Cooling System For Hardware

BY

Squadra Team

Final Project Report

For Control System Subject

Mansoura University

Faculty of Engineering

Department of Electronics and Communication Engineering

Level 200

Table of content

Overview

The cooling system is used to make the hardware remain safe from temperature damage by starting the fan. When the hardware is beginning to raise in temperature. When the temperature of the hardware exceeds the maximum temperature the hardware can handle, the micro-controller unit (MCU) in Arduino will shut down the whole system and initiate the alarm system.

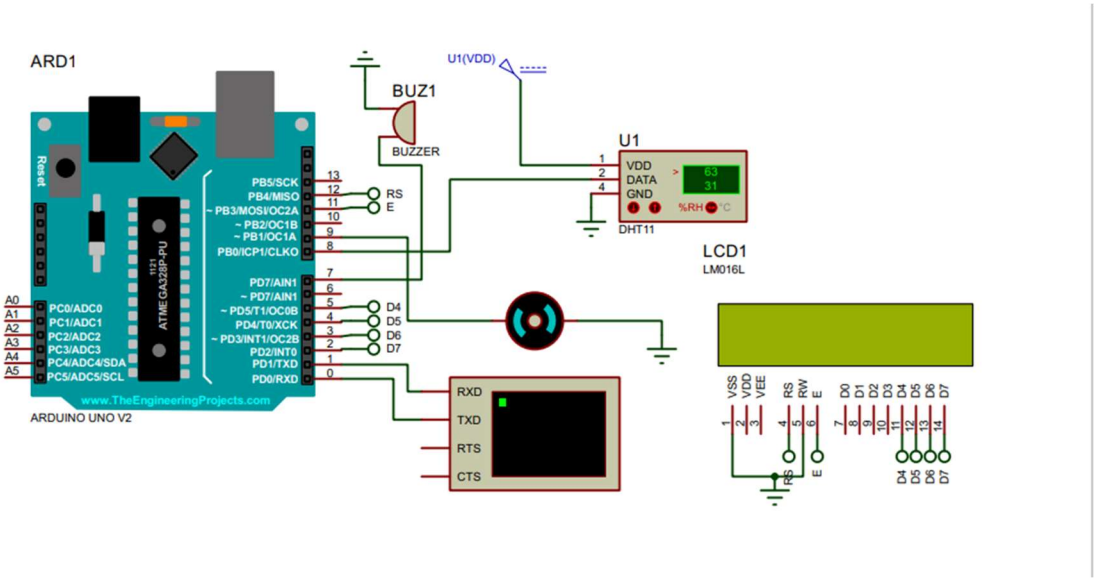
Section 1: Introduction

The cooling system is a system that keeps the hardware temperature in the safe range by measuring the temperature by a DHT11 sensor connected to an LCD to display the temperature reading and humidity. An alarm system is also connected to the cooling system to alert if the temperature exceeds the maximum hardware temperature.

1.1 Components

- LED
- DHT11 temperature sensor
- Fan
- A 330-ohm resistor
- Buzzer
- Regulator
- LCD
- A 12-volt battery
- Potentiometer
- Transistor (P2N2222A)
- Arduino UNO
- Wires

1.2 Schematic



- LCD connections:
  - VSS and RW pins are connected to the ground as VSS is the ground of the LCD and we want RW to write on LCD so it is connected to the ground
  - E and RS pins are connected with pins 11 and 12 respectively. The E pin is responsible for enabling the LCD and the RS pin is responsible for sending data by making it ‘1’ or instructions by making it ‘0’.
  - D4-D7 pins are connected with the Arduino in pins 5-2 respectively and they are used to send data to the LCD.
- Sensor connections:
  - Pin 1 is connected to the voltage source VDD.
  - Pin 2 is connected to pin 8 in the Arduino for communication between them.
  - Pin 4 is connected to the ground.
- The buzzer is connected to pin 7 in the Arduino.
- The fan is connected to pin 9 with the Arduino.

Section 2: System functionalities

2.1 Communication between the DHT sensor and Arduino

The DHT sensor is connected to the Arduino in only one digital pin as the sensor sends 40-bit-synchronous data to the Arduino serially and these 40-bit data are divided into “8-bit integral relative humidity (RH) data”, “8-bit decimal RH data”, “8-bit integral temperature (T) data”, “8-bit decimal T data” and “8-bit checksum”. One communication process is about 4ms. The communication is about the MCU sending a start signal to the sensor in order for the sensor to respond back with data. So we will divide the communication into two parts:

i. MCU sends a start signal to the sensor
ii. The sensor responds to the MCU.

