ABSTRACT

In this project, we design an Arduino based real time clock with date. A Real Time Clock or RTC is a battery powered clock that measures time even when there is no external power or the microcontroller is reprogrammed. An RTC displays clock and calendar with all timekeeping functions. The battery, which is connected to the RTC displays clock and calendar with all timekeeping functions. The battery, which is connected to RTC is a separate one and is not related or connected to main power supply. When the power is restored, RTC displays the real time irrespective of the duration for which the power is off. Such Real Time Clocks are commonly found in commonly found in computers and are often referred to as just CMOS.

Most microcontrollers and microprocessors have built in the timers for keeping time. But they work only when the microcontrollers and microprocessors have built in timers for keeping time.

INTRODUCTION

At this time most people in the whole world use an automated digital clock in their everyday use. Starting from the hand watch we use were to those huge street clocks every one of us are dependent on the display the make. In 21th century time being more than money, regarding this change our hobbies of checking our time every minute is dramatically increasing. About 99% of today’s digital clocks are made using microcontrollers which make them more hand able from the rest, those we can set the time to start any minute or second we want and also set an alarm for remainder so that the system will store the value in memory and then when the time reaches the alarm will be on. As the microcontroller consists almost all the logical external logic gate doesn’t exist.

MOTIVATION

Although keeping time can be without an RTC, using one has benefits. Low power consumption (Important when running from alternate power). Frees the mains system for time critical tasks. Some times more accurate that other methods. A Digital clock is an alternative to a traditional analogue clock. This type of clock shows numbers to display the time in digital format, such as on a watch, phone or an alarm clock. This can be in both 12 and 24 hour formats. So it’s important for children to understand the two before introducing them to a digital clock . Some motherboards are made without real time clocks. The real time clock is omitted either out of the desire to save money (as in the raspberry pi system architecture) or because Real Time Clock may not needed at all (as in the Arduino system architecture).

OBJECTIVES OF PROJECT

This Arduino based Real time clock is a digital clock to display real time using a RTC IC DS3132 which works on I2C protocol. Real time clock means it runs even after power failure. When power is reconnected, it displays the real time respective to the time and duration it was in off state. In this Arduino alarm clock project we have used a 8×32 module to display the time in - "hour, minute, seconds, date, month and year" format. An Alarm option is also added and we can set up the alarm time. Once alarm time it saved in internal EEPROM of Arduino, it remains saved even after reset or electricity failure. Real time clocks are commonly used in our computers, houses, offices and electronics device for keeping them updated with real time.

BLOCK DIAGRAM:

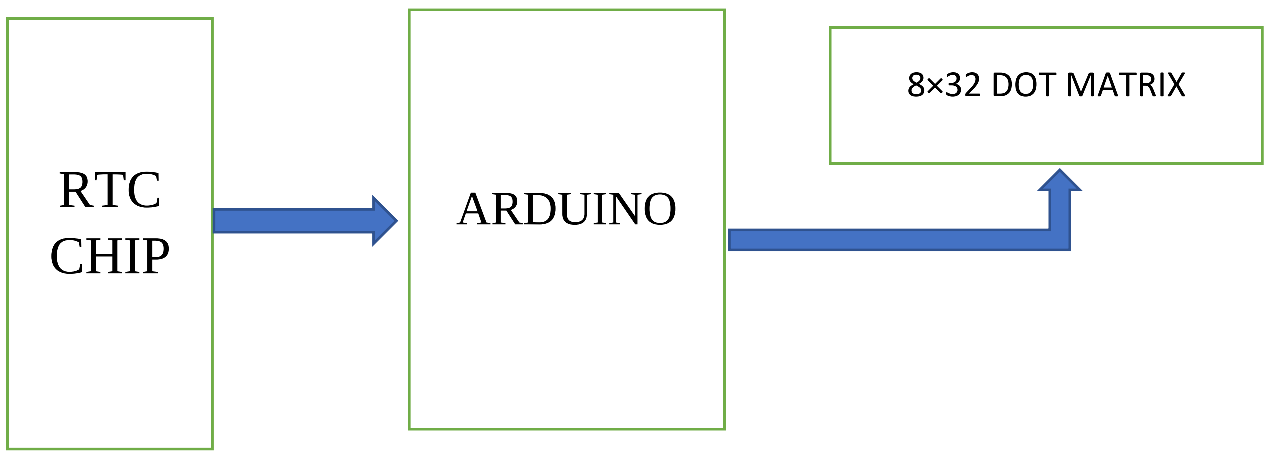

Fig 1: Block Diagram of digital clock

LITERATURE SURVEY

INTRODUCTION

Digital clocks can be very small and inexpensive devices that enhance the popularity of product designs, they are often incorporated into all kinds of devices such as cars, radios, televisions, microwave ovens, standard ovens, computers and cell phones. Sometimes their usefulness is disputed: a common complaint is that when time has to be set to Daylight Saving Time, many household clocks have to be readjusted. The incorporation of automatic synchronization by a radio time signal is reducing this problem (see Radio clock). Smart digital clocks, in addition to displaying time, scroll additional information such as weather and notifications.

