IR Signal and Temperature Controlled Fan

ECEN 106 – Computer Systems Final Project
Brodric Young, December 9, 2023

Explanation: What my project does:

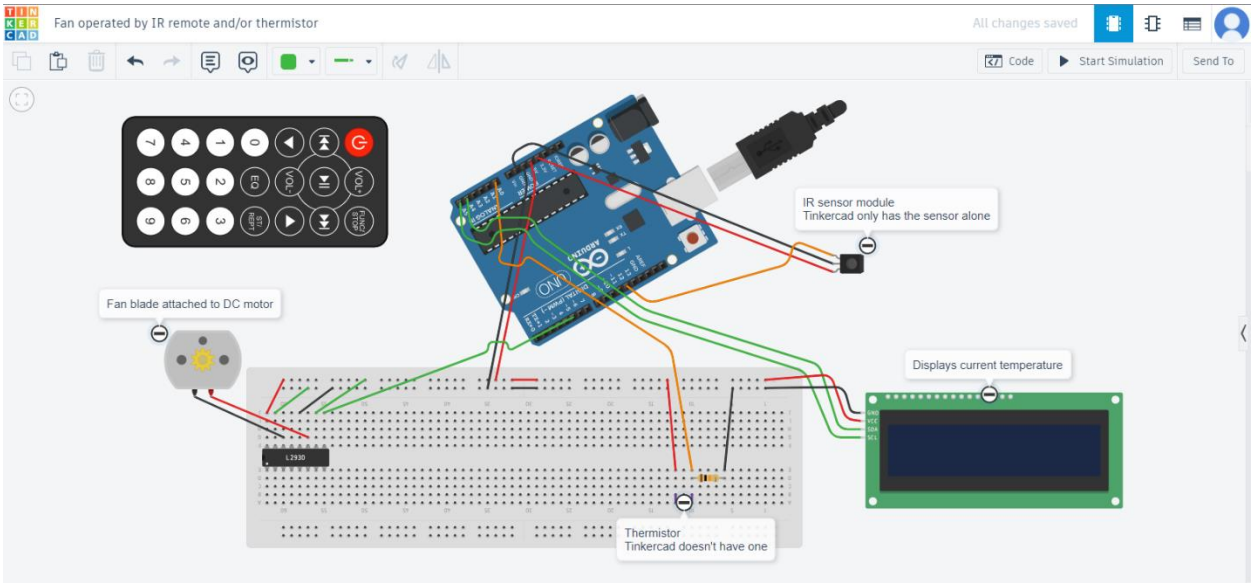
My project is simply just a fan operated by a DC motor along with a thermometer. There's an IR remote that you can press a button on that will turn on or turn off the motor regardless of the temperature which is displayed on an LCD. The motor will also automatically turn on when the temperature passes a certain limit, in this case I've set it to 70 degrees Fahrenheit, and turn off when the temperature goes below that limit. It also will print to the serial monitor any changes in the output state of the motor as well as what changed it (IR signal or temperature).

Explanation: How my project works:

When you press a button on the IR remote, it emits an infrared signal which the IR sensor module picks up and the software will check what the output state of the motor is (high or low) and toggle it the other regardless of what temperature it is.

