Jade Kuehnl, Richard Lopez, Truong Duong

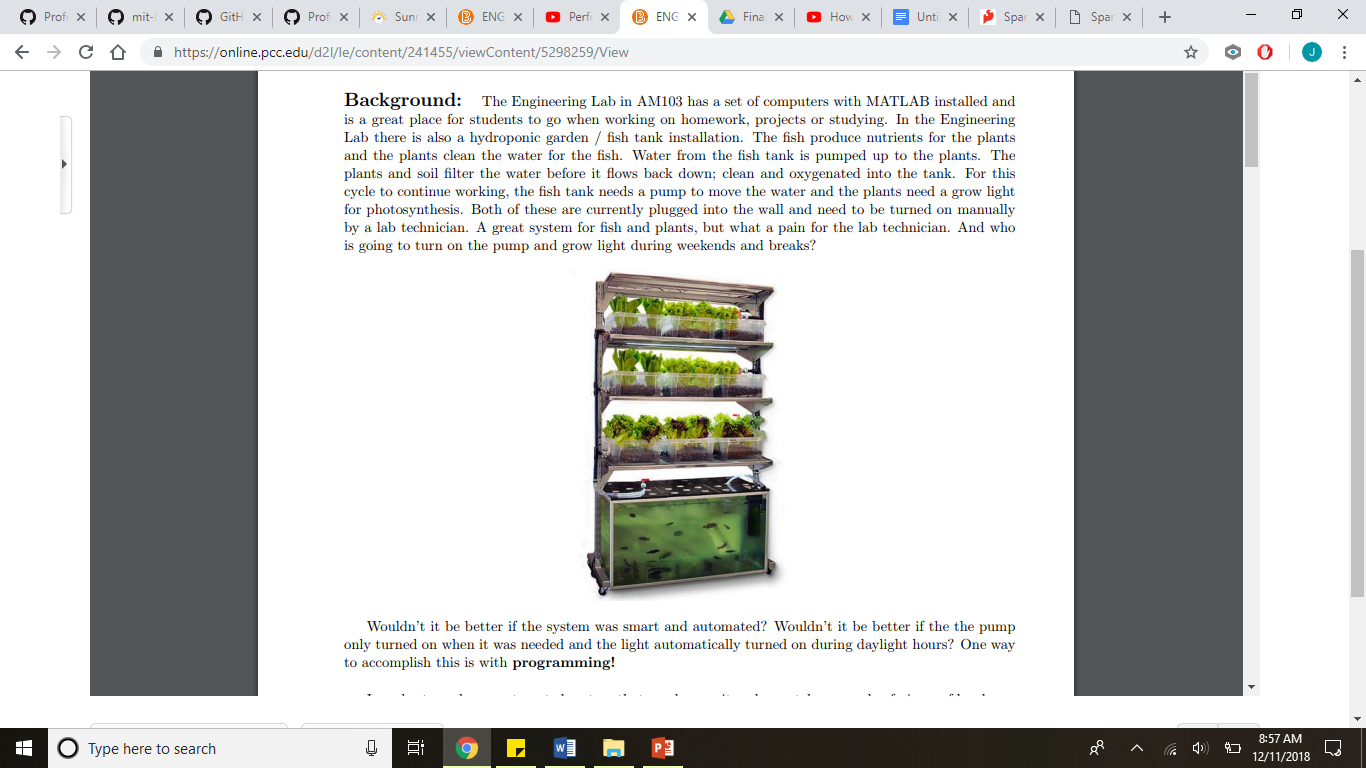
ENGR 114 Fall 2018

2018-12-11

## **Light Sensor and Relay**

#### **Problem Statement:**

Our group was tasked with making an automated system, using arduino and matlab, that can be monitored remotely to turn on the grow lights to a fish tank, based on the levels of light, using a light relay. We will be using thingspeak to direct the system by nautral daylight, using the sunrise and sunset times for Portland, OR. If, for any reason, the lab technician needs the lights off or on outside of the given times, they will be able to open Arduino, run the script, and enter ‘L’ to turn the lights off, and ‘H’ to turn the lights on.



Link for sunset/sunrise:

<https://sunrise-sunset.org/search?location=Portland%2C+OR%2C+United+States>

ThingSpeak API:

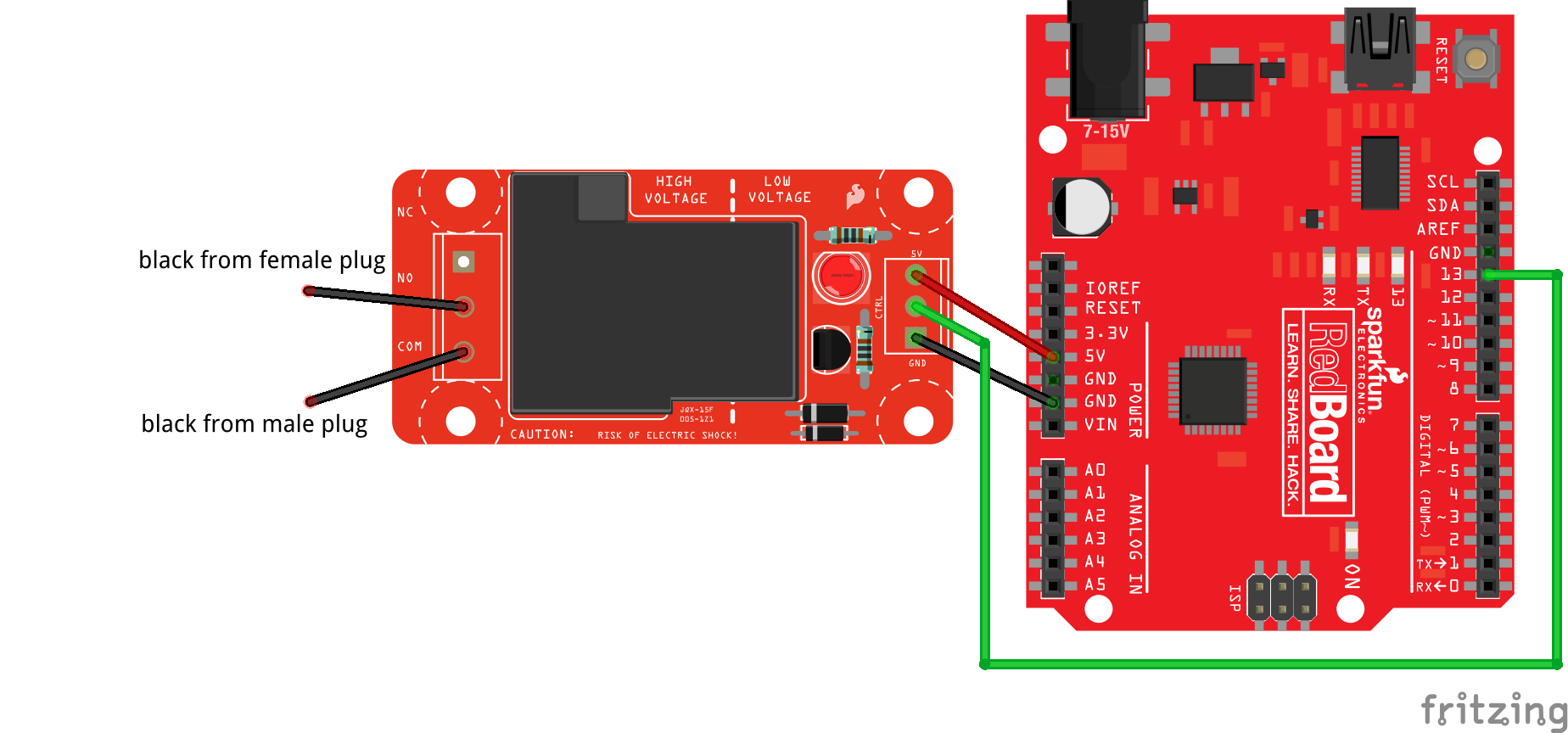
<https://api.sunrise-sunset.org/json?lat=47.608013&lng=-122.335167&date=today>

