EMERGENCY COMMUNICATION DEVICE FOR CHILDREN

A MINI PROJECT REPORT

Submitted by

BALAJI.D	**211420105013**
VIGNESH.N	**211420105338**
VINAY.K.R	**211420105339**
NAVEENKUMAR.P	**211420105070**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRICAL AND ELECTRONICS ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MAY 2023

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institute, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this Mini project report **"EMERGENCY COMMUNICATION DEVICE FOR CHILDREN"** is the bonafide work of **"BALAJI.D(211420105013), VIGNESH.N (211420105338), VINAY.K.R (211420105339), NAVEENKUMAR.P (211420105070)** who carried out the project work under my supervision.

SIGNATURE
Dr. S. SELVI, M.E, Ph.D.
HEAD OF THE DEPARTMENT
PROFESSOR
Department of Electrical and
Electronics Engineering,
Panimalar Engineering College,
Chennai - 600 123

SIGNATURE
Mr. K. KIRUBAKARAN, M.E
SUPERVISOR
ASSISTANT PROFESSOR
Department of Electrical and
Electronics Engineering,
Panimalar Engineering College,
Chennai - 600 123

Submitted for End Semester Mini Project Viva Voce held on...........................at Panimalar Engineering College, Chennai.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

Our sincere thanks to our Honourable Founder and Chairman **Dr. JEPPIAR, M.A., B.L., Ph.D.,** for his sincere endeavor us in his premier institution.

We would like to express our deep gratitude to our beloved Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., M.Phil., Ph.D.,** for his enthusiastic motivation which inspired us a lot in completing this project and our sincere thanks to our Directors **Mrs. C. VIJAYA RAJESWARI, Dr. C. SHAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for providing us with necessary facilities for the completion of this project.

We would like to express thanks to our Principal**, Dr. K. MANI, M.E., Ph.D.,** for extended his guidance and cooperation.

We would also like to thank our Head of the Department, **Dr. S. Selvi, M.E., Ph.D.,** Professor and Head of Department of Electrical and Electronics Engineering.

Personally, we thank our guide **Mr. K. KIRUBAKARAN, M.E.,** Assistant Professor in Department of Electrical and Electronics Engineering for the persistent motivation and support for this project, who at all times was the mentor of germination of this project from a small idea.

We express our sincere thanks to the project coordinators **Dr. S. DEEPA, M.E., Ph.D., & Dr. N. MANOJ KUMAR, M.E., Ph.D.,** in Department of Electrical and Electronics Engineering for the valuable suggestions from time to time at every stage of our project.

Finally, we would like to take the opportunity to thank our family members, faculty and non-teaching staff members of our department, friends, well-wishers who have helped us for the successful completion of our project.

ABSTRACT

India is the most population country in the world and there is lot of people living in this country including children and women, they go to schools and office, and when they are in danger some will fight and some don’t, in that time the smart work is to communicate with the police or parents to get help form them. This project deals with the communication device which can be worn by the children. This device does not need any expensive technology and it is user friendly. Both educated and uneducated people can use this device effectively. The main purpose of this device is to communicate their parents when the children are in danger or they feel insecure, this device is voice call communication device it can transfer the message from both side and it is easy to use.

TABLE OF CONTENT

LIST OF ABBREVIATIONS

Abbreviation	Description
ADC	Analog-to-Digital Converter
AMR	Adaptive Multi-Rate
BT	Bluetooth
CS	Coding Scheme
CSD	Circuit Switched Data
CTS	Clear to Send
DTE	Data Terminal Equipment (typically computer, terminal, printer)
DTR	Data Terminal Ready
DTX	Discontinuous Transmission
EFR	Enhanced Full Rate
EGSM	Enhanced GSM
ESD	Electrostatic Discharge
ETS	European Telecommunication Standard
FR	Full Rate
GPRS	General Packet Radio Service
GSM	Global Standard for Mobile Communications
HR	Half Rate
IMEI	International Mobile Equipment Identity
Li-ion	Lithium-Ion
MO	Mobile Originated
MS	Mobile Station (GSM engine), also referred to as TE
MT	Mobile Terminated
PAP	Password Authentication Protocol
PBCCH	Packet Broadcast Control Channel
PCB	Printed Circuit Board
PCL	Power Control Level
PCS	Personal Communication System, also referred to as GSM 1900
PDU	Protocol Data Unit
PPP	Point-to-point protocol
RF	Radio Frequency
RMS	Root Mean Square (value)
RTC	Real Time Clock
RX	Receive Direction
SIM	Subscriber Identification Module
SMS	Short Message Service
TDD	Time Division Distortion
TE	Terminal Equipment, also referred to as DTE

CHAPTER 1
INTRODUCTION

In today's world, parents often worry about the safety of their children, especially when they are away from home. Children are prone to getting lost, and in emergency situations, they may not be able to communicate their location or needs effectively. In such scenarios, an emergency communication device can prove to be a lifesaver.

An emergency communication device is a small, portable device that children can carry with them at all times. It is specifically designed to help them communicate with their parents or guardians in case of an emergency. These devices usually come with a range of features, including GPS tracking, two-way communication, and an SOS button.

The GPS tracking feature allows parents to keep an eye on their child's location in real-time. This feature is particularly useful when children are in unfamiliar locations or when they are traveling alone. The two-way communication feature enables children to talk to their parents or guardians whenever they need to, and vice versa. This feature can provide peace of mind to both parents and children, knowing that they can communicate with each other whenever necessary. The SOS button is another important feature of these devices, which allows children to call for help in case of an emergency.

Emergency communication devices are not only helpful in emergency situations but can also be used as a tool to teach children about responsibility and independence. By carrying and using these devices, children can learn to be more aware of their surroundings and take responsibility for their safety. Overall, emergency communication devices are a valuable tool for parents and guardians who want to ensure their children's safety. In the following pages, we will discuss in more detail the different types of emergency communication devices available for children, their features, and how they work. We will also explore the benefits and limitations of using these devices and provide some tips on how to choose the right device

CHAPTER 2
LITERATURE SURVEY

"**Development of an emergency communication device for children**" by K. Sivakumar, S. P. Yuvraj, S. Natarajan, and S. Shanmugam (2018) - This study focused on developing an emergency communication device for children that is small, lightweight, and easy to use. The device was equipped with a GPS tracker, two-way communication, and an SOS button. The authors tested the device's functionality and found that it was effective in communicating with parents or guardians during an emergency.

"**An emergency communication device for children using IoT**" by S. Anupriya and S. Sathish Kumar (2019) - This study proposed an emergency communication device for children that uses IoT technology. The device was designed to be worn as a watch and included features such as GPS tracking, two-way communication, and an SOS button. The authors tested the device's functionality and found that it was effective in communicating with parents or guardians during an emergency.

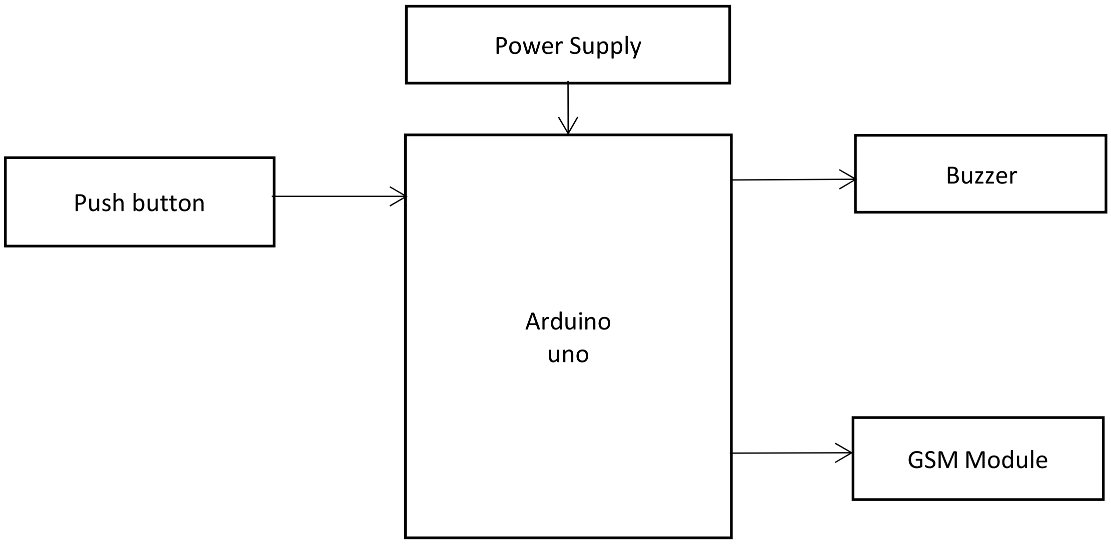
"**Design and implementation of an emergency communication system for children using wireless sensor network**" by S. P. Yuvraj and K. Sivakumar (2019) - This study proposed an emergency communication system for children using a wireless sensor network. The system included a wearable device with GPS tracking, two-way communication, and an SOS button, as well as a base station for parents or guardians to receive alerts. The authors tested the system's functionality and found that it was effective in communicating with parents or guardians during an emergency.

"**Evaluation of a wearable emergency communication device for children in outdoor environments**" by Y. Han and J. Lee (2020) - This study evaluated the performance of a wearable emergency communication device for children in outdoor environments. The device included GPS tracking, two-way communication, and an SOS button. The authors tested the device's functionality in various outdoor scenarios and found that it was effective in communicating with parents or guardians during an emergency.

"**Design and development of an emergency communication device for children with autism spectrum disorder**" by S. Rajendran and S. Balamurugan (2020) - This study focused on developing an emergency communication device for children with autism spectrum disorder (ASD). The device included features such as GPS tracking, two-way communication, and an SOS button, as well as a panic button for children with ASD. The authors tested the device's functionality and found that it was effective in communicating with parents or guardians during an emergency.

