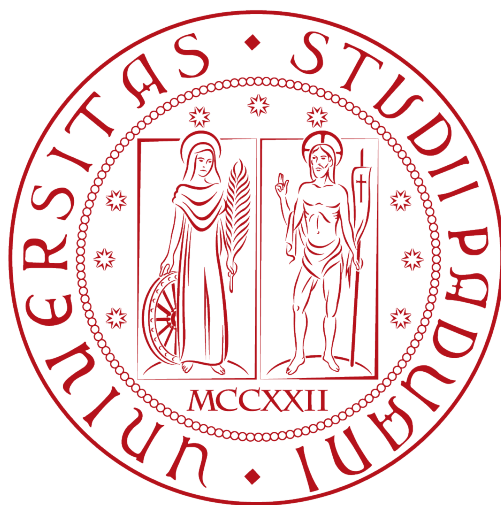


Università degli Studi di Padova

Dipartimento di Matematica

Corso di Laurea in Informatica



**Sviluppo di dispositivo embedded per la gestione
di un pluviometro con Arduino**

Tesi di laurea triennale

Relatore

Prof. Tullio Vardanega

Laureando

Marco Gregorini

Sommario

Premessa

Il seguente documento ha lo scopo di illustrare il lavoro svolto durante l'attività di stage dallo laureando Gregorini Marco, presso la sede del servizio informatico e di reti dell'agenzia regionale ARPAV, a Padova. Lo studente intende descrivere in modo critico e oggettivo le attività svolte durante il suo percorso di lavoro, della durata di circa trecentoventi ore, e i riscontri avuti con questa esperienza.

Contenuti

Nel documento vengono descritte le attività svolte per il completamento del progetto proposto da ARPAV, le necessità che hanno creato l'opportunità di lavoro per uno studente laureando e le prospettive che il progetto apre per il futuro. I contenuti vengono esposti in quattro capitoli:

1. **Profilo dell'Agenzia:** descrizione generale dell'azienda. Dall'organizzazione interna agli obbiettivi che si pone;
2. **Lo Stage per ARPAV:** lo stage visto dal punto di vista dell'ente. Dai motivi che hanno evidenziato una necessità di uno stage, alla presentazione del progetto;
3. **L'Attività di Stage:** lo stage visto dal punto di vista dello studente. Dalla pianificazione alla realizzazione;
4. **Valutazioni Finali:** analisi retrospettiva dell'esperienza fatta. Valutazione critica dei risultati ottenuti, descrizione delle capacità e abilità acquisite, valutazione dello svolgimento del lavoro svolto.

Nel documento sono state introdotte delle convenzioni topografiche per agevolare la lettura e la comprensione dei contenuti:

- **Glossario :** i termini che richiedono una definizione più accurata, di valore tecnico o di uso non comune, verranno descritti all'interno del Glossario e evidenziati nel testo con una g a pedice alla loro prima occorrenza: *parola_g*;
- **Termini Inglesi:** i termini in lingua inglese verranno evidenziati in corsivo: *parola*;

Indice

| | | |
|----------|---|-----------|
| 1 | Profilo dell'Agenzia | 1 |
| 1.1 | Cosa Offre: Prodotti e Servizi | 2 |
| 1.1.1 | I Prodotti di ARPAV | 2 |
| 1.1.2 | I Servizi di ARPAV | 5 |
| 1.2 | Organizzazione Interna | 8 |
| 1.2.1 | Processi di Sviluppo | 9 |
| 1.2.2 | Metodologie di Supporto ai Processi | 13 |
| 1.2.3 | Strumenti di Supporto ai Processi | 14 |
| 1.3 | Relazioni Esterne | 15 |
| 1.3.1 | Orientamento all'Innovazione | 15 |
| 2 | Lo Stage per ARPAV | 19 |
| 2.1 | L'Esigenza di uno Stage | 19 |
| 2.2 | Presentazione del Progetto | 20 |
| 2.2.1 | Obiettivo dello Stage | 22 |
| 2.2.2 | Finalità del Progetto | 24 |
| 2.3 | Vincoli di Progetto | 25 |

| | | |
|----------|---|-----------|
| 2.3.1 | Vincoli di Dominio | 25 |
| 2.3.2 | Vincoli Tecnologici | 26 |
| 2.3.3 | Vincoli Metodologici | 30 |
| 2.3.4 | Vincoli Temporalì | 30 |
| 3 | L'Attività di Stage | 31 |
| 3.1 | Pianificazione del Lavoro | 32 |
| 3.1.1 | Preparazione al Lavoro di Stage | 33 |
| 3.2 | Analisi dei Requisiti | 38 |
| 3.2.1 | Classificazione dei Requisiti | 38 |
| 3.2.2 | Individuazione dei Requisiti | 38 |
| 3.2.3 | Casi d'Uso | 41 |
| 3.2.4 | Dettagli Degni di Nota | 41 |
| 3.3 | Progettazione e Codifica | 41 |
| 3.3.1 | Introduzione Architettura Hardware | 41 |
| 3.3.2 | Introduzione Architettura Software | 43 |
| 3.3.3 | Architettura Libreria Scheda Master | 44 |
| 3.3.4 | Architettura Scheda Slave | 44 |
| 3.3.5 | Design Pattern Utilizzati | 44 |
| 3.3.6 | Dettagli Degni di Nota | 44 |
| 3.3.7 | Progettazione di Dettaglio e Codifica | 44 |
| 3.3.8 | L'Utilizzo dei Prototipi | 45 |
| 3.4 | Verifica e Validazione | 45 |
| 3.4.1 | Analisi Statica | 45 |

| | | |
|----------|--|-----------|
| 3.4.2 | <i>test</i> sul Sistema Slave | 45 |
| 3.4.3 | <i>test</i> sul Sistema Master Slave | 45 |
| 3.4.4 | <i>test</i> di Sistema | 45 |
| 3.4.5 | Dettagli Degni di Nota | 46 |
| 3.4.6 | Consuntivo Orario Finale | 46 |
| 4 | Valutazioni Finali | 47 |
| 4.1 | Raggiungimento degli Obbiettivi | 47 |
| 4.1.1 | Rendiconto degli Obbiettivi | 47 |
| 4.2 | Difficoltà Riscontrate | 47 |
| 4.3 | Bilancio Formativo | 47 |
| 4.3.1 | Preparazione Iniziale | 47 |
| 4.3.2 | Esperienze Acquisite | 48 |
| 4.3.3 | Considerazioni Finali | 48 |
| 5 | Glossario | 49 |

Elenco delle figure

| | | |
|-----|--|----|
| 1.1 | logo dell'agenzia | 1 |
| 1.2 | schema reti RESMIA | 4 |
| 1.3 | organigramma struttura ARPAV | 8 |
| 1.4 | organigramma Dipartimento Informatica e Reti | 9 |
| 1.5 | diagramma attività approvazione progetti | 10 |
| 1.6 | raffigurazione divisione problema approccio top-down | 11 |
| 1.7 | ciclo di vita a cascata | 12 |
| 1.8 | Ciclo di aggiornamento piano d'azione ARPAV | 16 |
| | | |
| 2.1 | Logo RaspiBO | 19 |
| 2.2 | Logo FabLab | 19 |
| 2.3 | Scheda Arduino UNO | 20 |
| 2.4 | Logo Arduino | 20 |
| 2.5 | Esempio protocollo MQTT | 21 |
| 2.6 | Prototipo pluviometro FabLab | 24 |
| 2.7 | Bascula di un pluviometro | 26 |
| 2.8 | Esempio di IDE Arduino | 26 |

| | | |
|------|--|----|
| 2.9 | Tiny RTC connessa a scheda Arduino UNO | 29 |
| 2.10 | Logo Qt | 29 |
| 2.11 | Logo GitHub | 30 |
| 2.12 | Logo Bitbucket | 30 |
| 3.1 | Modello a spirale | 31 |
| 3.2 | Diagramma di Gantt del periodo di Stage | 33 |
| 3.3 | Suddivisione lavoro prime due settimane | 33 |
| 3.4 | Manuale Arduino | 34 |
| 3.5 | Logo C++ | 36 |
| 3.6 | Microcip 24LC256 | 37 |
| 3.7 | Scheda <i>bread board</i> | 42 |
| 3.8 | Schema elettronico progetto | 42 |
| 3.9 | Foto prototipo <i>board</i> personalizzata | 43 |
| 3.10 | Schema elettronico <i>board</i> personalizzata | 43 |

Elenco delle tabelle

| | | |
|-----|---|----|
| 1.1 | mansioni dipartimento Servizio Informatico e Reti | 2 |
| 1.2 | progetti ARPAV | 3 |
| 1.3 | servizi ambientali ARPAV | 6 |
| 1.4 | servizi online di ARPAV | 7 |
| 3.1 | Suddivisione del lavoro | 32 |
| 3.2 | Ore per attività | 32 |
| 3.3 | Valori <i>timestamp</i> compressione | 35 |
| 3.4 | Conformazione <i>bit timestamp</i> | 35 |
| 3.6 | Requisiti | 40 |

Capitolo 1

Profilo dell'Agenzia

ARPAV_g è l'agenzia regionale per la prevenzione e protezione ambientale del Veneto, operativa dal 3 Ottobre 1997 in seguito alla Legge Regionale n32° del 18 Ottobre 1996.



Figura 1.1: logo dell'agenzia

Le attività competenti riguardano la tutela, il controllo, il recupero dell'ambiente e per la prevenzione e promozione della salute collettiva al fine di conseguire la massima efficacia nell'individuazione e nella rimozione dei fattori di rischio per l'uomo e per l'ambiente. Le funzioni principali dell'agenzia riguardano attività tecnico-scientifiche per il monitoraggio, tutela e prevenzione di acqua, aria (inquinamento acustico ed elettromagnetico negli ambienti di vita), suolo, rifiuti solidi e liquidi, radioattività ambientale ed infine ai rischi di incidenti rilevanti attività industriali. L'esercizio delle attività di monitoraggio e prevenzione vengono effettuate in coordinazione con le unità locali socio sanitarie.

L'agenzia è suddivisa in vari organi operativi, i quali hanno funzionalità specifiche a seconda del ruolo che ricoprono. La suddivisione degli incarichi e delle competenze e la corretta comunicazione fra i vari dipartimenti, permette di gestire questo vasto ente in modo efficiente e sistematico.

Lo *stage* in oggetto è stato svolto presso il servizio informatico e di reti. In questa sede vengono svolte le mansioni per la gestione dell'infrastruttura informatica e delle risorse strumentali hardware di tutta l'agenzia e attività di ricerca e sviluppo inerenti.

- gestione e coordinamento delle banche dati dell'agenzia;
- assistenza sulle applicazioni informatiche dell'agenzia;
- definizione degli indicatori ambientali e dei rapporti;
- fornitura degli standard operativi, architetture delle realizzazioni, attivazione e gestione tecnica dei portali internet/intranet;
- gestione connettività aziendale, voce dati;
- gestione tecnico operativa per il funzionamento, la manutenzione e la connettività delle reti di monitoraggio dell'azienda.


Tabella 1.1: mansioni dipartimento Servizio Informatico e Reti

1.1 Cosa Offre: Prodotti e Servizi

In questa sezione vengono elencati e descritti le tipologie di prodotti che ARPAV produce e i tipi diversi di servizi che offre, ponendo particolare attenzione a ciò che viene erogato dalla sede dello *stage*.

1.1.1 I Prodotti di ARPAV

Non essendo un'azienda a scopo di lucro, ma un'agenzia regionale, ARPAV è tenuta alla parità di bilancio. L'orientamento generale non è propenso alla distribuzione e vendita di prodotti quindi, ARPAV, per lo più, collabora con altre aziende o enti per realizzare progetti in ambito ambientale come *partner*, *subcontractor* o *leader*.

| Logo | Ruolo & Nome | Descrizione |
|---|---|---|
|  | <p>Ruolo: <i>Partner</i></p> <p>SedAlp: Programma Spazio Alpino</p> | <p>www.alpine-space.eu</p> <p>Sviluppo e <i>testing</i> di politiche e strumenti utili alla gestione del trasporto di sedimenti nei bacini alpini al fine di ridurre il rischio legato al trasporto solido e allo stesso tempo di migliorare la condizione ecologica degli ambienti acquatici e ripararli e ridurre l'impatto ambientale creato dalle centrali idroelettriche.</p> |

| Logo | Ruolo & Nome | Descrizione |
|---|---|--|
|  | Ruolo: <i>Leader</i> CAIMANs: Programma Med | www.medmaritimeprojects.eu/section/caimans Valutando l'impatto sulla qualità dell'aria da parte delle navi crociera e in generale delle navi passeggeri, il progetto mira a porre le basi per l'identificazione dei punti critici e per proporre orientamenti per futuri progetti e politiche transnazionali che affrontino la mitigazione dell'inquinamento atmosferico dovuto al traffico navale passeggeri. |
|  | Ruolo: <i>Partner</i> GuardEn: Programma South East Europe | www.southeast-europe.net/en/ Sviluppo e <i>testing</i> di un possibile quadro di riferimento finalizzato al supporto di un programma di implementazione di locali strategie per la gestione e prevenzione del rischio ambientale legato all'attività agricola e agroalimentare. In particolare per i territori interessati dall'inquinamento del suolo e dell'acqua, da proporre per l'applicazione alle aziende del settore. |
|  | Ruolo: <i>Partner</i> 3PClim: Programma Interreg IV Italia – Austria | www.interreg.net/it/programma/programma.asp Aggiornamento della climatologia delle Alpi orientali, con la produzioni di cartografie tematiche, elaborazioni e proiezioni climatiche. |
|  | Ruolo: <i>Partner</i> RE.S.M.I.A.: Programma POR – FESR Veneto | http://www.resmia.eu/ Progetto pilota di ricerca industriale con l'obiettivo di potenziare ed integrare la rete di monitoraggio ambientale a disposizione di ARPAV. |

Tabella 1.2: progetti ARPAV

Il dipartimento di informatica e reti, in collaborazione con CIVEN_g, durante il periodo stagistico stava seguendo con particolare attenzione **RE.S.M.I.A.**_g. Il progetto consiste nel potenziamento dell'infrastruttura delle stazioni di monitoraggio ambientale attualmente a

disposizione di ARPAV, con la progettazione ed installazione di sensori. Il nuovo concetto di stazione farà uso di tecnologia WSN_g, applicativi *Web-Based*_g e sarà caratterizzato dall'ottimizzazione della efficienza energetica del sistema.

