Standard Embedded Diploma - Mini Project 2

Project Title:

Stopwatch with Dual Mode (Increment and Countdown) Using ATmega32 and Seven-Segment Display.

Objective:

To design and implement a digital stopwatch with two operational modes: counting up (increment mode) and counting down (countdown mode). The project utilizes an ATmega32 microcontroller, six multiplexed seven-segment displays (common anode), and multiple push buttons for user interaction.

Project Overview:

This project involves creating a versatile stopwatch that operates in two distinct modes:

1. **Increment Mode (Default):** The stopwatch counts upwards from zero, displaying hours, minutes, and seconds on six seven-segment displays.

2. **Countdown Mode:** The stopwatch counts downwards from a user-defined time, functioning as a countdown timer.

Features:

- **Reset Function:** Resets the time to zero in both modes.
- **Pause Function:** Pauses the time counting, allowing the stopwatch to hold its current value.
- **Resume Function:** Resumes counting from the paused time.
- **Mode Toggle:** Switches between increment and countdown modes.

Hardware Components:

- **Microcontroller:** ATmega32
- **Display:** Six multiplexed seven-segment displays (common anode) driven by a 7447 BCD to seven-segment decoder connected to first 4 pins in PORTC.
- **Input:**

  10 push buttons for various functions:

  - **Reset Button:**

    Connected to PD2 (INT0) with internal pull up resistor – Resets the stopwatch.

  - **Pause Button:**

    Connected to PD3 (INT1) with external pull up resistor – pauses the stopwatch.

  - **Resume Button:**

    Connected to PB2 (INT2) with internal pull up resistor – Resumes the stopwatch.

  - **Mode Toggle Button:**

    Connected to PB7 – Toggles between increment and countdown modes.

  - **Hours Adjustment:**

    - Increment Hours: Connected to PB1 with internal pull up resistor
    - Decrement Hours: Connected to PB0 with internal pull up resistor

  - **Minutes Adjustment:**

    - Increment Minutes: Connected to PB4 with internal pull up resistor
    - Decrement Minutes: Connected to PB3 with internal pull up resistor

  - **Seconds Adjustment:**

    - Increment Seconds: Connected to PB6 with internal pull up resistor
    - Decrement Seconds: Connected to PB5 with internal pull up resistor

- **Alarm/Buzzer:**

  Connected to PD0 and Activated when the countdown reaches zero or during specific alert conditions.

- **LED Indicators:**

  - Counting Up LED: Connected to PD4 (Red)
  - Counting Down LED: Connected to PD5 (yellow)

Operation Description:

- **Increment Mode:** By default, the stopwatch starts in increment mode, where the time counts up continuously from zero. The time is displayed across six seven-segment displays, showing hours, minutes, and seconds. The red LED on PD4 is turned on to indicate increment mode.

- **Countdown Mode:** When the mode toggle button is pressed, the stopwatch switches to countdown mode. In this mode, the user can set the desired countdown time using the hour, minute, and second adjustment buttons. The countdown begins once the start/resume button is pressed. A yellow LED on PD5 is turned on to indicate countdown mode. An alarm is triggered when the countdown reaches zero.

System Requirements:

1. Use **ATmega32** Microcontroller.
2. **System Frequency:** 16Mhz
3. Configure **Timer1** in ATmega32 with **CTC** mode to count the stopwatch time.
4. Connect the six 7-segments in the project using the **multiplexed technique**. You should use **one 7447 decoders** for all 7-segments and control the enable/disable for each 7-segement using a **NPN BJT** transistor connect to one of the MCU pins. Like the below image:
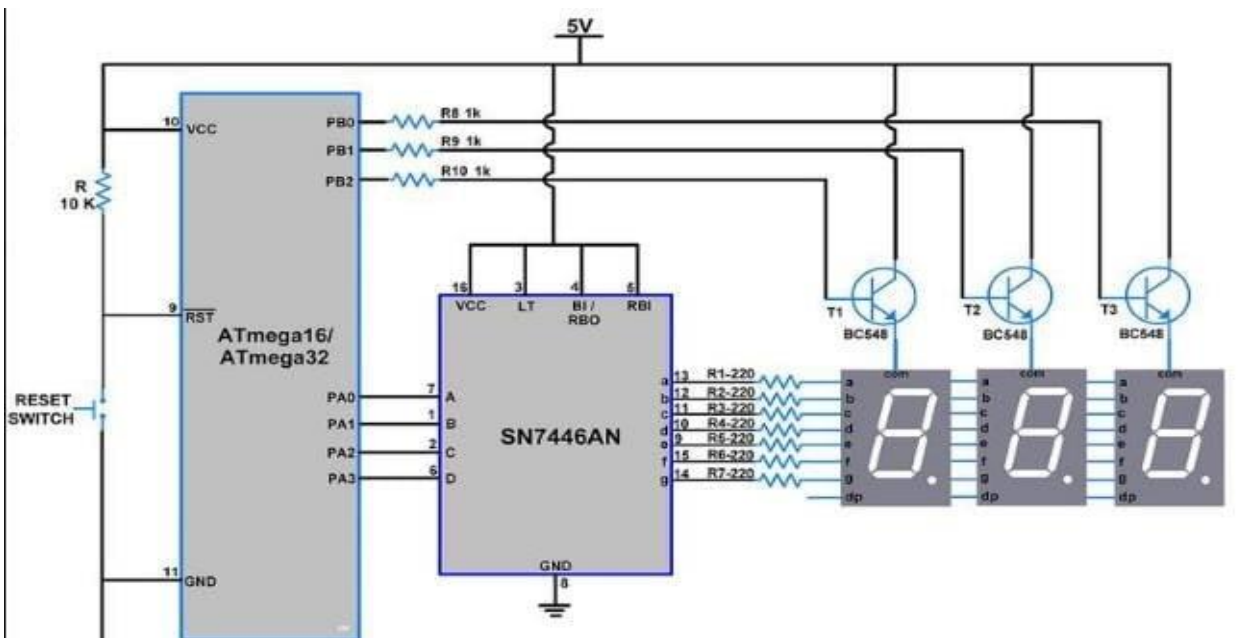
Note: The above image is just to illustrate the basic idea about the multiplexed 7-segments. Use The common anode decoder 7447 instead of the IC in the image.

5. We can connect more than one 7-segment display by using the multiplexing method. In this method, at a time one 7-segment display is driven by the Microcontroller and the rest are OFF. It keeps switching the displays using transistors. Due to the persistence of vision, it appears as a normal display. Use first 6-pins in PORTA as the enable/disable pins for the six 7-segments.

6. Stopwatch counting should start once the power is connected to the MCU.

7. Configure **External Interrupt INT0** with **falling edge**. Connect a push button with the internal pull-up resistor. If a falling edge detected the **stopwatch time should be <u>reset</u>**.

8. Configure **External Interrupt INT1** with **raising edge**. Connect a push button with the external pull-down resistor. If a raising edge detected the **stopwatch time should be <u>paused</u>**.

9. Configure **External Interrupt INT2** with **falling edge**. Connect a push button with the internal pull-up resistor. If a falling edge detected the **stopwatch time should be <u>resumed</u>**.

10. **Countdown Mode Setup:**

    To operate the stopwatch in countdown mode, follow these steps:

    a. **Pause the Timer:**

    Ensure that the timer is paused. This can be done by pressing the pause button connected to PD3 (INT1).

    b. **Toggle to Countdown Mode:**

    Use the mode toggle button connected to PB7 to switch the stopwatch from the default increment mode to countdown mode.

    c. **Adjust the Countdown Start Time:**

    Set the desired start time for the countdown using the adjustment buttons.

d. **Resume the Countdown:**

Once the desired countdown time is set, press the resume button connected to PB2 (INT2) to start the countdown.

e. **Buzzer Activation:**

When the countdown reaches zero, the buzzer connected to the alarm circuit will be triggered, alerting the user that the countdown has finished.

Check this video: https://youtu.be/-7c2NCQg5r4

Thank You

Edges For Training Team