



Ricerca di Sistema elettrico

Sistemi ibridi di accumulo per l'incremento dello sfruttamento della risorsa rinnovabile nell'ambito delle comunità energetiche (anno 2 di 3)

M. A. Ancona, L. Branchini, A. De Pascale, F. Ferrari, F. Melino, S. Ottaviano



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

SISTEMI IBRIDI DI ACCUMULO PER L'INCREMENTO DELLO SFRUTTAMENTO DELLA RISORSA RINNOVABILE NELL'AMBITO DELLE COMUNITÀ ENERGETICHE (ANNO 2 DI 3)

M. A. Ancona, L. Branchini, A. De Pascale, F. Ferrari, F. Melino, S. Ottaviano (Dipartimento di Ingegneria Industriale – Alma Mater Studiorum – Università di Bologna)

Aprile 2021

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico (oggi Ministero della Transizione Ecologica) - ENEA

Piano Triennale di Realizzazione 2019-2021 - II annualità

Obiettivo: *N. 1 - Tecnologie*

Progetto: *1.5 - Tecnologie, tecniche e materiali per l'efficienza energetica ed il risparmio di energia negli usi finali elettrici degli edifici nuovi ed esistenti*

Work package: *3 - Componenti e impianti innovativi per incrementare l'efficienza energetica e l'uso delle fonti rinnovabili negli edifici*

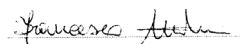
Linea di attività: *LA3.8 - Sistemi ibridi di accumulo per l'incremento dello sfruttamento della risorsa rinnovabile nell'ambito delle comunità energetiche – (anno 2 di 3)*

Responsabile del Progetto: Giovanni Puglisi, ENEA

Responsabile del Work package: Biagio Di Pietra, ENEA

Il presente documento descrive le attività di ricerca svolte all'interno dell'Accordo di collaborazione *"Sistemi ibridi di accumulo per l'incremento dello sfruttamento della risorsa rinnovabile nell'ambito delle comunità energetiche e Sistemi di microgenerazione per lo sfruttamento di biogas finalizzati alla produzione di acqua calda sanitaria nelle piccole isole"*

Responsabile scientifico ENEA: Ing. Biagio Di Pietra



Indice

SOMMARIO.....	4
INTRODUZIONE.....	6
1 STATO DELL'ARTE DELLE PIATTAFORME ELETTRONICHE PROGRAMMABILI OPEN SOURCE.....	8
1.1 PIATTAFORME ELETTRONICHE ARDUINO	9
1.1.1 Applicazioni: dall'hobbistica a didattica, industria e ricerca	9
Caratteristiche tecniche e funzionalità	11
1.1.2 Arduino IDE e software compatibili.....	18
1.2 PIATTAFORME ELETTRONICHE RASPBERRY	24
1.2.1 Modelli prodotti e caratteristiche tecniche	24
1.2.2 Sistema operativo e applicazioni.....	31
1.3 SOLUZIONE PROPOSTA PER IL BANCO PROVA DI EMULAZIONE	33
2 COSTITUZIONE DEL BANCO DI PROVA PER L'EMULAZIONE DI UN SISTEMA DI PRODUZIONE ED ACCUMULO DI IDROGENO ED ENERGIA ELETTRICA DA FONTE RINNOVABILE NON PROGRAMMABILE	35
2.1 IL MODELLO S.A.P.I.E.N.T.E E L'IMPIANTO PILOTA DEL LABORATORIO DI MECCANICA	35
2.2 TEST DI REGOLAZIONE SULLA MICRORETE INSTALLATA PRESSO IL LABORATORIO DI MECCANICA DELL'UNIVERSITÀ DI BOLOGNA	39
2.3 CARATTERIZZAZIONE SPERIMENTALE DELLA FUEL CELL.....	49
2.4 MODELLAZIONE IN SIMULINK DEI COMPONENTI D'IMPIANTO E COSTITUZIONE DEL LAYOUT	52
2.4.1 Modello di calcolo del pannello fotovoltaico.....	52
2.4.2 Modello di calcolo del dispositivo di accumulo elettrochimico	55
2.4.3 Modello di calcolo dell'elettrolizzatore	57
2.4.4 Modello di calcolo della fuel cell	59
2.4.5 Modello di calcolo del sistema di compressione ed accumulo dell'idrogeno prodotto	61
2.4.6 Modello di calcolo del sistema pompa di calore - accumulo termico.....	64
2.5 PROGETTAZIONE IN STATEFLOW DELLA STRATEGIA DI GESTIONE DEI FLUSSI ENERGETICI	67
2.5.1 Calibrazione delle sottosezioni d'impianto e definizione delle logiche di funzionamento	67
2.6 PROGRAMMAZIONE DELLE SCHEDE ARDUINO E MESSA A PUNTO DEL PROTOCOLLO DI COMUNICAZIONE ARDUINO-SIMULINK.....	74
2.6.1 Procedura di configurazione dei dispositivi Arduino e del modello Simulink	75
3 SIMULAZIONE DI FUNZIONAMENTO DEL SISTEMA ENERGETICO S.A.P.I.E.N.T.E. E TEST DELL'APPARATO DI EMULAZIONE SPERIMENTALE	83
3.1 ANALISI DELLE SIMULAZIONI EFFETTUATE E PROVE DI EMULAZIONE ARDUINO - SIMULINK	85
4 CONCLUSIONI.....	96
5 RIFERIMENTI BIBLIOGRAFICI	99
6 NOMENCLATURA	101
7 INDICE DELLE FIGURE.....	103
8 INDICE DELLE TABELLE	106
9 CURRICULUM DEL GRUPPO DI LAVORO	107

Sommario

Il presente documento costituisce il rapporto tecnico dell'attività di ricerca svolta durante la seconda annualità del Piano Triennale di Realizzazione 2019-2021, frutto della collaborazione tra l'ENEA e il Dipartimento di Ingegneria Industriale (DIN) dell'Alma Mater Studiorum – Università di Bologna, finalizzata allo studio di sistemi ibridi di accumulo per l'incremento dello sfruttamento della risorsa rinnovabile non programmabile nell'ambito delle comunità energetiche. Tale collaborazione nasce all'interno dell'Accordo di Programma che il Ministero dello Sviluppo Economico ed ENEA hanno stipulato nel 2019 finalizzato all'esecuzione delle linee di attività del Piano Triennale di Ricerca e Sviluppo di Interesse Generale per il Sistema Elettrico Nazionale.

Più in dettaglio, l'attività relativa alla seconda annualità del progetto è dedicata allo sviluppo di un laboratorio di ricerca finalizzato all'individuazione di strategie ottimizzate per la gestione real time di reti ibride complesse per la generazione, accumulo e conversione da fonti rinnovabili non programmabili, nell'ambito delle comunità energetiche.

Partendo infatti dalle linee guida tratteggiate in seguito alla prima annualità del progetto, durante la quale si è potuta definire l'architettura ideale per una rete di produzione, accumulo ed utilizzo di idrogeno nell'ambito civile e/o residenziale, nonché selezionarne le tecnologie più opportune e le taglie corrispondenti, la prosecuzione del progetto prevede di mettere a punto un'infrastruttura dedicata alla sperimentazione ed allo studio delle logiche di funzionamento e gestione della rete stessa. Il laboratorio in oggetto sarà quindi realizzato secondo le logiche dell'hardware in the loop e permetterà di sviluppare sistemi di emulazione e controllo (centraline) real time per la gestione ottimizzata di reti complesse.

L'ambiente operativo sarà infatti composto da un insieme di sistemi fisici e dispositivi elettronici di emulazione finalizzati a studiare le prestazioni delle tecnologie facenti parte del sistema energetico in esame e la loro integrazione con esso. I dispositivi oggetto di studio sono:

- un sistema di generazione elettrica da fonte rinnovabile non programmabile (pannelli fotovoltaici e/o generatori eolici);
- un sistema per la produzione di idrogeno (elettrolizzatore);
- un sistema di compressione e stoccaggio dell'idrogeno prodotto;
- un sistema di riconversione dell'idrogeno, basato su tecnologia fuel cell;

La realizzazione di un laboratorio del tipo hardware in the loop è finalizzata ad una maggiore flessibilità operativa e/o alla possibilità di esplorare e sperimentare preventivamente l'impiego di differenti soluzioni impiantistiche e taglie dei componenti coinvolti, servendosi pertanto di alcuni emulatori di sistema.

La presente linea di attività può essere articolata nelle seguenti fasi.

FASE I: Definizione dell'architettura della rete di idrogeno

Tale fase prevede la definizione dei sistemi e/o attrezzature necessari ad integrare quelli già disponibili presso il Dipartimento di Ingegneria Industriale e/o lo studio dell'adeguamento degli stessi al fine di realizzare un banco prova per la programmazione di centraline di controllo.

In tale fase si terrà conto dei criteri di scaling dei componenti da acquisire laddove le taglie in oggetto non dovessero risultare compatibili con la fase di sperimentazione prevista. Tra i sistemi da acquisire si prevedono dispositivi di emulazione al fine di rappresentare il reale comportamento di reti di idrogeno di taglie diverse nonché l'interazione con i sistemi attualmente installati presso il centro ricerche Casaccia dell'ENEA.

FASE II: Acquisizione componenti, installazione, test e collaudo

In base ai risultati della precedente fase si procederà all'installazione, presso la sede del Laboratorio di Meccanica del Dipartimento di Ingegneria Industriale dell'Università di Bologna, della rete di idrogeno in oggetto. Si procederà in tale fase anche allo sviluppo dei modelli virtuali alla base degli emulatori che saranno impiegati ed alla loro integrazione con i sistemi fisici installati.

La fase di test e collaudo permetterà di giungere alla piena operatività della rete ibrida di generazione, accumulo e conversione da fonti rinnovabili non programmabili ponendo le basi per la terza annualità del progetto, dedicata alla realizzazione di una centralina per la gestione ottimizzata real time del sistema.

Introduzione

Il costante aumento della domanda di energia su scala globale e le preoccupazioni ambientali legate allo sfruttamento massiccio delle risorse fossili stanno spronando l'universo scientifico allo studio di soluzioni tecnologiche più sostenibili per la produzione e l'accumulo dell'energia in tutte le sue forme, destinando risorse sempre maggiori alla ricerca di metodi per la riduzione delle emissioni inquinanti dirette ed indirette di tutta la filiera. Le sole tecnologie di approvvigionamento da fonti rinnovabili attualmente a disposizione non consentono un abbandono repentino dei tradizionali sistemi di produzione dell'energia da combustibili fossili a causa di limiti sull'efficienza, costi elevati e non programmabilità della produzione energetica; anche immaginando, in uno scenario futuribile, una copertura energetica mondiale proveniente da sole fonti rinnovabili, l'impatto ambientale dell'intera filiera sarebbe tutt'altro che trascurabile. Per quanto, infatti, le principali tecnologie di generazione elettrica da fonti rinnovabili, fotovoltaico, eolico ed idroelettrico su tutte, siano percepite culturalmente come risorse ad impatto ambientale nullo o quasi, bisogna distinguere tra le loro emissioni inquinanti dirette (dovute essenzialmente allo sfruttamento della fonte durante il periodo di funzionamento del dispositivo) e le emissioni indirette (ovvero quelle che non derivano direttamente dall'attività del sistema energetico bensì conseguenti all'intero ciclo di produzione e smaltimento dello stesso). Le principali problematiche che caratterizzano l'impatto ambientale dei sistemi da fonte rinnovabile sono infatti legate alle attività estrattive dei materiali impiegati per la loro fabbricazione, con le relative preoccupazioni politiche e sociali, nonché alle difficoltà di manutenzione e gestione delle centrali e del loro decommissioning a fine vita, del tutto similmente a quanto già accade per molti dei sistemi energetici tradizionali. D'altra parte, gli elevati *"time to market"* delle tecnologie più innovative (LNG – *Liquefied Natural Gas*, LH2 – *Liquefied Hydrogen*, fusione nucleare), spesso incrementati dalla sindrome NIMBY (*Not In My Back Yard*), rendono impossibile nel breve periodo la messa a punto delle infrastrutture necessarie alla transizione energetica su larga scala.

Un'interessante prospettiva in ambito civile/residenziale, decisamente meno affetta da problematiche di accettazione sociale, viene offerta dalle moderne microreti di generazione ibrida di tipo *"power to hydrogen to power"*, vale a dire impianti di generazione, accumulo e conversione dell'energia elettrica da fonte rinnovabile che si servono dell'idrogeno come vettore energetico utile allo stoccaggio a lungo termine. L'idrogeno gassoso prodotto in questi impianti dall'elettrolisi dell'acqua ed accumulato come riserva energetica è successivamente riconvertibile in energia elettrica tramite l'utilizzo di – ad esempio – celle a combustibile, senza emissioni inquinanti e climalteranti. In questo modo è possibile massimizzare lo sfruttamento in loco delle risorse rinnovabili quali vento e sole ed abbattere i costi dell'elettrolisi, altrimenti sconsigliato, consentendo un utilizzo *"off grid"* dell'impianto ed eliminando quindi perdite e costi dovuti al trasporto dell'energia dalle grandi centrali.

La minuziosa gestione dei flussi nonché un'appropriata logica di funzionamento, a seconda della collocazione geografica e delle necessità dell'utilizzatore, sono gli aspetti principali da tenere in considerazione per accrescere l'efficienza di tali sistemi, dal momento che le risorse energetiche rinnovabili di cui si servono sono intrinsecamente affette dalla non programmabilità, ed è quindi necessario, ai fini dell'indipendenza energetica, massimizzare il loro sfruttamento mediante la predisposizione di adeguati sistemi d'accumulo dell'energia, capaci di garantire autonomia anche a fronte di picchi della domanda ed eccessi di produzione.

In quest'ottica, la digitalizzazione dei sistemi di controllo ed acquisizione dati nonché di simulazione preventiva in fase di progettazione si rivela un passo fondamentale, ai fini di un corretto dimensionamento dei componenti e del corretto soddisfacimento dei consumi elettrici/termici. Le moderne piattaforme hardware e software open source per la progettazione elettronica, come Arduino e Raspberry, costituiscono un grosso potenziale per lo sviluppo tecnologico delle smart grids, consentendo di ridurre notevolmente i costi dovuti alla sperimentazione fisica durante la fase di studio, e garantendo un'elevata compatibilità con i più diffusi software di modellazione e simulazione dinamica.

L'attività di ricerca riportata nel presente documento si è concentrata pertanto sull'impiego di alcune piattaforme hardware low cost, più in particolare dei microcontrollori Arduino, per l'emulazione di alcuni dispositivi durante lo studio di un sistema per la produzione e l'accumulo di idrogeno da fonte rinnovabile. È in questo contesto, infatti, che si inserisce il lavoro svolto dal Dipartimento di Ingegneria Industriale (DIN) dell'Alma Mater Studiorum - Università di Bologna in collaborazione con ENEA, per il Piano Triennale di Realizzazione 2019-2021 dedicato allo studio di soluzioni tecnologiche per il raggiungimento dell'indipendenza energetica nei piccoli distretti urbani

1 Stato dell'arte delle piattaforme elettroniche programmabili open source

Nell'era della digitalizzazione dei sistemi e dell'IoT (*Internet Of Things*) sono sempre di più le applicazioni tecnologiche in cui vi è un'integrazione tra componenti elettronici e sistemi fisici per la gestione di processi ed apparati, tradizionalmente monitorati in maniera analogica. Basti pensare alla filiera dei trasporti, in particolare al settore *automotive*, in cui l'avvento dell'elettronica ha portato in breve tempo allo sviluppo di sistemi di sicurezza ed assistenza alla guida, nonché predisposizioni per la guida autonoma da parte del veicolo stesso, difficilmente immaginabili fino a qualche decennio fa. Le competenze acquisite nel tempo dai più grandi gruppi industriali e dal mondo della ricerca hanno permesso l'incursione delle tecnologie digitali verso tutti i campi di applicazione delle attività umane, migliorandone l'esperienza degli utenti coinvolti e permettendo, nella maggior parte dei casi, un vantaggioso processo di snellimento delle apparecchiature impiegate.

Tuttavia, nonostante la forte spinta economica verso la transizione digitale, molte delle tecnologie più innovative richiedono grandi risorse per poter essere sviluppate ed implementate nei più disparati campi di applicazione, ed il reperimento dei fondi necessari all'acquisto di apparecchiature e strumenti informatici per la sperimentazione è spesso causa di notevoli rallentamenti per la ricerca. L'impiego di strumenti hardware e software open source offre in tale contesto un notevole supporto al mondo della ricerca che da alcuni anni sta vedendo crescere di numero gli studi condotti e supportati da sistemi di questo tipo, in segno di una considerevole riduzione dei costi, spesso proibitivi, ed una migliore interoperabilità tra apparati e strutture.

Il movimento FOSS (Free and Open Source Software) nasce infatti come un universo libero dalle logiche di crescita tipicamente commerciali ed al contrario si pone in maniera del tutto trasparente e collaborativa per lo sviluppo di software sulla base delle richieste della comunità che vi partecipa [1]. Tale processo di innovazione è quindi totalmente decentralizzato ed in contrasto con la concezione ampiamente diffusa di software commerciale, il quale nella maggior parte dei casi viene scritto per l'utente prevedendone le future esigenze e non a partire dalle attuali necessità. Quest'ultima modalità di sviluppo presenta inoltre alcuni importanti limiti relativamente all'impossibilità di apportare modifiche al codice, che spesso risulta essere di proprietà esclusiva del produttore, ed agli alti costi delle licenze d'uso, tipicamente crescenti in base a specificità e funzionalità del prodotto. Al contrario, la comunità degli sviluppatori di software open source spesso dialoga intorno ai temi chiave dell'etica di sviluppo ed è fonte di conoscenza nella scienza (in molti casi, il software open source è un output dei processi di ricerca), producendo interdisciplinarietà tra i diversi campi di applicazione come la medicina e l'ingegneria. I progressi effettuati in questo senso hanno permesso nel tempo di approdare al paradigma open source anche per lo sviluppo di piattaforme hardware, tra le quali è possibile certamente annoverare Arduino e Raspberry Pi, nate proprio sulla base delle esigenze, prevalentemente di tipo scolastico ed accademico, di disporre di apparecchiature personalizzate e low cost per le attività di ricerca sperimentale ed insegnamento dei linguaggi di programmazione [2, 3].

La diffusione nel tempo di tali dispositivi ha interessato progressivamente molte aree tematiche differenti trovando applicazione in settori come la medicina, la biologia, la fisica ed anche l'ingegneria nelle sue varie declinazioni (elettronica e robotica in primis).

Anche in ambito energetico sono ormai in crescita i progetti di ricerca che si servono di Arduino e Raspberry quali strumenti di supporto ai fini della sperimentazione, garantendo così una netta riduzione dei costi ed una maggiore compatibilità tra sistemi open source. Relativamente al contesto delle microreti energetiche intelligenti è infatti possibile servirsi dei dispositivi Arduino per realizzare sistemi SCADA ("Supervisory Control And Data Acquisition") per la misura ed il monitoraggio delle variabili fisiche dell'impianto in tempo reale [4-6]. L'adozione di tali dispositivi permette inoltre una grande versatilità di configurazione dell'interfaccia finale di acquisizione, la quale può essere predisposta attraverso i principali

software commerciali come LabVIEW e Simulink, mediante le ormai diffuse librerie d'interazione con microcontrollori e schede programmabili, oppure essere programmata autonomamente via codice [7, 8]. L'implementazione delle piattaforme programmabili come Arduino e Raspberry consente inoltre di realizzare architetture in grado di operare da remoto, basandosi su server web per l'importazione e la manipolazione dei dati rilevati in ambiente sperimentale [9-11].

Oltre a quanto detto, l'impiego dei moderni microcontrollori programmabili può essere di fondamentale importanza anche per lo studio di ottimizzazione delle microreti energetiche, dal punto di vista della gestione algoritmica dei flussi. È infatti noto che i sistemi per la produzione dell'energia basati sulla risorsa rinnovabile non programmabile necessitano di un'attenta gestione degli scambi energetici tra componenti e quindi di sofisticate strategie di controllo che possano incrementarne l'efficienza e minimizzare gli sprechi [12, 13]. In tal senso, le moderne tecniche di simulazione ed emulazione dei sistemi fisici basate su pratiche di tipo HIL ("Hardware In the Loop") risultano una via ampiamente percorribile per la messa a punto dei sistemi di controllo benché ostacolata dalla necessità di strumentazione dedicata e molto costosa. Alcuni recenti studi hanno mostrato le possibilità d'impiego dei dispositivi Arduino e Raspberry per l'implementazione di architetture HIL e SIL ("Software In the Loop") per l'emulazione di alcune microreti energetiche di piccola taglia, determinando un chiaro vantaggio dal punto di vista della semplificazione generale delle apparecchiature e della riduzione dei costi [14-17].

Per una maggiore chiarezza espositiva, nel prossimo paragrafo verranno brevemente descritti i dispositivi Arduino e Raspberry attualmente in commercio, al fine di enunciarne le principali caratteristiche hardware, software e di implementazione con i principali software di calcolo utilizzati nell'ambito della ricerca scientifica.

1.1 Piattaforme elettroniche Arduino

1.1.1 Applicazioni: dall'hobbistica a didattica, industria e ricerca

Si può certamente affermare che uno dei principali punti di forza dei prodotti Arduino sia costituito dalla grande versatilità d'impiego. La piattaforma è stata progettata infatti attraverso un insieme di strutture informatiche davvero semplici ed intuitive, adatte a chiunque desideri imparare a programmare i microcontrollori per interagire fisicamente con gli eventi del mondo reale e virtuale.

Arduino è stato concepito infatti per fungere da sostegno ad amatori e professionisti senza particolari competenze informatiche nelle loro applicazioni: dagli hobbisti alle installazioni artistiche, dai designer all'architettura moderna e perfino all'ingegneria meccanica.

È immediato comprendere che il ruolo di discipline quali l'elettronica e l'informatica saranno sempre più centrali per lo sviluppo delle attività umane nei prossimi decenni ed in virtù di questo inesorabile processo di digitalizzazione l'introduzione nel mondo dell'istruzione di progetti ed applicazioni basate sull'utilizzo di piattaforme elettroniche come Arduino sta avendo grandi riscontri a livello globale.

È così che istituti universitari, scuole superiori e perfino medie stanno dedicando sempre più spazio all'insegnamento dei concetti di base della programmazione per realizzare progetti con Arduino utili allo studio durante i corsi, grazie anche all'enorme reperibilità a basso costo dei kit di prototipazione rapida (Figura 1) sviluppati ed ampiamente conosciuti all'interno della community.

Tuttavia, la potenzialità di Arduino sta anche nella possibilità di realizzare applicazioni molto complesse e precise, a patto di conoscenze informatiche decisamente più avanzate, al punto da essere abbondantemente impiegato in numerosi settori dell'industria moderna, e perfino dalla ricerca accademica. È in costante crescita, infatti, la comunità di professionisti che si serve di questi sistemi elettronici in ambito lavorativo, per risolvere problemi di lieve entità quali piccole applicazioni robotiche o di trasduzione dei segnali (Figura 2) sino all'utilizzo in processi decisamente più complicati utilizzati in ambito automotive per testare i sistemi automatici e le centraline elettroniche dei veicoli.

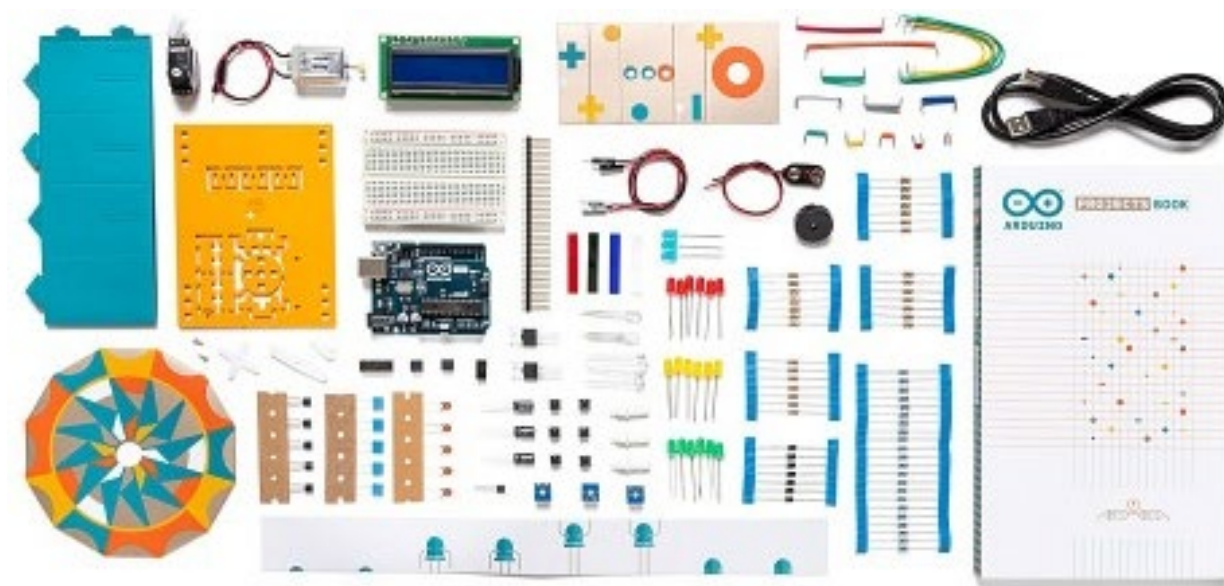


Figura 1 - Arduino starter kit [2]

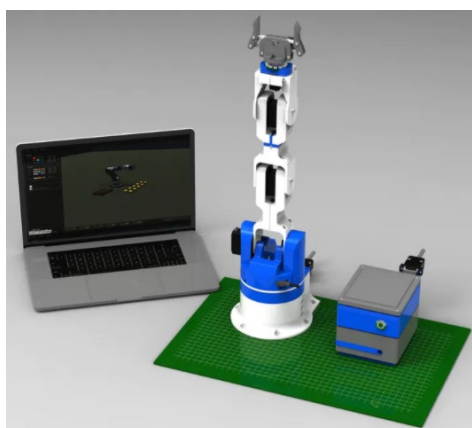


Figura 2 - Arduino robotic arm (sinistra) e Arduino controlled spoiler (destra) [2]

Anche nell'ambito della ricerca Arduino sta riscuotendo un grande successo, specialmente se si considerano le sue potenzialità in relazione ai costi, spesso motivo di arresto in ambito accademico, poiché sono ormai molti gli istituti universitari nel mondo ad aver introdotto nella didattica lo studio di tali dispositivi con conseguenti forti riflessioni sull'attività di ricerca, specialmente in ambito ingegneristico.

Per tali ragioni anche in ambito energetico ci sono buone prospettive di crescita per l'impiego di piattaforme elettroniche analoghe ad Arduino durante il controllo e l'automazione dei sistemi energetici, specialmente se si parla di applicazioni residenziali o di piccola taglia in cui sono fondamentali digitalizzazione ed automatizzazione dei sistemi in ottica di contenimento dei costi ed ottimizzazione di flussi e consumi, nonché per tutte quelle attività di trasduzione dei segnali da remoto ed analisi dei dati.

Caratteristiche tecniche e funzionalità

La varietà di microcontrollori proposti dalla gamma Arduino è, come già accennato, notevole e, seppure tutti i prodotti presenti in listino appartengano alla medesima categoria, le differenze in termini di prestazioni, destinazione d'uso e costi possono risultare determinanti nella scelta della scheda.

La **Tabella 1** riportata di seguito mostra la gamma completa di tutte le schede Arduino con le relative caratteristiche tecniche. Come è possibile notare dalla tabella, le caratteristiche delle schede proposte dalla gamma Arduino possono essere anche molto diverse da modello a modello, in quanto subordinate a scelte progettuali ed architetture elettroniche differenti. Le applicazioni dei microcontrollori Arduino sono infatti innumerevoli e nella scelta della scheda si deve tenere conto di condizioni operative e spazio utile all'installazione, che sta portando i costruttori alla realizzazione di schede sempre più compatte ed ottimizzate per l'implementazione su piccoli dispositivi elettromeccanici. Per l'impiego proposto in questa sede, le dimensioni del controller non hanno assunto particolare rilevanza in quanto non vi è stata alcuna necessità di utilizzo dinamico, si è altresì ritenuto opportuno considerare, ai fini della scelta, aspetti come il numero di porte disponibili alle connessioni, le prestazioni in termini di potenza di calcolo del processore e la robustezza del prodotto in relazione al prezzo contenuto. Tali requisiti, unitamente alla flessibilità d'impiego ed alla reperibilità di supporto in rete grazie al grande successo commerciale della scheda hanno permesso di optare per il modello Mega 2560, le cui caratteristiche sono riportate nella **Tabella 2**.

Tabella 1 - Gamma completa schede Arduino [2]

Nome	Processore	Operativo /ingresso tensione	Velocità CPU	Ingresso /uscita analogici	IO /PWM digitali	EEPROM [kB]	SRAM [kB]
101	Intel® Curie	3,3 V / 7-12 V.	32MHz	6/0	14/4	-	24
Gemma	ATtiny85	3,3 V / 4-16 V	8 MHz	1/0	3/2	0.5	0.5
LilyPad	ATmega168V ATmega328P	2,7-5,5 V / 2,7-5,5 V	8MHz	6/0	14/6	0.512	1
LilyPad SimpleSnap	ATmega328P	2,7-5,5 V / 2,7-5,5 V	8 MHz	4/0	9/4	1	2
LilyPad USB	ATMEGA32U4	3,3 V / 3,8-5 V	8 MHz	4/0	9/4	1	2.5
Mega 2560	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8
Micro	ATMEGA32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5
MKR1000	SAMD21 Cortex- M0 +	3,3 V / 5 V.	48MHz	7/1	8/4	-	32
professionista	ATmega168 ATmega328P	3,3 V / 3,35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6	0,512 1	1 2
Pro Mini	ATmega328P	3,3 V / 3,35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6	1	2
Uno	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1	2
Zero	ATSAMD21G18	3,3 V / 7-12 V	48 MHz	6/1	14/10	-	32
Dovuto	ATSAM3X8E	3,3 V / 7-12 V	84 MHz	12/2	54/12	-	96
ESPLORA	ATMEGA32U4	5 V / 7-12 V	16 MHz	-	-	1	2.5
Ethernet	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/4	1	2
Leonardo	ATMEGA32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5
Mega ADK	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8
Mini	ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	1	2

Tabella 1 – Continua

nano	ATmega168 ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	0,512 1	1 2
Yun	ATmega32U4 AR9331 Linux	5 V.	16 MHz 400MHz	12/0	20/7	1	2.5 16 MB
Arduino Robot	ATMEGA32U4	5 V.	16 MHz	6/0	20/6	1 KB (ATmega32u4) / 512 Kbit (I2C)	2,5 KB (ATmega32u4)
MKRZero	SAMD21 MCU ARM a bassa potenza Cortex-M0 + 32 bit	3.3 V	48 MHz	7 (ADC 8/10/12 bit) / 1 (DAC 10 bit)	22/12	No	32 KB

Tabella 2 - Caratteristiche tecniche scheda Arduino Mega 2560 [2]

Microcontroller	ATmega2560
Tensione operativa	5V
Tensione di ingresso (consigliata)	7-12V
Tensione di ingresso (limite)	6-20V
Pin I / O digitali	54 (di cui 15 forniscono output PWM)
Pin di ingresso analogico	16
Corrente DC per pin I / O	20 mA
Corrente CC per pin 3.3V	50 mA
Memoria flash	256 KB di cui 8 KB utilizzati dal bootloader
SRAM	8 KB
EEPROM	4 KB
Velocità di clock	16 MHz
LED_BUILTIN	13
Lunghezza	101.52 mm
Larghezza	53,3 mm
Peso	37 g

Per comprendere al meglio la struttura e le funzionalità specifiche del modello di controllore prescelto per la sperimentazione è possibile considerare la **Figura 3**, in cui è mostrata l'architettura di una scheda Arduino mega 2560. Ciò che si nota immediatamente, a partire dal lato sinistro della scheda, sono i due connettori, l'uno di alimentazione e l'altro di tipo USB (Universal Serial Bus). Quest'ultimo consente al dispositivo di connettersi al computer per caricare le istruzioni, denominate "sketch", sulla scheda e scambiare dati con il PC, oltre che per fornire alimentazione alla scheda stessa qualora l'assorbimento elettrico necessario all'applicazione non superi i limiti consentiti dalla porta USB, ovvero 5 V di tensione operativa.

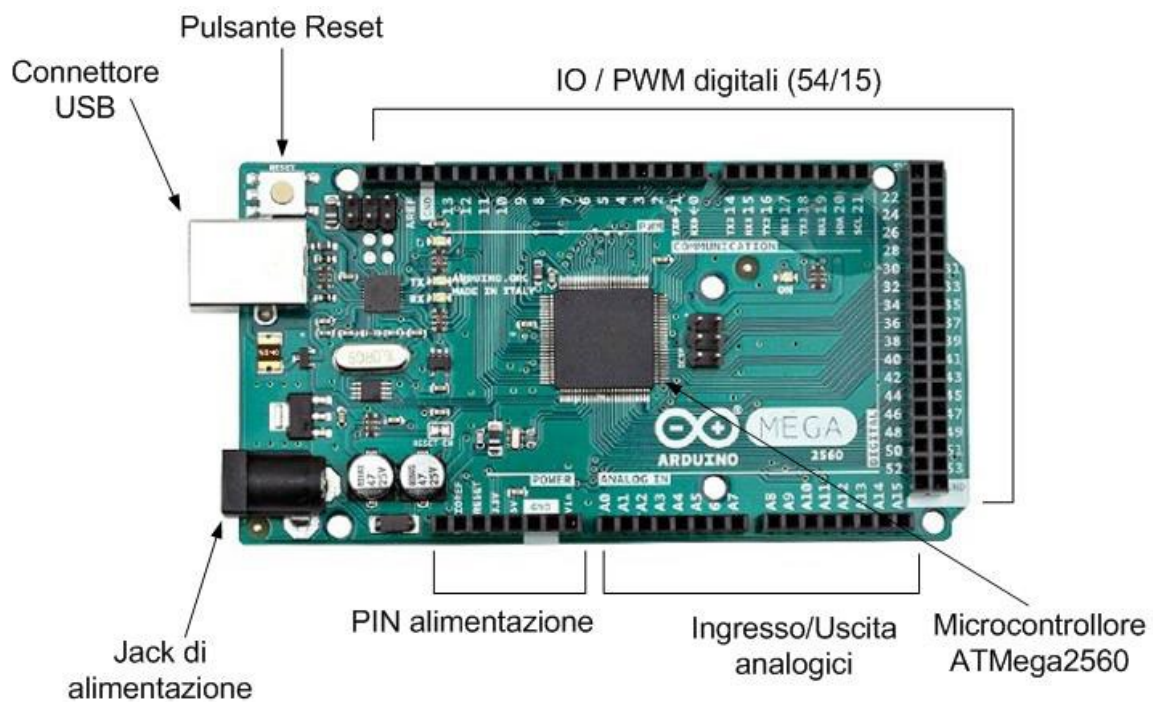


Figura 3- Scheda Arduino Mega 2560 [2]

Vi è infatti la possibilità che la connessione ad Arduino di eventuali sensori o dispositivi esterni porti ad avere un assorbimento elettrico durante il funzionamento superiore a quanto consentito dalla scheda o che si debba impiegare la stessa per applicazioni di tipo stand-alone in cui non è praticabile un collegamento di tipo USB. A tal proposito i progettisti hanno predisposto un connettore di tipo Jack da 3.5 mm (**Figura 4**) tramite cui è sufficiente un alimentatore per fornire energia alla scheda attraverso una normale presa di corrente standard.



Figura 4 - Alimentatore 9 V DC per Arduino [2]

Al centro della scheda salta subito all'occhio il vero cuore pulsante del dispositivo, ovvero il microcontrollore ATmega2560 prodotto da Atmel, azienda statunitense produttrice di semiconduttori e specializzata nella realizzazione di microcontrollori di tipo system-on-a-chip con memoria flash integrata.

Questo chip (**Figura 5**) rappresenta a tutti gli effetti il “cervello” del circuito stampato ed è composto da un processore (capace di eseguire le varie istruzioni) e differenti tipi di memoria (dedicate all'archiviazione dei dati in maniera permanente e volatile).



Figura 5 - Chip Atmel ATmega2560

Perimetralmente alla scheda sui lati destro, superiore e inferiore sono invece installati una serie di connettori che offrono un elevato numero di ingressi analogici e digitali tramite cui è possibile inviare o ricevere segnali elettrici variabili in tensione. È possibile suddividere queste porte in tre gruppi principali che, procedendo dall'alto in senso orario, corrispondono rispettivamente a 54 pin di tipo I/O digitale di cui ben 15 utilizzabili in PWM (Pulse Width Modulation), 16 ingressi/uscite di tipo analogico ed infine ulteriori 8 pin per gestire l'alimentazione della scheda. Per una maggiore chiarezza è di seguito riportata, in **Figura 7**, la pin-map di alcuni modelli di schede Arduino, tra cui la Mega2560.

Per avvisare l'utente della presenza di alimentazione e dello scambio dati in corso sono inoltre posizionati sulla scheda alcuni dispositivi LED (Light Emitting Diode), come mostrato in [Figura 6](#), che si illuminano quando sono attraversati da una corrente. Per la precisione il modello Mega2560 è dotato di 4 LED: tutto a destra è presente un LED contrassegnato dalla dicitura "ON" che testimonia la presenza di alimentazione alla scheda, mentre sulla sinistra si trovano altri tre LED che, partendo dall'alto, sono etichettati L, TX ed RX. Il primo è collegato al pin 13 mentre gli altri due rappresentano rispettivamente la trasmissione e la ricezione dei dati attraverso la porta di comunicazione seriale.

Infine, nell'angolo in alto a sinistra della scheda è presente il pulsante Reset che consente di riavviare e ripristinare il sistema del controllore qualora vi siano anomalie nel funzionamento.

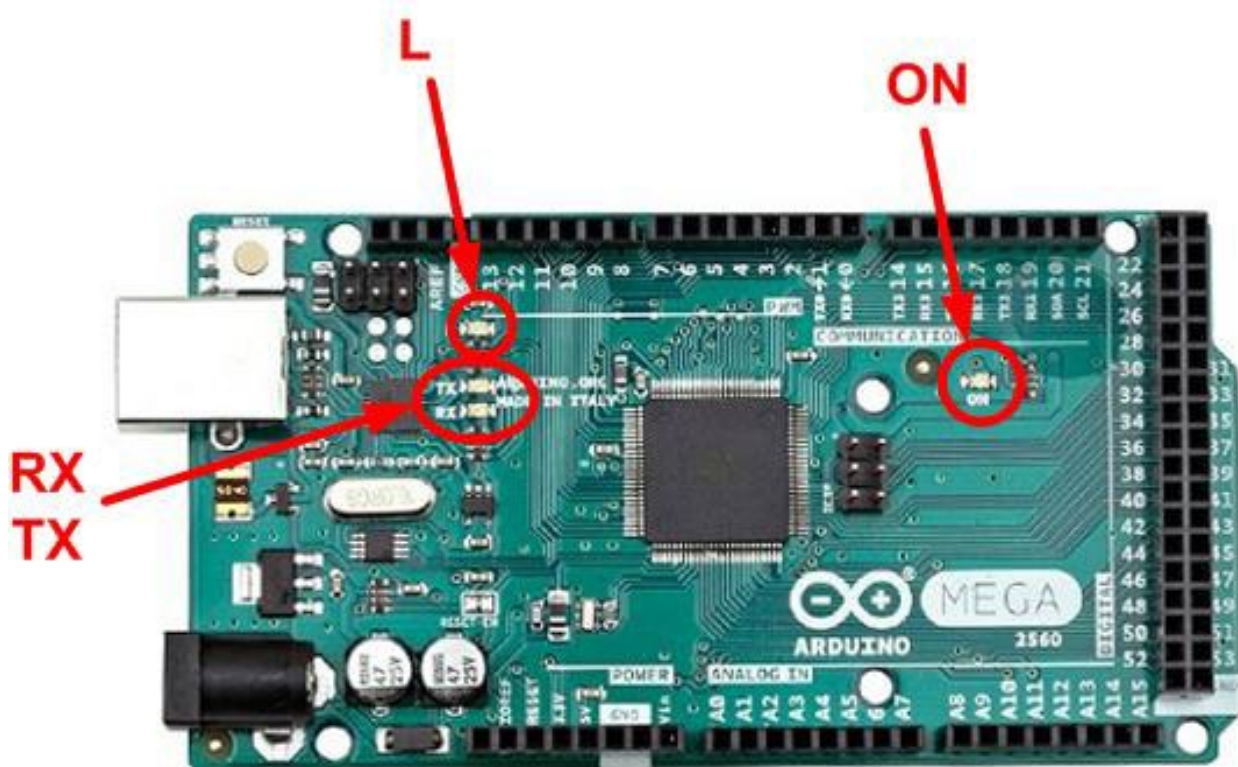
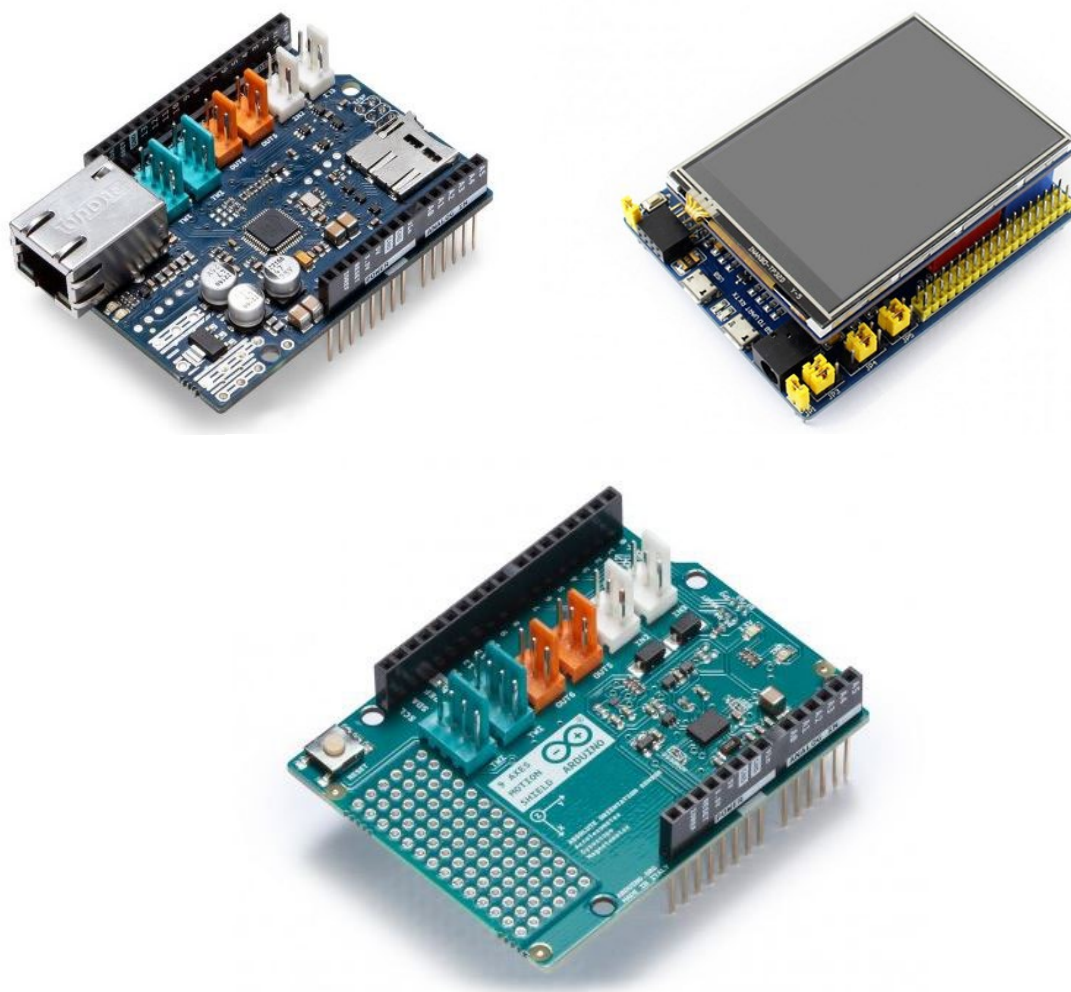


Figura 6 - Mappa dei LED Arduino Mega2560 R3 [2]

	Blocks	Target Hardware											
		Uno	Nano 3.0	Leonardo	MiRo	Mega 2560 ADK	Due	Robot Control Board	Robot Motor Board	MKR1000	MKR WIFI 1010	MKRZero	Nano 33 IoT
User Selectable Pins	Digital Input	0-13	0-13	0-19	0-19	0-53	0-53	0-3,5-17,19-22	0-19	0-14	0-14	0-14,32	0-13
	Digital Output	0-13	0-13	0-19	0-19	0-53	0-53	0-3,5-17,19-22	0-19	0-14	0-14	0-14,32	0-13
	Analog Input	0-5	0-7	0-11	0-11	0-15	0-11	0,1,2,3,4,5,6,7,11	0,1,2,3,4,5,6,11	0-6	0-6	0-6	0,1,2,3,6,7
	Analog Output	N/A	N/A	N/A	N/A	N/A	DAC0, DAC1	N/A	N/A	DAC0	DAC0	DAC0	DAC0
	PWM	3,5,6,9,10,11	3,5,6,9,10,11	3,5,6,9,10,11,13	3,5,6,9,10,11,13	2-13, 44-46	2-13	N/A	N/A	0-8, 10, 11, 18(A3), 19(A4)	0-8, 10, 18(A3), 19(A4)	0-8, 10, 18(A3), 19(A4)	2,3,4,5,6,9,10,11,12, 16(A2), 17(A3), 19(A5)
	Standard Servo Read	0-13	0-13	0-19	0-19	0-53	0-53	19,20,21,22	4,12,18,19	0-14, 15(A0), 16(A1), 17(A2), 18(A3), 19(A4), 19(A5), 21(A6)	0-14, 15(A0), 16(A1), 17(A2), 18(A3), 19(A4), 19(A5), 21(A6)	0-14, 15(A0), 16(A1), 17(A2), 18(A3), 19(A4), 19(A5), 21(A6)	0-13
	Standard Servo Write	0-13	0-13	0-19	0-19	0-53	0-53	19,20,21,22	4,12,18,19	0-14, 15(A0), 16(A1), 17(A2), 18(A3), 19(A4), 19(A5), 21(A6)	0-14, 15(A0), 16(A1), 17(A2), 18(A3), 19(A4), 19(A5), 21(A6)	0-14, 15(A0), 16(A1), 17(A2), 18(A3), 19(A4), 19(A5), 21(A6)	0-13
	Continuous Servo Write	0-13	0-13	0-19	0-19	0-53	0-53	19,20,21,22	4,12,18,19	0-14, 15(A0), 16(A1), 17(A2), 18(A3), 19(A4), 19(A5), 21(A6)	0-14, 15(A0), 16(A1), 17(A2), 18(A3), 19(A4), 19(A5), 21(A6)	0-14, 15(A0), 16(A1), 17(A2), 18(A3), 19(A4), 19(A5), 21(A6)	0-13
	External Interrupt	2,3	2,3	0,1,2,3,7	0,1,2,3,7	2,3,18-21	0-53	0,1,2,3,7	0,1,2,3,7	0-1,4-9, 16(A1), 17(A2)	0-1,4-9, 16(A1), 17(A2)	0-1,4-9, 16(A1), 17(A2)	2,3,9,10,11,13,15,19,21
	8Pi Slave Select (SS)	0-10	N/A	0-19	0-19	0-49, 53	4,10,52	0-19	0-19	0-7,11-14	0-7,11-14	0-7,11-14	0-10
Fixed Pins	8Pi MOBI	11/ ICSP-4	11	ICSP-4	ICSP-4	51/ ICSP-4	ICSP-4	ICSP-4	ICSP-4	8	8	8	11
	8Pi MI80	12/ ICSP-1	12	ICSP-1	ICSP-1	50/ ICSP-1	ICSP-1	ICSP-1	ICSP-1	10	10	10	12
	8Pi SCK	13/ ICSP-3	13	ICSP-3	ICSP-3	52/ ICSP-3	ICSP-3	ICSP-3	ICSP-3	9	9	9	13
	I2C SDA	A4	D4	2/ SDA	2	20	20	D2	D2	11	11	11	18(A4)
	I2C SCL	A5	D5	3/ SCL	3	21	21	D3	D3	12	12	12	19(A5)
	Serial Receive	Port 0: pin 0 Port 1: N/A Port 2: N/A Port 3: N/A	Port 0: pin 0 Port 1: N/A Port 2: N/A Port 3: N/A	Port 0: No Pins Port 1: pin 0 Port 2: N/A Port 3: N/A	Port 0: No Pins Port 1: pin 0 Port 2: N/A Port 3: N/A	Port 0: pin 0 Port 1: pin 19 Port 2: pin 17 Port 3: pin 15	Port 0: pin 0 Port 1: pin 19 Port 2: pin 17 Port 3: pin 15	Port 0: No pins Port 1: pin 0 Port 2: N/A Port 3: N/A	Port 0: No pins Port 1: pin 0 Port 2: N/A Port 3: N/A	Port 0: No Pins Port 1: pin 13	Port 0: No Pins Port 1: pin 13	Port 0: No Pins Port 1: pin 13	Port 0: pin 0 Port 1: N/A Port 2: N/A Port 3: N/A
	Serial Transmit	Port 0: pin 1 Port 1: N/A Port 2: N/A Port 3: N/A	Port 0: pin 1 Port 1: N/A Port 2: N/A Port 3: N/A	Port 0: No Pins Port 1: pin 1 Port 2: N/A Port 3: N/A	Port 0: No Pins Port 1: pin 1 Port 2: N/A Port 3: N/A	Port 0: pin 1 Port 1: pin 18 Port 2: pin 16 Port 3: pin 14	Port 0: pin 1 Port 1: pin 18 Port 2: pin 16 Port 3: pin 14	Port 0: No pins Port 1: pin 1 Port 2: N/A Port 3: N/A	Port 0: No pins Port 1: pin 1 Port 2: N/A Port 3: N/A	Port 0: No Pins Port 1: pin 14	Port 0: No Pins Port 1: pin 14	Port 0: No Pins Port 1: pin 14	Port 0: pin 1 Port 1: N/A Port 2: N/A Port 3: N/A
	TCP/IP Receive	10,11,12,13	N/A	10,11,12,13	N/A	10,50,51,52,53	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	TCP/IP Send	10,11,12,13	N/A	10,11,12,13	N/A	10,50,51,52,53	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	UDP Send	10,11,12,13	N/A	10,11,12,13	N/A	10,50,51,52,53	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	UDP Receive	10,11,12,13	N/A	10,11,12,13	N/A	10,50,51,52,53	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	ThingSpeak Write	10,11,12,13	N/A	10,11,12,13	N/A	10,50,51,52,53	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	WiFi TCP/IP Receive	10,11,12,13	N/A	10,11,12,13	N/A	10,50,51,52,53	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	WiFi TCP/IP Send	10,11,12,13	N/A	10,11,12,13	N/A	10,50,51,52,53	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	WiFi UDP Send	10,11,12,13	N/A	10,11,12,13	N/A	10,50,51,52,53	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	WiFi UDP Receive	10,11,12,13	N/A	10,11,12,13	N/A	10,50,51,52,53	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	CAN Transmit	9,11,12,13	N/A	9,ICSP-1,ICSP-3,ICSP-4	9,ICSP-1,ICSP-3,ICSP-4	9,50,51,52	10,ICSP-1,ICSP-3,ICSP-4	N/A	N/A	3,8,9,10	3,8,9,10	3,8,9,10	3,8,9,10
	CAN Receive	2,9,11,12,13	N/A	2,9,ICSP-1,ICSP-3,ICSP-4	2,9,ICSP-1,ICSP-3,ICSP-4	2,9,50,51,52	2,10,ICSP-1,ICSP-3,ICSP-4	N/A	N/A	3,7,8,9,10	3,7,8,9,10	3,7,8,9,10	3,2,8,9,10

Uno dei grandi vantaggi del sistema Arduino sta, come già accennato, nella facilità di espansione della scheda mediante un'infinità di componenti hardware che ne permettono l'ampliamento delle funzionalità. È possibile, infatti, attraverso le file di pin connettere delle "shield" di espansione, ossia ulteriori circuiti stampati, per dotare la scheda di interfaccia Ethernet per comunicare su reti locali ed IoT, display aggiuntivi muniti dei più vari sensori e piattaforme inerziali a più assi, nonché numerose ulteriori applicazioni (**Figura 8**).