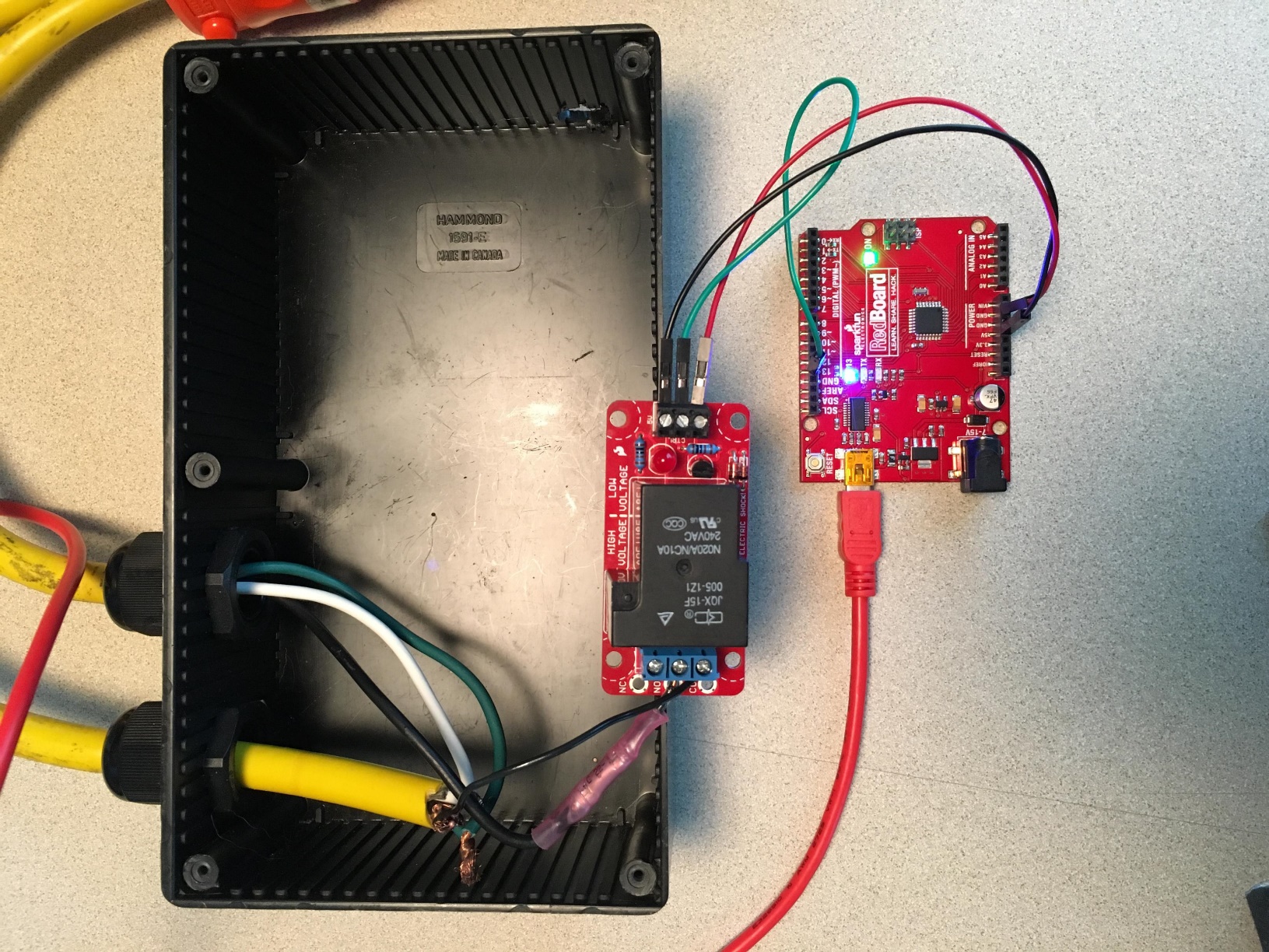
**Hardware Setup:**

Bill of Materials:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Part Name | Purpose | Item Name | URL | Price |
| RedBoard | Used to program and control Arduino by conneccting into correct pins | [SparkFun RedBoard - Programmed with Arduino](https://www.sparkfun.com/products/13975) | <https://www.sparkfun.com/products/13975> | 19.95 |
| USB cable | Connect RedBoard into a computer to connect with MATLAB/Arduino | [SparkFun USB Mini-B Cable - 6 Foot](https://www.sparkfun.com/products/11301) | <https://www.sparkfun.com/products/11301> | 3.95 |
| Bread Board | Connect negative and positive components into correct pins for LED light | [Breadboard - Self-Adhesive (White)](https://www.sparkfun.com/products/12002) | <https://www.sparkfun.com/products/12002> | 4.95 |
| Black breadboard holder | For an organized and inplace look for the Arduino and RedBoard. Helps keeps things grounded b/c of the wires | Arduino and Breadboard Holder | <https://www.sparkfun.com/products/11235> | 3.95 |
| LED | Shows us if our code is working when we want lights to turn on/off | [LED - Basic Red 5mm](https://www.sparkfun.com/products/9590) | <https://www.sparkfun.com/products/9590> | 0.35 |
| LED resistor | To control the power of the LED light as to not damage the board | [Resistor 330 Ohm 1/6th Watt PTH](https://www.sparkfun.com/products/8377) | <https://www.sparkfun.com/products/8377> | 0.25 |
| Jumper wires | Connecting electrical components of our project into the correct pins on the ReBoard | [Jumper Wires Premium 6" M/M Pack of 10](https://www.sparkfun.com/products/8431) | <https://www.sparkfun.com/products/8431> | 3.95 |
| Job Box (4 plugs) | Keep everything organized and safe around the fish tank |  | Obtained from engineering lab |  |
| 1 female wall plugs | To connect lights into |  | Obtained from engineering lab |  |
| 1 male wall plugs | To connect to the wall |  | Obtained from engineering lab |  |
| 1 relay | To pull data from thingspeak and relay information | [SparkFun Beefcake Relay Control Kit (Ver. 2.0)](https://www.sparkfun.com/products/13815) | <https://www.sparkfun.com/products/13815> | 8.95 |

**Hardware Schematic:**





**Hookup Guide:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Part | Pin | Connector | Pin | Part |
| Beefcake Relay | 5V | Jumper Wire | 5V | RedBoard |
| Beefcake Relay | CTRL | Jumper Wire | 13 | RedBoard |
| Beefcake Relay | GND | Jumper Wire | GND | RedBoard |
| Beefcake Relay | NO | Black Wire From Female Plug | Female Plug | Lights |
| Beefcake Relay | COM | Black Wire From Male Plug | Male Plug | Wall |

**MATLAB code:**

%% Arduino Group Project:

%% Objectives of code

% \* Display time, light on or off -

% \* O for on, L for off, Q for quit --

% \* Plot time on vs days --

% \* Replicate a natural cycle ---

%% Clear ,clc, close all

clear, clc, close all

%% Set up serial port and ensure it outputs data

Port = 'COM3'; % !!! Port the Arduino is connected to !!!

delete(instrfindall); % deletes any connected ports

a = serial(Port); % create the serial port

fopen(a); % open the serial port

pause(1); % pause for 1 second to make sure a connection is made

out = instrfind ('Port ',Port) % see if the port you specified is open

disp ('Serial Port is Open')

%% Pulling out data

format short g

data\_api = webread('https://api.thingspeak.com/channels/9/fields/1/last.json?timezone=America%2FLos\_Angeles');

% datestr(datetime,'HH:MM')

data\_to\_table = struct2table(data\_api);

data1 = data\_to\_table(:,1);

for i = data1

% Day Timesplit()

data\_to\_cell = table2array(data1);

split\_data = split(data\_to\_cell, '-');

data2 = split\_data(3,:);

split\_data1 = split(data2, 'T');

data = split\_data1(2,:);

split\_dat = split(data, ':');

hr = split\_dat (1,:);

hr\_mat = str2double(hr);

min = split\_dat(2,:);

min\_mat = str2double(min);

sec = split\_dat(3,:);

sec\_mat = str2double(sec);

end

%% EU Time

time\_EU = [(hr\_mat) min\_mat sec\_mat]

%hr\_mat = 1

%% Ask User for Input and send input over serial

%% Read API for sunrise/sunset

format short g

data\_api2 = webread('https://api.sunrise-sunset.org/json?lat=47.608013&lng=-122.335167&date=today');

data\_to\_table2 = struct2table(data\_api2);

data2 = data\_to\_table2(:,1);

data3 = table2array(data2);

data4 = struct2cell(data3);

sunrise\_T = data4(1);

sunset\_T = data4(2);

day\_length\_T = data4(3);

%% Sunrise converted

data\_to\_cell = table2array(sunrise\_T);

split\_data2 = split(sunrise\_T, ':');

sunrise\_hr = split\_data2 (1,:);

sunrise\_hr\_mat = str2double(sunrise\_hr)-8;

if sunrise\_hr\_mat<0

sunrise\_hr\_mat = 12 - abs(sunrise\_hr\_mat);

end

sunrise\_min = split\_data2(2,:);

sunrise\_min\_mat = str2double(sunrise\_min);

sunrise\_sec = split\_data2(3,:);

sunrise\_sec\_str = cellstr(sunrise\_sec);

sunrise\_sec\_1 = split(sunrise\_sec\_str);

sunrise\_sec\_mat = str2double(sunrise\_sec\_1(1));

time\_sunrise = [sunrise\_hr\_mat sunrise\_min\_mat sunrise\_sec\_mat];

fprintf('The sun will rise at %d:%d:%d sec am \n "turn On\" lights', sunrise\_hr\_mat, sunrise\_min\_mat, sunrise\_sec\_mat)

%% Sunset converted

data\_to\_cell\_set = table2array(sunset\_T);

split\_data3 = split(sunset\_T, ':');

sunset\_hr = split\_data3(1,:);

sunset\_hr\_mat = str2double(sunset\_hr)-8;

if sunset\_hr\_mat<0

sunset\_hr\_mat = 12 - abs(sunset\_hr\_mat);

end

sunset\_min = split\_data3(2,:);

sunset\_min\_mat = str2double(sunset\_min);

sunset\_sec = split\_data3(3,:);

sunset\_sec\_str = cellstr(sunset\_sec);

sunset\_sec\_1 = split(sunset\_sec\_str);

sunset\_sec\_mat = str2double(sunset\_sec\_1(1));

time\_sunset = [sunset\_hr\_mat sunset\_min\_mat sunset\_sec\_mat];

fprintf('\n The sun will set at %d:%d:%d sec pm \n "Turn Off\" lights', sunset\_hr\_mat, sunset\_min\_mat, sunset\_sec\_mat)

%% Day length converted

table2array(day\_length\_T);

split\_data4 = split(day\_length\_T, ':');

day\_length\_hr = split\_data4(1,:);

day\_length\_hr\_mat = str2double(day\_length\_hr);

day\_length\_min = split\_data4(2,:);

day\_length\_min\_mat = str2double(day\_length\_min);

day\_length\_sec = split\_data4(3,:);

day\_length\_sec\_str = cellstr(day\_length\_sec);

day\_length\_sec\_1 = split(day\_length\_sec\_str);

day\_length\_sec\_mat = str2double(day\_length\_sec\_1(1));

time\_day\_length = [day\_length\_hr\_mat day\_length\_min\_mat day\_length\_sec\_mat];

fprintf('\n The total amount of daylight for today will be %dhr %dmin %dsec\n ', day\_length\_hr\_mat, day\_length\_min\_mat, day\_length\_sec\_mat)

%% Converting to minutes, day length

if day\_length\_sec\_mat<=30

day\_length\_sec\_mat = 0;

else

day\_length\_sec\_mat = 1;

end

length\_min = (day\_length\_hr\_mat\*60) + day\_length\_min\_mat + day\_length\_sec\_mat;

%% Then convert time to US time am/pm

sunset\_EU\_hr\_mat = 12+sunset\_hr\_mat;

if (hr\_mat>= sunrise\_hr\_mat && hr\_mat<=sunset\_EU\_hr\_mat) %hr\_mat>=sunrise\_hr\_mat&& hr\_mat<=sunset\_hr\_mat && min\_mat>=sunrise\_min\_mat && min\_mat<=sunset\_min\_mat & sec\_mat>=sunrise\_sec\_mat & sec\_mat<=sunset\_sec\_mat)

fprintf('Now time is %d:%d:%d sec pm \n "turn On\" lights',hr\_mat,min\_mat,sec\_mat)

Light = 'Turn ON'

else

fprintf('Now time is %d:%d:%dsec am \n "Turn Off\" lights',hr\_mat,min\_mat,sec\_mat)

