

ISTITUTO DI ISTRUZIONE SUPERIORE ALBERT EINSTEIN
VIMERCATE

SISTEMA DI AUTOPUNTAMENTO PER UN TELESCOPIO ANALOGICO

CON L'UTILIZZO DI ARDUINO E STELLARIUM



**Tesina di
Leonardo Brambilla**

**Classe 5B
Indirizzo Elettronica e Elettrotecnica
Articolazione Automazione**

INDICE

INTRODUZIONE	1
L'ASTRONOMIA	2
BASI TEORICHE SULL'OSSERVAZIONE ASTRONOMICA	3
LA SFERA CELESTE	3
COORDINATE NEL CIELO	3
CARTA DEL CIELO BOREALE.....	5
FUNZIONAMENTO DI UNO STRUMENTO ASTRONOMICO.....	7
I TELESCOPI RIFLETTORI	7
IL TELESCOPIO NEWTON	8
MONTATURA EQUATORIALE.....	9
IL PUNTAMENTO AUTOMATICO "GOTO"	9
INTRODUZIONE AL PROGETTO E PROBLEMATICHE	10
DEFINIZIONE SPECIFICHE TECNICHE	11
COMPONENTI	12
TELESCOPIO SKY-WATCHER NEWTON 130/900 EXPLORER	12
STELLARIUM.....	13
CARATTERISTICHE	13
MICROCONTROLLORE	15
NUCLEO F401RE	16
RASPBERRY PI 3 Model B+	17
ARDUINO UNO	18
ARDUINO MEGA.....	18
ARDUINO NANO.....	19
SISTEMA DI COMUNICAZIONE (STELLARIUM/ARDUINO)	20
MOTORE.....	21
TEST PER CALCOLO DELLA POTENZA.....	21
SUPPORTO PER IL MOTORE	22
TRASMISSIONE A CINGHIA DENTATA.....	23
DRIVER.....	24
ENCODER.....	24
ALIMENTAZIONE	25
CONTROLLER PER L'INTERFACCIAMENTO CON L'UTENTE.....	26
PIANIFICAZIONE DEL LAVORO	27
REALIZZAZIONE DEL PROTOTIPO E DOCUMENTAZIONE DELLE FASI	28

1 RICERCA SOLUZIONI	28
1.1 HARDWARE	28
1.2 SOFTWARE	28
2 SIMULAZIONE DELLA SOLUZIONE E PROGRAMMAZIONE	29
2.1 PROGRAMMAZIONE ARDUINO MEGA (TELESCOPIO)	29
2.2 PROGRAMMAZIONE ARDUINO NANO (MOTORE)	34
2.3 PROGRAMMAZIONE ARDUINO UNO (CONTROLLER).....	35
3 PROGETTAZIONE SUPPORTO MOTORE.....	42
4 MONTAGGIO MOTORE	42
5 APPLICAZIONE DI PULEGGE E CINGHIE PER LA TRASMISSIONE.....	42
6 PROGETTAZIONE SCHEDA CONTROLLER E TELESCOPIO	43
7 TEST FUNZIONAMENTO ED EVENTUALI CORREZIONI.....	44
8 PRIMA OSSERVAZIONE.....	45
CONCLUSIONE.....	46
USER'S MANUAL	47
GENERAL INFORMATION	48
SYSTEM OVERVIEW	48
SYSTEM REQUIREMENTS.....	48
GETTING STARED.....	48
1. BALANCING	48
2. CALIBRATION.....	49
3. ELETTRICAL CONNECTION	49
4. STELLARIUM CONFIGURATION	49
5. USING THE SYSTEM	49
MAINTENANCE AND WARNINGS	50
BIBLIOGRAFIA	51
SITOGRAFIA	52

INTRODUZIONE

Il progetto, che ho deciso di realizzare, unisce due mie grandi passioni: l'astronomia e l'automazione. Sin da piccolo ho avuto molto interesse per tutto quello che riguarda l'universo e le sue particolarità, soprattutto lo studio dei pianeti e dei fenomeni astronomici che, da sempre, mi hanno affascinato e tutt'ora mi spingono ad un continuo aggiornamento sulle grandi scoperte pubblicate su riviste o su siti specializzati. La propensione, invece, per l'elettronica e per l'automazione è maturata nel corso del periodo scolastico e mi ha portato a creare vari progetti nei quali ho potuto esprimere ed evolvere le mie capacità tecniche e personali.

L'idea del sistema di autopuntamento per un telescopio analogico è nata dall'esigenza di automatizzare un telescopio con montatura equatoriale, regalatomi anni fa, e dotato di un auto-inseguitore, che permetteva (dopo aver puntato il telescopio sul corpo celeste scelto) di impostare una velocità fissa per mantenere l'oggetto osservato all'interno dell'oculare. Questo dispositivo, però, non rispondeva alla richiesta di poter creare un sistema di auto-puntamento e di "tracking" stellare, inoltre presentava un problema meno rilevante ma comunque importante riguardo all'alimentazione: utilizzava infatti delle batterie, di conseguenza il motore richiedeva, dopo poco tempo, la sostituzione di queste, causando una spesa eccessiva.

Ho voluto quindi applicare tutte le mie capacità in campo elettronico e informatico cercando di ideare una soluzione che riassumesse questi cinque anni di scuola nel miglior modo possibile, e che mettesse in atto tutte quelle esperienze che mi hanno formato durante il mio percorso scolastico e durante il periodo di alternanza scuola-lavoro. Oltre alle mie capacità tecniche, mi sono stati infatti di grande aiuto gli insegnamenti di carattere gestionale e progettuale derivati dall'esperienza lavorativa che mi hanno permesso di organizzare il lavoro in modo tale da elaborare varie soluzioni e scegliere quella che mi permetteva di realizzare in modo completo il progetto e di apportare eventuali modifiche aggiuntive in futuro. Per concludere, ho voluto inserire degli approfondimenti di fisica, di meccanica dei telescopi e di osservazione stellare, per spiegare i concetti chiave che si celano dietro questa incredibile scienza che è l'astronomia.

L'ASTRONOMIA

“La più sublime, la più nobile tra le fisiche scienze è senza dubbio l’Astronomia. L’uomo s’innalza per mezzo di essa al di sopra di se medesimo e giunge a conoscere la causa dei fenomeni più straordinari...”.

Giacomo Leopardi

L’astronomia è la scienza che studia i corpi celesti e i fenomeni ad essi relativi attraverso l’osservazione del cielo stellato. Già nell’antichità l’uomo era interessato a questi eventi, infatti, furono trovate su un osso diverse incisioni, fatte dagli uomini di Cro-Magnon, risalenti a circa 30.000 anni fa, che appaiono come possibili registrazioni delle fasi lunari. Se così fosse, questi calendari sarebbero i documenti astronomici più antichi mai scoperti, anticipando la scoperta della scrittura di circa 25.000 anni. Col tempo gli antichi utilizzarono i loro studi astronomici per l’agricoltura, la quale richiedeva una buona conoscenza delle stagioni, per i loro rituali religiosi e per le loro predizioni sul futuro, basate sulle posizioni dei corpi celesti. Il calcolo del tempo, quindi, è diventato sempre più accurato e le persone hanno imparato a determinare in anticipo i movimenti delle stelle. Durante il rapido sviluppo della navigazione tra il XVII e il XVIII secolo, l’astronomia ebbe un compito molto più importante, in quanto soluzione al problema di determinare la posizione durante i viaggi che si estendevano sempre più lontani dai porti di origine. Alle basi di questi sviluppi vi sono le leggi che governano il movimento dei pianeti scoperte da Copernico, Tycho Brahe, Keplero, Galilei e Newton. La ricerca astronomica ha cambiato la nostra visione del mondo, dalle concezioni geometriche, antropocentriche (dal greco, anthropos, "essere umano", e kentron, "centro", teorie per la quale l’uomo è al centro dell’universo) alla visione moderna di un vasto universo in cui la specie umana e la Terra ricoprono un ruolo insignificante. L’astronomia moderna è una scienza fondamentale, motivata principalmente dalla curiosità e dal desiderio di conoscere meglio ciò che ci circonda. Ha, inoltre, un ruolo centrale nel formare una visione scientifica del mondo, ovvero una visione basata sull’osservazione, su teorie accuratamente testate e su un ragionamento logico, creando così un modello di studio che non è influenzato o limitato da alcuna concezione o credenza filosofica, politica o religiosa. Infine, l’astronomia è una delle poche scienze in cui il lavoro di ricerca di un dilettante e di un amatore (l’astrofilo) può giocare un ruolo rilevante, fornendo dati sulle stelle variabili o scoprendo comete, novae, supernovae, asteroidi o altri oggetti dello spazio incontaminato.

BASI TEORICHE SULL'OSSERVAZIONE ASTRONOMICA

LA SFERA CELESTE

I nostri occhi ci permettono di avere una visione "stereoscopica" del mondo che ci circonda; ciò significa che possiamo percepire la profondità degli oggetti che osserviamo, fino a una certa distanza. Questa capacità, però, viene meno quando si cerca di guardare molto lontano (per esempio un chilometro) e quello che si presenta è un panorama piatto.

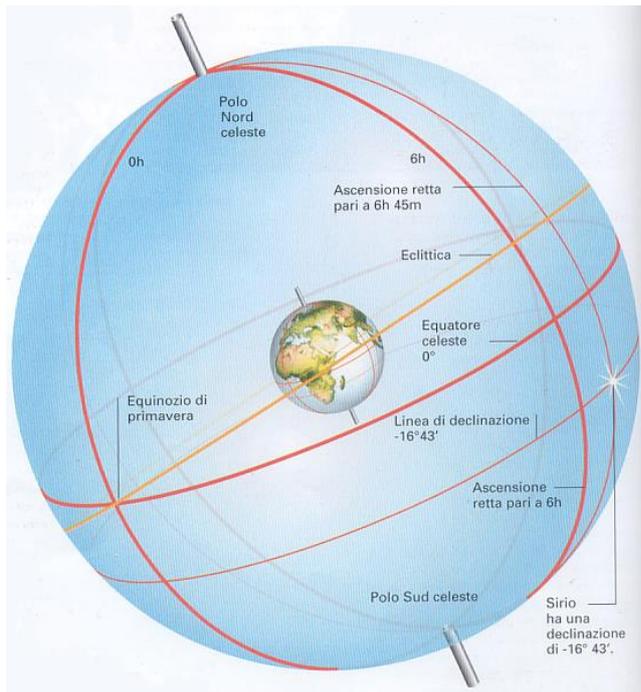
In cielo questa mancanza di profondità è esagerata: oggetti molto lontani tra loro nello spazio, come la Luna, una stella o la galassia di Andromeda, ci appaiono tutti alla stessa distanza. Questo giustifica il concetto di sfera celeste; noi possiamo trattare il cielo, per le osservazioni che vogliamo compiere, come se le distanze degli oggetti celesti da noi fossero tutte uguali.

COORDINATE NEL CIELO

Uno dei primi problemi da affrontare quando si osserva il cielo è quello di orientarsi in mezzo alle stelle. Un metodo rudimentale è quello di riunirle in gruppi chiamati costellazioni e servirsi poi di queste come punti di riferimento per

prendere dimestichezza con le distanze tra i corpi celesti. Esiste, però, un sistema più preciso per individuare gli oggetti in cielo: le *coordinate celesti*. Analogicamente alla Terra, dove è possibile individuare un qualsiasi punto della superficie tramite la latitudine e la longitudine, nell'astro si possono introdurre delle coordinate che ricalcano quelle terrestri.

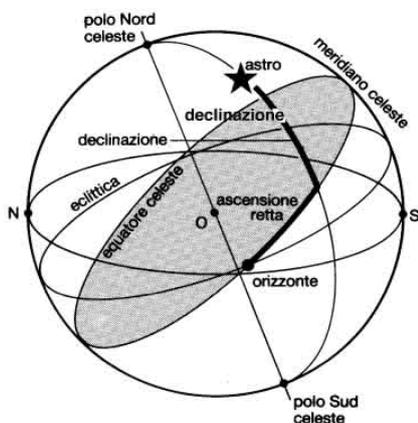
Vengono chiamati *Polo Nord* e *Polo Sud* i punti dati dalla proiezione dell'asse della Terra sulla sfera celeste: essi sono collegati proprio sulla verticale del Polo Nord e del Polo Sud terrestri. In modo analogo, l'equatore celeste è la linea che si trova esattamente sopra all'equatore terrestre. Possiamo così parlare di "latitudine" e "longitudine" in cielo, le quali prendono il nome rispettivamente di "declinazione" e "ascensione retta". Come sulla Terra, la declinazione si misura in



1. Localizzare una stella

La figura mostra come si può localizzare Sirio, la stella più luminosa del cielo, con le linee di ascensione retta e declinazione. Sirio ha una declinazione di $-16^{\circ}43'$ e un'ascensione retta pari a 6h 45m.

gradi, preceduti da un segno + o -, a seconda che ci si trovi a nord o a sud dell'equatore celeste, mentre l'ascensione retta si misura in ore, minuti e secondi di tempo, poiché un giorno è composto da 24 ore e in un angolo giro vi sono 360°, a ogni ora corrisponde un angolo di 15°.

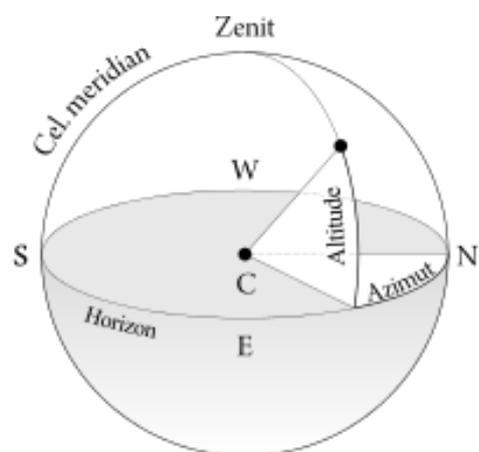


2. Raffigurazione delle coordinate equatoriali

La costruzione dei paralleli celesti, quindi, risulta relativamente facile, mentre non vale altrettanto per quanto riguarda i meridiani. Infatti, si potrebbe pensare che il "meridiano zero" corrisponda a quello terrestre (ovvero, quello passante per l'Osservatorio di Greenwich a Londra), ma in realtà non è così. Il moto di rivoluzione della Terra porterebbe questo meridiano, come gli altri, da una costellazione all'altra nel corso della giornata. Il sistema di coordinate internazionale, per risolvere il problema, utilizza un punto di riferimento di partenza che sia "fisso" il più a lungo possibile. Per trovarlo, è utile considerare il

moto di rivoluzione terrestre: esso porta il Sole a mutare la sua posizione in cielo lungo una linea chiamata "eclittica" (dal fatto che le eclissi avvengono quando la Luna si trova, piena o nuova, su questa linea); parte del suo cammino celeste avviene sopra l'equatore celeste e parte sotto. Le due intersezioni fra l'eclittica e l'equatore celeste determinano i punti dei due equinozi; il *punto vernale*, che coincide con l'*equinozio di primavera* per l'emisfero settentrionale, che è il punto di riferimento, dal quale si comincia a contare l'ascensione retta, verso est. Quindi, il meridiano celeste che passa per questo punto prende il nome di *coluro equinoziale*.

