



UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

Corso di Laurea triennale in Ingegneria Meccanica

Realizzazione di un Banco per il Foto-Segnalamento 2.0

Progettazione del Software

Construction of a test bed for the Photo-Signaling 2.0

Software Design

Relatore: Chiar.mo

Prof. Castellini Paolo

Tesi di Laurea di:

Canullo Tommaso

A.A. 2020 / 2021

Introduzione

La tesi in questione cercherà di spiegare in che modo, e con quali obiettivi, è avvenuta la realizzazione di una struttura, ideata per migliorare tecnologicamente il foto-segnalamento di individui. Il progetto infatti, nato in collaborazione con le forze dell'ordine, ha come scopo principale quello di acquisire (nel modo più efficiente possibile), non più una singola foto, bensì un numero sufficientemente elevato di immagini, scattate da diverse altezze e angolazioni, tali da rendere possibile l'identificazione anche con delle riprese parziali di un volto. Con queste ultime, infatti, si trova a dover lavorare frequentemente la polizia, visionando i nastri delle videocamere di sorveglianza, ad esempio.

L'intento della trattazione è quello di porre l'accento sul software progettato per controllare la struttura: si è scelto, per motivi di praticità, di utilizzare il programma LabView, della National Instruments. Insieme al candidato Daniele Lambertucci, addetto alla parte hardware, si è pensato di movimentare una trave arcuata di metallo leggero, sulla quale sono fissate tre telecamere, ovviamente ad altezze diverse. Grazie a un sostegno verticale, viene permessa la rotazione dell'arco, e l'acquisizione di fotografie all'interno di un angolo opportuno.

Fondamentale inoltre, è stato l'utilizzo di una scheda Arduino e di un motore elettrico Stepper, per cui è stato creato un apposito sub.VI. Quest'ultimo, in LabView, è un programma usato come sottoprogramma all'interno di uno più grande; l'argomento verrà trattato in seguito. Il driver e il motorino, essendo collegati al file principale, riescono a movimentare la struttura.

Materials and Methods

Specifica di progetto

Entrando nel vivo della questione, si vuole fin da subito chiarire qual è stata la specifica alla base della progettazione della struttura. L'obiettivo, come accennato in precedenza, era controllare automaticamente un sistema in grado di acquisire e salvare foto segnaletiche; le quali vengono scattate da una distanza e un'altezza consone per la messa a fuoco ottimale delle telecamere, consentendo così una visibilità quanto più completa e dettagliata del volto dell'individuo.

Analizzando i materiali e le apparecchiature a nostra disposizione, dopo varie ipotesi e confronti, si è giunti a delineare le dimensioni generali della struttura, nonché la logica con la quale programmare il software:

- i) altezza del sostegno di circa 2 metri;
- ii) raggio di curvatura di 1.5 metri;
- iii) angolo di moto della slitta, intorno ai 150°;
- iv) necessità (per le varie camere) di scattare simultaneamente le fotografie.

Cenni sull'hardware

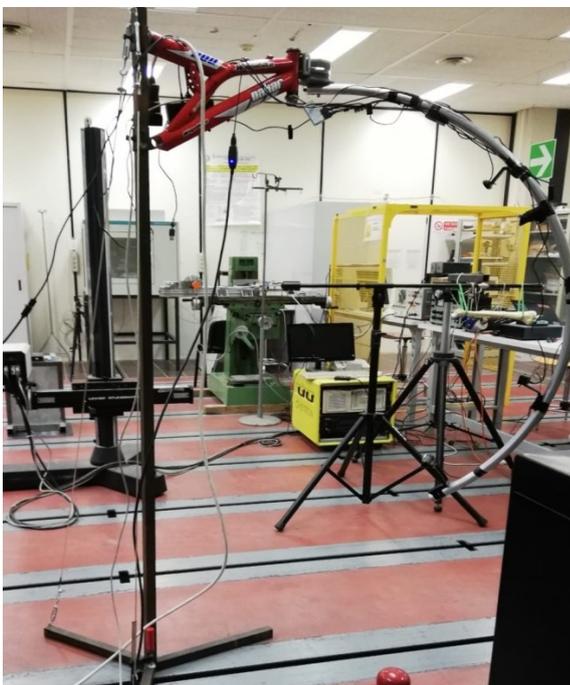


Questa versione 3D della struttura è stata realizzata dal collega Daniele Lambertucci, attraverso il programma cad "Solid Edge".

Il cubo azzurro rappresenta la posizione dove, una volta portata a termine la struttura, è stato posizionato il motore elettrico stepper.

I suoi collegamenti con il PC scorrono, in maniera poco visibile, dietro al pilastro verticale.

Il sostegno al motore, la cui utilità è anche quella di far arrivare il centro di rotazione della slitta proprio sopra la testa dell'individuo, è stato ricavato dal telaio di una bicicletta, e lavorato nell'officina dell'univpm. In un primo momento, la parte mobile della struttura aveva questa forma a L ribaltata (di 90° in senso orario); tuttavia, in seguito, siamo riusciti a montare un arco di metallo leggero, molto più facilmente muovibile e altrettanto funzionale, per un corretto posizionamento delle telecamere.



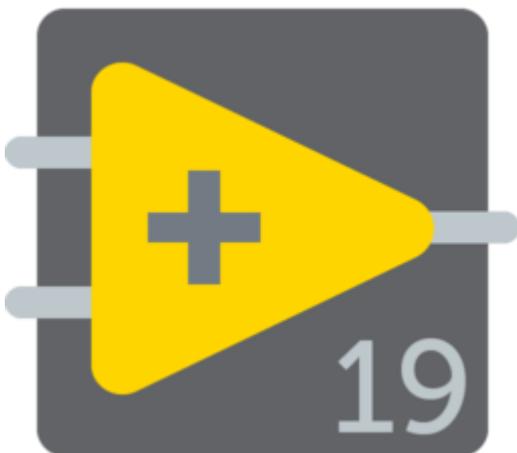
Così, infatti, appariva la struttura ottenuta dopo le lavorazioni in officina, con le telecamere fissate lungo l'arco. L'attrezzatura a disposizione richiedeva molti cavi, ma con tecnologie più avanzate e l'uso del wi-fi, il tutto potrebbe essere completamente svincolato dai fili.

Scelta del Programma: LabView

Per quanto concerne il software, la scelta è stata quella di utilizzare LabView, per vari motivi (oltre al consiglio del Prof. Castellini); la praticità del programma di proprietà della NI infatti, abbinata alla sua complessità e vastità applicativa, ha semplificato notevolmente il lavoro.

