CPRE/EE 491
MAY15-25
Project Design Document

Advisor: Lee Harkin
Client: Lee Harkin / Department of Electrical and Computer Engineering
Project Title: CyLocker Access System

Group Roles

Team Leader -> Nathan Castek
Team WebMaster -> Corey Coazzato
Team Communicator -> Nathan Lafferty
Team Key Concept -> Priyank Patel
Team Technical Leader -> Mohammad Syazwan

Revision History

Date	Revision	Description
10/15/2014	0.1	Initial Plan Formatting
10/27/2014	1.0	Finalized Plan Formatting

Executive Summary

The goal for this project is to provide an easy to use system for electronic access to senior design lockers. The current lockers are currently inadequate for several reasons, as detailed below. Our group feels that these may be improved in many ways, primarily by introducing an element of logic to the system.

Problem Statement

This project is to solve the problem of security and management of the storage and lockers in the senior design lab in Coover 1301. Currently, the lockers are being secured by padlocks, where the same locks are reused semester after semester. This is an issue as previous students may tamper with current student's projects, as well as the maintenance overhead of manually assigning lockers to students, or removing identifying information on the physical locks to attempt to prevent the security issues.

Table of Contents

System Level Design

System Requirements

Our system must perform under the following requirements:

- System will read student's ISU ID Card, providing access on a single swipe
- System must have a keyboard failback, such that students without ID Cards may still access the system
- System must allow administrative functionality, both local and remote, including altering access lists and overriding functionality
- LCU ("Locker Control Unit") shall be battery-powered and last a minimum of two full semesters
- The LCD display on the MCU ("Main Control Unit") shall display when locker batteries are below 20% and are in need of replacement
- Wireless transmitters must securely transmit data, ensuring only secure access to lockers
- LCU shall be secure, in that only a valid signal from the MCU will open the lock
- The MCU must store usernames/passwords/groups of users, and be easy to remotely manage to alter information
- The MCU must be able to control lockers a maximum of the distance of the room
- The MCU must be "locked-down", such that non-administrative users cannot alter or access any information

Operating Environment

The operating environment of the LCU and the locker units would be a type of classroom or workplace. Users would be able to safely store personal items in a designated locker with an automated lock. This lock would be controlled by the swipe of an id card. The user would be able to either swipe their personal card or type their id number in a corresponding card-reader. This control of access to the lockers is determined by the credentials stored on the LCU located in the room. Additionally, we are assuming a stable operating environment in a temperature-controller room, and we are already assuming physical security. As such, no physical security shall be implemented to the controlling units beyond a standard casing unit. Additionally, we are assuming the environment will be weather-controlled and will not be exposed to the elements.

In more detail, the following environmental conditions are assumed:

- Temperature ranging from 64 degrees fahrenheit to 84 degrees fahrenheit
- Locker dimensions 4' x 5.5' x 17''
- Several room obstacles, hindrances, and blockages of varying materials

Intended Users and Uses

The intended users for this product are students enrolled into Senior Design for the ECPE department at Iowa State University. Additionally users include management staff, who will remote into the MCU ("Main Control Unit") to manage the system by adding/removing user accounts to lockers, and managing battery supply units.

The lockers themselves will be "Bolt-Cutter safe", in that they are designed around the same difficulty as bringing a pair of bolt cutters into the room and cutting the doors off manually. As such, the lockers are intended to be secure from most standard means of digital cracking, and they shall also be physically safe from all but the persistent in physical damage.