Light = 'Turn OFF'

end

%% Plot %%%%%%%%%%%%%%%%%%%%%

%% Pulling out data

format short g

data\_api = webread('https://api.thingspeak.com/channels/9/fields/1/last.json?timezone=America%2FLos\_Angeles');

% datestr(datetime,'HH:MM')

data\_to\_table = struct2table(data\_api);

data1 = data\_to\_table(:,1);

for i = data1

data\_to\_cell = table2array(data1);

split\_data = split(data\_to\_cell, '-');

data\_table\_url\_1\_data2 = split\_data(3,:);

Year\_str = split\_data(1);

Year = str2double(Year\_str);

Year\_char = char(Year\_str);

Month\_str = split\_data(2);

Month = str2double(Month\_str);

Month\_char = char(Month\_str);

data\_table\_url\_1\_data2 = split\_data(3,:);

split\_data1 = split(data\_table\_url\_1\_data2, 'T');

Day\_str = split\_data1(1);

Day = str2double(Day\_str);

Day\_char = char(Day\_str);

end

%% 20 if statements

link = ['https://api.sunrise-sunset.org/json?lat=47.608013&lng=-122.335167&date='];

cn = char('-');

%url = [ link, Year\_char, cn Month\_char cn Day\_char ]

%webread(url)

if (Month==1 | Month ==3 | Month==5 | Month==7 | Month==8 | Month==10 | Month==12)

%Month =

D = 1:31; % Days in the month

D\_str = num2str(D);

D\_char = char(D\_str);

%%%

url\_1 = [ link, Year\_char, cn, Month\_char, cn D\_char(1) ];

web\_1 = webread(url\_1);

data\_table\_url\_1 = struct2table(web\_1);

data\_table\_url\_1\_data2 = data\_table\_url\_1(:,1);

data\_table\_url\_1\_data3 = table2array(data\_table\_url\_1\_data2);

data\_table\_url\_1\_data4 = struct2cell(data\_table\_url\_1\_data3);

day\_length\_T\_url\_1 = data\_table\_url\_1\_data4(3);

table2array(day\_length\_T\_url\_1);

url\_1\_split\_data4 = split(day\_length\_T\_url\_1, ':');

url\_1\_day\_length\_hr = url\_1\_split\_data4(1,:);

url\_1\_day\_length\_hr\_mat = str2double(url\_1\_day\_length\_hr);

url\_1\_day\_length\_min = url\_1\_split\_data4(2,:);

url\_1\_day\_length\_min\_mat = str2double(url\_1\_day\_length\_min);

url\_1\_day\_length\_sec = url\_1\_split\_data4(3,:);

url\_1\_day\_length\_sec\_str = cellstr(url\_1\_day\_length\_sec);

url\_1\_day\_length\_sec\_1 = split(url\_1\_day\_length\_sec\_str);

url\_1\_day\_length\_sec\_mat = str2double(url\_1\_day\_length\_sec\_1(1));

url\_1\_total\_time\_1 = url\_1\_day\_length\_hr\_mat\*60 + url\_1\_day\_length\_min\_mat + url\_1\_day\_length\_sec\_mat/60;

%%%

url\_2 = [ link, Year\_char, cn, Month\_char, cn, D\_char(5) ];

web\_2 = webread(url\_2);

data\_table\_url\_2 = struct2table(web\_2);

data\_table\_url\_2\_data2 = data\_table\_url\_2(:,1);

data\_table\_url\_2\_data3 = table2array(data\_table\_url\_2\_data2);

data\_table\_url\_2\_data4 = struct2cell(data\_table\_url\_2\_data3);

day\_length\_T\_url\_2 = data\_table\_url\_2\_data4(3);

table2array(day\_length\_T\_url\_2);

url\_2\_split\_data4 = split(day\_length\_T\_url\_2, ':');

url\_2\_day\_length\_hr = url\_2\_split\_data4(1,:);

url\_2\_day\_length\_hr\_mat = str2double(url\_2\_day\_length\_hr);

url\_2\_day\_length\_min = url\_2\_split\_data4(2,:);

url\_2\_day\_length\_min\_mat = str2double(url\_2\_day\_length\_min);

url\_2\_day\_length\_sec = url\_2\_split\_data4(3,:);

url\_2\_day\_length\_sec\_str = cellstr(url\_2\_day\_length\_sec);

url\_2\_day\_length\_sec\_1 = split(url\_2\_day\_length\_sec\_str);

url\_2\_day\_length\_sec\_mat = str2double(url\_2\_day\_length\_sec\_1(1));

url\_2\_total\_time\_1 = url\_2\_day\_length\_hr\_mat\*60 + url\_2\_day\_length\_min\_mat + url\_2\_day\_length\_sec\_mat/60;

%%%

url\_3 = [ link, Year\_char, cn, Month\_char, cn, D\_char(9) ];

web\_3 = webread(url\_3);

data\_table\_url\_3 = struct2table(web\_3);

data\_table\_url\_3\_data2 = data\_table\_url\_3(:,1);

data\_table\_url\_3\_data3 = table2array(data\_table\_url\_3\_data2);

data\_table\_url\_3\_data4 = struct2cell(data\_table\_url\_3\_data3);

day\_length\_T\_url\_3 = data\_table\_url\_3\_data4(3);

table2array(day\_length\_T\_url\_3);

url\_3\_split\_data4 = split(day\_length\_T\_url\_3, ':');

url\_3\_day\_length\_hr = url\_3\_split\_data4(1,:);

url\_3\_day\_length\_hr\_mat = str2double(url\_3\_day\_length\_hr);

url\_3\_day\_length\_min = url\_3\_split\_data4(2,:);

url\_3\_day\_length\_min\_mat = str2double(url\_3\_day\_length\_min);

url\_3\_day\_length\_sec = url\_3\_split\_data4(3,:);

url\_3\_day\_length\_sec\_str = cellstr(url\_3\_day\_length\_sec);

url\_3\_day\_length\_sec\_1 = split(url\_3\_day\_length\_sec\_str);

url\_3\_day\_length\_sec\_mat = str2double(url\_3\_day\_length\_sec\_1(1));

url\_3\_total\_time\_1 = url\_3\_day\_length\_hr\_mat\*60 + url\_3\_day\_length\_min\_mat + url\_3\_day\_length\_sec\_mat/60;

url\_4 = [ link, Year\_char, cn, Month\_char, cn, D\_char(13) ];

web\_4 = webread(url\_4);

data\_table\_url\_4 = struct2table(web\_4);

data\_table\_url\_4\_data2 = data\_table\_url\_4(:,1);

data\_table\_url\_4\_data3 = table2array(data\_table\_url\_4\_data2);

data\_table\_url\_4\_data4 = struct2cell(data\_table\_url\_4\_data3);

day\_length\_T\_url\_4 = data\_table\_url\_4\_data4(3);

table2array(day\_length\_T\_url\_4);

url\_4\_split\_data4 = split(day\_length\_T\_url\_4, ':');

url\_4\_day\_length\_hr = url\_4\_split\_data4(1,:);

url\_4\_day\_length\_hr\_mat = str2double(url\_4\_day\_length\_hr);

url\_4\_day\_length\_min = url\_4\_split\_data4(2,:);

url\_4\_day\_length\_min\_mat = str2double(url\_4\_day\_length\_min);

url\_4\_day\_length\_sec = url\_4\_split\_data4(3,:);

url\_4\_day\_length\_sec\_str = cellstr(url\_4\_day\_length\_sec);

url\_4\_day\_length\_sec\_1 = split(url\_4\_day\_length\_sec\_str);

url\_4\_day\_length\_sec\_mat = str2double(url\_4\_day\_length\_sec\_1(1));

url\_4\_total\_time\_1 = url\_4\_day\_length\_hr\_mat\*60 + url\_4\_day\_length\_min\_mat + url\_4\_day\_length\_sec\_mat/60;

url\_5 = [ link, Year\_char, cn, Month\_char, cn, D\_char(17) ];

web\_5 = webread(url\_5);

data\_table\_url\_5 = struct2table(web\_5);

data\_table\_url\_5\_data2 = data\_table\_url\_5(:,1);

data\_table\_url\_5\_data3 = table2array(data\_table\_url\_5\_data2);

data\_table\_url\_5\_data4 = struct2cell(data\_table\_url\_5\_data3);

day\_length\_T\_url\_5 = data\_table\_url\_5\_data4(3);

table2array(day\_length\_T\_url\_5);

url\_5\_split\_data4 = split(day\_length\_T\_url\_5, ':');

url\_5\_day\_length\_hr = url\_5\_split\_data4(1,:);

url\_5\_day\_length\_hr\_mat = str2double(url\_5\_day\_length\_hr);

url\_5\_day\_length\_min = url\_5\_split\_data4(2,:);

url\_5\_day\_length\_min\_mat = str2double(url\_5\_day\_length\_min);

url\_5\_day\_length\_sec = url\_5\_split\_data4(3,:);

url\_5\_day\_length\_sec\_str = cellstr(url\_5\_day\_length\_sec);

url\_5\_day\_length\_sec\_1 = split(url\_5\_day\_length\_sec\_str);

url\_5\_day\_length\_sec\_mat = str2double(url\_5\_day\_length\_sec\_1(1));

url\_5\_total\_time\_1 = url\_5\_day\_length\_hr\_mat\*60 + url\_5\_day\_length\_min\_mat + url\_5\_day\_length\_sec\_mat/60;

url\_6 = [ link, Year\_char, cn, Month\_char, cn, D\_char(21) ];

web\_6 = webread(url\_6);

data\_table\_url\_6 = struct2table(web\_6);

data\_table\_url\_6\_data2 = data\_table\_url\_6(:,1);

data\_table\_url\_6\_data3 = table2array(data\_table\_url\_6\_data2);

data\_table\_url\_6\_data4 = struct2cell(data\_table\_url\_6\_data3);

day\_length\_T\_url\_6 = data\_table\_url\_6\_data4(3);

table2array(day\_length\_T\_url\_6);

url\_6\_split\_data4 = split(day\_length\_T\_url\_6, ':');

url\_6\_day\_length\_hr = url\_6\_split\_data4(1,:);

url\_6\_day\_length\_hr\_mat = str2double(url\_6\_day\_length\_hr);

url\_6\_day\_length\_min = url\_6\_split\_data4(2,:);

url\_6\_day\_length\_min\_mat = str2double(url\_6\_day\_length\_min);

url\_6\_day\_length\_sec = url\_6\_split\_data4(3,:);

