

Treball de Fi de Grau

**Grau en Enginyeria en
Tecnologies Industrials**

**Disseny i construcció d'un circuit per detectar
objectes de color**

MEMÒRIA

Autor: Judit Martí López
Director: Rosa Rodríguez Montañés
Convocatòria: Juny 2017



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

En aquest projecte s'ha dissenyat i construït un sistema electrònic basat en un microcontrolador PIC que, a partir d'un sensor detector de color i proximitat, identifica els components RGB del color de l'objecte acostat.

Les dades generades pel sistema són presentades a l'usuari a través d'una aplicació per telèfon mòbil Android creada a tal efecte. La comunicació entre el sistema electrònic dissenyat i el telèfon mòbil es realitza a través de Bluetooth.

Sumari

RESUM	1
SUMARI	3
1. GLOSSARI	5
2. PREFACI	7
2.1. Origen i motivació del projecte	7
2.2. Requeriments previs	7
3. INTRODUCCIÓ	9
3.1. Objectius del projecte	9
3.2. Abast del projecte	9
4. COMPONENTS	11
4.1. Microcontrolador PIC16F690	11
4.1.1. Característiques generals	11
4.1.2. Diagrama de blocs	12
4.1.3. Pinout	12
4.1.4. UART	13
4.1.5. Interrupcions	14
4.1.6. MPLAB i PICkit2	16
4.2. Pantalla LCD	16
4.3. Mòdul Bluetooth	17
4.4. Convertidor de nivell de voltatge	18
4.5. Regulador de voltatge	19
5. SENSOR DE COLOR APDS9960	20
5.1. Protocol I ² C	21
5.2. Descripció del funcionament	24
5.3. Detecció de proximitat	25
5.4. Detecció de colors i llum ambient	26
5.5. Registre ENABLE	27
5.6. Funcions a destacar	28
6. APLICACIÓ ANDROID	30
6.1. Introducció al MIT <i>App Inventor 2</i>	30
6.2. Aplicació dissenyada per aquest projecte	33

7. IMPLEMENTACIÓ SISTEMA	36
7.1. Connexions dels components.....	36
7.1.1. PICKit2 i PIC16F690	36
7.1.2. Sensor de color APDS9960 i PIC16F690	36
7.1.3. Pantalla LCD i PIC16F690	37
7.1.4. Mòdul Bluetooth i PIC16F690	39
7.1.5. Alimentació amb pila del microcontrolador.....	40
7.2. Circuit final	40
7.3. Programació.....	41
8. FUNCIONAMENT I RESULTATS FINALS	45
8.1. Resultats mostrats a l'aplicació Android	46
9. PRESSUPOST	48
CONCLUSIONS	49
AGRAÏMENTS	50
BIBLIOGRAFIA	51

1. Glossari

CPU: de l'anglès *Central Processing Unit*, unitat central de processament. És un component de dispositius programables, com és un microcontrolador, que interpreta les instruccions contingudes en els programes i processa les dades.

EEPROM: de l'anglès *Electrically Erasable Programmable Read-Only Memory*. És un tipus de memòria ROM que pot ser programada, esborrada i reprogramada elèctricament.

EUSART: de l'anglès *Enhanced Universal Synchronous Asynchronous Receiver-Transmitter*. Dispositiu que converteix dades de paral·lel a sèrie o a la inversa per tal de transmetre o rebre informació, respectivament.

I/O: de l'anglès *Input/Output*, és a dir, Entrada/Sortida. És la configuració que poden tenir els pins dels components, a vegades només pot ser entrada o sortida i altres es pot configurar en funció del que es vol.

LCD: de l'anglès *Liquid Crystal Display*. És una pantalla fina i plana formada per un cert nombre de píxels col·locats davant una font de llum.

Microcontrolador: circuit integrat que inclou una CPU, unitats de memòria, ports d'entrada i sortida i perifèrics.

Perifèric: dispositiu auxiliar extern i independent que es connecta a la CPU.

PIC: de l'anglès *Peripheral Interface Controller*. És una família de microcontroladors fabricats per *Microchip Technology Inc.*

RAM: de l'anglès *Random Access Memory*. Memòria d'accés aleatori d'escriptura i lectura. És volàtil, és a dir, quan es deixa de tenir alimentació es perd.

ROM: de l'anglès *Read Only Memory*. Tipus de memòria electrònica de la qual només es pot llegir la informació que té gravada. Es pot utilitzar per guardar variables que no s'han de modificar. A diferència de la RAM, quan es deixa d'alimentar les dades segueixen guardades, no es perden, és a dir, és no volàtil.

SRAM: de l'anglès *Static Random Access Memory*. És un tipus de memòria RAM basada en semiconductors, capaç de mantenir les dades.

UART: de l'anglès *Universal Asynchronous Receiver-Transmitter*. Nom que s'utilitza de manera general per referir-se a l'EUSART.

2. Prefaci

2.1. Origen i motivació del projecte

La realització d'aquest projecte s'ha portat a terme com a Treball de Fi de Grau, necessari per concloure els estudis del Grau en Enginyeria en Tecnologies Industrials

Es va voler fer un projecte relacionat amb l'electrònica perquè anteriorment ja s'havia despertat l'interès per aquesta matèria en cursar alguna assignatura de projectes. Una d'aquestes assignatures es basava en la construcció i programació d'una figura de LEDs (Light Emitting Diode). A més, també s'ha cursat l'assignatura obligatòria del quart curs del grau, electrònica, una assignatura prou completa com per donar coneixements generals de diverses parts de l'electrònica.

En el moment de fer aquest projecte s'han estat cursant dues assignatures més optatives relacionades amb l'electrònica: una totalment pràctica en el que s'aprèn a emprar una gran varietat de dispositius electrònics bàsics, i l'altra basada en el disseny d'aplicacions per a microcontroladors. Les dues han estat molt útils per tenir encara més coneixements durant la realització d'aquest projecte.

2.2. Requeriments previs

El fet d'haver cursat l'assignatura obligatòria d'electrònica, com les optatives explicades anteriorment, facilita l'enteniment del camp de l'electrònica i ajuda a la realització d'aquest projecte. Per tant, es podria dir que és necessària una certa base de coneixements d'aquest camp.

3. Introducció

3.1. Objectius del projecte

L'objectiu principal d'aquest projecte és el disseny i construcció d'un sistema electrònic que, basat en un microcontrolador PIC i un sensor de color, és capaç de detectar objectes propers i els components RGB del seu color. Aquestes dades es mostren per una pantalla LCD i un telèfon mòbil. La comunicació es realitza amb senyals i protocol Bluetooth.

Així, altres objectius secundaris consisteixen en analitzar i aprendre a fer funcionar amb detall tots els components utilitzats, iniciar-se en la programació en C i crear una aplicació Android mitjançant el software *AppInventor2* del MIT (Massachusetts Institute of Technology). L'entorn de treball utilitzat per a la programació del microcontrolador és el MPLAB de microchip.

3.2. Abast del projecte

En aquest apartat s'expliquen les possibles limitacions que hi ha hagut durant la realització d'aquest projecte, ja que afecten tant al desenvolupament com al resultat del treball.

Primer, tot el material i components que s'han utilitzat són els que es tenen al laboratori d'electrònica de l'ETSEIB. La majoria d'aquests components són senzills ja que són utilitzats per fer pràctiques, però per aquest projecte han estat suficients i han permès assolir els objectius proposats.

En quant al microcontrolador en concret, s'ha utilitzat el llenguatge C per a la seva programació, amb el programa MPLAB, un programa gratuït del fabricant *Microchip*. Això pot ser una limitació en si mateix però en aquest projecte ha estat més que suficient. El microcontrolador utilitzat no disposa de possibilitat de depuració (debugger), fet que ha condicionat la manera de treballar amb el PIC durant el desenvolupament del projecte.

Finalment, en el tema de l'aplicació creada amb el programari *MIT AppInventor2* està clar que també limita les possibilitats per a l'aplicació resultant al ser una web concebuda amb fins didàctics. És una manera fàcil i gratuïta de començar en el món de la programació d'aplicacions per a Android i en aquest cas també ha estat suficient per l'assoliment dels objectius.

4. Components

En aquest apartat s'explicaran els components que complementen el circuit on l'element principal és el sensor de color. Aquest sensor s'explicarà en el següent capítol al ser una part central i complexa del projecte. Els components presentats aquí són el microcontrolador, la pantalla LCD, el mòdul Bluetooth, el regulador de tensió.

4.1. Microcontrolador PIC16F690

Un microcontrolador és un dispositiu electrònic fabricat en forma de circuit integrat que conté una CPU (unitat de processat), memòria de programa (ROM), memòria de dades (RAM) i una unitat d'entrada/sortida i perifèrics, que són circuits addicionals de control i comunicacions.

El microcontrolador utilitzat en aquest cas és el PIC16F690, de la família dels PIC (*Peripheral Interface Controller*), del fabricant *Microchip Technology Inc.* A continuació es detallaran les seves característiques.

4.1.1. Característiques generals

La taula següent s'ha extret del document *Datasheet* del microcontrolador en qüestió [1], on es poden veure les seves característiques generals.

Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	Comparators	Timers 8/16-bit	SSP	ECCP+	EUSART
	Flash (words)	SRAM (bytes)	EEPROM (bytes)							
PIC16F690	4096	256	256	18	12	2	2/1	Yes	Yes	Yes

Taula 4.1. Característiques generals del PIC16F690

Com s'observa a la taula, la memòria de programa és de 4096 paraules, on una paraula és un conjunt de 14 bits necessaris per encabir una instrucció bàsica del microcontrolador. Per altra banda, les memòries de dades SRAM, de l'anglès *Static Random Access Memory*, i EEPROM, de l'anglès *Electrically Erasable Programmable Read-Only Memory*, són de 256 bytes ambdues. El bus de dades té una amplada de 8 bits.

Així es pot veure que té 18 pins en els ports d'entrada/sortida, aquests són els que permeten comunicar el microcontrolador amb l'exterior. També té dos comparadors, dos timers de 8 bits i un de 16 bits.

4.1.2. Diagrama de blocs

A continuació es pot veure el diagrama de blocs del microcontrolador emprat al projecte, on es poden apreciar els diferents mòduls que té i com estan connectats entre sí.

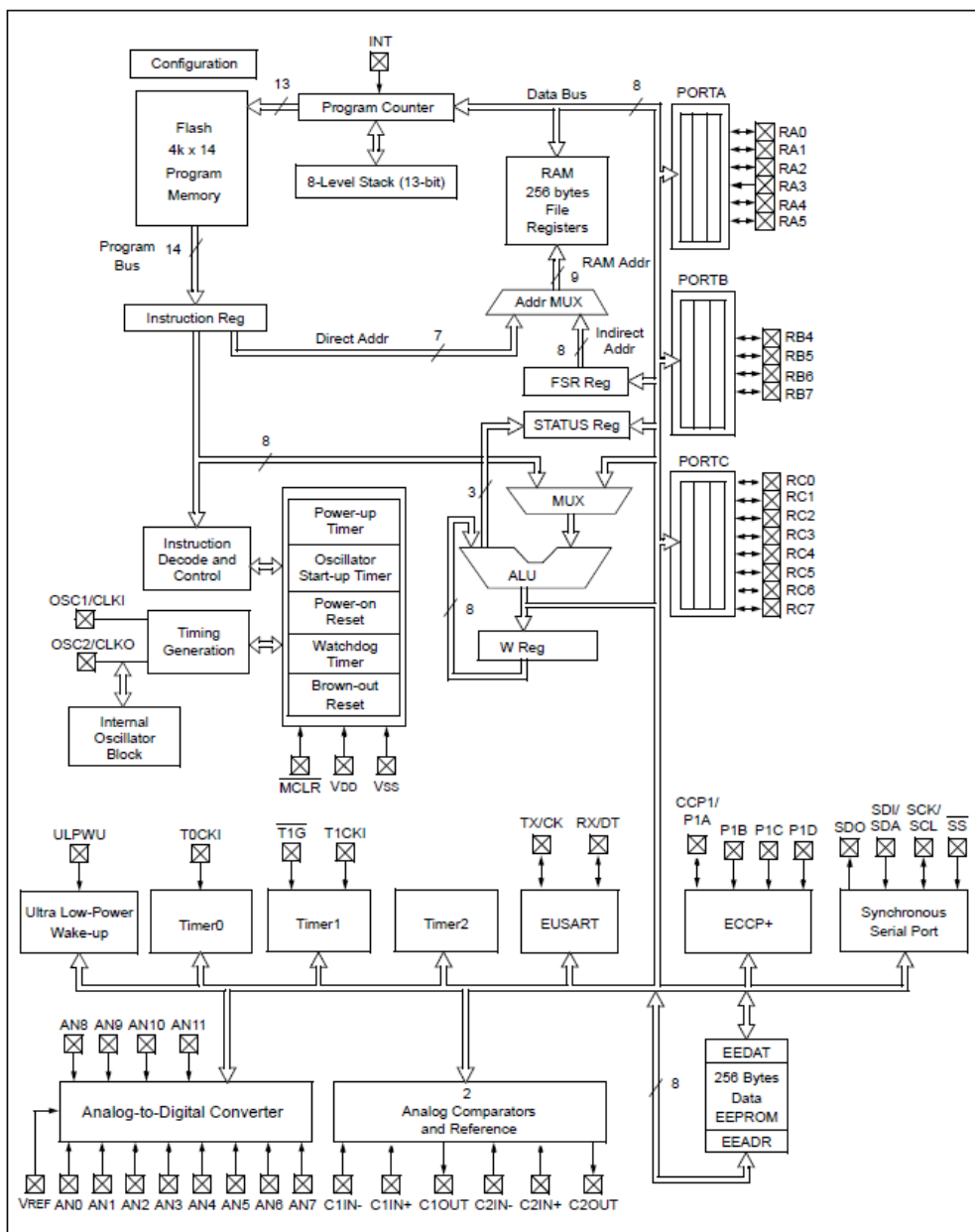


Figura 4.1. Diagrama de blocs del PIC16F690

4.1.3. Pinout

El pinout ens mostra la distribució dels senyals d'alimentació i del ports d'entrada i sortida

d'un microcontrolador, aquests ports permeten la comunicació amb l'exterior. En aquest cas és aquest:

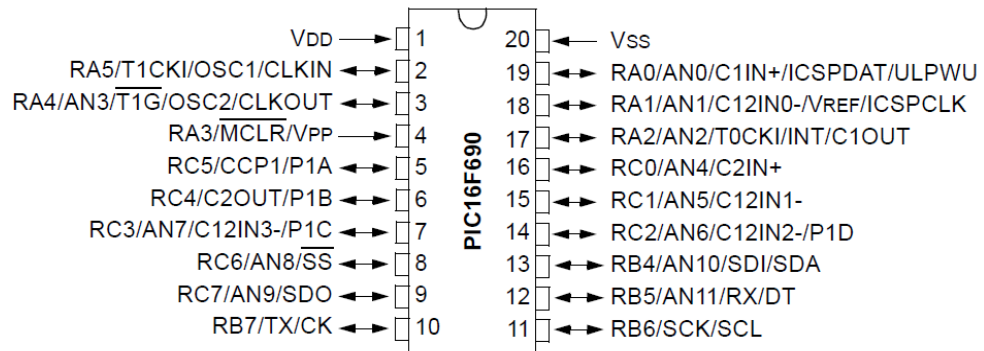


Figura 4.2. Diagrama de pins del PIC16F690

Com es veu a la figura, es tenen 20 pins, on dos dels quals són el d'alimentació, V_{DD} , i del terra, V_{SS} . La resta estan dividits en tres ports diferents, el PORTA, el PORTB i el PORTC, amb 5, 4 i 8 pins respectivament. La majoria d'aquests pins es poden comportar com entrades, quan reben informació de l'exterior, sortides, quan envien informació cap a l'exterior, o fins i tot bidireccionals, on es poden fer les dues coses en funció del programa que s'executi. De manera general els ports es configuren d'entrada o sortida de manera digital (segons un bit que li determini la direcció), sent un 0 per configuració de sortida i un 1 per entrada.

També es pot veure per la nomenclatura de la Figura 4.2 que, a part dels ports, els mateixos pins tenen altres funcions que porta implementades el microcontrolador, per utilitzar-les s'ha d'indicar la funció que es vol a cada pin en el moment de configurar el dispositiu.

4.1.4. UART

El mòdul UART, o EUSART com li diu Microchip (de l'anglès *Enhanced Universal Synchronous Asynchronous Receiver-Transmitter*), és un perifèric d'entrada/sortida (I/O de l'anglès *input/output*). És el que s'ha utilitzat per a la comunicació via Bluetooth entre el dispositiu mòbil i el microcontrolador. La seva principal funció és convertir les dades de paral·lel (banda interna del microcontrolador) a sèrie (comunicació externa) o a la inversa per tal de transmetre o rebre informació.

Els pins 10 i 12 del microcontrolador són, respectivament, les dues senyals per a les funcions per transmetre (sortida TX) i per rebre (entrada RX). En aquest treball s'ha utilitzat l'UART tant com a receptor com a transmissor. Concretament, per transmetre les dades al telèfon mòbil i poder visualitzar el color captat pel sensor i la seva composició RGB s'ha utilitzat com a transmissor; en canvi s'ha utilitzat com a receptor per a captar el que se li envia des del dispositiu mòbil amb l'aplicació Android.

En estat de repòs el senyal està a nivell alt V_{DD} (5V). La informació es transmet en bytes, és a dir, 8 bits. La trama consisteix en un bit d'inici, una paraula de 8 bits, un bit opcional de paritat i un bit de parada. Aquesta trama està esquematitzada a la *Figura 4.3*, on St és el bit d'inici (*Start bit*), del 0 al 8 són els bits del que es vol enviar (*Data bits*), [P] és el bit de paritat i finalment Sp és el bit de parada (*Stop bit*).

El bit d'inici provoca un flanc de baixada, és a dir, passa el senyal que estava a nivell alt cap a nivell baix. El bit de paritat indica si hi ha d'haver un número parell o senar d'1's a la paraula transmesa, això permet detectar errors però en aquest treball no s'ha utilitzat per no ser necessari. Finalment, el bit de parada es té en compte quan el senyal està cert temps a nivell alt un altre cop.

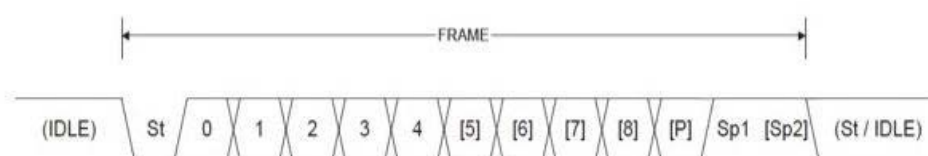


Figura 4.3. Esquema trama UART

4.1.5. Interrupcions

Una interrupció és una indicació d'un esdeveniment en el hardware o en el software d'un microcontrolador que provoca una ruptura de la seqüència d'instruccions que aquest realitza de manera que la CPU passa a executar una rutina especial de tractament d'aquest esdeveniment. Aquesta rutina especial s'anomena ISR (de l'anglès *Interrupt Service Routine*), després de la qual es torna al punt del programa en el que s'ha produït la ruptura.

Una de les interrupcions utilitzades en aquest treball és la generada pel *Timer0* del microcontrolador. Per poder utilitzar les interrupcions, i concretament aquesta, s'han d'habilitar les interrupcions globals i també les dels perifèrics. Per fer-ho s'utilitzen els registres i bits del PIC *INTCONbits.GIE* i *INTCONbits.PEIE*. A continuació (Figura 4.4) hi ha el diagrama de blocs del *Timer0*, que és útil a l'hora de dissenyar el servei d'interrupcions del programa.

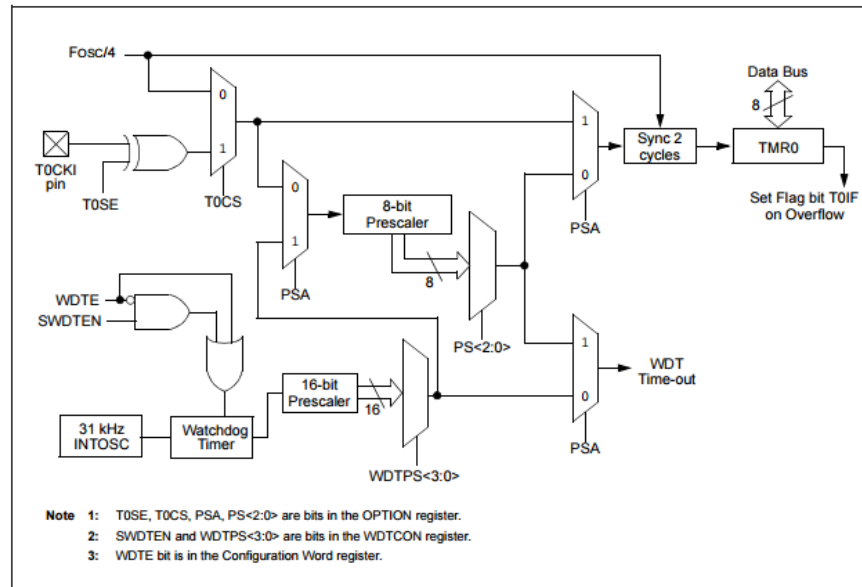


Figura 4.4. Diagrama de blocs del Timer0 del PIC16F690

També és útil el diagrama de blocs de les interrupcions, per saber així quines s'han d'habilitar en funció de la que s'utilitzi.

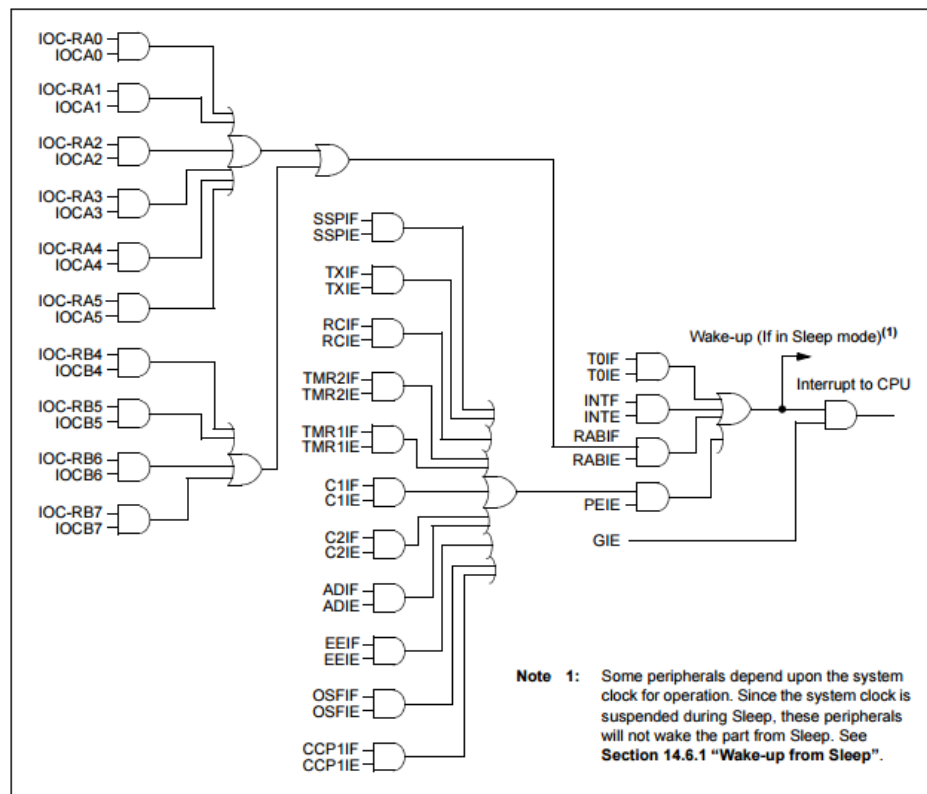


Figura 4.5. Diagrama de blocs de les interrupcions

4.1.6. MPLAB i PICKit2

Per a programar el microcontrolador s'han necessitat tant el software MPLAB com el hardware PICKit2 [2][3]. El primer és un editor IDE (de l'anglès *Integrated Development Enviroment*) gratuït ofert per *Microchip Technology Inc.* que permet programar en llenguatge C. Aquest software permet editar els arxius amb extensió .c, a més de tenir un preprocessor que realitza manipulacions abans de compilar. També conté un enllaçador que, com diu el seu nom, enllaça les llibreries referenciades amb el programa. Una vegada preprocessat, el compilador passa el codi C a llenguatge màquina, que és el codi que executa el microcontrolador, basat en 0 i 1. Un cop fet i depurat el programa s'utilitza el PICKit2 per carregar-lo a la memòria de programa ROM del microcontrolador des de l'ordinador. A la següent figura es mostren els pins del PICKit2, que aniran connectats directament al microcontrolador.

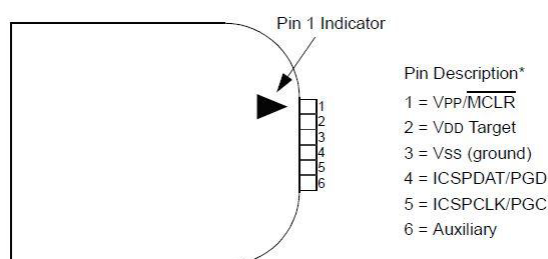


Figura 4.6. Pins del PICKit2

4.2. Pantalla LCD

S'ha utilitzat una pantalla LCD, de l'anglès *Liquid Crystal Display*, per visualitzar de manera senzilla les mostres preses pel sensor utilitzat, tot per anar fent proves abans de crear l'aplicació per al telèfon mòbil. El model emprat ha estat el Hitachi HD44780 controller de 16x2 [4].

Aquesta pantalla consta de 16 pins, dels quals V_{ss} és la connexió a terra i V_{dd} a 5V. També té el pin V_{ee} que serveix per alterar el contrast de la pantalla, el qual s'ajusta amb una resistència externa variable. Té tres pins de comandes de control: RS, R/W i E. El primer, RS (*Register select*) es posa a 1 per a que es puguin transmetre caràcters, ja que si es posa a 0 les dades que reb la pantalla s'entenen com a ordres. El senyal R/W (*Read/Write*) es posa a 1 per llegir la informació que reb de l'exterior i mostrar-la per la pantalla. Finalment el darrer d'aquests tres senyals és l'E (*Enable*) que serveix per activar o desactivar les funcions de la pantalla.

Els següents pins, del D0 al D7, corresponen al bus de dades i serveixen per transmetre o

rebre informació. Finalment els pins A i K són l'ànode i el càtode de la il·luminació de la pantalla, respectivament. A la següent figura es poden observar tots aquests pins.

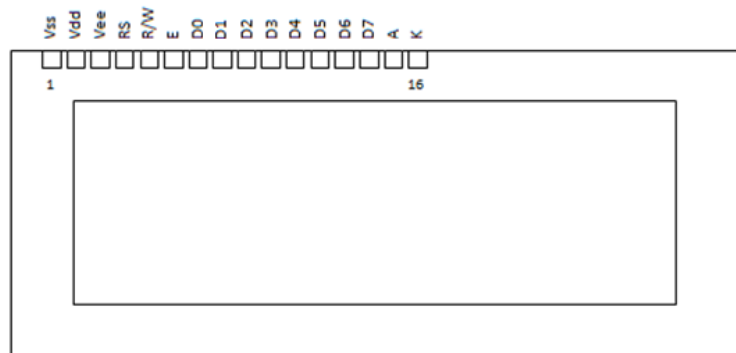


Figura 4.7. Diagrama de pins de la LCD HD44780

En la figura següent es poden veure els caràcters coneguts per la pantalla LCD i com fer que els mostri en el cas que el RS (*Register Select*) estigui a 1.