1.1.2 Arduino IDE e software compatibili

Una volta apprese le nozioni di base riguardo la parte hardware dei dispositivi Arduino è altrettanto importante fornire una rapida panoramica del software tramite cui è possibile programmare le schede elettroniche e monitorarne il funzionamento.

L'ambiente di sviluppo integrato di Arduino (Arduino IDE) permette infatti di pianificare tutte le istruzioni che indicheranno all'hardware cosa fare e come farlo mediante la creazione di un listato denominato "Sketch" realizzabile direttamente all'interno del software IDE e caricabile sulla scheda connessa al pc tramite cavo USB.

Come è possibile vedere in **Figura 9**, l'interfaccia dell'IDE si presenta come un semplice editor di testo la cui schermata principale risulta suddivisa in tre zone: la porzione di spazio più in alto raccoglie tutte le

principali icone di comando del software, tra cui quelle necessarie al riconoscimento ed alla sincronizzazione del modello di scheda connesso nonché quelle utili a creazione e salvataggio dello *sketch* (ovvero il codice contenente le istruzioni da caricare sulla scheda Arduino). Appena sotto è possibile notare la vera e propria finestra dell'editor, in cui verranno visualizzate tutte le istruzioni da caricare sulla scheda e sotto ad essa si trova la terza zona, ossia la banda nera entro cui vengono segnalati i messaggi dovuti ad eventuali errori durante la compilazione o il caricamento dello sketch.

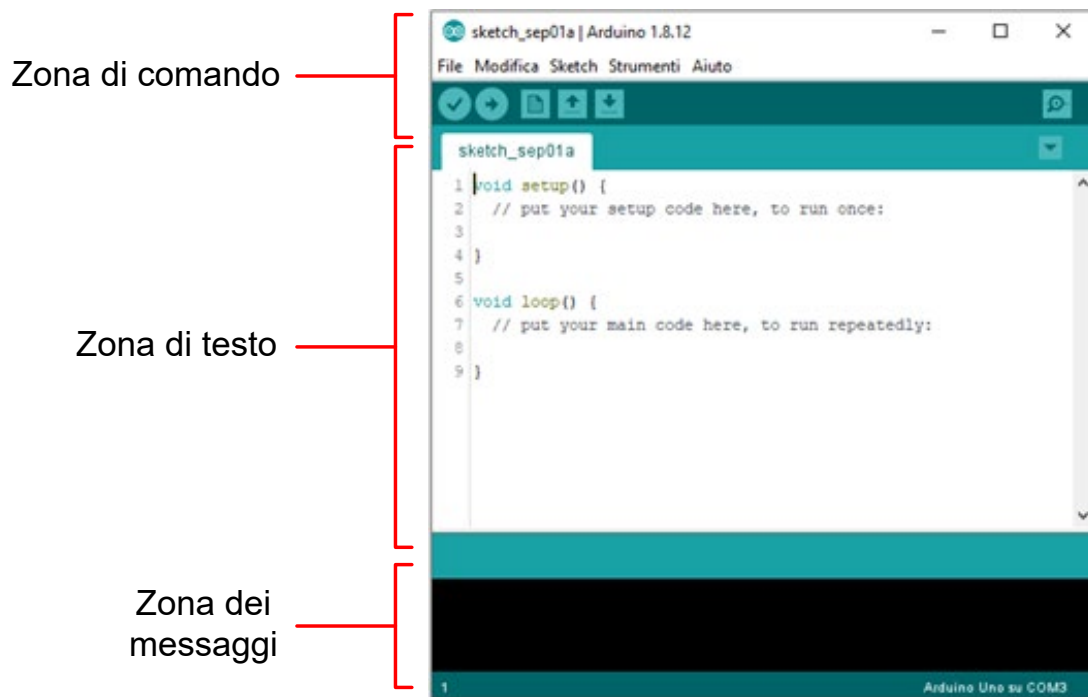


Figura 9 - Arduino IDE

In particolare, oltre alle classiche funzionalità, tipiche degli editor di testo, presenti nella fascia più alta della zona di comando è possibile notare appena sotto di esse una serie di icone dedicate all'interazione tra l'ambiente di programmazione e la scheda, come mostrato in Figura 10.

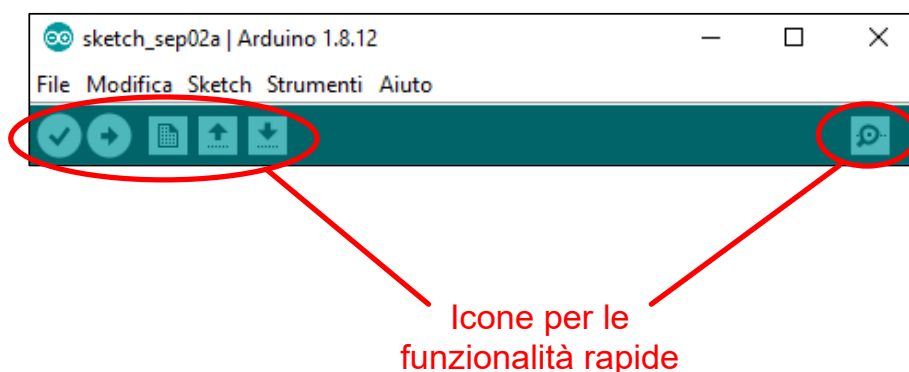


Figura 10 - Icone IDE

Ad ognuna di esse è associata una funzione che partendo da sinistra corrisponde rispettivamente all'azione:

- **Verifica:** consente di verificare che lo sketch di Arduino realizzato sia corretto e non contenga errori di sintassi
- **Carica:** dopo aver verificato lo sketch è possibile, cliccando su carica, realizzare l'upload del codice sulla scheda desiderata
- **Nuovo:** per aprire un nuovo foglio bianco
- **Apri:** per aprire uno sketch precedentemente realizzato e memorizzato in archivio
- **Salva:** per salvare lo sketch corrente
- **Monitor seriale:** per aprire la finestra del monitor seriale ([Figura 11](#)), ovvero quello strumento che permette di visualizzare istantaneamente i dati che la scheda Arduino comunica al terminale tramite la porta seriale COM.

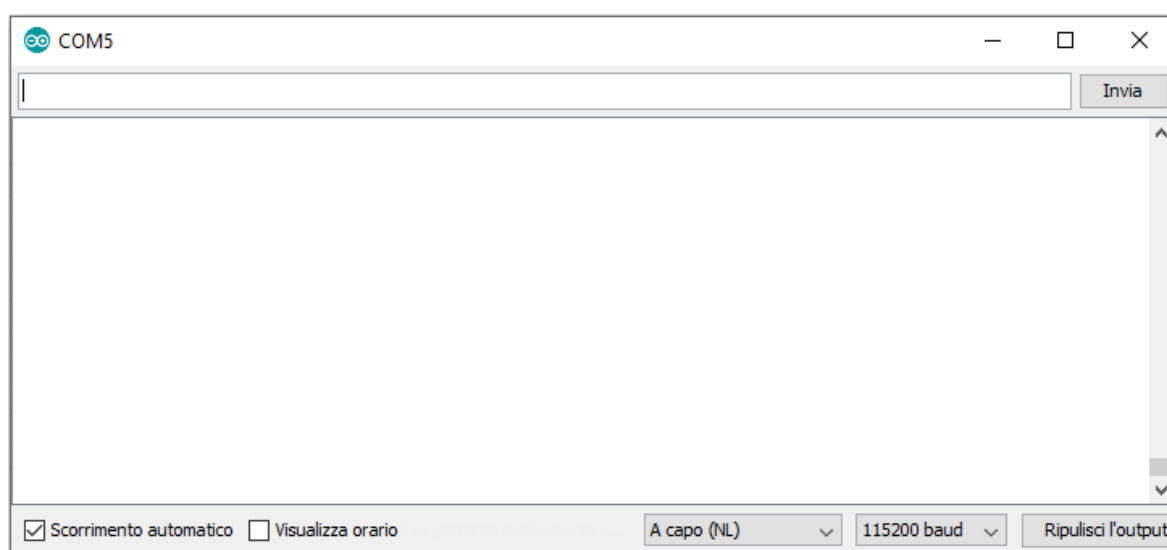


Figura 11 - Finestra di interfaccia del monitor seriale

L'ambiente di sviluppo Arduino IDE è scritto in Java ed il linguaggio adottato all'interno del medesimo per la creazione dei codici è derivato dal C/C++ anche se l'attività di programmazione è in realtà semplificata dall'integrazione, all'interno dell'IDE stesso, di alcuni strumenti quali il controllo della sintassi, delle parentesi e l'indentazione automatica del codice [21]. È inoltre possibile ridurre l'onere di programmazione servendosi dei numerosi pacchetti di personalizzazione direttamente scaricabili dalla sezione comandi e contenenti librerie precostituite con funzioni matematiche o specifiche per l'utilizzo di alcuni componenti hardware.

La struttura standard di ogni sketch è quindi pensata per facilitare l'utilizzo della scheda anche a chi non possiede particolari conoscenze di tipo informatico ed è infatti precostituita all'apertura dell'editor, dove viene richiesta la definizione di due sole funzioni, come mostrato in [Figura 12](#).

```

1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }

```

Figura 12 - Struttura dello Sketch

`void setup ()` e `void loop ()` serviranno rispettivamente ad inizializzare lo sketch tramite alcune impostazioni iniziali, come la tipologia di comunicazione scelta per scambiare dati con la scheda o la velocità della stessa (baud rate), le quali rimarranno invariate durante il funzionamento, e a definire le vere e proprie istruzioni da far eseguire ripetutamente al microcontroller.

L'ambiente di sviluppo di Arduino non costituisce però l'unica modalità di programmazione della scheda in quanto sono ormai diversi i software di programmazione/progettazione dinamica tramite cui è possibile interagire con i controllori Arduino per monitorarne il funzionamento ed inviare a questi delle nuove istruzioni.

Tra le numerose possibilità una menzione particolare va certamente a Matlab e Labview, che hanno sviluppato nel tempo dei veri e propri tool, gratuitamente fruibili dai relativi store, grazie ai quali è possibile continuare a lavorare mediante il linguaggio di programmazione proprietario del software (di tipo grafico nel caso di Simulink e Labview) anche per interagire con i dispositivi Arduino. Molte delle librerie messe a punto dalla community sono infatti decisamente intuitive e semplificano notevolmente il lavoro di progettazione e controllo in tempo reale della scheda grazie alle numerose funzioni predefinite già presenti nel tool.

È questo il caso dei tools (**Figura 13**) *Matlab/Simulink Support Package for Arduino Hardware* pensato da Mathworks e *LIFA (Labview Interface for Arduino)* della National Instruments, dedicati per l'appunto a quanto descritto finora.

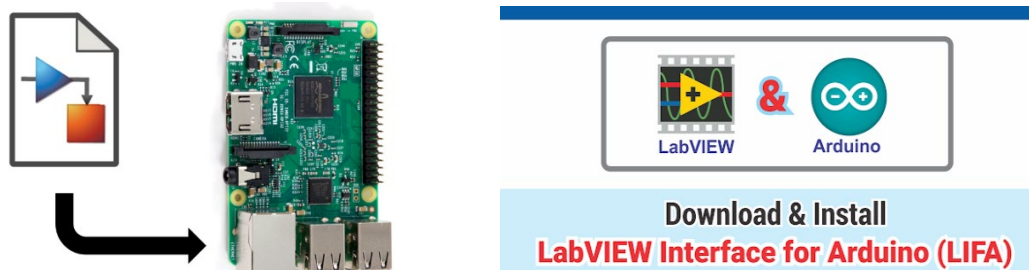


Figura 13 - Matlab Support Package for Arduino Hardware e LIFA

Per quanto riguarda il primo ne esistono due versioni distinte, l'una dedicata interamente alla progettazione in Matlab e l'altra in Simulink, le quali permettono allo stesso modo di comunicare con la scheda Arduino mediante la connessione via cavo USB. Tramite l'installazione del toolbox, che consente l'esecuzione continua del programma di interfaccia sulla scheda, è infatti possibile creare e caricare modelli

di Matlab/Simulink sul dispositivo, il quale li eseguirà in loop secondo le impostazioni selezionate all'interno del software di comando (Matlab/Simulink). Nel caso di Simulink il pacchetto in questione prevede un'ampia libreria dotata di blocchi predefiniti (Figura 14) per la configurazione e l'accesso a tutti i sensori, attuatori ed interfacce di comunicazione di cui è provvista la scheda Arduino, grazie ai quali risultano facilitate, rispetto al caso di una programmazione in codice, le attività di controllo di sensori e dispositivi elettromeccanici, quali piccoli motori DC, e di acquisizione dei dati analogici e digitali nonché l'elaborazione grafica dei segnali.

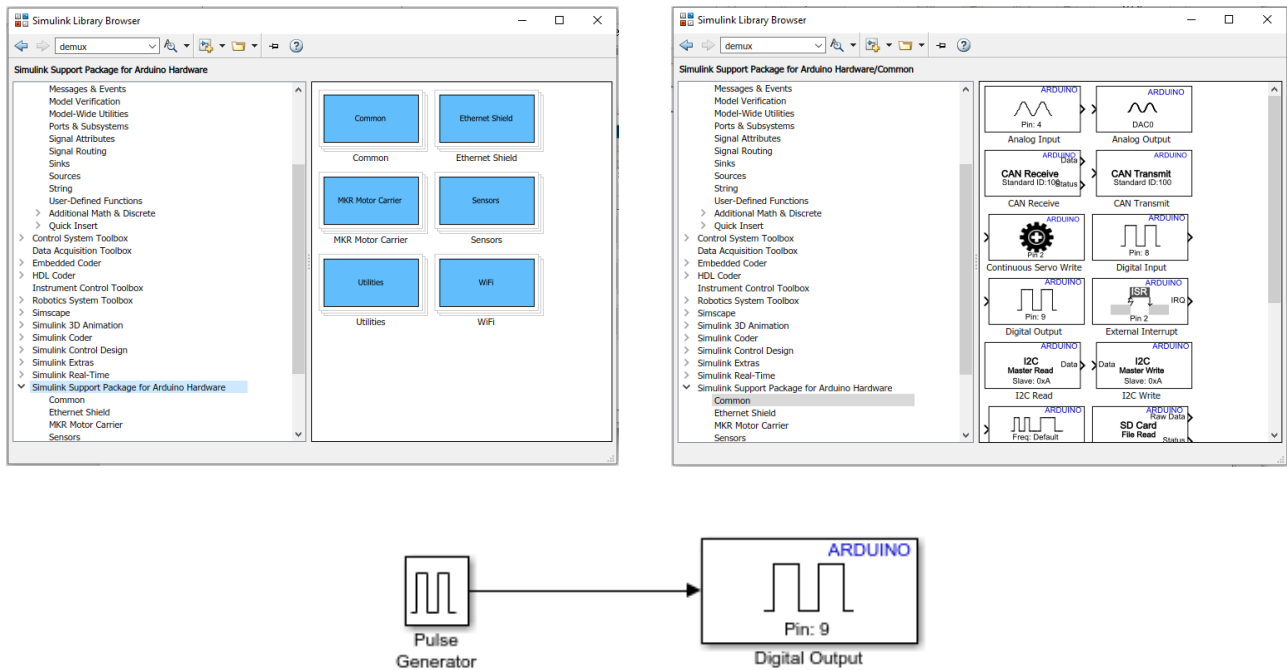


Figura 14 –Simulink Support Package for Arduino Hardware

I principali aspetti negativi legati all'utilizzo di questo toolbox sono essenzialmente dovuti alle caratteristiche tecniche specifiche di ciascun modello di scheda Arduino, in quanto non per tutti si ha la possibilità di utilizzare interamente la libreria a blocchi, ed in secondo luogo all'impossibilità di accedere ai codici sorgente delle funzioni incluse all'interno della libreria. Nell'eventualità di voler realizzare uno scambio dati più libero tra la scheda Arduino e l'ambiente Matlab/Simulink è quindi necessario adottare soluzioni meno intuitive rispetto al toolbox appena descritto e di conseguenza meno immediate nell'utilizzo. Servendosi di applicazioni più generiche quali ad esempio il noto *Instrument Control Toolbox* è infatti possibile, non senza passare questa volta attraverso l'IDE di Arduino, scambiare dati in modo personalizzato e senza limiti di intento, sulla base del tipo di comunicazione che si sceglie mettere a punto tra il dispositivo ed il terminale (Figura 15, Figura 16).

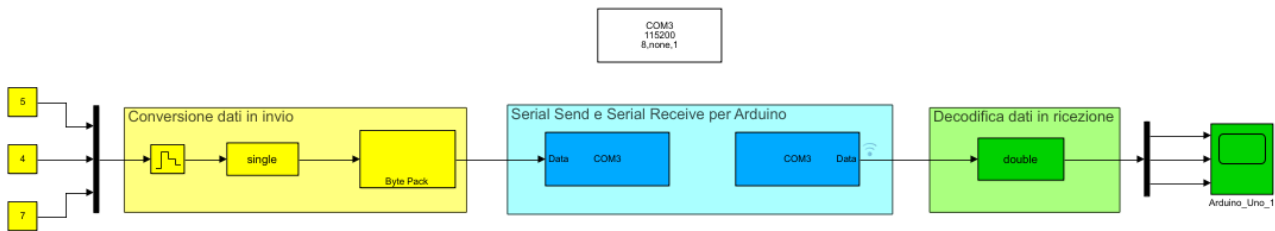


Figura 15 - Modello realizzato tramite l'Instrument Control Toolbox

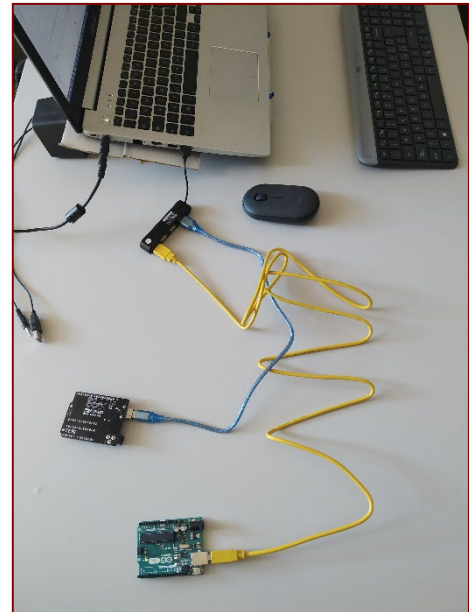
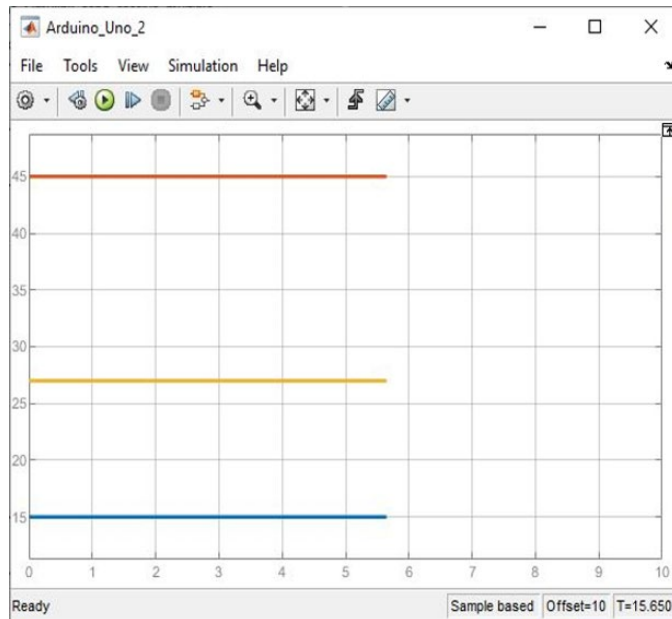


Figura 16 - Esempio di scambio dati tramite l'ICT

Analogamente a quanto visto per il software Matlab/Simulink è possibile interagire con le schede Arduino anche dall'ambiente di lavoro di Labview che come già accennato mette a disposizione un Toolkit del tutto simile a quello appena descritto. Grazie a *LINX* (**Figura 17**), il nuovo pacchetto di National Instruments in sostituzione al precedentemente citato *LIFA (Labview Interface for Arduino)*, si ha la possibilità di creare applicazioni avanzate sulle principali piattaforme hardware in commercio (Arduino, Raspberry, chipKIT, ecc.) semplificandone lo sviluppo attraverso l'utilizzo diretto del linguaggio grafico di Labview [20]. Come per l'*Instrument Control Toolbox* fruibile in ambiente Matlab/Simulink, anche in questo caso è possibile accedere alle periferiche I/O digitale ed analogico, PWM, I2C, SPI e UART e controllare le schede tramite connessione USB, seriale Ethernet o Wi-Fi, riducendo notevolmente l'onere di programmazione che richiederebbe una progettazione realizzata attraverso il solo IDE di Arduino.

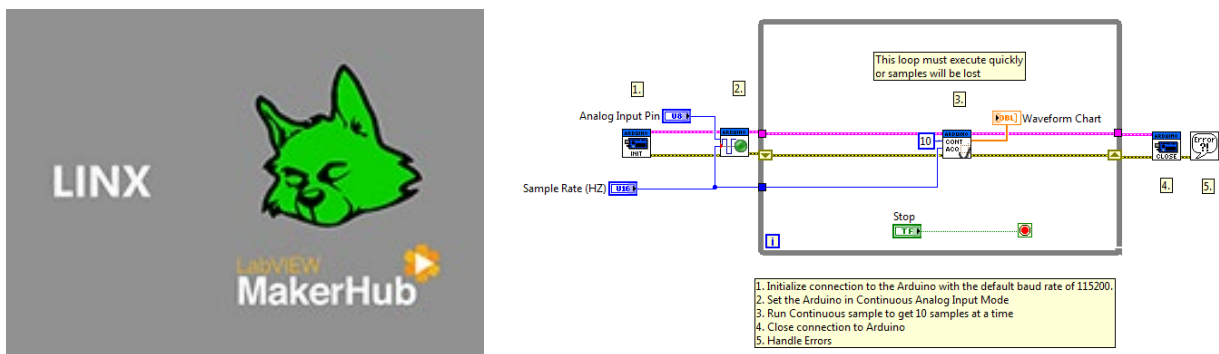


Figura 17 - Logo LINX ed esempio di applicazione

Riassumendo sono quindi numerose ad oggi le possibilità di interfacciare i dispositivi Arduino con software e linguaggi di programmazione di altri produttori oltre ai due presentati finora e sarebbe praticamente impossibile citarli tutti. Basti sapere che la scelta di rendere la piattaforma Arduino open source ha permesso negli anni agli sviluppatori di mettere a punto e diffondere attraverso la fitta community svariate soluzioni a tale scopo, sia per creare ed eseguire istruzioni sia per l'analisi dei dati durante il funzionamento delle schede. Questa caratteristica, come si è visto in precedenza, rende il progetto Arduino estremamente competitivo ed accessibile nell'ambito della progettazione elettronica, al punto da essere considerato un ottimo strumento di partenza per scopi didattici e di ricerca.

1.2 Piattaforme elettroniche Raspberry

Come già anticipato in fase di introduzione, le piattaforme Raspberry Pi costituiscono dei sistemi SBC (Single Board Computer), vale a dire dei veri e propri computer implementati su delle singole schede elettroniche di piccole dimensioni.

Nonostante la taglia ridotta (simile a quella di una carta di credito, come del resto si è visto per i dispositivi Arduino) ed un aspetto non immediatamente riconducibile a quello di un calcolatore elettronico, le schede Raspberry dispongono a tutti gli effetti della componentistica necessaria per operare come dei computer, comprese le porte di comunicazione per connettere i principali dispositivi di interfaccia con l'utente, come mouse, tastiera e monitor.

L'idea di un simile prodotto, non diversamente da quanto accadde per Arduino, nacque a metà degli anni duemila per rendere più accessibile e stimolante lo studio dell'informatica e dei linguaggi di programmazione nelle scuole, spronando i giovani studenti ad approfondire le strutture ed i sistemi informatici che stanno alla base dei moderni e tanto ambiti apparecchi elettronici multimediali, quali smartphone, tablet e laptop [22].

Ad oggi la Raspberry Pi Foundation può vantare un grande successo in tutto il mondo, specialmente grazie al nobile intento educativo dei suoi prodotti, la cui gamma è ormai vasta ed evoluta per poter soddisfare le più disparate richieste e fasce di prezzo. È proposta nel seguito una rapida panoramica dei principali modelli di schede elettroniche Raspberry Pi attualmente in commercio e delle loro caratteristiche tecniche.

1.2.1 Modelli prodotti e caratteristiche tecniche

Il primo dispositivo Raspberry Pi entrato in commercio risale al 2012 ed era disponibile in due versioni dotate rispettivamente di una memoria RAM pari a 256 MB e 512 MB. Da allora ne sono state prodotte

diverse versioni successive, di volta in volta migliorate nella loro componentistica di base e potenza di calcolo, nonché nella versatilità di utilizzo ed interazione con altri dispositivi elettronici.

Come si vedrà nel seguito, anche l'universo Raspberry dispone ormai di una radicata community di appassionati su scala internazionale e per tale ragione, parallelamente allo sviluppo delle schede SBC, sono state prodotte negli anni numerose apparecchiature elettroniche e kit di integrazione per l'utilizzo di Raspberry nelle più disparate applicazioni di tipo hobbistico.

Anche nel mondo accademico e industriale però, le potenzialità di Raspberry non sono passate inosservate e sono ormai molte le aziende produttrici di software per l'ingegneria ad offrire pacchetti informatici e strumenti di supporto per l'implementazione dei dispositivi Raspberry a scopi di sperimentazione, ricerca e sviluppo.

In **Tabella 3** è mostrata una sintesi dei modelli di Raspberry Pi e delle loro principali specifiche tecniche dal 2012 ad oggi.

Tabella 3 - Modelli di schede Raspberry Pi prodotti dalla nascita della fondazione [26]

	Raspberry Pi 1 Model A+	Raspberry Pi 1 Model A	Raspberry Pi 1 Model B	Raspberry Pi 1 Model B+	Raspberry Pi 2 Model B	Raspberry Pi 3 Model B	Raspberry Pi 3 Model B+	Raspberry Pi 3 Model A+	Raspberry Pi 4 Model B	Raspberry Pi Zero	Raspberry Pi Zero W (wireless)	Raspberry Pi Zero WH (wireless + porta GPIO pre-saldata)
Anno di rilascio	2014	2013	2012	2014	2015	2016	2018	2018	2019 / 2020*	2015	2017	2018
SoC	Broadcom BCM2835				Broadcom BCM2836	Broadcom BCM2837	Broadcom BCM2837B0	Broadcom BCM2837B0	Broadcom BCM2711	Broadcom BCM2835		
CPU	700 MHz single-core ARM1176JZF-S				900 MHz 32-bit quad-core ARM Cortex-A7	1.2 GHz 64-bit quad-core ARM Cortex-A53	1.4 GHz 64-bit quad-core ARM Cortex-A53 (ARMv8)	1.4 GHz 64-bit quad-core ARM Cortex-A53 (ARMv8)	4x Cortex-A72 1.5 GHz	1 GHz 32-bit single-core ARM1176JZF-S		
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, 1080p30 H.264 high-profile decoder ed encoder ^[2] 250 MHz per BCM2835 e BCM2836. 400 MHz dual core per BCM2837 (1080p60)							Broadcom VideoCore IV	Broadcom VideoCore VI @ 500 MHz			
Memoria (SDRAM)	256 MB (condivisa con la GPU)	256 o 512 MB (condivisa con la GPU)			1 GB (condivisa con la GPU) LPDDR2 (450 MHz)	1 GB (condivisa con la GPU) LPDDR2 (900 MHz)	1 GB (condivisa con la GPU) LPDDR2	512 MB (condivisa con la GPU) LPDDR2	1, 2, 4 o 8 GB (condivisa con la GPU)	512 MB (condivisa con la GPU)		
Porte USB 2.0	1 (attacco diretto sul chip)	2 (attraverso un hub USB integrato)	4 (attraverso un hub USB integrato)				4 (attraverso un hub USB integrato)	1	2	1 Micro-USB (direttamente dal chip BCM2835)		
Porte USB 3.0	0								2	0		
Input video	Connettore 15-pin MIPI Camera Interface (CSI), utilizzabile con fotocamere Raspberry Pi o Raspberry Pi NoIR									MIPI camera interface		
Output video	Un connettore RCA per il video composito ed una porta HDMI								2x micro-HDMI Connettore RCA	Mini-HDMI 1080p60, video composito via GPIO		

Tabella 3 - Continua

Input audio	Attraverso l' <u>interfaccia bus seriale I²S</u> (dalla revisione 2)									
Output audio	Jack da 3,5 mm, <u>HDMI</u> , attraverso l' <u>interfaccia bus seriale I²S</u> (dalla revisione 2)							<u>Mini-HDMI</u> , audio stereo tramite <u>GPIO</u>		
Memoria	<u>microSD</u>	<u>SD / MMC / SDI</u> <u>Q</u> card slot		<u>microSD</u>						
Collegamenti di rete	Nessuno		<u>Ethernet</u> 10/100 Mbit/s (<u>RJ-45</u>) ^[13]		Ethernet 10/100 Mb/s, <u>Wireless LAN</u> 802.11n, <u>Bluetooth</u> 4.1	Ethernet <u>Gigabit</u> (limitato comunque a 300 Mbps teorici), <u>Wireless LAN</u> 802.11b/g/n/ac, <u>Bluetooth</u> 4.2	<u>Wireless LAN</u> 802.11b/g/n/ac, <u>Bluetooth</u> 4.2	Ethernet 10/100/1000 Mbit/s, <u>Wireless LAN</u> 802.11b/g/n/ac dual band, <u>Bluetooth</u> 5.0	Nessuno	<u>Wireless LAN</u> 802.11n, <u>Bluetooth</u> 4.1
Connettori di I/O	40x <u>GPIO</u> , HAT	2x13 pin per <u>GPIO</u> , <u>SPI</u> , <u>I²C</u> , <u>UART</u> , +3,3 Volt, +5 Volt ^{[18][19]}		40x <u>GPIO</u>		40x <u>GPIO</u> e 4x adattatore <u>PoE</u>	40x <u>GPIO</u>	40x <u>GPIO</u>	40x <u>GPIO</u> "unpopulated header"	40x <u>GPIO</u>
Corrente (potenza) assorbita	200 mA (1 <u>W</u>)	300 mA (1.5 <u>W</u>)	700 mA (3,5 <u>W</u>)	600 mA (3,0 <u>W</u>)	800 mA (4,0 <u>W</u>)			600 mA (3W)	~160mA (0,8 <u>W</u>)	
Alimentazione	5 V via <u>MicroUSB</u> o <u>GPIO</u>					5V via <u>MicroUSB</u> , oppure via <u>GPIO</u> , oppure via <u>PoE</u> (con apposito HAT)	5V via <u>MicroUSB</u> , oppure via <u>GPIO</u>	5V via <u>USB-B-C</u> o via <u>GPIO</u>		
Dimensioni	65 mm x 54 mm	PCB: [(85,60 mm × 53,98 mm Tot: [(85,60 mm x 56 mm)					65 mm x 54 mm	85 mm x 56 mm	65 mm × 30 mm × 5 mm	
Sistemi operativi supportati	<u>Debian GNU/Linux</u> , <u>Fedora</u> , <u>Arch Linux</u> , <u>Gentoo</u> , <u>FreeBSD</u> e <u>RISC OS</u> (shared source)				Debian GNU/Linux, Fedora, Arch Linux, Gentoo, FreeBSD e RISC OS (shared source), <u>Windows 10 IoT</u> (dal Pi 2 Model B), <u>Android Nougat</u> (Pi 3 Model B) , RetroPie					

Per meglio comprendere le funzionalità di tali dispositivi è possibile servirsi di [Figura 18](#) e [Figura 19](#), che ritrae, a titolo esemplificativo, i modelli Raspberry Pi 1B e Raspberry Pi 4B.

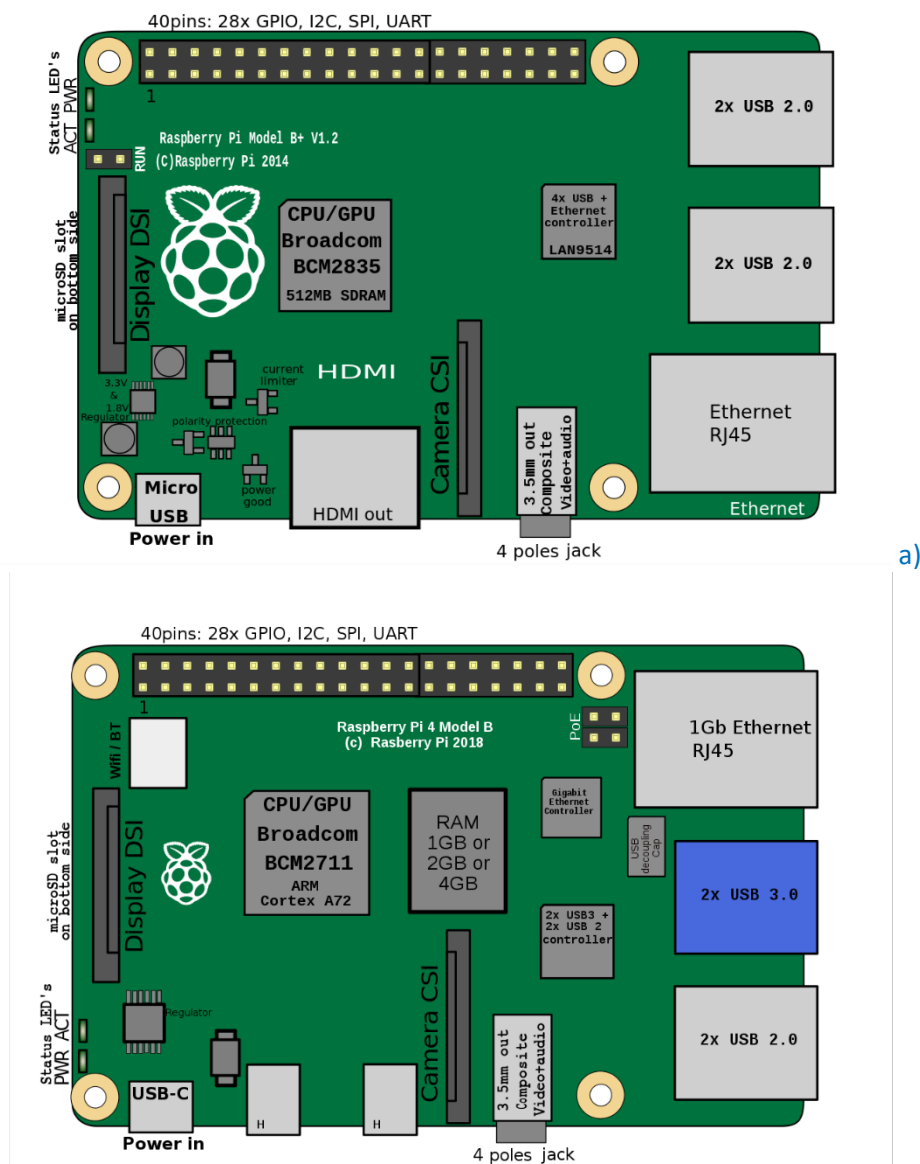
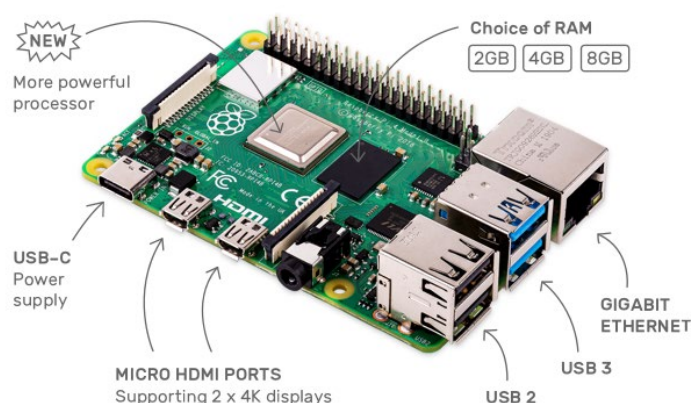


Figura 18 – riproduzione schematizzata dei modelli: a) Raspberry Pi 1B+ e b) Raspberry Pi 4B [26]



Come è possibile notare da **Figura 18**, i dispositivi Raspberry Pi dispongono generalmente di diversi connettori e porte del tutto simili a quelli presenti nei più comuni PC desktop e portatili. Per la maggior parte dei modelli sono infatti presenti alcune porte USB (di tecnologia 2.0, 3.0, micro o di tipo C a seconda del modello selezionato), una o più uscite video HDMI o micro-HDMI, una porta di rete Ethernet, un connettore audio jack da 3.5 mm ed uno slot per l'archiviazione dei dati attraverso schede di memoria di tipo SD (Secure Digital).

Pur trattandosi come già detto di un vero e proprio computer compatto, il "motore" del Raspberry Pi è completamente racchiuso all'interno di un unico chip che, nel caso del modello 1B corrisponde al *Broadcom BCM2835*. Tale componente è denominato "System On Chip" (SOC) in quanto consente di integrare al suo interno oltre che il processore anche tutte le altre periferiche che nella scheda madre di un PC tradizionale verrebbero montate esternamente su altri chip.

Per quanto riguarda il processore, esso è basato sull'architettura ARM, tipicamente impiegata per la realizzazione di numerose tipologie di dispositivi mobili, quali ad esempio telefoni cellulari e tablet, e per tale ragione le schede Raspberry risultano molto più simili dimensionalmente ad un moderno smartphone che ad un personal computer. L'architettura ARM (Advanced RISC Machine) racchiude al suo interno quella categoria di microprocessori il cui numero di istruzioni caratteristiche risulta ridotto rispetto a quello tipico di un computer tradizionale, generalmente basato su un'architettura x86 Intel o AMD. La sigla RISC sta infatti per "Reduced Instruction Set Computer" il che consente di eseguire i set di istruzioni in tempi minori rispetto al caso di un normale PC, per il quale si parla di CISC, ovvero "Complex Instruction Set Computer".

Insieme al processore, l'altro componente maggiormente responsabile delle prestazioni della scheda Raspberry è certamente la memoria RAM (Random Access Memory), ovvero la memoria volatile ad accesso casuale, che per i modelli più recenti può arrivare fino ad 8 Gigabyte di dimensione (il primo Raspberry Pi 1 disponeva di 256 MB di memoria RAM).

Per l'archiviazione di dati invece, Raspberry non dispone di memorie fisiche integrate nel chip ma prevede, come già accennato, un apposito slot per l'inserimento di una scheda SD (per alcuni modelli fino a 256 GB di capienza massima), sulla quale possono essere installati il sistema operativo ed altre applicazioni, nonché essere salvati i dati non volatili (**Figura 20**).

Dal punto di vista della connettività, le schede Raspberry offrono numerose opzioni di personalizzazione grazie alla presenza di una o più porte USB e dell'uscita Ethernet RJ45, tramite cui è possibile realizzare un collegamento alla rete. In questo modo infatti il dispositivo può essere messo in comunicazione con un router in modalità DHCP o direttamente ad un PC, realizzando quindi una rete LAN. Per alcuni modelli il collegamento alla rete può essere effettuato anche in modalità wireless, disponendo questi di una scheda di rete abilitata alla connettività Wi-Fi, ma più in generale per qualsiasi dispositivo è possibile implementare

tale funzionalità servendosi di opportune chiavette di estensione wireless-USB, ormai molto diffuse in commercio.

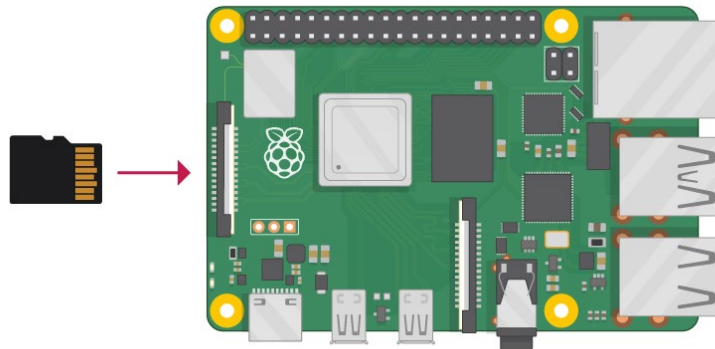


Figura 20 - slot per l'inserimento della scheda SD [3]

Un'ultima importante caratteristica hardware di Raspberry Pi riguarda la presenza di una periferica di basso livello, costituita da una serie di pin GPIO (General Purpose Input/Output) collocati ordinatamente lungo il bordo superiore del circuito stampato. Grazie a tali pin è possibile integrare Raspberry all'interno di circuiti elettronici esterni o permetterne l'interazione con altri dispositivi come Arduino.

È di seguito mostrata in **Figura 21** la pin map relativa ai modelli più recenti, dotate di un'intestazione a 40 pin (le prime versioni disponevano di un'intestazione ridotta a 26 pin).

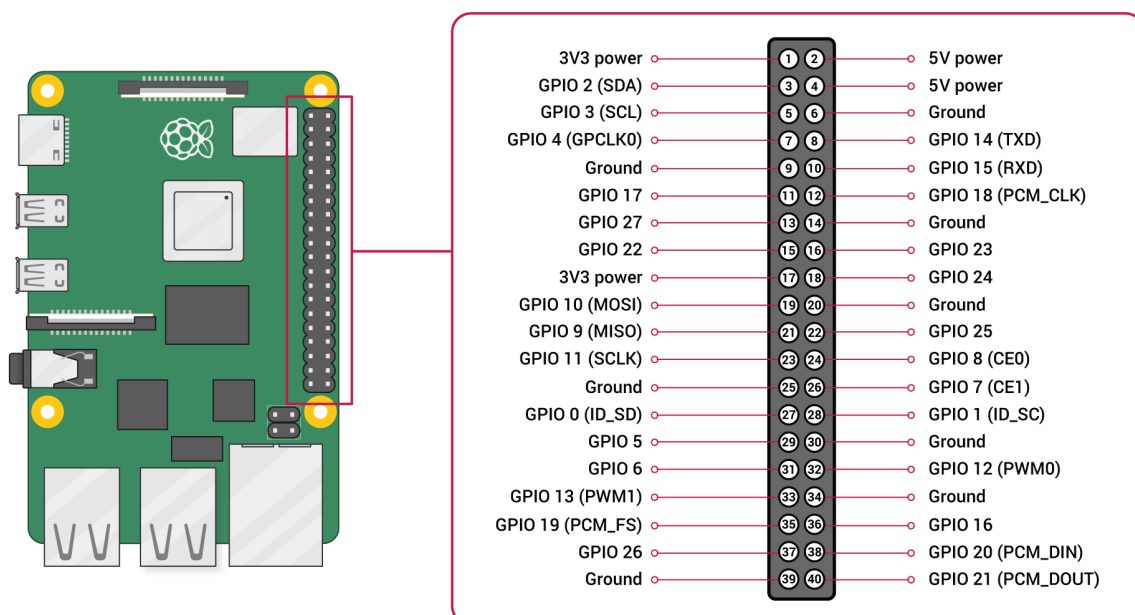


Figura 21 - pin map Raspberry Pi [3]

L'assegnazione ingresso/uscita dei pin GPIO può essere effettuata direttamente lato software attraverso numerosi linguaggi di programmazione ed è inoltre possibile servirsi di un'ampia varietà di librerie e funzioni per controllare i singoli pin, quali ad esempio PWM, SPI, I2C e seriale. Per quanto riguarda il voltaggio, la scheda è provvista di due pin a 5V e due 3V3, oltre che ad una serie di pin a 0V non configurabili. Un maggiore dettaglio circa la ripartizione delle tensioni sui pin è fornito dalla **Figura 22**.

