Using a Microcontroller to Automate a Green House for Experiments

M. Al-Kofahi, C. Chapman, B. Edison, E. Johnson, M. Harman, N. Zayyad, J. Seybert, C. Watters, R. Emme & T. Sliva

Department of Physics
Baker University
Baldwin City, KS 66006, USA.

W. Miller
Department of Biology and Chemistry
Baker University
Baldwin City, KS 66006, USA.

Abstract

In this work, an Arduino Mega 2560 R3 is used to automate the green house of Baker University, Kansas. This includes monitoring and controlling soil moisture and PH, light intensities, and temperature. Growth of plants are monitored by video/ image capture at preset periods, and recorded on a Data Logging Shield equipped with its own memory card. A copy of the logged data is transmitted wirelessly by XBee Pro 60mW Antenna to the base computer at the Biology Department. The system focuses on monitoring and controlling of all parameters of interest and on data collection. It allows for watering the plants using electromechanical solenoid valves controlled by the Arduino board; and allows for controlling light intensities and temperature inside the green house. Complete documentation of algorithms, materials used, methods, programming, and integration of the whole project is reported here in this paper.

Keywords: Arduino Mega 2560 R3, collaboration, hydraulics, temperature, humidity, light

Introduction

With the expansion of the undergraduate research programs at Baker University (BU), the annual cycle of dozens of new research projects stresses both faculty time and departmental resources. Thus, a growing need for increased data to support multiple projects was identified. Standard methods of survey, transect, capture & release, measurements, and observation still work for many projects. However, increasingly, students are asking relational research questions that require the collection of data over time.

Such as, does plant growth rate change with light intensity? Alternatively, moisture availability? Alternatively, soil pH? Alternatively, combinations of these variables? Students with these types of dynamic research questions need a way to capture data that in itself does not have a steep learning curve. Seeing the need, the Physics Department proposed to develop a simple and inexpensive data capturing system for Baker green house that would provide support for multiple courses offered by the science departments. The Physics Department class of 2016 accepted the challenged to develop such a system as their Senior Project. First, they surveyed the faculty of the Biology, Chemistry, and Math Departments to identify the nature of data that students might be able to incorporate into their research projects. The list was narrowed to five common threads: soil moisture, soil PH, light intensity, and temperature coupled with a way to measure change (video). Next, the class proposed to use a central microcontroller to automate the capture of the desired data and to allow experimental responses to be applied to the test subjects. They presented the design back to the science faculty, and received a small budget for implantation of a pilot project. Working in teams on the various parts, the class acquired the parts and pieces to build a bench top working model, which was followed by a demonstration presentation to the science faculty. After much discussion and questions, the project was approved and funded for installation as three experimental zones in the green house. Each zone capable of measuring and controlling all five-design factors. This paper presents the details of the design, discusses some of the challenges and solutions, and reports on the first experiments.

Materials and Methods

Interface

The greenhouse house at Baker University contains a large collection of mild climate plants distributed over three distinct environments and requires watering once a week. Each environment acts with its own ecosystem with plants that require unique temperature, moisture, and lighting needs. The aim of this project is to automate the greenhouse that houses such that the Arduino microcontroller serves each ecosystem and allow it to be controlled wirelessly. Integrating Arduino automation into the greenhouse would greatly reduce its regular maintenance, allow for more diverse plant life, and even improve the growth rates of existing plants. In addition to these benefits, implementing an automated greenhouse will start collaboration between the biology and physics departments while, increasing the biology department's capacity to perform plant based research. Choosing the Arduino microcontroller for the automation of the greenhouse offered many benefits over other platforms. Arduino is a simple and inexpensive microcontroller with access for customization, computing power, and simplified coding. Arduino offers a full line of "Garduino" peripherals that made sensors selection easier. Fig. 1 shows the detailed algorithm of the Arduino greenhouse microcontroller system built in the present work.

