# **Introduction**

This project stems from the ideas of convenience, ease of use, and reduced power consumption. With these three main ideas in mind, this team has been able to select a wide assortment of components that are able to be easily manipulated by the end user, while keeping a low power footprint for the devices themselves.

# **Design** **Constraint** **Analysis**

Component selection on this project is not the main limiting factor for the design process. However, the team’s method for selecting components has been to select a component that is necessary to meet our Project Specific Success Criteria and then follow by having the Software Engineer select a microcontroller that supports these devices.

## **Computation** Requirements

The only computation that will be necessary for our system in regards to hardware is performing an averaging calculation in order to approach the correct lux value for a given room. The light sensor’s value will be read to the main microcontroller and if the correct value is not read, the microcontroller will need to calculate a correct adjustment of the ambient and/or artificial lighting to obtain the value that is within 10% of the desired lux value, per our Project-Specific Success Criteria.

## **Interface Requirements**

Describe the general-purpose I/O functions that your device needs to perform, and identify the number of input/output pins required. Include PLDs that might be necessary to expand I/O capability (e.g., serial shift register) or provide support for some other I/O function (e.g., 7-segment display decoder). Also, identify any voltage/current constraints associated with each I/O signal (voltage swing required, current sourcing/sinking requirements, need for optical isolation, etc.).

The light detection of this project will require some sort of communication to the microcontroller itself. While the design process for this specific part is still in progress, many devices have been found to communicate via serial or analog output. All microcontrollers selected at this point contain the ability to communicate with 1 port of each SPI, USART, and I2C along with at least one entire port dedicated to Analog to Digital Conversion.

The motion detection will be a Personal Infrared detector that will simply provide a pulse if an object is detected entering a room. This pulse can be utilized to activate the lighting in a room whenever a person enters their house, for example.

On this project, there are two main types of LCD, text-only and graphic. Simple text-only displays will be used on the LINC, which will allow us to display the room being modified and the current lighting level. The graphic LCD will be used on the Remote, and contains a much larger number of pins to be used. This display will allow us to easily select the room being monitored and provide a much larger array of options that can be modified.

## **On-Chip Peripheral Requirements**

Identify the on-chip peripherals required and the number of channels each peripheral needs to support (e.g., 2 channels of 10-bit ADC, 4 channels of 8-bit PWM, 2 channels of RS-232 serial I/O, 3 input capture timer channels, 2 output compare timer channels, SPI, I2C, Ethernet, etc.).

This project will require multiple digital output devices, including USART input and output for the Bluetooth modules and I2C for the Real-Time Clock on the LINC.

For the remote portion of our project, there will be four main requirements of the microcontroller. First, the graphic LCD that will be used has 50 pins. This LCD is customizable to different configurations of input and output, such as 8-bit, 9-bit, 16-bit, and 18-bit. In order to run with as little delay as possible from within code, the Software Engineer chose to have a hardware setup that used the 16-bit configuration which is now a requirement that has been set into place for the microcontroller. The LCD has a touchscreen attached to it and the touchscreen uses a 4-pin connection that provides varying voltage levels for each of the 2-dimensional axes. This requires 4 ADC pins on the microcontroller itself and the ability to convert these to relatively concise values.

In order to keep a small form factor and ease of programming, USB support has been selected as mandatory by the team for this hardware device as well. Many devices can support USB programming after installing a 3rd-party bootloader, but it was desired by this team to use a microcontroller that supported this mode innately, without the need for unreliable code from unknown developers. USART communication will be required in order to communicate with the on-board Bluetooth module.

For the Light Interface Network Controller there will be 3 serial communication protocols in place. A Real-time clock will require an I2C method of serial communication to the main unit. The Bluetooth device that we will be using utilizes a USART communication, and finally the device will be programmed through ISP, which is accomplished through an SPI interface.

The only parallel interface will be required for the Character LCD display, which requires 6 I/O lines. This is a configurable device which can use a 4-bit or 8-bit system. The team selected the 4-bit system due the ability to use a single port off of an 8-bit microcontroller.

Finally, the Satellite portion of our project will contain similar requirements to the LINC unit, such as USART interface and SPI for ISP capability. In addition, the device will require a hardware interrupt in order to correctly process a Zero-Cross Detection of an AC signal. This style of pin is highly common on microcontrollers of today. In order to control artificial light dimming and a servo motor for blind-control, for ambient lighting, the device will require on-board timers that can be output for Pulse-Width Modulation, or PWM.

## **Off**-**Chip** **Peripheral** **Requirements**

Off of the chip, a Servo Motor will be used to control the amount of ambient light entering a room through the windows. The servo motor will be 360 degree control of the blinds and can interface with the Satellite unit to determine time of day.

The Bluetooth module is the only method of communication for all of the devices. This means that the module itself must be reliable and well-documented. The RN-42 was selected for both of these reasons and has been highly rated by consumers. The datasheet and user guides contain relevant and concise explanations for all methods of operation for the device.

In order to develop preferences, the system must be able to read a time of day. Initially, the design incorporated an Ethernet controller, the ENC28J60, which would be used to ping a time server. However, it was determined that a more efficient method of time-keeping would be using a Real-Time Clock, or RTC, with a low drift. We were able to find multiple devices that contained a swing of less than 1 minute per year.

