

NEXT-GEN INTELLIGENT HEALTH NETWORK FOR REAL TIME MONITORING AND AUTOMATED THERAPEUTIC RESPONSE SYSTEM

Devaki R, B.Arunthathi, R.K . Josika, S.Madhan, S.Geetha

SRG Engineering college, Namkkal

ABSTRACT:

The proposed system operates as an intelligent health-monitoring and therapeutic device specifically designed for elderly individuals suffering from swollen veins or circulatory discomfort. At the heart of the system lies the ESP8266 microcontroller, which serves as the central controller for data processing and wireless communication within a Body Area Network (BAN). Two temperature sensors are strategically placed to monitor the upper and lower body temperatures of the user. These sensors continuously collect data, which is transmitted to the ESP8266 for real-time analysis. The system's core principle relies on comparing the lower body temperature with the upper body temperature. In normal conditions, the temperature gradient remains consistent; however, in cases of circulatory issues such as swollen veins, the lower body may exhibit abnormal heat accumulation. When the ESP8266 detects that the lower body temperature exceeds the upper body temperature—indicating potential swelling or inflammation—it automatically activates a vibration motor. This motor delivers localized vibration therapy to the affected region, helping to enhance blood flow, reduce pain, and alleviate discomfort without the need for user intervention. Simultaneously, an LCD display provides real-time system updates, including current temperature readings and therapy status, ensuring that the user or caregiver is always informed about the individual's physiological condition and system response. The vibration therapy acts as a non-invasive and easily deployable method to support elderly individuals, particularly those with mobility issues, by addressing symptoms related to poor circulation or varicose veins. This smart system reduces the need for continuous manual monitoring and offers a proactive therapeutic response based on physiological data, enhancing both comfort and safety. By leveraging IoT, sensor technology, and automated actuation, the system provides a practical and efficient solution for elderly healthcare, focusing on early detection and response to changes in body temperature associated with vein-related issues.

CHAPTER 1 : INTRODUCTION

1.1 GENERAL

In recent years, the advancement of healthcare technologies has opened new possibilities for improving the quality of life for elderly individuals, especially those facing chronic conditions related to poor circulation and mobility. Among these conditions, swollen veins, particularly in the lower body, are common and often lead to discomfort, pain, and reduced movement. Traditional treatment methods often rely on physical therapy, medication, or caregiver assistance, which may not always be readily accessible

or timely. This has driven the need for intelligent systems that can monitor the body in real-time and offer immediate, non-invasive responses to physiological changes. The integration of smart health monitoring systems into daily life offers a proactive approach to healthcare, where early signs of discomfort can be detected and addressed without waiting for symptoms to worsen. This concept revolves around continuous monitoring of body parameters—such as temperature distribution—and using this data to make intelligent decisions that directly benefit the user. By comparing temperature differences between upper and lower body regions, early signs of inflammation or poor blood flow can be identified. When abnormalities are detected, the system automatically initiates a vibration-based therapy to stimulate circulation and reduce swelling. Such an approach ensures that elderly individuals receive prompt relief with minimal effort or intervention. Furthermore, the system provides real-time updates and health information, enhancing awareness and confidence among users and caregivers. This fusion of automated monitoring, intelligent decision-making, and therapeutic intervention not only promotes physical well-being but also encourages independence and dignity for the elderly. As the global population continues to age, the development and adoption of such intelligent healthcare systems will become increasingly vital in ensuring a higher standard of living, greater self-sufficiency, and timely care tailored to individual needs. This project represents a meaningful step forward in personalized and technology-driven elderly care.

1.2 SCOPE OF THE PROJECT:

The scope of this project lies in developing an intelligent and non-invasive health monitoring and therapy system tailored for elderly individuals suffering from circulatory issues, particularly swollen veins in the lower body. By continuously analyzing body temperature patterns, the system can detect early signs of inflammation or abnormal circulation and respond autonomously through vibration therapy to provide immediate relief. This approach minimizes the need for constant caregiver supervision and enables elderly users to experience greater independence and comfort. The project has potential applications in home healthcare, elder care facilities, and rehabilitation centers, where proactive and real-time responses are crucial. It also opens avenues for integrating advanced data analytics, AI-based health prediction, and remote monitoring in the future. By combining real-time physiological data interpretation with automated therapeutic response, the project demonstrates a scalable, affordable, and user-friendly solution that enhances the quality of life for the aging population.

1.3 EXISTING SYSTEM

1. **Manual Monitoring and Therapy:** In current healthcare practices, swollen vein conditions in elderly individuals are typically identified through manual observation by caregivers or healthcare professionals, which often results in delayed detection and limited real-time response to sudden discomfort or temperature changes in the body.

2. **Conventional Medical Devices:** Existing medical devices used for circulation-related therapy are generally bulky, expensive, and require clinical settings or professional supervision. They do not provide continuous body monitoring or automatic therapeutic action based on real-time physiological data.
3. **Lack of Intelligent Automation:** Most existing systems lack intelligent automation that can analyze temperature differences between body parts and respond accordingly. They rely heavily on human intervention, making them less effective in ensuring timely and personalized care for the elderly.

1.1.1 EXISTING SYSTEM DISADVANTAGES

- **Delayed Detection and Response:** Existing systems often rely on manual observation or scheduled check-ups, leading to delayed detection of symptoms like swelling or temperature imbalance, which can worsen the condition before treatment is administered.
- **Lack of Continuous Monitoring:** Most traditional approaches do not provide continuous, real-time monitoring of the user's physiological conditions. This limits the system's ability to detect sudden or subtle changes that may indicate early signs of health issues.
- **Dependency on Caregivers or Clinical Visits:** The elderly often depend heavily on caregivers or frequent clinical visits for diagnosis and treatment, which can be inconvenient, costly, and emotionally taxing, especially for individuals living alone or in remote areas.

1.2 LITERATURE SURVEY

1. " AN ENERGY-EFFICIENT OPTICAL WIRELESS OFDMA SCHEME FOR MEDICAL BODY-AREA NETWORKS"

Authors: Md Jahid Hasan, Mohammad Ali Khalighi, Volker Jungnickel, Luis Nero Alves, Bastien Be'chadergue

Year: 2022

Description: — The transfer of health monitoring data from multiple patients using wireless body-area networks requires the use of robust, and energy and bandwidth efficient multiple-access schemes. This paper considers the frequency-division multiple access for the wireless uplink to a fixed access point when using infrared signals to collect medical data from several patients inside an emergency waiting room. The conventional optical orthogonal scheme applies Hermitian symmetry to obtain real valued signals, which implies increased computational complexity. We consider a new approach transmitting only the real part of a complex-valued signal, where no such constraint is imposed. Based on the proposed scheme, and taking into account the limited dynamic range of an infrared light-emitting diode, we study the performance of direct current biased and asymmetrically clipped schemes, and show their advantage in terms of energy efficiency and computational complexity, as compared with the conventional schemes. For instance, we show that by using asymmetric clipping, around 35 mW less transmit power is needed to achieve a bit error rate of 10⁻³ in the considered scenario. We also demonstrate the robustness of the proposed scheme against multiple access interference.

2. " O3HSC : Outsourced Online/Offline Hybrid Signcryption for Wireless Body Area Networks"

Authors: Suhui Liu, Liqun Chen, Member, IEEE, Huaqun Wang, Shihui Fu, Member, IEEE, Lin Shi

Year:

2021

Description : Wireless body area networks (WBAN) enable ubiquitous monitoring of patients, which can change the future of healthcare services overwhelmingly. As the collected data of patients usually contain sensitive information, how to collect, transfer, store and share data securely and properly has become a concerning issue. Attribute-based encryption (ABE) can achieve data confidentiality and fine-grained access control simultaneously. Identity-based ring signature (IBRS) allows patients to prove their identity without leaking any extra (private) information. However, the heavy computational burden of ABE and IBRS is intolerable for most power-limited mobile devices, which account for a large proportion of WBAN devices. This paper combines the attribute-based online/offline encryption (ABOOE) and IBRS to achieve an outsourced online/offline hybrid signcryption (O3HSC) scheme. As far as we know, this scheme is the first signcryption scheme that adopts IBRS and satisfies online/offline signcryption simultaneously. O3HSC divides the key generation and signcryption into offline and online phases to increase the throughput of the central authority and save the power resources of mobile devices, respectively. Besides, outsourced decryption and public signature verification are also realized. O3HSC achieves security under CCA and CMIA, and the performance analysis shows that O3HSC is a lightweight and applicable scheme for WBAN.

3. " ANALYSIS OF VARICOSE VEINS OF LOWER EXTREMITIES BASED ON VASCULAR ENDOTHELIAL CELL INFLAMMATION IMAGES AND MULTI-SCALE DEEP LEARNING"

Authors: RUIZONG ZHU ¹, HUIPING NIU ², NINGNING YIN ³, TIANJIAO WU

Year:

2019

Description: — The doctor determines whether there are lesions in the human body through the diagnosis of medical images, and classifies and identifies the lesions. Therefore, the automatic classification and recognition of medical images has received extensive attention. Since the inflammatory phenomenon of vascular endothelial cells is closely related to the varicose veins of the lower extremities, in order to realize the automatic classification and recognition of varicose veins of the lower extremities, this paper proposes a varicose vein recognition algorithm based on vascular endothelial cell inflammation images and multi-scale deep learning, called MSDCNN. First, we obtained images of vascular endothelial cells in patients with varicose veins of the lower extremities and normal subjects. Second, multiple convolutional layers extract multi-scale features of vascular endothelial cell images. Then, the MFM activation function is used instead of the ReLU activation function to introduce a competitive mechanism that extracts more features that are compact and reduces network layer parameters. Finally, the network uses a 3 3 convolution kernel to improve the network feature extraction capability and use the 1 1 convolution kernel for dimensionality reduction

to further streamline network parameters. The experimental results tell us that the network has the advantages of high recognition accuracy, fast running speed, few network parameters, and is suitable for small-embedded devices

4. " **VARICOSE VEINS TREATMENT USING AUTOMATED STOCKINGS**"

Authors: K.Senthamil Selvan¹, Aravind.G I Chittumothu Pavankumar¹, Kiran Bhopate², Mallae

Year:

2023

Description: — This work presents a home based automated temporary treatment for patient healthcare to be made easier. Varicose veins are veins that are twisted and bulging. Varicose veins can form near the surface of the skin (superficial veins). Varicose veins most commonly affect the veins in the legs. The proposed system consists of a wearable device with vibrating motors connected to an ESP32. Here pressure sensors are used to measure variation of blood pressure in the upper and lower region of the body and if the reading of pressure is higher in the lower region compared to the upper region, then it will activate the motor to give in the exercise to subtle pain and the block of blood in the nerves. This exercise is also given in a periodic time interval without waiting until pressure variation is observed. These automated stockings will also behave as a normal compression stocking when vibration is not applied. In simple words the new designed stocking can perform in 3 ways. The signal acquired from the pressure sensor is processed by ESP32. This in turn activates the motor(to provide periodic vibrations). Patients will be able to receive rapid temporary therapy if this rehabilitation system is implemented.

5. " **IOT BASED VARICOSE VEIN PREVENTION DEVICE**"

Authors: A. S. Arshiya Mehaajabbin^{*1}, C. Hemanathan^{*2}, J. Lekha^{*3}, G. Kannan^{*4}

*Year:*2024

Description: Varicose veins are characterized by twisted and enlarged veins resulting from weakened vein walls. Blood is carried from the heart to the body via arteries, with veins returning blood to the heart, necessitating the return of blood from the legs against gravitational forces. Damage to vein valves leads to varicose veins, a condition linked to factors such as overweight, lack of exercise, hormonal fluctuations during pregnancy, and prolonged standing. This condition can cause significant discomfort and pain. Massage and exercise are effective in improving blood flow and alleviating symptoms. This paper proposes the constant monitoring of physiological conditions using sensor technology and the alleviation of symptoms through vibration and heat therapy. The integration of SPO₂, temperature, and muscle activity allows for the measurement of physiological indicators. This real-time analysis enables symptom detection, and discomfort is alleviated and blood circulation is enhanced through controlled vibration and temperature adjustments.

1.4 PROPOSED SYSTEM

1. **Intelligent Temperature-Based Detection:** The proposed system continuously monitors and compares the temperature between the upper and lower body to detect abnormalities like swelling, enabling early detection of circulatory issues without manual intervention.
2. **Automated Vibration Therapy Activation:** When the system identifies that the lower body temperature exceeds the upper body temperature, it automatically activates a vibration motor, delivering immediate, non-invasive therapy to improve blood circulation and reduce discomfort.
3. **Real-Time Health Status Display:** The system includes a real-time display interface that shows body temperature readings and therapy status, keeping users and caregivers informed about ongoing treatments and environmental conditions without the need for external devices.

1.2.1 PROPOSED SYSTEM ADVANTAGES

- **Real-Time Monitoring and Early Detection:** The system continuously monitors body temperature variations, allowing for early detection of swelling-related issues. This helps in preventing complications by addressing symptoms before they worsen, ensuring timely therapeutic action.
- **Autonomous and Non-Invasive Therapy:** The vibration motor is automatically triggered without requiring user input, providing a safe, non-invasive therapy that enhances blood circulation. This reduces the need for medications or clinical procedures in early stages.
- **Enhanced Comfort and Independence for Elderly Users:** By automating the detection and treatment process, the system empowers elderly individuals to manage their health independently. It minimizes reliance on caregivers and promotes a more comfortable and dignified lifestyle.

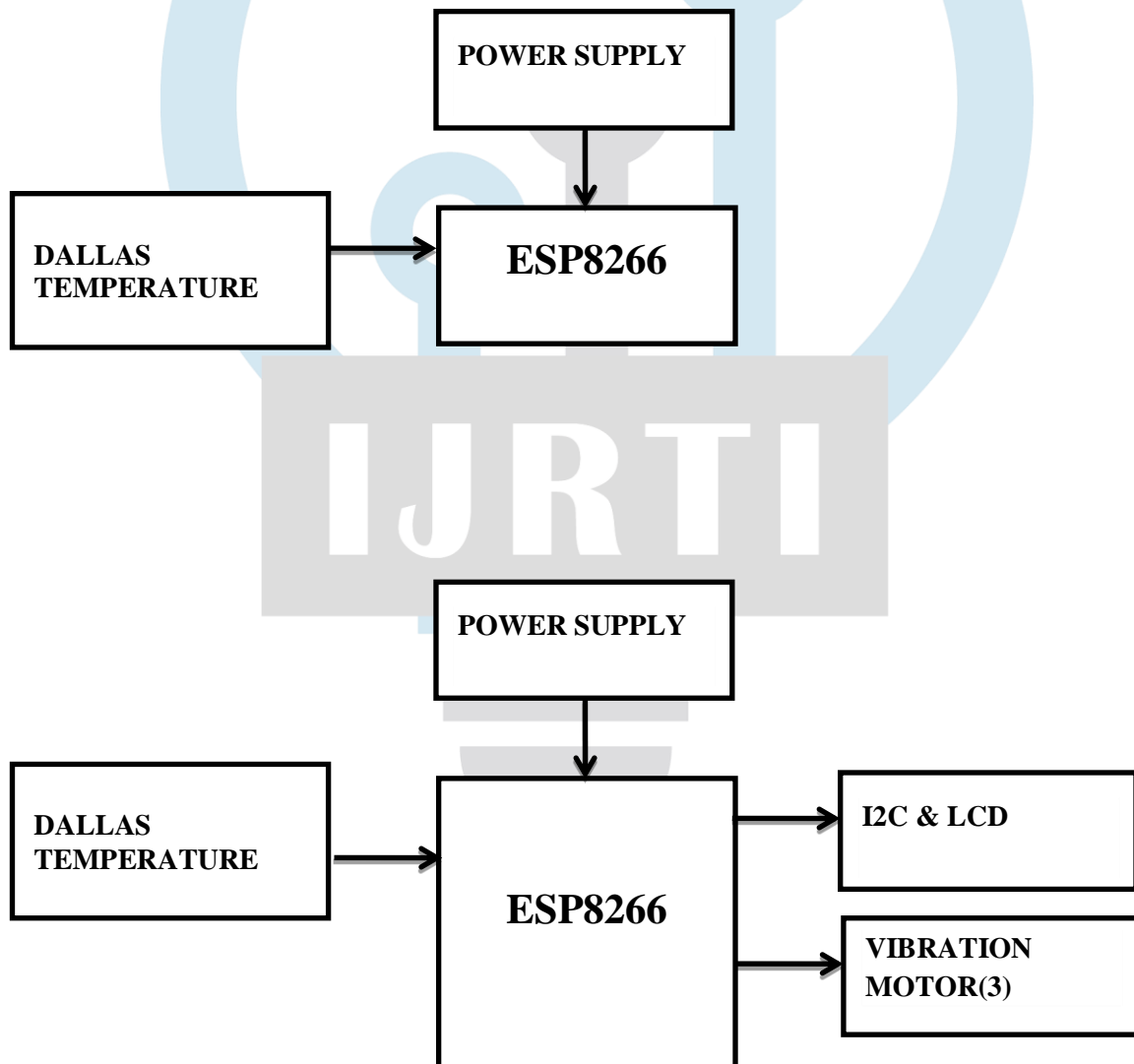
CHAPTER: 2 - PROJECT DESCRIPTION

2.1 GENERAL

The proposed project is an intelligent health monitoring and therapeutic system designed specifically to assist elderly individuals suffering from circulatory problems, particularly swollen veins in the lower body. The system aims to provide real-time physiological monitoring, early detection, and immediate therapeutic response through an automated, user-friendly solution. It works by continuously analyzing the temperature difference between the upper and lower body. In normal conditions, the body maintains a balanced temperature gradient, but when swelling or inflammation occurs, particularly due to poor circulation, the lower body temperature may rise. The system detects this abnormality and immediately activates a vibration motor, delivering targeted vibration therapy to the affected area. This therapy helps stimulate blood flow and reduce discomfort without requiring the user to take any manual action. Along with automated response, the system features a real-time display that shows current temperature readings and therapy status, ensuring that users and caregivers are always informed about ongoing conditions. The

project is developed with a focus on simplicity, automation, and comfort, making it especially useful for elderly individuals who may struggle with operating complex medical devices or accessing constant medical attention. Unlike existing systems that rely heavily on manual intervention and clinical visits, this project offers a portable, affordable, and non-invasive alternative that promotes self-care and improves the quality of life. It is highly suitable for use at home, in elder care centers, or remote locations, and serves as a valuable step toward integrating smart technology into daily health management for the aging population.

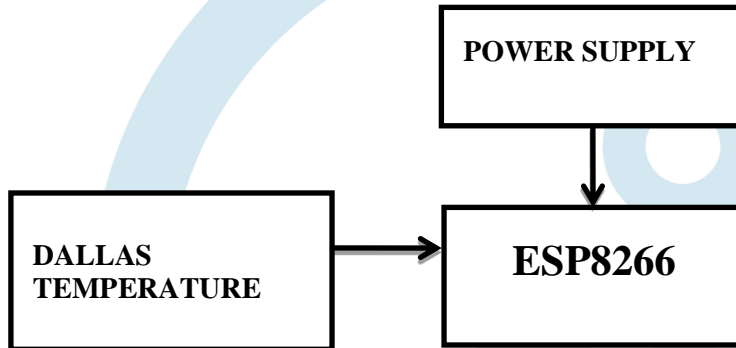
2.2 BLOCK DIAGRAM:



2.2 MODULE NAME

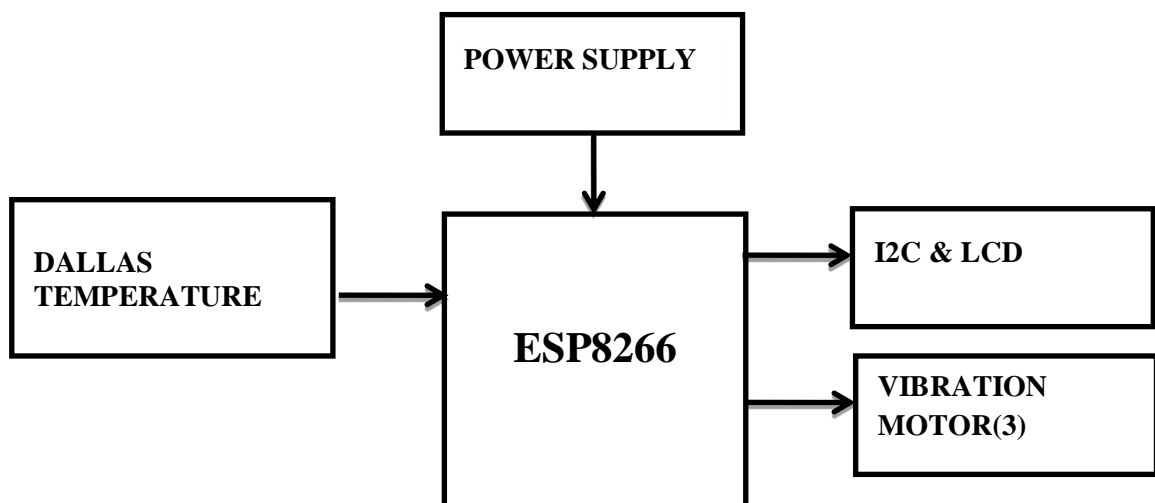
1. TEMPERATURE MONITORING AND ANALYSIS MODULE
2. VIBRATION THERAPY AND USER ALERT MODULE

TEMPERATURE MONITORING AND ANALYSIS MODULE:



This module is responsible for continuously collecting temperature data from both the upper and lower body regions. It processes the data in real-time and analyzes the temperature variations to detect abnormalities that may indicate swelling or poor circulation. The system is programmed to recognize a significant rise in lower body temperature compared to the upper body, which often signals the onset of inflammation or vein-related issues. This module ensures accurate and reliable monitoring without the need for manual measurement. It plays a critical role in identifying early warning signs and passing the analyzed data to the therapeutic module for further action.

