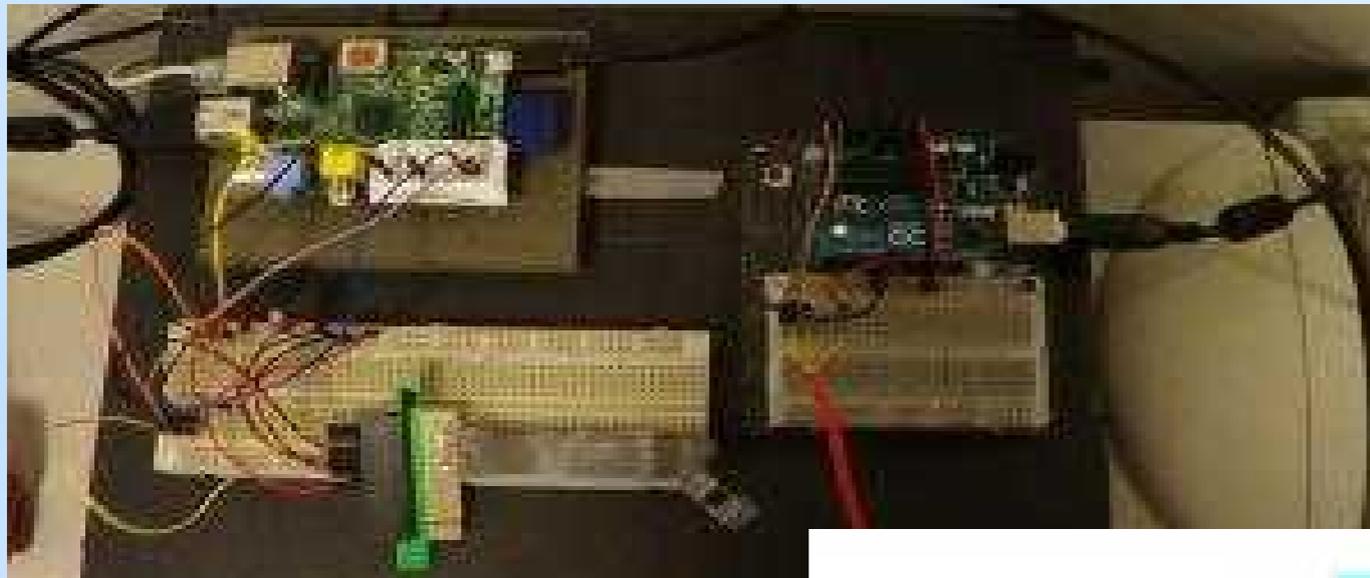


Stima: prototipo di stazione meteo



rete monitoraggio ambientale partecipativo





Smart city, smart citizen e citizen science

- Smart city: territorio urbano che permette di soddisfare le esigenze dei cittadini, delle imprese e delle istituzioni, mediante l'ausilio di strumenti innovativi e partecipazione attiva
- Ad esempio, reti di sensori per il controllo di parametri ambientali.
- Per la Comunità Europea, il grado di intelligenza di una città dovrebbe essere valutato secondo economia, mobilità, ambiente, persone, tenore di vita e governo.
- All'interno di questo ambiente, il cittadino può (deve) avere un ruolo attivo (smart citizen)
- Un esempio di partecipazione attiva può essere ritrovata nella citizen science, una modalità di ricerca scientifica condotta totalmente o in parte da scienziati non professionisti



Inquinamento

- la legislazione europea dice che in una città come Bologna bastano due o tre stazioni per la misura della qualità dell'aria
- sono sufficienti per monitorare gli aspetti generali dell'inquinamento urbano, poi però esistono punti di accumulo locali (sotto i portici, nelle strade strette), differenze tra il primo e l'ultimo piano, emissioni locali (impianti), anche episodiche (cantieri, ingorghi, caminetti accesi), inquinamento dentro le case...
- non basterebbero centinaia di centraline per monitorare tutta questa varietà di situazioni, e i costi delle reti di misura sono già adesso difficili da sostenere
- l'Agenzia Ambientale Europea promuove l'integrazione di strumenti diversi: satelliti, modelli, poche stazioni con strumenti avanzati e reti di microsensori a basso costo gestite da volontari (scuole, associazioni, cittadini)



per un problema multi-scala, monitoraggio multi-scala

- perciò sì, qualche grande pennello e poi pennelli piccoli per definire i dettagli





