EO
Techs

Project Documentation

"DuoSync Horologium"

Prepared By

Eng. Eslam Osama Saad

Table of Contents

(1) System Description:

(1.1) System Overview

DuoSync Horologium, a name combining "DuoSync" for dual functionality and "Horologium" for precision timekeeping, is an advanced clock designed with two distinct modes: time and date setting, and alarm setting. In the time and date setting mode, users can easily adjust the current time and date by entering the mode, setting hours, minutes, day, month, and year using increment/decrement buttons, and confirming the changes. The alarm-setting mode allows users to set an alarm by selecting the desired hour, minute, day, month, and year, activating the alarm, and ensuring it triggers a buzzer at the specified date and time. This dual-mode functionality makes DuoSync Horologium a precise and user-friendly time management device. The following Figure 1 is the implementation of our described embedded timekeeper on the simulation tool called proteus and it gives the initial look to our embedded DuoSync Horologium.

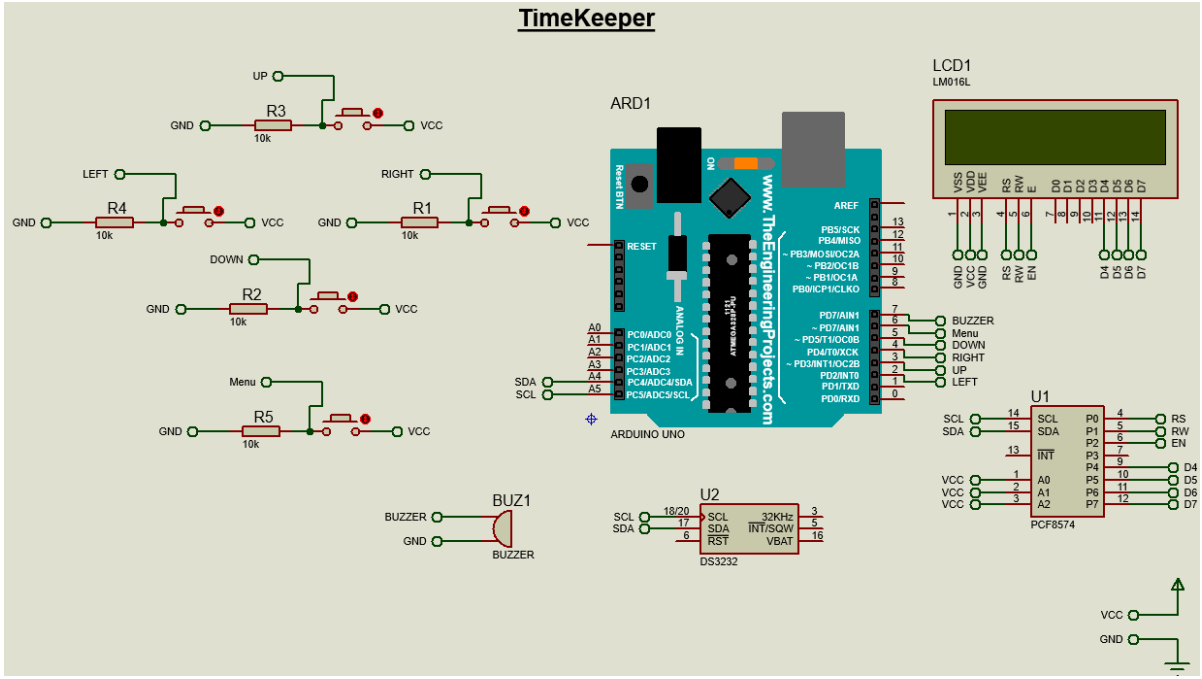

Fig.1 Proteus simulation of our DuoSync Horologium

(1.2) System Functionality

DuoSync Horologium's comprehensive functionality ensures accurate timekeeping and precise alarm settings, catering to detailed scheduling needs with ease and reliability.

1. Time and Date Setting Mode:

Entering this mode by pressing the Menu Pushbutton. The year digits will appear on 2x16 I2C LCD, allowing users to set the correct year using the increment/decrement (UP/DOWN) buttons (**Year Adjustment**). By pressing the Menu Pushbutton again first to save the change in year or press RIGHT or LEFT Pushbuttons to keep the current year and to switch to month adjustment. the same actions are going to be followed as happened in Year adjustment which is starting by showing the current date (Month or day) or current time (Hour, minute, or second). Then using UP/DOWN Pushbuttons to increase/decrease the value on the LCD, if you want to save the updated value press MENU Pushbutton or press RIGHT/LEFT Pushbutton to save the current value.

