Turn on/off Radio (previous assignment)

Scope: Device Software
Level: User Goal
Primary Actor: User
Stakeholders and Interests:

- User: Wants to listen to the radio or stop listening to it.

Preconditions:

- The device is powered and the radio receiver is operational.

Postconditions:

- The radio receiver is powered on and tunes in to the last played station.

Main Success Scenario:

The user presses the button associated with powering the radio. The microcontroller receives the signal and, if the radio is off, sends power to the radio receiver. The current frequency is searched and the station tuned in to. If the radio is already powered when the button is pressed, the microcontroller cuts off power to the radio receiver.

Extensions:

- Instead of a signal produced with a button press, the signal to power the radio is given when an alarm time is reached and the radio is the alarm's chosen sound.
- The power button for the radio is pressed, but its state does not change. The connection between the button and the microcontroller, or the connection between the microcontroller to the radio receiver, has been disrupted.

Special Requirements:

- The radio is set to the position of the tuning dial and should play the station of the position as quick as possible.

Technology and Data Variations:

- The tuning dial should be available to allow the choosing of a station and provide a station for the radio receiver to pick up once turned on.
- The volume dial should be available to allow the changing of the volume of the audio. The current setting should determine the volume when the radio is powered.
- The AM/FM switch determines the bandwidths of the stations the radio has to search for. This also determines the station playing when the radio is powered.
- The radio receiver uses the data provided to it by the microcontroller to pinpoint the desired station within the selected bandwidth.

Frequency of Occurrence:

- Several times a day.

Open Issues:

- There could be faulty wiring and/or dysfunctional components.

Set Alarm (previous assignment)

Scope: Device Software
Level: User Goal
Primary Actor: User
Stakeholders and Interests:

- User: Wants to set an alarm.

Preconditions:

- The device is powered and operational.
- The alarm is set to an undesirable time.

Postconditions:

- Once the alarm time is reached by the clock, the chosen sound should be emitted through the speakers continuously until turned off or put on snooze.

Main Success Scenario:

The user presses the button that commands the microcontroller to enter one of the alarms' setting mode. On the display, the previous set time is shown with blinking LEDs. The user selects the hour or minutes by pressing their corresponding buttons. The user then uses the add and subtract buttons to increase or decrease the currently selected measure of time. Once the desired time is displayed, the user presses the chosen alarm's set mode button again to save the time to memory. The user then uses a side switch (one for each of the two alarms) to set the sound to be emitted when the alarm time is reached (buzz or radio) or to turn the alarm off. If the radio is selected, the current of position of the dial is the station played.

Extensions:

- The alarm time is not saved or the time is not correct. This could be a problem with the memory or a glitch in the display at the time of the setting.
- The time is saved, but the alarm does not go off. The volume could be set to 0 or the alarm is turned off. Otherwise, there is a disruption between the saved time and the clock, or between the alarm and the speakers.

Special Requirements:

- The default time that the alarm is set to when it is used for the first time is 12:00 A.M.

Technology and Data Variations:

- The time jumps between A.M. and P.M. when the time is added or subtracted to 12:00, more easily by the hour setting.
- Alarm times are saved in designated spaces in the clock's internal memory.

Frequency of Occurrence:

- About once a day.

Open Issues:

- There could be faulty wiring and/or dysfunctional components.
- The user could need more than two alarms.

Creating a Working LED Display

Scope: Device Software

Level: User Goal

Primary Actor: User

Stakeholders and Interests:

- User: Wants to be able to see the clock time and the alarm time.

Preconditions:

- The device must be connected to a power source through the AC plug.

Postconditions:

- The LEDs accurately displays the clock time, changing to the correct digits at each minute mark. The LEDs then accurately displays the current time to set the alarm when an alarm is being set.

Main Success Scenario:

- The device is given power and the default time is translated through the microcontroller to the prefboard of LED lights. Based on the time required, the LEDs are lit to form the individual digits. The lights turn on and off to form the appropriate numbers as the user sets the time and manages the device's alarms. Once the user is finished, the LEDs turn on and off as each minute passes to form a display of the current time produced by the clock IC.

