Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4317: Senior Design II
Spring 2020

Dream Team
Autonomous Mower

Ulysses Aguilar
Phu Nguyen
Alex Ho
Jerry Olds
Aderinsola Oladaiye

Revision History

Revision	Date	Author(s)	Description
0.1	11.08.2019	UA, JO, AH, PN, OA	document creation
0.2	11.10.2019	UA, JO, AH, PN, OA	complete draft
0.9	11.11.2019	UA, JO, AH, PN, OA	official release 1
1.0	5.15.2020	UA, JO, AH, PN, OA	official release 2

Contents

List of Figures

List of Tables

1 INTRODUCTION

This section provides the reader with an overview of our autonomous mower. The features of the system, user interface, and primary operations of the system will be listed below.

1.1 FEATURES & FUNCTIONS

The two main operations of the mower are the autonomous mode, and the remote control mode. The user will have ability to choose what mode to run the mower with a client side computer. The user will be able to stop or pause the system at any time. In autonomous mode the system will be guided by GPS, and will determine a path using GPS boundaries. Once on the path, it will use cameras and Lidar sensors to detect obstacles and act accordingly. Updates for the system will only happen with major upgrade to the functionality.

1.2 EXTERNAL INPUTS & OUTPUTS

Name	Description	Use
On Board Server	Inputs and Outputs	The main control system of the mower
Client Computer	External Input and Output	Sends command to mower, and receives data from mower.
Remote Control	External Output	Sends user command to client which then get sent to mower.
VR Headset	External Output	Receives live feed from client computer which receives feed from the on board server through WIFI.
GPS	External Input	Used by mower to keep track of location
LIDAR	External Input	Produces distance measurements from surrounded object.
Radio Transmitter Receiver	External Input and Output	Sends correction data from the base GPS to the GPS on the vehicle.
Magnetometer	External Input	Produces angular displacement between the goal and the current orientation.

Table 2: Inputs and Outputs

1.3 PRODUCT INTERFACES



2 System Overview

Below you will find a simple diagram of our system architecture with detailed descriptions for each layer.

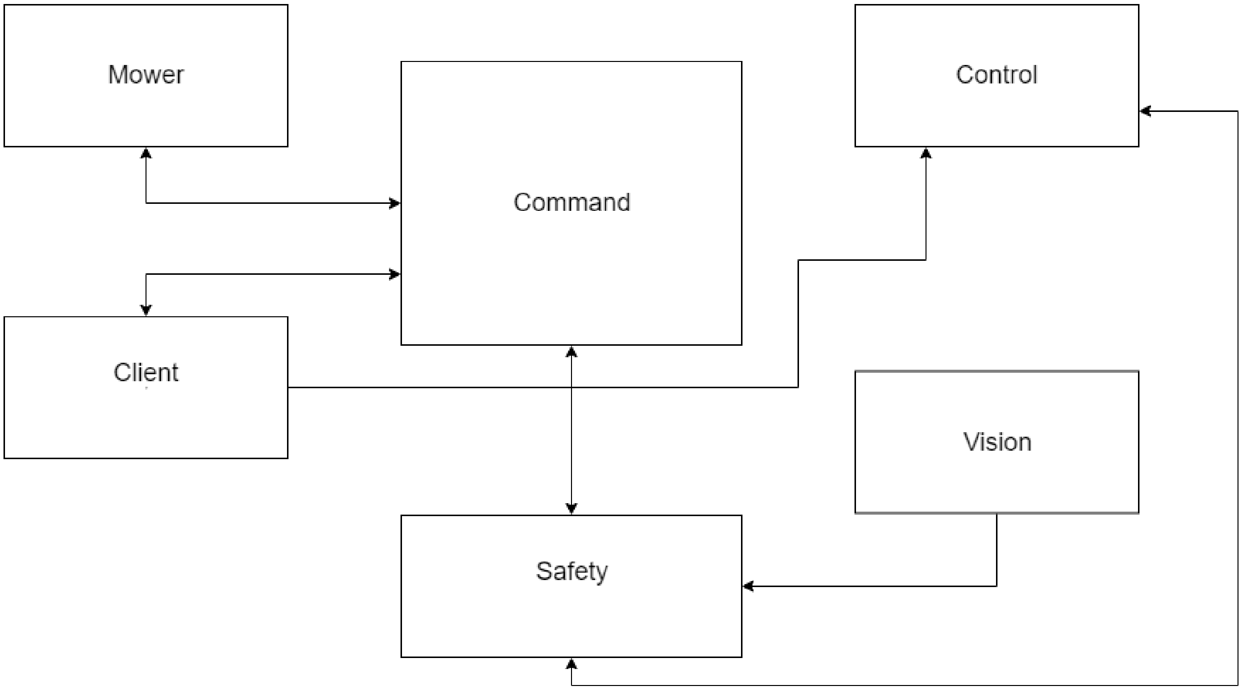


Figure 1: A simple architectural layer diagram

2.1 Command Layer Description

The Command layer is responsible for taking inputs from the other layers and creating outputs based on the inputs given. The command layer communicates with the client layer, safety layer, and the Mower layer. The client and command layers send connectivity messages to each other. The safety layer first checks if a command from the control layer is safe to execute. If the command is safe to execute, the safety layer will send the command to the Command layer. The Command layer will then send the command to an Arduino in the Mower layer. Commands that the Command layer will send to the Arduino will include powering the system on/off, movement of the system, and powering the blade on/off. An on-board server will be responsible for processing the inputs and sending the corresponding messages to the Arduino. The server will be using various ROS packages to process the different types of inputs and it will be connected to WiFi to communicate with the client machine.

2.2 Client Layer Description

This layer is mainly responsible for interfacing with the command layer to process data gotten from User Input from an Xbox controller and VR output from the VR headset all with the help from a simple GUI to keep track of the values gotten. The Client layer is also the computer we are going to use to help with the overall interfacing. The client can be set to run either one of two modes from the control layer which are the Autonomous or Manual mode .

2.3 Control Layer Description

The Control layer is responsible for taking in data from the client layer to select between manual and autonomous mode. In the autonomous mode, The navigation, the sensor, and the path finding are responsible to calculate the critical to void obstacles and coordinate its position. In manual mode, the only data will be getting is the data from the Xbox controller from the client. But the controls will not be implemented if they failed by the safety layer.

2.4 Safety Layer Description

The Safety layer is responsible for doing a series of safety checks for the system. These safety checks include: obstacle detection, connection loss, GPS boundary, and speed limits. When a command comes in it is checked for safety, and if it is declared okay it will be sent to command Interface which is responsible for sending commands to the mower. The safety layer has an interface that communicates with all other layers this is needed to be able to detect failures and safety issues. The safety layer has subsystems GPS, Obstacle avoidance, geo-fence system, and an interface that connects to vision layer, control, and command to make sure it has access to all sensors, cameras, and layers. Our system will use ROS, so our safety layer will have a node that is subscribed, and publishes to all nodes that are being used to check for issues.

2.5 Vision Layer Description

The vision layer is responsible for collecting 360 Camera and LIDAR data. It will take LIDAR data points, map it to a 3D area, and push it to the Safety Layer in a readable format. It will also be responsible for pushing a 360 camera video feed.

