

```

#include <Wire.h>
#include <RTC.h>

static DS1307 RTC;
#include "SevSeg.h"
SevSeg sevseg; //Instantiate a seven segment controller object

unsigned long set_times = millis(), timer = millis(), blink_time = millis(),
up_times = millis();
int set_things = A3, up_things = A2, which_one = 0, UP=0;
bool bt = true;
String hr, mi, dy, mo, yr;

String SDIGIT;

void setup() {
    RTC.begin();

    byte numDigits = 8;
    byte digitPins[] = {11, 10, 9, 8, 12, 13, A0, A1};
    byte segmentPins[] = {1, 2, 5, 6, 3, 4, 7};
    bool resistorsOnSegments = false; // 'false' means resistors are on digit
pins
    byte hardwareConfig = COMMON_CATHODE; // See README.md for options
    bool updateWithDelays = false; // Default 'false' is Recommended
    bool leadingZeros = false; // Use 'true' if you'd like to keep the leading
zeros
    bool disableDecPoint = true; // Use 'true' if your decimal point doesn't
exist or isn't connected

    sevseg.begin(hardwareConfig, numDigits, digitPins, segmentPins,
resistorsOnSegments,
                updateWithDelays, leadingZeros, disableDecPoint);
    sevseg.setBrightness(90);

    pinMode(set_things, INPUT);
    pinMode(up_things, INPUT);
}

void loop() {

    if (millis() > timer) {
        timer = millis() + 1000;

```

```

getAndDisplay();
SDIGIT = hr + mi + dy + mo;
sevseg.setNumber(SDIGIT.toInt());
}
sevseg.refreshDisplay();

if (digitalRead(set_things) == HIGH && which_one != 0) {
    while (digitalRead(set_things) == HIGH) {
        sevseg.refreshDisplay();
    }
    which_one++;
    if (which_one >= 6) {
        which_one = 0;
    }
}
else if (millis() > up_times + 5000 && which_one != 0) {
    which_one = 0;
}
set_times = millis();
blink_time = millis();
while (digitalRead(set_things) == HIGH) {
    if (millis() > timer) {
        timer = millis() + 1000;
        getAndDisplay();
    }
    if (millis() > set_times + 2000 && which_one == 0) {
        which_one = 1;
    }
    else if (which_one != 0) {
        which_one = 0;
    }
    if (which_one == 1 && millis() > blink_time && bt == true) {
        blink_time = millis() + 100;
        bt = false;
        SDIGIT = mi + dy + mo;
        sevseg.setNumber(SDIGIT.toInt());
    }
    else if (which_one == 1 && millis() > blink_time && bt == false) {
        blink_time = millis() + 100;
        bt = true;
        SDIGIT = hr + mi + dy + mo;
        sevseg.setNumber(SDIGIT.toInt());
    }
    sevseg.refreshDisplay();
}

```

```

}
if (digitalRead(up_things) == HIGH) {
    while (digitalRead(up_things) == HIGH) {
        sevseg.refreshDisplay();
    }
    up_times = millis()+5000;
    if (which_one == 1) {
        UP=RTC.getHours()+1;
        if(UP>23){
            UP=00;
        }
        RTC.setHours(UP);
    }
    else if (which_one == 2) {
        UP=RTC.getMinutes() + 1;
        if(UP>59){
            UP=00;
        }
        RTC.setMinutes(UP);
    }
    else if (which_one == 3) {
        UP=RTC.getDay() + 1;
        if(UP>31){
            UP=1;
        }
        RTC.setDay(UP);
    }
    else if (which_one == 4) {
        UP=RTC.getMonth() + 1;
        if(UP>12){
            UP=1;
        }
        RTC.setMonth(UP);
    }
    else if (which_one == 5) {
        long UPP=RTC.getYear() + 1;
        if(UPP>2060){
            UPP=1900;
        }
        RTC.setYear(UPP);
    }
}
}
}

```

```
void getAndDisplay() {
    hr = String (RTC.getHours());
    mi = String (RTC.getMinutes());
    dy = String (RTC.getDay());
    mo = String (RTC.getMonth());
    yr = String (RTC.getYear());
    String yyr = yr.substring (2, 3);
    String yyr2 = yr.substring (3, 4);
    yr = yyr + yyr2;
    if (hr == "0") {
        hr = "00";
    }
    else if (hr.toInt() < 10) {
        hr = "0" + hr;
    }
    if (mi == "0") {
        mi = "00";
    }
    else if (mi.toInt() < 10) {
        mi = "0" + mi;
    }
    if (dy == "0") {
        dy = "00";
    }
    else if (dy.toInt() < 10) {
        dy = "0" + dy;
    }
    if (mo == "0") {
        mo = "00";
    }
    else if (mo.toInt() < 10) {
        mo = "0" + mo;
    }
    if (yr == "0") {
        yr = "00";
    }
}
```