Project: Reprise of Locker Access System

(Design Document)

Group May 14–12

Mohammed Al Kaabi

Sherry Elsa Gungat

Nurul Izni Hazimi Abdul Aziz

Shichao Su

Client

Iowa State University (ECPE)

Advisor

Harker, Leland Edward

04/29/2013

Table of Contents

1. Problem Overview

1.1 Problem Statement

We are doing this project in order to solve problem regarding storage and lockers assignment in senior design lab in Coover 1301. Currently, the lockers are secured with standard combination padlocks. The disadvantage of using this type of lock is that the previous users may still know the combination number. Besides, some students might forget the combination number for their team's locker. Other than that, the padlock is not intuitive or not easy to use especially for first time users. Furthermore, administrators are not able to assign and update lockers easily and efficiently.

In order to solve these problems, we are creating a control system that allows locker access to authorized users by using their ISU card. The locker access system has to be easy to use while maintaining a basic amount of security. We are designing one user panel that communicates wirelessly with locker modules. The entire system will be comprised of two basic elements: a control box, and the locker module. Besides, this control system also allows administrators to access list updates and locker overrides. With this control system, it will be easier and safer for students to store their project and access their lockers.

2. System Overview

2.1 System Requirement

- The system will the read magnetic strip on the student ID card and unlock the corresponding door. Besides, it also allows manual entry on the keypad on the control box.
- It must allow administrative functionality including access list updates and locker overrides.
- The user panel must have a display that shows the status of the system. It will give direction to user whenever they use this system.
- The user panel will communicate wirelessly with the locker module. The wireless communication must be in two ways communication so that the user panel can send data to locker module in order to open the door. Similarly, locker module can send information to the user panel regarding the status of the battery in each locker module. The wireless data transmission needs to have the ability to transmit data from the longest distance or at least cover the big range of the dimension of the room.
- The battery must last at least one semester and the locker module needs to give warning when the battery is at 20% or lower.
- The SD card must be able to read and store database of names and locker numbers so that it can be updated easily by using computers.

