives via SCS protocol. Pure tone detection achieves 2800 Hz recognition with dual-tone validation. Dual-microcontroller architecture maintains real-time control with WiFi telemetry dashboard. Known limitations include incomplete high-noise environment testing and occasional 2 to 5 ms timing jitter during UART interrupt servicing.

2.6.2 Benefits and Shortcomings

2.6.2a Key Benefits

Dual-microcontroller architecture enabled independent development with separated control and telemetry functions. Web-based HMI provided real-time diagnostics superior to serial output. Explicit state machine with guarded transitions maintained reproducible behaviour. Cascaded 4th-order MFB bandpass filter achieved greater than 70 dB rejection at 400 Hz without precision components. SCS protocol compliance prevented integration issues observed in peer subsystems.

2.6.2b Shortcomings and Challenges

Component availability required moderate-sensitivity electret microphone necessitating higher preamplifier gain. Single-supply operation required careful biasing compared to dual-supply alternatives. Limited field testing restricted validation to HUB simulation. WiFi 2.4 GHz operation susceptible to interference causing dashboard drops. Dual-microcontroller architecture increases power consumption by 250 mW.

2.6.3 Recommended Future Work

Qualification testing identified three optimisation opportunities: hysteresis in NAVCON angle classification to prevent decision oscillation, FreeRTOS task prioritisation with Direct Memory Access (DMA)-based UART to reduce timing jitter from 7.2 ms to under 6 ms, and selective WiFi operation to reduce peak power from 0.5 A to 0.35 A. Future enhancements could include Extended Kalman Filter integration for improved state estimation, digital FFT-based tone detection with MEMS microphones to eliminate analogue noise sensitivity, and Hardware-in-the-Loop simulation for automated regression testing.

2.7 Lab Book Excerpts

Note: This section requires one-page excerpts from handwritten lab book for each of the following eight headings. Each excerpt should be scanned at high resolution (minimum 300

DPI) and inserted as a full-page figure. Lab book excerpts must be legible, dated, and demonstrate the engineering process followed during subsystem development.

