Oregon State University

ECE341
Final Project Design Document

AUTHORS

John Fletcher - 934-491-185
Sam Felsted - 934-505-492
Luka Radovic - 934-514-337

2025-03-09

Contents

1 Video Link **1**

2 Team Member Work Distribution **1**

3 System level Block Diagram **2**

4 System Description **3**

5 System Design Validation **3**

5.1 Top Level Architecture . . . 3
5.2 Block 1 Design Details, DC Voltage Regulator . . . 4
5.2.1 Diagram . . . 4
5.2.2 Description . . . 4
5.2.3 Interface Validation Table . . . 5
5.2.4 Artifacts . . . 7
5.3 Block 2 Design Details, MCU . . . 10
5.3.1 Diagram . . . 10
5.3.2 Description . . . 10
5.3.3 Interface Validation Table . . . 11
5.3.4 Block 2.1 DSP . . . 14
5.3.5 I2C Bus . . . 18
5.3.6 WIFI TRX . . . 20
5.4 Block 3 Design Details, Distance Sensor . . . 25
5.4.1 Diagram . . . 25
5.4.2 Description . . . 25
5.4.3 Interface Validation Table . . . 26
5.4.4 Artifacts . . . 29
5.5 Block 4 Design details, WiFi Antenna . . . 30
5.5.1 Diagram . . . 30
5.5.2 Description . . . 31
5.5.3 Interface Validation Table . . . 31
5.5.4 Artifacts . . . 33
5.6 Block 5 Design Details, Firmware . . . 36
5.6.1 Diagram . . . 36
5.6.2 Description . . . 36
5.6.3 Interface Validation Table . . . 37
5.6.4 Artifacts . . . 39

6 System Level Interface Validation Table **41**

7 Engineering Requirements **47**

8 Verification Process **48**

List of Figures **I**

Nomenclature **III**

References **IV**

9 Appendix A **V**

1 Video Link

A video demonstration of the project is available at `https://media.oregonstate.edu/media/t/1_jbm76fye`.

2 Team Member Work Distribution

Name	Contributions	Hours worked (total)
John Fletcher	• Wrote code for moving average DSP algorithm • Created Latex documentation • Simulated PCB Antenna • Wrote documentation sections 3,4,5.2, 5.3, 5.5 ,6,7,8,9 • Created Figures	35 hours
Sam Felsted	• Wrote ESP32 embedded code • Created ESP32 web server and HTML web interface • Wrote documentation sections 2, 3, 4, 5.6, 7 • Recorded final video • Added citations and page numbers • Maintained Git Repository	35 hours
Luka Radovic	• Recorded final video • Wrote nomenclature • Wrote Documentation section 4, 5.4	25 hours

3 System level Block Diagram

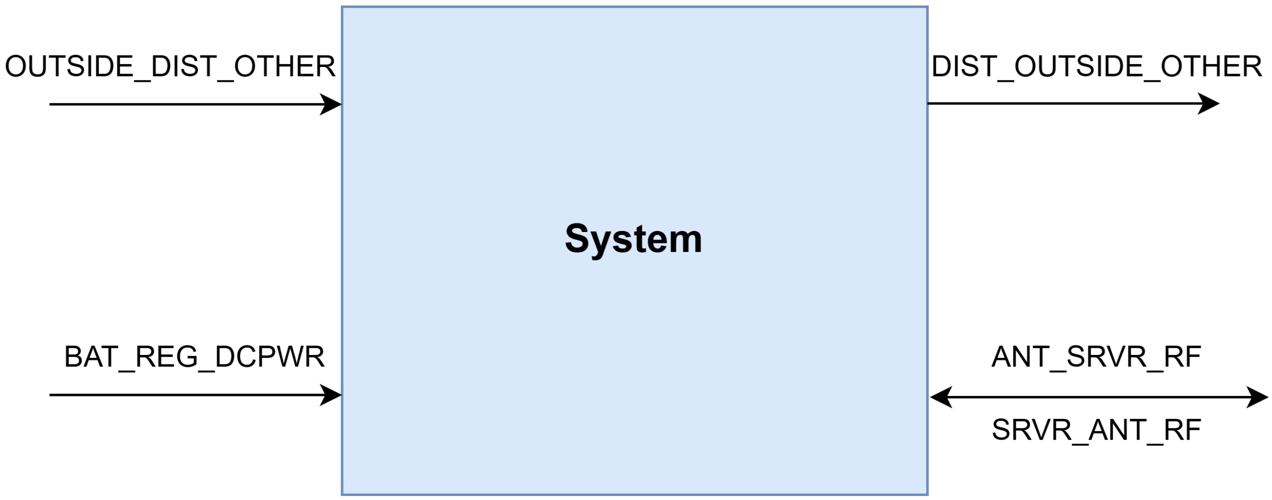

Figure 1: System level block diagram for the portable sensor

4 System Description

The Distance Sensor System is designed to actively measure and display the distance between the sensor and an object positioned directly in its line of sight, within a range of 0.1 to 1.2 meters. The core component of this system is the ESP32 microcontroller unit (MCU), which interfaces directly with a VL53L0X Time-of-Flight (ToF) distance sensor. The sensor measures distances by emitting an infrared laser pulse at a wavelength of 940 nm and timing how long the reflected pulse takes to return after hitting an object. The system maintains an accuracy margin of error within $\pm 10\%$, ensuring reliable performance suitable for our project/assignment applications. The distance measurements that are obtained from the VL53L0X sensor are processed by the ESP32 MCU using a digital signal processing (DSP) algorithm, specifically an M-tap moving average filter. This filtering process enhances measurement stability by reducing noise and smoothing out short-term variations, providing consistent and accurate data readings even under rapidly changing conditions. The processed sensor data is then communicated to the user through a Wi-Fi web server hosted directly by the ESP32 MCU. Users can access the measurement data by connecting to the dedicated Wi-Fi network named "ESP" and opening an HTTP request to the IP address 192.168.4.1 using a standard web browser. Power to the system is provided by a standard 9-volt battery. Voltage from this battery is regulated down to 5-volts output using an NCP1117 Low Dropout (LDO) voltage regulator, ensuring consistent operation of both the MCU and the sensor components.

5 System Design Validation

5.1 Top Level Architecture

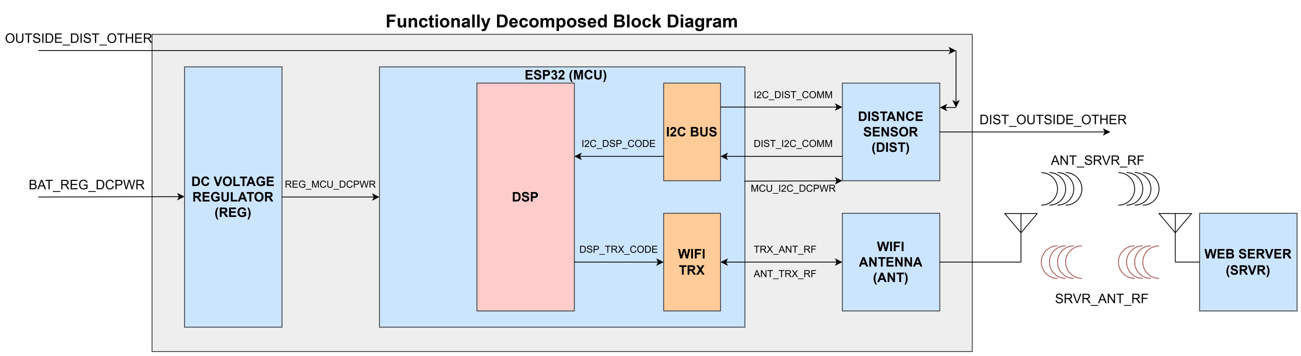

Figure 2: System Architecture

5.2 Block 1 Design Details, DC Voltage Regulator

5.2.1 Diagram

Figure 3: Voltage Regulator Diagram

5.2.2 Description

The DC voltage regulator block (VREG) is built around an NCP1117 Low Drop Out (LDO) voltage regulator. The implementation of the VREG in this block is specified to handle an input voltage range from 7-10V, and has a peak output current of 450mA.
The implementation of the voltage regulator is done on a PCB separate to that which is carrying the MCU. The VREG is implemented with reverse current protection from a schotky diode, as well as input and output decoupling.
The VREG board has 2 screw connectors as well as a vertical 2 pin connector for connecting external devices.

5.2.3 Interface Validation Table

outside_regulator_dcpwr

Interface Property	Value Justification	System Performance Justification
Vmin: 7 V	• This value was chosen as it is the minimum expected voltage of the battery with extra buffer	In the case of a charged 9V battery, the voltage should not ever be below 7V. This is also within the operational range of the NCP1117 LDO [[1], p. 3] regulator in the configuration used.
Vmax: 10 V	• This value was chosen to be above the maximum voltage of a 9V battery.	For a charged 9V battery, the voltage should not exceed 10V. This is also within the operational range of the NCP1117 LDO regulator [[1], p. 3] in the configuration used.
Inominal: 200 mA	• Nominal current draw of ESP32 while operating WiFi with 20mA buffer	Nominal current draw will vary greatly with operating condition, but this is a conservative estimate for an ESP32 [[2], p. 53]. 260 mA is the highest power configuration for this system.
Ipeak: 450 mA	• Peak current draw ESP32 [[2], p. 53] While transmitting WiFi packet at peak power with buffer for external devices and +50mA above peak output current for internal loss compensation	Nominal current draw will vary greatly with operating condition, but this is a conservative estimate.

regulator_outside_dcpwr

Interface Property	Value Justification	System Performance Justification
Vmin: 4.85 V	• Minimum output specified by datasheet	This is harsher than the NCP1117 [[1], p. 3] datasheet spec with added buffer.
Vmax: 5.15 V	• Maximum output specified by NCP1117 [[1], p. 3] datasheet	This is harsher than the datasheet spec with added buffer.
Inominal: 180 mA	• Nominal current draw of ESP32	Nominal current draw will vary greatly with operating condition, but this is a conservative estimate (Peak for an ESP32 is 260 in the highest power configuration)
Ipeak: 400 mA	• Peak current draw ESP32 While transmitting WiFi packet at peak power with buffer for external devices	Conservative estimate and within all bounds specified by the datasheet with buffer