Vi è un secondo metodo per individuare un astro in cielo, il quale sfrutta le cosiddette *coordinate altazimutali*. Per esempio, per localizzare una stella, bisogna segnare la sua proiezione sull'orizzonte e misurare l'angolo compreso tra questa e il punto cardinale sud: tale misura fornisce "l'*azimut*" della stella; misurando in gradi la distanza tra il corpo celeste e l'orizzonte avremo la seconda coordinata, detta "*altezza*". Questo sistema ha il vantaggio di riferirsi a elementi fissi, come l'orizzonte e il sud, ma ha il difetto di attribuire a ogni astro delle coordinate che variano in ogni istante a causa della rotazione terrestre.



3. Raffigurazione delle coordinate orizzontali, dette anche altazimutali

CARTA DEL CIELO BOREALE

La mappa in figura mostra il complesso delle costellazioni dell'emisfero boreale, cioè settentrionale. Il suo studio è fondamentale per l'osservazione, in quanto rappresenta le stelle e le costellazioni visibili durante l'anno, al di sopra dell'equatore.

Ovviamente, trattandosi della proiezione di una semisfera su di un cerchio, essa è particolarmente precisa. In realtà, la parte di cielo visibile è solo la metà, la quale non è sempre la stessa perché dipende da tre fattori:

- il primo è la *durata della notte* nel punto in cui ci si trova; infatti, con il trascorrere delle ore, la Terra ruota attorno al proprio asse da ovest verso est e la volta stellata sembra ruotare in direzione contraria, da oriente verso occidente. A causa di questo fenomeno le stelle e gli altri oggetti del cielo sorgono a oriente, raggiungono la massima altezza sull'orizzonte quando transitano sopra al punto cardinale sud (per un osservatore boreale) e tramontano ad occidente
- il secondo fattore che determina la porzione di cielo visibile è la *data dell'anno*. A causa del moto di rivoluzione della Terra, il Sole compie nel corso dell'anno un cammino apparente tra le costellazioni, sulla linea dell'eclittica. La porzione in cui si trova il Sole in una certa data non è visibile in quel determinato periodo, poiché le stelle che ne fanno parte non sarebbero osservabili mentre vi è la luce solare. Da un giorno all'altro la sezione di cielo visibile non cambierà di molto, infatti bisognerà aspettare qualche mese per poter osservare costellazioni, dopo che il Sole si sarà spostato sulla linea dell'eclittica. È per questo che ha senso parlare di costellazioni "*di stagione*", le quali sono posizionate in modo opposto alla nostra stella
- il terzo parametro è la *latitudine geografica* dell'osservatore (a parità di ora e data). Vi sono, infatti, delle costellazioni dell'emisfero australe che non saranno mai visibili, per esempio, dall'Italia.

Sulla mappa sono segnalati i gradi sul meridiano orizzontale e le ore sul bordo per calcolare le coordinate celesti. I mesi dell'anno identificano la parte meglio visibile alla mezzanotte locale. Infine, lungo l'eclittica sono segnalate le posizioni apparenti del Sole riferite all'inizio di ogni mese.

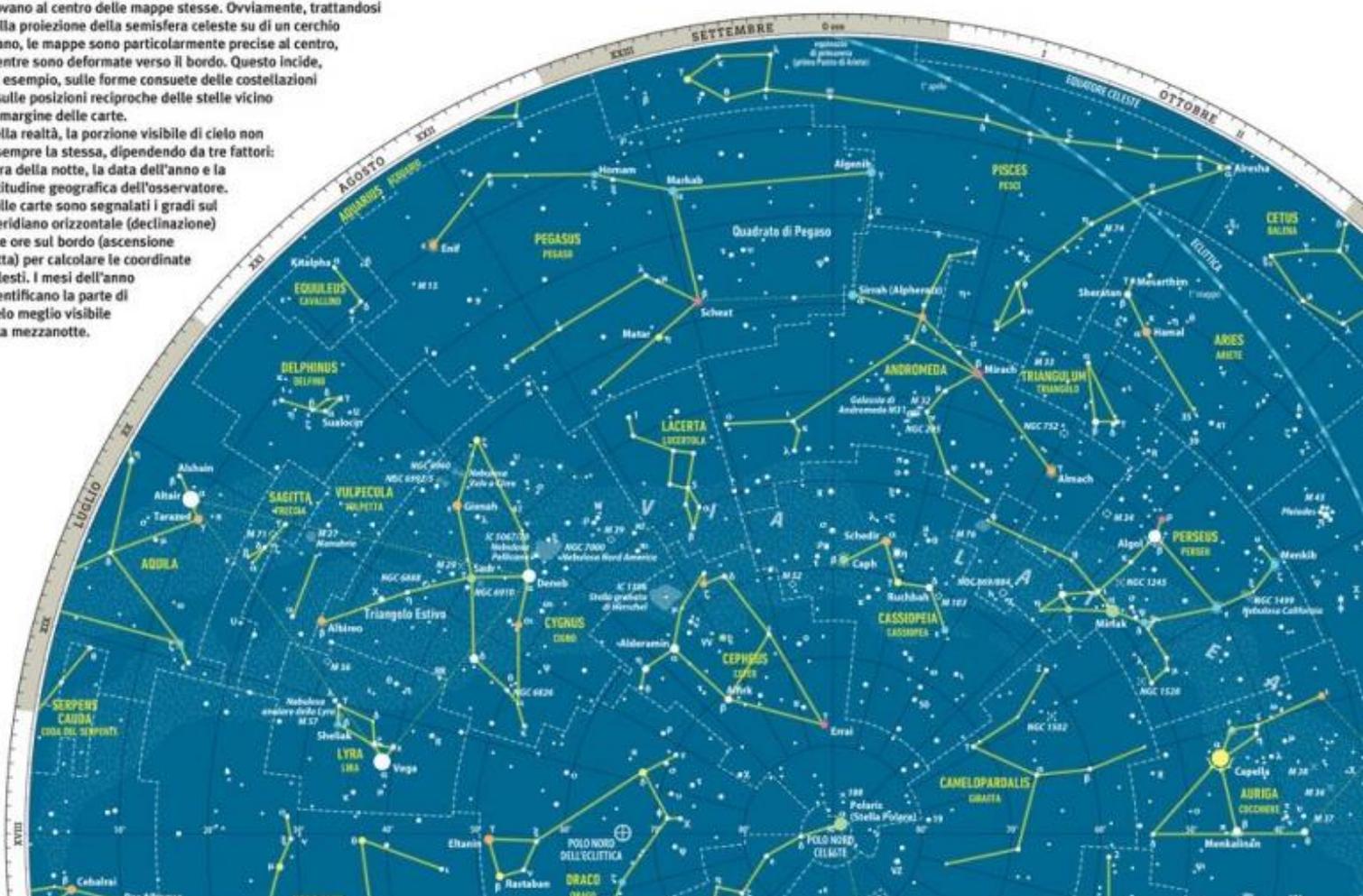
La mappa va letta tenendo presente che il Polo Nord celeste è al centro della mappa stessa, all'incirca nella posizione dove si trova la stella *Polare*. Nell'emisfero nord è opportuno saper,

quindi, individuare la posizione dell'astro, al centro della mappa. Essa rimane sempre fissa e ciò è dovuto al fatto che essa si trova quasi esattamente sul prolungamento ideale dell'asse di rotazione della terra. Quindi, per quanto quest'ultima ruoti, la stella non si sposterà, risultando come un riferimento costante a qualunque ora e in qualsiasi notte dell'anno. Le stelle e le costellazioni che si trovano in prossimità della Polare rimangono, quindi, sempre visibili: si tratta delle cosiddette stelle o costellazioni "circumpolari". La loro posizione varia, ma non tramonteranno e non sorgeranno mai. Questo è molto importante ai fini del progetto perché è un parametro che deve essere considerato per gestire al meglio il telescopio: siccome il programma per il calcolo del percorso ricerca la strada minore per arrivare all'oggetto selezionato, per quanto riguarda gli oggetti circumpolari, esso dovrà, inoltre, considerare il passaggio per la stella polare per poter raggiungere oggetti, per esempio, opposti in AR (ascensione retta).

Le carte principali mostrano l'aspetto del cielo boreale e del cielo australe, tenendo presente che Polo Nord e Polo Sud celesti si trovano al centro delle mappe stesse. Ovviamente, trattandosi della proiezione della semisfera celeste su di un cerchio piano, le mappe sono particolarmente precise al centro, mentre sono deformate verso il bordo. Questo incide, ad esempio, sulle forme consuete delle costellazioni e sulle posizioni reciproche delle stelle vicino ai margini delle carte.

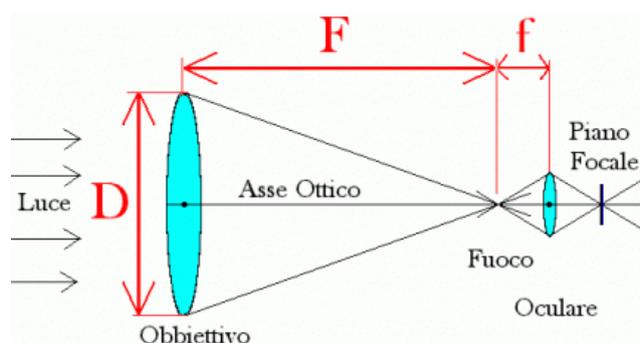
Nella realtà, la porzione visibile di cielo non è sempre la stessa, dipendendo da tre fattori: l'ora della notte, la data dell'anno e la latitudine geografica dell'osservatore. Sulle carte sono segnalati i gradi sul meridiano orizzontale (declinazione) e le ore sul bordo (ascensione retta) per calcolare le coordinate celesti. I mesi dell'anno identificano la parte di cielo meglio visibile alla mezzanotte.

CIELO BOREALE



FUNZIONAMENTO DI UNO STRUMENTO ASTRONOMICICO

I corpi celesti sono, per lo più, deboli ed estremamente lontani. Lo strumento astronomico permette di raccogliere la luce e ingrandire l'immagine. Esso, quindi, lavora in due direzioni differenti contemporaneamente: un obiettivo a lente nel telescopio rifrattore o uno specchio nel riflettore raccolgono la luce e la concentrano per formare l'immagine.



4. Cammino dei raggi luminosi in un telescopio rifrattore

Quest'ultima è a sua volta ingrandita da un *oculare*, che non è altro che una piccola e potente lente. Più è grande l'obiettivo, più aumenta la sua capacità di raccogliere la luminosità emessa dagli astri (quindi, esso incide sul dettaglio dell'immagine); è per questo motivo che molti strumenti professionali misurano diversi metri di diametro, permettendo di osservare galassie lontane.

L'obiettivo del telescopio è un sistema convergente, che focalizza, ad una certa distanza (*lunghezza focale*), l'immagine sul piano chiamato *fuoco*. La *focale* determina direttamente la dimensione della figura catturata nel fuoco. Alcuni strumenti professionali richiedono sistemi ottici che piegano il cammino della luce con l'aiuto di uno o due specchi intermedi (per la troppa distanza focale). La loro *lunghezza meccanica* quindi non supera più di tre volte il loro diametro, mentre la *lunghezza ottica* può, in certe configurazioni, superare di 20 volte il suo diametro.

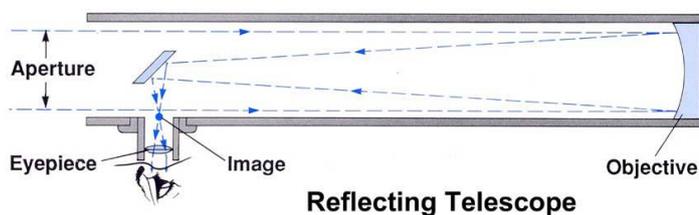
La potenza totale di uno strumento, in termini di luminosità e ricchezza, è quindi interamente determinata dal suo diametro. Inoltre, qualsiasi telescopio capovolge le immagini, contrariamente a cannocchiali terrestri e binocoli. Questo inconveniente è del tutto trascurabile in astronomia, ma esistono, comunque, accessori ottici denominati "*raddrizzatori terrestri*" che possono correggere il problema.

I TELESCOPI RIFLETTORI

In un telescopio *riflettore*, l'obiettivo è sostituito da uno specchio concavo, coperto in superficie da una pellicola d'alluminio spessa circa 1/10 di micron (1 micron = 1 millesimo di millimetro). Lo specchio è disposto al fondo di un tubo la cui estremità aperta è diretta verso l'astro da osservare. Il telescopio è quindi un riflettore per la luce. La convergenza dell'immagine nel fuoco è assicurata solamente dalla curvatura della superficie dello specchio e, di conseguenza, il telescopio non soffre

dell'*aberrazione cromatica* dei rifrattori (è un difetto nella formazione dell'immagine, dovuta al diverso valore di rifrazione delle diverse lunghezze d'onda che compongono la luce che passa attraverso il mezzo ottico, e si traduce in immagini che presentano ai bordi dei soggetti aloni colorati).

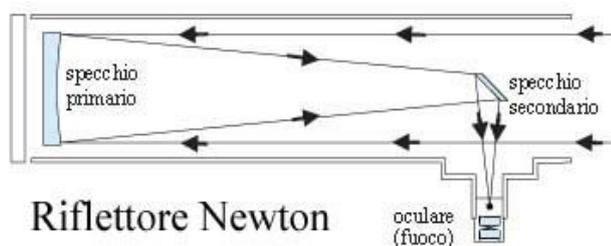
L'immagine formata dallo specchio è in seguito riflessa da un piccolo specchietto (che non altera la figura) verso il porta oculare. Questo, però, provoca una copertura dell'obbiettivo principale dalla luce che entra nel tubo: si parla quindi di "*ostruzione centrale*".