Le finalità principali del progetto sono:

- **minimizzazione dell'impatto ambientale dall'installazione delle stazioni di monitoraggio:** utilizzo di tecnologie per l'auto-alimentazione dell'impianto e utilizzo di tecnologie *wireless* per ridurre al minimo l'invasione ambientale e con sperimentazione di nuovi sensori elettrochimici nanostrutturati per il monitoraggio in sito di metalli pesanti;
- **riduzione dei costi di produzioni ed installazione:** utilizzo di nuovi *hardware* a basso costo con ottime prestazioni, bassa necessità di manutenzione e semplice installazione, anche in presenza di condizioni morfologiche del territorio estremamente critiche. Realizzazione di prodotti in grado di essere installati da qualunque persona, in particolare volontari sprovvisti di preparazione;
- **tutela dell'ambiente:** monitoraggio delle matrici ambientali anticipando anche le disposizioni legislative e ponendo attenzione alle tematiche sentite dall'opinione pubblica in tema di salute, quale ad esempio il monitoraggio di nanoparticelle in aria;
- **prevenzione dei rischi:** determinazione delle soglie minime di allarme dei parametri ambientali reperibili in tempi brevi per l'intero territorio della Regione del Veneto;

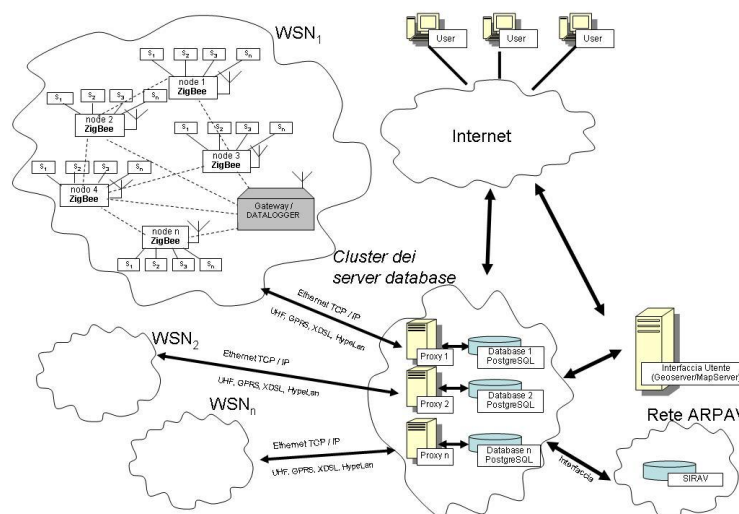


Figura 1.2: schema reti RESMIA

Il dipartimento di informatica e reti ha il compito di incrementare la struttura del sistema di salvataggio dei dati e di potenziare l'interfaccia *web* che rappresenta graficamente su un'opportuna mappa la dislocazione dei sensori e ed effettuare opportune interrogazioni sui dati forniti tramite *Map Server_g*.

1.1.2 I Servizi di ARPAV

L'agenzia regionale offre una vasta gamma di servizi, i quali possono essere suddivisi in due macro categorie: servizi ambientali e servizi online. I primi, offrono un servizio su richiesta o erogati da ARPAV, i secondi invece sono servizi passivi offerti dal portale *internet*, i cui fruitori possono accedervi tramite *network*. www.arpa.veneto.it/servizi-ambientali

Servizi Ambientali

| Nome | Possibili Fruttor | Descrizione |
|--|--|--|
| Acquisti pubblici verdi-GPP_g | Pubbliche amministrazioni locali o nazionali | Informazione delle pubbliche amministrazioni circa l'adozione di pratiche d'acquisto verdi che riducono l'uso di risorse naturali, la produzione di rifiuti, i rischi ambientali |
| Certificazioni ambientali | Imprese soprattutto piccole e medie | Diffusione all'interno del mondo produttivo di una nuova cultura di sistema per la gestione consapevole ed ecocompatibile dell'ambiente attraverso lo sviluppo di progetti, strumenti, protocolli ad hoc |
| Comunicazione | Cittadini | Promozione delle attività di educazione ed informazione ambientale dei cittadini |
| Progetti & Cooperazioni | Aziende, imprese, enti pubblici o regioni | Avvio e realizzazione di progetti, avvio di relazioni internazionali generalmente finanziati da fondi dell'Unione Europea |

| Nome | Possibili Fruttor | Descrizione |
|--|--|--|
| Grandi opere | Aziende coinvolte in appalti pubblici in Veneto | attività di audit preventivo e di monitoraggio ambientale per garantire la compatibilità ambientale, il corretto inserimento dal punto di vista urbanistico, ambientale, trasportistico e sociale delle Grandi Opere |
| Educazione per la sostenibilità | Chiunque | Attività di educazione, informazione e comunicazione ambientale, protezione della natura al fine di promuovere e sviluppare comportamenti sostenibili |
| IPPC_g e Servizi alle aziende | Aziende ed Imprese | Consulenze sul piano di monitoraggio e controllo in fase istruttoria per il rilascio dell'autorizzazione integrata ambientale, ispezioni integrate ambientali nelle aziende IPPC del Veneto |
| Pronta disponibilità | Dipartimenti di Prevenzione delle ULSS regionali Organi di polizia giudiziaria | Attività di analisi immediata di aria, acqua e suolo secondo le modalità previste |
| Rischio industriale | Industrie | Individuazione, classificazione e probabilità dei pericoli provenienti dalle industrie che utilizzano o detengono sostanze chimiche per le loro attività |
| Sicurezza impiantistica | Comuni, ASL, Prefettura, Procura | Verifica della corretta funzionalità di impianti e macchinari installati in ambienti di lavoro o di vita e soggetti a controlli periodici |

Tabella 1.3: servizi ambientali ARPAV

Servizi Online

| Nome | Descrizione |
|--|--|
| Accesso informazioni ambientali | Accesso del pubblico all'informazione ambientale detenuta o prodotta da soggetti pubblici avviene anche mediante l'utilizzo delle tecnologie informatiche e dei mezzi di telecomunicazione |

| Nome | Descrizione |
|--|--|
| Glossari Ambientali | Strumento di informazione aggiornata ed esaustiva che ARPAV mette a disposizione dei cittadini per favorire la comprensione di termini 'ambientali' maggiormente utilizzati |
| Iscrizione bollettini | Iscrizione alla mailing list consente di ricevere i bollettini Meteo direttamente nella propria casella di posta elettronica. Meteo Veneto, Dolomiti Meteo, Meteo Spiagge e Meteo Garda i bollettini per cui è disponibile il servizio |
| Iscrizione bollettini via sms | Sottoscrivendo un abbonamento è possibile ricevere via sms i contenuti di alcuni bollettini prodotti per l'area delle Dolomiti. Dolomiti meteo e Dolomiti Neve e Valanghe i bollettini per i quali è disponibile il servizio |
| Iscrizione newsletter | E' possibile ricevere periodicamente nella propria casella di posta le <i>newsletter</i> con informazioni su eventi, contenuti e attività |
| Iscrizione applicativo web ORSO | Programma per il monitoraggio del flusso dei rifiuti attraverso le Regioni d'Italia, con standard di riferimento comuni che garantiscano rappresentatività delle informazioni raccolte, oltre ad agevolare lo scambio di informazioni finalizzato alla corretta gestione dei rifiuti per i Comuni e gestori degli impianti |
| Iscrizione a IRRIFRA-ME | Servizio che permette alle aziende registrate di salvare il proprio profilo colturale e di personalizzare l'informazione irrigua fornita dal servizio comunicando in tempo reale dati locali |
| Link utili | Una rassegna di 'siti utili'; la suddivisione per argomenti e temi permette di individuare facilmente i riferimenti cercati |
| Richiesta pubblicazioni | Possibilità di richiedere una copia delle pubblicazioni editate da ARPAV attraverso posta elettronica e la compilazione di un modulo |

Tabella 1.4: servizi online di ARPAV

1.2 Organizzazione Interna

ARPAV è un'agenzia regionale che lavora per il completamento di progetti a livello regionale, nazionale ed internazionale ed offre tutta una serie di servizi a livello regionale e non. Per fare ciò necessita di una organizzazione interna ben strutturata, suddivisa in vari organi operativi, ognuno dei quali ricopre una funzionalità specifica e si interfaccia con un'altra secondo protocolli prestabiliti. ARPAV è dotata di una autonomia interna negli ambiti amministrativi, organizzativi e tecnico contabile e di diverse figure professionali che garantiscono un approccio multidisciplinare.

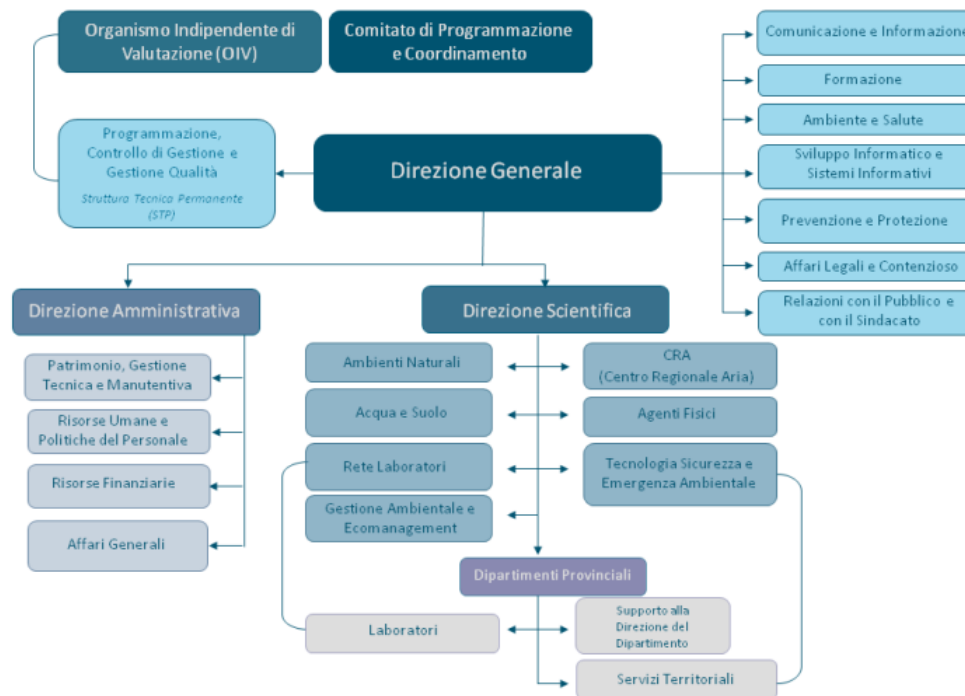


Figura 1.3: organigramma struttura ARPAV

A livello macroscopico è composta da una **Direzione Generale**, che a sua volta si ramifica in più aree funzionali di natura amministrativa e tecnico-scientifica, due **Dipartimenti regionali** e sette **Dipartimenti provinciali**. I dipartimenti regionali e provinciali per la realizzazione dei programmi ed attività di competenza godono di una autonomia gestionale nei limiti delle risorse loro assegnate dalla direzione generale.

La sede dello stage si trova all'interno dell'area della **direzione amministrativa**, in particolare nel sottoinsieme della **gestione tecnica**.

