POLITECNICO DI TORINO

I Facoltà di Ingegneria

Corso di Laurea in Ingegneria Biomedica

Spec. Strumentazione Biomedica

Tesi di Laurea Magistrale



Sistema Integrato e Portabile per il Monitoraggio di Potenziali Evocati Visivi: Dispositivo di stimolazione ottica.

Relatore:	
Prof. Ing. Danilo Demarchi	
Correlatori:	
Prof. Ing. Alessandro Sanginario	
Dott. Paolo Astengo	Candidato:
	Vizio Matteo

Indice

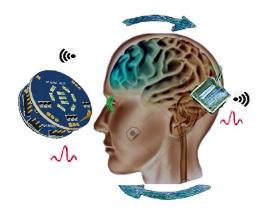
Obiettivo

1	Introdu	zione								6
2	Elemen	ti di fisiologia								8
	2.1	Elettrofisiologi	a oculare							. 8
			ziali evocati vi	sivi (VE	P o PE	SV)				. 8
		2.1.1.	1 VEP con sti	molator	e a LE	D				. 10
		2.1.2 Elettro	oretinogramm	a						. 10
	2.2	L'anatomia del	l'occhio .							. 11
		2.2.1 La nat	ura ed il comp	ortame	nto de	lle ond	e lumin	ose		. 12
		2.2.2 La foto	otrasduzione e	consid	erazio	ni anat	omiche	aggiun	tive	
		relativ	ve al nostro st	udio						. 14
		2.2.3 Le vie	visive .							. 17
3	Stato d	ell'arte								18
4	Analisi	delle specifich	e e delle scelt	e sul di	isnosi	tivo				22
•		Cenni sulla teci			зрозг					. 22
		Analisi disposit	· ·		lella b					. 24
		Descrizione e c	_						its	
	1.0	LAUNCHXL-CC								. 25
		4.3.1 BLE								. 29
		4.3.2 SPP								. 31
		4.3.3 GPIO,	GPTIMER, IOC	, PRCM						. 33
	4.4	Idee di sincron	izzazione		•		•		•	. 34
5	Hardwa	are								38
	5.1	Progettazione d	lel dispositivo							. 40
		5.1.1 Scelta	dei LED e test	fotome	trici					. 40
		5.1.2 Batter	ia a bottone							. 45
		5.1.3 Schem	ia, calcoli e cor	nsideraz	zioni					. 48
			1 Calcoli per t			batter	ia SONY	CC203	32 sotto	posta a
			carico							. 52
	5.2	Circuito compl	eto .							. 58
6	Firmwa	ıro								61
J		Sensor Control	ler Studio (SCS	5)						. 62
		Code Compose	•	•	•	•	•	•	•	. 67
	0.2	=	r "pwmled.c"		•	•	•	•	•	. 69
		O.Z.I Director	Pillicaic	•	-	•	•	-	-	. 57

	6.2.2 Funzione "sync_timer.c"							78
	6.2.3	"SPP_BLE_SE	RVER"					82
	6.2.4	"Serial port s	ervice'					84
7	PCB							86
	7.1 Altium I	Designer						86
	7.1.1	Schematico						86
	7.1.2	PCB .	•		•			91
8	Prove e test sui	dispositivi						97
9	Conclusioni							109
	9.1 Sviluppi	futuri .						110
1() Appendice							111

Bibliografia

Obiettivo



L'obiettivo del progetto SIPEV (Sistema integrato e portatile per il monitoraggio di potenziali evocati visivi) è lo sviluppo di un sistema completo di neuromonitoraggio intraoperatorio specifico per i potenziali evocati visivi (Visual Evoked Potentials - VEP). Questo sistema è composto da 3 dispositivi che comunicano e sono connessi tra di loro.

E' stato realizzato in collaborazione con altri 2 studenti del Politecnico di Torino ed ognuno di noi si è occupato di un dispositivo. Rispettivamente, in questo report verrà discusso e portato a realizzazione il dispositivo di stimolazione ottica per la generazione di potenziali evocati visivi; il dispositivo di lettura verrà realizzato da Miriam Obertino nella tesi :"Sistema integrato e portatile per il monitoraggio di potenziali evocati visivi: dispositivo di acquisizione" [1], mentre il dispositivo di unità centrale che sincronizza stimolo e lettura verrà realizzato da Davide Romano nella tesi: "Sistema integrato e portatile per il monitoraggio di potenziali evocati visivi unità centrale per la sincronizzazione di stimolo e lettura."[2]

Il progetto SIPEV si impegnerà a realizzare i dispositivi omettendo i cavi che nel GEPEV creavano ingombro durante l'operazione chirurgica al neurochirurgo con un conseguente incremento dello spazio di lavoro, riduzione dei costi di realizzazione del dispositivo e dei costi ospedalieri con una conseguente riduzione delle dimensioni dei dispositivi, rispetto a quelli attualmente in commercio.

Capitolo 1

Introduzione

L'obiettivo del progetto fissa già dei punti chiave e definisce in modo approssimativo il dispositivo che si andrà a realizzare per il progetto SIPEV. Il dispositivo e le considerazioni sul dispositivo stesso prenderanno vita di qui in avanti.

I potenziali evocati (PE) sono esami che valutano la risposta generata dal sistema nervoso (centrale o periferico) in seguito ad una stimolazione sensitiva o sensoriale.

Nell'uomo i segnali elettrofisiologici in risposta agli stimoli sensoriali possono essere agevolmente registrati su zone diverse dell'encefalo a seconda delle sedi in cui sono generati ed in modo non invasivo, utilizzando tecniche impiegate per la prima volta nel 1947.

In relazione all'organo sensoriale stimolato si possono ottenere:

- potenziali evocati visivi (VEPs);
- potenziali evocati somato sensoriali (SEPs);
- potenziali evocati uditivi tronco-encefalici (BAEPs).

Il monitoraggio real-time dei VEP permette al neurofisiologo di monitorare lo stato del nervo ottico durante gli interventi chirurgici. Nel caso in cui il nervo sia sottoposto a sollecitazioni critiche, il neurofisiologo avvisa immediatamente il neurochirurgo che interviene per evitare problemi al paziente.

Il progetto prende spunto da un precedente studio di fattibilità denominato GEPEV. In esso sono state riscontrate delle forti criticità sugli attuali dispositivi di stimolazione ottica per la generazione dei VEP ed il loro monitoraggio.

SIPEV risolverà queste problematiche sviluppandoci attorno un'elettronica mirata partendo anche dai risultati ottenuti da GEPEV; realizzando:

- 1. per la parte di stimolazione ottica il dispositivo con scelta della geometria e la tipologia di LED opportuna, verrà integrata un'elettronica per il controllo dello stimolo per la ricezione e trasmissione wireless ed una batteria per rendere il dispositivo usa e getta;
- 2. nella parte di lettura verranno integrati con elettronica integrata degli elettrodi e l'elettronica permetterà il pretrattamento del segnale con ricezione e trasmissione wireless;
- 3. ed infine un'unità centrale che si occuperà di gestire la trasmissione dei dati e la ricezione dei dati che arriveranno dai precedenti due dispositivi.

In uno step finale, i dati alla centrale saranno presi in esame da un neurofisiologo.

Capitolo 2

Elementi di fisiologia

2.1 Elettrofisiologia Oculare

L'elettrofisiologia oculare è una disciplina specialistica con lo scopo di esplorare l'attività elettrica associata all'apparato visivo attraverso specifici test diagnostici. E' possibile fare valutazioni, con appropriate metodiche di stimolazione e di registrazione, testando le funzionalità dei differenti elementi retinici e delle vie ottiche. I test più diffusi sono l'elettroretinogramma (ERG) e i potenziali evocati visivi (VEP) grazie ai quali è possibile valutare gravi alterazioni della percezione visiva che possono essere provocate da modificazioni patologiche della trasmissione dell'informazione visiva dai fotorecettori retinici fino alla corteccia cerebrale.

2.1.1 Potenziali Evocati Visivi (VEP o PEV)

I VEP descrivono dei fenomeni di trasduzione e di trasmissione dell'impulso nervoso lungo le vie visive, dai fotorecettori retinici fino alla corteccia cerebrale occipitale. Da questo si individueranno le variazioni dei potenziali bio-elettrici della corteccia occipitale creati da stimoli visivi. Essi si distinguono in VEP da stimolo strutturato (a variazione di contrasto) e **VEP** da stimolo non strutturato ſa variazione di luminanza). I potenziali evocati misurano il tempo di trasmissione dell'attività neuronale dalla retina alla corteccia occipitale e permettono di misurare la funzionalità e l'integrità del percorso; essi possono essere generati attraverso diversi protocolli di stimolo visivo, i più utilizzati in ambiente ambulatoriale sono il pattern e il flash. Nel caso del pattern, lo stimolo più utilizzato è una scacchiera (Figura 1) raffigurata su di un display, dove avviene l'alternanza di quadrati bianchi e neri. Durante il test, i quadrati neri diventano bianchi e viceversa.

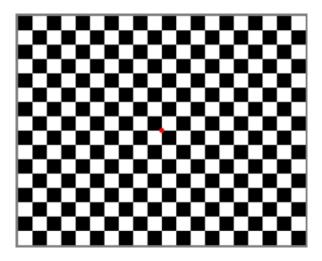


Figura 1 - Stimolo a scacchiera con punto rosso centrale da fissare. [4]

La risposta dipenderà da molti fattori:

- dimensione dei quadrati;
- frequenza del rovesciamento dei colori (pattern-reversal);
- luminosità del display;
- contrasto tra i quadrati bianchi e neri;
- concentrazione del paziente.

Un normale VEP ispezionato consiste nell'identificazione di una specifica forma d'onda (Figura 2) con tre picchi: N75, P100 e N135 dove il valore numerico assegnato, rappresenta il tempo in millisecondi di apparizione dello stimolo. La lettera "P" indica un picco positivo ed "N" un picco negativo. Il valore di ampiezza è valutato tra il picco N75 e il picco P100. Per la valutazione di un VEP viene utilizzato il P100 in quanto più stabile e ripetibile.

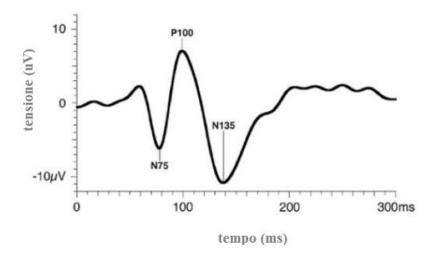


Figura 2 - Andamento forma d'onda del segnale VEP.

2.2.1.1 VEP con stimolatore a LED

L'applicazione dello stimolo a scacchiera è diventato lo stimolo standard per la valutazione dei

potenziali evocati visivi, questa tecnica però richiede soggetti attivi e cooperanti che non è possibile avere durante il monitoraggio intraoperatorio di pazienti sotto anestetico. La stimolazione a LED invece è un'alternativa alla stimolazione descritta prima ed ha il vantaggio di poter essere utilizzata anche quando la palpebra dell'occhio è chiusa. Lo svantaggio di non poter essere utilizzata per valutare l'acuità visiva e di non possedere una sensibilità da essere paragonata alla stimolazione a scacchiera risulta essere di particolare importanza nell'investigazione di malattie demielinizzanti (come ad es. sclerosi multipla). La stimolazione a LED è a tutti gli effetti simile ad uno stimolo da flash, questo tipo di stimolazione è preferibile nelle unità di terapia intensiva o durante il monitoraggio delle operazioni chirurgiche è invasiva altri in quanto meno di tipi di stimolazione.

2.1.2 Elettroretinogramma

L'elettroretinogramma (ERG) rappresenta un altro importante strumento elettrofisiologico per la diagnosi e lo studio della funzionalità delle diverse componenti retiniche. Come per i potenziali evocati visivi, anche in questo caso si distinguono due tipi di ERG a seconda dello stimolo che produce le risposte: elettroretinogramma da pattern (PERG) ed ERG da flash che è l'espressione

della somma dei vari potenziali che generano entrambi i fotorecettori (coni, bastoncelli o entrambi). Il contributo fornito dalla regione maculare o foveale, invece, può essere valutato tramite la registrazione dell'elettroretinogramma focale (fERG). [5] La risposta elettrica della retina sottoposta ad uno stimolo luminoso può essere valutata tramite tre elettrodi: un elettrodo attivo di tipo corneale o congiuntivale, un elettrodo di riferimento a livello della tempia e l'elettrodo di terra che minimizza il flusso di correnti di disturbo. La registrazione simultanea dei VEP e dell'ERG da pattern, in particolare durante il monitoraggio intraoperatorio, permette di verificare che lo stimolo luminoso abbia effettivamente raggiunto la retina anche nel momento in cui non è possibile registrare i VEP.

2.2 L'anatomia dell'occhio

L'occhio è il protagonista principale della generazione dei potenziali evocati e può essere diviso in tre strati che sono concentrici tra loro. Lo strato più esterno è costituito dalla sclera e dalla cornea. La prima è un tessuto connettivo che forma la parte bianca dell'occhio. Nella parte anteriore, la sclera dà origine alla cornea. Quest'ultima è una struttura trasparente che permette alla luce di penetrare nell'occhio. Lo strato medio dell'occhio è costituito dalla coroide, dal corpo ciliare e dall'iride. La coroide è uno strato di tessuto molto pigmentato posto al di sotto della sclera. La coroide include i fotorecettori, contiene inoltre vasi ematici che nutrono lo strato profondo dell'occhio. Il corpo ciliare contiene i muscoli ciliari, che sono connessi ad una lente, il cristallino, attraverso dei sottili tendini di tessuto connettivo chiamati fibre della zonula. Il cristallino ha il compito di mettere a fuoco la luce sulla retina, dove l'informazione visiva viene tradotta. I muscoli ciliari cambiano la forma del cristallino e permettono la focalizzazione dei raggi luminosi. L'iride è formata da due strati di cellule muscolari pigmentate, è localizzata davanti al cristallino ed è quella che determina il colore degli occhi. La pupilla invece è un foro, posizionato al centro dell'iride e permette alla luce di penetrare nella parte posteriore dell'occhio. L'iride regola il diametro della pupilla varia così facendo la quantità di luce che raggiunge la parte posteriore dell'occhio. Più internamente all'occhio vi è la retina, che è formata da tessuto nervoso contenente i fotorecettori, cellule sensibili alle onde luminose. I fotorecettori sono di due tipi, i coni e i bastoncelli, che percepiscono rispettivamente la luce intensa e quella crepuscolare. (coni e bastoncelli). La retina, pertanto, funziona come un fototrasduttore, trasformando energia luminosa in energia elettrica. Nella parte esterna della retina e attaccato alla coroide si trova l'epitelio pigmentato della retina. Questa struttura contiene la melanina, che assorbe la luce che arriva alla parte posteriore dell'occhio. Essa impedirà la riflessione attraverso la retina che causerebbe distorsione dell'immagine.

Due aree della retina sono molto importanti; una è la fovea, che è il punto centrale della retina dove dirige la luce proveniente dal centro del campo visivo. L'altra è la macula che è l'area della retina con la maggiore acuità visiva. Il disco ottico, è anche quella porzione di retina che non va tralasciata per la sua importanza. Essa è attraversata dal nervo ottico e di vasi ematici che irrorano l'occhio, regione che è sprovvista di recettori e costituisce un punto cieco della retina dove la luce non può generare impulsi elettrici e non verrà percepita. La cornea ed il cristallino sono strutture trasparenti e vengono attraversate dalla luce. Queste strutture dipendono dall'apporto ematico di nutrienti e la presenza dei vasi ostruirebbe parzialmente il passaggio alla luce.

Le strutture importanti dell'occhio sono mostrate in figura sotto riportata (Figura 3). [7]

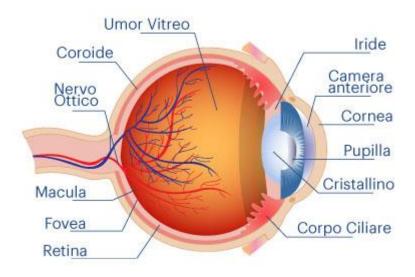


Figura 3- Principali strutture dell'occhio. [6]

2.2.1 La natura e il comportamento delle onde luminose

La luce è una forma di energia costituita da onde elettromagnetiche. In modo analogo ad altre forme di energia elettromagnetica, come le onde radio, i raggi X e i raggi gamma, la luce è parte dello spettro elettromagnetico (Figura 4). La luce visibile include onde elettromagnetiche con lunghezza d'onda compresa tra 350 nm e 750 nm dove a differenti colori corrispondono diverse lunghezze d'onda. La luce possiede tutte le caratteristiche delle onde, può essere riflessa e rifratta. La riflessione è un fenomeno dove le onde luminose urtano e rimbalzano su una superficie; la riflessione è importante nella fisiologia della visione poiché molte delle onde che percepiamo sono riflesse dagli oggetti che osserviamo. La luce che è assorbita dagli oggetti, invece, non è percepita. Possiamo percepire un oggetto di colore rosso in quanto esso riflette verso gli occhi una luce con lunghezza d'onda corrispondente al rosso (circa 700 nm), mentre vengono assorbite tutte le altre lunghezze d'onda. Mentre invece la rifrazione rappresenta il fenomeno per il quale le onde luminose cambiano la loro direzione nel passare attraverso materiali trasparenti, di densità differenti. Questa proprietà è importante nella visione in quanto, nel passaggio dagli oggetti ai fotorecettori, la luce deve passare attraverso diverse sostanze come l'aria, la cornea, il cristallino e gli umori acqueo e vitreo. [7]

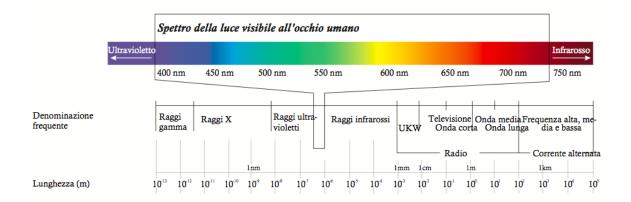


Figura 4 - Spettro elettromagnetico. [8]

Nella retina sono localizzati i fotorecettori: coni e bastoncelli. I bastoncelli sono coinvolti nella visione monocromatica in condizioni di luce poco intensa o crepuscolare. I coni ci forniscono la visione a colori, ma sono attivi soltanto quando la luce è intensa, cioè nella visione diurna. La retina è costituita da tre distinti strati (Figura 4):

- 1. uno strato interno costituito da cellule gangliari,
- 2. uno strato intermedio contenente le cellule bipolari,
- 3. ed uno strato esterno contenente i coni e i bastoncelli.

Nella retina sono presenti anche altre cellule, quelle amacrine e quelle orizzontali, che modulano la comunicazione cellulare nel tessuto nervoso retinico.

I coni e i bastoncelli, essendo posizionati nello strato esterno della retina, vengono eccitati dalla luce dopo che questa ha attraversato lo strato retinico interno e medio.

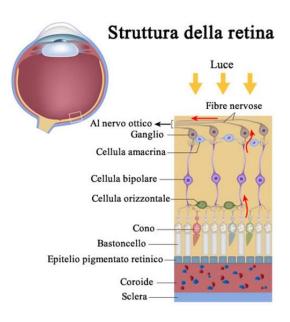


Figura 4 - Anatomia della retina. [9]

2.2.2 La fototrasduzione e considerazioni anatomiche aggiuntive relative al nostro studio

La fototrasduzione rappresenta il fenomeno mediante il quale lo stimolo luminoso viene convertito in segnale elettrico e viene captato a livello della retina dai fotorecettori: coni e bastoncelli. L'aspetto morfologico di questi 2 fotorecettori è simile, poiché entrambi sono costituiti da un segmento interno e da uno esterno (Figura 5). Quello più esterno assorbe l'energia luminosa, permettendo ai fotorecettori di eccitarsi. Quello interno contiene il nucleo cellulare e vari organuli e termina con vescicole contenenti il neurotrasmettitore. L'assorbimento della luce avviene da parte di particolari molecole dette fotopigmenti. Esse sono contenute nel segmento esterno. Nei recettori sono presenti 4 tipi di fotopigmenti. Un tipo è presente nei bastoncelli ed altri 3 tipi di fotopigmenti sono contenuti nei 3 tipi di coni. Esistono coni L che rispondono a lunghezze d'onda più lunghe, con massima sensibilità a 560 nm, detto anche "spettro del rosso". I coni M che rispondono a lunghezze d'onda medie con massima sensibilità a 530 nm, detto anche "spettro del verde". I coni S che rispondono alle lunghezze d'onda più corte, con la massima sensibilità a 420 nm o "spettro del blu". Ogni fotopigmento ha un proprio picco di assorbimento della luce che corrisponde ad una lunghezza d'onda che lo caratterizza, specifico per un determinato colore. Ciascuna molecola di fotopigmento contiene un componente chiamato retinale ed una proteina detta opsina. Il retinale è comune a tutti i fotopigmenti, invece il tipo di opsina presente determina quali lunghezze d'onda sono assorbite da un determinato pigmento cromatico. Il fotopigmento del bastoncello è chiamato rodopsina che è localizzata nella membrana del segmento esterno.

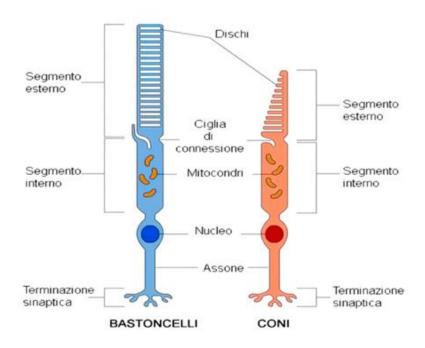


Figura 5 - Aspetto morfologico dei fotorecettori. [10]

I bastoncelli sono molto sensibili alla luce, ma la maggiore sensibilità è verso le lunghezze d'onda tra il blu ed il verde. In presenza però di luce intensa, tuttavia i bastoncelli sono saturati e non sono più in grado di dare informazioni sull'intensità della luce. Da ciò si deduce che i bastoncelli sono capaci di funzionare soltanto quando la luce è bassa, in gergo più tecnico luce crepuscolare. I coni invece sono meno sensibili dei bastoncelli alla luce e funzionano in presenza di luce crepuscolare. Ogni tipo di cono risponde alla luce in un determinato range di lunghezze d'onda a dispetto dei bastoncelli. Quelli che interessano il nostro studio sono quelli sensibili al rosso, molto sensibili ad una lunghezza d'onda di 560 nm, coni L (Figura 6).

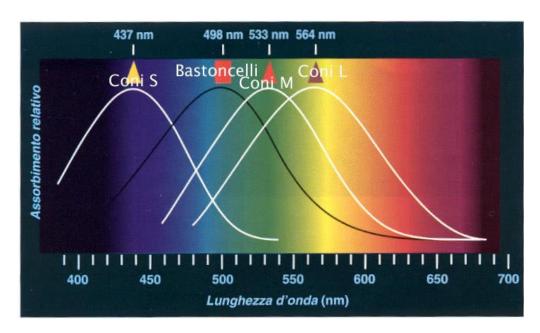


Figura 6 - Spettro di assorbimento dei diversi fotorecettori. [11]

A livello di questa lunghezza d'onda i coni rossi sono quelli che si occupano dell'assorbimento della luce. E' la rodopsina che si occupa di questa particolare funzione. Essa è una specifica proteina che se stimolata permette la visione dei colori. [15]

Quando ci si espone alla luce intensa i coni percepiscono i colori offuscati, questo significa che la rodopsina ha assorbito la luce, ma comporterà che non possa assorbire più luce, fino a quando non ritornerà nel suo stadio di riposo iniziale. Questo ciclo di rigenerazione del pigmento nell'occhio umano, si verifica su una scala temporale dei minuti. [14] [16] [17] [18]

La rodopsina non è il principale limite (facilmente by-passabile per la fisionomia dei coni a strati) che avremmo durante la stimolazione dell'occhio a palpebra chiusa ma avremmo anche:

- 1. attenuazione dovuta alla presenza della palpebra e
- 2. il non creare abbagliamento al paziente tramite fonte luminosa.

Difatti volendo impiegare i LED e metterli a contatto con la palpebra, ragionando su un normale ciclo visivo cui il paziente può essere sottoposto, l'abbagliamento è la condizione per la quale se la raggiungessimo non potremmo continuare con la stimolazione della retina e non vedremo, nell'acquisizione del segnale, nessun potenziale evocato.

Perciò per evitare l'abbagliamento dell'occhio esaminiamo in primis l'attenuazione dovuta dalla palpebra. Risultati ottenuti in uno studio condotto da un' istituto di ricerca universitario "Rensselaer Polytechnic Institute" in collaborazione con un istituto incentrato sulla ricerca illuminotecnica "Lighting Research Center" hanno portato ai seguenti risultati. [12] [13]

Lo studio è stato condotto posizionando direttamente una fonte calibrata molto piccola dietro la palpebra al centro della palpebra superiore con un rilevatore fisso (Figura 7.a e 7.b).

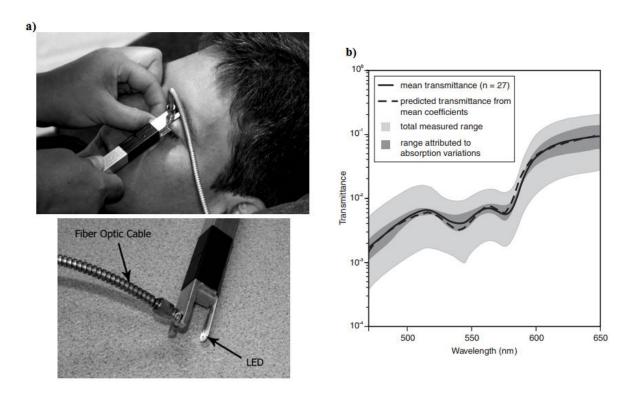


Figura 7- a) Posizione del LED sulla palpebra, b) misure di trasmittanza su lunghezze d'onda ricavata dalla palpebra di 27 pazienti.
[12][13]

La trasmittanza percentuale, ovvero la quantità di potenza termica scambiata tra due materiali [19], in questo caso tra il LED rosso e la palpebra, ottenuta a 630 nm è stata del 9 %.

Supponendo perciò che la palpebra attenui il 9 % dell'intensità luminosa totale che viene trasmessa dai LED alla palpebra stessa otteniamo comunque che l'intensità luminosa che arriverà sulla pupilla sarà di circa 900 mCd. [20] Quindi si arriverebbe ad avere seppur con la presenza della palpebra, ugualmente un buon valore di intensità luminosa sulla retina. Nel capitolo dell' Hardware definiremo tramite prove fotometriche e calcolo UGR, se i LED da noi scelti producono un abbagliamento sul paziente. Quest'ultimo è il secondo dei 2 limiti precedentemente elencati.

2.1.3 Le vie visive

Le cellule gangliari formano i neuroni ed esse danno origine ai potenziali d'azione che vengono poi trasportati nel sistema nervoso centrale. Gli assoni delle cellule costituiscono il nervo ottico, sono bensì 2 paia. I 2 nervi ottici fuoriescono dagli occhi a livello del disco ottico e confluiscono alla base del cervello, denominata lobo occipitale, per formare il chiasma ottico. A livello proprio del chiasma, metà degli assoni provenienti da ciascun occhio si incrocia per dirigersi verso l'altro lato del cervello (Figura 8).

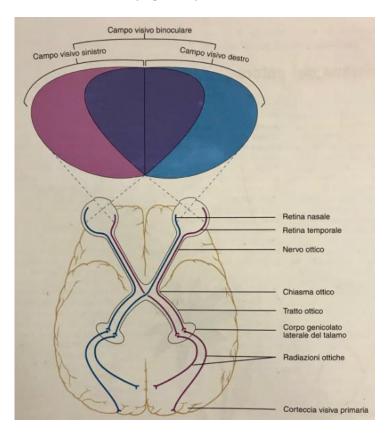


Figura 8 -Vie visive. [7]

Come da figura metà degli assoni provenienti da ciascun occhio si incrocia per dirigersi verso l'altro lato del cervello. I segnali relativi all'emicampo visivo di sinistra giungono alla porzione nasale della retina dell'occhio sinistro e a quella temporale dell'occhio destro. Allo stesso modo, i segnali provenienti dall'emicampo visivo di destra giungono all'emiretina dell'occhio destro e a quella temporale dell'occhio sinistro. Quindi entrambi gli occhi ricevono informazioni da entrambi i campi visivi. Gli assoni che arrivano dalle cellule gangliari nasali si incrociano, invece quelli che provengono dal lato temporale transitano ipsilateralmente. Si ottiene che tutto ciò che proviene dal campo visivo di destra viaggia attraverso assoni nel lato sinistro del cervello, invece i segnali che arrivano dal campo visivo di sinistra viaggiano nel lato destro del cervello. Perciò il campo visivo destro e sinistro sarà rappresentato rispettivamente nella corteccia visiva sinistra e destra.