CG RAM (1)	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000			0	@	P	`	P				-	9	3	α	p	
xxxx0001 (2)			!	1	A	Q	a	q			.	7	4	ä	q	
xxxx0010 (3)			"	2	B	R	b	r			「	イ	ツ	フ	θ	
xxxx0011 (4)			#	3	C	S	c	s			」	ウ	テ	ε	ω	
xxxx0100 (5)			\$	4	D	T	d	t			、	エ	ト	μ	Ω	
xxxx0101 (6)			%	5	E	U	e	u			・	オ	ナ	1	σ	Ü
xxxx0110 (7)			&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111 (8)			'	7	G	W	g	w			ヲ	キ	ヲ	ラ	q	π
xxxx1000 (1)			(8	H	X	h	x			イ	ク	ネ	リ	」	×
xxxx1001 (2))	9	I	Y	i	y			ウ	ケ	ル	レ	」	γ
xxxx1010 (3)			*	:	J	Z	j	z			エ	コ	ハ	レ	j	≠
xxxx1011 (4)			+	;	K	[k	<			オ	サ	ヒ	ロ	×	π
xxxx1100 (5)			,	<	L	¥	l	l			カ	シ	フ	ワ	φ	π
xxxx1101 (6)			-	=	M]	m	>			ユ	ズ	ハ	ン	も	÷
xxxx1110 (7)			.	>	N	^	n	→			ヨ	セ	ホ	°	π	
xxxx1111 (8)			/	?	O	_	o	←			ウ	ツ	マ	°	ö	■

Figura 4.8. Taula de caràcters de la LCD

4.3. Mòdul Bluetooth

Un mòdul Bluetooth permet la recepció i transmissió d'informació sense la necessitat de cables ni connectors. Permet crear xarxes domèstiques sense fils útils per a compartir informació emmagatzemada en diferents dispositius. En aquest projecte s'ha utilitzat el

mòdul Bluetooth to Serial HC-05, del fabricant *ITEAD Studio* [5]. La seva funció és transformar les dades en format decimal a codi binari de 8 bits per transmetre-les del mòbil al PIC i a la inversa per transmetre-les del PIC al dispositiu mòbil.

En la següent figura es poden veure els pins del mòdul: STATE, RXD, TXD, GND, VCC i EN. En aquest treball no s'utilitzen ni STATE ni EN, que serveixen per indicar quin és el mode de treball (mestre o esclau) i per accedir al mode de configuració, respectivament.



Figura 4.9. Pins del mòdul Bluetooth HC-05

Els pins V_{cc} i GND (terra) serveixen per subministrar energia elèctrica al mòdul Bluetooth, aquest model pot rebre entre 3,3V i 6V. Els pins RXD i TXD són els que s'utilitzen per a la comunicació serial UART, concretament RXD és per a la recepció de dades i TXD per a la seva transmissió. Estan connectats als pins 10 i 12 del PIC, respectivament.

En la comunicació mestre-esclau hi ha un dispositiu (mestre) que té control unidireccional sobre els altres (esclaus). En aquest cas de comunicació via Bluetooth cal un mestre i un o més esclaus. Aquest model en concret pot actuar com a mestre i com a esclau. En aquest treball el dispositiu mòbil actua com a mestre amb l'app Android, busca i s'emparella amb el dispositiu Bluetooth desitjat, així el mòdul Bluetooth és l'esclau.

4.4. Convertidor de nivell de voltatge

Al utilitzar un sensor que suporta un voltatge màxim de 3,8 V, s'ha d'utilitzar un convertidor, en aquest cas de 5 V a 3,3 V. Aquest convertidor és bidireccional, és a dir, pot passar dades tant del nivell alt al baix com a la inversa.

Té un total de dotze pins, sis al costat d'alt voltatge i sis més al de baix voltatge, com es pot veure a la *Figura 4.10*. Els dos del mig són, per a cada voltatge, el terra, GND, i l'alimentació, LV o HV segons si és voltatge baix o alt, respectivament. Els altres quatre van per parelles i poden passar les dades de voltatge alt a baix i de baix a alt. En el cas d'aquest projecte s'han utilitzat tots els pins, ja que el sensor en té un total de sis, inclosos el terra i l'alimentació.

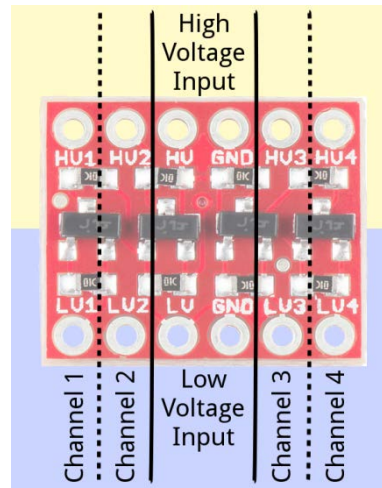


Figura 4.10. Pins del convertidor

4.5. Regulador de voltatge

Un regulador és un dispositiu que permet mantenir un nivell de tensió constant i adequat a les necessitats del circuit. Concretament s'ha utilitzat el model LM117, de la marca Texas Instruments [6] i el 7805.

En el cas d'aquest treball s'han utilitzat dos reguladors, un d'ells passa el voltatge de 5 V a 3,3 V per a tenir la tensió d'alimentació adient per al sensor. El segon s'utilitza només una vegada el microcontrolador té el programa final i tot està acabat per poder deixar d'utilitzar el PICkit2 i poder emprar el circuit autònomament en qualsevol lloc i moment. Aquesta segona utilització passa el voltatge de 9 V que dóna la pila utilitzada als 5 V que necessita el microcontrolador i tota la resta de components, excepte el sensor, per a funcionar.

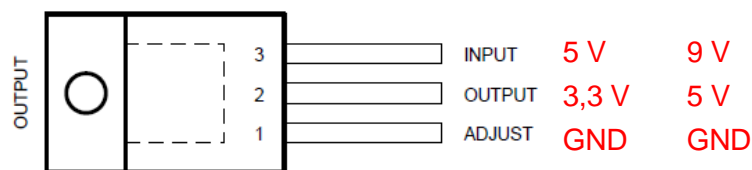


Figura 4.11. Pins del regulador LM117 i voltatges per al sensor i resta de components

5. Sensor de color APDS9960

Un sensor és un dispositiu que converteix senyals físiques en senyals elèctriques. Aquest en concret, el APDS9960 de la marca *Avago Technologies* [7], compta amb detecció de moviment, de proximitat, intensitat de llum ambiental i color. En el cas d'aquest projecte no s'utilitzarà la detecció de moviment, però sí la de proximitat i color. El protocol de comunicació que utilitza és el I²C.

La detecció de color dóna dades de la intensitat de la llum i la quantitat de vermell, verd i blau que conté. A continuació es pot veure el diagrama de blocs de com funciona aquest sensor, tant la part de la llum com la dels moviments. Aquesta arquitectura permet a les aplicacions mesurar de manera precisa la llum ambiental i el color.

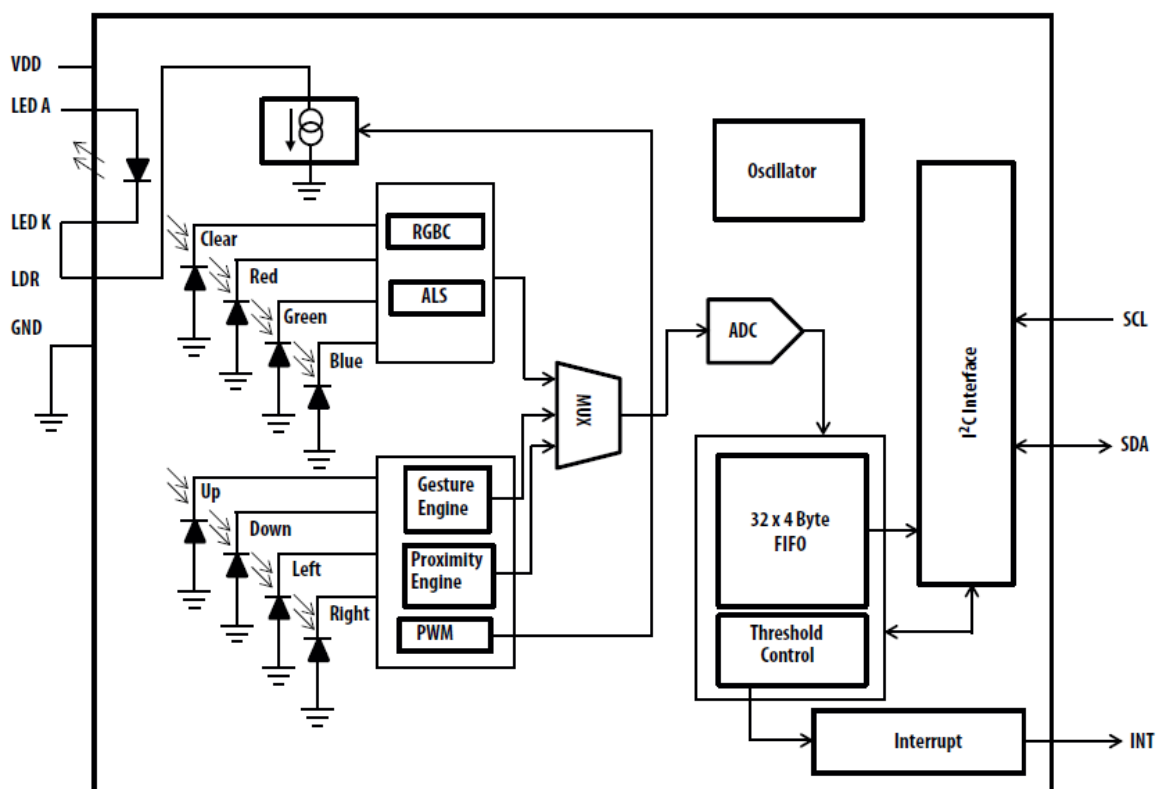


Figura 5.1. Diagrama de blocs del funcionament del sensor APDS9960

A les especificacions es pot veure que aquest component no pot treballar a més de 3,8 V, per això s'ha hagut d'utilitzar un regulador, explicat en l'apartat anterior, que converteix els 5 V d'alimentació del PICkit a 3,3 V. A més, al tenir més d'un pin que necessita aquesta tensió més baixa, s'ha hagut d'utilitzar un convertidor de 5 V a 3,3 V per als senyals lògics d'entrada i sortida del sensor. Una vegada solucionat aquest petit problema, el sensor ja pot

funcionar amb normalitat.

Els pins d'aquest sensor són VL, GND, VCC, SDA, SCL i INT. El primer pin, VL, és l'alimentació opcional del LED infraroig. En el cas d'aquest treball està connectat ja que aquest LED és el que fa servir la detecció de proximitat, és a dir, si la llum que ha emès el LED rebota i torna dins del sensor vol dir que hi ha un objecte a prop. Els pins GND i VCC són, respectivament, el terra i l'alimentació general del sensor (en aquest cas a 3,3 V gràcies al regulador).

Finalment, es tenen els tres pins de dades: SDA és el que transmet les dades, per això va connectat al pin SDA del microcontrolador, concretament el 13; SCL és el rellotge, que també va connectat al pin del rellotge del microcontrolador, concretament l'11; i per acabar es té el pin INT, que també coincideix amb el pin d'interruptió externa del microcontrolador, el pin 17 i el que fa són les interrupcions que, en el cas d'aquest projecte, s'activaran quan algun objecte s'apropi al sensor.



Figura 5.2. Pins del sensor APDS9960

5.1. Protocol I²C

Ja s'ha parlat sobre un altre tipus de comunicació i transmissió de dades, la seguida entre el microcontrolador i el mòdul Bluetooth, i que utilitza el mòdul UART. En aquest cas el protocol I²C és un altre tipus de comunicació que és el que utilitzen una gran majoria de sensors del mercat. Els seus principals avantatges són que és igual per a tots aquests sensors i això fa que la mateixa llibreria es pugui emprar per tots, només canviant l'adreça del sensor corresponent i la informació del registres particulars de cada cas. Això fa que es puguin emprar diversos components alhora connectats només a dos pins del microcontrolador i en funció d'aquesta adreça es decideix quin s'utilitza a cada moment. En el cas d'aquest projecte no és necessari utilitzar diferents sensors. L'adreça en concret del sensor APDS9960 és 0x39 en hexadecimal.

El protocol I²C està dissenyat com un bus mestre-esclau, és a dir, la transferència de dades sempre l'inicia el mestre i l'esclau és el que reacciona. En aquest cas el mestre és el microcontrolador, ja que és el que genera la senyal de rellotge. La comunicació passa per

una sèrie d'estats definits per si el mestre fa un canvi de nivell o no, és a dir, en cada flanc del rellotge s'agafa la dada que li passa l'esclau. A la següent figura es poden veure les condicions de Start i Stop, la primera quan les dades passen a nivell baix i la segona quan passen a nivell alt, les dues quan el rellotge està a nivell alt.

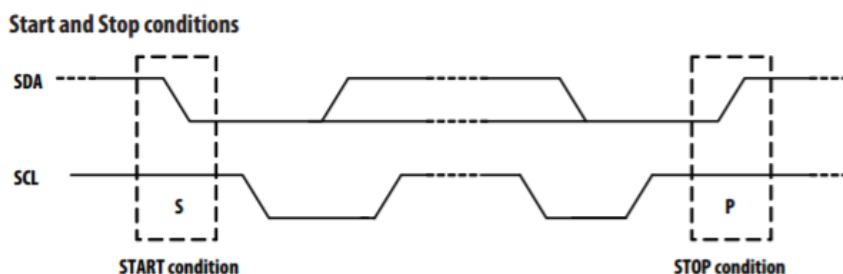


Figura 5.3. Condicions de Start i Stop

Una altra condició és la d'acknowledgement de l'esclau, la qual es basa en que quan s'acaba la transmissió del darrer bit, el bit de dades passa a nivell baix i és mantingut per l'esclau en aquest estat durant el següent flanc del rellotge, per seguir amb la comunicació just aquest acaba. Aquesta condició serveix per comprovar si el protocol funciona correctament, i la comunicació entre el mestre i l'esclau és correcta. A les dues següents figures es pot veure justament aquesta condició explicada (Figura 5.4), però també la contraria (Figura 5.5), el no acknowledgement.