L'organizzazione interna al dipartimento di informatica è riassumibile così:

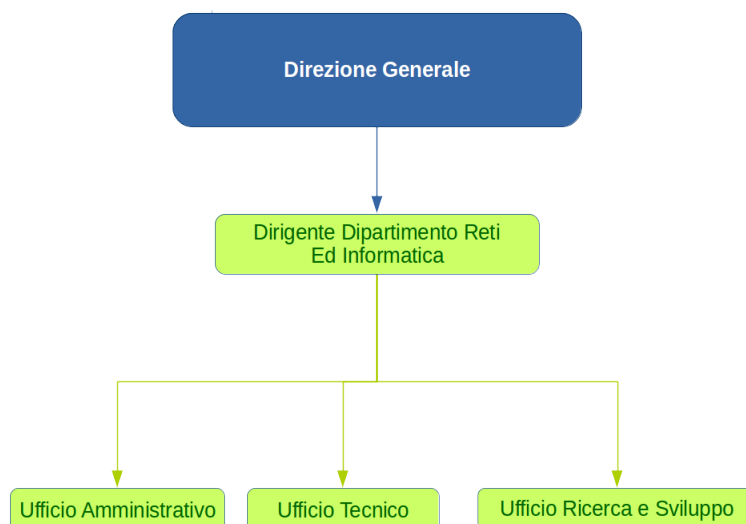


Figura 1.4: organigramma Dipartimento Informatica e Reti

- **Direttore:** figura a cui tutti fanno riferimento, ha il compito di relazionarsi con le aziende esterne per accordare collaborazioni nei progetti assegnati dalla **Direzione Generale**. Deve anche approvare i bilanci economici e gestire le risorse umane;
- **Ufficio Amministrativo:** ufficio con il compito di adempiere alle funzioni amministrative del dipartimento;
- **Ufficio di Ricerca e Sviluppo:** ufficio esecutivo in cui si sviluppano o si fa manutenzione sui progetti;
- **Ufficio Tecnico:** ufficio composto da tecnici addetti alla manutenzione del sistema di reti dell'intero ente.

1.2.1 Processi di Sviluppo

I processi di sviluppo all'interno di ARPAV non possono essere facilmente standardizzati, in quanto ogni settore operativo ha necessità di agire in modo differente rispetto agli altri. Ogni

dipartimento organizza i propri processi in base alla propria funzione e necessità, restando conforme alle linee guida dettate dalla **Direzione Centrale** tramite i seguenti documenti:

- **Piano Triennale Regionale di Educazione Ambientale**: documento di validità triennale, aggiornato annualmente, in cui si definiscono le linee guida dell'agenzia, gli obiettivi da raggiungere e le metodologie di crescita;
- **Documento di Programmazione per l'Informazione, la Formazione e l'Educazione Ambientale (IN.F.E.A.)**: documento in cui vengono definiti provvedimenti, progetti (di durata pluriennale) e le azioni che mirano all'adempimento degli obiettivi prefissati dal Piano Triennale di Educazione Ambientale;
- **Programma Triennale per la Trasparenza e Integrità**: documento di validità triennale, aggiornato annualmente, in cui vengono definiti i processi per garantire un livello di trasparenza, legalità e sviluppo della cultura dell'integrità.

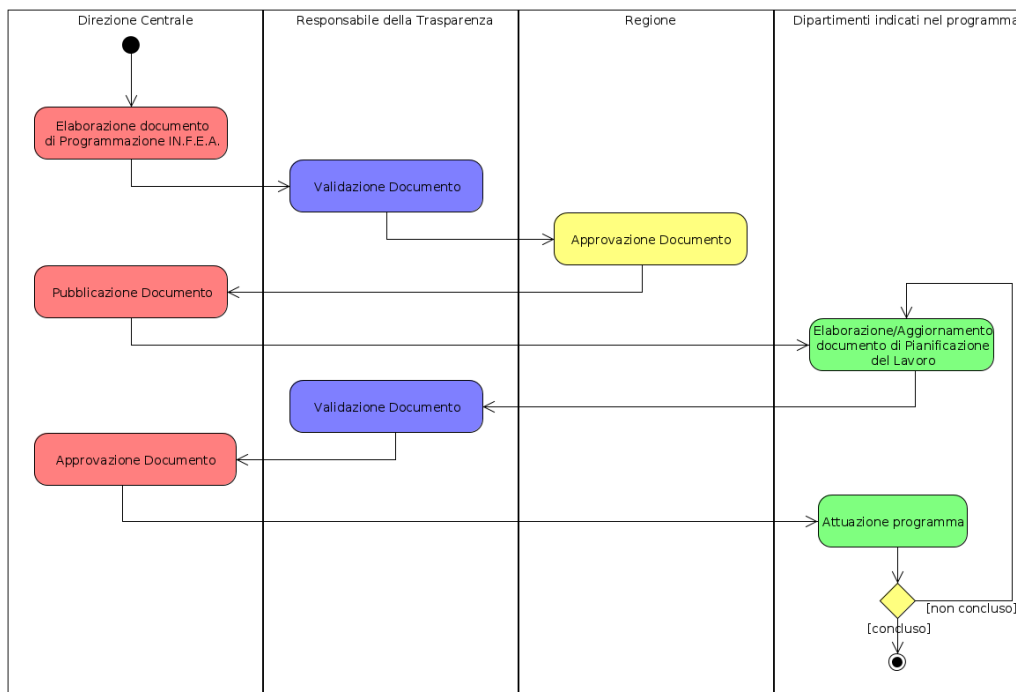


Figura 1.5: diagramma attività approvazione progetti

Nel documento di Programmazione IN.F.E.A. vengono determinate le finalità che i progetti triennali intendono perseguire. E' compito dei dipartimenti a cui è stata delegata la

realizzazione del progetto, dichiarare la gestione del *budget*, gli interventi di aziende private esterne e la pianificazione degli obiettivi intermedi, tramite la elaborazione, o aggiornamento, di un documento denominato *Pianificazione del Lavoro*. Al suo interno vengono definite le *milestone* che si intendono raggiungere alla fine di ogni anno e, in particolare, la suddivisione del lavoro su base annuale. Il documento viene quindi aggiornato in tre occasioni:

- Inizio anno: vengono definiti i traguardi che si realizzeranno durante l'anno in corso, le ore preventivate per attività e il *budget* che si presume di spendere;
- Dopo sei mesi: viene aggiornato lo stato di avanzamento del progetto con relative modifiche;
- Fine anno: viene aggiornato lo stato di avanzamento del progetto con relative modifiche del piano di lavoro triennale e ci si predispone la stesura del successivo anno di lavoro o la conclusione del progetto stesso.

Ogni incremento del documento viene visionato dal responsabile della trasparenza e approvato dalla Direzione Centrale.

In particolare, nel dipartimento sede dello *stage*, dovendo lavorare su progetti informatici ampi e complessi, l'approccio utilizzato è di tipo *top-down*. Progetti che coinvolgono l'impiego di risorse per periodi duraturi, richiedono una accurata e solida conoscenza delle problematiche che si devono affrontare e, in particolare, una robusta ed accurata progettazione. Tramite questo processo si suddivide un problema complesso in sottoproblemi più semplici. Ogni sottoproblema è descritto in modo dettagliato e, se complesso, a sua volta può essere suddiviso ulteriormente.

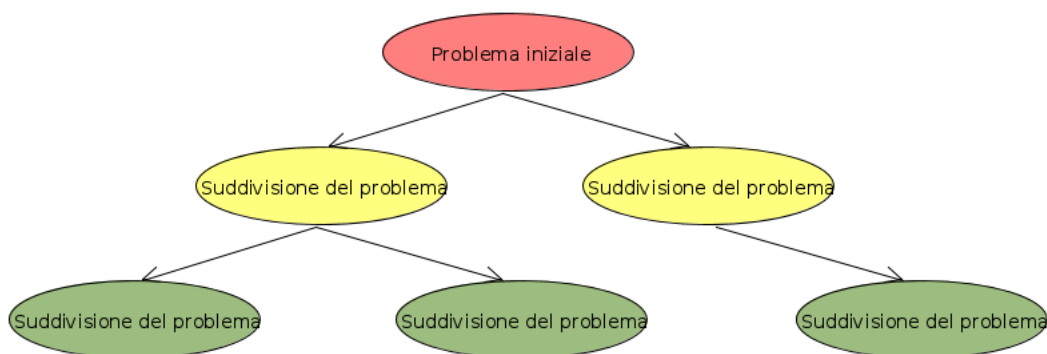


Figura 1.6: raffigurazione divisione problema approccio top-down

Vantaggi *top-down*:

- **Semplificazione del problema:** la suddivisione di un problema complesso in più sottoproblemi permette di trovare soluzioni efficientemente;
- **Suddivisione dei compiti:** la suddivisione dei problemi permette una programmazione parallela dei sottoproblemi indipendenti fra loro;
- **Programmazione semplificata:** una volta conclusa la programmazione, l'attività di programmazione risulterà semplice poiché si limiterà alla trascrizione in codice della progettazione ampiamente dettagliata;
- **Comprensione più semplice:** dividendo il problema in più parti rende più semplice la comprensione del lavoro, così da facilitarne una manutenzione futura.

Svantaggi *top-down*:

- **Fase di test:** non esistendo un eseguibile fino alla quasi conclusione del progetto, eseguire test può risultare complicato;
- **Scarse relazioni:** un approccio *top-down* sfavorisce le interazioni con gli *stakeholder*, in quanto devono attendere fino alle ultime fasi del progetto per visualizzare prototipi o bozze del progetto.

Il modello di ciclo di vita del *software* utilizzato è un modello a cascata, nel quale si prediligono delle fasi ben distinte di lavoro. Ogni fase ha un *input* di inizio e un *output* di fine, il quale è l'*input* fase successiva.

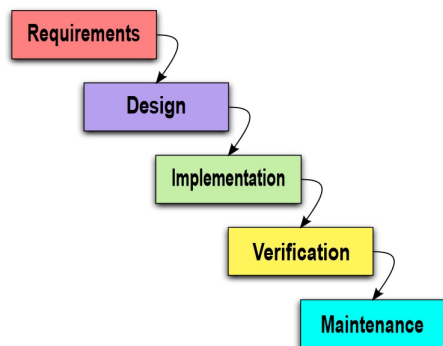


Figura 1.7: ciclo di vita a cascata

Dalla presentazione di un problema o un'opportunità (*input*), si inizia una fase di *analisi*, nella quale si studiano a fondo i problemi presentati e si produce un documento di *Analisi dei Requisiti* (*output*). Successivamente si procede con la progettazione, finita la quale, si inizia con la codifica e test. Conclusa quest'ultima fase si passa alla verifica e validazione ed infine, con il rilascio del prodotto, si procede con la manutenzione.

1.2.2 Metodologie di Supporto ai Processi

Di seguito vengono riportate le metodologie del dipartimento sede *stage*.

Analisi dei Requisiti

In questa fase si prende visione del o dei progetti assegnati dalla Direzione Centrale tramite il documento di Programmazione IN.F.E.A. . Gli analisti, interni o in caso di necessità esterni, sono gli attori principali per tutto questo periodo. Vengono studiati attentamente i capitolati e riportati tramite diagrammi *use case* le funzionalità riscontrate. In questa fase vengono effettuati studi per la ricerca di *partner* nel progetto. Eventualmente possono essere eseguiti rilievi su campo per approfondire ulteriormente i dati ambientali in possesso. Il responsabile del dipartimento ha il compito di visionare questa fase scrupolosamente, poiché dovrà approvare o meno scelte che graveranno sul totale del bilancio, ponendo molta attenzione a non sfiorare il *budget* a disposizione. Se necessario la fase di analisi può essere delegata a terzi.

Progettazione

Per la progettazione viene prima identificata una architettura generale ad alto livello che viene poi sviluppata in maggior dettaglio. Questa fase è cruciale per la riuscita del progetto. Il responsabile del dipartimento, se ritenuto necessario, può richiedere l'intervento di aiuti esterni per consultazioni specifiche.

Realizzazione e Test

Durante questa fase il responsabile del dipartimento ha un ruolo marginale, in quanto il lavoro di codifica non richiede particolare consumo di risorse, e si limita all'assegnazione dei compiti attraverso un sistema di *ticketing* per favorire una codifica in parallelo. Durante questa fase sono richiesti molti stagisti.

Manutenzione

Gran parte del lavoro che avviene all'interno del dipartimento di reti e informatica è di manutenzione. Vengono aperti costantemente *ticket* per la manutenzione di vecchi progetti e servizi. Il responsabile del dipartimento ha il compito ascoltare relative lamentele da parte degli usufruttuari dei servizi o prodotti, organizzare riunioni col team per la risoluzione dei problemi riscontrati e la suddivisione dei lavori di manutenzione.

1.2.3 Strumenti di Supporto ai Processi

Gestione Documenti

La documentazione di analisi dei requisiti e di progettazione è raccolta all'interno di *repository* gestite dai loro *server*, accessibili attraverso loro *network*. Per altri documenti vengono usati servizi di *Google* come *Google Drive*.

Ticketing

Durante la mia presenza in sede di *stage*, ho notato che gli strumenti di *ticketing*, per quanto venissero utilizzati per la maggior parte delle comunicazioni, era spesso necessario organizzare riunioni straordinarie avvisando gli interessati tramite comunicazioni via *mail*.

Gestione Versionamento

A supporto della fase di codifica viene utilizzato *GitHub* o *Bitbucket*. A seconda della tipologia del progetto possono essere necessarie *repository* private, quindi a pagamento, o pubbliche.

Gestione Calendario

Per la gestione delle attività di gruppo, come riunioni ufficiali per il resoconto delle attività o la risoluzione di conflitti interni, viene utilizzato lo strumento *Google Calendar* e il gruppo di *mailing list* interni ai *server* ARPAV per comunicazioni importanti.

1.3 Relazioni Esterne

Essendo un'agenzia regionale finalizzata al monitoraggio e alla tutela dell'ambiente, ARPAV non ha un vero e proprio *target* di clientela.

A seconda del ruolo che l'agenzia ricopre in un progetto (*partner* o *leader*) può collaborare con altre agenzie o aziende per fornire prodotti o migliorare infrastrutture. Altrimenti, attraverso i suoi servizi, può interfacciarsi con le singole imprese, industrie e privati cittadini.