Block Diagrams of the Concept

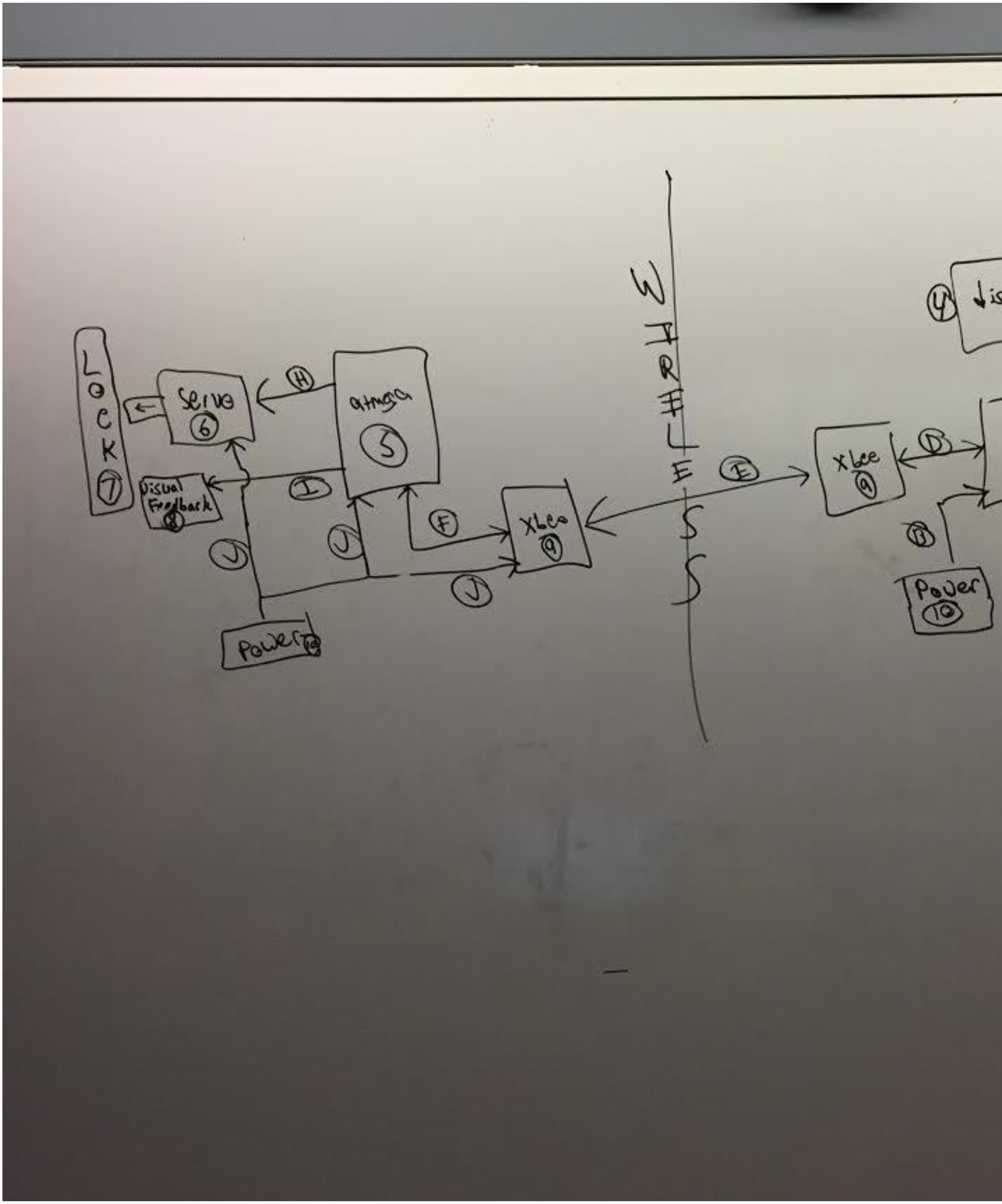
LOCK
7
Servo
6
H
atmega
5
Visual Feedback
8
I
J
F
Xbee
9
Power
10
WIRELESS
E
Xbee
9
D
B
Power
10
4