Capitolo 3

Stato dell'arte

Nel 1973, Wright et al. furono i primi che valutarono come il monitoraggio dei VEP durante interventi chirurgici per la rimozione di tumori orbitali potesse prevenire disturbi visivi postoperatori. [21]

In letteratura sono presenti diverse pubblicazioni in conflitto riguardo l'utilità del monitoraggio dei VEP durante operazioni chirurgiche. Sono molti i fattori che possono influenzare la riproducibilità dei VEP durante gli interventi chirurgici, come le variabili fisiologiche, le lezioni preesistenti della funzione visiva, il tipo di intervento con la quale deve essere sottoposto il paziente, i particolari agenti anestetici o il tipo di stimolatore scelto. A causa della difficile ripetibilità del segnale, è stato documentato che in molti pazienti non è possibile ottenere dei VEP stabili.

Nel 1976 Feinsod et al., Harding et al. 1989 utilizzarono uno dei primi dispositivi mediante lenti a contatto sclerale che tramite una stimolazione visiva generava dei potenziali evocati. Era invasivo poichè venivano fissate al paziente mediante punti di sutura (Figura 9).

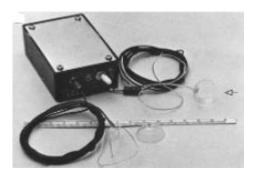


Figura 9 - Lente a contatto sclerale di Feinsod e Harding. [22]

Durante lo stesso anno venne proposto da Lesser [23] un dispositivo di stimolazione rappresentato da un paio di occhiali con LED rossi incorporati (Figura 10).



Figura 10 - Occhiale LED per stimolazione VEP. [23]

Ogni lente era costituita da 16 LED disposti in una matrice 4x4. Ogni LED consumava 10 mA ad un intensità pari a 1 mCd. Venne impiegato sia con palpebre chiuse che con palpebre aperte; nel primo caso si ottenne una luce diffusa, nel secondo si otteneva una serie di sorgenti a luce discreta. Chung durante questo studio applicativo, definì che l'illuminazione del dispositivo era variabile da 500 a 20000 lx.

Successivamente nel 1994, Pratt et al. [24] sviluppò un nuovo tipo di stimolatore costituito da 7 LED rossi con efficienza luminosa pari a 3 mCd ognuno a 20 mA. Un led posto centralmente ed un cerchio costituito dai 6 LED rappresentava il dispositivo cablato, fissato sull'occhio tramite una gomma di silicone e connesso al circuito driver tramite una coppia di fili. Lo stimolatore era compatto, facilmente applicabile sulla palpebra, durevole, resistente all'acqua e sterilizzabile a gas. Il limite grosso dell'utilizzo dell'occhiale a LED era però l'allineamento del dispositivo. Essendo costituite da lenti rigide, queste non si adattavano al profilo della bulbo oculare ed un minimo movimento causava il disallineamento dell'asse della luce con conseguente mancata registrazione dei VEP.

Nel 2010 Sasaki et al. [21] svilupparono un dispositivo di stimolazione a LED che riuscisse a garantire una corretta stimolazione della retina, con l'impiego anche di procedure chirurgiche. Il dispositivo era composto da 16 LED rossi ad alta luminosità, 100 mCd l'uno. I LED erano incorporati in un disco di silicone che rendeva la "lente" più flessibile per favorirne un miglior adattamento al bulbo oculare. Il dispositivo perciò possedeva una luminosità che andava da 5000 a 20000 lx. Ogni stimolo aveva una durata di 20 ms ed una frequenza di 1 Hz, fornendo al sistema 100 stimoli (Figura 11).





Figura 11 - Dispositivo stimolazione contenente 16 LED rossi con disposizione su paziente. [21]

Kodama [25] fece altrettanto e propose un dispositivo con caratteristiche intermedie tra un occhiale a LED ed una patch a LED. Era costituito da 15 LED rossi contenuti all'interno di una plastica rinforzata. L'intensità dei LED era di 175 mCd ciascuno e poteva essere variata da una corrente tra gli 0 e i 20 mA. Ogni stimolo aveva una durata di 40 ms ed una frequenza di 1,25 Hz, per un totale di 40 stimoli (Figura 12).





Figura 12 - Stimolazione a LED, con disposzione su paziente. [25]

Uno dei più recenti è lo studio realizzato da Luo et al. [26] nel 2015 che utilizzando il metodo di Kodama stimolava sui paziente con un nuovo dispositivo della Inomed. Questo dispositivo forniva uno stimolo di durata pari a 10 ms e utilizzava una frequenza di 1,1 Hz. Aveva una luminosità che variava fino anche a 26000 lx, ma si accorsero che già anche a 1000 lx sui pazienti si ottenevano registrazioni attendibili (Figura 13).

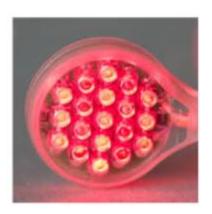




Figura 13 - Dispositivo stimolazione LED, disposizione su paziente. [26]

La tabella sottostante riassume tutti i progressi fatti nello studio dei VEP fino al 2015 (Tabella 1).

Utilizzato da	Nr. e colore led	Intensità Luminosa	Nr. di stimoli	Durata singolo stimolo (ms)	Frequen za stimoli (Hz)
Kodama et al.	15 Red (660nm)	0 to 175 mCd	40 to 100	40	1
Sasaki et al.	16 Red	100 mCd	100	20	1
Chung et al.	/	2.000 Lx	/	20	1
Kamio et al.	nio et al. 16 Red 100mCd		100	20	1
Luo et al.	19 Red (654nm)	Up to 26.000 Lx	100-200	10	1.1

Tabella 1 - Confronto caratteristiche dispositivi descritti.

Infine lo studio del 2016 condotto dalla Dottoressa Novelli del Politecnico di Torino ha puntato sul rimodernizzare il dispositivo di stimolazione. Partendo dalla realizzazione di un dispositivo costituito da 13 LED su PCB flessibile, si stimolava il paziente con uno stimolo stabile, non arrecando danni alla retina e poterlo utilizzare anche quando il paziente stesso non può cooperare (Figura 14). [27] Ed è proprio da quest'ultimo studio da cui partiremo per perfezionare ancora di più il dispositivo di stimolazione per renderlo più competitivo sul mercato.

Proseguendo nella monografia verranno definite meglio le considerazioni e le modifiche apportate.

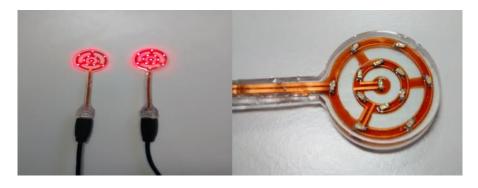


Figura 14 - Dispositivo di stimolazione GEPEV. [27]

Capitolo 4

Analisi delle specifiche e delle scelte sul dispositivo

Al giorno d'oggi possiamo contare su una vasta gamma di microcontrollori disponibili sul mercato. L'analisi delle modifiche per il conseguimento del progetto SIPEV ci ha spinto nella scelta di dimensioni contenute, a basso costo, con buone performance di calcolo e consumi.

L'impiego del dispositivo sarà oltre che ospedaliero (impiego nel reparto neurochirurgia) in un secondo momento potrà essere anche utilizzato al di fuori della sala operatoria per analisi di routine su pazienti in cura. Visto che il dispositivo SIPEV sarà costituito da 3 dispositivi i quali dialogheranno tra loro tramite tecnologia wireless, ci siamo interrogati su quanta banda di segnale in frequenza potevamo permetterci di usare senza andare ad interferire sulla frequenza utilizzata dalle già presenti apparecchiature elettromedicali della sala operatoria e la tecnologia wireless più opportuna.

4.1 Cenni sulla tecnologia Wireless

La tecnologia wireless oggigiorno è sempre più diffusa e tocca molti ambiti di interesse. L'utilizzo di questa tecnologia è dovuta soprattutto alla comodità nel suo utilizzo, l'assenza infatti di un collegamento concreto tra le parti migliora la fruibilità dei dispositivi. Questa tecnologia è vista nella sua totalità in chiave innovativa in maniera positiva, sia dagli utilizzatori che dai progettisti che puntano sempre più a rimodernizzare i dispositivi già in commercio rendendoli sempre più innovativi. [28]

Il mercato mette a disposizione molte tipologie di tecnologie wireless che si distinguono per: un'ottimizzazione del consumo energetico sul dispositivo munito di batteria, una maggiore predisposizione per una coesistenza con altri protocolli di comunicazione nell'ambiente circostante e raggio d'azione e velocità dei pacchetti differenti. Di seguito viene riportato l'elenco e la descrizione delle principali tecnologie wireless disponibili sul mercato.

- Wi-Fi è la tecnologia cablata più veloce con larghezza di banda e capacità fino a 54 Mbit/s. E' basata su IEEE 802.11a 802.11b, 802.11ge, 802.11 e ne utilizza un collegamento radio da 2,4 GHz e/o 5 GHz. Questa tecnologia è utilizzata per una vasta gamma di applicazioni, come dispositivi medici (sistemi di monitoraggio, strutture diagnostiche, etc.) , consumatori elettronici (tablet, smartphone, automazione industriale e domestica, sistemi di sicurezza, videosorveglianza etc (Figura 15). [30]
- ZigBee è lo standard utilizzato per creare reti mesh di dispositivi e sensori a bassa potenza. Usa la frequenza di lavoro del segnale radio a 2,4 GHz. ZigBee è ampiamente distribuito nei sistemi di automazione nell'industria, nella medicina, nella logistica, nei sistemi di contabilità del consumo di energia (Tabella 2) (Figura 15). [30]

- Bluetooth è una delle tecnologie più popolari per il collegamento di dispositivi, situato a breve distanza fino a diverse centinaia di metri. Il bluetooth ti consente di lavorare in modalità punto a punto ed in modalità stella che sono standard utilizzati per scambiare informazioni tra due dispositivi. Per la trasmissione la gamma è a 2,4 GHz. Il bluetooth è utilizzato nell'elettronica portatile (orologi, smartphone, etc.) in sensori, elettronica di consumo e ambito medico (Tabella 2). [30]
- Bluetooth low energy (BLE) è la versione bluetooth, finalizzata però ad ottenere un livello minimo di consumo. Non esiste una connessione permanente tra dispositivi BLE, essa consente la modalità di sospensione. Di conseguenza i dispositivi possono funzionare per più di un anno in modo autonomo, senza sostituire la batteria. E' impiegato in dispositivi per sportivi, medicina etc (Tabella 2) (Figura 15).
- Near Field Communication (NFC): è una tecnologia che opera ad una distanza di alcuni cm. L'NFC usa un segnale a 13,56 MHz. Lo scambio di dati avviene tra 2 dispositivi: il "lettore" ed il "trasponder". In questo caso, l'iniziatore è in grado non solo di trasmettere e viceversa dati ma anche di fornire potenza al trasponder. I dispositivi NFC non contengono la batteria (Tabella 2) (Figura 15).

	Bluetooth	Low power BT	Zig bee	NFC
Distanza	10-100m	>10m	> 100m	< 1 m
Potenza	1	0.1	0.6	0 (ric. dal reader)
Bit rate	3 Mbit/s	1 Mbit/s	0.25 Mbit/s	0.4 Mbit/s
Robustezza	Alta	Eccellente	Bassa	Alta
Freq. radio	2.4 GHz	2.4 GHz	2.4 GHz	UHF VHF
Mercato	Consumer	Healthcare, automotive, cellulari, home	industria	Pagamento elettronico,tick et, identificazione

Tabella 2 - Confronto tra alcune tecnologie wireless. [29]

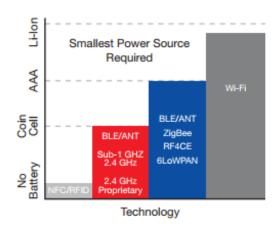


Figura 15 - Batterie rapportate con le tecnologie wireless. [30]

4.2 Analisi dispositivi sala operatoria e della banda concessa

La scelta della tecnologia wireless che impieghiamo non dovrà andare in conflitto con la frequenza delle apparecchiature elettromedicali. La struttura di una sala operatoria è costituita da: apparecchi e dispositivi elettromedicali, tavolo operatorio, impianto elettrico e di massa, impianto di illuminazione etc. La sicurezza è una caratteristica importante nell'utilizzo di un dispositivo medico e si intende anche come sicurezza sia intrinseca del MD(medical device) che come immunità del MD verso disturbi irradiati e quelli ricevuti. La normativa ai fini della realizzazione del nostro dispositivo è molto rigida nei confronti di questo aspetto. Essa deve garantire la convivenza in tutta sicurezza delle diverse apparecchiature elettromedicali. Tra le apparecchiature elettromedicali quelle che causano maggiormente disturbi sono: l'elettrobisturi e i laser. [31] [32]

L'elettrobisturi causa emissioni maggiori uguali ai 500 KHz con potenze sui 300 W, o laser per chirurgia con frequenze che vanno da 1 MHz a 20 MHz. Quello che desta maggiore preoccupazione è più che altro quella dell'elettrobisturi con la tecnologia da noi scelta molto presente in ambito operatorio. Per far fronte a questa scelta esiste la banda ISM(Industrial, scientific and Medical Radio Bond). Essa si riferisce ad un gruppo di bande radio o parti dello spettro radio che sono internazionalmente riservate all'uso di energia a radifrequenza (RF). Le bande ISM sono bande di frequenza aperta, che variano a seconda delle regioni e dei permessi. La banda ISM da 2,54 GHz è una banda comunemente accettabile per le operazioni in tutto il mondo e la sua rapida crescita ha favorito il suo impiego per dispositivi Bluetooth, Wi-fi e NFC. Pertanto sarà la tecnologia Bluetooth low energy (BLE) a determinare il corso dell'intero processo di sviluppo del nostro progetto. La nostra scelta è ricaduta sulla tecnologia BLE che è quella che si presta meglio verso questa convivenza, visto che esso sfrutta frequenza dell'ordine dei 2,4 GHz. Fatte le scelte opportune della tecnologia wireless impiegata e le nostre opportune valutazioni dovute anche all'impiego e possibilità di limitare i consumi e del fatto che il protocollo Bluetooth trasmette in un range fino a 100 metri e velocità di trasmissione dati di 1 Mbit/s - 305 Kbit/s, abbiamo scelto un dispositivo programmabile con questa tecnologia di trasmissione.

4.3 Descrizione e caratteristiche della scheda della Texas Instruments LAUNCHXL-CC2640R2F

La launchpad LaunchXL-CC2640R2F è una scheda che possiede sulla stessa il MCU cc2640R2F che fa parte della famiglia di microcontrollori cc26XX della Texas Instruments (TI) e si presta molto bene alla tecnologia wireless da noi scelta, perché contiene il protocollo di trasmissione e ricezione BLE.

Esso è un microcontrollore che possiede le seguenti caratteristiche:

- ARM Cortex M3;
- può lavorare ad una frequenza massima di clock di 48 MHz;
- possiede un Sensor Controller a bassissima potenza;
- e le seguenti principali periferiche:
- pin di GPIO;
- 4 moduli timer (GPTM);
- UART;
- ADC;
- SPI;
- Real-Time Clock (RTC).

Il chip cc2640R2F [33] è progettato per sistemi embedded a basso costo ed in particolare per applicazioni a bassa potenza. Funziona ad una tensione da 1,8 V a 3,8 V con i seguenti consumi in termini di bassa potenza:

- in modalità attiva RX: 5,9 mA;
- in modalità attiva TX: 6,1 mA;
- Standby: 1 μA;
- in modalità attiva del Sensor Controller: 8,2 μA/MHz.

Il dispositivo è programmabile tramite sistema JTAG, o come nel nostro caso più facilitato grazie alla presenza di un connettore USB (Figura 16). In una successiva parte di prototipazione definitiva, la JTAG tornerà utile come pin per un suo montaggio e programmazione su PCB (Capitolo 7 PCB).

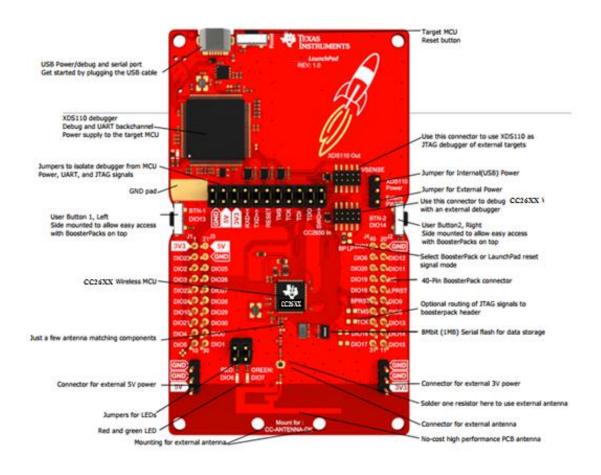


Figura 16 - Launchpad cc2640R2F. [34]

Il CC2640R2F è ideale per molte applicazioni, in cui è importante la durata della batteria. [35]

Il Bluetooth low energy(BLE) è incorporato nella RAM e funziona sul processore ARM Cortex M0. Questa disposizione migliora le prestazioni del sistema ed il consumo energetico liberando memoria flash. Di CC2640 esistono vari package con varie dimensioni (Tabella 3).

PART NUMBER	PACKAGE	BODY SIZE (NOM)
CC2640F128RGZ	VQFN (48)	7.00 mm × 7.00 mm
CC2640F128RHB	VQFN (32)	5.00 mm × 5.00 mm
CC2640F128RSM	VQFN (32)	4.00 mm × 4.00 mm

Tabella 3 - Vari package cc2640 e dimensioni. [33]

Sulla Launchpad cc2640R2F abbiamo il package RGZ ma in un fine realizzativo del dispositivo di stimolazione ci affideremo alla 5 mm x 5 mm quad flat no lead (QFN) con 32 pin, 15 dei quali sono delle GPIO (Figura 17.a e 17.b)(Tabella 4).

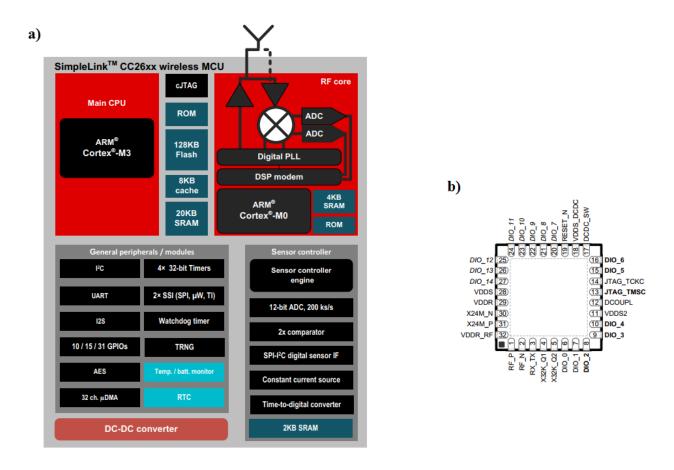


Figura 17 - a) Diagramma blocchi della cc2640, b) Pin di I/O package RHB. [33]

NAME	NO.	TYPE	DESCRIPTION
DCDC_SW	17	Power	Output from internal DC-DC
DCOUPL	12	Power	1.27-V regulated digital-supply decoupling
DIO_0	6	Digital I/O	GPIO, Sensor Controller
DIO_1	7	Digital I/O	GPIO, Sensor Controller
DIO_2	8	Digital I/O	GPIO, Sensor Controller, high-drive capability
DIO_3	9	Digital I/O	GPIO, Sensor Controller, high-drive capability
DIO_4	10	Digital I/O	GPIO, Sensor Controller, high-drive capability
DIO_5	15	Digital I/O	GPIO, High drive capability, JTAG_TDO
DIO_6	16	Digital I/O	GPIO, High drive capability, JTAG_TDI
DIO_7	20	Digital/Analog I/O	GPIO, Sensor Controller, Analog
DIO_8	21	Digital/Analog I/O	GPIO, Sensor Controller, Analog
DIO_9	22	Digital/Analog I/O	GPIO, Sensor Controller, Analog
DIO_10	23	Digital/Analog I/O	GPIO, Sensor Controller, Analog
DIO_11	24	Digital/Analog I/O	GPIO, Sensor Controller, Analog
DIO_12	25	Digital/Analog I/O	GPIO, Sensor Controller, Analog
DIO_13	26	Digital/Analog I/O	GPIO, Sensor Controller, Analog
DIO_14	27	Digital/Analog I/O	GPIO, Sensor Controller, Analog
JTAG_TMSC	13	Digital I/O	JTAG TMSC, high-drive capability
JTAG_TCKC	14	Digital I/O	JTAG TCKC
RESET_N	19	Digital input	Reset, active-low. No internal pullup.
RF_N	2	RF I/O	Negative RF input signal to LNA during RX Negative RF output signal to PA during TX
RF_P	1	RF I/O	Positive RF input signal to LNA during RX Positive RF output signal to PA during TX
RX_TX	3	RF I/O	Optional bias pin for the RF LNA
VDDR	29	Power	1.7-V to 1.95-V supply, typically connect to output of internal DC-DC
VDDR_RF	32	Power	1.7-V to 1.95-V supply, typically connect to output of internal DC-DC
VDDS	28	Power	1.8-V to 3.8-V main chip supply
VDDS2	11	Power	1.8-V to 3.8-V GPIO supply
VDDS_DCDC	18	Power	1.8-V to 3.8-V DC-DC supply
X32K_Q1	4	Analog I/O	32-kHz crystal oscillator pin 1
X32K_Q2	5	Analog I/O	32-kHz crystal oscillator pin 2
X24M_N	30	Analog I/O	24-MHz crystal oscillator pin 1
X24M_P	31	Analog I/O	24-MHz crystal oscillator pin 2
EGP		Power	Ground – Exposed Ground Pad

Tabella 4 - Descrizione pin del chip cc2640. [33]

Di seguito sono riportate le funzioni e i concetti chiave caratteristici della cc2640.

4.3.1 BLE

Il Bluetooth Low Energy (BLE) è un protocollo che consiste di due parti: l' Host ed il Controller. L'interfaccia tra i due costituisce la gestione della comunicazione tra le parti. [33]

Analizziamo le funzionalità dei diversi livelli (Figura 18).

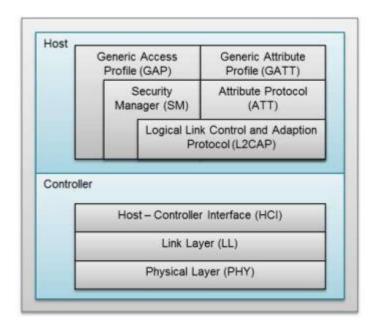


Figura 18 - Protocollo BLE. [33]

Il Controller è costituito dal Physical Layer ed il Link Layer (Figura 18).

Il Physical Layer (PHY) è il livello più basso dello stack e definisce le caratteristiche di base del BLE. Opera nella banda ISM definendo 40 canali a radiofrequenza, tra cui: i canali di advertising e i canali di dati. I canali di advertising sono tre e sono stati assegnati alla frequenza centrale, mentre per i 37 canali di dati viene utilizzato l' "adaptive frequency hopping" per ridurre al minimo le interferenze ed evitare altri problemi di trasmissione wireless. Facendo in questo modo si stabilisce che solo uno dei canali di dati disponibili viene selezionato per la comunicazione durante un dato intervallo di tempo. La velocità di trasmissione è di 1 Mb/s. [37]

Il Link Layer (LL) è il livello a cui corrisponde l'interazione della connessione tra due dispositivi. L'advertiser è un dispositivo che trasmette pacchetti di advertising (di riconoscimento), mentre i dispositivi che mirano solo a ricevere i dati sono chiamati scanner.

l BLE definisce i ruoli del master e dello slave: il master può gestire più connessioni, mentre ogni slave può essere collegato ad un solo master. Una rete composta da un master e il suo slave si chiama "piconet" ed un dispositivo BLE può appartenere solo ad una piconet. Dopo aver creato una connessione, il canale di comunicazione viene suddiviso in unità di tempo non sovrapposte

chiamate eventi di connessione. Un evento di connessione è considerato aperto fino a quando il master e lo slave continuano ad alternarsi nell'invio di pacchetti di dati. [37] I parametri di connessione più importanti sono:

- 1. connection interval
- 2. slave latency
- 3. supervision timeout
- 4. effective connection interval

Il connection interval è il tempo che intercorre tra due eventi di connessione ed è multiplo di 1,25 ms nell'intervallo tra 7,5 ms e 4 s.

Lo slave latency rappresenta il numero di eventi di connessione consecutivi che lo slave può ignorare ed è un numero intero compreso tra 0 e 499.

Il supervision timeout definisce il tempo massimo che può intercorrere tra due eventi di connessione ed è compreso tra 100 ms e 32 s.

L' effective connection interval è il tempo effettivo tra due eventi di connessione e supponendo che lo "slave latency" sia diverso da zero, corrisponde alla seguente formula:

Connection Interval $*(1 + Slave\ Latency)$

L'Host invece è costituito dai seguenti layer: L2CAP, ATT, GATT, Security Manager e GAP and Application Profile (Figura 18). [37]

- L' L2CAP è un protocollo ottimizzato basato sul Bluetooth standard il cui obiettivo principale è quello di multiplexare i dati dei tre protocolli del livello superiore.
- L' ATT (Attribute Protocol) è il livello che definisce la comunicazione e lo scambio di dati tra due dispositivi che svolgono il ruolo di server e client.
- Il GATT (Generic Attribute Protocol) determina il ruolo di client o server ed è indipendente dal ruolo di master e slave. Il GATT è basato sul livello precedente per il processo di ricerca e lo scambio di caratteristiche da un dispositivo a un altro.
- Il layer Security Manager che tutela lo scambio di informazioni tra due dispositivi connessi, un livello di sicurezza per il Link Layer e uno per l'ATT.
- Il GAP (Generic Access Profile) and Application Profile che è il livello più alto dello stack del BLE e definisce i ruoli, le modalità, le procedure per la ricerca, la gestione della creazione e della sicurezza della connessione. Il GAP specifica quattro ruoli:
 - 1. Broadcaster,
 - 2. Observer,

- 3. Peripheral,
- 4. Central.

Un broadcaster è un dispositivo che trasmette solo eventi di advertising e non supporta connessioni con altri dispositivi;

un observer è un dispositivo che riceve solo eventi di advertising quindi è complementare al broadcaster;

il central che è incaricato di avviare e gestire una connessione, mentre infine un dispositivo con il ruolo di peripheral è in grado di accettare la creazione di una connessione.

Una cosa molto importante è che un dispositivo può supportare diversi ruoli, ma solo uno alla volta.

4.3.2 SPP (Serial Port Profile)

La comunicazione che si verrà a creare tra server e client dovrà essere bidirezionale rispettivamente tra il dispositivo centrale e il dispositivo di stimolazione, permettendo un continuo scambio di informazioni. L'SPP o Serial port profile, è tratto da un'esempio disponibile su Simple Link Academy che più si avvicina a quello che volevamo realizzare. Essa può anche servire per raggiungere una comunicazione autonoma senza che ci siano interazioni con l'utente. Negli altri casi che abbiamo nel BLE crea interazioni tra i dispositivi definendo: master, slave (che sono i medesimi di central e peripheral). Nel master il dispositivo ricerca un altro dispositivo e avvia la connessione come per il centrale, [38] lo slave invece è il dispositivo che comunica la sua presenza e aspetta per la connessione, accettando la richiesta di connessione. [39]

Il client invece è il dispositivo che accede a delle risorse remote attraverso il protocollo GATT, descritto nel paragrafo 4.4.1, che trasmette dati tra due dispositivi connessi. Il server è il dispositivo con un database interno di connessioni, contiene anche altri parametri di connessione e fornisce i dati per poter accedere ad un client remoto.

