|  |  |
| --- | --- |
| Low Level Design | March 5RHGGW |
|  |  |

 by

 Christopher Bitler

 Walter Bruen

 Marcin Morys

Table of Contents

**Introduction**1

**Problem Statement & Proposed Solution**1

**System Description & Block Diagram**3-4

**System Requirementes**

Overall System5

Subsystem5-6

Future Enhancements6

**Low Level Design**

Schematics7

Standards8

Testing Plan8

Software Design8-12

**Bill of Materials**13

**Conclusion**14

**References**15

We as a society have become increasingly fast paced, and as a result we have less time to perform our daily chores. The chore of washing clothes has always been time-consuming and reoccurring. Needing to gather dirty clothes, wash them, dry them, and sort them can be a weekly hassle. Fortunately, much progress has been made to decrease the amount of time needed to be spent. From using a washboard and clothesline, we have advanced to automated washing machines and dryers. However, we the members of team Renewable Hybrid Greenhouse Guitars for Windows (RHGGW) believe there is still potential to reduce the amount of time this chore can take, and the potential to make the washing process less of a hassle overall.

**Problem Statement & Proposed Solution:**

**Introduction:**

Students living on Notre Dame’s campus only have two options when it comes to doing the laundry. They can either send their clothes out to the [St. Michael's Laundry](http://www.nd.edu/~stmikes/index.shtml) service, or they can do their own laundry using their dorm’s facilities. For those choosing to do their own laundry, using the dorms’ washing machines and dryers can be a frustrating experience. One thing we have found to be particularly frustrating is when a student takes the time to gather laundry and walk to the laundry room only to find that all the machines are in use. This problem is especially common on the weekends, a time when doing laundry is popular, and can even occur more than once per day per person. This nuisance is further exacerbated by the fact that there is only one laundry room per dorm. Because of this fact, a student might be in one corner of a dorm, on the top floor, and have to walk the entire length and height of the dorm just to get to the laundry room in the basement. Nobody wants to make this walk, particularly in the larger dorms like Dillon Hall, only to have the walk to be in vain. The problem basically boils down to not knowing if any or how many washers and dryers are available in a given laundry room.

To solve this problem, we will be creating a device that senses the amount current being drawn from the dryers and washing machines. This information will then be used to determine if a particular machine is running, and if it is, the machine’s start time will also be determined. This will allow us to determine the amount of time remaining before a load is finished. We intend to have this information available to all students by creating a website with the real time information mentioned earlier. By creating this website and notification system, washing one’s clothes should be more hassle free. With this system in place, students can check the website before going to the laundry room to ensure availability. In addition, the notification system might also reduce the amount of time clothes spend in the washer or dryer because it will help to prevent students from forgetting they have clothes being washed.

**System Description & Block Diagram:**





 A 120 volt alternating current power source will power both our sensor board and a washing machine or dryer. The power that goes to the laundry machine will also be passed through a current transformer, which will generate a current proportional to the current passing through it. This goes to an AD converter. The number generated by this will go to the microcontroller, which will process this information to determine the status of the machine. This information is transmitted via a ZigBee transmitter to the Ethernet board, which is powered by another 120 volt AC source. The information is received via another ZigBee antenna, and it passed to the SitePlayer via an RS232 serial interface. The SitePlayer is connected to the internet via an Ethernet port, whereupon a student may via the web page that is hosted on the SitePlayer.

**System Requirements:**

Requirements marked with an ‘\*’ will not be displayed in May

**Overall System:**

* Will sense the current draw of a washer or dryer
* Will communicate with Ethernet board a machine’s state
* Will display data on the real-time usage the washers and dryers

**Subsystem:**

* Web Interface
	+ Website will display which washers and dryers are in use
	+ Will display how much time is remaining on each machine in use
	+ Will display information in real-time
	+ Will be accessible via Firefox web browser
	+ Will be accessible from Notre Dame’s network
	+ User will be able to choose appropriate dorm from a list of dorms on campus
	+ Will be able to display real-time data on two adjacent washers while idle and while they run independent cycles
* Ethernet Board
	+ Ethernet board will receive information from up to 20 sensor boards via ZigBee\*
		- Will receive information from up to 30 feet
		- Will receive information from several sensor boards via ZigBee
		- Information received will be the current state of a washer/dryer
	+ Will act as the server for a dorm’s website
	+ Will be powered via a wall outlet
	+ Board will run off of 840 mW
	+ Will accept information sent serially from sensor boards
* Sensor Board
	+ Sensor boards will transmit information to Ethernet board
		- Information will be transmitted serially
		- Information will transmitted via ZigBee
		- Information will contain the current state of a washer/dryer
	+ Will be powered via wall outlet
	+ Will run off of 200 mW
	+ Will have a plug for a washer/dryer to plug into
	+ Will sense changes in current usage of a washer/dryer
	+ Will process current usage to determine state of a machine accurate to at least 1 minute
	+ Will be small enough to fit behind the washers/dryers in a dorm

**Future Enhancements:**

* Website will be accessible via other major browsers (IE8, Safari, Chrome, Opera)
* System will be able to notify students that a load of laundry is finished via email or text message

**Low Level Design:**

**Schematic:**

****

**Standards:**

Sensor board to Ethernet board will use the IEEE 802.15.4-2003 standard (ZigBee).