2. Alarm Setting Mode:

Entering this mode by pressing the RIGHT/LEFT Pushbuttons. The year digits will appear on 2x16 I2C LCD, allowing users to set the desired year using the increment/decrement (UP/DOWN) buttons (**Alarm Year setting**). By pressing the Menu Pushbutton again first to save the desired alarm year or press RIGHT or LEFT Pushbuttons to keep the current year as the desired alarm year and to switch to alarm month setting. the same actions are going to be followed as happened in Alarm Year setting which is starting by showing the current date (Month or day) or current time (Hour, minute, or second). Then using UP/DOWN Pushbuttons to increase/decrease the value on the LCD, if you want to save it as desired alarm value press MENU Pushbutton or press RIGHT/LEFT Pushbutton to set the current value as desired alarm value.

(2) System Design:

(2.1) Schematic Diagram

The following Figure 2 is the schematic diagram for the embedded Timekeeping device that includes several key components: an Arduino Uno, 5 Active high push buttons (Menu, Up, Down, Right, and Left), a breadboard,

DS13231 Module (RTC), Buzzer, an I2C 2x16 LCD, and various connecting wires. he diagram also features an LCD or LED display for showing time and date information, and active high push buttons for user input to set time, date, and alarm functions. Additionally, a buzzer is included for the alarm, connected to a microcontroller output pin. Pull-down resistors are used with the push buttons to ensure a defined logic level when the buttons are not pressed. The connections between these components, including power lines, ground lines, and signal lines, are clearly depicted, illustrating the flow of signals and power throughout the device.

