

Automatisme de control d'il·luminació solar directa mitjançant lames regulables

Autor: Alacreu Samper, Marc

Tutor: Gil Benso, Enrique

2017-2018

E.T.S.A. València

INDEX

0. Resum i paraules clau	4
1.Introducció	5
2.Motivació.....	7
3.Objectius	10
4.Antecedents.....	11
5.Metodologia i assaig.....	13
5.1 Marc teòric	13
5.1.1 Posicionament temporal	13
5.1.2 Posicionament solar	13
5.1.3 Canvi de base del vector posició des del sistema de coordenades global al sistema local.	19
5.1.4 Projectió d'ombra entre les lames.	29
5.2 Sistema de programari i maquinari	35
5.2.1 Maquinari	35
5.2.1.1 Maquinari Perifèric	36
5.2.1.2 Circuits	40
5.2.1.3 Prototip de sistema de lames	42
5.2.2 Programari	43
6. Conclusions.....	49
7. Bibliografia.....	50
8. Annex	51

0. Resum i paraules clau

Títol: Automatismes de control d'il·luminació solar directa mitjançant lames regulables.

Resum: Regulació automatitzada de la incidència solar directa en espais habitables mitjançant sistemes de protecció de lames regulables administrats amb automatismes basats en sistema arduino, tenint en compte la posició solar depenent del posicionament geogràfic i temporal del sistema, així com la orientació i inclinació d'aquest, per mantindre estable la incidència solar depenent de les preferències de l'usuari.

CASTELLÀ:

Título: Automatismos de control de iluminación solar directa mediante lamas regulables.

Resumen: Regulación automatizada de la incidencia solar directa en espacios habitables mediante sistemas de protección de lamas regulables administrados con automatismos basados en el sistema arduino, teniendo en cuenta la posición solar dependiente del posicionamiento geográfico y temporal del sistema, así como la orientación e inclinación de este, para mantener estable la incidencia solar dependiente de las preferencias del usuario.

ANGLÉS:

Title: Automatism for direct solar illumination control with adjustable slats

Summary: Automatic regulation of direct solar incidence in living spaces with systems of adjustable slats protection administered with automatismes based in Arduino system, considering solar position, geographic position and time position of the system, as well as the orientation and inclination of this, to keep stable the solar incidence considering the user preferences.

Paraules clau:

"regulation sun incidence"; "sun control", "domotic sun control".

1.Introducció

Amb el següent treball es vol ajudar, en la mesura de les possibilitats a resoldre un dels principals problemes que té actualment la humanitat: garantir la sostenibilitat del planeta Terra, reduint al mínim la petjada del nostre pas per aquesta.

No és el primer automatisme per a lames que es realitza, però no és una solució molt estesa, pot ser, per l'alt valor econòmic d'aquests sistemes.

Amb el sistema proposat en aquest treball, aquest problema es minimitza, pel baix cost que suposarà la seua implementació a qualsevol sistema d'aquestes característiques, i amb més motiu si es compara aquest cost amb la capacitat d'estalvi energètic que pot suposar la seua utilització.

La diferència de la despesa energètica, després d'implementar un sistema laminar convencional a qualsevol edifici, està estimat que pot arribar fins al 40% d'estalvi. Amb el sistema estudiat, aquesta eficiència pot augmentar considerablement, per la optimització continua de la seua disposició dependent de la posició calculada del Sòl en qualsevol moment.

El sistema té la capacitat d'adaptació, amb molt poques modificacions del programari, a qualsevol sistema convencional de lames, en qualsevol posició del planeta, i en qualsevol disposició d'aquest, amb una senzilla configuració, a l'abast de qualsevol usuari sense cap preparació tècnica una vegada instal·lat el dispositiu.

Açò és possible gràcies a la interfície de configuració dissenyada, que fa possible el funcionament del dispositiu una vegada instal·lat de forma autònoma, amb diferents programes de funcionament per adaptar-se a les necessitats canviant que es puguin donar per part de l'usuari, mitjançant la seua configuració senzilla i intuïtiva.

La funció general del dispositiu, és el càlcul de la posició angular solar en qualsevol lloc del planeta i en qualsevol moment, diferenciant les diferents situacions que es poden donar al llarg del dia (dia, nit, o que no li incidisca el Sòl directament per la seua disposició) i actuar depenent d'aquestes dades, accionant les lames per respondre eficaçment als paràmetres de configuració i personalització elegits per l'usuari. Entre altres configuracions, existeix la possibilitat de mantindre el percentatge de superfície solejada que incidix a través del sistema al llarg del temps, mantenint-se constant mentre incidisca el Sòl sobre el sistema, o bé impedir la radiació solar directa en qualsevol moment, mantenint la superfície radiada sobre les lames de protecció constant si el sistema ho permet, i en cas contrari, la menor més pròxima a la programada per l'usuari.

2.Motivació

És més que evident l'impacte que està produint actualment l'activitat humana sobre el medi ambient, degradant-lo de manera progressiva i alarmant, i la necessitat de posar tots els mitjans necessaris al nostre abast per reduir aquest impacte al mínim possible, per poder aconseguir la sostenibilitat del Medi com l'hem conegut fins ara.

La sostenibilitat bé influenciada per molts factors, sent un dels més importants el consum energètic, on la despesa energètica a les edificacions representa el 48% del consum total energètic: un 40% per al seu funcionament (il·luminació, calfament, refrigeració, etc.) i un 8% per la seua construcció. (Norbert Lechner, *Heating, cooling, lighting. Sustainable methods for architects*, pàg. 2).

Aquesta energia ve derivada en gran part del consum de combustibles fòssils, que produeix una gran quantitat de diòxid de carboni, principal causant de l'efecte hivernacle, i per tant del canvi climàtic, a més de ser un recurs esgotable a llarg termini.

Per tant, la reducció de les emissions de diòxid de carboni, un dels principals objectius establerts al Protocol Internacional de Kioto (1997), està directament relacionat amb la reducció del consum de combustibles fòssils. Aquest consum es pot reduir de dos maneres: substituint el seu ús pel d'energies renovables, i amb l'ús de l'energia d'una manera més eficient.

Així, la millora de l'eficiència energètica a les edificacions influeix directament a la millora de la sostenibilitat del medi ambient.

Amb l'objectiu de millorar aquesta eficiència a les edificacions, s'han creat una sèrie de mesures legislatives d'obligat compliment: a nivell estatal, els requisits mínims d'eficiència energètica als edificis els trobem al Codi Tècnic de l'Edificació (CTE), principalment al Document Bàsic d'Estalvi d'Energia (DB-HE). Per una altre costat, i seguint la Directiva Europea d'Eficiència Energètica 2010-31-UE, també cal tindre en compte el Real Decret 235/2013 sobre els Certificats d'Eficiència Energètica.

Per aquestes raons, la millora de l'eficiència energètica als edificis és un valor a tindre cada vegada més en compte al realitzar el seu disseny.

Una de les mesures que es poden emprar per aconseguir aquests objectius, és la correcta regulació de la influència de la radiació solar sobre les edificacions segons les necessitats de condicionament (escalfament o refredament).

Amb el dispositiu de lames regulables automatitzat que desenvolupem en aquest treball, podem augmentar considerablement l'eficiència d'aquests tipus de dispositius d'ombreat respecte dels sistemes fixes tradicionals o dels regulables manualment, que necessiten una manipulació constant per part de l'usuari per aconseguir la màxima eficiència, qüestió que es redueix considerablement amb aquest sistema automatitzat.

De la mateixa manera, aquest dispositiu també pot millorar sensiblement el confort lumínic dels espais segons les necessitats, evitant enlluernaments indesitjats, o la incidència de radiació solar directa en elements deteriorables, com és la situació que es dona a molts museus i espais d'exposició, podent adaptar la inclinació de les lames segons la posició del Sòl al llarg del temps, per mantindre estables les condicions de radiació desitjades.

3.Objectius

Amb el següent estudi volem assolir els següent objectius:

- Creació del programari per al càlcul del posicionament angular del Sòl de forma continua al llarg del temps i en qualsevol posició geogràfica respecte el plànol horitzontal del terra i respecte al Sud geogràfic (azimut i elevació).
- Creació del programari per al càlcul de la incidència solar angular del Sòl respecte a qualsevol disposició possible del sistema de lames (orientació i inclinació).
- Creació del programari per al càlcul de la posició angular de les lames adient, depenent de la posició angular del Sòl i de la quantitat directa de radiació solar que es vulga transmetre a través del sistema, amb la possibilitat de personalitzar aquest per l'usuari depenent de les necessitats d'ús.
- Detecció per part del programari de les diferents fases solars al llarg del dia (dia i nit), i de la incidència o no de la radiació directa sobre el sistema de lames, per modificar la seua reacció depenent de l'estat.
- Configuració del maquinari per a l'accionament previst de les lames per part del programari, i que faça possible la comunicació de les personalitzacions i configuracions adients entre l'usuari i el sistema, sense utilitzar cap maquinari addicional.

4. Antecedents

Al llarg de la història de l'arquitectura podem trobar nombrosos exemples de sistemes de protecció solars dissenyats davant la necessitat de condicionament dels diferents espais habitables.

A l'arquitectura de la Grècia clàssica, podem trobar la utilització de pòrtics i columnates als temples per crear espais ombrejats segons les necessitats, controlant així el solejament dels edificis adientment.

Aquests instruments de protecció van ser heretats per l'arquitectura romana, on podem trobar dispositius fixes, com pòrtics i galeries, a més d'altres elements mòbils a partir de sistemes lonaris atirantats. Un exemple el podem trobar al velari del Coliseu, on es protegia solarment la zona del públic amb aquest sistema.

Per altre costat, al llarg del món, apareixen altres sistemes semblants a diferents arquitectures vernacles: elements similars per a solucionar els mateixos condicionants. Com a exemple podem trobar a l'arquitectura tradicional xinesa i japonesa, grans alers de protecció i circulacions porticades.

A l'edat mitjana, també podem trobar construccions de protecció solar a l'arquitectura àrab mitjançant gelosies per controlar l'entrada de la llum.

Al renaixement, amb la recuperació de models i formes de construcció de l'arquitectura clàssica, podem trobar la utilització de porxos i galeries porticades, a més d'elements mòbils de cortinatges per canviar les proteccions segons les necessitats. Un exemple el podem trobar a la Procuratie Vecchie a la Plaça de Sant Marc a Venècia.

A principis del segle XX, Els germans Greene importen el concepte de protecció de l'arquitectura japonesa mitjançant grans alers a l'arquitectura Californiana (The Gamble House, Pasadena 1908). Al mateix temps, també existeixen altres exemples de la utilització d'instruments d'ombregat a l'arquitectura de Wright (The Robie House, Chicago o The Price Tower).

A la dècada dels anys 30 del segle passat comença ser utilitzat per Le Corbusier l'element més innovador i amb més influència sobre l'arquitectura moderna al respecte: la protecció solar estructural fixa o brise soleil, al que se li atribueix la invenció de la mateixa.

Aquest element sorgeix després de diferents assajos i errors de control per part de l'arquitecte per solucionar els problemes de solejament dels seus projectes, ja utilitzant aquest sistema al projecte de gratacels a La Marine (Argel 1938), la reconstrucció de la Manufactura Duval (Saint-Die 1945 i la Unité d'habitation de Marsella (1946).

Després dissenyarà la ciutat de Chandigarh, a l'Índia, on construirà la Torre de les Ombres, estudi a escala real de la influència del sistema dependent de diferents orientacions i la seua influència sobre les façanes.

Aquest element d'ombrejament va ser estés pel mateix Le Corbusier a diferents llocs del món, com a Brasil, on va assessorar la construcció del Ministeri d'Educació i MES, a Rio de Janeiro (Costa, Reidy, Niemeyer, 1938-46), edifici que es va convertir en emblemàtic dins de l'arquitectura moderna a Sud-Amèrica.

Al llarg del segle XX, amb l'avanç tecnològic respecte materials i enginyeria, es perfecciona aquest sistema amb nous materials i estudis del comportament dels diferents models, encara que mantenint l'essència del brise soleil originari, introduint els sistemes mitjançant lames mòbils, més eficients per la possibilitat d'adequar l'entrada del Sòl a les diferents necessitats al llarg de l'any i el dia.

A partir dels anys 80, amb la revolució informàtica, es milloren els sistemes de modelització per a l'estudi de l'optimització d'aquest sistema de protecció, així com es dona la possibilitat del control mitjançant automatismes de la direcció de les lames per aconseguir la màxima eficiència energètica.

Hui en dia, existeixen infinitats de productors de sistemes de protecció solar mitjançant lames a nivell industrial, i són utilitzats en infinitat de projectes, tant per la millora de l'eficiència energètica que aporten, com per la facilitat de ventilació en detriment d'altres sistemes opacs, podent arribar a produir un estalvi del 40% de l'energia consumida a l'edifici.

Els sistemes d'ombreat mitjançant lames tenen l'avantatge, davant dels sistemes de làmines opaques, de facilitar la ventilació aportant la mateixa ombra, facilitant un millor condicionament tèrmic dels espais.

5. Metodologia i assaig

5.1 Marc teòric

El programari proposat té la capacitat de calcular contínuament la posició angular solar per adaptar l'angle de les lames idòniament a les demandes de les diferents configuracions possibles per part de l'usuari.

Aquest procediment està basat en els següents algorismes de càlcul.

5.1.1 Posicionament temporal

Per als càlculs posteriors utilitzarem l'hora UTC (Temps Universal Coordinat). Aquesta coincideix amb l'hora al meridià de Greenwich. A València, com a exemple, aquesta hora s'obté restant dues hores a l'hora local a l'horari d'estiu, i una hora a l'horari d'hivern.

5.1.2 Posicionament solar

Per al càlcul de l'azimut i l'elevació del Sòl, ens caldrà conèixer la longitud i latitud de posicionament de l'aparell, així com el seu posicionament al llarg del temps. Per a realitzar aquest càlcul, utilitzarem els algorismes utilitzats per Jean Meus al llibre "Astronomical Algorithms".

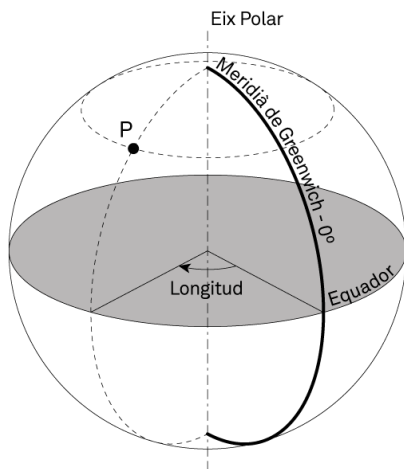


Figura 1: Longitud de posició a la Terra

Longitud:

Posicionament angular de l'artefacte sobre la superfície de la Terra respecte a la línia Nord-Sud que passa pel meridià de Greenwich. Els punts poden estar a l'est o a l'oest d'aquest. La mesura està compresa entre 0° i 180° si l'artefacte està a l'Est d'aquest meridià, i de 0° a -180° si està situat a l'Oest d'aquest. Els decimals seran en notació decimal.

