POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering

Master's Thesis

Prototyping of Energetically-Autonomous UWB Anchors for Rover Localization in Lunar Environment

Supervisors
Prof. Marcello Chiaberge
Dott. Marco Ambrosio

Candidate
Michelangelo Levati

Dicember 2022

To myself
"Believe in your dreams"

Abstract

The space race began in the 1950s, the world's great powers competed to reach farther and farther frontiers, mainly for political reasons related to the Cold War. From launching satellites into orbit to putting space stations into orbit, from the first man on the moon to conquer the Solar System. On July 20, 1969, Neil Armstrong and Buzz Aldrin left the first footprint on the Lunar surface, while Eugene Cernan, Ron Evans, and Harrison Schmitt were the last humans to leave Earth orbit in December 1972. To this day, the desire to return to the Moon continues, and more and more missions are being planned and completed for the grand return.

This thesis has its roots in an expedition to the Lunar surface during which a mobile rover must be able to pinpoint its location relative to a Lunar base. An Ultra-Wide Band Anchor network is chosen for localization. This technology uses wide-band radio waves to transmit information, which can be exploited for high-precision real-time localization by triangulating at least three signals. The greater the number of Anchors, the greater the accuracy as noise effects are reduced. The network consists of several Anchors, each equipped with a UWB module, to be placed at strategic locations with the rover itself so that it can communicate at any time with at least three Anchors. Thus, the goal is to prototype a series of Anchors, devices capable of working autonomously after placement on the Lunar ground.

The work was divided into four macro areas, including the identification of the best power supply system for the devices, the mechanical design of the Anchors, the electrical design of the Motherboard to manage the systems, and the distribution of the Anchors on the lunar ground. The four design phases were carried out simultaneously to ensure that they matched. For the power supply, a system was designed consisting of five solar panels and five Li-Po batteries, which ensure standby operation of the device even during the period of no light; the design was completed to minimize the size during transportation; the Motherboard is responsible for managing the charging system and powering up the UWB module under specific conditions; and for the method of distributing the Anchors, on the other hand, a code was developed in Python based on QR code recognition to identify the device and pick and place through a six-joint mechanical arm. In addition, the device was realized with the ability to close up to be moved, so that multiple missions of the same type could be carried out in different locations with the same Anchors.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis is rooted in the planning of a lunar mission involving a rover, whose location needs to be identified. The absence of a GPS localization system on the Moon implies the implementation of a different localization method, and consequently, the objective of this thesis is to design the Anchors[1] that make up the network for the localization method described below.

The mission involves setting up a mobile base at the Moon's South Pole, and Anchors are to be placed by the rover itself at specific locations around the base. The fact that the base is mobile dictates that the developed Anchors can be relocated multiple times, to reduce the number of devices needed to cover all areas of interest (weight and space requirements are thus minimized).

Another necessary characteristic is that Anchors must be energetically autonomous so that they remain active for the duration of the mission. This characteristic is particularly because the mission does not involve an operator, and an accidental shutdown would be fatal to the devices.

There are many possibilities for developing a location network, but the focus is on Indoor Positioning Systems[2] (IPS). IPS networks involve the use of several

[1] An Anchor is a device with known fixed positions.

[2] IPS is a network of devices used for localization where satellite technologies such as GPS are inaccurate or unusable.

technologies, including:[1][2]

- Lights
- Radio Waves technology
- Magnetic Fields
- Acoustic Signals
- Behavioral Analytics

For radio technologies, there is a method called Ultra-Wide Band (UWB). It refers to a transmission technique created to send and receive signals using pulses of energy at radio frequencies with extremely short duration (from a few tens of ps to a few ns), leading to an extremely wide field of view. Per the International Telecommunication Union Radio-communication Sector has defined as UWB a transmission system with a frequency strictly greater than $500MHz$.

Normal radio waves use varying energy levels and frequencies, instead, the UWB technology takes advantage of generating signals during specific time intervals. Therefore, this is the mechanism that allows the signal to be transmitted with a minimum energy loss. This feature fits perfectly with prototyping a device as small in size and consequently with a limited energy range as those to be developed.

The Time of Flight[3] (ToF) can be used to determine the location with UWB. It is possible to determine the distance by figuring out how long it takes for signals to travel between the transceivers. There are many different methods for UWB positioning, including Time Difference of Arrival (TDoA), Two-Way Ranging (TWR-ToF), and One-Way Ranging (OWR-ToF). The UWB module designed for this mission by the PIC4SeR[4] uses the OWR-ToF method shown in Figure 1.1.

[3]Time of flight denotes the measurement of the time taken by an object, particle, or wave to travel a certain distance in a given medium.

[4]PIC4SeR - Interdepartmental Center for Service Robotics. Torino, Italy.

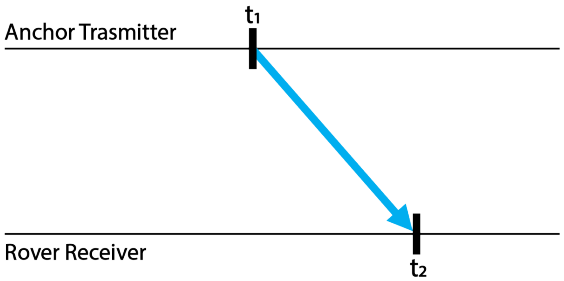
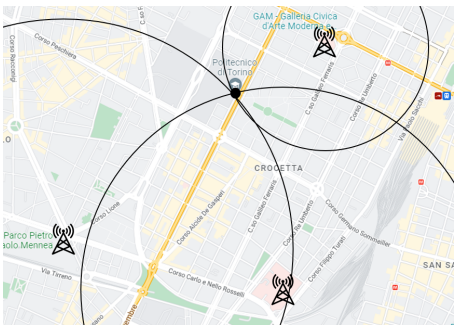


Figure 1.1: One-Way Ranging, ToF positioning techniques

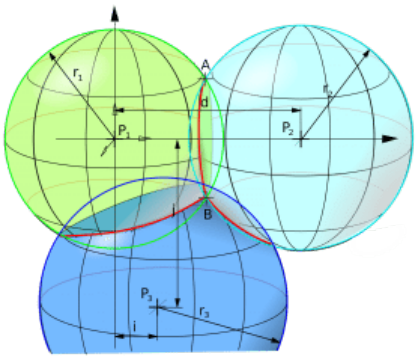
In OW-TOF mode is required perfect synchronism between the receiver and transmitter. The Anchor transmits toward the rover the information regarding the instant t_1 of successful transmission of the packet, termed Time Stamp. The rover receives the packet at instant t_2 and can calculate the time of flight of the signal and, consequently, the distance to the Anchor. The distance measurement improves as much as the synchronization system is more accurate.[3]

By triangulating at least three of the Anchor signals received by the rover it is possible to uniquely identify the rover's location (a mechanism similar to that used with GPS as shown in Figure 1.2b).

In particular, the distance measured between an Anchor and the rover corresponds to the circumference radius that can be drawn for each transmitter; intersecting three of them gives the exact location of the rover, as can be seen in Figure 1.2a. Using more than three transmitters allows the system to verify the position identified with the first three, consequently, the accuracy is increased.



(a) Triangulation in 2D



(b) Triangulation in 3D[4]

Figure 1.2: Triangulation process

In the indoor environment, some localization errors may be incurred due to the reflected waves of UWB signals against walls and obstacles, but through the use of specific algorithms, it is possible to filter the information to avoid this. In contrast, in the Lunar environment, the use of UWB transmitters is perfect, because the presence of obstacles that can produce the reflection effect is very small; if filtering algorithms and the presence of multiple transmitters with which verifications can be made are also considered, excellent levels of localization accuracy can be achieved.

After all these considerations, the goal of this thesis is to develop the energetically-autonomous Anchors prototype of the UWB network capable of locating the position of the rover in the Lunar environment with high accuracy. The work is divided into the following four macro areas: the identification of the best power supply system for the devices, the mechanical design of the Anchors, the electrical design of the Motherboard to manage the systems, and the distribution of the Anchors on the lunar ground.

Because of the structural features of the developed device, the name assigned to the Anchors is HasuNoHana[5].

[5] "*Hasu no hana*" means lotus flower in Japanese.

Chapter 2

Power Supply System

The starting step in being able to design a device that can work autonomously is to identify an appropriate energy source that can power the entire system for a relatively unlimited period.

To do this, it is also necessary to know how much power is required to power the whole thing.

This power system design phase is carried out, at first, on par with what is explained in detail in Chapters 3.

The UWB module's power requirement of about $660mW$ is estimated when operating at maximum capacity. To this must be added the power required by the Motherboard described in Chapter 4, it is initially capped at $25mW$.

In this chapter, the choice made for the power system to meet the imposed power limits is justified, and all the tests demonstrating its actual functionality are shown.

2.1 Energy Consumption Analysis

As preannounced, the UWB module requires $660mW$ of power to run at full capacity and $25mW$ for the power needed by the Motherboard. Taking into consideration the possibility of power losses that may incur the value considered as a ceiling is set at $700mW$. If it is to consider energy, it is possible to use Formula 2.1.

$$E = P \cdot \Delta t \quad [Wh] \tag{2.1}$$

As a result, $700mWh$ ensures operation in the operating mode, every hour. During the standby mode, instead, $25mWh$ is required every hour.

2.2 Charge Storage System

Therefore, the goal is to find find a power source that can manage to provide the energy needed to be able to operate for as long as possible. Another limitation imposed is also to miniaturize it to avoid excessive bulk in the transportation phase.

In the zero version of the anchors used as the basis for this project, a $650mA$ single-cell lithium polymer battery (simply called Li-Po battery) with a nominal voltage of $3.7V$ was used. Such a battery is capable of providing about $2.4Wh$, which provides an operating time of about 3 hours and 20 minutes.

An interesting feature of Li-Po batteries is that they are rechargeable batteries; in fact, they are used in all modern devices such as cell phones, drones, radio devices, etc. where weight and size are key issues.

Therefore, the idea is to use Li-Po batteries and develop a battery recharging system, to have the ability to store the energy produced and manage the time of use according to how much can be produced while trying to maximize it.

As mentioned earlier, this part of the design is carried out in conjunction with what is explained in Chapter 3 and taking into account what said about Li-Po batteries. Due to the inherent characteristics of the batteries themselves[5], it is evaluated to continue using the same type of battery in use for prototype zero ($650mAh$ charge at nominal voltage $3.7V$).

However, to have more charge available and to lengthen the operating time, it is decided at first to use five Li-Po batteries in parallel. This allows for the same rated voltage output from the batteries, but with a quintupled charge of $3.25Ah$. Five batteries with these characteristics are the minimum required to balance the power dissipation during the period without light.

Connecting batteries in parallel, however, could lead to premature battery wear

since the output currents may not be drawn equally from each one. This would also create imbalances in the charging system and possibly energy waste.

In addition, this solution does not allow the system to keep track of how much charge is left, and in case it runs out it would lead to an inevitable shutdown of HasuNoHana an alternative solution is needed.

The idea is to still use a five Li-Po battery system switching from one to the other when one is low as shown in Figure 2.5, rather than connecting them in parallel. This allows for five charging steps that each correspond to 20% of the total storable energy. As for batteries, they will be addressed in all chapters by numbering them in ascending order from 0 to 4.

Switches placed between each battery and the load can be used to do this. These will be managed by the Motherboard as explained in Section 4.3.

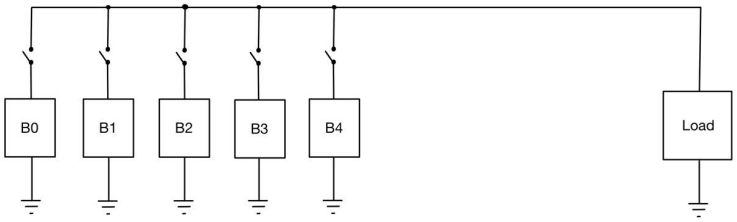


Figure 2.1: Graphic representation of the system consisting of five interchangeable batteries connected to the same load

2.3 Battery Recharging System

Established the method of energy storage, it is necessary to define how to produce it. On the Moon, for small and mobile devices like HasuNoHana, the best method to produce energy is definitely through solar energy.

The fact that the Moon does not have an atmosphere ensures greater solar irradiation on the surface than on the Earth, and this ensures that more energy can be produced. However, it must be kept in mind that the Synodic Month[1] has

[1]The Synodic Month corresponds to the time it takes the moon to make a revolution around itself and realign its position about the Earth and Sun.

a duration of 29 days 12 hours 44 minutes approximately and that any one point on the moon is irradiated for only half the month.[6]

Photovoltaic panels (PV) can be used to convert solar energy into electricity.

The absence of an atmosphere is also an advantage against the dust, that can not be deposited on the PVs. Since there is no wind, any deposits can only be due to external factors such as the approach of the rover[7], but this will be discussed more in Section 3.2 related to the design of the device.

This part of the design is also done in conjunction with what is explained in Chapter 3, so it is tried to accommodate the need to produce as much energy as possible through the use of PV, but at the same time use it in small size and number to try to have a versatile and small device.

The compromise found is to use five PVs, and as a product is identified a type of PV with short-circuit current of $30mA$, open-circuit voltage of $4V$ measured with an irradiance of $1000W/m^2$, and size of 45x45mm. These characteristics make it possible to produce about $100mW$ at that irradiance.

2.4 Sun Irradiance on Moon Surface

Having chosen solar energy as the power source to charge the batteries, it is necessary to conduct an analysis of the Sun irradiance trend on the lunar surface, to be able to determine the orientation with the greatest amount of energy that can be produced and then to estimate the number of hours of activity that can be guaranteed about consumption described in the Section: 2.1.

2.4.1 Insulation & Irradiance Mathematical Models

Through the study of the position of the Sun in the Lunar sky, it is possible to derive a mathematical model that can describe the Solar irradiance and insulation for a given selenographic latitude and longitude and PV tilt angle[2], along the Synodic Month.[8]

[2]PV inclination concerning the lunar surface.

The Moon's rotational axis is inclined to its orbital axis by 6.69°, while its orbital plane is inclined to the ecliptic by approximately 5.15°. As a result, the angle between the lunar equator and ecliptic is nearly constant at 1.54°.[6]

Assuming for simplicity that the Moon's rotational axis is perpendicular to the ecliptic plane and its equatorial plane is coplanar with the ecliptic plane, it is possible to derive first the position of the Azimuth[3] and the altitude of the Sun, starting from the diagram in Figure 2.2.

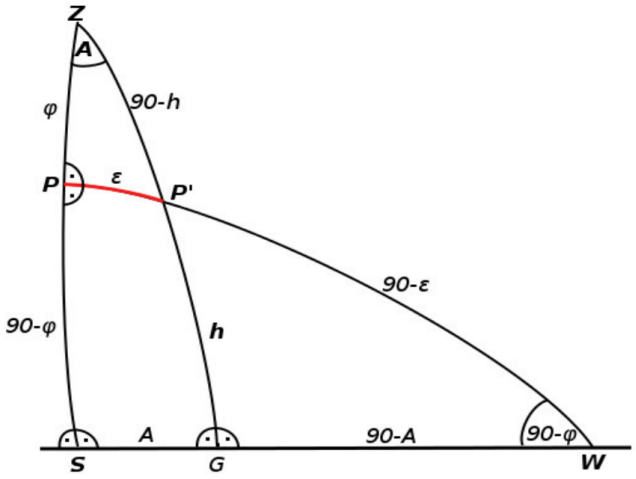


Figure 2.2: Graphic presentation of a suggested straightforward astrometric model for calculating the Sun's azimuth and elevation angle at a specific time and location on the Moon's surface. A - Azimut, Z - Zenith, S – South direction, W – West direction, P – Solar position at time t_0, P' - the Sun's position at time t', h – the Sun's altitude at time t'[8]

What is reported in Figure 2.2 refers to a system in which the observer is at selenographic latitude 0° and longitude φ>0°, and the problem becomes symmetrical for φ<0°.

From this graph, through trigonometric calculations, it is possible to derive the value of the azimuth coordinates, which are the altitude in Figure 2.2a and the Azimuth in Figure 2.2b, respectively.

[3]The Azimuth of a star is the arc of the horizon between a reference point and the vertical of the star itself. It is one of the two altazimuth coordinates, the other being the altitude.

$$A = arcsin(\frac{sin\,\varepsilon}{sin\,h}) \tag{2.2a}$$

$$h = arcsin(cos\,\varphi \cdot sin\,h) \tag{2.2b}$$

Expressing current day time (Δt) and Synodic Month (T_{SM}) in days, it is possible to evaluate the ε using the Formula 2.3 for every time instant.

$$\varepsilon = \Delta t \cdot 360 \cdot \frac{1}{T_{SM}} \quad [deg] \tag{2.3}$$

Having obtained the horizontal coordinates of the Sun for specific selenographic latitudes and longitudes, it can be calculated the angle of incidence[4] of the Sun's rays on a given plane expressed in the Formula 2.4.

$$\delta = arccos(sin\,h \cdot sin\,h_p + cos\,h \cdot cos\,h_p \cdot cos(A_n - A)) \tag{2.4}$$

As for the Total Solar Irradiance (TSI), it can be calculated from the data on the Moon's distance from the Sun (R) and Solar luminosity (L_{sol}) with the Formula 2.5 and it is obtained that it corresponds to $1363.03 W/m^2$.[9][6]

$$TSI = \frac{L_{sol}}{4\pi \cdot R^2} \quad [\frac{W}{m^2}] \tag{2.5}$$

Considering that δ can be carried out for any surface with any coordinates at any instant of time to obtain the irradiance (E_{dir}) during the Synodic Month, under these conditions, expressed in Formula 2.6.

$$E_{dir(t)} = TSI \cdot cos(\delta_{(t)}) \quad [\frac{W}{m^2}] \tag{2.6}$$

With the irradiance data, it is also possible to evaluate the insolation of the same plane, which is nothing more than the integral over time of the irradiance, obtained from Formula 2.7.

$$I = \int_{t_0}^{t_1} E_{dir(t)} dt \quad [\frac{kWh}{m^2}] \tag{2.7}$$

[4]The angle between the normal of a given plane and A direction of the incident sunlight.

By calculating the irradiance in discrete time with a time step Δt of the order of minutes, the Formula 2.7 can be approximated to the summation of the individual data per unit time, as expressed in the Formula 2.8.

$$I = \sum_{t_0}^{T_{SM}} E_{dir(t)} \cdot \Delta t \quad [\frac{kWh}{m^2}] \tag{2.8}$$

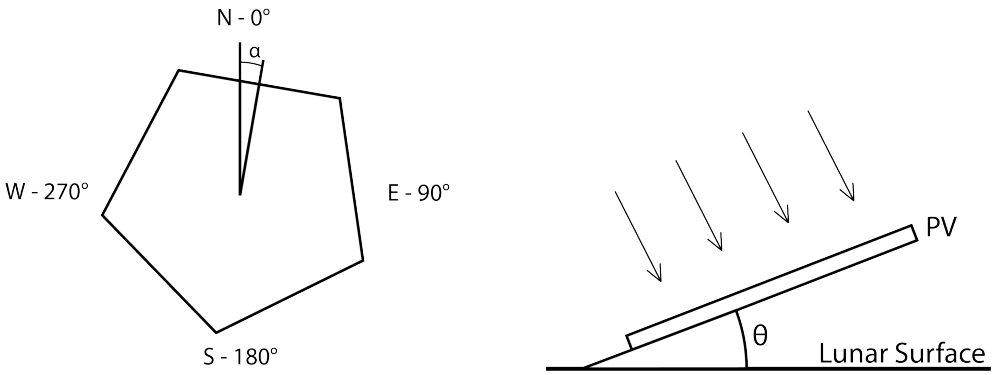
The code to be able to perform all the above calculations automatically is developed on "MATLAB version 2021a" and provided by Marcin Kaczmarzyk[5].[8]

2.4.2 Model Applied to HasuNoHana

Since HasuNoHana is designed to have an extensible spherical shape for the reasons described in Chapter 3, it is evaluated that the best arrangement of the PVs is by placing them one on each face of a pentagon.

Since the pentagon is an asymmetrical shape, it is necessary to determine what orientation the device should be placed on the ground and what tilt angle concerning the lunar surface is successful in producing the greatest amount of energy.

These evaluations can be made using the method described in Section 2.4 for different tilt angles θ and orientation angles α.



(a) Reference system used for PV orientation angle in North Hemifer

(b) Reference system used for PV tilt angle

Figure 2.3: Reference systems to evaluate sun irradiance and insulation with five PVs

[5]Mgr inż. Marcin Kaczmarzyk, Politechnika Rzeszowska, Katedra Budownictwa Ogólnego

Fixing the cardinal points on a circumference in which the reference system is described as follows: North corresponds to 0°, East to 90°, South to 180° and West to 270°; if the perpendicular of the face of one of the PVs forms on the circumference an angle α as shown in Figure 2.3a, then it follows that the vector containing the angles formed by all the perpendiculars of the faces will be that described in Formula 2.9.

$$orientation\_angle = [\alpha\ ;\ \alpha + 72\ ;\ \alpha + 144\ ;\ \alpha + 216\ ;\ \alpha + 288] \tag{2.9}$$

The reference system of PVs shown in Figure 2.3b, stipulates that the tilt angle θ of the PVs is 0° when the panel is parallel to the Lunar surface and 90° when it is perpendicular to the surface.

From Table 2.1 to Table 2.9 are the data calculated with the insolation model described in Section 2.4.1 with Formula 2.8, to identify for which tilt angle θ and orientation angle α the highest insolation occurs at different selenographic latitudes.

For the simulation, the parameters to be set to achieve sufficient accuracy to make the results reliable are evaluated.

The time step $\Delta\tau$ between the calculation of one position and the next of the Sun in the Lunar sky is set at 0.001 days, which corresponds to 1.44 minutes.

Regarding the orientation angle, it is chosen to set five steps of 15°, as the geometry of the device sees the panels displaced 72° from each other. Reducing the variation in angles between steps instead is considered unnecessary in retrospect given the results obtained.

As for the tilt angle, on the other hand, 0°, 30°, 45°, 60°, and 90° intervals are chosen. Again, given the results increasing the number of steps by reducing the variation would have yielded results indicating angles close to those identified with an insolation variation of no more than 1%.

As mentioned earlier the model is designed to make the evaluations in the Northern Hemisphere, therefore for the Southern Hemisphere, the results are symmetrical in the North-South and East-West directions.

	$\theta = 0°$	$\theta = 30°$	$\theta = 45°$	$\theta = 60°$	$\theta = 90°$
$\alpha = 0°$	1537,499	1383,947	1204,607	976,9646	473,2188
$\alpha = 15°$	1537,499	1383,948	1204,618	977,1976	496,8898
$\alpha = 30°$	1537,499	1383,947	1204,611	977,0276	486,6986
$\alpha = 45°$	1537,499	1383,948	1204,613	977,09	491,4457
$\alpha = 60°$	1537,499	1383,948	1204,616	977,1522	494,8459

Table 2.1: Total Solar insolation $[\frac{kWh}{m^2}]$ on PVs at varying tilt angle θ and orientation angle α relative to North of the first solar panel, at selenographic latitude $\Phi = 0°$

	$\theta = 0°$	$\theta = 30°$	$\theta = 45°$	$\theta = 60°$	$\theta = 90°$
$\alpha = 0°$	1514,142	1364,418	1188,986	964,4178	505,473
$\alpha = 15°$	1514,142	1364,523	1189,821	968,4527	511,8901
$\alpha = 30°$	1514,142	1364,68	1191,029	973,2052	507,4836
$\alpha = 45°$	1514,142	1364,658	1190,863	972,6107	509,2469
$\alpha = 60°$	1514,142	1364,489	1189,553	967,2372	510,8263

Table 2.2: Total Solar insolation $[\frac{kWh}{m^2}]$ on PVs at varying tilt angle θ and orientation angle α relative to North of the first solar panel, at selenographic latitude $\Phi = 10°$

	$\theta = 0°$	$\theta = 30°$	$\theta = 45°$	$\theta = 60°$	$\theta = 90°$
$\alpha = 0°$	1444,777	1306,788	1144,59	935,1309	546,6303
$\alpha = 15°$	1444,777	1307,06	1146,768	946,1149	547,9849
$\alpha = 30°$	1444,777	1307,468	1149,779	956,3204	547,0032
$\alpha = 45°$	1444,777	1307,411	1149,373	955,1176	547,3684
$\alpha = 60°$	1444,777	1306,972	1146,078	943,1627	547,7268

Table 2.3: Total Solar insolation $[\frac{kWh}{m^2}]$ on PVs at varying tilt angle θ and orientation angle α relative to North of the first solar panel, at selenographic latitude $\Phi = 20°$

	$\theta = 0°$	$\theta = 30°$	$\theta = 45°$	$\theta = 60°$	$\theta = 90°$
$\alpha = 0°$	1331,514	1213,227	1073,222	889,6277	592,5931
$\alpha = 15°$	1331,514	1213,852	1078,332	917,53	592,8263
$\alpha = 30°$	1331,514	1214,775	1084,463	930,8922	592,6562
$\alpha = 45°$	1331,514	1214,646	1083,681	929,4055	592,7186
$\alpha = 60°$	1331,514	1213,651	1076,791	913,0905	592,7809

Table 2.4: Total Solar insolation $[\frac{kWh}{m^2}]$ on PVs at varying tilt angle θ and orientation angle α relative to North of the first solar panel, at selenographic latitude $\Phi = 30°$

	$\theta = 0°$	$\theta = 30°$	$\theta = 45°$	$\theta = 60°$	$\theta = 90°$
$\alpha = 0°$	1177,793	1087,456	978,0903	882,9032	638,9411
$\alpha = 15°$	1177,793	1088,951	991,4099	892,1606	638,9731
$\alpha = 30°$	1177,793	1091,051	1001,74	901,3524	638,9501
$\alpha = 45°$	1177,793	1090,765	1000,552	900,2451	638,9586
$\alpha = 60°$	1177,793	1088,475	988,2258	889,5952	638,967

Table 2.5: Total Solar insolation $[\frac{kWh}{m^2}]$ on PVs at varying tilt angle θ and orientation angle α relative to North of the first solar panel, at selenographic latitude $\Phi = 40°$

	$\theta = 0°$	$\theta = 30°$	$\theta = 45°$	$\theta = 60°$	$\theta = 90°$
$\alpha = 0°$	988,2857	934,8861	890,4946	862,2659	681,8282
$\alpha = 15°$	988,2857	939,0322	902,0422	864,9867	681,8323
$\alpha = 30°$	988,2857	943,8489	911,3008	868,6371	681,8297
$\alpha = 45°$	988,2857	943,2413	910,2311	868,1507	681,8307
$\alpha = 60°$	988,2857	937,7983	899,2188	864,1332	681,8317

Table 2.6: Total Solar insolation $[\frac{kWh}{m^2}]$ on PVs at varying tilt angle θ and orientation angle α relative to North of the first solar panel, at selenographic latitude $\Phi = 50°$

	$\theta = 0°$	$\theta = 30°$	$\theta = 45°$	$\theta = 60°$	$\theta = 90°$
$\alpha = 0°$	768,7498	762,9211	815,1469	828,8617	718,2029
$\alpha = 15°$	768,7498	779,0118	817,5692	829,4872	718,2043
$\alpha = 30°$	768,7498	786,7445	820,7651	830,4095	718,2037
$\alpha = 45°$	768,7498	785,8817	820,3422	830,2812	718,204
$\alpha = 60°$	768,7498	776,4502	816,8136	829,2854	718,2042

Table 2.7: Total Solar insolation $[\frac{kWh}{m^2}]$ on PVs at varying tilt angle θ and orientation angle α relative to North of the first solar panel, at selenographic latitude $\Phi = 60°$

	$\theta = 0°$	$\theta = 30°$	$\theta = 45°$	$\theta = 60°$	$\theta = 90°$
$\alpha = 0°$	525,8558	634,5867	729,6262	784,0292	745,7744
$\alpha = 15°$	525,8558	636,3874	729,9439	784,1122	745,7757
$\alpha = 30°$	525,8558	638,7313	730,4161	784,2355	745,7751
$\alpha = 45°$	525,8558	638,4228	730,3501	784,2183	745,7754
$\alpha = 60°$	525,8558	635,8281	729,8412	784,0855	745,7755

Table 2.8: Total Solar insolation $[\frac{kWh}{m^2}]$ on PVs at varying tilt angle θ and orientation angle α relative to North of the first solar panel, at selenographic latitude $\Phi = 70°$

	$\theta = 0°$	$\theta = 30°$	$\theta = 45°$	$\theta = 60°$	$\theta = 90°$
$\alpha = 0°$	266,9841	505,8477	638,0173	729,1669	762,9465
$\alpha = 15°$	266,9841	505,9064	638,0282	729,1705	762,9477
$\alpha = 30°$	266,9841	505,994	638,0432	729,1739	762,9472
$\alpha = 45°$	266,9841	505,9817	638,0412	729,1736	762,9475
$\alpha = 60°$	266,9841	505,8875	638,0249	729,1697	762,9477

Table 2.9: Total Solar insolation $[\frac{kWh}{m^2}]$ on PVs at varying tilt angle θ and orientation angle α relative to North of the first solar panel, at selenographic latitude $\Phi = 80°$

What emerges from the simulation is that for selenographic latitudes closer to the equator the greatest insolation is obtained by arranging the PVs on the same plane as the lunar surface, whereas moving closer to the pole it is greater by increasing the tilt angle of the PVs. In contrast, the change in insolation is not particularly significant as the orientation angle changes.

However, the system for converting solar energy into electricity consisting of the set of PVs and MPPT devices described in Section 4.7 has limitations regarding the minimum power output from the PVs since the efficiency of the MPPT devices decreases if the input power is below a certain threshold. The orientation angle could therefore decrease the percentage of radiation useful for producing power.

A second simulation is carried out, on the data of greatest interest from the previous one (highlighted in green), in which the irradiance is filtered as useful only if it exceeded the threshold of $150 W/m^2$. The data from this simulation are shown in Table 2.10.

	$\alpha = 0°$	$\alpha = 15°$	$\alpha = 30°$	$\alpha = 45°$	$\alpha = 60°$
$\Phi = 0°$ & $\theta = 0°$	1469,816	1469,816	1469,816	1469,816	1469,816
$\Phi = 10°$ & $\theta = 0°$	1445,343	1445,343	1445,343	1445,343	1445,343
$\Phi = 20°$ & $\theta = 0°$	1372,517	1372,517	1372,517	1372,517	1372,517
$\Phi = 30°$ & $\theta = 0°$	1252,743	1252,743	1252,743	1252,743	1252,743
$\Phi = 40°$ & $\theta = 0°$	1087,940	1087,940	1087,940	1087,940	1087,940
$\Phi = 50°$ & $\theta = 0°$	879,283	879,283	879,283	879,283	879,283
$\Phi = 60°$ & $\theta = 60°$	798,647	791,190	780,563	781,130	793,875
$\Phi = 70°$ & $\theta = 60°$	753,564	747,380	736,746	737,498	749,591
$\Phi = 80°$ & $\theta = 90°$	735,705	730,376	722,788	723,881	732,287

Table 2.10: Total Solar insolation $[\frac{kWh}{m^2}]$ useful for energy production for different orientation angles α at selenographic latitude Φ and best efficiency tilt angle θ

From this second analysis, indeed, emerged that the insolation values turn out to be reduced compared to the previous ones by about 5-7%.

For design-related reasons, a limitation with the maximum achievable θ emerges; the reason is explained in Section 3.2. Therefore, the maximum allowed tilt angle is 45° according to what is evaluated in the simulations carried out; with this angle, there is about 81% insolation compared to the maximum obtainable with a θ of

90° for latitudes close to the poles, as can be seen in Tables 2.7-2.9.

This is because it is necessary to balance the energy production with the predetermined maximum size, moreover, with this solution the best ratio between energy produced and size is still obtained.

For this reason, the last three rows in Table 2.10 are to be replaced with those in Table 2.11.

	$\alpha = 0°$	$\alpha = 15°$	$\alpha = 30°$	$\alpha = 45°$	$\alpha = 60°$
$\Phi = 60° \ \& \ \theta = 45°$	780,716	770,906	756,017	755,126	775,335
$\Phi = 70° \ \& \ \theta = 45°$	693,505	685,257	670,254	670,666	689,162
$\Phi = 80° \ \& \ \theta = 45°$	600,406	593,518	580,194	580,820	597,053

Table 2.11: Total Solar insolation $[\frac{kWh}{m^2}]$ useful for energy production for different orientation angles α at selenographic latitude Φ and limitation of 45° to tilt angle θ

What is evident from the simulation is that for latitudes above 50, the best arrangement of the device on the ground is where the orientation angle is 0°.

If the insolation data of the individual panels are analyzed, it will be possible to see that one of the panels is not irradiated, but despite this, the device as a whole can produce more power than the other configurations. This is due to what is said about the minimum irradiance capable of actually producing power.

2.4.3 Energy & Max Operating Time Results

Having obtained the irradiance and total insolation data, an estimate can be made of how much energy the system is capable of producing.

As mentioned, the Synodic Month has a duration of 29 days 12 hours 44 minutes the diurnal period, and the period useful for energy production is approximately 14 days 18 hours 22 minutes. Combining the time data with the total insolation given in the Tables 2.10-2.11 and with the data of the PVs used (100mW produced at irradiance equal to $1kW/m^2$), it is possible to obtain the power produced over a Synodic Month at different latitudes. The results are shown in Figure 2.4.

By comparing this information with the HasuNoHana baseline consumption described in Section 2.1 (which corresponds to $25mWh$), an estimate can be made of how much excess energy is available and which can be used to activate the UWB module.

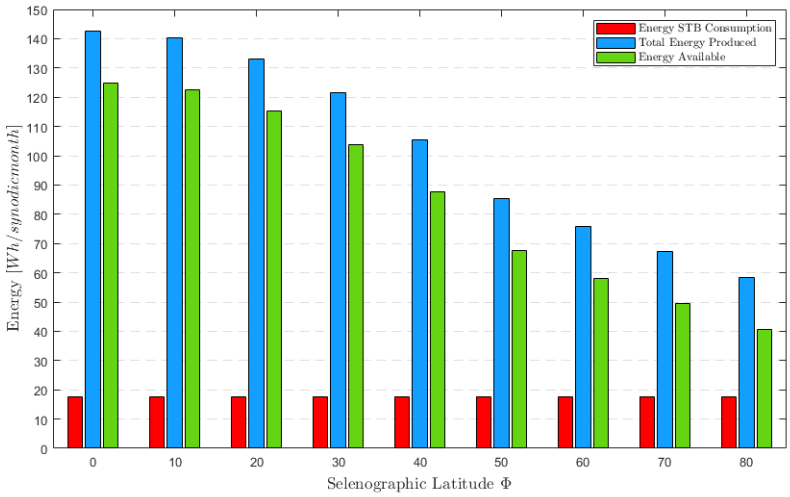


Figure 2.4: Energy analysis at different selenographic latitudes, with the best combination of orientation angle and tilt angle. Red bar - background energy consumption of the device during the standby state; Cyan bar - Energy produced by the charging system; Green bar - Energy available to power up the UWB module

Referring to the UWB module consumption data, which as mentioned in Section 2.1 is $660mWh$, it is possible to derive the times of use at different latitudes. The results are shown in Table 2.12.