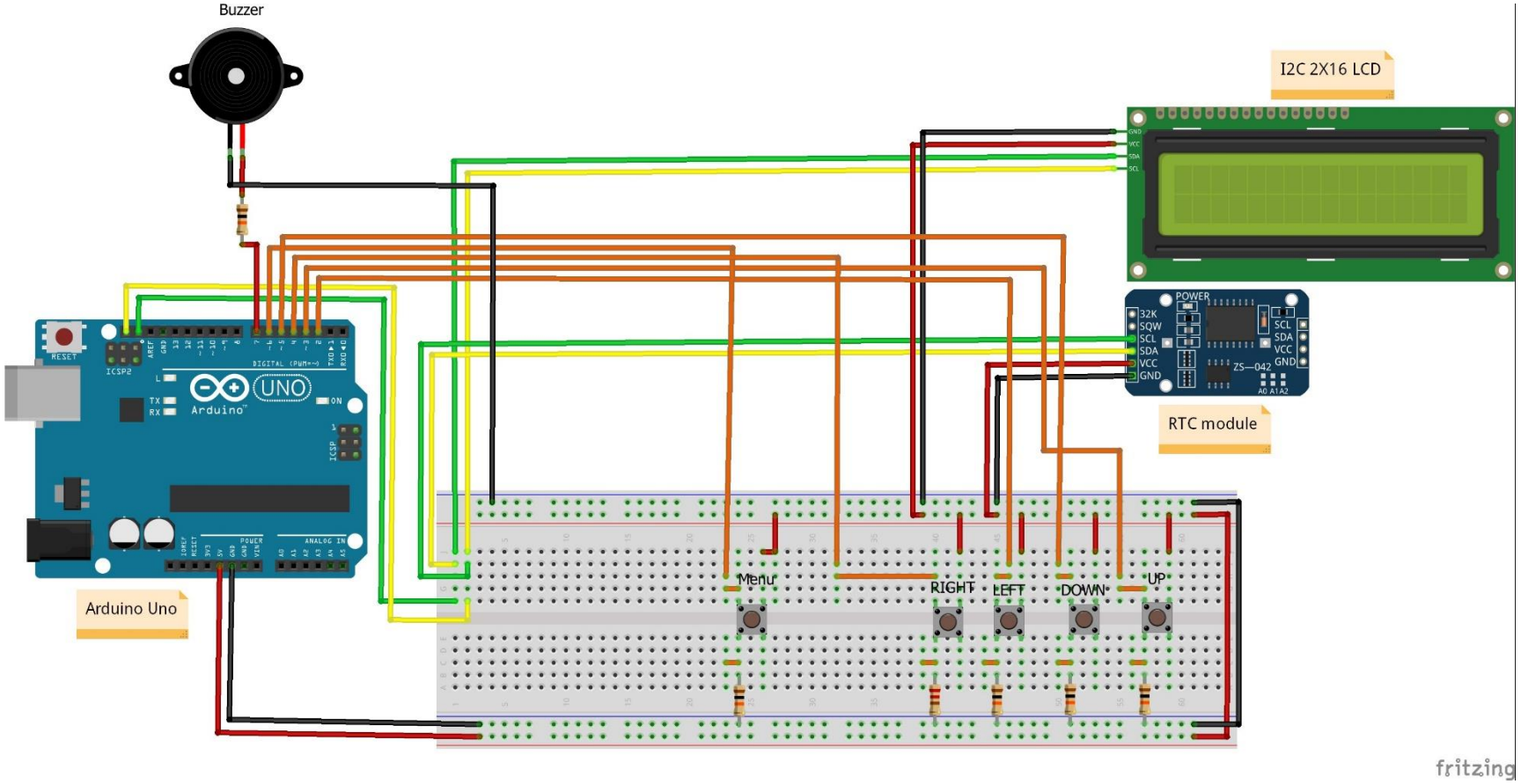

Fig. 2 Schematic Diagram of our DuoSync Horologium

(2.2) System Requirements

Components Description:

Arduino Uno: The Arduino Uno serves as the main microcontroller, responsible for managing timekeeping functions, user input, and interfacing with other components.

DS3231 RTC Module: The DS3231 RTC module provides accurate timekeeping functionality, ensuring the device maintains the correct time even when power is lost. It communicates with the Arduino Uno through the I2C (Inter-Integrated Circuit) protocol.

Breadboard: This is used to make temporary electrical connections between components.

I2C 2x16 LCD: The LCD display interface, specifically designed for I2C communication, presents time, date, and alarm information to the user. It connects to the Arduino Uno via the I2C protocol, requiring only two wires for communication (SDA and SCL).

Power Source (Adaptor): The device requires a reliable power source, such as a USB connection to a computer or a DC supply, to power the Arduino Uno and other components.

User Input Interface: Active high push buttons serve as the user input interface for setting time, date, and alarm functions. These buttons connect to specific digital input pins on the Arduino Uno.

Buzzer: An integrated buzzer is required for sounding the alarm at the specified time. It connects to a digital output pin on the Arduino Uno.

Pull-Down Resistors: Pull-down resistors are connected in parallel with each push button to ensure a stable low signal when the buttons are not pressed, preventing floating inputs.

Connections:

By establishing the following connections and utilizing the specified components, the timekeeping embedded device based on the Arduino Uno platform can provide accurate timekeeping, intuitive user interaction, and reliable alarm functionality.

- Arduino Uno to DS3231 RTC Module:
  The Arduino Uno communicates with the DS3231 RTC module via the I2C interface, connecting the SDA (data) and SCL (clock) pins of the RTC module to the corresponding pins on the Arduino Uno (A4 and A5).

- Arduino Uno to User Input Interface:
  The active high push buttons connect to specific digital input pins on the Arduino Uno, allowing user interaction for setting time, date, and alarm functions.

- Arduino Uno to Buzzer:
  A digital output pin on the Arduino Uno connects to the buzzer, activating it when the alarm triggers at the specified time.

- Pull-Down Resistors:
  Each push button connects in parallel with a pull-down resistor to ensure a stable low signal when the buttons are not pressed.

- Power Supply (Adaptor) to Arduino Uno:
  The power supply provides the necessary voltage and current to the Arduino Uno and other components, ensuring proper functionality and operation.

- Ground Connections:
  Ground connections are established between all components and the power supply to create a common reference point for electrical signals and ensure proper circuit operation.

(2.3) System Constraints

Display Constraints:

Character Limit: The I2C 2x16 LCD can only display 32 characters at a time (16 characters per line), which limits the amount of information shown to the user.

Refresh Rate: The speed at which the LCD updates may not be fast enough for very rapid inputs or complex animations.

Pushbutton Limitations:

Debouncing: Mechanical keypads can have issues with bouncing, requiring software debouncing to ensure accurate input.

Limited Inputs: With only 16 keys, the keypad can only provide a limited number of input combinations, affecting the complexity of operations that can be performed.

Software Limitations:

Make sure to not enter variable with size exceeds the max size of the variable's data type to avoid overflow. For example, we are using int data type and in Arduino IDE it contains 2 bytes which means not type value higher than $2^{(2*8)} - 1 = 65535$.

(2.4) Operating Environment

The device may be subjected to environmental factors such as temperature variations, humidity, and mechanical shocks. Components and materials should be selected to withstand these conditions and ensure reliable operation over time.

(2.5) Input & Output Formats

Inputs:

we have variables inform of int datatype to deal with increment and decrement of setting date and time or alarm date and time, but whenever we want to print these variables on LCD we need to change it to an array of chars or string.

Output:

Time and Date Display: The primary output of the project is the accurate display of current time and date information on the LCD display. Users can easily read the displayed time in hours, minutes, and seconds, along with the date, month, and year.

