



Allegati

Capitolo

[Quaderno dei compiti](#)

[Diari](#)

[Guida Modulo ACC](#)

[Guida Utilizzo Arduino YUN](#)

[Guida Modulo back-end](#)

[Guida Modulo domotics database](#)

[Guida Modulo front-end](#)

[Guida Modulo LDAP](#)

[Guida Installazione Apache Tomcat 9](#)



Quaderno dei compiti



Diari

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 13.02.2019

Lavori svolti

Oggi ci è stato consegnato il quaderno dei compiti del terzo progetto, dopo aver letto le richieste che ci sono state fatte abbiamo discusso con il committente, dopo c'è stata una piccola riunione di gruppo dove sono stati discussi i requisiti ed infine è stato fatto il gantt, il Repository su Git e Trello Board.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 15.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 15.02.2019

Lavori svolti

Giulio Progettazione di ACC, che sarà il sistema che verrà utilizzato per comunicare fra gli Arduino e il Server Domotics, progettazione del protocollo di comunicazione.

Progettazione del protocollo, il quale sarà basato su un ID, una KEY ed il protocollo HTTP.

Esempio di ID (ACC-Client-ID): `DMTSID0ABCDEFABCDEF`

Esempio di KEY (ACC-Client-KEY): `ABCDEFABCDEF`

Esempio di discover: `DMTSID0ABCDEFABCDEF-192.168.1.34-REQUEST`

Esempio di response: `DMTSID0ABCDEFABCDEF-192.168.1.34-REQUEST-ABCDEFABCDEF-192.168.1.2-ACCEPTED`

Mattia

Creazione macchina virtuale in Centos07 per il web server, installazione java, tomcat e configurazione(Da finire).
<https://lntut.com/how-to-setup-network-after-rhelcentos-7-minimal-installation/>

Paolo

Creazione Frontend e definizione dello stile di base per le pagine del sito web Ho creato una pagina di esempio per il LogIn con dei colori di base nero bianco e giallo. Mi sono reso conto che i colori fanno schifo e che lo stile di base della pagina fa abbastanza schifo.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 20.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 20.02.2019

Lavori svolti

Mattia

Oggi ho continuato la configurazione del server in tomcat, funziona ma la configurazione del file tomcat.service non funziona, penso sia la versione di Tomcat, ho installato la 16 devo provare con la 10. Ho configurato TomCat con IntelliJ e ho provato a fare l'hello world come prova per testare il funzionamento.

Giulio

Ho scritto le indicazioni base della comunicazione ACC-Server -> ACC-Client. La quale è molto simile a quella inversa, vengono stabiliti due sistemi di comunicazione perché possono entrambe le parti devono poter fare richieste alla parte opposta.

Abbiamo fatto la riunione di team, definendo i task di design.

definito le indicazioni di base per l'ACC-AutoConfiguration, il sistema che permettere il funzionamento autonomo del sistema.

definito le indicazioni di base per l'ACC-Client-Key-Generator.

Paolo

Abbiamo definito i colori di base per il sito web. Abbiamo ripensato allo stile delle pagine.

Ho iniziato ad installare Tomcat usando IntelliJ, di base vengono create delle cartelle src (contiene i file servlet e java) e la cartella web che contiene i siti di base .jsp.

Devo imparare ad usare le servlet e a configurarle con le pagine.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 22.02.2019

Lavori svolti

Paolo

Oggi ho imparato ad usare i servlet e i .jsp su tomcat e ho iniziato a capire come funziona la comunicazione database/server tomcat e come prendere i dati dal database e come aggiungerli. In seguito ho cercato la soluzione al problema degli arduino che abbiamo a disposizione che non vengono più prodotti. Ho scoperto che si può o usare l'arduino Yun oppure comprare gli arduino da siti secondari che li vendono a prezzi dai 40 ai 50 Fr.-

Giulio

Java tomcat Hello World, ho creato un hello world per tomcat, utilizzando sia le servlet sia i file **JSP**. per poter far funzionare le servlet bisogna configurare il file **web/WEB-INF/web.xml**, inserendo le informazioni della servlet precedentemente creata.

Codice Servlet:

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();

    out.println("<!DOCTYPE html><html><head><title>Domotics</title></head><body>");
    out.println("<h1>It Works!</h1>");
    out.println("<p>Return to <a href=\"index.jsp\">index.jsp</a></p>");
    out.println("</body></html>");
    out.close();
}

protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    doPost(request, response);
}
```

codice **web.xml**:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
                             http://java.sun.com/xml/ns/javaee/web-app_4_0.xsd"
          version="4.0">
    <servlet>
        <servlet-name>HelloWorldServlet</servlet-name>
        <servlet-class>HelloWorldServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>HelloWorldServlet</servlet-name>
        <url-pattern>/HelloWorld</url-pattern>
    </servlet-mapping>
</web-app>
```

Learn LDAP

Ho scaricato [Apache directory studio](https://directory.apache.org/studio/) (<https://directory.apache.org/studio/>), che è un programma per esplorare la struttura di LDAP, li mi sono connesso con il mio utente, ed ho trovato l'albero delle OU del mio utente. Lo ho inserito in uno script che ho trovato su [web development helpdesk](#) al link:

<http://helpdesk.objects.com.au/java/an-example-of-simple-ldap-authentication>. Il quale non funzionava, con l'aiuto del docente Raimondi, sono riuscito a capire dov'era l'errore, nel `DN`, bisogna inserire

`CN=nome.utente,OU=3,OU=I,OU=IN,OU=SAM,OU=allievi,DC=CPT,DC=local`, il primo elemento è il nome utente, gli elementi successivi invece sono la struttura di LDAP.

Mattia

Ho fatto il web server sull'arduino mega 2560 con lo shield Ethernet, dopo essermi informato sul funzionamento ho trovato una guida che spiegava passo passo come creare un web server sull'arduino. Al momento prende l'indirizzo dal dhcp ed è possibile aprire da browser la pagina html che stampa tramite l'indirizzo. Riesco anche a ricevere tutto ciò che il client ritorna al server bisogna solo estrapolare le informazione che abbiamo bisogno per effettuare le varie operazioni. Mi sono documentato su come fare un web server sull'arduino Leonardo, principalmente sulle differenze di configurazione e compilazione. Abbiamo escluso l'arduino Ethernet.

Nella scatola c'è tutto.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Finire di capire il funzionamento di LDAP e iniziare a collegare LDAP ad una pagina di Login decente.

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 27.02.2019

Lavori svolti

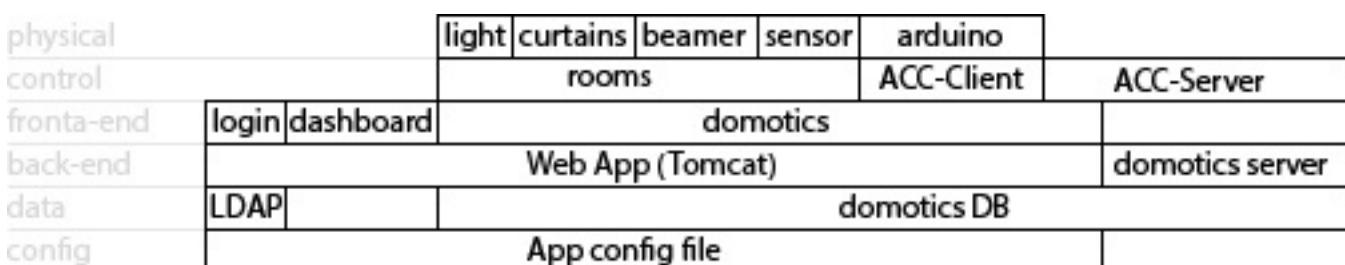
Mattia

Oggi ho presentato il secondo progetto, che mi ha preso circa un'oretta, dopodichè ho lavorato di nuovo sul web server dell'arduino 2560, estrapolo l'url dalle informazioni che il client invia al server. Ho iniziato a fare anche un po' di documentazione. Abbiamo inoltre deciso che utilizzeremo l'arduino Yun quindi l'arduino Leonardo e l'arduino Mega 2560 più shield è stato escluso perché sono fuori produzione o perché occupano troppo spazio.

Giulio

Structure

Oggi ho disegnato l'architettura di sistema dell'intero progetto, quindi ogni modulo e le loro relazioni.



La quale è composta dalla struttura di base, la quale carica i vari moduli, sulla quale si appoggia il livello dei dati (contentente il sistema di accesso tramite LDAP e il DB di domotics), sopra vi è il livello del web server Tomcat (con il back-end) ed il server di domotics, poi vi è il livello del front-end con i moduli delle varie pagine (login, dashboard, e domotics). Poi sopra vi sono i sotto moduli del front-end con sopra i moduli degli elementi fisici.

LDAP

Poi ho disegnato l'architettura del modulo di LDAP, che è composto da una classe di test, per capire come funziona la connessione di LDAP su Java e per capire quali parametri usare per collegarsi alla rete del CPT (TestLdapCpt).

Dopo di che è stata disegnata la classe `LdapConnector`, che verrà utilizzata per connettersi facilmente a LDAP con java.

TestLdapCtp**+static main(args[]): String: void****LdapConnector**

```
+static DEFAULT_PORT: int
+static DEFAULT_SECURITY_AUTHENTICATION: String
+static DEFAULT_INITIAL_CONTEXT_FACTORY: String
-domain: String
-port: int
-base: String
-security: String

+setDomain(domain: String): void
+getDomain(): String
+setPort(port: int): void
+getPort(): int
+setBase(base: String): void
+getBase(): String
+setSecurity(security: String): void
+getSecurity(): String
+LdapConnector(domain: String, port: int, base: String, security: String)
+LdapConnector(domain: String, port: int, base: String)
+LdapConnector(domain: String, base: String)
+getConnection(): String
+getDn(username: String): String
+getEnvironment(username: String, password: String): DirContext
+getDirContext(username: String, password: String): DirContext
```

Dopo di che ho implementato la classe `LdapConnector` siccome la classe `TestLdapCtp` è stata già implementata la scorsa lezione.

```
/*
 * Get the hashtable environment of the connection.
 *
 * @param username Username of the connection.
 * @param password Password of the connection.
 * @return Hashtable Environment of the connection.
 */
private Hashtable<String, String> getEnvironment(String username, String password) {
    Hashtable<String, String> environment = new Hashtable<String, String>();

    environment.put(Context.INITIAL_CONTEXT_FACTORY, DEFAULT_INITIAL_CONTEXT_FACTORY);
    environment.put(Context.PROVIDER_URL, getConnection());
    environment.put(Context.SECURITY_AUTHENTICATION, getSecurity());
    environment.put(Context.SECURITY_PRINCIPAL, getDn(username));
    environment.put(Context.SECURITY_CREDENTIALS, password);

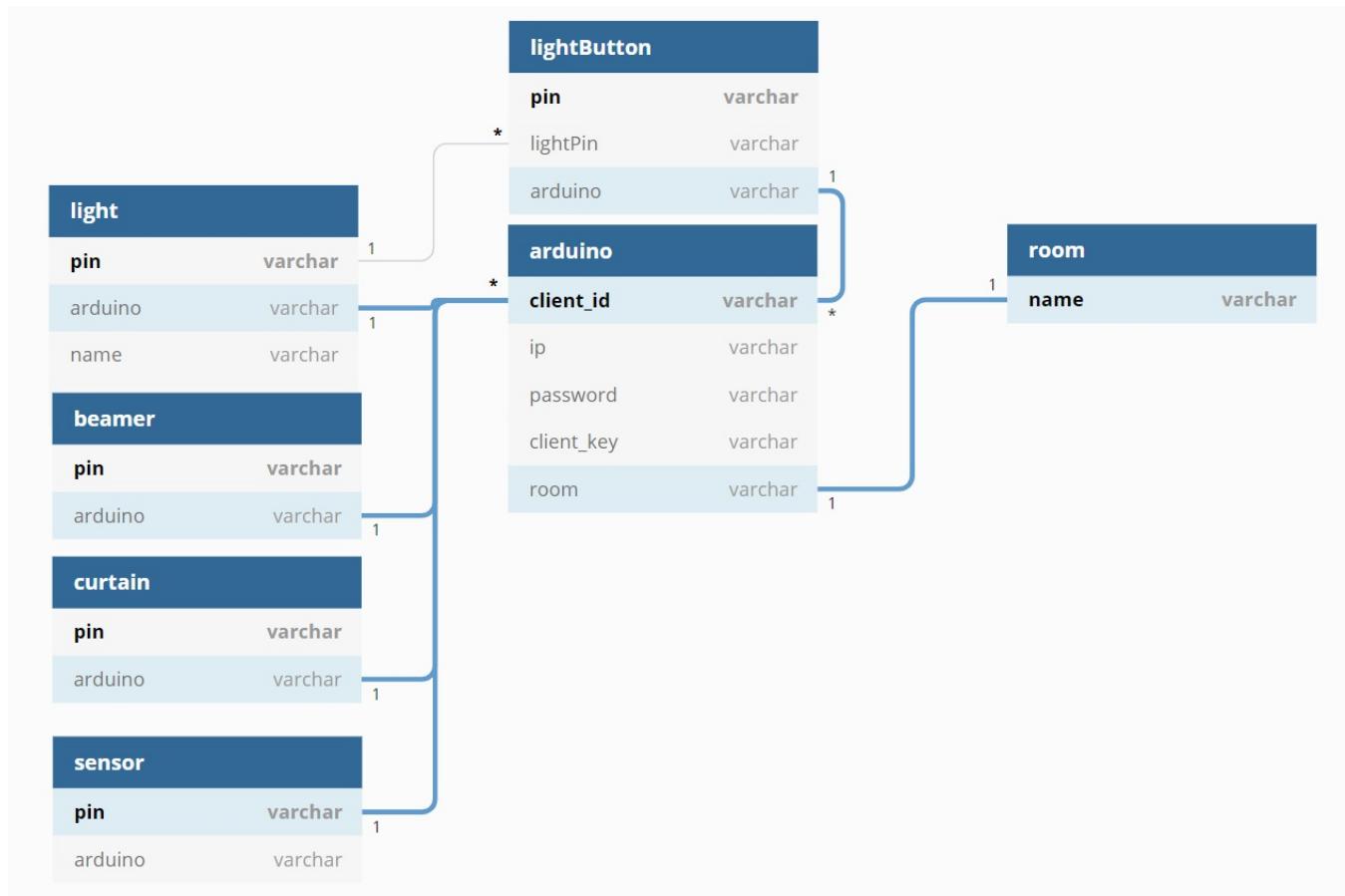
    return environment;
}
```