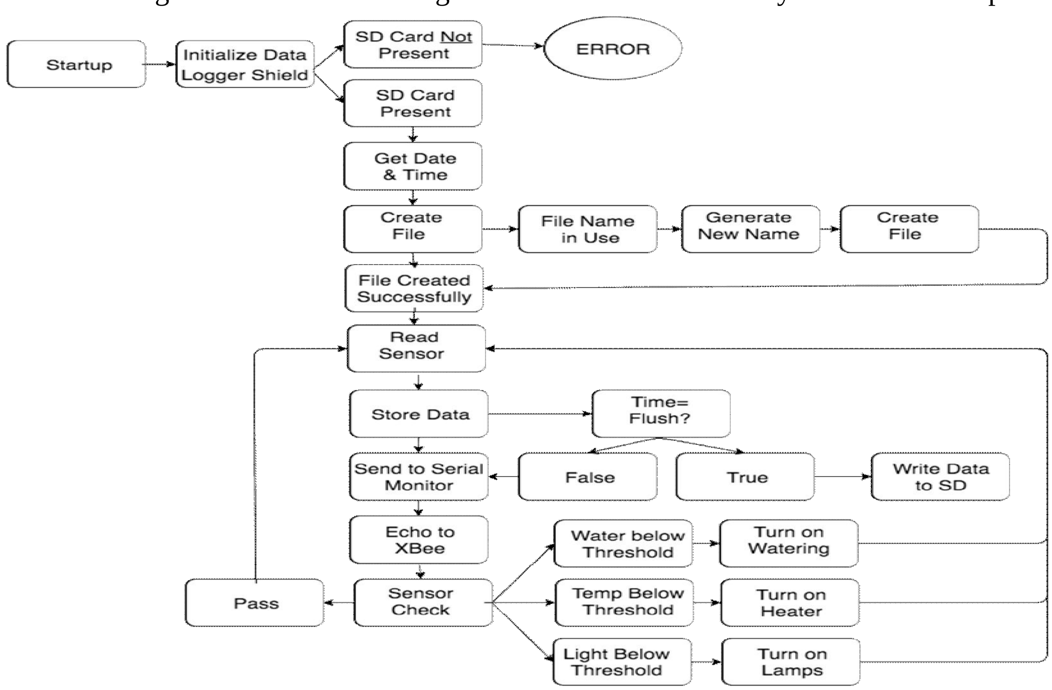

Fig. 1: Algorithm of the Arduino greenhouse microcontroller system

Within the scope of the project to monitor temperature, humidity, light, and water moisture, the need to store the data locally and simultaneously transmit it wirelessly to a classroom, and to a hallway, display became apparent. The backbone of the setup is the Arduino Mega 2560(Figure 2a) which has the processing power to monitor and transmit simultaneously and sufficient ports for future additions. The Mega comes standard with 54 digital I/O pins and 16 analog pins with additional space for prototyping and multiple shield mounts. An Ad fruit Data Logging Shield (Figure 2b) with an eight-gigabyte SD card was chosen to store data locally. Wireless transmission was successfully implemented with an XBee shield and XBee Pro 60mW Wire Antenna - Series 1 (802.15.4) (Figure 2c). The receiving computer was equipped with an XBee USB explorer paired with an XBee Pro 60mW Wire Antenna - Series 1 (802.15.4).

The Arduino Mega and accompanying sensor setup were installed inside a waterproof box. The box was mounted into the greenhouse near the breaker box. The mounted box protects the Arduino boards from the hot and humid environment of the greenhouse (Figure 2d).

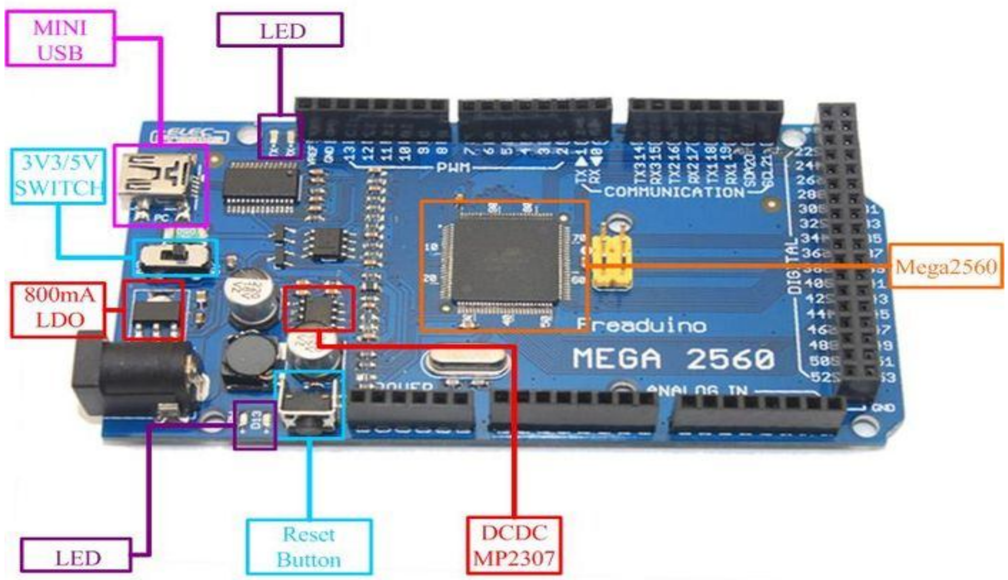

Fig. 2a: The Arduino Mega 2560 microcontroller board comes standard with 54 digital I/O pins and 16 analog pins with additional space for prototyping and multiple shield mounts.

