ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

---

# Clickers for ZJUI Undergraduate

---

BOWEN LI (Bowenli5@illinois.edu)
YUE QIU (yueq4@illinois.edu)
MU XIE (muxie2@illinois.edu)
QISHEN ZHOU (qishenz2@illinois.edu)

Sponsor : Timothy Lee

May 23, 2023

# Contents

# 1 Introduction

## 1.1 Purpose

Improving student engagement in class remains a challenge for instructors, not only for online classes but also for in-person classes. Engaging students through active learning pedagogues that rely on attendance is important for improving the quality of teaching. In fact, a 2010 meta-analysis found that attendance positively affects both course grades and GPA, and is the single strongest predictor of college grades. [1]. It is important to engage with them through active learning pedagogues that rely on student attendance. Thus, the clicker system can help stat the attendance rate effectively, and improve the students' concentration by encouraging them solve the in-class problems.

The I-clicker system used in UIUC for undergraduates was designed to solve these problems. It requires students to participate in class and answer questions to earn credit. Nowadays there are mainly two kinds of clickers: One is based on an embedded system and the other is based on a web application. Compared with the web applications running on laptops and cell phones, those wireless clickers have the following advantages: they are easy to carry, they can accurately match and identify user tags, they are difficult to cheat, and would not distract students. Unfortunately, most of the clickers used on campus have the following problems: inconsistency, high response delay, poor signal, and manual matching. To solve the problems above, we designed a better clicker system, ZIC (i.e. ZJUI Clicker), that achieves relatively low latency, 300 students capacity, and questionnaire-based matching. Additionally, we also replace the former power supply, that used to two 1.5v dry cells, into a 3.7v Li-ion rechargeable battery to reduce battery discarding.

## 1.2 Functionality

Our solution allow up to 300 users at the same time with relative low latency near 100ms, with the ideal working radius at 50 meters. The stand-by current of the Clicker is nearly 30mA, with the total stand-by power at nearly 1.2w. After a full charge, the Li-ion battery can continuous supply power for 60 hours in standby mode.

Our flexible hardware design allow us to update the firmware on Clicker, we can even update the hardware components with our detachable PCB design. We also made several innovative upgrades on the original operation logic, by adding a unique Clicker ID that binding with the user to simplify the registration logic, and implement software that can provide visual aid for answering progress, provide answer data analysis for Professors and remotely synchronize the answer history.

## 1.3 High-level Requirement Lists

1. The total power of portable clickers should be between 2W to 5W to effectively prolong the battery life for long-time operation.

2. The transceiver's effective working radius should be at least 50 meters and support up to 125 multichannel connections to cover most classroom structures. The base station should also have a Low Noise Amplifier module to improve the quality of signal reception and eliminate noise signals.
3. The data transmission between clicker and transceiver should be completed within 500 milliseconds to ensure timeliness of the polling results.

# 2 Design

## 2.1 Design Overview

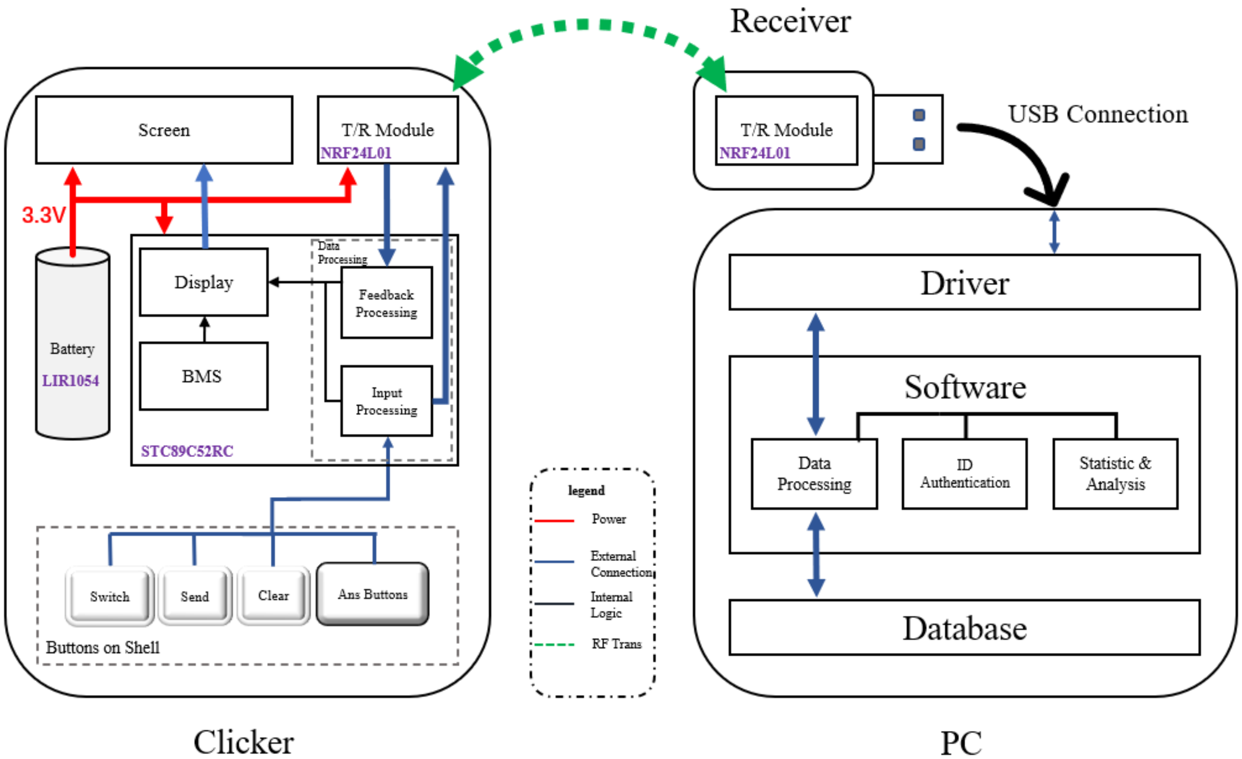


Figure 1: Module Blocks in Initial Design

### 2.1.1 Hardware overview

Two elements when we considering our design are cost and difficult level. Cost determines the probability of our product being put into use. Difficult level determines the degree level of completion of our final product. We have three versions of hardware design, each surpassing the previous one in terms of cost-effectiveness, aesthetic appeal, and overall performance.

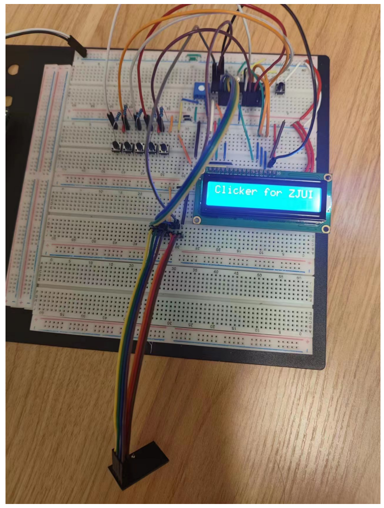


Figure 2: First Version: on Breadboard

The first version is an experimental prototype which we used to verify whether our project can be implemented essentially. It was built on a breadboard. However, before building it, we proved that all the functions were correct and tested the basic version using Arduino. The first version of the product had four primary hardware modules which were the screen display module, signal transmission module, buttons input module, and single chip microcomputer.

Following the idea of the first version, we have transferred the design on the breadboard to a PCB board. That is the second version of our product.

We have made some modifications in the second version. The first modification is the screen display. The LCD that was used in the previous version was too large to fit into our clicker, so we had to use a smaller OLED instead. We used IIC protocol to connect it to our PCB board. The second modification we made was the inclusion of a power supply module for our PCB board. This allowed us to use TYPE-C for power.

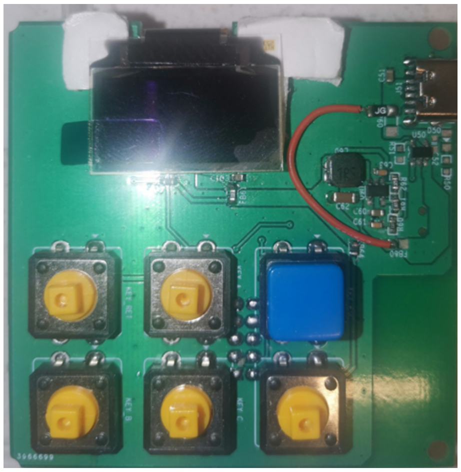

Figure 3: First PCB Board: Front Side

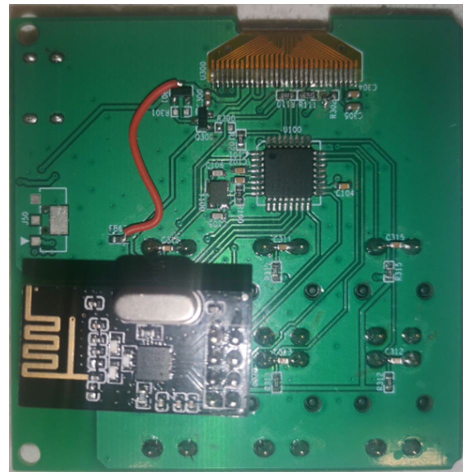

Figure 4: First PCB Board: Back Side

For the second version of our hardware design, we also have chosen ATmega328p as our MCU. ATmega328p is easier to program and burn, if we use Arduino as a development board. It decrease our labor cost on transfer the arduino IDE language to other MCU language.