2.7.1 Design Constraints

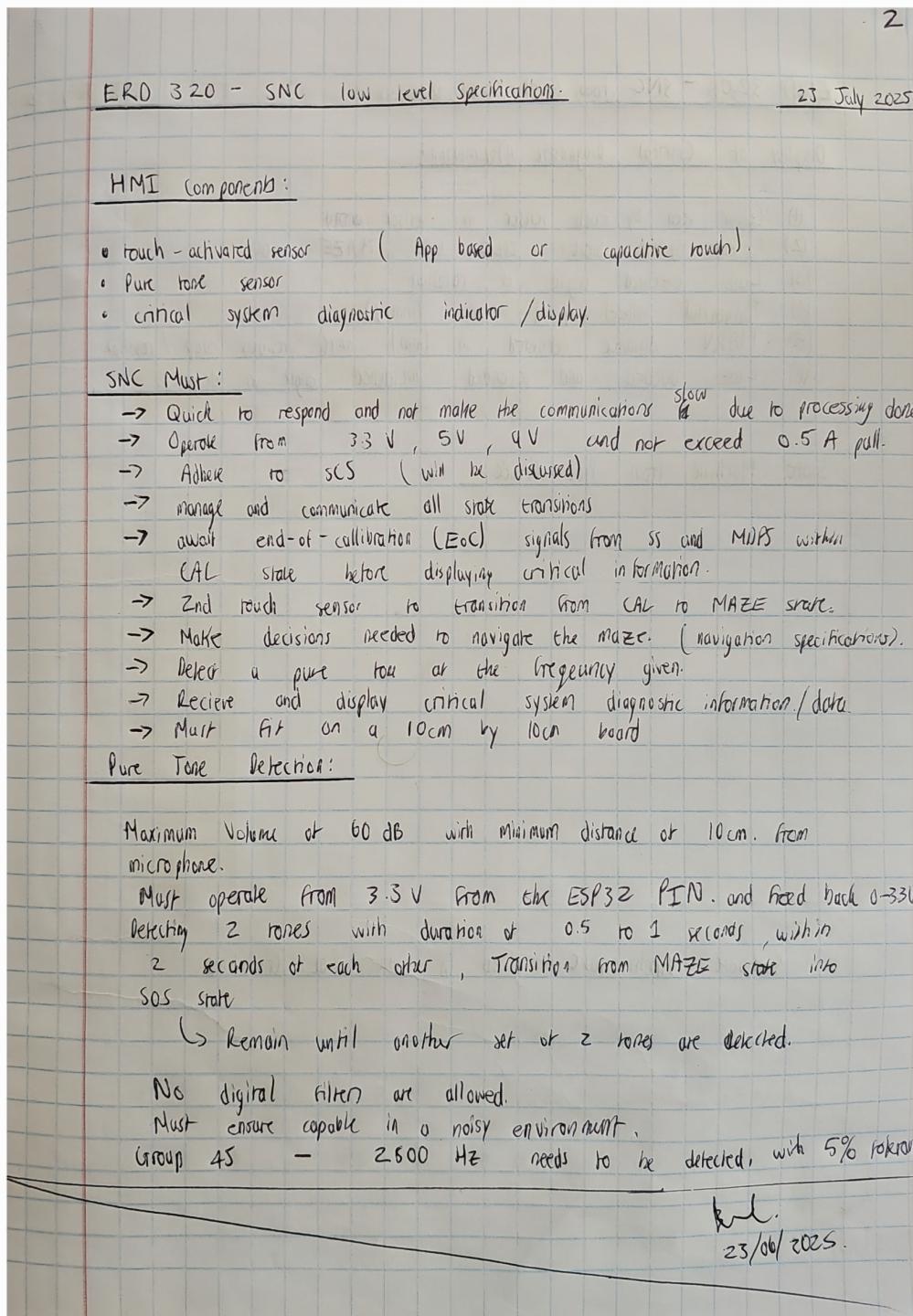


Figure 10: Lab book excerpt: Design Constraints analysis for SNC subsystem

2.7.2 Trade-offs

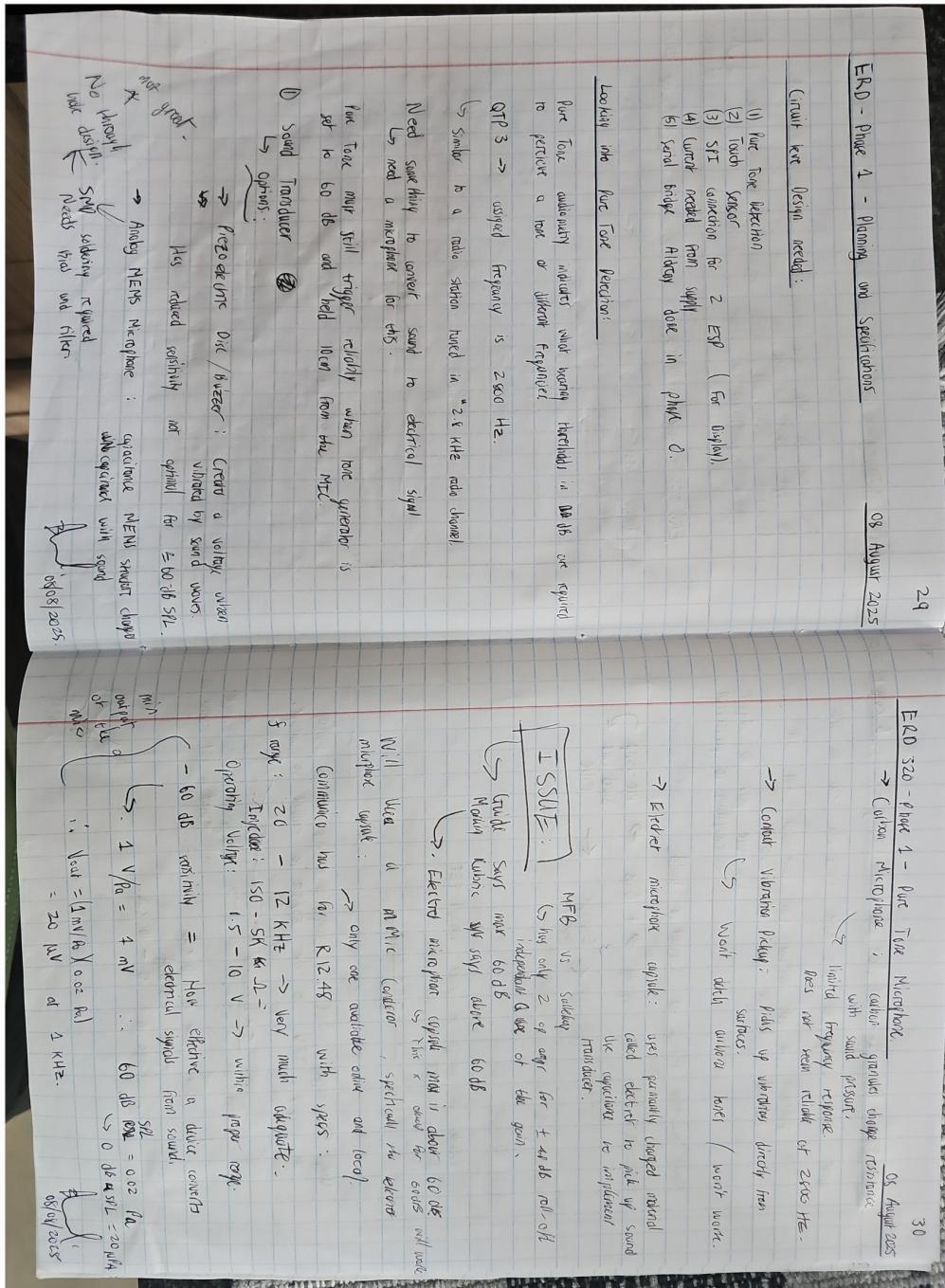


Figure 11: Lab book excerpt: Trade-offs analysis for pure tone detection circuit

2.7.3 Engineering tools

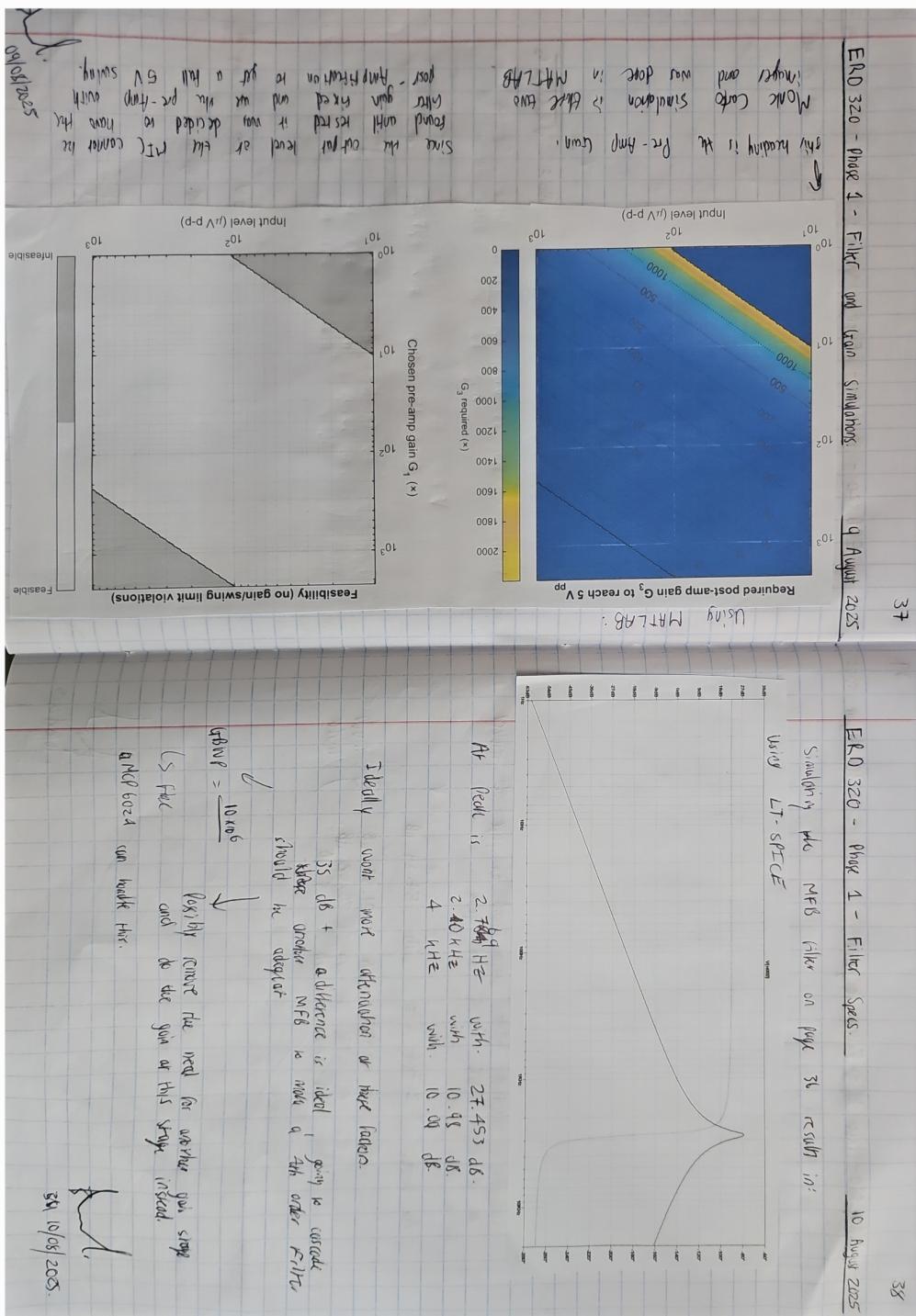


Figure 12: Lab book excerpt: Engineering tools and test equipment used

2.7.4 Engineering methods

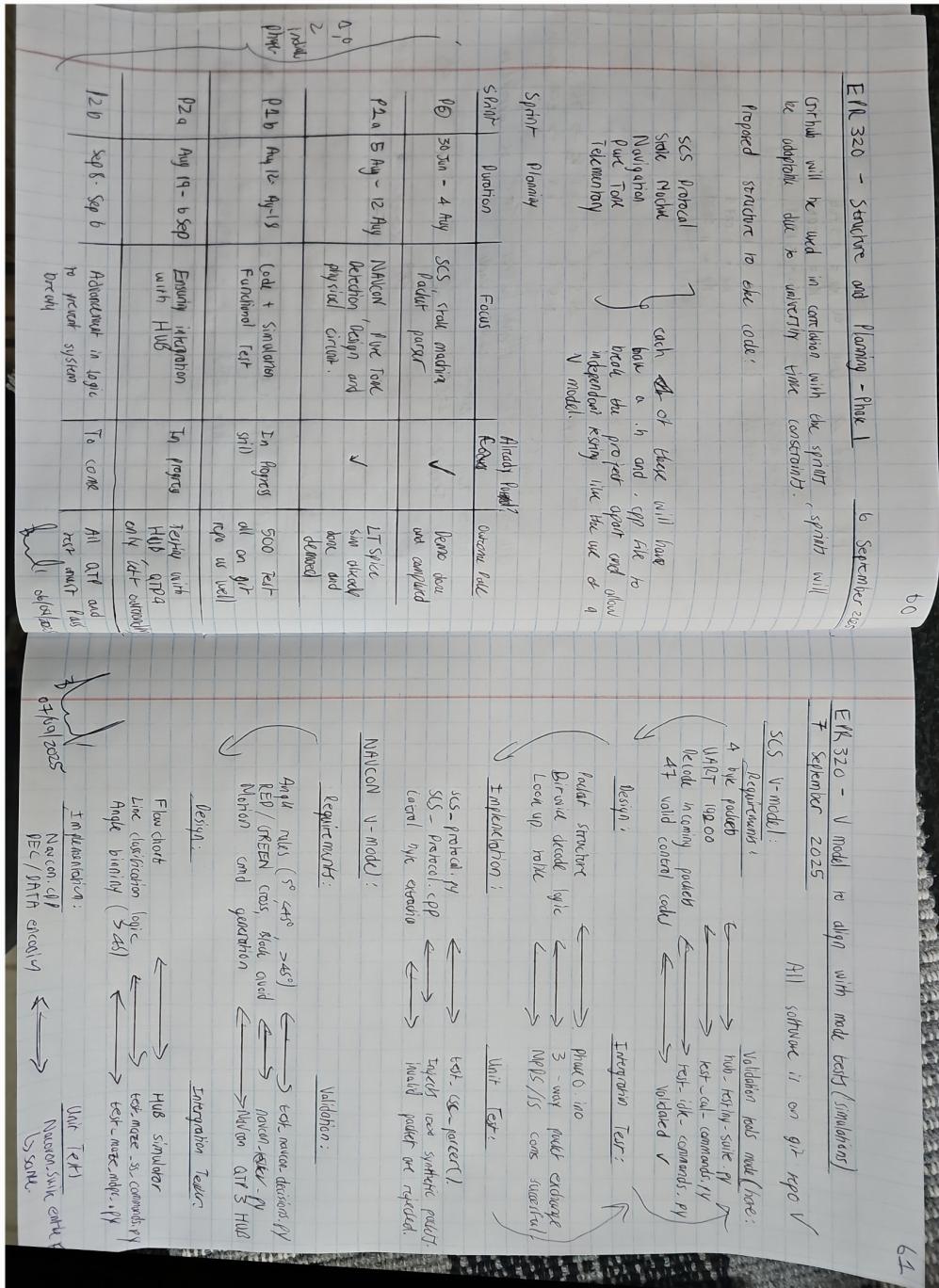


Figure 13: Lab book excerpt: Engineering methods and development process

2.7.5 Statements of requirements

<p><u>ERD 320 - A-Music-Eng NAVU Introduction</u></p> <p><u>23 July 2025</u></p> <p>Aim is to design and construct a Microcontroller-based Autonomous Rover Vehicle (NAVU) capable of navigating through a maze.</p> <p>Microcontroller can be any type and any language.</p> <p>NAVU has 3 sub-systems:</p> <ul style="list-style-type: none"> (1) Sensor (SS) (2) Spk - ad. Navigation Control (SNC) (3) Motor Drive and Power Supply (MDS). <p>Each subsystem must communicate via serial to:</p> <ul style="list-style-type: none"> (1) each other in an integrated system (2) HUB <p><u>HUB</u></p> <p>Rogue based simultaneous evaluation of the 3 systems</p> <p>Allows individual testing and calibration</p> <p><u>Non sub-system focus!</u></p> <p><u>BY</u> My section is the SNC sub-system.</p> <p><u>SNC High level overview</u></p> <p>→ Responsible for state transitioning.</p> <p>→ Display critical system diagnostics from other sub-systems.</p> <p>→ Fulltime Human Machine Interface (HMI) will be displayed.</p> <p>→ Implement navigation control algorithm.</p> <p>Encapsulates: communication, mission, NAVU movement, n. MDS.</p> <p><u>23/07/2025</u></p>	<p><u>ERD 320 - SNC low level Specification</u></p> <p><u>23 July 2025</u></p> <p>1</p> <p><u>SNC Must:</u></p> <ul style="list-style-type: none"> → Quick to respond and not make the communication slow due to processing done → Operate from 3.3 V, 5 V, 9 V and not exceed 0.5 A pull. → Able to SCS (will be discussed) → Detect and communicate all state transitions → able end-of-callibration (EOC) signals from SS and MDS within → CTL state before displaying critical information. → 2nd touch sensor to transition from CAR to MAZE state. → Make decisions needed to navigate the MAZE. (navigation specifications). → detect a point too far at the frequency given. → Receive and display critical system diagnostic information, data → Alert for an error in a local board <p><u>Point To be Determined:</u></p> <ul style="list-style-type: none"> Maximum volume of 60 dB with minimum distance of 10cm. from microphone. Must operate from 3.3V from the ESP32 PIN, and feed back 0.33V. Receiving 2 notes with duration of 0.5 to 1 second, within 2 seconds of each other. Transition from MAZE state into SOS state. ↳ Remain until another 1st or 2 notes are detected. No digital filter are allowed. Must detect capture in a noisy environment. Group 45 — 2500 Hz needs to be detected, with 5% tolerance <p><u>23/06/2025</u></p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figure 14: Lab book excerpt: Requirements specification for SNC subsystem

2.7.6 Development

ERL 320 - Phase 1 - Part Tone Detection Planning 8 August 2025

Want about voltage at 2.8 kHz from the -10 dB SPL

↳ Difficult to get without full auto-shield but can estimate it to be in micro volt

↳ We don't know anything everything to partition

Next → Flow of circuit is build

NIC-E1 PCB → Coupling + bias → pre-amp → band pass filter

Opamp → Envelope detector

Bal Resistor

→ Provide DC voltage to power internal FET

Limit current to safe levels.

$R_B = \frac{V_{DD} - V_{MIC}}{I_{MAX}}$

Using 5V supply where the compn will be the 3232 to ensure to don't

$\therefore R_A = \frac{5 - V_{MIC \text{ output}}}{0.5 \times 10^{-3}}$ → $V_{MIC \text{ output}}$ is not available

↳ Our max is 10 kΩ will need a 10 kΩ resistor as a current 0.5 mA

↳ Range is 0 - 10 kΩ.

31

ERL 320 - Part Tone Detection Planning 8 August 2025

Coupling capacitor:

$f_c = 2\pi \cdot R_{load} \cdot C_{coupl}$

Currently we have:

AC signal from microphone → this needs to be coupled to supply deck 0-5V

$I_{IN} = \frac{S - V_{MIC}}{R_1}$

What I_{MAX} is the max current carrying cap of transducer.

$R_2 = R_2$ to make 0.25 V

Current limit

This ~~variable~~ ~~constant~~ ~~fixed~~ L-type will be used to simulate this design to ensure full operation.

Adding a source of 25 mV to simulate the voltage from the microphone.

Resolution of normal voltage source is 100 mV → makes a just fine for smooth or a specific frequency.

Need to add: options detail = 2.0, $|R|_{total} = 1.0$, $|V|_{total} = 1.0$.

↳ Fixed issue.

New voltage about 2.5V with micro volt as the input.