Ideato nel 1986 per Apple, LabView viene principalmente utilizzato per programmi di acquisizione e analisi dati, controllo di processi, generazione di rapporti, o più generalmente per tutto ciò che riguarda l'automazione industriale, su diverse piattaforme: Windows, Linux e macOS. Esso è un ambiente di sviluppo integrato per il “linguaggio G”: a differenza di molti altri linguaggi, la sua sintassi non è scritta, ma grafica (Graphic language). Il programma G infatti, denominato VI (Virtual Instrument), “non esiste sotto forma di testo, ma può essere salvato solo come un file binario che può essere aperto e compilato esclusivamente da LabView”*.

Il software è basato sulla costruzione e interazione di circuiti logici, creati anche a partire da blocchi già pronti, utilizzati per il collegamento con l'hardware. Anche nel caso di questo progetto infatti, grazie ai driver di LabView per Arduino e al sub.VI creato per controllare il motore elettrico, è stato possibile realizzare un banco automatizzato controllato da un unico PC.

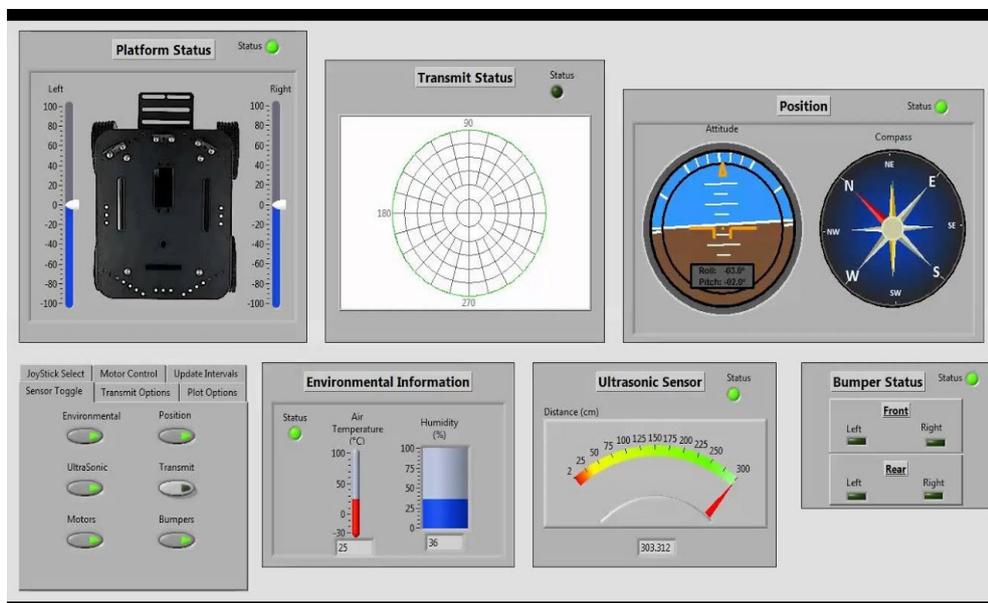


Logo dell'edizione “LabView 2019”.

*Wikipedia, Labview, Programmazione G.

Labview è composto da due parti principali, il Front Panel e il Block Diagram (Pannello Frontale e Schema a Blocchi): il primo è l'interfaccia utente di LabView, mentre il secondo rappresenta il codice (il circuito), così come viene scritto dal programmatore.

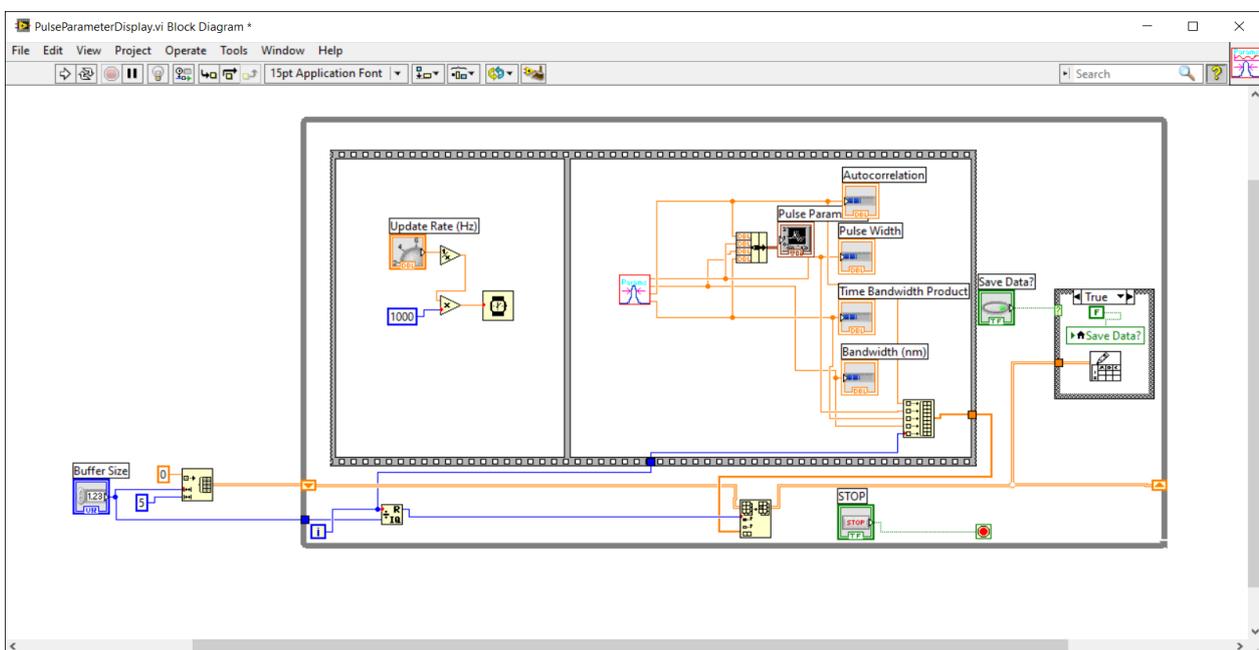
Il Front Panel comprende controlli e indicatori, che costituiscono rispettivamente i terminali interattivi d'ingresso e d'uscita all'interno del circuito. I principali controlli sono manopole, matrici, quadranti e array; essi "simulano i dispositivi d'ingresso degli strumenti e forniscono dati allo schema a blocchi del VI". Gli indicatori sono grafici, tabelle, strumenti di misura e molti altri; "simulano i dispositivi d'uscita degli strumenti e visualizzano i dati che lo schema a blocchi acquisisce o genera".



Alcuni tra i vari controlli presenti nel front panel.

*Wikipedia, Labview, Pannello Frontale.

