Glossary

Vision

1. **Introduction**: Having a two alarm system that doubles as an AM/FM radio fills a market need that isn't being adequately addressed at the current time.
2. **Positioning**
    a. Problem Statement: Many users need to have multiple alarms for the day, either because multiple people need their own alarms, or the user has multiple jobs, or even that the user simply has multiple events they need to be alerted for. Having multiple alarms on an alarm clock helps satisfy this need.
    b. Product Position Statement: Our dual alarm clock radio combines the flexibility of a dual alarm system with the convenience of having radio in a style that appeals to the user's sense of taste while being incredibly easy to use
3. **Stakeholder and User Description**
    a. Stakeholder Summary: Anyone who either needs to be on time for multiple event, or needs someone else to be on time for a certain event
    b. User Summary: The average customer will likely represent a person who is employed or sticks to a schedule that requires they show up to a place at a certain time.
    c. Key High-Level Goals and Problems of the Stakeholders: Being on time to an event, having someone else be on time for an event, and a desire to be able to use the radio, be it a song or talk show, as either an alarm or a source of entertainment in the morning.
    d. User-Level Goals: Being alerted at a certain time, and the ability to listen to radio
    e. User Environment: In the bedroom
4. **Product Overview**:
    a. Product Perspective: The product is relatively independent and self-contained, but it does rely on there being active radio stations in order to use some of its functionality.
    b. Assumptions and Dependencies: The alarm will only sound at the correct time if the clock itself is set to a correct time. Additionally, it needs to be constantly supplied with power, or else it can't continue keeping track of the time, and the clock will be off when it's plugged back in. Finally, the radio functionality requires active radio stations in order to play anything other than static.
5. **Product Features**
    a. Displays the time

b. Allows user to set time
c. Sounds an alarm at a specified time
d. Allows user to set alarm time
e. Plays radio
f. Allows the user to switch between AM radio and FM radio
g. Allows user to navigate through different AM/FM stations
h. Allows the user to connect the radio to the alarm, and play a station at a specified time, allows the user to set the time.
i. Snooze button

Supplemental specifications

Purpose

To create a fully functioning dual alarm clock that has am and fm capabilities.

Usability

1. Set time button
2. Set alarm button
3. Change frequency switch
4. Display for time and frequency

Reliability

The dual alarm clock will go off at the set time as long as it has been set.

Performance

1. Dual Alarms
    - **a.** The user will be able to set two alarms to go off at different times of the day
    - **b.** The user will be able to hit snooze and stop the alarm for a set time
2. Time
    - **a.** As long as the clock is set appropriately it will display the time
3. Radio
    - **a.** The clock can play am and fm stations

Supportability

Due to no internet connectivity updating the software isn’t possible

Design constraint

1. lack of internet connection meant the clock can’t auto adjust the time.
2. Done in Java

Helpful Definitions

(UI)= User interface
(Am)= Amplitude Modulation
(Fm)= Frequency Modulation
Hardware= physical Components of the clock radio
Stakeholders= everyone directly involved in the product or action.
Primary actor= who is doing the action/ being affected by it
Trigger = what causes the action or event to take place.

Use case 1

Title: Changing the wavelength
Primary actor: User
Goal/Context: Change from Am to Fm frequency
Scope: Clock radio software
Level: Hardware
Stakeholders: the user, everyone listening, broadcaster
Pre-Condition: The desire to change from the lousy Am stations to the Fm stations
Minimal Guarantee: That the radio will change from am to fm
Success Guarantee: You are successful if you change frequencies.
Trigger: the switch that changes between am and fm
Main success scenario: The radio changes from am to fm
Extensions: 1. The switch is broken and doesn't switch the frequencies
Technology and data variations list: varying methods to switch frequencies depending on make.

Use case 2

Title: Setting the alarms
Primary actor: user
Goal/context: make alarm sound at set time
Scope: clock radio software
Level: hardware
Stakeholders: user to be woken up, anyone depending on said user being up on time
Pre-Condition: Both alarms must be set the night before
Minimal Guarantee: the alarms will go off
Success Guarantee: the alarm goes off and wakes up the user
Trigger: the arrival of the set time
Main Success Scenario: the alarm goes off and wakes up the user
Extensions: 1. The alarm doesn't get saved
2. the user changes the time to a new desired time
Technology and data variations list: varying UI's that could affect setting the wake up time.

