ELEKTRONICA-ICT

Elektronische systemen 2 2023-2024

GREENHOUSE CONTROLLER

Author *Miguel Nunez*
Xander Aerts

Abstract

This paper introduces a novel approach for the development of a prototype controller tasked with sensor data acquisition tailored for greenhouse environments. The study focuses on the design, implementation, and evaluation of an integrated system facilitating real-time data collection and analysis to optimize growth conditions within greenhouses.

Content

1 Introduction

Greenhouse agriculture plays a crucial role in ensuring food security and sustainable food production amid an increasingly variable climate and environment. Precise monitoring and control of growth conditions within greenhouses are imperative for maximizing yield and minimizing resource consumption. In this context, this research focuses on developing a prototype controller capable of reading sensors to measure critical parameters influencing crop growth.

The proposed prototype is driven by the combination of an ATMega 328PU chip from an Arduino Uno and an ARM Cortex M4 processor from an Arduino 33 BLE. This combination of microcontrollers provides the computational power and connectivity necessary for real-time data collection and processing in a greenhouse environment.

The sensor values measured include air and soil humidity, temperature, electrical soil conductivity, and air pressure. These parameters are vital for understanding growth conditions and identifying potential stressors affecting plant growth.

The prototype design incorporates a modular architecture that offers flexibility for adjustments to various greenhouse configurations and crop types. By integrating electronics and sensor technologies, this research aims to deliver an accurate and reliable control system empowering growers to gain real-time insights and optimize their cultivation practices.

2 Hardware

2.1 PCB

2.1.1 Block diagram and schematic

The controller can be powered via micro USB or a 5V DC power jack. Programming of the chips can be done through the same micro USB port or via the ICSP1 and ICSP header.

The ATMega chip circuits are directly powered with 5V, while the ARM chip circuit is powered with 3.3V through the LDO and level shifter.

The ATMega16U serves as both a USB to serial converter and can be configured, via JP3, into different modes (Debugwire support, ISP programmer, or ISP header).

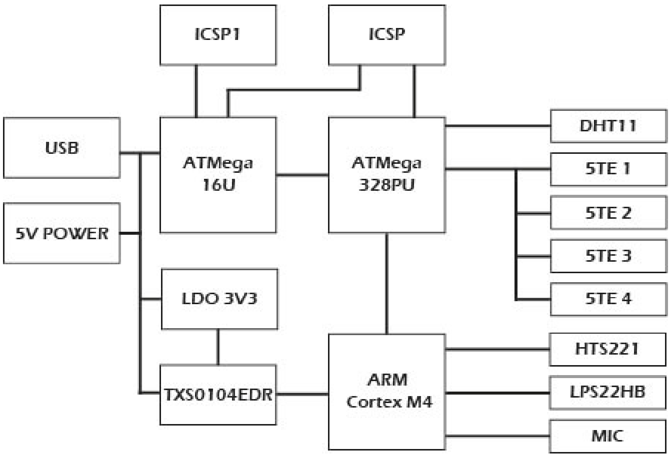

Figure 1: block diagram

The ATMega328PU circuit processes the DHT11 and 5TE sensors, while the ARM circuit handles the built-in sensors from the Arduino 33 Sense BLE.