Il Block Diagram è formato da funzioni, costanti e strutture, collegate tra loro da fili o bundle di fili. Le funzioni sono esse stesse dei VI, a un numero indefinito di ingressi e uscite, anche se non hanno un loro Front Panel e Block Diagram. D'altro canto, degli esempi di struttura sono la Flat Structure, capace di rendere simultanee le operazioni interne ad essa; oppure il For Loop, rappresentato come un rettangolo, che reitera una funzione tutte le volte impostate dal programmatore. Molto utile, inoltre, è la diversa colorazione dei fili: ad esempio i numeri interi scorrono su fili blu, mentre i numeri decimali in quelli arancioni.

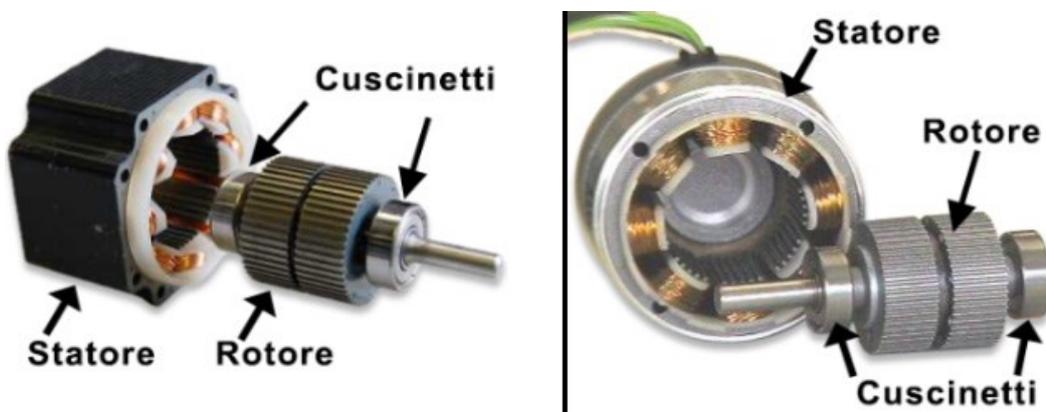


Esempio di codice nel block diagram.

Infine, come accennato in precedenza, LabView offre la possibilità di inserire dei VI all'interno di programmi di maggiori dimensioni. Attraverso il riquadro connettori, infatti, si ha la possibilità di dare una forma, un colore e un'identificazione a un VI, tramite un'icona, così da poterlo utilizzare come sottoprogramma (sub.VI).

Singola camera e movimentazione a step

In un primo momento, era stato realizzato un VI che muoveva la struttura azionando una sola camera; l'idea era quella che il programma sarebbe stato poi ampliato, fino ad arrivare a un numero opportuno di fotografie per ogni angolazione. Inoltre, ragionando con il collega Lambertucci, il Prof. Castellini e il Dott. Annessi (ricercatore all'Univpm), inizialmente avevamo pensato di acquisire una foto per ogni step del motore: partendo da un angolo totale di 150°, l'opzione migliore era quella di far eseguire al motore 5 passi, ognuno di 30°.

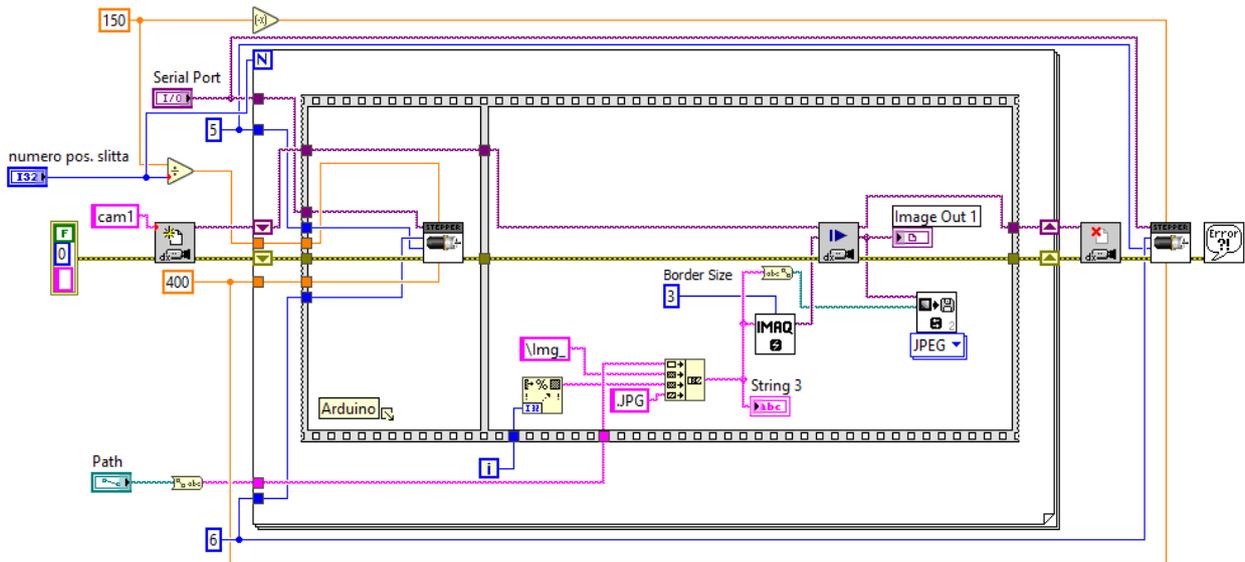


Questa è la struttura interna di un motore elettrico stepper; i fili che escono dallo statore creano il collegamento con la scheda Arduino.

La logica di progetto, relativamente a LabView, era quella di eseguire in sequenza le operazioni: ogni passo del motore veniva seguito dall'acquisizione di una fotografia, reiterando il ciclo per 5 volte. Alla fine, la slitta dov'era fissata la telecamera, veniva fatta tornare al punto di partenza, con un grande step di 150° (nel verso opposto a quello iniziale, assegnando come input un angolo negativo). Tutto ciò iniziando e spegnendo le webcam una sola volta (precedentemente allo step iniziale e quando tutte le foto sarebbero state scattate).

