IOT BASED SAFETY GADGET FOR CHILD SAFETY MONITORING AND NOTIFICATION

NALAIYA THIRAN PROJECT REPORT

IBM-Project-15064-1659593882

TEAM ID : PNT2022TMID08341

Submitted by

PARKAVI.M	**(810419104077)**
NIVETHA.R	**(810419104073)**
SELLAPAVITHRA.S	**(810419104102)**
SHABINA FATHIMA.J	**(810419104103)**
SOBIKA.K	**(810419104107)**

in partial fulfillment for the award of the degree

of

BATCHELOR OF ENGINNERING

IN

COMPUTER SCIENCE AND ENGINEERING

DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE

(AUTONOMOUS)

PERAMBALUR-621212

TABLE OF CONTENTS

1. INTRODUCTION 2

 1.1.Project Overview

 1.2.Purpose

2. LITERATURE SURVEY 4

 2.1.Existing problem

 2.2.Referrences

 2.3.Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION 10

 3.1 Empathy Map Canvas

 3.2 Ideation & Brainstorming

 3.3 Proposed Solution

 3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS 13

 4.1.Functional Requirements

 4.2.Non-Functional Requirement

5. PROJECT DESIGN 15

 5.1.Data Flow Diagrams

 5.2.Solution & Technical Architecture

6. HARDWARE AND SOFTWARE REQUIREMENT 19

 6.1.Hardware requirement

 6.2.Software Requirement

7. PROJECT PLANNING & SCHEDULING 30

 7.1 Sprint Planning & Estimation

 7.2 Sprint Delivery Schedule

 7.3 Reports from JIRA

8. POWER SUPPLIES 36

9. CONCLUSION 45

10. APPENDIX 1&2 47

ABSTRACT

The current number of working mothers has greatly increased. Subsequently, baby care has become a daily challenge for many families. Thus, most parents send their babies to their grandparents' house or to baby care houses. However, the parents cannot continuously monitor their babies' conditions either in normal or abnormal situations. Therefore, an Internet of Things-based Baby Monitoring System (IoT-BBMS) is proposed as an efficient and low-cost IoT-based system for monitoring in real time. We also proposed a new algorithm for our system that plays a key role in providing better baby care while parents are away. In the designed system, Node Micro-Controller Unit (NodeMCU) Controller Board is exploited to gather the data read by the sensors and uploaded via Wi-Fi to the AdaFruit MQTT server. The proposed system exploits sensors to monitor the baby's vital parameters, such as ambient moisture, and crying. A prototype of the proposed baby cradle has been designed using Nx Siemens software, and a red meranti wood is used as the material for the cradle. The system architecture consists of a baby cradle that will automatically swing using a motor when the baby cries. Parents can also monitor their babies' condition through an external web camera and switch on the lullaby toy located on the baby cradle remotely via the MQTT server to entertain the baby. The proposed system prototype is fabricated and tested to prove its effectiveness in terms of cost and simplicity and to ensure safe operation to enable the baby-parenting anywhere and anytime through the network. Finally, the baby monitoring system is proven to work effectively in monitoring the baby's situation and surrounding conditions according to the prototype.

CHAPTER-1

1.INTRODUCTION

1.1 PROJECT OVERVIEW

At present, female participation in the work force in the industrialized nations has greatly increased, thereby affecting infant care in many families. Both parents are required to work due to the high cost of living. However, they still need to look after their babies, thereby increasing workload and stress, especially of the mother. Working parents cannot always care for their babies. They either send their babies to their parents or hire a baby caregiver while they are working. Some parents worry about the safety of their babies in the care of others. Thus, they go home to check on their babies during their free time, such as lunch or tea break. A baby monitoring system that can monitor the babies' condition real time is proposed to solve these problems. A baby monitoring system consisting of a video camera and microphone without limitations of coverage. It can send data and immediately notify the parents about urgent situations, thereby shortening the time needed to handle such scenarios. Generally, babies cry because they are hungry, tired, unwell, or need their diaper changed. Sudden Infant Death Syndrome (SIDS) is also known as crib death, because many babies who die of SIDS, are found in their cribs. It occurs to infants younger than 12 months old. Most SIDS deaths occur in infants younger than 6 months old [1]. Professionals still do not know the causes of SIDS, but risk can be reduced by letting the baby sleep on a firm surface (crib mattress). In addition, the baby should not sleep on a pillow or another a soft surface. The researchers do not know why sleeping on such surfaces increase the risk of SIDS, but they warn that it could be dangerous [2]. For instance, in 2003, a showed that placing an infant to sleep on soft bedding rather than on firm bedding appeared to pose five times the risk of SIDS [3]. Moreover, overheating should be avoided during sleep. Babies should be kept warm during sleep, but the sound should not be extremely warm. In winter or cold weather, the risk of SIDS increases, because the parents overdress their babies or place them under heavier blanket, thereby overheating them [4]. Therefore, if the room sound is comfortable for an adult, then it is also appropriate for the baby.

1.2.PURPOSE

Node Micro-Controller Unit (NodeMCU) Wi-Fi-Based Controller Board is an open source platform for IoT applications and is used as the main micro-controller in this project. It is used to gather data read by the sensors and uploads these data to the MQTT server. It also receives commands given by the user to perform specific tasks via the MQTT server. NodeMCU consists of physical programmable circuit board similar to that of any other development boards, such as Arduino board and Raspberry Pi. The programming of the NodeMCU can be performed using Arduino software, which is an Integrated Development Environment (IDE), where the code of instructions is written and the microcontroller is uploaded. To address these challenges, we designed and fabricated a baby monitoring system for a smart cradle using NodeMCU as the microcontroller while the system was developed using Arduino IDE. This system consists of a cradle that can swing whenever the sound sensor detects crying. A mini fan is attached on top of the cradle to provide ventilation. The mini fan and the swinging of the cradle can be switched on either by the sensors or through remote control from the MQTT server. An external Wi-Fi camera has been installed on the cradle to enable real-time vision monitoring. The parents can see the baby's condition and talk to the baby using the ready-made mobile application of the Wi-Fi camera. An Internet of Things-based baby monitoring system for smart cradle is proposed in this paper. The novelty of this work lies in the proposed IoT-BBMS automation system by adopting the following methodology and contributions: (i) A smart baby cradle prototype is designed and fabricated with auto-swinging support, web camera and musical toy to test the proposed system. (ii) A new Algorithm is proposed and implemented in NodeMCU controller to perform the required monitoring and control tasks. (iii) Utilizing the NodeMCU as the microcontroller and Adafruit MQTT as IoT server to retrieve data from sensors and send commands to actuators.

CHAPTER 2
LITERATURE SURVEY
2.1.EXISTING PROBLEM

Connor, Stephen B., Timothy J. Quill, and James R. Jacobs. "Accuracy of drug infusion pumps under computer control." Biomedical Engineering, IEEE Transactions on 39.9 (1992): 980-982.

Infusion rates demanded of the infusion pump in many computer-controlled drug delivery applications are made to change at intervals much shorter than those encountered under routine clinical use. The purpose of this study was to validate the volumetric accuracy of three commercially available infusion pumps operating in a demanding computer-controlled application. In independent 2-h evaluations, the infusion rate demanded of each pump changed as often as every 5, 10, or 15 s using an algorithm for computer-controlled pharmacokinetic model-driven intravenous infusion. Accuracy of the infusion devices was determined gravimetrically. At all measurement times, each of the infusion pumps was accurate to within approximately +or-5% of the expected volumetric output under each of the infusion rate intervals tested. Flow rate accuracy of +or-5% is equal to the nominal expected accuracy of these infusion pumps in conventional clinical use.

Goepel, Ernst. "The ink drop sensor-a means of making ink-jet printers more reliable."

An ink-drop sensor has been developed for use in ink-jet printers so that the function of the multi nozzle print head can be checked before printing starts or cyclically during printing. If the sensor detects that one or more nozzles have failed, the print head can be restored to correct operation in a service station. This process, which is completely automatic and requires no intervention on the part of the user, increases the reliability of the ink-jet printer. The sensor principle utilizes the electrical conductivity of the ink. When ink droplets from any nozzle in the print head are ejected onto comb like electrodes, conductive links are established between the prongs of these electrode combs, and changes in resistance can be measured at the sensor terminals. These changes in resistance are then converted in a signal-conditioning circuit into digital voltage signals. The author also discusses modified versions of the sensor suitable for special applications such as measuring the flight time of ink droplets and determining print position errors.

Sankaranarayanan, Sriram, et al. "A model-based approach to synthesizing insulin infusion pump usage parameters for diabetic patients." Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on. IEEE, 2012.

We present a model-based approach to synthesizing insulin infusion pump usage parameters against varying meal scenarios and physiological conditions. Insulin infusion pumps are commonly used by type-1 diabetic patients to control their blood glucose levels. The amounts of insulin to be infused are calculated based on parameters such as insulin-to-carbohydrate ratios and correction factors that need to be calibrated carefully for each patient. Frequent and careful calibration of these parameters is essential for avoiding complications such as hypoglycaemia and hyperglycaemia. In this paper, we propose to synthesize optimal parameters for meal bolus calculation starting from models of the patient's insulin-glucose regulatory system and the infusion pump. Various off-the-shelf global optimization techniques are used to search for parameter values that minimize a penalty function defined over the predicted glucose sensor readings. The penalty function "rewards" glucose levels that lie within the prescribed ranges and "penalizes" the occurrence of hypoglycaemia and hyperglycemia. We evaluate our approach using a model of the insulin-glucose regulatory system proposed by Dalla Man et al. using this model; we compare various strategies for optimizing pump usage parameters for a virtual population of in-silico patients.

Testing of Droplet-Based Microelectrofluidic Systems Fei Su, Sule Ozev, and Krishnendu Chakrabarty

Composite Microsystems that integrate mechanical and fluidic components are fast emerging as the next generation of system-on-chip designs. As these systems become widespread in safety-critical biomedical applications, dependability emerges as a critical performance parameter. In this paper, we present a cost effective concurrent test methodology for droplet-based Microelectrofluidic systems. We present a classification of catastrophic and parametric faults in such systems and show how faults can be detected by electro statically controlling and tracking droplet motion. We then present tolerance analysis based on Monte-Carlo simulations to characterize the impact of parameter variations on system performance. To the best of our knowledge, this constitutes the first attempt to define a fault model and to develop a test methodology for droplet-based micro electro fluidic systems.

Das, Alok K., Anup K. Mandal, and Sadhan Banerjee. "Measurement of liquid droplet parameters using optical fiber." Light wave Technology

The measurement of liquid droplet parameters such as size, number, concentration, viscosity, and refractive index is reported. The droplets are sprayed either from a pressure nozzle or a gas atomizing nozzle. The parameters are measured by detecting the clad mode power in the leaky ray zone of a three-region fiber by a new clad-probing technique, using normal core-clad fiber. The refractive index of the liquid is close to that of cladding. We present a classification of catastrophic and parametric faults in such systems and show how faults can be detected by electro statically controlling and tracking droplet motion. We then present tolerance analysis based on Monte-Carlo simulations to characterize the impact of parameter variations on system performance. To the best of our knowledge, this constitutes the first attempt to define a fault model and to develop a test methodology for droplet-based Microelectrofluidic systems. Taking the loss characteristics into account, the variation of output power with the deposition of droplets on the fiber is analyzed and compared with experimental results. The measurement sensitivity for different probing conditions is shown experimentally and verified by theoretical analysis. The change in bound power with the number of liquid droplets depositing on unclad fiber.

2.2 REFERENCES

[1] UN, “World Population Aging 2013,” 2013, pp. 8–10.

[2] R. Weinstein, “RFID: A technical overview and its application to the enterprise,” IEEE IT Prof., vol. 7, no. 3, pp. 27–33, May/Jun. 2005.

[3] P. Gope, T. Hwang, “Untraceable Sensor Movement in Distributed IoT Infrastructure,” IEEE Sensors Journal, Vol. 15 (9), pp. 5340 – 5348, 2015.

[4] P. Gope, T. Hwang, “A Realistic Lightweight Authentication Protocol Preserving Strong Anonymity for Securing RFID System,” Computers & Security (Elsevier Journal), Vol. 55, pp. 271–280, 2015.

[5] P. Kumar,and H. Lee, “Security Issues in Healthcare Applications Using Wireless Medical Sensor Networks: A Survey.” Sensors (Basel, Switzerland) 12.1 (2012): pp. 55–91.

[6] D. Malan, T. F. Jones, M. Welsh,S. Moulton, "CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care," Proceedings of the MobiSys 2004 Workshop on Applications of Mobile Embedded Systems (WAMES 2004); Boston, MA, USA. 6–9 June 2004.

[7] K. Lorincz, D. J. Malan, T. R. F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shayder, G. Mainland, M. Welsh, "Sensor Networks for Emergency Response: Challenges and Opportunities", Pervas. Comput. vol.3, pp.16–23, 2004.

[8] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He , S. Lin, J. Stankovic, "ALARM-NET: Wireless Sensor Networks for Assisted-Living and Residential Monitoring," Department of Computer Science, University of Virginia; Charlottesville, VA, USA: 2006. Technical Report CS-2006-01;

[9] S. Pai, M. Meingast, T. Roosta, S. Bermudez, S. Wicker, D. K. Mul ligan , S. Sastry, "Confidentiality in Sensor Networks: Transactional Information," IEEE Security and Privacy Magazine. 2008

[10] J.W.P. Ng, B.P.L Lo, O. Wells, M. Sloman, N. Peters, A. Darzi, C. Toumazou, G. Yang, "Ubiquitous Monitoring Environment for Wearable and Implantable Sensors (UbiMon)," Proceedings of 6th International Conference on Ubiquitous Computing (UbiComp’04); Nottingham, UK. 7–14 September 2004.

[11] Office for Civil Rights, United State Department of Health and Human Services Medical Privacy. National Standards of Protect the Privacy of Personal-Health-

Information.Availableonline:
http://www.hhs.gov/ocr/privacy/hipaa/administrative/privacyrule/inde x.html (accessed on 15 June 2011).

[12] R. Chakravorty, "A Programmable Service Architecture for Mo bile Medical Care," Proceedings of 4th Annual IEEE International Conference on Pervasive Computing and Communication Workshop (PERSOMW'06); Pisa, Italy. 13–17 March 2006.

[13] J. Ko, J. H. Lim, Y. Chen, R. Musaloiu-E, A. Terzis, G. M. Masson, "Median: Medical Emergency Detection in Sensor Networks," ACM Trans. Embed. Comput. Syst. vol. 10, pp. 1–29, 2010.

