**ABSTRACT**

The project aims at fabricating a grass cutting machine system which makes the grass cutter motor running through solar energy. The “Solar Powered Grass Cutting Machine” is a robotic vehicle powered by solar energy that also avoids obstacles and is capable of automated grass cutting. The system uses 12V battery to power the vehicle movement motors as well as the grass cutter motor. A solar panel is used to charge the battery so that there is no need of charging it externally. The grass cutter and vehicle motors are interfaced to microcontroller that controls the working of all the motors. It is also interfaced to an ultrasonic sensor for obstacle detection. The microcontroller moves the vehicle motors in forward direction in case no obstacle is detected. On obstacle detection the ultrasonic sensor monitors it and the microcontroller thus stops the grass cuter motor to avoid any damage to the object/human/animal whatever it is and it also provides an alarm. Microcontroller then turns the vehicle as long as it gets clear of the object and then moves the grass cutter in forward direction again otherwise it changes the direction. And also fully automated using android app.

**ACKNOWLEDGEMENT**

At this pleasing moment of having successfully completed our project , We wish to convey our sincere thanks and gratitude to the management of our college and our beloved Chairman **Dr**. **V.NARAYANASAMY,** who has provided all the facilities to us.

We would like to express our sincere thanks to our beloved Vice Chairman **Mr.N.HARIKRISHNAN** andOur Director **Mrs.B.ARTHI HARIKRISHNAN**,for giving us the opportunity and support.

We would like to express our sincere thanks to our Principal **Mr.K.KALAISELVAN** for forwarding us to do our project and offering adequate duration in completing our project.

We are also grateful to the Head of Department **Mr.S.VIJIAYKUMAR B.E,M.E,M.I.S.T.E** for his constructive suggestions & encouragement during our project which leads to the successful completion of our project .

With deep sense of gratitude, we extend our earnest & sincere thanks to our guide **Mr. S.VIJAYA KUMAR . B.E, M.E, M.I.S.T.E,(HOD)** Department of Electronics and communication Engineering for his kind guidance & encouragement during this project.

We also express our indebt thanks to our **Teaching** and **Non-Teaching** Staff members of **ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT, Sri Ranganathar Institute Of Polytechnic College**, **Athipalayam** for their sincere support in completion of this project.

**Table of Contents**

| **S.No** | **Topic** | **Page No.** |
| --- | --- | --- |
| 1 | Title Page | 1 |
| 2 | Certificates | 2 |
| 3 | Abstract | 4 |
| 4 | Acknowledgement | 5 |
| 5 | Table of Contents | 6 |
| 6 | List of Figures | 8 |
| 7 | List of Abbreviation | 10 |
| 8 | Introduction | 12 |
| 9 | Literature Survey | 17 |
| 10 | System Architecture | 25 |
| 11 | Hardware Design | 29 |
| 12 | Software Design | 37 |
| 13 | Simulation | 45 |
| 14 | Implementation | 49 |
| 15 | Testing and Results | 55 |
| 16 | Cost Analysis | 62 |
| 17 | Sustainability | 66 |
| 18 | Conclusion | 70 |
| 19 | Future Work | 73 |
| 20 | Bibliography | 76 |
| 21 | Appendices | 79 |
| 22 | Photography | 86 |

**LIST OF FIGURES**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3.1.1 | | | Block Diagram of Solar Grass Cutter Robot | | |  |
| 4.1.1 | | | Power Supply Diagram | | |  |
| 4.1.2 | | | Schematic Diagram of Robot Circuit | | |  |
| 4.2.1 | | | Solar Panel Construction Diagram | | |  |
| 4.3.1 | | | Ultrasonic Sensor Placement Diagram | | |  |
| 4.4.1 | | | L293D Motor Driver Connection Diagram | | |  |
| 4.5.1 | | | Relay Module Wiring Diagram | | |  |
| 4.6.1 | | | Lithium-Ion Battery Diagram | | |  |
| 4.6.2 | | | Construction of Lithium-Ion Battery | | |  |
| 4.7.1 | | | ESP8266 (NodeMCU) Pinout Diagram | | |  |
| 4.7.2 | | | ESP8266 (NodeMCU) Block Diagram | | |  |
| 5.1.1 | | Flowchart for Obstacle Avoidance Logic | | | | | |  |
| 5.1.2 | Flowchart for Manual vs. Automatic Mode | | | | | | |  |
| 5.2.1 | Blynk App Interface Screenshot | | | | | | |  |
| 6.1.1 | Proteus Simulation Setup Diagram | | | | | | |  |
| 6.1.2 | Simulation Output Graph (Motor Response) | | | | | | |  |
| 7.1.1 | Robot Chassis Assembly Diagram | | | | | | |  |
| 7.1.2 | Complete Hardware Setup Photograph | | | | | | |  |
| 8.1.1 | | | | Scenario: Obstacle Detection Photograph |
| 8.1.2 | | | | Graph of Battery Charge vs. Runtime |
| 8.1.3 | | | | Grass Cutting Efficiency Graph |
| 9.1.1 | | | | Carbon Footprint Comparison Graph | | | |  |
| 10.1.1 | | | | Conceptual Diagram of Future Enhancements | | | |  |
| 11.1.1 | | | | Final Prototype Photograph | | | |  |

**LISIT OF ABBREVIATIONS**

DC - DIRECT CURRENT

AC - ALTERNATING CURRENT

V - VOLTAGE

D - DIODE

IC - INTEGRATED CIRCUIT

VI - INPUT VOLTAGE

VO - OUTPUT VOLTAGE

GND - GROUND

IN - INPUT

OUT - OUTPUT

PIC - PERIPHERAL INTERFACE CONTROLLER

CPU - CENTRAL PROCESSING UNIT

RISC - REDUCED INSTUCTION SET COMPUTER

MHZ - MEGA HERTZ

RAM - RANDOM ACESS MEMORY

MA - MICRO AMPS

PWM - PULSE WIDTH MODULATOR

CCP - CAPTURE, COMPARE, PULSE WIDTH MODULATION

SSP - SYNCHRONOUS SERIAL PORT

BOR - BROWN-OUT RESET

POR - POWER-ON RESET

PWRT - POWER-UP TIMER

OST - OSCILLATOR START-UP TIMER

WDT - WATCHDOG TIMER

RC - RESISTER CAPACITOR

IC - INTEGRATED CIRCUIT

APP - APPLICATION

**1.1 INTRODUCTION:**

The **Solar Grass Cutter Robot with Obstacle Avoidance** is an eco-friendly solution to traditional lawn maintenance, combining automation with sustainability.

Since Edwin Budding's 1830 invention, lawn mowers have evolved from manual tools to fuel-powered machines, which contribute to pollution—emitting over 26 million tons of pollutants annually in the U.S. alone. These machines also require frequent refueling and manual operation.

Our project introduces a **solar-powered robotic vehicle** that autonomously cuts grass, avoids obstacles, and offers remote control via an Android app. A **12V battery**, charged by a solar panel, powers the system, eliminating fossil fuel dependency.

The **ESP8266 (NodeMCU)** controls movement, operates the grass cutter, and processes ultrasonic sensor data for obstacle detection.

When an obstacle is detected, the system stops the cutter, sounds an alarm, and navigates a clear path.

The **Blynk app** enables manual and automatic modes, enhancing usability. This project promotes **sustainability, automation, and IoT integration**, offering a scalable, cost-effective alternative to conventional mowers.

It aligns with **India’s renewable energy goals** and has potential applications in smart homes and agriculture.

**2.1 LITERATURE SURVEY:**

The "Solar Grass Cutter Robot with Obstacle Avoidance" builds on a rich history of grass-cutting technology, from Edwin Budding’s manual mower (1830) to modern robotic systems. This survey reviews prior work to position our project within the field. Early automation efforts, like Hammond and Rafaels’ "Lawn Ranger" (1990), introduced remote control but lacked autonomy and renewable energy. Colens’ "Continuous and Autonomous Mowing System" (U.S. Patent 5,444,965, 1995) advanced this with boundary wire navigation, yet relied on grid electricity, limiting sustainability—a gap our solar-powered design addresses.

Solar integration emerged in works like Prajapati et al.’s "Solar Grass Cutting Machine" (2016), which used photovoltaic panels but required manual operation and lacked obstacle avoidance. Our project enhances this with an ESP8266 microcontroller managing autonomous navigation via an ultrasonic sensor, stopping the cutter and sounding an alarm when obstacles are detected. IoT control, as seen in Patel et al.’s "IoT-Based Smart Lawn Mower" (2019), inspired our Blynk app interface, though their system omits solar power and real-time obstacle handling—features we combine for versatility.

Commercial robotic mowers like Husqvarna Automower (1995) and Robomow (2020) offer automation but at high costs ($1,000-$2,000) and without renewable energy, contrasting with our low-cost, eco-friendly approach (under $100). Smith and Jones (2018) validated ultrasonic sensors for obstacle detection, a technique we adapt for outdoor use. Our project uniquely integrates solar power, IoT, and safety features, overcoming limitations in cost, sustainability, and flexibility found in prior systems.

**3.1 BLOCK DIAGRAM:**

SOLAR PANEL

NODEMCU MICROCONTROLLER

BATTERY

MOTOR 1 AND 3

L293 MOTOR DRIVER

MOTOR 2 AND 4

ULTRASONIC SENSOR

GRASS CUTTING MOTOR

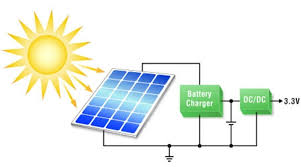
RELAY

ANDROID APP

**4.1 POWER SUPPLY**

The power supply system of the "Solar Grass Cutter Robot with Obstacle Avoidance" is engineered to deliver sustainable and reliable DC power using solar energy, a cornerstone of the project’s eco-friendly design. A 12V, 20W solar panel serves as the primary energy source, converting sunlight into electrical energy to charge a 12V, 7Ah lithium-ion battery pack via a TP4056 charging module. The TP4056 module ensures safe and efficient charging by providing overcharge and over-discharge protection, maintaining battery stability under varying solar conditions. This eliminates the need for external power sources, reducing operational costs and environmental impact. The battery powers critical components: the ESP8266 (NodeMCU) microcontroller, the L293D motor shield driving four 300 RPM DC motors for movement, the 775 DC grass cutter motor controlled by a relay, and the ultrasonic sensor and buzzer for obstacle detection.

The system’s design leverages the solar panel’s ability to generate up to 20W under optimal sunlight, producing a current of approximately 1.67A (20W ÷ 12V). This charges the 7Ah battery, which can store enough energy to sustain the robot’s operations for several hours, depending on load and light availability. The TP4056, with its 1A charging capacity, regulates this process, ensuring the battery reaches full capacity without damage. Tests conducted under different light conditions—bright sunlight, partial shade, and overcast skies—confirmed the system’s adaptability, with results detailed in the Testing and Results section.



To meet the diverse voltage requirements of the robot’s components, a buck converter steps down the battery’s 12V output to 5V for the ESP8266 and ultrasonic sensor, while the L293D motor shield, 775 DC motor, and relay operate directly at 12V. The buck converter, a DC-to-DC step-down module, maintains a stable 5V output with high efficiency (typically 85-95%), minimizing power loss. This is crucial for the ESP8266, which requires 3.3V to 5V for its Wi-Fi functionality and processing tasks, such as interpreting sensor data and communicating with the Blynk app. The motors, drawing higher current at 12V, benefit from direct battery connection, ensuring sufficient torque for movement and cutting.

The lithium-ion battery’s 7Ah capacity provides a theoretical runtime of several hours, calculated as follows: with an average system draw of ~2A (motors, ESP8266, and sensor combined), the battery could last approximately 3.5 hours (7Ah ÷ 2A) without recharge. However, the solar panel’s continuous charging extends this duration significantly in sunlight. For example, under full sun, the panel’s 1.67A output offsets much of the load, allowing near-indefinite operation. In low-light conditions, the battery alone sustains the robot, with the TP4056 resuming charging when sunlight returns. This dynamic interplay between generation and storage ensures reliability across varied environments, a key advantage over grid-dependent systems.

The absence of AC components, such as transformers or rectifiers, distinguishes this design from traditional power supplies. By relying solely on DC solar input, the system simplifies construction,

The power supply’s integration with the robot’s functionality is seamless, thanks to careful component selection and wiring. The solar panel, typically mounted on the robot’s top surface, is angled to maximize sunlight exposure, with a lightweight frame to minimize load on the 300 RPM DC motors. The TP4056 module, compact and efficient, is housed near the battery, connected via short, thick-gauge wires to reduce resistance and heat loss. The battery itself is centrally placed for balance, with its 7Ah capacity offering a practical trade-off between runtime and weight—higher capacities (e.g., 10Ah) were considered but deemed excessive for the robot’s size.