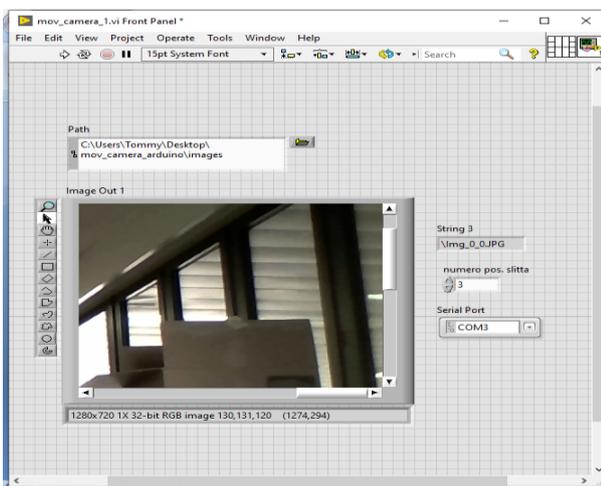
Per quanto riguarda il numero di iterazioni del ciclo, esso corrisponde a quello delle diverse posizioni occupate dalla slitta, impostate attraverso un controllo. Internamente al ciclo for, si trova una flat structure con due frame, il primo per la movimentazione del motore (attraverso l'apposito sub.VI), e quello successivo per l'acquisizione. Questa struttura garantisce che la foto venga scattata solo dopo che la slitta abbia completato lo step relativo.

Esternamente al ciclo for, nella parte iniziale (a sinistra), troviamo i valori in ingresso da fornire al motore passo-passo, e la fase di inizializzazione della camera. Alcuni di questi dati, come la frequenza, rappresentano i gradi di libertà disponibili per migliorare l'efficienza di ogni programma, una volta scritto il codice. D'altro canto, nella parte finale, a destra del ciclo for, troviamo le icone relative allo spegnimento della telecamera, e al ritorno della struttura al punto di partenza.



Il codice, così come viene mostrato nel block diagram.

Come si può notare dall'immagine, l'angolo in ingresso, per entrambe le icone che richiamano il sub.vi dello stepper, parte dalla stessa costante numerica (150), poi nel primo caso viene diviso per il numero di iterazioni del ciclo for (numero di posizioni che occuperà la slitta), mentre nella parte finale, esso viene negativizzato: il moto di ritorno infatti, sarà di modulo uguale, ma di verso opposto a quello di funzionamento.

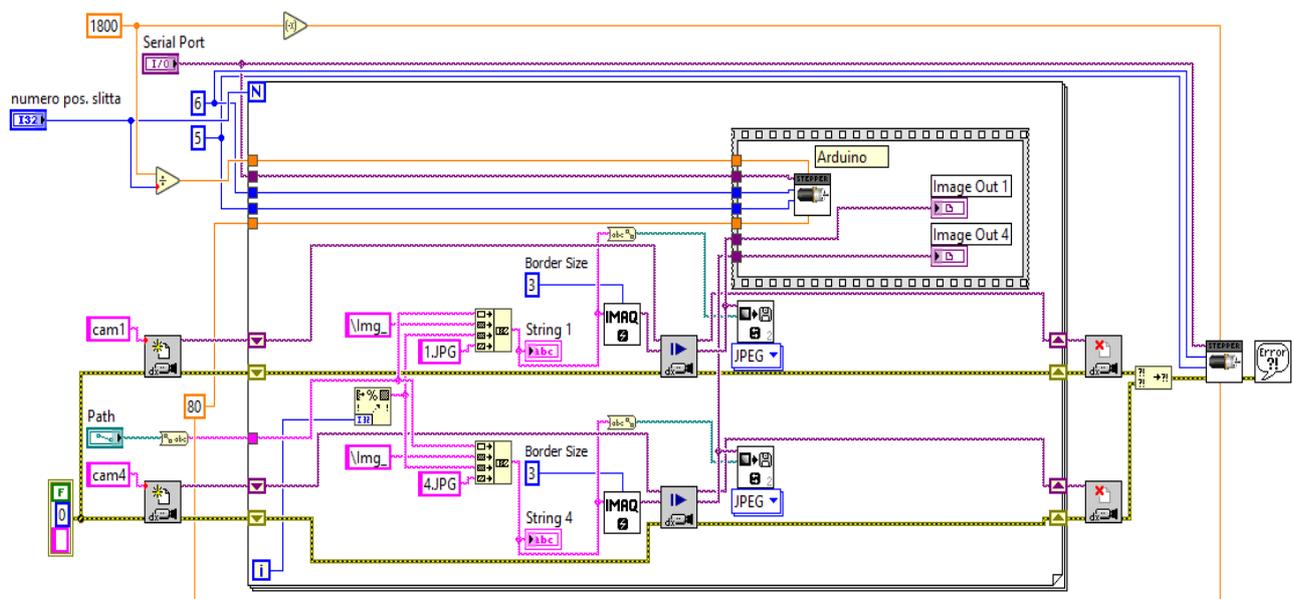


Il front panel, con i controlli per l'utente.

Due telecamere

Una volta portato a termine il programma per una sola webcam, e verificatone il funzionamento, abbiamo aumentato il numero, con la stessa logica di progetto. Per semplicità abbiamo preferito assegnare una linea per ogni fotocamera, inserendole entrambe all'interno del ciclo for.

Questa soluzione evidenziava dei problemi: con due telecamere da gestire il moto della struttura risultava molto meno fluido, evidenziando vibrazioni in partenza e all'arresto in ogni step, generando un risultato macchinoso, lento e anche rumoroso. Nonostante i numerosi tentativi di regolazione della frequenza, della velocità di movimento e dell'angolo di rotazione (tramite il substepping*), anche su consiglio del Professore, abbiamo preferito cambiare strategia.



Il codice, con l'aggiunta di una linea per la seconda telecamera.

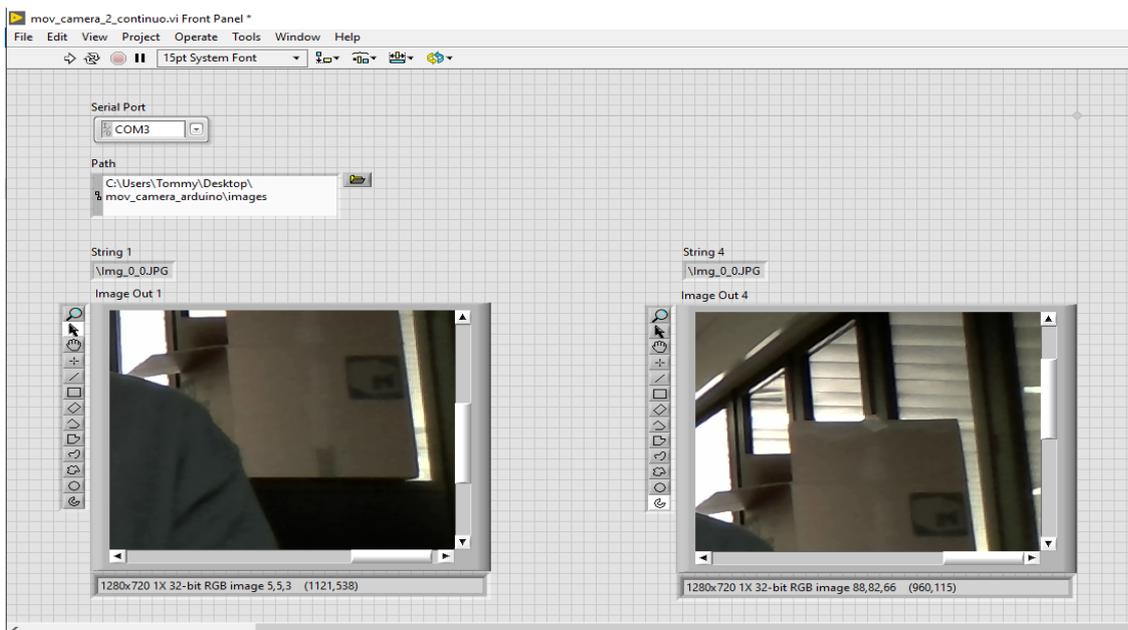
*Il substepping è una proprietà di Arduino, per la quale esso riesce, attraverso appositi tasti, a ridurre l'angolo impostato in Labview, muovendo il motore di un angolo frazionato; questa operazione solitamente riduce le vibrazioni prodotte dal moto.