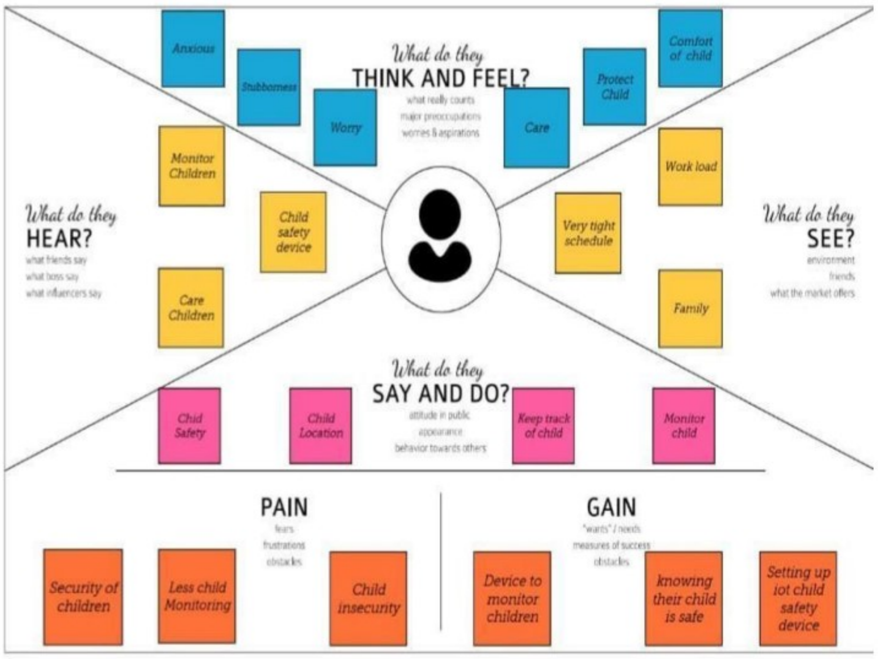
2.3.PROBLEM STATEMENT DEFINITION

Under fast-paced life conditions, everyone is busy in their professional life including parents. They leave the house early in the morning and come back before dinner time. Even the mothers are working. Thus, they do not have sufficient time to take care of their babies. Not all parents could afford a nanny to help them with their children. Then, after working for long hours, the mothers still have to manage the house and take care of their babies simultaneously. Parents might not have the time to soothe their baby to sleep or rock their baby back to sleep in the middle of the night. Studies about the effect of rocking a baby have been carried out and found that babies sleep better while being rocked or swung lightly because the rhythmic movement mimics the gentle rocking they felt while in their mothers' womb. Most available automated cradles are designed to rock non-stop. However, the rocking movement can make the baby nauseous and uncomfortable. Thus, allowing the automated cradle to rock the baby to sleep in the middle of the night is also a problem. Furthermore, some parents place their baby in a separate room. Therefore, parents could not hear the baby crying and could not be there to ease their baby back to sleep in the middle of the night. Other parents may be occupied with house chores. Thus, because they cannot hear their baby crying, they cannot attend to them immediately. Sometimes, the baby only needs a little distraction to return to deep sleep. Several types of baby cradles are available in stores, but they are expensive, and not everyone can afford them. In addition, the existing automatic cradles in the literature have many limitations in terms of functionality, cost, and communication technology support [12]–[15]. To the best of our knowledge, no previous studies have developed a smart cradle with IoT support from scratch, similar to that in the present study. To overcome this problem, a new automatic IoT-based baby monitoring system (IoT-BBMS) is designed, allowing the parents to access an account to monitor the baby's condition anywhere and anytime

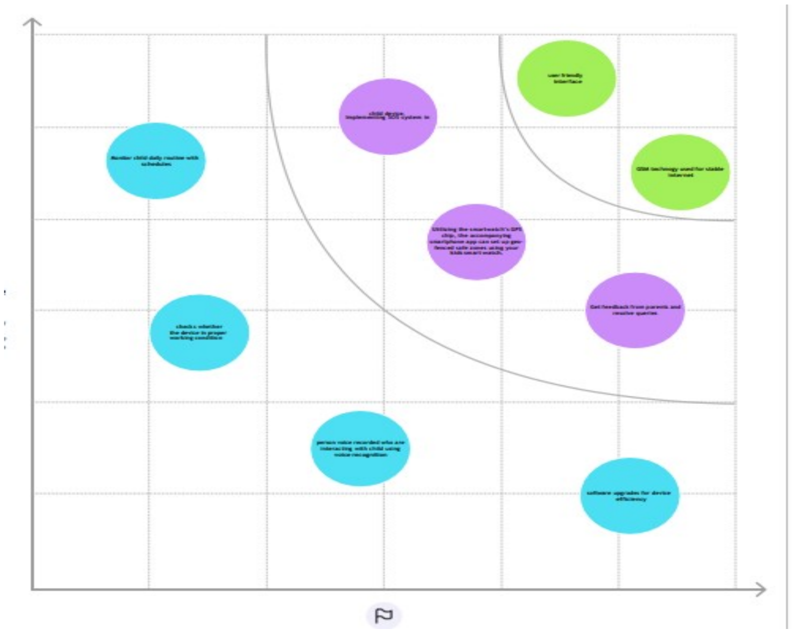
CHAPTER -3

IDEATION & PROPOSED SOLUTION

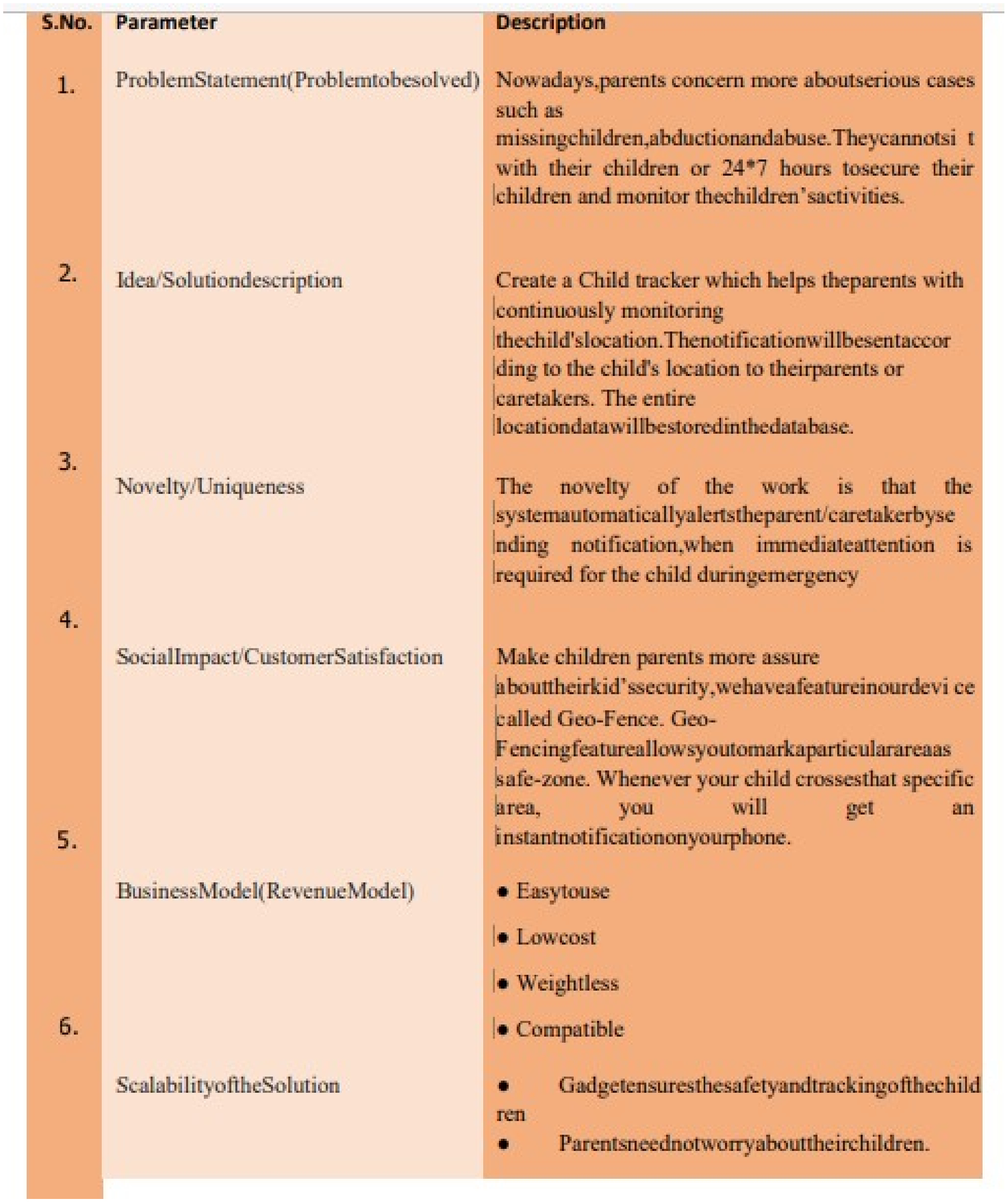
3.1 EMPATHY MAP



3.2 Ideation & Brainstorming

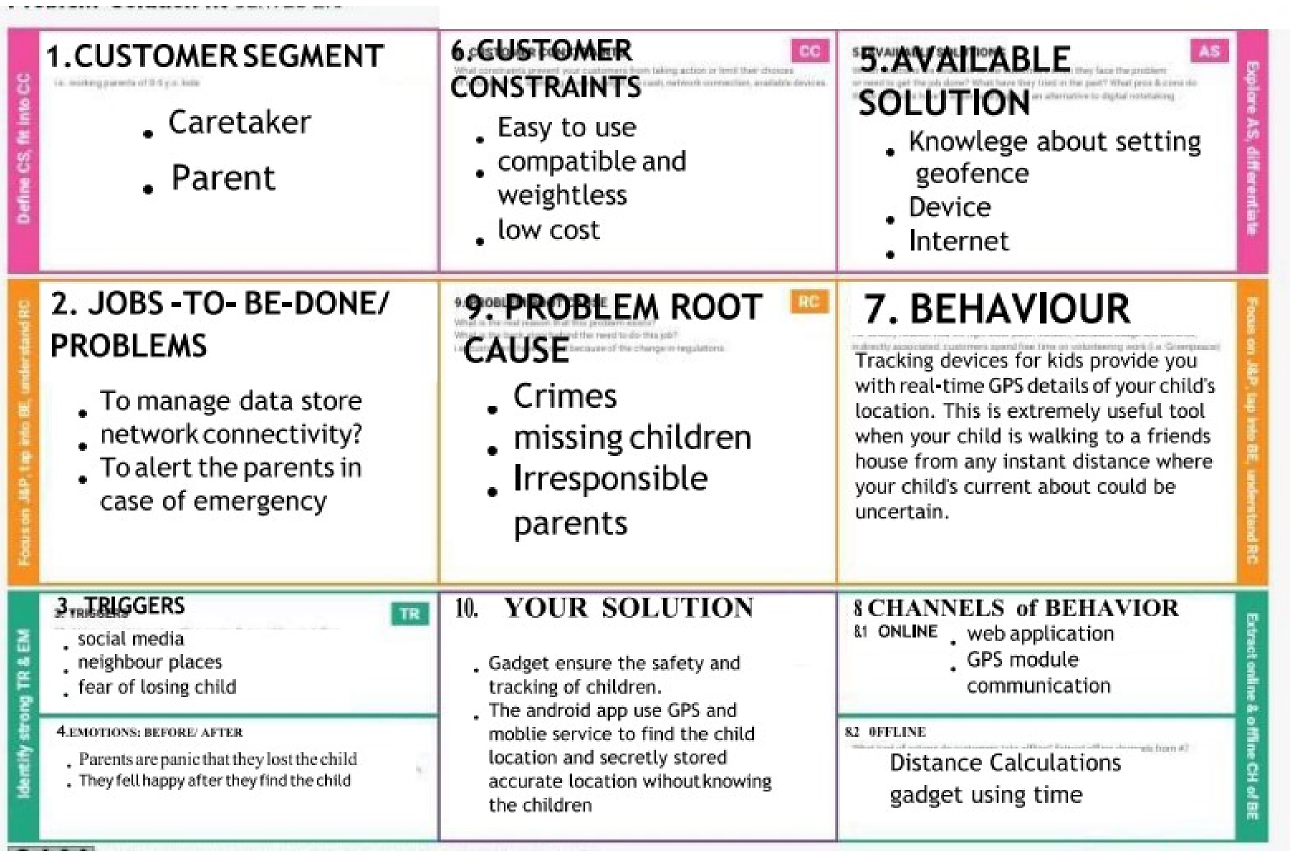

user friendly interface
child device. Implementing SOS system in
Monitor child daily routine with schedules
GSM techonogy used for stable internet
Utilizing the smartwatch's GPS chip, the accompanying smartphone app can set up geo-fenced safe zones using your kids smart watch.
checks whether the device in proper working condition
Get feedback from parents and resolve queries
person voice recorded who are interacting with child using voice recognition
software upgrades for device efficiency

3.3PROPOSED SOLUTI0N



S.No.	Parameter	Description
1.	ProblemStatement(Problemtobesolved)	Nowadays,parents concern more aboutserious cases such as missingchildren,abductionandabuse.Theycannotsi t with their children or 24*7 hours tosecure their children and monitor thechildren'sactivities.
2.	Idea/Solutiondescription	Create a Child tracker which helps theparents with continuously monitoring thechild'slocation.Thenotificationwillbesentaccor ding to the child's location to theirparents or caretakers. The entire locationdatawillbestoredinthedatabase.
3.	Novelty/Uniqueness	The novelty of the work is that the systemautomaticallyalertstheparent/caretakerbyse nding notification,when immediateattention is required for the child duringemergency
4.	SocialImpact/CustomerSatisfaction	Make children parents more assure abouttheirkid'ssecurity,wehaveafeatureinourdevi ce called Geo-Fence. Geo-Fencingfeatureallowsyoutomarkaparticulararareaas safe-zone. Whenever your child crossesthat specific area, you will get an instantnotificationonyourphone.
5.	BusinessModel(RevenueModel)	• Easytouse • Lowcost • Weightless • Compatible
6.	ScalabilityoftheSolution	• Gadgetensuresthesafetyandtrackingofthechild ren • Parentsneednotworryabouttheirchildren.