2.2 System Block Diagram

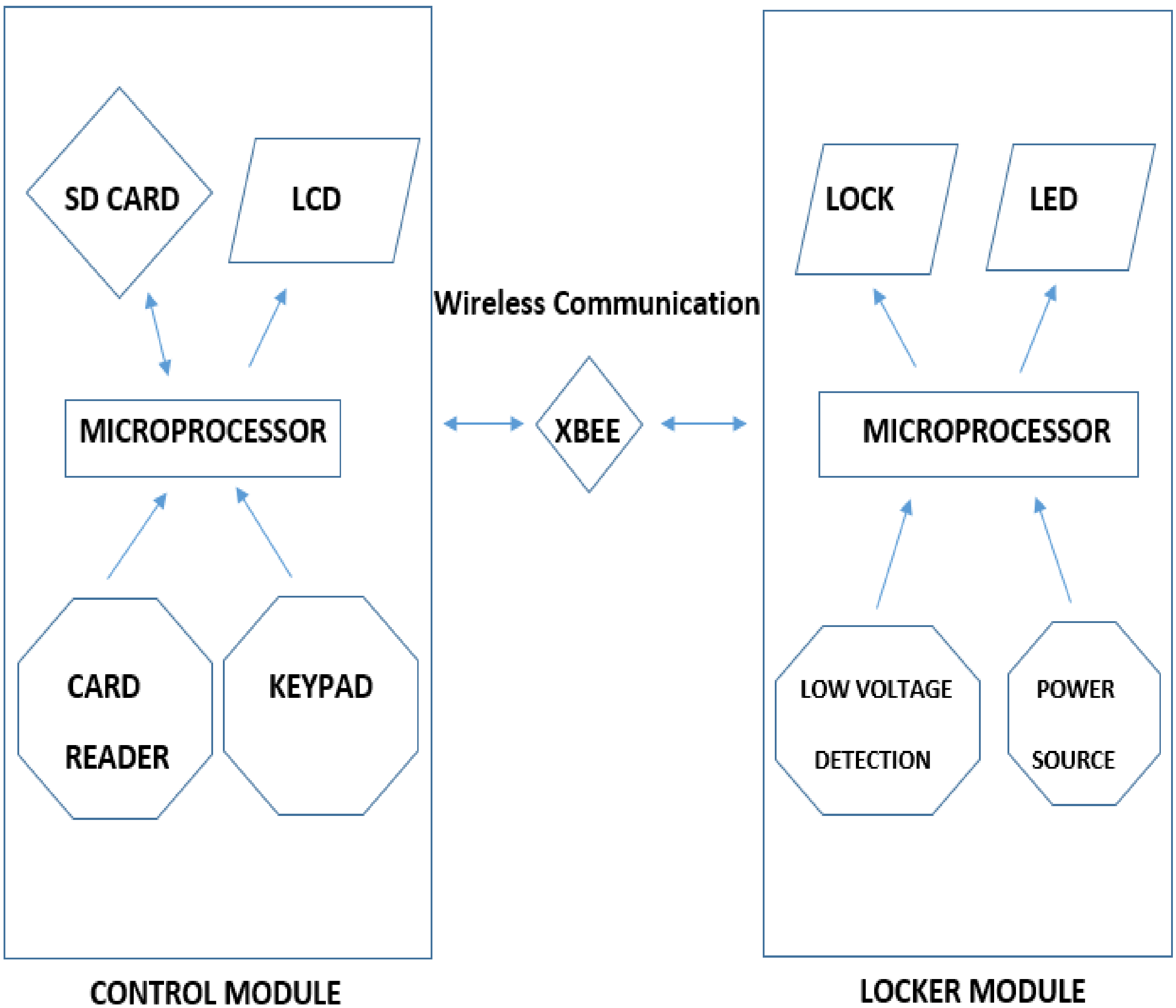

Figure 1: Block Diagram

2.3 Operating Environment

The control box and the locker module will be located inside the senior design room in Coover 1301. The temperature inside this room is approximately 69 to 74 degrees Fahrenheit. The dimension of the room is 44' x 28'. It has a lot of things inside such as tables, desktop, chairs and so on. These can be the obstacle for the communication between the control box and locker module if they are situated far from each other.

3. Design

3.1 Input and Output Specification

Input:

1) Control Module:

A- Keypad (Users will be prompted to enter their student ID number).

B- Card-Reader (Users will be able to swipe their card).

C- Xbee (Receive data wirelessly from the locker module).

2) Locker Module:

A- Low Voltage Detection (circuit that monitors the voltage level of the battery).

B- Xbee (Receive data wirelessly from the control module).

Output:

1)Control Module:

A- LCD Screen (output data in a user friendly form).

B- Xbee (Transmit Data to the locker modules).

2)Locker Module:

A- Low Voltage LED (it will light when the locker has low battery).

B- Servo (Unlock the locker if it receives the signal to do it).

C- Xbee (Transmit Data to the control modules).

Input/Output:

A- Xbee (Will be receiving and transmitting signals both ways to and from the Control and Locker Modules)

B- SD Card (This will be input and output in the Control module. Names, ID number, and locker number will be stored here)

3.2 User Interface Specification

The user interface will operate in different situation for the two intended users which are students and administrative.

I. Students:

Students will have two options to open their lockers. Firstly, they can swipe their ISU card and secondly they can type in their ID number on the keypad. If they have their name registered as one of the locker holders, the corresponding locker will be opened. If they are not registered, there will not be any lockers opened and LCD will show error.

II. Administrators:

Administrators can perform multiple tasks with the control box. They can swipe their ISU card or enter their ID number on the keypad. They will be able to use the keypad to open the lockers simultaneously or individually. Besides, they also can use the keypad to update information on SD card so that they do not have to take the SD card out so often. Lastly, they can pull the SD card out to update information most probably in the beginning and end of semester.