i. MCU sends a start signal to the sensor

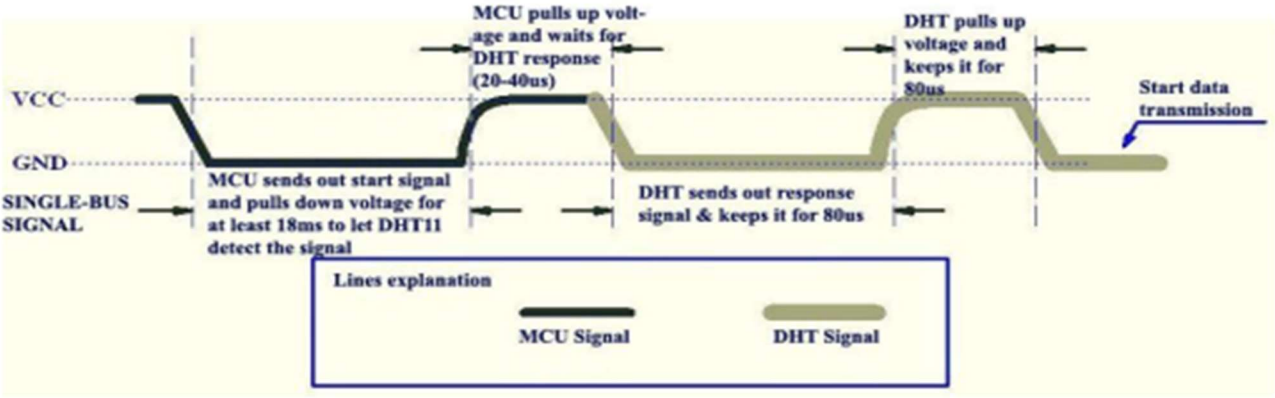


Figure 1: MCU sends a start signal and the sensor responds

Data Single-bus free status is at the high-voltage level. When the communication between MCU and DHT11 begins, the program of

MCU will set data single-bus voltage level from high to low and this process must take at least 18ms to ensure DHT's detection of MCU's signal, then MCU will pull up voltage and wait 20-40us for DHT's response.

ii. The sensor responds to the MCU

Once the sensor detects the start signal, it will send out a low-voltage-level-response signal which lasts 80us. Then the program of the sensor sets data single-bus voltage level from low to high and keeps it for 80us for the sensor's preparation for sending data.

When the sensor is sending data to MCU, every bit of data begins with the 50us low voltage level and the length of the following high voltage level signal determines whether data is '0' or '1' as shown in figure 2 and figure 3.

When the last bit of data is transmitted, the sensor pulls down the voltage level and keeps it for 50us. Then the single-bus voltage will be pulled up by the resistor to set it back to free status.

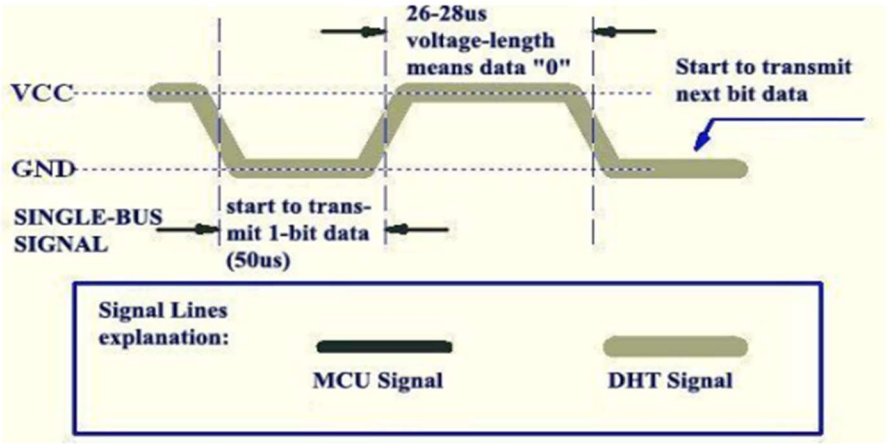


Figure 2: Data ‘0’ indication

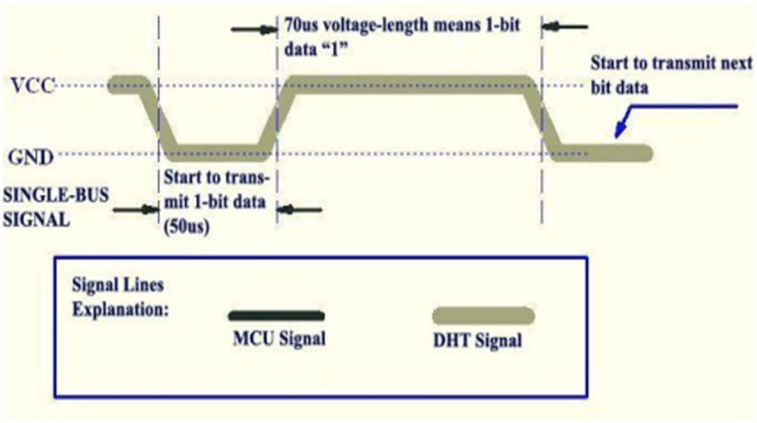


Figure 3: Data ‘1’ indication

2.2 Alarm system

The system is responsible for alerting if the hardware temperature exceeds the maximum limit that can handle by taking a signal from the MCU in the Arduino to do so and the schematic of it is in figure 4.

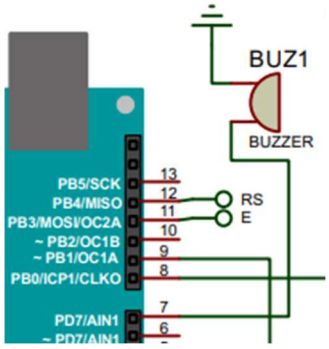


Figure 4: Alarm system

References

- DHT11 Hunidity & Tamperature Sensor Datasheet.
- LMB161A LCD Datasheet.
- P2N2222A – Amplifier Transistor NPN Silicon Datasheet.