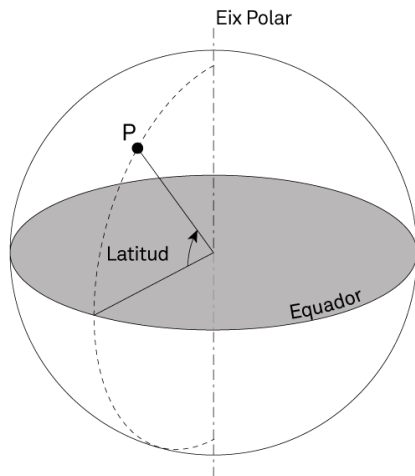


Figura 2: Longitud de posició a la Terra

Latitud:

Posicionament angular de l'artefacte sobre la superfície de la Terra i l'equador. La seua mesura està compresa entre 0° i 90° si està situat al Nord de l'equador, corresponent aquestes dos mesures a l'equador i al Pol Nord respectivament, i entre 0° i -90° si està situat al sud del mateix, corresponent aquesta última longitud límit al Pol Sud.

Dia Julià:

El dia Julià és una mesura de temps que es mesura com als dies solars passats comptats a partir de l'1 de gener de 4713 aC a les 12:00 del migdia. La fracció de dia es comptabilitza en notació decimal.

L'algoritme per calcular el dia Julià és el següent:

Si mes ≤ 2, aleshores y' = any - 1; m' = mes + 12

Si mes > 2, aleshores y' = any; m' = mes

$$A = \text{sencer}(y'/100)$$

$$B = 2 + A + (\text{sencer}(A/4))$$

$$\text{Dia Julià} = [\text{sencer}(365.25(y' + 4716))] + [\text{sencer}(30.6001 * (m' + 1))] + \text{dia} + B - 1524 \\ + \frac{\text{hora} + \frac{\text{minuts}}{60} + \frac{\text{segons}}{3600}}{24}$$

Centúria juliana a partir de gener del 2000 a les 12:00 del migdia (T):

Per al càlcul dels algorismes necessaris, utilitzarem la centúria T (36525 dies) transcorreguda entre la posició de temps en que volem mesurar l'azimut i l'elevació i l'1 de gener de l'any 2000 a les 12:00 del migdia:

Si el Dia Julià del 1 de gener de 2000 a les 12:00 = 2451545

$$T = [(\text{Dia Julià}) - 2451545]/36525$$

Longitud solar:

Angle de posicionament de la terra respecte al Sòl a l'el·líptica que recorre al voltant d'aquest, establint coma 0 graus l'equinocci nord de primavera, 90° el solstici nord d'estiu, 180^a l'equinocci nord de tardor i 270^a el solstici nord d'hivern.

Aquesta la calcularem amb el següent algoritme: L_0

$$L_0 (deg) = 280.46645 + 36000.76983T + 0.0003032T^2$$

Sent T la centúria Juliana transcorreguda a partir del 1 de gener de l'any 2.000 a les 12:00:00 UTC.

Anomalia mitjana solar:

Distància angular des del periheli de l'òrbita de la terra fins al punt en què aquest es situaria si la velocitat angular fora uniforme.

L'algoritme per al seu càlcul és el següent:

$$M(deg) = 357.5291 + 35999.0503T - 0.0001559T^2 - 0.00000048T^3$$

Sent T la centúria Juliana transcorreguda a partir del 1 de gener de l'any 2.000 a les 12:00:00 UTC.

Excentricitat de l'òrbita de la Terra:

Quantitat en que l'el·lipse de l'òrbita de la terra s'allunya d'una circumferència.

$$e = 0.016708617 - 0.000042037 * T - 0.0000001236 * T^2$$

Sent T la centúria Juliana transcorreguda a partir del 1 de gener de l'any 2.000 a les 12:00:00 UTC.

Equació del centre:

Diferència entre l'anomalia vertadera i l'anomalia mitja de la Terra a la seua òrbita el·líptica.

$$C(deg) = (1.9146 - 0.004847T - 0.000014T^2) * \sin(M) + (0.019993 - 0.000101T) * \sin(2M) + 0.00029 * \sin(3M)$$

Longitud solar vertadera:

$$L_{true} = C + L_0$$

Anomalia solar verdadera:

$$f(deg) = M + C$$

Distància de la Terra al Sòl:

$$R(AU) = \frac{1.000001018 * (1 - e^2)}{(1 + e * \cos(f))}$$

Hora angular de Greenwich:

$$H. ang. Green = 280.46061837 + 360.98564736629 * Dia_Julià - 2451545 + 0.000387933T^2 - \frac{T^3}{38710000}$$

Obliqüitat de l'equador:

$$Ob. eq. (deg) = 23 + \frac{26}{60} + \frac{21.448}{3600} - \frac{46.815}{3600}T - \left(\frac{0.00059}{3600}\right)T^2 + \left(\frac{0.001813}{3600}\right)T^3$$

Ascensió recta:

Distància angular de la Terra al primer punt d'Àries, mesurada en sentit horari.

$$As. Rec. (deg) = atan\left[\frac{\sin(L_{true}) * \cos(Ob. eq.)}{\cos(L_{true})}\right]$$

Declinació:

$$Decl. = asin[\sin(Ob. eq.) * \sin(L_{true})]$$

Angle horari:

Distància angular entre el meridià de la posició de l'aparell i el cercle horari de la Terra.

$$Ang. h. (deg) = H. ang. Gren + longitud - As. Rec.$$

Azimuth solar:

Distància angular sobre la horitzontal respecte al terra de la posició de l'artefacte respecte el Sud, mesurat en el sentit de les agulles del rellotge.

$$Azimut = atan\left[\frac{\sin(Ang. h.)}{\cos(Ang. h.) * \sin(latitud) - \tan(Decl.) * \cos(Latitude)}\right]$$

Elevació del Sòl:

Distància angular de la posició del Sòl respecte el plànol paral·lel a la superfície del terra a la posició de l'artefacte. Aquesta varia entre -90° i 90° , tenint valors positius quan el Sòl es troba sobre el plànol paral·lel a la superfície terrestre, i negatiu quan es troba per baix d'aquest.

$$Elevació = Asin[\sin(latitud) * \sin(Decl.) + \cos(latitud) * \cos(Decl.) * \cos(Ang. h.)]$$

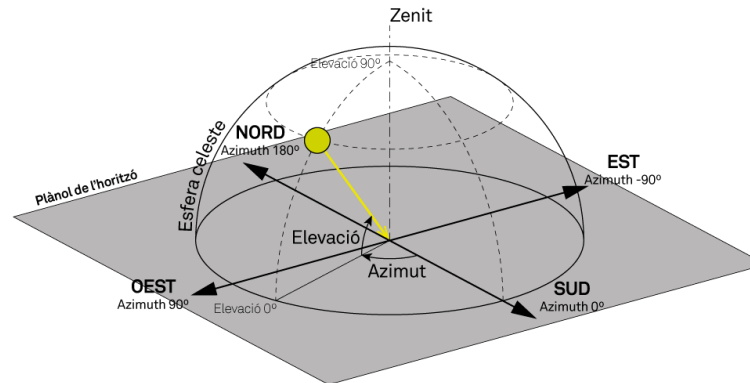


Figura 3: Elevació solar i azimut

5.1.3 Canvi de base del vector posició des del sistema de coordenades global al sistema local.

Amb els algorismes utilitzats al capítol anterior, hem pogut posicionar el Sòl respecte un sistema de coordenades esfèric global situat a qualsevol punt de la Terra. Aquest sistema tindrà com a eixos principals l'eix paral·lel a la superfície de la Terra al punt seleccionat en direcció Sud, i el plànol paral·lel a aquesta.

Però cal tindre en compte que, el que ens interessa realment, és la incidència solar sobre la superfície a protegir, i aquesta canviarà segons l'orientació angular relativa que adopte el nostre sistema de protecció respecte a la base de referència global citada anteriorment.

Canvi de base des del sistema de coordenades global esfèric, al sistema de coordenades global cartesià.

Si representem el Sòl com un punt a l'esfera celeste de radi 1, aquest punt el podem representar en un sistema cartesià, amb els eixos XY al plànol paral·lel a la superfície de la terra, amb X en direcció al Sud i Y en direcció a l'Oest, i Z perpendicular a aquests dos en direcció cap al cel (zenit).

L'origen serà el mateix que el sistema de coordenades anterior. Aquest sistema de coordenades el nomenarem sistema cartesià local.

L'expressió per al canvi de base a aquest nou sistema serà la següent:

$$x' = \cos(\text{elevació}) * \cos(\text{azimut})$$

$$y' = \cos(\text{elevació}) * \sin(\text{azimut})$$

$$z' = \sin(\text{elevació})$$

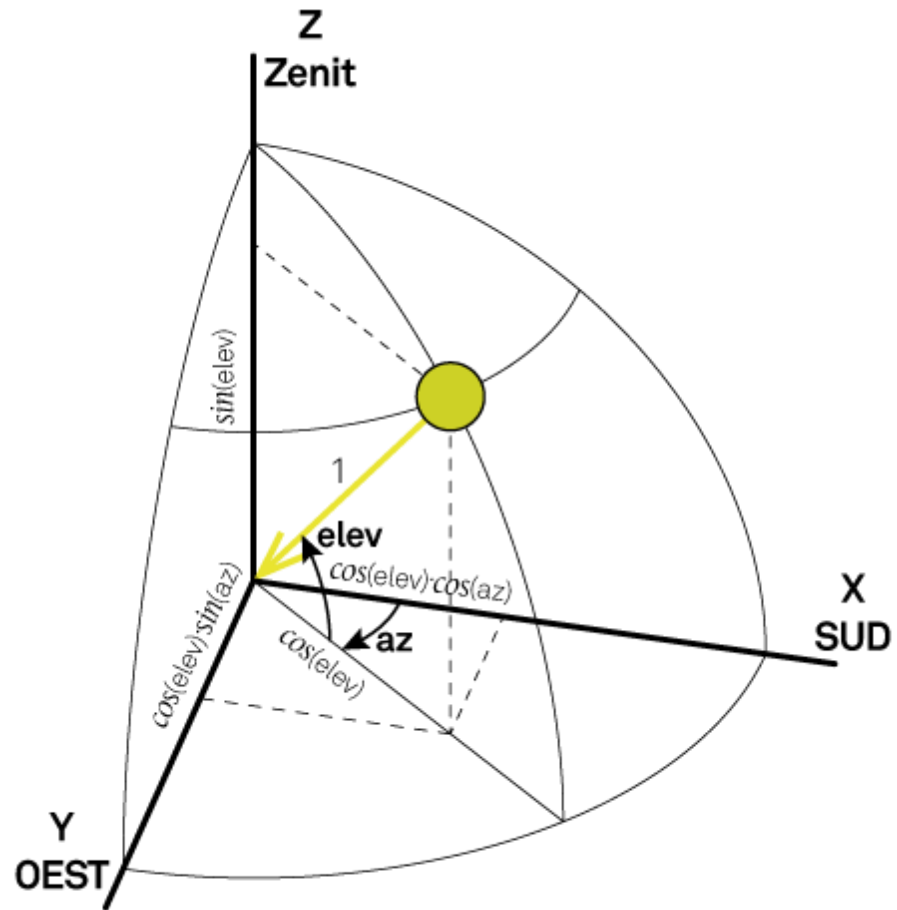


Figura 4: Canvi de coordenades esfèriques a coordenades cartesianes

Sistema de coordenades cartesià local

Definirem com a sistema de coordenades local de l'artefacte aquell sistema cartesià amb els següents eixos:

- Eix X'**: Perpendicular al plànol que conté els eixos de les lames i en direcció a la part dissenyada com a exterior de la protecció.
- Eix Y'**: Paral·lel als eixos de les lames i amb un angle de 90° respecte a l'eix X' en el sentit de les agulles del rellotge vist des de la denominada part superior de l'artefacte.
- Eix Z'**: paral·lel als altres dos i en direcció positiva cap a la denominada part superior de l'artefacte.

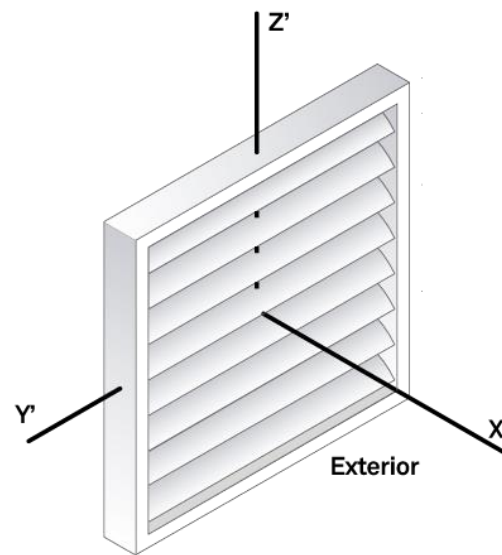


Figura 5: Eixos del sistema de coordenades cartesià local

Canvi de base del sistema de coordenades cartesià global al sistema de coordenades cartesià local

Utilitzarem el mètode d'Euler per especificar l'orientació angular del sistema de coordenades local respecte del global.

Aquest mètode, especifica qualsevol orientació angular que pugui adoptar el sistema local respecte el local mitjançant 3 girs, cadascun respecte als tres eixos del sistema global.

Aquest sistema, és utilitzat comunament en sistemes de navegació per indicar l'actitud dels mòbils, i per analogia utilitzarem les mateixes definicions convencionals.

Heading(φ): gir sobre l'eix Z del sistema de coordenades local, amb sentit positiu horari.

Pitch(θ): gir sobre l'eix Y del sistema de coordenades local, amb sentit positiu horari.

Roll(ϕ): gir sobre l'eix X del sistema de coordenades local, amb sentit positiu horari.