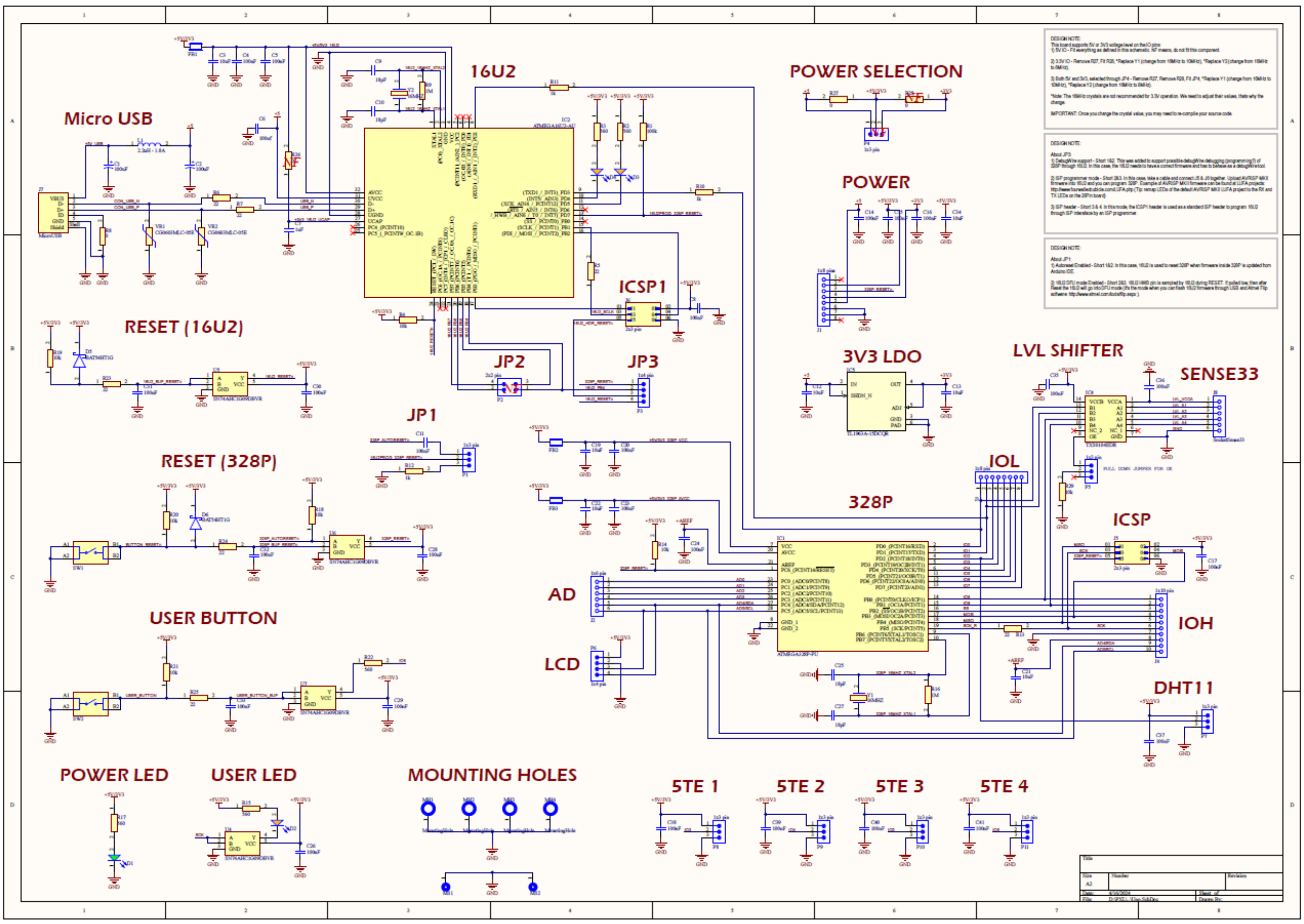

Figure 2: Schematic

2.1.2 PCB layout

The bottom plane comprises three distinct zones: ground, 5V, and 3V, as delineated in Figure 3 by respective blue, red, and green markings. This configuration facilitates easy access to the voltage channels for all components, while also partially mitigating electromagnetic interference (EMI) and electromagnetic susceptibility (EMS).

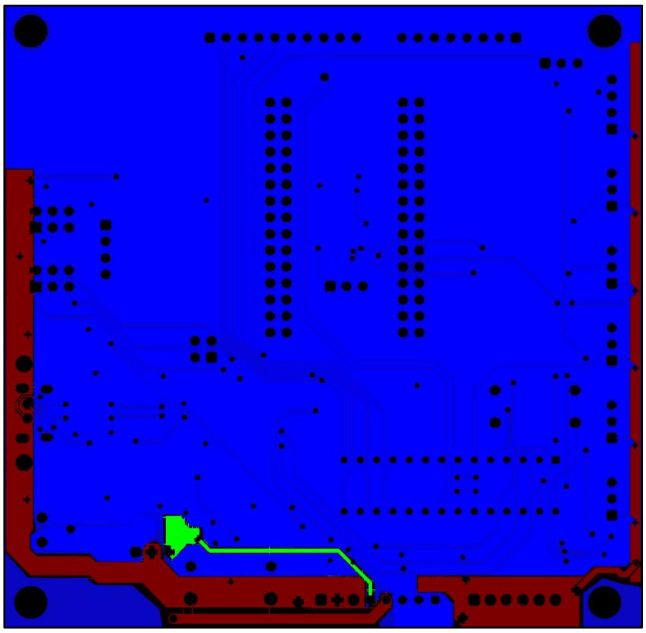

Figure 3: Bottom layer with ground, 5V and 3.3V polygons

The majority of signal traces are situated on the top layer. As depicted in Figure 4, the distribution is demarcated by a red colour for the Uno circuit, a green colour for the Sense BLE and level shifter circuit, and blue colour for the sensor connectors.