Il grosso della connessione a LDAP, è fatto in questo metodo, il quale crea una `Hashtable`, con tutti parametri che poi verranno inviati con la richiesta ad LDAP.

Ed infine ho scritto il test per la classe nel suo metodo `main`.

Paolo

Oggi ho creato il diagramma di base per il DB del sistema di domotica che contiene: -Le aule -I arduino -I componenti dell'arduino



In seguito abbiamo discusso sulla struttura generale del progetto. Suddividendolo in sei categorie:

- Fisico
- Controllo
- Front-end
- Back-end
- Data
- Configurazione

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro



Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 01.03.2019

Lavori svolti

Giulio

Ho aiutato Mattia a configurare IntelliJ con il progetto già esistente.

Ho implementato il modulo di autenticazione, quindi: l'interfaccia `Authenticator`, la quale viene implementata dai vari tipi di authenticatori, per esempio `LdapAuthenticator` (o `SqlAuthenticator`, che non verrà usato). Inoltre ho implementato una pagina di login di base (`web/login.html`), la quale serve solamente per testare che la servlet `LoginServlet` funzioni.

Poi ho implementato la pagina `App.jsp`, che per il momento servirà semplicemente per controllare che i cookies funzionino correttamente.

Ho disegnato le interfacce:

- login
- portal (l'interfaccia di base del portale)
- domotics (iniziato)

Mattia

Oggi ho testato la singola classe LDap che Giulio ha fatto (funziona), poi ho testato il login tramite LDap (non funziona), ho fatto il diagramma delle classi del LDap su StarUML e l'ho messo nella cartella design/portal, ho controllato il Gantt di base che doveva essere controllato e come struttura di base abbiamo deciso che ci può stare adesso bisogna completare quello finale con tutte le cose che dobbiamo fare e non solo in linea generale. Negli ultimi venti minuti mi sono documentato sull'utilizzo della jdbc per collegarsi tramite java al database.
<http://www.claudiodesio.com/java/jdbc.htm>

Paolo

Ho rifatto l'interfaccia web del login. Ho provato ad usare bootstrap ma ho capito che non riuscivo ad usarlo molto bene quindi ho ripristinato il vecchio sito creato senza bootstrap.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link href="css/style.css" rel="stylesheet" type="text/css">
    <meta charset="UTF-8">
    <title>Login Page</title>
</head>
<body class="defaultBg">
    <div class="log">
        <h1>Portal</h1>
        <form action="Login" method="post">
            <input class="text" type="text" placeholder="username" name="username">
            <input class="text" type="password" placeholder="password" name="password">
            <input class="but right" type="submit" value="login">
        </form>
    </div>
</body>
</html>
```

```
<br>
<br>
<span>© Domotics 2019</span>
<span class="right">cpt.local</span>
</div>
</body>
```

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 13.03.2019

Lavori svolti

Giulio

Risolto il bug sulla connessione di LDAP, ricercato il problema, il quale si trovava nella classe `portal.authentication.LdapAuthenticator` nel metodo `getBase0u()`, che inizialmente metteva la base della OU prima di quella di riferimento dell'allievo.

```
private String getBase0u() {
    return BASE_OU + this.sub0u;
}
```

Mentre dovrebbe essere il contrario

```
private String getBase0u() {
    return this.sub0u + BASE_OU;
}
```

Poi ho provato con il sore ad accedere con la OU dei docenti ("OU=docenti") ed ha funzionato.

In fine ho cercato di risolvere il problema dei cookie, dopo aver eseguito il login con la `portal.authentication.LoginServlet` non si andava mai alla pagina `App.jsp`, perchè non veniva settato l'attributo `user` nella sessione.

```
session.setAttribute("user", requestUsername);
```

Mattia

Ho messo apposto il web server sull'arduino Mega 2560 perchè riguardandolo mi sono accorto che non era il modo migliore per ricavare l'url e alla fine ho fatto così:

```
char c = client.read();
// read a byte, then
Serial.write(c);
// print it out the serial monitor
header += c;
if (c == '\n') {
    // send a standard http response header
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("Connection: close");

    client.println();
    client.println("<!DOCTYPE HTML>");
    client.println("<html><br><head>");
    client.println("<title>Web Server</title>");
    client.println("</head><br><body>");
    if (header.indexOf("GET /on") >= 0) {
        client.println("<h1>On</h1>");
    } else if (header.indexOf("GET /off") >= 0) {
        client.println("<h1>Off</h1>");
    } else{
        client.println("<h1>Undefined</h1>");
```

```
        }
        client.println("</body><br></html>");
        delay(1);
        client.stop();
    }
}
```

Abbiamo fatto una piccola riunione di gruppo con il docente e abbiamo deciso che il venerdì prima delle vacanze dobbiamo fare il punto della situazione. Sono andato avanti con JDBC ma un errore mi ha bloccato e non sono riuscito a risolverlo: Error: java.lang.ClassCastException: java.math.BigInteger cannot be cast to java.lang.Long

Paolo

Ho iniziato l'interfaccia della dashboard. Ho avuto un po' di problemi nell'allineare gli elementi del nav sulla sinistra.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link href="css/style.css" rel="stylesheet" type="text/css">
    <link href="lib/font/fontawesome.min.css" rel="stylesheet" type="text/css">
    <meta charset="UTF-8">
    <title>Dashboard</title>
    <script src="js/navJs.js"></script>
<body class="gray">
    <div class="hide-nav fixed" id="triggerShow" onclick="showNav()"><span class="config"><i class="fa fa-bars"></i></span></div>
    <nav class="text-white" id="mainNav">
        <div class="hide-nav right" id="triggerHide" onclick="hideNav()"><span class="config"><i class="fa fa-times"></i></span></div>
        <div class="nav-title">Portal</div>
        <div class="nav-container">
            <ul class="nav-item">
                <li class="nav-li">
                    <a class="" href="#"><div class="nav-element">Domotics</div></a>
                </li>
            </ul>
        </div>
        <div class="bottomContainer">
            <a><span class="config"><i class="fa fa-cog right"></i></span></a>
            <a href="login.html"><span class="config"><i class="fa fa-power-off"></i></span></a>
        </div>
    </nav>
</body>
</html>
```

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

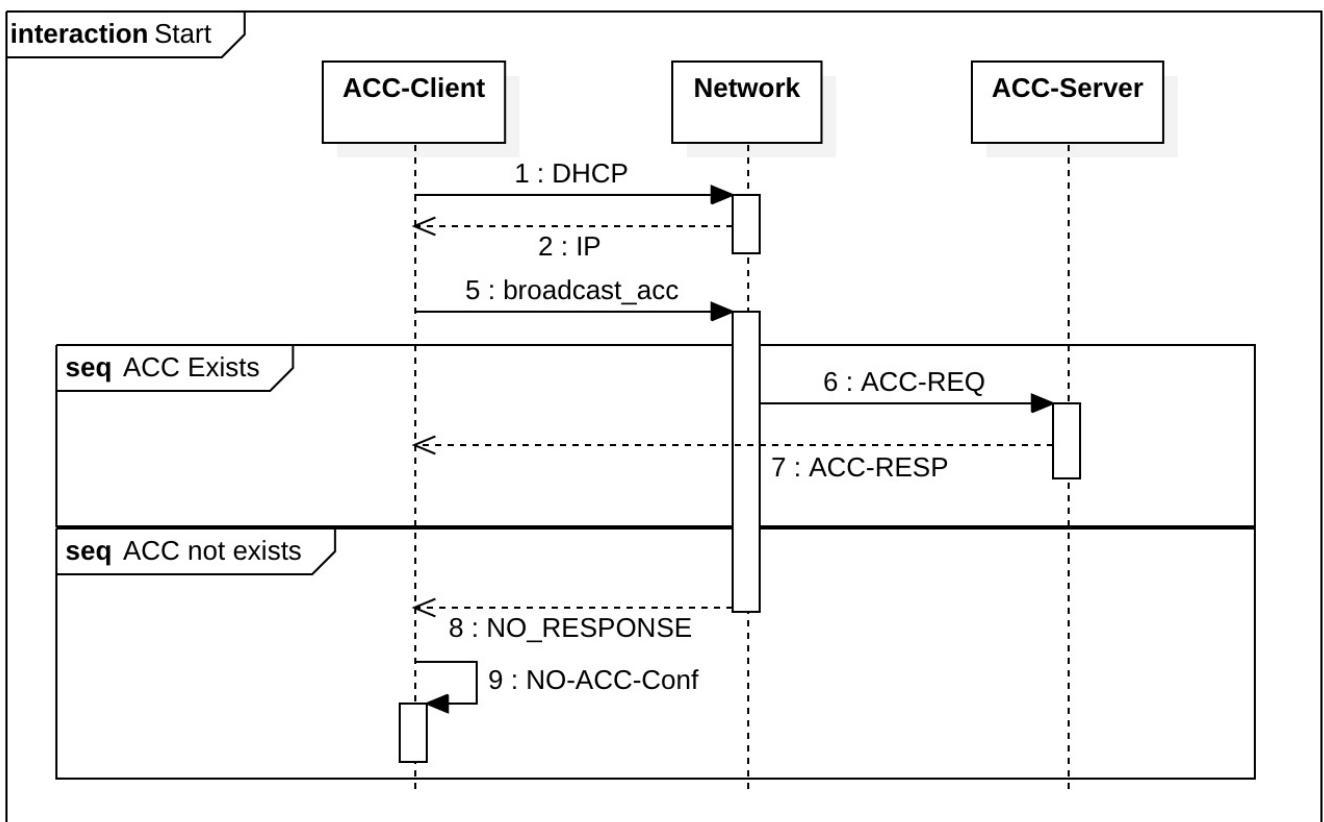
Canobbio, 15.03.2019

Lavori svolti

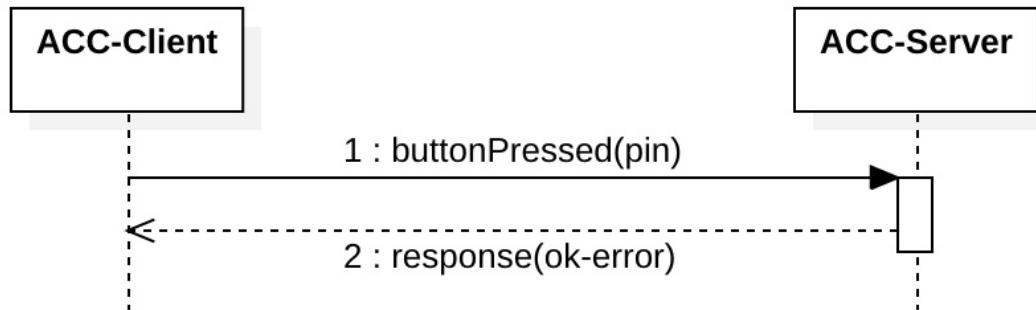
Giulio

ACC Comunicazione

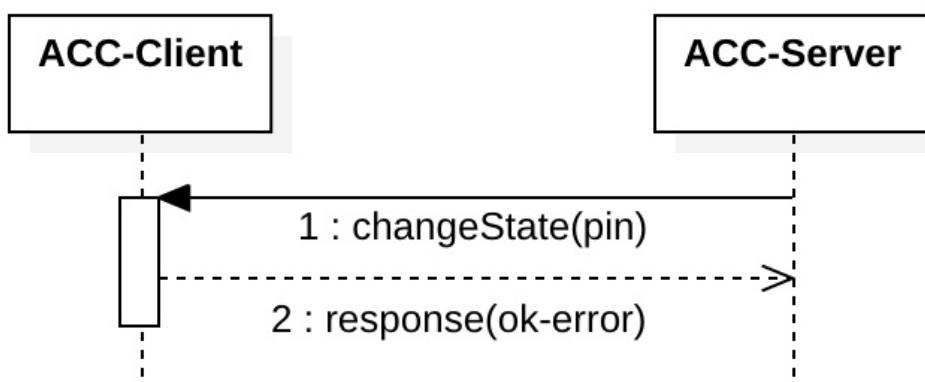
Ho disegnato il flusso di funzionamento del protocollo ACC, tramite un diagramma Swim lane, che permettono di mostrare il flusso del programma e le interazioni fra le varie macchine, suddividendo già le varie componenti.