url\_6\_day\_length\_sec\_str = cellstr(url\_6\_day\_length\_sec);

url\_6\_day\_length\_sec\_1 = split(url\_6\_day\_length\_sec\_str);

url\_6\_day\_length\_sec\_mat = str2double(url\_6\_day\_length\_sec\_1(1));

url\_6\_total\_time\_1 = url\_6\_day\_length\_hr\_mat\*60 + url\_6\_day\_length\_min\_mat + url\_6\_day\_length\_sec\_mat/60;

url\_7 = [ link, Year\_char, cn, Month\_char, cn, D\_char(25) ];

web\_7 = webread(url\_7);

data\_table\_url\_7 = struct2table(web\_7);

data\_table\_url\_7\_data2 = data\_table\_url\_7(:,1);

data\_table\_url\_7\_data3 = table2array(data\_table\_url\_7\_data2);

data\_table\_url\_7\_data4 = struct2cell(data\_table\_url\_7\_data3);

day\_length\_T\_url\_7 = data\_table\_url\_7\_data4(3);

table2array(day\_length\_T\_url\_7);

url\_7\_split\_data4 = split(day\_length\_T\_url\_7, ':');

url\_7\_day\_length\_hr = url\_7\_split\_data4(1,:);

url\_7\_day\_length\_hr\_mat = str2double(url\_7\_day\_length\_hr);

url\_7\_day\_length\_min = url\_7\_split\_data4(2,:);

url\_7\_day\_length\_min\_mat = str2double(url\_7\_day\_length\_min);

url\_7\_day\_length\_sec = url\_7\_split\_data4(3,:);

url\_7\_day\_length\_sec\_str = cellstr(url\_7\_day\_length\_sec);

url\_7\_day\_length\_sec\_1 = split(url\_7\_day\_length\_sec\_str);

url\_7\_day\_length\_sec\_mat = str2double(url\_7\_day\_length\_sec\_1(1));

url\_7\_total\_time\_1 = url\_7\_day\_length\_hr\_mat\*60 + url\_7\_day\_length\_min\_mat + url\_7\_day\_length\_sec\_mat/60;

url\_8 = [ link, Year\_char, cn, Month\_char, cn, D\_char(29) ];

web\_8 = webread(url\_8);

data\_table\_url\_8 = struct2table(web\_8);

data\_table\_url\_8\_data2 = data\_table\_url\_8(:,1);

data\_table\_url\_8\_data3 = table2array(data\_table\_url\_8\_data2);

data\_table\_url\_8\_data4 = struct2cell(data\_table\_url\_8\_data3);

day\_length\_T\_url\_8 = data\_table\_url\_8\_data4(3);

table2array(day\_length\_T\_url\_8);

url\_8\_split\_data4 = split(day\_length\_T\_url\_8, ':');

url\_8\_day\_length\_hr = url\_8\_split\_data4(1,:);

url\_8\_day\_length\_hr\_mat = str2double(url\_8\_day\_length\_hr);

url\_8\_day\_length\_min = url\_8\_split\_data4(2,:);

url\_8\_day\_length\_min\_mat = str2double(url\_8\_day\_length\_min);

url\_8\_day\_length\_sec = url\_8\_split\_data4(3,:);

url\_8\_day\_length\_sec\_str = cellstr(url\_8\_day\_length\_sec);

url\_8\_day\_length\_sec\_1 = split(url\_8\_day\_length\_sec\_str);

url\_8\_day\_length\_sec\_mat = str2double(url\_8\_day\_length\_sec\_1(1));

url\_8\_total\_time\_1 = url\_8\_day\_length\_hr\_mat\*60 + url\_8\_day\_length\_min\_mat + url\_8\_day\_length\_sec\_mat/60;

url\_9 = [ link, Year\_char, cn, Month\_char, cn, D\_char(33) ];

web\_9 = webread(url\_9);

data\_table\_url\_9 = struct2table(web\_9);

data\_table\_url\_9\_data2 = data\_table\_url\_9(:,1);

data\_table\_url\_9\_data3 = table2array(data\_table\_url\_9\_data2);

data\_table\_url\_9\_data4 = struct2cell(data\_table\_url\_9\_data3);

day\_length\_T\_url\_9 = data\_table\_url\_9\_data4(3);

table2array(day\_length\_T\_url\_9);

url\_9\_split\_data4 = split(day\_length\_T\_url\_9, ':');

url\_9\_day\_length\_hr = url\_9\_split\_data4(1,:);

url\_9\_day\_length\_hr\_mat = str2double(url\_9\_day\_length\_hr);

url\_9\_day\_length\_min = url\_9\_split\_data4(2,:);

url\_9\_day\_length\_min\_mat = str2double(url\_9\_day\_length\_min);

url\_9\_day\_length\_sec = url\_9\_split\_data4(3,:);

url\_9\_day\_length\_sec\_str = cellstr(url\_9\_day\_length\_sec);

url\_9\_day\_length\_sec\_1 = split(url\_9\_day\_length\_sec\_str);

url\_9\_day\_length\_sec\_mat = str2double(url\_9\_day\_length\_sec\_1(1));

url\_9\_total\_time\_1 = url\_9\_day\_length\_hr\_mat\*60 + url\_9\_day\_length\_min\_mat + url\_9\_day\_length\_sec\_mat/60;

url\_10 = [ link, Year\_char, cn, Month\_char, cn, D\_char(36:37) ];

web\_10 = webread(url\_10);

data\_table\_url\_10 = struct2table(web\_10);

data\_table\_url\_10\_data2 = data\_table\_url\_10(:,1);

data\_table\_url\_10\_data3 = table2array(data\_table\_url\_10\_data2);

data\_table\_url\_10\_data4 = struct2cell(data\_table\_url\_10\_data3);

day\_length\_T\_url\_10 = data\_table\_url\_10\_data4(3);

table2array(day\_length\_T\_url\_10);

url\_10\_split\_data4 = split(day\_length\_T\_url\_10, ':');

url\_10\_day\_length\_hr = url\_10\_split\_data4(1,:);

url\_10\_day\_length\_hr\_mat = str2double(url\_10\_day\_length\_hr);

url\_10\_day\_length\_min = url\_10\_split\_data4(2,:);

url\_10\_day\_length\_min\_mat = str2double(url\_10\_day\_length\_min);

url\_10\_day\_length\_sec = url\_10\_split\_data4(3,:);

url\_10\_day\_length\_sec\_str = cellstr(url\_10\_day\_length\_sec);

url\_10\_day\_length\_sec\_1 = split(url\_10\_day\_length\_sec\_str);

url\_10\_day\_length\_sec\_mat = str2double(url\_10\_day\_length\_sec\_1(1));

url\_10\_total\_time\_1 = url\_10\_day\_length\_hr\_mat\*60 + url\_10\_day\_length\_min\_mat + url\_10\_day\_length\_sec\_mat/60;

url\_11 = [ link, Year\_char, cn, Month\_char, cn, D\_char(40:41) ];

web\_11 = webread(url\_11);

data\_table\_url\_11 = struct2table(web\_11);

data\_table\_url\_11\_data2 = data\_table\_url\_11(:,1);

data\_table\_url\_11\_data3 = table2array(data\_table\_url\_11\_data2);

data\_table\_url\_11\_data4 = struct2cell(data\_table\_url\_11\_data3);

day\_length\_T\_url\_11 = data\_table\_url\_11\_data4(3);

table2array(day\_length\_T\_url\_11);

url\_11\_split\_data4 = split(day\_length\_T\_url\_11, ':');

url\_11\_day\_length\_hr = url\_11\_split\_data4(1,:);

url\_11\_day\_length\_hr\_mat = str2double(url\_11\_day\_length\_hr);

url\_11\_day\_length\_min = url\_11\_split\_data4(2,:);

url\_11\_day\_length\_min\_mat = str2double(url\_11\_day\_length\_min);

url\_11\_day\_length\_sec = url\_11\_split\_data4(3,:);

url\_11\_day\_length\_sec\_str = cellstr(url\_11\_day\_length\_sec);

url\_11\_day\_length\_sec\_1 = split(url\_11\_day\_length\_sec\_str);

url\_11\_day\_length\_sec\_mat = str2double(url\_11\_day\_length\_sec\_1(1));

url\_11\_total\_time\_1 = url\_11\_day\_length\_hr\_mat\*60 + url\_11\_day\_length\_min\_mat + url\_11\_day\_length\_sec\_mat/60;

url\_12 = [ link, Year\_char, cn, Month\_char, cn, D\_char(44:45) ];

web\_12 = webread(url\_12);

data\_table\_url\_12 = struct2table(web\_12);

data\_table\_url\_12\_data2 = data\_table\_url\_12(:,1);

data\_table\_url\_12\_data3 = table2array(data\_table\_url\_12\_data2);

data\_table\_url\_12\_data4 = struct2cell(data\_table\_url\_12\_data3);

day\_length\_T\_url\_12 = data\_table\_url\_12\_data4(3);

table2array(day\_length\_T\_url\_12);

url\_12\_split\_data4 = split(day\_length\_T\_url\_12, ':');

url\_12\_day\_length\_hr = url\_12\_split\_data4(1,:);

url\_12\_day\_length\_hr\_mat = str2double(url\_12\_day\_length\_hr);

url\_12\_day\_length\_min = url\_12\_split\_data4(2,:);

url\_12\_day\_length\_min\_mat = str2double(url\_12\_day\_length\_min);

url\_12\_day\_length\_sec = url\_12\_split\_data4(3,:);

url\_12\_day\_length\_sec\_str = cellstr(url\_12\_day\_length\_sec);

url\_12\_day\_length\_sec\_1 = split(url\_12\_day\_length\_sec\_str);

url\_12\_day\_length\_sec\_mat = str2double(url\_12\_day\_length\_sec\_1(1));

url\_12\_total\_time\_1 = url\_12\_day\_length\_hr\_mat\*60 + url\_12\_day\_length\_min\_mat + url\_12\_day\_length\_sec\_mat/60;

url\_13 = [ link, Year\_char, cn, Month\_char, cn, D\_char(48:49) ];

web\_13 = webread(url\_13);

data\_table\_url\_13 = struct2table(web\_13);

data\_table\_url\_13\_data2 = data\_table\_url\_13(:,1);

data\_table\_url\_13\_data3 = table2array(data\_table\_url\_13\_data2);

data\_table\_url\_13\_data4 = struct2cell(data\_table\_url\_13\_data3);