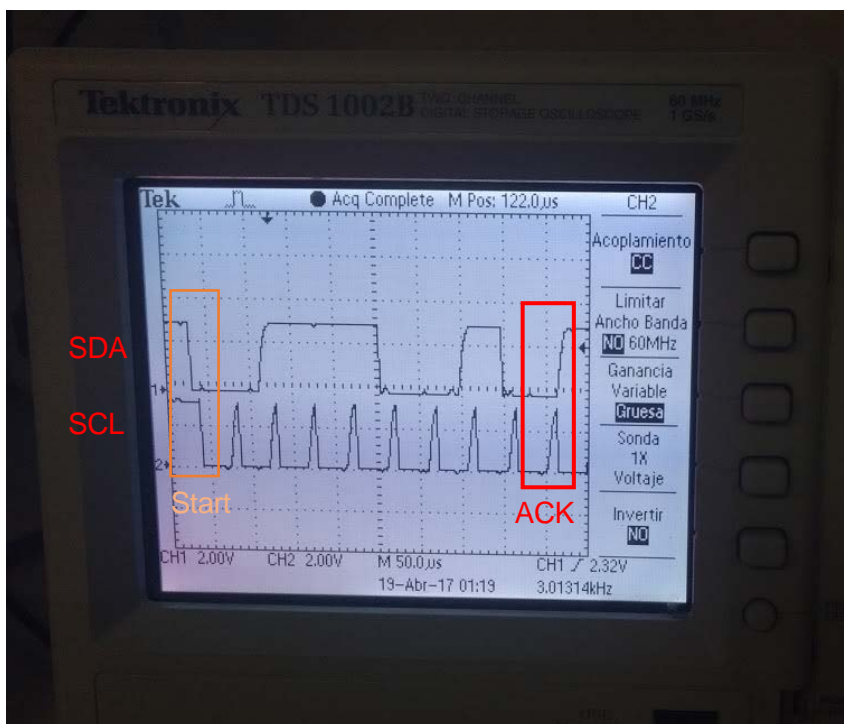


Figura 5.4. Condició d'acknowledgement de l'esclau

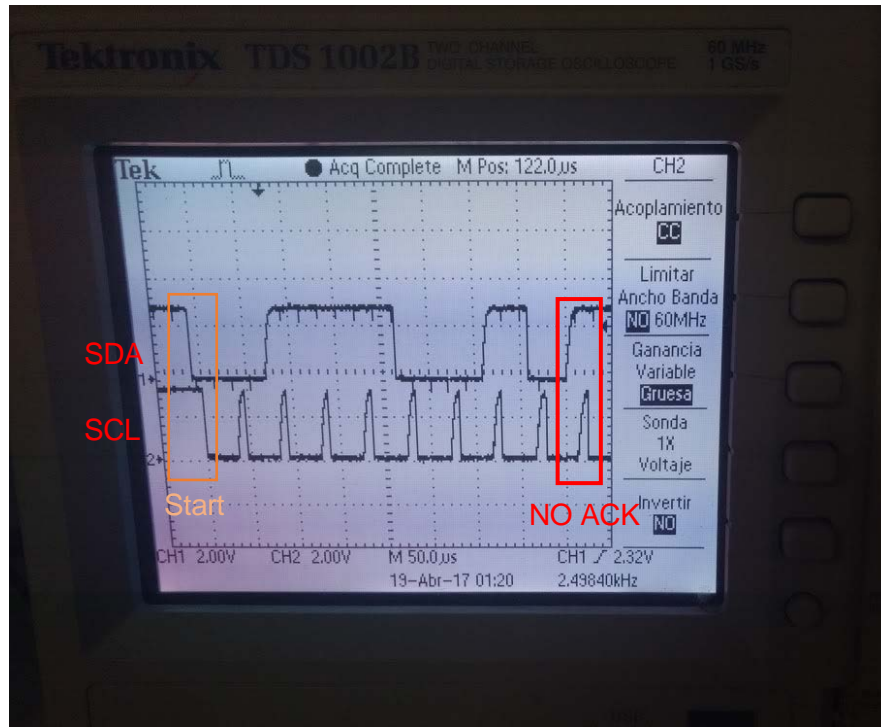


Figura 5.5. Condició de no acknowledgement de l'esclau

Finalment, es veurà a les dues següents figures el protocol basat en les condicions ja explicades més les dades a transmetre. Com es pot veure a la Figura 5.6, S és la condició de Start; W és el cas d'escriure, on hi ha un 0; A és l'acknowledgement; Sr és la condició de Start repetida; R és 1 i és el cas de llegir. Les parts sombreades es refereixen a la comunicació esclau-mestre mentre que les parts blanques són mestre-esclau. La Figura 5.7 il·lustra un exemple d'una comunicació qualsevol.

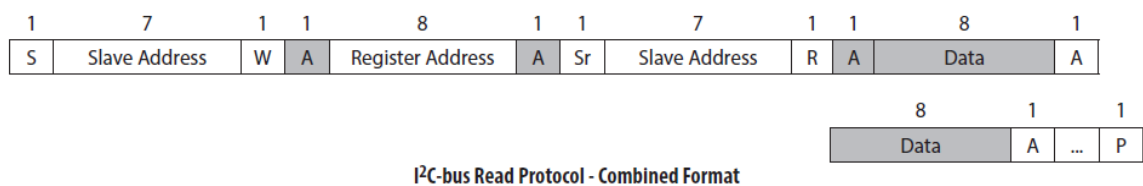


Figura 5.6. Protocol I²C

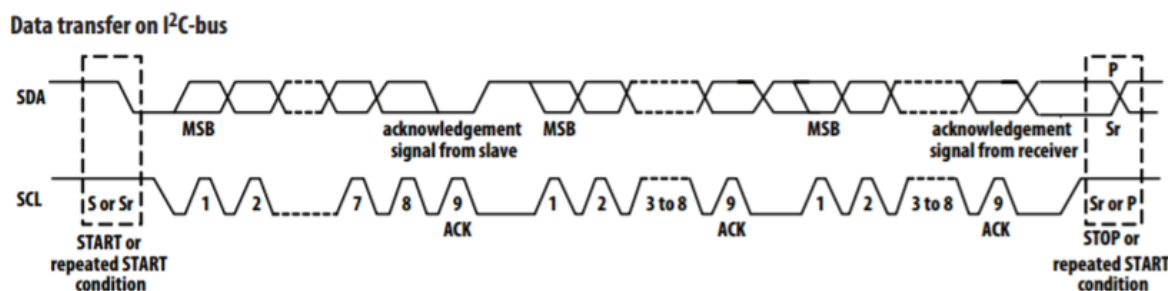


Figura 5.7. Exemple general d'una comunicació segons protocol I²C

5.2. Descripció del funcionament

La detecció de moviment, proximitat, color i llum ambient està controlada per una *state machine*, basada en una sèrie d'estats que passen d'un a l'altre en funció d'unes condicions. En aquest primer diagrama, *Figura 5.8*, es poden veure els estats d'una manera simplificada.

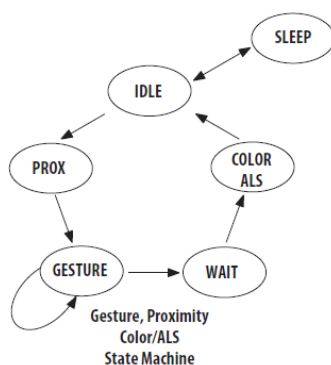


Figura 5.8. Diagrama d'estats simplificat del sensor de color

El diagrama que ve a continuació, *Figura 5.9*, explica d'una manera més detallada aquests estats i com es passa d'uns als altres. Es comença amb un POR, és a dir, un *Power on Reset*, que és quan el dispositiu s'inicialitza; passa directament a un estat de baix consum anomenat *Sleep*. Aquest baix consum consisteix en que l'oscil·lador i altres circuits interns no estan actius. Si hi ha una operació d'accés extern durant aquest estat, el nucli s'activa de manera temporal. Quan s'ha activat el Power ON bit (PON), l'oscil·lador i la resta de circuits interns s'activen, encara que el consum segueix sent baix fins que comencin les operacions. A continuació s'activa algun dels modes (proximitat, moviment o llum ambient) i hi ha una pausa per sortir del mode *Sleep*. Immediatament després s'entra dins de les operacions del mode seleccionat. Si hi ha més d'un mode activat, es segueix un cert ordre: IDLE, proximitat, moviment, espera, color i *Sleep* o IDLE, en funció de les opcions.

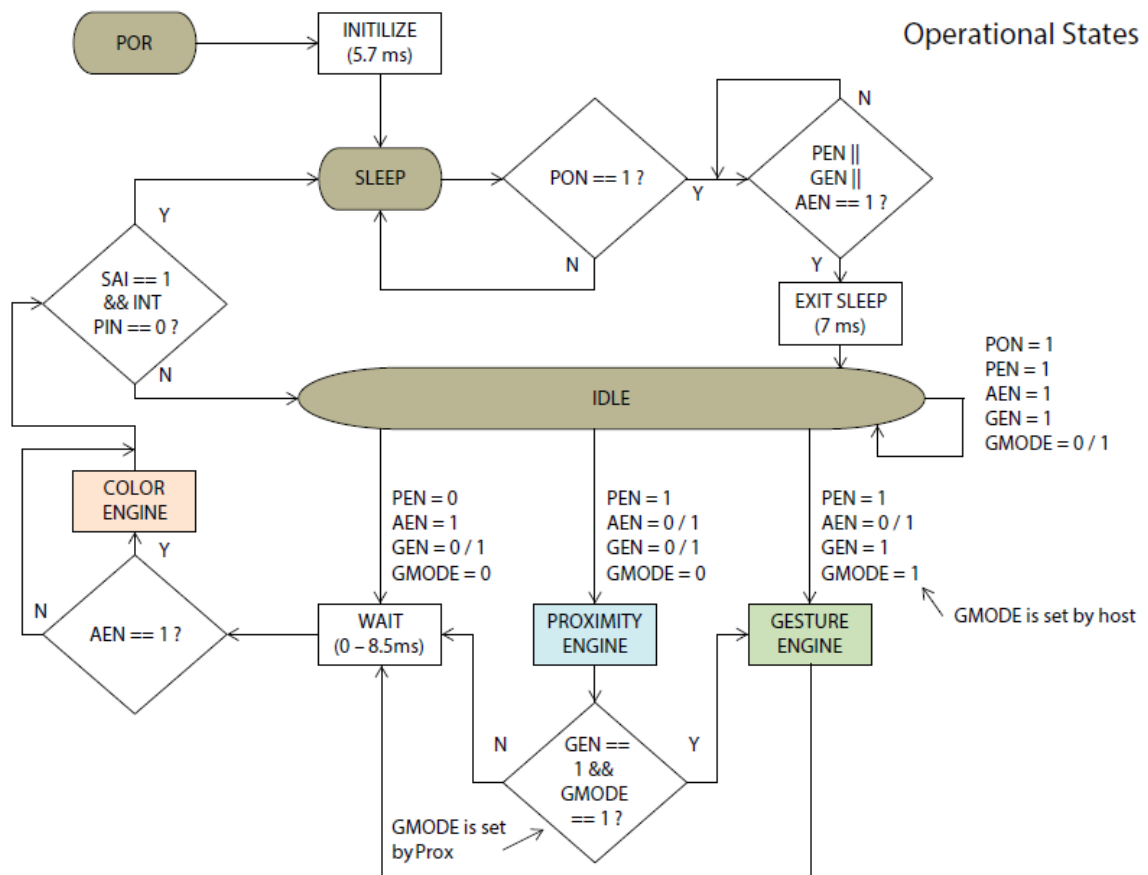


Figura 5.9. Diagrama d'estats detallat

5.3. Detecció de proximitat

Els resultats de la detecció de proximitat estan afectats per tres factors: l'emissió del LED infraroig, la recepció d'infrarojos i factors ambientals com la distància a l'objecte i la reflexivitat de la superfície d'aquest. Tot comença amb la detecció d'infrarojos de quatre fotodíodes i acaba amb el resultat de proximitat de 8 bits en el registre PDATA. El senyal d'aquests fotodíodes es combina, amplifica i compensa per optimitzar el rendiment. Aquests fotodíodes també són utilitzats en el mode de detecció de moviments, encara que no s'utilitza aquesta funcionalitat en aquest projecte.

Una interrupció és generada amb cada nou resultat de proximitat o en el moment en que els resultats de proximitat passen o queden per baix dels nivells fixats als registres PIHT i/o PILT. Per no causar falses interrupcions hi ha un filtre que només deixa que s'activen en el cas en que es superi el valor fixat en el registre PPERS. A continuació, a la *Figura 5.10*, es pot veure amb més detall els estats pels que passa el procés de detecció de proximitat.

