DESIGN AND IMPLEMENTATION OF AN ASSISTIVE NAVIGATION SYSTEM FOR ROAD ACCIDENT AVOIDANCE USING IOT

[1]Dr. G. Nirmala, [2]Aarthi N, [3]Ameerabee B, [4]Charumathi M, [5]Kavitha S, [6]Mr.A.Karthik
[1]Associate Professor, [6]Assistant Professor, [2,3,4,5]Student, Department of Electronics and Communication Engineering, Mahendra Institute of Technology, Namakkal(Dt), Tamilnadu, India.

ABSTRACT

In today's business ventures and high–risk environments, it is crucial to ensure the use of safety equipment such as helmets, to prevent accidents and protect staff. This project proposes the development of a Head Protector Wearing Identification System, utilizing computer vision and the capabilities of the Internet of Things (IOT). The system aims to enhance safety protocols by continuously monitoring and verifying helmet compliance. It utilizes a combination of hardware and software components. A high-resolution camera, integrated with an IOT devices, captures continuous images of individuals in the designated area. These images are processed using computer vision algorithms, including a pre-trained deep learning model for helmet detection. The steps involved in the implementation include dataset preparation, model training, and integration with the IOT device. The pre-trained model is responsible for identifying individuals wearing helmets and those without, providing real-time results. The IOT device, equipped with connectivity options, transmits this information to a centralized platform or cloud service. The IOT platform serves as a hub for data storage, analysis and visualization. It enables real- time monitoring of helmet compliance, allowing for the generation of alerts or warnings in case of non-compliance. Additionally, a web application can be developed to provide a user-friendly interface for data representation and management.

Keywords: *IOT contraption, IOT device, Internet of Things (IOT)*

INTRODUCTION

In present day enterprises and high-risk conditions, guaranteeing the utilization of wellbeing gear, like caps, is significant for forestalling mishaps and defending staff. This under taking proposes the improvement of a Head Protector Wearing Identification Framework utilizing the capacities of the Web of Things (IOT) and PC vision. The framework means to improve security conventions by constant observing and authorization of protective cap consistence. The framework utilizes a blend of equipment and programming parts. A high-goal camera, coordinated with an IOT gadget, catches constant pictures of people in the assigned region. These pictures are handled utilizing PC vision calculations, including a pre-prepared profound learning model for head protector location.

The means engaged with the execution incorporate dataset assortment, model preparation, and mix with the IOT gadget. The prepared model is liable for distinguishing people wearing caps and those without, producing immediate outcomes. The IOT gadget, outfitted with network choices, communicates this information to a unified stage or cloud administration. The IOT stage fills in as a center point for information capacity, examination, and perception. It works with ongoing observing of head protector consistence, considering the age of alarms or warnings on account of rebelliousness. Furthermore, a web application might be created to give an easy-to-understand connection point to representation and the executives of the gathered information. Contemplations like security, consistence with guidelines, adaptability, power utilization, and framework support have been considered into the plan. Standard updates and testing conventions guarantee the framework's exactness and flexibility to assorted work spaces. This Cap Wearing Recognition Framework presents a complete answer for advancing wellbeing principles in different enterprises. By consolidating IOT innovations with PC vision, it offers a flexible and versatile way to deal with

IJETS

checking and upholding protective cap use, adding to a more secure work space.

SYSTEM ANALYSIS

EXISTING SYSTEM

The safety of individuals within industrial and high-risk environments is paramount, prompting the exploration of technology-driven solutions to ensure compliance with safety protocols, particularly the use of helmets. This review examines the existing systems designed for helmet-wearing detection, with a focus on technologies employed, methodologies, and their effectiveness in real-world scenarios.

Several contemporary systems utilize a combination of computer vision, machine learning, and Internet of Things (IOT) technologies to address helmet compliance. One prevalent approach involves the deployment of cameras and image processing algorithms for real-time monitoring of individuals within designated areas. These systems often integrate pre-trained deep learning models, such as YOLO (You Only Look Once) or SSD (Single Shot Multi box Detector), for accurate and rapid helmet detection.

IOT devices, including Arduino, are commonly employed as hardware components to capture and process images locally. The integration of these devices with cloud-based platforms facilitates centralized data storage, analysis, and visualization. Such platforms enable real-time tracking of helmet compliance, generating alerts or notifications when non-compliance is detected. Existing systems also differ in their scalability, privacy considerations, and adaptability to diverse environments. Some systems are specifically tailored for particular industries, while others offer more generalized solutions. Additionally, advancements in edge computing have allowed for on-device processing, minimizing the need for constant internet connectivity. This review critically assesses the strengths and limitations of current helmet-wearing detection systems, considering factors such as accuracy, real-time responsiveness, and deployment feasibility. Insights gained from this review aim to inform the development of future systems, guiding researchers and practitioners in optimizing safety technologies.

FEATURES

Local Computer Vision Systems Some systems utilize computer vision algorithms, image processing, and machine learning models for helmet detection. These systems may operate on local servers, edge devices, or embedded systems without the need for IoT connectivity.

Camera-Based Solutions Standalone camera-based systems capture video or images in real-time and process the data locally to detect the presence or absence of helmets. These solutions are often deployed in controlled environments like construction sites or manufacturing facilities.

Edge Computing Edge computing technologies enable processing to occur closer to the data source, reducing the need for constant internet connectivity. Helmet detection models can be deployed on edge devices like Nvidia Jetson or Intel Movidius for real-time processing.

Closed-Circuit Television (CCTV) Systems Some existing CCTV systems incorporate helmet-wearing detection features. These systems typically rely on cameras placed strategically in specific locations to monitor compliance.

Stand-Alone Applications There might be standalone applications or software solutions designed for helmet detection that operate independently of IOT. These applications can be installed on local servers or devices for real-time monitoring.

PROPOSED SYSTEM

This proposed framework presents an imaginative security checking arrangement that utilizes a microcontroller, Wi-Fi module, eye blink sensor, gas sensor, IR sensor, and a signal, coming full circle in a hearty Web of Things (IOT) application. The framework is intended to identify and answer expected risks, accordingly upgrading generally well being conventions indifferent modern

conditions.

Microcontroller The microcontroller fills in as the focal handling unit, organizing the functionalities of different sensors and actuators. It processes approaching information, executes wellbeing calculations, and controls framework reactions.

Wi-Fi Module The Wi-Fi module empowers consistent network, permitting ongoing information transmission to a concentrated server or cloud stage. This network works with remote observing and information investigation, upgrading the framework's adaptability and openness.

Eye blink Sensor Is integrated to screen the sharpness and exhaustion levels of people. It distinguishes strange examples, flagging expected slips in consideration, and triggers alarms to forestall mishaps connected with tiredness.

Gas Sensor The gas sensor screens air quality by recognizing the presence of destructive gases. In case of gas holes or raised gas levels, the framework triggers prompt cautions and actuates security measures to forestall wellbeing perils.