1.3.1 Orientamento all'Innovazione

Il ruolo di ARPAV si può sintetizzare nelle seguenti parole chiave:

- Promozione e sostegno delle attività di informazione ed educazione ambientale dei cittadini, attraverso:
 - Coordinamento delle iniziative a livello regionale per la realizzazione di una rete di soggetti e di riferimenti, con lo scopo di ricercare sinergie ed economie di scala;
 - Formazione dei progettisti di azioni educative e dei formatori/educatori;
 - Monitoraggio e valutazione degli interventi;
 - Accreditamento di progetti di educazione ambientale.
- Gestione delle iniziative di educazione ambientale, attraverso:
 - Gestione diretta di iniziative di formazione e di educazione ambientale;
 - Compartecipazione ad iniziative gestite da altri soggetti attraverso diverse modalità;
 - Diffusione e divulgazione delle informazioni ambientali.

La *mission* di ARPAV si può quindi riassumere in:

Promuovere la crescita culturale della comunità regionale in termini di conoscenza, capacità, attitudini, motivazioni ed impegno morale, con lo scopo di contribuire individualmente e collettivamente a sostenere un ambiente salutare

Tratto Piano Triennale Regionale di Educazione Ambientale 2001-2003

Fare educazione ambientale e non semplice didattica necessita di focalizzare obiettivi e metodologie da adottare ben precisi. L'Agenzia, per entrare in azione nel modo più efficiente ed efficace possibile, adotta un piano articolato che risponde alle politiche di breve e medio periodo.

Le principali motivazioni che hanno portato l'Agenzia a questa scelta di fondo sono:

- Avere una visione globale;
- Condividere obiettivi comuni;
- Ricercare l'integrazione e le sinergie tra i numerosi soggetti attori, ognuno con una sua specificità ed un suo ruolo preciso;
- Distribuire in maniera ragionata ed efficiente gli interventi.

ARPAV utilizza un processo di controllo per il continuo aggiornamento dei propri piani d'azione.

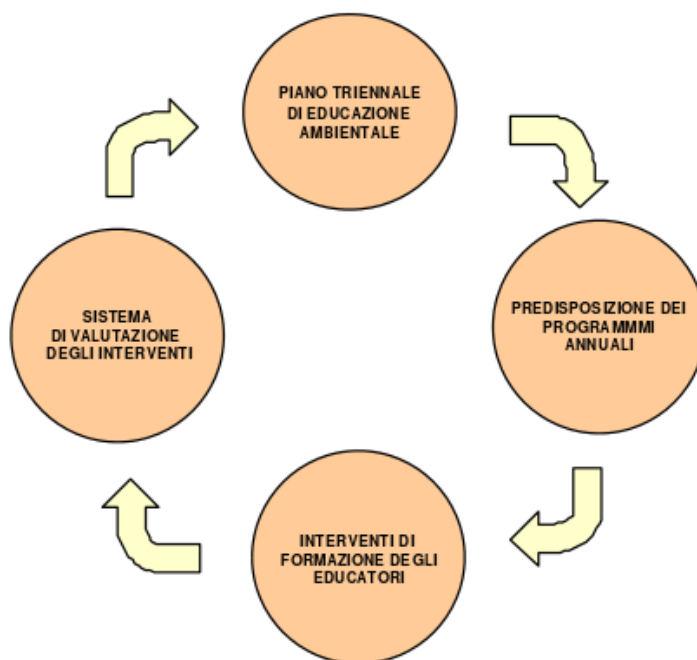


Figura 1.8: Ciclo di aggiornamento piano d'azione ARPAV

- **Predisposizione del piano triennale di educazione ambientale:** la redazione di un piano triennale, con caratteristiche di piano-processo, soggetto quindi a verifiche periodiche ed ai successivi adeguamenti, implica l'identificazione dei bisogni educativi della popolazione veneta e la conseguente scelta di obiettivi e strategie mirati a perseguire gli stessi. Tale operazione consente di garantire il massimo risultato delle azioni ARPAV perché identifica ed analizza le stesse in termini di costo-beneficio. Inoltre, prevede l'identificazione delle iniziative presenti sul territorio veneto, al fine di favorirne il coordinamento, nonché il coinvolgimento dei vari soggetti, stimolando il loro contributo in forma univoca e mirata ad identificati obiettivi;
- **Predisposizione dei programmi di attività annuali:** il piano triennale conterrà le linee guida per la predisposizione dei programmi di attività annuali, che consentono visibilità e trasparenza delle scelte effettuate e l'integrazione di tutte le energie ARPAV verso traguardi esplicitati. Nel documento vengono descritti progetti e azioni mirate sul campo che prevedono l'attuazione del piano di educazione;
- **Interventi di formazione dei progettisti di azioni educative e di formatori/educatori:** è importante garantire la preparazione di operatori che potranno in tutto il territorio regionale progettare interventi educativi in modo corretto e coerente con l'impostazione ARPAV. Ciò garantirà il massimo beneficio degli interventi stessi perché metodologicamente corretti ed integrati tra di loro. L'acquisizione di abilità metodologiche va considerato come un investimento perché tali operatori potranno incidere per un lungo tempo a venire.
- **Creazione di un sistema di monitoraggio e valutazione degli interventi:** Il piano prevede la creazione di un sistema di monitoraggio e valutazione degli interventi che verranno realizzati: indicatori, sistema informativo, criteri di valutazione, rapporti e loro destinatari.

Capitolo 2

Lo Stage per ARPAV

Il dipartimento di reti ed informatica deve affrontare la difficile sfida di mantenere la parità del bilancio, senza rinunciare alla qualità delle loro opere. Il responsabile del dipartimento, spinto da questa necessità, ha orientato la politica di scelta del personale, durante la fase di codifica, verso stagisti neo-laureati o laureandi. Questa scelta permette:

- **maggior efficacia:** l'utilizzo di stagisti permette di ottimizzare le risorse di bilancio, diminuendo l'impatto sul *budget*;
- **maggior efficacia:** menti fresche di studio e con bassa specializzazione permettono un'adesione a progetti nuovi ed innovativi con maggiore facilità rispetto a personale propenso a lavorare secondo processi prestabiliti.

2.1 L'Esigenza di uno Stage

Il dipartimento di reti ed informatica, al momento del mio inserimento, stava seguendo la progettazione e lo sviluppo di un pluviometro_g innovativo a basso costo da parte del *FabLab_g* di Verona. Questo incarico è stato proposto da parte di ARPAV per il progetto

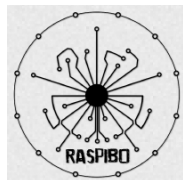


Figura 2.1: Logo RaspiBO



Figura 2.2: Logo FabLab

RE.S.M.I.A., il quale prevede il potenziamento dell'infrastruttura della rete di monitoraggio di ARPAV, tramite la progettazione di nuove stazioni meteorologiche. Contemporaneamente al progetto *FabLab*, il responsabile del dipartimento stava seguendo il *Gruppo Meteo* di *RaspiBO_g* il quale sta sviluppando stazioni meteorologiche a basso costo, al momento sprovviste di pluviometro. La combinazione di questi due fattori ha creato la necessità di sviluppo di un sistema *hardware* e *software* che permettesse la gestione di un pluviometro tramite tecnologie a basso costo, ma con prestazioni adeguate alle richieste.

2.2 Presentazione del Progetto

La *Small Project*, azienda italiana nata del 2004, appoggia il progetto *Arduino* e ne è diventata il produttore principale. Arduino è una scheda elettronica di piccole dimensioni, con un microcontrollore e circuiteria di contorno per la realizzazione di progetti o prototipi. Il progetto Arduino, ormai avviato da qualche anno, ora conta svariate tipologie di schede con caratteristiche di prestazioni e prezzo differenti fra loro. Queste schede permettono una facile programmazione e una elastica implementazione a seconda delle necessità che il proprio lavoro richiedono. Per il progetto di *stage*, è stato proposto l'utilizzo di una scheda *Arduino UNO* (www.arduino.cc/en/Main/ArduinoBoardUno).

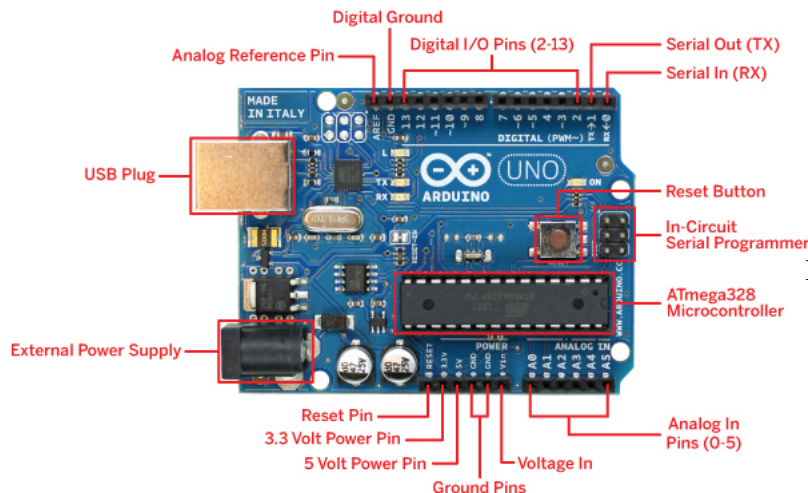


Figura 2.4: Logo Arduino

Figura 2.3: Scheda Arduino UNO

Il progetto prevede la progettazione e lo sviluppo di un dispositivo innovativo *embedded* per la gestione di un pluviometro, lo studio per una gestione e memorizzazione efficiente di

eventi di pioggia mediante standardizzazione su protocollo MQTT_g, sviluppato su *hardware* e *firmware* ATMEL_g.

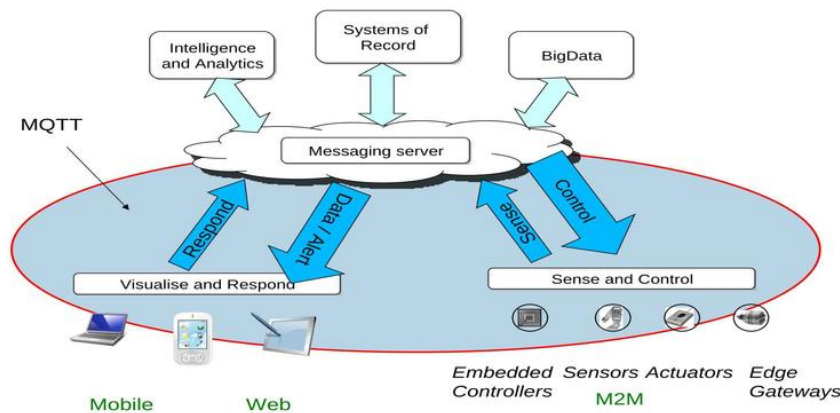


Figura 2.5: Esempio protocollo MQTT

Le componenti hardware saranno volutamente poco costose e quindi con limitate capacità di memorizzazione dati, pertanto sarà necessaria la ricerca di soluzioni che tengano in considerazione tale problematica. Nel progetto verrà richiesto l'utilizzo di due schede Arduino UNO, di cui una con lo scopo di gestione dei segnali ricevuti dall'*hardware* del pluviometro, salvataggio e gestione dei dati, l'altra con la funzionalità di interrogazione della prima. Si sono riscontrate così due tipologie di schede:

- **Scheda Slave:** scheda che si interfaccia direttamente con i segnali inviati dal pluviometro, che salva all'interno di una memoria i dati richiesti e li cancella in caso di necessità o richiesta;
- **Scheda Master:** scheda che permette l'interrogazione della *scheda Slave*, tramite protocollo prestabilito e rimbalzi così i dati ricevuti all'utente che ne ha fatto richiesta.

La *scheda Slave*, dovrà essere interamente progettata e realizzata all'interno del lavoro di *stage*, per quanto riguarda la *scheda master*, ciò che viene richiesto è la creazione di una libreria che permetta un facile accesso alle funzionalità di base richieste dal progetto e un'architettura tale che permetta facilmente integrazioni future. Viene richiesto in oltre che tutto progetto di *stage* venga rilasciato tramite *GNU GPL_g* su *github*, provvisto di documentazione dei metodi e componenti *software* associata per facilitare future modifiche o correzioni.

2.2.1 Obiettivo dello Stage

Gli obiettivi sono stati preposti durante i colloqui conoscitivi precedenti all'inizio dello *stage*, evidenziando quelle che sarebbero state le caratteristiche salienti sufficienti per considerare soddisfacente il lavoro completo del tirocinio.

- **Obiettivi esplorativi:** evidenziano il livello di capacità di indagine dello stagista nell'affrontare un ambiente nuovo in cui vengono utilizzate tecnologie mai affrontate durante il percorso accademico. Viene anche delineata l'attitudine ad analizzare i rischi legati all'utilizzo di componenti *hardware* inesplorati da parte dell'agenzia e la capacità di risoluzioni dei problemi legati alle limitate prestazioni delle schede Arduino;
- **Obiettivi funzionali:** includono i requisiti minimi richiesti delle funzionalità per considerare il lavoro di *stage* sufficiente alla propria conclusione. I requisiti descrivono le funzionalità che dovranno essere implementate nel progetto e rappresentano l'intento centrale di tutto il lavoro del tirocinio.
- **Obiettivi funzionali opzionali:** rappresentano l'insieme di requisiti che descrivono le funzionalità del progetto che non sono obbligatori per la completezza del lavoro di tirocinio, ma che rappresentano un valore aggiunto e desiderato da parte dei proponenti.