Ragionando su quali fossero le vibrazioni e le velocità massime della slitta, tali da consentire comunque una buona messa a fuoco delle telecamere, è nata l'idea di progettare un circuito che muovesse la struttura con un unico grande step di 150°, in maniera continuativa, senza interruzioni e sufficientemente lenta; reiterando gli scatti delle telecamere per il numero di volte necessario.

Era molto importante essere sicuri che, al momento della partenza della slitta, le telecamere fossero accese: altrimenti avremmo potuto perdere le prime foto. Al tempo stesso non potevamo permettere che le telecamere iniziassero a scattare anzitempo, rischiando di avere troppe foto da determinate angolazioni, e poche da altre.

La soluzione più efficace, dal nostro punto di vista, era quella di inserire un frame di flat structure nella parte iniziale del circuito, che rendesse simultanee l'inizializzazione delle telecamere e la partenza della slitta: al suo interno infatti abbiamo posto l'icona relativa al sub.vi del motore (con frequenza e angolo assegnati), e quella per l'inizializzazione di ogni telecamera.

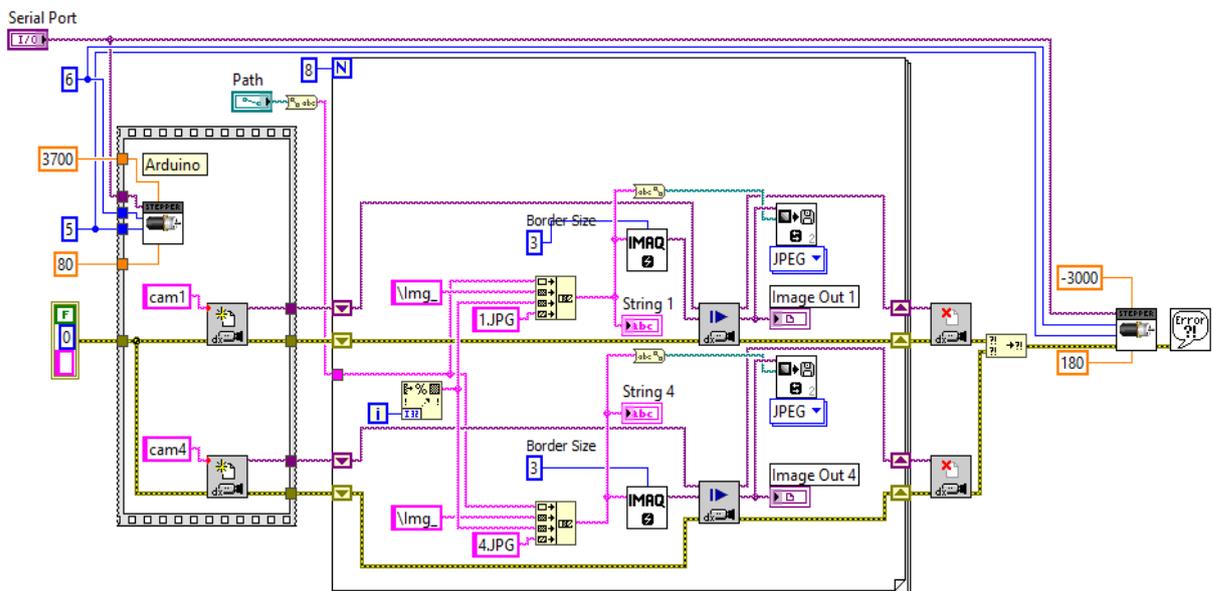
Una volta trovati i giusti dati da assegnare in ingresso al motore, tali da avere una vibrazione e una velocità limitate, si passava alla fase successiva della programmazione.



Il front panel, con gli spazi per controllare la qualità delle foto.

Il punto principale, ora, era essere sicuri che le telecamere acquisissero l'immagine nella maniera più corretta, rispettando le angolazioni richieste. In un primo momento sembrava imprescindibile l'aggiunta di un temporizzatore, così da separare ogni scatto da quello successivo, in maniera precisa e ben distribuita.

Tuttavia, semplicemente utilizzando un ciclo for, con all'interno le parti relative all'acquisizione (di entrambe le webcam), le fotografie salvate sembravano essere di buona qualità e separate da angoli omogenei; dopotutto infatti, il ciclo dovrebbe impiegare lo stesso tempo per ogni operazione, a meno di malfunzionamenti dell'hardware; quindi grazie alla flat structure iniziale, avevamo già garantito una distribuzione ottimale degli scatti, che in questo caso erano 8.

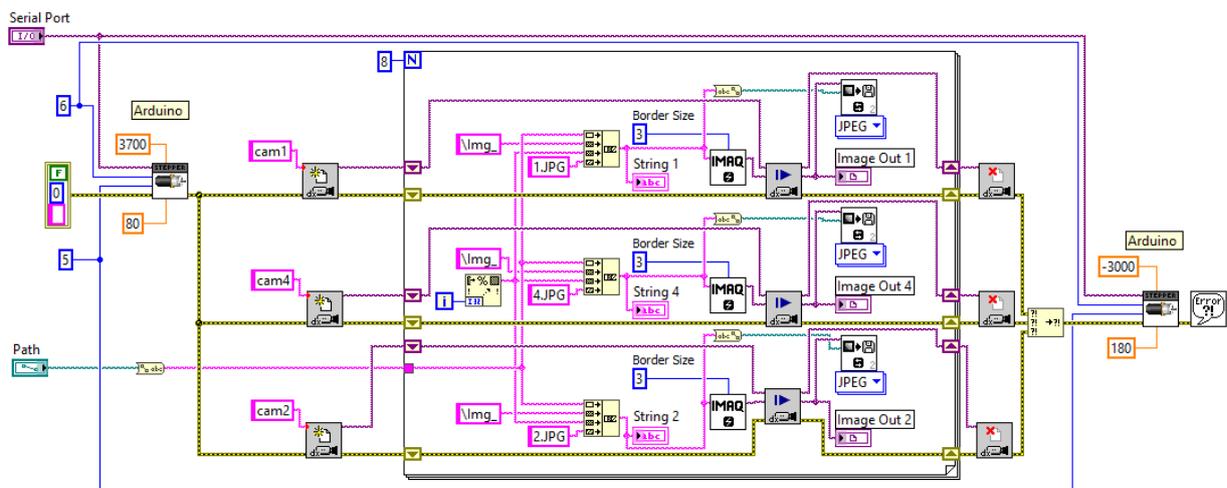


Il codice nel block diagram.