IR Sensor The infrared (IR) sensor is sent for nearness discovery. It guarantees the framework answers the presence of people in predefined dangerous zones, setting off ideal admonitions and executing security conventions.

Buzzer The bell goes about as a discernible caution, giving quick alarms in light of recognized security risks. It helps with alarming people nearby, provoking fast reactions to likely risks.

IOT Application The IOT application gives an easy to understand connection point to remote checking and control. It shows continuous information from the sensors, issues cautions, and permits clients to make preventive moves through a portable or online stage.

FEATURES

Thorough Risk Identification The coordination of various sensors works with an exhaustive way to deal with danger location, covering perspectives, for example, weariness, gas breaks, and nearness to risky zones. This guarantees a balanced security framework.

Constant Checking and Alarms The Wi-Fi module empowers ongoing information transmission, permitting nonstop checking of wellbeing boundaries. Quick cautions and warnings are set off, empowering quick reactions to potential security episodes.

Weariness Anticipation The eye blink sensor fills in as a weakness location system, distinguishing indications of sluggishness or diminished readiness. Proactive cautions assist with forestalling mishaps brought about by representative weariness, encouraging a more securework space.

Gas Release and Air Quality Checking The gas sensor persistently screens air quality and recognizes the presence of destructive gases. Fast recognition and alarms guarantee brief reactions to gas spills, limiting wellbeing gambles for staff.

Closeness Detecting for Security Zones The IR sensor adds to work environment security by recognizing the vicinity of people to predefined risky zones. This component upgrades situational mindfulness and decreases the gamble of mishaps in basic regions.

Discernible Cautions for Guaranteed Reaction The coordination of a bell gives perceptible cautions, offering an extra layer of correspondence in uproarious conditions. Discernible admonitions improve the perceivability of wellbeing notices, guaranteeing brief reactions.

Adaptability and Adaptability The IOT application upgrades framework adaptability, considering consistent sending in different modern settings. Distant availability gives adaptability, empowering checking and the board of security boundaries from various areas.

Adjustable Security Conventions The microcontroller's programmability takes into account the customization of security calculations in light of explicit industry necessities. This versatility guarantees that the framework can be custom-made to the special necessities of assorted working environment conditions.

BLOCK DIAGRAM

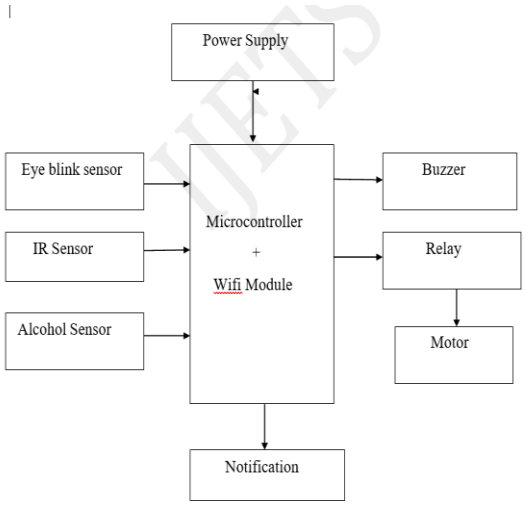


Fig.No 1. Helmet Implementation Model

DESCRIPTION

The power supply is applied to the microcontroller Wi-Fi and the hardware components like Eye blinking sensor, alcohol sensor and IR sensor are connected to the wifi module. Express will make the ESP8266 arrangement, or family, of Wi-Fi chips express. If systems, a

Fables semiconductor organization working out for Shanghai, China, the ESP8266 is incorporating the “ESP8285 and ESP8266EX chips “. ESP8266 Node MCU offers Arduino- likehardware IO event-driven API for network applications. Wi-Fi networking (can be uses as access point and / or station, host a webserver), connect to internet to fetch or upload data.

An eye blink sensor relies on infrared technology to detect if a person’s eye is closed. The sensor is made up of two components: an infrared transmitter and an infrared receiver. The transmitteremits infrared waves onto the eye, while the receiver searches for changes in the reflected waves, indicating that the eye has blinked. An infrared proximity sensor is an infrared proximity sensor that emits infrared light and also receives reflected infrared light and feedback on the presence of obstacles ahead. This module is made using Alcohol gas sensor MQ3. It is a low cost semiconductor sensor which can detect the presence of alcohol gases at concentrations from 0.05mg/ L to 10mg/L. The sensitive material used for this sensor is SnO2, whose conductivity is lower in clean air. An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audioto sound.

A DC motor is an electrical motor that uses direct current (DC) to produce mechanical force. The L298N motor driver module consists of an L298 moto driver IC, 78M05 voltage regulator, resistors, capacitors, power LED, 5V jumper in an integrated circuit. As a result, a notification issent to the user mobile via app which was installed in the user mobile application.

SYSTEM SPECIFICATIONS

HARDWARE EXPLANTION:

NODE MCU

Express will make the ESP8266 arrangement, or family, of Wi-Fi chips Express. If Systems, a fables semiconductor organization working out for Shanghai, China, then the ESP8266 is incorporating the “ESP8285 and ESP8266EX chips”. ESP8266EX (essentially alluded to as ESP8266) is a framework on-chip (SOC) that incorporates a“32-bit Tensilica microcontroller”, standard sophisticated fringe interfaces, control intensifier, receiving wire switches, RFbalun, low disorder get enhancer, channels and power organization modules under a little bundle. It provides capacities to 2.4 GHz Wi-Fi (802. 11 b/g/n, supporting WPA/WPA2), simple to computerized transformation (10-bit ADC), mostly

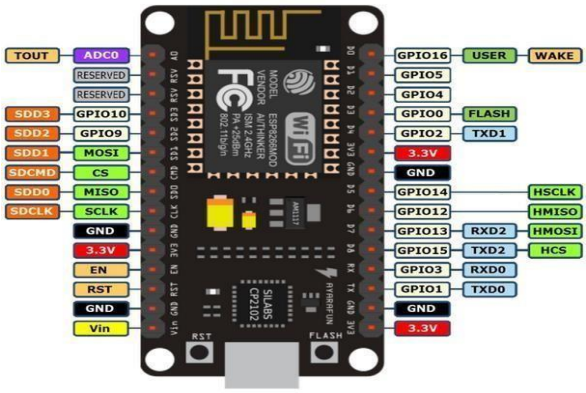
utilized information/yield (16 GPIO), I²S interfaces with DMA (offering pins to GPIO), Inter- Integrated circlet (I²C), serial peripheral interface (SPI), UART (on committed pins, as wellto a transmit-no one but UART might been abled on GPIO2), and heart beat width weak. TheESP8266, designed and manufactured by Express if Systems, contains the crucial elements of a computer. CPU, RAM, networking (WiFi) and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things (IOT) projects of kinds. However, asa chip, the ESP8266 is also hard to access and use. You must solder wires, with the appropriate analog voltage, to its pins for the simplest tasks such as powering it on or sending a keystroke to the "computer" on the chip. This is the level of integration is not a problem using the ESP8266 as an Embedded controller chip in mass-produced electronics. It is a huge burden for hobbyists, hackers, or students who want to experiment with it in their own IOT projects. The NodeMCU is available in various package styles. NodeMCU Dev Board is based on widely explored esp8266 System on Chip from Expressive. It combined features of WIFI access point and station + microcontroller and uses simple LUA based programming language. ESP8266 NodeMCU offers-

- Arduino-like hardware IO
- Event-driven API for network applications
- 10 GPIOs D0-D10, PWM functionality, IIC and SPI communication, 1-Wire andADC A0 etc. all in one board

- Wi-Fi networking (can be uses as access point and/or station, host awebserver), connect to internet to fetch or upload data.