5.2.4 Artifacts

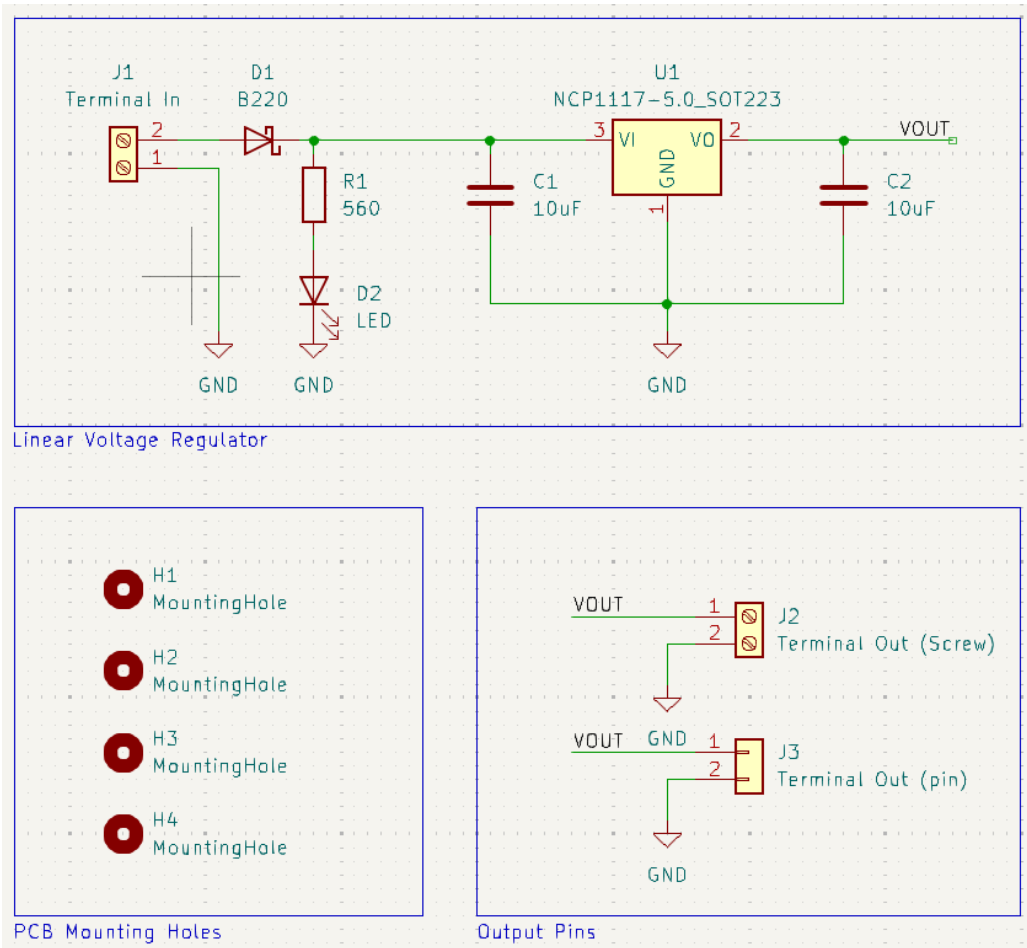

Figure 4: Voltage Regulator Schematic

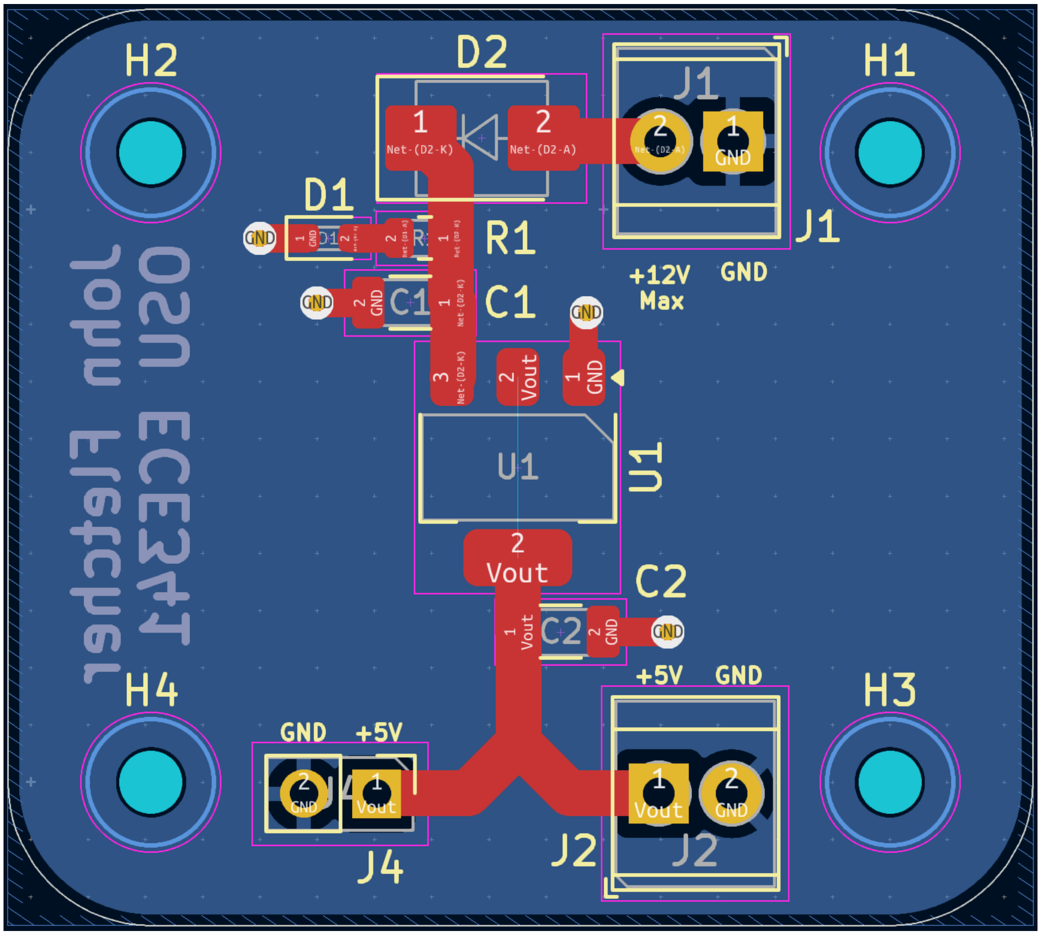

Figure 5: Voltage Regulator PCB Layout

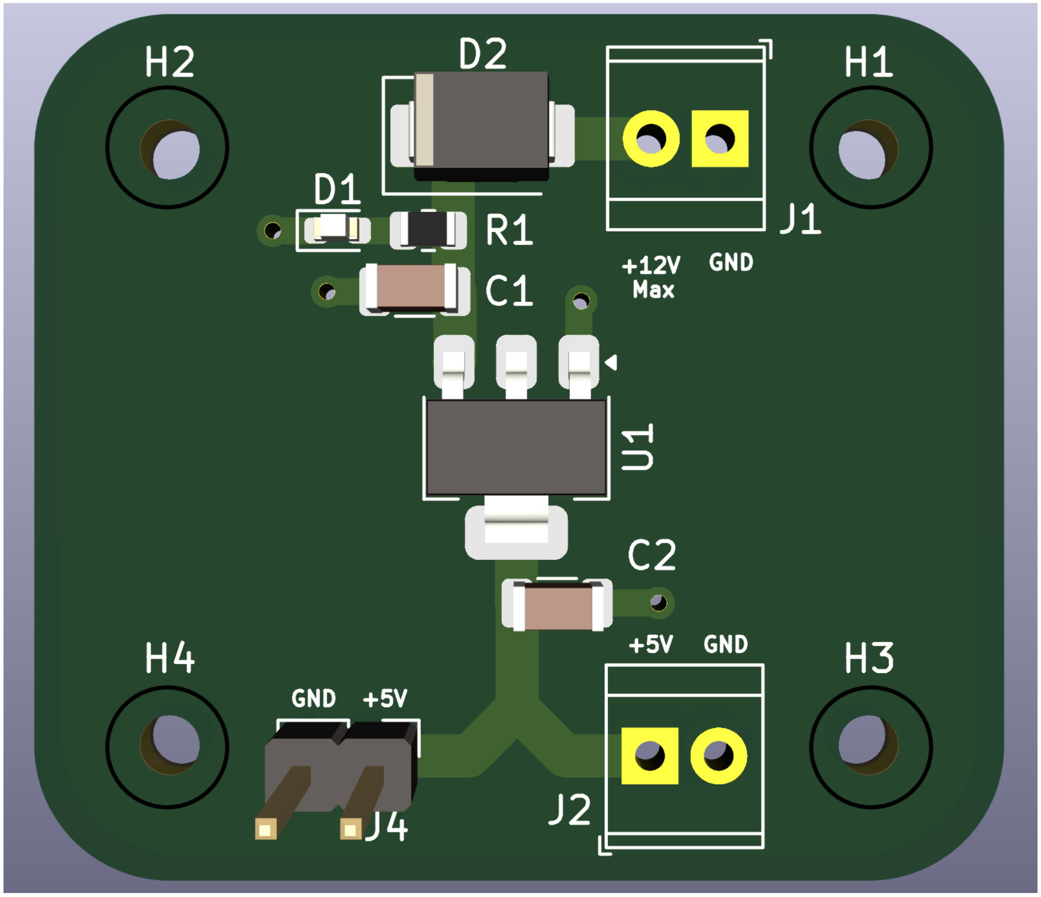

Figure 6: Voltage Regulator 3D Model

5.3 Block 2 Design Details, MCU

5.3.1 Diagram

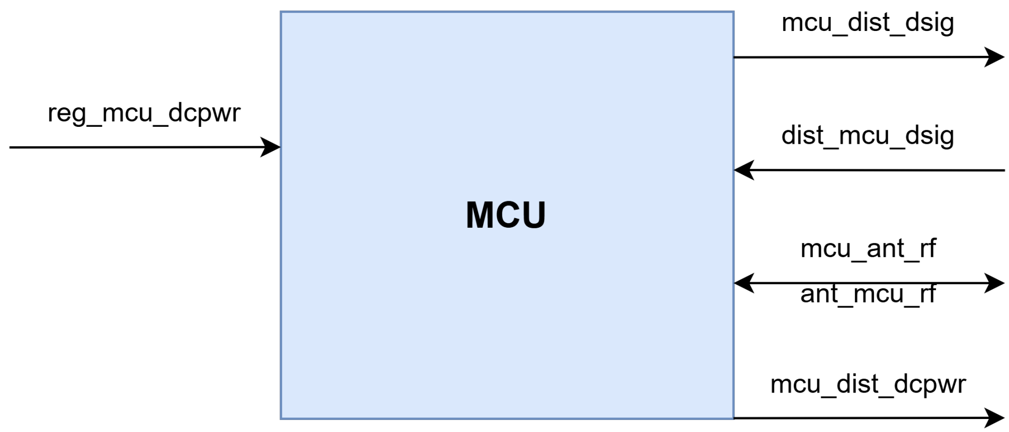

Figure 7: MCU Diagram

5.3.2 Description

The Micro-Controller Unit is an ESP32 WROOM microcontroller layed out on a development board pcb. The MCU contains 3 sub-blocks, those being the **DSP [Block 2.1]**, the **I2C Bus [Block 2.2]**, and the **WIFI TRX [Block 2.3]**. These sub blocks are handled within the MCU itself in the form of code (**DSP**), on chip hardware (**WIFI TRX**), or a mix of both (**I2C BUS**). The MCU block diagram shows the inputs and outputs at the level of the MCU. Note that the level of abstraction for Fig. 7 is one level above that of Fig. 2

5.3.3 Interface Validation Table

reg_mcu_dcpwr

Interface Property	Value Justification	System Performance Justification
Vmin: 4.85 V	• Maximum output specified by NCP1117 [[1], p. 3] datasheet	This is harsher than the datasheet spec with added buffer.
Vmax: 5.15 V	• Maximum output specified by NCP1117 [[1], p. 3] datasheet	This is harsher than the datasheet spec with added buffer.
Inominal: 180 mA	• Nominal current draw of ESP32 [[2], p. 53]	Nominal current draw will vary greatly with operating condition, but this is a conservative estimate (Peak for an ESP32 is 260 in the highest power configuration)
Ipeak: 400 mA	• Peak current draw ESP32 [[2], p. 53] While transmitting WiFi packet at peak power with buffer for external devices	Conservative estimate and within all bounds specified by the datasheet with buffer

mcu_dist_dsig

Interface Property	Value Justification	System Performance Justification
SCL: Clock Signal	• This signal is required for the I2C protocol is the clock source for synchronizing the DIST and I2C BUS.	The SCL pin is connected to the DIST SCL pin. [[3], p. 4]
SDA: Data Signal	• This signal is required for the I2C protocol and carries the data from the I2C BUS to the DIST.	The SDA pin is connected to the DIST SDA pin. [[3], p. 4]

dist_mcu_dsig

Interface Property	Value Justification	System Performance Justification
SCL: Clock Signal	• This signal is required for the I2C protocol is the clock source for synchronizing the DIST and I2C BUS.	The SCL pin is connected to the DIST SCL pin.
SDA: Data Signal	• This signal is required for the I2C protocol and carries the data from the I2C BUS to the DIST.	The SDA pin is connected to the DIST SDA pin.

mcu_ant_rf

Interface Property	Value Justification	System Performance Justification
Minimum Freq: 2206 MHz	• This value was found by taking the lower 3dB bandwidth of the antenna input |S11| in simulation.	This is a property of the physical design of the antenna and the matching network it is connected to.
Maximum Freq: 2630 MHz	• This value was found by taking the upper 3dB bandwidth of the antenna input |S11| in simulation.	This is a property of the physical design of the antenna and the matching network it is connected to.
