IIT Bombay

Electronic Design Lab

Laser Pointer for Stargazing Applications

Group Members:

Hitarth Choubisa
130070027

Shorya Consul
13D070027

Kumar Siddharth
13D070033

Supervisor:
Prof. Dipankar

Teaching Associate:
Astha Tyagi

Table of Contents

Abstract

The project is aimed at creating a "portable" planetarium for people, where a laser pointer can be directed to point at the star clicked upon on Stellarium on a computer. Stellarium is an open-source software which enables us to view the locations of stars on-screen based upon our location and time of day. The output of Stellarium will then be processed further to generate coordinates with respect to an observer at the specified location on the Earth's surface. This is then fed to a microcontroller, which generates the appropriate signals for the motor driver circuit. This will result in the motion of a couple of servo motors such that the laser pointer points in the intended direction. This will enable users to easily correlate the on-screen positions of the stars and heavenly bodies with their position in the night sky.

Objectives

- Designed an affordable, plug and play, portable planetarium
  The apparatus needs to be offer portability so that it can be easily carried by the user to whichever location he wants to use it at. Affordability is certainly a priority for this to be a feasible prototype.
- Accurate control of laser pointer
  The laser pointer should point at its intended target with minimal error. Large errors would render the apparatus ineffective.

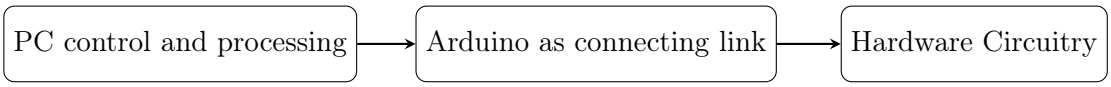
Introduction

The heavens have always been a source of wonder for Man. Yet, the charm of studying the stars is lost upon us when we cannot make sense of the arrangements and movements of these objects. With their constantly changing positions, dedication on the part of the observer is needed to truly be able to fathom these arrangements.

On the other hand, we have the open-source software, Stellarium. This freely available planetarium gives the user a view of the sky in real-time. It has numerous features to identify stars, plaents and other celestial bodies. The newer software builds are also equipped with a Telescope Control plug-in. This has been configured for some off-the-shelf telescopes. However, these are very expensive (upward of 60000 rupees) and hence, out of reach for most amateur enthusiasts. Moreover, a telescope is something that can only be used by a single person at a time, thereby rending such an expensive apparatus impractical for star-gazing in groups.

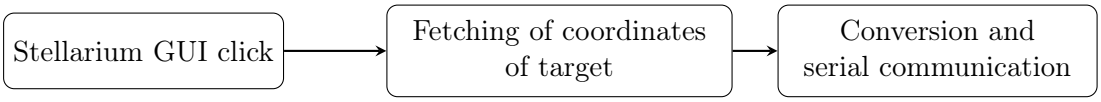
This is where we are attempting to step in. The Telescope Control plug-in gives us the facility of channelling the Stellarium output to a remote server. Using this, we have accomplished the control of a laser-pointer mounted on a pan-tilt mechanism through a microcontroller. A basic calibration mechanism has also been incorporated. The entire apparatus is several times cheaper than the aforementioned telescopes offered on the market and hence, much more affordable.

Block Diagrams



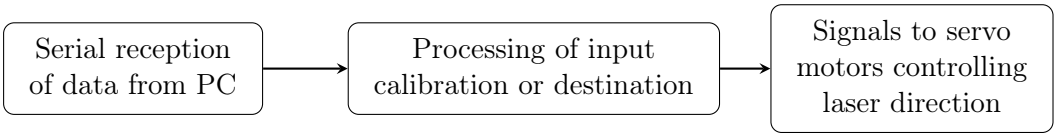
1. **PC control and processing**
   This block deals entails most of the software work and data processing. Can be divided into three sections:

   - When a user selects a star in Stellarium and commands it to be sent to the laser, our 'telescope server' script interacts with it and collects the coordinates of the concerned star in the RA/DEC coordinate system.
   - Another function in the same script converts the coordinates to the ALT/AZ coordinate system. These coordinates are the ones that make sense for an observer standing on the earth, only ALT/AZ makes sense. ALT/AZ conversion depends on one's location and time of observation.
   - The converted coordinates are serially communicated to the Arduino for further action.