VIBRATION THERAPY AND USER ALERT MODULE:



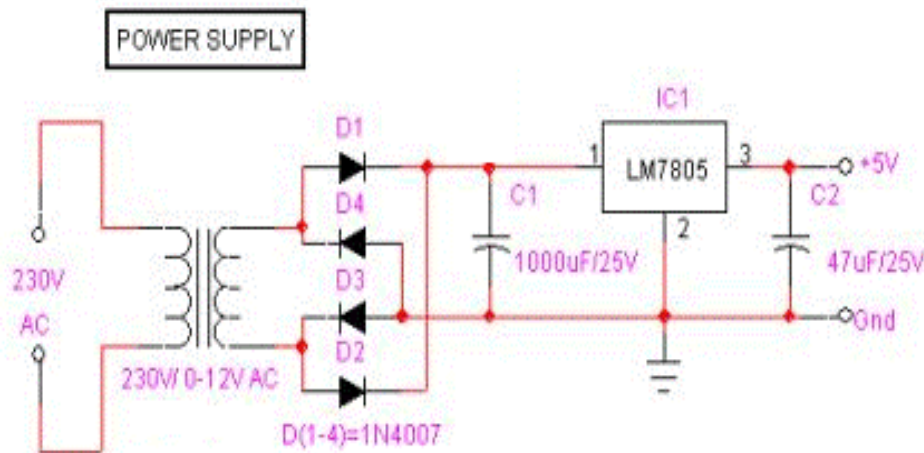
This module receives input from the monitoring module and activates the vibration motor when an abnormal temperature difference is detected. It delivers localized vibration therapy to stimulate blood circulation and reduce discomfort associated with swollen veins. In addition to initiating therapy, this module also manages the real-time user alert system by displaying temperature data and therapy status on an LCD. This keeps the user informed and reassured about their ongoing treatment. The automation within this module eliminates the need for external assistance and provides a non-invasive, user-friendly solution tailored for elderly care and improved health management.

BLOCK DIAGRAM DESCRIPTION

The proposed system operates as an intelligent health-monitoring and therapeutic device specifically designed for elderly individuals suffering from swollen veins or circulatory discomfort. At the heart of the system lies the ESP8266 microcontroller, which serves as the central controller for data processing and wireless communication within a Body Area Network (BAN). Two temperature sensors are strategically placed to monitor the upper and lower body temperatures of the user. These sensors continuously collect data, which is transmitted to the ESP8266 for real-time analysis. The system's core principle relies on comparing the lower body temperature with the upper body temperature. In normal conditions, the temperature gradient remains consistent; however, in cases of circulatory issues such as swollen veins, the lower body may exhibit abnormal heat accumulation. When the ESP8266 detects that the lower body temperature exceeds the upper body temperature—indicating potential swelling or inflammation—it automatically activates a vibration motor. This motor delivers localized vibration therapy to the affected region, helping to enhance blood flow, reduce pain, and alleviate discomfort without the need for user intervention. Simultaneously, an LCD display provides real-time system updates, including current temperature readings and therapy status, ensuring that the user or caregiver is always informed about the individual's physiological condition and system response. The vibration therapy acts as a non-invasive and easily deployable method to support elderly individuals, particularly those with mobility issues, by addressing symptoms related to poor circulation or varicose veins. This smart system reduces the need for continuous manual monitoring and offers a proactive therapeutic response based on physiological data, enhancing both comfort and safety. By leveraging IoT, sensor technology, and automated actuation, the system provides a practical and efficient solution for elderly healthcare, focusing on early detection and response to changes in body temperature associated with vein-related issues.

CHAPTER: 3 – HARDWARE AND SOFTWARE DESCRIPTION**3.1 HARDWARE DESCRIPTION:****3.1.1 POWER SUPPLY**

This section describes how to generate +5V DC power supply



The power supply section is the important one. It should deliver constant output regulated power supply for successful working of the project. A 0-12V/1 mA transformer is used for this purpose. The primary of this transformer is connected in to main supply through on/off switch& fuse for protecting from overload and short circuit protection. The secondary is connected to the diodes to convert 12V AC to 12V DC voltage. And filtered by the capacitors, which is further regulated to +5v, by using IC 7805.

INTERNET OF THINGS

The **internet of things (IoT)** is the network of physical devices, vehicles, buildings and other items embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data. In 2013 the Global Standards Initiative on Internet of Things (IoT-GSI) defined the IoT as "the infrastructure of the information society. The IoT allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, smart homes, intelligent transportation and smart cities. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure. Experts estimate that the IoT will consist of almost 50 billion objects by 2020.

INFRASTRUCTURE:

The Internet of Things will become part of the fabric of everyday life. It will become part of our overall infrastructure just like water, electricity, telephone, TV and most recently the Internet. Whereas the current Internet typically connects full-scale computers, the Internet of Things (as part of the Future Internet) will connect everyday objects with a strong integration into the physical world.

1. Plug and Play Integration

If we look at IoT-related technology available today, there is a huge heterogeneity. It is typically deployed for very specific purposes and the configuration requires significant technical knowledge and may be cumbersome. To achieve a true Internet of Things we need to move away from such small-scale, vertical application silos, towards a horizontal infrastructure on which a variety of applications can run simultaneously.

2. Infrastructure Functionality

The infrastructure needs to support applications in finding the things required. An application may run anywhere, including on the things themselves. Finding things is not limited to the start-up time of an application. Automatic adaptation is needed whenever relevant new things become available, things become unavailable or the status of things changes. The infrastructure has to support the monitoring of such changes and the adaptation that is required as a result of the changes.

3. Physical Location and Position

As the Internet of Things is strongly rooted in the physical world, the notion of physical location and position are very important, especially for finding things, but also for deriving knowledge. Therefore, the infrastructure has to support finding things according to location (e.g. geo-location based discovery). Taking mobility into account, localization technologies will play an important role for the Internet of Things and may become embedded into the infrastructure of the Internet of Things.

4. Security and Privacy

In addition, an infrastructure needs to provide support for security and privacy functions including identification, confidentiality, integrity, non-repudiation authentication and authorization. Here the heterogeneity and the need for interoperability among different ICT systems deployed in the infrastructure and the resource limitations of IoT devices (e.g., Nano sensors) have to be taken into account.

Data Management

Data management is a crucial aspect in the Internet of Things. When considering a world of objects interconnected and constantly exchanging all types of information, the volume of the generated data and the processes involved in the handling of those data become critical. A long-term opportunity for wireless communications chip makers is the rise of Machine-to-Machine (M2M) computing, which one of the enabling technologies for Internet of Things. This technology spans a broad range of applications. While there is consensus that M2M is a promising pocket of growth, analyst estimates on the size of the opportunity diverge by a factor of four [16]. Conservative estimates assume roughly 80 million to 90 million M2M units will be sold in 2014, whereas more optimistic projections forecast sales of 300 million units. Based on historical analyses of adoption curves for similar disruptive technologies, such as portable

MP3 players and antilock braking systems for cars, it is believed that unit sales in M2M could rise by as much as a factor of ten over the next five years, see Figure 2.29 [16]. There are many technologies and factors involved in the “data management” within the IoT context. Some of the most relevant concepts which enable us to understand the challenges and opportunities of data management are:

- Data Collection and Analysis
- Big Data
- Semantic Sensor Networking
- Virtual Sensors
- Complex Event Processing.

APPLICATION AREAS:

In the last few years the evolution of markets and applications, and therefore their economic potential and their impact in addressing societal trends and challenges for the next decades has changed dramatically. Societal trends are grouped as: health and wellness, transport and mobility, security and safety, energy and environment, communication and e-society, as presented in Figure 2.15. These trends create significant opportunities in the markets of consumer electronics, automotive electronics, medical applications, communication, etc. The applications in these areas benefit directly by the More-Moore and More-than-Moore semiconductor technologies, communications, networks, and software developments.

a) Cities

Smart Parking: Monitoring of parking spaces availability in the city.

Structural health: Monitoring of vibrations and material conditions in buildings, bridges and historical monuments.

Noise Urban Maps: Sound monitoring in bar areas and centric zones in real time.

Traffic Congestion: Monitoring of vehicles and pedestrian levels to optimize driving and walking routes.

Smart Lightning: Intelligent and weather adaptive lighting in street lights.

Waste Management: Detection of rubbish levels in containers to optimize the trash collection routes.

Intelligent Transportation Systems: Smart Roads and Intelligent Highways with warning messages and diversions according to climate conditions and unexpected events like accidents or traffic jams.

b) Environment

Forest Fire Detection: Monitoring of combustion gases and preemptive fire conditions to define alert zones.

Air Pollution: Control of CO₂ emissions of factories, pollution emitted by cars and toxic gases generated in farms.

Landslide and Avalanche Prevention: Monitoring of soil moisture, vibrations and earth density to detect dangerous patterns in land conditions.

Earthquake Early Detection: Distributed control in specific places of tremors.

c) Water

Water Quality: Study of water suitability in rivers and the sea for fauna and Eligibility for drinkable use.

Water Leakages: Detection of liquid presence outside tanks and pressure variations along pipes.

River Floods: Monitoring of water level variations in rivers, dams and reservoirs.

d) Energy Smart Grid, Smart Metering

Smart Grid: Energy consumption monitoring and management.

Tank level: Monitoring of water, oil and gas levels in storage tanks and cisterns.

Photovoltaic Installations: Monitoring and optimization of performance in solar energy plants.

Water Flow: Measurement of water pressure in water transportation systems.

Silos Stock Calculation: Measurement of emptiness level and weight of the goods.

e) Security & Emergencies

Perimeter Access Control: Access control to restricted areas and detection of people in non-authorized areas.

Liquid Presence: Liquid detection in data centres, warehouses and sensitive building grounds to prevent break downs and corrosion.

Radiation Levels: Distributed measurement of radiation levels in nuclear power stations surroundings to generate leakage alerts.

Explosive and Hazardous Gases: Detection of gas levels and leakages in industrial environments, surroundings of chemical factories and inside mines.

f) Industrial Control

M2M Applications: Machine auto-diagnosis and assets control.

Indoor Air Quality: Monitoring of toxic gas and oxygen levels inside chemical plants to ensure workers and goods safety.

Temperature Monitoring: Control of temperature inside industrial and medical fridges with sensitive merchandise.

Ozone Presence: Monitoring of ozone levels during the drying meat process in food factories.

Indoor Location: Asset indoor location by using active (ZigBee, UWB) and passive tags (RFID/NFC).

Vehicle Auto-diagnosis: Information collection from CAN Bus to send real time alarms to emergencies or provide advice to drivers.

g) Agriculture

Wine Quality Enhancing: Monitoring soil moisture and trunk diameter in vineyards to control the amount of sugar in grapes and grapevine health.

Green Houses: Control micro-climate conditions to maximize the production of fruits and vegetables and its quality.

Golf Courses: Selective irrigation in dry zones to reduce the water resources required in the green.

Meteorological Station Network: Study of weather conditions in fields to forecast ice formation, rain, drought, snow or wind changes.

Compost: Control of humidity and temperature levels in alfalfa, hay, straw, etc. to prevent fungus and other microbial contaminants.

h) Domestic & Home Automation

Energy and Water Use: Energy and water supply consumption monitoring to obtain advice on how to save cost and resources.

Remote Control Appliances: Switching on and off remotely appliances to avoid accidents and save energy.

Intrusion Detection Systems: Detection of window and door openings and violations to prevent intruders.

Art and Goods Preservation: Monitoring of conditions inside museums and art warehouses.

i) eHealth

Fall Detection: Assistance for elderly or disabled people living independent.

Medical Fridges: Control of conditions inside freezers storing vaccines, medicines and organic elements.

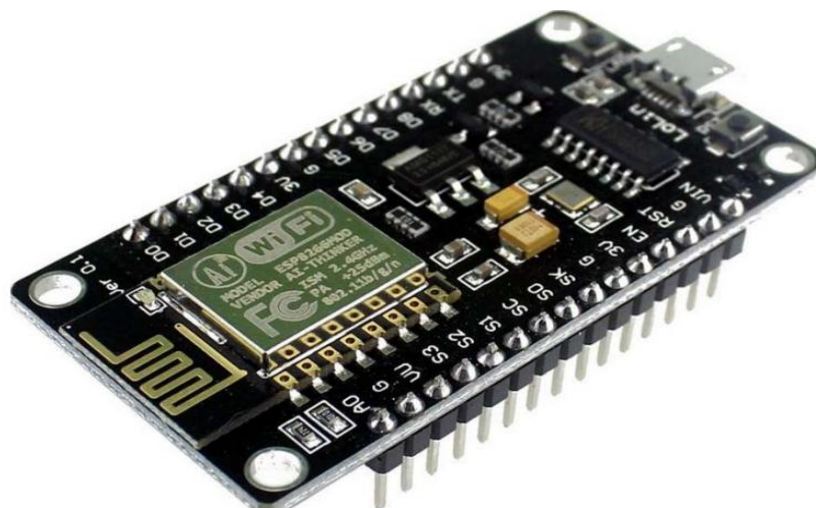
Sportsmen Care: Vital signs monitoring in high performance centres and fields.

Patients Surveillance: Monitoring of conditions of patients inside hospitals and in old people's home.

Ultraviolet Radiation: Measurement of UV sun rays to warn people not to be exposed in certain hours.

ESP-12E BASED NODEMCU

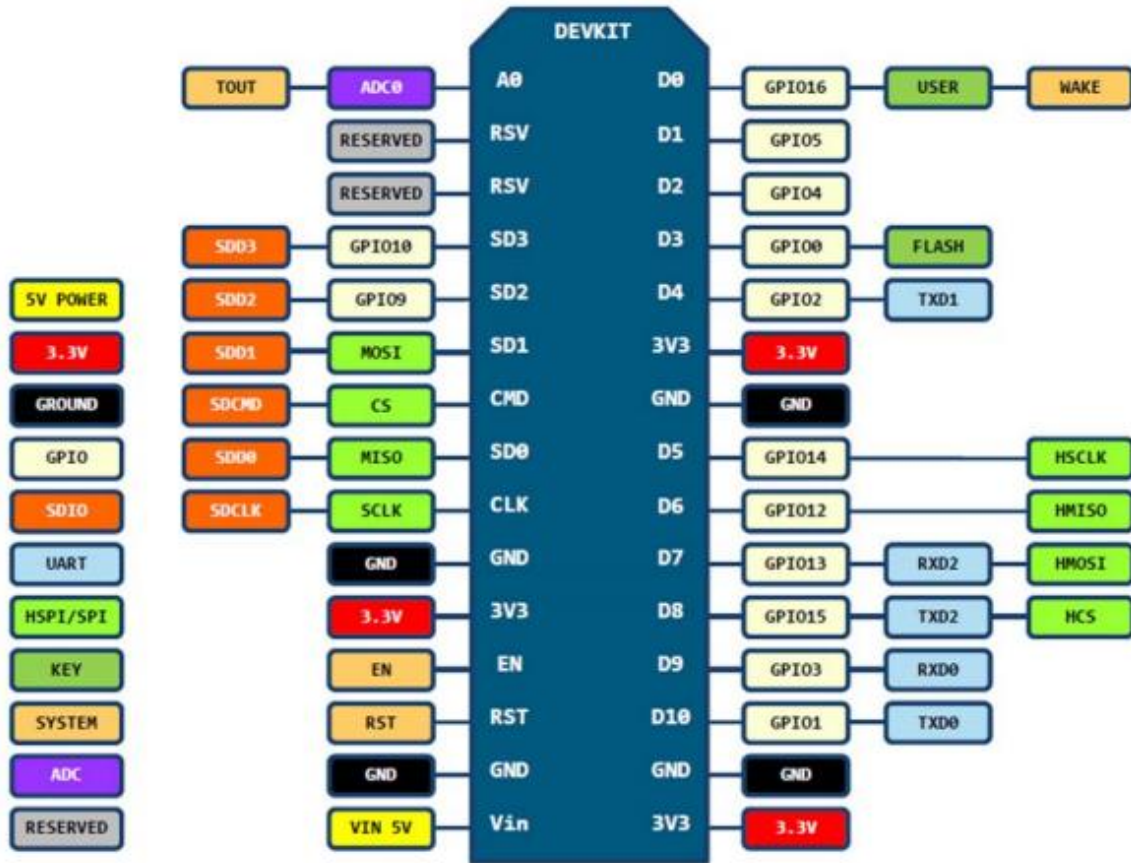
The ESP8266 is the name of a micro controller designed by Espressif Systems. The ESP8266 itself is a self-contained Wi-Fi networking solution offering as a bridge from existing micro controller to Wi-Fi and is also capable of running self-contained applications. This module comes with a built in USB connector and a rich assortment of pin-outs. With a micro USB cable, you can connect NodeMCU devkit to your laptop and flash it without any trouble, just like Arduino. It is also immediately breadboard friendly.



ESP-12E BASED NODEMCU

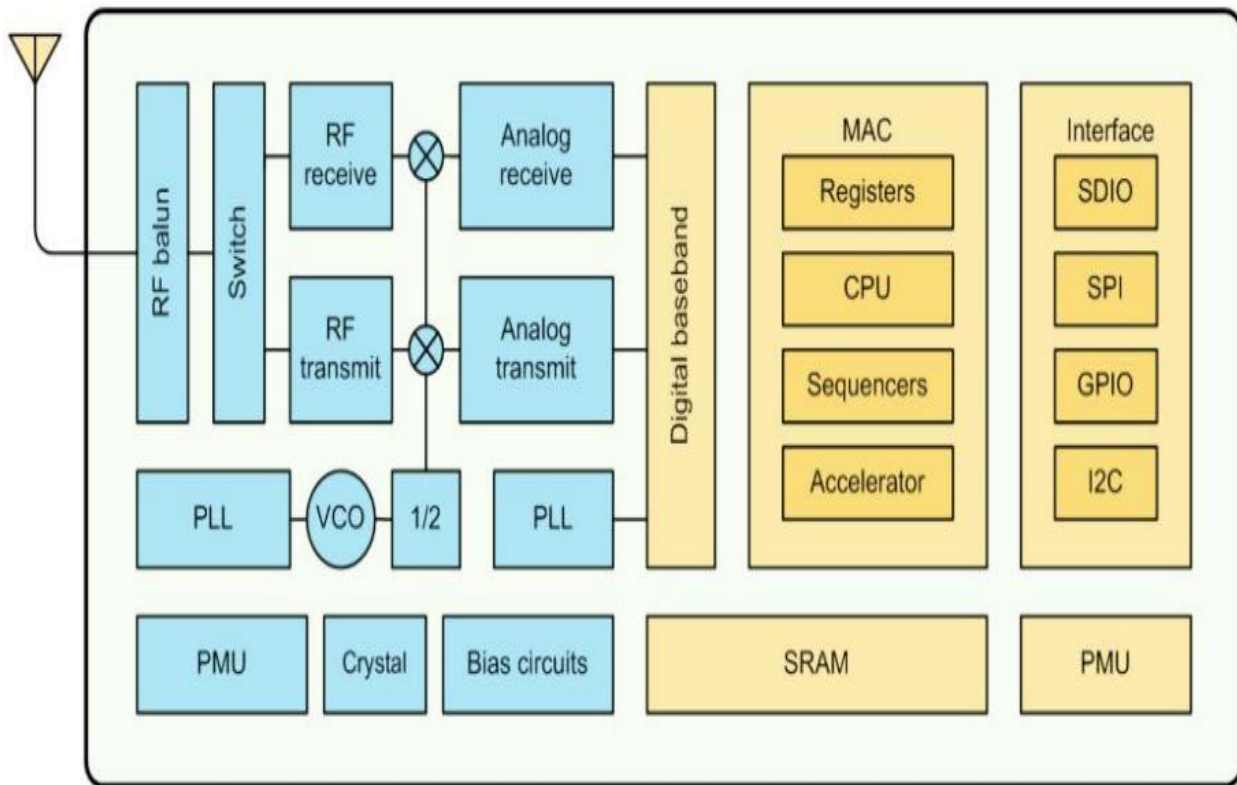
ESP-12E Wi-Fi module is developed by Ai-thinker Team. core processor ESP8266 in smaller sizes of the module encapsulates Tensilica L106 integrates industry-leading ultra-low power 32-bit MCU micro, with the 16-bit short mode, Clock speed support 80 MHz, 160 MHz, supports the RTOS, integrated Wi-Fi

MAC/BB/RF/PA/LNA, on-board antenna. The module supports standard IEEE802.11 b/g/n agreement, complete TCP/IP protocol stack. Users can use the add modules to an existing device networking, or building a separate network controller. ESP8266 is high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed Wi-Fi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.



NODEMCU PIN CONFIGURATION

ESP8266EX offers a complete and self-contained Wi-Fi networking solution; it can be used to host the application or to offload Wi-Fi networking functions from another application processor. When ESP8266EX hosts the application, it boots up directly from an external flash. It has an integrated cache to improve the performance of the system in such applications. Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any micro controller based design with simple connectivity (SPI/SDIO or I2C/UART interface). ESP8266EX is among the most integrated Wi-Fi chips in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.



