Time is all around us. Knowing what time it is allows us to organize our lives into manageable chunks. Perception of the passage of time is unique to each species, culture, and individual, and changes under different situations and biological mechanisms. The purpose of this project is to explore these notions of time and to design representations of time to inform us of the present moment, the past, and to set notifications for the future.

In this project your team will start by exploring the individual components of an alarm clock, which will then be assembled into a fully functional alarm clock. The clock should display the current time, allow the user to set the current time, allow the user to set an alarm time, allows the user to enable the alarm time, and allow the user to turn off the alarm when it goes off. The device should use a speaker of piezo-electric buzzer for the alarm and for feedback sounds as appropriate. It should employ an 8x8 matrix display to represent the current time, feedback during setting the alarm time, and a counterpart to the audio alarm. The recommended display is a single color 8x8 led matrix display with the MAX7219 driver. Multi-color or RGB LEDs may be used if they can be justified and utilized to enhance the interactive experience. The device can have up to 3 switches or knobs as inputs. Switches could be a “simple” as momentary push buttons, but may also include other sensors which can be triggered as switches. The device should interface with a battery powered RTC to keep the current time even when the microcontroller is powered off. The recommended RTC is the DS1307, but other more accurate ones may be substituted, or you may choose to connect to an atomic clock through a WWVB receiver.

**STEP ONE (THE MAXTRIX) 02/12/2015 + 02/17/2015**
In the first step, you should use your 8x8 matrix to display a pre-stored animation, scrolling fonts, geometries, or a mathematically based pattern. The main objective of this exercise is to learn to use your 8x8 LED matrix.
- Adafruit-GFX Library: https://github.com/adafruit/Adafruit-GFX-Library
- 8x8 Animation Tool: http://www.diyode.com/mini8x8/

**Deliverables:**
- Video or animated GIF or sequence of photos that demonstrate your working LED display with graphics. Posted to the class blog (per team).

**STEP TWO (STORING TIME) 02/19/2015**
In the second step, you should understand how the microcontroller stores time in a LONG variable in milliseconds and display this in the serial monitor. Then you should connect to your RTC using the Inter-Integrated Circuit (I²C) protocol (assuming your RTC uses I²C and not SPI or another communication protocol) to set the time on the RTC and then to retrieve that time after the microcontroller has been power cycled (i.e. powered off and then powered back on) and print to the serial monitor. You should then use the time alarm library to set an alarm and connect a single LED to blink 10 times once the alarm is activated.
- Time Library: https://www.pjrc.com/teensy/td_libs_Time.html
- TimeAlarms Library: https://www.pjrc.com/teensy/td_libs_TimeAlarms.html
- DS1307RTC Library: http://www.pjrc.com/teensy/td_libs_DS1307RTC.html

**Deliverables:**
- In class demonstration that your time and alarm code work

**STEP THREE (BUTTONS) 02/24/2015 + 02/26/2015**
In the third step, you should learn how to setup a button. You will learn to perform debouncing through hardware and through software. You will learn the difference between interrupt based code and polling based code. You will also learn to recognize typical button press events. You should watch the materials on debouncing. You will create a Fritzing Breadboard sketch of a debounced circuit for 4 buttons. You will also read up on how to recognize a single click, double click, and press/hold button event. You will use this knowledge with two (or more) input buttons to
create combinations of single click, double click, waits, and press/hold events to control a set of
LEDs. You will create a state diagram to document how your buttons work with each other.
- Video on Hardware Debouncing: https://www.youtube.com/watch?v=tmjuLtiAsc0
- One Button Library: http://www.mathertel.de/Arduino/OneButtonLibrary.aspx
- Finite State Machine diagram: http://madebyevan.com/fsm/

Deliverables:
- Fritzing Breadboard sketch file of your hardware debounced circuit (per individual).
- Video of your circuit demonstrating the recognition of a single click, double click, and
  press/hold from at least 2 buttons. The reaction should be to light up different LEDs under
different conditions. Linked/posted to class blog (per team).
- A PNG of your state diagram that represents how your buttons work together posted to
  blog (per team).

STEP FOUR (KNOBS and TONES) 03/03/2015 + 03/05/2015
In the fourth step, you will explore the use of rotational knob inputs along with making sounds. You
should first construct a basic circuit using your piezo speaker and the NPN transistor to power it.
You should use the tone library to play stored melodies/songs. Then you should attach your
Rotary Potentiometer to the circuit to control the volume. You may do this either through analog or
digital methods. You will also attach your Rotary Encoder to your circuit to change songs. You will
then make a video of your device cycling through songs as you change the rotary encoder and
changing volume levels as you change the Rotary Potentiometer.

Deliverables:
- PNG of your circuit on the table, uploaded to the class blog (per team)
- Video of your device playing different melodies as you change the Rotary Encoder and
  changing the volume levels as you change the Rotary Potentiometer. Link/post to class
  blog (per team)

STEP FIVE (FINAL STEP)
You will design an alarm clock that represents the current time, allows users to set the time, allows
users to set the alarm time, allows users to activate and deactivate the alarm, and provides
feedback when the alarm activates. You may use up to 3 switches or knobs for input. You may use
up to 80 LEDs.

Deliverables:
- The alarm clock
- A 5-minute in-class team presentation of your alarm clock concept, device, and demo
- An 11x17 print-out documenting your visualization concept, use for the presentation
- The 11x17 print-out as a PNG file linked/posted to the class blog
- 2-5 Images/Videos with descriptions of the design process linked/posted on the class blog
- 2-5 Images/Videos with descriptions of your prototype linked/posted on the class blog

The in-class team presentations are scheduled for Thursday, March 12th, 2015.