Alarm Functionality: The device features alarm functionality, allowing users to set alarms for specific times. When the set alarm time is reached, the buzzer activates, producing an audible alert to notify the user.

User Interaction: The output includes a user-friendly interface that enables users to interact with the device efficiently. Active high push buttons facilitate setting and adjusting the time, date, and alarm settings, enhancing user control and customization.

Visual Feedback: The LCD display provides visual feedback to users during the interaction process, confirming the input received and displaying relevant messages or prompts. This feedback ensures that users understand the actions they're performing and the current status of the device.

(3) FlowChart:

It was fully described in System Functionality section which is a subsection of System Description.

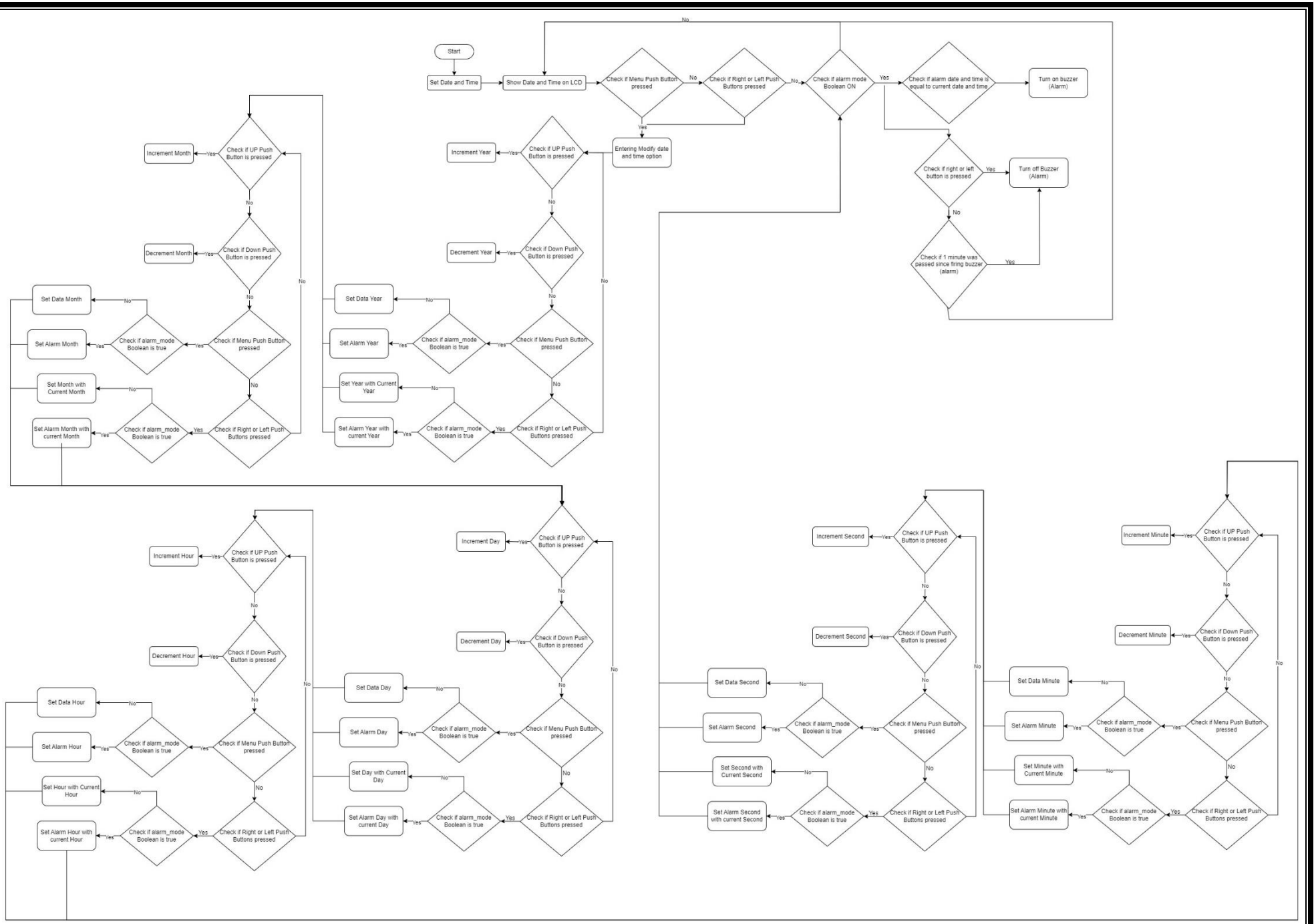

Fig. 3 Flowchart of our DuoSync Horologium