DIGITAL BOOK -2

What is our GOAL for this CLASS?

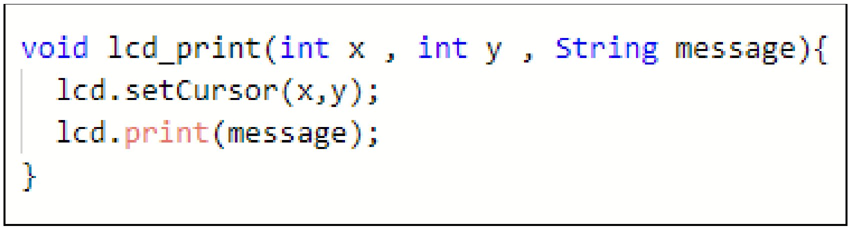
In this class, we understood how to generate customized character arrays for LCD displays. We created a welcome animation for the reading device and displayed the stories page-wise on the LCD display

What did we ACHIEVE in the class TODAY?

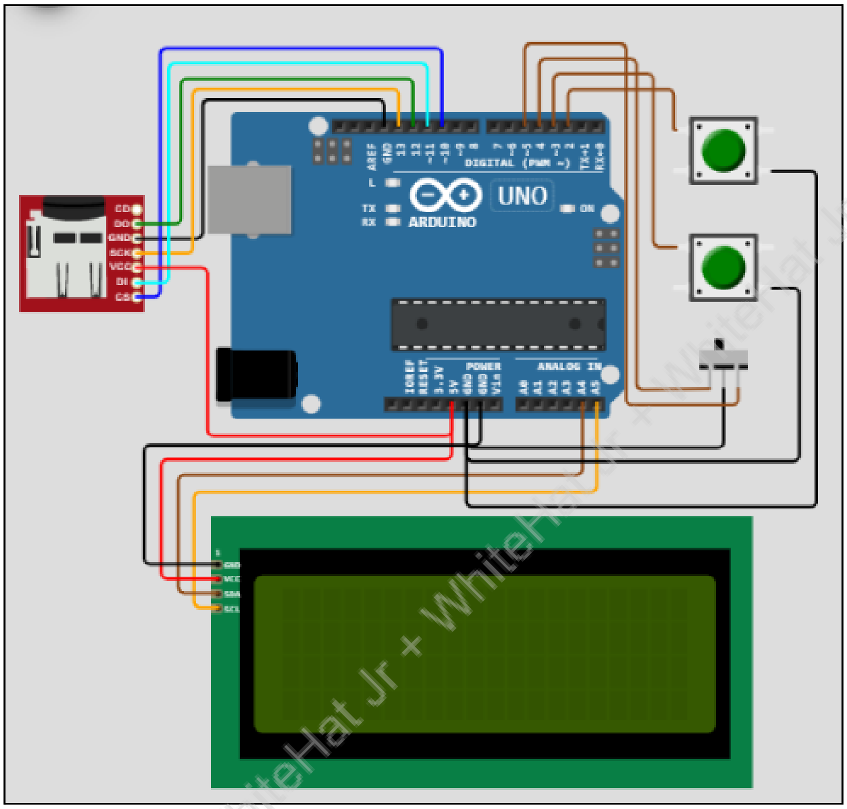
- Displayed an animation on the LCD screen.
- Displayed the contents of the book on an LCD screen.
- Implemented next page and previous page buttons.

How did we DO the activities?