Un master potrà essere connesso a più slave, massimo 8 dispositivi, mentre lo slave potrà essere connesso ad un solo master. Un master ed uno slave possono avere entrambi questi ruoli di client e di server, basta che si definiscano le funzionalità: struttura e flusso di dati nel GATT.

I dati possono essere letti, scritti, notificati etc. I dati scritti sono quelli dove il client invia i dati al server (p.e. la centrale manderà i 3 parametri di stimolazione). Nel caso della lettura il client leggerà i dati dal server su sua richiesta. La notifica invece è quando il server invia i dati al client senza che questo li abbia richiesti (p.e. l'invio del valore del timer alla centrale).

L'SPP Ble Server metterà direttamente in UART tutto ciò che riceve via BLE emulando una porta seriale. La struttura del progetto è simile a quella del SimpleBLEPeripheral (Figura 19). [36]

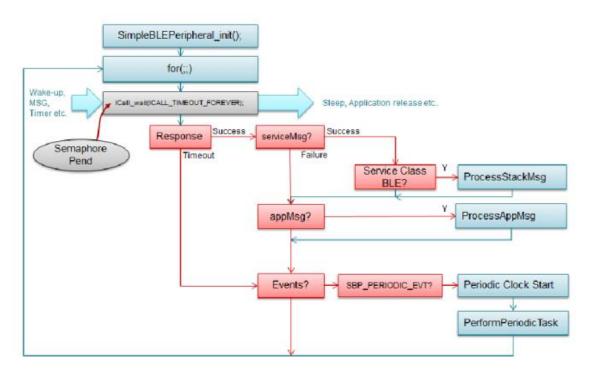


Figura 19- Flow chart Simple BLE pheripheral. [36]

Dopo l'inizializzazione iniziale, il BLE si pone subito in "advertising" e attende di essere individuato da un dispositivo che esegue scansioni per creare una connessione. Una volta creata la connessione, l'applicazione può ricevere dei dati dell'utente tramite UART dal livello SDI (Serial Data Interface) ed inviarli wireless come pacchetti di notifica al dispositivo connesso, se questo, svolge il ruolo di "client" ed ha abilitato la notifica. Viceversa i dati inviati dal "client" non arrivano direttamente all'applicazione del "server"; essi infatti vengono ricevuti a livello del profilo seriale e vengono trasmessi sull'UART tramite il livello SDI. È possibile trasferire i dati anche al livello dell'applicazione per un'ulteriore elaborazione. Partendo da questo codice, infatti si sono aggiunte le operazioni necessarie per abilitare la comunicazione cambiando: relativi parametri, acquisire il programma per l'elaborazione tramite SC del PWM e gestire l'invio della variabile timer per mantenere la sincronia tra i dati di stimolazione e lettura.

4.3.3 GPIO, GPTIMER, IOC, PRCM

Le porte di GPIO sono definite nella Tabella 4, come anche le porte IOC che mediante opportuni driver si possono programmare determinati pin a proprio piacimento. [33][40][41]

I timer programmabili (GPTIMER) possono essere utilizzati per contare o temporizzare eventi esterni che guidano i pin di ingresso del timer. Il general multipurpose timer module (GPTM) fornisce due timer da 16 bit (timer A e timer B) che possono essere configurati per funzionare in modo indipendente come timer oppure essere concatenati per funzionare come timer a 32 bit. Altre risorse di sistema includono un timer di sistema detto "systick" che fornisce un contatore a 24 bit . Il GPTM contiene 4 blocchi principali con varie modalità di funzionamento; i timer possono essere utilizzati a:

- 16 bit con 8 bit di prescaler o 32 bit denominato programmable one- shot timer;
- 16 bit con 8 bit di prescaler o 32 bit detto progammable periodic timer;
- 32 bit per ogni capture compare PWM per pin (CCP);
- 24 bit input- edge (attivo sul fronte) 24 bit timer capture modes
- 24 bit per PWM.

Altre modalità che possiede sono: count up, count down; rispettivamente conteggio in avanti ed indietro. O "free running", che continua a contare liberamente.

Il PRCM o Power reset and clock nanagement, è utilizzato per la configurazione e generazione del clock; tornerà utile per la sincronizzazione, come anche i timer sia per la sincronizzazione con oscillatore esterno che per il ricreare un segnale di ON/OFF programmando l'accensione e lo spegnimento dei LED.

4.4 Idea sincronizzazione

Come detto nell'obiettivo e nei passaggi successivi: Introduzione e Analisi delle specifiche e scelte sul dispositivo, il punto cardine su cui il SIPEV pone le fondamenta è innanzitutto la rimozione dei cavi.

I 3 dispositivi: dispositivo di stimolazione, dispositivo di lettura dei potenziali evocati e dispositivo centrale dovranno dialogare tramite modulo BLE mantenendo la comunicazione tra loro in modo continuo per far coincidere istante di stimolazione con l'istante di lettura.

I dispositivi di stimolazione e di prelievo saranno dispositivi usa e getta, perciò saranno alimentati da batteria a bottone opportunamente scelta; il problema dell'alimentazione non sussisterà per il dispositivo centrale che è come se fosse raffigurato da un PC che elabora i dati che arrivano dagli altri 2 dispositivi.

Tutti i dispositivi sono dotati di un timer interno lasciato libero di correre. Perciò vorremmo ottenere che per un istante di tempo (RTOS - Real Time Clock) nella quale la centrale manda un dato al dispositivo di stimolazione esso coincide proprio col medesimo istante (senza latenze dovute alla trasmissione del dato, o tempo operativo del dispositivo ricevente di elaborare il dato stesso) con la quale il dispositivo di stimolazione riceve il dato. Il "dato trasmesso" e "dato ricevuto" idealmente sarebbe trasmesso e ricevuto in tempo 0. Purtroppo questo "tempo reale" tra i dispositivi è di difficile se non impossibile realizzazione poiché troppo idealizzato, della latenza tra i dispositivi ci sarà sempre (Capitolo 8 Prove e test sui dispositivi) (Figura 20).

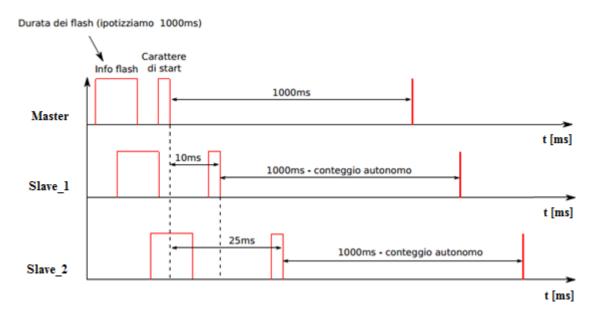


Figura 20 - Interpretazione schema temporale di sincronizzazione tra i dispositivi.

Perciò si partirà con l'avere i 3 dispositivi tutti connessi con rispettivi 3 connettori (che all'inizio verranno posizionati e poi verranno rimossi) ad un pulsante ON/OFF che provocherà un set del timer interno ad ognuno di essi. Nell'esatto istante in cui il pulsante verrà premuto tutti e tre i timer dei rispettivi 3 dispositivi partiranno nello stesso istante e si creerà così facendo una sincronizzazione dei dispositivi nel tempo (Figura 20). Per esempio ipotizziamo che il master mandi il pacchetto di "Carattere di start" con un pacchetto di "Info flash" iniziale. Questo pacchetto arriverà con un certo lasso di tempo (p.e. 10 ms) ad uno slave ed all'altro con una latenza chiaramente maggiore di quello di prima (p.e. 25 ms). Dopodiché gli slave inizieranno a fare le loro operazioni contenute nell'"Info flash". Appena terminate manderanno, chi prima, chi dopo al master tutto ciò che ha svolto nel tempo pre-settato (p.e. 1000 ms) ed il master infine allineerà le tempistiche dei due slave. Questa è l'interpretazione iniziale che abbiamo dato ad una possibile soluzione per sincronizzare i 3 dispositivi. Gli slave nell'interpretazione iniziale della sincronizzazione sono stati entrambi considerati come "dispositivi di stimolazione" per semplificare i ragionamenti, ben consapevoli che i 2 dispositivi che abbiamo a disposizione sono uno di stimolazione e l'altro di lettura. Questo ragionamento e interpretazione iniziale però delinea già molto bene l'approccio alla sincronizzazione dei dispositivi. Tutto questo ragionamento è stato fatto perché il BLE non presenta un suo tempo di clock interno con la quale pre-settare il clock dei timer, se ci fosse avremmo risolto la questione della sincronizzazione fin da subito impostando in ognuno di essi il valore di clock del BLE, (paragrafo 4.3 Descrizione e caratteristiche della Launchpad cc2640R2F) omettendo di fatto il pulsante esterno per sincronizzarli tutti assieme rendendo predittivo il sistema.

La frequenza con la quale lavora il chip cc2640 è di 24 MHz [33] che internamente viene raddoppiata. La tolleranza dell'oscillatore a cristallo è di 80 ppm e nelle 4 ore di possibile durata operatoria avremmo:

$$\frac{80}{1000000} = 8 * 10^{-5}$$

e nelle 4 ore avrò:

$$4 h * \frac{60 min}{1 h} * \frac{60 s}{1 min} * \frac{1000 ms}{1 s} = 14,4 * 10^6 ms$$

$$14,4*10^6 \ ms*8*10^{-5} = 1120 \ ms*\frac{1 \ s}{10^3 \ ms} = 1,12 \ s \cong 1 \ s$$

Perciò la massima deviazione dell'oscillatore dopo un tempo di intervento delle 4 ore è di 1 secondo mentre nelle 8 ore è il doppio e questo noi non possiamo permettercelo per questioni di mantenimento della sincronia tra i dispositivi.

Da datasheet [33] è dichiarato che è consentito posizionare un oscillatore esterno che abbiamo al massimo una frequenza che sia pari ad 1/4 della frequenza di lavoro del chip. Perciò avendo un frequenza pari a 48 MHz avremmo da 12 MHz in giù in frequenza da poter scegliere tra gli oscillatori presenti sul mercato. La soluzione quindi è data dall'impiego di un oscillatore esterno per risolvere il problema dovuto dall'oscillatore.

Il TCXO è un oscillatore di cristallo compensato in temperatura che offre eccellenti caratteristiche di temperatura combinato con un basso consumo energetico ma un conseguente rapido riscaldamento. [42]

Il TCXO presenta una stabilità in frequenza di 2 ppm e svolgendo i calcoli di prima in un tempo di intervento delle 8 ore otterremmo una deviazione dell'oscillatore di 0,057 sec; scostamento ben diverso in termini di tempo rispetto all'oscillatore presente sulla Launchpad. L'oscillatore di cristallo compensato in temperatura verrà utilizzato per fare dei test e delle misure tra i dispositivi, per rispettare i nostri standard di tensione abbiamo dovuto ricorrere all'impiego di questo oscillatore. Altri oscillatori in commercio sarebbero stati più indicati per il nostro progetto ma le tensioni che richiedono sono superiori di quella utilizzata dalla batteria a bottone impiegata.

Si procederà perciò in quest'ordine. Una volta partiti assieme, tramite il set del loro timer interno mediante un bottone esterno, viene creata la sincronia tra essi. I timer interni saranno liberi di correre ed il dispositivo centrale inizierà la ricerca tramite BLE dei dispositivi nelle sue vicinanze, appena lo troverà si connetterà ad esso.

Quando si connetterà al dispositivo di stimolazione manderà a discrezione dell'utente 3 parametri: durata dello stimolo, numero di stimoli e periodo.

Viceversa il dispositivo di stimolazione ricevuta l'autenticazione con dispositivo centrale, riceverà i 3 parametri e inizierà il suo ciclo di stimolazione sulle palpebre del paziente e rispettivamente al fronte di salita del PWM dei LED manderà al dispositivo centrale il tempo con la quale avrà effettuato questo stimolo. Successivamente per continuare a creare questa sincronia tra i 2, manderà i successivi tempi rispettivi del fronte di salita del PWM i-esimi alla centrale.

In un ultimo step la centrale riceverà i parametri di tempo che corrispondono al tempo in cui sono stati effettuati gli N stimoli dal dispositivo di stimolazione.

Caso analogo succederà per il sistema di lettura dei potenziali evocati, dove dopo essersi connesso con la centrale partirà registrando il segnale prelevato e manderà i dati alla centrale. Oltre i dati manderà anche il tempo con la quale questi dati sono stati acquisiti, che corrisponderà al tempo iniziale con la quale leggerà i dati dall'occipite. La centrale così facendo creerà computazionalmente una correlazione tra i dati della stimolazione e i dati della lettura. Creando in questo modo tramite i segnali di tempo ricevuti: quando il dispositivo di stimolazione ha effettivamente iniziato a stimolare l' N-esimo stimolo e quando il dispositivo di lettura ha iniziato la lettura dell'i-esimo VEP, un riferimento temporale tra tutti i dispositivi in gioco (Figura 21).

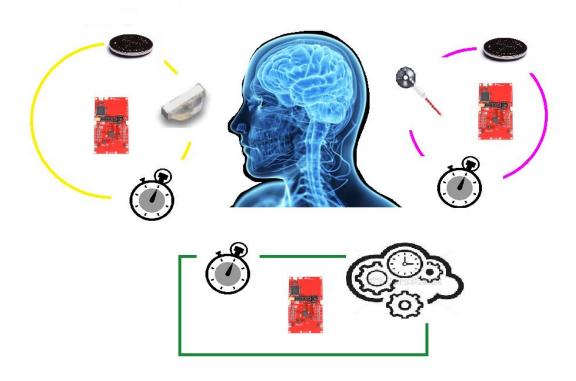


Figura 21 - Schema riassuntivo dell'accoppiamento dei 3 dispositivi.

Le ulteriori ipotesi avanzate, magari anche future accennate durante la sperimentazione di questo progetto della parte di sincronizzazione sono state oltre il cavetto tra dispositivo di stimolazione e lettura dello scalpo per mantenere la sincronizzazione dei dispositivi, di posizionare un buzzer sul dispositivo di stimolazione. Questo produrrebbe un cicalino ad ogni tick di stimolazione sulla palpebra e nel lato del prelievo a livello dell'occipite si posizionerebbe un ricevitore. Così facendo si creerebbe una possibile alternativa a quanto descritto nella tesi di sincronizzazione tra i dispositivi mediante oscillatore esterno. L'alternativa è il segnale di broadcast tra dispositivi, dove la centrale invia la stessa informazione a tutti i dispositivi appartenenti alla rete simultaneamente, presenti online in quel momento.

Capitolo 5

Hardware

Il progetto SIPEV prende spunto da un precedente studio di fattibilità, meglio noto come GEPEV. Durante questo studio sono state riscontrate delle criticità sugli attuali dispositivi di stimolazione ottica per la generazione dei potenziali evocati visivi ed il loro monitoraggio intraoperatorio. Questo studio da cui trarremo dalle conclusioni il nostro punto di partenza di studio, è ben definito dalla tesi Magistrale di Valeria Novelli: "Sviluppo di un dispositivo ottico per la genesi dei potenziali evocati intraoperatori : setup del protocollo di stimolazione, esecuzione delle misure, elaborazione ed analisi dei risultati". [27]

Nel GEPEV si è puntato e si è arrivati alla constatazione dei seguenti punti:

- la scelta del colore opportuno dei LED da impiegare sulle palpebre;
- l'intensità luminosa necessaria per poter effettuare la stimolazione sulle palpebre (nel nostro caso cambierà per questioni relative ai consumi);
- la geometria e layout di disposizione dei LED sulle palpebre.

La disposizione geometrica migliore dei LED, anche per convogliare meglio il fascio luminoso sulla pupilla è quella riportata nella figura sottostante (Figura 22). Dove in questo caso sono stati disposti in cerchi concentrici 13 LED.



Figura 22 - Colore e disposizione dei LED (GEPEV). [27]

In GEPEV il colore ottimale per i LED è il led rosso ad alta luminosità (hyper red). Esso è a basso consumo di potenza, con dimensioni ridotte (1.6 mm x 0.8 mm x 1.1 mm) di tipo SMD (Surface Mount Device). Essi posseggono un'intensità luminosa minima pari a 120 mCd ed una corrente di forward pari a 20 mA (con valori tipici pari a 230 mCd). [27]

In base a quanto ottenuto le caratteristiche legate al colore dei LED ed alla loro disposizione sul dispositivo da allocare sulla palpebra sono da prendere in considerazione. La

considerazione viene meno però per quanto riguarda la tipologia scelta per i LED con caratteristiche riguardanti soprattutto il consumo e la loro intensità luminosa. Essi cambieranno per soddisfare uno dei tanti obiettivi che si impegna a risolvere SIPEV; ovvero quello di rendere il dispositivo usa e getta. L'impiego di una batteria a bottone per il nostro progetto non sarà ottimale per questa tipologia di LED. Inoltre il BLE consumerà durante la trasmissione e la ricezione dei dati e sarebbe un consumo in più oltre l'elevata corrente impiegata dai LED.

Procediamo dunque con le scelte opportune che contraddistinguono l'efficacia del nostro progetto a livello della stimolazione.

Riassumendo SIPEV si proporrà di risolvere problematiche legate ai cablaggi che sono di intralcio all'esecuzione dell'operazione chirurgica; sviluppando, un sistema portatile con elettronica integrata ad-hoc per uso intraoperatorio, in tempo reale, che limiti i costi ed impieghi una comunicazione wireless con modulo BLE.

5.1 Progettazione del dispositivo

Successivamente alla definizione dell'hardware tratteremo la parte firmware che caratterizza il progetto maggiormente, attualmente infatti scegliamo di impiegare 8 LED, per svariate ragioni che tratteremo di qui in poi e verranno interfacciati tramite 8 porte di I/O presenti sulla nostra board (capitolo 6 Firmware) (Figura 17.b) (Tabella 4).

5.1.1 Scelta dei LED e test fotometrici

I led scelti nel nostro progetto sono led rossi ad alta intensità luminosa della Kingbright APA2107LSECK/J3-PRV (Figura 23) [20] a bassissimo consumo che contano di essere accesi a gruppi di 4 per salvaguardare ancora di più la batteria durante le operazioni di alimentazione del carico e disperdere meglio il calore dal die alla palpebra.



Figura 23 - Hyper red LED. [20]

Essi sono delle seguenti dimensioni: 2,1 mm x 1,0 mm x 0,6 mm e sono degli SMD (Figura 23).

Ognuno di loro consuma solo 2 mA, 10 volte di meno dei LED scelti per il GEPEV, con un angolo di visibilità di 140°. L'intensità luminosa emessa da ognuno di loro va da un minimo di 50 mCd ad un valore tipico di 120 mCd. Sono impiegabili con tensioni che vanno da un minimo di 1,5 V ad un valore tipico degli 1,8 V.

Se si ipotizza la palpebra di superficie pari $2~\rm cm^2$ stimolandola con gli $8~\rm led$ alla massima intensità luminosa che possono generare cioè $960~\rm mCd$ otterremo una luminanza di $480~\rm mCd/cm^2$. [43] [44]

Come evidenziato nella parte anatomica la palpebra attenua l'intensità luminosa che arriverà sulla retina ma riesce comunque a stimolarla. Ma per evitare l'abbagliamento del paziente verificare quanto attenua la palpebra o la durata di rigenerazione della rodopsina non basta. Il fattore abbagliamento dipende anche da quanta intensità luminosa nell'arco dei ms arriva sull'occhio. Spiegheremo e verificheremo questo limite; che è il secondo dei 2 limiti già citati precedentemente nel paragrafo 2.2.2 La fototrasduzione e considerazioni anatomiche aggiuntive; tramite dei test sugli 8 LED impiegati con l'impiego di un luxmetro dell' ATP Instrumentation (Figura 24.a e 24.b).

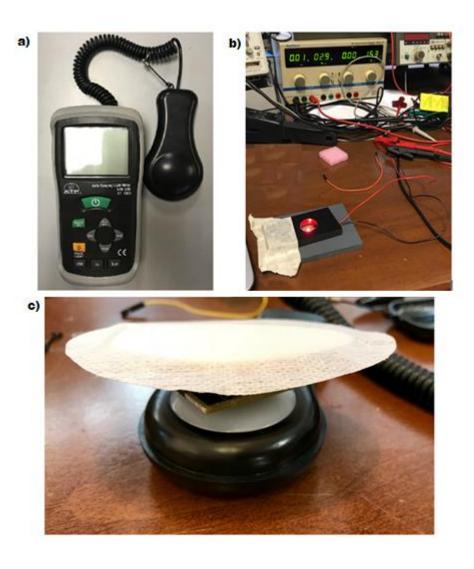


Figura 24 - a) Luxmetro ATP Instrumentation, b) PCB con LED saldato e alimentato con alimentatore, c) sensore mobile del luxmetro che misura l'illuminamento di sfondo.

Il test condotto ad una distanza dal target(occhio) di 1 cm ha evidenziato un valore di illuminamento pari a 514 $\rm lx$ per un singolo LED ovvero 4112 $\rm lx$ per gli 8 LED pari anche a 4112 $\rm cd/m^2$.

Il valore a contatto col target di illuminamento invece è stato di 11600 lx, misurato soltanto per puro interesse, nella stimolazione vera e propria infatti il dispositivo di stimolazione verrà posizionato ad 1 cm circa dalla palpebra del paziente.

Per il controllo dell'abbagliamento, i dispositivi progettati devono rispettare i limiti espressi in termini di indice UGR (Unified Glare Rating) definito dalla norma UNI EN ISO 12464-1 (Tabella 5). [45] [46]

Abbagliamento	UGR
Impercettibile	<18
Percettibile	18÷21
Fastidioso	22÷28
Intollerante	>28

Tabella 5 - Variazione dell'abbagliamento rapportato con l'UGR. [45] [47]

Questo indice può variare tra 10 (nessun abbagliamento) e 30 (abbagliamento fisiologico intollerabile). L'UGR tiene conto della luminanza di sfondo (soffitto, pareti).

Verifichiamo perciò l'UGR con il seguente calcolo [48] [49] (Figura 25) :

$$UGR = 8 * \log_{10} \left[\frac{0.25}{L_b} * \sum \frac{L^2 * \omega}{\rho^2} \right]$$

dove:

- L_b è la luminanza di sfondo [cd/m²];
- L è la luminanza di ogni LED rispetto alla distanza dall'occhio del paziente [cd/m²];
- ω è l'angolo solido tra i LED e l'occhio del paziente [sr] e
- ρ che è l'indice di posizione Guth per ogni LED, che è lo spostamento dalla linea longitudinale e trasversale.

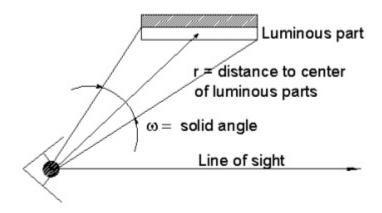


Figura 25- Distanza tra sorgente e occhio per calcolo UGR. [48]

L'indice di posizione di Guth è un valore che si può ricavare sia tramite opportuni grafici che per via esplicita tramite opportuna formula che tiene conto appunto della distanza longitudinale e trasversale tra occhio e sorgente. Questo indice si ricava con la seguente formula [50]:

$$\frac{1}{\rho} = \frac{d^2 * A}{d^2 + 1.5 * d + 4.6} + 0.12 * (1 - A)$$

con:

il coefficiente A pari a $e^{(-0.18*\frac{s^2}{d}+0.01*\frac{s^3}{d})}$;

d che è la distanza longitudinale tra l'occhio e la sorgente e

s che è la distanza trasversale tra l'occhio e la sorgente.

Procediamo ora col calcolo sostituendo e trovando i dati mancanti.

Supponendo che si lavori ad un distanza di 1 cm dalla palpebra, con queste dimensioni ridotte i termini di distanza longitudinale e trasversale dall'occhio alla sorgente si impongono dello stesso valore. Considerando infatti che la distanza tra occhio e sorgente LED è l'ipotenusa del triangolo rettangolo che si viene a creare, l'errore che si commette a imporre questa considerazione è irrilevante (Figura 26). [51]

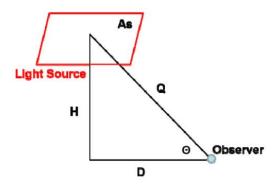


Figura 26 - Distanze tra sorgente luminosa e occhio. [51]

Il valore dell'indice di Guth date queste imposizioni alle distanze si calcola che vale 7,29 adimensionale.

Tramite luxmetro abbiamo misurato il valore dell'illuminamento di un singolo LED che era pari a 514 lx. Questo valore corrisponde ad un valore di flusso luminoso pari a 0,0514 lm (visto che 1 lx è pari a 1*10⁻⁴ lm/cm²). Visto che da datasheet conosciamo il valore dell'intensità luminosa misurata in mCd (per il calcolo servono le Cd) di un singolo LED, calcoliamo l'angolo solido in steradianti (sr) che caratterizza il "cono" di visibilità dell'occhiosorgente (Figura 25) [52]:

$$\omega = \frac{\Phi_v}{L_v}$$

dove

 Φ_v è il valore del flusso luminoso misurato il lumen [lm] e

L_v rispettivamente l'intensità luminosa misurata in candele [Cd].

Otteniamo un valore di angolo solido pari a 0.428 sr.

Ora dalla formula dell'UGR, ci manca solo più sapere il valore della luminanza di sfondo Lb.

Tramite un'ulteriore misura col luxmetro abbiamo calcolato il valore della luminanza di sfondo (Figura 24.c). La luminosità dello sfondo sarà rappresentata dalla luce che arriverà sull'occhio durante la durata dell'operazione chirurgica in sala operatoria, che non è quella derivante dalla luce emessa dai LED. Le lampade scialitiche utilizzate dai chirurghi per illuminare il piano operatorio hanno la caratteristica che utilizzate a distanze di 1 metro dal piano operatorio stesso permettono anche in presenza di ingombri tra la sorgente luminosa e il letto operatorio di avere comunque una visione nitida dei colori, permettendo al chirurgo di operare. [53] La luminosità di sfondo da noi misurata perciò è stata appunto posizionare il sensore del luxmetro, con al di sopra il PCB dei LED e il cerotto oculare e irradiarlo con una lampada posta ad altezza di 1 metro da esso.

Il valore ricavato dell'illuminazione di sfondo è stato di 34 lx.

Calcolando l'UGR per 2 e per 8 LED ricaviamo che per il primo otteniamo un' UGR di 12 mentre per il secondo di 17.

Confrontando questi risultati ottenuti con la Tabella 5, siamo nella soglia di impercettibilità dell'abbagliamento sulla palpebra.

5.1.2 Batteria a bottone

La batteria a bottone è un'opportuna batteria di ridotte dimensioni che è impiegata nella maggior parte dei casi per dispositivi piccoli come ad esempio anche per orologi da polso. La batteria che impiegheremo nel nostro dispositivo la quale è risultata la più opportuna è quella della SONY CR2032 Litio Manganese (Figura 27). [54][66] Essa presenta le seguenti caratteristiche: ha una tensione nominale di 3 V, una capacità di 220 mAh e dimensioni di 20 mm di diametro per 3,2 mm di altezza.

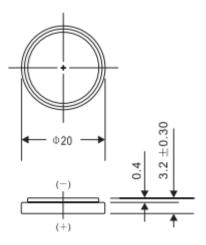


Figura 27 - Batteria a bottone SONY CR2032. [54]

La scelta è ricaduta sulla SONY per questioni legate a dei test svolti su vari carichi e con vari fornitori di altre batterie(Maxwell, Panasonic, etc.). Questi test hanno dimostrato che utilizzando correnti di scarica di 15 mA per poi passare a 30 mA sulle batterie SONY CR2032 essa riduceva del 9% la sua capacità. Questa perdita è stata riscontrata a tutti e 6 i competitor testati; ma la batteria della SONY è quella che si è comportata meglio durante questi test (Figura 28).