Input Impedance: 50 Ω (2.4 GHz)	• This is the standard Input Impedance for RF systems	This is a property of the physical design of the antenna and the matching network it is connected to.

antenna_mcu_rf

Interface Property	Value Justification	System Performance Justification
Minimum Freq: 2206 MHz	• This value was found by taking the lower 3dB bandwidth of the antenna input |S11| in simulation.	The WIFI Transceiver (WIFI TRX) connected to the antenna is not able to produce carriers below this value.
Maximum Freq: 2630 MHz	• This value was found by taking the upper 3dB bandwidth of the antenna input |S11| in simulation.	The WIFI Transceiver (WIFI TRX) connected to the antenna is not able to produce carriers above this value.

mcu_dist_dcpwr

Interface Property	Value Justification	System Performance Justification
Vmax: 5V	• 5V DC output pin from, MCU is regulated to 5V. The Mix input rating for the DIST is 5V.	5V pin from MCU can not provide more than this under normal operation.
Vmin: 3.3V	• 3.3V DC output pin from, MCU is regulated to 3.3V. The min input rating for the DIST is 3.3V.	3.3V pin from the MCU is well regulated and should not deviate significantly from this value

5.3.4 Block 2.1 DSP

DSP Block Diagrams Diagram

Figure 8: Top Level DSP Diagram

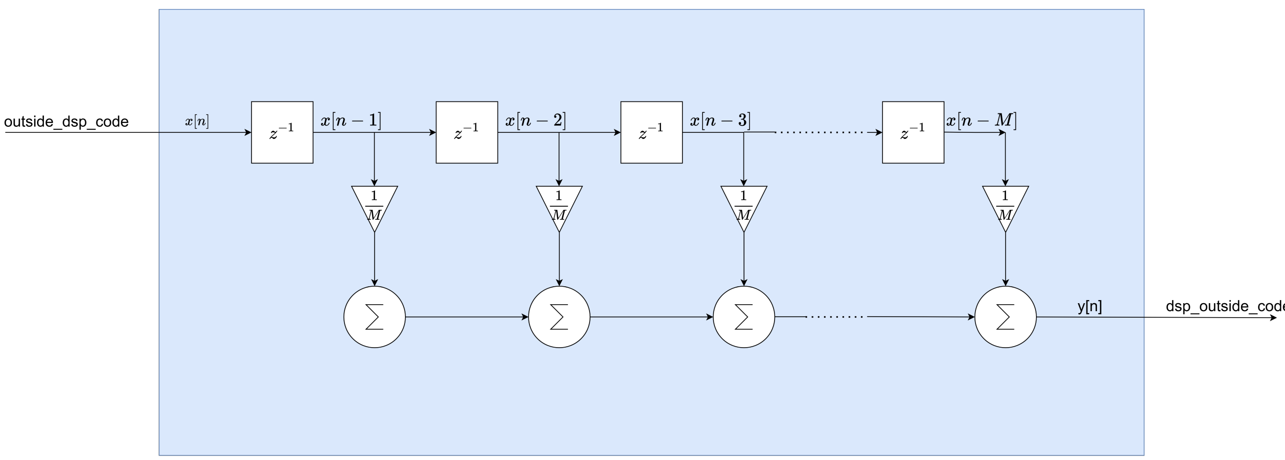

Figure 9: DSP Level Signal Flow Diagram

DSP Description

The DSP flow in this system is an M tap moving average filter. The filter was implemented in embedded C++ and is run onboard the MCU. On each MCU code loop iteration, the distance sensor loads a value into a buffer variable. The value from the sensor is then pulled by the DSP moving average function to compute the next output of the DSP block.

A moving average was deemed appropriate due to the fast measurement times of the distance sensor which provide conducive sample to sample variation for a smooth moving average. If the expected changes in value from sample to sample were expected to be very large or very (i.e. the target is moving very quickly) then this filter may not be appropriate. The primary function of a moving average filter is to smooth data and and increase stability

Interface Validation Table

outside_dsp_code

Interface Property	Value Justification	System Performance Justification
new_value: double	• a double was chosen as the input type because it the highest expected accuracy can be measured using a double. A double also allows for unit conversions as needed if desired in further implementations.	All inputs from the sensor should be integers which can be typecasted to doubles. There is no case where an input should ever require higher precision than a double.

dsp_outside_code

Interface Property	Value Justification	System Performance Justification
updated_value: double	• a double was chosen as the input type because it the highest expected accuracy can be measured using a double. A double also allows for unit conversions as needed if desired in further implementations.	There is no case where an output should ever require higher precision than a double.