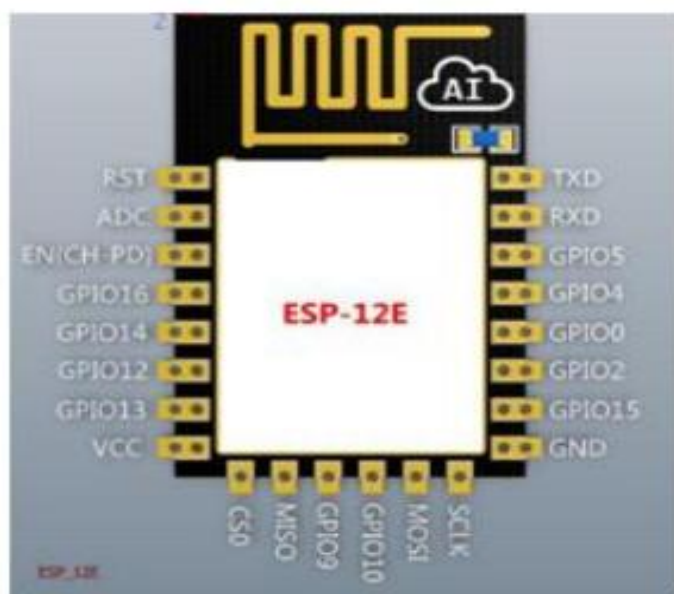
ESP-12E ARCHITECTURE

ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor, with on-chip SRAM, besides the Wi-Fi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; codes for such applications are provided in examples in the SDK. Espressif Systems' Smart Connectivity Platform (ESCP) demonstrates sophisticated system-level features include fast sleep/wake context switching for energy-efficient VoIP, adaptive radio biasing. For low-power operation, advance signal processing, and spur cancellation and radio co-existence features for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

Features

- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- Wi-Fi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes

- Support Smart Link Function for both Android and iOS devices
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IRDA, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation and 0.4s guard interval
- Deep sleep power < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20dBm output power in 802.11b mode
- Operating temperature range -40C ~ 125C



ESP-12E PIN CONFIGURATION

PIN DESCRIPTION

NO.	Pin Name	Function
1	RST	Reset the module
2	ADC	A/D Conversion result. Input voltage range 0-1v, scope: 0-1024
3	EN	Chip enable pin. Active high
4	IO16	GPIO16; can be used to wake up the chipset from deep sleep mode.
5	IO14	GPIO14; HSPI_CLK
6	IO12	GPIO12; HSPI_MISO
7	IO13	GPIO13; HSPI_MOSI; UART0_CTS
8	VCC	3.3V power supply (VDD)
9	CS0	Chip selection
10	MISO	Salve output Main input

11	IO9	GPIO9
12	IO10	GPIO10
13	MOSI	Main output slave input
14	SCLK	Clock
15	GND	GND
16	IO15	GPIO15; MTDO; HSPICS; UART0_RTS
17	IO2	GPIO2; UART1_TXD
18	IO0	GPIO0
19	IO4	GPIO4
20	IO5	GPIO5
21	RXD	UART0_RXD; GPIO3
22	TXD	UART0_TXD; GPIO1

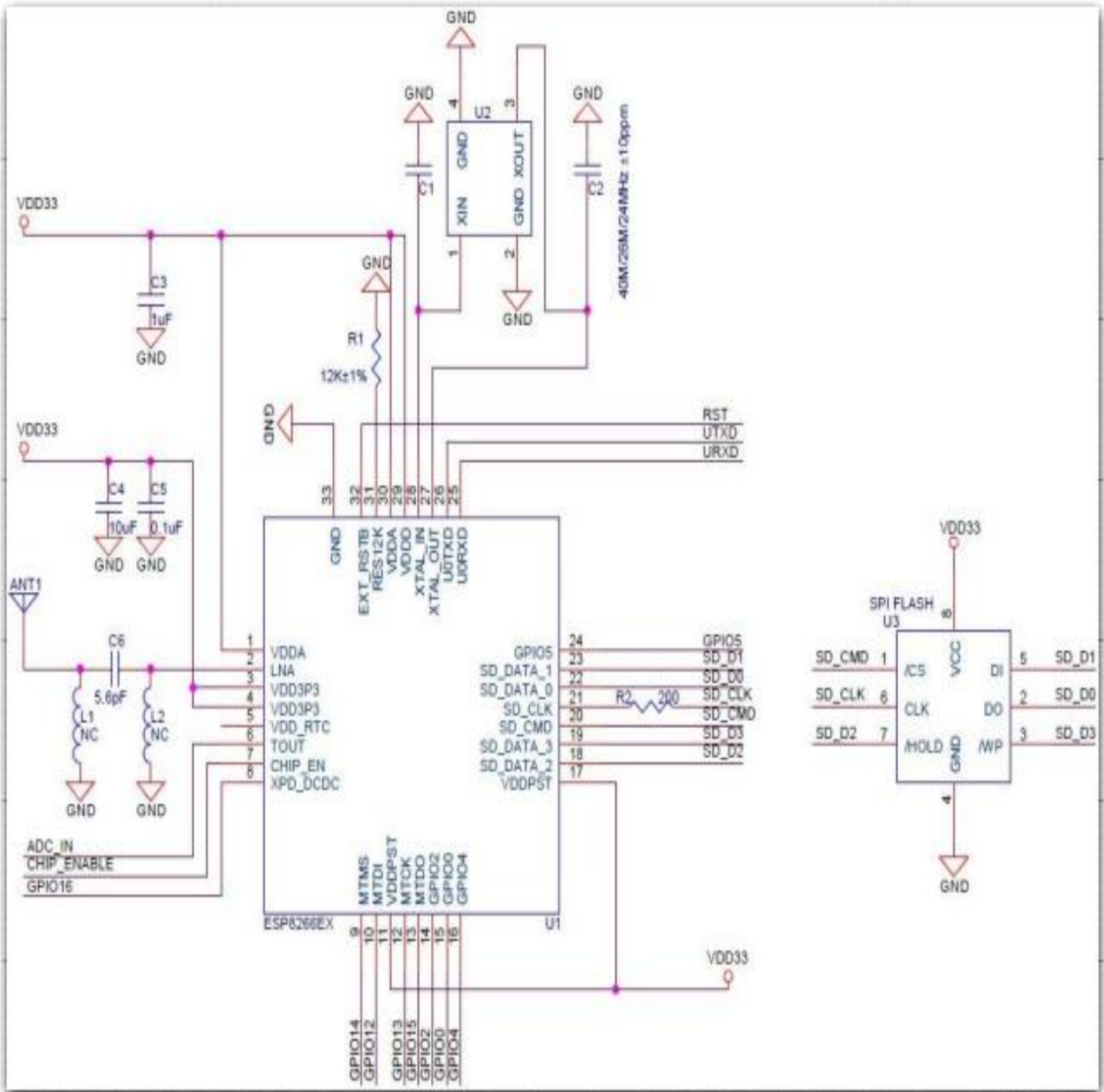
PIN MODE

Mode	GPIO15	GPIO0	GPIO2
UART	Low	Low	High
Flash Boot	Low	High	High

Receiver Sensitivity

Parameters	Min	Typical	Max	Unit
Input frequency	2412		2484	MHz
Input impedance		50		Ω
Input reflection			-10	dB
Output power of PA for 72.2Mbps	15.5	16.5	17.5	dBm
Output power of PA for 11b mode	19.5	20.5	21.5	dBm
Sensitivity				
DSSS, 1Mbps		-98		dBm
CCK, 11Mbps		-91		dBm
6Mbps (1/2 BPSK)		-93		dBm
54Mbps (3/4 64-QAM)		-75		dBm
HT20, MCS7 (65Mbps, 72.2Mbps)		-72		dBm
Adjacent Channel Rejection				
OFDM, 6Mbps		37		dB
OFDM, 54Mbps		21		dB
HT20, MCS0		37		dB
HT20, MCS7		20		dB

Schematic



Schematics of Esp-12E Wi-Fi Module

Functional Descriptions

MCU

ESP8266EX is embedded with Tensilica L106 32-bit micro controller (MCU), which features extra low power consumption and 16-bit RSIC. The CPU clock speed is 80MHz. It can also reach a maximum value of 160MHz. ESP8266EX is often integrated with external sensors and other specific devices through its GPIOs; codes for such applications are provided in examples in the SDK.

Memory Organization

Internal SRAM and ROM

ESP8266EX Wi-Fi SoC is embedded with memory controller, including SRAM and ROM. MCU can visit the memory units through iBus, dBus, and AHB interfaces. All memory units can be visited upon request, while a memory arbiter will decide the running sequence according to the time when these requests are received by the processor. According to our current version of SDK provided, SRAM space that is available to users is assigned as below:

- RAM size < 36kB, that is to say, when ESP8266EX is working under the station mode and is connected to the router, programmable space accessible to user in heap and data section is around 36kB.)
- There is no programmable ROM in the SoC, therefore, user program must be stored in an external SPI flash.

External SPI Flash

This module is mounted with an 4 MB external SPI flash to store user programs. If larger definable storage space is required, a SPI flash with larger memory size is preferred. Theoretically speaking, up to 16 MB memory capacity can be supported. Suggested SPI Flash memory capacity:

- OTA is disabled: the minimum flash memory that can be supported is 512 kB;
- OTA is enabled: the minimum flash memory that can be supported is 1 MB. Several SPI modes can be supported, including Standard SPI, Dual SPI, and Quad SPI.

Crystal

Currently, the frequency of crystal oscillators supported includes 40MHz, 26MHz and 24MHz. The accuracy of crystal oscillators applied should be $\pm 10\text{PPM}$, and the operating temperature range should be between -20°C and 85°C . When using the downloading tools, please remember to select the right crystal oscillator type. In circuit design, capacitors C1 and C2, which are connected to the earth, are added to the input and output terminals of the crystal oscillator respectively. The values of the two capacitors can be flexible, ranging from 6pF to 22pF, however, the specific capacitive values of C1 and C2 depend on further testing and adjustment on the overall performance of the whole circuit. Normally, the capacitive values of C1 and C2 are within 10pF if the crystal oscillator frequency is 26MHz, while the values of C1 and C2 are 10pF

Interfaces

Interface	Pin Name	Description
HSPI	IO12(MISO) IO13(MOSI) IO14(CLK) IO15(CS)	SPI Flash 2, display screen, and MCU can be connected using HSPI interface.
PWM	IO12(R) IO15(G) IO13(B)	Currently the PWM interface has four channels, but users can extend the channels according to their own needs. PWM interface can be used to control LED lights, buzzers, relays, electronic machines, and so on.
IR Remote Control	IO14(IR_T) IO5(IR_R)	The functionality of Infrared remote control interface can be implemented via software programming. NEC coding, modulation, and demodulation are used by this interface. The frequency of modulated carrier signal is 38KHz.
ADC	TOUT	ESP8266EX integrates a 10-bit analog ADC. It can be used to test the power-supply voltage of VDD3P3 (Pin3 and Pin4) and the input power voltage of TOUT (Pin 6). However, these two functions cannot be used simultaneously. This interface is typically used in sensor products.
I2C	IO14(SCL) IO2(SDA)	I2C interface can be used to connect external sensor products and display screens, etc.

Interface	Pin Name	Description
UART	UART0: TXD (U0TXD) RXD (U0RXD) IO15 (RTS) IO13 (CTS) UART1: IO2(TXD)	Devices with UART interfaces can be connected with the module. Downloading: U0TXD+U0RXD or GPIO2+U0RXD Communicating: UART0: U0TXD, U0RXD, MTDO (U0RTS), MTCK (U0CTS) Debugging: UART1_TXD (GPIO2) can be used to print debugging information. By default, UART0 will output some printed information when the device is powered on and is booting up. If this issue exerts influence on some specific applications, users can exchange the inner pins of UART when initializing, that is to say, exchange U0TXD, U0RXD with U0RTS, U0CTS.
I2S	I2S Input: IO12 (I2SI_DATA); IO13 (I2SI_BCK); IO14 (I2SI_WS); I2S Output: IO15 (I2SO_BCK); IO3 (I2SO_DATA); IO2 (I2SO_WS).	I2S interface is mainly used for collecting, processing, and transmission of audio data.

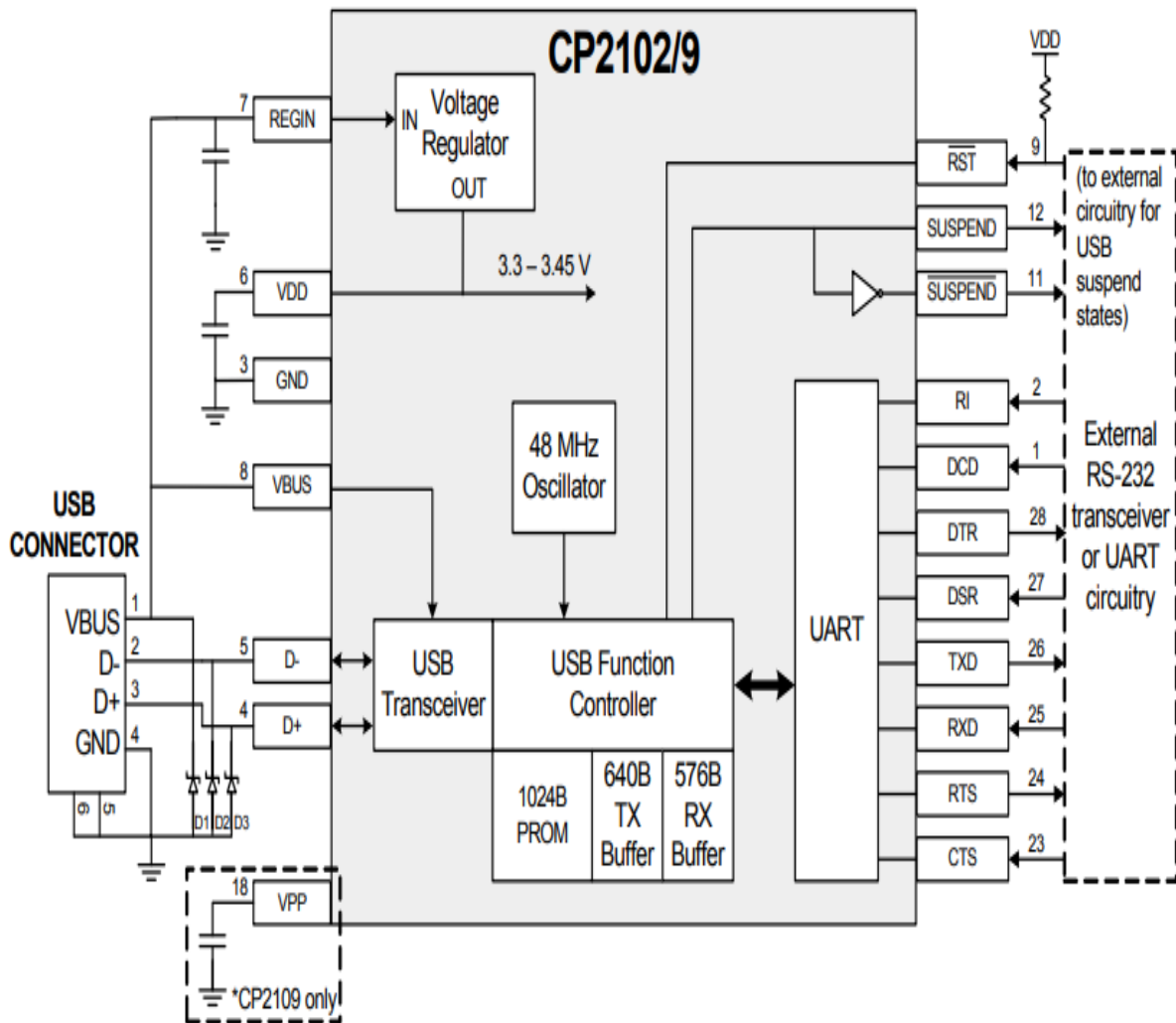
NodeMCU GPIO for Lua

The GPIO(General Purpose Input/Output) allows us to access to pins of ESP8266 , all the pins of ESP8266 accessed using the command GPIO, all the access is based on the I/O index number on the NoddMCU dev kits, not the internal GPIO pin, for example, the pin ‘D7’ on the NodeMCU dev kit is mapped to the

internal GPIO pin 13, if you want to turn 'High' or 'Low' that particular pin you need to called the pin number '7', not the internal GPIO of the pin. When you are programming with generic ESP8266 this confusion will arise which pin needs to be called during programming, if you are using NodeMCU devkit, it has come prepared for working with Lua interpreter which can easily program by looking the pin names associated on the Lua board. If you are using generic ESP8266 device or any other vendor boards please refer to the table below to know which IO index is associated to the internal GPIO of ESP8266.

Nodemcu dev kit	ESP8266 Pin	Nodemcu dev kit	ESP8266 Pin
D0	GPIO16	D7	GPIO13
D1	GPIO5	D8	GPIO15
D2	GPIO4	D9	GPIO3
D3	GPIO0	D10	GPIO1
D4	GPIO2	D11	GPIO9
D5	GPIO14	D12	GPIO10
D6	GPIO12		

D0 or GPIO16 can be used only as a read and write pin, no other options like PWM/I2C are supported by this pin. In our example in chapter 5 on blinking the blue LED, the blue LED is connected to GPIO2, it is defined as Pin4 (D4) in Lua script.

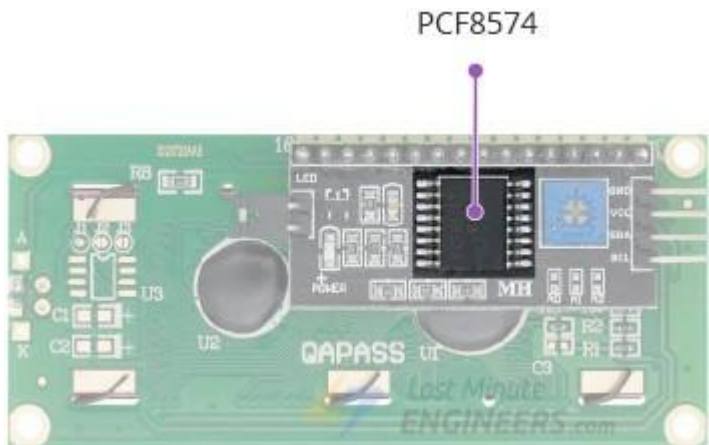
SINGLE-CHIP USB-TO-UART BRIDGE

The CP2102/9 is a highly-integrated USB-to-UART Bridge Controller providing a simple solution for updating RS232 designs to USB using a minimum of components and PCB space. The CP2102/9 includes a USB 2.0 full speed function controller, USB transceiver, oscillator, EEPROM or EPROM, and asynchronous serial data bus (UART) with full modem control signals in a compact 5 x 5 mm QFN-28 package. No other external USB components are required. The on-chip programmable ROM may be used to customize the USB Vendor ID, Product ID, Product Description String, Power Descriptor, Device Release Number, and Device Serial Number as desired for OEM applications. The programmable ROM is programmed on-board via the USB, allowing the programming step to be easily integrated into the product manufacturing and testing process. Royalty-free Virtual COM Port (VCP) device drivers provided by Silicon Laboratories allow a CP2102/9-based product to appear as a COM port to PC applications. The CP2102/9 UART interface implements all RS-232 signals, including control and handshaking signals, so existing system firmware does not need to be modified. In many existing RS-232 designs, all that is required to update the design from RS-232 to USB is to replace the RS232 level-translator with the

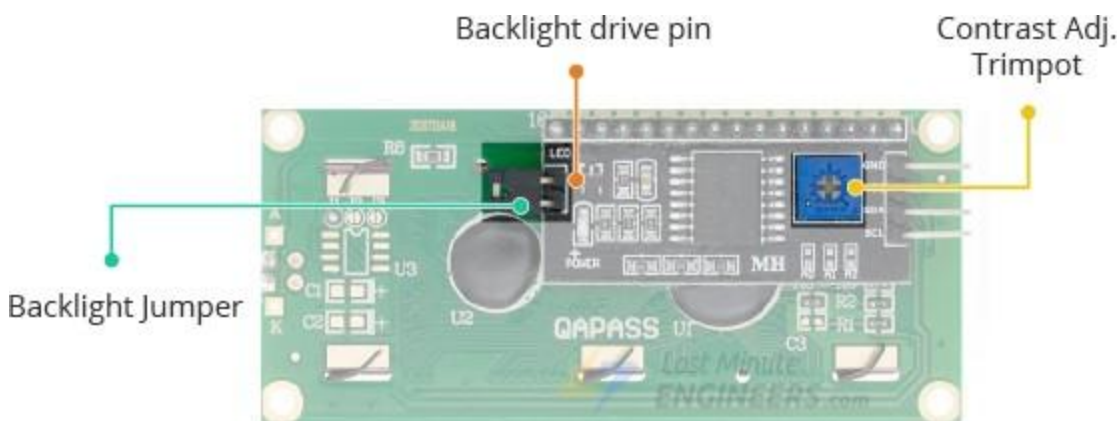
CP2102/9. Direct access driver support is available through the Silicon Laboratories USBXpress driver set.

I2C LCD Adapter

At the heart of the adapter is an 8-bit I/O expander chip – PCF8574. This chip converts the I2C data from an Arduino into the parallel data required for an LCD display.



The board also comes with a small trimpot to make fine adjustments to the display's contrast.



In addition, there is a jumper on the board that supplies power to the backlight. To control the intensity of the backlight, you can remove the jumper and apply external voltage to the header pin that is marked 'LED'.

I2C Address of LCD

If you are using multiple devices on the same I2C bus, you may need to set a different I2C address for the LCD adapter so that it does not conflict with another I2C device.

To do this, the adapter has three solder jumpers (A0, A1 and A2) or solder pads.

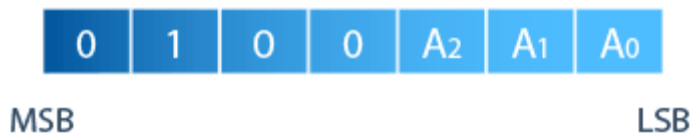


Each of these is used to hardcode the address. If a jumper is shorted with a blob of solder, it sets the address.

An important point here is that several companies manufacture the same PCF8574 chip, Texas Instruments and NXP Semiconductors, to name a few. And the I2C address of your LCD depends on the chip manufacturer.

If your LCD has Texas Instruments' PCF8574 chip:

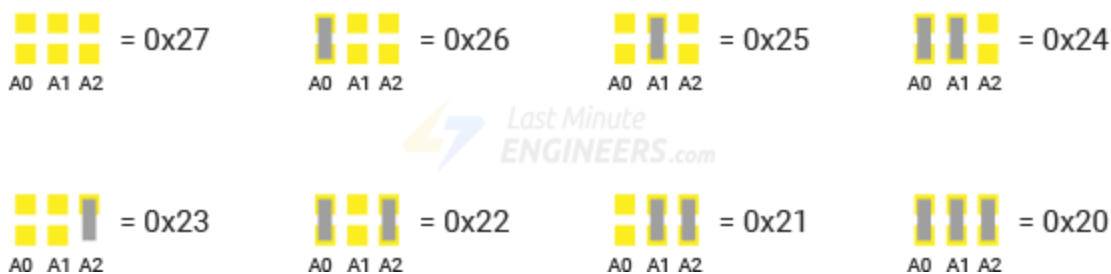
According to the Texas Instruments' datasheet, the three address selection bits (A0, A1 and A2) are placed at the end of the 7-bit I2C address register.