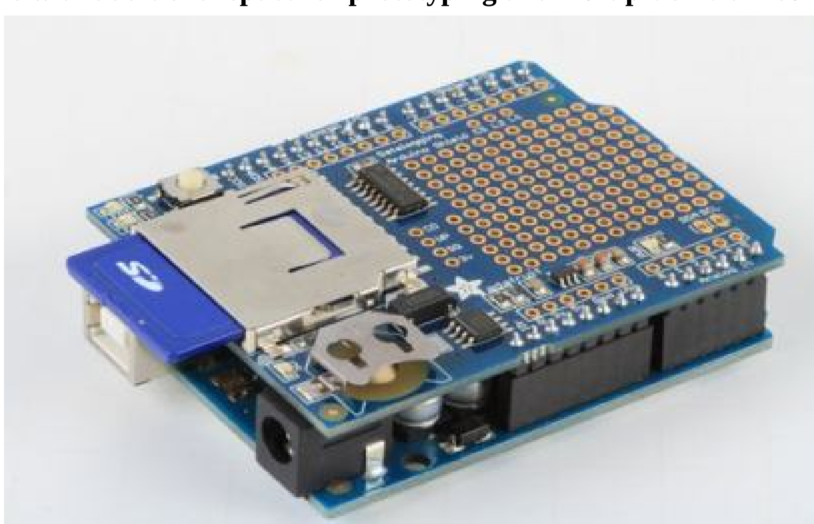

Fig. 2b: An Ad fruit Data Logging Shield with 8 GB SD card is used to store data locally.

Fig. 2c: XBee Pro 60mW Wire Antenna - Series 1 (802.15.4), 250kbps max data rate,60mW output (18dBm), 1 mile (1600m) range, and a built-in antenna. Fully FCC certified.

Light Monitoring:

In order to achieve an optimal growth environment, a way to measure and control the light intensity inside the greenhouse was needed. As shown in Fig. 3a, the wavelength of visible light extends from 400-700 nanometers.

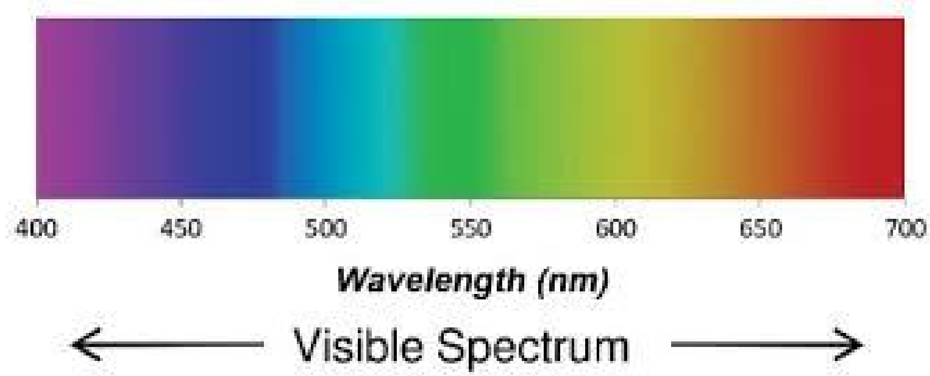

Fig. 3a: Visible light spectrum extends from 400nm to 700 nm.

Photo Light Sensitive Resistors (Photo resistor Opt resistors) 5mm GM5539 (Figure 3b) sold by Goeasy buy) [4]were selected. Initially the spectral peaks of these photo resistors were 540 nanometers; however, the ideal visible light for plant growth is around 650 nanometers. The range of the photo resistor was extended to the full visible light spectrum (400-750 nanometers) with the addition of a 10K resistor in series with the photo resistor (SIK). The photo resistors were wired into the analog pins on the Arduino board.

Fig. 3b: Photo Light Sensitive Resistor (Photo resistor Optoresistors), 5mm GM5539.

Once a working circuit for one photo resistor was tested, two additional circuits were built in parallel on the breadboard to represent each of the three monitoring zones in the greenhouse. Arduino code based on the SIK experiment displayed each of the three readings for light intensities on a serial port screen once every second.

Fig. 3c: Three-60W compact fluorescent lamp sold by Apollo Horticulture) [5]are used to boost light intensity in the greenhouse when needed.

Fig. 3d: A four Channel DC 5V Relay Module for Arduino sold by Geree) 6used to power the lamps.