2.6 Mower Layer Description

The mower layer is responsible for taking in commands for the mower and switching functionalities of the mower on and off. The arduino sublayer is the interface that takes in signals and commands for the command layer which holds the on board server. The arduino will also be able to send out signals to the command layer. The arduino is responible for keeping track of the status of other subsystems in the mower layer being the motor controller, blades and power.

3 Subsystem Definitions & Data Flow

This section has added detail of subsections for each layer and how they interact with each other.

(4).png

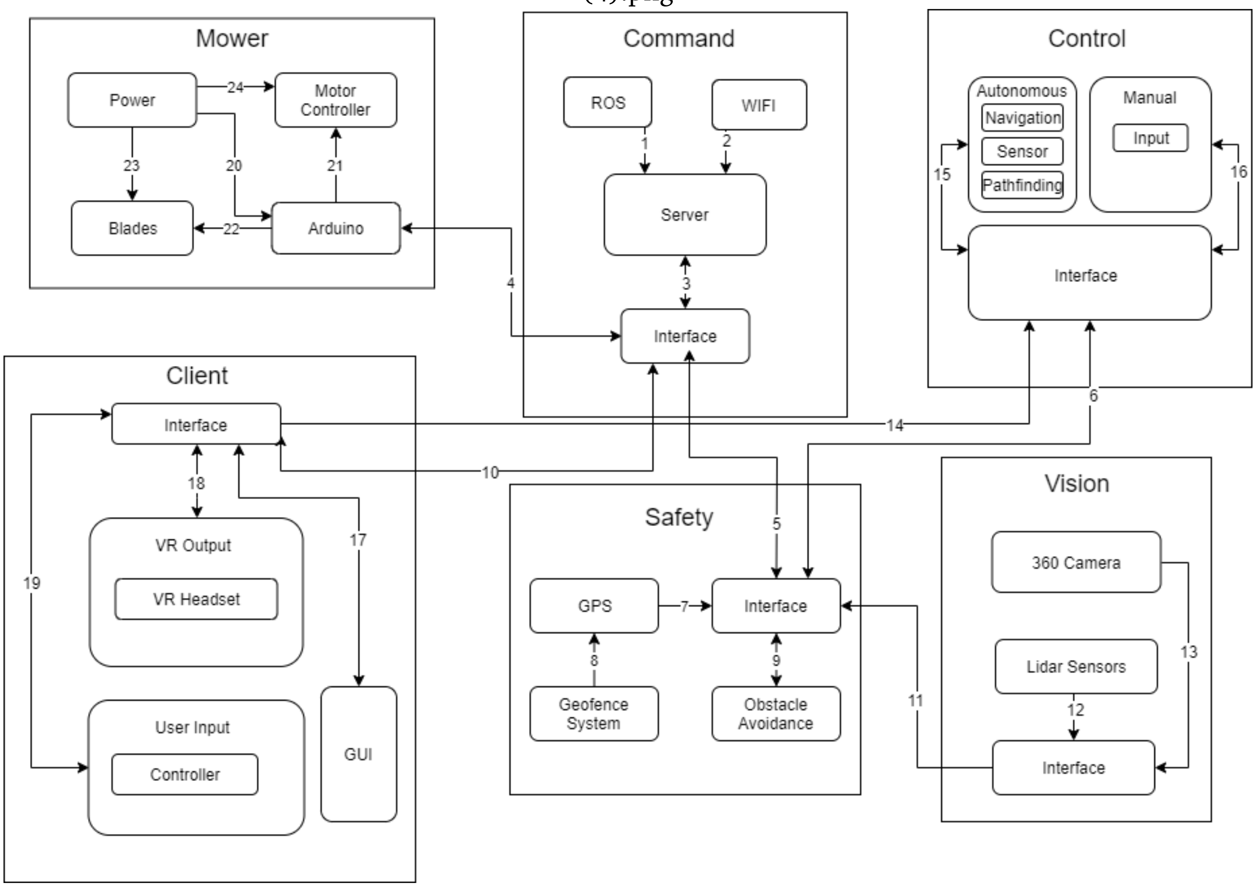


Figure 2: A simple data flow diagram

4 Command Layer Subsystems

Command layer subsystems include a Server, ROS, WiFi, and an interface for interaction with other layers. The Server will be responsible for processing inputs and generating outputs. ROS packages will be loaded on to the server to handle the different functionalities of the system. The server will also be connected to WiFi so that it can communicate with the client machine. The server will interact with an interface to receive inputs and send outputs to other layers.

4.1 Server

An on-board server will be responsible for processing the inputs and sending corresponding outputs to the different layers. The server that we will be using is the LCP-862.

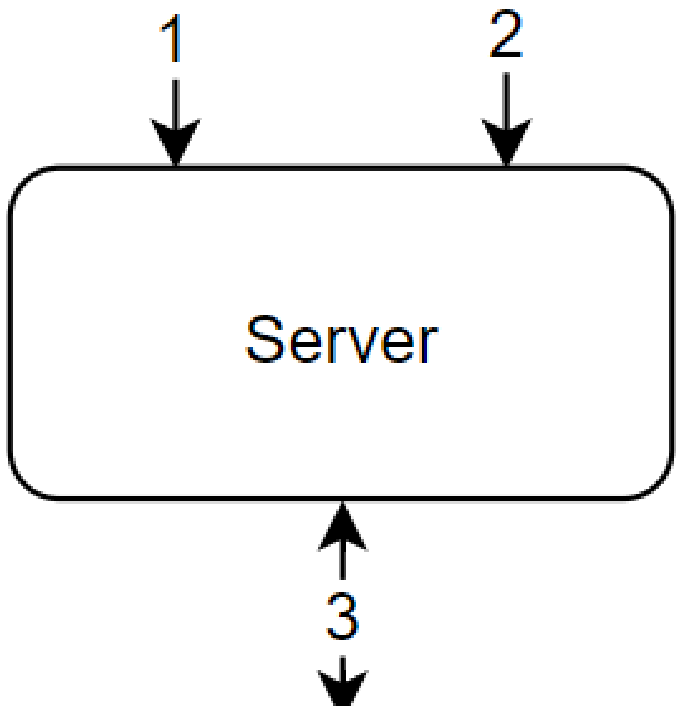


Figure 3: Command subsytem description diagram

4.1.1 Assumptions

- The server will operate under the assumption that there will be a constant connection to WiFi.
- The server will operate under the assumption that it will always have accurate GPS coordinates
- The server will operate under the assumption that it will be receiving accurate Lidar readings.

4.1.2 Responsibilities

The Server will be responsible for processing movement of the vehicle and it's other functions. The server will also be responsible for processing inputs from GPS, Lidar, and camera to find paths and avoid objects. Based on the inputs from the sensors and commands given from the client computer, the server will send messages to the Arduino to preform the functions of the lawn mower.

4.1.3 Subsystem Interfaces

Table 3: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	ROS packages for handling functionalities of system	input 1	N/A
#2	WiFi connection for communication with client machine	input 2	N/A
#3	Interface for accepting inputs and sending outputs to other layers	input 3	output 3

4.2 ROS

ROS packages will be loaded on the server to handle functionalities like control, mapping, and navigation.