Figure 1, Concept Block Diagram of Project

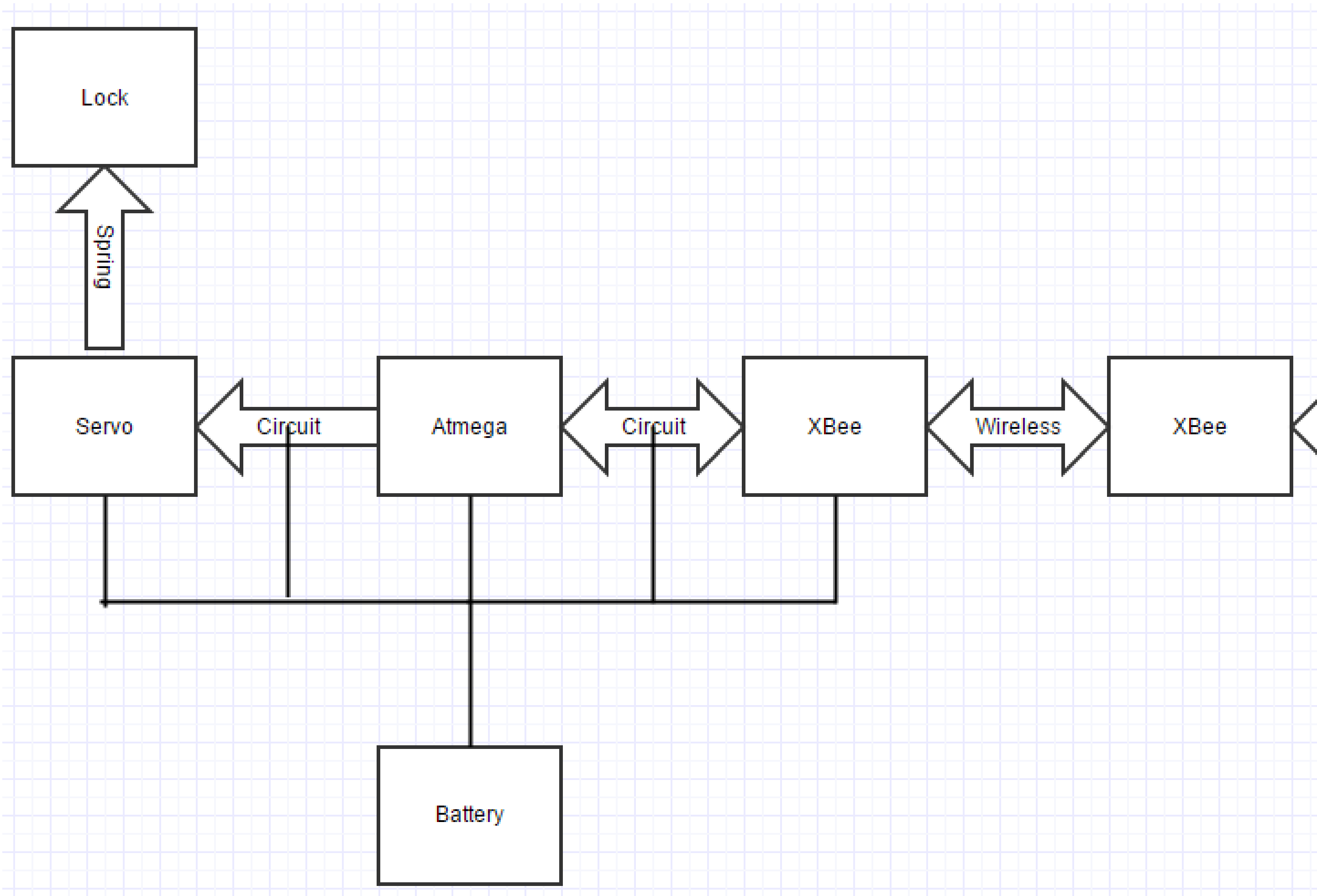

Figure 2, Block Diagram of Project with interconnections

Detailed Implementation

I/O Specification

Our input and output shall be as follow:

Output:

- LCD display on the MCU ("Main Control Unit") shall render username/password information, and display success/fail on opening a locker
- LCD display on the MCU shall render, specifically, which LCU's have critical battery levels
- Physical locker unlocking

Input:

- Magnetic-strip card reader
  - Reads ISU ID Card, providing information on magnetic strip

- Full qwerty-style keyboard for username/password entry

Interface Specifications

Our hardware interface shall be designed with human-usability in mind, such that the lockers are easy to unlock/open. Additionally, the MCU shall be constructed in such a way that the keyboard is easily accessible and typable, and that the LCD is using a large enough font to be legible. The LCD interface itself shall be as simple as two textboxes, one for username and one for password, as well as relevant success/fail messages.

Circuit Specifications

Power Circuit and Design

The power sources for the LCU ("Locker Control Unit") are batteries. The project will be using four 1.5 V AA batteries to power the ATmega328, servo, and Xbee units.

Atmega to Servo Circuit Specification

The I/O of the servo is controlled by the ATMEGA328. The servo is powered from the batteries, through a transistor, to conserve power.