PROXIMITY ENGINE

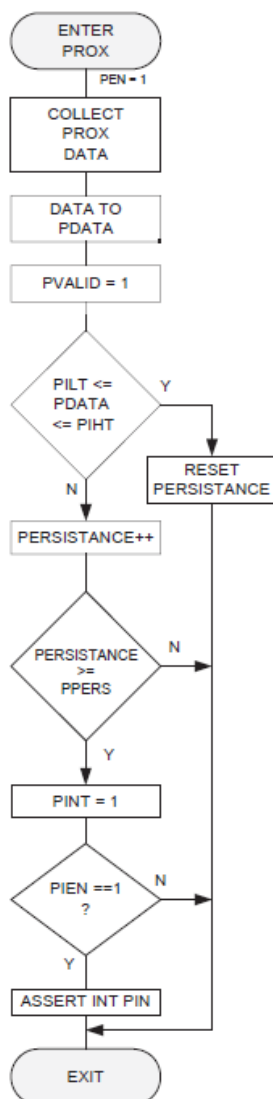


Figura 5.10. Diagrama de detecció de proximitat detallat

5.4. Detecció de colors i llum ambient

La funcionalitat de detecció de colors i llum ambient empra un array (un tipus d'estructura de dades en C que pot guardar una sèrie d'elements del mateix tipus) per mesurar la quantitat de vermell, verd i blau i a més la llum ambient. La detecció comença amb la rebuda als fotodíodes i acaba amb els resultats de 16 bits en el registre RGBC.

Una interrupció pot ser generada en qualsevol moment en que els resultats passin o estiguin per baix d'uns certs nivells fixats dels registres AILTL/AIHTL i/o AILTH/AIHTH. Com en l'anterior apartat, també existeix un filtre per evitar possibles falses interrupcions, així només

s'activa la interrupció en el cas que es sobrepassi el valor fixat per APERS. La *Figura 5.11* mostra aquest funcionament.

COLOR/ALS ENGINE

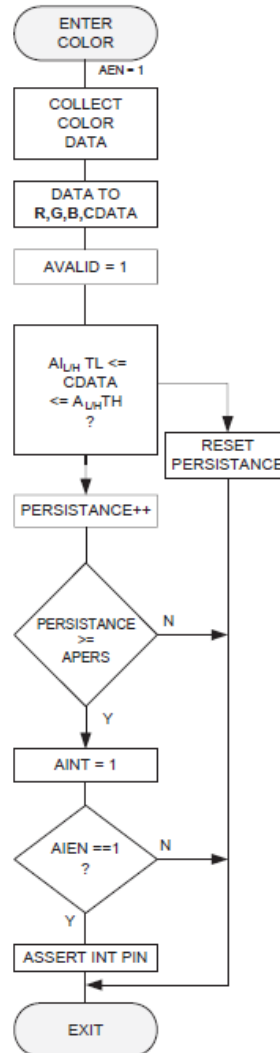


Figura 5.11. Diagrama de detecció de color i llum ambient detallat

5.5. Registre ENABLE

Aquest sensor està controlat i monitoritzat per registres de dades i d'ordres. Aquests registres donen una gran varietat de funcions per al control i poden llegir-se per determinar els resultats. Un dels registres importants és l'ENABLE, ja que és el que permet activar o no el propi dispositiu, els estats (proximitat, color, llum ambient, moviment) i les interrupcions, i es troba a l'adreça 0x80 en hexadecimal. Concretament, aquest registre està format per 8 bits, cada bit activa o desactiva certa funció del dispositiu. A la taula següent es pot veure

cada bit amb la seva funció:

Identificador	Bit	Descripció	Valors en aquest treball
Reserved	7	Reservat, s'escriu un 0.	0
GEN	6	Detecció de moviments habilitada, es pot activar si es donen les condicions.	0
PIEN	5	Interrupció de proximitat habilitada, es poden activar les interrupcions per proximitat en el cas d'haver les condicions adequades.	1
AIEN	4	Interrupció de llum ambient habilitada, es poden activar les interrupcions per llum ambient si es donen les condicions adequades.	0
WEN	3	Habilitació de l'espera.	0
PEN	2	Habilitació de la proximitat, si es donen les condicions s'activarà aquest mode.	1
AEN	1	Detecció de llum ambient habilitada, s'activa si es donen les condicions.	1
PON	0	Habilita l'oscil·lador intern perquè el dispositiu pugui funcionar.	1

Taula 5.1. Taula descriptiva dels bits del registre enable

5.6. Funcions a destacar

Les llibreries i arxius emprats per a fer funcionar aquest projecte estan a l'annex però en aquest apartat es vol veure en més detall alguna funció relacionada amb el sensor i els seus registres i com llegeix i escriu dades (amb el protocol I²C).

Primer, es pot destacar la funció *setMode*, que serveix per decidir en quin mode funcionarà el dispositiu, és a dir, si detectarà moviments, llum ambient, color i/o proximitat. Això s'aconsegueix agafant el bit adequat del registre ENABLE, explicat en l'anterior apartat, i donant-li el valor que es vol. Per exemple, si es vol activar la detecció de proximitat, es donarà com a valors d'entrada de la funció un 2, que és el bit d'aquest mode, i també un 1 per dir que es vol que s'habiliti (0 en el cas que es vulgui deshabilitar).

També són importants les funcions que s'utilitzen per llegir o escriure dades mitjançant el protocol I²C, ja que s'utilitzen sovint en el programa principal. Aquestes funcions són *wireWriteDataByte* per a escriure i *wireReadDataByte* per a llegir. Es poden destacar perquè dins tenen tots els estats pels que passa una transmissió I²C, com es mostra a la *Figura*

5.12, com ja s'ha explicat a l'apartat corresponent d'aquest capítol.

```
int wireWriteDataByte(unsigned char reg, unsigned char val)
{
    i2c_start();
    i2c_out_byte(APDS9960_I2C_ADDR_W);
    i2c_ack();
    i2c_out_byte(reg | 0x80);
    i2c_ack();
    i2c_out_byte(val);
    i2c_ack();
    i2c_stop();

    return 1;
}

unsigned char wireReadDataByte(unsigned char reg)
{
    /* Indicate which register we want to read from */
    char val = 0;
    i2c_start();
    i2c_out_byte(APDS9960_I2C_ADDR_W);
    i2c_ack();
    i2c_out_byte(reg | 0x80);
    i2c_ack();
    i2c_start();
    i2c_out_byte(APDS9960_I2C_ADDR_R);
    i2c_ack();
    val = i2c_in_byte();
    i2c_nack();
    i2c_stop();
    return(val);
}
```

Figura 5.12. Funcions `wireWriteDataByte` i `wireReadDataByte`

6. Aplicació Android

En aquest treball s'ha dissenyat una aplicació per a un telèfon mòbil amb sistema operatiu Android. Com ja s'ha explicat, per transmetre o rebre dades s'utilitza un mòdul Bluetooth. La funció principal d'aquesta aplicació és rebre les dades que agafa el sensor i mostrar-les, així com també fa la pantalla LCD. Com a model final s'han incorporat diverses funcions més, com generar un color, que es mostri per pantalla i utilitzar la pantalla del mòbil com a objecte per a que el sensor detecti el color; una altra funció és que generi el color amb les dades que rep del sensor, per així comparar el que s'ha generat per mostrar-li i el que ha rebut.

Per al disseny d'aquesta aplicació s'ha utilitzat la web *App Inventor 2* [8], que és un software gratuït dissenyat pel MIT (*Massachusetts Institute of Technology*). Aquesta web permet el disseny d'aplicacions per a mòbils Android de manera senzilla i intuïtiva, qualsevol persona pot fer-la servir sense saber res de programació d'aplicacions.

6.1. Introducció al MIT *App Inventor 2*

Primer, per accedir, s'ha d'entrar amb un compte de Google i després ja es pot crear un nou projecte. Un cop dins d'aquest projecte, es tenen dos entorns de treball. Un d'aquests entorns és el dissenyador, *Figura 6.1*, (al botó de dalt a la dreta *Designer*) i l'altre és el de blocs, *Figura 6.3*, (al botó de dalt a la dreta *Blocks*). El primer serveix per veure i modificar com queda l'aplicació i afegir elements que faran diferents funcions. L'apartat de blocs és on es programen tots aquests elements que s'han afegit al dissenyador, és a dir, se'ls hi dona una funció segons unes condicions. Per veure millor aquestes dues pantalles, els seus elements i com queda una vegada dissenyat s'han fet captures amb l'aplicació del treball ja acabada.

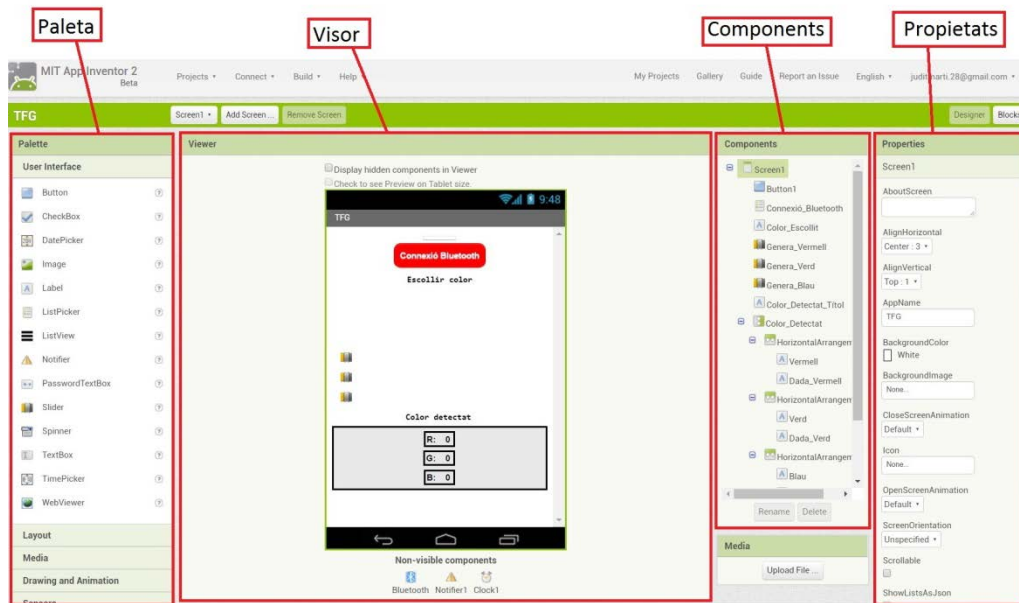


Figura 6.1. Pantalla per a dissenyar l'aplicació

En la *Figura 6.1* es pot veure com està organitzat l'entorn dissenyador. En el cas de la paleta, que està dividida en temes, permet seleccionar els elements que es volen incloure a l'aplicació. Així, es poden incloure elements visibles com botons, textos i *sliders* (controls lliscants) com també elements interns del telèfon mòbil com són el Bluetooth, notificadors o el propi rellotge. Tots aquests elements es posen arrossegant-los de la paleta fins la part que representa la pantalla del visor. A la *Figura 6.2* es tenen tots els elements utilitzats i a quina subdivisió es troben.

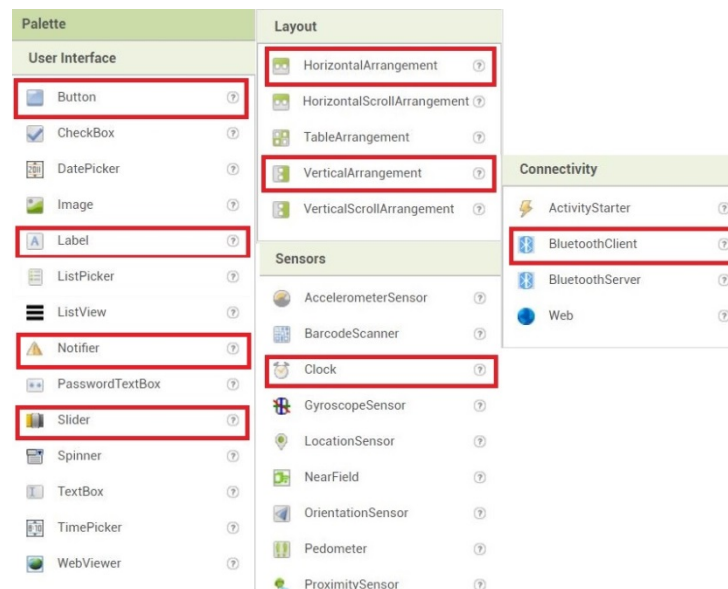


Figura 6.2. Elements utilitzats en l'aplicació

Cada element serveix per diferents operacions, que es descriuen a continuació.

Element	Funció
Label	Mostra un text o fixat o que canvia segons el que es programa
ListPicker	Botó que al ser presionat mostra una llista
Notifier	Permet mostrar quadres amb alertes
Slider	Barra que es pot deslligar per escollir diferents valors
HorizontalArrangement	Permet organitzar els elements de manera horitzontal
VerticalArrangement	Permet organitzar els elements de manera vertical
Clock	Utilitza el rellotge del dispositiu Android
BluetoothClient	Funcions Bluetooth del dispositiu Android

Taula 6.1. Elements utilitzats en l'aplicació amb la seva funció

Tornant a la *Figura 6.1*, el visor permet veure com queda l'aplicació i modificar l'ordre dels elements. Tot és molt senzill perquè simplement arrossegant es pot modificar. La part dels components només permet canviar els noms de cada component per a que tot sigui més clar i també eliminar-los. Els elements s'inclouen de manera automàtica a aquest apartat. Finalment, la part de propietats permet modificar diferents aspectes de cada element com són el color, la mida, la lletra, entre d'altres propietats. Una propietat important en aquesta aplicació és la dels *sliders*, en que s'ha de posar el valor mínim i màxim en funció del lloc on estarà el *slider*, ja que dependrà totalment del que es vol que faci l'aplicació.

Com s'ha dit anteriorment, apretant els botons de *Design / Blocks* es pot passar d'una interfície a una altra. En aquest cas la *Figura 6.3* mostra la pantalla dels blocs. En aquest entorn es pot donar funció a tots els elements que s'han inclòs en la pantalla de disseny. La part de *Blocks* classifica els diferents tipus de blocs en temes per poder-los trobar fàcilment. Tot és molt intuïtiu ja que només s'han d'arrossegar els blocs al visor i anar encaixant-los entre si. També hi ha una llista dels elements utilitzats i si es passa el cursor per damunt d'aquests, també tenen un conjunt de blocs específics amb funcions diferents a les generals.