Obiettivi R-map

- Raccogliere e distribuire dati ambientali rilevati dai cittadini
- Rendere disponibili questi dati ai servizi meteorologici, alle agenzie di prevenzione ambientale, alla protezione civile e istituti di ricerca
- Fornire feedback ai fornitori di dati in modo che essi abbiano servizi per testare e migliorare la qualità dei dati
- Divulgazione scientifica e sensibilizzazione ai temi ambientali
- Coinvolgimento di scuole e università a scopi formativi
- Creare un circolo virtuoso tra Enti Formativi, Pubbliche Amministrazioni, Aziende private e cittadini.



Soggetti coinvolti

- **ARPA Emilia Romagna SIMC**
 - Predisporre un disciplinare tecnico rispetto ai metodi di misura, elaborazione dei dati dei sensori e loro collocazione
 - Definisce i protocolli e i formati per la comunicazione dei dati
 - Eseguire un eventuale controllo di qualità e comunica i risultati
- **ARPA Regione Veneto**
 - Supporta la diffusione degli standard/fornisce infrastruttura hardware
 - Contribuisce allo sviluppo e sperimenta stazioni commerciali con supporto allo standard Rmap
 - Sperimenta nuova sensoristica a basso/medio costo
- **Cineca Consorzio Interuniversitario per il supercalcolo e l'innovazione tecnologica**
 - Sperimenta e sviluppa le tecnologie Stima
 - Supporto BigData
 - Fornisce servizi a valore aggiunto



Soggetti coinvolti

- **Dipartimento informatica Università di Bologna**
 - Esprime pareri sul progetto e prototipo hardware e software
 - Eventuali tesi/tirocini sul progetto software
- **RaspiBO:** gruppo informale di appassionati di elettronica ed informatica libera della zona di Bologna
 - Realizzano un prototipo hardware e software
 - Sperimentano il prototipo
 - Documentano la realizzazione del prototipo come openhardware e distribuiscono il software con licenza libera
- **Soggetto privato / startup**
 - Progetto commerciale per la vendita, installazione e manutenzione delle stazioni: DigitEco vincitrice appalto ARPAE e Acronet per progetto RainBo
- **Scuole:** collaborazione con un gruppo di scuole di Parma
- **Soggetti già attivi sul territorio (Meteonetwork)**



Cosa è Rmap

- Un insieme di specifiche:
- **Protocollo di rilevamento dati**
 - Collocazione sensori
 - Accuratezza dei sensori
 - Elaborazioni

http://www.raspibo.org/wiki/index.php/Gruppo_Meteo/DisciplinareStazione

- **Sistema per lo scambio dati**
 - Protocolli di trasmissione
 - Formati dati
 - Metadati

http://www.raspibo.org/wiki/index.php/Gruppo_Meteo/RFC-rmap



Implementazioni hardware e software

- Specifiche realizzazioni che aderiscono allo standard Rmap
- Possibilmente open hardware e open software

STIMA/Server R-map

(Acronet)



Specifiche Rmap



Campionamento di variabili meteorologiche

- Campionamento è il processo per ottenere una discretizzata sequenza di misure di una quantità.
- Campione è una singola misura, tipicamente una di una serie di letture “spot” di un sistema di sensoristica.
- Una osservazione (misurazione) è il risultato del processo di campionamento. Nel contesto di analisi di serie, un'osservazione è derivato da un numero di campioni.
- Variabili atmosferiche come la velocità del vento, temperatura, pressione e umidità sono funzioni di quattro dimensioni - due orizzontali, una verticale e una temporale. Esse variano irregolarmente in tutte e quattro, e lo scopo dello studio del campionamento è quello di definire le procedure di misura pratiche per ottenere osservazioni rappresentative con incertezze accettabili nelle stime delle medie e variabilità.



