Autonomous Solar Lawn Cutter (eGOAT)

Ali Hamdani, Anderson Kagel , Max Seberger

University of Central Florida School of
Electrical Engineering and Computer Science,
Orlando, FL

Abstract — A sponsored project by OUC and Duke Energy, the eGOAT (electronically Guided Omni-Applicable Trimmer) is a play on original commercial designs but on an industrial scale as an autonomous lawn mower. It features technology such as Lidar and LoRa technology. It has been designed to be efficient with the usage of power and efficient in its distribution. It has been worked on by three teams of students: Electrical-Computer, Mechanical and Computer Science. The project requirements stated by the sponsors have a number of constraints the engineers had to work with. The eGOAT is supposed to be in the same price range as commercial products on the market today but shall run at a steadier and more robust performance. It shall be able to navigate high grassy fields and be able to handle the challenges associated with uneven terrain and obstacles.

I. INTRODUCTION

The optimal location for PV panels for large generation of energy are normally in open areas such as a grassy field. One of the downsides of these types of locations is the landscaping required from keeping grass and other vegetation cut so it does not hinder the ability of the solar panels and operation of the solar farm as a whole. Solar panels require to be in direct sunlight to run optimally, and that is variable with cloud cover, storms, etc. Thus removing the external factor of grass and plant growth hindering the operation of a PV panel is imperative when running a solar farm. The main goal behind the eGOAT (electronically Guided Omni-Applicable Trimmer) is to cut the costs and potential risks involved with human ground maintenance. The eGOAT must be autonomous, low cost, smart, and effective. An autonomous system such as the eGOAT should be able to operate without frequent service or interference, while still providing an effective grass cutting service and ensuring the safety of solar infrastructure during operation. As the name suggests, the eGOAT will be powered electrically, drawing its energy from an array of integrated PV panels, unlike conventional, human-operated mowing equipment that run off of fossil fuels and require to be refilled frequently. This means that the eGOAT will be totally renewable, saving both the owner and environment the recurring costs associated with less eco-friendly options.

II. SYSTEM COMPONENTS

The eGOAT can be sectioned into key components with individual purposes for the project. Everything from powering and controlling other components. A number of components are being programmed and will be using a large amount of software to control.

A. Printed Circuit Board

The printed circuit board (PCB) is the framework of the electrical design by allowing all the electrical components to connect together. It will both house and power the microcontroller ESP 32 , and be able to power two vital components: Lidar and Raspberry PI. The PCB takes a 12 volt battery supply and steps down the voltages to two 5v distributions and one 3.3 volt line. The main step down circuit IC used was the LM2596 which has two versions a 5 and 3.3 volt. The pcb also houses 2 usb female ports to connect components needed.

B. ESP32 Real-Time Hardware Monitor

The Espressif ESP32 is a low-power system-on-a-chip using the 32-bit dual-core Xtensa LX6 microprocessor @160MHz; it has 512KB of RAM, 16MB of internal flash storage, and 34 GPIO pads. The ESP32 was chosen as our hardware monitor due to its low price, large number of GPIO pads, and integrated WiFi capabilities. As our hardware monitor, the ESP32 is responsible for detecting and responding to all emergency kill commands in real time by cutting power to all motors until command is rescinded. Because the long-range kill commands come in via LoRa, all LoRa messages are also routed through the ESP32 and passed to the main computer of the eGOAT if necessary.

C. YDLIDAR X4 Lidar

The YDLIDAR X4 Lidar is being used to understand where the eGOAT is and what is approaching it. The spinning Lidar that best meets our needs while still remaining affordable is the RPLidar A1M8- 360 Degree laser scanner. It is capable of a fast enough refresh rate and rotation frequency that will allow good measurements of the environment to be used in mapping.

D. Raspberry Pi Main Computer

The Raspberry Pi 3 Model B+ is a single-board-computer using a 64-bit Cortex-A53 quad-core processor @ 1.4GHz with 1GB of RAM. We ultimately decided to use the Raspberry Pi 3 Model B+ as the main onboard computer for the eGOAT. Although its hardware specifications are comparatively weaker than the other options we considered, its ease of integration and huge knowledge base are too powerful of a tool to pass up on. In order to make up for the deficit in performance, the team has elected to make use of a graphics coprocessor

The knowledge base available for implementing any model of the Raspberry Pi in a huge variety of applications completely dwarfs that of any number of single-board-computers, and the number of well-documented, open source solutions available (to both robotics specifically and technical issues in general) is truly staggering. In fact, the Raspberry Pi form factor, hardware compatibility/interoperability, and performance profile has become something of a standard and benchmark in this field, and the Raspberry Pi Model 3 B+ that was recently released offer respectable performance at a one of the lowest price points available.