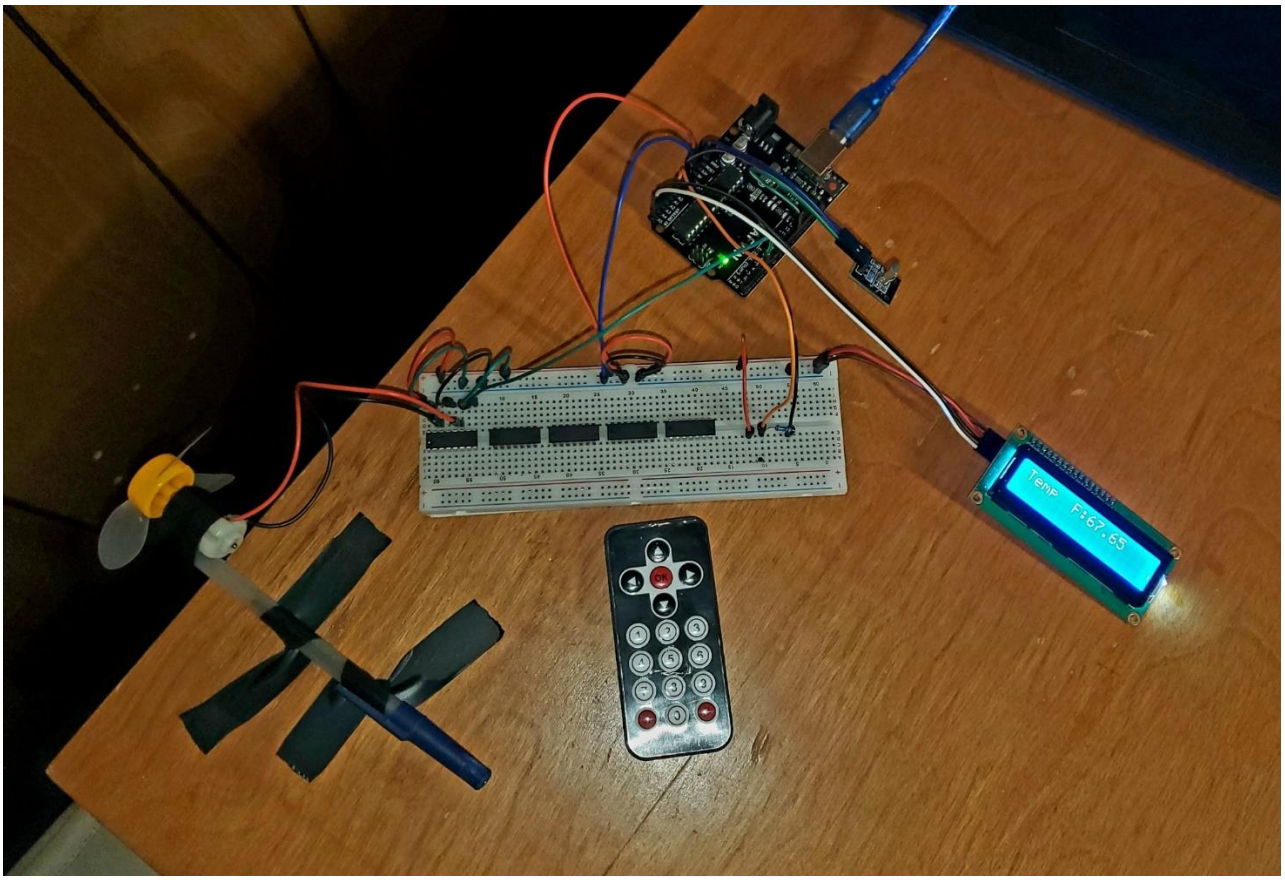
The thermistor senses the temperature and then the software will read that data and display it to the LCD. Then it will check if the data read is above a specified number and if it is then it will activate the motor. If it is activated due to the temperature, you can always turn it off with the IR remote and it will stay off. When the temperature goes back down below that specified number then it also will deactivate the motor.

Hardware: Wiring diagram



I created this wiring diagram myself.