Artifacts

```cpp
class MovingAverage {
private:
    std::vector<double> buffer;
    int window_size;
    double sum;
    int index;
    int count;

public:
    MovingAverage(int size) : window_size(size), sum(0), index(0), count(0) {
        buffer.resize(size, 0.0);
    }

    double update(double new_value) {
        sum -= buffer[index];
        buffer[index] = new_value;
        sum += new_value;

        index = (index + 1) % window_size;
        count = std::min(count + 1, window_size);

        return sum / count;
    }
};
```

Listing 1: MovingAverage Class Implementation

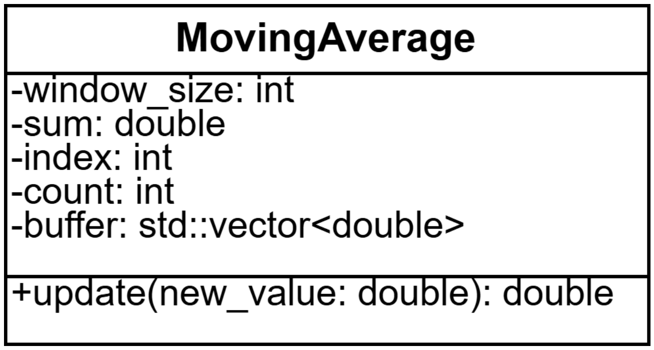

Figure 10: MovingAverage UML Class Diagram

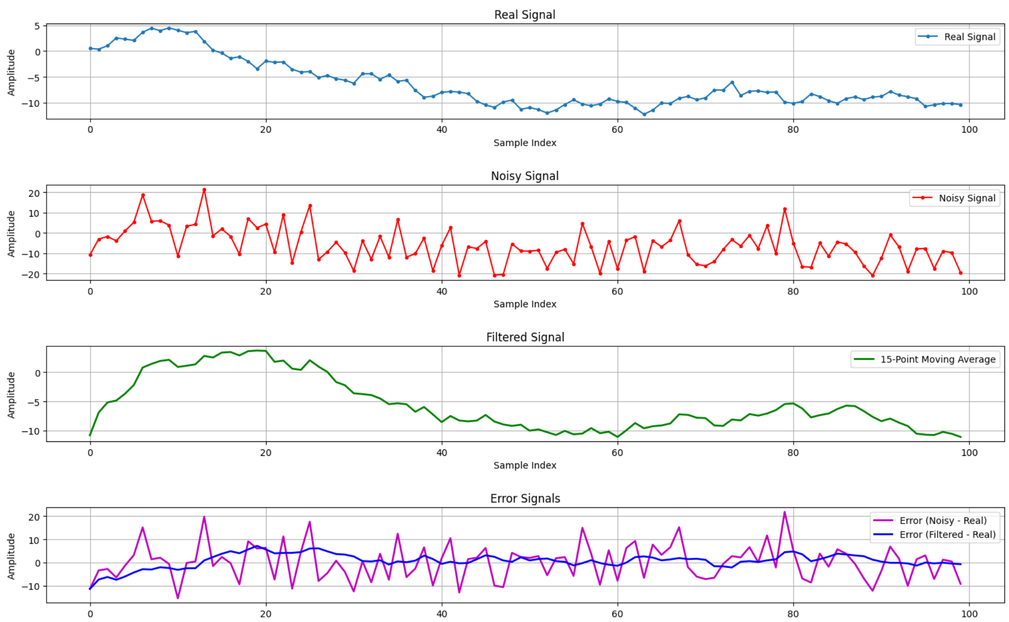

Figure 11: Example of Applying DSP to A Test Signal

5.3.5 I2C Bus

Diagram

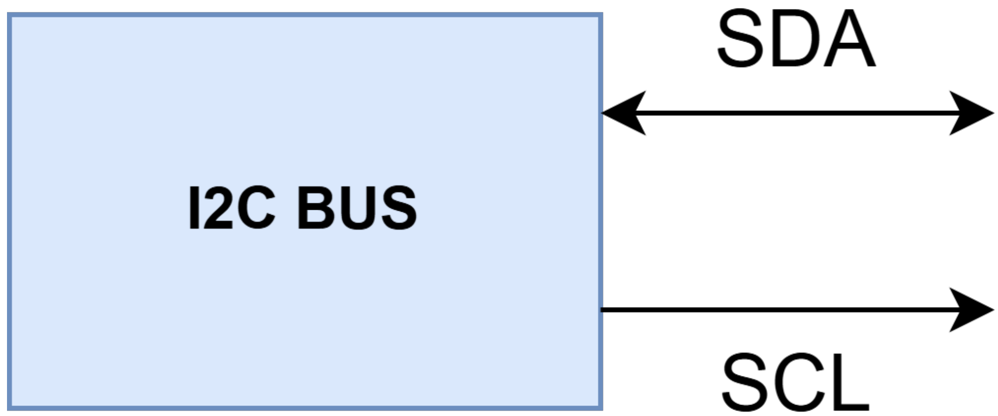

Figure 12: I2C Top Level Block Diagram

Description The I2C bus handles data used by the I2C communication interface. The I2C protocol has 2 signal lines, Serial Data (SDA), and Serial Clock (SCL). The SDA signal line is responsible for carrying the data to be used by devices connected to the bus. The SCL signal line distributes the clock signal which is used for device synchronization.

Interface Validation Table
i2c_outside_dsig

Interface Property	Value Justification	System Performance Justification
Max Clock Freq: 400 KHz	• This value was found from the ESP32 datasheet [[3], p. 38]	This value can not be exceeded because it is the maximum the I2C bus is capable of producing in Fast-Mode
Max Msg Length: 32byte	• This value was chosen as it is sufficiently large to handle any data the system will use [[3], p. 39]	The outputs from the I2C bus are user controlled. The MCU code which interfaces with the I2C bus has been written to never send data larger than 32bytes.
SCL: Clock Signal	• The SCL Clock signal is a property of the I2C protocol	The SCL signal is driven by an output pin on the ESP32 (MCU) board. Devices connected to this interface will be connected to this pin
SDA: Data Signal	• The SDA is the data carrier for the I2C protocol	The SDA signal is driven by an output pin on the ESP32 (MCU) board. Devices connected to this interface will be connected to this pin

outside_i2c_dsig

Interface Property	Value Justification	System Performance Justification
Max Clock Freq: 400 KHz	• This value was found from the datasheet [[3], p. 38]	This value can not be exceeded because it is the maximum the I2C bus is capable of producing in Fast-Mode.
Max Msg Length: 32byte	• This value was chosen as it is sufficiently large to handle any data the system will use. [[3], p. 39]	The outputs from the I2C bus are user controlled. The MCU code which interfaces with the I2C bus has been written to never send data larger than 32bytes.
SDA: Data Signal	• The SDA is the data carrier for the I2C protocol.	Outside devices will be connected to the i2c SDA pin.

5.3.6 WIFI TRX

Diagram

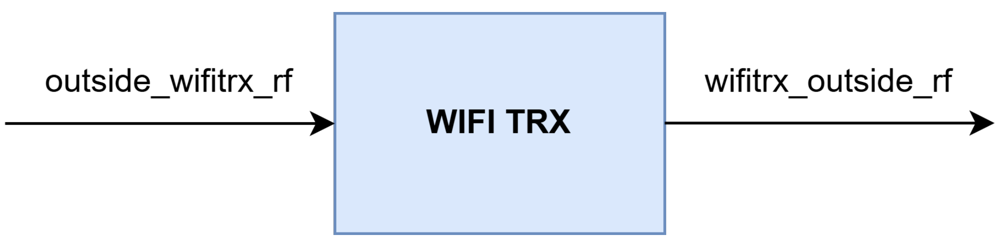

Figure 13: WIFI TRX Block Diagram

Description

The WIFI transceiver (WIFI TRX) in the ESP32 is an analog subsection of the ESP32 microcontroller responsible for handling the transmission and reception of WIFI signals. The WIFI TRX in the ESP32 WROOM is capable of transmitting and receiving signals which conform to the 802.11b/g/n WIFI standard which is the protocol utilized in this system.

Interface Validation Table

wifirx_outside_rf: Output

Interface Property	Value Justification	System Performance Justification
WiFi Protocols: 802.11b/g/n	• The WiFi protocol defines what types of routers and devices the system can inteface with	These protocols are the only ones which are supported by the ESP32 WROOM onboard RF Transceiver. [[2], p. 33]
Carrier Freq: 2412-2484 MHz	• The carrier frequency of the transmitted data is important for many parameters, like the range of the system. The center frequency is also one of the most important parameters for antenna design. The range of frequencies was chosen to meet FCC and IEEE standard requirements.	Output frequency is limited both by the bandwidth of the antenna, as well as the internal transceiver bandwidth limitations set by the chosen WiFi protocol. The stated range of values are given in the datasheet. [[2], p. 33]
TX Pwr Typ: 13-19.5 dBm	• The transmitter power is a determining factor in the maximum range which at which the transmitter can send data to the receiver without data loss. The value range is specified by FCC and IEEE standard requirements	The ESP32 applies a spectral mask to its output as well as filtering to conform to the 802.11 WiFi standard and requirements for output power. The range of values is due to varying peak TX power based on WiFi protocol and bandwidth. The stated range of values are given in the datasheet. [[2], p. 33]

Bit Rate: Up to 150Mbps (802.11n)	• Value found in datasheet	This value is bounded by the design of the transceiver and input waveforms. This value will not be exceeded due to the transceiver conforming to the 802.11n specification [[2], p. 33]
Min Guard Intvl: $0.4\mu s$	• Value found in datasheet	This value is bounded by the design of the transceiver and input waveforms. This value will not be exceeded due to the transceiver conforming to the 802.11n specification

outside_wifitrx_rf: Output

Interface Property	**Value Justification**	**System Performance Justification**
WiFi Protocols: 802.11b/g/n	• The WiFi protocol defines what types of routers and devices the system can inteface with	These protocols are the only ones which are supported by the ESP32 WROOM onboard RF Transceiver.
Carrier Freq: 2412-2484 MHz	• The carrier frequency of the transmitted data is important for many parameters, like the range of the system. The center frequency is also one of the most important parameters for antenna design. The range of frequencies was chosen to meet FCC and IEEE standard requirements.	Output frequency is limited both by the bandwidth of the antenna, as well as the internal transceiver bandwidth limitations set by the chosen WiFi protocol. The stated range of values are given in the datasheet.
RX Sensitivity Typ: -89 dBm dBm	• The transmitter power is a determining factor in the maximum range which at which the transmitter can send data to the receiver without data loss. The value range is specified by FCC and IEEE standard requirements	The ESP32 applies a spectral mask to its output as well as filtering to conform to the 802.11 WiFi standard and requirements for output power. The range of values is due to varying peak TX power based on WiFi protocol and bandwidth. The stated range of values are given in the datasheet. [[3], p. 9]