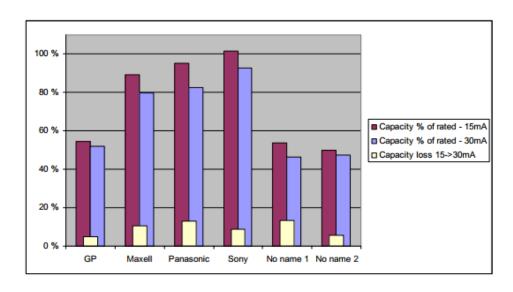


Figura 28 - Comportamento della capacità al variare della corrente. [55]

La batteria dovrà consentirci in ambito ambulatoriale(sala operatoria) una durata di almeno 8 ore. Essa dovrà alimentare il carico che sarà costituito dal chip con BLE e gli 8 hyper red LED. Perciò faremo dei test e faremo opportune considerazioni inserendo anche calcoli e componenti che ci aiuteranno a preservare la carica della batteria per utilizzarla al meglio.

Esistono diverse soluzioni che la Texas Instruments(TI) fornisce per il protocollo BLE. Nella cc2640 può funzionare come: broadcast, osservatore, ricerca dispositivi nelle vicinanze o connessione con i dispositivi ritrovati nelle vicinanze, periferico o centrale. Il dispositivo della TI cc2640 è il chip della gamma di quelli disponibili in commercio che consuma notevolmente di meno in modalità di trasmissione dati (Tabella 6).

	CC2540	CC2540T	CC2541	CC2541-Q1	CC2543	CC2544	CC2640
Max Output Power	+4dBm	+4dBm	0 dBm	0 dBm	+5 dBm	+4 dBm	+5 dBm
TX current (0 dBm)	26 mA	26 mA	18.2 mA	18.2 mA	26 mA	27 mA	6.1 mA
Size (QFN)	6x6	6x6	6x6	6x6	5x5	5x5	4x4, 5x5, 7x7
BLE-Stack Support	Yes	Yes	Yes	Yes	Broadcast only	Broadcast only	Yes
USB 2.0	Yes	Yes	-	-	-	Yes	-
Key Features	USB	High Temp. <125°C	I2C + Lower power	Automotive Qualified	Low cost	Low cost + USB	RTOS + Lowest power

Tabella 6 - Confronto dei chip con BLE della TI. [56]

L'impiego del BLE nella board, significa diversi consumi e diversa modalità di trattamento della CPU e delle periferiche. Le misure dei consumi sono le seguenti (Tabella 7).

Power Mode	Description	Current
Active	System CPU is running	1.45 mA + 31 µA/MHz
Idle	The power domain in which CPU resides is off	550 uA
Standby	The voltage domain in which CPU resides is off	1 uA
Shutdown	Only IOs maintain their operation. All voltage regulators, voltage and power domains are off	0.1 uA

Tabella 7 - Consumi CC2640. [56]

La modalità attiva ovvero quando la CPU sta svolgendo operazioni di trasmissione e ricezione dei dati è quella che consuma di più. Una volta stabilita una connessione i consumi si trovano ad essere compresi tra valori che vanno da un minimo di 2,59 mA per lo Standby ad un massimo di 7,66 mA per la trasmissione dei dati(TX) (Tabella 8).

	State	Time [µs]	Current [mA]	Comment
1	Pre-processing	1165	3.22	RTOS wake-up, radio setup, XTAL guard time
2	Radio preparation	132	3.99	Radio is turned on and in transition to RX
3	Recieve (RX)	129	6.48	The radio receiver listens for a packet from the master. Time depends on connection interval and SCA.
4	RX to TX transition	149	5.49	RX to TX transition
5	Transmit (TX)	90	7.66	The radio transmits a packet to the master on one of the 37 channels. Time is dependent on the amount of transmitted data
6	Post-processing and going to standby	775	2.59	BLE protocol stack processes the received packets and sets up the sleep timer in preparation for the next event. And then going to standby afterwards

Tabella 8 - Analisi nel dettaglio tra pacchetto dati mandato e ricevuto. [56]

Ponendoci nei caso "peggiore" cioè: utilizzando corrente massima tra quelle ricavate dalle prove (Tabella 8) riferite al consumo del BLE, in aggiunta al dispendio energetico degli 8 LED; valutiamo se riusciamo ad alimentare il tutto nelle 8 ore che abbiamo a disposizione, supponendo di alimentare l'intero circuito che si andrà a delineare con la batteria a bottone riportata precedentemente.

5.1.3 Schema calcoli e considerazioni

Il circuito che alimenterà il nostro carico è il seguente (Figura 29).

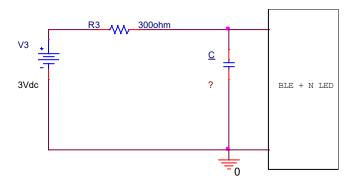


Figura 29 - Circuito qualitativo realizzato in pSpice.

Presenta a monte la batteria a bottone con una resistenza mentre a valle un condensatore col carico. L'aggiunta di un condensatore, fin qui non preso in considerazione, è utilizzata da anni per preservare il budget energetico nei dispositivi che impiegano questa tipologia di batterie. L'impiego dice Jensen [55] è necessario soprattutto per dispositivi che impiegano questa tecnologia di comunicazione col BLE.

Infatti durante i periodi in cui verrà richiesta più corrente, il condensatore agirà come fonte di energia primaria, mentre durante periodi di risparmio energetico, la batteria sarà la fonte di alimentazione principale e ricaricherà il condensatore.

Il confronto tra batteria con e senza l'impiego del condensatore è al di sotto riportato (Figura 30) confronto anche con la Figura 28.

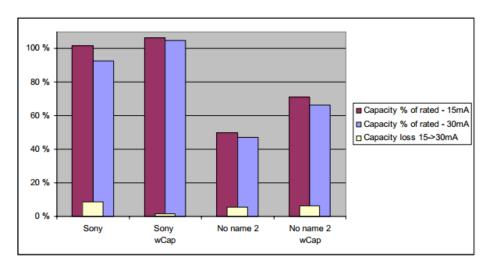


Figura 30 - Confronto batteria con e senza condensatore. [55]

Le batterie Sony con l'impiego del condensatore in parallelo, hanno aumentato la loro capacità effettiva ottenendo valori di capacità per correnti di scarica di 15-30 mA quasi identici.

Calcoliamo quindi il valore del condensatore ottimale per salvaguardare la batteria.

In primis alcune considerazioni:

- durante l'impiego di correnti elevate la batteria utilizzerà la tensione minima a disposizione (Vmin), causando un errore mettendoci però in sicurezza poichè, la batteria fornirà un pò più di energia rispetto al valore che andremo a ricavare;
- la corrente consumata dal circuito durante lo stato in cui il BLE è in modalità sleep è nel range dell' 1 μ A, verrà quindi omesso. [33]

Ci concentreremo quindi sul lato attivo del BLE, ovvero la trasmissione e la ricezione dei dati(TX/RX).

Riprendendo i dati contenuti in Tabella 8 utilizziamo le seguenti formule.

Calcoliamo prima l'energia totale dissipata durante gli stati di carico (Qdissipata).

$$Qdissipata = \sum In * tn$$

Per calcolare il valore della capacità del condensatore, la formula è:

$$C = \frac{\Delta Q}{Vmax - Vmin}$$

con la differenza di energia (ΔQ) data da:

$$\Delta Q = Qdissipata - \frac{Vmin}{Ri} * t_{totale}$$

Vmin è la tensione minima scelta a piacimento per i calcoli. Ri è la resistenza interna massima che il circuito riesce a gestire. Vmax è la tensione a fine vita della batteria che verrà stimata.

Vmax = 2,6 V

Vmin = 2 V

Ri = 270 Ω (siamo partiti a testare da 1 K Ω)

I valori dei consumi del carico sono i seguenti:

	I [mA]	t [µs]
LED (8 per volta)	2 mA	1000000 μs(massimo periodo di ON dei LED è di 1000 ms)
Pre processing	3,22 mA	1165 μs
TX/RX	5,905 mA	500 μs
Postprocessing	2,59 mA	775 μs
Sleep	0,432 mA	20000 μs

Con i precedenti valori e le constatazioni fatte il valore del condensatore (C) ottenuto è molto grande (Tabella 9).

Per valutare la fattibilità del valore del C e sceglierne un valore opportuno a livello commerciale, verifichiamo che il condensatore sia in grado di ricaricarsi durante il tempo di sleep del BLE.

Il tempo di ricarica del condensatore (tricarica) è dato dalla formula:

$$t_{ricarica} = Ri * C * \ln(\frac{Vp - Vmin}{Vp - Vmax})$$

dove Vp è il valore della batteria scarica (End Of Life) pari a 2,7 V preso come valore superiore al valore massimo di tensione della batteria.

Otteniamo un valore del tempo di ricarica pari a 7,51 s. Come accennato prima, dovrà essere del valore adeguato per permettere al condensatore di ricaricarsi ed il tricarica dovrà essere inferiore al tempo di sleep del BLE.

Il tempo di sleep è di 0,02 secondi, (con un consumo trascurabile di 0,432 mA). [57] [33]

Essendo più lungo dello stato di sleep occorre fare delle modifiche; è necessario ridurre Vmax o Ri.

Qdiss [A*s]	Tempo tot[s]	deltaQ [A * s]	C [F]	Vmin [V]	Vmax [V]	Vp [V]	Ri [Ω]	Tempo di sleep[s]	tricarica[5]
0,016008709	1,00244	0,008583227	0,01430537845	2	2,6	2,7	270	0,02	7,515984899
0,016008709	1,00244	0,006895618	0,01149269607	2	2,6	2,7	220	0,02	4,920025862
0,016008709	1,00244	0,004870486	0,00811747721	2	2,6	2,7	180	0,02	2,843258633
0,016008709	1,00244	0,002642842	0,00440473647	2	2,6	2,7	150	0,02	1,285683211
0,016008709	1,00244	-0,000698625	-0,00116437464	2	2,6	2,7	120	0,02	-0,271892211
0,016008709	1,00244	-0,004040091	-0,00673348575	2	2,6	2,7	100	0,02	-1,310275826
0,016008709	1,00244	-0,008441048	-0,01406841258	2	2,6	2,7	82	0,02	-2,244821079
0,016008709	1,00244	-0,013474821	-0,02245803477	2	2,6	2,7	68	0,02	-2,971689609
0,016008709	1,00244	-0,01979272	-0,03298786670	2	2,6	2,7	56	0,02	-3,594719778
0,016008709	1,00244	-0,026648313	-0,04441385454	2	2,6	2,7	47	0,02	-4,061992405
0,016008709	1,00244	-0,035398471	-0,05899745156	2	2,6	2,7	39	0,02	-4,477345851
0,016008709	1,00244	-0,044745231	-0,07457538474	2	2,6	2,7	33	0,02	-4,788860935
0,016008709	1,00244	-0,058246106	-0,09707684377	2	2,6	2,7	27	0,02	-5,10037602
0,016008709	1,00244	-0,075122201	-0,12520366757	2	2,6	2,7	22	0,02	-5,359971923
0,016008709	1,00244	-0,095373514	-0,15895585612	2	2,6	2,7	18	0,02	-5,567648646
0,016008709	1,00244	-0,117649958	-0,19608326353	2	2,6	2,7	15	0,02	-5,723406188
0,016008709	1,00244	-0,151064625	-0,25177437464	2	2,6	2,7	12	0,02	-5,879163731
0,016008709	1,00244	-0,184479291	-0,30746548575	2	2,6	2,7	10	0,02	-5,983002092
0,016008709	1,00244	-0,228488852	-0,38081475404	2	2,6	2,7	8,2	0,02	-6,076456617
0,016008709	1,00244	-0,278826586	-0,46471097595	2	2,6	2,7	6,8	0,02	-6,14914347
0,016008709	1,00244	-0,342005577	-0,57000929527	2	2,6	2,7	5,6	0,02	-6,211446487
0,016008709	1,00244	-0,410561504	-0,68426917369	2	2,6	2,7	4,7	0,02	-6,25817375
0,016008709	1,00244	-0,498063086	-0,83010514387	2	2,6	2,7	3,9	0,02	-6,299709095
0,016008709	1,00244	-0,591530685	-0,98588447565	2	2,6	2,7	3,3	0,02	-6,330860603
0,016008709	1,00244	-0,72653944	-1,21089906600	2	2,6	2,7	2,7	0,02	-6,362012111
0,016008709	1,00244	-0,895300382	-1,49216730393	2	2,6	2,7	2,2	0,02	-6,387971702
0,016008709	1,00244	-1,097813514	-1,82968918945	2	2,6	2,7	1,8	0,02	-6,408739374
0,016008709	1,00244	-1,320577958	-2,20096326353	2	2,6	2,7	1,5	0,02	-6,424315128
0,016008709	1,00244	-1,654724625	-2,75787437464	2	2,6	2,7	1,2	0,02	-6,439890883
0,016008709	1,00244	-1,988871291	-3,31478548575	2	2,6	2,7	1	0,02	-6,450274719

Tabella 9 - Calcoli C e tempo di ricarica C.

I calcoli hanno dimostrato che pur diminuendo il valore della Ri o abbassando la Vmax non si otteneva un valore del tempo di ricarica inferiore o uguale al tempo di sleep del BLE (Tabella 9). Si è puntato soprattutto a fare calcoli e prove diminuendo la Ri facendo rimanere la Vmax invariata proprio per essere certi di dare la giusta alimentazione ai LED (essendo ad alta intensità).

L'impiego di un condensatore perciò è da omettere dal nostro circuito di (Figura 29) proprio perché oltre il BLE abbiamo anche i LED che consumano e vanno a prelevare energia dalla batteria.

La soluzione sarà di attaccare direttamente la batteria ed alimentare il chip cc2640R2F e gli N LED. Facciamo dunque delle prove di durata specifico per il BLE e dei successivi calcoli di durata batteria/carico che ci permettano di capire se la batteria ci possa consentire di durare 8 ore una volta attivato il dispositivo.

5.1.3.1 Calcoli per testare la durata della batteria SONY 2032 sottoposta a carico

Appurato che il dato di targa della batteria è di 220 mAh lavoreremmo a circa la metà della carica dichiarata da datasheet. Questo perché la batteria col tempo non riuscirà più a garantire i 3 V di tensione e supposto di dover alimentare un carico che richiede 15 mA di corrente è come se fossimo in presenza di un carico resistivo di 600 Ω (questo perchè il carico non verrà costantemente alimentato). [66]

Facciamo ulteriori considerazione per calcolare la durata della batteria scelta in presenza di varie tipologie di carico ed il carico totale.

Ogni led che abbiamo a disposizione consuma 2 mA per un totale di 16 mA, poiché noi costantemente andremmo ad alimentarli; solo le porte di I/O saranno alimentate ad intervalli di 4 porte per volta chiudendo di fatto il circuito. Perciò il consumo totale è dovuto dall'apporto dei LED, dalla CPU, dal Bluetooth e dall'oscillatore esterno (TCXO).

Facciamo le nostre considerazioni e varie tipologie di calcolo del consumo.

Da datasheet ricaviamo che le porte di GPIO si comportano come un generatore di corrente a 4 mA. [33] Essendo collegati alle rispettive porte i led saranno sovra-alimentati anche in luminosità.

Prendendo in considerazione un PCB costituito da: chip della TI, BLE, N LED e la batteria stessa; calcoliamone la sua durata.

Partendo dal presupposto che abbiamo 16 mA come consumo totale degli 8 led allora otteniamo la seguente formula:

$$\frac{C_{Batteria}}{DC * I_{totLED}}$$

Il calcolo è influenzato da un fattore DC duty cycle, che indica il "tempo" con la quale i LED rimarranno accesi. Ipotizziamo perciò una durata di ON dei LED rispettivamente del 10 %. La durata in ore è di:

$$\frac{130 \, mAh}{1.6 \, mA} = 81 \, ore$$

Questa durata sarebbe la durata totale del circuito se fosse costituito soltanto dagli 8 LED.

Aggiungiamo perciò anche il consumo del chip, del BLE e dell'oscillatore esterno per calcolare la durata totale dell'intero circuito che abbiamo caratterizzato.

Il chip cc2640 funziona in modalità attiva a 3 V a 3 mA (Figura 31). [33]

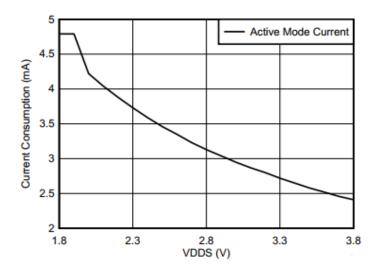


Figura 31 - Consumo MCU TI in modalità attiva. [33]

Il BLE invece consuma in modalità di trasmissione ed in modalità di ricezione dei dati (TX/RX). Da datasheet [33] sappiamo che il consumo in modalità di ricezione RX è di 5,9 mA mentre in modalità di trasmissione TX è di 6,1 mA. Visto che la maggior parte del tempo durante un'operazione chirurgica il dispositivo di stimolazione passerà molto più tempo a trasmettere dati che non a riceverli, quindi ipotizziamo che il consumo sia dovuto totalmente dalla modalità TX. Il BLE funziona a 3 V, a 25 °C.

Con l'utilizzo del MCU con il gruppo di LED si arriverebbe ad una durata di circa 28 ore, mentre utilizzando solo il gruppo di LED con il BLE la durata sarebbe di circa 17 ore.

Con l'aggiunta del TCXO [42] che consuma 4 mA a 3,3 V la durata cambia ulteriormente. Da datasheet la tensione dichiarata di utilizzo dell'oscillatore esterno è 3,3 V ma nel nostro caso però lavorando con una batteria a bottone a 3 V consumeremo di meno rispetto alle 4 mA dichiarate, bensì 3,7 mA.

La durata totale invece di tutte le correnti in gioco è di:

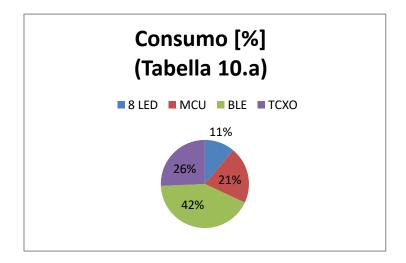
$$\frac{C}{\sum_{i \to \infty}^{i=0} I_i} = \frac{130 \, mAh}{1,6 \, mA + 3 \, mA + 6,1 \, mA + 3,7 \, mA} = 9 \, ore$$

Perciò per le 8 ore di un'ipotetica operazione chirurgica siamo tutelati (Tabella 10.a).

Aumentiamo il DC, ovvero il tempo di ON dei LED e rivediamo la durata che possiamo permetterci con i consumi che ci sono già in gioco e con la batteria scelta.

Come da Tabella 10.b con un DC del 25 % la durata è di 10 ore.

a)	DC [%]	I [mA]	Durata singoli [ore]	Durata totale con gli N componenti[ore]	Consumo [%]
8 LED	10	1,6	81		11,12
MCU	-	3	43		20,83
BLE	-	6,1	21		42,36
TCXO	-	3,7	35	9	25,69



b)	DC [%]	I [mA]	Durata singoli [ore]	Durata totale con gli N componenti[ore]	Consumo [%]
8 LED	25	0,64	203		4,76
MCU	-	3	43		22,33
BLE	ı	6,1	21		45,38
TCXO	-	3,7	35	10	27,53

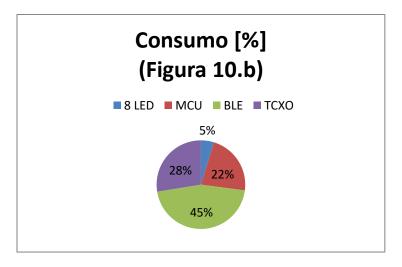


Tabella 10 - Tabella riassuntiva di tutti i componenti con consumi e durata della batteria con N LED, chip e BLE, a) DC = 10% con rispettivo diagramma a torta, b) DC = 25 % con rispettivo diagramma a torta.

Come si può notare dalle tabelle di prima (Tabella 10.a e 10.b), il BLE è quello che consuma di più nel circuito. I LED utilizzandoli ad intermittenza su 2 gruppi disgiunti tendono a contenere meglio i consumi.

In un PCB avremmo questo circuito composto dagli 8 LED, MCU e BLE; l'altro PCB ha la possibilità di omettere la parte relativa al BLE e il TCXO, quindi risparmierebbe notevolmente sulla durata in ore della batteria. Qui sotto riportate in Tabella 11.a e 11.b le prove fatte prima con i valori di DC a 10 % e 25 % della totalità dei due PCB oculari.

a)	DC [%]	I [mA]	Durata singoli [ore]	Durata totale[ore]	Consumo [%]
16 LED	10	3,2	41		16,84
2 MCU	-	6	22		31,57
BLE	-	6,1	21		32,11
TCXO	-	3,7	35	7	19,47

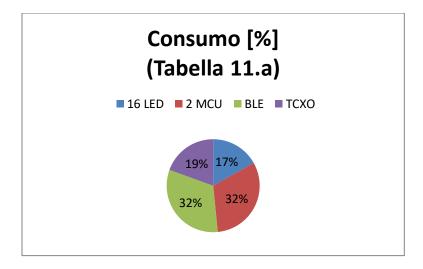


Tabella 11.a - Tabella riassuntiva del consumo considerando la totalità dei LED (DC=10 %) dei 2 PCB oculari, i 2 chip, il modulo BLE e l'oscillatore esterno con consumi e durata della batteria.

b)	DC [%]	I [mA]	Durata singoli [ore]	Durata totale[ore]	Consumo [%]
16 LED	25	1,28	102		7,49
2 MCU	-	6	22		35,12
BLE	-	6,1	21		35,71
TCXO	-	3,7	35	8	21,66

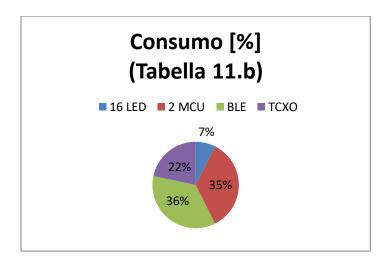


Tabella 11.b - Tabella riassuntiva del consumo considerando la totalità dei LED (DC=25 %) dei 2 PCB oculari, i 2 chip, il modulo BLE e l'oscillatore esterno con consumi e durata della batteria.

In aggiunta al calcolo teorico dei consumi, la Texas Instruments mette a disposizione uno strumento di calcolo per calcolare la potenza del Bluetooth. [58] [59] Questo strumento inoltre offre la possibilità di stimare la durata dell'opportuna batteria a bottone scelta in relazione al BLE e ad un carico customizzato dall'utente. La tabella del programma è stata modificata e nel calcolo delle attività combinate sono stati aggiunti i consumi rispettivamente: dei 16 LED (usati con un DC=25 %), il consumo totale del firmware utilizzato per la comunicazione BLE e il pilotaggio dei LED, il consumo dell'oscillatore esterno e l'alimentazione dei 2 chip per un totale di 15700 μ A (Figura 32). A conferma dei calcoli teorici svolti precedentemente (Tabella 11.b), la conferma della durata della batteria avviene proprio dal programma messo a disposizione della Texas Instrumnets (durata pari ad 8 ore).

Bluetooth low energy consumption

Expected battery life:

Supply Voltage	3
Battery capacity [mAh]	130
RF Configuration	Diff.
Output Power [dBm]	0
Advertising Interval [ms]	100
Advertising data length [#bytes]	7
Connection Interval [ms]	15
Connection data length [#bytes]	0

RX to TX transition Transmit (TX) Transmit (TX) Recieve (RX) RX to TX transition Transmit (TX) TX to RX transition Transmit (TX) TX to RX transition Recieve (RX) Post-Processing		3540,6 652,1 1689,6 370,8
Transmit (TX) TX to RX transition Recieve (RX) RX to TX transition Transmit (TX) TX to RX transition Tx to RX transition Recieve (RX) RX to TX transition Transmit (TX) TX to RX transition Tx to RX transition FX to RX transition FX to RX transition FX to RX transition Recieve (RX) Post-Processing	240 7,04 90 4,12 150 6 320 3,12 240 7,04 90 4,12 150 6 320 3,12 240 7,04 90 4,12 150 6 320 3,12 240 7,04 91 4,12 150 6 320 3,12 240 7,04 90 4,12 150 6 731 2,35	1689,6 370,8
TX to RX transition Recieve (RX) RX to TX transition Transmit (TX) TX to RX transition Recieve (RX) RX to TX transition Transmit (TX) TX to RX transition TX transmit (TX) TX to RX transition Recieve (RX) Post-Processing	90 4,12 150 6 320 3,12 240 7,04 90 4,12 150 6 320 3,12 240 7,04 90 4,12 150 6 731 2,35	370,8
Recieve (RX) RX to TX transition Transmit (TX) TX to RX transition Recieve (RX) RX to TX transition Transmit (TX) TX to RX transition Transmit (TX) TX to RX transition Recieve (RX) Post-Processing	150 6 320 3,12 240 7,04 90 4,12 150 6 320 3,12 240 7,04 90 4,12 150 6 320 3,12 240 7,04 90 4,12 150 8 731 2,35	
RX to TX transition Transmit (TX) TX to RX transition Recieve (RX) RX to TX transition Transmit (TX) TX to RX transition Recieve (RX) Post-Processing	320 3,12 240 7,04 90 4,12 150 6 320 3,12 240 7,04 90 4,12 150 6 731 2,35 t [us] 4067	
Transmit (TX) TX to RX transition Recieve (RX) RX to TX transition Transmit (TX) TX to RX transition Transmit (TX) Post-Processing	240 7,04 90 4,12 150 6 320 3,12 240 7,04 90 4,12 150 6 731 2,35 t [us] 4067	900,0
TX to RX transition Recieve (FIX) RX to TX transition Transmit (TX) TX to RX transition Recieve (FIX) Post-Processing	90 4,12 150 6 320 3,12 240 7,04 90 4,12 150 6 731 2,35 t [us] 4067	998,4
Recieve (RIX) RX to TX transition Transmit (TX) TX to RIX transition Recieve (RIX) Post-Processing	150 6 320 3,12 240 7,04 90 4,12 150 6 731 2,35 t [us] 4067	1689,6
RX to TX transition Transmit (TX) Txto RX transition Recieve (RX) Post-Processing	320 3,12 240 7,04 90 4,12 150 6 731 2,35 t [us] 4067	370,8
Transmit (TK) Tx to RX transition Recieve (RX) Post-Processing	240 7,04 90 4,12 150 6 731 2,35 t [us] 4067 dvertising event [uA]	900,0
TX to RX transition Recieve (FX) Post-Processing	90 4,12 150 6 731 2,35 t [us] 4067	998,4
Recieve (RX) Post-Processing	150 6 731 2,35 t [us] 4067 dvertising event [uA]	1689,6
Post-Processing	731 2,35 t [us] 4067 dvertising event [uA]	370,8
5	t [us] 4067 dvertising event [uA]	900,0
	dvertising event [uA]	1717,9
Total time of advertising event [us] Total time "current [us"mA]		16788.5
Average Current draw during advertising event [uA]	advertising: 169,1 uA	4128,0
4.0		,1 uA 69 Hours

32 Days

LAUNCHXL-CC264	Hardware	LAUNCHXL-CC2640R2
SimplePeripheral, SDK	Software	SimplePeripheral, SDK 1.50
	Revision	2.0
23-Ap	Date	23-Apr-18

State	Time [µs]	Current [mA]	Time * Current
Wake Up & Pre-processing Preparation for Recieve Recieve (RM) RX to TX transition Transmit (TX) Post-Processing (RF)	1105 225 205 78 121 746	3,11 4,18 6,17 4,23 6,99 2,29	3436,6 940,5 1264,9 329,9 845,8
Total time of connection event [us] Total time * ourrent [us*mA] Average Current draw during connection event [uA]	2480		8526,0 3437,9
Average current draw during connection:	569,5	ıA.	
Expected battery life:	228	Hours	

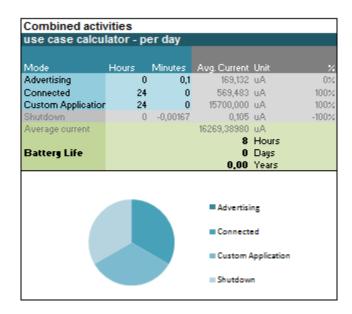


Figura 32 - Programma Excel per calcolare il consumo del BLE e durata batteria. [59]

5.2 Circuito completo

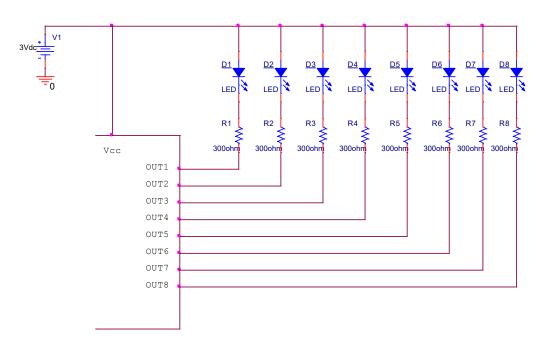


Figura 33 - Schema del circuito nella sua totalità.