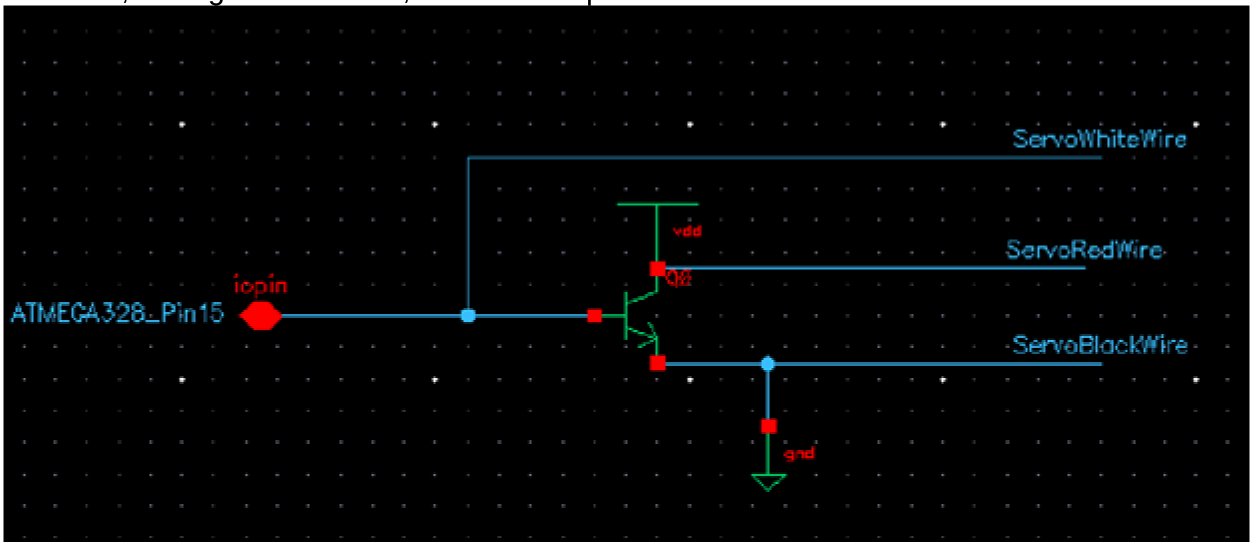

Figure 3, Atmega to Servo Circuit Schematic

Atmega to Visual Output Specification

The visual output (LED) will be powered by the batteries. The I/O of the ATMEGA will control what color the LED is.