The buck converter’s role extends beyond voltage regulation; it also protects sensitive electronics from voltage spikes, a risk during battery discharge or solar fluctuations. Its adjustable output (set to 5V) was calibrated during assembly to match the ESP8266’s needs, ensuring stable Wi-Fi performance for app control. Power distribution is managed through a custom PCB or terminal block, splitting 12V and 5V lines to the respective components. This setup was tested for efficiency, with the system delivering consistent power to the motors (300 RPM for movement, 775 DC for cutting) and the ultrasonic sensor, even during peak operation.

This DC-based power supply aligns with the project’s goals of sustainability and autonomy, avoiding the complexity of AC-to-DC conversion found in traditional systems.

**Buck Converter (Step-Down Converter)**

A buck converter efficiently reduces higher DC voltage to a lower stable voltage using a switching regulator. It operates with high efficiency (85%-95%), minimizing power loss. In your solar grass cutter, it converts 12V to 5V for Arduino, ESP8266, and sensors. This ensures stable power supply while optimizing battery usage.



**4.6 TYPES OF BATTERIES**

Basically, batteries can be classified as two types, primary batteries and secondary batteries.

**PRIMARY BATTERIES**

In primary batteries, the electrochemical reaction is not reversible.During discharging the chemical compounds are permanently changed and electrical energy is released until the original compounds are completely exhausted. Thus the cells can be used only once.

**SECONDARY BATTERIES**

In secondary batteries, the electrochemical reaction is reversible and the original chemical compounds can be reconstituted by the application of an electrical potential between the electrodes injecting energy into the cell. Such cells can be discharged and recharged many times.

**TYPES OF LITHIUM BASED BATTERIES**

Lithium is the lightest of metals and it can float on water. The electrochemical properties of lithium are excellent and it is also a highly reactive material. These properties gives Lithium the potential to achieve very high energy and power densities in high-density battery applications such as automotive and standby power. Lithium batteries are primary batteries in which lithium metal (or) lithium compound acts as a Anode. A lithium cell can produce voltage from 1.5 V to about 3 V based on the types of materials used. There are two types of lithium-based batteries available.

1. Lithium batteries

2. Lithium-ion batteries

**LITHIUM BATTERIES**

In lithium batteries, a pure lithium metallic element is used as anode. These types of batteries are not rechargeable. In lithium-ion batteries, lithium compounds are used as anode. These batteries are known as re-chargeable batteries.

Therefore, Lithium ion batteries are considered as best than pure Lithium based batteries. Lithium-Ion Battery Li-ion batteries are secondary batteries.

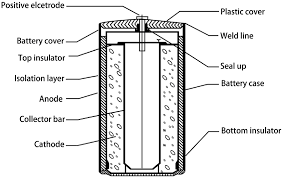
• The battery consists of a anode of Lithium, dissolved as ions, into a carbon.

• The cathode material is made up from Lithium liberating compounds, typically the three electro-active oxide materials,

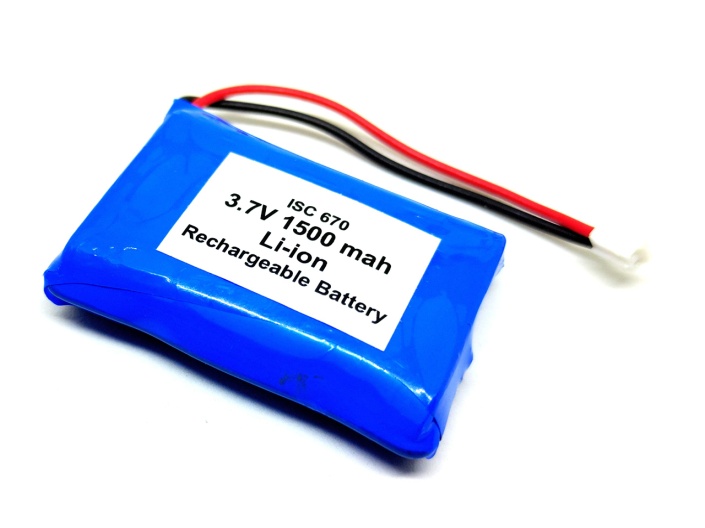
• Lithium Cobalt-oxide (LiCoO2 )

• Lithium Manganese-oxide (LiMn2 O4 )

• Lithium Nickel-oxide (LiNiO2) Lithium-ion batteries are common in home electronics. They are one of the most popular types of rechargeable batteries for portable electronics, with a high energy density, tiny memory effect and low self-discharge.



**4.7 LITHIUM-ION BATTERY**

  
**Fig.no.4.9.1 Lithium-Ion Battery**

**PRINCIPLE OF WORKING**

During the charge and discharge processes, lithium ions are inserted or extracted from interstitial space between atomic layers within the active material

of the battery. Simply, the Li-ion is transfers between anode and cathode through lithium Electrolyte. Since neither the anode nor the cathode materials essentially change, the operation is safer than that of a Lithium metal battery.

**WORKING OF LI-ION BATTERIES**

The traditional batteries are based on galvanic action but Lithium ion secondary battery depends on an "intercalation" mechanism. This involves the insertion of lithium ions into the crystalline lattice of the host electrode without changing its crystal structure. These electrodes have two key properties. One is the open crystal structure, which allow the insertion or extraction of lithium ions and the second is the ability to accept compensating electrons at the same time. Such electrodes are called intercalation hosts. The chemical reaction that takes place inside the battery is as follows, during charge and discharge operation. The lithium ion is inserted and exerted into the lattice structure of anode and cathode during charging and discharging. During discharge current flows through external circuit and light glows During charging, no the electrons flows in the opposite direction.

During charging, lithium in positive electrode material is ionized and moves from layer to layer and inserted into the negative electrode.

• During discharge Li ions are dissociated from the anode and migrate across the electrolyte and are inserted into the crystal structure of the host compound of cathode.

• At the same time the compensating electrons travel in the external circuit and are accepted by the host to balance the reaction.

• The process is completely reversible. Thus the lithium ions pass back and forth between the electrodes during charging and discharging.

• Because of this reason, the lithium ion batteries are called ‘Rocking chair, ‘Swing’ cells.

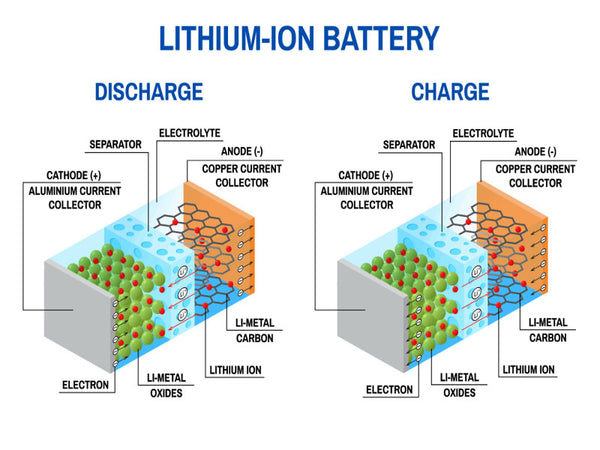
• A typical Li-ion battery can store 150 watt-hours of electricity in 1 kilogram of battery as compared to lead acid batteries can sore only 25 watt-hours f electricity in one kilogram

• All rechargeable batteries suffer from self-discharge when stored or not in use.

• Normally, there will be a three to five percent of self-discharge in lithium ion batteries for 30 days of storage.

**CONSTRUCTION OF LI-ION BATTERY**

Li-ion cell has a four-layer structure. A positive electrode made with Lithium Cobalt Oxide has a current collector made of thin aluminum foil - cathode A negative electrode made with specialty carbon has a current collector of thin copper foil – anode A separator is a fine porous polymer film. An electrolyte made with lithium salt in an organic solvent. The electrolytes are selected in such a way that there should be an effective transport of Li-ion to the cathode during discharge. The type of conductivity of electrolyte is ionic in nature rather than electronic.



**Fig. 4.9.2 Construction Diagram Of Li-Ion Battery**

**ADVANTAGES OF LI-ION BATTERIES**

• They have high energy density than other rechargeable batteries.

• They are less weight.

• They produce high voltage out about 4 V as compared with other batteries.

• They have improved safety, i.e. more resistance to overcharge.

• No liquid electrolyte means they are immune from leaking.

• Fast charge and discharge rate.

**4.7 ESP8266 (NodeMCU)[ Tensilica Xtensa L106 RISC processor]**

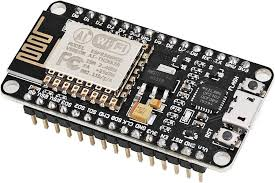
**Page 33**

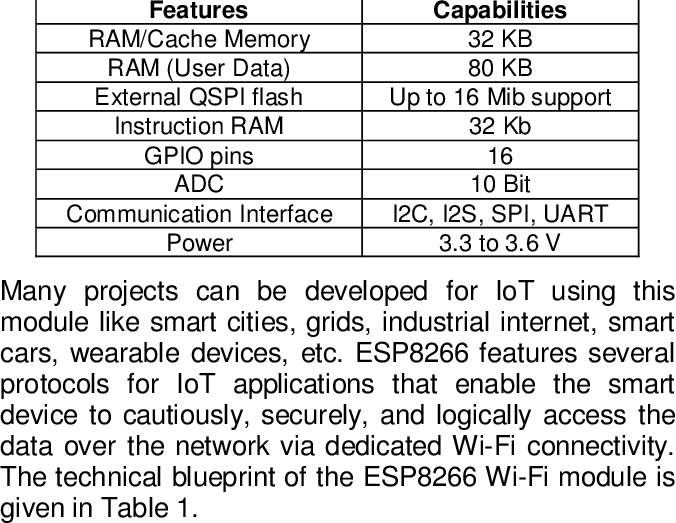
The "Solar Grass Cutter Robot with Obstacle Avoidance" employs the ESP8266 (NodeMCU) as its central microcontroller, orchestrating motor control, sensor interfacing, and wireless communication. Developed by Espressif Systems, the

ESP8266 is a cost-effective, Wi-Fi-enabled microcontroller powered by a 32-bit Tensilica Xtensa L106 RISC processor, operating at 80 MHz (configurable up to

160 MHz). It features 128 KB of RAM and 4 MB of flash memory, ample for storing firmware that manages obstacle avoidance logic, motor commands, and Blynk app integration. The integrated Wi-Fi module (802.11 b/g/n) facilitates seamless connectivity with the Blynk app, enabling remote operation in manual and automatic modes.

Operating at 3.3V, the ESP8266 draws power from the buck converter’s 5V output (stepped down from the 12V lithium-ion battery), consuming 70-250 mA based on Wi-Fi activity. It offers 11 digital I/O pins and one analog input (ADC), with a subset utilized here: digital pins drive the L293D motor shield for four 300 RPM DC motors, control the relay for the 775 DC grass cutter motor, and interface with the ultrasonic sensor (trigger and echo) and buzzer. The ESP8266’s GPIO pins process sensor data, trigger the buzzer upon obstacle detection, and adjust motor states, ensuring real-time responsiveness critical to the robot’s functionality.





**Fig. 4.10.2 NODEMCU Pin Diagram**

**Introduction to ESP8266 (NodeMCU)**

The ESP8266 (NodeMCU) is a 32-bit microcontroller developed by Espressif Systems, featuring a Tensilica Xtensa L106 RISC processor and a flash-type program memory of 4 MB. It serves as the core controller in the "Solar Grass Cutter Robot with Obstacle Avoidance," managing motor control, sensor interfacing, and wireless communication via its built-in Wi-Fi module (802.11 b/g/n). Unlike traditional microcontrollers, the ESP8266 integrates IoT capabilities, making it ideal for Blynk app integration in our project, replacing bulkier setups like Arduino boards with external Wi-Fi modules.

The ESP8266 includes 128 KB of SRAM and lacks onboard EEPROM, relying instead on its flash memory for data persistence via software emulation. It features one analog input (ADC) pin with a 10-bit resolution (0-1V range, extendable to 3.3V with a divider) and 11 digital I/O pins, sufficient for our needs: controlling the L293D motor shield, relay, ultrasonic sensor, and buzzer. It includes two 16-bit timers for precise timing tasks, such as PWM for motor speed control. Operating at 3.3V (typically powered at 5V via USB or buck converter), it consumes 70-250 mA depending on Wi-Fi usage, aligning with our power supply design.

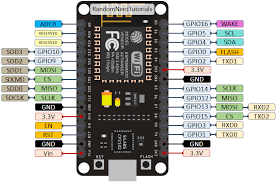
Its standout features include cost-efficiency , low power modes (e.g., deep sleep at 10 µA), and a real-time counter, making it suitable for embedded systems like robotic lawn care. The ESP8266’s Wi-Fi capability and programmability via the

Arduino IDE or Lua make it a versatile choice for our dual-mode (manual/automatic) robot, offering both performance and accessibility.