	$\Phi = 0°$	$\Phi = 10°$	$\Phi = 20°$
Available Hours	189.16	185.57	174.87

	$\Phi = 30°$	$\Phi = 40°$	$\Phi = 50°$
Available Hours	157.26	133.04	102.38

	$\Phi = 60°$	$\Phi = 70°$	$\Phi = 80°$
Available Hours	87.89	75.07	61.39

Table 2.12: Maximum available amount of hours for different selenographic latitudes Φ with the best combination of orientation angle and tilt angle

2.5 Most Effective Recharging Method

As mentioned earlier, batteries are used one at a time by the system, so that the charge of one battery corresponds to 20% of the total charge.

For design issues also related to the circuitry explained in detail in Chapter 4, it is decided to separate the charging system from the board power supply, so that the power and energy generated by the PVs are used only for charging the batteries and not for powering the UWB module and other components of the device.

This choice is made based on possible power losses due to the components needed should it be chosen to use them also as power supply, and this would cause longer charging times not balanced by the gain in operating time.

For separation, it is decided to insert switches between the output of the charging system and the various PV batteries so that they can be connected to the charging system at the right time, as shown in Figure 2.5.

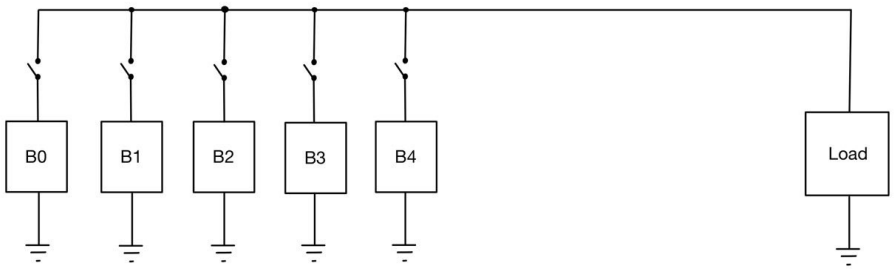


Figure 2.5: Graphic representation of the recharge system

However, recharging multiple batteries at the same time also involves some power loss, due to the absorption of the various switches.

For this reason, the best solution is to charge each battery individually, connecting it to the charging system only at the time when it is discharged and no other battery is connected to it.

The switches shown in Figure 2.5 do not correspond to the same ones shown in Figure 2.5, which are instead responsible for connecting the battery to the load. Through the use of two switches per battery, a standby state is achieved for those batteries that are not connected to either the load or the charging system.

2.6 Charge/Discharge Simulation

After the first mathematical analysis to define the maximum amount of hours available at different latitudes, it is considered necessary to conduct a simulation of the system to verify the actual validity of the results and to understand how the amount of hours obtained in Table 2.12 is distributed during the Synodic Month.

The simulation described in this Section is carried out through the "Simscape/Specialized Power System" library of Simulink and the model is divided into five main subsystems.

2.6.1 Simulation Description

The first subsystem is modeled to distribute the data obtained previously in Section 2.4.2, to simulate the real trend of solar irradiance on the various PV surfaces along the Synodic Month. Irradiance data relative to the selenographic latitude of 70° are chosen for the first simulations as shown in Figure 2.6.

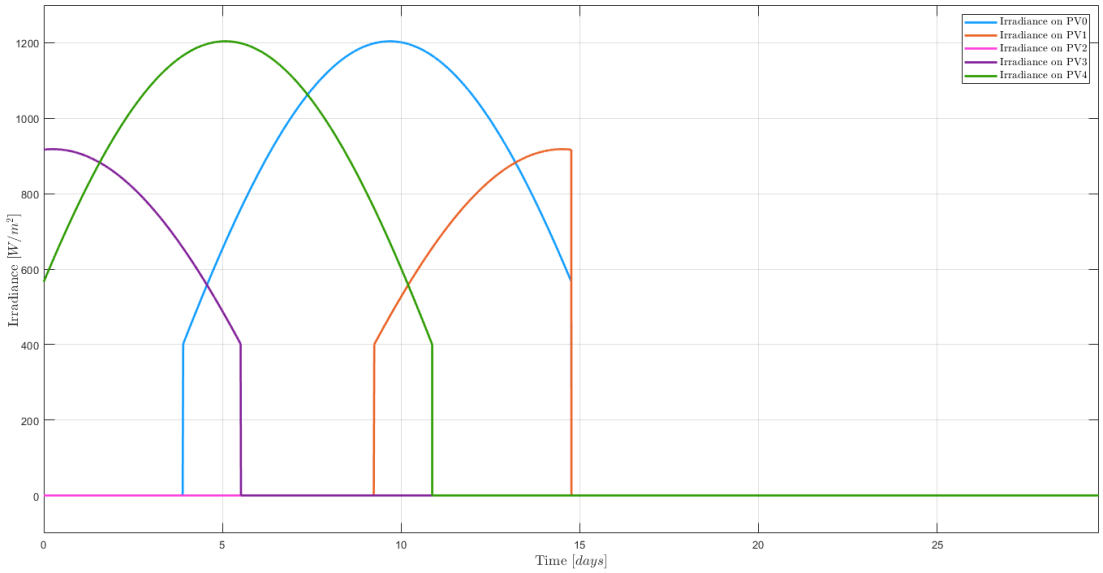


Figure 2.6: Irradiance trends on the five PVs at selenographic latitude $\Phi = 70°$ during the Synodic Month

With the data reported earlier in Table 2.11 it is also possible to calculate the average irradiance ($E_{average}$), which from Formula 2.10 turns out to be $1955 W/m^2$.

$$E_{average} = \frac{I}{\Delta t \cdot 24} \cdot \frac{\Delta t}{light\_days*} \tag{2.10}$$

The data for each PV is then sent as input for Simscape's functional blocks that are responsible for simulating power generation as irradiance changes. These blocks are modeled with the data from the panels that are going to be mounted on HasuNoHana, and tests are carried out using an oscilloscope to verify their correct operation so that the data obtained could be compared.

DC-DC converter blocks are then used at the output of the PVs that can regulate the output voltage to a level sufficient to ensure the charging of batteries connected to the charging system.

The output regulated by the DC-DCs is connected to the next subsystem, which are the five batteries and the switches used to connect them to the load or charging system. Again, as with the panels, data from the batteries chosen in the relevant functional blocks are set. The battery block can provide the battery status information, such as the voltage at which it is based on the remaining charge, and the current drawn or charging current flowing through it.

Using the information about the voltage of individual batteries, it is possible to determine when batteries are charged and when they need to be recharged. The subsystem that deals with adjusting the state of the switches uses a method best discussed in Section 4.2.1.

What this subsystem does, then, is to simulate the threshold comparators so that it has digital signals at its output that can be used in the block that simulates one of the functions implemented in the microcontroller, more specifically the one that handles precisely battery charging described in detail in Section 4.5.2.

The fourth subsystem is the one that simulates the function of the Microcontroller that handles the switches used to charge the batteries and those to connect them to the load. In addition, depending on the status of all five batteries, an output is generated to tell whether they can operate, or whether it is necessary to turn off the UWB module because they are discharged.

The active or standby state of the device is simulated using a variable load placed downstream of a DC-DC converter with a $3V$ output (they form the fifth subsystem) so that by going to change the load resistor a current can be imposed on it, and thus power is drawn, equal to those described in Section 2.1.

The recharging method described in Section 2.5 involves recharging the batteries individually, and for the simulation, it is to consider when it is possible to turn on the UWB module and when it is necessary to turn it off by the time required to recharge the batteries.

Considering the PV's' average irradiance, it is possible to determine how much time is required to recharge a battery, which corresponds to about 13.4 hours. The background consumption of the system is $25mWh$ and this implies that when on standby, a battery is capable of powering it for about 96 hours, which is vastly longer than the recharging time of a single battery (data are verifiable in simulations). This allows the batteries to be used to power the UWB module until at least one is fully charged.

Regarding the powering of the UWB module, on the other hand, the best solution in terms of energy efficiency sees the module turned on at the time when four of the five batteries are charged so that the one that is still discharged is recharged while the others are in use and consequently throughout the day there is always one battery connected to the charging system.

However, the choice can also be made based on the number of hours required to carry out operations with the rover. Or again based on how often it is needed to operate during daylight days.

No constraints are imposed on this data, so it is decided to run multiple simulations by varying the number of charged batteries needed to power the device between 3 and 5.

Another important detail to consider is the time needed for the last recharge. When fully charged, the batteries can keep the device alive for all the days of darkness, more precisely for 20 days. Considering the dark period to be as long as the light period as defined at the beginning, and considering that each battery corresponds to 20% of the total charge, at sunset, there is a need for all batteries

to be fully charged, to also ensure a residual charge at the time of the next sunrise. To do this in simulation is sufficient to use a time constraint in the subsystem responsible for turning on the UWB module.

Using the average irradiance figure obtained above, it is possible to estimate that it takes 11 hours to charge a battery, so for a full charge are required 54.6 hours, about 2.23 days. Therefore, it is sufficient to impose that 2.23 days before sunset the UWB module cannot be turned on even if the batteries are found to be charged. This figure can be verified a posteriori in simulation with the direct irradiance data.

In this chapter, it is decided to describe in detail the simulation which provides the UWB module activation as soon as four out of five batteries are charged since, as is previously noted, the optimal efficiency for no loss of energy production demands that a battery be linked to the charging system at all times. The data for the other two simulations are given in Appendix A and follow what is described in this chapter.

2.6.2 Simulation Results

The first check to be made is on the powers involved in the system, namely that produced and that absorbed. With this information, it is possible to understand when the UWB module is activated (consumption of about $700mW$) and when it is off because the batteries are charging and therefore the device is on standby (consumption of $25mW$).

These data can be seen in Figure 2.7.

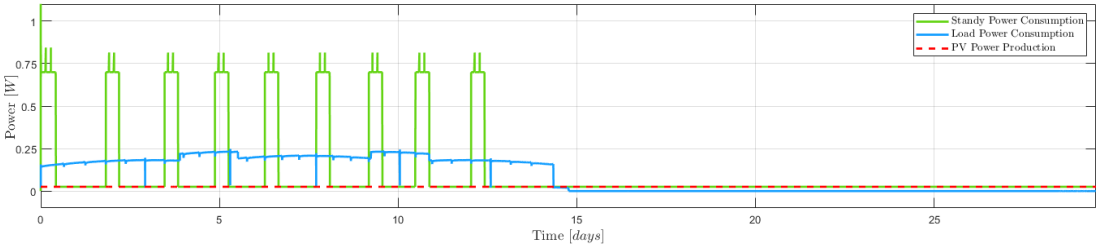


Figure 2.7: Power consumption and production trends along the Synodic Month. Simulation using 4 charged batteries to power on the UWB module

Beyond activation with the initial charge, it turns out that it is possible to activate the device eight times with a duration of about 9 hours each (for a total of 72 hours), interspersed with charging periods ranging from about 33 to 24 hours, which depend on the irradiance on the panels. Considering the residual charge inside one of the batteries, this data is in line with what is reported in Table 2.12.

One thing that may seem problematic is the power spikes that occur when switching from one battery to another. This problem is because the library switches function as current generators, which dictate the passage of a given current depending on its open/closed state.

In the implementation of the Motherboard, this problem does not occur because of what is described in Chapter 4.

With the power data, it is possible to verify what is shown about energies in Figure 2.4 shown earlier. By integrating the power over time, as the summation of the instantaneous powers for the sample time of the simulation, it is possible to obtain the total energy produced and consumed by the device during the Synodic Month as shown in Figure 2.8.

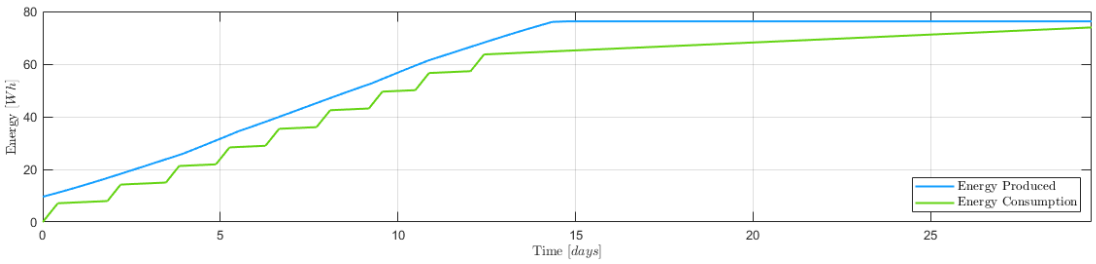
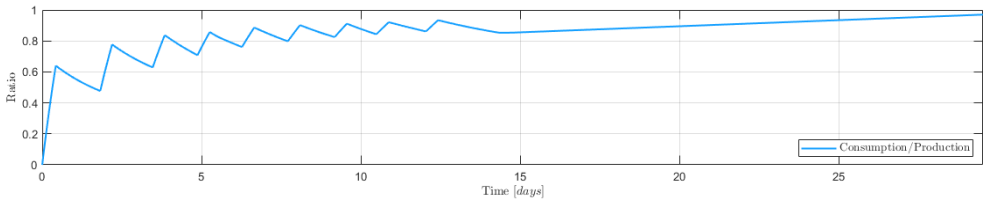
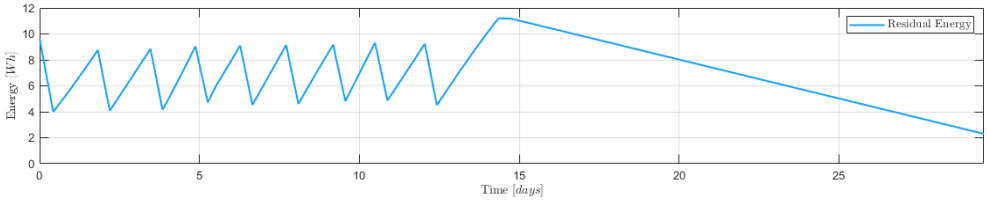


Figure 2.8: Energy production and consumption trends during the Synodic Month. Simulation using 4 charged batteries to turn on the UWB module

It is evident that the energy balance is in favor of the energy produced as it is always greater than the energy consumed, but it is nevertheless interesting to observe the trend of the ratio of energy consumption to production, as shown in Figure 2.9a, but also the trend of residual energy in the various batteries, as shown in Figure 2.9b, throughout the Synodic Month.



(a) Ratio of energy consumption to energy production during the Synodic Month



(b) Residual energy in the batteries during the Synodic Month

Figure 2.9: Energy-related output data. Simulation using 4 charged batteries to turn on the UWB module.

Another interesting thing to visualize is the trend of the percentage charge of the five batteries during the Synodic Month, shown in Figure 2.10. As can be seen, the initial charging of the batteries plans a battery in a low level of charge, to simulate the expected arrival time of the device from Earth to the Lunar base of about 4 days.

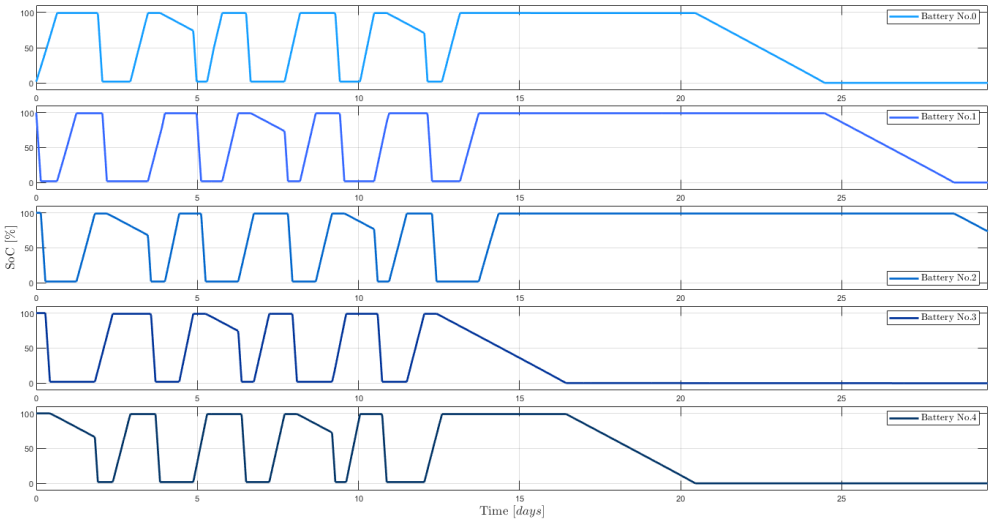


Figure 2.10: SoC trends of the five batteries during the Synodic Month. Simulation using 4 charged batteries to turn on the UWB module

During the dark days, the system is powered with the recharge made in the last 2.2 light days as described above. It can also be seen that at the end of the period of darkness, in which the device is in standby mode, there is a residual charge in one of the batteries, which is around 73%. Considering the times described so far, this charge is sufficient to ensure the life of the device until the other batteries are charged at the dawn of the second Synodic Month.

2.7 Final Considerations

What can be concluded from the results obtained is that using the charging system and batteries in the manner described, and in particular with the UWB module being turned on when the high level of charge of four batteries is reached, provides all the energy that can be produced with the PVs provided to be utilized and allows for sufficient working windows during the light days of the Synodic Month.

As mentioned earlier, other simulations are carried out that involves turning on the UWB module when the high level of charge of three and five batteries is reached. The results are given in Appendix A and what emerges from following the same guidelines for this simulation is that about powering up with five batteries required there is a drastic loss of efficiency in terms of hours, due to wasted energy at times when no battery appears to be connected to the charging system (at each activation).

While concerning turning on when the high threshold of three of the batteries is reached, we have that the amount of total hours is almost identical, what differs is the distribution of hours, which occurs more frequently, but for about 6 hours at a time, instead of 9 hours as in the simulation described here.

They both, therefore, turn out to be valid notes depending on the duration of the operations to be performed with the rover.

Chapter 3

Device's Prototyping

Simultaneously with the design of the supply system composed of five batteries and the battery charging system made of solar panels and managed by the Motherboard, the physical design of the anchors is carried forward, with some design constraints.

Prototype zero of the Anchors included a hemispherical base with a diameter of 11.5cm and a fixed rod 24cm high, on top of which is placed in the UWB module in a special container. No maximum size is imposed for this model, but an attempt is made to balance the size with the amount of energy that can be stored via batteries and the projection of energy via solar panels.

This is because there is the need to design a device as small as possible so that it does not have a large footprint during transport. At the same time, however, the UWB module must be placed at a height of at least 20cm, so that the beacon is higher than any obstacles and can operate properly.

Another thing taken into consideration during the design phase is the distribution of devices in designated locations. The placement described in detail in Chapter 5 involves the use of a mechanical arm mounting a gripper at the end.

To reconcile the requirements and what is said about the arrangement of the five solar panels in Section 2.4.2, it is decided to design the device by giving it a spherical shape when closed; with the possibility of telescopically extending part of it to ensure the position of the UWB module at a height of 25 cm and equipping it with five opening doors to protect the PVs during transport.

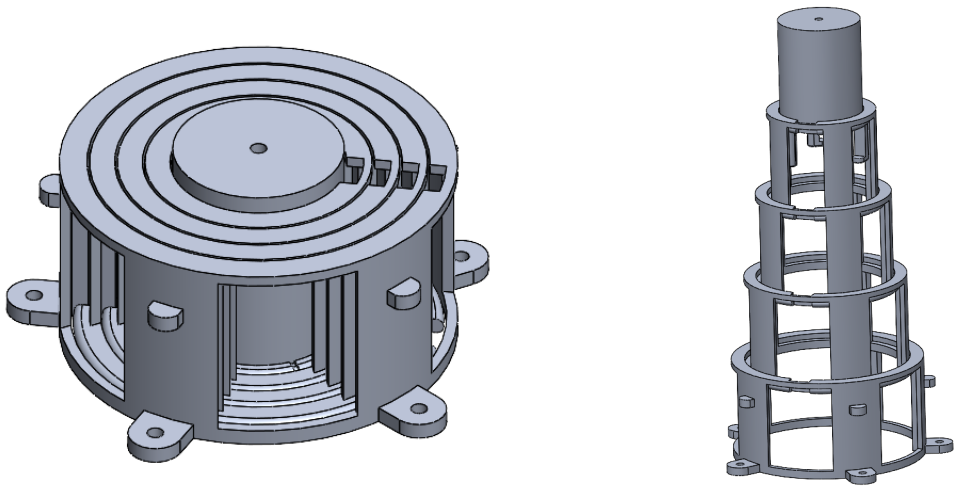
3.1 Telescopic Tower Development

As previously declared, the UWB module must be positioned at a height of at least 20cm for proper operation. The idea of using a telescope tower is born to meet this requirement and get a device that is at the same time more compact.

The tower is designed so that it can be extracted and retracted on itself, to reach 25cm of height at the maximum of its extension (compared to the base of the device). When closed it turns out to be a cylinder of 3.6cm in diameter and 4cm in height. The tower consists of five cylindrical stages, which for clarity have been numbered from 1 to 5 by assigning the lowest value to the outermost one.

Each cylinder has a height of 3.7cm, except for the last one which is 4cm. This is to allow the cables connecting the Motherboard to the UWB module to pass through.

Cylinder No.1 is attached to a base that prevents the other cylinders from going off-axis, which is in turn connected to the supporting structure of the device.



(a) Retracted configuration **(b)** Extended configuration

Figure 3.1: 3D model of the telescopic tower in the two configurations

For the prototype, the cylinders are printed with an Abs[1] filament extrusion 3D printer.

[1]Abs stands for acrylonitrile butadiene styrene, an amorphous thermoplastic material.

3.1.1 Tensegrity Actuated Prototype

For tower movement, there are several drive methods already used in space. The first of the two analyzed is a very interesting structural principle that has wide use in space is Tensegrity[2].[10][11]

Using a series of ropes, the displacement of cylinder No.2 relative to cylinder No.1 can be used to move all the others. The shift of the cylinder applies a force to the two ropes that connect it to cylinders No.1 and No.3, and this causes cylinder No.3 movement relative to cylinder No.2, equal to the shift as that of cylinder No.2 relative to cylinder No.1. The effect propagates for each cylinder causing a relative displacement of each relative to the previous one. A schematic of the operation is shown in Figure 3.2.

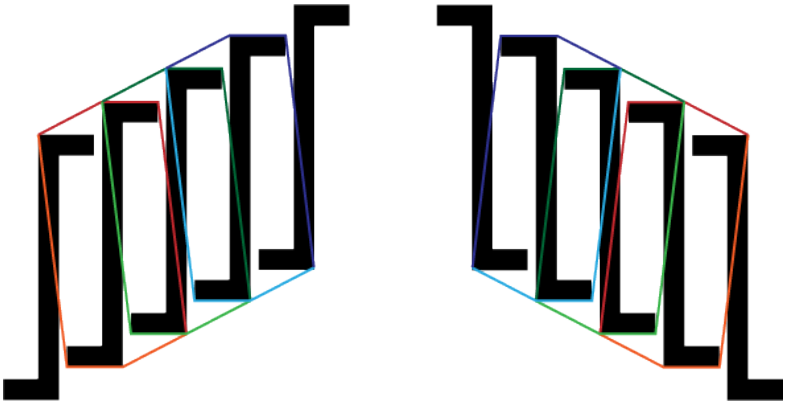


Figure 3.2: Schematic diagram of the model used to build the prototype involving the movement of the telescopic tower using tensegrity

To raise and lower the tower, it is therefore sufficient to make cylinder No.2 lift. For this movement, the idea is to use a leadscrew system.

A screw driven by a servo motor (fixed relative to the device) slides inside a bolt attached to cylinder No.2. Stops prevent the cylinder from rotating, thus allowing it to rise by the rotation of the screw. As anticipated, the movement of the cylinder triggers a chain reaction that causes the entire tower to extend and retract.

[2]Tensegrity is a structural theory that relies on the usage of isolated parts, such as rigid bars or struts (in compression), connected by a continuous web of cables or tendons (in tension) so that the rigid bars or struts do not touch each other.

This method would require minimal force for implementation, as it is sufficient to overcome the tower's weight force and the friction forces of the ropes and screw.

It is decided to use nylonline with a thickness of $0.4mm$, with a tensile strength of about $100N$, to make the prototype. Instead, the servo motor used is model "MG996R"[12]. A modification to the mechanical block is made to the servo that limits its movement to 180° so that it can be continuously rotated clockwise and counterclockwise. The choice to use this model is due to the quality of its metal gears, which give it excellent stability and the ability to develop greater force than a product with plastic gears.

However, a problem arises at the time of implementation. Devices that take advantage of tensegrity mainly use pulleys to run the ropes and locking systems to make the bindings. Finding a product that matched the dimensions of the telescopic tower is impossible, and attempts to tie the nylon ropes are equally complex without a locking mechanism. The knots tend to loosen when the ropes are subjected to pressure from cylinder movements.

The slippage of the nodes makes it impossible to complete the extension and retraction of the tower, and consequently, it is impossible to use this method in its current state.

After the prototype is built and some attempts to solve the problem, it is not considered productive to continue using tensegrity, although it remains an excellent alternative if a method to keep the ropes in tension can be developed.

Figure 3.3: Prototype of the model with tensegrity. Loosening of the nodes causes the inability to fully close the tower

3.1.2 Spring Actuated Prototype

As an alternative to the tensegrity principle, a second method is analyzed to be able to implement the movement of the telescopic tower. This requires the use of a servo motor, an axial compression spring[3] and a spool.

The design of the model can be seen in Figure 3.4.

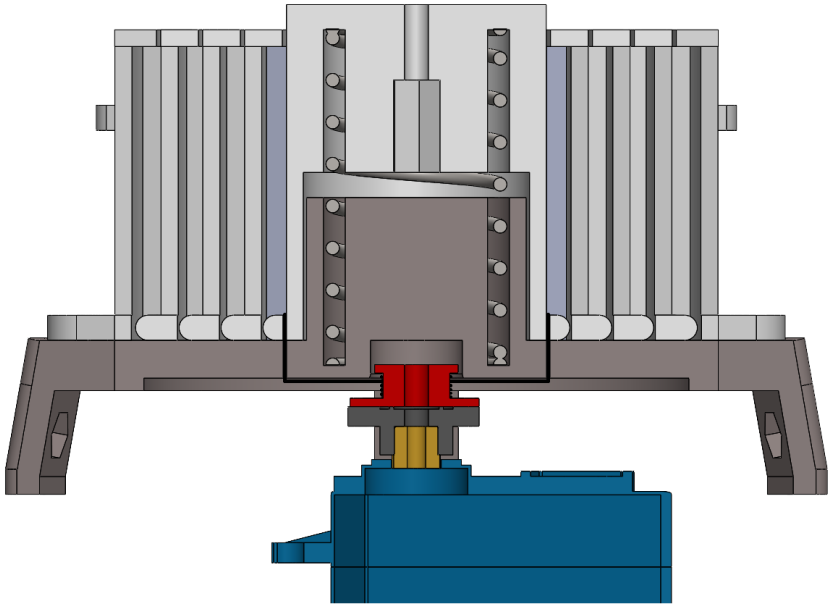


Figure 3.4: Servo-spool-spring model section for extending and retracting the telescopic tower in retracted mode. Blue - Servo motor, Red - Spool, Black - Gathering strings, Light Gray - Spring, White - Telescopic tower cylinders, Dark Gray - Fixed base to supporting structure

The idea is to use the spring inserted between the base of the tower and cylinder No.5, to take advantage of the elastic force greater than the weight force of the tower to carry out its full extension.

The servo motor is used as a motor that regulates the rotation of the spool. This is used to pick up tension rods, which are connected to the top of the telescopic tower. Thus they serve two purposes. The first is to enable the device to close by going to generate a moment around the coil greater than the tension applied to

[3]An axial compression spring develops a greater force as it is compressed, the load applied to the spring causes it to shorten. Applying a tensile force to this type of spring will deform it.

the tie rods by the spring. The second, on the other hand, is to accompany the upward movement of the tower, to prevent the device from being jolted into mid-air. HasuNoHana has a weight of about $0.5kg$ and a sudden release of the spring inside it could cause it to topple over in a low-gravity environment such as the lunar one. The realization of the model is necessary to define some characteristics of its parts.

The servo motor in question is the same as in the previous method, namely the "MG996R" model. It can develop a stall torque of about $100N \cdot cm$ when powered at $5V$, the servo power supply is described in Chapter refchap: Motherboard. This data is the basis from which one must start when designing the motor-spool-spring system.

As for the spool around which the wire is to be wound, it is decided to design one with a diameter of $8mm$, to develop a maximum tension of about $125N$ on the strings. The use of two strings makes it possible to have the resultant of the forces of the two tensions at the center of cylinder No.5 so that the displacement will be linear. Also by distributing the force over two ropes it distributes the tension equally between the two and this ensures that the individual tensions do not exceed the maximum value supportable by the nylon thread, which as mentioned is $100N$.

About the spring, it is necessary to define the characteristics since it is decided to design and produce an ad hoc part instead of adapting an existing product. Based on what has already been said, some constraints must be taken into account.

The first concerns the rest length, which must be equal to the distance between the base to which cylinder No.1 is attached and cylinder No.5 at its maximum extension. This measure is $163mm$. However, the rest length (L_0) is set at $180mm$ to balance the weight force of the tower and still be able to reach the maximum extension. Instead, $35mm$ are available for the length with maximum load (L_n).

The second constraint is again related to the fact that the spring must be contained within cylinder No.5, it follows that the outer diameter (D_e) of the spring must be at most equal to the inner diameter of the cylinder, which is $22mm$. For the inner diameter (D_i), on the other hand, there are no special restrictions and it is still set at $20mm$.

Another constraint comes instead from the maximum elastic force that can be developed, which cannot exceed the maximum tension applicable to the ropes (T_{max}), which is precisely $125N$. Having defined the spring size data mentioned above, it is then possible to use the travel[4] maximum (s_{max}) to impose a maximum elastic constant (k_e) via the system in Formula 3.1. At the same time, it must be kept in mind that the spring must be able to balance the Tower Weight Force (F_p) with the minimum travel (s_{min}) available to the spring when the tower is fully extended. A reasonable value for the spring constant is between 0.10-0.15N/mm.

$$\begin{cases} k_e < \frac{T_{max}}{s_{max}} \\ k_e > \frac{F_p}{s_{min}} \end{cases} \quad [N/mm] \tag{3.1}$$

Regarding the wire diameter, the number of turns, and the spring material, it is not necessary to impose limitations. These characteristics have been established by the manufacturing company, "Mollificio Pirani Srl" to comply with other constraints.

A graphical representation of the spring can be seen in Figure 3.5, the characteristics of which are specified in Table 3.1.

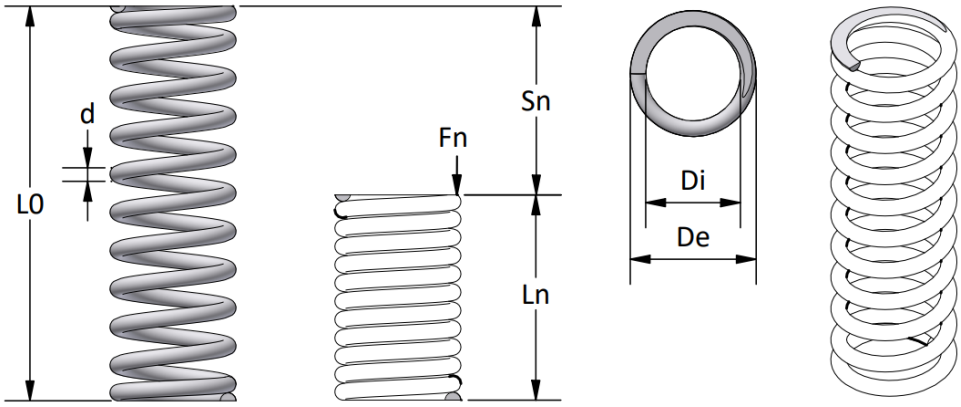


Figure 3.5: Designed spring representation

[4]The term travel means the Δdistance traveled by a spring, which corresponds to a given developed elastic force.

Abbreviation	Attribute	Value	Unit
/	Spring type	Compression	/
/	End	Closed, ground	/
/	Material	EN 10270-1/SM	/
/	Coils number	20	/
d	Wire diameter	1,20	mm
D_e	Outer diameter	22	mm
D_i	Inner diameter	20	mm
L_0	Length at rest	180	mm
L_n	Length with max. load	35	mm
S_n	Maximum travel	145	mm
F_n	Maximum load	18,125	N
k_e	Spring constant	0,125	N/mm

Table 3.1: Summary of values assigned to the designed axial compression spring

As explained in detail in Chapter 4, to avoid wasting energy, the servo motor will not be connected to the power supply all the time, but only when it is necessary to open and close the device. This condition implies that the stall force developed by the servo when not connected to the power supply is equal only to the friction between the gears. Since no data is provided in the datasheet regarding this condition, it is decided to add three hooks to cylinder No.5 to be attached to three other places under the spool of the servo motor.

When the tower reaches the fully retracted configuration the hooks will engage with each other so that the tower will rise if the spring force is greater than the servo stall force when it is not powered.

In designing the hooks, the relationship between the servo motor's rotation and the tower's lowering must be considered, so that the hooks do not slam into each other. The spool makes a complete rotation synchronous to that of the servo motor and for each revolution, the wires travel the equivalent of a circumference, which corresponds to about $25.13mm$.

As mentioned, the hooks that must match are three pairs, of which the three in the cylinder move vertically, while the three connected to the spool rotate about the axis of the servo. It follows that there is $8.37mm$ available to be traveled by the

cylinder hooks, between the passage of one hook and the other of those connected to the spool in the same position.

In addition, a mechanical limit switch[5] is also used as a safety system. to verify that the telescopic tower has been lowered. When the tower completes its lowering, one of the hooks closes the limit switch contact and the signal is sent to the Motherboard, which manages the movement of the servo motor by going to lock it. Thus, the insertion of the limit switch is intended to prevent the servo from going under stress trying to wind up the wires after it has closed while simultaneously preventing the wires from breaking. The system can be seen in Figure 3.6.