32

8/08/2025

8/08/2025

Figure 15: Lab book excerpt: Development activities and circuit prototyping

2.7.7 Simulations

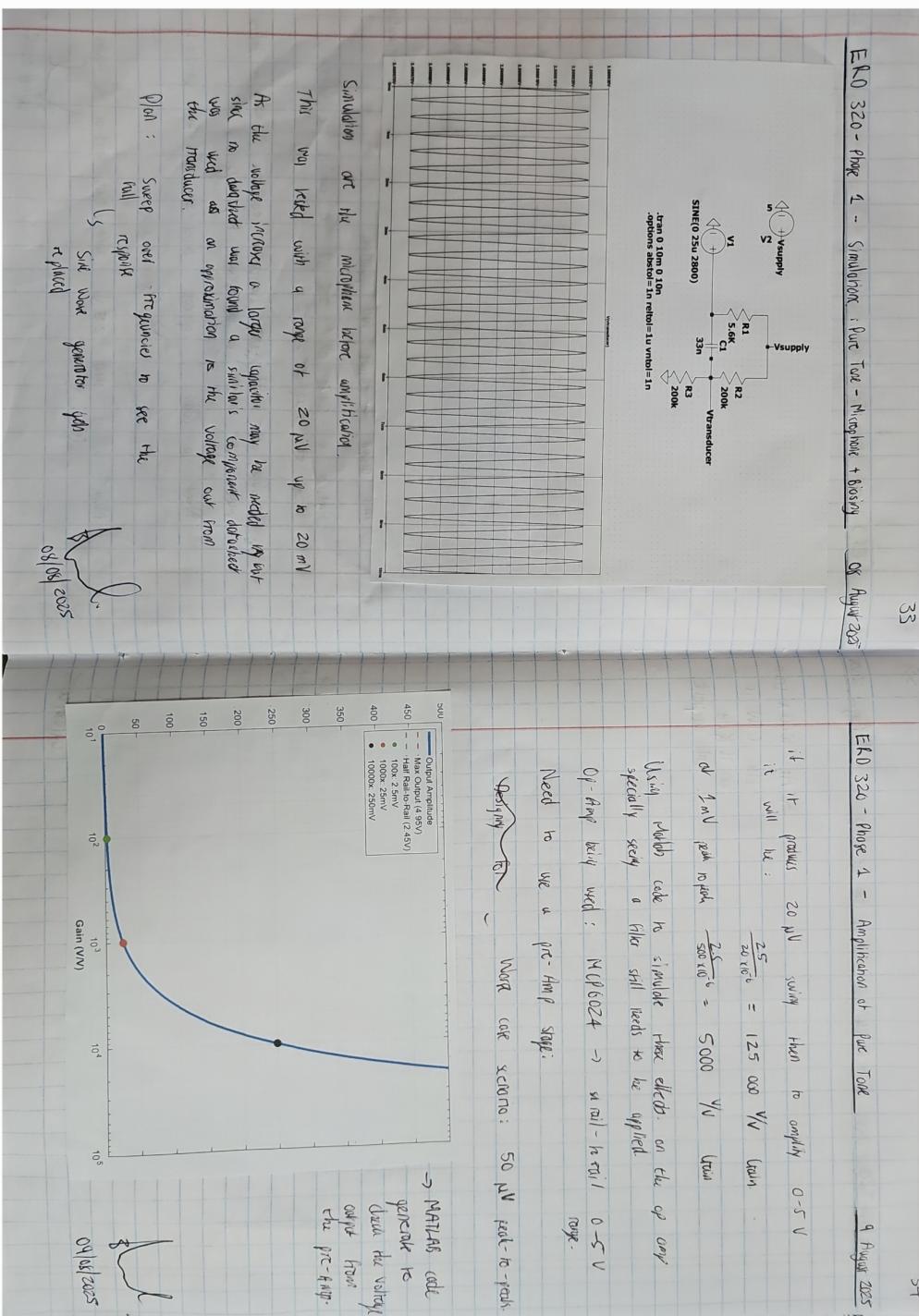


Figure 16: Lab book excerpt: SPICE simulation results and analysis

2.7.8 Approach to coding and testing

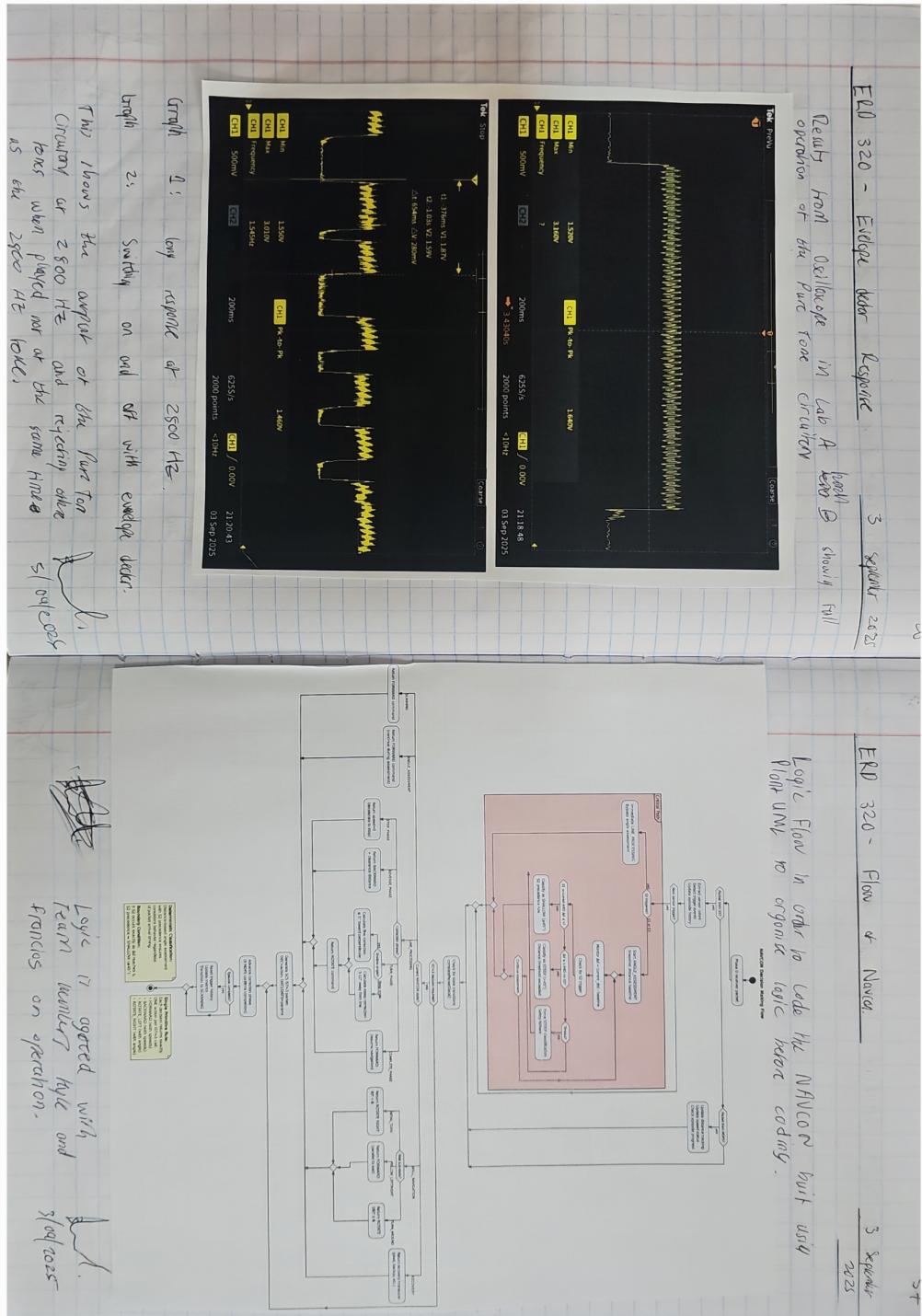


Figure 17: Lab book excerpt: Approach to firmware coding and testing

Instructions for adding lab book excerpts:

- 1) Scan each lab book page at minimum 300 DPI resolution
- 2) Save as high-quality PDF or PNG files
- 3) Name files: labbook_constraints.pdf, labbook_tradeoffs.pdf, etc.
- 4) Place files in 01_SNC/images/ directory
- 5) Replace placeholder boxes above with:
`\includegraphics[width=0.9\textwidth]
{01_SNC/images/labbook_constraints.pdf}`
- 6) Ensure lab book entries are dated, legible, and professionally presented
- 7) Each excerpt should occupy approximately one full page in the final report

3 Subsystem 2

3.1 Subsystem Needs Analysis

A crucial element of the autonomous vehicle known as MARV is the SS(sensor subsystem). In this context, one that is capable of navigating a maze for which the only guides to the end are coloured lines, representing different obstacles on a PVC mat.

The MARV will need to identify the obstacles by interpreting the colour of the lines on the maze. This means that SS will need some form of colour detection.

To prevent the disturbance of changing ambient light, SS will need to minimise its influence when performing colour detection.

The MARV will need to keep track of the MARVs orientation inside the maze. This means that SS will need to calculate the angle of incidence when crossing the lines.

SS will need to integrate with the full system with minimal change in functionality and operation.

For SS to integrate with the full system, SS will need to use the communication protocol provided, which will enable seamless communication with other subsystems.

Though SS is a subsystem, it needs to work independently as well for testing during development. This means that SS will need its own power supply.

The afore mentioned needs for SS are shown in a context diagram to better illustrate the subsystem's place in the system as a whole.

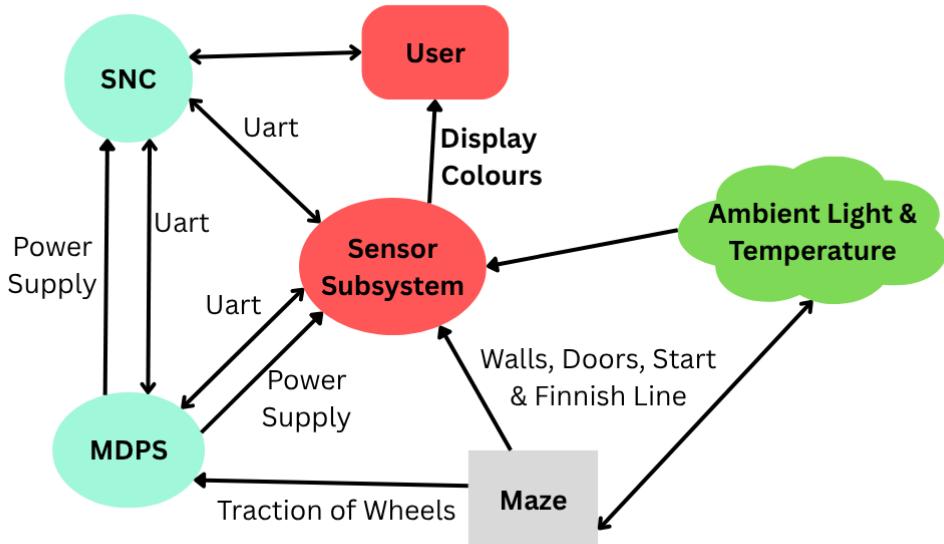


Figure 18: Context Diagram For The SS(Sensor Subsystem)

3.2 Subsystem Concept Exploration

- Refer to Kossiakoff, Chapter 6, 7 and 8
- Do a brief survey of literature on possible methods / circuits / designs that could meet the above needs, other than the prescribed subsystem. **MAKE SURE THAT THE DOCUMENTS YOU REFERENCE ARE INCLUDED IN SECTION 1.2!!!**

3.2.1 Colour Detection Sensors

3.2.2 Incidence Angle Detection

3.3 Subsystem Concept Definition: Planning

- Refer to Kossiakoff, Chapter 9
- Provide a single-figure subsystem functional diagram
 - o Extend on the practical guide diagram with more details, unique to your design.
 - o Include direction and nature of all internal and external interactions, including those with other subsystems and those with the outside world.
- Provide a single-figure subsystem architecture diagram
 - o Extend on the practical guide diagram with more details, unique to your design.
 - o Draw to component / software function level
 - o Include direction and nature of all internal and external interactions, both to the external world and to other subsystems
 - o Indicate clearly which components are designed and which components are bought off-the-shelf.
- **This must be more detailed than the diagrams in the practical guide; copies of the practical guide figures will receive 0 marks.**

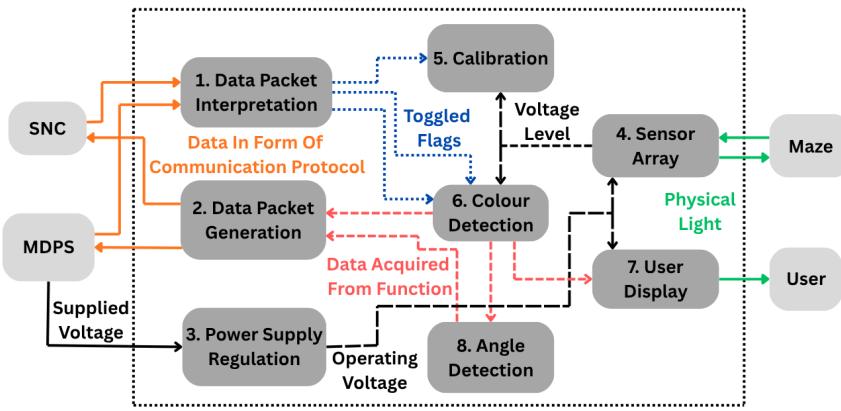


Figure 19: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

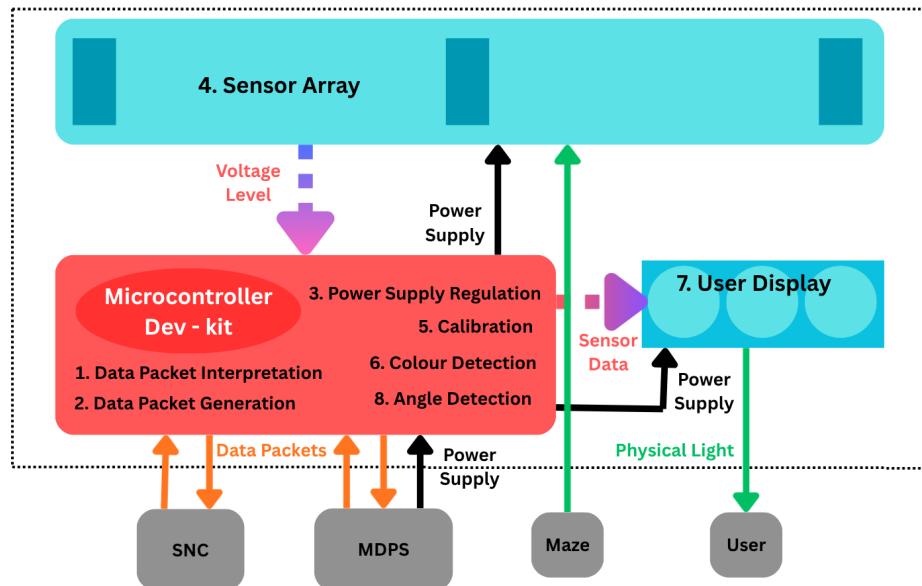


Figure 20: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

3.4 Subsystem Engineering Design

3.4.1 Constraints and trade-offs

3.4.1.1 Design Constraints

Power supply is confined to 0V to 5V with 1V margin.

Shape, the subsystem needs to be a rectangle with a much larger length than width to ensure integration at the front of the Marv as the sensor subsystem.

Size, the length must be roughly equal to the wheel base of the Marv, and the width must be short enough for the subsystem to be mounted at the front while keeping the Marv small enough to turn inside the maze with ample room.

3.4.1.2 Trade-offs

The speed of photodiodes vs the reduced complexity of the phototransistors. The phototransistor is its own amplifier, but has an inherently high output impedance, which reduces the reliable sample time of an ADC, while the photodiodes can be made to sample faster since the amplifier can be changed with much added complexity. Since size is a constraint, the phototransistor is chosen to reduce complexity in the physical circuit.

3.4.2 Tools and Methods

3.4.2.1 Engineering tools

Oscilloscope, used to monitor the reliability of the power supply, mapping out components for which no reliable datasheets could be found, and testing input/output signals during the developmental process.

LtSpice, a simulation tool used to simulate complex circuits to test calculated values for components and confirm the function of designed circuits before physical testing is done.

