

# QUADERNI di GEOFISICA

Sviluppo di una stazione sismica  
low-cost basata su tecnologia MEMS



ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

153

**Direttore Responsabile**

Valeria DE PAOLA

**Editorial Board**

Luigi CUCCI - Editor in Chief (luigi.cucci@ingv.it)  
Raffaele AZZARO (raffaele.azzaro@ingv.it)  
Christian BIGNAMI (christian.bignami@ingv.it)  
Mario CASTELLANO (mario.castellano@ingv.it)  
Viviana CASTELLI (viviana.castelli@ingv.it)  
Rosa Anna CORSARO (rosanna.corsaro@ingv.it)  
Domenico DI MAURO (domenico.dimauro@ingv.it)  
Mauro DI VITO (mauro.divito@ingv.it)  
Marcello LIOTTA (marcello.liotta@ingv.it)  
Mario MATTIA (mario.mattia@ingv.it)  
Milena MORETTI (milena.moretti@ingv.it)  
Nicola PAGLIUCA (nicola.pagliuca@ingv.it)  
Umberto SCIACCA (umberto.sciacca@ingv.it)  
Alessandro SETTIMI (alessandro.settimi1@istruzione.it)  
Andrea TERTULLIANI (andrea.tertulliani@ingv.it)

**Segreteria di Redazione**

Francesca DI STEFANO - Referente  
Rossella CELI  
Barbara ANGIONI

redazionecen@ingv.it

**REGISTRAZIONE AL TRIBUNALE DI ROMA N.174 | 2014, 23 LUGLIO**

© 2014 INGV Istituto Nazionale di Geofisica e Vulcanologia  
Rappresentante legale: Carlo DOGLIONI  
Sede: Via di Vigna Murata, 605 | Roma



ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

# QUADERNI di GEOFISICA

## Sviluppo di una stazione sismica low-cost basata su tecnologia MEMS

### *Development of a low-cost seismic station based on MEMS technology*

Antonino D'Alessandro<sup>1</sup>, Luca Greco<sup>1</sup>, Salvatore Scudero<sup>1</sup>, Marianna Siino<sup>1</sup>, Giovanni Vitale<sup>1,2</sup>,  
Roberto D'Anna<sup>1</sup>, Fabio Di Gangi<sup>1</sup>, Daniele Nicolosi<sup>1</sup>, Giuseppe Passafume<sup>1</sup>, Stefano Speciale<sup>1</sup>,  
Maria Catania<sup>2,1</sup>, Paolo Cosenza<sup>3</sup>, Luigi Foresta Martin<sup>3</sup>, Domenico Patanè<sup>4</sup>, Raffaele Martorana<sup>2,1</sup>,  
Sandro Rao<sup>1</sup>, Salvatore Stramondo<sup>1</sup>

<sup>1</sup>INGV | Istituto Nazionale di Geofisica e Vulcanologia, Osservatorio Nazionale Terremoti

<sup>2</sup>Università degli Studi di Palermo | Dipartimento di Scienze della Terra e del Mare

<sup>3</sup>INGV | Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Palermo

<sup>4</sup>INGV | Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Catania - Osservatorio Etno

Accettato 18 marzo 2019 | Accepted 18 March 2019

Come citare | How to cite D'Alessandro A. et al., (2019). Sviluppo di una stazione sismica low-cost basata su tecnologia MEMS. Quad.  
Geofis., 153: 1-60.

In copertina Mappa della sismicità strumentale (dal 1985 al 2014,  $M_L \geq 2.0$ , lside <http://iside.rm.ingv.it>) | Cover Map of the instrumental  
seismicity

153



# INDICE

<b>Riassunto</b>	<b>7</b>
<b>Abstract</b>	<b>7</b>
<b>Introduzione</b>	<b>7</b>
<b>1. Stato dell'arte</b>	<b>11</b>
MEMS in Sismologia	11
Accelerometri MEMS	12
<b>2. Il prototipo di stazione accelerometrica MEMS</b>	<b>14</b>
2.1 Schema di progetto	14
2.2 Hardware	15
2.3 Software	17
<b>3. Analisi della performance della stazione accelerometrica MEMS</b>	<b>18</b>
3.1 Analisi delle specifiche	19
3.2 Valutazione della performance mediante analisi spettrale e simulazione numerica	20
3.3 Analisi evento sismico di S. Ninfa (ML 3.1 del 30.10.2018)	22
<b>4. Conclusioni</b>	<b>27</b>
<b>Riconoscimenti</b>	<b>27</b>
<b>Ringraziamenti</b>	<b>28</b>
<b>Bibliografia</b>	<b>28</b>
<b>Appendici 1-15</b>	<b>31</b>



# Riassunto

L'impatto di un forte terremoto su una comunità urbana può essere ridotto con una azione tempestiva e ben pianificata da parte degli organi di Protezione Civile preposti. In tale prospettiva è di fondamentale importanza che gli organi preposti alla sorveglianza sismica del territorio nazionale si dotino di strumenti di osservazione puntuale in grado di fornire nell'immediato terremoto informazioni utilissime per la gestione ottimale delle emergenze. Ciò può essere fatto oggi mediante la realizzazione reti sismiche urbane, con funzionamento in tempo reale (OSU - Osservatori Sismici Urbani), in grado di fornire in maniera tempestiva ai centri per la gestione delle emergenze, la distribuzione dell'intensità dello scuotimento a seguito di un terremoto. Le mappe di scuotimento potrebbero consentire agli organi preposti, un'ottimale gestione delle priorità e la ripartizione delle risorse in modo da ottenere una significativa riduzione del numero di vittime a seguito di un forte terremoto. Le verifiche del grado di danneggiamento e di agibilità degli edifici potrebbero così essere effettuate secondo una logica prioritaria sulla base dei massimi scuotimenti misurati dalla rete sismica urbana. La realizzazione di OSU per le finalità su preposte è l'obiettivo principale del progetto MEMS finanziato dal MIUR nell'ambito del programma SIR (Scientific Independence of young Researchers). La prima sfida affrontata nell'ambito del progetto MEMS è stata la progettazione e sviluppo di una stazione accelerometrica idonea agli scopi del progetto. Il presente lavoro ha lo scopo di descrivere nel dettaglio, la stazione accelerometrica basata su tecnologia MEMS (Micro Electro-Mechanical Systems), implementata nell'ambito dell'omonimo progetto. Ciò che stato realizzato è una stazione accelerometrica dalle ottime performance, estremamente versatile e a bassissimo costo.

## Abstract

*The impact of a strong earthquake onto an urban community can be reduced taking timely actions managed by the authority in charge for the civil protection. In this view, it is of fundamental importance that the authority in charge for the seismic surveillance is provided with the appropriate monitoring tools able to give the most useful information for the best emergency management in the immediate post-event. Today this goal can be reached realizing urban-scale, real-time seismic monitoring networks (Osservatori Sismici Urbani, OSU, in Italian). The real-time OSU networks are able to promptly provide information to the emergency authority about the ground shaking at each node after an earthquake. Maps showing the spatial distribution of the ground shaking could help in the optimization of the priorities and a good management of the rescue resources in order to reduce the number of victims. Also the assessment of the damage of the buildings could be carried out according to the logic of priority given by the shakings measured by the OSU. The realisation of OSUs is the main objective of the MEMS project. This project is funded by the MIUR under the SIR program (Scientific Independence of young Researchers). The first challenge of the MEMS project was the design and the development of a devoted accelerometric stations. This work illustrates in detail the accelerometric station based on MEMS technology (Micro Electro-Mechanical Systems). The final product is a low-cost accelerometric station with high performance and great versatility.*

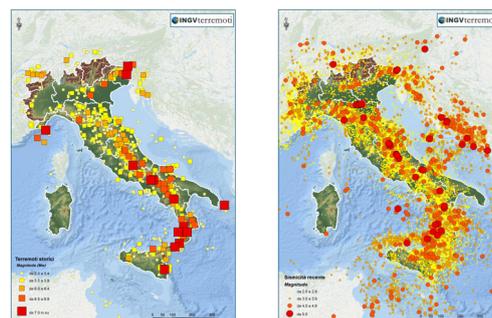
---

## Introduzione

L'elevato rischio sismico della penisola italiana è il risultato dell'elevata pericolosità sismica del

territorio nazionale e della presenza di numerosi centri urbani caratterizzati da nuclei storici altamente vulnerabili al terremoto. Dal 1900 ad oggi l'Italia è stata colpita da oltre 30 terremoti molto forti ( $M_w > 5.8$ ), la maggior parte dei quali hanno avuto effetti catastrofici, in particolare in termini di perdita di vite umane. Negli ultimi 35 anni la Rete Sismica Nazionale, gestita dall'Osservatorio Nazionale Terremoti dall'Istituto Nazionale di Geofisica e Vulcanologi (ONT-INGV), ha registrato oltre 150.000 eventi sismici (oltre 50 terremoti hanno avuto  $M_L > 5.0$ ) con epicentro sul territorio nazionale. In Figura 1, sono riportate la mappa della sismicità storica (dal 1000 al 2006, CPTI11, <http://emidius.mi.ingv.it/CPTI11/>) e la mappa della sismicità strumentale (dal 1985 al 2014,  $M_L \geq 2.0$ , [Iside http://iside.rm.ingv.it](http://iside.rm.ingv.it)) pubblicate su <https://ingvterremoti.wordpress.com/i-terremoti-in-italia/>.

**Figura 1** A sinistra: mappa della sismicità storica (dal 1000 al 2006, CPTI11, <http://emidius.mi.ingv.it/CPTI11/>); a destra: mappa della sismicità strumentale (dal 1985 al 2014,  $M_L \geq 2.0$ , [Iside http://iside.rm.ingv.it](http://iside.rm.ingv.it)); ambedue pubblicate su <https://ingvterremoti.wordpress.com/i-terremoti-in-italia/>



**Figure 1** Left: map of the historical seismicity (from 1000 to 2006, CPTI11, <http://emidius.mi.ingv.it/CPTI11/>); right: map of the instrumental seismicity. They are both published on <https://ingvterremoti.wordpress.com/i-terremoti-in-italia/>

Un semplice confronto visivo delle due mappe permette di comprendere uno dei punti chiave della scienza dei terremoti: i terremoti si ripetono lì dove sono già avvenuti in passato. Un territorio storicamente colpito da forti terremoti, lo sarà sicuramente anche in futuro. Anche se ad oggi non siamo ancora in grado di prevedere in maniera deterministica l'accadimento di un terremoto, siamo però sicuramente in grado, sulla base dello studio della sismicità storica e strumentale, degli studi geodetici e di geologia del terremoto, di individuare le aree a maggior pericolosità sismica. Le mappe di pericolosità sismica sono strumenti fondamentali di prevenzione, in particolare per la valutazione del rischio e degli interventi sulle aree densamente urbanizzate, e quindi esposte, del nostro paese.

Negli ultimi decenni, la crescita demografica e la conseguente espansione dei centri urbani, già iniziata dal dopoguerra, quasi sempre senza una opportuna pianificazione urbanistica in grado di tenere conto dei molteplici fattori di rischio, ha portato ad un aumento significativo dell'esposizione delle aree urbanizzate al rischio indotto dal terremoto.

Come ben noto, l'esposizione al terremoto dipende essenzialmente dall'intensità dello scuotimento e dalla vulnerabilità degli edifici colpiti dal sisma. L'intensità dello scuotimento, dipende da molteplici fattori caratterizzanti il sisma, quali la magnitudo, la distanza epicentrale e la profondità ipocentrale e da altri fattori locali, principalmente geologici e topografici, in grado di modificare sensibilmente lo scuotimento atteso al sito (effetti di sito). Gli studi di pericolosità, esposizione e vulnerabilità al terremoto, permettono oggi di determinare, quantomeno per alcune aree, mappe del rischio sismico di elevato dettaglio. È da questa conoscenza che bisogna partire per affrontare correttamente il "problema terremoto" ponendosi una semplice domanda: *come possiamo ridurre l'impatto di un forte terremoto?*

Tale domanda ammette numerose risposte egualmente valide, ma tutte richiedono forti investimenti sia in termini economici che di tempo. Una delle soluzioni, concettualmente la più semplice, è quella di ridurre la vulnerabilità al terremoto tramite interventi sistematici sul patrimonio edilizio, in particolare quello storico non antisismico. Ciò permetterebbe di ridurre

enormemente l'esposizione, permettendo di salvare sia vite umane, ma anche di mettere in sicurezza il patrimonio artistico e monumentale di inestimabile valore di cui è ricco il nostro paese. Una seconda soluzione potrebbe essere quella di pervenire ad un sistema di allertamento precoce per il terremoto (Early Warning System, EWS). Gli EWS per il terremoto [vedi ad esempio Gasparini et al., 2007; e riferimento al suo interno], potrebbero sicuramente ridurre l'impatto di un sisma fornendo allerte con preavvisi da qualche secondo a qualche decina di secondi prima che arrivi lo scuotimento di maggiore intensità. Tale allertamento precoce, sarebbe di grande utilità per attivare azioni automatizzate di messa in sicurezza di impianti industriali, linee ferroviarie, ecc., ma necessiterebbe lo sviluppo di un sistema di comunicazione alla popolazione, per diventare un efficace sistema di riduzione del rischio.

Senza entrare nel dettaglio degli approcci di riduzione della vulnerabilità sismica sopra accennati, per i quali si rimanda all'ampia bibliografia sulle moderne tecniche per la ristrutturazione e la messa in sicurezza degli edifici storici e di EWS per il terremoto, come già peraltro anticipato, questi approcci necessitano di importanti investimenti economici e di tempo.

N° giorni dopo il terremoto	N° persone estratte	N° persone vive	Probabilità di mantenimento in vita (%)
0	379	333	87.9
1	57	20	35.3
2-3	35	30	8.5
>4	77	0	0.0

**Tabella 1** Tabella sinottica delle estrazioni di persone dalle macerie a seguito del Terremoto dell'Irpinia del 1980 (Fonte: Protezione Civile Nazionale).

**Table 1** Table summarizing the rescued people after the 1980 Irpinia earthquake (source: Italian Civil Protection).

Negli ultimi decenni la comunità scientifica è andata alla ricerca nuove tecniche e approcci smart per la riduzione del rischio sismico, cercando soluzioni a basso costo e implementabili in tempi brevi. Anche se è impossibile prevedere deterministicamente un terremoto, è possibile sicuramente prevedere il suo effetto sul territorio dopo che è avvenuto. Questo oggi viene fatto mediante la creazione di mappe di scuotimento (shake maps, <http://shakemap.rm.ingv.it/shake/index.html>) a scala regionale, che però forniscono solo informazioni approssimative sul potenziale impatto di un terremoto. Affinché uno strumento come le mappe di scuotimento possa rivelarsi un utile supporto alla Protezione Civile, deve fornire informazioni di elevatissimo dettaglio, utili nell'ottimizzazione degli eventi di soccorso, finalizzati alla riduzione del numero di vittime.

La Tabella 1 mostra il numero di persone tratte in salvo, in funzione del tempo, in seguito al Terremoto dell'Irpinia ( $M_L$  6.9) del 1980 (Fonte: Protezione Civile Nazionale). Come è possibile osservare, oltre 86% delle persone rimaste in vita a seguito del terremoto, furono soccorse entro le prime 24 ore. Analisi statistiche condotte su un'ampia casistica di terremoti distruttivi, hanno evidenziato come la probabilità di estrarre persone vive dopo le prime 48-72 ore (the Golden hours) sia estremamente bassa e quindi come la tempestività degli eventi di soccorso sia fondamentale per la riduzione del numero di vittime. Un esempio noto, che permette di chiarire tale necessità, è il terremoto Iraniano del 2003 ( $M_L$  6.6). A fronte delle 43.000 vittime furono salvate solo 30 persone; infatti, anche se i soccorsi furono considerevoli (ben 1.600 soccorritori provenienti da 43 paesi diversi) questi arrivarono dopo ben 72 ore dalla scossa principale.

Tra gli strumenti proposti dalla comunità scientifica, spiccano sicuramente in termini di

innovatività, basso costo ed efficacia, lo sviluppo degli Osservatori Sismici Urbani (OSU). Gli OSU sono reti sismiche urbane, con trasmissione dati in tempo reale, in grado registrare fedelmente le accelerazioni generate da forti terremoti.

L'analisi in tempo reale delle forme d'onda registrate, permette di estrarre nell'immediato post-terremoto informazioni utilissime nella gestione per gli interventi di soccorso della Protezione Civile, sintetizzate in mappe dei principali parametri di scuotimento ovvero shake map a scala urbana. Una shake map a scala urbana permette di misurare direttamente (senza fare ricorso ad interpolazioni di dati su scala regionale) lo scuotimento risentito sull'area urbana coperta dalla rete a seguito sia del terremoto. Tale scuotimento misurato, tiene conto non solo dell'energia rilasciata dal terremoto, ma anche di eventuali effetti di amplificazione del moto legati a particolari condizioni geologiche-topografiche di sito. Incrociando poi le mappe di scuotimento, con quelle della vulnerabilità degli edifici, è possibile ottenere in tempi brevissimi scenari di danno di elevatissimo dettaglio e attendibilità. Queste informazioni favorirebbero l'ottimizzazione degli interventi di soccorso da parte degli organi di Protezione Civile competenti. I centri per la gestione dell'emergenza post-terremoto potrebbero utilizzare tali mappe per decidere le priorità di intervento al fine di minimizzare la perdita di vite umane, gestendo in maniera ottimale le risorse disponibili.

