Sergio Morales
SID: 861027158

CS 193: Design Project
Summer Study Abroad 2014:
Final Project Report

SUMMARY

The Integrated Appliance System ended up being implemented using ATmega1284 microcontrollers according to embedded system principles learned in previous courses, as listed in the System Design Plan and Schematic. In the end, some goals changed, such as attempting to install the IAS control and interface onto the Solar Thermal Closet Dryer, and several results were achieved differently than planned. Despite this, I was able to design and implement a system that is able to control or simulate the appliances.

In terms of the user interface, I used a concurrent synchronous finite state machine to allow the user to input options that allow the sensor data displayed, switches to actuate controls, or data to be sent to the Android application. For logging temperature and humidity data for the closet dryer, I ended up using one wire protocol to get the data in Celsius using the DHT22 sensor. Similarly, the protocol was used for the water heater's waterproof temperature sensor. Any information processed from this microcontroller to the one that handled controllers went through the USART communication. I ended up using this for sending flags in order to control the fans, as well as the motors and the signal to receive or transmit data through the Bluetooth module.

When controlling the fans, I went through several different trials and errors to get to the resulting method. At first, I tried using a MOSFET that could handle high voltage and current, but ended up having trouble powering the fan at the normal 12-volt speed. Because of this, I tried a relay that I had available thanks to the Chemical Engineering senior design team in charge of IAS. The problem with this however, is that it didn't allow me to control the speed of the fan. After careful research on specific transistors, I came across a logic-level n-type MOSFET. It

ended up allowing me to power the fan from the microcontroller at full speed. From this, it allowed for the variation in current through Pulse Width Modulation, and thus changing the speed on the fan. So in the end, I was able to set up the microcontroller in a way that could implement automated control of the Closet Dryer's temperature.

As for setting up the microcontroller to log data, I was able to utilize an Android device to set up communication to the IAS. This allowed me to be able to send a signal to the microcontroller that managed controls, such as the fan and the stepper motors. Additionally, from the user interface, I was able to set up an option where the user can send the the data gathered from the sensors to the Android device. This was done by relaying information to the controls microcontroller, and from there to the Android through the Bluetooth module in chunks of 8-bits. An attempt was made to create a full Android application dedicated to the IAS, but in the end due to constraints, I simply used a Bluetooth terminal application that handled connection to the Bluetooth module for me. An Android Studio application was produced that connects to the module, but is not able to handle sending data yet.

RESULTS

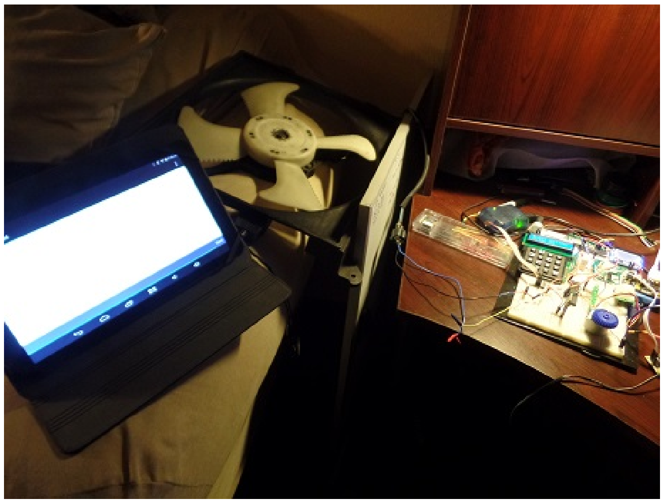
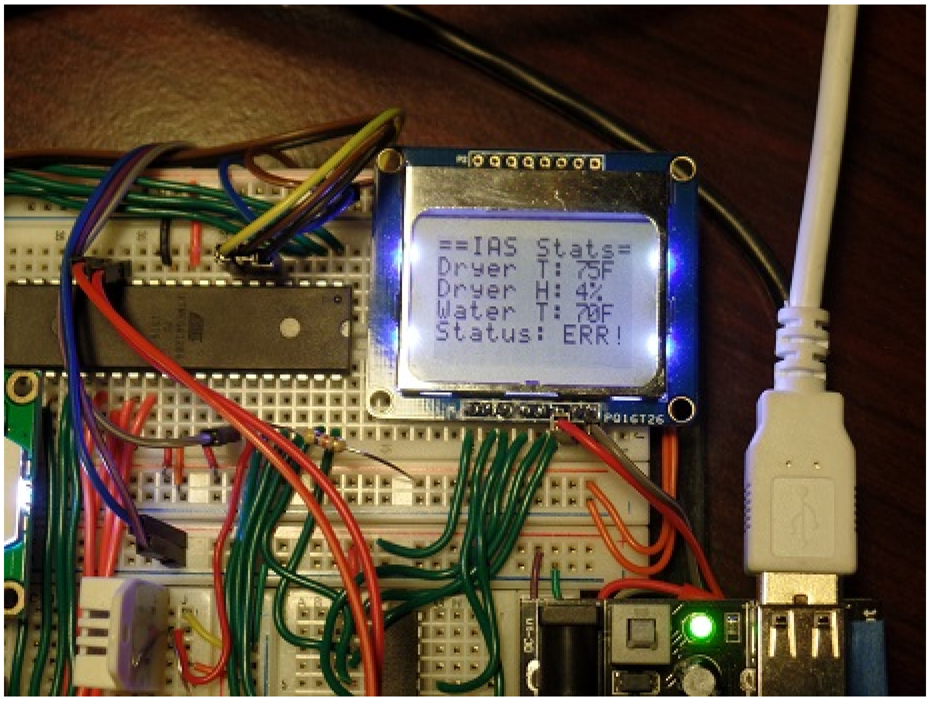