Bit Rate: Up to 150Mbps (802.11n)	• Value found in datasheet	This value is bounded by the design of the transceiver and input waveforms. This value will not be exceeded due to the transceiver conforming to the 802.11n specification
Min Guard Intvl: $0.4 \mu s$	• Value found in datasheet	This value is bounded by the design of the transceiver and input waveforms. This value will not be exceeded due to the transceiver conforming to the 802.11n specification [[3], p. 1]

5.4 Block 3 Design Details, Distance Sensor

5.4.1 Diagram

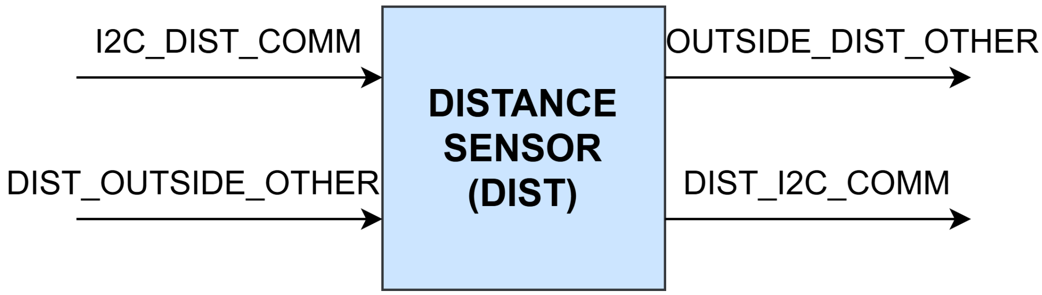

Figure 14: Distance Sensor Block Diagram

5.4.2 Description

The VL53L0X (DIST) implemented in this system is a Ranging Time-of-Flight(ToF) distance sensor based around a 940nm IR laser. The distance sensor determines the range to a target within 1.2m of range to an accuracy of 10% by measuring the time of flight of a transmitted laser pulse. The DIST conforms to the Class 1 Laser safety rating as specified under IEC 60825-1:2014. Communication between the DIST and MCU is handled via the I2C Bus interface. The I2C interface on the DIST consists of SDA and SCL pins which are connected to the MCU SDA and SCL pins. Power is provided to the DIST from the MCU by either a 3.3V or 5V DC and ground pins.

5.4.3 Interface Validation Table

For more detailed information on I2C interfaces, see section 5.3.6.

i2c_dist_comm

Interface Property	Value Justification	System Performance Justification
SCL: Clock Signal	• This signal is required for the I2C protocol is the clock source for synchronizing the DIST and I2C BUS.	The SCL pin is connected to the DIST SCL pin.
SDA: Data Signal	• This signal is required for the I2C protocol and carries the data from the I2C BUS to the DIST.	The SDA pin is connected to the DIST SDA pin.

dist_i2c_comm

Interface Property	Value Justification	System Performance Justification
SCL: Clock Signal	• This signal is required for the I2C protocol is the clock source for synchronizing the DIST and I2C BUS.	The SCL pin is connected to the DIST SCL pin.
SDA: Data Signal	• This signal is required for the I2C protocol and carries the data from the I2C BUS to the DIST.	The SDA pin is connected to the DIST SDA pin.

mcu_dist_dcpwr

Interface Property	**Value Justification**	**System Performance Justification**
Vmax: 5V	• 5V DC output pin from, MCU is regulated to 5V. The Mix input rating for the DIST is 5V.	5V pin from MCU can not provide more than this under normal operation. [[4], p. 2]
Vmin: 3.3V	• 3.3V DC output pin from, MCU is regulated to 3.3V. The min input rating for the DIST is 3.3V.	3.3V pin from the MCU is well regulated and should not deviate significantly from this value

_outside_dist_other: Input

Interface Property	**Value Justification**	**System Performance Justification**
Wavelength: 940 nm	• This is the wavelength specified in the datasheet.	Given in the datasheet and is defined by the design of the VCSEL (Vertical Cavity Surface Emitting Laser) [[4], p. 1]
Pmax: 4.0mW	• This value must be at or below 4.0mW to meet the Class 1A safety rating	This must always be less than the transmitted power which is limited to 4.0mW [[4], p. 1]
Laser Safety Rating: Class 1A	• Laser safety rating is important for making sure that the user has proper personal protective equipment (PPE) when using the device. The class 1 rating in this case means no PPE is required to use this device and that it is eyesafe.	This device will conform to this requirement as it has undergone substantial test and verification and is stated to comply with IEC 60825-1:2014. [[4], p. 1]

dist_outside_other: Output

Interface Property	**Value Justification**	**System Performance Justification**
Wavelength: 940 nm	• This is the wavelength specified in the datasheet.	Given in the datasheet and is defined by the design of the VCSEL (Vertical Cavity Surface Emitting Laser)
Pmax: 4.0mW	• This value must be at or below 4.0mW to meet the Class 1A safety rating	Given since the laser meets the Class 1A safety rating
Laser Safety Rating: Class 1A	• Laser safety rating is important for making sure that the user has proper personal protective equipment (PPE) when using the device. The class 1 rating in this case means no PPE is required to use this device and that it is eyesafe.	This device will conform to this requirement as it has undergone substantial test and verification and is stated to comply with IEC 60825-1:2014.

