
ENGINEERING
TEXAS A&M UNIVERSITY
Autonomous Lawnmower
Max Lesser, Vincent McMasters,
Josh Samaniego, Jonathan Poulose
ECEN 404-904 Team 30

Project Summary

Problem Statement: *“You will start with a lawn mower shell, add motors to propel the wheels, microcontroller to control everything, comms to a wifi network where area to be covered and route will be entered, and a power mechanism (docking station or other)”*

- Use lawn mower body and reconfigure for our needs
  - Electric motors for wheel and blade
  - Solar and grid tied charging
  - Docking station
  - Navigate autonomously using various sensors
  - Receive user information about area to mow and scheduling through app

Problems and Solutions

ENGINEERING
TEXAS A&M UNIVERSITY

Problems (looking towards interfacing):

- Front Mower wheels 0-degree turn
- Rear Mower wheels
  - Geared connection/transmission
  - Solid axle
- Blade drive attachment
  - How to mount blade motor
- MCU interface
- US sensors have issues detecting object with fabric
  - Readings for objects covered in fabric (f.e. person wearing sweatshirt) extremely inaccurate and unreliable

Solutions:

- Caster wheels in the front
  - Allows 0-degree turn
- Several Options for rear
  - Cut axel in 2 parts and mount motors to that, adjust for gearing ratio in code
  - Manufacture shaft and mount directly to wheels and motors for 1:1 ratio. (Need to ensure sufficient torque, but ratio stays the same, possibly easier motor mounting because the old shaft won't be in the way)
- Several Options for Blade also
  - Disassemble motor and reuse its driveshaft
  - Manufacture our own driveshaft
- MCU interface
  - Need to finalize format
- Person detection
  - Include Bump/contact sensor in front
  - More extensive tests with full array, may be possible to determine fabric with multiple sensors.

MCU Subsystem (Obstacle detection)

ENGINEERING
TEXAS A&M UNIVERSITY

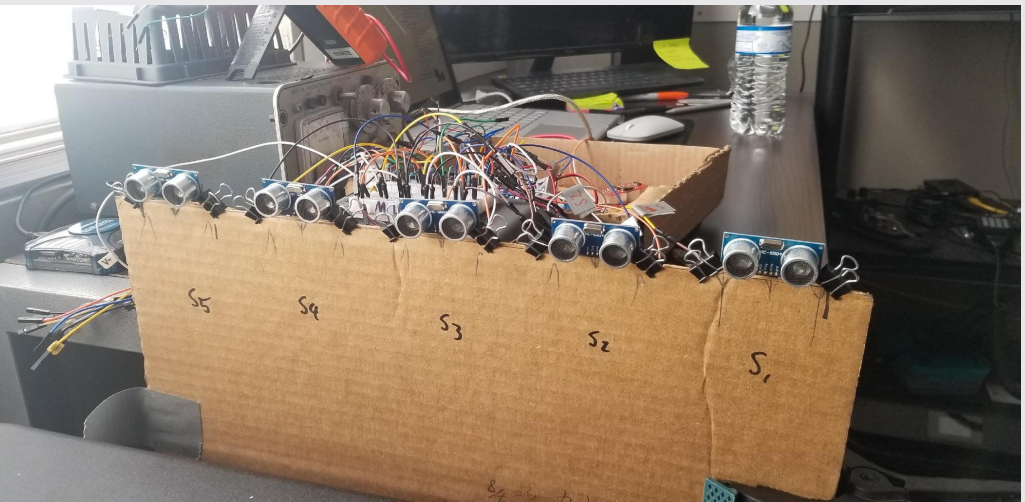
Object: Cardboard Box (11.4x13.4x31.75) cm (WxDxH) presenting 11.4x31.75cm side

- Test 1: approx. 70deg @ 40.6cm (resolved by 2 sensors)

distance1	float	0x8007FF60	289.93024
distance2	float	0x8007FF64	129.46112
distance3	float	0x8007FF68	18.224
distance4	float	0x8007FF6C	41.33312
distance5	float	0x8007FF70	38.4608
angleNormal	float	0x8007FD64	54.951046
Dist	float	0x8007FD60	37.373196

