Assignment 3

Look, Up in the Sky

Due Date: Monday, March 27, 2017 @ 23:55

ECE 4564 - Network Application Design

Learning Objectives

RESTful Interface

- Python Requests Library

Service and Data API’s

Event Notification

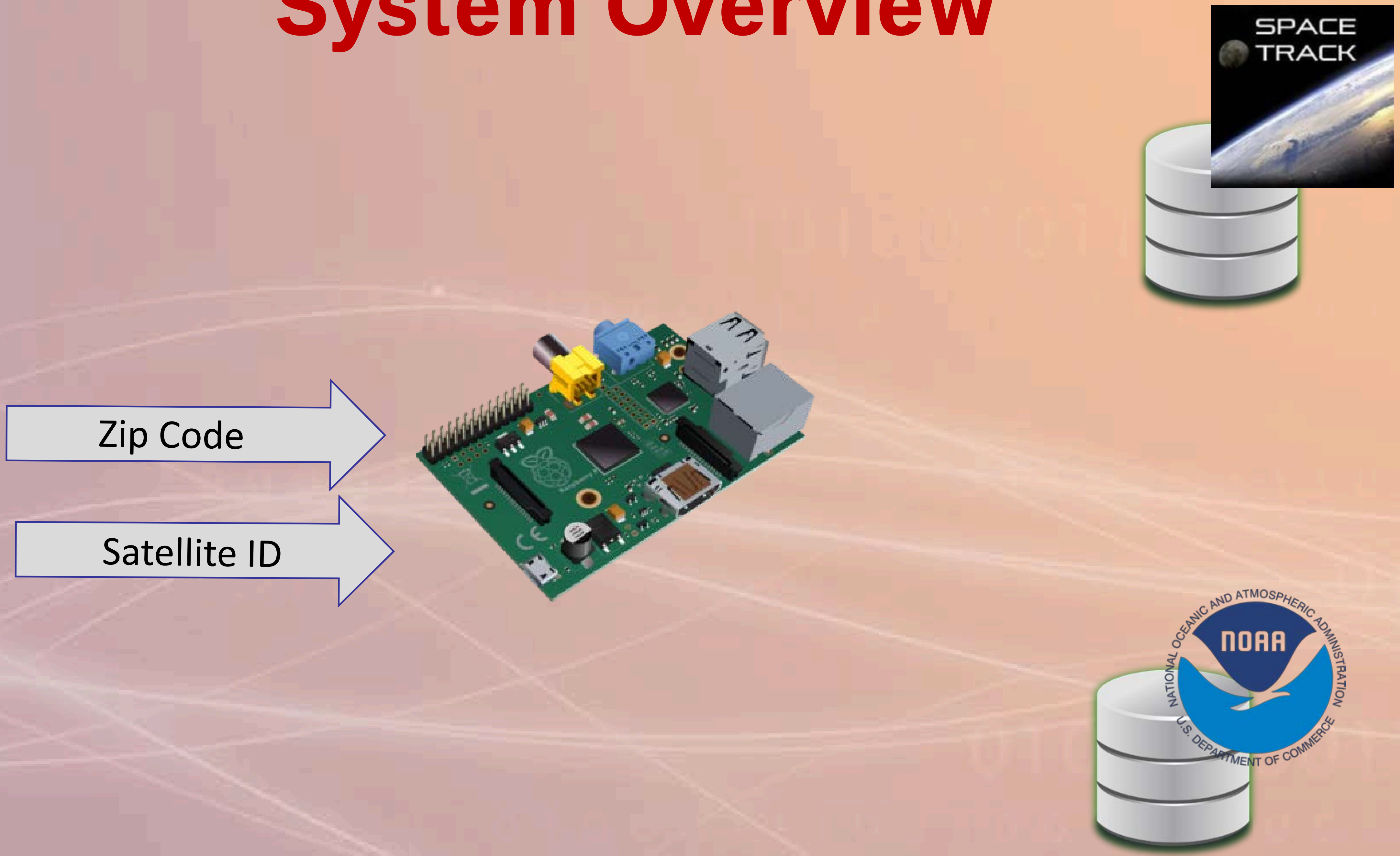
Raspberry Pi GPIO

Overview

Artificial Satellite Monitor Gateway that will query Space-Track and NOAA API

- Raspberry Pi functions as an Event Gateway
- Gateway receives zipcode and satellite identifier as input parameters
- Gateway makes RESTful queries to Space-Track API for required satellite data (TLE)
- Gateway makes RESTful queries to NOAA API for weather conditions in zipcode area on satellite sighting date/times
- Calculate satellite visibility date/times in zipcode area
- Gateway event notification:
  - flashes LED via GPIO
  - Generates sound
  - Sends SMS txt message