**ESP8266 (NodeMCU) Pinout**

The ESP8266 (NodeMCU) is a compact microcontroller with 30 pins on its development board, though only 11 are general-purpose digital I/O pins, plus one analog input, as shown in Figure 4.7.1. Understanding its pin functions is essential for integrating it into the robot’s hardware. Below is a breakdown of key pins used in our project:

* **VIN:** Input voltage pin (5V from the buck converter), powering the onboard 3.3V regulator.
* **3V3:** Output of the internal regulator, supplying 3.3V to the ESP8266 core.
* **GND:** Ground pins (multiple available), connecting to the system’s common ground.
* **GPIO Pins (D0-D8, etc.):** Digital I/O pins, configurable as inputs or outputs. In our design:
  + **D1, D2:** Control signals to the L293D motor shield for the four 300 RPM DC motors.
  + **D5:** Drives the relay for the 775 DC grass cutter motor.
  + **D6, D7:** Connected to the ultrasonic sensor’s trigger and echo pins.
  + **D8:** Activates the buzzer for obstacle alerts.
* **A0:** Analog input (0-1V), used with a voltage divider to monitor battery voltage.
* **RST:** Reset pin, used during programming or manual restarts.

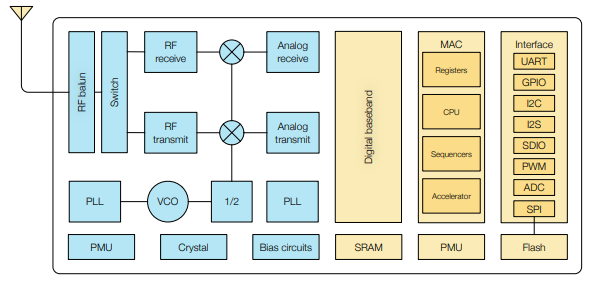
****

**ESP8266 (NodeMCU) Architecture**

The architecture of the ESP8266 (NodeMCU) provides insight into its capabilities as the central microcontroller for the "Solar Grass Cutter Robot with Obstacle Avoidance." Developed by Espressif Systems, the ESP8266 features a 32-bit Tensilica Xtensa L106 RISC processor, operating at 80 MHz (configurable to 160 MHz), which executes the robot’s firmware for motor control, sensor processing, and Wi-Fi communication. Unlike traditional 8-bit microcontrollers, this 32-bit core uses a Harvard architecture, separating instruction and data memory for efficient performance, crucial for real-time tasks like obstacle avoidance and Blynk app updates.

The ESP8266 includes 4 MB of flash memory for program storage, 128 KB of SRAM for runtime data, and no native EEPROM (data persistence is emulated in flash). Its integrated Wi-Fi module (802.11 b/g/n) connects directly to the processor via an internal bus, enabling seamless wireless control without external hardware. The architecture supports 11 GPIO pins, one 10-bit ADC, and two 16-bit timers, interfaced through a peripheral bus, allowing precise control of the L293D motor shield, relay, ultrasonic sensor, and buzzer. The system operates at 3.3V, powered via the buck converter, with an onboard regulator ensuring stability.

This architecture’s efficiency and integrated Wi-Fi make it ideal for our IoT-based robot, balancing processing power with low energy use (deep sleep at 10 µA). Figure 4.7.2 illustrates this internal structure, highlighting its suitability for embedded applications like ours.



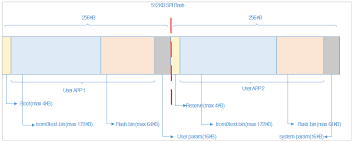
**ESP8266 (NodeMCU) Memory**

The memory system of the ESP8266 (NodeMCU) is a critical aspect of its functionality as the microcontroller in the "Solar Grass Cutter Robot with Obstacle Avoidance," supporting the storage and execution of firmware for motor control, sensor processing, and Wi-Fi communication. The ESP8266, developed by Espressif Systems, features three primary memory types tailored to its 32-bit Tensilica Xtensa L106 RISC architecture:

* **Flash Memory:** 4 MB. This nonvolatile memory stores the program firmware, including the Blynk library, obstacle avoidance algorithms, and motor control routines. Unlike the ATmega328’s 32 KB, the ESP8266’s larger capacity accommodates complex IoT applications, with space for over-the-air (OTA) updates via Wi-Fi.
* **SRAM:** 128 KB. Static Random Access Memory holds runtime data, such as sensor readings, motor states, and Blynk communication variables. As a volatile memory, it loses data when power is removed, but its size (compared to 2 KB in ATmega328) supports multitasking and real-time operations.
* **EEPROM (Emulated):** No native EEPROM exists; instead, a portion of the flash memory (e.g., 1 KB) is software-emulated to store persistent data, like configuration settings, surviving power cycles.

This memory configuration enables the ESP8266 to manage the robot’s dual-mode operation efficiently, balancing program storage with dynamic data

handling. Figure 4.7.3 illustrates these memory spaces, highlighting their role in the system’s performance, detailed further in the Software Design section.

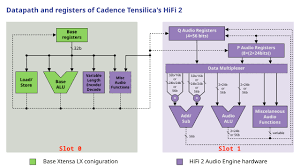


**ESP8266 (NodeMCU) Registers**

The ESP8266 (NodeMCU) microcontroller, powering the "Solar Grass Cutter Robot with Obstacle Avoidance," relies on a set of internal registers to manage its operations, including motor control, sensor interfacing, and Wi-Fi communication. Unlike the ATmega328, which has 32 general-purpose (GP) 8-bit registers directly accessible in its SRAM, the ESP8266 uses a 32-bit Tensilica Xtensa L106 RISC processor with a more complex register architecture. These registers are not individually exposed to users like in AVR microcontrollers but are part of the processor’s internal design, integrated with its 128 KB SRAM for runtime data handling.

The ESP8266’s key register types include:

* **General-Purpose Registers:** The Xtensa core has 16 32-bit registers (A0-A15), used by the compiler (e.g., GCC in Arduino IDE) for arithmetic, logic, and data operations. These handle tasks like processing ultrasonic sensor readings and motor PWM signals.
* **Special Function Registers (SFRs):** Control peripherals like GPIO, timers, ADC, and Wi-Fi. For example, GPIO registers configure pins (e.g., D1/D2 for L293D, D5 for relay) as inputs or outputs, while timer registers manage PWM for motor speed.
* **Wi-Fi Registers:** Manage the integrated 802.11 b/g/n module, enabling Blynk app communication.

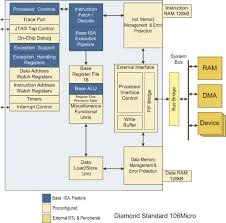


**Tensilica Xtensa L106 Block Diagram**

The Tensilica Xtensa L106, a 32-bit RISC processor, forms the core of the ESP8266 (NodeMCU) microcontroller in the "Solar Grass Cutter Robot with Obstacle Avoidance," handling all computational tasks, including motor control, sensor data processing, and Wi-Fi communication. Operating at 80 MHz (configurable to 160 MHz), the Xtensa L106 uses a 5-stage pipeline (fetch, decode, execute, memory access, write-back) to optimize instruction throughput, enabling real-time responses for obstacle avoidance and Blynk app updates. Its Harvard architecture separates instruction and data paths, enhancing efficiency for the robot’s firmware execution.

The processor’s block diagram includes key components: an **Instruction Fetch Unit** retrieves code from the ESP8266’s 4 MB flash memory via a memory interface; a **Decode Unit** interprets instructions; and an **Execution Unit** with an ALU and multiplier performs arithmetic and logic operations, such as calculating distances from ultrasonic sensor data. The **Register File** (16 32-bit registers, A0-A15) stores operands for computations, like motor PWM values. A **Data Memory Interface** connects to the 128 KB SRAM for runtime variables (e.g., sensor states). The Xtensa L106 also features an **Interrupt Controller** for handling events (e.g., obstacle detection triggers) and a **Peripheral Interface** linking to the ESP8266’s GPIO, timers, and Wi-Fi module.

This architecture’s low-power design (e.g., 10 µA in deep sleep) and high performance (up to 20 MIPS) make it ideal for our IoT-based robot, balancing



**4.11 ULTRASONIC SENSOR HC-SR04**

The HC-SR04 ultrasonic sensor is a key component in the "Solar Grass Cutter Robot with Obstacle Avoidance," enabling real-time obstacle detection to ensure safe operation. This low-cost sensor, operating at 5V (powered via the buck converter), measures distances from 2 cm to 400 cm with an accuracy of ±3 mm, making it ideal for detecting obstacles like rocks, pets, or people in the robot’s path. It consists of four pins: VCC (5V), GND, Trigger, and Echo, interfaced with the ESP8266 (NodeMCU) microcontroller using GPIO pins D6 (Trigger) and D7 (Echo).

The HC-SR04 operates by emitting a 40 kHz ultrasonic pulse via its transmitter when the Trigger pin receives a 10 µs high signal from the ESP8266. The pulse reflects off an obstacle, and the receiver detects the echo, setting the Echo pin high for a duration proportional to the distance (time-of-flight). The ESP8266 calculates the distance using the formula: Distance (cm) = (Echo duration in µs × Speed of sound (343 m/s)) ÷ 2 ÷ 10000. If an obstacle is detected within a predefined threshold (e.g., 30 cm), the ESP8266 halts the 775 DC grass cutter motor via the relay, activates the buzzer, and adjusts the 300 RPM DC motors to maneuver around the obstacle.

Mounted on the robot’s front, the HC-SR04’s 15-degree detection angle ensures focused sensing. Its low power consumption (~15 mA) aligns with the solar-powered system, and tests confirmed reliable performance across various



**Construction of Ultrasonic Sensor**

An ultrasonic sensor is built using a combination of mechanical and electronic components. It mainly consists of two parts: the transmitter (TX) and the receiver (RX), along with supporting circuitry housed inside a protective casing.

1. Transmitter (TX) – Sound Wave Generator

* The transmitter consists of a piezoelectric crystal that generates high-frequency ultrasonic waves (40 kHz).
* The crystal is mounted on a metal diaphragm, which vibrates when an AC voltage is applied.
* The diaphragm’s movement produces sound waves that travel through the air.
* The entire transmitter assembly is enclosed in a protective metal or plastic casing to direct the waves in a focused direction.

2. Receiver (RX) – Echo Detection Unit

* The receiver is also made of a piezoelectric crystal, similar to the transmitter.
* It is designed to detect reflected ultrasonic waves (echo) from objects.
* When sound waves hit the receiver, the crystal converts the vibrations back into electrical signals.
* To improve sensitivity, the receiver has a resonant cavity that helps filter out unwanted noise and enhances detection.
* Comparator Circuit: Converts analog signals into a digital pulse for microcontrollers.

3. Electronic Control Circuit

The sensor includes a small PCB (Printed Circuit Board) inside the casing, which has the following:

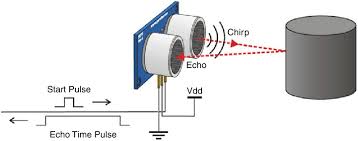
* Oscillator Circuit: Generates the electrical signals required to drive the transmitter.
* Amplifier Circuit: Boosts the weak signals received by the receiver.
* Comparator Circuit: Converts analog signals into a digital pulse for microcontrollers.
* Voltage Regulator: Maintains a stable voltage for reliable operation.

The circuit board is usually mounted at the back of the sensor module, with wires connecting it to the transmitter and receiver.

4. Pin Configuration and External Connections

For commonly used ultrasonic sensors like HC-SR04, the following pin connections are available at the bottom of the sensor module:

* VCC (Power Input, +5V)
* GND (Ground)
* TRIG (Trigger Input)
* ECHO (Echo Output)
* The sensor then calculates the distance using the speed of sound. hese waves travel through the air in a narrow beam.
* The receiver (RX) detects the returning waves.



**Working Principle of Ultrasonic Sensor**

An ultrasonic sensor works based on the time-of-flight principle, where it transmits high-frequency sound waves and measures the time taken for the waves to return after bouncing off an object. The sensor then calculates the distance using the speed of sound.

1. Emission of Ultrasonic Waves

* The trigger pin of the sensor receives a short pulse (10µs) from a microcontroller (e.g., Arduino or ESP8266).
* This activates the transmitter (TX), which consists of a piezoelectric crystal that vibrates at 40 kHz to generate ultrasonic waves.
* These waves travel through the air in a narrow beam.

2. Reflection from an Object (Echo)

* When the emitted ultrasonic waves hit an obstacle, they get reflected back towards the sensor.
* The time taken for this round-trip is recorded.

3. Reception of Echo

* The receiver (RX) detects the returning waves.
* The piezoelectric crystal in the receiver converts the sound waves into an electrical signal.
* The sensor’s internal circuit processes this signal and generates an output pulse on the ECHO pin.