In questo caso, le costanti 3700 e 3000, che rappresentano gli angoli, così come le frequenze (80 e 180 [Hz]), differiscono tra moto di andata e moto di ritorno. Ciò non dovrebbe esser corretto, a livello logico; tuttavia assegnando angoli uguali, si era notato che, forse per l'inerzia della struttura, essa tendeva a tornare molto più indietro rispetto al punto di partenza. Ciò potrebbe anche esser dovuto al fatto che all'andata necessitiamo di una frequenza minore, per evitare vibrazioni; mentre al ritorno, per non allungare troppo i tempi del ciclo, ci permettiamo una velocità maggiore.

Programma finale

Proprio come capitato nel passaggio da una a due telecamere, anche in questo caso, aggiungendo una terza linea, il programma riscontrava delle difficoltà: a livello hardware il movimento risultava comunque fluido e liscio, ma l'inizializzazione delle 3 webcam all'interno dello stesso frame di flat structure, unita all'avviamento del motore, risultava essere un'operazione di sovraccarico per il computer, evidenziando problemi al salvataggio delle foto o allo scatto (in simultaneità). Dopo svariate prove, si è pensato di eliminare la flat structure.

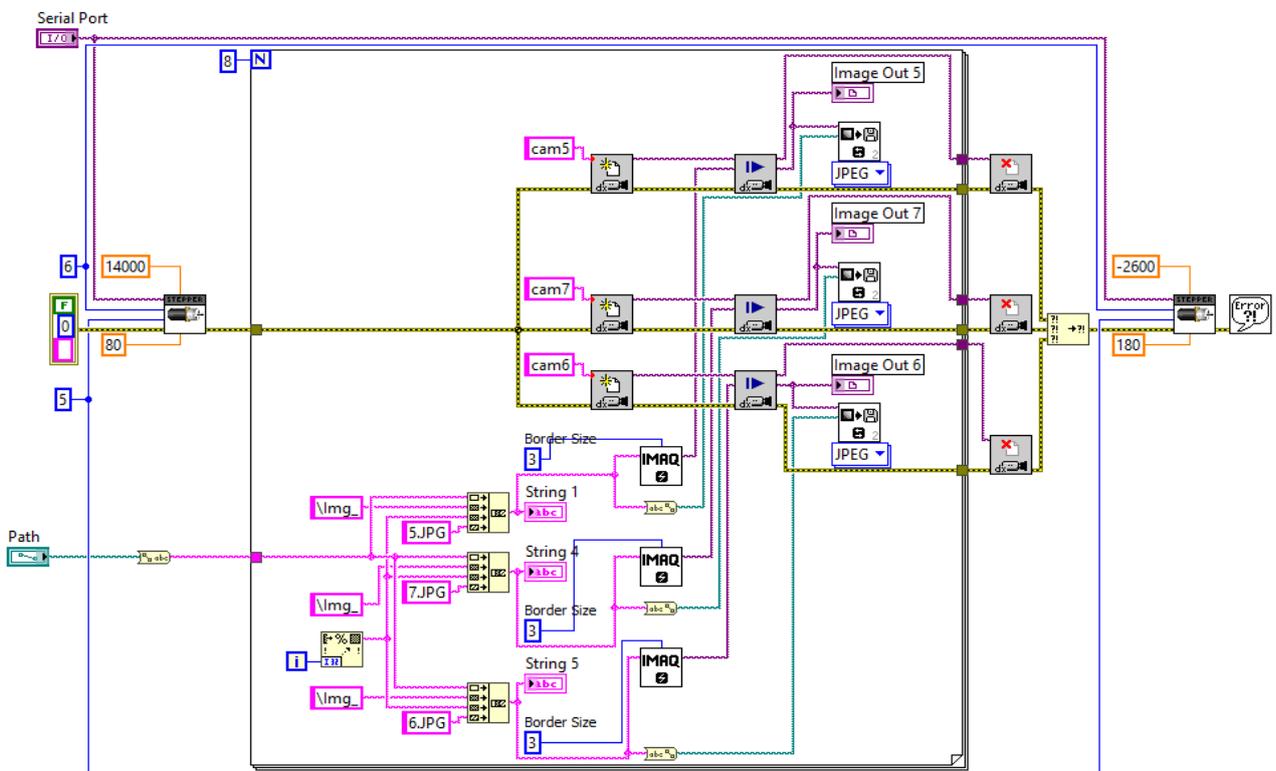


Block diagram del programma in questione.

Analizzando gli effetti di questa semplificazione del programma, si notava comunque una buona simultaneità allo scatto (per le varie camere), con le foto che venivano acquisite e salvate correttamente.

Purtroppo però, esse erano tutte concentrate in un angolo di circa 70-80°, a partire dal punto di inizio del movimento; ciò, forse, era dovuto al fatto che, senza più la flat structure, l'inizializzazione delle telecamere avveniva in maniera probabilmente non simultanea, ma sicuramente più rapida, rispetto alla partenza della slitta. Tutto ciò comportava, di conseguenza, uno svolgimento delle iterazioni del ciclo for, negli istanti iniziali di moto della struttura.

La soluzione alla quale si è giunti, è stata quella di inserire all'interno del ciclo for, anche la parte relativa all' inizializzazione delle camere, così da reiterare anche quel processo di volta in volta; così facendo, si guadagna un tempo in più, per separare gli scatti da un angolo maggiore (e sempre uguale), l'uno con l'altro. Alla sinistra del ciclo for, deve rimanere la sezione di avviamento del motore. Nella fase finale (a destra) troviamo invece la parte relativa allo spegnimento delle camere, nonché quella per il moto di ritorno della slitta, con in input un angolo negativo e una frequenza maggiore: non dovendo scattare foto infatti, possiamo permetterci di far tornare la struttura alla posizione iniziale in tempi più brevi, concedendo qualche vibrazione in più.



