ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 5, May 2021

Development of Economic and Early Detection of Sleep Apnea Monitoring System with Respiratory Channels

M. Rubesh Ram

Department of Biomedical Engineering, St.Peter's Institute of Higher Education and Research, Avadi, Chennai, India

***Abstract*---AN Obstructive Sleep Apnea (OSA) is a serious health disorder which contributes to cardiovascular complications, decreased work productivity, automobile accidents, and death. This condition is characterized by a temporary and repetitious cessation of breathing resulting from repeated upper airway closure during a person's sleep. OSA is classified by several different components. Apneas, Hypopneas and blood oxygen saturation (spo2) are the main constituents of this disorder. An apneic episode is a complete cessation of airflow lasting 10 sec or more, while a hypopnea is a reduction in respiratory airflow combined with a spo2 decrease of >4%. Additional factors, which the sleep specialist feels are clinically significant, may also be accounted for and included when diagnosing OSA. The instrument used in diagnosing OSA usually require the patient to undergo a sleep study called a polysomnogram (PSG). This overnight study consists of monitoring brain electrophysiological responses, heart rate and rhythms, respiration, blood oxygen levels, and leg movements. Electrodes are attached to the patient's skin and monitored by a sleep technician throughout the test. Data from this test is used to diagnose OSA, grade its severity and help determine optimal therapeutic approaches. So OSA has acquired an severe problem associated with pulmonary circulation. Normally the PSG test is carried out in clinics and hospital associated with sleep technician due to this patient's sleep gets disturbed in hospital environment within the same time . Therefore, the goal of this project is to design a portable, lightweight and low-cost sleep monitoring system to detect apnea for people with a snoring problem using flow sensor and spo2 we are going to detect OSA, Hypopneas and central sleep apnea and display the Indexes in the display as an real time monitoring. The Proposed system will help the doctors in early detection for OSA which will be more efficient to titrate with an PAP devices such as CPAP (continuous positive airway pressure) and with BPAP (bidirectional positive airway pressure). So due to our proposed system an lower grade peoples also gets diagnosed earlier with minimal cost.**

I. INTRODUCTION

- Sleep apnea is a disorder characterized by a reduction or pause of breathing(airflow) during sleep. It is common among adults but rare among children.
- The treatment of sleep may be either surgical or nonsurgical during apnea sleep usually disrupted due to inadequate breathing and poor oxygen levels in the blood.
- In this project we are going to detect the AHI index and record the waveforms and desaturation levels and with alarms doctors can easily analyze the AHI immediately and can go with Cpap titrations.

II. LITERATURE SURVEY

S.No	Title	Author	Description
1	Obstructive sleep apnea devices for out-of-center(ooc) testing, technology evaluation.	N.A.Collop,S.L.Tracy,V.Kapur et al. "Journal of clinical sleep medicine,vol.7,pp.531-548,2011.	• CT Image In DICOM Format • Median Filter 5x5x5 Mask • Unsharp Masking To Sharpen The Image • Thresholding Is Set To Eliminate Unwanted Tissues With High And Low Intensities • Morphological Closing And Opening Operation Is Carried Out To Eliminate The Artifacts Due To Thresholding.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 5, May 2021

			• Normal Tissues Are Eliminated By Dilation, Complementation, Intersection • Segmentation Is Carried Out By 3D Labelling Algorithm • Sensitivity=93.3%, Specificity=90%
2	An unobtrusive sleep monitoring system for the human sleep behaviour understanding.	Paolo barsocchi; Monica bianchini;Antonino crivello;Davide la rosa; 2016 7TH International conference on cognitive infocommunications (coginfocom)	• Multi tumor in Brain MRI image in DICOM format • Median filter to remove noise • Feature containing irregular tissues are pulled out • For the removed clusters thresholding is applied • Features such as local contrast, entropy, valuable correlation are extracted using GLCM • Classification is performed by combination of SVM and fuzzy c-means
3	A deep learing architecture for temporal sleep stage classification using multivariate And multimodal time series.	Chambon s, Galtier m, Arnal p, Wainrib g, Gramfort a.IEEE Trans neural syst rehabiltation engineering 2018.	• Anatomical and Functional images was extracted using Anisotropic diffusion Filter • Random Walk is conducted on PET image • Down hill region is detected and used as Initial seed for CT images • Graph cut is performed by Maximum flow algorithm • Performance of segmentation is analyzed using Dice Similarity Coefficient against Ground truth • Graph cut method has small cut problem. Hence combination of Random walk and graph cut is used