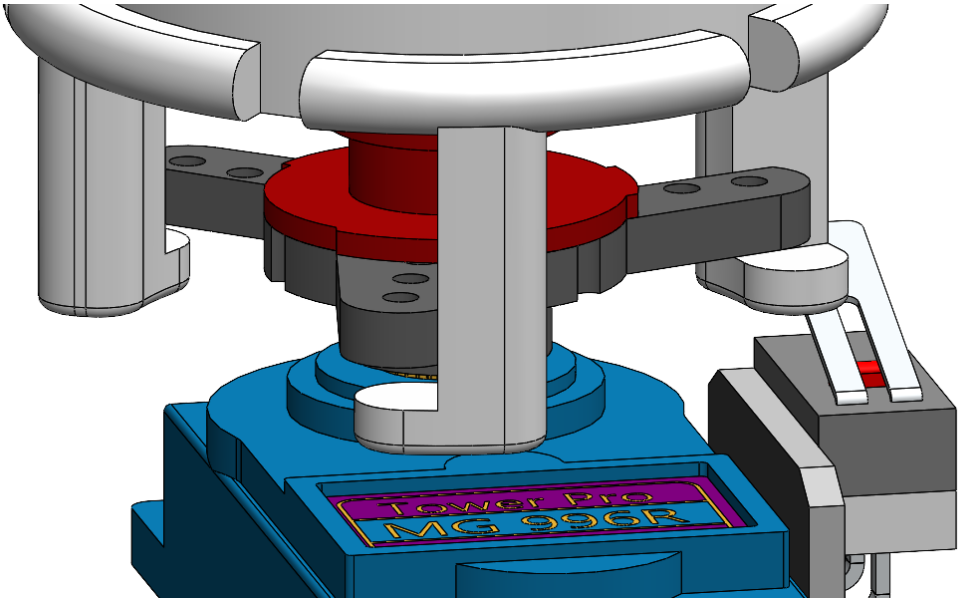


Figure 3.6: Detail of the telescopic tower locking hooks that activate the limit switch to send the signal that the tower has been retracted

When the tower has fully retracted the hooks connected to the servo prevent cylinder No.5 from rising and at the same time one of the cylinder hooks mechanically activates the limit switch.

As the prototype is made, a detail emerges that was not initially taken into

[5]A limit switch is a switch that is activated by the movement of an object or by a machine component.

account during the creation of the model, namely the need to insert a body inside the spring. This condition is necessary because the spring is very long and tends to exit the axis at mid-travel, although the ends are still engaged as shown in Figure 3.4.

The solution to this problem is implemented by simply drilling the two protuberances inserted into the inner diameter of the spring and holding it in place (that of cylinder No.5 and that of the base attached to the structure) and inserting a telescopic metal rod inside the hole. The rod is attached at both ends and its extension is equal to that of the telescopic tower.

The presence of a rigid body inside the spring thus prevents its transverse movement and thus keeps it on the axis so that the device can move properly.

3.2 Doors Development

Regarding the design of the doors, several factors must be considered before 3D modeling can be done.

The need to allocate the PVs in such a way that they are protected during transport, and are then placed in a position to try to optimize energy production by taking advantage of the best angle of incidence of the sun's rays, results first of all in the design of custom-made tilting doors.

The doors are located in the upper half of the device and their shape is designed based on the dimensions of the PVs chosen for the device, which have a size of 45x45x3mm and, as mentioned in Section 2.4.2, are arranged on the sides of a pentagon. With this arrangement, it is verified that the panels reach an angle of inclination at most 45° to the lunar ground.

However, in selenographic latitudes, a greater tilt angle would be more suitable to be able to increase energy production, but it is not possible to increase it more when considering the structure as a whole. This is because it would be necessary to increase the overall size of the device, resulting in a difference in energy production that is not as significant concerning the size.

The part of the doors that results inside the device when closed is designed flat with a 3mm hollow so that PVs can be accommodated inside. While the outer part

follows is convex, so that the set of panels follow the lines of the device skeleton and vice versa, to go to form a sphere $12cm$ in diameter when closed.

Designing the doors so that they can be opened and closed again is also an advantage of dust deposition. A buildup of dust on PV cells causes a reduction in energy production and also affects their temperature. In the absence of an atmosphere, and thus winds, there are no natural phenomena that can contribute to deposition, but the friction of the rover wheels due to displacement after device release, coupled with low gravity can be an actual problem.

However, if the Anchor turns out to be closed after release, it is sufficient for the rover to move away from it before opening to prevent dust deposition from occurring.[7]

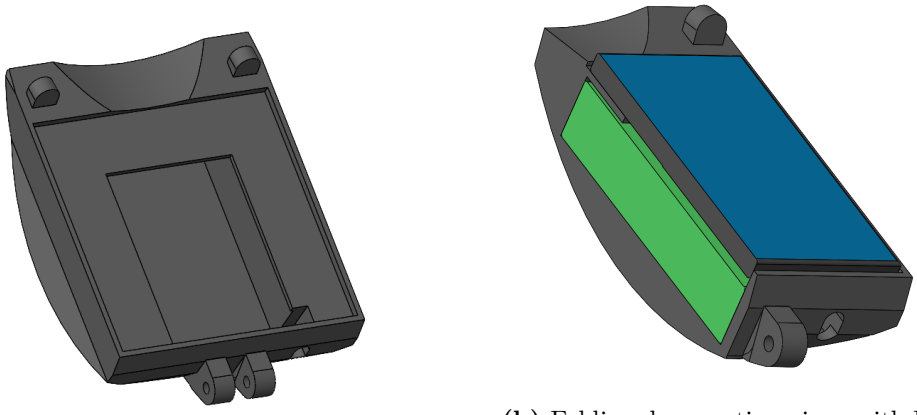
As for the collection of Anchors, provided for any changes in the location of operations, it is sufficient to proceed similarly, that is, by closing the device before the arrival of the rover.

The choice of using as many batteries as PVs as mentioned above, in addition to the energy issues already discussed in Chapter 2, is made in accord with the design of the doors. In fact, in the part below the PVs, with the chosen dimensions, an additional recess can be made that can accommodate one battery per door.

That allows for taking advantage of that space that would otherwise be wasted, which should not be underestimated in terms of sizing the entire device.

Picking up on what is said in Section 2.2 five batteries are sufficient to power the system during dark days, and thinking of using more would not be a great advantage for two main reasons. The first is that it would be necessary to take advantage of additional space inside the device or even go to increase its size; secondly, because the energy production by the PVs would be the same and the storage method as it has been described is considered sufficient.

In Figure 3.7 it can be seen the design chosen for the folding doors, with also the allocation of the battery inside it and the PV in the compartment provided.



(a) Folding door perspective view

(b) Folding door section view, with PV and Battery allocation

Figure 3.7: Doors containing the solar panels and batteries 3D model

On the supporting structure of the device, there are five axial hinges, each arranged in the center of one of the sides of the pentagon, which are to be coupled with those on each of the five doors. Metal rods are then used to hold the two parts of the hinges together and to allow the doors to rotate around each of the axes thus formed. These hinges are used to effectuate the rotation of the doors around the axes so that they can be opened and closed.

Regarding the movement of opening and closing doors, it is necessary to design a method that allows the opening and closing of each panel, taking into consideration the space available. The energy consumption of the opening and closing phases can be considered negligible, as they are sporadic events with a duration of fewer than ten seconds.

Again, parts are 3D printed with Abs filament for the prototype.

3.2.1 Gears Opening Prototype

Since the maximum force that can be developed by the servomotor is far greater than that required to counteract the spring force, it is deemed appropriate to exploit the servo itself to implement the opening and closing of the doors containing the panels and batteries.

The first method brought forward to carry out panel opening involves the use of a mechanism similar to epicyclic gearing. The mechanism is shown in Figure 3.8.

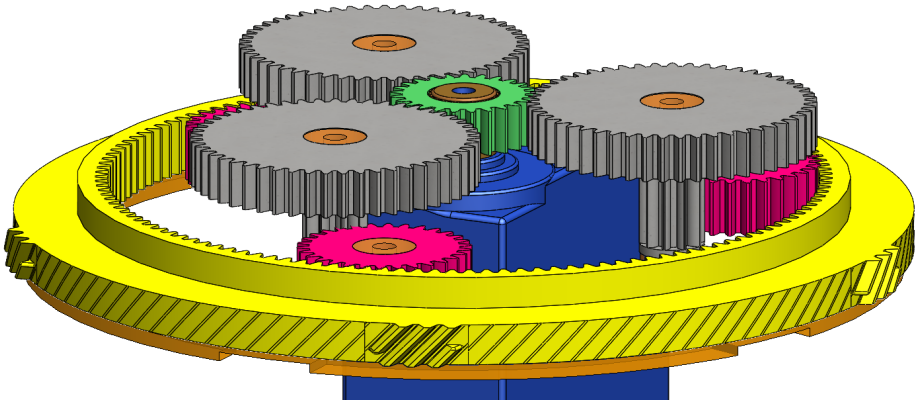


Figure 3.8: Epicyclic gearing-like mechanism actuated by the servo motor and used for door opening. Green - Drive wheel, Gray - First reduction stage, Pink - Second reduction stage, Yellow - Ring wheel for opening the doors, Orange - Bearings

The goal is to drive a ring bevel gear, which will interact with the bevel teeth located on the hinge of the five doors, thus causing them to open. As mentioned earlier in Section 3.1.2, the servo is used to raise and lower the telescopic tower and with each full rotation of the servo, the tower moves $25.13mm$. It turns out that 4.97 revolutions are required to make the full movement of the tower. Since the driving gear wheel is directly connected to the servo motor, it is necessary to carry out the gear train[6] principle to convert and properly drive the ring. The gear train can be calculated by the angular wheels speed (ω), its number of teeth (N), or its radius (r), as in Formula 3.2.

$$T = \frac{\omega_A}{\omega_B} = \frac{r_B}{r_A} = \frac{N_B}{N_A} \tag{3.2}$$

To achieve a synchronous opening of the four gates, it would be necessary to have a gear train between the driving gear and the ring gear between 0.04-0.03, so that by making the full opening of the telescopic tower in about five revolutions of the servo motor, the ring gear would make slightly less than 1/5 of revolution.

Two intermediate reduction stages are used to achieve this gear train. Three

[6]The gear train is a mechanics parameter to characterize how motion transfers from one gear wheel to another. It is denoted by the letter T for practical formulation.

wheels are used for each stage to create symmetry in the system and to avoid ring gear movement due to power transmission. The gears of the first stage are composed of the assembly of two different gears so that one interacts with the driving gear and one with the next reduction stage.

Specifically, the first stage gears are composed of a 51-tooth wheel that interacts with the 25-tooth driving gear, with a gear train of 0.49. The second gear wheel of the first stage composed of 10 teeth interacts with the 30-tooth gear of the second reduction stage, with a gear train of 0.33. Finally, the second reduction stage gear interacts directly with the ring gear, which is composed of 131 teeth, and the gear train characterizing this interaction is 0.23.

As a result, the gear train between the drive and ring gear is 0.037.

The number of gear teeth is defined together with the modulus (m) of the gear to allow the wheels to have an appropriate primitive diameter (d_p), as well as to provide the desired gear train. It follows that the modulus of the gears is $0.6mm$, where the relationship between the three dimensions is expressed in Formula 3.3.

$$m = \frac{d_p}{N} \quad [mm] \tag{3.3}$$

One detail of the ring gear is that the outer bevel teeth, which interact with those of the tipping gates, are not arranged around the entire circumference of the wheel. As can be seen in Figures 3.8 there are five groups of 4 teeth each, each of which interacts with one of the five ports.

With this configuration, it is possible to quickly tilt the doors about halfway up the telescopic tower. To open the doors simultaneously as the tower rises, it would be necessary to have a significantly smaller gearing ratio, a condition that is difficult to achieve because of space issues.

It is still necessary to specify that the gears used as reducers, rotate around a fixed axis. Infect they are held in place by screws connected directly to the supporting structure, but to allow smooth rotation they are inserted at the points of contact with the fixed parts of the bearings.

Similarly, under the ring gear is a bearing that also allows it to be held in place. In addition, even above the gear rig, five bearings locked by the base that holds the telescopic tower in place perform the same task.

The prototype model just described is made by printing all the necessary components using a 3D printer with an Abs filament.

As for the gears with straight teeth, the accuracy of the 3D printer used is sufficient, but the same cannot be said for the bevel teeth. Having an accuracy of $0.4mm$, when printing bevel teeth with a modulus of $0.6mm$ an excessive surface roughness due to the thickness of the individual layers. This drawback prevents the gears from matching as they should and the mechanism jams.

The solution would be to replace the method of producing the gears, going for metal production with much greater manufacturing precision. Therefore, it is decided to look for an alternative solution. It is nevertheless considered interesting to keep in mind the possibility of developing this type of solution at a later date when a suitable method for producing the described gears can be identified.

3.2.2 Winch Opening Prototype

The second solution analyzed to open and close doors containing PVs and batteries uses a principle similar to that used in the Middle Ages to raise and lower drawbridges. In addition, a torsion spring is required[7] whose spindle corresponds to the bar used to hold the two hinges flaps of the five doors in place.

The idea is to fruit the servo motor and a spool similar to what is described in Section 3.1.2, with the difference that the spring is torsion instead of compression. Two strings are attached to the ports' upper ends and balance the spring forces imparted by the springs. The elastic force then is used to open the doors and the mechanism that acts as a winch picks up the ropes for closing.

[7]A torsion spring stores rotational energy. A torsion spring is typically mounted on a spindle. The spindle is sized so that there is sufficient space between it and the inner side of the spring body.

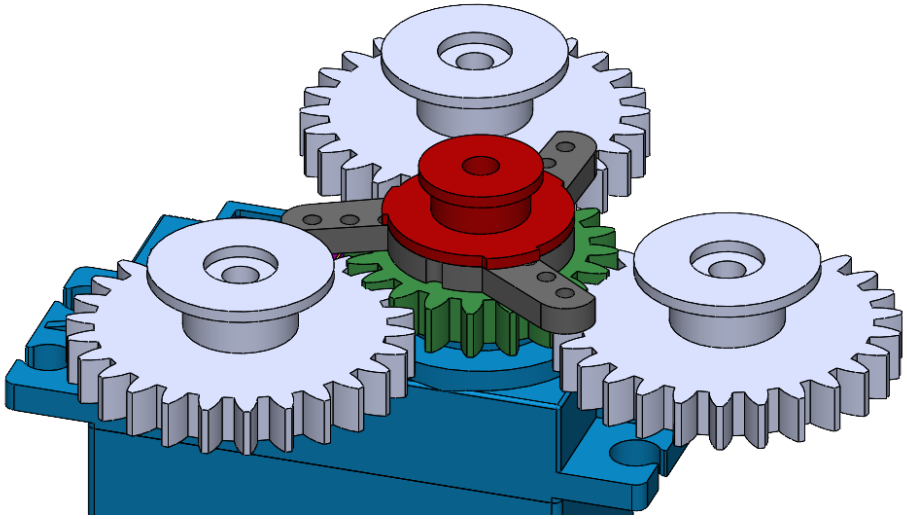


Figure 3.9: System for opening doors containing solar panels by using winches and torsion springs. Green - Driven gear, Gray - Driven gear with spool for wire collection

In the system, it is considered sufficient to use three winches to meet the amount of space available. This limitation is due to the presence of the locking hooks of the system used to move the telescopic tower. A winch consists of a gear connected to the one mounted on the servo motor and a spool that follows the rotation of the gear as shown in Figure 3.9.

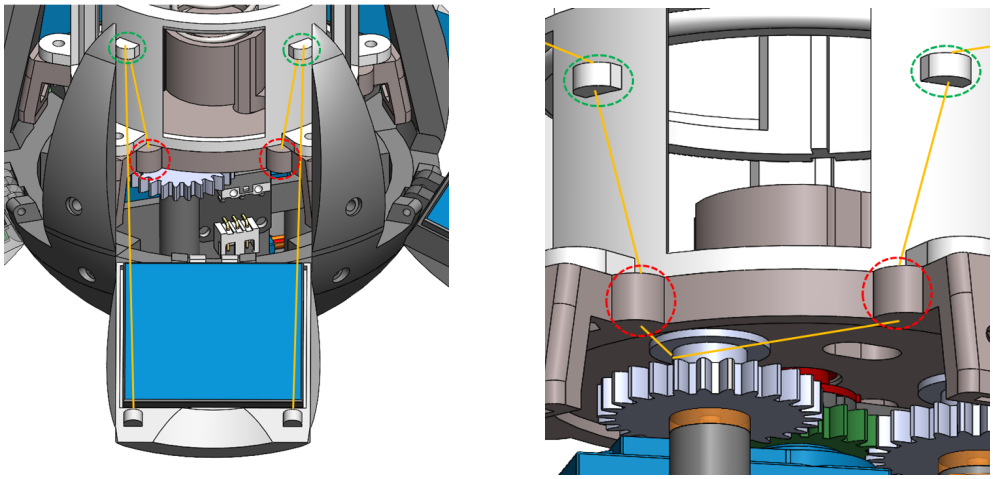
In this case, the opening of the doors occurs simultaneously with the raising of the tower, so the relationship between the travel of the tower and the amount of rope required to open/close the doors must be kept in mind. The rope requirement to make a complete movement of the doors is $106mm$.

The same process described in Section 3.2.1 is used to design the gears and spools that make up the winches. As mentioned, the drive wheel is fixed to the servo and thus makes 4.97 revolutions to effect the raising of the telescopic tower. Using a driving gear with 16 teeth and those of the winches with 26 teeth yields a gear train of 0.73 with the Formula 3.2; this implies that the winches' gears make a total of 3.63 revolutions, which must be sufficient to rewind the $106mm$ of rope required to effect the complete movement of the doors.

Hence, it is sufficient to adjust the diameter of the Spools so that the equivalent of 3.63 circumferences is $106mm$. The result is that the diameter must be $9.3mm$.

The ropes used in this system are the same ones used already described in Section 3.1.2.

Guides connected to cylinder No.1 of the telescopic tower and the base connecting it to the supporting structure are designed to hold the ropes in place, as shown in Figure 3.10.



(a) View from above **(b)** View from below

Figure 3.10: Guides that hold the ropes in place. Yellow - Ropes, Green - Guides placed on cylinder No.5, Red - Guides placed on the bottom of the telescopic tower

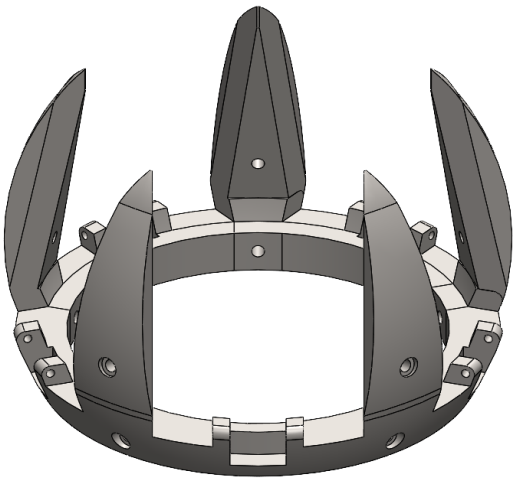
The springs chosen to make the opening are torsion springs with a torsion angle of 180°. This is because, at the moment when the center of gravity is pushed beyond the axis of rotation, the weight force of the doors acts as a counterweight and ensures their opening.

In addition, it is not deemed necessary to carry out the design of a specific spring as in the case of the telescopic tower, since it is possible to use a type available on the market whose datasheet cannot be retrieved to report exact data. During the realization of the prototype, it is possible to see that the spring develops sufficient spring force to perform the opening and, at the same time, is appropriate to be balanced by the servo motor force developed.

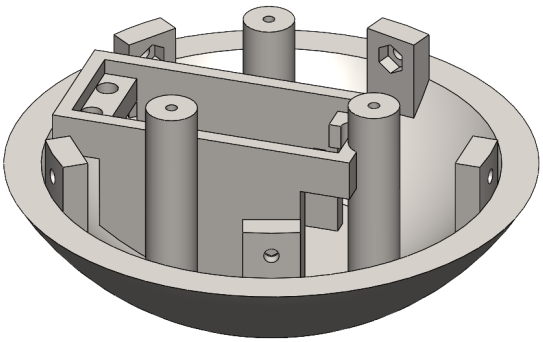
A prototype is made of the model described that is considered to be working, so this method of opening is considered appropriate for the device.

3.3 Shell Structure

The supporting structure of the device is designed by the components described in this chapter to give the $12cm$ diameter spherical shape. It consists of two parts so that all components of the device can be properly assembled.



(a) Upper part of the structure



(b) Lower part of the structure

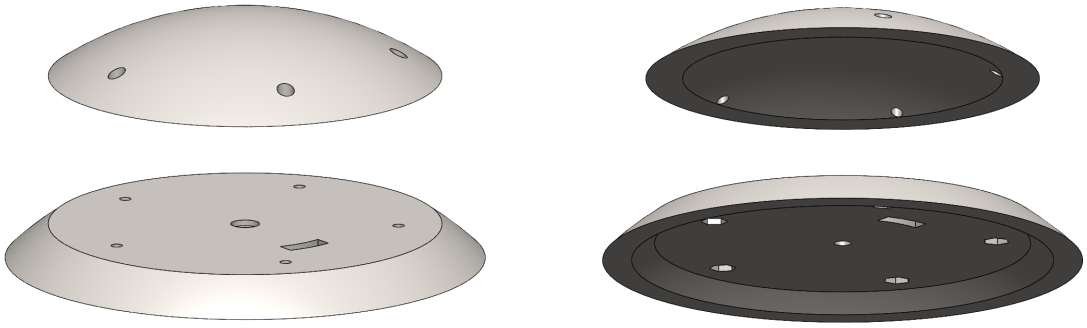
Figure 3.11: HasuNoHana shell 3D structure

The upper part shown in Figure 3.11a goes to fill the gaps between the doors and is also used to hold the base of the telescopic tower in place. In addition, this is where the hinge halves used to move the folding doors are located.

In Figure 3.11b it is possible to see the lower part of the structure, which is used to hold the servo motor in place and mounts three columns on which the winches used to open and close the folding doors are attached.

As for the UWB module, a special container is created to be connected to cylinder No. 5 of the telescopic tower, so that it can be brought to a raised position after the opening.

The container completes the spherical shape of the device and is composed of two parts as shown in Figure 3.12.



(a) View from above **(b)** View from below

Figure 3.12: UWB module container 3D model

The central hole is used to connect and attach the container to the telescopic tower, while the rectangular hole is used to run the electrical cables connecting the UWB module to the motherboard and power supply.

3.4 Full Prototype

Upon completion of all the design considerations explained in the chapter, and after testing the operation of the various opening and closing mechanisms for the doors and panels, it is possible to proceed with the composition of the prototype complete with all its parts to test the simultaneous operation of all components. Figure

A series of photos of the assembly of all parts can be seen in the appendix D. The prototype turns out to be mechanically functional in all respects.

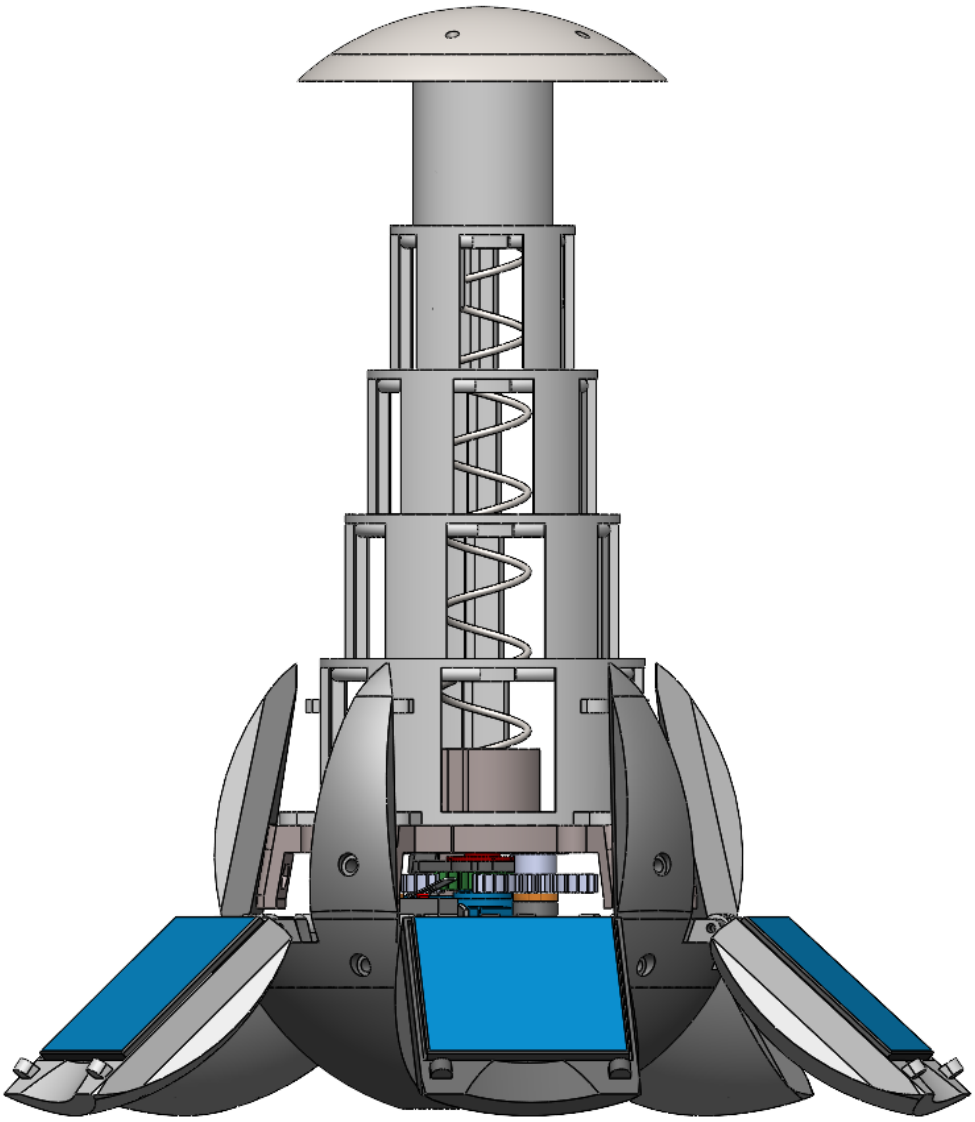


Figure 3.13: Full prototype 3D model

3.5 Final Considerations

First of all, as mentioned within the chapter the solutions for raising and lowering the tower, described in Section 3.1.1, and that for opening and closing the doors, described in Section 3.2.1, are considered valid of note, but not feasible with the

means available in the laboratory. However, if an external method of producing the two mechanisms could be identified, they appear to be very interesting from a design point of view.

One thing that is not considered necessary for the realization of this prototype is to evaluate an individual opening method for each door containing the PVs. For example, the use of pistons, combined with the irradiance studies in Chapter 2, could lead to the realization of a much more advanced prototype capable of adjusting the tilt angle of the PVs during the Synodic Month. This would lead to maximizing the energy production by the charging system.

For the realization of a prototype that is resistant to lunar conditions, one must first make a study of the types of materials to be used, as is explained for the electrical components of the boards in Section 4.8. For the prototype made, these considerations are not done; in fact, the chapter is focused on the idea of making an initial prototype that could work first of all on Earth.

Moreover, downstream of the realization of the prototype and based on subsequent evaluations made at the end of the design of the power supply and charging system described in Chapter 2 and the design of the Motherboard in Chapter 4, it is possible to make some considerations that concern the mechanical part of the design.

It is verified that by enlarging the size of the device by 20% (obtaining a sphere of $15cm$) it is possible to insert two batteries for each tilting door, which goes to double the amount of energy that can be stored. At the same time, it would be possible to go to use larger PVs and it would then be possible to go to produce a greater amount of energy than in the current design.

Under these conditions, the inclusion of a heating system as described in Section 4.8 could be considered.

Chapter 4

Motherboard

In Chapter 2 is established the device's power source, which is a system consisting of five Li-Po batteries recharged by five PV modules. To use and manage this charging system, however, it is necessary to design a Motherboard.

The Motherboard is managed by a Microcontroller that is responsible for controlling the various Input and Output signals (I/O) to manage the transition of each battery between the three states of charge (standby, and operating) during the operating time of the device; but also the I/O that will be used to handle the signals used by all the other devices connected as UWB module, and the components that will be responsible for opening and closing the device, namely: the servo motor, the limit switch.

Everything explained in this chapter has been designed from the ground up to meet HasuNoHana's predetermined task needs, so since they are all novel products they have also been tested to verify their proper operation. The chapter will analyze in detail all the design choices made with their justifications.

Furthermore, Section 4.8 contains considerations of the choices made and possible implementations that can be deployed in the future to have a more advanced version of the device.

4.1 Power Supply

The board needs a power supply system, which has been identified by the set of five batteries in parallel working alternately one by one. What needs to be taken into consideration, however, is that the batteries have a nominal voltage of $3.7V$ and that both the components on the board and the connected devices, i.e., the servo, limit switch, and UWB module require different voltage supplies.

In particular, the UWB module needs a power supply between $2.7V$ and $3.6V$ while the other components are chosen so that they can all operate with $5V$.

It is then necessary to convert the voltage of the power source (batteries) so that the various components can be supplied with the right voltage. Given that batteries operate with Direct Current[1] (DC), it is needful to use devices to convert the voltage from the batteries to those used by the devices.

Another factor to take into consideration is that the current required by the devices is also DC.

One device that can perform this type of conversion is the DC-DC Converter. It serves precisely to convert the DC input voltage to a different DC output value, keeping the power ratio between output and input as close as possible to 1.

The converter can keep the output stable, within a range due to construction factors, even if the input varies, and this aspect is very important because it is true that batteries have a nominal voltage of $3.7V$, but the voltage discharge curve is not constant at every level of charge. The output varies from $4.2V$ to $3V$.

Another issue to take into consideration is that for the voltage required to power the UWB module, the output voltage, set at $3V$, is lower than the nominal battery voltage, on the other hand, that to power the other devices is greater than $5V$.

For this reason, it is necessary to have two different supplies and therefore two different DC-DC Converters, specifically, the one with V_o equal to $3V$ will be of the Buck[2] type, while the one with V_o of $5V$ will be of the Boost[3] type.

[1]The electric current flows in a constant direction, the abbreviations AC and DC are often used to mean simply alternating and direct[13]

[2]Also called step-down is a converter, which steps down the voltage from supply to load.

[3]Also called step-up is a converter, which steps up the voltage from input to output.

The ADP2108AUJZ-3.0-R7 model by "Analog Devices®" is identified as a $3V$ DC-DC Buck, which complies with: the input voltage range; its maximum load current $600mA$, which is amply higher than the approximately $220mA$ required by the UWB module when the working voltage corresponds to $3V$; an efficiency around of 95%[14].

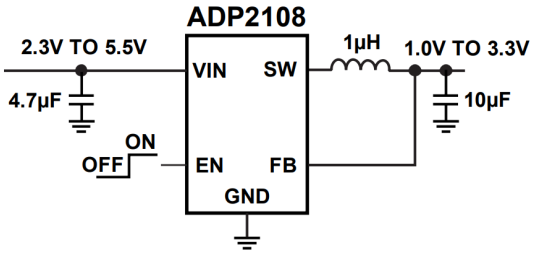


Figure 4.1: Typical application circuit for ADP2108 DC-DC Buck Converter[14]

As $5V$DC-DC Converter Boost is chosen, instead, the RP402N501F-TR-FE by "Nisshinbo Holdings®". This component is chosen because of its technical characteristics: the input voltage range starts from $0.7V$ up to a maximum of $4.8VV$; it has a maximum output current of $800mA$, which is almost double that required by the servo motor (the higher consumption component); and it has an efficiency of about 90%, for currents with intensities on the order of few mA and $400mA$, respectively, the current required by the entire board to be supplied and at the time the servo motor is driven[15].

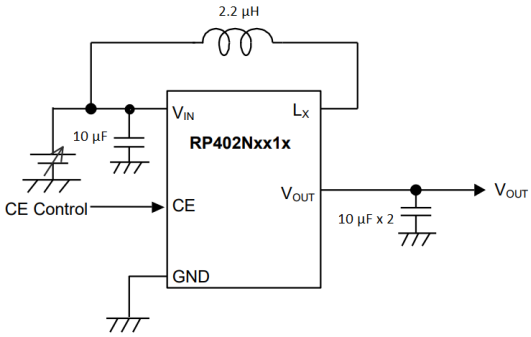


Figure 4.2: Typical application circuit for RP402N501F-TR-FE DC-DC Boost Converter[15]

4.2 Batteries SoC Analysis

To make the transition from one battery to another, one must first gather information about each battery's State Of Charge (SoC). Considering that the device will be a prototype, it is sufficient to set up two battery charge levels, corresponding to full charge and low charge.

Once this information is available, the Microcontroller will take care of what to do, depending on the status of each battery. The possible scenarios will be analyzed in detail in Section 4.5.

Information about SoCs can be collected in several online ways [16] (offline methods will not be mentioned, as it is necessary to have real-time measurements). Among all of these, there are:

- Chemical method
- Voltage method
- Current integration
- Kalman filtering
- Pressure method

However, it must be considered that accessing the liquid state of the battery, for the chemical method, and its internal pressure is difficult for the products chosen for this prototype and Li-Po batteries more generally; and that the Kalman filtering method, to be processed, requires specific components, that includes both current and voltage analysis.

As for current and voltage analysis methods, on the other hand, it must be considered that they are certainly less precise than the Kalman method, but certainly simpler to implement.

In addition, as said before for the design project choices, it is enough to establish the two thresholds indicating the state of full charge and low charge, without knowing the exact trend of the battery SoCs. The voltage analysis is the best solution to implement in this case.

4.2.1 Voltage Method Implementation

The purpose is to implement a circuit that can send a digital signal when the battery voltage exceeds two values. These two thresholds, henceforth, are referred to as High Threshold Voltage (V_{TH}) and Low Threshold Voltage (V_{TL}).

Starting from the model of a Li-Po battery with a Nominal Voltage of $3.7V$, it is possible to extrapolate V_{TH} and V_{TL} from the Voltage-SoC graph (as shown in Figures 4.3).

