

Treball final de grau

Estudi: Grau en Enginyeria Electrònica Industrial i Automàtica

Títol: Sistema de captura automàtica de pel·lícules en format Súper 8

Document: 1. Memòria

Alumne: Gerard Cano Egea

Tutor: Daniel Alexandre Macaya Masferrer

Departament: Enginyeria Elèctrica, Electrònica i Automàtica

Àrea: Enginyeria de sistemes i automàtica

Convocatòria (mes/any): setembre / 2018

ÍNDIX

1. INTRODUCCIÓ	2
1.1. Antecedents	2
1.2. Objecte	2
1.3. Abast i especificacions	3
2. ELEMENTS I FUNCIONALITATS DE L'ESCÀNER	4
2.1. Característiques i modificacions del projector original	4
2.2. Funcionament del sincronisme	6
2.3. Control del motor pas a pas	10
2.4. Sistema òptic	14
2.4.1. Elecció de la font de llum i lents	14
2.4.2. Configuració dels paràmetres de la càmera	20
2.5. Interfície d'usuari	23
3. PROGRAMACIÓ AMB ARDUINO	26
3.1. Taules de variables	26
3.2. Estratègia de control	27
3.3. Llibreries	30
4. PROCESSAMENT D'IMATGES AMB PYTHON	32
4.1. Taula de variables	32
4.2. Flux del programa	33
4.3. Detecció de fotograma negre	36
4.4. Eliminació dels contorns	37
5. RESUM DEL PRESSUPOST	40
6. CONCLUSIONS	41
7. RELACIÓ DE DOCUMENTS	42
8. BIBLIOGRAFIA	43
9. GLOSSARI	45
A. PROGRAMA ARDUINO	46
B. PROGRAMA PYTHON	50

1. INTRODUCCIÓ

Aquest projecte conté el disseny i les especificacions tècniques de l'electrònica necessària per adaptar un projector de pel·lícules Súper 8 i convertir-lo en un escàner que capti cadascun dels fotogrames de les pel·lícules en aquest mateix format.

1.1. Antecedents

Antigament moltes filmacions domèstiques es feien en format Súper 8, un format de pel·lícula totalment analògic. Degut al format d'aquestes gravacions amb el temps es poden degradar i se'n perd el contingut, d'aquí a que es busquin sistemes per fer la conversió d'aquestes a format digital.

Avui en dia existeixen convertidors especialitzats per aquests tipus de pel·lícules, però els preus d'aquets convertidors són elevats i la seva qualitat tampoc és molt alta. Una de les opcions que s'ha plantejat la gent que no vol perdre els records gravats en aquestes pel·lícules és la filmació de la pel·lícula directament sobre la projecció a la paret, però aquests tipus de conversions domèstiques tenen una qualitat molt baixa.

1.2. Objecte

L'objectiu d'aquest projecte serà el disseny i el muntatge d'un escàner que capturi de forma automàtica cadascun dels fotogrames que conté una pel·lícula en format Súper 8.

El disseny inclou les diferents modificacions mecàniques que s'apliquen sobre l'antic projector, així com els diferents components electrònics que s'encarreguen de governar el sistema i automatitzar el procés.

L'escàner també permetrà la intervenció de l'usuari, aconseguint així un control semiautomàtic sobre les diferents accions instal·lades i permeten definir característiques de la pel·lícula que es vol capturar.

Tots aquests elements es controlaran gràcies al codis informàtics que s'inclouen en el disseny i que defineixen la lògica del funcionament de l'escàner.

La finalitat és obtenir un escàner que digitalitzi tots els fotogrames i els guardi en una carpeta de l'ordinador, de manera que estiguin llestos per a poder fer després el muntatge de la pel·lícula amb un programa d'edició.

1.3. Abast i especificacions

L'abast d'aquest projecte inclou la documentació de tots els aspectes necessaris per a la realització del control de l'escàner, com el disseny del circuit electrònic i de la PCB, el programa que portarà el microcontrolador integrat que governarà el sistema, així com la selecció dels components adequats per a fer funcionar el sistema.

L'escàner ha de permetre detectar el moment exacta en el que un fotograma passa per la zona de projecció i en aquell moment disparar la càmera i automàticament passar-la en una carpeta de l'ordinador.

La PCB que controla l'escàner s'alimenta amb una font externa i funciona sota les ordres d'un Arduino Nano. El microcontrolador conté el programa de control que s'ha dissenyat carregat prèviament. A més, s'ha dissenyat un programa en codi Python que es comunica amb l'Arduino per analitzar cadascun dels fotogrames i determinar quan s'ha acabat la pel·lícula, i un cop s'han digitalitzat tots els fotogrames procedeix a editar les imatges per a deixar-les preparades per fer-ne el muntatge posteriorment.

L'escàner permet a través d'un selector controlar el sentit del passat de la pel·lícula així com seleccionar el mode de captura automàtica de la pel·lícula. A més, disposa d'un pantalla LCD i un botó que serveixen com a petita interfície gràfica. Aquesta interfície gràfica permet definir els metres de la pel·lícula que es vol convertir i al mateix temps proporcionar informació durant el procés de captura.

2. ELEMENTS I FUNCIONALITATS DE L'ESCÀNER

En aquest capítol es descriurà el sistema que s'ha dissenyat per dotar l'escàner de les funcions que s'han descrit en el capítol anterior, i el funcionament de cadascuna de les parts d'aquest sistema per justificar la utilització dels elements triats. Al mateix temps, també es descriuran les característiques del projector del que s'ha partit justificant quines parts del mateix s'han conservat.

2.1. Característiques i modificacions del projector original

En aquest cas s'ha partit d'un projector Sankyo Dualux 1000. Aquest projector es caracteritza principalment per poder projectar tant pel·lícules en format 8 mil·límetres estàndard com pel·lícules en format Súper 8 a elecció de l'usuari, a través d'una pestanya vermella que canvia l'àrea de projecció. Addicionalment també disposa d'una pestanya negra que permet ajustar manualment la posició del fotograma per assegurar que s'estigui projectant un fotograma sencer i no una part d'un i una altra part del següent. Aquests dos mecanismes s'han mantingut, tot i que el primer es manté sempre a la posició de Súper 8 i el segon s'utilitza com un ajust addicional a l'hora de fer el sincronisme.

Aquest projector també disposa d'un selector rotatiu de 5 posicions que permet seleccionar entre projectar la pel·lícula endavant, projectar-la endarrere, rebobinar o simplement passar-la endavant sense projectar. Aquest selector s'ha mantingut degut a que no només actua modificant senyals elèctrics sinó que també canvia posicions mecàniques en funció de si es vol anar endavant o de si es rebobina. En aquest cas s'ha ajustat elèctricament per a què funcioni com un selector de quatre posicions i et permeti triar entre tirar endavant, rebobinar, començar el procés automàtic de captura fotograma a fotograma o un mode OFF en el que està el motor parat. La cinquena posició addicional no se li ha donat cap ús. Els modes de funcionament descrits es poden observar a la Figura 1:

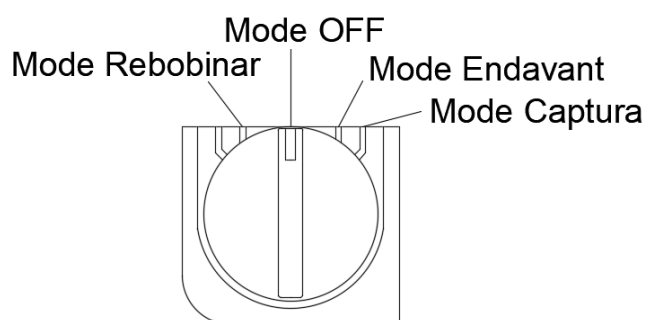


Figura 1. Modes de funcionament en funció del selector

A nivell òptic, el projector per defecte ve amb una lent inversora que es situa a menys de 5 mil·límetres del fotograma, tot i que aquesta distància es pot variar manualment per acabar d'ajustar l'enfoc. La idea és que com que el fotograma passa invertit per la zona de projecció aquest lent inverteix la imatge per poder veure el fotograma correctament i al mateix temps serveix per augmentar la imatge per a què sigui projectable a una certa distància. Aquesta lent s'ha substituït per una càmera reflex i pels diferents elements òptics que es detallen a l'apartat d'òptica.

Pel que fa a la llum que projecta la imatge prèvia a la lent, el projector per defecte té una bombeta que treballa a 8 volts i consumeix 50 watts. Aquesta s'ha substituït per una font de llum LED degut a que produïa molta calor i a diferència de la funció de projectar on la velocitat en que passen els fotogrames és molt elevada, en el procés de captura que s'ha desenvolupat la velocitat és més reduïda i per tant cada fotograma està més estona exposat a aquesta font de calor, poden arribar a posar en risc la integritat del fotograma.

Internament el projector funciona a través d'un motor d'inducció monofàsic alimentat a través d'un autotransformador sobre el qual l'usuari pot manualment actuar per modificar la velocitat del motor. A més, disposa del cablejat suficient per a través del selector realitzar les maniobres tan com per fer els canvis de gir del motor com per alimentar la bombeta. A la Figura 2 es pot observar com estava compost originalment el projector i els elements més destacats d'aquest:

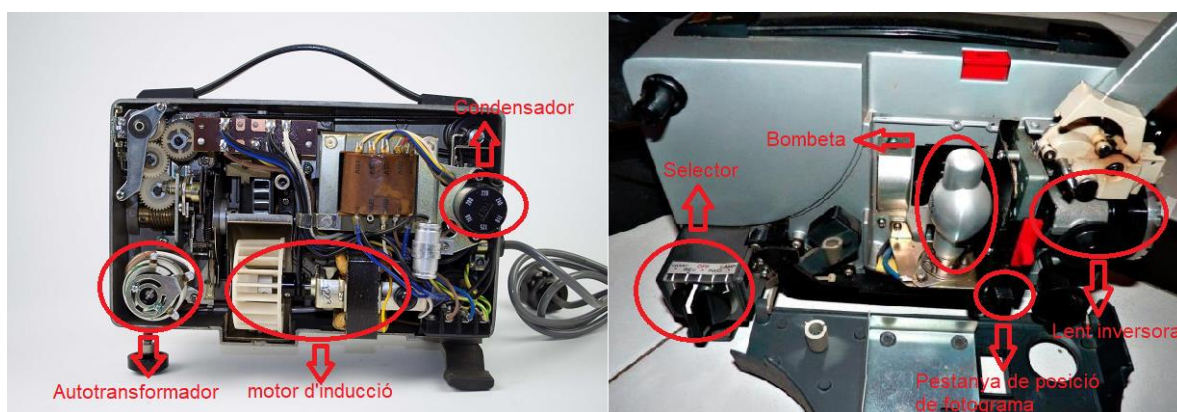


Figura 2. Distribució interna del projector original

Tot aquest sistema elèctric s'ha eliminat i s'ha mantingut únicament els elements que permeten la transmissió mecànica. El motor d'inducció s'ha substituït per un motor pas a pas i s'ha afegit un sensor inductiu que servirà per detectar el moment exacte en que es captura el fotograma, mentre que tots els elements que serveixen per les maniobres s'han substituït

per una PCB que servirà com a placa de control amb un microcontrolador com a eix central, en aquest cas una arduino nano. Aquesta placa va integrada a la tapa de l'escàner i s'alimenta d'una banda per una font d'alimentació externa de 12 volts i 2 ampers, i al mateix permet connexió a través del connector USB de l'arduino nano per comunicar-se amb un ordinador que executa un programa en codi python.

Adicionalment s'ha dissenyat un suport per el sensor inductiu que ajudarà en el sincronisme i una politja per fer la transmissió del motor, s'ha afegit un suport per mantenir fixa la càmera i els demés elements òptics, i finalment s'ha mecanitzat la tapa per poder adequar-se als nous components de que disposa l'escàner. Tots aquests elements es poden veure a la Figura 3 que representa la vista posterior de l'escàner obtingut a través de les modificacions.

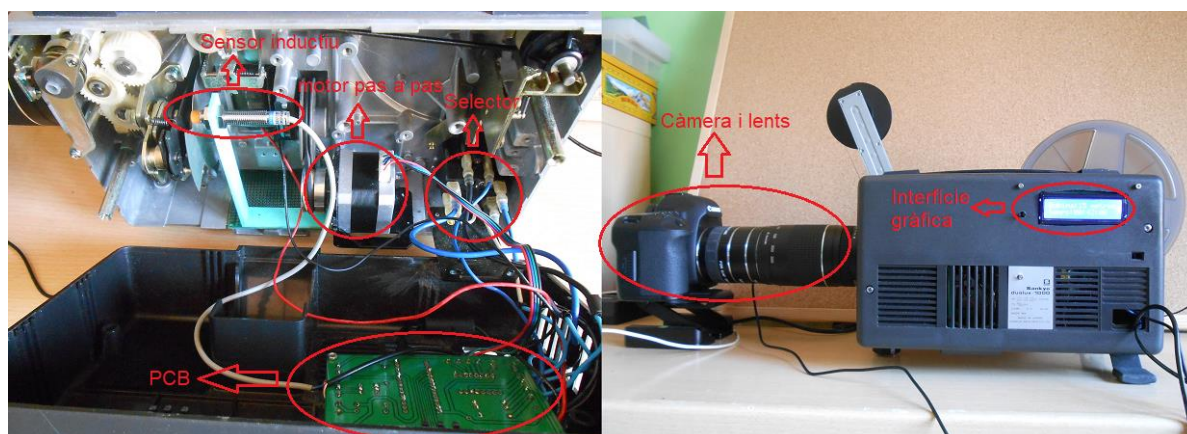


Figura 3. Distribució interna i vista posterior de l'escàner

2.2. Funcionament del sincronisme

El sincronisme es basa en detectar el moment exacte en que un fotograma nou es troba a l'àrea de projecció per disparar la càmera. El problema aquí és que no es té accés just en el punt on el fotograma es captura, i considerant les dimensions dels fotogrames no s'ha trobat una manera clara i òptima per detectar de forma real quan passa el següent fotograma. És per això que s'ha optat per un motor pas a pas degut a la seva precisió.

Aquí es quan entra en joc la pestanya negra que ja portava per defecte el projector i que servia per fixar la posició del fotograma a l'àrea de projecció, tal com es comentava a l'apartat anterior. La idea és aquesta peça fixa la posició del fotograma fins al punt que per molts passos que faci el motor pas a pas sempre mantindrà aquella posició, és a dir, que si es visualitzava un fotograma sencer es mantindrà així per mes voltes que faci el motor i si

estava fixat una posició on es veia mig fotograma del primer i mig del segon també es mantindrà amb aquesta posició encara que els fotogrames canviïn. Això és important ja que indica que encara que el motor giri els passos per fer únicament mig fotograma, el mecanisme es mouria però el fotograma no faria cap moviment fins assolir els passos suficients per passar el fotograma complet. Aquest sistema mecànic simplifica el sincronisme ja que gràcies a ell només cal saber els passos necessaris per a què passi únicament un fotograma, sense temor a que pugui quedar un fotograma a mig camí. Això requereix doncs d'un petit ajust previ al sistema de captura automàtica on cal ajustar aquesta pestanya per assegurar que es visualitza un fotograma sencer. Això es pot fer gràcies al selector triant el mode de passar la pel·lícula endavant sense capturar i a través de la càmera es pot visualitzar si el fotograma està ben situat.