Use Case 3

Title: Snoozing the day away
Primary actor: The user/ person being woken up but wishes to go back to sleep.
Goal/Context: to pause the alarm for 5 minutes after the user hits snooze.
Scope: Clock radio software
level: hardware.
Stakeholders: The person who has been woken up but wants to go back to sleep
Pre-condition: The alarm must be going off and the user wants to go back to sleep.
Minimal Guarantee: That the alarm will stop going off once snooze is pushed.
Success Guarantee: the user hits snooze once their woken up and the alarm stops for 5 minutes.
Trigger:snooze being hit by the user.
Main Success scenario: the user hits snooze once their woken up and the alarm stops for 5 minutes.
Extensions: 1. Snooze fails to stop the alarm.
2. The alarm doesn’t go off again after 5 minutes
Technology and data variations list: none

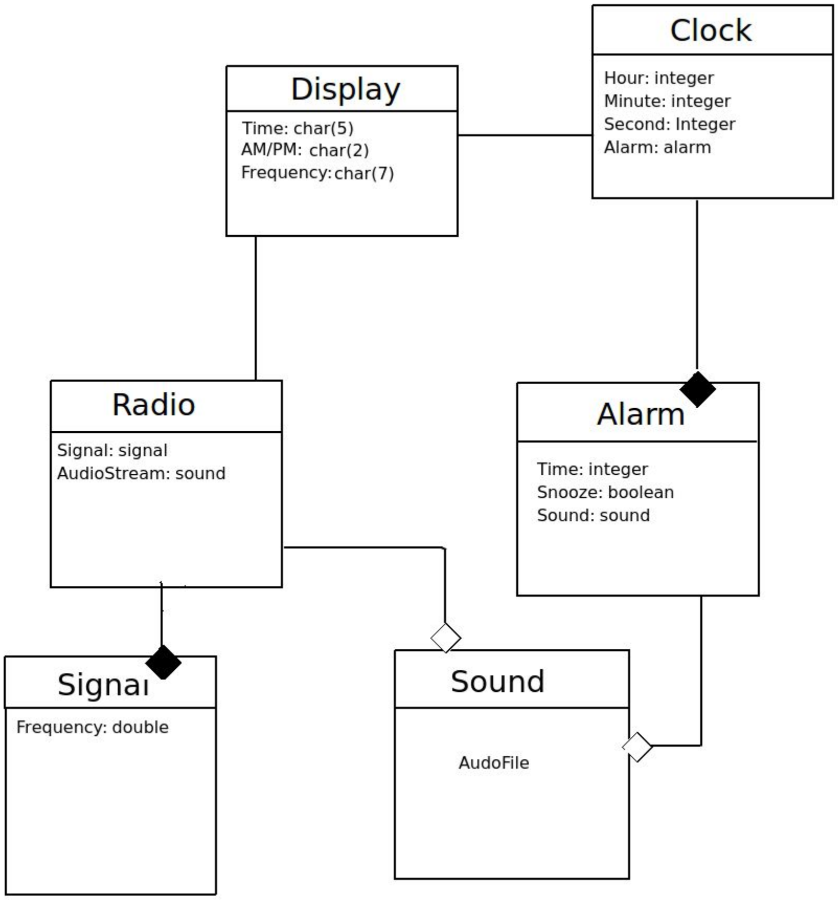
Use Case 4

Title: Turning off the alarm
Primary actor: The user who wants to turn off a previously set alarm.
Goal/Context: to turn off a previously set alarm.
Scope: Clock radio software.
Level: Hardware.
Stakeholders: the user who doesn't want to be woken up.
Pre-condition: The user must want to sleep in past the usual alarm time.
Minimal Guarantee: The alarm that was turned off wont go off.
Success Guarantee: After being turned off the alarm will not go off.
Trigger: the user turning off the alarm. via the alarm off/on button.
Main success scenario: The user is not woken up by the turned off alarm.
Extensions: 1. The alarm still goes off
2. The alarm can’t be set again.
Technology and data variations list: Possible different switches to turn off the alarm.