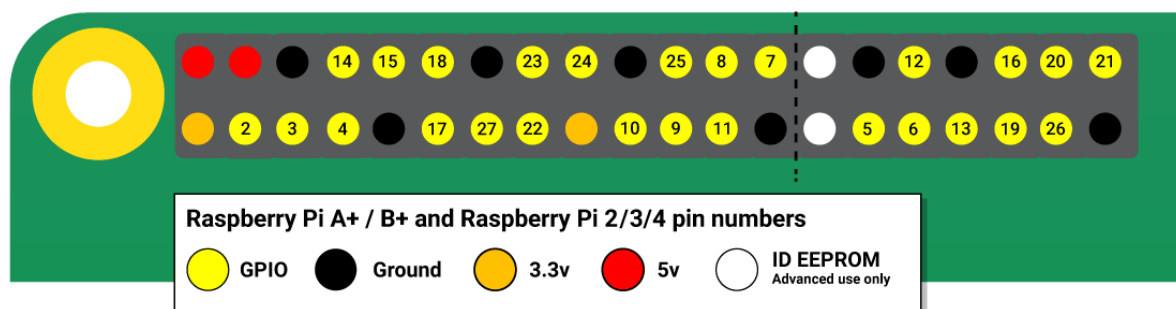


Figura 22 - tensioni pin GPIO [3]

Per quel che riguarda l'alimentazione della scheda è bene precisare che il Raspberry Pi non può essere alimentato direttamente da un PC ed è quindi necessario un alimentatore apposito (**Figura 23**) che sia in grado di garantire una corrente erogata minima di 1 A (possono essere necessari fino a 2 A nel caso di un collegamento a numerose periferiche). Questa necessità non costituisce tuttavia un grosso problema in quanto è possibile servirsi delle porte micro-USB o USB-C per collegare il dispositivo ad un comune caricabatterie per smartphone, una volta verificata la compatibilità dei parametri elettrici con la scheda.



Figura 23 – Alimentatore dedicato per Raspberry Pi 4 [3]

1.2.2 Sistema operativo e applicazioni

Secondo quanto visto finora, le caratteristiche hardware del Raspberry Pi hanno avuto nel tempo una rapida evoluzione, molto più di quanto non sia accaduto per i PC, e sono passate ad avere oggi specifiche tecniche decisamente prestazionali, garantendo alla piccola scheda potenze di calcolo e funzionalità degne di alcuni computer. Tuttavia, è bene ricordare che rispetto alla maggior parte dei moderni PC desktop e

laptop Raspberry Pi dispone di risorse certamente più limitate ed anche l'architettura del suo processore ARM, per quanto ottimizzata ed evoluta, non è compatibile con i tradizionali sistemi operativi normalmente in uso, quali Windows o Mac Os. Per tali ragioni, alla nascita del Raspberry Pi, si è scelto di mettere a punto un sistema operativo dedicato, che derivasse da un sistema di tipo "open source" basato su Linux, in modo tale da poterlo adattare per sfruttare al meglio le caratteristiche hardware della scheda.

Venne quindi sviluppato Raspbian (**Figura 24**), un sistema operativo creato apposta per il Raspberry Pi ed ufficialmente consigliato dalla Raspberry Foundation, la cui denominazione deriva dall'incontro tra il nome della stessa scheda e quello di Debian, una distribuzione Linux di grande successo sulla quale è stato basato anche Ubuntu. Raspbian rappresenta ancora oggi la principale scelta per molti utenti, grazie soprattutto alla grande disponibilità di strumenti e funzionalità previste, consolidate dalla presenza di oltre 35.000 pacchetti applicativi per il miglioramento della stabilità e delle prestazioni del software.



Figura 24 - logo Raspbian [18]

Oltre a Raspbian però sono ormai molti i sistemi operativi altrettanto adeguati che il Raspberry Pi è in grado di eseguire, quali ad esempio Ubuntu MATE, DietPi, RISC OS, Arch Linux ARM, OSMC (**Figura 25**) e molti altri, e la possibilità di montare il sistema operativo mediante una scheda di memoria SD consente il vantaggio di poter sperimentare diverse distribuzioni tra quelle citate semplicemente dotandosi di un maggior numero di SD card.



Figura 25 – loghi di alcuni dei sistemi operativi non ufficiali per Raspberry Pi

Ciascuno dei sistemi operativi eseguibili su Raspberry consente un'ampia possibilità di personalizzazione ed è pensato specificatamente per migliorarne un particolare aspetto operativo come la semplicità d'impiego per utenti alle prime armi, la leggerezza e la fluidità di esecuzione o la versatilità di utilizzo per la gestione di contenuti multimediali.

Una volta installato il sistema operativo, le funzionalità del Raspberry Pi possono certamente essere sfruttate in moltissimi modi, ma lo scopo fondante del progetto secondo la Raspberry Foundation fu proprio quello di promuovere e rendere accessibile al mondo dell'istruzione l'apprendimento dei linguaggi di programmazione, primo su tutti Python, al quale la piccola scheda elettronica deve parte del suo nome (la sigla Pi sta per Python). Oltre che per svolgere le ordinarie operazioni di un PC desktop, infatti, Raspberry Pi può essere utilizzato per progetti di informatica ed elettronica particolarmente complessi ed approfonditi, dall'amministrazione di reti e server alle applicazioni di robotica ed automazione, nonché per attività di misurazione ed analisi dei dati in fase di ricerca e sviluppo.

Nel prossimo paragrafo verranno brevemente descritti gli impieghi dei dispositivi Arduino e Raspberry nell'ambito della ricerca accademica e saranno analizzate le possibilità di implementazione degli stessi ai fini della sperimentazione in oggetto.

1.3 Soluzione proposta per il banco prova di emulazione

Nel presentare la soluzione proposta per la sperimentazione è bene ricordare che l'attività in oggetto di questa relazione prevede la definizione e la successiva installazione, presso il Laboratorio di Meccanica dell'Università di Bologna, delle attrezzature necessarie alla costituzione di un banco prova per la programmazione di centraline di controllo volte alla gestione ottimizzata dei flussi energetici nel contesto delle microreti energetiche e dei sistemi ad idrogeno.

Tale banco prova, basato sull'integrazione tra i componenti fisici della microrete ed alcuni dispositivi di emulazione, sarà poi impiegato durante la successiva annualità del progetto per lo studio delle strategie di gestione ottimizzate e per la programmazione dell'unità di controllo finale.

Lo studio bibliografico circa le possibilità di configurazione del sistema in ottica di emulazione e tecniche HIL/SIL nonché di riduzione dei costi e dei tempi della sperimentazione ha permesso di individuare alcune possibilità di integrazione tra i dispositivi Arduino e Raspberry trattati sinora e la microrete allestita presso il laboratorio. È stato infatti possibile definire, come approccio iniziale durante la corrente annualità del progetto, l'architettura del sistema elettronico da allestire e la sua messa in opera ai fini dell'emulazione della microrete.

In **Figura 26** è mostrato uno schema rappresentativo della soluzione proposta.

La soluzione proposta si basa su un approccio di tipo MIL ("Model in The Loop") e garantirà quindi di testare il software di controllo della microrete, progettato in ambiente Simulink, mediante la diretta interazione tra il PC ed alcune schede Arduino mega 2560 atte a riprodurre, unicamente a mezzo di un modello di calcolo sviluppato in codice dall'IDE di Arduino, il comportamento di ciascun componente fisico della rete. In questo modo è possibile sollecitare l'unità di controllo con step temporali molto ridotti, emulando di fatto le condizioni operative del sistema senza la necessità di operare con i reali sistemi fisici, bensì rimanendo in ambiente virtuale.

Un'immagine del banco prova è fornita in **Figura 27**.

L'integrazione dell'architettura corrente con i dispositivi di misura ed acquisizione e con i componenti fisici della microrete porrà le basi per la successiva annualità del progetto, rendendo operativo a tutti gli effetti l'apparato di emulazione HIL ("Hardware in The Loop") per il collaudo dell'unità di controllo elettronica.

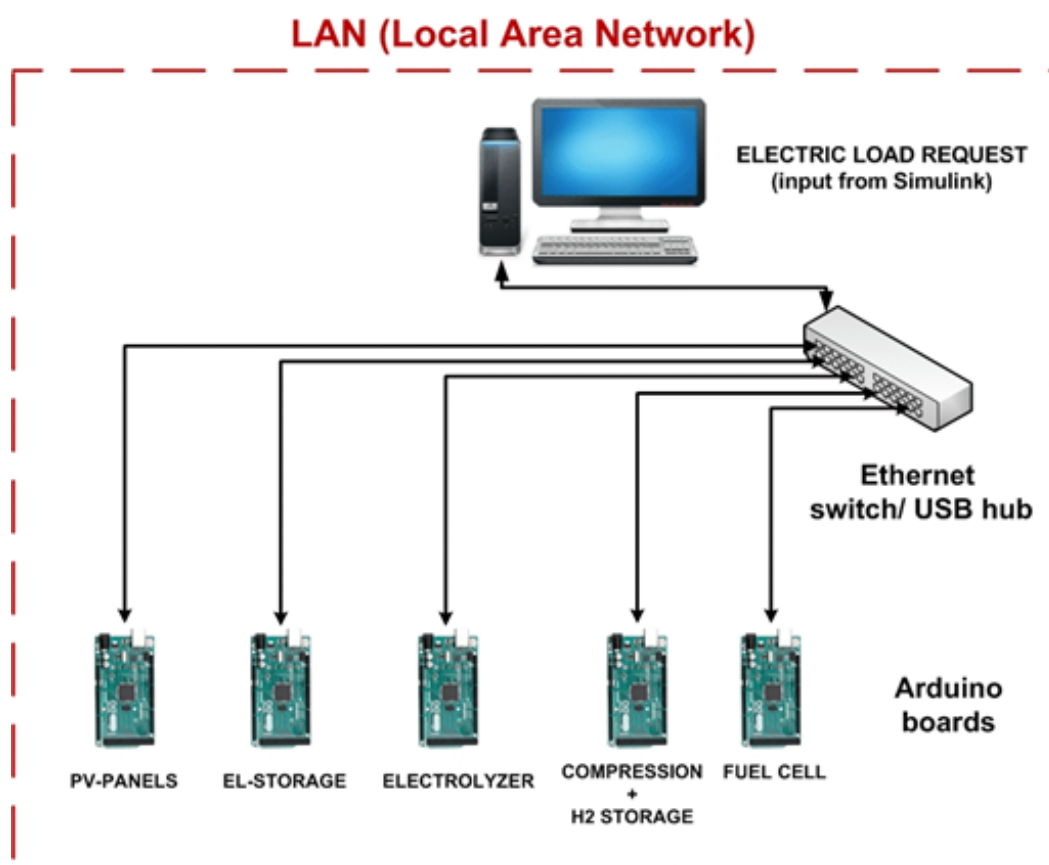


Figura 26 - Architettura del sistema di emulazione

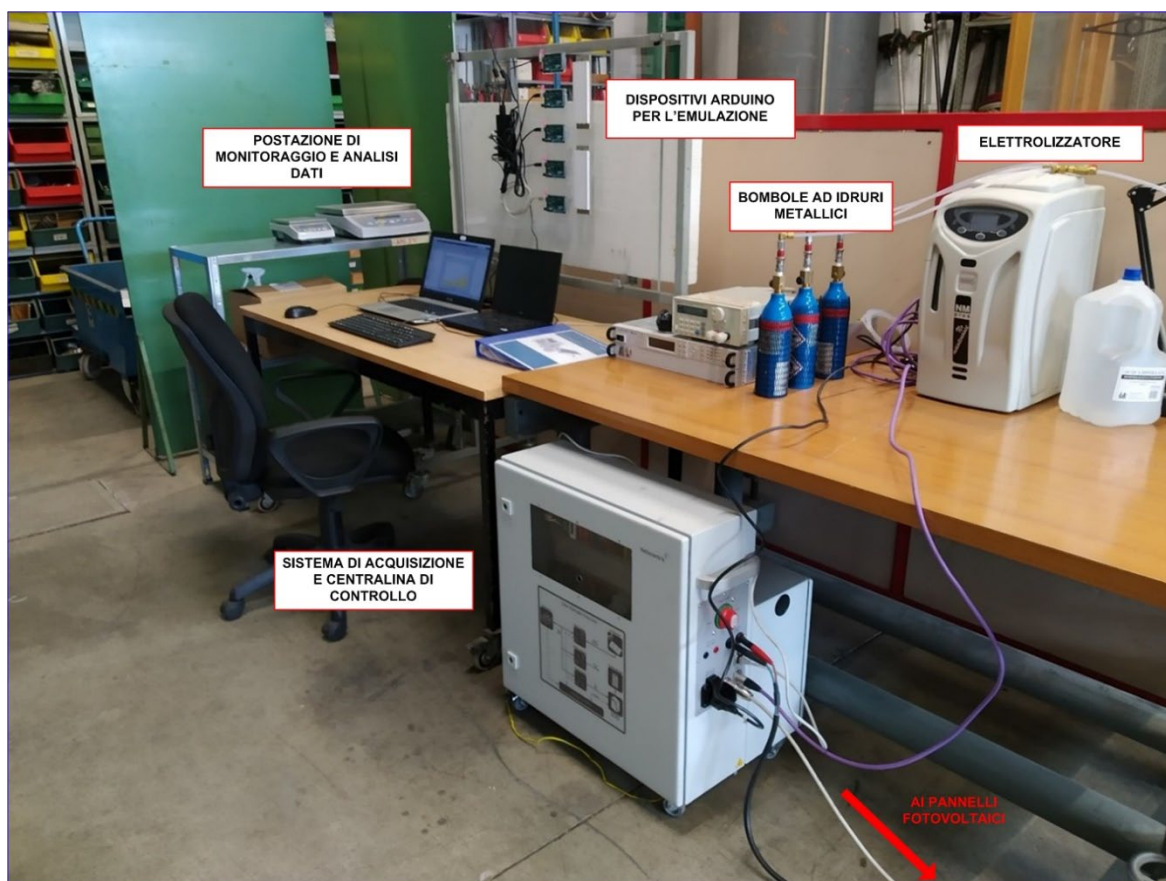


Figura 27 - banco prova di emulazione

2 Costituzione del Banco di prova per l'emulazione di un sistema di produzione ed accumulo di idrogeno ed energia elettrica da fonte rinnovabile non programmabile

In questa sezione si porrà l'attenzione sull'apparato elettronico oggetto della sperimentazione, con ricorrenti riferimenti al sistema energetico di cui si è voluto emulare e studiare il comportamento nelle differenti situazioni analizzate.

Si procederà nell'ordine con la presentazione dell'impianto di generazione elettrica e termica denominato S.A.P.I.E.N.T.E., attualmente installato presso il Centro Ricerche ENEA della Casaccia [23, 24], e delle attrezzature presenti nel laboratorio di Meccanica dell'Alma Mater Studiorum - Università di Bologna, utilizzate per la realizzazione del banco prova di emulazione [25].

Successivamente verranno illustrate le modalità tramite cui si è scelto di procedere per la costituzione del banco prova di emulazione e saranno descritti i modelli computazionali realizzati per ciascun componente d'impianto coinvolto nella sperimentazione.

In ultimo verranno analizzate e commentate le simulazioni effettuate, mettendo in luce le potenzialità e le criticità della metodologia adottata per lo studio in oggetto.

2.1 Il modello S.A.P.I.E.N.T.E e l'impianto pilota del laboratorio di meccanica

Come già accennato in precedenza, l'attività di emulazione oggetto di studio è stata finalizzata all'integrazione di una sezione di produzione, accumulo e conversione di idrogeno all'interno del sistema di produzione termica ed elettrica da fonte rinnovabile denominato S.A.P.I.E.N.T.E. (Sistema di Accumulo e Produzione Integrata di ENergia Termica ed Elettrica) attualmente in fase di sperimentazione presso il Centro Ricerche Enea della Casaccia.

Lo scopo di tale integrazione è la valutazione, ai fini di un aumento delle prestazioni d'impianto nel lungo periodo, dell'impiego di idrogeno come strategia di accumulo stagionale dell'energia prodotta in eccesso dai generatori non programmabili per applicazioni di tipo civile/residenziale.

L'architettura attuale del sistema è composta da:

- a) Un'unità di produzione dell'energia elettrica da moduli fotovoltaici in grado di sviluppare una potenza di picco complessiva di circa 11.4 kW;
- b) Un dispositivo di accumulo elettrochimico a litio di capacità massima pari a 16 kWh;
- c) Una pompa di calore di tipo aria/acqua ad inverter di potenza termica pari a 33 kW;
- d) Due accumuli termici inerziali da 1.5 m³;
- e) Un accumulo termico per l'acqua calda sanitaria da 1 m³;
- f) Due dry cooler (con potenza complessivamente pari a 140 kW) come dispositivo di emulazione dei fabbisogni termici.

Uno schema del modello S.A.P.I.E.N.T.E. nella configurazione attuale è rappresentato in **Figura 28**, mentre il layout modificato per l'introduzione del gruppo ad idrogeno è mostrato in **Figura 29**.

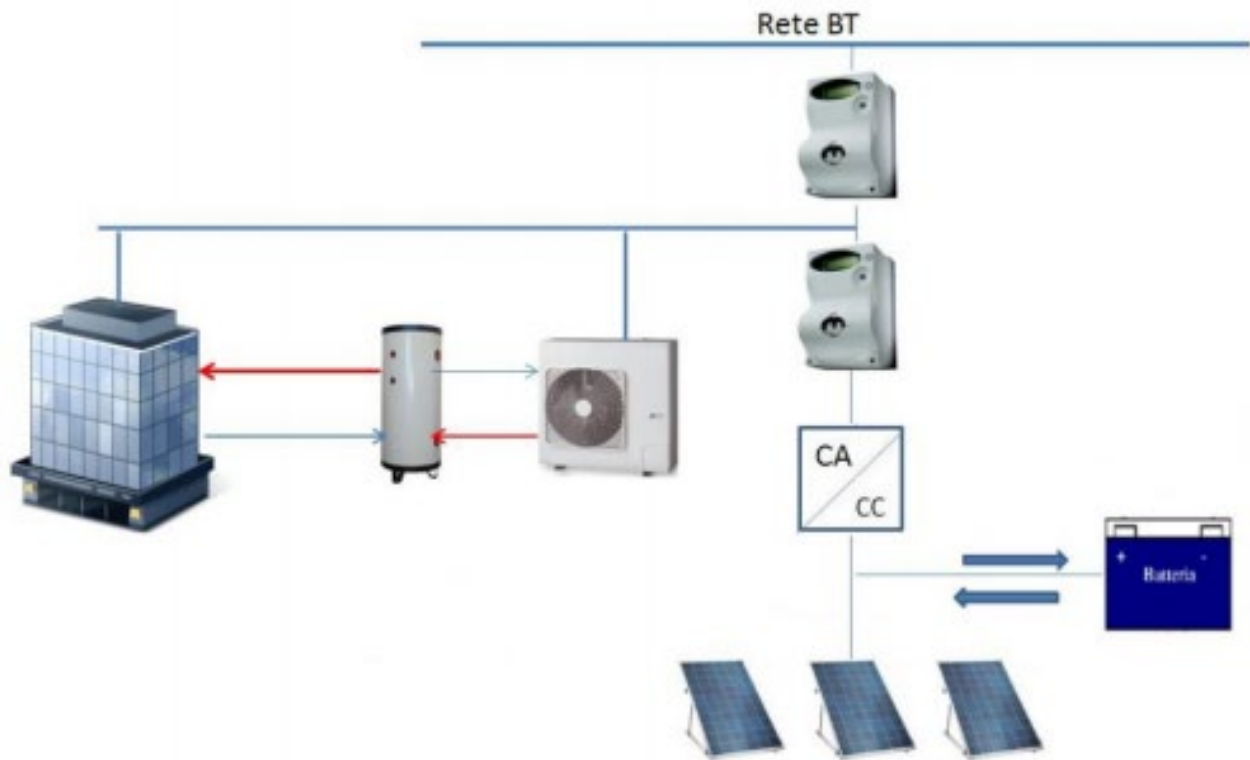


Figura 28 - Schema del modello S.A.P.I.E.N.T.E. installato presso il Centro ENEA di Casaccia [24]

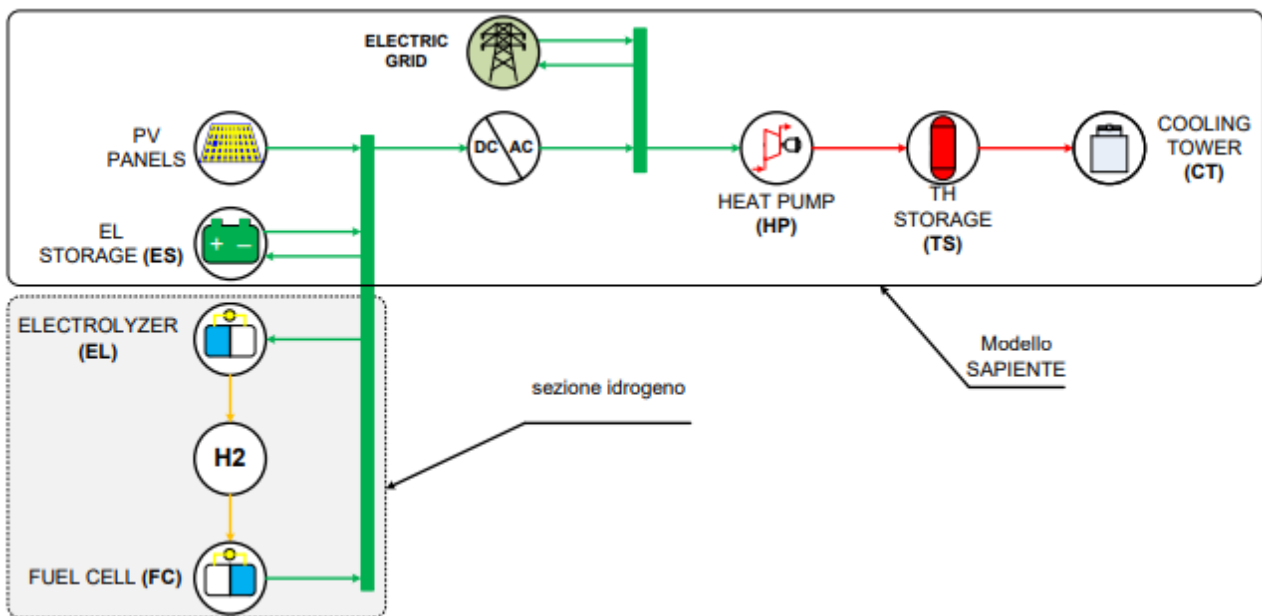


Figura 29 - Architettura aggiornata dell'impianto realizzato presso il Centro ENEA di Casaccia

Per ciò che concerne invece l'architettura di supporto alla sperimentazione, è stata allestita presso il Laboratorio Di Meccanica dell'Alma Mater Studiorum – Università Di Bologna una microrete energetica caratterizzata dalla presenza di un sistema per la produzione e l'accumulo di idrogeno abbinato a due generatori elettrici di tipo fotovoltaico. I componenti principali di tale sistema sono:

- due generatori fotovoltaici in silicio policristallino, connessi in parallelo tra loro, per un totale di 440 W di potenza elettrica complessiva (in condizioni STC: radiazione solare pari a 1000 W/m², temperatura celle pari a 25 °C e spettro solare AM 1.5) al 15% circa di efficienza [27];
- due dispositivi di accumulo elettrochimico (tecnologia AGM) connessi in serie per una capacità elettrica totale di 110 Ah ad una tensione nominale di 24 V (55 Ah a 12 V per elemento) [28];
- un regolatore della carica di tipo PWM, per la gestione dell'interconnessione tra pannelli fotovoltaici e batterie [29];
- un convertitore DC/DC di tipo Meanwell SD25B-12 per ridurre la tensione di output ed alimentare un eventuale carico in corrente continua [30];
- un inverter DC/AC di tipo Meanwell TS-700-224B [30];
- un elettrolizzatore di tipo PEM, capace di produrre 30L/ora di idrogeno a 10.5 bar di pressione;
- tre bombole ad idruri metallici, da 200 SL ciascuna a 10 bar, per lo stoccaggio dell'idrogeno prodotto [31].

In **Figura 30** è riportato uno schema rappresentativo del sistema descritto mentre in **Figura 31** sono rappresentati i componenti della microrete

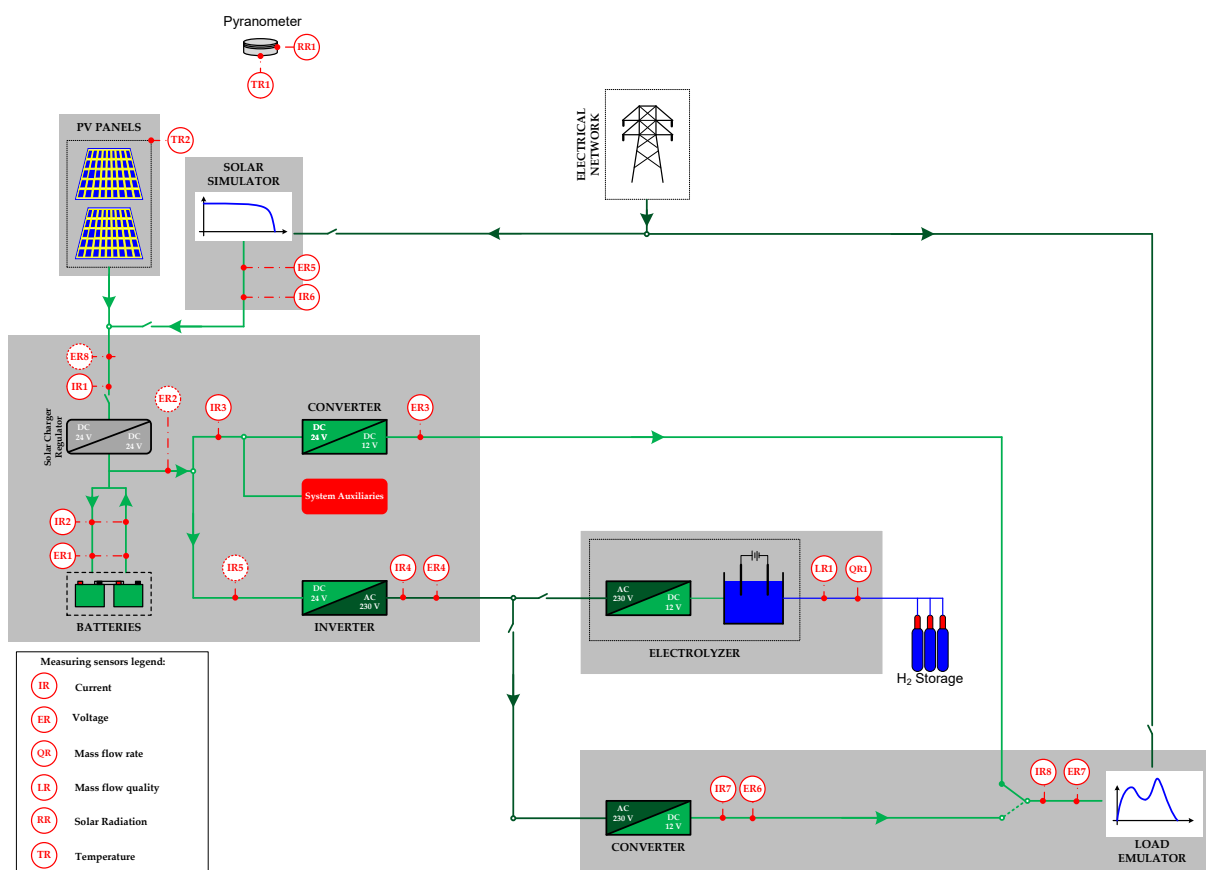


Figura 30 - Layout della microrete di generazione ed accumulo di idrogeno da generazione rinnovabile non programmabile

Il laboratorio dispone inoltre di una linea ad idrogeno, alimentata attraverso due bombole da 55 L l'una e 200 bar di pressione massima, e di una fuel cell commerciale di tecnologia PEM [32] (non connesse alla microrete di **Figura 30**). La fuel cell, progettata per operare a basse temperature (attorno ai 65 °C a pieno

carico), è in grado di produrre una potenza elettrica nominale pari a 2 kW ad una tensione di 28.8 V ed una corrente elettrica massima di 75 A, a fronte di un consumo massimo di idrogeno dichiarato di 26 L/min.

Alcuni dei principali componenti ed attrezzature del laboratorio sono mostrati in **Figura 31**.



pannelli fotovoltaici



elettrolizzatore tipo PEM



bombole ad idruri metallici



Batterie al piombo AGM



unità di controllo e regolazione



simulatore fotovoltaico



piranometro



carico elettronico

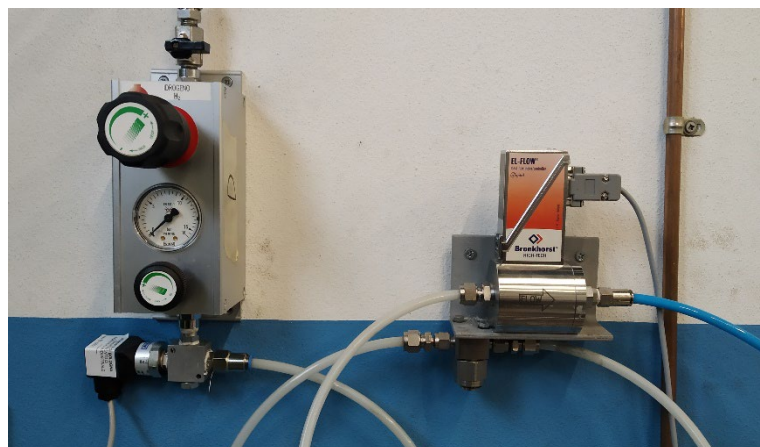


fuel cell tipo PEM

Figura 31 – Componenti ed attrezzature del Laboratorio di Microreti di Generazione ed Accumulo dell'Alma Mater Studiorum – Università di Bologna



Bombole di alimentazione per la linea ad idrogeno



Riduttore di bassa pressione e misuratore di portata con sistema di filtraggio per l'idrogeno

Figura 31 – Continua

2.2 Test di regolazione sulla microrete installata presso il Laboratorio di Meccanica dell'Università di Bologna

Si ritiene utile riportare in questa sede alcune delle prove sperimentali condotte sulla microrete precedentemente descritta; il paragrafo successivo è invece dedicato alla descrizione dei test condotti sulla fuel cell.

Per ragioni di sintesi, e data la rappresentatività dei dati ottenuti, si ritiene opportuno riportare – a titolo di esempio – solo alcuni dei test svolti nel luglio 2020. La finalità delle prove in oggetto è stata quella di verificare la possibilità di regolare il sistema favorendo l'accumulo di energia elettrica prevalentemente verso la batteria o, in alternativa, verso la produzione di idrogeno.

Più in dettaglio si presenterà:

- una prima prova (denominata test #1, nel seguito, da **Figura 32** a **Figura 39**) in cui la produzione di potenza elettrica dei pannelli fotovoltaici è prevalentemente dedicata alla carica della batteria e in modo limitato alla produzione di idrogeno;
- una seconda prova (test #2, nel seguito, da **Figura 40** a **Figura 47**) in cui si è osservato il sistema imponendo un carico variabile all'elettrolizzatore al fine di testarne la rapidità di risposta alla regolazione.

Nelle figure che seguono si riportano alcuni dei parametri monitorati, tra i quali:

- Irraggiamento solare [W/m^2];
- Potenza elettrica prodotta da fotovoltaico [W];
- Tensione moduli fotovoltaici [V];
- Corrente elettrica generata da fotovoltaico [A];

- Tensione delle batterie [V];
- Corrente elettrica in ingresso/uscita dalle batterie [A];
- Produzione di idrogeno [kg/timestep].

In **Figura 32** è riportato l'irraggiamento solare durante il test #1. Si osservano valori particolarmente elevati (pari a circa 1000 W/m²) con alcuni repentini tracolli causati da fenomeni di ombreggiamento momentanei.

L'andamento della potenza prodotta (**Figura 33**), della corrente in uscita dai pannelli (**Figura 34**) e della tensione (**Figura 35**) risulta coerente con la strategia di regolazione (*pulse with modulation*) adottata per la regolazione del sistema integrato fotovoltaico-batteria. Ne segue, come atteso, un andamento del rendimento di conversione dei pannelli fotovoltaici (si veda **Figura 36**) mai superiore al 5.0%.

La corrente in ingresso e in uscita dalla batteria (**Figura 37**) e la corrispondente tensione (**Figura 38**) confermano che per tutta la durata del test condotto lo stato della batteria è prevalentemente in ricarica. Infine, in **Figura 39**, si presenta la produzione di idrogeno monitorata durante il test e limitata al carico minimo dell'elettrolizzatore.

Per quanto riguarda il secondo test effettuato, in **Figura 40**, è presentato l'irraggiamento solare misurato durante la prova; si osservano valori leggermente inferiori rispetto a quanto riscontrato nel test #1. Anche in questo caso si è mantenuta la regolazione del sistema fotovoltaico-batteria in regime di *pulse with modulation*, come confermato dagli andamenti della potenza prodotta (**Figura 41**), della corrente in uscita dai pannelli (**Figura 42**) e della tensione (**Figura 43**) dei pannelli fotovoltaici. I valori riscontrati del rendimento della produzione fotovoltaica (**Figura 44**) sono pertanto in linea con quanto già verificato nel test #1. Come previsto nel test #2, i valori di corrente (**Figura 45**) e tensione (**Figura 46**) della batteria segnalano un prevalente stato di scarica della batteria a favore della produzione di idrogeno (**Figura 47**) imposta crescente dal carico minimo al massimo possibile.

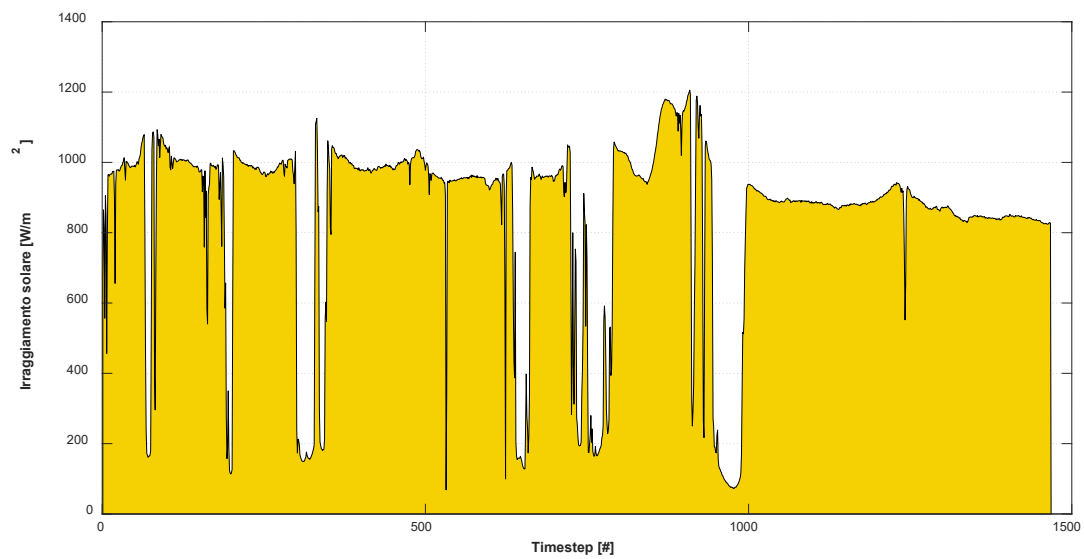


Figura 32 – Irraggiamento solare (test #1 – luglio 2020)

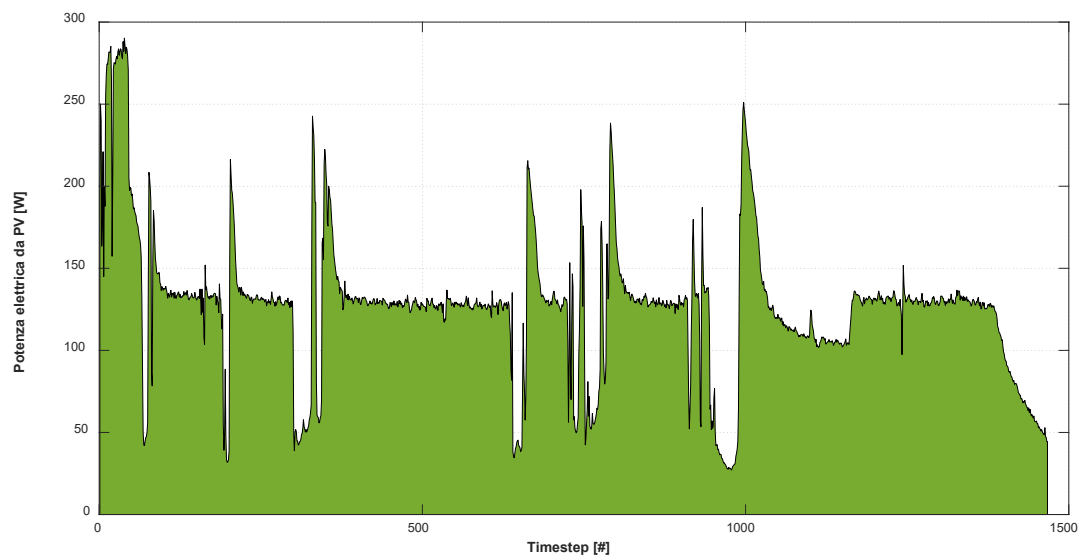


Figura 33 – Potenza prodotta dai pannelli fotovoltaici (test #1 – luglio 2020)

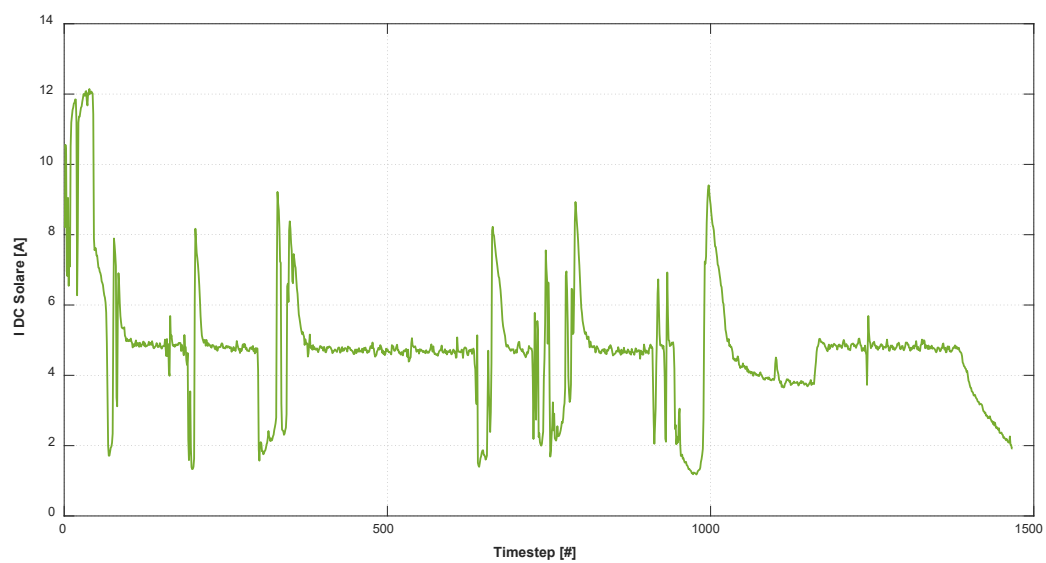


Figura 34 – Corrente prodotta dai pannelli fotovoltaici (test #1 – luglio 2020)

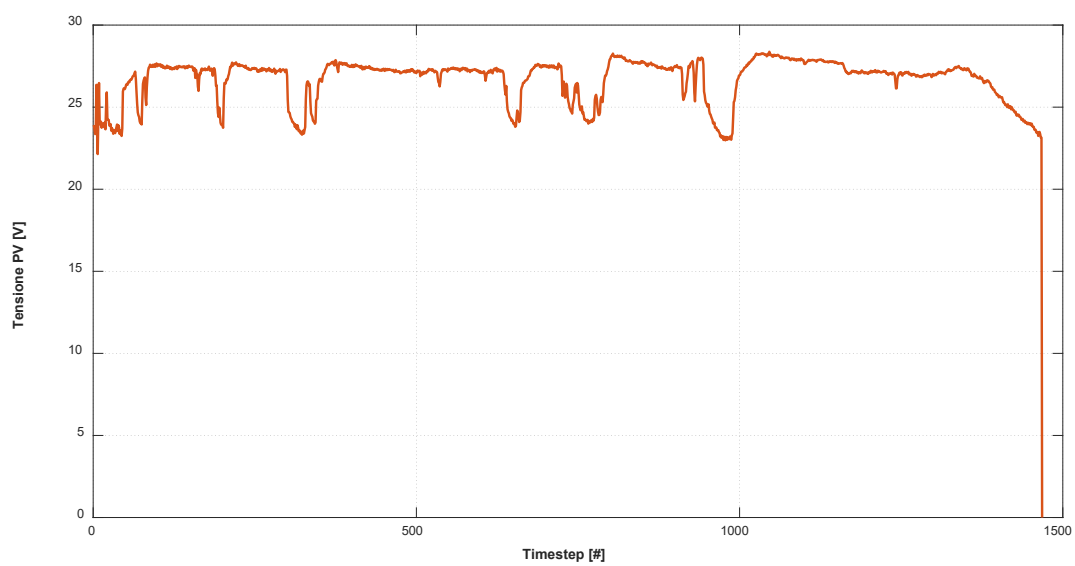


Figura 35 – Tensione dei pannelli fotovoltaici (test #1 – luglio 2020)

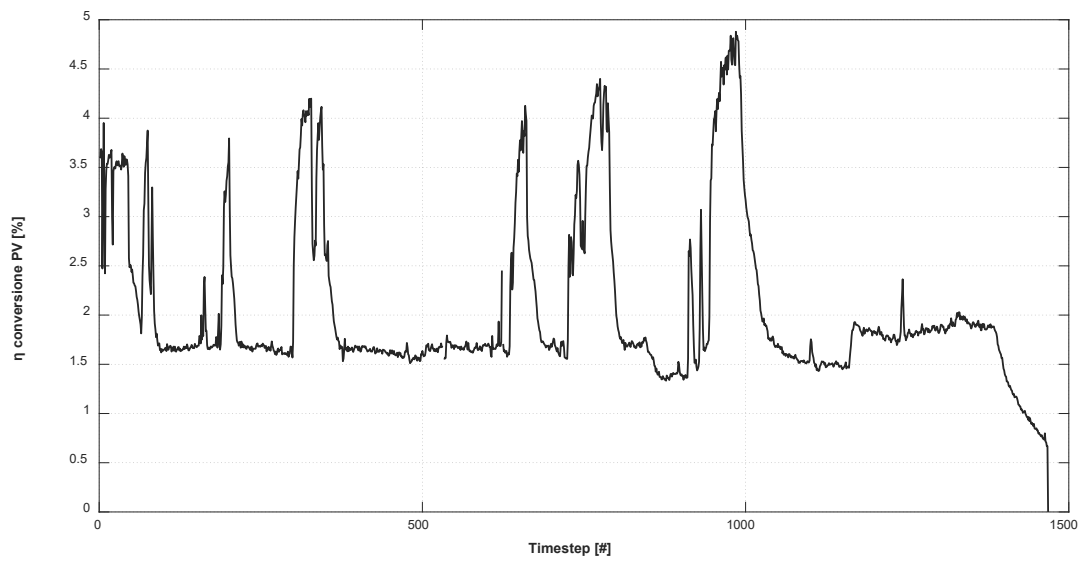


Figura 36 – Rendimento di conversione dei pannelli fotovoltaici (test #1 – luglio 2020)

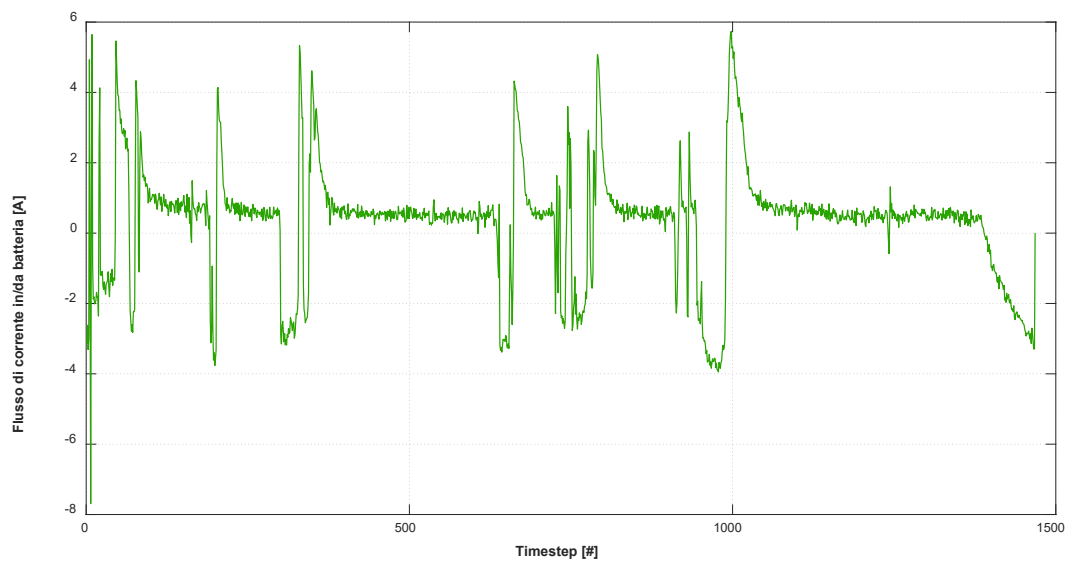


Figura 37 – Corrente in ingresso o in uscita dalla batteria (test #1 – luglio 2020)

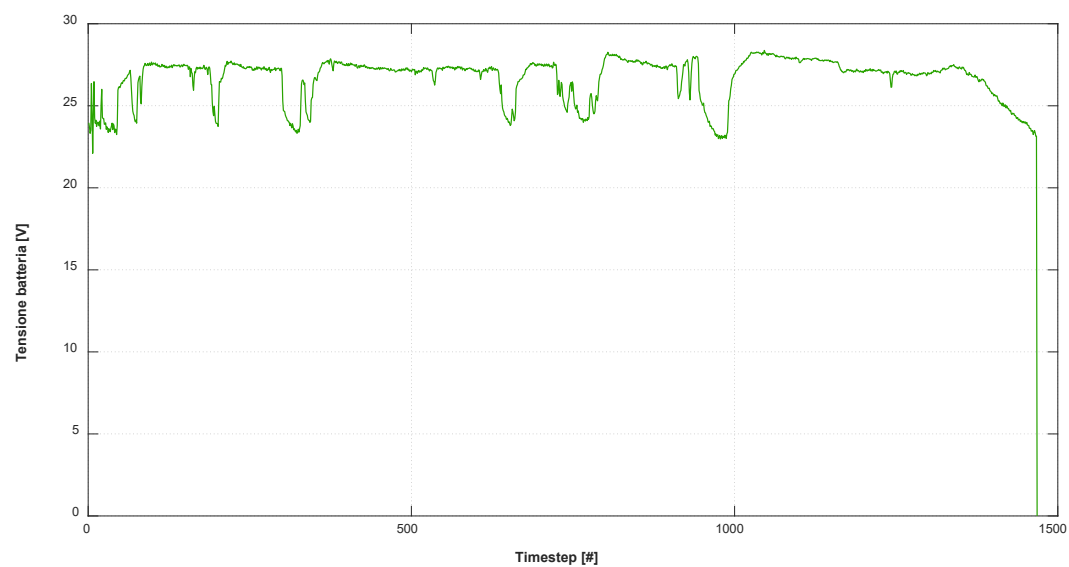


Figura 38 – Tensione della batteria (test #1 – luglio 2020)