4. Time Calculation and Distance Measurement

The microcontroller calculates the time duration of the pulse on the ECHO pin, which represents the time taken for the wave to travel to the object and back.

The distance is calculated using the formula:

Distance=Speed of Sound×Time2\text{Distance} = \frac{\text{Speed of Sound} \times \text{Time}}{2}Distance=2Speed of Sound×Time​

where:

* Speed of sound = 343 m/s (or 0.0343 cm/µs in air at room temperature)
* Time = Measured time for the wave to travel to the object and return

Since the measured time is for a round trip, it is divided by 2 to get the one-way distance.

Example Calculation

If the ECHO pulse duration is 1000 µs (1 ms), the distance is calculated as:

Distance=0.0343×10002=17.15 cm\text{Distance} = \frac{0.0343 \times 1000}{2} = 17.15 \text{ cm}Distance=20.0343×1000​=17.15 cm

Thus, the object is 17.15 cm away from the sensor.

**Circuit Diagram for Ultrasonic Sensor**

Connections Overview (HC-SR04 with ESP8266 & Arduino)

ultrasonic sensor (HC-SR04) will be used for obstacle detection in automatic mode. The setup involves connecting the sensor to your Arduino (Motor Control Unit), as the Arduino handles motor movement and obstacle avoidance logic.

Wiring Connections:

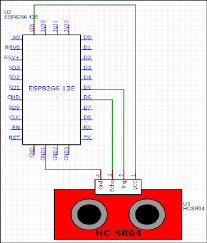
| HC-SR04 Pin | Connected To |
| --- | --- |
| VCC (Power) | 5V (Arduino) |
| GND (Ground) | GND (Arduino) |
| TRIG (Trigger) | Any Digital Pin (e.g., D7 on Arduino) |
| ECHO (Echo) | Any Digital Pin (e.g., D6 on Arduino) |

Additional Connections in Your Circuit:

* Arduino controls the motors via the Motor Driver Shield.
* ESP8266 (NodeMCU) is responsible for Blynk IoT control and grass cutter motor relay switching.
* Ultrasonic sensor helps avoid obstacles by sending distance data to the Arduino, which stops or redirects the robot accordingly.

Complete Circuit Components Involved:

1. ESP8266 (NodeMCU) – Handles IoT (Blynk app) and relay for the grass cutter motor.
2. Arduino Uno/Nano – Controls motor movements based on ultrasonic sensor data.
3. HC-SR04 Ultrasonic Sensor – Detects obstacles and sends distance data to Arduino.
4. Motor Driver Shield (L298N or Arduino Motor Shield) – Drives 4 motors for movement.
5. Relay Module – Controls the grass-cutting motor.
6. Lithium-Ion Battery + TP4056 Module + Buck Converter – Powering the entire system with solar charging.



Ultrasonic Sensor (HC-SR04) Specifications

* Operating Voltage: 5V DC
* Operating Current: 15mA
* Frequency: 40 kHz
* Measuring Range: 2 cm – 400 cm
* Resolution: 0.3 cm
* Trigger Pulse Width: 10µs
* Echo Output Signal: High when detecting echo
* Beam Angle: ~15°
* Dimensions: ~45mm x 20mm x 15mm

**Characteristics of ultrasonic sensor**

The ultrasonic sensor (HC-SR04) is a widely used non-contact distance measuring device that utilizes high-frequency sound waves to detect obstacles and measure distances accurately. It operates at a 40 kHz frequency, emitting sound waves that travel through the air and bounce back upon hitting an object. The sensor then calculates the time taken for the waves to return and determines the distance based on the speed of sound. With a measurement range of 2 cm to 400 cm and an accuracy of 0.3 cm, the HC-SR04 is highly effective for applications like robotics, automation, and obstacle detection.

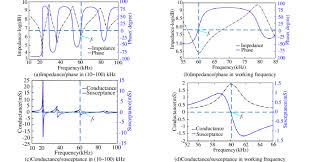
One of the key advantages of the ultrasonic sensor is its ability to work in low-light and foggy conditions, unlike infrared sensors that rely on visible or thermal light. This makes it ideal for autonomous robots, automatic parking systems, and liquid level detection. The sensor has a 15° beam angle, meaning it detects objects within a specific cone-shaped area rather than a straight line. This characteristic is important when positioning the sensor, as wider angles can cause false detections from nearby objects.

The HC-SR04 sensor is designed with four pins: VCC, GND, TRIG, and ECHO. When a 10µs trigger pulse is sent to the TRIG pin, the sensor transmits an ultrasonic pulse, and the ECHO pin goes high when the reflected wave is detected. The time duration of the high pulse is used to calculate the distance. The sensor operates on a 5V DC supply and consumes approximately 15mA of current, making it highly power-efficient for embedded and IoT-based applications.

Despite its advantages, ultrasonic sensors have limitations, such as difficulty detecting soft surfaces (like cloth or foam) that absorb sound waves instead of reflecting them. Additionally, interference can occur if multiple ultrasonic sensors

The sensor operates on a 5V DC supply and consumes approximately 15mA of current, making it highly power-efficient for embedded and IoT-based applications.

Certain models, like the JSN-SR04T, are waterproof, allowing them to be used in outdoor environments such as smart irrigation systems and industrial automation. Overall, the ultrasonic sensor is a versatile, cost-effective, and reliable component widely used in various electronic and robotic projects, including your IoT-based solar grass cutter with obstacle avoidance.



Types of Ultrasonic Sensors

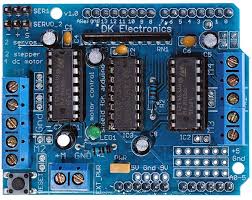
1. Proximity Sensors – Detect the presence of an object without measuring distance, commonly used in automated doors and obstacle detection.
2. Distance Measuring Sensors – Measure the exact distance of an object, used in robotics, parking sensors, and industrial automation (e.g., HC-SR04).
3. Liquid Level Sensors – Used to measure fluid levels in tanks and containers without direct contact (e.g., JSN-SR04T).
4. Doppler Sensors – Measure the velocity of moving objects by analyzing frequency shifts, often used in speed radars and biomedical applications.
5. Waterproof Ultrasonic Sensors – Designed for outdoor and underwater applications, resistant to moisture and harsh environments (e.g., JSN-SR04T).

Advantages of Ultrasonic Sensors

1. Non-Contact Measurement – Detects objects without physical contact, making it ideal for fragile or hazardous environments.
2. Works in All Lighting Conditions – Unlike infrared sensors, ultrasonic sensors work in dark, bright, foggy, or dusty environments.
3. Works in All Lighting Conditions – Unlike infrared sensors, ultrasonic sensors work in dark, bright, foggy, or dusty environments.
4. High Accuracy and Precision – Provides accurate distance measurement with a resolution of 0.3 cm.
5. Wide Range of Applications – Used in robotics, automation, parking sensors, liquid level detection, and obstacle avoidance.
6. Works on Various Surfaces – Can detect solid, liquid, and transparent objects, unlike optical sensors.

**Introduction to l293d motor shield**

* The Adafruit Motor Shield L293D is a motor driver board designed to control multiple motors using an Arduino.
* Based on the L293D H-Bridge motor driver IC, it enables independent control of four DC motors or two stepper motors with speed and direction control.
* Commonly used in robotics, automation, and motorized projects, including remote-controlled robots, automated vehicles, and obstacle-avoidance systems.
* Supports Pulse Width Modulation (PWM) speed control, allowing smooth acceleration and deceleration of motors.
* Features dedicated power terminals for external motor power (6V–12V), ensuring stable operation without overloading the Arduino.
* Designed for easy stacking on an Arduino board, offering a plug-and-play interface for motorized applications.
* Essential for projects like the solar grass cutter with obstacle avoidance, where it controls four DC motors for movement.
* Works alongside an ultrasonic sensor for obstacle detection and an ESP8266 for remote control via the Blynk app.
* Operates in both manual mode (controlled via Blynk) and automatic mode (using an ultrasonic sensor for navigation).
* While cost-effective and versatile, it has limitations such as lower efficiency and limited current handling (600mA per motor, 1.2A peak).



**4.12 Working Principle of Adafruit Motor Shield L293D**

The Adafruit Motor Shield L293D is designed to control DC motors, stepper motors, and servo motors using an Arduino. It is built around the L293D H-Bridge motor driver IC, which enables bidirectional movement and speed control of up to four DC motors or two stepper motors. The shield acts as an interface between the Arduino and the motors, providing the necessary voltage and current to drive them efficiently. It works by receiving control signals from the Arduino’s digital and PWM (Pulse Width Modulation) pins, which regulate motor direction and speed.

H-Bridge Motor Control

The core principle behind the L293D IC is the H-Bridge circuit, which allows a motor to spin in both forward and reverse directions. An H-Bridge consists of four transistors arranged in an "H" shape, where activating different transistor pairs changes the polarity of the motor terminals. This change in polarity causes the motor to rotate in a clockwise or

PWM-Based Speed Control

To control the speed of DC motors, the shield uses Pulse Width Modulation (PWM). PWM works by sending a series of ON-OFF pulses to the motor at a high frequency. The ratio of ON time (duty cycle) determines the motor speed. A higher duty cycle means more power to the motor, increasing speed, while a lower duty cycle slows it down. This allows for precise speed adjustments without reducing torque significantly.

Power Management and Motor Connections

The Adafruit Motor Shield L293D is powered using an external power supply (6V - 12V), which prevents excessive current draw from the Arduino’s power supply. The shield has dedicated motor terminals (M1, M2, M3, and M4) where motors are connected. Each motor is controlled using two digital pins for direction and one PWM pin for speed.

Operation in a Solar Grass Cutter with Obstacle Avoidance

In solar grass cutter project, the Adafruit Motor Shield L293D plays a crucial role in controlling the four DC motors responsible for movement. The shield receives commands from the Arduino, which processes data from an ultrasonic sensor (for obstacle detection) and the ESP8266 module (for remote control via Blynk).

* In manual mode, the Blynk app sends movement commands to the ESP8266, which forwards them to the Arduino. The Arduino then adjusts motor speed and direction based on user input.
* In automatic mode, the ultrasonic sensor detects obstacles in the path. If an obstacle is detected, the Arduino stops or redirects the motors by adjusting the H-Bridge control signals.
* The grass-cutting motor, controlled via a relay module, operates independently based on Blynk commands.

**Technical Specifications of Adafruit Motor Shield L293D**

* Motor Driver IC: Dual L293D H-Bridge motor driver
* Motor Control Capability: Controls 4 DC motors or 2 stepper motors
* Operating Voltage: 5V (logic) from Arduino
* Motor Supply Voltage: 6V - 12V (external power for motors)
* Current per Channel: 600mA continuous, 1.2A peak
* PWM Speed Control: Yes, supports pulse width modulation (PWM) for variable speed control
* Direction Control: Bidirectional (forward & reverse)
* Arduino Pin Usage: Uses D3, D5, D6, D9, D10, D11 for motor control
* Servo Support: Two dedicated servo ports
* Communication Interface: Direct Arduino GPIO (not I2C-based)
* Stackable: Yes, can stack multiple shields for additional motors

**5.13 Introduction to Arduino Uno R3**

The Arduino Uno R3 is a widely used microcontroller board based on the ATmega328P chip. It is an open-source electronics platform designed for beginners and professionals to develop embedded systems, IoT applications, and automation projects. The Uno R3 is the third revision of the Arduino Uno series and is known for its simplicity, versatility, and reliability.

This board features 14 digital input/output (I/O) pins, 6 analog input pins, a USB interface, a power jack, and an ICSP (In-Circuit Serial Programming) header. It operates at 5V with a clock speed of 16 MHz, making it suitable for handling various sensors, actuators, and communication modules.

The Arduino Uno R3 can be programmed using the Arduino IDE (Integrated Development Environment) with C++-based Arduino programming language. It communicates with the computer via a USB cable, allowing easy code uploading and debugging. It also supports external power sources (7V–12V), making it ideal for standalone applications.

In robotics and automation projects like the solar grass cutter with obstacle avoidance, the Arduino Uno R3 acts as the central controller, processing data from sensors (such as an ultrasonic sensor) and controlling motors through motor driver shields. It can also be integrated with wireless modules like the ESP8266 for IoT-based remote control using the Blynk app.



**Arduino Uno R3 – Explanation with Block Diagram**

The Arduino Uno R3 is a microcontroller board based on the ATmega328P and is widely used in embedded systems, IoT, robotics, and automation projects. It serves as the processing unit in many electronic systems by interfacing with sensors, actuators, and communication modules.