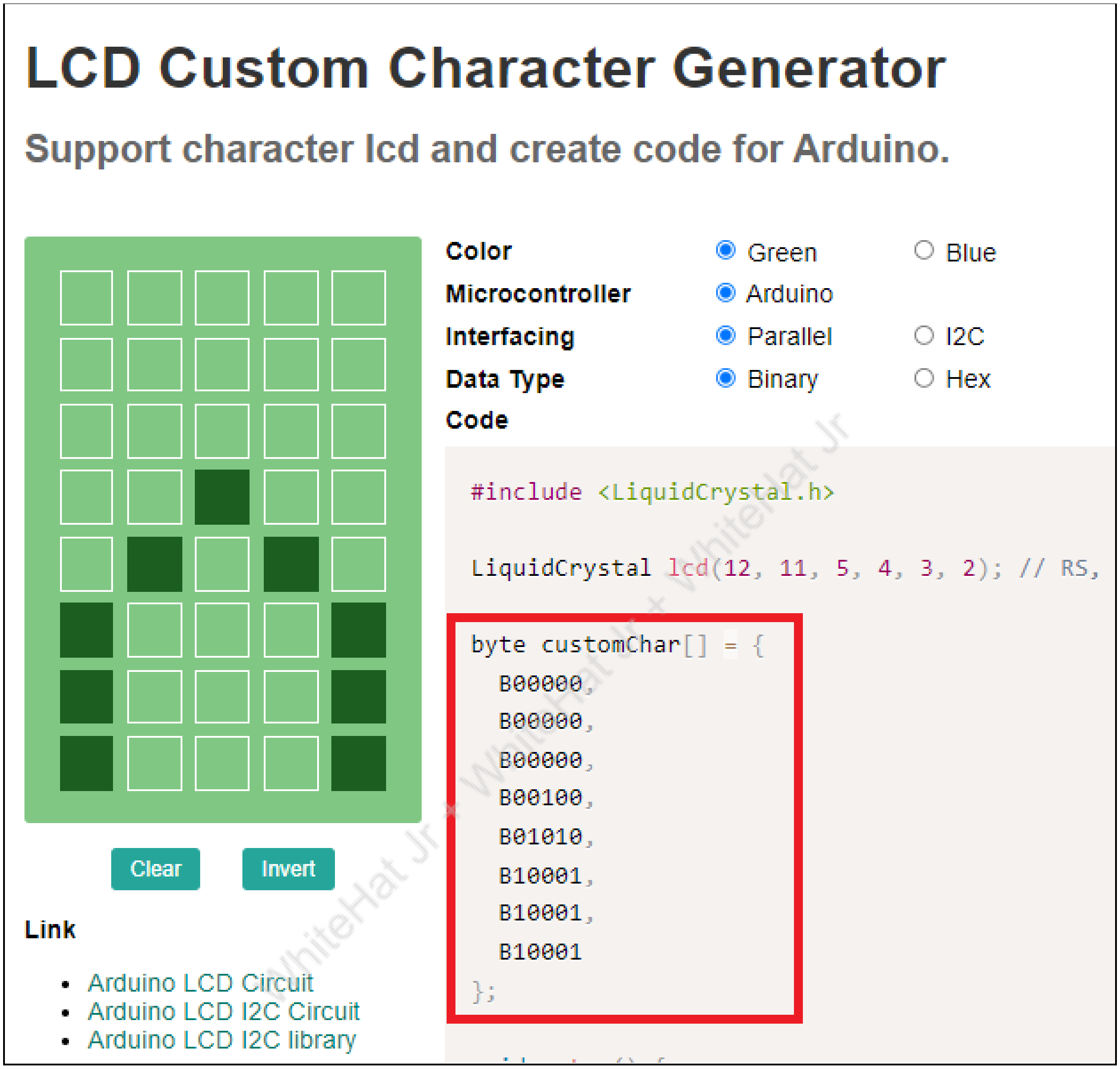
1. **LCD display**
   An LCD display is connected to the Arduino board.
   - Print the message "welcome to digibook" by defining the method **lcd_print()**.
   - **lcd_print(x,y,message)** takes 3 inputs -
     - **x, y:** x position and y position at which we want to start printing
     - **message:** String which holds the text to print.
   - The **setCursor(x,y)** to set the cursor position.
   - **print(message)** is used to print the message on the screen.



```
void lcd_print(int x , int y , String message){
  lcd.setCursor(x,y);
  lcd.print(message);
}
```



2. Understand the LCD grid display:



LCD Custom Character Generator

Support character lcd and create code for Arduino.

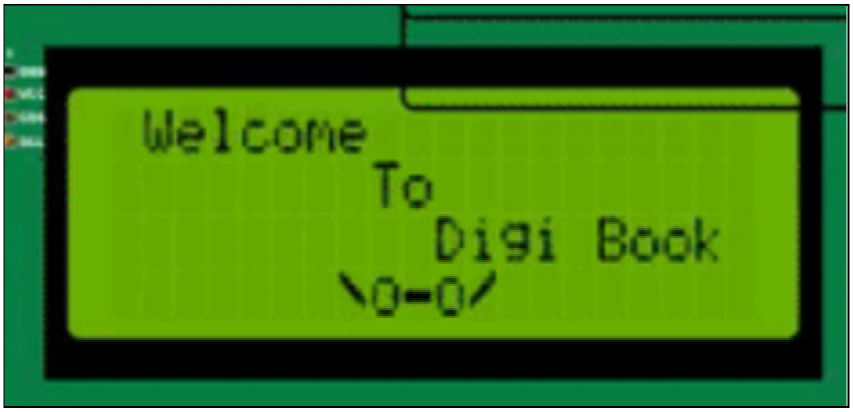
Color ● Green ○ Blue
Microcontroller ● Arduino
Interfacing ● Parallel ○ I2C
Data Type ● Binary ○ Hex
Code

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // RS,

byte customChar[] = {
  B00000,
  B00000,
  B00000,
  B00100,
  B01010,
  B10001,
  B10001,
  B10001
};
```

Link

- Arduino LCD Circuit
- Arduino LCD I2C Circuit
- Arduino LCD I2C library

3. **Display a spectacles animation on the LCD display:**



```
Welcome
      To
        Digi Book
     \0-0/
```

To display the above animation on the LCD display, follow the steps below:

1. Define the animations in **lcd_custom_character.h.**
   The animations of left_stick, right_stick, middle_stick, open_eye and closed_eye are defined in the file. include it using # include

```
#include "lcd_custom_character.h"
```

2. Define **animate()** to create the needed animation. We will use the **createChar()** method to assign an id to the customized characters

```
void animate() {
  lcd.createChar(0, left_stick);
  lcd.createChar(1, right_stick);
  lcd.createChar(2, open_eye);
  lcd.createChar(3, middle_stick);
  lcd.createChar(4, closed_eye);
}
```

3. Define **display_char()** method to display each character at a specific position using its id.

