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

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

#include <dhtnew.h>

DHTNEW mySensor(0);

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 dh, hr, mi;

String SDIGIT;

void setup() {
  RTC.begin();

  byte numDigits = 8;
  byte digitPins[] = {8, 9, 10, 11, 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 + dh + "C";
     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 >= 3) {
      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() > set_times + 2000 && which_one == 0) {
      which_one = 1;
    }
    else if (which_one != 0) {
      which_one = 0;
    }
    if (millis() > timer) {
      timer = millis() + 1000;
      getAndDisplay();
    }
    if (which_one == 1 && millis() > blink_time && bt == true) {
      blink_time = millis() + 100;
      bt = false;
      SDIGIT = mi + dh + "C";
       sevseg.setNumber(SDIGIT.toInt());
    }
    else if (which_one == 1 && millis() > blink_time && bt == false) {
      blink_time = millis() + 100;
      bt = true;
```

```
      SDIGIT = hr + mi + dh + "C";
       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);
    }
  }
}


void getAndDisplay() {
  dh = String ((mySensor.getTemperature(), 1));
  hr = String (RTC.getHours());
  mi = String (RTC.getMinutes());
  if (dh == "0"||isnan((mySensor.getTemperature(), 1))) {
    dh = "0000";
  }
  else if (dh.toInt() < 10) {
    dh = "000" + dh;
  }
  else if (dh.toInt() < 100) {
    dh = "00" + dh;
  }
  else if (dh.toInt() < 1000) {
    dh = "0" + dh;
  }
```

```
  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;
  }
}
```