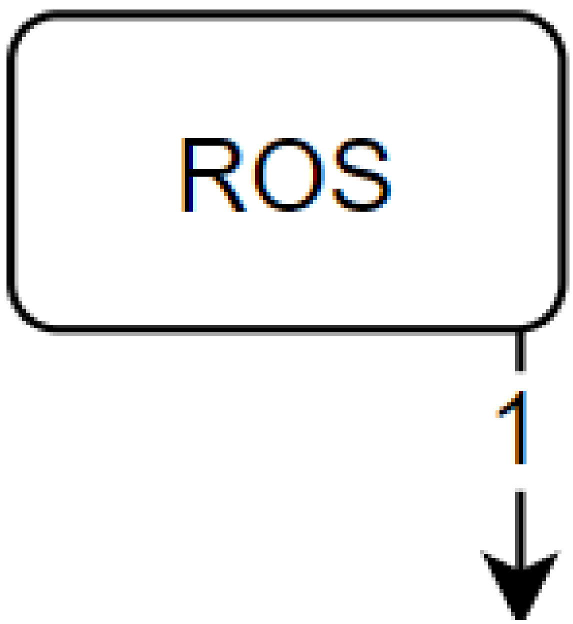


Figure 4: Command subsytem description diagram

4.2.1 Assumptions

- ROS packages will be loaded onto the server under the assumption that the server is running the Ubuntu 18.04 operating system

4.2.2 Responsibilities

- Handling control functionalities
- Simultaneous localization and mapping

- Autonomous navigation
- Obstacle detection and avoidance

4.2.3 Subsystem Interfaces

Table 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	ROS packages for handling functionalities of system	N/A	output 1

4.3 WiFi

WiFi connection is used for communication between client machine and on-board server.

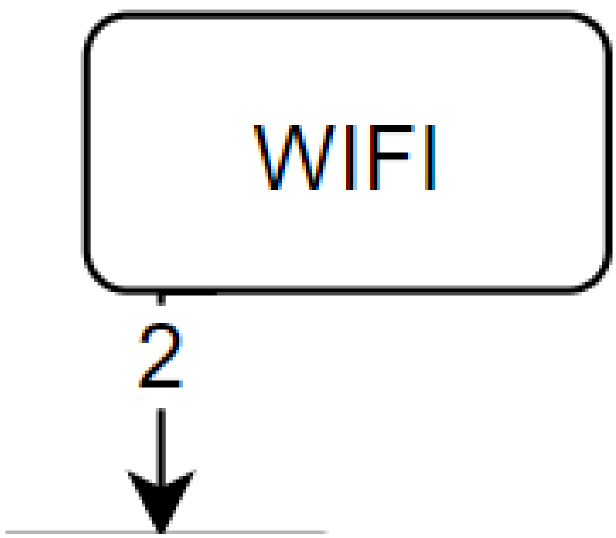


Figure 5: Command subsytem description diagram

4.3.1 Assumptions

- The system will operate under the assumption that it will have constant connection to WiFi in the area of operation.

4.3.2 Responsibilities

- Transport medium used for sending commands from client machine to on-board server.
- Transport medium used for sending video from camera on system to client machine.

4.3.3 Subsystem Interfaces

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#2	WiFi connection used for communication between client machine and on-board server	N/A	output 2

4.4 Interface

Interface used for accepting inputs and sending outputs to other layers of the system.

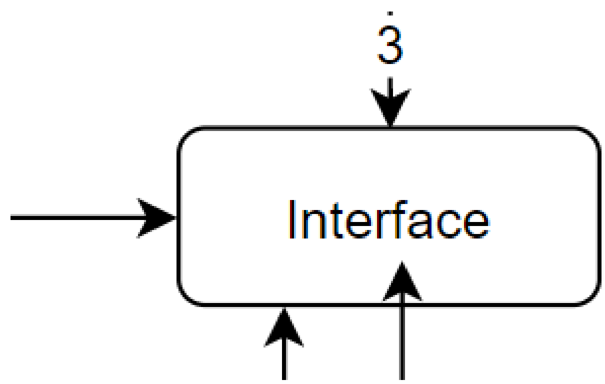


Figure 6: Command subsystem description diagram

4.4.1 Assumptions

-

4.4.2 Responsibilities

- Responsible for accepting commands from the safety layer and sending status messages back
- Responsible for sending messages to the client machine and receiving messages from the client machine.
- Responsible for sending commands to the Arduino to execute commands and receiving messages from the Arduino on what commands were executed.

4.4.3 Subsystem Interfaces

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#4	Sending commands to the Arduino to execute commands and receiving messages messages from Arduino on what commands were executed	input 4	output 4
#5	Accepting commands from safety layer and sending status messages	input 5	output 5
#10	Sending and receiving messages to and from the client machine	input 10	output 10

5 Client Layer Subsystems

This layer features the following subsytems and they include: the Graphic User Interface, User Input , the Virtual reality output and the Client interface.

5.1 User Input

The Controller is the main apparatus we are using for user input it will only be able to interact with the Client Interface.

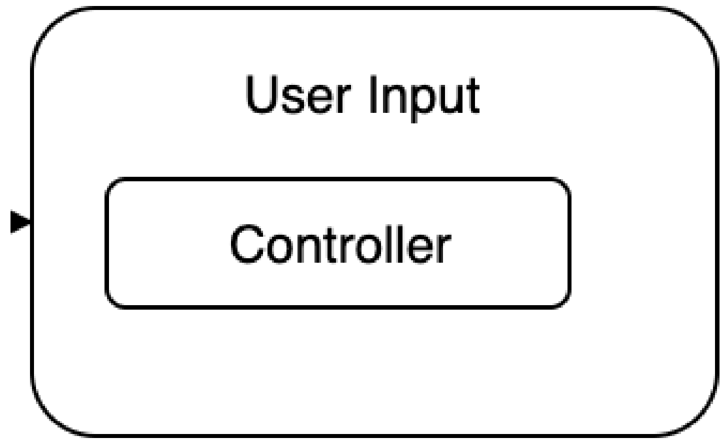


Figure 7: Example subsystem description diagram

5.1.1 Assumptions

The xbox controller will serve as our controller. User input subsystem will be unnecessary if in the control layer it is set to run the system in autonomous mode.

5.1.2 Responsibilities

The Controller is mainly responsible in taking input from the medium used for taking input, in our case it will be a xbox controller. The Controller also sends and receives input and information to and from the Client Interface.

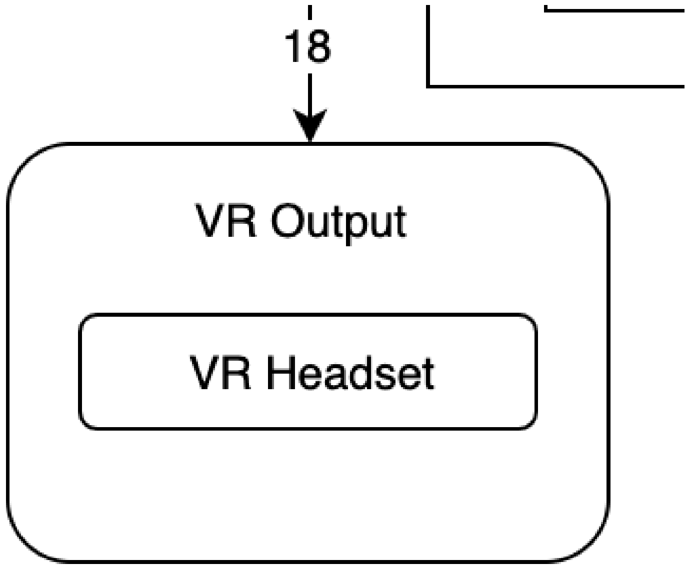
5.1.3 Subsystem Interfaces

Table 7: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Interacts with the main Client interface to send and receive input information	input 16	output 16

5.2 VR Output

This will be the main output of the Client layer because the data gotten from the Client interface will display those values as a video stream with help of a VR headset



5.2.1 Assumptions

The VR Output will rely on the Client interface to get the data from the Vision layer to display the video stream. The Video stream will be 360 degree view.