EXISTING SYSTEM

The first digital pocket watch was the invention of Austrian engineer Josef Pallweber who created his "jump-hour" mechanism in 1883. Instead of a conventional dial, the jump-hour featured two windows in an enamel dial, through which the hours and minutes are visible on rotating discs. The second hand remained conventional. By 1885 Pallweber mechanism was already on the market in pocket watches by Cortébert and IWC; arguably contributing to the subsequent rise and commercial success of IWC. The principles of Pallweber jump-hour movement had appeared in wristwatches by the 1920s (Cortébert) and are still used today (Chronoswiss Digiteur). While the original inventor didn't have a watch brand at the time, his name has since been resurrected by a newly established watch manufacturer.

Plato clocks used a similar idea but a different layout. These spring-wound pieces consisted of a glass cylinder with a column inside, affixed to which were small digital cards with numbers printed on them, which flipped as time passed. The Plato clocks were introduced at the St. Louis World Fair in 1904, produced by Ansonia Clock Company. Eugene Fitch of New York patented the clock design in 1903. 13 years earlier Josef Pallweber had patented the same invention using digital cards (different from his 1885 patent using moving disks) in Germany (DRP No. 54093). The German factory Aktiengesellschaft für Uhrenfabrikation Lenzkirch made such digital clocks in 1893 and 1894.

The earliest patent for a digital alarm clock was registered by D.E Protzmann and others on October 23, 1956, in the United States. Protzmann and his associates also patented another digital clock in 1970, which was said to use a minimal amount of moving parts. Two side-plates held digital numerals between them, while an electric motor and cam gear outside controlled movement.

PROPOSED SYSTEM

In this project, we will make a clock with the help of the Arduino and RTC module. As we know the Arduino can not display the actual time so we will use the RTC module to display the right time on the Dot matrix. Read all the step carefully it will help you a lot to make the clockwork.

The DS3231 Arduino code is like the DS1307 code and it works with both RTC chips.

The Arduino code underneath doesn't utilize any library for the DS3231 RTC, the Wire library is for the correspondence between the Arduino and the DS3231 utilizing the I2C convention.

Many electronics engineers depend upon it for their time-based projects but RTC is not completely reliable. It is battery operated and has to be replaced soon or later. Coming to IoT projects it occupies valuable SPI(Serial Peripheral Interface) pins and gets messy with wires tangled around. Solution….Here comes our hero NTP(Network time protocol).NTP is so accurate since it gets time from the internet. We are going to operate this protocol using a client-server mode. the process is so simple that our Node acts as a client and request an NTP packet from the server using UDP. In return, the server sends a packet to the client which parses the data. NTP is the universal time synchronisation protocol. Now let us light up our labs work station.

CONCLUSION

Adding a real time clock to an Arduino makes it possible to build devices that are aware of the current time and date. This can allow you to create fancy timers and delay circuits, or just build a really cool and unique digital clock.

HARDWARE REQUIREMENTS

INTRODUCTION OF ARDUINO NANO

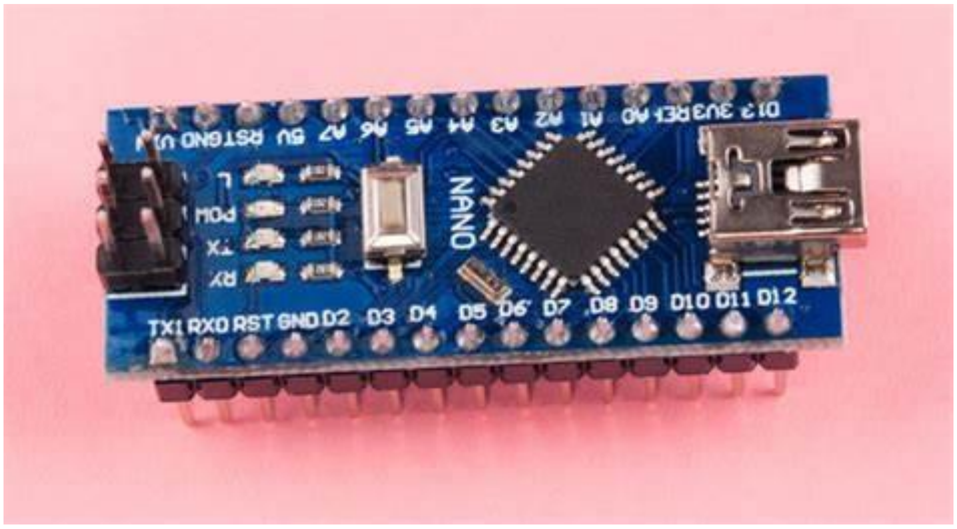
NANO
L
POW
TX
RX
TX1 RXD RST GND D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12
VIN GND RST 5V A7 A6 A5 A4 A3 A2 A1 A0 REF 3V3 D13