- Test 2: Directly in front of Array @ 42.5cm (resolved by all sensors)

angleNormal	float	0x8007FD64	-90.01698
Dist	float	0x8007FD60	40.059673



Status and Plans

- **Right Now**
  - Obstacle detection implemented on target platform
  - Mock-up sensor array tested with Pic
  - Construction plans for final assembly (unable to build due to weather)
  - Side facing detection implemented on pic
- **Upcoming**
  - Build Array module
  - Refine thresholds
    - Can't fully test and refine until we have final assembly to ensure correct sensor layout
    - Real obstacles vs ideal obstacles (can't predict what part of obstacle reflects US signal)
  - Side Facing obstacle detection function
    - Need power to test
    - Need assemble side array to fully implement (need dimensions)
  - Clean Up & Interface PIC
    - Remove excess code
    - Time operations
    - Share data with Nav MCU
- **GOAL: Begin integration end of next week**

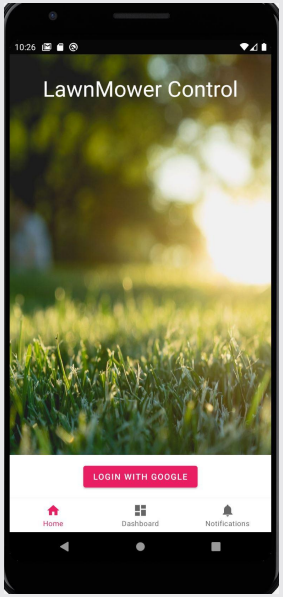
User Interface/Server Update

-Bugs all fixed

-Scheduling times

-Start & “Go home to docking station” successfully implemented through the firebase server

-Ready to receive statistics

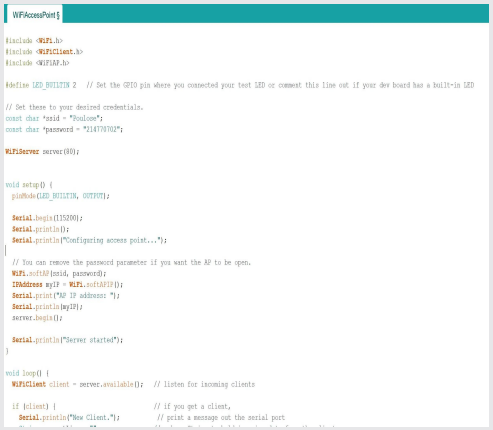


WiFi Update

- Using ESP32 DEVKIT V1
- Ran WiFiScan program
  - Set up as STA mode
- Ran WiFiAccessPoint Program
  - Set up as soft access point mode
- Started to write the code to be able to communicate with other ESP32

WiFiScan

```
#include "WiFi.h"

void setup()
{
    Serial.begin(115200);

    // Set WiFi to station mode and disconnect from an AP if it was previously connected
    WiFi.mode(WIFI_STA);
    WiFi.disconnect();
    delay(100);

    Serial.println("Setup done");
}

void loop()
{
    Serial.println("scan start");

    // WiFi.scanNetworks will return the number of networks found
    int n = WiFi.scanNetworks();
    Serial.println("scan done");
    if (n == 0) {
        Serial.println("no networks found");
    } else {
        Serial.print(n);
        Serial.println(" networks found");
        for (int i = 0; i < n; ++i) {
            // Print SSID and RSSI for each network found
            Serial.print(i + 1);
            Serial.print(": ");
            Serial.print(WiFi.SSID(i));
            Serial.print(" (");
            Serial.print(WiFi.RSSI(i));
            Serial.print(")");
            Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN)?" ":"*");
            delay(10);
        }
```



WiFiAccessPoint §

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WiFiAP.h>

#define LED_BUILTIN 2   // Set the GPIO pin where you connected your test LED or comment this line out if your dev board has a built-in LED

// Set these to your desired credentials.
const char *ssid = "FooCose";
const char *password = "2147787602";

WiFiServer server(80);


void setup() {
  pinMode(LED_BUILTIN, OUTPUT);