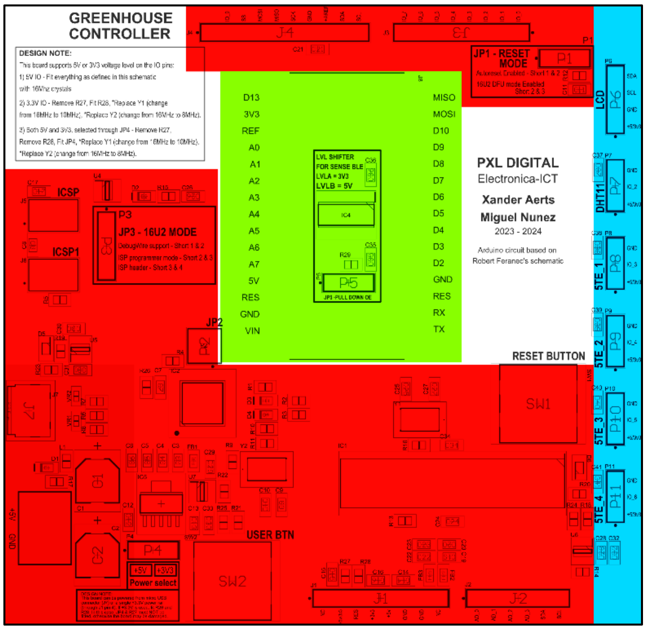

Figure 4: Top layer zones

2.1.3 Board connectors

2.1.3.1 Micro USB

The micro USB port serves the dual function of supplying power to the system and facilitating programming via the Arduino IDE. The USB connector was further mechanically reinforced via the addition of a metal bridge. Nevertheless, practical issues have been encountered with these connectors. Following exposure to the reflow oven, the data pins exhibited instances of short circuiting, impeding the successful uploading of sketches. After solving this short and despite recognition by the computer upon connection of the USB cable, sketch uploading functionality remains suboptimal.

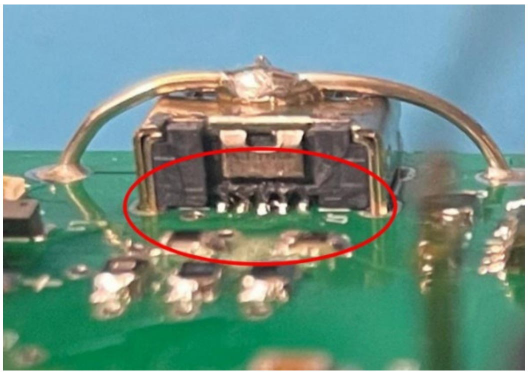

Figure 5: MicroUSB with shorted datapins

2.1.3.2 JST XH

JST type connectors are integrated on the PCB for connecting the sensor connectors and the LCD module, facilitating rapid assembly and disassembly.

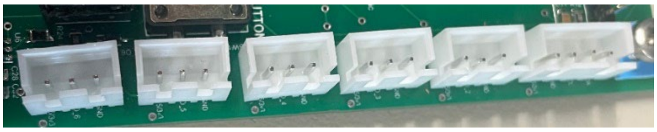

Figure 6: JST connectors for sensors and display

2.1.3.3 Terminal Block

A terminal block is incorporated to supply power to the controller independently of the USB connector. This terminal block is connected to a power jack on the enclosure, facilitating the seamless use of standard adapters.