Fig 2: Arduino NANO

- Arduino Nano is a small, complete, flexible and breadboard-friendly Microcontroller board, based on ATmega328p, developed by Arduino.cc in Italy in 2008 and contains 30 male I/O headers, configured in a DIP30 style.
- Arduino Nano Pinout contains 14 digital pins, 8 analog Pins, 2 Reset Pins & 6 Power Pins.
- It is programmed using Arduino IDE, which can be downloaded from Arduino Official site.
- Arduino Nano is simply a smaller version of Arduino UNO, thus both have almost the same functionalities.
- It comes with an operating voltage of 5V, however, the input voltage can vary from **7 to** 12V.

- Arduino Nano's maximum current rating is 40mA, so the load attached to its pins shouldn't draw current more than that.
- Each of these Digital & Analog Pins is assigned with multiple functions but their main function is to be configured as Input/Output.
- Arduino Pins are acted as Input Pins when they are interfaced with sensors, but if you are driving some load then we need to use them as an Output Pin.
- Functions like pinMode() and digitalWrite() are used to control the operations of digital pins while analogRead() is used to control analog pins.
- The analog pins come with a total resolution of 10-bits which measures the value from 0 to 5V.
- Arduino Nano comes with a crystal oscillator of frequency 16 MHz. It is used to produce a clock of precise frequency using constant voltage.
- There is one limitation of using Arduino Nano i.e. it doesn't come with a DC power jack, which means you can not supply an external power source through a battery.
- This board doesn't use standard USB for connection with a computer, instead, it comes with Type-B Micro USB.
- The tiny size and breadboard-friendly nature make this device an ideal choice for most applications where the size of the electronic components is of great concern.
- Flash memory is 16KB or 32KB that all depends on the At mega board i.e Atmega168 comes with 16KB of flash memory while Atmega328 comes with a flash memory of 32KB. Flash memory is used for storing code. The 2KB of memory out of total flash memory is used for a bootloader.
- The SRAM memory of 2KB is present in Arduino Nano.
- Arduino Nano has an EEPROM memory of 1KB.
- Each pin on the Nano board comes with a specific function associated with it.

- We can see the analog pins that can be used as an analog to a digital converter, where A4 and A5 pins can also be used for I2C communication.
- Similarly, there are 14 digital pins, out of which 6 pins are used for generating PWM.

WHY ARDUINO NANO

Arduino nano Using the constant voltage, the Arduino Nano is used to **produce a clock of a precise frequency.** The Arduino Nano has a compact size and mini USB cable than the Arduino UNO. We can use Nano instead of UNO because both operate on the microcontroller ATmega328p.The Arduino UNO is also easily available than Nano. It is considered as the standard board available in the market, which is easy for use for the starters or beginners. The Nano is available in PDIP (Plastic Dual - Inline Package), while Arduino UNO is available in TQFP (Plastic Quad Flat Pack).The Arduino UNO includes 6 Analog pin inputs, 14 digital pins, a USB connector, a power jack, and an ICSP (In-Circuit Serial Programming) header. The Arduino Nano includes an I/O pin set of 14 digital pins and 8 Analog pins. It also includes 6 Power pins and 2 Reset pins. The Arduino Nano is a small Arduino board based on ATmega328P or ATmega628 Microcontroller. The connectivity is the same as the Nano board is defined as a sustainable, small, consistent, and flexible board. It is small in size compared to the UNO board. The Arduino Nano is organized using the Arduino (IDE), which can run on various platforms. Here, IDE stands for Integrated Development Environment. The devices required to start our projects using the Arduino Nano board are and mini. The Arduino IDE software must be installed on our respected laptop or desktop. The mini USB transfers the code from the computer to the Arduino Nano board.