System Overview

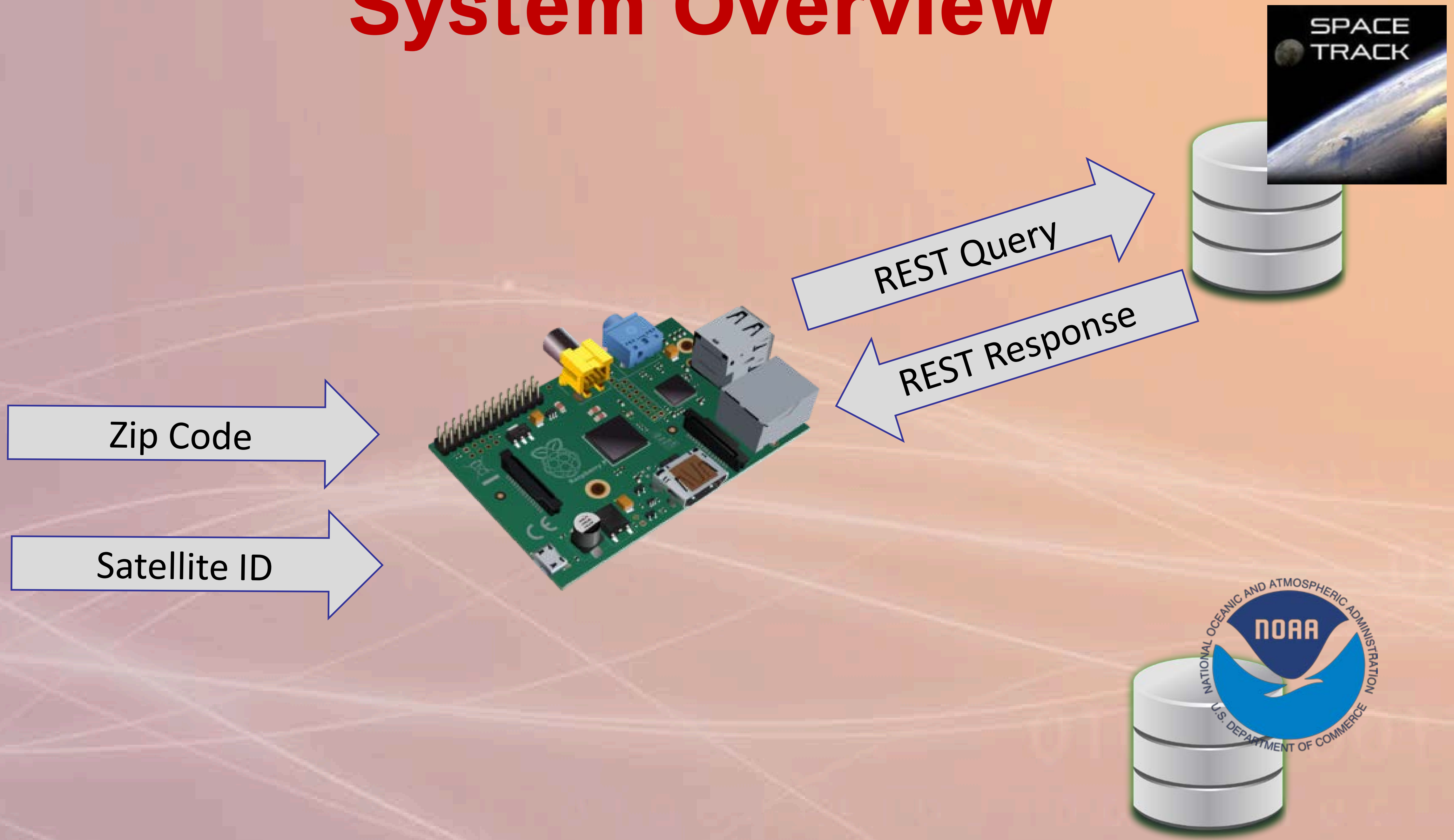


Zip Code

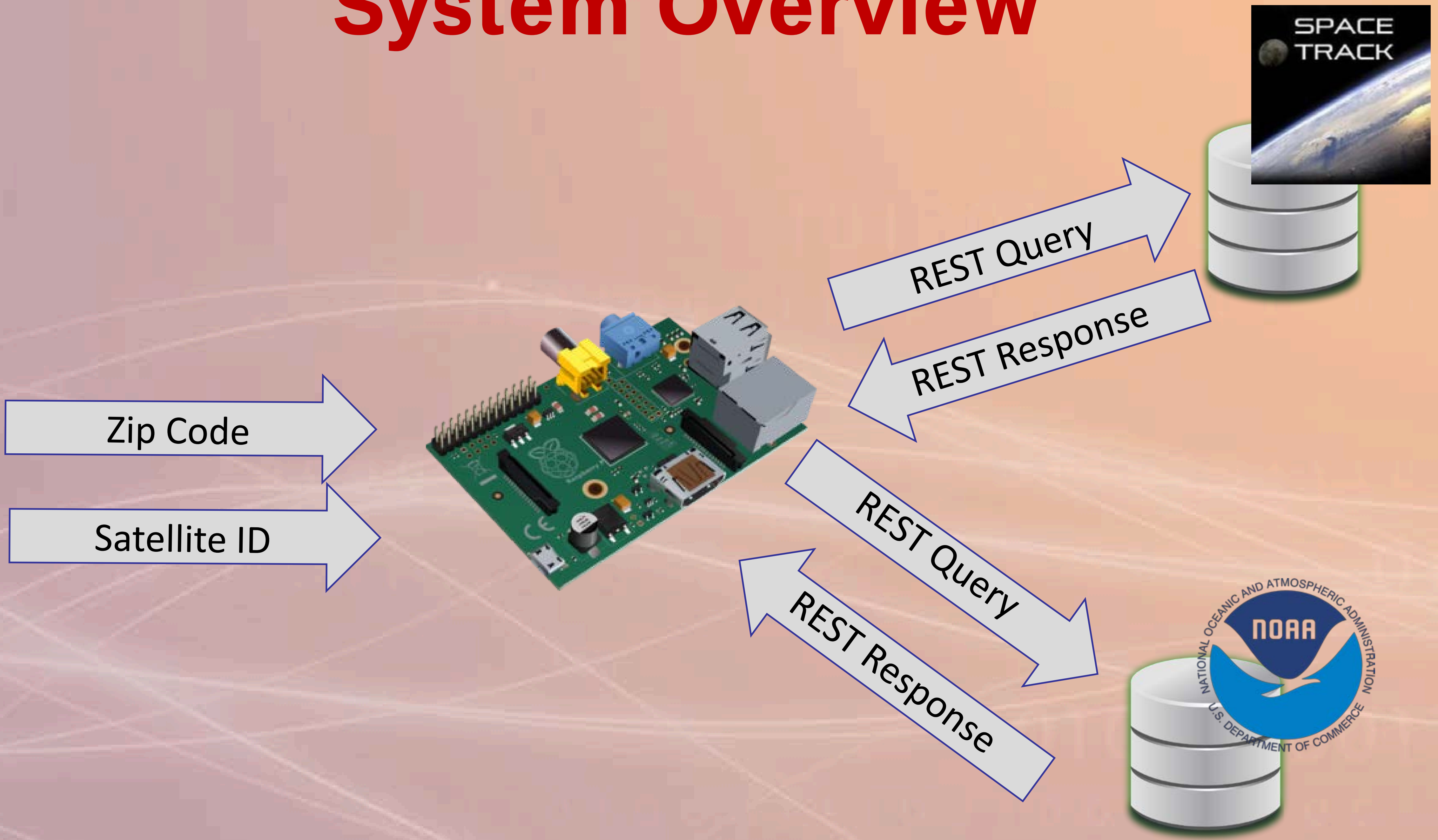
Satellite ID

SPACE TRACK
NOAA
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

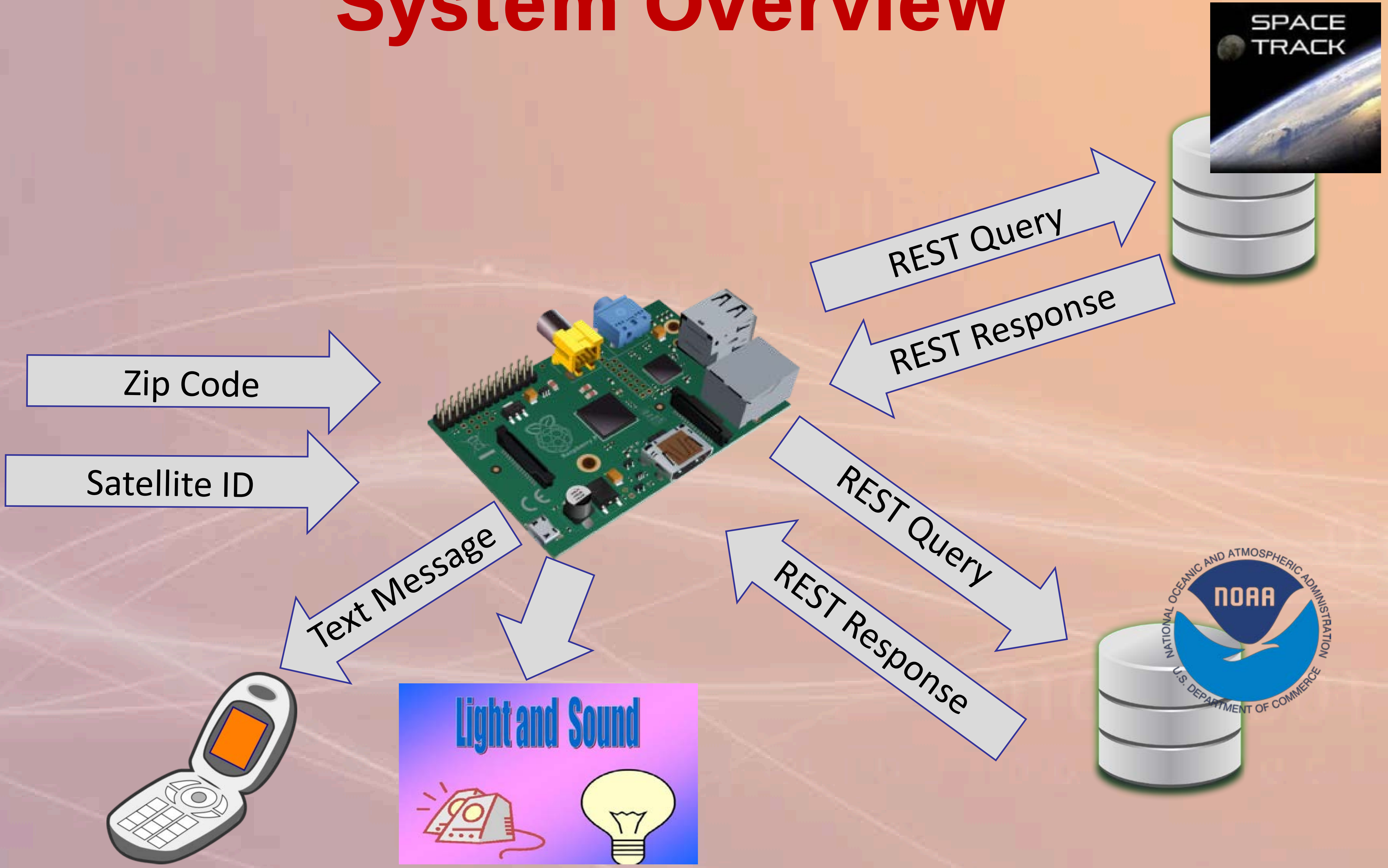
System Overview


SPACE TRACK
REST Query
REST Response
Zip Code
Satellite ID
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NOAA
U.S. DEPARTMENT OF COMMERCE

System Overview



System Overview



Zip Code

Satellite ID

REST Query

REST Response

REST Query

REST Response

Text Message

Light and Sound

SPACE
TRACK
NOAA
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

Approach

- Requires one Raspberry Pi
- Name your application ***icu.py***