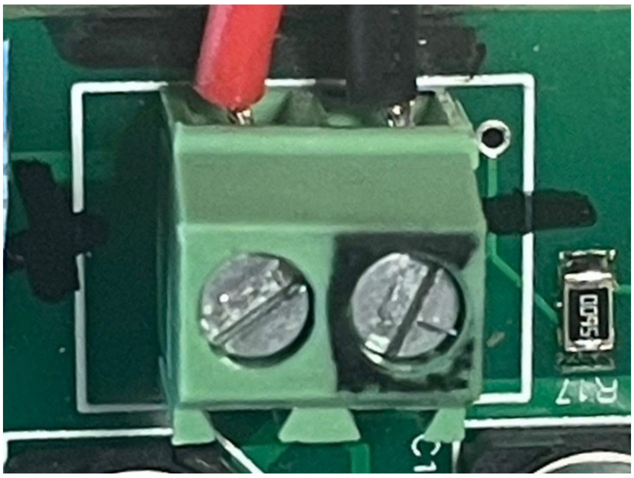

Figure 7: Terminal block for power

2.1.3.4 Headers

Several sockets, including J4, J3, J2, and J1, are provided to make the unused GPIOs accessible for expansions. Additionally, there are pin headers with specific functions: ICSP and ISCP1 for programming the ATMega ICs, JP3 for selecting the programming mode, JP1 for selecting the reset mode, JP2 for modifying the ATMega16u settings, P4 for power selection, and P5 for activating the pull-up of the level shifter. J8 is a custom-made footprint designed to duplicate the pins of the Arduino 33 Sense.

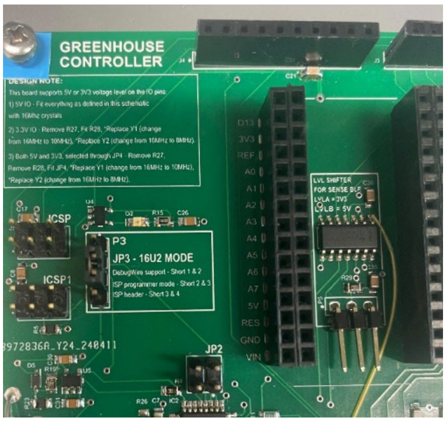

Figure 8: Pin headers and sockets

2.2 Sensoren

2.2.1 5TE

The 5TE 3-Prong Design sensor, developed by Decagon, is a versatile and cost-effective instrument used to measure volumetric water content (VWC), electrical conductivity (EC), and temperature in soils. Utilizing capacitance/frequency domain technology at a 70 MHz frequency, the 5TE sensor minimizes the effects of soil texture and salinity, thereby enhancing measurement accuracy across diverse soil types. The sensor's robust design includes a stainless steel electrode array for EC measurement and an onboard thermistor for temperature detection.

Figure 9: 5TE sensor

One of the key features of the 5TE sensor is its compatibility with the SDI-12 communication protocol, a widely adopted standard for interfacing environmental sensors with data loggers and acquisition systems. SDI-12 allows multiple sensors to share a common three-wire bus (power, ground, and data), facilitating efficient data collection from numerous sensors with unique addresses on a single bus. This protocol supports two-way communication, enabling the data logger to both send commands to and receive data from the sensor.

The SDI-12 implementation in the 5TE sensor supports standard commands for sensor identification, measurement initiation, and data retrieval. For instance, the INFO command (aI!) returns detailed sensor information, including the sensor's address, manufacturer, model, version, and serial number. The MEASUREMENT command (aM!) triggers the sensor to begin measurements, while the DATA command (aD0!) retrieves the measured values of VWC, EC, and temperature.

Decagon has optimized the 5TE sensor's SDI-12 circuit to support up to 62 sensors on a single bus, significantly exceeding the standard limit of 10 sensors. This enhancement is achieved through a low impedance variant of the standard SDI-12 circuit, although it necessitates careful bus management to handle potential faults and ensure reliable operation. The sensor’s digital communication capabilities, low power consumption, and robust encapsulation make it well-suited for long-term field deployments in diverse environmental conditions.

For detailed integration guidelines and to troubleshoot potential issues related to bus configuration, power supply, and grounding, users are encouraged to consult Decagon's comprehensive integrator guide. This document provides essential information for establishing reliable communication between the 5TE sensor and data acquisition systems, ensuring accurate and consistent environmental monitoring.

2.2.2 DHT11

The SEN-KY015TF DHT11 sensor is a cost-effective and reliable device used to measure temperature and relative humidity. It operates with a temperature range of 0°C to 50°C and a humidity range of 20% to 90% RH, providing accuracies of ±2°C and ±5% RH, respectively. The sensor communicates via a single-wire digital protocol, making it easy to integrate into various systems. Common applications include environmental monitoring, home automation, agricultural management, and industrial processes where basic environmental sensing is required.