III. SYSTEM ANALYSIS

3.1 PROPOSED SYSTEM:

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

3.2 REQUIREMENT SPECIFICATIONS

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

3.3 HARDWARE AND SOFTWARE SPECIFICATION

Software Requirement:

- **Compiler : Arduino IDE**
- **Language : C, C++**

Hardware Requirement:

- **Ardina Mega**
- **Arduino Nano**
- **Max 30100**
- **LCD**
- **Power Supply**

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 5, May 2021

3.4 BLOCK DIAGRAM

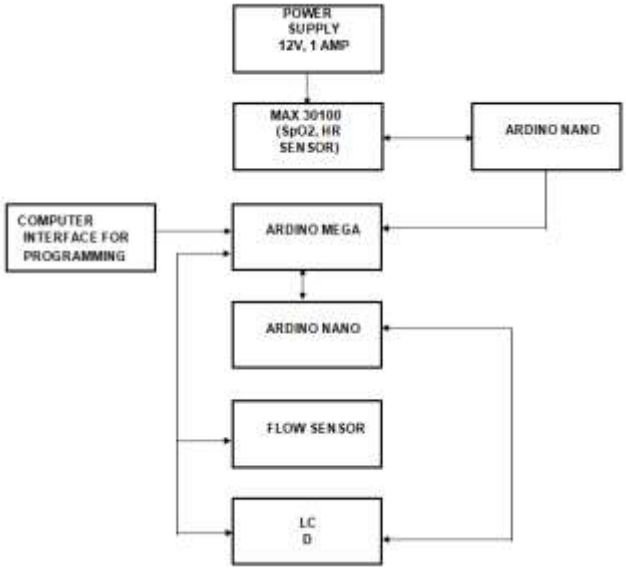


FIG 3.1

3.5 BLOCK DIAGRAM DESCRIPTION:

- **MAX 30100 :-**
  - Used for Spo2 and Heart rate Calculation.
- **ARDINO NANO:-**
  - Supply and interface for MAX30100 and ARDINO Mega.
- **COMPUTER USB:-**
  - For Programming and calculating Apneas.
- **ARDINO MEGA:-**
  - Using Rasberry Pie Software Programmed in ARDINO Mega to Detect Apneas.
- **ARDINO NANO:-**
  - Separate Ardino nano for ARDINO Mega Supply
- **FLOW SENSOR:-**
  - Attached with Ardino Mega to Calculate Flow .
- **LCD:-**
  - Calculated Apneas ,Spo2 and Heart Rate are Displayed Here.

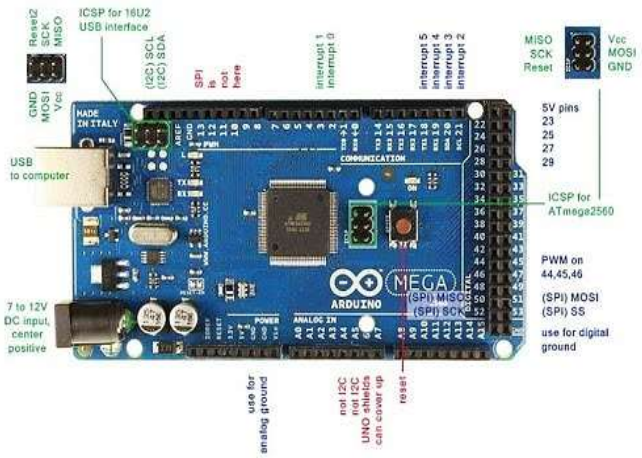
IV. SYSTEM DESIGN

4.1 HARDWARE MODULES DETAILS:

- **Ardina Mega**
- **Arduino Nano**
- **Max 30100**
- **LCD**
- **Power Supply**