But after designing, we found there is an issue. There is a big latency between pushing the buttons and screen display. Based on our analysis, it appears that the root cause of the issue lies in an incompatibility between the IIC transport protocol and the ATmega328p chip, which has resulted in the problem at hand. Analyzing information could be found on the verification section. And we replace the MCU to STM32F103 to reduce the latency. So that is the third version:

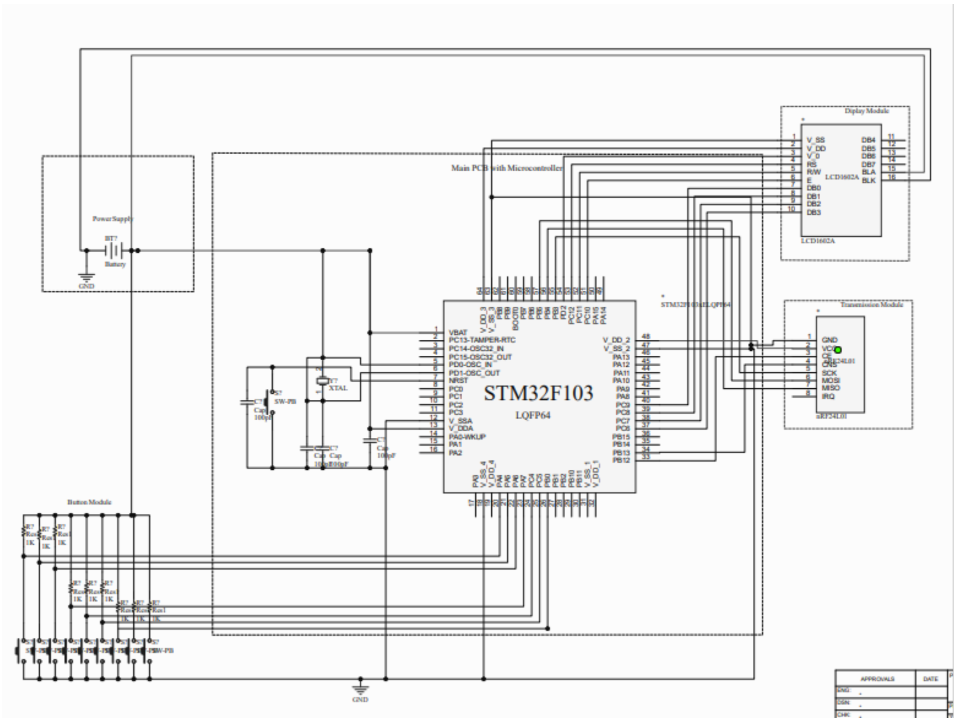


Figure 5: Schematic diagram in version 3

The third version of our product is smaller and more integrated. We have optimized the problem that occurred in the previous version. We used 16 pins of the OLED to connect with the other components but not IIC in the previous PCB and replaced the MCU ATmega328p with STM32F103. It significantly improved our final product, and now we have a microsecond delay. That is the total workline we have done so far.

Figure 6: Clicker version 3

For the future work, we will focus on resolving the signal reception problem by redesigning the antenna and the transmission code, and conduct further testing and optimization to ensure that the device functions properly across various distances and environmental conditions.

#### 2.1.2 Software Overview

The software contained integrated back and front ends, which are developed in Python. It is used to read and process the data from the receiver via USB port.
The user interface, built based on PyQt Library, is user-friendly and quick-response. All components in most interface is well-arranged in stratified layout, so the software could be adapted to zooming in and out. Users (professors) first login, choose the course, and then they can start an answer section with just a few clicks. The "History Query" function empowers professors to check, modify, and perform statistical analysis on the collected data. The software will also offer data visualization to user, which could be an intuitive attendance sector diagram or histogram of answer distribution. There is also an Intelligent Teaching Assistant System (ZITA) to provide teaching data analyses. As for the registration, professor could register their account in the backstage, which may be merged to the official website of institute in future. Last but not least, the method of binding student ID in the application is handy for professor.


Figure 7: Application Icon

### 2.2 Detailed Design

#### 2.2.1 Wireless 2.4GHZ Signal Transceiver Module

This module is divided into two parts, one is connected to the software side of the computer, the other is connected to the hardware side of the MCU. A DC power is also needed. We choose nRF24l01 chips as our signal transmission part. One chip on the hardware side should be connected to our MCU and DC power, the other chip on the software side should be connected to an Arduino interface to print the received data to the port monitor on the computer.

The signal transmission module has not changed much in our three versions of the hardware. This means that there have not been any significant changes in the hardware connections. However, we made modifications in the software. In the first two versions of the program based on the ATMega328p chip, we only sent information to the receiving end without checking if it was sent successfully. But in the third version based on the STM32F103 hardware program, we added an ACK check to the sender and receiver module and the results of the ACK check are displayed on the OLED screen.

### 2.2.2 Signal Conversion Driver

A driver is required on the USB interface. After receiving data stream from hardware, computer need a driver to Convert hardware signals into software signals. Similarly, when the software wants to send a feedback signal to the hardware, it should also use this driver to convert the software signal into the hardware signal.
**Functions of driver:**

1. Reception
   (a) Read response messages from USB port.
   (b) Filter out messages with incorrect format.
   (c) Extract ID and answer information from messages.
2. Sending
   (a) Add wrapper to feedback messages from Data Processing Unit.
   (b) Write Feedback message to USB port.

We choose module pySerial which encapsulates the access for the serial port to implement the driver. It provides back ends for Python running on Windows, OSX, Linux, BSD (possibly any POSIX compliant system) and IronPython. The module named "serial" automatically selects the appropriate back end.
The driver we design should be able to detect the port the Receiver is plugged into. To receive data from USB port, serial port should be initialized with Baud Rate = 9600, which refers to the rate of the effective data signal modulation carrier, that is, the number of times the carrier modulation state changes per unit time.
**Communication Protocol**
The communication protocol is embedded into the driver. In the primary communication protocol, we set the format of response to be ***"ID{7}+ANS{1}"***, totally 5 digits. Similarly, the format of feedback to be ***"ID{7}+Correctness{1}"***. For "Correctness" field, 1 is for correct, and 0 is for wrong. The driver should filter out messages with incorrect format by a helper function.

### 2.2.3 Application Interfaces

On the top of Login Interface, the ornament picture amalgamates the panoramas of ZJUI campus and UIUC, indicating the seamless collaboration between two University. All qualified professor will be registered in back stage, and they could login to use the application with username and password.

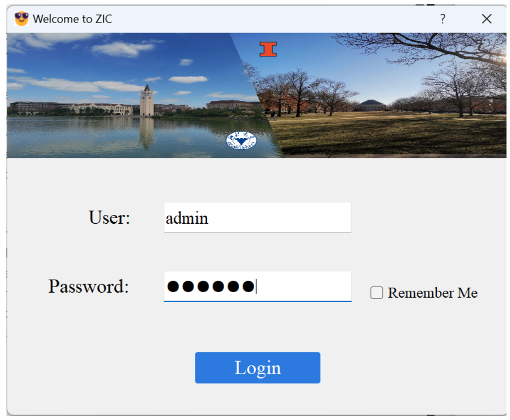


Figure 8: Login Window

After successful login, professor could choose the course from a course list. This list is tailored for each user and it could be the list of courses that the professor teaches in this semester. Course number is to indicate the index of classes within the same course. For the convenience of course selection, the course number will be updated automatically, based on the index of classes taught in the past.
During the class, once the professor decide to start a question, he/her ought to first specified the question number, a correct answer ('V' for vote, used to start voting), point and answer time. Again, in consideration of the convenience, question number, point and answer data will be set or updated automatically. One of the software innovative features is that the progress of answering will be displayed on a green bar. Professor could supervise the answering progress of student and decide whether extend the answer time. And a histogram of answer distribution will appear right after the answer section finishes.10