5.4.4 Artifacts

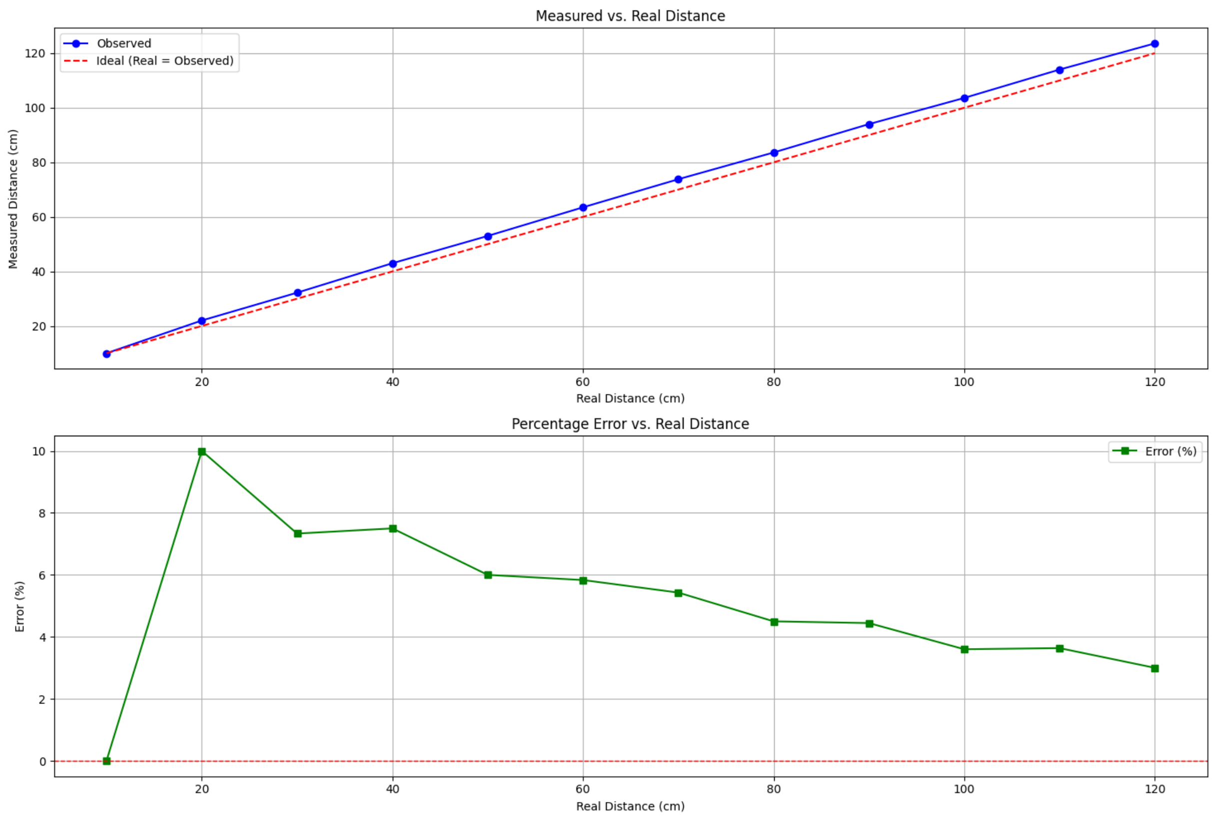

Figure 15: System Measured vs Real Range and Error (%) vs Real Range

5.5 Block 4 Design details, WiFi Antenna

5.5.1 Diagram

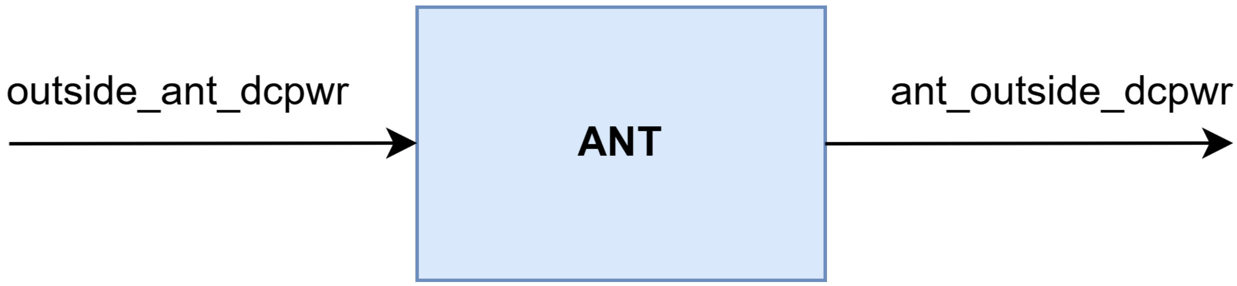

Figure 16: Antenna Top Level Block Diagram

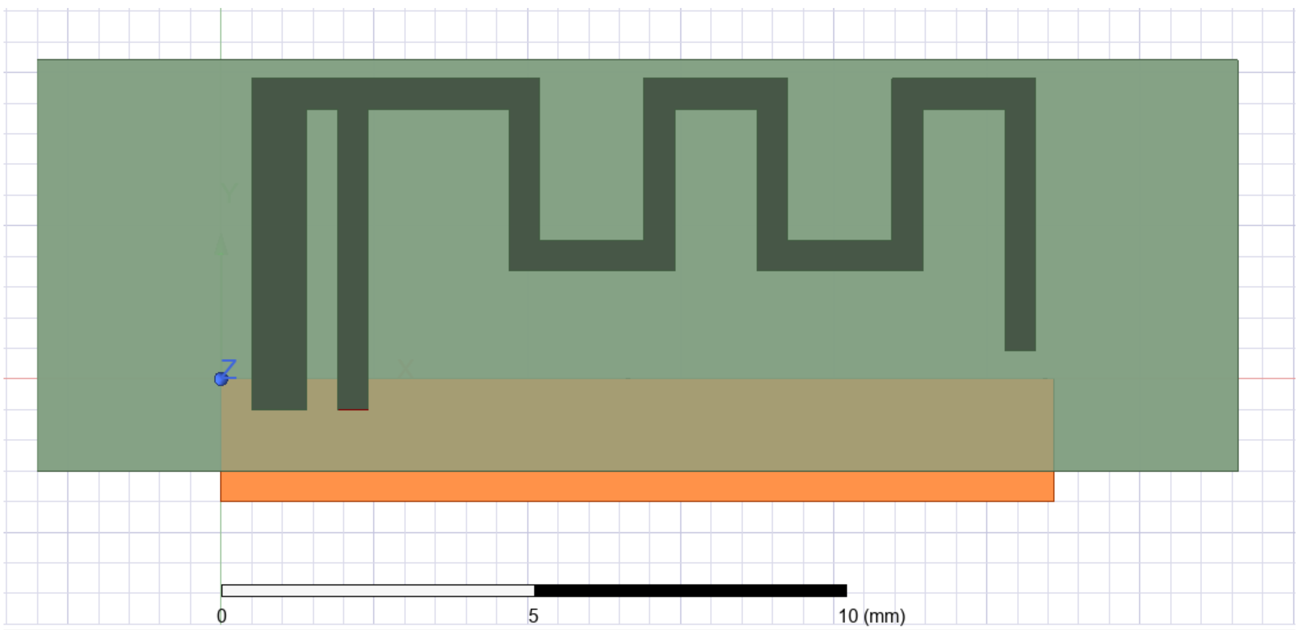

Figure 17: Top Down Model of Antenna

5.5.2 Description

The antenna used on the ESP32 WROOM is a 2.4GHz center frequency (CF) meandered inverted-F antenna on an FR-4 PCB. This design is ideal for limited space boards and is commonly used in conjunction with small micro controllers due to their simple design, good in-band matching, and relatively wide fractional bandwidth ($BWfrac = 15\% - 18\%$) (see artifacts). For the application of being used on a microcontroller board, a meandered inverted-F is also well suited due to its radiation pattern being relatively isotropic when resting face up (see artifacts).

5.5.3 Interface Validation Table

outside_antenna_rf

Interface Property	Value Justification	System Performance Justification
Minimum Freq: 2206 MHz	• This value was found by taking the lower 3dB bandwidth of the antenna input |S11| in simulation.	This is a property of the physical design of the antenna and the matching network it is connected to.
Maximum Freq: 2630 MHz	• This value was found by taking the upper 3dB bandwidth of the antenna input |S11| in simulation.	This is a property of the physical design of the antenna and the matching network it is connected to.
Input Impedance: 50 Ω (2.4 GHz)	• This is the standard Input Impedance for RF systems	This is a property of the physical design of the antenna and the matching network it is connected to.

antenna_outside_rf

Interface Property	Value Justification	System Performance Justification
Minimum Freq: 2206 MHz	• This value was found by taking the lower 3dB bandwidth of the antenna input |S11| in simulation.	The WIFI Transceiver (WIFI TRX) connected to the antenna is not able to produce carriers below this value.
Maximum Freq: 2630 MHz	• This value was found by taking the upper 3dB bandwidth of the antenna input |S11| in simulation.	The WIFI Transceiver (WIFI TRX) connected to the antenna is not able to produce carriers above this value.
Maximum Gain (at 2.4GHz): -10 dBi	• This value was found from simulation with 2dB subtracted for possible variation in manufacturing and/or dielectric performance.	The specified minimum gain has sufficient buffer from the simulated value that the real antenna should not be outside this range
Minimum Gain (at 2.4GHz): -30 dBi	• This value was found from simulation with 8dB subtracted for possible in manufacturing and/or dielectric performance.	The specified minimum gain has sufficient buffer from the simulated value that the real antenna should not be outside this range