Data Level

- Dati **Level I** , sono le letture dirette degli strumenti espresse in appropriate unità fisiche e georeferenziate
- Dati **Level II**, dati riconosciuti come variabili meteorologiche; possono essere ottenuti direttamente da strumenti o derivati dai dati Level I
- Dati **Level III** sono quelli contenuti in dataset internamente consistenti, generalmente su grigliato.
- I dati scambiati a livello internazionale sono livello II o livello III



Disciplinare per il rilevamento di dati

Per ora una proposta per:

- Schermi dalla radiazione
- Temperatura
- umidità

Prima prebozza disponibile a:

- http://www.raspibo.org/wiki/index.php/Gruppo_Meteo/DisciplinareStazione



Conceptual data model

These models, sometimes called domain models, are typically used to explore domain concepts with project stakeholders

DB-All.e Conceptual data model

- Il modello è orientato all'applicazione (bisogna capire cosa sono i dati, normalizzarli e ricondurli a metadati standard in fase di accoglienza), quindi si lavora pre e non post
- I dati sono legati ai metadati in modo univoco
- Una osservazione è univoca nello spazio dei suoi metadati
- L'unica possibilità di far coesistere due osservazioni dello stesso parametro nello stesso punto è attraverso il metadato "network" associabile alla classe dello strumento
- La tracciabilità di un sensore, una stazione, un osservatore nello spazio, nel tempo etc. Avviene attraverso il metadato "ident"
- Alcuni metadati sono table driven (level, timerange, network)



- Ogni dato può essere associato a un certo numero di attributi
- Nessuna dimensione è vincolata (intervalli temporali tra dati, numero attributi....)
- E' contemplata la gestione di previsioni; il datetime è sempre quello di verifica
- Misure e metadati hanno troncamenti sulle cifre significative dettati dalla loro possibilità reale di misura e stabiliti a priori
- Esistono due categorie di dato: una che varia tutti i metadati (osservazioni classiche) e l'altra che non prevede l'uso di alcuni metadati e che quindi sono da considerarsi come ulteriori metadati di quella singola stazione (constant station data: es. Nome stazione)
- Nessuno vieta di espandere i metadati estendendo esternamente questo data model



Logical data model (LDM)

LDMs are used to explore the domain concepts, and their relationships, of your problem domain

DB-All.e LDM

- **METADATI**

- *Datetime*: tempo di fine misurazione
- *Ana*: *Longitudine, latitudine* ed un *identificativo*
- *network*: definisce stazioni con caratteristiche omogenee (classe degli strumenti)
- *Time range*: Tr,P1,P2 indica osservazione o tempo previsione ed eventuale elaborazione “statistica”
- *Level*: TL1,L1,TL2,L2 le coordinate verticali (eventualmente strato)
- *Variable*: Btable parametro fisico

- **DATI**

- *Valori rappresentabili come interi, reali, doppia precisione, stringhe*
 - *Attributi (alla stregua di dati)*



Bufr

- The Binary Universal Form for the Representation of meteorological data (BUFR) is a binary data format maintained by the World Meteorological Organization (WMO).
- BUFR was designed to be portable, compact, and universal. Any kind of data can be represented, along with its specific spatial/temporal context and any other associated metadata. In the WMO terminology, BUFR belongs to the category of table-driven code forms, where the meaning of data elements is determined by referring to a set of tables that are kept and maintained separately from the message itself.
- Descriptors: all element descriptors will be found in BUFR specification section known as "Table B". The Table B definition of an element descriptor includes its number, short text definition, decoding parameters (bit width, scale factor, and bias), and type (numerical, character string, code table, etc.).