3.4 PROBLEM SOLUTION FIT



Define CS, fit into CC

1. CUSTOMER SEGMENT

- Caretaker
- Parent

6. CUSTOMER CONSTRAINTS CC

- Easy to use
- compatible and weightless
- low cost

5. AVAILABLE SOLUTION AS

- Knowlege about setting geofence
- Device
- Internet

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS -TO- BE-DONE/ PROBLEMS

- To manage data store
- network connectivity?
- To alert the parents in case of emergency

9. PROBLEM ROOT CAUSE RC

- Crimes
- missing children
- Irresponsible parents

7. BEHAVIOUR

Tracking devices for kids provide you with real-time GPS details of your child's location. This is extremely useful tool when your child is walking to a friends house from any instant distance where your child's current about could be uncertain.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

- social media
- neighbour places
- fear of losing child

4. EMOTIONS: BEFORE/ AFTER

- Parents are panic that they lost the child
- They fell happy after they find the child

10. YOUR SOLUTION

- Gadget ensure the safety and tracking of children.
- The android app use GPS and moblie service to find the child location and secretly stored accurate location wihout knowing the children

8 CHANNELS of BEHAVIOR

8.1 ONLINE

- web application
- GPS module communication

8.2 OFFLINE

Distance Calculations gadget using time

Extract online & offline CH of BE

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Then, we introduced a smart cradle that combines the concept of IoT with baby monitoring system. Subsequently, the selection of material for the smart cradle was carried out. All the hardware and materials used in building this system, which were suitable for a baby, were selected. The priority is to ensure the safety of the baby. The modelling phase is followed by the system design, determining the GUI of applications, and prototype phase. The system design is separated into two phases, namely, the cradle design and control system design. A cradle prototype for the baby monitoring system was designed. In the control system design, the types of electronic components were determined and purchased for implementation in the system. Then, coding was performed according to how the system was proposed. After the modelling phase, the designed baby monitoring system was then enhanced and optimized through several tests to achieve the expected outcome. Subsequently, he system was installed on the cradle prototype for the testing phase before finalizing the smart cradle. When the testing failed due to some coding errors or other problems, the testing phase was repeated until the cradle achieved the expected outcome that satisfied the research objectives

The information regarding the components required in the baby cradle was decided to ensure that they can be installed without errors. We also surveyed available baby cradles that includeda baby monitoring system in the market to gain some insights into the structure of the baby cradle. During this phase, the hardware and software components used in this study were selected. The hardware components included the following: • NodeMCU ESP8266 Wi-Fi Controller Board • 12V DC Power source • Four-channel 5 V relay module • Sound sensor module • Sound and humidity sensor • Mini fan • 12 V DC geared motor • Wireless security camera • Baby cradle The software components include the following: • Nx Siemens software

• Fritzing software • Proteus Stimulation • Arduino IDE software • MQTT Protocol server After the selection process of the components, we designed and fabricated the baby cradle, into which IoT-BBMS was subsequently installed.

4.2 NON FUNCTIONAL REQUIREMENTS

Usability

The system shall allow the users to access the system with pc using web application. The system uses a web application as an interface. The system is user friendly which makes the system easy

Availability

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

Scalability

Scalability is the measure of a system&#39;s ability to increase or decrease in performance and cost in response to changes in application and system processing demands.

Security

A security requirement is a statement of needed security functionality that ensures one of many different security properties of software is being satisfied.

Performance

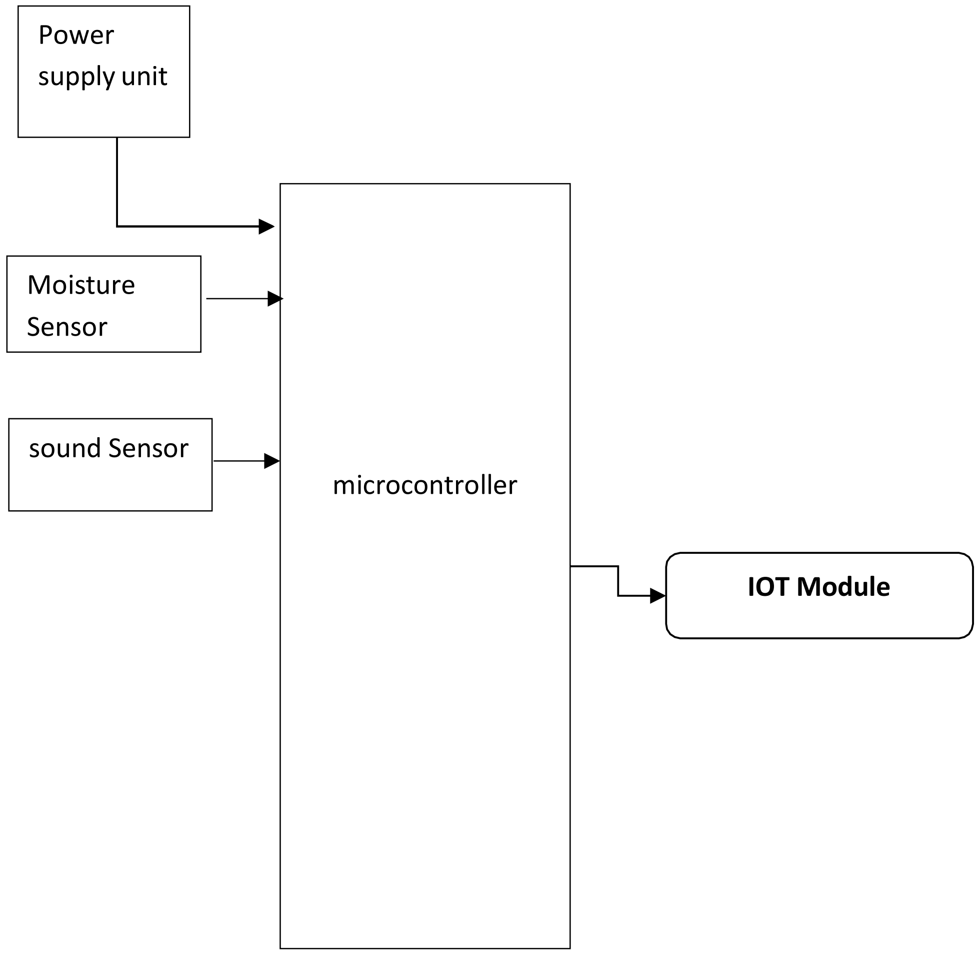
The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen.

Reliability

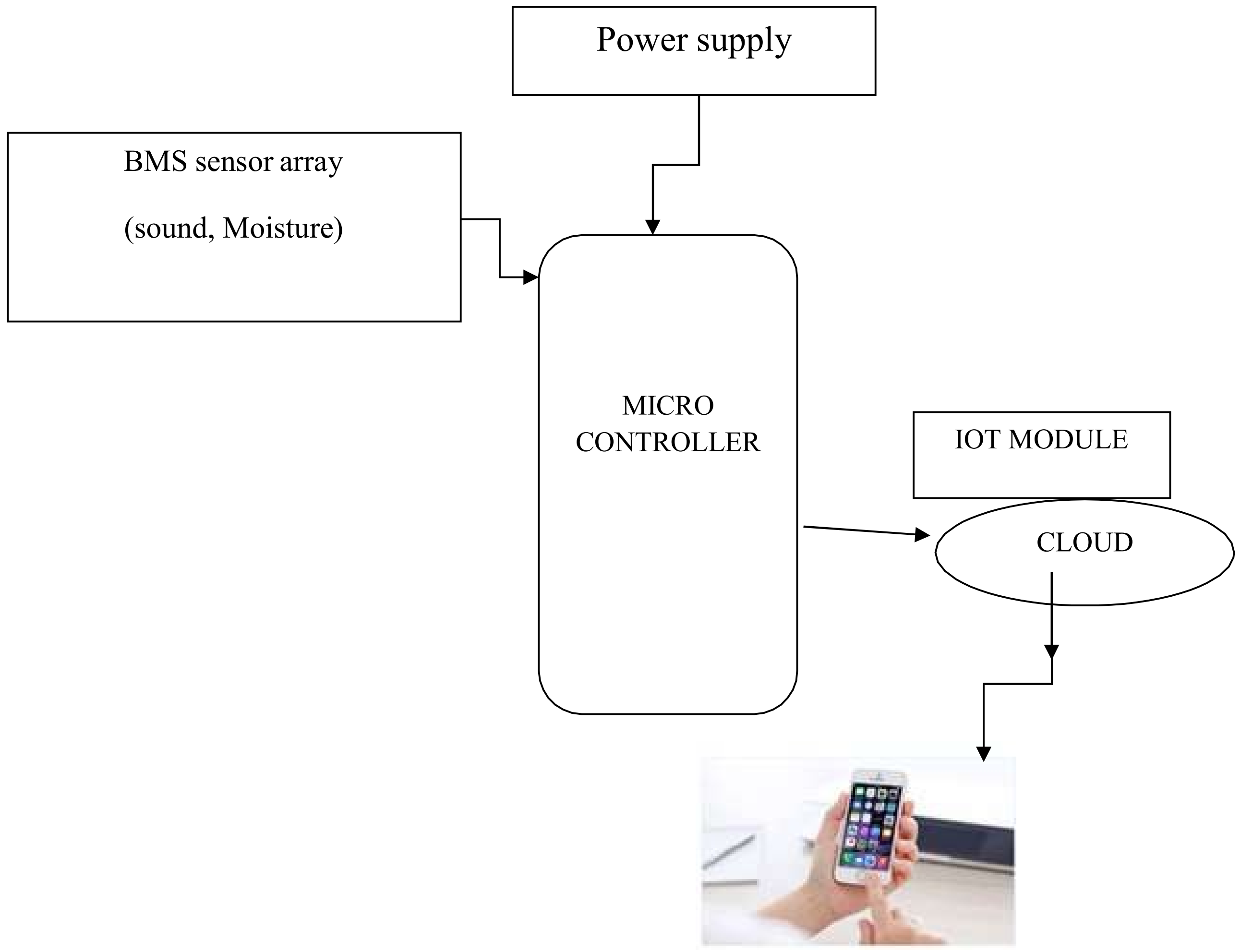
The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week. 24 hours a day.

5 PROJECT DESIGN

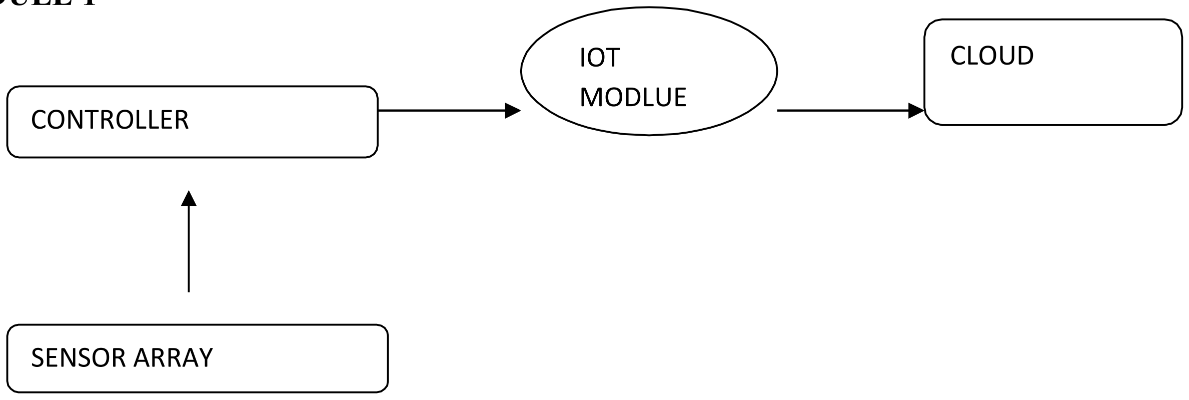
5.1 DATA FLOW DIAGRAMS



5.2 SOLUTION & TECHNICAL ARCHITECTURE



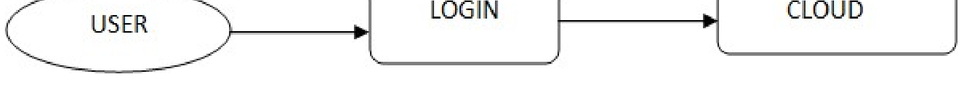
MODULE 1



The Internet of things refers to a type of network to connect anything with the Internet based on stipulated protocols through information sensing equipments to conduct information exchange and communications in order to achieve smart recognitions, positioning, tracing, monitoring, and administration. In this paper we briefly discussed about what IOT is, how IOT enables different technologies, about its architecture, characteristics & applications, IOT functional view & what are the future challenges for IOT.

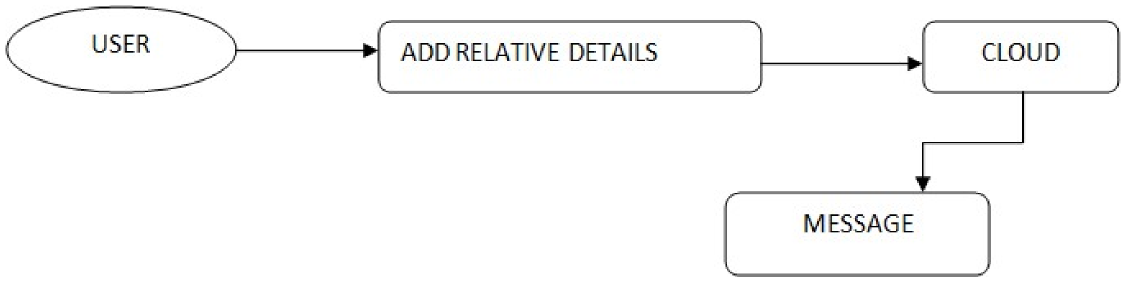
Internet of Things is a new revolution of the Internet. Objects make themselves recognizable and they obtain intelligence by making or enabling context related decisions thanks to the fact that they can communicate information about themselves. They can access information that has been aggregated by other things, or they can be components of complex services. This transformation is concomitant with the emergence of cloud computing capabilities and the transition of the Internet towards IPv6 with an almost unlimited addressing capacity

MODULE 2



Today we are going to build a registration system that keeps track of which users are admin and which are normal users. The normal users in our application are not allowed to access admin pages. All users (Admins as well as normal users) use the same form to login. After logging in, the normal users are redirected to the index page while the admin users are redirected to the admin pages.

MODULE 3



Smart phones are basic needs of our daily life. It’s like a small computer which gives you many facilities such as web browsing, downloading and many more but small data storage space and backup are major problem. On the other hand cloud computing provides efficient computational resources and secure data hosting services. But the data transmission amongtwo secure networks is performed over unsecured network. So need a design to secure data transfer.

