DHT11 & ESP32-CAM Smart-Device

Konrad Herbus
Bachelor of Computer Software Engineering
Ontario Tech University
Oshawa ON, Canada
herbus.konrad@gmail.com

***Abstract* — The final implementation of the DHT11 & ESP32-CAM smart device resulted in a greenhouse monitoring system that also incorporated an arduino and LCD display. All the devices were connected into an I2C bus where the arduino UNO served as the master. This master device would collect the data provided by the DHT11 sensor and send this data to both the LCD display and ESP32-CAM where it would respectively be either visually displayed for the client or assessed to determine if an alert needs to be sent. The ESP32-CAM would be consistently checking the values sent from the arduino to verify that the temperature does not fall below 10C or go above 35C and that the humidity does not exceed 90% or rate less than 50%. If these cases ever do arise the ESP32-CAM will automatically send the client an email detailing the alerted value, along with a photo attachment of the greenhouse. During the process of creating this smart device, a level shifter was applied to protect the ESP32-CAM from being damaged by the signals sent from the arduino. Furthermore, in order to send the data using serial transmission the data being sent by the arduino to the ESP32-CAM was first transformed into a string where it would then be parsed on the other end once it was fully recieved.**

I. Introduction

The goal of this project was to create a smart device that would incorporate the DHT11 sensor and ESP32-CAM. The final product created was a greenhouse sensor that will alert the user if the temperature or humidity exceeded the ideal conditions for a greenhouse. This alert will be sent via email and will update the client on the current problematic temperature or humidity value. Furthermore, this email will attach a photo of the greenhouse taken by the ESP32-CAM so that the recipient may assess any damages that have occurred. Finally, the temperature and humidity values taken by the DHT11 will also be sent to an LCD display that the client will be able to view while they are within their greenhouse.

II. Application

A. Connections

When connecting all the devices together an I2C bus was utilized with a master arduino and the rest being slaves, this was used to ensure effective data transfer, whilst also minimizing the risk of confusion between the arduino and ESP32-CAM. All devices share a common 5V VCC and ground connection supplied by the arduino UNO, their communication signals are all then promptly fed back into the master arduino. There are a few other caveats and supplementary devices used within the connection topology, but these are all addressed in the components section of this report.

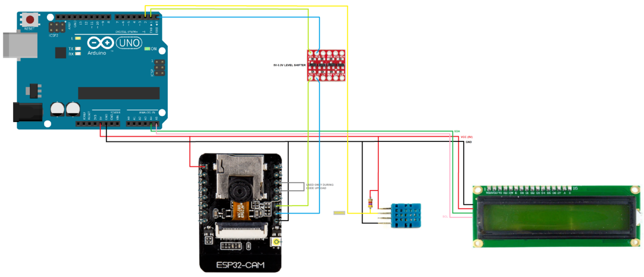

Fig. 1. Full connection diagram displaying all connected devices

B. Packages & Libraries

This project had many separate components that all needed to be managed and communicated with, and it's for that reason that multiple packages and libraries were involved in making this smart device run successfully. The two initial libraries used were the DHT-sensor-library and the Arduino-LiquidCrystal-I2C-library. As the name implies the DHT-sensor-library is crucial in order to effectively communicate with the DHT11 sensor, and was utilized within the master arduino sketch. Similarly, the Arduino-LiquidCrystal-I2C-library was also used in the same sketch code, but this library is used to interface with the LCD display outputting the values obtained from the DHT11 sensor.

The following two packages both pertain to the ESP32-CAM device, and can both be installed via the Arduino IDE. The first package ESP32 by Espressif Systems allows one to find and communicate with the ESP32-CAM device through the Arduino IDE, this is used to upload the sketches to the devices. Meanwhile, the second package ESP32 Mail Client is what provides the software uploaded to the ESP32-CAM with the ability to send emails.

C. Critical Greenhouse Values

In order to determine the appropriate upper and lower bound limits for the temperature and humidity within the greenhouse a scientific report was used. Titled "*Review of optimum temperature, humidity, and vapour pressure deficit for microclimate evaluation and control in greenhouse cultivation of tomato: a review*", this publication stated an upper failure ranging from 35-40C and 100%, along with a lower failure of 9-10C and 60-30% both for temperature and humidity respectively [1]. These values were then applied to the ESP32-CAM sketch where the software will verify that the temperature and humidity are within these ranges. In the event that the values exceed or fall below these thresholds, an email alert will be sent to the client.

III. Components

A. DHT11

Following the successful use of the DHT-sensor-library, the DHT11 sensor can be connected to both the arduino and I2C bus. The four pin outs found on the bottom of the DHT11 correspond to the VCC pin, the signal pin, an unused pin and the ground pin when working from left to right. The signal pin also requires a 10K Ohm pull up resistor between the signal connection and VCC connection. The purpose of this resistor is stated in section E. Resistor & Level Shifter that can be found below.