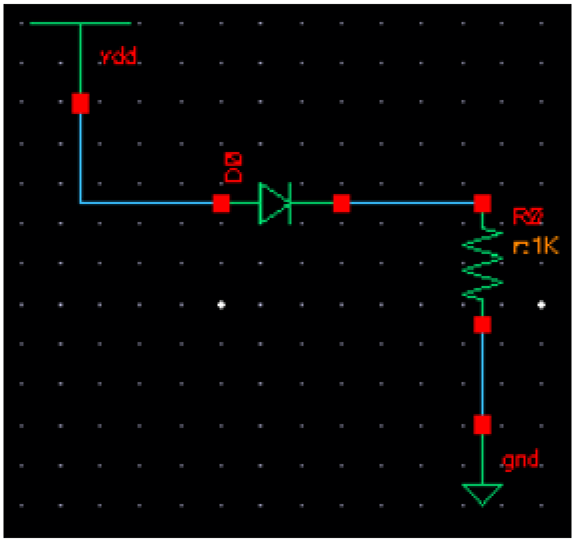

Figure 4, Atmega to Visual Output Circuit

Atmega to XBee Circuit Specification

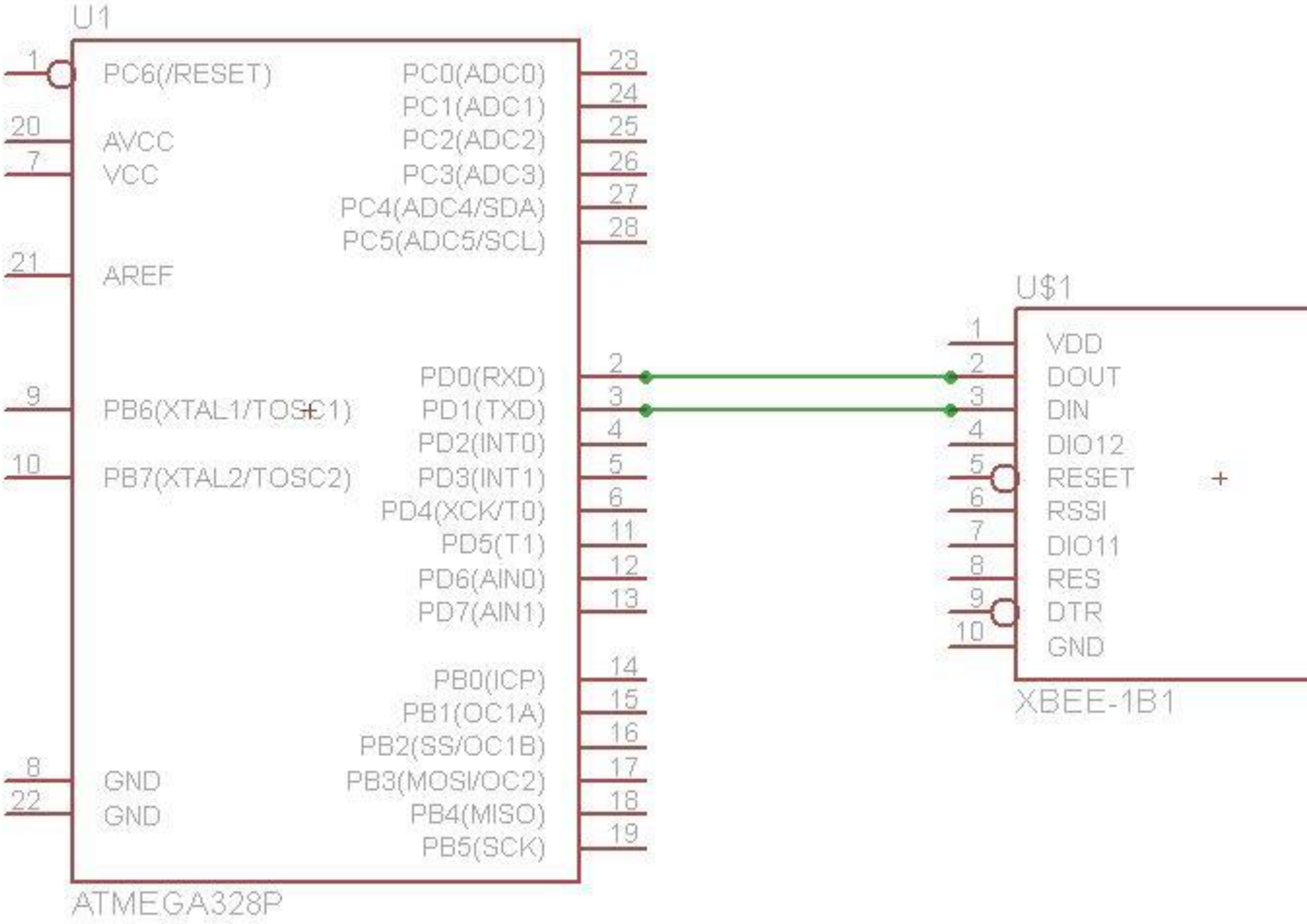

Figure 5, Atmega to XBee pin layout

There are only two pins of Atmega328 connected to the two pins of Xbee. The pins of Atmega328 are PD0 (RXD) and PD1 (TXD) whereas the pins of Xbee are Dout and Din. The Atmega328 and Xbee will be connected to the 5V (Pin VCC) and 3.3V (Pin VDD) voltage supply respectively.

Pi to XBee Circuit Specification

The raspberry pi and the Xbee shall connect using a converter chip, such that the XBee may be plugged directly into the Pi's data pins. Particularly, data pin 1 on the raspberry Pi will connect to DOUT on the Xbee, and data pin 2 on the raspberry Pi will connect to DIN on the XBee. Power shall be provided from similar data connections from the Pi to the XBee.

Pi to Display Connection Specification

The raspberry pi and the LCD Display module connect using a standard RCA cable. Power to the LCD display is connected using a wall-outlet and a special converter.

XBee to XBee Specification

XBee modules are embedded solutions providing wireless end-point connectivity to devices. These modules use the IEEE 802.15.4 networking protocol for fast point-to-multipoint or peer-to-peer networking.They are designed for high-throughput applications requiring low latency and predictable communication timing. XBee modules are ideal for low-power, low-cost applications.

Xbee can be used for wireless communication with low power consumption. It talks with well known UART interface and makes it easy to use. It is simple and straightforward if you only use 2 Xbee for communication. Xbee can communicate up to 300ft and operate on 3.3V @ 50mA. The board contains 6 10-bit ADC input pins and 8 digital I/O pins. Xbee also allows for 128-bit encryption.