Tal com s'ha comentat, el primer que es va fer va ser trobar experimentalment els passos amb els que passava un sol fotograma, però aquí sorgia un altre problema i és el fet que el projector just entre la font de llum i el fotograma té una peça giratòria metàl·lica acoblada sobre l'eix principal i que gira solidaria amb aquest, i en certes situacions a l'acabar els passos del motor i canviar de fotograma aquesta peça quedava tapant la llum i la càmera disparava un fotograma en negre. Això és degut a que tot i que el motor pas a pas no té pràcticament inèrcia, la politja motora d'aquest transmet el moviment a través d'una corretja llisa que produeix un cert lliscament, per tant encara que es fixin els mateixos passos per cada fotograma aquesta peça a vegades es podia trobar interposant-se amb la llum.

Aquesta peça està formada per tres aspes i la seva funció original en el projector era tancar la llum just en el moment en que s'estava produint un canvi de fotograma, de manera que durant la projecció de la pel·lícula no s'apreciés el canvi i al projectar-se a tanta velocitat tampoc es podia apreciar el moment en que la llum no arribava al fotograma. Però no només tenia aquesta funció, sinó que resulta que en última instància aquesta peça és la que li dona l'impuls al fotograma per fer el canvi, raó per la qual aquesta peça s'ha mantingut. S'ha pogut comprovar que quan per davant del fotograma hi passen tres vegades una aspa es produeix el canvi de fotograma. En base això, en comptes de fixar un nombre de passos pel motor, el que s'ha fet és afegir un sensor inductiu de proximitat a la posició idònia de manera que quan aquest sensor no està detectant l'aspa, s'assegura que l'aspa complementària que es trobaria prop del fotograma no està tapant la llum. Aquest mecanisme es pot veure representat a la Figura 4, on s'aprecia la relació de la posició del sensor amb l'àrea de projecció del fotograma per poder determinar el moment en que es pot disparar la càmera sense perdre la llum.

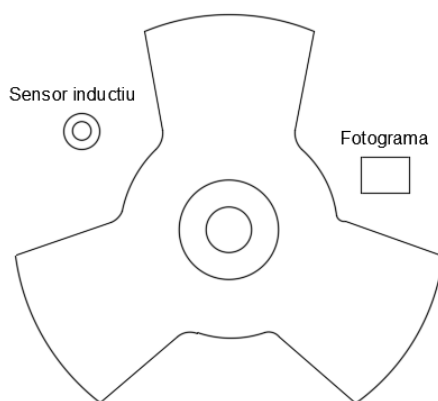


Figura 4. Esquema de detecció de fotograma

La idea doncs es fer que el motor vagi fent passos indefinidament fins que el sensor inductiu, que en aquest cas és el model LJ8 A3-2-Z/POR i detecta a 2 mil·límetres, hagi detectat i deixat de detectar tres vegades, el que és equivalent a dir que per el fotograma han passat tres aspes i per tant s'haurà produït un canvi de fotograma, a més del fet que es podrà assegurar que el fotograma mai li quedarà la llum tapada abans de que es dispari la càmera.

Un cop establert el moment exacte en que el fotograma ha canviat i es pot capturar, l'únic que cal és disparar la càmera. Per disparar la càmera es pot fer a través de software i fer-ho per temporització, o com és el cas d'aquest projecte també es pot fer per hardware a través d'un disparador extern per assegurar el moment exacte en que es pot capturar el fotograma. En cas de fer-ho amb disparador cada model de càmera té el seu propi tipus de connector, en el cas de la que s'ha utilitzat en aquest projecte que és una Canon EOS Mark II, utilitza un disparador RS-80N3. Aquest disparador ve normalment amb el seu botó per fer el dispar però el que s'ha fet es retallar-lo per mantenir simplement el connector que va a la càmera i poder automatitzar el dispar. Com es pot veure a la Figura 5 el disparador està compost de tres cables, un es el comú o massa, l'altre és el cable de dispar i l'últim és per enfocar.

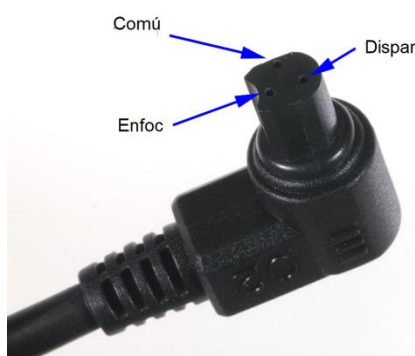


Figura 5. Pins del disparador

El funcionament es basa en que per provocar tan el dispar com l'enfoc cal curtcircuitar el cable corresponent amb el comú. En aquest cas només s'ha optat per utilitzar el del disparador ja que l'enfoc només serveix per veure, sense haver de tocar la càmera, com es veuria la foto un cop s'hagi fet sota les condicions d'enfoc prefixades, ja sigui amb enfoc manual o automàtic. Però tal com es comentarà a l'apartat d'òptica, a través de l'ordinador també es podrà observar com es veuria la foto enfocada sense haver de tocar la càmera.

Així doncs, per poder provocar el curtcircuit entre el cable de dispar i el comú s'ha optat per utilitzar l'integrat 4N35, el qual tal com es pot veure a la Figura 6 internament treballa com un optoacoblador el que ens permet curtcircuitar els dos cables de la càmera assegurant al mateix temps que la càmera està separada de qualsevol voltatge provinent de la PCB que la pogués malmetre.

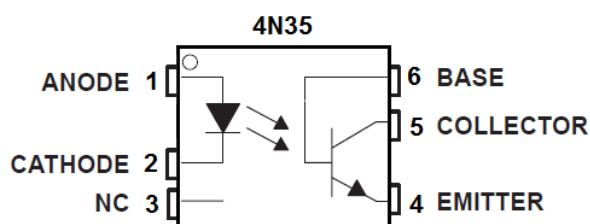


Figura 6. Esquemàtic intern del 4N35

L'únic que es necessita és una resistència connectada a una de les sortides del microcontrolador per alimentar el LED intern del 4N35, que en aquest cas s'ha utilitzat una de 220 ohms, i connectar els cables de comú i de dispar sense importar l'ordre als pins que representen el col·lector i l'emissor del fototransistor.

D'aquesta manera simplement activant la corresponent sortida del microcontrolador en el moment que s'ha produït un canvi de fotograma es podria anar capturant automàticament cadascun dels fotogrames. L'únic a tenir en compte és el fet que la càmera requereix d'un cert temps entre dispar i dispar. En aquest s'ha estipulat un temps de 200 mil·lisegons entre dispar i dispar.

Pel que fa a guardar les imatges, aquestes es podrien guardar a la memòria SD de la càmera i passar-les posteriorment a un ordinador. Però com ja s'ha comentat aquest projecte inclou un programa en python per processar les imatges, per tant, cal que aquestes es guardin directament a una carpeta de l'ordinador. Per fer això, es fa ús del programa propi de les càmeres Canon que és el Canon EOS Utility. Aquest programa entre d'altres

coses ens permet triar una carpeta on guardar les imatges de manera que simplement a través d'un USB connectat entre càmera i ordinador podem fer que cada foto feta per la càmera es guardi automàticament en una carpeta per ser processada.

L'últim detall a tenir en compte en el sincronisme és el mètode per parar el dispar de la càmera un cop la pel·lícula ja s'ha acabat. Lo ideal seria tenir algun sensor que indiqués quan ja no passa cap més fotograma per la zona de projecció, però tal com s'ha comentat anteriorment no s'hi té accés i el sincronisme es basa en saber cada quan ha de disparar però no es pot assegurar que està passant realment un fotograma. Per corregir això s'utilitza també el programa de python que el que fa és analitzar cada fotograma per veure si n'hi ha algun que sigui completament negre, i si aquest és el cas envia una senyal per port sèrie a través d'un USB a l'arduino per indicar-li que pari el motor i el dispar de la càmera. El fet que analitzi si el fotograma és completament negre és degut a que ja el projector original tenia una peça que mecànicament quan no hi ha cap pel·lícula passant baixa i s'interposa entre llum i fotograma.

2.3. Control del motor pas a pas

El motor pas a pas pel que s'ha optat és un Nema 17 model 17HD34008-22B. Aquest té un parell màxim de 12mN·m, un corrent nominal de fase de 1,5 amperers i una precisió de 1.8 graus per pas amb 200 passos per volta. Com ja s'ha comentat s'ha optat per un motor pas a pas degut a la seva precisió i a la poca inèrcia que ofereix, de manera que davant de la detecció del sensor inductiu aquest es para pràcticament a l'instant. Concretament aquest motor és del tipus bipolar, el que significa que està format per dos bobines i quatre cables tal com mostra l'esquema de la Figura 7.

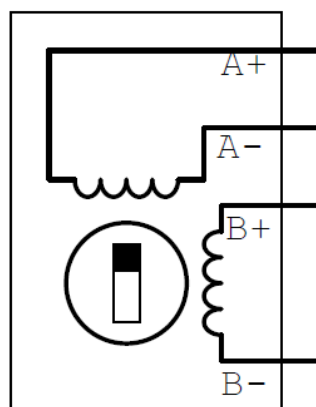


Figura 7. Esquemàtic intern del motor Nema 17

En el cas dels motors bipolars es requereix una seqüència d'activació de les bobines concreta per poder realitzar cadascun dels passos. Aquesta seqüència es pot apreciar a la Taula 1.

Pas	A+	A-	B+	B-
1	1	0	0	1
2	1	0	1	0
3	0	1	1	0
4	0	1	0	1
5	1	0	0	1
6	1	0	1	0
7	0	1	1	0
8	0	1	0	1

Taula 1. Seqüència d'activació de les bobines

Aquesta seqüència representa l'ordre amb que s'ha d'anar activant les bobines per a què el motor giri, de manera que al arribar al vuitè pas de la seqüència s'ha de tornar a continuar amb el primer fins a assolir els passos que es volen. Si es segueix la seqüència el motor gira en un sentit mentre que si fa la seqüència en sentit contrari el gir del motor s'inverteix. Com es pot veure, en aquesta seqüència les bobines reben corrent en ambdós sentits el que fa necessari un sistema com un pont en H per cadascun dels debanats. A això cal afegir-li el fet que les bobines del motor necessiten tensions superiors a les que podrien suportar pel seu corrent nominal. Per exemple, en el cas del nostre motor que té un corrent nominal de 1,5 amperes i una resistència de 2,3 ohms per fase, amb un voltatge de fase de 3,45 volts segons la llei d'Ohm tindríem el corrent nominal, però amb aquest valor de tensió el motor ni es mouria. Per a què el motor es mogui cal aplicar una tensió superior com són els 12 volts de la font que s'ha utilitzat, ara bé, amb aquesta tensió es produiria un corrent molt superior al nominal del motor i el cremaríem. Per solucionar tots aquests problemes i poder controlar el motor cal una electrònica de potència addicional. En el mercat existeixen diferents drivers que inclouen aquesta electrònica com és el cas del controlador DRV8825 que és pel que s'ha optat.

El controlador DRV8825 és un driver dissenyat expressament per controlar motors pas a pas. Inclou dos ponts en H, un per cada bobina, permeten així la inversió de sentit de les bobines. També incorpora reguladors d'intensitats que limiten la intensitat de fase de les bobines permeten així alimentar el motor a 12 volts. La idea és que el limitador de corrent interromp el senyal proporcionant una senyal PWM de manera que el valor mitjà de la intensitat que travessa la bobina no sigui mai superior a la nominal permesa pel motor.

Aquest valor límit d'intensitat s'ha de fixar manualment. Per fer-ho, el controlador DRV8825 disposa d'un potenciòmetre amb el que es pot regular la intensitat. Es pot estimar el valor de la intensitat fixada en funció d'un valor de referència que equival a la tensió entre el potenciòmetre i massa, tal com marca l'Equació 1:

$$I_{\max} = 2 \times V_{\text{ref}} \quad (\text{Eq. 1})$$

I_{\max} = Intensitat màxima fixada pel limitador en ampers.

V_{ref} = Voltatge entre potenciòmetre i massa en volts.

La intensitat límit s'ha de fixar procurant que aquesta sigui inferior a la màxima permesa pel driver, que en aquest cas és de 1,5 A, i en un valor proper al nominal del motor de manera que aquest pugui treballar a altes velocitats sense perdre molt de parell. Així doncs, segons l'Equació 1 i considerant que es vol limitar el corrent a 1,2 ampers es fixa un voltatge de referència de 0,6 volts. Ara bé, cal tenir en compte també la configuració dels passos triada. El DRV8825 permet una funció coneguda com microstepping que consisteix en obtenir passos inferiors al pas nominal del motor pas a pas, és a dir, en el cas del nostre motor que necessita realitzar 200 passos per fer una volta i per tant per cada pas realitza un gir de 1,8 graus, el DRV8825 permet augmentar el nombre de passos per volta obtenint així més resolució per pas. Concretament permet aconseguir fins a una resolució de 1/32 de pas. Per triar la resolució aplicada cal treballar sobre tres dels pins del driver que són M0, M1 i M2. En funció dels seus valors lògics es determina el mode de microstepping tal com marca la Taula 2:

Resolució	M0	M1	M2
Full step	LOW	LOW	LOW
1/2 step	HIGH	LOW	LOW
1/4 step	LOW	HIGH	LOW
1/8 step	HIGH	HIGH	LOW
1/16 step	LOW	LOW	HIGH
1/32 step	HIGH	LOW	HIGH

Taula 2. Configuració del microstepping

La importància de conèixer la configuració de la resolució dels passos recau en que en funció d'aquesta el corrent limitat varia. Per exemple, si s'utilitza algun tipus de microstepping el corrent màxim que arribarà a passar per la bobina en algun pas serà del 100% de la intensitat calculada a l'Equació 1, mentre que si s'utilitza el mode full step la

intensitat que arribarà a passar per les bobines en algun pas serà del 71% de la intensitat fixada en el limitador. En aquest cas la precisió que es requereix no és tan alta com per requerir el microstepping, per tant s'ha configurat el mode full step, de manera que el nombre de passos per fer una volta es mantenen al valor original del motor. Això comporta que al calcular l'Equació 1 cal tenir en compte que ara el corrent màxim que passa per les bobines serà un 71% del valor del limitador, el que permet fixar un voltatge de referència més gran. Tenint en compta això s'ha fixat un voltatge de referència de 0,85 volts per obtenir els 1,2 amperes desitjats. Cal tenir en compte que aquest mètode del voltatge de referència és una aproximació per no passar-nos del corrent límit la primera vegada que es connecta el motor, de manera que per acabar de fer l'ajust fi s'ha utilitzat un amperímetre per assegurar que es treballa amb la intensitat desitjada.