- Excellent few $ system on board for Internet of Things (IOT) projects.

NodeMCU is an eLua based firmware for the ESP8266 WiFi SOC from Espress if systems. The hardware is based on the ESP-12 module. The firmware is based on the Espress if NON- OS SDK 2.1.0 and uses a file system based on spiffs. The code repository consists of 98.1%C-code that



glues the thin Lua veneer to the SDK. Asynchronous event-driven programming model.

Fig. No 2 NODE MCU Pin Diagram

EYEBLINK SENSOR:

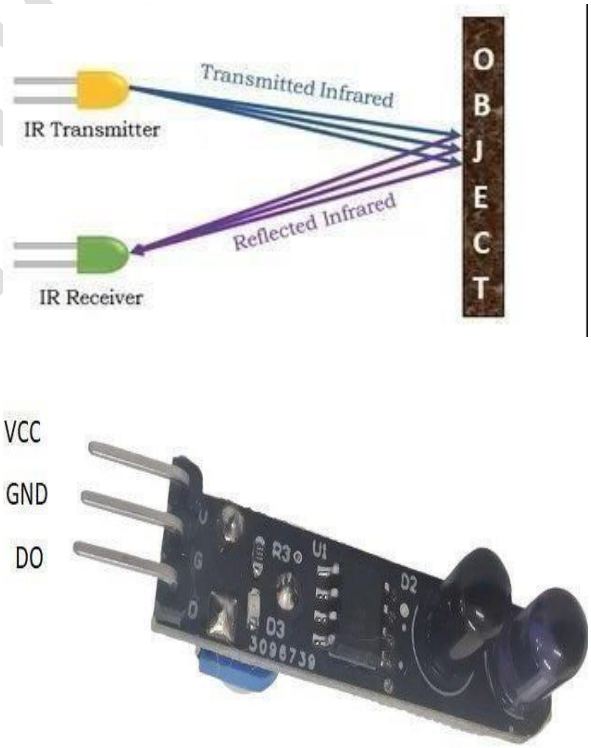


Fig. No 3 Eye Blink Sensor

The sensor is made up of two components: an infrared transmitter and an infrared receiver. The transmitter emits infrared waves onto the eye, while the receiver searches for changes in the reflected waves, indicating that the eye has blinked. The sensor's output is sent to a microcontroller board, such as Arduino, Raspberry

Pi, AVR, PIC, or other microcontrollers.

To create an eye blink sensor with an Arduino, one requires an eye blink sensor, an Arduino board, and a switch between the power supply and the Arduino. The sensor has three pins: GND(ground), 5V, and OP (output). The sensor's output is transmitted to the Arduino board, which can be programmed to execute various tasks based on the output**.**

Infrared Light Source An IR eye squint sensor regularly comprises of an infrared light- emanating diode (IR Drove). This IR Drove produces a light emission light towards the client'seyes.

Reflection and Retention At the point when the IR light shaft from the Drove experiences the client's eye, some of it is bounced off the cornea and other eye surfaces, while some is consumed by the eye.

IR Identifier A photo detector, for example, a photo transistor or photodiode, is situated to getthe mirrored IR light. The force of the mirrored IR light is impacted by the presence and development of the eyelids.

Flicker Discovery At the point when the client squints, the eyelids move and can hinder or to some extent block the way of the mirrored IR light. This causes an adjustment of the force of the IR light got by the photo detector.

Signal Handling The result from the photo detector is then handled by electronic hardware, which can intensify and channel the sign. The sensor can be intended to distinguish changes in the power of the mirrored IR light that compare to eye flickers.

Result and Understanding The sensor gives a result signal that can be utilized to recognize and gauge eye squints. By dissecting the example and length of these progressions in the IR signal, the framework can decide when a flicker happens and how lengthy it endures.

It's vital to take note of that the particular plan and part so of an IR eye flicker sensor might shift relying upon the producer and application. Furthermore, signal handling and information examination strategies assume a pivotal part in precisely distinguishing and deciphering eye squints from the sensor's result.

IR SENSOR

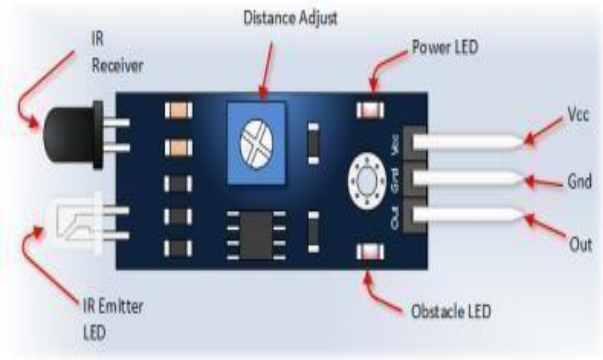


Fig. No 4 IR Sensor

An infrared Proximity Sensor is an infrared proximity sensor that emits infrared light and also receives reflected infrared light and feedback on the presence of obstacles ahead. Principle: Infrared transmitter tube is composed of an infrared light-emitting diode moment luminous body, with infrared radiation efficiency of the material (commonly gallium arsenide) made of PN junction, forward bias to PN junction injection current to excite infrared light. The infrared receiver tube is the infrared light signal in electrical signal semiconductor devices, its core component is a special material PN junction, and with the infrared light intensity increasing the current also increases the output analog signal.

When the detection direction encounters an obstacle (reflective surface), infrared light is reflected back to be received by the receiver tube, after the LM393 comparator processing output a digital signal, while the green indicator lights up, the detection distance can be adjusted through the potentiometer knob.

Specification

- Power:3.3V~5V
- Dimension: 39mm× 15.5mm(including the

IRLED)

- Mountingholes'size:3mm
- Detection range: 2cm~ 30cm(depending on the obstacle's color, farthest for white)
- Detection angle: 35°
- Recommended environment: indoor, to avoid the sunshine effect.