Once you are connected to the XBee with the serial tool, the configuration process can begin. Because there are lots of parameters that can be set on the XBee. Here are the most common problems:

- the 2 XBee are not in the same network
- one XBee is using encryption while the other does'nt
- both are using encryption but the encryption key is different

Pin Number	Description	Function
1	VCC	Power Supply
2	DOUT	UART Data Out
3	DIN/CONFIG	UART Data In
4	DO8*	Digital Output 8
5	RESET	Module Reset
6	PWM0/RSSI	PWM Output 0/RX signal Strength
7	PWM1	PMW Output 1
8	[RESERVED]	Do not connect
9	DTR/SLEEP_RQ/DI8	Pin Sleep Control Line or Digital Input 8
10	GND	Ground
11	AD4/DIO4	Analog Input 4 or Ditital I/O 4
12	CTS/DIO7	Clear-to-Send Flow Control or Digital I/O 7
13	CN/SLEEP	Module Status Indicator
14	VREF	Voltage Reference for A/D Inputs
15	ASSOCIATE/AD5/DIO5	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS/AD6/DIO6	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3/DIO3	Analog Input 3 or Ditital I/O 3
18	AD2/DIO2	Analog Input 2 or Ditital I/O 2
19	AD1/DIO1	Analog Input 1 or Ditital I/O 1
20	AD0/DIO0	analog Input 0 or Ditital I/O 0

Figure 6/7, XBee pin layout diagram

Xbee consists of 20 pins. 8 digital I/O pins that allow the device to function with input or output set by software. Of these digital pins 6 can function to give ADC input. 1 pin is designated for digital output only. 2 pins VCC and GND, provide power to the device. 1 pin also allows for voltage reference. The other 8 pins allow the Xbee to perform further functionality. 2 pins allow for UART data in and UART data out. 2 pins can function to allow PWM output. 3 pins are set to either control sleep or reset, one of which allows for a sleep indicator. And the final pin is reserved.

USB Connection Specification

The keyboard and card reader will be connected to the Pi using USB connection. The keyboard used is a qwerty keyboard normally used for desktops and laptops. The keyboard will give input of number and alphabet character (ISU ID number and Password of the locker) to the Pi in order to be processed and then send the appropriate information to open the respective locker.

Hardware Specifications

Locking Unit Specification

Servo Motor

The motor being used to control the locking and un-locking of the lock is a Parallax Standard Servo (#900-00005). This servo has interface capability with PWM capable devices and can hold any position from 0 to 180 degrees. The servo can apply 38 oz-in torque at 6VDC and can operate in the range of 4-6V. The servo also has a maximum current draw of 140+/-50mA at 6VDC and draws 15mA under static conditions.

The I/O of the servo motor is connected to the ATMEGA328 via a transistor. When the ATMEGA sends a signal to move the position of the servo, the increase in voltage causes the transistor to power the servo and move it to its desired location. When the servo does not receive a signal from the ATMEGA, it stays at its current location and operates under static current conditions. The servo is independently powered from the battery source in an order to avoid current spikes.

Lock

The lock being used is a Master Lock (1714) that is commonly used in athletic locker room lockers. The lock contains a 5-pin tumble cylinder for supreme security and is key operated. The lock is spring loaded with its resting position in the locked position. The lock is controlled by the servo motor, which is connected to the lock via metal fasteners. When the servo moves, it pulls the lock into the unlocked position and then moves the lock back into the locked position once the unlocking time (TBD) has expired. The spring loading capabilities of the lock make it so the lock can be in the locked position when the locker is open, then when one wants to close the locker, the lock is compressed to fit back in the locker and then expanded once fully in the locker.

LED (160-1057-ND)
A bi-colored (red/green) LED is used to provide the user feedback of the status of the locker. A green light indicates that the locker has been unlocked. A red light indicates that the battery in the locker needs to be replaced. The LED is 5mm and operates on a 5V power source from the ATMEGA328.

MCU Mount Specification

The MCU shall be housed in a casing unit, and a ‘mount’ shall securely hold it in place onto the lockers. This mount must support a modest amount of weight, and shall allow a 60 degree inclined keyboard.

Pi Controller Specification

The raspberry pi model we are using is the model B unit, which comes with 512 MB of ram, 2 usb ports, an ethernet port, and composite video. The processing speed and the RAM available will be more-than-sufficient for our tasks, and will be capable of withstanding any additional tasks future engineers may wish to have the device perform.

Additionally, a 16 gb micro-sd card will be used to store relevant data, and a usb splitter will also be required to connect the keyboard, wireless transmitter, and the card-reader.