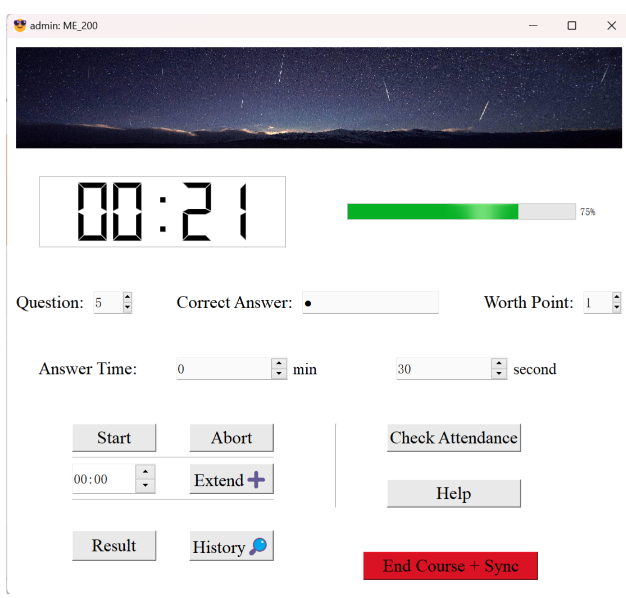


Figure 9: Answer Section

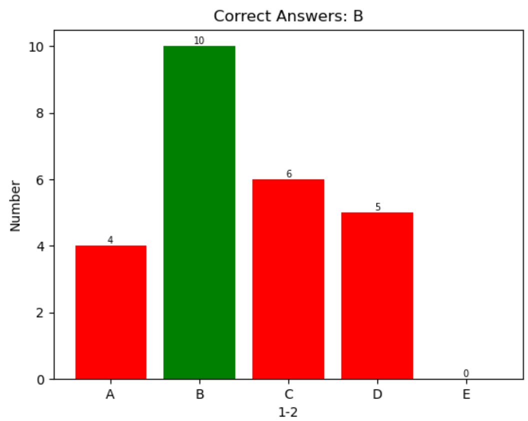


Figure 10: Student Answers

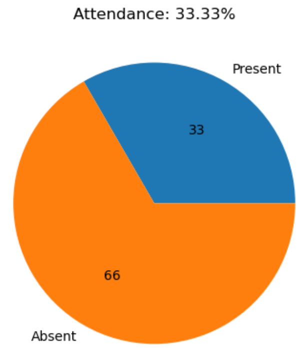


Figure 11: Attendance

#### 2.2.4 Information Enquiry and Database

The course history section displays various details for each class entry, including the class index, date, total number of students, number of students who attended the class, attendance rate, and the class topic. Professors have the ability to remove individual class records as needed. By selecting and double-clicking on the entries or clicking the 'Detail' button, professors can review the answer details of questions covered during the class.
The answer history displays various details for each entry, including the question index, correct answer, number of answers, number of correct answers, points awarded, and answer time. Professors have the ability to remove individual question records as needed. They can easily review the specifics of each student's answer by selecting and double-clicking on the entries or by clicking the 'Detail' button. Additionally, through batch processing, professors can conveniently view the details of several or all questions simultaneously. By double-clicking on a specific question, a histogram depicting the distribution of answers for that particular question will be displayed.

#### 2.2.5 Intelligent Teaching System and Data Analysis

This system will analyse all course data, and focuses on the attendance curve and courses with relatively high or low accuracy. Based on analyses, it will offer commands or strategies to professor for the sake of raising attendance or better teaching.

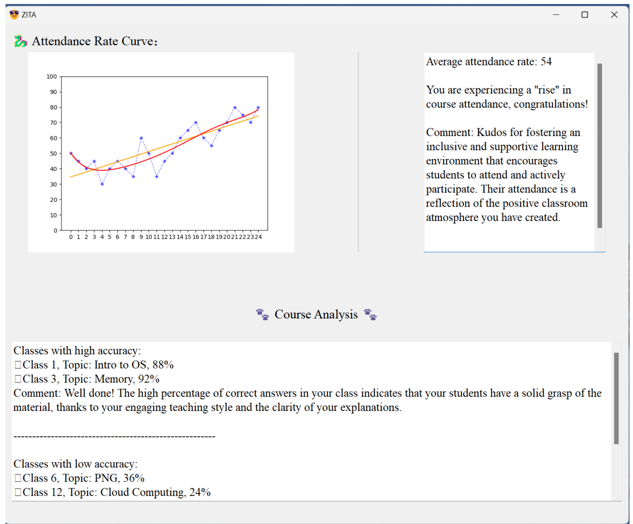


Figure 12: ZITA

### 2.2.6 Registration

Regarding registration, professors can conveniently create their accounts through the software's back end, which may eventually be merged with the official institute website. In practical use, each student is associated with a Clicker, necessitating a mapping between student IDs and Clicker IDs within the software's database. To streamline this process, professors can simply distribute a shared file to students, allowing them to fill in their mapping information. Upon collection of the shared file, professors can complete the registration with a single click, as a script automatically processes the mapping and finalizes the registration. This efficient method eliminates the need for manual entry of student information and Clicker IDs.

## 2.3 MCU Hardware Logic

### 2.3.1 MCU Hardware Logic Subsystem

We use a MCU to compile the display module, send and receive message from wireless transceiver module and receive hardware signal from keystroke. It also needs a voltage supply. In our first two versions, we use ATmega328p as the MCU. ATmega328p is a common and popular MCU at present, and its supply voltage is between 3.4V to 5.5V. This module should run all the time after power is on and send data stream to the wireless transceiver module upon receipt of a keystroke command. During power-on, MCU should always keep the screen refreshed, that is, update the contents of the screen every unit of time. On the third version, we used STM32F103 instead of ATmega328p. In addition to being more compatible with OLED, the STM32F103 also has a more accurate clock signal and a cheaper price. The hardware logic can be shown in the following figure:

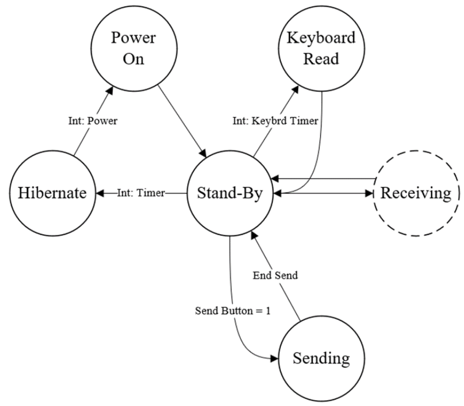


Figure 13: Progress State Diagram

When the power is turned on, the program will go into standby state and go to a while loop. It has a timer and hibernates when the timer goes to zero. Any button input will reset the timer and return to the standby state. If one presses the option button, an option like A, B, C, or D will be passed to the bar and the screen will display it. If one presses the send button, nrf24l01 module will work to do signal transmission with the PC terminal and data transmission with the Single microcomputer.

## 2.4 Component Selection

### 2.4.1 Micro-Controller

In our design, we have considered two different microcontrollers: the STM32F103 and the ATmega328p. Let's take a closer look at each microcontroller and their respective features.

The STM32F103 is a 32-bit microcontroller based on the ARM Cortex-M3 processor core. It is part of the STM32 series developed by STMicroelectronics. This microcontroller offers a range of features and capabilities, including a generous on-chip flash memory of 512KB and 96KB of RAM. It also includes various integrated peripherals such as UART, SPI, I2C, USB, CAN, timers, ADC, and more. The STM32F103 provides a significant number of General-Purpose Input/Output (GPIO) pins that can be configured for different purposes and can interface with external components. Additionally, it incorporates power-saving features that enable efficient power management, making it suitable for low-power applications. STMicroelectronics provides a comprehensive development ecosystem for the STM32 microcontrollers, including software development tools, libraries, and support for popular IDEs such as STM32CubeIDE, Keil MDK, and IAR Embedded Workbench. Due to its performance, power efficiency, and extensive peripheral support, the STM32F103 is widely used in various applications ranging from industrial automation to consumer electronics, medical devices, and robotics. [2]

On the other hand, the ATmega328p is an 8-bit microcontroller based on the AVR architecture, manufactured by Atmel. It features a 32KB flash memory for program code storage and 1KB of EEPROM for non-volatile data storage. With 23 general-purpose I/O pins and support for communication interfaces such as UART, SPI, and I2C, the ATmega328p offers versatility for a range of applications. The ATmega328p is particularly popular in the Arduino community, where it is widely used in Arduino development boards like the Arduino Uno. Its simplicity, ease of use, and extensive community support make it a favored choice among electronics enthusiasts and hobbyists. The ATmega328p provides a reasonable processing power and a rich feature set, allowing it to be utilized in a wide range of embedded systems and projects. [3]

Here is the comparison between two micro-controllers.

| Micro-controller | Power | Performance | Size | Pins | Interface | Dev-Tools |
|---|---|---|---|---|---|---|
| STM31F103 | - | + | - | + | + | + |
| ATmega328P | + | - | + | + | + | + |

After considering the I/O features, further development possibilities, and addressing the issues mentioned in the previous section, we have chosen the STM32F103 microcontroller as the preferred option for our system. The STM32F103 offers extensive I/O capabilities, including GPIO pins, an enabled SPI interface, and JTAG/SW connections for online debug mode. It benefits from a well-established ecosystem with ample resources and strong community support, making it a suitable choice for our project.

### 2.4.2 Transceiver

One of the most critical aspects of our design is the transceiver chip, as it determines the overall performance of the system. Our primary objective is to create a transmission system that is both low-power and stable. After careful evaluation, we have determined that the nRF24L01+ chip is the optimal choice for our project.