day\_length\_T\_url\_13 = data\_table\_url\_13\_data4(3);

table2array(day\_length\_T\_url\_13);

url\_13\_split\_data4 = split(day\_length\_T\_url\_13, ':');

url\_13\_day\_length\_hr = url\_13\_split\_data4(1,:);

url\_13\_day\_length\_hr\_mat = str2double(url\_13\_day\_length\_hr);

url\_13\_day\_length\_min = url\_13\_split\_data4(2,:);

url\_13\_day\_length\_min\_mat = str2double(url\_13\_day\_length\_min);

url\_13\_day\_length\_sec = url\_13\_split\_data4(3,:);

url\_13\_day\_length\_sec\_str = cellstr(url\_13\_day\_length\_sec);

url\_13\_day\_length\_sec\_1 = split(url\_13\_day\_length\_sec\_str);

url\_13\_day\_length\_sec\_mat = str2double(url\_13\_day\_length\_sec\_1(1));

url\_13\_total\_time\_1 = url\_13\_day\_length\_hr\_mat\*60 + url\_13\_day\_length\_min\_mat + url\_13\_day\_length\_sec\_mat/60;

url\_14 = [ link, Year\_char, cn, Month\_char, cn, D\_char(52:53) ];

web\_14 = webread(url\_14);

data\_table\_url\_14 = struct2table(web\_14);

data\_table\_url\_14\_data2 = data\_table\_url\_14(:,1);

data\_table\_url\_14\_data3 = table2array(data\_table\_url\_14\_data2);

data\_table\_url\_14\_data4 = struct2cell(data\_table\_url\_14\_data3);

day\_length\_T\_url\_14 = data\_table\_url\_14\_data4(3);

table2array(day\_length\_T\_url\_14);

url\_14\_split\_data4 = split(day\_length\_T\_url\_14, ':');

url\_14\_day\_length\_hr = url\_14\_split\_data4(1,:);

url\_14\_day\_length\_hr\_mat = str2double(url\_14\_day\_length\_hr);

url\_14\_day\_length\_min = url\_14\_split\_data4(2,:);

url\_14\_day\_length\_min\_mat = str2double(url\_14\_day\_length\_min);

url\_14\_day\_length\_sec = url\_14\_split\_data4(3,:);

url\_14\_day\_length\_sec\_str = cellstr(url\_14\_day\_length\_sec);

url\_14\_day\_length\_sec\_1 = split(url\_14\_day\_length\_sec\_str);

url\_14\_day\_length\_sec\_mat = str2double(url\_14\_day\_length\_sec\_1(1));

url\_14\_total\_time\_1 = url\_14\_day\_length\_hr\_mat\*60 + url\_14\_day\_length\_min\_mat + url\_14\_day\_length\_sec\_mat/60;

url\_15 = [ link, Year\_char, cn, Month\_char, cn, D\_char(56:57) ];

web\_15 = webread(url\_15);

data\_table\_url\_15 = struct2table(web\_15);

data\_table\_url\_15\_data2 = data\_table\_url\_15(:,1);

data\_table\_url\_15\_data3 = table2array(data\_table\_url\_15\_data2);

data\_table\_url\_15\_data4 = struct2cell(data\_table\_url\_15\_data3);

day\_length\_T\_url\_15 = data\_table\_url\_15\_data4(3);

table2array(day\_length\_T\_url\_15);

url\_15\_split\_data4 = split(day\_length\_T\_url\_15, ':');

url\_15\_day\_length\_hr = url\_15\_split\_data4(1,:);

url\_15\_day\_length\_hr\_mat = str2double(url\_15\_day\_length\_hr);

url\_15\_day\_length\_min = url\_15\_split\_data4(2,:);

url\_15\_day\_length\_min\_mat = str2double(url\_15\_day\_length\_min);

url\_15\_day\_length\_sec = url\_15\_split\_data4(3,:);

url\_15\_day\_length\_sec\_str = cellstr(url\_15\_day\_length\_sec);

url\_15\_day\_length\_sec\_1 = split(url\_15\_day\_length\_sec\_str);

url\_15\_day\_length\_sec\_mat = str2double(url\_15\_day\_length\_sec\_1(1));

url\_15\_total\_time\_1 = url\_15\_day\_length\_hr\_mat\*60 + url\_15\_day\_length\_min\_mat + url\_15\_day\_length\_sec\_mat/60;

url\_16 = [ link, Year\_char, cn, Month\_char, cn, D\_char(60:61) ];

web\_16 = webread(url\_16);

data\_table\_url\_16 = struct2table(web\_16);

data\_table\_url\_16\_data2 = data\_table\_url\_16(:,1);

data\_table\_url\_16\_data3 = table2array(data\_table\_url\_16\_data2);

data\_table\_url\_16\_data4 = struct2cell(data\_table\_url\_16\_data3);

day\_length\_T\_url\_16 = data\_table\_url\_16\_data4(3);

table2array(day\_length\_T\_url\_16);

url\_16\_split\_data4 = split(day\_length\_T\_url\_16, ':');

url\_16\_day\_length\_hr = url\_16\_split\_data4(1,:);

url\_16\_day\_length\_hr\_mat = str2double(url\_16\_day\_length\_hr);

url\_16\_day\_length\_min = url\_16\_split\_data4(2,:);

url\_16\_day\_length\_min\_mat = str2double(url\_16\_day\_length\_min);

url\_16\_day\_length\_sec = url\_16\_split\_data4(3,:);

url\_16\_day\_length\_sec\_str = cellstr(url\_16\_day\_length\_sec);

url\_16\_day\_length\_sec\_1 = split(url\_16\_day\_length\_sec\_str);

url\_16\_day\_length\_sec\_mat = str2double(url\_16\_day\_length\_sec\_1(1));

url\_16\_total\_time\_1 = url\_16\_day\_length\_hr\_mat\*60 + url\_16\_day\_length\_min\_mat + url\_16\_day\_length\_sec\_mat/60;

url\_17 = [ link, Year\_char, cn, Month\_char, cn, D\_char(64:65) ];

web\_17 = webread(url\_17);

data\_table\_url\_17 = struct2table(web\_17);

data\_table\_url\_17\_data2 = data\_table\_url\_17(:,1);

data\_table\_url\_17\_data3 = table2array(data\_table\_url\_17\_data2);

data\_table\_url\_17\_data4 = struct2cell(data\_table\_url\_17\_data3);

day\_length\_T\_url\_17 = data\_table\_url\_17\_data4(3);

table2array(day\_length\_T\_url\_17);

url\_17\_split\_data4 = split(day\_length\_T\_url\_17, ':');

url\_17\_day\_length\_hr = url\_17\_split\_data4(1,:);

url\_17\_day\_length\_hr\_mat = str2double(url\_17\_day\_length\_hr);

url\_17\_day\_length\_min = url\_17\_split\_data4(2,:);

url\_17\_day\_length\_min\_mat = str2double(url\_17\_day\_length\_min);

url\_17\_day\_length\_sec = url\_17\_split\_data4(3,:);

url\_17\_day\_length\_sec\_str = cellstr(url\_17\_day\_length\_sec);

url\_17\_day\_length\_sec\_1 = split(url\_17\_day\_length\_sec\_str);

url\_17\_day\_length\_sec\_mat = str2double(url\_17\_day\_length\_sec\_1(1));

url\_17\_total\_time\_1 = url\_17\_day\_length\_hr\_mat\*60 + url\_17\_day\_length\_min\_mat + url\_17\_day\_length\_sec\_mat/60;

url\_18 = [ link, Year\_char, cn, Month\_char, cn, D\_char(68:69) ];

web\_18 = webread(url\_18);

data\_table\_url\_18 = struct2table(web\_18);

data\_table\_url\_18\_data2 = data\_table\_url\_18(:,1);

data\_table\_url\_18\_data3 = table2array(data\_table\_url\_18\_data2);

data\_table\_url\_18\_data4 = struct2cell(data\_table\_url\_18\_data3);

day\_length\_T\_url\_18 = data\_table\_url\_18\_data4(3);

table2array(day\_length\_T\_url\_18);

url\_18\_split\_data4 = split(day\_length\_T\_url\_18, ':');

url\_18\_day\_length\_hr = url\_18\_split\_data4(1,:);

url\_18\_day\_length\_hr\_mat = str2double(url\_18\_day\_length\_hr);

url\_18\_day\_length\_min = url\_18\_split\_data4(2,:);

url\_18\_day\_length\_min\_mat = str2double(url\_18\_day\_length\_min);

url\_18\_day\_length\_sec = url\_18\_split\_data4(3,:);

url\_18\_day\_length\_sec\_str = cellstr(url\_18\_day\_length\_sec);

url\_18\_day\_length\_sec\_1 = split(url\_18\_day\_length\_sec\_str);

url\_18\_day\_length\_sec\_mat = str2double(url\_18\_day\_length\_sec\_1(1));

url\_18\_total\_time\_1 = url\_18\_day\_length\_hr\_mat\*60 + url\_18\_day\_length\_min\_mat + url\_18\_day\_length\_sec\_mat/60;

url\_19 = [ link, Year\_char, cn, Month\_char, cn, D\_char(72:73) ];

web\_19 = webread(url\_19);

data\_table\_url\_19 = struct2table(web\_19);

data\_table\_url\_19\_data2 = data\_table\_url\_19(:,1);

data\_table\_url\_19\_data3 = table2array(data\_table\_url\_19\_data2);

data\_table\_url\_19\_data4 = struct2cell(data\_table\_url\_19\_data3);

day\_length\_T\_url\_19 = data\_table\_url\_19\_data4(3);

table2array(day\_length\_T\_url\_19);

url\_19\_split\_data4 = split(day\_length\_T\_url\_19, ':');

url\_19\_day\_length\_hr = url\_19\_split\_data4(1,:);

url\_19\_day\_length\_hr\_mat = str2double(url\_19\_day\_length\_hr);

url\_19\_day\_length\_min = url\_19\_split\_data4(2,:);

url\_19\_day\_length\_min\_mat = str2double(url\_19\_day\_length\_min);

url\_19\_day\_length\_sec = url\_19\_split\_data4(3,:);

url\_19\_day\_length\_sec\_str = cellstr(url\_19\_day\_length\_sec);

url\_19\_day\_length\_sec\_1 = split(url\_19\_day\_length\_sec\_str);

url\_19\_day\_length\_sec\_mat = str2double(url\_19\_day\_length\_sec\_1(1));