2. **Arduino as connecting link**

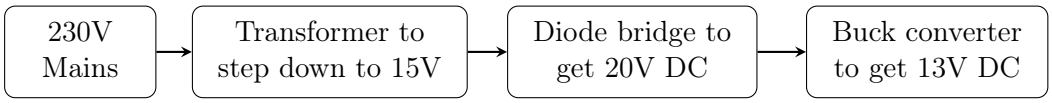
   - The Arduino receives data from the PC serially.
   - The Arduino generates the requisite signals for the servo motors (controlling the laser mount) to move.
   - It is powered by the USB cable attached to the PC (also used for serial communication).



3. **Hardware circuity**
   This subpart of the abstraction refers to the actual hardware behind the working the apparatus. It constitutes all the components that physically move or generate the appropriate voltages and currents. This comprises of:

   (a) 12V battery
   - Powers both the servos and the laser after appropriate voltage level conversion
   - A rechargeable battery used as the user should not have to replace such a bulky battery after each round of use.

   (b) Battery charging circuitry
   - Connected to the 230V mains
   - Provides a constant voltage to charge the battery



   (c) Laser and motor drivers
   The 12V voltage supply from the battery is stepped down to 5V via a linear regulator. A buck converter would have been preferable due to its higher efficiency but due to a lack of availability at the time, it was not used. The motors were fed this as their power supply while a pair of resistors were put as intermediates to give the laser an input of around 3V.

Circuit Diagram

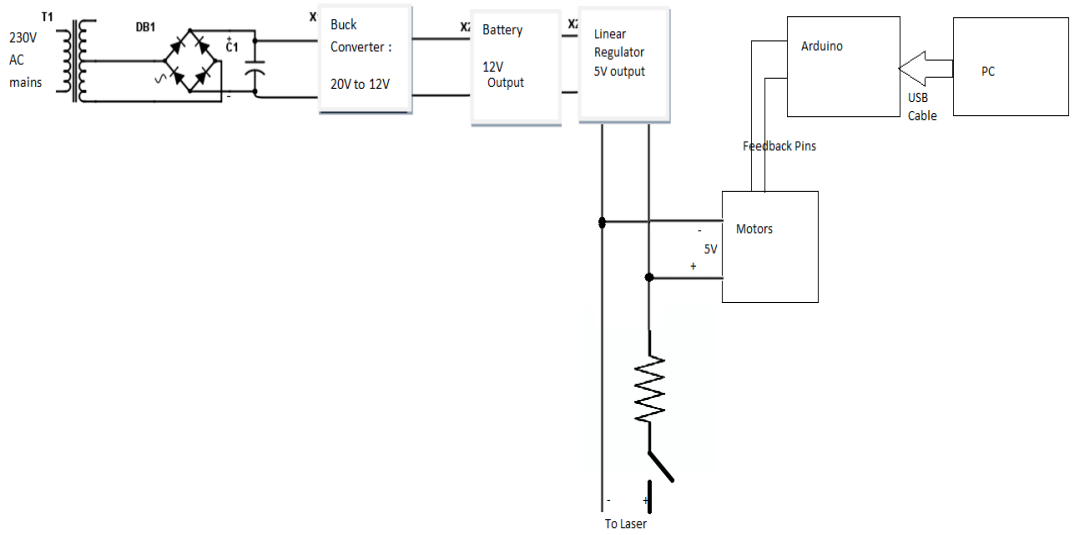


Figure 1: Circuit Diagram

Components Used

Component	Quantity	Price (if applicable)
Arduino	1	500
12V lead-acid rechargeable battery	1	350
5mW green laser pointer	1	325
Servo motors + mount	1	500
230V to 15V transformer	1	-
LM7805	1	-
5mW 10Ω resistor	2	-
Buck converter	1	-
Diode bridge	1	-
Heat sink	1	-