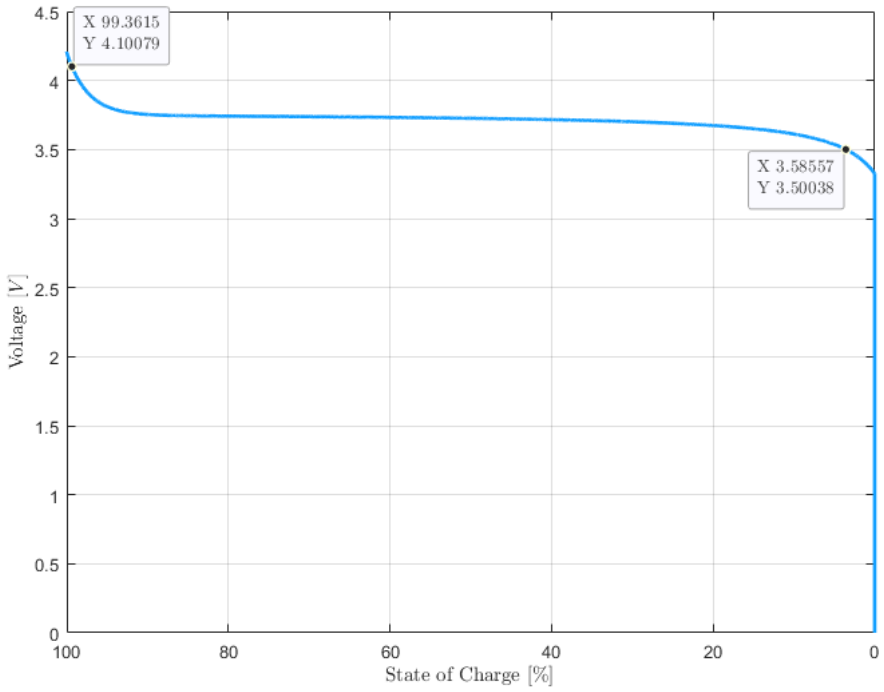


Figure 4.3: Voltage-SoC discharge graph for typical Li-Po batteries with nominal voltage 3.7V

For the chosen batteries, the Cut-off Voltage[4] corresponds to $3.2V$. It is necessary to choose V_{TL} so that it is greater than this voltage; the reference increasing value chosen is greater than 15% so that $V_{TL} = 3.5V$. This voltage threshold value corresponds to approximately 3.5% of the battery charge (as shown in Figure 4.3).

[4]The voltage below which the battery starts to wear out beyond repair.

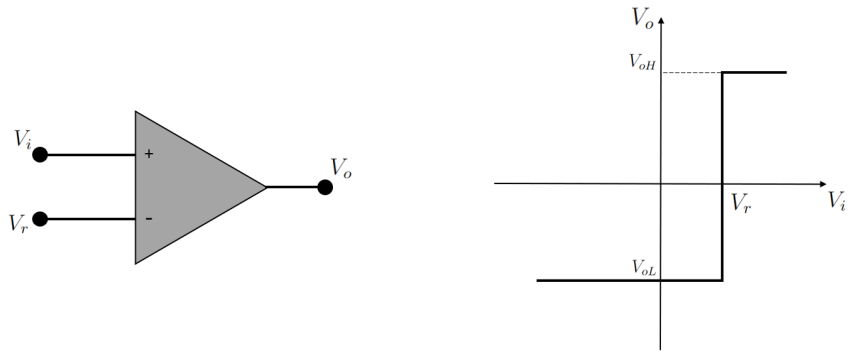
It is not convenient to discharge the battery completely because it wears out much faster, risking the loss of efficiencies with each discharge cycle and making it more unpredictable.

Similar reasoning can be performed to establish V_{TH}. The maximum voltage value that the battery reaches when it is at 100% charge is $4.2V$. Establishing a slightly lower V_{TH} ensures that the threshold will be reached and avoids wasting energy by continuing to keep an already charged battery connected to the PVs. Again, providing more energy, compared to the energy borne, could affect the battery's long-term state of wear and tear.

A reasonable value for the V_{TH} is $4.1V$ so that there is a margin of around 2.5% from the maximum $4.2V$ (as shown in Figure 4.3). This V_{TH} value corresponds to about 99% of the total charge.

The two thresholds have been identified, and a viable solution to collect the relative information of individual batteries is the Threshold Comparator, which can be implemented with an Operational Amplifier (op-amp).

Threshold comparators (shown in Figure 4.4a) are circuits with two possible digital output values, usually High (V_{oH}) or Low (V_{oL}) that compare the analog input value (V_i) with a reference one (V_r). As shown in Figure 4.4b, the output characteristic follows what is specified in Formula (4.1).



(a) Circuit representation **(b)** Output characteristic

Figure 4.4: Non-inverting Voltage Comparator

$$V_o = A_o \cdot (V_i - V_r) \rightarrow \begin{cases} V_o = V_{oL} & if\ V_i < V_r \\ V_o = V_{oH} & if\ V_i > V_r \end{cases} \quad Assuming\ A_o = 1. \tag{4.1}$$

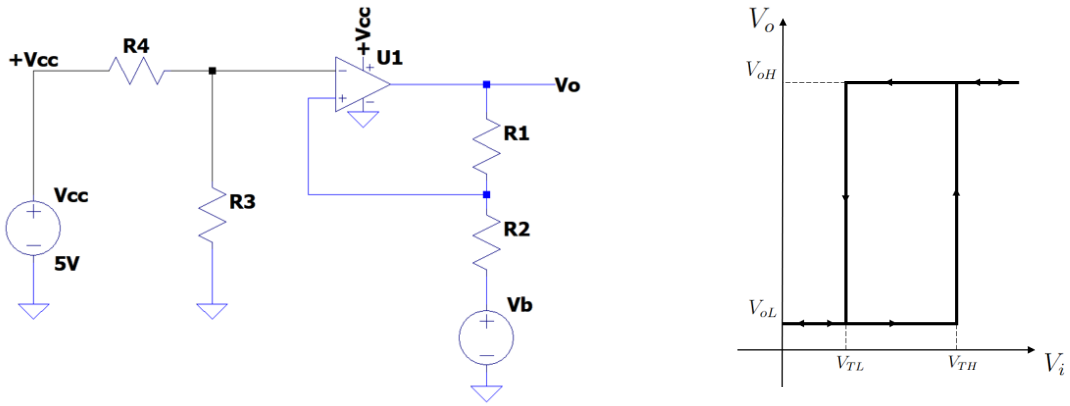
With the constraint that:

$$0 \leq V_i\ \&\ V_r \leq V_{cc} \tag{4.2}$$

Considering that there are two thresholds, and information must be collected for both V_{TH} and V_{TL}, miniaturizing the board is a key issue, and that there are five batteries to check, it is not possible to use two different voltage comparators for each battery.

Another factor to take into account is that using two voltage comparators means that there is no information when neither threshold has not been exceeded. It is needed to keep track of the fact that if the battery has reached V_{TH} and it is connected to the load, the battery must be used until the threshold V_{TL} is reached and vice versa when it is charging.

Thus V_o must remain high from when V_{TH} is reached until the battery is discharged. Conversely, it must remain low from when it reaches V_{TL} until it is fully charged. This process is called hysteresis and can be implemented with a single op-amp (shown in Figure 4.5).



(a) Circuit representation

(b) Output characteristic

Figure 4.5: Voltage comparator with hysteresis

4.2.2 Voltage Comparator with Hysteresis Dimensioning

Since the circuit to be used has been identified and the thresholds have been set, it is possible to proceed with the dimensioning of the resistors present. Note that there are no resistor values in this section, as they are given in Section 4.2.3 where a simulation of the operation of the developed circuit is shown.

$$\begin{cases} V^+ = \dfrac{V_o \cdot R_2}{R_1 + R_2} + \dfrac{V_b \cdot R_1}{R_1 + R_2} \\ V^+ = V_{R_3} \end{cases} \tag{4.3}$$

Remembering that by definition threshold comparator has only two possible outputs as given in the Formula (4.1) and combining the result of the Formula (4.3) it results in that:

$$V_{TH} = V_{R_3} \cdot \frac{R_1}{R_1 + R_2} - V_{OL} \cdot \frac{R_2}{R_1} \tag{4.4a}$$

$$V_{TL} = V_{R_3} \cdot \frac{R_1}{R_1 + R_2} - V_{OH} \cdot \frac{R_2}{R_1} \tag{4.4b}$$

Finding V_{R_3} in one of the two equations in the Formula (4.4) and substituting it into the other gives the ratio between the resistances R_1 and R_2 as given in the Formula (4.5).

$$\frac{R_2}{R_1} = \frac{V_{TH} - V_{TL}}{V_{OH} - V_{OL}} \tag{4.5}$$

R_1 can be chosen arbitrarily, on the order of hundreds of $K\Omega$, while R_2 is derived from the Formula (4.5) (the values chosen are given at the end of this section).

To set R_3 and R_4, instead, it is possible to start from the voltage divider in Formula (4.6).

$$V_{R_3} = V_{cc} \cdot \frac{R_3}{R_3 + R_4} \tag{4.6}$$

V_{R_3} can be estimated from the Formula (4.4b) and by doing the inverse of Formula (4.6) it is possible to determine the ratio between R_3 and R_4.

$$\frac{R_3}{R_4} = \frac{V_{R_3}}{V_{cc} - V_{R_3}} \tag{4.7}$$

The choice of values for these resistors is not arbitrary, but certain characteristics of the chosen op-amp must be taken into consideration. The first figure is the Input Bias Current[5] (I_{bias}), which for the LPV32x series by "Texas Instruments®" is on the order of tens of nA[17]. Minimizing consumption to avoid wasting power is a priority. So the maximum value to be imposed for I_{R_3} is $100\mu A$.

After these two considerations, a reasonable value for Ir3 is on the order of tens of μA, which is amply sufficient to overcome the influence of I_{bias}.

$$R_3 = \frac{V_{R_3}}{I_{R_3}} \tag{4.8}$$

Now that I_{R_3} has been established, R_3 can be calculated from Ohm's Law (Formula (4.8)), and consequently, from Formula (4.7) the value of R_4 can also be determined.

4.2.3 Voltage Comparator with Hysteresis Simulation

According to the datasheets of the LPV32x series of op-amps[17], considering the established values for the thresholds V_{TH} and V_{TL} and what is stated in the Formula (4.2), the value of V_{cc} chosen is $5V$.

Applying what is said in Section 4.2.2, optimal values of the resistances are calculated and shown in Figure 4.6. To verify the integrity and operation of the designed circuit, a simulation is carried out with the software "Analog Devices Inc. LTspice®".

Through the simulation shown in Figure 4.7, it can be seen that the output behaves as expected. The switching of the output occurs at the voltages indicated by Cursor1 and Cursor2, which correspond to V_{TH} and V_{TL}, respectively.

With these values of the resistances, V_{R_3} is about $3.65V$, and consequently, the current I_{R_3} is around $10\mu A$. Power dissipation is negligible.

[5]The current flowing in reverse at the input of the op-amp.

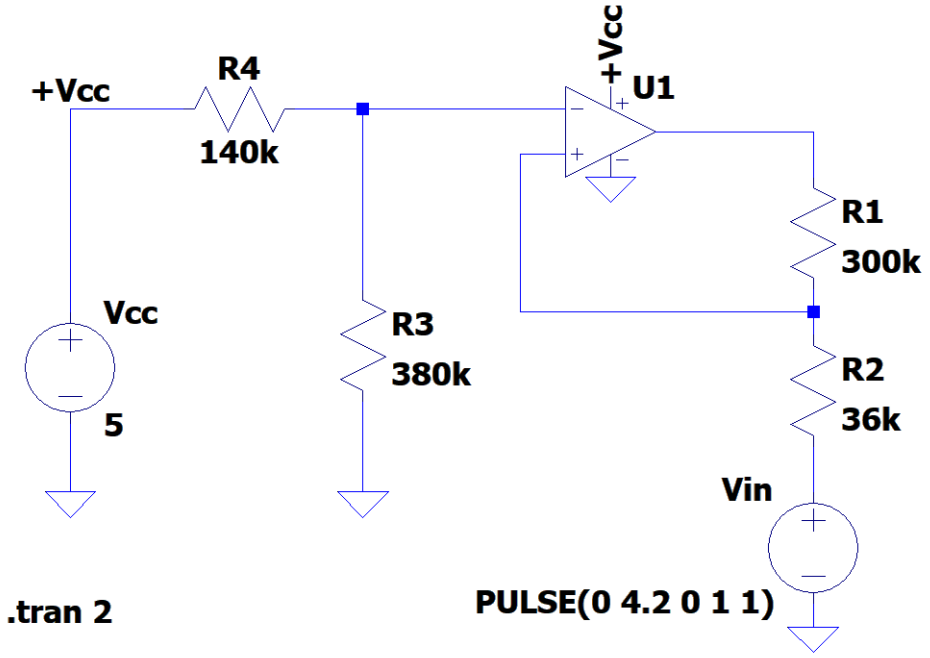


Figure 4.6: LTspice Voltage Comparator circuit simulation

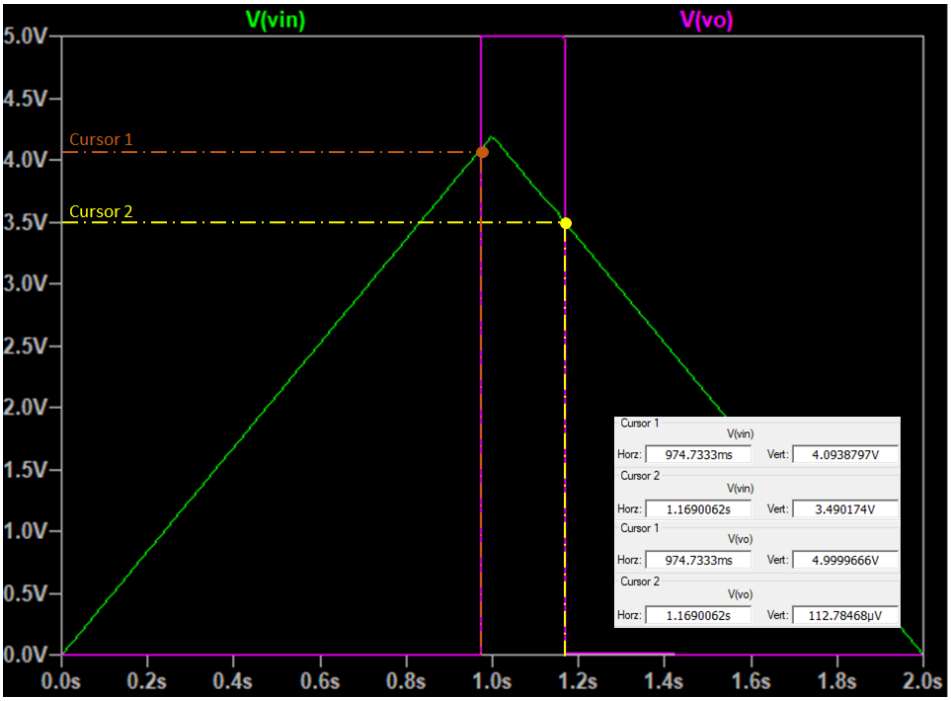


Figure 4.7: Voltage Comparator I/O simulation on LTspice

4.3 Three Way Switching System Development

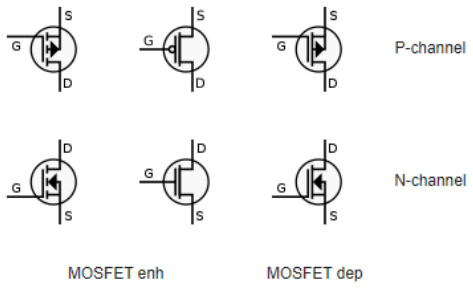
After analyzing the method for identifying the state of the battery, or more precisely, when it is charged and when it needs to be recharged, it is possible to design a system for switching between states such as:

- Charging
- Sandby
- Operating

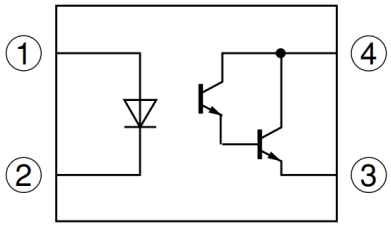
This system can be implemented using two switches for each battery, one that will connect, when closed, the battery to the charging system, and one that will connect it, to the load when closed. The standby state is simply achieved by keeping both switches open.

Both switches of each battery are operated by the microprocessor depending on signals collected by the Voltage Comparators, this topic is analyzed in Section 4.5.

The first choice for designing the circuit is to identify the type of component to be used as a switch. The focus fell on two different types of components, namely, MOSFETs[6] and Optoisolators, respectively in Figures 4.8a and 4.8b.



(a) Schematic representation of different types of MOSFET: G. gate, S. source, D. drain



(b) Schematic representation of an Optoisolator with: 1. Anode, 2. Cathode, 3. Emitter, 4. Collector[18]

Figure 4.8: Schematic representation of MOSFETs and Optoisolator

[6]The metal-oxide-semiconductor field-effect Transistor.

Optoisolators operate through coupling between an LED and a photosensitive element, which enables the transfer of current between the emitter and the collector as the brightness of the LED changes. The advantage of these devices lies in the lack of electrical continuity between the two circuits.

To the detriment of the advantage provided by Optoisolators, however, there is that they are devices that take up more space compared to MOSFETs, precisely because of the presence of the LED; but even worse, Optoisolators require a relatively high current to power the LED.

The types of MOSFETs shown in Figure 4.8a are N-MOS[7] and P-MOS[8], which, in comparison with the Optoisolators: allow more current flow between drain and source than between emitter and collector; have significantly less power dissipation when working with low voltages on the load; have smaller dimensions. In this case, the electrical continuity problem is less important, because in the case under analysis there is little voltage difference between the control and load circuits. It follows that MOSFETs are preferable to Optoisolators in this case.

N-MOS and P-MOS differ concerning the method of conduction between drain and source. The former call up electrons between the two N-interfaces above a P-substrate; on the contrary, P-MOS have an N-substrate and call up between one P-interface and the other holes. This affects the size of the devices because holes move at about one-third the speed of the electrons, and for the same current flow, P-MOSs must be significantly larger.

Another thing to consider when choosing the type of transistor is the location of the switch. As can be seen in Figure 4.9, it is possible to use MOSs in two possible configurations, namely the Low side or High side. N-MOS is most suitable for the Low side, while P-MOS is recommended for the High side.

The first difference between the two configurations lies in the position of the switch relative to the load, which in the case of the Low side is precisely between the load and the ground, while in the case of the High side it is between the supply and the load.

[7]N-type metal-oxide-semiconductor logic (field-effect transistors).

[8]P-type metal-oxide-semiconductor logic (field-effect transistors).

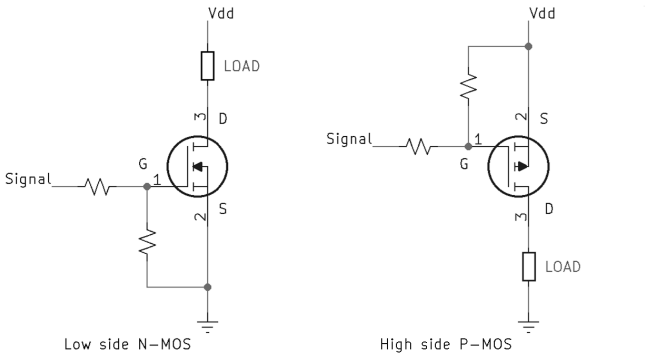


Figure 4.9: Example of High and Low side MOSFET typical circuits

By using two switches in Low side, one upstream and one downstream of the battery, it happens that there are two different groundings. The first one puts the batteries in common with the recharge system and the second one with the load. An event that does not occur with the High side configuration, where each component is connected to a common ground.

Another difference concerns the potential difference between gate and source. The MOS acts as an open switch when the potential difference between the gate and the source $V_{GS(TH)}$ is below the threshold voltage set by the datasheet, instead, it is a closed switch when this threshold is exceeded.

If the $V_{GS(TH)}$ is exceeded with a small difference, however, several problems could arise, including the risk that noise or voltage fluctuations could cause the switch to open and close; the resistance between drain and source R_{DS} increases, and as a result, the component draws more power; the maximum current limit I_D that can flow between drain and source is drastically reduced, and if exceeded there is a risk of burning out the component.

The potential difference in the High side configuration could create the above problem considering that: the gate voltage corresponds to the Microcontroller's supply voltage i.e., $5V$ as said in Section 4.1; the battery voltages are nominally $3.7V$; there are voltage drops due to the diodes used to prevent current from flowing between one battery and the other.

On the other hand, having two separate groundings implies taking more care in handling battery state changes because of the voltage surges they may cause, but this issue will be analyzed in Section 4.6.

4.3.1 N-MOS Dimensioning & Power Dissipation

In the end, it is chosen to use N-MOSs in the Low side position instead of P-MOS as shown in Figure 4.10. This choice accommodates the need to miniaturize and minimize power losses.

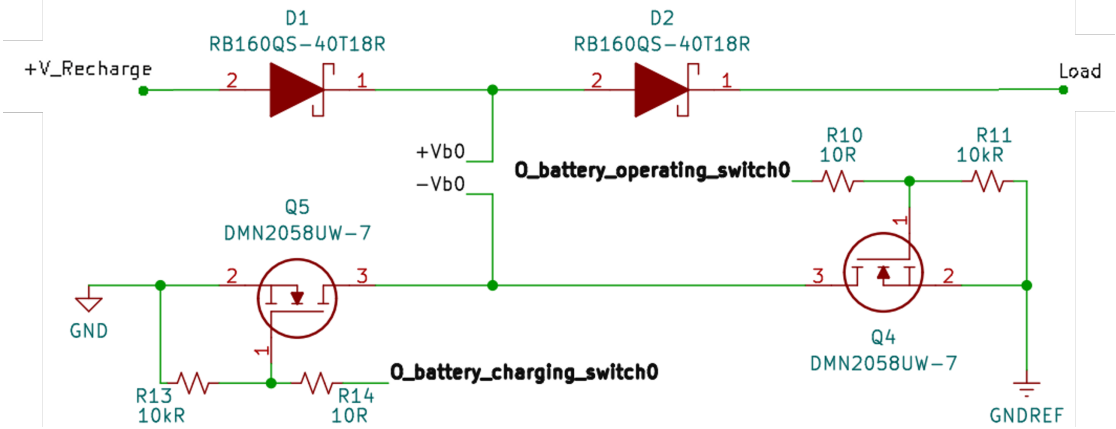


Figure 4.10: Chosen circuit to switch every battery between the three states: Charging, Standby, Operating; in bold are the digital control signals

In particular, the chosen product is the DMN2058UW by "Diodes Incorporated®" which fully reflects all the required features. As reported in the datasheet, the maximum current I_D is $3A$ and is much higher than the one required by the loads, the $V_{GS(TH)}$ at a lower value at $700mV$ at room temperature, the minimum Breakdown Voltage is $20V$, and also has a R_{DS} of the order of tens of $m\Omega$ that ensures low power dissipation.[19]

The power dissipation due to switching is irrelevant, this is because there is a capacitance between gate and source that allows current I_D to flow when fully charged. Considering that the switches change state only twice several hours apart between battery use, the current required by the gate is only needed to charge the capacitor and the charge required by the DMN2058UW is of the order of nC[19].

Instead, resistive power dissipation $P_{resistive}$ can be estimated With Formula 4.9.

$$P_{resistive} = (I_{LOAD}^2 \cdot R_{DS}) \cdot \frac{V_{out}}{V_{in}} \tag{4.9}$$

As a result, $P_{resistive}$ is on the order of a few tens of mW during the use of the UWB module, while it is practically negligible when the module is on standby.

The 10Ω resistor placed at the gate input of Figure 4.10 serves to dampen the time constant τ of capacitor charging. While the resistor between gate and source is the pull-down resistor.

4.4 Switches for Connected Devices

As anticipated, HasuNoHana can be relocated to different spots, and this implies the presence of a servo motor and a limit switch to manage its opening and closing.

As can be seen in Section 4.5, during its operating time, the UWB module is turned off when part of the batteries are discharged to allow for full recharging.

The need to use these devices only during certain periods, shorter or longer, requires the use of switches to allow them to be turned on and off. This is to avoid having wasted power.

In Section 4.3.1, an optimal solution has already been analyzed to implement switches through the use of N-MOS in low-side configuration, and this solution can also be used for this application.

Figure 4.11 shows how DMN2058UW is used following the above to enable the servo motor, limit switch, and UWB module to be turned on/off through an output signal from the Microprocessor.

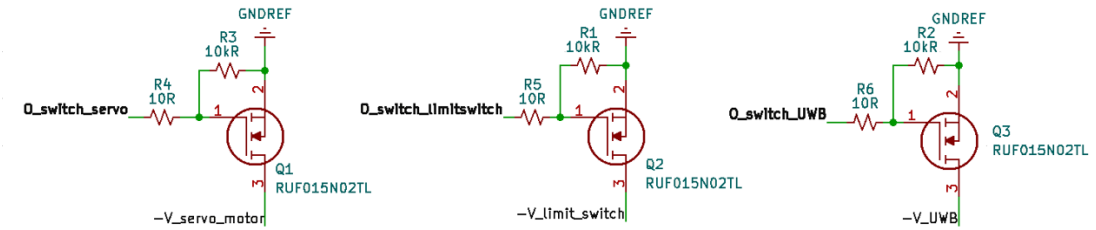


Figure 4.11: Chosen circuit to switch on/off servo motor, limit switch, and UWB module; in bold are the digital control signals

4.5 Microcontroller

The Microcontroller is the core of the Motherboard and is responsible for executing the code capable of performing all the operations necessary for HasuNoHana to operate.

Before the proper choice is made, however, some evaluations must be made. It is necessary to figure out how many pins are needed to control all the Inputs and Outputs (I/O) of the various components; analyze the speed in processing the necessary operations; verify that the required power supply is compatible with those designed in Section 4.1; and so on.

4.5.1 I/O Description

First, it is necessary to declare the I/O required and their type, whether Digital or Analog. In particular, Figure 4.12 lists all the Inputs (4.12a) and Outputs (4.12b) required for the electrical components arranged on the board.

```
pinMode(I_threshold_comparator[0], INPUT);
pinMode(I_threshold_comparator[1], INPUT);
pinMode(I_threshold_comparator[2], INPUT);
pinMode(I_threshold_comparator[3], INPUT);
pinMode(I_threshold_comparator[4], INPUT);
pinMode(I_battery_fault, INPUT);
```

(a) Input from components on the board

```
pinMode(O_battery_charging_switch[0], OUTPUT);
pinMode(O_battery_charging_switch[1], OUTPUT);
pinMode(O_battery_charging_switch[2], OUTPUT);
pinMode(O_battery_charging_switch[3], OUTPUT);
pinMode(O_battery_charging_switch[4], OUTPUT);

pinMode(O_battery_operating_switch[0], OUTPUT);
pinMode(O_battery_operating_switch[1], OUTPUT);
pinMode(O_battery_operating_switch[2], OUTPUT);
pinMode(O_battery_operating_switch[3], OUTPUT);
pinMode(O_battery_operating_switch[4], OUTPUT);

pinMode(O_switch_servo, OUTPUT);
pinMode(O_switch_UWB, OUTPUT);
pinMode(O_switch_limitswitch, OUTPUT);
```

(b) Output for components on the board

Figure 4.12: Components on the board I/O

The number that succeeds the Input names related to the threshold comparators (*I_threshold_comparator*[*i*]) and the Outputs of the switches that manage the connection of the batteries with the charging system (*O_battery_charging_switch*[*i*]) and the loads to operate (*O_battery_operating_switch*[*i*]) indicate the corresponding battery number. The other three outputs, on the other hand, handle the switches that are responsible for turning the servo motor on and off, the limit switch, and the UWB module described in Section 4.4. All these described I/O are Digital.

Figure 4.13, on the other hand, shows the I/O required for devices that are not located on the board.

```
pinMode(I_limit_switch_signal, INPUT);
pinMode(I_all_device_ready, INPUT);
pinMode(I_open_device, INPUT);
pinMode(I_close_device, INPUT);
```

(a) Input from components on the board

```
pinMode(O_device_open, OUTPUT);
servo_motor.attach(O_servo_signal);
```

(b) Output for components on the board

Figure 4.13: Components on the board I/O

The first Input (*I_limit_switch_signal*) gives information about the limit switch's state to tell whether HasuNoHana is open or closed. The other Inputs, on the other hand, are connected with the UWB module and are used to receive commands to operate or not to operate depending on the state of the other devices (*I_all_device_ready*) and the commands to open (*I_open_device*) and close (*I_close_device*) HasuNoHana. These inputs are Digital.

As for the Outputs, the first one communicates to the UWB module the state of the open/close device (*O_device_open*) after performing the open/close operation, this Output is Digital. On the other hand, the Output that handles the movement of the servo motor (*O_servo_signal*) is PWM (Pulse-Width modulation).

The total number of I/O pins required amounts to 24 Digital pins and 1 PWM pin.

4.5.2 Microcontroller Choice

Several products are analyzed including the ATMEGA2560 and ATMEGA2560V by "Microchip Technology / Atmel ®". Other PIC-series Microcontrollers are also evaluated, but discarded because of the need to implement from scratch the functions needed to perform certain operations. Functions present instead in the ATMEGA series software when running the boot loader.

The capability to operate with a supply voltage between $4.5V$ and $5.5V$, the 54 programmable pins (both Digital, Analog, and PWM types), and the switching frequency up to 16 MHz make the ATMEGA2560 an excellent candidate for the required specifications.[20]

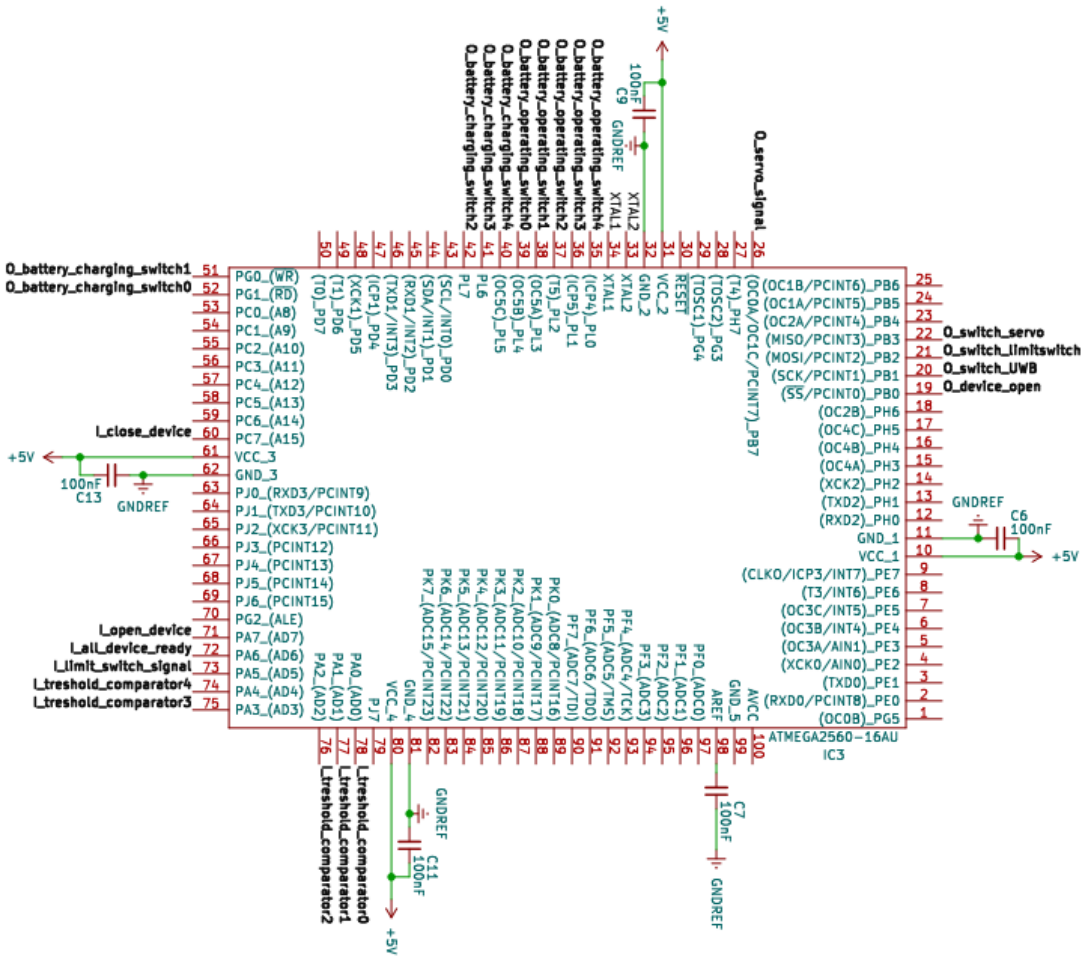


Figure 4.14: ATMEGA2560 pins connections

On the other hand, ATMEGA2560V meets, in the same way, the required needs but works with lower power, the required power supply is $3V$. As a con, however, it has that the maximum frequency is $8MHz$, but given the specifications, this is a negligible problem. However, the ATMEGA2560V at the time of design is unavailable for a long time, but it should be kept in mind for possible future prototypes.

After these evaluations, the model chosen is the ATMEGA2560.

As can be seen in Figures 4.14, the I/Os are connected to their respective pins, and the four power systems are connected taking into account the capabilities required in the product datasheet.

4.5.3 Crystal Oscillator

The Microcontroller needs a crystal oscillator, which is an electrical circuit that uses a piezoelectric crystal to select the right oscillation frequency, the device is used as a clock.[21]

The crystal's frequency is indicated by ATMEGA2560, which corresponds to $16MHz$.

LFXTAL062487Reel by "IQD Frequency Products ®" is chosen from the various compatible crystals, and the circuit implemented to use it as a clock is shown in Figure 4.15 and connects to the Microcontroller via the XTAL1 and XTAL2 inputs shown in Figure 4.14.[22]

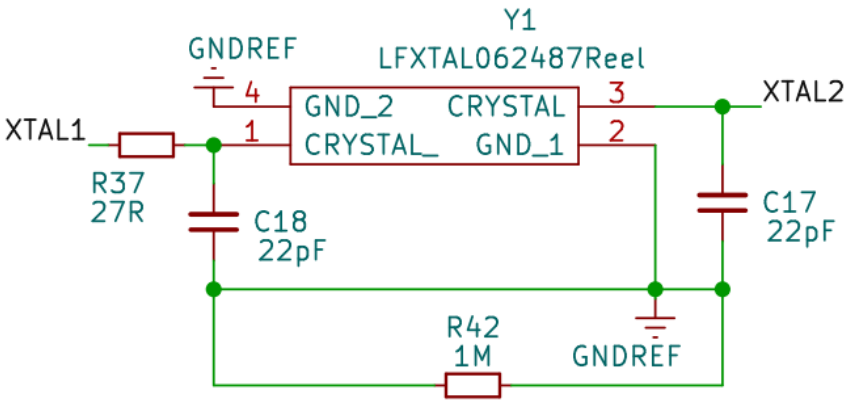


Figure 4.15: Crystal oscillator circuit used as a clock for the Microcontroller

4.6 Code Explanation & Test

The code loaded on the Microcontroller is written in Wiring (a language based on a C/C++ library) and consists of four functions designed *ad hoc* to operate properly. They have been defined separately and are called in the *loop*() sequentially as shown in the Listing 4.1.

```
1  void loop() {
2
3    // Manage batteries
4    BATTERY_MANAGEMENT();
5
6    // Manage the UWB module activation
7    UWB_MANAGEMENT();
8
9    // Open or close the device if needed
10   if(digitalRead(I_open_device) == HIGH){
11      OPEN_DEVICE();
12   }
13   else if(digitalRead(I_close_device) == HIGH){
14      CLOSE_DEVICE();
15   }
16
17   // Repete operation every second
18   delay(1000);
19   first_activation = 0;
20 }
```

Listing 4.1: Loop Funcion

This section discusses in detail the operations of each function called in *loop*(), but a complete view of the code is available in Appendix B. Also included in Appendix B are the two *.h* files in which a timer function and a function to perform summing have been implemented.