```
void display_char(int x, int y, int id) {
  lcd.setCursor(x, y);
  lcd.write(id);
}
```

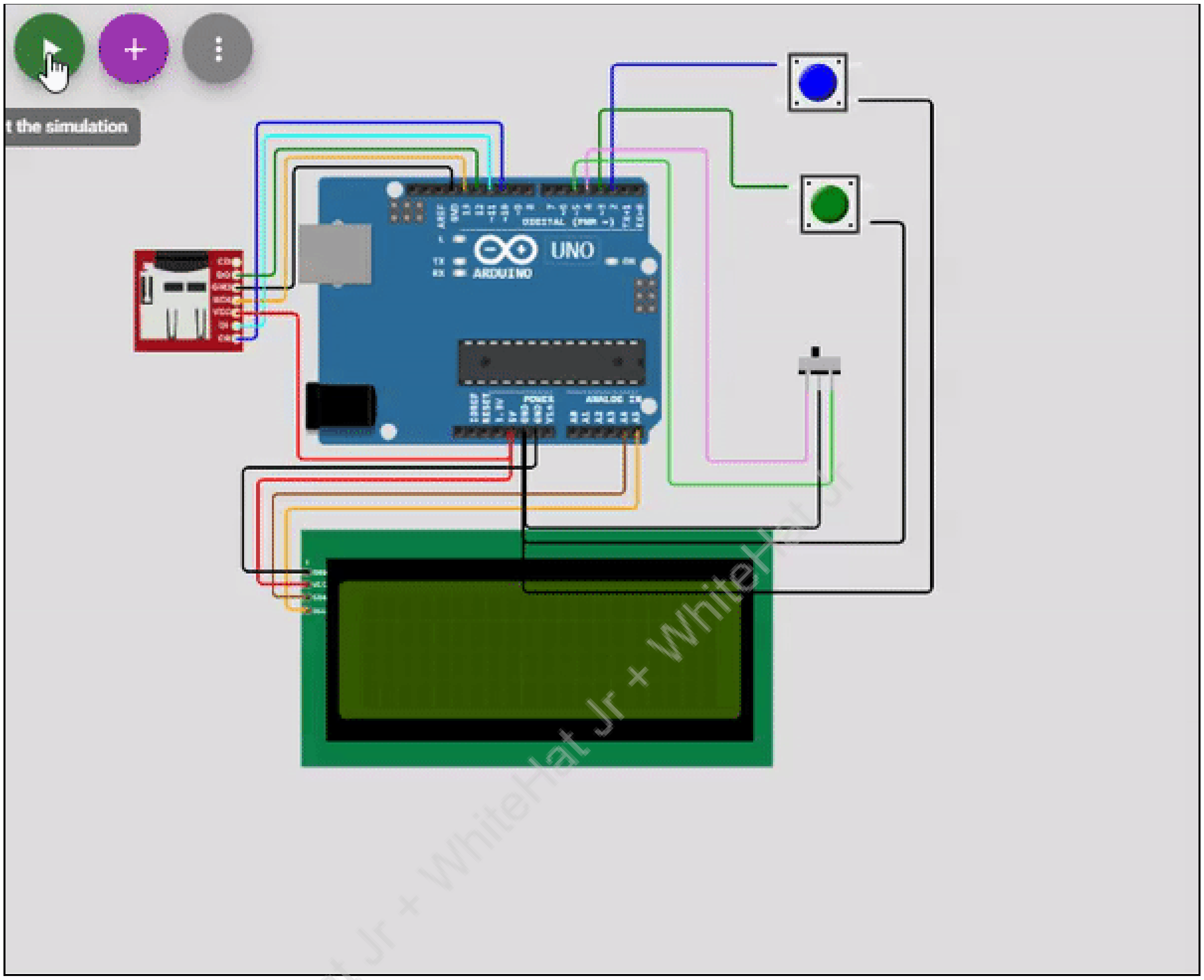
4. Use a for loop to create the blinking effect of the eye.

```
void animate() {

  lcd.createChar(0, left_stick);
  lcd.createChar(1, right_stick);
  lcd.createChar(2, open_eye);
  lcd.createChar(3, middle_stick);
  lcd.createChar(4, closed_eye);

  for (int i = 0; i < 5; i++) {
    display_char(11, 3, 1);
    display_char(9, 3, 3);
    display_char(7, 3, 0);
    if (i % 2 == 0) {
      display_char(8, 3, 2);
      display_char(10, 3, 2);
      delay(1100);
    }
    else {
      display_char(8, 3, 4);
      display_char(10, 3, 4);
      delay(100);
    }
  }
}
```

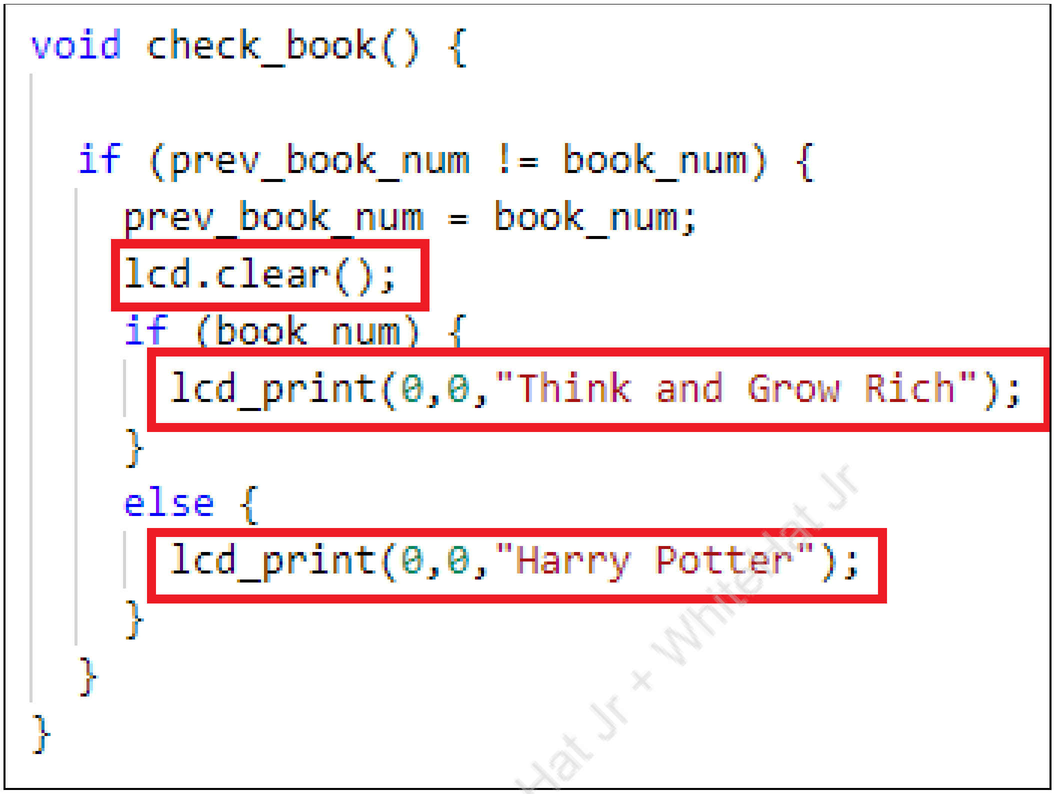
5. Call animate from the **setup()** metod.

