```
//fan_new2
//9.5.16
//Credit  Arduino Playground author for temp102 routine
//Otherwise developed by Julian Rogers
//This is a controller for an under-radiator fan unit.
//The system has a room temperature sensor which works as a thermostat.
//A temperature sensor on the radiator turns off the fan when the radiator is cold.
//Two thermostatic temperatures can be preset by potentiometers inside the unit's cas.e
//A switch selects either or turns off the fans.

const int waterMin = 320;  // min water temp for fans to operate
const byte apinHigh = 1;  //analog pin for high temp preset
const byte apinLow = 0;   // analog pin for low temp preset
const byte pwmPin = 3;    // pwm pin drives fan motors
const byte swPin1 = 8;    // sets high temp
const byte swPin2 = 9;    // sets low temp
float timing = 60000;     // time fans on before temp sensors & switch checked
int setTemp;              // value of thermostatic temp from preset pots
byte pos;                 //switch position: 0 = off, 1 = low, 2 = high

#include <Wire.h>
#define TMP102_I2C_ADDRESS1 0x48  // I2C address TMP102 A0 to GND (0x48 = 72 = 1001000 for GND, 73 for vcc)
#define TMP102_I2C_ADDRESS2 0x49  // I2C address TMP102 A0 to vcc (0x48 = 72 = 1001000 for GND, 73 for vcc)

void setup() {
 pinMode(pwmPin, OUTPUT);
 pinMode(swPin1, INPUT_PULLUP);  //for the temperature selection switch
 pinMode(swPin2, INPUT_PULLUP);  //ditto
 analogWrite(pwmPin, 0);
 Serial.begin(9600);
 Wire.begin();
}

////////////////////////////////////////////////////////////////////////////////
// function to read temperature
int getTemp102(byte tmp102Add){
  byte firstbyte, secondbyte; //these are the bytes we read from the TMP102 temperature registers
  int val; /* an int is capable of storing two bytes, this is where we "chuck" the two bytes together. */

  float convertedtemp; /* We then need to multiply our two bytes by a scaling factor, mentioned in the datasheet.
*/

  //float correctedtemp;
  // The sensor overreads? I don't think it does!


  /* Reset the register pointer (by default it is ready to read temperatures)
   You can alter it to a writeable register and alter some of the configuration -
   the sensor is capable of alerting you if the temperature is above or below a specified threshold. */

  Wire.beginTransmission(tmp102Add); // start talking to sensor
  Wire.write(0x00);
  Wire.endTransmission();
  Wire.requestFrom(tmp102Add, 2);
  Wire.endTransmission();

  firstbyte      = (Wire.read());
  /*read the TMP102 datasheet - here we read one byte from
   each of the temperature registers on the TMP102*/
  secondbyte     = (Wire.read());
  /*The first byte contains the most significant bits, and
   the second the less significant */
  val = firstbyte;
  if ((firstbyte & 0x80) > 0) {
    val |= 0x0F00;
  }
  val <<= 4;
  /* MSB */
  val |= (secondbyte >> 4);
  // LSB is ORed into the second 4 bits of our byte.

  convertedtemp = val*0.625; // temp x 10

  int temp = (int)convertedtemp;
  return temp;
}

////////////////////////////////////////////////////////////////////////////////

void loop() {
  // get preset temp values
  int tmp1 = analogRead(apinHigh);
  int tmp2 = analogRead(apinLow);

  //map to range 16 - 24 deg c.
  tmp1 = map(tmp1,0,1023,160,240);
  tmp2 = map(tmp2,0,1023,160,240);
  Serial.print("high set temp  ");
  Serial.println(tmp1);
  Serial.print("low set temp  ");
  Serial.println(tmp2);
```

```
  //read temp sensors
  int airTemp = getTemp102(TMP102_I2C_ADDRESS2);
  int radTemp = getTemp102(TMP102_I2C_ADDRESS1);
  Serial.print("rad temp  ");
  Serial.println(radTemp);
  if(radTemp == 0){
    radTemp = waterMin;
    Serial.println("rad sensor problem! ");
  }

  Serial.print("air temp  ");
  Serial.println(airTemp);
  if(airTemp == 0){
    airTemp = tmp1;
    Serial.println("air sensor problem! ");
  }


  // check switch position
  if(digitalRead(swPin1) == LOW){
    setTemp = tmp1;
    pos = 2;
  }
  if(digitalRead(swPin2) == LOW){
    setTemp = tmp2;
    pos = 1;
  }
  if(digitalRead(swPin1) == HIGH && digitalRead(swPin2) == HIGH){
    pos = 0;
  }
  Serial.print("switch: (0 is off, 1 is low, 2 is high)  ");
  Serial.println(pos);

   //if (pos && airTemp < setTemp){
   if (pos && airTemp <= setTemp && radTemp >= waterMin){

      // ramp up the fan speed gradually - this prevents vibrations setting in from cheap fans.
      for(int i = 220; i < 256; i++){

      analogWrite(pwmPin, i);
      delay(200);
    }

    delay(timing); //wait for about a minute before the next cycle of temp measurement etc

   }

   analogWrite(pwmPin, 0); // turn off the fans briefly to kill any undue vibration

   delay(1000);

}
```