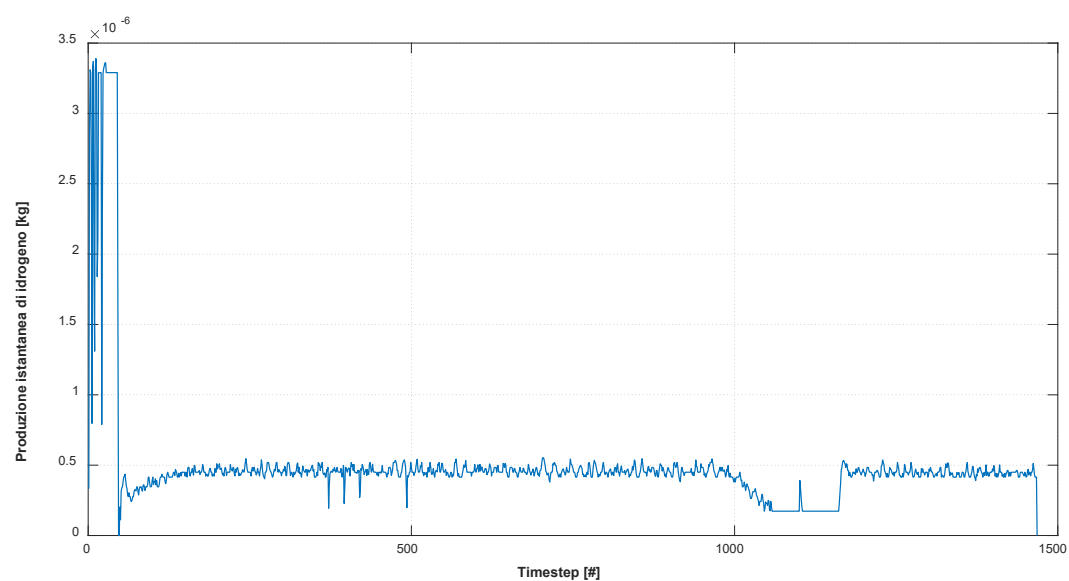


Figura 39 – Idrogeno prodotto (test #1 – luglio 2020)

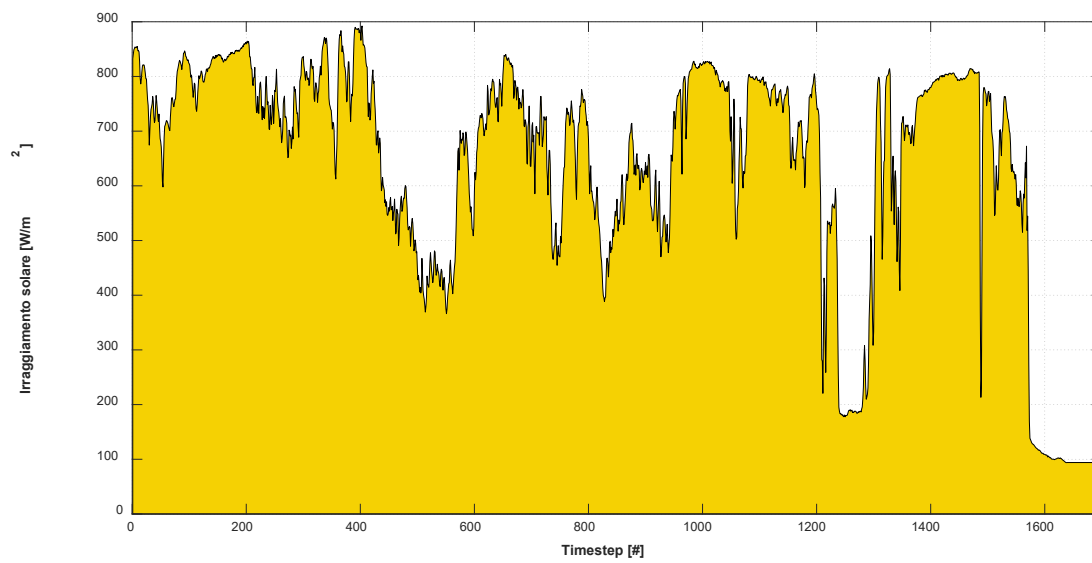


Figura 40 – Irraggiamento solare (test #2 – luglio 2020)

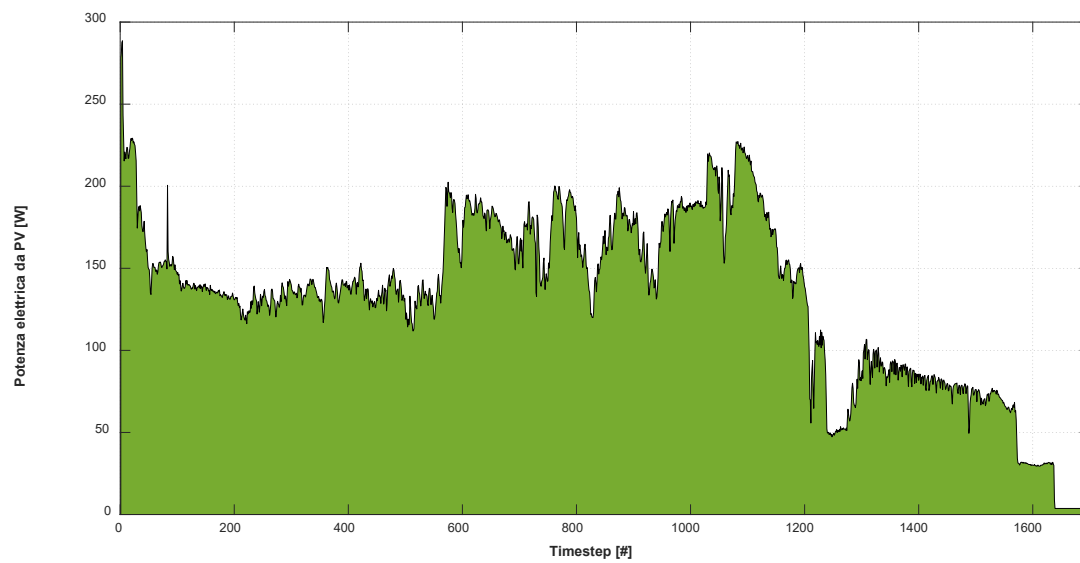


Figura 41 – Potenza prodotta dai pannelli fotovoltaici (test #2 – luglio 2020)

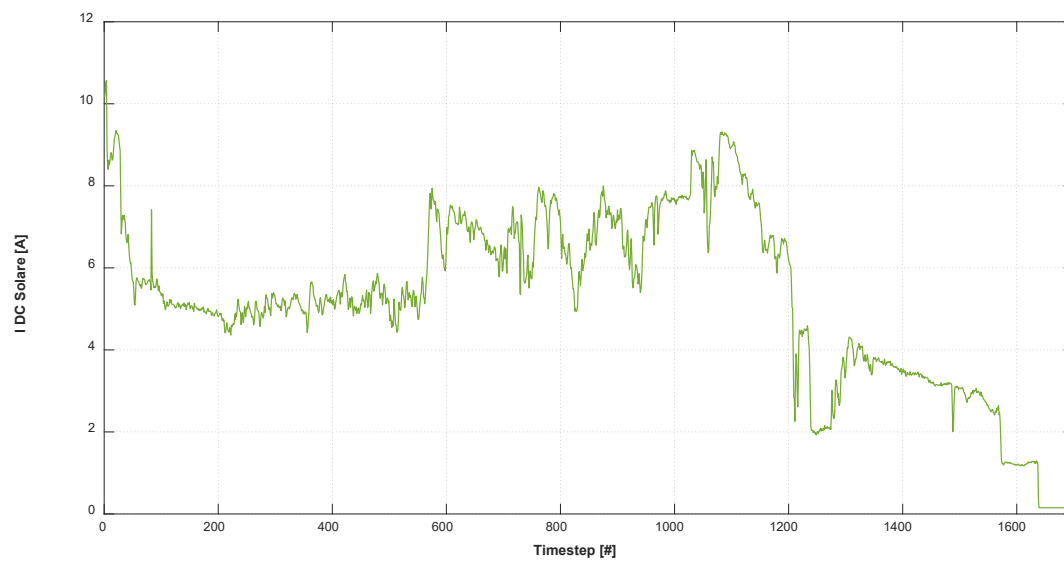


Figura 42 – Corrente prodotta dai pannelli fotovoltaici (test #2 – luglio 2020)

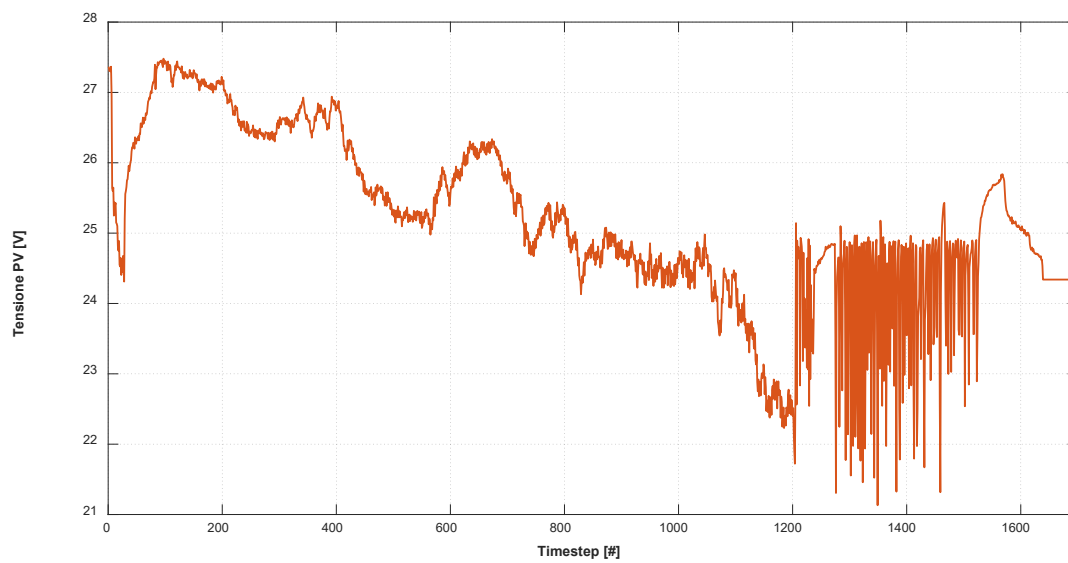


Figura 43 – Tensione dei pannelli fotovoltaici (test #2 – luglio 2020)

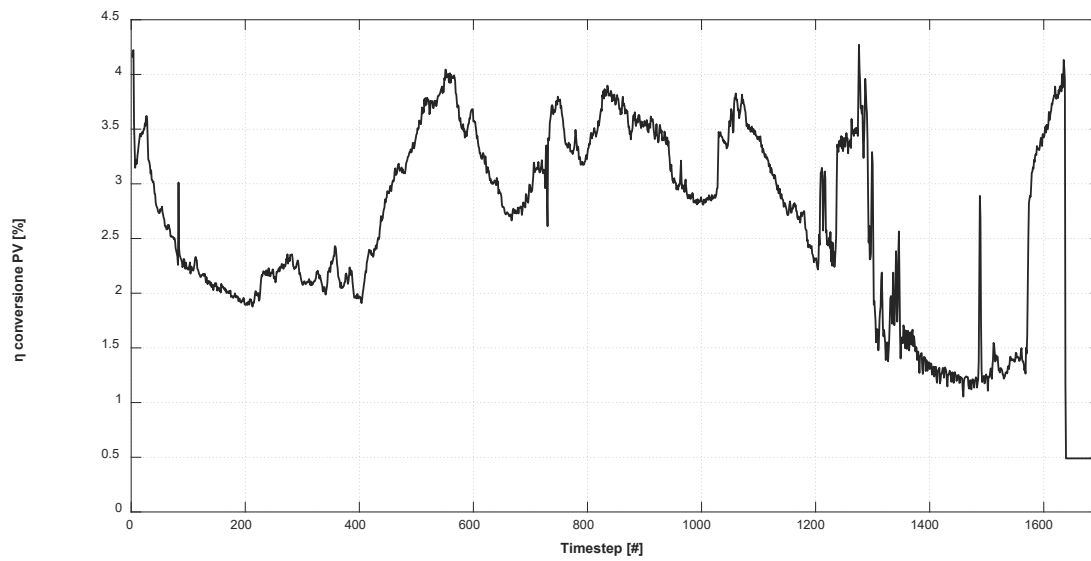


Figura 44 – Rendimento di conversione dei pannelli fotovoltaici (test #2 – luglio 2020)

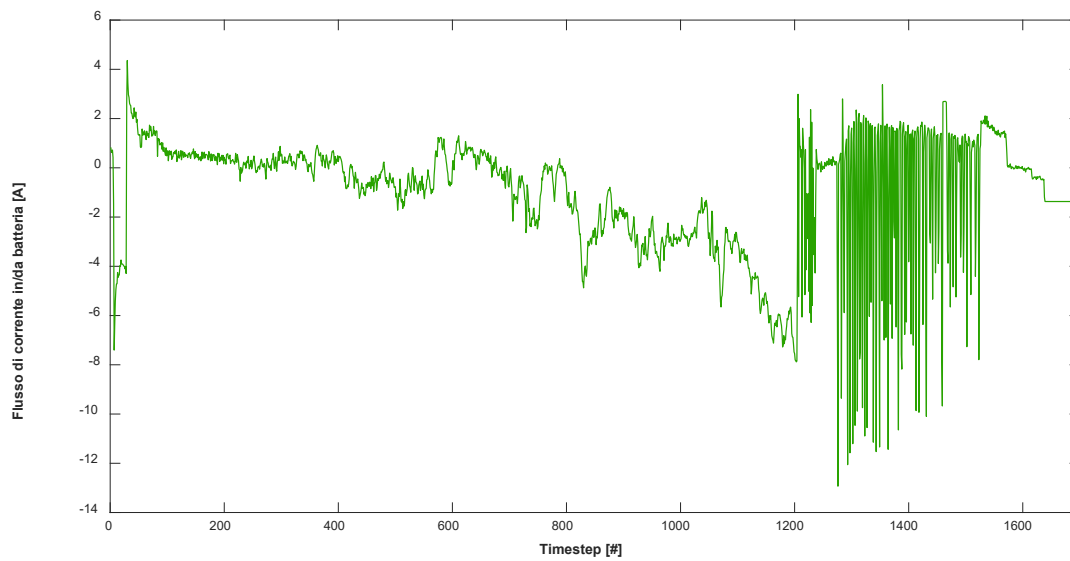


Figura 45 – Corrente in ingresso o in uscita dalla batteria (test #2 – luglio 2020)

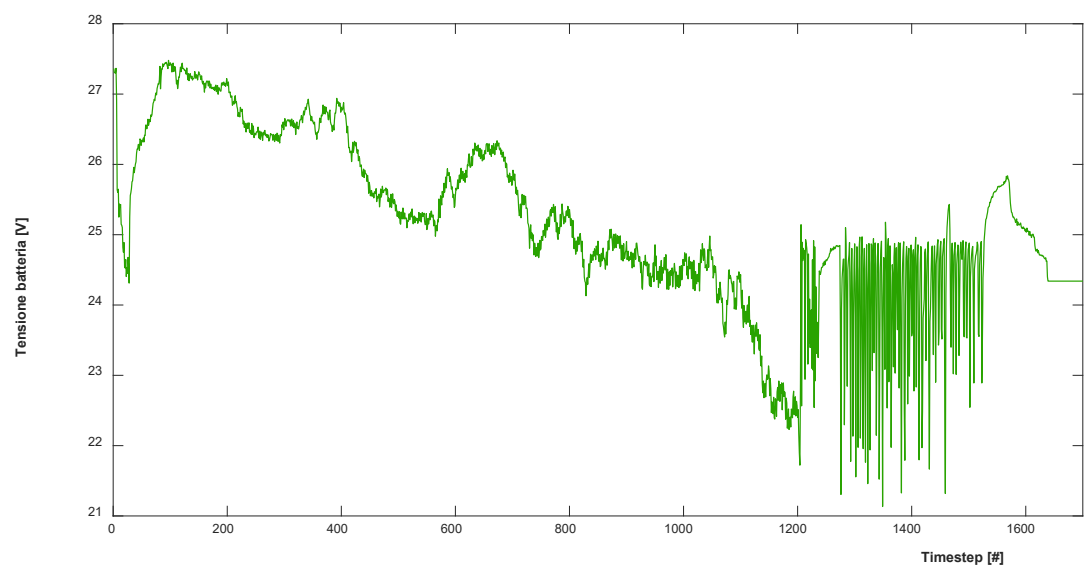


Figura 46 – Tensione della batteria (test #2 – luglio 2020)

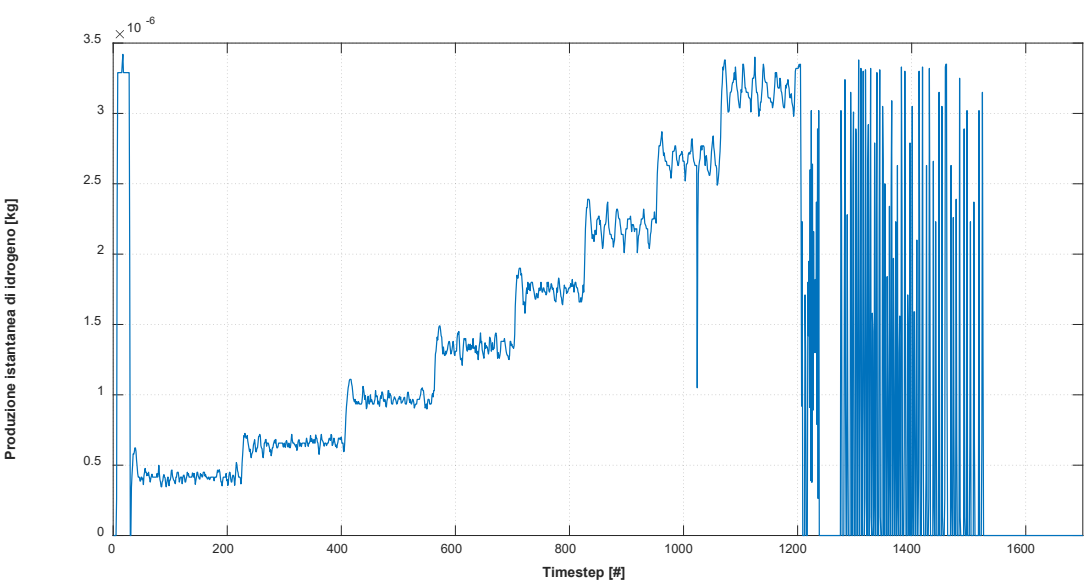


Figura 47 – Idrogeno prodotto (test #2 – luglio 2020)

2.3 Caratterizzazione sperimentale della Fuel cell

Per l'allestimento della linea ad idrogeno presso il Laboratorio di Meccanica e la successiva integrazione del ramo d'impianto all'interno della microrete si è ritenuto opportuno realizzare una calibrazione sperimentale della cella a combustibile di tecnologia PEM sopra citata, in modo tale da poterne monitorare le curve caratteristiche durante il funzionamento e, di conseguenza, i consumi al fine di dimensionare correttamente la linea di alimentazione.

In seguito ad una prima fase di rodaggio e preparazione del dispositivo, secondo quanto consigliato dal costruttore, si è passati ad analizzare il comportamento della cella in differenti condizioni di carico, procedendo per variazioni a gradino della corrente richiesta. Per il controllo della cella durante le rilevazioni si è fatto uso di una piattaforma di acquisizione dati (DAQ) National Instrument, opportunamente collegata al dispositivo mediante l'uso di alcuni sensori ed attuatori per la misura delle principali grandezze fisiche, ed è stato sviluppato in ambiente LabVIEW un sistema di controllo che permettesse di monitorare in tempo reale la dinamica del test.

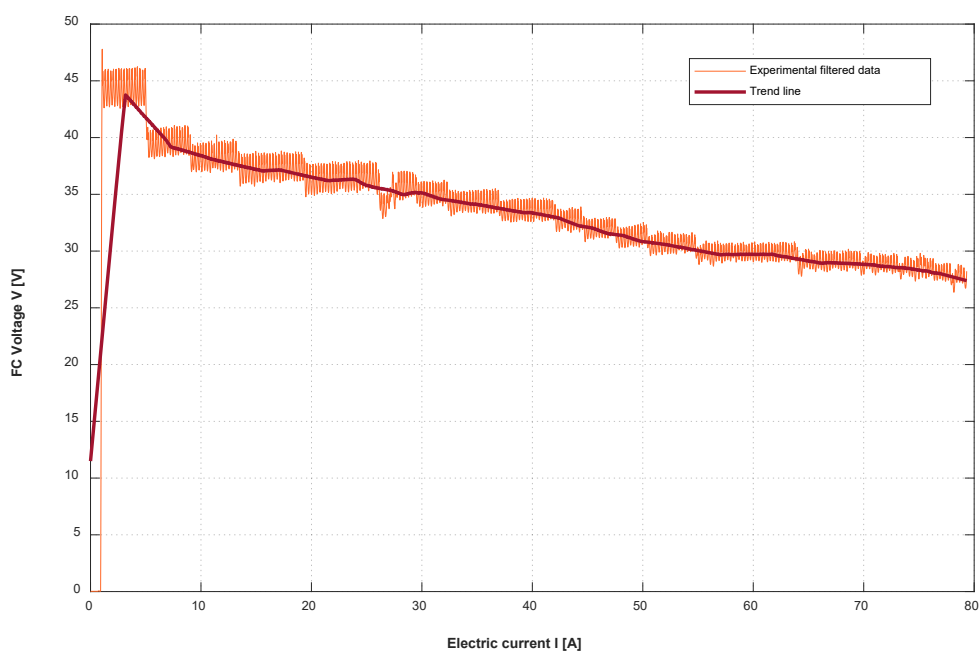
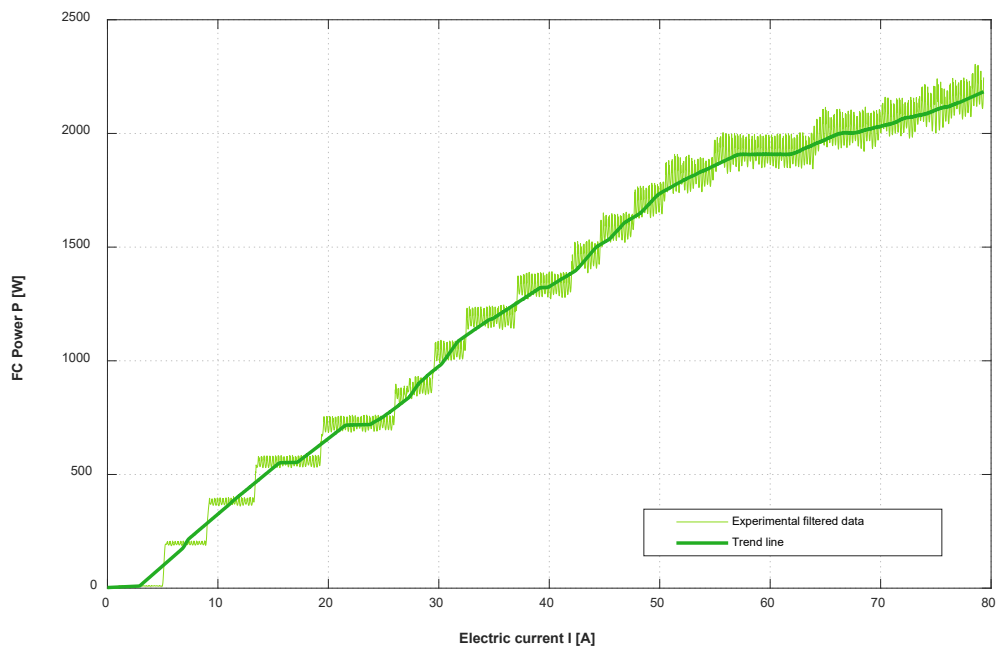
Una tabella riassuntiva contenente i principali parametri operativi della cella, forniti dal costruttore, è indicata in [Figura 48](#).

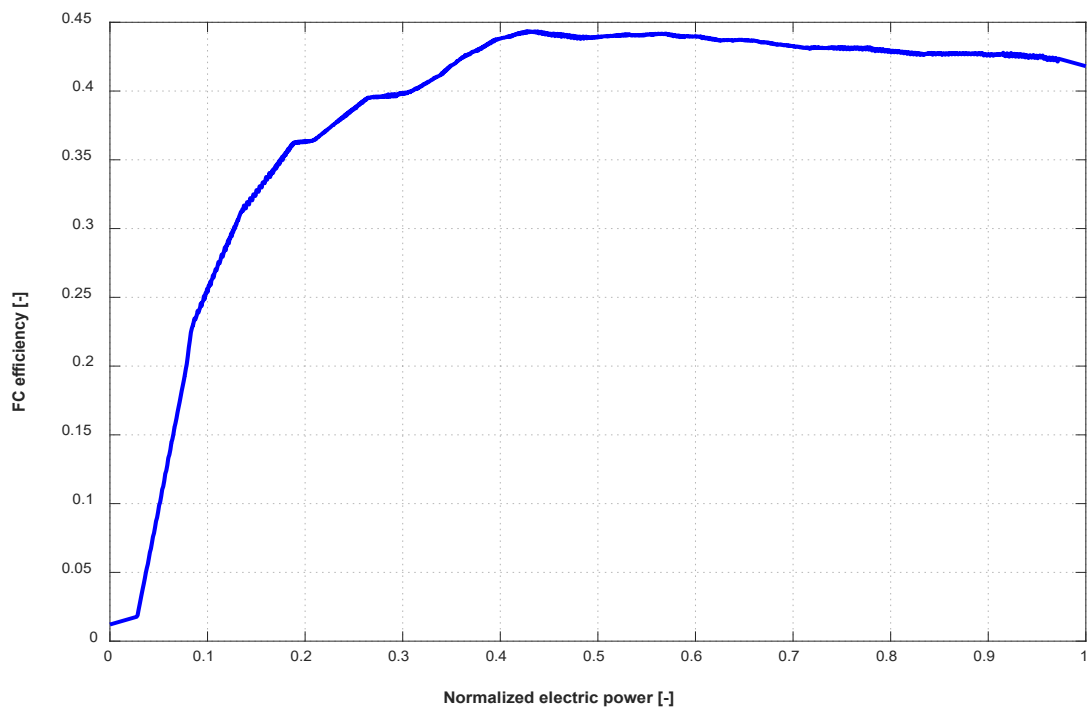
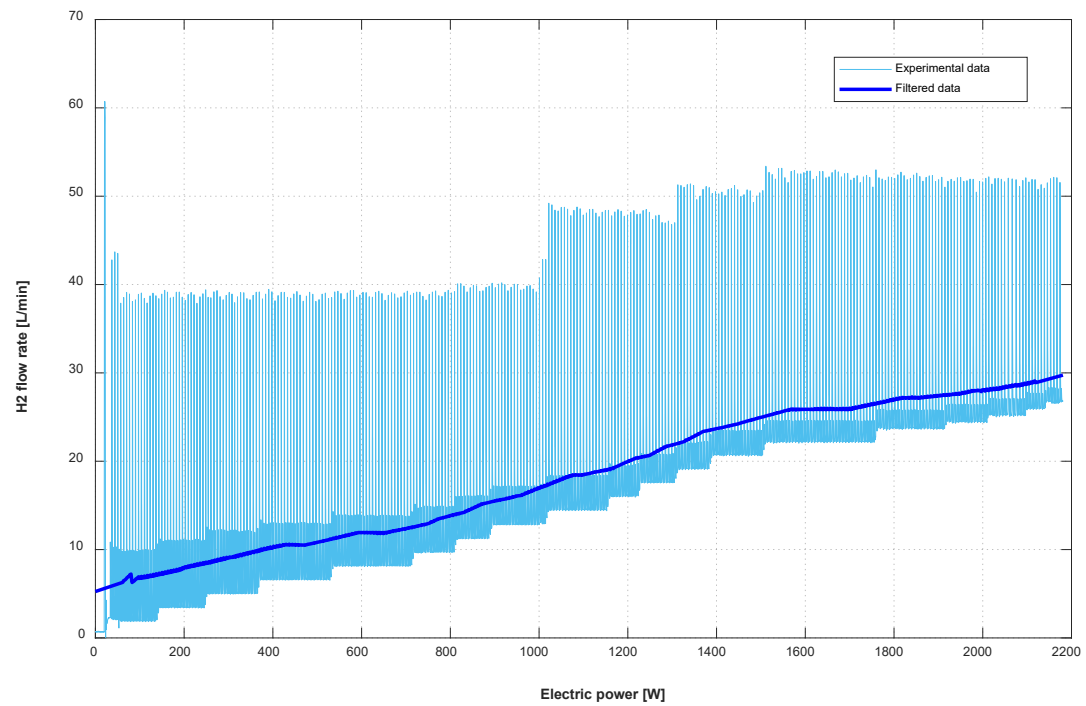
Per maggiori informazioni si rimanda al datasheet del dispositivo inserito in allegato.

Type of fuel cell	PEM
Number of cells	48
Rated Power	2000W
Performance	28.8V @ 70A
H2 Supply valve voltage	12V
Purging valve voltage	12V
Blower voltage	12V
Reactants	Hydrogen and Air
External temperature	5 to 30°C
Max stack temperature	65°C
H2 Pressure	0.45-0.55bar
Hydrogen purity	≥ 99.995 % dry H2
Humidification	self-humidified
Cooling	Air (integrated cooling fan)
Weight (with fan & casing)	10Kg (±200grams)
Controller	2.5Kg (±100grams)
Dimension	30.3cm x 35cm x 18.3cm
Flow rate at max output*	26 L/min
Start up time	≤ 30S at ambient temperature
Efficiency of stack	40% @ 28.8V
Low voltage shut down	24V
Over current shut down	90A
Over temperature shut down	65°C
External power supply**	13V(±1V), 5A~8A

Sono di seguito mostrate alcune curve sperimentali ricavate durante i test effettuati sulla cella nel Laboratorio di Meccanica dell'Università di Bologna.

In particolare, in **Figura 49** è riportata la curva di potenza elettrica ottenuta in uscita dalla cella per differenti valori di corrente richiesti dal carico e, contestualmente, in **Figura 50** è mostrato l'andamento della tensione misurata ai capi del dispositivo durante il medesimo test. Nelle **Figura 51** e **Figura 52** sono infine proposti rispettivamente il consumo di idrogeno da parte del dispositivo e la curva di efficienza della cella in funzione del carico ottenuta a partire dai dati sperimentali.





2.4 Modellazione in Simulink dei componenti d'impianto e costituzione del layout

Per la costituzione dei modelli computazionali si è fatto riferimento all'attività svolta durante la prima annualità di questo progetto, volta a ricostruire i modelli fisici dei componenti d'impianto ed a studiarne le prestazioni nel tempo ai fini dell'inserimento di una sezione di produzione e stoccaggio dell'idrogeno come strategia per aumentare l'indipendenza energetica del sistema.

Si è scelto quindi di tradurre tali modelli di calcolo - validati sperimentalmente secondo quanto riportato nel report relativo al primo anno del progetto - in ambiente Simulink con opportune semplificazioni dettate dalle necessità di interfacciarsi con i dispositivi elettronici Arduino, le cui prestazioni in termini di memoria disponibile e potenza di calcolo non sono certamente confrontabili con quelle proprie del PC ospitante il modello Simulink complessivo.

2.4.1 Modello di calcolo del pannello fotovoltaico

Il modello matematico a cui si è fatto riferimento in prima analisi corrisponde a quello preso in esame durante la prima annualità del progetto. Tale modello può essere facilmente rappresentato attraverso il circuito elettrico equivalente del dispositivo in **Figura 53** in cui, I_L rappresenta la corrente elettrica generata dalla radiazione incidente sul pannello, I_d corrisponde alla corrente di diodo, R_p e R_s sono le resistenze equivalenti in parallelo e in serie ed I_{PV} e V_{PV} sono rispettivamente la corrente elettrica e la tensione di output del pannello.

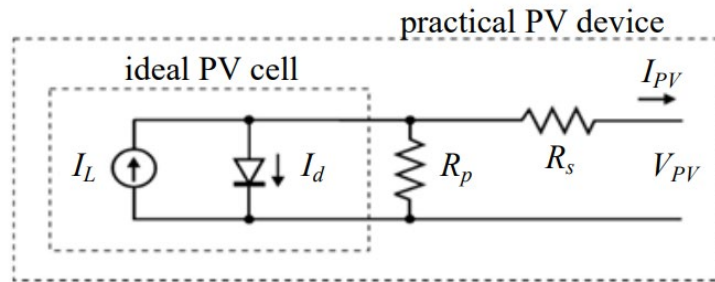


Figura 53 - Circuito equivalente di un pannello fotovoltaico

Sono di seguito riportate le equazioni che descrivono il modello:

$$I = I_{pv} - I_0 \left[\exp \left(\frac{V + R_s I}{V_t a} \right) - 1 \right] - \frac{V + R_s I}{R_p} \quad (1)$$

$$I_{pv} = (I_{pv,n} + K_I \Delta T) \frac{G}{G_n} \quad (2)$$

$$I_{pv,n} = \frac{R_p + R_p}{R_p} I_{sc,n} \quad (3)$$

$$I_0 = \frac{I_{sc,n} + K_I \Delta T}{\exp((V_{oc,n} + K_V \Delta T) / a V_t) - 1} \quad (4)$$

$$V_t = \frac{N_s k T}{q} \quad (5)$$

$$\Delta T = T - T_n \quad (6)$$

dove I_{pv} [A] è la corrente elettrica generata dalla luce incidente sul pannello solare, I_0 [A] è la corrente di saturazione del diodo, $I_{pv,n}$ [A] è la corrente elettrica generata dalla luce incidente sul pannello solare in condizioni nominali, $I_{sc,n}$ [A] è la corrente di corto-circuito in condizioni nominali, V_t [V] è la tensione termica, $V_{oc,n}$ [V] è la tensione a circuito aperto in condizioni nominali, N_s [-] è il numero di celle connesse in serie, q [C] è la carica di un elettrone, k [J/K] è la costante di Boltzmann, a [-] è la costante del diodo, K_V [V/K] è il coefficiente di correzione tensione/temperatura a circuito aperto, K_I [A/K] è il coefficiente di correzione corrente/temperatura di corto-circuito, T [K] è la temperatura operativa del pannello, T_n [K] è la temperatura nominale, G [W/m²] è l'irraggiamento sulla superficie del pannello e G_n [W/m²] è l'irraggiamento nominale.

Servendosi delle equazioni riportate sopra si è realizzato il modello computazionale in ambiente Simulink (Figura 54 e Figura 55), tramite cui è stato possibile simulare, stabiliti i valori del profilo di irraggiamento solare e della temperatura misurata al pannello, le curve di funzionamento caratteristiche dei moduli fotovoltaici installati presso il laboratorio di microreti, assegnando opportunamente ai parametri coinvolti nelle equazioni i dati di targa dei pannelli fotovoltaici in dotazione, generalmente ritrovabili nel corrispettivo datasheet.

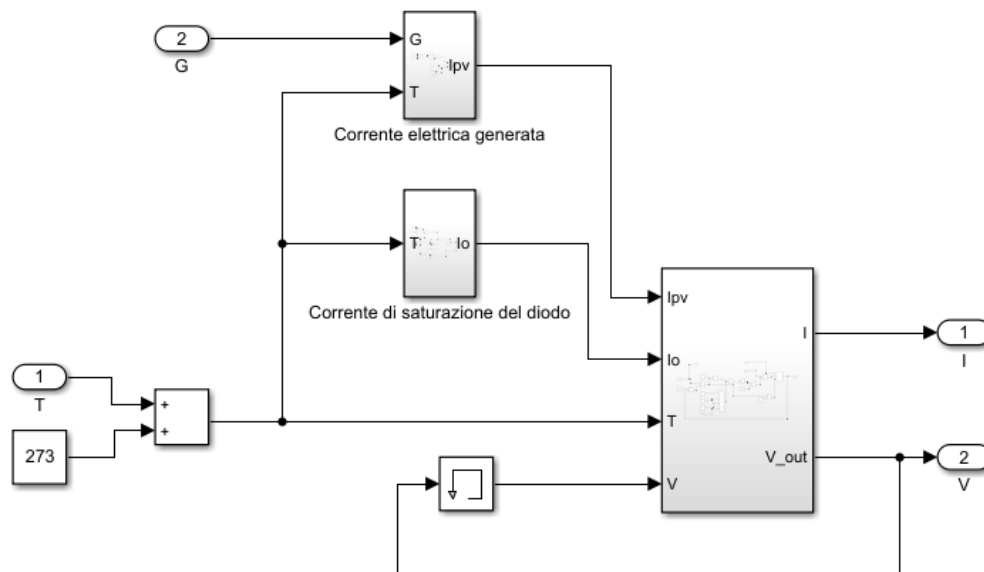


Figura 54 - modello di calcolo del pannello fotovoltaico

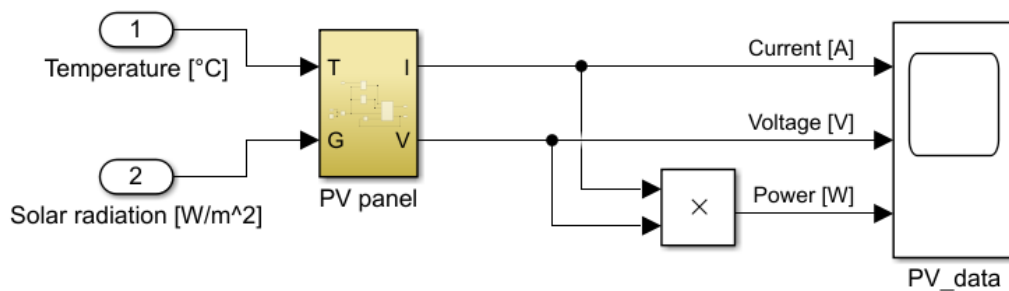


Figura 55 – vista complessiva del modello Simulink relativo al pannello fotovoltaico

Al fine di rendere tale modello computazionale facilmente riproducibile mediante un codice caricabile sulla scheda Arduino preposta, si è scelto di procedere con una caratterizzazione del modulo fotovoltaico ottenibile attraverso una serie di simulazioni di funzionamento dello stesso, volte al calcolo della potenza elettrica generata al variare dei valori di irraggiamento solare e temperatura del pannello. In questo modo è stato possibile, dopo aver opportunamente adimensionalizzato i valori di potenza prodotta ottenuti, definire una correlazione diretta tra i dati input (irraggiamento e temperatura del modulo) e la potenza elettrica generata dal pannello, servendosi di una regressione multipla a partire dai dati di output delle simulazioni di funzionamento realizzate (**Tabella 4** e **Tabella 5**).

Tabella 4 - Lookup table per la caratterizzazione del pannello fotovoltaico

P_{el,max}=222.5 [W]		Potenza elettrica generata [W]					
		Temperatura pannello [°C]					
		25°C	35°C	45°C	55°C	65°C	75°C
Irraggiamento solare [W/m ²]	0	0.00	0.00	0.00	0.00	0.00	0.00
	50	4.08	3.06	1.98	0.85	0.00	0.00
	100	15.67	14.64	13.54	12.39	11.18	9.92
	150	27.26	26.20	25.09	23.91	22.69	21.41
	200	38.84	37.76	36.62	35.43	34.18	32.89
	250	50.41	49.31	48.15	46.93	45.67	44.35
	300	61.96	60.84	59.66	58.43	57.14	55.80
	350	73.51	72.37	71.17	69.91	68.60	67.24
	400	85.04	83.88	82.66	81.38	80.04	78.66
	450	96.57	95.38	94.13	92.83	91.48	90.07
	500	108.08	106.87	105.60	104.27	102.89	101.47
	550	119.58	118.34	117.05	115.70	114.30	112.85
	600	131.06	129.80	128.48	127.11	125.69	124.22
	650	142.53	141.25	139.91	138.51	137.06	135.57
	700	153.99	152.68	151.32	149.89	148.42	146.90
	750	165.44	164.10	162.71	161.26	159.76	158.22
	800	176.87	175.51	174.09	172.61	171.09	169.53
	850	188.29	186.90	185.45	183.95	182.40	180.81
	900	199.69	198.27	196.79	195.27	193.69	192.08
	950	211.07	209.62	208.12	206.57	204.97	203.33
	1000	222.44	220.96	219.43	217.85	216.23	214.56

Tabella 5 - Lookup table per il calcolo della potenza elettrica generata normalizzata

$P_{el,max}=1$ [-]		Potenza elettrica generata adimensionalizzata [-]					
		Temperatura pannello [°C]					
		25°C	35°C	45°C	55°C	65°C	75°C
Irraggiamento solare [W/m²]	0	0.000	0.000	0.000	0.000	0.000	0.000
	50	0.018	0.014	0.009	0.004	0.000	0.000
	100	0.070	0.066	0.061	0.056	0.050	0.045
	150	0.123	0.118	0.113	0.107	0.102	0.096
	200	0.175	0.170	0.165	0.159	0.154	0.148
	250	0.227	0.222	0.216	0.211	0.205	0.199
	300	0.278	0.273	0.268	0.263	0.257	0.251
	350	0.330	0.325	0.320	0.314	0.308	0.302
	400	0.382	0.377	0.371	0.366	0.360	0.354
	450	0.434	0.429	0.423	0.417	0.411	0.405
	500	0.486	0.480	0.475	0.469	0.462	0.456
	550	0.537	0.532	0.526	0.520	0.514	0.507
	600	0.589	0.583	0.577	0.571	0.565	0.558
	650	0.641	0.635	0.629	0.623	0.616	0.609
	700	0.692	0.686	0.680	0.674	0.667	0.660
	750	0.744	0.738	0.731	0.725	0.718	0.711
	800	0.795	0.789	0.782	0.776	0.769	0.762
	850	0.846	0.840	0.833	0.827	0.820	0.813
	900	0.897	0.891	0.884	0.878	0.871	0.863
	950	0.949	0.942	0.935	0.928	0.921	0.914
	1000	1.000	0.993	0.986	0.979	0.972	0.964

Da cui:

$$P_{el,max,adm} = -0.013484962 + 0.001027414 \cdot G - 0.000597143 \cdot T \quad (7)$$

dove G [W/m²] è l'irraggiamento sulla superficie del pannello e T [°C] è la temperatura operativa del pannello.

2.4.2 Modello di calcolo del dispositivo di accumulo elettrochimico

Per quanto riguarda la modellizzazione del dispositivo di accumulo elettrochimico si è ritenuto, in prima approssimazione, che questo si comportasse come un comune serbatoio di processo, comunicante con il sistema attraverso un ingresso ed un'uscita, e che la sua interazione con gli altri componenti d'impianto fosse essenzialmente rivolta allo studio della più efficiente strategia d'accumulo energetico al variare delle condizioni climatiche (periodo estivo o periodo invernale) e della necessità energetica da parte delle utenze in gioco. Per tali ragioni, si è potuto assumere che i processi di scarica e carica fossero descritti dalla medesima curva caratteristica V -SOC (**Figura 56**) ottenuta sperimentalmente attraverso le rilevazioni effettuate in laboratorio e relative alla finestra operativa della batteria durante il processo di scarica.

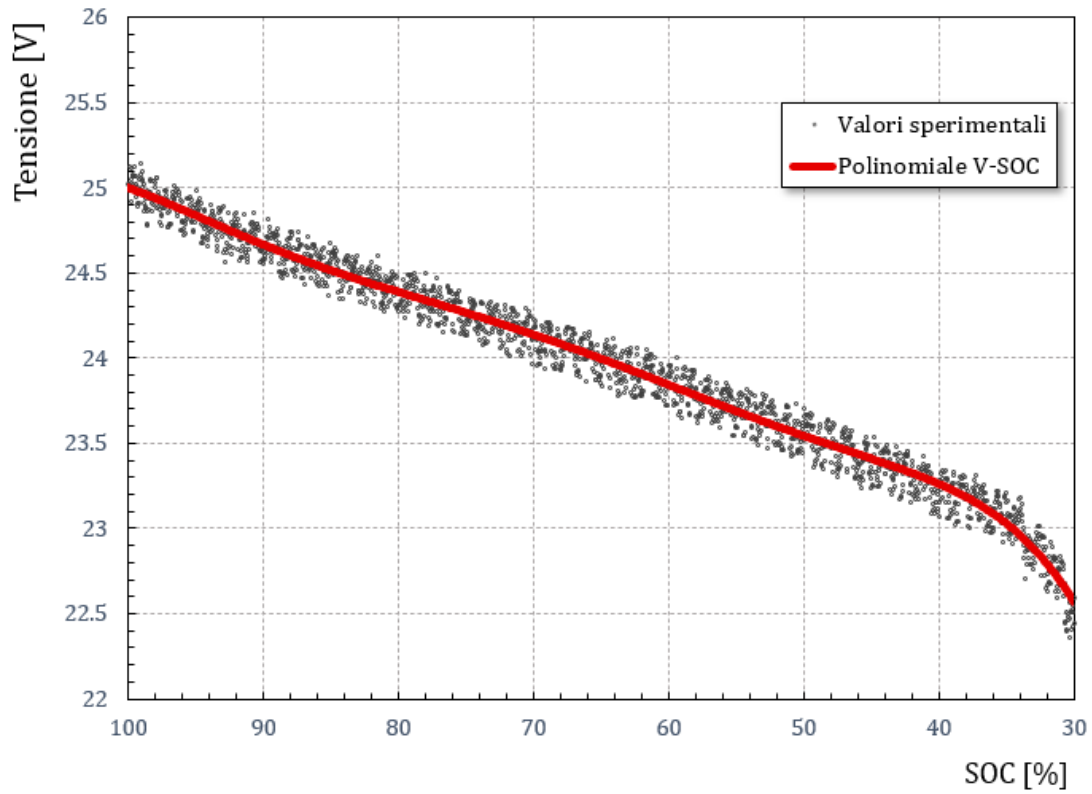


Figura 56 – Curva caratteristica V-SOC della batteria durante il processo di scarica

Una volta nota la correlazione tra la tensione ai capi della batteria ed il suo stato di carica, si è potuto modellizzare le fasi di carica e scarica mediante le seguenti equazioni:

$$V_{batt,i-1} = f(SOC_{i-1}) \quad (8)$$

$$I_{batt,i} = \frac{P_i}{V_{batt,i-1}} \quad (9)$$

$$C_{batt,i} = C_{batt,i-1} + I_{batt,i} \cdot \Delta t \quad (10)$$

$$SOC_i = \frac{C_{batt,i}}{C_{batt,max}} \cdot 100 \quad (11)$$

In cui $C_{batt,i}$ e $C_{batt,i-1}$ sono rispettivamente le capacità della batteria in [Ah], relative all'istante di tempo considerato ed a quello precedente, $C_{batt,max}$ [Ah] rappresenta la capacità massima della batteria, $I_{batt,i}$ [A] e P_i [W] corrispondono rispettivamente alla corrente di carica/scarica ed alla potenza elettrica ricevuta/erogata dalla batteria nell'istante di tempo i , SOC_i ed SOC_{i-1} sono nell'ordine lo stato di carica della batteria all'istante considerato ed a quello precedente, ed infine $V_{batt,i-1}$ [V] rappresenta la tensione della batteria relativa allo step temporale precedente, ottenuta mediante la correlazione sperimentale di cui sopra.