Ethernet ZigBee to SitePlayer microcontroller will use an RS232 serial standard.

The SitePlayer will connect to the internet via IEEE 802.3 standard (Ethernet).

**Testing Plan:**

-We will test both the sensor and Ethernet boards’ ability to run off of a 120 V alternating current sources by powering the boards with 120 V and making sure they still run properly

-We will test the current sensor by making sure the current transformer generates a current proportional to the current running through it

-We will also test the current sensor by running the power line for a high power device, such as a soldering iron, and measuring an increase in output current from the current transformer

-We will test the AD converter by connecting an LCD to view the digitally converted analog signal from the current transformer

-We will ensure the ZigBee transceiver on the Ethernet board will be able to receive messages from more than one server board by setting up several transmitters and sending information from the boards simultaneously

-We will ensure the ZigBee can communicate to the SitePlayer by making a webpage display different items based on the value sent over the ZigBee transceivers

-We will ensure the SitePlayer can display web pages by creating a basic website viewing it via the Firefox browser

**Software Design:**

ADC:

The project requires a way to sense whether the washing machine or dryer is on or off. For this we have used a current sensor and an analog to digital input pin. This needs to be accurate enough to tell whether the machine is idle or not. The functionality could also be expanded to tell which cycle the machines are on to better predict how much longer they will be in use. The current running through the line is transformed using a current transformer. The other end of the transformer is connected to a small and a larger resistor. The small resistor is current limiting resistor for a backwards diode connected parallel to the larger resistor. The voltage is measured across the resistor and diode. Since this signal is still AC, it is sampled over a second at pin A0. The maximum value is used as the output of the ADC. The function ADC\_check() outputs the maximum value of this sampling. The function init\_ADC() sets up the registers for analog to digital conversion. This subsystem has been tested using first DC voltages on the A0 pin and then by connecting soldering irons to the plug used for this. Eventually, this will be tested on actual washers and dryers.

ZigBee:

 The project requires communication between sensor boards and the host microcontroller board. A ZigBee wireless daughter card has been used for this. The microcontroller board communicates to the daughter card using SPI at 5 MHz. Pins C4, C5, and C3 are used for the MISO, MOSI, and clock respectively. The microcontroller board is the master and supplies the clock. There is also a microcontroller pin that goes from high to low to signal the start of a communication and from low to high to signal the end. Communication to the microcontroller consists of setting registers and writing to the ZigBee daughter card’s buffer. There is also a sleep pin and a reset pin connected to the ZigBee that can respectively be used to reset the ZigBee and to change state without writing to the control register. There is also an interrupt pin that the ZigBee uses to signal interrupts. The 802.15.4 protocol is being used for ZigBee to ZigBee communication. This project uses the data message protocol to send three types of messages: sensor to host address requests, host to sensor address assignments, and sensor to host status communications. This project uses automatic hardware supplied acknowledgements that follow the ZigBee protocol. Examples of our messages are below. The ZigBee communication has been thoroughly tested by sending and acknowledging all types of messages while watching using a ZigBee sniffer. The communication flow has also been watched after integration with our overall software and has performed very robustly.

802.15.4 Protocol for Data Messages

Length

FCF

Sequence #

Addressing

Payload

FCS

0x0D Allows space for all message types

Binary: 1000 0110 1000 1000

Starts at 0 and increases by one after every message

Includes destination address, destination pan address, and source address

See table for message types

Automatically created at the hardware level

Types of ZigBee Messages

Type of Message

Data sent

Washer address request

0x15

Dryer address request

0x27

Address assignment

2 byte address

Sensor Data

2 bytes of data

Message types are determined by host and sensors based upon the source and destination addresses

Communication:

The sensor boards receive their addresses from the host microcontroller board using a homemade DHCP like protocol. Assigned addresses for washing machines start at 0xB6B1 and the addresses for dryers start at 0xA3B1. When a microcontroller requests an address, they send a message informing the host what type of sensor they are. The host recognizes this as an address request by checking the source address and seeing that it is out of the range used by washers and dryers. In the future, a code for address request may be included in the data portion of the message to improve robustness. Address assignments are then sent to the sensor boards. These messages are easily recognized as address assignments because they are the only message type sent from the host board to the sensor board. Sensor boards are only in receive mode directly after requesting an address and receiving an acknowledgement. Data messages are then sent from sensor boards to the host board. Since the sensors now have a designated address, all of their messages are now understood to be data updates. To improve robustness in the future, a code may be included before the sensor data in the payload portion of the ZigBee message. This system has been tested successfully as part of our overall system. Host and sensor boards have been started up in a variety of orders and been left running for extended amounts of time without any failures.