E. Accelerometer/Gyroscope

The chip chosen for the eGOAT is the MPU-6050 3-axis accelerometer/gyroscope. The accelerometer will be used to send data on its orientation and if there are any abrupt forces being applied to the eGOAT. This helps the system control it's speed and direction. The accelerometer is attached to the GPIO pins of the Raspberry Pi and integrated into the ROS navigation stack in order to be used for inertial navigation.

F. Motor Controller

The eGOAT will be using a pair of Cytron 10A dual channel DC motor controllers to control the speed and direction the motors operate. We are using two dual control controllers for the four motors needed (two drive motors and two trimmer motors). These motor controllers have inputs for motor power and ground as well as a direction signal and a pulse-width-modulation signal for each channel.

G. Trimmer Motors

The eGOAT will be using Ryobi String Trimmer motors. That will be off the shelf to help cut the grass. They will be connected to a motor controller that helps control the speed and direction using pulse width modulation. They will be attached at the front

H. Drive Motors

The drive motors selected are two Lynxmotion 12V 90 rpm brushed dc gear motors. The motors come with hall sensor encoders for communication with our hardware controller to help understand the orientation of the motors at a certain point in time. The nominal voltage is 12V with a max stall current of 10A. These motors also include a 26.9:1 PG45 Gearbox for gear changing for different situations while moving.

III. SYSTEM CONCEPT

The main goal behind the eGOAT is to cut the costs and potential risks involved with human ground maintenance. As the name suggests, the eGOAT will be powered electrically, drawing its energy from an array of integrated PV panels, unlike conventional, human-operated mowing equipment that run off of fossil fuels and require to be refilled frequently. This means that the eGOAT will be totally renewable, saving both the owner and environment the recurring costs associated with less eco-friendly options. The AI-guided task planning software of the eGOAT will be able to detect obstacles in the environment, identify locations and objects in need of trimming, and construct a sequences of tasks to accomplish mission objectives and avoids any potential damage to important infrastructure. The system will be using a variety of components such as raspberry pi , Lidar, esp , and voltage step down modules. To help the eGOAT run autonomously and functionally.

A. System Hardware Concept

The figure below shows how the hardware will be connected via power and data lines. As you can see all the components have either power or data connected to the printed circuit board for the eGOAT

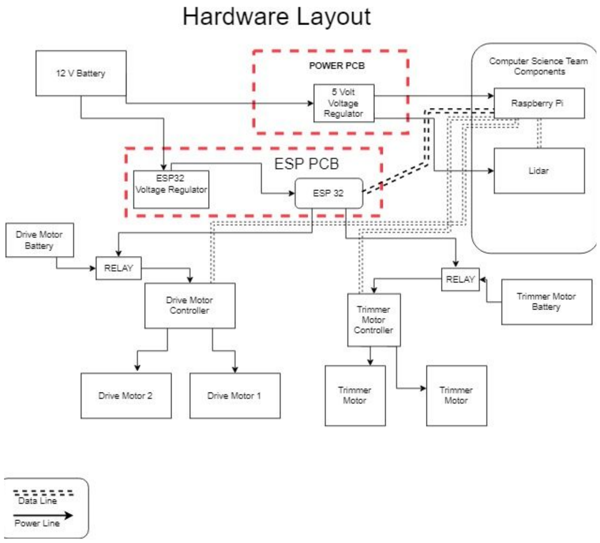


Fig. 1. Block diagram presenting major system components

IV. HARDWARE DETAIL

A. Voltage Step Down Circuits

The voltage step down circuits comprise of 5 elements: Switching Regulator, Input Capacitor, Inductor, Diode, and Output capacitor. The step down circuits purpose is to step down the 12V battery to different voltages needed to operate the ESP 32, Raspberry Pi and Lidar lines. A system level view is provided in the figure below.