Si ottiene il circuito precedente (Figura 33) con i led collegati alle 8 porte di I/O. Inoltre il codice implementato (che tratteremo di seguito nel capitolo 6 Firmware), ci ha permesso di pilotare i pin dal 7 al 14 rappresentati sul circuito dalle porte OUT1 fino alla OUT8. Queste 8 porte sono abilitate a gruppi di 4 e fanno entrare in blinking i rispettivi LED. Così facendo avremmo sulla palpebra una distribuzione dei LED più uniforme disperdendo in maniera più omogenea il calore prodotto dall'intensità luminosa dei LED sulla palpebra stessa. Come visto nel paragrafo 5.1.1 ognuno di questi LED provocherà un'intensità luminosa pari a 50mCd minimo e tipico di 120mCd per un totale di circa 1000mCd su occhio per generare il potenziale evocato visivo.

Perciò si ricreerà sulla palpebra del paziente il seguente segnale intervallato che comporrà nella sua totalità il flash (Figura 34).

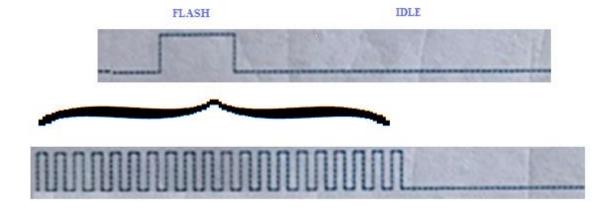


Figura 34 - Flash prodotto dalla sequenza ON OFF dei LED.

Il segnale intervallato che costituirà il flash è realizzato tramite i 2 timer impiegati a livello software, rispettivamente timer A e timer B (Figura 35).



Figura 35 - Timer A e B la loro somma costituisce l'intera durata del flash di stimolazione.

Inoltre ogni LED ha in serie una resistenza che protegge il LED dai picchi di corrente che arrivano dalla batteria. Il valore opportuno di R è pari ad un valore maggiore o uguale a:

$$\frac{\Delta V}{I}$$

Riconduciamo la porta di I/O come un generatore di corrente otteniamo il circuito sottostante (Figura 36).

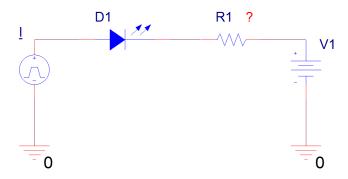


Figura 36 - Circuito per calcolare la R.

Applicando la legge di Kirchhoff in tensione alla maglia(KVL) si ottiene:

$$\frac{V_1 - V_{D1}}{I}$$

Il valore ottenuto di R1 è di 300 Ω . Il valore commerciale che scegliamo della R1 o Ri-esima è perciò di 300 Ω . Questa resistenza sarà impiegata e avrà il medesimo valore su tutte le porte di I/O (OUT1÷OUT8 sul circuito di Figura 33) in serie al rispettivo LED i-esimo.

Un prodotto grezzo iniziale utile per simulare l'effettivo blinking degli N led tramite le 8 porte di I/O è il seguente (Figura 37). Nelle Figura 37.a e 37.b abbiamo una basetta millefori con gli 8 LED saldati nel top mentre nel bottom abbiamo le 8 resistenze da 300 Ω connesse a 8 jumper più 1 che sarà connesso al GND. Nella Figura 37.c il tutto è montato sulla launchpad della TI cc2640R2F nelle rispettive porte di I/O col GND.

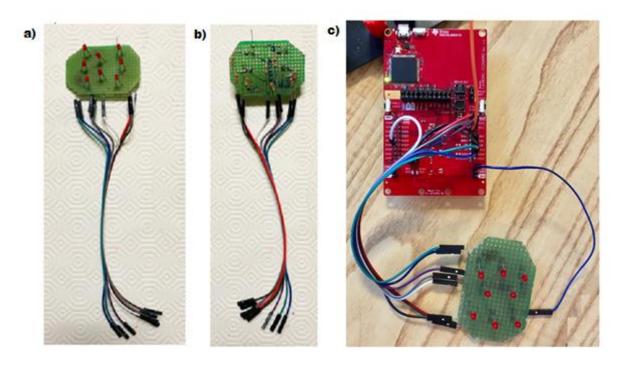


Figura 37 - a) Prodotto grezzo led fronte, b) prodotto grezzo led retro, c) montaggio su board per blinking test.

Capitolo 6

Firmware

Il programma come detto nei capitoli precedenti si articola in una prima serie di task elaborati in Sensor Controller Studio (SCS) ed altri porgrammi dedicati soprattutto a creare la connessione BLE con gli altri dispositivi gestito tramite Code Composer Studio (CCS). [60] [61]

Perciò scindiamo in 2 parti sofware: la caratterizzazione del segnale PWM e la gestione della comunicazione BLE.

Sensor Controller (SC) e Code Composer Studio (CCS) sono 2 programmi messi a disposizone dalla TI e creano interfacce per poter modificare, compilare, debuggare gli sketch di codice sulla nostra Launchpad.

Code Composer Studio (CCS) è un programma che "ingloba" per così dire il Sensor Controller (SC), poiché solo tramite CCS si può interagire con la programmazione e riprogrammazione della launchpad. Diverso utilizzo ha appunto SC che è una partizione dell'ARM Cortex M3 ed è ideale per l'interfacciamento di sensori esterni e per la raccolta di dati analogici e digitali in modo automatico mentre il resto del sistema è in modalità sleep. [62]

6.1 Sensor Controller Studio (SCS)

Ricreiamo un segnale PWM in modo da gestire l'accensione e lo spegniemnto dei LED visti nella Figura 33.

La centrale in veste di client può impostare a proprio piacimento (a discrezione dell'utente) i parametri di durata di accensione dei led, il periodo (tempo di ON + tempo di OFF) e numero di stimolazioni. Il Flow Chart sottostante per rende più capibili i vari step che andremo a compiere per realizzare questa procedura iniziale (Figura 38).

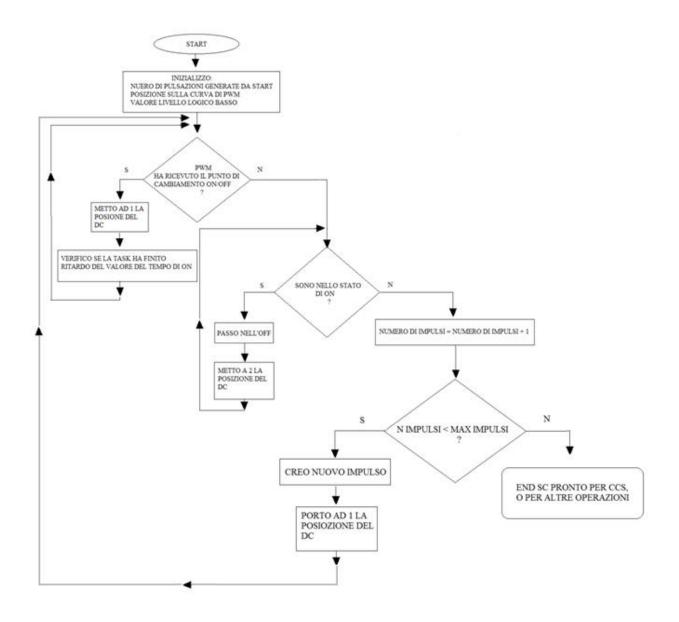


Figura 31 - Flow Chart SC.

Il programma Sensor Controller è costituito da 3 sottoprogrammi che successivamente verranno convertiti, generando un driver, costituito da un insieme di file sorgente C da compilare nell'applicazione CCS (ARM Cortex-M3/M4). Le 3 task che compilate costituiscono i vari file sorgente sono: l'initialization code, l'execution code e il termination code (Figura 39).

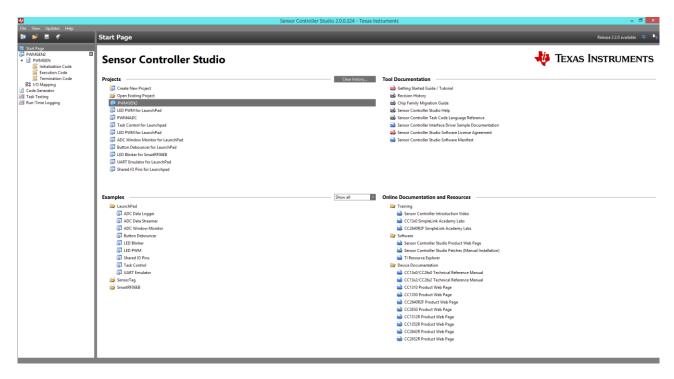


Figura 39 - Schermata iniziale di SC, con in alto a sinistra le 3 task.

L'inizialization code è il cappello introduttivo del nostro programma ed è la parte di inizializzazione del programma mentre l'execution code è il corpo centrale contenente le operazioni sulle variabili del task precedente. Il codice è qui di seguito definito, descritto e commentato.

Si possono definire 4 tipi di dato per caratterizzare il sistema:

- cfg: utilizzato per eseguire la configurazione di runtime dell'attività, prima dell'avvio dell'attività;
- input: utilizzato per passare i dati di input all'attività (ad esempio parametri dinamici per un sensore esterno);
- output: utilizzato per trasferire i dati di output all'applicazione CPU di sistema;
- stato: variabili interne utilizzate per memorizzare lo stato dell'attività tra le iterazioni.

Utilizziamo 2 di queste variabili:cfg e state. Con cfg definiamo:

- dutyval: valore che diamo all'inizio al valore del DC del PWM, varia da 0 a PWM_period;
- maxpulses: numero massimo di impulsi che possono essere generati;
- period: periodo del PWM.

Mentre con state definiamo:

- dutylow: livello logico basso del DC del PWM;
- dutypos: posizione nell'andamento del PWM;
- numpulses: numero di impulsi generati da start.

Dopo aver definito questa struttura di dati iniziali, scriviamo il codice nelle rispettive task.

Initialization Code:

Abbiamo inizializzato il numero di stimoli o impulsi, con inizializzazione della posizione del duty cycle che sarà o quello che contraddistinguerà l'ON o quello che contraddistinguerà l'OFF.

```
state.numpulses = 0;
state.dutypos = 0;
```

Calcolo della lunghezza della parte OFF rispetto al valore dutyval che corrisponde al valore di durata dell'ON.

```
state.dutylow = cfg.period - cfg.dutyval;
```

Pianifica l' esecuzione della task dopo 1 tick.

```
fwScheduleTask(1);
```

Execution Code:

Controlla se è stato ricevuto il punto di cambiamento ON/OFF impostando la GPIO in output a livello logico alto(HIGH) cioè la posizione del mio duty cycle andrà ad 1 generando un'interrupt alla CPU in M3, verificando che la task abbia finito.

```
if (state.dutypos == 0) {
    gpioSetOutput(AUXIO_O_PWM_1);
    state.dutypos = 1;
    fwGenAlertInterrupt();
```

Pianifichiamo la prima esecuzione dopo 1 tick, variabile dutyval settata a 1, cfg sono delle variabili definite prima che la task inizi a "ciclare".

```
fwScheduleTask(cfg.dutyval);
```

Inizia ora la parte di caratterizzazione della parte bassa (LOW), azzerando la GPIO e passando poi nella seconda parte della posizione del duty cycle che equivale alla parte dell'OFF dei led.

```
} else if (state.dutypos == 1) {
    gpioClearOutput(AUXIO_O_PWM_1);
    state.dutypos = 2;
    fwGenAlertInterrupt();
    fwScheduleTask(state.dutylow);
} else {
```

Se no aumenta il numero di impulsi per iterare nel periodo i tempi di ON e i tempi di OFF.

```
state.numpulses += 1;
```

Se la sequenza di istruzioni definite ha ricevuto il numero di impulsi richiesti dall'utente (impostati nella centrale),il codice si fermerà generando l'accensione e lo spegnimento dei led. Inizierà un nuovo ciclo di impulsi, impostando l'uscita a livello logico 1 (HIGH), ritornando alla posizione 1 che corrisponde all'accensione (ON) dei LED.

```
if (state.numpulses < cfg.maxpulses) {
         gpioSetOutput(AUXIO_O_PWM_1);
         state.dutypos = 1;
         fwGenAlertInterrupt();
         fwScheduleTask(cfg.dutyval);
     }
}</pre>
```

Il pin scelto dalla quale uscirà il nostro segnale PWM generato dal SC è il pin DIO6, scelto appositamente dalla lista di pin disponibili per il SC, per poterlo compilare su CCS (Figura 40).



 $Figura\ 40\ -\ SC\ programma\ con\ DIO_6\ selezionato\ per\ generazione\ esterna\ del\ PWM.$

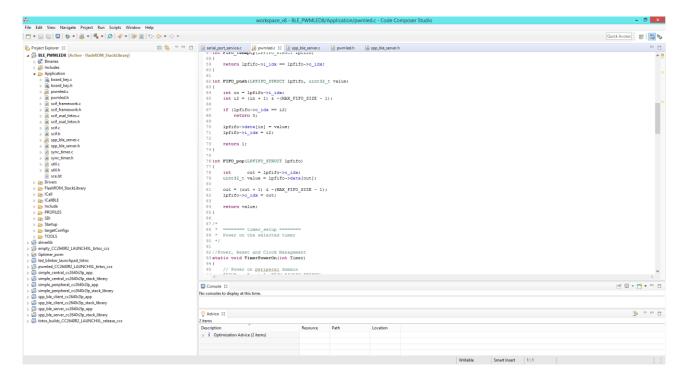
6.2 Code Composer Studio (CCS)

Una volta terminati questi passaggi che impostano il nucleo principale del nostro codice si generano tramite sempre il programma Sensor Controller Studio dei driver che saranno impiegati poi nel codice di Code Composer Studio (CCS) per la gestione delle connessioni bluetooth. Il valore del SC verrà passato sul pin DIO6 dell'MCU e tramite CCS si accederà a questo pin dando la possibilità di attivare e disattivare a piacimento le task elaborate nel SC. Gli altri pin sono liberi di poter essere impiegati per altri scopi.

L'idea ora è proprio quella di attivare a nostra discrezione il programma elaborato col Sensor Controller (Figura 41) che gestisce l'accensione e spegnimento dei led con un PWM customizzabile dall'esterno. Visto che vogliamo ottenere un segnale PWM analogo per gli 8 LED ma che funzioni ad intermittenza a gruppi di 4, dovremo utilizzare 2 timer per ricreare questo intervallo di ON/OFF. Difatti mentre un gruppo di LED sarà ON l'altro gruppo sarà OFF e viceversa, per questioni di distribuire meglio i LED sulla palpebra utilizzandone di più contenendo i consumi disperdendo meglio la temperatura dal die sulla palpebra stessa, come ampiamente descritto nel Capitolo 5 relativo alla parte Hardware dei LED.

Il PWM totale che verrà creato viene denominato FLASH (Figura 34) (Figura 35). Esso verrà creato con l'utilizzo di 2 timer che gestiranno il setup iniziale per metterlo poi in uscita sugli N pin delle porte di I/O. Utilizzeremo 8 pin per accendere rispettivamente 8 hyper red led (paragrafo 5.1.1 Scelta dei LED e test fotometrici). L'obiettivo è appunto di ricavare un segnale ON/OFF molto rapido in frequenza come da Figura 34. Questo rapido cambio di stato dei led non verrà percepito dall'occhio umano che non avrà il tempo di notare un accensione e spegnimento dei led, ma li vedrà perennemente allo stato logico alto(ON) e in un secondo momento basso(OFF). Tutto questo avverrà nello stesso istante, pur essendo i led sfalsati in ordine di tempo dell'ordine dei ms.

Otterremo tutto ciò creando il segnale PWM elaborato dal Sensor Controller col timer A e B elaborati in Code Composer Studio. In uscita otterremmo un segnale di ON/OFF con frequenza maggiore rispetto al PWM del SC che verrà palleggiato sulle porte di I/O a gruppi di 4 accendendo e spegnendo i led. Successivamente manderà il valore interno del timer, corrispondente al fronte di salita del segnale FLASH, alla centrale; questo invio avverrà all'inizio di ciascun fronte di salita caratterizzante il valore "dutyval" impostato dalla centrale stessa.



 ${\bf Figura~41-Code~Composer~Studio~interfaccia.}$

6.2.1 Sketch "pwmled.c"

Procediamo col codice "pwmled.c" e "pwmled.h" di CCS, prima spezzettato e commentato e in un secondo momento lasciato per intero. Qui di seguito vi è un Flow chart per comprendere meglio la descrizione successiva del codice (Figura 42).

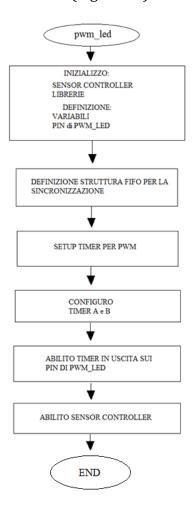


Figura 42 - Flow Chart pwmled.

Le prime righe di questo codice sono di definizione delle librerie standard dell' MCU con l'aggiunta dell'interfaccia principale per il driver SC compilati e generati da SCS "scif.h" e quelle relative al BLE "icall_ble.h" con un richiamo alla funzione definita come "sync_timer.h" (paragrafo 6.2.2 Funzione "sync_timer.c"). Quest'ultima funzione servirà per la sincronizzazione mediante un pin del chip cc2640, il DIO_19, utilizzato per avere la sincronizzazione dei timer di tutti e 3 i dispositivivi, settati esternamente da un pulsante esterno su una breadboard.

BV() invece è una macro vettoriale che tornerà utile strada facendo nel corso del del codice. [63]

```
#include <driverlib/cpu.h>
#include <driverlib/ioc.h>
```

```
#include <driverlib/timer.h>
#include <driverlib/prcm.h>

#include "scif.h"
#include "scif_framework.h"

#include "sync_timer.h"
#include <ti/sysbios/BIOS.h>
#include "icall_ble_api.h"
#include "pwmled.h"

#define BV(x) (1 << (x))</pre>
```

Ora vi è la definizione delle variabili utilizzate andando ad utilizzare il timer denominato GPT0_BASE ed un periodo di 10000 (è la somma dei valori dei 2 timer A e B caratterizzati).

Definizione dei pin dalla porta di in/output (I/O) 7 fino alla porta 14, per definire le porte di GPIO impiegate per mettere in uscita gli 8 led scelti. Sono stati scelti dal 7 al 14 anche per questioni di disposizione sulla CPU. Nel package da 32 pin RGZ (Figura 17.b) [33] infatti i pin sono tra loro vicini e per una realizzazione su PCB saranno facilmente accessibili (Capitolo 7 PCB). La prima parte è di definizione dei pin, mentre la seconda definita LED_MASK consentirà di abilitare e disabilitare nello stesso istante tutte le porte di I/O utilizzate.

```
#define PWM LED 0
                         IOID 7
#define PWM LED 1
                         IOID 8
#define PWM LED 2
                         IOID 9
#define PWM LED 3
                         IOID 10
#define PWM LED 4
                         IOID 11
#define PWM LED 5
                         IOID 12
                         IOID 13
#define PWM LED 6
#define PWM LED 7
                         IOID 14
#define PWM LED MASK
                         ((1 \ll PWM LED 0) | \setminus
                          (1 << PWM LED 1) | \
                          (1 << PWM LED 2) | \
                          (1 << PWM LED 3) | \
                          (1 << PWM LED 4) | \
                          (1 << PWM LED 5) | \
                          (1 << PWM LED 6) | \
                          (1 \ll PWM LED 7))
```

Funzioni utili per il SC rispettivamente per inizializzare il driver SC che utilizza due funzioni di Callback per gestire i due segnali di interrupt TaskAlert e ControlReady del programma elaborato, per acquisirlo in CCS, sono di sotto riportate.

```
static void scCtrlReadyCallback(void);
static void scTaskAlertCallback(void);
```

Ora definiamo una struttura dati di nome FIFO_struct che contiene un array di interi a 32 bit, di lunghezza pari a 8 bit. Dentro questa struttura ci sono 2 indici: i_idx e o_idx. Dichiarando la funzione FIFO_init mette i due indici inizializzati a 0, perciò i_idx punterà alla prima casella e o_idx punterà alla medesima. Il Sensor Controller a questo punto genererà il PWM (ON/OFF) e quando avrò terminato di generare un periodo tramite le variabile Sync_Timer utilizzerò la funzione FIFO_push che metterà come parametri nella struttura dati il valore che assume il Sync_Timer in quell'esatto istante di tempo.

```
FIFO STRUCT fifo time;
int FIFO init(LPFIFO STRUCT lpfifo)
    lpfifo->i idx = 0;
    lpfifo->o idx = 0;
}
int FIFO_isEmpty(LPFIFO STRUCT lpfifo)
    return lpfifo->i_idx == lpfifo->o_idx;
}
int FIFO push(LPFIFO STRUCT lpfifo, uint32 t value)
    int in = lpfifo->i idx;
    int i2 = (in + 1) \frac{1}{6} ~ (MAX FIFO SIZE - 1);
    if (lpfifo->0 idx == i2)
        return 0;
    lpfifo->data[in] = value;
    lpfifo->i idx = i2;
    return 1;
}
```

Dopo aver fatto la push metterò il valore dentro la prima casella ed incrementerò l'i_idx alla posizione successiva. Nel SPP_BLE _SERVER verrà completato il passaggio del valore del timer via BLE alla centrale.

```
int FIFO_pop(LPFIFO_STRUCT lpfifo)
{
    int        out = lpfifo->o_idx;
    uint32_t value = lpfifo->data[out];

    out = (out + 1) & ~(MAX_FIFO_SIZE - 1);
    lpfifo->o_idx = out;

    return value;
}
```

Adesso abilitiamo i timer, con abilitazione della PRCM (paragrafo 4.3.3 GPIO, GPTIMER, IOC, PRCM) che configura la funzione del clock delle periferiche. Attiverà l'alimentazione del dominio delle periferiche all'interno del dominio del microprocessore (PERIPH = GPTO, GPT1, GPT2, GPT3). Sincronizzo ora le impostazioni caricate (Load), e finchè non trovo un dato caricato diverso e mi restituisce che se è stato cambiato e abilita la modalità "Run" della periferica da abilitare. Poi abiliterò il Timer (SleepEnable), consentendo alla periferica di funzionare anche se l'MCU è in modalità sleep.

```
static void TimerPowerOn(int Timer)
{
    PRCMPowerDomainOn(PRCM_DOMAIN_PERIPH);
    PRCMLoadSet();
    while (!PRCMLoadGet());

    PRCMPeripheralRunEnable(Timer);
    PRCMLoadSet();
    while (!PRCMLoadGet());

    PRCMPeripheralSleepEnable(Timer);
}
```

Dopo aver settato il clock, gestisco le porte di GPIO. Le abilito facendo gli stessi processi fatti per le periferiche sulla variabile Timer.

```
static void GPIOPowerOn(void)
{
    PRCMPowerDomainOn(PRCM_DOMAIN_PERIPH);
    while(PRCMPowerDomainStatus(PRCM_DOMAIN_PERIPH) != PRCM_DOMAIN_POWER_ON);
    PRCMPeripheralRunEnable(PRCM_PERIPH_GPIO);
    PRCMLoadSet();
    while ( !PRCMLoadGet() );
}
```

Ora configuro il corpo centrale dello sketch di generazione dei PWM per i due gruppi di led. Configuro il PWM (allocando all'interno delle due strutture create prima) abilitando la periferica TIMERO e la richiamando la struttura di abilitazione delle porte di GPIO. Disabilito entrambi i timer [33], (TIMER_BASE = GPTO_BASE definito nella parte di inizializzazione delle variabili), per azzerare se avessero qualcosa ancora in memoria. Il timer 0 è un timer generico (GPTM), che fornisce due timer a 16 bit. Il GPTM può essere configurato per funzionare come un singolo timer a 32 bit, due timer a 16 bit o come modulo PWM. Noi lo configuriamo come 2 timer da 16 bit ciascuno. Tramite "split" divido i due Timer A e B. [41]

```
void PwmConfig(void)
{
    TimerPowerOn(PRCM_PERIPH_TIMERO);
    GPIOPowerOn();

    TimerDisable(TIMER_BASE, TIMER_BOTH);

TimerConfigure(TIMER_BASE,TIMER_CFG_SPLIT_PAIR|TIMER_CFG_A_PWM|TIMER_CFG_B_PWM);
```

Definendo il Timer B come:

```
TimerLoadSet(TIMER_BASE, TIMER_B, TIMER_PERIOD);
TimerPrescaleSet(TIMER_BASE, TIMER_B, 0);
TimerMatchSet(TIMER_BASE, TIMER_B, 3000);
TimerPrescaleMatchSet(TIMER_BASE, TIMER_B, 0);
TimerStallControl(TIMER_BASE, TIMER_B, false);
TimerLevelControl(TIMER_BASE, GPT CTL TBPWML, true);
```

ed il timer A, come l'opposto del B.

```
TimerLoadSet(TIMER_BASE, TIMER_A, TIMER_PERIOD);
TimerPrescaleSet(TIMER_BASE, TIMER_A, 0);
TimerMatchSet(TIMER_BASE, TIMER_A, 7000);
TimerPrescaleMatchSet(TIMER_BASE, TIMER_A, 0);
TimerStallControl(TIMER_BASE, TIMER_A, false);
TimerLevelControl(TIMER_BASE, GPT_CTL_TAPWML, false);
```

Otterrò questi due segnali A, B con le caratteristiche impostate e richieste (Figura 34) (Figura 35) (Figura 43). Con l'oscilloscopio abbiamo visualizzato quanto impostato dove nella traccia 1 si ha il segnale che esce dal SC dal DIO6 impostato, mentre la traccia 2 è il segnale PWM ricostruito tramite il timer A ed il timer B. La traccia 2 presenta il segnale di ON come "annerito", frutto dei picchi continui di frequenza costruiti dai 2 timer, che verranno intervallati sui 2 gruppi di LED (timer A che si occuperà del gruppo A di LED (N=4); timer B che si occuperà del gruppo B di LED (N=4)).

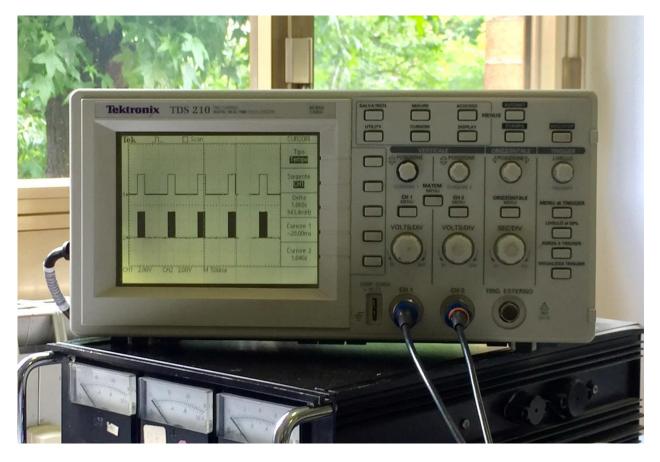


Figura 43 - Oscilosscopio che ritrae le due tracce 1,2 ricavate la prima dal SC la seconda con i 2 timer A e B.

Questi due segnali ricostruiti verranno messi sulle porte (GPIO) in output inizializzate in testa al codice, dove sono connessi i nostri 8 LED.