Sensor Board Operation:

After a reset, the sensor board initializes all of its components and then begins sending messages every 30 seconds requesting an address from a host board. After address reception, the board enters a main while loop. In this loop, the ADC data is sampled constantly. Data updates are sent to the host board after a change in ADC value by a noticeable amount or after about 20 seconds as determined by a timer interrupt. The minimum amount of change in the ADC value that requires for an update message will be determined after sampling real data from a washing machine and a dryer. If this does not receive acknowledgements from the host board after sending multiple messages over a 15 second period, it will clear its address and start sending address requests. This is in case the host board is restarted or loses power. This system has been tested successfully as part of our overall system. Host and sensor boards have been started up in a variety of orders and been left running for extended amounts of time without any failures.

Host Board Operation:

After reset, the host board initializes all of its components and then waits for 60 seconds. This 60 second delay ensures that any sensor boards connected to the host before a reset will reset their old addresses. The host board then turns its ZigBee to receive mode and waits for messages. It is important to note that when setting addresses, the host board looks at its address array for 0 values. These addresses are considered inactive and are used for new address assignments. Other important information is located on the flow chart. This system has been tested successfully as part of our overall system. Host and sensor boards have been started up in a variety of orders and been left running for extended amounts of time without any failures.

Host Board to SitePlayer Communication:

This communication uses a standard serial asynchronous connection at a baud rate of 9600. This communication is one way. Only the host board can send data. Data transmissions follow the protocol set out by the site player board. An 0x80 message is sent first to signify a write to memory of one byte. This is followed by a one byte transmission of the address in memory to write to and finally a one byte transmission of the data to be written. This code has been successfully tested by checking the data in memory locations on the site player with PC software supplied by its manufacturer.

Overall code will be tested by adding multiple boards to the network and testing how it responds after powering on and off boards in multiple different orders. Important cases are how the network responds when a sensor is added or removed, when the main board loses power and is restarted, and when all boards lose power and are restarted.

**Bill of Materials:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type** | **Description** | **Generic part #** | **Package** | **Distributer par #** | **Quantity** |
| Voltage Regulator | 5 V | LD-1117 | DPAK | D: 497-1237-1 | 1 |
| Voltage Regulator | 3.3 V | LD-1117 | DPAK | D: 497-1235-1 | 1 |
| Ceramic Resonator | 20.0 MHz | Murata CSTCE20M0V53Z-R0 | SMD | D: 490-4717 | 1 |
| Schottky Diode | 20 V1 A low Vf | - | SOD-123FL | D: MBR120VLSFT1GOSCT | 2 |
| .1 uF Capacitor | 16 V, XR7 | - | 0603 | D: 399-1096-1 | 8 |
| 10 uF capacitor | 16 V, Y5V | - | 1206 | D: 445-4042-1 | 2 |
| 300 Ω Resistor | 1/8 W | - | 0805 | D: P300ACT | 5 |
| LED SMT | Green Clear | - | 0805 | D: 160-1414 | 2 |
| 10k Resistor | 1/8 W | - | 0805 | D: P10.0KCCT | 1 |
| Push Button | Available | - | - | - | 1 |
| 100 Ω Resistor | 1/8 W | - | 0805 | D: RMCF1/10100FRCT | 5 |
| 2x5 Pin Connector | Available | - | - | - | 2 |
| LED SMT | Red Clear | - | - | D: 160-1415 | 2 |
| Voltage Regulator | 5 V | LD-1117 | DPAK | D: 497-1237-1 | 1 |
| 2x10 Pin Connector | Available | - | - | - | 1 |
| PIC18F6722 | PIC Microcontroller | PIC18F6722 | TQFP64-10x10 | Microchip | 1 |
| 470 uF Capacitor | 16 V | - | Radial | D: 493-1286 | 3 |
| 180 pF Capacitor | 50 V | - | 0603 | D: 311-1072-1 | 2 |
| 1.0 uF Capacitor | 16 V, XR7 | - | 0603 | D: 587-1241-1 | 1 |
| Ethernet Jack | Magnetics included | J00-0062NL | - | D: 553-1480 | 1 |
| 10 pF Capacitor | 50 V | - | 0805 | D: 311-1099-1 | 2 |
| Power Transformer | 120V AC-9V AC | BV030-7151.0 | - | D: 567-1030-5 | 1 |
| Bridge Rectifier | Available | - | GBU4S | - | 1 |
| Current Transformer | 1000:1 60Hz Transformer | - | - | D: TE1010 | 1 |
| 1k Ω Resistor | 1/8 Watt | - | 805 | D: RHM1.00KCCT | 1 |

**Conclusion:**

Using the design described above, we plan on creating a sensor network to remotely monitor the state of the washing machines and dryers located on campus. We believe this could aid in the efficiency of the laundry process. Some other schools colleges and universities already have such a system set up. Ours will be unique in that it is student designed, and can be easily retrofitted to older machines instead of having to insert additional circuitry into preexisting circuit designs.

**References:**

**Spec Sheets:**

PIC18LF6722 Microcontroller:

<https://twiki.cse.nd.edu/twiki/pub/EESD0910/TaskStuff/18F8722.pdf>

AT86RF231 ZigBee controller

<https://twiki.cse.nd.edu/twiki/pub/EESD0910/ZigBee/AT86RF231.pdf>

**Software Manuals:**

SitePlayer

<http://www.netmedia.com/siteplayer/docs/001212/SitePlayer_Software_Manual.pdf>