Since there are 3 address inputs, which can take 2 states, either HIGH/LOW, we can therefore create 8 (2^3) different combinations (addresses).

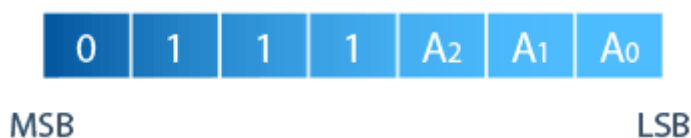
By default, all 3 address inputs are pulled HIGH using onboard pullups, giving the PCF8574 a default I2C address of 0100111_{Binary} or 0x27_{Hex}.

By shorting the solder jumpers, the address inputs are pulled LOW. If you were to short all three jumpers, the address would be 0x20. The range of all possible addresses spans from 0x20 to 0x27. Please see the illustration below.



If your LCD has NXP's PCF8574 chip:

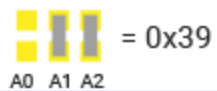
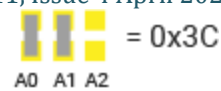
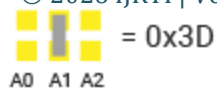
According to the NXP Semiconductors' datasheet, the three address selection bits (A0, A1 and A2) are also placed at the end of the 7-bit I2C address register. But the other bits in the address register are different.



Since there are 3 address inputs, which can take 2 states, either HIGH/LOW, we can therefore create 8 (2^3) different combinations (addresses).

By default, all 3 address inputs are pulled HIGH using onboard pullups, giving the PCF8574 a default I2C address of 0111111_{Binary} or 0x3F_{Hex}.

By shorting the solder jumpers, the address inputs are pulled LOW. If you were to short all three jumpers, the address would be 0x38. The range of all possible addresses spans from 0x38 to 0x3F. Please see the illustration below.

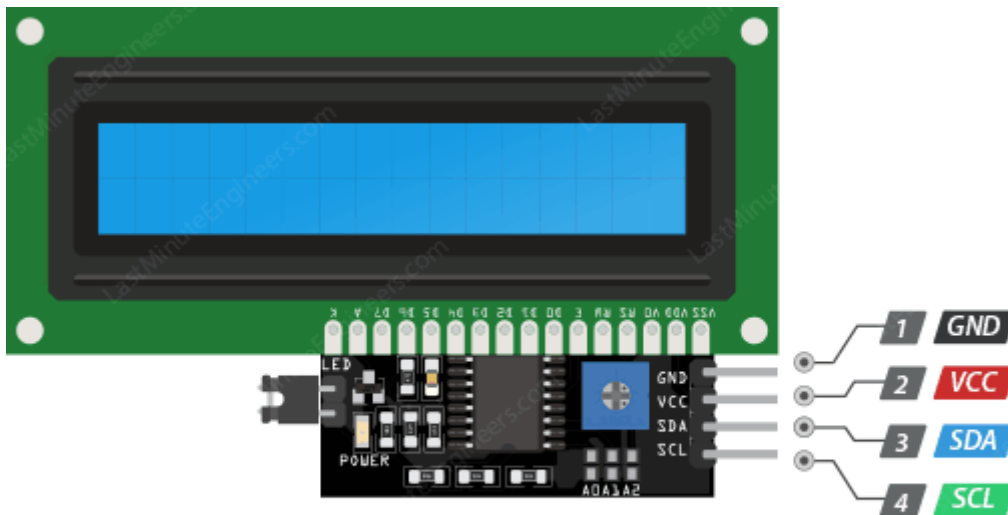


So your LCD probably has a default I2C address $0x27_{Hex}$ or $0x3F_{Hex}$. However it is recommended that you find out the actual I2C address of the LCD before using it.

Luckily there's an easy way to do this. We will see that later in the tutorial.

I2C LCD display Pinout

An I2C LCD has only 4 pins that connect it to the outside world. The connections are as follows:



I2C LCD Pinout



GND is a ground pin. Connect it to the ground of the Arduino.

VCC supplies power to the module and LCD. Connect it to the Arduino's 5V output or an external 5V power supply.

SDA is the I2C data pin. Connect it to the Arduino's I2C data pin.

SCL is the I2C clock pin. Connect it to the Arduino's I2C clock pin.

Hooking up an Arduino Uno to an I2C LCD display

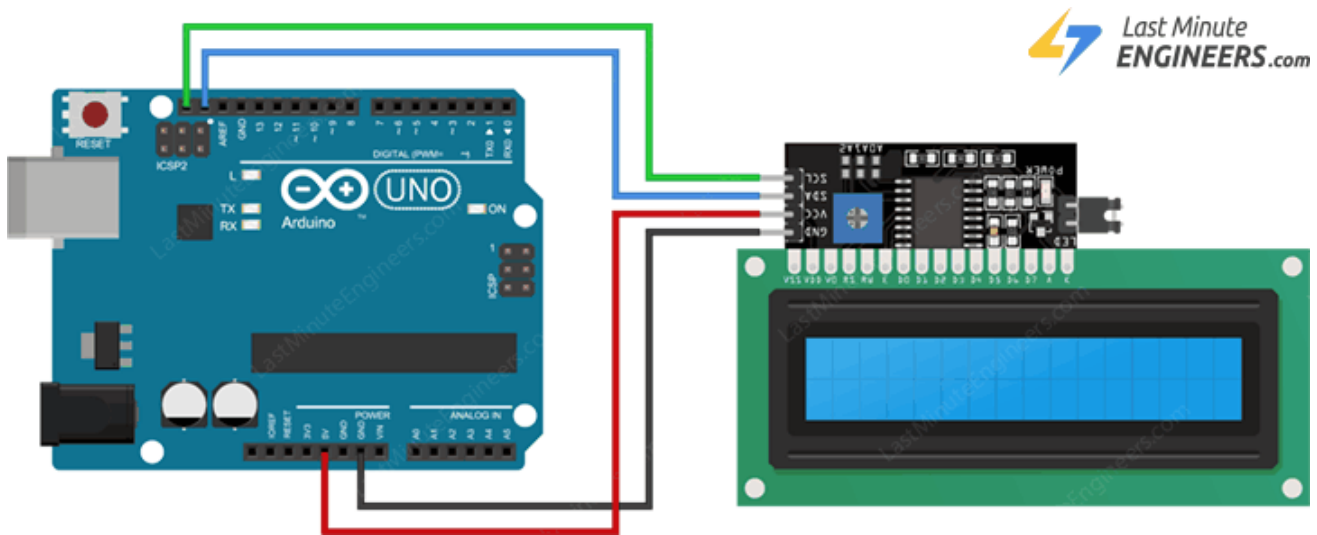
Connecting an I2C LCD is much easier than connecting a standard LCD. You only need to connect 4 pins instead of 12. Start by connecting the VCC pin to the 5V output on the Arduino and GND to ground.

Now we are left with the pins which are used for I2C communication. Note that each Arduino board has different I2C pins that must be connected accordingly. On Arduino boards with the R3 layout, the SDA (data line) and SCL (clock line) are on the pin headers close to the AREF pin. They are also known as A5 (SCL) and A4 (SDA).

If you are using a different Arduino board, please refer to the table below.

	SCL	SDA
Arduino Uno	A5	A4
Arduino Nano	A5	A4
Arduino Mega	21	20
Leonardo/Micro	3	2

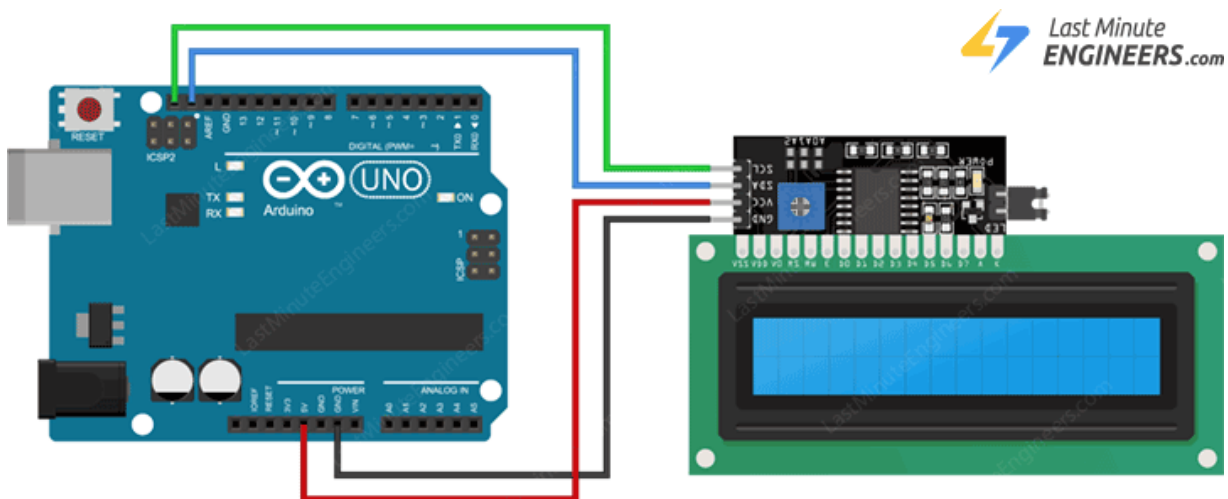
The following diagram shows you how to wire everything up.



Adjusting The LCD Contrast

After wiring up the LCD you'll need to adjust the contrast of the display. On the I2C module you will find a potentiometer that you can rotate with a small screwdriver.

Plug in the Arduino's USB connector to power the LCD. You will see the backlight lit up. Now as you turn the knob on the potentiometer, you will start to see the first row of rectangles. If that happens, Congratulations! Your LCD is working fine.

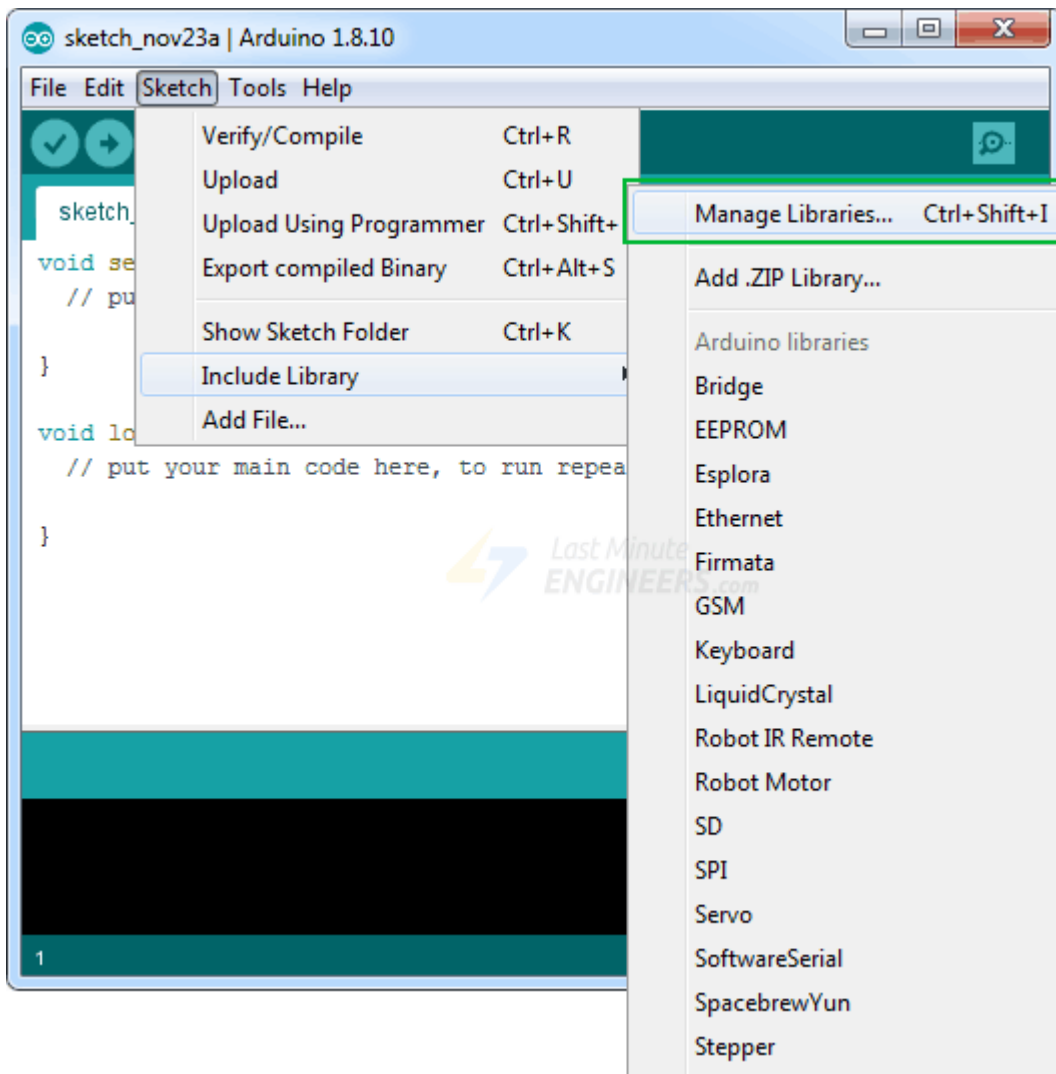


Once this is done, we can start programming the LCD.

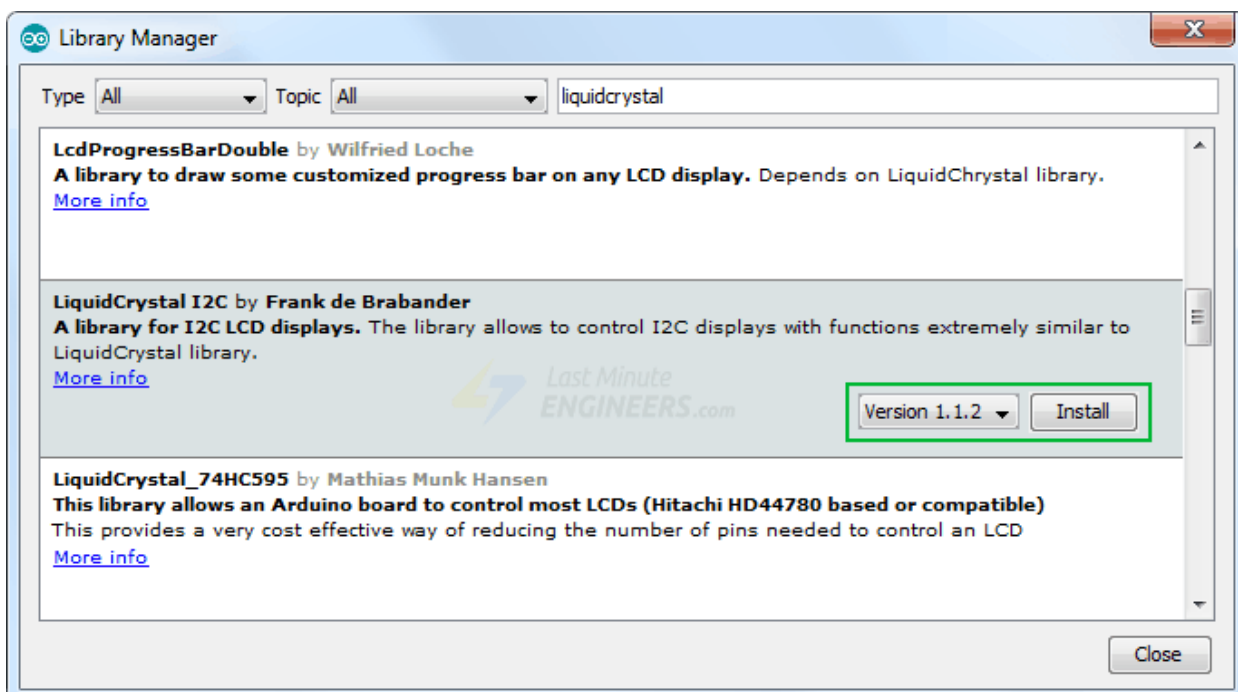
Library Installation

To drive an I2C LCD you must first install a library called LiquidCrystal_I2C. This library is an enhanced version of the LiquidCrystal library that comes with your Arduino IDE.

To install the library navigate to Sketch > Include Libraries > Manage Libraries... Wait for Library Manager to download the library index and update the list of installed libraries.



Filter your search by typing 'liquidcrystal'. There should be some entries. Look for the LiquidCrystal I2C library by Frank de Brabander. Click on that entry, and then select Install.



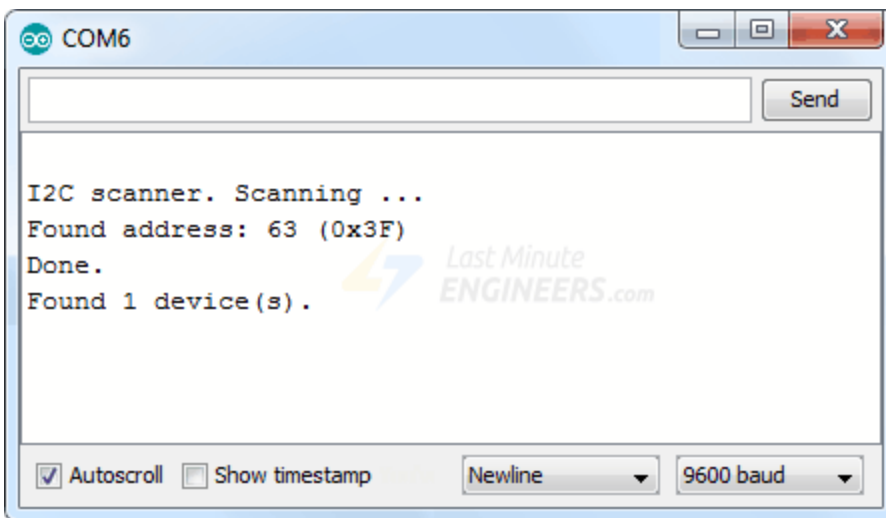
Determining the I2C Address

The I2C address of your LCD depends on the manufacturer, as mentioned earlier. If your LCD has a Texas Instruments' PCF8574 chip, its default I2C address is $0x27_{Hex}$. If your LCD has NXP Semiconductors' PCF8574 chip, its default I2C address is $0x3F_{Hex}$.

So your LCD probably has I2C address $0x27_{Hex}$ or $0x3F_{Hex}$. However it is recommended that you find out the actual I2C address of the LCD before using it. Luckily there's an easy way to do this, thanks to the Nick Gammon.

Nick wrote a simple I2C scanner sketch that scans your I2C bus and returns the address of each I2C device it finds.

Load this sketch into your Arduino and then open your Serial Monitor. You will see the I2C address of your I2C LCD display.



Please note this address. You will need it in later examples.

Basic Arduino Sketch – Hello World

The following test sketch will print 'Hello World!' on the first line of the LCD and 'LCD Tutorial' on the second line.

But, before you proceed to upload the sketch, you need to make a small change to make it work for you. You must pass the I2C address of your LCD and the dimensions of the display to the constructor of the LiquidCrystal_I2C class. If you are using a 16×2 character LCD, pass the 16 and 2; If you're using a 20×4 LCD, pass 20 and 4. You got the point!

If all goes well, you should see something like this on the display.



Code Explanation:

The sketch begins by including the LiquidCrystal_I2C library.

First of all an object of LiquidCrystal_I2C class is created. This object takes three parameters `LiquidCrystal_I2C(address, columns, rows)`. This is where you need to enter the address you found earlier, and the dimensions of the display.

```
LiquidCrystal_I2C lcd(0x3F,16,2);
```

Once you have declared a LiquidCrystal_I2C object, you can access the object methods specific to the LCD.

In 'setup' we call three functions. The first function is `init()`. It initializes the LCD object. The second function is `clear()`. This clears the LCD screen and moves the cursor to the top left corner. And third, the `backlight()` function turns on the LCD backlight.

```
lcd.init();
lcd.clear();
lcd.backlight();
```

After that we set the cursor position to the third column of the first row by calling the function `lcd.setCursor(2, 0)`. The cursor position specifies the location where you want the new text to be displayed on the LCD. The upper left corner is assumed to be `col=0, row=0`.

```
lcd.setCursor(2,0);
```

Next, the string 'Hello World!' is printed by calling the `print()` function.

```
lcd.print("Hello world!");
```

Similarly the next two lines of code set the cursor position to the third column of the second row, and print 'LCD Tutorial' on the LCD.

```
lcd.setCursor(2,1);
lcd.print("LCD Tutorial");
```

Other useful functions of the Library

There are some useful functions you can use with `LiquidCrystal_I2C` objects. Some of them are listed below:

- `lcd.home()` function is used to position the cursor in the upper-left of the LCD without clearing the display.
- `lcd.blink()` function displays a blinking block of 5×8 pixels at the position at which the next character is to be written.
- `lcd.cursor()` displays an underscore (line) at the position at which the next character is to be written.
- `lcd.noBlink()` function turns off the blinking LCD cursor.
- `lcd.noCursor()` hides the LCD cursor.
- `lcd.scrollDisplayRight()` function scrolls the contents of the display one space to the right. If you want the text to scroll continuously, you have to use this function inside a for loop.
- `lcd.scrollDisplayLeft()` function scrolls the contents of the display one space to the left. Similar to above function, use this inside a for loop for continuous scrolling.

Create and Display Custom Characters

If you find the characters on the display dull and boring, you can create your own custom characters (glyphs) and symbols for your LCD. They are extremely useful when you want to display a character that is not part of the standard ASCII character set.

As discussed earlier in this tutorial a character is made up of a 5×8 pixel matrix, so you need to define your custom character within that matrix. You can use the `createChar()` function to define a character.

To use `createChar()` you first set up an array of 8 bytes. Each byte in the array represents a row of characters in a 5×8 matrix. Whereas, 0 and 1 in a byte indicate which pixel in the row should be ON and which should be OFF.