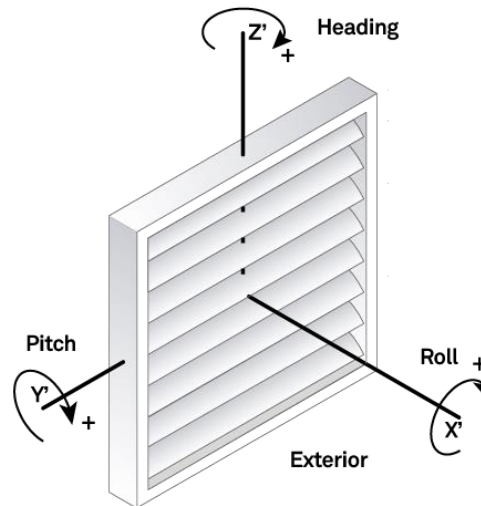


Figura 6: Orientació local angular

Canvi de base amb gir del sistema local sobre l'eix z'(Heading)

Sent $C_{Heading}^{Global}$ la matriu de canvi de base del sistema global al sistema local girat sobre l'eix Z' (Heading):

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = C_{Heading}^{Global} * \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

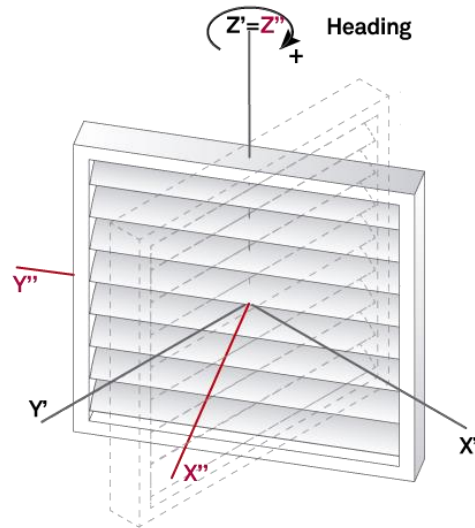


Figura 7: Gir del panell de lames sobre l'eix Z' (Heading)

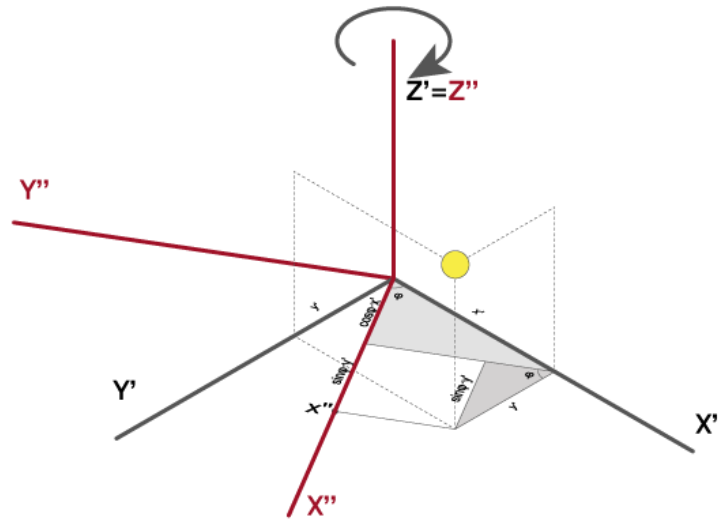


Figura 8: Canvi de base del sistema global al local. Component x''

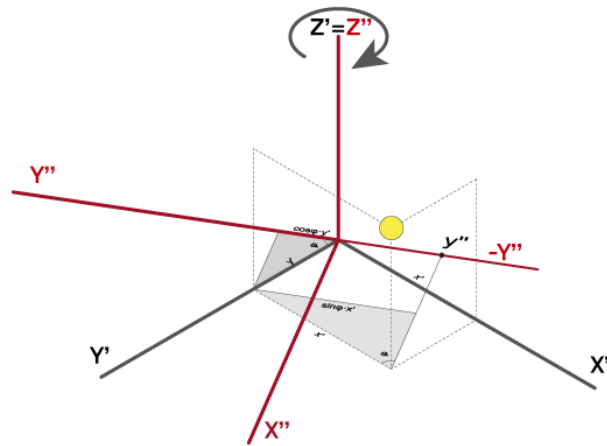


Figura 9: Canvi de base del sistema global al local. Component y''

Canvi de base amb gir del sistema local sobre l'eix y' (Pitch)

Sent C_{Pitch}^{Global} la matriu de canvi de base del sistema global al sistema local girat sobre l'eix Y' (Pitch):

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = C_{Pitch}^{Global} * \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

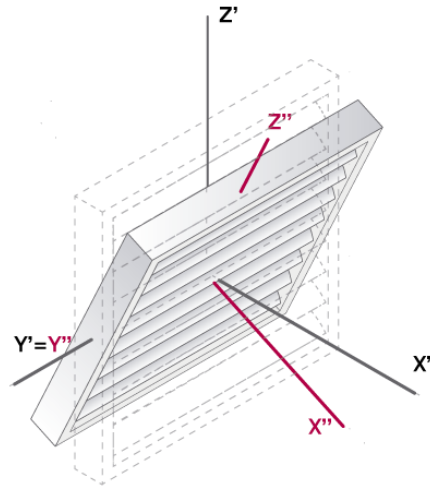


Figura 7: Gir del panell de lames sobre l'eix Y' (Pitch)

Canvi de base amb gir del sistema local sobre l'eix X'(Roll)

Sent C_{Roll}^{Global} la matriu de canvi de base del sistema global al sistema local girat sobre l'eix X'(Roll):

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = C_{Roll}^{Global} * \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

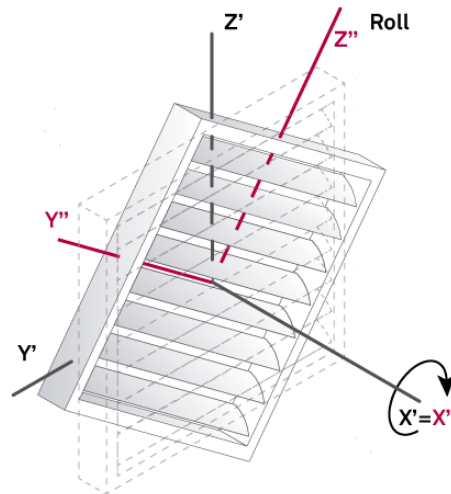


Figura 7: Gir del panell de lames sobre l'eix X' (Roll)

Canvi de base amb els tres girs (Heading, Pitch i Roll) conjuntament

Aleshores, podem obtenir una matriu de canvi de base, per obtenir la posició solar al sistema de coordenades local després de realitzar els tres girs nomenats a partir de la base inicial:

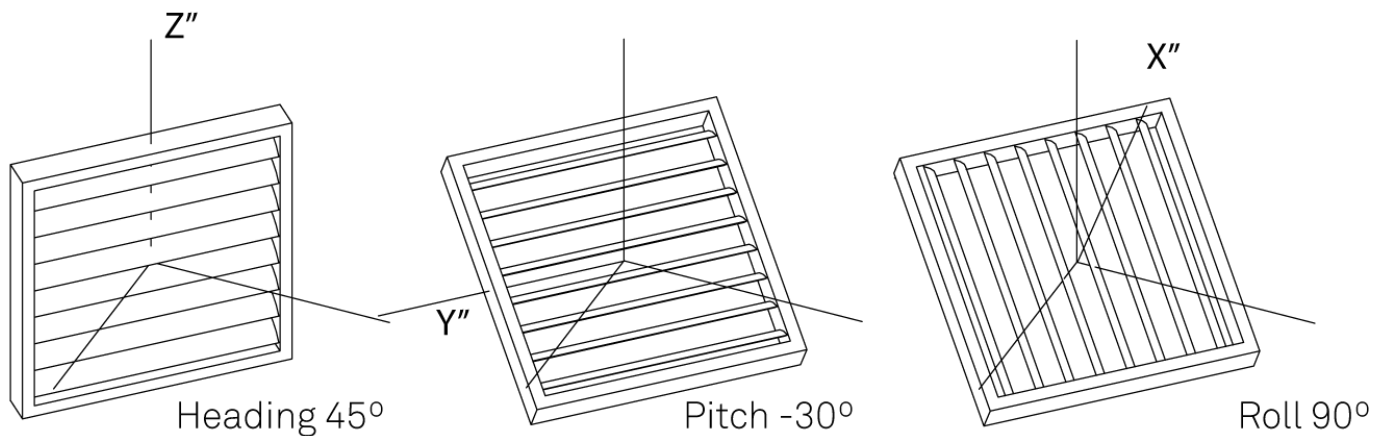
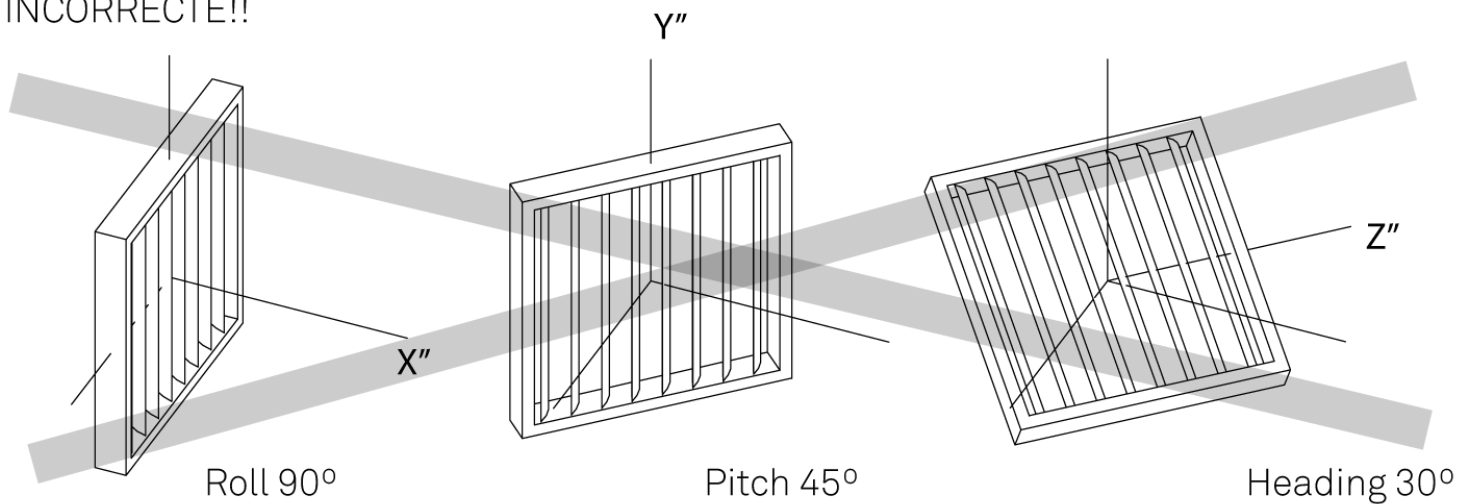
$$C_{Local}^{Global} = C_{Heading}^{Global} * C_{Pitch}^{Global} * C_{Roll}^{Global}$$

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\varphi & \cos\theta\sin\varphi & \sin\theta \\ \cos\varphi\sin\theta\sin\phi - \sin\varphi\cos\phi & \sin\varphi\sin\theta\sin\phi + \cos\varphi\cos\phi & \cos\theta\sin\phi \\ \cos\varphi\sin\theta\cos\phi + \sin\varphi\sin\phi & \sin\varphi\sin\theta\cos\phi - \cos\varphi\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

Cal tindre en compte que si canviem l'ordre al realitzar els girs per indicar la posició angular final, canviarà la matriu final, ja que la multiplicació de matrius no té la propietat commutativa i per tant els resultats no seran vàlids amb aquesta matriu final.

Per tant, per indicar els canvis angulars cal seguir sempre el mateix ordre de girs si volem utilitzar aquesta matriu de canvi de base general: Heading, Pitch i Roll consecutivament, i en aquest ordre.

INCORRECTE!!



CORRECTE

Figura 8:Seqüència incorrecta i correcta per introduir correctament el posicionament angular del dispositiu

5.1.4 Projecció d'ombra entre les lames.

Modelització.

Els sistemes de protecció solar basats en lames mòbils, solen estar compostos comunament per una seriació de lames extrusionades amb els eixos d'extrusió disposats paral·lelament a certa equidistància. El moviment de cadascuna de les lames es sol produir al voltant d'un eix paral·lel a aquest eix d'extrusió, i l'angle de moviment es manté constant en totes les lames .

Aquesta és la disposició que utilitzarem per al nostre prototip per ser la més comuna, fent coincidir la direcció de l'eix Y" del sistema local amb la direcció d'extrusió, l'eix X" local amb la direcció perpendicular a la superfície que conté tots els eixos de les lames, i l'eix Z" local perpendicular als altres dos, i per tant, al plànol anterior.

Aquestes característiques, junt al fet que els rajos de llum solar són paral·lels, fan que les diferents lames projecten la mateixa ombra sobre la lama consecutiva, i per tant, l'estudi del comportament de dos lames es pot fer extensiu a tot el conjunt, sense importar la seua mesura total.

Així, cada lama projectarà una ombra sobre el plànol de la següent, que formarà una línia entre la part ombrejada i il·luminada. Aquesta línia serà paral·lela al perfil que la provoca, i per tant, serà paral·lela a l'eix d'extrusió de les lames.

Aquesta línia es pot determinar com la intersecció del plànol que conté la línia del perfil que genera l'ombra i els rajos de llum de Sòl que intersequen amb aquesta, i la superfície de la lama.

L'hipotètic plànol límit entre llum i ombra descrit, serà perpendicular al plànol format pels eixos X"Z" del sistema de coordenades local.

També es dona el cas que aquest plànol interseccarà al mateix punt de qualsevol secció de la lama realitzada per qualsevol plànol perpendicular a l'eix d'aquesta, per tant l'estudi de qualsevol secció es pot generalitzar per al conjunt de totes elles.

L'angle que formen aquest plànol límit amb un hipotètic plànol que continga els eixos X''Y'' el nomenarem angle de projecció solar (AngSolar).

$$AngSolar = \text{atan2}\left(\frac{z''}{x''}\right)$$

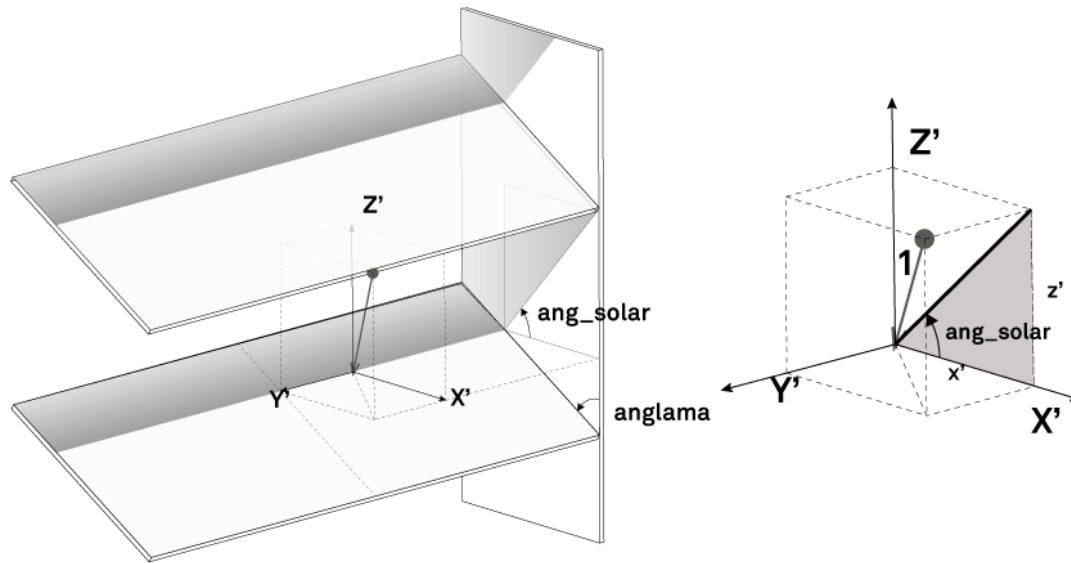


Figura 9: AngSolar

Utilitzant la funció atan2 es tindrà en compte el quadrant en que es posiciona el Sòl, ja que aquesta es determina depenent dels signe de z" i x".

Tenint en compte aquestes característiques, sols ens caldrà per a definir el model la distància entre els eixos de les lames i la longitud de la secció d'aquestes.

Les lames les modelitzarem amb perfil recte i amb espesor 0. Açò donarà certa variació entre el model i la realitat, en creixement quan menor siga l'angle entre la lama i el plànol límit d'ombra, però el podem despreciar per la seua poca magnitud, i perquè al mateix temps aquesta inexactitud sempre es decantarà en pro de l'ombregat interior del sistema.

Calcul de la zona ombrejada a la secció de les lames i el seu manteniment constant variant els seus angles d'inclinació.

Sent:

EntreLames= Separació entre els eixos de les lames.