Come è possibile notare dal grafico sopra mostrato si è assunto uno stato di carica minimo della batteria durante la fase di scarica pari al 30% della capacità massima della stessa, in linea con quanto tipicamente indicato per prevenire situazioni di scarica profonda negli accumulatori di tipo AGM. Allo stesso modo, per una maggiore fedeltà del modello Simulink (Figura 57 e Figura 58) sono stati imposti dei vincoli anche alla fase di carica della batteria, fissando all'80% lo stato di carica massimo ed imponendo per la massima corrente di carica un valore limite in [A] pari al 30% della capacità massima della batteria.

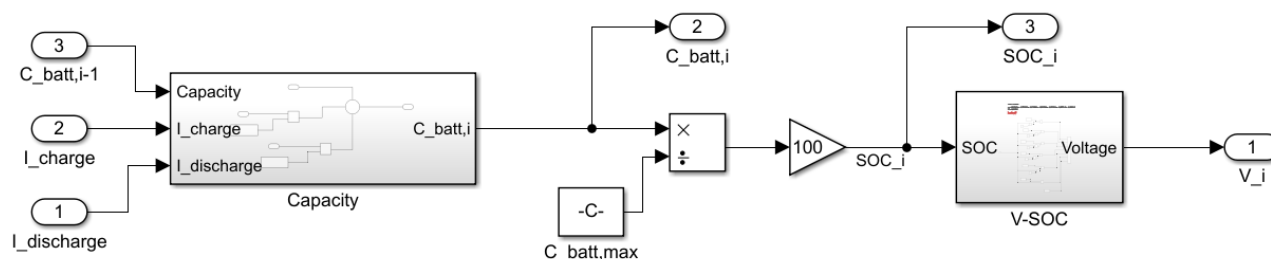


Figura 57 – modello di calcolo della batteria

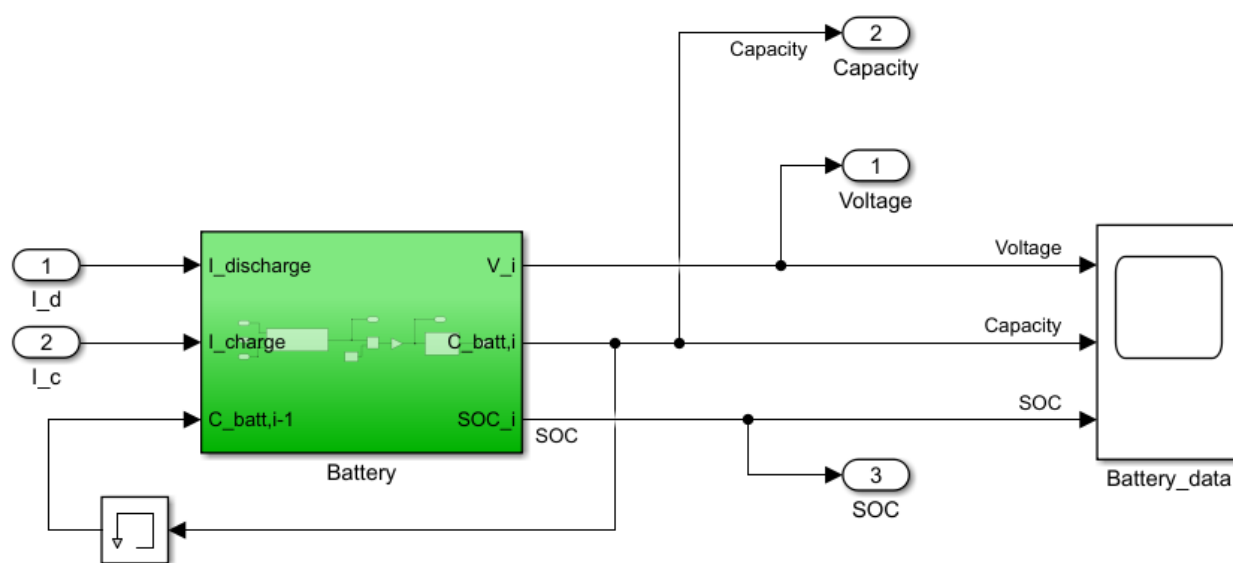


Figura 58 – vista complessiva del modello Simulink relativo alla batteria

2.4.3 Modello di calcolo dell'elettrolizzatore

Per la realizzazione del modello di calcolo dell'elettrolizzatore si è fatto riferimento alla caratterizzazione sperimentale del generatore di idrogeno di tipo PEM alimentato in AC installato presso il Laboratorio di Microreti di Generazione ed Accumulo dell'Alma Mater Studiorum - Università Di Bologna. Le rilevazioni effettuate su tale dispositivo, funzionante in regime di voltaggio variabile, hanno prodotto una correlazione polinomiale tra l'efficienza del macchinario e la potenza elettrica assorbita dallo stesso per realizzare la conversione elettrochimica. Tale curva di efficienza è stata poi opportunamente adimensionalizzata rispetto alla potenza introdotta nel dispositivo, in modo tale da poter scalare il comportamento del generatore per differenti taglie del medesimo.

Una rappresentazione della suddetta curva caratteristica è proposta in Figura 59.

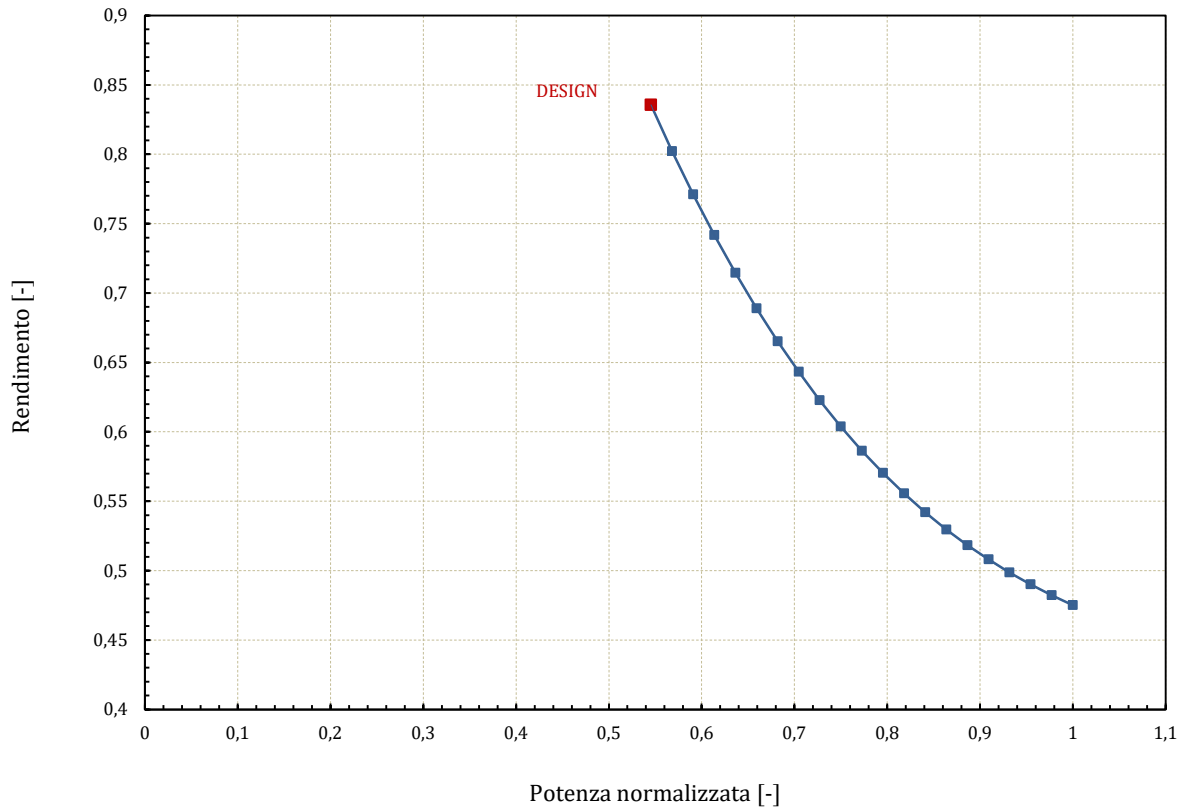


Figura 59 - Efficienza dell'elettrolizzatore in funzione della potenza normalizzata.

Come si nota in **Figura 59**, il campo di funzionamento dell'elettrolizzatore è limitato ad un intervallo di regolazione ridotto, che ha come estremo inferiore il suo punto di design, caratterizzato da massimo rendimento e minimo assorbimento elettrico, e come estremo superiore il punto a minima efficienza di produzione e di conseguenza massima potenza assorbita.

Una volta nota l'efficienza operativa del macchinario, il calcolo della portata massica di idrogeno prodotto nell'unità di tempo è espresso dalla seguente relazione:

$$\dot{m}_{H_2} = \frac{P_{el,in,E} \cdot \eta_E}{HHV_{H_2}} \quad (12)$$

Dove $P_{el,in,E}$ [kW] è la potenza elettrica assorbita dal generatore, η_E [-] è l'efficienza del generatore e HHV_{H_2} [kJ/kg] rappresenta il potere calorifico superiore dell'idrogeno.

I modelli Simulink realizzati sono presentati in **Figura 60** e **Figura 61**.

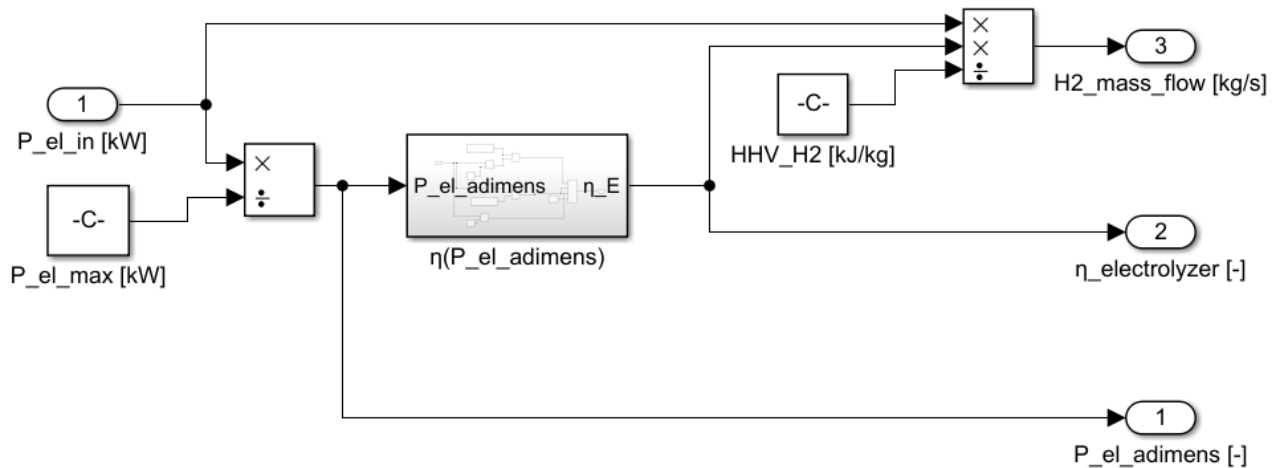


Figura 60 - modello di calcolo dell'elettrolizzatore

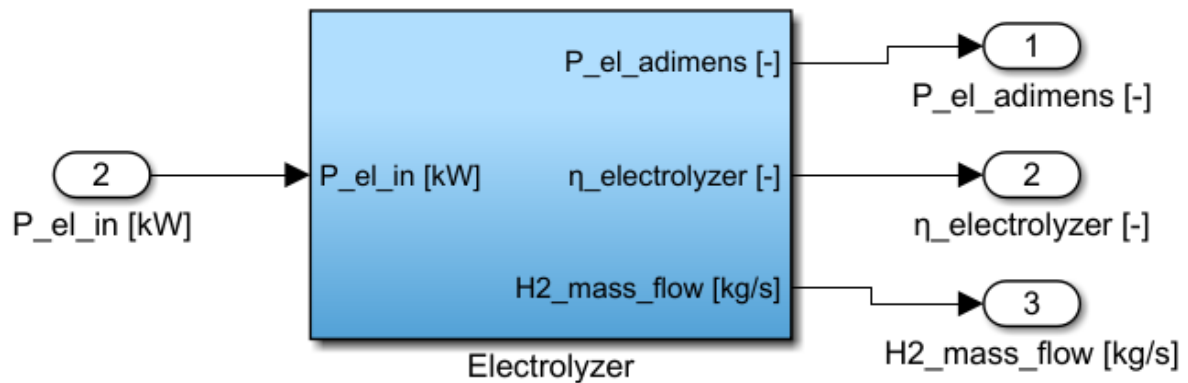


Figura 61 - vista complessiva del modello Simulink relativo all'elettrolizzatore

2.4.4 Modello di calcolo della fuel cell

Il modello di calcolo relativo alla fuel cell è stato realizzato a partire dalla caratterizzazione sperimentalmente ottenuta sul dispositivo in dotazione presso il Laboratorio di Microreti di Generazione ed Accumulo dell'Università di Bologna.

Anche in questo caso, come per l'elettrolizzatore, è stato utile normalizzare la potenza elettrica fornita istantaneamente dal dispositivo rispetto al suo punto di design, al fine di scalare la curva di efficienza della fuel cell per più taglie della stessa, come mostrato in **Figura 62**.

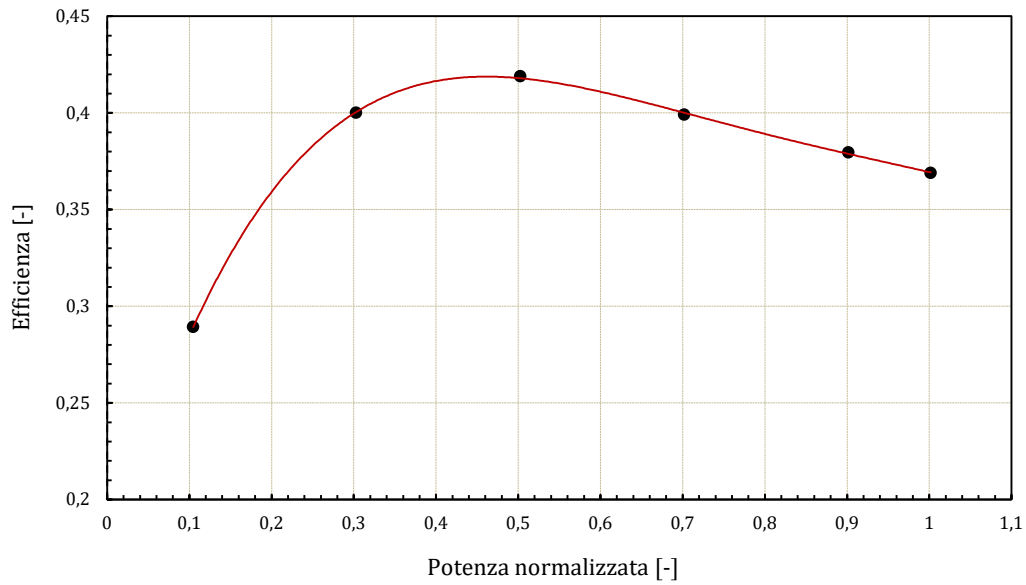


Figura 62 - Curva di efficienza della fuel cell al variare della potenza elettrica normalizzata

Diversamente rispetto a quanto accade per l'elettrolizzatore, il range di funzionamento della fuel cell risulta maggiormente esteso e permette ad essa di operare a partire dal 10% del suo carico di design fino alla massima potenza richiesta.

Per il calcolo della portata massica di idrogeno necessaria al dispositivo per il suo funzionamento si è fatto infine riferimento alla seguente relazione:

$$\dot{m}_{H_2} = \frac{P_{el,out,FC}}{\eta_{FC} \cdot LHV_{H_2}} \quad (13)$$

in cui $P_{el,out,FC}$ [kW] rappresenta la potenza elettrica fornita in uscita dalla fuel cell, η_{FC} [-] è l'efficienza della fuel cell e LHV_{H_2} [kJ/kg] il potere calorifico inferiore dell'idrogeno.

In **Figura 63** e **Figura 64** sono presentate le implementazioni Simulink dei modelli in oggetto.

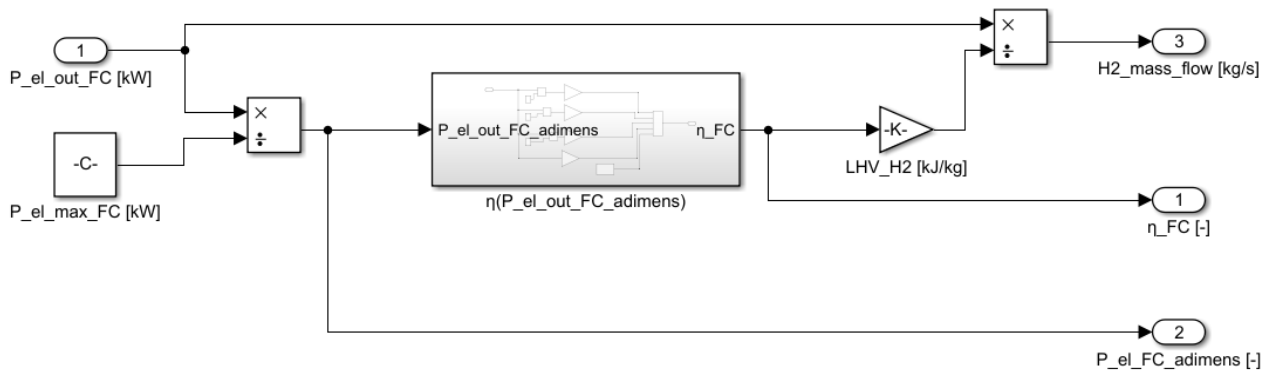


Figura 63 - modello di calcolo della fuel cell

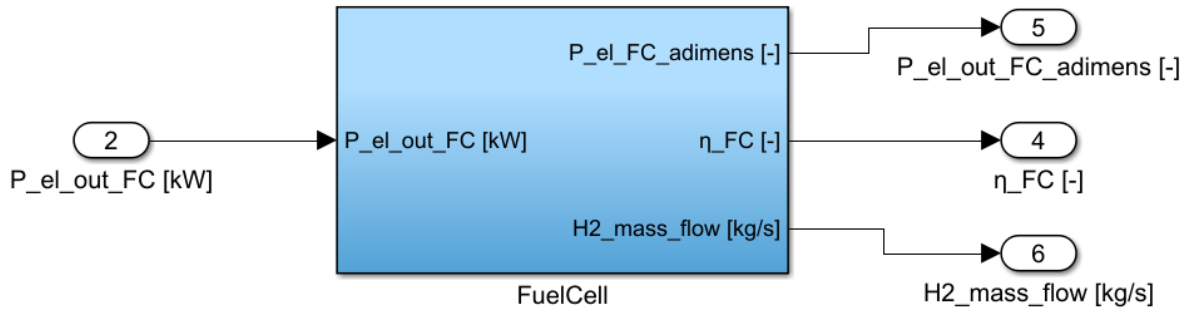


Figura 64 - vista complessiva del modello Simulink relativo alla fuel cell

2.4.5 Modello di calcolo del sistema di compressione ed accumulo dell'idrogeno prodotto

Per la sezione d'impianto dedicata all'idrogeno proveniente dall'elettrolizzatore si è scelto di predisporre un serbatoio per lo stoccaggio pressurizzato del fluido prodotto e di permetterne il riempimento attraverso l'introduzione di un compressore volumetrico alternativo.

Come noto, la necessità di realizzare elaborazioni del fluido attraverso macchine operatrici caratterizzate da alti rapporti di compressione comporta inevitabili problemi dovuti alla parziale conversione in calore dell'energia in gioco, con conseguente stress termico e meccanico sui materiali di cui è costituito il compressore. Tali fenomeni obbligano a frazionare la compressione del fluido in più stadi attraverso l'impiego di un treno di compressori di dimensioni via via decrescenti e caratterizzati dal medesimo rapporto di compressione che, come noto, consente di minimizzare il lavoro speso per l'elaborazione a parità di temperatura del fluido in aspirazione.

Si è pertanto deciso di modellizzare il sistema mediante il calcolo del lavoro di compressione, e dunque dell'assorbimento elettrico, ad opera di un treno di compressori suddiviso in tre stadi tra loro inter-refrigerati (**Figura 65**).

$$\beta_i = \sqrt[m]{\beta} \quad (14)$$

$$P_{el,COMP} = \frac{\dot{m}_{H_2} \cdot C_{p_{H_2}} \cdot T_{in} \cdot \left(\beta^{\frac{R_{H_2}}{C_{p_{H_2}}}} - 1 \right)}{\eta_{tot} \cdot \eta_{adimens}} \quad [kW] \quad (15)$$

$$\eta_{adimens} = [(-0.1444 \cdot \dot{m}_{H_2, ratio} - 0.34) \cdot (\beta_{ratio}^2)] + [(0.1034 \cdot \dot{m}_{H_2, ratio} + 0.7758) \cdot (\beta_{ratio})] + [(0.385 \cdot \dot{m}_{H_2, ratio} + 0.2193)] \quad [-] \quad (16)$$

$$\begin{cases} \dot{m}_{H_2, ratio} = \frac{\dot{m}_{H_2}}{\dot{m}_{H_2, DES}} \\ \beta_{ratio} = \frac{\beta}{\beta_{DES}} \end{cases}$$

Dove m indica il numero di compressori adottati, $\dot{m}_{H_2}$ [kg/s] è la portata massica di idrogeno elaborato dal compressore, $C_{p_{H_2}}$ [kJ/kg·K] è il calore specifico a pressione costante dell'idrogeno, T_{in} [K] è la temperatura del fluido in ingresso, R_{H_2} [kJ/kg·K] corrisponde alla *costante specifica* per l'idrogeno in fase

gas, η_{tot} [-] e $\eta_{adimens}$ [-] rappresentano rispettivamente il rendimento complessivo della macchina e quello adimensionalizzato rispetto ai valori di design della portata massica e del rapporto di compressione.

I modelli Simulink realizzati sono presentati in [Figura 66](#) e [Figura 67](#).

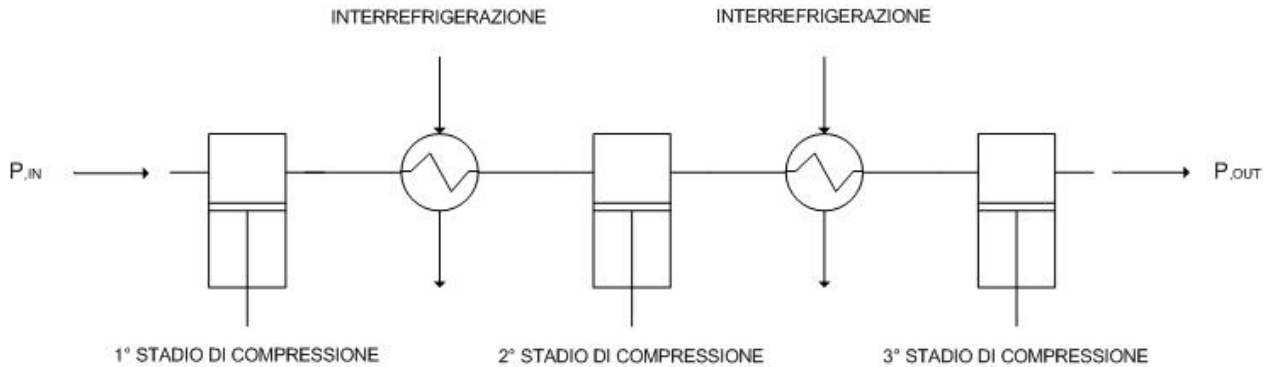


Figura 65 - Schema semplificato dell'inter-refrigerazione

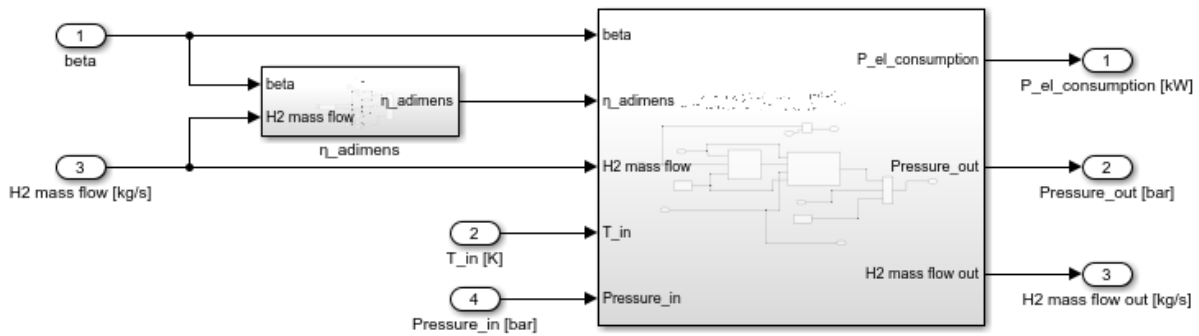


Figura 66 - modello di calcolo del sistema di compressione

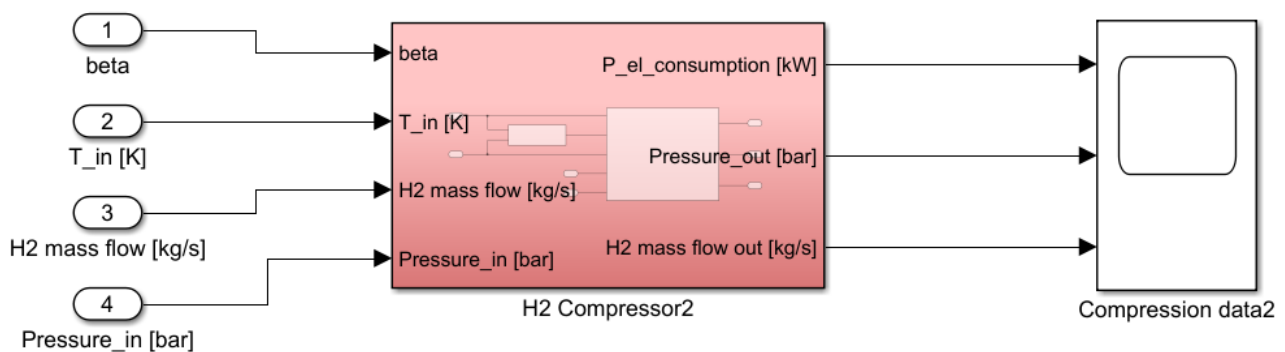


Figura 67 - vista complessiva del modello Simulink relativo al sistema di compressione dell'idrogeno prodotto

Per quanto riguarda il serbatoio per lo stoccaggio pressurizzato dell'idrogeno elaborato dal compressore, è bene ricordare che il comportamento di tale fluido in condizioni reali, specialmente per pressioni medio-alte, risulta piuttosto differente da quello di un gas perfetto, come mostrato in [Figura 68](#).

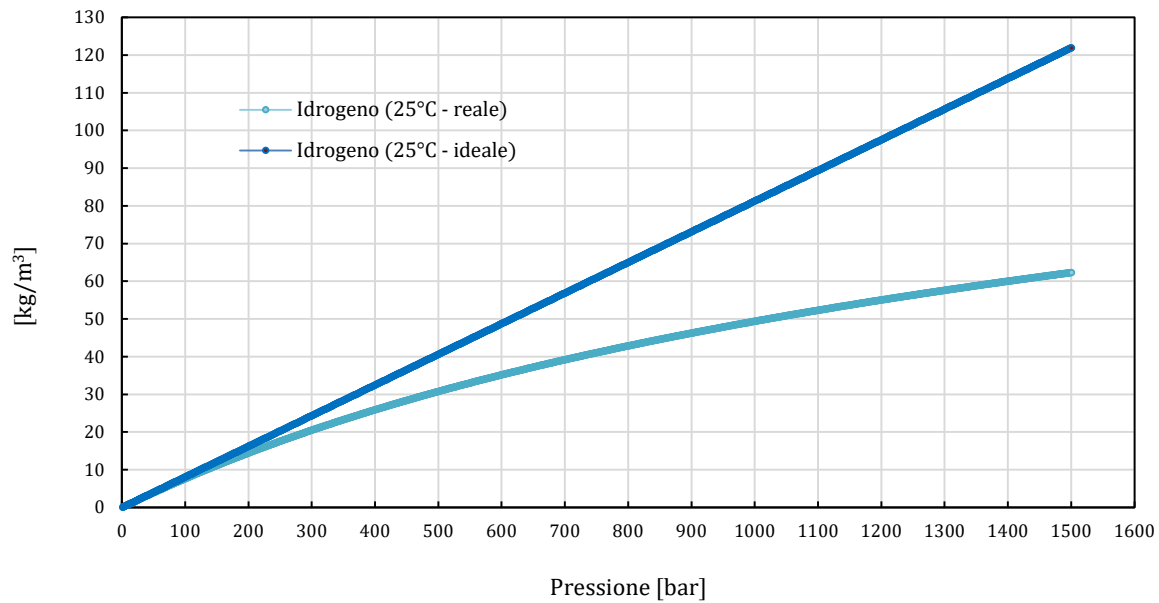


Figura 68 - densità dell'idrogeno al variare della pressione

Ai fini di una maggiore precisione del modello per il calcolo dei parametri operativi durante le fasi di riempimento e svuotamento del serbatoio sarebbe necessario tenere conto di questa caratteristica, tuttavia, evitando eccessive complicazioni derivanti dall'esplicitazione dell'*Equazione di stato dei gas reali* o *legge di Van Der Waals*, si è ritenuto opportuno semplificare il modello facendo riferimento all'*Equazione di stato dei gas perfetti*, tramite cui monitorare i valori di pressione e riempimento del generico serbatoio cilindrico.

$$\left(P + \frac{an^2}{V^2}\right)(V - nb) = nRT \quad [\text{Van Der Waals}] \quad (17)$$

$$PV = nRT \quad [\text{Equazione di stato dei gas perfetti}] \quad (18)$$

Tale scelta è giustificata dal fatto che il principale ruolo del sistema di accumulo è circoscritto alla simulazione del progressivo aumento o decremento della pressione che si realizza all'interno dello stesso durante le fasi di carica e scarica del gas, in modo tale da studiare la risposta del sistema di compressione e valutarne i parametri operativi nonché l'assorbimento elettrico.

Il modello di calcolo del serbatoio di stoccaggio dell'idrogeno e la sua vista complessiva sono rispettivamente mostrati in **Figura 69** e in **Figura 70**.

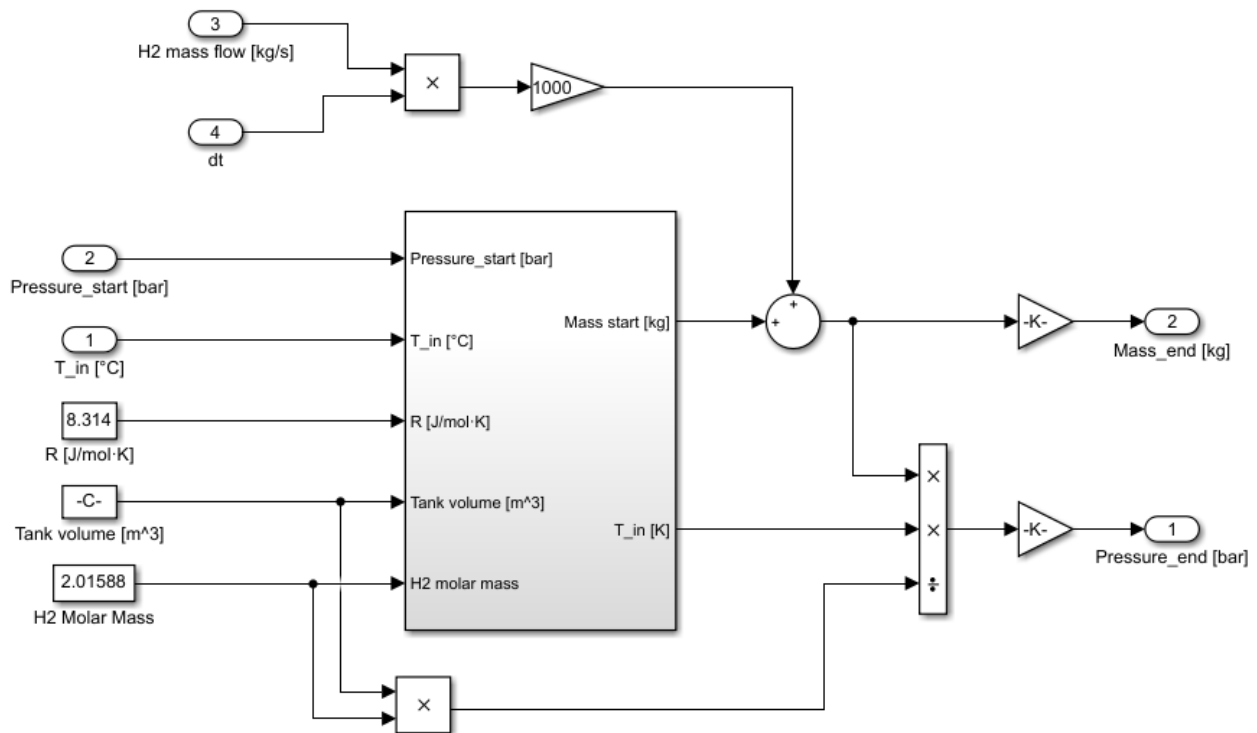


Figura 69 - modello di calcolo del serbatoio di stoccaggio dell'idrogeno

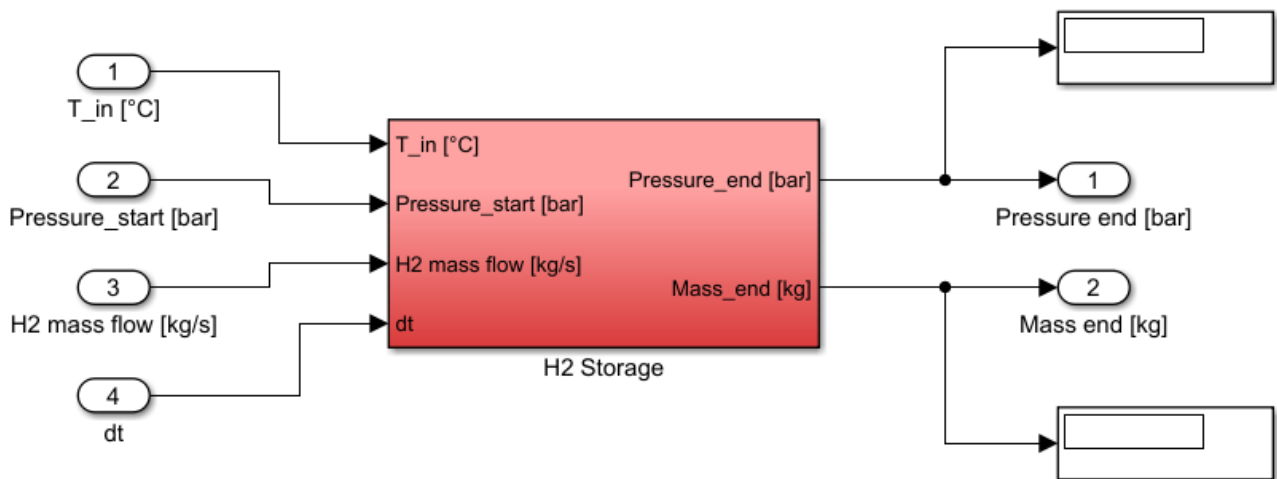


Figura 70 - vista complessiva del modello Simulink relativo al serbatoio di stoccaggio dell'idrogeno prodotto

2.4.6 Modello di calcolo del sistema pompa di calore - accumulo termico

Il modello di calcolo relativo al ramo di produzione ed accumulo dell'energia termica è stato realizzato a partire da un valore assegnato del COP per la pompa di calore. In questo modo, nota la potenza elettrica in ingresso al sistema è possibile conoscere la potenza termica istantaneamente prodotta dalla pompa di calore e di conseguenza monitorare gli scambi energetici con l'utenza. Il modello di calcolo riferito al serbatoio di accumulo termico è stato quindi realizzato imponendo il bilancio energetico per ogni iterazione del programma a partire dalla quota di energia termica immagazzinata all'interno dell'accumulo e dalla

potenza termica istantaneamente richiesta dall'utenza, come mostrato in Figura 71 - modello di calcolo dell'accumulo termico **Figura 71**.

Anche nel caso dei componenti in oggetto sono stati implementati i modelli presentati in **Figura 72** e in **Figura 73**.

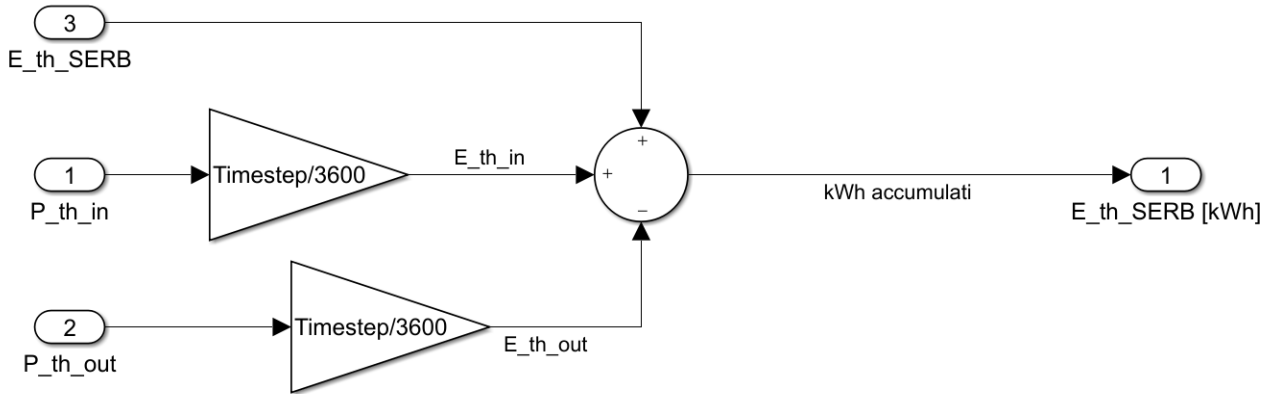


Figura 71 - modello di calcolo dell'accumulo termico

Per un maggiore controllo del livello di acqua calda sanitaria presente all'interno del serbatoio sono stati definiti due parametri, denominati FULL e LOW_TH, per fornire al sistema di controllo un feedback sull'eventuale superamento dei limiti imposti in fase di design, similmente a quanto fatto per il sistema di accumulo elettrochimico.

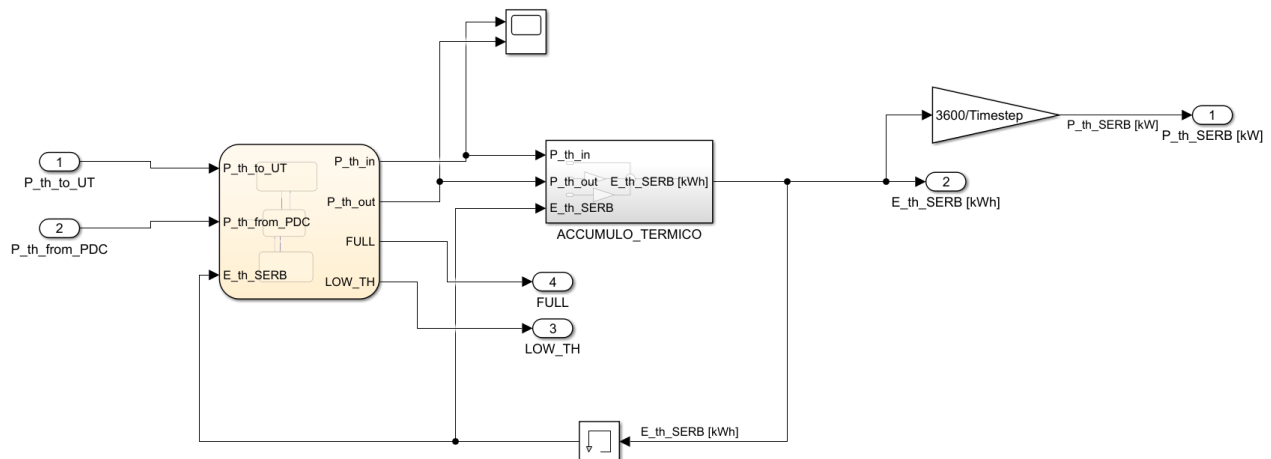


Figura 72 - modello di calcolo del sistema pompa di calore - accumulo termico

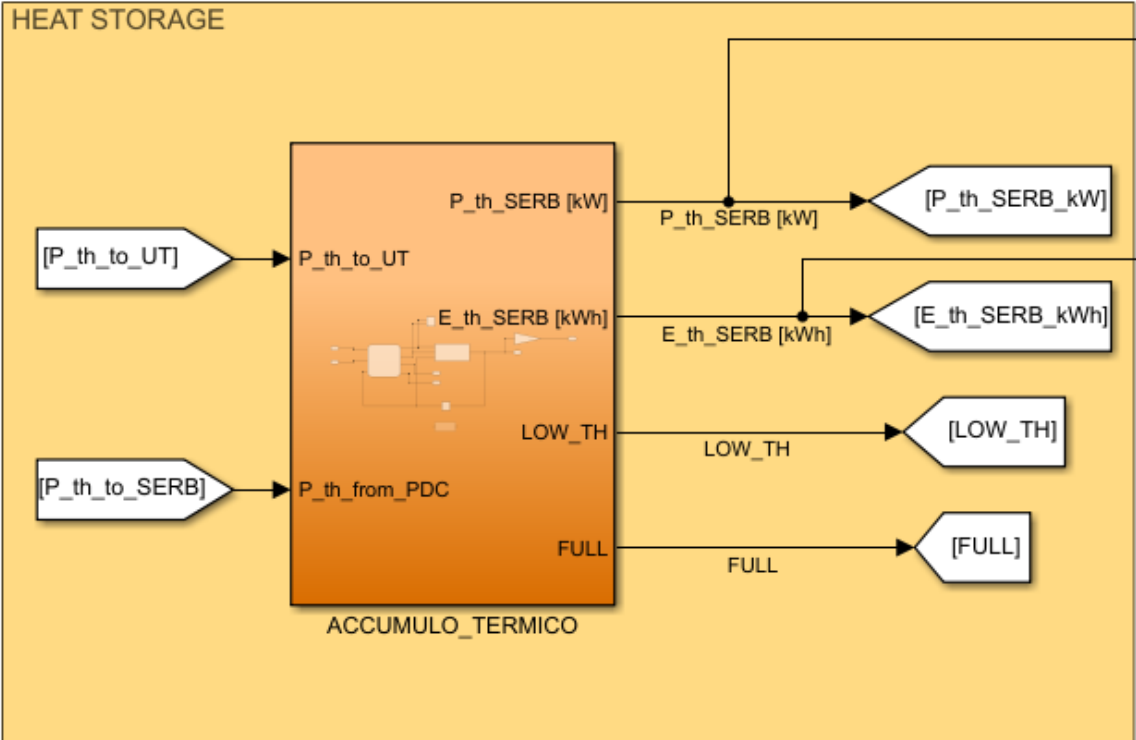


Figura 73 - vista complessiva del modello Simulink relativo al sistema pompa di calore - accumulo termico

2.5 Progettazione in Stateflow della strategia di gestione dei flussi energetici

I modelli di calcolo proposti finora sono stati realizzati in modo tale da simulare il comportamento di ciascun componente al variare dei suoi parametri operativi ma, al fine di riprodurre ed analizzare la risposta dell'intero impianto nelle diverse condizioni al contorno, è necessario includere in un'unica routine computazionale il modello di calcolo di ciascun elemento d'impianto ed implementare uno schema logico che ne consenta l'interazione in modo controllato. La possibilità di intervenire rapidamente nella variazione dei parametri operativi o sulla logica propria della strategia di regolazione del sistema rappresenta un importante principio in fase di modellazione, specialmente per il supporto che un'adeguata progettazione può restituire in termini di gestione degli errori e delle incongruenze logiche, tuttavia non sempre una programmazione di tipo grafico, come quella adoperata in ambiente Simulink, risulta essere la scelta migliore a tale scopo. A dispetto di una maggiore intuitività rispetto ai più tradizionali linguaggi di programmazione in codice, infatti, per i linguaggi grafici il migliore controllo dei processi logici, specialmente quando questi coinvolgono differenti modelli di calcolo indipendenti tra loro all'interno dello stesso sistema, richiede talvolta conoscenze molto approfondite e la modellazione può risultare un'operazione complessa e laboriosa che aumenta le probabilità di sfociare in errori semantici. Tuttavia, l'ampia libertà di personalizzazione dell'ambiente di lavoro di Simulink attraverso i numerosi strumenti e applicativi messi a disposizione da Mathworks, nonché la possibilità di ricorrere a blocchi di programmazione in codice di Matlab, garantisce una grande flessibilità al software che in questo modo consente di adattare la modellazione a seconda delle necessità dell'utente.

A tale proposito, per la messa a punto della logica decisionale ci si è serviti di Stateflow, ovvero un software concepito da Mathworks per potersi interfacciare perfettamente con Simulink e Matlab mediante una progettazione grafica che consente di avere un'esperienza ibrida tra i due. Stateflow offre infatti un linguaggio di programmazione grafico che include diagrammi e tabelle di transizione di stato, diagrammi di flusso e tabelle di verità tramite cui è possibile scegliere in che modo gli algoritmi di MATLAB e i modelli di Simulink debbano reagire a segnali di input, eventi e condizioni nel tempo. In questo modo si può quindi configurare la logica decisionale, combinatoria e sequenziale di un qualunque sistema, nonché simularla e rappresentarla come blocco in un modello Simulink o eseguirla come oggetto in MATLAB, inoltre, l'utilizzo di un linguaggio basato su animazioni grafiche consente di analizzare ed eseguire il debug dell'algoritmo realizzato mentre questo è ancora in esecuzione.

L'adozione di Stateflow come strumento di ottimizzazione all'interno dell'ambiente Simulink ha permesso quindi di procedere con la modellazione realizzando dapprima la calibrazione dei sottosistemi d'impianto che necessitavano una di logica propria per il funzionamento (**Figura 74**) e successivamente implementando la strategia di controllo dell'intero sistema, attraverso la quale stabilire le gerarchie tra tutti i componenti d'impianto e di conseguenza le priorità sui flussi energetici.

Nel seguito verrà descritta la procedura di modellazione delle logiche attraverso l'utilizzo di Stateflow per tutte le sezioni d'impianto calibrate ed infine la messa a punto del sistema di controllo complessivo, con particolare attenzione alla strategia di funzionamento individuata.

2.5.1 Calibrazione delle sottosezioni d'impianto e definizione delle logiche di funzionamento

Come già accennato, per una corretta comunicazione tra i modelli di calcolo sinora presentati è necessario suddividere il sistema complessivo nelle sue diverse sezioni, distinguendole a seconda del ruolo ricoperto durante il funzionamento e realizzandone la calibrazione sulla base delle condizioni al contorno considerate. Con il termine calibrazione si vuole indicare la procedura di definizione della logica decisionale, nonché di configurazione e settaggio di tutti i parametri necessari all'attivazione ed al funzionamento in autonomia del sottosistema considerato quando questo è coinvolto all'interno del modello di calcolo complessivo.

In **Figura 74** è mostrato il layout semplificato dell'impianto con le relative sottosezioni.