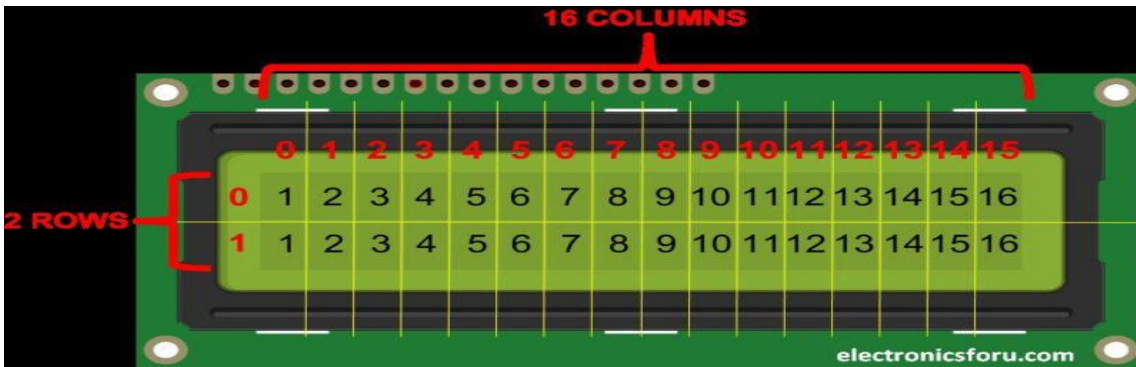
All these user defined characters are stored in the CGRAM of the LCD.

LIQUID CRYSTAL DISPLAY:

LCD screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

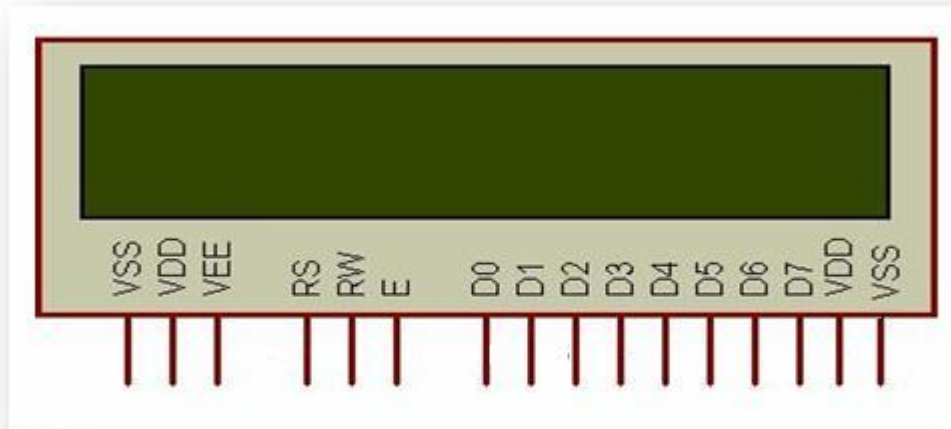
A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor

position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a LCD.



We come across LCD displays everywhere around us. Computers, calculators, television sets, mobile phones, digital watches use some kind of display to display the time. An LCD is an electronic display module which uses liquid crystal to produce a visible image. The 16x2 LCD display is a very basic module commonly used in projects. The 16x2 translates to a display 16 characters per line in 2 such lines. In this LCD each character is displayed in a 5x7 pixel matrix.

16X2 LCD PINOUT DIAGRAM:



PIN NO.	FUNCTION	NAME
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	VCC
3	Contrast adjustment; the best way is to use a variable resistor such as a potentiometer. The output of the potentiometer is connected to this pin.	Vo / VEE

	Rotate the potentiometer knob forward and backwards to adjust the LCD contrast.	
4	Selects command register when low, and data register when high	RS (Register Select)
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given; Extra voltage push is required to execute the instruction and EN(enable) signal is used for this purpose. Usually, we make it en=0 and when we want to execute the instruction we make it high en=1 for some milliseconds. After this we again make it ground that is, en=0.	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight VCC (5V)	Led+
16	Backlight Ground (0V)	Led-

RS (REGISTER SELECT):

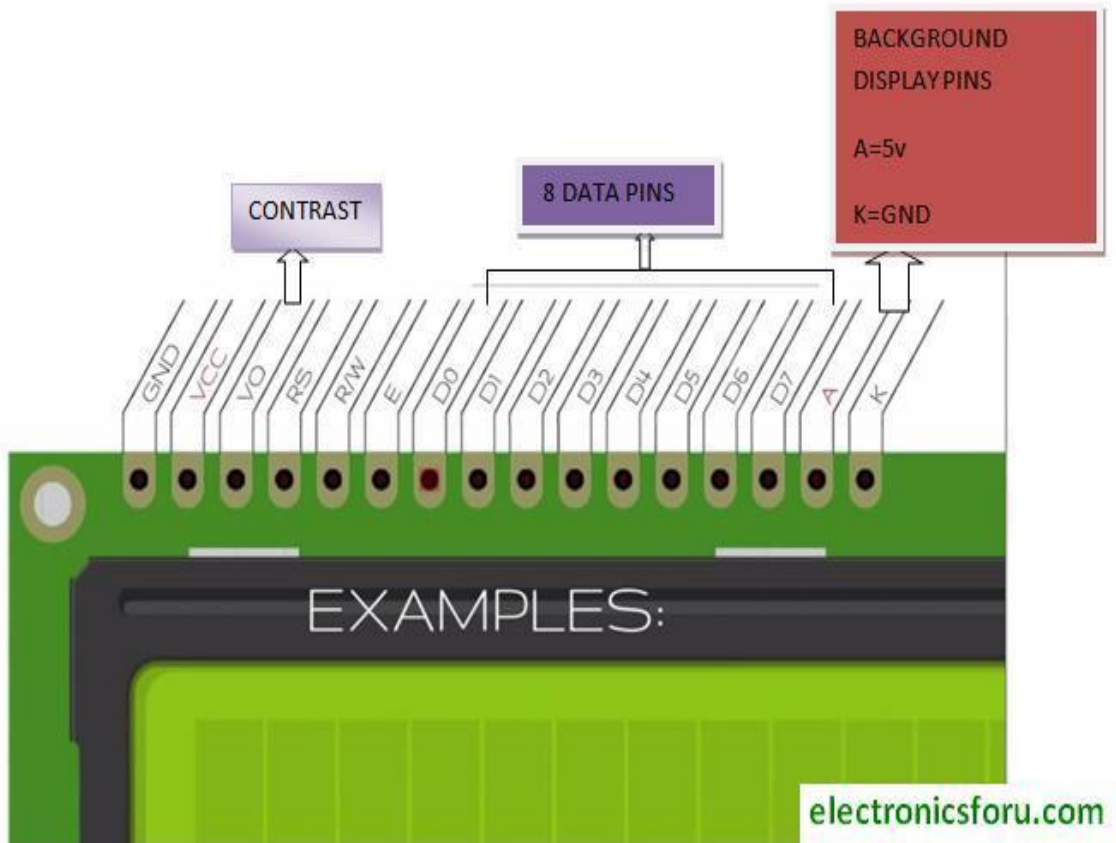
A 16X2 LCD has two registers, namely, command and data. The register select is used to switch from one register to other. RS=0 for command register, whereas RS=1 for data register.

COMMAND REGISTER:

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. Processing for commands happens in the command register.

DATA REGISTER:

The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. When we send data to LCD it goes to the data register and is processed there. When RS=1, data register is selected.

**IMPORTANT COMMAND CODES FOR LCD:**

SR.NO.	HEX CODE	COMMAND TO LCD REGISTER	INSTRUCTION
1	01	Clear display screen	
2	02	Return home	
3	04	Decrement cursor (shift cursor to left)	
4	06	Increment cursor (shift cursor to right)	
5	05	Shift display right	
6	07	Shift display left	

7	08	Display off, cursor off
8	0A	Display off, cursor on
9	0C	Display on, cursor off
10	0E	Display on, cursor blinking
11	0F	Display on, cursor blinking
12	10	Shift cursor position to left
13	14	Shift cursor position to right
14	18	Shift the entire display to the left
15	1C	Shift the entire display to the right
16	80	Force cursor to beginning (1st line)
17	C0	Force cursor to beginning (2nd line)
18	38	2 lines and 5×7 matrix

DISPLAYING CUSTOM CHARACTERS ON 16X2 LCD:

Generating custom characters on LCD is not very hard. It requires the knowledge about custom generated random access memory (CG-RAM) of LCD and the LCD chip controller. Most LCDs contain Hitachi HD4478 controller. CG-RAM is the main component in making custom characters. It stores the custom characters once declared in the code. CG-RAM size is 64 byte providing the option of creating eight characters at a time. Each character is eight byte in size.

CG-RAM address starts from 0x40 (Hexadecimal) or 64 in decimal. We can generate custom characters at these addresses. Once we generate our characters at these addresses, now we can print them on the LCD at any time by just sending simple commands to the LCD. Character addresses and printing commands are below.

In the table above you can see starting addresses for each character with their printing commands. The first character is generated at address 0x40 to 0x47 and is printed on LCD by just sending simple command 0 to the LCD. The second character is generated at address 0x48 to 0x55 and is printed by sending 1 to LCD.

CG-RAM Characters	CG-RAM Address (Hexadecimal)	Commands to display Generated Characters
1 st Character	0x40	0
2 nd Character	0x48	1
3 rd Character	0x56	2
4 th Character	0x64	3
5 th Character	0x72	4
6 th Character	0x80	5
7 th Character	0x88	6
8 th Character	0x96	7

electronicsforu.com

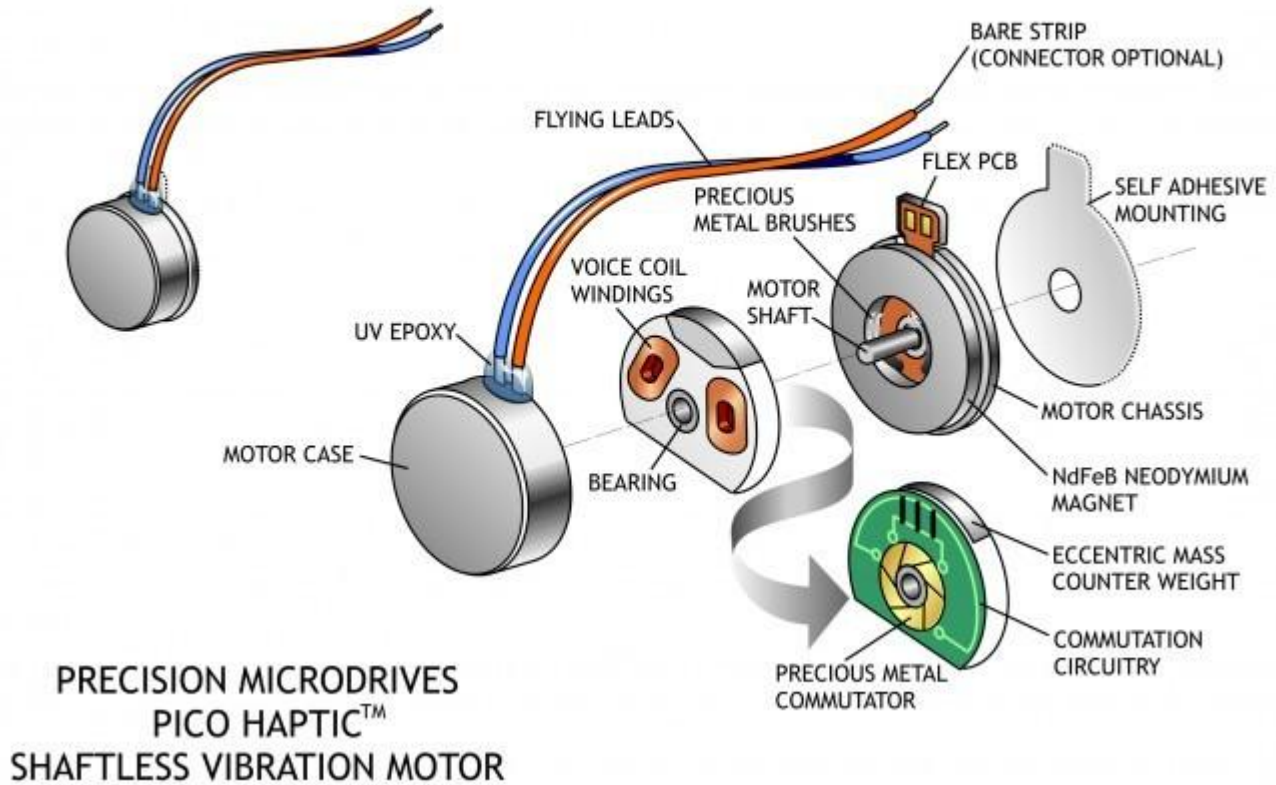


Coin Vibration Motors

Precision Microdrives currently produces coin vibration motors, also known as shaftless or pancake vibrator motors, generally in Ø8mm - Ø12mm diameters for our Pico Vibe range. Pancake motors are compact and convenient to use. They integrate into many designs because they have no external moving parts, and can be affixed in place with a strong permanent self-adhesive mounting system.

Enclosures can easily be moulded to accept the coin form of our shaftless vibration motors. Within the coin motor range, we offer both leaded and spring & pad mountable versions. Like all of our vibration motors, we are happy to quote for variations to the base design such as a modification to the lead length and also connectors.





Exploded coin motor

Applications

Due to their small size and enclosed vibration mechanism, coin vibrating motors are a popular choice for many different applications. They are great for haptics, particularly in handheld instruments where space can be at a premium:

- Mobile phones
- RFID scanners
- Industrial tools or equipment user interfaces
- Portable instruments
- Medical applications

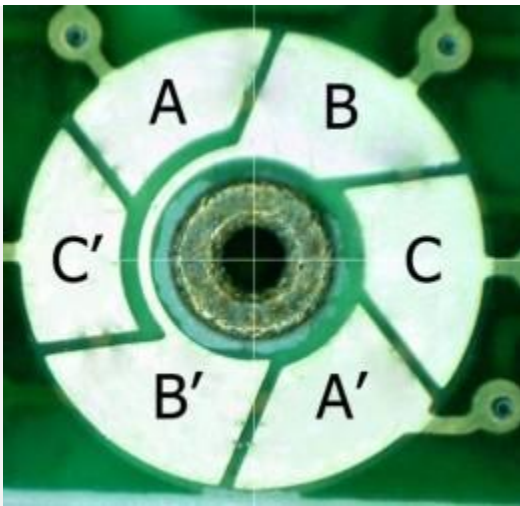
General Layout And Operation

Our coin or pancake vibrating motors are all Eccentric Rotating Mass (ERM) motors. Therefore they can be driven in the same manner as their pager motor counterparts. They have the same motor drive principles, including H-bridge circuitry for active braking.

Brushed coin vibration motors are constructed from a flat PCB on which the 3-pole commutation circuit is laid out around an internal shaft in the centre. The vibration motor rotor consists of two 'voice coils' and a small mass that is integrated into a flat plastic disc with a bearing in the middle, which sits on a shaft. Two brushes on the underside of the plastic disc make contact to the PCB commutation pads and provide power to the voice coils which generate a magnetic field. This field interacts with the flux generated by a disc magnet that is attached to the motor chassis.

The commutation circuit alternates the direction of the field through the voice coils, and this interacts with the N-S pole pairs that are built into the neodymium magnet. The disc rotates and, due to the built-in off-centred eccentric mass, the motor vibrates!

Equivalent Coin Motor Circuitry, Commutation, Terminal Resistance

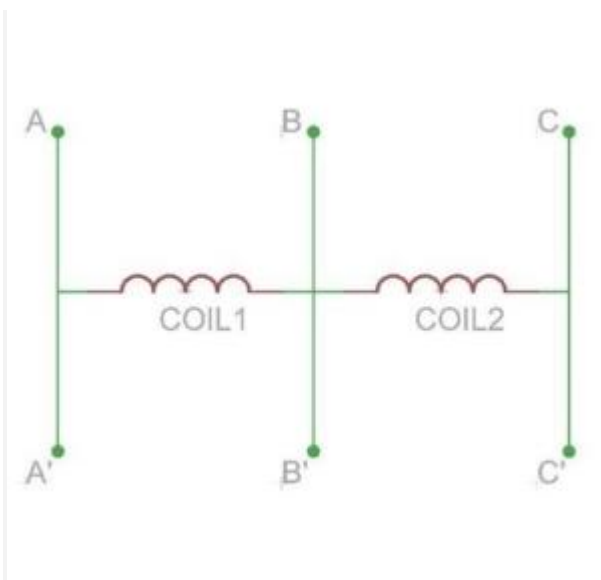


Coin Motor Commutation Circuitry

The commutator is formed by 6 segments connected to two coils. The equivalent circuit is shown on the right. The coils can be magnetized in 6 different ways, effectively making this a 6 pole machine. However, a peculiarity of this commutation design is that during one rotation the resistance through the brushes is not constant.

For a third of the revolution the brushes “see” the two coils in series instead of only one; which is why in some orientations the resistance seen by a circuit will be double and therefore the start current half the rated value.

The current figures presented in Conformity Limits Specifications sections of datasheets represent the worst case current draw; i.e. where the brushes see only one coil.



Coin Motor Equivalent Circuit

Start Voltages And Drive Signals

The full term 'Maximum Start Voltage' is the lowest voltage that you can apply to the motor and still be sure that it will start.

Coin vibration motors have a relatively high start voltage (compared to cylinder pager vibration motors) which must be considered in designs. Typically this is around 2.3v (all coin vibration motors have a nominal voltage of 3v), and failure to respect this could result in motors not starting when the application is lying in certain orientations. This problem arises because, in the vertical orientation, the coin vibrating motor must force the eccentric mass over the top of the shaft on the initial cycle.

Due to the start voltage issue, we recommend that coin type vibrating motors are switched hard on and off at a voltage above the guaranteed start voltage unless a well-tested haptics driver is used.

Mounting



Self-adhesive vibration motor mounting

Coin vibration motors are designed to be easy to mount. They come with either spring PCB connectors or a high-strength long life self-adhesive backing sheet that is pre-attached to the underside of the chassis. The adhesive allows for a secure mounting of the vibration motor to a wide range of surfaces such as PCBs or flat internal surfaces of the enclosure and makes manufacturing installation fast and clean.

Three brands of adhesive are typically used on our coin vibrator motors depending on availability (they all have very similar specifications).



8mm coin (shaftless) vibration motor

These are:

- 3M VHB 9448
- Sony 4000T
- Nitto 5000NS

These 0.16mm thick adhesive tapes typically offer a 180 deg peeling strength of 15N/20mm, and a tensile strength of around 20N/10mm. The acrylic adhesive is considered to be resistant to most solvents, UV light, moisture and temperature extremes.

As with all adhesives, the final bond strength is dependent on the cleanliness of the mating surface. It is recommended that this mating surface is clean, dry, and offers a good unified fit to the motor backing-plate (on which the self-adhesive pad is stuck).

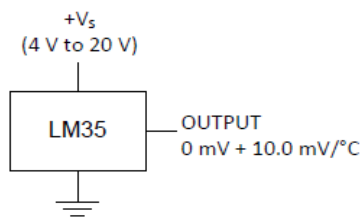
Spring PCB vibrator motors have spring-loaded fingers on the motor which mat with pads on the PCB. This makes assembly easier for applications where it's desired to have the motor mounted to the enclosure. Also, higher frequency harmonics are absorbed and reduced by the rubber 'boots' that enclose these kinds of motors.

If extra security is required, consider the moulding securing walls within the enclosure body. This technique is commonly used in mobile phones to ensure that the maximum amount of vibration is transmitted through the case.

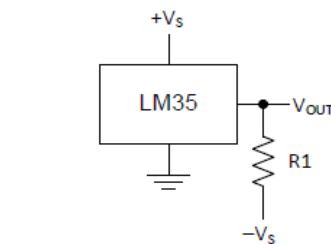
makes interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60 μA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 device is rated to operate over a -55°C to 150°C temperature range, while the LM35C device is rated for a -40°C to 110°C range (-10° with improved accuracy). The LM35-series devices are available packaged in hermetic TO transistor packages, while the LM35C, LM35CA, and LM35D devices are available in the plastic TO-92 transistor package. The LM35D device is available in an 8-lead surface-mount small-outline package and a plastic TO-220 package.

LM35 temperature sensor circuit configuration

**Basic Centigrade Temperature Sensor
(2°C to 150°C)**



Full-Range Centigrade Temperature Sensor



Choose $R_1 = -V_S / 50 \mu\text{A}$
 $V_{\text{OUT}} = 1500 \text{ mV}$ at 150°C
 $V_{\text{OUT}} = 250 \text{ mV}$ at 25°C
 $V_{\text{OUT}} = -550 \text{ mV}$ at -55°C

LM35 can be used in two circuit configurations. Both yields different results. In first configuration you can only measure the positive temperature from 2 degree Celsius to 150 degree Celsius. In this first configuration we simply power LM35 and connect the output directly to analog to digital converters. In the second configuration we can utilize all the sensor resources and can measure the full range temperature from -55°C to 150°C . This configuration is little complex but yields high results. We have to connect an external resistor in this case to switch the level of negative voltage upwards. The external resistor value can be calculated from the formula given below the configuration circuit. The second configuration circuit can be made in various ways. To see about the second configuration circuits visit the LM35 [datasheet](#) by Texas Instruments. Texas Instruments [data sheet](#) enlists the circuit with clear component values.

Although the first configuration did not need a resistor at output side but i recommend to connect an 80 k to 100 k resistor between Vout and GND pin. When i performed several experiments i noticed that the readings some time fluctuate and the Vout pin floats. So a resistor between Vout and GND tights the Vout pin low and prevents the pin from floating.

PARAMETER	VALUE
Accuracy at 25°C	±0.5°C
Accuracy from -55 °C to 150°C	±1°C
Temperature Slope	10 mV/°C

The accuracy parameters for both the configurations are different. Average accuracy level is +- 1 degree Celsius for both configurations. But the accuracy level decreases for temperature between 2 degree to 25 degree centigrade. Now that we have discussed about the lm35 temperature sensor pin out, structure, linear scale factor and accuracy level it's time to list down the steps on how to measure temperature using Lm35 temperature sensor.

Formula to convert voltage to temperature

The formula to convert voltage to centigrade temperature for Lm35 is

Centigrade Temperature = Voltage Read by ADC / 10 mV(mills Volt)

I divided by 10 mV because Linear scale factor is for Lm35 is 10mV.