4.1.1 ARDUINO MEGA:

FIG 4.1



Specifications:

- The ATmega2560 is a Microcontroller
- The operating voltage of this microcontroller is 5volts
- The recommended Input Voltage will range from 7volts to 12volts
- The input voltage will range from 6volts to 20volts
- The digital input/output pins are 54 where 15 of these pins will supply PWM o/p.
- Analog Input Pins are 16
- DC Current for each input/output pin is 40 mA
- DC Current used for 3.3V Pin is 50 mA
- Flash Memory like 256 KB where 8 KB of flash memory is used with the help of boot loader
- The static random access memory (SRAM) is 8 KB
- The electrically erasable programmable read-only memory (EEPROM) is 4 KB
- The clock (CLK) speed is 16 MHz
- The USB host chip used in this is MAX3421E
- The length of this board is 101.52 mm
- The width of this board is 53.3 mm
- The weight of this board is 36 g

Key Benefits:

- Low cost
- Consistent board format

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 5, May 2021

- 10x faster processing
- Added connectivity

Key Applications:

- Low cost PC/tablet/laptop
- IoT applications
- Media centre
- Robotics
- Industrial/Home automation
- Server/cloud server
- Print server
- Security monitoring
- Web camera
- Gaming
- Wireless access point
- Environmental sensing/monitoring (e.g. weather station)

- **Power**

The Arduino uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board FTDI chip. Maximum current draw is 50 mA.
- **GND.** Ground pins.

- **Memory**

The ATmega1280 has 128 KB of flash memory for storing code (of which 4 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

- **Input and Output**

Each of the 54 digital pins on the uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (**interrupt **5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 2 to 13 and 44 to 46.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 50 (**MISO**), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out

on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.

- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I^2C: 20 (SDA) and 21 (SCL).** Support I^2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I^2C pins on the Duemilanove or Diecimila.

The uno has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 5, May 2021

There are a couple of other pins on the board:

- AREF**.** Reference voltage for the analog inputs. Used with analogReference().
- Reset**.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

- **Communication**

The Arduino uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega1280 provides four hardware UARTs for TTL (5V) serial communication. An FTDI FT232RL on the board channels one of these over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash
when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).
A SoftwareSerial library allows for serial communication on any of the uno's digital pins.
The ATmega1280 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the ATmega1280 datasheet.

- **Programming**

The Arduino uno can be programmed with the Arduino software (download). For details, see the reference and tutorials.
The ATmega1280 on the Arduino uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).
You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

- **Automatic (Software) Reset**

Rather than requiring a physical press of the reset button before an upload, the Arduino uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega1280 via a 100 nanofarad capacitor.

- **Physical Characteristics and Shield Compatibility**

The maximum length and width of the uno PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The uno is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the uno and Duemilanove

4.1.2 ARDUINO NANO:

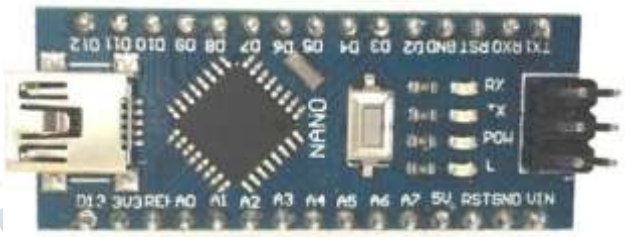


FIG 4.1

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

Memory
The ATmega328 has 32 KB, (also with 2 KB used for the bootloader. The ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output
Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 5, May 2021

kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB- to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() function. Analog pins 6 and 7 cannot be used as digital pins. Additionally, some pins have specialized functionality:

- I2C: A4 (SDA) and A5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analogReference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Nano's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. To use the SPI communication, please see ATmega328 datasheet.

Programming

The Arduino Nano can be programmed with the Arduino software (download). Select "Arduino Duemilanove or Nano w/ ATmega328" from the Tools > Board menu (according to the microcontroller on your board). The ATmega328 on the Arduino Nano comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar.