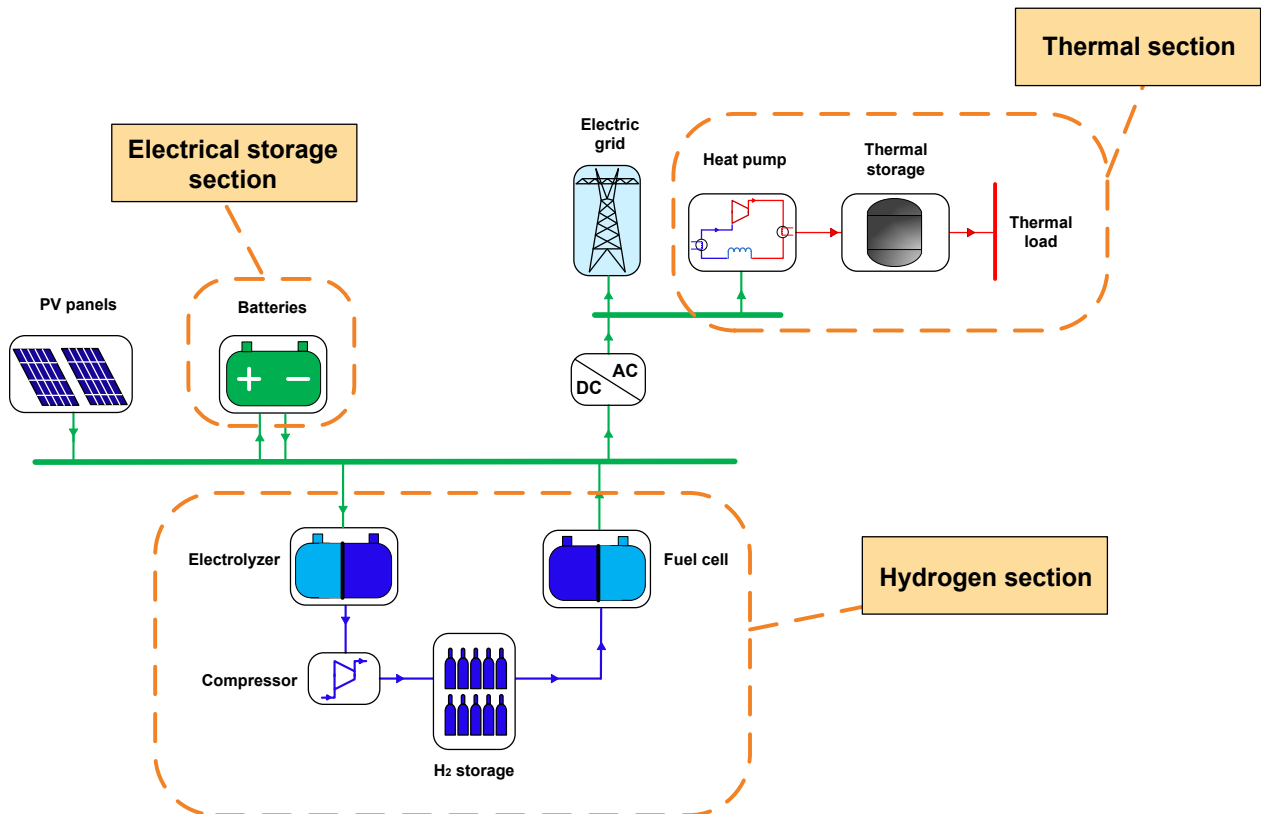


Figura 74 - layout del sistema modellizzato ed individuazione delle sue sottosezioni

Partendo dalla porzione d'impianto dedicata a produzione, accumulo e riconversione dell'idrogeno, essa è costituita da un elettrolizzatore, un sistema di stoccaggio pressurizzato e da una fuel cell, come illustrato in **Figura 75**.

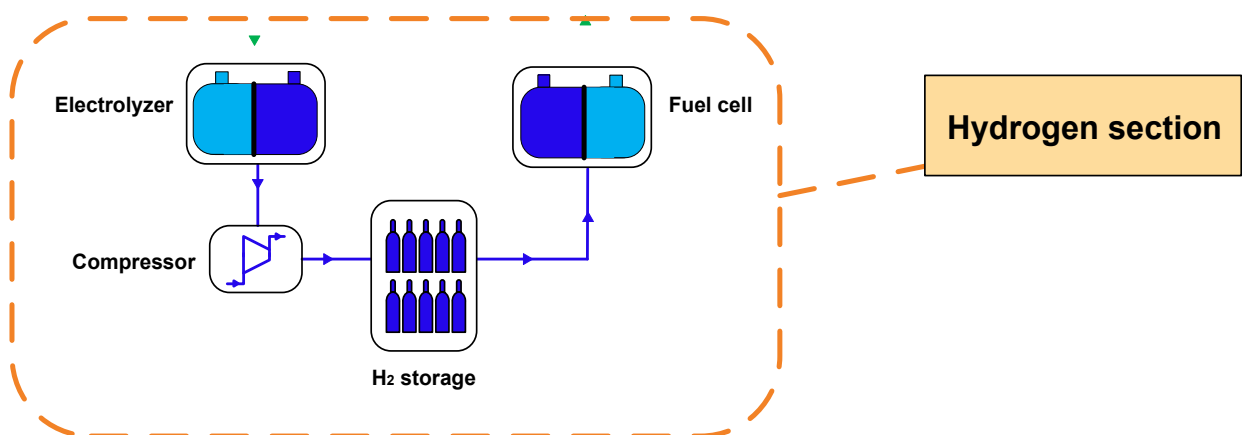


Figura 75 - sezione di produzione, compressione e stoccaggio dell'idrogeno

Nel caso specifico del sottosistema in esame, è stato necessario mettere a punto una logica che permettesse ai componenti di scambiarsi istruzioni autonomamente, sulla base delle condizioni operative, ed eventualmente interrompere il funzionamento laddove non vi fossero più i presupposti per l'attivazione della linea ad idrogeno. Le funzionalità offerte da Stateflow, come già detto, ben si prestano a tale scopo grazie alla possibilità di realizzare diagrammi di flusso e tabelle di verità posizionabili direttamente all'interno del modello Simulink ed interagenti con esso, pertanto si è adoperato un blocco Stateflow per implementare la macchina a stati che regolasse il sistema (**Figura 76**).

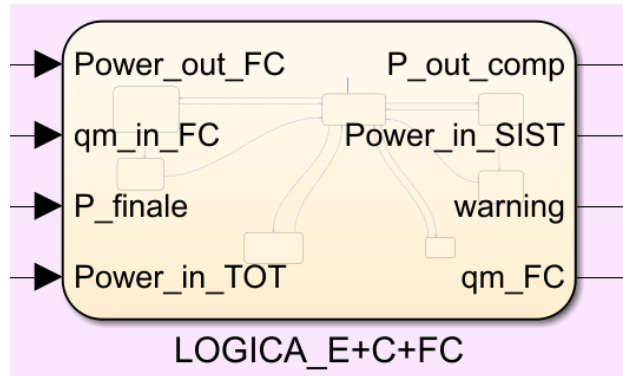


Figura 76 - blocco Stateflow inserito nel modello Simulink

In questo modo è stato possibile definire in maniera semplice e chiara la logica operativa ed i parametri per il monitoraggio delle fasi di produzione e riconversione dell'idrogeno, e quindi riempimento e svuotamento del serbatoio, nonché realizzare un controllo in retroazione che garantisca la giusta ripartizione di potenza elettrica in ingresso ad elettrolizzatore e compressore qualora vi fossero le condizioni per avviare la produzione (**Figura 77**). Inoltre, l'utilizzo della macchina a stati di Stateflow ha permesso di monitorare la logica durante l'esecuzione del codice ed individuarne repentinamente gli eventuali errori (**Figura 78** e **Figura 79**).

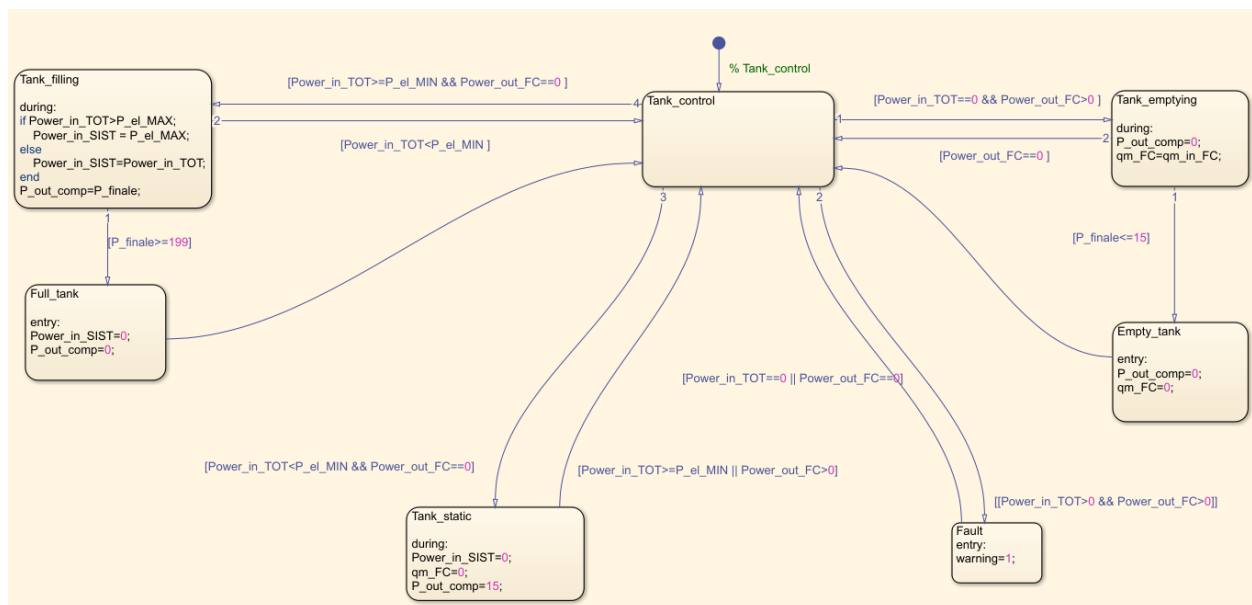


Figura 77 - diagramma di flusso della macchina a stati progettata mediante Stateflow

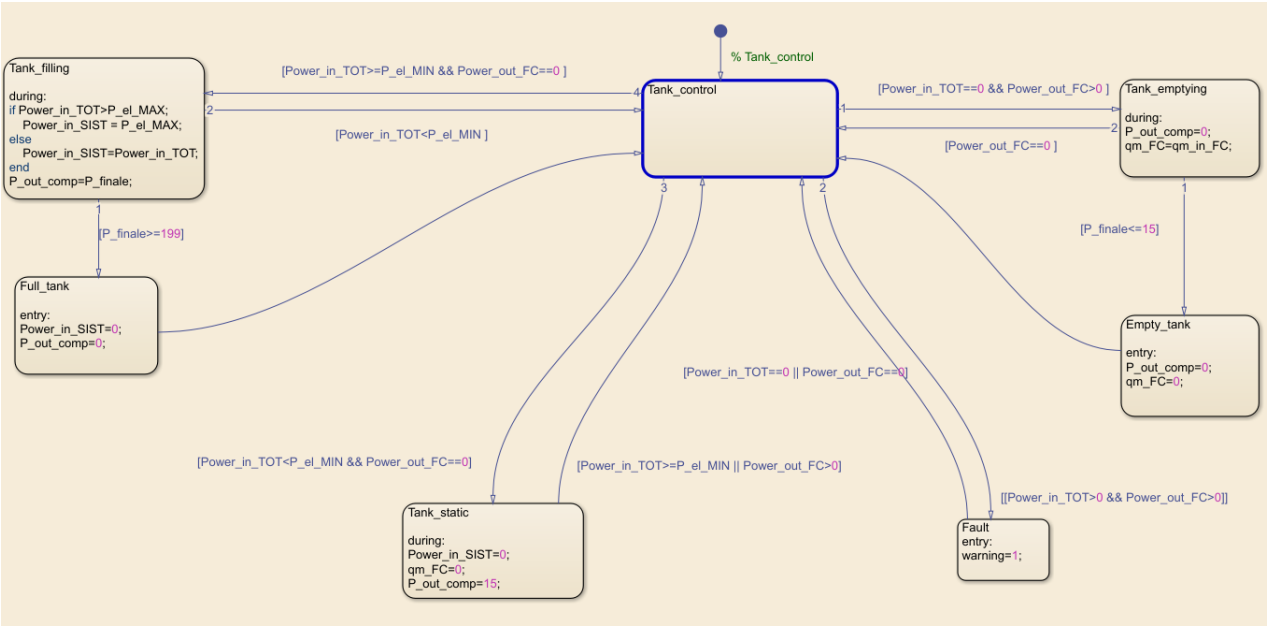
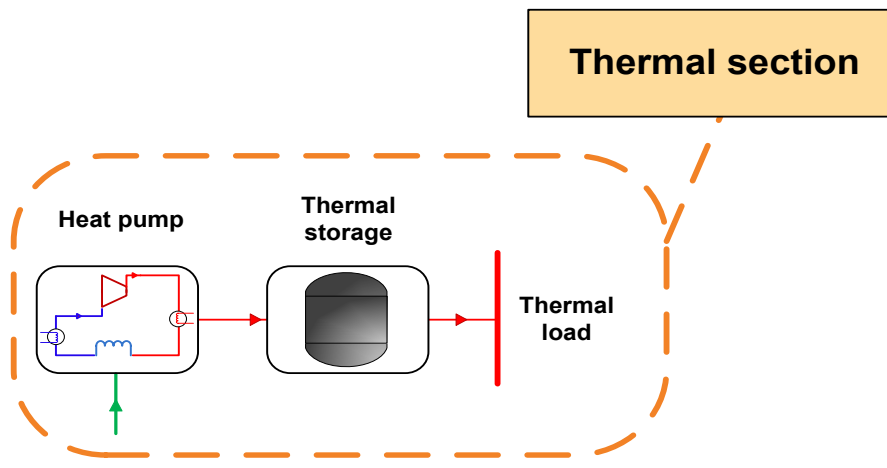


Figura 78 - attivazione degli stati correnti nel diagramma Stateflow durante l'esecuzione del codice

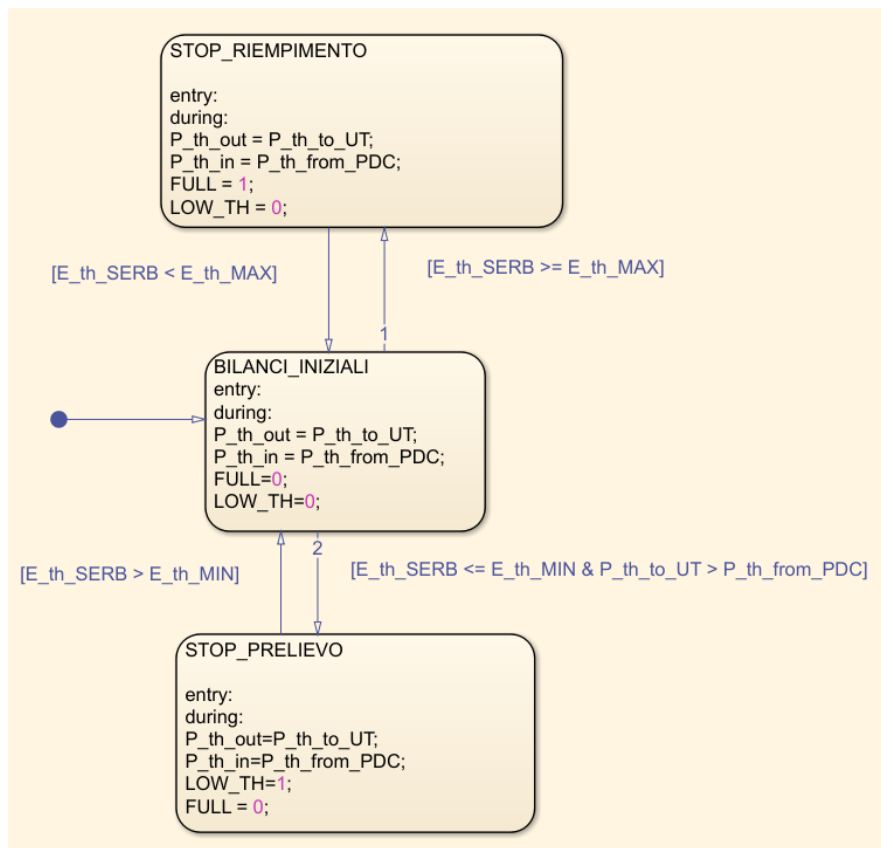
TYPE	NAME	VALUE	PORT
	P_finale		1
	Power_in_TOT		2
	Power_in_SIST		1
	Power_out_FC		3
	qm_FC		2
	warning		3
	P_out_comp		4
	P_el_MIN	(P_el_E_in_MIN+P_el_COMP_MAX)	
	P_el_MAX	(P_el_E_max+P_el_COMP_MAX)	
	qm_in_FC		4

Figura 79 - pannello di controllo delle variabili di Stateflow

Passando al sottosistema di produzione ed accumulo termico, esso, come mostrato in **Figura 80**, si compone di una pompa di calore di tipo aria/acqua ed un serbatoio di accumulo termico all'interno del quale viene immagazzinata l'acqua calda sanitaria da fornire all'utenza. Anche in questo caso è stato opportuno sviluppare una logica interna al sottosistema affinché i suoi elementi costituenti comunicassero tra loro in modo coerente con la logica d'insieme dell'intero sistema, riconoscendo in autonomia le condizioni di attivazione o meno rispetto al funzionamento dell'impianto complessivo. In particolare, si è potuto così mettere a punto un codice che scandisse le fasi di riempimento e prelievo dal serbatoio di accumulo, definendo alcune variabili di controllo tramite cui monitorare il livello del fluido al suo interno e comunicarlo al sistema prima di eseguire le istruzioni successive.



a)



b)

Figura 80 - a) sottosistema di produzione ed accumulo termico, b) macchina a stati della sezione d'impianto

Per quanto riguarda infine il sottosistema dedicato all'accumulo dell'energia elettrica proveniente dai pannelli fotovoltaici (**Figura 81**), pur non essendo questo costituito da più di un componente, è stato necessario regolarne la logica interna mediante Stateflow a causa del fatto che le fasi di carica e scarica della batteria vengano gestite separatamente, seppure all'interno dello stesso modello di calcolo.

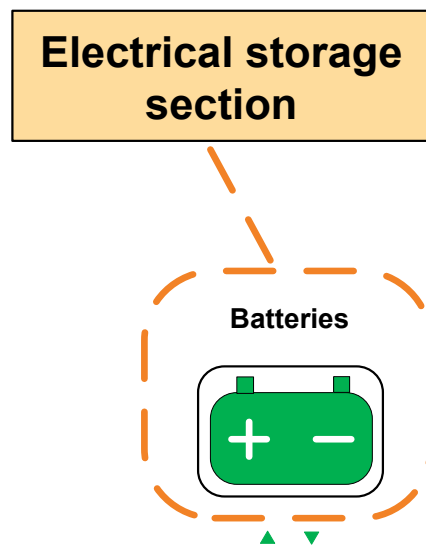


Figura 81 - sottosistema di accumulo elettrochimico

Per poter interfacciare correttamente il dispositivo di accumulo elettrochimico con il resto del sistema è stato infatti necessario stabilire le gerarchie tra i componenti in modo tale da indirizzare i flussi energetici secondo la strategia di regolazione scelta per il funzionamento complessivo e, a tal proposito, si è deciso di implementare la gestione delle fasi di carica e scarica della batteria direttamente all'interno del blocco Stateflow principale, ovvero quello dedicato alla programmazione della logica decisionale dell'intero sistema.

Uno schema rappresentativo della strategia di regolazione dell'impianto è mostrato in [Figura 83](#).

Il diagramma di flusso mostrato stabilisce quindi la logica con cui avvengono gli scambi energetici tra i componenti di tutto l'impianto durante la simulazione di funzionamento e, di conseguenza, il blocco Stateflow che lo rappresenta attraverso una macchina a stati ([Figura 82](#)) costituisce senz'altro il componente più importante di tutto il modello di calcolo realizzato.

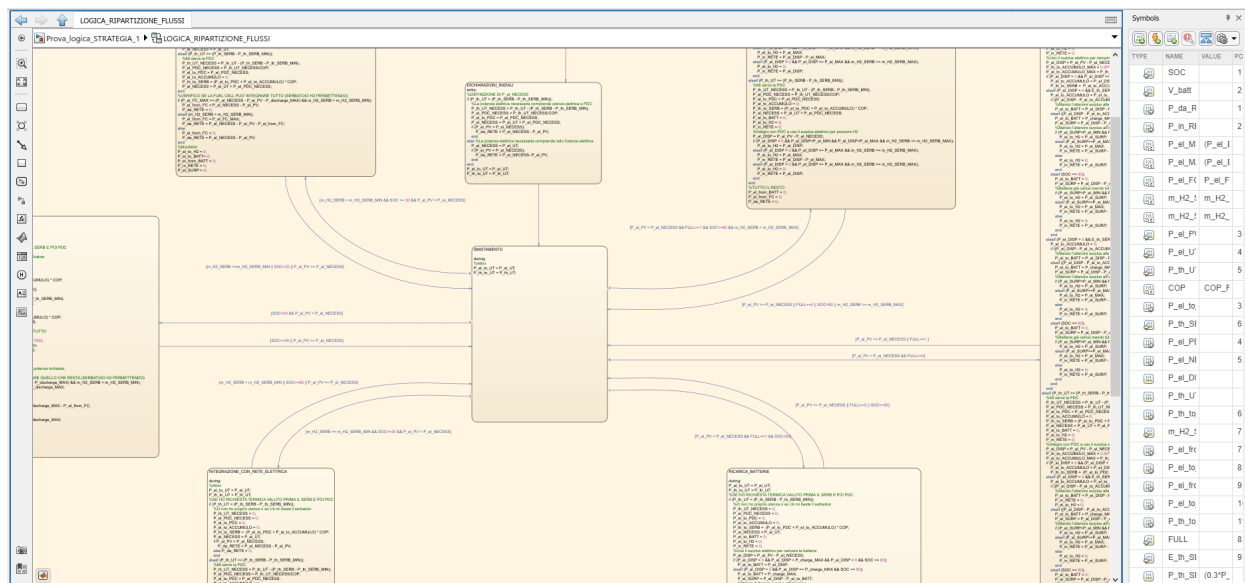


Figura 82 - macchina a stati del modello complessivo

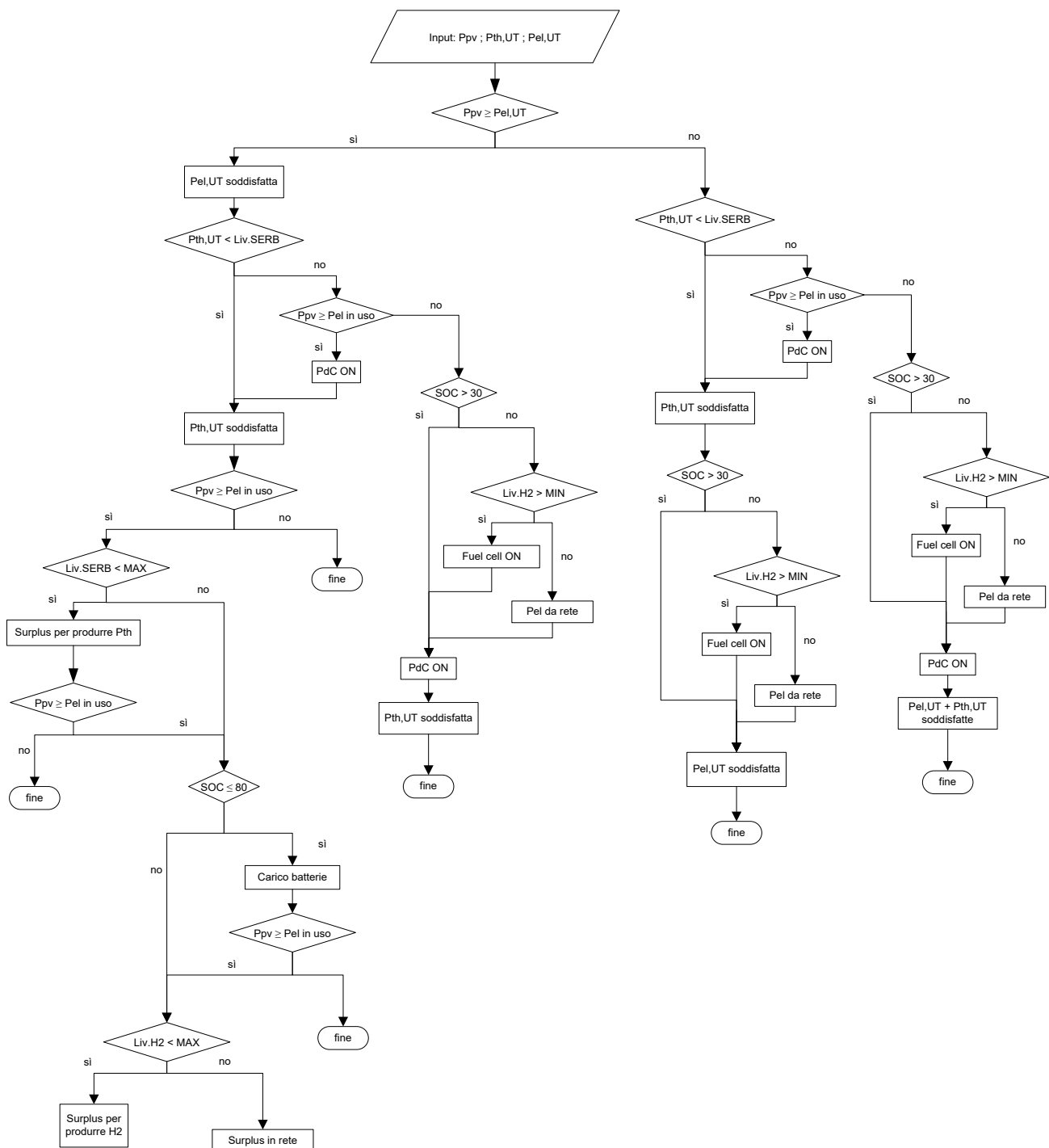


Figura 83 - flowchart della strategia di funzionamento del sistema complessivo

Come mostrato dalla flowchart della strategia di funzionamento, per le fasi di sovrapproduzione elettrica da parte dei generatori fotovoltaici si è ipotizzato in prima istanza di dare la precedenza all'accumulo dell'energia termica, mediante l'accensione programmata della pompa di calore, e soltanto in seguito al raggiungimento del livello massimo previsto per l'accumulo inerziale, od eventualmente in caso di un surplus di potenza elettrica superiore al massimo assorbimento della pompa di calore, passare all'accumulo elettrochimico mediante le batterie ed infine alla conversione in idrogeno.

2.6 Programmazione delle schede Arduino e messa a punto del protocollo di comunicazione Arduino-Simulink

Come già brevemente descritto in fase di introduzione, l'attività di emulazione si è basata sull'utilizzo di alcune schede elettroniche Arduino Mega 2560 da interfacciare con il software di modellazione e controllo Simulink per la riproduzione del comportamento di alcuni componenti d'impianto tra quelli presentati in precedenza.

A tale scopo è stato necessario reperire informazioni sulla possibilità di interconnettere le schede elettroniche al modello in esecuzione su Simulink, concentrandosi sulla necessità di mettere a punto un protocollo di scambio dati costruito sulla base delle esigenze operative. Quanto già detto circa la possibilità di programmare i microcontrollori adottati direttamente in ambiente Simulink attraverso le librerie di supporto fornite dalla community di Mathworks è necessario a comprendere le problematiche riscontrate durante questa fase della sperimentazione. Lo strumento più largamente utilizzato per poter adoperare le piattaforme Arduino in ambiente Matlab/Simulink è certamente il SPFAH (*Matlab/Simulink Support Package for Arduino Hardware*), vale a dire una libreria di funzioni precostituite tramite cui sfruttare le potenzialità di Arduino in modalità "plug&play", senza cioè doversi preoccupare della sua programmazione in codice attraverso l'IDE né del protocollo di comunicazione utilizzato, se non mediante semplici operazioni di impostazione (Figura 84). Tale caratteristica costituisce un grande vantaggio per tutti gli utenti che necessitano di acquisire segnali digitali o analogici provenienti da sensori connessi al chip o realizzare sistemi di controllo in tempo reale per piccoli apparati elettronici ed elettromeccanici quali dispositivi audio, LED, motori elettrici ed attuatori. In questo modo è infatti possibile servirsi del solo linguaggio grafico proprio di Simulink, senza passare dall'IDE di Arduino, per poter programmare la scheda e successivamente caricare il modello realizzato su di essa mediante alcuni comandi rapidi dedicati del pacchetto software.



Figura 84 - modalità di programmazione e controllo SPFAH [33]

Le funzionalità messe a disposizione dalla libreria sono quindi limitate ad un certo tipo di applicazioni e di conseguenza le modalità di comunicazione tra software e scheda sono subordinate ad esse, risultando inappropriate dal punto di vista informatico se si cerca una maggiore libertà nello scambio dati.

Per tali ragioni è stato necessario estendere la ricerca di uno strumento informatico che consentisse di comunicare liberamente con le schede Arduino attraverso Simulink ma che allo stesso tempo non rappresentasse un ostacolo dal punto di vista delle competenze necessarie per il suo utilizzo in materia di programmazione in codice. Il miglior compromesso ottenuto in questa sede è costituito dalla libreria ICT (*Instrument Control Toolbox*), fruibile anch'essa come il precedente pacchetto all'interno della community di Mathworks e, molto utilizzata proprio per le sue caratteristiche di ampia personalizzazione. Tale strumento, permette infatti di connettere direttamente Matlab/Simulink a dispositivi elettronici di

misurazione ed analisi (**Figura 85**), quali oscilloscopi, emulatori di carico ed alimentatori, mediante driver riconosciuti automaticamente o protocolli di comunicazione personalizzati tra i più comuni (GPIO, VISA, TCP/IP e UDP). Il toolbox è inoltre dotato di un supporto integrato per l'utilizzo dei protocolli seriali TCP/IP, UDP, I2C, SPI e Bluetooth per la comunicazione remota con altri computer e schede a circuiti integrati (PCB) da MATLAB. Sono incluse nella raccolta anche alcune funzioni e applicazioni per l'utilizzo del protocollo MODBUS, che consentono la comunicazione con attrezzature di automazione industriale come i controllori logici programmabili (PLC) e i controllori di automazione programmabile (PAC).

Supported Hardware



Figura 85 - dispositivi elettronici compatibili con l'ICT [33]

Servendosi quindi del pacchetto ICT è stato possibile mettere a punto un protocollo di comunicazione molto semplice per realizzare lo scambio dati in tempo reale tra le schede Arduino e Simulink, passando però in questo caso attraverso la programmazione separata dei microcontrollori mediante l'IDE di Arduino. A differenza del pacchetto SPFAH, infatti, il cui l'intento è proprio quello di poter programmare le schede caricando ed eseguendo direttamente in loop su di esse il modello di calcolo realizzato in Simulink, nel caso dell'ICT le funzionalità offerte si concentrano principalmente sulla messa a punto del protocollo di comunicazione tra il software ed altri dispositivi elettronici, al fine di semplificare ed ottimizzare le fasi di testing ed analisi dei modelli di calcolo sviluppati.

Se da un lato la necessità di editare separatamente i codici di istruzioni per ciascuna scheda Arduino rappresenta una complicazione della procedura rispetto a quanto visto in precedenza, la possibilità di gestire contemporaneamente lo scambio dati con più dispositivi all'interno dello stesso modello di calcolo Simulink costituisce il vero vantaggio di questa scelta.

2.6.1 Procedura di configurazione dei dispositivi Arduino e del modello Simulink

La modalità di comunicazione adottata per lo scambio dati tra schede e PC è stata incentrata su una trasmissione di tipo seriale, fisicamente supportata da una connessione standard USB (*Universal Serial Bus*).

In questo modo, servendosi di un hub USB, è stato possibile connettere tutti i dispositivi Arduino al computer e permetterne il flusso di dati bidirezionale. Uno schema esemplificativo dell'architettura realizzata è proposto in **Figura 86**.

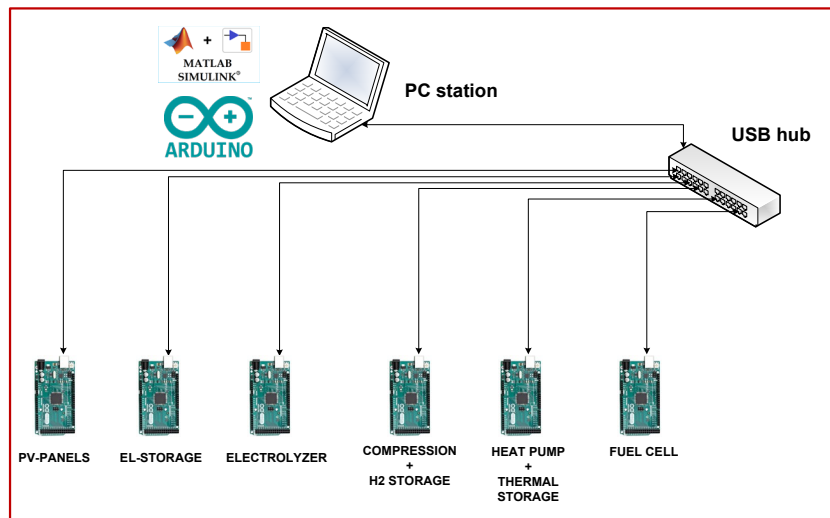


Figura 86 - Architettura del sistema elettronico realizzato

Una volta realizzati i collegamenti fisici per tutta la strumentazione hardware, si è potuto approntare il protocollo di comunicazione software mediante l'utilizzo del pacchetto ICT (Figura 87), più in particolare servendosi dei blocchi *Serial Configuration*, *Serial Send* e *Serial Receive*. Come già detto, tali funzionalità hanno consentito di configurare la sola modalità di trasmissione dei dati e, pertanto, per la programmazione delle schede Arduino è stato necessario procedere attraverso l'ambiente di sviluppo proprio di quest'ultimo.

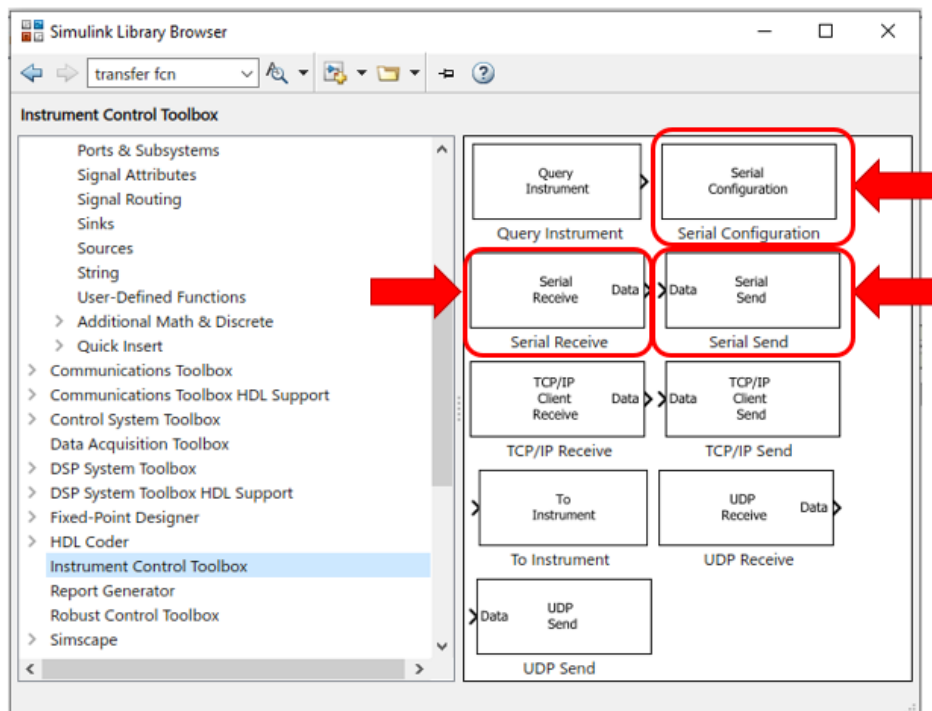


Figura 87 - funzionalità della libreria ICT

Per presentare la metodologia seguita durante questa fase si procederà dapprima descrivendo le operazioni effettuate in ambiente Simulink e successivamente si passerà alla parte di programmazione in codice Arduino delle schede, considerando d'ora in poi, a titolo puramente esemplificativo, quanto fatto per il pannello fotovoltaico, in quanto la procedura in questione risulta esattamente identica per ciascuno dei modelli di calcolo relativi ai componenti sopra esposti.

Avendo scelto di connettere la strumentazione attraverso lo standard USB, è stato anzitutto necessario configurare le porte COM (Communication port) emulate dagli adattatori USB ed assegnate a ciascun dispositivo connesso al PC. Attraverso tali assegnazioni è infatti possibile rilevare e riconoscere automaticamente le schede Arduino connesse ed impostare per ognuna di queste un canale di trasmissione dati dedicato. Tale configurazione è operabile attraverso il blocco *Serial Configuration* (Figura 88), accessibile come già detto dalla libreria ICT, tramite cui è possibile impostare il numero della porta COM assegnata dal PC al dispositivo Arduino, visualizzabile direttamente dall'IDE dello stesso o dal pannello di controllo del calcolatore, ed anche alcuni altri parametri come il *baud rate*. Quest'ultimo è un parametro caratteristico dei sistemi di telecomunicazioni ed in particolare rappresenta la velocità di trasmissione dei dati in simboli al secondo, dove ogni simbolo può essere costituito da uno o più bit di dati. I valori baud rate standard tipicamente impiegati per le applicazioni con Arduino sono 9600 e 115200 ma è possibile modificare tale parametro attraverso la funzione di setup accessibile dall'ambiente di sviluppo della scheda. Per quanto riguarda le voci restanti è possibile mantenere le impostazioni di default ai fini della procedura.

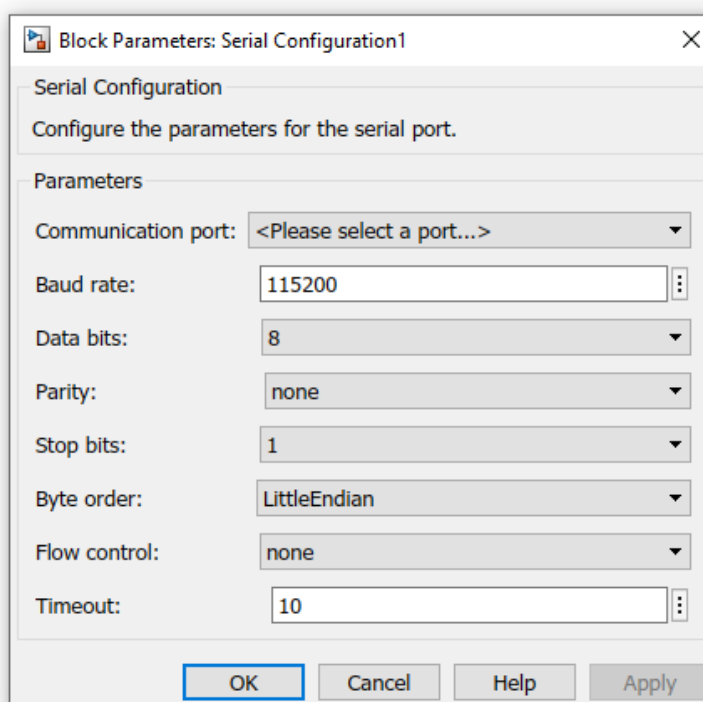
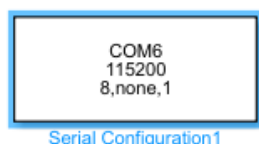


Figura 88 - Serial configuration

Procedendo ora nell'intento di inviare dei dati da Simulink al dispositivo Arduino assegnato alla porta COM in questione, è necessario servirsi dei seguenti blocchi connessi tra loro in sequenza, come mostrato in Figura 89:

- **Input** o blocco rappresentante la sorgente del segnale da inviare

- **Mux** o multiplexer, ovvero un collettore di dati in grado di raccogliere tutti i segnali in ingresso, qualora fossero più di uno, e combinarli in un vettore di uscita
- **Rate transition**, per impostare la velocità di invio del pacchetto dati
- **Cast to single**, per convertire ciascun dato da passare ad Arduino in un dato di tipo *single* (32 bit, 4 byte)
- **Byte pack**, per raccogliere tutti i dati di tipo *single* ed organizzarli in un vettore del tipo selezionato, in questo caso *uint32*.
 - **Serial send**, per inviare infine il pacchetto di dati finora configurato al dispositivo esterno mediante il protocollo di comunicazione seriale adottato.

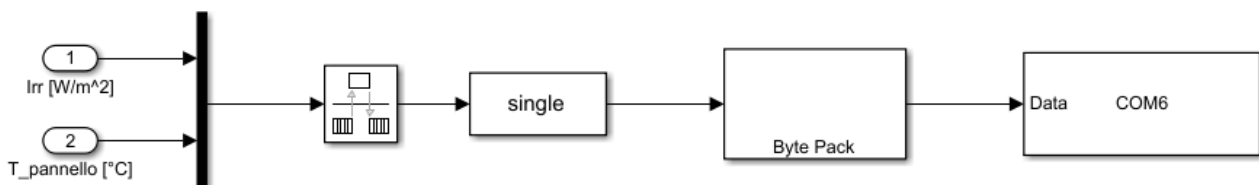


Figura 89 - blocchi per l'invio di dati da Simulink ad Arduino

Una volta passato il pacchetto di informazioni alla scheda Arduino, si potrà scegliere di elaborare questi dati attraverso una serie di istruzioni per poi restituire al calcolatore i risultati finali delle operazioni mediante una procedura di codifica e decodifica del tutto analoga a quella appena presentata. Lasciando per ora in sospeso la programmazione in codice del dispositivo, della quale si parlerà più avanti, si assuma che la fase di rielaborazione dei dati ricevuti sia andata a buon fine e che vi siano le condizioni per un nuovo scambio di informazioni, questa volta nel verso opposto.

I blocchi di funzioni del pacchetto ICT da utilizzare in tal caso sono (Figura 90):

- **Serial receive**, per ricevere il pacchetto di informazioni codificate da Arduino
- **Cast to double**, per convertire ciascun dato del pacchetto in arrivo nel formato *double*
- **Output** o strumento di visualizzazione del dato ricevuto.

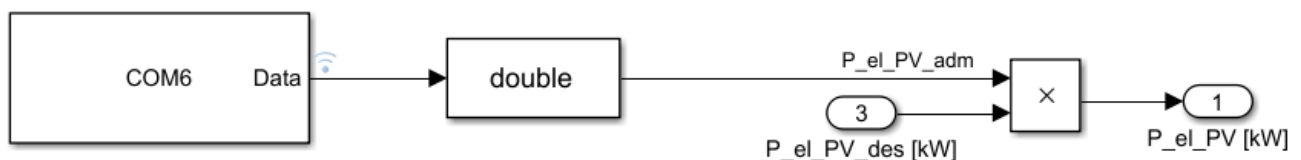


Figura 90 - blocchi per l'invio di dati da Arduino a Simulink

Tramite quest'altra disposizione i pacchetti di dati elaborati dal codice eseguito su Arduino verranno di volta in volta restituiti al calcolatore e riconvertiti in formato *double* per essere riutilizzati all'interno del modello Simulink.

In questo modo il layout complessivo del modello realizzato si presenterà come in **Figura 91**.

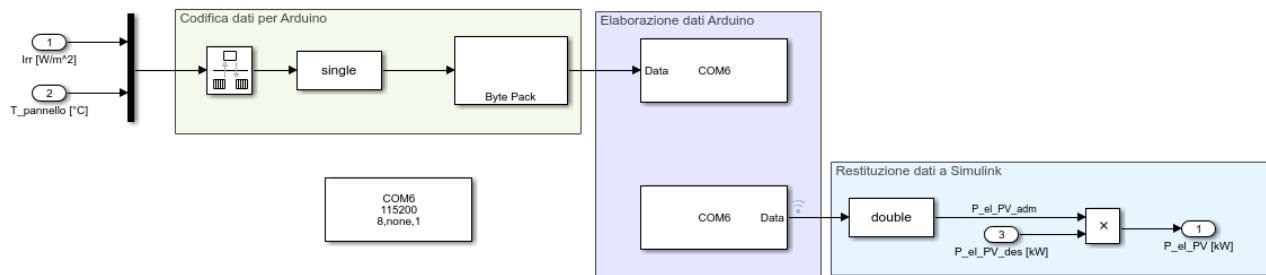


Figura 91 - layout modello Simulink per lo scambio dati con Arduino

Come per il blocco *Serial configuration*, anche *Serial receive* (**Figura 93**) e *Serial send* (**Figura 92**), nonché i relativi blocchi complementari, necessitano di essere adeguatamente configurati attraverso le finestre di impostazioni associate a ciascuno di essi. Sono mostrate nel seguito le schermate relative ai parametri selezionati per ogni blocco durante le fasi di invio e ricezione.

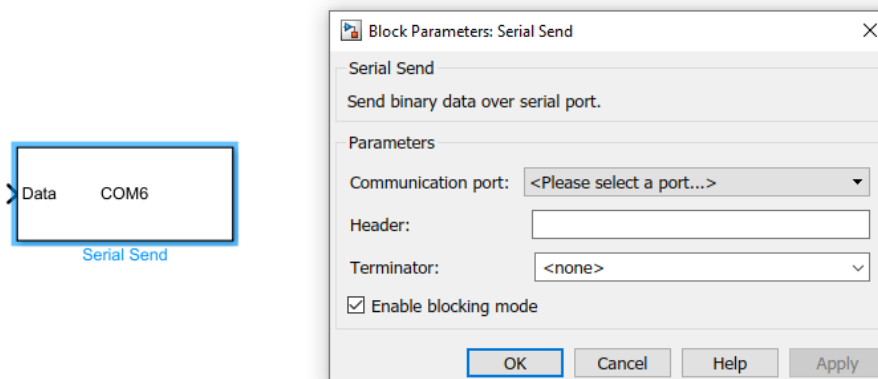


Figura 92 - Serial Send

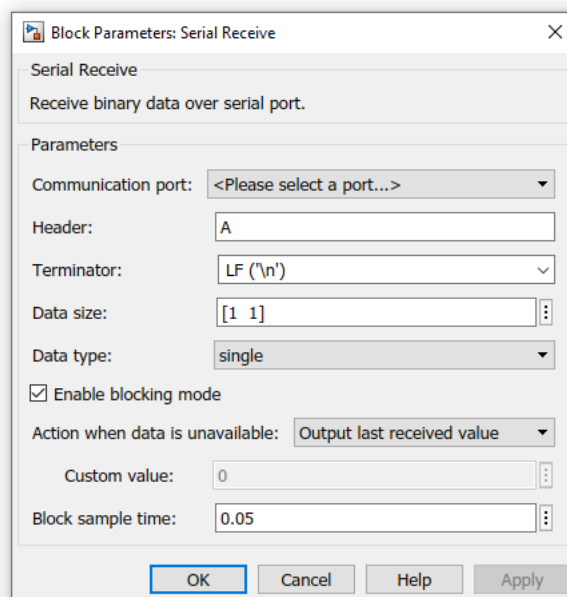
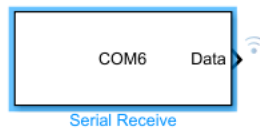


Figura 93 - Serial Receive

Con riferimento a [Figura 92](#) e [Figura 93](#), dopo aver impostato correttamente il numero della porta COM corrispondente al dispositivo in questione, vi è la possibilità di modificare alcuni altri parametri quali ad esempio le variabili *Header* e *Terminator*, ovvero due stringhe di testo che è possibile stampare sul monitor seriale di Arduino all’inizio ed alla fine di ogni loop, in modo tale da verificarne la sincronizzazione temporale con il PC durante l’esecuzione del programma. Per quanto riguarda invece il solo blocco Receive è invece importante prestare attenzione alla voce *Data size* che serve a specificare la dimensione del vettore di dati che si vuole ricevere da Arduino. Nel caso del modello di calcolo relativo al pannello fotovoltaico tale vettore è definito da una sola posizione, una riga ed una colonna, poiché la variabile da restituire a Simulink corrisponde solamente al valore di potenza elettrica generata dal pannello, tuttavia, qualora vi fossero più variabili di output per uno stesso modello di calcolo è essenziale indicare correttamente il numero di elementi che compongono il vettore di ritorno altrimenti non sarà possibile visualizzare correttamente tutti i dati elaborati.