Software di decodifica Bufr

- **WREPORT**: a featureful C++ library for BUFR and CREX encoding and decoding
<http://sourceforge.net/p/wreport/home/Home/>
- **ecCodes** <https://software.ecmwf.int/wiki/display/ECC/ecCodes+Home>



Formato JSON

- <http://www.json.org/json-it.html>
- JSON (JavaScript Object Notation) è un semplice formato per lo scambio di dati. Per le persone è facile da leggere e scrivere, mentre per le macchine risulta facile da generare e analizzarne la sintassi.
- Rispetta il data model che ci siamo dati
- Ogni elemento è il report con i dati di una certa stazione per un certo istante di riferimento.
- Fornito mediante postprocessatore **json** di Arkimet, ma disponibile anche come tool da riga di comando



Formato GeoJSON per punti sparsi

- <http://geojson.org/> <https://macwright.org/2015/03/23/geojson-second-bite.html>
- the Internet Engineering Task Force (IETF) ha pubblicato in Agosto 2016 RFC 7946
- E' un formato molto verboso (soprattutto per le serie temporali) ma
- Rispetta il data model che ci siamo dati (anche se in modo non ottimale)
- E' uno dei formati vettoriali interpretati da **GDAL/OGR**
 - OpenLayers
 - QGIS
 - ...
- Fornito mediante postprocessatore **json** di Arkimet, ma disponibile anche come tool da riga di comando



Protocolli per R-map

- **MQTT** (Message Queue Telemetry Transport) è un protocollo publish/subscribe particolarmente leggero, adatto per la comunicazione M2M tra dispositivi con poca memoria o potenza di calcolo e server o message broker.
- **AMQP** (Advanced Message Queuing Protocol) è protocollo per comunicazioni attraverso code di messaggi. Sono garantite l'interoperabilità, la sicurezza, l'affidabilità, la persistenza. Nella sua implementazione Rabbitmq esporta un broker MQTT e fornisce delle api web
- Json è il formato per il payload

E' fondamentale:

- Integrazione con le funzioni e le specifiche richieste dalle applicazioni per la **domotica**
- Integrazione con applicazioni per la **telefonia mobile** per la rilevazione dello spessore neve e altri parametri



Metadati su MQTT

- Ogni topic corrisponde ai metadati univoci, mentre il payload è composto dal valore e dall'istante temporale
- **/IDENT/COORDS/NETWORK/TRANGE/LEVEL/VAR**
 - **IDENT**: identificativo per stazioni mobili, “-” per stazioni fisse
 - **COORDS**: nella forma *lon,lat*. Le coordinate sono espresse nell forma *int(valore*10⁵)* con eventuale segno negativo
 - **NETWORK**: massimo 16 caratteri
 - **TRANGE**: nella forma *indicator,p1,p2*
 - *Indicator* e *p2* interi senza segno, *p1* intero con eventuale segno negativo. “-” per valori non significativi
 - **LEVEL**: nella forma *type1,l1,type2,l2*
 - *Type1*, *type2* interi con eventuale segno negativo, *l1* e *l2* interi con eventuale segno negativo. “-” per valori non significativi
 - **VAR**: nella forma *BXXYYY*
- Il payload è in formato JSON: `{ “v”: VALUE, “t”: TIME, “a”: { “BXXYYY”: VALUE, ... } }`
 - *VALUE*: valore in formato CREX
 - *TIME*: formato YYYY-mm-ddTHH:MM:SS.MSC (secondi e millisecondi opzionali)
 - Gli attributi (“a”) sono opzionali



RMAP web services

Composizione degli URL per un HTTP GET request

La "base" della richiesta è quella standard:

`/version/ident/coords/network/timerange/level/bcode/`

Ad esempio:

`http://rmap.cc/v0.1/-/1207738,4460016/locali/
254,0,0/103,2000,-,-/B12101`



Serie temporale

Serie temporale mensile, giornaliera e annuale:

/ident/coords/network/timerange/level/bcode/timeseries/year

/ident/coords/network/timerange/level/bcode/timeseries/year/month

/ident/coords/network/timerange/level/bcode/timeseries/year/month/day

Ad esempio:

/-/1207738,4460016/locali/254,0,0/103,2000,-,-/B12101/timeseries/2011

/-/1207738,4460016/locali/254,0,0/103,2000,-,-/B12101/timeseries/2011/01

/-/1207738,4460016/locali/254,0,0/103,2000,-,-/B12101/timeseries/2011/01/13



Serie spaziale e sommario

Serie spaziale di una rete, con granularità oraria (± 30 minuti):

`/**/NETWORK/TIMERANGE/LEVEL/BCODE/spatialseries/YEAR/MONTH/DAY/HOUR`

Riassuntivo

`/**/**/**/**/**/summaries`

`/**/**/**/**/**/summaries`

- `/**/NETWORK/**/**/summaries`
- `/-/COORDINATES/NETWORK/**/**/summaries`
- `/IDENT*/NETWORK/**/**/summaries`
- `/**/NETWORK/TIMERANGE/LEVEL/BCODE/summaries/YEAR/MONTH`
- `/**/NETWORK/TIMERANGE/LEVEL/BCODE/summaries/YEAR/MONTH/DAY`



Il prototipo di stazione sviluppato da RaspiBO





I principi dell'Open Source Hardware (OSHW)

- **hardware** il cui progetto è reso **pubblico** in modo che chiunque possa **studiare, modificare, distribuire, realizzare, e vendere** il progetto o l'hardware basato su di esso
- La fonte dell'hardware, il progetto da cui è stato realizzato, è disponibile nel **formato migliore per apportarvi modifiche**. Idealmente, l'hardware open source **utilizza componenti e materiali disponibili, processi standard, infrastruttura aperta, contenuti senza restrizione e strumenti di progettazione open-source** per massimizzare la capacità degli individui di produrre e utilizzare l'hardware.
- L'hardware open source dà alle persone la libertà di controllare la loro tecnologia, la condivisione della conoscenza ed incoraggia il commercio attraverso lo scambio aperto di progetti.

Open Source Hardware Association
<http://www.oshwa.org/definition/italian/>



La Definizione 1.0 dell'Open Source Hardware

- La documentazione :La documentazione deve includere i file del progetto nel formato preferito per apportare modifiche, ad esempio, il formato nativo del file di un programma CAD
- Il Software necessario:
 - Le interfacce sono sufficientemente documentate tale che si possa scrivere il software open source che consente al dispositivo di funzionare
 - Il software necessario è rilasciato sotto una licenza open source
- I lavori derivati: permettere modifiche e lavori derivati
- La redistribuzione libera
- L'attribuzione: riportare l'attribuzione ai licenzianti quando si fa la distribuzione; può richiedere che i lavori derivati abbiano un nome o un numero di versione diversi dal progetto originale
- Nessuna discriminazione di persone o gruppi
- La licenza non deve essere specifica per un prodotto



STIMA: vista d'insieme

- Open source hardware e software
- Utilizzo delle piattaforme più diffuse e board di prototipizzazione
- Disegni hardware con Kicad (by Daniele Castellari)
- 5 moduli hardware che soddisfano differenti esigenze
 - **Consumi:** alimentazione tramite rete, batterie con pannello solare, batterie
 - **Collegamento:** all'interno della casa, in esterno con un cavo ethernet e PoE, in esterno in postazione fissa ravvicinata, in esterno in postazione mobile
 - **Domotica:** sono collegabili attuatori
- Un modulo con funzioni **server**
 - Database
 - Web server
 - NTP server
 - Sviluppo



E' anche un “framework” per makers

- E' possibile utilizzare il modulo base per lo sviluppo del firmware
- Sono installate le librerie Arduino con i file di specifiche per i microcontrollori utilizzati
- Come build system si può utilizzare ARDUINO
- Tutte le librerie personalizzate sono già installate
- L'aggiornamento avviene tramite git / pip
- E' possibile “scriptare” il build, upload e configurazione delle board