CHAPTER-5

HARDWARE AND SOFTWARE REQUIREMENT

5.1.HARDWARE REQUIREMENT

Hardware details:

- Iot Module
- sensors
- Arduino controller
- Power supply
- LCD display
- CPU type : Intel Pentium 4
- Clock speed : 3.0 GHz
- RAM size : 512 MB
- Hard disk capacity : 40 GB
- Monitor type : 15 Inch color monitor
- Keyboard type : internet keyboard

5.2. SOFTWARE REQUIREMENTS

Software details:

- Proteus
- Arduino studio
- Mysql
- Dreamweaver
- Operating System : Windows OS,WAMP server
- Language : PHP, Embedded c

AVR

The **AVR** is a modified Harvard architecture 8-bit RISC single-chip microcontroller, which was developed by Atmel in 1996. The AVR was one of the first microcontroller families to use on-chip flash memory for program storage, as opposed to one-time programmable ROM,EPROM, or EEPROM used by other microcontrollers at the time.

The AVR is a modified Harvard architecture machine, where program and data are stored in separate physical memory systems that appear in different address spaces, but having the ability to read data items from program memory using special instructions.

Basic families

AVRs are generally classified into following:

tinyAVR — the ATtiny series

- 0.5–16 kB program memory
- 6–32-pin package
- Limited peripheral set

megaAVR — the ATmega series

- 4–512 kB program memory
- 28–100-pin package
- Extended instruction set (multiply instructions and instructions for handling larger program memories)
- Extensive peripheral set

XMEGA — the ATxmega series

- 16–384 kB program memory
- 44–64–100-pin package (A4, A3, A1)
- Extended performance features, such as DMA, "Event System", and cryptography support.
- Extensive peripheral set with ADCs

Application-specific AVR

- Mega AVRs with special features not found on the other members of the AVR family, such as LCD controller, USB controller, advanced PWM, CAN, etc.

FPSLIC (AVR with FPGA)

- FPGA 5K to 40K gates
- SRAM for the AVR program code, unlike all other AVRs
- AVR core can run at up to 50 MHZ

32-bit AVRs

In 2006 Atmel released microcontrollers based on the 32-bit AVR32 architecture. They include SIMD and DSP instructions, along with other audio- and video-processing features. This 32-bit family of devices is intended to compete with the ARM-based processors. The instruction set is similar to other RISC cores, but it is not compatible with the original AVR or any of the various ARM cores.

Device architecture

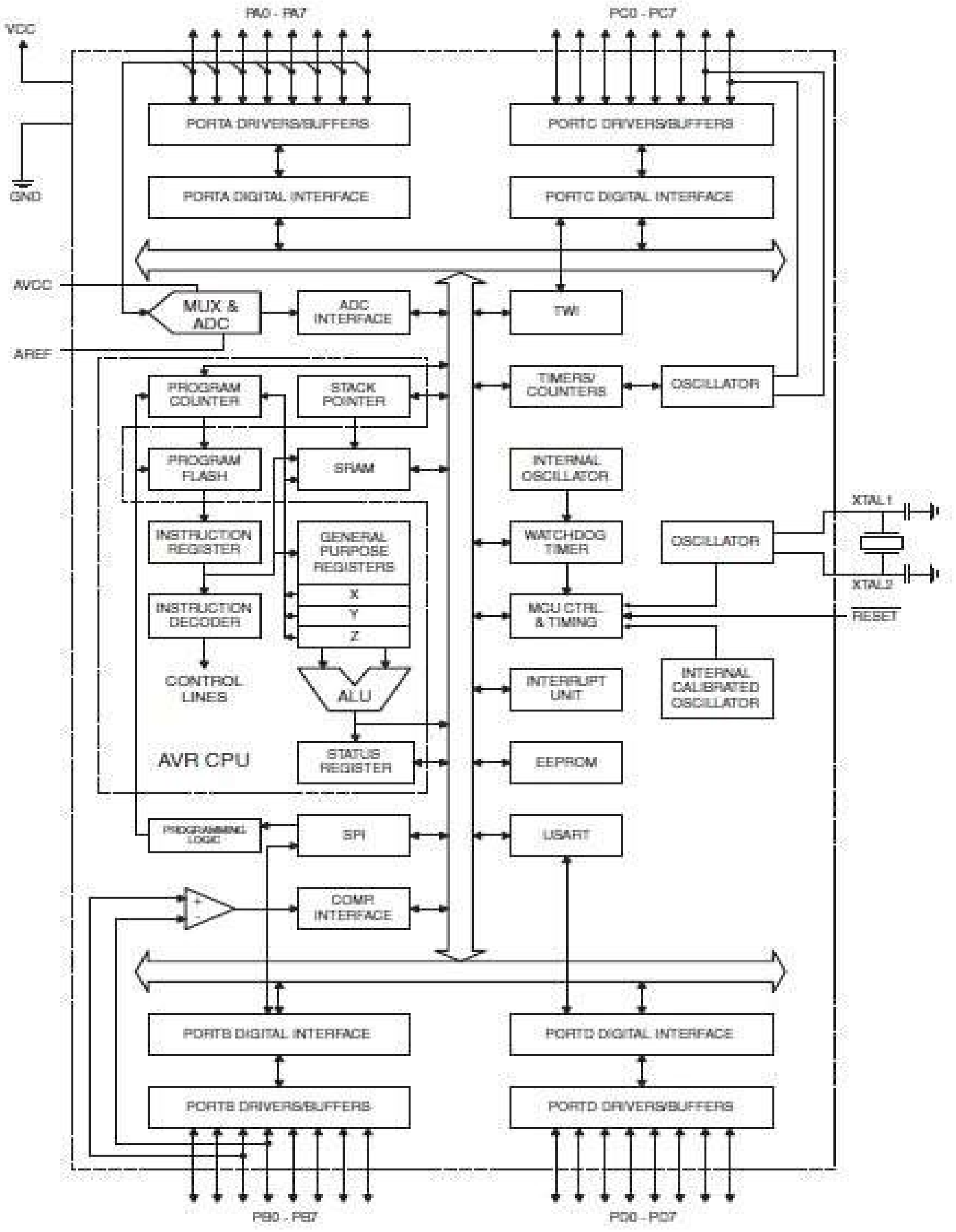
Flash, EEPROM, and SRAM are all integrated onto a single chip, removing the need for external memory in most applications. Some devices have a parallel external bus option to allow adding additional data memory or memory-mapped devices. Almost all devices (except the smallest TinyAVR chips) have serial interfaces, which can be used to connect larger serial EEPROMs or flash chips.

Program memory

Program instructions are stored in non-volatile flash memory. Although the MCUs are 8-bit, each instruction takes one or two 16-bit words.

The size of the program memory is usually indicated in the naming of the device itself (e.g., the ATmega64x line has 64 kB of flash, while the ATmega32x line has 32 kB).

There is no provision for off-chip program memory; all code executed by the AVR core must reside in the on-chip flash. However, this limitation does not apply to the AT94 FPSLIC AVR/FPGA chips.



Internal data memory

The data address space consists of the register file, I/O registers, and SRAM.

Internal registers

The AVRs have 32 single-byte registers and are classified as 8-bit RISC devices.

In the tiny AVR and mega AVR variants of the AVR architecture, the working registers are mapped in as the first 32 memory addresses (0000_{16}–$001F_{16}$), followed by 64 I/O registers (0020_{16}–$005F_{16}$). In devices with many peripherals, these registers are followed by 160 “extended I/O” registers, only accessible as memory-mapped I/O (0060_{16}–$00FF_{16}$).

Actual SRAM starts after these register sections, at address 0060_{16} or, in devices with “extended I/O”, at 0100_{16}.

Even though there are separate addressing schemes and optimized opcodes for accessing the register file and the first 64 I/O registers, all can still be addressed and manipulated as if they were in SRAM.

The very smallest of the tiny AVR variants use a reduced architecture with only 16 registers (r0 through r15 are omitted) which are not addressable as memory locations. I/O memory begins at address 0000_{16}, followed by SRAM. In addition, these devices have slight deviations from the standard AVR instruction set. Most notably, the direct load/store instructions (LDS/STS) have been reduced from 2 words (32 bits) to 1 word (16 bits), limiting the total direct addressable memory (the sum of both I/O and SRAM) to 128 bytes. Conversely, the indirect load instruction's (LD) 16-bit address space is expanded to also include non-volatile memory such as Flash and configuration bits; therefore, the LPM instruction is unnecessary and omitted.

In the XMEGA variant, the working register file is not mapped into the data address space; as such, it is not possible to treat any of the XMEGA's working registers as though they were SRAM. Instead, the I/O registers are mapped into the data address space starting at the very beginning of the address space. Additionally, the amount of data address space dedicated to I/O registers has grown substantially to 4096 bytes (0000_{16}–$0FFF_{16}$). As with previous generations, however, the fast I/O manipulation instructions can only reach the first 64 I/O register locations (the first 32 locations for bitwise instructions). Following the I/O registers, the XMEGA series sets aside a 4096 byte range of the data address space, which can be used optionally for mapping the internal EEPROM to the data address space (1000_{16}–$1FFF_{16}$). The actual SRAM is located after these ranges, starting at 2000_{16}.

GPIO ports

Each GPIO port on a tiny or mega AVR drives up to eight pins and is controlled by three 8-bit registers: DDRx, PORTx and PINx, where x is the port identifier.

- DDRx: Data Direction Register, configures the pins as either inputs or outputs.
- PORTx: Output port register. Sets the output value on pins configured as outputs. Enables or disables the pull-up resistor on pins configured as inputs.
- PINx: Input register, used to read an input signal. On some devices (but not all, check the datasheet), this register can be used for pin toggling: writing a logic one to a PIN x bit toggles the corresponding bit in PORT x, irrespective of the setting of the DDR x biT

Xmega AVR have additional registers for push/pull, totem-pole and pullup configurations.

EEPROM

Almost all AVR microcontrollers have internal EEPROM for semi-permanent data storage. Like flash memory, EEPROM can maintain its contents when electrical power is removed.

In most variants of the AVR architecture, this internal EEPROM memory is not mapped into the MCU's addressable memory space. It can only be accessed the same way an external peripheral device is, using special pointer registers and read/write instructions, which makes EEPROM access much slower than other internal RAM.

However, some devices in the Secure AVR (AT90SC) family use a special EEPROM mapping to the data or program memory, depending on the configuration. The XMEGA family also allows the EEPROM to be mapped into the data address space.

Since the number of writes to EEPROM is not unlimited — Atmel specifies 100,000 write cycles in their datasheets — a well designed EEPROM write routine should compare the contents of an EEPROM address with desired contents and only perform an actual write if the contents need to be changed.

Note that erase and write can be performed separately in many cases, byte-by-byte, which may also help prolong life when bits only need to be set to all 1s (erase) or selectively cleared to 0s (write).

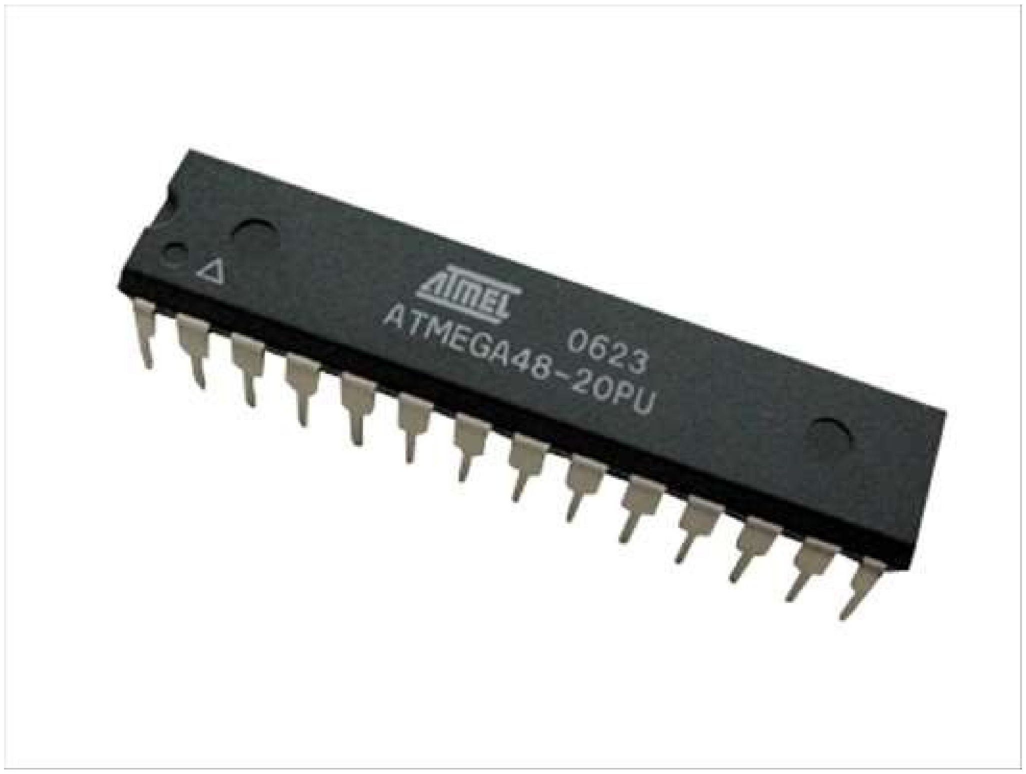


FIG NO: ATMEGA8A

Program execution

Atmel's AVRs have a two-stage, single-level pipeline design. This means the next machine instruction is fetched as the current one is executing. Most instructions take just one or two clock cycles, making AVRs relatively fast among eight-bit microcontrollers.

The AVR processors were designed with the efficient execution of compiled C code in mind and have several built-in pointers for the task.

Instruction set

The AVR instruction set is more orthogonal than those of most eight-bit microcontrollers, in particular the 8051 clones and PIC microcontrollers with which AVR competes today. However, it is not completely regular:

- Pointer registers X, Y, and Z have addressing capabilities that are different from each other.
- Register locations R0 to R15 have different addressing capabilities than register locations R16 to R31.
- I/O ports 0 to 31 have different addressing capabilities than I/O ports 32 to 63.
- CLR affects flags, while SER does not, even though they are complementary instructions. CLR set all bits to zero, and SER sets them to one. (Note that CLR is pseudo-op for EOR R, R; and SER is short for LDI R,$FF. Math operations such as EOR modify flags, while moves/loads/stores/branches such as LDI do not.)
- Accessing read-only data stored in the program memory (flash) requires special LPM instructions; the flash bus is otherwise reserved for instruction memory.