Quanto fatto finora consente di impostare lo scambio dati personalizzato con Arduino dal solo punto di vista del software Simulink ma prima di riuscire effettivamente a comunicare con i dispositivi è necessario, come anticipato, programmare la scheda attraverso un codice direttamente compilato nell’ambiente di sviluppo integrato della piattaforma. Prendendo sempre come esempio il caso del pannello fotovoltaico, un semplice codice rappresentativo del modello di calcolo corrispondente al componente in questione è mostrato di seguito, in [Figura 94](#).

Come è possibile notare, anche per quanto riguarda l’IDE di Arduino è necessario approntare una funzione apposita che permetta di scomporre il pacchetto di dati binari in arrivo in un array di 4 bytes denominati *uint8_t* e successivamente, di ricostituire tale pacchetto con i nuovi dati elaborati prima di reindirizzarlo al calcolatore. Una volta poi inizializzata la porta seriale all’interno del *void setup* attraverso l’istruzione *Serial.begin* ed indicato il valore di baud rate che si vuole impostare per la comunicazione, in questo caso 115200, è possibile compilare il *void loop*, ovvero la sezione dedicata alle vere e proprie istruzioni da eseguire su Arduino, che in questa sede corrispondono essenzialmente ad operazioni di tipo algebrico-matematico. Includendo all’inizio del codice la libreria dedicata *Math.h* è infatti possibile servirsi di numerose funzioni precostituite grazie alle quali risulta immediato elaborare dati e variabili in ingresso.

Anche in questo caso è bene pianificare delle istruzioni per la stampa di Header e Terminator, in modo da avere sempre un riscontro della sincronizzazione tra scheda e PC.

Nel capitolo seguente verranno presentate le simulazioni di funzionamento realizzate sul modello Simulink e le prove di comunicazione dell'apparato di emulazione Arduino-Simulink, del quale si evinceranno le criticità e le eventuali implementazioni

Nel prossimo paragrafo verrà applicata al sistema modellizzato la routine di calcolo precedentemente descritta e saranno evidenziati pregi e difetti della modalità di comunicazione adottata dall'architettura Simulink - Arduino ai fini della realizzazione di un apparato sperimentale di emulazione.

```

1  #include <math.h>
2
3  // Decodifica e conversione del pacchetto dati in bytes
4  typedef union {
5  float number;
6  uint8_t bytes[4];
7  } FLOATUNION_t;
8
9  // Definizione ed inizializzazione variabili
10 FLOATUNION_t Irraggiamento; //[W/m^2]
11 FLOATUNION_t T_pannello; //[°C]
12 FLOATUNION_t P_el_PV_adm;//[ -]
13
14 void setup() {
15     Serial.begin(115200);
16 }
17
18 void loop(){
19     //Assegnazione dati Simulink alle variabili Arduino
20     do {
21         Irraggiamento.number = getFloat(); // Give your float a value
22         T_pannello.number = getFloat(); // Give your float a value
23         delay(50);
24     } while (Irraggiamento.number > 1000 or
25             Irraggiamento.number < 0 or
26             T_pannello.number != 42);
27
28
29     // Operazioni e calcoli
30     P_el_PV_adm.number = -0.013484962 + 0.001027414 * Irraggiamento.number
31     - 0.000597143 * T_pannello.number;
32
33     // Stampa dell'Header: Importante per la corretta sincronizzazione
34     Serial.write('A');
35
36     // Stampa del dato da restituire a Simulink
37     for (int i=0; i<4; i++){
38         Serial.write(P_el_PV_adm.bytes[i]);
39     }
40
41     // Stampa Terminator
42     Serial.print('\n');
43
44     // Delay: Usare lo stesso dei blocchi Simulink
45     delay(50);
46 }
47
48 //Funzione per scomporre il pacchetto dati proveniente da Simulink
49 float getFloat(){
50     int cont = 0;
51     FLOATUNION_t f;
52     while (cont < 4 ){
53         f.bytes[cont] = Serial.read();
54         cont = cont +1;
55     }
56     return f.number;
57 }

```

Figura 94 - esempio del codice di calcolo implementato sulle schede Arduino

3 Simulazione di funzionamento del sistema energetico S.A.P.I.E.N.T.E. e test dell'apparato di emulazione sperimentale

Il presente capitolo è relativo alla descrizione della fase di test sul banco prova ideato e installato al fine di verificarne la sua piena operatività come strumento per la programmazione di centraline per il controllo delle reti complesse.

A tal fine, come precisato nei capitoli precedenti, il banco prova è stato settato al fine di riprodurre il sistema energetico S.A.P.I.E.N.T.E. integrato con una sezione di produzione, stoccaggio e conversione di idrogeno.

Come ipotesi per l'inserimento della sezione ad idrogeno all'interno del sistema S.A.P.I.E.N.T.E. si è assunto che la stessa potesse essere impiegata come strategia di accumulo su base temporale più ampia rispetto alle batterie presenti nell'impianto, imponendo – rispetto alla disponibilità di surplus elettrico proveniente dalla generazione fotovoltaica – un livello gerarchico inferiore rispetto al sistema di accumulo elettrochimico. Sulla base della logica implementata, il comportamento atteso – fermo restando il pieno soddisfacimento del fabbisogno elettrico e termico dell'utenza – è quello di una prevalente produzione di idrogeno durante i mesi estivi ed un suo utilizzo come strategia di accumulo su base settimanale. Si sottolinea che l'ipotesi di un accumulo stagionale (produzione durante i mesi estivi e impiego prevalente durante i mesi invernali) non risulta compatibile con i volumi e le massime pressioni previste per lo stoccaggio di idrogeno.

Per ciò che concerne i dati di input, nei grafici in [Figura 95](#) , [Figura 96](#) , [Figura 97](#) sono mostrati rispettivamente la produzione elettrica dell'impianto fotovoltaico durante l'anno e le richieste di potenza elettrica e termica dell'utenza su base annuale, a cominciare dal mese di aprile e considerando uno step di calcolo pari a 5 minuti. È infine bene ricordare che è stata limitata la finestra operativa delle batterie da uno stato di carica massimo, pari all'80% della capacità, e da uno minimo, imposto al 30%; inoltre, per una maggiore fedeltà ai reali processi di carica del dispositivo, è stato imposto un valore limite della corrente di carica fissato al 30% della capacità.

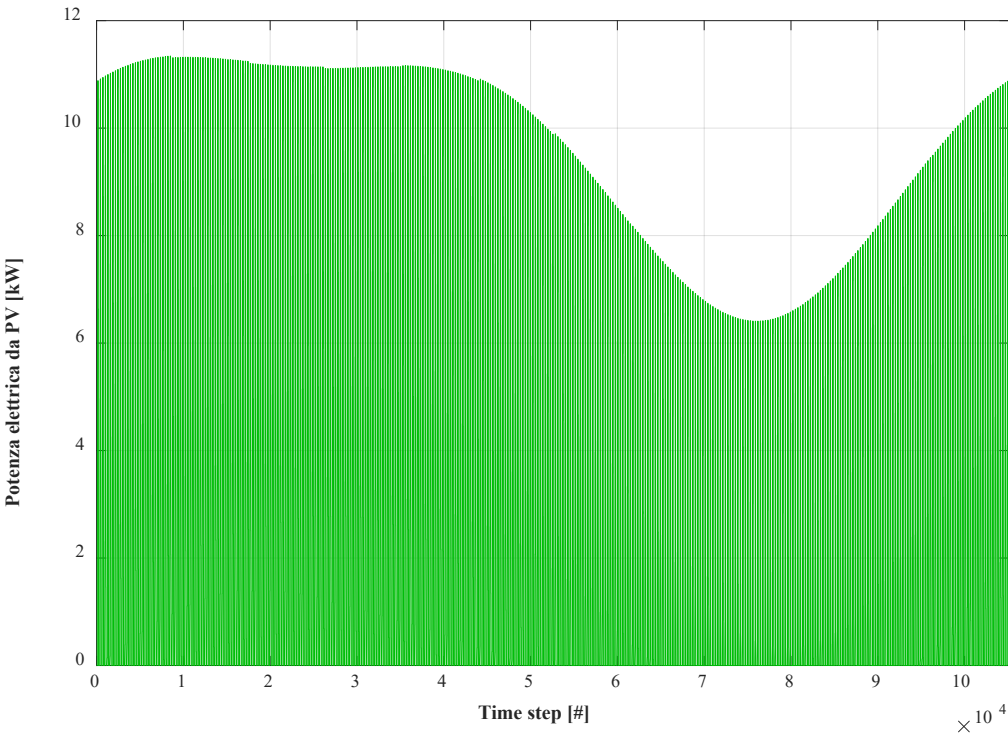


Figura 95 - potenza elettrica da PV per il periodo considerato (time step: 5 minuti)

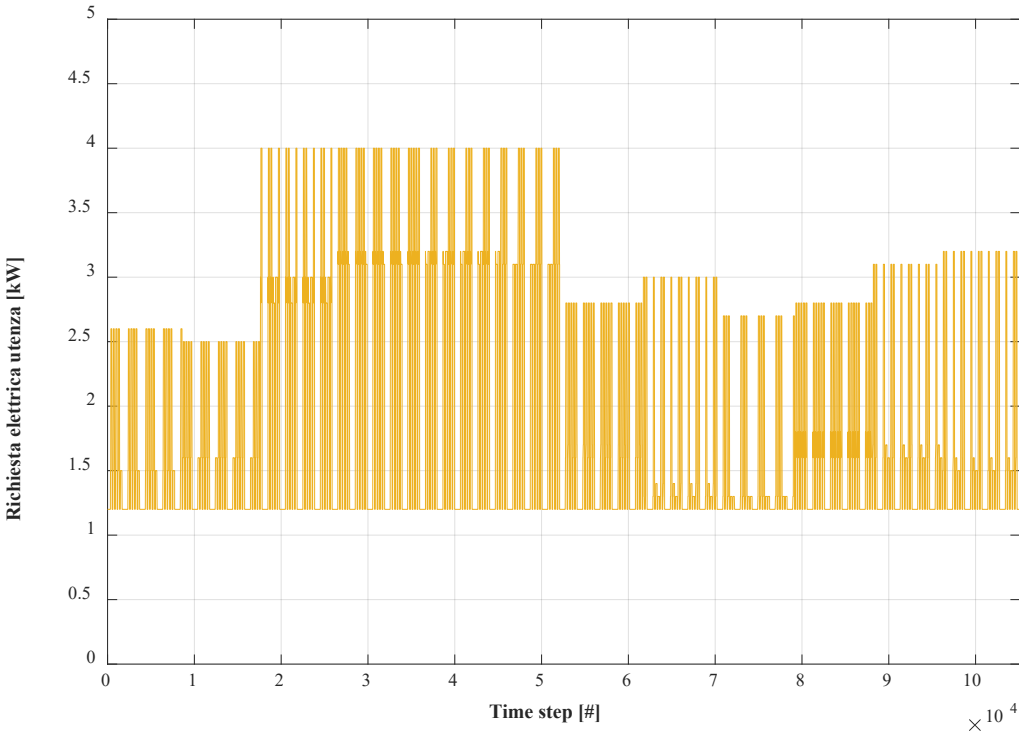
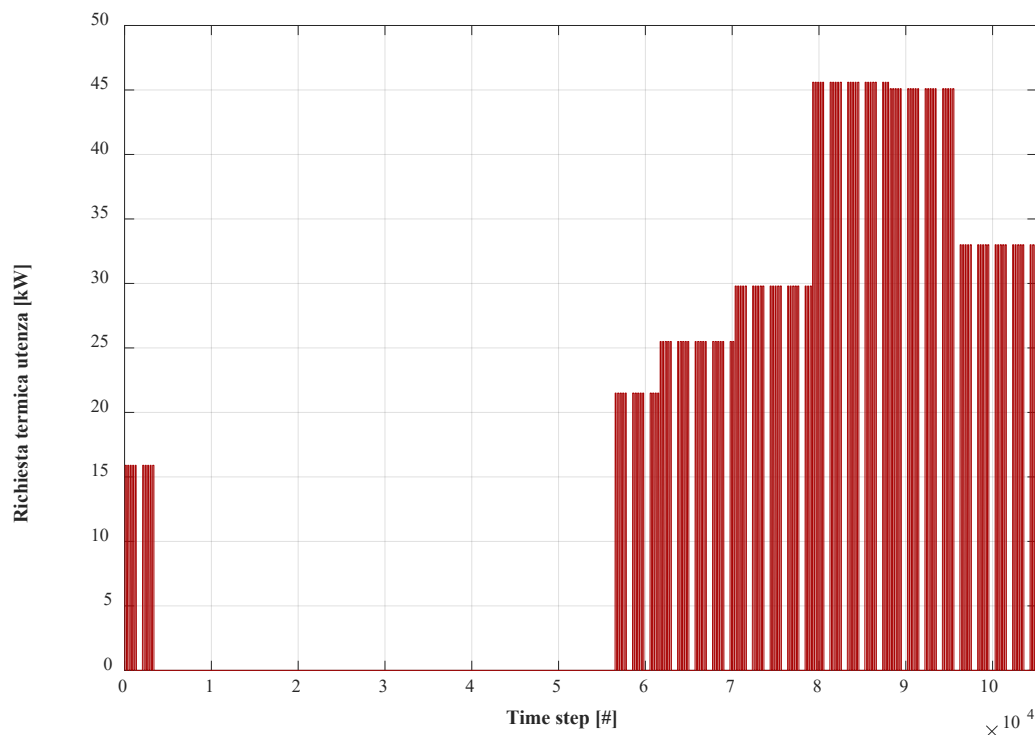


Figura 96 - richiesta elettrica dell'utenza per il periodo considerato (time step: 5 minuti)



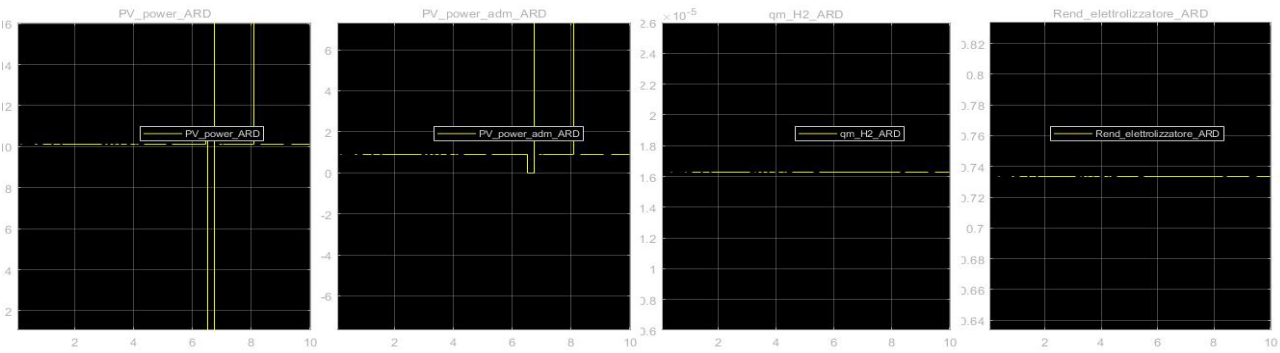


Figura 98 - segnali di output instabili

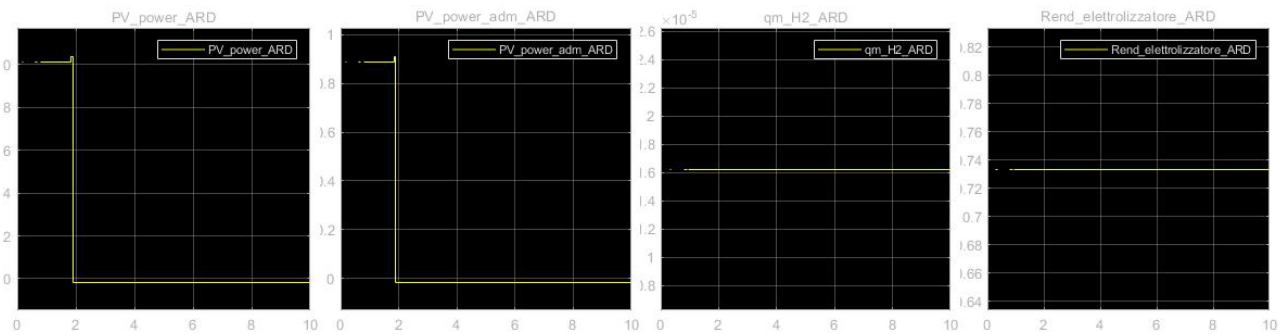


Figura 99 - segnali di output stabilizzati

Fase 2: Riproduzione su base annuale del modello S.A.P.I.E.N.T.E.

Durante tale fase è stato simulato il funzionamento del sistema per fissate condizioni di design su base annuale e considerando un timestep di 5 minuti. Tale fase ha permesso di validare la logica di smistamento dei flussi pensata attraverso la creazione della macchina a stati sviluppata in Stateflow.

A partire dai risultati ottenuti durante la prima annualità di progetto è fornita la **Tabella 6**, riassuntiva delle taglie dei principali sistemi inseriti nella simulazione svolta.

Tabella 6 - taglie e parametri operativi principali del sistema

fotovoltaico	11.4 kW nominali
batterie	C = 650 Ah, SOC _{min} = 30%, SOC _{max} = 80%
elettrolizzatore	5 kW nominali (operante a carico flottante)
cella a combustibile	6 kW nominali
accumulo termico	100 kWh, livello minimo pari al 30% del totale
serbatoio di idrogeno	1 m ³ , range di utilizzo da 15 bar a 200 bar

In **Figura 100** è riportata la richiesta elettrica della pompa di calore, il cui andamento, unitamente a quello del livello di acqua calda sanitaria presente nell’accumulo termico e mostrato in **Figura 101**, testimonia una trascurabile richiesta termica da parte dell’utenza per la gran parte dell’anno solare ed un picco di consumi

corrispondente registrato durante i mesi più freddi. A conferma di ciò è possibile osservare l'andamento del surplus di potenza elettrica conferito alla rete (**Figura 102**) e di quella invece integrata nei momenti di maggiore richiesta (**Figura 103**), i cui valori più elevati si riferiscono rispettivamente al periodo estivo per la prima ed a quello invernale per la seconda. Riguardo la sottosezione dedicata alla produzione di idrogeno è utile analizzare i consumi elettrici dell'elettrolizzatore (**Figura 104**) e del sistema di compressione (**Figura 106**), nonché la portata di idrogeno prodotto (**Figura 105**), che si registra più intensa in corrispondenza dei periodi di sovrapproduzione elettrica, con un conseguente aumento dei valori di pressione (**Figura 107**), e dunque di livello (**Figura 108**), all'interno del serbatoio di idrogeno durante il medesimo periodo, coerentemente con quanto visto in precedenza circa le logiche di gestione dei flussi previste dalla macchina a stati. Ne segue, come previsto, una maggiore autonomia elettrica del sistema rispetto alla rete durante i mesi estivi, durante i quali, nonostante gli alti carichi, l'intervento della cella a combustibile consente una maggiore copertura del fabbisogno giornaliero. I grafici relativi alla potenza elettrica erogata dalla cella, il suo consumo di idrogeno e l'efficienza di conversione sono mostrati in **Figura 109**, **Figura 110** e **Figura 111**.

Anche per quanto riguarda il sistema di accumulo elettrochimico è possibile confermare quanto detto sinora, poiché il periodo di maggiore attività è in linea con gli andamenti del sistema di produzione, accumulo e riconversione dell'idrogeno. È infatti possibile osservare dalle **Figura 112**, **Figura 113**, **Figura 114**, **Figura 115**, rispettivamente relativi a tensione, stato di carica e potenza elettrica erogata/ricevuta dalla batteria, come il maggior numero di cicli di carica e scarica completa siano concentrati nei mesi più caldi dell'anno, a causa della maggiore necessità di potenza elettrica da parte dell'utenza.

Le simulazioni svolte in questa fase, dimostrano l'efficacia di Stateflow come software per l'implementazione della logica di gestione e controllo. Si precisa che le simulazioni della fase due sono state svolte disattivando la modalità real time, essendo riferite ad un intero anno di durata.

Il banco prova realizzato si conferma pertanto uno strumento utile e appropriato per la programmazione di centraline di controllo per la gestione e l'ottimizzazione di reti complesse.

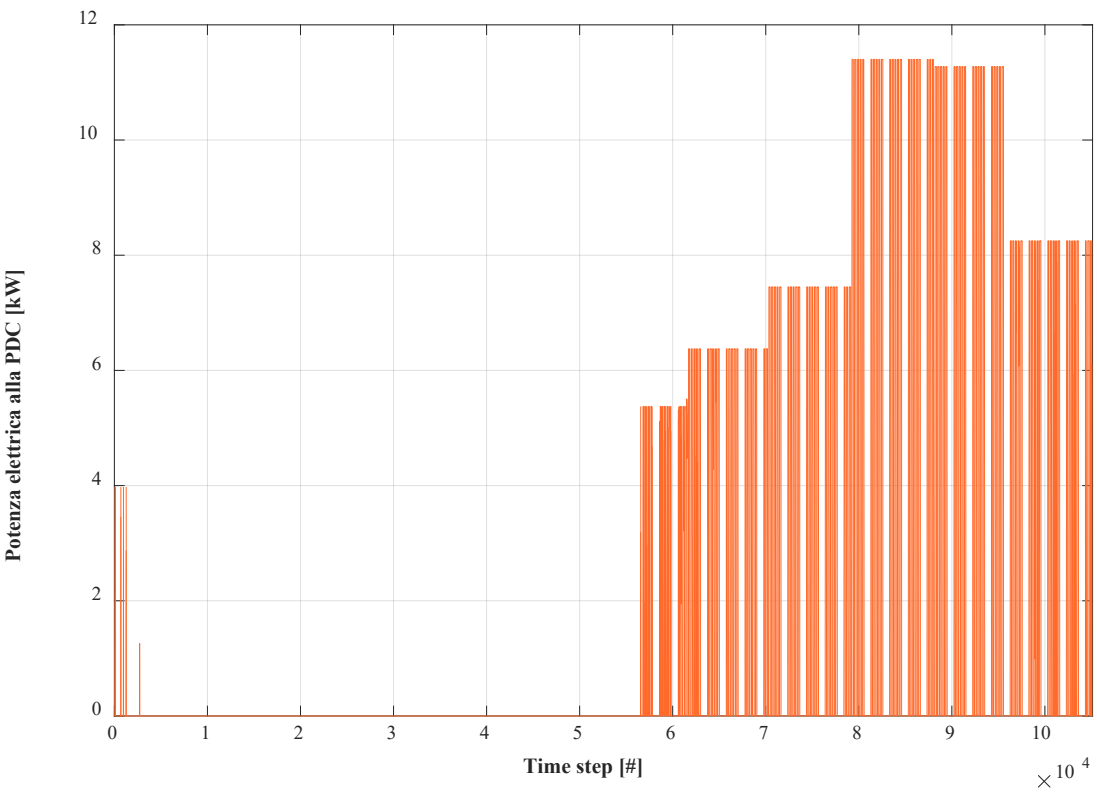


Figura 100 - potenza elettrica richiesta dalla pompa di calore (time step: 5 minuti)

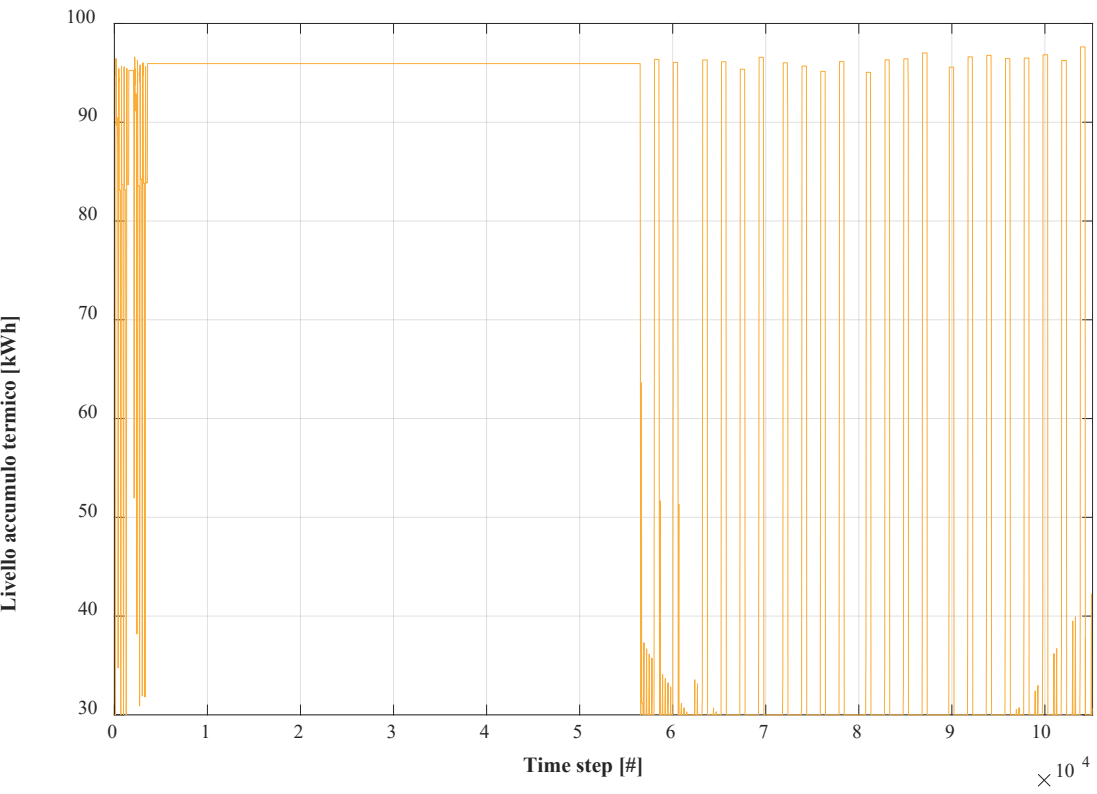


Figura 101 - livello di acqua calda nell'accumulo termico (time step: 5 minuti)

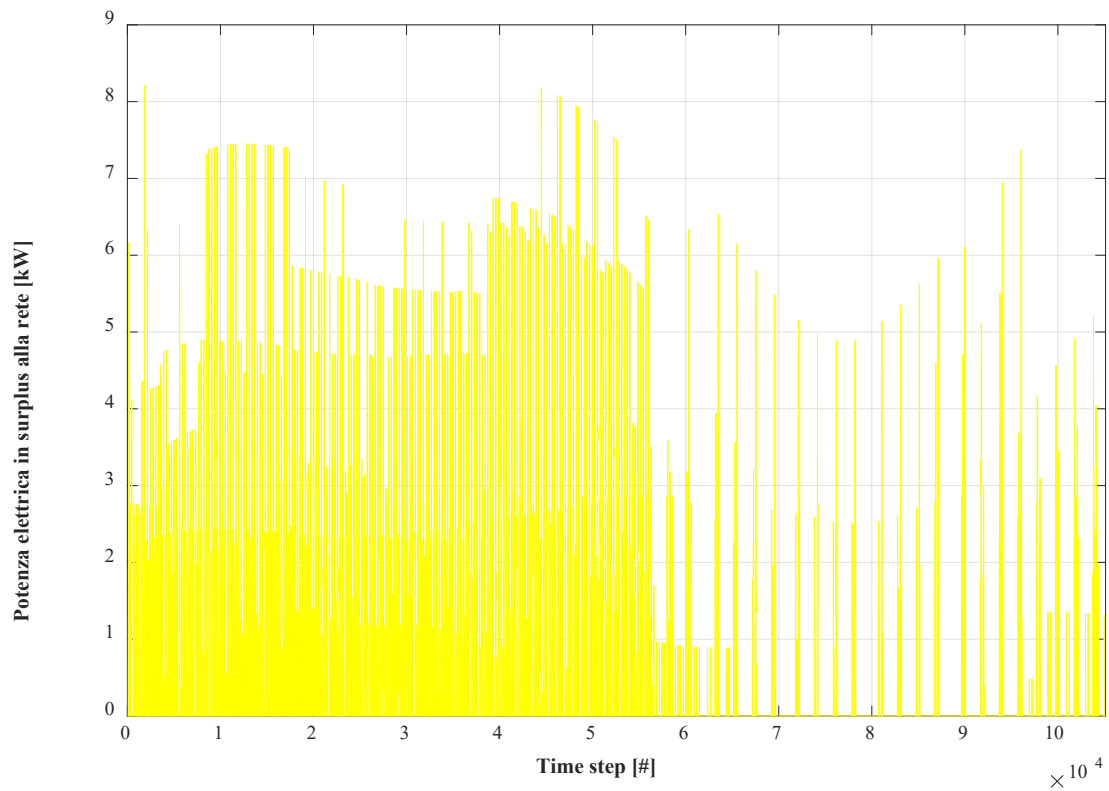


Figura 102 - potenza elettrica in surplus alla rete (time step: 5 minuti)

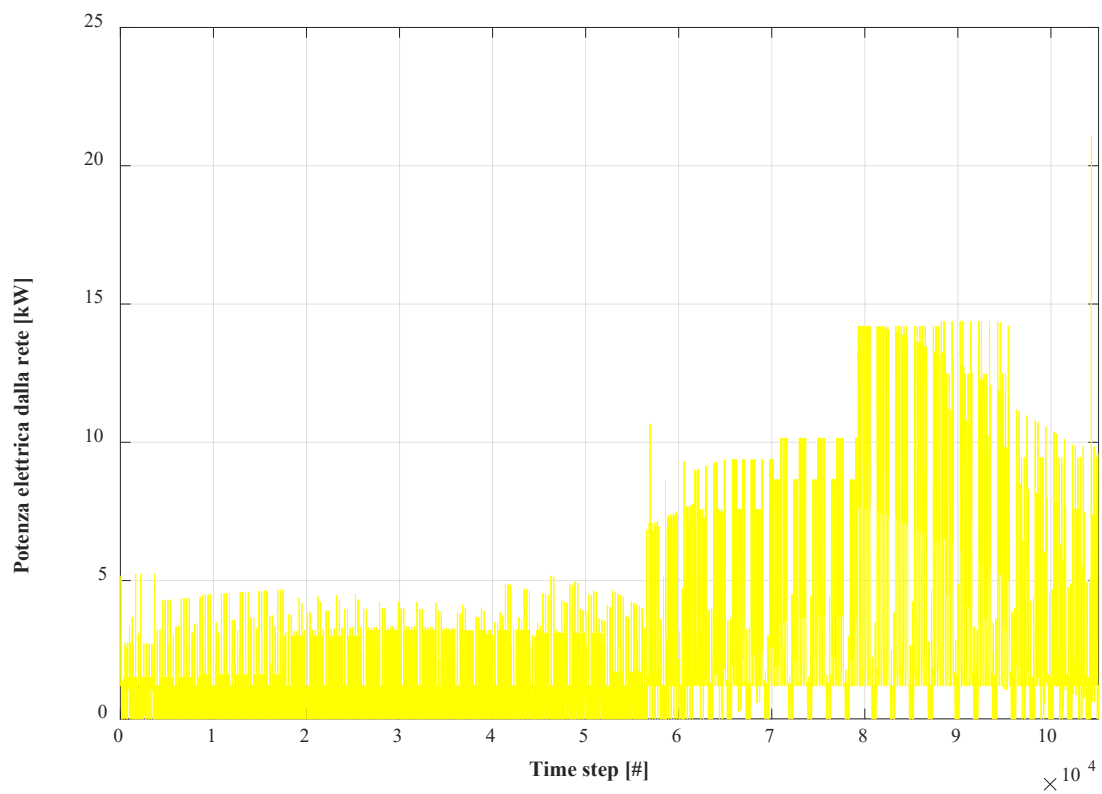


Figura 103 - potenza elettrica proveniente dalla rete (time step: 5 minuti)

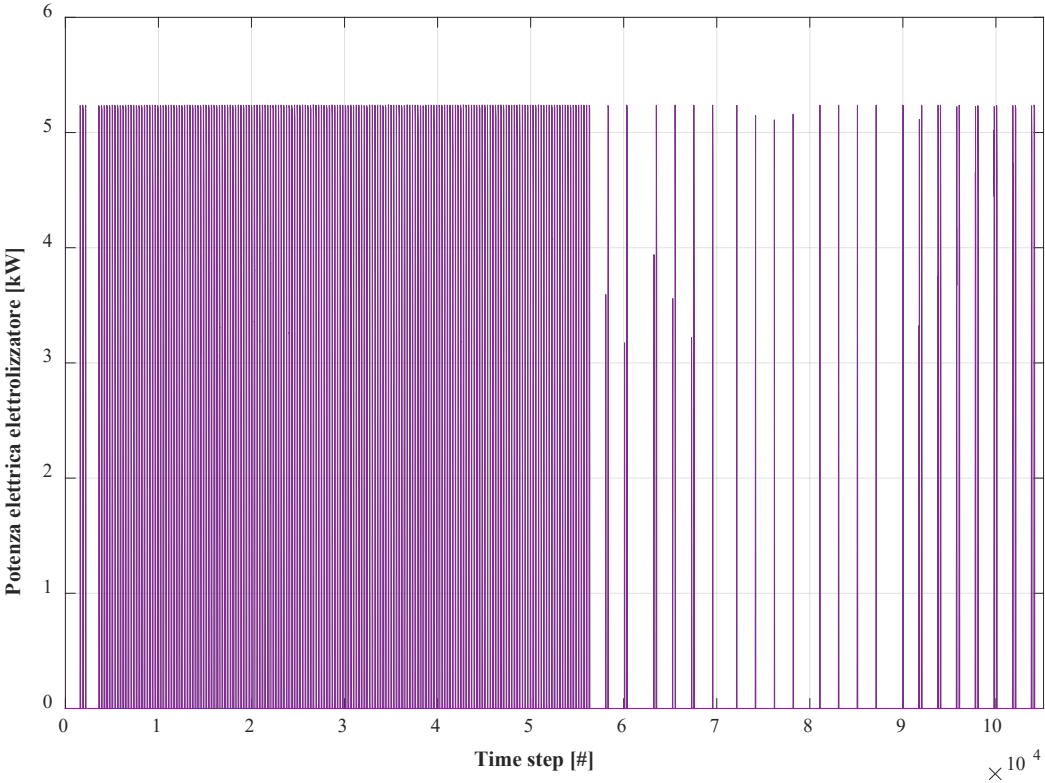


Figura 104 - potenza elettrica in ingresso all'elettrolizzatore (time step: 5 minuti)

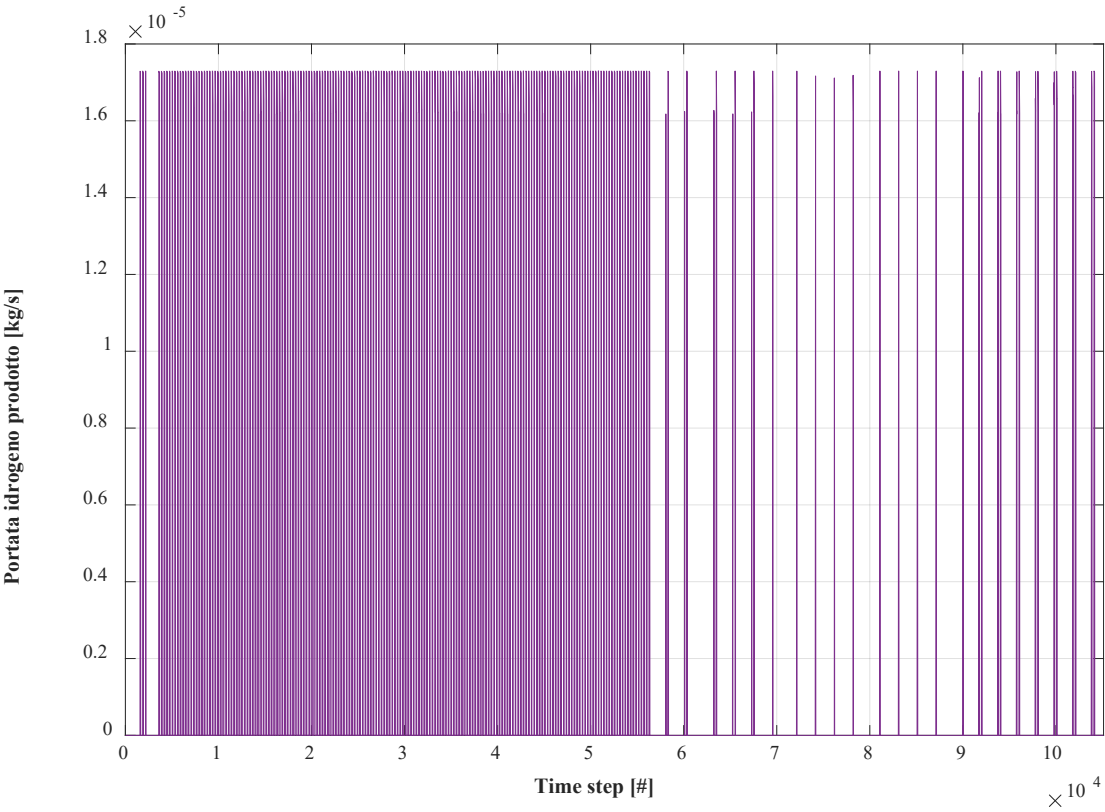


Figura 105 - portata di idrogeno prodotto (time step: 5 minuti)

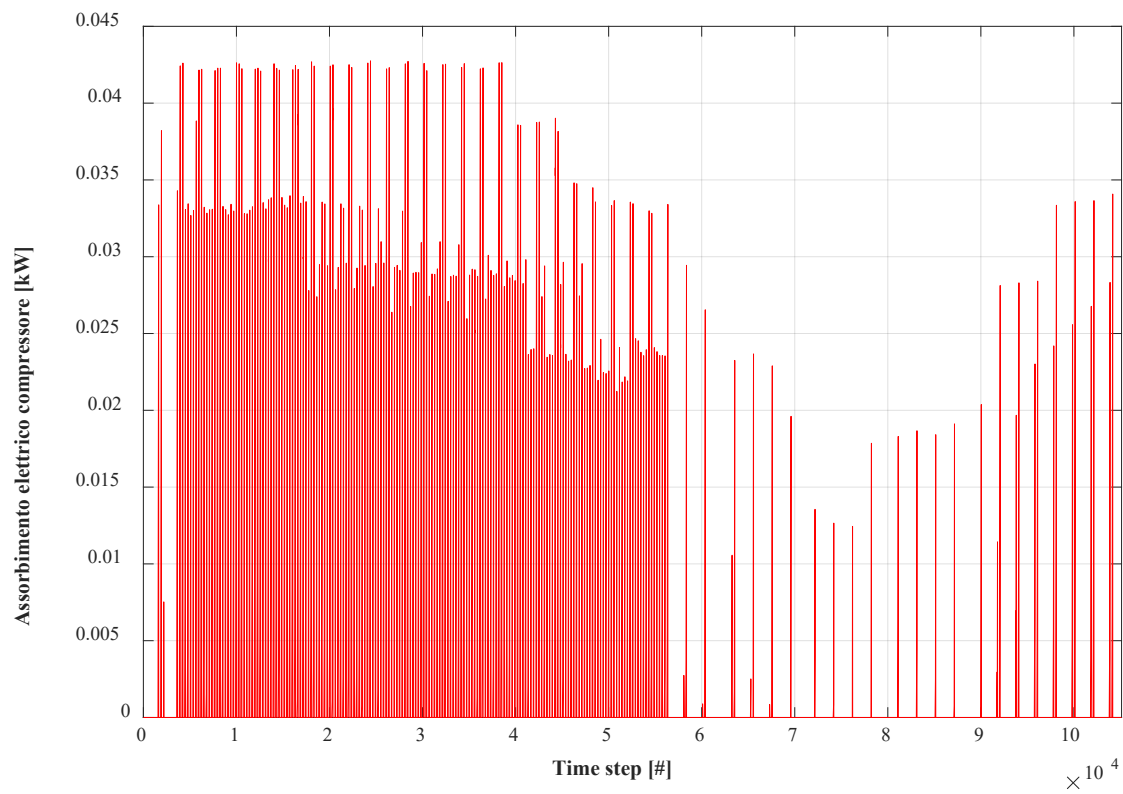


Figura 106 - assorbimento elettrico del compressore (time step: 5 minuti)

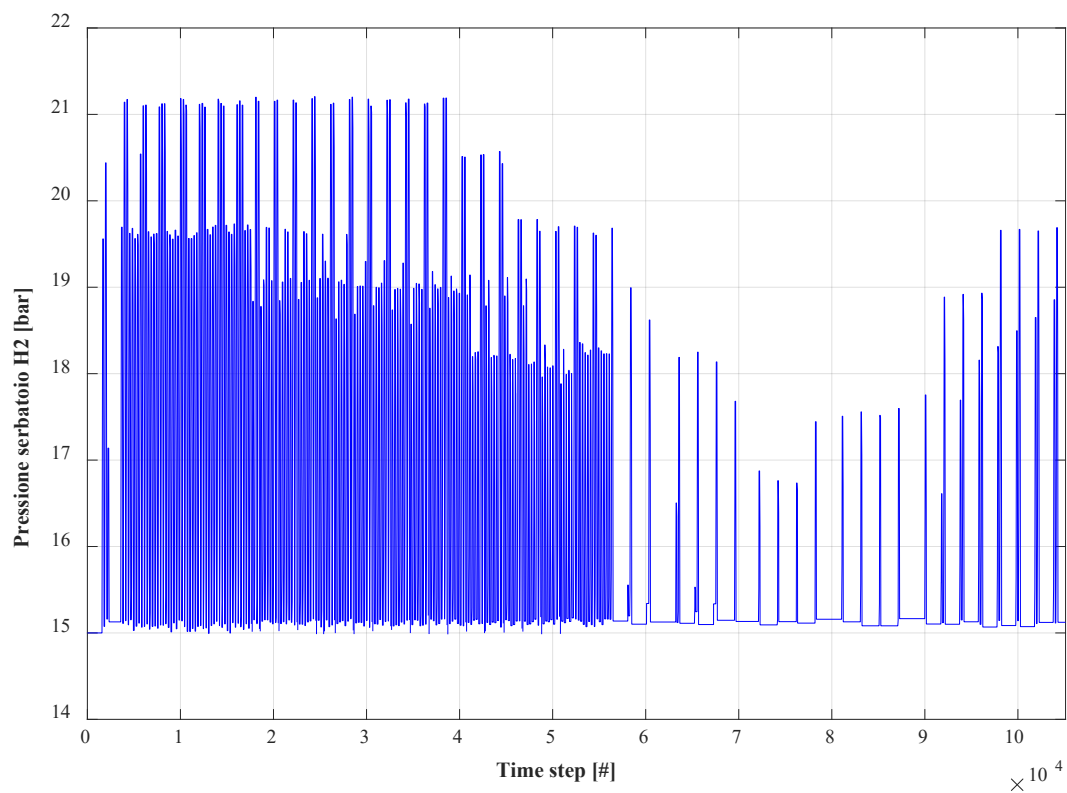


Figura 107 - pressione interna al serbatoio di H₂ (time step: 5 minuti)

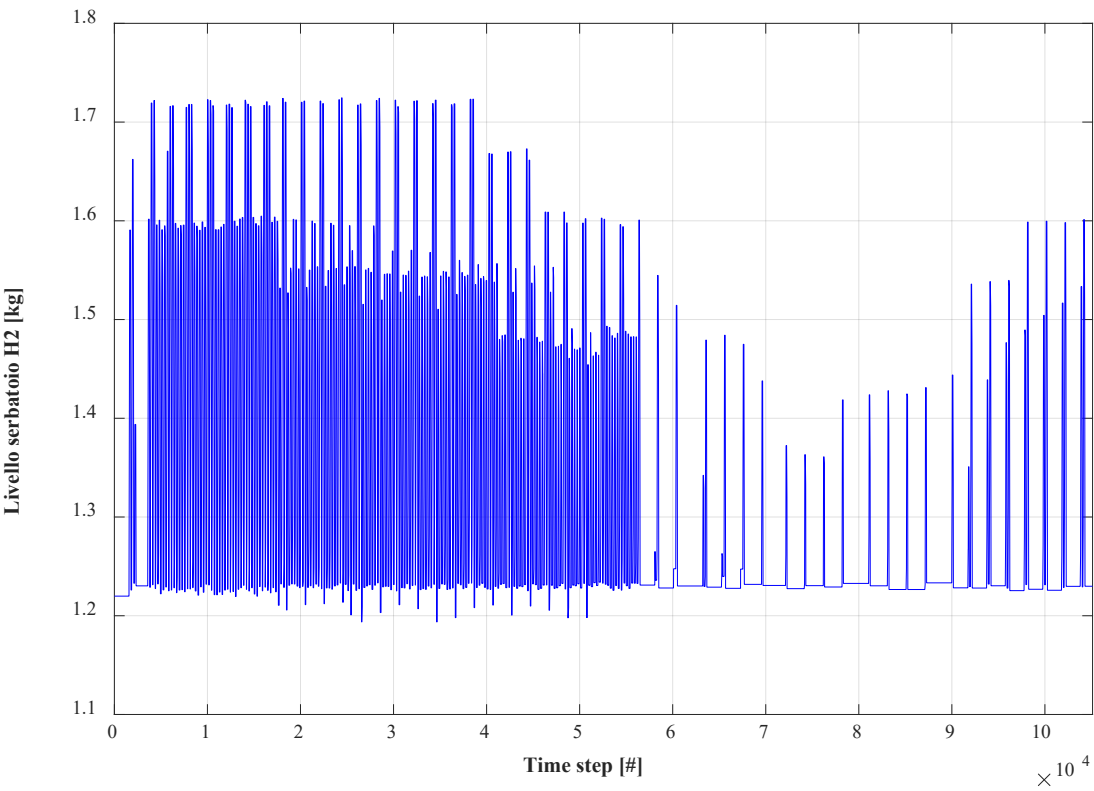


Figura 108 - livello del serbatoio di H₂ (time step: 5 minuti)

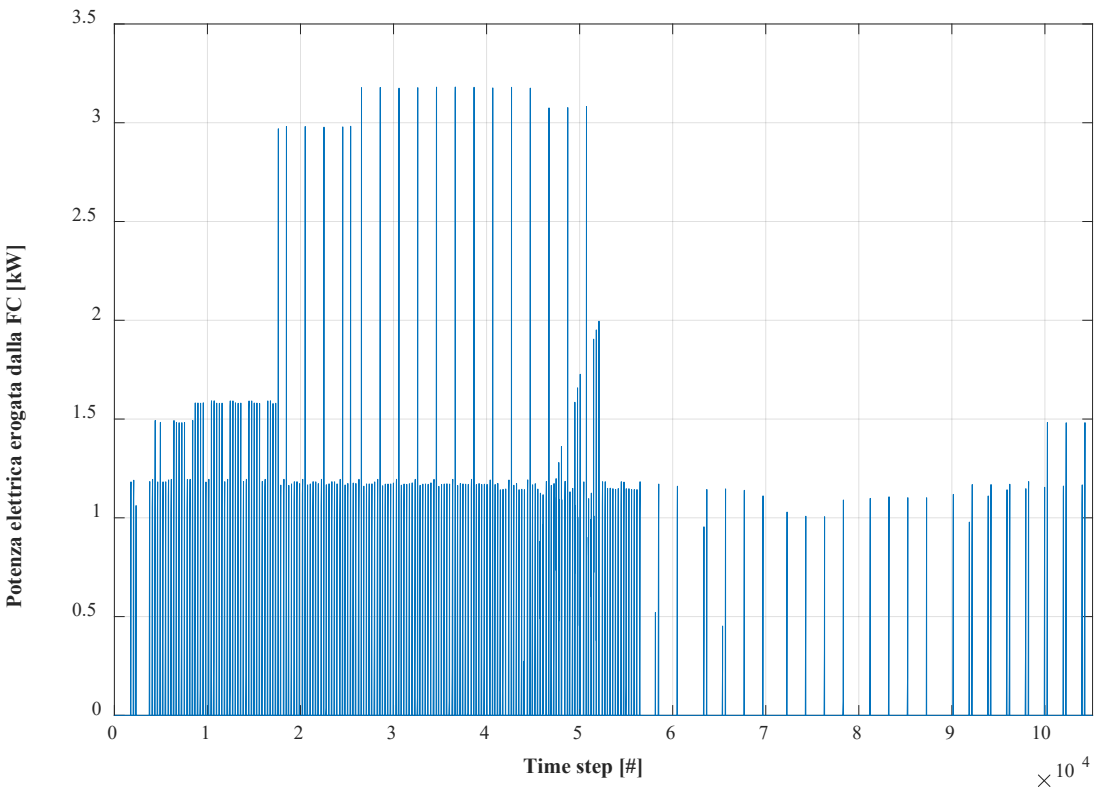


Figura 109 - produzione elettrica della FC (time step: 5 minuti)