3.3 Hardware Specification

Control Module

- **Microcontrollers**

The terminal and both locker and control modules use ATmega328. All code for our microcontrollers was written in the Arduino Processing C Programmer/Compiler. The ATMEGA328 is a 28 pin microcontroller that we will be able to be removed and reprogram later. This is an important feature since most of the locker module will need to be programmed with their identifiers before they can be installed. Below are the specifications of the microcontroller: Below are the specifications of the microcontroller:

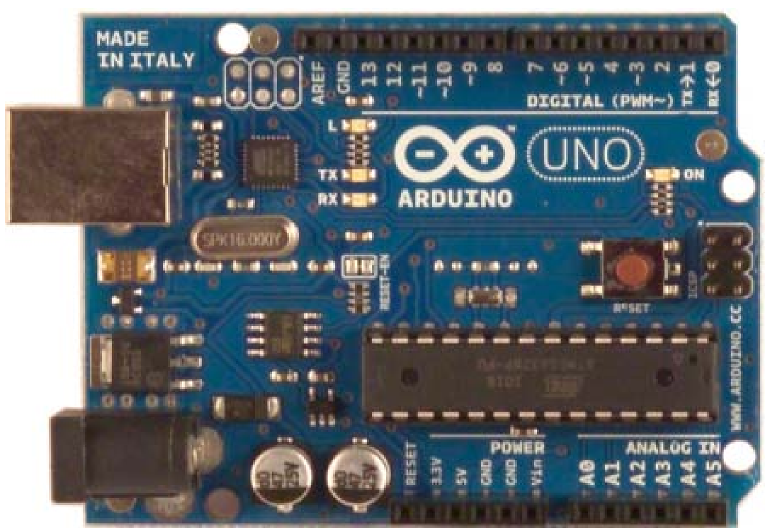

Figure 2: Arduino Uno (WIth Atmega 328 Processor)

Operating Voltage : 5V

Digital I/O Pins : 14

Analog Input Pins : 6

DC Current per I/O Pin : 40mA

Clock Speed : 16MHz

Power Consumption : 0.21 W (at 5V)

❖ Wireless Communication

For wireless communication, we are using Xbee 1mW Trace Antenna – Series 1(802.15.4). This module takes the 802.15.4 stack and allows very reliable and simple communication between microcontrollers. This module can be programmed easily by X-CTU. We set all transmission is done in serial format at 9600 baud, no parity bit, and no stop bit and set all the Xbees in the same IP address. We also used the same Xbee as the receiver and transmitter in the locker modules. The specifications for the Xbee is shown below:

Figure 3: Xbee Series 1

Indoor range : up to 100 ft (30m)

Outdoor Range : up to 300ft (100m)

Transmit power : 1mW

Supply Voltage : 2.8-3.4V

Transmit Current : 45mA at 3.3V

Receive/idle current : 50mA at 3.3V

Power Consumption : 0.15W at 3.3V (for Transmit)

: 0.17W at 3.3V (for Receive)

❖ Keypad

The keypad used is a COM-08653, 12 button keypad. The pins of the keypad are connected to a single analog pin of the microcontroller by using voltage-resistor method. When one of the buttons is pressed, a specific analog voltage is measured at the microcontroller, which will correspond to the button pressed.

Figure 4:12-Button Keypad

❖ Magnetic Stripe Reader

An AP-MSR200 is a simple magnetic stripe reader, which outputs the card number in the form of serial data. The card reader is connected to the microcontroller via female DB9 female connector. The connector cannot be connected directly to the microcontroller because of the large voltage differential between the microcontroller and the connector. This causes issue in communicating with the microcontroller. A logical 0 or low is 0V and a 1 or high is +5V for the microcontroller. Meanwhile, the connector sends -3V to -25V as low and +3V to +25V for high. Therefore, a level shifting chip is needed.

Figure 5: Magnetic Card Reader

- **Level Shifter**

We use a MAX232 serial driver in order to turn the serial output to TTL (Transistor-Transistor Logic). Once it's in TTL form, we can read it at the ATMEGA328 serial input.

- **SD Card**

There is one SD card attached to the card terminal which accepts an SD card with a pre-formatted comma separated value file on it. This CSV file will be read by the ATMEGA328. The card reader will store the information for locker access. Admin will be able to pull the SD card out from the card module to do any information update by using computer.

- **LCD Screen (ADM2004D-FL-YBS)**

The screen will show the status of the system. We will use a 4x20 screen which will be connected to the terminal and 5V power source.

Locker Module

- **The Lock**

The locking mechanism was supplied by the client. It is a metal lock that will be unlocked and locked by the servo.

- **Power Source**

The power source for the locker module is batteries. We will be using four 1.5 V AA batteries.