  Serial.begin(115200);
  Serial.println();
  Serial.println("Configuring access point...");

  // You can remove the password parameter if you want the AP to be open.
  WiFi.softAP(ssid, password);
  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(myIP);
  server.begin();

  Serial.println("Server started");
}

void loop() {
  WiFiClient client = server.available();   // listen for incoming clients

  if (client) {                             // if you get a client,
    Serial.println("New Client.");          // print a message out the serial port
```

Motors Subsystem Status

Motors subsystem ready to integrate

Navigation Subsystem Update

- Transferred MATLAB simulation code to C
- Successfully created 32 file project to run on the MCU
- Working on inserting the code for it to work with real time signals from shaft encoders, PWM, and motors
- Should be ready to integrate on time in reference to our schedule
- Code on next slide

Navigation cont.

ENGINEERING
TEXAS A&M UNIVERSITY

```c
#include <stddef.h>
#include <stdlib.h>
#include "rtwtypes.h"
#include "mower_sim_types.h"

/* Function Declarations */
extern void mower_sim(double mower_width, double D0, double MAX_LEGS, double
    leg_number, double r_e, double dt, double wheel_diam,
    double Cw, double shaft_encoder_levels, double
    shaft_encoder_steps, double v_mow_normal_mph, double
    v_mow_turn_mph, double mph_to_mps, double mps_to_rpms,
    double rpms_to_mps, double deg_to_rad, double rad_to_deg,
    const unsigned char a[524547], double lat_start_deg,
    double lon_start_deg, double heading_deg[10000], double
    L_rpms_desired[10000], double R_rpms_desired[10000],
    double L_rpms_actual[10000], double R_rpms_actual[10000],
    double v_left_actual_mps[10000], double
    v_right_actual_mps[10000], double L_wheel_exact_deg,
    double R_wheel_exact_deg, const double t[10000], double
    next_heading_deg, const double L_shaft_enc[10000], const
    double R_shaft_enc[10000], const double x[10000], const
    double y[10000], double d[10000], const double lat_deg
    [10000], const double lon_deg[10000], double frame_num);
```

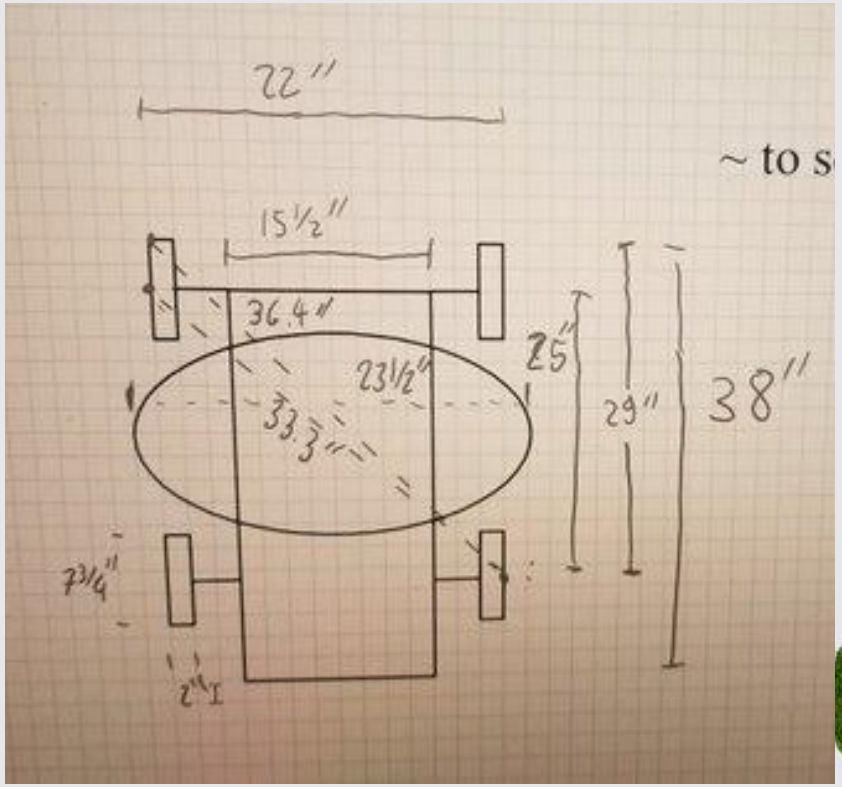
mower_sim.h (main function and variable initializations)