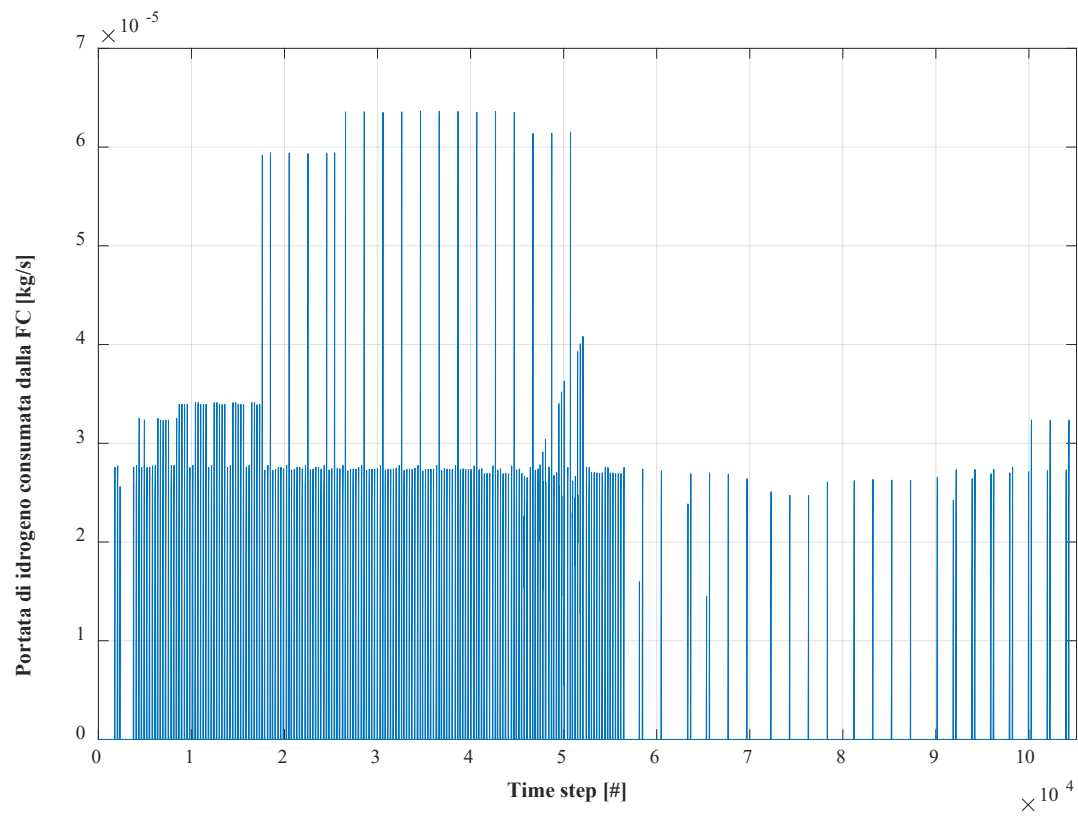


Figura 110 - consumo di idrogeno della FC (time step: 5 minuti)

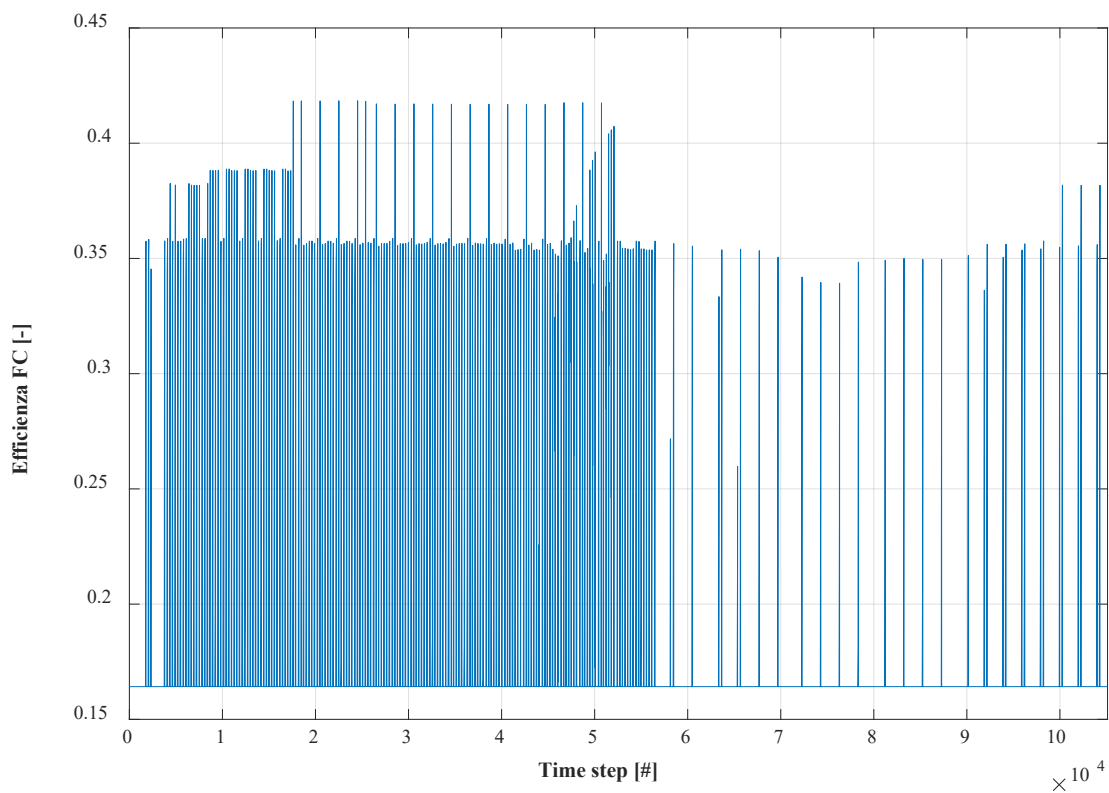


Figura 111 - efficienza della FC (time step: 5 minuti)

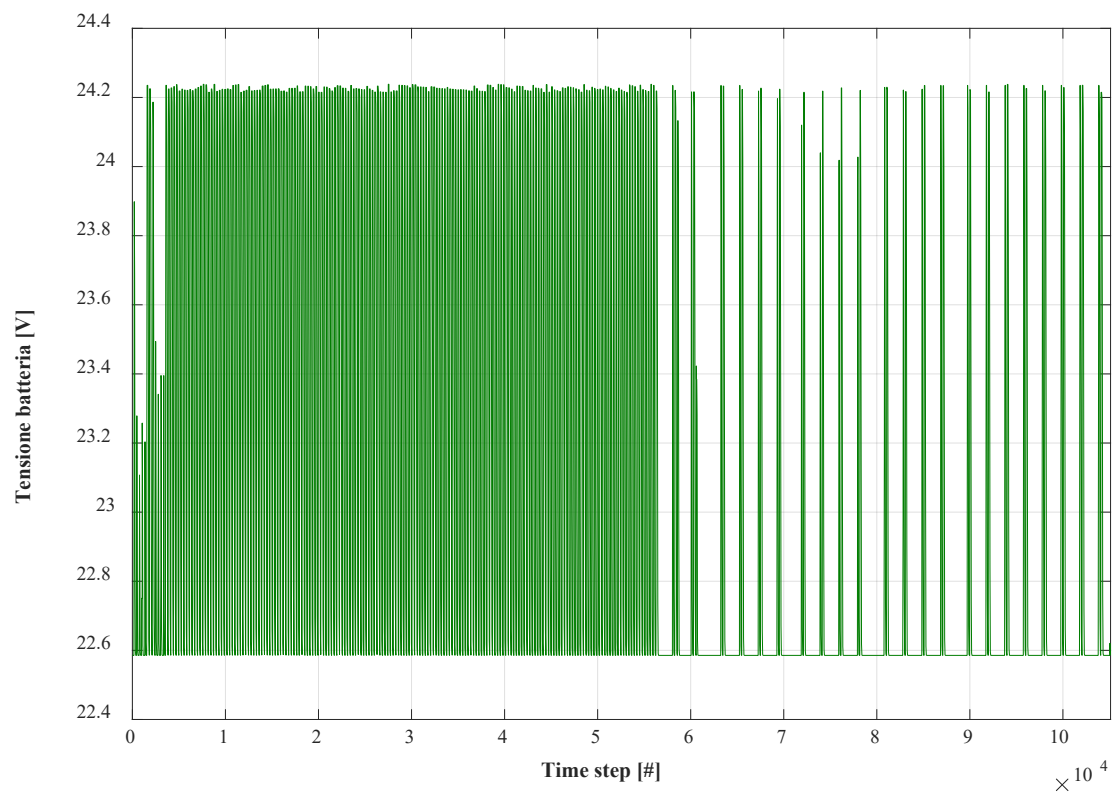


Figura 112 - tensione della batteria (time step: 5 minuti)

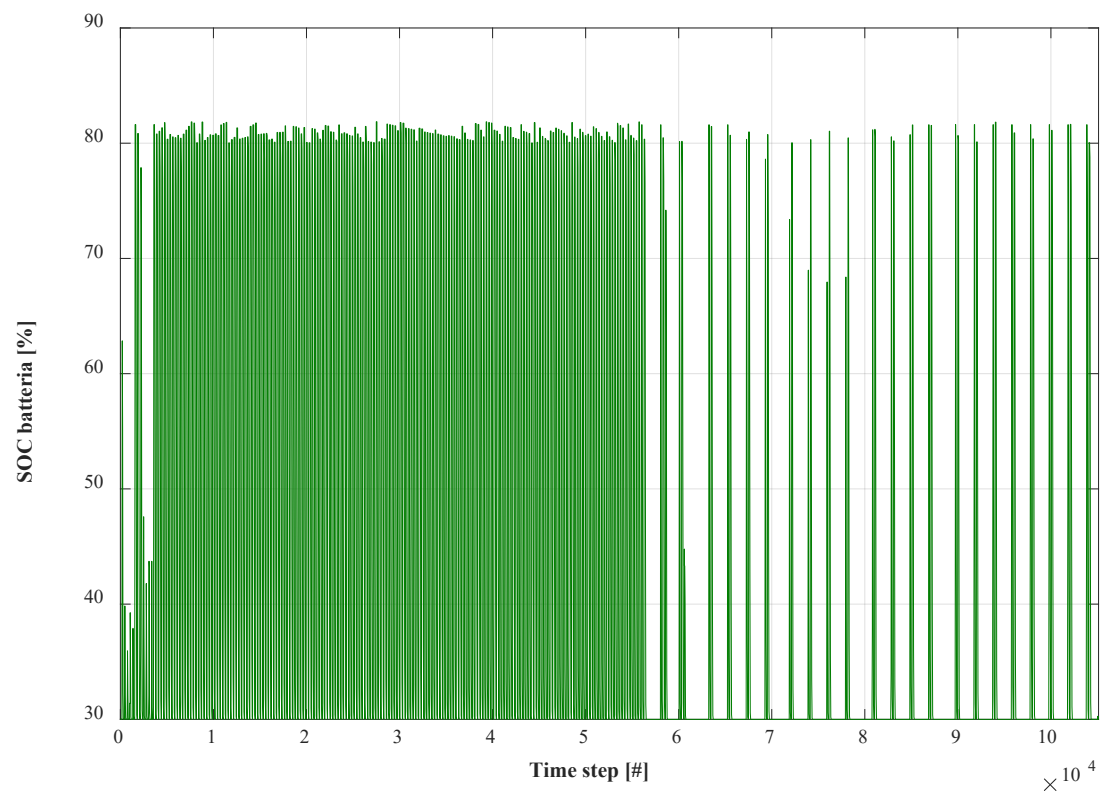


Figura 113 - SOC (state of charge) della batteria (time step: 5 minuti)

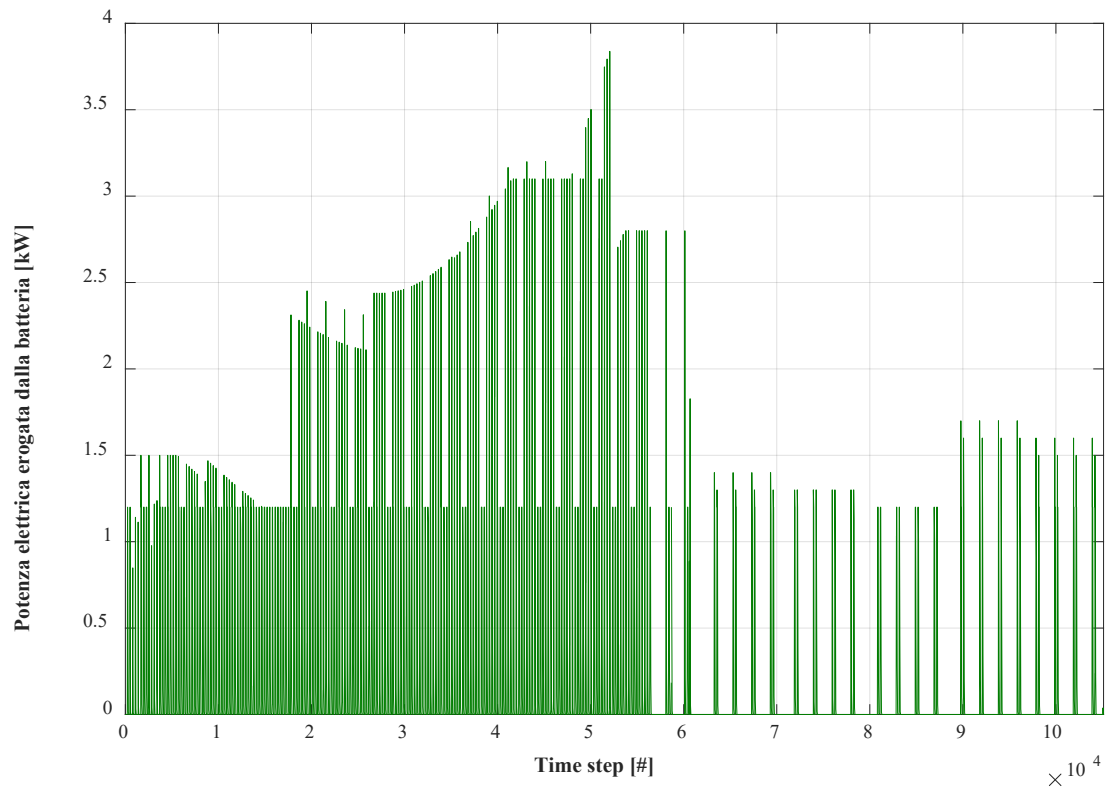


Figura 114 - potenza elettrica erogata dalla batteria (time step: 5 minuti)

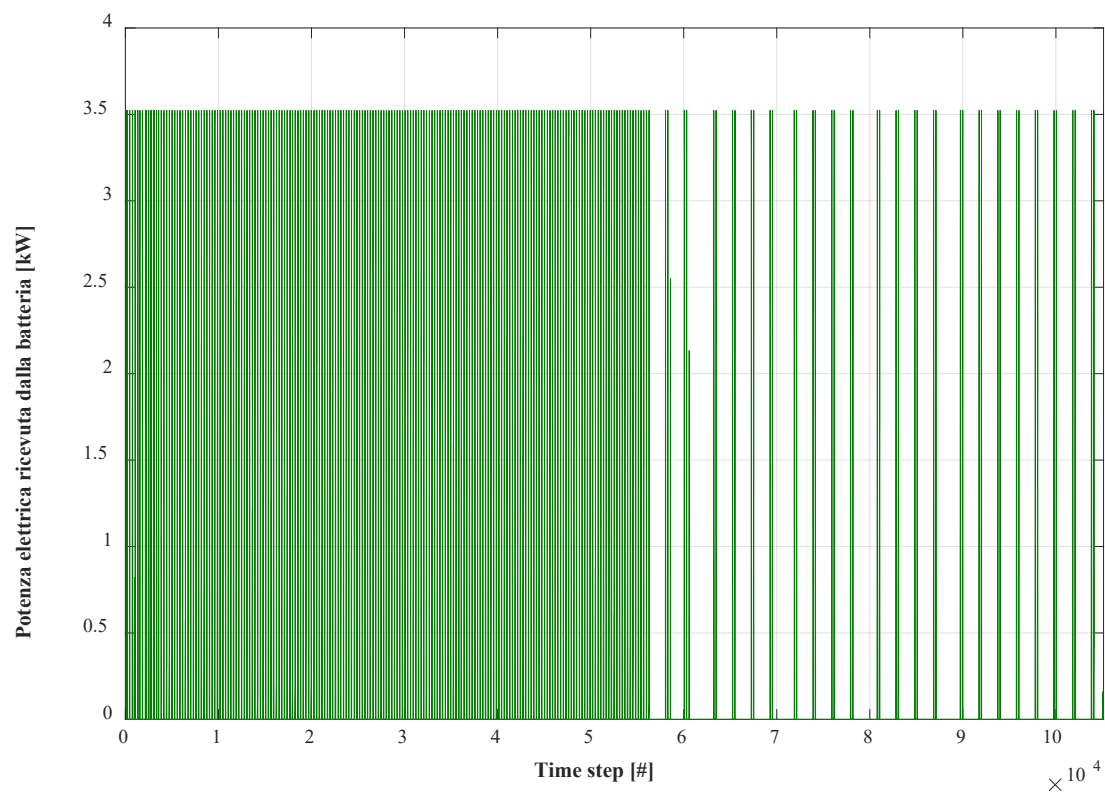


Figura 115 - potenza elettrica ricevuta dalla batteria (time step: 5 minuti)

4 Conclusioni

Il presente documento è il risultato della collaborazione tra l'ENEA e il Dipartimento di Ingegneria Industriale (DIN) dell'Alma Mater Studiorum – Università di Bologna, finalizzata allo studio di sistemi ibridi di accumulo per l'incremento dello sfruttamento della risorsa rinnovabile non programmabile nell'ambito delle comunità energetiche. Tale collaborazione nasce all'interno dell'Accordo di Programma che il Ministero dello Sviluppo Economico ed ENEA hanno stipulato nel 2019 finalizzato all'esecuzione delle linee di attività del Piano Triennale della Ricerca e Sviluppo di Interesse Generale per il Sistema Elettrico Nazionale.

L'accordo di collaborazione tra l'ENEA e il Dipartimento di Ingegneria Industriale (DIN) dell'Alma Mater Studiorum – Università di Bologna si riferisce al Piano Triennale di Realizzazione 2019-2021; Il presente documento descrive la ricerca svolta durante la seconda annualità.

L'attività di ricerca in oggetto è finalizzata allo studio della produzione e successivo impiego di idrogeno come strategia di accumulo per incrementare lo sfruttamento delle risorse rinnovabili non programmabili nell'ambito delle comunità energetiche. L'obiettivo della ricerca è l'ottenimento dell'indipendenza energetica nel settore civile/residenziale con particolare riferimento ai piccoli distretti cittadini.

La linea di attività del secondo anno è dedicata allo sviluppo di un laboratorio di ricerca finalizzato all'individuazione di strategie ottimizzate per la gestione real time di reti ibride complesse per la generazione, accumulo e conversione da fonti rinnovabili non programmabili, nell'ambito delle comunità energetiche.

Più in dettaglio l'attività svolta ha consentito di realizzare un'infrastruttura dedicata alla sperimentazione ed allo studio delle logiche di funzionamento e gestione delle reti energetiche. Il laboratorio in oggetto è stato realizzato secondo le logiche dell'hardware in the loop al fine di sviluppare sistemi di emulazione e controllo (centraline) real time per la gestione ottimizzata di reti complesse.

Di seguito si riporta una sintesi dei temi affrontati dalla ricerca svolta e dei principali risultati conseguiti.

Definizione dell'architettura della rete di idrogeno

Si è proceduto – preliminarmente – a definire lo stato dell'arte delle piattaforme elettroniche programmabili open source con particolare attenzione ai dispositivi di tipo Arduino e Raspberry. La ricerca condotta ha permesso di individuare una soluzione basata su un approccio di tipo MIL ("Model in The Loop") su cui testare il software di controllo delle reti progettato in ambiente Simulink. L'architettura del sistema di emulazione progettata e installata presso il laboratorio di meccanica del Dipartimento di Ingegneria Industriale dell'Università di Bologna è schematizzata in **Figura 26**. Si tratta di una serie di schede Arduino mega 2560 atte a riprodurre, unicamente a mezzo di un modello di calcolo sviluppato in codice dall'IDE di Arduino, il comportamento di ciascun componente fisico della rete. In questo modo è possibile sollecitare l'unità di controllo con step temporali molto ridotti, emulando di fatto le condizioni operative del sistema senza la necessità di operare con i reali sistemi fisici, bensì rimanendo in ambiente virtuale.

Installazione del banco prova di emulazione

L'infrastruttura di emulazione realizzata è stata pertanto integrata con i dispositivi fisici (pannelli fotovoltaici, sistemi di accumulo elettrochimico, elettrolizzatore, stoccaggio idrogeno in idruri metallici) presenti nel laboratorio di meccanica del DIN, così come mostrato in **Figura 27**.

Più in dettaglio, i dispositivi fisici inclusi nel banco prova sono:

- due generatori fotovoltaici in silicio policristallino, connessi in parallelo (440 W di potenza elettrica totale in condizioni STC);

- due dispositivi di accumulo elettrochimico (tecnologia AGM) connessi in serie (capacità elettrica totale di 110 Ah ad una tensione nominale di 24 V);
- un regolatore della carica di tipo PWM, per la gestione dell'interconnessione tra pannelli fotovoltaici e batterie;
- un convertitore DC/DC di tipo Meanwell SD25B-12;
- un inverter DC/AC di tipo Meanwell TS-700-224B;
- un elettrolizzatore di tipo PEM (capacità 30L/ora di idrogeno a 10.5 bar di pressione);
- tre bombole ad idruri metallici per lo stoccaggio dell'idrogeno prodotto (200 SL ciascuna a 10 bar).

In aggiunta, si è proceduto all'installazione di una linea ad idrogeno, alimentata attraverso due bombole da 55 L l'una e 200 bar di pressione massima, e di una fuel cell commerciale di tecnologia PEM (non connesse alla microrete). La fuel cell, progettata per operare a basse temperature (attorno ai 65 °C a pieno carico), è in grado di produrre una potenza elettrica nominale pari a 2 kW ad una tensione di 28.8 V ed una corrente elettrica massima di 75 A, a fronte di un consumo massimo di idrogeno dichiarato di 26 L/min.

Programmazione del banco prova di emulazione

Il banco prova installato è stato settato per riprodurre il funzionamento del sistema SAPIENTE installato presso l'ENEA – sede di Casaccia con l'integrazione di un sistema di produzione, stoccaggio e conversione di idrogeno. A tal fine si è prima proceduto alla programmazione – mediante Simulink – dei modelli di ogni singolo componente della rete in oggetto.

I modelli di calcolo, validati sperimentalmente secondo quanto riportato nel report relativo al primo anno di attività, sono stati implementati in modo da interfacciarsi con i dispositivi elettronici Arduino tenendo conto delle prestazioni di tali sistemi in termini di memoria disponibile e potenza di calcolo.

Progettazione della logica di controllo e gestione

Per la realizzazione della strategia di controllo e gestione si è impiegato il software Stateflow che – come noto – è un linguaggio di programmazione grafico che include diagrammi e tabelle di transizione di stato, diagrammi di flusso e tabelle di verità tramite cui è possibile scegliere in che modo gli algoritmi di MATLAB e i modelli di Simulink debbano reagire a segnali di input, eventi e condizioni nel tempo. In questo modo è stato possibile configurare la logica decisionale, combinatoria e sequenziale di tutti i sotto-sistemi e della rete nel suo complesso. In **Figura 82** è illustrata la flow chart (macchina a stati) del sistema di controllo e gestione nel suo complesso.

Infine, servendosi del pacchetto ICT è stato realizzato un protocollo di comunicazione per lo scambio dati in tempo reale tra le schede Arduino e Simulink, attraverso la programmazione separata dei microcontrollori mediante l'IDE di Arduino. La modalità di comunicazione adottata per lo scambio dati tra schede e PC è stata incentrata su una trasmissione di tipo seriale, fisicamente supportata da una connessione standard USB (*Universal Serial Bus*). In questo modo, servendosi di un hub USB, è stato possibile connettere tutti i dispositivi Arduino al computer e permetterne il flusso di dati bidirezionale.

Verifica della comunicazione con i dispositivi di emulazione

L'impiego dei dispositivi Arduino per lo scambio dati in tempo reale con il modello Simulink ha dimostrato di essere di facile impiego anche se caratterizzato da una stabilità del segnale inversamente proporzionale al numero di schede dell'apparato. Tale fenomeno è stato notevolmente mitigato variando i parametri di campionamento delle schede stesse ed aggiungendo al codice di ciascuna scheda dei controlli circa i valori ricevuti dal calcolatore all'inizio di ogni iterazione. In questo modo è stato possibile ottenere un significativo miglioramento dei segnali trasmessi in termini di stabilità e nitidezza come mostrato in **Errore. L'origine riferimento non è stata trovata..**

Verifica del banco prova e emulazione del sistema SAPIENTE

La verifica del banco prova di emulazione realizzato è stata fatta riproducendo il modello S.A.P.I.E.N.T.E. integrato con una sezione di produzione, stoccaggio e riconversione di idrogeno nell'ipotesi di collocazione dello stesso in un contesto di tipo civile/residenziale. In questo modo è stato possibile stimare l'effetto dell'unità di produzione e stoccaggio dell'idrogeno come strategia per l'accumulo a lungo termine dell'energia elettrica prodotta in eccesso dal generatore rinnovabile non programmabile nonché approfondire la tematica delle simulazioni/emulazioni in *real time*, mediante le piattaforme elettroniche descritte nei precedenti paragrafi, per delineare con maggior precisione il comportamento dinamico del sistema e porre le basi per le successive fasi di sviluppo della centralina elettronica secondo un approccio di tipo HIL (*Hardware In the Loop*).

I risultati conseguiti confermano la funzionalità del banco prova realizzato come strumento di emulazione e programmazione di centraline di controllo mediante approccio hardware-in-the-loop.

5 Riferimenti bibliografici

- [1] Joshua M. Pearce, "Building Research Equipment with Free, Open-Source Hardware", Science Magazine, vol. 337, 14 sept 2012.
- [2] Arduino, <https://www.arduino.cc/>
- [3] Raspberry, <https://www.raspberrypi.org/>
- [4] I. Allafi, T. Iqbal, "Low-Cost SCADA System Using Arduino and Reliance SCADA for a Stand-Alone Photovoltaic System", Journal of Solar Energy, 2018.
- [5] I. Gonzalez, A. J. Calderon, "Integration of open-source hardware Arduino platform in automation systems applied to Smart Grids/Micro-Grids", Sustainable Energy Technologies and Assessments 36 (2019) 100557.
- [6] A. S. Ali, Z. Zanzinger, D. Debose, B. Stephens, "Open Source Building Science Sensors (OSBSS): A low-cost Arduino-based platform for long-term indoor environmental data collection", Building and Environment 100 (2016) 114-126.
- [7] D. F. Merchan, J. A. Peralta, A. Vasquez-Rodas, L. I. Minchala, D. Astudillo-Salinas, "Open source SCADA system for advanced monitoring of industrial processes", 2017 International Conference on Information Systems and Computer Science.
- [8] I. G. Perez, A. J. Calderon Godoy, M. Calderon Godoy, "Integration of Open Source Arduino with LabVIEW-based SCADA through OPC for Application in Industry 4.0 and Smart Grid Scenarios", Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019), pages 174-180.
- [9] L. O. Aghenta, M. T. Iqbal, "Development of an IoT Based Open Source SCADA System for PV System Monitoring", 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE).
- [10] L. Sarinda Jayasinghe, T. Iqbal, G. Mann, "Low-Cost and Open Source SCADA Options for Remote Control and Monitoring of Inverters", 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE).
- [11] C. Vargas-Salgado, J. Auilla-Leon, C. Chinas-Palacios, E. Hurtado-Perez, "Low-cost web-based Supervisory Control and Data Acquisition system for a microgrid testbed: A case study in design and implementation for academic and research applications", Heliyon 5 (2019).
- [12] Y. Li, F. Nejbatkhah, "Overview of control, integration and energy management of microgrids", J. Mod. Power Syst. Clean Energy (2014) 2(3):212–222.
- [13] A. Chaouachi, R. M. Kamel, R. Andoulsi, K. Nagasaka, "Multiobjective Intelligent Energy Management for a Microgrid", IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 60, NO. 4, APRIL 2013.
- [14] A. Kuzin, D. Lukichev, G. Demidova, A. Anuchin, "Concept of Hardware-in-the-loop Test Platform for Microgrid with Multi-Agent Approach", 2020 IEEE 61st International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCon).
- [15] M. Schvarcbacher, B. Rossi, "Smart Grids Co-Simulations with Low-Cost Hardware", 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications.
- [16] Y. Huo, G. Gruosso, L. Piegari, "Power Hardware In The Loop Simulator Of Photovoltaic Plant For Smart Grid Interaction Analysis", IEEE (2017).
- [17] Y. Gangat, D. Grondin, T. Issoufaly, N. Coquillas, M. Benne, J. P. Chabriat, "Simulation & Emulation platform for smart grid technologies", The 2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC).
- [18] Raspbian community, <https://www.raspbian.org/>
- [19] Mathworks support for Arduino, <https://it.mathworks.com/help/supportpkg/arduino/ug/pin-mapping-on-arduino-blocks.html>
- [20] National Instruments, <https://www.ni.com/it-it.html>

- [21] Margolis. M, *Arduino Cookbook*, O'Reilly Media Inc., Sebastopol, CA 95472, 2012.
- [22] G. Bellazzi, *Alla scoperta del Raspberry Pi, IL piccolo PC Linux da 35\$*, Prima Edizione, 2013.
- [23] Impianto S.A.P.I.E.N.T.E. Centro Ricerche ENEA Casaccia,
https://www.opendaydellaricerca.enea.it/images/2019sktourcasaccia/01_SCHEDAsapienteOPENDAY_2019.pdf
- [24] B. Di Pietra, A. Pannicelli, S. Piccinelli, A. Ciappa, F. Monteleone, C. Bommarito, G. Di Sarra, G. Pace, "Progettazione di soluzioni impiantistiche ibride per gli edifici a potenza zero", 2017.
- [25] Ancona, M. A., Bianchi, M., Branchini, L., De Pascale, A., Melino, F., Peretto, A., ... & Scarponi, L. B. (2017). From solar to hydrogen: Preliminary experimental investigation on a small scale facility. *International Journal of Hydrogen Energy*, 42(33), 20979-20993.
- [26] Wikipedia, https://it.wikipedia.org/wiki/Pagina_principale
- [27] Heckert solar, <https://www.heckertsolar.com/>
- [28] Banner Batterien, <https://www.bannerbatterien.com/it/Home>
- [29] Steca, <https://www.steca.com/index.php?Home-en>
- [30] Meanwell, <https://www.meanwell.com/>
- [31] Heliocentris, <http://heliocentrisacademia.com/>
- [32] Horizon Fuel Cell Technologies, <https://www.horizonfuelcell.com/>
- [33] Mathworks, <https://it.mathworks.com/>

6 Nomenclatura

Simboli

A	Ampiezza della zona esponenziale della batteria [V]
B	Costante temporale inversa [Ah ⁻¹]
C	Capacità [Ah]
<i>e</i>	Errore [%]
<i>E</i>	Potenziale [V]
<i>F</i>	Costante di Faraday [C/mol]
<i>G</i>	Radiazione solare [W/m ²]
G	Energia libera di Gibbs specifica [kJ/kmol]
H	Entalpia [kJ]
<i>i</i>	Corrente elettrica [A]
<i>i'</i>	Corrente filtrata [A]
<i>J</i>	Densità di corrente elettrica [A/cm ²]
K	Costante di polarizzazione [V/Ah]
LHV	Lower Heating Value [kJ/kg]
<i>m</i>	Massa [kg]
<i>n</i>	Portata molare [mol/s]
<i>p</i>	Pressione [bar]
P	Potenza [W]
<i>q</i>	Carica di un elettrone [C]
Q	Capacità batteria [Ah]
R	Resistenza [Ω]
<i>s</i>	Entropia specifica [J/molK]
S	Entropia [kJ/K]
T	Temperatura [°C]
V	Tensione [V]
Z	Numero di valenza degli ioni [-]

Lettere greche

Δ	Differenza
η	Efficienza [-]

Pedici ed apici

a	Anodo
acc	Accumulo
act	Attivazione
batt	Batteria
c	Catodo
cell	Cella
ch	Carica
d	Diodo
el	Elettrico
exp	Esponenziale/Sperimentale
in	Input
max	Massimo
min	Minimo
nom	Nominale

OC	Open circuit
ohm	Ohmico
p	Parallelo
rev	Reversibile
RR	Radiazione solare
s	Serie
sc	Corto-circuito
tn	Termo-neutro

Acronimi

AC	Alternating current
AEM	Anion Exchange Membrane
AFC	Alcaline Fuel Cell
AM	Amplitude modulation
BASE	Beta Alumina Solid Electrolyte
DC	Direct current
DMFC	Direct Methanol Fuel Cell
MCFC	Molten Carbonate Fuel Cell
NOCT	Nominal Operating Cell Temperature
PAFC	Phosphoric Acid Fuel Cell
PEFC	Polymer Electrolyte Fuel Cell
PEM	Proton Exchange Membrane
PV	Fotovoltaico
PWM	Pulse width modulation
SOC	State of Charge
SOEC	Solid Oxide Electrolysis Cell
SOFC	Solid Oxide Fuel Cell
STC	Standard test conditions
YSZ	Yttria Stabilized Zirconia

7 Indice delle figure

Figura 1 - Arduino starter kit [2]	10
Figura 2 - Arduino robotic arm (sinistra) e Arduino controlled spoiler (destra) [2]	10
Figura 3- Scheda Arduino Mega 2560 [2]	14
Figura 4 - Alimentatore 9 V DC per Arduino [2]	15
Figura 5 - Chip Atmel ATmega2560	15
Figura 6 - Mappa dei LED Arduino Mega2560 R3 [2]	16
Figura 7 - pin map Arduino [19]	17
Figura 8 - Ethernet shield, Display TFT touch shield e Motion shield per differenti modelli Arduino [2]	18
Figura 9 - Arduino IDE	19
Figura 10 - Icone IDE	19
Figura 11 - Finestra di interfaccia del monitor seriale	20
Figura 12 - Struttura dello Sketch	21
Figura 13 - Matlab Support Package for Arduino Hardware e LIFA	21
Figura 14 - Simulink Support Package for Arduino Hardware	22
Figura 15 - Modello realizzato tramite l'Instrument Control Toolbox	23
Figura 16 - Esempio di scambio dati tramite l'ICT	23
Figura 17 - Logo LINX ed esempio di applicazione	24
Figura 18 - riproduzione schematizzata dei modelli: a) Raspberry Pi 1B+ e b) Raspberry Pi 4B [26]	28
Figura 19 - Raspberry Pi 4B [3]	29
Figura 20 - slot per l'inserimento della scheda SD [3]	30
Figura 21 - pin map Raspberry Pi [3]	30
Figura 22 - tensioni pin GPIO [3]	31
Figura 23 - Alimentatore dedicato per Raspberry Pi 4 [3]	31
Figura 24 - logo Raspbian [18]	32
Figura 25 - loghi di alcuni dei sistemi operativi non ufficiali per Raspberry Pi	32
Figura 26 - Architettura del sistema di emulazione	34
Figura 27 - banco prova di emulazione	34
Figura 28 - Schema del modello S.A.P.I.E.N.T.E. installato presso il Centro ENEA di Casaccia [24]	36
Figura 29 - Architettura aggiornata dell'impianto realizzato presso il Centro ENEA di Casaccia	36
Figura 30 - Layout della microrete di generazione ed accumulo di idrogeno da generazione rinnovabile non programmabile	37
Figura 31 - Componenti ed attrezzature del Laboratorio di Microreti di Generazione ed Accumulo dell'Alma Mater Studiorum - Università di Bologna	38
Figura 32 - Irraggiamento solare (test #1 - luglio 2020)	41
Figura 33 - Potenza prodotta dai pannelli fotovoltaici (test #1 - luglio 2020)	41
Figura 34 - Corrente prodotta dai pannelli fotovoltaici (test #1 - luglio 2020)	42
Figura 35 - Tensione dei pannelli fotovoltaici (test #1 - luglio 2020)	42
Figura 36 - Rendimento di conversione dei pannelli fotovoltaici (test #1 - luglio 2020)	43
Figura 37 - Corrente in ingresso o in uscita dalla batteria (test #1 - luglio 2020)	43
Figura 38 - Tensione della batteria (test #1 - luglio 2020)	44
Figura 39 - Idrogeno prodotto (test #1 - luglio 2020)	44
Figura 40 - Irraggiamento solare (test #2 - luglio 2020)	45
Figura 41 - Potenza prodotta dai pannelli fotovoltaici (test #2 - luglio 2020)	45
Figura 42 - Corrente prodotta dai pannelli fotovoltaici (test #2 - luglio 2020)	46
Figura 43 - Tensione dei pannelli fotovoltaici (test #2 - luglio 2020)	46
Figura 44 - Rendimento di conversione dei pannelli fotovoltaici (test #2 - luglio 2020)	47
Figura 45 - Corrente in ingresso o in uscita dalla batteria (test #2 - luglio 2020)	47
Figura 46 - Tensione della batteria (test #2 - luglio 2020)	48

Figura 47 – Idrogeno prodotto (test #2 – luglio 2020)	48
Figura 48 - specifiche tecniche della Fuel cell	49
Figura 49 – potenza elettrica erogata dalla cella al variare della corrente richiesta	50
Figura 50 - andamento della tensione della cella al variare della corrente richiesta	50
Figura 51 - consumo di idrogeno in funzione della potenza elettrica erogata	51
Figura 52 - curva di efficienza della cella in funzione del regime di carico	51
Figura 53 - Circuito equivalente di un pannello fotovoltaico	52
Figura 54 - modello di calcolo del pannello fotovoltaico	53
Figura 55 – vista complessiva del modello Simulink relativo al pannello fotovoltaico	54
Figura 56 – Curva caratteristica V-SOC della batteria durante il processo di scarica.....	56
Figura 57 – modello di calcolo della batteria	57
Figura 58 – vista complessiva del modello Simulink relativo alla batteria	57
Figura 59 - Efficienza dell’elettrolizzatore in funzione della potenza normalizzata.....	58
Figura 60 - modello di calcolo dell'elettrolizzatore	59
Figura 61 - vista complessiva del modello Simulink relativo all'elettrolizzatore.....	59
Figura 62 - Curva di efficienza della fuel cell al variare della potenza elettrica normalizzata	60
Figura 63 - modello di calcolo della fuel cell	60
Figura 64 - vista complessiva del modello Simulink relativo alla fuel cell.....	61
Figura 65 - Schema semplificato dell'inter-refrigerazione	62
Figura 66 - modello di calcolo del sistema di compressione	62
Figura 67 - vista complessiva del modello Simulink relativo al sistema di compressione dell'idrogeno prodotto	62
Figura 68 - densità dell'idrogeno al variare della pressione.....	63
Figura 69 - modello di calcolo del serbatoio di stoccaggio dell’idrogeno	64
Figura 70 - vista complessiva del modello Simulink relativo al serbatoio di stoccaggio dell'idrogeno prodotto	64
Figura 71 - modello di calcolo dell'accumulo termico.....	65
Figura 72 - modello di calcolo del sistema pompa di calore - accumulo termico	65
Figura 73 - vista complessiva del modello Simulink relativo al sistema pompa di calore - accumulo termico.....	66
Figura 74 - layout del sistema modellizzato ed individuazione delle sue sottosezioni	68
Figura 75 - sezione di produzione, compressione e stoccaggio dell'idrogeno	68
Figura 76 - blocco Stateflow inserito nel modello Simulink	69
Figura 77 - diagramma di flusso della macchina a stati progettata mediante Stateflow.....	69
Figura 78 - attivazione degli stati correnti nel diagramma Stateflow durante l'esecuzione del codice	70
Figura 79 - pannello di controllo delle variabili di Stateflow.....	70
Figura 80 - a) sottosistema di produzione ed accumulo termico, b) macchina a stati della sezione d’impianto	71
Figura 81 - sottosistema di accumulo elettrochimico	72
Figura 82 - macchina a stati del modello complessivo	72
Figura 83 - flowchart della strategia di funzionamento del sistema complessivo	73
Figura 84 - modalità di programmazione e controllo SPFAH [33]	74
Figura 85 - dispositivi elettronici compatibili con l'ICT [33]	75
Figura 86 - Architettura del sistema elettronico realizzato.....	76
Figura 87 - funzionalità della libreria ICT.....	76
Figura 88 - Serial configuration	77
Figura 89 - blocchi per l'invio di dati da Simulink ad Arduino	78
Figura 90 - blocchi per l’invio di dati da Arduino a Simulink	78
Figura 91 - layout modello Simulink per lo scambio dati con Arduino.....	79
Figura 92 - Serial Send	79
Figura 93 - Serial Receive.....	80
Figura 94 - esempio del codice di calcolo implementato sulle schede Arduino	82

Figura 95 - potenza elettrica da PV per il periodo considerato (time step: 5 minuti).....	84
Figura 96 - richiesta elettrica dell'utenza per il periodo considerato (time step: 5 minuti)	84
Figura 97 - richiesta termica dell'utenza per il periodo considerato (time step: 5 minuti)	85
Figura 98 - segnali di output instabili	86
Figura 99 - segnali di output stabilizzati	86
Figura 100 - potenza elettrica richiesta dalla pompa di calore (time step: 5 minuti)	88
Figura 101 - livello di acqua calda nell'accumulo termico (time step: 5 minuti).....	88
Figura 102 - potenza elettrica in surplus alla rete (time step: 5 minuti).....	89
Figura 103 - potenza elettrica proveniente dalla rete (time step: 5 minuti).....	89
Figura 104 - potenza elettrica in ingresso all'elettrolizzatore (time step: 5 minuti)	90
Figura 105 - portata di idrogeno prodotto (time step: 5 minuti)	90
Figura 106 - assorbimento elettrico del compressore (time step: 5 minuti)	91
Figura 107 - pressione interna al serbatoio di H ₂ (time step: 5 minuti)	91
Figura 108 - livello del serbatoio di H ₂ (time step: 5 minuti).....	92
Figura 109 - produzione elettrica della FC (time step: 5 minuti).....	92
Figura 110 - consumo di idrogeno della FC (time step: 5 minuti)	93
Figura 111 - efficienza della FC (time step: 5 minuti)	93
Figura 112 - tensione della batteria (time step: 5 minuti)	94
Figura 113 - SOC (state of charge) della batteria (time step: 5 minuti)	94
Figura 114 - potenza elettrica erogata dalla batteria (time step: 5 minuti).....	95
Figura 115 - potenza elettrica ricevuta dalla batteria (time step: 5 minuti)	95

8 Indice delle tabelle

Tabella 1 - Gamma completa schede Arduino [2]	12
Tabella 2 - Caratteristiche tecniche scheda Arduino Mega 2560 [2]	13
Tabella 3 - Modelli di schede Raspberry Pi prodotti dalla nascita della fondazione [26]	26
Tabella 4 - Lookup table per la caratterizzazione del pannello fotovoltaico	54
Tabella 5 - Lookup table per il calcolo della potenza elettrica generata normalizzata	55
Tabella 6 - taglie e parametri operativi principali del sistema	86

9 Curriculum del gruppo di lavoro

Il gruppo di lavoro del Dipartimento di Ingegneria Industriale dell'Alma Mater Studiorum – Università di Bologna che ha partecipato alla presente ricerca è composto da: Prof. Francesco Melino (responsabile scientifico dell'attività), dal Prof. Andrea De Pascale (membro del gruppo di lavoro), dall'Ing. Lisa Branchini (membro del gruppo di lavoro), dall'Ing. Maria Alessandra Ancona (membro del gruppo di lavoro), dall'Ing. Federico Ferrari (membro del gruppo di lavoro) e dall'Ing. Saverio Ottaviano (membro del gruppo di lavoro).

In breve:

Maria Alessandra Ancona, Ricercatrice di Sistemi e Macchine per l'Energia e l'Ambiente (ING/IND09), svolge attività di ricerca nel campo delle smart grid con particolare riferimento ai metodi e alle strategie di integrazione tra sistemi di accumulo dell'energia e generatori elettrici e termici rinnovabili non programmabili. Autrice di oltre 30 pubblicazioni scientifiche, prevalentemente su riviste e congressi internazionali.

Lisa Branchini, Ricercatrice Senior di Sistemi e Macchine per l'Energia e l'Ambiente (ING/IND09), svolge attività di ricerca – sia analitica che sperimentale – nel settore dei sistemi microcogenerativi innovativi oltre che nel campo dell'integrazione tra generatori rinnovabili e dispositivi di accumulo dell'energia elettrica e termica. Autrice di oltre 50 pubblicazioni scientifiche, prevalentemente su riviste e congressi internazionali.

Andrea De Pascale, Professore Associato di Macchine a Fluido (ING/IND08), responsabile del Laboratorio di Tecnologie di Micro-Generazione del Dipartimento di Ingegneria Industriale dell'Università di Bologna. Vice-Direttore del Corso di Dottorato In Meccanica e Scienze Avanzate dell'Ingegneria. Autore di oltre 100 pubblicazioni scientifiche, prevalentemente su riviste e congressi internazionali, svolge attività di ricerca sui temi del recupero energetico di cascami termici a bassa entalpia.

Federico Ferrari, Assegnista di Ricerca, svolge attività di ricerca – sia sperimentale che analitica – nell'ambito delle micro-reti energetiche, in particolare su sistemi per la produzione, l'accumulo e la riconversione dell'idrogeno.

Francesco Melino, Professore Associato di Macchine a Fluido (ING/IND08) responsabile del Laboratorio di Microreti di Generazione ed Accumulo dell'Università di Bologna, Vicedirettore del Centro Interdipartimentale di Ricerca Industriale su Fonti Rinnovabili, Ambiente, Mare ed Energia (CIRI-FRAME) dell'Università di Bologna e Chair dell'Industrial & Cogeneration Committee dell'American Society of Mechanical Engineering (ASME). Autore di oltre cento pubblicazioni scientifiche prevalentemente su riviste e congressi internazionali, svolge attività di ricerca sui temi delle reti complesse di trasporto dell'energia elettrica, termica frigorifera e del combustibile.

Saverio Ottaviano, Assegnista di Ricerca, svolge attività di ricerca sperimentale su sistemi di micro-generazione basati sul recupero di cascami termici a bassa entalpia. Autore di diverse pubblicazioni scientifiche prevalentemente su riviste e congressi internazionali.