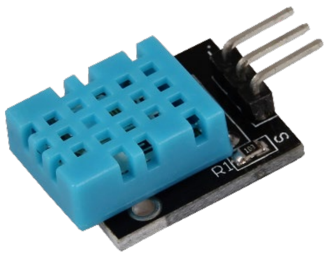

Figure 10: DHT11 sensor

Integrating the DHT11, expands its utility in complex and remote environmental monitoring setups. This integration facilitates reliable data acquisition and enhances the sensor's applicability in diverse and demanding environments.

2.2.3 Sense 33 BLE

The Arduino Nano 33 BLE Sense is an excellent platform for embedded machine learning, integrating numerous sensors alongside its nRF52840 microcontroller running on Arm® Mbed™ OS. This board is equipped with a powerful 2.4 GHz Bluetooth® 5 Low Energy module, allowing for seamless wireless communication using the ArduinoBLE library.

For motion detection, the LSM9DS1 inertial measurement unit combines a 3D accelerometer, gyroscope, and magnetometer, enabling precise orientation, motion, and vibration sensing. Audio applications benefit from the MP34DT05 omnidirectional digital microphone, which captures sound for real-time analysis, supported by the PDM library. Proximity and gesture detection are facilitated by the APDS9960 sensor, which can be programmed to recognize hand gestures and control the built-in RGB LED. The LPS22HB barometric pressure sensor provides accurate pressure measurements and can calculate altitude, while the HTS221 sensor measures both temperature and humidity with high accuracy, making it ideal for environmental monitoring.

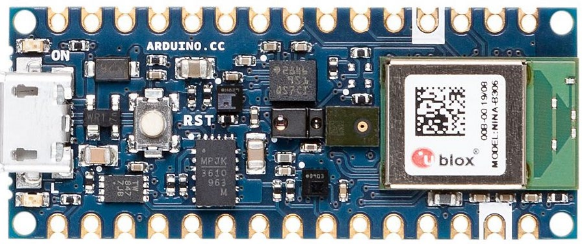

Figure 11: Arduino Nano 33 sense BLE

2.3 LCD display

For visualizing data in this project, we utilized the HD44780 16x2 LCD display, known for its simplicity and effectiveness in displaying alphanumeric characters. This module, measuring 80 mm x 36 mm x 12.5 mm, features a 16-character by 2-line resolution, making it ideal for clear and concise data presentation. It supports both 3.3V and 5V operating voltages and offers a wide operating temperature range from -10°C to +60°C, ensuring reliable performance in various environments.

The HD44780 display includes a white LED backlight and operates in both 8-bit and 4-bit parallel modes, offering flexibility in microcontroller interfacing. Key pins like RS, RW, and E facilitate easy control over data and command registers, while the VO pin allows for contrast adjustment. The module's robust design and straightforward connection to microcontrollers like Arduino make it an excellent choice for embedding real-time data visualization into projects.

2.4 Behuizing

In this project, an enclosure was designed and 3D-printed in PLA (polylactic acid) to protect and house the electronic components. The choice of PLA offers several advantages: it is biodegradable, easy to print, and has a low shrinkage factor, ensuring accurate results. The enclosure is designed with ventilation openings in an organic shape representing air bubbles, creating an aesthetically pleasing and functional design that allows efficient air circulation.

3D printing offers significant advantages in the prototyping phase of a project. It enables designers to quickly iterate and test physical models, saving time and costs compared to traditional manufacturing methods. Additionally, complex geometries and customized designs can be easily realized without extra production costs. This accelerates the development cycle and allows for rapid adjustments based on feedback and testing, ultimately leading to a superior final product.