Hardware: Photo



Software: Arduino sketch

```
#include <IRremote.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define RECEIVER_PIN 12   // Signal Pin of IR receiver to Arduino Digital Pin 12
#define OUTPUT_PIN 13     // Output Pin connected to your breadboard

#define ENABLE_PIN 5
#define DIRA_PIN 3
#define DIRB_PIN 4

#define DHTPIN A0 // Analog pin for temperature sensor

IRrecv irrecv(RECEIVER_PIN);
decode_results results;
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Set the LCD address to 0x27 for a 16
chars and 2 line display
bool hot = HIGH;

void setup() {
  Serial.begin(9600);
  pinMode(OUTPUT_PIN, OUTPUT);
  irrecv.enableIRIn(); // Enable the IR receiver
  pinMode(ENABLE_PIN, OUTPUT);
  pinMode(DIRA_PIN, OUTPUT);
  pinMode(DIRB_PIN, OUTPUT);
  lcd.init();                      // Initialize the LCD
  lcd.backlight();
}

void loop() {
  handleIRTask();     // IR receiver task
  displayTempTask();  // Temperature sensor task
  runMotorTask();     // DC motor task
}

void handleIRTask() {
  if (irrecv.decode(&results)) {  // If an IR signal is received
    bool outputState = !digitalRead(OUTPUT_PIN); // Toggle the output state
    digitalWrite(OUTPUT_PIN, outputState); // Apply the new output state to the
output pin
    Serial.print("IR Signal Received. Output state changed to ");
    Serial.println(outputState == HIGH ? "HIGH" : "LOW");
```

```
    irrecv.resume(); // Receive the next value
  }
}


void displayTempTask() {
  int tempReading = analogRead(DHTPIN);
  double tempK = log(10000.0 * ((1024.0 / tempReading - 1)));
  tempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * tempK * tempK )) *
tempK );       //  Temp Kelvin
  float tempC = tempK - 273.15;            // Convert Kelvin to Celsius
  float tempF = (tempC * 9.0) / 5.0 + 32.0; // Convert Celsius to Fahrenheit

  lcd.setCursor(0, 0);
  lcd.print("Temp  F:");
  lcd.setCursor(8, 0);
  lcd.print(tempF);

  delay(500); // Adjust as needed
}


void runMotorTask() {
  int tempReading = analogRead(DHTPIN);
  double tempK = log(10000.0 * ((1024.0 / tempReading - 1)));
  tempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * tempK * tempK )) *
tempK );       //  Temp Kelvin
  float tempC = tempK - 273.15;            // Convert Kelvin to Celsius
  float tempF = (tempC * 9.0) / 5.0 + 32.0; // Convert Celsius to Fahrenheit

  int iteration = 0;
  float test_temp = 80;
  if (tempF > test_temp) {
    while (hot == HIGH) {
      iteration += 1;
      if (iteration < 2) {
        Serial.print("Output state changed to HIGH, Temp over ");
        Serial.println(test_temp);
      }
      digitalWrite(OUTPUT_PIN, HIGH);

        analogWrite(ENABLE_PIN, 250); // Enable the motor
        digitalWrite(DIRA_PIN, HIGH); // Set motor direction or any other motor
control logic
        digitalWrite(DIRB_PIN, LOW);
```

```
      if (irrecv.decode(&results)) {  // If an IR signal is received
        digitalWrite(OUTPUT_PIN, LOW); // Apply the new output state to the
output pin
        hot = LOW;
        analogWrite(ENABLE_PIN, 0);
        Serial.print("IR Signal Received. Output state changed to LOW");
        irrecv.resume(); // Receive the next value
      }

      displayTempTask();  // Temperature sensor task

      int tempReading = analogRead(DHTPIN);
      double tempK = log(10000.0 * ((1024.0 / tempReading - 1)));
      tempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * tempK * tempK
)) * tempK );       //  Temp Kelvin
      float tempC = tempK - 273.15;            // Convert Kelvin to Celsius
      float tempF = (tempC * 9.0) / 5.0 + 32.0; // Convert Celsius to Fahrenheit

      if (tempF < (test_temp - 0.5)) {
        Serial.print("Output state changed to LOW, Temp under ");
        Serial.println(test_temp);
        hot = LOW;
        analogWrite(ENABLE_PIN, 0);
        digitalWrite(OUTPUT_PIN, LOW);
      }
    }

  }
  bool irState = digitalRead(OUTPUT_PIN); // Read the state of the IR receiver

  if (irState == HIGH) {  // Check if the IR receiver state is high
    analogWrite(ENABLE_PIN, 250); // Enable the motor
    digitalWrite(DIRA_PIN, HIGH); // Set motor direction or any other motor
control logic
    digitalWrite(DIRB_PIN, LOW);
  }
    if (irState == LOW) {  // Check if the IR receiver state is low
    analogWrite(ENABLE_PIN, 0); // Disable the motor
    digitalWrite(DIRA_PIN, HIGH); // Set motor direction or any other motor
control logic
    digitalWrite(DIRB_PIN, LOW);
    }
}
```

Software: Citation

I got the original code for IR sensor/remote, the thermometer and display, and running the DC motor from the “Super Starter Kit for Arduino Uno (CH340)” Dropbox. I used ChatGPT to help me combine them together into one sketch and work with each other. I then made my own adjustments such as the temperature limit to activate/deactivate the motor if its above/below that limit.

Discussion:

I wanted to add an SR latch so you could activate/deactivate the motor by pressing a button on the breadboard too but I couldn’t get it working. I would do that if I had more time and skills and I hope I didn’t blow out my OR gate chip or something. I definitely learned to at least read the C++ programming language during this project. Going into it I didn’t know anything but I can understand what’s going on in the code and can write a little bit myself, based off my small knowledge of python. But now that I’m finished with this project, I was invited to teach the lesson in seminary in my hometown and I had an idea while working on this project to use the IR remote and sensor module with a slideshow to advance the slides from a distance. I researched it a bit and I think it’s definitely possible so I’m going to work on that now after learning from this project.