5. Funzionamento di un telescopio riflettore

IL TELESCOPIO NEWTON

Questo strumento per l'osservazione stellare viene anche chiamato *Riflettore Newtoniano*. La maggior parte dei telescopi sono di questa tipologia, la quale prende il nome dal suo inventore, Isaac Newton. L'obbiettivo è costituito da uno specchio parabolico (*specchio primario*) che riflette e focalizza il fascio ottico in ingresso su uno specchietto ellittico posto su un supporto a 45° (*specchio secondario*) e da questo riflesso verso l'oculare attraverso un'apertura praticata sul fianco del tubo principale.



6. Funzionamento del telescopio newton

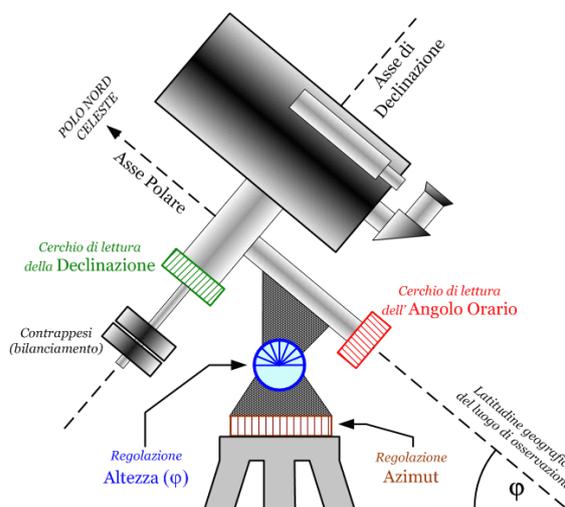
Il vantaggio principale è la sua totale *acromaticità*, in quanto la riflessione, al contrario della rifrazione, porta allo stesso punto di fuoco tutti i colori dello spettro visibile, per contro la lavorazione della superficie degli specchi va effettuata ad un livello superiore a quello delle lenti se si vogliono ottenere buoni

risultati. Inoltre la presenza sul cammino ottico della *ostruzione*, provocata dallo specchio secondario e dal suo supporto, provoca un'interferenza nel cammino della luce, che si traduce in un contrasto lievemente inferiore a quello degli strumenti a lenti, fermo restando il potere risolutivo, connesso al diametro. I pregi del riflettore Newton sono quindi *l'ottima resa e l'incisività delle immagini* anche ad ingrandimenti elevati al centro del campo osservato. Un problema degli strumenti di questo tipo (facilmente superabile con un minimo di pratica) è invece la necessità di

una buona *collimazione* delle ottiche per ottenere prestazioni elevate. Nella collimazione lo specchio primario e quello secondario sono perfettamente in asse tra loro ed in squadra col foceggiatore (dispositivo di messa a fuoco, posto sull'oculare).

MONTATURA EQUATORIALE

Uno strumento astronomico può essere installato su una montatura costruita da due assi, di cui uno è perpendicolare al primo. Quest'ultimo è inclinato in modo tale che, rispetto al piano orizzontale, sia sollevato di un angolo uguale alla latitudine del luogo di osservazione, in direzione del Polo Nord celeste (nell'emisfero boreale). Viene quindi utilizzata l'ascensione retta per inquadrare tutti gli astri compresi tra ovest ed est. Il secondo asse



7. Principio della montatura equatoriale

detto "*asse di declinazione*", inoltre, permette di spostarsi liberamente su qualunque parte del cielo situata tra l'orizzonte nord e l'orizzonte sud, passando per lo zenit (è il punto immaginario che sta esattamente sopra la testa dell'osservatore). Così facendo, si ottiene una montatura equatoriale.

IL PUNTAMENTO AUTOMATICO "GOTO"

Alcune montature equatoriali di gamma medio-alta sono equipaggiate, all'origine o come optional, di un dispositivo di puntamento automatico chiamato *goto* ("andare verso", in inglese). Al suo interno vi è un piccolo computer che custodisce i nomi e le coordinate di molte stelle, galassie, ammassi e nebulose. Grazie alle sue capacità di calcolo, il goto è anche capace di trovare pianeti o comete di cui si conoscono solo i parametri orbitali. Questa soluzione è estremamente utile per gli oggetti difficili da reperire come alcuni asteroidi o piccole nebulose.



8. Telescopio con incorporato un sistema di puntamento "Goto"

INTRODUZIONE AL PROGETTO E PROBLEMATICHE

Il progetto si pone di creare un sistema che sia simile alla strumentazione del goto, in modo che l'utente possa sfruttare l'autopuntamento per una ricerca stellare semplificata (rispetto ad un utilizzo analogico della strumentazione). L'uso dell'elettronica in Astronomia, però, agli occhi di un appassionato, appare come una barriera tra l'appassionato e l'oggetto della sua passione: l'utilizzo di strumenti automatizzati per l'osservazione porta infatti alla perdita di una delle caratteristiche principali di questa disciplina, ovvero, il metodo di osservazione. Per questo motivo, prima di affrontare il progetto, vi è stato uno studio della montatura e della sfera celeste per poter apprendere la dinamica con la quale il telescopio debba muoversi e le peculiarità del cielo affinché si possano visualizzare le bellezze dello spazio profondo apprezzando appieno ciò che si è imparato e visto.

Durante la lavorazione vi è stato un inconveniente che ha portato alla modifica del progetto, in quanto l'asse di controllo delle coordinate DEC è meccanicamente impossibilitato ad ospitare un motore di controllo, siccome non dispone di una gestione a 360° dell'asse. Per questo motivo sarebbe inutile motorizzare questa movimentazione, quindi si è scelto di creare un controller per la comunicazione con l'utente, il quale sarà informato di quanti gradi muovere manualmente lo strumento. Per quanto riguarda l'asse di Declinazione, questo è stato, comunque, automatizzato sia per la realizzazione di un auto-puntamento sia per il mantenimento dell'oggetto nell'oculare, una volta posizionato il telescopio.

Inoltre, lo studio teorico ha portato alla luce la grande accuratezza richiesta per questo progetto, soprattutto per quanto riguarda la tipologia di strumentazione e la precisione che deve essere applicata nel puntamento: essendo, il telescopio, un dispositivo costruito in modo molto accurato, non si è voluto modificarne la struttura per evitare eventuali danni all'oggetto; quindi si è scelto di rendere il più esatto possibile il sistema di controllo, per minimizzare l'errore. Sommariamente si è deciso di puntare sui seguenti punti:

- massima precisione dei motori
- massimo controllo del sistema
- diminuzione dell'errore presente nel sistema
- memorizzazione costante e precisa della posizione del telescopio.

Questa valutazione ha permesso di definire le specifiche del sistema in modo ordinato.

DEFINIZIONE SPECIFICHE TECNICHE

Caratteristiche d'insieme del progetto:

- controllo automatico dell'asse di Ascensione Retta del telescopio
- sistema gestito da un microcontrollore
- software per le coordinate stellari che permette la localizzazione dell'oggetto celeste
- controller per la gestione del telescopio e dell'asse di Declinazione
- motore ad alta precisione e potenza per diminuire l'errore nel movimento del telescopio
- alimentazione diretta, senza uso di batterie, con la sola tensione di rete 230V AC
- memorizzazione continua della posizione
- riconoscimento della posizione tramite encoder di alta precisione e velocità
- utilizzo di un Driver (scheda di controllo per il motore passo passo) per il micro-stepping
- supporti 3D per i motori che permettano il loro posizionamento senza creare intralcio alla struttura del telescopio
- possibilità di implementazioni future, grazie ad un controllo flessibile (utilizzo di un microcontrollore con una considerevole quantità di porte I/O).

Struttura:

La struttura del sistema è stata sviluppata cercando di rispondere all'estrema precisione richiesta. Sono stati valutati la qualità costruttiva, le caratteristiche tecniche e il costo dei componenti, per creare una soluzione efficace ed efficiente, che abbia un budget finanziario inferiore ad una montatura motorizzata in commercio.

COMPONENTI

TELESCOPIO SKY-WATCHER NEWTON 130/900 EXPLORER

Il telescopio utilizzato è uno *Skywatcher N 130/900 Explorer* posto su una montatura EQ-2. Quest'ultima è di tipo equatoriale ed è ottima per allineare precisamente l'ottica alla stella polare, anche se non dispone della possibilità di utilizzare un *cannocchiale polare* (un dispositivo che permette di avere maggiore precisione nel puntamento). La calibrazione permette, con movimenti in Ascensione Retta e Declinazione, di puntare e seguire con precisione gli oggetti. Infine, lo strumento è bilanciato attraverso un apposito contrappeso.

Questo telescopio è stato catalogato nella gamma medio-bassa, ma, grazie al suo grande diametro, permette una grande raccolta di luce indispensabile per la visione di oggetti al di là del sistema solare.



9. Telescopio n 130/900 explorer skywatcher

OTTICA	
Tipo	Riflettore
Tecnica di fabbricazione	Newton
Apertura (mm)	130
Lunghezza focale (mm)	900
Potere di risoluzione	0,88
Capacità di raccolta della luce	350
Max. ingrandimento utile	260
FOCHEGGIATORE	
Tecnica di fabbricazione	Cremagliera
Connessione (lato oculare)	1,25"
MONTATURA	
Tecnica di fabbricazione	EQ-2
Tipo di montatura	equatoriale
CAMPI DI UTILIZZO	
Luna e pianeti	si
Nebulose e Galassie	si
Osservazione della natura	non consigliato
Astrofotografia	non consigliato

STELLARIUM

Stellarium è un planetario gratuito Open Source per il computer. Mostra un cielo realistico in 3D proprio come si vedrebbe a occhio nudo con un binocolo o un telescopio.

È stato scelto questo programma per la sua accuratezza nei dettagli e per la sua precisione, in tempo reale, della posizione dei corpi celesti. Inoltre, essendo un programma aperto allo sviluppo (vi è una Community che ne cura l'aggiornamento), ha presente al suo interno un plug-in che permette la connessione con un telescopio tramite le porte COM. Questo collegamento ha lo scopo di visualizzare la posizione del telescopio.



10. Logo Stellarium

CARATTERISTICHE

CIELO

- catalogo predefinito di oltre 600.000 stelle
- cataloghi extra con oltre 177 milioni di stelle
- catalogo predefinito di oltre 80.000 oggetti del cielo profondo
- catalogo extra con oltre 1 milione di oggetti del cielo profondo
- asterismi e illustrazioni delle costellazioni
- costellazioni per oltre 20 culture diverse
- immagini degli oggetti del cielo profondo
- Via Lattea realistica
- atmosfera altrettanto realistica con alba e tramonto
- pianeti e loro satelliti

INTERFACCIA

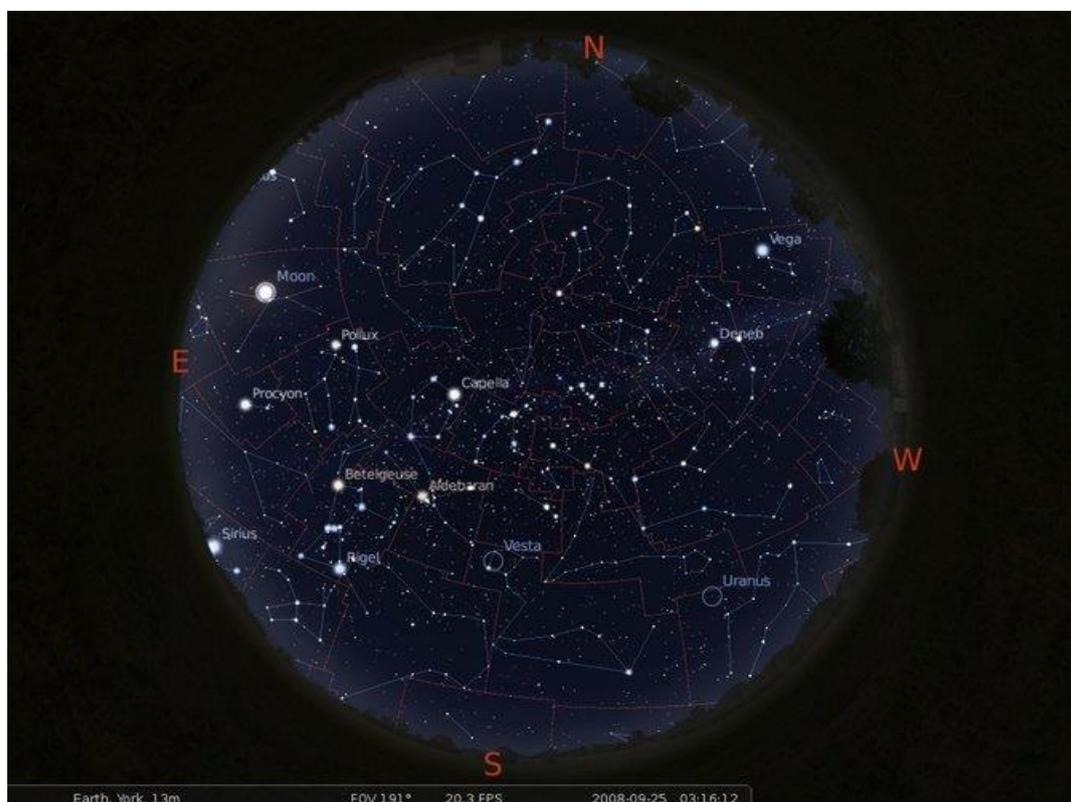
- zoom potente
- controllo del tempo
- interfaccia multilingua
- proiezione fisheye (a occhio di pesce) per le cupole dei planetari
- proiezione a specchio sferico per cupole non professionali
- interfaccia grafica e completo controllo da tastiera
- controllo del telescopio

VISUALIZZAZIONE

- coordinate equatoriali e altazimutali
- scintillazione delle stelle
- stelle cadenti
- code delle comete
- simulazione dei brillamenti Iridium
- simulazione delle eclissi
- simulazione di nove e supernove
- scenari 3D
- paesaggi sostituibili, ora con proiezione sferica

PERSONALIZZAZIONE

- sistema di plug-in per aggiungere satelliti artificiali, per simulare oculari, per configurare telescopi e molto altro
- capacità di aggiungere nuovi oggetti del sistema solare da risorse online...
- possibilità di aggiungere i vostri oggetti del cielo profondo, paesaggi, immagini di costellazioni, script...