Python script, used to test soft functions like calculations in a code-friendly environment before implementing said functions in an embedded environment.

SolidEdge, a CAD software to design physical hardware to be 3D printed for the subsystem.

3.4.2.2 Engineering methods

An agile prototyping approach is followed for any CAD designs for 3D printing. The material is cheap and there is no labour involved, meaning that multiple iterations can be implemented and tested in a short time at low developmental cost.

A waterfall approach is followed for any circuitry, such as the real-time display, since the design and implementation of circuitry can be more expensive physically and in terms of labour. A more systematic approach is required. Requirements for the circuit are considered when using relevant literature like textbooks and datasheets, to design an initial concept. After this, a detailed design is achieved through simulations like LTspice to confirm the function. Implementation follows on a breadboard and is tested before being implemented as a prototype and tested again.

A more agile approach is followed for soft functions, like angle calculation, since this is the cheapest part of development in terms of physical cost. After an initial concept is established, multiple iterations of simulation and testing in the Python script environment.

3.4.3 Selected design details

Sensor casing. This will be the only 3D printed element presented. It is necessary to protect the sensor array from the disturbance caused by a change in ambient light, as well as to maintain constant orientation of radiating LEDs and opto-coupled phototransistors to minimize disturbances.

Circuit elements included will be the following:

1. LED Array for the sensor array. The LED Array will be 3 RGB(red, green, and blue) LEDs that flash light down on the maze to be reflected.
2. The phototransistor array. This will be opto-coupled with the LED array to record the reflected light from the maze to identify the colour currently being sensed.
3. Real-time display. This will be implemented with an array of 3 RGB LEDs to interface the colour currently being detected by each sensor to the user.

Software elements included will be the following:

1. Data packet interpretation/generation. This is necessary to integrate with the communication protocol provided in the guide.
2. Calibration. This is a necessary algorithm to provide a dynamic and at all times accurate set of data to test and identify the light being reflected back from the maze.
3. Colour detection. This algorithm will use afore afore-mentioned acquired calibration data to identify the colour currently being seen on the maze to relay usable data to other functions like angle detection and data packet generation.
4. Incident angle detection. This algorithm is necessary to calculate the angle of incidence of the Marv to compensate for small errors that can disorient the Marv.

3.4.3.1 Statements of requirements

Briefly state the high-level requirements of your individually designed circuits and / or code segments.

3.4.3.2 Development

- Demonstrate top-down definition with bottom-up implementation (or some other approach commensurate with your selected design approach) of your selected detail designs, motivated by flow charts, design equations, or some other guiding input.