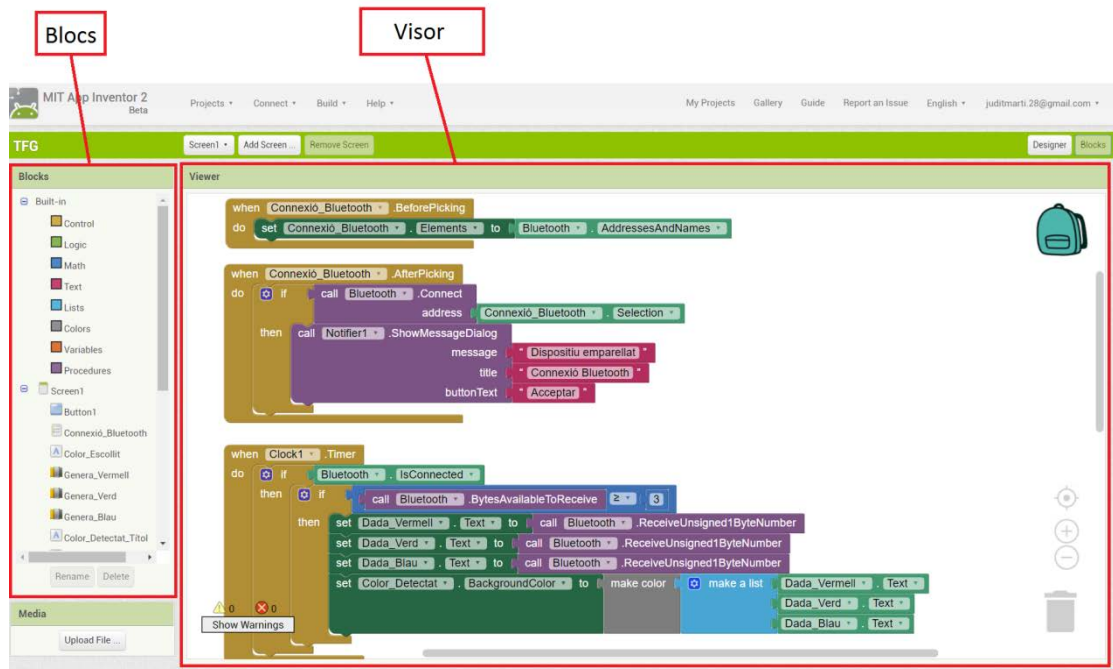


Figura 6.3. Pantalla per a donar funció a l'aplicació

6.2. Aplicació dissenyada per aquest projecte

En el cas d'aquest projecte no s'ha vist necessària la utilització de més d'una pantalla, ja que es busca que l'aplicació i el microcontrolador es comuniquin de manera automàtica una vegada emparellat el mòdul Bluetooth amb el dispositiu Android, a més de voler una aplicació senzilla i fàcil d'utilitzar.

Així, en la pantalla, una vegada oberta l'aplicació, es veuen un botó, un títol de secció, un espai en blanc, els *sliders*, un altra títol per dividir seccions i tres textos per als resultats.

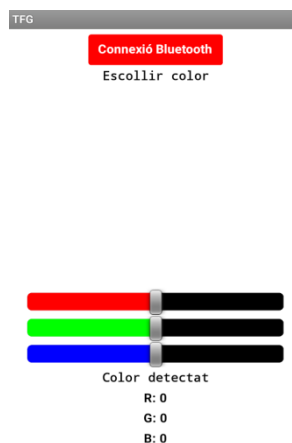


Figura 6.4. Pantalla quan s'inicia l'aplicació

Primer de tot, el botó “Connexió Bluetooth” fa que aparegui una llista dels dispositius Bluetooth que el telèfon té guardats. A l’escollir el desitjat, en aquest cas el mòdul Bluetooth, surt un quadre amb un missatge a la pantalla anunciant que s’ha emparellat correctament. Es pot veure com s’ha programat amb els blocs a la següent figura.

```

when Connexió_Bluetooth .BeforePicking
do set Connexió_Bluetooth . Elements to Bluetooth . AddressesAndNames

when Connexió_Bluetooth .AfterPicking
do if
  call Bluetooth .Connect
      address Connexió_Bluetooth . Selection
  then call Notifier1 .ShowMessageDialog
      message " Dispositiu emparellat "
      title " Connexió Bluetooth "
      buttonText " Acceptar "
    
```

Figura 6.5. Blocs per escollir connexió Bluetooth i quadre amb missatge

La segona part de l’aplicació és un quadre de text molt ampli que s’omple del color que s’escull als sliders. Per aconseguir això s’han utilitzats els blocs de la figura a continuació.

```

when Genera_Vermell .PositionChanged
thumbPosition
do set Color_Escollit . BackgroundColor to make color make a list
  Genera_Vermell . ThumbPosition
  Genera_Verd . ThumbPosition
  Genera_Blau . ThumbPosition

when Genera_Verd .PositionChanged
thumbPosition
do set Color_Escollit . BackgroundColor to make color make a list
  Genera_Vermell . ThumbPosition
  Genera_Verd . ThumbPosition
  Genera_Blau . ThumbPosition

when Genera_Blau .PositionChanged
thumbPosition
do set Color_Escollit . BackgroundColor to make color make a list
  Genera_Vermell . ThumbPosition
  Genera_Verd . ThumbPosition
  Genera_Blau . ThumbPosition
    
```

Figura 6.6. Generació del color amb tres sliders

Finalment es té un quadre de text que dona començament a la tercera part de l’aplicació amb el missatge “Color detectat”. Com bé diu el seu nom en aquest apartat surten les dades que es reben del sensor. És a dir, el sensor detecta el color que se li ha mostrat, tant si és un objecte qualsevol com la pròpia pantalla del telèfon amb la opció d’escollir color, el programa del microcontrolador agafa aquestes dades que ha detectat i les envia mitjançant

l'UART al mòdul Bluetooth i aquest al dispositiu Android. Aquestes dades s'agafen amb un bloc i s'escriuen als quadres de text que hi ha a baix de tot de la pantalla. La figura a continuació mostra com s'ha aconseguit amb els blocs.

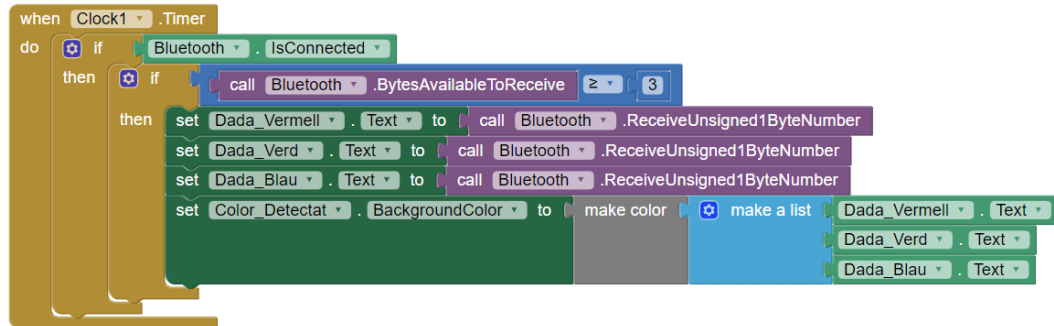


Figura 6.7. Rebuda de dades i generació del color

Com es pot veure, s'ha utilitzat el rellotge i només s'envien les dades si es compleixen certes condicions. Cada vegada que passa un segon l'aplicació mira si el Bluetooth està connectat, ja que si no hi està surt un requadre molt molest avisant que el Bluetooth està desconnectat. En el cas de que s'hagi emparellat el mòdul Bluetooth i el mòbil es mira si els bytes que estan disponibles per rebre són tres o més (les tres dades de les components del color). Si també es compleix es posen els textos de baix de tot als valors rebuts i, a més, el requadre que agafa tots aquests quadres de text genera un color amb les dades rebudes, per així poder comparar el que s'havia mostrat i el que el sensor acaba detectant.

7. Implementació sistema

7.1. Connexions dels components

Per començar, es vol explicar com estan connectats tots els components entre sí de manera individual i, finalment, tot el circuit complet. Tot això es farà mitjançant taules per explicar la funció de cada connexió.

7.1.1. PICKit2 i PIC16F690

Primer, s'explicaran les connexions entre el PICKit2 i el microcontrolador, ja que és per on s'alimentarà tot el sistema durant l'etapa de desenvolupament. En la següent taula es poden veure els pins, les connexions, i les funcions de cada una. En el cas del PICKit2 es tenen sis pins, però només s'utilitzen cinc en aquest treball.

Pin PICKit2	Pin microcontrolador	Funció
1	4	V_{pp} / MCLR
2	1	Alimentació (V_{DD})
3	20	Terra (GND)
4	19	ICSPDAT / PGD
5	18	ICSPCLK / PGC

Taula 7.1. Taula de connexions PICKit2 - PIC16F690

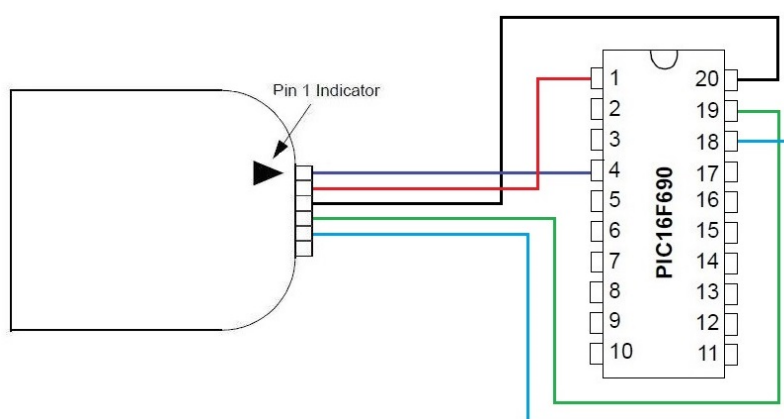


Figura 7.1. Connexions PICKit2 - PIC16F690

7.1.2. Sensor de color APDS9960 i PIC16F690

En aquest apartat s'explicaran les connexions entre l'element principal del circuit, és a dir, el

sensor, amb el microcontrolador. Es compliquen un poc les connexions perquè no són directes ja que el microcontrolador està alimentat a 5 V i el sensor de color només a 3,3 V. Per aconseguir això, les connexions que normalment anirien directes del microcontrolador al sensor han de passar per un convertidor de nivell lògic i aquest està alimentat per un regulador que, a més, s'alimenta als mateixos 5 V del microcontrolador. A continuació s'indiquen les connexions en una taula i una figura per tenir-ho clar.

Pin APDS9960	Pin microcontrolador	Funció
VL	-	Alimentació del LED infraroig a 3,3V
GND	20	Terra
VCC	-	Alimentació a 3,3V del sensor
SDA	13 (SDA)	Transmissió de dades
SCL	11 (SCL)	Relloctge
INT	17 (INT)	Interrupcions

Taula 7.2. Taula de connexions APDS9960 - PIC16F690

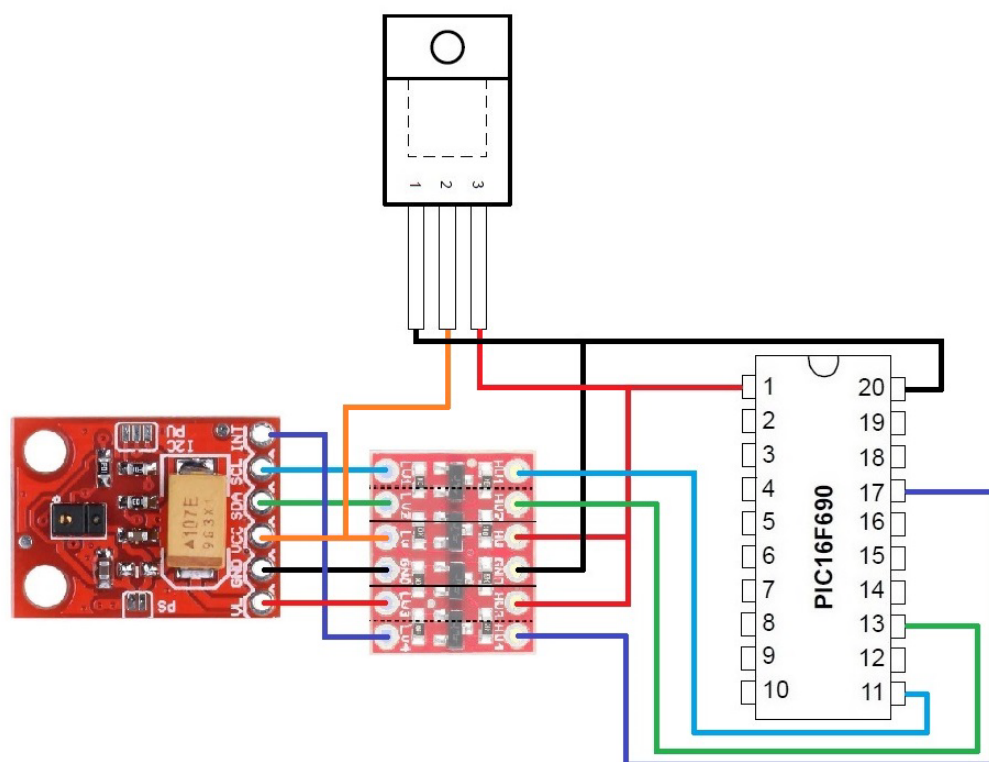


Figura 7.2. Connexions APDS9960 - PIC16F690

7.1.3. Pantalla LCD i PIC16F690

En aquest cas, a més de la pròpia pantalla LCD i el microcontrolador, s'han d'utilitzar dues

resistències, una fixa i una altra variable. La resistència fixa té la funció de limitar el corrent pel LED de la pantalla per evitar-li danys per escalfament i la variable permet regular el contrast de la LCD per a que es pugui veure correctament. En el cas dels pins de transmissió de dades, és a dir els Dx (sent x un nombre entre 0 i 7, ambdós inclosos), només s'utilitzen quatre, en aquest cas D4, D5, D6 i D7. La resta de pins estan explicats a l'apartat corresponent. A la taula i figura següent es tenen les connexions explicades i dibuixades, respectivament.