5.2.2 Responsibilities

VR Output will acquire data from the Client interface and feed it to the VR headset. It will also send feedback loops to the Client interface to understand what is happening in the VR output. It will make the VR headset to output a video stream after receiving sufficient input.

5.2.3 Subsystem Interfaces

Table 8: Subsystem interfaces

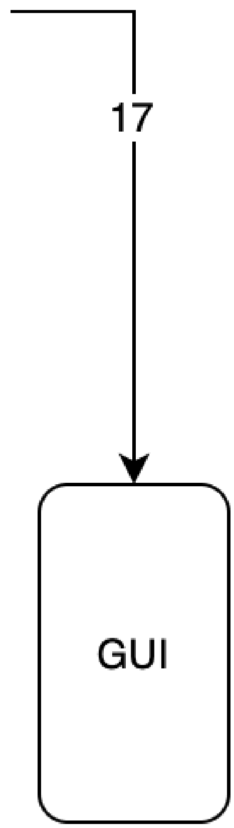
ID	Description	Inputs	Outputs
#1	Back and forth interaction with client interface to get sufficient data to be converted into video stream in the VR headset	input 18	output 18

5.3 GUI

The GUI is the main front end of this layer if not the entire system.

5.3.1 Assumptions

No assumptions



5.3.2 Responsibilities

The GUI provides a user interface for the Client interface to interact with.

5.3.3 Subsystem Interfaces

Table 9: Subsystem interfaces

ID	Description	Inputs	Outputs
#xx	Displays a user interface with help from interacting with the client interface	input 17	output 17

5.4 Client Interface

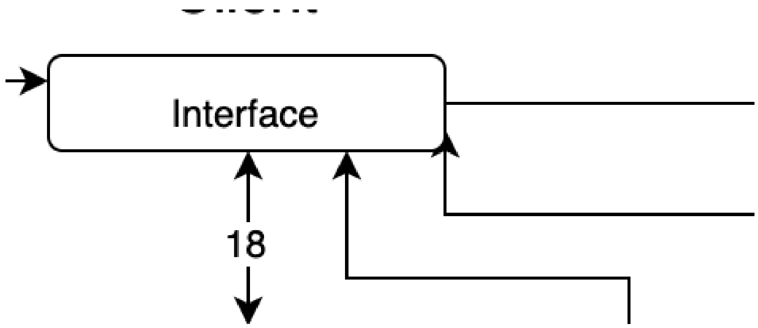
This is the main interface of the client layer that interfaces with most of the other layers of the architecture design.

5.4.1 Assumptions

No assumptions

5.4.2 Responsibilities

It will interact with all subsytems in the client layer and the other interfaces of other layers



5.4.3 SUBSYSTEM INTERFACES

Table 10: Subsystem interfaces

ID	Description	Inputs	Outputs
#xx	Interacts with other layers and sub sytems	input 10 input 17 input 18 input 19	output 10 output 14 output 17 output 18 output 19

6 Control Layer Subsystems

The control layer subsystems include a Autonomous, Manual, and an interface for interaction with other layers. The user will be able to select between the autonomous and manual mode. The selections are provided in the graphical user interface from the client computer.

6.1 Autonomous subsystem

The autonomous subsystem is responsible to store the position of the obstacles on the field after sensing them, calculate the path of the lawnmower based on the obstacles position, and navigate through the field.

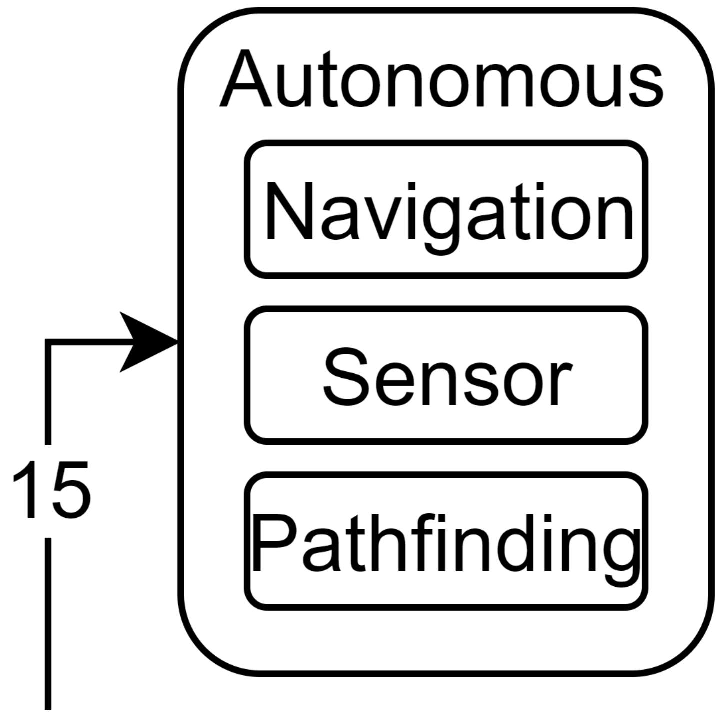


Figure 8: Autonomous subsystem description diagram

6.1.1 Assumptions

- The autonomous mode can be selected by the user.
- The autonomous mode has database to store position of obstacle along with a pre-calculated path.
- The control of autonomous mode is passed by the safety layer.

6.1.2 Responsibilities

- The autonomous mode has a two-way connection toward the interface subsystem, which is sending navigating data to other layers to issue commands control the lawnmower or to check on the safety test of the safety layer.
- The autonomous mode can calculate its path base on the obstacles presenting on the field.

6.1.3 Subsystem Interfaces

Table 11: Subsystem interfaces

ID	Description	Inputs	Outputs
#15	Connection between autonomous subsystem and control interface subsystem	input 15	output 15

6.2 Manual subsystem

The manual subsystem is responsible for the manual control ability of the lawnmower.