(1) Switching Regulator (LM2596) is a 5 pin IC that helps step down the input voltage. It has an input voltage of 0 to 40v and the versions being used will be the the 5.0 and 3.3.
(2) The input capacitor is used mainly for protecting the supply side of the circuit from any transients that may damage the supply , in this case our battery
(3) The output capacitor is used to filter the output to output a cleaner voltage signal
(4) The inductor is to help the circuit run in continuous mode
(5) The diode we used is a fast acting diode so when the circuit turns on the current going through the inductor is high.

The LM2596 has five pins that need to be connected to certain parts of the circuit as you can see in the figure below.

(1) Input pin: Supply voltage terminal
(2) Output pin: supplies the output signal (5 or 3.3)
(3) Ground pin: grounding of circuit
(4) Feedback: A wire that reads the IC the output voltage for feedbacking IC
(5) On/Off pin: Needs to be grounded or untouched to keep the IC on , needs to be energized to turn off the IC and shut off usage of circuit.

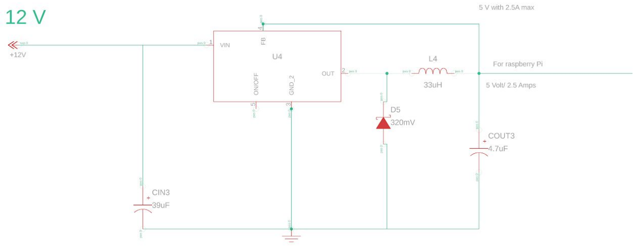


Figure 3: Step down circuit for 12 to 5v

The step down circuits output a 5v and 3.3v signal with a maximum amperage of 2.5A. They provide overcurrent and thermal protection to both supply and load sides. The supply side will be through terminal blocks and all three signals will get their own terminal block for better current distribution along the PCB. The three components being powered through the stepdowns are the ESP32 , Raspberry Pi B+ and Lidar. The Lidar and B+ need 5V connections but have different amperage requirements and the ESP has a 3.3V specification. The reason for the stepdown is to integrate one power supply for all the data transmitting components. One supply ensures that everything will be working in unity , eliminating the contingency of different power supplies losing charge and disrupting the operation of the system.

B. Motor Controller

The motor controllers selected will be external of the pcb because of the fact that they are not pcb acceptable but will be connected via data line from the Raspberry Pi.. The motors need a very steady voltage source so a seperate battery to power the motors through the controllers will be utilized.It is best to not run the current through our pcb because of the high current draw that can occur. We chose not to because current will be high and fluctuating and can impart a transient response back to our main battery.

The motor controller for the drive motors and trimmer motors will be the Cytron MDD10A. The MDD10A allows for two motors to be controlled so this ideal knowing that we have two drive motors that need to be in sync for proper operation. It has an input of 5 pins: Ground, Pwm1, Pwm2, DIR1, DIR2. The PWM pins are the input speed for both the designated motors, which would be connected to ESP32. The DIR or direction input pins would also be data lines connected to

ESP32. The MDD10A also has a terminal block for the motors and power of 6 Pins: Motor outputs for 1 and 2, and the Power terminals of the battery, The reason for the MDD10A is for the simplicity of two motor accessibility and control. The drive motors selected are brushed DC gear motors with an operating nominal voltage of 12 Volts. The drive motors have an encoder connector that helps connect the positive and negative terminals of the motor. There are 6 pin connections on the drive motors: Positive, Negative , Ground, Hall sensor VCC, Hall sensor A Vout and Hall sensor B Vout. The Hall sensor Vout pins will be connected to the ESP32. The hall effect sensors will help the ESP 32 understand how the motors are oriented in a rotational sense.

To control the trimmer motors we will be using the same type of controller , the Cytron MDD10A. The motor controller for the trimmer motors will have the same functionality as the drive motors. The only pin that may not be used is the DIR pin or direction pin. There would be no need to change the direction of the trimmer motors (clock or counterclockwise) because they are just tools that will be continuously running and performing lawn cutting. The trimmer motors have not been selected yet because of our mechanical team's decision timing. We have been assured that our motor controller for the trimmers will be able to operate the voltage of the trimmer motors. The MDD10A has a voltage operation range of 5 to 30 volts, which is large range to select trimmer motors.

C. Drive and Trimmer Motors

The drive motors selected are two Lynxmotion 12V 90 rpm brushed dc gear motors. The motors come with hall sensor encoders for communication with our hardware controller to help understand the orientation of the motors at a certain point in time. The nominal voltage is 12V with a max current if stalling were to occur at 10A. These motors also include a 26.9:1 PG45 Gearbox for gear changing for different situations while moving. The figure below shows the signal the hall sensor encoders will send to the motor controller computer which aids the computer with understanding the distance traveled and orientation of the motors.