Una altre dels avantatges del DRV8825 és que ens permet a través de només dos senyals executar la seqüència dels passos, un per marcar el sentit de gir i l'altre per indicar l'execució del pas, evitant així haver d'anar executant la seqüència mostrada a la Taula 1.

Així doncs l'esquema de connexió entre el conjunt de driver, motor i de l'arduino seria el mostrat a la Figura 8.

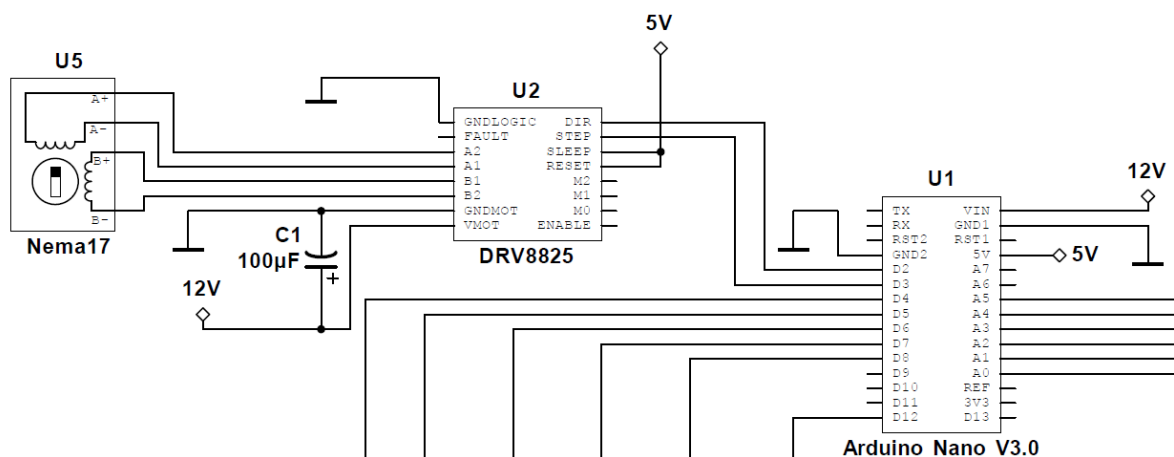


Figura 8. Connexió DRV8825, motor i microcontrolador

Com es pot apreciar el DRV88250 requereix de dos tensions d'alimentació, una de 5 volts pels senyals de control i els 12 volts per poder induir les bobines del motor. L'alimentació del motor requereix d'un condensador de 100µF pel desacoblament i les masses de les dues tensions convé que estiguin amb una massa comuna. Com ja s'ha comentat s'ha deixat sense connectar els pins M0, M1 i M2 per utilitzar el mode full step, mentre que els pins sleep i

reset s'han connectat directament a 5 volts ja que aquests no s'utilitzen i s'activen al detectar un zero.

Pel que fa al control de la velocitat del motor, aquesta s'ajusta per programa fixant el temps entre pas i pas. Sempre s'ha intentat fixar la velocitat màxima permesa pel motor, però cal tenir en compte que com més alta és la seva velocitat menys parell pot oferir. En el cas dels modes endavant i endarrere s'ha fixat un temps de 0,9 mil·lisegons, mentre que en el mode captura és de 1,45 mil·lisegons.

2.4. Sistema òptic

L'òptica és una de les parts més fonamentals d'aquest projecte ja que determina la qualitat amb la que es capta cada fotograma i per tant la qualitat final de la pel·lícula. L'objectiu original en aquest àmbit era que amb el sistema òptic que s'utilitzés s'aconegués que el fotograma ocupés el total de la foto feta per la càmera, el que implicaria que es captaria amb la qualitat màxima permesa per la càmera, però en aquest sentit una de les problemàtiques que s'ha trobat és que la pròpia estructura del projector del que s'ha partit no permet acostar la càmera fins a més d'uns 6 centímetres del fotograma, amb lo qual considerant les dimensions del fotograma fa impossible ocupar el total de la mida de la foto a aquesta distància, d'aquí a que s'hagi hagut de dissenyar un programa en python que de forma automàtica editi les imatges a fi de retallar el contorn innecessari de les imatges.

2.4.1. Elecció de la font de llum i lents

Com a tot sistema òptic una de les parts més importants és la font de llum que s'utilitza per visualitzar l'objecte. El tenir una bona llum sempre simplifica el sistema de lents necessàries, a més del fet que òpticament sempre és més fàcil corregir l'excés de llum que la seva falta.

Com ja s'ha comentat a l'apartat de modificacions del projector original, s'ha eliminat la bombeta que portava per defecte degut a la calor que generava i s'ha substituït per una font LED. Concretament s'ha substituït per una bombeta LED d'1 W i 12 V. A l'hora de triar aquesta font de llum un dels paràmetres que s'ha tingut en compte és la temperatura del color de la llum. Aquest paràmetre que es quantifica en Kelvins indica el tipus de llum blanca que genera, especificant si es tracta d'una llum blanca calenta o freda, que és el mateix a dir que té una tonalitat més rogenca o més blavosa. Això es pot veure representat a la Figura 9.

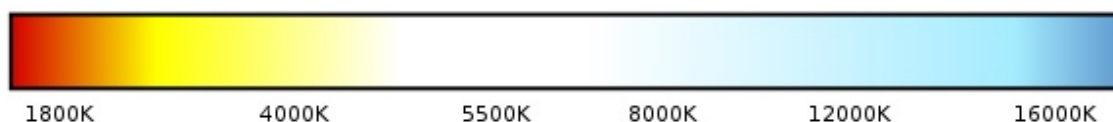


Figura 9. Escala de temperatura de color de la llum

En base això s'ha optat per utilitzar una font de llum que tingui una temperatura de color lo més neutre possible, és a dir que sigui blanc pur, d'aquesta manera s'assegura que els colors captats per la càmera seran lo més fidels possibles als originals del fotograma. Concretament aquesta bombeta que s'ha triat té una temperatura de color de 6000 K, el que la fa idònia per la raó que s'ha descrit. També cal remarcar que una altra necessitat a nivell de llum és que aquesta es reparteixi homogèniament per tot el fotograma, això depèn de la capacitat lumínica de la font de llum i també de l'angle de dispersió d'aquest. En aquest cas el LED que s'ha triat té una intensitat lumínica de 100 lúmens que es bastant alta i al mateix temps l'angle del feix de llum és de 60°, assegurant que il·luminarà el total del fotograma.

Referent al sistema de lents que s'ha utilitzat, ja s'ha comentat que s'ha substituït la lent inversora que portava per defecte i s'ha optat per utilitzar una càmera reflex model Canon EOS 5D Mark II. La lent original permetia una gran ampliació degut a que es trobava molt a prop del fotograma, però al intentar fotografiar ja fos amb una càmera reflex o una digital a través de la lent, tot i poder-se apreciar una gran ampliació també hi havia l'inconvenient de que la càmera capturava la forma esfèrica exterior de la lent, provocant que les cantonades del fotograma es perdessin, raó per la qual s'ha descartat. D'altre banda, s'ha optat per una càmera reflex degut a que permet acoblar-hi diferents tipus de lents en funció de la situació, el que la fa molt flexible, a més del fet que a diferència de les càmeres digitals permet una configuració de la mateixa més precisa. Un exemple d'això és el zoom, on en el cas de les càmeres digitals el zoom consisteixen en estirar els píxels amb la conseqüent pèrdua de qualitat, mentre que en les reflex el zoom depèn del objectiu però sempre manté la resolució original. Pel que fa al model concret de càmera que s'ha utilitzat, aquesta és caracteritzada per tenir una mida de sensor de 36x24 mm, que en píxels és 5616 x 3744, el que equival a dir que qualsevol foto feta per aquesta càmera tindrà les mides equivalents a les del seu sensor en píxels. En base a això, s'ha determinat l'objectiu i el conjunt de lents necessàries per captar el fotograma amb la màxima resolució possible.

A l'hora de triar un objectiu un dels paràmetres més importants és la distància focal. La distància focal és un paràmetre que expressa la distància en mil·límetres entre el sensor de la càmera(pla focal) i el centre òptic de la lent de l'objectiu, que és el punt on els rajos de

Llum es creuen i es dirigeixen al sensor de la càmera. A nivell pràctic el valor de la distància focal determina el zoom òptic de l'objectiu així com el seu angle de visió, ja que per focals grans l'angle de visió és més petit, el que ens permet apreciar detalls amplificats de l'objecte a fotografiar, mentre que amb focals curtes l'angle de visió és molt ampli de manera que es podrà apreciar el global de l'objecte a fotografiar però no tan amplificat. Per poder calcular la distància focal de l'objectiu cal tenir present l'esquema de la Figura 10 que representa com la llum travessa l'objecte que es vol fotografiar i passa per la lent del objectiu fins a arribar al sensor de la càmera.

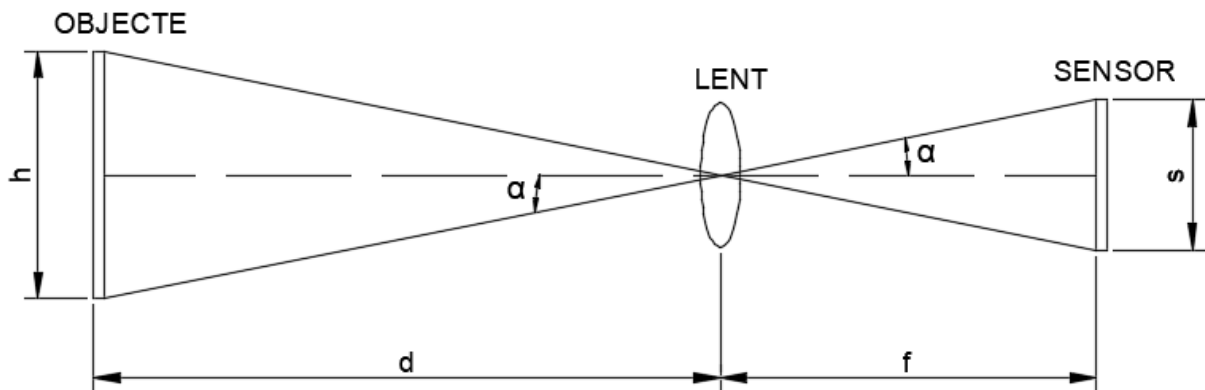


Figura 10. Esquema de la llum en la captura del fotograma

En base a l'esquema de la Figura 10 i amb l'ús de trigonometria es poden obtenir l'Equació 2 i l'Equació 3 que serviran per trobar la focal necessària considerant que es busca obtenir que el fotograma ocupi el 100% de la mida del sensor de la càmera.

$$\alpha = \arctan\left(\frac{H}{2D}\right) \quad (\text{Eq. 2})$$

$$F = \frac{S}{2\tan\alpha} \quad (\text{Eq. 3})$$

α = Angle de visió en funció de la mida del sensor i la distància focal.

H = altura de l'objecte que es vol capturar en mil·límetres.

D = Distància des de l'objectiu fins a l'objecte a capturar en mil·límetres.

S = Altura del sensor de la càmera en mil·límetres.

F = Distància focal en mil·límetres.

Per poder aplicar aquestes equacions cal tenir en compte les dimensions dels fotogrames d'una pel·lícula en format Súper 8. Aquestes es poden apreciar a la Figura 11.

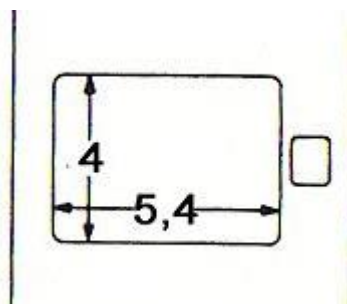


Figura 11. Dimensions d'un fotograma Súper 8

Es pot apreciar que tot i que les pel·lícules Súper 8 es consideren d'una amplada de 8 mil·límetres, l'àrea de projecció real es bastant inferior. Així doncs, aplicant les Equacions 2 i 3 a partir de les alçades del fotograma, del sensor de la càmera i de la distància aproximada de 6 centímetres a la que es troba el fotograma, es pot trobar que la distància focal requerida és superior a 300 mm. El problema aquí és que els objectius amb una distància focal tan gran són enormes i s'utilitzen típicament per capturar imatges a grans distàncies com és el cas de fotografiar partits de futbol. A més, aquest càlcul no té en compta la distància mínima d'enfoc, que es defineix com la distància mínima entre el sensor de la càmera i l'objecte a fotografiar, de manera que si ens acostem més a l'objecte aquest es desenfoca. El problema aquí és que com més gran és la focal més gran és la distància mínima d'enfoc, és a dir, més lluny ens hem de situar per a poder enfocar, i en el cas dels objectius de 300 mm de focal la distància d'enfoc es situa a metres. Aquest càlcul doncs és una justificació a la problemàtica de no poder fer que el fotograma ocupi el total de la foto que es comentava a l'inici d'aquest apartat.

Normalment quan s'intenta fotografiar un objecte molt petit s'opta per la macrofotografia. La macrofotografia consisteix en poder capturar l'objecte desitjat a una distància molt curta obtenint així una maximització de 1:1 o superior. La macrofotografia esta basada en lents que permeten disminuir al màxim la distància mínima d'enfoc permetent acostar la càmera inclús a mig centímetre del objecte. Si es calculés altre cop la focal necessària amb les Equacions 2 i 3 considerant una distància de 0,5 cm del fotograma el resultat seria que es requereix una focal de 30 mm. Els objectius amb una focal d'aquest tipus ja de per si tenen una distància mínima d'enfoc curta, i amb algun element macro com un anell inversor es podria assegurar que amb aquesta focal es podria enfocar a mig centímetre, i per tant es maximitzaria el fotograma sense necessitat d'una focal molt gran. Però com ja s'ha

comentat, al tenir la limitació dels sis centímetres de distància no es pot efectuar una macrofotografia òptima. El que si se'n pot extreure de tot això és que d'una banda com més focal tingui l'objectiu més maximització del fotograma s'obté però a canvi es requereix d'algun element basat en macrofotografia que permeti reduir la distància d'enfoc del objectiu.

En base a això s'ha optat per l'objectiu Canon EF 75-300mm de la Figura 12, que té la particularitat que la seva distància focal és alta i variable, el que fa ideal per poder ajustar-lo fins a trobar un valor de focal que permeti la màxima ampliació i al mateix temps un bon enfoc. Cal tenir en compte que aquest objectiu té una distància mínima d'enfoc d'1,5 metres, de manera que tal com s'ha comentat requereix d'elements que disminueixin la seva mínima distància d'enfoc.



