

1 single digit Nixie clock

by Jānis Alnis 2014-2016

Young people are fascinated by Nixies. They have not seen such neon lights in the present solid-state era. IN-18 is the largest neon digit indicator produced during USSR times that was used in scales on food markets and in scientific equipment.

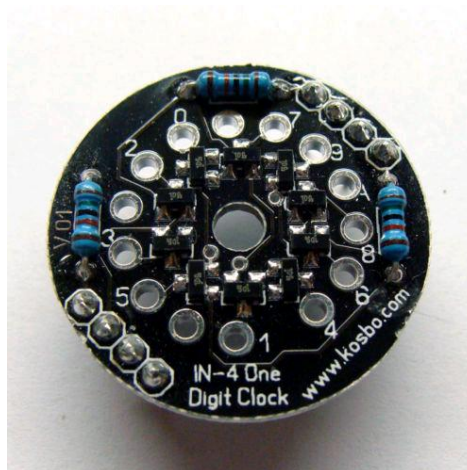
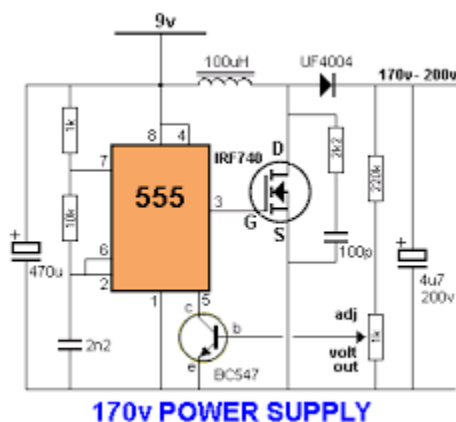
Below is described how to make a single digit Nixie clock that shows time in the following order:

tens of hours blank hours ones blank tens of minutes blank minutes ones longer blank.

<https://youtu.be/RFq13pH0-NU>

Single digit simplifies the boring soldering task as only one nixie needs to be connected instead of 4 or 6. Single digit add dynamics to the clock and also makes clock cheaper. A single IN-18 tube on E-bay costs around 30 EUR. Any other Nixie you get hold of can be used with the circuit described here.

Indicator needs 170...200V. **Lifetime expectancy is around a half a year of continuous usage. Due to cathode oxidation parts of some digits get dim and stop shining.** Increasing voltage helps for some time but eventually one needs to replace with a new tube. HV circuit uses a NE555 timer, FET and coil. Circuits available on www. Photo camera flash board could be possibly used (unconfirmed).



Digits 0...9 are switched on/off using **K155ID1** chip (also from old USSR stock). 10 HV SMD transistors can also be used and that is actually a more compact solution.