Fig. 2. DHT11 sensor

B. ESP32-CAM

The ability to take photos and subsequently email these photos to any desired user is entirely managed by the ESP32-CAM, with supplementary data obtained by the DHT11 sensor being provided via the arduino UNO RX and TX pins. This connection between the arduino and the ESP32-CAM also includes a level shifter which is explained in section E. Resistor & Level Shifter as seen below. Furthermore, the connection between the GPIO 0 and GND pins both found on the ESP32-CAM is temporary. This temporary connection is used to put the device in flash mode, and allows for the sketch code to be uploaded using the Arduino IDE, following a successful upload the wire must be disconnected.

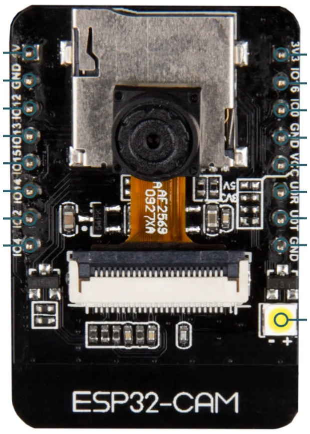

Fig. 3. ESP32-CAM

C. LCD Display

When implementing the LCD display the connected wires were a very simple I2C connection with the VCC and GND pins connecting to the shared 5V and ground. Meanswhile the SDA and SCL wires led to the A4 and A5 arduino pins to allow for communication between the LCD and arduino. The information shown on the display was the data collected by the DHT11 sensor and was sent to the LCD using the Arduino-LiquidCrystal-I2C-library package.

Fig. 4. LCD Display

D. Arduino UNO

The arduino is the main component behind this entire smart device and servers as the master in the I2C bus. This device collects the data from the DHT11 sensor before sending those values to both the LCD and ESP32-CAM. The arduino is also vital in uploading the sketch code to the ESP32-CAM, due to the fact that the ESP32-CAM does not have a USB port where the sketch would be sent from a PC.

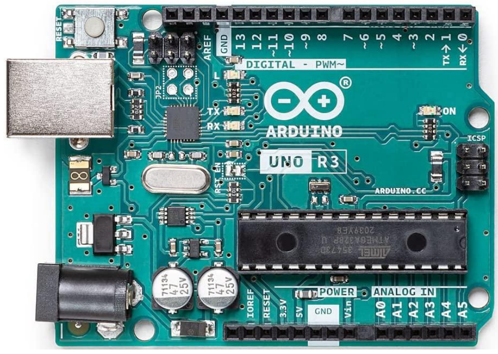

Fig. 5. Arduino UNO

E. Resistor & Level Shifter

The DHT11 sensor requires a 10K Ohm pull up resistor to ensure proper communication between the sensor and arduino. This resistor is connected between the VCC and Out pins of the DHT11 device.

When connecting the RX and TX pins of both the arduino and ESP32-CAM it's important to remember the difference in voltages between the two devices. The ESP32-CAM requires a voltage of 3.3V, while the arduino supplies 5V. This difference is the reason that the voltage level shifter was incorporated with the RX to U0T and TX to U0R connection. Simply using a few pull-up resistors the level shifter ensures that the ESP32-CAM wont be damaged by the signals supplied from the arduino UNO.

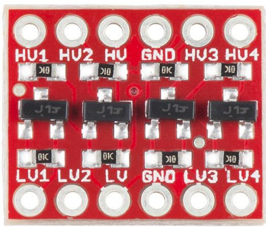

Fig. 6. Level Shifter

IV. Challenges

This project came with many challenges throughout its development, from the various device connections to the number of packages and libraries required. One major challenge was the process of sending serial data from the Arduino UNO to the ESP32-CAM. The data being sent consisted of the temperature and humidity variables obtained from the DHT11 sensor. When sending serial data the data can only be sent as either a string or a byte, meaning that the two pieces of data would need to be organized into a single string before being sent. Once the string data had been fully received by the ESP32-CAM, it would need to be parsed using the strtok() function and converted back into an integer using the atoi() function.

V. Conclusion

After connecting all the devices along with their respective safeguards and going through the process of writing out and uploading the two sketch files using the arduino IDE, this greenhouse monitoring smart device is fully functional. This project came with a number of different challenges that were all addressed using a variety of both hardware and software solutions. Ideally, all the hardware will be housed within a watertight container with the exception of DHT11 sensor, where the pin connections would be insulated with rubber and fed through to the inside of the container to ensure the humidity in the greenhouse environment does not damage any of the electronics

VI. References

[1] R. R. Shamshiri, J. W. Jones, K. R. Thorp, D. Ahmad, H. C. Man, and S. Taheri, "Review of optimum temperature, humidity, and vapour pressure deficit for microclimate evaluation and control in greenhouse cultivation of tomato: a review," Sciendo, 16-Feb-2018. [Online]. Available: http://archive.sciendo.com/INTAG/intag.2018.32.issue-2/intag-2017-0005/intag-2017-0005.pdf. [Accessed: 26-May-2022].

[2] Veeru, "ESP32 Cam Based Motion Triggered Image Capturing Device," *Electronics Innovation*, 21-Jun-2020. [Online]. Available: https://electronicsinnovation.com/esp32-cam-based-motion-triggered-image-capturing-device/. [Accessed: 26-May-2022].