- **Servo -** 900-00014

This servo is smaller and lighter than the standard servo. It acts as the electromechanical device to open the lock mechanism. The microcontroller in the locker module will send signal to the servo and then the servo will operate the door to open and close completely. Servo consumes quite a lot of power even in idle state. Therefore, we connect it to a transistor that will switch it on and off in order to reduce the power consumption. It requires 4.8 to 6 V to operate. It can operate load with 1.2kg at 4.8 V and 1.4kg at 6V.

Figure 6: Servo

- **Low Voltage Detection Circuit**

This circuit is a voltage divider circuit that will monitor the voltage level in the batteries for locker module. The values of resistors used are both 10k ohm. The microcontroller will read the voltage from analog pin. When the locker has low battery, LED will light up as an indication that the batteries need to be replaced. We set the lowest voltage to be 3.0V and the LED should light up when the battery reaches this level.

3.4 Software Specification

Code Documentation:

Control Module:

All of the initialization needed for connections and devices will be done in the *void setup()* function. At first, we need to initialize the input and output ports of the Atmega328. In our code, *void loop()* stands for the *void main()* with a *while (1)* statement in it. Basically, all of the code should be in the *void loop ()* function. Input from the keypad is taken as a form of 0-5 volts which translates into 10-Bit resolution via a voltage divider. A function *char keypressed(int input)* should take the input from the voltage divider and return the key pressed in a form of character. *int sendData(int address)* is going to be the function responsible of sending data to the Locker Modules. It should take an address as input and return a conformation as a form of integer (ex 1 = done, 2=low battery and 0= connection failed).

Locker Module:

In the locker module the code is much smaller and simpler. It receives a message through Xbee and opens the locker. The module should return a confirmation that the locker is opened. While idle, it should put the system in sleep mode to save batteries. Sleep mode turns off the devises connected to the terminal using a transistor. It should turn on the Xbee for a second at least each 5 seconds to check for signal and then turns it off again. If a signal is received, it should turn on all of the devices and work according to the signal .The locker module should also detect low battery voltage and send a signal to the Control Module to inform it that it is running out of power source.