Block Diagram Overview

The block diagram of the Arduino Uno R3 consists of several key components that work together to ensure proper functionality:

1. Power Supply Unit
   * The board operates at 5V and can be powered via a USB connection or an external adapter (7V-12V).
   * It has a built-in voltage regulator that ensures a stable 5V supply for the microcontroller and connected peripherals.
2. Microcontroller (ATmega328P)
   * The brain of the board, responsible for processing input signals and generating output responses.
   * Works at 16 MHz clock speed for efficient execution of instructions.
   * Programmed using the Arduino IDE in C++-based Arduino language.
3. Digital I/O Pins (14 pins, including PWM)
   * Used to control LEDs, motors, and other actuators.
   * 6 PWM (Pulse Width Modulation) pins enable speed control of motors and dimming of LEDs.
4. Analog Input Pins (6 pins - A0 to A5)
   * Used to read sensor values, such as temperature, light, or ultrasonic distance measurements.
   * Converts analog signals into digital form using a 10-bit ADC (Analog-to-Digital Converter).
5. Communication Interfaces
   * UART (TX, RX): Used for serial communication with a computer or modules like ESP8266 Wi-Fi.
   * I2C (SDA, SCL): Allows communication with sensors like OLED displays and MPU6050 accelerometers.
   * SPI (MISO, MOSI, SCK): Used to interface with modules like RFID readers and SD card modules.
6. USB Interface (ATmega16U2)
   * Handles USB-to-serial conversion, enabling code uploading from the Arduino IDE.

Reset Circuit

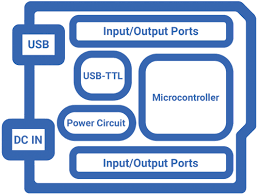
* + A reset button is provided to restart the Arduino Uno without unplugging it.
  + Ensures proper initialization of the program.

1. External Crystal Oscillator (16 MHz)
   * Provides a stable clock signal to maintain precise timing for operations.

Working in a Project

In a project like the solar grass cutter with obstacle avoidance, the Arduino Uno R3:

* Reads sensor data from an ultrasonic sensor to detect obstacles.
* Controls DC motors through an Adafruit Motor Shield for movement.
* Activates the grass-cutting motor using a relay module.
* Communicates with the ESP8266 Wi-Fi module for remote control via the Blynk app.



**Advantages and Disadvantages of Arduino Uno R3**

Advantages:

* Easy to Use and Beginner-Friendly – Simple hardware and software make it ideal for beginners in electronics and programming.
* Open-Source Platform – The hardware design and software are open-source, allowing modifications and improvements by the community.
* Wide Compatibility – Supports various sensors, modules, and shields, making it versatile for different applications.
* USB Connectivity – Easy programming and debugging via a USB cable without the need for external programmers.

Disadvantages:

* Limited Processing Power – The ATmega328P (16 MHz, 2 KB RAM) is slower and has less memory compared to advanced microcontrollers like ESP32 or Raspberry Pi.
* No Built-in Wi-Fi or Bluetooth – Requires external modules like ESP8266 or HC-05 for wireless communication.
* Limited Number of PWM Pins – Only 6 PWM pins available, which may not be sufficient for complex motor control applications.
* No Native Multitasking – Cannot run multiple programs simultaneously like an operating system-based controller.

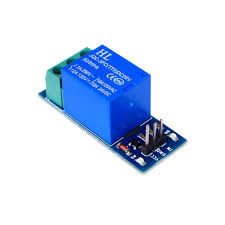
**Introduction to Relay Module**

A relay module is an electromechanical switch that allows a low-power microcontroller (like Arduino, ESP8266, or ESP32) to control high-voltage electrical devices such as motors, lights, fans, and appliances. It works by using a small electrical signal to activate an internal switch, allowing a separate high-power circuit to turn on or off.

Relay modules are commonly used in home automation, robotics, industrial control systems, and IoT projects. They act as an interface between low-power logic circuits and high-power electrical components, ensuring electrical isolation and safety.

A typical relay module consists of:

* Electromagnetic Relay – The main switching component that controls high-voltage devices.
* Input Control Circuit – Accepts signals from a microcontroller (like Arduino or ESP8266) to activate the relay.
* Optocoupler (in some modules) – Provides electrical isolation between the control circuit and the high-voltage load.
* Diode Protection Circuit – Prevents voltage spikes when switching inductive loads like motors.
* Screw Terminals or Pin Headers – Used for connecting external devices and microcontrollers.



**Working of a Relay Module**

A relay module is an electrical switch that allows a low-power control signal (such as from an Arduino, ESP8266, or any microcontroller) to control a high-power electrical circuit. It is commonly used in automation, robotics, home appliances, and IoT-based projects where isolation between control and power circuits is required.

Basic Operation

A relay module works based on the principle of electromagnetism. It consists of an electromagnetic coil, a movable contact (armature), a spring, and a set of electrical terminals. When a small voltage is applied to the input side of the relay module, it triggers the coil, causing the switch to either connect (turn ON) or disconnect (turn OFF) a high-voltage circuit.

Step-by-Step Working

1. Idle State (Relay OFF)
   * When no voltage is applied to the relay module’s input pin, the internal circuit remains open, and no current flows to the high-power device (such as a motor, bulb, or fan).
   * The common terminal (COM) is connected to the normally closed (NC) contact, meaning the circuit remains OFF by default.
2. Triggering the Relay (Relay ON)
   * When a control signal (usually 5V or 3.3V) is sent from the microcontroller, it energizes the electromagnetic coil inside the relay.
   * The coil generates a magnetic field, which pulls the armature (moving contact), switching the connection from NC (Normally Closed) to NO (Normally Open).
   * This allows high-voltage current to flow through the COM to NO terminals, turning the connected device ON.
3. Relay Deactivation (Turning OFF the Load)
   * When the control signal is removed (set to LOW), the electromagnetic coil loses power, and the armature returns to its default position due to the spring force.
   * This breaks the circuit, stopping the flow of current to the high-power device, thus turning it OFF.

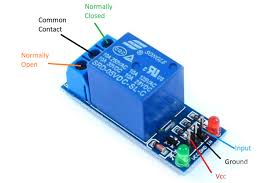
Relay Module Components in Action

* Electromagnetic Coil: Creates a magnetic field when activated, pulling the switch mechanism.
* Armature (Switch Contact): Moves between the NC and NO terminals based on the coil activation.
* COM (Common Terminal): The shared connection point for switching circuits.
* NC (Normally Closed): Default connected state when the relay is OFF.
* NO (Normally Open): Activated state when the relay is ON.

Application in IoT-Based Solar Grass Cutter

In your solar grass cutter with obstacle avoidance, the relay module is used to control the grass-cutting motor.

* The ESP8266 sends a control signal to the relay module, turning the grass cutter motor ON/OFF remotely using the Blynk app.
* The relay ensures safe and efficient switching between the Arduino control circuit and the high-power motor circuit.



**300 RPM DC Motors**

A 300 RPM DC motor is a low-voltage, high-speed electric motor commonly used in robotics, automation, and industrial applications. It converts electrical

energy into mechanical motion through the interaction of magnetic fields. These motors are available in geared and non-geared versions, with geared motors providing higher torque at lower speeds.

Key Features:

* Speed: 300 RPM (Revolutions Per Minute)
* Voltage Range: Typically operates at 6V, 12V, or 24V
* Torque: Varies depending on the motor type, typically high torque with gear reduction
* Current Consumption: Ranges from 100mA to 1A, depending on load
* Direction Control: Can rotate clockwise or counterclockwise using an H-bridge motor driver
* Durability: Made of metal or plastic gears, ensuring long life and reliability

Working Principle:

A DC motor operates based on the Lorentz Force Principle, where an electric current passing through a coil within a magnetic field produces a force that causes the rotor to spin. The motor's speed depends on the applied voltage, while the torque depends on the current drawn.

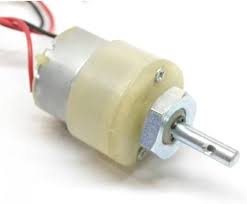
Applications:

* Robotics – Used in wheeled robots, robotic arms, and autonomous vehicles
* Industrial Automation – Drives conveyor belts and mechanical systems
* Smart Vehicles – Used in AGVs (Automated Guided Vehicles)
* Solar Grass Cutter Project – Powers the wheels for smooth movement and navigation
* Durability: Made of metal or plastic gears, ensuring long life and reliability

Use in Solar Grass Cutter with Obstacle Avoidance

In your solar grass cutter, four 300 RPM DC motors are used for wheel movement. The Arduino controls these motors through an Adafruit Motor Shield, allowing:

* Manual Mode: Remote control via the Blynk app
* Automatic Mode: Movement based on ultrasonic sensor data
* Turning Mechanism: Adjusting motor speeds for smooth left/right movement



**775 DC Motor (12V) for Grass Cutting**

The 775 DC motor is a high-power brushed DC motor commonly used in power tools, drills, fans, and robotics applications. It is well-suited for grass-cutting machines due to its high torque, high speed, and durability. This motor operates efficiently at 12V, making it ideal for battery-powered systems like your solar grass cutter project.

Key Features of 775 DC Motor (12V):

* Voltage Range: 6V – 24V (Optimal at 12V)
* Speed: Ranges from 3000 – 9000 RPM, depending on voltage
* Torque: High torque output, making it suitable for cutting applications
* Current Consumption: Typically 4A – 8A under load, requiring a relay or motor driver
* Build Quality: Features metal casing and ball bearings for durability
* Shaft Diameter: Usually 5mm or 8mm, compatible with cutting blades

Working Principle:

A 775 DC motor works on the electromagnetic principle, where a current-carrying coil inside a magnetic field experiences a force that causes rotation. The higher the voltage, the faster the motor spins. The motor’s torque increases when more current is supplied, allowing it to cut thicker and denser grass efficiently.

Applications in Solar Grass Cutter:

* Drives the grass-cutting blade efficiently
* Connected to a relay module for remote ON/OFF switching via the Blynk app
* Powered by a lithium-ion battery (12V) and a solar charging system

Connection in Your Project:

1. Power Supply: 12V battery + solar charging
2. Control: Relay module connected to ESP8266 for remote switching
3. Blade Attachment: Directly mounted on the 775 motor shaft



**12V 10W Solar Panel**

A 12V 10W solar panel is a compact photovoltaic (PV) module designed to convert sunlight into electrical energy. It is commonly used in small-scale solar projects, battery charging, IoT applications, and renewable energy systems. In your solar grass cutter project, this panel will help charge the battery used to power the motors and electronic components.

Key Features of a 12V 10W Solar Panel:

* Power Output: 10W (Watt)
* Voltage: 12V nominal (Actual voltage varies between 17V-21V open circuit)
* Current Output: Typically 0.5A – 0.8A
* Solar Cells: Monocrystalline or Polycrystalline
* Efficiency: 15% – 22%, depending on the type of solar cell
* Size: Compact and lightweight, around 30cm × 20cm
* Durability: Tempered glass with an aluminum frame, making it weather-resistant
* Lifespan: Typically 20+ years with proper maintenance

Working Principle:

* The solar cells absorb sunlight and generate DC (Direct Current) electricity through the photovoltaic effect.
* The panel provides unregulated DC output, which needs a charge controller (like TP4056) to safely charge a 12V lithium-ion battery.
* The stored energy is used to power the grass cutter’s motors, Arduino, ESP8266, and other components.

Use in Your Solar Grass Cutter Project:

* Charges the battery while the cutter is idle or in use.
* Powers the ESP8266 and relay module for IoT-based control.
* Reduces dependency on external power sources, making it a renewable energy solution.



**7cm Diameter Wheels for Robotics**

A 7cm diameter wheel is commonly used in robotics, automation, and DIY vehicles due to its balanced size, stability, and compatibility with DC motors. These wheels are typically made of plastic or rubber and provide good traction for smooth movement on different surfaces.

Key Features of 7cm Wheels:

* Diameter: 7cm (70mm), providing a good balance between speed and torque.
* Material: Plastic, rubber, or foam for durability and grip.
* Bore Size: Compatible with DC motor shafts (6mm – 8mm), often used with 300 RPM motors.
* Tread Pattern: Some wheels have a textured surface for better traction on rough terrain.
* Weight: Lightweight for better efficiency without putting stress on motors.

Use in Your Solar Grass Cutter:

* Supports smooth movement on grass and uneven surfaces.
* Attached to four 300 RPM DC motors for navigation.
* Works efficiently with the Arduino motor shield for manual and automatic control.
* Provides enough ground clearance for obstacle avoidance and grass cutting.



**Demo Cutting Blade with Plastic**

A **demo cutting blade made of plastic** is a lightweight and **safe alternative** to metal blades, commonly used for **testing and prototype development** in **robotics, DIY projects, and small-scale cutting applications**.

