



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** XI **Month of publication:** November 2023

DOI: <https://doi.org/10.22214/ijraset.2023.56921>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com



Intelligent Packaging System

Rahul M¹, Prof. Shruthi. N²

¹Bachelor of Engineering in Telecommunication Engineering, B.I.T. Bengaluru

²Assistant Professor, Dept. of Electronics and Telecommunication Engineering, B.I.T., Bengaluru

Abstract: *Ecommerce, also known as electronic commerce refers to the buying and selling of various goods and services online. In the recent days the word “Ecommerce” has become very familiar due to the internet boom and due to the kind of comfort (and other services) it provides to its users. Many companies have either established their business or have increased their profits with the help of Ecommerce. There are various studies which show that there is an exponential increase in the Ecommerce sales and services and it is very much evident that it will reach new heights in the upcoming years.*

Ecommerce businesses has seen a 265% growth rate, from \$1.3 trillion in 2014 to \$4.9 trillion in 2021 also higher. With such a mass boom in Ecommerce people don't prefer to buy goods offline since they receive more benefits by buying online. But unfortunately, there are a few problems that both the company and its customers are facing. And gradually such problems have been increasing and major issues were with regard to the faulty delivery system. To handle such situations, we propose a solution to avoid any delivery related problems that are caused during delivery and to put a stop to such problems by making sure every customer gets the products as it is by the dealer directly without any interference while delivering. This report discusses about the part of IoT in secured packaging solution, the proposed approach.

I. INTRODUCTION

A. Overview

The Intelligent Packaging solution aims to use an electronic packaging solution to combat the problem of opening packages during transportation as well as to measure the characteristics of the product, the inner and outer atmosphere of the package. The project IPS aims to use electronic packaging solutions to effectively trigger an alert when a package is opened. This is done by using many sensors in a failsafe system. The problem with single sensor-based systems is that they do not correlate data from different means. IPS uses many sensors that continuously track the physical parameters inside the package to ascertain if the package has been opened or there has been some rise in temperature (for pharmaceutical and temperature sensitive products). Once this alert has been sent to those concerned, they can take necessary action. If there is no alert and a normal delivery takes place, the customer uses his/her mobile phone to scan a QR code displayed on the IPS kit. This will result in an OTP that is received at the customer's mobile and entering the OTP with the kit will reset the device. [1] The IPS kit is then removed from the box and given to the delivery executive to be reused again. Internet of thing (IoT) is not only an interesting research topic but also a booming industrial trend. Although the basic idea is to bring things or objects into the web, there are various approaches because an IoT system is highly application oriented. Some problems that always arise within the packaging of products are difficulty and inaccuracy in determining appropriate packaging options according to type and condition of the product to be packaged. Incorrect decision of packaging option can cause loss in quality, physical damage to product, packed products might get spoiled, especially perishable and time sensitive products. The smart locker may be a modular and expandable solution developed especially for parcel collection. The functionality are often managed locally.

Security of knowledge is of primary concern and therefore the system is fully compliant with all data protection standards. No resident data is stored locally thanks to the enterprise level cloud-based control system. With the rise of ecommerce, there is an Intelligent packaging solution increasing need to manage online purchase deliveries effectively. Now here is this more apparent than within apartment complexes nationwide.

Today, Internet is migrating from connecting people to connecting things, leading to the new concept of Internet of thing (IoT). This new trend brings things or objects into the Internet and generates new applications and business. It is predicted that 212 billion devices are going to be installed by 2020. These things, ranging from indoor wearable devices to outdoor environmental sensors, became new sources generating data on Internet, together making the entities on Internet more aware of the real world. This brings new applications or revolutions in many fields such as transportation, healthcare, home, industrial, agriculture automation freshness and extend shelf life during the distribution process (Yam et al, 2005).

Packaging design that combines elements and design guidelines are oriented to the environment, also play a crucial role in informing and directing consumers to keep buying. The increasing packaging types and options nowadays also necessitates more systematic and precise thanks to select the simplest packaging option for a certain type of product. [2] Decision making processes are often faced with a good range of unique conditions, uncertain, dynamic, time consuming and sophisticated nature.

B. Problem Statement

Definition Increased customer experience expectations and product complexity are driving traditional packaging techniques obsolete. In addition to this, traditional packaging methods do not provide the customer as well as the manufacturer with important information about the product during shipping Fig 1.1 shows delivery mishandling & Fig 1.2 shows damaged product received.



Fig 1.1: Delivery executive mishandling goods



Fig 1.2: Damaged laptop delivered



Fig 1.3: Delivery mishaps



To overcome this, we need to develop an intelligent packaging system that use the Internet of Things and Machine Learning for authenticating, securing, and tracking high value packages Fig 1.3 shows replacement of actual product with soap bar.

C. Objectives

- 1) To Design a kit that is able to detect any kind of intrusion in its space.
- 2) To integrate a kit to give information about all the necessary parameters of the environment when the product is kept inside the package.
- 3) To alert everyone who is in charge of the delivery cycle of the product.
- 4) To include a reset at the end of the delivery cycle and be reusable on the next instance. □ To develop a kit using Node Mc module to verify the authentication of the customer.

II. LITERATURE SURVEY

Following IEEE papers were surveyed for literature survey. here we have summarized the key points available in the papers.

A. *Shahriar Rahman Fahim [1]*

A security box is designed for safe transportation of important and confidential documents. In this proposed system, the state and the location of the security box is continuously monitored from a base station. The box is equipped with a biometric fingerprint sensor and a keypad for entering the passcode. An onboard display shows the current status of the box. In case of the accidental and unwanted opening of the box, a feedback system is provided for notifying the base station.

B. *JiuWen, Yanchun Sun [2]*

In express delivery, couriers will generate a mass of trajectory logs when delivering shipments. To analyze these logs is of great value for the promotion of express delivery service. For any research based on trajectory data, map-matching plays an important role. Existing map-matching algorithms are designed mainly for cars or walks, or ignoring means of transportation. Although these methods can be applied to courier trajectory map-matching and so in this article, we design a map-matching service especially for courier trajectories. As far as we know, existing map-matching algorithms are designed mainly for cars or walks, or ignoring means of transportation

C. *Saipunidzam Mahamad, Suziah Sulaiman, Wong Yi Leng [3]*

There are numerous courier services available locally or worldwide. However, the most services only applicable at their respective company's whereby the pricing information is often hidden and customers usually need to call the company or visit their respective website or application to get quote, then make a price comparison with other company's courier services quotes for them to get their preferable quotation.

D. *Wei Tu [4]*

This article presents an online crowd sourced delivery approach for on-demand food. Facilitated by Internet-of-Things and 3G/4G/5G technologies, public riders can be attracted to act as crowd sourced workers delivering food by means of shared bicycles or electric motorbikes. An online dynamic optimization framework comprising order collection, solution generation, and sequential delivery processes is presented.

E. *Ge Wang Jinsong Han Chen [5]*

Smart packaging adds sensing abilities to traditional packages. This paper investigates the possibility of using RF signals to test the internal status of packages and detect abnormal internal changes. Towards this goal, we design and implement a nondestructive package testing and verification system using commodity passive RFID systems, called Echoscope.

III. HARDWARE & SOFTWARE REQUIREMENTS

A. Hardware Requirements

1) ARM

ARM is a family of instruction set architectures for computer processors based on a reduced instruction set computing (RISC) architecture developed by British company ARM Holdings. A RISC-based ARM processor approach reduces costs, heat and power use.

Such reductions are desirable traits for light, portable, battery-powered devices— including smart phones, laptops, tablet and notepad computers, and other embedded systems. A simpler design facilitates more efficient multi-core CPUs and higher core counts at lower cost, providing improved energy efficiency for servers.

a) *LPC2148*

The LPC2148 microcontrollers are based on a 32/16 bit ARM7TDMI-S CPU. > 16/32-bit ARM7TDMI-S microcontroller in a tiny LQFP64 package.

- 8 to 40 kB of on-chip static RAM and 32 to 512 kB of on-chip flash program memory.
- 128 bit wide interface/accelerator enables high speed 60 MHz operation.
- In-System/In-Application Programming (ISP/IAP) via on-chip boot-loader software.
- Single flash sector or full chip erase in 400ms and programming of 256 bytes in 1ms.
- One or two (LPC2141/2 vs. LPC2144/6/8) 10-bit A/D converters provide a total of s per channel. □6/14 analog inputs, with conversion times as low as 2.44
- Single 10-bit D/A converter provide variable analog output.
- Two 32-bit timers/external event counters (with four capture and four compare channels each), PWM unit (six outputs) and watchdog.
- Low power real-time clock with independent power and dedicated 32 kHz clock input.