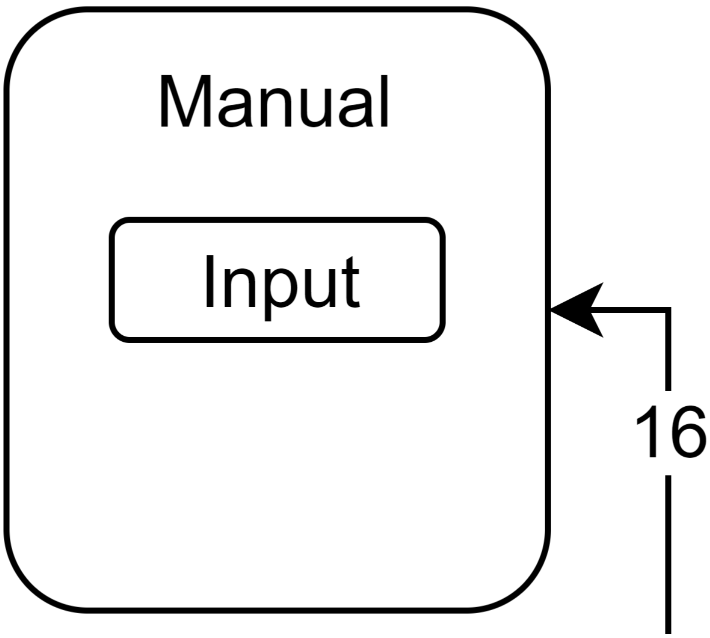


Figure 9: Manual subsystem description diagram

6.2.1 Assumptions

- The manual mode can be selected by the user.
- The control of manual mode is passed by the safety layer.

6.2.2 Responsibilities

- The autonomous mode has a two-way connection toward the interface subsystem, which is sending control data from the user to other layers to issue commands to control the lawnmower or to check on the safety test of the safety layer.

6.2.3 Subsystem Interfaces

Table 12: Subsystem interfaces

ID	Description	Inputs	Outputs
#16	Connection between manual subsystem and control interface subsystem	input 16	output 16

6.3 Control interface subsystem

The interface of the control layer to communicate with other layers of the system.

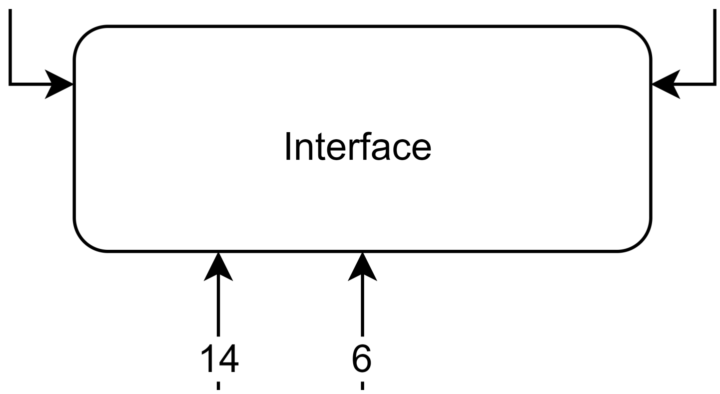


Figure 10: Control interface subsystem description diagram

6.3.1 Assumptions

- The data through put and latency are adequate to handle all the data coming in or leaving the bus.

6.3.2 Responsibilities

- Receive the raw data from the client interface to control the lawnmower.
- Send data to the safety layer for approvals on autonomous and manual control, and receive response to the approval requests.
- Receive data from the autonomous subsystem to perform checks, and send check-result to autonomous subsystem.
- Receive data from the manual subsystem to perform checks, and send check-result to manual subsystem.

6.3.3 Subsystem Interfaces

Table 13: Subsystem interfaces

ID	Description	Inputs	Outputs
#14	Connection between client interface subsystem and control interface subsystem	input 14	N/A
#06	connection between safety interface subsystem and control interface subsystem	input 06	output 06
#15	Connection between autonomous subsystem and control interface subsystem	input 15	output 15
#16	Connection between manual subsystem and control interface subsystem	input 16	output 16

7 Safety Layer Subsystems

The following subsystems where picked for the safety layer to make it easier to make safety checks for our system.

7.1 GPS

The GPS subsystem is used to keep track of our position, orientation, and speed. It receives information from the geofence system to be able to determine if our location is allowed. The GPS subsystem will also tell us if our GPS connection is lost. If any of the data measurements from the GPS are out of our safety range flags will be set and sent to the interface subsystem as needed. The GPS data is also used by the control layer to navigate, so orientation, speed and position are sent to the interface as well.

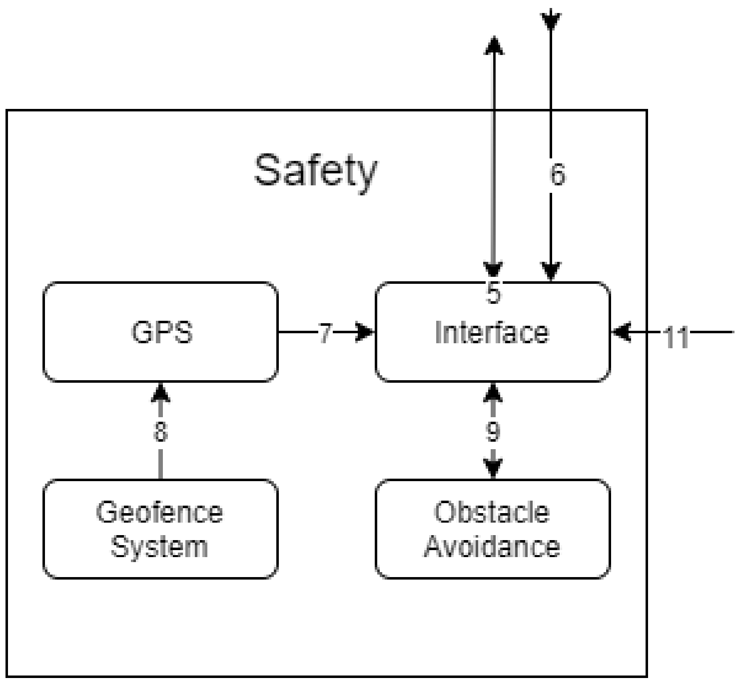


Figure 11: safety subsystem description diagram

7.1.1 Assumptions

It is assumed that any Layer or subsystem that needs GPS data will be linked to the safety layers interface to receive GPS information or safety checks.

7.1.2 Responsibilities

The GPS subsystem is responsible for delivering position, speed, orientation, and GPS connection data to the interface. It is also responsible to for setting off flags if our goefence is close to being crossed. It will also set off flags if any of our safety ranges for position, orientation and speed are passed.

7.1.3 GPS Interfaces

Table 14: GPS interfaces

ID	Description	Inputs	Outputs
#8	Receives position of geofence	input 8	N/A
#7	Sends GPS Data, and GPS safety checks, and receives safety check request	input 7	output 7

7.2 Geofence system

The geofence system is setup here and constantly sends geofence position to GPS subsystem.