Use Case 5

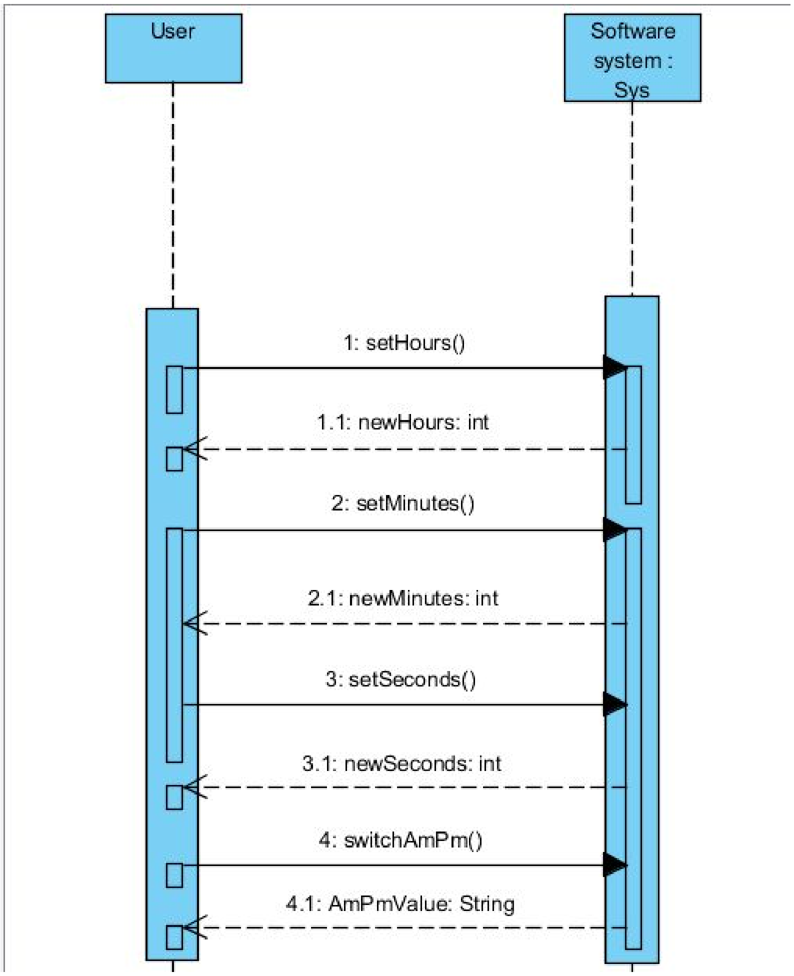
Title: Changing the time
Primary actor: The owner of the clock who wants to change the time.
Goal/Context: to change the time from the current displayed time to the actual time.
Scope: Clock radio software
Level: Hardware.
Stakeholders: Anyone who uses the clock to tell the time throughout the day.
Pre-condition: The time on the clock is different than what the user wants it to display.
Minimal Guarantee: After the user changes the time, the new correct time will be displayed by the clock.
Success Guarantee: The clock will display the correct time after being changed.
Trigger: The user holding the change time button.
Main Success Scenario: the new correct time is displayed and stays correct.
Extensions: 1. The clock fails to display the correct time after being changed.
Technology and data variations: Possible variations in change time button.