A questo punto è lecito porsi una seconda domanda: *per quale motivo uno strumento così utile per fini di sorveglianza e Protezione Civile sta iniziando a svilupparsi solo in questi ultimi anni?*

La risposta è semplice; sino a qualche anno fa la realizzazione di un OSU sarebbe stata anch'essa estremamente onerosa sia in termini economici che di tempo, a causa degli elevati costi della strumentazione sismica (sensore, datalogger, sistema di alimentazione e trasmissione dati) e alle tempistiche legate alla complessità della loro messa in opera.

Fortunatamente, i recenti sviluppi tecnologici nel campo dei sistemi MEMS (Micro Electro-Mechanical Systems), permettono oggi di realizzare stazioni accelerometriche a costi estremamente contenuti, di piccolo ingombro e di facile installazione, ideali per la realizzazione di reti sismiche urbane ad elevata densità di punti di misura. Negli anni 90, gli accelerometri MEMS hanno rivoluzionato l'industria automobilistica e la sicurezza degli autoveicoli, permettendo la realizzazione dei moderni sistemi di airbag. In questo ultimo decennio, i dispositivi MEMS hanno trovato largo impiego in molteplici campi, ed oggi sono ampiamente impiegati in computer portatili, periferiche di gioco, tablet, smartphone, droni, ecc. Grazie ai grandi successi commerciali, la ricerca e lo sviluppo della tecnologia MEMS prosegue attivamente in tutto il mondo. Oggigiorno, la sensibilità e il range dinamico di tali sensori sono tali da permettere la registrazione di terremoti di moderata magnitudo anche a distanza di diverse decine di chilometri [Cocharn et al., 2012; D'Alessandro and D'Anna, 2013-2014a; Evans et al., 2014]. Inoltre, grazie al loro basso costo e alle loro ridotte dimensioni, gli accelerometri MEMS possono essere facilmente installati all'interno di aree urbane per la realizzazione di reti sismiche ad alta densità di stazioni. In California e in Giappone, reti interamente costituite da sensori MEMS sono in fase di sviluppo come la Quake-Catcher Network [Cochran et al., 2009; Chung et al., 2011; Kohler et al., 2013; Chung et al., 2015] gestita dall'Università di Stanford, la Community Seismic Network [Clayton et al., 2011; Kohler et al., 2013] gestita dal California Institute of Technology e la Home Seismometer Network [Horiuchi et al., 2009] gestita dalla Japan Meteorological Agency. Tali sistemi, sono oggi ben consolidati e diffusi e spesso rappresentano anche la base per sistemi di allerta precoce per il terremoto [Pozzi et al., 2011; Zheng et al., 2011; Lawrence et al., 2014; Houque et al., 2015; Kim et al., 2017].

In Italia, la realizzazione degli OSU è iniziata pochi anni fa [D'Alessandro, 2016], a seguito del finanziamento da parte del MIUR (Ministero dell'Istruzione, dell'Università e della Ricerca) nell'ambito del programma SIR (Scientific Independence of young Researchers), del progetto MEMS (Monitoring of Earthquakes through MEMS Sensors: project for the realization of a prototype of real-time urban seismic network based on MEMS technology).

Lo scopo del presente lavoro è quello di illustrare nel dettaglio la stazione sismica accelerometrica, basata su sensoristica MEMS, che è stata sviluppata nell'ambito dell'omonimo progetto. Questa è stata la prima sfida affrontata, il cui successo oggi sta portando all'implementazione di diversi OSU in un territorio ad elevata sismicità come la Sicilia [Siino et al., 2018], in particolare in corrispondenza dei centri urbani a maggior rischio sismico ricadenti in Sicilia Orientale [D'Alessandro et al., 2014b; D'Alessandro et al., 2018a; D'Alessandro et al., 2018b].

Nel presente lavoro, dopo un breve stato dell'arte sulla tecnologia MEMS e il suo impiego in campo sismologico, saranno descritti nel dettaglio l'hardware e il software della stazione accelerometrica. Sarà valutata la performance della stazione, tramite simulazioni numeriche e dati sperimentali, e infine, verranno fatte alcune considerazioni sulle potenzialità di tale strumento per la realizzazione di OSU e più in generale per la sismologia strong motion.

## 1. Stato dell'arte

I sensori MEMS sono stati riconosciuti come una delle tecnologie più promettenti del XXI secolo, capaci di rivoluzionare sia il mondo industriale, sia quello dei prodotti di largo consumo. I microsistemi elettromeccanici, non sono altro che un insieme di dispositivi di varia natura (meccanici, elettrici ed elettronici) integrati in forma altamente miniaturizzata su uno stesso substrato di silicio, che coniugano le proprietà elettriche degli integrati a semiconduttore con proprietà opto-meccaniche [Jones and Nedadic, 2013]. Si tratta dunque di sistemi "intelligenti" che abbinano funzioni elettroniche e meccaniche in uno spazio ridottissimo.

Il funzionamento di un MEMS si può descrivere considerando il circuito integrato come il "cervello" del sistema che rende possibile il monitoraggio dell'ambiente circostante tramite gli altri dispositivi (sensori) presenti sullo stesso chip. In questo modo il sistema può raccogliere le informazioni misurando fenomeni meccanici, termici, biologici, ottici e magnetici; l'elettronica processa le informazioni derivate dai sensori, monitorando le variazioni avvenute nell'intervallo di tempo di osservazione nell'ambiente circostante. Un dispositivo MEMS è quindi un sistema in grado di captare informazioni dall'ambiente traducendo le grandezze fisiche in impulsi elettrici, di elaborare tali informazioni facendo uso di opportune logiche, rispondendo con alcune azioni. I sensori possono misurare fenomeni di varia natura: meccanica (accelerazioni e pressioni, per fare alcuni esempi), termica (temperatura e flusso di calore), biologica (potenziale cellulare), chimica (pH), ottica (intensità della radiazione luminosa, spettroscopia), magnetica (intensità del flusso). La tecnologia MEMS promette di rivoluzionare intere categorie di prodotti e settori di ricerca e sviluppo proprio per il fatto di integrare in uno stesso dispositivo le funzioni più diverse. Un minuscolo chip di silicio diventa ora un sensore di pressione, ora un accelerometro, ora un giroscopio e così via. I vantaggi dei MEMS si possono riassumere nel fatto che sono in grado di eseguire le stesse funzioni di rilevazione, elaborazione e attuazione di oggetti molto più ingombranti e costosi.

### 1.1 MEMS in sismologia

L'impiego della tecnologia MEMS nel campo della sismologia risale a tempi relativamente recenti. Il primo utilizzo, documentato in letteratura, di un accelerometro MEMS per applicazioni nel campo dei terremoti risale al 1979 all'Università di Stanford. Soltanto 15 anni più tardi i dispositivi MEMS, da prototipi sperimentali, sono diventati prodotti commerciali impiegati su vasta scala per airbag, sistemi di navigazione, monitoraggio di vibrazioni di macchine o strutture, e anche per il monitoraggio sismico [Eddy and Sparks, 1998; Maluf and Kirt, 2004; Scudero et al., 2018; D'Alessandro et al., 2019].

Nella prima edizione di “Instrumentation in Earthquake Seismology” [Havskov & Alguacil, 2004], un riferimento bibliografico importante per chi lavora nel campo della sismologia strumentale, gli accelerometri MEMS non sono nemmeno menzionati; soltanto nella seconda edizione, pubblicata quattro anni più tardi, appare una apposita sezione. Quindi i sensori MEMS rappresentano oggi una nuova frontiera nel monitoraggio sismico, consentendo il superamento di molte delle limitazioni dei sensori tradizionali: dimensioni, peso, costi, uscita analogica [Merchant, 2009]. Inoltre, insieme ai progressi tecnologici raggiunti in termini di capacità di calcolo, di trasmissione dati e di capacità di storage, si sono aperte le porte ad una molteplicità di potenziali applicazioni nell’ambito della sismologia o in settori strettamente collegati. A fronte degli innumerevoli vantaggi e potenzialità, naturalmente sono presenti anche alcune limitazioni, essenzialmente legate alle performance di tali sensori che non raggiungono ad oggi quelle degli analoghi “fratelli maggiori”.

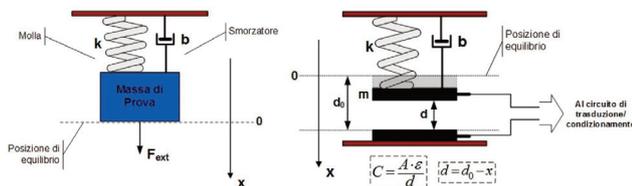
Gli accelerometri MEMS possono essere divisi in due grandi categorie: accelerometri per misure di accelerazione stazionaria, ovvero in grado di rilevare accelerazioni statiche costanti come ad esempio l’accelerazione di gravità e in accelerometri per misure di accelerazione che variano nel tempo come ad esempio quelle generate da urti o vibrazioni. I primi in genere hanno una banda passante che va da 0 a 500 Hz e possono essere usati anche come inclinometri, mentre i secondi hanno una banda passante compresa tra qualche decina di Hz fino alle decine di kHz. Gli accelerometri MEMS di tipo capacitivo hanno diversi vantaggi: possono misurare sia accelerazioni statiche che dinamiche, sono poco sensibili alle variazioni di temperatura, hanno un’ampia banda passante, hanno un’elevata sensibilità, alta linearità, bassa dissipazione di potenza ed un costo molto basso. Gli accelerometri di questo tipo si prestano bene ad essere integrati, assieme al relativo sistema di trasduzione e condizionamento del segnale, in un unico chip di silicio. Tali sensori sono quelli oggi più utilizzati per applicazioni sismologiche e descritti con maggiore dettaglio nel paragrafo seguente. Per maggiori dettagli sulla tecnologia MEMS si rimanda a Jones and Nedadic [2013] e D’Alessandro et al., [2019].

## 1.2 Accelerometri MEMS

Gli strumenti tradizionalmente usati in sismometria sono in grado di misurare il movimento del suolo, (più comunemente la sua velocità o la sua accelerazione) sulla base di un sistema massa-molla con smorzamento, la cui oscillazione è convertita in un segnale elettrico continuo (Figura 2). A parte le dimensioni, il sistema di funzionamento di un dispositivo MEMS di tipo capacitivo (il più diffuso) è del tutto analogo. La massa di quest’ultimo, di dimensioni ridottissime ( $\sim 0.1 \mu\text{g}$ ), per essere in grado di produrre un segnale elettrico rilevabile, deve essere sottoposta a delle condizioni di sollecitazione superiori ad una soglia minima; per questo motivo i sensori MEMS rispondono in maniera migliore al c.d. strong motion, e alle frequenze più elevate dello spettro sismico. Pertanto, per le loro specifiche costruttive, essi sono adatti a rilevare l’accelerazione del moto del suolo piuttosto che la sua velocità.

Gli accelerometri capacitivi sono i più diffusi in quanto le misurazioni sono affidabili, il processo di produzione è relativamente semplice e conseguentemente i costi sono contenuti. Anche la stazione accelerometrica descritta in questo lavoro si basa su questa tipologia di sensore. Gli accelerometri capacitivi MEMS si basano su un sistema massa-molla-smorzatore (Figura 2) miniaturizzato, direttamente posizionato su di un substrato di silicio, che sfrutta come principio di funzionamento la variazione della capacità elettrica di un condensatore al variare della distanza tra le sue armature. La massa stessa, realizzata con materiale conduttivo, costituisce un’armatura del condensatore; l’altra (o le altre due se costruito con tecnologia differenziale), sono invece fisse alla struttura del dispositivo. La massa viene tenuta sospesa

grazie ad un elemento elastico in modo che le armature non entrino in contatto. Uno schema funzionale dell'accelerometro capacitivo è riportato Figura 2, a destra. La massima deflessione che può subire la massa è pari alla distanza tra le armature in condizione di equilibrio (con forze applicate nulle). Un apposito circuito infine genera un segnale elettrico proporzionale alla capacità del condensatore e quindi all'accelerazione subita dal sistema.



**Figura 2** A sinistra schema di sistema Massa-Molla-Smorzatore di un accelerometro; a destra schema funzionale di un accelerometro capacitivo.

*Figure 2 Left: scheme of a damped mass-spring system employed for accelerometers; right; functioning scheme of a capacitive accelerometer.*

Ogni sensore è generalmente costituito da diversi condensatori al fine di ottenere valori complessivi di capacità maggiori e conseguentemente maggiori valori di differenza di capacità a parità di spostamento, valori che altrimenti potrebbero essere troppo piccoli da osservare. Le dimensioni di un singolo sensore capacitivo sono dell'ordine di alcune centinaia di  $\mu\text{m}$ , con spessori dell'armature di poche decine di  $\mu\text{m}$ . Ogni sensore capacitivo è in grado di misurare solamente una componente del moto, pertanto la maggior parte dei dispositivi MEMS integra tre diversi sensori. L'imballaggio finale di un sensore triassiale con uscita digitale, come quelli impiegati nella realizzazione della stazione accelerometrica, è di pochi mm per lato.

La tecnica di realizzazione è abbastanza semplice, in quanto è la stessa con la quale si realizza un qualsiasi circuito integrato (processo fotolitografico). Il sistema elettrico che si occupa della traduzione è nella pratica un circuito schematizzabile con un'architettura di misura ad anello aperto. Le armature fisse del condensatore sono quindi pilotate da un segnale sinusoidale (oppure da un'onda quadra) ad altissima frequenza (es. 100 kHz) generato da un circuito oscillante: alla prima armatura viene fornito il segnale non sfasato, mentre alla seconda viene fornito lo stesso segnale ma sfasato di  $180^\circ$ . Il segnale in uscita dall'elettrodo centrale passa attraverso un opportuno amplificatore; in uscita da quest'ultimo si ha un segnale periodico (del tutto analogo a quello generato dall'oscillatore) la cui ampiezza è proporzionale alla capacità differenziale. Successivamente un demodulatore ed un filtro "passa basso" fanno sì che il segnale in uscita (disponibile all'utente) contenga l'informazione sul segno dell'accelerazione applicata e sia ripulito dalle componenti ad alta frequenza dovute al segnale di pilotaggio.

Il segnale in uscita sarà chiaramente analogico o digitale a seconda del tipo di sfasatore utilizzato. Uno sfasatore analogico permette una variazione continua dello sfasamento, mentre uno sfasatore digitale è costituito da una cascata di elementi (per semplicità, indicati come bit) che consentono soltanto incrementi discreti, pesati con logica binaria.

Le specifiche di un accelerometro MEMS dipendono in ultima analisi dal tipo di applicazione; un accelerometro per applicazioni sismologiche dovrebbe garantire sensibilità e risoluzione sufficienti per la registrazione di forti terremoti anche in area epicentrale in tutta la banda di frequenze di interesse per la sismologia, in particolare per la branca dell'ingegneria sismica. È fondamentale quindi scegliere il sensore con le giuste specifiche e procedere con l'esecuzione di opportuni test per la caratterizzazione del rumore autoprodotta o self-noise. Il self-noise è il risultato della combinazione di diversi tipi di sorgenti; alcune di esse possono essere facilmente stimate in modo deterministico; altre hanno natura stocastica e richiedono una specifica modellazione. Il self-noise dell'accelerometro MEMS impiegato per la

realizzazione della stazione sismica in oggetto è stato già oggetto di studio; una completa caratterizzazione è riportata in D'Alessandro et al. [2017].

## 2. Il prototipo di stazione accelerometrica MEMS

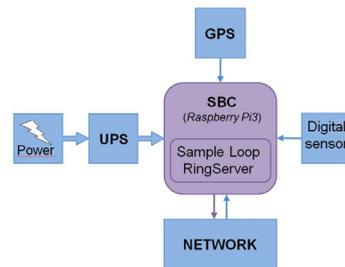
Nell'ambito del progetto MEMS è stata sviluppata una stazione sismica open-source a basso costo. La stazione accelerometrica digitale all-in-one è stata progettata in modo da avviarsi autonomamente, subito dopo essere stata alimentata, ed in pochi secondi, iniziare ad inviare i dati in tempo reale acquisiti da un apposito sensore MEMS digitale. La stazione è in grado di trasmettere dati in rete, utilizzano il protocollo di trasmissione standard miniSEED, tramite sia WiFi, che cavo Ethernet. La stazione è dotata di GPS (Global Positioning System) per la sincronizzazione dei segnali al tempo UTC (Coordinated Universal Time) ed un piccolo gruppo UPS (Uninterruptible Power Supply) che le garantisce 6 ore di autonomia dalla rete elettrica.

### 2.1 Schema di progetto

La stazione sismica MEMS è stata realizzata secondo lo schema di progetto riportato in Figura 3. Questo prevede una fonte di alimentazione (rete elettrica, batteria o altra fonte) per le componenti elettroniche, un UPS finalizzato a stabilizzare la tensione di alimentazione e garantire il funzionamento della stazione anche in caso di sospensione della fonte di alimentazione, un sistema di sincronizzazione dei segnali acquisiti con accuratezza temporale inferiore ad 1 ms, un sensore di movimento performante, capace di percepire lo scuotimento generato da un forte terremoto, un sistema di acquisizione e compressione dei dati infine un sistema di trasmissione dati in tempo reale basato su protocollo miniSEED (Ring-Server).

Figura 3 Schema di progetto della stazione sismica MEMS.

Figure 3 Working scheme of the MEMS seismic station.



Gli elementi che compongono la stazione sismica, per semplicità indicati di seguito come Power, UPS, GPS, SENSOR, NETWORK e SBC, descritti di seguito possono essere chiaramente sostituiti con componenti o apparati che svolgono le stesse funzioni, ma che magari hanno caratteristiche tecniche differenti o migliori. Di seguito verranno descritti dettagliatamente le funzionalità e le caratteristiche tecniche di interesse delle varie parti schematizzate.