Pin LCD	Pin microcontrolador	Funció
VSS	20	Terra
VDD	1	Alimentació
V0	-	Regulació del contrast de la pantalla
RS	6	Register Select (LCD transmet o rep dades)
RW	5	Read / Write (llegeix o escriu dades)
E	8	Enable (activar o no funcions de la LCD)
D0, D1, D2, D3	-	Transmissió de dades (no s'utilitzen)
D4	16	Transmissió de dades
D5	15	Transmissió de dades
D6	14	Transmissió de dades
D7	7	Transmissió de dades
A	1	Ànode del LED de la pantalla
K	20 (amb resistència)	Càtode del LED de la pantalla

Taula 7.3. Taula de connexions LCD - PIC16F690

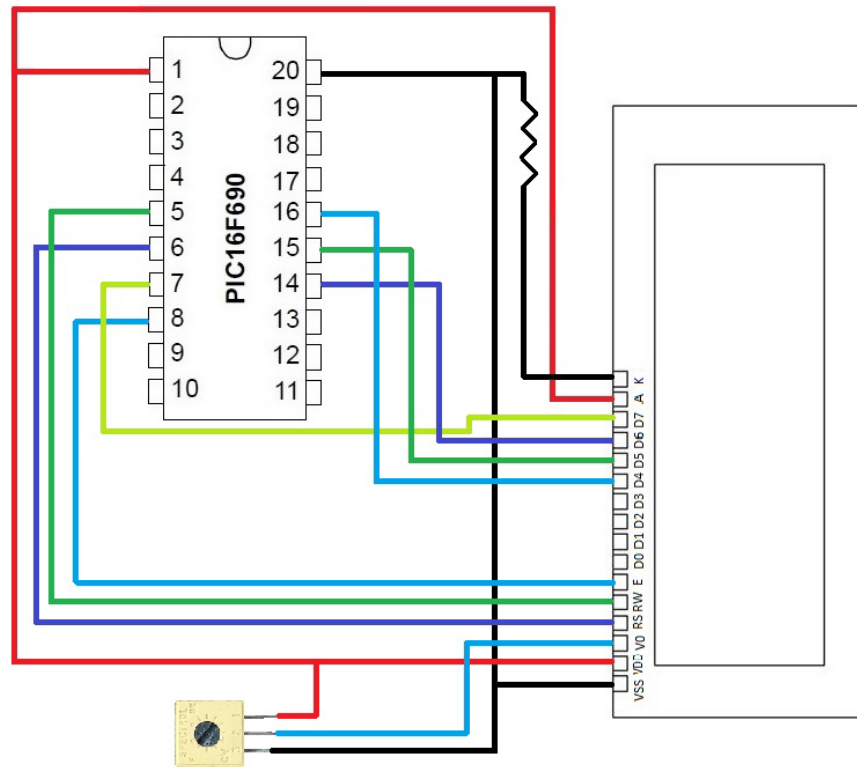


Figura 7.3. Connexions LCD - PIC16F690

7.1.4. Mòdul Bluetooth i PIC16F690

Aquí es detallen les connexions entre el mòdul Bluetooth i el microcontrolador. En aquest cas no s'utilitzen ni el pin STATE ni el pin EN com ja s'ha justificat anteriorment.

Pin Bluetooth	Pin microcontrolador	Funció
RXD	10	Recepció de dades
TXD	12	Transmissió de dades
GND	20	Terra
VCC	1	Alimentació

Taula 7.4. Taula de connexions mòdul Bluetooth - PIC16F690

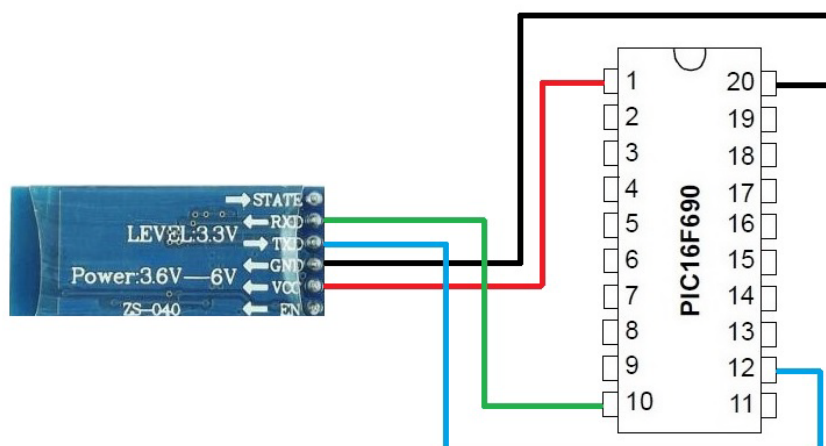


Figura 7.4. Connexions mòdul Bluetooth - PIC16F690

7.1.5. Alimentació amb pila del microcontrolador

Finalment, una vegada es té tot el projecte acabat, ja no es necessita la connexió amb l'ordinador del PICkit2, així que es substitueix l'alimentació per una pila de 9 V. No s'ha vist necessari fer una taula amb les funcions al ser només alimentacions i terra.

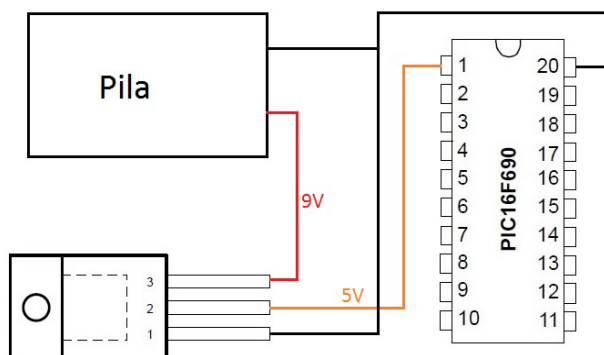


Figura 7.5. Connexions pila – PIC16F690

7.2. Circuit final

Finalment, la següent figura mostra la fotografia de les connexions reals anteriorment explicades en el circuit. S'han fet totes les connexions de manera ordenada i amb un codi de colors per a que sigui més fàcil d'entendre i visualitzar. Els cables vermells són tots alimentacions, mentre que els negres són terres. Així els cables blau i verd, a falta de més colors, s'han hagut d'utilitzar de manera indiferent per a la resta de connexions. S'ha intentat posar de forma que es pugui saber de manera ràpida i fàcil quin cable va a cada lloc.

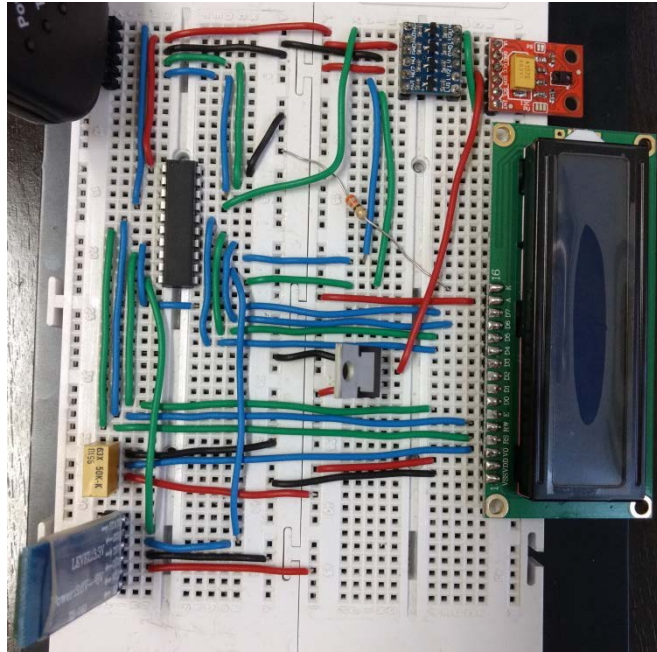


Figura 7.6. Connexions del circuit

7.3. Programació

En aquest apartat s'explicarà el codi que s'ha programat per al microcontrolador. Aquest codi bàsicament agafa les dades del sensor i les mostra per la LCD i les envia pel mòdul Bluetooth al dispositiu mòbil. S'ha utilitzat llenguatge C al ser el més utilitzat en la programació d'aplicacions basades en microcontroladors PIC. A l'Annex es troben els arxius utilitzats per al funcionament del programa principal.

Per començar, s'han d'incloure tots els arxius que es necessiten per al bon funcionament del programa. Això es fa amb el codi que hi ha a continuació.

```
// S'inclouen els arxius necessaris per al funcionament del programa
#include <pic16f690.h>
#include <xc.h>
#define _XTAL_FREQ 4000000
#include <stdio.h>
#include <stdbool.h>
#include "lcd_HD44780_PIC16.h"
#include "uart.h"
#include "APDS9960.h"
```

Figura 7.7. Arxius que s'inclouen al programa principal

La configuració del microcontrolador també té un codi concret per habilitar certes funcions. A la següent figura s'hi pot veure. En aquest projecte deshabilitem totes les opcions disponibles excepte la de l'oscil·lador intern.

```
// Configuració
#pragma config FOSC = INTRCIO
#pragma config WDTE = OFF
#pragma config PWRTE = OFF
#pragma config MCLRE = OFF
#pragma config CP = OFF
#pragma config CPD = OFF
#pragma config BOREN = OFF
#pragma config IESO = OFF
#pragma config FCMEN = OFF
```

Figura 7.8. Configuració del microcontrolador

En el llenguatge C es poden referir pins concrets amb noms per a poder referir-se a ells i no haver de recordar cada pin el que era, sinó que amb el nom assignat ja es sap de forma més intuïtiva. A continuació es veuen els pins en concret de la connexió I²C, és a dir, la del microcontrolador amb el sensor.

```
// Definició dels pins per a l'I2C
#define SDA_PIN RB4 //
#define SCL_PIN RB6 //
#define SDA_DIR TRISB4
#define SCL_DIR TRISB6
```

Figura 7.9. Configuració dels pins I²C

Com a part final de la inicialització general del programa, s'inicialitzen les variables en RAM i en aquest cas són globals (es poden utilitzar a tot el programa, no només a funcions concretes).

```
unsigned int id;
int t;
int LECTURA;
```

Figura 7.10. Inicialització de les variables

A continuació comença el programa principal del programa i es passa a inicialitzar a 0 els ports del microcontrolador. També es defineixen per començar tots els ports com a sortides, excepte el bit 2 del port A, que és el de les interrupcions, que es defineix com a entrada, ja que seran les interrupcions del sensor les que entraran en aquest pin. S'ha definit aquí directament perquè és només un pin i queda clar des del principi. També es defineixen tots els pins com a digitals.

Una vegada inicialitzat tot el comentat fins ara, es configuren els bits 5 i 7 del port B per separat del que ja es tenia perquè són els ports que van a la transmissió i recepció del mòdul Bluetooth. Així, a l'estar definits per separat de la inicialització del principi, es pot veure clarament què és cada cosa. En el cas del bit 5 del port B es defineix com a entrada i el 7 com a sortida, al ser la recepció i transmissió, respectivament. També es defineix la UART amb la freqüència que es vol (determinada per les característiques del mòdul

Bluetooth) i s'habiliten les interrupcions. A la següent figura es té el codi de la inicialització dels ports i de la UART.

```
// Programa principal
int main(void)
{
    // Es posen tots els ports a 0 per inicialitzar
    PORTA = 0;
    PORTB = 0;
    PORTC = 0;

    TRISA = 0b00000100; // Bit 2 del Port A es defineix com a entrada (interrupcions)
    TRISB = 0b00000000; // Tots els ports B com a sortida
    TRISC = 0b00000000; // Tots els ports C com a sortida
    ANSEL = 0X00; // digital
    ANSELH = 0X00; // digital

    // Inicialització de la UART per a la comunicació Bluetooth
    TRISBbits.TRISB5=1; // Bit 5 del port B activat d'entrada
    TRISBbits.TRISB7=0; // Bit 7 del port B activat de sortida
    UART_Init(9600); // Activar UART amb la freqüència que volem
    PIE1bits.RCIE=1; //habilitar interrupcions
```

Figura 7.11. Inicialització dels ports i la UART

A continuació es passa a la configuració de les interrupcions. El flag es posa a 0 per inicialitzar. El flag és un bit que indica si hi ha una interrupció activa (nivell alt) o no (nivell baix). Hi ha una altra variable que indica l'habilitació de les interrupcions, és l'enable. Si aquesta darrera està a 1, significa que les interrupcions estan habilitades. En canvi el flag es refereix a si s'ha produït una interrupció o no. S'habiliten les interrupcions dels perifèrics, les globals, les externes i es diu que la interrupció es vol en el flanc de baixada.

```
// Inicialització del servei d'interrupcions
INTF = 0; // Reset del flag de les interrupcions externes
INTE = 1; // Habilitació de les interrupcions externes
PEIE = 1; // Habilitació de les interrupcions dels perifèrics
GIE = 1; // Habilitació d'interrupcions global
OPTION_REG = 0; // Habilitació de l'interrupció en el flanc de baixada
```

Figura 7.12. Habilitació de les interrupcions

La pantalla LCD té una llibreria amb diverses funcions per a realitzar diferents operacions. Un exemple pot ser el codi següent en que s'inicialitza la pantalla en el mode de 4 bits i s'espera una estona a que s'inicialitzi correctament. La funció LCDCmd() és l'encarregada d'enviar comandes i realitza diferents funcions segons el nombre que se li dona. En el cas de la següent figura es veu que un fa que s'encengui i l'altre esborra la pantalla. Finalment, una comanda que s'utilitzarà molt en aquest treball és posicionar el que es vol escriure a un lloc concret de la pantalla amb LCDGotoXY(x,y), escriure un string amb LCDWriteString() i també LCDWriteInt() per quan es tracta d'escriure enters.

```

LCDInit_signals(); // Inicialització de la LCD en mode 4 bits
__delay_ms(100);

LCDCmd(0b00001100); // Pantalla ON
LCDCmd(0b00000001); // Es treu tot el que podria haver a la pantalla

LCDGotoXY(4,0);
__delay_ms(100);
LCDWriteString("Trebball"); // Escriu un string
LCDGotoXY(3,1);
LCDWriteString("Fi de Grau");

```

Figura 7.13. Exemple de Funcions de la LCD

```

LCDWriteString("EN: "); // Escriure un string
LECTURA = wireReadDataByte(APDS9960_ENABLE); // Es llegeix la mesura
LCDWriteInt(LECTURA,3); // S'escriu la mesura

```

Figura 7.14. Funció per escriure caràcteris i dades numèriques a la LCD

Al capítol del sensor ja es van explicar les funcions importants d'aquest així que no es tornarà a repetir aquí. Per acabar, s'ha de parlar del que es necessita per a fer funcionar la UART. El que s'ha utilitzat en aquest projecte ha estat el que es mostra a la figura següent, que bàsicament envia dades per la UART.

```

// Enviar dades per la UART
UART_Write( (R_data_L + (R_data_H <<8)) );
UART_Write( (G_data_L + (G_data_H <<8)) );
UART_Write( (B_data_L + (B_data_H <<8)) );

```

Figura 7.15. Funció per enviar dades per la UART

8. Funcionament i resultats finals

En aquest capítol s'explica què passa en cada moment desde que s'inicia el circuit fins que es transmeten les dades. Per començar, s'ha fet un diagrama de flux per veure tot el funcionament d'una manera esquemàtica i ràpida.