Light Actuation:

Initially actuation was to combine the lighting automation with the temperature automation. But thermal management and light requirements did not always line up thus temperature was interfaced to the greenhouses thermostat for the gas fired heater and vent fans while the light was directed to zone specific elimination. To automate the lighting requirement, a 60W compact fluorescent lamp (Fig. 3c) sold by Apollo Horticulture) [five], which emits a light spectrum suitable for plant growth between 450-700 nm equivalents to a 300W incandescent bulb, was selected. Photo resistors monitor light intensity. To control the light intensity, a 5 VDC 4 channel solid state relay board sold by Geree) [6] (Fig. 3d) was used to switch on the light bulbs when the light intensity is lower than the pre-set value for 10 minutes. The coding for automation was designed to turn the light on only after the photo resistor receives readings below the desired value consistently for 10 minutes. This is to avoid the bulb turning on and off frequently on a partly cloudy day.

Once the bulb turns on it will stay on until the photo, resistor readings are within the specified range for 5 minutes. After the light turns off, the process will repeat as necessary until night, when Arduino will keep the lights off throughout the night and turn them back on in the morning. Because the solid-state relay board needed 60 mA of current and the Arduino Mega delivers only 40 mA, an external power supply was used to provide power for the relay.

Soil Moisture Monitoring and Irrigation System Development

One of the overriding goals of this project was to control the greenhouse such that it would be a suitable environment for experimentation. A key part of experimentation is control of the water the plants receive. This was achieved in two ways. First with the Arduino, the inputs on the interface allowed for the selection of the time between watering periods and the length of each watering period. Secondly, the use of a drip irrigation system allowed control of the quantity of water each plant received. Based on size of pot and necessity of water, each plant could receive exactly as much water as it needed through the interface settings as well as dripper heads.

The Soil Moisture Meter Testing Module made by Atomic Market) one was chosen for measuring the moisture level because of its low cost and availability (Fig. 4a). The module came with a soil probe, but after testing it, it was clear that the probe was insufficient for the desired long term monitoring due to corrosion. After just two days of exposure to moist soil, the probe showed signs of serious deterioration due to the acid in the soil as well as oxidation. Because the goal of the project was long term uninterrupted monitoring, a homemade probe was designed using two galvanized nails stuck through a square of inch thick foam board an inch apart. The foam square was then covered in electrical tape to hold the nails in place and to create water resistance. These homemade probes were used with the original Atomic Market modules for integration with the Arduino microcontroller system. The sensors proved to be a reliable and cost effective long-term solution.

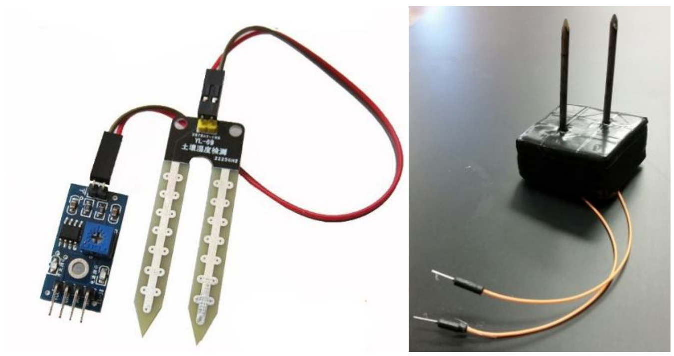

Fig. 4a: Soil Moisture Meter Testing Module, Soil Humidity Sensor by Atomic Market, and the homemade 2-nails probe.

The second goal of this part of the project was to effectively and reliably deliver water to the plants of the green house. The dripper system option was chosen because it allows for controlling the amount of water delivered to each individual plant in the most efficient way, and allows for conserving the most amount of water. A homemade PVC piping system with electromechanical solenoid valves was designed for the transport of water from its source to the desired areas of the greenhouse. This option was selected because it was the most cost effective solution and because of the greenhouses relatively constant conditions. The Arduino system controls the electromechanical solenoid valves and determines when water is to be released, the duration of watering, and shut off the system.

The PVC piping system was designed using a set of valves and water regulation components that allow full control of water delivery to the dripper system of the desired zones in the best way possible. A ¾” 12V Brass Electric Solenoid Valve manufactured by Vktech)[2] was connected directly to the water source (Fig. 4b). Then a set of ½” 12V Plastic Water Solenoid Valve manufactured by Ad fruit Industries) 3was used for watering the three zones in the greenhouse (Fig. 4c).

Fig. 4b: Vktech 3/4 Inch Brass Electric Solenoid Valve Water Air Fuels N/C DC 12V.