Modulo base: vista d'insieme

circa 75€

- Sviluppato su **Raspberry**
- Distribuzione Pidora 2014 (fedora 20); migrazione a Centos 7
- Software completamente pacchettizzato RPM
- Repository software pubblico
<http://rmapv.rmap.cc/repo/rmap/fedora/20/RPMS/arm/repoview/>
- Per ora una immagine SD da 8G scaricabile e pronta all'uso
- Il modulo gestisce direttamente la sensoristica su I2C
- Testata connessione alla LAN con Ethernet WIFI e GSM
- Gestione di una propria LAN con dhcp server, dns server e nat



Trasporto

Il concetto di trasporto in Stima è simile ma non rigidamente aderente ai concetti del modello ISO-OSI.

Nel caso dei trasporti passivi il suo compito è fornire un canale logico-affidabile di comunicazione end-to-end per fornire servizi al soprastante livello che in Stima è JsonRPC.

Nel caso dei trasporti attivi corrisponde al protocollo (Session Layer) per la pubblicazione dei dati su un server (broker).



Trasporti passivi

In pratica i trasporti "passivi" permettono di eseguire procedure remote codificate in formato json specifiche dell'implementazione Stima; quelli attivi permettono la pubblicazione su server (broker) dei messaggi aderenti allo standard R-MAP.

Trasporti Passivi:

- Seriale
- TCP/IP
- Bluetooth (serial port profile)
- NRF24



Trasporti attivi

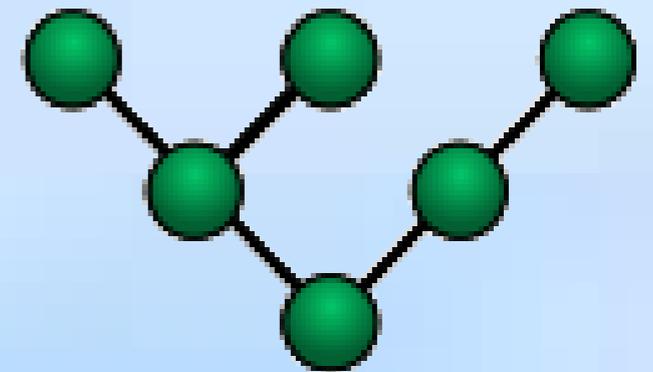
Trasporti attivi:

- MQTT
- AMQP



Trasporti

- Trasporto **Bluetooth** (HC-05)
- Trasporto **Seriale**
 - Principalmente per configurazione e debug
 - Piccole distanze via cavo
- Trasporto **TCP/IP**
 - Collegamenti tramite cavo ethernet a breve e media distanza
- Trasporto **RF24Network**
 - OSI Network Layer using nRF24L01(+) radios 2.4GHz ISM
 - 50/150m in aria libera
 - Host Addressing. Each node has a logical address on the local network.
 - Message Forwarding. Messages can be sent from one node to any other, and this layer will get them there no matter how many hops it takes.
 - Ad-hoc Joining. A node can join a network without any changes to any existing nodes.



Tree



RF24Network Addressing and Topology

Each node must be assigned an 15-bit address by the administrator. This address exactly describes the position of the node within the tree. The address is an octal number. Each digit in the address represents a position in the tree further from the base.

- Node 00 is the base node.
 - Nodes 01-05 are nodes whose parent is the base.
 - Node 021 is the second child of node 01.
 - Node 0321 is the third child of node 021, and so on.
 - The largest node address is 05555, so 3,125 nodes are allowed on a single channel.
- Alla libreria distribuita è stata aggiunta la crittografia e frammentazione e ricomposizione del payload



Per ora 4 tipi di trasporto

I moduli possono essere così caratterizzati dal trasporto supportato, se eseguono RPC, se richiedono RPC, se pubblicano su MQTT.