CHAPTER 3
EXISTING METHOD

3.1 BLOCK DIAGRAM



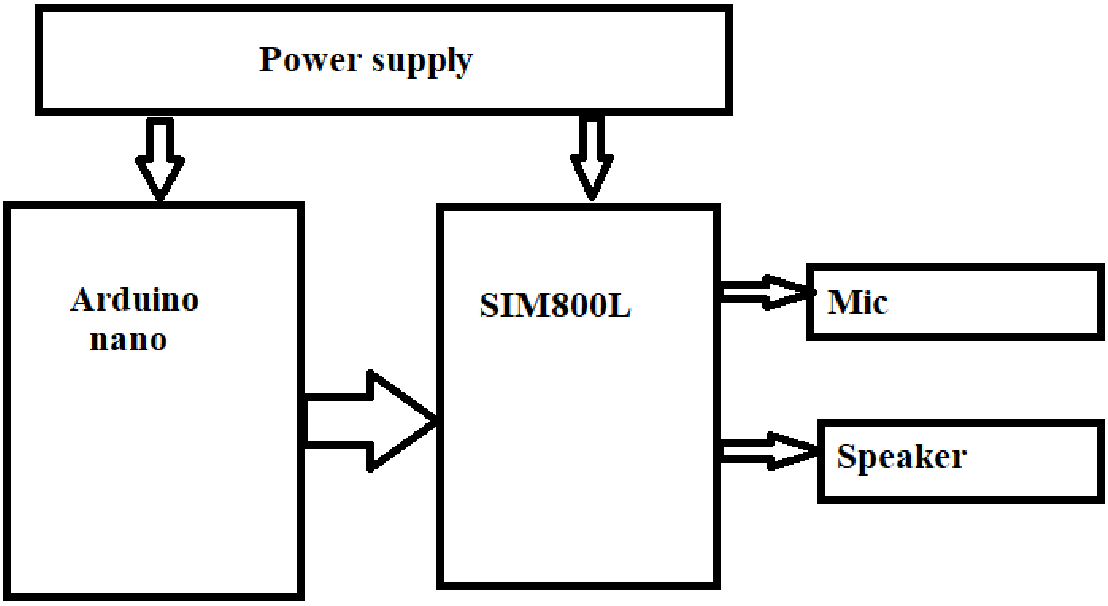
3.2 DESCRIPTION

There are several existing methods for child safety tracking devices, including:
GPS trackers: These are small devices that use GPS technology to track the location of a child. They can be attached to a child's clothing, backpack, or shoes and can be monitored by a parent's smartphone or computer.
Smartwatches: Some smartwatches are designed specifically for children and come with built-in GPS tracking. These devices allow parents to track their child's location and communicate with them via text messages or voice calls.
Bluetooth trackers: Bluetooth trackers are small devices that can be attached to a child's belongings, such as a backpack or keychain. They use Bluetooth technology to communicate with a smartphone app and can alert parents when their child is out of range.
RFID tags: RFID (Radio Frequency Identification) tags are small devices that can be

attached to a child's clothing or belongings. They can be used to track the child's location and can be monitored by a parent's smartphone or computer.

CHAPTER 4
PROPOSED METHOD

4.1 BLOCK DIAGRAM



4.2 DESCRIPTION

This is the device used for the communication in the emergency period this device is easy to use by just pressing one button and is small in size so easy to carry by the children. We use Arduino nano, SIM800L, Mic, speaker and 3.7 V battery for the power supply.

This device is work by just pressing the button in the Arduino nano. The microcontroller have the code in it, and the phone number of the parents or guardian, is inserted in the code in the microcontroller. The SIM800L is used to communicate with other devices, the speaker is mouth piece of the device and mic is a device that translates sound vibrations in the air into electronic signals and scribes them to a recording medium or over a loudspeaker. It used to record the sound and send that to the another.

CHAPTER 5

HARDWARE EQUIREMENTS

5.1 ARDUINO NANO

Arduino Nano is an intelligent development board designed for building faster prototypes with the smallest dimension. Arduino Nano being the oldest member of the Nano family, provides enough interfaces for your breadboard-friendly applications. At the heart of the board is **ATmega328 microcontroller** clocked at a frequency of 16 MHz featuring more or less the same functionalities as the Arduino Duemilanove. The board offers 22 digital input/output pins, 8 analog pins, and a mini-USB port.

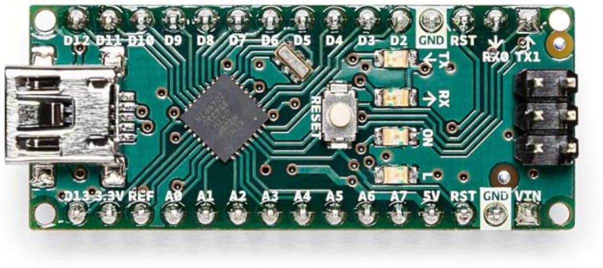

D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RX0 TX1
TX
RX
RESET
ON
L
D13 3.3V REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN

Figure 5.1 Arduino Nano

5.1.1 TARGET AREAS

Maker, Security, Environmental, Robotics and Control Systems

5.1.2 FEATURES

5.1.2.1 ATmega328 Microcontroller

High-performance low-power 8-bit processor

Achieve up to 16 MIPS for 16 MHz clock frequency

32 kB of which 2 KB used by bootloader

2 kB internal SRAM

1 kB EEPROM

32 x 8 General Purpose Working Registers

Real Time Counter with Separate Oscillator

Six PWM Channels

Programmable Serial USART

Master/Slave SPI Serial Interface

Power

Mini-B USB connection

6-20V unregulated external power supply (pin 30)

5V regulated external power supply (pin 27)

Sleep Modes

Idle

ADC Noise Reduction Power-save

Power-down Standby

Extended Standby

I/O

22 Digital

8 Analog

9 PWM Output

5.1.3 APPLICATION

Arduino Nano is the first embedded microcontroller in the Nano series with minimum functionalities, designed for mini projects from the maker community.

With a large number of input/output pins gives the advantage of utilizing several serial communications like UART, SPI and I2C. The hardware is compatible with Arduino IDE, Arduino CLI and web editor.

Security: The high-performance and low-power capabilities gives the chance to develop security based applications like access control systems using fingerprint sensors. The flexibility to interface sensors and external devices using serial communication has improved the scope of utility.

Environmental: The low-power feature of the microcontroller and the power supply options for the board has enhanced the ability to implement remote IoT projects related to environmental issues.

Robotics: Robotics has always been the favorite area of exploration for the Maker community and with this tiny embedded hardware you can now create complex and advanced robotic applications.

Related Products

Arduino Nano 33 BLE

Arduino 33 IoT

Arduino Micro

5.1.4 BLOCK DIAGRAM

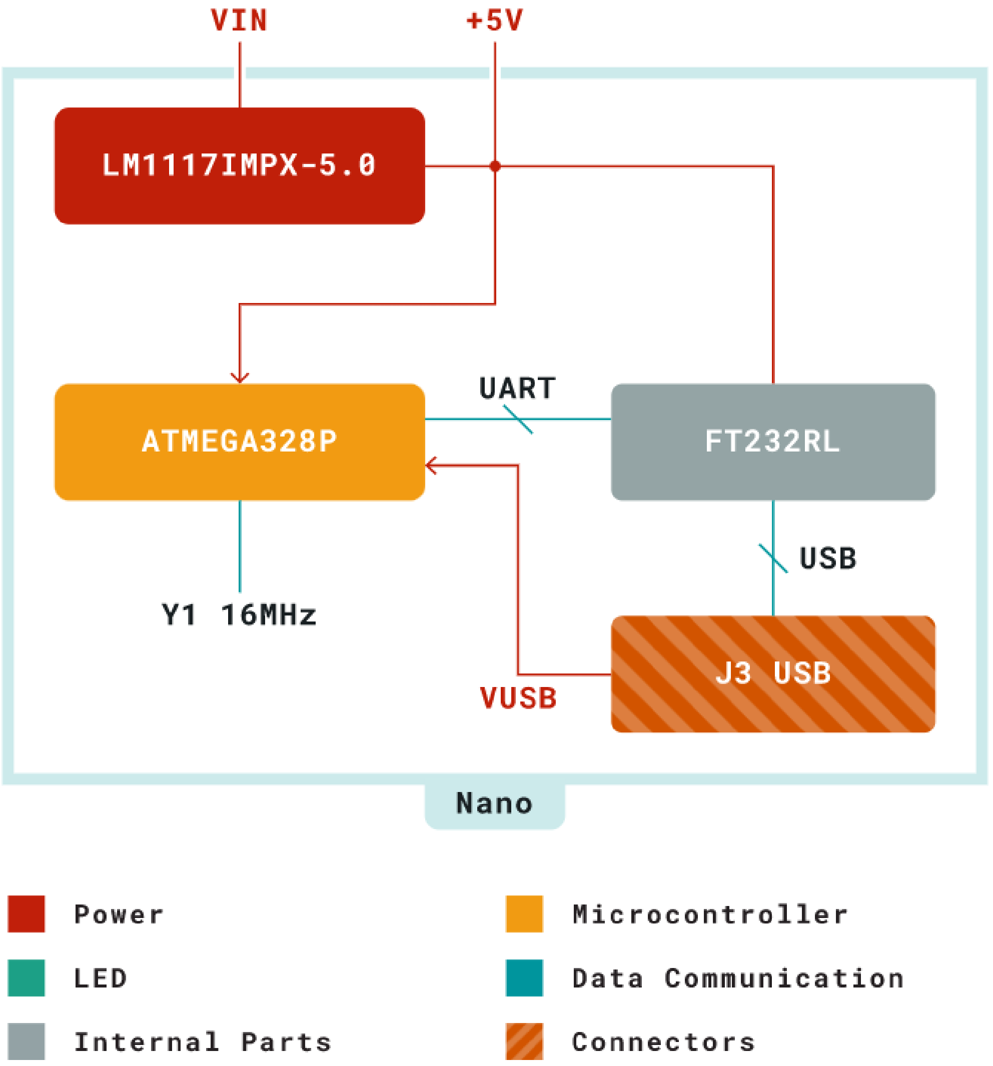


Figure 5.2 Block diagram

5.1.5 PROCESSOR

The primary processor in the Arduino Nano v3.3 board is the high-performance and low-power 8-bit ATmega328 microcontroller that runs at a clock frequency of 16 MHz. The ability to interface external devices through serial communication supported by the chip with UART TTL (5V), I2C (TWI) and SPI. Arduino Nano can be programmed with Arduino software reducing the entry barriers for new users. Smallest dimension embedded hardware makes it a perfect choice for breadboard-friendly projects from the maker community.