11. Acquisizione della schermata di Stellarium; visualizzazione del cielo stellato

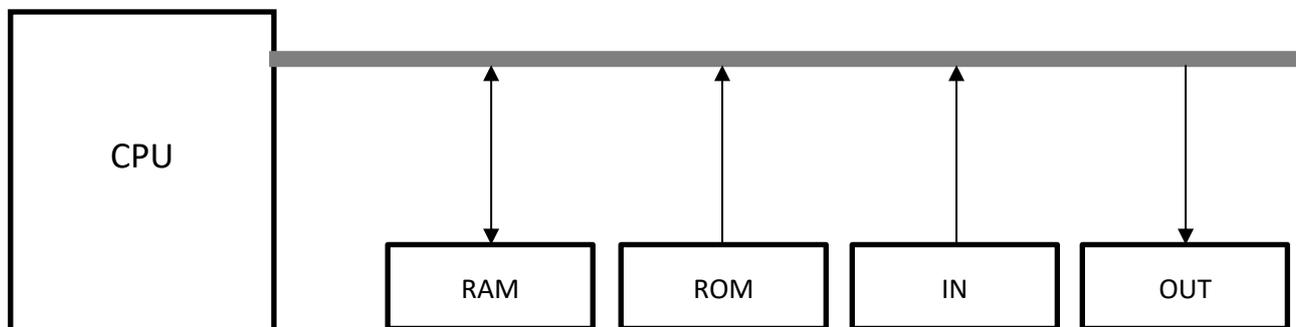
MICROCONTROLLORE

Il microcontrollore è un dispositivo elettronico a larga scala d'integrazione, che include funzionalità sia hardware che software, le quali permettono di interfacciare la logica di controllo con circuiti e dispositivi esterni per la creazione di un sistema automatico.

Per quanto riguarda l'hardware, è dotato di circuiti organizzati per funzionalità interne e per la comunicazione con dispositivi esterni. Invece, dal punto di vista software è dotato di capacità di calcolo logico-aritmetiche (superiori rispetto ad un cervello umano), che permettono di leggere ed eseguire un programma. L'utilizzo di questo sistema di controllo permette, quindi, la gestione di una struttura complessa sia in ambito elettronico sia in ambito informatico.

La ricerca del microcontrollore si è basata sulle specifiche tecniche del progetto, scegliendo così il dispositivo che avesse proprietà conformi a quanto richiesto, le quali possono essere suddivise nei principali elementi, comuni a tutti i microcontrollori:

- *unità centrale di elaborazione CPU*
- *sezione di memoria (RAM, ROM ecc.)*
- *sezioni di input-output*



12. Elementi principali dei microcontrollori

L'unità centrale di elaborazione *Central Processing Unit (CPU)* è il "cervello" del microcontrollore poiché

- interpreta ed esegue il programma
- gestisce l'unità aritmetico-logica *Arithmetic Logic Unit (ALU)* per l'esecuzione delle istruzioni di programma
- attiva e *sincronizza* i dispositivi che cooperano all'esecuzione del programma

Le sezioni di memoria sono destinate al mantenimento del programma, che la CPU ha in carico, e dei dati sui quali la CPU opera. In particolare vengono distinti:

- memoria ROM, nella quale vengono memorizzati i dati strutturali dell'elaboratore
- memoria RAM, nella quale vengono memorizzati il programma, che la CPU deve eseguire, e i dati sui quali deve operare

Infine, le sezioni di input-output fungono da interfaccia tra la CPU e le unità esterne al microcontrollore. In particolare:

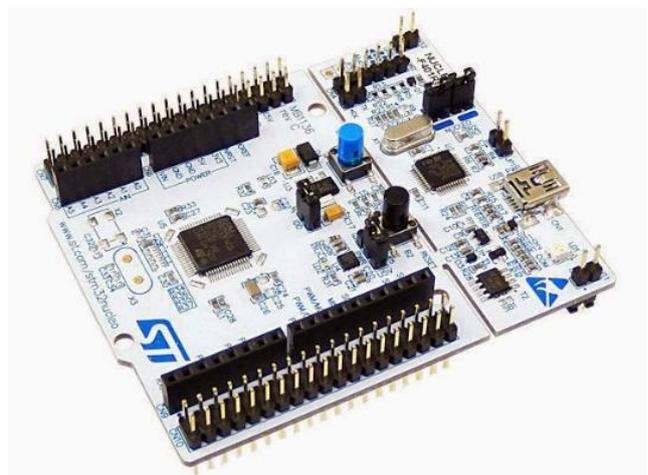
- porte di INPUT utilizzate per ricevere, ad esempio, segnali da controllare o per leggere lo stato di sensori, interruttori
- porte di OUTPUT utilizzate per controllare dispositivi esterni come relè o led

Per il progetto, quindi, è stato molto importante valutare la potenza e la portata di ogni singolo componente del microcontrollore: dalla capienza della memoria RAM alla quantità di porte disponibili.

Di seguito vengono riportate le soluzioni valutate per la scelta del microcontrollore:

NUCLEO F401RE

La Nucleo F401RE è una board sviluppata da STMicroelectronics che implementa il controllore STM32F401RE, che con 512KB di Flash Memory risulta particolarmente flessibile dal punto di vista della customizzazione grazie ai pin morpho con cui è possibile accedere a tutti i pin del microcontrollore a 32 bit con la possibilità di apportare modifiche senza



13. Scheda Nucleo F401RE

complicazioni. In aggiunta vi è la possibilità di alimentazione mediante USB o attraverso i power-pin fino a 12V.

La Nucleo Board dispone di pin meccanicamente collegabili con quelli delle schede Arduino, anche se non sempre le shield di Arduino sono elettronicamente compatibili.

La programmazione avviene attraverso l'IDE On-Line MBED, a livello hardware invece mediante l'STLink, quasi una scheda a sé stante con installato il controllore STM32F103, che si occupa della sola programmazione della board.

Per il progetto, la Nucleo risulta una scheda ancora in via di sviluppo che necessita di un ambiente di programmazione più semplificato: le librerie sono di difficile reperimento o non sono ancora state realizzate, infatti il problema principale è la mancanza di una grande community che permetta

un'evoluzione continua dell'IDE. Se si cercano informazioni in Internet (per eventuali problematiche), ci si può accorgere di come vi siano pochi elementi in rete da cui poter trarre spunto, per cui con le poche informazioni reperibili, il lavoro di gestione di eventuali implementazioni al progetto si sarebbe complicato maggiormente, comportando un dispendio maggiore di tempo e un innalzamento dei costi. Di conseguenza la scheda è stata scartata, in quanto difficile da programmare e priva di un adeguato supporto.

RASPBERRY PI 3 Model B+

La scheda rappresenta l'aggiornamento della famosa Dev board della medesima casa costruttrice, il Raspberry PI 3. Questo sviluppo garantisce un innalzamento prestazionale e una connettività di rete decisamente superiore rispetto al suo predecessore. E' equipaggiata con un SoC Broadcom BCM2837B0 da 1,4 GHz, un chip che, oltre a garantire migliori performance grazie all'aumento della frequenza di clock (PI 3 si ferma a 1,2 GHz), promette una migliore efficienza e gestione termica.



14. Scheda Raspberry PI 3 model B+

Questa scheda può essere considerata un computer in miniatura, infatti ha un sistema operativo derivante da Linux che permette di utilizzare la scheda come se fosse un PC (naturalmente con potenze limitate). Sostanzialmente unisce la potenza di un microprocessore a quella di un microcontrollore. Questa sua particolarità è stata particolarmente valutata per la possibilità di utilizzarla sia come computer, sia come scheda di controllo: si è pensato, quindi, di installare Stellarium su di essa e di sfruttare la stessa scheda per gestire il telescopio.

L'idea è stata, quasi subito, accantonata a causa dei requisiti del programma per la visualizzazione delle stelle, il quale richiedeva delle proprietà grafiche troppo elevate per questa scheda. Si è allora valutata l'opzione di utilizzare la scheda solamente come microcontrollore, il che è risultato più complicato rispetto all'utilizzo di una scheda Arduino, soprattutto per quanto riguarda la semplicità di programmazione, e più dispendioso in termini economici.

ARDUINO UNO

Il microcontrollore è l'ATmega328 prodotto da Atmel, ha una velocità di 16MHz, una memoria flash da 32KB, una RAM da 2KB e una memoria EEPROM da 1KB. L'alimentazione della scheda avviene tramite porta usb o tramite apposito connettore. In caso siano collegati sia il cavo usb sia il connettore di alimentazione, la scheda è capace di scegliere automaticamente la fonte di alimentazione esterna.

Possiede 14 pin digitali programmabili come ingressi o uscite (i quali hanno anche la capacità di essere utilizzati per funzioni dedicate come la generazione di segnale PWM o la comunicazione UART) e 6 ingressi per l'acquisizione ed elaborazione di segnali analogici.

Si è scelto di utilizzare questo microcontrollore per gestire il dispositivo di controllo del telescopio (controller), infatti la sua compattezza e la sua potenza di calcolo sono quanto richiesto per sviluppare il componente studiato.



15. Scheda Arduino Uno



16. Scheda Arduino Mega 2560

ARDUINO MEGA

Arduino Mega 2560 Rev3 è una scheda elettronica basata sul microcontrollore Atmega2560. Dispone di 54 ingressi/uscite digitali (14 delle quali possono essere utilizzate come uscite PWM), 16 ingressi analogici, 4 porte seriali hardware UART, un oscillatore a 16 MHz, una connessione USB, un jack di alimentazione, un header ICSP e un pulsante di reset. Esso contiene tutto il necessario per supportare il microcontrollore; è sufficiente connettere Arduino Mega ad un computer tramite un cavo USB o alimentarlo con un trasformatore AC/DC o una batteria per iniziare.

La sua grande quantità di porte seriali e di pin permette di gestire più dispositivi in contemporanea. Questa sua caratteristica è essenziale per il progetto, infatti è stato scelto questo microcontrollore per gestire il telescopio e la comunicazione con Stellarium, il controller e l'Arduino che controlla dell'Encoder.

ARDUINO NANO

Arduino Nano è una scheda piccola, completa e adatta alla breadboard basata su ATmega328P (Arduino Nano 3.x). Ha più o meno la stessa funzionalità dell'Arduino Duemilanove, ma in un pacchetto diverso. Manca solo un jack di alimentazione CC e funziona con un cavo USB Mini-B invece di uno standard. Arduino Nano può essere alimentato, inoltre, esternamente con un'alimentazione non regolata da 6-20 V (pin 30) o con un alimentatore esterno regolato a 5 V (pin 27). La fonte di alimentazione viene automaticamente selezionata sulla sorgente di tensione più alta. L'ATmega328P ha 32 KB (anche con 2 KB usati per il bootloader), l'ATmega328P ha 2 KB di SRAM e 1 KB di EEPROM. Ciascuno dei 14 piedini digitali funzionano a 5 volt. Ogni pin può fornire o ricevere un massimo di 40 mA e ha una resistenza di pull-up interna (scollegata di default) di 20-50 kOhm. Per le sue dimensioni ridotte, questo microcontrollore è l'ideale per affiancare l'Arduino Mega che è stato disposto per il controllo del telescopio, perché vi era la necessità di dividere la gestione del motore da quella degli encoder.



17. Scheda Arduino Nano

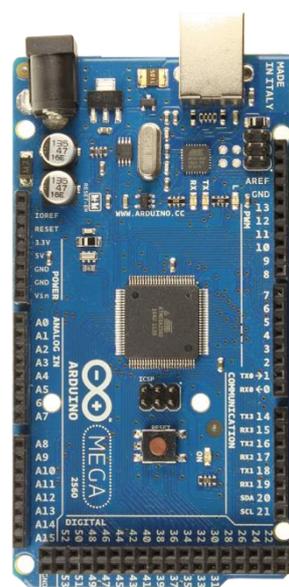
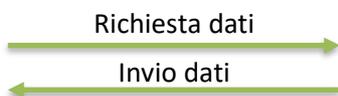
SISTEMA DI COMUNICAZIONE (STELLARIUM/ARDUINO)

Il sistema di comunicazione tra Arduino e Stellarium adotta un protocollo già esistente, essendo quest'ultimo già stato implementato nel planetario in questione. In questo caso si è utilizzato il protocollo LX200 della Meade.

Appena connesso Arduino a Stellarium, quest'ultimo chiede al telescopio (Arduino) le coordinate a cui sta puntando in quel momento. Questa operazione avviene inviando prima il comando #:GR# del protocollo LX200 con il quale chiede la coordinata di Ascensione Retta. La risposta che Stellarium si aspetta è del tipo: HH:MM.T# oppure HH:MM:SS#. I valori delle componenti della coordinata di AR sono inviati singolarmente. Una volta ottenuti dei valori che Stellarium reputa validi, chiede successivamente la coordinata di DEC inviando il comando #:GD#. La risposta che si aspetta è del tipo sDD*MM# oppure sDD*MM'SS# dove "s" è il segno "+" o "-".

Dopo aver ricevuto quest'ultima coordinata, Stellarium, per rimanere aggiornato sulla situazione, richiederà nuovamente la coordinata di AR e poi ancora quella di DEC in un ciclo continuo.

Dopo qualche istante per la connessione apparirà il mirino arancione del telescopio nel punto individuato dalle coordinate. Cliccando su di esso appariranno in alto a sinistra le coordinate inviate da Arduino. Infine, per puntare una stella, bisognerà selezionare il corpo celeste scelto e con il comando "CTRL 1" Stellarium invierà le coordinate del Target al telescopio.



18. Comunicazione tra Stellarium e Arduino

MOTORE

La scelta del motore da utilizzare per lo spostamento del telescopio si è basata principalmente su queste caratteristiche:

- precisione
- velocità
- potenza motrice (coppia)
- dimensioni.

È quindi stata scelta la tipologia di motore denominata “passo passo”, per la grande precisione che esercita nella sua rotazione, ma soprattutto per l’elevata coppia che fornisce anche a grandi velocità. Infatti, I motori passo-passo (detti anche step motor o anche stepper) sono caratterizzati nel panorama dei motori elettrici da una serie di particolarità che ne fanno la scelta ideale per tutte quelle applicazioni che richiedono precisione nello spostamento angolare e nella velocità di rotazione con la massima semplicità d'uso, quali la robotica e i servomeccanismi in genere.

TEST PER CALCOLO DELLA POTENZA

Per definire le specifiche tecniche del motore si è scelto un approccio sperimentale, effettuando prove di movimentazione con un motore stepper già in possesso, il quale dispone della possibilità di variare la corrente acquisita e di variare la potenza di utilizzo.

Il risultato di questa sperimentazione ha portato ad acquistare un motore con le seguenti caratteristiche: un motore a doppio albero con un albero posteriore lungo 13,5 mm e un diametro dell'albero di 8 mm; i fori filettati sono disponibili sul cappuccio posteriore che può essere utilizzato per i freni, encoder per volantini.