4.6.1 BATTERY_MANAGEMENT()

The first thing to handle is to perform battery management, which is done using the $BATTERY\_MANAGEMENT()$ function. The function consists of a loop in which at each iteration the battery switches corresponding to the index [i] of that iteration are managed.

At the beginning of each cycle, the threshold comparator signal with hysteresis related to the current battery is read. As explained in Section 4.2.2, the digital signal sent to the Microcontroller from the threshold comparator corresponding to

that battery is read and saved in a variable so that it can be reused later in the other functions.

The code for the $BATTERY\_MANAGEMENT()$ function can be seen in Listing 4.2.

```
void BATTERY_MANAGEMENT(){

  //battery in charge: battery_switch_state[x] == 0   ;  operating[x]=LOW  && charging[x]=HIGH
  //battery waiting:   battery_switch_state[x] == 1   ;  operating[x]=LOW  && charging[x]=LOW
  //battery operating: battery_switch_state[x] == 2   ;  operating[x]=HIGH && charging[x]=LOW

  for (int i=0; i<5; i++){
  // Current battery threshold acquisition
    threshold_comp_value[i] = digitalRead(I_threshold_comparator[i]);
    if(threshold_comp_value[i] == LOW && battery_switch_state[i] == 2){
      // Switch to another Battery if the operating one has a low charge
      if(i!=4){
        // Battery [i+1] set to operating
        digitalWrite(O_battery_operating_switch[i+1], HIGH);
        digitalWrite(O_battery_charging_switch[i+1], LOW);
        battery_switch_state[i+1] = 2;
        delay(10);
      }
      else {
        // Battery [0] set to operating
        digitalWrite(O_battery_operating_switch[0], HIGH);
        digitalWrite(O_battery_charging_switch[0], LOW);
        battery_switch_state[0] = 2;
        delay(10);
      }
      // Current battery [i] in standby (ready to be charged
      digitalWrite(O_battery_operating_switch[i], LOW);
      digitalWrite(O_battery_charging_switch[i], LOW);
      battery_switch_state[i] = 1;
      delay(10);
    }
    else if (threshold_comp_value[i] == HIGH && battery_switch_state[i] == 0){
      // Current battery [i] in standby ready to be used
      digitalWrite(O_battery_operating_switch[i], LOW);
      digitalWrite(O_battery_charging_switch[i], LOW);
      battery_switch_state[i] = 1;
      delay(10);
    }
    else if(threshold_comp_value[i] == LOW && battery_switch_state[i] == 1 && SUM5(battery_switch_state)>5){
      // Current battery [i] in charge
      digitalWrite(O_battery_operating_switch[i], LOW);
      digitalWrite(O_battery_charging_switch[i], HIGH);
      battery_switch_state[i] = 0;
      delay(10);
    }
  }
}
```

Listing 4.2: Battery management function

In the comments at the beginning of the $BATTERY\_MANAGEMENT()$ function, it can be seen what the three states of battery x are, the index they correspond to, and the corresponding switch combinations assigned to that battery.

The Microcontroller checks what it acquires from the threshold comparator of each battery and executes one of the following commands:

- If a battery $[i]$ is connected to the load and its voltage drops below the minimum threshold, then the next battery $[i+1]$ is put into operating mode and connected to the load. After a delay greater than the time constant that determines the effective closure of the N-MOS, battery $[i]$ is disconnected from the load and put into standby mode. If the battery under consideration is number [4] (the last one), number [0] (the first one) is considered the next.

- If a battery $[i]$ is in charging mode and its voltage rises above the maximum threshold value, then the battery is put into standby mode by disconnecting it from the charging system. Again, the delay is used to verify the actual state change from closed to open and vice versa of the N-MOS.

- If battery $[i]$ is in standby mode, found to be discharged, and no other battery is charging, then it is put into charging mode and connected to the charging system. Checking that no battery is charging is done by the $SUM5$ function which sums the values for the states of the batteries, if the total is greater than 5 then besides the battery in operating mode, the others are all in standby mode.

- If none of the previous events occur then no operation is performed on the current battery

To verify the actual operation of these functions, a test can be carried out using "Arduino Mega" which is equipped with the ATMEGA2560 Microcontroller, corresponding to the one chosen for the Motherboard.

By connecting the I/Os to a digital probe of the oscilloscope, it is possible to see the correct behavior of the Outputs signals related to the switches of all the batteries as the Inputs of the threshold comparators change, as shown in Figure 4.16.

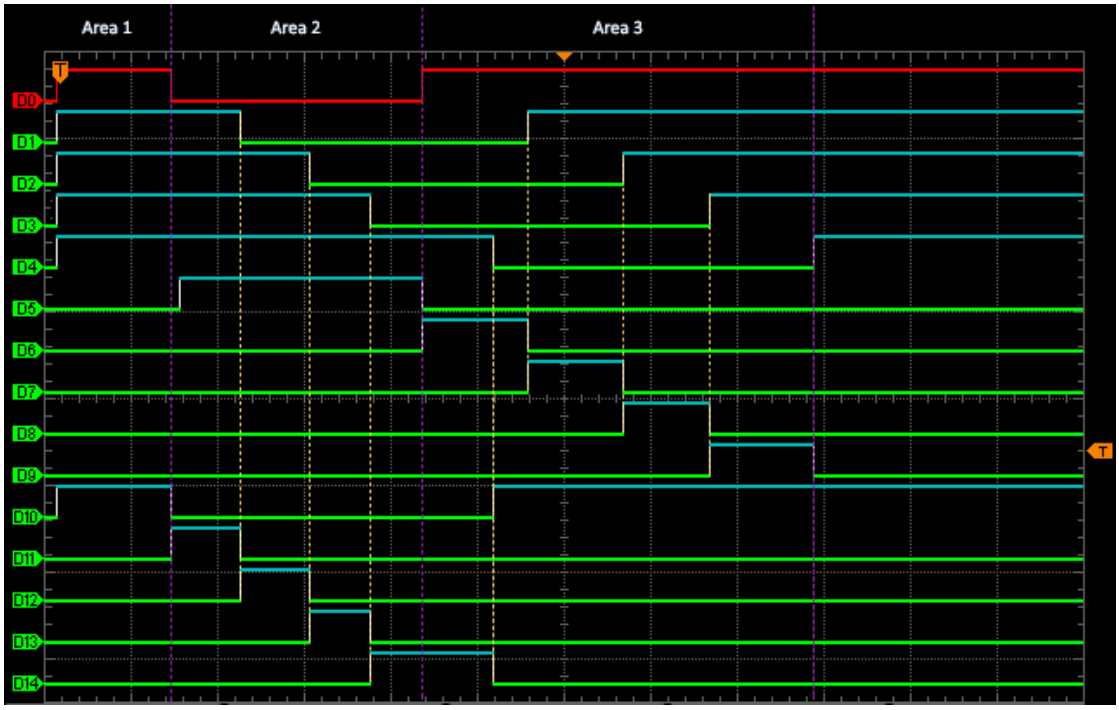


Figure 4.16: Simulation of the battery management function: D0-D4 (in order of battery 0-4) are the Inputs of the threshold comparators; D5-D9 (in order of battery 0-4) are the Outputs of the switches connecting the batteries to the charging system; D10-D14 are the Outputs (in order of battery 0-4) of the switches connecting the batteries to the load

In Area 1 of Figure 4.16, it can be seen that all the batteries are charged, and only when battery[0] is found to be discharged is it put on charge.

In Area 2, the batteries are interchanged sequentially as the Input signals indicate that the one in use is discharged, but without being recharged as battery[0] still needs to be charged.

In Area 3 as and when the Inputs indicate that the batteries have been recharged, it is disconnected from the charging system by opening the relevant switch and connecting that of the next battery.

In Chapter 2 it is shown that when a battery is in operating mode and reaches its minimum threshold it is always the case that the next battery is ready to operate if everything operates as designed. However, a battery failure could lead to the system shutting down permanently, as the power system would fail irreparably.

4.6.2 BATTERY_MANAGEMENT() with Fault Protection

After verifying that the function for battery management, explained in Section 4.6.1, worked out the inclusion of a battery fault protection system. This is because should the fault condition of a battery arise (which would result as discharged), the system would unnecessarily try to recharge it. As a result, the device would encounter shutdown the moment all other batteries are found to be discharged.

The protection system consists of inserting an inverting threshold comparator[9] placed between the charging system and the batteries so that it can send a signal to the Microcontroller the moment the battery is completely discharged.

To do this, the comparator threshold is set at 3V, which is below the Cut-off Voltage of the batteries ($3.2V$). When the threshold is reached, the Microcontroller will consider the battery faulty and exclude it from the system.

The updated code is shown in Listing 4.3 and the operating principle is very similar to that previously explained in Section 4.6.1.

The first change is the condition set at the beginning. This condition is based on the flag that indicates whether a battery is already in a fault state, and it mandates that the management operations be performed only in the case where the fault flag of the current battery is not active.

The activation of the flag occurs after the battery is connected to the charging system, in case the threshold comparator should send the fault signal. Also upon activation of the flag, the battery is disconnected from the charging system and put into standby mode, so that it does not affect subsequent cycles with the management of other batteries. Again, to ensure that no energy is wasted during the charging phase, the number of charged batteries required to activate the UWB module is reduced by one.

[9]When the constant reference voltage is applied to the + input, the comparator is said to be inverting: in this case, the output voltage of the op-amp goes high when the variable input (applied on the - input) drops below the threshold voltage.

```
void BATTERY_MANAGEMENT(){

    //battery_switch_state[x] == 0 battery in charge: op[x]=LOW  && ch[x]=HIGH
    //battery_switch_state[x] == 1 battery waiting:   op[x]=LOW  && ch[x]=LOW
    //battery_switch_state[x] == 2 battery operating: op[x]=HIGH && ch[x]=LOW
    //battery_fault_flag[x]   == 1 battery fault:     don't operate on this battery

  for (int i=0; i<5; i++){
    // Current battery threshold acquisition
    threshold_comp_value[i] = digitalRead(I_threshold_comparator[i]);
    // Go to the next battery if the current is at fault
    if(battery_fault_flag[i] == 0){
      // Change battery if the current is connected to the load and has a low charge level
      if(threshold_comp_value[i] == LOW && battery_switch_state[i] == 2){
        digitalWrite(O_battery_operating_switch[battery_ready[0]], HIGH);
        battery_switch_state[battery_ready[0]] = 2;
        // Update the queue of charged batteries
        for (int k=0; k<5; k++){
          battery_ready[k] = battery_ready[k+1];
        }
        // Current battery [i] in standby (ready to be charged)
        digitalWrite(O_battery_operating_switch[i], LOW);
        battery_switch_state[i] = 1;
        delay(10);
      }
      // Current battery on standby if it is fully charged
      else if (threshold_comp_value[i] == HIGH && battery_switch_state[i] == 0){
        digitalWrite(O_battery_charging_switch[i], LOW);
        battery_switch_state[i] = 1;
        delay(10);
        // Add the battery charged to the queue of the charged battery vector
        for (int j=0; j<5; j++){
          if(battery_ready[j] == 99){
            battery_ready[j] = i;
            j=6;
          }
        }
      }
      // Recharge the current battery if low charge level and no other is connected to the recharge system
      else if(threshold_comp_value[i] == LOW && battery_switch_state[i] == 1 && sum1.sum(5,battery_switch_state)>5){
        digitalWrite(O_battery_charging_switch[i], HIGH);
        battery_switch_state[i] = 0;
        delay(10);
      }
      // If battery fault activate the flag and disconnect the battery from the recharge system
      if(battery_switch_state[i] == 0 && battery_switch_state[i] == 0 && digitalRead(I_battery_fault) == HIGH){
        digitalWrite(O_battery_charging_switch[i], LOW);
        battery_fault_flag[i] = 1;
        battery_switch_state[i] = 1;
        n_battery_to_work -= 1;
      }
    }
  }
}
```

Listing 4.3: Battery management function with fault protection

With this procedure, the Microcontroller avoids connecting the battery to the charging system, but it is also necessary to modify the process that decides which battery is to be connected to the load.

The first step is to create a queue of charged batteries. When the battery

connected to the load is found to be discharged, the Microcontroller takes the index of the first available battery from the queue and connects the corresponding battery to the load, the queue flows, so that the index of the battery just used is removed.

The queue is updated when a battery is fully recharged by going to the bottom and inserting the index corresponding to the newly recharged battery.

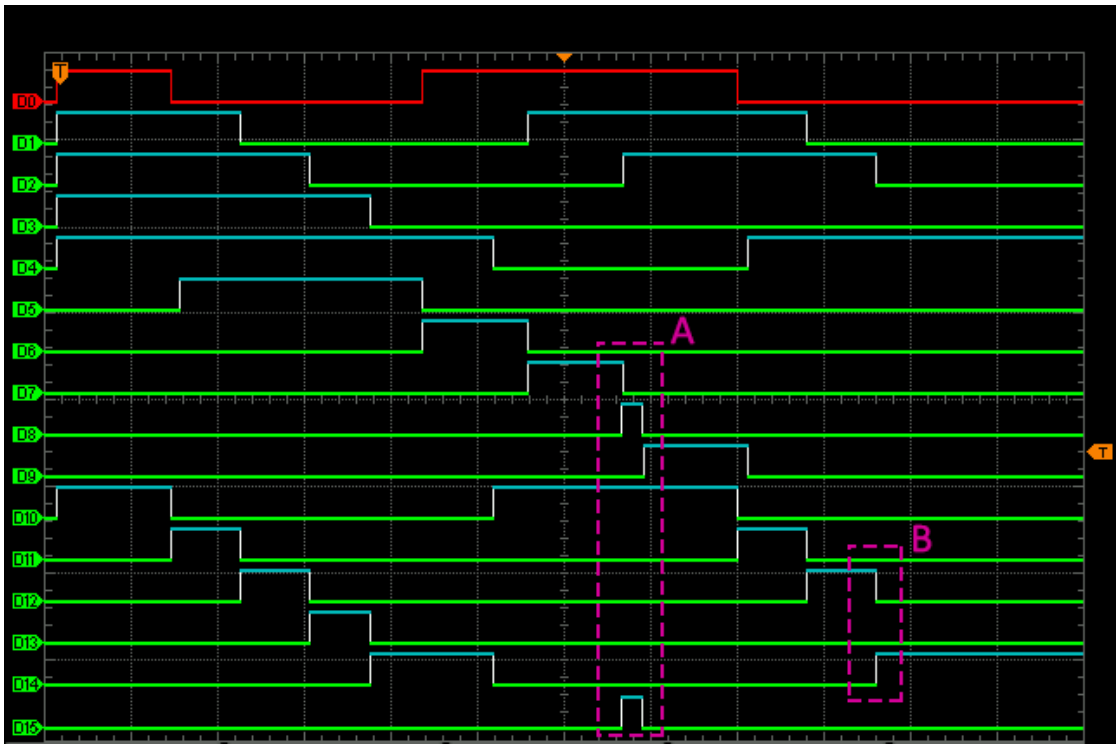


Figure 4.17: Simulation of the battery management function: D0-D4 (in order of battery 0-4) are the Inputs of the threshold comparators; D5-D9 (in order of battery 0-4) are the Outputs of the switches connecting the batteries to the charging system; D10-D14 (in order of battery 0-4) are the Outputs of the switches connecting the batteries to the load; D15 is the Input that indicates a fault in the current battery

As can be seen in Area A of Figure 4.17, when the fault of battery No.3 (whose input related to threshold voltage is D3, and the outputs of the switches connecting it to the charging system and load are on D8 and D13, respectively) occurs, it is disconnected from the charging system.

In Area B, it can be seen that when battery No.2 is discharged, it is switched directly to No.4, without using No.3, which had experienced the fault.

4.6.3 UWB_MANAGEMENT()

Once the battery management is completed, the power on and off of the UWB module is managed with the $UWB\_MANAGEMENT()$ function, which is shown in Listing 4.4.

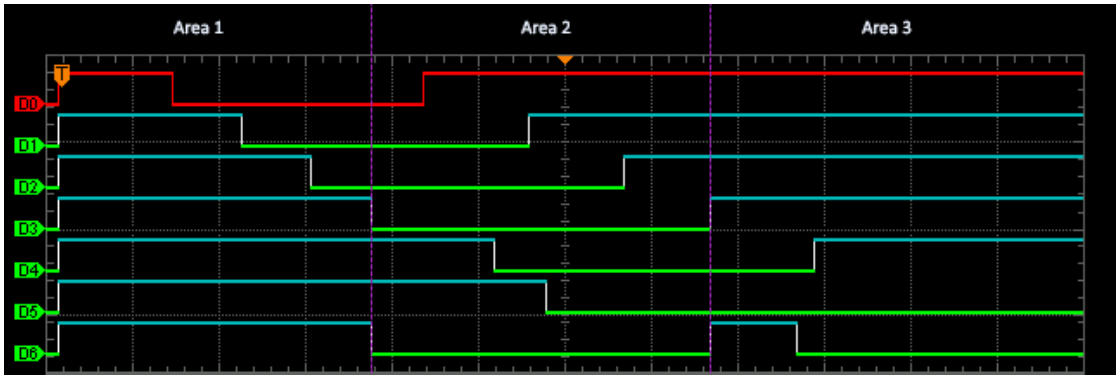
```
void UWB_MANAGEMENT(){

  // Standby signal to switch on/off UWB board according to SOC batteries
  battery_full_charge = 0;
  for(int i=0; i<5; i++){
    if(threshold_comp_value[i] == HIGH){
      battery_full_charge += 1;
    }
  }
  // When battery are ready -> power on UWB
  if((battery_full_charge>n_battery_to_work-1 && other_not_ready_flag == 0 && digitalRead(O_switch_UWB) == LOW
      ) || first_activation == 1){
    digitalWrite(O_switch_UWB, HIGH);
    delay(10);
    // Start the timer 1 and increase the counter
    timer_attempt.set(UWB_attempt_delay);
    other_not_ready_count += 1;
  }
  // Low charge Batteries -> shootdown UWB
  else if(battery_full_charge<2 && digitalRead(O_switch_UWB) == HIGH) {
    digitalWrite(O_switch_UWB, LOW);
    delay(10);
  }
  // Turn OFF the UWB if the signal from the other device is missing for more than 1 min
  if(digitalRead(O_switch_UWB) == HIGH && digitalRead(I_all_device_ready) == LOW && timer_attempt.check() == 1){
      // Start the timer for the standby period
      timer_attempt_delay.set(UWB_activation_delay1);
      delay(10);
      // Increase timer after 5 attempts
      if(other_not_ready_count == 5){
        timer_attempt_delay.set(UWB_activation_delay2);
        delay(10);
      }
      digitalWrite(O_switch_UWB, LOW);
      delay(10);
      other_not_ready_flag = 1;
  }
  else if(digitalRead(O_switch_UWB) == HIGH && digitalRead(I_all_device_ready) == HIGH){
    other_not_ready_count = 0;
  }
  // Reset the flag to do another attempt after the delay
  if(other_not_ready_flag == 1 && timer_attempt_delay.check() == 1){
    other_not_ready_flag = 0;
  }
}
```

Listing 4.4: UWB module managment

The variable $battery\_full\_charge$ indicates the number of currently charged batteries and is used to turn ON/OFF the UWB module according to the total charge. This behavior can be seen in Area 1 of Figure 4.18a.

When four of the five batteries are found to be charged, the switch of the UWB module is closed to check whether the other devices are also ready for use, as shown in Area 1 of Figure 4.18b. On the first power on of the device, the condition "$first\_activation == 1$" is always true to ensure UWB module power on, it will be reset later at the end of the *loop*() function. When only one battery is found to be charged the UWB module is turned off.

As in Area 2 of Figure 4.18a, it is not possible to power on the UWB module again, until at least four batteries have been recharged as in Area 3. The number of batteries needed is set as a variable at the beginning and can be changed if a malfunction occurs in one of the batteries as described in Section 4.6.2.



(a) D0-D4 (in order of battery 0-4) are the Inputs of the threshold comparators; D5 is the Input that provides information about the other UWB modules; D6 is the Output related to the switch that allows the UWB module to be switched on and off

(b) D5 is the Input that provides information about the other UWB modules; D6 is the Output related to the switch that allows the UWB module to be switched on and off

Figure 4.18: Simulation of the UWB management function

However, turning on the UWB device when the other modules are not yet ready is an unnecessary waste of energy. In addition, uncoordinated times for powering up the modules lead to premature shutdown of one or more anchors relative to the others, which can cause a ripple effect in which the sum of premature shutdowns could lead to modules never powering up simultaneously.

When the UWB module is on but does not receive the backup signal related to the other UWB modules a timer is triggered with $timer\_attempt.set()$. If this timer exceeds one minute, the variable $other\_not\_ready_flag$ is activated, which prevents the UWB module from being turned on for a period managed by a second timer ($timer_attempt_delay$). Power-ups continue every hour for five times, and then the delay between attempts to one day is increased. Daily attempts continue until the Input signal indicates that all modules can operate. This avoids excessive consumption even during nighttime hours.

In Area 2 of Figure 4.18b, it can also be seen that if the signal is interrupted for a short period (less than 1 minute), for any reason unrelated to the actual operation of the other devices, the UWB module remains in operation without performing a shutdown.

If the interruption time is more than 1 minute, it will be necessary to wait 1 hour to make a new attempt to switch on, as shown in Area 3 of Figure 4.18b.

4.6.4 OPEN_DEVICE() & CLOSE_DEVICE()

The last two functions are related to the opening and closing of HasuNoHana by using the servo motor. The code related to these two functions is in Listing 4.5 and 4.6.

As can be seen in Listing 4.1, these two functions are called only during the use of the UWB module is notified to open or close the device, via its two Inputs.

```
void OPEN_DEVICE(){

  digitalWrite(O_switch_limitswitch, HIGH);
  delay(1000);
  if(digitalRead(I_limit_switch_signal) == HIGH){
    digitalWrite(O_switch_servo, HIGH);
    delay(1000);
    servo_motor.write(servo_opening);
    delay(7000);
    servo_motor.write(servo_stop);
    delay(100);
    // Open switch to avoid energy loss
    digitalWrite(O_switch_servo, LOW);
    // Opened device signal activated
    digitalWrite(O_device_open, HIGH);
  }
  digitalWrite(O_switch_limitswitch, LOW);
}
```

Listing 4.5: Open device function

Regarding the $OPEN\_DEVICE()$ function, the contact of the switch connected to the limit switch is closed to check the status of the device (remember that the servo motor and limit switch are normally disconnected to avoid wasted power).

If the device is found to be closed (Input of the limit switch HIGH) then proceed by turning on the servo and sending the motion command for 7 seconds (time required to make the opening).

After that servo motor and limit, switch are turned off and the *O_device_open* signal is set to HIGH to notify the UWB module that the opening has occurred.

```
void CLOSE_DEVICE(){

  digitalWrite(O_switch_limitswitch, HIGH);
  delay(1000);
  if(digitalRead(I_limit_switch_signal) == LOW){
    digitalWrite(O_switch_servo, HIGH);
    delay(1000);
    servo_motor.write(servo_closing);
    while(digitalRead(I_limit_switch_signal) == LOW){
      delay(100);
    }
    servo_motor.write(servo_stop);
    delay(100);
    // Open switch to avoid energy loss
    digitalWrite(O_switch_servo, LOW);

    // Closed device signal activated (not open)
    digitalWrite(O_device_open, LOW);
  }
  digitalWrite(O_switch_limitswitch, LOW);
}
```

Listing 4.6: Close device function

The closure is done with the same principle using the $CLOSE\_DEVICE()$ function, but to avoid forcing closure, a while loop is inserted that is interrupted when the limit switch signal becomes HIGH (i.e., when the device is effectively closed). This is because the servo movement may not always be accurate.

Instead of using a timer as in the open function, you wait for one of the hooks on cylinder No.5 to go to activate the limit switch signal by pressing it.

The usefulness of the limit switch, however, is also to prevent the two functions from being activated when not needed. As shown in Figure 4.19, if the device is found to have already been opened or closed, it is needed again to perform the same open/close operation, the limit switch signal prevents the operation from being repeated. An opening can only be performed after closing and vice versa.

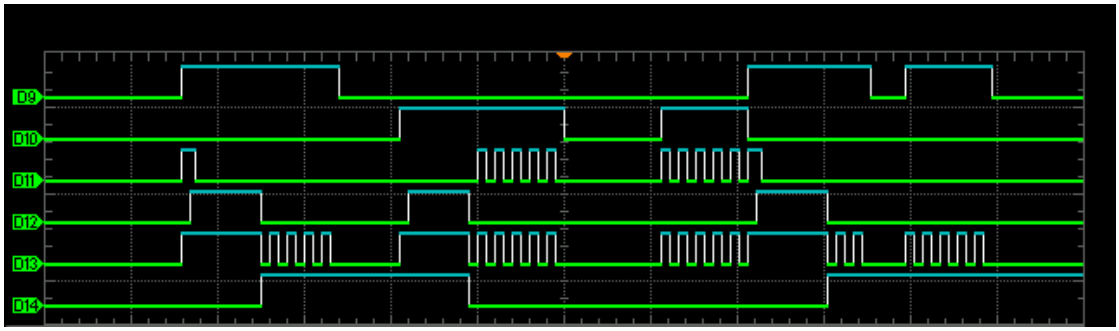
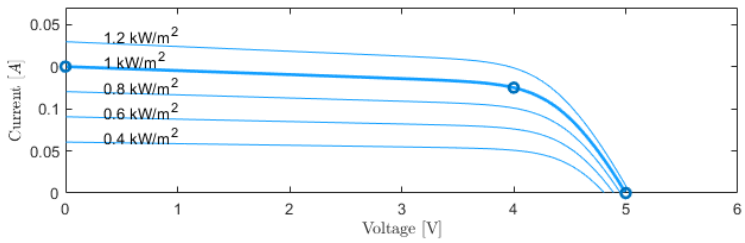


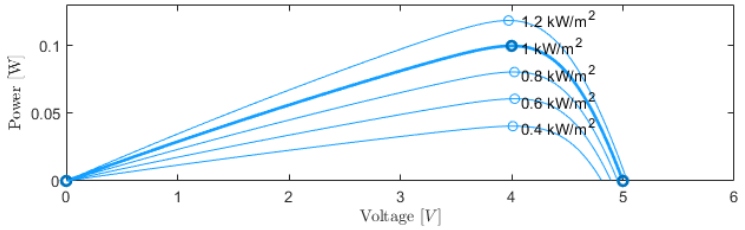
Figure 4.19: Simulation of Open and Close device function: D9 is the device open command input from the UWB module; D10 is the device close command input from the UWB module; D11 is the limit switch input signal; D12-D13 are the Output signals to turn the servo motor and the limit switch on and off, respectively; D14 is the Output that communicates the current state (open/closed) of the device to the UWB module

4.7 PV Support Boards

The PV Support Boards are necessary to use PVs at their maximum efficiency. In Figure 4.20 it is possible to see that maximum power is obtained at a specific working voltage that varies slightly as irradiance varies.



(a) Voltage-Current curves



(b) Voltage-Power curves

Figure 4.20: Characteristic curves of chosen PV at different irradiation values

To maximize the solar energy provided by the panels, it is necessary to use a technique to maximize the power output of a device working with a variable source. One of these techniques is Maximum Power Point Tracking also called MPPT.

The MPPT method uses the Voltage-Current curves shown in Figure 4.20a, by varying the voltage by a small amount, either positively or negatively, and evaluating the power from the actual and previous steps to assess with which of the two voltages the greatest value is obtained. If the power increases with the variation, the voltage will continue to vary with the same sign until the maximum obtainable value is reached, otherwise the sign of the variation is reversed.

The algorithm may follow different canons of variation depending on the design. There are both feedforward[10] and feedback methods[11].[23]

The output of the MPPT has a variable voltage then, but it must be stabilized at a value higher than that of the batteries at 100% charge, to be sure of charging them.

As seen in Section 4.2.1 the maximum voltage that the batteries can reach is $4.2V$, but it is decided to stop charging when $4.1V$ is reached to avoid wasting current. This imposes a minimum limit, but taking into account possible voltage drops it is better to choose a higher voltage.

A reasonable output voltage is 5V, which can be achieved using the same DC-CD converter used to regulate the output of the power supply and described in Section 4.1.

It must be taken into account that PVs have different orientations from each other because of what is said in Section 2.4. As a result, the irradiance and consequently the working point of each PV may not coincide (there are times when some are off and others are working). For this reason, it is better to use one MPPT

[10]The term feedforward methods denotes all those methods that do not require the measurement of power output from the PV panel during operation. These methods generally require less computational effort and lower the cost of the sensors. They involve, however, reduced panel efficiency since the panel rarely reaches the MPP.

[11]The term feedback-based refers to those methods that require controlling the power output from PVs during operations. The computational effort and cost of the sensors to be used are greater, but they provide greater efficiency in achieving the MMP.

for each PV so that management occurs independently of the others.

Therefore, the board designed in Figure 4.21 is to be considered about one PV, and it follows that five PV Support Boards will be needed.

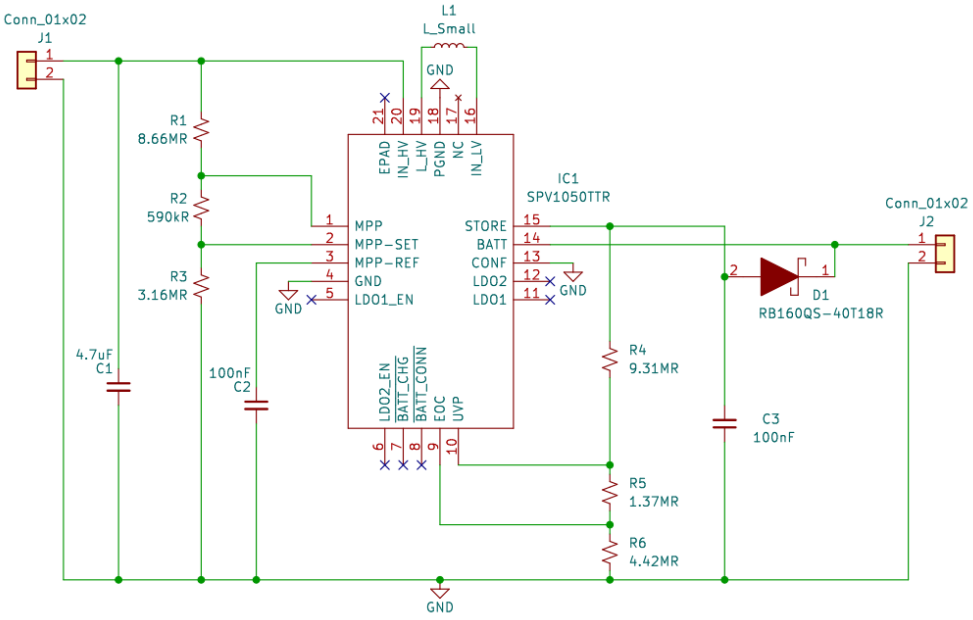


Figure 4.21: PV Support Board's electrical configuration J1 is the input connector, while inputs 1 and 2 represent the positive and negative voltages of the PV, respectively. J2 is the motherboard's output connector, with inputs 1 and 2 representing the positive voltage and ground

The best candidate to perform this task is the SPV1050TTR device by "STMicroelectronics ®" corresponding to component IC1 in Figure 4.21. By enabling MPPT mode with resistors R1-R3 (adjusted according to PV data) and low and high protection triggers through resistors R4-R6, it is possible to supply the load, which in this case is the battery to be recharged, with the maximum power that can be produced.[24]

This device is selected due to its characteristics. First, it is a low-power device and in buck-boost configuration needs a current on the order of μA, taking it directly from what PVs are capable of producing. In addition, the DC-DC converter that regulates the charging output has an efficiency greater than 80%.[24]

4.8 Final Considerations

As in Chapter 3, here too some considerations can be made about future implementations that can be deployed for future prototypes.

The choice to use a Microcontroller with many more Inputs than needed for the prototype also falls on the vision of future implementations. All of the proposals listed here, as well as any other controls implementable, require Inputs and there are currently 30 more available.

First of all, it should be kept in mind that when designing a device intended for a space mission, it is necessary to use space-certified components, and all the components in this chapter are not.

However, since this is the prototype, an attempt is made to minimize production costs, which with specialized components would instead be greatly increased.

Another variable to take into account is as anticipated the temperature of the lunar surface, which has an overhang of more than 300°C. Normal electrical components are not able to withstand such surges and it would be better to keep them at a constant temperature of around 20-25°C. The reason why it would be necessary to implement a heating system, which then needs a significantly larger power supply than the current one.

Another improvement that could be made concerns battery threshold comparators. There are several types of devices called Battery Gauges on the market that are capable of collecting a variety of information from the battery to which they are connected beyond its SoC.

Following what has been seen in Section 2.4.2 about irradiance and insolation, and taking into account considerations made about possible implementations in Section 3.5, in case a decision is made to make the PVs independent in aperture system, it might be interesting to develop a method capable of adjusting the tilt of individual PVs during the Synodic Month. In this way, one could have higher irradiance for a longer period accordingly and higher energy and greater UWB module operating time.

Chapter 5

Devices Placement on Lunar Environment

Having finished prototyping the device itself, it is necessary to consider a method of releasing the Anchors. Initially, the idea is to design a special release system to be mounted on the rover, which requires the use of a storage facility for the Anchors and consequently a considerable use of space. For this reason, this solution is immediately discarded.

Optimizing the space available on the rover and using the means already available to the rover is the most efficient solution, which is why it was considered to use the rover-mounted robotic arm to complete the release of the various Anchors.

The robotic arm mounted on the rover is the "KINOVA Gen3 lite" from KINOVA®[25]. As shown in Figure 5.1 it is a robotic arm equipped with six revolute joints[1], which implies that the arm has six Degrees of Freedom[2] (DoF). Defining a point with six DoF allows its orientation to be defined in addition to its position.

In addition, a gripper capable of grasping objects is mounted as an end effector.

[1]A kinematic pair with one degree of freedom is called a revolute joint. As a constraint, the two bodies can only move in pure rotation around a shared axis.

[2]In physics, the number of Degrees of Freedom of a material point is the number of independent variables required to uniquely determine its position in space.