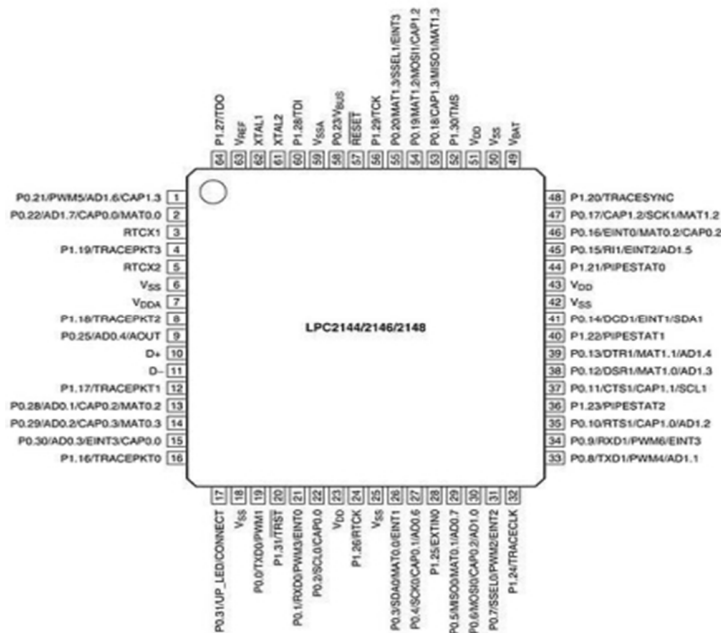


Fig 3.1.1: ARM LPC2148

b) *Architecture Overview*

The ARM7TDMI-S is a general purpose 32-bit microprocessor, which offers high performance and very low power consumption. The ARM architecture is based on Reduced Instruction Set Computer (RISC) principles, and the instruction set and related decode mechanism are much simpler than those of microprogrammed Complex Instruction Set Computers (CISC). This simplicity results in a high instruction throughput and impressive real-time interrupt response from a small and cost-effective processor core Fig 3.1.1 shows ARM LPC2148 pin diagram.

The ARM7TDMI-S processor has two instruction sets:

- The standard 32-bit ARM set.
- A 16-bit Thumb set.

The Thumb set's 16-bit instruction length allows it to approach twice the density of standard ARM code while retaining most of the ARM's performance advantage over a traditional 16-bit processor using 16-bit registers. This is possible because Thumb code operates on the same 32-bit register set as ARM code. Thumb code is able to provide up to 65 % of the code size of ARM, and 160 % of the performance of an equivalent ARM processor connected to a 16-bit memory system. The particular flash implementation in the LPC2141/42/44/46/48 allows for full speed execution also in ARM mode. It is recommended to program performance critical and short code sections (such as interrupt service routines and DSP algorithms) in ARM mode. The impact on the overall code size will be minimal but the speed can be increased by 30% over Thumb mode.

c) On-chip Flash Memory

The LPC2141/42/44/46/48 incorporate a 32 kB, 64 kB, 128 kB, 256 kB and 512 kB flash memory system respectively. This memory may be used for both code and data storage. Programming of the flash memory may be accomplished in several ways. It may be programmed In System via the serial port. The application program may also erase and/or program the flash while the application is running, allowing a great degree of flexibility for data storage field firmware upgrades, etc.

d) On-Chip Static RAM

On-chip static RAM may be used for code and/or data storage. The SRAM may be accessed as 8-bit, 16-bit, and 32-bit. The LPC2141, LPC2142/44 and LPC2146/48 provide 8 kB, 16 kB and 32 Kb of static RAM respectively. In case of LPC2146/48 only, an 8 kB SRAM block intended to be utilized mainly by the USB can also be used as a general purpose RAM for data storage and code storage and execution.

2) IR sensor

An infrared sensor is an electronic device that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measure only infrared radiation, rather than emitting it that is called a passive IR sensor. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation. These types of radiations are invisible to our eyes, that can be detected by an infrared sensor. The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode that is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, the resistances and the output voltages will change in proportion to the magnitude of the IR light received Fig 3.1.2 shows IR Sensor.

Working Principle - "An infrared sensor circuit is one of the basic and popular sensor module in an electronic device. This sensor is analogous to human's visionary senses, which can be used to detect obstacles and it is one of the common applications in real-time."

Specifications

- 5V Operating voltage
- I/O pins are 5V and 3.3V compliant
- Range: Up to 5cm
- Adjustable Sensing range
- Built-in Ambient Light Sensor
- 20mA supply current

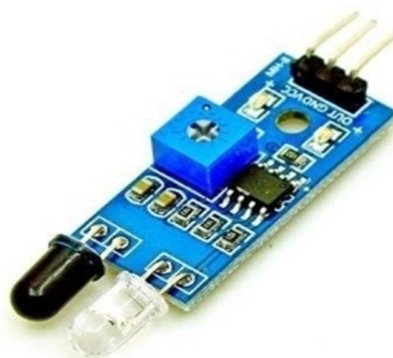


Fig 3.1.2: IR sensor

3) LDR Sensor

An electronic component like LDR or light-dependent resistor is responsive to light. Once light rays drop on it, then immediately the resistance will be changed. The resistance values of an LDR may change over several orders of magnitude. The resistance value will be dropped when the light level increases.

The resistance values of LDR in darkness are several megaohms whereas in bright light it will be dropped to hundred ohms. So due to this change in resistance, these resistors are extremely used in different applications. The LDR sensitivity also changes through the incident light's wavelength Fig 3.1.3 shows the LDR Symbol used in circuit diagram.

In electronic circuits, the LDR symbol is used that mainly depends on the resistor symbol; however, it illustrates the light rays in the arrows form. In this way, it follows the same principle which is used for phototransistor & photodiode circuit symbols wherever arrows are utilized to demonstrate the light dropping on these types of components Fig 3.1.4 shows the LDR Sensor.

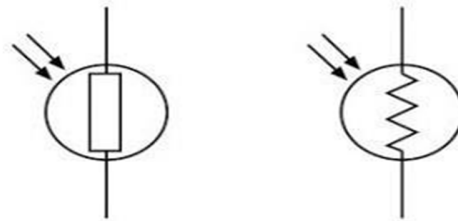


Fig 3.1.3: LDR symbol



Fig 3.1.4: LDR Sensor

4) LCD

Liquid Crystal Display is a flat panel display technology commonly used in TVs and computer monitors. Instead of firing electrons at a glass screen, an LCD has backlight that provides light to individual pixels arranged in a rectangular grid Fig 3.1.5 shows LCD display.

- The operating voltage of this display ranges from 4.7V to 5.3V.
- The display bezel is 72 x 25mm.
- The operating current is 1mA without a backlight.
- PCB size of the module is 80L x 36W x 10H mm.
- HD47780 controller.
- LED color for backlight is green or blue.
- Number of columns – 16.
- Number of rows – 2.

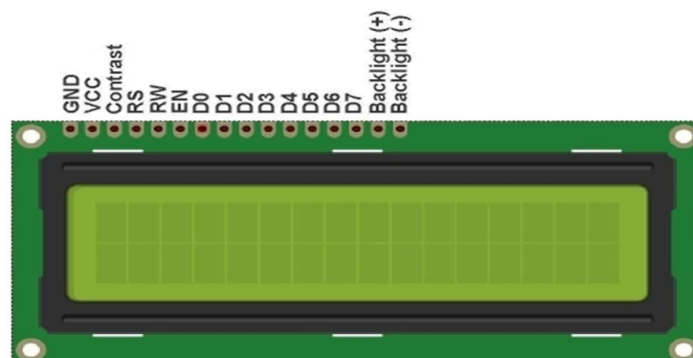


Fig 3.1.5: LCD

5) DC Motor

A DC motor or direct current motor is an electrical machine that transforms electrical energy into mechanical energy by creating a magnetic field that is powered by direct current. When a DC motor is powered, a magnetic field is created in its stator. The field attracts and repels magnets on the rotor; this causes the rotor to rotate. To keep the rotor continually rotating, the commutator that is attached to brushes connected to the power source supply current to the motors wire windings Fig 3.1.6 shows DC motor. One of the reasons DC motors are preferred over other types of motors is their ability to precision control their speed, which is a necessity for industrial machinery. DC motors are able to immediately start, stop, and reverse—an essential factor for controlling the operation of production equipment.