In questo diagramma si può notare la richiesta dhcp dell'ACC-Client (Arduino), il quale quando riceve un IP esegue la broadcast ACC-Server Discover, se riceve una risposta con la sua configurazione si autoconfigura, altrimenti si mette in modalità no ACC-Server.

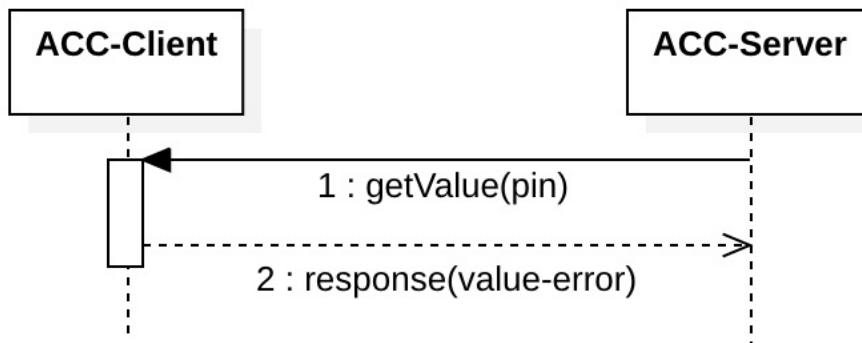
interaction buttonPressed

In questo diagramma si può notare l'invio del messaggio che un bottone è stato premuto da parte del ACC-Client verso l'acc server, il quale si preoccuperà poi di accedere la lampada corretta.

interaction changeState

Nel diagramma soprastante si può notare che l'ACC-Server richiede al client di cambiare lo stato di un pin.

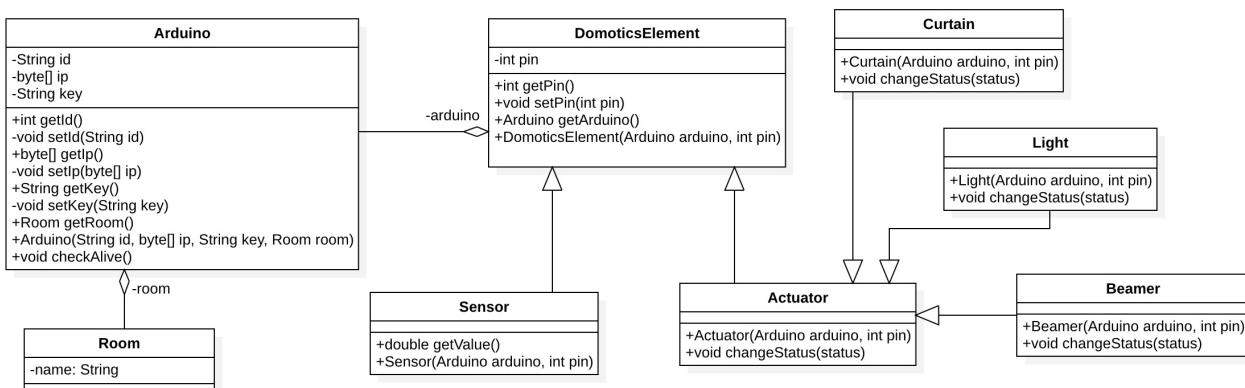
interaction getData



Nel diagramma si può notare che l'ACC-Server richiede il valore di un determinato pin del ACC-Client.

ACC Dati

Ho Disegnato il diagramma delle classi del modulo ACC-Server, per la parte degli elementi presenti nell ACC (room, arduino, lights, beamers, curtains and sensors).



Mattia

Oggi sono stato dietro al sistema JDBC per tutte le 4 ore. Innanzitutto mi sono informato maggiormente su come utilizzare e ho trovato un esempio molto interessante su come funziona.

<https://www.html.it/pag/66419/configurare-un-driver-per-luso-di-jdbc/> <http://www-db.deis.unibo.it/courses/SIL-A/PDF/JDBC-2p.pdf> <https://www.javatpoint.com/example-to-connect-to-the-mysql-database>

http://www.ntu.edu.sg/home/ehchua/programming/java/jdbc_basic.html

<https://www.vogella.com/tutorials/MySQLJava/article.html>

```

import java.sql.*;

public class JDBCApp {
    public static void main (String args[]) {
        try {
            // Carichiamo un driver di tipo 1 (bridge jdbc-odbc)
            String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
            Class.forName(driver);
            // Creiamo la stringa di connessione
            String url = "jdbc:odbc:myDataSource";
            // Otteniamo una connessione con username e password
  
```

```
Connection con =
    DriverManager.getConnection (url, "myUserName", "myPassword");
// Creiamo un oggetto Statement per poter interrogare il db
    Statement cmd = con.createStatement ();
// Eseguiamo una query e immagazziniamone i risultati
// in un oggetto ResultSet
    String qry = "SELECT * FROM myTable";
    ResultSet res = cmd.executeQuery(qry);
// Stampiamone i risultati riga per riga
    while (res.next()) {
        System.out.println(res.getString("columnName1"));
        System.out.println(res.getString("columnName2"));
    }
    res.close();
    cmd.close();
    con.close();
} catch (SQLException e) {
    e.printStackTrace();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
}
```

Il problema di questo codice è il driver, non ho capito come funziona il driver, ho trovato tante possibilità di driver ma nessuna di queste funzionava e tutte davano errori differenti. Non sono riuscito a risolvere il problema quindi lo finirò la prossima volta.

Paolo

Protesta ClimateStrike

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 20.03.2019

Lavori svolti

Giulio

Riguardato e controllato il design del modulo ACC con Mussi.

Cambiato la password dell'arduino yun, per farlo bisogna caricare il seguente sketch sull'Arduino. Dopo di che aprire la porta seriale e connettersi ed utilizzare i comandi di UNIX per cambiare la password, quindi `passwd`.

```
long linuxBaud = 250000;

void setup() {
    SERIAL_PORT_USBVIRTUAL.begin(115200); // open serial connection via USB-Serial
    SERIAL_PORT_HARDWARE.begin(linuxBaud); // open serial connection to Linux
}

boolean commandMode = false;

void loop() {
    // copy from USB-CDC to UART
    int c = SERIAL_PORT_USBVIRTUAL.read(); // read from USB-CDC
    if (c != -1) { // got anything?
        if (commandMode == false) { // if we aren't in command mode...
            if (c == '~') { // Tilde '~' key pressed?
                commandMode = true; // enter in command mode
            } else {
                SERIAL_PORT_HARDWARE.write(c); // otherwise write char to UART
            }
        } else { // if we are in command mode...
            if (c == '0') { // '0' key pressed?
                SERIAL_PORT_HARDWARE.begin(57600); // set speed to 57600
                SERIAL_PORT_USBVIRTUAL.println("Speed set to 57600");
            } else if (c == '1') { // '1' key pressed?
                SERIAL_PORT_HARDWARE.begin(115200); // set speed to 115200
                SERIAL_PORT_USBVIRTUAL.println("Speed set to 115200");
            } else if (c == '2') { // '2' key pressed?
                SERIAL_PORT_HARDWARE.begin(250000); // set speed to 250000
                SERIAL_PORT_USBVIRTUAL.println("Speed set to 250000");
            } else if (c == '3') { // '3' key pressed?
                SERIAL_PORT_HARDWARE.begin(500000); // set speed to 500000
                SERIAL_PORT_USBVIRTUAL.println("Speed set to 500000");
            } else if (c == '~') { // '~' key pressed?
                SERIAL_PORT_HARDWARE.write((uint8_t *)"\xff\0\0\x05XXXX\x7f\xf9", 11); // send "bridge shutdown" command
                SERIAL_PORT_USBVIRTUAL.println("Sending bridge's shutdown command");
            } else { // any other key pressed?
                SERIAL_PORT_HARDWARE.write('~'); // write '~' to UART
                SERIAL_PORT_HARDWARE.write(c); // write char to UART
            }
            commandMode = false; // in all cases exit from command mode
        }
    }
}

// copy from UART to USB-CDC
c = SERIAL_PORT_HARDWARE.read(); // read from UART
if (c != -1) { // got anything?
    SERIAL_PORT_USBVIRTUAL.write(c); // write to USB-CDC
}
```

<https://www.arduino.cc/en/Guide/ArduinoYun>

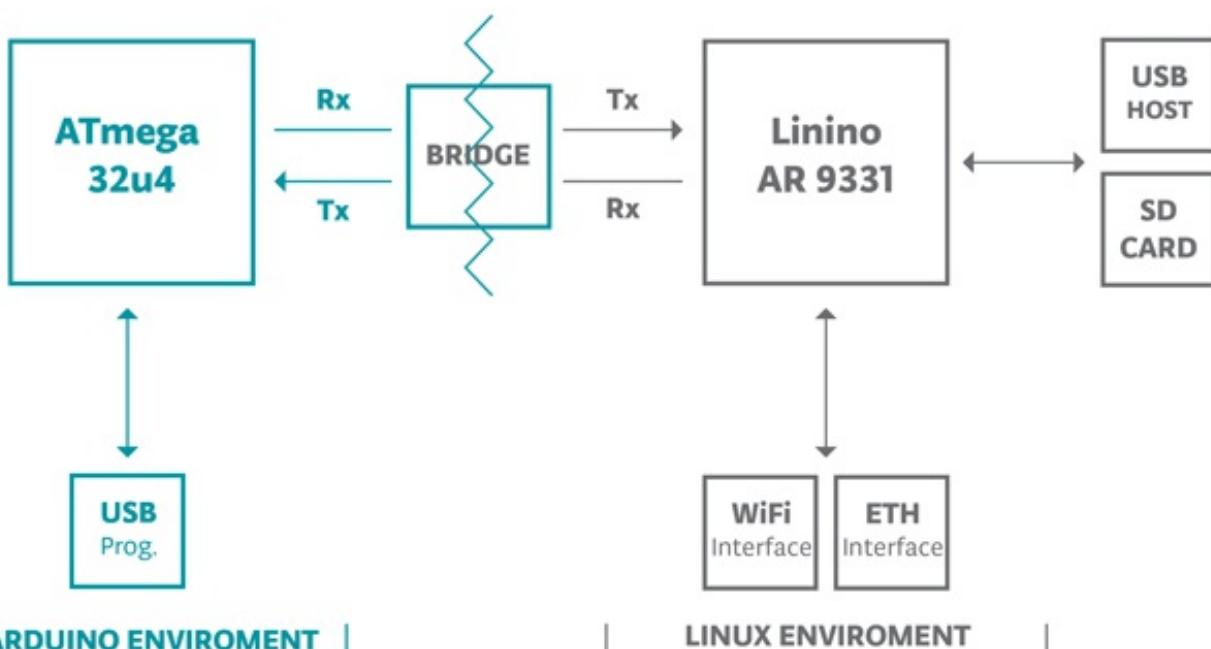
<http://forum.arduino.cc/index.php?topic=221431.msg1610137#msg1610137>

<https://www.arduino.cc/en/Tutorial/Bridge>

Dopo di che ho cercato di capire come funziona l'arduino yun. Il quale è composto principalmente di due componenti, un arduino con la scheda ATmega 32u4, connessa alla porta microUSB ed ai PIN.