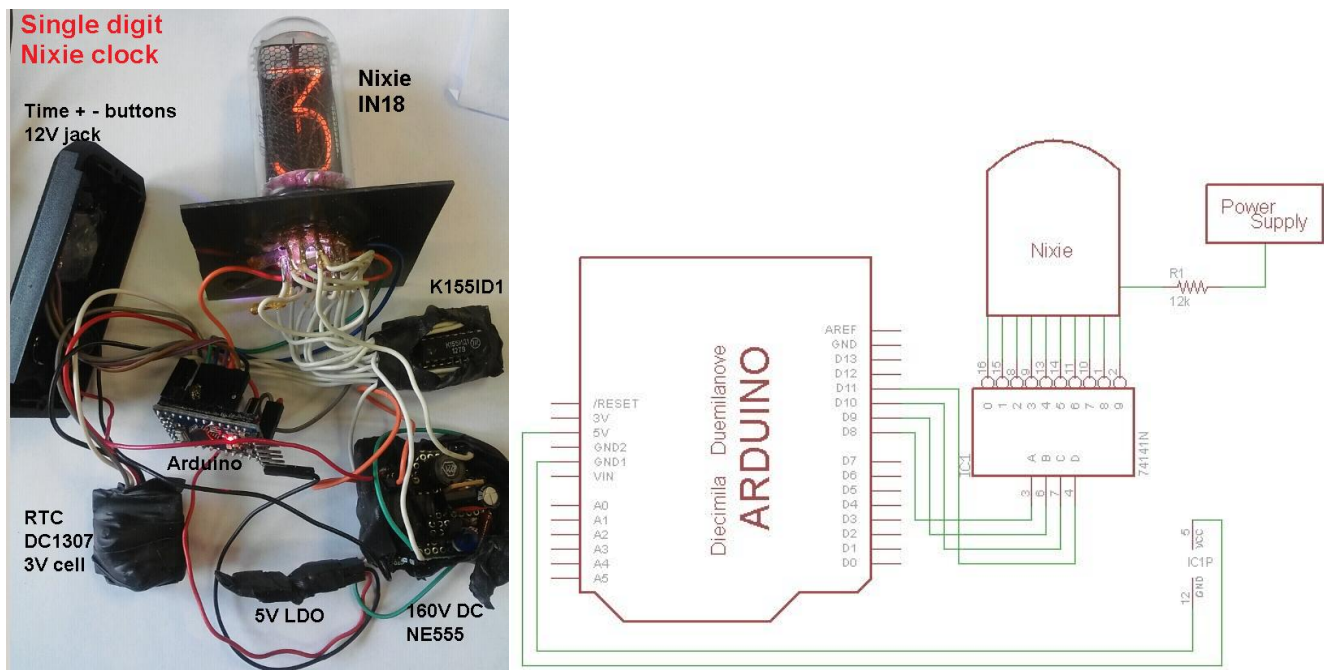
Atmega328 on a Arduino board controls the circuit.

Time is kept precisely by a RTC chip DS1317 backed up with a 3V lithium cell. But it is better to use some other chip breakout board from E-bay with a built-in quartz as it is factory trimmed.

Time is corrected by two buttons „+” and „-”.

A RGB LED give additional visual effects. Color is chosen arbitrary for each time value.

LM7805 reduces voltage from 12V to 5V to drive the Arduino and K155ID1 chip.



Any other Nixie you get hold of can be used with the circuit described in the attached file. Arduino program code is also attached that works with a library ArduinoRTC_DS1307.

First you need to assemble HV module and check that you can light Nixie digits.

Independent task is to learn to read our RTC chip using Arduino.

Final task is to bring everything together.

Literature:

Single digit Nixie clock: <http://www.lucadentell.it/en/2011/12/08/orologio-nixie-2-il-driver/>

```

//Single Nixie tube clock uses Arduino, DS1307 RTC, 12V to 180V with NE555, K155ID1 Nixie driver.
//Sequencially shows hour_tens, hour_ones, minutes_tens, minutes_ones

// code for Arduino0022 Get DS1307 library and paste in Arduino Libraries folder.

#include <WProgram.h>
#include <Wire.h>
#include <DS1307.h>

int rtc[7];
byte rr[7];
int ledPin = 13;

void setup()
{
pinMode(3, OUTPUT); pinMode(5, OUTPUT); pinMode(6, OUTPUT); // rgb LEDs for illumination
pinMode(2, INPUT); pinMode(4, INPUT); // buttons to adjust time by + and -1 minute

digitalWrite(2, HIGH);
digitalWrite(4, HIGH);

pinMode(8, OUTPUT); pinMode(9, OUTPUT); pinMode(10, OUTPUT); pinMode(11, OUTPUT); // Nixie driver K155ID1

DDRC |= _BV(2) | _BV(3); // POWER:Vcc Gnd
PORTC |= _BV(3); // VCC PINC3
pinMode(ledPin, OUTPUT);
Serial.begin(9600);
// RTC.get(rtc,true);
// if(rtc[6]<12){
//   RTC.stop();
//   RTC.set(DS1307_SEC,0);
//   RTC.set(DS1307_MIN,0);
//   RTC.set(DS1307_HR,8);
//   RTC.set(DS1307_DOW,3);
//   RTC.set(DS1307_DATE,10);
//   RTC.set(DS1307_MTH,12);
//   RTC.set(DS1307_YR,14);
//   RTC.start();
// }
//RTC.SetOutput(LOW);
//RTC.SetOutput(HIGH);
// RTC.SetOutput(DS1307_SQW1HZ);
//RTC.SetOutput(DS1307_SQW4KHZ);
//RTC.SetOutput(DS1307_SQW8KHZ);
RTC.SetOutput(DS1307_SQW32KHZ); // useful to check with frequency counter and trim quartz with variable capacitor.
}

void loop()
{
int i, mm, hh, m1, m2, h1, h2;
RTC.get(rtc,true);
for(i=0; i<7; i++) {Serial.print(rtc[i]); Serial.print(" ");} Serial.println();
mm=rtc[1]; hh=rtc[2];

PORTB =hh/10; delay(500); // show number of tens of hours
PORTB =10; delay(100); // switch Nixie off for some time
PORTB =hh%10; delay(500); // show number of hours
PORTB =10; delay(500);
PORTB =mm/10; delay(500); // show number of tens of minutes
PORTB =10; delay(100);
PORTB =mm%10; delay(500); // show number of minutes
PORTB =10; delay(2000);

analogWrite(3, random(50)); analogWrite(5, random(50)); analogWrite(6, random(50)); // RGB led at random color for fun illumination

while (digitalRead(4)==0) { // if button pressed for time adjustment
mm=mm+1;