ADVANTAGES OF ARDUINO NANO

Arduino is its ready to use structure. As Arduino comes in a complete package form which includes the 5V regulator, a burner, an oscillator, a micro-controller, serial communication interface, LED and headers for the connections. You don’t have to think about programmer connections for programming or any other interface. Just plug it into USB port of your computer and that’s it. Your revolutionary idea is going to change the world after just few words of coding. During coding of Arduino, you will notice some functions which make the life so easy. Another advantage of Arduino is its automatic unit conversion capability. You can say that during debugging you don’t have to worry about the units conversions. Just use your all force on the main parts of your projects. You don’t have to worry about side problems. If you want to measure voltage using ATmega8 micro-controller and want to display the output on computer screen then you have to go through the whole process. The process will start from

learning the ADC’s of micro-controller for measurement, went through the learning of serial communication for display and will end at USB – Serial converters.

FEATURES OF ARDUINO

The Arduino board is designed in such a way that it is very easy for beginners to get started with microcontrollers. This board especially is breadboard friendly, and that's why it is very easy to handle the connections. Let’s start with powering the Board.

Powering you Arduino Nano:

There are total three ways by which you can power your Nano.

USB Jack: Connect the mini USB jack to a phone charger or computer through a cable and it will draw power required for the board to function

Vin Pin: The Vin pin can be supplied with an unregulated 6-12V to power the board. The on-board voltage regulator regulates it to +5V.

+5V Pin: If you have a regulated +5V supply then you can directly provide this o the +5V pin of the Arduino.

Input/output:

There are total 14 digital Pins and 8 Analog pins on your Nano board. The digital pins can be used to interface sensors by using them as input pins or drive loads by using them as output pins. A simple function like pinMode() and digitalWrite() can be used to control their operation. The operating voltage is 0V and 5V for digital pins. The analog pins can measure analog voltage from 0V to 5V using any of the 8 Analog pins using a simple function like analogRead().

These pins apart from serving their purpose, can also be used for special purposes, which are discussed below:

Serial Pins 0 (Rx) and 1 (Tx): Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.

External Interrupt Pins 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.

PWM Pins 3, 5, 6, 9 and 11: These pins provide an 8-bit PWM output by using **Analog Write()** function.

SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK): These pins are used for SPI communication.

In-built LED Pin 13: This pin is connected with a built-in LED. When pin 13 is HIGH – LED is on and when pin 13 is LOW, it is off.

I2C A4 (SDA) and A5 (SCA): Used for IIC communication using Wire library.

AREF: Used to provide reference voltage for Analog inputs with **Analog Reference()** function.

Reset Pin: Making this pin LOW, resets the microcontroller.

SPECIFICATIONS OF ARDUINO:

Microcontroller	**ATmega328**
Architecture	AVR
Operating Voltage	5 V
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog IN Pins	8
EEPROM	1 KB
DC Current per I/O Pins	40 mA (I/O Pins)
Input Voltage	7-12V
Digital I/O Pins	22 (6 of which are PWM)
PWM Output	6
Power Consumption	19 mA
PCB Size	18 x 45 mm
Weight	7 g
Product Code	A000005