- Tipo di motore: Bipolar Stepper
- Angolo di passo: 1.8 deg
- Coppia di tenuta: 1.26Nm(178.4oz.in)
- Corrente/fase nominale: 2.8A
- Tensione nominale: 2.5V
- Resistenza di fase: 0.9ohms
- Induttanza: 2.5mH \pm 20%(1KHz)
- Dimensione telaio: 57 x 57mm
- Altezza del motore: 56mm
- Diametro dell'asse: Φ 8mm

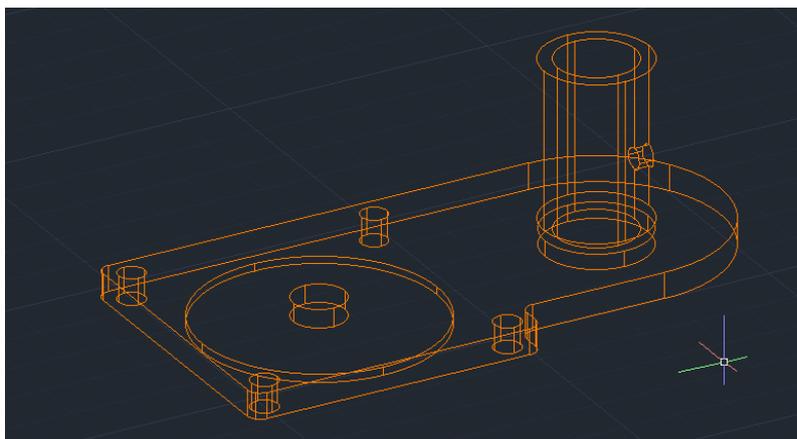
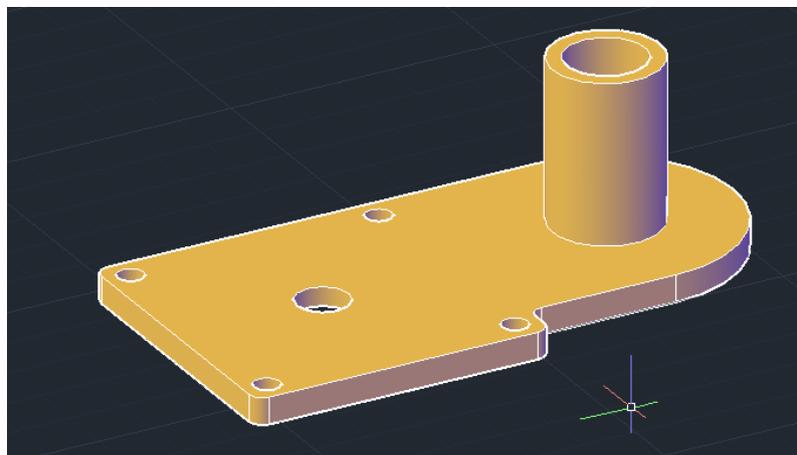


19. Motore Passo Passo utilizzato

SUPPORTO PER IL MOTORE

Per quanto riguarda il montaggio del motore si è scelto di creare un supporto in 3D che permetta un posizionamento saldo ed efficace, siccome l'elevata precisione richiesta non ammette oscillazioni.

Per il disegno è stato utilizzato il software AutoCAD della Autodesk, il cui utilizzo è stato appreso a scuola. Le linee del supporto sono state create seguendo le caratteristiche fisiche del telescopio, cercando di creare un oggetto ergonomico e non ingombrante. Infine, per la stampa è stata utilizzata la stampante 3D che è a disposizione a scuola, una Sharebot 42: stampante 3D professionale di quarta generazione che utilizza la tecnologia FFF (Fused Filament Fabrication).



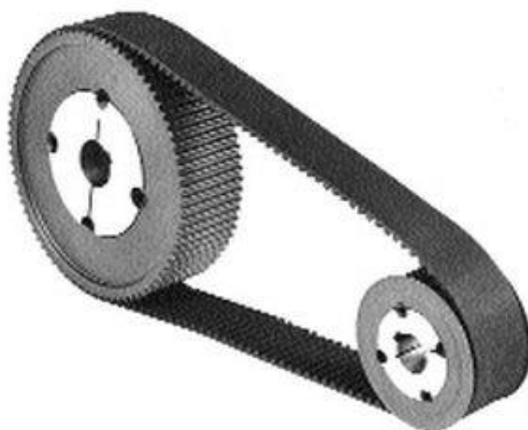
20. Disegno 3D del supporto per il motore

TRASMISSIONE A CINGHIA DENTATA

Si è scelto di utilizzare una trasmissione a cinghia dentata per collegare il motore all'asse perché essa unisce i vantaggi della catena e degli ingranaggi a quelli della cinghia, ma senza le limitazioni normalmente associate a questi tipi convenzionali di trasmissione. Non esiste allungamento, nessun contatto di metallo con metallo, nessun bisogno di lubrificazione. Le installazioni di cinghie dentate sono eccezionalmente versatili, con potenze trasmesse fino a 450 KW e per velocità variabili da meno di 0,5 m/s ad oltre 50 m/s, inoltre rendono possibili importanti risparmi in peso, spazio e costruzione, senza alcun sacrificio di rendimento. Esse sono progettate e costruite con cura estrema, con passi, profondità del dente, larghezza e altre quote estremamente precise; i materiali impiegati nella fabbricazione di queste cinghie consistono in componenti con particolare resistenza a trazione. I denti stampati sulla cinghia sono progettati per accoppiarsi esattamente con le cave praticate sulla puleggia. Questi denti entrano ed abbandonano le scanalature con dolcezza e con attrito trascurabile; la loro funzione è simile a quella dei denti di un ingranaggio.

I VANTAGGI OFFERTI DALLE CINGHIE DENTATE

- Assenza di manutenzione
- Ampia gamma di profili e sviluppi
- Elevato rendimento
- Soluzione economica.



21. Trasmissione a cinghia dentata

DRIVER



22. Driver microstep TB6600

Il driver stepper supporta il controllo della velocità e della direzione. È possibile impostare il micro passo e la corrente di uscita con 6 DIP switch. Esistono in tutto 7 tipi di micro passi (1, 2 / A, 2 / B, 4, 8, 16, 32) e 8 tipi di controllo corrente (0,5 A, 1 A, 1,5 A, 2 A, 2,5 A, 2,8 A, 3,0 A, 3.5A). Tutti i terminali di segnale adottano l'isolamento da fotoaccoppiatore ad alta velocità, migliorando la sua capacità di interferenza ad alta frequenza.

Come dispositivo professionale, è in grado di pilotare motori passo-passo ibridi a due fasi.

Il driver TB6600 è un driver per motore passo-passo professionale e di facile utilizzo che può controllare un motore passo-passo a due fasi. È compatibile con Arduino e altri microcontrollori che possono emettere un segnale digitale a impulsi da 5 V. Il driver del motore passo-passo Arduino TB6600 ha una vasta gamma di alimentazione, 9 ~ 42VDC, ed è in grado di emettere corrente di picco 4A, sufficiente per la maggior parte dei motori passo-passo.

Il driver stepper supporta il controllo della velocità e

ENCODER

La scelta degli encoder è molto complicata: essendo componenti essenziali per il progetto le loro caratteristiche avrebbero condizionato il progetto drasticamente, quindi vi è stata un'attenta valutazione e la scelta finale è ricaduta sugli AMT103 della CUI.

Gli encoder CUI AMT sono modulari estremamente precisi e robusti, adatti per l'uso in ambienti industriali. Il loro design li rende resistenti a sporco, polvere e olio. Altre caratteristiche sono una gamma di temperature operativa da -40°C a 125°C, un semplice assemblaggio e un basso assorbimento di corrente. Inoltre, gli Encoder AMT di CUI sono ideali per sistemi automatizzati.

La serie CUI AMT10 utilizzata offre un basso consumo energetico e utilizza la tecnologia ASIC capacitiva brevettata e le uscite di tensione CMOS.



23. Encoder AMT103-V

CARATTERISTICHE

- tecnologia ASIC capacitiva
- basso consumo energetico
- uscite CMOS
- 16 risoluzioni selezionabili tramite DIP switch
- indice di impulso
- design del pacchetto modulare
- versioni diritte (radiali) e ad angolo retto (assiali)
- 9 opzioni di fori di montaggio per versione radiale
- 8 opzioni di fori di montaggio per versione assiale
- intervallo di temperatura operativa da -40°C a 100°C

ALIMENTAZIONE

Per l'alimentazione si è utilizzato un alimentatore switching da 24V con un'erogazione di corrente massima di 15A (secondo le richieste del sistema).

Specifiche:

- materiale della custodia: alluminio
- tensione in ingresso: 230V / 110V AC
- tensione in uscita: 24V DC
- corrente in uscita: max. 15°
- temperatura di esercizio: 0 - 40 °C
- protezione da cortocircuito, protezione da surriscaldamento, protezione contro le sovratensioni



24. Alimentatore 24V 15A

CONTROLLER PER L'INTERFACCIAMENTO CON L'UTENTE

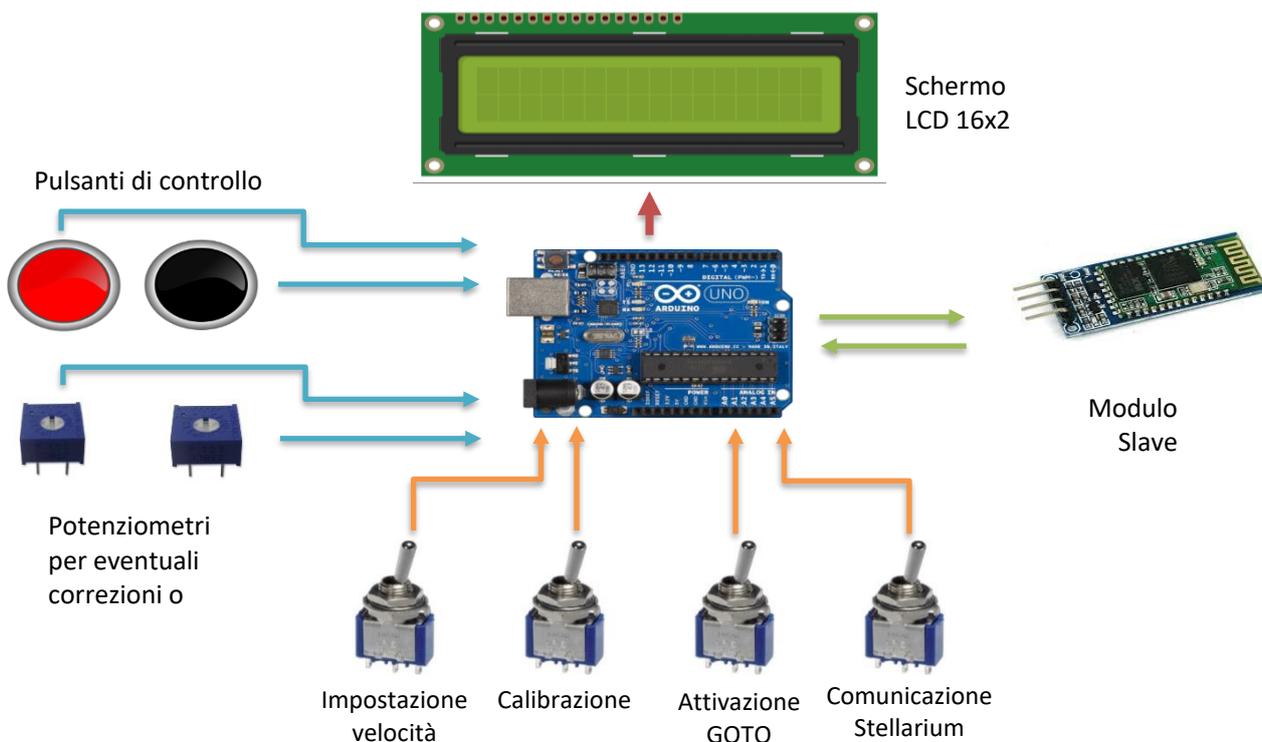
Il controller è stato sviluppato per poter controllare le funzioni principali del progetto:

- attivazione/disattivazione comunicazione con Stellarium
- attivazione/disattivazione Goto
- calibrazione
- impostazione velocità
- visualizzazione coordinate del telescopio
- visualizzazione coordinate di target
- comunicazione della movimentazione della strumentazione.

È stato utilizzata una comunicazione Bluetooth dato che vi è una gestione del sistema molto ravvicinata e la velocità di trasferimento dati è molto elevata.

Inoltre, è stata studiato un posizionamento dei componenti ergonomico e semplice agli occhi dell'utente per facilitarli ulteriormente il controllo del telescopio.

SCHEDA E COMPONENTI:



PIANIFICAZIONE DEL LAVORO



REALIZZAZIONE DEL PROTOTIPO E DOCUMENTAZIONE DELLE FASI

1 RICERCA SOLUZIONI

Il progetto è molto ambizioso e complesso: questo è dovuto alla complicata gestione del programma e all'estrema precisione richiesta. L'osservazione stellare, infatti, non è un'attività semplice e facilmente accessibile perché, per poter visualizzare per esempio i pianeti o le nebulose, servono sofisticate strumentazioni, le quali hanno costi elevati. Se aggiungiamo la complessità meccanica che, con le sole conoscenze di elettronica e di automazione, ha rappresentato un ostacolo, è evidente che si sono dovuti raggiungere dei compromessi per la poca esperienza nel campo, in particolare invece di ricostruire la base strutturale del telescopio per ricreare una montatura Go To (adatta a questo progetto), si è voluto utilizzare il corpo del telescopio in uso che, comunque, permette una buona visualizzazione di alcuni corpi celesti di grandi dimensioni (come la Luna, Marte, Saturno).

Inizialmente sono state valutate molte soluzioni per questo progetto perché il problema fondamentale è stato, appunto, ricercare un metodo per sostenere i motori su una struttura semplice come la EQ-2. Inoltre, la comunicazione tra Arduino e Stellarium aveva molte possibilità di realizzazione (alcune più semplici, altre più complesse). Quindi, come metodo di approccio al progetto, si è scelto di testare le varie idee, per controllarne l'efficacia. Questo ha portato all'esclusione di alcuni progetti che richiedevano molti materiali e troppi fondi. Valutando attentamente si è giunti alle seguenti soluzioni:

1.1 HARDWARE

Il corpo del telescopio non dispone di alloggiamenti predisposti per entrambi i motori e non permette un controllo a 360° sull'asse di Declinazione. Come unica disposizione del costruttore della montatura, vi è solo il supporto per un motore sull'asse AR (creato appositamente per controllare automaticamente la velocità della stella tramite l'auto-inseguitore, di cui si è parlato nell'introduzione). Ricostruire l'intera struttura avrebbero permesso di creare una montatura di tipologia altazimutale, la quale è più facile da gestire con un microcontrollore, ma per mancanza di tempo e denaro non si è voluto procedere con questa soluzione. Si è optato quindi di automatizzare l'asse dove è permessa la motorizzazione e di creare un controller per l'utente per poter controllare manualmente il secondo asse.