Fig. 4c: A ½” 12V Plastic Water Solenoid Valve manufactured by Adafruit Industries.

Between the main valve and the zone valves, a regulation unit consisting of a DIG Screen Filter, Pressure Regulator, and Backflow Device, was placed. This regulation unit ensured that when the water reached the drip irrigation system it would be filtered and within the proper pressure range for the drippers. After installation of the plumbing and hardware into the greenhouse, the system was wired to the Arduino. The water sensors were connected to one of the analog pins on the Arduino board. The sensor gives a reading for the resistance of the soil between 100-1020 (arbitrary units), with the lowest reading indicating full soil saturation and the highest reading indicating no moisture present. The solenoid valves on the other hand were connected to digital pins via a relay circuit consisting of a N-channel power MOSFET Transistor, 1N4001 Diode, and a 1K Ohm resistor. This relay circuit was connected to a 12V 5A power adapter that was separate from the Arduino to ensure protection to the Arduino itself as well as the other systems.

The coding to control the solenoid valves first opened the desired zone valve and then, after a couple second delay, opened the main valve to release water into the system. After an allotted amount of time the system was closed closing first the main valve and then, after a few seconds, the zone valve. The code was written so that only one zone could be watered at a time to ensure the desired amount of water was delivered during each watering period. The final step to the irrigation system would be to integrate it with Arduino microcontroller interface. Originally, the Arduino would initiate the watering of a zone based on the readings of the moisture sensor falling below a threshold. However, the option of watering schedule on a set time interval based on the water needs of the plants in each watering zone was found to be more reliable and was implemented in the project. The Soil Moisture and Irrigation System algorithm for the three zones of the greenhouse is shown in Fig. 4d.

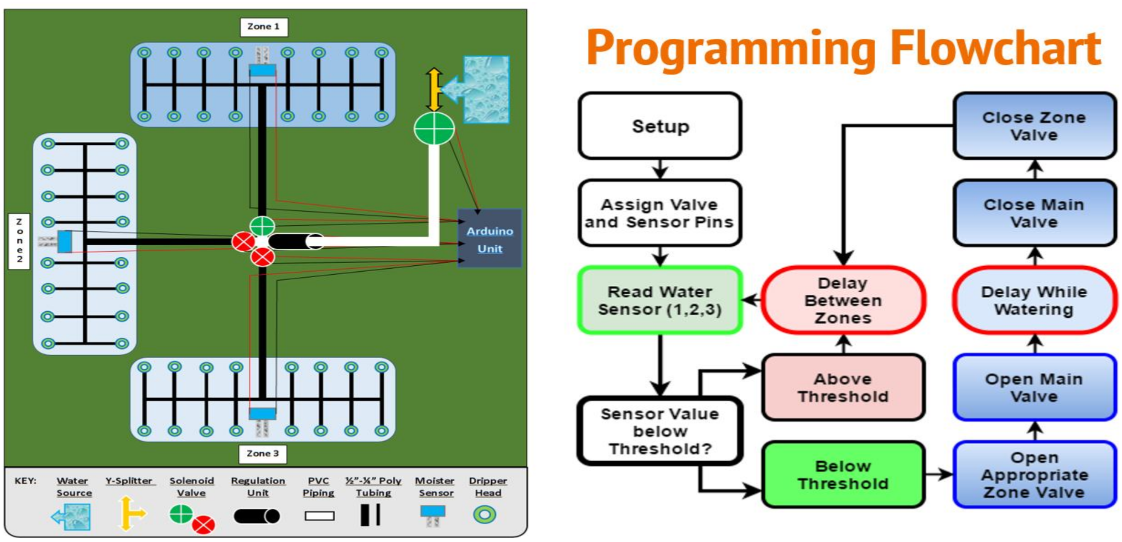

Fig. 4d: The Soil Moisture and Irrigation System algorithm of the greenhouse project.

Temperature

To monitor and record the temperature in the greenhouse, an AM2302 DHT22 Temperature-Humidity sensor sold by Ad fruit) [7] was selected (Fig. 6). This sensor measures both temperature and humidity from the surrounding area every 2 seconds between -40° to +80° Celsius, with a +/- 0.5° Celsius accuracy. The temperature sensor yields a digital signal on a data pin that can be read by the Arduino interface. After initializing the sensor, the code commands the sensor to take readings every 2 seconds. The DHT22 sensor is low cost with a bigger range of detecting temperature/humidity. Detected values are stored in both degrees Fahrenheit and Celsius (Temperature) and percentage (Humidity). The sensor requires no calibration and it can be interfaced with a thermostat controller.