Obiettivi Minimi Richiesti

Di seguito vengono riportati gli obiettivi minimi che sono stati richiesti al momento di inizio *stage*:

- **Obiettivi esplorativi:**
 - *Acquisizione conoscenze elettronica di base:* dovendo affrontare un lavoro con schede *hardware*, connettendo cavi, creando connessioni con dispositivi ausiliari e correnti elettriche, è strettamente necessario acquisire le conoscenze di base che permettano di lavorare in sicurezza e senza causare guasti alle strumentazioni;
 - *Comprensione ambiente Arduino:* il mondo Arduino offre una vasta scelta di modi in cui operare, dalla tipologia di linguaggio, agli strumenti di sviluppo.
 - *Valutazione prestazione Arduino:* uno dei punti focali stabiliti ad inizio *stage* è stato comprendere il prima possibile se la scheda fornita fosse sufficientemente adeguata per le richieste del progetto. Pur non essendo un vero e proprio

obiettivo, mi è stato preposto come tale a causa della sua importanza, poiché la ricerca di un nuovo componente, o anche la sola acquisizione *online*, avrebbe provocato considerevoli ritardi.

- **Obiettivi funzionali:**

- *Progettazione e sviluppo scheda Slave:* punto focale del lavoro di *stage* è lo sviluppo della scheda connessa al pluviometro, che dovrà:
 - * Riconoscere gli impulsi inviati dal prototipo di pluviometro;
 - * Riconoscere segnali falsi positivi e non registrare dati infondati;
 - * Immagazzinare in una memoria i *timestamp*;
 - * Ottimizzare la gestione della memoria;
 - * Comunicare tramite *protocollo I2C_g* con le componenti a lei connesse;
 - * Comunicare tramite protocollo I2C con la componente Master a due vie;
 - * Ritorno dei dati memorizzati;
 - * Cancellazione dei dati;
 - * Tenere traccia di eventuali *overflow* dovuti alle basse prestazioni di memoria;
 - * Progettazione elastica che permetta l'inserimento futuro di nuovi protocolli di comunicazione o inserimento di nuovi componenti hardware;
- *Progettazione e sviluppo libreria Master:* secondo punto focale del progetto, sviluppo di una libreria che possa essere inserita in un contesto dove è già esistente una scheda Master a cui viene connesso il pluviometro con scheda Arduino integrata con il sistema Slave:
 - * Comunicazione tramite protocollo I2C con la scheda Slave del pluviometro;
 - * Progettazione elastica che permetta l'inserimento futuro di nuovi protocolli per la comunicazione con la scheda Slave;
 - * Richiesta per la ricezione dei dati inseriti in memoria della scheda Slave;
 - * Richiesta per la ricezione dei dati ancora non letti;
 - * Richiesta per la cancellazione dei dati già letti;
 - * Richiesta per la cancellazione dell'intera memoria della scheda Slave;

Obiettivi Massimi Raggiungibili

Di seguito vengono riportati gli obiettivi preposti ad inizio *stage*, che non sono vincolanti alla conclusione dello stesso:

- **Obiettivi esplorativi:**

- *Acquisizione conoscenze elettronica semi-avanzate*: imparare a formare uno schema elettronico di una scheda in modo da realizzare una componente *hardware* nuova partendo da componenti separate e l'utilizzo di strumenti che ne permettano l'assemblo.

- **Obiettivi funzionali opzionali:**

- *Implementazione del codice su scheda Arietta_g* : adattamento, implementazione e test del codice precedentemente approvato su Arduino UNO in una scheda Arietta;
- *Interfaccia di configurazione e ricezioni dati web o desktop*: progettazione e realizzazione di un *tool web* o *desktop* per la relazione diretta *computer/web-scheda* Master;
- *Comunicazione 3G con scheda Master*: implementazione di un dispositivo per la comunicazione via 3G tra dispositivo *mobile* e la scheda Master;

2.2.2 Finalità del Progetto



Figura 2.6: Prototipo pluviometro FabLab

Nella realtà ARPAV il suddetto lavoro di *stage*, oltre ad integrare una parte mancante del progetto RE.S.M.I.A., trova largo margine di prospettive future. Il progetto verrà rilasciato con licenza di utilizzo GNU GPL, quindi liberamente utilizzabile da chiunque. FabLab di Verona ha già lanciato un progetto parallelo successivo al lavoro di *stage*: *Tutti misurano, OpenPluvio*. Il progetto riprende il lavoro portato a termine e lo continua per introdurre un sistema di misurazione ambientale casalingo, amatoriale, che possa inviare i dati delle misurazioni effettuate a *serve* dedicati. Lo scopo è facilitare il possesso di strumentazione di

controllo ambientale a basso costo e la creazione una nuova rete di monitoraggio che possa integrare o arricchire i dati delle reti ufficiali.

2.3 Vincoli di Progetto

In questa sezione verranno descritti i vincoli del progetto che sono stati stabiliti ad inizio *stage*, catalogati nel seguente modo:

- **Vincoli di dominio:** descrivono i vincoli di funzionalità, affidabilità e manutenibilità;
- **Vincoli tecnologici:** descrivono gli strumenti di lavoro e le tecnologie obbligatorie;
- **Vincoli metodologici:** descrivono le metodologie che imposte durante fase di *stage*;
- **Vincoli temporali:** descrivono tappe obbligatorie da seguire durante l'attività lavorativa;

2.3.1 Vincoli di Dominio

Il sistema da realizzare dovrà immagazzinare una discreta quantità di dati in uno spazio di memoria ristretto. La necessità di sviluppare un algoritmo di compressione dei dati sarà estremamente necessario. I dati che dovranno essere immagazzinati vengono definiti *timestamp*. I *timestamp* sono un *metadato* composto da più elementi:

- **Data:** nel formato gg / mm / aaaa (esempio 12 05 2014);
- **Ora:** nel formato hh / mm / ss (esempio 12 02 35);
- **Contatore:** un numero che deve identificare il numero di *timestamp* dall'ultima lettura effettuata.

Ogni *timestamp* verrà salvato in memoria successivamente all'interruzione di una basculata da parte del pluviometro. Una basculata avviene quando la vaschetta della bilancia viene riempita fino ad un livello tale che il peso dell'acqua fa cadere la vaschetta. A seconda della tipologia di bilancia il segnale può essere inviato diversamente.

Durante il lavoro di *stage* verrà usato un pluviometro ancora sperimentale, quindi lo strumento di misurazione della bilancia potrebbe risultare instabile e produrre dei segnali

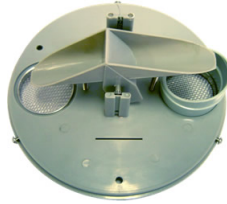


Figura 2.7: Bascula di un pluviometro

falsi positivi, con il risultato di immagazzinare erroneamente dei *timestamp*. Uno dei vincoli principali del sistema dovrà essere l'affidabilità dei dati, i quali dovranno essere affidabili al 100%.

Il codice prodotto verrà rilasciato con licenza d'uso GPL, questo richiede che il codice sia chiaro e mantenibile. Per questo motivo è richiesta la presenza di documentazione che descriva le componenti *software* e i metodi che contengono.

2.3.2 Vincoli Tecnologici

In questa sezione vengono riportate le tecnologie richieste per il completamento del lavoro stagistico.

- **Arduino IDE:**

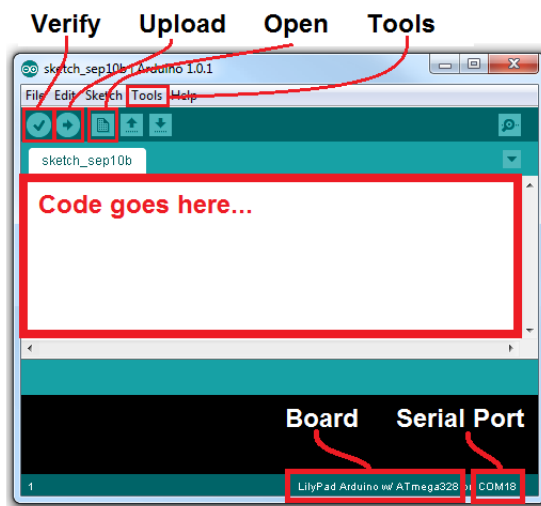


Figura 2.8: Esempio di IDE Arduino

Arduino IDE è un semplice *framework* di programmazione multi-piattaforma che permette di caricare nella scheda Arduino il codice sorgente che si vuole implementare. E' il *software* ufficiale fornito da Arduino e permette un utilizzo semplice della piattaforma. L'interfaccia dell'IDE permette di lavorare su di uno *sketch* nel quale si deve caricare tutto il codice che si intende implementare nella *board*. Sono obbligatorie le dichiarazioni di due metodi fondamentali:

- **void setup()** : metodo che viene evocato ogni volta che la scheda viene riavviata, all'interno del quale devono essere dichiarate le modalità di utilizzo dei *pin* connessi alla scheda;
- **void loop()** : metodo che viene ripetuto continuamente dalla scheda in un *loop* infinito, finché viene mantenuta energia o non viene effettuato il *reset*.

La sua semplicità rende questo strumento anche altrettanto limitato nella flessibilità di utilizzo. Per organizzare il codice in maniera strutturata per classi è necessario l'utilizzo di uno strumento secondario. L'IDE Arduino infatti permette la compilazione e l'*upload* del codice presente solo nello *sketch*. Senza l'utilizzo di un *framework* esterno, bisognerebbe dichiarare tutte le classi all'interno di un unico *file*.

- **Arduino UNO:**

Come descritto in precedenza mi è stato richiesto l'utilizzo di una scheda Arduino UNO. Le specifiche tecniche di questa *board* sono:

- *Microcontroller*: ATmega328;
- *Operating Voltage*: 5V;
- *Input Voltage (recommended)*: 7-12V;
- *Input Voltage (limits)*: 6-20V;
- *Digital I/O Pins* : 14 (of which 6 provide PWM output);
- *Analog Input Pins*: 6;
- *DC Current per I/O Pin*: 40 mA;
- *DC Current for 3.3V Pin*: 50 mA;
- *SRAM*: 2 KB (ATmega328);
- *EEPROM*: 512 Byte (ATmega328);
- *Clock Speed*: 16 MHz;
- *Length*: 68.6 mm;
- *Width*: 53.4 mm;
- *Weight*: 25 g.

La scelta di questa scheda sono ricadute per i seguenti motivi:

- *Affidabilità*: il prodotto 100% italiano ha alle spalle una lunga storia di successi che hanno reso questo prodotto famoso per la sua affidabilità e robustezza;
- *Feedback utenti*: poiché il prodotto è in commercio da qualche anno, i numerosi utenti che ne hanno fatto utilizzo hanno creato una solida rete di *feedback* e risoluzione dei problemi;
- *Riutilizzo software*: queste schede sono già state ampiamente usate per uno svariato numero di progetti con licenza gratuita, questo comporta la facile reperibilità *online* di codice utile;
- *Componenti esterne*: molti produttori di componenti *hardware* esterni, come sensori o schede di interfacce, forniscono librerie di interfacce perfettamente funzionanti e testati con l'ambiente Arduino;
- *Prezzo*: il prezzo di una scheda Arduino si aggira sui 20 euro, mentre il microcontrollore ATmega328 dai 3 ai 5 euro. Se si utilizzassero solamente schede di questo tipo per il progetto, il risultato finale non risulterebbe economico. Durante la fase di codifica verranno usate due schede Arduino UNO per velocizzarne lo sviluppo, invece, il prodotto finale sarà composto da *board* personalizzata che utilizzerà un microcontrollore con il codice già caricato, compatta ed economica.

- **Tiny-RTC (Real Time Clock Module)**

Componente *hardware* per la misurazione del tempo. *Tiny-RTC* non è altro che un modulo *hardware* con la funzionalità di orologio configurabile tramite *sketch* Arduino e provvisto di protocolli di I2C per l'interrogazione del modulo per il ritorno di data ed ora.

Il modulo deve essere connesso alla *board* Arduino tramite *bus I2C* come mostrato nella figura 2.8.

Specifiche tecniche:

- 3.0-5.5V input voltage
- Waterproof
- -55°C to +125°C temperature range
- $\pm 0.5^\circ\text{C}$ accuracy from -10°C to $+85^\circ\text{C}$
- 1 Wire interface

Descrizione delle connessioni:

- **BAT**: Battery voltage
- **GND**: Ground Ground
- **VCC**: Power the module and charge the battery

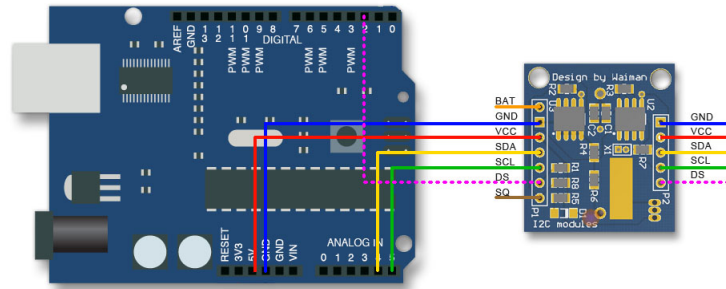


Figura 2.9: Tiny RTC connessa a scheda Arduino UNO

- **SDA**: I2C data for the RTC
- **SCL**: I2C clock for the RTC
- **DS**: Sensor output
- **SQ**: Square wave output

- Qt Creator



Figura 2.10: Logo Qt

Non mi è stato propriamente imposto l'utilizzo di *Qt-Creator*, ma dal momento che mi era stato vivamente consigliato e durante il mio iter universitario ho avuto occasione di utilizzarlo frequentemente, è stato concordato sin dall'inizio il suo utilizzo per la codifica del codice.