Additionally, some chip-specific differences affect code generation. Code pointers (including return addresses on the stack) are two bytes long on chips with up to 128 kBytes of flash memory, but three bytes long on larger chips; not all chips have hardware multipliers; chips with over 8 kB of flash have branch and call instructions with longer ranges; and so forth.

The mostly regular instruction set makes programming it using C (or even Ada) compilers fairly straightforward. GCC has included AVR support for quite some time, and that support is widely used. In fact, Atmel solicited input from major developers of compilers for small microcontrollers, to determine the instruction set features that were most useful in a compiler for high-level languages.

MCU speed

The AVR line can normally support clock speeds from 0 to 20 MHz, with some devices reaching 32 MHz. Lower-powered operation usually requires a reduced clock speed. All recent (Tiny, Mega, and Xmega, but not 90S) AVRs feature an on-chip oscillator, removing the need for external clocks or resonator circuitry. Some AVRs also have a system clock prescaler that can divide down the system clock by up to 1024. This prescaler can be reconfigured by software during run-time, allowing the clock speed to be optimized.

Since all operations (excluding multiplication and 16-bit add/subtract) on registers R0–R31 are single-cycle, the AVR can achieve up to 1 MIPS per MHz, i.e. an 8 MHz processor can achieve up to 8 MIPS. Loads and stores to/from memory take two cycles, branching takes two cycles. Branches in the latest "3-byte PC" parts such as ATmega2560 are one cycle slower than on previous devices.

Development

AVRs have a large following due to the free and inexpensive development tools available, including reasonably priced development boards and free development software. The AVRs are sold under various names that share the same basic core, but with different peripheral and memory combinations. Compatibility between chips in each family is fairly good, although I/O controller features may vary.

Features

AVRs offer a wide range of features:

- Multifunction, bi-directional general-purpose I/O ports with configurable, built-in pull-up resistors
- Multiple internal oscillators, including RC oscillator without external parts
- Internal, self-programmable instruction flash memory up to 256 kB (384 kB on XMega)
- In-system programmable using serial/parallel low-voltage proprietary interfaces or JTAG
- Optional boot code section with independent lock bits for protection
- On-chip debugging (OCD) support through JTAG or debugWIRE on most devices
- The JTAG signals (TMS, TDI, TDO, and TCK) are multiplexed on GPIOs. These pins can be configured to function as JTAG or GPIO depending on the setting of a fuse bit, which can be programmed via ISP or HVSP. By default, AVRs with JTAG come with the JTAG interface enabled.
- Debug WIRE uses the /RESET pin as a bi-directional communication channel to access on-chip debug circuitry. It is present on devices with lower pin counts, as it only requires one pin.
- Internal data EEPROM up to 4 kB
- Internal SRAM up to 16 kB (32 kB on XMega)
- External 64 kB little endian data space on certain models, including the Mega8515 and Mega162.

    - The external data space is overlaid with the internal data space, such that the full 64 kB address space does not appear on the external bus and accesses to e.g. address 0100_{16} will access internal RAM, not the external bus.
    - In certain members of the XMega series, the external data space has been enhanced to support both SRAM and SDRAM. As well, the data addressing modes have been expanded to allow up to 16 MB of data memory to be directly addressed.
    - AVRs generally do not support executing code from external memory. Some ASSPs using the AVR core do support external program memory.
- 8-bit and 16-bit timers
    - PWM output (some devices have an enhanced PWM peripheral which includes a dead-time generator)
    - Input capture that record a time stamp triggered by a signal edge
- Analog comparator
- 10 or 12-bit A/D converters, with multiplex of up to 16 channels
- 12-bit D/A converters
- A variety of serial interfaces, including
    - I^2C compatible Two-Wire Interface (TWI)
    - Synchronous/asynchronous serial peripherals (UART/USART) (used with RS-232, RS-485, and more)
    - Serial Peripheral Interface Bus (SPI)
    - Universal Serial Interface (USI): a multi-purpose hardware communication module that can be used to implement an SPI I^2C or UART interface.
- Brownout detection
- Watchdog timer (WDT)
- Multiple power-saving sleep modes
- Lighting and motor control (PWM-specific) controller models
- CAN controller support
- USB controller support
    - Proper full-speed (12 Mbit/s) hardware & Hub controller with embedded AVR.
    - Also freely available low-speed (1.5 Mbit/s) (HID) bit banging software emulations

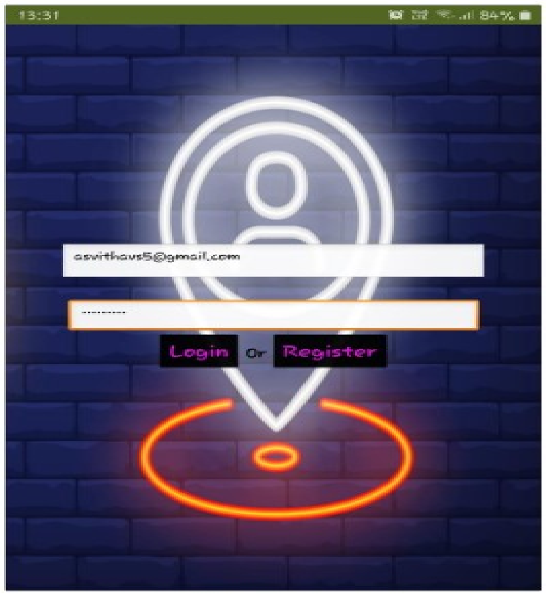
- Ethernet controller support
- LCD controller support
- Low-voltage devices operating down to 1.8 V (to 0.7 V for parts with built-in DC–DC up converter)
- picoPower devices
- DMA controllers and "event system" peripheral communication.
- Fast cryptography support for AES and DES

CHAPTER 7

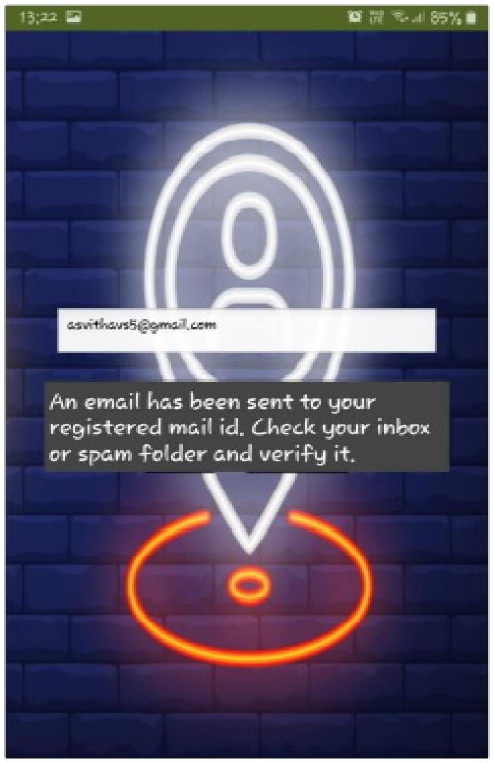
Project Planning & Scheduling

7.1 Sprint Planning & Estimation

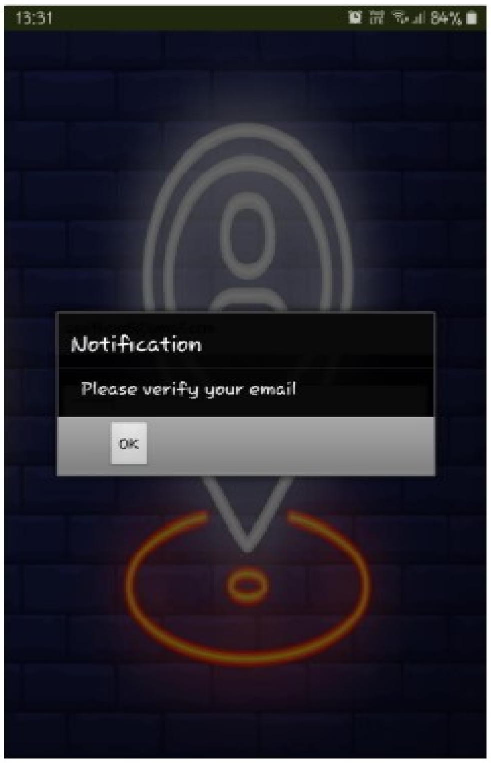
Registration: USN–1:Asauser,Icanregisterfortheapplicationbyenteringmyemailand password



USN 2 As an user I will recive notification



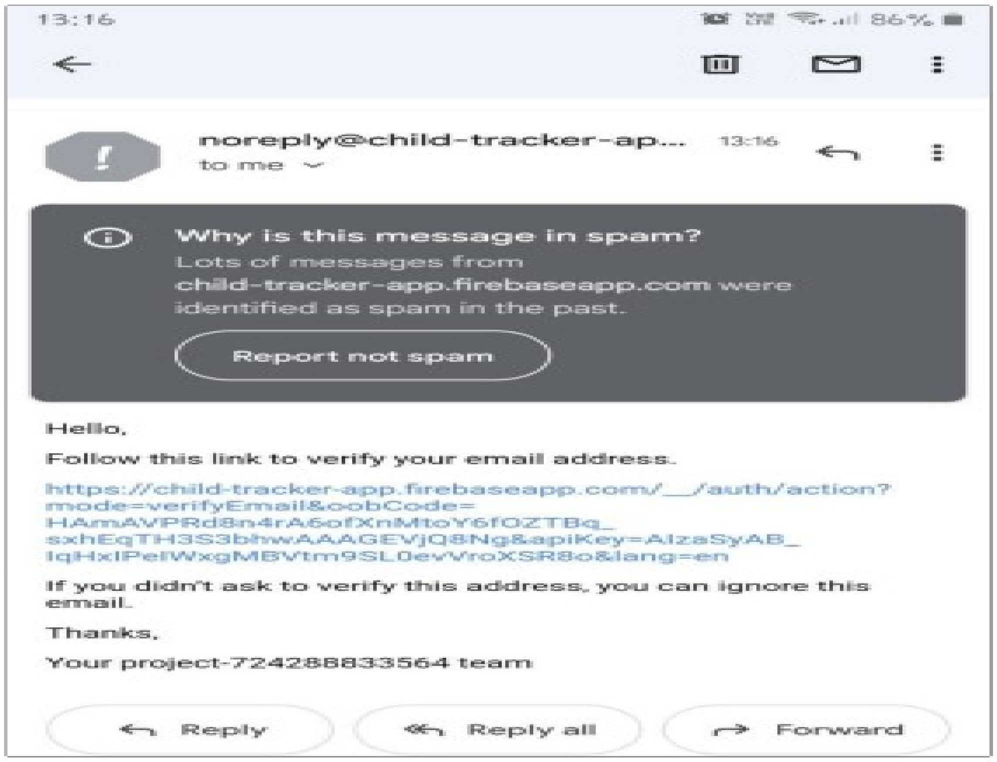
An email has been sent to your registered mail id. Check your inbox or spam folder and verify it.



Notification

Please verify your email

Verification email is sent to the registered email address.


13:16
86%
noreply@child-tracker-ap...
13:16
to me
Why is this message in spam?
Lots of messages from child-tracker-app.firebaseapp.com were identified as spam in the past.
Report not spam
Reply
Reply all
Forward

Hello,

Follow this link to verify your email address.

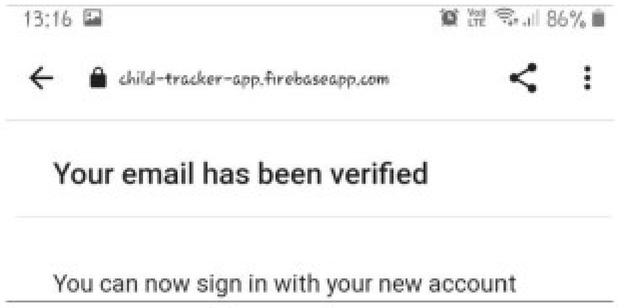
https://child-tracker-app.firebaseapp.com/__/auth/action?mode=verifyEmail&oobCode=HAmAVPRd8n4rA6ofXnMtoY6fOZTBq_sxhEqTH3S3bhwAAAGEVjQ8Ng&apiKey=AIzaSyAB_IqHxIPeIWxgMBVtm9SL0evVroXSR8o&lang=en

If you didn't ask to verify this address, you can ignore this email.