Poi disabiliterò tutte le porte (LED_MASK) molto comodo ai fini del codice.

```
IOCPortConfigureSet(PWM_LED_0, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_1, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_2, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_3, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_4, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_5, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_6, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_7, IOC_PORT_GPIO, IOC_STD_OUTPUT);
GPIO clearMultiDio(PWM_LED_7, IOC_PORT_GPIO, IOC_STD_OUTPUT);
```

Inizializziamo ora il Sensor Controller dove viene definito il Setup del driver (elaborati da SCS come insieme di programmi sorgente C).

```
scifOsalInit();
scifOsalRegisterCtrlReadyCallback(scCtrlReadyCallback);
scifOsalRegisterTaskAlertCallback(scTaskAlertCallback);
scifInit(&scifDriverSetup);
```

Si imposta la precisione del nostro timer con precisione dei ms (Figura 44). L'oscilloscopio in figura sottostante, rappresenta segnale di periodo 1000 ms con duty cycle di 250 (in base a dei test fatti per testare il codice avendo impostato da centrale: Numero pulsazioni=65000, dutyval=250, periodo=1000).

```
scifStartRtcTicksNow(0x00010000 / 1000);
}
```

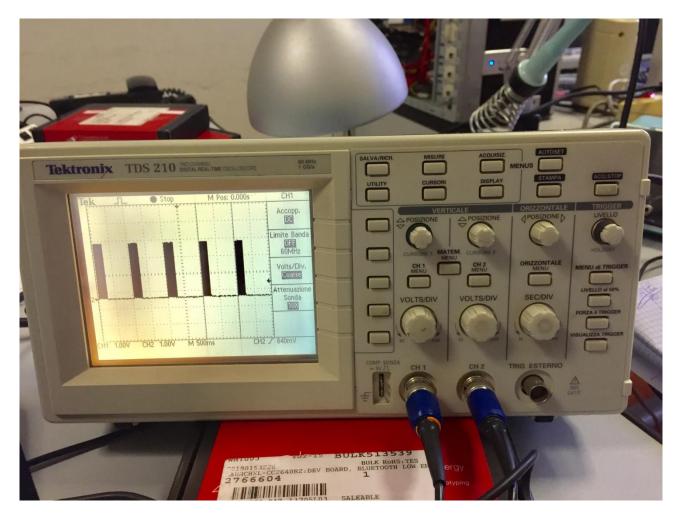


Figura 44 - Precisione verificata con l'oscilloscopio del PWM creato per pilotare i blinking dei LED.

Generiamo allora il PWM coi parametri del SC. La macro BV ora torna utile poichè esegue questa generazione sulla task del SC dove abbiamo definito i parametri che definiscono il nostro segnale (cfg.maxpulses, cfg.dutyval, cfg.period). Questo è il motivo per cui è stata inclusa e la FIFO_init che metterà il valore del timer (TCXO) nella struttura FIFO.

```
void PwmStartGen(void)
{
    FIFO_init(&fifo_time);
    scifStartTasksNbl(BV(SCIF_PWMGEN_TASK_ID));
}
```

Inizio il PWM disabilitando i Timer, e imposto tutte le porte di I/O dal pin IOID 7 al pin IOID 14.

Condizione di stop del segnale PWM, definito come una semplice "Clear" di tutte le porte di I/O quando la somma dei valori di entrambi i timer è uguale al valore del periodo cioè la somma del' ON e dell' OFF. [41]

```
void PwmStop(void)
{
    //Disabilito timer, entrambi
    TimerDisable(TIMER_BASE, TIMER_BOTH);
    HWREG(TIMER_BASE + GPT_O_TAV) = TIMER_PERIOD;
    HWREG(TIMER_BASE + GPT_O_TBV) = TIMER_PERIOD;

IOCPORTCONFIGURESET(PWM_LED_O, IOC_PORT_GPIO, IOC_STD_OUTPUT);
    GPIO_clearMultiDio(PWM_LED_MASK);
}
```

Ciclo continuo del Ready del SC con generazione di Interrupt o Alert per quanto riguarda il SC. Rispettivamente quando sarà nella posizione del dutycycle fronte di salita la funzione FIFO_push metterà nella struttura dati il valore del SyncTimer (paragrafo 6.2.2 Funzione "sync_timer.c"). Perciò se avrò ricevuto il parametro inizierà la struttura PWMStart se non l'avrà ricevuto il set esterno andrà in PWMStop.

Lo stesso risultato ottenuto su oscilloscopio lo abbiamo ottenuto su Launchpad cc2640r2f con i LED che hanno iniziato a blinkare(come detto prima però questo ON/OFF frequenziale è talmente veloce che è impercettibile il cambiamento dei LED ad occhio nudo) (Figura 46).

6.2.2 Funzione "sync_timer.c"

La funzione "sync_timer.c" serve per la gestione del timer di sincronizzazione. Il Flow chart sottostante aiuta a comprendere meglio ciò che di seguito verrà spiegato a livello di codice (Figura 45).

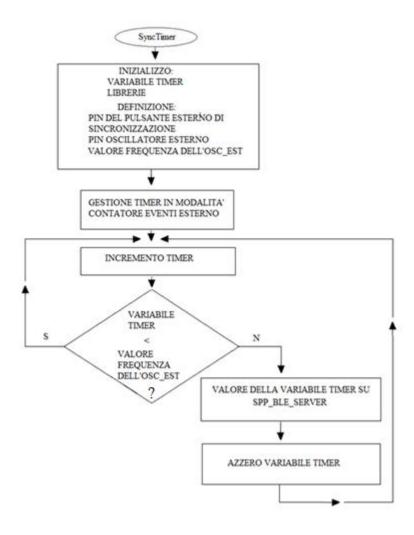


Figura 45 - Flow Chart della funzione sync timer.

L'idea per utilizzare un oscillatore esterno è quella di mettere un timer non in modalità free running ma in modalità di contatore di eventi esterno. Il nostro timer è connesso ad un pin dedicato, al quale noi connettiamo l'oscillatore. L'oscillatore genera una frequenza e noi configuriamo il timer in modo tale che lui si incrementa ogni volta in cui c'è un particolare tipo di evento su questo pin. Per esempio rispettivo al fronte di salita. Questo si incrementa e parallelamente a questo che si incrementa c'è un registro che impostiamo con un valore massimo che questo registro può assumere. Inizierà a incrementarsi e quando arriva al nostro valore si azzera e genera un' interrupt. Questo interrupt lo colleghiamo alla routine di

synctimer. Contiamo un numero di eventi pari alla frequenza del nostro oscillatore diviso la frequenza che vogliamo ottenere.

In questo caso vogliamo ottenere una frequenza di aggiornamento di 1KHz, un contatore che si incrementa a ritmo di 1 ms, quindi se metto un oscillatore esterno da 32 KHz si incrementerà fino a 32 (32KHz/1000)per poi azzerarsi generando un'interrupt. Nel nostro caso l'oscillatore esterno ha una frequenza di lavoro di 10 MHz. [42]

Definizione delle librerie iniziali relative al PIN, il GPTIMER e la "board.h" che contiene tutti i pin della cc2640r2f con la loro precisa funzione. Sul pin IOID_0 è connesso il pulsante per la sincronizzazione dei dispositivi e per far partire il timer e l'IOID_1 è il pin dove è connesso il nostro oscillatore esterno a 10 MHz.

Richiamo al pulsante esterno che crea una callback, e alla definizione di un GPTIMER. (quando viene premuto il pulsante cambia stato e fa partire il timer al pin definito da quella funzione).

```
static void buttonCallbackFxn(PIN_Handle handle, PIN_Id pinId);
static void timerCallback(GPTimerCC26XX_Handle handle, GPTimerCC26XX_IntMask
interruptMask);
```

Dichiarazione delle variabili globali "SyncTimer" e "SyncNum" che potranno essere utilizzate in qualsiasi sketch che faccia parte della cartella dove sto lavorando.

```
volatile unsigned long SyncTimer;
volatile unsigned int SyncNum;
```

Configurazione del pin, definendo la particolarità di avere una R di pull-up interna che sostituisce la resistenza fisica posta in serie al pulsante.

```
static const PIN_Config buttonPinTable[] = {
    BTN_CLEAR | PIN_GPIO_OUTPUT_DIS | PIN_INPUT_EN | PIN_PULLUP | IN_HYSTERESIS,
    PIN_TERMINATE
};
static PIN_Handle pinBtnHandle;
static PIN State pinBtnState;
```

Implementazione delle funzioni di interrupt del timer ad 1 ms, incrementando il SyncTimer; gestendo in secondo tempo l'evento di pressione del pulsante.

```
static void timerCallback (GPTimerCC26XX Handle handle, GPTimerCC26XX IntMask
interruptMask)
    SyncTimer++;
}
static void buttonCallbackFxn(PIN Handle handle, PIN Id pinId)
    if (!PIN getInputValue(pinId)) {
        switch (pinId) {
        case BTN CLEAR:
            SyncTimer = 0;
            SyncNum++;
            break;
        default:
            // Do nothing
            break;
        }
    }
```

Inizializzazione timer del sistema che genera un interrupt utilizzando il GPTimer ogni 1 ms. Inizializzo SyncTimer e SyncNum (variabili che emulano il tick counter) ed inizializzo il GPTIMER ai suoi valori predefiniti. Essi sono: configurazione in larghezza di 16 bit.

```
void SyncTimerStartClear(void)
{
    PIN_setConfig(pinBtnHandle, PIN_BM_IRQ, BTN_CLEAR | PIN_IRQ_NEGEDGE);
}

void SyncTimerStopClear(void)
{
    // Disattiva interrupt
    PIN_setConfig(pinBtnHandle, PIN_BM_IRQ, BTN_CLEAR | PIN_IRQ_DIS);
    SyncNum=0;
}

int SyncTimerInit(void)
{
    GPTimerCC26XX_Handle hTimer;
    GPTimerCC26XX_Params params;
    SyncTimer = 0;
    SyncNum = 0;

    IOCPortConfigureSet(EXT_VCXO_PIN, IOC_PORT_MCU_PORT_EVENT2, IOC_STD_INPUT);
    IOCIOPortPullSet(EXT_VCXO_PIN, IOC_NO_IOPULL);
```

Programmo il timer ad avere un interrupt a 1 ms, avviando il GPTIMER.

```
GPTimerCC26XX_setLoadValue(hTimer, EXT_VCXO_FREQ / 1000);
GPTimerCC26XX_setMatchValue(hTimer, EXT_VCXO_FREQ / 1000);
GPTimerCC26XX_setCaptureEdge(hTimer, GPTimerCC26XX_POS_EDGE);
GPTimerCC26XX_registerInterrupt (hTimer, timerCallback,

GPT_INT_CAPTURE_MATCH);

GPTimerCC26XX_start(hTimer);

pinBtnHandle = PIN_open(&pinBtnState, buttonPinTable);
PIN_registerIntCb(pinBtnHandle, buttonCallbackFxn);
PIN_setConfig(pinBtnHandle, PIN_BM_IRQ, BTN_CLEAR | PIN_IRQ_DIS);

return 0;
}
```

Attivo perciò il SyncTimer nella sua procedura di azzeramento e successiva disattivazione della procedura di azzeramento.

```
void SyncTimerStartClear(void)
{
    PIN_setConfig(pinBtnHandle, PIN_BM_IRQ, BTN_CLEAR | PIN_IRQ_NEGEDGE);
}
void SyncTimerStopClear(void)
{
    PIN_setConfig(pinBtnHandle, PIN_BM_IRQ, BTN_CLEAR | PIN_IRQ_DIS);
    SyncNum=0;
}
```

6.2.2 "SPP BLE Server"

Il dispositivo centrale sarà il "client" del nostro progetto mentre quello di stimolazione è il "server" come anche il dispositivo di lettura.

Il codice che gestisce l'intero sistema di invio dei dati e comunicazione è stato sviluppato all'interno del progetto "SPP_BLE_SERVER" e verranno solo evidenziati i punti chiave inseriti da noi per descriverne il funzionamento e le modifiche apportate. Questo perché come detto nel paragrafo 4.3.2 SPP è un esempio trovato sulla piattaforma della Texas Instruments.

Dopo le varie inizializzazioni principali si ha la task:" SimpleBLEPeripheral_taskFxn", dove viene chiamata la funzione di inizializzazione "SimpleBLEPeripheral_init" che imposta il ruolo di "peripheral" al dispositivo a questo punto si ha il ciclo principale, eseguito continuamente finchè il chip è alimentato ed al suo interno si hanno diversi controlli:

- 1. l'individuazione di una richiesta di riconoscimento da parte della centrale e successivo invio della risposta;
- 2. la verifica dell'avvenimento della procedura di sincronizzazione e la conseguente segnalazione alla centrale tramite l'invio di un pacchetto;
- 3. verifica della struttura che contiene il valore del timer, estraendo il valore del timer tramite la variabile o_idx (paragrafo 6.2.1 Sketch "pwmled.c"), incrementandola in un secondo istante.

Infine si avrà un controllo sugli eventi postati tra cui quello relativo all'ICall, alla gestione della comunicazione Bluetooth, o all'esecuzione di un'operazione periodica.

I parametri iniziali che gestiscono e inizializzano la connessione sono quelli riportati nel paragrafo 4.3 Descrizione e caratteristiche della Launchpad cc2640R2F, in modo da garantire due connessioni simultanee gestite da un'unica centrale e permettere uno scambio di dati tale da non perdere informazioni. Quando un dispositivo svolge il ruolo di "central" deve gestire più di una connessione, con dei limiti sull'impostazione dei parametri di connessione, descritti nella sezione relativa al protocollo del Bluetooth Low Energy. Infatti quando almeno una connessione è già stata creata, per consentire un tempo di elaborazione sufficiente per la ricerca di un nuovo dispositivo, il minimo "Connection Interval" utilizzabile è pari a:

$$12.5 + 5 * n$$
 millisecondi

con "n" che è il numero di connessioni correnti. Per esempio con un numero di connessioni pari a 2, entrambe devono utilizzare un intervallo di connessione minimo pari a 22,5 ms consentendo l'esecuzione della scansione per ristabilire una connessione.

Il massimo numero di connessioni specificato per quando si utilizza Code Composer Studio è di 6 connessioni supportate. [64] Il "Connection Interval" è doppio per il dispositivo di stimolazione (15 ms) (Tabella 12) rispetto che a quello di lettura, dovuto al fatto che più dispositivi comunicano con lo stesso master. Conducendo dei test su valori diversi del "Connection Interval" si è giunti alle seguenti conclusioni, che permettono alle due

connessioni di non entrare in conflitto tra loro, preservandovi la connessione e i dati. Utilizzando per esempio il valore del "Connection Interval" di prima(pari a 22,5 ms), per consentire l'esecuzione della scansione da parte del master per stabilire la connessione dovevamo porre il periodo dell'invio dei dati superiore a questo valore poiché comportava una perdita di campioni, con problemi anche legati alla limitata memoria della centrale. Il dispositivo di lettura perciò deve avere un "Connection Interval" pari a 7,5 ms garantendo la corretta trasmissione del segnale campionato ed il dispositivo di stimolazione invece ha un "Connection Interval" di 15 ms, multiplo di quello del dispositivo di lettura. La Tabella 12 sottostante è riassuntiva dei ragionamenti fatti, gli altri parametri sono rimasti invariati nel codice sorgente.

	Intervalli di tempo		
	CENTRALE-VEP	CENTRALE-LED	
Connection Interval	7,5 ms	7,5 ms 15 ms	
Slave Latency	0	0	
Supervision Timeout	10 s	10 s	
Advertising Interval	100 ms	100 ms	

Tabella 12 - Intervalli di comunicazione tra i dispositivi.

6.2.3 "Serial port service"

Il "serial port service" è utilizzato per permettere l'invio di dati tramite BLE ad alta velocità. Questo file contiene le funzioni che permettono lo scambio bidirezionale di dati tra il server ed il client, mantenendo il codice originale disponibile all'interno del progetto "SPP_BLE_Server", con l'aggiunta di un protocollo di comunicazione che gestisse lo scambio di informazioni tra i vari dispositivi.

Abbiamo svolto dei test utilizzando una serie di caratteri alfanumerici (A, L, ?)per vedere come reagiva la centrale a questo invio. Avvalendoci dei programmi "Putty/Teraterm" (programmi che permettono di creare una sessione remota emulando il terminale a cui ci si vuole riferire), verificavamo le operazioni che stavano compiendo con i 3 dispositivi connessi.

La funzione che gestisce la ricezione dei dati provenienti dalla centrale è il "SerialPortService_WriteAttrCB" all'interno della quale, nel momento in cui viene accettata la richiesta del client di scrivere dei dati, vengono riconosciuti i caratteri ricevuti. Sulla base di questi dati vengono avviate procedure diverse (Tabella 13):

- 1. la centrale dopo aver creato le connessioni, richiede un carattere di riconoscimento a ciascuno dei dispositivi;
- 2. avvenuta la sincronizzazione, i server inviano una stringa per confermare la conclusione dell'operazione;
- 3. la centrale con un carattere di "start" avvia l'acquisizione e la stimolazione, mentre tramite dei pulsanti della Lanchpad stessa invia caratteri per l'interruzione o ripresa del monitoraggio o della stimolazione.

CLIENT = CENTRALE		SERVER = STIMOLAZIONE LED
"?"	-	"L "(LED)
Dopo la sincronizzazione il server è pronto ad acquisire	CANALE DI COMUNICAZIONE	"ОК"
"S"	COMUNICAZIONE	Avvio Stimolazione e SC
"A"	-	Interruzione Stimolazione e SC
"R"		Ripresa Stimolazione e SC

Tabella 13 - Comunicazione tra Client e Server.

Quindi quando il dispositivo di stimolazione abiliterà il canale di comunicazione con la centrale ricevendo i 3 parametri (maxpulses, dutyval, period), vengono portati in accensione gli N LED (Figura 46).

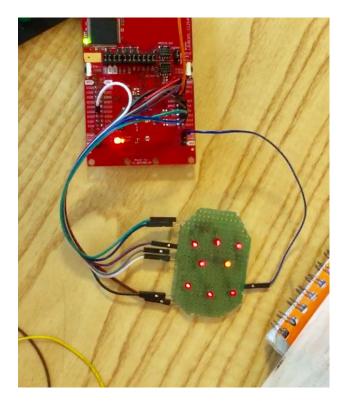


Figura 46 - Blinking LED sulla board della TI.

Capitolo 7

PCB

7.1 Altium Designer

Altium Designer è un software di tipo EDA (Electronic Design Automation) atto al disegno di sistemi elettronici come PCB, FPGA e circuiti integrati. Il sistema dispone di librerie preimpostate fornendo tutti gli elementi per la creazione di librerie custom.

In Altium Designer realizzare un progetto rappresenta un collegamento fra tutti i documenti che costituiscono il progetto stesso. Il progetto sarà costituito da due sotto cartelle definite: "Schematico" e "PCB". Sullo schematico realizzeremo tutto lo schema circuitale attraverso elementi simbolici che definiscono dispositivi e componenti. Una volta definito lo schematico lo assoceremo al PCB dove aggiungeremo il layout del dispositivo e l'immagine 3D.

Tramite Altium realizzeremo 2 PCB sia per l'occhio destro che per l'occhio sinistro, per un totale di 4 PCB (2 per occhio). In un PCB saranno posizionati: il chip della TI, l'antenna per il BLE, l'oscillatore esterno e la batteria. Nell'altro PCB, che sarà posizionato sottostante al precedente, ci saranno gli 8 LED con i vari connettori per alimentarli. I PCB saranno uguali per entrambi gli occhi, sia per quanto riguarda lo schematico sia per quanto riguarda il PCB. La particolarità sarà legata all'utilizzo in un pcb su un occhio del modulo BLE mentre nell'altro non sarà abilitato. Questo perché sarebbe di troppo alimentare anche il BLE, per questioni di consumo, sia perché utilizzarne 2 per compiere le medesime funzioni sarebbe uno spreco soprattutto sui già citati consumi (paragrafo 5.1.2 Batteria a bottone).

Definiamo perciò i vari circuiti partendo col definire lo Schematico e successivamente il PCB.

7.1.1 Schematico

Lo schematico come detto in precedenza è la definizione del circuito che vogliamo realizzare andando a definire i pin della cc2640 che utilizzeremo con i vari componenti connessi ad essi con opportuni collegamenti, l'antenna per il BLE, la batteria di alimentazione, l'oscillatore esterno, i vari connettori per la programmazione del chip e l'interconnessione con altri pcb (Figura 47).

Entriamo nel dettaglio in tutto ciò e definiamo i vari circuiti partendo da quello principale: il componente BLE- CC2640R2F.

Esso è il chip della Texas Instruments con package 5 x 5. Abbiamo scelto il chip con queste dimensioni per andare a guadagnare spazio sul pcb. Il chip della TI è un quad flat no leads, perciò tutta la sua base dovrà essere saldata sul pcb, ed al di sotto di esso tutto ciò che verrà saldato sarà portato direttamente ad un piano di massa per non avere disturbi (Figura 17.b). Già dalla scelta dei componenti che vogliamo impiegare, o dobbiamo per motivi legati alla casa produttrice del dispositivo, siamo soggetti a vincoli progettuali. In questo caso dovremo

adeguarci agli spazi disponibili su pcb considerando il fatto che tutto ciò che salderemo nella parte opposta dove è stato saldato il chip della TI sarà portato a massa.

L'impiego di un'antenna per la comunicazione BLE, anche se scelta più piccola [65], rispetto a quella originariamente utilizzata nella Launchpad della TI adottata all'inizio del progetto, ci impone per questione di spazi all'impiego di 2 pcb su cui posizionare tutti i componenti. Nella parte del PCB (paragrafo 7.1.2 PCB) dichiareremo più approfonditamente questo passaggio.

Nel circuito di Figura 48.a sono definiti tutti i pin che fanno parte dell'alimentazione del chip, gli oscillatori da 24 MHz e da 32 KHz oltre i pin che saranno utilizzati per la programmazione del chip.

BLE-CC2640R2F

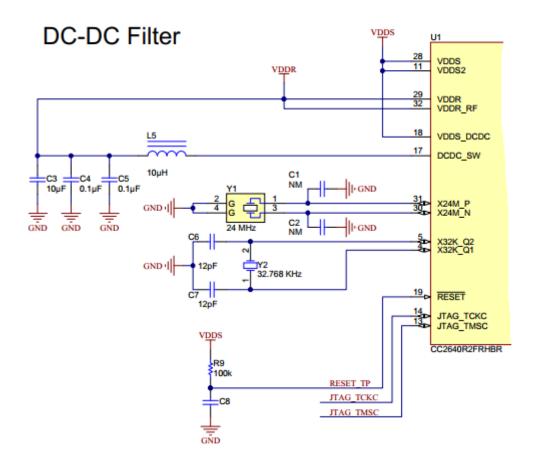


Figura 48.a - Parte sinistra del Chip.

Nella Figura 48.b invece sono dichiarati i pin dove verranno connessi gli 8 LED con rispettive resistenze, oltre ad un led di debug per quando verrà programmato il chip, il filtro passa banda connesso all'antenna del BLE e l'oscillatore esterno (X1).

I pin DIO_0 e DIO_6 sono rispettivamente i pin di sincronizzazione del timer interno gestito esternamente dal pulsante di sincronizzazione e il pin del Sensor Controller. Il DIO_0 battezzato come SYNC, nel circuito sottostante, sarà caratterizzato da una connessione che uscirà esternamente al pulsante di set del clock interno del chip per poter settare il timer interno. Rispettivamente ci saranno 3 cavi che costituiranno il "nasello" tra i due PCB e saranno: cavo di SYNC, cavo di EYE_SC e GND. Il pin DIO_6 chiamato EYE_SC passerà i parametri acquisiti da uno dei 2 PCB col BLE all'altro PCB. Ipotizziamo per un attimo che l'occhio destro abbia il pcb allocato sulla palpebra con il connettore uscente dall'EYE_SC, esso si connetterà al connettore EYE_SC dell'occhio sinistro. Il pcb dell'occhio destro riceverà tramite BLE i parametri esternamente acquisendoli tramite il chip e li manderà sull'altro chip dell'occhio sinistro. In questo modo anche senza l'utilizzo del BLE (per quanto riguarda l'occhio sinistro) avrà ugualmente i parametri dichiarati dall'utente e sarà in grado di pilotare i led oltre ad essere sincronizzato come il pcb dell'occhio destro col timer interno grazie al DIO_0 o SYNC del circuito.

I connettori J10, J11, J12, J13 sono invece stati posizionati per migliorare l'ancoraggio meccanico sul perimetrale del PCB.

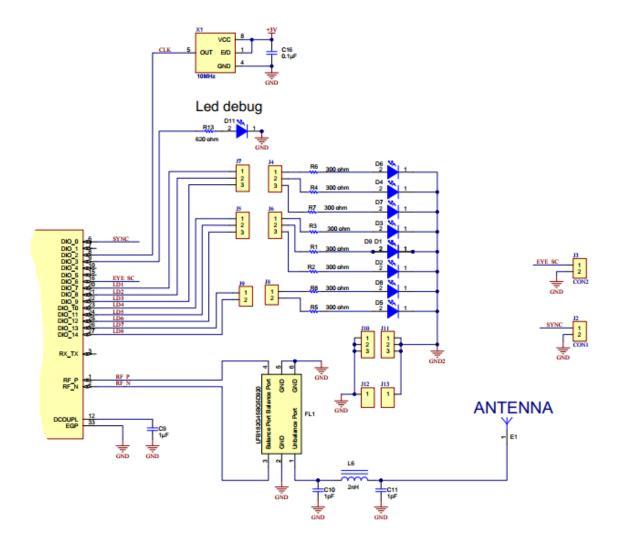


Figura 48.b - Circuito di destra del chip.

L'alimentazione sarà garantita da una batteria a bottone BT1 con condensatori di by-pass in parallelo, non tanto come supplemento di carica quanto piuttosto per eliminare delle eventuali interferenze che si possono creare col segnale elettrico principale (Figura 49) posti su ogni pin di alimentazione.

Battery and decoupling capacitors

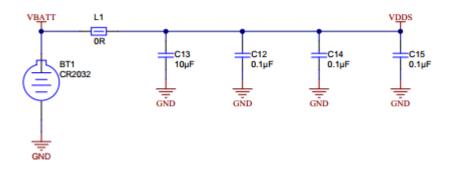


Figura 49 - Batteria a bottone di alimentazione con condensatori.

L'ultima parte di circuito è costituita dal connettore J1 (Figura 50) che è utilizzato per programmare il chip, esso si appoggerà al LED di debug visto in Figura 48.b connesso al DIO 3.

MCU Programming Connector

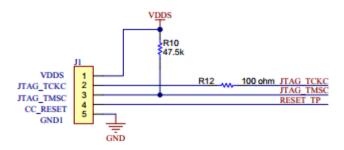


Figura 50 - Connettore per programmare il chip.

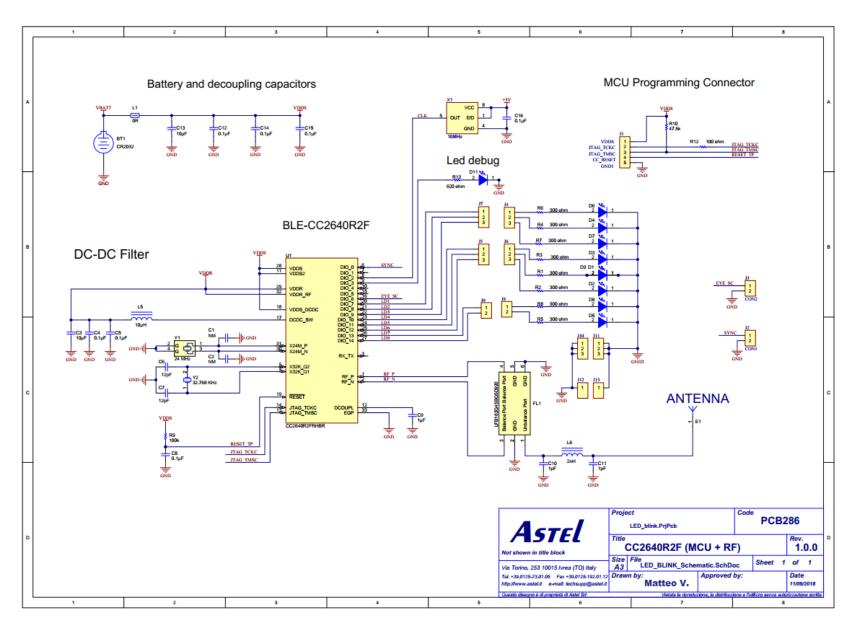


Figura 47 - Schematico completo.