```
void setup() {
  Serial.begin(9600);
  SD.begin();
  next.setDebounceTime(50);
  prev.setDebounceTime(50);
  select_book.setDebounceTime(50);
  open_book.setDebounceTime(50);

  lcd.init();
  lcd.backlight();
  lcd.clear();

  welcome();
  animate();
  delay(1000);
}
```

Reference Output:

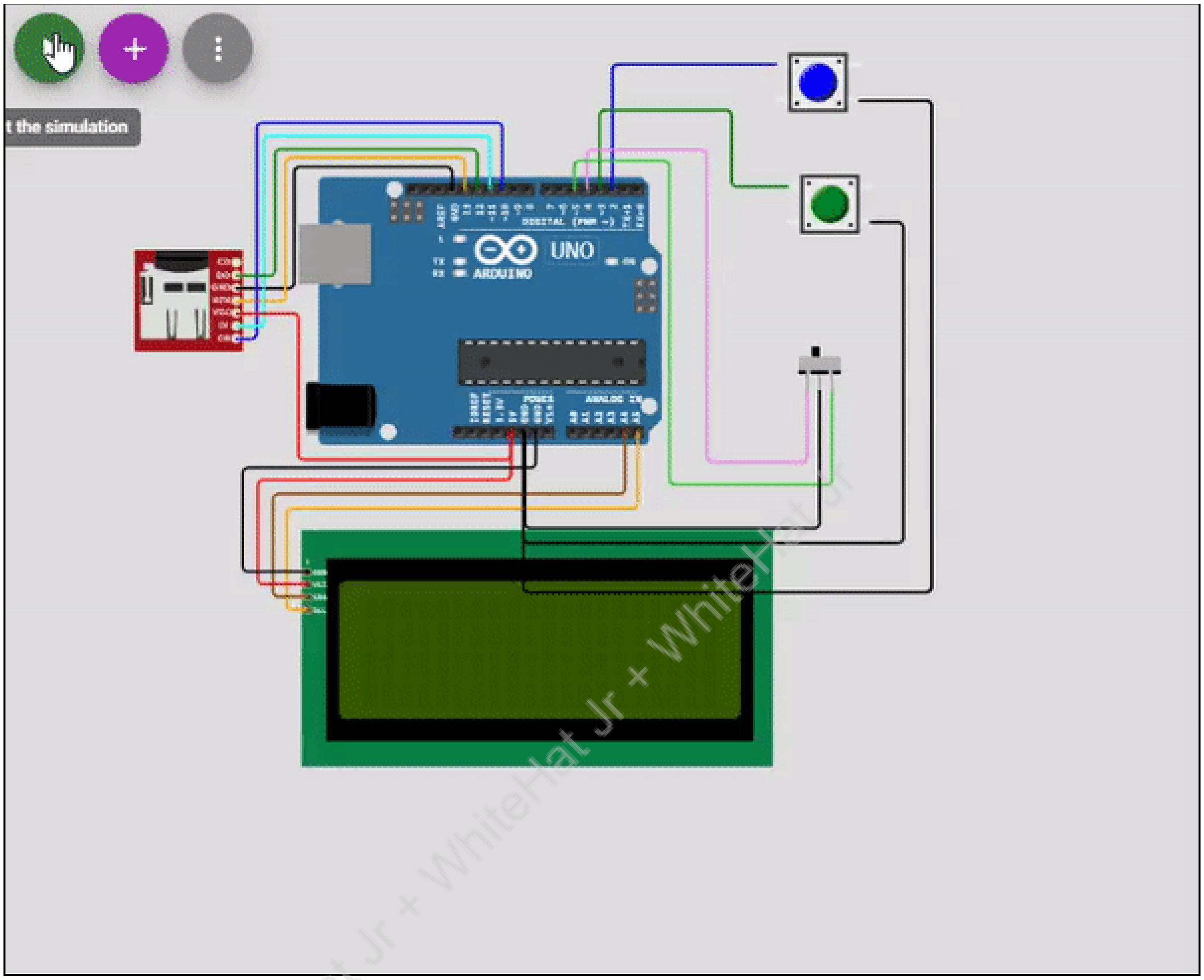


4. **Display the contents of the book on the LCD display by changing the check_book() method.**

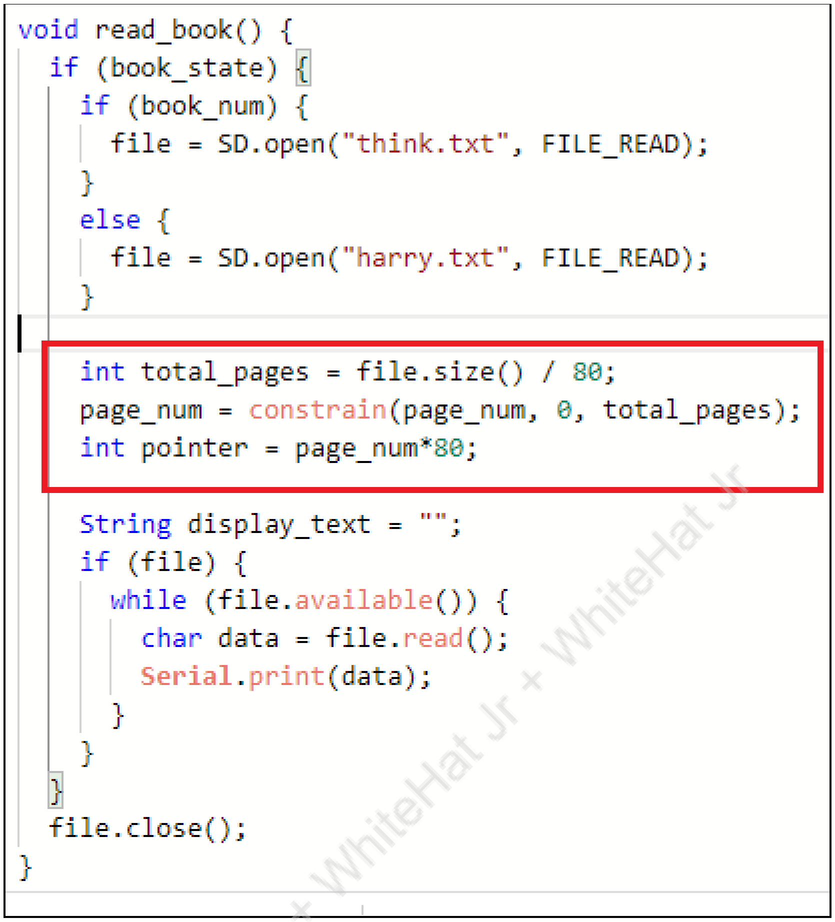


```
void check_book() {

  if (prev_book_num != book_num) {
    prev_book_num = book_num;
    lcd.clear();
    if (book_num) {
      lcd_print(0,0,"Think and Grow Rich");
    }
    else {
      lcd_print(0,0,"Harry Potter");
    }
  }
}
```

Reference Output:

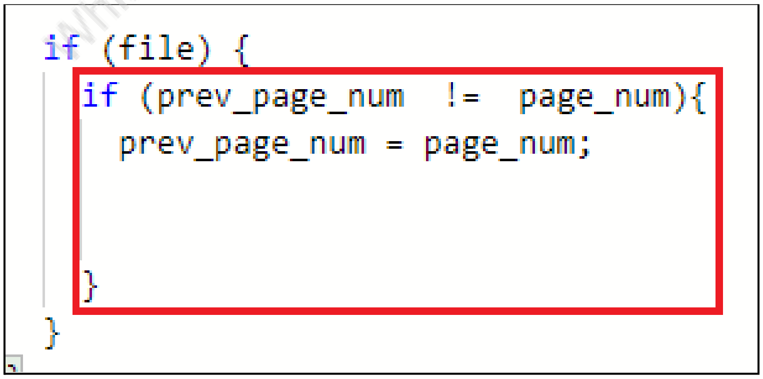


5. **Define page numbers so that each page can be displayed one at a time.**
   The LCD display can print 80 characters at a time. So, break down the book into 80 characters per page.



```
void read_book() {
  if (book_state) {
    if (book_num) {
      file = SD.open("think.txt", FILE_READ);
    }
    else {
      file = SD.open("harry.txt", FILE_READ);
    }

    int total_pages = file.size() / 80;
    page_num = constrain(page_num, 0, total_pages);
    int pointer = page_num*80;

    String display_text = "";
    if (file) {
      while (file.available()) {
        char data = file.read();
        Serial.print(data);
      }
    }
  }
  file.close();
}
```

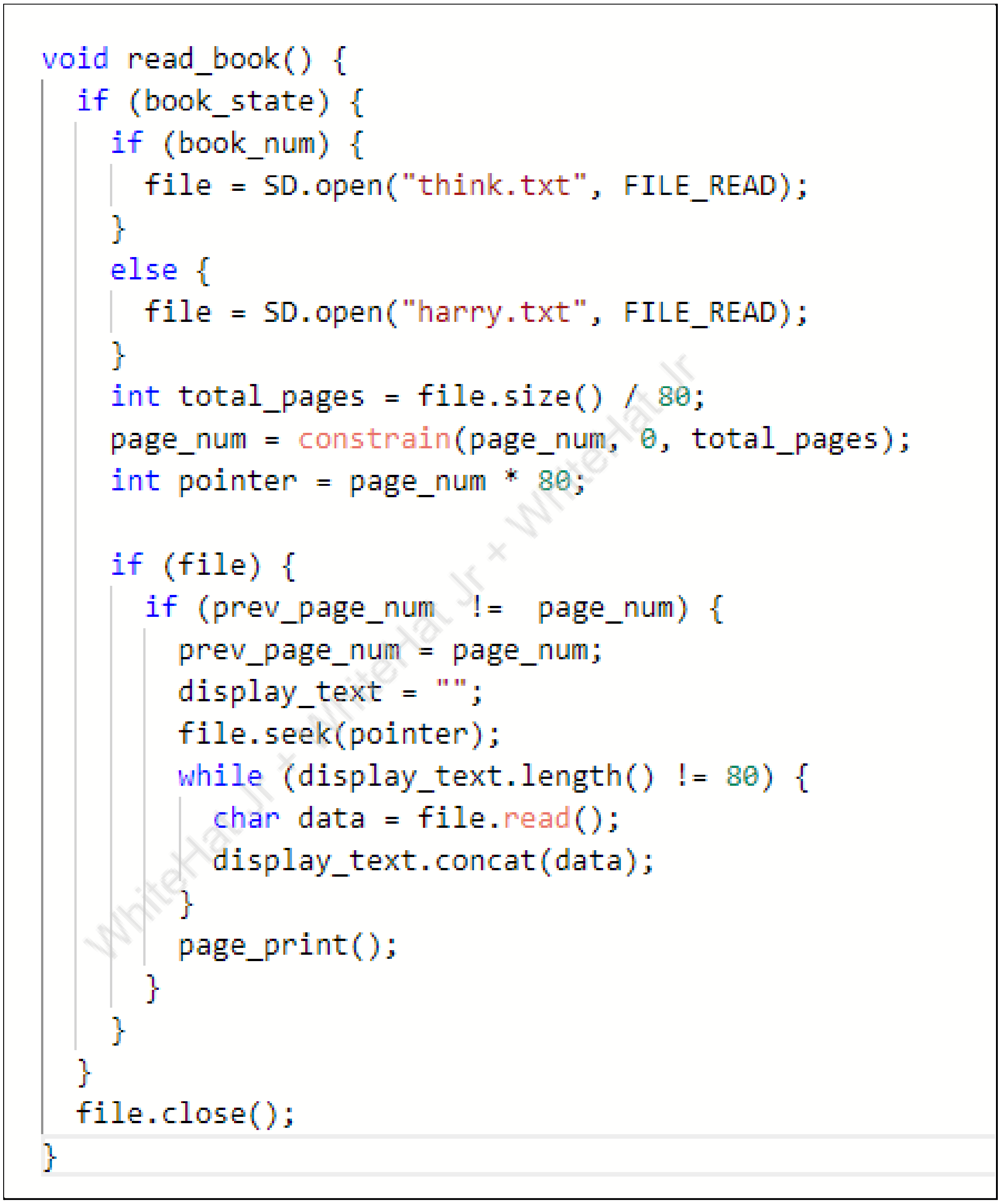
6. **Inside the if condition of the read_book(), remove the while loop and write an if condition to keep track of the pages.**



```
if (file) {
  if (prev_page_num  !=  page_num){
    prev_page_num = page_num;


  }
}
```

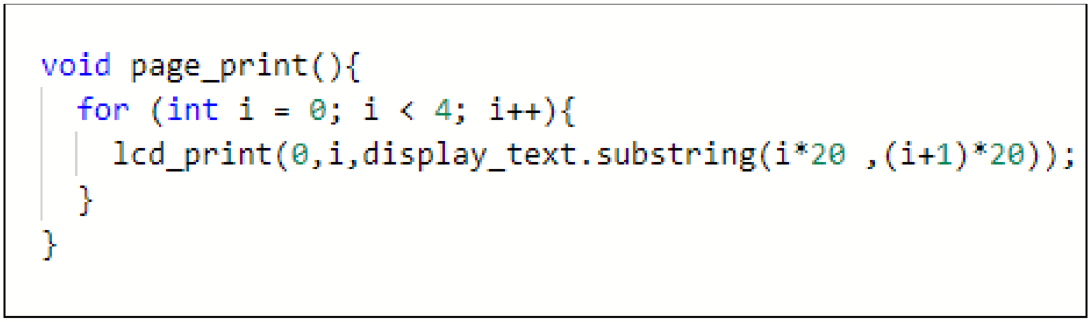
7. **Starting from the beginning of the book, store and display 80 characters at a time.**
   - **a.** Write **file.seek(pointer);** to move to the beginning of the file.
   - b. Store the next 80 characters to the variable **dispay_text.**

c. print the variable **display_text**.



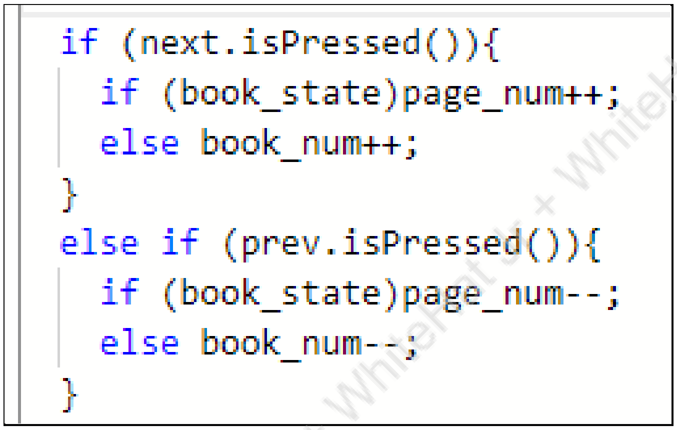
```
void read_book() {
  if (book_state) {
    if (book_num) {
      file = SD.open("think.txt", FILE_READ);
    }
    else {
      file = SD.open("harry.txt", FILE_READ);
    }
    int total_pages = file.size() / 80;
    page_num = constrain(page_num, 0, total_pages);
    int pointer = page_num * 80;

    if (file) {
      if (prev_page_num  !=  page_num) {
        prev_page_num = page_num;
        display_text = "";
        file.seek(pointer);
        while (display_text.length() != 80) {
          char data = file.read();
          display_text.concat(data);
        }
        page_print();
      }
    }
  }
  file.close();
}
```

8. **Define page_print method to print one complete page.**
   a. 80 characters are to be displayed using 20 columns and 4 rows.
   b. So print 20 characters in each of these 4 rows using a **for** loop.



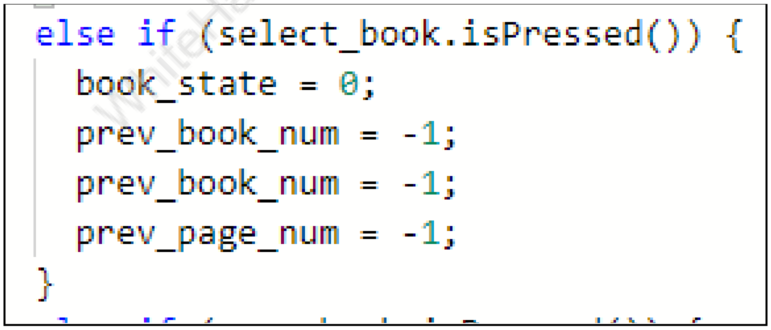
```
void page_print(){
  for (int i = 0; i < 4; i++){
    lcd_print(0,i,display_text.substring(i*20 ,(i+1)*20));
  }
}
```

9. **Redefine the next and previous buttons to go to next and previous pages**
    a. When book_state is 0, next and previous means next and previous books.
    b. When book_state is 1, next and previous means next and previous books.



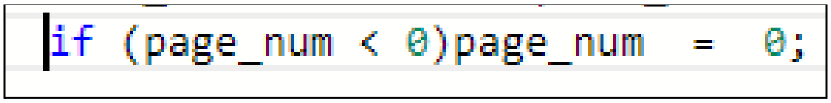
```
if (next.isPressed()){
  if (book_state)page_num++;
  else book_num++;
}
else if (prev.isPressed()){
  if (book_state)page_num--;
  else book_num--;
}
```

10. **Reset the page numbers when the next or previous book is selected**