Figure 1: The IAS prototype with an Android device and the 12V fan.

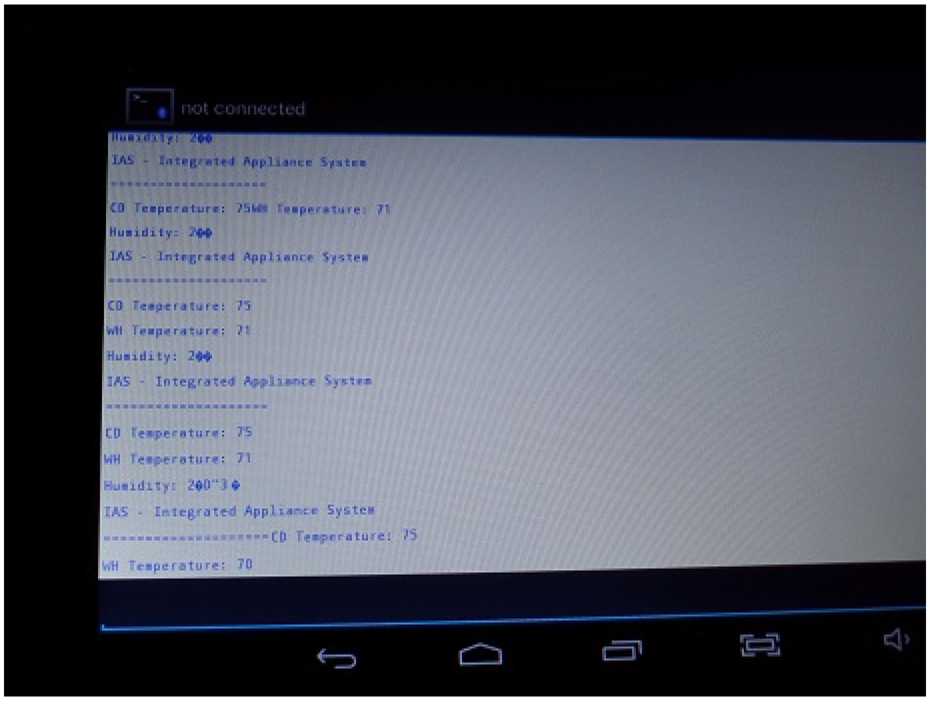
As a result, I was able to create a working prototype for the interface and controls aspect of the Integrated Appliance System. As pictured above, I utilized all the components mentioned in the design plan. Turning on the system booted up the menu for the IAS, allowing the user to set certain controls or options. Among them, the user may view sensor data, actuate appliances, send data reports, and change some options, such as Celsius to Fahrenheit. Note that for actuating certain appliances, such as the Water Heater, Space Heater, and the A/C, they are only simulated in this system, as the focus was set for mainly on the Closet Dryer, in which the fan was the main component. The following gives an example of what the end user will be presented if they click on 'IAS Stats' menu from boot up:



```
==IAS Stats=
Dryer  T: 75F
Dryer  H: 4%
Water  T: 70F
Status: ERR!
```

Figure 2: Temperatures, Humidity, and the Status of motion detected.

Besides being able to control the appliances, gathering sensor data and making it savable was another important aspect of the system. This proved to be a challenge however, as the Bluetooth module served to be unstable at some points, and relaying characters from one microcontroller to another, and finally to the phone via Bluetooth via USART produced a lot of bugs that needed fixing. Fortunately, I was able to get the Android device to receive temperature and humidity data. Thankfully, the C program I developed had no problem sending ASCII format letters to the Android Bluetooth Terminal application, as pictured below:



```
Humidity: 2��
IAS - Integrated Appliance System
====================
CD Temperature: 75WH Temperature: 71
Humidity: 2��
IAS - Integrated Appliance System
====================
CD Temperature: 75
WH Temperature: 71
Humidity: 2��
IAS - Integrated Appliance System
====================
CD Temperature: 75
WH Temperature: 71
Humidity: 2�0"3�
IAS - Integrated Appliance System
====================CD Temperature: 75
WH Temperature: 70
```

Figure 3: Data received from the Bluetooth Report function on the interface.