1.2 SOFTWARE

La ricerca di una soluzione software ha richiesto più tempo: l'approccio che Stellarium utilizza per collegarsi ad un telescopio ha una serie di scelte, le quali sono state sottoposte a una valutazione approfondita per il prosieguo del progetto. Infatti si cercava un protocollo o un'applicazione che avesse gestito l'interazione tra il sistema informatico e quello meccanico. Inizialmente si è provato ad utilizzare un'applicazione esterna a Stellarium, la quale doveva configurare automaticamente la comunicazione con l'Arduino del telescopio, ma si è poi rilevato che il programma utilizzato accettava solo telescopi commerciali.

La soluzione finale, quindi, è stata utilizzare una comunicazione seriale con l'utilizzo di un protocollo di comunicazione di un telescopio commerciale già inserito all'interno di Stellarium. Così facendo non si è dovuto implementare un nuovo sistema di trasmissione dati, mentre si è riusciti ad utilizzare un protocollo completo, le cui caratteristiche sono facilmente reperibili in Internet.

2 SIMULAZIONE DELLA SOLUZIONE E PROGRAMMAZIONE

La programmazione di Arduino, essendo molto semplice e intuitiva, permette l'impostazione di valori fittizi per poter simulare la soluzione senza utilizzare nessun dispositivo hardware, se non il microcontrollore stesso. Questo ha permesso di sviluppare in modo completo i programmi per i microcontrollori citati di seguito, calcolando teoricamente la risoluzione ottenuta e le riduzioni meccaniche e software create. Il lavoro è stato successivamente finalizzato con le prove sperimentali della strumentazione.

2.1 PROGRAMMAZIONE ARDUINO MEGA (TELESCOPIO)

```
//Librerie
#include <stdlib.h>
#include <stdio.h>
#include <Encoder.h>
#include <avr/wdt.h>

//Posizione stella polare
#define ARPolar 9110
#define DECPolar 321251

//Dichiarazione encoder e variabili
Encoder myEnc(2, 3);
long oldPosition = -999;
int EST = 5;
int OVEST = 4;
int CONTROL = 6;
int BTSTATE = 23;
int V = 0, S = 0, G = 0;
boolean stringComplete = false;
int b = 0;
int a = 0;
char input[20];
char inputBT[20];
char txAR[10];
char txDEC[11];
volatile long float ARtel, DECtel, oldAR;
long ARtarget, DECtarget;
long ARdist, ARdiff, DECdiff;
long arHH, arMM, arSS;
char SIGNtel;
long decDEG, decMM, decSS;
long memoDEC;
long latitude;
long betaN, deltaN, betaS, deltaS;
int A = 0;
boolean circumpolar, enablegoto;
char c = ' ';
int flag = 0;

//Impostazioni iniziali: Posizione, Seriali, Pin, WDT, Timer1
void setup() {
```

```

ARtarget = ARTel = ARPolar;
DECtarget = DECTel = DECPolar;
latitude = 164215;
circumpolar = false;
enablegogo = false;
Serial.begin(9600);
Serial1.begin(9600);
pinMode(EST, OUTPUT);
pinMode(OVEST, OUTPUT);
pinMode(CONTROL, OUTPUT);
pinMode(BTSTATE, INPUT);
cli();
TCCR1A = 0;
TCCR1B = 0;
TCNT1 = 0;
OCR1A = 3124;
TCCR1B |= (1 << WGM12);
TCCR1B |= (1 << CS12) | (1 << CS10);
TIMSK1 |= (1 << OCIE1A);
sei();
cli();
wdt_reset();
WDTCSR |= (1 << WDCE) | (1 << WDE);
WDTCSR = (0 << WDIE) | (0 << WDE) | (0 << WDP3) | (1 << WDP2) | (1 << WDP1) | (1 << WDP0);
sei();
}
//Timer per comunicazione Bluetooth con il controller
ISR(TIMER1_COMPA_vect) {
switch (a) {
case 0:
Serial1.print("#:AR#");
Serial1.println(ARTel);
break;
case 1:
Serial1.print("#:DEC#");
Serial1.println(DECTel);
break;
case 2:
Serial1.print("#:TAR#");
Serial1.println(ARTarget);
break;
case 3:
Serial1.print("#:TDEC#");
Serial1.println(DECtarget);
break;
}
a = a + 1;
if (a > 3) {
a = 0;
}
wdt_reset();
}
//Loop principale
void loop() {
bluetooth();
if (Serial.available() > 0) {
communication();
}
if ((digitalRead(BTSTATE) == HIGH) && (flag == 0)) {
delay(2000);
Serial1.println("#1");
flag = 1;
} else if (digitalRead(BTSTATE) == LOW) {
flag = 0;
}
}

```

```

}
AR_encoder();
enablegoto = true;
ARdiff = ARtarget - ARtel;
DECdiff = DECtarget - DECTel;
if ((A == 1) && (DECdiff == 0)) {
  A = 0;
  if (ARtel >= 43200) {
    ARtel = ARtel - 43200;
  } else {
    ARtel = ARtel + 43200;
  }
  DECtarget = memoDEC;
  DECdiff = DECtarget - DECTel;
}
if (ARdiff < 20) {
  digitalWrite(CONTROL, HIGH);
} else {
  digitalWrite(CONTROL, LOW);
}
if ((A == 0) && (DECdiff == 0) && (ARdiff == 0)) {
  digitalWrite(EST, LOW);
  digitalWrite(OVEST, LOW);
}
if ((enablegoto == true) && ((ARdiff != 0) || (DECdiff != 0))) {
  goto_object();
  Serial1.println("#:GO#");
}
}
//-----COMUNICAZIONE STELLARIUM-----
void communication() {
  int i = 0;
  input[i++] = Serial.read();
  delay(5);
  while ((input[i++] = Serial.read()) != '#') {
    delay(5);
  }
  input[i] = '\0';
  if (input[1] == ':' && input[2] == 'G' && input[3] == 'R' && input[4] == '#') {
    transmitAR();
  }
  if (input[1] == ':' && input[2] == 'G' && input[3] == 'D' && input[4] == '#') {
    transmitDEC();
  }
  if (input[1] == ':' && input[2] == 'Q' && input[3] == '#') {
    digitalWrite(EST, LOW);
    digitalWrite(OVEST, LOW);
  }
  if (input[0] == ':' && input[1] == 'S' && input[2] == 'r') {
    getAR();
  }
  if (input[0] == ':' && input[1] == 'S' && input[2] == 'd') {
    getDEC();
  }
  if (input[0] == ':' && input[1] == 'M' && input[2] == 'S' && input[3] == '#') {
    Serial.print("0");
  }
}
//-----COMUNICAZIONE BLUETOOTH CONTROLLER-----
void bluetooth() {
  if (stringComplete == true) {
    inputBT[b] = '\0';
    if (inputBT[1] == 'S' && inputBT[2] == '1') {
      S = 1;
    }
  }
}

```

```

}
if (inputBT[1] == 'S' && inputBT[2] == '0') {
  S = 0;
}
if (inputBT[1] == 'G' && inputBT[2] == '1') {
  G = 1;
}
if (inputBT[1] == 'G' && inputBT[2] == '0') {
  G = 0;
}
if (inputBT[1] == 'V' && inputBT[2] == '1') {
  V = 1;
}
if (inputBT[1] == 'V' && inputBT[2] == '0') {
  V = 0;
}
if (inputBT[1] == 'C') {
  ARtel = ARPolar;
  DECtel = DECPolar;
}
b = 0;
stringComplete = false;
}
}
//-----TRASMISSIONE DATI AR STELLARIUM-----
void transmitAR() {
  arHH = floor(ARtel) / 3600;
  arMM = (floor(ARtel) - arHH * 3600) / 60;
  arSS = (floor(ARtel) - arHH * 3600) - arMM * 60;
  sprintf(txAR, "%02d:%02d:%02d#", int(arHH), int(arMM), int(arSS));
  Serial.print(txAR);
}
//-----TRASMISSIONE DATI DEC STELLARIUM-----
void transmitDEC() {
  (DECtel < 0) ? SIGNtel = 45 : SIGNtel = 43;
  decDEG = abs(floor(DECtel)) / 3600;
  decMM = (abs(floor(DECtel)) - decDEG * 3600) / 60;
  decSS = (abs(floor(DECtel)) - decDEG * 3600) - decMM * 60;
  sprintf(txDEC, "%c%02d%c%02d:%02d#", SIGNtel, int(decDEG), 223, int(decMM), int(decSS));
  Serial.print(txDEC);
}
//-----RICEZIONE DATI AR STELLARIUM-----
void getAR() {
  Serial.print("1");
  ARtarget = (atol(input + 4)) * 3600 + (atol(input + 7)) * 60 + atol(input + 10);
}
//-----RICEZIONE DATI DEC STELLARIUM-----
void getDEC() {
  Serial.print("1");
  DECtarget = (atol(input + 5)) * 3600 + (atol(input + 8)) * 60 + atol(input + 11);
  if (input[4] == '-') {
    DECtarget *= (-1);
  }
  memoDEC = DECtarget;
  betaN = (324000 - DECtel) + (324000 - DECtarget);
  deltaN = betaN + ((648000 - betaN) / 2);
  betaS = ((-324000) - DECtel) + ((-324000) - DECtarget);
  deltaS = abs(betaS) + ((648000 - abs(betaS)) / 2);

  if (((latitude > 0) && (DECtarget > (324000 - latitude))) || ((latitude < 0) && (DECtarget < ((-324000) - latitude)))) {
    circumpolar = true;
  } else {
    circumpolar = false;
  }
}

```

```

ARdist = ARtarget - ARtel;
if (ARdist < (-43200)) {
  ARdist = ARdist + 86400;
}
if (ARdist > 43200) {
  ARdist = ARdist - 86400;
}

if ((latitude > 0) && (circumpolar = true) && ((abs(ARdist) * 15) > deltaN)) {
  DECTarget = 324000;
  A = 1;
}
if ((latitude < 0) && (circumpolar = true) && ((abs(ARdist) * 15) > deltaS)) {
  DECTarget = (-324000);
  A = 1;
}
}
//-----FUNZIONE AVVIAMENTO MOTORE-----
void goto_object() {
  if (A != 1) {
    if ((ARdiff > 0 && ARdiff <= 43200) || (ARdiff <= (-43200))) {
      go_east();
    }
    if ((ARdiff > 43200) || (ARdiff < 0 && ARdiff > (-43200))) {
      go_west();
    }
  }
  if (DECTarget > DECTel) {
    go_north();
  }
  if (DECTarget < DECTel) {
    go_south();
  }
}
//-----AVVIO MOTORE AR-----
void go_east() {
  digitalWrite(EST, HIGH);
  digitalWrite(OVEST, LOW);
}
//-----AVVIO MOTORE AR-----
void go_west() {
  digitalWrite(EST, LOW);
  digitalWrite(OVEST, HIGH);
}
//-----AVVIO MOVIMENTAZIONE VIRTUALE DEC-----
void go_north() {
  DECTel = DECTel + 20;
}
//-----AVVIO MOVIMENTAZIONE VIRTUALE DEC-----
void go_south() {
  DECTel = DECTel - 20;
}
//-----FUNZIONE ENCODER-----
void AR_encoder() {
  long newPosition = myEnc.read();
  if (newPosition != oldPosition) {
    oldPosition = newPosition;
    ARtel = ARPolar + (double)0.098876953125 * newPosition;
  }
  if (ARtel > 86400) {
    ARtel = ARtel - 86400;
  }
  if (ARtel < 0) {
    ARtel = ARtel + 86400;
  }
}

```

```

}
}

//-----INTERRUPT BLUETOOTH-----
void serialEvent1() {
  while (Serial1.available()) {
    char inChar = (char)Serial1.read();
    inputBT[b++] = inChar;
    if (inChar == '#') {
      stringComplete = true;
    }
    delay(5);
  }
}
}

```

2.2 PROGRAMMAZIONE ARDUINO NANO (MOTORE)

```

//Dichiarazione variabili
int PUL = 4;
int DIR = 3;
int ENA = 2;
int OVEST = 5;
int EST = 6;
int CONTROL = 7;
int vel = 50;
int c, o, e;

//Impostazione Pin
void setup() {
  pinMode(PUL, OUTPUT);
  pinMode(DIR, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(OVEST, INPUT);
  pinMode(EST, INPUT);
  pinMode(CONTROL, INPUT);
}

//Loop principale
void loop() {
  o = digitalRead(OVEST);
  e = digitalRead(EST);
  c = digitalRead(CONTROL);
  if (c == 1) {
    vel = 500;
  } else {
    vel = 50;
  }
  if (e == 1) {
    digitalWrite(DIR, LOW);
    digitalWrite(ENA, HIGH);
    digitalWrite(PUL, HIGH);
    delayMicroseconds(vel);
    digitalWrite(PUL, LOW);
    delayMicroseconds(vel);
  } else if (o == 1) {
    digitalWrite(DIR, HIGH);
    digitalWrite(ENA, HIGH);
    digitalWrite(PUL, HIGH);
    delayMicroseconds(vel);
    digitalWrite(PUL, LOW);
    delayMicroseconds(vel);
  }
}
}