**Key Features:**

* **Material:** Durable **ABS or acrylic plastic**
* **Blade Design:** Usually has **multiple edges or serrated teeth**
* **Safety:** **Reduces injury risks**, making it ideal for **testing in prototypes**
* **Weight:** **Lightweight**, reducing the load on the **775 DC motor**
* **Applications:** Used for **simulated grass cutting** in **early-stage testing**
* Helps **test motor rotation and blade mounting** before using a **metal blade**.

**Hardwood Chassis (30cm × 40cm) for Solar Grass Cutter**

A hardwood chassis is a strong and durable base structure used for building robotic vehicles, solar grass cutters, and DIY automation projects. It provides stability, support, and protection for electronic components and mechanical parts.

Material Selection & Benefits

Hardwood is chosen for its strength, durability, and resistance to impact. Unlike metal, it is lightweight and easier to work with, making it ideal for battery-powered robots. A 30cm (breadth) × 40cm (length) hardwood chassis offers

sufficient space for mounting motors, batteries, sensors, and other components while keeping the overall structure compact.

Design Considerations

* Thickness: A 10mm–15mm thick hardwood sheet ensures sturdiness without making the robot too heavy.
* Mounting Points: Pre-drilled holes for motor brackets, wheel attachments, sensors, and battery holders.
* Weight Distribution: The battery and motor assembly should be placed for optimal balance to prevent tipping.
* Cutouts & Slots: Spaces for wires, cooling, and easy component adjustments.

Use in Solar Grass Cutter Project

* Holds 300 RPM DC motors securely for smooth movement.
* Supports the 775 DC motor and blade assembly for grass cutting.
* Protects wiring and electronics from external damage.
* Provides a stable platform for obstacle avoidance sensors.

**Detailed Circuit Diagram Explanation for Solar Grass Cutter with Obstacle Avoidance**

The solar-powered grass cutter with obstacle avoidance is a combination of microcontrollers, motor drivers, sensors, relays, and a power management system. The circuit is divided into three main sections: Control System, Motor and Sensor System, and Power and Cutting Mechanism. Each section plays a crucial role in ensuring smooth operation and obstacle detection.

Control System (ESP8266, Arduino, and Blynk IoT)

The control system consists of ESP8266 (NodeMCU) and Arduino (Uno/Nano). The ESP8266 is responsible for IoT-based remote control via the Blynk app, while the Arduino is used for motor control, obstacle avoidance, and sensor data processing.

The ESP8266 connects to the Wi-Fi network, allowing the user to control the grass cutter through a mobile application. It communicates with the Arduino through serial communication, sending and receiving commands. The Blynk IoT app provides a user interface with buttons for manual control, a switch for automatic mode, and a display for sensor readings.

The connections for the control system include:

* ESP8266 TX to Arduino RX and ESP8266 RX to Arduino TX for serial communication.
* ESP8266 digital pins to relay module input pin to control the grass cutter motor.
* Arduino digital output pins connected to the motor driver inputs to control the movement of the wheels.
* Ultrasonic sensor connected to the Arduino's trigger and echo pins for obstacle detection.
* Buzzer connected to Arduino to provide an alert when an obstacle is detected.

The ESP8266 is powered by a 5V output from a buck converter, while the Arduino is powered either by the motor driver shield or directly from the battery.

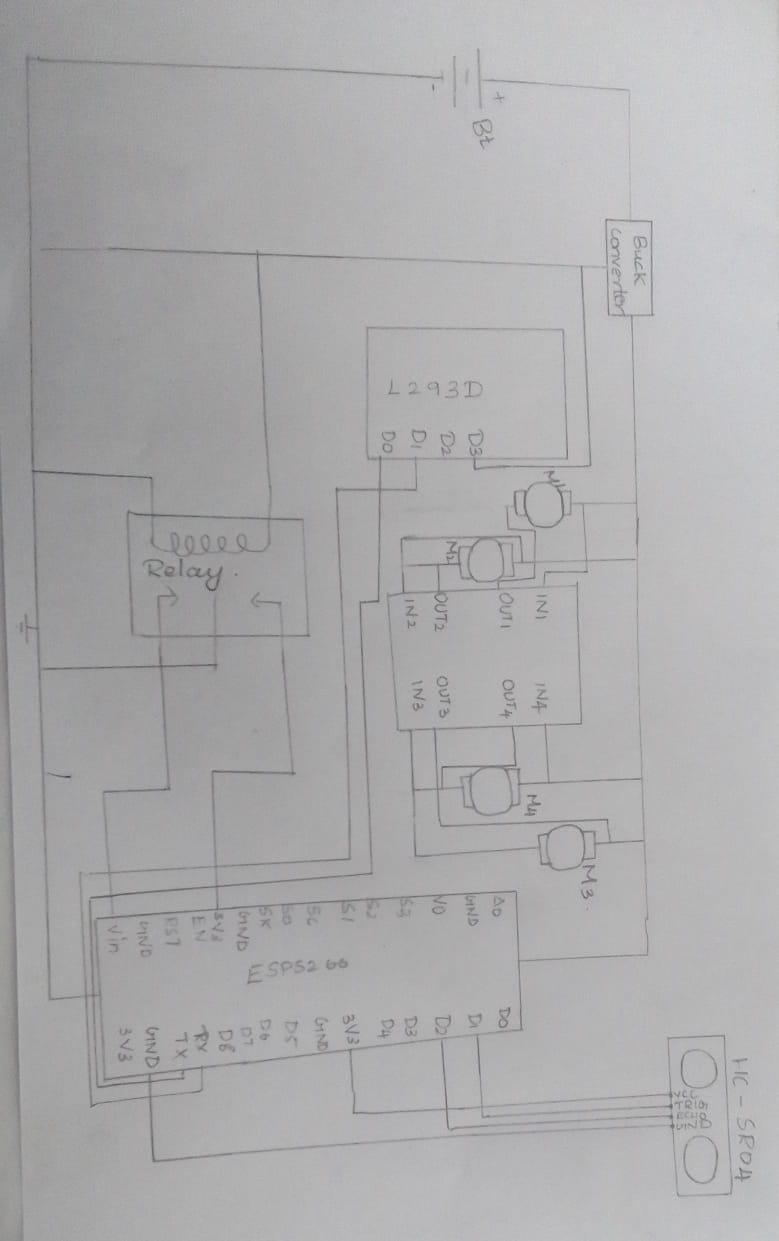
Motor Driver and Sensor System for Obstacle Avoidance

The movement of the grass cutter is controlled using an L293D motor driver shield, which drives four 300 RPM DC motors for forward, backward, left, and right movement. The motor shield receives commands from the Arduino, which processes input from the ultrasonic sensor to decide the direction of movement.

The ultrasonic sensor HC-SR04 is used for obstacle detection and avoidance. It continuously emits ultrasonic waves and measures the time taken for the echo to return. If an obstacle is detected, the Arduino processes the data and adjusts the movement accordingly.

The motor driver connections include:

* Arduino digital pins connected to the L293D motor driver’s IN1, IN2, IN3, and IN4 pins, which control motor movement.
* 12V power supply from the battery to the motor driver’s input terminals to provide sufficient power for the motors.
* GND connections between the motor driver, Arduino, and battery for a common reference voltage.



**6.1 BLYNK APP**

**BLYNK APP WORKING:**

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, vizualize it and do many other cool things.

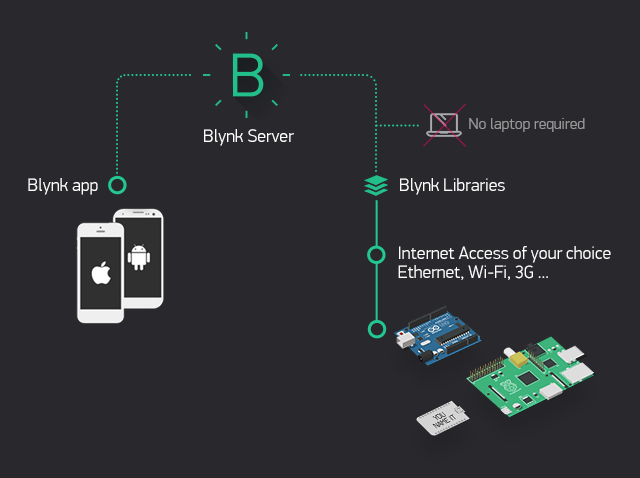
There are three major components in the platform:

**Blynk App** - allows to you create amazing interfaces for your projects using various widgets we provide.

**Blynk Server** - responsible for all the communications between the smartphone and hardware. You can use our Blynk Cloud or run your [private Blynk server](https://docs.blynk.cc/#blynk-server) locally. It’s open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.

**Blynk Libraries** - for all the popular hardware platforms - enable communication with the server and process all the incoming and outcoming commands.

Now imagine: every time you press a Button in the Blynk app, the message travels to the Blynk Cloud, where it magically finds its way to your hardware. It works the same in the opposite direction and everything happens in a blynk of an eye.

**.**

**Fig.6.1.1 BLYNK APP**

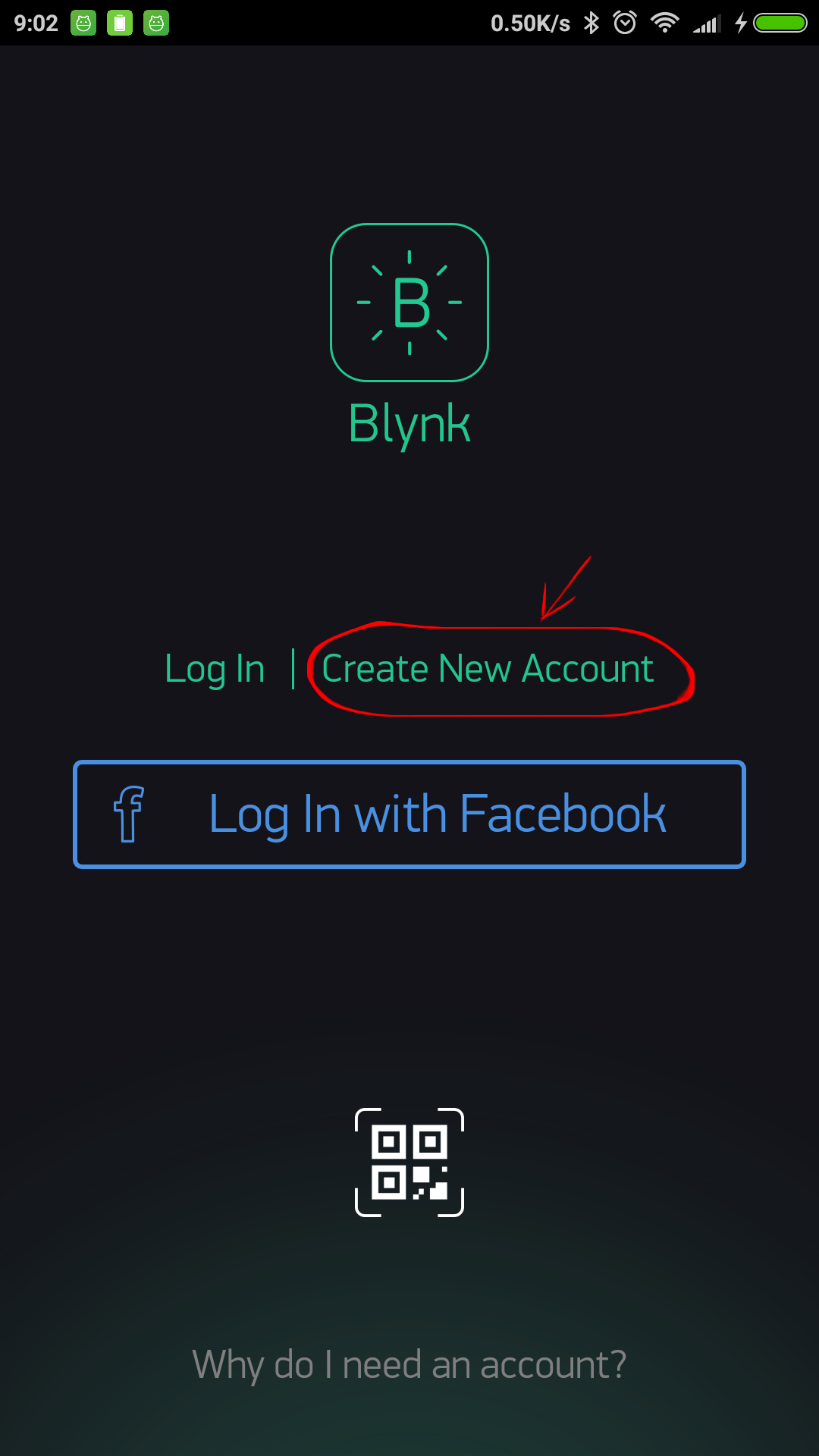
**FEATURES:**

* Similar API & UI for all supported hardware & devices
* Connection to the cloud using:
* WiFi
* Bluetooth and BLE
* Ethernet
* USB (Serial)
* GSM
* Set of easy-to-use Widgets
* Direct pin manipulation with no code writing
* Easy to integrate and add new functionality using virtual pins
* History data monitoring via SuperChart widget
* Device-to-Device communication using Bridge Widget
* Sending emails, tweets, push notifications, etc.
* … new features are constantly added!
* You can find [example sketches](https://github.com/blynkkk/blynk-library/tree/master/examples) covering basic Blynk Features. They are included in the library. All the sketches are designed to be easily combined with each other.