The trimmer motors are ryobi electric weed wacker heads that have expanding cutting strings while in operation. They are controlled by the motor controller and have front placement on the eGOAT as seen in figure 3.

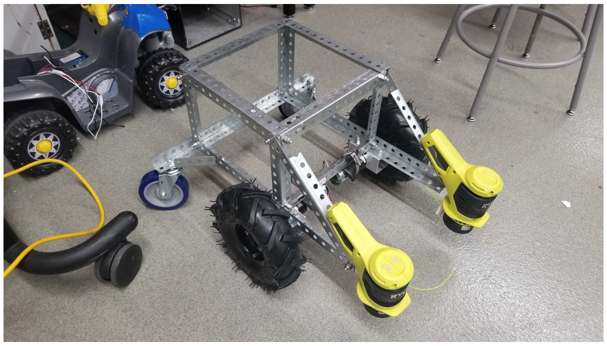


Figure 6: Early stage of eGOAT creation with trimmer motors and drive motors attached.

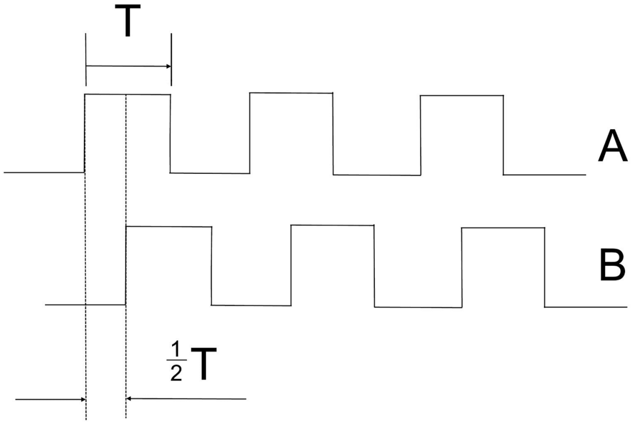


Figure 5: Hall effect sensor output signal

D. Battery

Our design comprises of 3 batteries one 12 volt battery for the PCB components, one 12 Volt Battery for the drive motors, and one 18 volt battery for the trimmer motors. Three different batteries were chosen for our design for the main reason of safety, simplicity and protection. We wanted a seperate battery for the PCB components so that they will have a stable power source that will not be interrupted or damaged by transient voltage feedback from motors. each individual motors is capable of drawing up to 10 Amps at stall operation. At that high of a current it can be detrimental to operation of our components if the battery was shared with the computers and the motors. For safety of our components we thought it was important for two seperate batteries to run two seperate systems. The trimmer motor battery* will be separate as well knowing that it has to be a different voltage. The reason we decided not to use a step up from a 12 volt source was the instability it can create and the nature of the motors which can create transient current to run through the system. The trimmer motors have not been selected in time for us due to the mechanical team's decision timing. The only concrete information we had received was that the motors would be 18 volt due to off the shelf ratings of trimmer motors. The following equation was used to help calculate which specifications for batteries we needed.

V - Battery Voltage, I - PCB current draw ,

T - operation time (hours),

X - total watt hours , C - battery capacity (amp-hours)

$$V * I = P \; ; \; P * T = X; \quad X \div V = C$$

Equation 1 : Battery Capacity calculation

VI. SOFTWARE DETAILS

The command and control paradigm of the eGOAT can be best modeled as a hybrid deliberative-reactive architecture utilizing three control layers. They are, from top to bottom, the administrative/communication layer, the planning/control layer, and the reactive/manipulation layer. In general each layer is subservient to the commands of the layers above it. Circumstances in which this hierarchy is violated are discussed in the *Hardware Controller Safety Interrupt* section below.

A. Sensor Integration

Each sensor is modeled internally with a statistical model of its accuracy. This model represents the statistical likelihood that a sensor will detect an obstacle in a given position given that position actually *is* occupied by an obstacle. The equation representing this probability is given in equation 2 below. These statistical models of each sensor will be found experimentally as development continues

The readings of each sensor on this position are then synthesized into a statistical point map and dynamically updated as more data becomes available. Baye's theorem is used to combine these probabilities and is given below in equation 2. .