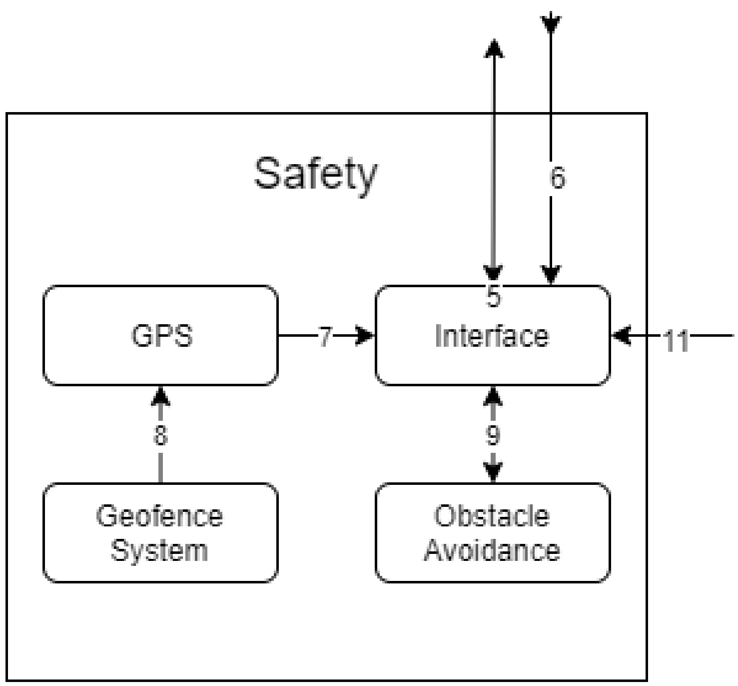


Figure 12: safety subsystem description diagram

7.2.1 Assumptions

It is assumed that the GPS subsystem will be able to accept position of geofence at a regular rate.

7.2.2 Responsibilities

The geofence subsystem is responsible for establishing a geofence and for sending the position to the GPS subsystem at a reasonable rate that the GPS node can handle.

7.2.3 Geofence Interfaces

Table 15: Geofence interfaces

ID	Description	Inputs	Outputs
#8	send position of geofence	N/A	output 8

7.3 Obstacle avoidance

The obstacle avoidance subsystem is responsible for detecting obstacles from camera, and LIDAR data received from the safety layer interface. If an obstacle is detected flags will be set and a protocol to avoid the obstacle will be sent to the interface which will then be sent to the command layer.

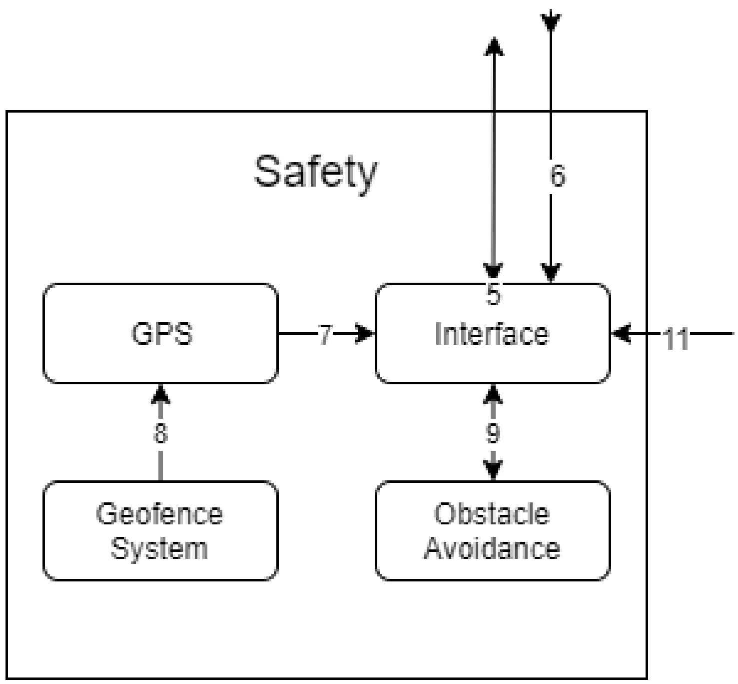


Figure 13: safety subsystem description diagram

7.3.1 Assumptions

It is assumed that this subsystem will have access to, the vision layer which contains camera, and LIDAR data. This subsystem will have access to position, and will then be able to send commands to the mower to avoid the obstacle and continue on the calculated path.

7.3.2 Responsibilities

The obstacle avoidance subsystem is responsible for detecting obstacle using data from the vision layer. If an obstacle is found the system will be able to create a command protocol to slow down the mower, avoid the obstacle, and allow the mower to continue on its set path.

7.3.3 Obstacle avoidance Interfaces

Table 16: Obstacle avoidance interfaces

ID	Description	Inputs	Outputs
#9	Receive data from vision layer through interface	input 9	N/A
#9	Receive GPS position data through interface	input 9	N/A
#9	Receive path from control layer through interface	input 9	N/A
#9	Send protocol to avoid obstacle	N/A	output 9

7.4 Interface

The Interface subsystem is used to route data from the safety layer, and data the safety layer need to function properly

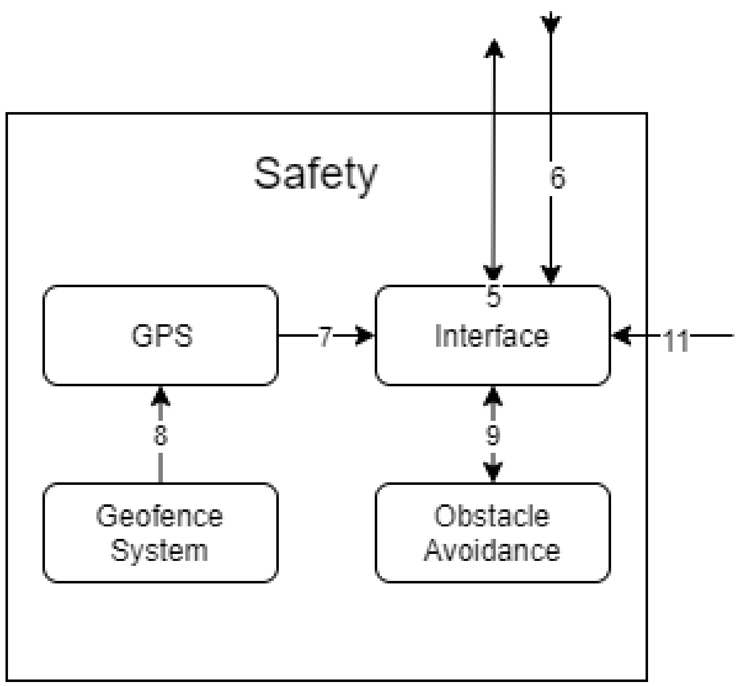


Figure 14: safety subsystem description diagram

7.4.1 Assumptions

It is assumed that any subsystem that is needed for safety, and any layer that needs safety information is connected in some way to this interface.

7.4.2 Responsibilities

The interface subsystem is responsible for properly sending and receiving data between the safety layer and all other layers that are connected. it is responsible for proper routing data, commands, and safety checks to the proper layers, and subsystems.

7.4.3 Interfaces

Table 17: Interfaces

ID	Description	Inputs	Outputs
#7	Receive data from GPS	input 7	N/A
#9	send GPS position data to obstacle avoidance subsystem	N/A	output 9
#11	Receive vision data	input 11	output 9
#6	receive path from control layer	input 6	N/A
#9	send avoidance protocol to command layer	N/A	output 9
#9	send vision data to obstacle node	N/A	output 9
#9	receive protocol to avoid obstacle	input 9	N/A
#6	send safety checks to control layer	N/A	output 6
#5	send safety checks to command layer	N/A	output 5
#5	Receive status updates from command	input 5	N/A

8 Vision Layer Subsystems

The visual subsystem is responsible for collecting, processing, and sending all visual data from the sensors and cameras.