Manufactured by Nordic Semiconductor, the nRF24L01+ is a wireless transceiver module that operates at a frequency of 2.4 GHz. This frequency falls within the globally available ISM (Industrial, Scientific, and Medical) band, which is designated for a wide range of wireless applications. The module offers data rates ranging from 250 kbps to 2 Mbps, ensuring efficient and swift transmission of data. To facilitate communication with a microcontroller or other devices, the module utilizes the Serial Peripheral Interface (SPI).[4]

One noteworthy feature of the nRF24L01+ is its ability to provide 126 available channels for communication. This enables users to avoid interference from other wireless devices that may operate within the same frequency band. By having a multitude of channels to choose from, we can ensure reliable and uninterrupted data transmission. Additionally, the nRF24L01+ is designed to operate at low power, making it particularly suitable for devices powered by batteries.[4]

Another advantage of the nRF24L01+ is its support for Nordic Semiconductor's Enhanced

ShockBurst™ protocol. This protocol enhances packet handling and incorporates automatic retransmission of lost packets, significantly improving the reliability of wireless communication. This feature ensures that data packets are transmitted accurately and any lost packets are automatically retransmitted, thus minimizing errors and maintaining the integrity of the transmitted information.[4]

Furthermore, the nRF24L01+ offers an impressive communication range of up to 100 meters in open space. This range is more than sufficient for our needs, as it surpasses the size of a typical classroom environment. With such a reliable transmission range, we can be confident that our wireless communication will remain stable and uninterrupted within the designated area.[4]

Considering both power consumption and transmission range, we have carefully selected the nRF24L01+ chip as the ideal transceiver for our project. Its low-power operation, extensive communication channels, support for the Enhanced ShockBurst™ protocol, and impressive transmission range make it an excellent choice for creating a low-power and stable wireless transmission system. By incorporating the nRF24L01+ chip into our design, we can ensure the optimal performance of our overall system.

### 2.4.3 Display Module

In our design, we require a display module that offers fast response speed, high resolution, a suitable size, and low power usage. After careful consideration, we have narrowed down our options to two possibilities: an LCD screen and an Organic Light-Emitting Diode (OLED) display with the SSD1306 display controller.[5][6] Let's compare the features of these two options in the following table:

| Features | LCD | OLED |
|---|---|---|
| Response Speed | Moderate | Fast |
| Resolution | High | High |
| Size | Large | Small |
| Power Usage | Relatively higher | Low |

Based on our evaluation of the size and resolution requirements, we have ultimately decided to choose the OLED display module.

The OLED display offers several advantages over an LCD screen. First and foremost, OLED displays have a faster response speed, which means that the display updates quickly and provides a smooth user experience. This is particularly important for applications where real-time data or fast interactions are involved.

Additionally, OLED displays typically have high resolutions, ensuring crisp and detailed visuals. This is advantageous when it comes to displaying intricate graphics, text, or images that require fine details. It enhances the overall visual quality of the display, making it more appealing to users.

Another benefit of OLED displays is their versatility in terms of size. They are available in various sizes, allowing us to select a size that best fits our design requirements. Whether we need a small display for a portable device or a larger display for a more prominent interface, OLED technology offers flexibility in choosing the appropriate size.

Furthermore, OLED displays are known for their low power usage. Unlike LCD screens that require a backlight to illuminate the pixels, OLED displays emit light individually for each pixel. This means that only the pixels that need to be illuminated are powered on, resulting in lower energy consumption. This is especially crucial for our design, as we aim to create a system that is energy-efficient and optimized for low power usage.

Considering all these factors, including the size and resolution requirements, we have concluded that the OLED display module with the SSD1306 display controller is the most suitable choice for our project. Its fast response speed, high resolution, various size options, and low power usage make it an ideal fit for our design needs. By incorporating the OLED display module into our system, we can ensure an excellent visual experience while maintaining energy efficiency.

#### 2.4.4 DC Regulator

There are various methods available to implement DC regulators. However, after careful consideration of factors such as cost and size, we have made the decision to utilize the current mode monolithic buck switching regulator JW5222. This particular chip offers the capability to convert input voltages ranging from 2.5 to 6 volts to lower voltage levels, encompassing the voltage range of both the Lithium Battery and USB power sources. Furthermore, it is packaged in a compact SOT-23 form factor, which makes it suitable for integration within our design. Additionally, the JW5222 provides built-in short circuit protection and thermal protection features, ensuring the safety and reliability of our overall design.[7]

#### 2.4.5 Charging Controller

The MCP73831 is a highly integrated single-cell lithium-ion/lithium-polymer battery charge management controller. It is designed to provide complete charge management for single-cell lithium-ion or lithium-polymer rechargeable batteries. The MCP73831 offers a compact and cost-effective charging solution with minimal external components.[8]

#### 2.4.6 Battery

The BL-5C battery has been selected for our system. It is a rechargeable lithium-ion battery commonly used in mobile phones and portable electronic devices.

The BL-5C battery operates at a nominal voltage of 3.7 volts and has an end charge voltage of 4.2 volts. This voltage range aligns well with the input requirements of our power supply.

One of the primary reasons for choosing the BL-5C battery is its rectangular shape with rounded corners. This slim, rectangular design is an ideal fit for our system, providing a compact form factor. In comparison to standard 18650 lithium batteries, the BL-5C battery occupies less space.

Additionally, the BL-5C battery offers a high capacity of 1800 mAh. This substantial volume enables our design to maintain power for extended periods, enhancing the overall runtime and usability of our system.

#### 2.4.7 Basic Component

Ultimately, by selecting resistors and capacitors in the 0603 and 0402 packages with a 1% tolerance, we strike a balance between assembly complexity and design size while upholding the stability of our system. This approach enables efficient assembly processes, compact design implementation, and reliable performance in our electronic system.

## 2.5 Hardware Schematic Design

#### 2.5.1 DC Regulator

The crucial component of the power supply subsystem is the DC regulator, which serves the purpose of reducing the input voltage and maintaining a stable DC voltage output.

In the provided diagram (Figure 19), the integrated circuit labeled U60 corresponds to the JW5222 switching regulator. Based on the JW5222 datasheet [7], the output voltage of this component can be determined using the following equation:

$$V_{FB} = V_{OUT} \frac{R_3}{R_2 + R_3} \tag{1}$$

To achieve the desired output voltage, resistor R61 and R62 are selected with values of $12K\Omega$ and $30K\Omega$, respectively, resulting in the calculated output voltage mentioned.

$$V_{cc} = V_{BAT} \times 0.714 \in [2.6, 3] \text{ V} \tag{2}$$

To ensure a stable DC voltage for other parts of the circuit, several high-pass filters are implemented. These filters, denoted by C60, C61, C62, and C63, are designed to filter out high-frequency noise from the power supply, thereby maintaining a clean and stable DC voltage.

Additionally, several magnetic beads, represented by FB61, FB62, FB63, and FB64, are employed to separate different subsystems. This separation helps to reduce any potential interference or influence between the subsystems, promoting better overall system performance and reliability.

### 2.5.2 Charging Circuit & Battery

In the provided diagram (Figure 20), the integrated circuit U50 represents the MCP73831 charging controller, which is specifically designed for Lithium batteries. The socket J50 is designed to accommodate the BL5-C battery, a common battery used in cellphones. The connection of U50 follows the recommendations provided in the MCP73831 datasheet. Additionally, an LED labeled D50 is connected to indicate the charging state of the battery, providing visual feedback to the user.

The charging current is determined by following equation[8]:

$$I_{REG} = \frac{1000V}{R_{PROG}} \tag{3}$$

According to fig.n, $R_{PROG} = 2k\Omega$, the charging current would be $500mA$. However, in the assembly stage, we change the resistor to $4.2K\Omega$ to reduce the charging current.

The junction J51 corresponds to a USB Type-C socket, which serves as the power input for the charging circuit. Preceding the charging circuit, a PTC (Positive Temperature Coefficient) device is utilized as a safety measure to prevent potential short circuits. This PTC device acts as a fuse, automatically disconnecting the circuit when the current exceeds 1A, thus protecting the components from high current and potential damage.

### 2.5.3 Micro-Controller

In the provided diagram (Figure 21), the integrated circuit U100 represents the STM32F103 microcontroller, known for its power and popularity in the industry. To ensure accurate timing and synchronization, an 8 MHz external crystal oscillator is connected to pins 12 and 13 of the chip, serving as the main clock source for the microcontroller.

The display module is fully connected to the GPIO pin group D, simplifying both the PCB layout design and software implementation. An SPI interface is enabled to establish a connection with the network module, facilitating data exchange and communication.

The keyboard is connected to the microcontroller using 8 GPIO pins, allowing for input from the user. Additionally, all voltage supply connections incorporate high-pass filters, which effectively filter out high-frequency noise and ensure a clean and stable DC power supply to the circuit.

It is worth noting that pins 89, 90, 72, 76, and 77 are connected to a socket for the JTAG/SW (Joint Test Action Group/Serial Wire) connection. This connection enables online debug mode, providing the capability to debug and analyze the system during development and testing phases.

### 2.5.4 Transceiver Circuits

However, it is not feasible to utilize nRF24L01 transceiver modules in our final design due to their large size, which would result in an overall increase in the size of the system. As a result, we have opted to integrate the nRF24L01 module circuits directly onto our main PCB board.

Considering that there is no requirement to modify the design of the circuit from the module, we have implemented the nRF24L01 chip by referencing its recommended design, as mentioned in the nRF24L01+ datasheet[4]. For detailed visual representations, please refer to the schematic diagram and the PCB design screenshots provided in the Appendix.

In the provided diagram (Figure 22), the integrated circuit labeled U311 represents the nRF24L01+ transceiver module, which is connected to the microcontroller unit (MCU) through the SPI interface. Additionally, a pin used for external interruption is also connected to the MCU to facilitate communication.