2.3.3 Vincoli Metodologici

Come metodologie di lavoro mi sono state lasciate molte libertà, fatta eccezione per:

- **Sistema di controllo di versionamento**



Figura 2.11: Logo GitHub



Figura 2.12: Logo Bitbucket

Il responsabile del dipartimento mi ha richiesto l'utilizzo di *git hub* come *repository* del mio codice per la parte riguardante ARPAV. Invece il responsabile di FabLab ha richiesto l'utilizzo di un *repository* di *bit-bucket* privato per il codice da utilizzare durante il periodo di *testing* dei dispositivi assieme al prototipo pluviometro.

- **Rilascio continuo di prototipi**

FabLab di Verona ha contemporaneamente molti progetti in corso d'opera, di cui buona parte assegnati sempre dal dipartimento di reti ed informatica di Padova. Con l'inizio del mio lavoro, l'officina ha dovuto spostare maggiormente il baricentro dell'attenzione sul progetto pluviometro. Per questo motivo mi è stato richiesto di organizzare il mio lavoro con modalità di progettazione e sviluppo attraverso *prototipi*.

- **Relazioni avanzamento lavoro**

Alla fine di ogni settimana ero tenuto ad aggiornare FabLab dei progressi ottenuti tramite una breve relazione da inviare via *mail* direttamente al responsabile dell'officina.

2.3.4 Vincoli Temporal

L'Università di Padova rende obbligatorio lo svolgimento del lavoro di *stage* nell'arco di 300 - 320 ore. Durante i colloqui conoscitivi è stato redatto un documento di pianificazione del lavoro dove si sancivano le ore di lavoro effettive e come queste saranno suddivise. Il lavoro di *stage* è stato dunque stabilito doversi svolgere in 320 nell'arco di 9 settimane. Un vincolo esterno importante da considerare è stata la necessità di avere sufficiente codice da avere un prototipo funzionante prima delle vacanze di Natale, da consegnare a *FabLab* di Verona per poter iniziare i primi *test* sul prototipo di pluviometro.

Capitolo 3

L'Attività di Stage

In questo capitolo viene raccontata l'esperienza di lavoro di *stage* e come questa sia stata portata a termine, tramite pianificazione e suddivisione delle attività. Il capitolo si sviluppa in quattro macro sezioni che rappresentano approssimativamente l'avanzamento cronologico di ogni attività svolta durante lo *stage*. Si intende approssimativamente poiché durante ognuna di queste attività, occasionalmente, o ripetutamente come nel caso di codifica e test, alcune mansioni venivano riprese e ripetute. Questo è stato necessario in particolar modo a causa dello sviluppo secondo prototipi, i quali richiedevano continui cicli di analisi, sviluppo e test.

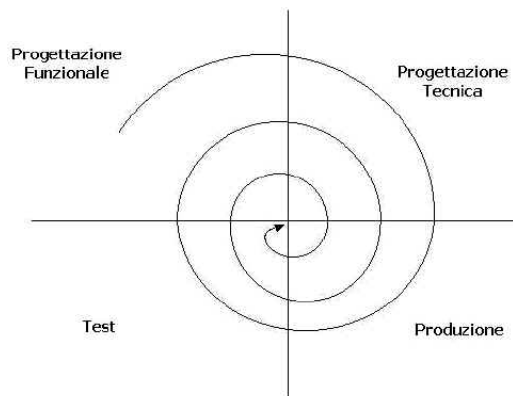


Figura 3.1: Modello a spirale

3.1 Pianificazione del Lavoro

Per conseguire gli obiettivi di *stage*, sono state preventivate circa 320 ore lavorative suddivise in 10 settimane durante l'arco temporale dal 17 Novembre 2014 al 30 Gennaio 2015, comprendente una pausa di una settimana per le vacanze natalizie. La suddivisione temporale effettuata a monte, assieme al responsabile interno ARPAV, è stata composta con una grana molto grossa, lasciandomi ampia libertà nell'organizzazione del lavoro a seconda delle mie necessità.

| Attività / Settimana | 1° | 2° | 3° | 4° | 5° | 6° | 7° | 8° | 9° | 10° |
|---|----|----|----|----|----|----|----|----|----|-----|
| <i>Analisi dei Re- quisiti</i> | 32 | 24 | / | / | / | / | / | / | / | / |
| <i>Progettazione Architetturale</i> | / | 8 | 32 | 20 | / | / | / | / | / | / |
| <i>Progettazione Dettagliata</i> | / | / | / | 12 | 25 | 10 | / | / | / | / |
| <i>Codifica</i> | / | / | / | / | / | 18 | 24 | 24 | 20 | 10 |
| <i>Testing</i> | / | / | / | / | 7 | 4 | 8 | 8 | 12 | 22 |

Tabella 3.1: Suddivisione del lavoro

Per ogni attività sono state preventivate le seguenti ore:

| Attività | Ore |
|-------------------------------------|-----|
| <i>Analisi dei Requisiti</i> | 56 |
| <i>Progettazione Architetturale</i> | 60 |
| <i>Progettazione Dettagliata</i> | 47 |
| <i>Codifica</i> | 96 |
| <i>Testing</i> | 61 |

Tabella 3.2: Ore per attività

Nella figura (3.2) vengono evidenziate le suddivisioni temporali effettuate per lo svolgimento dello *stage* e il susseguirsi delle attività. Le sezioni più chiare rappresentano i giorni non

lavorativi del fine settimana, mentre le sezioni evidenziate di nero raffigurano le vacanze di Natale.

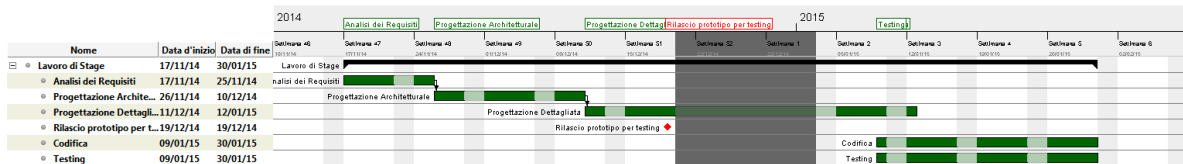


Figura 3.2: Diagramma di Gantt del periodo di Stage

3.1.1 Preparazione al Lavoro di Stage

In questa sezione illustro le modalità e gli esiti degli studi intrapresi per comprendere gli strumenti di lavoro. Le prime due settimane sono state suddivise in due sotto *task*:

- **Studio del Dominio:** periodo dedicato all'acquisizione delle conoscenze di base delle tecnologie da utilizzare;
- **Catalogazione dei Requisiti:** periodo dedicato allo studio dei requisiti e la catalogazione di quest'ultimi anche in relazione alle conoscenze appena acquisite.



Figura 3.3: Suddivisione lavoro prime due settimane

Studio del Dominio

L'universo Arduino si è affermato sul mercato ormai da diversi anni e ci sono molte tipologie di strumenti a supporto degli utenti neofiti e non. Prima dell'inizio dello *stage* il responsabile del dipartimento di reti ed informatica mi ha dotato del manuale di base Arduino (*"Il Manuale di Arduino"* Apogeo editore), grazie al quale ho potuto avvantaggiarmi acquisendo sufficienti conoscenze basilari che mi permettessero poi uno studio delle operazioni più complesse in sede di *stage*.



Figura 3.4: Manuale Arduino

Entrato in possesso della strumentazione e dei dispositivi *hardware* Arduino, mi sono affidato ai *tutorial* presenti nel sito di Arduino (<http://www.arduino.cc/en/Guide/HomePage>) e degli esempi contenuti all'interno degli *sketch* dell'IDE Arduino. Per la ricerca delle librerie esterne da utilizzare durante lo sviluppo del progetto ho ricercato soluzioni appropriate nel sito Arduino, ma nel caso queste non fossero sufficienti o non adatte alle mie esigenze, ho effettuato ricerche più accurate utilizzando *Google* come motore di ricerca.

Per la risoluzione di problemi, maggior parte delle volte, ho fatto uso del *forum* dedicato (<http://forum.arduino.cc/>) nel quale moltissimi utenti, che hanno già avuto a che fare con problematiche simili, hanno illustrato soluzioni funzionanti. Nel caso queste non fossero sufficienti ho richiesto aiuto tramite *mail* al responsabile del progetto del prototipo del pluviometro del FabLab di Verona.

Studio delle Tecnologie

- **Arduino UNO:**

Arduino UNO deve il suo nome perché trattasi della prima scheda rilasciata dalla casa. Le caratteristiche tecniche di questa scheda le permettono di supportare svariate tipologie di progetti con esigenze differenti. Per quanto riguarda il mio progetto, ciò che più mi ha preoccupato era la limitata capacità di memoria interna al microcontrollore ATmega328 di 512Byte. I dati da memorizzare, i *timestamp*, sono composti da più elementi:

- *Data*: giorno/mese/anno;
- *Ora*: ore/minuti/secondi;
- *Contatore*: rappresentante il numero di *timestamp* dall'ultima lettura.

Usando in totale 8 byte per ogni singolo *timestamp* (3Byte per data, 3Byte per l'ora e 2Byte per il contatore), il numero massimo di *timestamp* memorizzabili nella

EEPROM del microcontrollore si limitano a 64. Ho così deciso di comprimere i dati per sfruttare al meglio la memoria, utilizzando 6 byte invece che 8:

| Dato | Range valore | Non compresso | Compresso |
|------------------|------------------------|---------------|---------------|
| <i>Giorno</i> | 1-31 | 8bit | 5bit |
| <i>Mese</i> | 1-12 | 8bit | 4bit |
| <i>Anno</i> | 0-99 | 8bit | 7bit |
| <i>Ore</i> | 0-23 | 8bit | 5bit |
| <i>Minuti</i> | 0-59 | 8bit | 6bit |
| <i>Secondi</i> | 0-59 | 8bit | 6bit |
| <i>Contatore</i> | variabile ¹ | 16bit | 15bit |
| Totale | - | 64bit (8Byte) | 48bit (6Byte) |

Tabella 3.3: Valori *timestamp* compressione

| Byte | Composizione | | | | | | | | Simbolo | Valore |
|---------|--------------|----|----|---|---|----|----|----|-----------|-----------|
| 1° Byte | m | m | m | g | g | g | g | g | <i>g</i> | Giorno |
| 2° Byte | a | a | a | a | a | a | a | m | <i>m</i> | Mese |
| 3° Byte | mi | mi | mi | o | o | o | o | o | <i>a</i> | Anno |
| 4° Byte | s | s | s | s | s | mi | mi | mi | <i>o</i> | Ore |
| 5° Byte | c | c | c | c | c | c | c | s | <i>mi</i> | Minuti |
| 6° Byte | c | c | c | c | c | c | c | c | <i>s</i> | Secondi |
| | | | | | | | | | <i>c</i> | Contatore |

Tabella 3.4: Conformazione *bit timestamp*

Utilizzando un algoritmo di compressione che porti la struttura di un *timestamp* come descritto in tabella (3.4) si migliora la capacità della memoria da 64 *timestamp* salvabili a 85, lasciando disponibili anche 2Byte per il salvataggio di dati supplementari. Ciononostante, effettuando una indagine iniziale, ho scoperto che durante una pioggia di forte intensità può verificarsi una basculata ogni due secondi. Utilizzando una memoria così limitata poco dopo 2 minuti si avrebbe un *overflow* di memoria. Sottoposti questi dati al responsabile del dipartimento sono stati subito predisposti provvedimenti. Mi è stato proposto l'utilizzo di un *microchip* di memoria supplementare nel quale salvare i *timestamp* dalle capacità superiori.

¹ : il contatore deve avere un *range* di valori sufficientemente alto da non andare in *overflow*. Inizialmente avevo valutato 16bit (0-65535), ma anche 15bit (0-32767) comprendono un insieme di numeri accettabile.

- C++



Figura 3.5: Logo C++

La scelta del linguaggio, C++, è stata una scelta importante per lo sviluppo del progetto. Le motivazioni che mi hanno spinto a scegliere questo linguaggio sono le seguenti:

- **Memoria Arduino UNO:** la scheda Arduino UNO possiede una *memoria SRAM_g* di 2Kb appena. Utilizzare al meglio la poca memoria disponibile è assolutamente fondamentale;
- **Riutilizzo del codice:** la quasi totalità del codice già presente *online* e le librerie che supportano i moduli *hardware* sono interamente scritti in C++;
- **Conoscenza:** C++ è il linguaggio di programmazione con cui ho lavorato di più durante il mio percorso accademico.