All'altro canto c'è un chip Linino AR 9331 che esegue Linux OpenWRT che è una distro per sistemi embedded, il chip Linino è collegato alla porta USB tipo A, allo slot per la scheda microSD, all'interfaccia Wi-Fi ed alla porta Ethernet.

I due componenti sono collegati tramite un bridge seriale.



Trovato il programma bridge e testato con il pin 13 (on board), eseguendo le richieste via browser funziona, si accende e si spegne.

```
// richiesta per accedere il led sull'arduino
http://10.20.4.111/cgi-bin/luci/arduino/digital/13/1
// richiesta per spegnere il led sull'arduino
http://10.20.4.111/cgi-bin/luci/arduino/digital/13/0
```

<https://forum.arduino.cc/index.php?topic=231399.0>

<http://arduinoindiarobomart.blogspot.com/2015/11/in-this-tutorial-we-are-learning-to-how.html>

Creata programma "blink" (ino/blink) per arduino yun, sia la parte di arduino che la parte di linux in python.

parte arduino:

```
#include <Bridge.h>
#include <stdio.h>

/**
 * Arduino blink on pin 13 sketch.
 */
```

```
* @author giuliobosco (giuliobva@gmail.com)
* @version 1.0 (2019-03-20 - 2019-03-20)
*/
// Here we will hold the values coming from Python via Bridge
char D13value[2];

void setup() {
    // Zero out the memory we're using for the Bridge
    memset(D13value, 0, 2);

    // Initialize digital pins 12 and 13 as output
    pinMode(13, OUTPUT);

    // Start using the Bridge
    Bridge.begin();
}

void loop() {
    // Write current value of D13 to the pin (basically turning it on or off)
    Bridge.get("D13", D13value, 2);
    int D13int = atoi(D13value);
    digitalWrite(13, D13int);

    // An arbitrary amount of delay to make the whole thing more reliable
    delay(10);
}
```

parte python:

```
#include <Bridge.h>
#include <stdio.h>

# Arduino blink on pin 13 sketch
#
# @author giuliobosco (giuliobva@gmail.com)
# @version 1.0 (2019-03-20 - 2019-03-20)

import sys
sys.path.insert(0, '/usr/lib/python2.7/bridge')

from time import sleep
from bridgeclient import BridgeClient as bridgeclient

value = bridgeclient()

for idx in range(0, 100):
    value.put('D13', '1')
    sleep(0.1)
    value.put('D13', '0')
    sleep(0.1)

print("I hope you enjoyed the light show\n")
```

Mattia

Alla fine il problema del JDBC lo ha risolto Paolo che sapeva come farlo e io mi sono occupato di riorganizzare il trello guardando ciò bisognava fare e ciò che mancava. Ho guardato poi i cambiamenti che ci sono stati nel git ed in fine mi sono informato su come fare ACC-Server.

Paolo

Oggi mi sono occupato di far funzionare JDBC il driver che permette di far comunicare Java e MySql.

Qui troverete la documentazione di come installare i driver suddivisa in ... semplici passaggi.

1 Scaricare i driver adatti a MySql

Per scaricare i driver corretti per MySql andare su questo sito: <https://dev.mysql.com/downloads/connector/j/>

selezionare il proprio sistema operativo (nel mio caso platform independent) in seguito scaricare la versione uguale a quella di MySql (consiglio 8.0.15).

2 Creare la classe per leggere i Dati

Bisogna creare una classe base di Java e aggiungere.

```
import java.sql.*;
```

Per usare i driver bisogna spostare la cartella .jar nella stessa posizione del .class del programma, aggiungere un collegamento al driver in questo modo:

```
Class.forName("com.mysql.cj.jdbc.Driver");
```

3 connettersi al Database

Per connettersi al database bisogna usare username e password quindi si dev'essere sicuri che il programma non sia accessibile a tutti, per connettersi si usa:

```
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/nomeDatabase", "username",  
"password");  
Statement st = con.createStatement();
```

4 Fare delle Query

Per fare delle Query bisogna semplicemente usare una stringa in sql per esmpio:

```
String query = "SELECT * FROM dati";  
ResultSet rs = st.executeQuery(query);  
while(rs.next()){  
    String name = rs.getString("name");  
}  
con.close();
```

Questa porzione di codice ritornerà tutti i nomi nella tabella Dati.

Possibili Errori ricevuti

Se vi dice che "Loading class com.mysql.jdbc.Driver. This is deprecated." è perché dalla nuova versione la stringa dentro Class.forName() contiene cj invece nelle vecchie versione non lo contiene.

Se vi dice che la zona oraria non è valida basta scrivere dentro a MySQL Workbench:

```
SET @@global.time_zone = '+01:00';  
SET @@session.time_zone = '+01:00';
```

Nel orario bisogna scrivere la propria zona.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro



Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

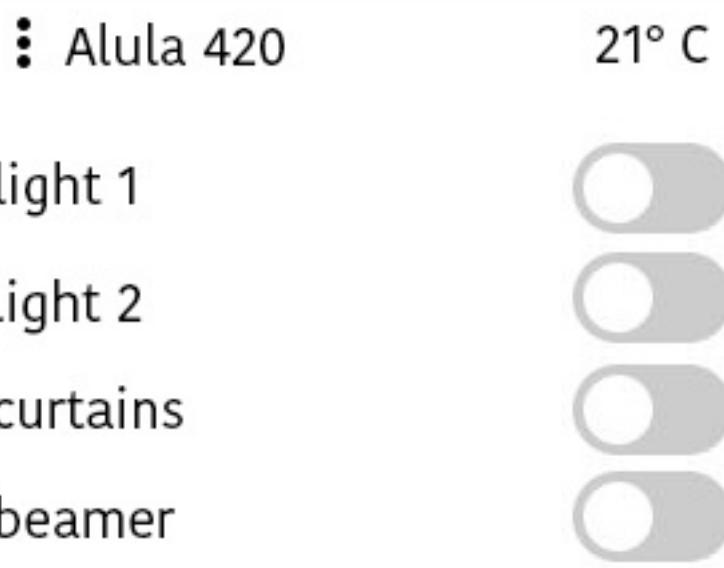
Canobbio, 22.03.2019

Lavori svolti

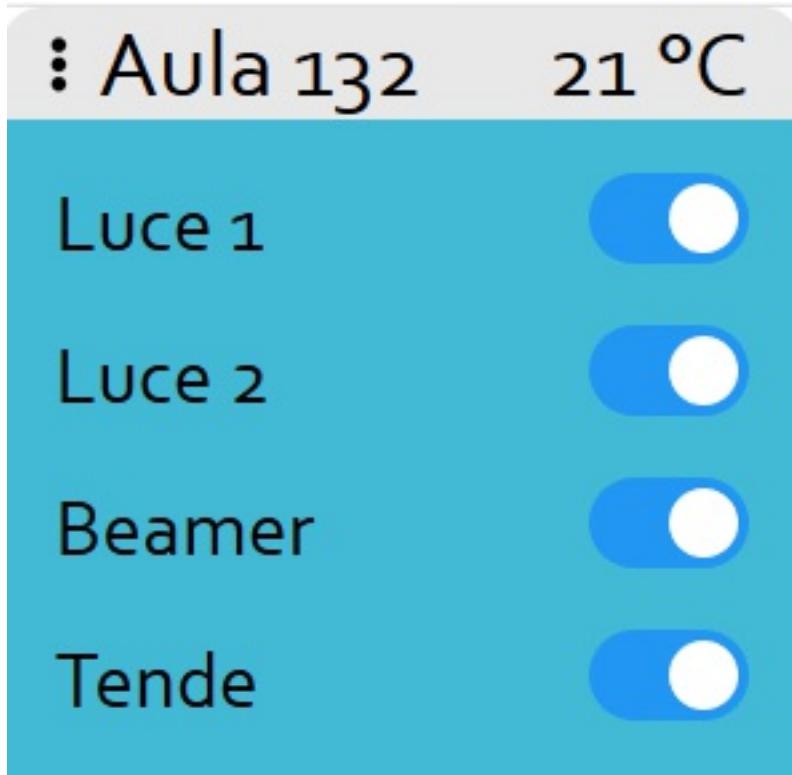
Giulio

Mattia

Oggi ho iniziato a fare la parte di frontend che riguarda la parte che riguarda le possibili opzioni per controllare le luci, il breamer e le tende dell'aula. Ciò che dovevo fare è questo:



Questo è ciò che ho fatto:



Adesso per la prossima volta ciò che devo fare è che quando si clicca appaiono e scompaiono i checkbox. Il tutto è stato fatto senza utilizzare librerie bootstrap.

Paolo

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 27.03.2019

Lavori svolti

Giulio

Diviso li file sorgente python da quelli arduino nelle rispettive cartelle `py` e `ino`.

Iniziato a sviluppare la parte di ACC-Client di arduino (ACC-Client.ino), configurazione dei PIN, cioè settare se i pin devo essere di input o output.

Cercato di capire come cambiare il valore di più PIN tramite python. Dopo diche iniziato a scrivere il codice, variabili per i valori dei pin, assegnare la memoria condivisa con OpenWRT e setup dei pin in INPUT e di quelli in OUTPUT.

```
#include <Bridge.h>
#include <stdio.h>

// assegnazione variabile
char D9value[2];

void setup() {
    // assegnazione memoria condivisa
    memset(D9value, 0, 2);

    // setup della modalita del pin
    pinMode(9, OUTPUT);
}

void loop() {
    // aggiornamento stato del pin 9
    Bridge.get("D9", D9value, 2);
    int D9int = atoi(D9value);
    digitalWrite(9, D9int);
}
```

Mattia

Oggi ho migliorato la grafica dell'interfaccia grafica della parte frontend per quanto riguarda i colori e tramite javascript ho fatto in modo che se si clicca compare o scompare la parte sotto con i checkbox. Poi abbiamo cambiato idea e invece di fare tutta la grafica a mano possiamo usare le librerie bootstrap quindi il resto della lezione mi sono studiato come utilizzare bootstrap.

```
function myFunction() {
    if(document.getElementById("aulaSettings").style.display == "block"){
        document.getElementById("aulaSettings").style.display = "none";
    }else{
        document.getElementById("aulaSettings").style.display = "block"
    }
}
```

Paolo

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 29.03.2019

Lavori svolti

Giulio

Copiato la VM per il server domotics sul server VMWare, del professor Barchi. Per eseguire questa operazione, il server gestisce solamente macchine compatibili con le macchine virtuali **VMWare working station 11.x**, quindi ho dovuto cambiare la compatibilità su VMWare Fusion, andando nelle impostazioni della VM, poi su compatibility, advanced options ed impostare **Use Hardware Version** su 11. Dopo di che bisogna esportare la VM in OVF, quindi cliccare su file, export selezionare il tipo di file OVF. Poi bisogna importare il file OVF sul server. Eseguendo questa operazione si riscontrava un errore, nella cifratura del disco virtuale. Il quale può venire risolto eliminando il file **.mf** presente nella cartella di esportazione della VM.

Mattia

Oggi ho dovuto fare un'intervista su skype con un possibile datore di lavoro per l'Inghilterra. Poi aggiunge le sliderbar per le luci nel pannello di controllo per l'aula. E poi Giulio mi ha passato l'arduino Yun e ho cominciato a documentarmi su come funziona, come deve essere utilizzato e i suoi sistemi di funzionamento.
<https://www.arduino.cc/en/Guide/ArduinoYun> Sono arrivato al capitolo open.WRT

Paolo

Oggi ho modificato il sito dashboard togliendo bootstrap e rifacendo da capo tutto. Ho creato un nav responsive con uno stile modulare per aggiungere elementi e mantenere sempre lo stesso design. Ho avuto un po' di problemi a crearlo responsive per colpa del css tag visibility.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 03.04.2019

Lavori svolti

Mattia

Oggi mi sono dedicato alla documentazione del progetto, in particolare ho fatto:

- 1.1 Informazioni sul progetto
- 1.3 Scopo
- 1.4 Analisi del dominio

Ho iniziato a fare l'analisi dei requisiti ma mi sono fermato perchè abbiamo fatto una riunione.

Dalla riunione è uscito fuori che non riusciremo ad implementare tutto il progetto completamente in particolare non riusciremo a fare molto probabilmente la parte fisica di implementazione dei moduli di controllo fisici alle luci, tende e beamer dell'aula non riusciremo a farli ma faremo delle simulazioni del prodotto ad esempio al posto delle luci dell'aula utilizzeremo dei led o dei motori per le tende. Inoltre dovremo documentare in maniera accurata come utilizzare l'arduino yun e il server in Tomcat.