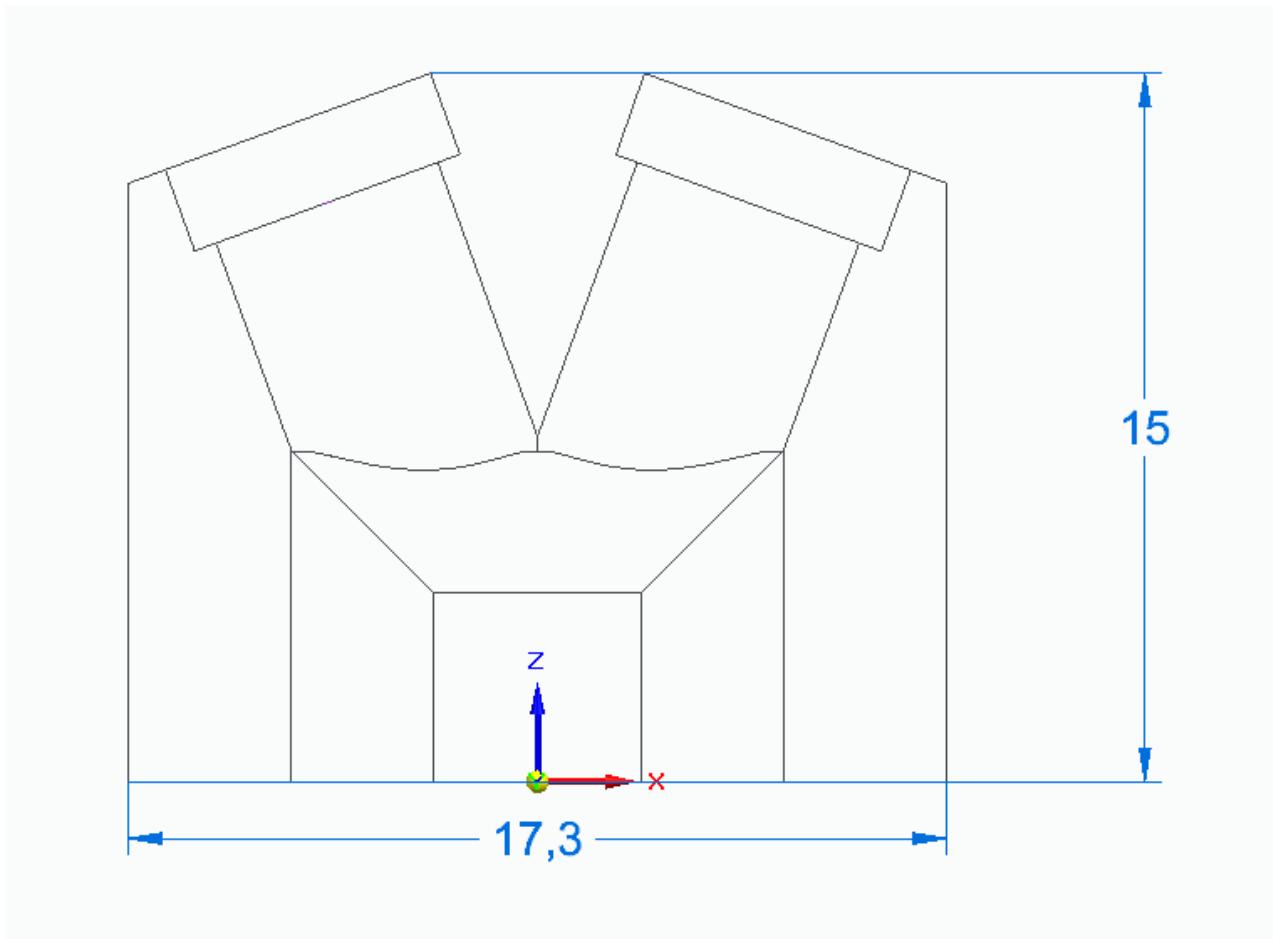


Figure 21: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

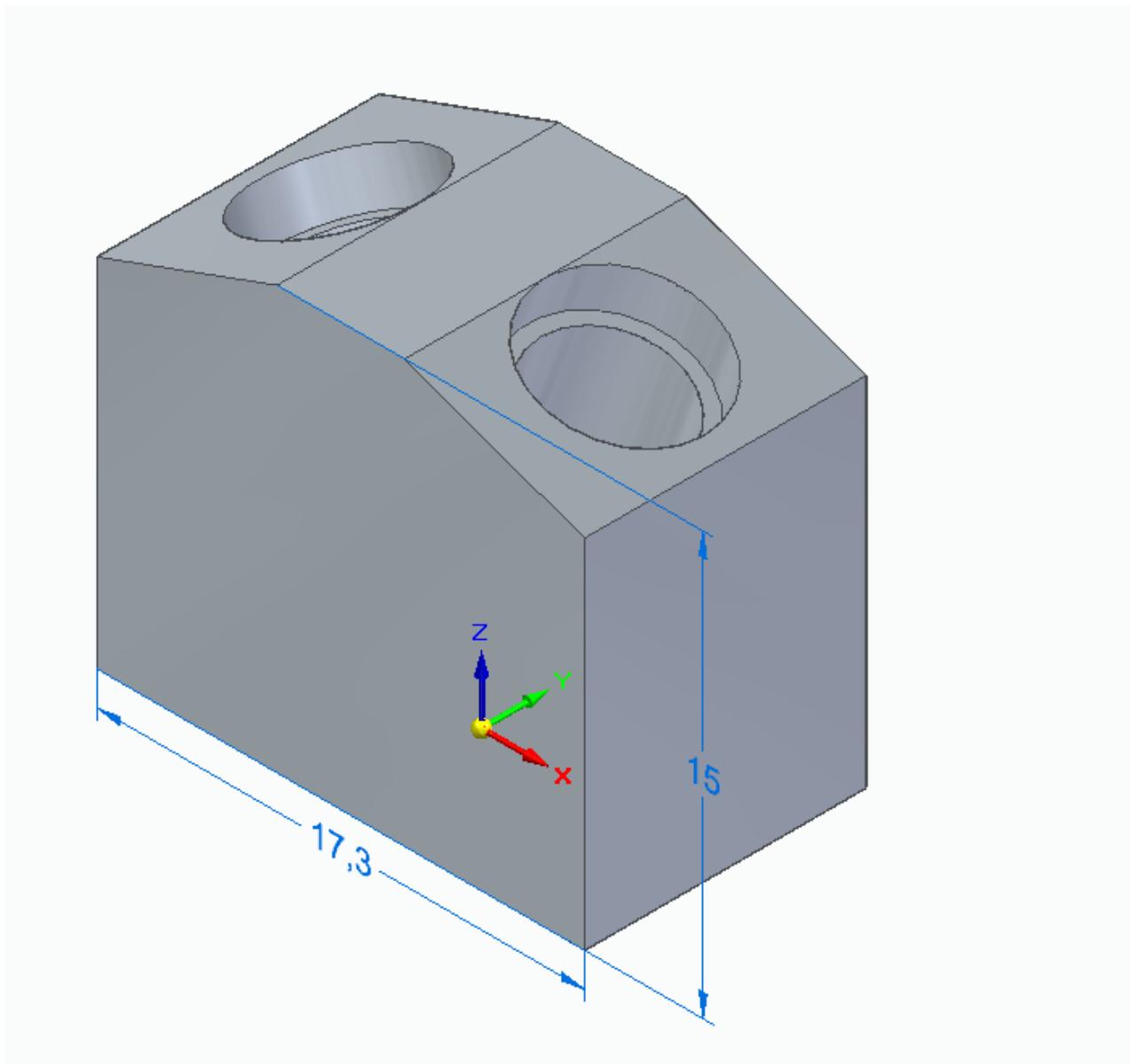


Figure 22: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

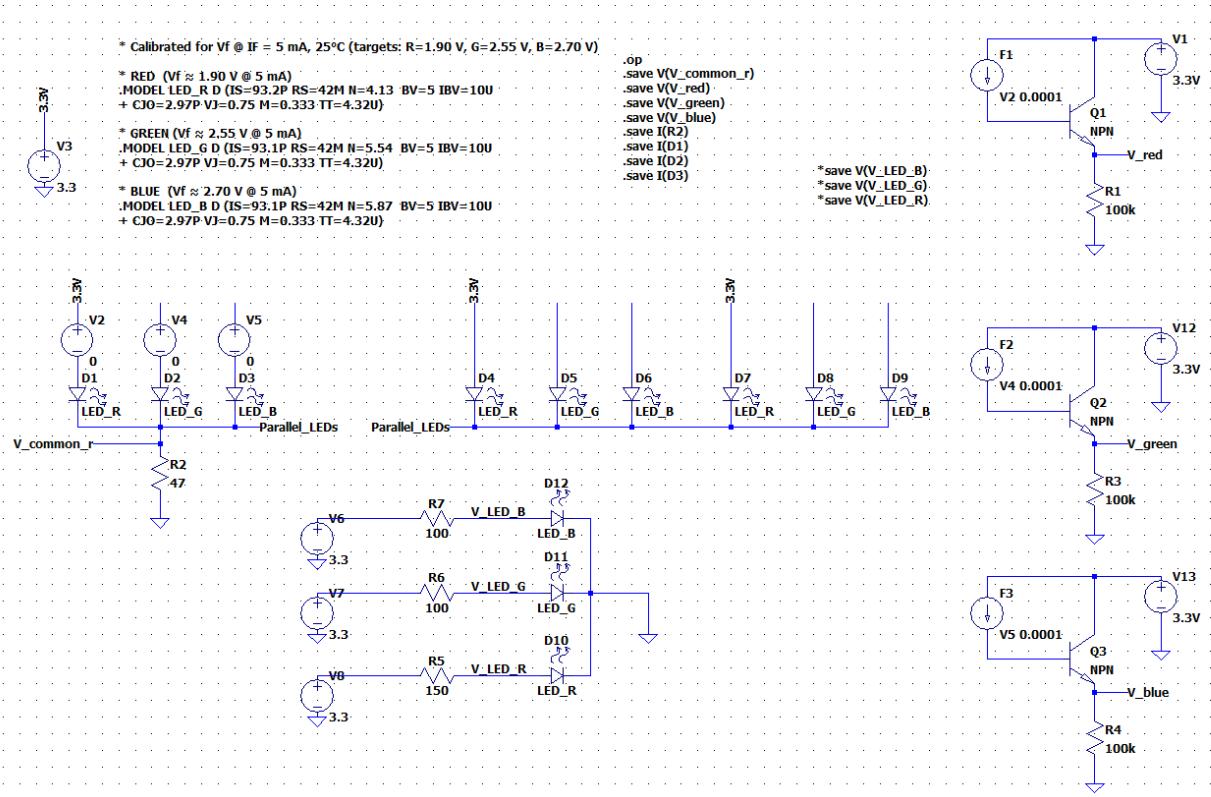


Figure 23: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

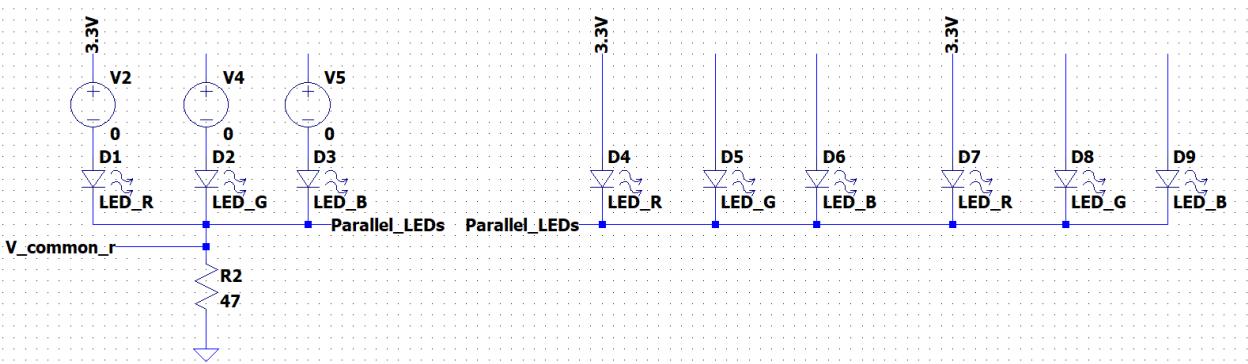


Figure 24: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

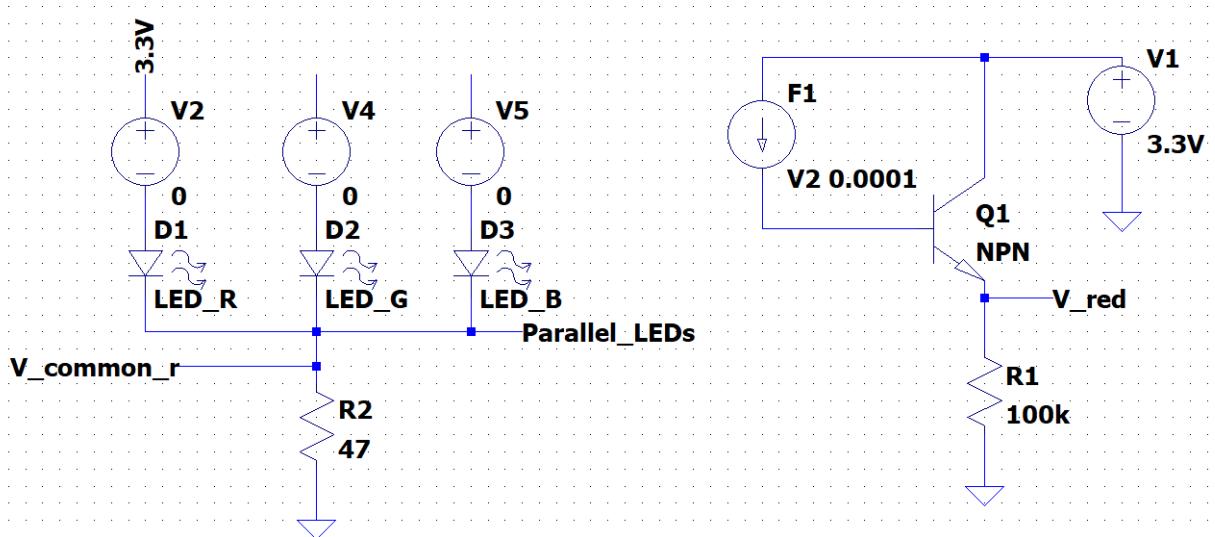


Figure 25: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

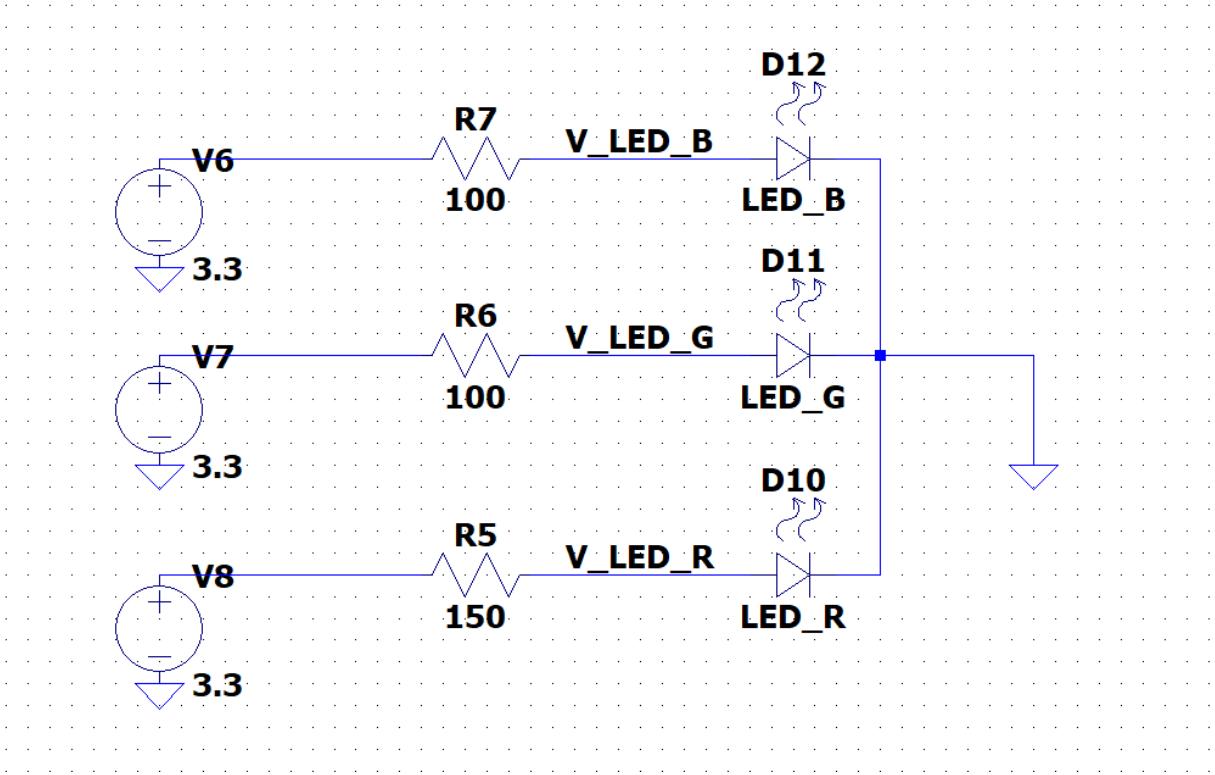


Figure 26: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

3.4.3.3 Simulations

- Where appropriate, demonstrate circuit simulations, test bench operation, or some other form of computer verification of operation prior to deployment.

3.4.3.4 Approach to coding and testing

- Describe your approach to microcontroller coding and code testing, with selected examples.

3.5 Subsystem Qualification Tests Results

- *Include ONLY the following details of the qualification tests in table format:*
 - o Specification to be verified (as per the MARV practical guide)*
 - o Expected Results, based on your design efforts and simulations. Be specific!*
 - o Provide the measured results of your tests*

3.6 Subsystem Conclusions and Recommendations

- *Does the subsystem adhere to the requirements set out? If not, why not?*
- *What are the benefits and shortcomings of the subsystem?*
- *What future work do you recommend for the subsystem?*