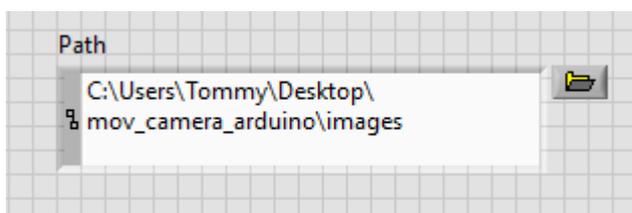
Così si presentano il block diagram e il front panel del programma finale.

Approfondimenti

Le icone dello stepper, sia quella antecedente che la successiva al ciclo for, sono collegate a tutti gli elementi interni al ciclo, e anche alla parte di spegnimento delle camere, tramite un filo giallo e nero. Quest'ultimo, che, per collegarsi alle linee di ogni telecamera, si dirama, rappresenta l'errore: esso è di fondamentale importanza, perché permette al software di esplicitare qualsiasi malfunzionamento che potrebbe verificarsi una volta che il programma è stato lanciato; specificando inoltre in quale zona è nato il problema. Alla fine del circuito, prima di collegarsi con la parte finale (la seconda icona del motore), i fili vengono ricongiunti tramite un "merge".

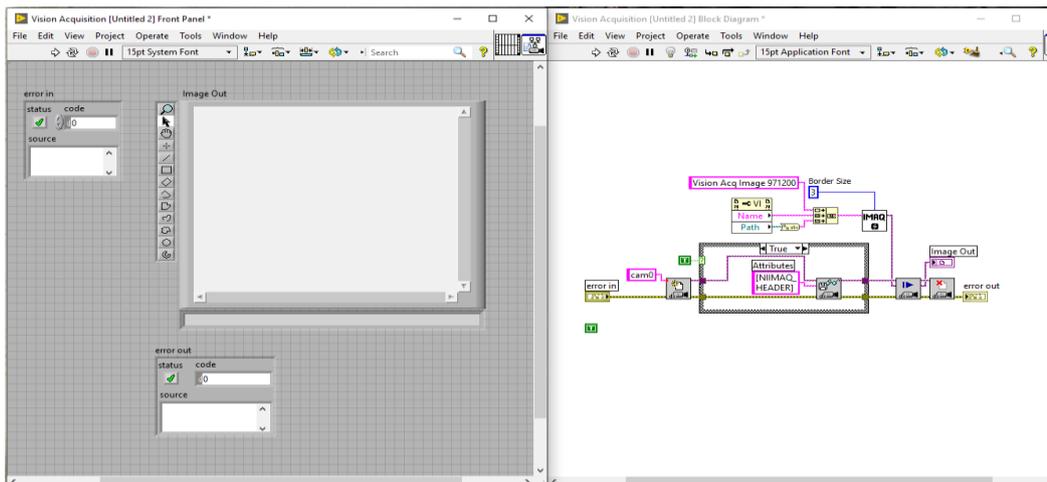
La linea dell'errore, oltre a svolgere la sua funzione principale, funge anche da linea di ordine logico e cronologico con la quale vengono effettuate le operazioni, rendendo notevolmente più semplice la lettura del codice.

Un'ulteriore nota, viene scritta per spiegare in che modo vengono salvate le foto all'interno del PC: osservando il block diagram, e in particolare la parte sinistra del circuito, prima del ciclo for, troviamo un'icona, chiamata Path. Questo elemento, dal colore celestino acqua, rappresenta il "cammino" che devono fare le foto per essere salvate dove si preferisce. Il path ha un suo corrispettivo nel front panel: qui l'utente può, molto semplicemente, scrivere in quale cartella labview deve andare a salvare le foto.



L'icona del Path nel block diagram, con la sua interfaccia nel front panel. L'icona in giallo, nell'immagine superiore, serve per convertire il dato che arriva al path.

Proseguendo ancora, è stato fondamentale l'utilizzo del sub.vi (messo a disposizione da Labview) addetto all'acquisizione di immagini: esso prende il nome di "vision acquisition", ed è formato principalmente da una fase di inizializzazione, una di funzionamento e infine dallo spegnimento delle telecamere; come si nota facilmente, ha avuto un ruolo centrale in ogni VI creato per questo progetto.



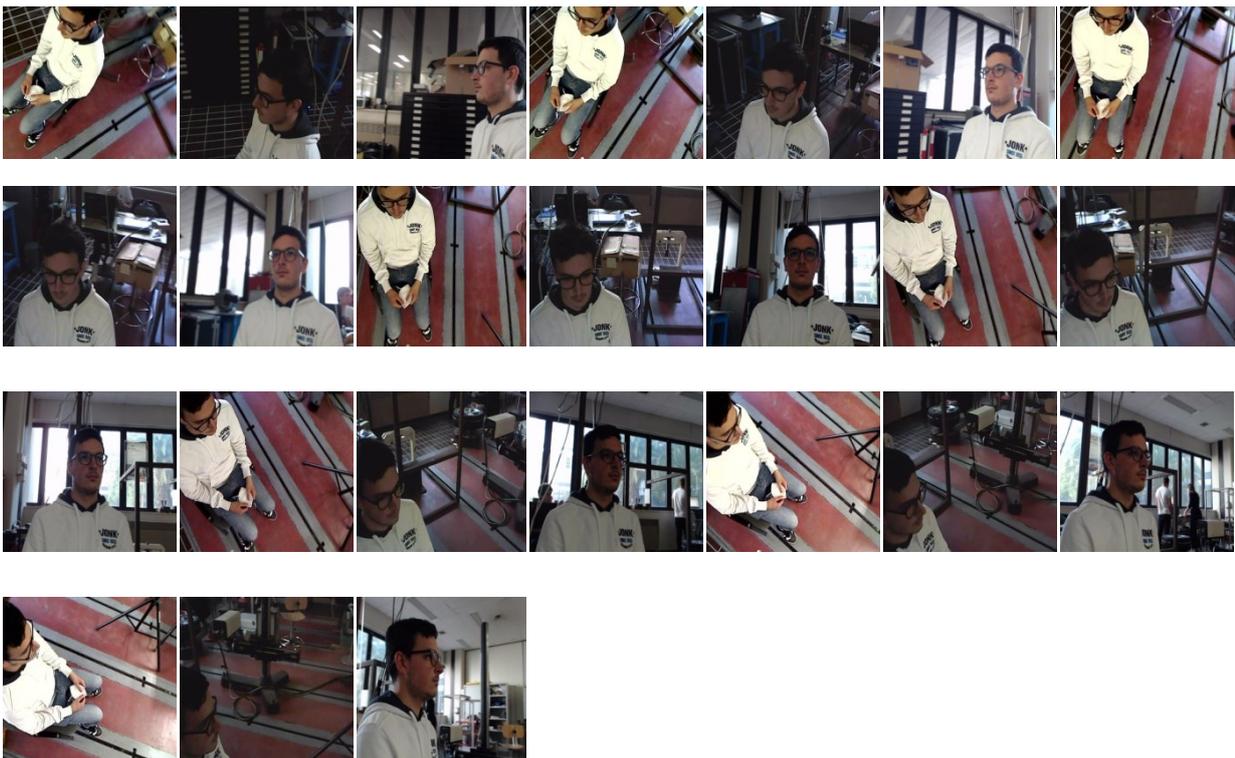
Il front panel e il block diagram del "vision acquisition".