Table 1 PINOUTS:

Pin No.	Symbol	Descriptions
1	DOUT	Digital output
2	AOUT	Analog output
3	GND	Ground
4	VCC	Positive power supply (3.3V-5.0V)

ALCOHOL SENSOR

Fig.No 5 Alcohol Sensor

This module is made using Alcohol Gas Sensor MQ3. It is a low cost semiconductor sensor which can detect the presence of alcohol gases at concentrations from 0.05 mg/L to 10mg/L. The sensitive material used for this sensor is SnO2, whose conductivity is lower in clean air. It's conductivity increases as the concentration of alcohol gases increases. It has high sensitivity to alcohol and has a good resistance to disturbances due to smoke, vapor and gasoline. This module provides both digital and analog outputs. MQ3 alcohol sensor module can be easily interfaced with Microcontrollers, Arduino Boards, Raspberry Pi etc.

This alcohol sensor is suitable for detecting alcohol concentration on your breath, just like your common Breathalyzer. It has a high sensitivity and fast response time. Sensor provides an analog resistive output based on alcohol concentration. The drive circuit is very simple all it needs is one resistor. A simple interface could be a 0-3.3V ADC.

DC GEAR MOTOR:

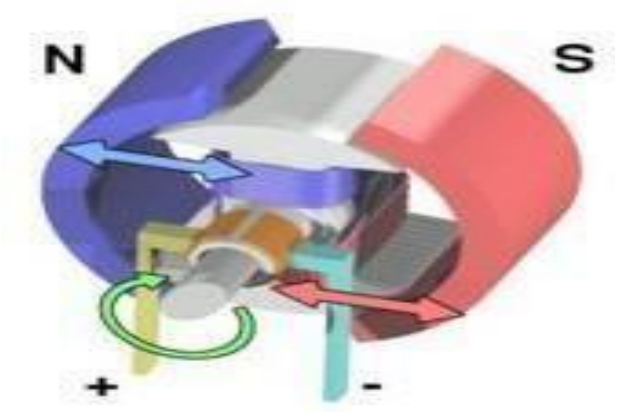


Fig. No 6 DC Gear Motor

A DC motor is an electrical motor that uses direct current (DC) to produce mechanical force. The most common types rely on magnetic forces produced by currents in the coils. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current in part of the motor.

DC motors were the first form of motors widely used, as they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances. The universal motor, a lightweight brushed motor used for portable power tools and appliances can operate on direct current and alternating current. Larger DC motors are currently used in propulsion of electric vehicles, elevator and hoists, and in drives for steel rolling mills. The advent of power electronics has made replacement of DC motors with AC motors possible in many applications.

Electromagnetic motors: A coil of wire with a current running through it generates an electromagnetic field aligned with the center of the coil. The direction and magnitude of the magnetic field produced by the coil can be changed with the direction and magnitude of the current flowing through it.

A simple DC motor has a stationary set of magnets in the stator and an armature with one or more winding so insulated wire wrapped around a soft iron core that concentrates the magnetic field. The windings usually have multiple turns around the core, and in large motors there can be several parallel current paths. The ends of the wire winding are connected to a commutator. The commutator allows each armature coil to be energized in turn and connects the rotating coils with the external power supply through brushes. (Brushless DC motors have electronics that switch the DC current to each coil on and off and have no brushes.)

The total amount of current sent to the coil, the coil's size, and what it is wrapped around decide the strength of the electromagnetic field created.

The sequence of turning a particular coil on or off dictates what direction the effective electromagnetic fields are pointed. By turning on and off coils in sequence, a rotating magnetic field can be created. These rotating magnetic fields interact with the magnetic fields of the magnets (permanent or electromagnets) in the stationary part of the motor (stator) to create a torque on the armature which causes it to rotate. In some DC motor designs, the stator fields use electromagnets to create their magnetic field which allows greater control over the motor.

At high power levels, DC motors are almost always cooled using forced air. Different number of stator and armature field as well as how they are connected provide different inherent speed and torque regulation characteristics. The speed of a DC motor can be controlled by changing the voltage applied to the armature. Variable resistance in the armature circuit or field circuit allows speed control. Modern DC motors are often controlled by power electronics systems which adjust the voltage by "chopping" the DC current into on and off cycles which have an effective lower voltage.

Since the series-wound DC motor develops its highest torque at low speed, it is often used in traction applications such as electric locomotives, and trams. The DC motor was the mainstay of electric traction drives on both electric and diesel-electric locomotives, street-cars/trams and diesel electric drilling rigs for many years. The introduction of DC motors and an electrical grid system to run machinery starting in the 1870s started a new second Industrial Revolution. DC motors can operate directly from rechargeable batteries, providing the motive power forthe first electric vehicles and today's hybrid cars and electric cars as well as driving a host of cordless tools. Today DC motors are still found in applications as

IJETS

small as toys and disk drives, or in large sizes to operate steel rolling mills and paper machines. Large DC motors with separately excited fields were generally used with winder drives for mine hoists, for high torque as well as smooth speed control using thermistor drives. These are now replaced with large AC motors with variable frequency drives.

If external mechanical power is applied to a DC motor it acts as a DC generator, a dynamo. This feature is used to slow down and recharge batteries on hybrid and electric cars or to return electricity back to the electric grid used on a street car or electric powered train line when they slow down. This process is called regenerative braking on hybrid and electric cars. In diesel electric locomotives they also use their DC motors as generators to slow down but dissipate theenergy in resistor stacks. Newer designs are adding large battery packs to recapture some of thisenergy.

BUZZER

An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm,music, bell & siren.

Fig. No 7 Buzzer Pin Configuration

The **pin configuration of the buzzer** is shown below. It includes two pins namely positive and negative. The positive terminal of this is represented with the '+' symbol or a longer terminal. This terminal is powered through 6Volts whereas the negative terminal is represented with the'- 'symbol or short terminal and it is connected to the GND terminal.

Electromechanical

This buzzer was launched in the year 1831 by an American Scientist namely Joseph Henry but, this was used in doorbell sun till they were eliminated in1930 in support of musical bells, which had a smooth tone.

Piezoelectric

These buzzers were invented by manufacturers of Japanese & fixed into a broad range of devices during the period of 1970s – 1980s. So, this development primarily came due to cooperative efforts through the manufacturing companies of Japanese. In the year 1951, they recognized the Application Research Committee of Barium Titanate that allows the corporations to be cooperative competitively & bring about numerous piezoelectric creations.

Specifications

The **specifications of the buzzer** include the following.