url\_19\_total\_time\_1 = url\_19\_day\_length\_hr\_mat\*60 + url\_19\_day\_length\_min\_mat + url\_19\_day\_length\_sec\_mat/60;

url\_20 = [ link, Year\_char, cn, Month\_char, cn, D\_char(76:77) ];

web\_20 = webread(url\_20);

data\_table\_url\_20 = struct2table(web\_20);

data\_table\_url\_20\_data2 = data\_table\_url\_20(:,1);

data\_table\_url\_20\_data3 = table2array(data\_table\_url\_20\_data2);

data\_table\_url\_20\_data4 = struct2cell(data\_table\_url\_20\_data3);

day\_length\_T\_url\_20 = data\_table\_url\_20\_data4(3);

table2array(day\_length\_T\_url\_20);

url\_20\_split\_data4 = split(day\_length\_T\_url\_20, ':');

url\_20\_day\_length\_hr = url\_20\_split\_data4(1,:);

url\_20\_day\_length\_hr\_mat = str2double(url\_20\_day\_length\_hr);

url\_20\_day\_length\_min = url\_20\_split\_data4(2,:);

url\_20\_day\_length\_min\_mat = str2double(url\_20\_day\_length\_min);

url\_20\_day\_length\_sec = url\_20\_split\_data4(3,:);

url\_20\_day\_length\_sec\_str = cellstr(url\_20\_day\_length\_sec);

url\_20\_day\_length\_sec\_1 = split(url\_20\_day\_length\_sec\_str);

url\_20\_day\_length\_sec\_mat = str2double(url\_20\_day\_length\_sec\_1(1));

url\_20\_total\_time\_1 = url\_20\_day\_length\_hr\_mat\*60 + url\_20\_day\_length\_min\_mat + url\_20\_day\_length\_sec\_mat/60;

url\_21 = [ link, Year\_char, cn, Month\_char, cn, D\_char(80:81) ];

web\_21 = webread(url\_21);

data\_table\_url\_21 = struct2table(web\_21);

data\_table\_url\_21\_data2 = data\_table\_url\_21(:,1);

data\_table\_url\_21\_data3 = table2array(data\_table\_url\_21\_data2);

data\_table\_url\_21\_data4 = struct2cell(data\_table\_url\_21\_data3);

day\_length\_T\_url\_21 = data\_table\_url\_21\_data4(3);

table2array(day\_length\_T\_url\_21);

url\_21\_split\_data4 = split(day\_length\_T\_url\_21, ':');

url\_21\_day\_length\_hr = url\_21\_split\_data4(1,:);

url\_21\_day\_length\_hr\_mat = str2double(url\_21\_day\_length\_hr);

url\_21\_day\_length\_min = url\_21\_split\_data4(2,:);

url\_21\_day\_length\_min\_mat = str2double(url\_21\_day\_length\_min);

url\_21\_day\_length\_sec = url\_21\_split\_data4(3,:);

url\_21\_day\_length\_sec\_str = cellstr(url\_21\_day\_length\_sec);

url\_21\_day\_length\_sec\_1 = split(url\_21\_day\_length\_sec\_str);

url\_21\_day\_length\_sec\_mat = str2double(url\_21\_day\_length\_sec\_1(1));

url\_21\_total\_time\_1 = url\_21\_day\_length\_hr\_mat\*60 + url\_21\_day\_length\_min\_mat + url\_21\_day\_length\_sec\_mat/60;

url\_22 = [ link, Year\_char, cn, Month\_char, cn, D\_char(84:85) ];

web\_22 = webread(url\_22);

data\_table\_url\_22 = struct2table(web\_22);

data\_table\_url\_22\_data2 = data\_table\_url\_22(:,1);

data\_table\_url\_22\_data3 = table2array(data\_table\_url\_22\_data2);

data\_table\_url\_22\_data4 = struct2cell(data\_table\_url\_22\_data3);

day\_length\_T\_url\_22 = data\_table\_url\_22\_data4(3);

table2array(day\_length\_T\_url\_22);

url\_22\_split\_data4 = split(day\_length\_T\_url\_22, ':');

url\_22\_day\_length\_hr = url\_22\_split\_data4(1,:);

url\_22\_day\_length\_hr\_mat = str2double(url\_22\_day\_length\_hr);

url\_22\_day\_length\_min = url\_22\_split\_data4(2,:);

url\_22\_day\_length\_min\_mat = str2double(url\_22\_day\_length\_min);

url\_22\_day\_length\_sec = url\_22\_split\_data4(3,:);

url\_22\_day\_length\_sec\_str = cellstr(url\_22\_day\_length\_sec);

url\_22\_day\_length\_sec\_1 = split(url\_22\_day\_length\_sec\_str);

url\_22\_day\_length\_sec\_mat = str2double(url\_22\_day\_length\_sec\_1(1));

url\_22\_total\_time\_1 = url\_22\_day\_length\_hr\_mat\*60 + url\_22\_day\_length\_min\_mat + url\_22\_day\_length\_sec\_mat/60;

url\_23 = [ link, Year\_char, cn, Month\_char, cn, D\_char(88:89) ];

web\_23 = webread(url\_23);

data\_table\_url\_23 = struct2table(web\_23);

data\_table\_url\_23\_data2 = data\_table\_url\_23(:,1);

data\_table\_url\_23\_data3 = table2array(data\_table\_url\_23\_data2);

data\_table\_url\_23\_data4 = struct2cell(data\_table\_url\_23\_data3);

day\_length\_T\_url\_23 = data\_table\_url\_23\_data4(3);

table2array(day\_length\_T\_url\_23);

url\_23\_split\_data4 = split(day\_length\_T\_url\_23, ':');

url\_23\_day\_length\_hr = url\_23\_split\_data4(1,:);

url\_23\_day\_length\_hr\_mat = str2double(url\_23\_day\_length\_hr);

url\_23\_day\_length\_min = url\_23\_split\_data4(2,:);

url\_23\_day\_length\_min\_mat = str2double(url\_23\_day\_length\_min);

url\_23\_day\_length\_sec = url\_23\_split\_data4(3,:);

url\_23\_day\_length\_sec\_str = cellstr(url\_23\_day\_length\_sec);

url\_23\_day\_length\_sec\_1 = split(url\_23\_day\_length\_sec\_str);

url\_23\_day\_length\_sec\_mat = str2double(url\_23\_day\_length\_sec\_1(1));

url\_23\_total\_time\_1 = url\_23\_day\_length\_hr\_mat\*60 + url\_23\_day\_length\_min\_mat + url\_23\_day\_length\_sec\_mat/60;

url\_24 = [ link, Year\_char, cn, Month\_char, cn, D\_char(92:93) ];

web\_24 = webread(url\_24);

data\_table\_url\_24 = struct2table(web\_24);

data\_table\_url\_24\_data2 = data\_table\_url\_24(:,1);

data\_table\_url\_24\_data3 = table2array(data\_table\_url\_24\_data2);

data\_table\_url\_24\_data4 = struct2cell(data\_table\_url\_24\_data3);

day\_length\_T\_url\_24 = data\_table\_url\_24\_data4(3);

table2array(day\_length\_T\_url\_24);

url\_24\_split\_data4 = split(day\_length\_T\_url\_24, ':');

url\_24\_day\_length\_hr = url\_24\_split\_data4(1,:);

url\_24\_day\_length\_hr\_mat = str2double(url\_24\_day\_length\_hr);

url\_24\_day\_length\_min = url\_24\_split\_data4(2,:);

url\_24\_day\_length\_min\_mat = str2double(url\_24\_day\_length\_min);

url\_24\_day\_length\_sec = url\_24\_split\_data4(3,:);

url\_24\_day\_length\_sec\_str = cellstr(url\_24\_day\_length\_sec);

url\_24\_day\_length\_sec\_1 = split(url\_24\_day\_length\_sec\_str);

url\_24\_day\_length\_sec\_mat = str2double(url\_24\_day\_length\_sec\_1(1));

url\_24\_total\_time\_1 = url\_24\_day\_length\_hr\_mat\*60 + url\_24\_day\_length\_min\_mat + url\_24\_day\_length\_sec\_mat/60;

url\_25 = [ link, Year\_char, cn, Month\_char, cn, D\_char(96:97) ];

web\_25 = webread(url\_25);

data\_table\_url\_25 = struct2table(web\_25);

data\_table\_url\_25\_data2 = data\_table\_url\_25(:,1);

data\_table\_url\_25\_data3 = table2array(data\_table\_url\_25\_data2);

data\_table\_url\_25\_data4 = struct2cell(data\_table\_url\_25\_data3);

day\_length\_T\_url\_25 = data\_table\_url\_25\_data4(3);

table2array(day\_length\_T\_url\_25);

url\_25\_split\_data4 = split(day\_length\_T\_url\_25, ':');

url\_25\_day\_length\_hr = url\_25\_split\_data4(1,:);

url\_25\_day\_length\_hr\_mat = str2double(url\_25\_day\_length\_hr);

url\_25\_day\_length\_min = url\_25\_split\_data4(2,:);

url\_25\_day\_length\_min\_mat = str2double(url\_25\_day\_length\_min);

url\_25\_day\_length\_sec = url\_25\_split\_data4(3,:);

url\_25\_day\_length\_sec\_str = cellstr(url\_25\_day\_length\_sec);

url\_25\_day\_length\_sec\_1 = split(url\_25\_day\_length\_sec\_str);

url\_25\_day\_length\_sec\_mat = str2double(url\_25\_day\_length\_sec\_1(1));

url\_25\_total\_time\_1 = url\_25\_day\_length\_hr\_mat\*60 + url\_25\_day\_length\_min\_mat + url\_25\_day\_length\_sec\_mat/60;

url\_26 = [ link, Year\_char, cn, Month\_char, cn, D\_char(100:101) ];

web\_26 = webread(url\_26);

data\_table\_url\_26 = struct2table(web\_26);

data\_table\_url\_26\_data2 = data\_table\_url\_26(:,1);

data\_table\_url\_26\_data3 = table2array(data\_table\_url\_26\_data2);

data\_table\_url\_26\_data4 = struct2cell(data\_table\_url\_26\_data3);

day\_length\_T\_url\_26 = data\_table\_url\_26\_data4(3);

table2array(day\_length\_T\_url\_26);

url\_26\_split\_data4 = split(day\_length\_T\_url\_26, ':');