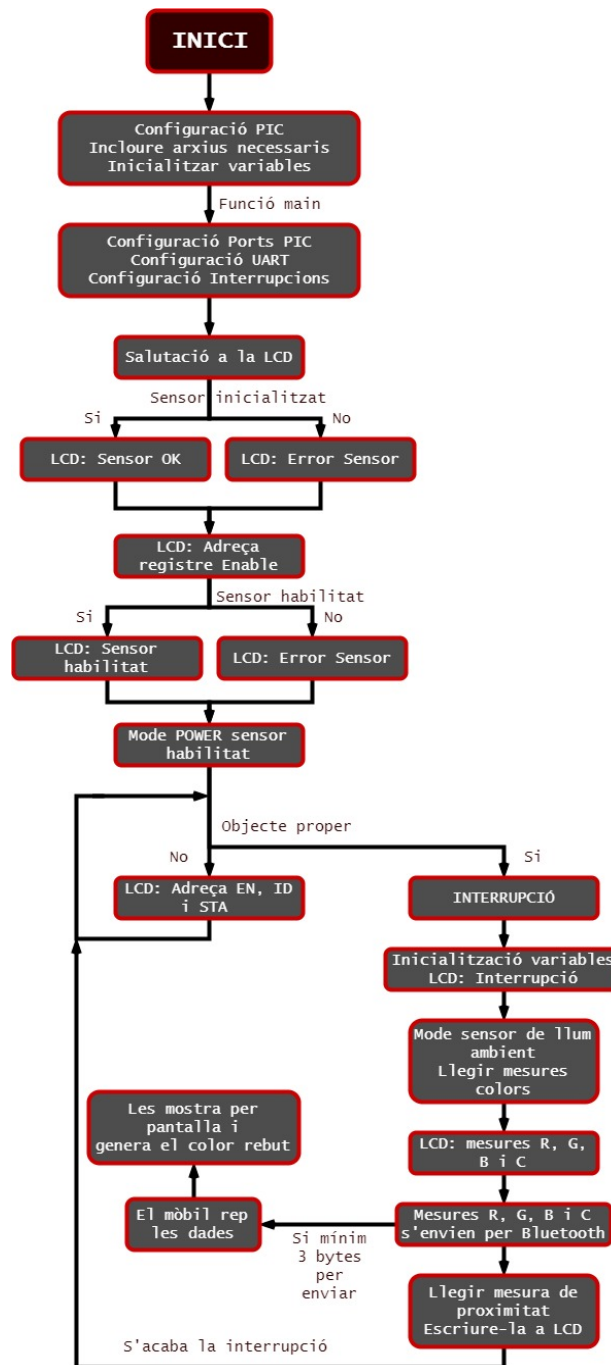


Figura 8.1. Diagrama de flux del funcionament del sistema

Com es pot veure, tot comença amb una configuració dels arxius necessaris per al bon funcionament del programa, també es configura el microcontrolador i s'inicialitzen les variables en ROM (són globals per a tot el programa i no només en funcions concretes). Una vegada dins de la funció principal es configuren els ports, la UART i les interrupcions del microcontrolador i es fa escriure una presentació a la pantalla LCD.

A continuació es mira si el sensor s'ha inicialitzat correctament i es mostra per la LCD si tot està bé o no. Fa el mateix amb l'adreça del registre enable per veure si s'ha habilitat el sensor o no i també mostra la conclusió per pantalla. Després s'activa el mode d'engegat del sensor i així s'arriba al bucle del programa, que pot ser interromput en qualsevol moment per les interrupcions ja activades en el cas que s'acosti un objecte al sensor. Així, si no s'acosta res, el bucle mostra constantment per la LCD les adreces de diversos registres del sensor, per així saber en cada moment què és el que es té connectat i si hi pot haver alguna cosa malament.

En el cas que es produeixi una interrupció, es surt del bucle i s'inicialitzen unes variables que només s'utilitzen en el codi de la interrupció. A la LCD s'avisa que hi ha hagut una aturada del codi normal i ho mostra uns segons. Així s'activa el mode de detecció de colors del sensor i es llegeixen les mesures per després mostrar-les per la LCD. A continuació s'envien per la UART però aquestes només són rebudes pel mòbil si mínim s'estan enviant tres bytes. Aquesta és una condició que no es veu en el codi sinó en els blocs de l'aplicació per a Android.

A continuació, una vegada enviades les dades per Bluetooth, es llegeix la mesura de proximitat, es borren les dades del color de la LCD i es mostra com de pròxim estava l'objecte del sensor. Finalment acaba la interrupció i es torna al bucle del codi normal del programa.

8.1. Resultats mostrats a l'aplicació Android

Els resultats del que detecta el sensor depenen molt de la quantitat de llum ambient que hi ha en el moment. Així, si no hi ha gaire llum, detectarà un color semblant però més fosc del que és l'objecte. Si, en canvi, hi ha massa llum, no dona un bon resultat. Per explicar-ho millor es posarà un exemple, si se li posa un color viu desde la pantalla del mòbil massa aprop del sensor, es detectarà un component del color a més de 255 (ja que les dades llegides del sensor són dos bytes i el valor màxim d'un byte és 255, el que s'envia al mòbil és només un byte) i s'enviarà un resultat erroni al mòbil, ja que per generar els colors no entén si s'envia un 300.

Una vegada s'han vist aquestes limitacions després d'anar experimentant amb objectes, lluminositats i el propi dispositiu Android, s'ha pogut fer una certa calibració i s'han pogut

obtenir uns resultats bastant acurats. Per exemple, a continuació es mostren els tres colors bàsics enviats des de l'aplicació Android al sensor (part superior de la figura) i aquest els detecta (part inferior). El color generat per l'aplicació és bastant similar.



Figura 8.2. Colors bàsics mostrats i detectats a l'aplicació Android

També s'han fet altres proves amb mesclades de colors, algunes han sortit millor que altres. Per exemple a la següent figura es poden veure dos colors bastant acurats i en canvi el groc no l'agafa bé, detecta molt de blau encara que no s'hagi generat realment, es pot veure un blau de 147 quan realment s'havia enviat 0.

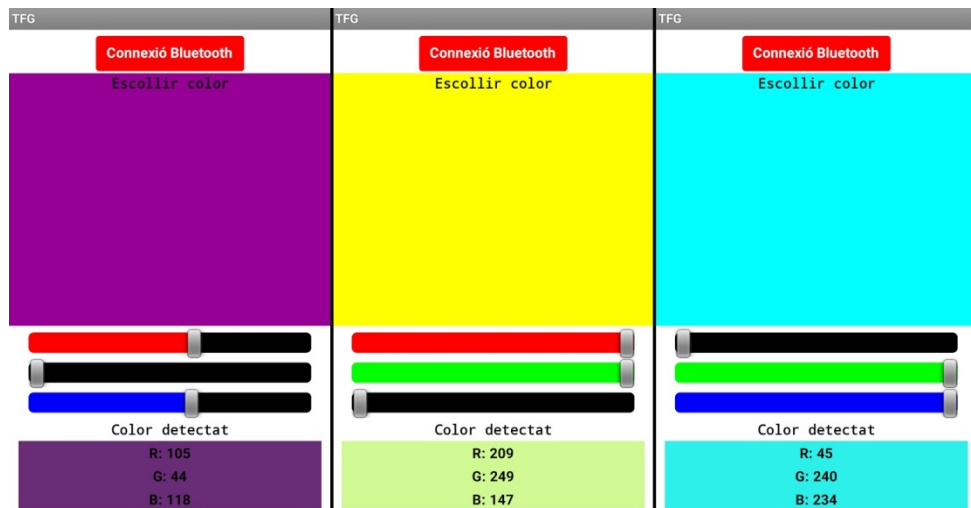


Figura 8.3. Colors bàsics mostrats i detectats a l'aplicació Android

9. Pressupost

En aquest apartat es realitza un pressupost econòmic d'aquest projecte.

COSTOS DE MATERIALS			
Concepte	Quantitat	Preu unitari (€)	Preu total (€)
Protoboard	1	18,55	18,55
Microcontrolador PIC16F690	1	2,70	2,70
Programador PICkit2	1	24,99	24,99
MPLAB IDE	-	0,00	0,00
Sensor APDS9960	1	7,85	7,85
Pantalla LCD	1	3,05	3,05
Mòdul Bluetooth HC-05	1	4,90	4,90
Convertidor de nivell de voltatge	1	3,93	3,93
Regulador de voltatge	2	1,00	2,00
Cables	2m	0,05	0,10
Resistència fixa	1	1,19	1,19
Resistència variable	1	3,39	3,39
Pila 9V	1	1,30	1,30
Connector pila 9V	1	0,42	0,42
MIT App Inventor 2	-	0,00	0,00
Cost total de materials			74,37
COSTOS DE PERSONAL			
Concepte	Preu/hora (€/h)	Hores (h)	Preu total (€)
Treball estudiant	15	300	4500,00
Tutories	50	10	500,00
Cost total de personal			5000,00
COST TOTAL DEL PROJECTE			5074,37

Conclusions

Aquest projecte tenia l'objectiu principal de dissenyar i construir un circuit electrònic capaç d'identificar objectes de color. Per realitzar aquest treball s'han hagut d'estudiar i entendre tots els elements que s'han utilitzat, per així entendre a connectar-los entre sí, programar-los i finalment entendre els resultats. S'ha après durant aquests mesos que al treballar en un projecte més complex s'ha d'anar fent les coses a poc a poc, ja que així tot s'entén millor.

Amb aquest projecte s'han pogut aprofundir en aspectes de l'electrònica que no s'havien pogut fer tant durant les assignatures passades d'electrònica, ja que eren sobretot teòriques. S'han pogut aplicar conceptes i entendre d'altres millor a base de posar-los en pràctica. S'ha pogut profunditzar en aprendre el funcionament dels components utilitzats. Per exemple, mostrar el que es vulgui per una pantalla LCD o crear una aplicació per a Android que rebí dades per Bluetooth. Al principi semblava que era molt complicat de fer però amb dedicació i a base d'entendre com funciona tot s'ha vist que es pot fer.

Els resultats obtinguts han estat satisfactoris ja que el circuit funciona com estava previst i el sensor detecta els colors bastant propers al que se li mostra. Té algunes limitacions en funció de la llum ambient, per exemple, però tot ha sortit com s'havia especificat inicialment.

Durant la realització d'aquest projecte hi ha hagut alguns problemes i contratemps que s'han hagut d'anar solucionant a mida que es produïen però s'ha aconseguit el que es pretenia.

Agraïments

S'agraeix a la professora i directora d'aquest treball Rosa Rodríguez, del departament d'electrònica de l'ETSEIB, per tota l'ajuda en la realització d'aquest treball. En tot moment ha estat disposta a ajudar i solucionar dubtes que hagin anat sorgint durant aquest projecte.

També s'agraeix al departament d'electrònica de l'ETSEIB per deixar el laboratori i el material necessari per a la realització d'aquest projecte.

Finalment, es vol agrair a la família, que sempre són un suport per seguir endavant.

Bibliografia

- [1] Microchip. *PIC16F690 Datasheet*, 2015.
<http://ww1.microchip.com/downloads/en/DeviceDoc/41262E.pdf>
- [2] Microchip. *PICkit 2 User's Guide*, 2008.
<http://ww1.microchip.com/downloads/en/DeviceDoc/51553E.pdf>
- [3] Microchip Technology Inc. *MPLAB XC8 C COMPILER User's Guide*, 2012.
- [4] HITACHI. *HD44780U LCD Datasheet*, 2016.
<https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>
- [5] ITead Studio. *HC-05 Datasheet*, 2010.
http://www.robotshop.com/media/files/pdf/rb-ite-12-bluetooth_hc05.pdf
- [6] Texas Instruments. *LM117 Datasheet*, 2013.
<http://www.ti.com/lit/ds/snvs774n/snvs774n.pdf>
- [7] Avago Technologies. *APDS-9960 Datasheet*, 2013.
https://cdn.sparkfun.com/assets/learn_tutorials/3/2/1/Avago-APDS-9960-datasheet.pdf
- [8] Ai2.appinventor.mit.edu. *MIT App Inventor 2*, 2017.
<http://ai2.appinventor.mit.edu/>