Ci è stato richiesto di fare un analisi dei requisiti cambiando priorità e aggiungendo i requisiti. Quindi ho iniziato ad estrapolare tutti i requisiti dal trello ed infine li ho ordinati e formattati in delle tabelle di prova.

Analisi dei Requisiti

ID: REQ-01

Nome: Domotizzazione dell'aula

Priorità: 1

Versione: 1.0

Note:

Sotto requisiti

- 001: Scegliere quale arduino utilizzare
- 002: Gestione tende
- 003: Gestione beamer
- 004: Gestione luci
- 005: Front-end
- 006: Back-end
- 007: Guida di utilizzo

ID: REQ-02

Nome: Front-end

Priorità: 1

Versione: 1.0

Note:

Sotto requisiti

001: Sito Web

ID: REQ-03

Nome: Sito Web

Priorità: 1

Versione: 1.0

Note:

Sotto requisiti

001: modulo gestione dell'aula

002: modulo gestione delle aule

003: modulo luci/tende/beamer

004: modulo login

ID: REQ-04

Nome: Back-end

Priorità: 1

Versione: 1.0

Note:

Sotto requisiti

001: Server in python

002: Server in TomCat

003: Client in arduino

004: LDap connector per il login

005: JDBC per prendere i valori dal data base con java

ID: REQ-05

Nome: Guida di utilizzo

Priorità: 1

Versione: 1.0

Note:

Sotto requisiti

001: Guida di utilizzo dell'arduino yun

002: Guida di utilizzo e implementazione del server in Tomcat

ID: REQ-06

Nome: Simulazione controllo delle luci

Priorità: 1

Versione: 1.0

Note:

Sotto requisiti

001: Simulazione del controllo delle luci con l'arduino tramite dei led

ID: REQ-07

Nome: Simulazione controllo tende

Priorità: 3

Versione: 1.0

Note:

Sotto requisiti

001: Simulazione del controllo delle luci con l'arduino tramite dei motori

002: Decidere quali motori utilizzare

ID: REQ-08

Nome: Implementazione modulo fisico del beamer

Priorità: 3

Versione: 1.0

Note:

Sotto requisiti

001: Controllare le tende tramite l'arduino

ID: REQ-09

Nome: Implementazione modulo fisico delle luci

Priorità: 3

Versione: 1.0

Note:

Sotto requisiti

001: Controllare le luci attraverso l'arduino

ID: REQ-010

Nome: Implementazione modulo fisico delle tende

Priorità: 3

Versione: 1.0

Note:

Sotto requisiti

001: Controllare le tende attraverso l'arduino con dei motori

Giulio

Ho installato java e tomcat sulla VM, seguendo la guida [Install Tomcat 9.0.13](#), per CentOS7.

Dopo di che abbiamo avuto la riunione generale con il sore, dove abbiamo discusso il punto a cui siamo arrivati con il progetto, ed riaddattato gli obbiettivi, tra i quali i moduli delle tende e del beamer sono stati declassati a priorità 3, mentre ci sono due nuovi grandi obbiettivi, che sono la guida per Apache Tomcat e la guida per l'Arduino Yun.

Nella terza ora ho installato Apache Httpd, come reverse proxy, seguendo la guida proposta su:

<https://linuxconfig.org/how-to-set-up-apache-webserver-proxy-in-front-of-apache-tomcat-on-red-hat-linux>, con la differenza che i passi relativi a `tomcat` (e `tomcat-webapps`) come l'installazione tramite `yun` e l'avviamento del servizio. Ed ho aggiornato la guida, con il capitolo `configure apache reverse proxy`.

configurazione di apache:

```
<VirtualHost example.domain.com:80>
    ServerName domotics.domain.com

    ProxyRequests Off
    ProxyPass /examples ajp://localhost:8009/examples
    ProxyPassReverse /examples ajp://localhost:8009/examples
</VirtualHost>
```

Paolo

Oggi ho creato la classe idManager.java che permette di fare i controlli sugli identificativi base degli arduino (ip, key) usando gli id. Per fare questi controlli mi connetto al database e controllo che i dati corrispondono e se non

corrispondono li modifco così da ovviare problemi futuri. Per fare tutto questo uso i driver che spiego nel diario del 20.3.2019.

Questa classe permette all'utente di connetersi al database e riceve la connessione

```
public Connection connectDB(String user, String pass, String databaseURL) throws SQLException, ClassNotFoundException{
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = DriverManager.getConnection(databaseURL, user, pass);
    return conn;
}
```

Questa classe permette di fare una query sul database.

```
public ResultSet query(Connection c, String query) throws SQLException{
    // create the java statement
    Statement st = c.createStatement();

    // execute the query, and get a java resultset
    ResultSet rs = st.executeQuery(query);
    st.close();

    return rs;
}
```

Questo metodo permette di fare un update sul database

```
public void update(Connection c, String update) throws SQLException{
    Statement st = c.createStatement();
    st.executeUpdate(update);
}
```

Questo metodo controlla se l'ip è cambiato in caso lo aggiorna.

```
public void checkIp(Connection c, String id, String ip) throws SQLException{
    ResultSet rs = query(c, "SELECT ip from arduino where client_id="+id);
    if(ip != rs.getString("ip")){
        update(c,"update arduino set ip =" +ip+ " where client_id=" +id);
    }
}
```

Questa classe controlla se la key esiste se non esiste usa il metodo create key

```
public String checkKey(Connection c, String id) throws SQLException{
    ResultSet rs = query(c, "SELECT ip from arduino where client_id=" +id);
    if(rs.getString("client_key") != null){
        return rs.getString("client_key");
    }
    ResultSet ds = query(c, "SELECT ip from arduino where client_id=" +id);
    boolean flag = true;
    String key = "";
    while(flag){
        key = createKey();
        flag = false;
        while(ds.next()){
            if(ds.getString("client_key").equals(key)){
                flag = true;
            }
        }
    }
    return key;
}
```

Questa classe crea un codice esadecimale che viene inserito se manca la key.

```
public String createKey(){
    char[] hex = {'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};
```

```
String result = "";
for(int i = 0; i < 12; i++){
    int rand = (int)(Math.random()*16);
    result += hex[rand];
}
return result;
```

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 05.04.2019

Lavori svolti

Giulio

installato git sulla VM, poi configurato il proxy per git

```
yum install git  
git config --global http.proxy "http://username:password@proxy.host.local:port"
```

Per poter scaricare da github il codice di domotics.

Leggere il valore di un pin analogico dell'arduino tramite bridge con python sulla parte OpenWRT dell'arduino.

<https://www.arduino.cc/en/Tutorial/Bridge>

<https://www.arduino.cc/en/Reference/YunPut>

<https://www.arduino.cc/en/Reference/YunBridgeLibrary>

<http://forum.arduino.cc/index.php?topic=363376.0>

<http://forum.arduino.cc/index.php?topic=345921.0>

Sistemato il codice dell `IdManager`, javadoc e code style.

```
public class IdManager {  
    /**  
     * Get the connection to the database.  
     *  
     * @param username Database username.  
     * @param password password  
     * @param server Database server address.  
     * @param port Database port.  
     * @param database Database.  
     * @return Database connection.  
     * @throws SQLException Error with the sql server.  
     * @throws ClassNotFoundException Jdbc Driver not found.  
     */  
    public Connection getDbConnection(String username, String password, String server, int port,  
        String database) throws SQLException, ClassNotFoundException {  
        String connectionString = "jdbc:mysql://" + server + ":" + port + "/" + database;  
        Class.forName("com.mysql.jdbc.Driver");  
        return DriverManager.getConnection(connectionString, username, password);  
    }  
}
```

In questa parte di codice, ho scorporato la stringa di connessione, in maniera che l'utente debba solamente fornire i dati di connessione senza preoccuparsi della struttura della connection string.

Creata la class `JdbcConnector`, la quale permette di riutilizzare la stessa connessione al database in più operazioni. Ed integrato la classe `IdManager` con il `JdbcConnector`.

Implementato le classi modello del sistema, `Room`, `Arduino` e `Light`.

Mattia

Io mi sono concentrato sulla documentazione, ho intabellato l'analisi dei requisiti, ho scritto le informazioni sul progetto, l'abstract, l'analisi di dominio, lo scopo. Ho iniziato l'analisi dei mezzi e la progettazione in particolare ho

prese lo schema generale del progetto e l'ho descritto.

Paolo

Ho iniziato ad installare i software di base sul server, come MariaDB e Java, questo per poter testare JdbcConnector.java sul Server ma ho avuto molti problemi che mi hanno impedito di fare il test tra cui un interprete di un link http per scaricare le librerie di jdbc.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 13.02.2019

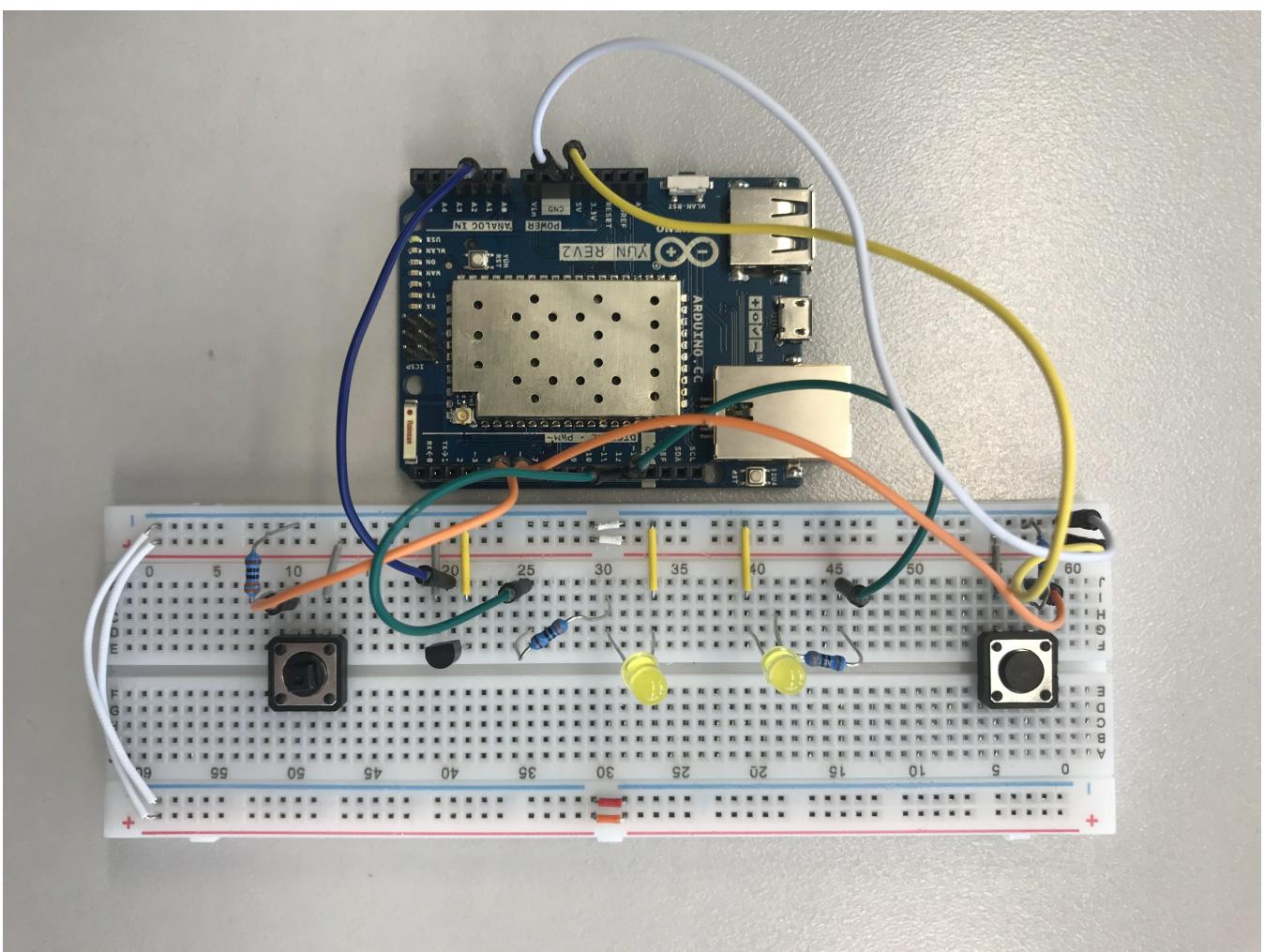
Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 10.04.2019

Lavori svolti

Giulio

Montato la base degli elementi di domotics sulla breadboard, cioè due led e due bottoni.



Sul montaggio era già presente anche un sensore di temperatura, il quale faceva andare in corto circuito l'arduino. Per questo motivo è stato sostituito con un sensore [Thermistor Kit\(2pcs\)](#). Il quale è stato messo in serie ad una resistenza.