INTRODUCTION TO RTC DS3231

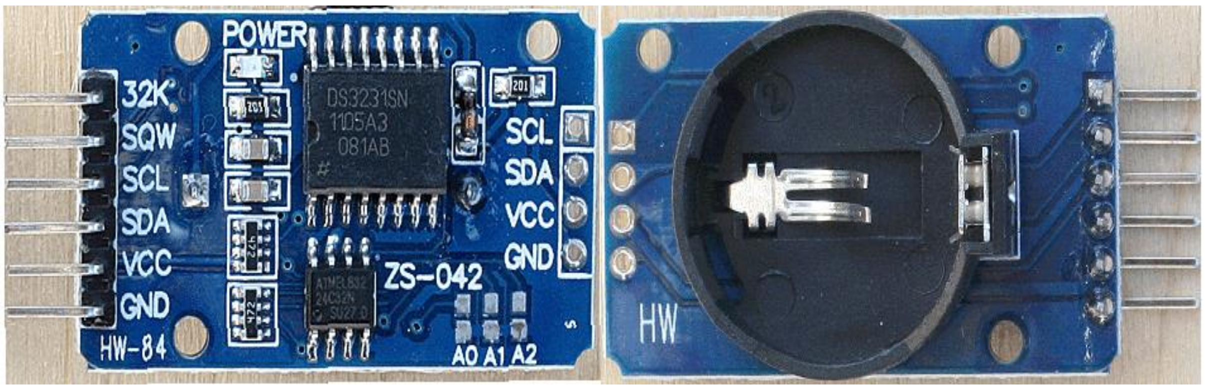

Fig 3: Real Time Clock DS3231

The DS3231 RTC module is a time tracking device that gives the current time and date. The word RTC is meant **R**eal **T**ime **C**lock. The RTC module made of clock chip DS3231. This module is generally used in computers, laptops, mobiles, embedded system applications devices, etc. to provide time and date. RTC module works on the I2C protocol. The module provides details such as second, minute, hour, day of the week, day of the month, month, and year including correction for leap year. One more interesting thing It can operate either in 12 Hour or in 24 Hour format. It's can be used in projects like containing data-logging, clock-building, time stamping, timers, and alarms. The bottom side of the module has a battery holder for a 20mm 3V lithium coin cell. Any CR2032 battery can fit in the battery holder. The DS3231 incorporates a battery input and maintains accurate timekeeping when the main power to the device is interrupted.The built-in power-sense circuit continuously monitors the status of the input (VCC) power supply to detect power failures and automatically switches to the backup supply. So, it can continue to maintain the time and date while device power is incorrect. The DS3231 chip is a low-cost accurate Real Time Clock IC with communication over I2C Interface. The DS3231 is 16 pin IC, Although it needs only 8 pins off the available 16 pins to function. The IC is operated on low power consumption when it operated on a battery then it consumes less than 500 nA power. It manages all timekeeping functions. This Chip clock/calendar can provide seconds, minutes, hours, day, date, month, and year information. This IC clock operates in the format of either the 24-hour or 12-hour with AM/PM indicator. The end of the month date is automatically adjusted the months which has less than 31 days. It can corrections for leap year. The DS3231 is driven by a 32kHz temperature compensated crystal oscillator (TCXO). It’s highly immune to external temperature changes. So the external temperature can not affect the oscillation frequency of this crystal. Temperature Compensated Crystal Oscillator(TCXO) is packaged inside the RTC chip, making the whole package bulky. Also a temperature sensor inside the IC package.

Which compensates the frequency changes by adding or removing clock ticks so that the timekeeping stays on track.That’s the reason Temperature Compensated Crystal Oscillator provides a stable and accurate reference clock and maintains the RTC to within ±2 minutes per year accuracy. RTC module also comes with a 32 bytes Atmel 24C32 EEPROM chip which having limited read-write cycles. It can be used to save settings or clock/calendar. It uses the I2C interface for communication and shares the same I2C bus as DS3231. We can be changed easily the EEPROM I2C address using the three A0, A1, and A2 solder jumpers. Each one of these is used to hardcode in the address and we can set the address by shorting the jumper in the proper way.

According to the 24C32 EEPROM chip’s datasheet, these 3 bits are placed at the end of the 7-bit I2C address, just before the Read/Write bit. These 3 address inputs can take 2 states, either HIGH/LOW. So, we can create 8 (23) different combinations(addresses).The address set by shorting the jumper. The DS3231 is a extremely accurate I^{2}C real- time clock **(**RTC) .

The DS3231 has an internal temperature-compensated crystal oscillator (TCXO) which is not affected by temperature and therefore has higher accuracy so that it can be accurate to a few minutes per year.In this blog, Apogeeweb will briefly introduce the basic information of DS3231**,** as well as the pins and functions of DS3231. Most RTCs use an external 32kHz timing crystal**,** which is used to keep time with low current draw. And that's all right and good, but those crystals have a slight drift, particularly when the temperature changes (the temperature changes the oscillation frequency quite slightly, but it adds up!) This RTC is in a beefy package because the crystal is within the chip! And there's a temperature sensor right next to the integrated crystal. This sensor compensates for shifts in frequency by adding or removing clock ticks so that timekeeping remains on track.

