Title:

Timers: Implementation of a traffic control system.

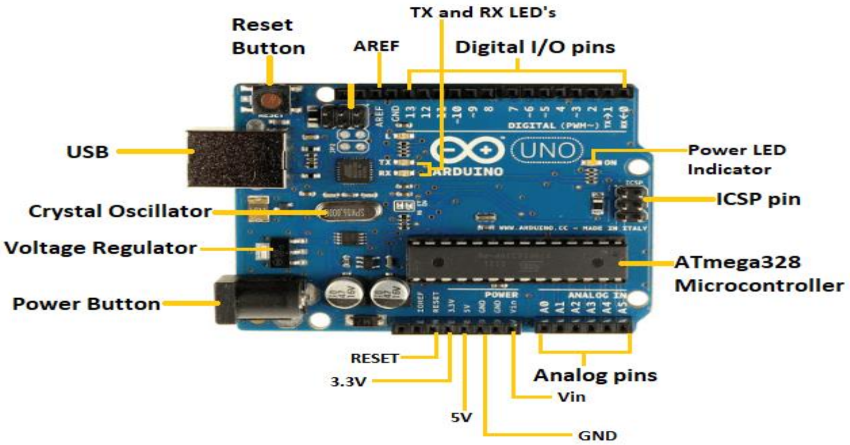
Theory and Methodology:

Timers are integral components of microprocessors, crucial for measuring and controlling time intervals. Timers play a crucial role in the realm of electronics, providing a fundamental time base that synchronizes the operations of various electronic components within a sequential logic circuit. Without a reliable time base, electronic devices would lack the coordination needed to execute specific actions at designated intervals.

A timer, essentially a register, automatically increments or decrements its value, serving as a clock-like component that measures time events. This concept is particularly pertinent in the context of microcontrollers like Arduino, where timers are integral pieces of hardware embedded in the controller.

The specific timers used in this experiment are explored, with a focus on AVR microcontrollers. AVR timers are categorized into two types: 8-bit and 16-bit timers. The distinction lies in the width of the register, where an 8-bit timer utilizes an 8-bit wide register and a 16-bit timer operates with a 16-bit wide register. This implies that an 8-bit timer can count up to 2^8, like 256 steps, ranging from 0 to 255. Similarly, a 16-bit timer extends its counting capability to 2^{16}, totalling 65536 steps from 0 to 65535.

Overview of Arduino Uno Board:



Apparatus:

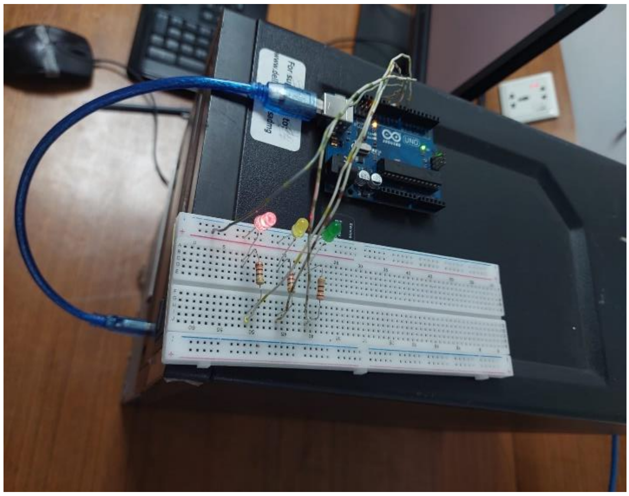
- Computer with Arduino IDE
- Arduino Uno
- LED lights (RED, GREEN, and YELLOW)
- 220 ohms resistors (Three)
- Jumper wires

Experimental Procedure:

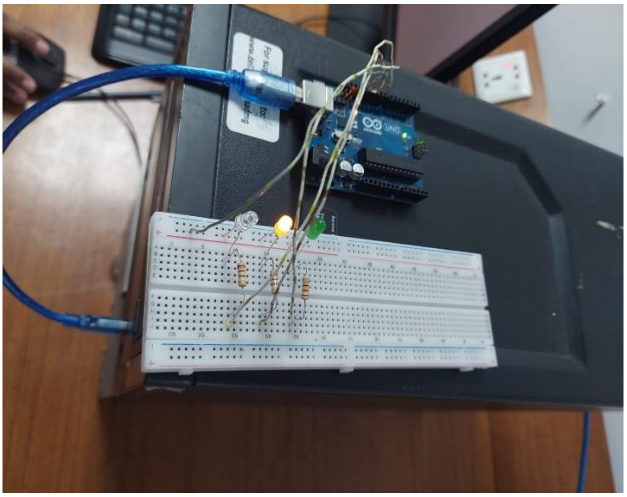
Firstly, we've understood the theory and methodology of the Arduino Uno. We've collected the essential materials, including an Arduino Uno, LEDs, resistors, and jumper wires. Divided into two parts, the first involved constructing a Red LED circuit for LED light blinking. Following the lab manual, we connected the Arduino board to the computer and wrote the code for LED blinking, subsequently uploading it. Then, we focused on the Traffic Control System, where we constructed a circuit and wrote the corresponding code in the Arduino IDE. To enhance timing accuracy, we introduced a timer pre-scaler setting, incorporating it into the delay timer function for precise synchronization. Lastly, we've burned the modified code into the Arduino board, completing the experiment. This condensed procedure aimed to provide us with hands-on timer utilization experience for the Traffic Control System.

Hardware Picture's:

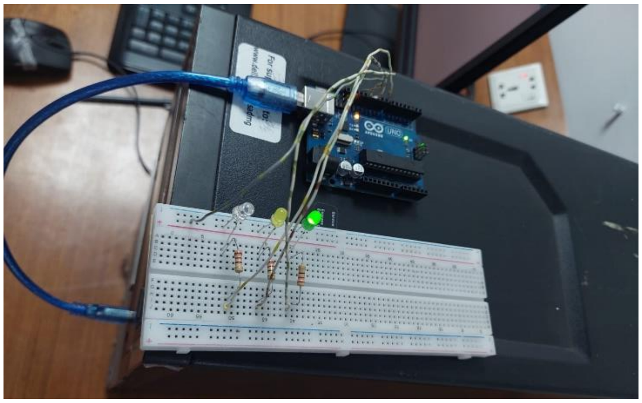
Traffic Control System



Red Led is On



Yellow Led is On



Green Light is On

Discussion:

The experiment delved into the practical use of timers in a Traffic Control System using Arduino Uno. Timers, crucial in electronics for time measurement, were explored with a focus on AVR microcontrollers, highlighting the difference between 8-bit and 16-bit timers.

Starting with a basic Red LED circuit, the experiment provided a hands-on introduction to Arduino Uno, emphasizing the importance of understanding the hardware. The subsequent phase involved creating a Traffic Control System, where the incorporation of a timer pre-scaler setting showcased the practical application of timers for precise synchronization.

The simple setup with LEDs, resistors, and jumper wires facilitated a straightforward learning experience. Burning the modified code into the Arduino board underscored the real-world application of the experiment.

Conclusion:

In conclusion, the experiment effectively demonstrated the practical use of timers in a Traffic Control System with Arduino Uno. The hands-on approach, beginning with a basic LED circuit, equipped participants with valuable skills in circuit construction, coding, and timer utilization. This condensed procedure ensures an efficient learning experience, emphasizing the significance of timers in achieving accuracy in electronic projects. The insights gained extend beyond traffic control, forming a foundation for broader applications in embedded systems and microcontroller-based projects. Overall, the experiment successfully connected theory to practical skills, enhancing understanding of timer implementation in electronic circuits.

Reference(s):

1. https://www.arduino.cc
2. ATMega328 manual
3. https://www.avrfreaks.net/forum/tut-c-newbies-guide-avr-timers
4. http://maxembedded.com/2011/06/avr-timers-timer0