- Color is black
- The frequency range is 3,300Hz
- Operating Temperature ranges from – 20°Cto+60°C
- Operating voltage ranges from 3V to 24V DC
- The sound pressure level is 85dBA or 10cm
- The supply current is below15Ma

IJETS

MOTOR DRIVER:

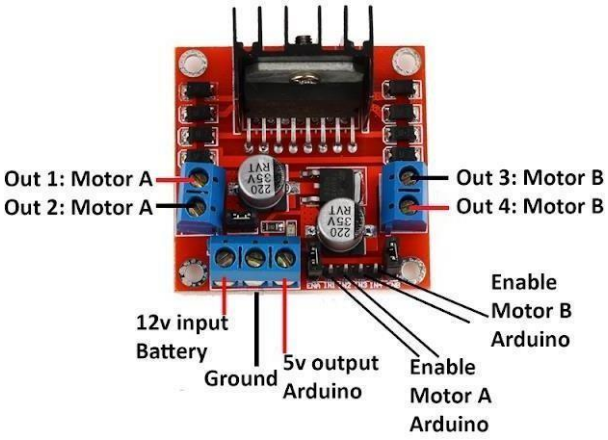


Fig. No 8 Motor Drive

L298N Motor Driver Module

78M05 Voltage regulator will be enabled only when the jumper is placed. When the power supply is less than or equal to 12V, then the internal circuitry will be powered by the voltage regulator and the 5V pin can be used as an output pin to power the microcontroller. The jumper should not be placed when the power supply is greater than 12V and separate 5V should be given through 5V terminal to power the internal circuitry.

ENA&ENB pins are speed control pins for Motor A and Motor B while IN1&IN2 and IN3& IN4 are direction control pins for Motor A and Motor B.

Features & Specifications

- Driver Model: L298N2A
- Driver Chip: Double H Bridge L298N
- Motor Supply Voltage (Maximum): 46V
- Motor Supply Current (Maximum): 2A
- Logic Voltage: 5V
- Driver Voltage: 5-35V
- Driver Current:2A
- Logical Current:0-36mA
- Maximum Power (W): 25W
- Current Sense for each motor
- Heat sink for better performance
- Power-On LED indicator

Power supply

Fig. No 9 Battery

A battery is a source of electric power consisting of one or more electrochemical cells with external connections [1] for powering electrical devices. When a battery is supplying power, its positive terminal is the cathode and its negative terminal is the anode. [2] The terminal marked negative is the source of electrons that will flow through an external electric circuit to the positive terminal. When a battery is connected to an external electric load, a redox reaction converts high-energy reactants to lower-energy products, and the free-energy difference is delivered to the external circuit as electrical energy. Historically the term "battery" specifically referred to a device composed of multiple cells; however, the usage has evolved to include devices composed of a single cell. [3]

Primary (single-use or "disposable") batteries are used once and discarded, as the electrode materials

are irreversibly changed during discharge; a common example is the alkaline battery used for flashlights and a multitude of portable electronic devices. Secondary (rechargeable) batteries can be discharged and recharged multiple times using an applied electric current; theoriginal composition of the electrodes can be restored by reverse current.
Examples include the lead–acid batteries used in vehicles and lithium-ion batteries used forportable electronics such as laptops and mobile phones.

Batteries come in many shapes and sizes, from miniature cells used to power hearing aids and wristwatches to, at the largest extreme, huge battery banks the size of rooms that provide standby or emergency power for telephone exchanges and computer data centers.
Batteries have much lower specific energy (energy per unit mass) than common fuels such as gasoline. In automobiles, this is somewhat offset by the higher efficiency of electric motors in converting electrical energy to mechanical work, compared to combustion engines.
Batteries convert chemical energy directly to electrical energy. In many cases, the electrical energy released is the difference in the cohesive [17]or bond energies of the metals, oxides, or molecules undergoing the electrochemical reaction. For instance, energy can be stored in Znor Li, which are high-energy metals because they are not stabilized byd-electron bonding, unlike transition metals.
Batteries are designed so that the energetically favorable redox reaction can occur only when electrons move through the external part of the circuit.
A battery consists of some number of voltaic cells. Each cell consists of two half-cells connected in series by a conductive electrolyte containing metal cations.
One half-cell includes electrolyte and the negative electrode, the electrode to which anions (negatively charged ions) migrate; the other half-cell includes electrolyte and the positive electrode, to which cations (positively charged ions) migrate. Cations are reduced (electrons are added) at the cathode, while metal atoms are oxidized (electrons are removed) at the anode.
[18] Some cells use different electrolytes for each half-cell; then a separator is used to prevent mixing of the electrolytes while allowing ions to flow between half-cells to complete the electrical circuit.

SOFTWARE TESTING

SOFTWAR E EXPLANTION Embedded System

An embedded system is an application that contains at least one programmable computer (typically in the form of a microcontroller, a microprocessor or digital signal processor chip) and which is used by individuals who are, in the main, unaware that the system is computer- based.

Introduction

Looking around, we find ourselves to be surrounded by various types of Embedded Systems. Be it a digital camera or a mobile phone or a washing machine, all of them has some kind of processor functioning inside it. Associated with each processor is the embedded software. If hardware forms the body of an embedded system, embedded processor acts as the brain, and embedded software forms its soul. It is the embedded software which primarily governs the functioning of embedded systems.
During infancy years of microprocessor-based systems, programs were developed using assemblers and fused into the EPROMs. There used to be no mechanism to find what the program was doing. LEDs, switches, etc. were used to check correct execution of the program. Some ‘very fortunate’ developers had In-circuit Simulators (ICEs), but they were too costly and were not quite reliable as well.
As time progressed, use of microprocessor-specific assembly-only as the programming language reduced and embedded systems moved onto C as the embedded programming language of choice. C is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size

efficiency, etc. are prime requirements.

Initially C was developed by Kernighan and Ritchie to fit into the space of 8K and to write (portable) operating systems. Originally it was implemented on UNIX operating systems. A sit was intended for operating systems development, it can manipulate memory addresses. Also, it allowed programmers to write very compact codes. This has given it there put at ion as the Language of choice for hackers too.

As assembly language programs are specific to a processor, assembly language didn't offer portability across systems. To overcome this disadvantage, several high-level languages, including C, came up. Some other languages like PLM, Modula-2, Pascal, etc. also came but couldn't find wide acceptance. Amongst those, C got wide acceptance for not only embedded systems, but also for desktop applications. Even though C might have lost its sheen as mainstream language for general purpose applications, it still is having a strong-hold in embedded programming. Due to the wide acceptance of C in the embedded systems, various kinds of support tools like compilers & cross-compilers, ICE, etc. came up and all this facilitated development of embedded systems using C. Subsequent sections will discuss what is Embedded C, features of C language, similarities and difference between C and embedded C, and features of embedded C programming.

EMBEDDED SYSTEMS PROGRAMMING

Embedded systems programming is different from developing applications on a desktop computer. Key characteristics of an embedded system, when compared to PCs, are as follows. Embedded devices have resource constraints (limited ROM, limited RAM, limited stack space, less processing power) Components used in embedded system and PCs are different; embeddedsystems typically use smaller, less power consuming components. Embedded systems are moretied to the hardware.

Two salient features of Embedded Programming are code speed and code size. Code speed is governed by the processing power, timing constraints, whereas code size is governed by available program memory and use of programming language. Goal of embedded system programming is to get maximum features in minimum space and minimum time.