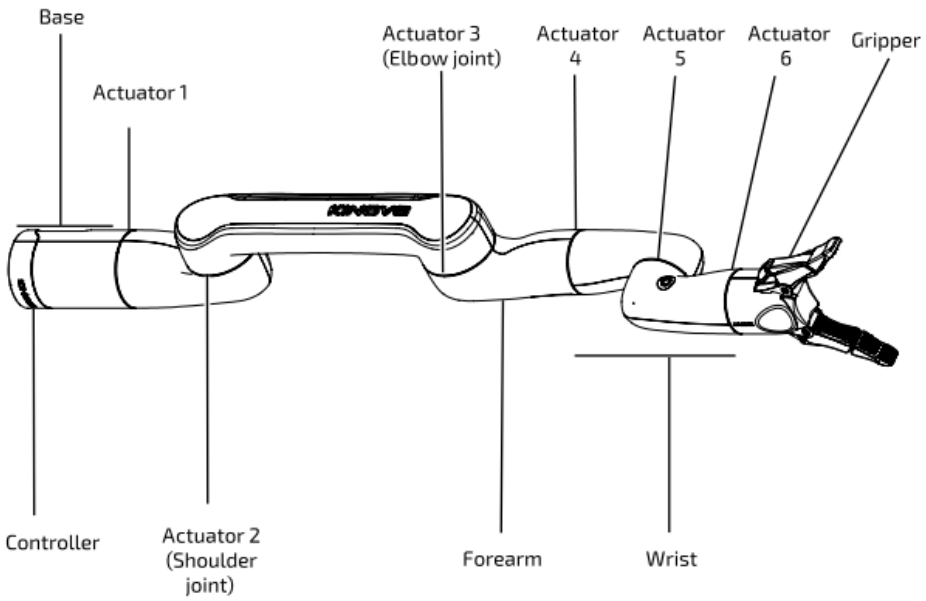


Figure 5.1: Robotic main components[25]

This chapter describes the method to drop and recover HasuNoHana, moreover the results obtained through simulations on Gazebo and physical tests performed with the robotic arm.

5.1 Anchors Recognition

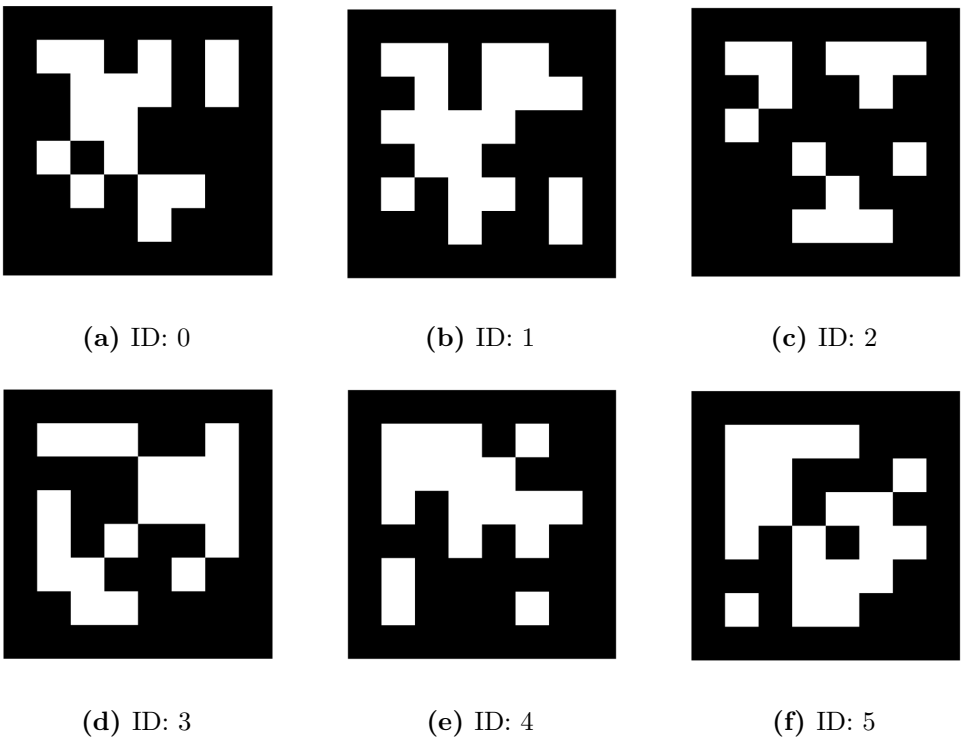
First, it is necessary to talk about the method of recognizing the Anchors. As mentioned in the previous chapters, the designed Anchors can be opened and closed by the servo motor mounted in them, which can then be picked up and relocated at different locations depending on the area where the rover operations are needed.

In addition, each UWB module mounted on the Anchors has a unique number that identifies and differentiates it from the others. So as stated, it is necessary to distinguish one Anchor from another. The method chosen for identification uses the camera mounted on the arm end effector to recognize the tag placed on the various Anchors.

5.1.1 AprilTags

The AprilTag detection software calculates the precise 3D position, orientation, and identity of the tags. In that they are a kind of two-dimensional bar code, AprilTags conceptually resemble QR Codes. Tags can be detected more reliably and over a wider area because they are made to encode much smaller data payloads (between 4 and 12 bits). They are also created for high localization accuracy.

The tag family used for the Anchors is "$tag36h11$" and the tags and their IDs are shown in Figure 5.2.



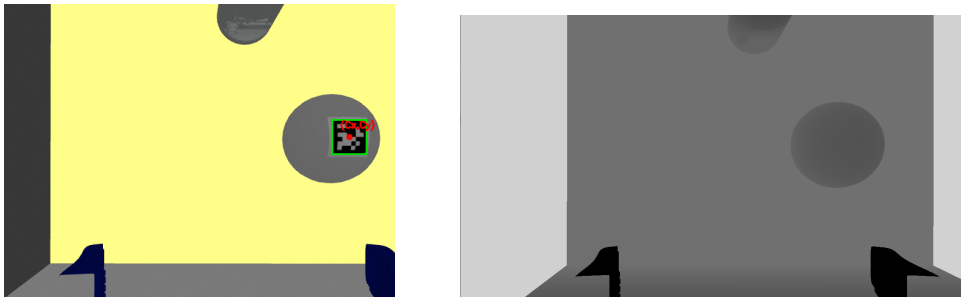
(a) ID: 0 **(b)** ID: 1 **(c)** ID: 2

(d) ID: 3 **(e)** ID: 4 **(f)** ID: 5

Figure 5.2: AprilTAgs from tag36h11 family

The tag recognition procedures are achieved through the use of a "RealSense Depth Camera D435" from Intel Corporation®[26], which is capable of providing color and depth images[3] a shown in Figure 5.3.

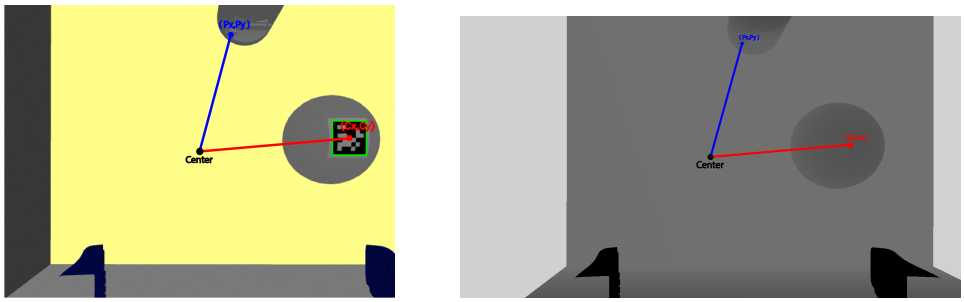
[3]A depth image contains data about the distance between the viewpoint and the objects' surfaces.



(a) Color frame **(b)** Depth frame

Figure 5.3: Tag detection in Gazebo simulation

A problem is that the depth image and the color image are not aligned with each other, as can be seen in Figure 5.3. The two images have different resolutions (640x480 for the color camera and 1280x720 for the depth), but the problem can be solved by remapping the position of the dots in the depth image. In Figure 5.4 that the ratio between the segments in depth and color frame is the same for every point if they start from the center of the two images. Thus, the corresponding pixel in the depth frame can be identified with Formula 5.1.



(a) Color frame **(b)** Depth frame

Figure 5.4: Tag detection in Gazebo simulation with image calibration

$$P_{depth} = P_{center_{depth}} - (P_{center_{color}} - P_{color}) * ratio \quad [pixel] \tag{5.1}$$

In this way, it is possible to collect information about the *distance* between the camera and the center of the tag, which is needed to identify the tag position in the camera reference frame (Formula 5.2). It is also possible to identify the tag orientation.

The data collected by the camera can be processed through the Python libraries "cv2" and "apriltag." With the first library functions, it is possible to manipulate the image data so that the tags can be recognized in a gray-scale[4] image from the AprilTag algorithm. It is easier to identify the presence of a tag in a gray-scale image because the complexity while applying the morphological operations is reduced. More in detail by using these libraries, it is possible to elaborate the information collected by specific topics published from the camera.

Next, a function takes care of identifying the tag by calculating the central pixel indices (Cx, Cy) in the image. As shown in Figure 5.5, the reference frame of the image differs from that of the camera, so it is necessary to perform a transformation via the Formulas 5.2. Also, the *distance* of the tag center in meters is needed, which is obtained from the corresponding pixel in the depth image. By doing so, the newly obtained points (x_{tag}, y_{tag}) belong to the camera reference frame.

$$\begin{cases} x_{tag} = \frac{(Cx - Width_{pixel}/2) * distance}{Focal\_Distance\_X} & [m] \\ y_{tag} = \frac{Cy - (Height_{pixel}/2) * distance}{Focal\_Distance\_Y} & [m] \end{cases} \tag{5.2}$$

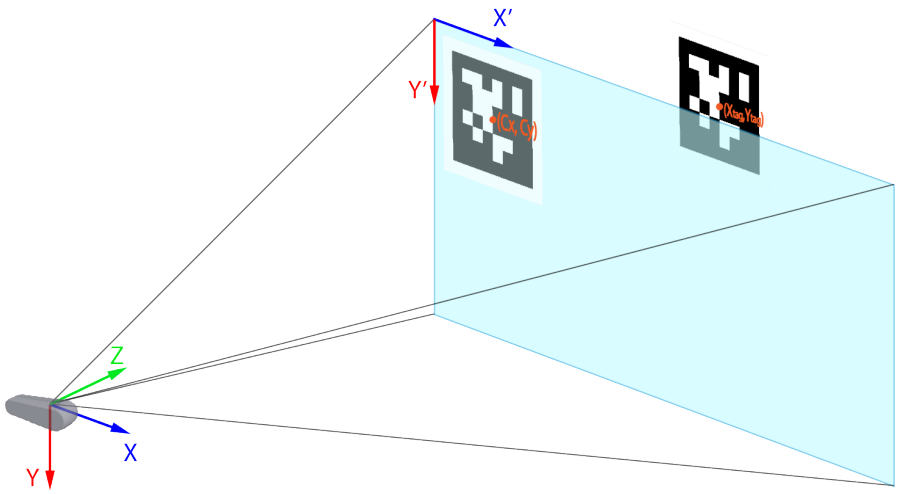


Figure 5.5: (X_{tag}, Y_{tag}) - tag center in the camera reference frame (XYZ) $[m]$, (C_x, C_y) - tag center in the image reference frame (X'Y') $[pixel]$

[4]Gray-scale images are black-and-white raster digital images consisting of a range of grays from white to black.

5.1.2 Recognition Algorithm Limits

However, the AprilTag recognition algorithm has limitations in tag recognition that need to be considered during the tag placement phase on the device and for arm positioning during grasping operations. In particular, the following characteristics need to be considered:[27]

- Tag Rotation does not affect the recognition algorithm.
- The algorithm can identify a tag with a viewpoint angle up to $\pm 80°$, but identification performance decreases as the angle increases if it exceeds $\pm 50°$.
- Tag Size affects recognition accuracy, but it is possible to identify with high accuracy a tag that occupies pixels in the order of a few tens.
- Detection is not guaranteed if partial or complete Border Occlusion occurs.

A point is chosen at the arm placement stage in favor of image acquisition by the camera. From this viewpoint, the camera can identify the presence of the tag on the entire holding tray. Figure 5.4 shows the identified tag to perform the operations described without any problem.

5.2 Pick & Place

The operations to be performed during release and recovery operations work, it is necessary to explain how communication between the various components takes place.

The robot and all its components are managed through Robot Operating System[5] (ROS) nodes and topics. As anticipated in Section 5.1.1 to access image information, subscribers are used for the related topics of depth and color cameras. Similarly, nodes and topics are used for the movements of the robotic arm joints, for the planning of the paths to be followed, for all other functionalities necessary for the movement, as well as for performing simulation using Gazebo and Rviz.

[5]Robot Operating System is a set of frameworks for robot development and programming.

5.2.1 Anchor Release

To perform the pickup of the Anchor from the tray and to release it on the lunar ground, a Python function is written to handle the movements of the robotic arm and to use the tag recognition function described in Section 5.1.1. The function is divided into several steps, all with a check to verify that one is completed before it can move on to the next. The steps are:

- First, the arm reaches the position at which to perform tag recognition. This position is a saved joint position, and it is reached by using inverse kinematics[6] to calculate the path. The position for tag recognition is chosen so that the camera can see the entire tray.
- In this position it is possible to perform tag recognition. This operation uses the camera to take a snapshot of the tray and to return the location of the tag center. The recognition function will identify all tags present, but as shown in Figure 5.2 each tag has its ID. you can then select the desired ID to select its coordinates. Coordinates that refer to the camera frame, must be transformed into the robot's base frame.
- Since the position of the tag in the base frame is acquired, a function creates three waypoints, each along one of the coordinates (xyz). These waypoints are needed to reach the tag's position with linear movements along the axes, leaving the one along z for last. The path created is a polygonal chain (red path in Figure 5.6). Breaking the movement in this way is necessary to prevent the gripper from bumping into the Anchors. Before performing the movement, the gripper is opened
- Subsequently, when the arm is in position on the Anchor, the gripper is closed. In this case, the command to close the gripper is not full closure, but partial closure (75%); if this is not the case, the close function would return an error due to the inability to reach the fully closed position due to the presence of the Anchor.

[6]Inverse kinematics is the process of determining the parameters of an articulated, flexible object that satisfy the attainment of the desired pose, based on the positioning of the ends only.

- If the grab is completed, another linear movement is made along the z coordinate to again avoid collisions against other anchors and to reach a favorable position for the next movement (green path in Figure 5.6).
- Movement to reach the anchor release position is still split into two parts. In reaching the first waypoint, the arm is free to move along an optimal path (orange path in Figure 5.6). The second waypoint has the same xy coordinates as the previous one, while the z varies by a sufficient delta to come close to the ground with the bottom of the Anchor (cyan path in Figure 5.6). Both positions are saved joint positions.
- At this point it is possible to open the tag open the gripper to release the device on the ground.
- Moving the robotic arm along the z coordinate, again allows it to reach a position where the gripper closure can be performed without going to impact the Anchor.
- The final step is to return to a position defined as home position (still saved joint position) with any path.

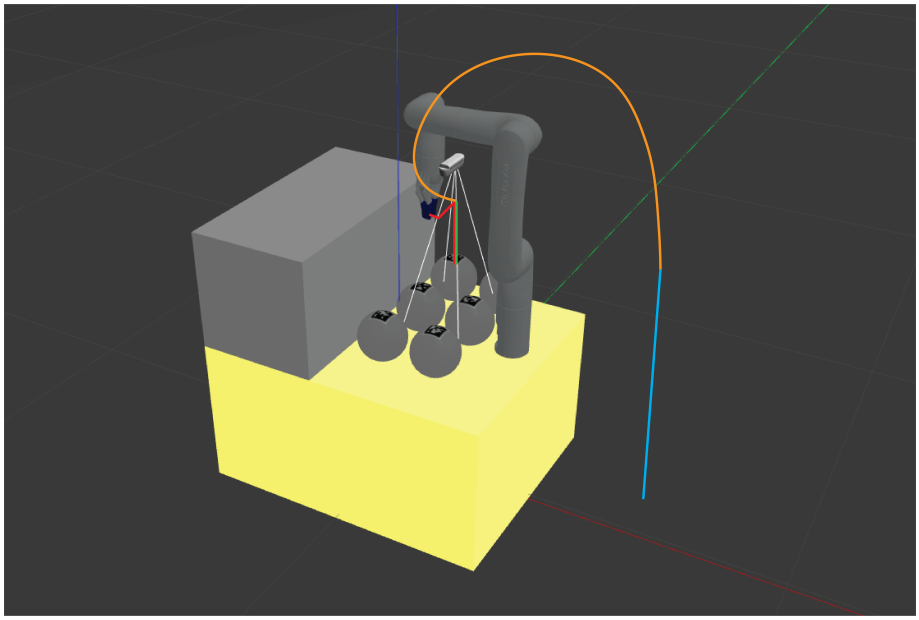


Figure 5.6: Path to perform anchor grab from the pool and release on the ground

5.2.2 Anchor Recovery

For the recovery operation, the rover will get close enough to the Anchor to have it within reach of the robotic arm. The approach is made possible by position calculation using the UWB module.

The rover will reach a distance of a few meters from the Anchor and will send it in signal to perform its closure. After that, with the device closed, the UWB module will remain on to allow the rover to get closer to a position favorable for collection. From this point on, the operations to be performed are almost identical to the release operations described in Section 5.2.1.

The difference is that the initial location to perform tag recognition will not be on the Anchor Storage Tray of course, but with the camera looking at the ground behind the rover. Conversely, once the Anchor is collected, it will be placed back in the tray.

Appendix A

Simulink Simulation Results

This appendix shows all the figures related to the outputs of the simulation described in Chapter 2. In particular, data regarding the power and energy produced by the panels and used by the device are shown. The simulations involve turning on and using the UWB module when a different charge level is reached for each simulation. For all simulations, the irradiance for power generation by PVs is that for the selenographic latitude of 70°.

The first one corresponds to 100% of the total charge, i.e., when all five batteries are charged. This is the worst simulation, as the power produced is significantly less than the others. This is because more than three hours of charging are lost each time the UWB module is turned on. After all, before a battery can be connected to the charging system, it must be discharged. The total operating time, in this case, is less than 60 hours (excluding activation with initial charge).

The second, on the other hand, is when four of the batteries are fully charged, which corresponds to 80% of the total charge. Since you do not fully charge the device, recharging the last battery at each cycle occurs while using the UWB module. The operating time of the UWB module is approximately 72 hours (excluding activation with initial charge).

The third corresponds to reaching 60% of the total charge, that is when three batteries are fully charged. In this case, the difference with the simulation involving 80% lies in the fact that the power-ups occur more frequently, but with a shorter operating time of 6 hours each, for a total of 72 hours (excluding the first activation).

A.1 Five charged batteries required

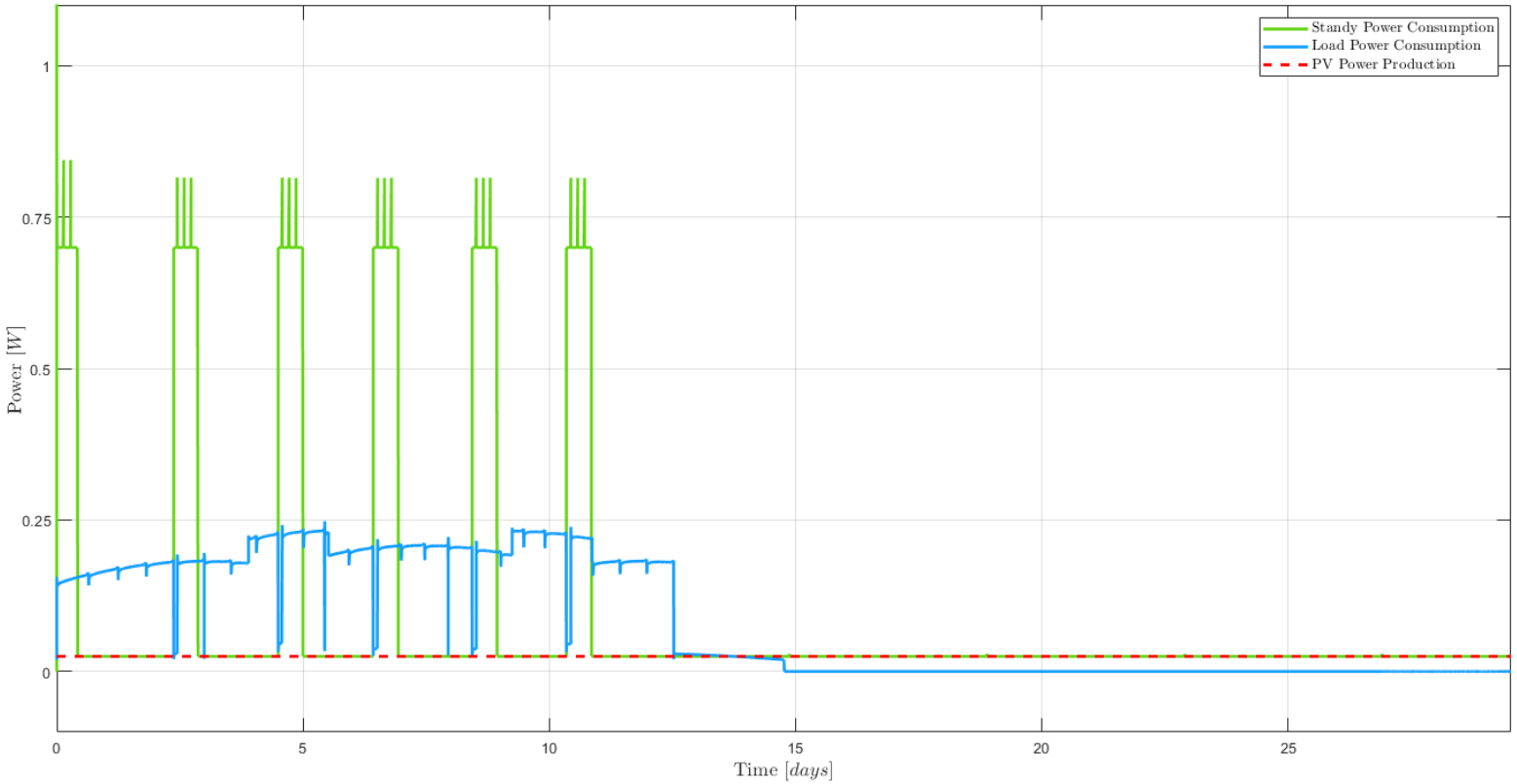


Figure A.1: Power consumption and production trends along the synodic month; the UWB module is turns on at 100% charge

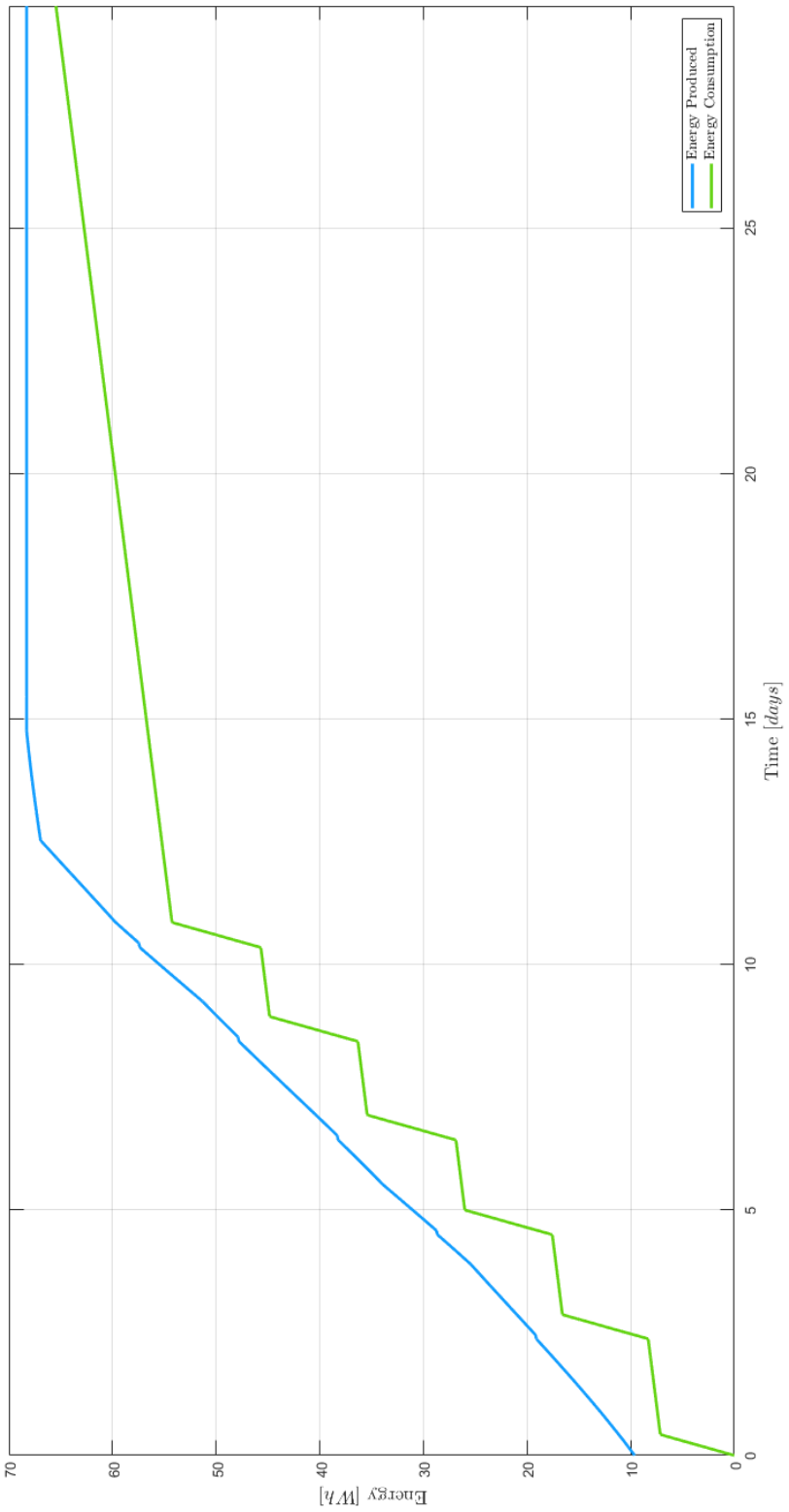
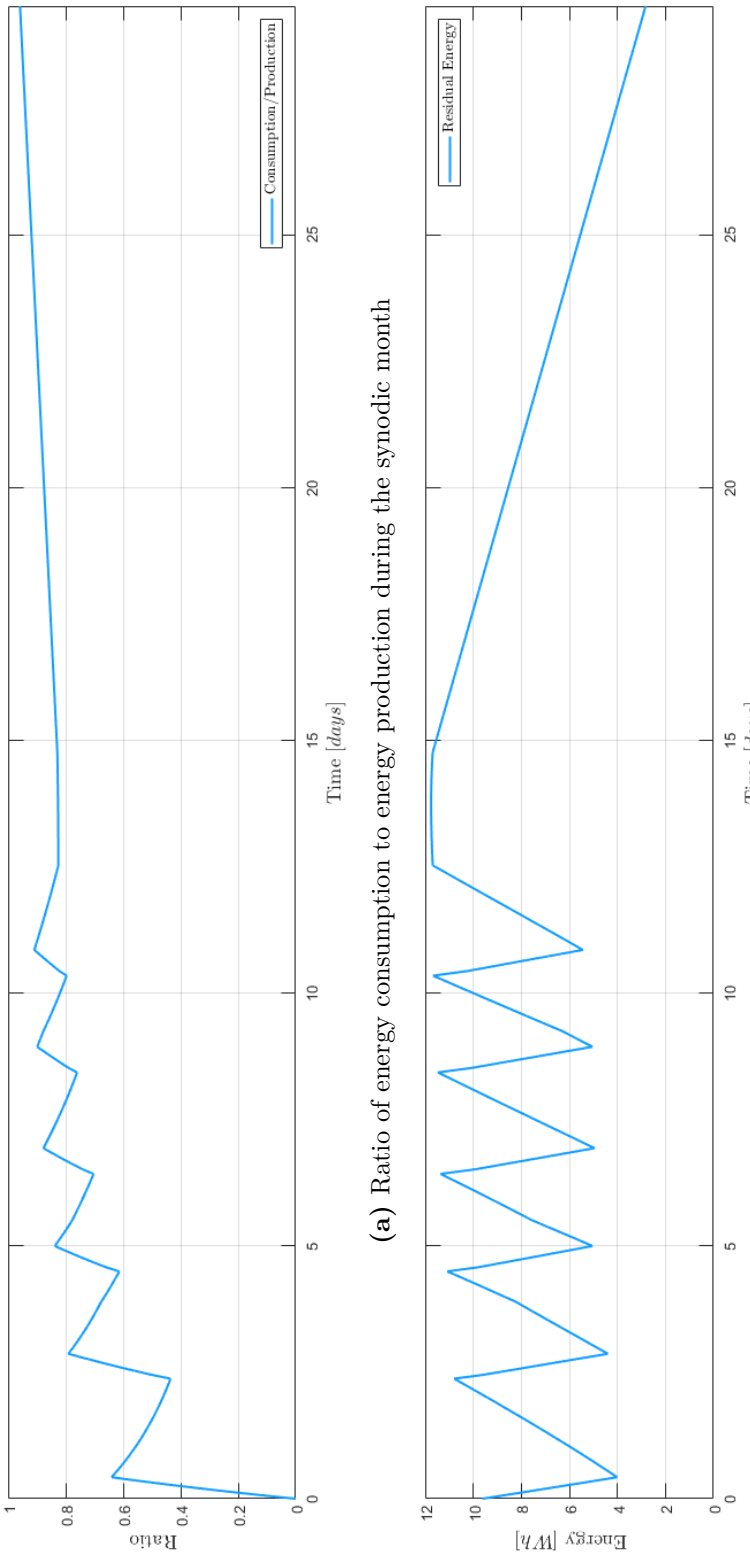


Figure A.2: Energy production and consumption trends during the synodic month; the UWB module turns on at 100% charge



(a) Ratio of energy consumption to energy production during the synodic month

(b) Residual energy in the batteries during the synodic month

Figure A.3: Energy-related output data; the UWB module is turns on at 100% charge

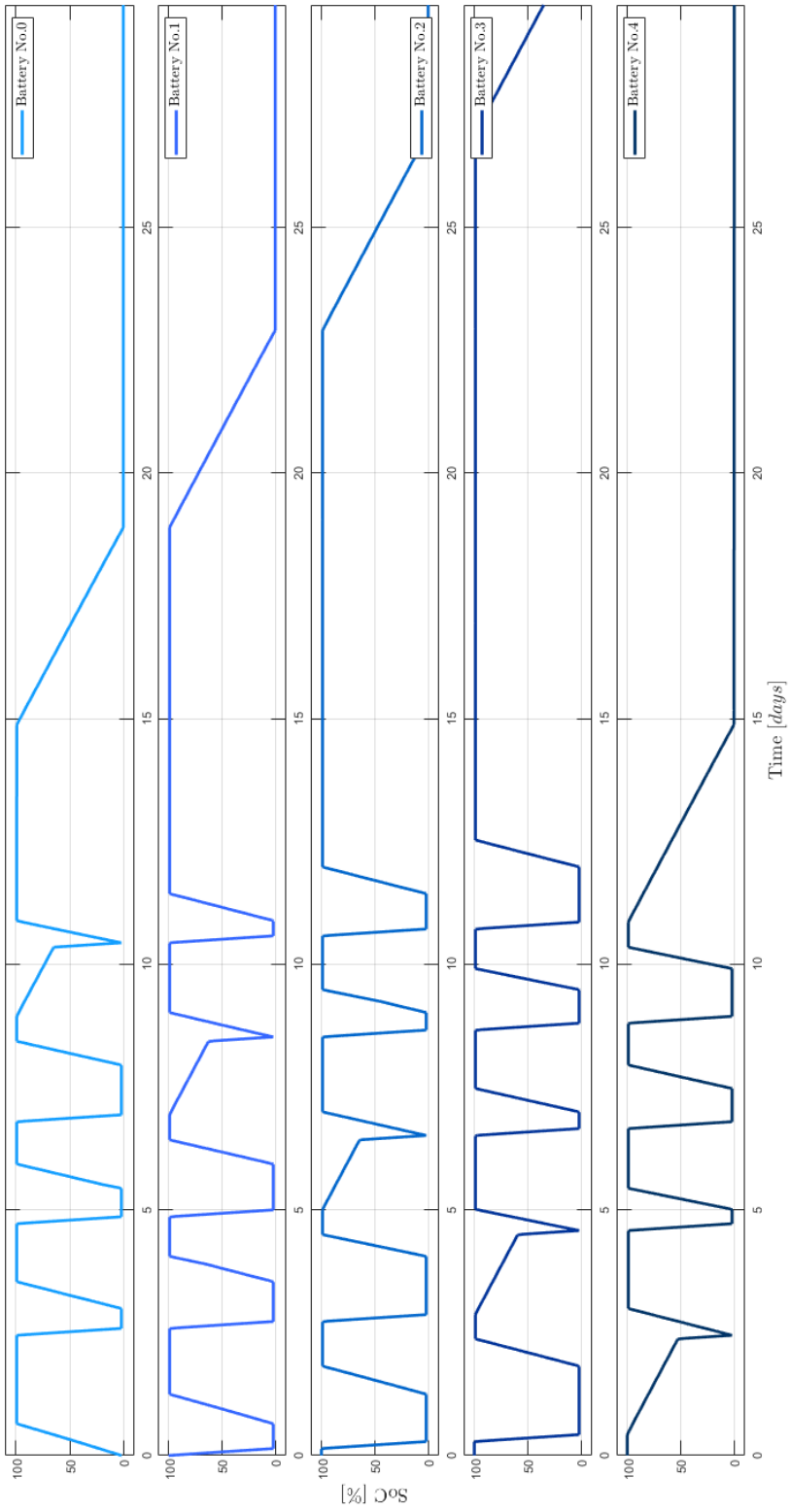


Figure A.4: SoC trends of the five batteries during the synodic month; the UWB module is turns on at 100% charge

