SMART BELL

PROJECT DESCRIPTION

The Smart Bell System is an Arduino-based automatic bell ringing system designed for schools, colleges, offices, and institutions. It rings a bell automatically at predefined time schedules, displays current time and next bell time on an LCD, and allows time editing and clock setting using a 4×4 keypad. This system eliminates manual bell operation and ensures accurate, timely bell ringing with flexibility to update schedules.

PIN CONNECTIONS

LCD (I2C – Address 0x27)

LCD Pin	Arduino Pin
VCC	5V
GND	GND
SDA	A4
SCL	A5

Keypad (4×4)

Keypad Pin	Arduino Pin
R1	9
R2	8
R3	7
R4	6
C1	5
C2	4
C3	3
C4	2

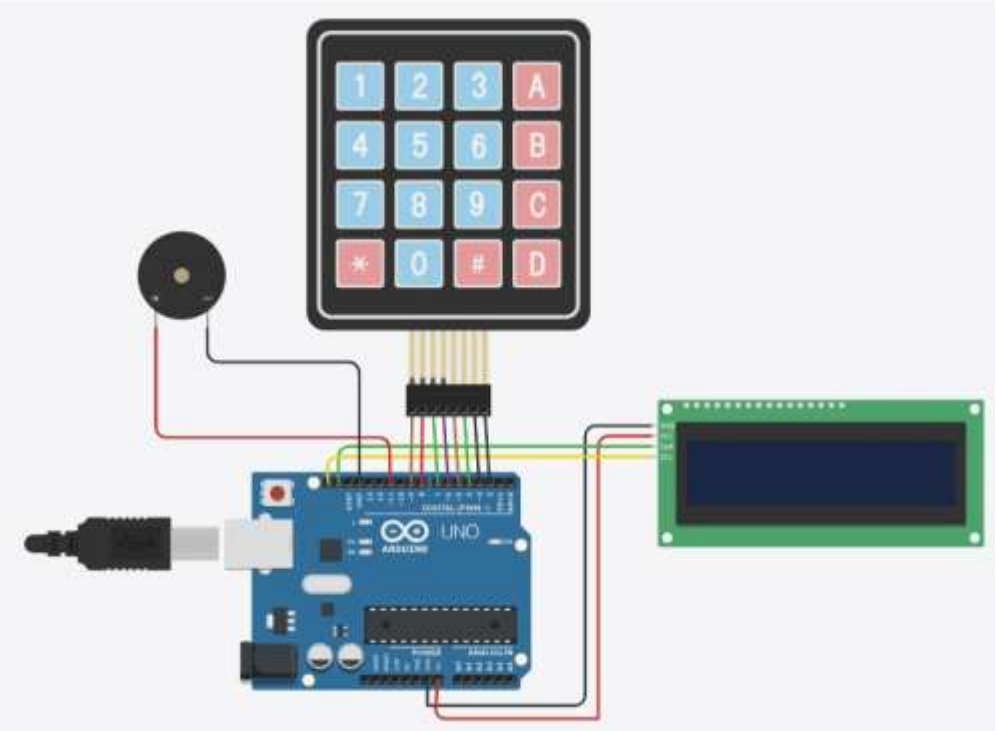
Relay Module

Relay Pin	Arduino Pin
VCC	5V
GND	GND
IN	D10

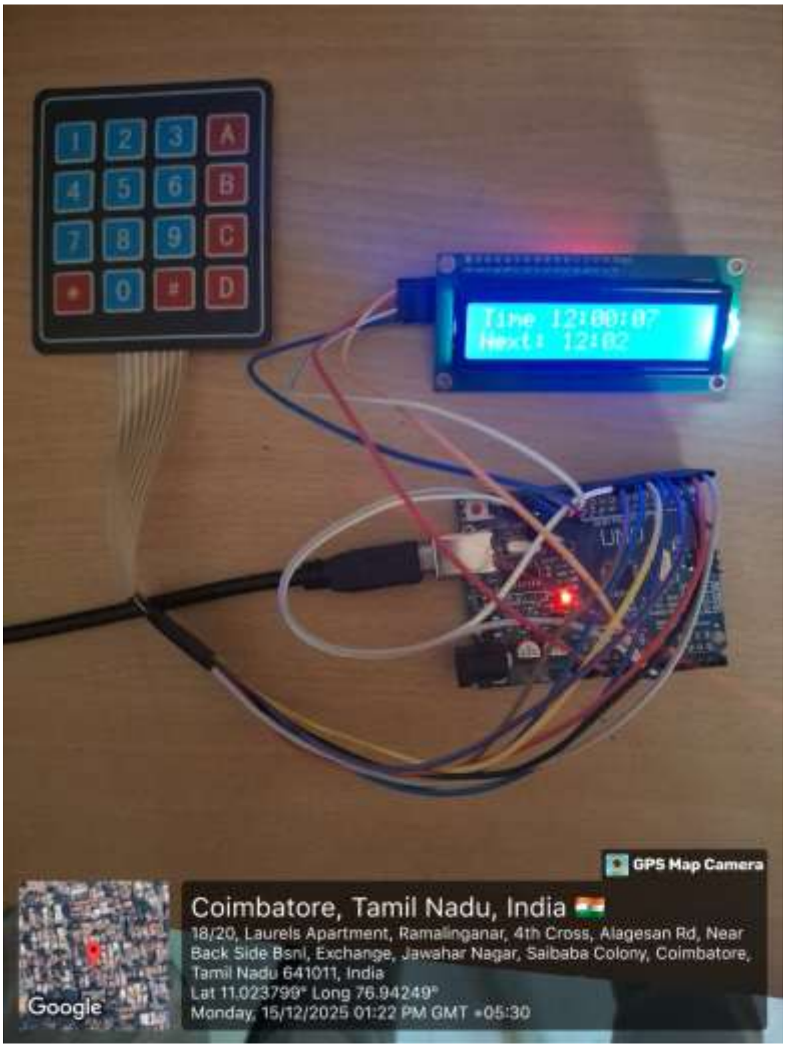
Buzzer

Buzzer Pin	Arduino Pin
+	D11
–	GND

CIRCUIT DIAGRAM



PROJECT PHOTOS


Time 12:00:07
Next: 12:02
GPS Map Camera
Coimbatore, Tamil Nadu, India
18/20, Laurels Apartment, Ramalinganar, 4th Cross, Alagesan Rd, Near Back Side Bsnl, Exchange, Jawahar Nagar, Saibaba Colony, Coimbatore, Tamil Nadu 641011, India
Lat 11.023799° Long 76.94249°
Monday, 15/12/2025 01:22 PM GMT +05:30
Google

WORKING PROCEDURE

1. **System Initialization**
   The LCD initializes and displays *"School Bell Ringing"*. The relay and buzzer remain OFF, and the internal clock starts from the preset time.
2. **Real-Time Clock Operation**
   The Arduino uses millis() to update seconds, minutes, and hours. The current time is shown on the LCD in HH:MM:SS format.
3. **Bell Schedule Storage**
   Bell timings are stored as HHMM strings in an array, allowing multiple scheduled bell times.
4. **Display Operation**
   The LCD displays the current time on the first line and the next scheduled bell time on the second line.
5. Keypad Control
   The keypad is used to view bell count, scroll bell times, edit bell schedules, and set the system clock.
6. **Bell Time Editing**
   The user selects a bell number, enters a new time, and saves it using the keypad.
7. **Clock Setting**
   The system time can be updated by entering a new HHMM value through the keypad.
8. **Automatic Bell Ringing**
   When the current time matches a stored bell time, the relay and buzzer activate for 3 seconds and then turn OFF automatically.

CODE

```
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

// Pins
int relayPin = 10;
int buzzerPin = 11; // Active Buzzer

// Time
int hourNow = 12;
int minuteNow = 0;
int secondNow = 0;
String bellTimes[8] = {"1202","1205","1210","1215","1230","1300","1400","1500"};
int totalTimes = 8;
unsigned long lastSecondCheck = 0;

// Keypad
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] =
{
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {9, 8, 7, 6};
byte colPins[COLS] = {5, 4, 3, 2};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
int mode = 0;
int selectedBell = -1;
String inputBuffer = "";

// Next Bell
 String getNextBell() {
  String now = "";
  if(hourNow < 10) now += "0";
  now += hourNow;
  if(minuteNow < 10) now += "0";
  now += minuteNow;
  for(int i=0;i<totalTimes;i++){
    if(bellTimes[i] > now) return bellTimes[i];
  }
  return "None";
}

// Display Time & Next
void displayMain(){
  lcd.setCursor(0,0);
  lcd.print("Time ");
  if(hourNow<10) lcd.print("0");
  lcd.print(hourNow);
  lcd.print(":");
  if(minuteNow<10) lcd.print("0");
  lcd.print(minuteNow);
  lcd.print(":");
  if(secondNow<10) lcd.print("0");
  lcd.print(secondNow);
  lcd.setCursor(0,1);
  lcd.print("Next: ");
  String nxt = getNextBell();
  if(nxt=="None") lcd.print("None     ");
  else {
    lcd.print(nxt.substring(0,2));
    lcd.print(":");
    lcd.print(nxt.substring(2,4));
    lcd.print("     ");
  }
}

void setup(){
  pinMode(relayPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  digitalWrite(relayPin, LOW);
  digitalWrite(buzzerPin, LOW);
  lcd.init();
  lcd.backlight();
  lcd.print("School Bell Ringing");
  delay(1500);
  lcd.clear();
}

void loop(){

  // Tick clock every 1 second
  if(millis() - lastSecondCheck >= 1000){
    lastSecondCheck = millis();
    secondNow++;
    if(secondNow >= 60){
      secondNow = 0;
      minuteNow++;
      if(minuteNow >= 60){
        minuteNow = 0;
        hourNow++;
        if(hourNow >= 24) hourNow = 0;
      }
    }
  }
  if(mode == 0) displayMain();
  char key = keypad.getKey();
  if(!key) return;

  // Show total bells
  if(key == '*'){
    lcd.clear();
    lcd.print("Periods: ");
    lcd.print(totalTimes);
    delay(2000);
    lcd.clear();
    return;
  }
```

```
// Scroll bell times
if(key == '#'){
  lcd.clear();
  for(int i=0;i<totalTimes;i++){
    lcd.setCursor(0,0);
    lcd.print("Bell ");
    lcd.print(i+1);
    lcd.setCursor(0,1);
    lcd.print(bellTimes[i].substring(0,2));
    lcd.print(":");
    lcd.print(bellTimes[i].substring(2,4));
    delay(1200);
    lcd.clear();
  }
  return;
}

// A = Edit Bell
if(key == 'A'){
  mode = 1;
  lcd.clear();
  lcd.print("Select Bell (1-8)");
  selectedBell = -1;
  inputBuffer="";
  return;
}
// C = Set Clock
if(key == 'C'){
  mode = 2;
  inputBuffer="";
  lcd.clear();
  lcd.print("Set Time HHMM");
  lcd.setCursor(0,1);
  return;
}

// Number Pressed
if(key >= '0' && key <= '9'){
  if(mode == 1 && selectedBell == -1){
    selectedBell = key - '1';
    if(selectedBell < 0 || selectedBell >= totalTimes){
      lcd.clear();
      lcd.print("Invalid Bell");
      delay(1000);
      mode = 0;
      return;
    }
    lcd.clear();
    lcd.print("Bell ");
    lcd.print(selectedBell+1);
    lcd.setCursor(0,1);
    lcd.print("Old:");
    lcd.print(bellTimes[selectedBell]);
    delay(1000);
    lcd.clear();
    lcd.print("New HHMM:");
    inputBuffer="";
    return;
  }
  if(inputBuffer.length() < 4){
    inputBuffer+=key;
    lcd.setCursor(0,1);
    lcd.print(inputBuffer + "   ");
  }
  return;
}

// B = SAVE
if(key == 'B'){
  if(inputBuffer.length() != 4){
    lcd.setCursor(0,1);
    lcd.print("Invalid");
    delay(800);
    mode=0;
    return;
  }
  if(mode == 1){
    bellTimes[selectedBell] = inputBuffer;
    lcd.clear();
    lcd.print("Bell Saved");
    delay(1000);
    mode=0;
  }
  if(mode == 2){
    hourNow = inputBuffer.substring(0,2).toInt();
    minuteNow = inputBuffer.substring(2,4).toInt();
    secondNow = 0;
    lcd.clear();
    lcd.print("Time Updated");
    delay(1000);
    mode=0;
  }
```

```
    inputBuffer="";
    return;
  }

  // RING CHECK HHMM only
  String nowTime = "";
  if(hourNow<10) nowTime+="0";
  nowTime+=hourNow;
  if(minuteNow<10) nowTime+="0";
  nowTime+=minuteNow;
  for(int i=0;i<totalTimes;i++){
    if(nowTime == bellTimes[i] && secondNow == 0){
      lcd.setCursor(0,1);
      lcd.print("RINGING!!!!");
      digitalWrite(relayPin, HIGH);
      digitalWrite(buzzerPin, HIGH);
      delay(3000);
      digitalWrite(relayPin, LOW);
      digitalWrite(buzzerPin, LOW);
    }
  }
}
```