5.1.6 POWER TREE

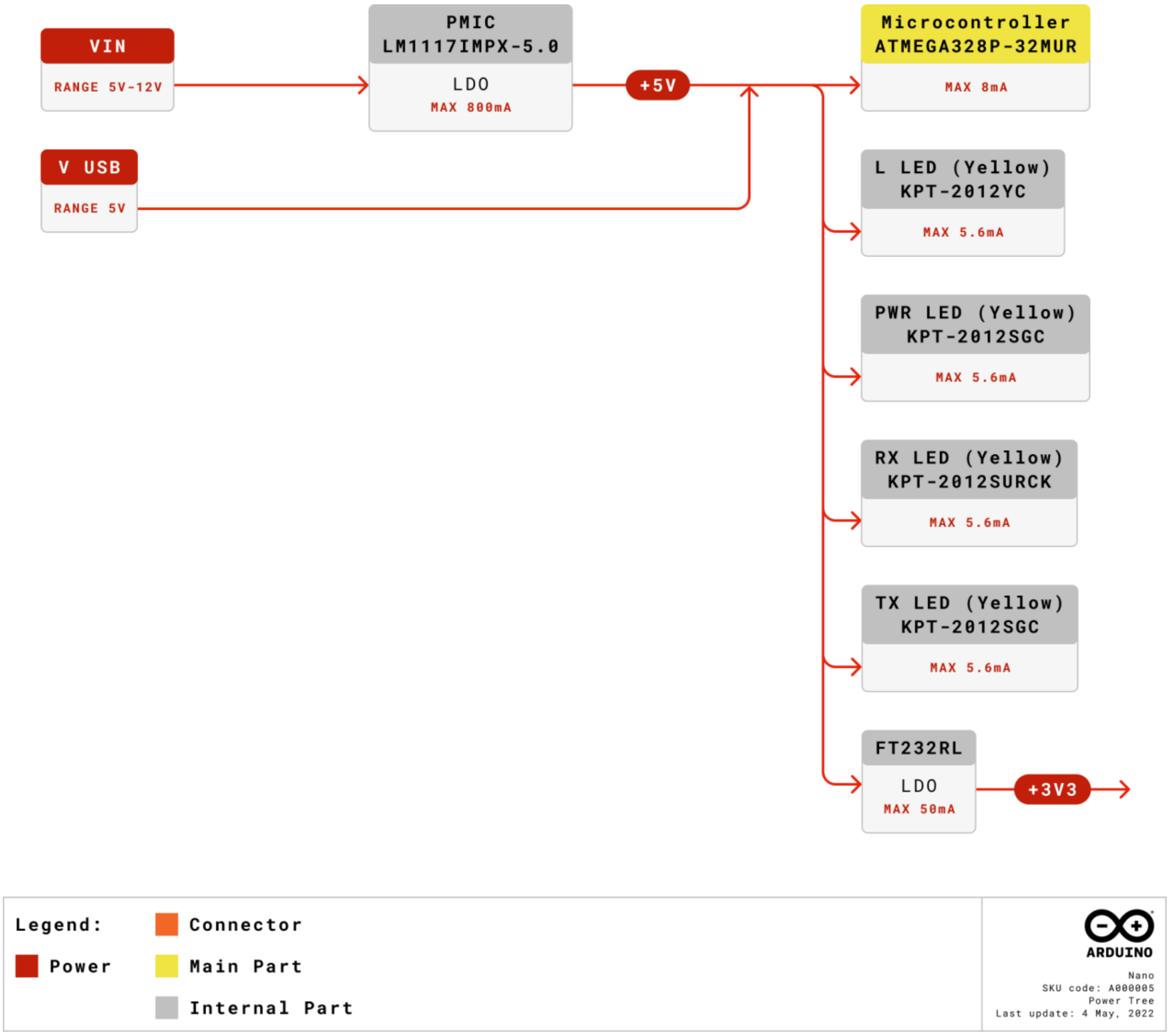


Figure 5.3 Power Tree

The Arduino Nano can be powered by either the USB port or alternatively via VIN. The input supply of VIN is regulated by an LDO so the supply is limited to 5V for the optimal functioning of the board. There is also another regulator which limits the voltage to 3.3V for powrering the components with low voltage requirements.

5.1.7 BOARD OPERATION

Getting Started – IDE

If you want to program your Arduino® Nano while offline you need to install the Arduino® Desktop IDE **[1]** To connect the Arduino Uno to your computer, you'll need a Micro-B USB cable. This also provides power to the board, as indicated by the LED.

Getting Started - Arduino Web Editor

All Arduino® boards, including this one, work out-of-the-box on the Arduino Web Editor **[2]**, by just installing a simple plugin. The Arduino Web Editor is hosted online, therefore it will always be up-to-date with the latest features and support for all boards. Follow **[3]** to start coding on the browser and upload your sketches onto your board.

Sample Sketches

Sample sketches for the Arduino® can be found either in the "Examples" menu in the Arduino® IDE or in the "Documentation" section of the Arduino website **[4]**

Online Resources

Now that you have gone through the basics of what you can do with the board you can explore the endless possibilities it provides by checking exciting projects on ProjectHub **[5]**, the Arduino® Library Reference **[6]** and the online store **[7]** where you will be able to complement your board with sensors, actuators and more.

5.1.8 CONNECTOR PINOUTS

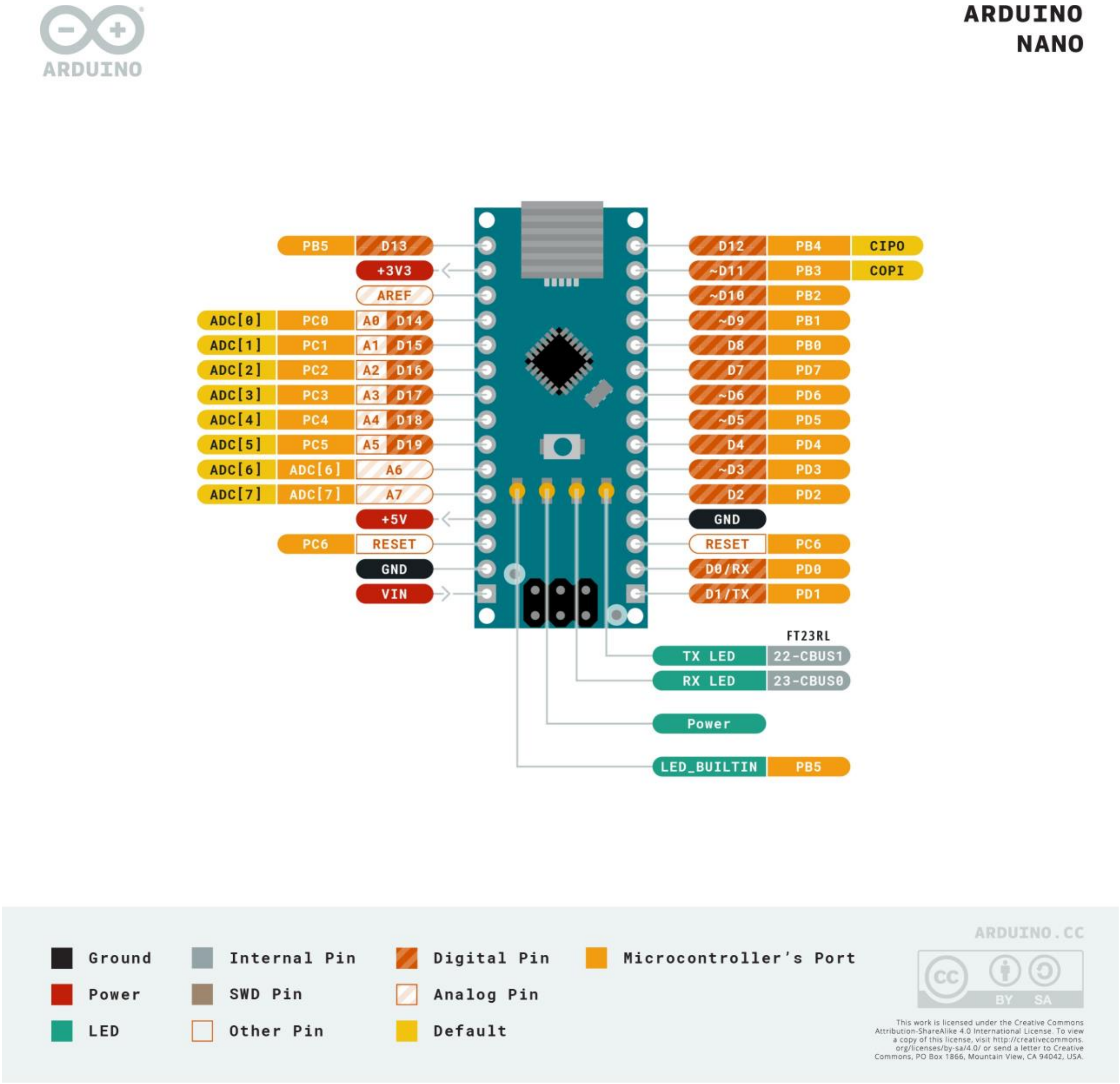


Figure 5.4 pin diagram

ANALOG

Pin	Function	Type	Description
1	+3V3	Power	5V USB Power
2	A0	Analog	Analog input 0 /GPIO
3	A1	Analog	Analog input 1 /GPIO
4	A2	Analog	Analog input 2 /GPIO
5	A3	Analog	Analog input 3 /GPIO
6	A4	Analog	Analog input 4 /GPIO
7	A5	Analog	Analog input 5 /GPIO
8	A6	Analog	Analog input 6 /GPIO
9	A7	Analog	Analog input 7 /GPIO
10	+5V	Power	+5V Power Rail
11	Reset	Reset	Reset
12	GND	Power	Ground
12	VIN	Power	Voltage Input