The capacitors labeled C313, C314, C315, C316, and C317, with respective values of 1nF, 1nF, 1nF, 10nF, and 33nF, serve as power supply noise filters. These capacitors are responsible for ensuring a clean DC voltage supply for the transceiver chip, thereby enhancing its performance and reliability.

Regarding the antenna, no modifications have been made to the original design. As the team lacks expertise in antenna design, it was decided to maintain the original configuration to avoid any potential negative impact on the overall functionality of the system.

### 2.5.5 Display Circuit

The display module is implemented into our circuit with recommend design, which is shown in the appendix.

In the provided diagram (Figure 23), the display module is connected to the microcontroller unit (MCU) using an 8-pin parallel interface. The primary advantage of utilizing a parallel interface is its faster transmission speed, enabling the system to respond promptly after a key press or input.

On the left side of Figure n, there is a power switch depicted. This power switch is responsible for controlling the power supply to the display module. To effectively reduce power consumption when the display is not in use, a switch is implemented to turn off the screen or put it into a low-power mode. This feature helps optimize energy usage and prolong the overall battery life or power efficiency of the system. The voltage of $V_{display}$ will have following relationship with DISPLAY_ON.

$$V_{diplay} = \begin{cases} V_{BAT} & \text{if } DISPLAY\_ON = HIGH, \\ 0 & \text{if } DISPLAY\_ON = LOW. \end{cases} \quad (4)$$

#### 2.5.6 Keyboard Circuit

There're two different ways to implement the keyboard circuit. In the first method, the button is connected directly to the MCU pins, which is the simplest way to implement a keyboard. The circuit diagram is shown below. When button pressed, the voltage on the button pin is set to high.

The second way to implement a keyboard matrix like design. A keyboard matrix is a grid-like arrangement of keys on a keyboard that allows multiple keys to be pressed simultaneously and detected by the keyboard controller. It is a common design used in most modern computer keyboards and helps reduce the number of electrical connections required for each key.

In a keyboard matrix, the keys are organized in rows and columns, forming a matrix. Each key corresponds to a unique intersection of a row and a column. When a key is pressed, it connects the corresponding row and column, completing an electrical circuit.

The micro-controller continuously scans the rows and columns of the matrix to detect any closed circuits, indicating pressed keys. By using a combination of row and column scanning, it can determine which keys are being pressed and stored in memory for future use. See section for more details about the keyboard scan software design.

In the provided diagram (Figure 24), all the buttons are connected to two different pins of the microcontroller unit (MCU), forming a keyboard matrix. The keyboard matrix design allows for efficient usage of the available pins on the MCU while enabling multiple button inputs.

However, in the current design, only six buttons are utilized, which means the full potential of the keyboard matrix is not being maximized. Nonetheless, it is worth noting that the design allows for future extension and expansion of the keyboard layout to a 4x4 configuration. This means that additional buttons can be added, providing more input options and enhancing the functionality of the system in the future.

### 2.6 PCB Layout Design

#### 2.6.1 PCB Layout Overview

In our system, the PCB design consists of three main parts: the core board, the keyboard board, and the test bench. The core board and keyboard board are connected together with sockets to form our product. We implement this design mainly to reduce the overall size of the PCB board and make the whole system compact. And the test bench is used for online debug. By dividing the design into these three components, the core board, keyboard board, and test bench, we are able to efficiently manage the placement and routing of various components and subsystems. This modular approach simplifies assembly, maintenance, and troubleshooting processes, allowing for flexibility and scalability in the design.

All the PCB Layout are included in Appendix A (Figure 25 - 30). Each part will be explained in detail below.

### 2.6.2 Core Board

The core board is where all the processing units are located. It houses the STM32 microcontroller (MCU), the nRF24 transceiver, and the OLED screen. In the front side of the core board, the MCU is positioned in the middle. The transceiver circuitry, designed according to the datasheet, is located in the top left corner. The bottom part of the board is dedicated to the connections for the OLED screen. Additionally, the power supply circuitry is situated to the left of the MCU. On the back side of the core board, there are two sockets located at the bottom left and right corners. These sockets are used to connect the core board with the keyboard board.

One of the challenges encountered during the design of the core board was routing the wires from the MCU to each subsystem. To address this challenge, adjustments were made to the schematic design, placing the interfaces as close to each other as possible. This proximity simplifies the routing process. For example, the GPIO D group pins and the display module are positioned adjacent to each other to streamline the routing.

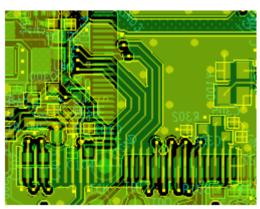

Figure 14: Parallel Interface to Display

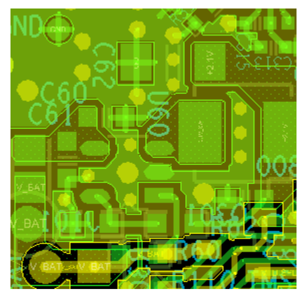


Figure 15: Power Supply Wire

### 2.6.3 Keyboard Board

The keyboard board is designed to be replaceable and serves specific functions. It incorporates a charging circuit, USB sockets, six key buttons, and a battery socket. Notably, the board features large areas of copper, as depicted in the figure. These copper areas are designed to handle large currents in the power supply wires. The width of these copper areas is specifically designed to accommodate a current of 1A, ensuring efficient power distribution.

### 2.6.4 Test Bench Board

Another noteworthy feature of our design is the inclusion of a test bench for JTAG/SW debug mode of the MCU. On the core board, several touch points are placed for connection purposes. The test bench, on the other hand, is equipped with several pins with springs to establish a connection between the core board and the JTAG socket. This setup facilitates convenient debugging and testing of the MCU.

# 3 Cost and Schedule

## 3.1 Cost Analysis

### 3.1.1 Labor Cost

Our hourly rate is $10 per hour for each member and 20 hours per week.
We need 12 weeks to finish our project. The Labor cost is approximately:

$$4 * 10 * 20 * 12 * 2.5 = 24000\$$$

### 3.1.2 Cost per Product

| Part | Unit Price |
|---|---|
| 3D Printing Consumables | |
| Nrf24l01 chips | 5 CNY |
| STM32F103ZE | 5 CNY |
| OLED | 6 CNY |
| PCB print | |
| BATTERY | 10 CNY |
| OTHER RESISTENCE AND CAPACITANCE | 3 CNY |
| Total | less than 30CNY |

Including the cost of 3D printing and PCB fabrication, the total cost should be less than 50 CNY.

## 3.2 Schedule

see Appendix B

# 4 Verification

Verification is a critical part of the development process as it helps to ensure that the designed system performs optimally and meets the requirements set for the system. The testing process was broken down into three main areas: combination circuit testing, one-to-one signal transmission and reception testing, and multiple-to-one signal transmission and reception testing.

## 4.1 Verification on hardware circuit

The combination circuit testing involved ensuring that all the components of the system worked collectively. The voltage levels, signal flow, current flow and data transmission between all the components involved in the system was tested, and we ensured that no two components interfered with one another.

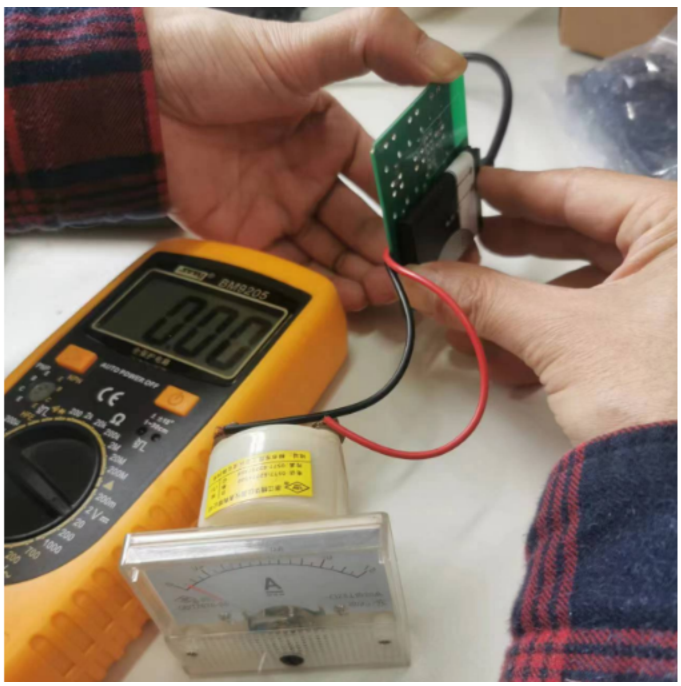

Figure 16: Verification on the current of each component

## 4.2 Verification on one-to-one signal transmission

The one-to-one signal transmission and reception testing focused on ensuring the correct reception of signals between two devices. In this test, a sender and a receiver were used to assess the quality of signal transmissions. Throughout the test, the sent data was verified against the received data to check for any loss or errors in the transfer.

```
23:40:56.376 -> □      -107
23:40:56.884 -> □      -107
23:40:57.290 ->
23:40:57.800 -> □      -107
23:40:58.208 -> □      -107
23:40:58.820 -> □      -107
23:40:59.431 -> □      -107
23:41:00.044 -> □      -107
```

Figure 17: incorrect form

```
04:43:35.338 -> □   2015381
04:48:09.528 -> B   2015381
04:48:10.560 -> D   2015381
04:48:11.448 -> B   2015381
04:48:12.572 -> A   2015381
```

Figure 18: correct form

## 4.3 Verification on multiple-to-one signal transmission

Regarding multiple-to-one signal transmission and reception, we verified the delay that may occur due to collisions through model testing. This ensures that there would not be significant collisions when handling multiple-to-one signal transmissions and receptions within 200 to 250 devices at the same time.