ACC-Client.ino

Dopo di che ho implementato L'ACC-Client.ino, che funziona correttamente.

Per i pin di output prende i valori dagli indirizzi di memoria e li aggiorna sui pin.

```
//inizializzo la variabile di buffer
char D13value[2];
// assegno la dimensione al buffer
memset(D13value, 0, 2);
// prendere il valore di d13 e lo inserisco nel buffer
Bridge.get("D13", D13value, 2);
// trasformo in int il valore string
int D9int = atoi(D13value);
// scrivo sul pin output il valore
digitalWrite(13, D9int);
```

<https://www.arduino.cc/en/Reference/YunGet>

Mentre per i pin di input ho utilizzato due differenti strategie, fra i pin digitali e quelli analogici, quelli digitali funzionano in maniera simile a quelli di output, con la differenza che eseguono l'operazione inversa, quindi

```
// inizializzo la variabile di buffer
char D6value[2];
// assegno la dimensione al buffer
memset(D6value, 0, 2);
// prendere il valore del pin 6 e scriverlo nella variabile D6int
int D6int = digitalRead(6);
// trasformare il valore in stringa con buffer
itoa(D6int, D6value, 2);
//scrivo il valore in d6
Bridge.put("D6", D6value);
```

Per arrivare a questa soluzione funzionante ho utilizzato le seguenti pagine

- <http://www.cplusplus.com/reference/cstdlib/itoa/>
- <https://www.arduino.cc/en/Reference/BridgeClientConstructor>
- <https://forum.arduino.cc/index.php?topic=188998.0>
- <https://www.arduino.cc/en/Reference/YunPut>

Mentre per i pin analogici ho utilizzato una strategia molto più semplice, trasformare il valore di input direttamente in una stringa e scriverla nella variabile A0 o A1.

```
// aggiorno stato della variabile con valore della porta
Bridge.put("A0", String(analogRead(0)));
```

Ho trovato questa soluzione su: <http://forum.arduino.cc/index.php?topic=286400.0>

Mentre dal lato di OpenWRT ho scritto questo semplice script in python che alterna i due led e stampa sulla console il valore delle porte.

```
import sys
sys.path.insert(0, '/usr/lib/python2.7/bridge')

from time import sleep
from bridgeclient import BridgeClient as bridgeclient

value = bridgeclient()

while True:
    value.put('D13', '1')
    value.put('D12', '0')
    value.put('D11', '1')
    sleep(0.1)
    value.put('D13', '0')
    value.put('D12', '1')
    value.put('D11', '0')
    sleep(0.1)
    print(value.get('D6') + " " + value.get('D5') + " " + value.get('A0') + " " + value.get('A1'));
```

Mattia

Malato

Paolo

Ho aiutato Giulio con il montaggio del Arduino prendendo i componenti che c'erano bisogno. In seguito ho aggiornato lo schema con i nuovi componenti e ho sistemato il cable managment sullo schema.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 12.04.2019

Lavori svolti

Giulio

Ho creato la funzione per trasformare il valore del sensore di temperatura `thermistor` letto sulla porta dell'arduino in volt, in un valore in gradi celsius (Questa operazione verrà eseguita in python su OpenWRT). Per trasformare il voltaggio in gradi celsius bisogna utilizzare la seguente formula:

$$T = \frac{1}{\ln\left(\frac{RT}{R0}\right) + \frac{1}{B}} - T0$$

Che ho trovato su: https://create.arduino.cc/projecthub/Marcazzan_M/how-easy-is-it-to-use-a-thermistor-e39321 dopo di che ho riscritto la formula presente nella pagina da codice arduino a python

```
RT0 = 10000          # thermistor datasheet values
B = 3977            # thermistor datasheet values
# ---
VCC = 5             # supply voltage
R = 10000           # resistor: 10kOhm
# ---
T0 = 25 + 273.15   # temperature T0 from thermistor datasheet
# ---
VRT = float(brige.get('A1'))    # read value of the pin A1 of the arduino
VRT = (5.00 / 1023.00) * VRT   # conversion to voltage
VR = VCC - VRT           # inverse of voltage
RT = VRT / (VR / R)       # resistence of RT
# ---
ln = math.log(RT / RT0)     # logarithm of RT by RT0
TX = (1 / ((ln / B) + (1 / T0))) # temperature on the thermistor
# ---
TX = TX - 273.15          # conversion from kelvin to celsius
```

per riscrivere il codice ho dovuto utilizzare qualche accorgimento, come la funzione `log`, che in python è nella libreria `math`. Mentre per trasformare la stringa presa dal bridge ho trovato la soluzione su <https://www.digitalocean.com/community/tutorials/how-to-convert-data-types-in-python-3> che sarebbe utilizzare la funzione `float("5.1")`, mentre la funzione inversa per poter concatenare un float con una stringa bisogna utilizzare la funzione `str(4.1)`.

Il datasheet del thermistor è disponibile al seguente link: <http://www.farnell.com/datasheets/2166231.pdf>

Prima ho creato un programma per capire come funziona il sensore `thermistor`, dopo di che ho implementato il modulo intero. `py/acc_client.py/thermistor`.

C Thermistor

```
float RT0
float B
float T0
float vcc
float r

void __init__(float vcc, float r)
float getKelvin(voltage)
float getCelsius(voltage)
```

Test del modulo (temperatura ambiente: 24°C):

Durante il test del modulo la funzione `getCelsius` non funzionava, mentre quella `getKelvin` funzionava. La funzione `getCelsius` doveva richiamare la funzione `getKelvin` e sottrargli 273.15, la sottrazione non veniva eseguita, quindi la funzione era sbagliata. Dopo aver sistemato questo problema il modulo funzionava correttamente.

Infine ho disegnato il modulo di controllo delle luci dalla parte di Arduino OpenWRT

C Lights

```
void __init__(bridge)
void lightSet(pin, value)
void lightOn(pin)
void lightOff(pin)
```

La quale ha un metodo per settare lo stato dei pin, uno per accendere ed uno per spegnere il pin.

```
class Lights:
    # create light controller with the arduino bridge
    def __init__(self, bridge):
        self.bridge = bridge

    # light set value
    def lightSet(self, pin = '13', value = '0'):
        self.bridge.put('D' + str(pin), str(value))

    # turn light on
    def lightOn(self, pin = '13'):
        self.lightSet(pin, '1')

    # turn light off
    def lightOff(self, pin = '13'):
        self.lightSet(pin, '0')
```

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione



In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 17.04.2019

Lavori svolti

Giulio

Ho risolto il problema del JdbcConnector, che non aveva l'orario diverso dal server. per risolvere questo problema ho trovato la soluzione su: <https://stackoverflow.com/questions/26515700/mysql-jdbc-driver-5-1-33-time-zone-issue> la quale è composta di aggiungere la seguente stringa alla stringa di connessione al database:

```
?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC
```

Scrittura delle classi `Room`, `Arduino` e `Light` dell ACC-Server. Le quali rappresentano le tabelle del database. Ed anno anche quale metodo specifico. Per esempio i metodi per tradurre un ResultSet di MySQL in una lista di oggetti Arduino.

```
public static List<Arduino> getArduinos(ResultSet sqlResultSet) throws SQLException {
    List<Arduino> arduinos = new ArrayList<>();

    while (sqlResultSet.next()) {
        arduinos.add(new Arduino(
            sqlResultSet.getString("client_id"),
            sqlResultSet.getString("ip"),
            sqlResultSet.getString("client_key"),
            sqlResultSet.getString("root_password"),
            Room.get(sqlResultSet.getString("room"))
        ));
    }

    return arduinos;
}
```

Iniziato a scrivere l'ACC-Client.py, creando le classi:

KeyManager

```
class KeyManager:
    def __init__(self, key=''):
        self.key = key

    def check_key(self, key):
        return self.key == key

    def get_key(self):
        return self.key
```

HttpServer

```
class HttpServer(BaseHTTPRequestHandler):
    def send_header_json(self):
        self.send_response(200)
        self.send_header("Content-type", "text/json")
        self.end_headers()
```

```
def do_GET(self):
    self.send_header_json()
    self.wfile.write(bytes("{\"status\":\"OK\", \"key\":\""+ self.server.key_manager.get_key() +
    "\"}", "utf-8"))
```

AccClient

```
class AccClient:
    @staticmethod
    def start(key_manager):
        try:
            server = HTTPServer((hostName, hostPort), HttpServer)
            server.key_manager = key_manager
            print(time.asctime(), "Server Starts - %s:%s" % (hostName, hostPort))
            server.serve_forever()
        except KeyboardInterrupt:
            server.server_close()
            pass
```

In fine ho iniziato ad implementare la classe `ResponseRender` utilizzata da `HttpServer`, per eseguire le richieste del server ed elaborare la risposta.

Mattia

Paolo

Oggi ho sistemato un paio di problemi con il diagramma fritzing del arduino e insieme a Mattia abbiamo messo a posto il Gantt togliendo le parti impossibili da finire per le tempistiche.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 03.05.2019

Lavori svolti

Giulio

Sistemato il KeyManager del ACC-Client.py, quando si controlla la chiave controlla anche l'indirizzo del server.

```
if len(self.key) > 0 and len(self.server_address) > 0:  
    return self.key == key
```

Poi ho documentato e commentato tutto il codice dell'ACC-Client.py.

Dopo di che ho aggiunto e testato al modello Java delle luci i metodi per l'utilizzo dei JSON e la richiesta dello stato delle luci, quindi eseguire una richiesta all'arduino per richiedere lo stato del pin.

```
/**  
 * Get the request string of the status of the light.  
 *  
 * @return Url get request string.  
 */  
private String getRequestString() {  
    return "http://" + this.arduino.getIp() + ":8080/acc?key=" + this.arduino.getKey() + "&pin=" +  
this.pin;  
}  
  
/**  
 * Get the actual status of the arduino.  
 *  
 * @return Status of the arduino. -1 if IOException or NumberFormatException.  
 * @throws IOException Error while http request.  
 */  
public int getStatus() throws IOException {  
    try {  
        String response = GetRequest.get(this.getRequestString());  
        JSONObject jo = new JSONObject(response);  
        return Integer.parseInt(jo.getString("message"));  
    } catch (MalformedURLException | NumberFormatException ignored) {  
        return -1;  
    }  
}
```

Oltre allo stato della luce, ho implementato i metodi cambiare (toggle) lo stato della luce.

```
/**  
 * Toggle the status of the light.  
 *  
 * @throws IOException Error while http request.  
 */  
public void toggleLight() throws IOException {  
    if (getStatus() == this.LIGHT_ON) {  
        turnOff();  
    } else {  
        turnOn();  
    }  
}
```

Dopo di che ho creato la classe LightButton, che rappresenta i bottoni fisici collegati all'arduino.

```
/**  
 * Check if the pin is valid.  
 *  
 * @param pin      Pin to check.  
 * @param connector Connector to domotics database.  
 * @throws SQLException Error on the MySQL database.  
 * @throws IOException Error with the http get request.  
 */  
private void checkPin(int pin, JdbcConnector connector) throws SQLException, IOException {  
    String query = "SELECT * FROM domotics.lightButton WHERE pin='" + pin + "' AND arduino='" +  
    this.arduino.getId() + "';";  
    ResultSet rs = connector.query(query);  
    rs.next();  
    int dbPin = rs.getInt("pin");  
    if (dbPin != pin) {  
        String message = "On arduino: " + this.arduino.getId() + " no light button on pin: " + pin;  
        throw new IOException(message);  
    }  
}  
  
/**  
 * Load the light of the button.  
 *  
 * @param connector Connector to domotics database.  
 * @throws SQLException      Error on the MySQL  
 * @throws ClassNotFoundException MySQL Driver class not found.  
 */  
private void loadLight(JdbcConnector connector) throws SQLException, ClassNotFoundException {  
    String query = "SELECT l.pin, l.arduino FROM domotics.lightButton lb JOIN light l ON lb.lightPin =  
    l.pin WHERE lb.pin='" + this.buttonPin + "' AND lb.arduino='" + this.arduino.getId() + "';";  
    ResultSet rs = connector.query(query);  
    rs.next();  
    int pin = rs.getInt("pin");  
    Arduino arduino = new Arduino(connector, rs.getString("arduino"));  
    this.light = new Light(pin, arduino);  
}
```

Ed ho configurato i LightButton con l'AccServlet, per poter esaudire le richieste dell'Arduino quando un bottone viene premuto.

```
if (parameters.containsKey("key")  
    && parameters.containsKey("pin")  
    && parameters.containsKey("set")) {  
  
String arduinokey = parameters.get("key")[0];  
String arduinoIp = request.getRemoteAddr();  
  
String buttonPinString = parameters.get("pin")[0].substring(1);  
int buttonPin = Integer.parseInt(buttonPinString);  
  
int setStatus = Integer.parseInt(parameters.get("set")[0]);  
LightButton lb = new LightButton(buttonPin, arduinoIp, arduinokey, jdbc);  
Light light = lb.getLight();  
  
if (setStatus == light.LIGHT_ON) {  
    light.turnOn();  
} else if (setStatus == light.LIGHT_OFF) {  
    light.turnOff();  
}
```

Mattia

Oggi mi sono concentrato per quattro ore sulla documentazione, ho riguardato le cose che c'erano già, poi ho fatto il diagramma uml del JDBC e ho documentato la progettazione di esso, ho fatto quasi tutti i punti della progettazione, ho completato l'Arduino Connection Controller Client che però non ho ben capito come funziona quindi è un attimo in pausa, dell'arduino Yun, , mi manca l'architettura dell'Arduino Connection Controller Server, del server Domotics e del Web Server in Tomcat quindi non li ho potuti fare. Poi ho iniziato a documentare l'implementazione e ho completato LDAP e Database e ho iniziato a fare JDBC.