The subsystem presented fully adheres to the requirements set out

3.7 Lab Book Excerpts

3.7.1 Design Constraints

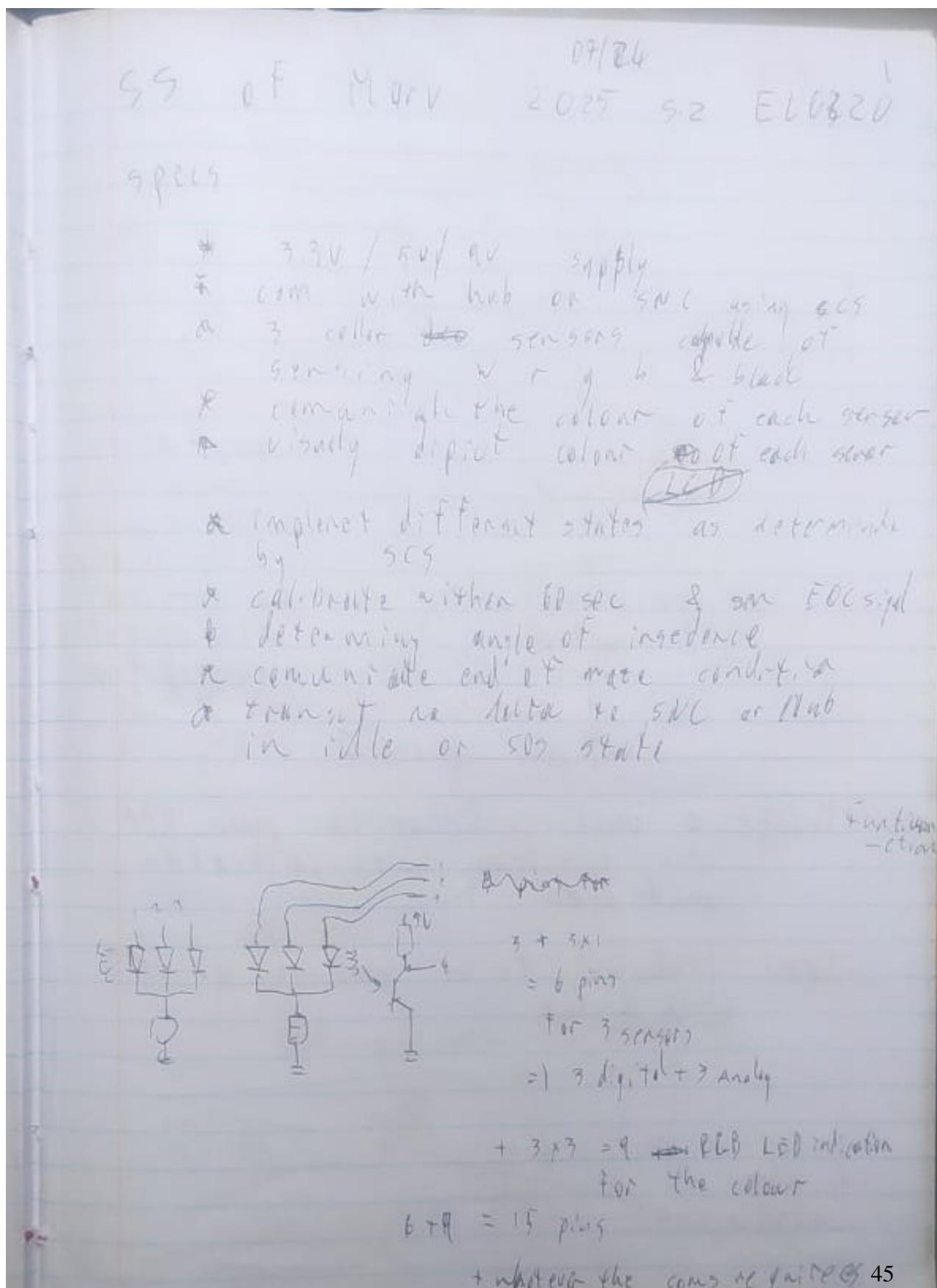


Figure 27: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

3.7.2 Trade-offs

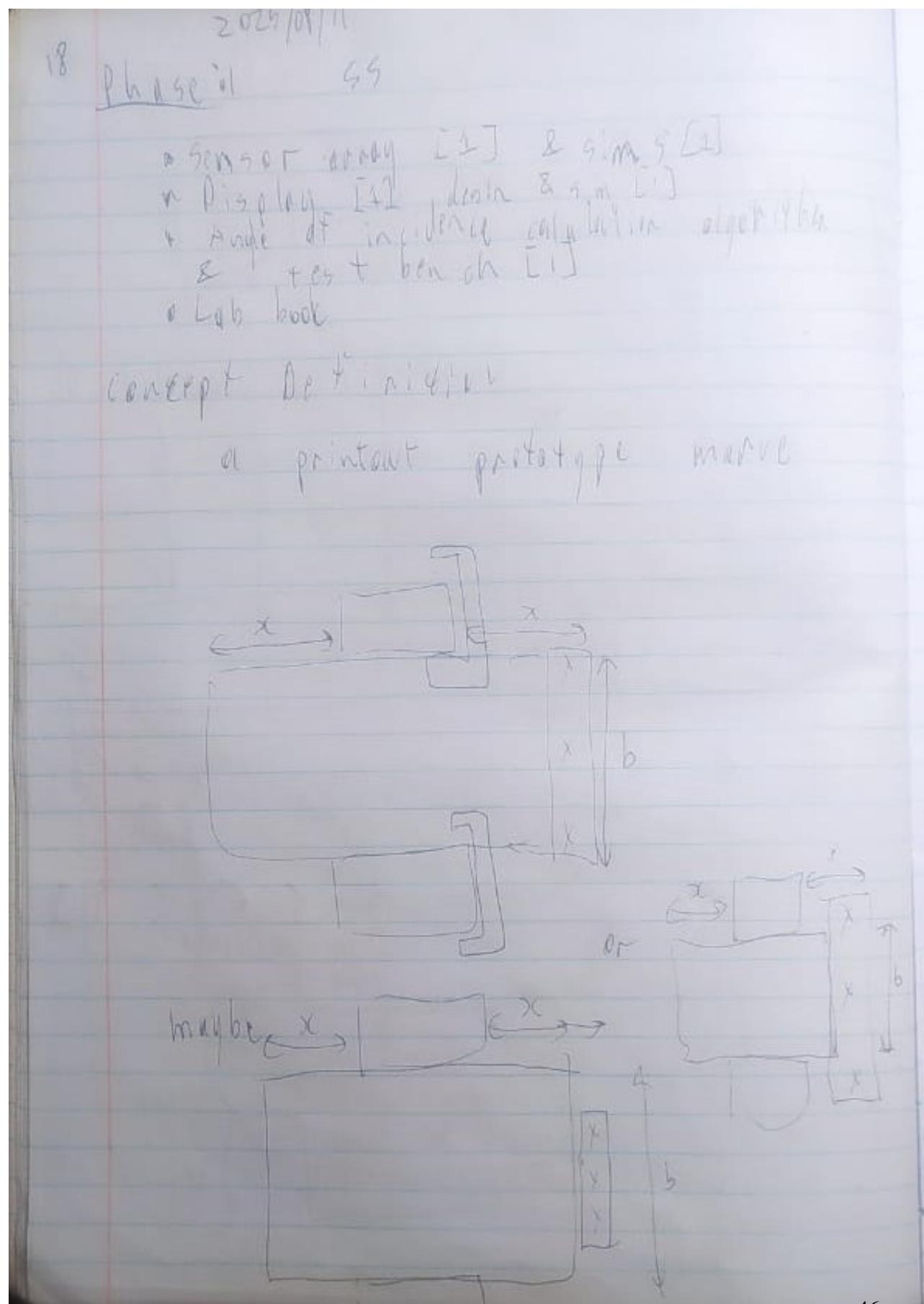


Figure 28: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

3.7.3 Engineering tools

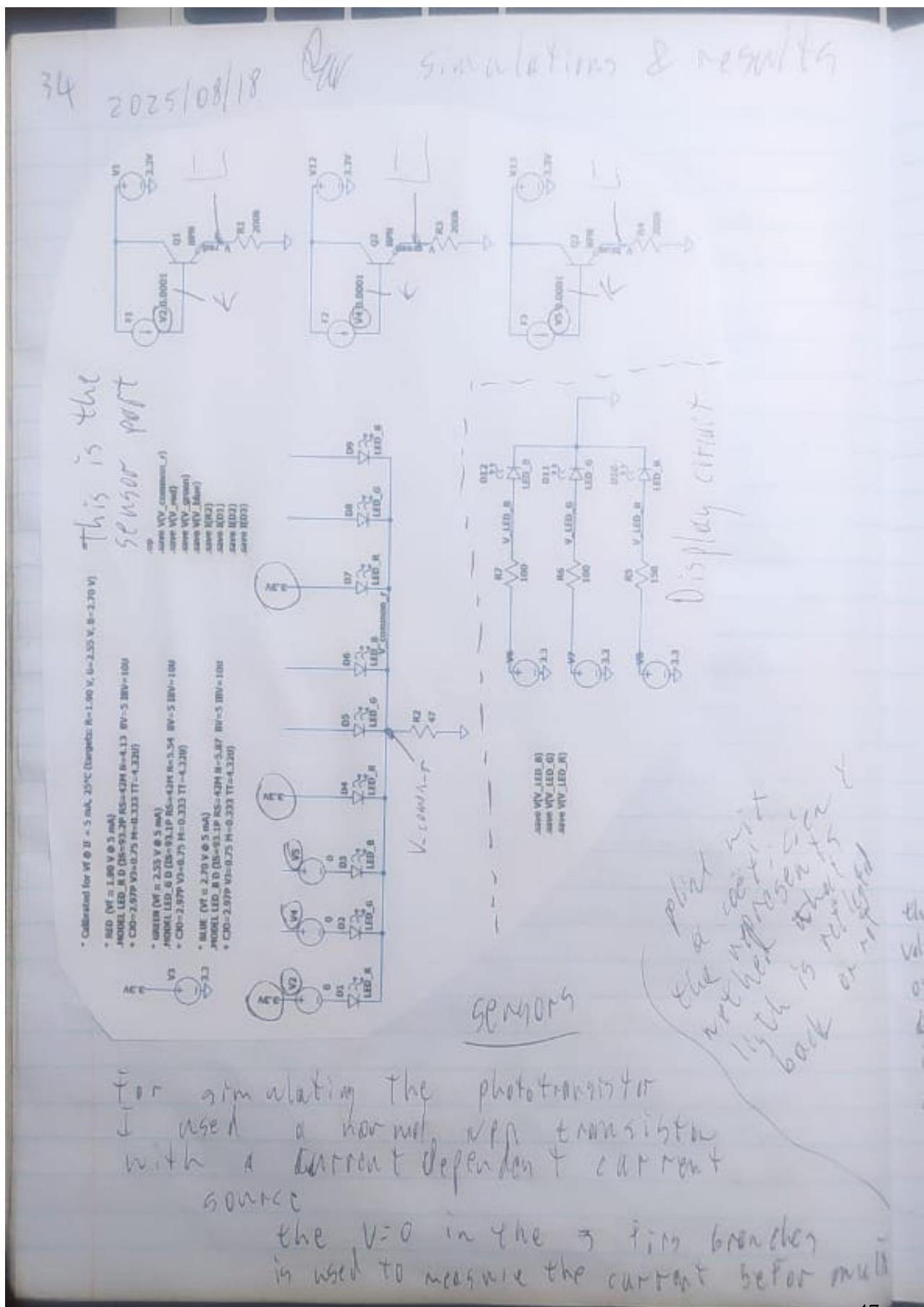


Figure 29: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

3.7.4 Engineering methods

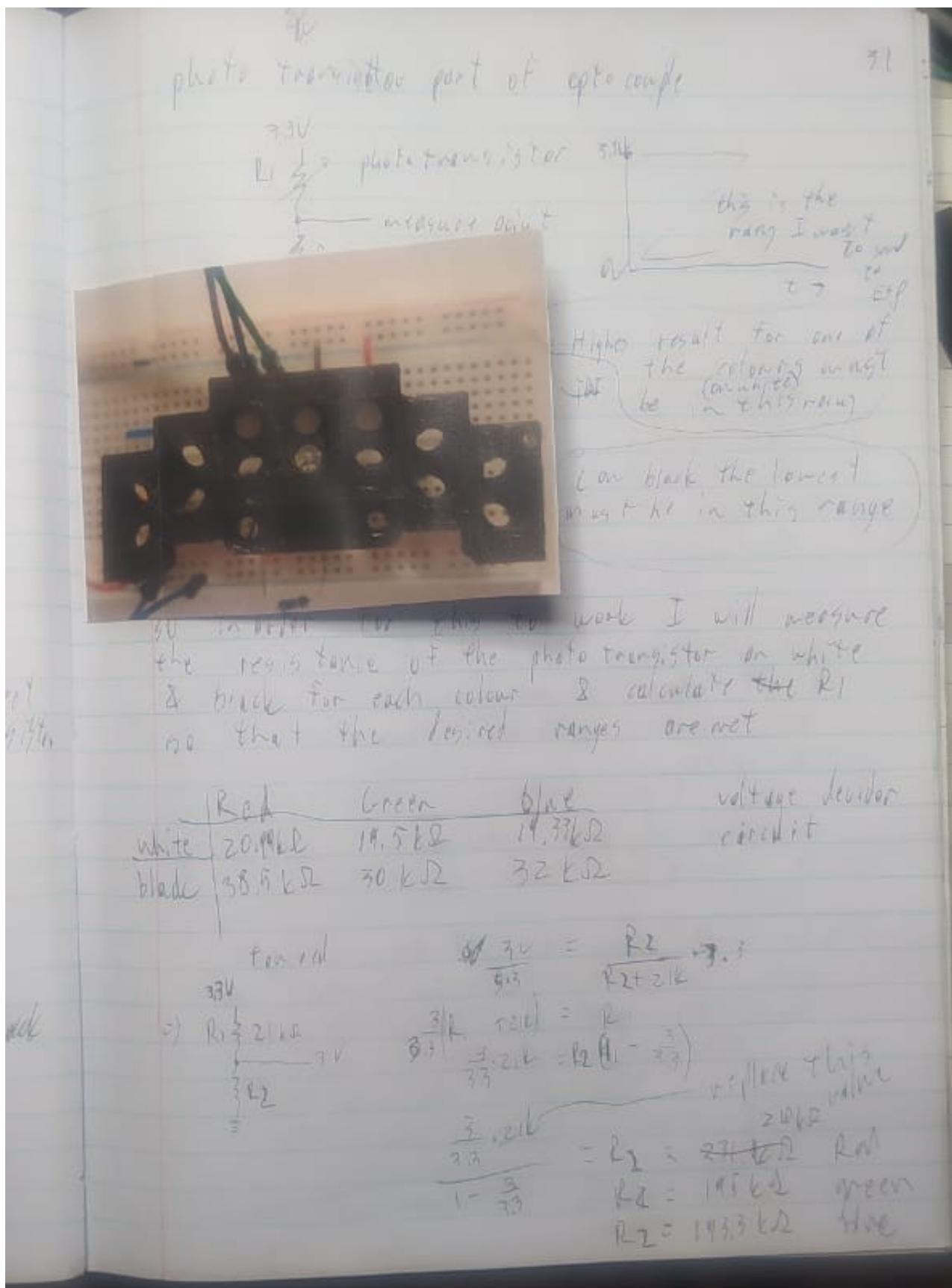


Figure 30: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

3.7.5 Statements of requirements

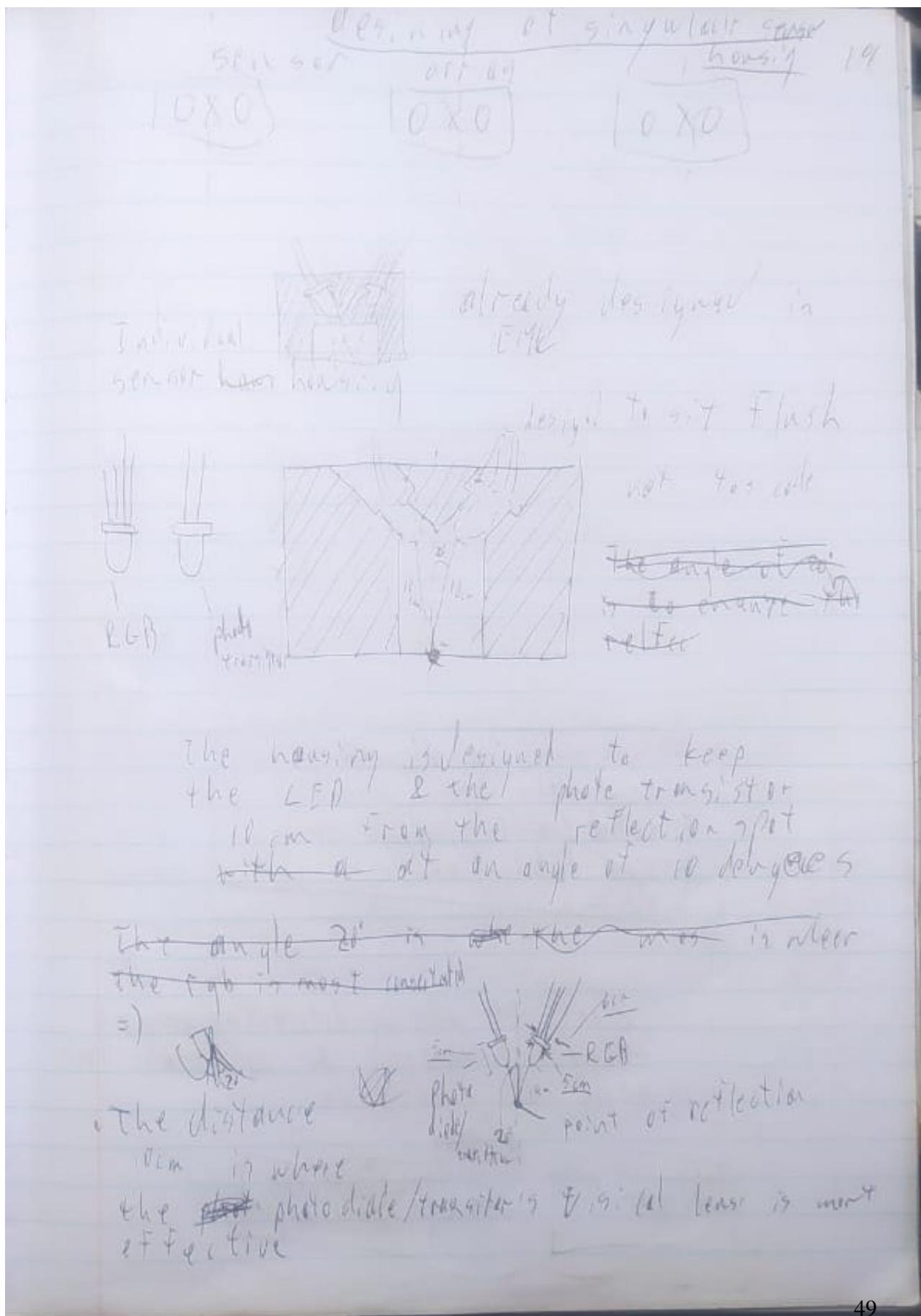


Figure 31: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

3.7.6 Development

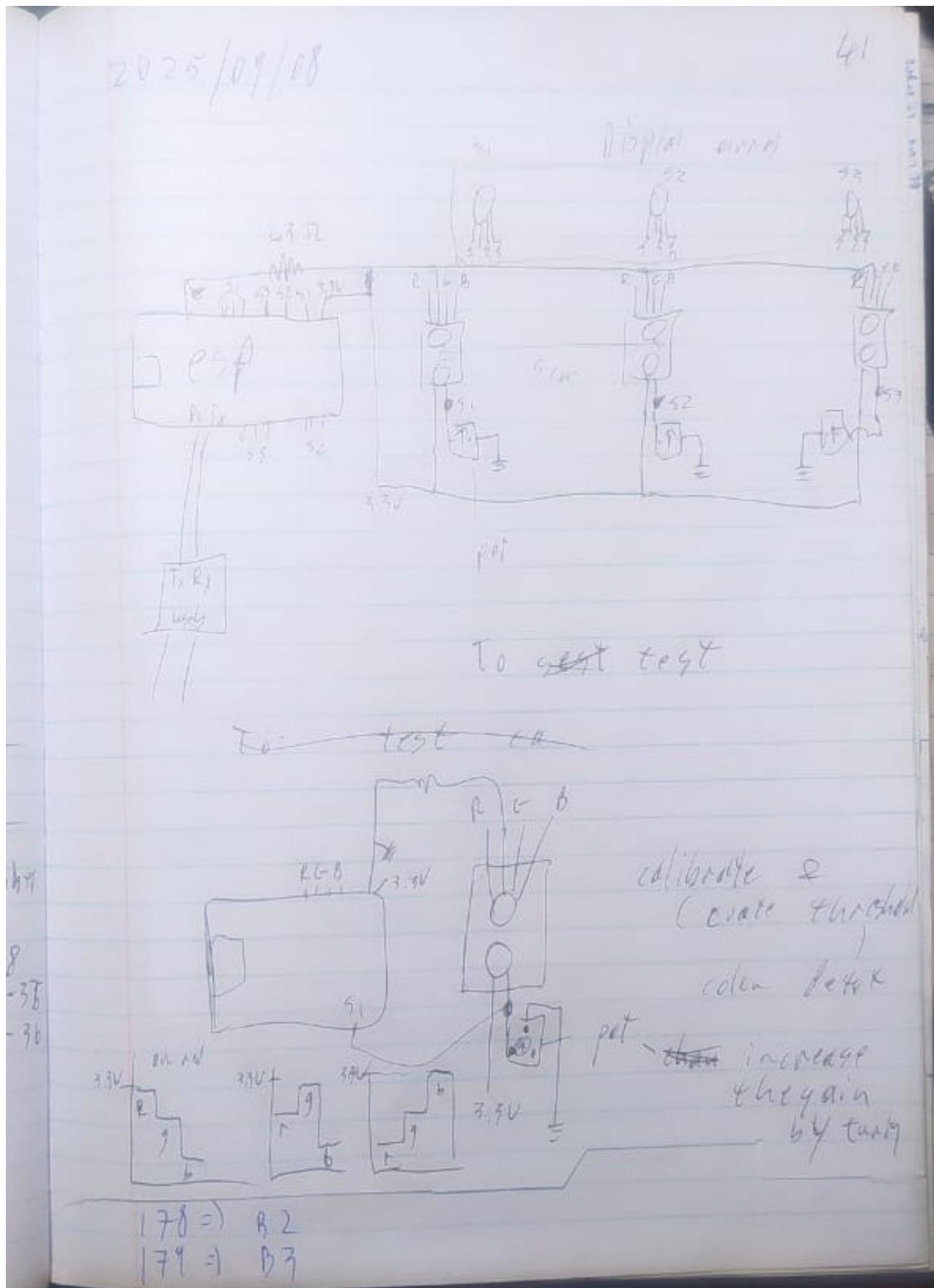


Figure 32: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation 50

3.7.7 Simulations

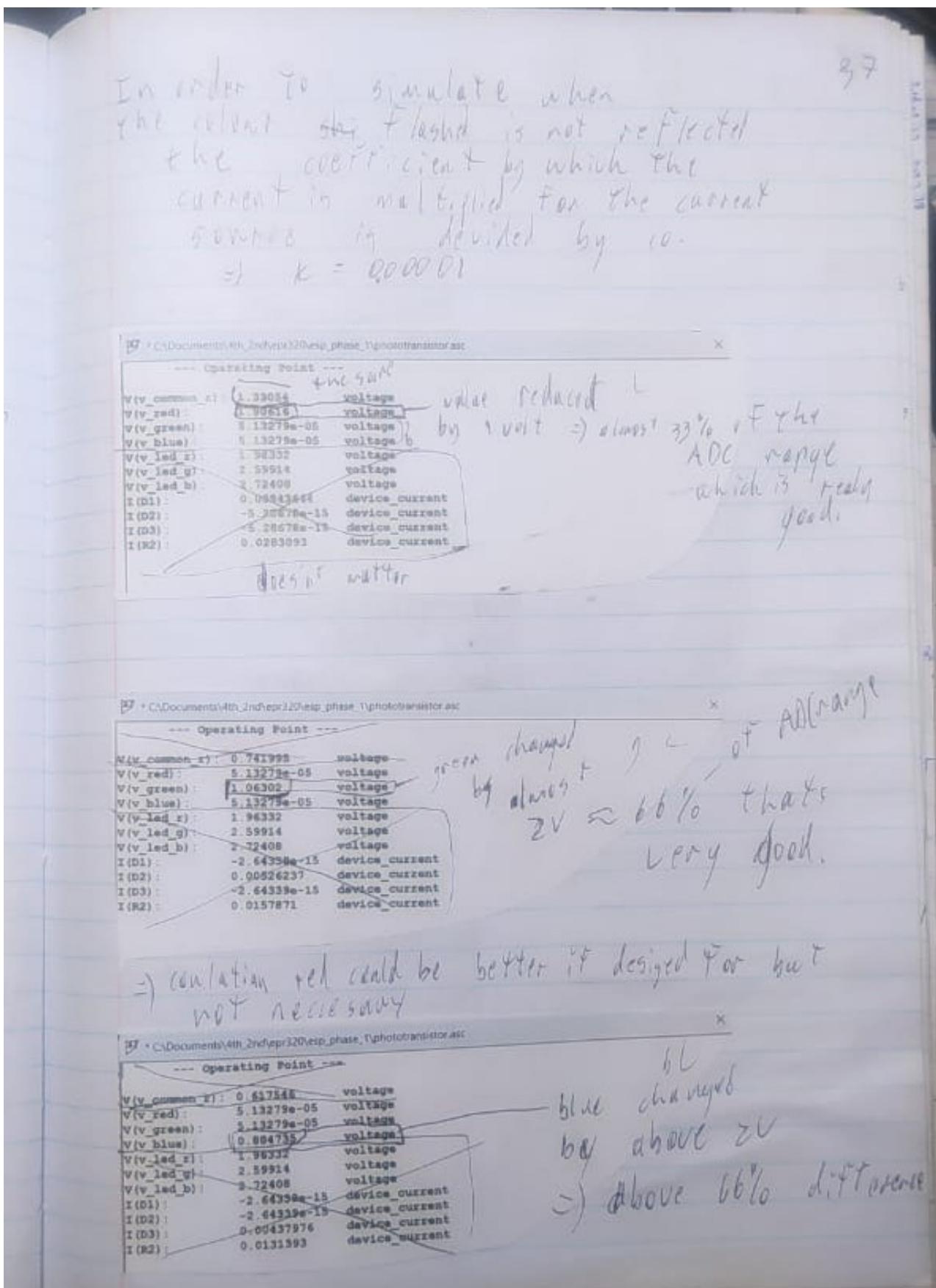


Figure 33: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