Da queste motivazioni ne sono derivate delle conseguenze che mi hanno permesso una maggiore efficienza del progetto e una maggiore efficacia nella realizzazione:

- **Memoria Arduino UNO:** con C++ la gestione della memoria viene delegata al programmatore, rendendo la programmazione più complicata, ma più efficace. Grazie a questo fattore del linguaggio è possibile sapere con certezza che la scheda non si bloccherà mai a causa dell'esaurimento della memoria SRAM;
- **Riutilizzo di codice:** la presenza di codice già pronto e funzionante velocizza la parte del lavoro di codifica;
- **Conoscenza:** la previa conoscenza del linguaggio mi permette di procedere con velocità durante la codifica e di usare le *featuring* del linguaggio in modo efficiente ed efficace.

- **EEPROM 24LC256**

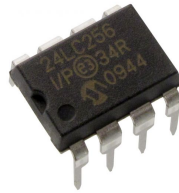


Figura 3.6: Microcip 24LC256

Hardware scelto per compensare la bassa capacità di memoria di Arduino UNO. Questo componente, possedendo una capacità di 32Kb, colma il problema del salvataggio dei *timestamp*, passando da 85 a 5461.

E' possibile interfacciarsi con la memoria tramite la libreria *Wire*.

- **Protocollo I2C:**

E' stato richiesto come vincolo di progetto l'utilizzo del protocollo I2C per lo scambio di informazioni tra le componenti *hardware*. Questo protocollo possiede ottime capacità di scambio di informazione a basso consumo a discapito di velocità, raggio d'azione e flessibilità. In Arduino il protocollo I2C è ampiamente supportato da una libreria interna denominata *Wire*, nel cui sito (<http://www.arduino.cc/en/reference/Wire>) si possono trovare tutte le informazioni per il suo utilizzo.

Studiando a fondo le specifiche tecniche della libreria *Wire* ho valutato che la banda di 32Byte per protocollo poteva essere un problema che limitasse la velocità di trasmissione dei dati. E' stato quindi necessario prendere in considerazione questo elemento come fattore di possibili problematiche.

- **Tiny-RTC**

Il modulo *hardware* in questione funziona sia come orologio, cioè ritornando data ed ora precisa in caso di richiesta, sia da *sveglia* o *allarme*. E' infatti possibile collegare il segnale di *interrupt* con la scheda Arduino e ricevere così una interruzione qualora l'ora preimpostata sia giunta. Oltre a ciò, la *Tiny-RTC* non presenta ulteriori particolarità, se non il fatto che è predisposto dai produttori un protocollo tramite la libreria *Wire* per l'interrogazione dell'*hardware*.

3.2 Analisi dei Requisiti

3.2.1 Classificazione dei Requisiti

I requisiti sono stati suddivisi per ambito, tipologia e priorità:

- **Ambito**

- *Scheda Slave (S)* : requisito legato alla scheda Slave del progetto;
- *Libreria Master (M)* : requisito legato alla libreria Master del progetto;
- *Altro (A)* : requisito che esprime un ambito diverso dai precedenti.

- **Tipologia**

- *Funzionale (F)* : requisito che esprime una funzionalità del progetto;
- *Vincolo (V)* : requisito che esprime un vincolo che il progetto deve soddisfare;
- *Prestazionale (P)*: requisito che esprime una condizione di *performance* che il progetto deve soddisfare.

- **Priorità**

- *Requisito Obbligatorio (O)* : requisito obbligatori ai fini del completamento dello *stage*;
- *Requisito Desiderabile (D)* : requisito che fornisce valore aggiunto al progetto di *stage*, ma che non rappresenta una condizione necessaria per la conclusione dei lavori.

3.2.2 Individuazione dei Requisiti

La maggior parte del materiale informativo, che mi ha permesso di compilare la lista dei requisiti, è stato acquisito durante le prime riunioni, precedenti l'inizio dei lavori, assieme al responsabile del dipartimento e il responsabile di FabLab. Contemporaneamente allo studio del dominio, tenendo informati tramite *mail* i responsabili dei progetti di FabLab e RaspiBO, ho identificato nuovi requisiti di progetto.

| Identificativo | Descrizione |
|----------------|--|
| <i>SVO1</i> | Riconoscere gli <i>interrupt</i> inviati dal pluviometro |
| <i>SVO2</i> | Riconoscere segnali falsi positivi inviati dal pluviometro |

| Identificativo | Descrizione |
|----------------|---|
| <i>SVO3</i> | Salvare i <i>timestamp</i> in un formato compresso nel <i>microchip</i> di memoria 24LC256 |
| <i>SVO4</i> | Tener traccia dell'ultimo <i>timestamp</i> letto dalla scheda Master |
| <i>SVO5</i> | Interrogare tramite I2C la <i>Tiny-RTC</i> per recuperare ora e data |
| <i>SVO6</i> | Fornire un protocollo di interrogazione I2C per la scheda Master |
| <i>SVO7</i> | Progettare il <i>software</i> in modo da poter implementare nuovi protocolli di interfacciamento in modo semplice |
| <i>SPO8</i> | Inviare i <i>timestamp</i> compressi tramite protocollo I2C per velocizzarne il processo del 20% |
| <i>SPO9</i> | Organizzare l'invio dei dati in modo da non riempire la memoria SRAM della scheda Slave o Master, senza superare quindi i 1200Kb ² |
| <i>SPO10</i> | Organizzare la gestione della memoria secondo un algoritmo <i>round-robin</i> |
| <i>SFO11</i> | Inviare i <i>timestamp</i> tramite protocollo I2C non ancora letti alla scheda Master |
| <i>SFO12</i> | Inviare i <i>timestamp</i> presenti in memoria |
| <i>SFO13</i> | Cancellare i <i>timestamp</i> già letti, mantenendo quelli da leggere |
| <i>SFO14</i> | Cancellare i <i>timestamp</i> presenti in memoria e ripristinare al valore di <i>default</i> i contatori |
| | |
| <i>MVO15</i> | Interrogare tramite I2C la scheda Slave |
| <i>MVO16</i> | Progettare il <i>software</i> della libreria in modo da poter implementare nuovi protocolli di interfacciamento in modo semplice |
| <i>MFO17</i> | Richiedere i <i>timestamp</i> non ancora letti |

²1200Kb sono un valore che si presta ad essere ottimale in una scheda Arduino UNO, in cui sono presenti 2Kb di memoria SRAM

| Identificativo | Descrizione |
|----------------|--|
| <i>MFO18</i> | Richiedere i <i>timestamp</i> presenti in memoria |
| <i>MFO19</i> | Richiedere la cancellazione dei <i>timestamp</i> già letti |
| <i>MFO20</i> | Richiedere il ripristino della memoria |
| <i>MFO21</i> | Convertire i dati da formato compresso in un formato semplice da utilizzare |
| | |
| <i>AVO22</i> | Fornire la documentazione <i>online</i> della struttura <i>software</i> e dei metodi del progetto |
| | |
| <i>SVD23</i> | Progettazione e realizzazione <i>board</i> personalizzata con il codice funzionante della scheda Slave |
| <i>MFD24</i> | Convertire i dati da formato compresso in JSON |
| <i>AVD25</i> | Creare un <i>software</i> per l'interfaccia con una scheda Master Arduino o Arietta |
| <i>AVD26</i> | Creare una interfaccia <i>web</i> per l'interazione con la scheda Master con Arietta |
| <i>AVD27</i> | Trasportare il codice della scheda Slave da <i>board</i> Arduino a <i>board</i> Arietta |

Tabella 3.6: Requisiti

Strumenti Utilizzati

Per il tracciamento dei requisiti sono stati utilizzati dei foglio *online* di *Google Drive*, i quali venivano aggiornati di volta in volta durante il periodo di tracciamento. Alla fine del processo di individuazione, i requisiti sono stati approvati dal responsabile interno ARPAV e dal responsabile del FabLab di Verona. Il ritardo nelle risposte del responsabile di RasbiBO non mi ha permesso di avere un suo riscontro iniziale nella trattazione dei requisiti.

3.2.3 Casi d'Uso

Trattandosi di un progetto in cui le relazioni sistema utente sono limitate, ho ritenuto non fosse necessario introdurre grafici di casi d'uso.

3.2.4 Dettagli Degni di Nota

Durante la realizzazione avanzata del prodotto, RasbiBO mi ha richiesto di aggiungere un particolare requisito al sistema del progetto:

| | |
|------------|--|
| <i>SPO</i> | il sistema deve essere in grado di supportare richieste di lettura da parte della scheda Master ogni 5 secondi |
|------------|--|

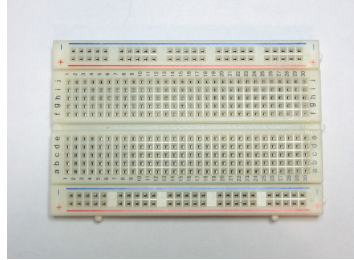
Il seguente requisito è stato introdotto a progetto già in stato avanzato di lavoro. Questo tipo di richiesta vede pressoché inutile l'utilizzo di un dispositivo esterno di memoria, poiché ogni 5 secondi, al massimo, possono essere salvati circa 10 *timestamp*. Poiché questi vengono letti in così poco tempo che mitigano il rischio di *overflow*, a meno di interruzione delle comunicazioni. Dopo un'attenta valutazione durante una riunione con il dirigente interno di ARPAV, ho deciso che il carico di lavoro per soddisfare questo requisito non era tale da impedirmi di finire il mio lavoro in tempo. Ho dunque creato un *fork* del progetto da dedicare per RasbiBO, in cui i *timestamp* compressi vengono salvati all'interno della memoria EEPROM della scheda, mentre il *branch* principale avrebbe seguito la normale tabella dei requisiti inizialmente accordata con FabLab di Verona.

3.3 Progettazione e Codifica

Definiti i limiti del progetto con l'approvazione dei requisiti, ho iniziato l'attività di progettazione. Prima di iniziare a la parte relativa le componenti *software*, ho realizzato la struttura *hardware* del progetto, in modo da avere la base su cui lavorare.

3.3.1 Introduzione Architettura Hardware

Le componenti *hardware* sono state connesse fra di loro utilizzando una *bread board*, una basetta di plastica che permette rapide connessioni fra strumenti *hardware*.

Figura 3.7: Scheda *bread board*

L'utilizzo di questo strumento ha permesso durante il periodo di *stage* un facile e veloce cambiamento delle connessioni delle componenti, facilitando la ricerca della struttura corretta.

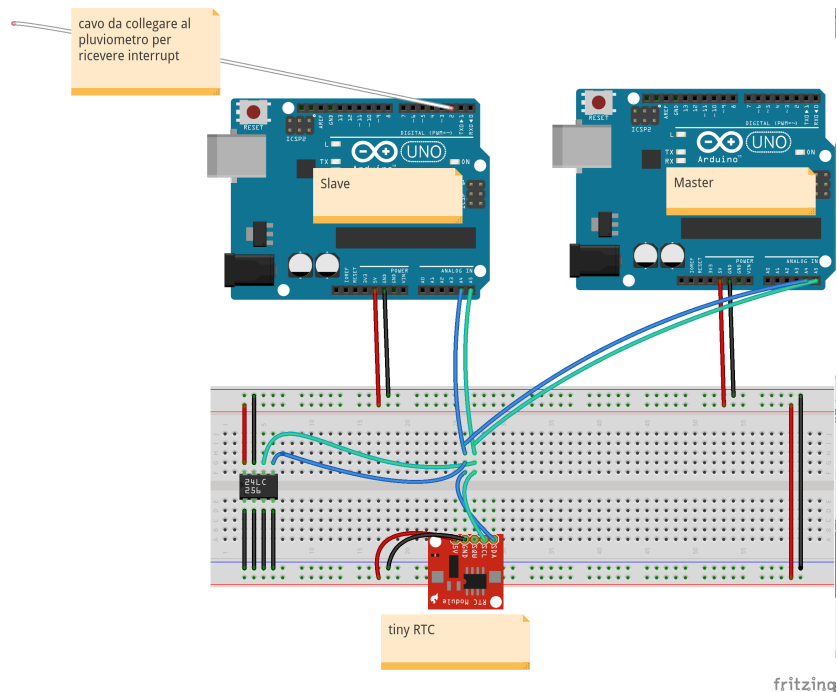


Figura 3.8: Schema elettronico progetto

Nella precedente figura (3.8) viene riportato lo schema elettronico delle connessioni delle componenti. Tramite l'utilizzo della *bread board* è stato possibile simulare un *bus* di comunicazione I2C, che mette in relazione tutte le componenti fra di loro. Successivamente con la realizzazione della *board* Slave personalizzata, verrà creato un blocco unico che incorporerà il microcontrollore ATmega328, il microchip 24LC256 e la *Tiny-RTC*, lasciando

due entrate per la connessione I2C con una scheda Master esterna e un componente *reed*_g³ per la lettura degli eventi del pluviometro.



Figura 3.9: Foto prototipo *board* personalizzata

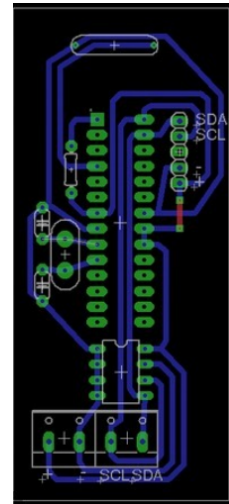


Figura 3.10: Schema elettronico *board* personalizzata

Verso la fine dei lavori durante la fase di *testing* finale è stato utilizzata la *board* personalizzata visibile in figura (3.9), con relativo schema elettronico in (3.10).