The laser pointer was purchased online on Amazon. The product name as listed on Amazon.in is *5mW 532nm Astronomy Mid-open Green Beam Light Laser Pointer Pen Class 3A Black*. The Arduino was also purchased over Amazon. The pan tilt mechanism along with the servo motors were sourced from RoboKits. The 12V lead-acid battery was purchased from Gala on Lamington Road. The components without prices listed were sourced from WEL lab.

Apparatus

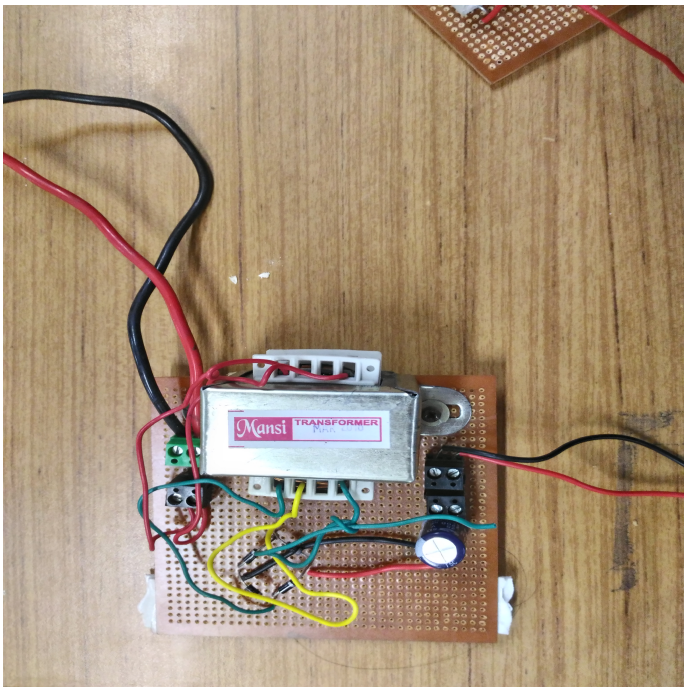


Figure 2: Battery charging circuit

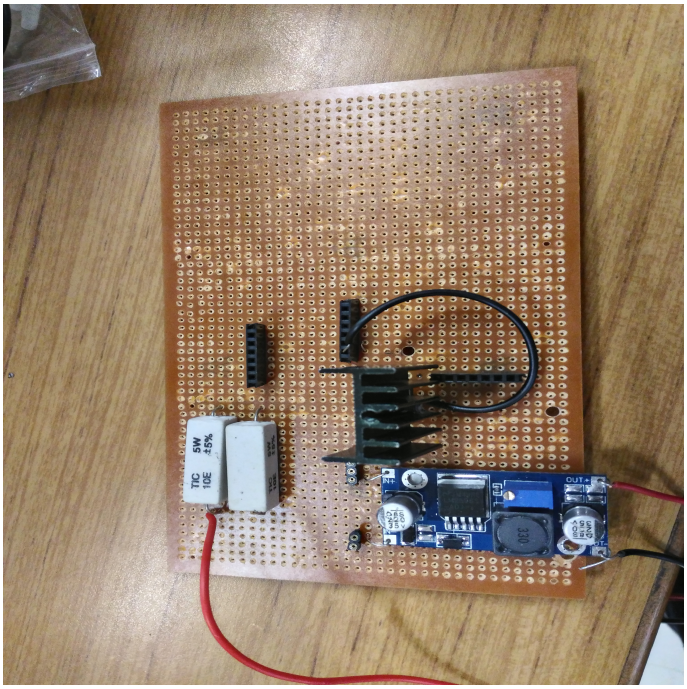


Figure 3: Circuit to driver motors and laser

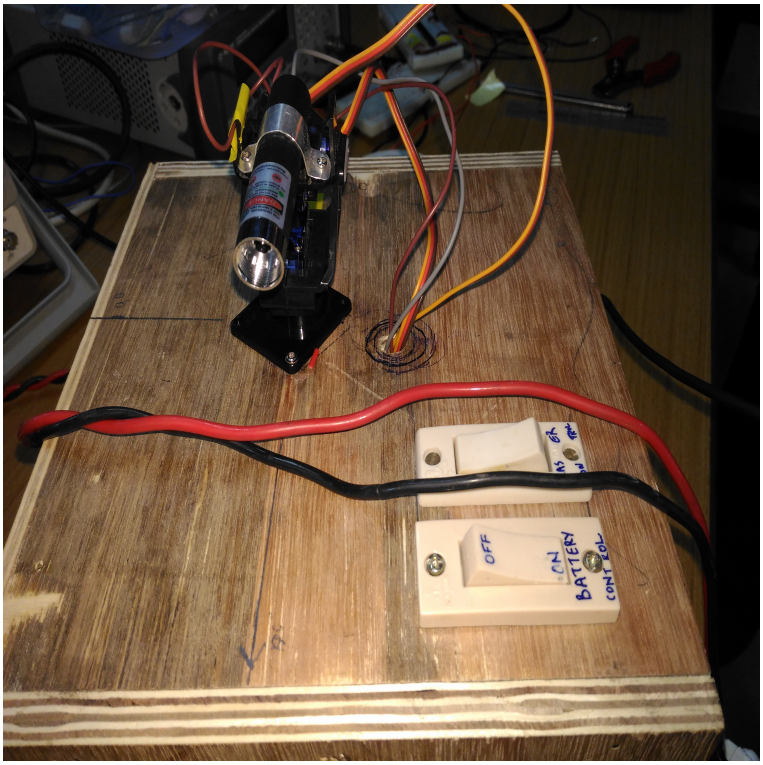