**POWER:** o fonte di alimentazione primaria, presenta due tipi di sotto sistemi che possono intercambiarsi in funzione del tipo di alimentazione localmente disponibile (rete elettrica o batterie/fotovoltaico). Se la stazione sarà alimentata da rete elettrica, verrà utilizzato un convertitore AC-DC, se alimentata tramite batteria o impianto fotovoltaico, verrà utilizzato un convertitore DC-DC. In entrambi i casi le caratteristiche tecniche più importanti da attenzionare sono il range di tensione in ingresso e la tensione/amperaggio d'uscita.

**UPS:** o riserva di potenza non interrompibile (Uninterruptible Power Supply), costituito da una batteria di grande capacità (batteria tampone) con annesso regolatore di carica, sistema di bypass e convertitore DC-DC per adattare l'uscita in tensione al carico. Questo apparato permette al

sistema di ottenere ore o giorni di autonomia in assenza di alimentazione primaria. In particolare il sistema conserva energia all'interno di una batteria di grande capacità, che viene caricata tramite il regolatore di carica prelevando energia dalla fonte di alimentazione primaria, quando questa è disponibile. Durante la fase di carica il sistema di bypass interviene collegando direttamente la fonte di alimentazione primaria all'ingresso del convertitore DC-DC. Nel caso in cui venga a mancare l'alimentazione primaria, il sistema di bypass provvede in tempi brevissimi a collegare all'ingresso del convertitore DC-DC la batteria tampone, garantendo una tensione stabile all'uscita dell'UPS, anche durante la fase di transizione. Inoltre, il convertitore DC-DC, permette di proteggere la parte elettronica a seguito di eventuali sbalzi di tensione e corrente provenienti dalla fonte di alimentazione primaria. Le caratteristiche tecniche più importanti da attenzionare sono il range di tensione/amperaggio in ingresso, la tensione/amperaggio d'uscita e capacità della batteria tampone in Ah.

**GPS:** l'utilizzo di questo apparato è necessario per effettuare con elevata precisione (errore di sincronizzazione massimo pari a poche decine di nanosecondi) la sincronizzazione del segnale sismico con il tempo UTC (tempo di riferimento per la comunità scientifica), ovvero il tempo del meridiano zero (di Greenwich) ed effettuare la georeferenziazione della stazione sismica con una buona precisione (intervallo di confidenza della localizzazione dell'ordine di un paio di metri). La sincronizzazione temporale di precisione è fondamentale per rendere i dati acquisiti validi per l'uso scientifico e poterne effettuare elaborazione utili anche ad esempio per localizzazione di terremoti. Le caratteristiche tecniche più importanti da attenzionare sono il tempo di aggiornamento dei dati e l'incertezza del segnale PPS (pulsazione per secondo).

**SENSOR:** è il sensore digitale scelto per "rilevare" il moto del suolo. Esso va selezionato sulla delle esigenze specifiche del progetto, tenendo in considerazione le numerose specifiche quali: la banda di frequenza, la sensibilità, il fondo scala, la dinamica, il range di linearità, la frequenza di campionamento in uscita, il numero di bit, il self-noise strumentale, ecc. La scelta del sensore è determinante e va fatta con grande attenzione, verificando in ogni caso sperimentalmente le specifiche dichiarate dalla casa produttrice. Inoltre, tra tutti gli elementi qui elencati, è quello che può oscillare su fascia di prezzo maggiore, in funzione delle sue performance.

**NETWORK:** sono tutti quei dispositivi che permettono di interfacciarsi ad una rete dati. Vi sono diverse tecnologie per connettersi, WiFi, Ethernet, Bluetooth, GSM o Satellitare. Il criterio di scelta per il dispositivo da utilizzare dipende fortemente dal traffico dati, dall'affidabilità della tecnologia che si vuole utilizzare e il costo annuale per mantenere la rete attiva e funzionale.

**SBC:** o Single Board Computer è il cuore del sistema, un computer miniaturizzato realizzato interamente su unica scheda, con a bordo un sistema operativo e dotato di tutte le periferiche necessarie a svolgere i processi di incameramento, compressione e trasmissione dati. Le principali caratteristiche tecniche da osservare sono legate al sistema operativo ed alle applicazioni che si vogliono far eseguire sul sistema, in particolare la tipologia e performance del processore, il suo clock, le dimensioni della RAM e della ROM.

## 2.2 Hardware

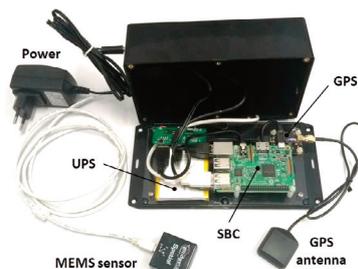
Seguendo le specifiche di progetto, sono stati individuati prodotti commerciali di prototipazione rapida per realizzare la prima versione della stazione sismica MEMS (Figura 4). La stazione, da considerarsi assolutamente *smarth*, *customizabile* e *low-cost*, nella configurazione di seguito descritta, ha un costo complessivo di circa 200 € (vedi anche appendice 15). L'utilizzo di parti meno o più performanti può chiaramente far variare il prezzo notevolmente. Le dimensioni del case per l'elettronica, estremamente ridotte (220 x 110 x 60 mm), rendono tale stazione compatta, leggera ed estremamente versatile.

**POWER:** prevedendo di installare le stazioni MEMS all'interno di edifici di varie pubbliche

amministrazioni (Comuni, Province, Regione), e considerando che tutti gli apparati della stazione sismica sperimentale sono alimentabili a 5 Volt, è stata scelta come fonte primaria d'alimentazione, un alimentatore di rete elettrica. Esso ha la particolarità di poter ricevere in ingresso un intervallo di tensione alternata nel campo 110 - 230 Volt e 40 - 60 Hertz, restituendo in uscita 5 Volt stabilizzati e 2.5 Ampere di corrente massima (tale valore risulta sovradimensionato di un fattore di protezione pari a 2.5 in quanto l'apparato in realtà consuma circa 1 Ampere nei momenti più critici). L'alimentatore è inoltre munito di dispositivi di protezione per eventuali cortocircuiti o sovraccarichi tramite l'ausilio di fusibili auto ripristinanti e protezione elettronica da cortocircuito.

Figura 4 Stazione Sismica MEMS.

Figure 4 MEMS seismic station.



**UPS:** per le stesse ragioni di cui sopra, si è deciso di utilizzare come sistema UPS un Power Bank a 5 Volt, con capacità di 5200 mAh, con batterie ai polimeri di litio. Il Power Bank è dotato di protezione integrata contro eccessiva scarica o sovraccarica del pacco batterie, protezione per eventuali cortocircuiti o sovraccarichi dell'uscita tramite l'ausilio di fusibili auto ripristinanti e protezione elettronica da cortocircuito (con l'eccezione che può fornire picchi di assorbimento di 5 Ampere). I valori di tensione/amperaggio d'ingresso sono rispettivamente 5V/1A; i valori di tensione/amperaggio d'uscita sono rispettivamente 5V/2A. La capacità della batteria tampone è sufficiente a garantire 6 ore di autonomia dalla fonte di alimentazione primaria.

**GPS:** Il Phidget 1040 è un GPS commerciale prodotto dall'azienda canadese Phidget; tramite i driver forniti dal costruttore è possibile realizzare facilmente e in vari linguaggi di programmazione, piccoli listati per l'estrapolazione dei dati d'interesse come il tempo UTC (ma anche altre informazioni sulla navigazione come velocità, posizione, direzione e qualità del segnale). Munito di batteria tampone al litio ricaricabile da 3V, è capace di effettuare "hot start" mantenendo l'orario precedente tramite un RTC (Real Time Clock) interno. Con questo sistema all'accensione si ottiene un allineamento temporale nell'arco di un secondo. La batteria tampone completamente carica, garantisce di mantenere le informazioni orarie sull'RTC per un mese senza alimentazione; il tempo di ricarica richiede 24 ore. L'accuratezza temporale garantita dal costruttore è inferiore a 5µs, può agganciare fino a 14 satelliti contemporaneamente, di cui elabora la stringa di informazione GNSS (Global Navigation Satellite System).

**DIGITAL SENSOR:** Il PhidgetSpatial precision 1043 0/0/3 è un accelerometro commerciale a 3 assi fabbricato dall'azienda canadese Phidget; tramite i driver forniti dal costruttore è possibile realizzare facilmente in vari linguaggi di programmazione piccoli listati per l'estrapolazione dei dati relativi all'accelerazioni misurate dai tre assi. Il sensore a riposo entra in modalità di alta precisione misurando accelerazioni entro  $\pm 2g$ . Il 1043 trova applicazione anche come inclinometro o accelerometro per la navigazione inerziale. Maggiori dettagli sull'accelerometro MEMS 1043 sono riportati al sottoparagrafo 3.1.

**SBC:** la SBC scelta è il Raspberry Pi 3 modello B. Il Raspberry è una SBC a basso costo opensource con sistema operativo derivante dal Debian, nominato Raspbian. Sviluppata dai suoi ideatori per creare una scheda di riferimento per lo sviluppo di progetti di rete, presenta enormi capacità di calcolo, ottime risorse hardware e tutte le periferiche necessarie per l'applicazione in campo sismologico. Il costo ridotto e la funzione "plug-and-play" fanno di Pi una scheda

accessibile a tutti, che dispone di numerose opzioni di connettività. Pi è la soluzione perfetta per esperimenti server o dispositivi di monitoraggio/sicurezza. Dotata di Chipset BCM2837 Broadcom a 1,2 GHz, Quad-core a 64 bit ARM Cortex-A53, LAN wireless 802.11 b/g/n con antenna integrata in chip, Bluetooth 4.1 (classico e a risparmio energetico), Co-processore multimediale Videocore IV® dual core, Memoria LPDDR2 da 1 GB, Connettore microUSB per un alimentatore da 2,5 A, porta Ethernet 10/100, 4 porte USB 2.0, 40 pin GPIO, Connettore display DSI, Slot scheda microSD. In particolare vengono utilizzate due porte USB per connettere GPS ed Accelerometro Phidget, la porta Ethernet; in casi eccezionali, dove non è possibile cablare fisicamente il cavo CAT5 all'interno della struttura ospitante, si procede ad attivare la connessione LAN wireless. Inoltre essa si fa carico di eseguire i processi di incameramento, compressione in miniSEED dei dati tramite codice Python ed effettua la trasmissione dei pacchetti creati tramite l'applicativo Ring Server della IRIS (Incorporated Research Institutions for Seismology).

## 2.3 Software

Le componenti software della stazione possono essere schematicamente divise: i programmi forniti da parti terze per il funzionamento dei componenti e per l'implementazione del protocollo Seedlink (<https://ds.iris.edu/ds/nodes/dmc/services/seedlink/>) e software sviluppato per l'acquisizione e la scrittura del segnale sismico e per la sincronizzazione della scheda con il dispositivo GPS.

Per l'impiego dell'accelerometro MEMS Phidget 1043 e del dispositivo GPS è necessaria l'installazione delle relative librerie "libphidget22" per sistema operativo Linux fornite dal produttore nella sua pagina web (<https://www.phidgets.com/downloads/phidget22/libraries/linux/libphidget22.tar.gz>).

Il protocollo Seedlink è uno standard internazionale per la trasmissione robusta di dati su reti internet che usa protocollo TCP/IP per l'effettiva connessione e l'invio dei dati. Permette anche la gestione di un buffer di dati in uscita per gestire eventuali perdite momentanee di connessione ed evitare la perdita del segnale in tali casi. Esso è normalmente gestito da un Seedlink server, che ha il compito di fornire i dati sismici come un servizio web. Nella stazione MEMS il programma sever utilizzato è il software "ringserver" (<https://seiscode.iris.washington.edu/projects/ringserver>) ed il servizio web Seedlink è disponibile sulla porta standard 18000.

Per lo sviluppo delle componenti necessarie per l'acquisizione e l'impacchettamento del segnale sismico è stato usato il linguaggio di programmazione Python (<https://www.python.org/>).

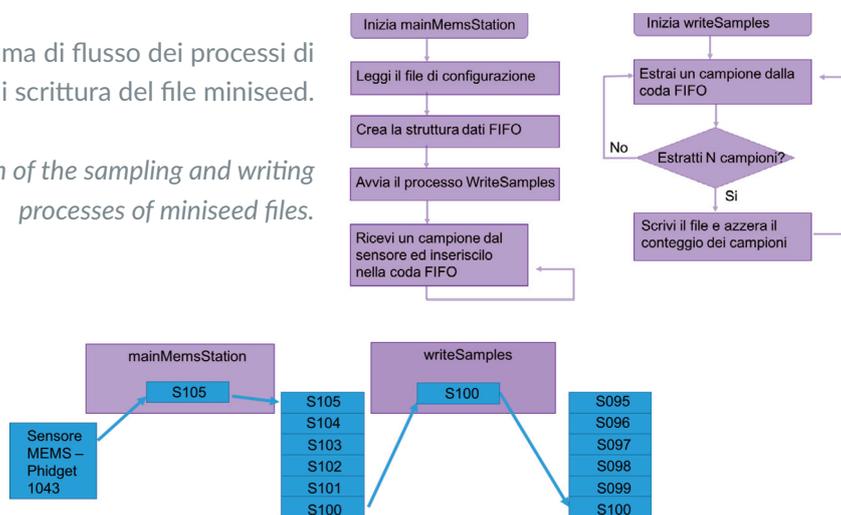
Il programma principale per il funzionamento della stazione è "mainMemsStation.py", che gestisce le due operazioni di base per la trasmissione dei dati: la ricezione del segnale dal sensore e l'associazione ad un preciso intervallo temporale; l'effettiva scrittura dei pacchetti secondo lo standard "miniseed" (<http://ds.iris.edu/ds/nodes/dmc/data/formats/miniseed/>). Per rendere paralleli i due processi di lettura dati e di scrittura file, il programma principale si occupa solamente della lettura e, in fase di partenza, avvia un processo separato chiamato "writeSamples" che ha il compito di ricevere i dati letti dal processo principale e di scrivere i file miniseed quando ha a disposizione il numero di campioni necessario per chiudere il file. Il numero di campioni dipende dalla durata del pacchetto ed è fornito ad entrambi i processi tramite un file di configurazione chiamato "set.conf". Lo scambio di dati tra i due processi avviene tramite una struttura dati condivisa, nello specifico una coda FIFO (First-in First-out) nella quale i dati vengono restituiti secondo lo stesso ordine in cui sono stati scritti. Lo schema a blocchi dei due processi è mostrato nella Figura 5; un esempio di come i campioni vengano letti e scambiati tra i due processi tramite la struttura dati condivisa è presente in Figura 6.

L'istante temporale di riferimento di ogni pacchetto è l'orario di sistema, con la precisione di

millisecondi, in cui il pacchetto è stato ricevuto. L'orario di sistema è costantemente aggiornato tramite un software separato chiamato "gpsDate.py". Questo si interfaccia con il dispositivo GPS presente nella stazione e ogni N secondi (dove N è un parametro fissato prima della fase di avvio del processo) richiede l'orario GPS ed aggiorna quello di sistema.

**Figura 5** Diagramma di flusso dei processi di campionamento e di scrittura del file miniseed.

*Figure 5 Working diagram of the sampling and writing processes of miniseed files.*



**Figura 6** Esempio di utilizzo della coda FIFO per il passaggio di dati dal sensore al processo di scrittura file miniseed.

*Figure 6 Example of the use of FIFO queue managing the data transfer from the sensor to the miniseed writing process.*

L'ultimo applicativo sviluppato in python è stato "deleteOldFiles.py", che semplicemente si occupa di svuotare la cartella dei file miniseed creati dal processo "writeSamples". La cartella deve essere periodicamente svuotata per evitare di riempire eccessivamente la memoria SD del sistema e deteriorare le prestazioni (la memoria presente è sufficiente per pochi giorni di registrazione). Nella versione attuale del software, il processo mantiene nella cartella solo i file relativi alle ultime 24 ore di segnale. Il software "ringserver", inoltre, ha un buffer interno di trasmissione dati (1Gb di default, ma estendibile cambiando gli appositi parametri) che utilizza per garantire l'invio ritardato del segnale quando la connessione non è temporaneamente disponibile.

Per una procedura passo-passo, dalla configurazione completa, ai codici Python implementati, sino all'avvio in acquisizione della stazione MEMS, si rimanda alle Appendici 1-14.

### 3. Analisi della performance della stazione accelerometrica MEMS

La performance della nostra stazione sismica, in termini di capacità di percepire un evento sismico, o meglio registrarlo con un buon rapporto segnale-rumore, è principalmente funzione delle specifiche tecniche del sensore MEMS utilizzato e in particolare della sua banda di frequenza, sensibilità, rumorosità, linearità e fondo scala. In Figura 7 vengono riportate le specifiche dell'accelerometro MEMS utilizzato nella realizzazione della stazione sismica in oggetto, ovvero il modello 1043\_0 (PhidgetSpatial Precision 0/0/3 High Resolution) prodotto dalla ditta canadese "Phidgets".

Tale dispositivo integra a bordo un accelerometro a tre assi di tipo capacitivo e un circuito interno di trasduzione di tipo digitale, per cui le uscite risultano già in un formato digitale e proporzionali alla grandezza misurata. L'accelerometro MEMS è in grado di misurare sia accelerazioni costanti

(quindi utilizzabile anche come gravimetro e inclinometro) che variabili nel tempo (per misure di oscillazioni indotte da fenomeni transitori come un terremoto). È dotato di tre elementi sensibili orientati lungo tre assi ortogonali tra loro per effettuare misure vettoriali delle accelerazioni. Tutti gli accelerometri prodotti dalla Phidgets sono calibrati individualmente attraverso misure assolute di gravità lungo i tre assi; questo garantisce l'elevata accuratezza dei sensori.