Figura 12. Objectiu Canon EF 75-300 mm

El primer dels elements que s'ha utilitzat acompanyant l'objectiu són tres tubs d'extensió de 12, 20 i 36mm respectivament representats a la Figura 13. Aquests tubs van situats entre el cos de la càmera i l'objectiu, augmentant així la distància entre el centre òptic de la lent i el sensor de la càmera, reduint així la distància mínima d'enfoc i permetent acostar-se més al fotograma, amb la conseqüent maximització de la imatge. L'avantatge d'aquests tubs és que internament estan buits, és a dir, no tenen cap lent i per tant deixen passar directament la llum. L'únic inconvenient que presenten és que disminueixen molt la llum que arriba al sensor de la càmera.



Figura 13. Tubs extensió de 12, 20 i 36 mm

Adicionalment als tubs d'extensió s'ha afegit una lent d'aproximació de 10 diòptries tal com es pot veure a la Figura 14, i va roscada directament sobre l'objectiu. Aquesta lent té un

funcionament semblant a una lupa, ja que genera una imatge virtual amplificada respecte a l'original. La idea és que igual que els tubs d'extensió, redueix la distància mínima d'enfoc amplificant al mateix temps la imatge. Tant la reducció de la distància mínima d'enfoc com el radi d'amplificació augmenten com més gran és el valor de les diòptries de la lent. En aquest cas s'ha optat per una lent de 10 diòptries ja que és el valor màxim que té una lent individual. A diferència dels tubs d'extensió, aquestes lent no disminueix la llum que arriba al sensor però si se'n posen varies seguides pot arribar a crear aberracions a la llum.



Figura 14. Lent d'aproximació de 10 diòptries

Pel que fa a la configuració del objectiu aquest s'ha deixat a 75 mm de focal, ja que al intentar posar focals més grans la distància mínima d'enfoc és massa gran com per compensar-ho únicament amb els tubs d'extensió i la lent d'aproximació d'escrits. A més, afegir més elements per reduir la distància d'enfoc podria provocar aberracions a la imatge, especialment si es volgués afegir més diòptries amb lents d'aproximació, ja que l'objectiu al variar la focal provoca una separació molt gran entre les lents d'aproximació i la lent de la càmera, generant aberracions als contorns de la imatge.

Com es pot apreciar, la focal que s'ha utilitzat no és tan gran considerant que per càlculs es requeria una focal de més de 300 mm per a què el fotograma ocupi el total de la foto, però aquí cal tenir en compte la maximització extra que aporten les 10 diòptries, a més del fet que com ja s'ha comentat es faria impossible enfocar a focals més grans a una distància de 6 centímetres. Tot i així, sota aquestes condicions s'ha aconseguit que el fotograma tingui una resolució aproximada de 1500x1100 píxels, que tot i que no arriba la resolució de 1080p(1920x1080), està per sobre de la resolució de 720p que ja es considera HD. De totes maneres, cal considerar que la qualitat ve limitada per l'estat de conservació dels fotogrames de la pel·lícula Súper 8, així que encara que es pugui dir que té una resolució de HD en nombre de píxels, a nivell visual sempre dependrà de la pel·lícula original en última instància.

Cal comentar que sota aquestes condicions una altra opció seria utilitzar un objectiu macro de focal gran i fixa, com podria ser de 100mm. Aquest objectius tenen l'avantatge de tenir una focal gran i una distància mínima d'enfoc curta per defecte, el que permeten prendre fotografies macro sense estar molt a prop. A canvi solen costar mínim cinc vegades més que l'objectiu que s'ha utilitzat. De fet, l'objectiu utilitzat en principi està pensat per utilitzar-se com a zoom d'alta capacitat, però s'ha aprofitat la seva gran focal per convertir-lo en un objectiu macro.

2.4.2. Configuració dels paràmetres de la càmera

Respecte a la configuració de la càmera s'ha optat per disparar-la en mode manual ja que ens permet un ajust més fi en funció de la llum que arriba a la càmera. Els tres paràmetres de configuració més importants a l'hora de disparar en mode manual són l'obertura del diafragma, la velocitat de l'obturador i la sensibilitat ISO.

L'obertura del diafragma és un paràmetre que determina la quantitat de llum que arriba al sensor. El seu valor es sol indicar amb un format com podria ser F1,4 o F32, essent F1,4 una obertura molt gran i F32 una obertura petita. És el paràmetre més important dels que s'ha configurat perquè a més de regular la quantitat de llum té un impacte molt important a l'enfoc, que com ja s'ha comentat és un dels problemes més grans degut a que ha calgut utilitzar un objectiu de focal molt gran que solen ser per disparar a molta distància. El fet és que com més tancat està el diafragma menys llum arriba però més gran és la zona que està enfocada. Aquí és quan entra en joc el que es coneix com a profunditat de camp, que tal com representa la Figura 15 indica l'amplada de la zona on la càmera serà capaç de capturar la imatge de forma nítida.



Figura 15. Representació de la profunditat de camp

Com ja s'ha comentat tots els objectius tenen una distància mínima d'enfoc però en general són capaços d'enfocar a l'infinit, és a dir, més enllà de la distància mínima d'enfoc teòricament poden captar-ho tot. El problema és que quan s'utilitzen elements de macrofotografia com és el cas dels tubs d'extensió i de les lents d'aproximació descrites,

l'amplada a la que enfoca la càmera, que teòricament va des de la mínima distància d'enfoc fins a l'infinit, s'escurça considerablement. Això es pot compensar tancant més el diafragma, augmentant així la profunditat de camp i assegurant així que el fotograma que es captura quedi situat dins la zona de captura nítida de la càmera. En base això, i gràcies a que s'ha optat per una font que aporta una gran quantitat de llum, es podria fixar un valor de diafragma molt alt, assegurant així una nitidesa perfecta del fotograma, tot i així en aquest cas s'ha fixat un valor de F16 degut a que un dels vidres que envolten el fotograma a l'àrea de projecció té una ratllada i experimentalment s'ha pogut comprovar que per valors per sobre de F16 la càmera enfoca aquesta ratllada.

El segon paràmetre que s'ha regulat és la velocitat de l'obturador. Aquest paràmetre bàsicament indica el temps que tarda la càmera a tancar el diafragma al disparar. La idea és que com més tarda a tancar el diafragma més llum entra, però pot provocar que si l'objecte a captar s'està movent el sensor capti aquest moviment i es vegi borrós. En aquest cas com que el fotograma està quiet en el moment de disparar la càmera només s'ha hagut de tenir en compte el temps fixat per programa entre dispar i dispar, de manera que s'ha fixat una velocitat d'obturador inferior a aquest temps per assegurar que no es perdi cap fotograma i al mateix temps s'ha ajustat tenint en compte la quantitat de llum que arriba. Com que entre fotograma i fotograma passen aproximadament uns 700 ms s'ha fixat un valor per defecte de 1/30 que equival a 33 ms.

L'últim paràmetre que s'ha configurat és la sensibilitat ISO. Aquest igual que els dos anteriors també afecta la quantitat de llum que arriba al sensor de la càmera en funció del seu valor, però de la mateixa manera que l'obertura del diafragma afecta a la profunditat de camp i la velocitat del obturador afecta al temps de dispar, la sensibilitat ISO afecta a un tercer factor que és la sensibilitat a la llum. La sensibilitat ISO pot prendre valors entre 100 i 6400, de manera que com més gran és el valor més sensible a la llum és el sensor i per tant més llum capta. Normalment la sensibilitat ISO s'utilitza per compensar la falta o l'excés de llum quan es requereixen uns valors de diafragma i velocitat d'obturació concrets, però també cal saber quan es trien valors de ISO grans per obtenir més lluminositat pot aparèixer soroll digital, que es tracta de punts que embruten la imatge. En aquest cas, es té prou llum com per poder mantenir la ISO a un valor molt baix com 160, de manera que es minimitza el soroll digital.

A mode de resum es pot veure la Figura 16 que representa els efectes de cadascun dels paràmetres configurats.

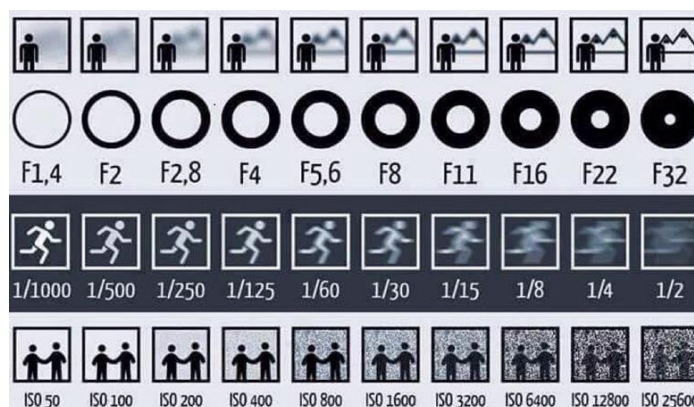


Figura 16. Efecte del diafragma, obturador i ISO

Cal comentar que excepte el valor d'obertura de diafragma, el valor dels altres dos paràmetres són modificables per assegurar una exposició correcta del fotograma, ja que cal considerar que no totes les pel·lícules es van revelar igual, de manera que algunes per defecte són més fosques i cal reajustar aquets paràmetres.

S'ha comentat que s'utilitza el programa propi de les càmeres Canon que és el Canon EOS Utility. En el apartat de sincronisme es comenta que et permet indicar la carpeta on es guardaran les fotos però addicionalment també permet ajustar el valor d'aquests tres paràmetres des de l'ordinador sense tocar la càmera, a més del fet que té un visualitzador que et permet veure el mateix que capta la càmera en funció dels valors dels paràmetres ajustats. Això és molt pràctic ja que permet assegurar que el fotograma es veu correctament sense el risc de moure la càmera. La interfície gràfica i les funcions que permet el programa Canon EOS Utility es poden veure a la Figura 17:

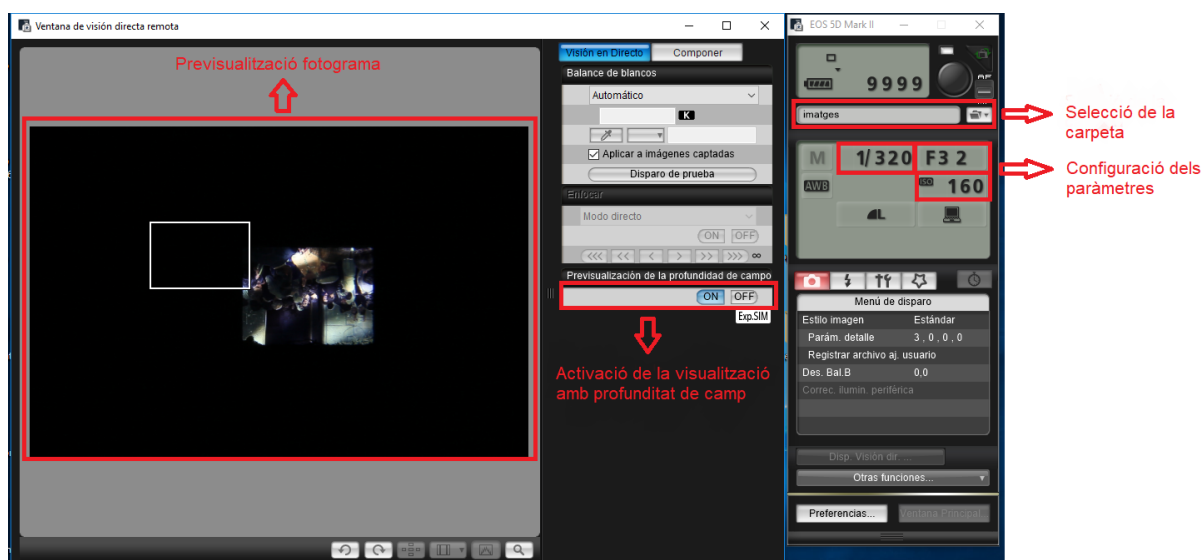


Figura 17. Interfície Canon EOS Utility

A l'hora d'utilitzar el Canon EOS Utility cal tenir dos coses en compte, la primera és que si es manté oberta la finestra per visualitzar el fotograma la càmera no es pot disparar. La segona és que per visualitzar correctament el fotograma cal posar a ON la visualització de la profunditat de camp, ja que per molt que es fixi una posició de diafragma molt tancada aquesta posició només l'executa al fer la foto, mentre que per defecte té el diafragma el màxim d'obert i no es pot apreciar com quedarà la foto realment. Activant aquesta opció el que fem es tancar el diafragma a la posició configurada poden visualitzar com quedarà la foto un cop es dispari.

2.5. Interfície d'usuari

Com ja s'ha comentat s'ha incorporat una petita interfície gràfica composta principalment per una pantalla LCD de 16x2 dígit i un botó, que permetrà obtenir informació durant el procés de captura. Les connexions amb el microcontrolador són les que es poden veure a la Figura 18:

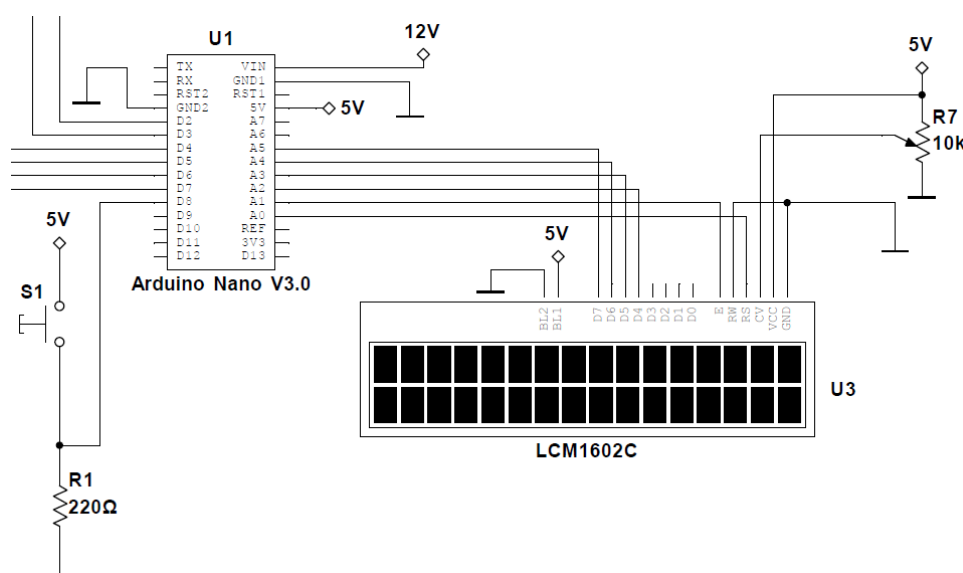


Figura 18. Connexió Botó i LCD

El botó s'ha connectat a 5 V i amb una resistència de pull up al corresponent pin del microcontrolador. Pel que fa a la LCD, que s'ha triat el model LCM 1602C, es comunica amb el microcontrolador a través dels quatre pins de dades D7, D6, D5 i D4, mentre que els altres quatre pins de dades restants no s'han connectat ja que no són necessaris per la comunicació. Pel que fa al pin enable que permet habilitar la pantalla, i el pin RS que permet definir el registre de la memòria de la LCD sobre el que es treballarà, es connecten a un dels pins del microcontrolador per poder-ne controlar el seu valor, mentre que el RW que serveix

per indicar si s'escriu o es llegeix es connecta a massa ja que només volem escriure. Finalment s'ha afegit un potenciòmetre de 10k Ω connectat en el pin CV del display que és l'encarregat de controlar el contrast de la pantalla. Respecte a l'alimentació de la LCD, aquesta requereix 5 V en dos pins, un és el Vdd que és l'alimentació pròpia per el funcionament de la LCD, i el pin BL2 que és la il·luminació de la pantalla.