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Nano is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that

the bootloader can have a shorter timeout, as the lowering of DTR can be well- coordinated with the start of the upload. This setup has other implications. When the Nano is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Nano. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a

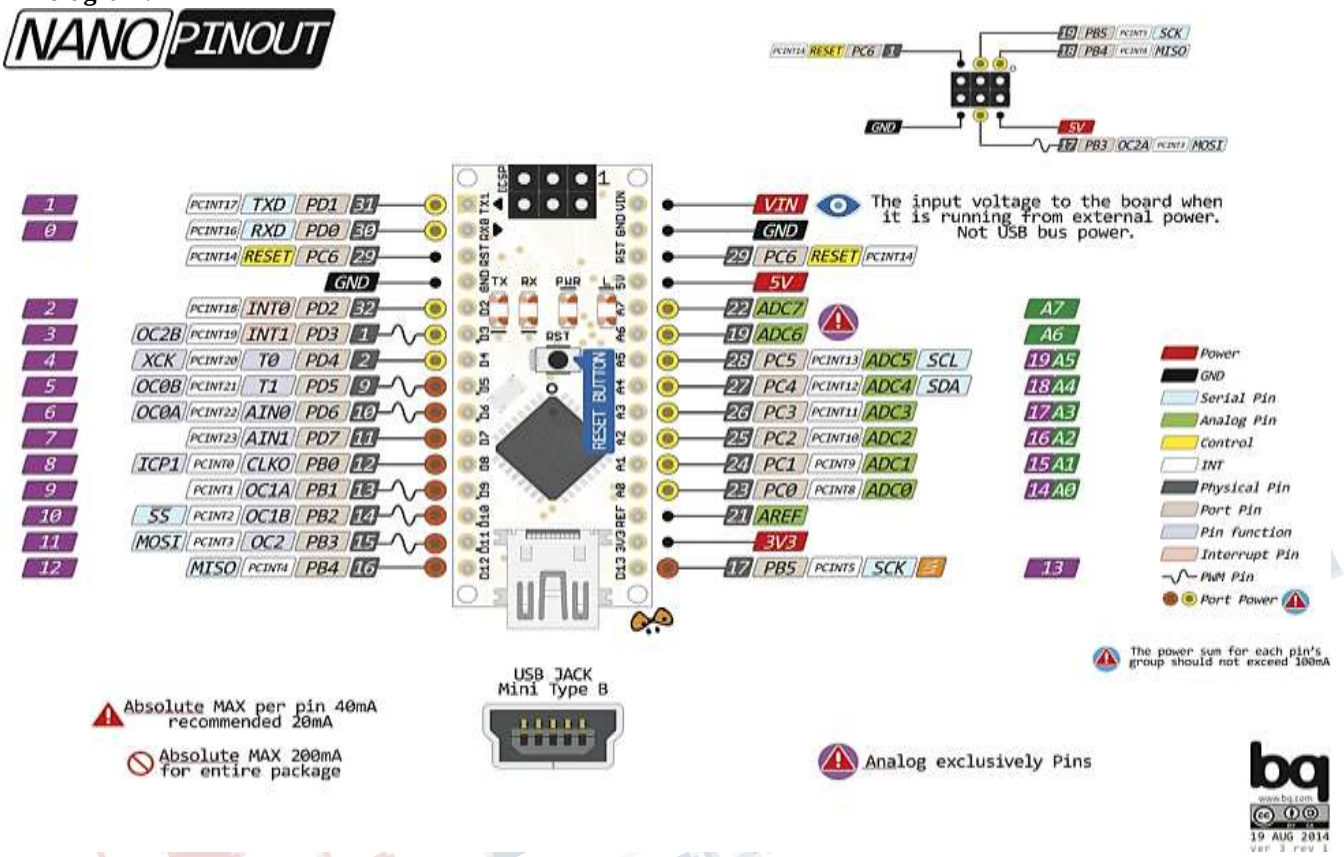
ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 5, May 2021

second after opening the connection and before sending this data.