Specifications

- Operating Voltage(V): 12
- Rated Speed (RPM): 200
- Rated Torque(kg-cm): 1.5
- Stall Torque(kg-cm): 5.4
- Load Current (A): 0.3
- No Load Current (A): 0.06



Fig 3.1.6: DC motor

6) Power Supply

Power supply unit It has 230V,50Hz Ac supply This supply can converted into 5V Dc, because all component (Sensor) need Constant 5V Dc supply So voltage regulator (LM7805) can provide constant 5V Dc supply to component. One of the reasons DC motors are preferred over other types of motors is their ability to precision control their speed, which is a necessity for industrial machinery. DC motors are able to immediately start, stop, and revers an essential factor for controlling the operation of production equipment.

Table 1.1: POWER SUPPLY

Description	Specification
Voltage	100 to 240 VAC (+or - 10%).
	16 A maximum at 100 to 120 VAC and 1451-W output. 16 A maximum at 200 to 240 VAC and 3051-W output. For current ratings of plugs, see Figure 6-3 on page 6-154 .
Frequency	50 to 60 Hz (nominal) (+ or - 3 Hz for full range).

7) MEMS (ADXL335)

An accelerometer is an electromechanical device that will measure acceleration force. It shows acceleration, only due to cause of gravity i.e. g force. It measures acceleration in g unit. On the earth, 1g means acceleration of 9.8 m/s² is present. On moon, it is 1/6th of earth and on mars it is 1/3rd of earth. Accelerometer can be used for tilt-sensing applications as well as dynamic acceleration resulting from motion, shock, or vibration Fig 3.1.7 shows MEMS sensor.

- a) The ADXL335 gives complete 3-axis acceleration measurement.
- b) This module measures acceleration within range ± 3 g in the x, y and z axis
- c) The output signals of this module are analog voltages that are proportional to the acceleration.
- d) It contains a polysilicon surface-micro machined sensor and signal conditioning circuitry.
- e) Acceleration deflects the moving mass and unbalances the differential capacitor which results in a sensor output voltage amplitude which is proportional to the acceleration.
- f) Phase-sensitive demodulation techniques are then used to determine the magnitude and direction of the acceleration.

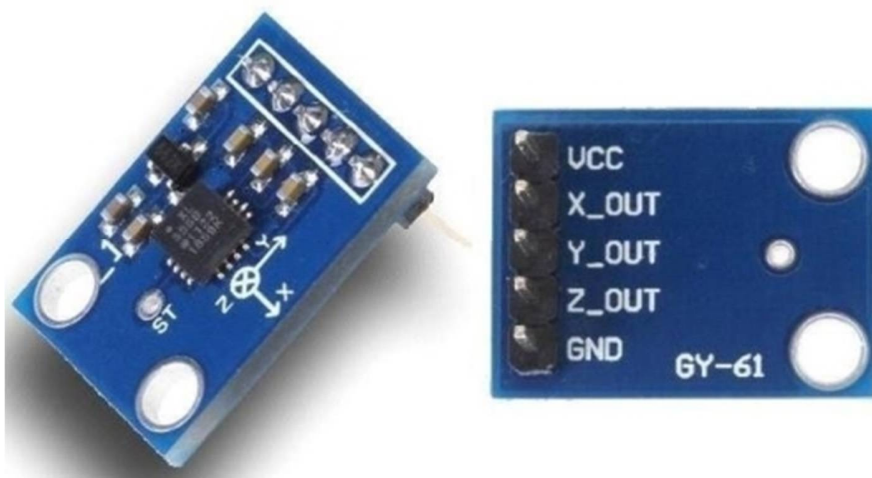


Fig 3.1.7: MEMS

8) Keypad (4x4)

Keypad is used as an input device to read the key pressed by the user and to process it. 4x4 keypad consists of 4 rows and 4 columns. Switches are placed between the rows and columns. A key press establishes a connection between the corresponding row and column, between which the switch is placed Fig 3.1.8 shows keypad.



Fig 3.1.8: keypad(4x4)

9) *GPS*

- a) GPS signals are located in RF spectrum allocated by the International Telecommunications Union (ITU) between 1559 MHz and 1610 MHz and designated as Radionavigation Satellite Service (space-to-earth) (RNSS s-e).
- b) GPS signals are centered at 1575.42 MHz and registered in the ITU to occupy spectrum basically between 1560 MHz and 1590 MHz.
- c) This accommodates transmission of the narrowband C/A-code signal, as well as the P(Y)-code and M-code signals that require more bandwidth.
- d) This same spectrum is also shared by other Global Navigation Satellite System (GNSS) service providers and their space-based augmentation services.
- e) The Russian GLONASS operates in the upper end of the RNSS band between 1590 MHz and 1610 MHz Fig 3.1.9 shows GPS module.



Fig 3.1.9: *GPS*

10) *WI-FI (ESP8266)*

WiFi module, also known as serial to WIFI module, which belongs to the transmission layer of IoT. The function is to convert serial port or TTL level into embedded module which can conforming to WiFi wireless network communication standard, with built-in wireless network protocol IEEE802. Node MCU is an open source LUA based firmware developed for ESP8266 wifi chip. By exploring functionalit with ESP8266 chip, NodeMCU firmware comes with ESP8266 Development board/kit i.e Nod MCU Development board. Node MCU Dev Kit/board consist of ESP8266 wifi enabled chip. The ESP8266 is a low- cost Wi- Fi chip developed by Express if Systems with TCP/IP protocol Using such serial protocols we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards etc. Fig 3.1.10 shows WI-FI module.

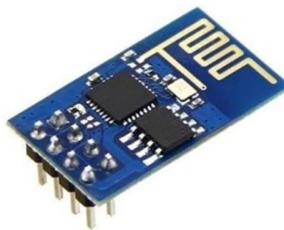


Fig 3.1.10: WI-FI(ESP8266)

B. Software Tools Description

1) *Embedded C Langauge*

Embedded C Programming Language, which is widely used in the development of Embedded Systems, is an extension of C Program Language. The Embedded C Programming Language uses the same syntax and semantics of the C Programming Language like main function, declaration of datatypes, defining variables, loops, functions, statements, etc. An Embedded System can be best described as a system which has both the hardware and software and is designed to do a specific task. As mentioned earlier, Embedded Systems consists of both Hardware and Software. If we consider a simple Embedded System, the main Hardware Module is the Processor. The Processor is the heart of the Embedded System and it can be anything like a Microprocessor, Micro-controller, DSP, CPLD (Complex Programmable Logic Device) and FPGA (Field Programmable Gated Array)



Factors for Selecting the Programming Language

The following are few factors that are to be considered while selecting the Programming Language for the development of Embedded Systems.

- Size: The memory that the program occupies is very important as Embedded Processors like Micro-controllers have a very limited amount of ROM.
- Speed: The programs must be very fast i.e. they must run as fast as possible. The hardware should not be slowed down due to a slow running software.
- Portability: The same program can be compiled for different processors.
- Ease of Implementation
- Ease of Maintenance
- Readability

2) *Arduino IDE*

The Arduino IDE is incredibly minimalistic, yet it provides a near-complete environment for most Arduino-based projects. The middle section of the IDE is a simple text editor that where we can enter the program code. The bottom section of the IDE is dedicated to an output window that is used to see the status of the compilation, how much memory has been used, any errors that were found in the program, and various other useful messages.

Projects made using the Arduino are called sketches, and such sketches are usually written in a cut-down version of C++ (a number of C++ features are not included). Because programming a microcontroller is somewhat different from programming a computer, there are a number of device-specific libraries (e.g., changing pin modes, output data on pins, reading analog values, and timers). This sometimes confuses users who think Arduino is programmed in an Arduino language. However, the Arduino is, in fact, programmed in C++. It just uses unique libraries for the device.

