# Assignment 7 – Alarm Clock

[*Submit on Blackboard*](https://blackboard.usc.edu/)

## Goals

* Connect speaker and play tones
* Connect device to Particle cloud
* Connect to IFTTT and control device

Overview

This assignment is to create an alarm clock that has a speaker that can play a song or a buzzer sound as well as a snooze button. The alarm set time and whether it is enabled / disabled will be controlled from a mobile device via IFTTT.

Components

* Argon
* Breadboard
* 1 x speaker
* 1 x TMP36
* 1 x LED
* 2 x push buttons (plus any necessary resistors)
* Jumper wire (standard male-male)

Requirements

* Create a Fritzing breadboard prototype layout of your design. Once you’re satisfied with the design, connect the device
* It is recommended to code the device in stages, and test at each stage

**Global variables**

* You can create other variables as needed, but these are required
* **int melodyNotes[16]** and **int melodyNotesDuration[16]** to store the pattern of notes for the melodies your alarm will play

**Stage 1: Speaker and Tones**

* Connect the speaker, LED, and push button (push button discussed later)
* Create three different “songs” for the alarm. One can be a simple alarm buzzer, but the other two should have distinct melodies (at least 16 notes). You can use the list of musical notes and frequencies (or do something different).
* Test the system playing the notes

**Stage 2: Configure Alarm Settings via IFTTT**

* Create an IFTTT applet that allows a button press to enable / disable the alarm on the argon. This means if the alarm time arrives but the alarm status is disabled, then the alarm will NOT sound
  + Hint: Using the IFTTT **Button widget** trigger
* Create a IFTTT applet that allows you to choose between the song or the buzzer sound that will play on the argon when the alarm goes off
  + Hint: Using the IFTTT **Button widget** trigger

**Stage 3: Set Alarm Time via IFTTT**

* Create an IFTTT applet to set a time and start an alarm on the argon.
  + Hint: Using the IFTTT **Date & Time** trigger, you can create a **Particle** function action)
* IFTTT only allows time triggers at :00, :15:, :30, and :45 minutes each hour. It is recommended (not required) to create another **Button** **widget** trigger that you can use for testing

**Stage 4: Configure timing and buttons (Stop and Snooze)**

* The logic of alarm is as follows (note these are not meant to necessary be exact coding, just a general flow)
  + If alarm is disabled
    - Snooze button, stop button, and alarm time trigger have no effect
  + If alarm is enabled
    - If alarm time trigger goes off
      * The selected song will play indefinitely until the stop button is pressed
      * If the snooze button is pressed, the song will be paused for a short time (probably 10 min, but for testing you can use a few seconds)
* Hint: In order to manage the timing of events, use **millis()** instead of **delay()**. Note that a delay of few milliseconds between individual notes is fine

Extra Credit

* IFTTT does not allow for arbitrary alarm times. Use the Particle console (or app) to set an arbitrary alarm time (e.g. 4:58 pm).

Required naming convention (replace # with the current assignment number)

* **Project Name** 
  + itp348\_a#\_lastname\_firstname
* **Zip File** (include entire project folder)
  + itp348\_a#\_lastname\_firstname.zip

## Deliverables

1. A compressed file containing your project. Follow the guidelines for full credit.

Here are the instructions for submission

1. Navigate to your project folder.
2. Include the *entire* folderin a zip file
3. Rename the zip file based on naming convention
4. Upload zip file to Blackboard site for our course
5. A photograph of your device connected to USB with the blue light on.
6. A (very) short video demonstrating your project functioning

## Grading

|  |  |
| --- | --- |
| Item | Points |
| Fritzing layout | 5 |
| Stage 1: Speaker and Tones | 10 |
| Stage 2: Configure Alarm Settings via IFTTT | 5 |
| Stage 3: Set Alarm Time via IFTTT | 5 |
| Stage 4: Configure timing and buttons | 15 |
| Total | 40 |