### 3.1 Analisi delle specifiche

L'accelerometro integrato nel MEMS modello 1043\_0 può funzionare in due modalità: alta sensibilità (Precision), per accelerazioni comprese nell'intervallo  $\pm 2g$ , e in modalità "Backup", per accelerazioni superiori. La tabella annessa alla Figura 7 mostra le specifiche dell'accelerometro triassiale nelle due modalità di funzionamento secondo quanto riportato nei datasheet forniti dalla Phidgets. Visto che la modalità Backup si attiva solo per accelerazioni difficilmente immaginabili per un terremoto, nella descrizione che segue circa le specifiche riportate nella tabella di cui sopra, si farà riferimento solo a quelle relative alla modalità Precision.

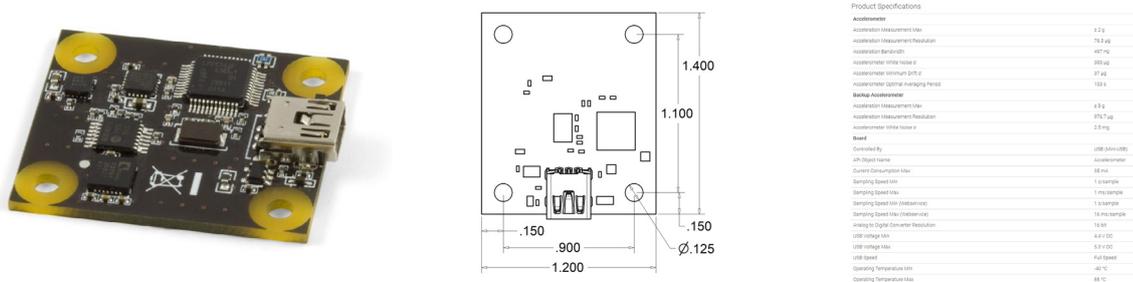


Figura 7 MEMS modello 1043\_0 (PhidgetSpatial Precision 0/0/3 High Resolution) prodotto dalla ditta canadese "Phidgets" (le misure riportate sono in pollici) da: <https://www.phidgets.com/?&prodid=31>

Figure 7 MEMS model 1043\_0 (PhidgetSpatial Precision 0/0/3 High Resolution) produced by the Canadian company "Phidgets" (measures are given in inches) from: <https://www.phidgets.com/?&prodid=31>

Il primo parametro, denominato *Acceleration Measurement Max* definisce il range di accelerazioni misurabili con il sensore per cui si garantisce un segnale in uscita lineare. Tale parametro, così come gli altri riportati, è espresso usando l'unità di misura g (valore convenzionale per g pari a  $9.80665 \text{ m/s}^2$ ) e l'intervallo indicato è chiaramente simmetrico. La massima accelerazione misurabile in modulo è strettamente collegata alla massima deflessione che può avere la massa interna al dispositivo. Si sottolinea che i valori riportati si riferiscono alla massima accelerazione misurabile senza distorsioni dovute alla non linearità. Tale massima accelerazione misurabile non è da intendersi come la massima accelerazione sopportabile dal sensore: quest'ultima, molto maggiore, si riferisce infatti allo stress oltre il quale si ha la rottura fisica del dispositivo. Il valore riportato per la modalità Precision è di  $\pm 2g$ ; questo significa che il sensore è in grado di registrare anche fortissime accelerazioni, come quelle indotte da un terremoto catastrofico, rimanendo in regime lineare. A titolo esemplificativo si ricorda che secondo il modello della pericolosità sismica del territorio italiano (Ordinanza PCM 3519 del 28 aprile 2006, All. 1b), le massime accelerazioni attese sul territorio italiano con una probabilità di eccedenza del 10% in 50 anni (riferite ad un bedrock sismico con velocità delle onde S maggiore di 800 m/s) non superano in modulo 0.3 g. Il secondo parametro riportato è l'*Acceleration Measurement Resolution*. Tale valore esprime

la sensibilità del sensore, ovvero la variazione più piccola di accelerazione che il nostro sensore è in grado di misurare. Il valore riportato è di 76.3  $\mu\text{g}$ , che implica un range dinamico maggiore di 44 dB.

Il terzo parametro riportato è l'*Acceleration Bandwidth* che definisce la banda di frequenze in cui il sistema ha risposta lineare. Il valore riportato è di 497 Hz, ciò significa che il nostro accelerometro è in grado di rispondere linearmente a tutte le frequenze (partendo dalla componente continua, ovvero 0 Hz) inferiori a tale valore. Come ben noto le frequenze massime generate da un terremoto sono in genere dell'ordine delle decine di Hertz. Quelle di interesse per l'ingegneria sismica, ovvero quelle che possono creare maggiori danni agli edifici innescando fenomeni di risonanza nelle strutture, sono comprese in un intervallo ancora più ristretto (circa 0.2 - 20 Hz) e quindi ampiamente contenute nella banda di frequenza dell'accelerometro MEMS qui impiegato.

Il quarto parametro, denominato *Accelerometer White Noise*, esprime la potenza totale del rumore casuale. Tale rumore è auto-generato da tutti i dispositivi elettro-meccanici. Il valore riportato, pari a 280  $\mu\text{g}$ , esprime la deviazione standard del rumore autoprodotta. È chiaro quindi che, qualsiasi accelerazione sismica inferiore a tale valore sarebbe difficilmente rilevabile nei segnali acquisiti dal sensore. Tuttavia, come evidenziato nel sotto-paragrafo, tale valore di deviazione standard risulta dello stesso ordine di grandezza di quello del rumore sismico atteso in aree urbane e quindi per tali ragioni non costituisce un fattore significativamente limitativo.

Altre specifiche di indubbio interesse, riportate nella Tabella di Figura 7, sono chiaramente i consumi, la frequenza di campionamento massima, il numero di bit, l'intervallo di temperatura di funzionamento, ecc. Tutti questi parametri, così come gli altri riportati, risultano ottimi per le finalità a cui è destinata la stazione sismica. In particolare, la possibilità di avere un segnale a 16 bit a una frequenza massima di campionamento di 1000 Hz, rende tale sensore idoneo anche ad un eventuale impiego nel campo della sismica applicata.

### 3.2 Valutazione della performance mediante analisi spettrale e simulazione numerica

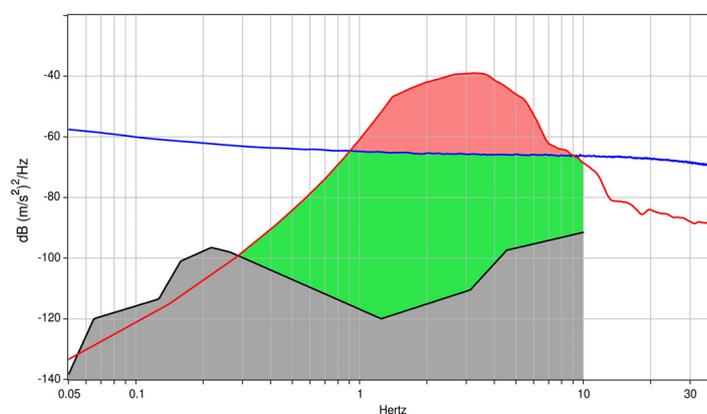
Al fine di valutare la performance del sensore accelerometrico MEMS impiegato nella realizzazione della stazione oggetto del presente lavoro, D'Alessandro et al., [2017] hanno esaminato il rumore autoprodotta da complessivi 10 accelerometri MEMS, eseguendo una attenta analisi in termini di Power Spectral Density (PSD) e Allan Variance. Tale analisi ha permesso una esaustiva caratterizzazione del sensore accelerometrico montato a bordo del device 1044\_0 3/3/3 PhidgetSpatial Precision High Resolution, lo stesso montato a bordo del device 1043 0/0/3 (l'unica differenza è che il 1044 monta a bordo anche un giroscopio e un magnetometro ambedue triassiali). Per gli scopi del presente lavoro, si fa riferimento solo allo PSD medio dell'accelerometro MEMS, in relazione ai modelli di noise sismico e agli spettri di terremoto, rimandando al lavoro di D'Alessandro et al., [2017], per più approfondite analisi.

È stato quindi calcolato il PSD del self-noise dell'accelerometro MEMS 1043, determinato a partire dall'analisi di 72 ore di segnale registrati presso un sito caratterizzato da basso rumore ambientale. Gli spettri di potenza calcolati per le tre componenti sono risultati assai simili; è stato quindi calcolato il loro valore medio riportato in Figura 8. Nell'intervallo di frequenze 0.05-40 Hz, lo spettro di potenza risulta compreso tra circa -58 e -70 dB  $(\text{m/s}^2)^2/\text{Hz}$  e mostra un andamento monotonicamente decrescente in funzione delle frequenze. In Figura 8 inoltre, per un confronto, è riportato anche lo spettro di riferimento superiore di Peterson [1993]. Tale spettro indica la potenza massima attesa sulla componente verticale del rumore sismico in particolari condizioni, ovvero per siti ubicati lontani da centro urbani e caratterizzati da bedrock affiorante.

Il rumore auto-prodotto dal sensore MEMS appare ben più alto rispetto alla curva superiore di

riferimento di Peterson [1993] ma in realtà, per gli scopi che a cui è destinata la suddetta stazione sismica, tale valore risulta più che accettabile. Infatti, tale curva può essere considerata un riferimento vincolante in caso di installazione di una stazione sismica standard, ma non può essere considerato un riferimento nel nostro caso in quanto tali stazioni nascono per la realizzazione di reti sismiche urbane. Come ben noto, i centri urbani e le aree industrializzate sono una fonte importante di rumore sismico (in particolare per frequenze > 1 Hz); l'installazione poi non su bedrock affiorante, porta ad un'ulteriore amplificazione del rumore sismico. Questi due effetti sommati sono in grado di generare un rumore sismico caratterizzato da un PSD in grado di superare la curva superiore di Peterson [1993] di ben 20-30 dB.

Ad ogni modo, dalla Figura 8 è possibile osservare come, in un ampio intervallo di frequenze, lo spettro del terremoto preso in considerazione superi nettamente in potenza quello del rumore autoprodotta del sensore MEMS in oggetto.



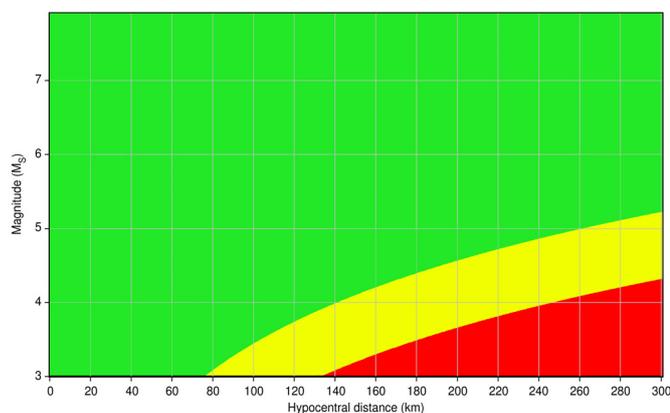
**Figura 8** Spettro di potenza medio del rumore autoprodotta dal dispositivo MEMS 1043\_0 (linea blu) con sovrapposto lo spettro di potenza medio delle componenti orizzontali di un terremoto poco profondo di  $M_L$  4 registrato ad una distanza epicentrale di circa 35 km, da un accelerometro larga banda a bilanciamento di forza di tipo professionale (linea rossa, NS Episensor Kinemetrics). Per comparazione viene anche riportato lo spettro superiore del rumore ambientale (Peterson, [1993], linea nera).

*Figure 8 Mean power spectrum of the MEMS 1043\_0 self-noise (blue line) and the mean power spectrum of the horizontal components of a shallow  $M_L$  4 earthquake recorded at epicentral distance of 35 km by a broadband professional sensor (red line, NS Episensor Kinemetrics). As comparison is also reported the high noise model of seismic background noise (Peterson, [1993], black line).*

Al fine di generalizzare le conclusioni di cui sopra e determinare la capacità della stazione accelerometrica avvertire un terremoto al variare della distanza epicentrale e della magnitudo, sono state realizzate alcune simulazioni facendo uso della ben nota legge di attenuazione proposta da Ambraseys et al. [1996] e Simpson [1996]. Tale legge empirica, ottenuta dall'analisi di 422 terremoti europei con profondità ipocentrale inferiore a 30 km, magnitudo compresa tra 4 e 7.9 e distanza epicentrale compresa tra 0 e 260 km, è stata utilizzata per determinati i valori di PGA (componenti orizzontali) per terremoti di magnitudo ( $M_S$ ) compresa tra 3 e 7.8 e distanza epicentrale compresa tra 0 e 300 km. Tali valori di PGA sono stati confrontati con i valori di deviazione standard del rumore elettronico dell'accelerometro triassiale integrato nel dispositivo MEMS modello 1043\_0.

La figura 9 mostra i valori di magnitudo e di distanza ipocentrale per i quali l'accelerometro

MEMS in oggetto sarebbe ipoteticamente in grado di registrare i valori di PGA sulle componenti orizzontali con un rapporto segnale rumore maggiore di 10. La figura mostra come le PGA generate da eventi di magnitudo 3 potrebbero essere registrate fino a distanze ipocentrali di circa 70 km nell'ipotesi di bedrock affiorante e di circa 130 km nell'ipotesi di suolo soffice affiorante. Le PGA generate da terremoti di magnitudo superiore potrebbero essere registrate anche a distanze ipocentrali nettamente superiori.



**Figura 9** Il grafico mostra, in funzione della magnitudo ( $M_s$ ) e della distanza ipocentrale, la performance di detezione dell'accelerometro integrato nel MEMS 1043\_0, ottenuta attraverso il confronto tra i valori di deviazione standard del rumore autoprodotta dal dispositivo e i valori di PGA attesi sulle componenti orizzontali utilizzando la legge empirica di Ambraseys et al. [1996] e Simpson [1996]. In verde e in giallo le aree dove le PGA attese sulle componenti orizzontali superano di un fattore 10 la deviazione standard del rumore autoprodotta dal dispositivo MEMS, riferiti rispettivamente a siti con bedrock e sedimenti soffici affioranti.

*Figure 9 Plot of earthquake magnitude ( $M_s$ ) and epicentral distance. The earthquake detection performance of the accelerometer integrated onto the MEMS 1043\_0, is obtained with the comparison between the standard deviation values of the self-noise with the PGA values expected on the horizontal component calculated with the empirical relationship provided by Ambraseys et al. [1996] and Simpson [1996]. The green and red areas indicate the conditions in which the expected horizontal PGA are 10 times greater with respect the standard deviation of the MEMS self noise. They are referred to bedrock substrate and soft substrate respectively.*

### 3.3 Analisi evento sismico di S. Ninfa ( $M_L$ 3.1 del 30.10.2018)

A luglio del 2018 è stata installata una stazione MEMS presso il comune di Santa Ninfa (TP). La stazione è stata installata all'interno dell'edificio municipale, presso un vano ubicato al 2° piano (Figura 10). I dati sono acquisiti con continuità presso il server ubicato presso il Laboratorio di Monitoraggio e Metodi Sismici (Laboratorio MEMS), Laboratorio realizzato in convenzione con il Dipartimento di Scienze della Terra e del Mare dell'Università di Palermo.

Il laboratorio MEMS dispone di sistemi hardware e software dedicati alla ricezione e all'archiviazione in tempo reale dei segnali provenienti dalle stazioni accelerometriche. Le unità hardware che compongono il sistema sono, progettate e testate per fornire un funzionamento affidabile e prolungato nel tempo, sono state inserite in un unico rack (Figura 10). Tale rack ospita spazi di archiviazione ridondanti per garantire la continuità del servizio e per una gestione sicura dei dati. Il software di gestione della rete è il SeisComp3 (<https://www.seiscomp3.org/>) sviluppato in Germania dal centro di ricerca in geoscienze (GFZ, Potsdam). Il programma è costituito da vari moduli per l'acquisizione, l'elaborazione, l'analisi interattiva e la distribuzione.

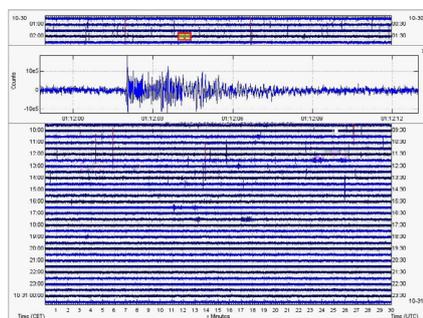
Le stazioni accelerometriche sfruttano il protocollo seedlink per la trasmissione dati; il sistema in funzione al Laboratorio MEMS riceve i file mSEED dalle stazioni e li ricompone in un'unica traccia giornaliera (una traccia per componente) che viene quindi archiviata. Il sistema può anche funzionare da ringserver, e distribuire i dati alla sala Sismica di Roma, o a qualunque altra istituzione.



**Figura 10** Sinistra: Edificio del Comune di S. Ninfa, all'interno del quale (2° piano) è stata installata una stazione accelerometrica MEMS; destra: Sistema di acquisizione e archiviazione dati realizzato presso il Laboratorio MEMS.