LongLames= Longitud de la secció de les lames.

AngLama=Angle de les lames respecte de la vertical en sentit antihorari.

AngSolar= Angle de projecció solar respecte al planol X"- Y"

LongSolar=Longitud il·luminada de la secció de cadascuna de les lames.

LongOmbra=Longitud en ombra de la secció de cadascuna de les lames.

Aleshores:

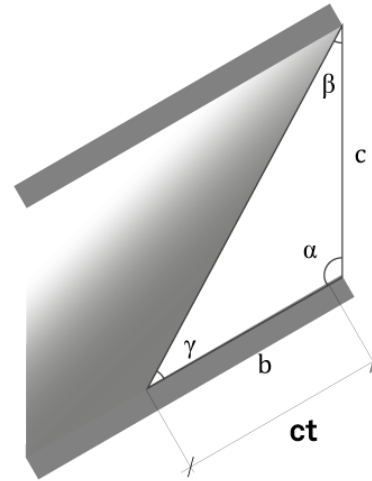
$$\alpha = \text{AngLama}$$

$$\beta = \pi/2 - \text{AngSolar}$$

$$\gamma = \pi/2 - \text{AngLama} + \text{AngSolar}$$

$$b = \text{LongSolar}$$

$$c = \text{EntreLames}$$



Tenint en compte el Teorema dels sinus:

$$\frac{a}{\sin\alpha} = \frac{b}{\sin\beta} = \frac{c}{\sin\gamma}$$

Aleshores:

$$\frac{\text{LongSolar}}{\sin(\pi/2 - \text{AngSolar})} = \frac{\text{EntreLames}}{\sin(\pi/2 - \text{AngLama} + \text{AngSolar})}$$

$$\frac{\text{LongSolar}}{\cos(\text{AngSolar})} = \frac{\text{EntreLames}}{\cos(\text{AngLama} - \text{AngSolar})}$$

$$\text{LongSolar} = \frac{\text{EntreLames} * \cos(\text{AngSolar})}{\cos(\text{AngLama} - \text{AngSolar})}$$

$$\text{LongOmbra} = \text{LongLames} - \text{LongSolar}$$

En el cas que volguérem mantindre la longitud solejada de les lames constant ($LongSolar$) (i per tant, la longitud sombrejada ($LongOmbra$) també ho seria), variant l'angle de les lames ($AngLama$) segons canvia l'angle de projecció solar ($AngSolar$), aleshores:

$$\cos(AngLama - AngSolar) = \frac{EntreLames * \cos(AngSolar)}{LongSolar_{ct}}$$

$$AngLama = AngSolar + \arccos\left(\frac{EntreLames * \cos(AngSolar)}{LongSolar_{ct}}\right)$$

Podent així mantenir la insolació directa de les lames constant, i per tant, la protecció solar directa de la zona a protegir també.

Es dona la singularitat que quan \arccos es major de 1, es crea una indeterminació. Açò es causat perquè no es possible complir la finalitat amb l'angolar que fa que es produísca esta situació.

En aquesta situació caldrà augmentar $LongSolar$ fins que $AngLama$ siga possible.

Aquesta situació es dona quan $EntreLames * \cos(AngSolar) = LongSolar$, i per tant $AngLama = AngSolar + \arccos(1) = AngSolar$

Angle d'inclinació de les lames per mantindre constant la superfície solejada a través del sistema de lames

Es pot donar el cas que volguérem mantindre constant la proporció de superfície solejada que passa a través del sistema de protecció, per tamisar l'entrada de Llum directa de manera constant, i per tant, caldria mantindre constant la superfície solejada al plànol interior del nostre aparell a estudiar.

Tenint en compte:

$$\alpha = \text{AngLama}$$

$$\beta = \pi/2 - \text{AngSolar}$$

$$\gamma = \pi/2 - \text{AngLama} + \text{AngSolar}$$

$$b = \text{LongLames}$$

$$c = \text{LongOmbraPer} \text{ (constant)}$$

Sent LongOmbraPer la longitud de la secció ombrejada a la part interior de l'aparell, aleshores, segons el Teorema dels Sinus anteriorment:

$$\frac{\text{LongLames}}{\sin(\pi/2 - \text{AngSolar})} = \frac{\text{LongOmbraPer}}{\sin(\pi/2 - \text{AngLama} + \text{AngSolar})}$$

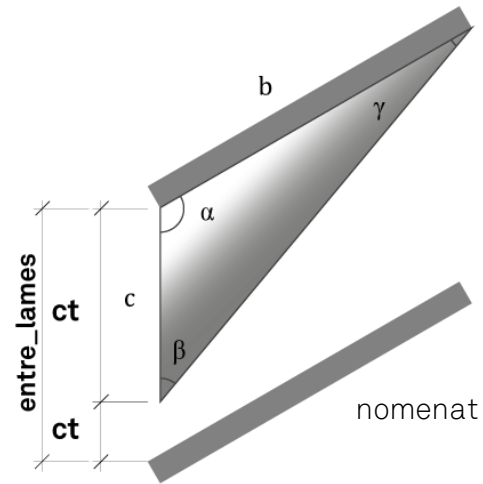
$$\frac{\text{LongLames}}{\cos(\text{AngSolar})} = \frac{\text{LongOmbraPer}}{\cos(\text{AngLama} - \text{AngSolar})}$$

$$\cos(\text{AngLama} - \text{AngSolar}) = \frac{\text{LongOmbraPer} * \cos(\text{AngSolar})}{\text{LongLames}}$$

$$\text{AngLama} = \text{AngSolar} + \text{acos}\left(\frac{\text{LongOmbraPer} * \cos(\text{AngSolar})}{\text{LongLames}}\right)$$

En aquest cas podríem tindre indeterminació si $\text{LongOmbraPer} * \cos(\text{AngSolar}) > \text{LongLames}$

Aquest cas es podria produir en el cas que la distància entre lames fos major que la longitud de la lama. Aquest cas és indesitjable ja que no podríem tancar per complet l'entrada de llum, i per tant no contemplarem la opció d'aquest disseny de dispositiu, i per tant, no es podrà produir.



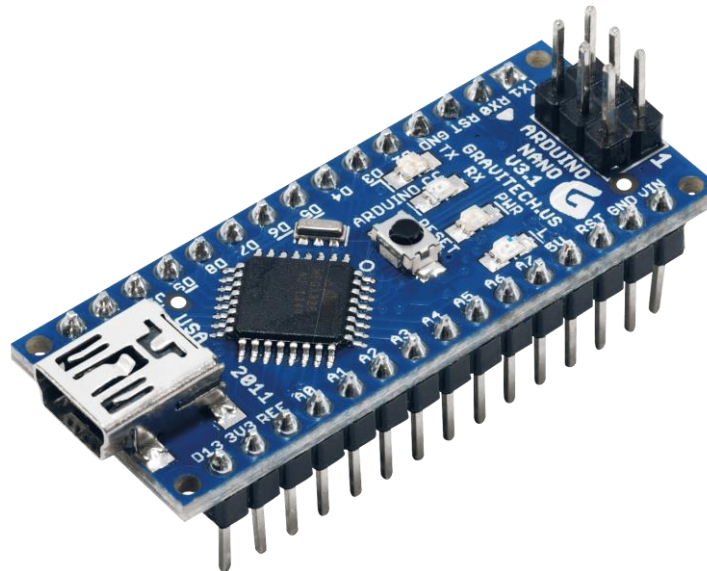
5.2 Sistema de programari i maquinari

5.2.1 Maquinari

Microprocessador

Utilitzem com a base de treball el sistema de maquinari Arduino, constituït essencialment per una placa de circuit imprès amb un microcontrolador de codi obert amb diverses entrades i eixides per comunicar dades amb els perifèrics per desenvolupar objectes interactius autònoms.

En aquest cas utilitzem el model Arduino Nano, amb el microprocessador ATmega 328.



5.2.1.1 Maquinari Perifèric

Per a la comunicació del microcontrolador amb l'entorn utilitzarem els següent perifèrics:

Servomotor

Un servomotor és un dispositiu actuator que té la capacitat de moure's un determinat angle i quedar-se en eixa posició establement, depenent de l'ample de polsos de la senyal de control (PWM).

Aquest està format per un motor, un circuit de control i una caixa reductora.

Per al control d'aquest al programari utilitzarem la llibreria *servo*, proporcionada per arduino, i la llibreria *Servo*, per controlar correctament la velocitat angular de gir.

Utilitzarem per al prototip el Servo SG90 Tower Pro amb capacitat de gir de 180°, i amb un torque de 1.8 Kg·cm.

Teòricament, segons les especificacions del producte, amb un ample de pols d'1, 1'5 i 2 milisegons es posiciona a -90°, 0°, i 90°, i els angles intermedis proporcionalment a aquests.

La realitat és que l'aparell no respon a la precisió desitjada amb aquestes premisses, i ha calgut crear una funció (`girservo()`) mitjançant programari per calibrar-lo i que responga correctament.

Aquest calibrat el realitzarem buscant mitjançant el mètode assaig/error l'ample de pols que posicione el servo en les desenes d'angle correctament i repartir proporcionalment l'interval entre cada posició mesurada pels graus intermedis, aconseguint així una resposta correcta a les necessitats.



RTC (rellotge temps real)

Dispositiu de càlcul de temps real en les unitats temporals que utilitzem normalment, amb capacitat de càlcul de l'hora, la data i l'any. Aquest dispositiu incorpora alimentació per mantindre les dades en cas de pèrdua d'alimentació. En aquest cas utilitzarem el model DS3231.

La comunicació amb el microprocessador es produeix per sèrie a través d'un bus bidireccional I2C.

A més a més el mòdul incorpora una EEPROM AT24C32 que pot ser utilitzada per a emmagatzemar dades des del microprocessador.

Utilitzarem la llibreria de programari DS3231, creada per Eric Ayars, per al seu control, i la llibreria AT24Cxx, creada per Manjutah CV, per guardar les dades de personalització a la EEPROM.



Potenciòmetre

Element amb una resistència variable, que s'utilitza com a divisor de tensió.

Amb aquest element, podem comunicar valors diferents de tensió al microprocessador depenent de la posició angular que li apliquem al maquinari de control. La relació entre el voltatge i les dades a extraure pot ser personalitzada al programari.

Aquestes dades les utilitzarem per a l'entrada de dades necessàries per a la configuració de l'aparell programada.



Polsadors

Operadors elèctrics que tanquen un circuit quan es mantenen polsats.

Instal·larem una resistència de 330 ohms a la connexió de 5V forçant una lectura de tensió alta quan no està polsat. Quan el polsem, tancarem el circuit amb el Ground i donarem una lectura de tensió baixa. Així evitem que el polsador es quede en un estat flotant al obrir el circuit amb el Ground, facilitant així la lectura digital.

Aquests els utilitzarem per al canvi de menus i seleccions a la configuració programada de l'aparell.

Per al seu ús utilitzarem la llibreria button creada per Michael Adams.



Dispositiu LCD

Pantalla LCD on el processador té la capacitat de comunicar-se visualment amb l'usuarida.

Utilitzarem un model amb capacitat de 2 línies amb 16 dígits cadascuna.

La comunicació amb el microprocessador es produeix també per sèrie a través d'un bus bidireccional I2C.

Per al seu control utilitzarem la llibreria LiquidCrystal, creada per DFROBOTS, i la llibreria LCD creada per Francisco Malpartida.

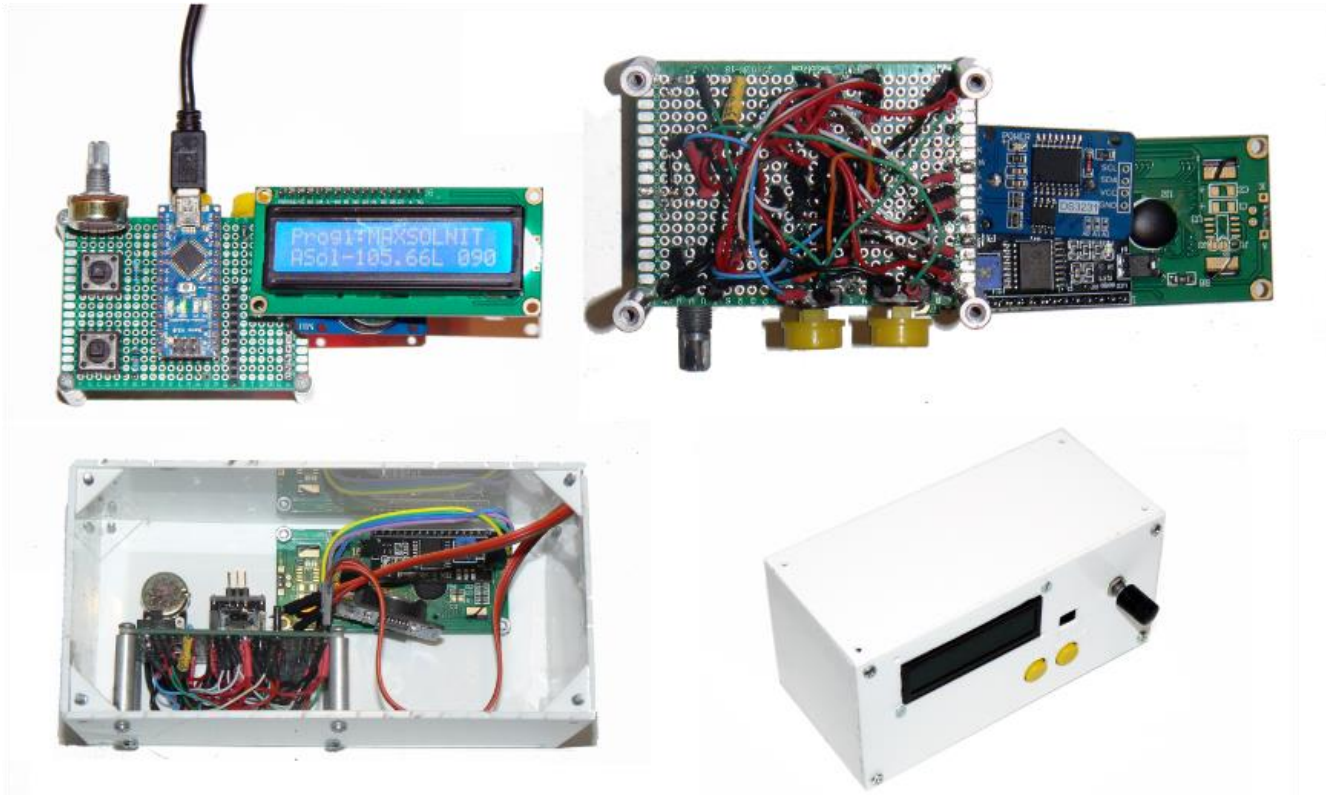
Per al seu correcte funcionament hem programat les funcions necessàries per a que responga a les necessitats: indicar-nos l'estat de funcionament i facilitar el procés de configuració i personalització de l'aparell.



5.2.1.2 Circuits

Tots aquests elements han sigut comprovats primerament per vore si l'esquema de connexió era correcte amb connexions mitjançant protoboard.

Quan s'ha comprovat el seu correcte funcionament, s'han muntant en una placa de prototips mitjançant cablejat i soldadura, per millorar i assegurar les seues connexions, i poder ensamblar correctament amb la caixa de protecció dissenyada.