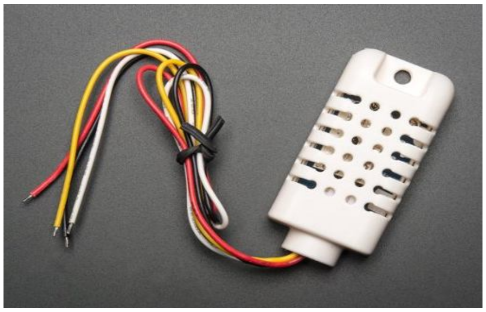

Fig. 6: AM2302 (wired DHT22) temperature-humidity sensor [ADA393] sold by Ad fruit) seven.

Video Capture

In order to monitor the growth of plants in the greenhouse, a camera is needed. Originally, the plan was to capture live video for the plants but when this idea put to implementation it was clear that although it can be done its not practical and not really needed. Video capture requires handling high stream of data that the Arduino microcontroller cannot handle. Therefore, the plan was modified to capturing images for the plants in the green house at a pre-determined rate, like one every 30 minutes. This was suffecient for the needs of the Biology Department.

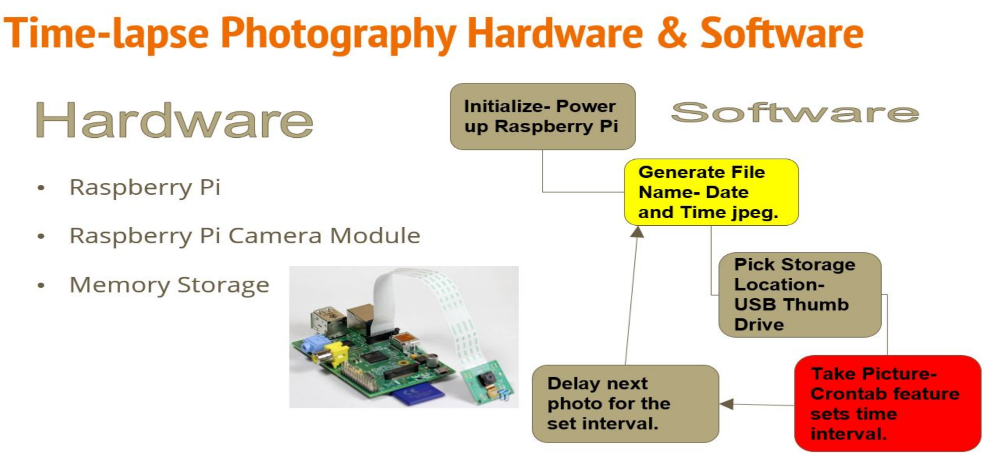

Fig. 7: A block diagram showing the time-lapse photography hardware and software

In order to improve the speed of data transfer, a dedicated Raspberry Pi microcontroller with Camera ModuleOV7670 sold by Next) seven (Fig. 7) and a Memory Storage circuit was used. The circuit hardware allows for image capture and video capture and for storing them on the memory card. For two weeks, the assembly of the camera took place along with its programming.

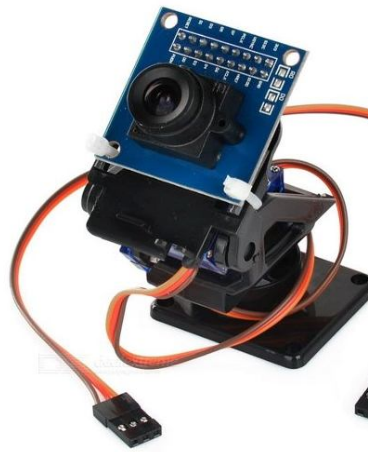

Fig. 7: A 2-Axis FPV Camera Cradle Head and OV7670 Camera Set sold by Next) seven.

The Raspberry Pi was simple to assemble and easier to program. The time between photos can be customized on the interface for convenience. A sealable case was purchased for the camera in order to keep it safe from humidity and any other conditions that could cause damage to the equipment. In order to have the best picture for the camera hole were made so that the lens could have a clear view of the different plants and would be sealed in place to prevent anything from getting in the case.