Paolo

Oggi ho iniziato ha cambiare il frontend per renderlo modulare tramite AngularJS, quindi ho scaricato AngularJS 1.7.8 da <https://angularjs.org> dove mi sono anche documentato per riuscire a caricare come delle views la pagina AulaModel.htm, e per ripeterla per ogni aula.

```
var app = angular.module("App", ["ngRoute"]);
app.config(function($routeProvider) {
    $routeProvider
        .when("/", {
            templateUrl : "AulaModel.htm"
        });
});

angular.module('ngRepeat', []).controller('repeat', function($scope) {
    $scope.rooms = [
        {name:'Aula 323', temp: 21},
        {name:'Aula 423', temp: 22},
        {name:'Aula 422', temp: 23}
    ];

    $scope.lights = [
        {desc:'porta', value: true},
        {desc:'finestra', value: false},
    ]
});
```

La prima funzione servirà per gestire le view della pagina, la seconda funzione invece gestisce la creazione delle aule a seconda del file json.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 13.02.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 08.05.2019

Lavori svolti

Giulio

Aggiunto il controllo di esistenza del LightButton prima di eseguire le query sul ResultSet.

```
private void checkPin(int pin, JdbcConnector connector) throws SQLException, IOException {
    String query = "SELECT * FROM domotics.lightButton WHERE pin=" + pin + " AND arduino=" + this.arduino.getId() + ";" ;
    ResultSet rs = connector.query(query);

    if (rs.next()) {
        int dbPin = rs.getInt("pin");
        if (dbPin != pin) {
            String message = "On arduino: " + this.arduino.getId() + " no light button on pin: " + pin;
            throw new IOException(message);
        }
    } else {
        throw new SQLException("Pin not found");
    }
}
```

Creata JSON builder, utilizzato per le risposte standard di ACC.

```
public class JsonBuilder {

    public static String ERROR = "ERROR";

    public static String OK = "OK";

    public static String getJsonResponse(String status, String message) {
        JSONObject jo = new JSONObject();
        jo.put("status", status);
        jo.put("message", message);
        return jo.toString();
    }

    public static String getJsonResponseError(String message) {
        return getJsonResponse(ERROR, message);
    }

    public static String getJsonResponseOk(String message) {
        return getJsonResponse(OK, message);
    }
}
```

Creata il ControllerServlet, utilizzato per esaudire le richieste del front-end, per esempio accendere e spegnere una luce.

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    Map<String, String[]> parameters = request.getParameterMap();
    String responseString = "";
    try {
        JdbcConnector jdbc = DomoticsJdbcC.getConnector();
        jdbc.openConnection();
```

```
if (parameters.containsKey("light") && parameters.containsKey("toggle")) {  
    String lightPin = parameters.get("light")[0];  
    int xIndex = lightPin.indexOf('x');  
    String arduinoId = lightPin.substring(0, xIndex);  
    int pin = Integer.parseInt(lightPin.substring(xIndex + 1));  
    System.out.println(pin + " a " + arduinoId);  
  
    Arduino arduino = new Arduino(jdbc, arduinoId);  
    Light light = new Light(pin, arduino);  
    light.toggleLight();  
    System.out.println(light.getJson());  
    responseString = JsonBuilder.getJsonResponseOk("");  
} else {  
    responseString = JsonBuilder.getJsonResponseError("no light");  
}  
  
} catch (SQLException | ClassNotFoundException e) {  
    responseString = JsonBuilder.getJsonResponseError(e.toString());  
}  
  
response.getOutputStream().println(responseString);  
}
```

Modificato tutti i metodi `getJson()` delle classi modello, il metodo ritorna un `JSONObject`, mentre i metodi `getJsonString()` ritorna l'oggetto json in stringa.

Mattia

Anche oggi ho continuato con la documentazione, ho cambiato completamente i requisiti perchè così com'erano non mi convincevano, poi ho aggiunto i software e gli hardware utilizzati, ho riletto tutta la documentazione e ho sistemato ciò che c'era di sbagliato o gli errori.

Paolo

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 10.05.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 10.05.2019

Lavori svolti

Giulio

Rimosso le aperture di connection dalle classi di modello, questa operazione deve essere eseguita solamente una volta all'inizio del suo utilizzo.

Ho iniziato a scrivere la documentazione sul funzionamento dell'arduino YUN.

Modificato l'acc per eseguire il comando toggle:

```
while not self.is_interrupted():
    if self.is_status_changed():
        if self.is_status_changed() and self.status == '1':
            print(self.status)
            # if status changed execute request
            response = self.execute_http_req()
            print(response)
    sleep(0.1)
```

Sistemato l'auto reload della pagina web.

Ricaricando la pagina ogni volta, i dati non vengono aggiornati, perché vengono inseriti sempre i dati salvati in cache, quindi ho escogitato il metodo per richiederli ogni volta, aggiungendo una versione alla richiesta (che dal lato server non viene gestita, ma obbliga il client richiedere i dati al server, e non utilizzare la cache).

Controller:

```
// setto il controller
app.controller('RoomsController', ['$scope', '$sce', 'RoomsService', function ($scope, $sce,
roomsService) {

    // versione della richiesta
    var versionCount = 0;
    // caricare i valori dal service
    function load() {
        roomsService.getRooms(versionCount).then(function (data) {
            // se sessione scaduta invia alla pagina di login
            if (data.logout && data.logout == 'logout') {
                window.location = "index.html#/login";
            }

            // aggiorna i dati nella pagina
            $scope.rooms = data.rooms;
        });
        // aggiorna la versione della richiesta
        versionCount++;
    }

    // carica i dati
    load();
    // setta intervallo con carico dei dati ogni 5000 ms
    setInterval(load, 5000);
}]);
```

Service:

```
// configuro factory
app.factory('RoomsService', ['$http', function($http) {
    // lista dei servizi
    var service = {};
    // url di base per richieste
    var urlBase = "/data/rooms";

    // get room service
    service.getRooms = function (version) {
        // creo url con url di base e versione
        var url = urlBase + "?v=" + version;
        // eseguo richiesta e ritorno la risposta del server
        return $http({
            method: 'GET',
            url: url
        }).then(function (response){
            return response.data;
        },function (error){
            return error;
        });
    };
    return service;
}]);
```

Mattia

Sistemato descrizione del database nella progettazione, aggiornato nuovamente i requisiti, ho sistemato la guida su tomcat, ho crato la guida di utilizzo che allegheremo al tutto sul database spiegando ogni singolo passaggio, ho creato lo schema fritzing della breadboard del arduino che viene usata per fare la simulazione e ho aggiunto il sommario.

Paolo

Oggi ho sistemato tutti i commenti nei programmi, in seguito ho iniziato a corregere anche i diari e la doc.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 15.05.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 15.05.2019

Lavori svolti

Giulio

Mattia

Ho aggiunto una descrizione ai requisiti spiegando ciò che non siamo riusciti a fare, ho modificato i requisiti perché mi sono nuovamente accorto che i sottorequisiti non mi convincevano, ho aggiunto un'introduzione alla progettazione di LDAP, ho creato la guida che documentava tutto su LDAP, dal perché lo abbiamo scelto, dalla progettazione all'implementazione.

Paolo

Ho continuato la doc e ho iniziato la guida del frontend. Ho revisionato la guida del database/JDBC e la guida TomcatCentOS.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 17.05.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 17.05.2019

Lavori svolti

Giulio

Oggi ho passato le prime 3 ore a cercare di eseguire il progetto sul server, ma ho avuto dei problemi con le librerie, non funzionavano sia le librerie relative a mysql (`mysql-connector-java-8.0.13.jar`), che quelle di json (`jsonio.jar`).

Il problema si presentava quando veniva uploadato il file `.war` (che contiene il progetto compilato per applicazioni java web), e veniva avviato si presentava un errore, il quale indica che non viene trovata la classe `JSONObject` (contenuta in `jsonio.jar`).

```
17-May-2019 15:24:54.278 SEVERE [http-nio-8080-exec-9]
org.apache.catalina.core.ApplicationContext.log FAIL - Application at context path [/web_war]
could not be started
    org.apache.catalina.LifecycleException: Failed to start component
[StandardEngine[Catalina].StandardHost[localhost].StandardContext[/web_war]]
        at org.apache.catalina.util.LifecycleBase.handleSubClassException(LifecycleBase.java:441)
        at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:198)
        at org.apache.catalina.manager.ManagerServlet.start(ManagerServlet.java:1395)
        at org.apache.catalina.manager.HTMLManagerServlet.start(HTMLManagerServlet.java:698)
        at org.apache.catalina.manager.HTMLManagerServlet.doPost(HTMLManagerServlet.java:223)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:660)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:741)
        at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:231)
        at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
        at
org.apache.catalina.filters.CsrfPreventionFilter.doFilter(CsrfPreventionFilter.java:136)
        at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)
        at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
        at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:53)
        at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)
        at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
        at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:199)
        at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:96)
        at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:607)
        at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:139)
        at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:92)
        at
org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.java:668)
        at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:74)
        at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:343)
        at org.apache.http11.Http11Processor.service(Http11Processor.java:408)
        at org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:66)
        at org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:791)
        at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1417)
        at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
        at java.lang.Thread.run(Thread.java:748)
Caused by: java.lang.NoClassDefFoundError: org/json/JSONObject
```

```
at java.lang.Class.getDeclaredMethods0(Native Method)
at java.lang.Class.privateGetDeclaredMethods(Class.java:2701)
at java.lang.Class.getDeclaredMethods(Class.java:1975)
at org.apache.catalina.util.Introspection.getDeclaredMethods(Introspection.java:133)
at
org.apache.catalina.startup.WebAnnotationSet.loadMethodsAnnotation(WebAnnotationSet.java:285)
at
org.apache.catalina.startup.WebAnnotationSet.loadApplicationServletAnnotations(WebAnnotationSet.java:138)
at
org.apache.catalina.startup.WebAnnotationSet.loadApplicationAnnotations(WebAnnotationSet.java:69)
at
org.apache.catalina.startup.ContextConfig.applicationAnnotationsConfig(ContextConfig.java:328)
at org.apache.catalina.startup.ContextConfig.configureStart(ContextConfig.java:772)
at org.apache.catalina.startup.ContextConfig.lifecycleEvent(ContextConfig.java:299)
at org.apache.catalina.util.LifecycleBase.fireLifecycleEvent(LifecycleBase.java:123)
at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5007)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:183)
... 30 more
Caused by: java.lang.ClassNotFoundException: org.json.JSONObject
at
org.apache.catalina.loader.WebappClassLoaderBase.loadClass(WebappClassLoaderBase.java:1328)
at
org.apache.catalina.loader.WebappClassLoaderBase.loadClass(WebappClassLoaderBase.java:1157)
... 43 more