Domain Model

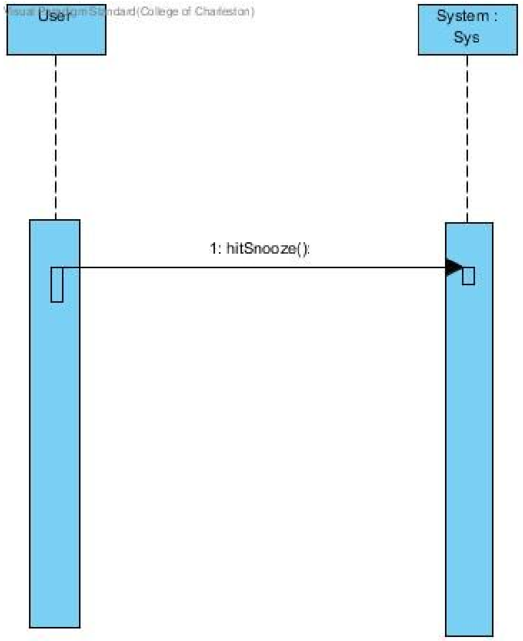


Sequence Diagrams

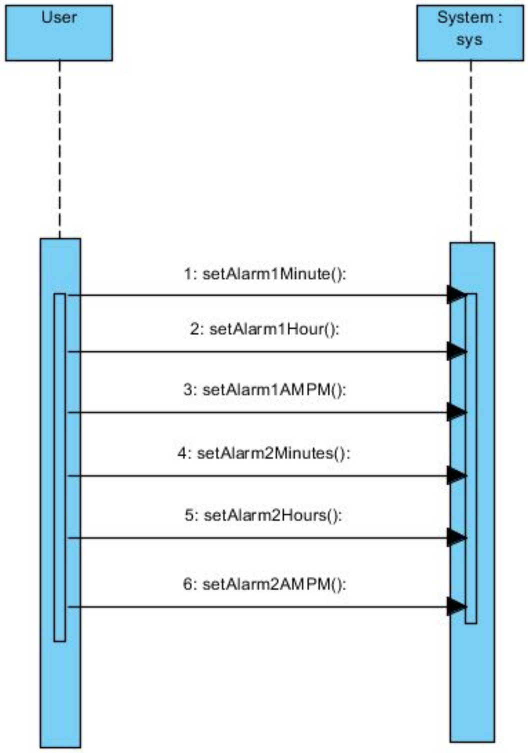
Changing the time


User
Software system : Sys
1: setHours()
1.1: newHours: int
2: setMinutes()
2.1: newMinutes: int
3: setSeconds()
3.1: newSeconds: int
4: switchAmPm()
4.1: AmPmValue: String

snooze



Change alarms



Operation Contracts:

Name: SetAlarm(alarmTime: Time)

Responsibility: Sets the time that the alarm goes off

Cross References: Use Case: “Setting the Alarm”, “User turns alarm off”

Exceptions: The alarmTime given is not a valid Time

Preconditions: clockTime is already set

Postconditions: The alarm will be set to go off at the specified alarmTime.

Name: ChangeWavelength(waveLength: WaveLength)

Responsibility: Tunes the radio to the given wavelength

Cross References: Use Cases: Changing the Wavelength

Exceptions: The wavelength given is not valid

Preconditions: The user tries to set the wavelength that the radio can receive.

Postconditions: The radio is tuned to the given wavelength

Name: Snooze()

Responsibility: Turns off the alarm for 10 minutes, then activates is again

Cross References: Use Cases: Snoozing the Day Away

Exceptions: none

Preconditions: The alarm is already going off

Postconditions: The alarm is turned off and the alarmTime is set for 10 minutes later

----

Name: setClock(clockTime: Time)

Responsibility: Sets clockClockto a different specified time

Cross References: Use Cases: Changing the Time

Exceptions: none

Preconditions: clockClock is set or needs to be changed

Postconditions: The new set clockClock is displayed

Name: alarmOff()

Responsibility: Turns off the alarm

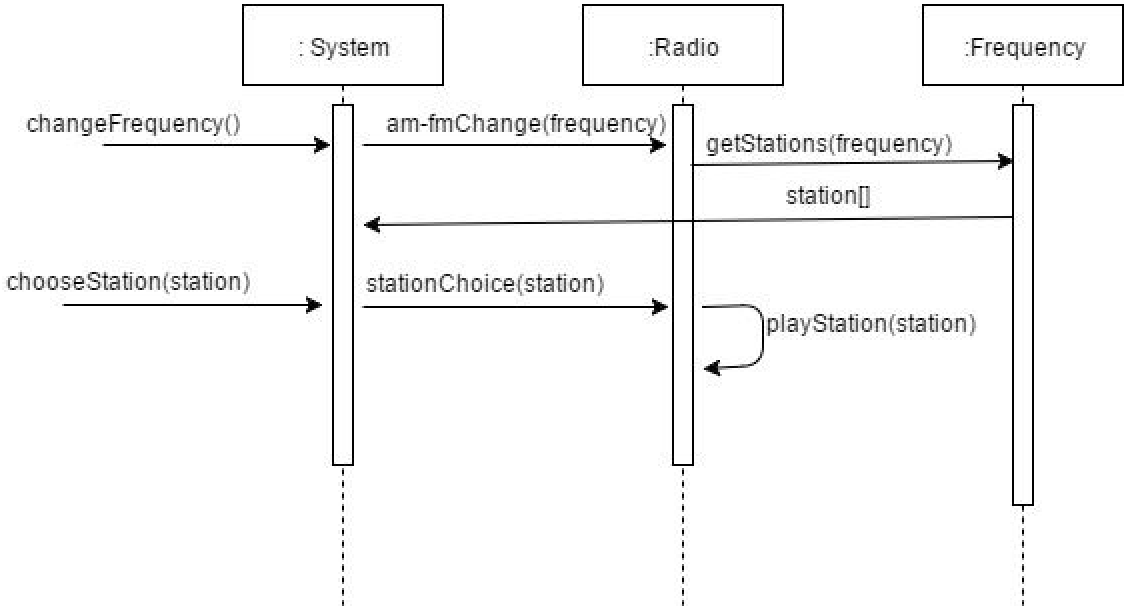
Cross References: Use Cases: Turning off the alarm