```c
while (leg_number < MAX_LEGS) {
    next_heading_deg += 90.0;
    d[n] = 0.0;

    /*  resent distance each turn completion */
    leg_number++;

    /*  make the leg distance smaller every so often so the the mower will go */
    /*  in a pattern of decreasing box sizes */
    D_meters = leg_number - 2.0;
    if (!(D_meters > 0.0)) {
        D_meters = 0.0;
    }

    D_meters = D0 - floor(D_meters / 2.0) * mower_width / 2.0;

    /*  %     fprintf('Leg number, D_meters = %i %f\n',leg_number,D_meters) */
    while (d[n] < 0.9 * D_meters) {
        /*  at 90% of desired distance start turning */
        n++;

        /*  update distance-within-leg vector */
        d[n] = d[n - 1] + (v_left_actual_mps[n - 1] + v_right_actual_mps[n - 1]) /
```

mower_sim.c (snippet of navigation code)

Docking Station Update

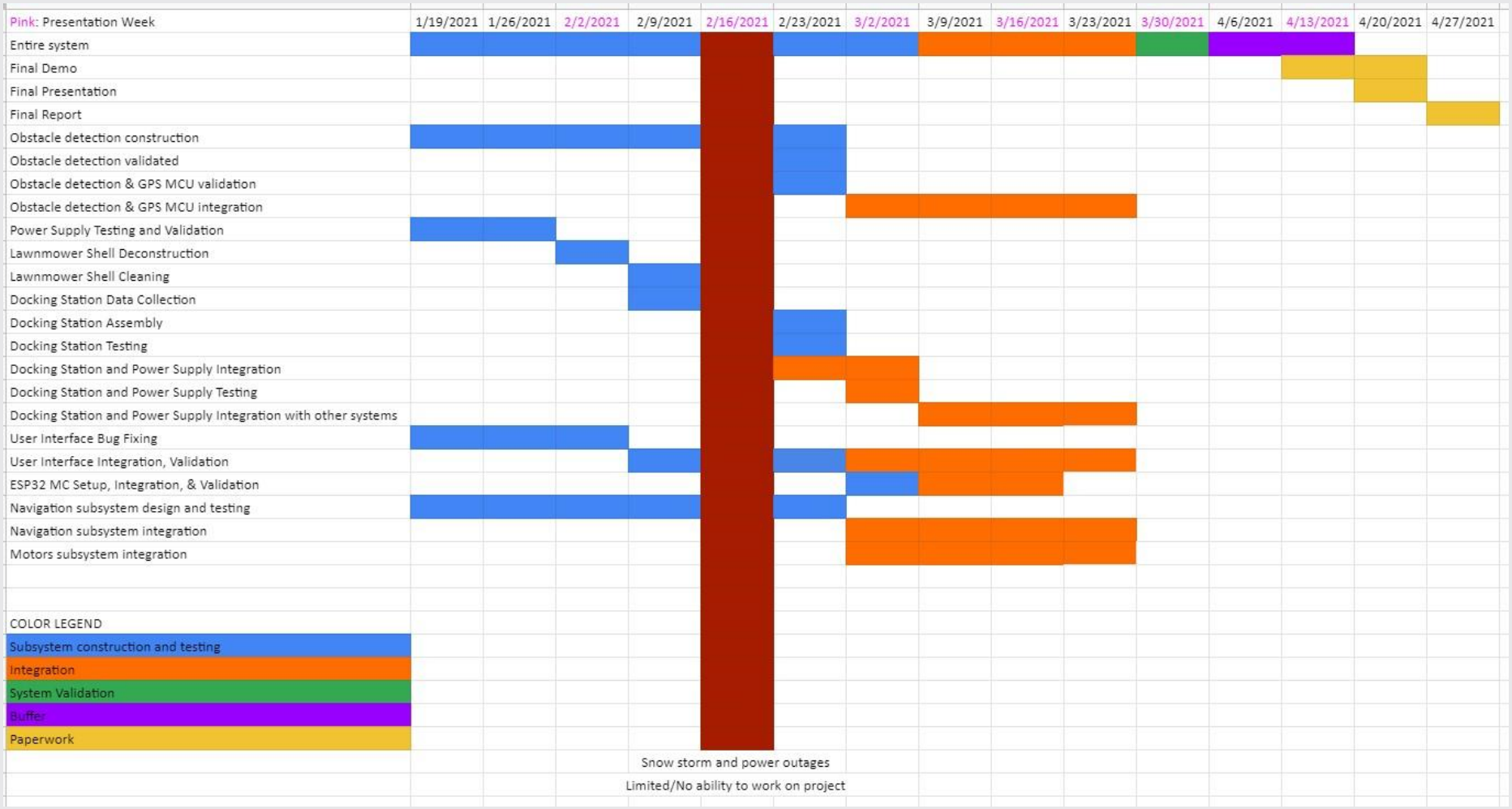
- Lawnmower has been disassembled and cleaned
  - Need to figure out when it can be brought to the FEDC
- Next step is to assemble the docking station



Approximately to scale, each square represents 2 inches by 2 inches

Schedule for the Semester

ENGINEERING
TEXAS A&M UNIVERSITY



Validation Plan

ENGINEERING
TEXAS A&M UNIVERSITY

Requirement title and heading, as found in the FSR	Abridged text of requirement (see FSR for full text)	Methodology	Status	Responsible team member	Validated By
3.2.1 Functional/Performance requirements	Performance requirements for the compelates system				
3.2.2.1 Operational Stamina	The system shall mow grass continuously for no less than 1 hour in flat, obstacle free terrain	Time operation of system under normal conditions		All	
3.2.1.2 Duty cycle	The system shall perform operations at least every 7 days.	Validate Solar/grid charging rate against power consumption in above test		All	
3.2.1.3 Obstacle Avoidance	The system shall avoid all obstacles that are harmful to the overall system, running over small obstacles that only impact blade sharpness or appearance is acceptable	Individual evaluation of obstacle detection and navigation function under realistic input. Full system validation in Realistic terrain		J. Samaniego & M. Lesser	
3.2.1.4 Navigation	The system shall use internal sensors to follow a spiral pattern within the boundaries outlined by the user	Simulation of Nav-code, Full system test under realistic conditions		J. Samaniego	
3.2.1.5 Obstacle detection Range and Threshold	The system shall be aware of any obstacle in the hypothetical box extending forward from the front of the mower to 2M, as wide as the widest point of the mower	Individual test of Obstacle detection function, in final configuration		M. Lesser	
3.2.2 Physical characteristics	Physical Requirements on the completed system				
3.2.2.1 Mass	The mowing unit shall not exceed a maximum weight of 70lbs	Weigh final system		All	