*Figure 10 Left. The S. Ninfa Town Hall hosting the MEMS accelerometric station at the 2<sup>nd</sup> floor; right the acquisition system at the MEMS Laboratory.*

La visualizzazione dei dati è affidata al software Swarm, sviluppato dall'Alaska Volcano Observatory (<https://volcanoes.usgs.gov/software/swarm/index.shtml>). Swarm è un'applicazione Java sviluppata per l'analisi e la visualizzazione real-time di forme d'onda sismiche, in grado di connettersi e leggere da diverse sorgenti dati, sia statiche che dinamiche (Earthworm waveservers, IRIS DMCs, file SEED, SAC, ASCII). Swarm è open source, estremamente versatile in quanto include strumenti per l'analisi dati in tempo reale, sia nel dominio temporale che delle frequenze, e una piattaforma per la gestione di mappe dinamiche.



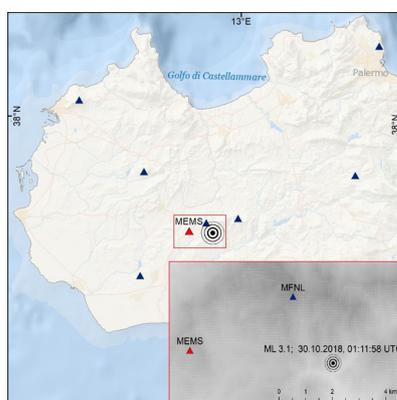
**Figura 11** Screenshot relativo all'helicorder (Swarm) dei segnali registrati dalla stazione accelerometrica di S. Ninfa, in acquisizione presso il Laboratorio MEMS; in evidenza il terremoto di  $M_L$  3.1, di giorno 30.10.2018, ore 01:11:58 UTC (Lat. 37.77, Lon. 12.94, prof. 13 km, <http://cnt.rm.ingv.it/event/21026611?timezone=UTC>).

*Figure 11 Screenshot of the helicorder (Swarm) of the signals recorded by the accelerometric station installed in S. Ninfa and received in the MEMS Laboratory. The box highlights the  $M_L$  3.1 earthquake occurred on October 10th, 2018, h. 01:11:58 UTC (Lat. 37.77, Lon. 12.94, depth. 13 km, <http://cnt.rm.ingv.it/event/21026611?timezone=UTC>).*

La Figura 11 mostra l'helicorder relativo circa 24 ore di segnale della stazione sismica MEMS installata a S. Ninfa (TP). È possibile osservare in evidenza la forma d'onda registrata il 30.10.2018, in occasione di un piccolo evento sismico ( $M_L$  3.1) avvenuto intorno alle ore 1:11 UTC. Per maggiori dettagli sul tale evento sismico, si rimanda al seguente link: <http://cnt.rm.ingv.it/event/21026611?timezone=UTC>.

La Figura 12 mostra l'ubicazione dell'evento sismico, della stazione sismica MEMS e delle stazioni della Rete Sismica Nazionale prossime all'area epicentrale. La stazione MEMS di S. Ninfa, risulta ubicata a circa 5.3 km dall'epicentro, mentre la stazione della Rete Sismica Nazionale più vicina risulta essere quella di Monte Finestrelle (MFNL), ubicata a circa 2.8 km dall'epicentro.

In Figura 13, vengono riportati i segnali di accelerazione sulle 3 componenti, registrati dalla stazione di S. Ninfa, a seguito del terremoto di  $M_L$  3.1 e ricevuti in real-time presso il centro di acquisizione dati del Laboratorio MEMS. La frequenza di campionamento dei segnali è di 250 Hz; i segnali sono stati filtrati in passabanda tra 1 e 35 Hz.



**Figura 12** Mappa con ubicata la stazione MEMS di S. Ninfa (triangolo rosso), in relazione alle stazioni della Rete Sismica Nazionale (triangoli blu) e al terremoto di  $M_L$  3.1 (cerchi concentrici neri), avvenuto il 30.10.2018, ore 01:11:58 UTC (Lat. 37.77, Lon. 12.94, prof. 13 km, <http://cnt.rm.ingv.it/event/21026611?timezone=UTC>).

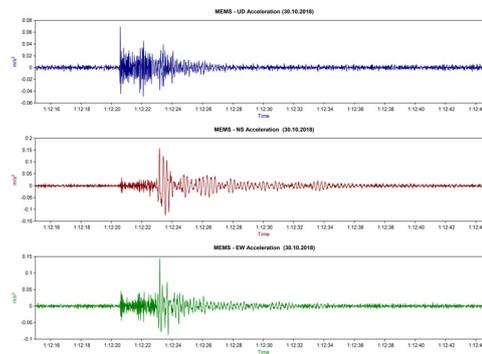
Nel riquadro in basso, in evidenza la stazione MEMS di S. Ninfa, la stazione della Rete Sismica Nazionale di Monte Finestrelle (MFNL) e il terremoto in oggetto.

*Figure 12 Maps showing the MEMS station located in S. Ninfa (red triangle), the stations of the Italian Seismic Network (blue triangles) and the epicenter of the  $M_L$  3.1 earthquake (black circles) occurred on October 10th, 2018, h. 01:11:58 UTC (Lat. 37.77, Lon. 12.94, depth. 13 km, <http://cnt.rm.ingv.it/event/21026611?timezone=UTC>). In the box in the lower right corner a zoom on the S. Ninfa station, the Monte Finestrelle station (MFNL) and the epicenter of the considered earthquake.*

Dalla Figura 13 è possibile osservare come il terremoto di  $M_L$  3.1 sia stato registrato con un buon rapporto segnale-rumore; risultano facilmente riconoscibili l'arrivo di varie fasi sismiche, con differenti valori di ampiezza e frequenza. In Figura 14, le stesse forme d'onda vengono confrontate con quelle registrate dalla stazione sismica di MNFL. La stazione MNFL è equipaggiata di sensore larga-banda (120s) Guralp modello CMG-3EX. Essendo il sensore di MNFL un velocimetro, per rendere i segnali confrontabili (dopo opportuna deconvoluzione per la funzione di trasferimento), i segnali di velocità sono stati convertiti in accelerazione mediante derivazione numerica.

Da confronto dei segnali di Figura 14 è possibile innanzitutto osservare la perfetta sincronia temporale delle principali fasi sismiche; le stazioni si trovano ambedue in prossimità dell'epicentro e hanno distanze epicentrali comparabili (circa 5.3 per la stazione MEMS di S.

Ninfa, circa 2.8 per la stazione MNFL). La differenza tra le principali fasi sismiche individuate è dell'ordine del centesimo di secondo, chiaramente legate alla, seppur minima, differenza tra le distanze epicentrali.



**Figura 13** Segnali sismici (accelerazioni, segnali filtrati passa banda tra 1 e 35 Hz) registrati dalla stazione MEMS di S. Ninfa, relativi al terremoto di  $M_L$  3.1, di giorno 30.10.2018 (larghezza finestre 30 secondi).

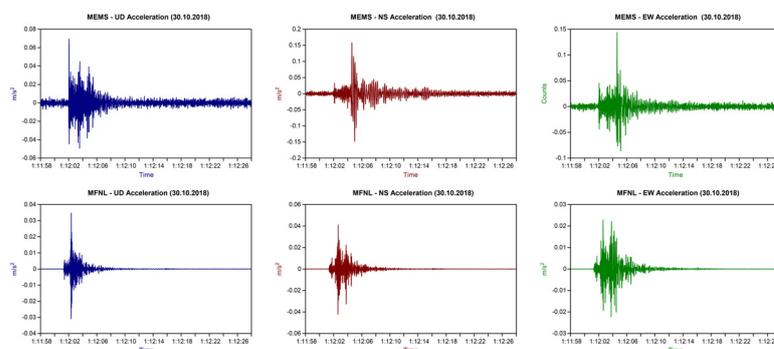
*Figure 13* Seismic signals (band-pass filtered acceleration between 1 and 35 Hz) recorded at the MEMS station located in S.Ninfa after the  $M_L$  3.1 earthquake occurred on October 10th, 2018 (30 seconds time window).

Il confronto tra le componenti verticali, permette di rilevare una differenza di ampiezza tra le forme d'onda. In particolare si osserva una accelerazione massima in valore assoluto (PGA) sulla componente verticale, pari a  $0.069 \text{ m/s}^2$  per la stazione MEMS e di  $0.035 \text{ m/s}^2$  per la stazione MNFL. Tale differenza tra le PGA aumenta sensibilmente se si pone attenzione alle componenti orizzontali. In tale caso abbiamo  $0.159$  e  $0.144 \text{ m/s}^2$ , per le PGA sulle componenti NS e EO rispettivamente della stazione di MEMS di S. Ninfa, e  $0.042$  e  $0.023 \text{ m/s}^2$ , per le PGA sulle componenti NS e EO rispettivamente della stazione di MNFL.

Riassumendo, mentre la PGA della componente verticale registrata dalla stazione MEMS risulta circa il doppio di quella registrata dalla stazione MNFL, le PGA delle componenti orizzontali risultano sicuramente più amplificate, di un fattore, mediato per le due componenti, pari a circa 4.7. Tale effetto di amplificazione, particolarmente evidente sulle componenti orizzontali, è chiaramente legato a fenomeni di amplificazioni locali. Tali amplificazioni del moto del suolo potrebbero essere generate da particolari condizioni geologico/topografiche al sito, ma più verosimilmente da un effetto di risonanza dell'edificio. Tale effetto di risonanza, è plausibile dalla semplice ispezione delle forme d'onda di Figura 13 e del confronto di quelle di Figura 14. È possibile infatti osservare, in particolare in Figura 13, come le componenti orizzontali del moto del suolo siano caratterizzate da una parte iniziale a più alta frequenza e da una parte finale a più bassa frequenza, quasi monocromatica. Tale parte a bassa frequenza si estende per diversi secondi e non è presente nei segnali della stazione di MNFL.

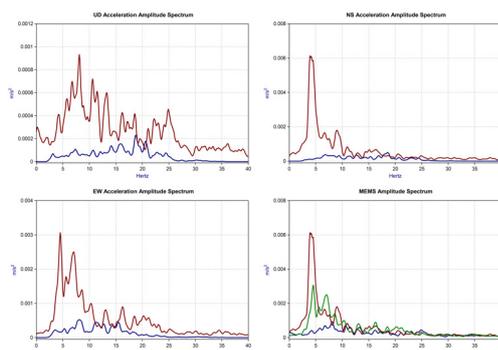
Sono stati quindi determinati gli spettri di ampiezza normalizzati relativi alle forme d'onda di terremoto registrati dalle due stazioni in oggetto. Gli spettri di ampiezza sono stati determinati utilizzando una finestra temporale contenente l'intero evento, con finestrazione di tipo Hanning. La Figura 15 mostra il confronto tra gli spettri di ampiezza, in relazione alle stesse componenti, della stazione MEMS di S. Ninfa e di MNFL. Quello che si evince da questa comparazione è che tale differenza in ampiezza tra i segnali delle due stazioni, è principalmente dovuta probabili fenomeni di risonanza dell'edificio comunale. Tale fenomeno di risonanza, particolarmente evidente sulla componente NS, genera una forte amplificazione in uno stretto intervallo di frequenze centrato intorno a circa 4 Hz. Utilizzando la ben nota formula semplificata proposta

dallo NTC2008 (Norme Tecniche per le Costruzioni - D.M. 14 Gennaio 2008) per il calcolo del periodo di vibrazione  $T$  in funzione dell'altezza della costruzione  $H$  ovvero:  $T = C1 \times H^{(2/3)}$ ,  $C1$  è un coefficiente che dipende dalla tipologia strutturale, possiamo verificare la suddetta ipotesi. Imponendo nella precedente formula  $C1 = 0.075$  (valore valido per costruzioni con struttura a telaio in calcestruzzo armato), otterremo un valore di  $H$  pari a circa 6 m, perfettamente compatibile con l'altezza l'edificio del Comune di S. Ninfa e con l'installazione della nostra stazione in corrispondenza del 2° piano.



**Figura 14** Segnali sismici (accelerazioni, dati non filtrati) registrati dalla stazione MEMS di S. Ninfa e dalla stazione della Rete Sismica Nazionale di Monte Finestrelle (sigla internazionale MFNL), relativi al terremoto di  $M_L$  3.1, di giorno 30.10.2018 (larghezze finestre 30 secondi).

*Figure 14* Seismic signals (unfiltered acceleration) recorded at the MEMS station located in S.Ninfa and from the Monte Finestrelle station (International code MFNL) belonging to the Italian Seismic Network, after the  $M_L$  3.1 earthquake occurred on October 10th, 2018 (30 seconds time window).



**Figura 15** Spettri di ampiezza dei segnali sismici (accelerazioni, dati non filtrati) registrati dalla stazione MEMS di S. Ninfa (linea rossa) e dalla stazione della Rete Sismica Nazionale di Monte Finestrelle (sigla internazionale MFNL; linea blu), relativi al terremoto di  $M_L$  3.1, di giorno 30.10.2018 (<http://cnt.rm.ingv.it/event/21026611?timezone=UTC>).

*Figure 15* Amplitude spectra of the seismic signals (unfiltered accelerations) recorded at the MEMS station located in S. Ninfa (red line) and from the Monte Finestrelle station (International code MFNL; blue line) belonging to the Italian Seismic Network, after the  $M_L$  3.1 earthquake occurred on October 10th, 2018, (<http://cnt.rm.ingv.it/event/21026611?timezone=UTC>).

## 4. Conclusioni e sviluppi futuri

Nell'ambito del progetto MEMS è stata progettata e realizzata una stazione accelerometrica basata su tecnologia MEMS, estremamente versatile e a bassissimo costo. La stazione in oggetto è stata implementata per rispondere a delle specifiche esigenze del progetto MEMS ovvero la realizzazione di reti sismiche urbane. Gli OSU o Osservatori Sismici Urbani, sono una concreta risposta alla sempre più impellente necessità di creare strumenti di riduzione del rischio sismico. La stazione in oggetto, descritta in estremo dettaglio nelle sue componenti hardware e software, risulta oggi idonea a registrare fedelmente terremoti di media-forte magnitudo, restituendo in tempo reale osservazioni puntuali e precise circa lo scuotimento risentito a seguito di un terremoto.

La stazione MEMS è in grado di registrare, sincronizzare, pre-elaborare e trasmettere dati in tempo reale (anche ad altissima frequenza). L'accelerometro triassiale MEMS di cui è stata equipaggiata, garantisce la capacità di registrare fedelmente terremoti di media-forte magnitudo in tutto il range di frequenze di interesse della sismologia. Per le sue caratteristiche, la stazione si presta non solo nella realizzazione di OSU, ma anche per il monitoraggio dinamico dello stato di salute di strutture e l'impiego in emergenza [Moretti et al., 2016]. Nella sua attuale configurazione, la stazione ha un costo complessivo di circa 200 €.

Le peculiarità degli accelerometri MEMS, ovvero le ridottissime dimensioni, la capacità di funzionare in qualsiasi posizione (nessuna esigenza di livellamento) e la possibilità di essere impiegati congiuntamente con un giroscopio e una bussola elettronica per la sua definizione univoca dell'orientamento spaziale [D'Alessandro et al., 2014c], rendono tali strumenti estremamente versatili e impiegabili anche in strumentazione per ambienti estremi come sensori da fondo mare [D'Alessandro et al., 2009; D'Anna et al., 2009; Mangano et al., 2011].

Tale stazione, così come è stata concepita, può essere facilmente customizzata per specifiche esigenze, semplicemente sostituendo le varie parti che la compongono, con quelle ritenute più idonee allo scopo. È così possibile pensare di sfruttare tale stazione anche per il monitoraggio di altri parametri, semplicemente sostituendo o aggiungendo nuovi sensori. La stazione MEMS, oggi progettata per misurare forti accelerazioni, in un prossimo futuro potrebbe diventare una stazione per il monitoraggio di pressione, temperatura, pH, campo magnetico, ecc., semplicemente cambiando il sensore montato a bordo. La presenza di una SBC permette inoltre la facile implementazione di algoritmi dedicati all'analisi in tempo reale dei segnali registrati, in uno schema di risorse di elaborazione distribuite ovvero on-site.

Gli sviluppi futuri prevedono l'integrazione di un convertitore Analogico-Digitale per l'acquisizione di segnali generati da sensori analogici e l'accoppiamento anche con un sensore velocimetrico a banda allargata [Vitale et al., 2018].

## Riconoscimenti

Le attività descritte nel presente lavoro sono state realizzate nell'ambito del progetto MEMS - Monitoring of Earthquakes through MEMS Sensors: project for the realization of a prototype of real-time urban seismic network based on MEMS technology (36 Mesi); progetto finanziato dal MIUR (Ministero dell'Istruzione, dell'Università e della Ricerca) nell'ambito del programma SIR (Scientific Independence of young Researchers), Codice Progetto RBSI14HL4E, Responsabile Scientifico (Principal Investigator) Antonino D'Alessandro, Decreto Direttoriale 25 giugno 2015 n. 1350.

Le attività sono state inoltre svolte presso il Laboratorio MEMS (Monitoraggio e Metodi Sismici) realizzato grazie alla Convenzione tra l'Osservatorio Nazionale Terremoti dell'Istituto Nazionale di Geofisica e Vulcanologia e il Dipartimento di Scienze della Terra e del Mare

dell'Università degli Studi di Palermo (Prot. Gen. INGV N. 15729 del 23.11.2017; Prot. DiSTeM N. 1736 del 13.11.2017); responsabili del Laboratorio Antonino D'Alessandro e Raffaele Martorana.

## Ringraziamenti

Si ringrazia il Comune di S. Ninfa per la concessione di ospitalità della stazione MEMS e per il supporto tecnico-logistico. Si ringrazia la Sezione INGV di Palermo per la disponibilità all'utilizzo di laboratori e officina meccanica.