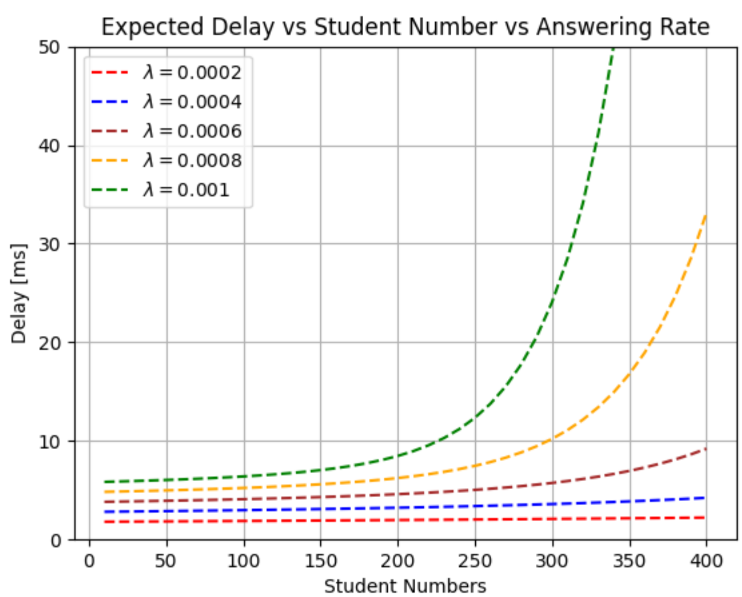


Figure 19: The Relationship Between the Number of Clickers and Delay

# 5 Conclusion

## 5.1 Accomplishments

We successfully met nearly all the high-level requirements from our design document, except the operation radio radius. Our Clicker is fitting in small-sized hands that can be operate by one hand, and the expected battery life at full charge is 50h. The wireless communication modules can handle simultaneous data upload of a large number of users, and provide feedback to the transmitter. The designed software on terminal device is user friendly, can successfully receive the answers and provide data analysis, history review and synchronize among different computers.

## 5.2 Uncertainties

Our login system is protected and encrypted by the MD5 algorithms. There are two risks: The MD5 algorithm is proved not to be entirely reliable, and gradually being replaced by other algorithms. The other risks is that all our authentication information is stored in the database, including the MD5 results. However, the access token to our database is stored locally. It is possible that if the access token is leaked, our data can not guarantee safety. Our wireless operation radius is currently not enough, which is expected to 120m indoors. The antenna we design have poor performance when they are not aligned, as a result, we should re-design our antenna, and add Power Amplifiers if necessary.

## 5.3 Future Work

We only test our system with simulations, seven clickers and one receive terminals due to the limitation of cost and labor. Our shells are 3D printed and there are some problems with structural strength. In the future, we first could extended our system to running with more clickers to meet the real scenario and verity the real performance with possible wireless signal interference. We also have the following plans to enhance our clickers in the future:

1. Adding a battery-life indicator to the user interface, that the students could know when to recharge their own clickers.
2. Adding remote distributed database instead of self-maintained database to enhance the security and availability.
3. Adding a website to allow students review and download their own grades, update their own information and communicate with the instructors.

## 5.4 Ethical Considerations

We promise to follow the IEEE code of ethics as follow:

During the development and use of our project, we must commit ourselves to the highest ethical standards and avoid any breaches. One important consideration is that our devices use 2.4GHz Wi-Fi as a data transmission medium, which may interfere with the normal operation of some vital devices, particularly medical devices. This violates the first principle of the IEEE Code of Ethics, which is to hold paramount the safety, health, and welfare of the public [9]. To address this, we must propose limits on the wireless frequency range of equipment used and specify the scenarios in which they can be used, as well as use warnings to relevant groups.

We should also take responsibility for any problems that arise from the use of our products. According to the sixth principle of the IEEE Code of Ethics, we must maintain and improve our technical competence and only undertake technological tasks for others if we are qualified by training or experience or after full disclosure of pertinent limitations [9]. Therefore, during the development process, we should ensure that our designs and codes are easy to maintain and upgrade for additional functions, user-friendly, and reliable since they will be used by various students for a long time.

More importantly, we should keep the users' privacy information safety. According to the Ethics 1.1 [9], we should protect others' privacy. Thus, we clearly declare that our i-clicker system would never collect other personal information from the users. Our system would only collect users' name for check-in and answering statistic, meanwhile, we would never collect the location information and the contact information. The users' name would be only collected and managed by the teachers and professors, and the central data management system and the communication among the devices would be encrypted to prevent unauthorized information leak.

# References

[1] M. Crede, S. G. Roch, and U. M. Kieszczynka, “Class attendance in college a meta-analytic review of the relationship of class attendance with grades and student characteristics,” *Review of Educational Research*, vol. 80, no. 2, pp. 272–295, 2010.

[2] *High-density performance line arm®-based 32-bit mcu with 256 to 512kb flash, usb, can, 11 timers, 3 adcs, 13 communication interfaces*, STM32F103xC Datasheet, Rev. 13, STMicroelectronics, Jul. 2018.

[3] *Megaavr® data sheet*, ATmega328P Datasheet, Rev. B, Microchip Technology, Sep. 2020.

[4] *Single chip 2.4ghz transceiver product specification*, nRF24L01+, Rev. 1.0, NORDIC Semiconductor, Sep. 2008.

[5] *Semiconductor technical data*, SSD1306, Rev. 1.0, SOLOMON SYSTECH, Sep. 2007.

[6] *Product specification*, OEL Display Module, Rev. B, GoldenMorning Electronic, May 2013.

[7] *2.5a, 6v, 1mhz synchronous step-down converter*, JW5222 Datasheet, Rev. 0.94, Joulwatt, 2016.

[8] *Miniature single-cell, fully integrated li-ion, li-polymer charge management controllers*, MCP73831, Rev. H, Microchip Technology, Jun. 2020.

[9] IEEE. “”IEEE Code of Ethics”.” (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 02/08/2020).