Pin diagram:



4.1.3l MAX30100:

MAX30100 is an integrated pulse oximeter and heart-rate monitor sensor solution. It's an optical sensor that derives its readings from emitting two wavelengths of light from two LEDs – a red and an infrared one – then measuring the absorbance of pulsing blood through a photodetector. This particular LED colour combination is optimized for reading the data through the tip of one's finger. It is fully configurable through software registers and the digital output data is stored in a 16-deep FIFO within the device.It has an I2C digital interface to communicate with a host microcontroller.

The pulse oximetry subsystem in MAX30100 consists of ambient light cancellation (ALC), 16-bit sigma delta ADC, and proprietary discrete time filter. It has an ultra-low-power operation which makes it ideal for battery operated systems. MAX30100 operates on a supply in the range of 1.8 to 3.3V.It can be used in wearable devices, fitness assistant devices, medical monitoring devices, etc.The MAX30100 operates from 1.8V and 3.3V power supplies and can be powered down through software with negligible standby current, permitting the power supply to remain connected at all times.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 5, May 2021

- **Pin Configuration of MAX30100 Pulse Oximeter Heart Rate Sensor Module:-**

SN	PINS	DEFINITION OF PINS
1	VIN	Input voltage (1.8V to 5.5V)
2	SCL	IIC-SCL
3	SDA	IIC-SDA
4	INT	MAX30100INT
5	IRD	MAX30100 IR_DRV
6	RD	MAX30100 R_DRV
7	GND	Ground

Specifications and Features of MAX30100 Pulse Oximeter Heart Rate Sensor Module:-

- It is an integrated pulse oximetry and heart rate monitor sensor solution.
- Integrated LEDs, Photo Sensor, and High-Performance Analog Front -End
- Complete Pulse Oximeter and Heart-Rate Sensor Solution Simplifies Design
- Measures absorbance of pulsing blood
- I2C interface plus INT pin
- Tiny 5.6mm x 2.8mm x 1.2mm 14-Pin Optically Enhanced System-in- Package
- Ultra-Low-Power Operation Increases Battery Life for Wearable Devices
- Programmable Sample Rate and LED Current for Power Savings
- Ultra-Low Shutdown Current (0.7µA, typ)
- Advanced Functionality Improves Measurement Performance
- High SNR Provides Robust Motion Artifact Resilience
- Integrated Ambient Light Cancellation
- High Sample Rate Capability
- Fast Data Output Capability

- **Applications of MAX30100 Pulse Oximeter Heart Rate Sensor Module:-**

- Fitness Assistant Devices
- Medical Monitoring Devices
- Wearable Devices

- **4.1.48 LCD 16X2:**

LCD (Liquid Crystal Display) 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.

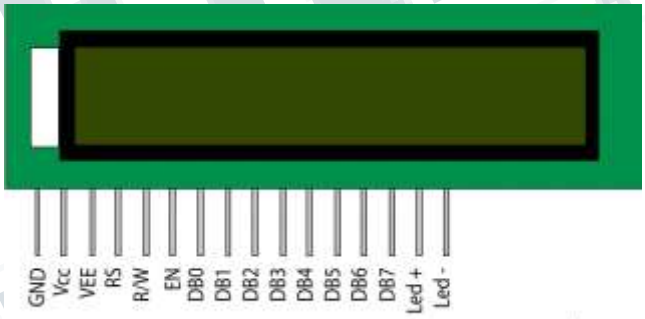


Fig 4.8

- **4.2 SOFTWARE REQUIREMENTS**

ARDUINO IDE:

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.
Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 5, May 2021

professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

- **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.

- **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.
Open Recent:
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.
Save as:
Allows saving the current sketch with a different name.
Page Setup:
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.
Copy for Forum
Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.
Copy as HTML
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.
Select All
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/Decrease Indent
Adds or subtracts a space at the beginning of each selected

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 5, May 2021

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.

Find Next

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

Find Previous

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 bootloader on the board; you will need to use Tools > Burn Bootloader 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 o 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 Genuine board but is useful if you purchase a new AT mega microcontroller (which normally come 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.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 5, May 2021

V. OUTPUT