url\_26\_day\_length\_hr = url\_26\_split\_data4(1,:);

url\_26\_day\_length\_hr\_mat = str2double(url\_26\_day\_length\_hr);

url\_26\_day\_length\_min = url\_26\_split\_data4(2,:);

url\_26\_day\_length\_min\_mat = str2double(url\_26\_day\_length\_min);

url\_26\_day\_length\_sec = url\_26\_split\_data4(3,:);

url\_26\_day\_length\_sec\_str = cellstr(url\_26\_day\_length\_sec);

url\_26\_day\_length\_sec\_1 = split(url\_26\_day\_length\_sec\_str);

url\_26\_day\_length\_sec\_mat = str2double(url\_26\_day\_length\_sec\_1(1));

url\_26\_total\_time\_1 = url\_26\_day\_length\_hr\_mat\*60 + url\_26\_day\_length\_min\_mat + url\_26\_day\_length\_sec\_mat/60;

url\_27 = [ link, Year\_char, cn, Month\_char, cn, D\_char(104:105) ];

web\_27 = webread(url\_27);

data\_table\_url\_27 = struct2table(web\_27);

data\_table\_url\_27\_data2 = data\_table\_url\_27(:,1);

data\_table\_url\_27\_data3 = table2array(data\_table\_url\_27\_data2);

data\_table\_url\_27\_data4 = struct2cell(data\_table\_url\_27\_data3);

day\_length\_T\_url\_27 = data\_table\_url\_27\_data4(3);

table2array(day\_length\_T\_url\_27);

url\_27\_split\_data4 = split(day\_length\_T\_url\_27, ':');

url\_27\_day\_length\_hr = url\_27\_split\_data4(1,:);

url\_27\_day\_length\_hr\_mat = str2double(url\_27\_day\_length\_hr);

url\_27\_day\_length\_min = url\_27\_split\_data4(2,:);

url\_27\_day\_length\_min\_mat = str2double(url\_27\_day\_length\_min);

url\_27\_day\_length\_sec = url\_27\_split\_data4(3,:);

url\_27\_day\_length\_sec\_str = cellstr(url\_27\_day\_length\_sec);

url\_27\_day\_length\_sec\_1 = split(url\_27\_day\_length\_sec\_str);

url\_27\_day\_length\_sec\_mat = str2double(url\_27\_day\_length\_sec\_1(1));

url\_27\_total\_time\_1 = url\_27\_day\_length\_hr\_mat\*60 + url\_27\_day\_length\_min\_mat + url\_27\_day\_length\_sec\_mat/60;

url\_28 = [ link, Year\_char, cn, Month\_char, cn, D\_char(108:109) ];

web\_28 = webread(url\_28);

data\_table\_url\_28 = struct2table(web\_28);

data\_table\_url\_28\_data2 = data\_table\_url\_28(:,1);

data\_table\_url\_28\_data3 = table2array(data\_table\_url\_28\_data2);

data\_table\_url\_28\_data4 = struct2cell(data\_table\_url\_28\_data3);

day\_length\_T\_url\_28 = data\_table\_url\_28\_data4(3);

table2array(day\_length\_T\_url\_28);

url\_28\_split\_data4 = split(day\_length\_T\_url\_28, ':');

url\_28\_day\_length\_hr = url\_28\_split\_data4(1,:);

url\_28\_day\_length\_hr\_mat = str2double(url\_28\_day\_length\_hr);

url\_28\_day\_length\_min = url\_28\_split\_data4(2,:);

url\_28\_day\_length\_min\_mat = str2double(url\_28\_day\_length\_min);

url\_28\_day\_length\_sec = url\_28\_split\_data4(3,:);

url\_28\_day\_length\_sec\_str = cellstr(url\_28\_day\_length\_sec);

url\_28\_day\_length\_sec\_1 = split(url\_28\_day\_length\_sec\_str);

url\_28\_day\_length\_sec\_mat = str2double(url\_28\_day\_length\_sec\_1(1));

url\_28\_total\_time\_1 = url\_28\_day\_length\_hr\_mat\*60 + url\_28\_day\_length\_min\_mat + url\_28\_day\_length\_sec\_mat/60;

url\_29 = [ link, Year\_char, cn, Month\_char, cn, D\_char(112:113) ];

web\_29 = webread(url\_29);

data\_table\_url\_29 = struct2table(web\_29);

data\_table\_url\_29\_data2 = data\_table\_url\_29(:,1);

data\_table\_url\_29\_data3 = table2array(data\_table\_url\_29\_data2);

data\_table\_url\_29\_data4 = struct2cell(data\_table\_url\_29\_data3);

day\_length\_T\_url\_29 = data\_table\_url\_29\_data4(3);

table2array(day\_length\_T\_url\_29);

url\_29\_split\_data4 = split(day\_length\_T\_url\_29, ':');

url\_29\_day\_length\_hr = url\_29\_split\_data4(1,:);

url\_29\_day\_length\_hr\_mat = str2double(url\_29\_day\_length\_hr);

url\_29\_day\_length\_min = url\_29\_split\_data4(2,:);

url\_29\_day\_length\_min\_mat = str2double(url\_29\_day\_length\_min);

url\_29\_day\_length\_sec = url\_29\_split\_data4(3,:);

url\_29\_day\_length\_sec\_str = cellstr(url\_29\_day\_length\_sec);

url\_29\_day\_length\_sec\_1 = split(url\_29\_day\_length\_sec\_str);

url\_29\_day\_length\_sec\_mat = str2double(url\_29\_day\_length\_sec\_1(1));

url\_29\_total\_time\_1 = url\_29\_day\_length\_hr\_mat\*60 + url\_29\_day\_length\_min\_mat + url\_29\_day\_length\_sec\_mat/60;

url\_30 = [ link, Year\_char, cn, Month\_char, cn, D\_char(116:117) ];

web\_30 = webread(url\_30);

data\_table\_url\_30 = struct2table(web\_30);

data\_table\_url\_30\_data2 = data\_table\_url\_30(:,1);

data\_table\_url\_30\_data3 = table2array(data\_table\_url\_30\_data2);

data\_table\_url\_30\_data4 = struct2cell(data\_table\_url\_30\_data3);

day\_length\_T\_url\_30 = data\_table\_url\_30\_data4(3);

table2array(day\_length\_T\_url\_30);

url\_30\_split\_data4 = split(day\_length\_T\_url\_30, ':');

url\_30\_day\_length\_hr = url\_30\_split\_data4(1,:);

url\_30\_day\_length\_hr\_mat = str2double(url\_30\_day\_length\_hr);

url\_30\_day\_length\_min = url\_30\_split\_data4(2,:);

url\_30\_day\_length\_min\_mat = str2double(url\_30\_day\_length\_min);

url\_30\_day\_length\_sec = url\_30\_split\_data4(3,:);

url\_30\_day\_length\_sec\_str = cellstr(url\_30\_day\_length\_sec);

url\_30\_day\_length\_sec\_1 = split(url\_30\_day\_length\_sec\_str);

url\_30\_day\_length\_sec\_mat = str2double(url\_30\_day\_length\_sec\_1(1));

url\_30\_total\_time\_1 = url\_30\_day\_length\_hr\_mat\*60 + url\_30\_day\_length\_min\_mat + url\_30\_day\_length\_sec\_mat/60;

url\_31 = [ link, Year\_char, cn, Month\_char, cn, D\_char(120:121) ];

web\_31 = webread(url\_31);

data\_table\_url\_31 = struct2table(web\_31);

data\_table\_url\_31\_data2 = data\_table\_url\_31(:,1);

data\_table\_url\_31\_data3 = table2array(data\_table\_url\_31\_data2);

data\_table\_url\_31\_data4 = struct2cell(data\_table\_url\_31\_data3);

day\_length\_T\_url\_31 = data\_table\_url\_31\_data4(3);

table2array(day\_length\_T\_url\_31);

url\_31\_split\_data4 = split(day\_length\_T\_url\_31, ':');

url\_31\_day\_length\_hr = url\_31\_split\_data4(1,:);

url\_31\_day\_length\_hr\_mat = str2double(url\_31\_day\_length\_hr);

url\_31\_day\_length\_min = url\_31\_split\_data4(2,:);

url\_31\_day\_length\_min\_mat = str2double(url\_31\_day\_length\_min);

url\_31\_day\_length\_sec = url\_31\_split\_data4(3,:);

url\_31\_day\_length\_sec\_str = cellstr(url\_31\_day\_length\_sec);

url\_31\_day\_length\_sec\_1 = split(url\_31\_day\_length\_sec\_str);

url\_31\_day\_length\_sec\_mat = str2double(url\_31\_day\_length\_sec\_1(1));

url\_31\_total\_time\_1 = url\_31\_day\_length\_hr\_mat\*60 + url\_31\_day\_length\_min\_mat + url\_31\_day\_length\_sec\_mat/60

Y\_data= [ url\_1\_total\_time\_1 url\_2\_total\_time\_1 url\_3\_total\_time\_1 url\_4\_total\_time\_1

url\_5\_total\_time\_1 url\_6\_total\_time\_1 url\_7\_total\_time\_1 url\_8\_total\_time\_1

url\_9\_total\_time\_1 url\_10\_total\_time\_1 url\_11\_total\_time\_1 url\_12\_total\_time\_1

url\_13\_total\_time\_1 url\_14\_total\_time\_1 url\_15\_total\_time\_1 url\_16\_total\_time\_1

url\_17\_total\_time\_1 url\_18\_total\_time\_1 url\_19\_total\_time\_1 url\_20\_total\_time\_1

url\_21\_total\_time\_1 url\_22\_total\_time\_1 url\_23\_total\_time\_1 url\_24\_total\_time\_1

url\_25\_total\_time\_1 url\_26\_total\_time\_1 url\_27\_total\_time\_1 url\_28\_total\_time\_1

url\_29\_total\_time\_1 url\_30\_total\_time\_1 url\_31\_total\_time\_1 ];

plot(D,Y\_data)

end

%% Close the serial port

fclose(a);

delete(a)

clear a;

disp('Serial Port is closed')

**Arduino Code:**

/\*

Physical Pixel

An example of using the Arduino board to receive data from the computer. In

this case, the Arduino boards turns on an LED when it receives the character

'H', and turns off the LED when it receives the character 'L'.

The data can be sent from the Arduino Serial Monitor, or another program like

Processing (see code below), Flash (via a serial-net proxy), PD, or Max/MSP.