- Invoke with following command line switches
  - -z indicates zipcode of viewing area
    - assume viewing locations will be restricted to the continental United States and Alaska and Hawaii
  - -s indicates NOMAD ID of satellite to view
  - Example:

```
icu.py –z 24060 –s 25544
```

- Generate event notifications (light,sound,txt msg) 15 minutes prior to beginning of viewable event

Space Track

Get TLE orbital elements for a NORAD ID satellite and date

Note:
This code example uses urllib and urllib2 libraries. You are to use Python requests library.

PyEphem

```
import sys
import math
import ephem

iss = ephem.readtle("ISS (ZARYA)",
        "1 25544U 98067A   09270.78646569  .00012443  00000-0  87997-4 0  6860",
        "2 25544  51.6377 140.0905 0009007 135.9273 312.2213 15.74420558622113")

obs = ephem.Observer()
obs.lat = '38.0'
obs.long = '-122.0'

for p in range(3):
        tr, azr, tt, altt, ts, azs = obs.next_pass(iss)
        while tr < ts :
                obs.date = tr
                iss.compute(obs)
                print "%s %4.1f %5.1f" % (tr, math.degrees(iss.alt), math.degrees(iss.az))
                tr = ephem.Date(tr + 60.0 * ephem.second)
        print
        obs.date = tr + ephem.minute
```

brainwagon

A Viewable Event

1. Satellite is sufficiently above the horizon
2. Sun is sufficiently below the horizon
3. Satellite is reflecting sunlight
4. Weather condition: Clear Sky