Equation 2: Sensor Model of Hypothetical Sensor S

Probability of sensor detection (S) given occupied cell (o)

$$P(S|o)$$

Equation 2: Baye's Theorem

$$P(o|S) = \frac{P(S|o)P(o)}{P(S)} = \frac{P(S|o)P(o)}{P(S|o)P(o) + P(S|\neg o)P(\neg o)}$$

B. Lidar

The Lidar is the primary rangefinding sensor that the eGOAT will use to construct its occupancy map. It allows for highly accurate rangefinding capabilities in all directions around eGOAT with a resolution measured in centimeters. However, the Lidar is only capable of collecting range data in a 2D plane around the eGOAT, meaning that it will need help from other sensors to detect difficult terrain or obstacles that sit on the ground below the Lidar's vertical range.

C. Simultaneous Localization and Mapping (SLAM)

SLAM is designed to create and update a map within an unknown environment while keeping track of the robots location. SLAM is not a specific algorithm, but more of a set of algorithms and technologies that work on solving localization (where the device is) and mapping the landscape at the same time. Another important point of SLAM is that it must work in real-time. SLAM finds the position by combining the measurements of points made over a set of frames. SLAM's way of optimization is finding the best position for the camera to be able to always deduce its location. The consensus is that the best method is the bundle adjustment. The bundle adjustment gives the best starting configuration and gives the minimum error. A key problem with it is that it is very time consuming, a solution to this is the advent of multicore computing. It allows one core to focus on mapping in real time and the bundle adjustment can take its time in another core.

Another key feature of SLAM is loop closure, a large scale mapping technique that is most certainly needed for something like the eGOAT. The key points of this feature is to reduce error by associating a previous tracked location with a current one. Over more and more iterations the map will become much more accurate. Another valuable technique is relocalisation which allows the device to reset it position to a previously safer view that is similar to the one it is currently seeing when the now registered position results in a system failure.

D. Computational Hardware

The computational requirements of the eGOAT are twofold: first a central "main brain" computer that has most of the processing power, and second an array of lesser computation hardware that is slaved to the central computer. In order to be an attractive solution to the landscaping needs of our sponsors, the eGOAT must be able to perform its function in a variety non-standard locales that were not designed with its use in mind with as little installation and spool-up time as possible. Essentially, the system must be capable of operating generally and effectively in any number non-engineered environments with little or no pre-existing information.

This fact, along with the large number of potential obstacles and wide-ranging set of possible tasks the eGOAT must perform, will require the implementation of some kind of simultaneous localization and mapping (SLAM) algorithm, as well as the algorithms to handle navigation; sensor filtering and synthesizing; and short-term and long-term task planning just to name a few. The relatively large processing power and multitasking required by these algorithms necessitates the inclusion of a reasonably powerful computer with an operating system and graphics processing in order to coordinate the various functionalities and subsystems of the eGOAT as well as interpret the data feed from the sensor suite and servos in the amount of time available.

However, the inclusion of an operating system as well as the amount of processing it is responsible for also means that we can not depend on this central computer to always operate in real-time. Therefore, we must include some computation hardware that we can rely on to control those critical subsystems and functions that absolutely must operate correctly in real time. These smaller subsystem controllers will be slaved and bussed directly to the main computer to handle a large assortment of task, including: accept higher level directives from the main brain, expand upon those directives to deliver commands to low-level hardware systems, receive and cache feedback from sensors and servos, and respond to those emergent inputs that require an immediate response for the safety of humans or the system itself. These controllers will also be responsible for monitoring the internal hardware status performance of the subsystems and will be capable of immediately shutting down these subsystems and notifying the main computer in the case that continued operation would endanger the system's integrity in some way, such as electrical surge or malfunction, mechanical failure, loss of sensor input, or other erroneous behavior.

E. Task Planning and Navigation

As the eGOAT explores and constructs a map of its environment, it also builds and maintains a prioritized list of tasks to complete. The system then uses this data in a path planning algorithm to find the optimal path to complete its objects. In robotics the optimal path is most the one with the physically shortest distance. The eGOAT's main objective is to cut all the grass available and that would include all physical space not occupied by a obstacle. This essentially means we have to design our algorithm cover all the registered map data while avoiding obstacles. It is more reminiscent of a pacman scenario where the walls are our boundaries, the ghost are our obstacle, and we must touch on all the pellets in order to consider it a completed job.