8×32 DOT MATRIX

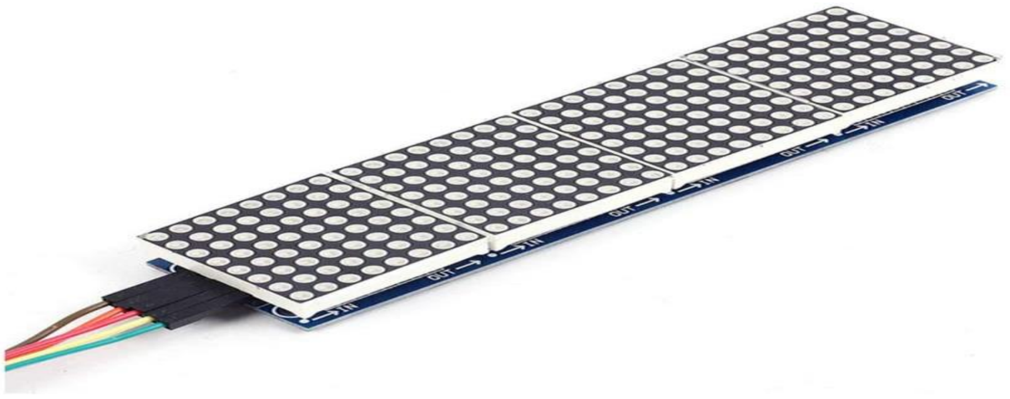

Fig 4: 8×32 Dot matrix

A dot matrix is a 2-dimensional patterned array, used to represent characters, symbols and images. Most types of modern technology use dot matrices for display of information, including mobile phones, televisions, and printers. The system is also used in textiles with sewing, knitting and weaving.

An alternate form of information display using lines and curves is known as a vector display, was used with early computing devices such as air traffic control radar displays and pen-based plotters but is no longer used. Electronic vector displays were typically monochrome only, and either leave the interiors of closed vector shapes unfilled, or perform slow, time-consuming and often non-uniform shape-filling, as on pen-based plotters.

In printers, the dots are usually the darkened areas of the paper. In displays, the dots may light up, as in an LED, CRT, or plasma display, or darken, as in an LCD.

JUMPER WIRES

Fig 3 : Jumper wires

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn’t get much more basic than jumper wires. Though jumper wires come in a variety of colors, the colors don’t actually mean anything. This means that a red jumper wire is technically the same as a black one. But the colors can be used to your advantage in order to differentiate between types of connections, such as ground or power. While jumper wires are easy and inexpensive to purchase, it can also be a fun task to challenge students to make their own. Doing so requires insulated wire and wire strippers. However, beware that it is important not to nick the wire when stripping off the insulation. Jumper wires typically come in three versions: male-to-male, male-to-female and female-to-female. The difference between each is in the end point of the wire. Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into. Male-to-male jumper wires are the most common and what you likely will use most often. When connecting two ports on a breadboard, a male-to-male wire is what you’ll need.

