DESIGN AND DEVELOPMENT OF HARDWARE POMODORO TIMER USING ATMEGA328P BOARDS

Aryan Karamtoth
Department of Information Technology
Kakatiya Institute of Technology and Science Warangal

Laxmi Koundinya
Department of Computer Science and Engineering (Networks)
Kakatiya Institute of Technology and Science Warangal

ABSTRACT

In the era of digital electronics like mobile phones and other smart devices, we are bombarded with a lot of information, often unnecessary. Pomodoro timers have become an efficient way for people to make the best out of their time but the use of it on digital devices makes it distracting due to notifications from various apps. To overcome this, we developed a Hardware Pomodoro Timer which does away with all of the digital display devices and uses only traditional LEDs and Piezo Buzzers powered by an ATMega328P board to remind the user about focus and break times. This paper demonstrates the design and development of such device.

INTRODUCTION

Time is of the essence in the modern time and with the rise of social media and engagement-based applications, people are going through a shortage of time. This shortage of time is due to the overuse of social media apps. These apps tend to be very addictive in nature and make the user spend way too much time on them, causing the user to waste a lot of their productive time, thereby, hampering their studies or work life.

There have been many methods to overcome this issue. One such method is the Pomodoro Technique. Pomodoro Technique is a time management method which involves breaking work into certain time typically 25 minutes and followed by a short break.

Pomodoro Technique can be applied by an individual using a Pomodoro Timer app on their mobile phone or laptop. There are many pomodoro timers available across many operating systems and app stores for use.

However, one of the major challenges here is to stay focused from the prying eyes of your smartphone's social media apps. Your eyes are frequently drawn towards the screen and its contents, thereby reducing your focus on your productive work.

To overcome this issue, we have utilized Arduino Uno, powered by ATMega328P. to create a Hardware Pomodoro Timer. This custom-built appliance eliminates visual distractions of the user, by giving a buzzer to signal the end of each interval as well as utilize the on-board LEDs to indicate the flow of time. This appliance allows the user to concentrate completely on their tasks without any distractions and increase the focus, enhance the productivity too.

NEED FOR THE DEVELOPMENT

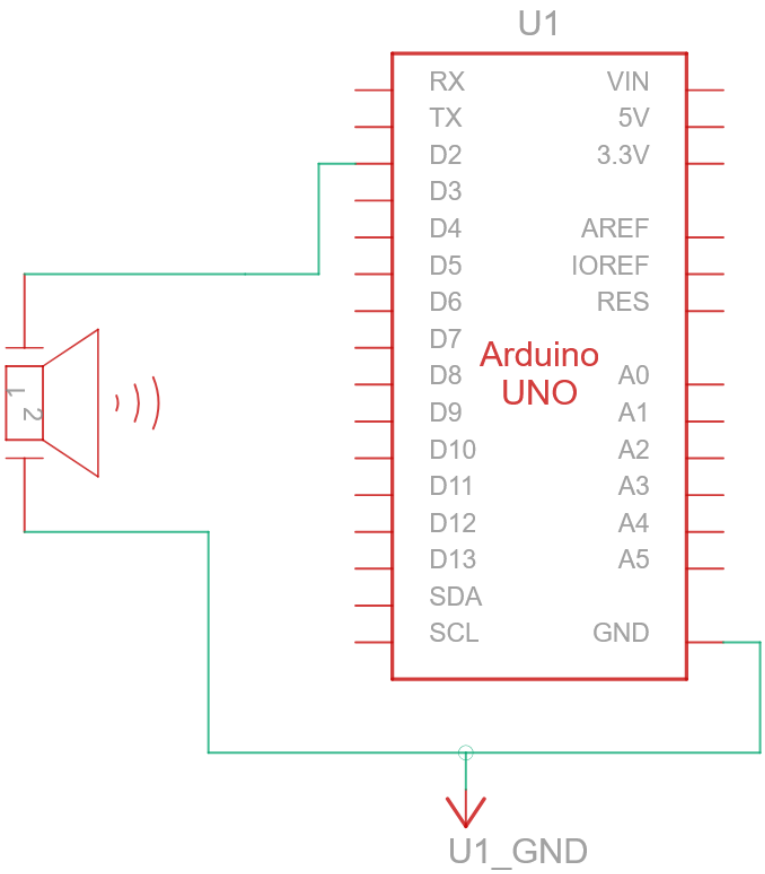
In general, digital tools are designed to help users stay organized and focused but they often become the main sources of distraction. Moreover, digital screens can be very distracting due to multitasking, notification, etc. For the individuals who are trying to avoid such distractions but couldn't, this Hardware Pomodoro Timer might be helpful.