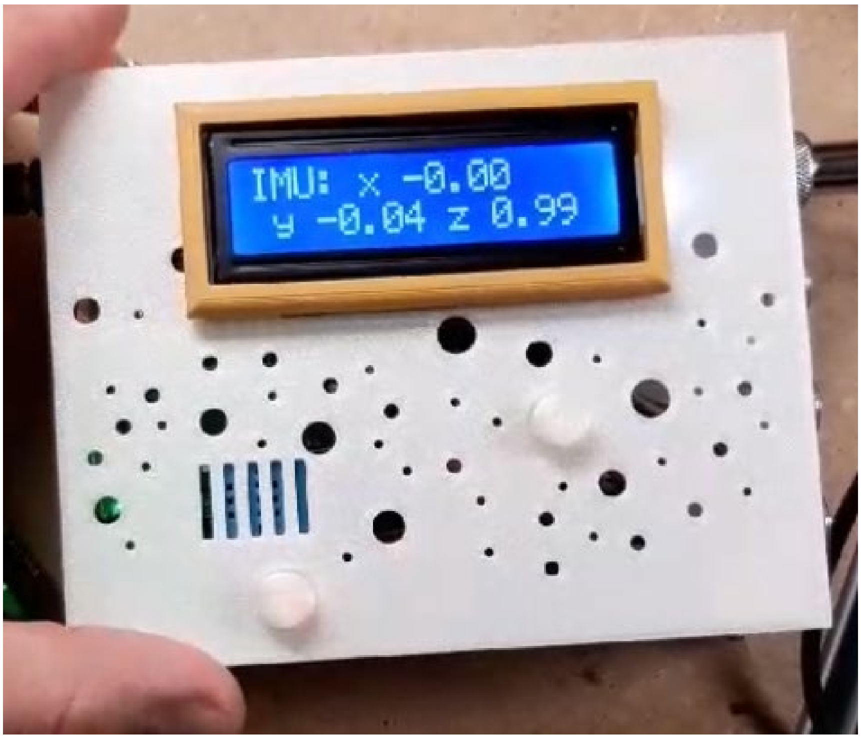

Figure 12: Enclosure front panel

3 Software

3.1 Bootloader

The installation of the bootloader software on the ATMega16U2 chip was performed using AVRDude. However, issues persisted in programming the ATMega328PU chip via USB following this procedure. Despite successful recognition of the USB and chips and the seamless power supply to the system, programming difficulties remained. Even when we tried using the new ATMega328PU chip with a working bootloader from Arduino themselves we were unable to program it with the Arduino IDE. Due to this reason we can confidently say that part of the problem lies with that IC. And given the time constraints, it was decided to use an off the shelf ATMega328PU chip using an off the shelf Arduino UNO board and subsequently place the programmed chip into the socket of our system. Besides AVRDude on its own we also tried to use the build in “burn bootloader” functionality of the Arduino IDE which involved another Arduino board and used AVRDude at its core. This would have burned the bootloader on the ATMega328PU chip but this also was in vein partially due to a limited time.

3.2 Application software

The application software is split into two parts, one being the Arduino Nano 33 BLE Sense that collects the onboard sensor data while the other is the integrated Arduino uno that will collect the external sensor data. In the following blocks we will further explain the code and the different problems that we encountered.

3.2.1 Arduino Nano 33 BLE Sense (Sender)

The Nano Contains multiple onboard sensors where we were given the task to display those values on a display together with the external temperature/humidity sensor and the ground-sensors. The Nano will firstly gather all the sensor data and store them in variables. After that it will combine those values in a message that will be send to the Arduino Uno over UART. An example of such a message would be the following:

```
temp>24.5>hum>52.47>pres>102.5>imu>-0.2, 0.05, -0.42>col>12, 13, 8>*>
```

The ‘>’ sign is used to separate the different sensor values and most importantly, to make parsing that data on the uno’s side much easier. When finished the message will be transmitted to the uno over the Serial1 port, which is connected to the external pins and wait for 3 seconds before reading the data again and redoing the cycle all over again.

3.2.2 Integrated Arduino Uno (Receiver)

The Uno will on its turn connect to the ground-sensor and wait for 3 seconds during startup to allow the sensors to start. We found that the sensors themselves are realy slow in for example sending back data when requested so allowing the sensors to connect properly and start will give us a more reliable application. After the startup the main loop will continuously go through the following 3 main blocks in order:

3.2.2.1 Select button

In the first block, it will see if the userbutton was pressed and if so it will set the screenstate variable plus one. This value will later be used to tell the board what sensor data to show on the lcd screen. The main problem when implementing this functionality was the bounce effect of the button, basically the Arduino whould read more the one button press due to the mechanical functionality of it. This was resolved by implementing a software Schmitt-trigger that would not only look at the current state of the button but also at the previous state to ensure that there realy is a new buttonpress.