Results

Interface

Each team built and tested their sensors on their own before submitting a working setup and code. After receiving all of the parts, the stacking headers to the data logger and XBee wireless shields were soldered together. Each individual sensor was integrated into one working code. The system recorded the light, temperature, humidity, and water moisture every sixty seconds. Initially the data was monitored through the serial COM port on the Arduino board. After confirming correct sensor function and calibration, the local data storage was achieved through the Ad fruit Data Logger shield. The system generates a new CSV file each day with the current date as the title. In the case of multiple files being created on the same day, numbers will be sequentially added to the end of the date. After a file was successfully created, opened, and written to locally, the next step was wireless data transmission. The XBee antennas have the ability to use wireless networks to extend their range and connect to any device on the network. Baker University network security protocol would not allow the XBee antennas to access the school's wireless network. Instead, each transmitter and receiver was paired for point-to-point data transmission by assigning the IP addresses of each antenna to each other. The transmitted data was viewable through the receiving computers serial monitor. Fig. 8 illustrates the operation model for the Arduino Greenhouse Microcontroller, which involves data storage on SD card and real-time transmission of the data to the base computer in the Biology Department through an XBee transceiver system.

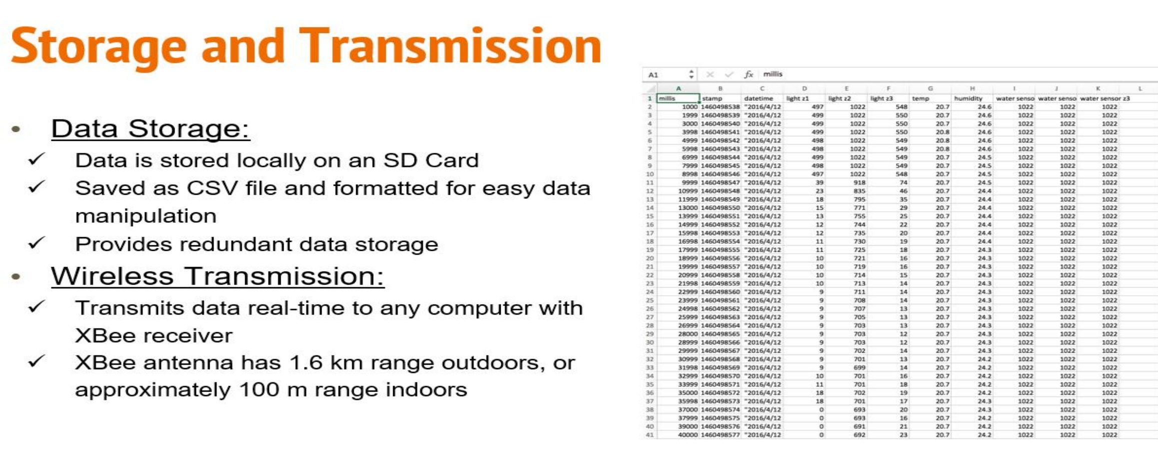

Fig. 8: The operation of the Arduino Greenhouse Microcontroller involved data storage on SD card and real-time transmission to the base through an XBee transceiver system.

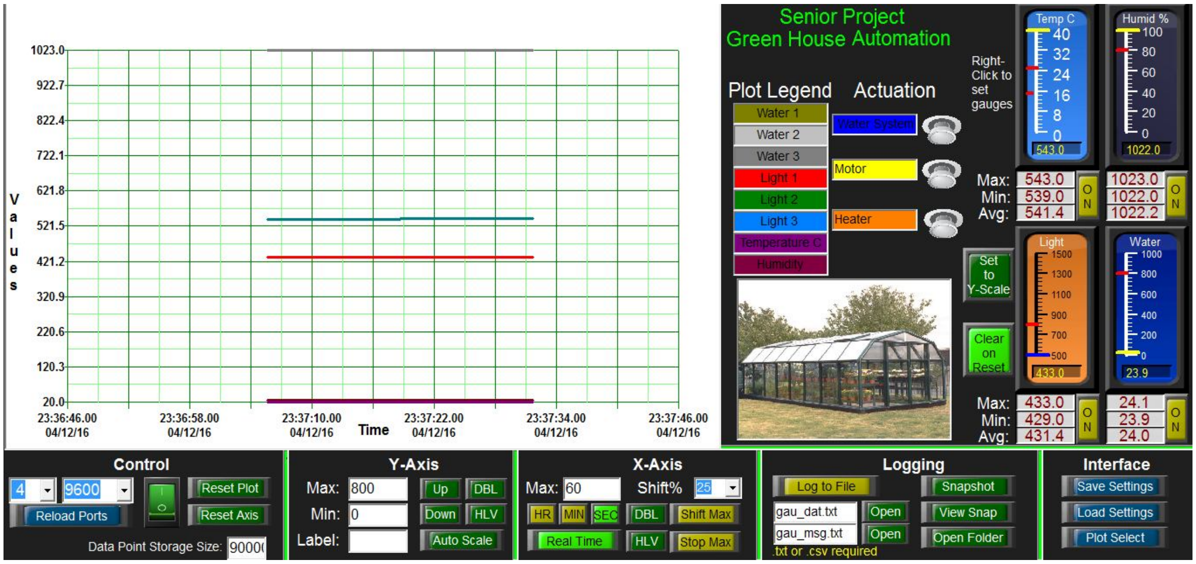

Fig. 9: The Control Panel for BU Greenhouse Project

Several serial monitoring programs were tried for the control panel and Maker Plot) [nine] was selected (Fig.9). Maker Plot offered the customization and reliability that we needed for this project. The Maker Plot interface setup is very easy to customize and alarms can be added to each individual monitoring component.