```

```

if (mm == 60){ mm=0; hh=hh+1; if (hh == 24){hh=0;}};
RTC.stop();
RTC.set(DS1307_MIN,mm);
RTC.set(DS1307_HR,hh);
RTC.start();
PORTB=mm%10; delay(200); // show number of minutes;
}

```

```

while (digitalRead(2)==0) {
mm=mm-1;
if (mm==-1){mm=59; hh=hh-1; if (hh==-1){hh=23;}}
RTC.stop();
RTC.set(DS1307_MIN,mm);
RTC.set(DS1307_HR,hh);
RTC.start();
PORTB=mm%10; delay(200); // show number of minutes;
}

```

```

if (Serial.available()) { // This set of functions allows the user to change the date and time
delay (100); Serial.println("Would you like to set the date and time now? Y/N");
Serial.flush(); delay(2000);
if (Serial.read() == 'y' || Serial.read() == 'Y')
{ Serial.read(); setTime(); Serial.print("The current date and time is now: "); }
}
}

```

```

byte readByte() {
while (!Serial.available()) delay(10);
byte reading = 0;
byte incomingByte = Serial.read();
while (incomingByte != '\n') {
if (incomingByte >= '0' && incomingByte <= '9')
reading = reading * 10 + (incomingByte - '0');
else;
incomingByte = Serial.read();
}
Serial.flush();
return reading;
}

```

// This set of codes is allows input of data

```

void setTime() {
Serial.flush();
Serial.print("Please enter the current year, 00-99. - ");
int year = readByte();
Serial.println(year);

```

```

Serial.print("Please enter the current month, 1-12. - ");
int month = readByte();
Serial.println(month);

```

```

Serial.print("Please enter the current day of the month, 1-31. - ");
int monthday = readByte();
Serial.println(monthday);

```

```

Serial.println("Please enter the current day of the week, 1-7.");
Serial.print("1 Sun | 2 Mon | 3 Tues | 4 Weds | 5 Thu | 6 Fri | 7 Sat - ");
int weekday = readByte();
Serial.println(weekday);

```

```

Serial.print("Please enter the current hour in 24hr format, 0-23. - ");
int hour = readByte();

```

```
Serial.println(hour);

Serial.print("Please enter the current minute, 0-59. - ");
int minute = readByte();
Serial.println(minute);
int second = 0;
Serial.println("The data has been entered.");

RTC.stop();
RTC.set(DS1307_SEC,second);
RTC.set(DS1307_MIN,minute);
RTC.set(DS1307_HR,hour);
RTC.set(DS1307_DOW,weekday);
RTC.set(DS1307_DATE,monthday);
RTC.set(DS1307_MTH,month);
RTC.set(DS1307_YR,year-2);
RTC.start();
}
```