University of Colorado Boulder

Project Document

MCEN 5115 Graduate Project

Authors: Andreas Brecl

December 8, 2022

A. Introduction

For my project I decided to create a program that displays astronomy information on an LCD for the Sun, Moon, Mars, Saturn, Jupiter, and Venus. The information displayed with be the rise and set times for all of these celestial objects and the date of the next new moon and full moon. The purpose of doing this project is to help me know when to expect celestial objects when I go and do astronomy at night. Knowing the rise and set times helps you determine how close and how clear the planet is to Earth, along with knowing where to look for it in the sky at different points in the night. Knowing when a new moon or full moon occurs helps you determine how dark the night sky will be due to illumination from the moon. The position of celestial objects can be calculated via equations due to the predictable motions of the planets, Moon, and Sun. This means that the only two outside factors we need to get these desired items is the location of observation and the current date.

B. Electrical Design

While this project was primarily software heavy, there was still some electrical design that needed to occur to make the display work. Three primary electrical elements were needed. The first being a Raspberry Pi 3 B+, second a logic level shifter, and third a LCD2004 module. The purpose of the Raspberry Pi is to control the software logic and run calculations for the desired astronomy data. Along with this, the Raspberry Pi would be used for it's I2C capabilities via GPIO to communicate with the LCD display. The LCD2004 that I purchased has a built in potentiometer to control the screen contrast and a shift register chip that is compatible with the I2C communication protocol. Though both the LCD and Raspberry Pi have I2C, the logic levels are not the same. The LCD operates at a 5V logic level and the Raspberry Pi at a 3.3V logic level. To get these to operate at the same level, a logic level shifter was used to allow for these two systems to operate at the same voltage. Below is a wiring diagram of how this was accomplished.

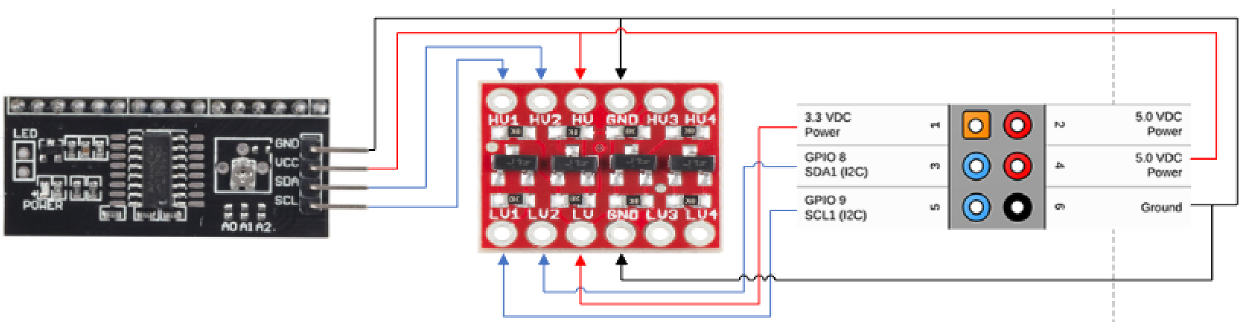

Figure 1: Electrical Diagram

C. Software Design

For my software to operate, I incorporated two external libraries. The first library being PyEphem astronomy library. PyEphem allows you to calculate things like the location of celestial bodies and how it will appear to an observer standing in a specific position on Earth at a specific date and time. This library utilizes the predictable motion of the planets, Sun, and Moon to allow you to calculate their position at any time in the past or future. To make this library operate, I had to define the current date (pulled from the clock on the Raspberry Pi) and the coordinates of Boulder, CO (40.011, -105.265). With this, I can create an observer object using this library in which I can find desired information for each celestial body. Though this provides the information, these values are in UTC time which is not particularly useful for me. Because of this, all the values were converted to Mountain Time.