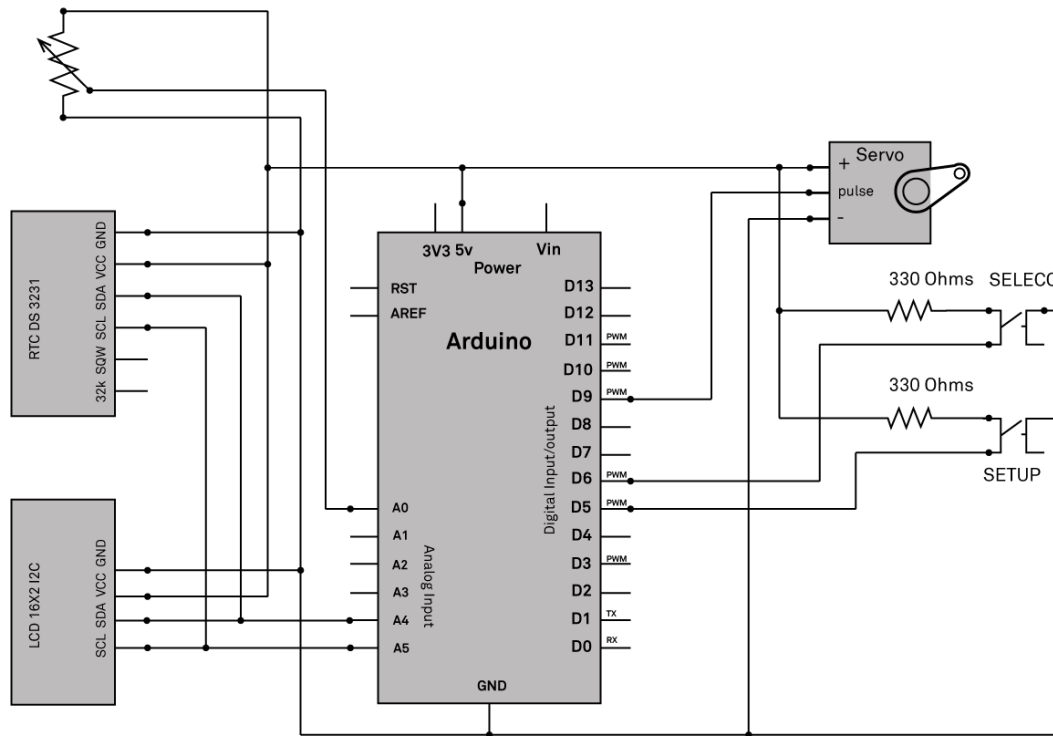
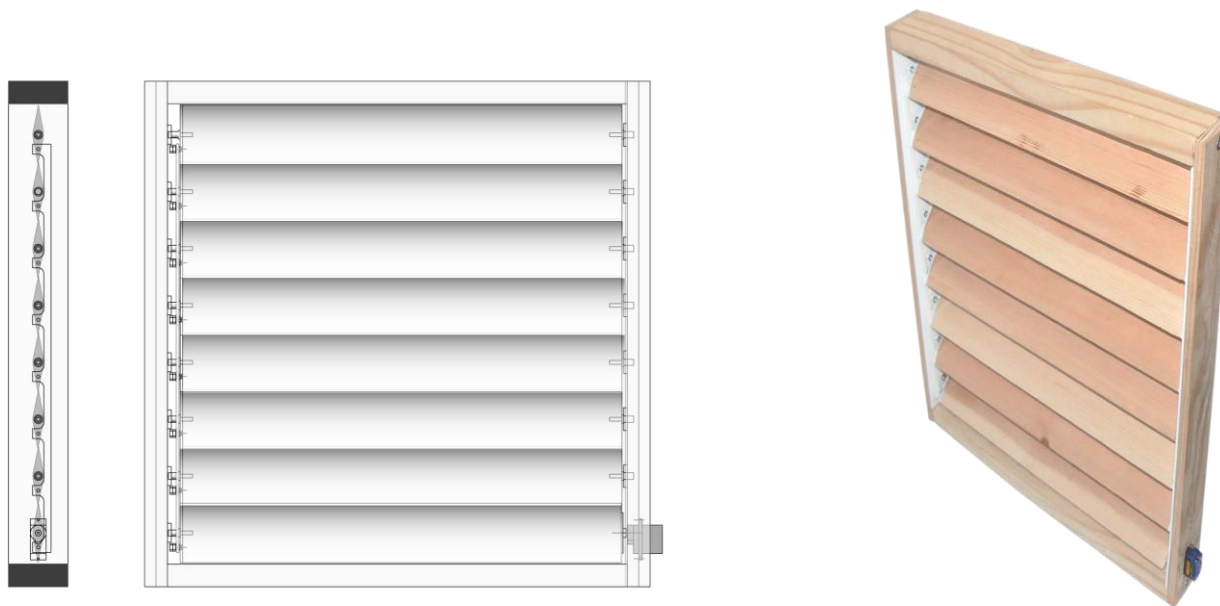


Figura 10: Esquema de connexions

5.2.1.3 Prototip de sistema de lames

S'ha construït com a prototip un sistema de lames de 40X40X8 cm amb 8 lames motoritzades mitjançant el servomotor, amb una caixa de protecció per a protegir el circuit i disposar dels mecanismes de comandament i la pantalla LCD.



5.2.2 Programari

El microprocessador arduino és programat mitjançant l'IDE proporcionat per la pròpia empresa fabricant, i utilitza com a codi el sistema de programació Processing, molt similar al codi C++.

Llibreries

Al ser un sistema de codi obert, existeixen milers de llibreries i exemples de programació lliures de drets compartides per la pròpia comunitat d'usuaris del maquinari, que faciliten certes feines metòdiques de la seua programació.

Aquestes són les llibreries que utilitzarem en el nostre cas, per la correcta utilització dels perifèrics, i facilitar el procediment de càlcul:

- **DS3231.h** Control del rellotge perifèric DS3231.
- **Wire.h** Comunicació amb el perifèrics mitjançant el protocol I2C.
- **math.h** Facilita els càlculs matemàtics. Conté la funció cos, sin, atan2, etc.
- **Servo.h** Llegeix les dades donades pel servo i les codifica segons les necessitats.
- **LCD.h** Codifica les ordres adequades per a la presentació de dades a la pantalla LCD.
- **LiquidCrystal_I2C.h** Millora la comunicació I2C amb la pantalla LCD.
- **I2Cdev.h** Millora la connexió dels dispositius I2C.
- **Button** Facilita la lectura dels polsadors i millora el seu funcionament.
- **AT24Cxx.h** Possibilita la utilització de la memòria interna del rellotge DS3231. Aquesta la utilitzarem per guardar permanentment les dades necessàries de configuració, perquè no es perden als talls de llum o reinicialitzacions.

Funcions

El codi està dividit en diverses funcions que realitzen tasques concretes, i que anirem executant-les depenent de les necessitats d'execució i càlcul del sistema. La utilització d'aquestes facilita la divisió i racionalització de tasques alhora de dissenyar programes complexos.

Funció Setup()

Primera funció que realitza el programa, i sols es realitza una vegada a l'inici. En aquesta s'executen les accions necessàries per a la configuració del sistema i la seua inicialització correcta. En el nostre cas, la utilitzem generalment per configurar la comunicació correcta amb el perifèrics i la lectura de les dades guardades de configuració a la eep.rom del rellotge SD3231 i la seua recodificació. Aquesta, és necessària perquè guarden les dades en bytes a la memòria, que sols té capacitat de magatzemar des del 0 fins al 255. Com utilitzem dades més extenses, amb decimals i signe, dividim les diferents dades en diferents entrades de memòria. Per tant, caldrà recodificar aquestes entrades per obtindre el valor de la variable final que és el que utilitzem per als diferents càlculs.

Aquesta funció és necessària per a qualsevol programació dels microprocessadors Arduino, i té que existir de forma obligatòria.

Les dades de configuració guardades a la eep.rom del rellotge en diferents entrades són les següents:

- Longitud geogràfica
- Latitud geogràfica
- Heading
- Pitch
- Roll
- Programa
- Ang_lama_pers
- Percent

Com a exemple, indicarem la descomposició en entrades de la longitud geogràfica:

Entrades de la dada	Nº memòria eep.rom	Funció
conflong100	1	Centenes
conflong1	2	Decenes
conflong01	3	Unitats
conflong001	4	Decimals
conflongsigne	5	signe

La recodificació per llegir la longitud guardada serà la següent:

```
if (eep.read(5)==0)
{
  conflongsigne2=-1;
}
else
{
  conflongsigne2=1;
}
*Pt_longitud=(eep.read(1)*10+eep.read(2)+eep.read(3)*0.1+eep.read(4)*0.01)*conflongsigne2;
```

Funció Loop()

Aquesta funció també és obligatòria al sistema Arduino, i té la particularitat que s'executa de forma reiterativa i continua mentre estiga en funcionament el sistema.

Des d'aquesta, s'executen com ja em dit contínuament les ordres establertes, discriminant aquelles operacions que necessitem per al correcte funcionament del sistema depenent de les dades que van canviant al llarg del temps, i que són aportades o bé automàticament pel maquinari, o bé manualment pel propi usuari interactuant amb els dispositius de configuració.

En aquest cas, principalment executa les següents activitats:

- Elecció de les funcions pertinents a executar depenent de les variables menu, programa, selecc i conf.
- Elecció de les diferents configuracions de representació de dades a la pantalla LCD.
- Lectura de l'estat dels polsadors i potenciòmetre, per si canvia aquest per part de la manipulació de l'usuari, canviar les variables de configuració i per tant la relació de funcions que s'executen.

Funcions setup___

Aquestes funcions són dedicades a la recollida de dades de configuració del dispositiu mitjançant els polsadors i el potenciòmetre, convertint la lectura de les variacions d'aquests en les dades per a la configuració per part de l'usuari del dispositiu.

Aquesta configuració es possible mitjançant l'entrada de dades a través dels dos botons disponibles i del potenciòmetre.

El botó *selecc* ens serveix per canviar de data d'entrada dintre de cada menú o submenú de configuració, variant les dades a configurar girant el potenciòmetre.

El botó setup ens servirà per canviar el nivell de menú al que ens trobem. Existeixen tres nivells de menú: El nivell 0, on es troba en execució la funció del programa seleccionat i amb el botó *selecc* podem modificar la informació que ens aporta la pantalla, el nivell 1, on podem canviar amb el botó *selecc* la configuració que volem introduir, i el nivell 2, on modifiquem les dades de configuració de la variable seleccionada al nivell 1.

Amb aquestes funcions configurarem :

- El programa de funcionament en execució.
- El percentatge de superfície solejada o el percentatge de les lames en ombra (dades que s'utilitzen al programa0).

- L'angle de les lames per a la posició standby (quan el Sòl no incideix sobre el sistema, i al programa2).
- La data i l'hora.
- La longitud de localització terrestre i la latitud.
- El Heading, el Pitch i el Roll de posicionament angular de l'element respecte a l'eix de coordenades local.
- L'engegada del funcionament en mode test (indica les dades de càlcul i simula el funcionament de les lames a cada hora en punt del dia elegit, canviant a l'hora següent cada cinc segons).

Una vegada s'han inserit les dades per part de l'usuari, també guarden aquestes a la eeprom del rellotge.

Funcions lcd_____

Aquestes s'utilitzen per a la representació en la pantalla LCD de les dades necessàries per a la correcta configuració del dispositiu i transmetre informació a l'usuari.

Al nivell de menú 0, ens informa en temps real de l'hora, la data, el azimuth, l'elevació del Sòl, l'angle de les lames, la variable AngSolar descrita anteriorment, i el programa de configuració en funcionament.

Al nivell de menú 1, ens informa de les dades de configuració actuals de cada variable de configuració.

Al nivell 2, ens informa de les dades que va elegint l'usuari, amb el gir del potenciòmetre, per a l'elecció de la configuració de cada dada.

Funció programa0()

Amb aquest programa controlem les lames per respondre als paràmetres de configuració i personalització elegits per l'usuari. És manté el percentatge elegit de superfície solejada que incideix a través del sistema al llarg del temps, i es manté constant mentre incidisca el Sòl sobre el sistema, o bé impedeix totalment la radiació solar directa, mantenint la superfície radiada sobre les lames de protecció constant si el sistema ho permet, i en cas contrari, la menor més pròxima a la programada per l'usuari.

Quan és de nit, o el Sòl no irradia sobre les lames, manté les lames en una posició d'standby també elegida per l'operador.

Funció programa1()

Manté les lames en direcció al Sòl perquè entre la màxima radiació possible.

Quan és de nit, o el Sòl no irradia sobre les lames, manté les lames en posició standby, aquesta també elegida per l'operador.

Funció programa2()

Manté les lames en posició standby .

Funció programa3()

Tanca les lames del sistema.

Funció posicionsolar()

Calcula la posició angular del Sòl respecte al sistema angular de coordenades global. Per aquest càlcul necessita la data, l'hora, la longitud de posicionament terrestre i la latitud, retornant l'azimut i l'elevació angular solar.

Funció canvidebase()

Canvia les coordenades de posició angular del Sòl al sistema de coordenades cartesià local, depenent de la posició relativa angular d'aquest respecte el primer, amb les dades aportades de l'usuari Heading, Pitch i Roll. Retorna la variable AngSolar, utilitzada a la funció programa0 i programa1.

Girservo()

Calibració del servo per a que responga correctament a les ordres de posicionament de les lames que executa el sistema

6. Conclusions

El treball present compleix amb les expectatives creades per l'autor sobre el control de la il·luminació directa, ajudant a fer més eficient aquest sistema de protecció solar.

Al mateix temps obri un gran ventall de possibilitats per a nous estudis al respecte utilitzant les funcions creades per tal de perfeccionar el sistema, tenint en compte altres variables, i tenint en compte aspectes que van més lluny de la finalitat d'aquest estudi, com poden ser el control lumínic de les estàncies per a un millor confort, l'eliminació d'enlluernaments indesitjats de forma automatitzada o el control de la temperatura.



7. Bibliografía

Marc Figueras; Diccionari d'astronomia de posició

Guillermo de la Paz Pérez; Brise-soleil, recurso arquitectónico de control solar. Evolución y propuesta de diseño optimizado para Camagüey. Arquitectura y Urbanismo vol. XXXIII, no 2, 2012

Norbert Lechner; Heating, cooling, lighting. Sustainable methods for architects. Editorial John Wiley & sons. 2015

David Brooks; Arduino Uno and Solar Position Calculations. Institute for Earth Science Research and Education .
www.instesre.org

Jean Meeus; Astronomical algorithms. Editorial Willmann-Bell. 1991

Gonzalo Ferrer Mínguez;; Integración Kalman de sensores inerciales INS con GPS en un UAV. Proyecto final de carrera. RSLab TSC UPC.2009.