3.7.8 Approach to coding and testing

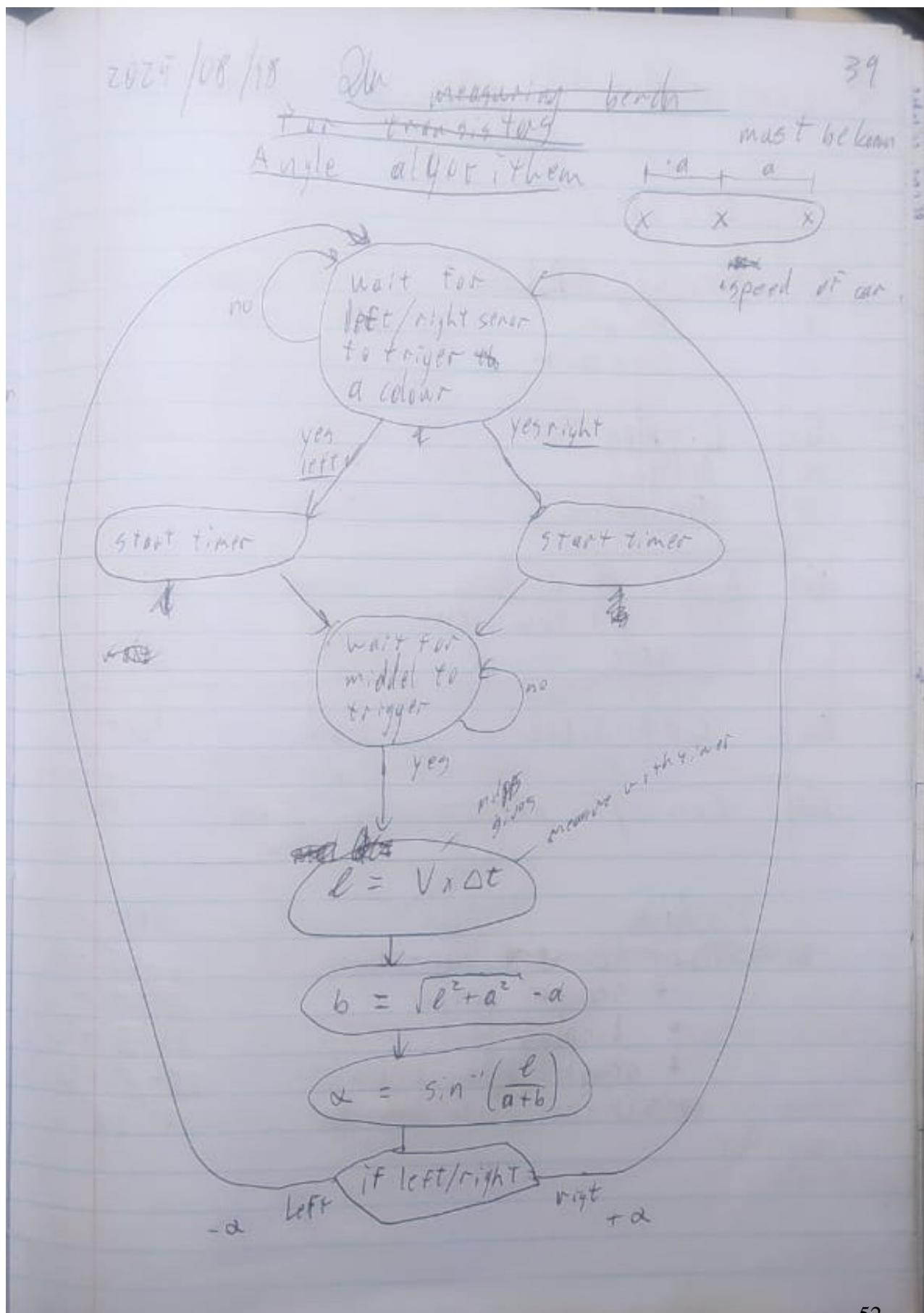


Figure 34: Pure Tone Detection Flow: Analog Signal Chain and Digital Validation

4 Subsystem 3: MDPS

4.1 Subsystem Needs Analysis

The design and construction of a Microcontroller-based Autonomous Robotic Vehicle (MARV) capable of navigating through a maze using predetermined navigation rules requires the development of multiple interacting subsystems. These subsystems must communicate effectively to function as a cohesive unit and achieve the overall task. In particular, a Sensor Subsystem (SS) is required to detect and measure the MARV's surrounding environment, a State-and-Navigation Control (SNC) subsystem is needed to interpret sensor data and execute decision-making protocols, and a Motor Driver and Power Supply (MDPS) subsystem must supply power to all components and control the MARV's movement. Together, these three subsystems fulfil the essential functions of detection, processing, and actuation.