3) *Telegram Bots*

Bots: An introduction for developers Bots are third-party applications that run inside Telegram. Users can interact with bots by sending them messages, commands and inline requests. You control your bots using HTTPS requests to our Bot API.

a) *What can I do with bots?*

A chat with @TechCrunchBot also showing search results from the @gif inline-bot ☐☐To name just a few things, you could use bots to:

- Get customized notifications and news. A bot can act as a smart newspaper, sending you relevant content as soon as it's published.
- Integrate with other services. A bot can enrich Telegram chats with content from external services. ☐☐Gmail Bot, GIF bot, IMDB bot, Wiki bot, Music bot, YouTube bot, GitHub bot
- Accept payments from Telegram users. A bot can offer paid services or work as a virtual storefront. ☐☐Demo Shop Bot
- Create custom tools. A bot may provide you with alerts, weather forecasts, translations, formatting or other services.
- Markdown bot, Sticker bot, Votebot, Like bot
- Build single- and multiplayer games. A bot can offer rich HTML5 experiences, from simple arcades and puzzles to 3D-shooters and real-time strategy games.
- Build social services. A bot could connect people looking for conversation partners based on common interests or proximity.
- Do virtually anything else. Except for dishes — bots are terrible at doing the dishes.

b) *How do bots work?*

At the core, Telegram Bots are special accounts that do not require an additional phone number to set up. Users can interact with bots in two ways: Send messages and commands to bots by opening a chat with them or by adding them to groups. This is useful for chat bots or news bots like the official TechCrunch bot. Send requests directly from the input field by typing the bot's @username and a query. This allows sending content from inline bots directly into any chat, group or channel. Messages, commands and requests sent by users are passed to the software running on your servers. Our intermediary server handles all encryption and communication with the Telegram API for you. You communicate with this server via a simple HTTPS-interface that offers a simplified version of the

Telegram API. We call that interface our Bot API Fig 3.1.11 shows logo of Botfather used to connect to IOT System There's a... bot for that. Just talk to BotFather (described below) and follow a few simple steps. Once you've created a bot and received your authorization token, head down to the Bot API manual to see what you can teach your bot to do.

Bots have no online status and no last seen timestamps, the interface shows the label 'bot' instead. □ Bots have limited cloud storage — older messages may be removed by the server shortly after they have been processed.

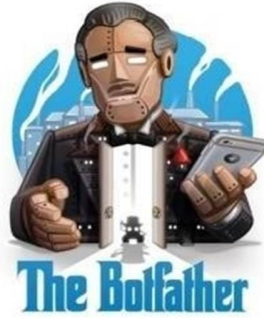


Fig 3.1.11: Botfather logo

IV. WORKING

A. Methodology

- 1) The project aims to use electronic packaging solutions to effectively trigger an alert when a package is opened.
- 2) This design has many sensors that continuously track the physical parameters inside the package to ascertain if the package has been opened or there has been some rise in temperature (for pharmaceutical and temperature-sensitive products).
- 3) Once this alert has been sent to those concerned, they can take necessary action.
- 4) This will result in an OTP that is received at the customer's mobile and entering the OTP with the kit will reset the device. 1st factor of authentication is OTP matching, so user will enter the OTP they receive through the app, onto to the keypad. This OTP will be matched with the OTP being generated by the microcontroller
- 5) The IPS kit is then removed from the box and given to the delivery executive to be reused again.

B. System Flow

The IPS development kit will be made into a compact space and placed inside the package that is to be secured. The package will have enough space for the kit to work properly. The sensors that are used in the IPS development kit will be turned on and will start working. The kit will travel within the box till the destination where it will be removed after verification of the data. Fig 4.1 shows System flow diagram.