Una de les funcions que permet aquesta interfície es mostrar una estimació del temps que tardarà el procés de captura en funció dels metres de pel·lícula, mentre que amb el botó l'usuari pot canviar els metres de la pel·lícula que vol capturar en intervals de 15 metres. Aquesta acció només es permet quan amb el selector es té seleccionat el mode OFF i la visualització queda representada a la Figura 19:

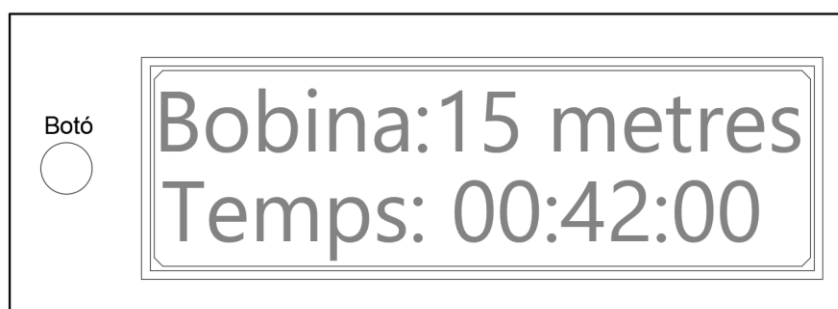


Figura 19. Introducció dels metres i estimació de temps

El fet que només es permeti canviar els metres en intervals de 15 metres és degut a que antigament les gravacions que es feien en format Súper 8 tenien un format estàndard que permetia gravar un nombre de fotogrames fixa. De fet aquest format solia de ser de cintes de 15 metres, que considerant les mides dels fotogrames això equival a uns 3750 fotogrames, o bé múltiples d'aquest fins a 60 metres. El problema és que aquest tipus de gravació era irreversible i no especialment barata, de manera que un cop haves començat a gravar ja no podies eliminar la gravació ni sobreescriure, per tant la gent aprofitava cada carret de gravació fins a esgotar-lo. Així doncs, lo més normal és que cada pel·lícula tinguí tots el fotogrames revelats i per tant sigui de la mida màxima de la cinta. Per això es deixa fixar 15, 30, 45 o 60 metres i per això també el temps que s'estima s'ha aproximat considerant que la pel·lícula conté tots els fotogrames equivalents al màxim de metres i tenint en compta la velocitat del motor, de manera que el temps està sobreestimat. Cal tenir en compta que aquesta estimació de temps és únicament del procés de captura, ja que el temps que tarda el programa de python a editar les imatges depèn en gran mesura del ordinador des del qual s'executa.

Un cop s'inicia el procés de captura automàtica tal com mostra la Figura 20, es van actualitzant en temps real el nombre de fotogrames capturats així com el temps estimat va disminuint a mode de cronòmetre.

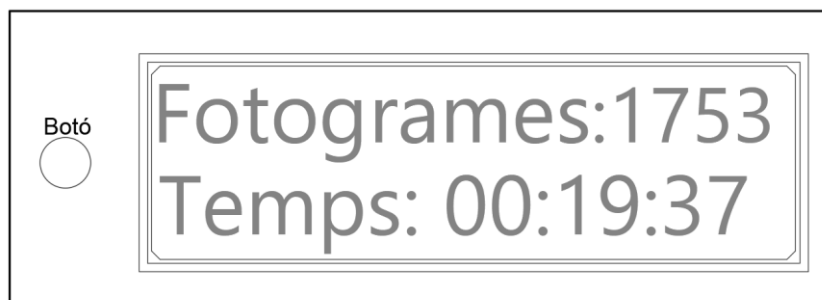


Figura 20. Visualització del fotogrames actuals i temps restant de captura

Finalment, quan es rep la corresponent senyal del programa de python indicant que s'ha acabat la pel·lícula, la pantalla mostra el que representa la Figura 21 on s'indiquen el nombre de fotogrames totals capturats, el que permetrà comprovar si hi ha aquest mateix nombre de fotos a la carpeta triada per així determinar que no hi ha hagut cap problema, i d'altre banda indica que el procés d'edició de les imatges a través de python està en marxa. En aquest punt ja es podria desconnectar el projector de l'ordinador, ja que a partir d'aquí ja no hi ha comunicació entre el programa d'arduino i el de python i només queda esperar a que s'hagin acabat d'editar totes les imatges.

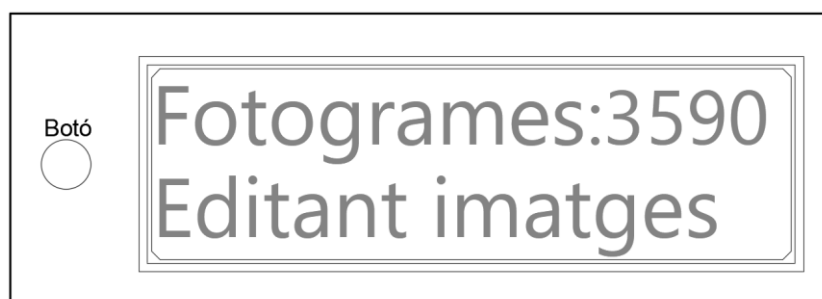


Figura 21. Visualització del fotogrames totals i indicació de captura acabada

3. PROGRAMACIÓ AMB ARDUINO

En aquest capítol es descriurà el funcionament del programa d'arduino, el qual és l'encarregat de gestionar principalment el sincronisme a través del control del motor pas a pas i de la lectura de les diferents senyals exteriors que rep tan del selector com del sensor inductiu.

Gràcies a que es requereix una comunicació via USB amb un ordinador per poder-se comunicar amb el programa de python, es podran carregar modificacions del programa per aquesta mateixa via en cas que sigui necessari.

3.1. Taules de variables

Per a poder entendre la programació que s'ha dut a terme, en aquest apartat es descriuran les variables d'entrada i sortida a través de la Taula 3 i la Taula 4, així com les variables internes que s'han utilitzat amb una descripció de la seva funció dins la Taula 5.

Nom de l'entrada	Pin arduino	Descripció
Rebobinar	D4	S'activa amb el selector en mode OFF o en mode rebobinar
Endavant	D5	S'activa amb el selector en mode OFF o en mode endavant
Capturar	D6	S'activa amb el selector en mode Captura
Inductiu	D7	S'activa en rebre la senyal del sensor inductiu
Boto	D8	S'activa al pulsar el botó que canvia per pantalla els metres de la pel·lícula

Taula 3. Variables d'entrades del programa arduino

Nom de la sortida	Pin arduino	Descripció
DirPin	D2	Especifica el sentit de gir del motor
StepPin	D3	Dóna l'ordre d'executar un pas del motor
Camera	D12	Activa el optoacobrador per disparar la càmera
RS	D14	Envia la senyal de selecció de registre a la LCD
EN	D15	Envia la senyal d'habilitació a la LCD
Dades4	D16	Envia la dada de comunicació pel canal 4 de la LCD
Dades5	D17	Envia la dada de comunicació pel canal 5 de la LCD
Dades6	D18	Envia la dada de comunicació pel canal 6 de la LCD
Dades7	D19	Envia la dada de comunicació pel canal 7 de la LCD

Taula 4. Variables de sortides del programa arduino

De la Taula 4 que representa les sortides, cal comentar que les últimes sis sortides que són les que s'utilitzen per comunicar-se amb la LCD físicament estan connectades a pins analògics de l'arduino, però a la taula s'han especificat com a pins digitals ja que l'arduino permet configurar els pins analògics que no s'utilitzen com a digitals, de manera que el pin A0 que seria el primer pin analògic passa a ser el D14, ja que l'últim pin digital és el D13, i així successivament fins el pin D19.

Nom de la variable	Descripció
Hores	Enregistra les hores que es mostren a la LCD
Minuts	Enregistra els minuts que es mostren a la LCD
Segons	Enregistra els minuts que es mostren a la LCD
Metres	Enregistra els metres de pel·lícula triats per pantalla
MemBoto	Variable per assegurar que s'ha deixat de polsar el botó que canvia els metres
Comptador	Comptador per enregistrar les vegades que el sensor inductiu detecta i deixa de detectar
Detecció	Variable per guardar el moment que el sensor inductiu detecta
Fotogrames	Enregistra el nombre de fotogrames capturats
ModeEndavant	Variable per indicar que s'està passant la pel·lícula sense capturar-la
ModeRebobinar	Variable per indicar que s'està rebobinant la pel·lícula
ModeCaptura	Variable per indicar que s'està capturant la pel·lícula
Python	Variable que enregistra el que es llegeix per port sèrie provinent del programa de python

Taula 5. Variables internes del programa arduino

3.2. Estratègia de control

El programa es basa en detectar quina posició del selector s'ha triat i actuar en conseqüència, ja sigui modificant la velocitat i el sentit de gir del motor com canviant el que es mostra per la pantalla de la LCD.

Si el selector es troba en la posició d'endavant o de rebobinar el programa simplement executa passos contínuament en el sentit corresponent fins que es trïi una posició del selector diferent. Aquest dos modes de funcionament són manuals i ajuden principalment a passar en primera instància la pel·lícula fins que aquesta s'hagi enganxat al rebobinador, i permet rebobinar la pel·lícula un cop s'ha acabat de capturar. Tant en el mode endavant com en el mode rebobinar s'activa una memòria per saber que s'ha passat per aquests estats.

La Figura 22 mostra el diagrama de flux simplificat que segueix el programa:

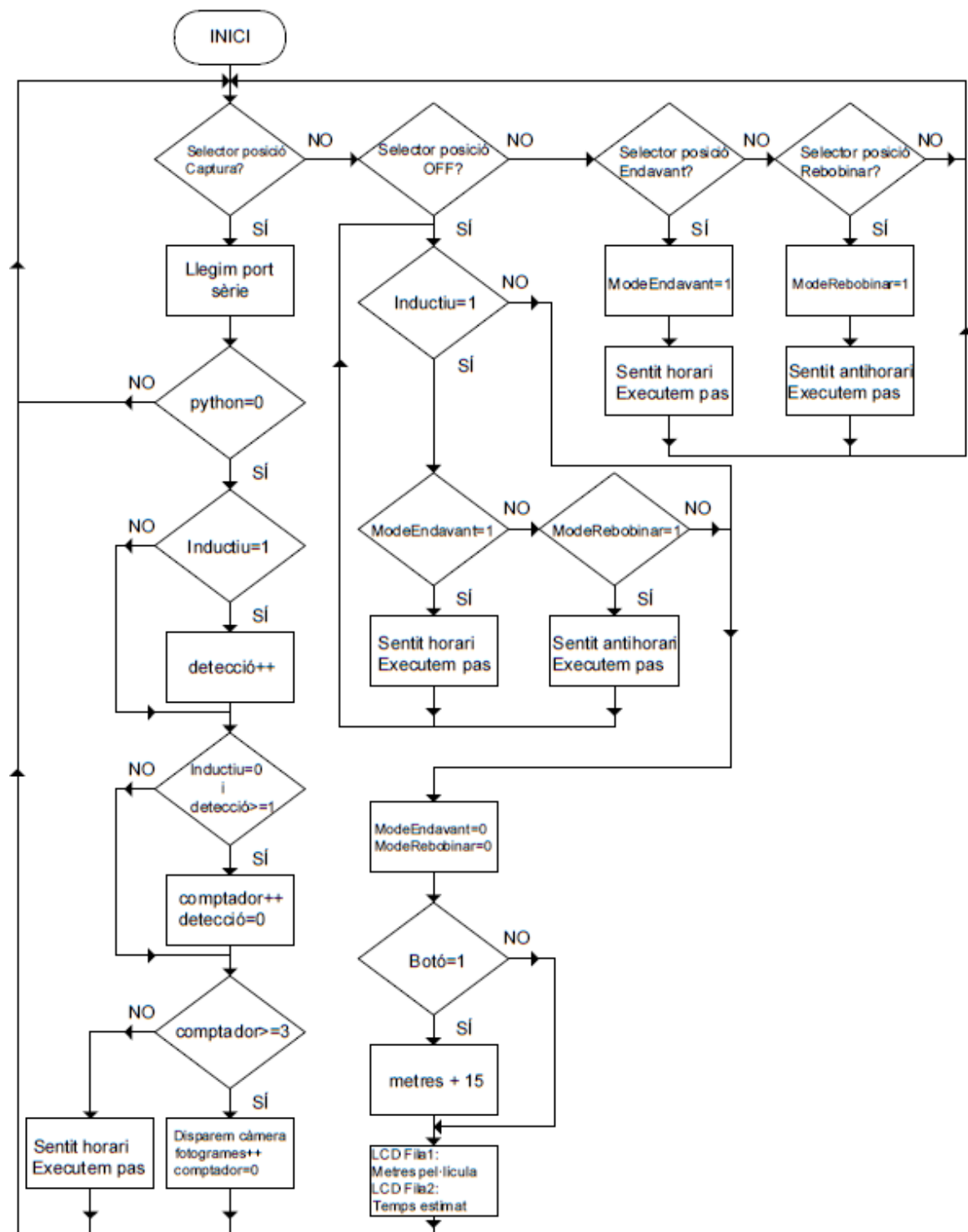


Figura 22. Diagrama de flux del programa d'arduino

Si el selector està a la posició OFF es comprova si el sensor inductiu està detectant. Si està detectant vol dir que s'està tapant la llum i no es pot visualitzar el fotograma. En aquest cas, en funció de si veníem del estat endavant o del estat rebobinar que es sap gràcies a la

corresponent memòria, es gira en el mateix sentit del estat corresponent fins que el sensor inductiu deixa de detectar. Això es fa per assegurar que cada vegada que estem en el mode OFF es pugui visualitzar el fotograma i és convenient per exemple la primera vegada de passar una pel·lícula, per poder apreciar el moment en que comencen els fotogrames i assegurar que òpticament es visualitza correctament. Si en el mode OFF no es detecta el sensor inductiu, aquest és simplement un estat d'espera on es pot introduir els metres de la pel·lícula que es capturarà a través del botó i al mateix temps et mostra per pantalla el temps aproximat que durarà la captura.