3.2.2.2 Volume Envelope	The volume envelope of the Lawnmower shall be less than or equal to 30 inches in height, 25 inches in width, and 40 inches in length.	Measure final system		All	
3.2.2.3 Mounting	All components shall be mounted in a fashion to resist vibration incidental to lawnmowing, with service once every 6 months when used weekly	Test connection manually, not able to perform full test, will have to extrapolate results gained from other tests		All	
3.2.3 Electrical Characteristics	Definitions of expected external inputs and outputs				
3.2.3.1 Inputs	The system shall not be damaged by any possible inputs or signals produced by the system, No user input shall result in the system engaging in unsafe or damaging operations	Provide mower with all possible input signals and observe performance		All	
3.2.3.1.1 Power Consumption	The power consumption of the lawnmower unit shall not exceed 200 Watts.	Measure battery discharge level after use		V. McMasters	
3.2.3.1.2 Input Voltage Level	The input voltage level for the Lawnmower shall be 14.2 VDC to 14.4 VDC.	Measure input voltage to battery from charging station		V. McMasters	
3.2.3.1.3 External Commands	The Lawnmower system shall receive external commands from the User Interface via a WIFI connection. Details will be outlined in the ICD.	Attempt to transmit Nav/Schedule commands from app and see if mower responds		J. Poulose	
3.2.3.2.1 Data Output	The mowing unit shall inform the user of problems and fault conditions through the UI app via WIFI.	Generate Fault conditions on mower and see if reports to user		J. Poulose	
3.2.3.2.2 Diagnostic Output	The MCU shall include a hardware debugging port that may be interfaced to a computer for Diagnostics.	Physical validation of hardware access ports		M. Lesser	M. Lesser
3.2.3.3 Connectors	(Electrical) Connectors shall be resistant to vibration incidental in lawnmowing, with service no more than once per 6 months when used weekly	Extrapolate from other physical trials		All	
3.2.3.4 Wiring	The wiring for signal and power interfaces shall be routed clear of any moving internal parts, and clear of all possible outside interference. And protected as appropriate	Visual inspection of completed system		All	
3.2.4 Environmental Requirements	The Lawn mowing system shall operate in all environmental conditions that traditional residential lawn mowers operate and lawn care activities take place.	Test in as many environmental conditions as possible and extrapolate			
3.2.4.1 Thermal	The Lawnmower shall operate in temperatures ranging from 40°F to 120°F.	see 3.2.4		All	
3.2.4.2 External Contamination	The Lawn mower shall be immune to dust and debris. The Lawn mower systems shall either be protected from, or insensitive to ingress of debriess 1mm or larger, as well as dust.	see 3.2.4		All	