CIRCUIT DETAILS

CIRCUIT DIAGRAM

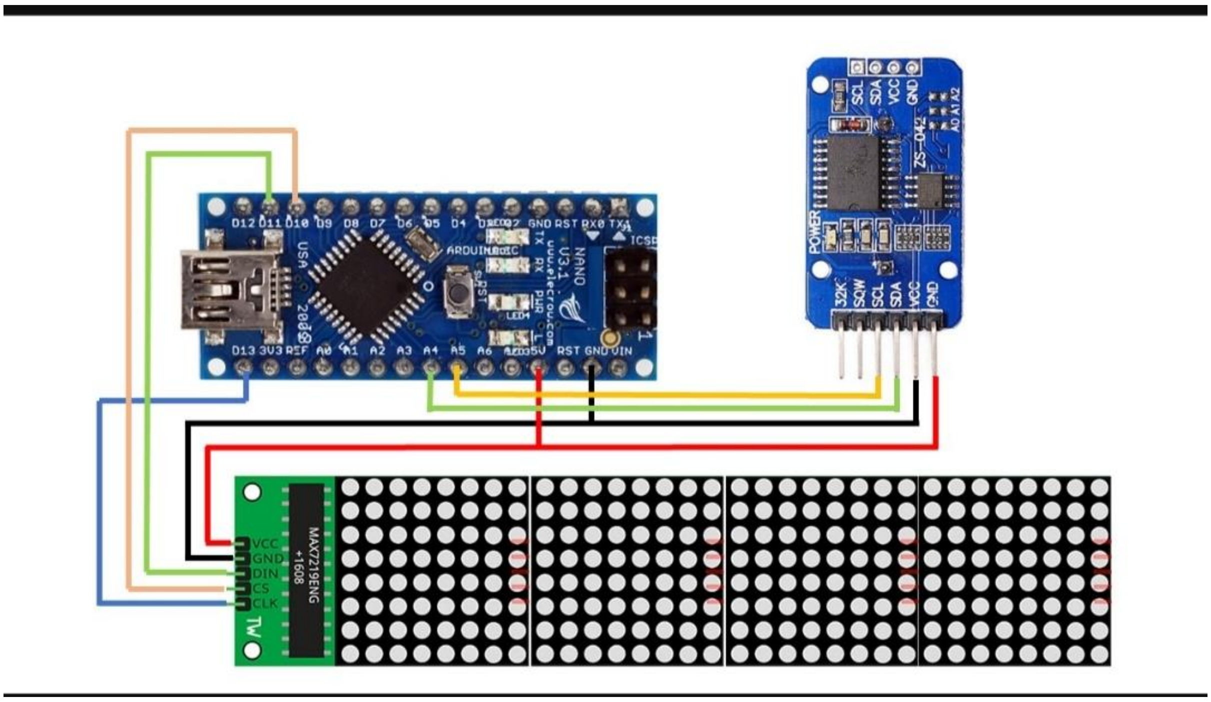

Fig 4: Circuit Diagram of Digital Clock

WORKING PRINCIPLE

- Initially first the code which is written in the embedded c language should be dumped or copied into the component of Arduino NANO before copying or dumped the Arduino should be powered up by the battery or the USB cable may be connected in between the laptop and Arduino NANO
- Then according to the giving code in the Arduino the Real Time Clock will send the signal to the Arduino Nano
- According to the received date by the RTC, it will forward the date to the 8×32 dot matrix.
- 8×32 Dot matrix displays the Time , Date, Day and Temperature.
- As we are using the Real Time Clock the will be updated every time.
- If there is any power lose also it will be update the time and date.

- **ADVANTAGES**

- It is light and compact.

- From your computer's desktop to the screen outside of the local bank, you will be able to see the time clearly and easily using this particular method. The fact that digital clocks are also much easier to see makes them a better choice than their more traditional cousin.

- Digital clock is that there is less chance of it showing the incorrect time. Traditional clocks can sometimes get stuck, where their components and hands just do not move for a variety of reasons. This can tell you the wrong time when you look at the clock and cause you to run late or leave too early for a certain engagement.

- Easy tracking with a digital clock, employee time tracking records will not only be accurate, safe and secure, it will be extremely easy to track number of hours each employee ha logged in.

- Exact time resume- if there is any power failure, there will not be any time display on the clock. But as soon as the electric power resumes, the clock will show the right time automatically.

APPLICATIONS

Because digital clocks can be very small and inexpensive devices that enhance the popularity of product designs, they are often incorporated into all kinds of devices such as cars, radios, televisions, microwave ovens, standard ovens, computers and cell phones. Sometimes their usefulness is disputed: a common complaint is that when time has to be set to Daylight Saving Time, many household clocks have to be readjusted. The incorporation of automatic synchronization by a radio time signal is reducing this problem (see Radio clock). Smart digital clocks, in addition to displaying time, scroll additional information such as weather and notifications.