Exceptions: no alarm set

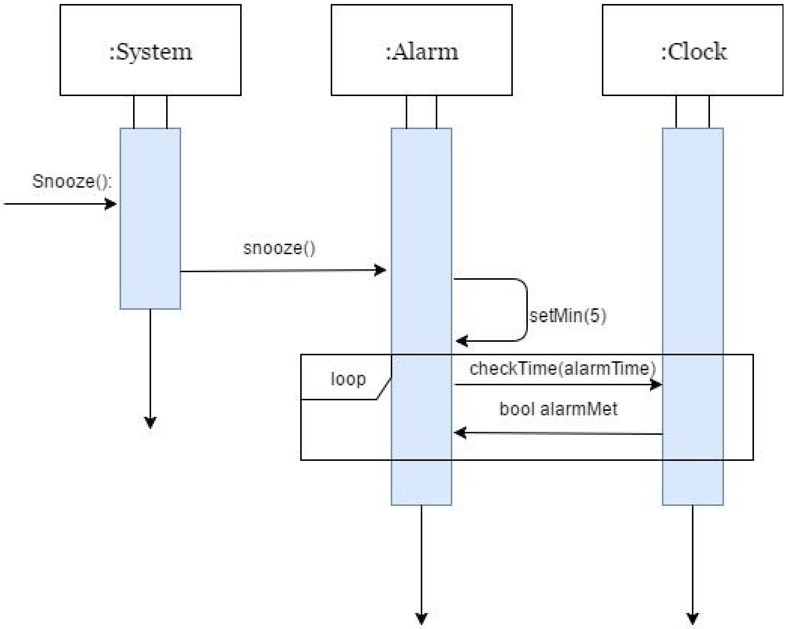
Preconditions: An alarm is set or the alarm is currently sounding