Embedded systems are programmed using different type of language

- Machine Code
- Low level language, i.e., assembly
- High level language like C, C++,Java, Ada, etc.
- Application-level language like Visual Basic, scripts, Access, etc.

Assembly language maps mnemonic words with the binary machine codes that the processor uses to code the instructions. Assembly language seems to be an obvious choice for programming embedded devices.

However, use of assembly language is restricted to developing efficient codes in terms of size and speed. Also, assembly codes lead to higher software development costs and code portability is not there.

Developing small codes are not much of a problem, but large programs/projects become increasingly difficult to manage in assembly language. Finding good assembly programmers has also become difficult nowadays. Hence high-level languages are preferred for embedded systems programming.

Use of C in embedded systems is driven by following advantages it is small and reasonably simpler to learn, understand, program and debug. C Compilers are available for almost all embedded devices in use today, and there is a large pool of experienced C programmers.

Unlike assembly, C has advantage of processor-independence and is not specific to any particular microprocessor/ microcontroller or any system. This makes it convenient for a user to develop

programs that can run on most of the systems. As C combines functionality of assembly language and features of high-level languages, C is treated as a 'middle-level computer language' or 'high level assembly language'. It is fairly efficient. It supports access to I/O and provides ease of management of large embedded projects.

Many of these advantages are offered by other languages also, but what sets C apart from others like Pascal, FORTRAN, etc. is the fact that it is a middle level language; it provides direct hardware control without sacrificing benefits of high-level languages. Compared to other high-level languages, C offers more flexibility because C is relatively small, structured language; it supports low-level bit-wise data manipulation. Compared to assembly language, CCode written is more reliable and scalable, more portable between different platforms (with some changes). Moreover, programs developed in C are much easier to understand, maintain and debug. Also, as they can be developed more quickly, codes written in C offers better productivity.

C is based on the philosophy 'programmers know what they are doing'; only the intentions are to be stated explicitly. It is easier to write good code in C & convert it to an efficient assembly code (using high quality compilers) rather than writing an efficient code in assembly itself. Benefits of assembly language programming over C are negligible when we compare the ease with which C programs are developed by programmers.

Objected oriented language, C++ is not apt for developing efficient programs in resource constrained environments like embedded devices. Virtual functions & exception handling of C++ are some specific features that are not efficient in terms of space and speed in embedded systems. Sometimes C++ is used only with very few features, very much as C.

Ada, also an object-oriented language, is different than C++. Originally designed by the U.S. DOD, it didn't gain popularity despite being accepted as an international standard twice (Ada83 and Ada95). However, Ada language has many features that would simplify embedded softwaredevelopment.

Java is another language used for embedded systems programming. It primarily finds usage in high-end mobile phones as it offers portability across systems and is also useful for browsing applications.

Java programs require Java Virtual Machine (JVM), which consume lot of resources. Hence it is not used for smaller embedded devices. Dynamic C and B# are some proprietary languages which are also being used in embedded applications. Efficient embedded C programs must be kept small and efficient; they must be optimized for code speed and code size.

Good understanding of processor architecture embedded C programming and debugging tools facilitate this.

Difference between C and embedded C:

Though C and embedded C appear different and are used indifferent contexts, they have more similarities than the differences. Most of the constructs are same; the difference lies in their applications.

C is used for desktop computers, while embedded C is for microcontroller-based applications. Accordingly, C has the luxury to use resources of a desktop PC like memory, OS, etc. While programming on desktop systems, we need not bother about memory.

However, embedded C has to use with the limited resources (RAM, ROM, I/Os) on an embedded processor. Thus, program code must fit into the available program memory. If code Exceeds the limit, the system is likely to crash. Compilers for C (ANSI C) typically generate OS dependent executables. Embedded C requires compilers to create files to be downloaded to the microcontrollers/microprocessors where it needs to run. Embedded compilers give access to all resources which is not provided in compilers for desktop computer applications. Embedded systems often have the real-time constraints, which is usually not there with desktop computer applications.

Embedded systems often do not have a console, which is available in case of desktop applications. So, what basically is different while programming

with embedded C is the mindset; for embedded applications, we need to optimally use the resources, make the program code efficient, and satisfy real time constraints, if any. All this is done using the basic constructs, syntaxes, and function libraries of 'C'.

Keil C51 C Compilers

- Direct C51 to generate a listing file
- Define manifest constants on the command line
- Control the amount of information included in the object file
- Specify the level of optimization to use
- Specify the memory models

Specify the memory space for variables The Keil C51 C Compiler for the 8051 microcontroller is the most popular 8051 C compiler in the world. It provides more features than any other 8051 C compiler available today.

The C51 Compiler allows you to write 8051 microcontroller applications in C that, once compiled, have the efficiency and speed of assembly language. Language extensions in the C51 Compiler give you full access to all resources of the 8051.

ARDUINO IDE

The Arduino Integrated Development Environment-or Arduino Software (IDE)- contains a texted it or for writing code, a message area, a text console, a tool bar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

ARDUINO IDE- Arduino Integrated Development Environment

Arduino (/rdwino/) is a company, project, and user community for open-source hardware and software that designs, manufactures, and sells microcontroller kits and single-board microcontrollers for digital device construction.

The software is licensed under either the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), allowing anyone to manufacture Arduino boards and distribute software, while its hardware products are covered by a CC BY-SA license. The designs of Arduino boards make use of a variety of microprocessors and controllers. [2] Commercially available Arduino boards can be found on the official website or through authorized distributors. Sets of digital and analog input/output (I/O) pins can be connected to various expansion boards (also known as "shields") or breadboards (used for prototyping) and other circuits on the boards. The boards have serial communications interfaces, some of which are USB (Universal Serial Bus), and they can also be used to load programs. The standard API for programming the microcontrollers, also known as the Arduino Programming Language, is inspired by the Processing language and can be used with a modified version of the Processing IDE. The microcontrollers can be programmed using the C and C++ programming languages. The Arduino project provides a Go-based command line tool and integrated development environment (IDE) in addition to traditional compiler tool chains.

The Arduino project began in 2005 as a tool for students at the Interaction Design Institute in Ivrea, Italy, [3]. Its goal was to provide novices and professionals alike with a low-cost and simple method for creating devices that use sensors and actuators to interact with their surroundings. Simple robots, thermostats, and motion detectors are typical examples of such items geared toward novice hobbyists.

The bar where some of the project's founders used to meet is where the name Arduino comes from. It is located in Ivrea, Italy. The bar was named after Arduino of Ivrea, who was King of Italyfrom1002to 1014and was margrave of the March of Ivrea.[4] History The first Arduino was

IJETS

made at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy.[3]

The students used a $50 BASIC Stamp microcontroller for the Arduino project. In 2003 Hernando Barragan made the improvement stage wiring as an Expert's proposition project at IDII, under the oversight of Massimo Baozi and Casey Reas. Casey Reas is known for co- making, with Ben Fry, the Handling improvement stage. The objective of the project was to develop low-cost, easy-to-use tools for non-engineers to use in digital project creation.