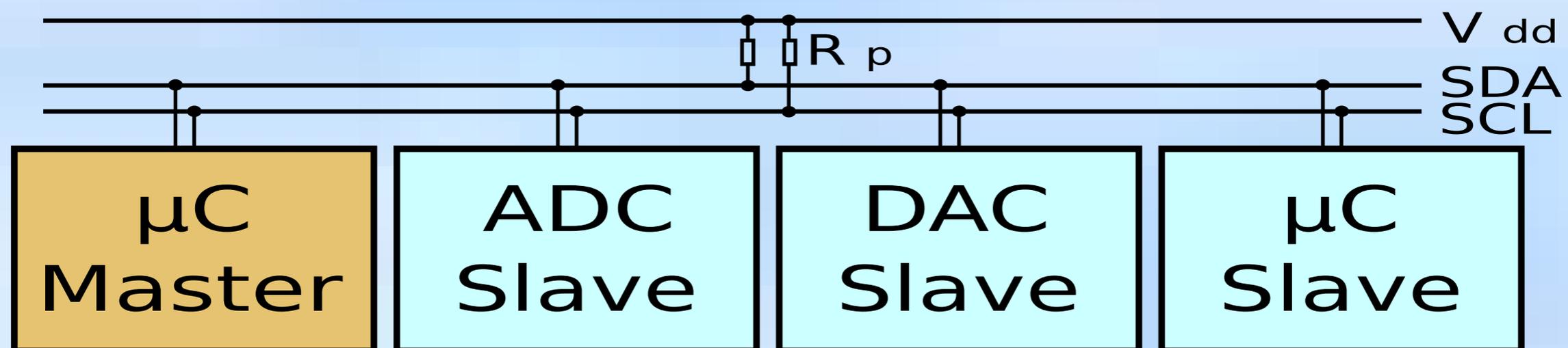
I moduli prototipati in r-map sono così denominati e caratterizzati:

	Attivo / Passivo	Trasp: Seriale	Trasp: TCP/IP	Trasp: radio RF24 Network	publish on MQTT	funzioni server	Alimentazione
Modulo Base	Attivo	Si	Si	Da sviluppare	Si	Si	Rete Batterie con pannello solare
Modulo Bluetooth	Passivo	Si	No	Si	No	No	Batterie
Modulo master	Attivo / Passivo	Si	Si	Si	Si	No	Ethernet PoE
Modulo satellite	Passivo	Si	No	Si	No	No	Batterie (con pannello solare)
Modulo GSM	Attivo/Passivo	Si	(Si)	Si	Si	No	Batterie con pannello solare



BUS I2C

- Il protocollo i2c prevede l'utilizzo di un bus formato da due linee bidirezionali. Le due linee, chiamate "scl" e "sda" rispettivamente, trasportano la tempistica di sincronizzazione (chiamata anche "clock") e i dati.
- Abbiamo scelto il bus i2c in quanto:
 - È diventato lo standard di fatto per una serie di integrati tra cui i sensori
 - Si possono collegare fino a 127 dispositivi
 - La comunicazione è bidirezionale (read e write) con velocità assolutamente sufficienti per i nostri scopi





Json-rpc

- Json
 - JavaScript Object Notation ed è un formato adatto ad immagazzinare varie tipologie di informazioni, e quindi a scambiare queste informazioni tra applicazioni client/server.
 - JSON possiede una struttura semplicissima
- JSON-RPC is lightweight remote procedure call protocol similar to XML-RPC. It's designed to be simple!

Esempi:

```
--> {"jsonrpc": "2.0", "method": "subtract", "params": {"subtrahend": 23, "minuend": 42}, "id": 3}
```

```
<-- {"jsonrpc": "2.0", "result": 19, "id": 3}
```

```
--> {"jsonrpc": "2.0", "method": "subtract", "params": {"minuend": 42, "subtrahend": 23}, "id": 4}
```

```
<-- {"jsonrpc": "2.0", "result": 19, "id": 4}
```