



Digital Clock Project Report

The link to the circuit: <https://www.tinkercad.com/things/hWKjfHUz3XQ-fantastic-bombul/editel?sharecode=0FCQslyzDe9UGbBbx6tJy91MuQy6iCLH6FYhAGfZQFs>

Timer

```
noInterrupts(); //First, we need to disable interrupts by the following code.

TCCR1A = 0;
TCCR1B = 0;
TCNT1 = 0;

OCR1A = 62500; //Then, let's set compare match register to set our frequency as 16MHz/256

TCCR1B |= (1 << WGM12); //CTC Mode withh OCR1A
TCCR1B |= (1 << CS12); //Prescaler set to 256
TIMSK1 |= (1 << OCIE1A); //enable timer interrupt

interrupts(); //enable all interrupt

ISR(TIMER1_COMPA_vect) {
  ss = ss + 1; //increment second every time timer increases
}
```

Clock

With second incrementing every timer period, code just need to set minute and hour accordingly. If clock is set for AM/PM mode, code just checks if it is past noon or not, takes mode of hour with 12 and shows the clock.

Alarm

Alarm is off by default and it shown by a bell icon with no filling. If it is open, it will be shown with a bell icon with filling. When minute and hour of alarm matches with the clock the piezo buzzer will start to ring. We can snooze this by B4 button and the alarm will ring after 5 minutes. Since this snooze action can make minute increment past 60 minutes, code will set the hour if it has past the 60 minute threshold. To turn off the alarm, just push on B2 button and alarm will be disabled.

Temperature

TMP36 sensor sends a signal ranging between 0-1024. Taking the input and dividing it by 1024 will give us a percentage. This percentage is giving us a voltage between 0-1.75V. To get the 5V value, just multiply it by 5. By this multiplication, we are measuring between 0-175. To make minimum reading as -50, let's subtract 0.5. To convert millivolt to degree, just multiply it by 100.

This is done for Celcius. To convert this into Fahrenheit, just us the simple formula.

Fahrenheit = Celcius * 1.8 + 32

Adjust alarm and clock

```
if (digitalRead(timeB) == 1 && !isAlarmSetup) {
    lastTimeB = 1;

    //hold
    if (timeFlag == 0) {
        timeFlag = 1; //this flag is needed since we need to track if we are holding or not
        start = ss;
    }
    if (((ss + 60) - start) % 60 == 3) { //with the help of the clock, we can easily know we hold the button for 3 seconds
        isTimeSetup = 1;
    }
}
else if (digitalRead(timeB) == 0) {
    timeFlag = 0; //when we release the button we can reset flag to 0
}
```

Holding B1 and B2 for 3 seconds will activate timeSetup and AlarmSetup modes respectively. Since they are similar, let's just focus on timeSetup.

isTimeSetup firstly makes the focused area blink. With the help of millis() function, focused area (e.g. clock minute) will blink by clearing that part and showing the clock again.

With isTimeSetup, pressing the button will change mode since it checks for that mode every time it is pressed. Since press function checks for the PULL-UP, firRelease variable is there to make the first button release ineffective. First adjusting applies to minute. Pressing B1 will change the adjusting to hour. Pressing it again will reset timeSetup and make system function as default.