Dijkstra's, A*, and other common path planning algorithms may still be valuable in certain cases, but as it stands they are only looking to find the shortest distance, and in our case the optimal path isn't the shortest distance from point A to point B. We require a that all locations are visited and these algorithms don't keep that in mind. To remedy this, we have constructed our ROS navigation stack to include a local planner, a global planner, and an exploration planner. The global and local planners, both implemented by the *Gmapping* ROS package, are responsible for point-to-point navigation and long-term mapping respectively. The exploration planner extends this using the *frontier_exploration* ROS package to formulate a path that optimally visits (and cuts) each cell.

Once the pathfinding algorithm has constructed a path to its next objective, this path is broken up into a series of waypoints and commands that can be performed by the hardware. These commands can generally be separated into two groups: time-based commands, which can be queued and performed asynchronously such as changing the eGOAT's speed or toggling the trimmers, and sequential commands, which must be performed one after another such as move commands. Upon receiving confirmation from the hardware and sensor data that a task has been completed, the task is removed from the onboard computers queues as well. The figure below shows a block diagram for Navigation and Sensor usage.

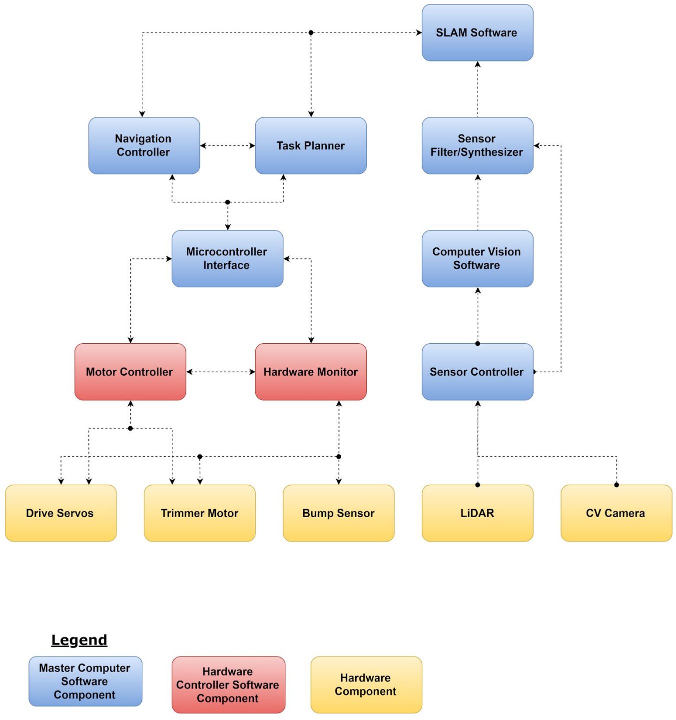


Figure 6: Navigation Stack

F. Hardware Controller Safety Interrupt

The hardware controller is primarily concerned with running the software needed to respond to situations that must be dealt with in real time, especially those concerning the safety of the eGOAT and the humans around it. These safety features use interrupt driven software to immediately cut power to all motors and come in the form of two hardware components capable of interrupting the hardware controller: a physical kill switch attached to the eGOAT and a LoRa radio transceiver.

The physical kill switch is a large, red button with two channels, one active high and the other active low. When pressed, one channel of the kill switch physically cuts power to the trimmer motors, providing a hard-wired fallback to ensure that the eGOAT is incapable of cutting while pressed. The other channel sends a digital signal telling the hardware controller to open the relays connecting the drive and trimmer motors to the batteries, stopping all motor movement.

The LoRa transceiver is the primary means by which the eGOAT can receive a kill command remotely, and because the hardware controller must be able to halt the movement and trimming of the robot in a fast reactionary manner it also behaves as the LoRa communications monitor. There are predefined commands which open or close the motor relays when received in the same way as the physical kill switch, and any non-emergency LoRa messages are forwarded to the main computer via UART.

The figures below summarizes the relevant software interrupt control flow and provides code snippets of their implementation.