# Appendix A Schematic Diagram & PCB Layout

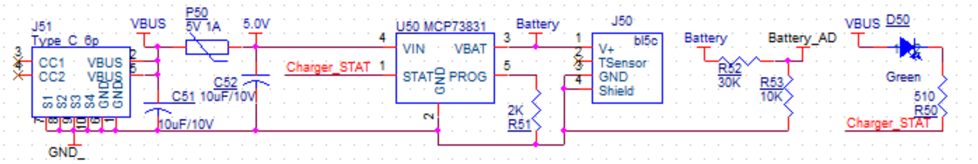


Figure 20: Charging Circuit Schematic Diagram

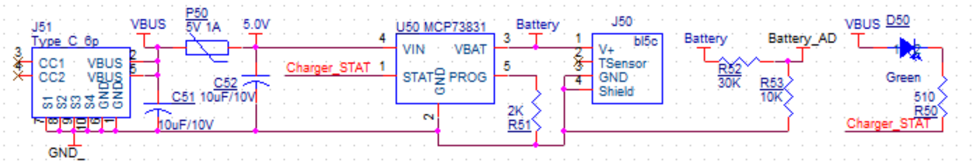


Figure 21: Charging Circuit Schematic Diagram

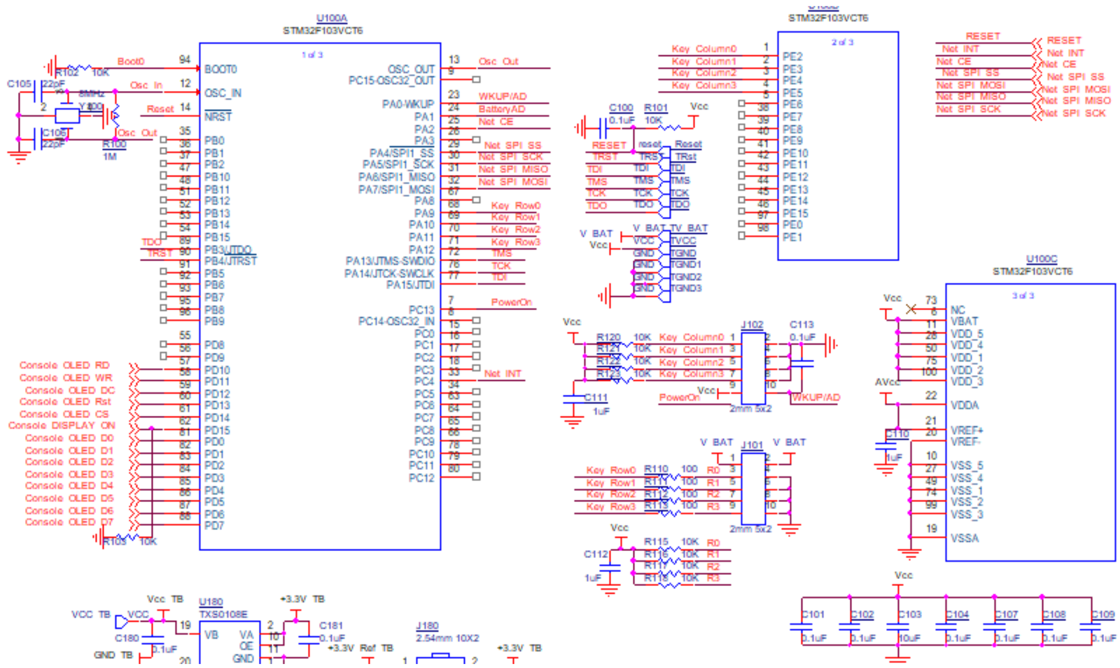

Figure 22: Micro-Controller Schematic Diagram

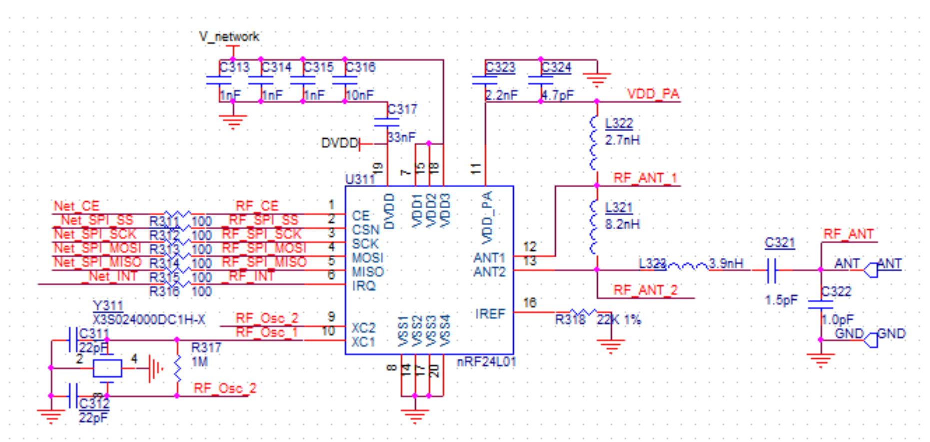

Figure 23: Transceiver Schematic Diagram

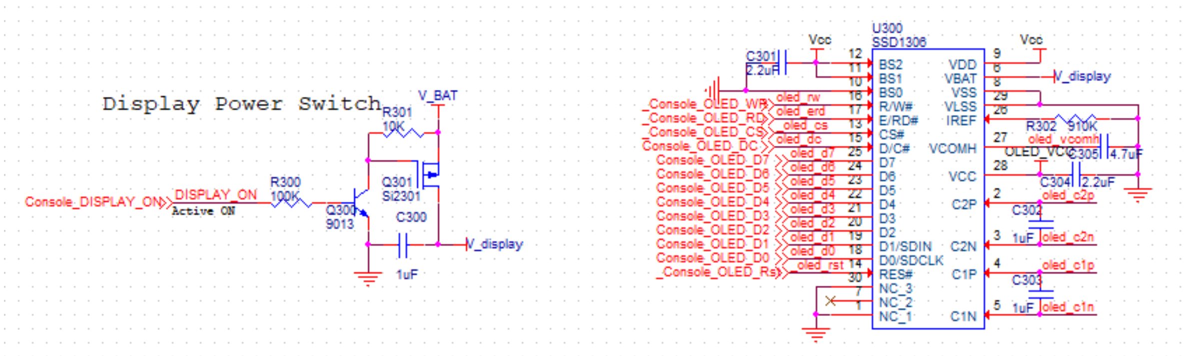


Figure 24: Display Module Schematic Diagram

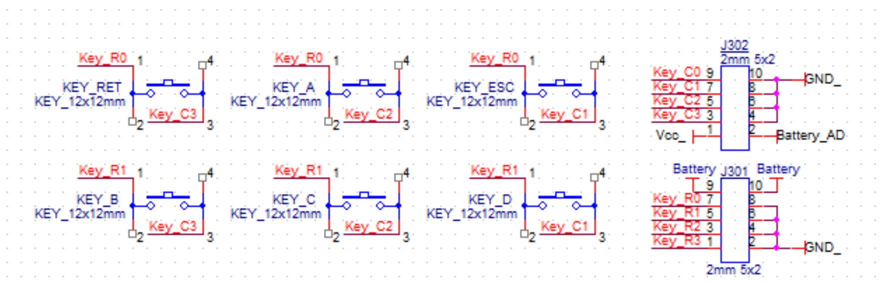


Figure 25: Keyboard Schematic Diagram

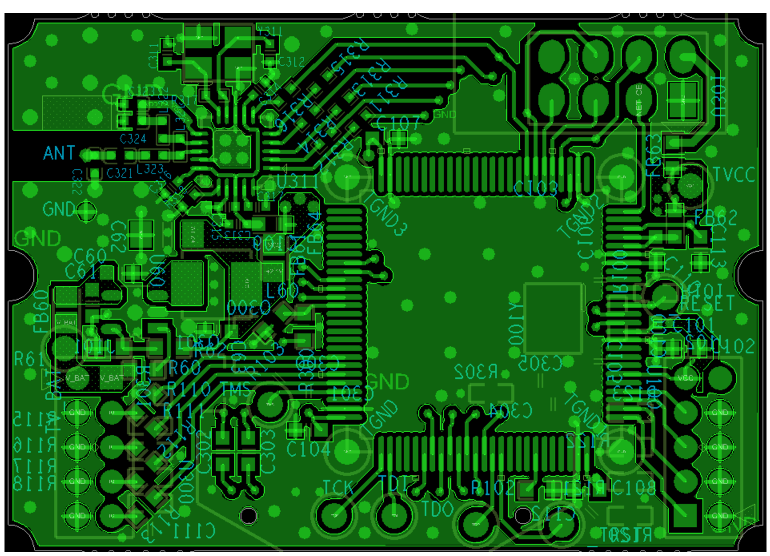


Figure 26: Core Board (Front)

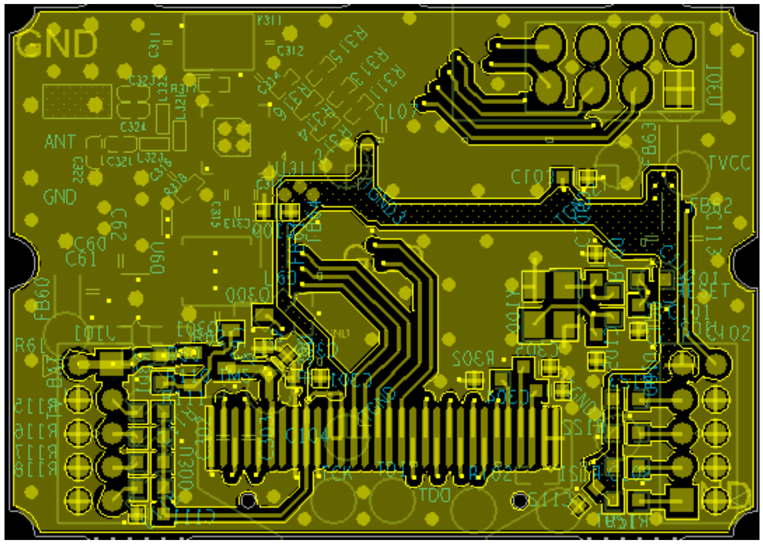

Figure 27: Core Board (Back)

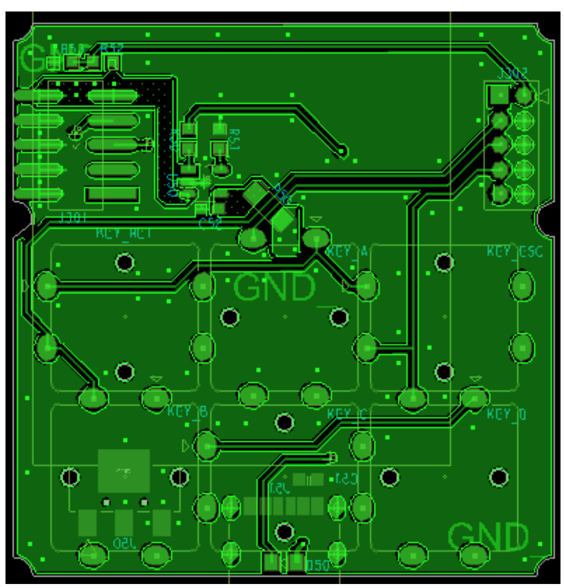

Figure 28: Keyboard Board (Front)