Figure 4: Assembled box

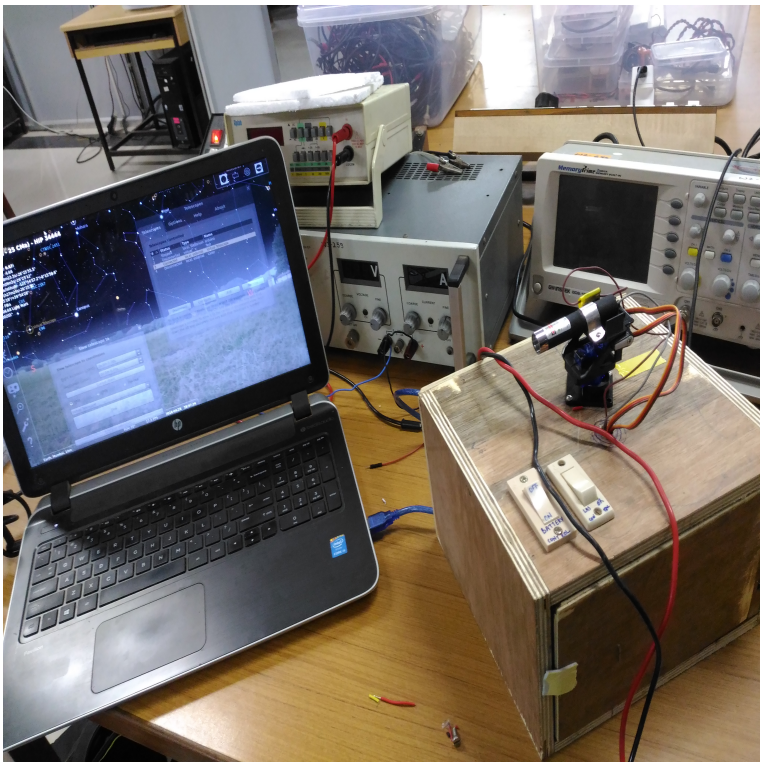


Figure 5: Complete set-up

Results

We completed the design and construction of a prototype of our product. This was taken for field testing on the sixth floor terrace of Hostel 13 at around 11.30pm on 28th April 2016. Several stars and planets were visible at this time. The mechanism was calibrated with respect to Jupiter. After this, the laser was made to point at several stars and planets such as Alkaid, Mizar and Mars. An error of several degrees in both the the altitude and azimuthal directions was observed. Moreover, this error did not seem systemic (regular).