3.2.2.2 Receiver

After that it will check if there are UART messages available in the internal buffer if so, it will keep on reading them and adding them to a string until it encounters the '*' symbol meaning that the message is complete. After which it will set the parse flag to true, which gives the signal to the following parse block to start processing the message.

3.2.2.3 Parser

When the parse flag is set to true, the parser will begin with reading out the ground-sensors data and store them in the according variables. Immediately after that it will start processing the message by cutting it into pieces at the previously described '>' symbol. Afterwards it will use this data together with the agreed-upon format to parse the sensor data out of the message and store them also in the proper locations. When all these steps are done it will enter the "switch case" where, according to the screenstate variable (that was set by the user button), the according sensordata will be shown on the screen.

A half a second delay will end the loop and restart the process once again.

3.2.3 Why UART

The reason for UART was because of its simplicity and functionality. When looking at all the options: I²C, SPI and UART we decided to utilize UART because it was meant to transmit messages in the form of characters in contrary of I²C which would need a more sophisticated format to function. With UART we were able to create a string message from which we new that we would be able to parse that on the uno's side very easily while it would require more research to do that with I²C. So the main reason that we didn't choose for SPI and I²C was due to more complexity and a lack of earlier experience with SPI.

3.2.4 Encountered problems

When working with the 2x16 lcd display we tried multiple library's in the hopes that it would work on the nano but after multiple attempts we were unsuccessful to have it working even with a level-shifter in between to allow for 5v logic/power on the lcd's side. The reason for this was the fact that all the librarys weren't written for the arm architecture but for the atmel architecture. Because of this we decided to utilize an Arduino Uno (that was later integrated on the pcb) to drive the display. Later on we also used that microcontroller to read the external ground-sensors and the external temperature/humidity sensor. Due to insufficient time we haven't been able to try and connect the ground-sensors to the Nano board and decided to leave it that way.

4 Result

The final result is a controller capable of reading sensor data from various sensors and displaying it on an LCD screen. Three hardware issues were identified:

The provision of a 3.3V power line between the LDO and the level shifter was missing. This was resolved by connecting the components with a rework cable.

The resistor intended to pull the OE pin of the level shifter to ground needed to be pulled high to the 5V line. We addressed this by connecting a THT resistor to an accessible 5V pin via the header.

Powering the Arduino 33 BLE through the VIN pin did not provide sufficient voltage to make that part of the system operational. We had to create internal connections from the MicroUSB of the 33 BLE to the 5V terminal block of the PCB.

In a subsequent revision of this board, these issues can be easily corrected in the design. The logical next step is to establish communication between this system and an external database or other devices via Bluetooth, enabling data logging. Based on this data, analyses can be performed, and it may also be possible to control infrastructure elements such as activating sprinklers, opening or closing windows, and adjusting curtains.

5 Discussion

The completion of this project marks a significant learning experience, highlighting both the challenges and the successes encountered throughout its development. Despite facing hardware issues during the implementation phase, the process provided invaluable insights into troubleshooting and problem-solving within an electronics project context. Moreover, the successful integration of various sensors and the LCD display demonstrates the feasibility of the proposed controller concept.

This project lays a solid foundation for further research and development in the field of greenhouse control systems. By addressing the identified hardware issues and refining the design, future iterations of the controller can be optimized for enhanced performance and reliability. Additionally, the potential for integrating communication capabilities, such as Bluetooth connectivity, opens up avenues for data logging and remote monitoring, thus increasing the system's versatility and utility.

Furthermore, the broader implications of this project extend beyond its immediate scope, suggesting its potential as a building block for a comprehensive greenhouse control solution. As part of a larger system, the controller developed in this project could serve as a crucial component in automating and optimizing greenhouse operations. By leveraging sensor data to regulate environmental conditions such as temperature, humidity, and lighting, the system could contribute to improved crop yield and resource efficiency.

6 Conclusion

Looking forward, the success of this project opens doors to further research and development, with opportunities for refining the controller's design and enhancing its functionality. By addressing hardware limitations and exploring avenues for communication improvements, we can pave the way for a more robust and reliable greenhouse management solution.

Moreover, the broader implications of this project are significant. It lays the groundwork for future advancements in agricultural technology, with the potential to revolutionize farming practices and contribute to global food security. As we continue to push the boundaries of what is possible, this project serves as a testament to the power of innovation and collaboration in addressing the challenges of tomorrow.