Following the above steps and tutorial you can easily interface lm35 temperature sensor with any microcontroller that has a build in analog to digital converter pins. Almost all the microcontroller today have build in ADC.

Pin Configuration:

Pin Number	Pin Name	Description
1	Vcc	Input voltage is +5V for typical applications
2	Analog Out	There will be increase in 10mV for raise of every 1°C. Can range from -1V(-55°C) to 6V(150°C)
3	Ground	Connected to ground of circuit

Lm35 Working Principle (Understanding Lm35 Linear Scale Factor)

In order to understand the working principle of Lm35 temperature sensor we have to understand the linear scale factor. In the features of Lm35 it is given to be **+10 mills volt per degree centigrade**. It means that with increase in output of 10 mills volt by the sensor Vout pin the temperature value increases by one. For example if the sensor is outputting 100 mills volt at Vout pin the temperature in centigrade will be 10 degree centigrade. The same goes for the negative temperature reading. If the sensor is outputting -100 mills volt the temperature will be -10 degree Celsius.

Features

- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full -55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost Due to Wafer-Level Trimming
- Operates From 4 V to 30 V
- Less Than 60-μA Current Drain
- Low Self-Heating, 0.08°C in Still Air
- Non-Linearity Only $\pm 1/4^\circ\text{C}$ Typical
- Low-Impedance Output, 0.1 Ω for 1-mA Load
- Small and hence suitable for remote applications
- Available in TO-92, TO-220, TO-CAN and SOIC package
- Output voltage is directly proportional (Linear) to temperature (i.e.) there will be a rise of 10mV (0.01V) for every 1°C rise in temperature.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage +35V to -0.2V

Output Voltage +6V to -1.0V

Output Current 10 mA

Storage Temp.;

TO-46 Package -60°C to +180°C

TO-92 Package -60°C to +150°C

SO-8 Package -65°C to +150°C

TO-220 Package -65°C to +150°C

Lead Temp.:

TO-46 Package,

(Soldering, 10 seconds) 300°C

TO-92 and TO-220 Package,

(Soldering, 10 seconds) 260°C

SO Package

Vapor Phase (60 seconds) 215°C

Infrared (15 seconds) 220°C

ESD Susceptibility 2500V

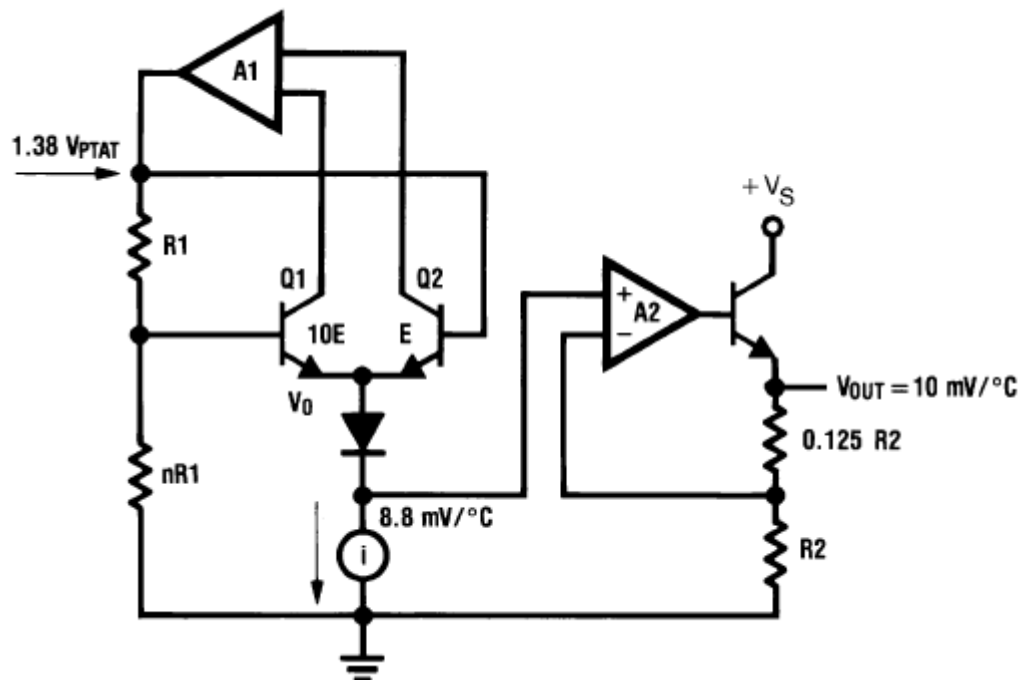
Specified Operating Temperature Range: TMIN to TMAX

LM35, LM35A -55°C to +150°C

LM35C, LM35CA -40°C to +110°C

LM35D 0°C to +100°C

Functional block diagram



The temperature-sensing element is then buffered by an amplifier and provided to the VOUT pin. The amplifier has a simple class A output stage with typical 0.5-Ω output impedance as shown in the Functional Block Diagram. Therefore the LM35 can only source current and its sinking capability is limited to 1 μA.

Applications

The LM35 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.01°C of the surface temperature. This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature. This is especially true for the TO-92 plastic package, where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature. To minimize this problem, be sure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the LM35 die's temperature will not be affected by the air temperature. The TO-46 metal package can also be soldered to a metal surface or pipe without damage. Of course, in that case the V-terminal of the circuit will be grounded to that metal. Alternatively, the LM35 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and varnishes such as Humiseal and epoxy paints or dips are often used to insure that moisture cannot corrode the LM35 or its connections. These devices are sometimes soldered to a small light-weight heat fin, to decrease the thermal time constant and speed up the response in slowly-moving air. On the other hand, a small thermal mass may be added to the sensor, to give the steadiest reading despite small deviations in the air temperature

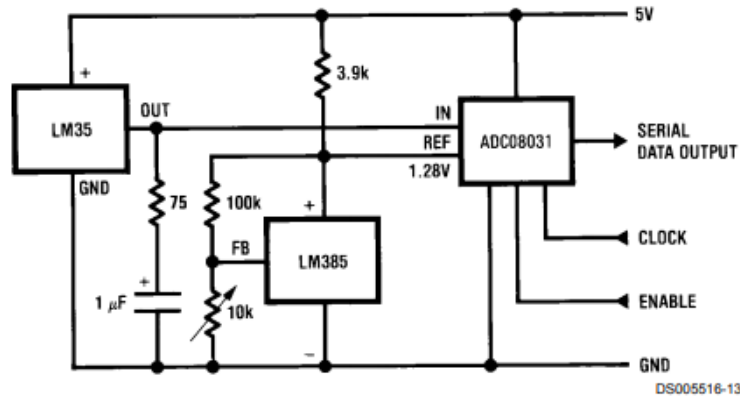


FIGURE 13. Temperature To Digital Converter (Serial Output) (+128°C Full Scale)

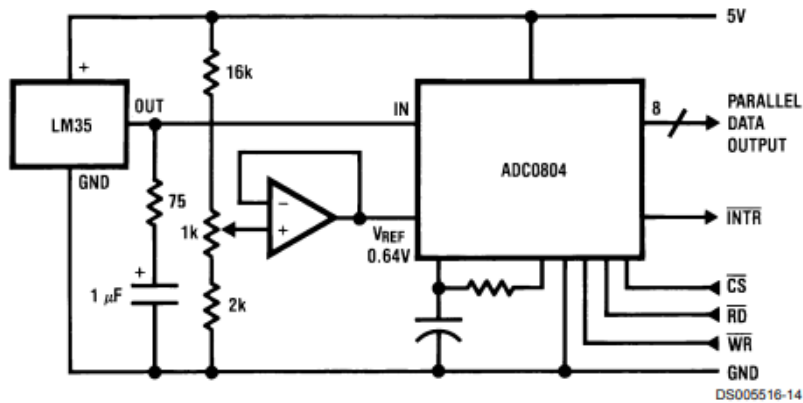


FIGURE 14. Temperature To Digital Converter (Parallel TRI-STATE™ Outputs for Standard Data Bus to μ P Interface) (128°C Full Scale)

CAPACITIVE LOADS Like most micropower circuits, the LM35 has a limited ability to drive heavy capacitive loads. The LM35 by itself is able to drive 50 pf without special precautions. If heavier loads are anticipated, it is easy to isolate or decouple the load with a resistor; see Figure 3. Or you can improve the tolerance of capacitance with a series R-C damper from output to ground; see Figure 4. When the LM35 is applied with a 200Ω load resistor as shown in Figure 5, Figure 6 or Figure 8 it is relatively immune to wiring capacitance because the capacitance forms a bypass from ground to input, not on the output. However, as with any linear circuit connected to wires in a hostile environment, its performance can be affected adversely by intense electromagnetic sources such as relays, radio transmitters, motors with arcing brushes, SCR transients, etc, as its wiring can act as a receiving antenna and its internal junctions can act as rectifiers. For best results in such cases, a bypass capacitor from VIN to ground and a series R-C damper such as 75Ω in series with 0.2 or 1 μF from output to ground are often useful. These are shown in Figure 13 and Figure 14.

LM35 Temperature Sensor Applications:

- Measuring temperature of a particular environment
- Providing thermal shutdown for a circuit/component
- Monitoring Battery Temperature
- Measuring Temperatures for HVAC applications.

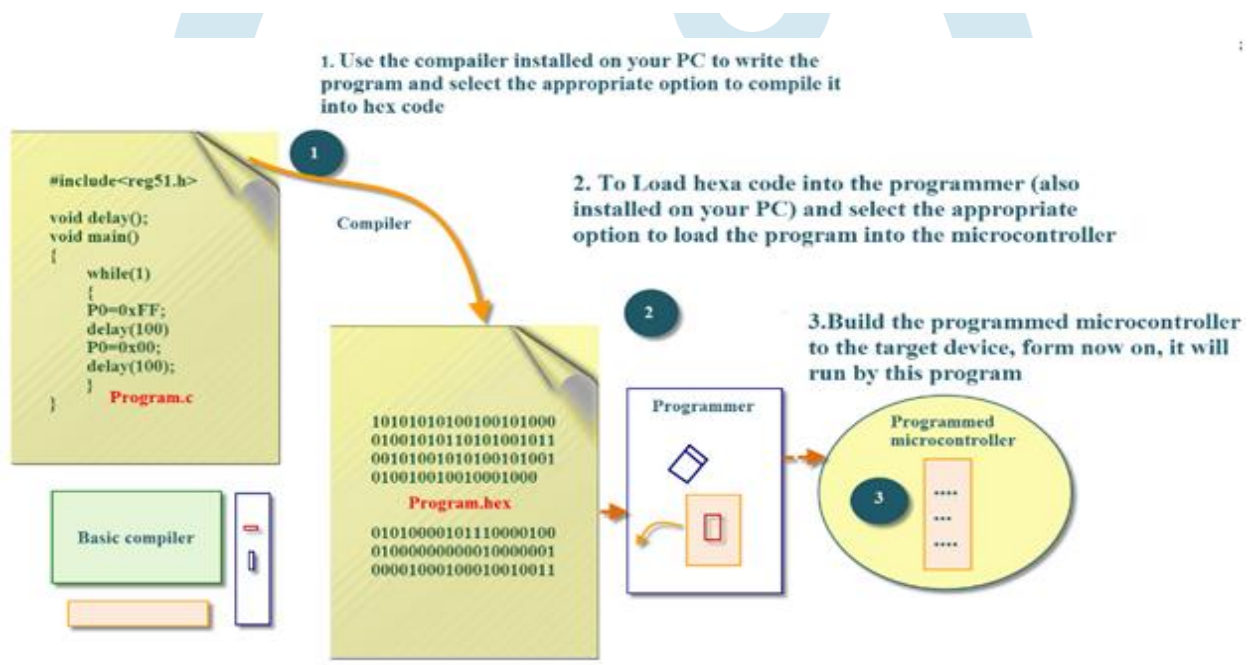
3.1. SOFTWARE REQUIREMENTS:

3.1.2 EMBEDDED C

Embedded C is most popular programming language in software field for developing electronic gadgets. Each processor used in electronic system is associated with embedded software.

Embedded C programming plays a key role in performing specific function by the processor. In day-to-day life we used many electronic devices such as mobile phone, washing machine, digital camera, etc. These all device working is based on microcontroller that are programmed by embedded C.

Let's see the block diagram representation of embedded system programming:



The Embedded C code written in above block diagram is used for blinking the LED connected with Port0 of microcontroller.

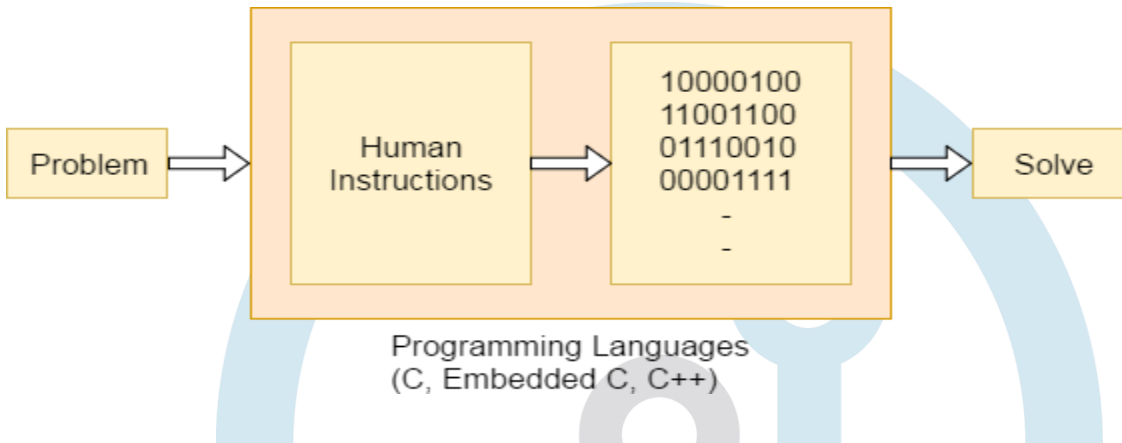
In embedded system programming C code is preferred over other language. Due to the following reasons:

- Easy to understand
- High Reliability
- Portability
- Scalability

EMBEDDED SYSTEM PROGRAMMING:

Basic Declaration

Let's see the block diagram of Embedded C Programming development:

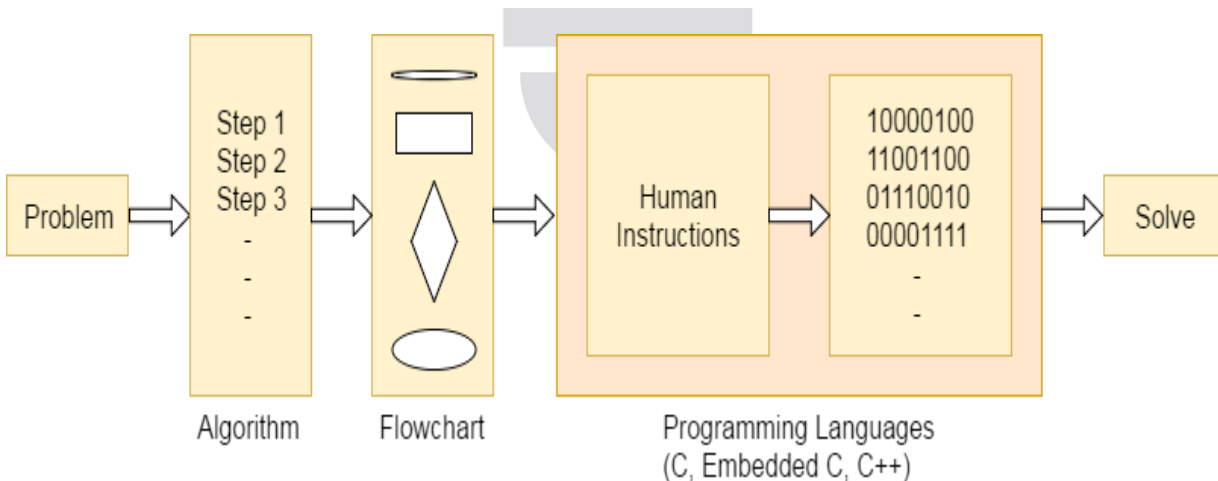


Function is a collection of statements that is used for performing a specific task and a collection of one or more functions is called a programming language. Every language is consisting of basic elements and grammatical rules. The C language programming is designed for function with variables, character set, data types, keywords, expression and so on are used for writing a C program.

The extension in C language is known as embedded C programming language. As compared to above the embedded programming in C is also have some additional features like data types, keywords and header file etc is represented by `#include<microcontroller name.h>`.

BASIC EMBEDDED C PROGRAMMING STEPS:

Let's see the block diagram representation of Embedded C Programming Steps:



The microcontroller programming is different for each type of operating system. Even though there are many operating system are exist such as Windows, Linux, RTOS, etc but RTOS has several advantage for embedded system development.

EMBEDDED SYSTEMS:

Embedded System is a system composed of hardware, application software and real time operating system. It can be small independent system or large combinational system.

Our Embedded System tutorial includes all topics of Embedded System such as characteristics, designing, processors, microcontrollers, tools, addressing modes, assembly language, interrupts, embedded c programming, led blinking, serial communication, lcd programming, keyboard programming, project implementation etc.

SYSTEM:

System is a way of working, organizing or performing one or many tasks according to a fixed set of rules, program or plan.

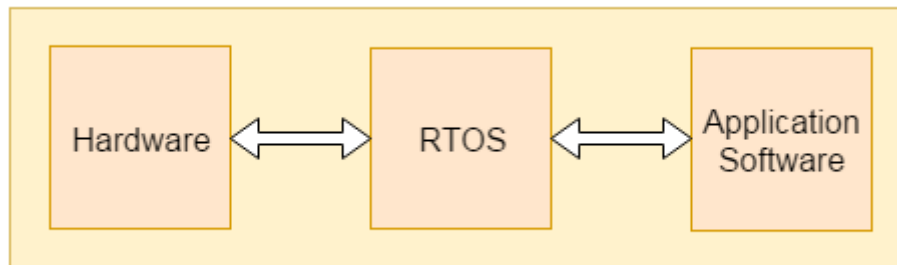
It is an arrangement in which all the unit combined to perform a work together by following certain set of rules in real time computation. It can also be defined as a way of working, organizing or doing one or many tasks according to a fixed plan.

An Embedded System is a system that has software embedded into computer-hardware, which makes a system dedicated for a variety of application or specific part of an application or product or part of a larger system.

An embedded system can be a small independent system or a large combinational system. It is a microcontroller-based control system used to perform a specific task of operation.

An embedded system is a combination of three major components:

- **Hardware:** Hardware is physically used component that is physically connected with an embedded system. It comprises of microcontroller based integrated circuit, power supply, LCD display etc.
- **Application software:** Application software allows the user to perform varieties of application to be run on an embedded system by changing the code installed in an embedded system.
- **Real Time Operating system (RTOS):** RTOS supervises the way an embedded system work. It act as an interface between hardware and application software which supervises the application software and provide mechanism to let the processor run on the basis of scheduling for controlling the effect of latencies.



CHARACTERISTICS OF EMBEDDED SYSTEM:

- An embedded system is software embedded into computer hardware that makes a system dedicated to be used for variety of application.
- Embedded system generally used for do specific task that provide real-time output on the basis of various characteristics of an embedded system.
- Embedded system may contain a smaller part within a larger device that used for serving the more specific application to perform variety of task using hardware-software intermixing configuration.
- It provides high reliability and real-time computation ability.

Advantages:

- Same hardware can be used in variety of application.
- Lesser power requirement
- Lower operational cost of system
- Provide high performance and efficiency

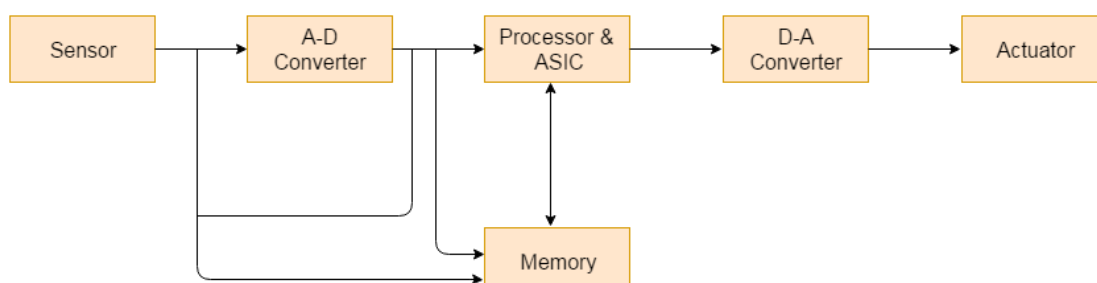
Disadvantages:

- Developing a system required more time. Due to functional complexity.
- Skilled engineers required because one mistake may result in destroying of complete project.

Designing of an embedded system

BASIC STRUCTURE OF AN EMBEDDED SYSTEM:

Let's see the block diagram shows the basic structure of an embedded system.