Conclusion

The Arduino Mega 2560 R3 was selected as a central micro controller to automate the monitoring and actuation of the functions soil moisture, PH, light intensities, and temperature. The effects of these parameters on the growth of plants would be documented live through video/ image capture of the plants of interest at preset periods. Measured parameters were recorded live on an Arduino Data Logging Shield equipped with its own memory card, and a copy of the data was transmitted wirelessly live through an XBee Pro 60mW Wire Antenna - Series 1 (802.15.4) to the base control computer housed at the Biology Department. The system focused on monitoring all changes in the parameters of interest and on data collection. It also allowed for watering the plants using electromechanical solenoid valves controlled by the Arduino board; controlled light intensities using project directed lamps and temperature inside the green house by interfacing with the building control system. The full project was carried out by a team of students doing their senior projects, which is a requirement for graduation From the physics department at Baker University. Complete documentation of algorithms, materials used, methods, programming, and integration of the whole project is reported here in this paper. All stated objectives for this project were achieved. The interface allows student experimenters to control the actuation of the watering system and the light monitoring are at their discretion. Research performed by faculty and students will require unique growth parameters that can be tailored through programming the Arduino microcontroller system.

Acknowledgments

This project was supported by the Baker Faculty & Student Grant Project, the Biology Department Student Research Grant Project, and the Physics Department Research Support System.

References

Atomic-Market, Soil Moisture Meter Testing Module, Soil Humidity Sensor Compatible with Arduino. Retrieved from https://www.amazon.com/Moisture-Humidity-Compatible-Atomic-Market/dp/B00TMD43BS

Vktech, 2W-200-20 3/4 Inch Brass Electric Solenoid Valve Water Air Fuels N/C DC 12V. Retrieved from https://www.amazon.com/Vktech-2W-200-20-Brass-Electric-Solenoid/dp/B00ETPGB8M

Adafruit, Plastic Water Solenoid Valve - 12V - 1/2" Nominal. Retrieved from https://www.adafruit.com/product/997

Goeasybuy, 20pcs Photo Light Sensitive Resistor Photo resistor Opt resistor 5mm. Retrieved from https://www.amazon.com/goeasybuy-Sensitive-Resistor-Photoresistor-Optoresistor/dp/B01CGCNO34/ref=sr_1_1?ie=UTF8&qid=1471905917&sr=8-1&keywords=photo+resistors+GM5539

Apollo Horticulture, 60 Watt CFL Compact Fluorescent Grow Light Bulb for Plant Growing - 6500K. Retrieved fromhttps://www.amazon.com/Apollo-Horticulture-Compact-Fluorescent-Growing/dp/B00PHLHD3Y/ref=pd_sim_sbs_60_10?ie=UTF8&psc=1&refRID=X1VCCRCJ4XVB14XQAZ17

GEREE, 4 Channel DC 5V Relay Module for Arduino Raspberry Pi DSP AVR PIC ARM. Retrieved from https://www.amazon.com/GEREE-Channel-Module-Arduino-Raspberry/dp/B01DDJYNDE/ref=sr_1_16?ie=UTF8&qid=1471908005&sr=8-16&keywords=solid+state+relays+for+arduino

Adafruit, AM2302 (wired DHT22) temperature-humidity sensor ADA393. Retrieved from https://www.amazon.com/Adafruit-AM2302-temperature-humidity-sensor-ADA393/dp/B00SK6940I/ref=sr_1_2?ie=UTF8&qid=1472359070&sr=8-2&keywords=AM2302+DHT22

Next, 2-Axis FPV Camera Cradle Head + OV7670 Camera Set for Robot / R/C Car ARD0856. Retrieved from https://www.amazon.com/Next-2-Axis-Camera-Cradle-ARD0856/dp/B01CXNYDH2/ref=sr_1_21?ie=UTF8&qid=1472365914&sr=8-21&keywords=OV7670

Marker plot, Data Acquisition and Graphical Plotting Software for Microcontrollers. Retrieved from http://makerplot.com/introducing_makerplot.htm