COMPONENTS

We used the following components for this project.

Components: Piezo Buzzer, Arduino Uno, Jumpers, Soldering Iron, Solder, Desoldering kit, Computer, Voltage Source

SCHEMATIC



DEVELOPMENT

(i) Getting Ready

Firstly, we need to get our components ready for soldering and giving connections to various components across the board. Heat your soldering iron and get your components ready for soldering and giving connections to the board

(ii) Giving Connections

Get the board ready and take a jumper. Connect the Ground Pin (GND) to the negative pole of the voltage source using the jumper.

Take a piezo buzzer and two jumpers. Remove the connecting end of one end of the jumper such that the copper wiring inside is exposed. Connect this jumper from D0 or D2 pin and solder it to the positive end of the piezo buzzer. Connect the negative end of the buzzer to the ground.

(iii) Interfacing

For interfacing the hardware, we need a computer and a USB connector for Arduino Uno. Connect the USB to Arduino and connect the USB connector to the computer for interfacing.

Open the Arduino IDE for interfacing the board and components.

Variable Declarations

Initialise the buzzerPin as 2 (D2), timers as 25 mins and 5 mins and declare current and start time.

```
//declaration dump
const int buzzerPin = 2; //buzzer pin no 2 (digital)
unsigned long startMillis; //start millis
unsigned long currentMillis; //current millis
const unsigned long periodFlashDoro = 1500000; // 25 mins
const unsigned long periodEndPomodoro = 300000; // 5 mins
```

Functions

startPomodoro() is used to start the pomodoro timer

```
//function to buzz at the starting of pomodoro
void startPomodoro(){
  digitalWrite(buzzerPin,HIGH);
  delay(500);
```

```
  digitalWrite(buzzerPin,LOW);
}
```

flashDoro() is used to indicate the working of the pomodoro timer

```
//function to make the led flash to indicate the on going pomodoro timer
void flashDoro(){
  digitalWrite(LED_BUILTIN, HIGH);
  delay(500);
  digitalWrite(LED_BUILTIN, LOW);
  delay(500);
}
```

breakFlash() is used to indicate the break timer

```
//flash timer during 5 mins break
void breakFlash(){
  digitalWrite(LED_BUILTIN, HIGH);
  delay(3000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(3000);
}
```

endPomodoro() is used to indicate the end of pomodoro

```
//buzz sound to indicate the end of pomodoro
void endPomodoro(){

  digitalWrite(buzzerPin,HIGH);
  delay(5000);
  digitalWrite(buzzerPin,LOW);
}
```

Average start() and loop() functions

```
//average setup loop
void setup()
{
  Serial.begin(8600);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
```

```
}


//average main loop (main boy)
void loop()
{
  //start the pomodoro
  startPomodoro();
  startMillis = millis();
  while(millis() - startMillis < periodFlashDoro) {
    flashDoro();
  }
  //play at the end of pomodoro
  endPomodoro();
  startMillis = millis();
  while(millis() - startMillis < periodEndPomodoro) {
    breakFlash();
  }

}
```

That's it for the interfacing part.