In conclusione, LabView, nel front panel, espone anche un'interfaccia relativa alle varie camere, molto comoda per visualizzare subito le foto, man mano che vengono scattate (quando il programma è in "run"), prima che esse vengano salvate nell'apposita cartella. Inoltre, in questa sezione, sono presenti vari tool, per facilitare la modifica grafica ed estetica delle immagini acquisite.

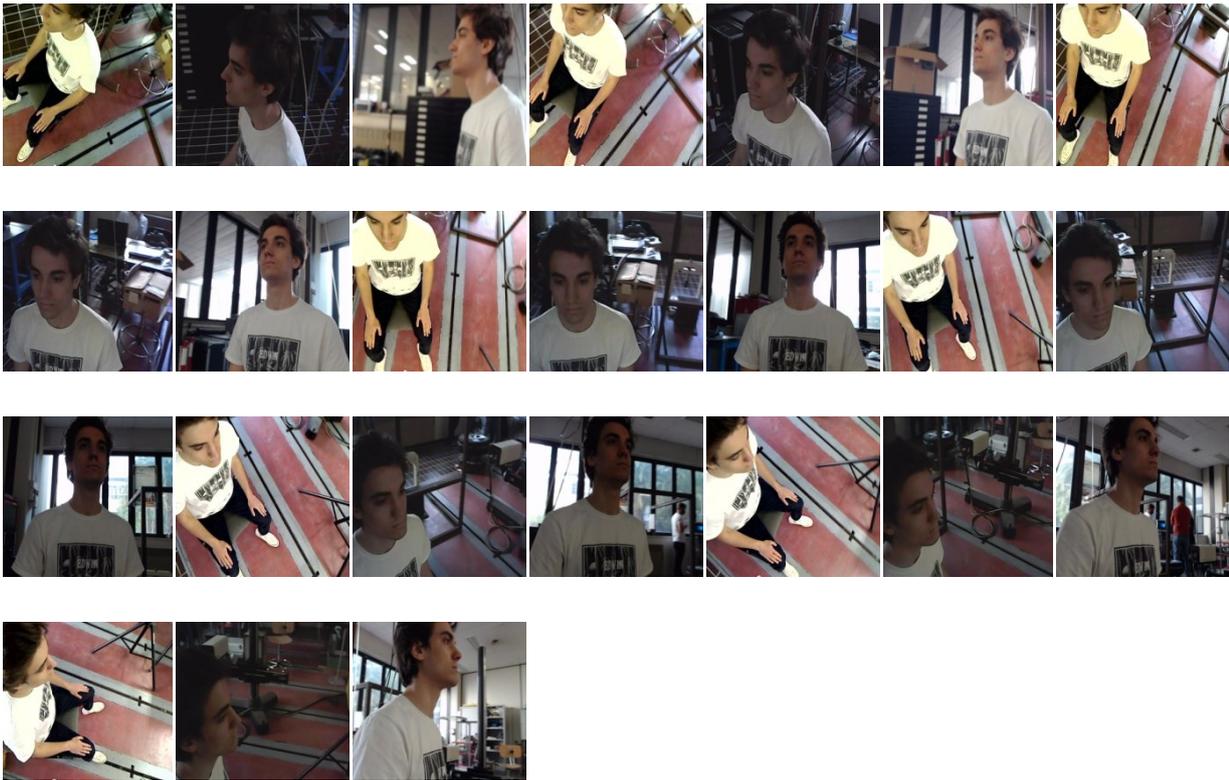
Risultati

Portato a termine il programma, è stato utile eseguire qualche prova, direttamente su noi stessi, stabilendo l'adeguatezza e l'efficienza della struttura. Siamo riusciti ad acquisire numerose foto, le quali sono state salvate correttamente nella cartella prestabilita, anche con un ordine preciso: prima si trovano tutte le foto della prima angolazione, e così via.

Il controllo del programma risulta agevole e confortevole, anche svolgendo l'operazione realmente; la comodità della struttura inoltre, con una seduta per l'individuo e l'arco mobile ben distanziato, facilita anche le operazioni di ingresso e uscita dalla zona di inquadratura delle camere. Infine, la velocità aumentata nella fase di ritorno produce qualche rumore in più, ma evita tempi morti nel caso si dovessero foto-segnalare molte persone in rapida successione.



Questi sono gli scatti ottenuti con Daniele come modello, mentre il sottoscritto era al controllo del programma. Successivamente abbiamo invertito i ruoli, effettuando ulteriori prove.



Come accennato in precedenza, questo progetto è stato realizzato con materiale messo a disposizione dall'università; l'unica spesa è stata l'acquisto del motore elettrico. Migliorando la qualità delle telecamere, nonché i metodi di fissaggio delle stesse sull'arco mobile, si riuscirebbe a lavorare con delle immagini sempre più chiare, evidenziando ogni tratto somatico, e particolarità di un volto.

Conclusioni

L'esperienza è stata davvero utile, perché ha consentito di avvicinarmi al mondo della programmazione, ed acquisire conoscenze molto importanti per un ingegnere moderno. Dopo aver portato a termine questo progetto, inoltre, sono ancora più incuriosito e appassionato dai rami della mecatronica, e della progettazione meccanica.

Insieme al collega Daniele, e anche grazie all'aiuto del Professor Castellini, siamo riusciti a far fronte a un progetto impegnativo e complesso, che ci ha permesso di ampliare notevolmente le nostre competenze, di sperimentare il lavoro di squadra e quello del progettista, in prima persona.

Ringrazio, infine, il Professor Castellini e il Dr. Alessandro Annessi, per averci seguito e aiutato in questo percorso formativo.

Bibliografia e sitografia

- NI Site: <https://www.ni.com>
- Arduino: <https://www.arduino.cc>
- Wikipedia: <https://en.wikipedia.org>