8.1 360 Camera

The 360 Camera Subsystem is responsible for getting visual data from the 360 camera and passing it to the interface.

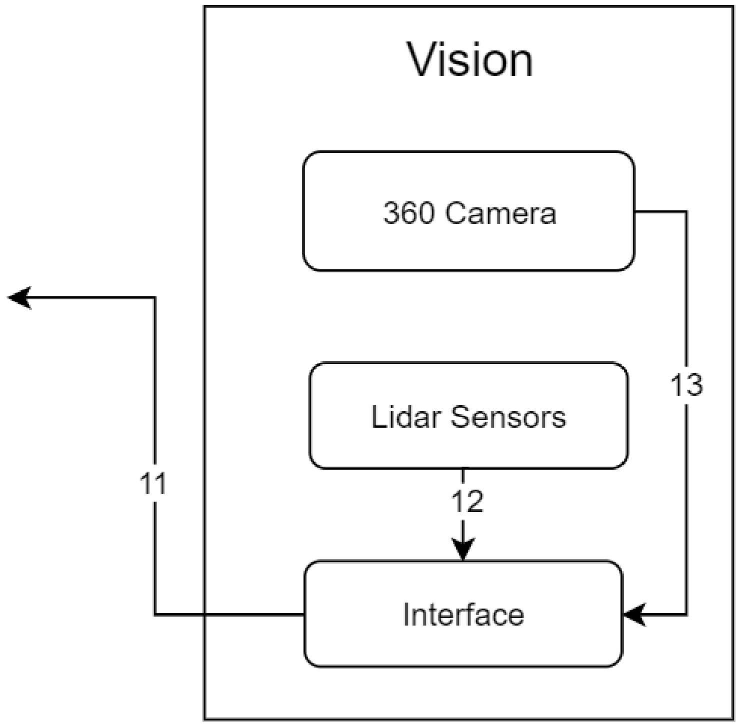


Figure 15: Vision Subsystem Description Diagram

8.1.1 Assumptions

The 360 Camera subsystem assumes the 360 Camera is connected and is able to output data as an RTSP video stream to the Vision Layer.

8.1.2 Responsibilities

The 360 Camera Subsystem is solely responsible for getting data from the 360 camera and passing it in a readable format.

8.1.3 Subsystem Interfaces

Table 18: Subsystem interfaces

ID	Description	Inputs	Outputs
#13	360 Camera	N/A	Interface

8.2 LIDAR Sensors

The LIDAR sensor subsystem is responsible for getting point cloud data from the LIDAR sensor, processing/mapping the data, and outputting it in a readable format to the interface.

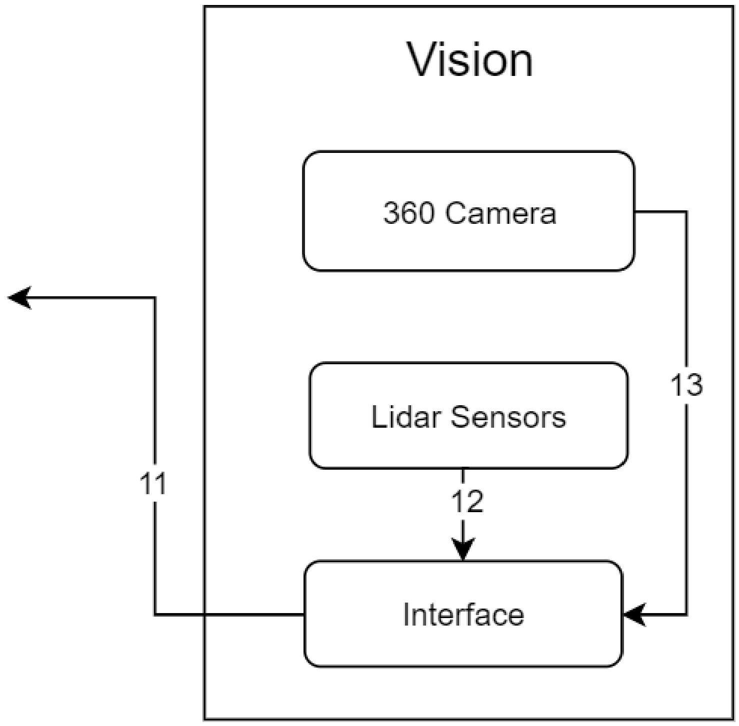


Figure 16: Vision Subsystem Description Diagram

8.2.1 Assumptions

The LIDAR sensor subsystem assumes the LIDAR sensors are connected and is able to output data to the Vision Layer.

8.2.2 Responsibilities

The LIDAR sensor subsystem is responsible for getting data from the LIDAR sensors and passing it in a readable format.

8.2.3 Subsystem Interfaces

Table 19: Subsystem interfaces

ID	Description	Inputs	Outputs
#12	LIDAR Sensors	N/A	Interface

8.3 Interface

The interface subsystem is responsible for processing 360 camera and LIDAR sensors and outputting it to other layers as needed.

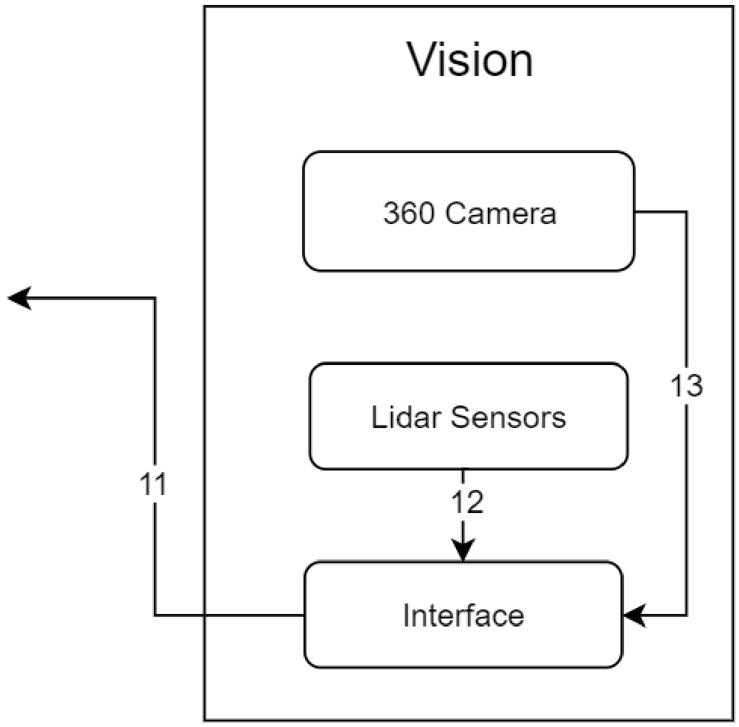


Figure 17: Vision Subsystem Description Diagram

8.3.1 Assumptions

The interface subsystem assumes the 360 camera and LIDAR sensors are connected and are feeding data readable data.

8.3.2 Responsibilities

The interface subsystem is responsible for passing data from the 360 Camera and LIDAR sensors to other layers.

8.3.3 Subsystem Interfaces

Table 20: Subsystem interfaces

ID	Description	Inputs	Outputs
#13	360 Camera data	input 13	N/A
#12	LIDAR Sensor data	input 12	N/A
#11	Interface data output	N/A	output 11

9 Mower Layer Subsystems

The mower layer is very simple it uses its subsystems to handle commands and sends status reports of the mower to let the system now what is on and what is off.