En el mode de captura automàtica el primer que es fa és llegir pel port sèrie el que es rep per part del programa de python. La idea és que si no es detecta un zero que és el valor predeterminat de la variable vol dir que el programa de python ens ha enviat un 1 i per tant s'entra en un estat indeterminat evitant el gir del motor i el dispar de la càmera. En cas que no es rebi res, tal com s'havia comentat en el apartat de sincronisme es comença el procés de detectar tres vegades el sensor inductiu. Per fer aquest procés a nivell de codi s'ha hagut d'utilitzar una variable addicional per enregistrar la detecció del sensor. La idea és que aquesta s'activa quan el sensor inductiu detecta i es desactiva quan el sensor deixa de detectar. A més, el comptador que s'utilitza per saber quan s'ha de disparar la càmera només s'incrementa quan el sensor inductiu no detecta i aquesta variable està activa, d'aquesta manera s'assegura que el que s'està comptant és les vegades que el sensor inductiu detecta i deixa de detectar i per tant que en el moment que es dispara la càmera no hi ha cap objecte interposant-se amb la llum.

Adicionalment al flux principal del programa, aquest també compte amb una interrupció que s'executa cada segon. El seu funcionament principal és executar el compta enrere del temps estipulat per a què acabi la captura de la pel·lícula. Precisament s'ha fixat aquest temps d'interrupció per poder anar restant segon a segon. A més de restar la variable que enregistra els segons, executa la lògica per anar visualitzant el temps correctament per pantalla, així com per actualitzar el nombre de fotogrames capturats en cada moment.

El compta enrere només l'efectua si estem en el mode captura i sempre i quan no es rebi cap senyal del programa python. Si és rep un 1 del programa python, que és l'equivalent a dir que es pari la captura, la interrupció canvia el que es mostra per pantalla i indica el nombre final de fotogrames capturats així com un missatge indicant que la captura ha finalitzat i que s'està executant el procés d'edició d'imatges.

El diagrama de flux corresponent de la interrupció es pot veure representat a la Figura 23:

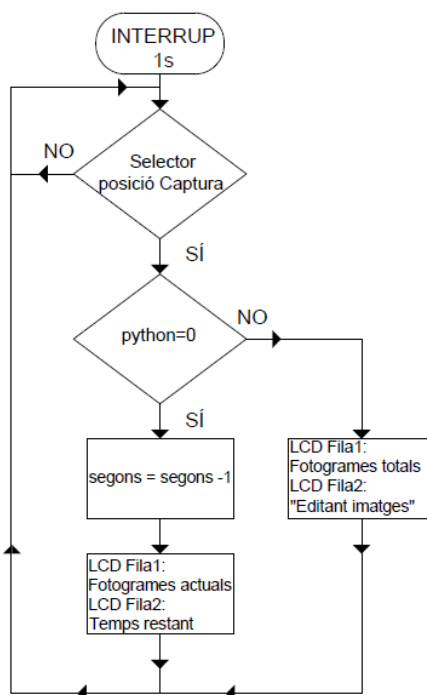


Figura 23. Diagrama de flux de la interrupció

3.3. Llibreries

En aquest programa bàsicament s'utilitzen dos llibreries. La primera d'elles és la LiquidCrystal, la qual és interna del entorn arduino i s'encarrega de gestionar la comunicació i controlar mòduls LCD genèrics. La llibreria permet fer la comunicació amb 4 bits tal com es vol i permet connectar els pins de la LCD als pins del microcontrolador que vulguem.

En general, per poder realitzar qualsevol acció com escriure sobre la pantalla s'ha d'establir un valor binari a cadascun dels bits: Data4, Data5, Data6, Data7, enable i RS. Fent això, el que s'aconsegueix és definir una posició de la memòria interna de la LCD on es troba un dels possibles caràcters que es pot visualitzar. Així doncs, un dels avantatges d'utilitzar aquesta llibreria és que no ens cal saber de memòria els bits per a cada caràcter que vulguem representar, sinó que simplement utilitzant la funció adient la llibreria ja s'encarrega de gestionar el valor intern dels bits. Entre les possibles funcions que ens permet la llibreria, s'han utilitzat les següents:

`Lcd.begin()`. Aquesta funció inicia la LCD i la prepara per a utilitzar-la. Addicionalment permet triar el nombre de files i columnes amb les que es treballarà.

`Lcd.clear()`. Neteja el que hi ha escrit a la pantalla per evitar que es sobreescriuin caràcters.

`Lcd.setCursor()`. Aquesta funció permet establir una adreça a través de dos variables de 8 bits cadascuna, de manera que es defineix la posició on s'escriurà el caràcter.

`Lcd.print()`. Mostra per pantalla el que es vulgui, ja siguin caràcters de text o variables de programa.

La segona llibreria utilitzada és la llibreria `TimerOne`. Aquesta s'ha hagut d'afegir ja que no ve per defecte al entorn d'arduino i és la que ens permet generar la interrupció temporitzada cada segon.

L'atmega328 que és el microcontrolador en el que està basat arduino consta de dos timers interns que permeten generar interrupcions temporitzades en funció de la freqüència base del seu cristall. Per poder configurar el temps de les interrupcions cal programar els seus registres interns, el que pot ser bastant tediós, per això la llibreria `TimerOne` gestiona internament els registres i ens simplifica la generació de la interrupció ja que ens permet senzillament fixar el temps d'interrupció que es vol numèricament. Les dos funcions principals que s'utilitzen d'aquesta llibreria són:

`Timer1.initialize()`. Aquesta funció permet establir el temps en mil·lisegons amb el que s'executarà la interrupció temporitzada.

`Timer1.attachInterrupt()`. Lliga la interrupció temporitzada a una funció amb el nom que es vulgui. De manera que cada vegada que es compleixi el temps fixat executarà aquella funció.

4. PROCESSAMENT D'IMATGES AMB PYTHON

En aquest capítol es descriurà el funcionament del programa de python que permetrà d'una banda determinar el final de la pel·lícula, i editar cadascuna de les imatges per eliminar el contorn innecessari a fi de deixar una foto amb la mida del fotograma. D'aquesta manera es compensa el fet que òpticament no sigui possible que el fotograma ocupi el 100% de la mida de la foto.

S'ha optat per utilitzar codi python degut a la gran versatilitat d'aquest llenguatge, el que li permet executar les funcions bàsiques de qualsevol llenguatge d'alt nivell i al mateix temps disposa d'un gran nombre d'extensions que permeten orientar el codi de python cap a diferents estils funcionals. En aquest cas l'extensió principal utilitzada és la de OpenCV, que es tracta d'una llibreria lliure creada per Intel que permet accions relacionades amb la visió per computació. Aquesta extensió permet convertir al codi de python en un programa d'edició d'imatges. L'únic inconvenient de python és que es tracta d'un programa interpretat, el que fa que el seu temps d'execució no sigui molt ràpid i depengui en gran mesura de les capacitats del processador on s'executa.

El codi python al ser un programa interpretat requereix de tenir instal·lat el compilador corresponent per a poder ser executat a qualsevol ordinador. Tot i així, en aquest projecte s'inclou tan el codi com un executable creat a partir d'aquest que ja conté internament el compilador, el que permet executar-lo des de qualsevol ordinador sense haver de tenir instal·lat l'entorn de python.

4.1. Taula de variables

Per a poder entendre amb més detall la programació que s'ha dut a terme, en aquest apartat es descriuran a través de la Taula 6 les funcions de les diferents variables internes que s'han utilitzat.

Nom de la variable	Descripció
Ts	Guarda la data de creació de la foto anterior
Fts	Guarda la data de creació de la foto següent
Comptador	Enregistra el nombre de fotogrames correctes
Rang_negre_baix	Límit inferior del rang de valors per detectar negre
Rang_negre_alt	Límit superior del rang de valors per detectar negre
Captura	Indica si estem en el bucle per detectar fotogrames negres
X_tot	Valor acumulat de la coordenada x en el retall del contorn

Nom de la variable	Descripció
X_mitjana	Valor mitjà de la coordenada x en el retall del contorn
Y_tot	Valor acumulat de la coordenada y en el retall del contorn
Y_mitjana	Valor mitjà de la coordenada y en el retall del contorn
W_tot	Valor acumulat de l'amplada en el retall del contorn
W_mitjana	Valor mitjà de l'amplada en el retall del contorn
H_tot	Valor acumulat de l'amplada en el retall del contorn
H_mitjana	Valor mitjà de l'amplada en el retall del contorn
Port	Guarda el nom del port entrat per pantalla
Carpeta_origen	Guarda la ruta de la carpeta d'origen entrada per pantalla
Carpeta_nova	Guarda la ruta de la carpeta final entrada per pantalla
Arduino	Guarda les característiques del port sèrie utilitzat
List_of_files	Llista de les fotos de la carpeta d'origen
Latest_file	Foto de la carpeta d'origen amb data de creació més gran
Imatge	Imatge original guardada amb forma de matriu
Mask	Màscara creada a partir dels rangs mínims i màxim de negre
Moments	Moment creat a partir de la màscara
Area	Valor de l'àrea formada pels píxels negres
Num_rows	Número de columnes de la imatge
Num_cols	Número de files de la imatge
Rotation_matrix	Valor de la matriu per girar la imatge 180 graus
Imatge_girada	Imatge girada 180 graus expressada com a matriu
Fotograma	Còpia de la imatge original girada expressada com a matriu
Imgray	Imatge convertida en escala de grisos
Thresh	Valor del threshold per detectar els contorns no desitjats
Contours	Valor dels punts que determinen el contorn amb una àrea més gran
Cnt	Valor dels contorns expressat com a matriu
Crop	Imatge retallada a partir del valor dels contorns
Foto	Indica quants fotogrames editats es tenen
IMG	String per poder indicar el número de fotograma en el nom

Taula 6. Variables del programa python

4.2. Flux del programa

En aquest apartat es mostra la lògica que segueix el programa per assolir el funcionament desitjat. Es tracta d'un programa purament seqüencial que bàsicament es divideix en dos funcions principals: detectar fotogrames negres per determinar el final de la pel·lícula, i editar una per una les imatges capturades.

Inicialment el programa demana la introducció per pantalla de les dades necessàries per executar el programa. Aquestes són respectivament el port sèrie on està connectat l'USB de l'arduino, el qual s'ha d'introduir en un format com podria ser COM4 per exemple, la ruta de

la carpeta on s'ha triat que es guardin les fotos originals fetes per la càmera, i la ruta de la carpeta on es vol que es guardin les fotos editades. La idea de fer això és que al no ser dades fixes el programa es pot executar des de qualsevol ordinador. Un cop introduïdes les dades el programa mira si a la carpeta on es guarden les fotos fetes per la càmera està buida, i si és així és manté a l'espera. Això s'ha dissenyat així ja que el procediment es basa en que calen unes preparacions prèvies a la captura de la pel·lícula, com són comprovar la visibilitat del fotograma des de el Canon EOS Utility i triar les corresponents carpetes, ja sigui des de el programa de Canon com des de el de python. De manera que un cop tot està preparat i s'executa el programa de python, només cal posar el selector en el mode de captura automàtica.

Un cop fet això el programa de python comença a detectar imatges a la carpeta i va contínuament llegint l'última imatge de la carpeta. Cada vegada que llegeix una imatge analitza si es tracta d'un fotograma negre i si és així incrementa el valor d'un comptador. La idea és detectar tres fotogrames negres seguits per determinar que estem al final de la pel·lícula. El fet que es necessitin tres fotogrames negres seguits es per assegurar que no sigui que algun fotograma concret de la pel·lícula estigui malmès i es revelés malament, de manera que si detecta per exemple dos fotogrames negres però el tercer no ho és, reinicia el valor del comptador. Quan es produeix el cas que detecta tres fotogrames negres seguits, el programa de python envia un 1 a l'arduino via port sèrie per indicar-li que pari el motor i el dispar de la càmera. Al mateix temps reinicia el valor del comptador i varia el valor de la variable que indica si estem en el mode captura, permetent sortir del bucle que analitzava contínuament les imatges en busca de fotogrames negres.

A partir d'aquí ja només actua el programa de python, el que permet desconectar els dos cables USB tant de la càmera com de l'arduino. En aquest punt, el programa va passant per cadascuna de les imatges i va fent la corresponent edició de les mateixes. Per poder passar per cadascuna de les imatges el programa treballa dins una estructura de bucle que analitza totes les imatges que tenen una extensió JPG, i per poder editar les imatges en l'ordre correcta el programa va extraient la data de creació de cascuna de les imatges. Cada vegada que edita una imatge la guarda a la corresponent carpeta que s'ha triat a l'inici del programa, i un cop ha determinat que no hi ha més imatges per editar s'espera 5 segons i ell mateix es tanca.

A la Figura 24 és pot veure el diagrama de flux del programa que mostra de forma simplificada la lògica del seu funcionament.

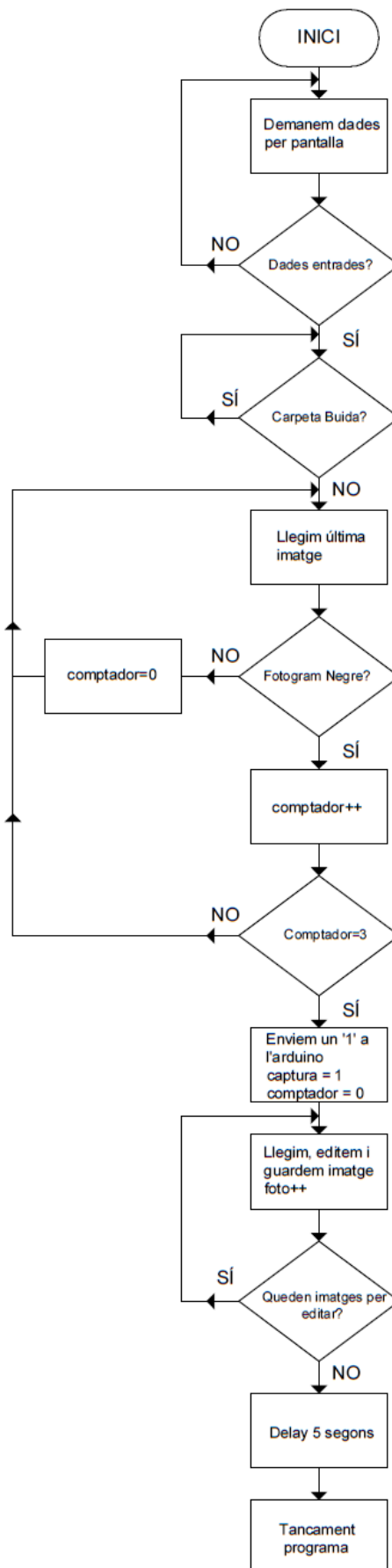


Figura 24. Diagrama de flux del programa de python

4.3. Detecció de fotograma negre

En el apartat anterior es mostrava el diagrama de flux simplificat del programa, però no s'entrava en detall respecte a com analitza les imatges. En aquest dos últims apartats s'especifica les dos funcions principals que són la detecció de fotograma negre i l'eliminació dels contorns.

En les dues funcions cal tenir present que en el món de la visió per computació les imatges es tracten com a matrius, on cada punt de la matriu representa un píxel i està especificat per una coordenada x i una coordenada y. Addicionalment cada punt de la matriu o píxel té una valor que determina el color d'aquell píxel. Com que en format digital el color es basa en el sistema RGB, cada píxel estarà conformat per tres valors d'entre 0 i 255. Tal com mostra la Figura 25, això és equivalent a dir que al tractar una imatge en color estarem treballant davant de tres matrius de dimensions equivalents però valors d'intensitats diferents.