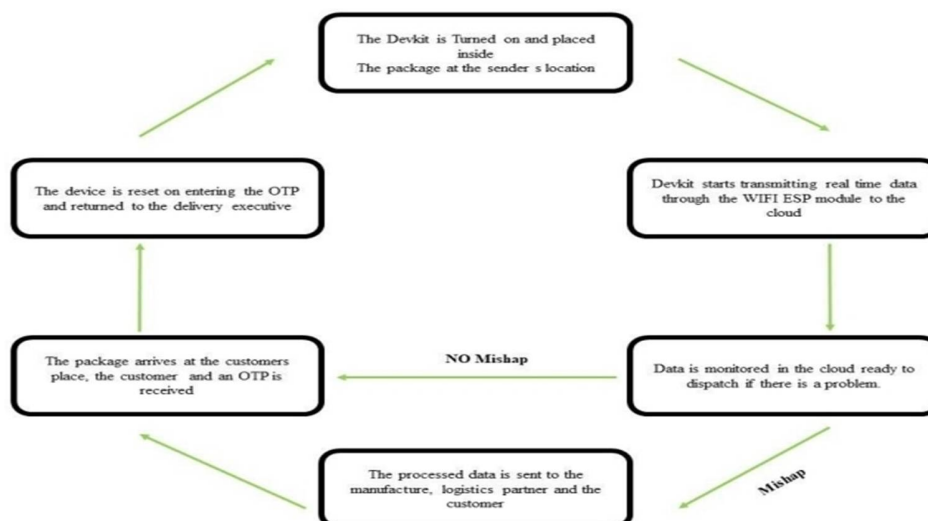


Fig 4.1: SYSTEM FLOW

C. Block Diagram

Fig 4.2 shows Block diagram of the System

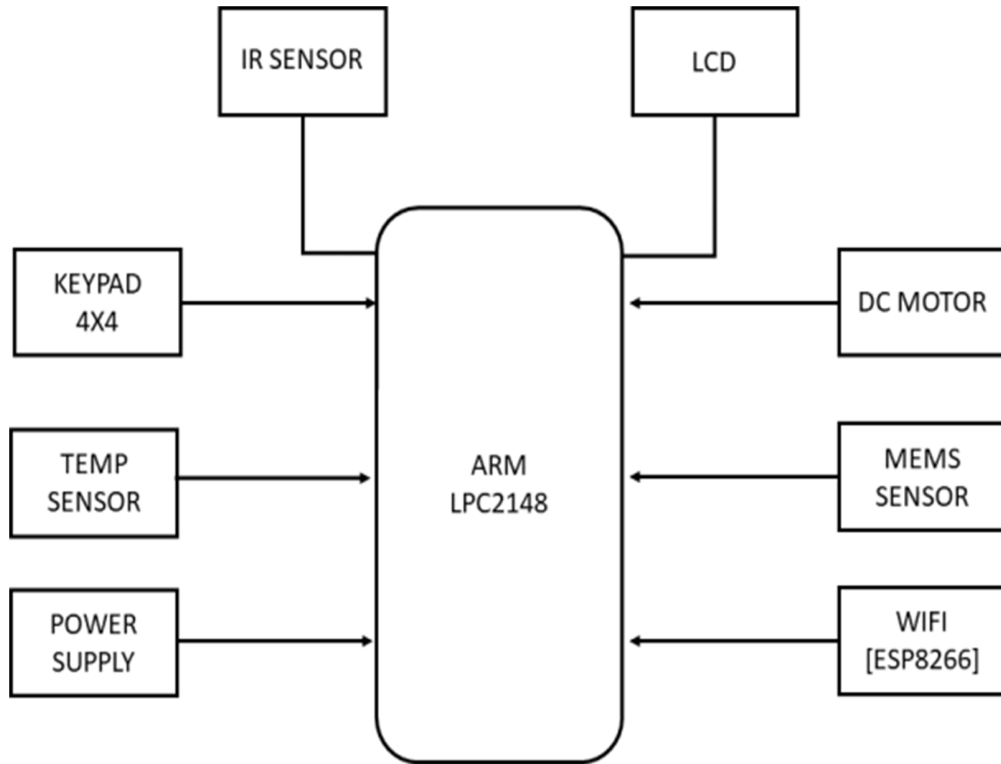


Fig 4.2: BLOCK DIAGRAM

D. Flow Chart & Description

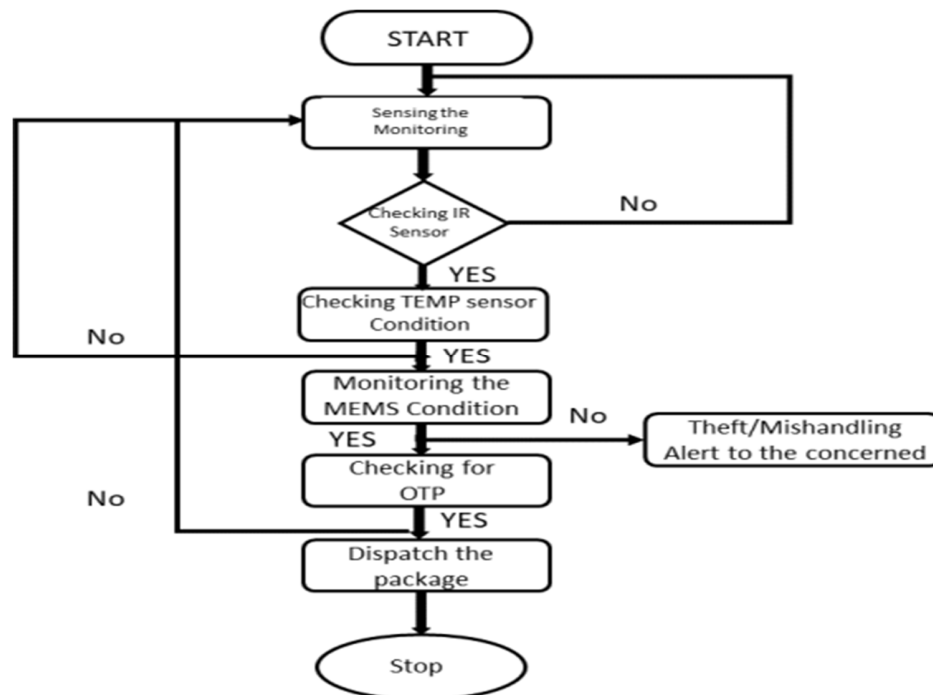


Fig 4.3: FLOW CHART

- 1) *Step 1:* The IPS development kit is first placed in the package that is to be tracked. This is placed inside the package along with the contents of the package. This is considered very easy as it is as simple as just placing the IPS development kit along with the package to be secured. For the kit to work properly and without any interruptions it must be placed as close to the package contents as possible. In the case of any food products or any hazardous chemicals that are being transported, the food products may be enclosed in another layer of packaging to protect its integrity. If the package is transporting any electronic cargo, there will be no problem in the placement of the package as there will be no electrical or electronic interference with the module. The sender must place the kit in such a way that the kit is not moving on its own inside the box. Once the kit is securely placed and it is tested for its sturdiness, the package that is containing the IPS development kit as well as the shipping contents can be sealed and locked. The IPS development kit is enabled. Fig 4.3 shows the Flow chart.
- 2) *Step 2:* The IPS development kit starts running the program that is coded into it and hence starts transmitting data to the cloud. The data on the cloud is visible on the web interface that has been built for this specific purpose. The website updates in real-time to show the change in data if any to the parties concerned. This allows them to take any immediate action that may be necessary to ensure the integrity of the package.
- 3) *Step 3:* The data is monitored in real-time on the cloud as well as in the IPS development kit. In the event if a mishap occurs: If there are any changes in the data that suggest a break in the integrity of the package if there is any mishap with respect to the package then an alert is sent to the manufacturer of the product (the sender), the logistics partner (the person shipping the parcel) as well as the customer (the person receiving the parcel). This enables them to take any action that is necessary to either stop the package in transit or to verify its integrity. If there is no mishap during transit: If there is no mishap in the entire duration that the package is being shipped, then the package arrives safely at the destination and no alert is raised to inform all the concerned parties. The customer scans a QR code that is present on the IPS development kit and the customer receives an OTP.
- 4) *Step 4:* On receiving the OTP, the IPS development kit is reset and it is ready to be taken back by the shipping company and installed on another parcel and the cycle of the IPS development kit continues.

E. Algorithm

- 1) Initially switch on the Dev-Kit
- 2) Check and verify if all the sensors are working and monitor the sensors data in real time using cloud technology
- 3) Check if IR sensor is working. If yes move on to check TEMP sensor condition and update the data on cloud, if no enter the loop to check for the sensors working again
- 4) Next move on to check for the TEMP sensor Sensor condition, if yes and its working move on to check MEMS Sensor condition and update the data of TEMP sensor sensor on cloud, if No go back to the process of sensing and monitoring and check the working of sensors again.
- 5) Next move on to check the working of MEMS sensor if yes move on to check for OTP and update the data on the cloud, if no alert the ecommerce owner for theft or mishandling of the product.
- 6) Next check for OTP and send the OTP to the customer's cellular device and ask him to enter the OTP on the Dev-Kit keypad
- 7) Dispatch the package and switch off the Dev-Kit. 8. Take the Dev-Kit back for reuse.

V. RESULTS

The below Figures from Fig 4.4 – Fig 4.12 show the working of the IPS System at each stage and also the updates recieved in the telegram simultaneously.



Fig 4.4 Picture OfThe Model



Fig 4.5 Components Inside The Model



Fig 4.6 Product to be Placed



Fig 4.7 LID is Closing



Fig 4.8 Product is Placed



Fig 4.9 X&Y-AXIS of the MEMS Sensor



Fig 4.10 Temperature Sensor Readings



Fig 4.11 WAITING FOR OTP



Fig 4.12 Updates Received on Telegram



A. Applications

IPS DevKit can be used in many industries that may transport expensive that may require utmost security. Some of them are:

- 1) Pharmaceutical Industries • Food Industries
- 2) Electronic Industries etc.

The idea of IPS kit with few modifications can be used to provide safety for belongings and to notify the owner if there is any harm, damage or misplacement. IPS as a solution can be deployed in the logistics industry based on the needs of the client and targeted products.

B. Advantages

- 1) Traceability/Supply chain management of the product will be easier and handy. □□ Products can be saved from getting damaged or being cloned.
- 2) Company's market reputation won't be harmed.
- 3) Minimum Human Intervention
- 4) Reusable.
- 5) Tamper proof delivery

C. Future Scope

Below Feature can be added

- 1) Digital nameplate
- 2) Online administration
- 3) Multiple deliveries: Multiple parcels can be delivered each day by different service providers totally without delivery delays.
- 4) Time-saving: Saves time as no need for unnecessary trips to parcel collection offices.
- 5) Fast availability: No waiting times at goods collection points and no time lost due to shipping.

VI. CONCLUSION

We are designing and implementing the non-destructive package testing and verification solution using an electronic packaging solution called IPS development kit. The kit will be placed inside the package with enough space to work properly. The kit will travel within the box till the destination where it will be removed after the delivery. If there is any tampering or intrusion during the delivery process an alert will be sent. By this we are maintaining the integrity of the product as well as the reputation of the companies that use this system and along with that, we are hoping to ensure customer satisfaction by getting them the product they requested without any problem. In the project phase 1 we have understood the hardware requirements and the methodology to implement them and integrate them to the ARM microcontroller. We will be using different kind of sensors and gaining knowledge about them and hence has given a different perspective on what else can be done using these sensors. We will be learning embedded C language in order to write the code for the working of the Dev-Kit. This will help us in future as we gain knowledge on how to write code.

VII. PHASE 2 PLANNER

With the basic study and understanding of various hardware components & software tools, The second phase shall deal with procuring the hardware, building the cloud, building the Dev-Kit and setting up the project. This will be achieved by integrating both hardware and software together to achieve the desired result.

Table 1.2: Phase 2 Planner

1 st Week	Assessing The Groundwork Of Phase-1
2 nd -3 rd Week	Procuring The Required Hardware And Software Components
4 th -5 TH Week	Designing And Building The Dev-kit
6 th -7 th Week	Designing And Developing The Cloud Database
8 th -9 th Week	Integrating The Embedded C Code To The Dev-kit
10 th -11 ^h Week	Final Testing Debugging And Refining The System
12 th Week	Final Project Presentation



VIII. ACKNOWLEDGEMENT

It would be our privilege to express our heartfelt gratitude and respect to all those who guided us in the completion of this Project Work Phase 2. We are highly indebted to Rajya Vokkaligara Sangha trust for introducing a great institution for education. We express our deep sense of gratitude to our Principal. Dr. Aswath M. U, for providing excellent academic environment which enabled us to complete the Project Work Phase 2. We would like to express our heartfelt gratitude to our Head of Department, Department of Electronics and Telecommunication Engineering, Dr.Rajeswari M, for her constant support during the tenure of the Project Work Phase 2 and throughout the course of Engineering in the Institution. We would like to express our heartfelt gratitude to our Project coordinator, Prof. Sudha.B, Assistant Professor, Department of Telecommunication Engineering, for her constant support and guidance during the course of the Project Work. We would like to express our heartfelt gratitude to our Project Internal Guide, Prof. N. Shruthi, Assistant Professor, Department of Telecommunication Engineering, for her constant support and guidance during the course of the Project Work Phase 2. We are grateful to all the Teaching and Non-Teaching Staff, Department of Electronics and Telecommunication Engineering, who have been an inspirational support. Lastly, we would like to thank our parents and friends for their constant help and moral support throughout the completion of the Project Work Phase 2.