In2005, Massimo Banzi, DavidMellis, another IDII student, and David Cuatrilloes expanded Wiring by adding support for the less expensive ATmega8 microcontroller. The Wiring platform consisted of a printed circuit board (PCB) with an ATmega128 microcontroller, an IDE based on Processing, and library functions that made it simple to program the microcontroller.[5] Arduino was the name of the new project that came from a fork of Wiring.[5] The initial core team of Arduino consisted of Massimo Banzi, David Cuatrilloes, TomIgor, Gianluca Martino, and David Melis.[3] After the platform was finished, lighter and cheaper versions were made available to the open-source community. In the middle of 2011,it was estimated that over 300,000 official Arduinos had been produced for commercial use [6], and in 2013, 700,000 official boards were in use [7].

LEGACY IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Microsoft Windows, mac OS, and Linux) that is written in the Java programming language.

It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source codefor the IDE is released under the GNU General Public License, version 2.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting thesketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also includedwith the IDE distribution. The Arduino IDE employs the program argued to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

From version 1.8.12, Arduino IDE windows compiler supports only Windows 7 or newer OS. On Windows Vista or older one gets "Unrecognized Win32 application" error when trying to verify/upload program. To run IDE on older machines, users can either use version 1.8.11, or copy "Arduino-builder" executable from version 11 to their current install folder as its independent from ID.

RESULTS AND DISCUSSION

APP IMPLEMENTATION

The software connection is based on IOT with the help of Node MCU and a WIFI- module, which is interfaced with the Blynk app to store values and output results. The image shows the online state, while the other shows offline state. The Wi-Fi module is connected to the microcontroller and Blynk app to display and store values. The phone screen displays helmet status, notification, and calls for the rider. The Screen will be transparent to avoid distracting the rider. The starter circuit receives input data from the relay circuit and turns on the vehicle’s ignition. The app’s received data is stored in a database, specifically the Firebase database, which offers real- time database instance.

It ensures that the work achieves its intended goal, both components must be connected and operated simultaneously. The testing, fine-tuning, and debugging phases are essential in the design process and take place when the hardware and software components are integrated. The tool is divided into two categories: interface programming and sequence programming.

The accelerometer sensor detects when the bike rider falls down due to an accident. When the alcohol sensor detects that the rider is under the influence, the buzzer will sound. The hardwarethat can detect an accident, whether the rider is wearing a helmet or not, and ensures the rider'ssafety at all times. The system includes an alcohol sensor and an eye tracking sensor to prevent accidents and ensure the safety of the bike rider. The motor stops running and the system detects drownziness to prevent accidents.

Fig.No 10 KIT OUTPUT IMAGE BASED ON USER CONDITON

RESULT ON USER CONDITION

USERCONDITION	MQ3-SENSORREADING	CONDITION OF BIKE
Drunk and No Helmet	Positive	Online
Drunk and Wearing helmet	Negative	Online
Sober and No helmet	Positive	Online
Sober and Wearing Helmet	Positive	Offline

The smart helmets developed and tested for various conditions to find out how effectively it operates. There are mainly 4 different conditions the smart helmets is tested for. When the user is drunk and he is not wearing any helmet, the bike will not start. The proximity IR sensor will detect no helmet and the MQ-3 Alcohol sensor will detect alcohol and disable the ignition of the bike. When the user is wearing helmet the proximity sensor will give positive signal but since the user is drunk the MQ-3sensorwillgive negative reading and as a result the bike will notbe able to start.

When the user is sober the MQ-3 sensor will give positive reading but since the user is not wearing any helmet so the proximity IR sensor will send negative signal and the bike will not start. It means Bike conditions is 0. There is only one condition where the bike will start which is considered as bit1.The user is sober so the MQ-3 sensor will send positive signal and the user is wearing the helmet so the proximity sensor will also send positive signal. As a result, the bike can now started. The accelerometer sensor detects when the bike rider falls down due to anaccident. When the alcohol sensor detects that the rider is under the influence, the buzzer will sound, it shows the hardware that can detect an accident, whether the rider is wearing a helmet or not, and ensures the rider's safety at all times. The system includes an

alcohol sensor and an eye tracking sensor to prevent accidents and ensure the safety of the bike rider. The motor stops running and the system detects drowsiness to prevent accidents.

By integrating sensors and data analytics the system can detect potential collision risks and alert drivers to take evasive actions, reducing the chances of accidents. In the event of accident, the system can automatically send distress signal to emergency services, providing quick response and assistance to those involved. The system can analyze traffic data and suggest alternative routes to drivers to avoid congested or accident-prone areas, improving overall road safety.

An intuitive interface can enhance user experience, making it easier for drivers to interact with the system and receive timely alerts and notifications.

The system can collect and analyzed at a on road accidents and near-misses, providing valuable insights for authorities to implement targeted safety measures. These devices can provide real-time data on helmet usage, adding additional layers of information for comprehensive safety monitoring. Another potential enhancement is the implementation of Augmented Reality (AR) technology for safety monitoring. This would allow for immersive training experiences that simulate hazardous situations, helping employees understand the importance of wearing a helmet in a realistic environment. Additionally, exploring the use of edge computing for on- device processing of image data could improve real-time processing capabilities and reduce reliance on continuous internet connectivity. Finally. Expanding the frameworks capabilities to monitor other safety parameters beyond helmet compliance, such as the detection of other personal protective equipment (PPE) or identification of hazardous behaviors, would further enhance its effectiveness.

IMPLEMENTING THE SOURCE CODE IDE UNDER GNU

The General Public License is the incompatible with proprietary software. The main reason is that the GPL compels a user to make the source code available when distributing any copies of the software, and that all modifications to the original source are also licensed under the GPL. In any GPL-licensed source code is incorporated into another project. The

Software under the GPL may be run for all purposes and even as a tool for creating proprietary Software, such as when using GPL-licensed compliers. The program or code written in the Arduino IDE is often called as sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software.

Fig. No 6.2 Source Code Implementation for Arduino IDE

CONCLUSION

The proposed IOT-based Head Protector Wearing Location Framework presents a ground breaking answer for address basic wellbeing worries in modern working environments. Throughthe mix of PC vision, IOT innovations, and ongoing observing, this framework offers a completeway to deal with guaranteeing the predictable utilization of security head protectors among faculty. As we close, a few central issues highlight the importance and possible effect of this creative security arrangement.

Upgraded Security Authorization The framework's continuous observing abilities give an uncommon degree of wellbeing implementation. Quick discovery of cap wearing in fringement empowers brief restorative activity, diminishing the gamble of head wounds.

Cloud-Based Examination for Informed Direction Utilizing cloud stages for information capacity and examination enables associations with significant experiences. Verifiable information investigation, pattern recognizable proof, and consistence rates offer an establishment for informed independent direction and nonstop wellbeing upgrades.