A.2 Four charged batteries required

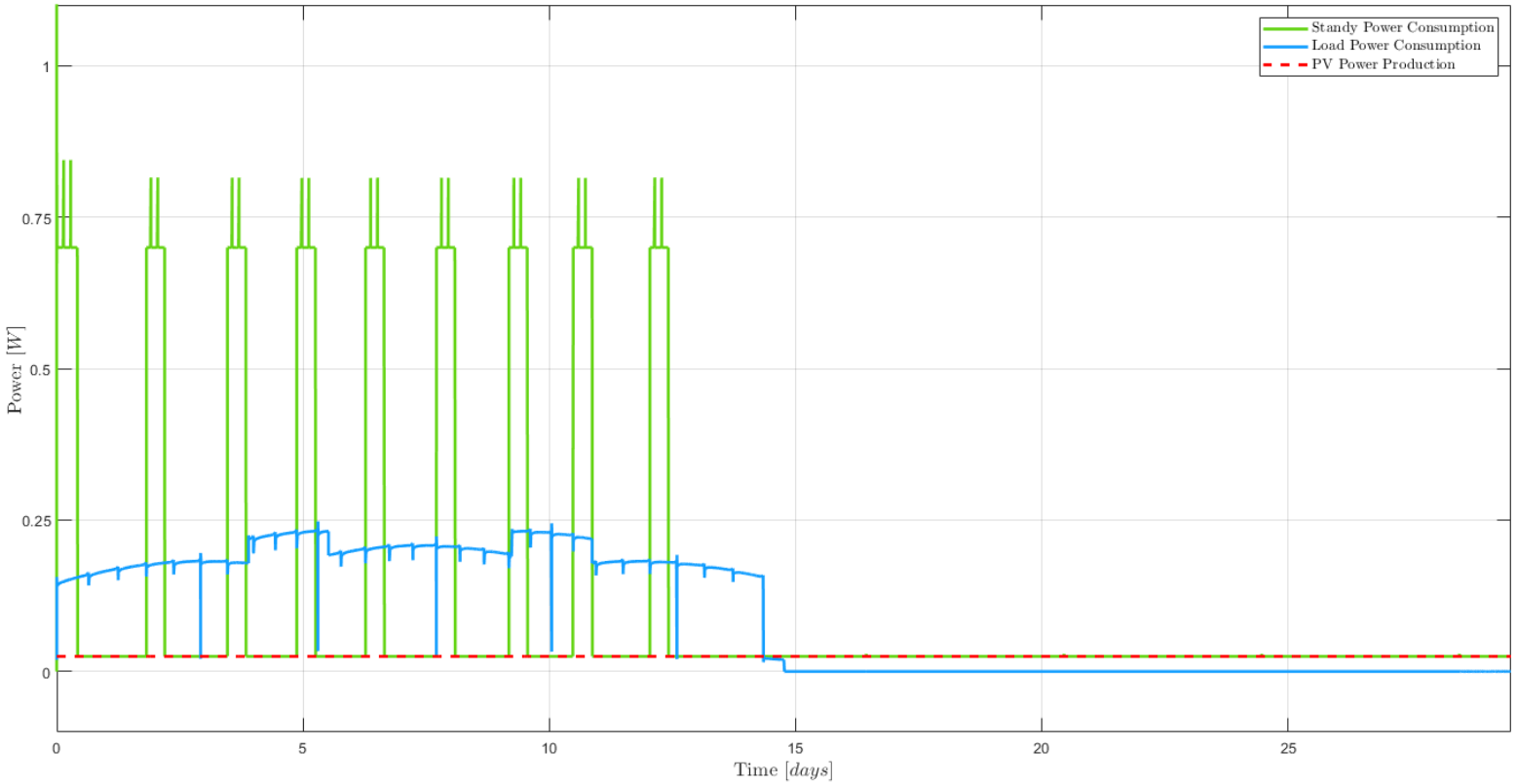


Figure A.5: Power consumption and production trends along the synodic month; the UWB module is turns on at 80% charge

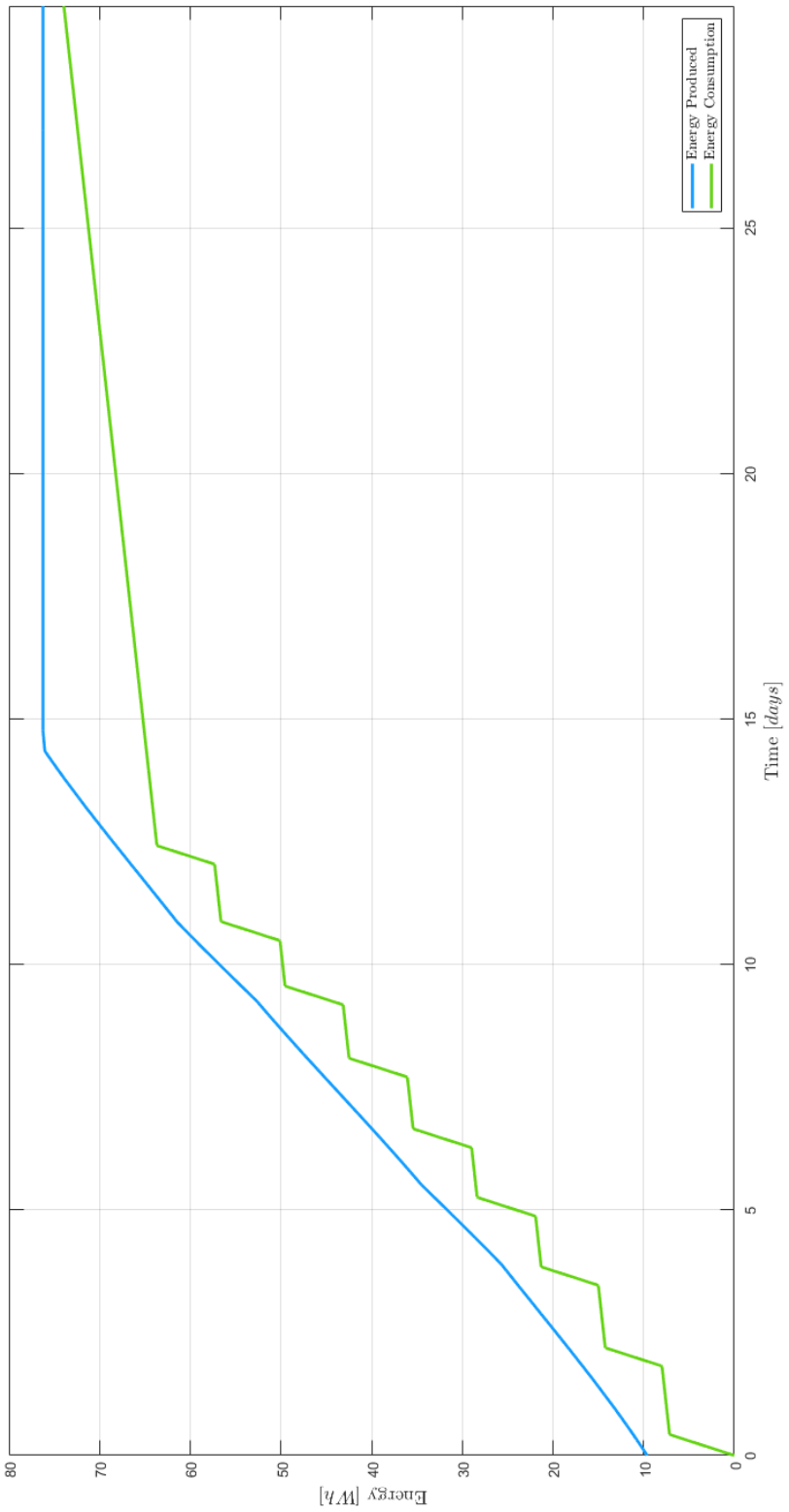
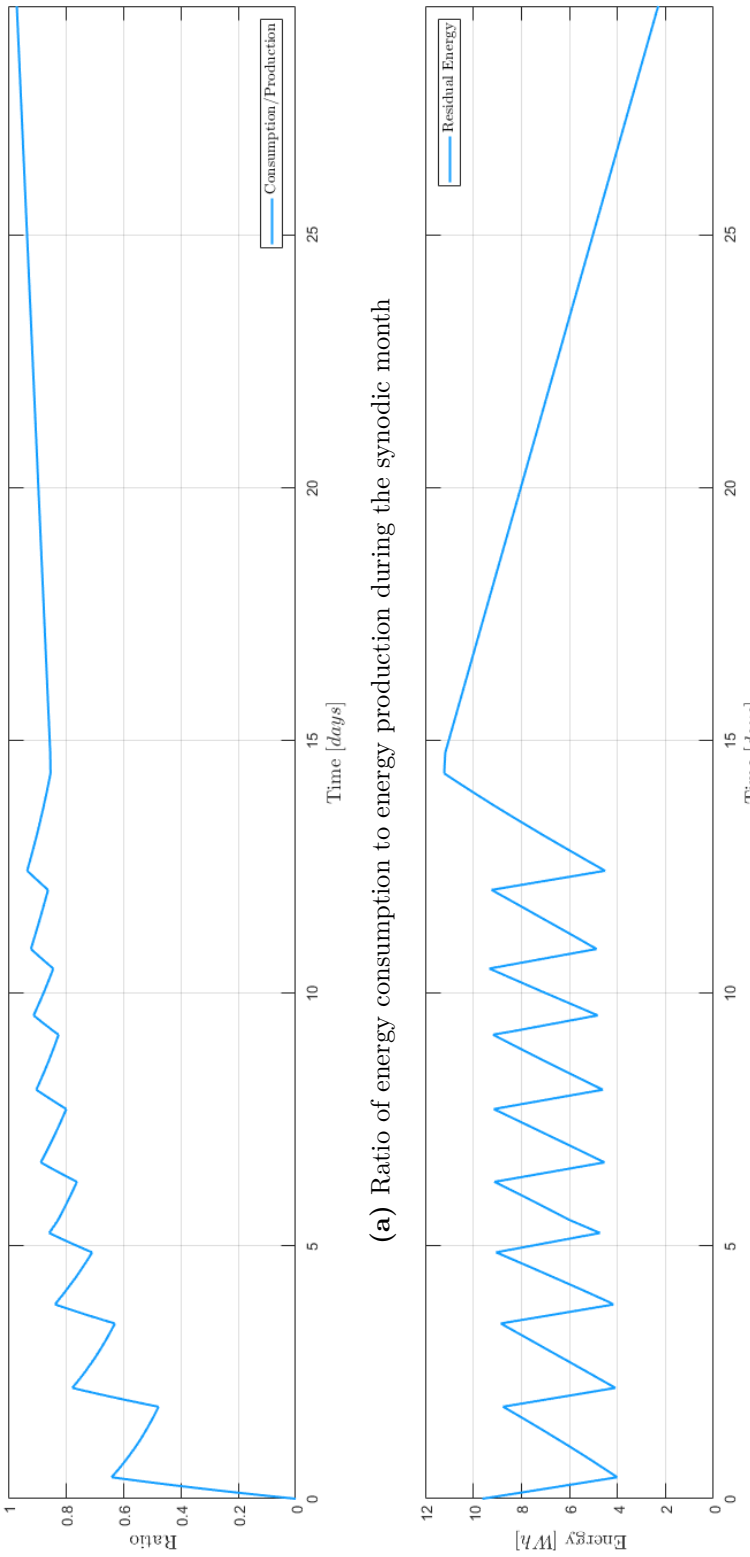


Figure A.6: Energy production and consumption trends during the synodic month; the UWB module is turns on at 80% charge



(a) Ratio of energy consumption to energy production during the synodic month

(b) Residual energy in the batteries during the synodic month

Figure A.7: Energy-related output data; the UWB module is turns on at 80% charge

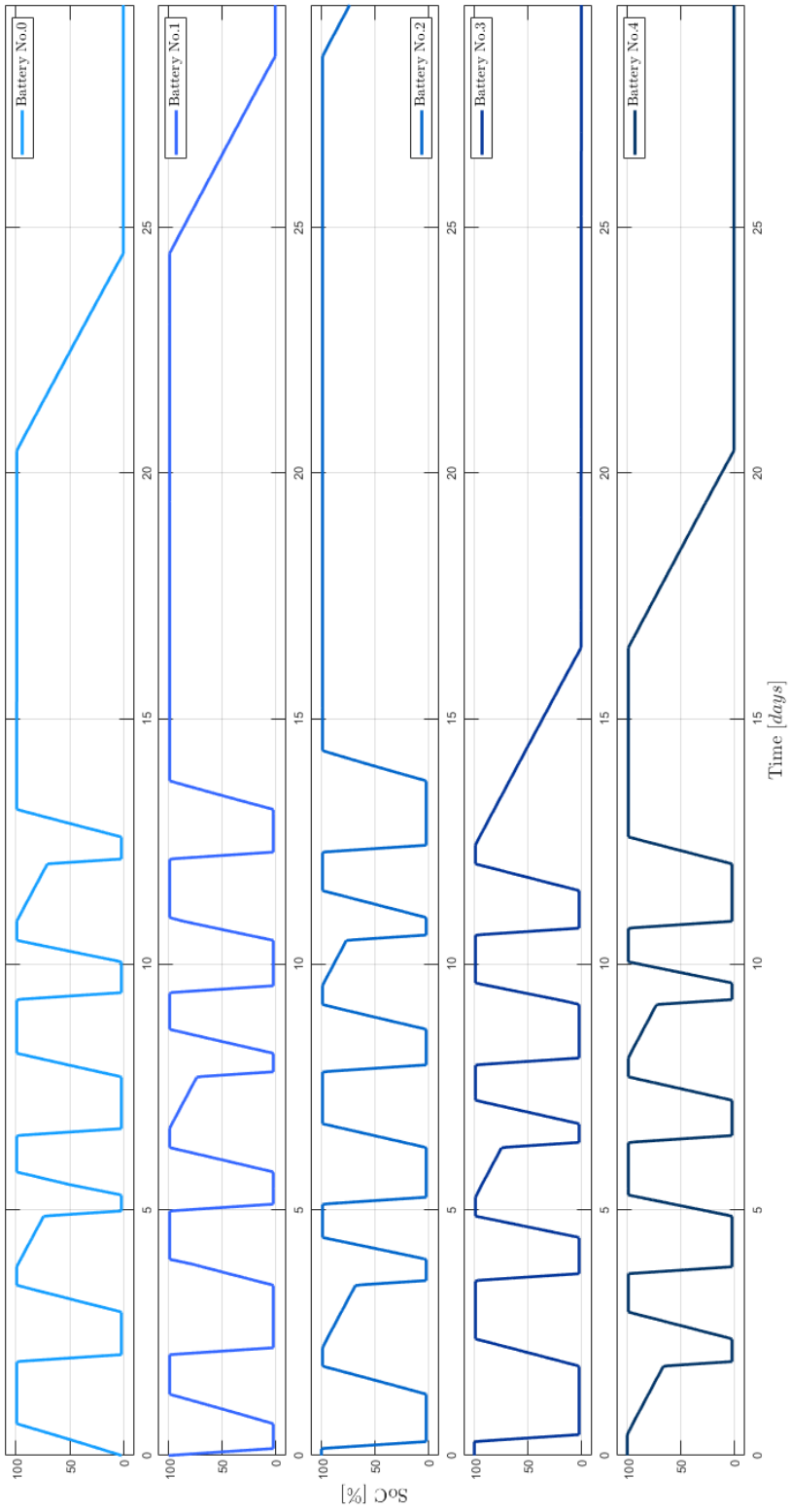


Figure A.8: SoC trends of the five batteries during the synodic month; the UWB module is turns on at 80% charge

A.3 Three charged batteries required

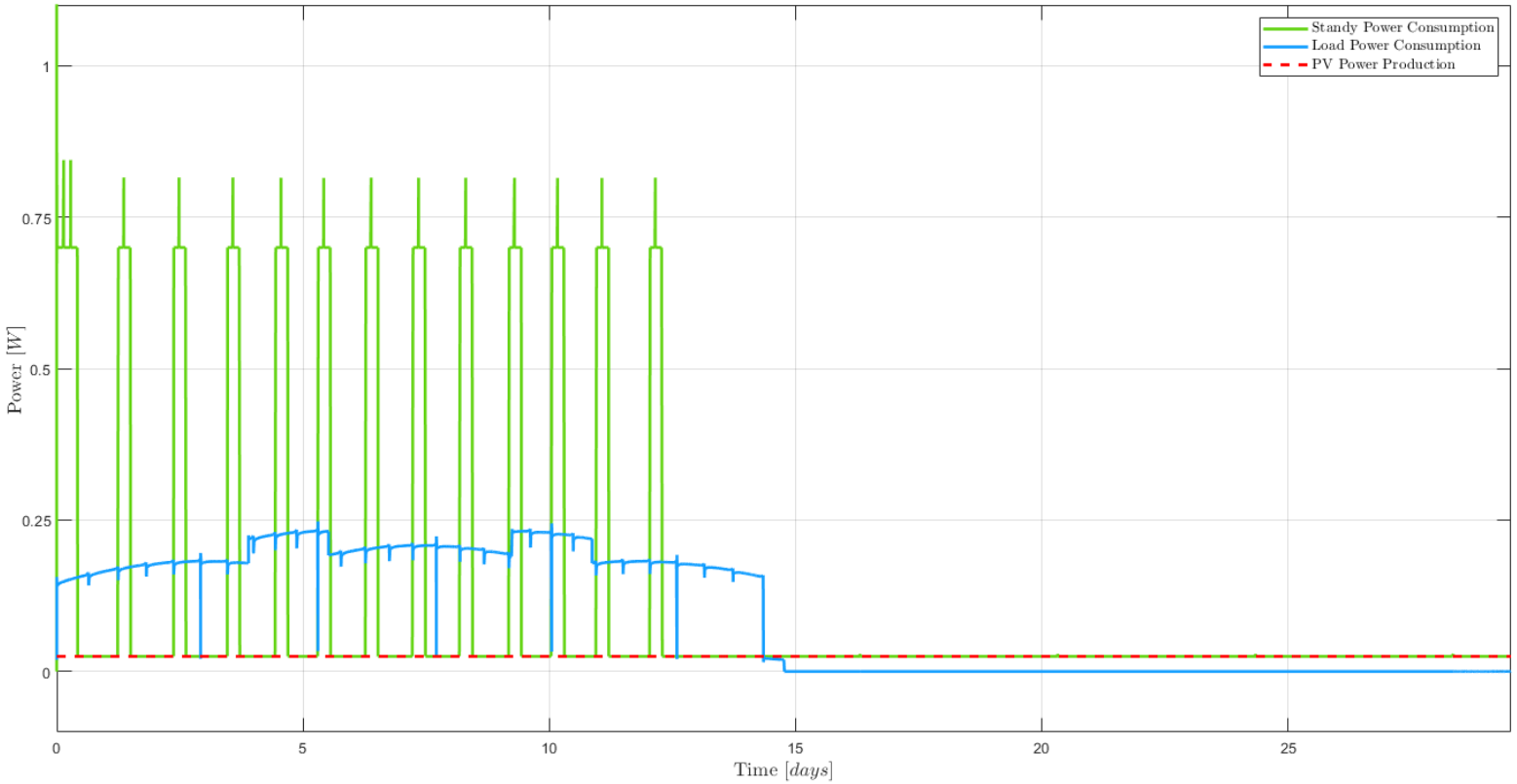


Figure A.9: Power consumption and production trends along the synodic month; the UWB module is turns on at 60% charge

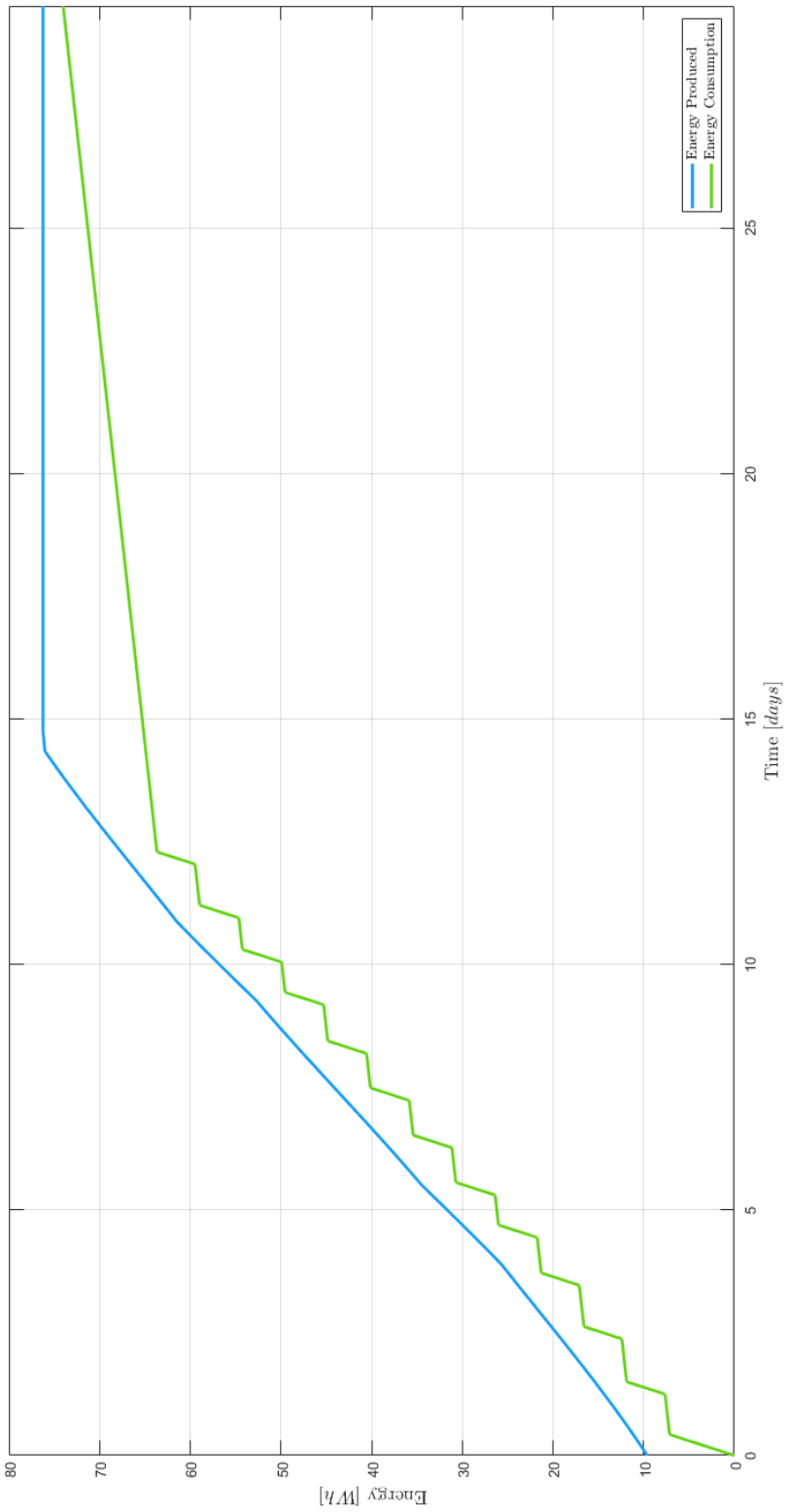
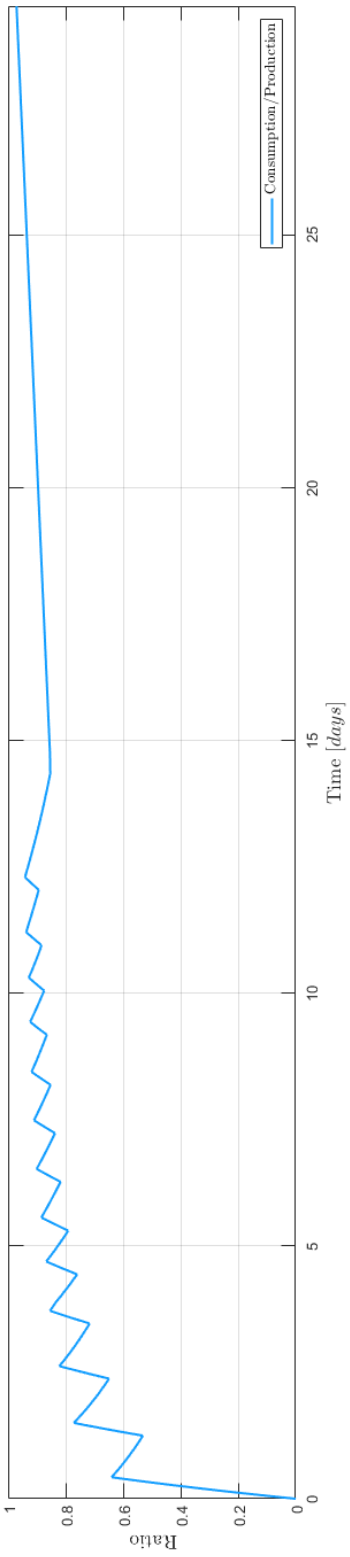
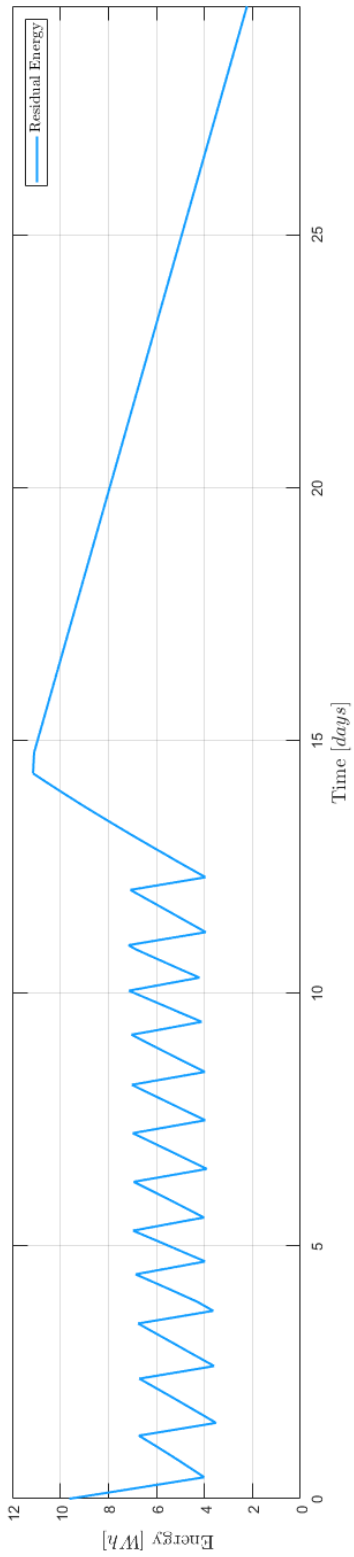


Figure A.10: Energy production and consumption trends during the synodic month; the UWB module is turns on at 60% charge



(a) Ratio of energy consumption to energy production during the synodic month



(b) Residual energy in the batteries during the synodic month

Figure A.11: Energy-related output data; the UWB module is turns on at 60% charge

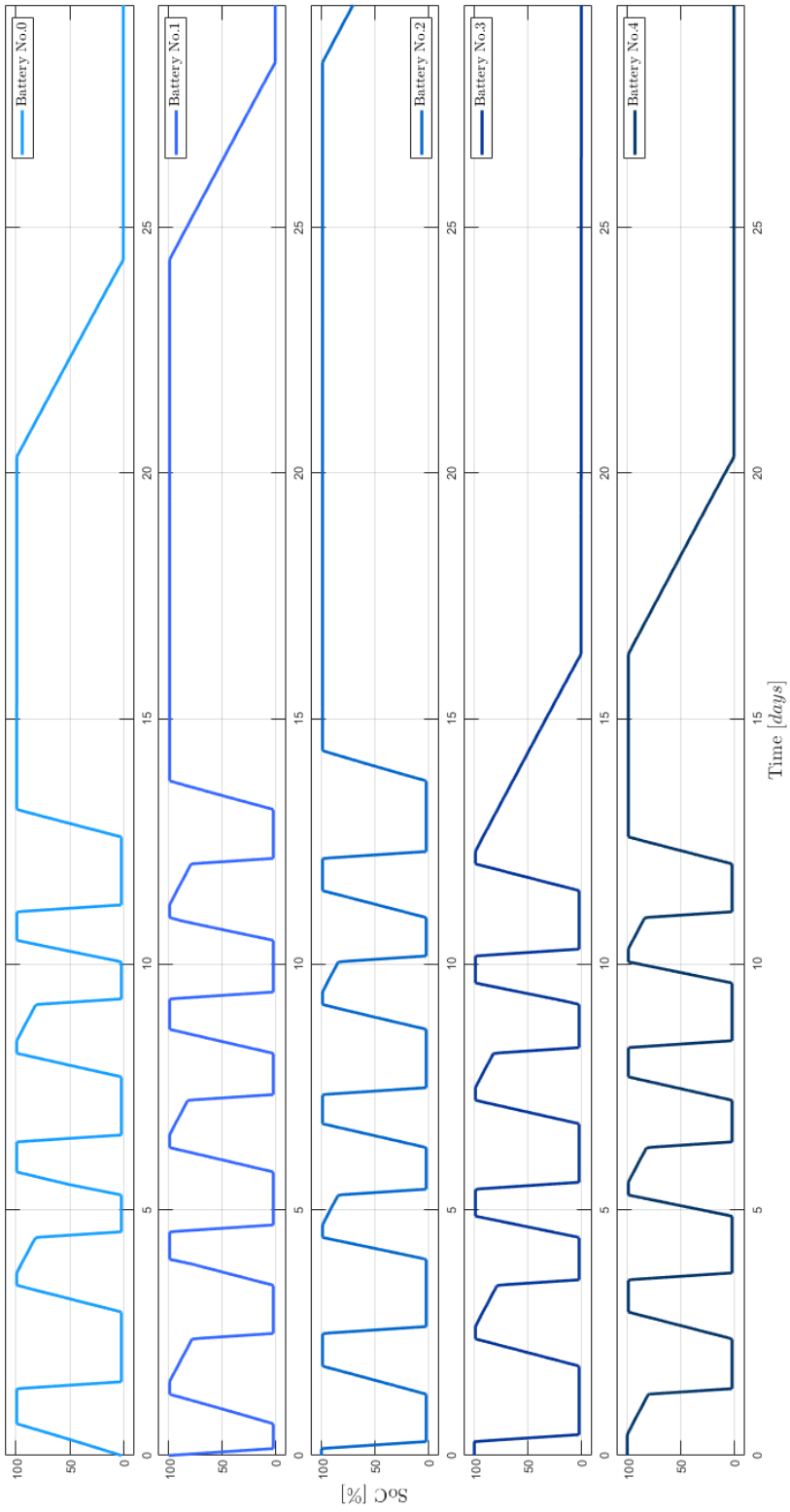


Figure A.12: SoC trends of the five batteries during the synodic month; the UWB module is turns on at 60% charge

Appendix B

Motherboard Code

This appendix shows the code loaded on the Motherboard and the header files used to implement the timer and sum functions needed.

In Section 4.6 the workings of the functions called in *loop*() are explained in detail. Instead, here, the initial set of variables used for completeness is also given in full.

In addition, there are *Serial.print*() in the various functions, which are used for debugging by plotting on the screen the progress of the various operations performed in response to the simulated inputs.