For the AC dimming of the light bulbs, a different kind of circuit was used than the team was used to seeing. The design incorporated a photocoupler, optoisolator triac driver, and the triac itself. The photocoupler, the M11AA1 provides a zero-cross detection for the AC signal, which then allows the optoisolator triac driver, the MOC3012m, to trigger the triac, the BT136.

## **Power** **Constraints**

Describe any constraints associated with power consumption/dissipation (e.g., battery powered, A.C. powered with battery backup, A.C. powered only, heat dissipation constraints due to operating environment and/or physical size of package).

In order to keep a thin form factor, the Remote will be battery-powered with a 3.7V Lithium Ion Polymer battery. This battery will be rechargeable through a USB port, which will also be its factory programming input. Seeing as the device will be battery powered, considerations must be put in place that will allow the device to be placed in a low-power mode and powered up with ease.

The Satellite will require tapping into the household’s 120VAC power supply. The power required by most of the bulbs in consideration is very low by design, so the heat dissipation required by the PCB will not be considerable. Current consumption will not be above 1A on any of the circuit directly related to the PCB, so traces may be widened as a precaution of dealing with Alternating Current, but it is by no means required.

The real-time clock will utilize a simple Lithium coin cell battery that will run for a minimum of 9 years. This will allow the user to not be concerned with the system losing power to the house for any amount of time and needing to recalibrate the system’s clock.

## **Packaging** **Constraints**

Describe any packaging-related constraints based on the intended application (e.g., needs to be small enough to be worn on a wrist, needs to mount in a standard 1-U rack space, needs to be made of material tough enough to withstand use by Liberal Arts majors, etc.).

The Remote will be small enough to be handheld and easily stored, keeping within the form factor of a cell phone. The graphic LCD and touchscreen are both the same size as the iPhone/iPod touchscreen, sitting at 3.5” diagonally. The reset switch will be recessed within the packaging and not easily pressed; for instance, a small hole that has a button that can be pressed by a ballpoint pen.

The packaging for the satellite will need to be small enough to be mounted adjacent to a standard light switch (PCB can be mounted on backside of drywall, but light sensor will be next to light switch).

## **Cost** **Constraints**

While the majority of this project’s content is a new approach to lighting control, the Multi-Room INSTEON Lighting Control Kit was the closest that our team was able to find. The cost of this kit alone is currently $662.89 and does not include its own remote system – the system relies on a smartphone that the consumer already owns.

Since the budget of this project for Research and Development is already down to $1200, the Hardware Engineer and Software Engineer have already speculated that Project LASAR could produce a product for a mere fraction of the cost of this product already in the market.

# **Component** **Selection** **Rationale**

One of the main component selection quandaries encountered was the selection of the microcontroller and LCD for the Remote unit. Before the decision was made to use USB, the Software Engineer had already planned on incorporating an Atmega328P, in a Dual Inline Package, as he had for the Satellite. However, as the design process continued, the Systems Engineer recommended a transition that would make the Remote into a much slimmer form factor.

Two questions came of this, how would the new device be programmed in a manner that would not require male headers to be protruding from the device, and what type of LCD would allow for the minimal amount of extra buttons necessary? To answer this question, we moved away from simple text-only LCDs and into the realm of graphic LCDs. One of the most relevant sites found was crystalfontz.com, which contained a multitude of different types of graphic LCDs. A decision was made to go with a touch screen LCD that would get rid of buttons almost entirely.

The decision was made between the CFAF240320K-024T-TS, a 2.4” graphic LCD with touchscreen, and the CFAF320240F-035T-TS, a 3.5” graphic LCD with touchscreen. The 3.5” touchscreen was selected for two primary reasons; the first and foremost was that the documentation for the 2.4” screen simply appeared unprofessional with a questionable pinout for the device. Second, the 3.5” contained support on separate websites that were able to vouch for its use and also contained open source libraries to aid in the initialization process.

With a device that consisted of a 50-pin connector and 4-pin analog touch screen connector, the next phase was to examine microcontrollers that were capable of supporting this and a programming method that was slim-line in nature. Most Atmega microcontrollers have the ability to be loaded with a bootloader that contains code allowing for USB programming, but this would require an ISP just to load the bootloader on initially. Being that these Surface Mount Devices, or SMDs, could not be placed into sockets, this would completely defeat the purpose of a separate programming mechanism.

For this, the Software Engineer examined the AT90USB1287. As the name suggests, the device differed from the Atmega328 and Atmega128 by having been explicitly designed for USB compatibility with both the pins for D+ and D-, but also a bootloader shipped within the device already. For the slim-line feature, the Micro-USB interface was chosen, allowing for a very small connection to the device itself.

# **Summary**

This project, while still in major development, has many parts that are similar in nature which allowed for a relatively painless component selection process. Many devices communicate with serial communication, and the devices that do not maintain a simple connection that does not require a great deal of hardware manipulation. While the most difficult portion of this project will be writing the code for all of the devices, similar communication protocols will allow the Software Engineer to write very modular and re-usable code across all of the different microcontrollers and an adherence to Atmel’s Atmega line will also result in a great deal of cross-compatibility.