Focusing on the MDPS subsystem, there is a need to control the MARV's speed and direction based on commands received from the State-and-Navigation Control (SNC) subsystem, and it must measure the MARV's speed, distance travelled, and rotational displacement. The MDPS must also supply power to all subsystems to enable operation independent of external power sources. In addition, the MDPS must be able to receive and transmit information to and from the SNC and SS.

4.2 Subsystem Concept Exploration

Various approaches can be considered to address the needs of the Motor Driver and Power Supply (MDPS) subsystem. By reviewing multiple sources and evaluating the advantages and disadvantages of each option, a more informed decision can be made regarding the concept best suited for the intended application.

4.2.1 Movement

Standard wheels, continuous tracks, cams, or propellers are some of the many options that can be considered when exploring different modes of movement for the MARV.

TABLE VI: Comparison of Different Motion Systems

System	Advantages	Disadvantages
Wheels	Simple Design, Low Cost, Manoeuvrability	Dependence on Flat Surfaces
Tracks	Improved Traction, Better Weight Distribution	Complex Design, Lower Speed
Cams	Simple Design, Precision Motion	Friction Losses, High Torque Required
Propellers	Independence of Ground Terrain	Complex Design, Manoeuvrability

Table VI shows that wheeled systems offer a simple design, low production cost, and good manoeuvrability, but they perform best on flat surfaces and may struggle on rough or uneven

terrain. Tracked systems provide improved traction and better weight distribution, making them suitable for uneven and smooth surfaces [?], but they are mechanically more complex and generally slower. Cam-based mechanisms is just a lever arm which spins off-centre from the shaft, which can achieve precise motion and maintain a simple design [?], but they suffer from friction losses and require high torque if lifting a system of the ground. Propeller-based systems allow the vehicle to move independently of ground terrain, which is advantageous for unconventional surfaces [?], but they introduce higher complexity and manoeuvrability is significantly reduce when confined to small areas such as the maize. Considering the MARV's operational environment being a maze with reasonably flat surfaces and the desire for low-cost, with easily controllable motion, a wheeled system is the most appropriate choice. It provides the necessary balance of simplicity and precision that is required to accurately navigate the maze.

4.2.2 Power

A simple voltage regulator or dc-dc (buck-boost) converter can be considered to regulate the power of the system.

TABLE VII: Comparison of Power Systems

System	Advantages	Disadvantages
Voltage Regulator	Simple Design, Low Cost, Few Components	Large Losses, Only Step-Down Operations Possible
Buck Converter	Wide Input Voltage, Stable Outputs, Smaller Losses	Cost, More Complex

From Table VII, voltage regulators offer a simple design with low cost and require relatively few components, making them easy to integrate. However, they are limited to step-down operation and can exhibit larger power losses compared to switching converters [?]. Buck converters, provide stable outputs over a wide input voltage range and are generally more efficient, but they are more complex and costly, requiring additional components and careful design considerations [?]. Considering simplicity, low cost, and ease of integration as the driving factors, the voltage regulator is the most suitable choice for the power subsystem, as it meets the operational needs of supplying stable voltage to the subsystems without introducing unnecessary complexity.

4.2.3 Communication

I^2C , SPI, and UART are widely used communication protocols in embedded systems due to their simplicity and reliability, and were therefore considered as potential options for the MARV subsystem.

TABLE VIII: Comparison of Communication Protocols

Protocol	Advantages	Disadvantages
I^2C	Simple two-wire interface, Multiple slaves and multi-master, error checking	Requires addressing, Half-duplex
SPI	Full-duplex, Single-master systems	Separate slave select lines for each device, Complex wiring for multiple slaves, Limited multi-master support
UART	Simple two-wire interface, bidirectional, asynchronous, configurable baud rate, widely supported for device-to-device communication	Requires matching baud rates between devices

From Table VIII, I^2C provides a simple two-wire interface and supports multiple slave devices with built-in error checking, but complexity increases with addressing. SPI offers full-duplex capability, making it ideal for high-speed data transfers, but adding multiple devices increases wiring complexity and it does not easily support multi-master configurations. UART is a simple, asynchronous two-wire protocol that supports bidirectional communication, making it a reliable device-to-device communication protocol [?]. Given the MARV’s requirement for asynchronous communication between microcontroller subsystems, and the limited number of devices involved, UART is the most appropriate choice, due to its ease of implementation to meet the operational needs of the system.

4.3 Subsystem Concept Definition: Planning

The Motor Driver and Power Supply (MDPS) subsystem provides both the mechanical actuation and electrical power required for the MARV to operate. Functionally, the MDPS accepts motion commands from the SNC and converts them into controlled wheel movement using its motor actuation hardware. Through this, the subsystem enables the MARV to accelerate, decelerate and perform rotational movements required to navigate the maze. In addition to motion control, the MDPS is responsible for supplying suitable regulated voltage to all connected subsystems. This ensures that the system will operate reliably under varying load conditions. The chosen power approach uses a simple active voltage regulation strategy, which aligns with the MARV’s requirement for stable, low-noise supply rails without unnecessary circuit complexity. The MDPS also incorporates on-board measurement capabilities to support system-level navigation. Using wheel-mounted rotary encoders, the subsystem determines the distance travelled and the angular rotation achieved during turns. These measurements are transmitted to the SNC via asynchronous UART communication.

Conceptually, the MDPS therefore acts as both the “movement unit” and the “power distribution unit” of the MARV. It executes movement on request, regulates the system’s electrical supply, and provides movement data to the SNC. Within the broader system architecture,

the MDPS serves as the interface between the SNC's navigation decisions and the MARV's physical behaviour in the maze environment.

The overall operation of the MDPS subsystem can be seen with the functional block diagram shown in Figure 35 below.

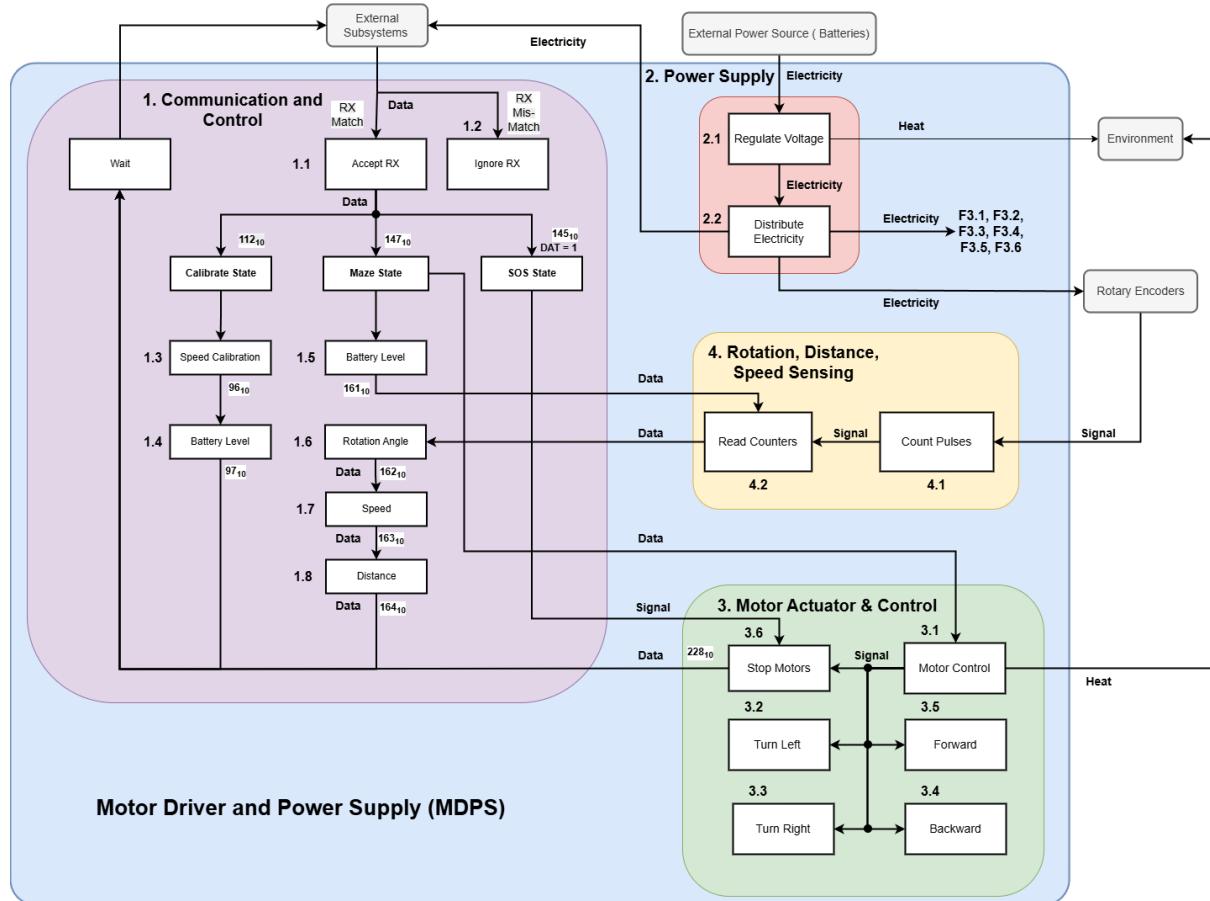


Figure 35: MDPS Functional Block Diagram

The overall architecture of the MDPS subsystem can be seen with the architectural diagram shown in Figure ?? below.

4.4 Subsystem Engineering Design

- Refer to Kossiakoff, Chapter 13, 15 and 16
- In Section 2.8, attach a one-page excerpt from your handwritten lab book as an example for each heading in this section (eight pages in total)

4.4.1 Constraints and trade-offs

4.4.1.1 Design Constraints

- Where were you constrained in implementation of the concept?

4.4.1.2 Trade-offs

- Which trade-offs did you consider? How did you make your eventual selections, and why?

4.4.2 Tools and Methods

4.4.2.1 Engineering tools

- Discuss the engineering tools that you used (hardware tools, simulation packages, etc) and motivate their use. This does NOT mean physical hand tools like soldering irons, pliers, etc!

4.4.2.2 Engineering methods

- Discuss the engineering methods that you used (design methods, prototyping approach, simulation, test benches, etc) and motivate their use. Did you apply agile, waterfall, V-shaped development? If you used parameter sweeps and CAD-based design methods, describe them here.

4.4.3 Selected design details

- Provide selected details of your design, including circuit schematics, simulation results. Start out by listing which design elements (circuits, code snippets) you will present.

4.4.3.1 Statements of requirements

Briefly state the high-level requirements of your individually designed circuits and / or code segments.

4.4.3.2 Development

- Demonstrate top-down definition with bottom-up implementation (or some other approach commensurate with your selected design approach) of your selected detail designs, motivated by flow charts, design equations, or some other guiding input.

4.4.3.3 Simulations

- Where appropriate, demonstrate circuit simulations, test bench operation, or some other form of computer verification of operation prior to deployment.

4.4.3.4 Approach to coding and testing

- Describe your approach to microcontroller coding and code testing, with selected examples.

4.5 Subsystem Qualification Tests Results

- Include ONLY the following details of the qualification tests in table format:

- o Specification to be verified (as per the MARV practical guide)
 - o Expected Results, based on your design efforts and simulations. Be specific!
 - o Provide the measured results of your tests

4.6 Subsystem Conclusions and Recommendations

- Does the subsystem adhere to the requirements set out? If not, why not?
- What are the benefits and shortcomings of the subsystem?
- What future work do you recommend for the subsystem?

4.7 Lab Book Excerpts

- 4.7.1 Design Constraints**
- 4.7.2 Trade-offs**
- 4.7.3 Engineering tools**
- 4.7.4 Engineering methods**
- 4.7.5 Statements of requirements**
- 4.7.6 Development**
- 4.7.7 Simulations**
- 4.7.8 Approach to coding and testing**