```

2.3 PROGRAMMAZIONE ARDUINO UNO (CONTROLLER)

```
//Librerie
#include <LiquidCrystal.h>
#include <stdlib.h>
#include <stdio.h>

//Dichiarazione variabili
#define STELLARIUM 13
#define GOTO 12
#define CALIBRAZIONE 11
#define VELOCITA 10
#define MENO 9
#define PIU 8

int S, G, C, V;
int M = 0, P = 0, M1 = 0, P1 = 0;
int MS, MG, MC, MV;
int CAL, PRESS = 0;
const int rs = 7, en = 6, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
int pot1, pot2, val = 0, pagina = 0;
char AAR[20];
char c = ' ';
char input[20];
int i = 0;
boolean stringComplete = false;
String BT = "OFF";
String inString = "";
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

//Impostazioni seriali, pin, LCD
void setup() {
  Serial.begin(9600);
  Serial.println("Arduino with HC-06 is ready");
  pinMode(STELLARIUM, INPUT);
  pinMode(GOTO, INPUT);
  pinMode(CALIBRAZIONE, INPUT);
  pinMode(VELOCITA, INPUT);
  pinMode(MENO, INPUT);
  pinMode(PIU, INPUT);
  lcd.begin(16, 2);
  Intro();
  lcd.clear();
}

//Loop principale
void loop() {
  if (stringComplete) {
    input[i] = '\0';
    if (input[1] == '1') {
      BT = "ON";
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("COMUNICAZIONE");
      lcd.setCursor(0, 1);
      lcd.print("TELESCOPIO ON");
      delay(2000);
      lcd.clear();
    }
    if (input[1] == '.' && input[2] == 'A' && input[3] == 'R' && input[4] == '#') {
      if (pagina == 0) {
        int cont = 0;
        for (int i = 0; i < 7; i = i + 1) {
          lcd.setCursor(i, 1);
        }
      }
    }
  }
}
```

```

    lcd.print(" ");
}
for (int i = 5; i < 21; i = i + 1) {
    if ((input[i] != '-') && (input[i] != '0') && (input[i] != '1') && (input[i] != '2') && (input[i] != '3') && (input[i] != '4') && (input[i] !=
'5') && (input[i] != '6') && (input[i] != '7') && (input[i] != '8') && (input[i] != '9')) {
        } else {
            lcd.setCursor(cont, 1);
            lcd.print(input[i]);
            cont = cont + 1;
            if (cont == 6) {
                i = 21;
            }
        }
    }
}
for (int i = 0; i < 21; i = i + 1) {
    input[i] = ' ';
}
}
if (input[1] == ':' && input[2] == 'D' && input[3] == 'E' && input[4] == 'C' && input[5] == '#') {
    if (pagina == 0) {
        int cont = 7;
        for (int i = 0; i < 9; i = i + 1) {
            lcd.setCursor(i + 7, 1);
            lcd.print(" ");
        }
        for (int i = 5; i < 21; i = i + 1) {
            if ((input[i] != '-') && (input[i] != '0') && (input[i] != '1') && (input[i] != '2') && (input[i] != '3') && (input[i] != '4') && (input[i] !=
'5') && (input[i] != '6') && (input[i] != '7') && (input[i] != '8') && (input[i] != '9')) {
                } else {
                    lcd.setCursor(cont, 1);
                    lcd.print(input[i]);
                    cont = cont + 1;
                    if (cont == 17) {
                        i = 21;
                    }
                }
            }
        }
        for (int i = 0; i < 21; i = i + 1) {
            input[i] = ' ';
        }
    }
}
if (input[1] == ':' && input[2] == 'T' && input[3] == 'A' && input[4] == 'R' && input[5] == '#') {
    if (pagina == 1) {
        int cont = 0;
        for (int i = 0; i < 7; i = i + 1) {
            lcd.setCursor(i, 1);
            lcd.print(" ");
        }
        for (int i = 5; i < 21; i = i + 1) {
            if ((input[i] != '-') && (input[i] != '0') && (input[i] != '1') && (input[i] != '2') && (input[i] != '3') && (input[i] != '4') && (input[i] !=
'5') && (input[i] != '6') && (input[i] != '7') && (input[i] != '8') && (input[i] != '9')) {
                } else {
                    lcd.setCursor(cont, 1);
                    lcd.print(input[i]);
                    cont = cont + 1;
                    if (cont == 6) {
                        i = 21;
                    }
                }
            }
        }
        for (int i = 0; i < 21; i = i + 1) {
            input[i] = ' ';
        }
    }
}

```

```

    }
  }
}
if (input[1] == ':' && input[2] == 'T' && input[3] == 'D' && input[4] == 'E' && input[5] == 'C' && input[6] == '#') {
  if (pagina == 1) {
    int cont = 7;
    for (int i = 0; i < 9; i = i + 1) {
      lcd.setCursor(i + 7, 1);
      lcd.print(" ");
    }
    for (int i = 5; i < 21; i = i + 1) {
      if ((input[i] != '-') && (input[i] != '0') && (input[i] != '1') && (input[i] != '2') && (input[i] != '3') && (input[i] != '4') && (input[i] !=
'5') && (input[i] != '6') && (input[i] != '7') && (input[i] != '8') && (input[i] != '9')) {
        } else {
          lcd.setCursor(cont, 1);
          lcd.print(input[i]);
          cont = cont + 1;
          if (cont == 17) {
            i = 21;
          }
        }
      }
    }
    for (int i = 0; i < 21; i = i + 1) {
      input[i] = ' ';
    }
  }
}
if (input[1] == ':' && input[2] == 'G' && input[3] == 'O' && input[4] == '#') {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("TELESCOPIO IN");
  lcd.setCursor(0, 1);
  lcd.print("MOVIMENTO");
  delay(2000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("COORDINATE");
  lcd.setCursor(0, 1);
  lcd.print("AGGIORNATE");
  delay(2000);
  lcd.clear();
}
i = 0;
stringComplete = false;
}
comandi();
if (M != M1) {
  if (M == 0) {
    pagina++;
    for (int l = 0; l < 16; l++) {
      lcd.setCursor(0, 0);
      lcd.print("");
      lcd.setCursor(0, 1);
      lcd.print("");
      lcd.scrollDisplayRight();
      delay(50);
    }
    lcd.clear();
    if (pagina > 2) {
      pagina = 0;
    }
  }
}
M1 = M;
}

```

```

if (P != P1) {
  if (P == 0) {
    pagina--;
    for (int l = 0; l < 16; l++) {
      lcd.setCursor(0, 0);
      lcd.print("");
      lcd.setCursor(0, 1);
      lcd.print("");
      lcd.scrollDisplayLeft();
      delay(50);
    }
    lcd.clear();
    if (pagina < 0) {
      pagina = 2;
    }
  }
  P1 = P;
}
switch (pagina) {
  case 0:
    lcd.setCursor(0, 0);
    lcd.print("AR:");
    lcd.setCursor(7, 0);
    lcd.print("DEC:");
    lcd.setCursor(13, 0);
    lcd.print("TEL");
    break;
  case 1:
    lcd.setCursor(0, 0);
    lcd.print("AR:");
    lcd.setCursor(7, 0);
    lcd.print("DEC:");
    lcd.setCursor(13, 0);
    lcd.print("TAR");
    break;
  case 2:
    lcd.setCursor(0, 0);
    lcd.print("VEL: ");
    lcd.setCursor(5, 0);
    lcd.print("1000");
    lcd.setCursor(0, 1);
    lcd.print("BT: ");
    lcd.setCursor(4, 1);
    lcd.print(BT);
    break;
}
S = digitalRead(STELLARIUM);
G = digitalRead(GOTO);
C = digitalRead(CALIBRAZIONE);
V = digitalRead(VELOCITA);
M = digitalRead(MENO);
P = digitalRead(PIU);
pot1 = analogRead(0);
pot2 = analogRead(1);

```

}
 //-----SCRITTA INTRODUZIONE-----

```

void Intro() {
  lcd.setCursor(0, 0);
  lcd.print("CONTROLLO REMOTO");
  lcd.setCursor(0, 1);
  lcd.print("TELESCOPIO");
  delay(2000);
  for (int l = 0; l < 16; l++) {

```

```

lcd.setCursor(0, 0);
lcd.print(" ");
lcd.setCursor(0, 1);
lcd.print(" ");
lcd.scrollDisplayLeft();
delay(150);
}
}

```

//-----INVIO COMANDI AL TELESCOPIO-----

```

void comandi() {
if ((S == 1) && (MS == 0)) {
  Serial.println("#S1");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("COMUNICAZIONE");
  lcd.setCursor(0, 1);
  lcd.print("STELLARIUM ON");
  delay(1000);
  lcd.clear();
  MS = 1;
} else if ((S == 0) && (MS == 1)) {
  Serial.println("#S0");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("COMUNICAZIONE");
  lcd.setCursor(0, 1);
  lcd.print("STELLARIUM OFF");
  delay(1000);
  lcd.clear();
  MS = 0;
}
if ((G == 1) && (MG == 0)) {
  Serial.println("#G1");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("GOTO");
  lcd.setCursor(0, 1);
  lcd.print("ABILITATO");
  delay(1000);
  lcd.clear();
  MG = 1;
} else if ((G == 0) && (MG == 1)) {
  Serial.println("#G0");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("GOTO");
  lcd.setCursor(0, 1);
  lcd.print("DISABILITATO");
  delay(1000);
  lcd.clear();
  MG = 0;
}
if ((C == 1) && (MC == 0)) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("CALIBRAZIONE");
  delay(1000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("CALIBRARE LA");
  lcd.setCursor(0, 1);
  lcd.print("POSIZIONE? NO SI");
  Calibrazione();
  if (CAL == 1) {

```

```

Serial.println("#C");
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("POSIZIONE");
lcd.setCursor(0, 1);
lcd.print("CALIBRATA");
delay(2000);
lcd.clear();
} else {
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("POSIZIONE");
lcd.setCursor(0, 1);
lcd.print("NON CALIBRATA");
delay(2000);
lcd.clear();
}
do {
lcd.setCursor(0, 0);
lcd.print("RESETTARE");
lcd.setCursor(0, 1);
lcd.print("CALIBRAZIONE");
C = digitalRead(CALIBRAZIONE);
if (C == 0) {
MC = 1;
lcd.clear();
CAL = 0;
}
} while (MC == 0);
} else if ((C == 0) && (MC == 1)) {
MC = 0;
}
if ((V == 1) && (MV == 0)) {
Serial.println("#V1");
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("VELOCITA");
lcd.setCursor(0, 1);
lcd.print("ALTA");
delay(1000);
lcd.clear();
MV = 1;
} else if ((V == 0) && (MV == 1)) {
Serial.println("#V0");
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("VELOCITA");
lcd.setCursor(0, 1);
lcd.print("BASSA");
delay(1000);
lcd.clear();
MV = 0;
}
}
//-----CALIBRAZIONE-----
void Calibrazione() {
do {
M = digitalRead(MENO);
P = digitalRead(PIU);
if (M != M1) {
if (M == 0) {
CAL = 0;
PRESS = 1;
}
}
}

```

```
M1 = M;
}
if (P != P1) {
  if (P == 0) {
    CAL = 1;
    PRESS = 1;
  }
  P1 = P;
}
} while (PRESS == 0);
PRESS = 0;
}
//-----INTERRUPT BLUETOOTH-----
void serialEvent() {
  while (Serial.available()) {
    char inChar = (char)Serial.read();
    input[i++] = inChar;
    if (inChar == '#') {
      stringComplete = true;
    }
    delay(5);
  }
}
```

3 PROGETTAZIONE SUPPORTO MOTORE

Lo studio per creare il supporto del motore è avvenuto valutando la struttura della montatura, cercando di sfruttare l'alloggio già predisposto dal costruttore e prendendo le misure per poter poi sviluppare una struttura 3D che sostenesse il motore, il tutto per poter posizionare il motore in modo tale da applicare una trasmissione a cinghie dentate (l'ideale per avere una buona trazione). Quest'ultima è stata studiata approfonditamente, calcolando le varie dimensioni delle pulegge (le quali sono state acquistate da un'azienda specializzata).

4 MONTAGGIO MOTORE

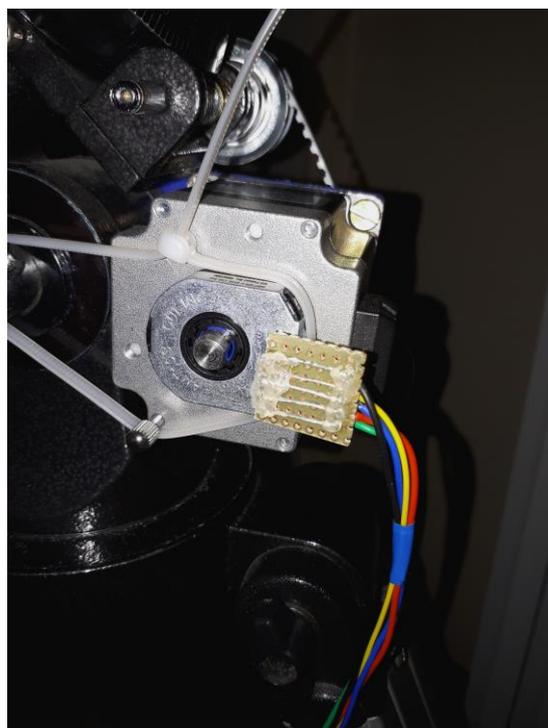
Il supporto studiato si è rivelato subito perfetto ed ha permesso di alloggiare il motore nel modo voluto. Si sono dovute però applicare alcune fascette per mantenere stabile la posizione, siccome il materiale del supporto è leggermente flessibile.

5 APPLICAZIONE DI PULEGGE E CINGHIE PER LA TRASMISSIONE

Sono state applicate le due pulegge agli alberi, fissandole con alcune viti. All'interno delle due viene fatta scorrere la cinghia dentata, la quale si va ad adagiare perfettamente nelle insenature predisposte. Infine sono state effettuate prove di movimentazione.