Atmega Microcontroller Specification

The Atmega328 is a microcontroller chip produced by Atmel. It is an 8-bit microcontroller that has 32K of flash memory, 1K of EEPROM, and 2K of internal SRAM. The Atmega328 has 28 pins. It has 14 digital I/O pins, of which 6 can be used as PWM outputs and 6 analog input pins. These I/O pins account for 20 of the pins.

Pin Number	Description	Function
1	PC6	Reset
2	PD0	Digital Pin (RX)
3	PD1	Digital Pin (TX)
4	PD2	Digital Pin
5	PD3	Digital Pin (PWM)
6	PD4	Digital Pin
7	Vcc	Positive Voltage (Power)
8	GND	Ground
9	XTAL 1	Crystal Oscillator
10	XTAL 2	Crystal Oscillator
11	PD5	Digital Pin (PWM)
12	PD6	Digital Pin (PWM)
13	PD7	Digital Pin
14	PB0	Digital Pin
15	PB1	Digital Pin (PWM)
16	PB2	Digital Pin (PWM)
17	PB3	Digital Pin (PWM)
18	PB4	Digital Pin
19	PB5	Digital Pin
20	AVcc	Positive voltage for ADC (power)
21	AREF	Reference Voltage
22	GND	Ground
23	PC0	Analog Input
24	PC1	Analog Input
25	PC2	Analog Input
26	PC3	Analog Input
27	PC4	Analog Input
28	PC5	Analog Input