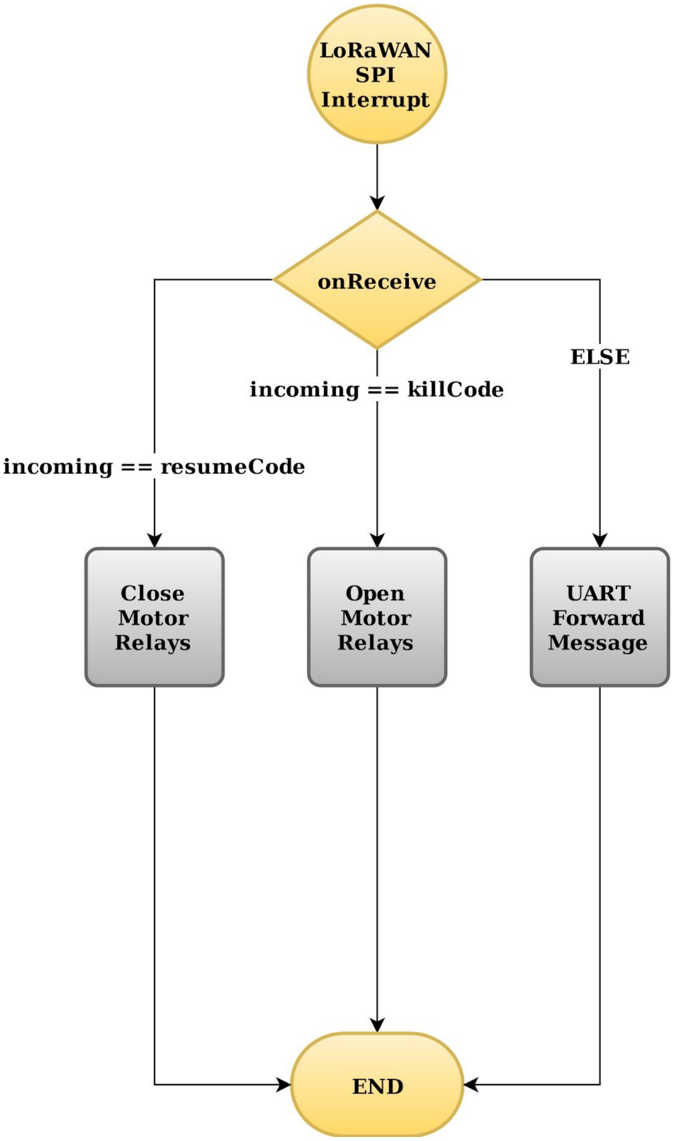
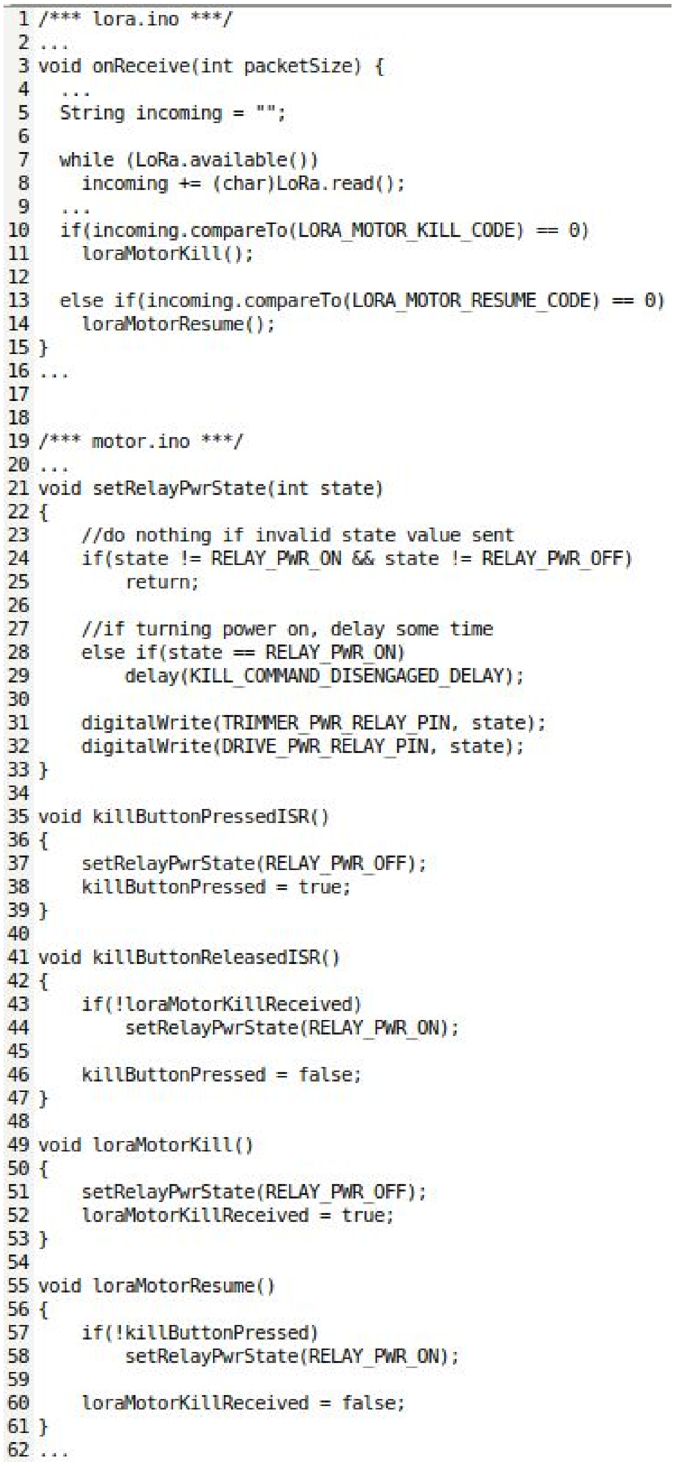