DIGITAL

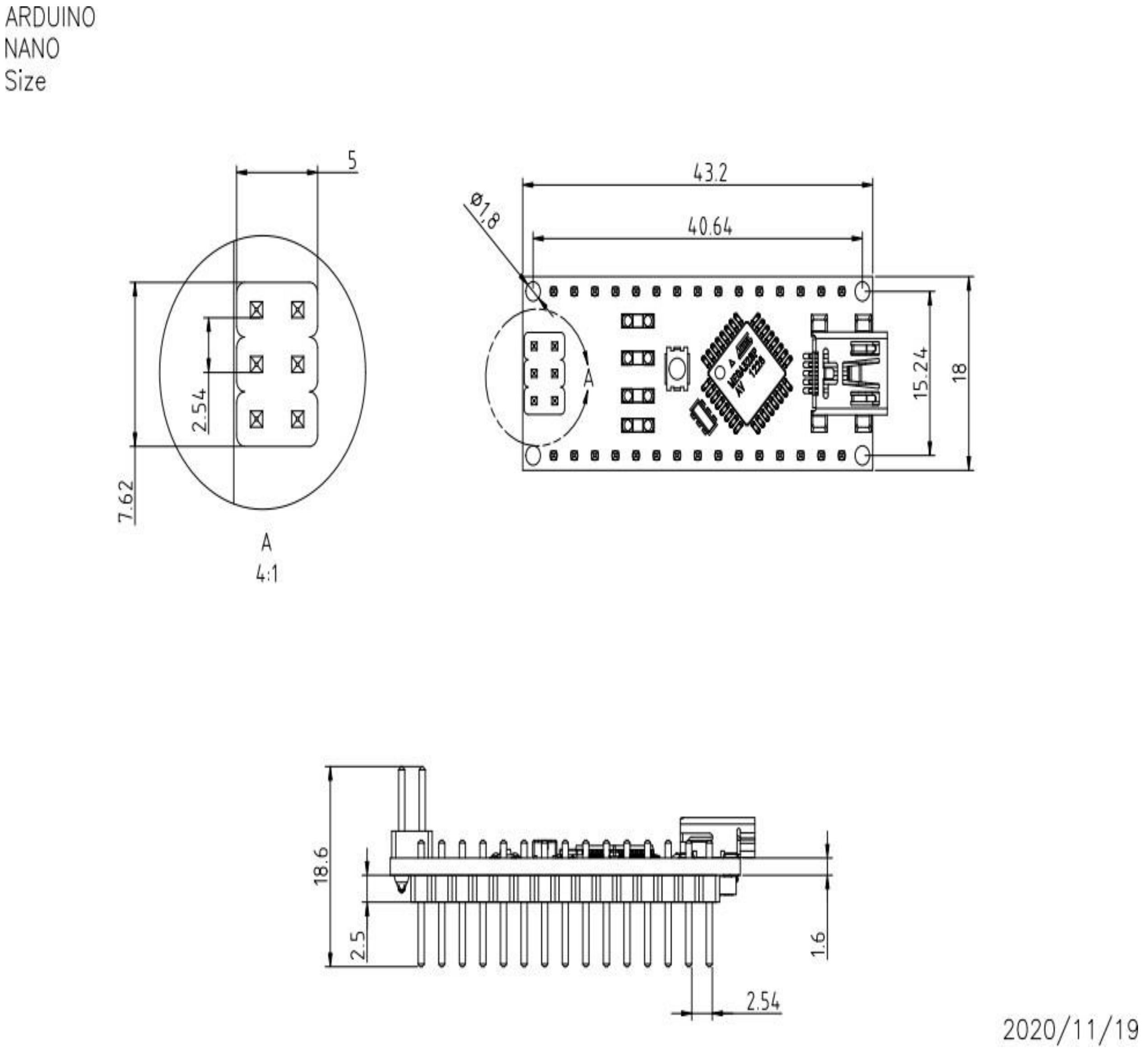
Pin	Function	Type	Description
1	D1/TX1	Digital	Digital Input 1 /GPIO
2	D0/RX0	Digital	Digital Input 0 /GPIO
3	D2	Digital	Digital Input 2 /GPIO
4	D3	Digital	Digital Input 3 /GPIO
5	D4	Digital	Digital Input 4 /GPIO
6	D5	Digital	Digital Input 5 /GPIO
7	D6	Digital	Digital Input 6 /GPIO
8	D7	Digital	Digital Input 7 /GPIO
9	D8	Digital	Digital Input 8 /GPIO
10	D9	Digital	Digital Input 9 /GPIO
11	D10	Digital	Digital Input 10 /GPIO
12	D11	Digital	Digital Input 11 /GPIO
13	D12	Digital	Digital Input 12 /GPIO
14	D13	Digital	Digital Input 13 /GPIO

15	Reset	Reset	Reset
16	GND	Power	Ground

ATmega328

Pin	Function	Type	Description
1	PB0	Internal	Serial Wire Debug
2	PB1	Internal	Serial Wire Debug
3	PB2	Internal	Serial Wire Debug
4	PB3	Internal	Serial Wire Debug
5	PB4	Internal	Serial Wire Debug
6	PB5	Internal	Serial Wire Debug

5.1.9 MECHANICAL INFORMATION



5.1.10 CERTIFICATIONS

Declaration of Conformity CE DoC (EU)

We declare under our sole responsibility that the products above are in conformity with the essential requirements of the following EU Directives and therefore qualify for free movement within markets comprising the European Union (EU) and European Economic Area (EEA).

Declaration of Conformity to EU RoHS & REACH 211 01/19/2021

Arduino boards are in compliance with RoHS 2 Directive 2011/65/EU of the European Parliament and RoHS 3 Directive 2015/863/EU of the Council of 4 June 2015 on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

Substance	Maximum Limit (ppm)
Lead (Pb)	1000
Cadmium (Cd)	100
Mercury (Hg)	1000
Hexavalent Chromium (Cr6+)	1000
Poly Brominated Biphenyls (PBB)	1000
Poly Brominated Diphenyl ethers (PBDE)	1000
Bis(2-Ethylhexyl} phthalate (DEHP)	1000
Benzyl butyl phthalate (BBP)	1000
Dibutyl phthalate (DBP)	1000
Diisobutyl phthalate (DIBP)	1000

Arduino Boards are fully compliant with the related requirements of European Union Regulation (EC) 1907 /2006 concerning the Registration, Evaluation, Authorization and Restriction of Chemicals (REACH). We declare none of the SVHCs (https://echa.europa.eu/web/guest/candidate-list-table), the Candidate List of Substances of Very High Concern for authorization currently released by ECHA, is present in all products (and also package) in quantities totaling in a concentration equal or above 0.1%. To the best of our knowledge, we also declare that our products do not contain any of the substances listed on the "Authorization List" (Annex XIV of the REACH regulations) and Substances of Very High Concern (SVHC) in any significant amounts as specified by the Annex XVII of Candidate list published by ECHA 1907 /2006/EC.

5.2 SIM800L

5.2.1 Introduction

This document describes SIM800 hardware interface in great detail.

This document can help user to quickly understand SIM800 interface specifications, electrical and mechanical details. With the help of this document and other SIM800 application notes, user guide, users can use SIM800 todesign various applications quickly.

5.2.2 SIM800 Overview

Designed for global market, SIM800 is a quad-band GSM/GPRS module that works on frequencies GSM 850MHz, EGSM 900MHz, DCS 1800MHz and PCS 1900MHz. SIM800 features GPRS multi-slot class 12/ class10 (optional) and supports the GPRS coding schemes CS-1, CS-2, CS-3 and CS-4.

With a tiny configuration of 24*24*3mm, SIM800 can meet almost all the space requirements in users' applications, such as M2M, smart phone, PDA and other mobile devices.

SIM800 has 68 SMT pads, and provides all hardware interfaces between the module and customers' boards.

Support up to 5*5*2 Keypads.

One full function UART port, and can be configured to two independent serial ports.

One USB port can be used as debugging and firmware upgrading.

Audio channels which include a microphone input and a receiver output.

Programmable general purpose input and output.

One SIM card interface.

Support Bluetooth function.

Support one PWM.

PCM

SIM800 is designed with power saving technique so that the current consumption is as low as 1.2mA in sleep mode.

SIM800 integrates TCP/IP protocol and extended TCP/IP AT commands which are very useful for data transfer applications. For details about TCP/IP applications

5.2.3 SIM800 KEY FEATURES

Feature	**Implementation**
Power supply	3.4V ~4.4V
Power saving	Typical power consumption in sleep mode is 1.2mA (BS-PA-MFRMS=9)
Frequency bands	● SIM800 Quad-band: GSM 850, EGSM 900, DCS 1800, PCS 1900. SIM800 can search the 4 frequency bands automatically. The frequency bands also can be set by AT command “AT+CBAND”. For details, please refer to *document [1]*. ● Compliant to GSM Phase 2/2+
Transmitting power	● Class 4 (2W):GSM850,EGSM900 ● Class 1 (1W):DCS1800,PCS1900
GPRS connectivity	● GPRS multi-slot class 12（default）

5.2.4 OPERATING MODES

Mode	Function	
Normal operation	GSM/GPRS SLEEP	Module will automatically go into sleep mode if the conditions of sleep mode are enabling and there is no on air or hardware interrupt (such as GPIO interrupt or data on serial port). In this case, the current consumption of module will reduce to the minimal level. In sleep mode, the module can still receive paging message and SMS.
	GSM IDLE	Software is active. Module registered to the GSM network, and the module is ready to communicate.
	GSM TALK	Connection between two subscribers is in progress. In this case, the powerconsumption depends on network settings such as DTX off/on, FR/EFR/HR, hopping sequences, antenna.
	GPRS STANDBY	Module is ready for GPRS data transfer, but no data is currently sent or received. In this case, power consumption depends on network settingsand GPRS configuration.
	GPRS DATA	There is GPRS data transfer (PPP or TCP or UDP) in progress. In this case, power consumption is related with network settings (e.g. power control level); uplink/downlink data rates and GPRS configuration (e.g. used multi-slot settings).
Power off	Normal Power off by sending the AT command “AT+CPOWD=1” or using the PWRKEY. The power management unit shuts down the power supply for the baseband part of the module, and only the power supply for the RTC is remained. Software is not active. The serial port is not accessible. Power supply (connected to VBAT) remains applied.	
Minimum functionality mode	AT command “AT+CFUN” can be used to set the module to a minimum functionality mode without removing the power supply. In this mode, the RF part of the module will not work or the SIM card will not be accessible, or both RF part and SIM card will be closed, and the serial port is still accessible. The power consumption in this mode is lower than normal mode.	

5.2.5 SIM800 FUNCTIONAL DIAGRAM

The following figure shows a functional diagram of SIM800:

- GSM baseband engine
- PMU
- RF part
- Antenna interfaces

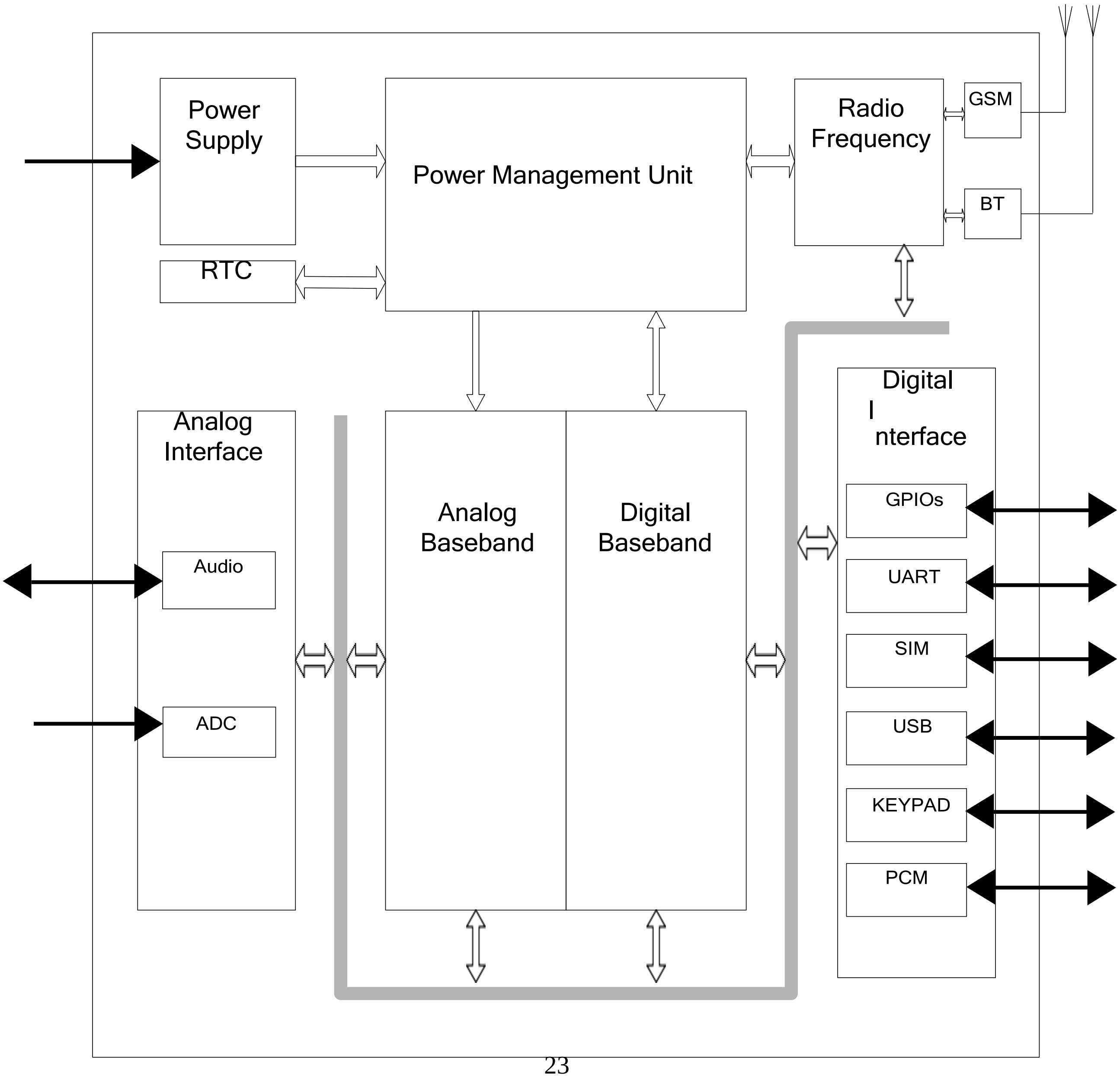


Figure 5.5 SIM800 FUNCTIONAL DIAGRAM

5.2.6 PIN OUT DIAGRAM

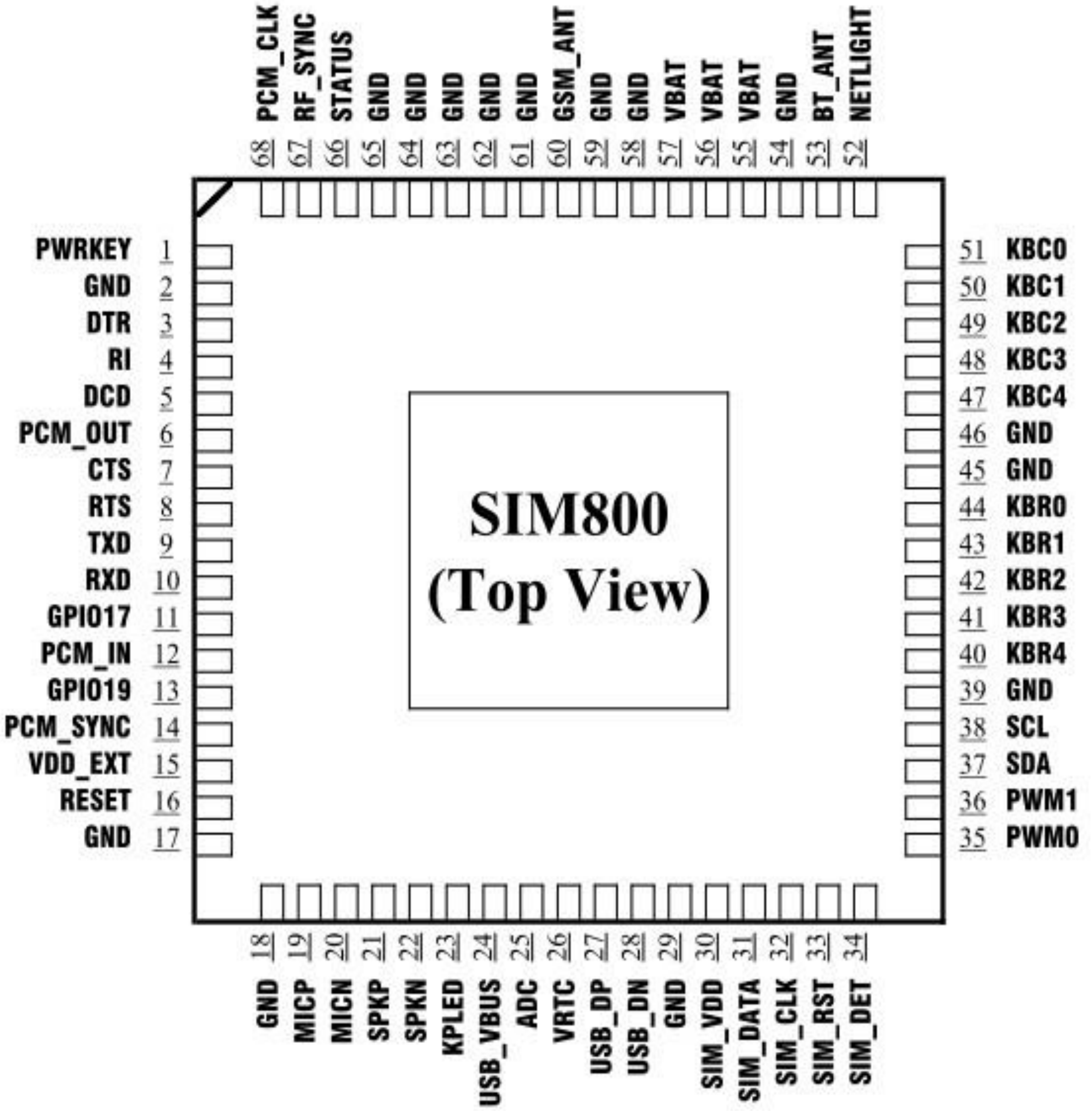


Figure 5.6 PIN OUT DIAGRAM

5.2.7 APPLICATION INTERFACE

Power Supply

The power supply range of SIM800 is from 3.4V to 4.4V. Recommended voltage is 4.0V. The transmitting burst will cause voltage drop and the power supply must be able to provide sufficient current up to 2A. For the VBAT input, a bypass capacitor (low ESR) such as a 100µF is strongly recommended.

For the VBAT input, a 100uF Tantalum capacitor (CA low ESR) and a 1uF~10uF Ceramics capacitor CB are strongly recommended .The 33pF and 10pF capacitors can effectively eliminate the high frequency interference. A 5.1V/500mW Zener diode is strongly recommended, the diode can prevent chip from damaging by the voltage surge. These capacitors and Zener diode should be placed as close to SIM800 VBAT pins as possible.

6 PCB Layout

Usually, most electronic products with good performance are based on good PCB layout. Poor PCB layout will lead to lots of issues, like TDD noise, SIM card undetected, etc. The final solution for these problems is to redesign PCB layout. Making good PCB layout will save developing schedule and cost as well.

This section will give some guidelines on PCB layout, in order to eliminate interfere or noise by greatest degree, and save product development period.

PIN Assignment

Before the placement of the PCB design, customer should learn well about PIN assignment in order to get reasonable layout with so many external components. Please refer to figure 2 for the details.

Principle of PCB Layout

During layout, we should pay attention to the following interfaces, like Antenna, power supply, SIM cardinterface, audio interface, and so on.

Antenna

There are some suggestions for components placing and routing of GSM and Bluetooth RF traces:

- ○ The RF connector is used for conducted test, so keep it as close to the GSM_ANT pin as possible;
- ○ Antenna matching circuit should be closed to the antenna;
- ○ Keep the RF traces as 50Ω；
- ○ The RF traces should be kept far away from the high speed signals and strong disturbing source.
- ○ If using a RF cable, kept it far away from SIM card, power ICs;

It is recommended that GSM antenna and Bluetooth antenna be placed as far as better.

SIM Card Interface

SIM card holder will take much more space on board, and there has no anti-EMI component inside, so, SIM card interface always be interfered. So, pay attention to this interface during layout. Ensure SIM card holder far way from antenna or RF cable. And it’s better to put SIM card holder near module, And it’s better to add ESD component to protect clock, data, reset and SIM_VDD signals which should be far away from power and high-speed signal

Audio Interface

In order to avoid TDD noise, or current noise, or some other noise, the signal trace of audio should far away from antenna and power, and it is recommended to surround audio traces by ground. And do not rout audio trace and VBAT trace parallel.

5.2.8 RECOMMENDED PCB LAYOUT

Based on above principles, recommended layout is shown in the following illustration.