3.2.4.3 Rain and extreme weather	The Lawn mower shall not operate in rain. It shall be able to withstand exposure to the elements when parked in the docking station.	see 3.2.4		All	
3.2.4.4 Humidity	The Lawnmower shall function temporarily in conditions of up to 100% humidity, but requires lower humidity or higher maintenance for long term storage and performance.	see 3.2.4		All	
3.2.4.5 Soil Moisture	The Lawn mower shall be able to operate on moist, but not wet solid, on level terrain.	see 3.2.4		All	
3.2.4.6 Distance from Router (WIFI connection distance)	The Lawnmower shall be able to communicate with the network at the operating site from at least 100 ft and through at least 1 wall of wood/drywall construction	see if mower responds when specified distance from WIFI router		J. Poulose	
3.2.4.7 Sky clearance	The Lawnmower shall be able to operate with light to medium foliage overhead	test GPS unit in mounting configuration under specified overhead cover		M. Lesser	
3.2.4.8 Vibration	The Lawnmower system shall operate without failure, under vibration incidental to lawn mowing for at least 6 months, when operated once weekly for 1 hour.	see 3.2.4		All	
3.2.5 Failure Propagation and protocols	No failure shall cause to mower to endanger bystanders				
3.2.5.1 Blade error	The lawnmower's user interface will notify the user if the blade is stuck on an obstacle. In this case the blade will shut down automatically	Simulate stuck blade and validate mower sends error message and disabled blade drive		J. Samaniego & J. Poulose	
3.2.5.2 Mower stuck	If the mower becomes stuck in terrain it shall power down, disabling the blade and alert the user.	simulate/force mower to become stuck and monitor response		J. Samaniego & J. Poulose	
3.2.5.3 Lost Wifi connection	In cases where the WIFI connection to the user device is lost the mower will continue on its planned route and return to the rest position.	disable WIFI router and monitor response		All	
3.2.5.4 Lost GPS connection	If the mower loses GPS connection it shall attempt to follow the planned route to the best ability.	disable GPS module and monitor response		M. Lesser & J. Samaniego	
3.2.5.5 System Failure	In cases of system failure the mower shall alert the user through the UI, disable the main blade and return to the start position, if possible	Simulate system failure, (f.e. by disconnecting sensors O.S.) and monitor response		All	
3.2.5.6 Critical System Failure	In Critical Failure cases, that is situations in which the MCU loses all ability to control the mower or it's subsystems the lawnmower blade will shut off.	Power down MCU during operation and monitor response		All	

Legend
Not Yet Validated
Falied
Passed
Revised/adjusted

Video Link

Attached here is the link to the video of our presentation:

https://youtu.be/IHkTR2UbbH4