8. Annex

Programari Íntegre

```

/*Configurar a hora UTC.
No es contempla el Daylight Saving Time (DST)
http://www.worldtimeserver.com/hora-exacta-UTC.aspx
*/
#include <DS3231.h> //llibreria rellotge
#include <Wire.h> //llibreria per a utilitzar I2C
#include <math.h> //llibreria calculs matemàtics
#include <Servo.h> //llibreria per a utilitzar el servo
#include <LCD.h> //llibreria per a utilitzar el LCD
#include <LiquidCrystal_I2C.h> //llibreria per a utilitzar el LCD
#include "I2Cdev.h" //llibreria per millorar la connexió I2C
#include <Button.h> //llibreria ús de botons
#include <AT24Cxx.h>

#define i2c_address 0x57

AT24Cxx eep(i2c_address,32);

//Crea elements del rellotge
DS3231 Clock;
bool Century=false;
bool h12;
bool PM;
byte ADay, AHour, AMinute, ASecond, ABits;
bool ADy, A12h, Apm;
int year, month, date, DoW, hour, minute, second,day;

//Crea elements de configuració
int confdata=0;
int confdia,confmes,confany;
int confhora=0,confh,confmin,confseg;
int confhead=0,confhead100,confhead1; // eep 15,16
int confx=0,confx100,confx1; // eep 11,12
int confy=0,confy100,confy1; // eep 13,14
int heading,roll,pitch;
float longitud,latitut;
int conflong=0,conflong100,conflong1,conflong01,conflong001,conflongsigne,conflongsigne2; // eep
1,2,3,4,5
int conflat=0,conflat100,conflat1,conflat01,conflat001,conflatsigne,conflatsigne2; //eep 6,7,8,9,10

```

```

int programa; // eep 17
int ang_lama_pers; // eep 18
int percent; // eep 19

//variables de configuració
int zone=1; //GMT (diferència horària de UTC de la zona horària)
PERSONALITZABLE
float entre_lames=4; // distància en l'eix z en cm entre lames
PERSONALITZABLE
float long_lames=4; //longitud en l'eix x de la lama en cm
PERSONALITZABLE

//crea variables posicionsolar
int A,B;
long JD_whole,JDx;
float Lon,Lat,JD_frac,T,L0,M,e,C,f,Obl,GrHrAngle,L_true,R,RA,Decl,HrAngle,elev,azimuth;
float long_solar;
float ang_solar;
float LongOmbraPer;
int menu;
int test;
int selecc;
float ang_lama;

float *Pt_ang_lama=&ang_lama;
int *Pt_programa=&programa;
int *Pt_ang_lama_pers=&ang_lama_pers;
int *Pt_percent=&percent;
float *Pt_azimuth=&azimuth;
float *Pt_elev=&elev;
int *Pt_menu=&menu;
int *Pt_test=&test;
int *Pt_heading=&heading;
float *Pt_longitut=&longitut;
float *Pt_latitut=&latitut;
int *Pt_pitch=&pitch;
int *Pt_roll=&roll;

//Crea constants

```

```

#define DEG_TO_RAD 0.01745329
#define PI 3.141592654
#define TWOPI 6.28318531

//Defineix elements del LCD
#define I2C_ADDR 0x3F
#define BACKLIGHT_PIN 3
#define En_pin 2
#define Rw_pin 1
#define Rs_pin 0
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7

LiquidCrystal_I2C lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pin);

Servo servo1;          // crea objecte tipus servo nomenat servo1

Button button1(6); // BOTÓ SELECCIÓ

Button button2(5); // BOTÓ SETUP
int compte=200;
int conf=0;

void setup() {
  Serial.begin(115200); // Engega el monitor serial 115200 bps
  //inicialitza els botons

  button1.begin();
  button2.begin();
  Wire.begin();
  lcd.begin(16,2); //definició del LCD com 16x2
  Serial.begin(115200); // Engega el monitor serial 115200 bps

  // Engega la llum LCD
  lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
  lcd.setBacklight(HIGH);
  lcd.home(); // go home

```

```

// Conectar servo1 al pin 9
servo1.attach(9,900,2560);

*Pt_menu=0;
selecc=1;
*Pt_ang_lama_pers=eep.read(18);
*Pt_heading=eep.read(15)*10+eep.read(16);

if (eep.read(5)==0)
    {
        conflongsigne2=-1;
    }
else
    {
        conflongsigne2=1;
    }
*Pt_longitut=(eep.read(1)*10+eep.read(2)+eep.read(3)*0.1+eep.read(4)*0.01)*conflongsigne2;

if (eep.read(10)==0)
    {
        conflatsigne2=-1;
    }
else
    {
        conflatsigne2=1;
    }
*Pt_latitut=(eep.read(6)*10+eep.read(7)+eep.read(8)*0.1+eep.read(9)*0.01)*conflatsigne2;

*Pt_roll=eep.read(11)*10+eep.read(12);

*Pt_pitch=eep.read(13)*10+eep.read(14);
*Pt_percent=eep.read(19)-100;
*Pt_programa=eep.read(17);
*Pt_test=0;
}

void loop()
{

```



```

while (menu==0)
{
  if (programa==0)
  {
    lcd.setCursor (0,0);
    lcd.print("Prog0:PERS ");
    programa0();
  }
  if (programa==1)
  {
    lcd.setCursor (0,0);
    lcd.print("Prog1:MAXSOL");
    programa1();
  }
  if (programa==2)
  {
    lcd.setCursor (0,0);
    lcd.print("Prog2:ESTATIC ");
    programa2();
  }
  if (programa==3)
  {
    lcd.setCursor (0,0);
    lcd.print("Prog3:TANCAT ");
    programa3();
  }
  lcdselecc(selecc);

  if (button1.pressed())
  {
    if (selecc==3)
    {
      selecc=1;
    }
    else
    {
      selecc++;
    }
  }
}

```

```

    lcd.clear ();
}
if (button2.pressed())
{
    *Pt_menu=1;
    lcd.clear ();
}
}
while (menu==1)
{
    menu1(conf);
    if (button1.pressed())
    {
        if (conf==11)
        {
            conf=0;
        }
        else
        {
            conf++;
        }
    }
    lcd.clear ();
}
if (button2.pressed())
{
    if (conf==1)
    {
        *Pt_menu=0;
        conf=0;
    }
    else
    {
        *Pt_menu=2;
    }
    lcd.clear ();
}
}
while (menu==2)

```

```
{
  if (conf==0)
  {
    setupprog();
  }
  if (conf==2)
  {
    setupsolar();
  }
  if (conf==3)
  {
    setuppers();
  }
  if (conf==4)
  {
    setupdata();
  }
  if (conf==5)
  {
    setuphora();
  }
  if (conf==6)
  {
    setuplat();
  }
  if (conf==7)
  {
    setuplong();
  }
  if (conf==8)
  {
    setupheading();
  }
  if (conf==9)
  {
    setuppitch();
  }
  if (conf==10)
  {
```

```

        setuproll();
    }
    if (conf==11)
    {
        setuptest();
    }
    lcd.clear ();
}
Serial.println(eep.read(18));
Serial.println(" ");
}

// PROGRAMES DE FUNCIONAMENT

void programa0()
{
    posicionsolar();
    canvibase(roll,pitch,heading,azimuth,elev);
    if (elev>0) // És de dia
    {
        if ((ang_solar>-PI/2)&&(ang_solar<PI/2)) // el Sòl jncideix a les lames
        {
            if (percent>=0)
            {
                LongOmbraPer=entre_lames*(1-percent*0.01);
                *Pt_ang_lama=ang_solar+acos(LongOmbraPer*cos(ang_solar)/long_lames);
            }
            else
            {
                long_solar=-percent*0.01*long_lames;
                if (long_solar>=entre_lames*cos(ang_solar))
                {
                    *Pt_ang_lama=ang_solar+acos(entre_lames*cos(ang_solar)/long_solar);
                }
                else
                {
                    *Pt_ang_lama=ang_solar;
                }
            }
        }
    }
}

```

```

        girservo(ang_lama/DEG_TO_RAD);
        lcd.setCursor (12,0);
        lcd.print("AUTO");
    }
    else // El Sòl no incideix a les lames
    {
        girservo(ang_lama_pers);
        *Pt_ang_lama=ang_lama_pers*DEG_TO_RAD;
        lcd.setCursor (13,0);
        lcd.print("OMB");
    }
}
else // és de nit
{
    girservo(ang_lama_pers);
    *Pt_ang_lama=ang_lama_pers*DEG_TO_RAD;
    lcd.setCursor (12,0);
    lcd.print("NIT ");
}
}
void programa1()
{
    posicionsolar();
    canvibase(roll,pitch,heading,azimuth,elev);
    if (elev>0) // És de dia
    {
        if ((ang_solar>-PI/2)&&(ang_solar<PI/2)) // el Sòl jncideix a les lames
        {
            *Pt_ang_lama=(PI/2)+ang_solar;
            girservo(ang_lama/DEG_TO_RAD);
            lcd.setCursor (12,0);
            lcd.print("AUTO");
        }
        else // El Sòl no incideix a les lames
        {
            girservo(ang_lama_pers);
            *Pt_ang_lama=ang_lama_pers*DEG_TO_RAD;
            lcd.setCursor (13,0);
            lcd.print("OMB");
        }
    }
}

```

```

    }
  }
  else // és de nit
  {
    girservo(ang_lama_pers);
    *Pt_ang_lama=ang_lama_pers*DEG_TO_RAD;
    lcd.setCursor (12,0);
    lcd.print("NIT ");
  }
}
void programa2()
{
  girservo(ang_lama_pers);
  *Pt_ang_lama=ang_lama_pers*DEG_TO_RAD;
}
void programa3()
{
  girservo(179);
  *Pt_ang_lama=179*DEG_TO_RAD;
}

```

// SELECCIÓ MENUS DE CONFIGURACIÓ

```

void menu1(int conf)
{
  if (conf==0)
  {
    lcd.setCursor (0,0);
    lcd.print("PROGRAMA");
    lcdprograma();
  }
  if (conf==1)
  {
    lcd.setCursor (0,0);
    lcd.print("EXIT");
  }
  if (conf==2)
  {
    lcd.setCursor (0,0);
  }
}

```

```

lcd.print("SUP INCID SOLAR");
lcdsolar();
}
if (conf==3)
{
lcd.setCursor (0,0);
lcd.print("ANG LAMA PERS");
lcdpers();
}
if (conf==4)
{
lcd.setCursor (0,0);
lcd.print("DATA");
lcddata();
}
if (conf==5)
{
lcd.setCursor (0,0);
lcd.print("HORA");
lcdrellotge();
}
if (conf==6)
{
lcd.setCursor (0,0);
lcd.print("LATITUT");
lcdlatitut();
}
if (conf==7)
{
lcd.setCursor (0,0);
lcd.print("LONGITUT");
lcdlongitut();
}
if (conf==8)
{
lcd.setCursor (0,0);
lcd.print("HEADING SUD");
lcd.setCursor (0,1);
}

```

```

//heading=eep.read(15)*10+eep.read(16);
lcdheading();
}
if (conf==9)
{
lcd.setCursor (0,0);
lcd.print("GIR SOBRE EIX Y");
lcdpitch();
}
if (conf==10)
{
lcd.setCursor (0,0);
lcd.print("GIR SOBRE EIX X");
lcdroll();
}
if (conf==11)
{
lcd.setCursor (0,0);
lcd.print("TEST");
lcdtest();
}
}

```

```

// SETUPS DE CONFIGURACIÓ
void setupprog()
{
lcd.setCursor (0,0);
lcd.print("PROGRAMA SET ");
while (menu==2)
{
if (button1.pressed())
{
if (programa>=3)
{
*Pt_programa=0;
}
else
{
*Pt_programa=programa+1;
}
}
}
}

```



```

    }
  }
  lcdprograma();
  if (button2.pressed())
  {
    *Pt_menu=1 ;
  }
}
}
void setupsolar()
{
  lcd.setCursor (0,0);
  lcd.print("INCIDENC SOL SET");
  while (menu==2)
  {
    int percent2=map(analogRead(A0),0,1020,0,200);
    eep.write(19,percent2);
    *Pt_percent=percent2-100;
    lcdsolar();
    if (button2.pressed())
    {
      *Pt_menu=1 ;
    }
  }
}
void setuppers()
{
  lcd.setCursor (0,0);
  lcd.print("ANG LAMA PER SET");
  while (menu==2)
  {
    *Pt_ang_lama_pers=map(analogRead(A0),0,1020,1,179);
    girservo(ang_lama_pers);
    lcdpers();
    if (button2.pressed())
    {
      *Pt_menu=1 ;
    }
  }
}

```

```

    writeang();
}
void setupdata()
{
    lcd.setCursor (0,0);
    lcd.print("DATA SETUP");
    while (menu==2)
    {
        if (confdata==0)
        {
            confdia=map(analogRead(A0),0,1020,1,31);
            Clock.setDate(confdia);
            lcddata();
        }
        if (confdata==1)
        {
            int confmes=map(analogRead(A0),0,1020,1,12);
            Clock.setMonth(confmes);
            lcddata();
        }
        if (confdata==2)
        {
            int confany=map(analogRead(A0),0,1020,15,30);
            Clock.setYear(confany); //Set the year (Last two digits of the year)
            lcddata();
        }
        if (button1.pressed())
        {
            if (confdata==2)
            {
                confdata=0;
                *Pt_menu=1 ;
            }
            else
            {
                confdata++;
            }
        }
        if (button2.pressed())

```

```

        {
            *Pt_menu=1 ;
            confdata=0;
        }
    }
}
void setuphora()
{
    lcd.setCursor (0,0);
    lcd.print("HORA SET");
    while (menu==2)
    {
        if (confhora==0)
        {
            confh=map(analogRead(A0),0,1020,0,23);
            Clock.setHour(confh);
            lcdrellotge();
        }
        if (confhora==1)
        {
            confmin=map(analogRead(A0),0,1020,0,59);
            Clock.setMinute(confmin);
            lcdrellotge();
        }
        if (confhora==2)
        {
            confseg=map(analogRead(A0),0,1020,0,59);
            Clock.setSecond(confseg);
            lcdrellotge();
        }
        if (button1.pressed())
        {
            if (confhora==2)
            {
                confhora=0;
                *Pt_menu=1 ;
            }
            else
            {

```

```

        confhora++;
    }
}
if (button2.pressed())
{
    *Pt_menu=1;
    confhora=0;
}
}
}
void setuplong()
{
    lcd.setCursor (0,0);
    lcd.print("LONGITUT SETUP");
    while (menu==2)
    {
        if (conflong==0)
        {
            conflong100=map(analogRead(A0),0,1020,0,17);
            eep.write(1, conflong100);

        }
        if (conflong==1)
        {
            conflong1=map(analogRead(A0),0,1020,0,9);
            eep.write(2, conflong1);

        }
        if (conflong==2)
        {
            conflong01=map(analogRead(A0),0,1020,0,9);
            eep.write(3, conflong01);

        }
        if (conflong==3)
        {
            conflong001=map(analogRead(A0),0,1020,0,9);
            eep.write(4, conflong001);
        }
    }
}

```

```

    }
    if (conflong==4)
    {
        conflongsigne=map(analogRead(A0),0,1020,0,1);
        eep.write(5,conflongsigne);
        lcd.setCursor (8,1);
        lcd.print("SIGNE");

    }
    if (eep.read(5)==0)
    {
        conflongsigne2=-1;
    }
    else
    {
        conflongsigne2=1;
    }
    *Pt_longitut=(eep.read(1)*10+eep.read(2)+eep.read(3)*0.1+eep.read(4)*0.01)*conflongsigne2;
    lcdlongitut();
    if (button1.pressed())
    {
        if (conflong==4)
        {
            conflong=0;
            *Pt_menu=1 ;
        }
        else
        {
            conflong=conflong+1;
        }
    }
    if (button2.pressed())
    {
        conflong=0;
        *Pt_menu=1 ;
    }
}
}
void setupplat()