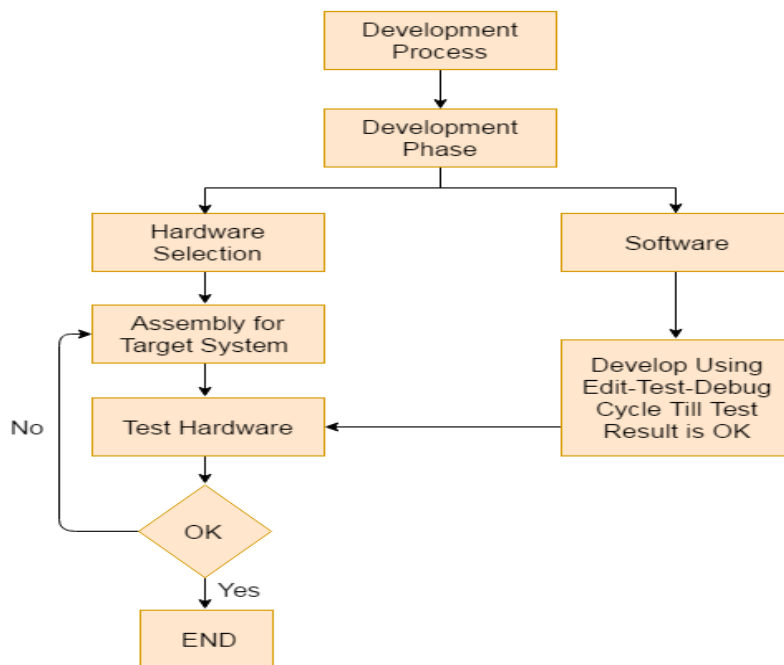


- **Sensor:** Sensor used for sensing the change in environment condition and it generate the electric signal on the basis of change in environment condition. Therefore it is also called as transducers for providing electric input signal on the basis of change in environment condition.
- **A-D Converter:** An analog-to-digital converter is a device that converts analog electric input signal into its equivalent digital signal for further processing in an embedded system.
- **Processor & ASICs:** Processor used for processing the signal and data to execute desired set of instructions with high-speed of operation. Application specific integrated circuit (ASIC) is an integrated circuit designed to perform task specific operation inside an embedded system.
- **D-A Converter:** A digital-to-analog converter is a device that converts digital electric input signal into its equivalent analog signal for further processing in an embedded system.
- **Actuators:** Actuators is a comparator used for comparing the analog input signal level to desired output signal level for providing the error free output from the system.

DESIGN STEPS REQUIRED FOR THE DEVELOPMENT OF EMBEDDED SYSTEM:

Designing steps required for embedded system are different from the design process of another electronic system.

Let's see a flow chart represent the design steps required in the development of an embedded system:



EMBEDDED SYSTEM TOOLS AND PERIPHERALS:

Compiler:

Compiler is used for converting the source code from a high-level programming language to a low-level programming language. It converts the code written in high level programming language into assembly or machine code. The main reason for conversion is to develop an executable program.

Let's see the operations performed by compiler are:

- Code generation
- Code optimization
- Parsing
- Syntax direct translation
- Preprocessing

Cross-Compiler:

If a program compiled is run on a computer having different operating system and hardware configuration than the computer system on which a compiler compiled the program, that compiler is known as cross-compiler.

Decompiler:

A tool used for translating a program from a low-level language to high-level language is called a decompiler. It is used for conversion of assembly or machine code to high-level programming language.

Assembler:

Assembler is embedded system tool used for translating a computer instruction written in assembly language into a pattern of bits which is used by the computer processor for performing its basic operations. Assembler creates an object code by translating assembly language instruction into set of mnemonics for representing each low-level machine operation.

DEBUGGING TOOLS IN AN EMBEDDED SYSTEM:

Debugging is a tool used for reducing the number of error or bugs inside a computer program or an assembled electronic hardware.

Debugging of a compact subsystem is difficult because a small change in one subsystem can create bugs in another system. The debugging used inside embedded system differs in terms of their development time and debugging features.

Let's see the different debugging tools used in embedded system are:

Simulators:

Simulator is a tool used for simulation of an embedded system. Code tested for microcontroller unit by simulating code on the host computer. Simulator is used for model the behavior of the complete microcontroller in software.

Functions of simulators:

Let's see the functions performed by simulator are:

- It defines the processing or processor device family with various version of target system.
- It monitors the detailed information of a source code and symbolic arguments as the execution goes for each single step of operation.
- It simulates the ports of target system for each single step of execution.
- It provides the working status of RAM.
- It monitors the response of system and determines the throughput.
- It provides the complete meaning of the present command.
- It monitors the detailed information of the simulator commands entered from the keyboard or selected from the menu.
- It facilitates synchronization of internal peripherals and delays.

MICROCONTROLLER STARTER KIT:

For developing an embedded system based project a complete microcontroller starter kit is required. The major advantage of this kit over simulator is that they work in real-time operating condition. Therefore it allows the easy input/output functional verification. Consider a microcontroller starter kit consists of:-

- Hardware Printed Circuit Board (PCB)
- In-System Programmer (ISP)
- Some embedded system tools like compiler, assembler, linker, etc
- Sometimes, there is a requirement of an Integrated Development Environment (IDE)

The above component available in microcontroller starter kit is completely enough and the cheapest option available for developing simple microcontroller projects.

Emulators:

An emulator is a software program or a hardware kit which emulates the functions of one computer system into another computer system. Emulators have an ability to support closer connection to an authenticity of the digital object.

It can also be defined as the ability of a computer program in electronic device to emulate another program or device. It focusing on recreating the original computer environment and helps a user to work on any type of application or operating system.

PERIPHERAL DEVICES IN EMBEDDED SYSTEMS:

Communication of an embedded system with an outside environment is done by using different peripheral devices as a combination with microcontroller.

Let's see the different peripheral devices in embedded system are:-

- Universal Serial Bus (USB)
- Networks like Ethernet, Local Area Network(LAN) etc
- Multi Media Cards (SD Cards, Flash memory, etc)
- Serial Communication Interface (SCI) like RS-232, RS-485, RS-422, etc
- Synchronous Serial Communication Interface like SPI, SSC and ESSI
- Digital to Analog/ Analog to Digital (DAC/ADC)
- General Purpose Input/Output (GPIO)
- Debugging like In System Programming (ISP), In Circuit Serial Programming (ICSP), BDM Port, etc

CRITERIA FOR CHOOSING MICROCONTROLLER:

Choosing a microcontroller is essential process in designing of embedded system. While selecting a microcontroller, make sure that it meets the system need and it must be cost effective. We need to decide whether an 8-bit, 16-bit or 32-bit microcontroller is best suitable for the computing needs of a task.

In addition to above, the following points need to be kept in mind while selecting a microcontroller:-

- **Speed:** The operational speed of the microcontroller or the highest speed microcontroller can support.
- **Packaging:** Packaging is important for improving the assembling, space and prototyping of an end-product.
- **RAM and ROM:** On the basis of operation of embedded system and memory need for storage data and programs the type of microcontroller required for designing system is decided.
- **Count of I/O pins:** The number of input and output devices connected with the system plays an essential role in choosing the type of microcontroller.
- **Cost per unit:** It is important in terms of final cost of the product in which the microcontroller is to be used.
- **Power consumption:** Power consumption plays an important role for maintaining the efficiency of an embedded system

3.1.3 ARDUINO SOFTWARE IDE

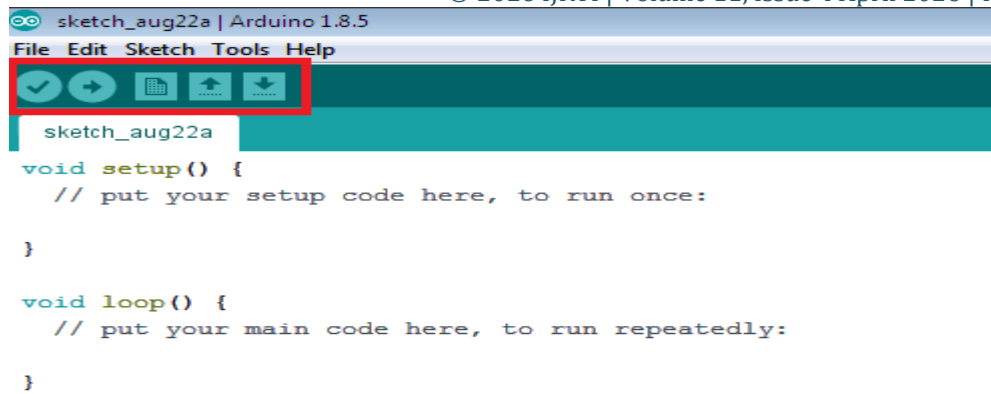
The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with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









WRITING SKETCHES:

Programs written using Arduino Software (IDE) are called **sketches**. These sketches are written in the text editor and are saved with the file extension **.ino**. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

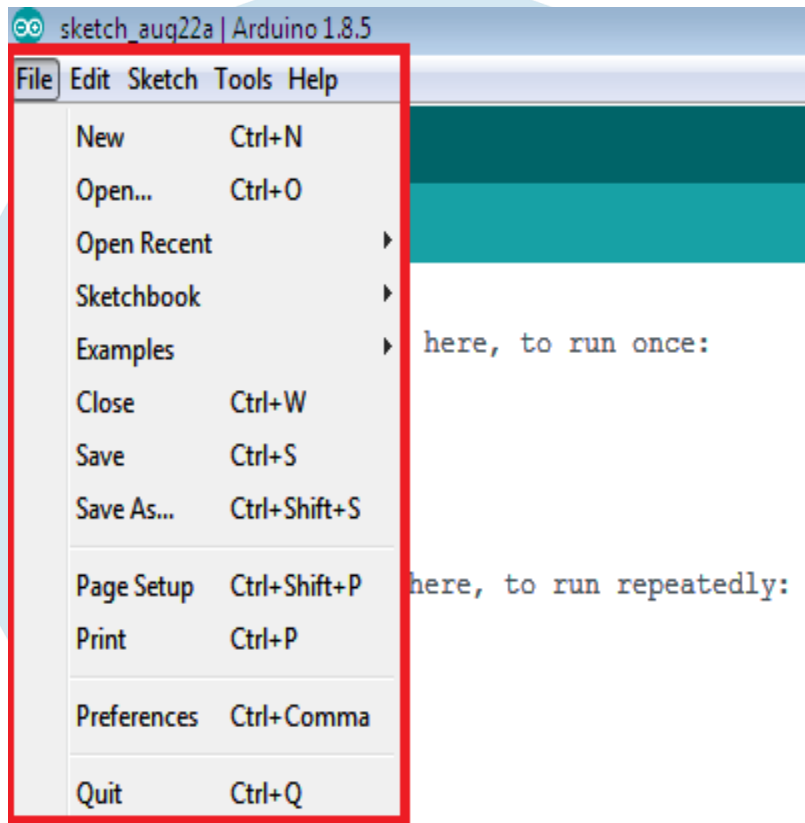
NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.



	<p>Verify Checks your code for errors compiling it.</p>
	<p>Upload Compiles your code and uploads it to the configured board. See uploading below for details.</p> <p>Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"</p>
	<p>New Creates a new sketch.</p>
	<p>Open Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.</p> <p>Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File Sketchbook menu instead.</p>
	<p>Save Saves your sketch.</p>
	<p>SerialMonitor Opens the serial monitor.</p>

Additional commands are found within the five menus: **File**, **Edit**, **Sketch**, **Tools**, and help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

FILE:



➤ ***New***

Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.

➤ ***Open***

Allows loading a sketch file browsing through the computer drives and folders.

➤ ***OpenRecent***

Provides a short list of the most recent sketches, ready to be opened.

➤ ***Sketchbook***

Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.

➤ ***Examples***

Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.

➤ ***Close***

Closes the instance of the Arduino Software from which it is clicked.

➤ **Save**

Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.

➤ **Saveas...**

Allows saving the current sketch with a different name.

➤ **PageSetup**

It shows the Page Setup window for printing.

➤ **Print**

Sends the current sketch to the printer according to the settings defined in Page Setup.

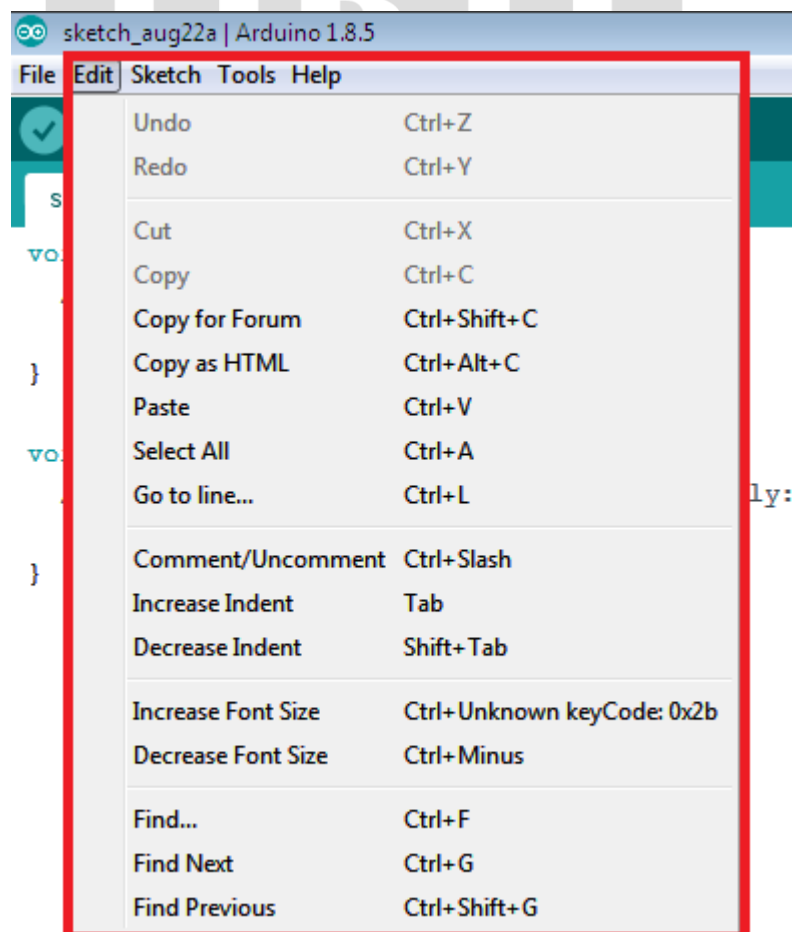
➤ **Preferences**

Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.

➤ **Quit**

Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

EDIT:



➤ ***Undo/Redo***

Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.

➤ ***Cut***

Removes the selected text from the editor and places it into the clipboard.

➤ ***Copy***

Duplicates the selected text in the editor and places it into the clipboard.

➤ ***CopyforForum***

Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.

➤ ***CopyasHTML***

Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.

➤ ***Paste***

Puts the contents of the clipboard at the cursor position, in the editor.

➤ ***SelectAll***

Selects and highlights the whole content of the editor.

➤ ***Comment/Uncomment***

Puts or removes the // comment marker at the beginning of each selected line.

➤ ***Increase/DecreaseIndent***

Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.

➤ ***Find***

Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.

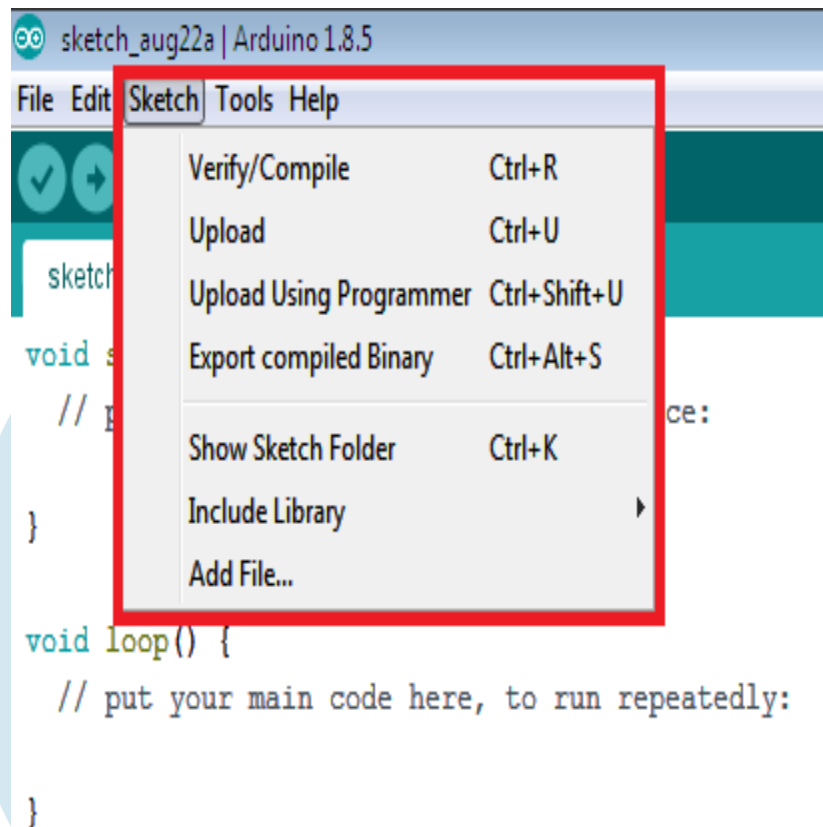
➤ ***FindNext***

Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.

➤ ***FindPrevious***

Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

SKETCH:



➤ ***Verify/Compile***

Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.

➤ ***Upload***

Compiles and loads the binary file onto the configured board through the configured Port.

➤ ***Upload Using Programmer***

This will overwrite the boot loader on the board; you will need to use Tools > Burn Boot loader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a *Tools -> Burn Bootloader* command must be executed.

➤ ***Export Compiled Binary***

Saves a .hex file that may be kept as archive or sent to the board using other tools.

➤ ***Show Sketch Folder***

Opens the current sketch folder.

➤ ***Include Library***

Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see [libraries](#) below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.

➤ ***Add File...***

Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side of the toolbar.

TOOLS:

➤ ***Auto Format***

This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.

➤ ***Archive Sketch***

Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.

➤ ***Fix Encoding & Reload***

Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.

➤ ***Serial Monitor***

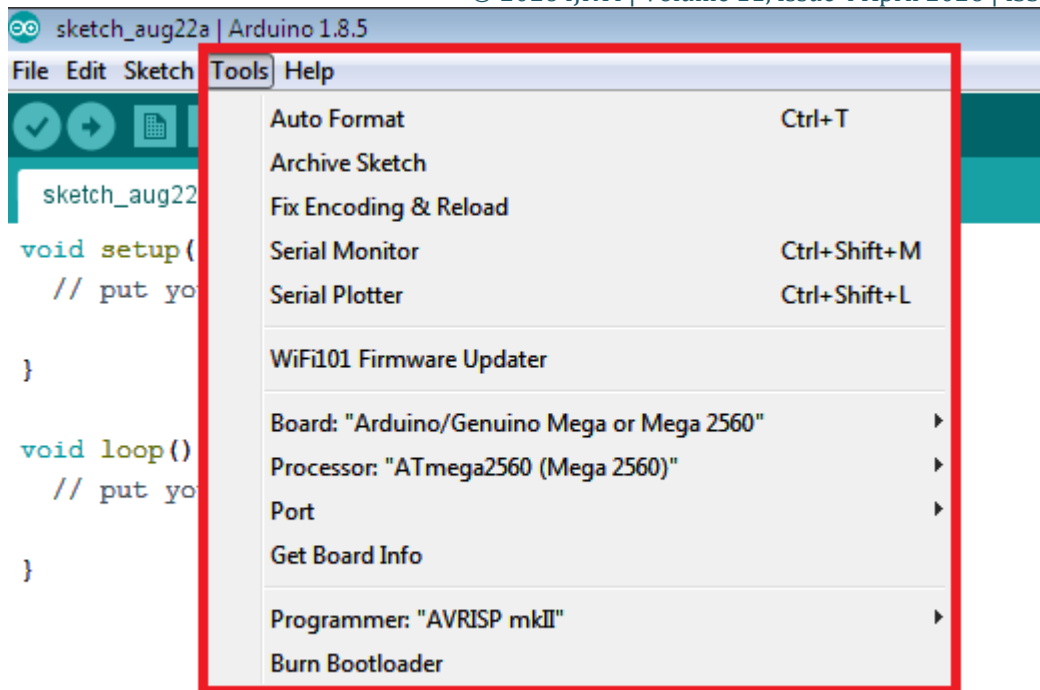
Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.

➤ ***Board***

Select the board that you're using. See below for [descriptions of the various boards](#).

➤ ***Port***

This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.



➤ **Programmer**

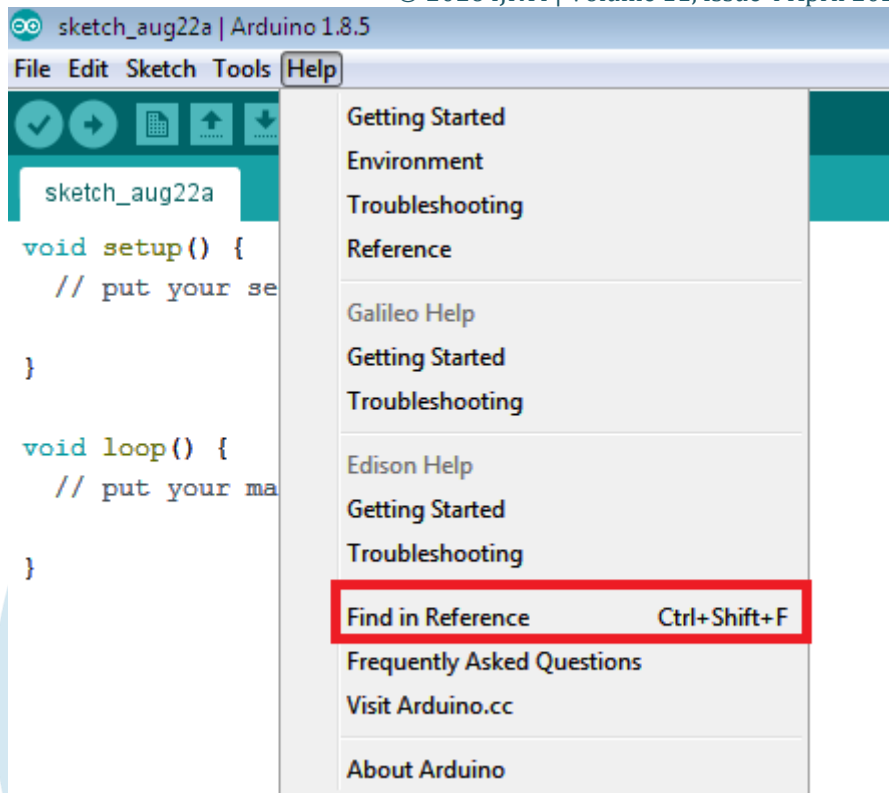
For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a boot loader to a new microcontroller, you will use this.

➤ **Burn Boot loader**

The items in this menu allow you to burn a boot loader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new ATmega microcontroller (which normally comes without a boot loader). Ensure that you've selected the correct board from the **Boards** menu before burning the boot loader on the target board. This command also set the right fuses.