7.1.2 PCB

Una volta realizzato lo schematico ed importarlo sul PCB con i vari componenti, compiliamo e validiamo lo schematico per vedere se tutti i componenti sono stati connessi correttamente, non ci siano doppioni e tutti i componenti vengano presi dalle librerie.

Abilitiamo i cambiamenti, eseguendo il tutto (Execute Changes) (Figura 51) ed una volta validato (Validate Changes) si può procedere col PCB.

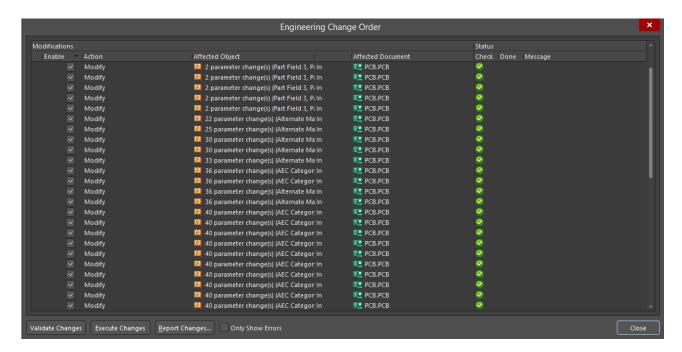


Figura 51 - Lista ECO per importare lo schematico nel PCB.

Prima di procedere alla fase di sbroglio dei componenti che utilizzeremo, è necessario definire la forma che vogliamo dare alla nostra scheda. La forma sarà quella di un cerchio da 3 cm di diametro e spessore 1 mm per ogni PCB per ridurre gli ingombri. Come detto prima saranno 2 questi pcb e andremo ad allocarci sopra i nostri vari componenti. Una volta decisa la sua dimensione si può iniziare a disporre i vari elementi sul piano di lavoro. La disposizione degli elementi ha seguito due principi cardine: disposizione degli elementi vicini in modo tale da minimizzare la lunghezza delle piste, allocare il più vicino possibile gli oscillatori al chip come anche l'antenna. Definito lo sbroglio dei componenti sui PCB li abbiamo collegati. Le piste sono state messe più spesse nei punti di maggiore afflusso di corrente trasportata. Avremo piste sottili per trasporto di segnali e piste leggermente più spesse per quanto riguarda le connessioni di trasporto di energia. Le piste non devono essere intersecate per evitare di formare cortocircuiti. Definendo colori opportuni si sapeva se si stava tracciando una connessione o posizionando un componente o sul top oppure sul bottom del pcb. Una volta terminato il tracciato delle piste, abbiamo disposto il piano di massa, anch'esso con profilo tondeggiante escludendo l'antenna del BLE. Esso serve per schermare i led evitando di sottoporli a disturbi, proteggere gli oscillatori e portare a GND tutti i componenti connessi ad esso; come ad esempio anche la base del chip della TI. Difatti il pad centrale del chip ha sia la funzione di collegamento a GND che di dissipazione dell'energia del case del chip (Capitolo 5 Hardware). Il primo PCB dove abbiamo posizionato la batteria ed il chip della TI è il seguente (Figura 52).

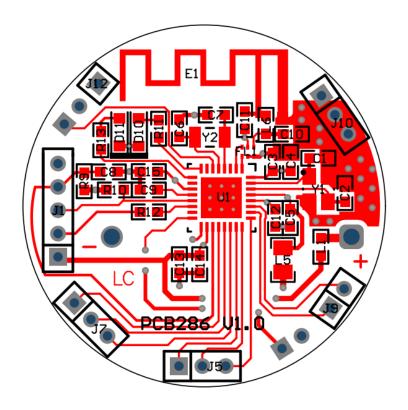


Figura 52 - PCB top layer.

Nel top del PCB abbiamo posizionato il Chip della TI (U1), gli oscillatori (Y1,Y2) e l'antenna (E1).

Sullo stesso layer sono stati posizionati i connettori per la connessione ai LED (J7,J5,J9), i connettori J12 e J10 sono stato posizionati per un più facile alloggiamento del secondo PCB che si andrebbe a montare al di sopra di esso evitando che il pcb si possa muovere e per la connessione a massa. E' stato anche posizionato il J1 per la programmazione del chip, con il rispettivo LED di debug D11 e l'oscillatore esterno.

I connettori J3 e J2 (EYE_SC e SYNC) sono stati per comodità messi sul bottom dello stesso PCB dove è posizionato lo zoccolino della batteria a bottone (BT1) (Figura 53); in un secondo momento sarà posizionata la batteria stessa per alimentare la totalità dei 2 PCB. In una futura vendita tra lo zoccolino e la batteria sarà messa una linguetta in plastica atta ad interrompere l'alimentazione della batteria col PCB. Una volta pronta per la stimolazione, in piena regola con i dispositivi usa e getta, verrà rimossa e il PCB sarà alimentato.

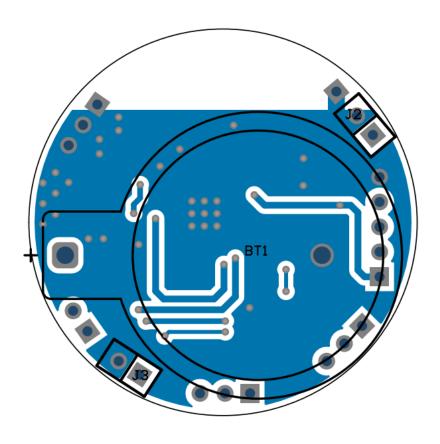


Figura 53 - PCB bottom layer.

L'ulteriore PCB sarà composto dalla rimanente parte del circuito, ovvero i connettori per alimentare i LED (J4,J6,J8), i connettori J11,J13 o connettori di terra e stabilizzazione dei pcb posti nel bottom (Figura 54). Resistenze dei LED e LED stessi (D1, D2, D3, D4, D5, D6, D7, D8) sono stati posti nel top del PCB (Figura 55).

E' stato posizionato anche uno slot (D9) per eseguire delle prove utilizzando un layout diverso, mettendo un LED centralmente al PCB definito (Figura 55).

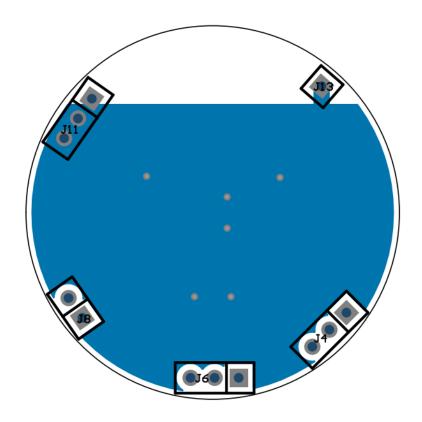


Figura 54- PCB bottom layer.

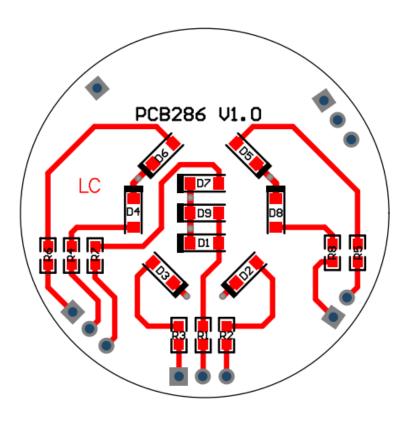


Figura 55 - PCB top layer.

Le sottostanti immagini in 3D rendono meglio l'idea su quanto descritto prima e sul PCB che si è andato a definire (Figura 56)(Figura 57).

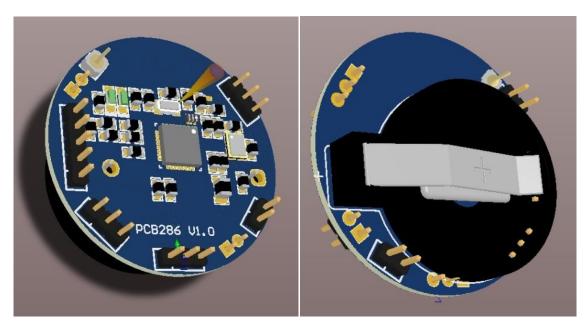


Figura 56 - Top e Bottom del PCB del chip e della batteria.

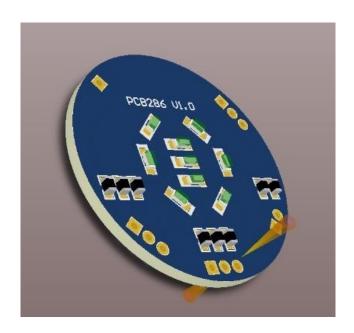


Figura 57 - Top layer PCB con gli N LED.

Uno dei 2 PCB è riportato nella figura 58, dove sono stati saldati i LED per fare dei successivi test.

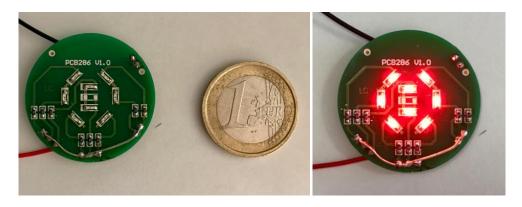


Figura 58 - PCB realizzato con i LED saldati.

Capitolo 8

Prove e test sui dispositivi

Come criterio di sincronizzazione si è deciso all'inizio di optare per misurare la latenza del BLE nella condizione in cui due connessioni simultanee fossero gestite da un solo master. Così facendo se ottenevamo una latenza di invio-ricezione deterministica, si sarebbe potuto conoscere il ritardo tra il sistema di stimolazione (LED) e quello di acquisizione(VEP) nell'esecuzione dei comandi inviati dalla centrale, sfruttandolo per sincronizzare le loro attività. Così facendo quando il dispositivo di stimolazione iniziava a stimolare sulla palpebra al tempo "T", nel medesimo istante in cui si avviava la stimolazione si avviava anche l'acquisizione nello stesso istante. Facendo dei test però si è notato che questa possibilità non si poteva sfruttare, mediante l'ausilio dei risultati ottenuti proprio da queste prove. Infatti, come sottolineato anche in articoli relativi alle caratteristiche del BLE, le latenze di trasmissione ed elaborazione dati si sono dimostrate, nelle condizioni di lavoro descritte, piuttosto casuali e variabili. [37]

Il test è stato condotto collegando tutte e 3 i dispositivi al pulsante di sincronizzazione. Una volta connessi si preme l'interruttore per azionare i timer dei dispositivi e la centrale iniziava la ricerca dei dispositivi e si autentica con essi. Un secondo pulsante presente sulla breadboard consentiva di fare uno "stop" cioè interruzione di connessione tra i dispositivi per poter poi in un secondo momento procedere con altre letture ri-premendo il pulsante di "start" (Figura 59).

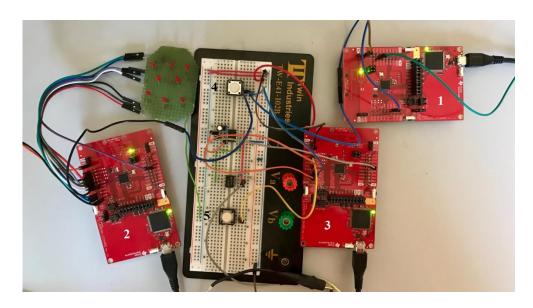


Figura 59 - Dispositivi collegati al circuito di sincronizzazione: 1 dispositivo di lettura VEP, 2 dispositivo di stimolazione, 3 dispositivo centrale, 4 pulsante di sincronizzazione e 5 pulsante di stop.

I valori di latenza sono stati ottenuti osservando sull'oscilloscopio (Figura 60) a quattro canali della Teledyne, con distanze temporali di 5 ms/div. Ognuna delle 3 tracce corrisponde ad un dispositivo.



Figura 60 - Oscillatore Teledyne raffigurante uno degli N test condotti sulle latenze.

Il primo test è stato svolto guardando il collegamento tra il dispositivo di stimolazione ed il dispositivo centrale ossia tra fronte di discesa della centrale con fronte di salita del dispositivo di stimolazione. Osservando poi la latenza tra fronte di salita della centrale con fronte di salita del dispositivo di lettura (Figura 59) (Figura 61) (Tabella 14).

		DISTANZA [ms]			
		LED-CENTRALE	VEP-CENTRALE	LED-VEP	
10	TEST	17 ms	1,5 ms	15 ms	
20	TEST	19,5 ms	4 ms	15 ms	
30	TEST	6 ms	10,5 ms	7 ms	
40	TEST	11,5 ms	4 ms	5,5 ms	
50	TEST	12,5 ms	12,5 ms	7 ms	

Tabella 14 - Connessione in primis tra dispositivo di stimolazione con centrale.

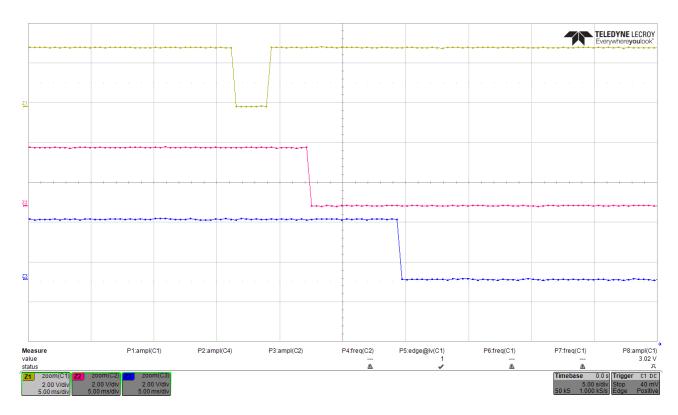


Figura 61 - Screenshoot dell'oscilloscopio rappresentante: traccia gialla centrale, traccia rosa LED, traccia blu VEP.

Il secondo test sulle latenze invece è stato opposto a quello di prima, ovvero prima si andava a vedere la distanza tra fronte di discesa della centrale con fronte di salita del dispositivo di lettura e dopo tra fronte di salita della centrale con fronte di salita del dispositivo di stimolazione (Figura 59) (Figura 62) (Tabella 15).

	DISTANZA [ms]			
	LED-CENTRALE	VEP-CENTRALE	LED-VEP	
6o TEST	12,5 ms	1,5 ms	15 ms	
7o TEST	14,5 ms	12,5 ms	4,5 ms	
8o TEST	15,5 ms	3 ms	15 ms	
9o TEST	11,5 ms	4,5 ms	10 ms	
10o TEST	15 ms	3,5 ms	15 ms	

Tabella 15 - Connessione in primis tra dispositivo di stimolazione con centrale.

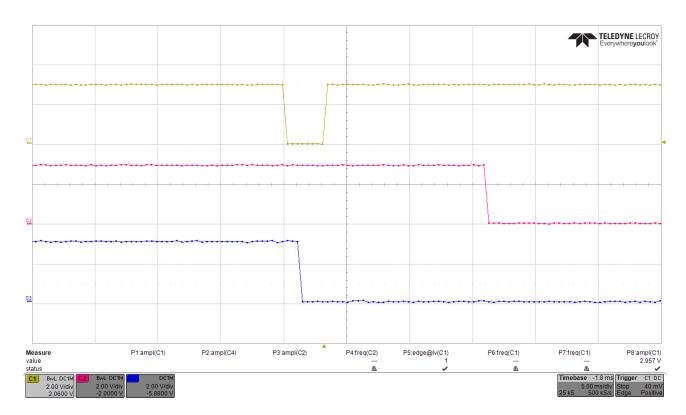


Figura 62 - Screenshoot dell'oscillopio rappresentante: traccia gialla centrale, traccia rosa LED, traccia blu VEP.

Vista la non predittività dei valori di latenza, preso atto di questo e facendo riferimento ai metodi presenti in letteratura, si è optato per la soluzione adottata nel nostro firmware spiegata nel paragrafo 4.4 Idee di sincronizzazione.

La prova successiva alla misura delle latenze tra i dispositivi è stata una prova di funzionamento globale tra i dispositivi.

Il banco di prova è riportato nella Figura 63.a dove sono riportati i 3 dispositivi connessi alla breadboard che ne gestisce la sincronizzazione, l'oscilloscopio ed il generatore di segnale utilizzato per simulare la trasmissione di segnali tra dispositivo di lettura e centrale.

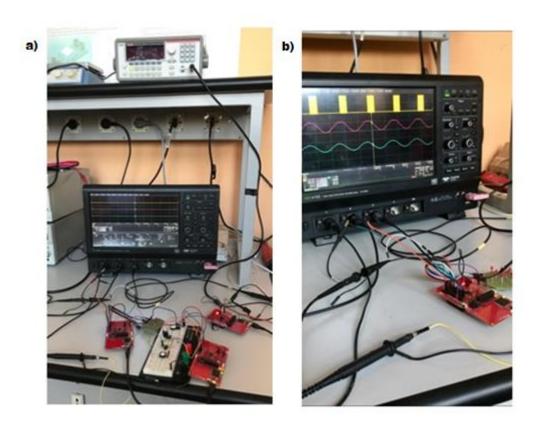


Figura 63 - a) Banco allestito per le prove tra i dispositivi con oscilloscopio, breadboard con i 3 dispositivi e generatore di segnali posizionato sul dispositivo di lettura, b) acquisione lettura oscilloscopio tramite sonda sul dispositivo di stimolazione.

Abbiamo fatto diversi test variando i parametri che la centrale manda al dispositivo di stimolazione rispettivamente: N pulsazioni, Duty cycle e periodo per il dispositivo di stimolazione ed dispositivo di lettura, tramite generatore di segnale un'onda sinusoidale di frequenza pari a 1 Hz. I cambiamenti dei parametri sui 3 dispositivi sono stati osservati tramite Putty e tramite l'oscilloscopio.

In presenza dei seguenti parametri:

N-pulsazioni: 65000;

Duty_c: 250;

Periodo: 1000.

Putty appena attivate le schede rappresentava la schermata sottostante (Figura 64):

```
COM7 - PuTTY
Hello from SPP BLE Client! With Data Length Extension suppor
Auto connecting...
Connected!
Discovering services...
MTU Size: 131
Found Serial Port Service...
Reading UUIDs...Data Char Found...CCCD for Data Char Found..
Notification enabled...
Auto connecting...
Connected!
COM8 - Putty
Hello from SPP BLE Server! With Data Length Extension support
Advertising...
CONNECTED...
MTU Size: 131
COM5 - PuTTY
Hello from SPP BLE Server! With Data Length Extension support!
Advertising...
CONNECTED...
Indirizzo dispositivo connesso:0xB0912269FC24
MTU Size: 131
```

Figura 64 - Schermate Putty preconnessione: centrale, LED, VEP.

Successivamente sviluppata la connessione tra i dispositivi: la centrale manda i parametri di riconoscimento e si autentica con VEP e LED, dopodichè come da Figura 65 riceve i parametri impostati dalla centrale (visti prima) e vengono ricevuti dal dispositivo di stimolazione. La centrale inizierà a ricevere i parametri di tempo che arrivano dal dispositivo di stimolazione e lettura con aggiunta dei dati dell'ADC (Figura 66).

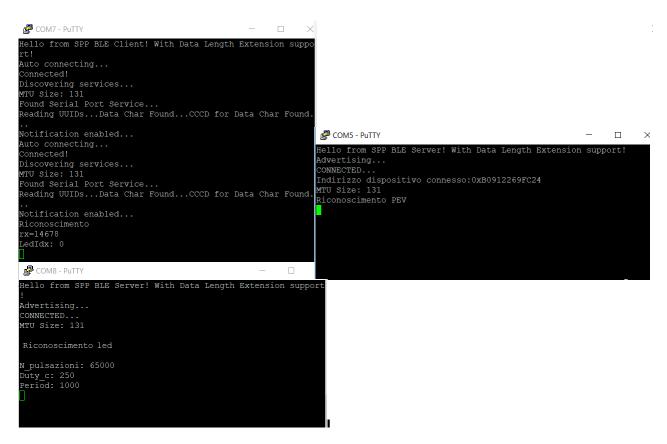


Figura 65 - Schermate putty rappresentanti: centrale, LED e a destra VEP.

```
COM7 - PuTTY
                                                         Hello from SPP BLE Client! With Data Length Extension suppor
Auto connecting...
Connected!
Discovering services...
MTU Size: 131
Found Serial Port Service...
Reading UUIDs...Data Char Found...CCCD for Data Char Found.
Notification enabled...
Auto connecting...
Discovering services...
MTU Size: 131
Found Serial Port Service...
Reading UUIDs...Data Char Found...CCCD for Data Char Found.
Notification enabled...
Riconoscimento
rx=14678
LedIdx: 1
Sincronizzazione
Start inviato
36967
36963
36971
36967
36968
36969
36969
36966
36969
COM8 - PuTTY
Hello from SPP BLE Server! With Data Length Extension support!
Advertising...
CONNECTED...
MTU Size: 131
N_pulsazioni: 65000
___
Duty_c: 250
Period: 1000
                START:
STOP:
COM5 - PuTTY
Hello from SPP BLE Server! With Data Length Extension suppor
Advertising...
CONNECTED...
Indirizzo dispositivo connesso:0xB0912269FC24
MTU Size: 131
Riconoscimento PEV
Sincronizzazione e avvio
START:
Invio
```

Figura 66 - Screen shoot putty: centrale, LED e VEP durante elaborazione dati di trasmissione e ricezione.

Da oscilloscopio si poteva osservare queste 3 tracce (Figura 67) (Figura 63.b).

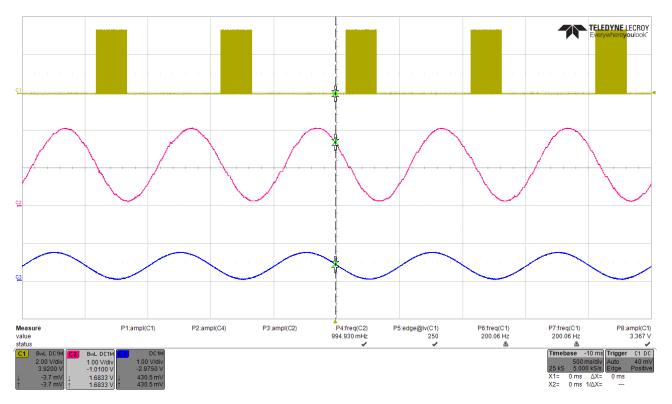


Figura 67 - Schermata oscilloscopio.

Dove la traccia gialla è quella elaborata dal dispositivo di stimolazione rispetto ai 3 parametri che gli sono stati inviati dalla centrale e le altre 2 tracce sono quella del dispositivo di lettura e centrale raffiguranti la trasmissione e la ricezione dei dati dell'ADC.

Cambiando i parametri si osservava il cambiamento della traccia del dispositivo di stimolazione. Il test è stato ancora condotto con queste 2 combinazioni:

N-pulsazioni: 65000;

Duty_c: 500;

Period: 1500;

e

N-pulsazioni: 3

Duty_c: 250

Period: 1000

osservando il cambio drastico della traccia soprattutto per quanto riguardava il numero di pulsazioni o numero di stimoli e valore del duty cycle rispetto ai parametri precedenti (Figura 68) (Figura 69).

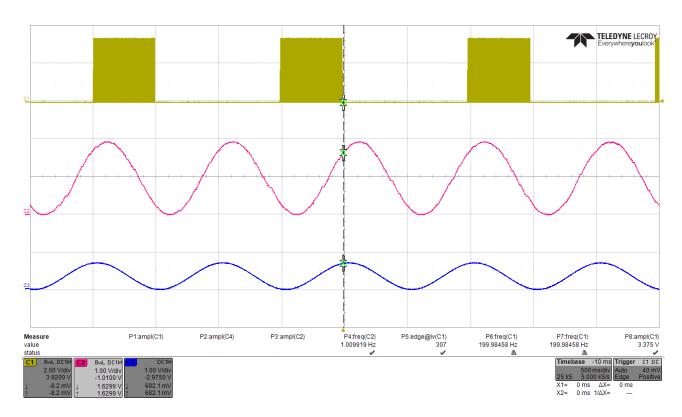


Figura 68 - Schermata oscilloscopio con i parametri: 65000, 500, 1500.

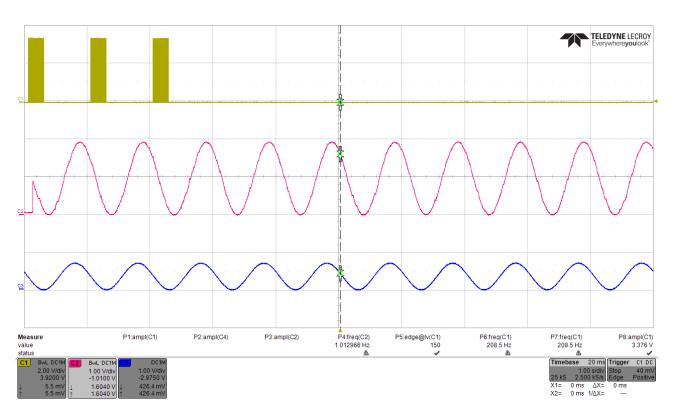


Figura 69 - Schermata oscilloscopio con i parametri: 3, 250, 1000.

Un'ulteriore test è stato condotto per quanto riguardava l'oscillatore esterno, utile per la sincronizzazione dei dispositivi. Esso sarà saldato esternamente su un plug in tutti e 3 i dispositivi (Figura 70.a).

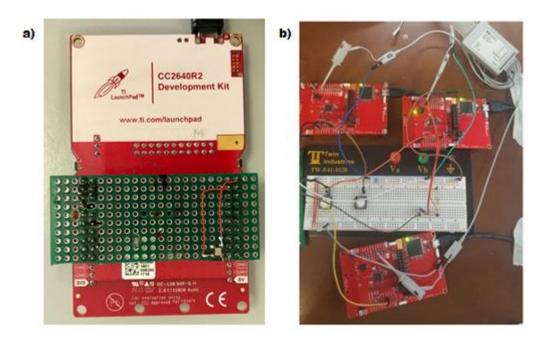


Figura 70 - a) Launchpad con Plug dell'oscillatore esterno, b) dispositivo connessi a 3 sonde per prelievo del segnale ad onda quadra su opportuno pin che simula il clock dell'oscillatore esterno.

Con l'ausilio di un oscilloscopio connesso al pin IOID_1 (Figura 70.b) del chip abbiamo verificato l'allineamento nell'istante di tempo (fronte di salita) dell'onda quadra che veniva messa in uscita al pin mediante dei timer impostati a livello software. Abbiamo verificato così facendo che tutti e 3 i dispositivi erano allineati sul fronte di salita dell'onda quadra, indice che erano tutti sincronizzati nel tempo, pronti per le operazioni di scambio(ricezione e trasmissione) dei dati tramite BLE (Figura 71).

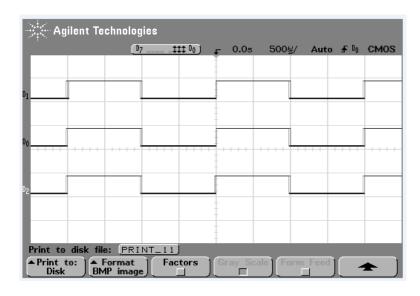


Figura 71 - Oscilloscopio raffigurante le 3 tracce ognuna riferita all'oscillatore presente su dispositivo.

In un ultimo test è stato svolto sul dispositivo ed il PCB realizzato, dove alimentandolo tramite la sola batteria a bottone si sono svolti gli stessi test precedentemente trattati, questa volta però totalmente senza fili. Inizialmente abbiamo fatto in modo che la LaunchpadXL CC2640R2F fosse alimentata dalla sola batteria a bottone. Essa verrà disposta su un plug connessa sui pin della launchpad (Figura 72.b e 72.c) e nel bottom abbiamo sistemato il pcb con gli 8 LED connessi ai rispettivi pin (Figura 72.a).

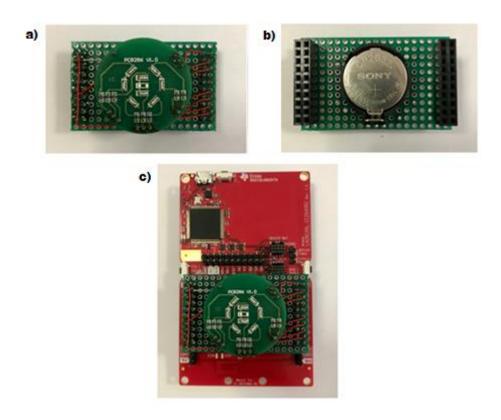


Figura 72 - a) plug con PCB coi LED connessi, b) plug con batteria a bottone, c) plug connesso alla LaunchpadXL CC2640R2F.