```

```

{
  lcd.setCursor (0,0);
  lcd.print("LATITUT SETUP");
  while (menu==2)
  {
    if (conflat==0)
    {
      confla100=map(analogRead(A0),0,1020,0,8);
      eep.write(6,conflat100);

    }
    if (conflat==1)
    {
      confla1=map(analogRead(A0),0,1020,0,9);
      eep.write(7,conflat1);
    }
    if (conflat==2)
    {
      confla01=map(analogRead(A0),0,1020,0,9);
      eep.write(8,conflat01);
    }
    if (conflat==3)
    {
      confla001=map(analogRead(A0),0,1020,0,9);
      eep.write(9,conflat001);
    }
    if (conflat==4)
    {
      conflatsigne=map(analogRead(A0),0,1020,0,1);
      eep.write(10,conflatsigne);
      lcd.setCursor (8,1);
      lcd.print("SIGNE");

    }
    if (eep.read(10)==0)
    {
      conflatsigne2=-1;
    }
    else

```

```

    {
        conflatsigne2=1;
    }
    *Pt_latitut=(eep.read(6)*10+eep.read(7)+eep.read(8)*0.1+eep.read(9)*0.01)*conflatsigne2;

    lcdlatitut();
    if (button1.pressed())
    {
        if (conflat==4)
        {
            conflat=0;
            *Pt_menu=1 ;
        }
        else
        {
            conflat++;
        }
    }
    if (button2.pressed())
    {
        conflat=0;
        *Pt_menu=1 ;
    }
}
}
void setupheading()
{
    lcd.setCursor (0,0);
    lcd.print("HEADING SUD SET");
    while (menu==2)
    {
        if (confhead==0)
        {
            confhead100=map(analogRead(A0),0,1020,0,35);
            eep.write(15,confhead100);
        }
        if (confhead==1)
        {

```

```

        confhead1=map(analogRead(A0),0,1020,0,9);
        eep.write(16,confhead1);

    }
    *Pt_heading=eep.read(15)*10+eep.read(16);
    lcdheading();
    if (button1.pressed())
    {
        if (confhead==1)
        {
            confhead--;
            *Pt_menu=1 ;
        }
        else
        {
            confhead++;
        }
    }

    if (button2.pressed())
    {
        *Pt_menu=1;
        confhead=0;
    }
}
}
void setuppitch()
{
    lcd.setCursor (0,0);
    lcd.print("GIR EIX Y SETUP");
    while (menu==2)
    {
        if (confy==0)
        {
            confy100=map(analogRead(A0),0,1020,0,35);
            eep.write(13,confy100);
        }
        if (confy==1)
        {

```



```

        confy1=map(analogRead(A0),0,1020,0,9);
        eep.write(14,confy1);
    }
    *Pt_pitch=eep.read(13)*10+eep.read(14);
    lcdpitch();
    if (button1.pressed())
    {
        if (confy==1)
        {
            confy--;
            *Pt_menu=1 ;
        }
        else
        {
            confy++;
        }
    }
    if (button2.pressed())
    {
        confy=0;
        *Pt_menu=1 ;
    }
}
}
void setuproll()
{
    lcd.setCursor (0,0);
    lcd.print("GIR EIX X SETUP");
    while (menu==2)
    {
        if (confx==0)
        {
            confx100=map(analogRead(A0),0,1020,0,35);
            eep.write(11,confx100);
        }
        if (confx==1)
        {
            confx1=map(analogRead(A0),0,1020,0,9);
            eep.write(12,confx1);
        }
    }
}

```

```

    }
    *Pt_roll=eep.read(11)*10+eep.read(12);
    lcdroll();
    if (button1.pressed())
    {
        if (confx==1)
        {
            confx--;
            *Pt_menu=1 ;
        }
        else
        {
            confx++;
        }
    }
    if (button2.pressed())
    {
        confx=0;
        *Pt_menu=1 ;
    }
}
}
}
void setuptest()
{
    lcd.setCursor (0,0);
    lcd.print("TEST SETUP      ");
    while (menu==2)
    {
        if (button1.pressed())
        {
            if (test==1)
            {
                *Pt_test=0;
            }
            else
            {
                *Pt_test=1;
            }
        }
    }
}

```

```

    lcdtest();
    if (button2.pressed())
    {
        *Pt_menu=1 ;
    }
}

// CALCULS POSICIONAMENT SOLAR

void posiciosolar()
{
    if (test==0)
    {
        second=Clock.getSecond();           // Agafa l'hora del DSC
        minute=Clock.getMinute();
        hour=Clock.getHour(h12,PM);
        if (h12)                             // Passa l'hora a 24h
        {
            if (PM)
            {
                hour=hour+12;
                if (hour>=24)
                {
                    hour=hour-24;
                }
            }
        }
    }
}
else
{
    second=0;
    minute=0;
    if (Clock.getSecond()-(int(Clock.getSecond()/10))*10==0||Clock.getSecond()-(int(Clock.getSecond()/10))*10==5)
    {
        if (hour>=23)
        {
            hour=6;
        }
    }
}
}

```

```

    }
    else
    {
        hour=hour+2;
    }
    delay(1300);
}
}

    int day=Clock.getDate();
    int month=Clock.getMonth(Century);
    int year=Clock.getYear();
    year=2000+year; //posa milers i centenes a l'any

if (month<=2)
{
    year--; month+=12;
}
    A=year/100;
    B=2-A+A/4;

JD_whole=(long)(365.25*(year+4716))+(int)(30.6001*(month+1))+(day)+B-1524; //Dia Julià, part sencera
    Lon=(longitud)*DEG_TO_RAD;
    Lat=(latitud)*DEG_TO_RAD; // Passa LONGITUD I LATITUD a RAD
    JD_frac=(hour+minute/60.+second/3600.)/24.-.5; // dia Julià, part fraccional
    T=JD_whole-2451545; T=(T+JD_frac)/36525.; // segle des de 1 de gener de 2015 a les 12:00 UT
    L0=DEG_TO_RAD*fmod(280.46645+36000.76983*T,360); //longitud solar
    M=DEG_TO_RAD*fmod(357.5291+35999.0503*T,360); //Solar mean anomaly
    e=0.016708617-0.000042037*T; //excentricitat de l'òrbita de la terra
    C=DEG_TO_RAD*((1.9146-0.004847*T)*sin(M)+(0.019993-0.000101*T)*sin(2*M)+0.00029*sin(3*M));
    // sun's equation of center
    f=M+C;//solar true anomaly
    Obl=DEG_TO_RAD*(23+26/60.+21.448/3600.-46.815/3600*T);//oblicuitat de l'equador
    JDx=JD_whole-2451545;
    GrHrAngle=280.46061837+(360*JDx)%360+.98564736629*JDx+360.98564736629*JD_frac;
    //Greenwich hour angle
    GrHrAngle=fmod(GrHrAngle,360.);
    L_true=fmod(C+L0,TWOPI);
    R=1.000001018*(1-e*e)/(1+e*cos(f));//distància de la Terra al Sòl
    RA=atan2(sin(L_true)*cos(Obl),cos(L_true));//Right ascension
    Decl=asin(sin(Obl)*sin(L_true));//Declinació

```

```

        HrAngle=DEG_TO_RAD*GrHrAngle+Lon-RA;//Hour angle
*Pt_azimuth=atan2(sin(HrAngle),cos(HrAngle)*sin(Lat)-tan(Decl)*cos(Lat));
// Azimuth measured eastward from north.
        *Pt_elev=asin(sin(Lat)*sin(Decl)+cos(Lat)*(cos(Decl)*cos(HrAngle)));//elevació del sol
}
float canvibase(int gir_x,int gir_y,int gir_z,float az,float el)
{
    Serial.print("azimuth: ");
    Serial.print(az/DEG_TO_RAD);
    Serial.print("\t");
    Serial.print("elev: ");
    Serial.println(el/DEG_TO_RAD);
    Serial.print("roll: ");
    Serial.print(gir_x);
    Serial.print("\t");
    Serial.print("pitch: ");
    Serial.print(gir_y);
    Serial.print("\t");
    Serial.print("heading: ");
    Serial.println(gir_z);
    gir_x=gir_x*DEG_TO_RAD;
    gir_y=gir_y*DEG_TO_RAD;
    gir_z=gir_z*DEG_TO_RAD;

    //canvi de base esfèrica a base cartesiana
    float Sol_x=cos(el)*cos(az);
    float Sol_y=cos(el)*sin(az);
    float Sol_z=sin(el);

    Serial.print("Sol_global:");
    Serial.print(Sol_x);
    Serial.print(",");
    Serial.print(Sol_y);
    Serial.print(",");
    Serial.print(Sol_z);
    Serial.print(")");
    Serial.print("\t");
    float ang_solar_g=atan2(Sol_z,Sol_x);

```

```

Serial.print("ang_solar global: ");
Serial.println(ang_solar_g/DEG_TO_RAD);

float Sol_Lx;
float Sol_Ly;
float Sol_Lz;
Sol_Lx=Sol_x*(cos(gir_y)*cos(gir_z))+Sol_y*(cos(gir_y)*sin(gir_z))-Sol_z*sin(gir_y);
Sol_Ly=Sol_x*(cos(gir_z)*sin(gir_y)*sin(gir_x)-
sin(gir_z)*cos(gir_x))+Sol_y*(cos(gir_z)*cos(gir_x)+sin(gir_z)*sin(gir_y)*sin(gir_x))+Sol_z*(cos(gir_y)
*sin(gir_x));

Sol_Lz=Sol_x*(cos(gir_z)*sin(gir_y)*cos(gir_x)+sin(gir_z)*sin(gir_x))+Sol_y*(sin(gir_z)*sin(gir_y)*cos(
gir_x)-cos(gir_z)*sin(gir_x))+Sol_z*(cos(gir_y)*cos(gir_x));

Serial.print("Sol_local:");
Serial.print(Sol_Lx);
Serial.print(",");
Serial.print(Sol_Ly);
Serial.print(",");
Serial.print(Sol_Lz);
Serial.print(")");
Serial.print("\t");

if ((int) (Sol_Lx*100)==0)
{
  ang_solar=PI/2;
}
else
{
  ang_solar=atan2(Sol_Lz,Sol_Lx);
}
Serial.print("angle_solar local: ");Serial.println(ang_solar/DEG_TO_RAD);

return ang_solar;
}

// AUXILIARS I CALIBRAT

void writeang()

```

```

{
    eep.write(18,ang_lama_pers);
}
void girservo(int angle)
{
    if (angle>=0 && angle<10)
        {
            servo1.writeMicroseconds(900+((angle)*7));
        }
    if (angle>=10 && angle<20)
        {
            servo1.writeMicroseconds(970+((angle-10)*9));
        }
    if (angle>=20 && angle<30)
        {
            servo1.writeMicroseconds(1060+((angle-20)*8));
        }
    if (angle>=30 && angle<40)
        {
            servo1.writeMicroseconds(1140+((angle-30)*9));
        }
    if (angle>=40 && angle<50)
        {
            servo1.writeMicroseconds(1230+((angle-40)*8));
        }
    if (angle>=50 && angle<60)
        {
            servo1.writeMicroseconds(1310+((angle-50)*9));
        }
    if (angle>=60 && angle<70)
        {
            servo1.writeMicroseconds(1400+((angle-60)*11));
        }
    if (angle>=70 && angle<90)
        {
            servo1.writeMicroseconds(1510+((angle-70)*10));
        }
    if (angle>=90 && angle<100)

```

```

    {
        servo1.writeMicroseconds(1710+((angle-90)*12));
    }
    if (angle>=100 && angle<110)
    {
        servo1.writeMicroseconds(1830+((angle-100)*9));
    }
    if (angle>=110 && angle<120)
    {
        servo1.writeMicroseconds(1920+((angle-110)*10));
    }
    if (angle>=120 && angle<130)
    {
        servo1.writeMicroseconds(2020+((angle-120)*9));
    }
    if (angle>=130 && angle<140)
    {
        servo1.writeMicroseconds(2110+((angle-130)*10));
    }
    if (angle>=140 && angle<150)
    {
        servo1.writeMicroseconds(2210+((angle-140)*8));
    }
    if (angle>=150 && angle<160)
    {
        servo1.writeMicroseconds(2290+((angle-150)*9));
    }
    if (angle>=160 && angle<=170)
    {
        servo1.writeMicroseconds(2380+((angle-160)*8));
    }
    if (angle>=170 && angle<180)
    {
        servo1.writeMicroseconds(2460+((angle-170)*9));
    }
}

// DADES AL LCD
void lcdprograma()

```



```

{
  if (programa==0)
  {
    lcd.setCursor (0,1);
    lcd.print("Prog0:PERS ");
  }
  if (programa==1)
  {
    lcd.setCursor (0,1);
    lcd.print("Prog1:MAXSOL");
  }
  if (programa==2)
  {
    lcd.setCursor (0,1);
    lcd.print("Prog2:ESTATIC ");
  }
  if (programa==3)
  {
    lcd.setCursor (0,1);
    lcd.print("Prog3:TANCAT ");
  }
}
void lcdselecc(int selecc)
{
  if (selecc==1)
  {
    lcdangsolar();
    lcdanglama();
  }
  if (selecc==2)
  {
    lcdazimuth();
    lcdelev();
  }
  if (selecc==3)
  {
    lcdrellotge();
    lcddia();
  }
}

```

```

}
void lcdangsolar()
{
    posicionsolar();
    canvibase(roll,pitch,heading,azimuth,elev);
    lcd.setCursor (0,1);
    lcd.print("ASol");
    if ((ang_solar/DEG_TO_RAD)>=100)
    {
        lcd.setCursor (4,1);
        lcd.print(" ");
        lcd.setCursor (5,1);
        lcd.print((ang_solar/DEG_TO_RAD),2);
    }
    if (((ang_solar/DEG_TO_RAD)<100)&&((ang_solar/DEG_TO_RAD)>=10))
    {
        lcd.setCursor (4,1);
        lcd.print(" 0");
        lcd.setCursor (6,1);
        lcd.print((ang_solar/DEG_TO_RAD),2);
    }
    if (((ang_solar/DEG_TO_RAD)<10)&&((ang_solar/DEG_TO_RAD)>=0))
    {
        lcd.setCursor (4,1);
        lcd.print(" 00");
        lcd.setCursor (7,1);
        lcd.print((ang_solar/DEG_TO_RAD),2);
    }
    if (((ang_solar/DEG_TO_RAD)<0)&&((ang_solar/DEG_TO_RAD)>-10 ))
    {
        lcd.setCursor (4,1);
        lcd.print("-00");
        lcd.setCursor (7,1);
        lcd.print((abs ((ang_solar/DEG_TO_RAD)),2));
    }
    if (((ang_solar/DEG_TO_RAD)>-100)&&((ang_solar/DEG_TO_RAD)<=-10))
    {
        lcd.setCursor (4,1);
        lcd.print("-0");
    }
}