The second library I incorporated was the RPLCD library. This library allows me to configure my string messages in a way that can be sent via I2C to the LCD display. Once a message is sent via I2C, the message is held on the LCD until it is cleared via another command. This is useful as I can cycle through data without having to continuously send data out to the LCD. This also allows me to use sleep functionality in my code. Using these two methods, I can then use a process outlined in the figure below to structure the way my code will operate.

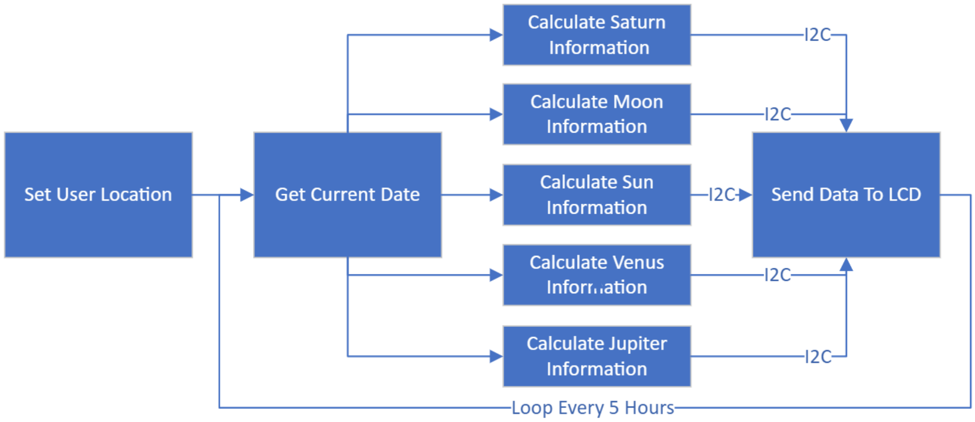

Figure 2: Simplified Software Loop

First, a set user location is defined in the software. This user location and current date is passed into a class that calculates all of the celestial object information. In this class, the rise and set time, next new and full moon, and conversion to Mountain Time is completed. Next, the system formats this data and sends the information to my LCD class. This class then sends the formatted information to the LCD screen. The LCD screen is set to cycle the information every 10 seconds. The system will then pull the current date every 5 hours to update the information.

D. Results

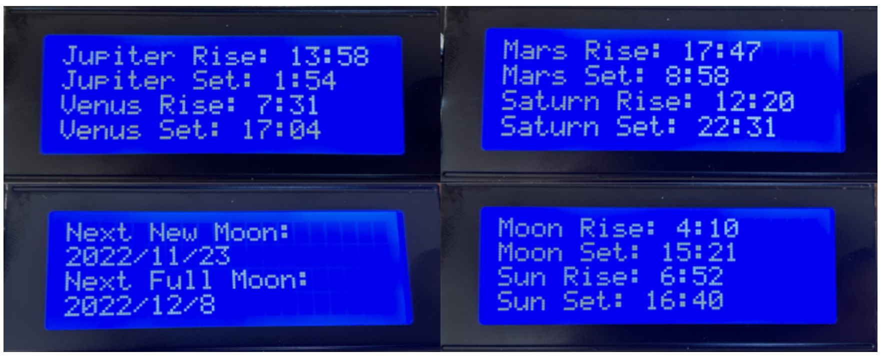

Figure 3: Images From Screen

Above is an example of the results after implementation of the software. This result was taken November 21, 2022. The screen behaved as expected where it continuously looped through these 4 screens pausing for 10 seconds for each. All of these results are accurate to within a few minutes of what actually occurred and what other online sources stated that they would be.

E. Conclusion and Future Improvements

In conclusion this project provided me with an easy way to get daily information on positioning of celestial objects. It is accurate to within a few minutes of the actual rise and set times. In the future, I would like to add the functionality to have the total percentage of the moon illuminated. Along with this, I potentially would like to switch the LCD screen for an OLED screen to make it look much nicer. Finally, adding a display on and off button would be useful as I do not need it constantly running.

Appendix A

Code Repository: `https://github.com/andreasbrecl/astronomy-lcd`
Video of Project: `https://youtu.be/1jpZKJqHv6k`