3.3.2 Introduzione Architettura Software

Le due componenti principali del sistema, scheda Slave e Master, devono dunque relazionarsi fra di loro tramite un protocollo prestabilito I2C, che consente lo scambio di informazioni qualora queste siano richieste. La libreria *Wire* mette a disposizione una serie di funzioni che permettono la creazione di un sistema *Slaves and Master* con Arduino. Stabilito l'indirizzo della scheda Slave⁴.

³ la bascula del pluviometro è provvista di un magnete, che ad ogni basculata, oscilla davanti al *reed* posto dentro la board personalizzata. Questo processo produrrà la chiusura del contatto e quindi la creazione di una interruzione (*interrupt*)

⁴nel caso specifico è stato impostato l'indirizzo 2, ma può essere facilmente modificato dal codice sorgente, qualora questo indirizzo sia stato già inserito in un sistema *Slaves and Master*

3.3.3 Architettura Libreria Scheda Master

[:] Sezione in cui descrivo l'architettura della parte riguardante la libreria Master che permette di interfacciarsi con la Scheda Slave facilmente.

3.3.4 Architettura Scheda Slave

[:] Sezione in cui descrivo l'architettura della Scheda Slave, che si interfaccia direttamente con il Pluviometro (questa parte è più ampia della precedente).

3.3.5 Design Pattern Utilizzati

[:] Sezione in cui elenco, motivo e descrivo i Design Pattern utilizzati nell'architettura del progetto.

3.3.6 Dettagli Degni di Nota

[:] Sezione in cui espongo alcuni dettagli rilevanti dell'architettura e soprattutto le difficoltà avute a causa delle limitate risorse hardware delle schede in mio possesso.

3.3.7 Progettazione di Dettaglio e Codifica

[:] Introduzione della sezione in cui spiego come mi sono adattato alle continue richieste di prototipi da parte dell'agenzia e come ho organizzato la progettazione in modo da rendere il codice facilmente modificabile e riutilizzabile per una codifica incrementale.

Dettagli Degni di Nota

[:] Sezione in cui descrivo alcuni metodi della libreria Master che permettono un facile interfacciamento con la scheda Slave. In aggiunta come sia possibile aggiungere facilmente nuove funzionalità al sistema senza alcuna iterazione, ma incrementando il codice.

3.3.8 L'Utilizzo dei Prototipi

[:] Breve sezione in cui spiego come, a causa delle necessità dell'ente, mi sia trovato costretto ad un continuo ciclo di Progettazione, Codifica, *test*, Progettazione Incrementale, Codifica, *test*..., approccio diverso da quello affrontato durante il mio percorso accademico.

3.4 Verifica e Validazione

[:] Descrizione dell'attività di verifica e validazione e breve descrizione delle sottosezioni successive

3.4.1 Analisi Statica

[:] Elenco e descrizione degli strumenti utilizzati per l'analisi statica e alcuni valori che meritano d'esser presi in considerazione.

3.4.2 *test* sul Sistema Slave

[:] Sezione in cui descrivo in dettagli i *test* effettuati sulla scheda Slave, in progressione con gli incrementi effettuati.

3.4.3 *test* sul Sistema Master Slave

[:] Descrizione dei *test* effettuati sul sistema Slave integrato con le interazioni con la scheda master in progressione con gli incrementi effettuati.

3.4.4 *test* di Sistema

[:] Descrizione dei *test* effettuati sull'intero sistema.

3.4.5 Dettagli Degni di Nota

[:] Descrizione del problema degli impulsi falsi positivi inviati dal pluviometro, riscontrati durante i *test* dell'intero sistema e soluzione.

3.4.6 Consuntivo Orario Finale

[:] Descrizione oggettiva e motivazione fra le discrepanze di orario preventivato ed effettivo in relazione agli obbiettivi raggiunti

Capitolo 4

Valutazioni Finali

4.1 Raggiungimento degli Obbiettivi

[:] Descrizione della sezione e introduzione delle sue sottosezioni

4.1.1 Rendiconto degli Obbiettivi

[:] Descrizione oggettiva fra obbiettivi soddisfatti e non.

4.2 Difficoltà Ricontrate

[:] Elenco e descrizione delle difficoltà avute durante il lavoro di Stage con relativa soluzione attuata.

4.3 Bilancio Formativo

[:] Descrizione della sezione e introduzione delle sue sottosezioni.

4.3.1 Preparazione Iniziale

[:] Descrizione delle mie capacità iniziali e come queste mi siano state utili per il periodo di stage; corsi utili per la mia attività specifica e cosa invece mi è mancato.

4.3.2 Esperienze Acquisite

[:] Elenco e descrizione esperienze acquisite durante il periodo di stage suddivise fra competenze e conoscenze.

4.3.3 Considerazioni Finali

[:] Sezione in cui riassumo l'esperienza di stage facendone un'analisi obbiettiva. Considerando le mie capacità iniziali e relazionandole con l'offerta proposta dall'agenzia, le difficoltà riscontrate e la mia preparazione e capacità di analizzare e affrontare i problemi riscontrati. Come mi siano stati più utili alcuni corsi universitari e come invece altri, se fatti in modo diverso, avrebbero potuto conferirmi maggior competenze per la riuscita dello stage. Ed infine cosa mi ha trasmesso lo stage in termini di competenze per l'inserimento nel mondo del lavoro.

Capitolo 5

Glossario

Il glossario segue le seguenti regole topografiche:

1. **Acronimi:** **ASE** (Acronimo Solo d'Esempio):
descrizione dell'acronimo;
2. **Termini:** **Termine:**
descrizione del termine.

A

- **ARPAV** (Agenzia Regionale per la Prevenzione e Protezione Ambientale nel Veneto) :
agenzia regionale le cui attività competenti riguardano la tutela, il controllo, il recupero dell'ambiente e per la prevenzione e promozione della salute collettiva al fine di conseguire la massima efficacia nell'individuazione e nella rimozione dei fattori di rischio per l'uomo e per l'ambiente;
- **Applicativi Web-Based:**
programmi che funzionano su piattaforma Internet/Intranet e vengono visualizzati ed eseguiti da un browser qualsiasi. La sola installazione che richiedono è quella che viene eseguita su un unico computer dedicato chiamato Web-server;
- **ATMEL:**
società produttrice di semiconduttori con sede a San Jose, California (USA), fondata nel 1984 e specializzata nella realizzazione di microcontrollori di tipo system-on-a-chip con memoria flash integrata.

C

- **CIVEN (Coordinamento Interuniversitario Veneto per le Nanotecnologie) :**
associazione senza fini di lucro con lo scopo di progettare e realizzare iniziative di formazione, di ricerca, di sperimentazione industriale e di trasferimento tecnologico al mondo imprenditoriale nell'ambito del settore delle nanotecnologie;
- **Comitato provinciale di coordinamento :**
comitato il cui fine è di garantire il coordinamento delle attività del dipartimento provinciale di ARPAV con le attività delle competenti strutture della provincia e dei comuni. Il comitato ha compiti di consulenza e di proposta; in particolare:
 - formula al direttore generale dell'ARPAV proposte per la definizione dei programmi annuali di attività;
 - verifica l'andamento e i risultati delle attività programmate, esprimendo al direttore generale dell'ARPAV valutazioni e proposte.

D

-

F

- **FabLab (Fabrication Laboratory):**
piccola officina che offre servizi personalizzati di fabbricazione digitale. Un fab lab è generalmente dotato di una serie di strumenti computerizzati in grado di realizzare, in maniera flessibile e semi-automatica, un'ampia gamma di oggetti. Tra questi vi sono prodotti tecnologici generalmente considerati di appannaggio esclusivo della produzione di massa.

G

- **GPP (Green Public Procurement) :**
modalità di acquisto, rivolta e adottata dalle pubbliche amministrazioni locali o nazionali, basata su criteri ambientali oltre che sulla qualità e prezzo di prodotti e servizi. GPP significa:

- acquistare solo ciò che è indispensabile;
 - considerare l'impatto ambientale del prodotto lungo tutto il suo ciclo di vita e non solo al momento dell'utilizzo;
 - stimolare in senso ambientalmente sostenibile l'innovazione di prodotti e servizi;
 - adottare comportamenti d'acquisto responsabili e dare il buon esempio nei confronti dei cittadini;
 - sviluppare la comunicazione e lo scambio di informazioni tra gli enti locali, le imprese e i consumatori.
- **GNU GPL (GNU General Public License) :**
licenza per software libero, originariamente stesa nel 1989 da Richard Stallman per distribuire i programmi creati nell'ambito del Progetto GNU della *Free Software Foundation* (FSF) per il progetto GNU;

I

- **I2C (Inter Integrated Circuit) :**
bus composto da almeno un master ed uno slave (letteralmente capo, padrone e sottoposto, schiavo), trattasi di un sistema di comunicazione seriale bifilare utilizzato tra circuiti integrati il cui vantaggio risiede nell'utilizzo di sole due linee di comunicazione, limitandone la velocità e distanza;
- **IPPC (Integrated Pollution Prevention and Control) :**
direttiva che prevede un nuovo approccio per la riduzione degli impatti ambientali con la graduale applicazione di un insieme di soluzioni tecniche (impiantistiche, gestionali e di controllo) per evitare, o, qualora non sia possibile, ridurre, le emissioni di inquinanti nell'aria, nell'acqua e nel suolo, comprese misure relative ai rifiuti.

M

- **Map-Server:**
ambiente multiplatforma di sviluppo e fruizione *Open Source* finalizzato alla rappresentazione di dati geospaziali;
- **Memoria SRAM (Static Random Access Memory):**
tipo di RAM volatile che non necessita di refresh. I banchi di memorie SRAM consentono di mantenere le informazioni per un tempo teoricamente infinito, hanno bassi tempi di lettura e bassi consumi, specialmente in condizioni statiche;

- **MQTT (MQ Telemetry Transport):** protocollo di messaggistica leggero posizionato in cima a TCP/IP disegnato per le situazioni in cui è richiesto un basso impatto e dove la banda è limitata.

O

- **OIV (Organismo Indipendente di Valutazione):** organismo, ai sensi dell'articolo 14 del decreto legislativo n. 150/2009, che svolge, all'interno di ciascuna amministrazione, un ruolo fondamentale nel processo di misurazione e valutazione delle strutture e dei dirigenti e nell'adempimento degli obblighi di integrità e trasparenza posti alle amministrazioni.

P

- **PDCA (Plan Do Check Act):** modello studiato per il miglioramento continuo della qualità in un'ottica a lungo raggio. Serve per promuovere una cultura della qualità che è tesa al miglioramento continuo dei processi e all'utilizzo ottimale delle risorse. Questo strumento parte dall'assunto che per il raggiungimento del massimo della qualità sia necessaria la costante interazione tra ricerca, progettazione, test, produzione e vendita. Per migliorare la qualità e soddisfare il cliente, le quattro fasi devono ruotare costantemente, tenendo come criterio principale la qualità.
- **Pluviometro:** strumento utilizzato per misurare la quantità di pioggia caduta. Esso fa parte della dotazione di strumenti principali di una comune stazione meteorologica;

R

- **RaspiBO:** gruppo informale di appassionati di elettronica ed informatica libera della zona di Bologna volto alla progettazione e sviluppo di nuove idee (www.raspibo.org/wiki/index.php/Pagina_principale);
- **RE.S.M.I.A. (Reti e Stazioni di Monitoraggio Innovative per l'Ambiente) :** progetto di ricerca industriale con l'obiettivo di potenziare ed integrare la rete di monitoraggio ambientale a disposizione di ARPAV;

- **Reed**

componente costituito da due lamine, realizzate con materiale ferromagnetico (una lega di ferro-nichel), parzialmente sovrapposte e separate tra loro di qualche decimo di millimetro. Sulle lamine contrapposte sono riportati dei contatti (generalmente in oro diffuso). Le lamine vengono sigillate all'interno di un piccolo contenitore di vetro riempito di gas inerte (azoto o argon). Le estremità delle lamine (opposte ai contatti) fuoriescono dal contenitore e costituiscono i terminali del contatto. In presenza di un campo magnetico le lamine diventano sede di flusso magnetico e sulle estremità si formeranno poli di segno opposto che tenderanno ad attrarsi. Se il campo magnetico è sufficientemente forte (100-200 amperspire), la forza d'attrazione vince la resistenza a flessione, e queste attraendosi chiuderanno il contatto.

S

- **Subcontractor:**

situazione che si verifica quando ad un'azienda o un ente viene delegato parte di un lavoro già sotto contratto;

W

- **Web-Based:**

in informatica si intende per Web-Based qualsiasi cosa, applicazione, sistema informatico, *hardware*, accessibile/fruibile via *web* per mezzo di un *network*.

- **WSN (Wireless Sensor Network):**

indica una determinata tipologia di rete che, caratterizzata da una architettura distribuita, è realizzata da un insieme di dispositivi elettronici autonomi in grado di prelevare dati dall'ambiente circostante e di comunicare tra loro;