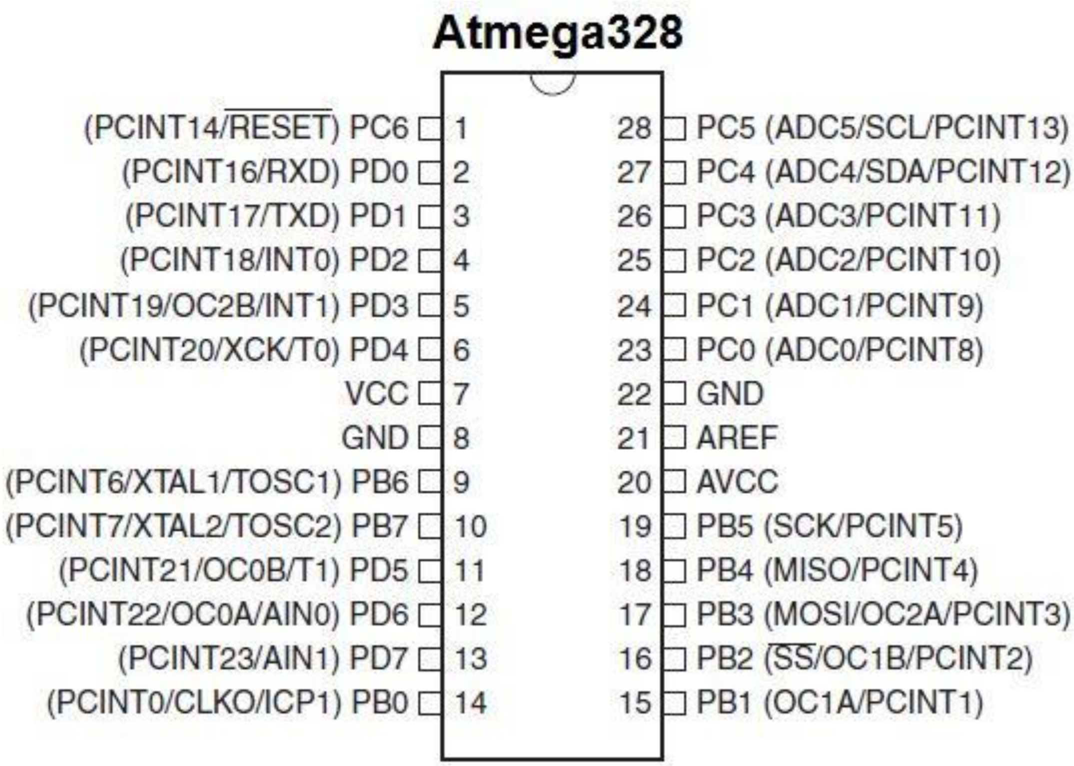

Figure 8/9, Atmega Pin Diagram

20 of the pins function as I/O ports, meaning they can function as an input to the circuit or as output, set by software. 14 of the pins are digital pins, of which 6 can function to give PWM output. 6 of the pins are for analog input/output. 2 of the pins are for the crystal oscillator, providing a clock pulse for the Atmega chip. A clock pulse is used to synchronize the communication between the microcontroller and the device(s) it is connected to. 2 of the pins, Vcc and GND, provide power to the microcontroller. The Atmega328 is a low-power chip, so it only needs between 1.8-5.5V of power to operate. The ADC has 3 pins set aside for it to function- AVCC, AREF, and GND. AVCC is the power supply, positive voltage, that for the ADC. GND is the power supply ground. AREF is the reference voltage that the ADC uses to convert an analog signal to its corresponding digital value. The last pin is the RESET pin.

Software Specifications

Atmega Microcontroller Software Specification

The ATmega328 is loaded with a bootloader that allows for the user to upload new code without the use of an external hardware programmer. The user can load the ATmega328 on an Arduino board and while using the open-source Arduino environment can easily write and upload code on to the microprocessor. The Arduino environment runs on windows, Mac OSX, and Linux. The environment is written in Java and based on processing, avr-gcc, and other open source software.

The ATmega328 contains 32 Kbytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 32 bits wide, the Flash is organized as16K x 16. For software security, the Flash Program memory space is divided into two sections, Boot Loader Section and Application Program.

Pi Controller Software Specification

The raspberry Pi shall be loaded with RASBIAN operating system to minimize development costs and to improve future usability and maintainability. The operating system features a full C-standard library, as well as additional scripting languages as necessary.

The core application shall be written in C-standard code, and the code can be either written directly onto the device using a keyboard, or remotely using Secure Shell or Secure-File-Transfer-Protocol.

Simulations and Modeling

Implementation

To simulate our system, we will ran each sub-component of our system through a series of simulations on a breadboard. First, we simulated the ATMEGA sending the command to power the servo and unlock the lock. When we run this simulation we are looking to see if the servo is moving in a smooth manner (no jittery behavior) and that the servo is moving the exact number of degrees we instructed it to. Furthermore, we will make sure that the servo is able to lock and un-lock the lock upon command. Next, we simulated the different LED colors on a breadboard by sending a signal from the ATMEGA to light each color of the LED.

We also will simulate the student ID card swiped at the magnetic strip reader and student ID number inserted through keyboard are able to open the locker via WiFi (data transfer from/to the Xbee of Pi controller to/from the Xbee of Atmega at the locker unit). Next, the LCD need to show the information of user and the battery status.

Issues/Challenges

When simulating the servo we had some minor issues with the servo having anomalous behavior. The challenge we have is what type of batteries we gonna choose and how long they will last. We also face a problem on whether or not to use a voltage regulator to power up the Xbee and Atmega328.

Testing Procedures and Specifications

Electrical Component Testing/Procedures/Specifications

All electrical components ATmega, Xbee, and servo will be tested on breadboard first and then it will be implemented into PCB layout. The voltage supply for Atmega , Xbee and servo need to be met in order to make those components working perfectly. The Xbee is tested to make sure that it can receive and transmit data in a certain range.

Hardware Components Testing/Procedures/Specifications

All hardware components will be tested to ensure that they meet our high level system requirements. First, the servo will be tested by sending a signal, from the ATMEGA, to move the servo a corresponding number of degrees. Then, one will verify that the servo did move the intended number of degrees. Next, the LED will be tested to ensure that the correct color lights are being shown at each time. A signal will be sent from the ATMEGA to power the red, green, or no color light, and one will verify the color of the light being shown to ensure that it is indeed the correct color.

Software Components Testing/Procedures/Specifications

All software for this project will be tested rigorously. In particular, specific emphasis will be given to testing security and secure information transfer. This testing will ensure that no outside force will reasonably be able to infect our system, and gain unwarranted access.

Additional Information

Electronic Data Sheets

Atmega328 Datasheet

The ATmega48PA/88PA/168PA/328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48PA/88PA/168PA/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

http://www.atmel.com/Images/doc8161.pdf

Xbee Datasheet

The XBee family of embedded RF modules provides OEMs with a common footprint shared by multiple platforms, including multipoint and ZigBee/Mesh topologies, and both 2.4 GHz and 900 MHz solutions. OEMs deploying the XBee can substitute one XBee for another, depending upon dynamic application needs, with minimal development, reduced risk and shorter time-tomarket.

https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf