Joshua Yang

Sang Han

BEE 425 Design Project Report

**Introduction**

In this project, we designed and produced a digital clock with an Arduino microcontroller without using Real Time Clock (RTC) module.

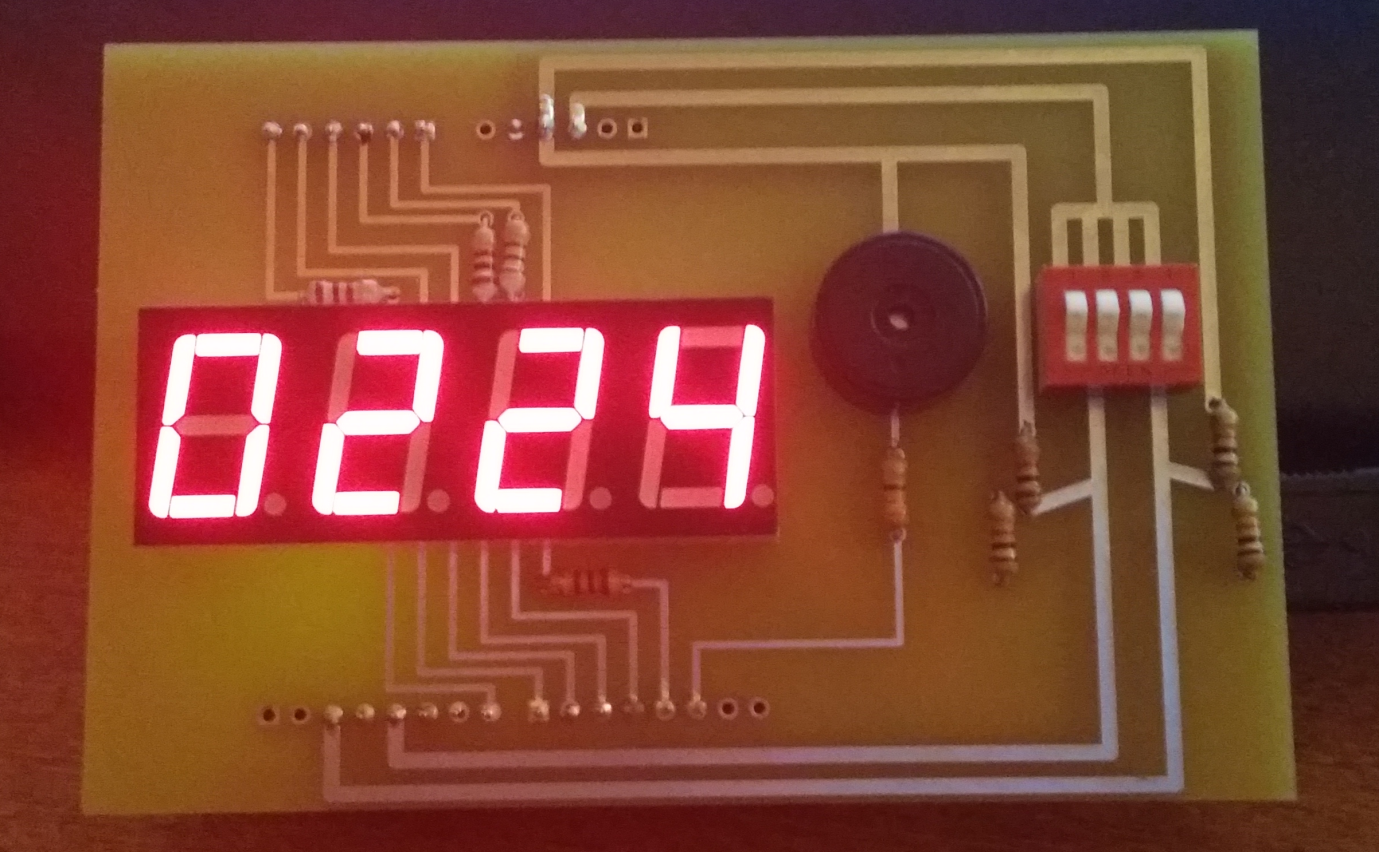
**Design Description**

List of parts

* 1 – Common cathode 4 digit 7 segment display
* 4 – SPST Switches
* 4 – 1K resistors
* 4 – 100 resistors
* 1 – 330 resistor
* 1 – Voltage controlled Buzzer
* 1 – Arduino Uno Microcontroller

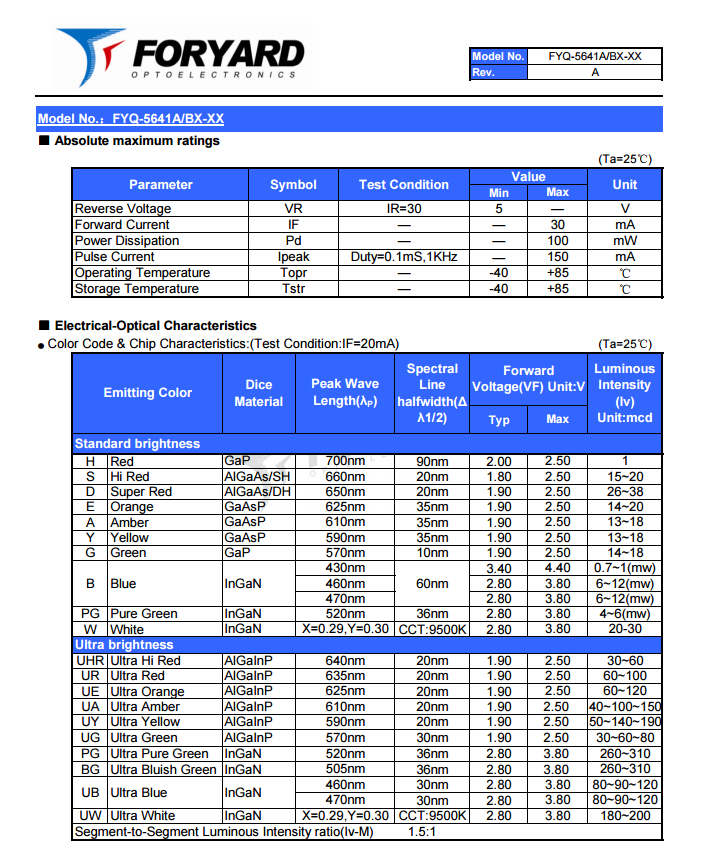
Function of the digital clock

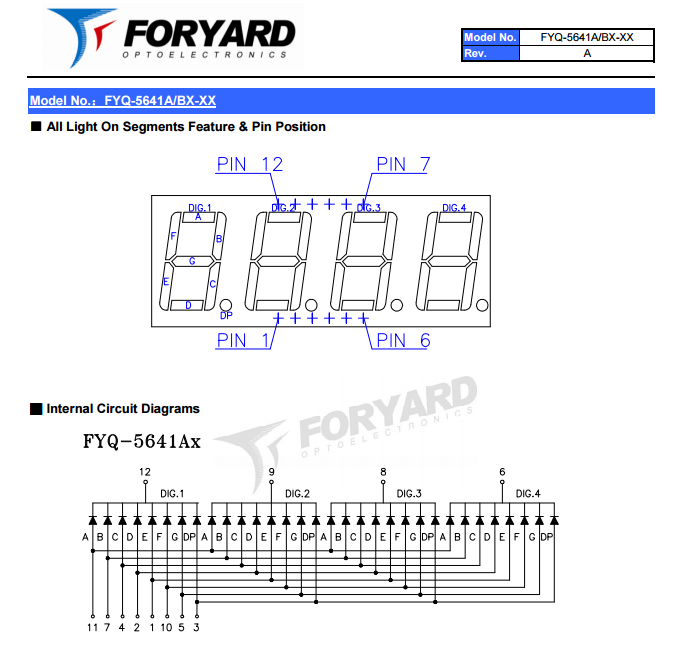
Our digital clock is a 24 hour clock with a functioning alarm system. It is controlled by 4 switches. Switches 1 and 2 are used to determine the different modes of the clock and switches 3 and 4 are used to change time variables. The dot LED (DP) is used to tell the user if the clock is in setting mode or not.



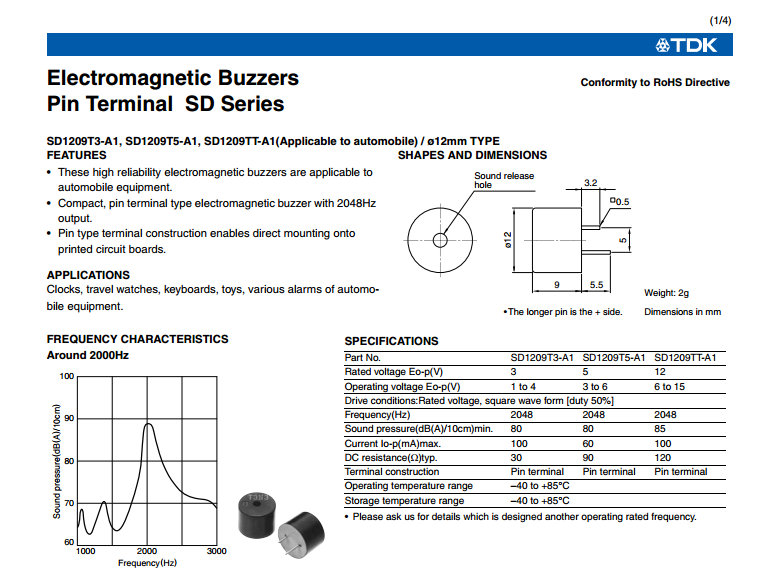
Datasheet

5641AS 4 Digit 7 segment display

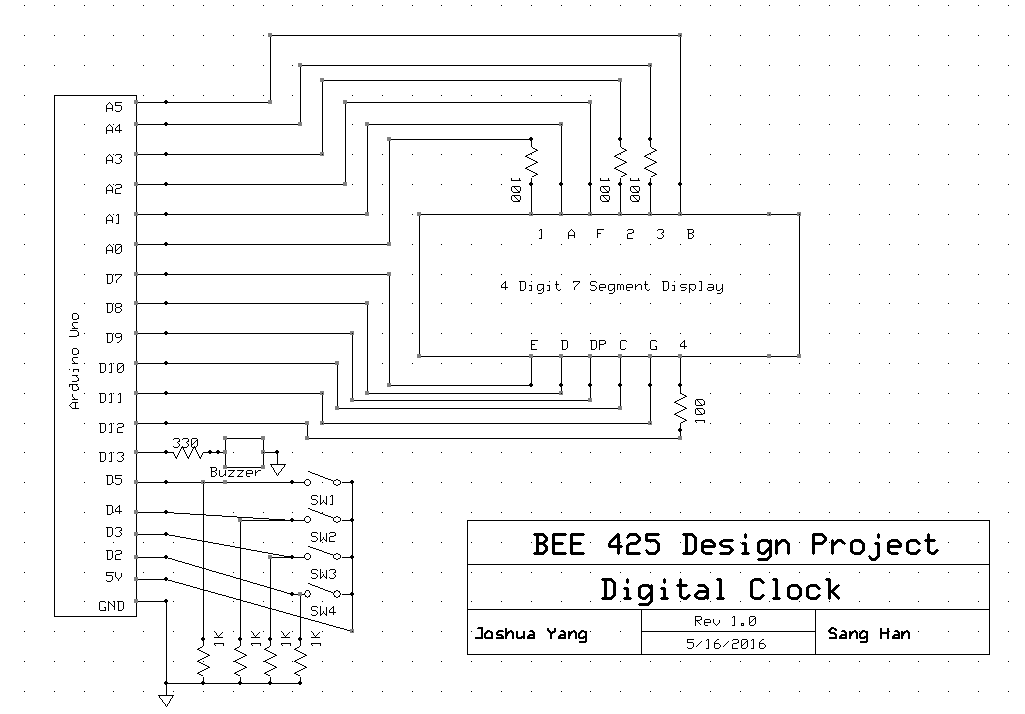




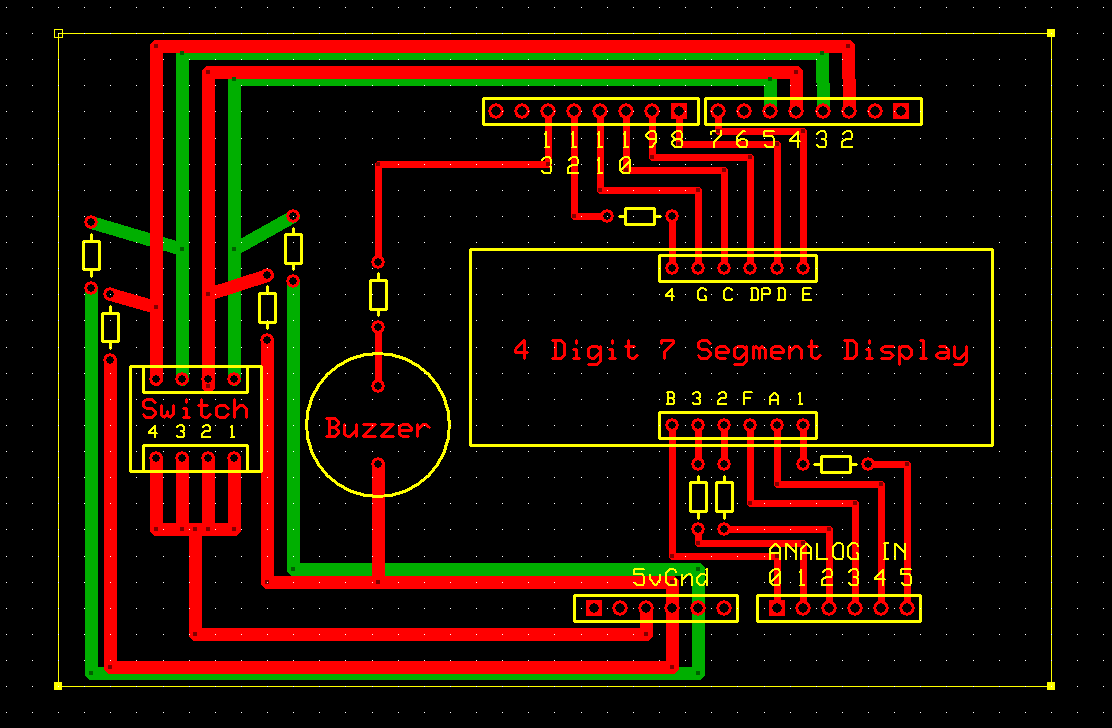
Electromagnetic Buzzer



Schematic



PCB design



Arduino source code

#define segA 18// connecting segment A

#define segB 14// connecting segment B

#define segC 10// connecting segment C

#define segD 8// connecting power D

#define segE 7// connecting segment E

#define segF 17// connecting segment F

#define segG 11//connecting segment G

#define DP 9// connecting power Dot

#define ones 12// connecting power ones digit

#define tens 15// connecting power tens digit

#define hund 16// connecting power hundreds digit

#define thou 19// connecting power thousands digit

#define alarm 5// connecting switch 1

#define setTime 4// connecting swtich 2

#define cHour 3// connecting switch 3

#define cMin 2// connecting switch 4

#define buzzer 13// connecting buzzer

void displaydigit (int digit) {

switch (digit)

{

case 0:// show ”0” on display

digitalWrite(segA, HIGH);

digitalWrite(segB, HIGH);

digitalWrite(segC, HIGH);

digitalWrite(segD, HIGH);

digitalWrite(segE, HIGH);

digitalWrite(segF, HIGH);

digitalWrite(segG, LOW);

break;

case 1:// show ”1” on display

digitalWrite(segA, LOW);

digitalWrite(segB, HIGH);

digitalWrite(segC, HIGH);

digitalWrite(segD, LOW);

digitalWrite(segE, LOW);

digitalWrite(segF, LOW);

digitalWrite(segG, LOW);

break;

case 2:// show ”2” on display

digitalWrite(segA, HIGH);

digitalWrite(segB, HIGH);

digitalWrite(segC, LOW);

digitalWrite(segD, HIGH);

digitalWrite(segE, HIGH);

digitalWrite(segF, LOW);

digitalWrite(segG, HIGH);

break;

case 3:// show ”3” on display

digitalWrite(segA, HIGH);

digitalWrite(segB, HIGH);

digitalWrite(segC, HIGH);

digitalWrite(segD, HIGH);

digitalWrite(segE, LOW);

digitalWrite(segF, LOW);

digitalWrite(segG, HIGH);

break;

case 4:// show ”4” on display

digitalWrite(segA, LOW);

digitalWrite(segB, HIGH);

digitalWrite(segC, HIGH);

digitalWrite(segD, LOW);

digitalWrite(segE, LOW);

digitalWrite(segF, HIGH);

digitalWrite(segG, HIGH);

break;

case 5:// show ”5” on display

digitalWrite(segA, HIGH);

digitalWrite(segB, LOW);

digitalWrite(segC, HIGH);

digitalWrite(segD, HIGH);

digitalWrite(segE, LOW);

digitalWrite(segF, HIGH);

digitalWrite(segG, HIGH);

break;

case 6:// show ”6” on display

digitalWrite(segA, HIGH);

digitalWrite(segB, LOW);

digitalWrite(segC, HIGH);

digitalWrite(segD, HIGH);

digitalWrite(segE, HIGH);

digitalWrite(segF, HIGH);

digitalWrite(segG, HIGH);

break;

case 7:// show ”7” on display

digitalWrite(segA, HIGH);

digitalWrite(segB, HIGH);

digitalWrite(segC, HIGH);

digitalWrite(segD, LOW);

digitalWrite(segE, LOW);

digitalWrite(segF, LOW);

digitalWrite(segG, LOW);

break;

case 8:// show ”8” on display

digitalWrite(segA, HIGH);

digitalWrite(segB, HIGH);

digitalWrite(segC, HIGH);

digitalWrite(segD, HIGH);

digitalWrite(segE, HIGH);

digitalWrite(segF, HIGH);

digitalWrite(segG, HIGH);

break;

case 9:// show ”9” on display

digitalWrite(segA, HIGH);

digitalWrite(segB, HIGH);

digitalWrite(segC, HIGH);

digitalWrite(segD, HIGH);

digitalWrite(segE, LOW);

digitalWrite(segF, HIGH);

digitalWrite(segG, HIGH);

break;

break;

}

}

void showtime (int number) //displays 4 digit number

{

// e.g. we have "1234"

displaydigit(number/1000); // segments are set to display "1"

digitalWrite(thou, LOW); // first digit on,

digitalWrite(hund, HIGH); // other off

digitalWrite(tens, HIGH);

digitalWrite(ones, HIGH);

delay (1);

number = number%1000; // remainder of 1234/1000 is 234

displaydigit(number/100); //// segments are set to display "2"

digitalWrite(thou, HIGH); // first digit is off

digitalWrite(hund, LOW); // second digit is on

delay (1); // and so on....

number = number%100;

displaydigit(number/10);

digitalWrite(hund, HIGH);

digitalWrite(tens, LOW);

delay (1);

number =number%10;

displaydigit(number);

digitalWrite(tens, HIGH);

digitalWrite(ones, LOW);

delay (1);

};

int timeconversion(long seconds) { //converts given number(seconds) to 4 digits representing hours and minutes

// if the input was 45130

int hours = seconds/3600 ; // hours will be 12

seconds = seconds % 3600; // the remainder is 1930

int minutes = seconds/60; // minutes will be 32

int convertedtime = (int)hours \* 100 + minutes; // this will add 1200 + 32 and return 1232

return convertedtime ;

}

void setup() {

pinMode(7, OUTPUT);

pinMode(8, OUTPUT);

pinMode(9, OUTPUT);

pinMode(10, OUTPUT);

pinMode(11, OUTPUT);

pinMode(12, OUTPUT);

pinMode(13, OUTPUT);

pinMode(14, OUTPUT);

pinMode(15, OUTPUT);

pinMode(16, OUTPUT);

pinMode(17, OUTPUT);

pinMode(18, OUTPUT);

pinMode(19, OUTPUT);

pinMode(2, INPUT);

pinMode(3, INPUT);

pinMode(4, INPUT);

pinMode(5, INPUT);

}

long currenttime = 0;

long currenthour = 0;

long currentmin = 0;

unsigned long previousMillis = 0;

int alarmsetting = 0;

int settingTime = 0;

int changeHour = 0;

int changeMin = 0;

long alarmTime = 0;

long alarmhour = 0;

long alarmmin = 0;

int test = 0;

void loop ()

{

alarmsetting = digitalRead(alarm); // read the 4 switches

settingTime = digitalRead(setTime);

changeHour = digitalRead(cHour);

changeMin = digitalRead(cMin);

unsigned long currentMillis = millis(); // use millis() to keep track of time

// Mode 2 alarm setting mode (Switch 1 ON, Switch 2 OFF)

if (alarmsetting == HIGH && settingTime == LOW) {

digitalWrite(DP, HIGH); //ligth up the dot display to indicate that the clock is in settings mode

if (changeHour == HIGH) { // when switch 3 is on in Mode 2

if ((unsigned long)(currentMillis - previousMillis) >= 500) { //wait .5 seconds and add 3600 to //alarmhour (3600 seconds = 1hour)

alarmhour = alarmhour + 3600;

if(alarmhour >= 86400) { //if the hours goes higher than 23, reset it to 0

alarmhour = 0;

}

previousMillis = currentMillis;

}

}

if (changeMin == HIGH) { // when switch 4 is on in Mode 2

if ((unsigned long)(currentMillis - previousMillis) >= 500) { //wait .5 seconds and add 60 to alarmmin //(60 seconds = 1min)

alarmmin = alarmmin + 60;

if(alarmmin >= 3600) { //if the minutes goes higher than 59, reset it to 0

alarmmin = 0;

}

previousMillis = currentMillis;

}

}

alarmTime = alarmhour+alarmmin;

showtime(timeconversion(alarmTime)); // send it to timeconversion to get 4 digit value and displays alarm time

}

//Mode 3 currenttime setting mode (Switch 1 ON, Switch 2 ON)

else if (alarmsetting == HIGH && settingTime == HIGH) {

digitalWrite(DP, HIGH); //ligth up the dot display to indicate that the clock is in settings mode

if (changeHour == HIGH) { // when switch 3 is on in Mode 3

if ((unsigned long)(currentMillis - previousMillis) >= 500) { //wait .5 seconds and add 3600 to //currenthour (3600 seconds = 1hour)

currenthour = currenthour + 3600;

if(currenthour >= 86400) { //if the hours goes higher than 23, reset it to 0

currenthour = 0;

}

previousMillis = currentMillis;

}

}

if (changeMin == HIGH) { // when switch 4 is on in Mode 2

if ((unsigned long)(currentMillis - previousMillis) >= 500) { //wait .5 seconds and add 60 to alarmmin (60 //seconds = 1min)

currentmin = currentmin + 60;

if(currentmin >= 3600) { //if the minutes goes higher than 59, reset it to 0

currentmin = 0;

}

previousMillis = currentMillis;

}

}

currenttime = currenthour+currentmin;

showtime(timeconversion(currenttime)); // send it to timeconversion to get 4 digit value and displays changed //current time

}

//Mode 4 alarm mode (swtich 1 off, switch 2 on)

else if (alarmsetting == LOW && settingTime == HIGH) {

digitalWrite(DP,LOW); //turn off the dot display to indicate that the clock is time showing mode

if ((unsigned long)(currentMillis - previousMillis) >= 1000) { // every second add 1 to current time

currenttime++;

previousMillis = currentMillis;

}

if (currenttime >= 86400) { //if the current goes higher than 2359, reset it to 0

currenttime = 0;

}

showtime(timeconversion(currenttime)); //show current time

if (alarmTime == currenttime) { // if the alarm time set is equal to current time

digitalWrite(buzzer,HIGH); // turn on the alarm (doesn't turn off until switch 2 is OFF)

}

}

// Mode 1 clock mode (switch 1 off, switch 2 off)

else {

digitalWrite(buzzer,LOW); // switch 2 on if alarm was triggered, turns off the alarm

digitalWrite(DP,LOW); //turn off the dot display to indicate that the clock is time showing mode

if ((unsigned long)(currentMillis - previousMillis) >= 1000) { // every second add 1 to current time

currenttime++;

previousMillis = currentMillis;

}

if (currenttime >= 86400) { //if the current goes higher than 2359, reset it to 0

currenttime = 0;

}

showtime(timeconversion(currenttime)); //show current time

}

}

**Theory of operation**

The 4 digit segment display must be multiplexed in order to show 4 different digits. This means that only one digit should be powered up at a time. Therefore the digits are constantly blinking but it is blinking too fast so to the human eye so it seems like all 4 digits are on at the same time. This is done by the controlling the Arduino I/O pins to send out voltages at desired time.

The Arduino I/O pins have the ability to read data. In our case the data is voltage which tells us the state of the switch. If the switch is on the Arduino reads HIGH and if the switch is off the Arduino reads LOW.

The Arduino at every clock checks for the state of the 4 switches. And using the readings of the switches 1 and 2 it puts the clock in to 4 different modes.

Mode 1 (Switch 1 = OFF, Switch 2 = OFF, DP = OFF)

Current time mode.

It always shows current time.

Mode 2 (Switch 1 = ON, Switch 2 = OFF, DP = ON)

Alarm setting mode.

In this mode the user can check alarm time and set alarm time.

In mode 2, the clock checks for state of switches 3 and 4. If switch 3 is ON the hour of the alarm time goes up by 1 every .5 seconds until it goes up to 23. If switch 4 is ON the minutes of the alarm time goes up by 1 every .5 seconds until it goes up to 59.

Mode 3 (Switch 1 = ON, Switch 2 = ON, DP = ON)

Current time changing mode.

In this mode the user can change current time.

Same as in mode 2, the clock checks for the state of switches 3 and 4 and change current time values accordingly.

Mode 4 (Switch 1 = OFF, Switch 2 = ON, DP = OFF)

Alarm activation mode.

In this mode the alarm is turned on. Current time will be shown just as in mode 1 but the buzzer will ring when the current time is equal to the alarm time. Put it back to mode 1 to turn off the buzzer.

**Conclusions**

This project taught us how to use the Arduino Microcontroller to design and build a working system. We also learned how to design a PCB board which will be useful in our career as an engineer. The clock is working but there are things that could have been better. In our design the resistors are only connected to the I/O pins that power the each digit but instead the resistors should connect I/O pins that power each segment. For future version we would fix this issue and instead of using switches a button should be used for more user friendly version.

**Acknowledgement**

Arnold Berger, Professor, UWB