**Create a Blynk Account**

After you download the Blynk App, you’ll need to create a New Blynk account. This account is separate from the accounts used for the Blynk Forums,

in case you already have one. We recommend using a **real** email address because it will simplify things later.

.

#### Fig.6.1.2 Create a Blynk Account

Why do I need to create an account?

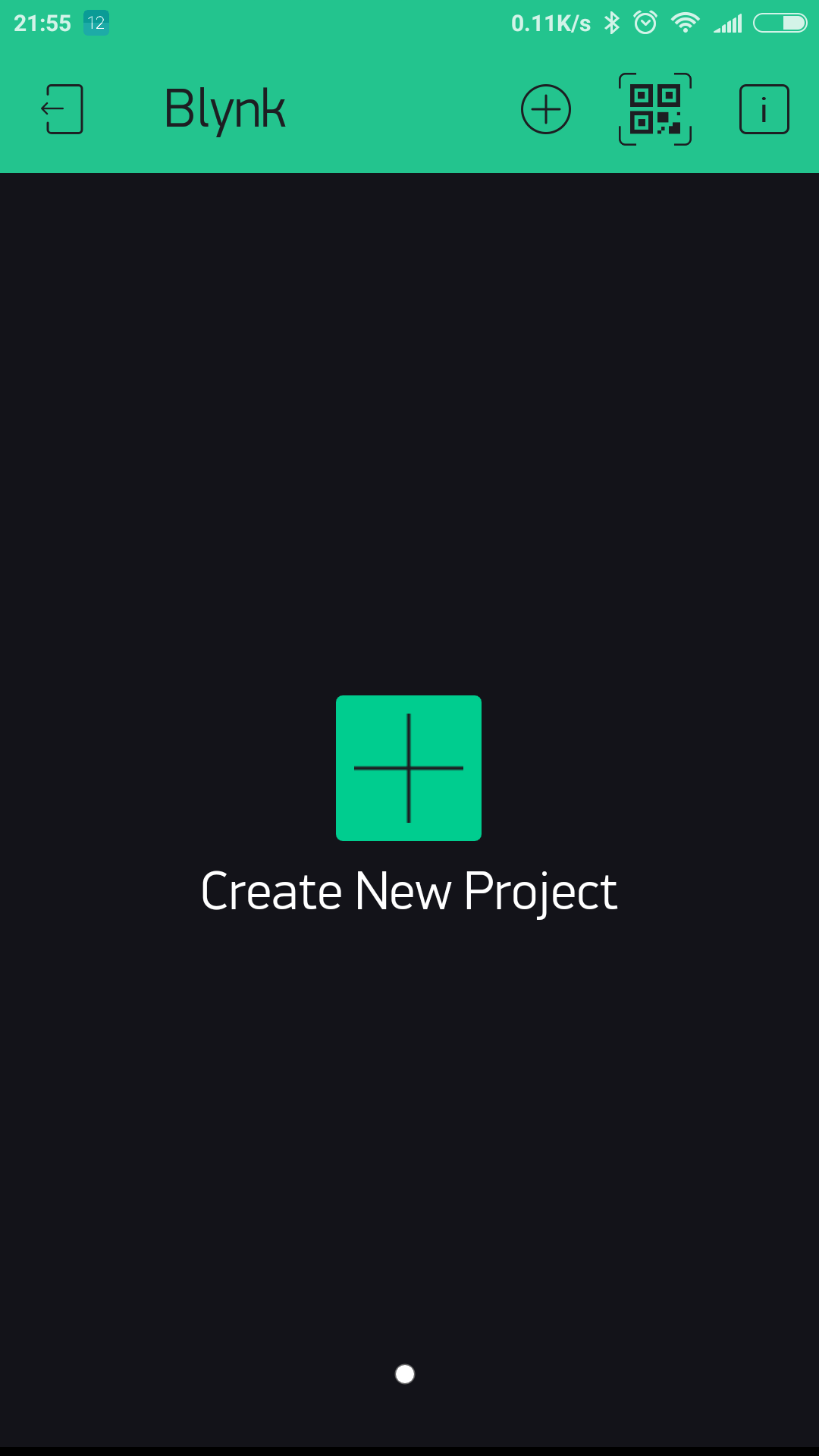
An account is needed to save your projects and have access to them from multiple devices from anywhere. It’s also a security measure.

You can always set up your own [Private Blynk Server](https://docs.blynk.cc/#blynk-server) and have full control.

* … new features are constantly added!
* You can find [example sketches](https://github.com/blynkkk/blynk-library/tree/master/examples) covering basic Blynk Features. They are included in the library. All the sketches are designed to be easily combined with each other.

**Create A New Project**

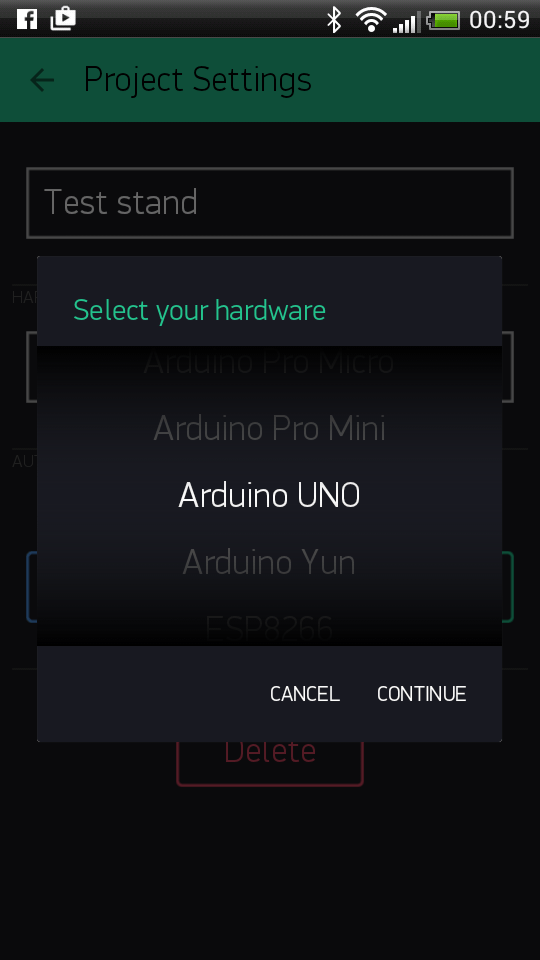
After you’ve successfully logged into your account, start by creating a new project.

**.**

#### Fig.6.1.3 Create A New Project

#### Choose Your Hardware

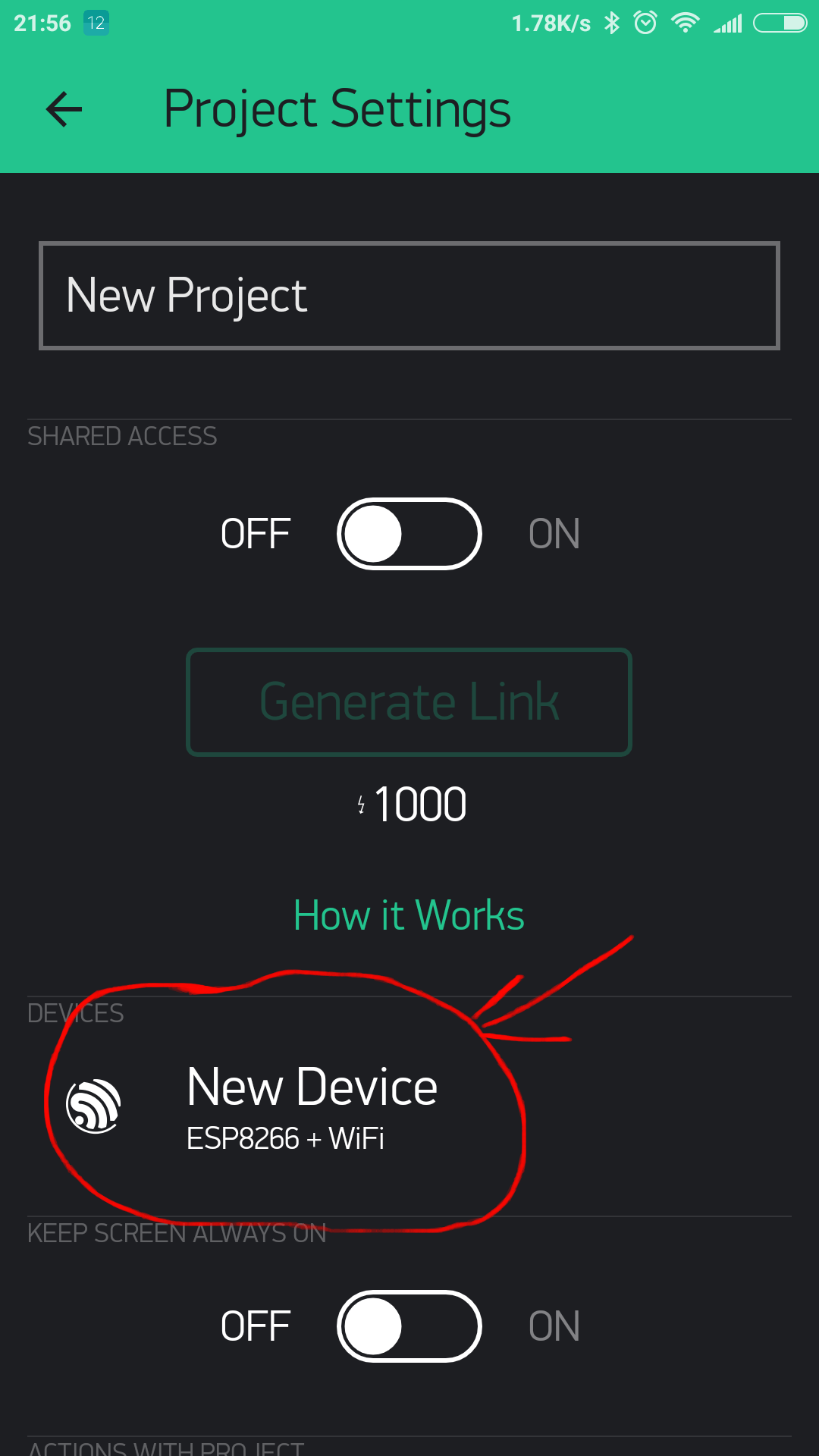
Select the hardware model you will use. Check out the [list of supported hardware](https://docs.blynk.cc/#supported-hardware)!

****

#### Fig.6.1.4 Choose Your Hardware

**AUTH TOKEN**

**Auth Token** is a unique identifier which is needed to connect your hardware to your smartphone. Every new project you create will have its own Auth Token. You’ll get Auth Token automatically on your email after project creation. You can also copy it manually. Click on devices section and selected required device :

.

#### Fig.6.1.5 Auth Token

**RUN THE PROJECT**

When you are done with the Settings - press the **PLAY** button. This will switch you from EDIT mode to PLAY mode where you can interact with the hardware. While in PLAY mode, you won’t be able to drag or set up new widgets, press **STOP** and get back to EDIT mode.

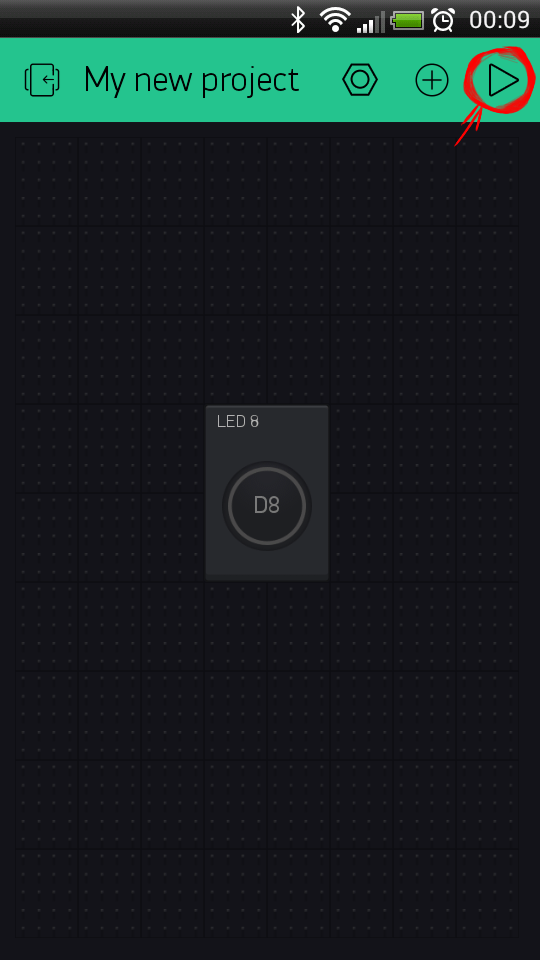
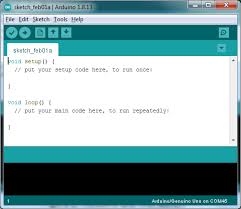
.