Clear Sky

- A "viewable event" occurs when the sky is "Clear" or "Mostly Clear".
- Use 15-day weather forecast data.
- Refer to the following glossary link for more information:
  - http://forecast.weather.gov/glossary.php

Event Notification

- 15 minutes prior to satellite appearing
  - start audible cue (can be tone, spoken text, a song)
  - start visual cue (flashing LED - on - one second, off - one second)
  - send sms text message containing information relevant to satellite viewing

- stop audible and visual cues after satellite appears

Grading

- Report: 10% - 10 points
  - 1 to 2 pages, Single-spaced, Submit as PDF

- Validation with GTAs: 90%
  - RESTful acquisition of satellite TLE – 20 points
    - Must access Space-Track using requests library
  - RESTful acquisition of weather forecast – 20 points
    - Must access NOAA using requests library
  - Calculation of five “viewable” dates/times of satellite – 20 points.
    - Note: each “viewable” date/time refers to different orbit
  - Event Notification
    - Flash LED via GPIO pin – 5 points
    - Send SMS txt message – 15 points
    - Generate sound – 10 points

GTA’s will ask you to test your code specifying different locations and satellites

Grading

- Handle HTTP error codes generated by your REST calls

Validation

When started, your application will:

1. Print satellite TLE
2. Print longitude and latitude of specified zipcode
3. Print forecast information for zipcode area (next 15 days)
   - Summarize output to show dates and sky conditions
4. Print satellite's ephemeris data from PyEphem
5. Print the next five "viewable" date/times for specified satellite
   - Include satellite position, direction of travel and duration of visibility
   - Print an appropriate message if weather conditions prohibit five "viewable" events over a 15-day forecast window
6. Halt your application
7. Adjust Raspberry Pi system time to coincide with a viewable event
8. Restart application and "wait" for the next "viewable" event
9. At the appropriate moment
   - Flash LED and generate sound
   - Send SMS txt notification containing next viewing information

Report

You must document the design, and outcomes in a brief written report. Your report should contain the following items.

- At the top of the first page of your report, include: your names (as recorded by the university); your email address; and the assignment name (e.g., “ECE 4564, Assignment 3”). Do not include your Virginia Tech ID number or your social security number.

- The body of the report must contain the following sections. Use section numbers and headings to organize your report.
  - Section 1 – Objectives: Provide a description of the design objectives and general approach to the design. Include a system diagram showing your system’s end-to-end function.
  - Section 2 – Team member responsibilities
  - Section 3 – Conclusions: Discuss the outcome of your design and any problems encountered and resolutions; what you learned by doing this project; and any experiences that were particularly good or bad.

Python Style

Follow style guide PEP0008 when writing and commenting your code

https://www.python.org/dev/peps/pep-0008/

What You Turn In

All assignments must be submitted through Canvas, no later than the due date of 2017 March 27 @ 23:55

Your assignment should be a single zip or tarball (i.e. tar.gz, tar.bz) which contains the following:

- All source code you wrote for this assignment
  - Python code running on gateway Rpi
  - Report PDF file

Assignment References

REST in Python

- Talking REST
- Requests

Basic Astronomical Computations

- What is Ephemeris?
- PyEphem

Space-Track and NOAA API

https://www.space-track.org/auth/login

http://www.ncdc.noaa.gov/cdo-web/webservices/v2

Academic Integrity

- For this assignment, it is expected that a team’s work is their own.
- The code you turn in must be your own (i.e. you need to have written your assignment).
- You are allowed to copy and paste example code from other websites, but you must include a comment in your code that attributes the website you copied the code from (i.e. original author’s name and URL to the original code).
- You can discuss the assignment with other teams.
- However, you cannot just tell another team the answer to a particular problem.

Final Thoughts

In many cases, engineers are expected to just make things work given a particular design constraint (e.g. software package to use or are limited to a particular hardware platform).

You will likely run into similar situations in this class while designing and implementing your assignments and project.

.

When you’re stuck, try searching online for a solution. Many times others have tried something similar and documented their experiences for others to learn and benefit from

If you find a neat way of doing something on your Raspberry Pi, please share your findings in a discussion post on Canvas.

Do not publically post answers to assignments, or example code until after the assignment due date.

Contact your instructor or GTA’s as soon as you encounter a problem you’re unable to solve.

Don’t wait until right before the assignment is due.