REFERENCES

- [1] Authors: Shahriar Rahman Fahim "Development of a Remote Tracking Security Box with Multi-Factor Authentication System Incorporates with a Biometric Sensing Device" Year Of Publication: 2019
- [2] Authors: JiuWen, Yanchun Sun "Map-Matching Service Designed for Courier Trajectories" Published In: 11 September 2017, IEEE Xplore.
- [3] Authors: SaipunizamMahamad, SuziahSulaiman, Wong Yi Leng "An Integrated Courier Services Application: A New UserExperience." Date Added to IEEE Xplore: 04 February 2019 4 Authors: Wei Tu "Online Crowd sourced Delivery for On-Demand Food" Published In: 03 september 2019
- [4] Authors: Ge Wang Jinsong Han Chen " Verifiable Smart Packaging with PassiveRFID" IEEE Xplore: September 2016
- [5] http://en.wikipedia.org/wiki/Internet_of_Things



PROGRAMME OUTCOMES

A. Course Outcomes

Project Work Phase 2(18TEP83) After completion of the course, student will be able to

1. Understand, Identify and analyze a real world problem based on the societal/environmental requirements and work towards resolving it
2. Make use of literature survey, compile project requirements, evaluate existing solutions and plan for a better solution.
3. Compile the applications of the project. Discuss the extent of objectives achieved and evaluate the overall impact on society and environment,
4. Collaborate with teammates and communicate effectively to manage all aspects of the project including finance, time and resources
5. Explain the project work and demonstrate results using appropriate ICT tools in oral and written medium.
6. Understand the ethical and professional responsibility.

B. Programme Outcomes

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and engineering specialization to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences.

PO3: Design/Development of solution: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct Investigation of Complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project Management Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long Learning: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

C. Programme Specific Outcomes (PSO)

PSO1: Analyze and Design Analog & Digital modules for a given specification and function.

PSO2: Implement functional blocks of hardware-software co-designs for Embedded Systems, Signal Processing, Communication and Networking Applications



D. CO-PO-PSO Mapping

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO-1	3	3	3	2	-	2	3	-	2	3	-	2	3	3
CO-2	2	2	-	3	2	-	2	2	3	-	1	3	2	3
CO-3	2	2	2	2	-	2	1	2	2	3	-	-	1	-
CO-4	3	-	-	3	2	2	2	3	3	3	3	3	3	3
CO-5	-	1	2	-	3	-	-	1	3	2	-	2	1	-
CO-6	3	-	2	3	2	3	3	-	2	2	2	2	-	2