Thanks,

Your project-724288833564 team

After verification is done

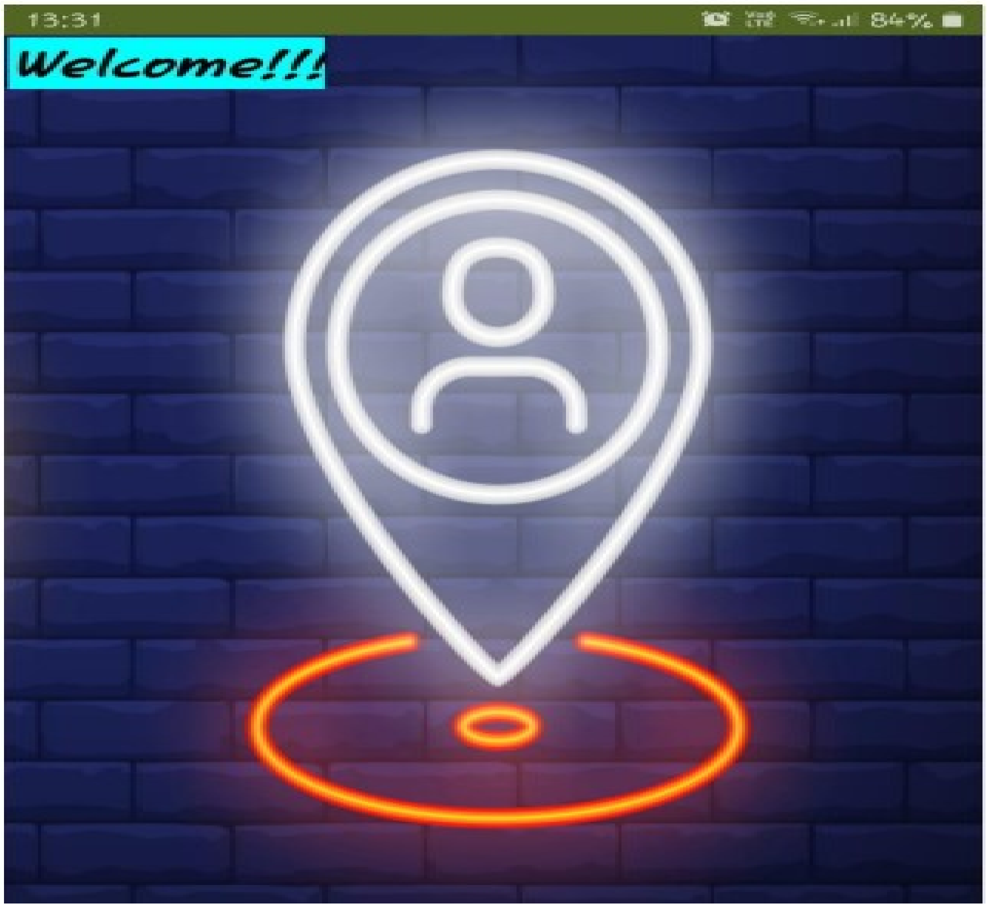


Your email has been verified

You can now sign in with your new account

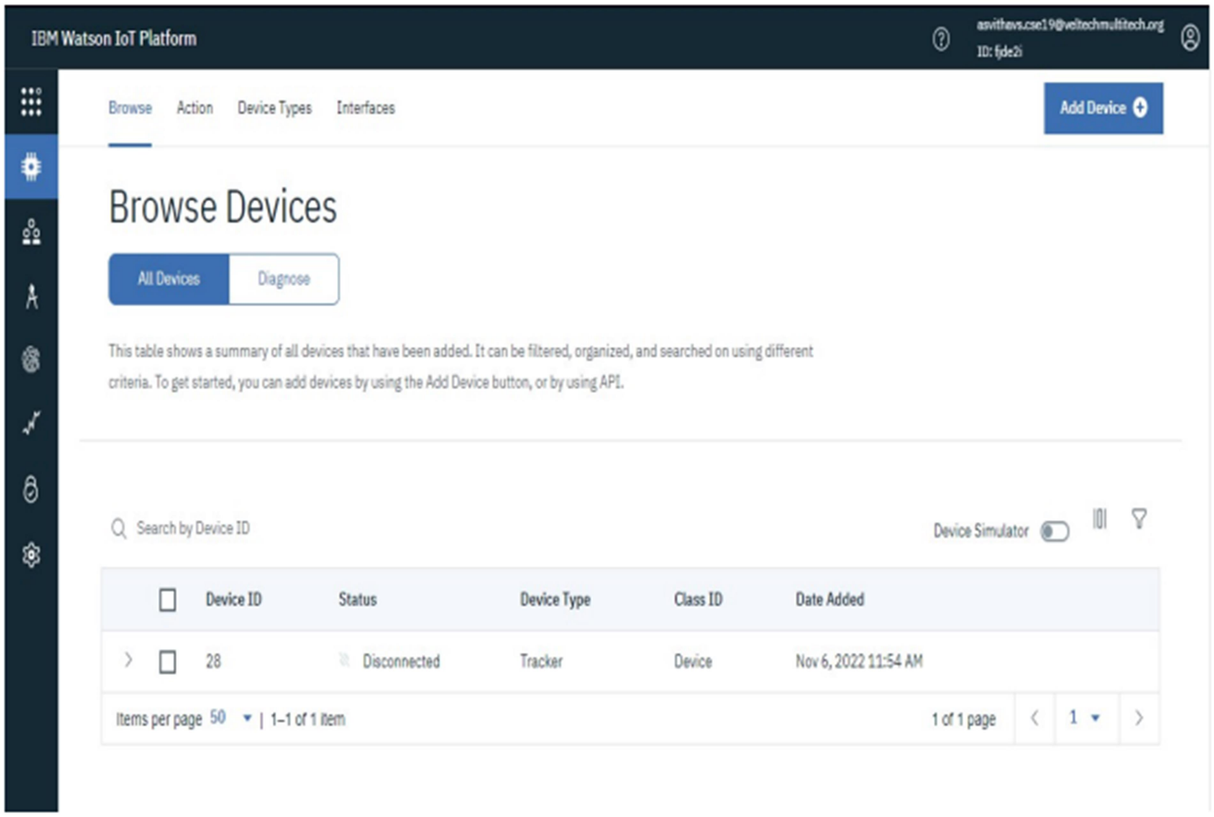
USN-3:

As a user,I can login to the application by entering email & password



USN-4: Integrating the IBM Watson IoT Platform and Cloudant DB with then odered.

● Launching IBM IoT Watson



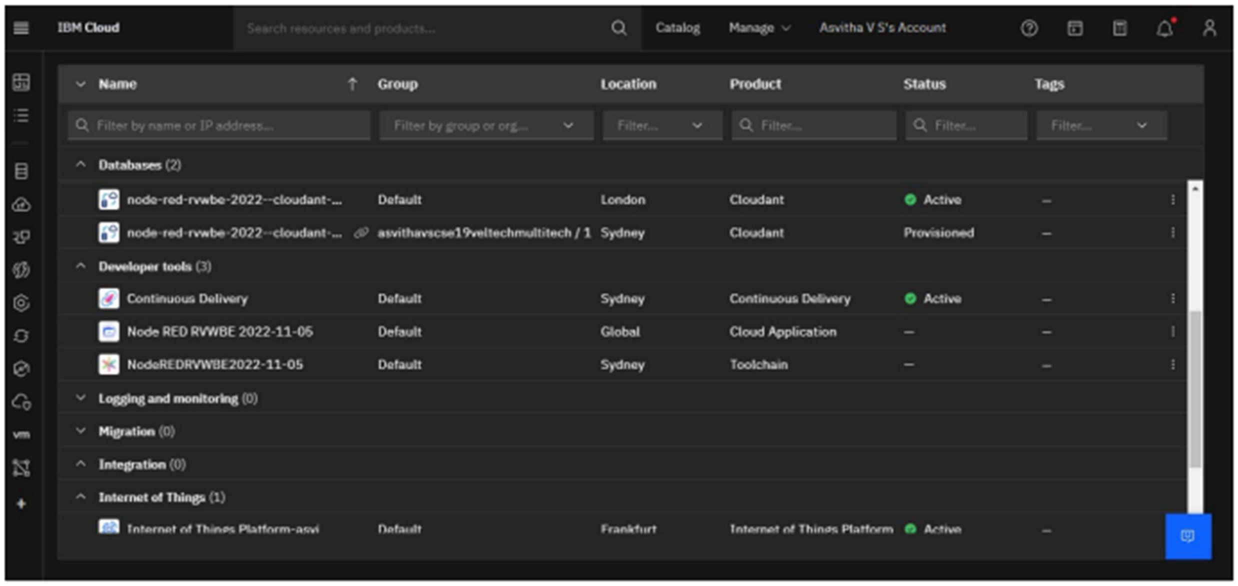
Browse Devices

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Device ID	Status	Device Type	Class ID	Date Added
28	Disconnected	Tracker	Device	Nov 6, 2022 11:54 AM

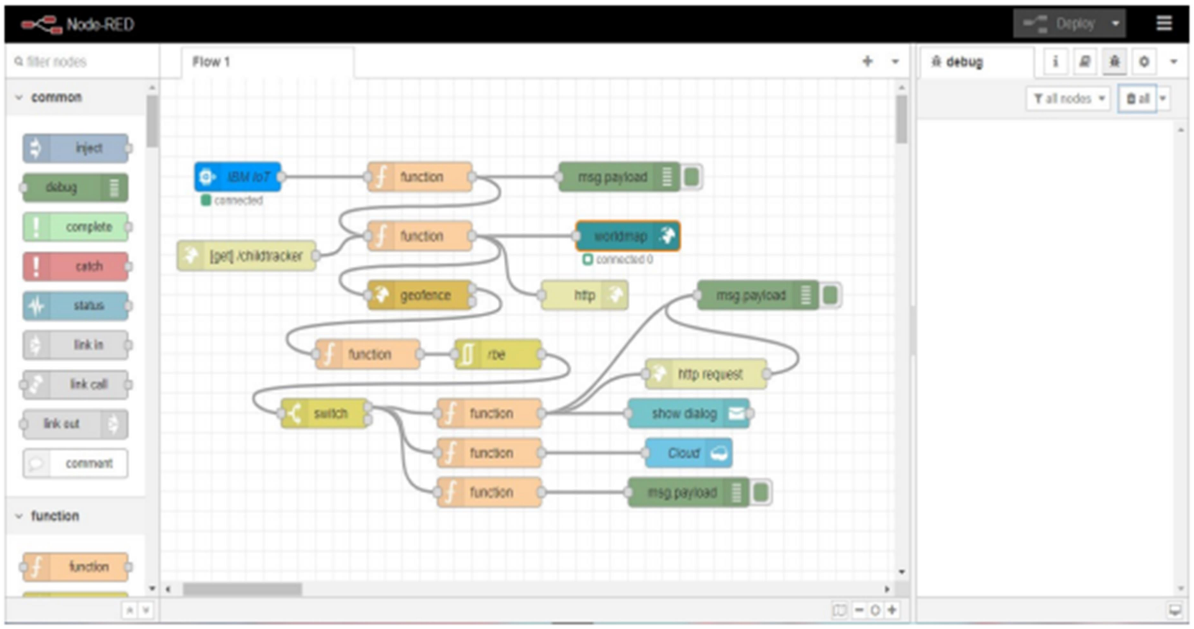
Items per page 50 | 1–1 of 1 item   1 of 1 page   1

Implementing the node-red in IBM cloud

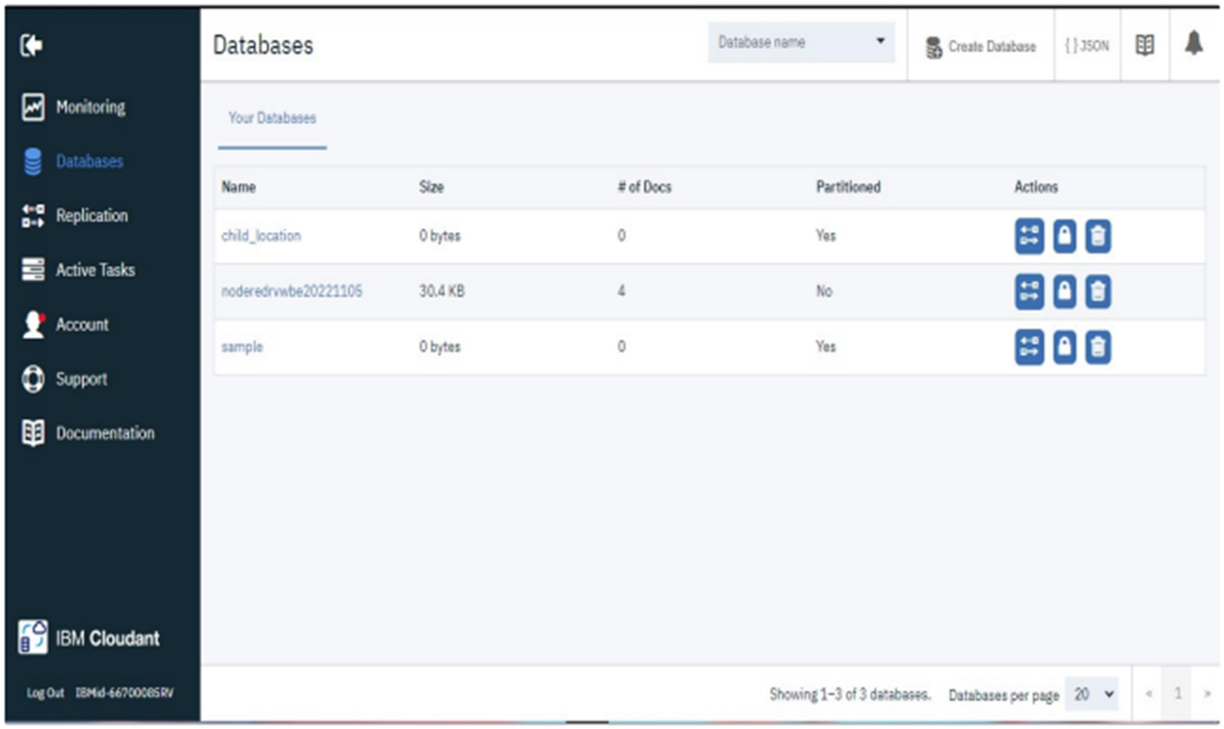


Name	Group	Location	Product	Status	Tags
Databases (2)					
node-red-rvwbe-2022--cloudant-...	Default	London	Cloudant	● Active	–
node-red-rvwbe-2022--cloudant-...	asvithavscse19veltechmultitech / 1	Sydney	Cloudant	Provisioned	–
Developer tools (3)					
Continuous Delivery	Default	Sydney	Continuous Delivery	● Active	–
Node RED RVWBE 2022-11-05	Default	Global	Cloud Application	–	–
NodeREDRVWBE2022-11-05	Default	Sydney	Toolchain	–	–
Logging and monitoring (0)					
Migration (0)					
Integration (0)					
Internet of Things (1)					
Internet of Things Platform-asvi	Default	Frankfurt	Internet of Things Platform	● Active	–

Designing the node-red workflow for our project



Launch the cloudant DB and create a database to store the location data

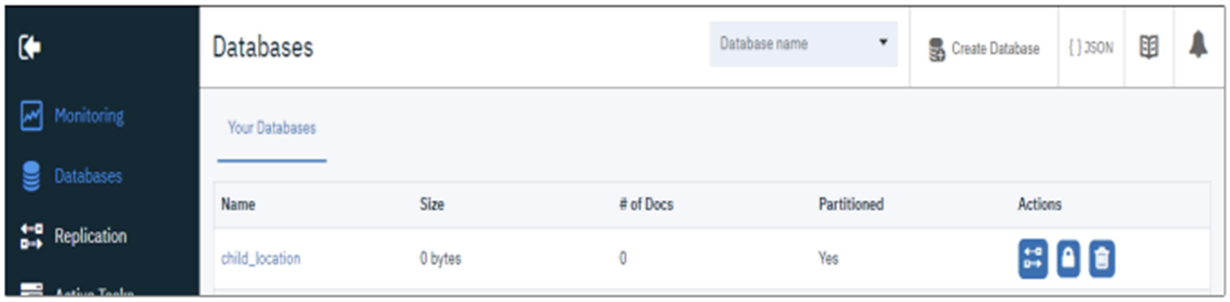


Databases

Name	Size	# of Docs	Partitioned	Actions
child_location	0 bytes	0	Yes	
noderedrvwbe20221105	30.4 KB	4	No	
sample	0 bytes	0	Yes	