5.5.4 Artifacts

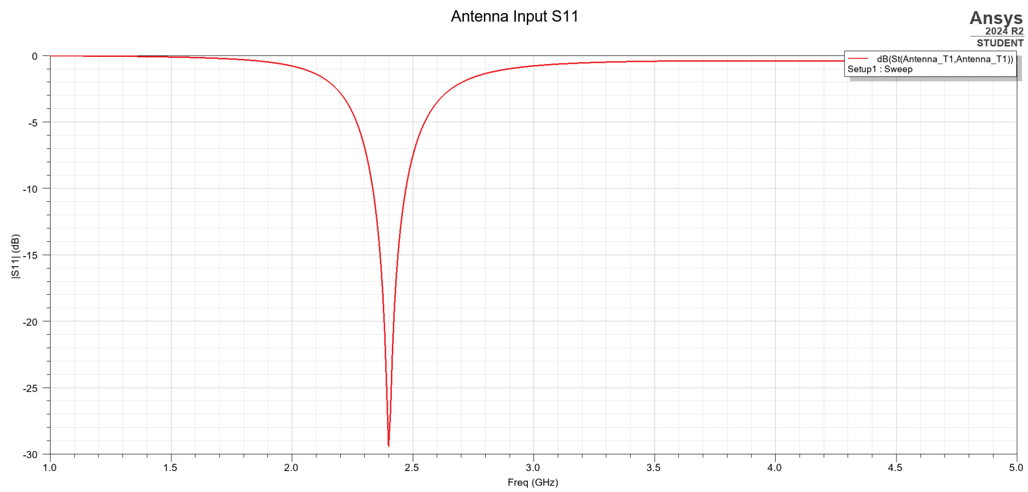

Figure 18: Antenna Simulated S11

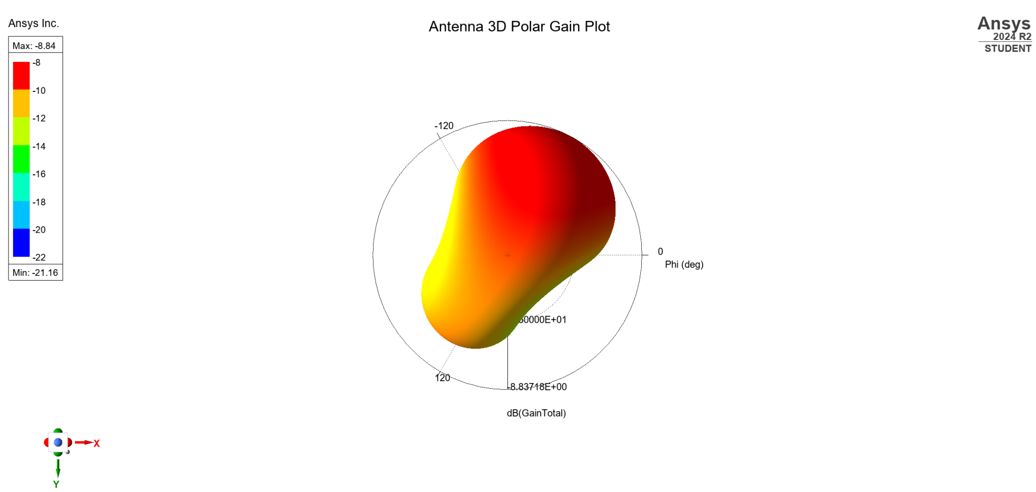

Figure 19: Antenna Gain Pattern At 2.4GHz From +Z

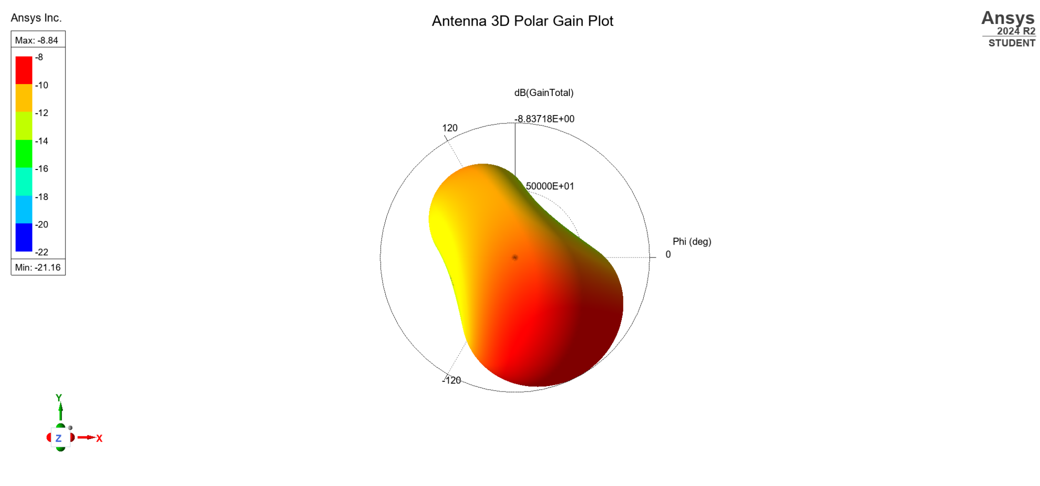

Figure 20: Antenna Gain Pattern At 2.4GHz From -Z

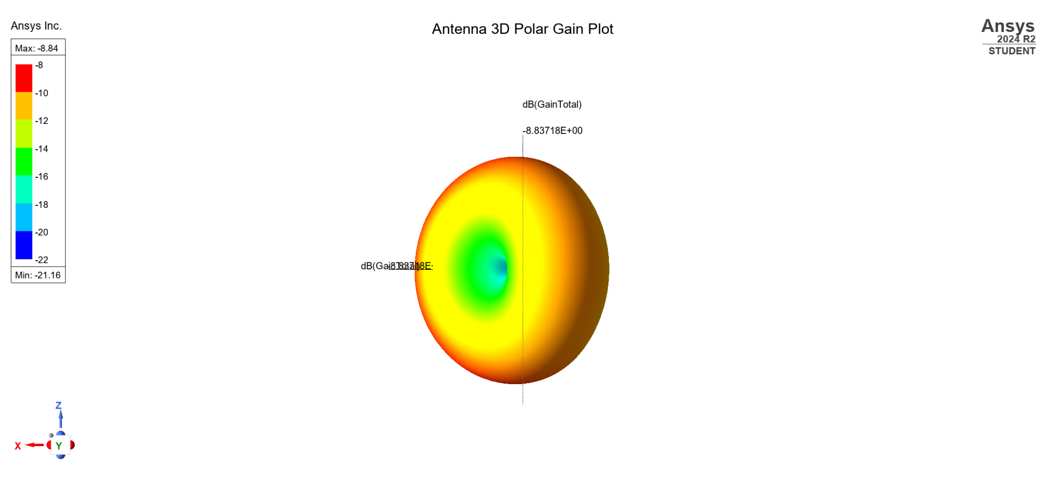

Figure 21: Antenna Gain Pattern At 2.4GHz From +Y

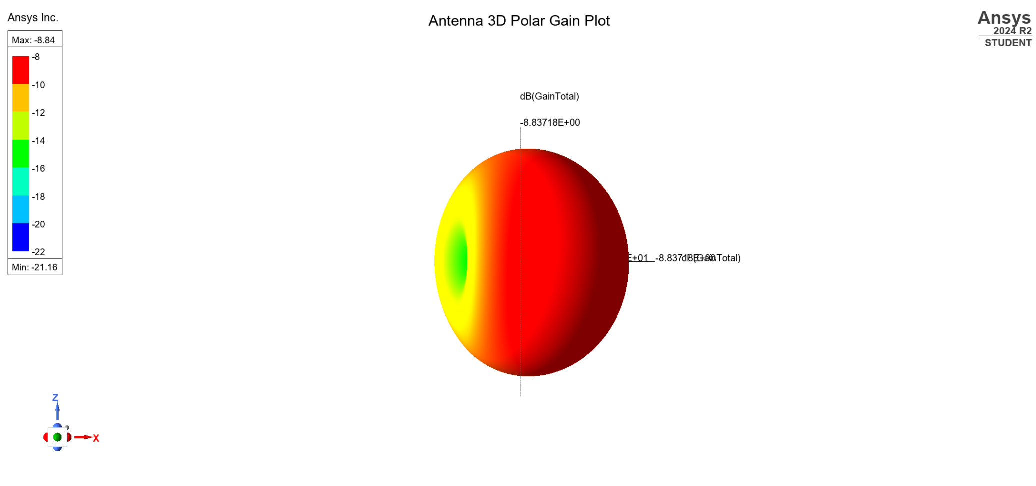

Figure 22: Antenna Gain Pattern At 2.4GHz From -Y

Oregon State University College of Engineering

5.6 Block 5 Design Details, Firmware

5.6.1 Diagram

5.6.2 Description

Firmware for the distance sensor is written in C++ and implements the Arduino micro controller framework using PlatformIO. This project utilizes the Adafruit VL53l0X library for interfacing with the ranging sensor and the me-no-dev AysncWebServer library for networking. Hosted on the ESP32 is an web server which sends HTML data to the user over Wi-Fi to provide a GUI client for reporting the system's output data.

Reported data from the sensor gets processed through the moving average filter object and then sent to the network interface object. The user's HTML client runs a JavaScript subroutine to fetch data updates

5.6.3 Interface Validation Table

esp32_main_firmware

Interface Property	Value Justification	System Performance Justification
lox: Adafruit_VL53L0X	• This is the API object for interfacing with the DIST	This follows the example code provided by Adafruit [5]
networkInterface: NetworkInterface	• This is the API object for interfacing with the DIST	This object is based on the AsyncWebServer library [6]
window_size: int	• This was chosen to provide smooth data on the output	10 samples is sufficent to provide a clean filter
filter: moving average	• This implements the moving average algorithm.	The algorithm is described previously in section **5.3**
scanI2C()	• This is used for debugging the I2C bus.	The program will loop through every address on the bus to scan for devices. full code is avaible on the GitHub.

Setup()	• All program configuration occurs here	This follows the standard Arduino code structure and the template provided by Adafruit [5].
Loop()	• This function gets called every cycle of the MCU and implements the data read of the DIST, applies filtering, and send the data to the web server.	This follows the standard Arduino code structure and the template provided by Adafruit [5].

esp32_network_interface

Interface Property	Value Justification	System Performance Justification
lox: Adafruit_VL53L0X	• This is the API object for interfacing with the DIST sensor.	• Ensures accurate distance measurements. • Maintains system efficiency by utilizing a well-supported library.
server: AsyncWebServer	• Enables asynchronous web server communication.	• Improves responsiveness and reduces latency. • Enhances system stability by handling multiple connections efficiently.
localIP: IPAddress	• Stores the ESP32's local network IP address.	• Allows reliable network communication. • Ensures correct routing and connectivity.

localGateway: IPAddress	• Defines the gateway address for network communication.	• Ensures proper routing of packets. • Facilitates communication with external networks.
subnet: IPAddress	• Specifies the subnet mask for network segmentation.	• Ensures correct IP address allocation. • Enhances security by defining network boundaries.
pos: double	• Stores positional data as a double for high precision.	• Provides sufficient accuracy for calculations. • No need for higher precision than a double.

5.6.4 Artifacts

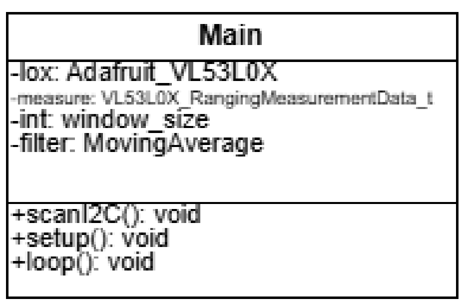

Figure 23: Main UML Class Diagram

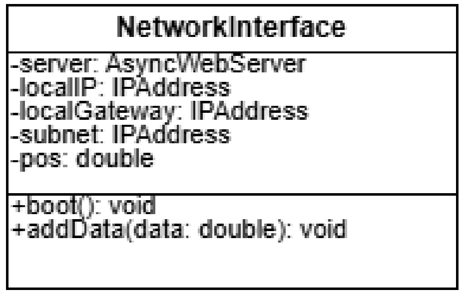

Figure 24: Networking Interface UML Class Diagram

```
<script>
  function updateSensor() {
    fetch("/sensor") // Request data from ESP32
            .then(response => response.json()) // Parse JSON response
            .then(data => {
              document.getElementById("sensorPos").innerText = (data.pos + " (cm)");
            })
            .catch(error => console.error("Error fetching data:", error));
  }

  setInterval(updateSensor, 500); // Update every second
</script>
```

Listing 2: JavaScript subroutine

The full firmware is hosted on GitHub and can be accessed at: `https://github.com/SamFelsted/ESP32DistanceSensor`

6 System Level Interface Validation Table

BAT_REG_DCPWR: Input

Interface Property	Value Justification	System Performance Justification
Vmin: 7 V	• This value was chosen as it is the minimum expected voltage of the battery with extra buffer	In the case of a charged 9V battery, the voltage should not ever be below 7V. This is also within the operational range of the NCP1117 LDO regulator in the configuration used [[1], p. 3] .
Vmax: 10 V	• This value was chosen to be above the maximum voltage of a 9V battery.	For a charged 9V battery, the voltage should not exceed 10V. This is also within the operational range of the NCP1117 LDO regulator in the configuration used. [[1], p. 3]
Inominal: 200 mA	• Nominal current draw of ESP32 while operating WiFi with 20mA buffer [[2], p. 53]	Nominal current draw will vary greatly with operating condition, but this is a conservative estimate

Ipeak: 450 mA	• Peak current draw ESP32 While transmitting WiFi packet at peak power with buffer for external devices and +50mA above peak output current for internal loss compensation	Conservative estimate and within all bounds specified by the datasheet with buffer

OUTSIDE_DIST_OTHER: Input

Interface Property	Value Justification	System Performance Justification
Wavelength: 940 nm	• This is the wavelength specified in the datasheet.	Given in the datasheet and is defined by the design of the VCSEL (Vertical Cavity Surface Emitting Laser)
Pmax: 4.0mW	• This value must be at or below 4.0mW to meet the Class 1A safety rating	This must always be less than the transmitted power which is limited to 4.0mW
Laser Safety Rating: Class 1A	• Laser safety rating is important for making sure that the user has proper personal protective equipment (PPE) when using the device. The class 1 rating in this case means no PPE is required to use this device and that it is eyesafe.	This device will conform to this requirement as it has undergone substantial test and verification and is stated to comply with IEC 60825-1:2014.

DIST_OUTSIDE_OTHER: Output

Interface Property	Value Justification	System Performance Justification
Wavelength: 940 nm	• This is the wavelength specified in the datasheet.	Given in the datasheet and is defined by the design of the VCSEL (Vertical Cavity Surface Emitting Laser)
Pmax: 4.0mW	• This value must be at or below 4.0mW to meet the Class 1A safety rating	Given since the laser meets the Class 1A safety rating
Laser Safety Rating: Class 1A	• Laser safety rating is important for making sure that the user has proper personal protective equipment (PPE) when using the device. The class 1 rating in this case means no PPE is required to use this device and that it is eyesafe.	This device will conform to this requirement as it has undergone substantial test and verification and is stated to comply with IEC 60825-1:2014.