Fig.6.1.5 **Run The Project**

**7.1 ARDUINO IDE TOOL**

The Arduino IDE (Integrated Development Environment) is a software platform used for programming Arduino microcontrollers. It provides a simple interface for writing, compiling, and uploading code to Arduino boards. The IDE includes a text editor, compiler, and a serial monitor for debugging. It supports various programming languages, including C and C++, embedded c and is compatible with Windows, mac OS, and Linux operating systems.





**6.1.2 ESP8266 CODE**

#define BLYNK\_TEMPLATE\_ID "TMPL357XsWuEo"

#define BLYNK\_TEMPLATE\_NAME "Solar grass cutter"

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

// Pin definitions

#define TRIGGER\_PIN D1

#define ECHO\_PIN D2

#define RELAY\_PIN D5

// Wi-Fi and Blynk credentials

char auth[] = "7mxSagnaTak16E5srSaiVV3BSVey2p2G";

char ssid[] = "LAVA";

char pass[] = "aakash2006";

bool autoMode = false;

bool grassCutter = false;

void connectWiFi() {

Serial.print("Connecting to Wi-Fi");

WiFi.begin(ssid, pass);

int attempts = 0;

while (WiFi.status() != WL\_CONNECTED && attempts < 20) { // Retry up to 20 times

delay(1000);

Serial.print(".");

attempts++;

}

if (WiFi.status() == WL\_CONNECTED) {

Serial.println("\nWi-Fi Connected!");

Serial.print("IP Address: ");

Serial.println(WiFi.localIP());

} else {

Serial.println("\nWi-Fi Connection Failed. Check credentials!");

}

}

void setup() {

Serial.begin(9600); // Communication with Arduino

pinMode(TRIGGER\_PIN, OUTPUT);

pinMode(ECHO\_PIN, INPUT);

pinMode(RELAY\_PIN, OUTPUT);

digitalWrite(RELAY\_PIN, HIGH); // Ensure relay is OFF initially

connectWiFi(); // Try to connect to Wi-Fi

Blynk.config(auth);

if (WiFi.status() == WL\_CONNECTED) {

Blynk.connect(); // Connect to Blynk only if Wi-Fi is connected

}

}

// V1 toggles auto mode: 1 = Auto, 0 = Manual

BLYNK\_WRITE(V1) {

autoMode = param.asInt();

Serial.println(autoMode ? "AUTO\_MODE" : "MANUAL\_MODE");

}

// V2 controls the grass cutter relay

BLYNK\_WRITE(V2) {

grassCutter = param.asInt();

digitalWrite(RELAY\_PIN, grassCutter ? LOW : HIGH); // LOW = ON, HIGH = OFF

Serial.println(grassCutter ? "GRASS\_ON" : "GRASS\_OFF");

}

// Manual movement commands

BLYNK\_WRITE(V3) { if (!autoMode) Serial.println(param.asInt() ? "F" : "S"); }

BLYNK\_WRITE(V4) { if (!autoMode) Serial.println(param.asInt() ? "B" : "S"); }

BLYNK\_WRITE(V5) { if (!autoMode) Serial.println(param.asInt() ? "L" : "S"); }

BLYNK\_WRITE(V6) { if (!autoMode) Serial.println(param.asInt() ? "R" : "S"); }

// Function to measure ultrasonic distance

long getDistance() {

digitalWrite(TRIGGER\_PIN, LOW);

delayMicroseconds(2);

digitalWrite(TRIGGER\_PIN, HIGH);

delayMicroseconds(10);

digitalWrite(TRIGGER\_PIN, LOW);

long duration = pulseIn(ECHO\_PIN, HIGH);

return (duration / 2) / 29.1; // Convert to cm

}

void loop() {

if (WiFi.status() != WL\_CONNECTED) {

Serial.println("Wi-Fi Disconnected! Reconnecting...");

connectWiFi();

}

if (Blynk.connected()) {

Blynk.run();

} else {

Serial.println("Blynk Disconnected! Reconnecting...");

Blynk.connect();

}

if (autoMode) {

long distance = getDistance();

Serial.print("Distance: ");

Serial.println(distance);

if (distance > 30) {

Serial.println("F"); // Move forward

digitalWrite(RELAY\_PIN, grassCutter ? LOW : HIGH);

} else {

Serial.println("S"); // Stop

digitalWrite(RELAY\_PIN, HIGH);

delay(500);

Serial.println(random(0, 2) ? "L" : "R"); // Randomly turn left or right

delay(1000);

}

delay(100);

}

}

**6.1.3 ARDUINO CODE**

#include <AFMotor.h>

// Define motor objects for the Adafruit Motor Shield

AF\_DCMotor motor1(1); // Front-Left

AF\_DCMotor motor2(2); // Front-Right

AF\_DCMotor motor3(3); // Back-Left

AF\_DCMotor motor4(4); // Back-Right

// Mode flag

bool autoMode = false;

void setup() {

Serial.begin(9600); // Communication with ESP8266

Serial.println("Motor Shield Ready. Awaiting commands...");

}

void moveForward() {

// Set a moderate speed and run all motors forward

motor1.setSpeed(150); motor2.setSpeed(150);

motor3.setSpeed(150); motor4.setSpeed(150);

motor1.run(FORWARD); motor2.run(FORWARD);

motor3.run(FORWARD); motor4.run(FORWARD);

Serial.println("Moving Forward");

void moveBackward() {

// Set a moderate speed and run all motors backward

motor1.setSpeed(150); motor2.setSpeed(150);

motor3.setSpeed(150); motor4.setSpeed(150);

motor1.run(BACKWARD); motor2.run(BACKWARD);

motor3.run(BACKWARD); motor4.run(BACKWARD);

Serial.println("Moving Backward");

}

void turnLeft() {

// For a left turn, have the left motors run backward and the right motors forward

motor1.setSpeed(255); motor2.setSpeed(255);

motor3.setSpeed(255); motor4.setSpeed(255);

motor1.run(BACKWARD); motor3.run(BACKWARD);

motor2.run(FORWARD); motor4.run(FORWARD);

Serial.println("Turning Left");

}

void turnRight() {

// For a right turn, reverse the directions compared to turnLeft()

motor1.setSpeed(255); motor2.setSpeed(255);

motor3.setSpeed(255); motor4.setSpeed(255);

motor1.run(FORWARD); motor3.run(FORWARD);

motor2.run(BACKWARD); motor4.run(BACKWARD);

Serial.println("Turning Right");

}

void stopMotors() {

// Stop all motors

motor1.run(RELEASE); motor2.run(RELEASE);

motor3.run(RELEASE); motor4.run(RELEASE);

Serial.println("Motors Stopped");

}

void processCommand(String command) {

command.trim(); // Remove extra whitespace/newline

if (command.length() == 0) return;

if (command == "AUTO\_MODE") {

autoMode = true;

Serial.println("Switched to AUTO mode");

}

else if (command == "MANUAL\_MODE") {

autoMode = false;

Serial.println("Switched to MANUAL mode");

}

else if (command == "F") {

moveForward();

}

else if (command == "B") {

moveBackward();

}

else if (command == "L") {

turnLeft();

}

else if (command == "R") {

turnRight();

}

else if (command == "S") {

stopMotors();

}

else {

Serial.print("Unknown command: ");

Serial.println(command);

}

}

void loop() {

// Check if there is data available from the ESP8266

if (Serial.available() > 0) {

**7.1 CONCLUSION**

The IoT-based solar grass cutter with obstacle avoidance is an efficient, eco-friendly, and autonomous system designed to simplify grass-cutting tasks. By integrating Arduino and ESP8266, the system enables both automatic obstacle detection and remote control via the Blynk app, ensuring flexibility and user convenience. The ultrasonic sensor enhances safety by detecting obstacles and adjusting movement accordingly, while the relay-controlled 775 DC motor ensures effective grass cutting.

The solar-powered battery system reduces dependency on conventional electricity, making the project sustainable and energy-efficient. The use of a buck converter and TP4056 charging module ensures proper power regulation and battery management. The L293D motor driver efficiently controls the four-wheel drive system, allowing smooth navigation over different terrains.

This project demonstrates the potential of IoT, automation, and renewable energy in everyday applications. With further enhancements, such as AI-based navigation, GPS tracking, or machine learning for optimized grass-cutting patterns, the system can be upgraded for commercial use. Overall, this project provides a cost-effective, smart, and environment-friendly solution for modern lawn maintenance.

**8.1 FUTURE WORKS**

The IoT-based solar grass cutter with obstacle avoidance can be further enhanced with advanced technologies to improve efficiency, functionality, and adaptability. Future improvements can focus on the following areas:

* AI-Based Navigation and Path Optimization: Implementing machine learning algorithms can help the grass cutter analyze terrain and optimize movement patterns, reducing unnecessary energy consumption and improving cutting efficiency.
* GPS-Based Autonomous Operation: Integrating a GPS module can allow the system to operate autonomously within predefined boundaries, enabling precise lawn mapping and geofencing for better coverage.
* Enhanced Obstacle Detection with LiDAR or Multiple Sensors: Instead of relying on a single ultrasonic sensor, incorporating LiDAR or multiple ultrasonic sensors can significantly improve obstacle detection accuracy and avoidance capabilities, especially in complex environments.
* Solar Efficiency Optimization with MPPT: Using a Maximum Power Point Tracking (MPPT) controller can optimize the solar panel's energy output, ensuring better charging efficiency and longer battery life for extended operation.
* Automatic Blade Height Adjustment: A smart cutting mechanism that detects grass height and adjusts the blade height automatically can improve cutting uniformity and efficiency.
* Real-Time Data Monitoring and Cloud Connectivity: Integrating IoT platforms like Firebase, ThingsBoard, or MQTT can enable real-time data logging, allowing users to monitor battery levels, motor performance, and system health remotely.
* Voice Control and Smart Home Integration: Adding compatibility with Google Assistant or Alexa can enable users to control the grass cutter using voice commands, making it more convenient and user-friendly.

**9.1 BIBLIOGRAPHY**

1. **Books & Journals**
   * Mehta, D. S., & Ramesh, T. (2018). *IoT-Based Smart Automation Systems*. Springer Publications.
   * Kumar, P., & Sharma, R. (2020). *Renewable Energy and Sustainable Technologies*. Wiley.
   * Gupta, A., & Verma, S. (2019). *Embedded Systems and Robotics with Arduino & ESP8266*. McGraw-Hill.
   * Singh, J., & Patil, R. (2021). *Autonomous Robotics and AI-based Navigation*. IEEE Xplore.
2. **Research Papers & Articles**
   * Brown, K., & Zhang, L. (2022). "Smart Lawn Maintenance with IoT and AI." *International Journal of Smart Technologies*, Vol. 12, Issue 3, pp. 245-260.
   * Wilson, M., & Jones, P. (2019). "Energy-Efficient Robotics: Solar-Powered Innovations." *Renewable Energy Research Journal*, Vol. 8, No. 4, pp. 112-130.
   * Reddy, V., & Thomas, C. (2020). "Obstacle Avoidance in Autonomous Robots Using Ultrasonic Sensors." *IEEE Transactions on Robotics*, Vol. 15, Issue 2, pp. 87-102.
3. **Web Sources**
   * Arduino Official Documentation: <https://www.arduino.cc>
   * ESP8266 Community Forum: <https://www.esp8266.com>
   * Blynk IoT Platform: <https://blynk.io>
   * Adafruit Motor Shield Guide: https://learn.adafruit.com/adafruit-motor-shield
   * ResearchGate Papers on Solar-Powered Robotics: <https://www.researchgate.net>
   * IEEE Xplore Digital Library: <https://ieeexplore.ieee.org>
4. **Datasheets & Manuals**
   * **Arduino Uno R3 Datasheet** – Arduino.cc
   * **ESP8266 NodeMCU Technical Reference** – Espressif Systems
   * **L293D Motor Driver Datasheet** – Texas Instruments
   * **Ultrasonic Sensor HC-SR04 Datasheet** – SparkFun Electronics
   * **775 DC Motor Technical Specification** – Johnson Electric
   * **12V Solar Panel and Battery Charging Guide** – Renogy Solar

**10.1 PHOTOGRAPHY**