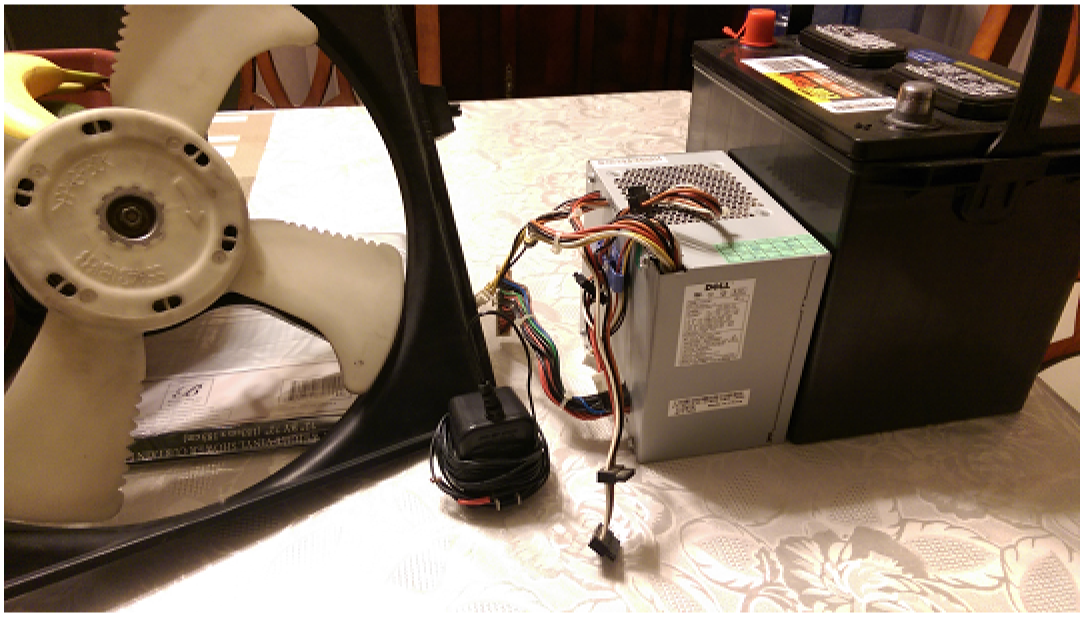


Figure 4: 9V adapter, PC power supply, 12V car battery for power to the fan.

As for the fans, I experimented with a few different options. In the end, I wound up using the Dell power supply for the 12V because it was a safer option, and worked well in conjunction with the MOSFET in terms of wiring up the fan.

CONCLUSIONS

In conclusion, there were several things that resulted differently from what was expected based off of the design plan and schematic. First off, for the data report that is to be generated when selected through the IAS interface menu, which was implemented, but not to the degree that I expected, specifically with providing energy usage. I was planning on utilizing average energy usage (kWh) based off of residential appliances, but faced some difficulties trying to implement that in my program, so in the end, the report only included temperature data. Moreover, the Bluetooth Terminal application sufficed in terms of controlling and getting sensor data remotely, but what I had initially planned was on creating an Android application myself, through Android Studio. In the end I was able to get my application to connect to the Bluetooth module, but did not have enough time to finalize the parsing of sensor data and sending signals to control the fan and motors.

Other factors influenced the resulting IAS controls and interface, specifically with the advancement with the collaborating senior design team. For one, we had planned to install a working prototype of the dryers, which shifted the focus of the controls and interface to work mainly with the fans, because I would not have to simulate the controls as I would have with the other appliances. In addition, they opted out for the Air Conditioning aspect of the IAS in the end, and ended up refining the solar collector design, which changed the expected temperatures that the system would work with. So towards the end, I was preparing to install my prototype in an attempt to test how practical the fan controls would be. I was hoping to try to create a PCB to minimize the size and get rid of the breadboards as well. Unfortunately due to issues with the contractors unable to position the closet dryer in an area where the solar collector could draw enough heat, this delayed our attempt to try to control the fans.

In conclusion, the Android application and the energy reports were the only things that I was unable to refine to an extent to which it would satisfy the planning and design that I had made previously. One thing I plan on doing is to work more on the Android app, and be able to test out the controls by installing it on the Closet Dryer prototype within Spring quarter.