Postconditions: The alarm will not sound or will stop sounding

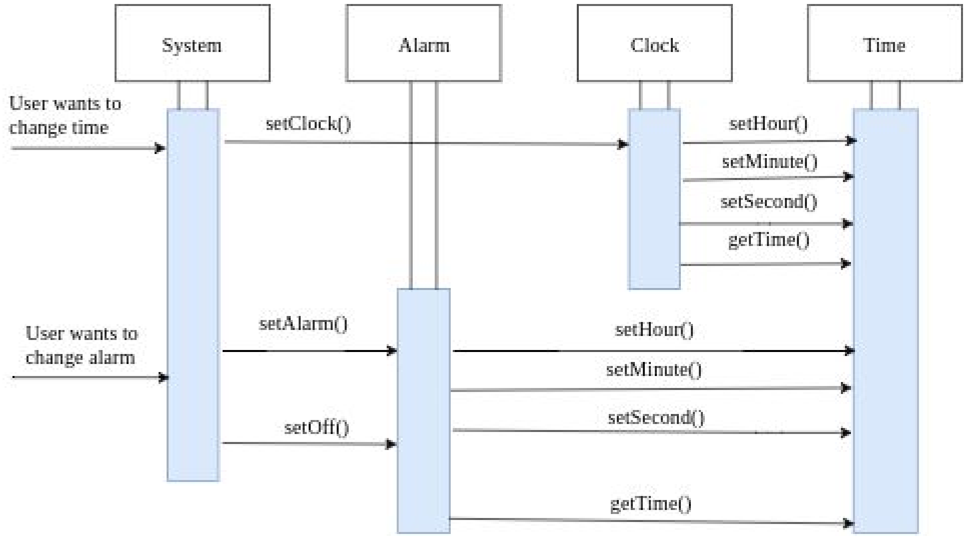
Change Wavelength



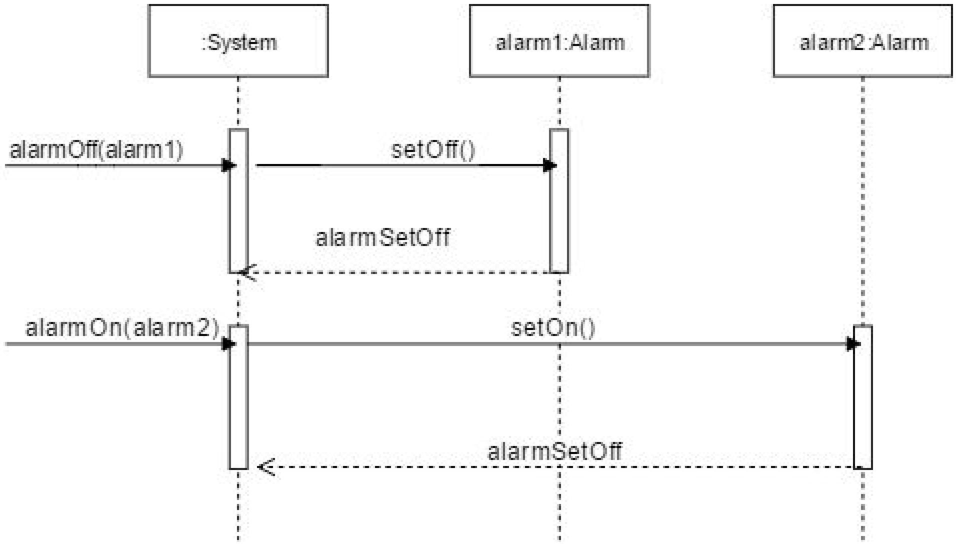
snooze



setAlarm, setClock

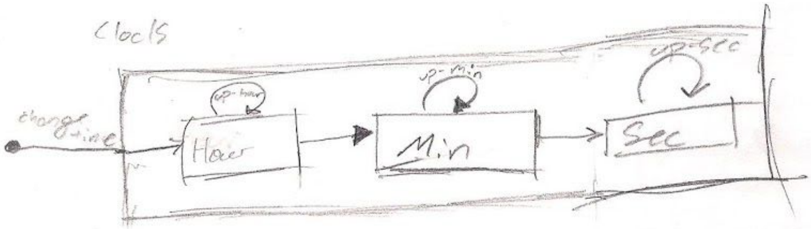


Turn off Alarm

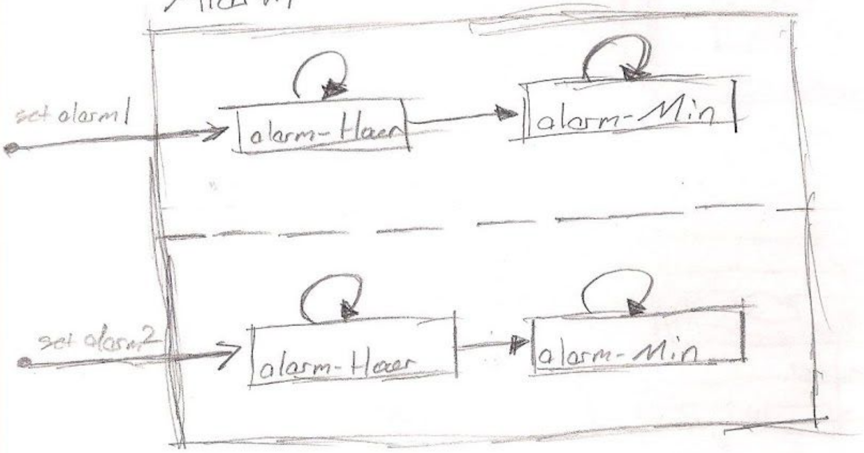


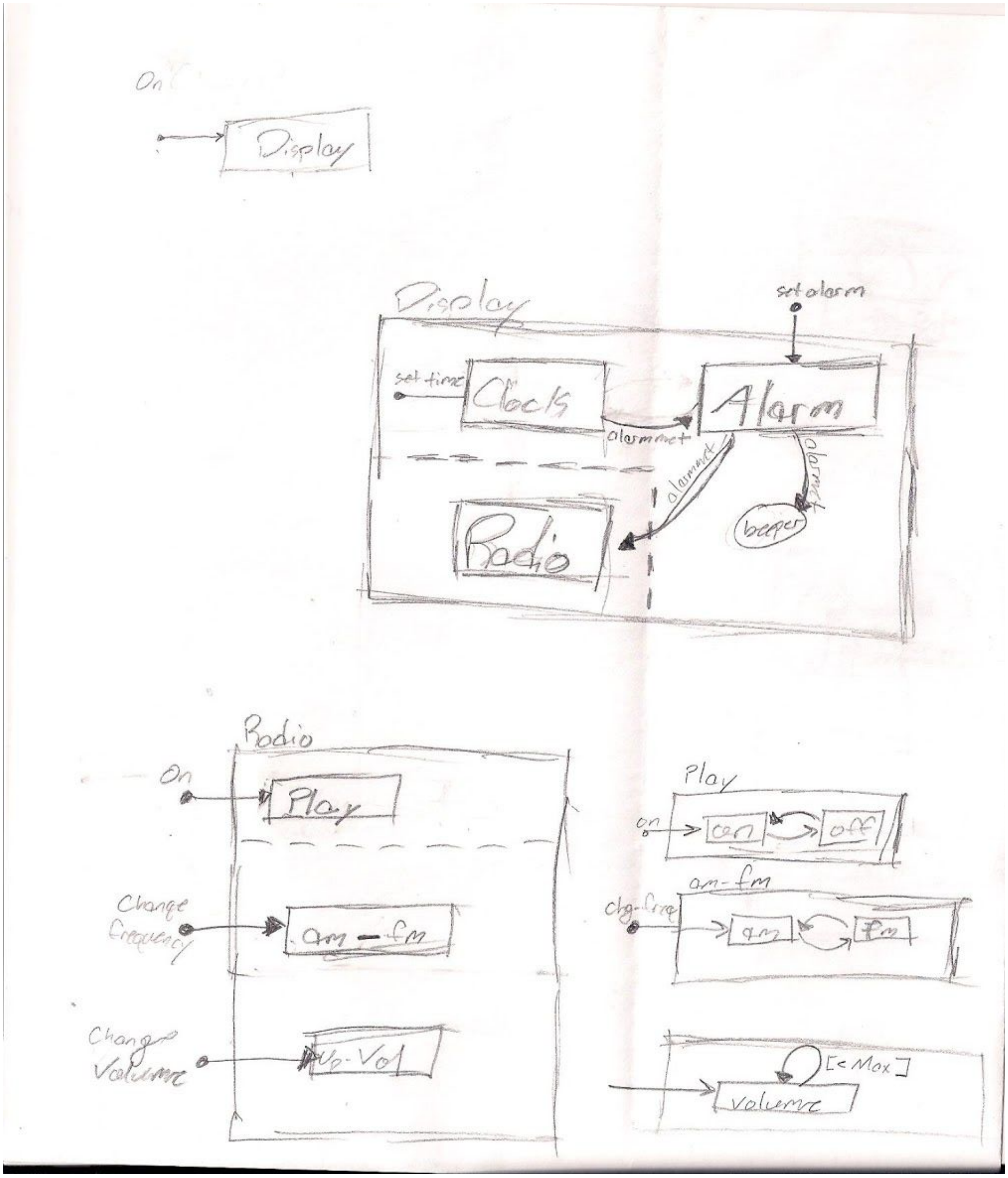
State Diagrams

Clock


change time
up-hour
up-min
up-sec
Hour
Min
Sec

Alarm


set alarm1
alarm-Hour
alarm-Min
set alarm2
alarm-Hour
alarm-Min


On
Display
Display
set alarm
set time
Clocks
Alarm
alarm met
alarm met
alarm met
beeper
Radio
Radio
On
Play
Change Frequency
am — fm
Change Volume
Up-Vol
Play
on
on
off
am-fm
chg-freq
am
fm
[< Max]
volume