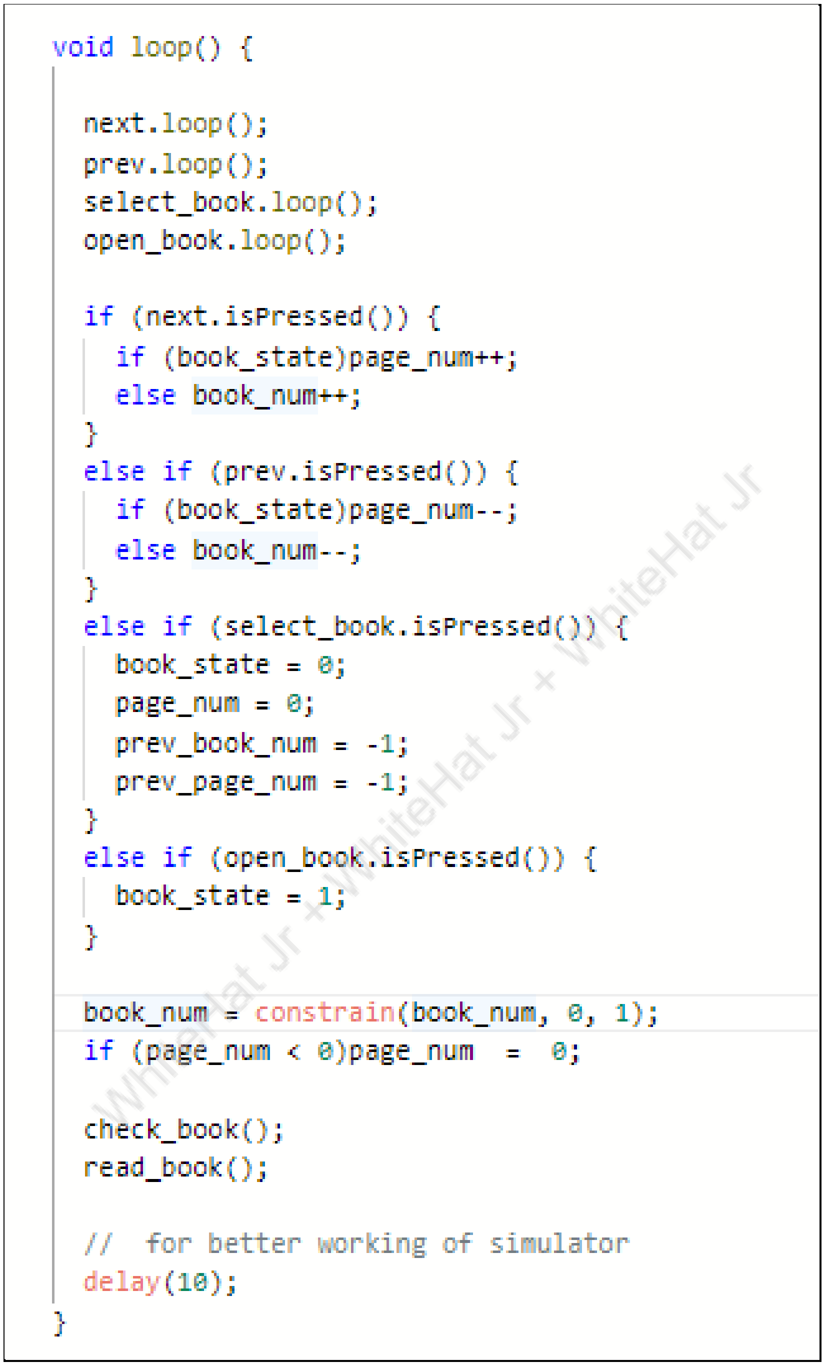
```
else if (select_book.isPressed()) {
  book_state = 0;
  prev_book_num = -1;
  prev_book_num = -1;
  prev_page_num = -1;
}
```

11. **Make sure that the page_num variable doesn't go below 0.**



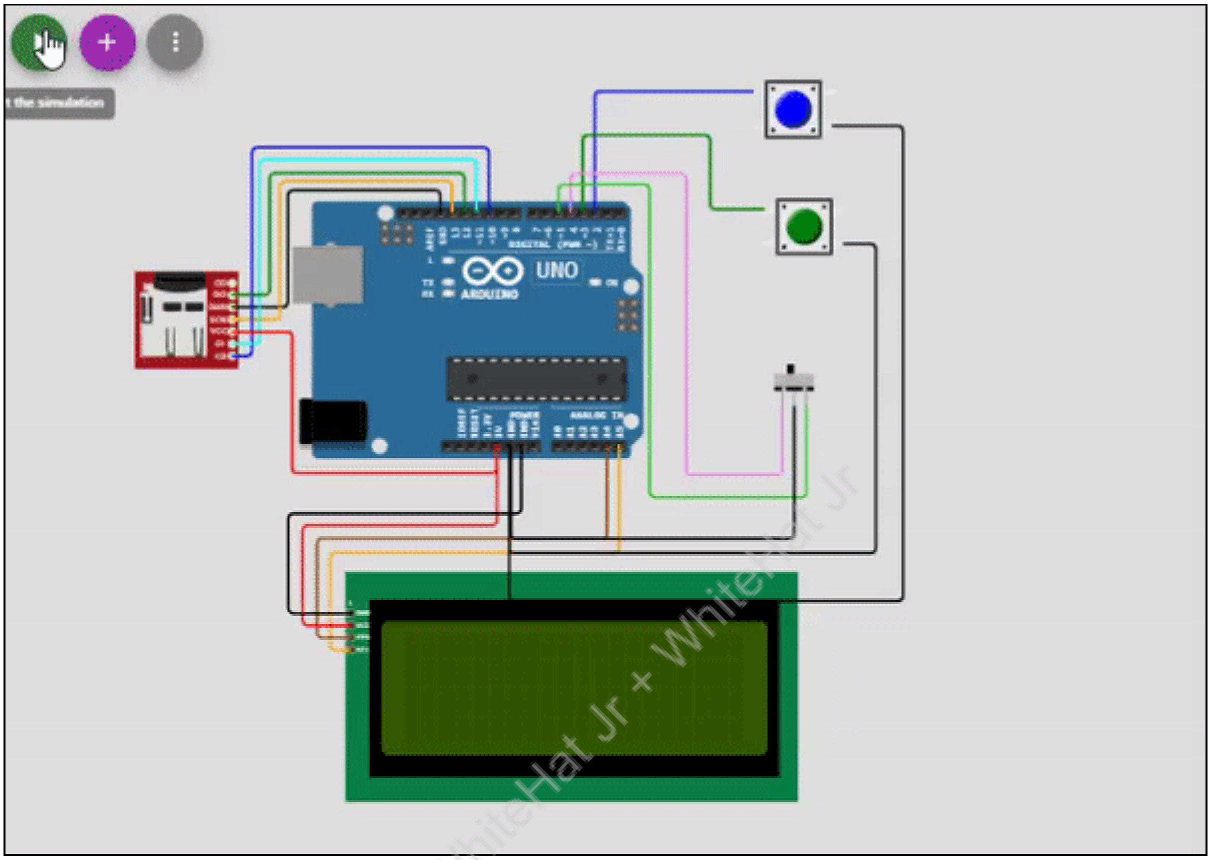
```
if (page_num < 0)page_num  =  0;
```

12. **After all the changes, the loop() method looks as shown below:**



```
void loop() {

  next.loop();
  prev.loop();
  select_book.loop();
  open_book.loop();

  if (next.isPressed()) {
    if (book_state)page_num++;
    else book_num++;
  }
  else if (prev.isPressed()) {
    if (book_state)page_num--;
    else book_num--;
  }
  else if (select_book.isPressed()) {
    book_state = 0;
    page_num = 0;
    prev_book_num = -1;
    prev_page_num = -1;
  }
  else if (open_book.isPressed()) {
    book_state = 1;
  }

  book_num = constrain(book_num, 0, 1);
  if (page_num < 0)page_num  =  0;

  check_book();
  read_book();

  //  for better working of simulator
  delay(10);
}
```

Reference Output:



Click here to view the reference video.

What's NEXT?

In the next class, we will learn how to use an RTC module and create a smart clock using it.

Expand Your Knowledge

Read more about LCD displays here