IoT Cooling System & Anomaly Detection Driven Behavior

Deyshonique Stevenson

**Project Part 1: Initial Design and Planning**

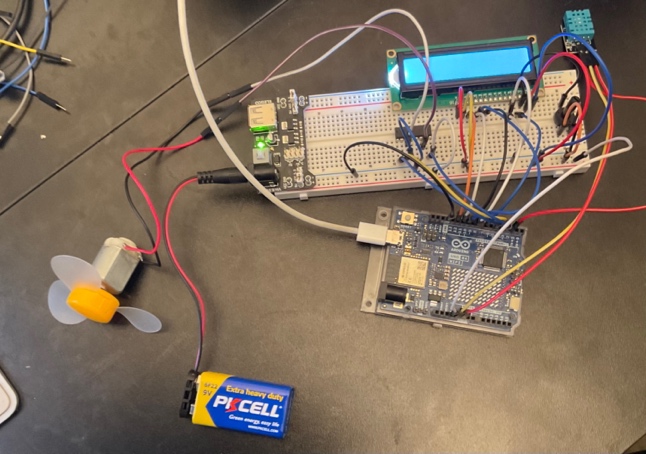
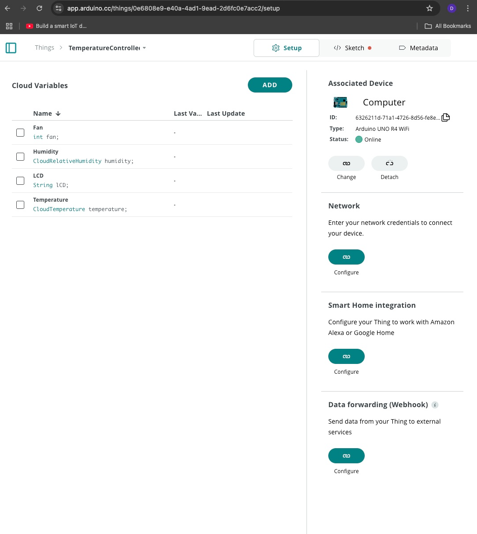
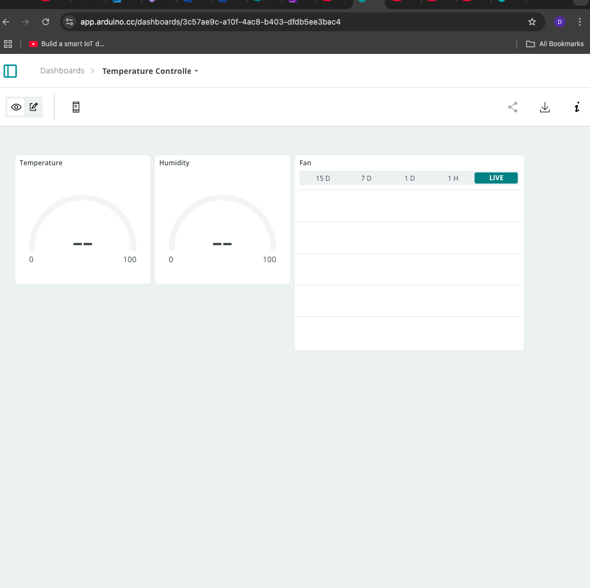
1. The application of my IoT system will revolve around the cooling issue for the heavy computing issue relating to ML/AI models. Once my temperature and humidity senses too much het being produced, it will increase its speed or stall to not disrupt and ensure business continuity is enabled. I will be utilizing an LCD, 5V Relay, and fan blade and 3-6V motor as my actuators. I will be utilizing a temperature and humidity module as my sensor.
2. System Architecture and Hardware Requirements
   1. Key Components/Hardware Requirements:
      1. Wi-Fi enabled microcontroller – to integrate all other components to perform tasks of my IoT system
      2. Sensor – to detect amount of heat
      3. Actuators – to control the fan according to the amount of heat detected and reflect what decisions need to be made
      4. Cloud Service – to collect, and monitor sensor data
      5. Power Module – To use for power conversion
      6. Adapter – to power IoT system
   2. System Architecture:

A diagram of a cloud project

Description automatically generated

1. I plan to use the following hardware for my IoT system:
   1. 5V Relay
   2. Power Supply Module
   3. LCD
   4. Fan blade AND 3-6V Module
   5. Temperature and Humidity Module
   6. Breadboard
   7. Male to male Wires
   8. 9V Adapter
   9. **L293D Motor Driver IC**

**Project Part 2: IoT Project Development**

1. The only change I've made thus far in my project is removing the 5V Relay as I will not need it. I have completed all the wiring for my IoT Project:
   1. 
2. I’ve connected my Arduino Uno R4 to Arduino Cloud. I have been able to create my Thing along with its variables. I’ve also created a Dashboard that will monitor the temperature levels, the humidity levels, and the rate of the fan over time.
   1. 
   2. 
3. **Project Summary:**
   1. During my project thus far, I have been able to ensure the fan and associating dc motor is working. I’ve also been able to ensure the temperature and humidity sensor is working. I will now need to enforce my code and modify it as needed. A challenge I’ve faced is getting the code to upload to the Arduino UNO R4. I was able to upload it to the Arduino UNO R3 with no problems; however, the Arduino UNO R3 is not compatible with Arduino Cloud. I am still troubleshooting this issue.
   2. Links I’ve referenced: <https://www.youtube.com/watch?v=0Fcyj5g26MM>
4. **Code:**

//includes library for LCD

#include <LiquidCrystal.h> //ensure LiquidCrystal library is included

int tempPin = 0;

LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

//include library for DHT sensor

#include <DHT\_Async.h>

#define DHT\_SENSOR\_TYPE DHT\_TYPE\_11

//assign DHT sensor to pins

static const int DHT\_SENSOR\_PIN = 2;

DHT\_Async dht\_sensor( DHT\_SENSOR\_PIN, DHT\_SENSOR\_TYPE );

//assign fan as boolean

boolean fan\_on = false;

//define pins for fan

#define ENABLE 5

#define DIRA 3

#define DIRB 4

// will run only once

void setup() {

lcd.begin(16, 2);

pinMode(ENABLE,OUTPUT);

pinMode(DIRA,OUTPUT);

pinMode(DIRB,OUTPUT);

digitalWrite(DIRA,HIGH); //one way

digitalWrite(DIRB,LOW);

digitalWrite(ENABLE,LOW); // enable off

}

/\*

\* Poll for a measurement, keeping the state machine alive. Returns

\* true if a measurement is available.

\*/

static bool measure\_environment( float \*temperature, float \*humidity )

{

static unsigned long measurement\_timestamp = millis( );

/\* Measure once every four seconds. \*/

if( millis( ) - measurement\_timestamp > 3000ul )

{

if( dht\_sensor.measure( temperature, humidity ) == true )

{

measurement\_timestamp = millis( );

return( true );

}

}

return( false );

}

// code in void loop will control the rate of the fan as it adjusts to humidity and temperature

void loop() {

float temperature;

float humidity;

if( measure\_environment( &temperature, &humidity ) == true )

{

if (temperature > 29.0) {

digitalWrite(ENABLE,HIGH);

if(! fan\_on) {

lcd.println("High temperature - turn on fan");

fan\_on = true;

}

} else {

digitalWrite(ENABLE,LOW);

if(fan\_on) {

lcd.println("Low temperature - turn off fan");

fan\_on = false;

}

}

lcd.print( "T = " );

lcd.print( temperature, 1 );

lcd.print( " deg. C, H = " );

lcd.print( humidity, 1 );

lcd.println( "%" );

}

}