```

```

    lcd.setCursor (6,1);
    lcd.print((abs ((ang_solar/DEG_TO_RAD))),2);
}
if ((ang_solar/DEG_TO_RAD)<=-100)
{
    lcd.setCursor (4,1);
    lcd.print("-");
    lcd.setCursor (5,1);
    lcd.print((abs ((ang_solar/DEG_TO_RAD))),2);
}
}
void lcdanglama()
{
    lcd.setCursor (11,1);
    lcd.print("L");
    if ((ang_lama/DEG_TO_RAD)>=100)
    {
        lcd.setCursor (12,1);
        lcd.print(" ");
        lcd.setCursor (13,1);
        lcd.print((ang_lama/DEG_TO_RAD));
    }
    if (((ang_lama/DEG_TO_RAD)<100)&&((ang_lama/DEG_TO_RAD)>=10))
    {
        lcd.setCursor (12,1);
        lcd.print(" 0");
        lcd.setCursor (14,1);
        lcd.print((ang_lama/DEG_TO_RAD));
    }
    if (((ang_lama/DEG_TO_RAD)<10)&&((ang_lama/DEG_TO_RAD)>=0))
    {
        lcd.setCursor (12,1);
        lcd.print(" 00");
        lcd.setCursor (15,1);
        lcd.print((ang_lama/DEG_TO_RAD));
    }
    if (((ang_lama/DEG_TO_RAD)<0)&&((ang_lama/DEG_TO_RAD)>-10 ))
    {
        lcd.setCursor (12,1);

```

```

    lcd.print("-00");
    lcd.setCursor (15,1);
    lcd.print(abs ((ang_lama/DEG_TO_RAD)));
}
if (((ang_lama/DEG_TO_RAD)>-100)&&((ang_lama/DEG_TO_RAD)<=-10))
{
    lcd.setCursor (12,1);
    lcd.print("-0");
    lcd.setCursor (14,1);
    lcd.print(abs ((ang_lama/DEG_TO_RAD)));
}
if ((ang_lama/DEG_TO_RAD)<=-100)
{
    lcd.setCursor (12,1);
    lcd.print("-");
    lcd.setCursor (13,1);
    lcd.print(abs ((ang_lama/DEG_TO_RAD)));
}
}
void lcdazimuth()
{
    posiciosolar();
    azimuth=azimuth/DEG_TO_RAD;
    lcd.setCursor (0,1);
    lcd.print("Az");
    if (azimuth>=100)
    {
        lcd.setCursor (2,1);
        lcd.print(" ");
        lcd.setCursor (3,1);
        lcd.print(azimuth,2);
    }
    if ((azimuth<100)&&(azimuth>=10))
    {
        lcd.setCursor (2,1);
        lcd.print(" 0");
        lcd.setCursor (4,1);
        lcd.print(azimuth,2);
    }
}

```

```

if ((azimuth<10)&&(azimuth>=0))
{
  lcd.setCursor (2,1);
  lcd.print(" 00");
  lcd.setCursor (5,1);
  lcd.print(azimuth,2);
}
if ((azimuth<0)&&(azimuth>-10 ))
{
  lcd.setCursor (2,1);
  lcd.print("-00");
  lcd.setCursor (5,1);
  lcd.print((abs (azimuth)),2);
}
if ((azimuth>-100)&&(azimuth<=-10))
{
  lcd.setCursor (2,1);
  lcd.print("-0");
  lcd.setCursor (4,1);
  lcd.print((abs (azimuth)),2);
}
if (azimuth<=-100)
{
  lcd.setCursor (2,1);
  lcd.print("-");
  lcd.setCursor (3,1);
  lcd.print((abs (azimuth)),2);
}
}
void lcdelev()
{
  posiciosolar();
  elev=elev/DEG_TO_RAD;
  lcd.setCursor (8,1);
  lcd.print("E");
  if (elev>=100)
  {
    lcd.setCursor (9,1);
    lcd.print(" ");
  }
}

```

```

    lcd.setCursor (10,1);
    lcd.print(elev,2);
}
if ((elev<100)&&(elev>=10))
{
    lcd.setCursor (9,1);
    lcd.print(" 0");
    lcd.setCursor (11,1);
    lcd.print(elev,2);
}
if ((elev<10)&&(elev>=0))
{
    lcd.setCursor (9,1);
    lcd.print(" 00");
    lcd.setCursor (12,1);
    lcd.print(elev,2);
}
if ((elev<0)&&(elev>-10 ))
{
    lcd.setCursor (9,1);
    lcd.print("-00");
    lcd.setCursor (12,1);
    lcd.print((abs (elev)),2);
}
if ((elev>-100)&&(elev<=-10))
{
    lcd.setCursor (9,1);
    lcd.print("-0");
    lcd.setCursor (11,1);
    lcd.print((abs (elev)),2);
}
if (elev<=-100)
{
    lcd.setCursor (9,1);
    lcd.print("-");
    lcd.setCursor (10,1);
    lcd.print((abs (elev)),2);
}
}

```

```

void lcddia ()
{
    if (Clock.getDate()<10)
        {
            lcd.setCursor (8,1);
            lcd.print("0");
            lcd.setCursor (9,1);
            lcd.print(Clock.getDate());
        }
    else
        {
            lcd.setCursor (8,1);
            lcd.print(Clock.getDate());
        }
    lcd.setCursor (10,1);
    lcd.print("/");
    if (Clock.getMonth(Century)<10)
        {
            lcd.setCursor (11,1);
            lcd.print("0");
            lcd.setCursor (12,1);
            lcd.print(Clock.getMonth(Century));
        }
    else
        {
            lcd.setCursor (11,1);
            lcd.print(Clock.getMonth(Century));
        }
    lcd.setCursor (13,1);
    lcd.print("/");
    if (Clock.getYear()<10)
        {
            lcd.setCursor (14,1);
            lcd.print("0");
            lcd.setCursor (15,1);
            lcd.print(Clock.getYear());
        }
    else
        {

```

```

        lcd.setCursor (14,1);
        lcd.print(Clock.getYear());
    }
}
void lcdsolar()
{
    if (percent>=100)
    {
        lcd.setCursor (0,1);
        lcd.print(" ");
        lcd.setCursor (1,1);
        lcd.print(percent);
    }
    if ((percent<100)&&(percent>=10))
    {
        lcd.setCursor (0,1);
        lcd.print(" 0");
        lcd.setCursor (2,1);
        lcd.print(percent);
    }
    if ((percent<10)&&(percent>=0))
    {
        lcd.setCursor (0,1);
        lcd.print(" 00");
        lcd.setCursor (3,1);
        lcd.print(percent);
    }
    if ((percent<0)&&(percent>-10 ))
    {
        lcd.setCursor (0,1);
        lcd.print("-00");
        lcd.setCursor (3,1);
        lcd.print(abs (percent));
    }
    if ((percent>-100)&&(percent<=-10))
    {
        lcd.setCursor (0,1);
        lcd.print("-0");
        lcd.setCursor (2,1);

```



```

    lcd.print(abs (percent));
}
if (percent<=-100)
{
    lcd.setCursor (0,1);
    lcd.print("-");
    lcd.setCursor (1,1);
    lcd.print(abs (percent));
}
}
void lcdpers()
{
    if (ang_lama_pers>=100)
    {
        lcd.setCursor(0,1);
        lcd.print(ang_lama_pers);
    }
    if ((ang_lama_pers<100)&&(ang_lama_pers>=10))
    {
        lcd.setCursor(0,1);
        lcd.print("0");
        lcd.setCursor(1,1);
        lcd.print(ang_lama_pers);
    }
    if (ang_lama_pers<10)
    {
        lcd.setCursor(0,1);
        lcd.print("00");
        lcd.setCursor(2,1);
        lcd.print(ang_lama_pers);
    }
}
void lcdanglama(int anglama)
{
    if (anglama>=100)
    {
        lcd.setCursor(10,1);
        lcd.print(anglama);
    }
}

```

```

if ((anglama<100)&&(anglama>=10))
{
    lcd.setCursor(10,1);
    lcd.print("0");
    lcd.setCursor(11,1);
    lcd.print(anglama);
}
if (anglama<10)
{
    lcd.setCursor(10,1);
    lcd.print("00");
    lcd.setCursor(12,1);
    lcd.print(anglama);
}
}
void lcddata ()
{
    if (Clock.getDate(<10)
        {
            lcd.setCursor (0,1);
            lcd.print("0");
            lcd.setCursor (1,1);
            lcd.print(Clock.getDate());
        }
        else
        {
            lcd.setCursor (0,1);
            lcd.print(Clock.getDate());
        }
        lcd.setCursor (2,1);
        lcd.print("/");
        if (Clock.getMonth(Century)<10)
        {
            lcd.setCursor (3,1);
            lcd.print("0");
            lcd.setCursor (4,1);
            lcd.print(Clock.getMonth(Century));
        }
        else

```

```

    {
        lcd.setCursor (3,1);
        lcd.print(Clock.getMonth(Century));
    }
    lcd.setCursor (5,1);
    lcd.print("/");
    if (Clock.getYear()<10)
    {
        lcd.setCursor (6,1);
        lcd.print("0");
        lcd.setCursor (7,1);
        lcd.print(Clock.getYear());
    }
    else
    {
        lcd.setCursor (6,1);
        lcd.print(Clock.getYear());
    }
}
void lcdrellotge ()
{
    if (test==1)
    {
        if (hour<10)
        {
            lcd.setCursor (0,1);
            lcd.print("0");
            lcd.setCursor (1,1);
            lcd.print(hour);
        }
        else
        {
            lcd.setCursor (0,1);
            lcd.print(hour);
        }
        lcd.setCursor (2,1);
        lcd.print(":");
        if (minute<10)
        {

```

```

        lcd.setCursor (3,1);
        lcd.print("0");
        lcd.setCursor (4,1);
        lcd.print(minute);
    }
    else
    {
        lcd.setCursor (3,1);
        lcd.print(minute);
    }
    lcd.setCursor (5,1);
    lcd.print(":");
    if (second<10)
    {
        lcd.setCursor (6,1);
        lcd.print("0");
        lcd.setCursor (7,1);
        lcd.print(second);
    }
    else
    {
        lcd.setCursor (6,1);
        lcd.print(second);
    }
    lcd.setCursor (9,1);
    lcd.print("UTC");
}
if (test==0)
{
    if (Clock.getHour(h12,PM)<10)
    {
        lcd.setCursor (0,1);
        lcd.print("0");
        lcd.setCursor (1,1);
        lcd.print(Clock.getHour(h12,PM));
    }
    else
    {
        lcd.setCursor (0,1);

```

```

    lcd.print(Clock.getHour(h12,PM));
}
lcd.setCursor (2,1);
lcd.print(":");
if (Clock.getMinute(<10)
{
    lcd.setCursor (3,1);
    lcd.print("0");
    lcd.setCursor (4,1);
    lcd.print(Clock.getMinute());
}
else
{
    lcd.setCursor (3,1);
    lcd.print(Clock.getMinute());
}
lcd.setCursor (5,1);
lcd.print(":");
if (Clock.getSecond(<10)
{
    lcd.setCursor (6,1);
    lcd.print("0");
    lcd.setCursor (7,1);
    lcd.print(Clock.getSecond());
}
else
{
    lcd.setCursor (6,1);
    lcd.print(Clock.getSecond());
}
lcd.setCursor (9,1);
lcd.print("UTC");
}
}
void lcdlongitut()
{
    if (longitut>=100)
    {
        lcd.setCursor (0,1);

```

```

    lcd.print(" ");
    lcd.setCursor (1,1);
    lcd.print(longitut);
}
if ((longitut<100)&&(longitut>=10))
{
    lcd.setCursor (0,1);
    lcd.print(" 0");
    lcd.setCursor (2,1);
    lcd.print(longitut);
}
if ((longitut<10)&&(longitut>=0))
{
    lcd.setCursor (0,1);
    lcd.print(" 00");
    lcd.setCursor (3,1);
    lcd.print(longitut);
}
if ((longitut<0)&&(longitut>-10 ))
{
    lcd.setCursor (0,1);
    lcd.print("-00");
    lcd.setCursor (3,1);
    lcd.print(abs (longitut));
}
if ((longitut>-100)&&(longitut<=-10))
{
    lcd.setCursor (0,1);
    lcd.print("-0");
    lcd.setCursor (2,1);
    lcd.print(abs (longitut));
}
if (longitut<=-100)
{
    lcd.setCursor (0,1);
    lcd.print("-");
    lcd.setCursor (1,1);
    lcd.print(abs (longitut));
}

```

```

}
void lcdlatitut()
{
  if (latitut>=100)
  {
    lcd.setCursor (0,1);
    lcd.print(" ");
    lcd.setCursor (1,1);
    lcd.print(latitut);
  }
  if ((latitut<100)&&(latitut>=10))
  {
    lcd.setCursor (0,1);
    lcd.print(" 0");
    lcd.setCursor (2,1);
    lcd.print(latitut);
  }
  if ((latitut<10)&&(latitut>=0))
  {
    lcd.setCursor (0,1);
    lcd.print(" 00");
    lcd.setCursor (3,1);
    lcd.print(latitut);
  }
  if ((latitut<0)&&(latitut>-10 ))
  {
    lcd.setCursor (0,1);
    lcd.print("-00");
    lcd.setCursor (3,1);
    lcd.print(abs (latitut));
  }
  if ((latitut>-100)&&(latitut<=-10))
  {
    lcd.setCursor (0,1);
    lcd.print("-0");
    lcd.setCursor (2,1);
    lcd.print(abs (latitut));
  }
  if (latitut<=-100)

```

```

    {
        lcd.setCursor (0,1);
        lcd.print("-");
        lcd.setCursor (1,1);
        lcd.print(abs (latitut));
    }
}
void lcdheading()
{
    if (heading>=100)
        {
            lcd.setCursor (0,1);
            lcd.print(heading);
        }
    if ((heading<100)&&(heading>=10))
        {
            lcd.setCursor (0,1);
            lcd.print("0");
            lcd.setCursor (1,1);
            lcd.print(heading);
        }
    if (heading<10)
        {
            lcd.setCursor (0,1);
            lcd.print("00");
            lcd.setCursor (2,1);
            lcd.print(heading);
        }
}
void lcdpitch()
{
    if (pitch>=100)
        {
            lcd.setCursor (0,1);
            lcd.print(pitch);
        }
    if ((pitch<100)&&(pitch>=10))
        {
            lcd.setCursor (0,1);

```



```

        lcd.print("0");
        lcd.setCursor (1,1);
        lcd.print(pitch);
    }
    if (pitch<10)
    {
        lcd.setCursor (0,1);
        lcd.print("00");
        lcd.setCursor (2,1);
        lcd.print(pitch);
    }
}
void lcdroll()
{
    if (roll>=100)
    {
        lcd.setCursor (0,1);
        lcd.print(roll);
    }
    if ((roll<100)&&(roll>=10))
    {
        lcd.setCursor (0,1);
        lcd.print("0");
        lcd.setCursor (1,1);
        lcd.print(roll);
    }
    if (roll<10)
    {
        lcd.setCursor (0,1);
        lcd.print("00");
        lcd.setCursor (2,1);
        lcd.print(roll);
    }
}
void lcdtest()
{
    if (test==0)
    {
        lcd.setCursor (0,1);

```

```
    lcd.print("OFF ");
  }
  if (test==1)
  {
    lcd.setCursor (0,1);
    lcd.print("ON ");
  }
}
```

Diagrama de flux del programari