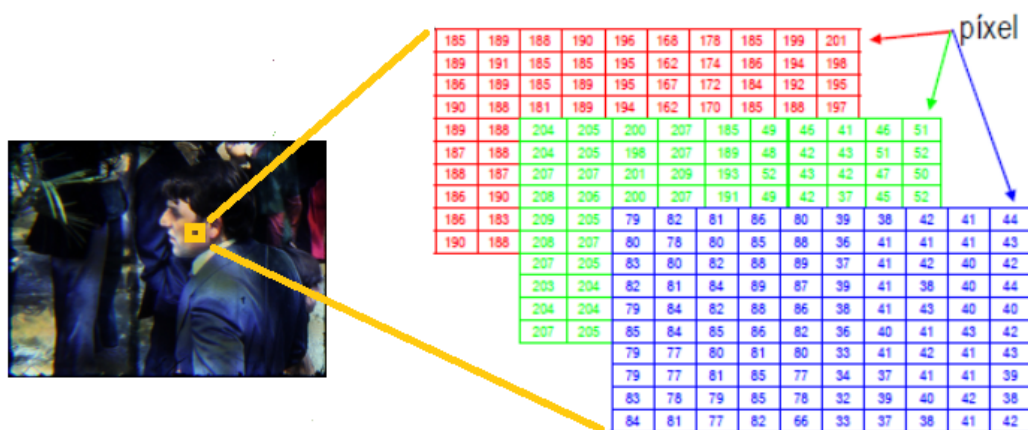


Figura 25. Representació d'imatges com a matrius

En base al concepte de la Figura 25, la funció de detectar un fotograma negre parteix d'estipular un rang de valors per els quals un píxel es considerarà negre. Com que en el sistema RGB el negre pur es situa en un valor de zero, s'ha estipulat un rang d'entre zero i deu en forma de matriu d'una fila i tres columnes de manera que aquest rang es podrà comparar amb els valors dels tres colors de cadascun dels píxels. En base a aquest rang es pot crear el que es coneix com una màscara. Una màscara no és res més que la representació d'una imatge que es mostra únicament en dos colors, blanc o negre, de manera que representa els píxels dintre del rang en blanc i els que estan fora amb negre. A mode d'exemple, a la Figura 26 es pot apreciar una imatge original i al seu costat l'equivalent aplicant la màscara amb el rang definit.



Figura 26. Imatge original i la seva màscara de negre

Cal tenir en compte que la màscara s'aplica a la imatge original captada per la càmera, és a dir, quan el fotograma ocupa una petita part del total de la foto mentre que tot el voltant és negre. De manera que el que realment es veuria a la imatge amb la màscara feta seria el mateix que es representa a la Figura 26 i al voltant tot un rectangle en blanc que representa el negre de la foto original.

Un cop detectats els píxels que estan en negre gràcies a la màscara es genera el moment dels punts que estan dins el rang. El moment és un concepte matemàtic que s'utilitza per trobar la forma que adopten un conjunt de punts. La idea és que l'extensió de OpenCV no només ens permet general el moment dels punts detectats, sinó que a partir de la forma generada per aquests ens permet calcular-ne la seva àrea. Així doncs, quan ja no queda cap fotograma i la càmera està detectant una foto completament negre ens donarà una àrea més gran que quan si hi ha un fotograma, ja que en aquest cas se li haurà de restar l'àrea pròpia del fotograma que no detecti com a negre. Experimentalment s'ha trobat un valor de 5300000000 com a òptim, de manera que quan detecta una àrea superior a aquesta es considera que és un fotograma negre.

4.4. Eliminació dels contorns

La segona de les funcions respecte a l'edició de les imatges consisteix en eliminar el contorn negre del voltant del fotograma. Aquesta funció es comença a executar un cop s'ha confirmat el final de la pel·lícula i la càmera ja no està disparant.

Previ al retall del contorn també es gira la imatge 180 graus, ja que cal tenir en compte que la pel·lícula passa invertida per la zona de projecció. Per generar el gir de la imatge el programa busca el punt mig d'aquesta obtenint el nombre de files i columnes i dividint aquest valors per la meitat. L'extensió de OpenCV ens permet calcular la corresponent matriu de rotació especificant el punt mig de la imatge i l'angle de gir desitjat. Tal com es

comentava a l'apartat anterior la imatge es representa pròpiament com una matriu on els punts on coincideixen files i columnes són els píxels, i al mateix temps l'origen de coordenades es situa a l'extrem superior esquerre de la imatge. Per tant, no és d'estranyar que es requereixi una matriu de transformació per moure les coordenades de cada píxel.

Per detectar els contorns de la imatge es realitza un procés semblant al de generar la màscara. També es crea un rang de valors límit amb el que es diferencia el negre de la resta de colors, però en aquest cas prèviament es converteix la imatge en escala de grisos. Això és degut a que en escala de grisos els colors només estan representats entre una escala de negre pur o blanc pur, de manera que no ens cal analitzar els tres colors de l'escala RGB. A més, al analitzar només blanc o negre permet ser més precís, ja que a diferència de quan es calculava l'àrea de la màscara aquí cal detectar a la perfecció el negre del contorn per a l'hora de retallar la imatge assegurar que no es retalla una part del fotograma o bé que es vegi una part molt amplia del contorn negre. En aquest cas el límit que s'ha establert per detectar negre també és d'entre 0 i 10. En base a aquest rang, el programa utilitza una funció de l'OpenCV que permet trobat les coordenades de tots els punts que delimiten i separen àrees amb colors dins el rang d'àrees amb colors fora del rang. Aquí però ha calgut afegir-li la condició que ens agafi únicament les coordenades dels punts de contorn que generen una àrea més gran, ja que sinó es creaven petites àrees de contorn degut a les imperfeccions del fotograma i no sempre donava les mateixes coordenades. Cal tenir en compte que el contorn negre que rodeja el fotograma no és un negre pur, a més del fet que tot i que el fotograma té una forma rectangular aquesta no es perfecte i té unes certes deformacions provocant que el programa pugui detectar minúscules àrees negres. En base a això, un exemple del contorn que trobaria el programa seria el que es representa a la Figura 27:



Figura 27. Contorns del fotograma

Com es pot apreciar el contorn no es perfecte ja que agafa alguna part del fotograma com a fora del contorn, és a dir, ho considera com a negre. Això es podria solucionar fent més exigent el límit de negre baixant-lo per sota de 10 per exemple, ara bé, això podria ocasionar que en algunes situacions retallés una petita part innecessària de contorn. Això sol passar quan en un dels extrems del fotograma hi ha algun color tirant a blanc, de manera que amb la llum es provoca que el negre del contorn del mateix costat arribi a considerar-lo com a part del fotograma. Per solucionar-ho s'ha optat per generar una funció del OpenCV que genera un rectangle delimitador en funció dels punts de contorns obtinguts. La idea és que aquest rectangle generat busca una coordenada x i y d'origen i genera una amplada i una alçada de manera que tots els punts del contorn estiguin inclosos dins el rectangle. Per a fer això, busca el punt de contorn amb una coordenada x més propera a l'origen de coordenades, de manera que aquesta serà la coordenada x d'origen del rectangle, i al mateix fa amb la coordenada y. Mentre que per trobar l'amplada i l'alçada fa el contrari i busca els punts de contorn amb una coordenada x i y lo més lluny possible de l'origen. La delimitació del rectangle es pot veure a la Figura 28:



Figura 28. Rectangle delimitador en funció dels contorns

A partir d'aquí l'únic que cal es dir-li que retalli les imatges a partir de les coordenades del rectangle. Ara bé, com ja s'ha comentat en funció dels colors del fotograma o bé perquè algun fotograma hagi quedat mal revelat, es pot donar la situació en que un contorn englobi part del voltant negre del fotograma. Per corregir això i fer que el retall sigui lo més constant possible, cada vegada que s'obté les coordenades del rectangle delimitador d'un dels fotogrames es guarden i es va fent una mitjana de manera que el retall s'acaba fent amb les coordenades mitjanes i no amb les individuals de cada imatge. A més, s'ha afegit una condició de manera que si d'algun fotograma s'obté unes coordenades molt diferenciades del valor de la mitjana, se li assigna a aquell fotograma concret el valor de la mitjana, evitant així retalls incorrectes degut a imperfeccions de fotogrames concrets.

5. RESUM DEL PRESSUPOST

El cost associat al disseny, a la creació del circuit imprès, al muntatge de la PCB, la instal·lació dels components de l'escàner, la programació, el mecanitzat i la posada en marxa és de mil nou-cents vuitanta-tres euros amb trenta-un cèntims, sense IVA.

6. CONCLUSIONS

Un cop realitzat el projecte es pot concloure que s'han complert amb les especificacions que s'havien definit. S'ha obtingut un escàner capaç de capturar automàticament els fotogrames continguts en una pel·lícula en format Súper 8, i aquests finalment es guarden en format digital a una carpeta de l'ordinador a l'espera del muntatge de la pel·lícula, tal com s'havia plantejat.

Aquest objectiu s'ha aconseguit partint d'un projector de Súper 8 tal com s'havia descrit a l'inici d'aquest projecte. Això ha comportat avantatges com és el fet que ha simplificat el muntatge mecànic, però també ha originat problemes pel que fa a l'òptica, que han acabat afectant a la qualitat final de les imatges i per tant de les pel·lícules. Tot i així, considerant que els fotogrames acaben tenint una resolució en nombre de píxels superior al mínim per considerar-se HD, i a la vista de les pel·lícules obtingudes a partir dels fotogrames capturats, es pot considerar que s'ha complert l'objectiu d'obtenir una qualitat superior a la que s'obtindria per altres mètodes com podria ser la gravació sobre la pròpia projecció de la pel·lícula a la paret, a més del fet que l'únic element que té un preu elevat de tot aquest sistema és la pròpia càmera.

A nivell crític, cal considerar l'imprevist de requerir un programa addicional per editar les imatges i compensar el fet que els fotogrames no ocupen el total de la imatge, el que ha comportat que l'escàner tingui una dependència d'un ordinador extern.

Tot i aquest fet, no deixa de ser un sistema de captura automàtica fotograma a fotograma que substitueix sistemes manuals i de menys qualitat, i que compleix el principal objectiu de digitalitzar pel·lícules que amb el seu format original s'acabarien perdent.

Gerard Cano Egea

Graduat en Enginyeria Electrònica Industrial i Automàtica

Girona, 3 de setembre de 2018

7. RELACIÓ DE DOCUMENTS

El present projecte es troba desenvolupat al llarg de cinc documents, aquests són la memòria, els plànols, el plec de condicions, l'estat d'amidaments i el pressupost.

8. BIBLIOGRAFIA

CANON INC. Manual software EOS Utility ver.2.9. Madrid. 2010.

ELECTRONICS KATRANGI TRADING. Proximity inductive 8BX. Full característiques sensor LJ8A3-2-Z/BX. China. Abril 2010.

FOTO24. Accesorios para Canon EOS 5D Mark II.(<https://www.foto24.com/accesorios-canon-5d-mark-ii>, 20 d'abril de 2018)

FULTON, W. Calculate distance or size of an object in a photo imatge.(<https://www.scantips.com/lights/subjectdistance.html>, 12 d'abril de 2018)

GUILLÉN, M. Diafragma, obturación e ISO.(<https://hellocreatividad.com/diafragma-obturacion-iso>, 15 d'abril de 2018)

LLAMAS, L. Motores paso a paso con arduino y driver A4988 o DRV8825.(<https://www.luisllamas.es/motores-paso-paso-arduino-driver-a4988-drv8825/>, 2 de març de 2018)

LLAMAS, L. Controlar arduino con python y la librería pyserial.(<https://www.luisllamas.es/controlar-arduino-con-python-y-la-libreria-pyserial/>, 16 d'abril de 2018)

MARCO, B. Lentes de aproximación o dióptricas.(<https://blasfotografia.com/blog/lentes-de-aproximacion-o-dioptricas>, 24 de març de 2018)

MARCO, B. Tubos y fuelles de extensión.(<https://blasfotografia.com/blog/tubos-y-fuelles-de-extension>, 1 d'abril de 2018)

OJEDA, J. Longitud focal, distancia de enfoque y profundidad de campo.(<http://fotografiaparaprinicipiantes.blogspot.com.es/2012/06/longitud-focal-distancia-de-enfoque-y.html>, 12 de març de 2018)

OPENCV. OpenCV-Python Tutorials.

(https://docs.opencv.org/3.4.1/d6/d00/tutorial_py_root.html, 4 d'abril de 2018)

RUBIO, M. Cómo hacer un disparador Casero para cámaras Canon.(

<https://www.fotografonocturno.com/como-hacer-un-disparador-casero-para-camaras-canon>,
2 de març de 2018)

SWEIGART, A. Automate the boring stuff with python: Practical programming for total beginners. Editorial No Starch Press. San Francisco. 2015.

TEXAS INSTRUMENTS. DRV8825 Stepper Motor Controller IC. Full característiques DRV8825. Texas. Juny 2014.

VISHAY SEMICONDUCTORS. Optocoupler, phototransistor output, with base connection. Full característiques 4N35. Gener 2010.

9. GLOSSARI

EF	Electronic Focus. Es diu dels objectius amb una muntura capaç d'acoblar-se a totes les càmeres EOS de Canon.
HD	High Definition. Es diu del sistema d'imatge amb una resolució de 1280x720 píxels o superior.
ISO	International Organization for Standardization. Organització creadora d'estàndards internacionals com és el cas de l'escala de sensibilitat fotogràfica.
JPG	Joint Photographic Experts Group. Estàndard de compressió i codificació d'arxius i imatges.
LCD	Liquid Cristal Display. Pantalla formada per monocromes situats davant una font de llum.
LED	Light Emiting Diode. Es diu de les fonts de llum constituïdes per un material semiconductor i dos terminals.
PCB	Printed Circuit Board. Es diu de les plaques electròniques de circuit imprès.
PWM	Pulse Width Modulation. Tècnica per controlar la quantitat d'energia que s'envia a una càrrega modificant el cicle de treball d'un senyal periòdic.
RGB	Red, Green and Blue. Model de representació de color en funció de la composició i intensitat dels colors primaris de la llum.
SD	Secure Digital. Dispositiu en format de targeta de memòria per a dispositius portàtils.
USB	Universal Serial Bus. Bus industrial que defineix el protocol per comunicar perifèrics i dispositius electrònics.