PROGRAM CODE

```
#include <MD_Parola.h>

#include <MD_MAX72xx.h>

#include <SPI.h>

#include "Font_Data.h"

#include <DS3231.h>

#include <Wire.h>

DS3231 Clock;

bool Century=false;

bool h12;

bool PM;

byte  dd,mm,yyy;

uint16_t  h, m, s;

#define MAX_DEVICES 4

#define HARDWARE_TYPE MD_MAX72XX::FC16_HW

#define CLK_PIN   13

#define DATA_PIN  11

#define CS_PIN    10

MD_Parola P = MD_Parola(HARDWARE_TYPE,CS_PIN, MAX_DEVICES);

#define SPEED_TIME 75

#define PAUSE_TIME  0

#define MAX_MESG  20

char szTime[9];

char szMesg[MAX_MESG+1] = "";
```

```
uint8_t degC[] = { 6, 3, 3, 56, 68, 68, 68 };
uint8_t degF[] = { 6, 3, 3, 124, 20, 20, 4 };
char *mon2str(uint8_t mon, char *psz, uint8_t len)
{
  static const __FlashStringHelper* str[] =
  {
    F("Jan"), F("Feb"), F("Mar"), F("Apr"),
    F("May"), F("Jun"), F("Jul"), F("Aug"),
    F("Sep"), F("Oct"), F("Nov"), F("Dec")
  };
  strncpy_P(psz, (const char PROGMEM *)str[mon-1], len);
  psz[len] = '\0';
 return(psz);
}
char *dow2str(uint8_t code, char *psz, uint8_t len)
{
  static const __FlashStringHelper*  str[] =
  {
  F("Sunday"), F("Monday"), F("Tuesday"),
  F("Wednesday"), F("Thursday"), F("Friday"),
  F("Saturday"), F("Sunday")
  };
 strncpy_P(psz, (const char PROGMEM *)str[code-1], len);
  psz[len] = '\0';
  return(psz);
```

```
}
void getTime(char *psz, bool f = true)
{
  s = Clock.getSecond();
  m = Clock.getMinute();
    sprintf(psz, "%02d%c%02d", h, (f ? ':' : ' '), m);
    if (Clock.getHour(h12,PM)>=13 || Clock.getHour(h12,PM)==0)
  {
    h = Clock.getHour(h12,PM) - 12;
  }
  else
  {
    h = Clock.getHour(h12,PM);
  }
 }
void getDate(char *psz)
{
   char  szBuf[10];
  dd=Clock.getDate();
   mm=Clock.getMonth(Century);
   yyy=Clock.getYear();
   sprintf(psz, "%d %s %04d",dd , mon2str(mm, szBuf, sizeof(szBuf)-1),(yyy + 2000));
}
void setup(void)
{
  P.begin(2);

  P.setInvert(false);

  Wire.begin();

   P.setZone(0,  MAX_DEVICES-4, MAX_DEVICES-1);

   P.setZone(1, MAX_DEVICES-4, MAX_DEVICES-1);

  P.displayZoneText(1, szTime, PA_CENTER, SPEED_TIME, PAUSE_TIME, PA_PRINT, 
PA_NO_EFFECT);

  P.displayZoneText(0, szMesg, PA_CENTER, SPEED_TIME, 0,PA_PRINT , 
PA_NO_EFFECT);

  P.addChar('$', degC);

  P.addChar('&', degF);

}

void loop(void)

{

  static uint32_t lastTime = 0;

  static uint8_t  display = 0;

  static bool flasher = false;

 P.displayAnimate();

  if (P.getZoneStatus(0))

  {

    switch (display)

    {

      case 0:

      P.setPause(0,1000);

      P.setTextEffect(0, PA_MESH, PA_BLINDS);

      display++;
      dtostrf(Clock.getTemperature(), 3, 1, szMesg);
      strcat(szMesg, "$");
            break;
  case 1:
        P.setTextEffect(0, PA_OPENING, PA_GROW_DOWN);
        display++;
        dtostrf((1.8 *Clock.getTemperature() )+32, 3, 1, szMesg);
        strcat(szMesg, "&");
 break;
  case 2:
      P.setFont(0, numeric7Seg);
     P.setTextEffect(0, PA_PRINT, PA_NO_EFFECT);
     P.setPause(0,0);
    if (millis() - lastTime >= 1000)
    {
    lastTime = millis();
   getTime(szMesg, flasher);
    flasher = !flasher;
    }
    if(s==00&& s<=30){
    display++;
P.setTextEffect(0, PA_PRINT, PA_SCROLL_UP);
    }
break; case 3:
        P.setFont(0,nullptr);
        P.setTextEffect(0, PA_SCROLL_LEFT, PA_SCROLL_LEFT);

        display++;

        dow2str(Clock.getDoW()+1, szMesg, MAX_MESG); // Added +1 or +2 to get correct
Day of Week

   break;

      default:

       P.setTextEffect(0, PA_SCROLL_LEFT, PA_SCROLL_LEFT);

        display = 0;

        getDate(szMesg);

        break;

    }

 P.displayReset(0);

  }

}
```

CONCLUSION

- Thus, this is how our main system has been built up.
- Now-a days different pattern of digital clocks are available in the market but most of them are of very price and low quality.
- Most of these cannot provide the time accurately for longer period as those are designed with IC’s like 555 timer.But our designed multipurpose digital clock is accurate because of its Real Time Clock module that keeps time update.
- We have completed this project successfully and have successfully made a 12 hour digital clock.

REFERENCE:

1. "DEPATISnet DEPATISnet-Startseite". depatisnet.dpma.de. Retrieved 2021-11-08.
2. ^ "Home page". JosefPallweber.com. Archived from the original on 2015-10-*01*. Retrieved 2015-11-07.
3. ^ Jump up to:*a* *b* Churm, Thomas M. (November 5, 2013). "A Short History of Digital Clocks and Watches". Alarm Clock Blog. Archived from the original on March 1, 2016. Retrieved 2016-02-28. (PDF). German Patent and Trademark Office (in German). Retrieved 2015-11-07.
4. ^ Imperial Patent Office (October 27, 1890). "Patent No. 54093"
5. ^ "Is digital more precise?". The German Clock Museum. April 2015. Archived from the original on *2015-09-23*. Retrieved 2015-11-07.
6. ^ US2768332A, Protzmann, Donald E.; Phaneuf, Edgar A. & Doyle, Malcolm G., "Timing device", issued 1956-10-23
7. ^ "The History of the Digital Watch". h2g2. *April 30, 2003*. Archived from the original on November 6, 2015. Retrieved 2015-11-07.
8. ^ "Radio Controlled LED Alarm Clock Instruction Manual — SM2442" (PDF). Zeon Ltd. Archived from the original (DOC) on 2016-03-03. Retrieved 2015-11-07.