However, this does not seem to arise from the implementation. Rather, the servo motors seem to have an inherent inaccuracy. Although the reason has not been exactly ascertained, some runs within the lab seem to back this up. We believe that a more precise set of servo motors would likely deliver better accuracy but at the cost of an increase in the price of the prototype.

It is worth noting, however, that despite this inaccuracy, the mechanism served as a fair guide to an amateur regarding celestial positions. The laser beam itself is an excellent guide for any observer to sight the targeted star or planet. Moreover, the entire apparatus is much cheaper than the corresponding telescopes currently available on the market.

Problems Faced

- Understanding how Stellarium interfaced with telescopes using its default Telescope Control plug-in was difficult. We wanted to build our product based on this plug-in as it would enable general users to use it with minimal configuration of their PCs.
- Serial communication of characters interfered with the normal functioning of the background "Telescope Server" Python program. This was worked around by sending strings instead.
- Current drivers to drive the laser module proved to be too bulky to design. Given that there is already a driver circuit within this module, we compared the performance of the laser with both current driver circuits and voltage sources. The latter gave a much brighter beam consistently.
- The buck converter used to generate the 5V needed to power the laser and motors seemed incapable of withstanding such high loads; the output would fall sharply upon loading. This problem was not observed with another buck converter. However, due to the unavailability of another buck converter, a linear regulator was used instead.
- Initially, it was intended that calibration be done with respect to any

star. However, a flaw was spotted in such an implementation as stars "slew" across the night sky with the progression of time. Hence, such calibration would lead to gradually increasing offsets for the user. Polaris remains approximately fixed in the sky, making it an ideal calibration point. Note that despite this, the present implementation allows recalibration at any point with respect to any star.

Road Ahead

There is plenty of room for improvement on this initial prototype. Some of these have been listed below.

- Two-way communication between the Arduino and Stellarium could be implemented to enable the reticle on Stellarium to more accurately reflect the progress of the laser pointer.
- Scripts that combine audio commentary with automated motion of the laser pointer could be made to serve as demonstrative tools for novice users.
- Three-point calibration may be implemented. This would allow for the system to be used on inclines as well.
- More precise servo motors could be used to improve accuracy. Depending on the accuracy provided by these motors, finer calibration could also be achieved.
- A more efficient battery charger could be designed. Although this is not central to the prototype itself, the current circuit is rudimentary in that it is simply a constant voltage source. A charger that is better aligned to the charging profile of the battery would increase the efficiency of the charging circuit.

References

1. **Stellarium Wiki:**
`http://www.stellarium.org/wiki/index.php/Telescope_Control_plug-in`
2. **Python GUI Programming:**
`http://www.tutorialspoint.com/python/python_gui_programming.htm`
3. **Serial communication between Python and Arduino:**
`http://stackoverflow.com/questions/24074914/python-to-arduino-serial-read-write`

4. **Telescope Control with Arduino and Stellarium:**
`http://yoestuveaqui.es/blog/telescope-control-with-stellarium-and-arduino/`

5. **Horizontal Coordinate System:**
`https://en.wikipedia.org/wiki/Horizontal_coordinate_system`

Code

The code has not been reproduced here in the interests of saving paper. However, these have been uploaded on YAWP. An additional README file has also be uploaded with a link to a repository containing the relevant codes (Python and Arduino). The README file would also give list the details of operation of the code when loaded on the relevant hardware (PC or microcontroller).