APPENDIX

```
#include<LPC214x.h>
#include<stdio.h>
#include "lcd.h"
#include "Serial.h"
#include "adc.h"
#define IR (1<<12)
#define IN1 (1<<13)
#define IN2 (1<<14)
void Lid_close();
void Lid_close();
void Temp_Read();
void sensor_Check();
void LDR_Check();
void Lid_open();
void Lid_close();
unsigned char keypad (void);
void WAITING_FOR_OTP();
```



```
void delay(int i);

int in=0,in1=0,in2=0;

int ldr=0,X,Y;

int gps;

int lo=0,lc=0;

int X_val,i;

int Y_val;

unsigned char x_buf[20];

unsigned char y_buf[20];

unsigned char msg_buff[100],keybuf[6],ch1;

#define r1 (1<<16)

#define r2 (1<<17)

#define r3 (1<<18)

#define r4 (1<<19)

#define c1 (1<<20)

#define c2 (1<<21)

#define c3 (1<<22)

#define c4 (1<<23)

unsigned char keyboard[] = {'0','4','8','C',

'1','5','9','D',

'2','6','A','E',

'3','7','B','$'

};

unsigned char keypressed = 0;

unsigned char key=0;
```




```
int key_num = 0,m=0,l=1,z=0;
```

```
int w=0, x=0, y=0;
```

```
int cnt1, cnt2;
```

```
#define LCD_KEY_DIR    IO1DIR
```

```
#define LCD_KEY_SET    IO1SET
```

```
#define LCD_KEY_CLR    IO1CLR
```

```
#define SCAN_FINGER (1 << 12)
```

```
#define READ_FINGER (1 << 28)
```

```
#define WAIT_FINGER (1 << 10)
```

```
#define ROW1 (1 << 11)
```

```
#define ROW2 (1 << 10)
```

```
#define ROW3 (1 << 3)
```

```
#define ROW4 (1 << 2)
```

```
#define COL1 (1 << 7)
```

```
#define COL2 (1 << 6)
```

```
#define COL3 (1 << 5)
```

```
#define COL4 (1 << 4)
```

```
#define FIRST_ROW    0
```

```
#define SECOND_ROW   4
```

```
#define THIRD_ROW
```

```
#define FOURTH_ROW
```

```
#define KEY_OUTPUT
```

```
#define KEY_INPUT
```

```
void init_key();
```

```
8
```

```
12
```



```
(COL1 |COL2 |COL3 | COL4);
```

```
(ROW1 |ROW2 |ROW3 | ROW4);
```

```
void scan_row1();
```

```
void scan_row2();
```

```
void scan_row3();
```

```
void scan_row4();
```

```
unsigned char scan_key();
```

```
void scan_row(unsigned char i);
```

```
void putSLcd(unsigned char *st)
```

```
{
```

```
for( ;*st ;lcd_putchar(*st++) );
```

```
}
```

```
void init_key()
```

```
{
```

```
LCD_KEY_DIR = (ROW1 |ROW2 |ROW3 | ROW4); //rows as OUTPUT
```

```
LCD_KEY_DIR = ~(COL1 |COL2 |COL3 | COL4); //columns as INPUT
```

```
LCD_KEY_SET = (ROW1 |ROW2 |ROW3 | ROW4);
```

```
}
```

```
unsigned char scan_key()
```

```
{
```

```
scan_row1();
```

```
scan_row2();
```

```
scan_row3();
```

```
scan_row4();
```

```
return key;
```



```
}  
  
void scan_row1()  
  
{  
  
LCD_KEY_CLR = ROW1;  
  
delay(20);  
  
scan_row(FIRST_ROW);  
  
LCD_KEY_SET = ROW1;  
  
delay(20);  
  
}  
  
void scan_row2()  
  
{  
  
LCD_KEY_CLR = ROW2;  
  
delay(20);  
  
scan_row(SECOND_ROW);  
  
LCD_KEY_SET = ROW2;  
  
delay(20);  
  
}  
  
void scan_row3()  
  
{  
  
LCD_KEY_CLR = ROW3;  
  
delay(20);  
  
scan_row(THIRD_ROW);  
  
LCD_KEY_SET = ROW3;  
  
delay(20);  
  
}
```



```
void scan_row4()
{
LCD_KEY_CLR = ROW4;
delay(20);
scan_row(FOURTH_ROW);
LCD_KEY_SET = ROW4;
delay(20);
}

void scan_row(unsigned char i)
{
if(!(IO1PIN & COL1))
{
while(!(IOPIN1 & COL1))
{
continue;
}
key_num = keyboard[0+i];
key = keyboard[0+i];
keypressed = 1;
}
else if(!(IOPIN1 & COL2))
{
while(!(IOPIN1 & COL2))
{
continue;
}
}
```



```
key_num = keyboard[1+i];  
  
key = keyboard[1+i];  
  
keypressed = 1;  
  
}  
  
else if(!(IOPIN1 & COL3))  
  
{  
  
while(!(IOPIN1 & COL3))  
  
{  
  
continue;  
  
}  
  
key_num = keyboard[2+i];  
  
key = keyboard[2+i];  
  
keypressed = 1;  
  
}  
  
else if(!(IOPIN1 & COL4))  
  
{  
  
while(!(IOPIN1 & COL4))  
  
{  
  
continue;  
  
}  
  
key_num = keyboard[3+i];  
  
key = keyboard[3+i];  
  
keypressed = 1;  
  
}  
  
}
```




```
int main()

{

I00DIR &=~ IR;
I00DIR |= IN1;

I00DIR |= IN2;

init_lcd();

init_adc0();

uart1_init();

lcd_clear();

lcd_putstring(0,0,"INTELLIGENT");

lcd_putstring(0,1,"PACKAGE SYSTEM");

delay(10000);

init_key();

while(1) // Repeat(loop) forever
{

// lcd_data_write(keypad());

sensor_Check();

Temp_Read();

}

}

void MESSAGE_RECEIVE_PERMISSION()

{

while(uart1_getkey()!='*');

i=0;

while((ch1=uart1_getkey())!='#')
```



```
{  
  
msg_buff[i] = ch1;  
  
i++;  
  
}  
  
msg_buff[i]='\0';  
  
uart1_puts(msg_buff);  
  
lcd_clear();  
  
lcd_putstr(0,0,"OTP:");  
  
lcd_putstr(1,0,msg_buff );  
  
}  
  
void WAITING_FOR_OTP(void)  
  
{  
  
unsigned char ch;  
  
lcd_command_write(1);          // Entry mode  
  
l=1;  
  
m=0;  
  
z=1;  
  
while(l)  
  
{  
  
lcd_clear();  
  
lcd_putstr(0,0,"WAITING FOR:");  
  
lcd_putstr(1,0,"OTP....." );  
  
MESSAGE_RECEIVE_PERMISSION();  
  
delay(10000);  
  
m=0;
```



```
if(z==1)

{

while(1)
z=0;

lcd_clear();

lcd_putstring(0,0,"ENTER OTP:");

{

key = scan_key();

if(keypressed == 1)

{

lcd_putchar(key);

keypressed = 0;

keybuf[m++]=key;

delay(1000);

if(m==4)

{

keybuf[m]='\0';

l=0;

}

}

delay(1000);

}

while(1 )

{

if((strcmp((char *)msg_buff,keybuf))==0)

{
```



```
// bank_selection();

lcd_command_write(1);           // Entry mode

lcd_putstring(0,0,"VALID OTP");

l=1;

m=0;

z=1;

Lid_open();

}

}

}

}

}

void Lid_open(){

IO0CLR |= IN1;

IO0SET |= IN2;

delay(10000);

IO0CLR |= IN1;

IO0CLR |= IN2;

delay(10000);

}

void Lid_close(){

IO0SET |= IN1;

IO0CLR |= IN2;

delay(10000);

IO0CLR |= IN1;
```



```
IO0CLR |= IN2;

delay(10000);

}

void sensor_Check()

{

if(lo==0)

{

lo=1;

lcd_clear();

lcd_putstring(0,0,"PLACE OBJECT");

lcd_putstring(1,0,"INSIDE");

delay(10000);

}

if((IO0PIN&IR)==0 && in==0)

{

in=1;

lcd_clear();

lcd_putstring(0,0,"OBJECT PLACED");

delay(10000);

lcd_clear();

lcd_putstring(0,0,"REMOVE HANDS");

delay(10000);

lcd_clear();

lcd_putstring(0,0,"LID CLOSING");

delay(10000);
```




```
Lid_close();

}

if(in==1 && in1==0)

{

LDR_Check();

}

if(in1==1)

{

gps++

X_val = adc_read(ADC0,CHANNEL_3);

sprintf(x_buf,"X:%d",X_val);

Y_val = adc_read(ADC0,CHANNEL_4);

sprintf(y_buf,"Y:%d",Y_val);

delay(10000);

lcd_clear();

lcd_putstr(0,0,x_buf);

lcd_putstr(1,0,y_buf);

delay(10000);

if(gps==5 || gps==10 || gps==15 || gps==19)

{

Temp_Read();

LDR_Check();

// lcd_clear();

// lcd_putstr(0,0,"LDR VALUE:");

// lcd_putstr(1,0,Ldr_buf);
```



```
if(!(X_val>268 && X_val<268 && Y_val>340 && Y_val<350)
)
{
// lcd_clear();

// lcd_putstring(0,0,X_val);

// lcd_putstring(1,0,Y_val);

lcd_clear();

lcd_putstring(0,0,"THE POSITION OF");

lcd_putstring(1,0,"PACKAGE CHANGING");

delay(1000);

}

}

uart1_puts("$OTP#");

WAITING_FOR_OTP();

}

}

void Temp_Read()

{

short Temp_val;

float Temp;

unsigned char Temp_buf[20];

Temp_val=adc_read(ADC0,CHANNEL_1);

Temp=Temp_val*3.3/1023;

Temp=Temp*10;

sprintf(Temp_buf,"Temp:%0.2f Deg.Celcius\n",Temp);

lcd_clear();
```



```
lcd_putstring(0,0,Temp_buf);

delay(10000);

}

void LDR_Check(){

// int Ldr_val;

//int ldr;

unsigned char Ldr_buf[20];

ldr = adc_read(ADC0,CHANNEL_2);

sprintf(Ldr_buf,"Ldr:%d PPM",ldr);

delay(10000);

lcd_clear();

lcd_putstring(0,0,Ldr_buf);

delay(10000);

if(ldr>=85)

{

lcd_putstring(0,0,"SOMEONE TRY TO");

lcd_putstring(1,0,"OPEN THE PACKAGE");

delay(10000);

}

// if(Ldr_val>850)

// {

// lcd_clear();

// lcd_putstring(0,0,"OPEN BOX");

// delay(10000);

// }
```



```
// else

// {

//     lcd_clear();

//     lcd_putstring(0,0,"SEALED BOX");

//     delay(10000);

// }

}

//unsigned char keypad (void)                // keypad operation function

// {

//     unsigned char key;

//     IOCLR1|=(c1|c2|c3|c4|r1|r2|r3|r4);

//     while(1)

//     {

//         IOCLR1|=c1;

//         IOSET1|=(c2|c3|c4);                // first column = 0

//         if((IOPIN1&r1)==0)

//         {

//             key='7';

//             delay(10000);

//             return key;

//         }

//         else if((IOPIN1&r2)==0)

//         {

//             key='8';

//             delay(10000);
```



```
//          return key;

//      }

//      else if((IOPIN1&r3)==0)

//      {

//          key='9';

//          delay(10000);

//          return key;

//      }

//      else if((IOPIN1&r4)==0)

//      {

//          key='/';

//          delay(10000);

//          return key;

//      }

//      IOCLR1|=c2;

//      IOSET1|=(c1|c3|c4);          // second column = 0

//

//      if((IOPIN1&r1)==0)

//      {

//          key='4';

//          delay(10000);

//          return key;

//      }

//      else if((IOPIN1&r2)==0)

//      {
```




```
//      key='5';

//      delay(10000);

//      return key;

//      }

//      else if((IOPIN1&r3)==0)

//      {

//      key='6';

//      delay(10000);

//      return key;

//      }

//      else if((IOPIN1&r4)==0)

//      {

//      key='*';

//      delay(10000);

//      return key;

//      }

//      IOCLR1|=c3;

//      IOSET1|=(c1|c2|c4);      // third column = 0

//      if((IOPIN1&r1)==0)

//      {

//      key='1';

//      delay(10000);

//      return key;

//      }

//      else if((IOPIN1&r2)==0)
```



```
//      {  
  
//      key='2';  
  
//      delay(10000);  
  
//      return key;  
  
//      }  
  
//      else if((IOPIN1&r3)==0)  
  
//      {  
  
//      key='3';  
  
//      delay(10000);  
  
//      return key;  
  
//      }  
  
//      else if((IOPIN1&r4)==0)  
  
//      {  
  
//      key='-';  
  
//      delay(10000);  
  
//      return key;  
  
//      }  
  
//      IOCLR1|=c4;  
  
//      IOSET1|=(c1|c2|c3);          // forth column = 0  
  
//  
  
//      if((IOPIN1&r1)==0)  
  
//      {  
  
//      lcd_command_write(0x01);  
  
//      delay(10000);  
  
//      }  

```



```
//     else if((IOPIN1&r2)==0)

//     {

//         key='0';

//         delay(10000);

//         return key;

//     }

//     else if((IOPIN1&r3)==0)

//     {

//         key='=';

//         delay(10000);

//         return key;

//     }

//     else if((IOPIN1&r4)==0)

//     {

//         key='+';

//         delay(10000);

//         return key;

//     }

// }

// }

void delay(int i)

{

int j,k;

for(j=0;j<i;j++)

for(k=0;k<1275;k++);
```