The circuit:

- LED connected from digital pin 13 to ground

created 2006

by David A. Mellis

modified 30 Aug 2011

by Tom Igoe and Scott Fitzgerald

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/PhysicalPixel

\*/

// Arduino IoT Project - Light Relay Group

// Author: Richard Lopez, Jade Kuehnl, Truong Duong

// Course/Term: ENGR114 Fall 2018

// Date: 12/11/18

// Description: This Arduino code should turn on and of the relay accordingly to a

// Matlab script, and how the code communicate to it over serial.

// NOTE: This code was not created by any member of the team, the Arduino IoT Project member just added more comment into the code.

const int ledPin = 13; // the pin that the LED is attached to

int incomingByte; // a variable to read incoming serial data into

void setup() {

// initialize serial communication:

Serial.begin(9600);

// initialize the LED pin as an output:

pinMode(ledPin, OUTPUT);

}

void loop() {

// see if there's incoming serial data:

if (Serial.available() > 0) {

// read the oldest byte in the serial buffer:

incomingByte = Serial.read();

// if it's a capital H (ASCII 72), turn on the LED:

if (incomingByte == 'H') {

digitalWrite(ledPin, HIGH);

}

// if it's an L (ASCII 76) turn off the LED:

if (incomingByte == 'L') {

digitalWrite(ledPin, LOW);

}

}

}

#### **Results:**

#### **Future Work:**

To build on this project, it would make sense to add a Graphical User Interface or a button on the Arduino for the user to override the times when lights are automatically on or off.

#### **License:**

|  |
| --- |
| ## creative commons |
|  |  |
|  | # Attribution-NonCommercial-ShareAlike 4.0 International |
|  |  |
|  | Creative Commons Corporation (“Creative Commons”) is not a law firm and does not provide legal services or legal advice. Distribution of Creative Commons public licenses does not create a lawyer-client or other relationship. Creative Commons makes its licenses and related information available on an “as-is” basis. Creative Commons gives no warranties regarding its licenses, any material licensed under their terms and conditions, or any related information. Creative Commons disclaims all liability for damages resulting from their use to the fullest extent possible. |
|  |  |
|  | ### Using Creative Commons Public Licenses |
|  |  |
|  | Creative Commons public licenses provide a standard set of terms and conditions that creators and other rights holders may use to share original works of authorship and other material subject to copyright and certain other rights specified in the public license below. The following considerations are for informational purposes only, are not exhaustive, and do not form part of our licenses. |
|  |  |
|  | \* \_\_Considerations for licensors:\_\_ Our public licenses are intended for use by those authorized to give the public permission to use material in ways otherwise restricted by copyright and certain other rights. Our licenses are irrevocable. Licensors should read and understand the terms and conditions of the license they choose before applying it. Licensors should also secure all rights necessary before applying our licenses so that the public can reuse the material as expected. Licensors should clearly mark any material not subject to the license. This includes other CC-licensed material, or material used under an exception or limitation to copyright. [More considerations for licensors](http://wiki.creativecommons.org/Considerations\_for\_licensors\_and\_licensees#Considerations\_for\_licensors). |
|  |  |
|  | \* \_\_Considerations for the public:\_\_ By using one of our public licenses, a licensor grants the public permission to use the licensed material under specified terms and conditions. If the licensor’s permission is not necessary for any reason–for example, because of any applicable exception or limitation to copyright–then that use is not regulated by the license. Our licenses grant only permissions under copyright and certain other rights that a licensor has authority to grant. Use of the licensed material may still be restricted for other reasons, including because others have copyright or other rights in the material. A licensor may make special requests, such as asking that all changes be marked or described. Although not required by our licenses, you are encouraged to respect those requests where reasonable. [More considerations for the public](http://wiki.creativecommons.org/Considerations\_for\_licensors\_and\_licensees#Considerations\_for\_licensees). |
|  |  |
|  | ## Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License |
|  |  |
|  | By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions. |
|  |  |
|  | ### Section 1 – Definitions. |
|  |  |
|  | a. \_\_Adapted Material\_\_ means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image. |
|  |  |
|  | b. \_\_Adapter's License\_\_ means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License. |
|  |  |
|  | c. \_\_BY-NC-SA Compatible License\_\_ means a license listed at [creativecommons.org/compatiblelicenses](http://creativecommons.org/compatiblelicenses), approved by Creative Commons as essentially the equivalent of this Public License. |
|  |  |
|  | d. \_\_Copyright and Similar Rights\_\_ means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights. |
|  |  |
|  | e. \_\_Effective Technological Measures\_\_ means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements. |
|  |  |
|  | f. \_\_Exceptions and Limitations\_\_ means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material. |
|  |  |
|  | g. \_\_License Elements\_\_ means the license attributes listed in the name of a Creative Commons Public License. The License Elements of this Public License are Attribution, NonCommercial, and ShareAlike. |
|  |  |
|  | h. \_\_Licensed Material\_\_ means the artistic or literary work, database, or other material to which the Licensor applied this Public License. |
|  |  |
|  | i. \_\_Licensed Rights\_\_ means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license. |
|  |  |
|  | j. \_\_Licensor\_\_ means the individual(s) or entity(ies) granting rights under this Public License. |
|  |  |
|  | k. \_\_NonCommercial\_\_ means not primarily intended for or directed towards commercial advantage or monetary compensation. For purposes of this Public License, the exchange of the Licensed Material for other material subject to Copyright and Similar Rights by digital file-sharing or similar means is NonCommercial provided there is no payment of monetary compensation in connection with the exchange. |
|  |  |
|  | l. \_\_Share\_\_ means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them. |
|  |  |
|  | m. \_\_Sui Generis Database Rights\_\_ means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world. |
|  |  |
|  | n. \_\_You\_\_ means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning. |
|  |  |
|  | ### Section 2 – Scope. |
|  |  |
|  | a. \_\_\_License grant.\_\_\_ |
|  |  |
|  | 1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to: |
|  |  |
|  | A. reproduce and Share the Licensed Material, in whole or in part, for NonCommercial purposes only; and |
|  |  |
|  | B. produce, reproduce, and Share Adapted Material for NonCommercial purposes only. |
|  |  |
|  | 2. \_\_Exceptions and Limitations.\_\_ For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions. |
|  |  |
|  | 3. \_\_Term.\_\_ The term of this Public License is specified in Section 6(a). |
|  |  |
|  | 4. \_\_Media and formats; technical modifications allowed.\_\_ The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material. |
|  |  |
|  | 5. \_\_Downstream recipients.\_\_ |
|  |  |
|  | A. \_\_Offer from the Licensor – Licensed Material.\_\_ Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License. |
|  |  |
|  | B. \_\_Additional offer from the Licensor – Adapted Material.\_\_ Every recipient of Adapted Material from You automatically receives an offer from the Licensor to exercise the Licensed Rights in the Adapted Material under the conditions of the Adapter’s License You apply. |
|  |  |
|  | C. \_\_No downstream restrictions.\_\_ You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material. |
|  |  |
|  | 6. \_\_No endorsement.\_\_ Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i). |
|  |  |
|  | b. \_\_\_Other rights.\_\_\_ |
|  |  |
|  | 1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise. |
|  |  |
|  | 2. Patent and trademark rights are not licensed under this Public License. |
|  |  |
|  | 3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties, including when the Licensed Material is used other than for NonCommercial purposes. |
|  |  |
|  | ### Section 3 – License Conditions. |
|  |  |
|  | Your exercise of the Licensed Rights is expressly made subject to the following conditions. |
|  |  |
|  | a. \_\_\_Attribution.\_\_\_ |
|  |  |
|  | 1. If You Share the Licensed Material (including in modified form), You must: |
|  |  |
|  | A. retain the following if it is supplied by the Licensor with the Licensed Material: |
|  |  |
|  | i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated); |
|  |  |
|  | ii. a copyright notice; |
|  |  |
|  | iii. a notice that refers to this Public License; |
|  |  |
|  | iv. a notice that refers to the disclaimer of warranties; |
|  |  |
|  | v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable; |
|  |  |
|  | B. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and |
|  |  |
|  | C. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License. |
|  |  |
|  | 2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information. |
|  |  |
|  | 3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable. |
|  |  |
|  | b. \_\_\_ShareAlike.\_\_\_ |
|  |  |
|  | In addition to the conditions in Section 3(a), if You Share Adapted Material You produce, the following conditions also apply. |
|  |  |
|  | 1. The Adapter’s License You apply must be a Creative Commons license with the same License Elements, this version or later, or a BY-NC-SA Compatible License. |
|  |  |
|  | 2. You must include the text of, or the URI or hyperlink to, the Adapter's License You apply. You may satisfy this condition in any reasonable manner based on the medium, means, and context in which You Share Adapted Material. |
|  |  |
|  | 3. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, Adapted Material that restrict exercise of the rights granted under the Adapter's License You apply. |
|  |  |
|  | ### Section 4 – Sui Generis Database Rights. |
|  |  |
|  | Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material: |
|  |  |
|  | a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database for NonCommercial purposes only; |
|  |  |
|  | b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material, including for purposes of Section 3(b); and |
|  |  |
|  | c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database. |
|  |  |
|  | For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights. |
|  |  |
|  | ### Section 5 – Disclaimer of Warranties and Limitation of Liability. |
|  |  |
|  | a. \_\_Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.\_\_ |
|  |  |
|  | b. \_\_To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.\_\_ |
|  |  |
|  | c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability. |
|  |  |
|  | ### Section 6 – Term and Termination. |
|  |  |
|  | a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically. |
|  |  |
|  | b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates: |
|  |  |
|  | 1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or |
|  |  |
|  | 2. upon express reinstatement by the Licensor. |
|  |  |
|  | For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License. |
|  |  |
|  | c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License. |
|  |  |
|  | d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License. |
|  |  |
|  | ### Section 7 – Other Terms and Conditions. |
|  |  |
|  | a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed. |
|  |  |
|  | b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License. |
|  |  |
|  | ### Section 8 – Interpretation. |
|  |  |
|  | a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License. |
|  |  |
|  | b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions. |
|  |  |
|  | c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor. |
|  |  |
|  | d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority. |
|  |  |
|  | > Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the “Licensor.” Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at [creativecommons.org/policies](http://creativecommons.org/policies), Creative Commons does not authorize the use of the trademark “Creative Commons” or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses. |
|  | > |
|  | > Creative Commons may be contacted at creativecommons.org |