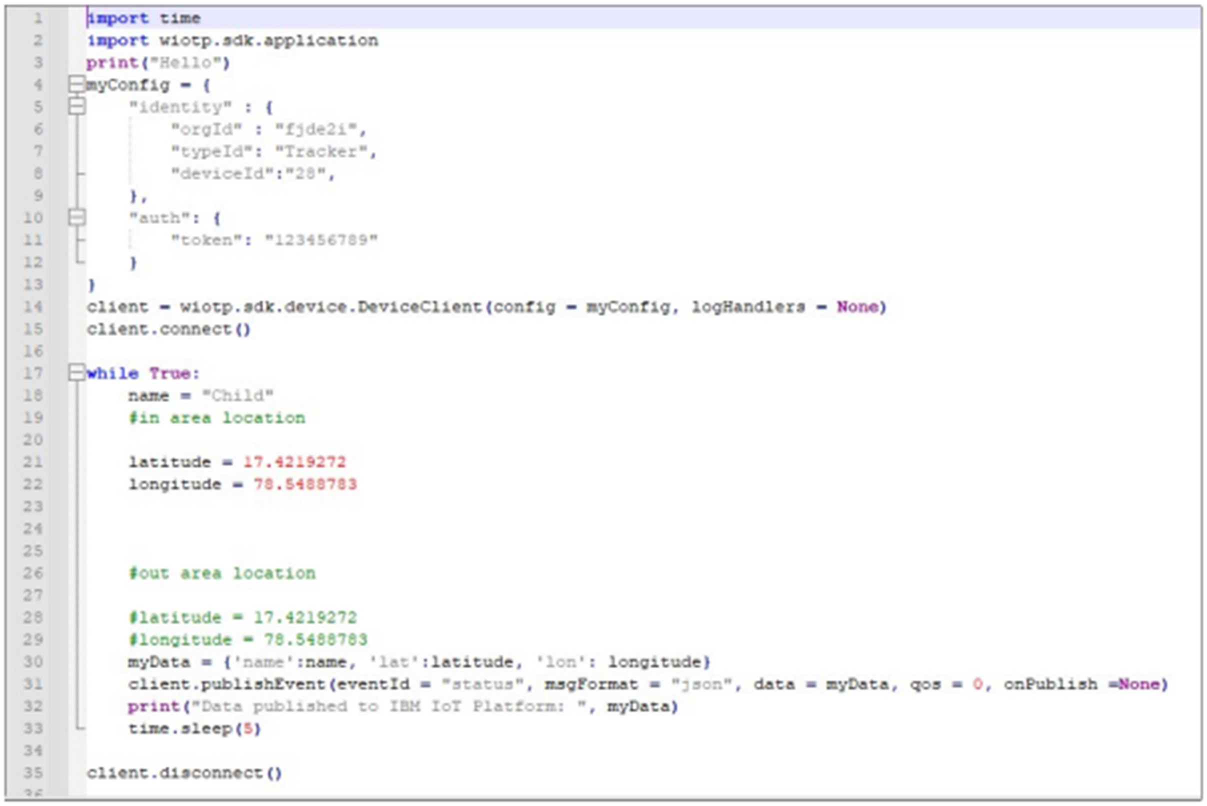
Showing 1-3 of 3 databases. Databases per page 20

For our project we are creating a database called child_loaction.



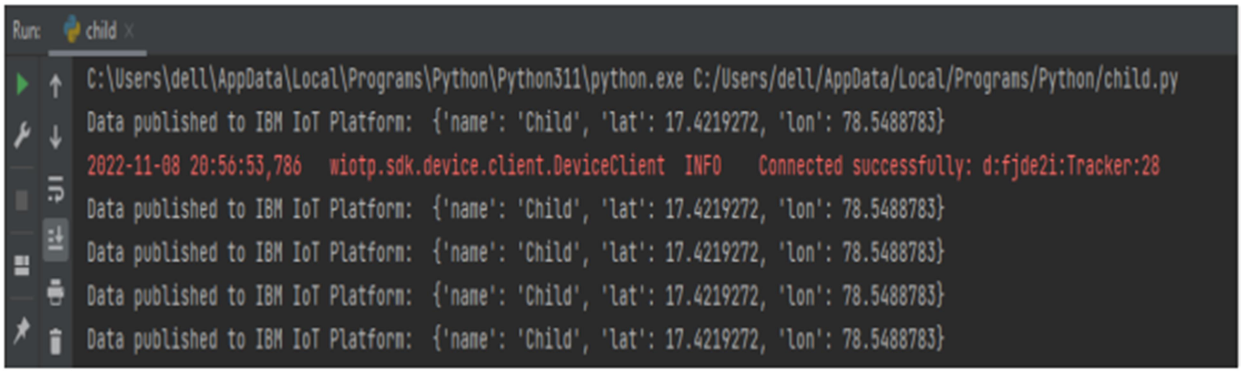
Name	Size	# of Docs	Partitioned	Actions
child_location	0 bytes	0	Yes	

USN–5:Developing the Python code for connecting with IBM Watson IoT platform.



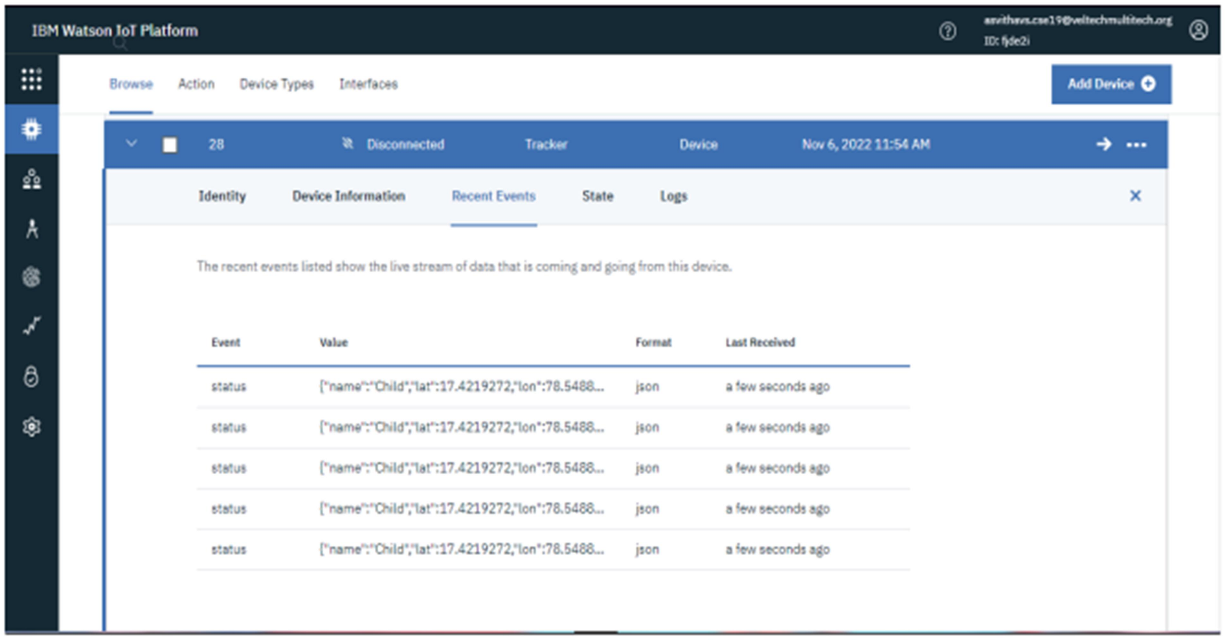
```
import time
import wiotp.sdk.application
print("Hello")
myConfig = {
    "identity" : {
        "orgId" : "fjde2i",
        "typeId": "Tracker",
        "deviceId":"28",
    },
    "auth": {
        "token": "123456789"
    }
}
client = wiotp.sdk.device.DeviceClient(config = myConfig, logHandlers = None)
client.connect()

while True:
    name = "Child"
    #in area location

    latitude = 17.4219272
    longitude = 78.5488783



    #out area location

    #latitude = 17.4219272
    #longitude = 78.5488783
    myData = {'name':name, 'lat':latitude, 'lon': longitude}
    client.publishEvent(eventId = "status", msgFormat = "json", data = myData, qos = 0, onPublish =None)
    print("Data published to IBM IoT Platform: ", myData)
    time.sleep(5)

client.disconnect()
```

Connected successfully with IBM IoT Watson



```
C:\Users\dell\AppData\Local\Programs\Python\Python311\python.exe C:/Users/dell/AppData/Local/Programs/Python/child.py
Data published to IBM IoT Platform:  {'name': 'Child', 'lat': 17.4219272, 'lon': 78.5488783}
2022-11-08 20:56:53,786   wiotp.sdk.device.client.DeviceClient  INFO   Connected successfully: d:fjde2i:Tracker:28
Data published to IBM IoT Platform:  {'name': 'Child', 'lat': 17.4219272, 'lon': 78.5488783}
Data published to IBM IoT Platform:  {'name': 'Child', 'lat': 17.4219272, 'lon': 78.5488783}
Data published to IBM IoT Platform:  {'name': 'Child', 'lat': 17.4219272, 'lon': 78.5488783}
Data published to IBM IoT Platform:  {'name': 'Child', 'lat': 17.4219272, 'lon': 78.5488783}
```

IoT platform reciving childs location



IBM Watson IoT Platform

anvithavs.cse19@veltechmultitech.org
ID: fjde2i

Browse Action Device Types Interfaces Add Device

28 Disconnected Tracker Device Nov 6, 2022 11:54 AM

Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

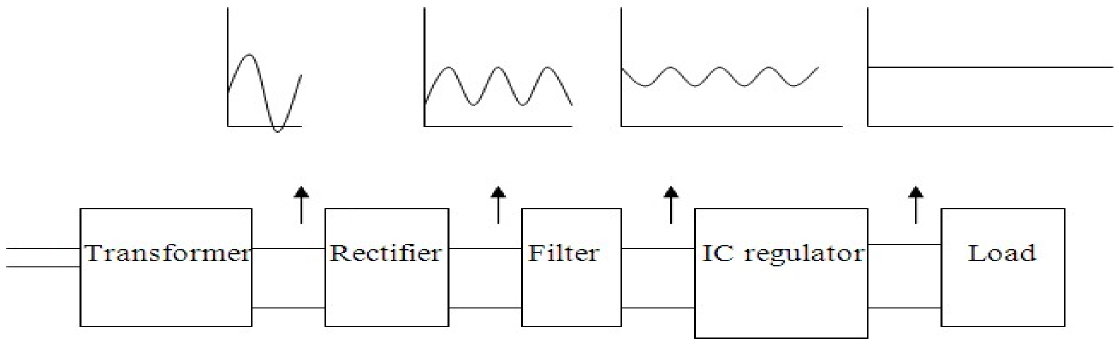
Event	Value	Format	Last Received
status	{"name":"Child","lat":17.4219272,"lon":78.5488...	json	a few seconds ago
status	{"name":"Child","lat":17.4219272,"lon":78.5488...	json	a few seconds ago
status	{"name":"Child","lat":17.4219272,"lon":78.5488...	json	a few seconds ago
status	{"name":"Child","lat":17.4219272,"lon":78.5488...	json	a few seconds ago
status	{"name":"Child","lat":17.4219272,"lon":78.5488...	json	a few seconds ago

CHAPTER-8

POWER SUPPLIES

The present chapter introduces the operation of power supply circuits built using filters, rectifiers, and then voltage regulators. Starting with an ac voltage, a steady dc voltage is obtained by rectifying the ac voltage, then filtering to a dc level, and finally, regulating to obtain a desired fixed dc voltage. The regulation is usually obtained from an IC voltage regulator unit, which takes a dc voltage and provides a somewhat lower dc voltage, which remains the same even if the input dc voltage varies, or the output load connected to the dc voltage changes.

A block diagram containing the parts of a typical power supply and the voltage at various points in the unit is shown in fig 19.1. The ac voltage, typically 120 V rms, is connected to a transformer, which steps that ac voltage down to the level for the desired dc output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation. A regulator circuit can use this dc input to provide a dc voltage that not only has much less ripple voltage but also remains the same dc value even if the input dc voltage varies somewhat, or the load connected to the output dc voltage changes. This voltage regulation is usually obtained using one of a number of popular voltage regulator IC units.



IC VOLTAGE REGULATORS

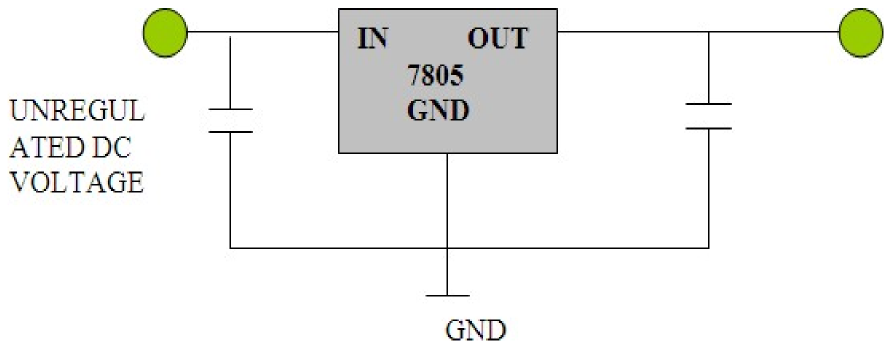
Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC. Although the internal construction of the IC is somewhat different from that described for discrete voltage regulator circuits, the external operation is much the same. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustably set voltage.

A power supply can be built using a transformer connected to the ac supply line to step the ac voltage to desired amplitude, then rectifying that ac voltage, filtering with a capacitor and RC filter, if desired, and finally regulating the dc voltage using an IC regulator. The regulators can be selected for operation with load currents from hundreds of mille amperes to tens of amperes, corresponding to power ratings from mill watts to tens of watts.

THREE-TERMINAL VOLTAGE REGULATORS

Fig shows the basic connection of a three-terminal voltage regulator IC to a load. The fixed voltage regulator has an unregulated dc input voltage, Vi, applied to one input terminal, a regulated output dc voltage, Vo, from a second terminal, with the third terminal connected to ground. For a selected regulator, IC device specifications list a voltage range over which the input voltage can vary to maintain a regulated output voltage over a range of load current. The specifications also list the amount of output voltage change resulting from a change in load current (load regulation) or in input voltage (line regulation).