3.5 Circuit Schematic

❖ **Control Module**

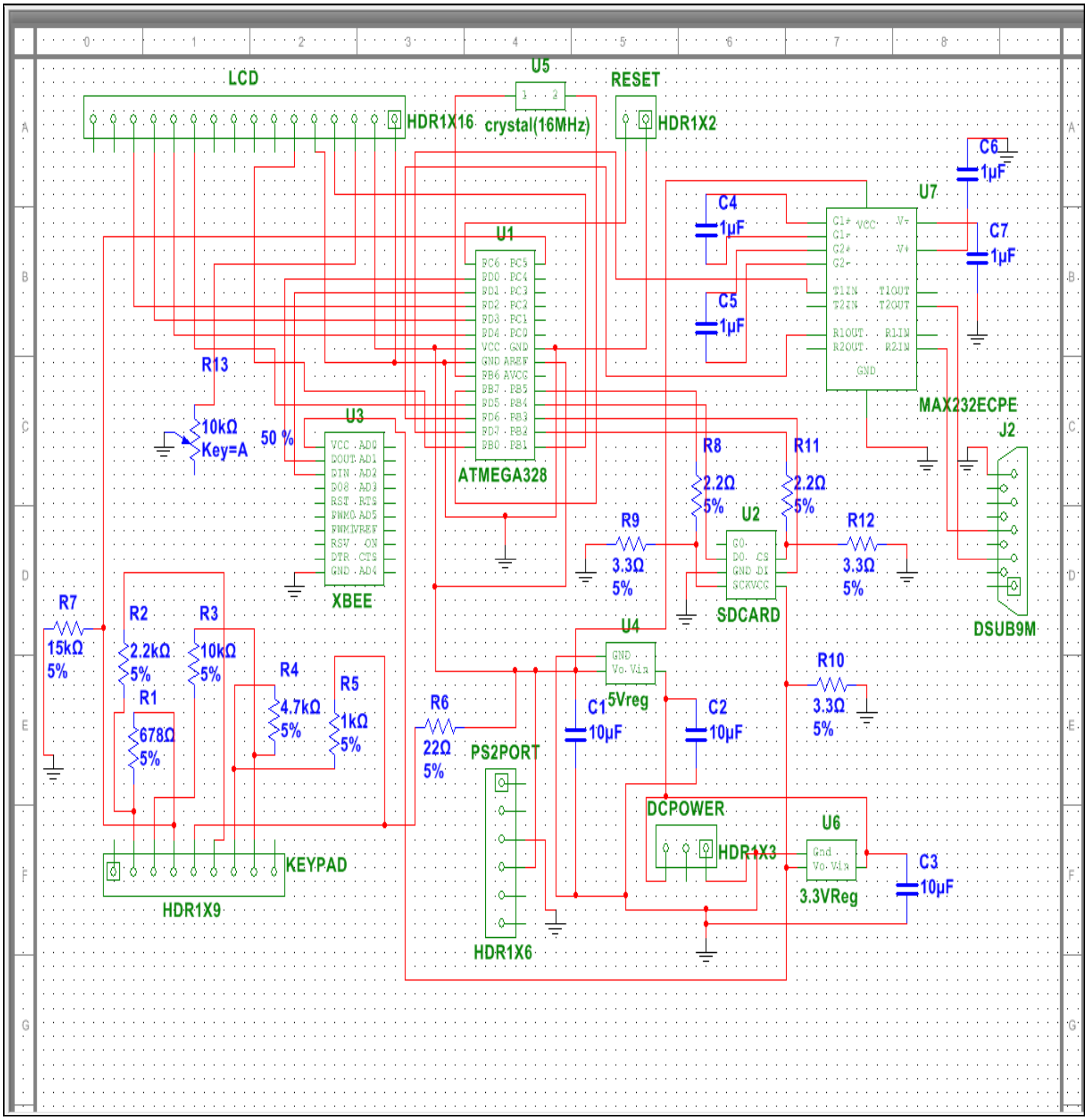

Figure 7: Control Module Circuit Schematic

❖ Locker Module

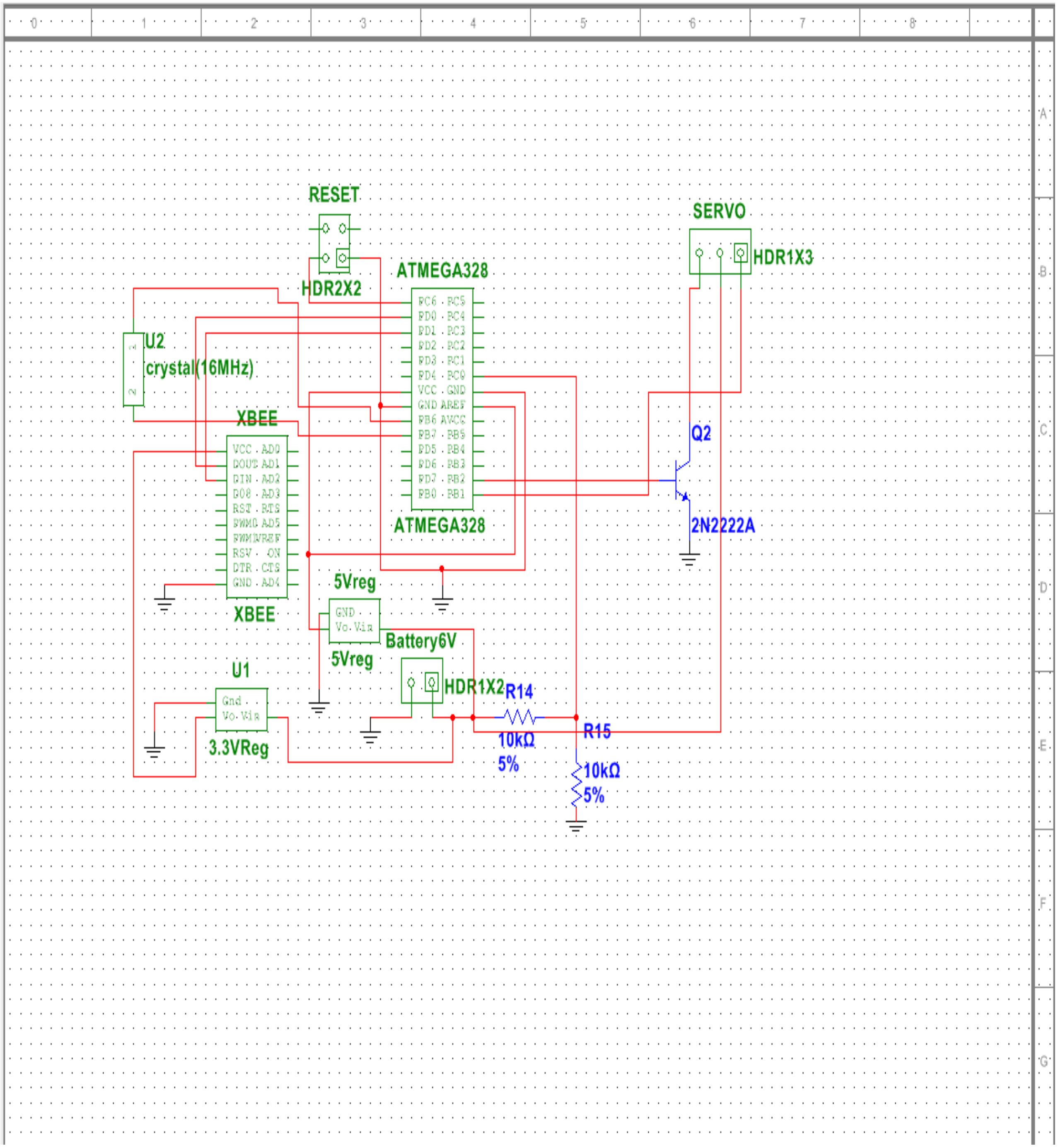

Figure 8: Locker Module Circuit Schematic

3.6 Cost

Parts and Materials	Unit price	Quantity	Total
Atmega328	$5.50	2	$11.00
Xbee series 1	$22.95	2	$45.90
Keypad (COM08653)	$13.95	1	$13.95
LCD Screen (ADM2004D-FL-YBS)	$17.62	1	$17.62
SD card	$9.95	1	$9.95
Magnetic card reader	$44.95	1	$44.95
Servo (900-00014)	$10.95	1	$10.95
Lock	$10	1	$10.00
Voltage Regulator (5V)	$0.95	1	$0.95
Voltage Regulator (3.3V)	$1.95	1	$1.95
Npn Transistor	$0.50	1	$0.50
Resistors	$0.25	14	$3.50
Capacitors	$0.45	4	$1.80
16 MHz Crystal	$0.95	2	$1.90
MAX232CPE	$1.85	1	$1.85
Total			**$176.77**

4. Standard

- **Xbee**

The standard for the Xbee that we are using is **IEEE 802.15.4**. This standard focuses on a low cost wireless communication of nearby devices. It is suitable for our project because the two Xbees (one in Control module and another one in Locker module) will be communicating in a close distance.

5. Testing Process and Result

In order to test the design, we tested it by part. Once we are done with one part and confirm that it is working properly and reliably, we proceed with another part. Then, after testing component by component, we tested the whole system.

- **Keypad**

We tested the keypad by connecting it to analog pin 5 of the microprocessor. LCD screen will display the button that is being pressed and the analog voltage corresponding to the button. We pressed one button at a time and we verified that the LCD screen shows the correct button. It demonstrates that the keypad works.

- **SD Card**

We tested the SD card by storing our information (student ID number) into it by using computer and then connected the SD card to the microprocessor. The microprocessor is also connected to the keypad and LCD screen. After we typed in our student ID number on the keypad, the LCD screen displays *'student ID matched'*. It shows that the SD Card is able to communicate with the microprocessor.

- **Wireless Communication**

We use Xbee for wireless communication. We tested it by connecting one Xbee to the locker module and another one to the control module. Firsty, we tested the communication with the two Xbee are placed close to each other. Then, we increased the distance of communication to 20m. From this two testing, we verified the Xbee can communicate efficiently for both near and far distance.

- **Testing the whole system**

In order to test the whole system, we use one control module and three locker module. We sent signal to each locker module from the control module. When we send signal to locker 1, the servo will unlock the locker. This indicates that the control module are able to differentiate each locker module and communicate with the correct locker.