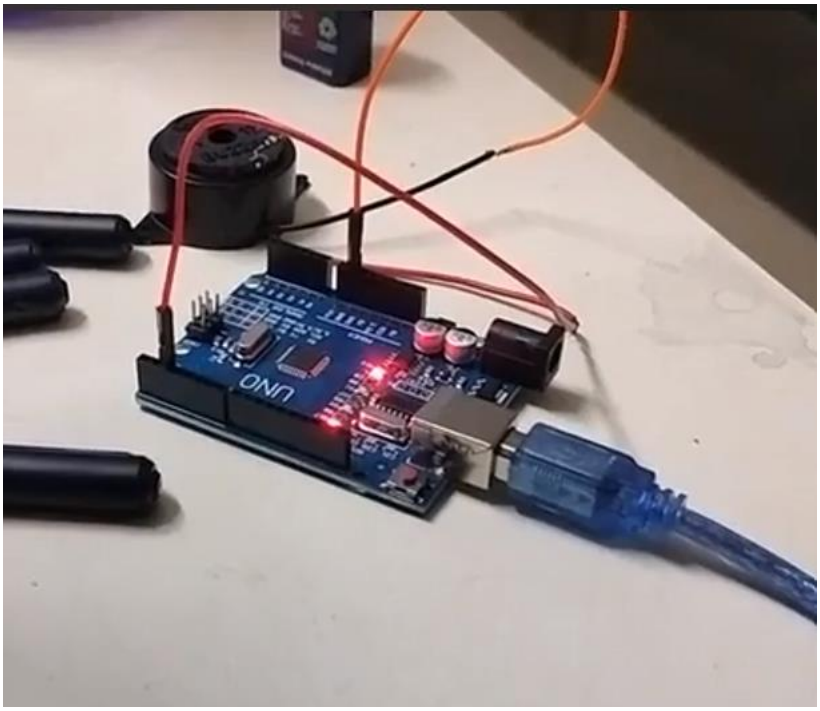
CODE

```
//declaration dump
const int buzzerPin = 2; //buzzer pin no 2 (digital)
unsigned long startMillis; //start millis
unsigned long currentMillis; //current millis
const unsigned long periodFlashDoro = 1500000; // 25 mins
const unsigned long periodEndPomodoro = 300000; // 5 mins


//function to buzz at the starting of pomodoro
void startPomodoro(){
  digitalWrite(buzzerPin,HIGH);
  delay(500);
  digitalWrite(buzzerPin,LOW);

}

//function to make the led flash to indicate the on going pomodoro timer
void flashDoro(){
  digitalWrite(LED_BUILTIN, HIGH);
```

```
  delay(500);
  digitalWrite(LED_BUILTIN, LOW);
  delay(500);
}

//flash timer during 5 mins break
void breakFlash(){
  digitalWrite(LED_BUILTIN, HIGH);
  delay(3000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(3000);
}

//buzz sound to indicate the end of pomodoro
void endPomodoro(){

  digitalWrite(buzzerPin,HIGH);
  delay(5000);
  digitalWrite(buzzerPin,LOW);
}

//average setup loop
void setup()
{
  Serial.begin(8600);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
}


//average main loop (main boy)
void loop()
{
  //start the pomodoro
  startPomodoro();
  startMillis = millis();
  while(millis() - startMillis < periodFlashDoro) {
    flashDoro();
  }
  //play at the end of pomodoro
  endPomodoro();
  startMillis = millis();
  while(millis() - startMillis < periodEndPomodoro) {
    breakFlash();
  }
```

```
}
```

PHYSICAL MODEL



SUMMARY AND CONCLUSION

This paper showed you how to build a basic Hardware Pomodoro Timer using Arduino Uno powered by ATMega328P.

In the times of overuse of digital devices and social media apps, an average human struggles to be productive in his tasks. The average screen time of a teenager is 7 hours and most of that time is spent on social media apps. Teenagers are mostly school or college students so the overuse of devices takes a toll on their productivity.

Pomodoro Technique evolved as a measure to improve productivity and reduce distractions. However, it is not easy to stay disciplined as most of the Pomodoro Timers are apps on phones and laptops which can easily distract a person from doing their work.

Hence, this Hardware Pomodoro Timer has been developed as an attempt to reduce distractibility by doing away with digital screens and any worrisome phone notifications and ensure that the user is completely focused on his work while keeping track of their Pomodoro sessions.

REFERENCES

1. Felicitas Biwer, Wisnu Wiradhany, Mirjam G.A. oude Egbrink, Anique B.H. de Bruin: Understanding effort regulation: Comparing 'Pomodoro' breaks and self-regulated breaks. 2023. British Journal of Educational Psychology. https://doi.org/10.1111/bjep.12593
2. F. Rejón-Guardia, M.A. García-Sastre, G.A. Vich-I-Martorell: the time management competence and the effectiveness of the university student. 2018. 10th International Conference on Education and New Learning Technologies. https://doi.org/10.21125/edulearn.2018.0766