ANT_SRVR_RF: Output

Interface Property	Value Justification	System Performance Justification
WiFi Protocols: 802.11b/g/n	• The WiFi protocol defines what types of routers and devices the system can inteface with	These protocols are the only ones which are supported by the ESP32 WROOM onboard RF Transceiver.
Carrier Freq: 2412-2484 MHz	• The carrier frequency of the transmitted data is important for many parameters, like the range of the system. The center frequency is also one of the most important parameters for antenna design. The range of frequencies was chosen to meet FCC and IEEE standard requirements.	Output frequency is limited both by the bandwidth of the antenna, as well as the internal transceiver bandwidth limitations set by the chosen WiFi protocol. The stated range of values are given in the datasheet.
TX Pwr Typ: 13-19.5 dBm	• The transmitter power is a determining factor in the maximum range which at which the transmitter can send data to the receiver without data loss. The value range is specified by FCC and IEEE standard requirements	The ESP32 applies a spectral mask to its output as well as filtering to conform to the 802.11 WiFi standard and requirements for output power. The range of values is due to varying peak TX power based on WiFi protocol and bandwidth. The stated range of values are given in the datasheet.

RX Sensitivity Typ: -89 dBm	• RX sensitivity defines the minimum received power required to maintain data integrity. The maximum RX sensitivity (lowest value) dictates the maximum range at which the transmitter can be spaced from the receiver while still sending uncorrupted data	The minimum receiver sensitivity is specified in the datasheet, and is primarily a product of the transceiver front end low noise amplifier noise figure (NF) as well as the NF of following components. The system sensitivity will meet the specified value for all expected use cases since the measured sensitivity values are assuming impairments to the system which will not be present under expected worst case for this device.

SRVR_ANT_RF: Input

Interface Property	Value Justification	System Performance Justification
WiFi Protocols: 802.11b/g/n	• The WiFi protocol defines what types of routers and devices the system can inteface with	These protocols are the only ones which are supported by the ESP32 WROOM onboard RF Transceiver.
Carrier Freq: 2412-2484 MHz	• The carrier frequency of the transmitted data is important for many parameters, like the range of the system. The center frequency is also one of the most important parameters for antenna design. The range of frequencies was chosen to meet FCC and IEEE standard requirements.	Output frequency is limited both by the bandwidth of the antenna, as well as the internal transceiver bandwidth limitations set by the chosen WiFi protocol. The stated range of values are given in the datasheet.
RX Sensitivity Typ: -89 dBm	• The transmitter power is a determining factor in the maximum range which at which the transmitter can send data to the receiver without data loss. The value range is specified by FCC and IEEE standard requirements	The ESP32 applies a spectral mask to its output as well as filtering to conform to the 802.11 WiFi standard and requirements for output power. The range of values is due to varying peak TX power based on WiFi protocol and bandwidth. The stated range of values are given in the datasheet.

7 Engineering Requirements

The following engineering requirements were to be met, with the implementation following the requirement:

The system must be battery operated The system operates on a 9v battery.

The system must sense distances from 0.1-1.2m with a margin of error no greater than $\pm 10\%$ The datasheet of the DIST [[4], p. 23] reports a range of up to 2 meters, with an accuracy of 4%.

The system must display data measured by the distance sensor to the user The system displays the data to the user over Wi-Fi, VIA a web server running on the MCU.

8 Verification Process

The following procedure is to be followed to test the operation of the system:

- 1. Connect a 9V battery to the DC voltage regulator
- 2. Verify that the MCU has powered on
- 3. Connect MCU to web server
- 4. Confirm that the web server is pulling range data from the MCU
- 5. In a clear area, place markings on the floor in a straight line from 0 to 1.5m with 10cm spacing
- 6. Place the system at the 0 range mark with distance sensor pointing downrange
- 7. Starting at a range of 10 cm, place a high reflectance target (ex. soda can, water bottle, etc.) infront of the distance sensor
- 8. Wait for the measured range on the distance sensor to stabalize (1-5sec)
- 9. If range does not stabilize, increase number of DSP filter taps
- 10. Note range measured by the distance sensor and the actual range of the target
- 11. Check if the measured range is within 10% of the actual target range
- $R_{error}\% = 100 * |\frac{R_{measured} - R_{actual}}{R_{actual}}|$
- 12. Increase range by 10 cm and repeat from step 8 until the target range reached 1.2 m

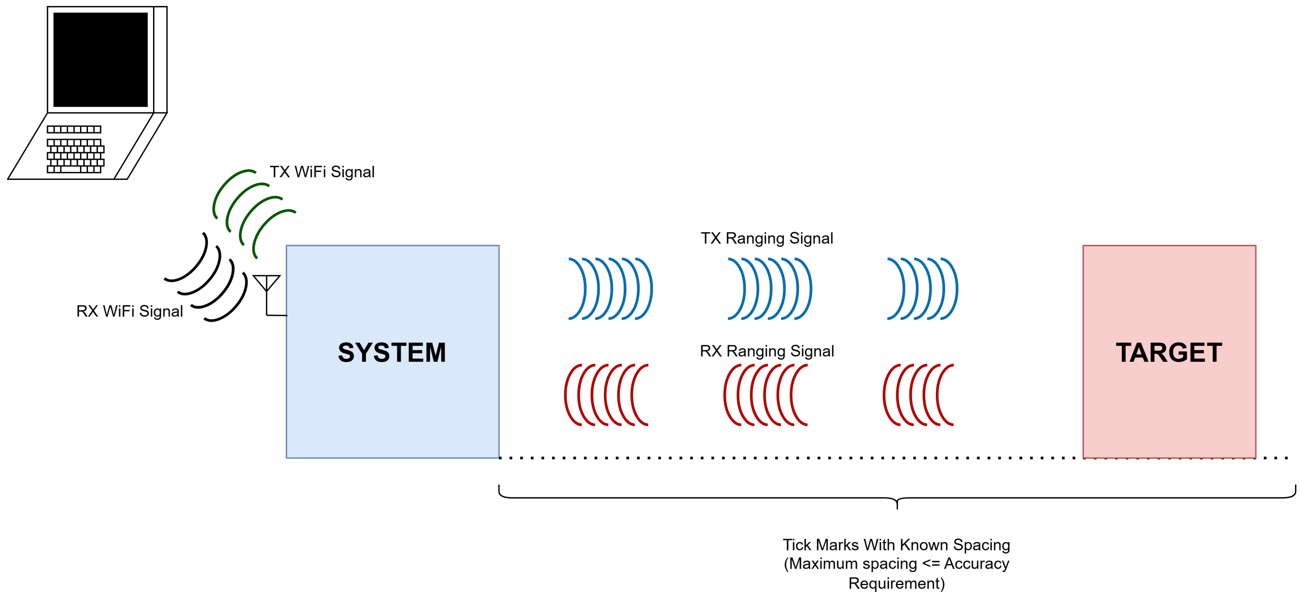

Figure 25: Diagram of Test Range Configuration

List of Figures

1 System level block diagram for the portable sensor 2
2 System Architecture 3
3 Voltage Regulator Diagram 4
4 Voltage Regulator Schematic 7
5 Voltage Regulator PCB Layout 8
6 Voltage Regulator 3D Model 9
7 MCU Diagram 10
8 Top Level DSP Diagram 14
9 DSP Level Signal Flow Diagram 14
10 MovingAverage UML Class Diagram 17
11 Example of Applying DSP to A Test Signal 17
12 I2C Top Level Block Diagram 18
13 WIFI TRX Block Diagram 20
14 Distance Sensor Block Diagram 25
15 System Measured vs Real Range and Error (%) vs Real Range 29
16 Antenna Top Level Block Diagram 30
17 Top Down Model of Antenna 30
18 Antenna Simulated S11 33
19 Antenna Gain Pattern At 2.4GHz From +Z 33

20 Antenna Gain Pattern At 2.4GHz From -Z 34
21 Antenna Gain Pattern At 2.4GHz From +Y 34
22 Antenna Gain Pattern At 2.4GHz From -Y 35
23 Main UML Class Diagram 39
24 Networking Interface UML Class Diagram 40
25 Diagram of Test Range Configuration I

Nomenclature

ANT	Antenna
DIST	Distance Sensor
DSP	Digital Signal Processing
GUI	Graphic User Interface
I2C	Inter-Integrated Circuit Communication Protocol
LDO	Low Dropout (Voltage Regulator)
MCU	Micro-Controller Unit
PCB	Printed Circuit Board
REG	Voltage Regulator
RF	Radio Frequency

References

[1] *A Low-Dropout, Positive, Fixed, and Adjustable Voltage Regulators*, onsemi, 2021, rev. 31. [Online]. Available: https://www.onsemi.com/pdf/datasheet/ncp1117-d.pdf

[2] *ESP32 Series Datasheet*, Espressif Systems, 2023, version 4.8. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[3] *ESP32-WROOM-32*, Espressif Systems, 2018, rev. 2.4. [Online]. Available: https://www.mouser.com/datasheet/2/891/esp-wroom-32_datasheet_en-1223836.pdf?srsltid=AfmBOorngC4GQVnhz8CPW1UjiclwHT-u3-iV7GzNcMGrov7LBgsieAdp

[4] *Time-of-Flight ranging sensor*, STMicroelectronics, 2024, rev. 6. [Online]. Available: https://www.st.com/resource/en/datasheet/vl53l0x.pdf

[5] "Adafruit vl53l0x," 2025. [Online]. Available: https://github.com/adafruit/Adafruit_VL53L0X?tab=readme-ov-file

[6] "Asynchronous http and websocket server library for esp32, esp8266 and rp2040," 2025, v3.7.2. [Online]. Available: https://github.com/ESP32Async/ESPAsyncWebServer/wiki

9 Appendix A