## Bibliografia

- Ambraseys N.N., Simpson K.A. & Bommer J.J., (1996). *Prediction of horizontal response spectra in Europe*. Earthquake Engineering and Structural Dynamics, 25(4), 371-400.
- Clayton R.W., Heaton T., Chandy M., Krause A., Kohler M., Bunn J., Guy R., Olson M., Faulkner, M., Cheng M., Strand L., Chandy R., Obenshain D., Liu A., Aivazis M., (2011). *Community Seismic Network*. Annals of Geophysics, 54, 6, 2011; doi: 10.4401/ag-5269.
- Cochran E.S., Lawrence J.F., Christensen C., Jakka R.S., (2009). *The Quake-Catcher Network: Citizen Science Expanding Seismic Horizons*. Seismological Research Letters, Vol. 80, N. 1, doi: 10.1785/gssrl.80.1.26.
- Cochran E.S., Lawrence J.F., Kaiser A., Fry B., Chung A., Christensen C., (2012). *Comparison between low-cost and traditional MEMS accelerometers: a case study from the M 7.1 Darfield, New Zealand, aftershock deployment*. Annals of Geophysics, vol. 54, n. 6, pp. 728-737, doi: 10.4401/ag-5268.
- Chung A.I., Neighbors C., Belmonte A., Miller M., Sepulveda H.H., Christensen C., Jakka R.S., Cochran E.S., Lawrence J.F., (2011). *The Quake-Catcher Network Rapid Aftershock Mobilization Program Following the 2010 M 8.8 Maule, Chile Earthquake*. Seismological Research Letters, vol. 82, n. 4, doi: 10.1785/gssrl.82.4.526.
- Chung A.I., Cochran E.S., Kaiser A.E., Christensen C.M., Yildirim B. & Lawrence J.F., (2015). *Improved rapid magnitude estimation for a community-based, low-cost MEMS accelerometer network*. Bulletin of the Seismological Society of America, Vol. 105, No. 3, pp. 1314-1323, June 2015, doi:10.1785/0120140232.
- D'Alessandro A., D'Anna, G., Luzio, D., Mangano, G., (2009). *The INGV's new OBS/H: analysis of the signals recorded at the Marsili submarine volcano*. Journal of Volcanology and Geothermal Research, vol. 183, n. 1-2, pp. 17-29, ISSN: 03770273, doi: 10.1016/j.jvolgeores.2009.02.008.
- D'Alessandro A., D'Anna G., (2013). *Suitability of low cost 3 axes MEMS accelerometer in strong motion seismology: tests on the LIS331DLH (iPhone) accelerometer*. Bulletin of the Seismological Society of America, vol. 103, n. 5, pp. 2906-2913.
- D'Alessandro A., (2014a). *Monitoring of Earthquakes through MEMS Sensors*. Current Science, vol 107, 733-734.
- D'Alessandro A., Luzio D., D'Anna G., (2014b). *Urban MEMS based seismic network for post-earthquakes rapid disaster assessment*. Advances in Geosciences, vol. 40, pp. 1-9, ISSN: 16807340, doi: 10.5194/adgeo-40-1-2014.
- D'Alessandro A., D'Anna G., (2014c). *Retrieval of Ocean Bottom and Downhole Seismic sensors orientation using integrated MEMS gyroscope and direct rotation measurements*. Advances in Geosciences, vol. 40, pp. 11-17, ISSN: 16807340, doi: 10.5194/adgeo-40-11-2014.
- D'Alessandro A., (2016). *Tiny accelerometers create Europe's first urban seismic network*. EOS, vol. 97, pp. 16-20, ISSN: 00963941, doi: 10.1029/2016EO048403.
- D'Alessandro A., Vitale G., Scudero S., D'Anna R., Costanza A., Fagiolini A., Greco L., (2017a). *Characterization of MEMS accelerometer self-noise by means of PSD and Allan Variance analysis*. 7<sup>th</sup>

- International Workshop on Advances in Sensors and Interfaces - Conference Paper, IWASI 2017, Vieste, Italy, 15-16 June 2017, pp. 159-164, ISBN: 978-150906706-0, doi: 10.1109/IWASI.2017.7974238.
- D'Alessandro A., D'Anna R., Greco L., Passafiume G., Scudero S., Speciale S., Vitale G., (2018a). *Monitoring Earthquake through MEMS Sensors (MEMS project) in the town of Acireale (Italy)*. 5th IEEE International Symposium on Inertial Sensors and Systems, Conference Paper, INERTIAL 2018; Como, Italy; 26 - 29 March 2018, ISBN: 978-153860894-4, doi: 10.1109/ISISS.2018.8358143.
- D'Alessandro A., Vitale G., Scudero S., D'Anna R., Passafiume G., Greco L., Speciale S., Patanè D., Torrisi O., Di Prima S., Magiagli S., Tusa G., (2018b). *Real-time urban seismic network and structural monitoring by means of accelerometric sensors: Application to the historic buildings of Catania (Italy)*. IEEE International Conference on Environmental Engineering, Conference Paper, EE 2018; Milan; Italy; 12 - 14 March 2018; pp 1-5, ISBN: 978-153864182-8, doi: 10.1109/EE1.2018.8385254.
- D'Alessandro A., Scudero S., Vitale G., (2019). *A Review of the Capacitive MEMS for Seismology*. Sensors, vol. 19, n. 14, pp 1-22, EISSN 1424-8220, doi: 10.3390/s19143093.
- D'Anna G., Mangano G., D'Alessandro A., D'Anna R., Passafiume G., Speciale S., Amato A., (2009). *Il nuovo OBS/H dell'INGV*. Quaderni di Geofisica INGV, n. 65, pp. 1-26, ISSN: 15902595.
- Eddy D.S. and Sparks D.R., (1998). *Application of MEMS technology in automotive sensors and actuators*. Proceedings of the IEEE 86.8 (1998): 1747-1755.
- Evans J.R., Allen R.M., Chung A.I., Cochran E.S., Guy R., Hellweg M., Lawrence J. F., (2014). *Performance of Several Low-Cost Accelerometers*. Seismological Research Letters, Vol. 85, N. 1, doi: 10.1785/0220130091.
- Hoque R., Hassan S., Sadaf M.A., Galib A. & Karim T.F., (2015). *Earthquake monitoring and warning system*. Proceedings of 2015 3rd IEEE International Conference on Advances in Electrical Engineering (ICAEE), 17-19 December, 2015, Dhaka, Bangladesh. 109-112.
- Gasparini P., Manfredi G., Zschau J., (2007). *Earthquake Early Warning Ststems*. Springer, pp.1-349, ISBN 978-3-540-72240-3.
- Havskov J., Alguacil G., (2004). *Instrumentation in earthquake seismology*. Springer, Vol. 358. Dordrecht.
- Horiuchi S., Horiuchi Y., Yamamoto S., Nakamura H., Wu C., Rydelek P.A., Kachi M., (2009). *Home seismometer for earthquake early warning*. Geophysical Research Letters, 36(5): L00B04.
- Jones T.B., Nenadic N.G., (2013). *Electromechanics and MEMS*. Cambridge University Press, pp. 559, ISBN 978-0-521-76483-4.
- Kim Y., Kang T. S. & Rhie J., (2017). *Development and Application of a Real-Time Warning System Based on a MEMS Seismic Network and Response Procedure for the Day of the National College Entrance Examination in South Korea*. Seismological Research Letters, 88(5), 1322-1326.
- Kohler M.D., Heaton T.H., Cheng M.H., (2013). *The community seismic network and quake-catcher network: enabling structural health monitoring through instrumentation by community participants*. Proc. SPIE 8692, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2013, 86923X (April 19, 2013), doi:10.1117/12.2010306; <http://dx.doi.org/10.1117/12.2010306>.
- Lawrence J.F., Cochran E.S., Chung A., Kaiser A., Christensen C.M., Allen R., Baker J.W., Fry B., Heaton T., Kilb D., Kohler M.D., Taufer M., (2014). *Rapid Earthquake Characterization Using MEMS Accelerometers and Volunteer Hosts Following the M 7.2 Darfield, New Zealand, Earthquake*. Bulletin of the Seismological Society of America, vol. 104, n. 1, pp. 184-192.
- Mangano G., D'Alessandro A., D'Anna G., (2011). *Long-term underwater monitoring of seismic areas: design of an Ocean Bottom Seismometer with Hydrophone and its performance evaluation*. OCEANS 2011 IEEE - Conference Paper, 6-9 June, Santander, Spain, pp. 1-9, ISBN: 978-1-4577-0086-6, doi: 10.1109/Oceans-Spain.2011.6003609.
- Maluf N., Kirt W., (2004). *Introduction to microelectromechanical systems engineering*. Artech House.
- Merchant J., (2009). *MEMS Applications in Seismology*. Presented at the Seismic Instrumentation Technology Symposium, Palm Springs.
- Moretti M., D'Alessandro A. et al., (2016). *Sismiko: emergency network deployment and data sharing for the*

- 2016 *Amatrice seismic sequence*. *Annals of Geophysics*, vol. 59, n. 5, pp. 1-8, ISSN: 15935213, doi: 10.4401/ag-7212.
- Peterson J., (1993). *Observation and modelling of background seismic noise*. U.S. Geol. Surv. Open-File Rept., Albuquerque, New Mexico, 93-322.
- Pozzi M., Zonta D., Trapani D., Athanasopoulos N., Amditis A.J., Bimpas M., Garetos A., Stratakos Y.E. & Ulieru D., (2011). *MEMS-based sensors for post-earthquake damage assessment*. In *Journal of Physics: Conference Series* 305, 012100.
- Simpson K.A. (1996). *Attenuation of strong ground-motion incorporating near-surface foundation conditions*. Ph.D. thesis, University of London.
- Siino M., D'Alessandro A., Adelfio G., Scudero S., Chiodi M., (2018). *Multiscale processes to describe the eastern Sicily seismic Sequences*. *Annals of Geophysics*, vol. 61, n. 2, SE228, pp. 1-17, ISSN: 15935213, doi: 10.441/ag-7711.
- Scudero S., D'Alessandro A., Greco L., Vitale G., (2018). *MEMS technology in seismology: A short review*. IEEE International Conference on Environmental Engineering, Conference Paper, EE 2018; Milan, Italy; 12 - 14 March 2018; pp 1-5 ISBN: 978-153864182-8, doi: 10.1109/EE1.2018.8385252.
- Vitale G., Greco L., D'Alessandro A., Scudero S., (2018). *Bandwidth extension of a 4.5 Hz geophone for seismic monitoring purpose*. IEEE International Conference on Environmental Engineering, Conference Paper, EE 2018; Milan; Italy; 12 - 14 March 2018; pp 1-5 ISBN: 978-153864182-8, doi: 10.1109/EE1.2018.8385253.
- Zheng H., Shi G., Zeng T. & Li B., (2011). *Wireless earthquake alarm design based on MEMS accelerometer*. In: *Consumer Electronics, Communications and Networks (CECNet), 2011 International Conference on* (pp. 5481-5484). IEEE.

# APPENDICE

## Appendice 1: Primo avvio

Il materiale necessario per iniziare ad assemblare la nostra stazione sismica a basso costo è:

- Raspberry Pi 3 B;
- scheda microSD da almeno 4 Gb (consigliata 16 Gb)
- alimentatore microUSB 5V, 2.5A;
- power-bank come UPS (non indispensabile);
- schermo e cavo HDMI;
- tastiera, mouse;
- connessione ad internet o via Ethernet o WiFi, possibilmente libera senza Firewall;

La procedura da eseguire su un pc d'appoggio Windows è la seguente (su MAC o Linux vi sono procedure da terminale che si trovano online).

- Scaricare da Internet il programma SD-Formatter;
- Scaricare da Internet il programma Win32DiskImage;
- Scaricare da Internet il programma Putty SSH;
- Scaricare da Internet il programma di Desktop remoto VCN;
- Effettuare l'installazione dei programmi precedentemente scaricati.
- Scaricare dal sito ufficiale della Raspberry Pi Foundation il file NOBS per installazione off-line;
- Effettuare se necessario l'estrazione dell'immagine ISO;
- Inserire la microSD nel pc, lanciare l'applicativo SD-Formatter ed eseguire la formattazione della scheda tramite interfaccia utente, al termine potete chiudere il programma;
- Avviare Win32DiskImage, selezionare tramite l'icona browser il percorso in cui si trova l'immagine ISO del sistema operativo che si vuole installare sulla SBC. Da notare che ora sarà possibile cliccare sul pulsante scrivi. Quindi premere su scrivi e alla fine della procedura avremo il sistema operativo scritto sulla nostra microSD pronto per il primo avvio.
- Inserire la microSD nell'apposto alloggiamento, collegare il monitor HDMI, mouse, tastiera ed alimentatore;
- Accendere il display HDMI e successivamente inserire nella presa di rete l'alimentatore del Raspberry.
- Inizialmente apparirà una schermata quadrata color arcobaleno, successivamente si inizierà a vedere il boot da terminale ed in fine apparirà il Desktop utente. Se necessario le credenziali sono: name=pi e password=raspberry.

Terminate le procedure sopra elencate l'installazione del sistema operativo è terminata. Ora è possibile lavorare direttamente con la SBC come sul proprio PC. Per i più esperti o per chi vuole utilizzare il dispositivo da terminale remoto, si ricorda la possibilità di utilizzare Putty SSH o Desktop remoto (è necessario aprire un terminale e digitare "sudo raspi-config"). Si aprirà una finestra rudimentale dove è possibile tramite tastiera effettuare le operazioni di settaggio di sistema. La voce di nostro interesse è "hardware setting", selezionando questa voce si aprirà un ulteriore menu in cui appaiono diverse periferiche o opzioni. Andando su quella di nostro interesse "Server SSH", attiviamola premendo invio; verrà chiesto se vogliamo attivare il server SSH: nuovamente confermiamo con un invio. Dopo qualche istante sarà possibile effettuare la connessione SSH; è consigliabile riavviare il sistema. Stessa procedura per il servizio VNC.

## Appendice 2: Putty SSH e VNC

L'informazione necessaria per l'accesso in remoto al dispositivo e l'attivazione dei servizi richiesti, è il suo IP in rete. Per reperire l'indirizzo IP del proprio dispositivo basta aprire un terminale e digitare "ifconfig".

Il risultato sarà una schermata riassuntiva dei parametri di rete. Qualora il Raspberry fosse collegato alla rete tramite il cavo ethernet, le informazioni che ci interessano sono quelle relative a "eth0"; in particolare nella serie di informazioni visualizzate, è di nostro interesse l'indirizzo IP preceduto dal nome "inet". Qualora il Raspberry fosse collegato alla rete tramite WiFi, le informazioni a cui fare riferimento sono "Indirizzo IPv4" della "scheda LAN wireless Wi-Fi".

È possibile utilizzare il Raspberry Pi tramite Putty SSH; ciò permette di operare direttamente da terminale remoto.

Aprire Putty SSH inserire nelle apposite caselle IP del raspberry, nome utente, password e premere connetti;

Se è tutto corretto si aprirà il terminale remoto in una finestra, altrimenti ripetere l'operazione precedente prestando più attenzione, spesso si incorre in errori di battitura;

È possibile utilizzare il Raspberry Pi tramite VNC; ciò permette di operare direttamente da Desktop remoto.

Aprire VNC inserire nell'appositi caselle IP raspberry, nome utente, password e premere connetti;

Se è tutto corretto, si aprirà in una finestra il Desktop Remoto, altrimenti ripetere l'operazione precedente prestando più attenzione, stando attenti a non incorrere in errori di battitura;

### Appendice 3: Replicare il nostro prototipo

Una domanda lecita da effettuarsi prima di imbarcarsi in uno sviluppo è “come posso replicarne velocemente 1, 10, 100, 1000?”.

La risposta è semplice, Win32DiskImage non effettua soltanto scritture su microSD, ne può creare anche una copia, un’immagine ISO di ciò che contiene, e successivamente riversare la ISO personalizzata su altre microSD. Per risparmiare tempo, effettivamente sarebbe necessario che la microSD madre sia la più piccole possibile (4 Gb). Ecco come procedere:

Spegnere il Raspberry Pi, scollegare tutto, rimuovere la scheda microSD e inserirla al computer d'appoggio;

Avviare Win32DiskImage, dopo di che cliccare l'icona del browser, puntare la cartella dove voler salvare l'immagine ISO e inserire il nome file nella casella di testo in basso, premere “apri”;

Da notare che il tasto scrivi non può essere cliccato in quanto il programma non rileva nessun file con quel nome nella cartella e capisce che si sta per effettuare una lettura;

Premere il tasto “leggi” e aspettare che l'operazione termini; alla fine troveremo la nostra ISO personale nella directory puntata con il nome inserito;

Per copiare il nostro prototipo ripetere la procedura di primo avvio utilizzando l'immagine ISO appena creata.

Piccolo riepilogo: ora si è in grado di effettuare il primo avvio, lavorare con il Desktop, Desktop remoto, server SSH e sappiamo come duplicare i nostri progetti su n-macchine.

#### Appendice 4: Installazione ring-server di IRIS

Il software "ring-server" di IRIS DMC è in grado di eseguire una scansione continua di una o più directory per i record di dati miniSEED e di servirli come flussi di dati in tempo reale utilizzando il protocollo SeedLink. Questo software consente a un centro di raccolta dati di creare dati miniSEED per ridistribuire i propri dati in tempo reale. Eseguitabile solo in Linux, Solaris o Mac OSX è interamente scritto in C. Tramite il file di configurazione è possibile impostare il software ring-server come processo per creare record di dati Mini-SEED da 512 byte.