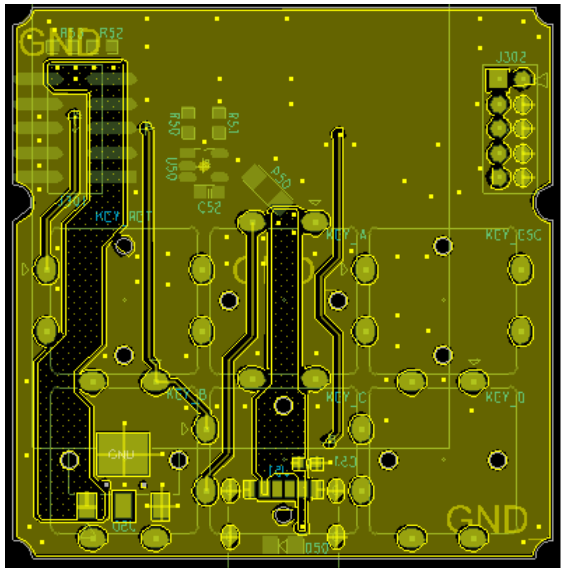


Figure 29: Keyboard Board (Back)

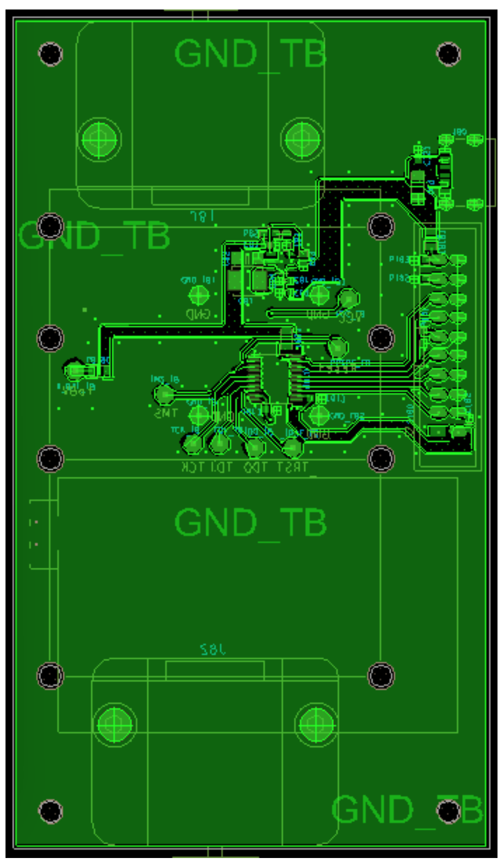


Figure 30: Test Bench Board (Front)

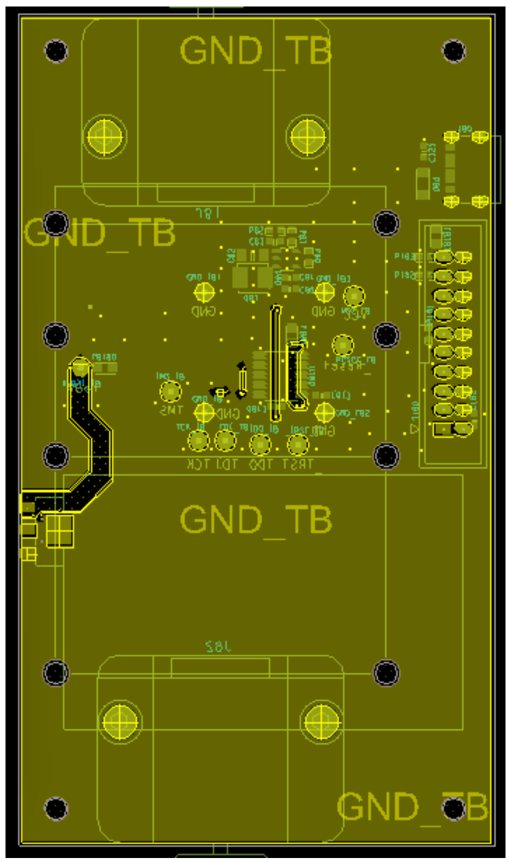


Figure 31: Test Bench Board (Back)

# Appendix B Schedule Table

| | Bowen Li | Yue Qiu | Qishen Zhou | Mu Xie |
|---|---|---|---|---|
| 2.20 - 2.26 | Learn the knowledge on Arduino and SCM. Buy the related hardware components. | Learn how to build a communication protocol. Buy wireless transmission chips and Start to write proposal. | Review on how to build a database. Start to write the code on software side. | Learn the detailed knowledge on PCB design and start to write proposal and design document. |
| 2.27 - 3.5 | Design switch logic circuit and liquid crystal display. Connect it to arduino. | Realize wireless signal between computer and nrf24l01 module. | Write the communication protocol with C or C++. | Connect nrf24l01 module to Arduino. Finish relative program code on hardware. |
| 3.6 - 3.12 | Finish the whole code on hardware with Mu Xie. Finish proposal with others. | Write basic data processing program on software part and realize data exchange with database. | Build up a database with MySQL locally. Write proposal. | Write the hardware code in Arduino with C++. Ensure signal transmission on all the components. |
| 3.13 - 3.19 | Learn how to transfer the language on arduino to SCM. Write design document. | Finish design document with others. Resolve multiple channel conflicts and limitation on many to one signal transmission issues. | Finish design document. Debug for the whole program code on hardware and software until now. | Consult different types of SCM workbooks and choose the most suitable SCM by burning program to it one by one. |

| | | | | |
|---|---|---|---|---|
| 3.20 - 3.26 | Build our I-clicker prototype version 1.0 with Arduino and purchase typical SCM for choice. | Realize sending a signal on the computer side and receive it on the arduino terminal. | Help with Yue Qiu in this week. Realize sending a signal on arduino terminal and receive it on the computer side. | Test. Ensure all the signal transmission on hardware side is as expected and same with version 1.0. |
| 3.27 - 4.2 | Burn the programs on Arduino to SCM and transfer all the connection to breadboard. This is our prototype version 2.0. | Ensure all the signal transmission in prototype version 2.0 is same with previous version. | Finish data process with Yue Qiu. Realize machine code identify detection. | Integrate hardware part and weld. Work together with Bowen Li. |
| 4.3 - 4.9 | Continue with prototype version 2.0. Deal with the code if language in Arduino and SCM have several difference. | Realize signal transmission many-to-one and one-to-many. Add machine code for each nrf24l01 chip. | Complete a mature data process program. Make and refine the front end with Yue Qiu | Refine the hardware design on prototype version 2.0 for the next PCB design. |
| 4.10 - 4.16 | Replace power supply with button battery. Test the prototype with software part. | Structure computer side applications to evoke the whole system with Qishen Zhou. Structure front end. | Research on how to realize 256-to-one wireless signal transmission. | Design the shell of the I-clicker with software modeling tools. |
| 4.17 - 4.23 | Simplify the prototype version 2.0. Create a PCB design sample and print it. | Modify communication channels and protocols with Qishen Zhou. Try to induce interference. | Modify the communication chanel and frequency band.Try to connect host to as many i-clickers as possible. | Debug for the final I-clickers prototype. Refine the Wireless receive part on USB port on computer. |

| | | | | |
|---|---|---|---|---|
| 4.24 - 4.30 | Finish Version 3.0. Test all the i-clicker system. Handle signal problems in different scenarios. | Encrypt all the signal transmission. | Decryption the data on database. Try to give different feedback to different results. | Connect integrated prototype with PCB board and the plastic Shell with 3D print. Weld. That is final version. |
| 5.1 - 5.7 | Prepare for the final demonstration and report. | Prepare for the final demonstration and report. | prepare for the final demonstration and report. | prepare for the final demonstration and report. |
| 5.8 - 5.14 | Report and demo | Report and demo | Report and demo | Report and demo |

# Appendix C RV Table

| Requirement | Verification |
|---|---|
| 90% of responses from Clickers should be received correctly | 1. Test the driver with multiple receivers send concurrently.<br>2. Check whether the database has collected more than 90% of the responses and all response are in the correct format. |
| Almost all of feedback to Clickers should be sent correctly with low latency | 1. After sending a response, clickers will wait for corresponding feedback in a short period.<br>2. Check whether almost all Clickers could get feedback expected within a shord period after ”Answer Section”. |
| Attendance should be recorded accurately | 1. Start a short ”Course Section” with several ”Answer Sections”, and use a number of specific Clickers to answer one or more questions.<br>2. Check if all students’ attendance is recorded if and only if they once answer a question and their responses are also received by driver successfully. And inspect attendance information in student list is matched correctly. |
| All scores and options should match the responses from Clickers | After an ”Answer Section”, compare the message delivered by driver with final result to inspect whether the options are the same as the one in message as well as whether only the correct answers score. This process could be done by a subroutine. |
| All enquiry on historical data should be correctly and completely | Maintain a log of statistic data, compare the log with the historical data provided by Data Processing Unit after a spectrum of periods. This process could be done by a subroutine. |

| | |
|---|---|
| All feedback should be generated and send correctly within low latency | Examine the feedback messages delivered to the driver after "Answer Section". Each feedback ought to be matched with student's ID and answer. |
| All ID Binding should be Correct | 1. After a series of answering, check whether the score/status on Data processing Unit is matched to testers' operations.<br>2. Inspect whether the identity binding is one-to-one mapping. |
| Efficient to Operate | The method of registration should be handy to operate. |
| Stable database interface | Any error of database operations should be detected and fixed by program immediately during Course Sections. |
| Permanent and correct record | All entries in database should be stored correctly after a long-term. |