Fixed Positive Voltage Regulators:



The series 78 regulators provide fixed regulated voltages from 5 to 24 V. Figure 19.26 shows how one such IC, a 7812, is connected to provide voltage regulation with output from this unit of +12V dc. An unregulated input voltage Vi is filtered by capacitor C1 and connected to the IC's IN terminal. The IC's OUT terminal provides a regulated + 12V which is filtered by capacitor C2 (mostly for any high-frequency noise). The third IC terminal is connected to ground (GND). While the input voltage may vary over some permissible voltage range, and the output load may vary over some acceptable range, the output voltage remains constant within specified voltage variation limits. These limitations are spelled out in the manufacturer's specification sheets. A table of positive voltage regulated ICs is provided in table 19.1.

TABLE 19.1 Positive Voltage Regulators in 7800 series

IC Part	**Output Voltage (V)**	**Minimum Vi (V)**
7805	+5	7.3
7806	+6	8.3
7808	+8	10.5
7810	+10	12.5
7812	+12	14.6
7815	+15	17.7
7818	+18	21.0

7824	+24	27.1

MAX 232:

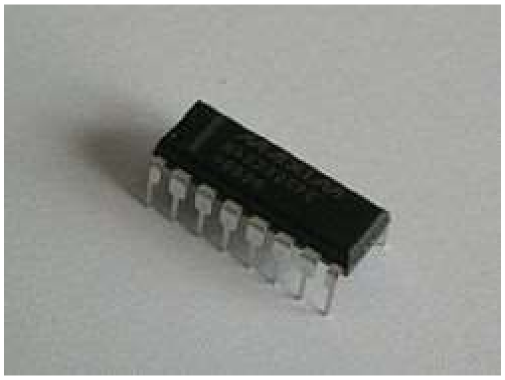
The **MAX232** is an integrated circuit, first created by Maxim Integrated Products, that converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits. The MAX232 is a dual driver/receiver and typically converts the RX, TX, CTS and RTS signals.

The drivers provide RS-232 voltage level outputs (approx. ± 7.5 V) from a single + 5 V supply via on-chip charge pumps and external capacitors. This makes it useful for implementing RS-232 in devices that otherwise do not need any voltages outside the 0 V to + 5 V range, as power supply design does not need to be made more complicated just for driving the RS-232 in this case.

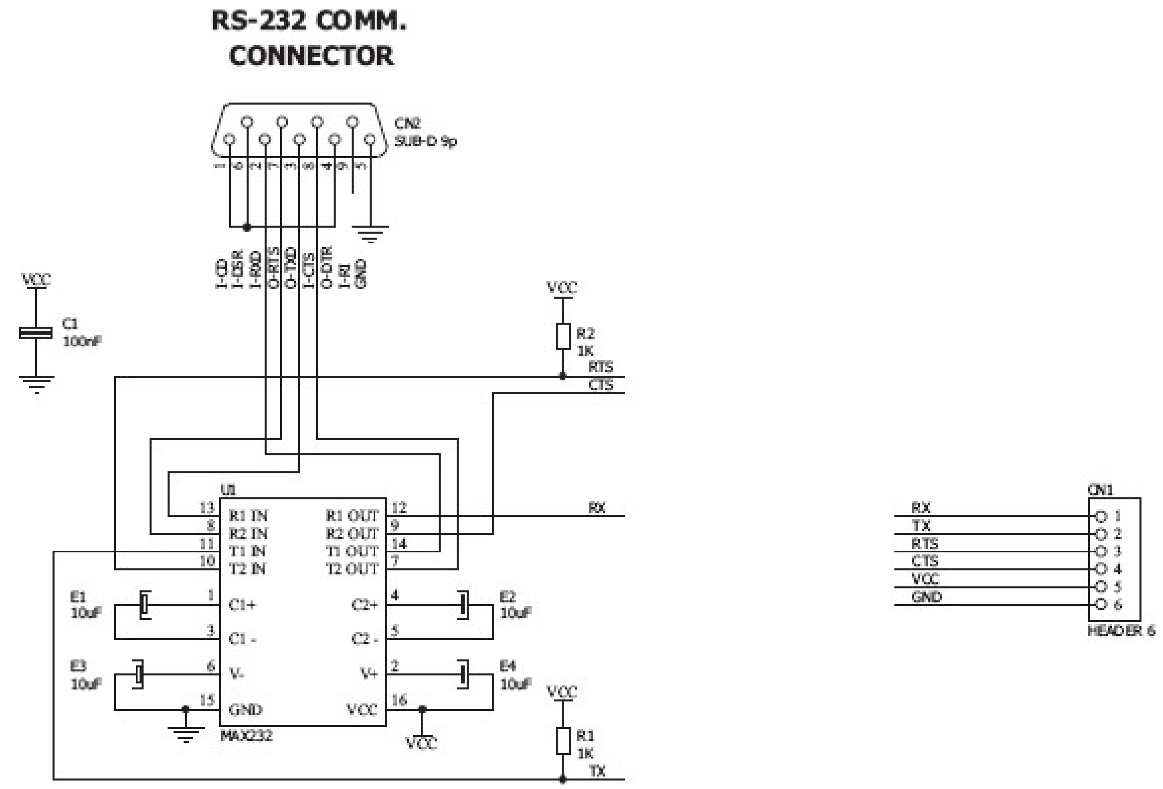
The receivers reduce RS-232 inputs (which may be as high as ± 25 V), to standard 5 V TTL levels. These receivers have a typical threshold of 1.3 V, and a typical hysteresis of 0.5 V.

The later MAX232A is backwards compatible with the original MAX232 but may operate at higher baud rates and can use smaller external capacitors – 0.1 μf in place of the 1.0 μF capacitors used with the original device.

The newer MAX3232 is also backwards compatible, but operates at a broader voltage range, from 3 to 5.5 V



MAX232 CIRCUIT



VOLTAGE LEVEL

It is helpful to understand what occurs to the voltage levels. When a MAX232 IC receives a TTL level to convert, it changes a TTL Logic 0 to between +3 and +15 V, and changes TTL Logic 1 to between -3 to -15 V, and vice versa for converting from RS232 to TTL. This can be confusing when you realize that the RS232 Data Transmission voltages at a certain logic state are opposite from the RS232 Control Line voltages at the same logic state. To clarify the matter, see the table below. For more information see RS-232 Voltage Levels.

RS232 Line Type & Logic Level	**RS232 Voltage**	**TTL Voltage to/from MAX232**
Data Transmission (Rx/Tx) Logic 0	+3 V to +15 V	0 V
Data Transmission (Rx/Tx) Logic 1	-3 V to -15 V	5 V
Control Signals (RTS/CTS/DTR/DSR) Logic 0	-3 V to -15 V	5 V
Control Signals (RTS/CTS/DTR/DSR) Logic 1	+3 V to +15 V	0 V

SENSORS

BABY MOISTURE SENSORS

Baby moisture sensors measure the volumetric water content in Baby. Since the direct gravimetric measurement of free Baby moisture requires removing, drying, and weighting of a sample, Baby moisture sensors measure the volumetric water content indirectly by using some other property of the Baby, such as electrical resistance, dielectric constant, or interaction with neutrons, as a proxy for the moisture content. The relation between the measured property and Baby moisture must be calibrated and may vary depending on environmental factors such as Baby type, sound, or electric conductivity. Reflected microwave radiation is affected by the Baby moisture and is used for remote sensing in hydrology and agriculture. Portable probe instruments can be used by farmers or gardeners.

Baby moisture sensors typically refer to sensors that estimate volumetric water content. Another class of sensors measure another property of moisture in Babys called water potential; these sensors are usually referred to as Baby water potential sensors and include tensiometers and gypsum blocks Measuring Baby moisture is important for agricultural applications to help farmers manage their irrigation systems more efficiently.

Knowing the exact Baby moisture conditions on their fields, not only are farmers able to generally use less water to grow a crop, they are also able to increase yields and the quality of the crop by improved management of Baby moisture during critical plant growth stages.
In urban and suburban areas, landscapes and residential lawns are using Baby moisture sensors to interface with an irrigation controller. Connecting a Baby moisture sensor to a simple irrigation clock will convert it into a "smart" irrigation controller that prevents irrigation cycles when the Baby is already wet, e.g. following a recent rainfall event. Golf courses are using Baby moisture sensors to increase the efficiency of their irrigation systems to prevent over-watering and leaching of fertilizers and other chemicals into the ground.

SOUND SENSORS

Sound sensors are vital to a variety of everyday products. For example, household ovens, refrigerators, and thermostats all rely on sound maintenance and control in order to function properly. Sound control also has applications in chemical engineering. Examples of this include maintaining the sound of a chemical reactor at the ideal set-point, monitoring the sound of a possible runaway reaction to ensure the safety of employees, and maintaining the sound of streams released to the environment to minimize harmful environmental impact.

While sound is generally sensed by humans as “hot”, “neutral”, or “cold”, chemical engineering requires precise, quantitative measurements of sound in order to accurately control a process. This is achieved through the use of sound sensors, and sound regulators which process the signals they receive from sensors.

From a thermodynamics perspective, sound changes as a function of the average energy of molecular movement. As heat is added to a system, molecular motion increases and the system experiences an increase in sound. It is difficult, however, to directly measure the energy of molecular movement, so sound sensors are generally designed to measure a property which changes in response to sound. The devices are then calibrated to traditional sound scales using a standard (i.e. the boiling point of water at known pressure). The following sections discuss the various types of sensors and regulators.

Sound sensors are devices used to measure the sound of a medium. There are 2 kinds on sound sensors: 1) contact sensors and 2) noncontact sensors. However, the 3 main types are thermometers, resistance sound detectors, and thermocouples. All three of these sensors measure a physical property (i.e. volume of a liquid, current through a wire), which changes as a function of sound. In addition to the 3 main types of sound sensors, there are numerous other sound sensors available for use.

CHAPTER-9

CONCLUSION

A smart cradle with a baby monitoring system over IoT has been designed and fabricated to monitor a baby's vital parameters, such as crying condition, humidity, and ambient temperature. NodeMCU was used as the main controller board in the project's circuit design, because it had a built-in Wi-Fi module, which enabled the implementation of IoT concept in the developed system. The demand of IoT was achieved by using the NodeMCU due to its simplicity and open-source nature. Red meranti wood was used as the material to build the baby's cradle, because of its general use in woodworks and due to its workability. Improvements were made during the enhancement phases to ensure that the research outcomes achieved the objectives. The finished prototype was tested by using a mobile phone with a baby crying ringtone, which was placed in the cradle. When the mobile phone rang for a few seconds, the cradle started swinging because of the system's assumption that the baby was crying due to the detected sound. A notification was sent to the mobile phone of the user to signal that the baby is crying. The temperature and humidity of the surroundings were determined, and the mini fan was turned on if the measured temperature was above 28 ◦C. With the aid of NodeMCU, the parents can control the baby cradle and the mini fan using mobile apps or an Internet-connected computer. Realtime vision monitoring was achieved with the help of the wireless camera. The user can monitor the baby through the camera mobile application and talk to the baby through the built-in microphone on the wireless camera. The total cost of the developed system is greatly reduced to approximately RM 700 per unit, which is suitable for mass production after finalizing the prototype. Our system's GUI needs to be improved to overcome limitations of both Adafruit.io MQTT server webpage and MQTT Dash mobile application. We will develop our own web-based and Android-based dashboards for laptops, PCs and smartphones, to add more monitoring and controlling functionalities based on our system requirements. Another limitation of the developed system is the wireless camera used, which is can only be connected to a local network. Parents can only view the section where the camera is positioned when they are connected to the same network as that of the wireless camera. TransFlash card can be used for the camera to record the baby's activities, but it is not considered real-time monitoring. Therefore, for future works, the wireless camera can be changed into an IP camera to enable IP hosting viewing in the network. The parents can type the set IP address for the IP camera in the network browser to monitor the baby’s conditions in real time. In addition, other future works can be conducted to further improve this system. A lighter and safer material, such as soft plastic, can be used to replace the wood materials to ensure the safety of the baby and reduce the weight of the baby cradle. A sound sensor with better quality can be implemented for better noise capturing; along with some coding change.

10.APPENDIX 1

Source Code:

```
#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <PubSubClient.h>



const char* ssid = "SMART-G";

const char* password = "10112019";


#define WT A0

#define CY D0



#define ID "yachlf"

#define DEVICE_TYPE "ESP8266"

#define DEVICE_ID "CHILD"

#define TOKEN "IOT-1234567"


char server[] = ID ".messaging.internetofthings.ibmcloud.com";

char publish_Topic1[] = "iot-2/evt/Data1/fmt/json";

char publish_Topic2[] = "iot-2/evt/Data2/fmt/json";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ID ":" DEVICE_TYPE ":" DEVICE_ID;
```

```
WiFiClient wifiClient;
PubSubClient client(server, 1883, NULL, wifiClient);

void setup() {
   Serial.begin(115200);
//    dht.begin();
   Serial.println();
   WiFi.begin(ssid, password);
   while (WiFi.status() != WL_CONNECTED) {
     delay(500);
     Serial.print(".");
   }
   Serial.println("");
   Serial.println(WiFi.localIP());

   if (!client.connected()) {
      Serial.print("Reconnecting client to ");
      Serial.println(server);
      while (!client.connect(clientId, authMethod, token)) {
         Serial.print(".");
         delay(500);
      }
      Serial.println("Connected TO IBM IoT cloud!");
   }
```

```
}


long previous_message = 0;
void loop() {
   client.loop();
   long current = millis();
   if (current - previous_message > 500) {
       previous_message = current;
        float hum = map(analogRead(A0), 0, 1023, 100, 0);
        float temp = map(digitalRead(D0), 0, 1, 100, 0);
        if (isnan(hum) || isnan(temp)  ){
   Serial.println(F("Failed to read sensor!"));
   return;
  }


  Serial.print("Cry Sound: ");
  Serial.print(temp);
  Serial.print(" Wed Level: ");
  Serial.print(hum);
  Serial.print("%");


      String payload = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
             payload += ",\"Cry Sound\":";
             payload += temp;
             payload += "}}";
   Serial.print("Sending payload: ");
   Serial.println(payload);

   if (client.publish(publish_Topic1, (char*) payload.c_str())) {
      Serial.println("Published successfully");
   } else {
      Serial.println("Failed");
   }
  String payload1 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
        payload1 += ",\"Wed Level\":";
        payload1 += hum;
        payload1 += "}}";
        Serial.print("Sending payload: ");
        Serial.println(payload1);
        Serial.println('\n');

    if (client.publish(publish_Topic2, (char*) payload1.c_str())) {
      Serial.println("Published successfully");
   } else {
      Serial.println("Failed");
   }
}
```

Appendix 2

Github link : https://github.com/IBM-EPBL/IBM-Project-15064-1659593882

Demo Video Link: https://youtu.be/Hjw8GzF_XQ8