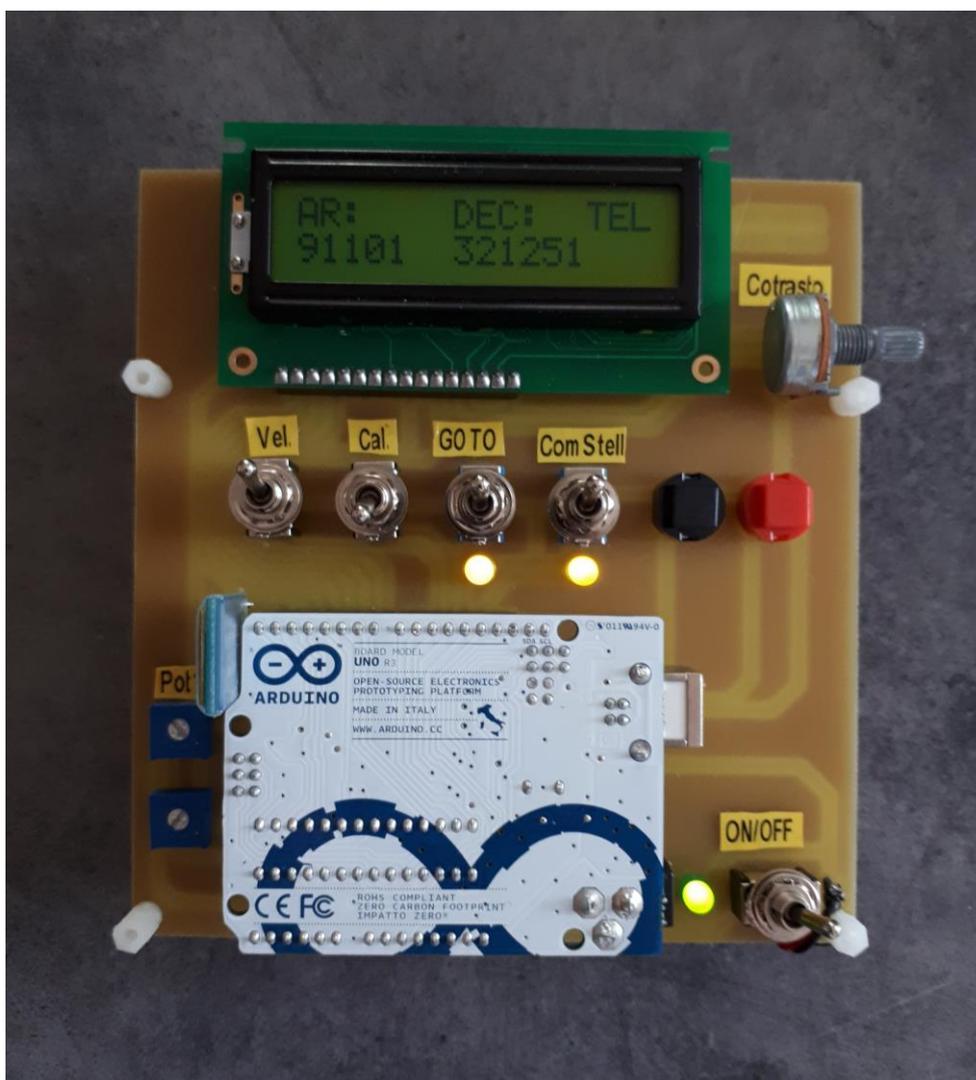
26. Foto: motore e trasmissione montati



25. Foto: encoder montato e cavo trasmissione dati

6 PROGETTAZIONE SCHEDA CONTROLLER E TELESCOPIO

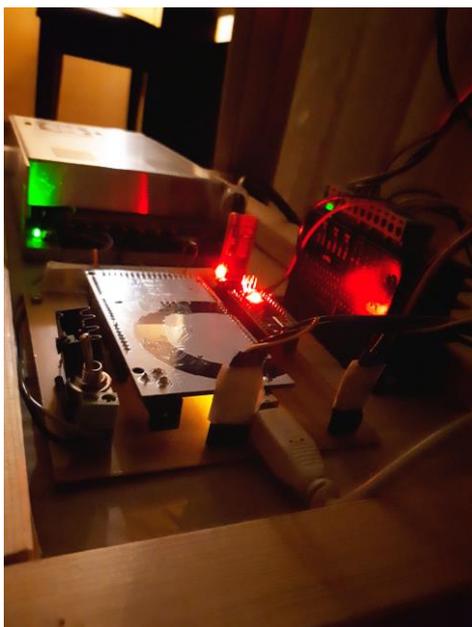
Tramite l'utilizzo del programma denominato OrCAD sono state sviluppati i PCB per il controller e per il telescopio. Sono stati realizzati a scuola grazie all'utilizzo di una fresatrice, riuscendo a stampare le schede su basette ramificate in vetronite.



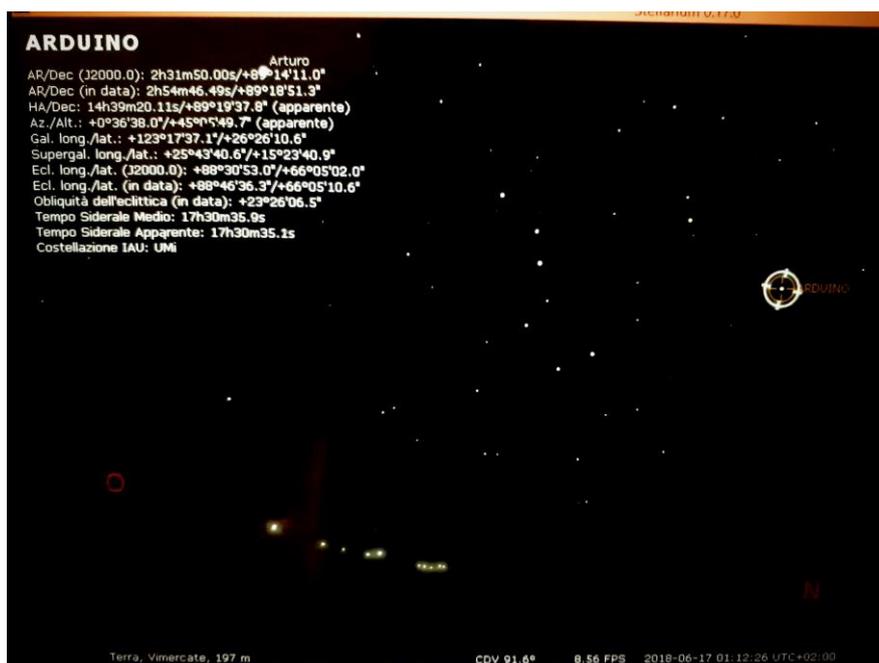
27. Foto: Controller

7 TEST FUNZIONAMENTO ED EVENTUALI CORREZIONI

Dopo aver realizzato la struttura meccanica, si è assemblata la parte elettronica su un supporto (studiato per essere posizionato sul treppiedi del telescopio) e si sono svolti vari test di funzionamento, i quali hanno messo in luce alcuni inconvenienti e alcune problematiche del codice sviluppato, soprattutto di entità teorica (calcoli errati). Apportando le giuste correzioni e modifiche, si è arrivati ad ottenere un risultato soddisfacente a livello sperimentale.



29. Foto: disposizione della parte elettronica durante i test



28. Foto: Rilevamento posizione telescopio su Stellarium

8 PRIMA OSSERVAZIONE

L'ultimo passo di questo progetto è stato verificare il funzionamento nel complesso del sistema di auto puntamento. Si è scelto come prima osservazione la Luna perché rappresenta l'oggetto più vicino e grande nel nostro cielo stellato.

Il risultato finale è stato soddisfacente: il telescopio si è portato sull'asse di Ascensione Retta richiesto e la gestione della comunicazione con l'utente è stata esaustiva. Quindi, si è potuto osservare le varie peculiarità della dell'astro senza nessun problema.



32. Foto: Impostazione della strumentazione



31. Foto: Oculare



30. Foto: prima osservazione della Luna

CONCLUSIONE

In conclusione, il progetto si è rivelato complesso come inizialmente previsto e, secondo un mio parere personale, è riuscito ad esporre appieno tutte le capacità che ho acquisito durante questi anni di scuola:

- risolvere problematiche maggiori e minori
- impiegare capacità tecniche (elettroniche ed informatiche)
- utilizzare una buona capacità di valutazione dei componenti
- ricercare informazioni
- dimostrare capacità gestionali.

Questo progetto, inoltre, mi ha avvicinato di più alla mia passione, ovvero l'astronomia, perché mi ha permesso di approfondire il lato teorico di questa scienza e il lato tecnico dell'osservazione stellare. Sommariamente, il sistema da me sviluppato è stato sia una sfida personale sia un metodo per continuare ad apprendere ciò che per me è ancora ignoto e che mi interessa.

Infine, vorrei ringraziare il professore Salvatore Pagano per la sua disponibilità, durante tutto lo sviluppo del progetto, e tutti coloro che mi hanno sostenuto e aiutato nel raggiungere questo traguardo.



33. Foto: progetto concluso

Leonardo Brambilla

USER'S MANUAL

Self-Pointing system for an analog telescope



GENERAL INFORMATION

SYSTEM OVERVIEW

This self-pointing system for an analog telescope allows to control an instrumentation for stellar observation automatically.

The purpose of this project is simplify the search for celestial bodies to the user, so it can observe the planets or stars that would be difficult to found manually.

SYSTEM REQUIREMENTS

The telescope operates with computer devices because it works with Stellarium, an open-source planetary application, that get at the instrumentation all the data for the localization of the celestial objects. The application requires a computer with good quality of memory and video component. the controller, once started, will automatically connect via Bluetooth to the telescope's Arduino, so, after installation of the device, it can be used immediately without any further configuration.

GETTING STARED

1. BALANCING

Telescope should be balanced before each observing session. Balancing reduces stress on telescope mount and allows precise control of micro-adjustment. A balanced telescope is specially critical when using the optional clock drive for astrophotography. The telescope should be balanced after all accessories (eyepiece, camera, etc.) have been attached. Before balancing your telescope, make sure that your tripod is balanced and on a stable surface.

1.1 RA BALANCING

- 1) Adjust altitude of the mount to between 15° and 30° by using the altitude adjustment T-bolt.
- 2) Slowly unlock the R.A. and Dec. lock knobs. Rotate the telescope until both the optical tube and counterweight rod is horizontal to the ground, and the telescope tube is to the side of the mount.
- 3) Tighten the Dec. lock knob.
- 4) Move counterweight along counterweight rod until telescope is balanced and remains stationary when released
- 5) Tighten counterweight thumb screws to hold the counterweight in its new position.

1.2 DEC BALANCING

All accessories should be attached to the telescope before commencing with balancing the declination axis. The R.A. axis should be balanced before proceeding with Dec. balancing.

- 1) Adjust altitude of the mount to between 60° and 75°.
- 2) Release the R.A. lock knob and rotate R.A. axis so that the counterweight rod is in horizontal position. Tighten the R.A. thumbscrew.
- 3) Unlock the Dec. thumbscrew and rotate telescope tube until it is paralled to the ground.
- 4) Slowly release telescope and determine which direction the telescope rotates. Loosen telescope tube rings and slide telescope tube forward or backward in the clamps to balance Dec. axis.

5) Once telescope no longer rotates from its parallel starting position, re-tighten tube rings and the Dec. lock knob. Reset altitude axis to your local latitude.

2. CALIBRATION

Before starting to connect the electronics, to allow the telescope to correctly track the apparent motion of celestial bodies, you must align your mount by directing its straight Ascension axis to the celestial North pole. the procedure is as follows:

- loosen the declination axis knob and rotate the telescope tube until the index of the graduated circle is at a 90 ° angle and lock the axis.
- loosen the declination axis knob and rotate the telescope tube until the telescope is pointing at the polar star (use the finder as well) and lock the axis.

3. ELETTRICAL CONNECTION

Insert the power plug and switch on the Arduino Nano, which controls the motor, and the Bluetooth module using the special switch.

Connect the USB cable of the Arduino Mega, which controls the telescope, at the computer with Stellarium.

4. STELLARIUM CONFIGURATION

Before configuring Stellarium we need to know which communication port our computer has associated with Arduino. So, connect Arduino to the computer and then go to control Panel/System/Hardware click on "device management". In the list you must click on "Ports (COM and LPT)" and, at this point, you know which communication port is associated with Arduino.

Now let's move on Stellarium. Once the installation is complete and the program is configured according to your location, go to Configuration/Plugins/Telescope control and activate the "Load at start up" box. Click on "configure" and go to "Telescopes:" and click on "Add". On the "Telescope controlled by:" select the first option "Stellarium, directly through a serial port", then give the telescope a name, activate the "Start / start all" 'start of the program', indicate which serial port is associated with Arduino and in the model of the device selected "Meade LX200 (compatible)"(this is the communication protocol that was adopted to make Arduino dialogue with Stellarium.

At this point you can give the OK. In the "Telescopes:" window you will see the presence of the telescope you have just entered. If you click on "Start" you will connect Stellarium with Arduino, but nothing will happen because in Arduino we have not loaded any sketch, but Stellarium already recognizes it as a peripheral.

5. USING THE SYSTEM

After configuration, select the celestial body to be displayed and use the command "CTRL" + "telescope number" (depending on how many devices are connected). In this way, Stellarium will send the coordinates of the selected object to the Arduino Mega, which will move the AR axis in the right position and will inform the user (through the controller) about how to move the DEC axis. Finally, you will have to look through the eyepiece.

MAINTENANCE AND WARNINGS

This device is very sensitive, so the utmost care is required in handling the instrumentation. Also, be very careful and don't touch the power supply near high voltage cables when the power supply is turned on and the mechanical transmission when is in movement.

As far as maintenance is concerned, clean the device from dust periodically to avoid dangerous static charges and always check the stability of the structure.



BIBLIOGRAFIA

- KARTTUNEN H., KROGER P., OJA H., POUTANEN M., DONNER K.J., *Fundamental Astronomy*, Heidelberg New York Dordrecht Londra, Springer, 2017, sesta edizione
- *Libreria geografica, Atlante di Astronomia*, Novara, Geo4map S.r.l., 2016
- SPELTI I., *Sette lezioni di Astronomia. Corso introduttivo*, Edizioni del faro, 2015
- BOURGEE P., LACROUX J., DUPONT-BLOCH N., *À l'affut des étoiles. Guide pratique de l'astronome amateur*, Parigi, Dunod, 2007, 16° edizione
Traduzione: TANGA P. (a cura di), *Manuale pratico di Astronomia*, Bologna, Zanichelli editore s.p.a., 2010, seconda edizione italiana
- ORTOLANI G., VENTURI E. (a cura di), *Manuale di Elettrotecnica Elettronica e Automazione*, , Hoepli, 2017, seconda edizione
- *Datasheet dei componenti*

SITOGRAFIA

- <http://stellarium.org/it/>
- <https://www.arduino.cc/>
- <http://astro.neutral.org/arduino/arduino-telescope-control.shtml>
- http://stellarium.sourceforge.net/wiki/index.php/Telescope_Control_plugin
- <http://ascom-standards.org/Developer/AppStart.htm>
- [https://www.astronomiamo.it/Articolo.aspx?Arg=i telescopi a specchi telescopi newton](https://www.astronomiamo.it/Articolo.aspx?Arg=i%20telescopi%20a%20specchi%20telescopi%20newton)
- https://www.astroshop.it/telescopi/skywatcher-telescopio-n-130-900-explorer-eq-2/p,5013#tab_bar_1_select
- https://www.google.it/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwjWrZTCqbjbAhUFh6YKHZBaCvwQFggoMAA&url=https%3A%2F%2Fwww.atti.it%2Fmedia%2Fget%2Fmanuale-tecnico-cinghie-atti_ab0f1.pdf&usg=AOvVaw0orevVZaTfWSEbXbZxrXwf
- <https://www.mouser.it/cui-inc/ù>