Extensions:

- One or more of the LEDs are faulty or burnt out, producing a disfigured display of numbers.
- The LEDs are not lit. The connection between the lights and the microcontroller is disrupted or power is not reaching the LEDs.
- The display is not correctly representing of the data within the microcontroller. The data has not been aligned correctly with the pulses required to light the LEDs to produce the correct digit.

Special Requirements:

- The LEDs can change color and brightness to fit the user's preference. This is accessed by cycling through the colors and levels of brightness by pressing their corresponding buttons until the desired look is met.

Technology and Data Variations:

- The LEDs are positioned in such a way that will produce all the required digits of the required time.

Frequency of Occurrence:

- Nearly constant. Should not be working only if the device is not powered.

Open Issues:

- There could be faulty wiring and/or dysfunctional components.

Pressing Snooze

Scope: Device Software
Level: User Goal
Primary Actor: User
Stakeholders and Interests:

- User: Wants to temporarily turn off an alarm.

Preconditions:

- The device is powered.
- An alarm has been set and is currently going off.

Postconditions:

- The alarm is silenced for a count of five minutes. At that point, it sounds again until dismissed or snoozed again.

Main Success Scenario:

Once the time set in an alarm is reached on the clock time, the alarm's chosen sound is played over the clock's speakers continuously. The user hits the snooze button and the audio is stopped. A temporary alarm is set on a five minute counter with the same chosen sound. Once the counter runs out, the alarm sounds again. The user hits the snooze button again and the temporary alarm of five minutes is created again. This continues until the alarm is dismissed, the power is cut, or the alarm is reset.

Extensions:

- The snooze button is hit at any point other than during an alarm. The signal produced by the button is ignored.
- The snooze button is pressed during an alarm, but the alarm is not stopped. The button is somehow disconnected from the rest of the device.
- The snooze button is hit and the alarm is silenced, but the alarm does not sound again after five minutes. The snooze button actions may be configured to mimic the dismiss button or the dismiss button was hit instead.

Special Requirements:

- The system needs to be able to create a temporary alarm that starts from the moment the snooze button is hit and ends exactly five minutes later. The temporary alarm is immediately replaced with a new temporary alarm if the snooze button is pressed again. If the alarm is dismissed or turned off, the temporary alarm is erased and the space left for the next time the snooze button is hit during an alarm.

Technology and Data Variations:

- Memory space is saved in the device's RAM for the data of the two alarms, the clock time, and the temporary, five-minute alarm. The snooze button produces the data for the temporary alarm through the microcontroller.

Frequency of Occurrence:

- Once or twice a day.

Open Issues:

- There could be faulty wiring and/or dysfunctional components.

Pressing Dismiss

Scope: Device Software

Level: User Goal

Primary Actor: User

Stakeholders and Interests:

- User: Wants to dismiss the alarm until the next time the clock reaches the alarm time.

Preconditions:

- The device is powered.
- An alarm has been set and is currently going off.

Postconditions:

- The alarm is silenced until the next time the clock reaches the alarm time.

Main Success Scenario:

Once the time set in an alarm is reached on the clock time, the alarm's chosen sound is played over the clock's speakers continuously. The user hits the dismiss button and the audio is stopped. The alarm is canceled and is readied for the next time the clock time matches the alarm's set time.

Extensions:

- The dismiss button is hit at any point other than during an alarm. The signal produced by the button is ignored.
- The dismiss button is hit, but the alarm is not silenced. The connection between the dismiss button and the rest of the device has been disrupted.
- The dismiss button is hit and the alarm is silenced, but the alarm does not sound again on its next cycle. There is a problem with the memory or the speakers have malfunctioned. The alarm may have also been turned off.

Special Requirements:

- Once the dismiss button is hit, the alarm is rewritten to be readied for when the clock time and the alarm time matches again.

Technology and Data Variations:

- Memory space is saved in the device's RAM for the data of the two alarms, the clock time, and the temporary, five-minute alarm. When the dismiss alarm is hit, a part of the data for the alarm is changed to reset the alarm for the next cycle.

Frequency of Occurrence:

- Once a day.

Open Issues:

- There could be faulty wiring and/or dysfunctional components.