9.1 Power

Power is what we will use to provide power to our system along with battery level data.

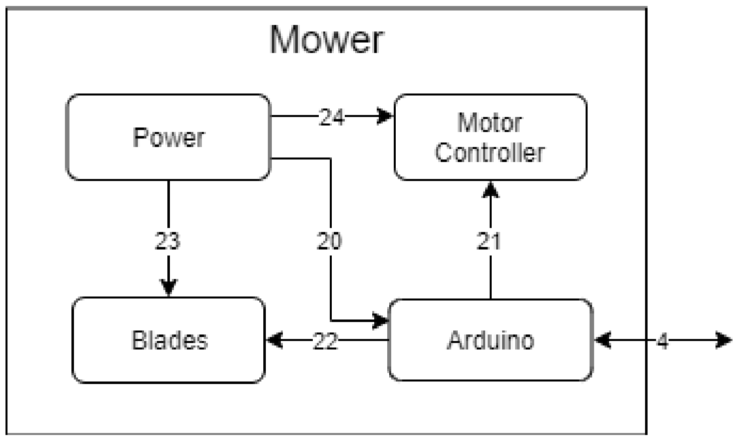


Figure 18: Mower layer description diagram

9.1.1 Assumptions

It is assumed that all power outputs levels are appropriate for the hardware.

9.1.2 Responsibilities

the Power subsystem is responsible for providing power to the mower layer, and sending battery charge levels to the arduino.

9.1.3 Power Interfaces

Table 21: Power interfaces

ID	Description	Inputs	Outputs
#24	sends power to motor controller	N/A	output 24
#23	Sends power to blades	N/A	output 23
#23	Sends power and charge levels to arduino	N/A	output 20

9.2 Blades Subsystem

The blades subsystem will keep track of the status of the blades, and will accept commands for turning the blade on and off.

9.2.1 Assumptions

it is assumed that the node is only responsible for turning the blades on and of not deciding whether the blades should be on or off.

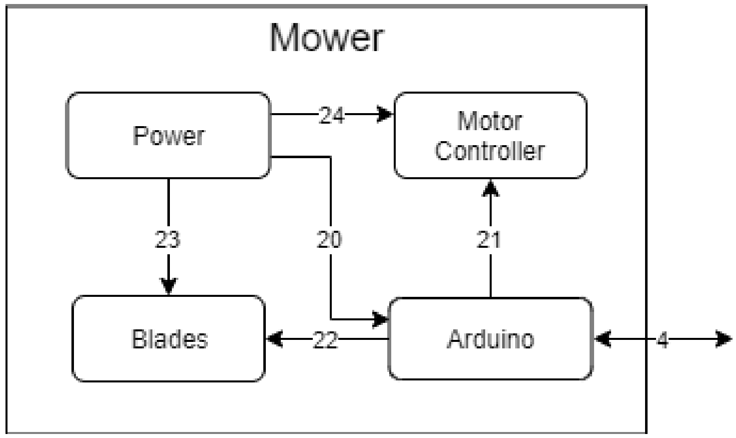


Figure 19: Mower layer description diagram

9.2.2 Responsibilities

the blades subsystem is responsible for keeping track if blades are on or off and turning the blades on and off given the command from the arduino.

9.2.3 Blade Interfaces

Table 22: Blade interfaces

ID	Description	Inputs	Outputs
#23	receives power to blades	input 23	N/A
#23	receives commands from arduino	input 22	N/A

9.3 Motor Controller

The motor controller subsystem will take in commands from the arduino and convert them into commands the motors can actually achieve our system is open loop so no feed back will be used.

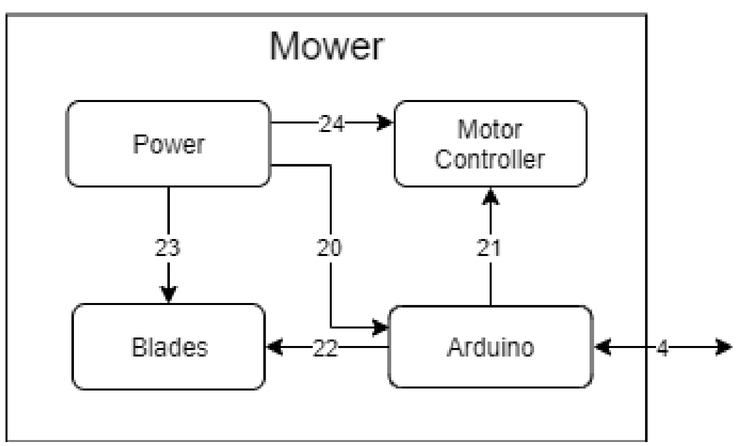


Figure 20: Mower layer description diagram

9.3.1 Assumptions

It is assumed the system is differential drive, and commands from the arduino are understandable from the arduino.

9.3.2 Responsibilities

The motor controller is responsible for providing appropriate power to the two motor given a certain command from the arduino.

9.3.3 Motor controller Interfaces

Table 23: Motor controller interfaces

ID	Description	Inputs	Outputs
#24	Receives power to motor controller	input 24	N/A
#21	Receives commands from arduino	input 21	N/A

9.4 Arduino

The arduino subsystem acts as our interface into the mower layer. It will keep track of our mowers status, receive commands, and send commands withing the mower layer.

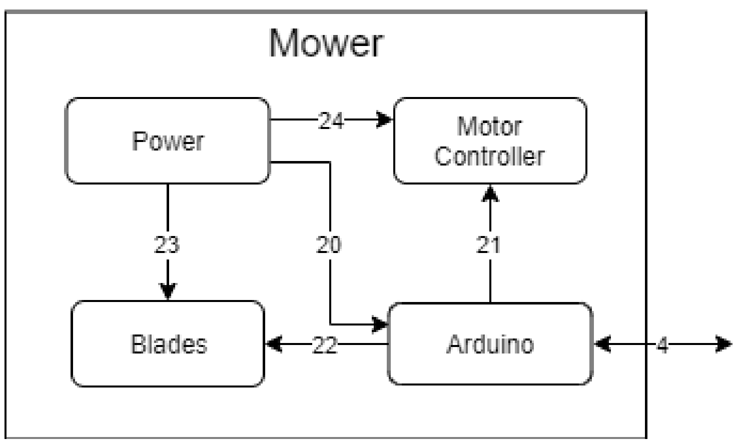


Figure 21: Mower layer description diagram

9.4.1 Assumptions

It is assumed that the arduino is setup to have communication with the command layer and to each subsystem within the mower layer.

9.4.2 Responsibilities

the arduino subsystem is responsible for keeping track of the status of all functionallity of the mower, and interfacing commands from the command layer to the mower.

9.4.3 Arduino Interfaces

Table 24: Arduino Interfaces

ID	Description	Inputs	Outputs
#21	sends commands to motor controller	N/A	output 21
#22	Sends toggle commands to blades	N/A	output 22
#20	Receives power and charge levels	input 20	N/A
#4	Receives commands from command layer	input 4	N/A
#4	Sends status updates on mower to command layer	N/A	output 4

REFERENCES