Versatility and Flexibility The IOT engineering guarantees versatility, making the framework relevant to various modern settings. Its flexibility to changing circumstances and various work situations makes it an adaptable answer for various workplaces.

Easy to understand Point of interaction for Further Developed Acknowledgment The improvement of an easy to use versatile/web application adds to higher client acknowledgment. By giving a direct connection point to checking and overseeing protective cap consistence, the framework turns out to be more open to faculty and bosses.

Proactive Wellbeing Measures The computerized ready framework adds to proactive security measures. Prompt warnings empower bosses to make quick restorative moves, cultivating a culture of wellbeing and counteraction.

Information Security and Protection Contemplations: Perceiving the significance of

Carryout AR advancements for security preparing purposes. Make vivid preparation encounters that recreate unsafe situations, permitting staff to grasp the significance of cap use in a reasonable climate.

- **Edge Figuring for Ongoing Handling**

Explore the possibility of integrating edge registering for on-gadget handling of picture information. This can improve constant handling abilities, diminishing reliance on ceaseless web availability.

- **Multi-Boundary Security Observing**

Stretch out the framework's capacities to screen extra wellbeing boundaries past protective cap consistence. This might incorporate the discovery ofother individual defensive hardware (PPE) or the distinguishing proof of perilous ways of behaving.

- **Man-made brainpower (Computer-based intelligence) for Irregularity Discovery**

Incorporate high level simulated intelligence information security and protection, the framework integrates hearty measures to safeguard delicate data. This approach guarantees consistence with guidelines and fabricates trust in the framework's execution.

FUTURE ENHANCEMENT

The proposed IOT-based Cap Wearing Identification Framework establishes a strong starting point for upgrading work environment wellbeing. As innovation keeps on advancing, there are a few energizing roads for future turn of events and improvement, growing the framework's capacities and effect on security conventions.

- **Coordination with Wearable Gadgets** Investigate the mix of protective cap based wearable gadgets with implanted sensors. These gadgets can give continuous information on cap use, contributing extra layers of data for exhaustive security checking.

- **Increased Reality (AR) for Preparing** procedures, like peculiarity location calculations,to recognize inconspicuous deviations from ordinary wellbeing conditions. This can work on the framework's capacity to distinguish strange circumstances that could present dangers.

- **Versatile Application Upgrades**

Upgrade the usefulness of the portable/web application with highlights like gamification, security difficulties, or constant criticism to additionally draw in clients and advance a positive wellbeing society.

- **Prescient Investigation for Security Patterns**

Carry out prescient examination to expect wellbeing patterns and possible areas of concern. By examining verifiable information, the framework can give bits of knowledge into arising security examples and dangers.

- **Worldwide Normalization and Coordinated effort**

Advocate for worldwide normalization in work

environment security advancements. Team up with industry partners, administrative bodies, and innovation designers to lay out normal conventions and interoperability principles.

SCREENSHOTS

A2.1 CODING FOR SOURCE CODE OF IDE UNDER GNU (GENERALPUBLICLICENSE) VERSION 2

Fig No A2.1 Coding for source code of IDE under GNU

A2.2 CODING FOR SENSOR VARIABLE DECLARATION AND VLS ASSIGN

Fig No A2.2 Coding for sensor variable declaration

A2.3 CODING FOR SETTING DISABLE POINTS

Fig No A2.3 Coding for setting disable points and wi-fi credentials

A2.4 CODING TO PUT SET UP CODE TO RUN ONCE

Fig No A2.4 Coding to put set up code to run code

A2.5 CODING TO CHECK THE CONDITIONS BASED ON THE USER

Fig No A2.5 Coding to check the conditions based on the user

RESULT OUTPUTS:

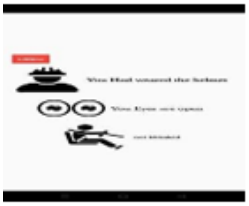


Fig No A2.6 App Offline

Fig No A2.7 App Online

HELMET/DROWZINESSALERT

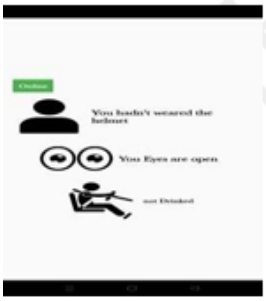


Fig No A2.8. Hadn't wear Helmet

Fig No. A2.9 Had wear the Helmet

REFERENCES:

[1] S. Kalaiarasi, G. Nirmala, “GSM based accident prevention systems for next generation traffic control using parallel monitoring technique”, International Research Journal in Advanced Engineering and Technology (IRJAET),Vol.2,no.2,pp. 949-953,2016.

[2] Subramani, Prabu, Khalid Nazim Abdul Sattar, Rocío Pérez de Prado, Balasubramanian Girirajan, and Marcin Wozniak. 2021. "Multi-Classifier Feature Fusion-Based Road Detection for Connected Autonomous Vehicles" *Applied Sciences* 11, no. 17: 7984.

[3] V. Ellappan, B. Sindhusaranya and T. Hailu, "Multi Inteligent Traffic Light Optimization techniques by appling Modified Component Analysis Algorithm," *2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, Chennai, India, 2019, pp. 316-318.

[4] E. Leela V. Annapoorani, P. Rathna, C. Priyanka3, B. Maheshwari “Health Monitoring and Tracking System for Soldiers Using Internet of Things (IoT)”, 2021/12 International Journal of Advanced Research in Science, Communication and Technology, Volume 12, Issue 2, Pages 58-65.

[5] Md. AtiqurRahman, S.M Ahsanuzzaman, Ishman Rahman, “IoT Based Smart Helmet and Accident Identification System, “IEEEJune 2020.

[6] Sayanee Nanda, Harshada Joshi, Smita Khairnar, Sasi, "An IOT Based Smart System for Accident Prevention and Detection", 2018 IEEE.

[7] S.Keerthiga, F.Anishya and R.P.Kaaviya Priya "Accidents Prevention in Industry using IOT", Asian Journal of Applied Science and Technology.

[8] Nikhil Kumar, DebopamAcharya, and Divya Lohani" An IoT Based Vehicle Accident detection and Classification System using Sensor Fusion," IEEE July 14,2020.

[9] VivekKinage and PiyushPatil, " IoT Based Intelligent System For Vehicle Accident Prevention And Detection At Real Time," IEEE May 07,2020.

[10]SwethaBergonda, Shruti, Sushmita, "IoT Based Vehicle Accident Detection and Tracking System Using GPS Modem", International Journal of Innovative Science and Research Technology Volume 2, Issue 4, April– 2017.

[11] Bharath G S, MeghanaBukkapatnam, Hitesh N, Shria Dhananjay Jadhav, Shashank T K "NB-IoT based Road Accident Alert System, International Journal of Engineering Research &Technolog Vol. 11 Issue 03, March-2022.

IJETS