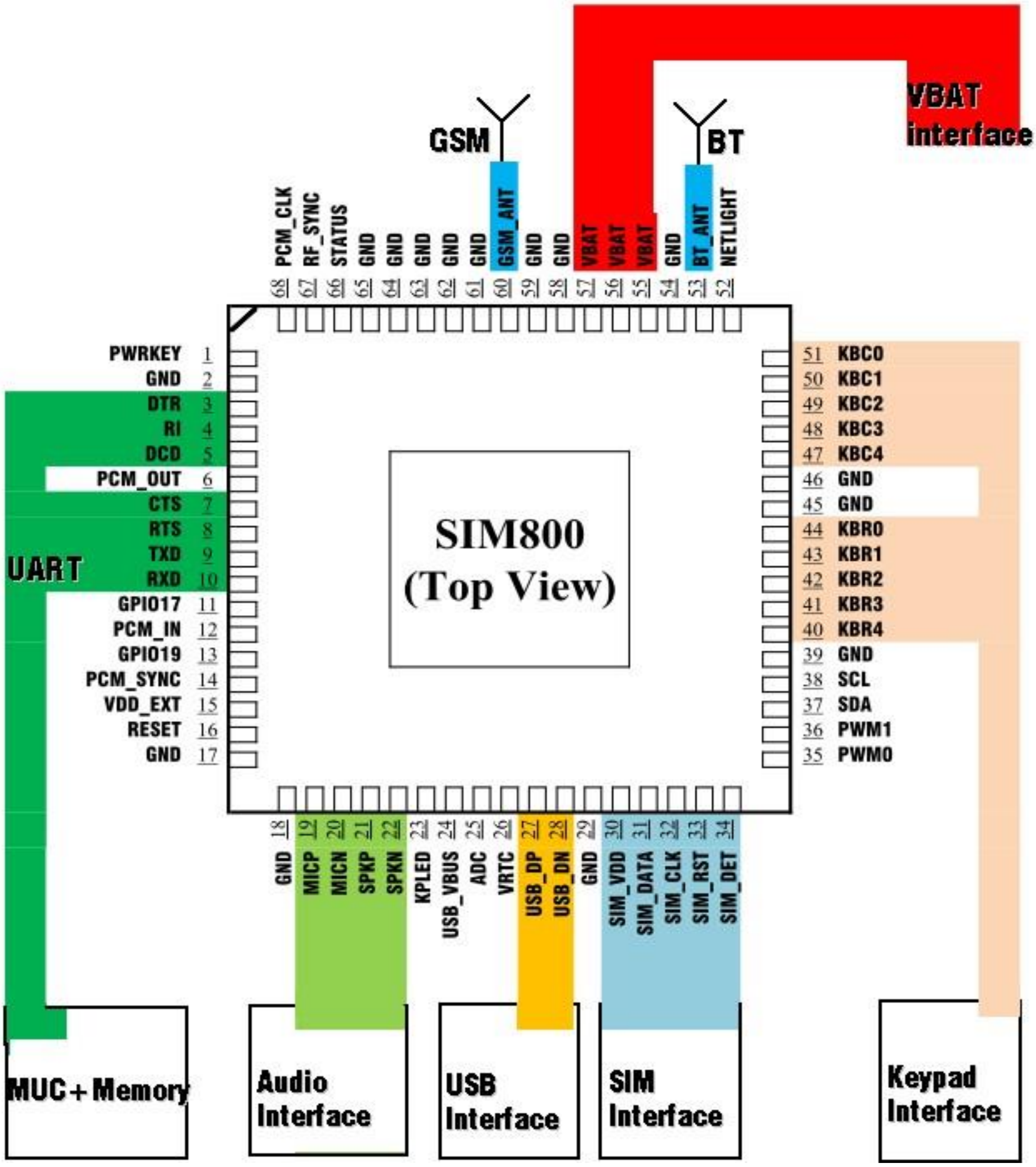


Figure 5.7 Recommended PCB layout

5.2.9 TOP AND BOTTOM VIEW OF SIM800

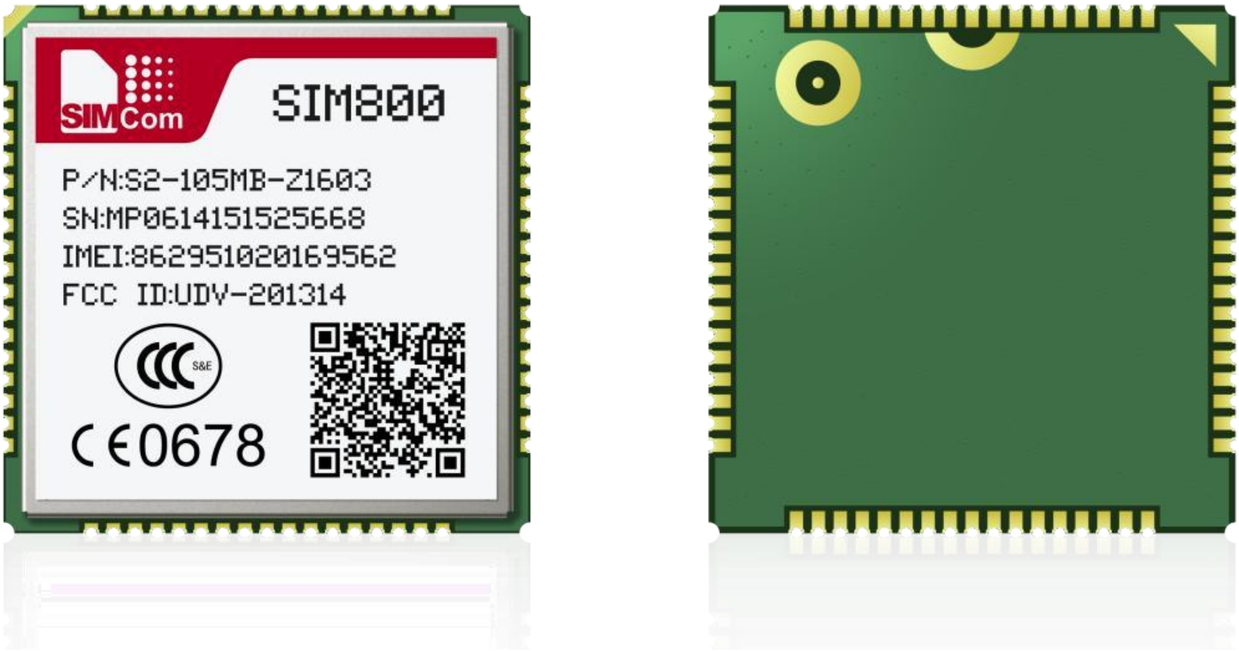


Figure 5.8 TOP AND BOTTOM VIEW OF SIM800

5.3 SPEAKERS

speakers are standard output devices used with computer systems that enable the listener to listen to a sound as an outcome. Some speakers are used once they have been linked to a computer, while others may be connected to any type of sound system.

FIGURE 5.9 SPEAKER

FEATURES

- Speaker of a computer helps to hear sound.
- Visual audio mode is available.
- It also made alert sounds .
- Audio discription that read aloud.
- Narrator is main key.

5.4 MIC

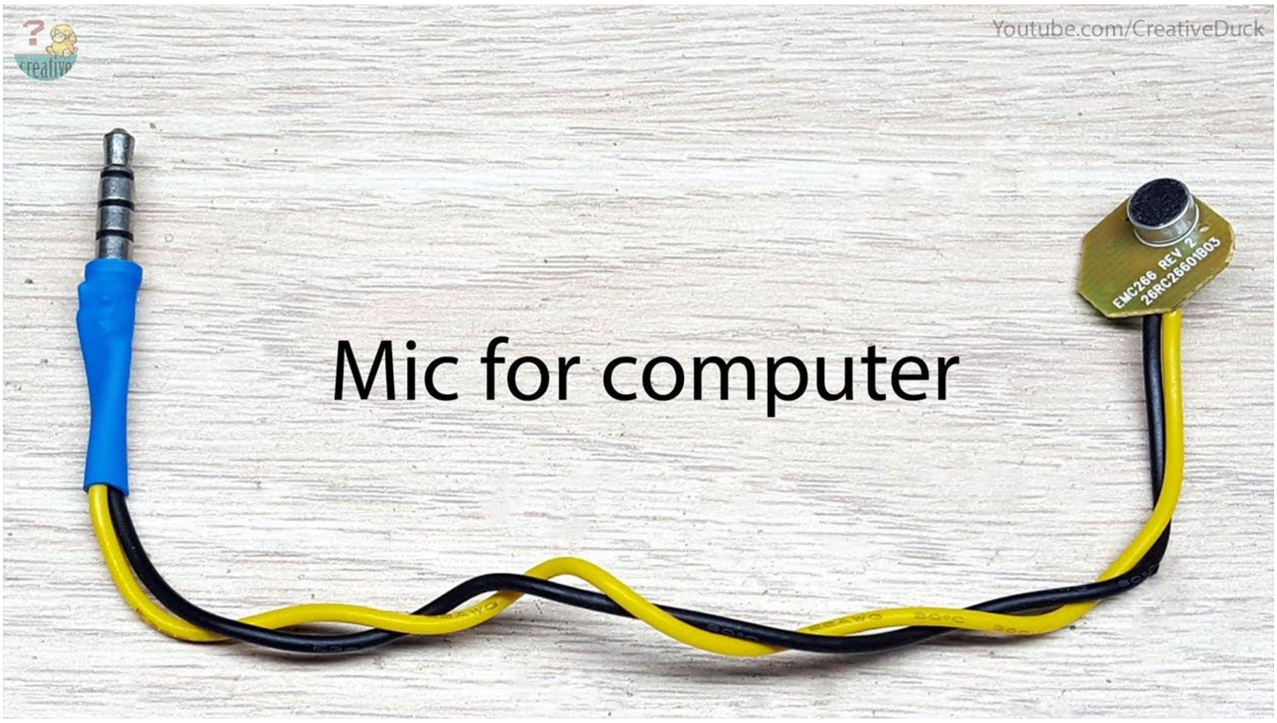


FIGURE 5.10 MIC

A microphone is a device that translates sound vibrations in the air into electronic signals and scribes them to a recording medium or over a loudspeaker. Microphones enable many types of audio recording devices for purposes including communications of many kinds, as well as music vocals, speech and sound recording.

The function of a microphone in a computer is to pick up sound, just like a microphone used by a singer on stage, and convert the audio sound waves into electrical signals. From there, these electrical signals travel to the sound card, where they are converted into digital signals, able to be processed by the computer.

5.5 INTERNET OF THINGS (IoT)

- The term Internet of Things generally refers to scenarios where network connectivity and computing capability extends to objects, sensors and everyday items not normally considered computers, allowing these devices to generate, exchange and consume data with minimal human intervention.

- Enabling Technologies: The concept of combining computers, sensors, and networks to monitor and control devices has existed for decades. The recent confluence of several technology market trends, however, is bringing the Internet of Things closer to widespread reality. These include Ubiquitous Connectivity, Widespread Adoption of IP-based Networking, Computing Economics, Miniaturization, Advances in Data.

- Connectivity Models: IoT implementations use different technical communications models, each with its own characteristics. Four common communications models described by the Internet Architecture Board include: Device-to-Device, Device-to-Cloud, Device-to-Gateway, and Back-End Data-Sharing. These models highlight the flexibility in the ways that IoT devices can connect and provide value to the user.

IoT devices are implemented using both hardware and software components. Dedicated hardware components are used to implement the interface with the physical world, and to perform tasks which are more computationally complex. Microcontrollers are used to execute software that interprets inputs and controls the system. This module discusses the roles of both the hardware and software components in the system.

The functions of common hardware components are described and the interface between the software and hardware through the microcontroller is explained. IoT devices often use an operating system to support the interaction between the software and the microcontroller. We will define the role of an operating system in an IoT device and how an IoT operating system differs from a standard one.