A. PROGRAMA ARDUINO

```
#include <LiquidCrystal.h> //Llibreria LCD
#include <TimerOne.h> //Llibreria per configurar timer

LiquidCrystal lcd(14, 15, 16, 17, 18, 19); // (RS,EN,d4,d5,d6,d7) Definim Pins LCD

//Definim els pins a utilitzar
const int dirPin = 2;
const int stepPin = 3;
const int camara = 12;
const int rebobinar = 4;
const int capturar = 6;
const int endavant = 5;
const int inductiu = 7;
const int boto = 8;
const int RS = 14;
const int EN = 15;
const int Dades4 = 16;
const int Dades5 = 17;
const int Dades6 = 18;
const int Dades7 = 19;

//Definició de variables
int hores = 0, minuts = 10, segons = 0, metres = 15, MemBoto = 0, comptador = 0;
int deteccio = 0;
int fotogrames = 0;
int ModeEndavant = 0, ModeRebobinar = 0, ModeCaptura = 0;
char python = '0';

void setup() {
  Serial.begin(9600); //Definim velocitat port sèrie

  //Definim pins com a entrades o sortides
  pinMode(dirPin, OUTPUT);
  pinMode(stepPin, OUTPUT);
  pinMode(camara, OUTPUT);
  pinMode(rebobinar, INPUT);
  pinMode(capturar, INPUT);
  pinMode(endavant, INPUT);
  pinMode(inductiu, INPUT);
  pinMode(RS, OUTPUT);
  pinMode(EN, OUTPUT);
  pinMode(Dades4, OUTPUT);
  pinMode(Dades5, OUTPUT);
  pinMode(Dades6, OUTPUT);
  pinMode(Dades7, OUTPUT);
  pinMode(boto, INPUT);

  lcd.begin(16, 2); // Fixem el número de caràcters i files
  Timer1.initialize(1000000); // Interrupció cada 1 segon
  Timer1.attachInterrupt(ISR_Blink); //Interrupció associada a ISR_Blink
}

void loop(){
  // Mode captura automàtica
  while((digitalRead(capturar) == HIGH) && (digitalRead(endavant) == HIGH) &&
    (digitalRead(rebobinar) == LOW) && (python == '0')){
    ModeCaptura = 1; //Memòria per saber que s'està en el mode captura
    if (Serial.available() > 0){ //Si el port sèrie està disponible
      python = Serial.read(); //Llegim el que hi ha en el port sèrie
    }
    if(digitalRead(inductiu) == HIGH){ //Si l'inductiu detecta
      deteccio = deteccio + 1;
    }
    if((digitalRead(inductiu) == LOW) && deteccio >= 1){ //Inductiu deixa de detectar
      comptador = comptador + 1;
    }
  }
}
```

```

    deteccio=0;
}
if(comptador<3){ //Mentre l'inductiu no detecta tres vegades seguim girant
    digitalWrite(dirPin, HIGH); //Establím direcció endavant
    digitalWrite(stepPin, HIGH); // Executem un pas
    delayMicroseconds(750); //Definim temps entre pas i pas
    digitalWrite(stepPin, LOW);
    delayMicroseconds(700);
}else if(comptador>=3){ //Si es detecta per tercera vegada
    digitalWrite(l2,HIGH); //Disparem la càmera
    fotogrames=fotogrames +1; //Incrementem el nombre de fotogrames
    delay(200); //Donem temps a la càmera per fer la foto
    digitalWrite(l2,LOW);
    comptador=0; //Reiniciem el comptador
}
}
//Mode endavant. Passa la pel·lícula fins que es pari manualment
while((digitalRead(rebobinar)==LOW) && (digitalRead(endavant)==HIGH) &&
(digitalRead(capturar)==LOW)){
    ModeEndavant=1; //Activem la memòria de mode endavant
    ModeRebobinar=0; //Reiniciem la memòria del mode rebobinador
    digitalWrite(dirPin, HIGH); //Executem passos endavant
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(550);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(350);
}
//Mode rebobinar
while((digitalRead(rebobinar)==HIGH) && (digitalRead(endavant)==LOW) &&
(digitalRead(capturar)==LOW) ){
    ModeRebobinar=1; //Activem la memòria de rebobinar
    ModeEndavant=0; //Reiniciem la memòria del mode endavant
    digitalWrite(dirPin, LOW); //Executem passos endarrere
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(550);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(350);
}
//Mode OFF
while((digitalRead(rebobinar)==HIGH) && (digitalRead(endavant)==HIGH)){
    delay(100); //Petit delay per el canvi de velocitat
    python='0'; //Reiniciem la variable rebuda pel codi python
    fotogrames=0; //Reiniciem el nombre de fotogrames
    deteccio=0; //Reiniciem la variable de deteccio
    comptador=0; //Reinicime el comptador de detecció
    while(ModeEndavant==1){ //Si veniem del mode endavant
        digitalWrite(dirPin, HIGH); //Seguim girant en el mateix sentit
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(550);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(350);
        if(digitalRead(inductiu)==HIGH){ //Procés de detecció de l'inductiu
            deteccio=deteccio+1;
        }
        if((digitalRead(inductiu)==LOW) && deteccio>=1){
            comptador=comptador+1;
            deteccio=0;
        }
        if(comptador==2){ //Quan l'inductiu deixa de detectar 2 cops
            comptador=0; //Parem motor i reiniciem variables
            ModeEndavant=0;
        }
    }
}
while(ModeRebobinar==1){ //Si veniem del mode rebobinar
    digitalWrite(dirPin, LOW); //Seguim girant en el mateix sentit
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(550);
    digitalWrite(stepPin, LOW);
}

```



```

    delayMicroseconds(350);
    if(digitalRead(inductiu)==HIGH){//Procés de detecció de l'inductiu
        deteccio=deteccio+1;
    }
    if((digitalRead(inductiu)==LOW) && deteccio>=1){
        comptador=comptador+1;
        deteccio=0;
    }
    if(comptador==2){//Quan l'inductiu deixa de detectar 2 cops
        comptador=0;//Parem motor i reiniciem variables
        ModeRebobinar=0;
    }
}
//Activació del botó per canviar els metres de bobina
if(digitalRead(boto)== HIGH && MemBoto==0){
    metres=metres + 15;
    MemBoto=1;//Activem memòria botó
}else if(digitalRead(boto)== LOW){//Si es deixa de polsar
    MemBoto=0;//Reiniciem memòria botó
}
//Temps en funció dels metres de bobina
if(metres==15){
    hores=0;
    minuts=42;
    segons=0;
}
if(metres==30){
    hores=1;
    minuts=24;
    segons=0;
}
if(metres==45){
    hores=2;
    minuts=6;
    segons=0;
}
if(metres==60){
    hores=2;
    minuts=48;
    segons=0;
}
if(metres==75){//Si metres són 75 tornem a 15
    metres=15;
}
if(ModeCaptura==1){//Si veniem del mode captura
    lcd.clear();//Netegem pantalla
    ModeCaptura=0;//Reiniciem memòria
}
lcd.setCursor(0,0); //Escrivim a la primera fila els metres de bobina
lcd.print("Bobina:");
lcd.print(metres);
lcd.print(" metres");
lcd.setCursor(0,1); // A la segona fila escrivim el temps.
lcd.print("temps:");
if(hores<10){ //Escrivim hores.Afegim un zero al davant si esta per sota de 10
    lcd.print("0");}
lcd.print(hores);
lcd.print(":");
if(minuts<10){//Escrivim minuts.Afegim un zero al davant si esta per sota de 10
    lcd.print("0");}
lcd.print(minuts);
lcd.print(":");
if(segons<10){//Escrivim segons.Afegim un zero al davant si esta per sota de 10
    lcd.print("0");}
lcd.print(segons);
}
}

```

```

void ISR_Blink() //Funció de la interrupció cada 1 segon
{
    //Si estem en el mode automàtic i no s'ha acabat la pel·lícula
    if((python=='0') && ((digitalRead(capturar)==HIGH) &&
(digitalRead(endavant)==HIGH) && (digitalRead(rebobinar)==LOW))) {
        segons=segons-1; //anem restant segons
        lcd.clear();//Netegem la pantalla
        lcd.setCursor(0,0); //Escrivim a la primera fila els fotogrames capturats
        lcd.print("Fotogrames:");
        lcd.print(fotogrames);
        lcd.setCursor(0,1); // A la segona fila escrivim el temps
        lcd.print("temps:");
        if(hores<10){ //S'executa la lògica per anar baixant el temps
            lcd.print("0");}
            lcd.print(hores);
            lcd.print(":");
        if(minuts<10 && minuts>=0){
            lcd.print("0");
        }else if(minuts<0 && hores>0){
            minuts=59;
            hores= hores-1;
        }else if(minuts<0 && hores<=0){
            minuts=0;
        }
        lcd.print(minuts);
        lcd.print(":");
        if(segons<10 && segons>=0){
            lcd.print("0");
        }else if(segons<0 && minuts>0){
            segons=59;
            minuts= minuts-1;
        }else if(segons<0 && minuts<=0 && hores>0){
            segons=59;
            minuts=59;
            hores=hores-1;
        }
        lcd.print(segons);
    }else if((python=='1')){//Si el codi python avisa que s'ha acabat la pel·lícula
        lcd.clear();//Netegem la pantalla
        lcd.setCursor(0,0); //Escrivim a la primera fila els fotogrames totals
        lcd.print("Fotogrames:");
        lcd.print(fotogrames);
        lcd.setCursor(0,1); // A la segona fila escrivim el temps
        lcd.print("Editant imatges");//Indiquem que el procés d'edició està en marxa
    }
}

```

B. PROGRAMA PYTHON

```

import serial#Llibreria per comunicacions per port sèrie
import time #Llibreria per poder fer delays
import cv2 # Llibreria OpenCv, per processar imatges
import glob #llibreria per accedir a carpetes i arxius
import os #llibreria per obtenir dades dels arxius de les carpetes
import numpy as np #Llibreria per operar amb matrius i operacions matemàtiques
import sys #Llibreria que ens permet fer el shutdown del programa

ts=0 #variable que guarda la data de la creació de la foto anterior
comptador=0 #Variable per comptar fotogrames negres
rang_negre_baix = np.array([0,0,0], dtype=np.uint8)#Establim límit mínim màscara
rang_negre_alt = np.array([10,10,10], dtype=np.uint8)#Límit màxim màscara
captura=0 #Variable per indicar que encara s'estan fent fotos
foto=1#Variable per saber la foto en la que estem
x_tot=0 #Valor acumulat de la coordenada x del rectangle delimitador
y_tot=0 #Valor acumulat de la coordenada y del rectangle delimitador
w_tot=0 #Valor acumulat de l'ampada w del rectangle delimitador
h_tot=0 #Valor acumulat de l'alçada h del rectangle delimitador

#Demanam informació a l'usuari per pantalla

print("Introdueix el port COM de l'arduino:")
Port=input()
print("Introdueix la ruta de la carpeta d'origen:")
Carpeta_origen = input()
print("Introdueix la ruta de la nova carpeta:")
Carpeta_nova=input()

arduino=serial.Serial(Port,9600)#Establim el port sèrie de l'arduino

while captura==0: #Mentre estem en el mode captura
    list_of_files = glob.glob(r'+Carpeta_origen'+ '*.jpg')#Llegim arxius jpg
    if list_of_files!=[]:#Si la carpeta no està buida
        latest_file = max(list_of_files, key=os.path.getctime)#Agafem l'últim arxiu
        imatge=cv2.imread(latest_file)#Llegim la imatge
        mask = cv2.inRange(imatge, rang_negre_baix, rang_negre_alt)#Creem màscara
        moments = cv2.moments(mask) #Generem el moment dels píxels dintre del rang
        area = moments['m00'] #Calculem l'àrea dels objectes que detecta la càmera
        if area >= 5300000000.0:#Si detecte un fotograma completament negre
            comptador=comptador+1 #Incrmentem comptador
            if comptador==3:#Si detecta tres fotogrames negres seguits
                arduino.write(b'1')#Enviem un "1" a l'arduino per parar la captura
                captura=1#Indiquem que s'ha acabat la captura
                comptador=0#Reiniciem comptador
            else:#Si és un fotograma normal reinicia el comptador
                comptador=0

for img in glob.glob(r'+Carpeta_origen'+ '*.jpg'):#Llegim fotos de carpeta d'origen
    fts=os.path.getmtime(img) #Llegim la data de creació de la foto
    if fts > ts: #Si la data de creació de la foto actual és superior a l'anterior.
        ts=fts #Guardem la data actual amb la variable antiga
        imatge=cv2.imread(img)#Llegim la imatge
        num_rows, num_cols = imatge.shape[:2]#Agafem el nombre de files i columnes
        #Calculem la matriu de rotació considerant el punt mig i per girar 180°
        rotation_matrix = cv2.getRotationMatrix2D((num_cols/2, num_rows/2), 180, 1)
        #Girem la imatge
        imatge_girada =cv2.warpAffine(imatge, rotation_matrix, (num_cols,num_rows))
        fotograma=imatge_girada.copy()#Fem una còpia de la imatge girada
        #La convertim de RGB a escla de grisos per detectar el fons negre
        imgray = cv2.cvtColor(imatge_girada,cv2.COLOR_BGR2GRAY)
        #Definim el llindar del color
        ret,thresh = cv2.threshold(imgray, 10, 255,cv2.THRESH_BINARY)
        #Trobem els contorns

```

```

img, contours, hierarchy = cv2.findContours(thresh, cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE)
#Trobem el contorn amb l'àrea més gran
contours = sorted(contours, key = cv2.contourArea, reverse = True)[:1]
cnt = contours[0] #Guardem els punts del contorn en una matriu
#Trobem les coordenades del rectangle delimitador
x,y,w,h = cv2.boundingRect(cnt)
if foto>1: #Excepte el fotograma
    #Si el retall difereix molt de la mitjana
    if (x>(x_mitjana+80) or x<(x_mitjana-80)):
        x=x_mitjana #Agafem les coodenades del retall anterior
    if (y>(y_mitjana+80) or y<(y_mitjana-80)):
        y=y_mitjana
    if (w>(w_mitjana+80) or w<(w_mitjana-80)):
        w=w_mitjana
    if (h>(h_mitjana+80) or h<(h_mitjana-80)):
        h=h_mitjana
    x_tot=x_tot + x #Valor acumulat de x
    x_mitjana=int(x_tot/foto) #Mitjana de x
    y_tot=y_tot + y #Valor acumulat de y
    y_mitjana=int(y_tot/foto) #Mitjana de y
    w_tot=w_tot + w #Valor acumulat de w
    w_mitjana=int(w_tot/foto) #Mitjana de w
    h_tot=h_tot + h #Valor acumulat de h
    h_mitjana=int(h_tot/foto) #Mitjana de h
    #Reatallem pel rectangle delimitador
    crop = fotograma[y_mitjana:y_mitjana+h_mitjana,x_mitjana:x_mitjana
+w_mitjana]
    IMG=str(foto) #Convertim la variable foto a string
    #Guardem cada imatge a la carpeta triada
    cv2.imwrite(r'+Carpeta_nova'+IMG+IMG+'.jpg',crop)
    foto=foto+1 #Incrementem el comptador de fotogrames

time.sleep(5) #Delay de 5 segons
sys.exit #Tanquem el programa

```