È possibile ottenere il codice sorgente di ring-server dal seguente link IRIS:

<https://seiscode.iris.washington.edu/projects/ringserver/files>

#### Istruzioni per l'installazione ed avvio di ring-server:

Aprire un terminale e puntare la directory "Download"

```
cd
```

```
cd Downloads/
```

Scaricare il pacchetto ring-server da compilare dal sito della IRIS

```
wget https://github.com/iris-edu/ringserver/archive/v2017.052.tar.gz
```

Scompattare il pacchetto appena scaricato

```
tar zxvf v2017.052.tar.gz
```

Entrare nella cartella scompattata

```
cd ringserver-2017.052/
```

Effettuare la compilazione del ring-server (richiederà qualche minuto)

```
sudo make
```

Creare una cartella "ringserver" nella directory principale dell'utente

```
cd
```

```
sudo mkdir ringserver
```

Spostare all'interno della cartella appena creata l'eseguibile del ring-server:

```
sudo mv /home/pi/Downloads/ringserver-2017.052/ringserver /home/pi/ringserver
```

Creare le cartelle "data" e al suo interno "miniseed" (all'interno di miniseed, verranno scritti i pacchetti miniSEED da trasmettere):

```
cd
```

```
sudo mkdir data
```

```
cd data/
```

```
sudo mkdir miniseed
```

Puntare la cartella ringserver e creare le cartelle "ring", "tlog"

```
cd
```

```
cd ringserver/
```

```
sudo mkdir ring
```

```
sudo mkdir tlog
```

Creare il file di configurazione del ringserver

```
sudo touch ring.conf
sudo echo 'RingDirectory ring
DataLinkPort 16000
SeedLinkPort 18000
ServerID "XX Seismic Network"
TransferLogDirectory tlog
TransferLogRX 0
MSeedScan /home/pi/data/miniseed/ StateFile=scan.state InitCurrentState=y' >
/home/pi/ringserver/ring.conf
```

Nel caso questo ultimo passaggio non vada a buon fine, creare manualmente un file “ring.config” e copiare al suo interno le seguenti stringhe:

```
RingDirectory ring
DataLinkPort 16000
SeedLinkPort 18000
ServerID "XX Seismic Network"
TransferLogDirectory tlog
TransferLogRX 0
MSeedScan /home/pi/data/miniseed/ StateFile=scan.state InitCurrentState=y
```

### **Eeguire il ring-server**

Ring-server può essere avviato da terminale come un normale processo.

```
Dare i permessi di esecuzione e puntare la cartella del ring-server
cd
sudo chmod 777 ringserver
cd ringserver/
```

```
Avviare il ring-server
./ringserver ring.conf> ring.log 2>& 1
```

Il seguente comando oltre a eseguire il server dichiara di leggere il file di configurazione “ring.conf” e di creare un file di stato “ring.log” all’interno della cartella.

Quando il server di segnalazione riceve un segnale TERM o INT (cioè Control-C) si spegnerà in modo pulito, salvandolo allo stato corrente di buffer e stato di scansione.

## Appendice 5: Installazione ObsPy

Il passo successivo consiste nel salvare i dati acquisiti. L'opzione migliore per questo è l'utilizzo di un formato utilizzato da molti sismometri il miniSeed. Il formato miniSeed da un'intestazione (tipica nei metadati) seguita dal segnale sotto formato compresso. Per utilizzare il formato MiniSeed è opportuno utilizzare librerie rodate e testate negli anni, come la libreria in linguaggio python ObsPy. L'unico prerequisito è avere una connessione a Internet sul Raspberry Pi. Aprire un terminale ed eseguire i seguenti comandi:

Aprire il file di sistema "sources.list" utilizzato dal sistema per effettuare gli aggiornamenti

```
sudo nano /etc/apt/sources.list
```

Aggiungere alla fine di questo file il repository alle librerie Obspy

```
deb http://deb.obspy.org wheezy main
```

Salvare il file appena modificato con "Control-O + Invio + Invio" ed uscire con "Control-X + Invio";

Scaricare le chiavi del server che verrà interrogato per effettuare l'installazione della libreria ObsPy e aggiungerle nella lista delle chiavi di accesso

```
wget --quiet -O -
```

```
https://raw.githubusercontent.com/obspy/obspy/master/misc/debian/public.key
```

```
sudo apt-key add -
```

Effettuare l'aggiornamento del sistema e l'installazione della libreria

```
sudo apt-get update -y
```

```
sudo apt-get install python-obspy -y
```

## Appendice 6: installazione driver Phidgets

Aprire un terminale, ed eseguire i seguenti comandi:

```
Installare i driver libusb su RaspberryPy  
cd  
sudo apt-get install libusb-1.0-0-dev -y
```

Scaricare le chiavi del server che verrà interrogato per effettuare l'installazione della libreria ObsPy e aggiungerle nella lista delle chiavi di accesso:

```
sudo wget -qO- http://www.phidgets.com/gpgkey/pubring.gpg  
sudo apt-key add -
```

Creare il file "phidgets.list" per inserire l'indirizzo del server dove scaricare i driver e librerie  
sudo echo 'deb http://www.phidgets.com/debian jessie main' > /etc/apt/sources.list.d/phidgets.list

Effettuare l'aggiornamento del sistema e l'installazione dei driver phidgets  
sudo apt-get update -y  
sudo apt-get install libphidget22 libphidget22-dev phidget22networkserver libphidget22java  
phidget22admin -y

Puntare la cartella "Download" ed effettuare il download delle librerie Python  
cd  
cd Download  
wget https://www.phidgets.com/downloads/phidget22/libraries/any/Phidget22Python.zip

Installare unzip e scompattare le librerie Python  
sudo apt-get install zip unzip -y  
sudo unzip Phidget22Python.zip

Puntare la cartella scompattata ed effettuare l'installazione delle librerie Python  
cd Phidget22Python  
sudo python setup.py install

## Appendice 7: Installazione apache web server

Per effettuare un'interfaccia grafica della stazione sismica è stato scelto di creare una piccola pagina web in cui è possibile settare alcuni parametri d'interesse del campionamento. L'installazione avviene tramite terminale eseguendo i seguenti comandi.

Installazione di apache

```
sudo apt-get install apache2 -y
```

Installazione del supporto php di apache

```
sudo apt-get install php libapache2-mod-php -y
```

## Appendice 8: gpsDate.py

Il processo gpsDate.py ha il compito di tenere aggiornato l'orario di sistema. Nel codice è presente un parametro fisso chiamato "refreshTime", in questo caso impostato a 3600 secondi, che impone il tempo che deva passare tra un aggiornamento dell'orario ed il successivo.

La prima parte del codice si occupa del corretto funzionamento del dispositivo GPS tramite le librerie "Phidget22". Se il dispositivo non è disponibile, il codice non aggiorna il sistema ma cerca di ristabilire la connessione con esso dopo dieci secondi.

La seconda parte si occupa dell'effettivo aggiornamento e di fermare il processo per "refreshTime" secondi.

```
#!/usr/bin/env python

import os
import sys
import time

from Phidget22.Devices.GPS import *
from Phidget22.PhidgetException import *
from Phidget22.Phidget import *
from Phidget22.Net import *
from phidgetAttrlib import *

os.nice(-20)

refreshTime = 60*60

while True:

    try:
        gps = GPS()
    except RuntimeError as e:
        time.sleep(10)
        continue

    try:
        gps.setOnAttachHandler(GPSAttached)
        gps.setOnDetachHandler(GPSDetached)
        gps.openWaitForAttachment(5000)
    except PhidgetException as e:
        print(e)
        time.sleep(10)
        continue

    try:
        GPSTime = gps.getTime()
        GPSDate = gps.getDate()
        year = GPSDate.tm_year
        day = GPSDate.tm_mday
        month = GPSDate.tm_mon
        hour = GPSTime.tm_hour
```

```
minute = GPSTime.tm_min
second = GPSTime.tm_sec
microsecond = GPSTime.tm_ms
print("date -s \"%2d/%2d/%4d %2d:%2d:%2d\" % (month,day,year,hour,minute,second))
      os.system("sudo date --utc --set=\"%2d/%2d/%4d %2d:%2d:%2d.%3d\" %
(month,day,year,hour,minute,second,microsecond))
      time.sleep(refreshTime)

except PhidgetException as e:

    print(e)
    time.sleep(10)
    continue

try:

    gps.close()
except PhidgetException as e:
    time.sleep(10)
    continue

exit(0)
```

### Appendice 9: deleteOldFiles.py

Il processo "deleteOldFile.py" cancella tutti i file che hanno una data di creazione più vecchia di un numero specificato di secondi. In questo caso cancella i file più vecchi di 60\*60\*24 secondi.

```
#!/usr/bin/env python

import os
import sys
import time
from stat import *

os.nice(-20)
dataPath = '/home/pi/data/miniseed'
deadline = 60*60*24

while True:

    try:

        now = time.time()
        filelist = os.listdir(dataPath)

        for seed_file in filelist:
            st = os.stat(os.path.join(dataPath,seed_file))

            if st[ST_MTIME] < (now-deadline):
                os.remove(os.path.join(dataPath,seed_file))

            time.sleep(deadline/2)

    except IOError as e:

        print('ERROR')
        print(e)
        time.sleep(10)
        continue

    exit(0)
```

**Appendice 10: mainMemsStation.py**

Il processo "mainMemsStation.py" è quello fondamentale per il l'acquisizione del segnale. Nel ciclo principale sono presenti i controlli sul funzionamento dell'accelerometro e sull'impostazione dei suoi parametri.

Le altre parti importanti del codice riguardano la ricezione dei campioni da parte del dispositivo, definita nella funzione "AccelerationChangeHandler" e l'avvio del processo parallelo di scrittura dei campioni (la funzione "writeSamplesINT32") il cui codice è contenuto nel file "phidgetAttLib.py".

```
import os
import sys
import time
import numpy as np
import math

from Queue import Queue
from threading import Thread
from obspy.core import Trace,Stream,UTCDateTime
from Phidget22.Devices.Accelerometer import *
from Phidget22.PhidgetException import *
from Phidget22.Phidget import *
from Phidget22.Net import *
from seismemslib import *
from phidgetAttlib import *

def AccelerationChangeHandler(e, acceleration, timestamp):

    global qx
    global qy
    global qz
    global t
    global ts

    a0 = prepareINT32data(acceleration[0])
    a1 = prepareINT32data(acceleration[1])
    a2 = prepareINT32data(acceleration[2]*(-1))
    ts.put(timestamp)

    qx.put( a0 )
    qy.put( a1 )
    qz.put( a2 )
    newtime = UTCDateTime()
    t.put(newtime)

## START MAIN CODE
os.nice(-20)

## LOCAL VARIABLES
i = 0
n = 0
startThread = 1
```

```

## GLOBAL VARIABLES
global qx,qy,qz,t,ts,last_ts
global f

qx = Queue()
qy = Queue()
qz = Queue()
t = Queue()
ts= Queue()

## READ CONF FILE

conf_values = readConfFile()
datapoints = float(conf_values['sampling_frequency'])* float(conf_values['seed_dim'])
f = int( 1000/( float(conf_values['sampling_frequency'] ) ) )

## OPEN DEVICES
## Infinite Loop, 5 second wait for HW fail

while True:
    ## DEVICE SETUP

    try:

ch = Accelerometer()
    ch.setOnAttachHandler(AccelerometerAttached)
    ch.setOnDetachHandler(AccelerometerDetached)
    ch.setErrorHandler(ErrorEvent)
    ch.openWaitForAttachment(5000)
    ch.setDataInterval( int( 1000/( float(conf_values['sampling_frequency'] ) ) ) )
    ch.setAccelerationChangeTrigger(0)
    ch.setOnAccelerationChangeHandler(AccelerationChangeHandler)
    break

    except PhidgetException as e:
print("Phidget Exception SETUP %i: %s" % (e.code, e.details))
        #ch.close()
        time.sleep(0.5)
        continue
    except RuntimeError as e:
print("Runtime Exception %s" % e.details)
        #ch.close()
        time.sleep(0.5)
        continue
    except IOError as e:
print("Exception %s" % e.details)
        #ch.close()
        time.sleep(0.5)
        continue

```

```
except Exception as e:
print("Exception %s" % e.details)
    #ch.close()
    time.sleep(0.5)
    continue

print("MEMS STATION IS ON. Insert 'exit' to stop listening")
print(ch.getAttached())

while True :

    if startThread == 1:

fireWriter = Thread(target=writeSamplesINT32, args=(qx,qy,qz,t, ts, conf_values))
    fireWriter.start()
    startThread = 0

    else:

        if not fireWriter.isAlive():
fireWriter = Thread(target=writeSamplesINT32, args=(qx,qy,qz,t, ts, conf_values))
            fireWriter.start()
print("THREAD NOT ALIVE")

        time.sleep(10)

    ## CLOSE DEVICES
print("Sampler Close. Goodbye")

exit(0)
```

## Appendice 11: phidgetAttLib.py

Nel file phidgetAttLib.py sono presenti le funzioni relative al collegamento ed al riconoscimento dell'accelerometro e la funzione che descrive il comportamento del processo parallelo di scrittura dei campioni chiamata "writeSamplesINT32".

In questa versione del codice i campioni vengono salvati nel file con una codifica intera a 32 bit con segno. Visto che il dato relativo al campione è fornito dal sensore come un numero in virgola mobile, la conversione è effettuata moltiplicando il valore per  $2^{28}$ . Tale procedimento restituisce valori corretti solo per numeri in ingresso compresi tra più e meno 7 g.

Il codice interno alla funzione si occupa di leggere i campioni dalla struttura dati condivisa con il programma principale e scrivere il file miniseed, usando le librerie "obspy", quando il numero di campioni ricevuti è sufficiente per chiudere il file secondo le dimensioni definite nel file set.conf. Dopo una analisi empirica del comportamento del sensore, si è notato che sporadicamente restituisce un campione prima del tempo voluto (1 millisecondo dopo invece che 4 millisecondi, campionando a 250 Hz), anticipando di fatto il campionamento dei successivi campioni. Il codice è stato adattato a questo comportamento, scartando sequenzialmente tre campioni errati e inserendo nel file solo il quarto campione in modo da ristabilire il corretto campionamento.

```
import numpy
from seismemslib import *
from obspy.core import Trace,Stream,Stats, UTCDateTime
import matplotlib.pyplot as plt
```

```
from Phidget22.Devices.Accelerometer import *
from Phidget22.PhidgetException import *
from Phidget22.Phidget import *
from Phidget22.Net import *
```

```
global precision,f
```

```
conf_values = readConfFile()
f = int( 1000/( float(conf_values['sampling_frequency'] ) ) )
precision = 2**28
```

```
#Converting data from phidget 16bit float output to
#standard INT32 format (C-style)
```

```
def prepareINT32data(in_val):
    p = float(precision)
    out_f = numpy.floor(in_val*p)

    return numpy.int32(out_f)
```

```
def AccelerometerAttached(e):
```

```
    global f
    try:

        attached = e
        attached.setDataInterval(f)
        attached.setAccelerationChangeTrigger(0)
```

```

print("\nAttach Event Detected (Information Below)")
print("=====")
print("Library Version: %s" % attached.getLibraryVersion())
print("Serial Number: %d" % attached.getDeviceSerialNumber())
print("Channel: %d" % attached.getChannel())
print("Channel Class: %s" % attached.getChannelClass())
print("Channel Name: %s" % attached.getChannelName())
print("Device ID: %d" % attached.getDeviceID())
print("Device Version: %d" % attached.getDeviceVersion())
print("Device Name: %s" % attached.getDeviceName())
print("Device Class: %d" % attached.getDeviceClass())
print("Sampling GFrequency: %d " % attached.getDataInterval() )
    print("\n")

    except PhidgetException as e:
print("Phidget Exception %i: %s" % (e.code, e.details))

def AccelerometerDetached(e):
    detached = e
    try:

print("\nDetach event on Port %d Channel %d" % (detached.getHubPort(), detached.getChannel()))

    except PhidgetException as e:
print("Phidget Exception %i: %s" % (e.code, e.details))

def ErrorEvent(e, eCode, description):
print("Error %i : %s" % (eCode, description))

def writeSamplesINT32(qx, qy, qz ,t, ts, conf_v):

seed_size = float(conf_v['sampling_frequency'])* float(conf_v['seed_dim'])

x=numpy.zeros([seed_size], dtype=numpy.int32)
y=numpy.zeros([seed_size], dtype=numpy.int32)
z=numpy.zeros([seed_size], dtype=numpy.int32)

if (float(conf_v['sampling_frequency'] ) < 250):
stat_x = createStats("HNN", conf_v)
stat_y = createStats("HNE", conf_v)
stat_z = createStats("HNZ", conf_v)
    #print('CANALE H')

else :

stat_x = createStats("CNN", conf_v)
stat_y = createStats("CNE", conf_v)

```

```

stat_z = createStats("CNZ", conf_v)
    #print('CANALE C')

index = 0
missCount = 0
firstTs = -1

while True :

    try:

        if firstTs == -1:

            firstTs = ts.get(True,2)
            ts.task_done()
            refTs = firstTs

        else:

            firstTs = refTs
            refTs = ts.get(True,2)
            ts.task_done()

    if ( refTs - firstTs ) == 1 and ( refTs - firstTs ) > 0 and missCount < 3:
        missCount = missCount +1
    print(' Miss {}'.format(missCount))
        qx.get(True,2)
        qy.get(True,2)
        qz.get(True,2)

        qx.task_done()
        qy.task_done()
        qz.task_done()

    else:

        x[index] = qx.get(True,2)
        y[index] = qy.get(True,2)
        z[index] = qz.get(True,2)

        qx.task_done()
        qy.task_done()
        qz.task_done()

    if index == 0 :
        starttime= t.get(True,2)
    else :

        t.get(True,2)