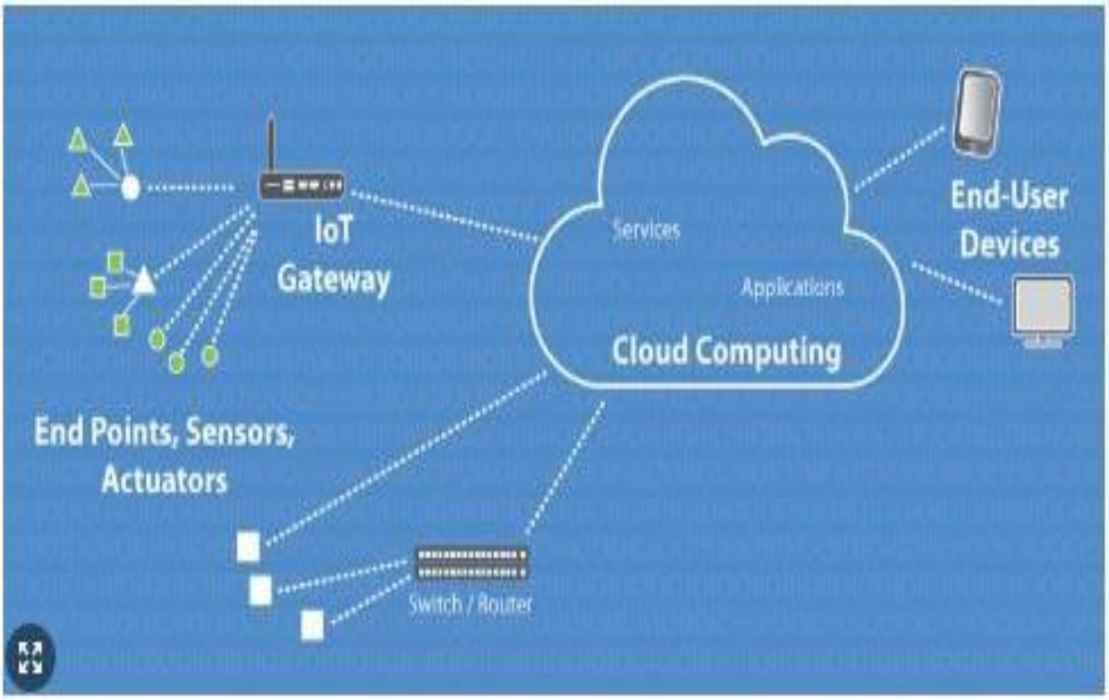


Figure 5.11 Cloud computing

How IoT works:

An IoT ecosystem consists of web-enabled smart devices that use embedded processors, sensors and communication hardware to collect, send and act on data they acquire from their environments. IoT devices share the sensor data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analyzed or analyzed locally.

Sometimes, these devices communicate with other related devices and act on the information they get from one another. The devices do most of the work without human intervention, although people can interact with the devices -- for instance, to set them up, give them instructions or access the data.

The connectivity, networking and communication protocols used with these web-enabled devices largely depend on the specific IoT applications deployed.

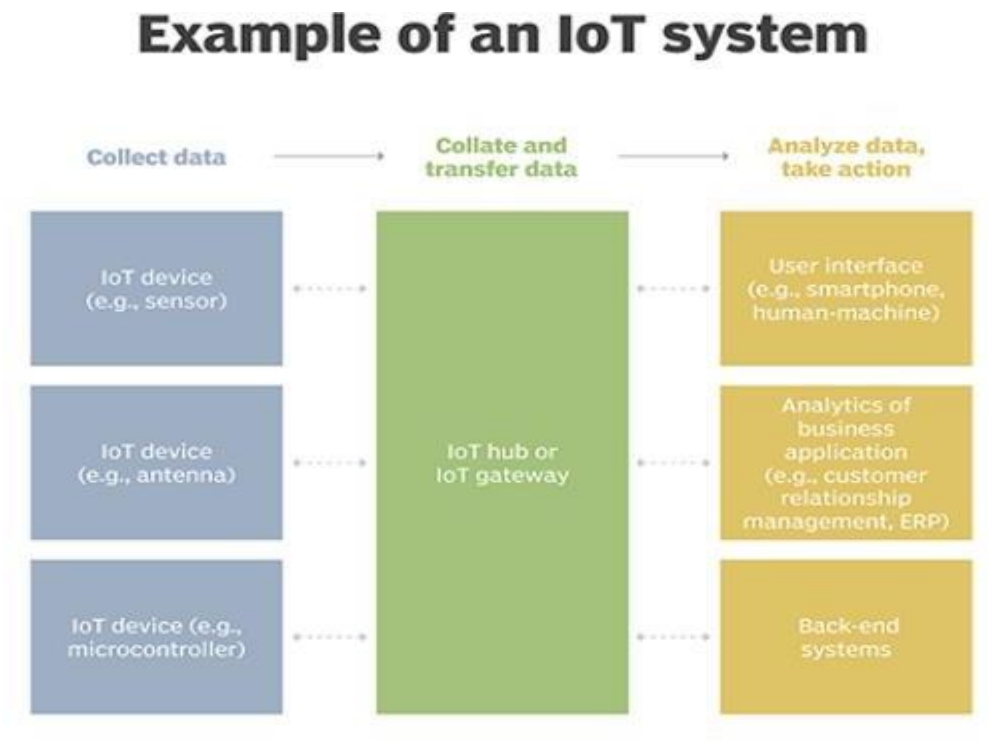

Example of an IoT system
Collect data
Collate and transfer data
Analyze data, take action
IoT device (e.g., sensor)
IoT device (e.g., antenna)
IoT device (e.g., microcontroller)
IoT hub or IoT gateway
User interface (e.g., smartphone, human-machine)
Analytics of business application (e.g., customer relationship management, ERP)
Back-end systems

Figure 5.12 Example of an IoT system

5.5.1 Benefits of IoT:

The internet of things offers a number of benefits to organizations, enabling them to:

- monitor their overall business processes
- improve the customer experience
- save time and money
- enhance employee productivity
- integrate and adapt business models
- make better business decisions and
- generate more revenue.

IoT encourages companies to rethink the ways they approach their businesses, industries and markets and gives them the tools to improve their business strategies.

Consumer and enterprise IoT applications

There are numerous real-world applications of the internet of things, raging from consumer IoT and enterprise IoT to manufacturing and industrial IoT (IoT). IoT applications span numerous verticals, including automotive, telecommunication and more.

In the consumer segment, for example, smart homes that are equipped with smart thermostats, smart appliances and connected heating, lighting and electronic devices can be controlled remotely via computers, smartphones or other mobile devices.

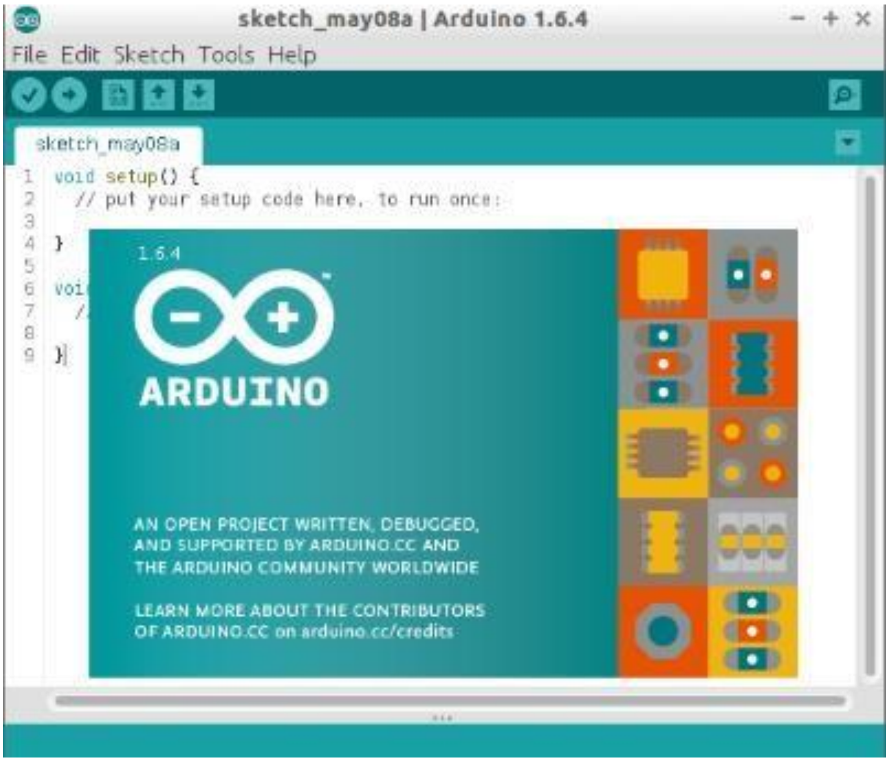
Wearable devices with sensors and software can collect and analyze user data, sending messages to other technologies about the users with the aim of making users' lives easier and more comfortable. Wearable devices are also used for public safety -- for example, improving first responder’s response times during emergencies by providing optimized routes to a location or by tracking construction workers' or firefighters' vital signs at life-threatening sites.

CHAPTER 6

SOFTWARE REQUIREMENT

6.1 ARDUINO IDE

The Arduino IDE is incredibly minimalistic, yet it provides a nearcomplete environment for most Arduino-based projects. The top menu bar has the standard options, including “File” (new, load save, etc.), “Edit” (font, copy, paste, etc.), “Sketch” (for compiling and programming), “Tools” (useful options for testing projects), and “Help”. The middle section of the IDE is a simple text editor that where you 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.



```
void setup() {
  // put your setup code here, to run once:

}

voi
  /

}
```

AN OPEN PROJECT WRITTEN, DEBUGGED, AND SUPPORTED BY ARDUINO.CC AND THE ARDUINO COMMUNITY WORLDWIDE

LEARN MORE ABOUT THE CONTRIBUTORS OF ARDUINO.CC on arduino.cc/credits

Figure 6.1 Arduino IDE software

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. While more advanced projects will take advantage of the built-in tools in the IDE, most projects will rely on the six buttons found below the menu bar.

1. The check mark is used to verify your code. Click this once you have written your code.
2. The arrow uploads your code to the Arduino to run.
3. The dotted paper will create a new file.
4. The upward arrow is used to open an existing Arduino project.
5. The downward arrow is used to save the current file.
6. The far right button is a serial monitor, which is useful for sending data from the Arduino to the PC for debugging purposes.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures.

User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the

GNU tool chain, also included with the IDE distribution.The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

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.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures.

User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution.

The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

6.2 EMBEDDED C

Embedded C Programming is the soul of the processor functioning inside each and every embedded system we come across in our daily life, such as mobile phone, washing machine, and digital camera. Each processor is associated with an embedded software. The first and foremost thing is the embedded software that decides functioning of the embedded system. Embedded C language is most frequently used to program the microcontroller.

Earlier, many embedded applications were developed using assembly level programming. However, they did not provide portability. This disadvantage was overcome by the advent of various high-level languages like C, Pascal, and COBOL. However, it was the C language that got extensive acceptance for embedded systems, and it continues to do so. The C code written is more reliable, scalable, and portable; and in fact, much easier to understand.

C language was developed by Dennis Ritchie in 1969. It is a collection of one or more functions, and every function is a collection of statements performing a specific task.

C language is a middle-level language as it supports high-level applications and low-level applications. Before going into the details of embedded C programming, we should know about RAM memory organization.

6.2.1 Salient features of the language

- C language is a software designed with different keywords, data types, variables, constants, etc.
- Embedded C is a generic term given to a programming language written in C, which is associated with a particular hardware architecture.
- Embedded C is an extension to the C language with some additional header files. These header files may change from controller to controller.
- The microcontroller 8051 #include is used. The embedded system

designers must know about the hardware architecture to write programs. These programs play prominent role in monitoring and controlling external devices. They also directly operate and use the internal architecture of the microcontroller, such as interrupt handling, timers, serial communication and other available features.