}

Lcd

```
#include <LPC21xx.H>          /* LPC214x definitions */
```

```
#include "lcd.h"
```

```
#define LCD_BACK_LIGHT_TIMEOUT 1000
```

```
#define LCD_DATA_DIR  IO0DIR
```

```
#define LCD_DATA_SET      IO0SET
```

```
#define LCD_DATA_CLR      IO0CLR
```

```
#define LCD_CTRL_DIR  IO0DIR
```

```
#define LCD_CTRL_SET   IO0SET
```

```
#define LCD_CTRL_CLR   IO0CLR
```

```
#define LCDRS
```

```
#define LCDRW
```

```
#define LCDEN
```

```
(1 << 15)
```

```
(1 << 16)
```

```
(1 << 17)
```

```
#define LCD_D4 (1 << 18)
```

```
#define LCD_D5 (1 << 19)
```

```
#define LCD_D6 (1 << 20)
```

```
#define LCD_D7 (1 << 21)
```

```
#define LCD_DATA_MASK
```

```
#define LCD_BUSY_FLAG
```

```
#define LCD_IO0PIN
```

```
/**
```

```
(LCD_D4 | LCD_D5 | LCD_D6 | LCD_D7)
```

```
LCD_D7
```



IOOPIN

*****1111

Function Name : delay()

Description :This function suspends the tasks for specified ticks.

Input : ticks:no of ticks in multiple of 1 usec

task: task to be suspended

Output : void

Note :

```
*/  
  
void delay_1(int count)  
{  
int j=0,i=0;  
for(j=0;j<count;j++)  
{  
/* At 60Mhz, the below loop introduces  
delay of 10 us */  
for(i=0;i<35;i++);  
}  
}  
/**  
  
***** **
```

Function Name : wait_lcd()

Description :

Input :



Output : Void

Note :

***** **

```
*/  
  
void wait_lcd( void )  
{  
LCD_DATA_DIR &= ~LCD_BUSY_FLAG;  
LCD_CTRL_CLR |= LCDRS;  
LCD_CTRL_SET |= LCDRW |LCDEN;  
while(LCD_IOPIN & LCD_BUSY_FLAG);          /* wait for busy flag to become low */  
LCD_CTRL_CLR |= LCDEN | LCDRW;  
LCD_DATA_DIR |= LCD_DATA_MASK;  
delay_1(100);  
}  
/**
```

***** **

Function Name : lcd_command_write()

Description :

Input :

Output : Void

Note :

***** **

```
*/  
  
void lcd_command_write( unsigned char command )  
{
```



```
unsigned char temp=0;

unsigned int temp1=0;

temp=command;

temp=(temp>>4)&0x0F;

temp1=(temp<<18)&LCD_DATA_MASK;

LCD_CTRL_CLR = LCDRS;

LCD_CTRL_SET = LCDEN;

LCD_DATA_CLR = LCD_DATA_MASK;

LCD_DATA_SET = temp1;

delay_1(10000);

LCD_CTRL_CLR = LCDEN;

temp=command;

temp&=0x0F;

temp1=(temp<<18)&LCD_DATA_MASK;

delay_1(100*2);

LCD_CTRL_CLR |= LCDRS;

LCD_CTRL_SET |= LCDEN;

LCD_DATA_CLR = LCD_DATA_MASK;

LCD_DATA_SET = temp1;

delay_1(10000);

LCD_CTRL_CLR |= LCDEN;

wait_lcd();

}

/**

***** **
```



Function Name : set_lcd_port_output()

Description :

Input :

Output : Void

Note :

***** **

*/

```
void set_lcd_port_output( void )
```

```
{
```

```
LCD_CTRL_DIR |= ( LCDEN | LCDRS | LCDRW );
```

```
LCD_CTRL_CLR |= ( LCDEN | LCDRS | LCDRW );
```

```
LCD_DATA_DIR |= LCD_DATA_MASK;
```

```
}
```

```
/* *
```

***** **

Function Name : lcd_clear()

Description :

Input :

Output : Void

Note :

***** **

*/

```
void lcd_clear( void)
```

```
{
```

```
lcd_command_write( 0x01 );
```



```
}  
  
/**  
  
***** **  
  
Function Name :      lcd_gotoxy()  
  
Description :  
  
Input      :  
  
Output     :      Void  
  
Note      :  
  
***** **  
  
*/  
  
int lcd_gotoxy( unsigned int x, unsigned int y)  
  
{  
  
int retval = 0;  
  
if( (x > 1) && (y > 15) )  
  
{  
  
retval = -1;  
  
} else {  
  
if( x == 0 )  
  
{  
  
lcd_command_write( 0x80 + y );  
  
} else if( x==1 ){  
  
lcd_command_write( 0xC0 + y );  
  
}  
  
}  
  
return retval;
```



```
/* command - position cursor at 0x00 (0x80 + 0x00) */
```

```
/* command - position cursor at 0x40 (0x80 + 0x00) */
```

```
}
```

```
/**
```

```
***** **
```

Function Name : lcd_data_write()

Description :

Input :

Output : Void

Note :

```
***** **
```

```
*/
```

```
void lcd_data_write( unsigned char data )
```

```
{
```

```
unsigned char temp=0;
```

```
unsigned int temp1=0;
```

```
temp=data;
```

```
temp=(temp>>4)&0x0F;
```

```
temp1=(temp<<18)&LCD_DATA_MASK;
```

```
LCD_CTRL_SET |= LCDEN|LCDRS;
```

```
LCD_DATA_CLR = LCD_DATA_MASK;
```

```
LCD_DATA_SET = temp1;
```

```
LCD_CTRL_CLR |= LCDEN;
```

```
temp=data;
```



```
temp&=0x0F;

temp1=(temp<<18)&LCD_DATA_MASK;

LCD_CTRL_SET |= LCDEN|LCDRS;

LCD_DATA_CLR = LCD_DATA_MASK;

LCD_DATA_SET = temp1;

LCD_CTRL_CLR |= LCDEN;

wait_lcd();

}
```

Function Name : lcd_putchar()

Description :

Input :

Output : Void

Note :

*/

```
void lcd_putchar( int c )
```

```
{
    lcd_data_write( c );
}
```

Function Name : lcd_putstring()

Description :



Input :

Output : Void

Note :

***** **

*/

```
void lcd_putstring( unsigned char line,unsigned char pos, char *string )
```

```
{
```

```
unsigned char len = MAX_CHAR_IN_ONE_LINE; lcd_gotoxy
```

```
( line,pos);
```

```
while(*string != '\0' && len--)
```

```
{
```

```
lcd_putchar( *string );
```

```
string++;
```

```
}
```

```
}
```

```
/**
```

***** **

Function Name : lcd_backlight_on()

Description :

Input :

Output : Void

Note :

***** **

*/



```
/**
```

```
***** **
```

```
Function Name : turn_off_lcd_back_light()
```

```
Description :
```

```
Input      :
```

```
Output     : Void
```

```
Note      :
```

```
***** **
```

```
*/
```

```
/**
```

```
***** **
```

```
Function Name :      init_lcd()
```

```
Description :
```

```
Input      :
```

```
Output     :   Void
```

```
Note      :
```

```
***** **
```

```
*/
```

```
void init_lcd( void )
```

```
{
```

```
set_lcd_port_output();
```

```
delay_1(100*2);
```

```
lcd_command_write(0x28);    /* 4-bit interface, two line, 5X7 dots.    */
```

```
lcd_command_write(0x02);
```

```
lcd_command_write(0x06);
```

```
/* cursor home                */
```



```
/* Entry mode */

lcd_command_write(0x0F); /* display on cursor blinking */

}

/**

*****

Function Name :      test_cmd()

Description :

Input      :

Output     :      Void

Note      :

*****

*/

void test_cmd(void)

{

lcd_command_write(0x38);

lcd_command_write(0x08);

lcd_command_write(0x0C);

lcd_command_write(0x10);

lcd_command_write(0x14);

lcd_command_write(0x01);

}

// Interface length 8-bit 2 line

//Display off,cursor off

//Display on cursor off

//Move cursor left by 1 char

//Move cursor right by 1 char

//Clear display
```



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)