Figure 7: Hardware Interrupt Flow Chart



```
/*** lora.ino ***/
...
void onReceive(int packetSize) {
  ...
  String incoming = "";

  while (LoRa.available())
    incoming += (char)LoRa.read();
  ...
  if(incoming.compareTo(LORA_MOTOR_KILL_CODE) == 0)
    loraMotorKill();

  else if(incoming.compareTo(LORA_MOTOR_RESUME_CODE) == 0)
    loraMotorResume();
}
...


/*** motor.ino ***/
...
void setRelayPwrState(int state)
{
    //do nothing if invalid state value sent
    if(state != RELAY_PWR_ON && state != RELAY_PWR_OFF)
        return;

    //if turning power on, delay some time
    else if(state == RELAY_PWR_ON)
        delay(KILL_COMMAND_DISENGAGED_DELAY);

    digitalWrite(TRIMMER_PWR_RELAY_PIN, state);
    digitalWrite(DRIVE_PWR_RELAY_PIN, state);
}

void killButtonPressedISR()
{
    setRelayPwrState(RELAY_PWR_OFF);
    killButtonPressed = true;
}

void killButtonReleasedISR()
{
    if(!loraMotorKillReceived)
        setRelayPwrState(RELAY_PWR_ON);

    killButtonPressed = false;
}

void loraMotorKill()
{
    setRelayPwrState(RELAY_PWR_OFF);
    loraMotorKillReceived = true;
}

void loraMotorResume()
{
    if(!killButtonPressed)
        setRelayPwrState(RELAY_PWR_ON);

    loraMotorKillReceived = false;
}
...
```

Figure 8: Interrupt Service Routines Code Snippet

VII. BOARD DESIGN

The board design mostly comprises of power management but houses the ESP32 and the discharge circuitry needed as seen in the figure below. The board provides an efficient and simple connection to power the components needed for the project. The PCB was a standard size for eagle and contains everything needed to connect everything seamlessly from supply to load. The board design was created in a way to help the student that is soldering do efficient and quality work. All but one part were all smd parts and an oven was needed at one point to help solder the ESP32. Extra modifications after the soldering were needed to help the performance of the power modules. Time and budget being a constraint we needed quick but effective ways to power components using our PCB.

THE ENGINEERS

Ali Hamdani is a senior electrical engineering student who has specified courses in power systems. He will be beginning a career in the utility industry as a substation and relay engineer at Pepco. He has passed his FE exam and plans to attain his PE in the future.

Anderson Kagel is a senior computer engineering student. His career goals are to continue his working in simulation and embedded systems as a software engineer. He plans to one day work full time in robotics and other intelligent electronic applications.

Max is a 22 year-old Electrical Engineering student. After he graduates he has already made a commitment to work in the United States Air Force as an developmental engineer. He hopes to be tasked on R&D projects that allow him to help develop new capabilities for today's war fighters.

ACKNOWLEDGEMENT

The authors wish to acknowledge the assistance and support of Dr.s Samuel Richie, Solar lawn cutter Green Team: Mechanical and Computer Science Teams and the University of Central Florida.