Help:

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

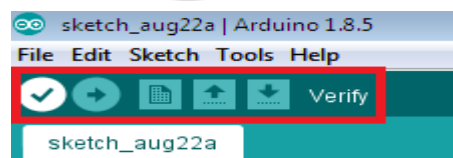


➤ *Find in Reference*

This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

SKETCHBOOK:

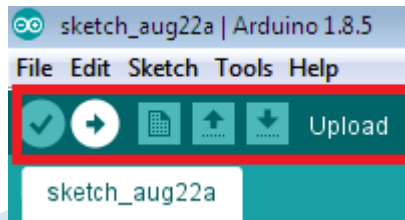
The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the **File > Sketchbook** menu or from the **Open** button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the **Preferences** dialog.



Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino.

TABS, MULTIPLE FILES, AND COMPILATION:

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

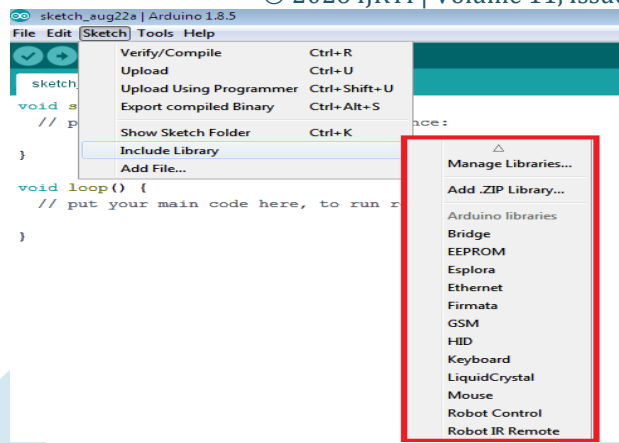
UPLOADING:

Before uploading your sketch, you need to select the correct items from the **Tools > Board** and **Tools > Port** menus. The boards are described below. On the Mac, the serial port is probably something like `/dev/tty.usbmodem241` (for an Uno or Mega2560 or Leonardo) or `/dev/tty.usbserial-1B1` (for a Duemilanove or earlier USB board), or `/dev/tty.USA19QW1b1P1.1` (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be `/dev/ttyACMx`, `/dev/ttyUSBx` or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the **Upload** item from the **Sketch** menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino **bootloader**, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The boot loader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

LIBRARIES:

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the **Sketch > Import Library** menu. This will insert one or more `#include` statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its `#include` statements from the top of your code.



There is a [list of libraries](#) in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these [instructions for installing a third-party library](#).

THIRD-PARTY HARDWARE:

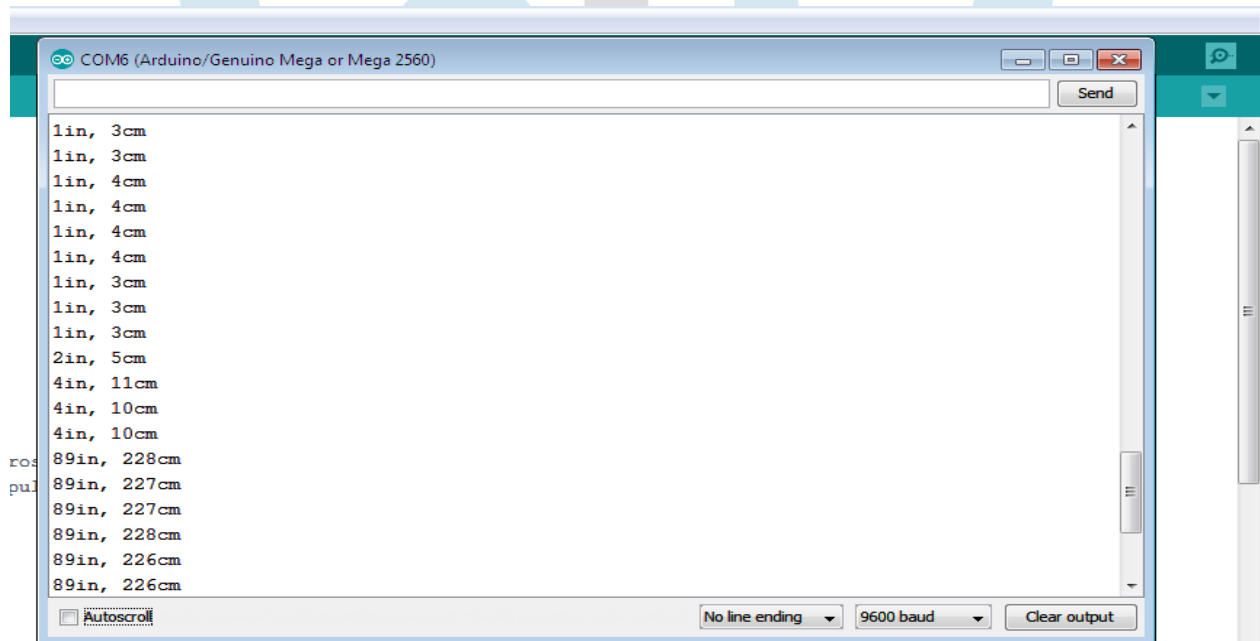
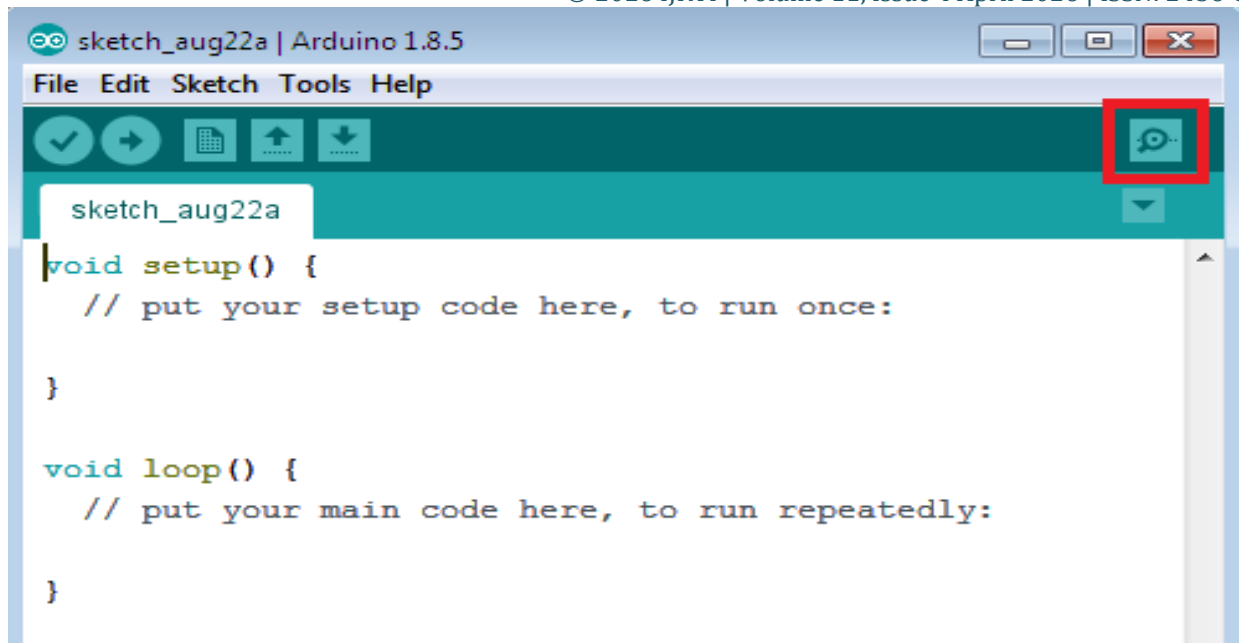
Support for third-party hardware can be added to the **hardware** directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the **hardware** directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

For details on creating packages for third-party hardware, see the [Arduino IDE 1.5 3rd party Hardware specification](#).

SERIAL MONITOR:

This displays serial sent from the Arduino or Genuino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to **Serial.begin** in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.

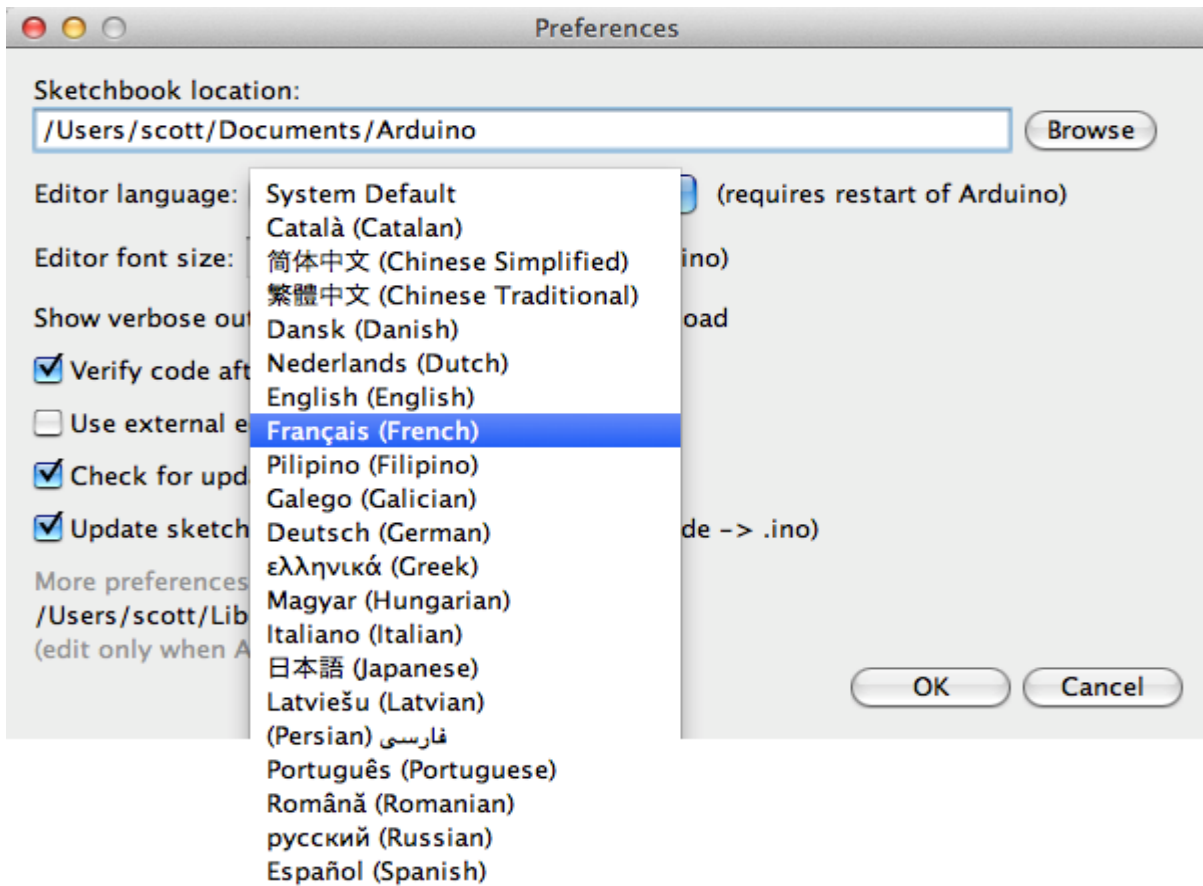
You can also talk to the board from Processing, Flash, MaxMSP, etc (see the [interfacing page](#) for details).



PREFERENCES:

Some preferences can be set in the preferences dialog (found under the **Arduino** menu on the Mac, or **File** on Windows and Linux). The rest can be found in the preferences file, whose location is shown in the preference dialog.

LANGUAGE SUPPORT:



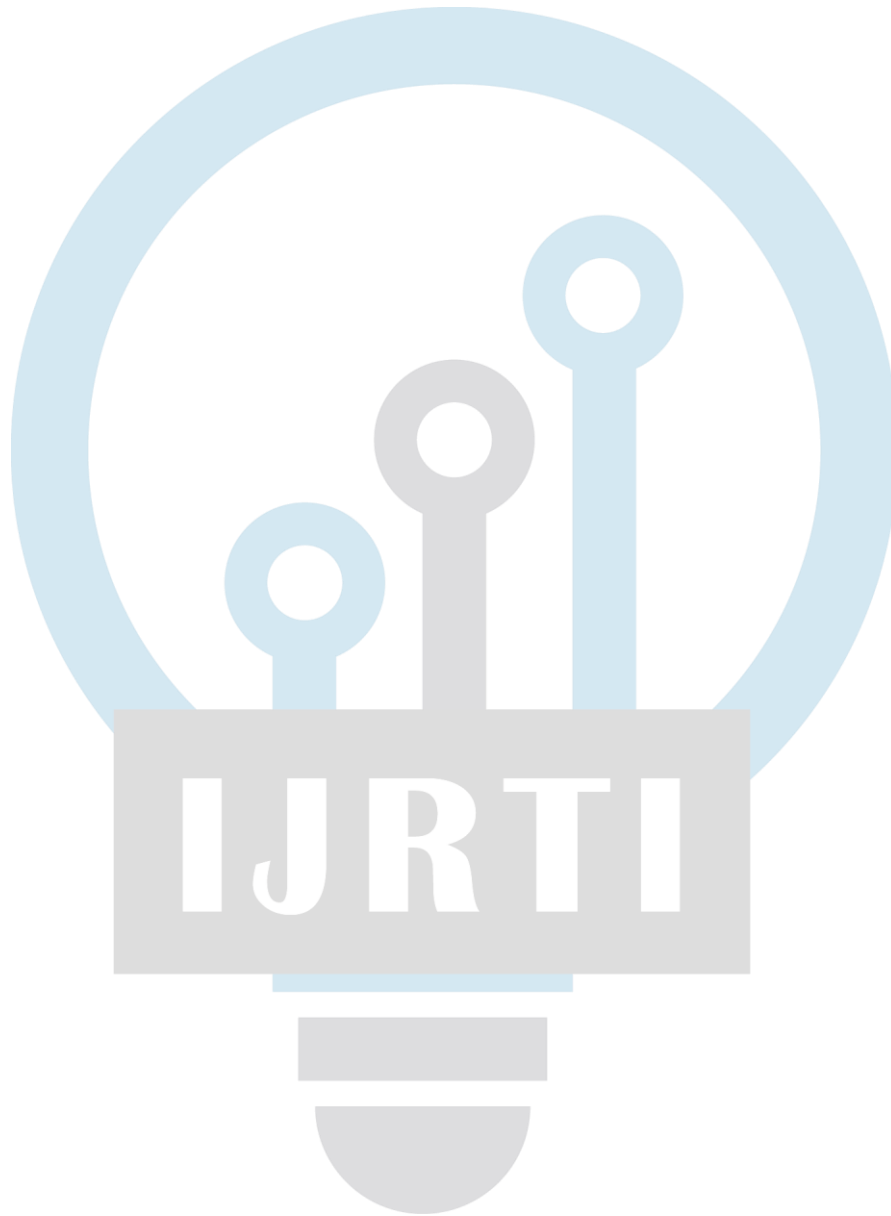
Since version 1.0.1, the Arduino Software (IDE) has been translated into 30+ different languages. By default, the IDE loads in the language selected by your operating system. (Note: on Windows and possibly Linux, this is determined by the locale setting which controls currency and date formats, not by the language the operating system is displayed in.)

If you would like to change the language manually, start the Arduino Software (IDE) and open the **Preferences** window. Next to the **Editor Language** there is a dropdown menu of currently supported languages. Select your preferred language from the menu, and restart the software to use the selected language. If your operating system language is not supported, the Arduino Software (IDE) will default to English.

You can return the software to its default setting of selecting its language based on your operating system by selecting **System Default** from the **Editor Language** drop-down. This setting will take effect when you restart the Arduino Software (IDE). Similarly, after changing your operating system's settings, you must restart the Arduino Software (IDE) to update it to the new default language.

CHAPTER: 4 - OUTPUT

1.1 OUTPUT:



CHAPTER: 5 -FUTURE ENHANCEMENT

5.1 APPLICATIONS

- **Home-Based Elderly Care:** This system can be effectively used in home environments to monitor and manage circulatory health in elderly individuals. It ensures timely detection of swelling and provides automatic therapy, reducing the need for caregiver intervention.
- **Rehabilitation Centers:** In rehabilitation facilities, the system can aid in post-treatment recovery for patients with poor circulation or vein-related issues by offering consistent monitoring and therapy, improving healing outcomes through automated care solutions.
- **Remote Health Monitoring:** For elderly individuals living in remote or rural areas with limited access to healthcare, this system offers a reliable, portable solution for continuous monitoring and localized therapy, ensuring proactive and independent health management.

5.2 FUTURE ENHANCEMENT:

In the future, this system can be enhanced by integrating advanced features such as wireless connectivity for remote health monitoring and data sharing with healthcare professionals or caregivers through cloud-based platforms. Incorporating AI and machine learning algorithms can improve the system's ability to predict health risks based on historical data, enabling more personalized and accurate therapeutic responses. Additional sensors can be included to monitor other vital signs such as heart rate, blood pressure, and oxygen levels, creating a more comprehensive health monitoring solution. Mobile app integration could allow users and family members to receive real-time alerts and reports, improving engagement and responsiveness. Voice assistance and speech feedback can be added for better accessibility, especially for visually impaired or less tech-savvy users. Furthermore, miniaturization and wearable designs can make the system more comfortable and discreet for continuous use. These future enhancements will significantly broaden the scope, usability, and effectiveness of the system in elderly healthcare.

5.3 ADVANTAGES:

The proposed system offers several significant advantages, particularly in enhancing elderly healthcare and promoting independent living. One of the primary benefits is its ability to provide real-time monitoring and early detection of swelling or poor blood circulation, reducing the risk of complications through timely intervention. The system operates autonomously, activating vibration therapy without the need for user input or caregiver assistance, making it ideal for individuals with limited mobility. Its non-invasive therapeutic approach ensures safety and comfort, eliminating reliance on medications or clinical procedures for minor circulatory issues. Additionally, the real-time display keeps users informed about their health status, fostering confidence and awareness. The system is compact, affordable, and easy to use, making it accessible for home-based care. By minimizing the need for regular hospital visits and

manual monitoring, it eases the burden on caregivers and healthcare providers, while significantly improving the quality of life and self-care capabilities of elderly individuals.

5.3 CONCLUSION:

In conclusion, the proposed system presents a smart, efficient, and user-friendly solution for monitoring and managing circulatory issues, particularly swollen veins in elderly individuals. By leveraging continuous temperature monitoring and intelligent decision-making, the system ensures early detection of abnormal physiological conditions and delivers immediate, non-invasive vibration therapy to alleviate discomfort and improve blood circulation. Its automated operation reduces the need for constant caregiver involvement, empowering elderly users to manage their health more independently. The real-time display of health status further enhances transparency and awareness, making the system reliable and informative. Designed with simplicity and practicality in mind, it is well-suited for home use, rehabilitation centers, and remote care applications. By integrating proactive healthcare with responsive therapeutic intervention, the system not only improves the quality of life for aging individuals but also supports the broader goal of accessible and affordable healthcare. Overall, it offers a meaningful advancement in the field of elderly wellness and personalized medical support.

5.5 REFERENCES:

- [1] Martino Alessandrini et al. “A Computational Framework to Benchmark Basket Catheter Guided Ablation in Atrial Fibrillation”. In: *Frontiers in Physiology* 9 (2018). ISSN: 1664-042X. DOI: 10.3389/fphys.2018.01251. URL: <https://www.frontiersin.org/journals/physiology/articles/10.3389/fphys.2018.01251>.
- [2] Elad Anter et al. “Activation mapping with integration of vector and velocity information improves the ability to identify the mechanism and location of complex scar-related atrial tachycardias”. In: *Circulation: Arrhythmia and Electrophysiology* 11.8 (2018), e006536.
- [3] C.D. Cantwell et al. “Techniques for automated local activation time annotation and conduction velocity estimation in cardiac mapping”. In: *Computers in Biology and Medicine* 65 (2015), pp. 229–242. ISSN: 0010-4825. DOI: <https://doi.org/10.1016/j.compbimed.2015.04.027>. URL: <https://www.sciencedirect.com/science/article/pii/S0010482515001456>.
- [4] Sam Coveney et al. “Probabilistic Interpolation of Uncertain Local Activation Times on Human Atrial Manifolds”. In: *IEEE Transactions on Biomedical Engineering* 67.1 (Jan. 2020), pp. 99–109. ISSN: 0018-9294, 1558-2531. DOI: 10.1109/TBME.2019.2908486. URL: <https://ieeexplore.ieee.org/document/8684876/> (visited on 10/12/2021).
- [5] R Dubois et al. “Global and Directional Activation maps for cardiac mapping in electrophysiology”. In: *2012 Computing in Cardiology*. 2012, pp. 349–352.

[6] Josselin Duchateau et al. “Performance and limitations of noninvasive cardiac activation mapping”. In: Heart rhythm 16.3 (2019), pp. 435–442.

[7] Kazutaka Gima and Yoram Rudy. “Ionic current basis of electrocardiographic waveforms: model study”. In: Circulation research 90.8 (2002), pp. 889–896

[8] Natasja M S de Groot et al. “Critical appraisal of technologies to assess electrical activity during atrial fibrillation: a position paper from the European Heart Rhythm Association and European Society of Cardiology Working Group on eCardiology in collaboration with the Heart Rhythm Society, Asia Pacific Heart Rhythm Society, Latin American Heart Rhythm Society and Computing in Cardiology”. In: EP Europace 24.2 (Dec. 2021), pp. 313–330. ISSN: 1099-5129. DOI: 10.1093/europace/euab254. eprint: <https://academic.oup.com/europace/articlepdf/24/2/313/42370306/euab254.pdf>. URL: <https://doi.org/10.1093/europace/euab254>.

[9] David E Haines et al. “Electrographic flow mapping for atrial fibrillation: theoretical basis and preliminary observations”. In: Journal of Interventional Cardiac Electrophysiology 66.4 (2023), pp. 1015–1028.

[10] Basna Mohammed Salih Hasan and Adnan Mohsin Abdulazeez. “A review of principal component analysis algorithm for dimensionality reduction”. In: Journal of Soft Computing and Data Mining 2.1 (2021), pp. 20–30.