Capitolo 9

Conclusioni

L'argomento presentato in questa tesi è stato quello di realizzare un sistema wireless di stimolazione ottica per la generazione dei VEP in modo da far fronte ai limiti e alle problematiche degli attuali dispositivi di stimolazione utilizzati durante gli interventi chirurgici. Partendo infatti dall'analisi dello stato dell'arte del dispositivo di stimolazione si sono analizzati i risultati ottenuti nel progetto GEPEV per continuare il miglioramento del dispositivo di stimolazione. Utilizzando il layout dei LED scelto nel GEPEV, si sono scelti degli opportuni LED a basso consumo oltre ad ulteriori componenti hardware quali batteria a bottone e oscillatore esterno non contemplati nel progetto GEPEV.

Il progetto SIPEV prevede 3 dispositivi: dispositivo di stimolazione, dispositivo di acquisizione e unità centrale sincronizzati tra loro durante lo svolgimento delle attività. Si è cercato di realizzare questo tipo di sistema e si è raggiunto un prototipo preliminare delle tre parti con la collaborazione di Miriam Obertino e Davide Romano.

Inizialmente si è scelto il protocollo opportuno per condurre una comunicazione wireless compatibile in ambito ospedaliero oltre che a basso consumo energetico. La scelta è ricaduta sul dispositivo della Texas Instruments LaunchpadXL-cc2640R2F che contenesse il modulo BLE. Inizialmente lo si è programmato in modo tale che gestisse l'accensione e lo spegnimento dei LED in base ai parametri esterni passati dall'unità centrale. In un secondo step grazie anche agli esempi messi a disposizione dalla Texas Instruments si è gestito il protocollo di comunicazione BLE, con la scelta anche degli opportuni parametri per garantire la connessione simultanea tra i dispositivi. Così facendo si è mantenuta la connessione tra i due server ed il master, garantendo il continuo scambio di dati tra l'unità centrale e i due dispositivi di stimolazione e di acquisizione.

La sfida più importante di questo lavoro di tesi infatti è stata quella di trovare una soluzione per la procedura di sincronizzazione tra i dispositivi della rete. Nell'ambito della stimolazione e monitoraggio dei VEP è necessario garantire l'allineamento tra la stimolazione e l'acquisizione per poi eseguire il post-processing con averaging correttamente. I VEP sono risposte evocate sommerse all'interno del tracciato EEG e solo dopo un'operazione di media su tratti di segnale in cui si ha la risposta con una certa ricorrenza, è possibile estrapolarle ed analizzarle. Questo metodo ha richiesto l'uso di un oscillatore esterno ad alta precisione che comandasse l'incremento di un timer (libero di correre) opportunamente programmato sui tre dispositivi. Nel momento di avvio della procedura di monitoraggio i timer vengono azzerati nello stesso istante tramite un pulsante e questo fa sì che le operazioni svolte dai tre moduli siano caratterizzate dalla stessa base tempi. L'operazione di allineamento della stimolazione con l'acquisizione è resa possibile grazie all'invio dei valori del timer all'inizio di ogni periodo di stimolazione da parte del dispositivo di stimolazione ed in testa ad ogni blocco di campioni acquisiti da parte del dispositivo di acquisizione.

Non è stata direttamente testata questa procedura in quanto le informazioni immagazzinate dalla centrale non potevano essere direttamente processate a causa della scarsa capacità di

memoria e di calcolo del processore. Tuttavia si è verificato l'allineamento dei tre timer tramite oscilloscopio e a seconda dell'oscillatore scelto è garantito nel tempo un jitter massimo differente.

Una modifica finale ha portato all'allocamento del PCB di stimolazione progettato con la LaunchpadXL-CC2640R2F alimentandola solamente con la batteria a bottone. Così facendo si è testato il blinking dei LED in base ai parametri ricevuti da unità centrale solo però a livello monoculare. In un futuro sviluppo del dispositivo l'impiego dei vari test sviluppati nel capitolo 8 saranno anche replicati per entrambi gli occhi con i PCB progettati per completare nella sua totalità la sincronizzazione, ricezione e trasmissione dei dati con l'unità centrale.

Dalla parte di acquisizione del segnale invece, si è individuato un circuito di condizionamento del segnale che ne garantisse un filtraggio ed un'amplificazione adeguata prima dell'acquisizione da parte del microcontrollore. Si è scelto il ModularEEG Analog Board, il quale però ha presentato caratteristiche non ottimali in quanto a filtraggio e attenuazione del rumore di rete. A tale circuito si sono collegati degli elettrodi in Ag/Cl a coppetta per il prelievo del segnale EEG e in un primo momento si sono eseguite misure su volontari senza includere il sistema SIPEV, ma solamente osservando i VEP acquisiti direttamente sull'oscilloscopio in modalità averaging triggerato sul segnale di stimolazione del generatore di segnali. Una volta ottenuti dei prelievi soddisfacenti, si è proceduto col testare l'acquisizione e l'invio del segnale EEG alla centrale e visualizzare il risultato sull'oscilloscopio come per i test precedenti. Dopo una serie di misure effettuate su volontari, i risultati si sono rivelati ripetibili, lasciando però spazio ad un certo margine di miglioramento.

Grazie a questo lavoro si sono poste le basi per proseguire col progetto SIPEV e avvicinarsi maggiormente ad un possibile prodotto finito tenendo conto delle valutazioni fatte e dei risultati ottenuti in questa fase di studio, per poi eseguire dei test con l'intero sistema su volontari sani inizialmente e pazienti in un secondo momento.

9.1 Sviluppi futuri

Nell'ambito del progetto del dispositivo di stimolazione si sono riscontrati problemi a livello hardware poiché dovendo alimentare il PCB progettato con una batteria a bottone, col tempo non riuscirà più a garantire una corretta alimentazione di tutti i componenti elettronici, soprattutto per quanto riguarda l'oscillatore esterno. Perciò una modifica futura che si dovrà fare in questo ambito sarà sicuramente di adottare un porta batterie per due celle da 3 V in modo da portare l'alimentazione a 6 V introducendo un piccolo regolatore di tensione a 3V3. Per limitare maggiormente gli spazi si potrebbero impiegare delle batterie più sottili come le CR2016 o le CR2025 spesse la metà delle CR2032 andando però incontro al grado di affidabilità delle CR2032 sui dispositivi che impiegano la tecnologia BLE.

Oltre a questa modifica hardware il problema riscontrato per la centrale e per il dispositivo di acquisizione è la ridotta memoria presente sul dispositivo della TI scelto. Per questo motivo il processore verrà sostituito con un altro più performante dotato comunque sempre di comunicazione wireless BLE.

Capitolo 10

Appendice

Il codice del paragrafo 6.1 Sensor Controller Studio (SCS) è il seguente nella sua totalità, diviso nelle 2 task precedentemente descritte.

```
//Initialization code:
state.numpulses = 0;
state.dutypos = 0;
state.dutylow = cfg.period - cfg.dutyval;
fwScheduleTask(1);
//Execution code:
if (state.dutypos == 0) {
    gpioSetOutput(AUXIO O PWM 1);
    state.dutypos = 1;
    fwGenAlertInterrupt();
    fwScheduleTask(cfg.dutyval);
} else if (state.dutypos == 1) {
    gpioClearOutput(AUXIO_O_PWM_1);
    state.dutypos = 2;
    fwGenAlertInterrupt();
    fwScheduleTask(state.dutylow);
} else {
    state.numpulses += 1;
    if (state.numpulses < cfg.maxpulses) {</pre>
        gpioSetOutput(AUXIO O PWM 1);
        state.dutypos = 1;
        fwGenAlertInterrupt();
        fwScheduleTask(cfg.dutyval);
    }
```

Programma elaborato in CCS per lo sketch "pwmled.c" che gestisce il blinking dei LED creando un PWM.

```
#include <driverlib/cpu.h>
#include <driverlib/ioc.h>
#include <driverlib/timer.h>
#include <driverlib/prcm.h>
#include "scif.h"
#include "scif framework.h"
#include "sync timer.h"
#include <ti/sysbios/BIOS.h>
#include "icall_ble_api.h"
#include "pwmled.h"
#define BV(x) (1 << (x))
                         GPTO BASE
#define TIMER BASE
#define TIMER PERIOD
                         10000
                     IOID_15
IOID_8
IOID_9
IOID_10
IOID_11
IOID_12
IOID_13
IOID_14
#define PWM LED 0
#define PWM_LED_1
#define PWM_LED_2
#define PWM_LED_3
#define PWM_LED_4
#define PWM_LED_5
#define PWM_LED_6
#define PWM LED 7
                         ((1 << PWM LED 0) | \setminus
#define PWM LED MASK
                           (1 << PWM LED 1) | \
                           (1 << PWM LED 2) | \
                           (1 << PWM LED 3) | \
                           (1 << PWM LED 4) | \
                           (1 << PWM LED 5) | \
                           (1 << PWM_LED_6) | \
                           (1 \ll PWM LED 7))
static void scCtrlReadyCallback(void);
static void scTaskAlertCallback(void);
FIFO STRUCT fifo time;
int FIFO_init(LPFIFO STRUCT lpfifo)
    lpfifo->i idx = 0;
    lpfifo->o idx = 0;
}
int FIFO_isEmpty(LPFIFO STRUCT lpfifo)
    return lpfifo->i idx == lpfifo->o idx;
}
int FIFO push(LPFIFO STRUCT lpfifo, uint32 t value)
    int in = lpfifo->i idx;
    int i2 = (in + 1) & ~ (MAX FIFO SIZE - 1);
```

```
if (lpfifo->o idx == i2)
        return 0;
    lpfifo->data[in] = value;
    lpfifo->i idx = i2;
   return 1;
}
int FIFO_pop(LPFIFO_STRUCT lpfifo)
           out = lpfifo->o idx;
    int
    uint32 t value = lpfifo->data[out];
    out = (out + 1) & \sim (MAX FIFO SIZE - 1);
    lpfifo->o idx = out;
   return value;
}
   ====== timer setup ======
   Power on the selected timer
static void TimerPowerOn(int Timer)
    PRCMPowerDomainOn(PRCM DOMAIN PERIPH);
    PRCMLoadSet();
   while ( !PRCMLoadGet() );
    PRCMPeripheralRunEnable(Timer);
    PRCMLoadSet();
   while ( !PRCMLoadGet() );
   PRCMPeripheralSleepEnable(Timer);
}
static void GPIOPowerOn(void)
    PRCMPowerDomainOn (PRCM DOMAIN PERIPH);
    while (PRCMPowerDomainStatus (PRCM DOMAIN PERIPH) != PRCM DOMAIN POWER ON);
    PRCMPeripheralRunEnable (PRCM PERIPH GPIO);
   PRCMLoadSet();
    while ( !PRCMLoadGet() );
}
void PwmConfig(void)
    TimerPowerOn(PRCM PERIPH TIMERO);
    GPIOPowerOn();
    TimerDisable(TIMER BASE, TIMER BOTH);
```

```
TimerConfigure (TIMER BASE,
TIMER CFG SPLIT PAIR | TIMER CFG A PWM | TIMER CFG B PWM);
     TimerLoadSet(TIMER BASE, TIMER B, TIMER PERIOD);
     TimerPrescaleSet(TIMER BASE, TIMER B, 0);
     TimerMatchSet(TIMER BASE, TIMER B, 3000);
     TimerPrescaleMatchSet(TIMER BASE, TIMER B, 0);
     TimerStallControl(TIMER BASE, TIMER B, false);
     TimerLevelControl(TIMER BASE, GPT CTL TBPWML, true);
     TimerLoadSet(TIMER BASE, TIMER A, TIMER PERIOD);
     TimerPrescaleSet(TIMER BASE, TIMER A, 0);
     TimerMatchSet(TIMER BASE, TIMER A, 7000);
     TimerPrescaleMatchSet(TIMER BASE, TIMER A, 0);
     TimerStallControl(TIMER BASE, TIMER A, false);
     TimerLevelControl(TIMER BASE, GPT CTL TAPWML, false);
    IOCPortConfigureSet(PWM_LED_0, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_1, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_2, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_3, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_4, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_5, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_5, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_7, IOC_PORT_GPIO, IOC_STD_OUTPUT);
     GPIO clearMultiDio(PWM LED MASK);
     // Initialize the sensor controller
     scifOsalInit();
     // Register alert callback
     scifOsalRegisterCtrlReadyCallback(scCtrlReadyCallback);
     scifOsalRegisterTaskAlertCallback(scTaskAlertCallback);
     scifInit(&scifDriverSetup);
     scifStartRtcTicksNow(0x00010000 / 1000);
}
void PwmStartGen(void)
     FIFO init(&fifo time);
     scifStartTasksNbl(BV(SCIF PWMGEN TASK ID));
void PwmStart(void)
{
     bool bIrq = CPUcpsid();
     TimerEnable(TIMER BASE, TIMER BOTH);
     IOCPortConfigureSet(PWM LED 0, IOC PORT MCU PORT EVENTO, IOC STD OUTPUT);
     IOCPortConfigureSet(PWM_LED_1, IOC_PORT_MCU_PORT_EVENTO, IOC_STD_OUTPUT);
     IOCPortConfigureSet(PWM_LED_2, IOC_PORT_MCU_PORT_EVENTO, IOC_STD_OUTPUT);
     IOCPortConfigureSet (PWM LED 3, IOC PORT MCU PORT EVENTO, IOC STD OUTPUT);
```

```
IOCPortConfigureSet(PWM LED 4, IOC PORT MCU PORT EVENT1, IOC STD OUTPUT);
      IOCPortConfigureSet(PWM LED 5, IOC PORT MCU PORT EVENT1, IOC STD OUTPUT);
      IOCPortConfigureSet (PWM LED 6, IOC PORT MCU PORT EVENT1, IOC STD OUTPUT);
      IOCPortConfigureSet(PWM LED 7, IOC PORT MCU PORT EVENT1, IOC STD OUTPUT);
      if (!bIrq)
            CPUcpsie();
}
void PwmStop(void)
      //Disabilito timer, entrambi
      TimerDisable (TIMER BASE, TIMER BOTH);
      HWREG(TIMER BASE + GPT O TAV) = TIMER PERIOD;
      HWREG(TIMER BASE + GPT O TBV) = TIMER PERIOD;
     IOCPortConfigureSet(PWM_LED_0, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_1, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_2, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_3, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_4, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_5, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_6, IOC_PORT_GPIO, IOC_STD_OUTPUT);
IOCPortConfigureSet(PWM_LED_7, IOC_PORT_GPIO, IOC_STD_OUTPUT);
      GPIO clearMultiDio(PWM LED MASK);
}
static void scCtrlReadyCallback(void)
static void scTaskAlertCallback(void)
{
      scifClearAlertIntSource();
      if (scifTaskData.pwmgen.state.dutypos == 1)
            FIFO push (&fifo time, SyncTimer);
            PwmStart();
      } else {
            PwmStop();
      scifAckAlertEvents();
```

```
Sketch su CCS della funzione "sync timer.c".
```

```
#include <ti/drivers/PIN.h>
#include <ti/drivers/timer/GPTimerCC26XX.h>
#include <ti/sysbios/BIOS.h>
#include <xdc/runtime/Types.h>
#include "board.h"
#define BTN CLEAR
                        IOID 0
#define EXT VCXO PIN
                       IOID 1
#define EXT VCXO FREQ
                      10000000UL
static void buttonCallbackFxn(PIN Handle handle, PIN Id pinId);
static void timerCallback (GPTimerCC26XX Handle handle, GPTimerCC26XX IntMask
interruptMask);
volatile unsigned long SyncTimer;
volatile unsigned int SyncNum;
static const PIN Config buttonPinTable[] = {
    BTN_CLEAR | PIN_GPIO_OUTPUT_DIS | PIN_INPUT_EN | PIN_PULLUP |
PIN HYSTERESIS,
    PIN TERMINATE
};
static PIN Handle pinBtnHandle;
static PIN State pinBtnState;
static void timerCallback (GPTimerCC26XX Handle handle, GPTimerCC26XX IntMask
interruptMask)
    SyncTimer++;
}
static void buttonCallbackFxn(PIN Handle handle, PIN Id pinId)
        if (!PIN getInputValue(pinId)) {
        switch (pinId) {
        case BTN CLEAR:
            SyncTimer = 0;
            SyncNum++;
            break:
        default:
            // Do nothing
            break;
        }
    }
}
int SyncTimerInit(void)
    GPTimerCC26XX Handle hTimer;
    GPTimerCC26XX Params params;
```

```
SyncTimer = 0;
    SyncNum
             = 0;
    IOCPortConfigureSet(EXT_VCXO_PIN, IOC_PORT_MCU_PORT_EVENT2, IOC_STD_INPUT);
    IOCIOPortPullSet(EXT_VCXO_PIN, IOC_NO_IOPULL);
    GPTimerCC26XX_Params_init(&params);
                          = GPT CONFIG 16BIT;
    params.width
                          = GPT MODE EDGE_COUNT_UP;
    params.mode
    params.debugStallMode = GPTimerCC26XX DEBUG STALL OFF;
    hTimer = GPTimerCC26XX_open(CC2640R2 LAUNCHXL GPTIMER1A, &params);
    if(hTimer == NULL)
        return 1;
    GPTimerCC26XX setLoadValue(hTimer, EXT VCXO FREQ / 1000);
    GPTimerCC26XX setMatchValue(hTimer, EXT VCXO FREQ / 1000);
    GPTimerCC26XX_setCaptureEdge(hTimer, GPTimerCC26XX_POS_EDGE);
GPTimerCC26XX_registerInterrupt(hTimer, timerCallback,
GPT INT CAPTURE MATCH);
    GPTimerCC26XX start(hTimer);
    pinBtnHandle = PIN open(&pinBtnState, buttonPinTable);
    PIN registerIntCb (pinBtnHandle, buttonCallbackFxn);
    PIN setConfig (pinBtnHandle, PIN BM IRQ, BTN CLEAR | PIN IRQ DIS);
    return 0;
}
void SyncTimerStartClear(void)
    PIN setConfig (pinBtnHandle, PIN BM IRQ, BTN CLEAR | PIN IRQ NEGEDGE);
}
void SyncTimerStopClear(void)
{
    PIN setConfig (pinBtnHandle, PIN BM IRQ, BTN CLEAR | PIN IRQ DIS);
    SyncNum=0;
}
```

Bibliografia:

- [1] Miriam Obertino. "Sistema integrato e portatile per il monitoraggio di potenziali evocati visivi: Dispositivo di acquisizione." Tesi di Laurea Magistrale, Politecnico di Torino, Torino, 2018.
- [2] Davide Romano. "Sistema integrato e portatile per il monitoraggio di potenziali evocati visivi: Unità Centrale per la sincronizzazione di stimolo e lettura." Tesi di Laurea Magistrale, Politecnico di Torino, Torino, 2018.
- [3] http://docente.unife.it/enrico.granieri/materiale-didattico/
- [4] Creel, D.J. (anno) Visual Evoked Potentials. Disponibileall'indirizzo http://webvision.instead technologies.com/part-xii-the-electroretinogram-erg/11-3-visually-evoked-potentials-by-donnell-j-creel/
- [5] Parisi, V., Coppola, G.(s.d.) Elettrofisiologia Oculare, FGeditore.
- [6] https://www.acuvue.it/salute-occhi/anatomia-occhio
- [7] Stanfield, C. & German, W. (2009) Fisiologia. Edises.
- [8] http://www.led-tec.net/it/definizione-di-luce/]
- [9] https://www.my-personaltrainer.it/salute-benessere/retina.html
- [10] https://medicinaonline.co/2018/01/28/fotorecettori-differenza-tra-coni-e-bastoncelli/
- [11] http://www.anisn.it/matita_ipertesti/visione/tretipi.htm
- [12] Bierman A, Figueiro MG, Rea MS. "Measuring and predicting eyelid spectral transmittance.] Biomed Opt.". Internet: www.lrc.rpi.edu/resources/newsroom/projectsheets.asp]. (2011)
- [13] Andrew Bierman , Mariana G. Figueiro, Mark S. Rea "Measuring and predicting eyelid spectral transmittance." Journal of Biomedical Optics, vol. 16(6), 1June 2011
- [14] https://docs.agi32.com/AGi32/Content/references/Photopic_Mesopic_Scotopic_-_Concepts.htm
- [15] http://www.lionettiformazione.it/2015/10/15/il-ciclo-della-visione/
- [16] Prof. Giuseppe Boccignone "Dall'occhio al cervello: fototrasduzione nella retina." Internet:
- http://homes.dsi.unimi.it/~boccignone/GiuseppeBocignone_webpage/Modelli_Percezione.ht ml

- [17] Wolfgang Baehr , Samuel M. Wu ,Alan C. Bird. (2003) "The retinoid cycle and retina disease" Elsevier.[Online], pp. 2957-2958 , Available: www.sciencedirect.com
- [18] Alfred F. Bliss. (August 26, 1947). "Themechanism of retinal vitamin a formation." Journal of Biological Chemistry. [Online]. pp. 165-178. Avalaible: http://www.jbc.org/
- [19] https://www.chimica-online.it/download/trasmittanza.htm
- [20] Datasheet LED hyper red"Kingbright". Internet :http://www.KingbrightUSA.com/ApplicationNotes.
- [21] Sasaki, T., Itakura, T., Suzuki, K., Kasuya, H., Munakata, R., Muramatsu, H. & al. (2010) Intraoperative monitoring of visual evoked potential: introduction of a clinically useful method. J. Neurosurg., 112, 285-294.
- [22] Feinsod, M., Selhorst, J. B., Hoyt, W., & Wilson, C. B. (1976). Monitoring optic nerve during craniotomy. J. Neurosurg., 44, 29-31.
- [23] Lesser, R. P., Lüders, H., Klem, G. & Dinner, D. S. (1985) Visual potentials evoked by light-emitting diodes mounted in goggles *Cleveland Clinic Quarterly*, *52*,, 223-228.
- [24] Pratt, H., Martin, W., ,, Zaaroor, M., &Schacham, S. (1994). A highintensity, gogglemounted flash stimulator for short-latency visual evoked potentials. *Electroencephalography and clinical Neurophysiology*, *92*, 469-472.
- [25] Kodama, K., Goto, T., Sato, A., Sakai, K., Tanaka, Y., &Hongo, K. (2010). Standard and limitation of intraoperative monitoring of the visual evoked potential. *ActaNeurochir*, 152, 643-648.
- [26] Luo, Y., Regli, L., Bozinov, O., & Sarnthein, J. (2015). Clinical utility and limitations of intraoperative monitoring of visual evoked potentials. *PLoS ONE*, doi:10.1371/journal.pone.0120525.
- [27] Valeria Novelli. "Sviluppo di un dispositivo ottico per la genesi dei potenziali evocati intraoperatori: setup del protocollo di stimolazione, esecuzione delle misure, elaborazione ed analisi dei risultati." Tesi di Laurea Magistrale, Politecnico di Torino, Torino, 2016.
- [28] ANIE Automazione, "Guida per la tecnologia Wireless." Maggio, 2013.
- [29] Antonio Sciarappa. "Tecnologie e Soluzioni Applicative Wireless (Networking 3)." Politecnico Telemedicina, 2016.
- [30] Texas Instruments . "Wireless Connectivity." 2017.
- [31] MicrotherapeuticsSas. "Gli Elettromedicali In Sala Operatoria."
- [32] https://www.techopedia.com/definition/27785/industrial-scientific-and-medical-radio-band-ism-band
- [33] Texas Instruments." CC2640 SimpleLink™ Bluetooth® Wireless MCU."February, 2015.

- [34] Texas Instruments. "LaunchPadDevelopment KitQuick Start Guide".2015.
- [35] Joakim Lindh, Christin Lee and Marie Hernes. "Measuring Bluetooth Low Energy Power Consumption." February ,2015
- [36] CC2640 and CC2650 SimpleLink— Bluetooth® low energy Software Stack 2.2.1, Developer's Guide.
- [37] Carles Gomez, Joaquim Oller e JosepParadells., "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology", In: Sensors 12.9 (2012), pp. 11734-11753.
- [38] http://dev.ti.com/tirex/#/?link=Software%2FSimpleLink%20CC2640R2%20SDK%2FExamples%2FDevelopment%20Tools%2FCC2640R2%20LaunchPad%2Fblestack%2Fsimple_central
- [39] http://dev.ti.com/tirex/#/?link=Software%2FSimpleLink%20CC2640R2%20SDK% 2FExamples%2FDevelopment%20Tools%2FCC2640R2%20LaunchPad%2Fblestack%2Fsimple_peripheral
- [40] http://dev.ti.com/tirex/content/simplelink_cc2640r2_sdk_1_50_00_58/docs/driverlib_cc13xx_cc26xx/cc26x0r2/driverlib/index.html
- [41] Texas Instruments. "CC13x0, CC26x0 SimpleLink™ Wireless MCU, Technical Reference Manual."February, 2015
- [42] RoHS COMPLIANT. "ECS-TXO-2520." Revised: 04.02.17.
- [43] "Illuminotecnica Fondamenti", Corso di Fisica Tecnica Ambientale.
- [44] Andrea Nicolini. "Illuminotecnica." Università degli Studi di Perugia.
- [45] https://www.luxemozione.com/2016/01/controllo-dellabbagliamento-ugr-unified.html
- [46] http://disanoledtechnology.it/it/ugr-e-flicker-per-la-salute-e-il-benessere-negli-uffici/
- [47] "NORME ILLUMINOTECNICHE."UNI EN 12464-1.
- [48] https://docs.agi32.com/AGi32/Content/adding_calculation_points/Calculations_UGR_Concepts.htm
- [49] Commission Internationale de l'Eclairage (CIE) . "Calculation and Presentation of the Standard CIE UGR Table for Indoor Lighting Luminaires." Mar. 10, 2008.
- [50] "Introduzione al calcolo illuminotecnico." ITI Omar, Novara
- [51] Yung-Sheng Chen, Chung-Yi Lin, Chun-Ming Yeh, Chie-Tong Kuo, Chih-Wei Hsu, Hsiang-Chen Wang. "Anti-glare LED lamps with adjustable illumination light field.", 2014.

- [52] http://www.ohmslawcalculator.com/mcd-to-lumens
- [53] Dott. IngLuca Armisi. "Illuminazione in Sala Operatoria Esigenze cliniche, criteri di valutazione e innovazione tecnologica." Fondazione PTV Policlinico Tor Vergata.
- [54] Eunicell. "3V Lithium manganese Dioxide Button Cell Data Sheet."
- [55] Mathias Jensen. "Coin cells and peak current draw." 2010.
- [56] Joakim Lindh, Christin Lee and Marie Hernes. "Measuring Bluetooth Low Energy Power Consumption." February 2015–Revised January 2017.
- [57] Christin Lee and Fredrik Kervel ."Running Bluetooth® low energy on CC2640 Without 32 kHz Crystal."June 2016–Revised April 2017.
- [58] https://e2e.ti.com/support/wireless_connectivity/bluetooth_low_energy/f/538/t/498652
- [59] Bluetooth Power Calculator Tool. url: http://www.ti.com/tool/BTPOWER-CALC
- [60] http://www.ti.com/tool/SENSOR-CONTROLLER-STUDIO
- [61] http://www.ti.com/tool/CCSTUDIO
- [62] CC26xx, CC13xx Sensor Controller Studio Version 2.0.
- [63] http://dev.ti.com/tirex/content/simplelink_academy_cc2640r2sdk_1_14_02_04/modules/sc_01_project_from_scratch/sc_01_project_from_scratch.html
- [64] MultiRole. url: https://github.com/ti-simplelink/ble-sdk-210-extra/tree/master/Projects/ble/multi_role
- [65] Audun Andersen, Application NoteAN043, Topic: "Small Size 2.4 GHz antenna", 2008
- [66] SONY MicroBattery. "Lithium Manganese Dioxide Battery Data Sheet CR2032."