6.2.2 The basic additional features of the embedded software

Data types:

The data type refers to an extensive system for declaring variables of different types like integer, character, float, etc. The embedded C software uses four data types that are used to store data in the memory. The 'char' is used to store any single character; 'int' is used to store integer value, and 'float' is used to store any precision floating point value.

The size and range of different data types on a 32-bit machine is given in the following table. The size and range may vary on machines with different word sizes.

Keywords:

There are certain words that are reserved for doing specific tasks. These words are known as keywords. They are standard and predefined in the Embedded C.

Keywords are always written in lowercase. These keywords must be defined before writing the main program. The basic keywords of an embedded software are given below:

NAME	FUNCTIONS
sbit	Accessing of single bit
bit	Accessing of bit addressable memory of RAM
sfr	Accessing of sfr register by another name

Figure 6.2 Tables of keywords

sbit: This data type is used in case of accessing a single bit of SFR register.

- Syntax: sbit variable name = SFR bit
- Ex: sbit a=P2^1
- Explanation: If we assign p2.1 as ‘a’ variable, then we can use ‘a’ instead of p2.1 anywhere in the program, which reduces the complexity of the program.

Bit: This data type is used for accessing the bit addressable memory of RAM (20h-2fh).

- Syntax: bit variable name
- Ex: bit c
- Explanation: It is a bit sequence setting in a small data area that is used by a program to remember something.

SFR: This data type is used for accessing a SFR register by another name. All the SFR registers must be declared with capital letters.

- Syntax: SFR variable name = SFR address of SFR register
- Ex: SFR port0=0x80
- Explanation: If we assign 0x80 as ‘port0’, then we can use 0x80 instead of port0 anywhere in the program, which reduces the complexity of the program.

SFR Register: The SFR stands for ‘Special Function Register’. Microcontroller 8051 has 256 bytes of RAM memory. This RAM is divided into two parts: the first part of 128 bytes is used for data storage, and the other of 128 bytes is used for SFR registers.

All peripheral devices like I/O ports, timers and counters are stored in the SFR register, and each element has a unique address.

6.2.3 The Structure of an Embedded C Program

- comments
- preprocessor directives
- global variables
- main() function

{

- local variables
- statements
- …………..
- …………..

}

- fun(1)

{

- local variables
- statements
- …………..
- …………..

}

CHAPTER 7
HARDWARE
KIT

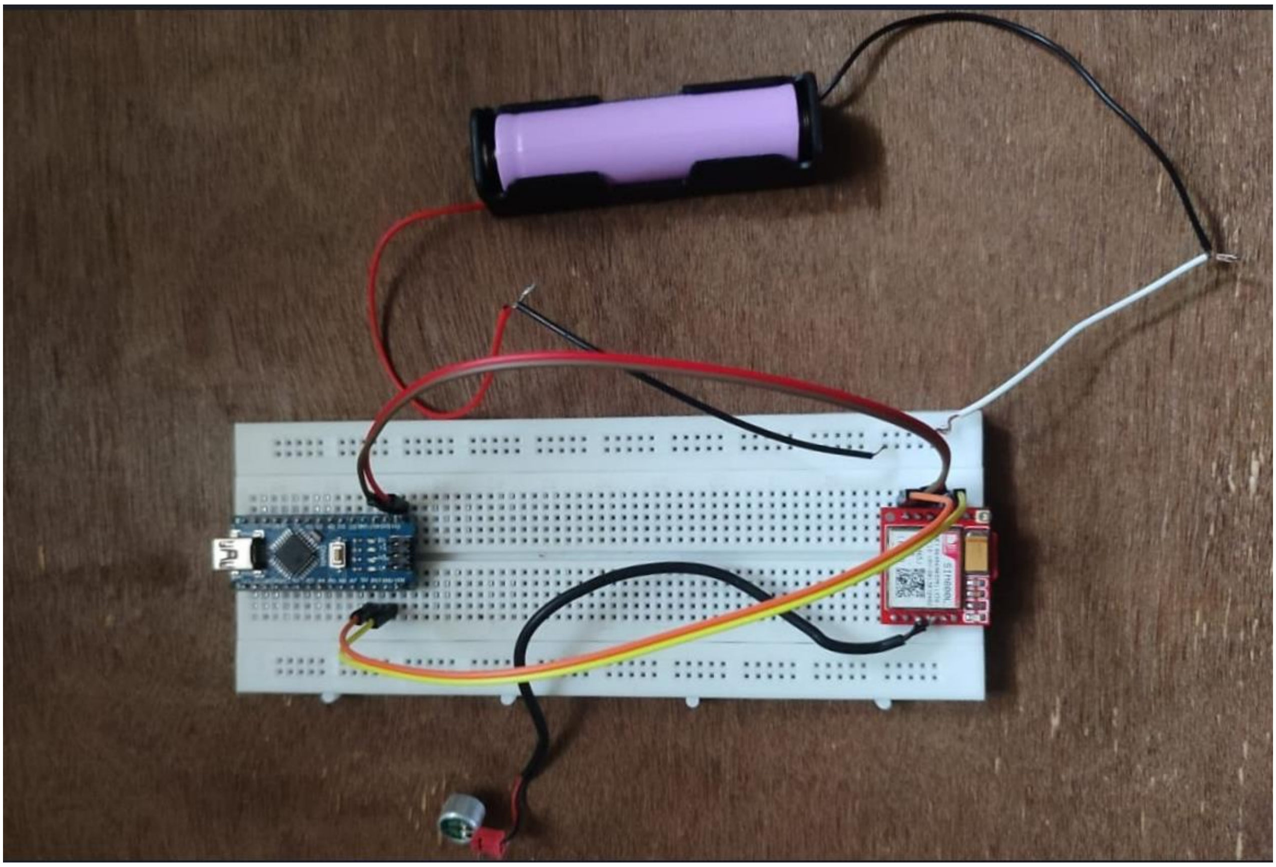


Figure 7.1 HARDWARE KIT

This is the device used for the communication in the emergency period this device is easy to use by just pressing one button and is small in size so easy to carry by the children. We use Arduino nano, SIM800L, Mic, speaker and 3.7 V battery for the power supply.

This device is work by just pressing the button in the Arduino nano. The microcontroller have the code in it, and the phone number of the parents or guardian, is inserted in the code in the microcontroller. The SIM800L is used to communicate with other devices, the speaker is mouth piece of the device and mic is a device that translates sound vibrations in the air into electronic signals and scribes them to a recording medium or over a loudspeaker. It used to record the sound and send that to the another.

APPENDIX

I.Code for the prototype

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

static const int RXPin = 4, TXPin = 3;
static const uint32_t GPSBaud = 9600;

TinyGPSPlus gps;
SoftwareSerial ss(RXPin, TXPin);

void setup() {
  Serial.begin(115200);
  ss.begin(GPSBaud);
}

void loop() {
  while (ss.available() > 0) {
    gps.encode(ss.read());
    if (gps.location.isUpdated()) {
      Serial.print("Latitude= ");
      Serial.print(gps.location.lat(), 6);
      Serial.print(" Longitude= ");
      Serial.println(gps.location.lng(), 6);
    }
  }
#include <SoftwareSerial.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <String.h>
#include <GSM.h>

#define PINNUMBER ""

SoftwareSerial gpsSerial(10, 11);
TinyGPSPlus gps;
SoftwareSerial gsmSerial(7, 8);
GSM gsmAccess;
```

```
void setup() {
  Serial.begin(9600);
  gpsSerial.begin(9600);
  gsmSerial.begin(9600);
  Serial.println("GPS and GSM modules initialized");
}

void loop() {
  while (gpsSerial.available() > 0) {
    if (gps.encode(gpsSerial.read())) {
      Serial.print("Latitude= ");
      Serial.print(gps.location.lat(), 6);
      Serial.print(" Longitude= ");
      Serial.println(gps.location.lng(), 6);
      sendSMS(gps.location.lat(), gps.location.lng());
    }
  }
}

void sendSMS(float lat, float lng) {
  String message = "Latitude: " + String(lat, 6) + ", Longitude: " + String(lng, 6);
  Serial.println("Sending SMS...");
  if (gsmAccess.begin(PINNUMBER) == GSM_READY) {
    SMS sms;
    sms.beginSMS("PHONE_NUMBER");
    sms.print(message);
    sms.endSMS();
    Serial.println("SMS sent successfully");
  } else {
    Serial.println("GSM initialization failed");
  }}
```

CONCLUSION

Emergency communication devices for children can be an essential tool for keeping children safe in emergency situations. These devices can provide a means of communication for children who may not have access to a phone or other communication device during an emergency, allowing them to quickly and easily contact their parents or guardians.

When selecting an emergency communication device for children, it is important to consider several factors. The device should be easy for children to use and understand, durable enough to withstand potential accidents or rough handling, and provide quick access to emergency services when necessary. GPS tracking can also be a valuable feature, allowing parents or guardians to locate their child in an emergency.

One of the main advantages of emergency communication devices for children is that they can provide peace of mind for parents or guardians, knowing that their child has a way to reach them in case of an emergency. In situations where a child may be separated from their parent or guardian, such as at a crowded event or in a public space, these devices can be particularly valuable.

Another advantage is that emergency communication devices can help children feel more independent and confident in their ability to handle emergency situations. With a device that allows them to easily communicate with a trusted adult, children may be more willing to explore and engage with the world around them, knowing that they have a safety net if something goes wrong.

There are several different types of emergency communication devices available for children, each with their own set of advantages and disadvantages. One common type is a GPS-enabled tracker, which allows parents or guardians to monitor their child's location and receive alerts if they leave a designated area.

REFERENCES

[1] M. Adil Khan; Kashif Nisar; Sana Nisar; BS Chowdhry; Ethisham Lodhi; Jawad Khan; Muhammad Reazul Haque "An Android-based tracking device".

[2] Do Ngoc HungVo Minh-ThanhQuoc Luong Huy Nguyen Triet "Design and Implementation of tracking the position".

[3] Pratibha Pawaskar "Smart location Assisted with Android Application and Save Our Souls Transmission".

[4] Nisha Sharma, Somen Acharya, Kaushal Kumar, Narendra Singh, and O.P. Chaurasia "Design and Implementation of a communications navigation Technology for the Visually Impaired".

[5] Sristi Agarwal, Sakshi Vaval, Varsha Chawla, Ms Nupur Agarwal, Ms Kavita Namdev "Smart alert based on IoT and Android".

[6] Nisha Sharma, Somen Acharya, Kaushal Kumar, Narendra Singh, and O.P. Chaurasia "Smart Walking Tracking System based on ArtificialIntelligence".