B.1 Summ function

```
#ifndef sum_h
#define sum_h

class SUM{
  public:
    int sum(int,int*);
};

int SUM::sum(int quantity, int vector[]){
  int result = 0;
  for(int x=0;x<quantity;x++){
    result += vector[x];
  }
  return result;
}
#endif
```

Listing B.1: Summ function

B.2 Timer function

```
#ifndef mytimer_h
#define mytimer_h
/*This timer works in minutes*/

class MyTimer{
  private:
    int delay_required;
    unsigned t1, dt;
    int active = 0;
  public:
    MyTimer();
    void set(int);
    int check();
};

MyTimer::MyTimer(){
  active = 0;
}

void MyTimer::set(int n){
  delay_required = n;
  t1 = millis()/60000;
  active = 1;
}

int MyTimer::check(){
  int ret = 0;
  if(active == 1){
    dt = millis()/60000 - t1;
    if(dt > delay_required){
      ret = 1;
      active = 0;
    }
  }
  else{ret = 0;}
  return ret;
}
#endif
```

Listing B.2: Timer function

B.3 Matherboard Code

```
#include <Servo.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "MyTimer.h"
#include "SUM.h"


// INPUT pin value
  // Signal from threshold comparator of battery [i]
  const int I_threshold_comparator[5] = {22, 23, 24, 25, 26};
  boolean threshold_comp_value[5];

  // Signal from limit switch (HIGH if the device is completely closed)
  const int I_limit_switch_signal = 27;

  // Signal from UWB to start operating if all the other devices are ready
  const int I_all_device_ready = 28;

  // Signal from UWB to open and close the device
  const int I_open_device = 29;
  const int I_close_device = 30;
  const int I_battery_fault = 31;

// OUT pin value
  // Servo initialization
  Servo servo_motor;

  // Pin for switches to charge battery [i]
  const int O_battery_charging_switch[5] = {40, 41, 42, 43, 44};
  // Pin for switches to operate with battery [i]
  const int O_battery_operating_switch[5] = {45, 46, 47, 48, 49};

  // N-MOS for servo motor, limit switch, UWB
  const int O_switch_servo = 50;
  const int O_switch_limitswitch = 51;
  const int O_switch_UWB = 52;
  const int O_device_open = 53;

// VARIABLE set
  // Battery switches initial state
  int battery_switch_state[5] = {2, 1, 1, 1, 1};
  SUM sum1 = SUM();

  // Servo motor movement definition
  const int O_servo_signal = 13; // PWM pin
  const int servo_closing = 80;
  const int servo_stop = 90;
  const int servo_opening = 100;

  // UWB activation variable
  int first_activation = 1;
  int battery_full_charge = 0;
  int other_not_ready_flag = 0;
  int other_not_ready_count = 0;
  int time_start;
  int time_end;
  int n_battery_to_work = 4;
  MyTimer timer_attempt = MyTimer(); //min
  MyTimer timer_attempt_delay = MyTimer(); //min
  int UWB_attempt_delay = 1;         //min
  int UWB_activation_delay1 = 60;    //min
```

```
63   int UWB_activation_delay2 = 60*24; //min
64
65   // Battery management variable
66   int battery_fault_flag[5] = {0, 0, 0, 0, 0};
67   int battery_ready[6] = {99, 99, 99, 99, 99, 99};
68
69
70 void setup() {
71     Serial.begin(9600); // open the serial port at 9600 bps:
72   // INPUT pin setup
73     pinMode(I_threshold_comparator[0], INPUT);
74     pinMode(I_threshold_comparator[1], INPUT);
75     pinMode(I_threshold_comparator[2], INPUT);
76     pinMode(I_threshold_comparator[3], INPUT);
77     pinMode(I_threshold_comparator[4], INPUT);
78     pinMode(I_limit_switch_signal, INPUT);
79     pinMode(I_all_device_ready, INPUT);
80     pinMode(I_open_device, INPUT);
81     pinMode(I_close_device, INPUT);
82     pinMode(I_battery_fault, INPUT);
83
84   // OUT pin setup
85     pinMode(O_battery_charging_switch[0], OUTPUT);
86     pinMode(O_battery_charging_switch[1], OUTPUT);
87     pinMode(O_battery_charging_switch[2], OUTPUT);
88     pinMode(O_battery_charging_switch[3], OUTPUT);
89     pinMode(O_battery_charging_switch[4], OUTPUT);
90
91     pinMode(O_battery_operating_switch[0], OUTPUT);
92     pinMode(O_battery_operating_switch[1], OUTPUT);
93     pinMode(O_battery_operating_switch[2], OUTPUT);
94     pinMode(O_battery_operating_switch[3], OUTPUT);
95     pinMode(O_battery_operating_switch[4], OUTPUT);
96
97     pinMode(O_switch_servo, OUTPUT);
98     pinMode(O_switch_UWB, OUTPUT);
99     pinMode(O_switch_limitswitch, OUTPUT);
100    pinMode(O_device_open, OUTPUT);
101
102  // Servo attachment and setup
103    pinMode(O_device_open, OUTPUT);
104    servo_motor.attach(O_servo_signal);
105    servo_motor.write(servo_stop);
106
107  // Battery setup
108    digitalWrite(O_battery_operating_switch[0], HIGH);
109
110 // Send to UWB that the device is closed and ready to operate
111 digitalWrite(O_device_open, LOW);
112
113   //Initialize queue of battery charged
114   delay(1000);
115     for(int i=1;i<5;i++){
116       if(digitalRead(I_threshold_comparator[i])==HIGH){
117         for(int j=0;j<5;j++){
118           if(battery_ready[j]==99){
119             battery_ready[j] = i;
120             j=6;
121           }
122         }
123       }
124     }
125 }
126
127
128 void OPEN_DEVICE(){
```

```
129
130   digitalWrite(O_switch_limitswitch, HIGH);
131   delay(1000);
132   if(digitalRead(I_limit_switch_signal) == HIGH){
133     digitalWrite(O_switch_servo, HIGH);
134     delay(1000);
135     Serial.print("\n\rSTART Opening");
136     servo_motor.write(servo_opening);
137     Serial.print("\n\rOPENING...");
138     delay(7000);
139     servo_motor.write(servo_stop);
140     delay(100);
141     // Open switch to avoid energy loss
142     digitalWrite(O_switch_servo, LOW);
143     // Opened device signal activated
144     digitalWrite(O_device_open, HIGH);
145     Serial.print("\n\rEND Opening");
146   }
147   digitalWrite(O_switch_limitswitch, LOW);
148 }
149
150
151 void CLOSE_DEVICE(){
152
153   digitalWrite(O_switch_limitswitch, HIGH);
154   delay(1000);
155   if(digitalRead(I_limit_switch_signal) == LOW){
156     digitalWrite(O_switch_servo, HIGH);
157     delay(1000);
158     Serial.print("\n\rSTART Closure");
159     Serial.print("\n\rClosing...");
160     servo_motor.write(servo_closing);
161     while(digitalRead(I_limit_switch_signal) == LOW){
162       delay(100);
163     }
164     servo_motor.write(servo_stop);
165     Serial.print("\n\rEND Closure");
166     delay(100);
167     // Open switch to avoid energy loss
168     digitalWrite(O_switch_servo, LOW);
169     // Closed device signal activated (not open)
170     digitalWrite(O_device_open, LOW);
171   }
172   digitalWrite(O_switch_limitswitch, LOW);
173 }
174
175
176 void BATTERY_MANAGEMENT(){
177
178       //battery_switch_state[x] == 0 battery in charge: op[x]=LOW  && ch[x]=HIGH
179       //battery_switch_state[x] == 1 battery waiting:   op[x]=LOW  && ch[x]=LOW
180       //battery_switch_state[x] == 2 battery operating: op[x]=HIGH && ch[x]=LOW
181       //battery_fault_flag[x]   == 1 battery fault:     don't operate on this battery
182
183   for (int i=0; i<5; i++){
184     Serial.print("\n\r Battery ");
185     Serial.print(i);
186     // Current battery threshold acquisition
187     threshold_comp_value[i] = digitalRead(I_threshold_comparator[i]);
188     Serial.print("\n\r\tThreshold signal: ");
189     Serial.print(threshold_comp_value[i]);
190     Serial.print("\n\r\tInitial state: ");
191     Serial.print(battery_switch_state[i]);
192     // Go to the next battery if the current is at fault
193     if(battery_fault_flag[i] == 0){
194       // Change battery if the current is connected to the load and has a low charge level
```

```
195     if(threshold_comp_value[i] == LOW && battery_switch_state[i] == 2){
196       digitalWrite(O_battery_operating_switch[battery_ready[0]], HIGH);
197       battery_switch_state[battery_ready[0]] = 2;
198       delay(10);
199       Serial.print("\n\r\tLow charge: switching to Battery ");
200       Serial.print(battery_ready[0]);
201       // Update the queue of charged batteries
202       Serial.print("\n\r\t\tQueue Battery charged: ");
203       Serial.print("[ ");
204       for (int k=0; k<5; k++){
205         battery_ready[k] = battery_ready[k+1];
206       Serial.print(battery_ready[k]);
207       Serial.print(" ");
208       }
209       Serial.print("]");
210       // Current battery [i] in standby (ready to be charged)
211       digitalWrite(O_battery_operating_switch[i], LOW);
212       battery_switch_state[i] = 1;
213       delay(10);
214     }
215     // Current battery on standby if it is fully charged
216     else if (threshold_comp_value[i] == HIGH && battery_switch_state[i] == 0){
217       digitalWrite(O_battery_charging_switch[i], LOW);
218       battery_switch_state[i] = 1;
219       delay(10);
220       // Add the battery charged to the queue of the charged battery vector
221       for (int j=0; j<5; j++){
222         if(battery_ready[j] == 99){
223           battery_ready[j] = i;
224             Serial.print("\n\r\tBattery charged and added at the end of the queue: [ ");
225             delay(10);
226             for (int o=0;o<5;o++){
227             Serial.print(battery_ready[o]);
228             Serial.print(" ");
229             }
230             Serial.print(" ]");
231           j=6;
232         }
233       }
234     }
235     // Recharge the current battery if low charge level and no other is connected to the recharge system
236     else if(threshold_comp_value[i] == LOW && battery_switch_state[i] == 1 && sum1.sum(5,battery_switch_state)>5){
237     Serial.print("\n\r\tConnecting to charge system");
238       digitalWrite(O_battery_charging_switch[i], HIGH);
239       battery_switch_state[i] = 0;
240       delay(10);
241     }
242     // If battery fault activate the flag and disconnect the battery from the recharge system
243     if(battery_switch_state[i] == 0 && battery_switch_state[i] == 0 && digitalRead(I_battery_fault) == HIGH){
244       digitalWrite(O_battery_charging_switch[i], LOW);
245       battery_fault_flag[i] = 1;
246       battery_switch_state[i] = 1;
247       n_battery_to_work -= 1;
248       Serial.print("\n\r\t   Fault founded");
249       Serial.print("\n\r\t   Number of battery needed to turn on UWB module: ");
250       Serial.print(n_battery_to_work);
251     }
252   }
253   // Final state information
254   Serial.print("\n\r\tFault flag: ");
255   Serial.print(battery_fault_flag[i]);
256   Serial.print("\n\r\tFinal state: ");
257   Serial.print(battery_switch_state[i]);
258   Serial.print("\n\r\t  Switch to Charge: ");
259   Serial.print(digitalRead(O_battery_charging_switch[i]));
260   Serial.print("\n\r\t  Switch to Load: ");
```

```
261     Serial.print(digitalRead(O_battery_operating_switch[i]));
262     delay(100);
263   }
264   Serial.print("\n\rFinal Queue Battery ful charge: [ ");
265   for (int o=0;o<5;o++){
266     Serial.print(battery_ready[o]);
267     Serial.print(" ");
268   }
269   Serial.print("]");
270 }
271
272
273 void UWB_MANAGEMENT(){
274
275   // Standby signal to switch on/off UWB board according to SOC batteries
276   battery_full_charge = 0;
277   for(int i=0; i<5; i++){
278     if(threshold_comp_value[i] == HIGH){
279       battery_full_charge += 1;
280     }
281   }
282   Serial.print("\n\r Full charge Battery: ");
283   Serial.print(battery_full_charge);
284   delay(10);
285
286   // When battery are ready -> power on UWB
287   if((battery_full_charge>n_battery_to_work-1 && other_not_ready_flag == 0 && digitalRead(O_switch_UWB) == LOW
          ) || first_activation == 1){
288     digitalWrite(O_switch_UWB, HIGH);
289     delay(10);
290     // Start the timer 1 and increase the counter
291     timer_attempt.set(UWB_attempt_delay);
292     other_not_ready_count += 1;
293   }
294   // Low charge Batteries -> shootdown UWB
295   else if(battery_full_charge<2 && digitalRead(O_switch_UWB) == HIGH) {
296     digitalWrite(O_switch_UWB, LOW);
297     delay(10);
298   }
299
300   //Print the state
301   if(digitalRead(O_switch_UWB) == HIGH){
302     Serial.print("\n\r UWB module ON");
303       if(digitalRead(O_switch_UWB) == HIGH && digitalRead(I_all_device_ready) == LOW){
304       Serial.print("\n\r    Attempt: ");
305       Serial.print(other_not_ready_count);
306       Serial.print("\n\r    Waiting other devices");
307     }
308   }
309   else if(digitalRead(O_switch_UWB) == LOW && other_not_ready_flag == 0){
310     Serial.print("\n\r UWB module OFF: LOW charge system");
311   }
312   else{
313     Serial.print("\n\r UWB module OFF: Waiting for new attempt");
314   }
315
316   // Turn OFF the UWB if the signal from the other device is missing for more than 1 min
317   if(digitalRead(O_switch_UWB) == HIGH && digitalRead(I_all_device_ready) == LOW && timer_attempt.check() == 1){
318       // Start the timer for the standby period
319       timer_attempt_delay.set(UWB_activation_delay1);
320       delay(10);
321       // Increase timer after 5 attempts
322       if(other_not_ready_count == 5){
323         Serial.print("\n\r    Increase Delay");
324         timer_attempt_delay.set(UWB_activation_delay2);
325         delay(10);
```

```
326       }Serial.print("\n\r    Turn-OFF UWB module (Other devices not Working)");
327       digitalWrite(O_switch_UWB, LOW);
328       delay(10);
329       other_not_ready_flag = 1;
330
331     }
332     else if(digitalRead(O_switch_UWB) == HIGH && digitalRead(I_all_device_ready) == HIGH){
333       other_not_ready_count = 0;
334       Serial.print("\n\r Other devices operating");
335     }
336     delay(10);
337
338     // Reset the flag to do another attempt after the delay
339     if(other_not_ready_flag == 1 && timer_attempt_delay.check() == 1){
340       other_not_ready_flag = 0;
341     }
342 }
343
344
345 void loop(){
346     Serial.println("\n\r ------ Start Loop ------");
347
348     // Threshold acquisition from threshold comparator
349     //THRESHOLD_ACQUISITION();
350
351     // Manage batteries
352     Serial.print("\n\rSTART Battery Management");
353     delay(10);
354     BATTERY_MANAGEMENT();
355     Serial.println("\n\rEND Battery Management");
356
357     // Open or close the device if needed
358     if(digitalRead(I_open_device) == HIGH){
359       OPEN_DEVICE();
360     }
361     else if(digitalRead(I_close_device) == HIGH){
362       CLOSE_DEVICE();
363     }
364
365     // Manage the UWB module activation
366     Serial.print("\n\rSTART UWB Management");
367     UWB_MANAGEMENT();
368     Serial.println("\n\rEND UWB Management");
369
370     // Repeat operation every 5 sec
371     Serial.println("\n\r ------ End Loop ------\n\r");
372     delay(5000);
373     first_activation = 0;
374 }
```

Listing B.3: Matherboard Code

Appendix C

Motherboard & Support Board Wiring Diagrams & Components

This appendix contains wiring diagrams of the Boards designed during HasuNoHana prototyping and the lists of all the components on the boards.

The Motherboard is used for the management of battery charging and usage, but also the opening and closing of the device after release on the lunar ground and the powering up of the UWB module.

The Support Boards, on the other hand, are used for regulating the powers produced by the PVs. There are five of these boards in the device, one for each panel, and each of them is directly connected then to the Motherboard.

C.1 Motherboard

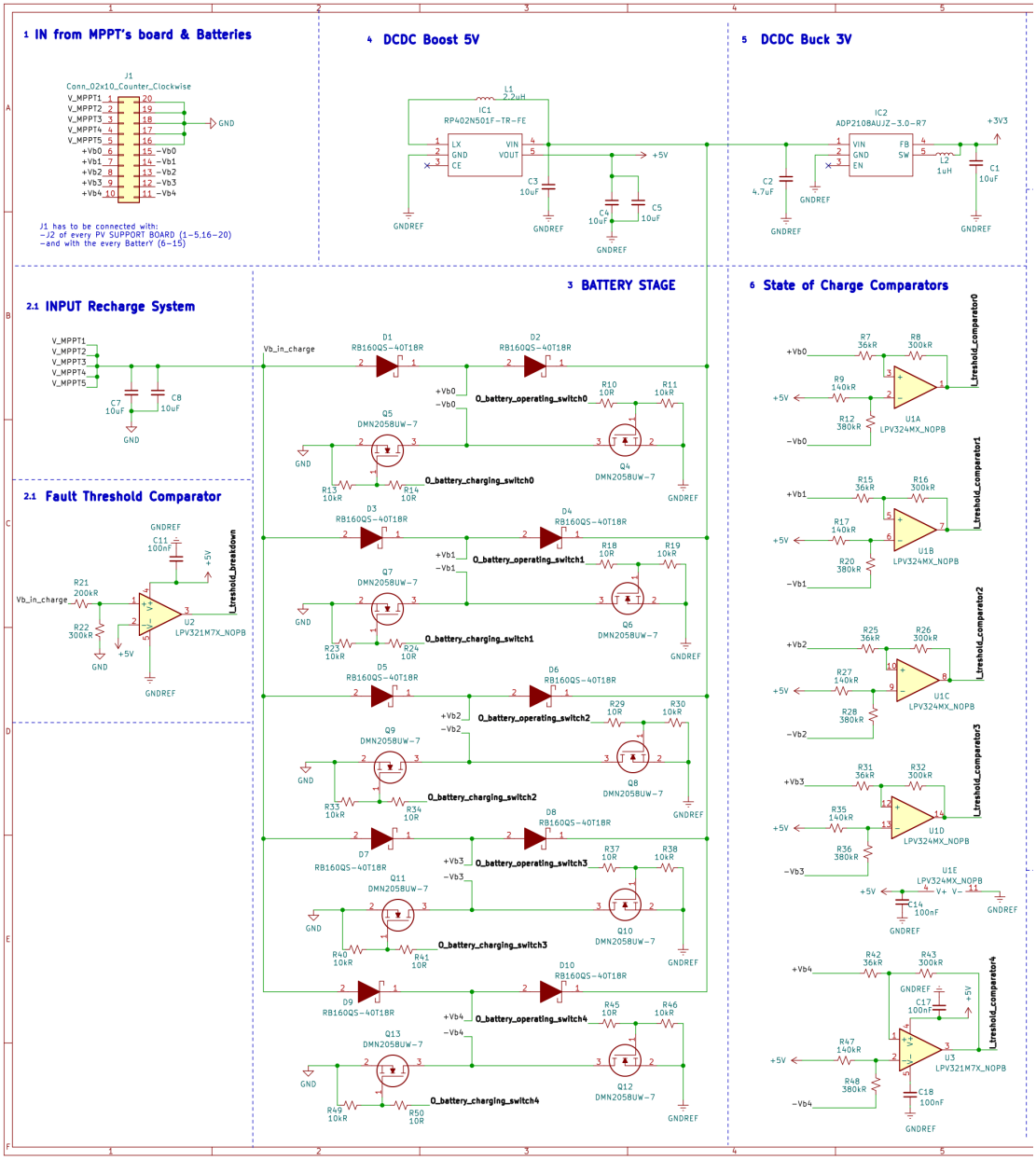


Figure C.1: Motherboard Electrical diagram (left)

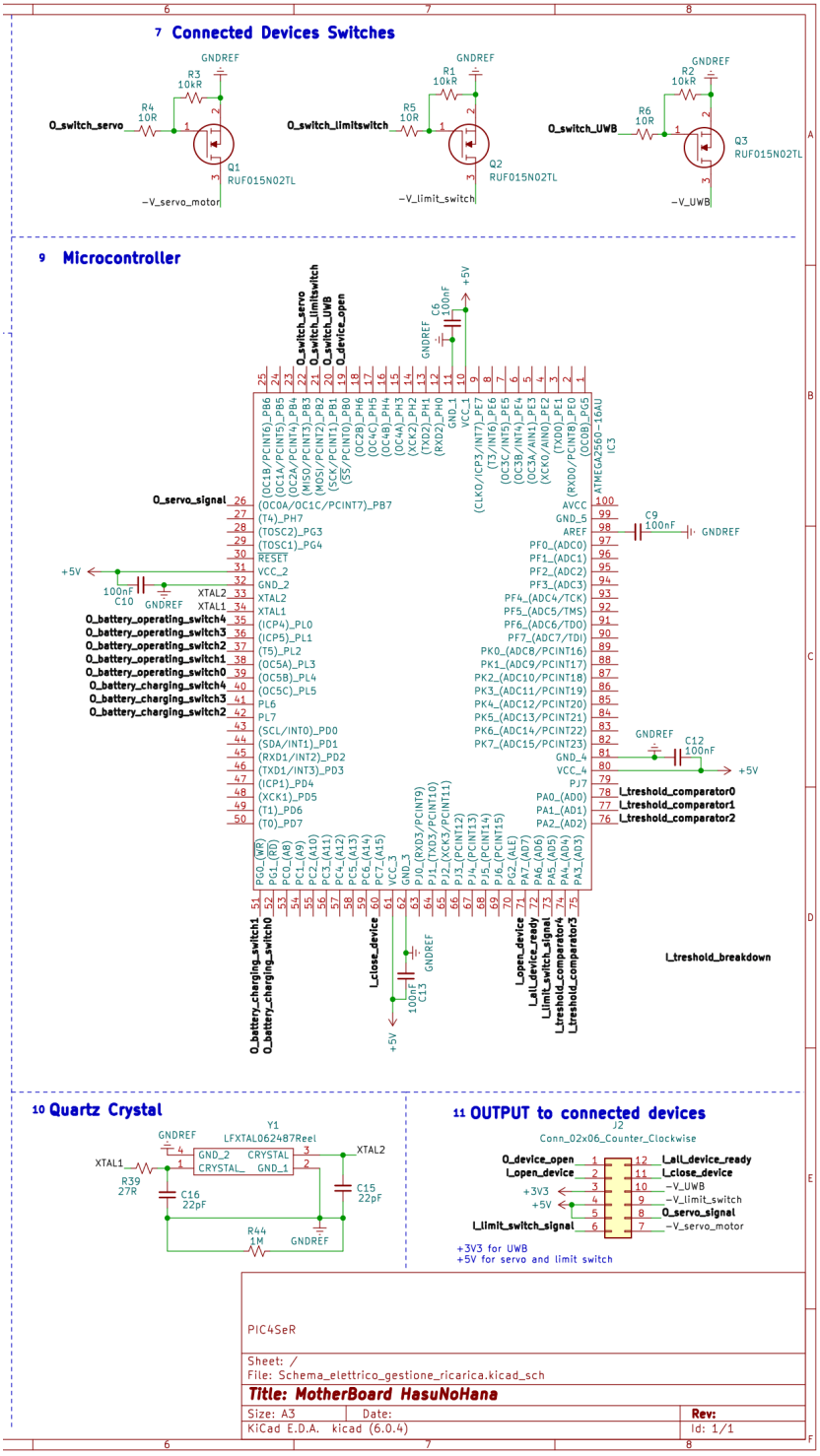


Figure C.2: Motherboard Electrical diagram (right)

Reference	Description	Manufacturing Number	Quantity
C2	Multilayer Ceramic Capacitors MLCC SMD/SMT 4.7UF 6.3V 10% 0603	GCJ188C70J475KE02D	1
C1, C3-C5, C7, C8	Multilayer Ceramic Capacitors MLCC SMD/SMT 10UF 6.3V 10% 0603	GRM188R60J106KE47D	6
C6, C9-C14, C17, C18	Multilayer Ceramic Capacitors MLCC SMD/SMT .1UF 25V 5% 0603	VJ0603Y104JXXAC	9
C15, C16	Multilayer Ceramic Capacitors MLCC SMD/SMT 0603 22pF 250V C0G 1% HI FREQ	VJ0603D220FXPAJ	2
D1-D10	Schottky Diodes & Rectifiers SCHOTTKY BARRIER DIODE	RB160QS-40T18R	10
IC1	Switching Voltage Regulators PWM/VFM Step-up DCDC Converter with Synchronous Rectifier	RP402N501F-TR-FE	1
IC2	Compact, 600 mA, 3 MHz, Step-Down DC-to-DC Converter	ADP2108AUJZ-3.0-R7	1
IC3	8 bit AVR Microcontroller 16MHz 4, 256kb Flash, 8kb RAM, I2C SPI 100-Pin TQFP	ATMEGA2560-16AU	1
L1	Power Inductors SMD 2.2 UH 20%	MCHK1608T2R2MKN	1
L2	Power Inductors SMD 1.0 UH 20%	PE-0603CLH1R0STS	1
Q1-Q3	N-channel MOSFET Transistor, 1.5 A, 20 V, 3-Pin TUMT	RUF015N02TL	3
Q4-Q13	N-Channel MOSFET, 3.7 (State) A, 4.6 (Steady) A, 20 V, 3-Pin SOT-23 Diodes Inc	DMN2058UW-7	10
R21	Thin Film Resistors SMD 200Kohms .5% 25ppm	TNPW0603200KDEEA	1
R8, R16, R22, R26, R32, R43	Thin Film Resistors SMD 300. Kohms 0.1%	TNPW0603300KBEEA	6
R39	Thin Film Resistors SMD .1W 27ohms 1% 0603 50ppm Auto	MCT06030C2709FP500	1
R7, R15, R25, R31, R42	Thin Film Resistors SMD 36K OHM .1% 25PPM 1/6W	PTN0603E3602BST1	5
R44	Thin Film Resistors SMD .125W 1Mohms .1% 0603 25ppm	MCT0603MD1004BP500	1
R4-R6, R10, R14, R18, R24, R29, R34, R37, R41, R45, R50	Thin Film Resistors SMD 10ohms .1% 25ppm	TNPW060310R0BETA	13
R1-R3, R11, R13, R19, R23, R30, R33, R38, R40, R46, R49	Thick Film Resistors 0603 10Kohm 1% Anti Surge AEC-Q200	SDR03EZPF1002	13
R9, R17, R27, R35, R47	Thin Film Resistors SMD 140Kohms .1% 25ppm	TNPW0603140KBEEN	5
R12, R20, R28, R36, R48	Thin Film Resistors SMD 1/10W 379K Ohms 0.5%	RT0603DRE07379KL	5
U1	Batteries 1-4 Comparator	LPV324MX_NOPB	1
U2, U3	Battery 5 Comparator	LPV321M7X_NOPB	2
Y1	Crystals 16MHz 22pF -10C 60C	LFXTAL062487Reel	1

Table C.1: Motherboard electrical components list

C.2 Support Board

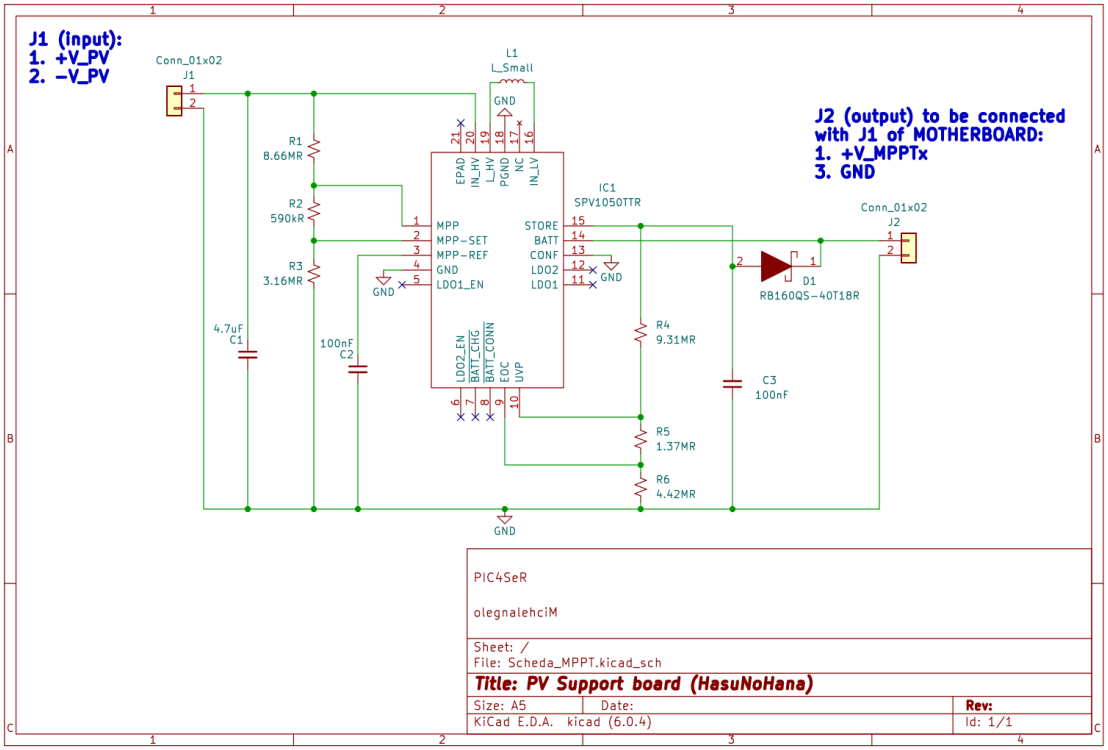


Figure C.3: Support Board Electrical diagram

Reference	Description	Manufacturing Number	Quantity
C1	Multilayer Ceramic Capacitors MLCC SMD/SMT 4.7UF 6.3V 10% 0603	GCJ188C70J475KE02D	1
C2, C3	Multilayer Ceramic Capacitors MLCC SMD/SMT .1UF 25V 5% 0603	VJ0603Y104JXXAC	2
D1	Schottky Diodes & Rectifiers SCHOTTKY BARRIER DIODE	RB160QS-40T18R	1
IC1	STMICROELECTRONICS - SPV1050TTR BATT CHARGER, LITHIUM, 0.07A, VFQFPN-20	SPV1050TTR	1
R1	Thick Film Resistors SMD 1/10watt 8.66Mohms 0.01	CRCW06038M66FKEA	1
R2	Thick Film Resistors 1/10watt 590Kohms 1%	CRCW0603590KFKTA	1
R3	Thick Film Resistors 1/10watt 3.16Mohms 0.01	CRCW06033M16FKEA	1
R4	Thick Film Resistors 1/10watt 9.31Mohms 1%	CRCW06039M31FKEA	1
R5	Thick Film Resistors 1.37M OHM 1%	RK73H1JTTDD1374F	1
R6	Resistori a pellicola spessa SMD 4.42M ohm 1% 0.1W AE	RK73H1JRTTD4424F	1

Table C.2: Support Board electrical components list

Appendix D

Assembly process of HasuNoHana

This Appendix shows the Photos of the assembly process of the HasuNoHana working mechanical prototype.

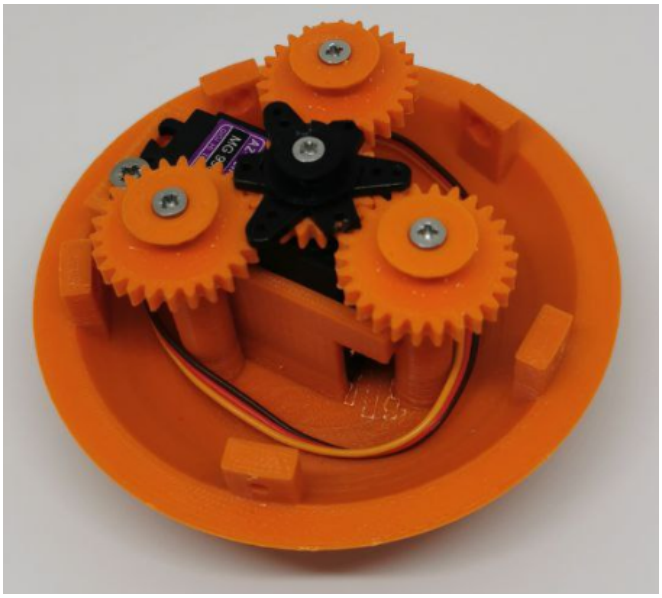


Figure D.1: HasuNoHana base with servo and gears

Figure D.2: HasuNoHana shell

Figure D.3: HasuNoHana shell and folding doors

Figure D.4: HasuNoHana full mechanical prototype

Bibliography

[1] K. Curran, Eoghan Furey, T. Lunney, Jose A. Santos, D. Woods, and A. McCaughey. «An evaluation of indoor location determination technologies». In: *Journal of Location Based Services* (2011) (cit. on p. 2).

[2] M. Chelly and N. Samama. «New techniques for indoor positioning combining deterministic and estimation methods». In: *European Navigation Conference Global Navigation Satellite Systems* (2009) (cit. on p. 2).

[3] F. Lazzari. «Localizzazione di un UAV mediante tecnica TOA in sistemi UWB». MA thesis. Pisa: Università di Pisa, 2015 (cit. on p. 3).

[4] *Principe de fonctionnement de géolocalisation par triangulation*. Info Hogh Tech. URL: `https://info-high-tech.com/geolocalisation-par-triangulation-cest-quoi-exactement/` (cit. on p. 3).

[5] NASA. «Li-Ion Batteries for Space Applications». In: *The 2nd Conference on Advances in Lithium Battery Technologies for Mobile Applications: Lithium lon and Lithium Polymer* (2006). URL: `https://trs.jpl.nasa.gov/bitstream/handle/2014/41721/06-3920A.pdf?sequence=1` (cit. on p. 6).

[6] NASA. «Moon Fact Sheet». In: *the NSSDCA* (). URL: `https://nssdc.gsfc.nasa.gov/planetary/factsheet/moonfact.html` (cit. on pp. 8–10).

[7] NASA. «Design Considerations for Lunar Base Photovoltaic Power Systems». In: *21st Photovoltaic Specialists Conference* (1990). URL: `https://ntrs.nasa.gov/api/citations/19910004946/downloads/19910004946.pdf` (cit. on pp. 8, 37).

[8] M. Kaczmarzyk, M. Gawronski, and G. Piatkowski. «Global database of direct solar radiation at the Moon's surface for lunar engineering purposes». In: *EDP Sciences* (2018). URL: https://www.e3s-conferences.org/articles/e3sconf/pdf/2018/24/e3sconf_solina2018_00053.pdf (cit. on pp. 8, 9, 11).

[9] NASA. «Sun Fact Sheet». In: *the NSSDCA* (). URL: https://nssdc.gsfc.nasa.gov/planetary/factsheet/sunfact.html (cit. on p. 10).

[10] European Space Agency. «Deployable tensegrity structure». In: *Space Engineering Technology* (). URL: https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Deployable_tensegrity_structure (cit. on p. 29).

[11] R. Skelton. «Tensegrity Approaches to In-Space Construction of a 1g Growable Habitat». In: *NASA* (2016). URL: https://www.nasa.gov/feature/tensegrity-approaches-to-in-space-construction-of-a-1g-growable-habitat/ (cit. on p. 29).

[12] *MG996R data sheet*. AZDelivery. URL: https://pdf1.alldatasheet.com/datasheet-pdf/view/1131873/ETC2/MG996R.html (cit. on p. 30).

[13] Neil Storey. *Electronics: A Systems Approach*. Gosport, England: Ashford Colour Press Ltd, 2009 (cit. on p. 49).

[14] *ADP2108AUJZ data sheet*. Norwood, Massachusetts: Analog Devices. URL: https://www.mouser.it/datasheet/2/609/ADP2108-1503378.pdf (cit. on p. 50).

[15] *RP402N501F data sheet*. Tokyo, Giappone: Nisshinbo Holdings Inc. URL: https://www.nisshinbo-microdevices.co.jp/en/pdf/datasheet/rp402-ea.pdf (cit. on p. 50).

[16] V. Pop, H. J. Bergveld, P. H. L. Notten, and P. P. L. Regtien. «State-of-the-art of battery state-of-charge determination». In: *2005 IOP Publishing Ltd* (2005). URL: ftp://ftp.isi.edu/end2end/end2end-interest-1990.mail (cit. on p. 51).

[17] *LPV321 data sheet*. Dallas, Texas: Texas Instruments. URL: https://www.ti.com/lit/ds/symlink/lpv324-n.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1658331028752&ref_url=https%253A%252F%252Fwww.mouser.it%252F (cit. on p. 56).

[18] *PC365NJ0000F data sheet*. Osaka, Japan: Sharp. URL: https://global.sharp/products/device/lineup/data/pdf/datasheet/pc365n_e.pdf (cit. on p. 58).

[19] *DMN2058UW data sheet*. Plano, Texas: Diodes Incorporateds. URL: https://www.diodes.com/assets/Datasheets/DMN2058UW.pdf (cit. on p. 61).

[20] *ATMEGA2560 data sheet*. Chandler, Arizona: Microchip Technology / Atmel. URL: https://datasheet.datasheetarchive.com/originals/distributors/Datasheets_SAMA/4a5a397fa76c39b5297f4121ace1d84e.pdf (cit. on p. 65).

[21] Phillip A. Laplante. *Comprehensive Dictionary of Electrical Engineering*. Springer, 1999, p. 152 (cit. on p. 66).

[22] *LFXTAL062487Reel data sheet*. IQD Frequency Products. URL: https://www.mouser.it/datasheet/2/741/LFXTAL062487Reel-1000954.pdf (cit. on p. 66).

[23] N. Ventani and M. Montagni. *Tecniche d'inseguimento del Maximum Power Point Algoritmi di Ricerca*. Tech. rep. Firenze: Università degli studi di Firenze, 2013 (cit. on p. 79).

[24] *SPV1050TTR data sheet*. Ginevra, Switzerland: STMicroelectronics. URL: https://www.st.com/resource/en/datasheet/spv1050.pdf (cit. on p. 80).

[25] *KINOVA Gen3 lite data sheet*. Boisbriand, Quebec, Canada: Kinova. URL: https://artifactory.kinovaapps.com/ui/native/generic-documentation-public/Documentation/Gen3%5C%20lite/Technical%5C%20documentation/User%5C%20Guide/Gen3_lite_USER_GUIDE_R03.pdf (cit. on pp. 82, 83).

[26] *RealSense Depth Camera D435 data sheet*. Santa Clara, California: Intel Corporation. URL: https://www.intelrealsense.com/wp-content/uploads/2022/05/Intel-RealSense-D400-Series-Datasheet-April-2022.pdf (cit. on p. 84).

[27] J. Forkel, B. Forkel Wuensche, and H.j. Wuensche. «Determining and Improving the Localization Accuracy of AprilTag Detection». In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020. URL: https://ieeexplore.ieee.org/abstract/document/9197427/authors#authors (cit. on p. 87).