```

```

t.task_done()

index = index + 1

if index == seed_size:

    #WRITE X FILE
stat_x.starttime = starttime
stat_x.npts = index
stream =Stream([Trace(data=x, header=stat_x)])
                                stream.write('/home/pi/data/miniseed/x-%s-%s-%s-
%s:%s:%s:%s.mseed'%(starttime.year,starttime.month,starttime.day,starttime.hour,starttime.minute,start
time.second,starttime.microsecond),
format='MSEED',encoding='INT32',reclen=512)
        #WRITE y FILE
stat_y.starttime = starttime
stat_y.npts = index
stream =Stream([Trace(data=y, header=stat_y)])
                                stream.write('/home/pi/data/miniseed/y-%s-%s-%s-
%s:%s:%s:%s.mseed'%(starttime.year,starttime.month,starttime.day,starttime.hour,starttime.minute,start
time.second,starttime.microsecond),
format='MSEED',encoding='INT32',reclen=512)
        #WRITE Z FILE
stat_z.starttime = starttime
stat_z.npts = index
stream =Stream([Trace(data=z, header=stat_z)])
                                stream.write('/home/pi/data/miniseed/z-%s-%s-%s-
%s:%s:%s:%s.mseed'%(starttime.year,starttime.month,starttime.day,starttime.hour,starttime.minute,start
time.second,starttime.microsecond),
format='MSEED',encoding='INT32',reclen=512)

index = 0

if ( refTs - firstTs ) == 1 and ( refTs - firstTs ) > 0 and missCount ==3:
print('Last Miss {} {}'.format(refTs , firstTs))

x[index] = qx.get(True,2)
y[index] = qy.get(True,2)
z[index] = qz.get(True,2)

qx.task_done()
qy.task_done()
qz.task_done()

if index == 0 :
    starttime= t.get(True,2)
    #print(starttime)
else :

    t.get(True,2)

```

```

t.task_done()
index = index + 1

if index == seed_size:

    #WRITE X FILE
stat_x.starttime = starttime
stat_x.npts = index
stream =Stream([Trace(data=x, header=stat_x)])
stream.write('/home/pi/data/miniseed/x-%s-%s-%s-
%s:%s:%s:%s.mseed'%(starttime.year,starttime.month,starttime.day,starttime.hour,starttime.minute,start
time.second,starttime.microsecond),
format='MSEED',encoding='INT32',reclen=512)
    #WRITE y FILE
stat_y.starttime = starttime
stat_y.npts = index
stream =Stream([Trace(data=y, header=stat_y)])
stream.write('/home/pi/data/miniseed/y-%s-%s-%s-
%s:%s:%s:%s.mseed'%(starttime.year,starttime.month,starttime.day,starttime.hour,starttime.minute,start
time.second,starttime.microsecond),
format='MSEED',encoding='INT32',reclen=512)
    #WRITE Z FILE
stat_z.starttime = starttime
stat_z.npts = index
stream =Stream([Trace(data=z, header=stat_z)])
stream.write('/home/pi/data/miniseed/z-%s-%s-%s-
%s:%s:%s:%s.mseed'%(starttime.year,starttime.month,starttime.day,starttime.hour,starttime.minute,start
time.second,starttime.microsecond),
format='MSEED',encoding='INT32',reclen=512)

index = 0
missCount = 0

except Exception as e:
print("NO DATA IN THE QUEUE" + str(e))
print("INDEX EXCEPTION : {}".format(index))

if index>0:

print("FLUSH DATA : {}".format(index))

stat_x.starttime = starttime
stat_x.npts = index
stream =Stream([Trace(data=x, header=stat_x)])
stream.write('/home/pi/data/miniseed/x-%s-%s-%s-%s:%s:%s:%s.mseed%'

(starttime.year,starttime.month,starttime.day,starttime.hour,starttime.minute,starttime.second,starttime.m
icrosecond),
format='MSEED',encoding='INT32',reclen=512)

```

```
stat_y.starttime = starttime
stat_y.npts = index
stream =Stream([Trace(data=y, header=stat_y)])
    stream.write('/home/pi/data/miniseed/y-%s-%s-%s-%s:%s:%s:%s.mseed'%(starttime.year,start-
time.month,starttime.day,starttime.hour,starttime.minute,starttime.second,starttime.microsecond),
format='MSEED',encoding='INT32',reclen=512)

stat_z.starttime = starttime
stat_z.npts = index
stream =Stream([Trace(data=z, header=stat_z)])
    stream.write('/home/pi/data/miniseed/z-%s-%s-%s-%s:%s:%s:%s.mseed'%(starttime.year,start-
time.month,starttime.day,starttime.hour,starttime.minute,starttime.second,starttime.microsecond),
format='MSEED',encoding='INT32',reclen=512)

    index = 0
    misCount = 0

    continue

print("end of thread")
```

## Appendice 12: Seismemslib.py

Il file seismemslib.py contiene la funzione necessaria per la lettura e trasformazione in una struttura dati di tipo "dictionary", utilizzabile nel resto del codice python, del file set.conf e quella che crea l'header del file miniseed.

```
from obspy.core import Trace,Stream,Stats
```

```
def readConfFile():
```

```
def getFreq(conf_values,value):  
conf_values['sampling_frequency']= value
```

```
def getViewGraph(conf_values,value):  
conf_values['view-graph']= value
```

```
def getStart(conf_values,value):  
    return 0  
    #conf_values['view-graph']= value
```

```
def getEnd(conf_values,value):  
    return 0  
    #conf_values['view-graph']= value
```

```
def getNetwork(conf_values,value):
```

```
conf_values['network']= value
```

```
def getStation(conf_values,value):
```

```
conf_values['station']= value
```

```
def getLocation(conf_values,value):
```

```
conf_values['location']= value
```

```
def getSeedDim(conf_values,value):
```

```
conf_values['seed_dim']= value
```

```
def getFiltering(conf_values,value):
```

```
conf_values['filtering']= value
```

```
options = {'sampling_frequency': getFreq,  
          'view-graph': getViewGraph,  
          'start': getStart,  
          'end': getEnd,  
          'network':getNetwork,  
          'station': getStation,  
          'location': getLocation,
```

```
        'seed_dim':getSeedDim,
        'filtering':getFiltering
    }

    conf_v = {}

f = open('set.conf', 'r')

    for line in f:
if line[0] not in ('#', ' ', '\n'):
values = line.split(':')

if values[0] != 'end':
    #print values
temp_val = values[1].split('\n')

options[values[0]](conf_v,temp_val[0])

    return conf_v

def createStats(channel, conf_v):

stats = Stats()

stats= {'network': conf_v['network'],
        'station': conf_v['station'],
        'location': conf_v['location'],
        'channel': channel,
        'npts': int(float(conf_v['sampling_frequency'])*float(conf_v['seed_dim'])),
        'sampling_rate': conf_v['sampling_frequency'],
        'mseed' : {'dataquality' : 'D'},
        'starttime': ''}

return stats
```

### Appendice 13: set.conf

Nel file set.conf è possibile specificare i valori per la configurazione del sensore e quelli relativi all'identificazione della specifica stazione sismica. Nello specifico, i possibili parametri sono:

sampling\_frequency: la frequenza di campionamento

network: la rete a cui la stazione afferisce (ad esempio IV)

station: il nome della stazione

location: la posizione della stazione

seed\_dim: la dimensione in secondi del file miniseed da creare.

```
#SETTING FILE
```

```
#
```

```
#ADD property:value after the start: line
```

```
start:
```

```
sampling_frequency:250
```

```
network:IV
```

```
station:MEMS
```

```
location:00
```

```
seed_dim:5
```

```
end
```

## Appendice 14: Avvio Automatico

Per impostare l'avvio di un processo all'avvio del sistema operativo, si può sfruttare il modulo crontab del sistema operativo Raspbian. Aprire un terminale ed eseguire i seguenti comandi.

Entrare nel file di configurazione del crontab;

```
sudo crontab -e;
```

Inserire la seguente stringa infondo al file crontab:

```
@reboot sudo sh /home/pi/Desktop/swMEMS/start_acquisition.sh &;
```

Salvare il file appena modificato con “Control-O + Invio + Invio” ed uscire con “Control-X + Invio”;

Il comando “@reboot” permette di eseguire l'istruzione che la segue ad ogni avvio del sistema, mentre il carattere “&” dichiara di eseguire questa istruzione come processo indipendente. In particolare la stringa inserita permette di eseguire un file “.sh”, il file “.sh” in questione avvia tutti i processi necessari rendere il sistema una stazione sismica accelerometrica. Di seguito viene riportato il file “start\_acquisition.sh” creato utilizzando l'editor di testo.

Avvio del ring-server

```
cd  
cd /home/pi/ringserver  
./ringserver ring.conf > ring.log 2>&1
```

Puntare la cartella del progetto MEMS

```
cd  
cd /home/pi/Desktop/swMEMS
```

Avvio del processo di cancellazione dei file vecchi

```
sudo python deleteOldFile.py &
```

Avvio del processo di sincronizzazione con l'orario UTC

```
sudo python gpsDateDaemon.py &
```

Avvio del processo di acquisizione

```
sudo python mainMemsStation.py &
```

### **Allegato 15: Conti alla mano**

Una stazione sismica MEMS è costituita da:

Scheda microSD da 16 Gb.	10.00 €
Raspberry Pi 3	40.00 €
Alimentatore 5V 2,5A.	10.00 €
UPS da almeno 5200 mA.	20.00 €
Contenitore e componenti per l'inscatolamento	20.00 €
Phidgets Spatial 0/0/3	70.00 €
Phidgets GPS	30.00 €

Una singola stazione MEMS costa quindi circa 200 €.



# QUADERNI di GEOFISICA

ISSN 1590-2595

<http://istituto.ingv.it/le-collane-editoriali-ingv/quaderni-di-geofisica.html/>

I QUADERNI DI GEOFISICA (QUAD. GEOFIS.) accolgono lavori, sia in italiano che in inglese, che diano particolare risalto alla pubblicazione di dati, misure, osservazioni e loro elaborazioni anche preliminari che necessitano di rapida diffusione nella comunità scientifica nazionale ed internazionale. Per questo scopo la pubblicazione on-line è particolarmente utile e fornisce accesso immediato a tutti i possibili utenti. Un Editorial Board multidisciplinare ed un accurato processo di peer-review garantiscono i requisiti di qualità per la pubblicazione dei contributi. I QUADERNI DI GEOFISICA sono presenti in "Emerging Sources Citation Index" di Clarivate Analytics, e in "Open Access Journals" di Scopus.

QUADERNI DI GEOFISICA (QUAD. GEOFIS.) welcome contributions, in Italian and/or in English, with special emphasis on preliminary elaborations of data, measures, and observations that need rapid and widespread diffusion in the scientific community. The on-line publication is particularly useful for this purpose, and a multidisciplinary Editorial Board with an accurate peer-review process provides the quality standard for the publication of the manuscripts. QUADERNI DI GEOFISICA are present in "Emerging Sources Citation Index" of Clarivate Analytics, and in "Open Access Journals" of Scopus.

# RAPPORTI TECNICI INGV

ISSN 2039-7941

<http://istituto.ingv.it/le-collane-editoriali-ingv/rapporti-tecnici-ingv.html/>

I RAPPORTI TECNICI INGV (RAPP. TEC. INGV) pubblicano contributi, sia in italiano che in inglese, di tipo tecnologico come manuali, software, applicazioni ed innovazioni di strumentazioni, tecniche di raccolta dati di rilevante interesse tecnico-scientifico. I RAPPORTI TECNICI INGV sono pubblicati esclusivamente on-line per garantire agli autori rapidità di diffusione e agli utenti accesso immediato ai dati pubblicati. Un Editorial Board multidisciplinare ed un accurato processo di peer-review garantiscono i requisiti di qualità per la pubblicazione dei contributi.

RAPPORTI TECNICI INGV (RAPP. TEC. INGV) publish technological contributions (in Italian and/or in English) such as manuals, software, applications and implementations of instruments, and techniques of data collection. RAPPORTI TECNICI INGV are published online to guarantee celerity of diffusion and a prompt access to published data. A multidisciplinary Editorial Board and an accurate peer-review process provide the quality standard for the publication of the contributions.

# MISCELLANEA INGV

ISSN 2039-6651

[http://istituto.ingv.it/le-collane-editoriali-ingv/miscellanea-ingv.html](http://istituto.ingv.it/le-collane-editoriali-ingv/miscellanea-ingv.html/)

MISCELLANEA INGV (MISC. INGV) favorisce la pubblicazione di contributi scientifici riguardanti le attività svolte dall'INGV. In particolare, MISCELLANEA INGV raccoglie reports di progetti scientifici, proceedings di convegni, manuali, monografie di rilevante interesse, raccolte di articoli, ecc. La pubblicazione è esclusivamente on-line, completamente gratuita e garantisce tempi rapidi e grande diffusione sul web. L'Editorial Board INGV, grazie al suo carattere multidisciplinare, assicura i requisiti di qualità per la pubblicazione dei contributi sottomessi.

MISCELLANEA INGV (MISC. INGV) favours the publication of scientific contributions regarding the main activities carried out at INGV. In particular, MISCELLANEA INGV gathers reports of scientific projects, proceedings of meetings, manuals, relevant monographs, collections of articles etc. The journal is published online to guarantee celerity of diffusion on the internet. A multidisciplinary Editorial Board and an accurate peer-review process provide the quality standard for the publication of the contributions.

**Coordinamento editoriale e impaginazione**

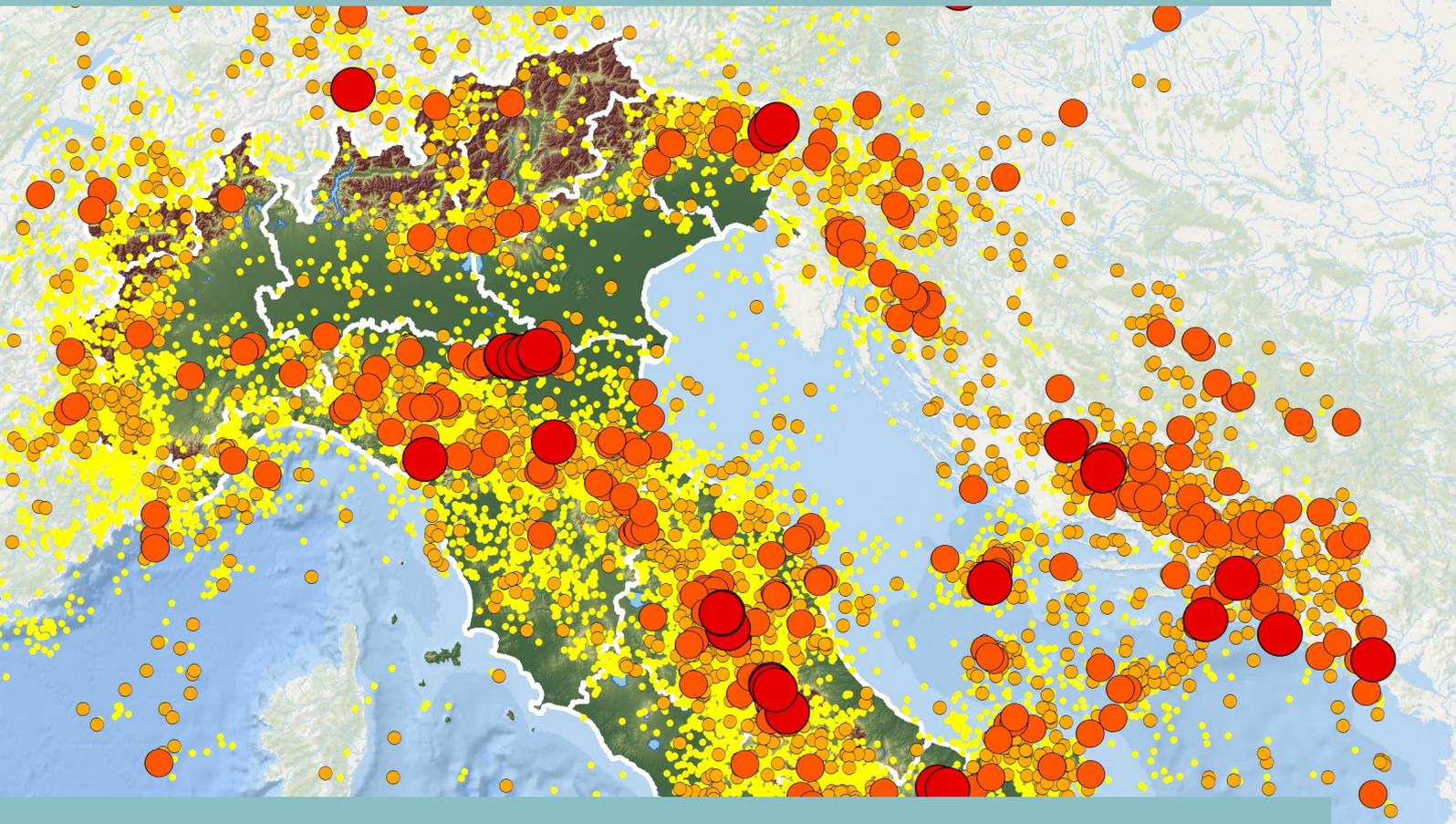
Francesca DI STEFANO, Rossella CELI  
Istituto Nazionale di Geofisica e Vulcanologia

**Progetto grafico e impaginazione**

Barbara ANGIONI  
Istituto Nazionale di Geofisica e Vulcanologia

©2019  
Istituto Nazionale di Geofisica e Vulcanologia  
Via di Vigna Murata, 605  
00143 Roma  
tel. +39 06518601

[www.ingv.it](http://www.ingv.it)



ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