17-May-2019 15:24:54.278 INFO [http-nio-8080-exec-9]
org.apache.catalina.core.ApplicationContext.log HTMLManager: list: Listing contexts for virtual
host 'localhost'
```

Per cercare di risolvere questo problema ho le seguenti soluzioni:

- incluso la libreria nel file `.war`
- copiato la libreria nelle librerie di tomcat
- copiato la libreria nella cartella di avvio di tomcat (dove vi è la libreria `tomcat.jar`)
- copiato la libreria nella cartella della JRE
- aggiunto la libreria alla CLASSPATH di java
- aggiunto la libreria alla CLASSPATH di tomcat
- forzato l'avvio di tomcat con la libreria
- aggiunto la libreria alla libreria interna di tomcat (`tomcat.jar`)

Nessuna di queste soluzioni ha avuto un esito positivo.

Dopo diche ho scritto la guida sull'ACC.

Mattia

Ho completato definitivamente la guida di LDAP mettendo apposto alcune cose, ho aggiunto una lista nell'introduzione dei requisiti per illustrare tutti i cambiamenti e le modifiche che ci sono state durante il corso del progetto. Abbiamo fatto una riunione con il docente per vedere il punto della situazione, ciò che mancava, ciò che potevamo aggiungere e abbiamo discusso di alcuni quesiti che ci eravamo posti durante il progetto. Ho aggiunto la progettazione che abbiamo fatto su come avremmo implementato tutta la comunicazione e i protocolli implementati. Ho iniziato a fare i test case dei requisiti.

Paolo

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione



In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Domotics | Diario di lavoro - 22.05.2019

Ruberto Mattia, Paolo Guebeli, Bosco Giulio

Canobbio, 17.05.2019

Lavori svolti

Oggi stiamo rifinendo le guide, finiendo la presentazione, sistemando l'arduino per la demo, con la luce sulla breadboard. Dopo di che abbiamo stampato tutto quanto e rilegato.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Informarsi e capire cosa e come si devono fare le varie cose.

Guida Modulo ACC

Capitolo	Pagina
Indice	2
ACC (Arduino Connection Controller)	3
ACC-Server - autoconf	3
ACC-Server - set	4
ACC-Server - set toggle	4
ACC-Server - il codice	4
ACC-Client (Arduino Connection Controller - Client)	5
ACC-Client-ID	5
ACC-Client-Key	5
ACC-Client - alive	5
ACC-Client - set	5
ACC-Client - get	6
ACC-Client - il codice	6

ACC (Arduino Connection Controller)

È un protocollo che abbiamo ideato per comunicare con facilità con l'Arduino, mentre lo stavamo progettando ci siamo accorti che questo protocollo può essere esteso per qualunque micro controllore che possa essere connesso ad una rete LAN, siccome è basato sul protocollo HTTP, per poterlo utilizzare basta un server HTTP personalizzato ed un client HTTP per eseguire le richieste al server ACC.

Il server HTTP e il client, sono entrambi sia sul server ACC che sul client ACC (Microcontrollore).

Per semplicità il server verrà chiamato **ACC-Server** mentre il lato client verrà denominato **ACC-Client**.

Questo protocollo, come un modulo a sé stante, dal progetto **domotics**, quindi potrà venir utilizzato anche da altri progetti.

L'idea del funzionamento di questo modulo, è poter inviare dei valori da settare sui pin, oppure richiedere lo stato dei pin. Questo protocollo deve funzionare in maniera "sicura" se vi è presente un ACC-Server, oppure in maniera autonoma se il suo server.

La modalità **sicura** utilizza una chiave per scambiare i valori fra l'ACC-Client e l'ACC-server, la chiave è una stringa esadecimale di 12 caratteri. Quando la modalità sicura è abilitata, per richiedere i valori al microcontrollore o settare dei valori sui pin, mentre nella modalità senza l'ACC-Server, chiunque conosce l'indirizzo IP del server ed il funzionamento del protocollo può inviare comandi o richiedere valori all'ACC-Client.

ACC-Server (Arduino Connection Controller - Server)

L'ACC-Server, è composto di un server HTTP ed un elemento per creare le richieste HTTP. Il server HTTP, ha bisogno di una pagina, la quale deve essere in grado di interpretare due richieste:

- **autoconf**: Questa richiesta richiede tramite il suo ID, la quale ritorna la chiave di comunicazione.
- **set**: Questa richiesta deve contenere, la chiave di comunicazione, il pin ed il valore, questa serve per aggiornare l'ACC-Server nel caso in cui un pin (per esempio bottone), cambia stato.

L'ACC-Server, si basa sul un database, che viene utilizzato per risolvere le richieste, per esempio quale input deve accendere o spegnere quale luce. Per il database è stata fatta una guida apposita:

[doc/Documentazione/Guide/GuidaFunzionamentoDatabase.md](#).

ACC-Server - autoconf

La richiesta deve essere:

```
http://<serverAddress>:<serverPort>/acc?autoconf&id=<ACC-Client-ID>
```

E la risposta sarà

```
{"id": "<ACC-Client-ID>", "key": "<ACC-Client-KEY>", "server_address": "<serverAddress>:<serverPort>"}
```

Tutte le risposte saranno inviate in formato JSON, questo per facilitare il l'interpretazione da parte del client.

ACC-Client-ID e ACC-Client-KEY sono spiegati nel capitolo successivo

ACC-Server - set

Quando cambia lo stato di un pin digitale di input sull'arduino, (per esempio la pressione di un bottone) questo deve notificarlo al server, per permettere al server di eseguire le giuste operazioni, per esempio modificare lo stato di altri pin.

La richiesta deve essere:

```
http://<serverAddress>:<serverPort>/acc?key=<ACC-Client-KEY>&pin=<changedPin>&set=<pinStatus>
```

La risposta sarà:

```
{"status":"OK","message":"<valoreSettato>"}
```

ACC-Server - set toggle

In alcuni casi potrebbe essere comodo avere una funzione toggle, per esempio per i bottoni, eseguire la richiesta **set**, la quale semplicemente indica che il pin ha cambiato stato. Per eseguire questa richiesta:

```
http://<serverAddress>:<serverPort>/acc?key=<ACC-Client-KEY>&pin=<chang>&set=toggle
```

La risposta sarà:

```
{"status":"OK","message":"<valoreSettato>"}
```

ACC-Server - il codice

Nel caso di domotics l'ACC-Server, è stato implementato in java, per poterlo integrare direttamente con il modulo del web e per riutilizzare in questi due ambienti gli stessi modelli dei dati.

Quindi è stata creata una pagina del web server, con le funzionalità dell'ACC-Server ([src/acc/AccServlet.java](#)), la quale sarà resa disponibile dal webserver all'indirizzo: <http://<serverAddress>:<serverPort>/acc> .

ACC-Client (Arduino Connection Controller - Client)

Anche L'ACC-Client, è composto di un server HTTP ed un client, il server rimane in ascolto sulla porta **8080**, per la ricezione delle richieste dell'ACC-Server, mentre il client esegue le richieste all'ACC-Server, quando cambia lo stato di un bottone (o di un pin digitale di input).

ACC-Client-ID

Codice identificativo di un ACC-Client, formato da 12 numeri esadecimali:

```
1234567890ABCD
```

ACC-Client-KEY

È un codice di comunicazione fra il ACC-Client e ACC-Server, viene utilizzato per riconoscere che le informazioni sono autentiche. Questa viene generata dal ACC-Server ed inviata al client al momento della configuazione. Anch'essa è formata da 12 numeri esadecimali:

```
1234567890ABCD
```

ACC-Client - alive

Per controllare che l'arduino sia attivo e funzioni correttamente, per il quale eseguire la seguente richiesta:

```
http://<ACC-ClientIP>:<ACC-ClientPort>/alive
```

La relativa risposta sarà:

```
{"status":"OK"}
```

ACC-Client - set

Per settare il dei valori sui pin di output del arduino, eseguire la seguente richiesta all'ACC-Client:

```
http://<ACC-ClientIP>:<ACC-ClientPort>/acc?key=<ACC-Client-KEY>&pin=<pinToSet>&set=<valueToSet>
```

La relativa risposta sarà:

```
{"status":"OK","message":"<setValue>"}
```

ACC-Client - get

Per richiedere all'arduino dei valori di dei pin di input o output, bisogna eseguire la seguente richiesta all'ACC-Client:

```
http://<ACC-ClientIP>:<ACC-ClientPort>/acc?key=<ACC-Client-KEY>&pin=<requiredPin>
```

La relativa risposta sarà:

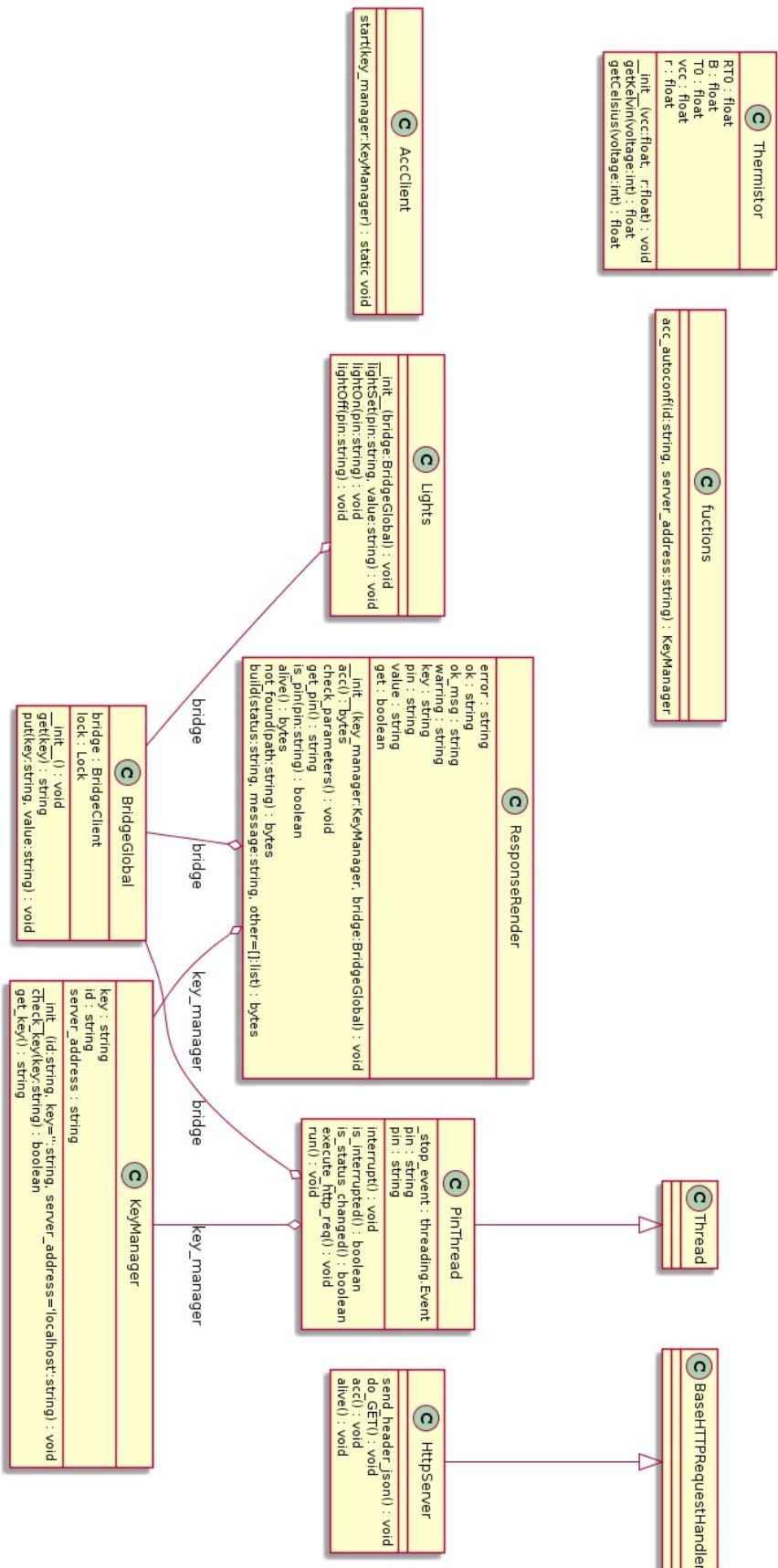
```
{"status":"OK","message":"<value>"}
```

ACC-Client - il codice

Nel caso dell'ACC-Client il codice è stato scritto in python, siccome l'Arduino YÚN (Lilino) permette di scrivere il codice in python ed eseguirlo a stretto contatto con il codice sul lato Arduino. Per una spiegazione più accurata del funzionamento dell'Arduino, usare la guida [doc/Documentazione/Guide/arduinoYun.md](#) .

Principalmente il codice è strutturato in due parti, il codice eseguito sull'Arduino, che semplicemente aggiorna lo stato dei pin di output (prendendo i valori dal Bridge condiviso con la parte Lilino) e aggiorna lo stato del bridge con i pin di input. Codice: [ino/ACC-Client.ino/ACC-Client.ino.ino](#) .

Mentre il codice relativo a Lilino, si occupa di eseguire il web server, esaudire le richieste dell'ACC-Server e di inviare al server, i cambiamenti dei pin di input, come quelli dei bottoni. Il codice: [py/acc-client.py/](#) . Il tutto è strutturato come mostrato nell'immagine sottostante, che rappresenta il diagramma delle classi.



Guida utilizzo Arduino YÙN

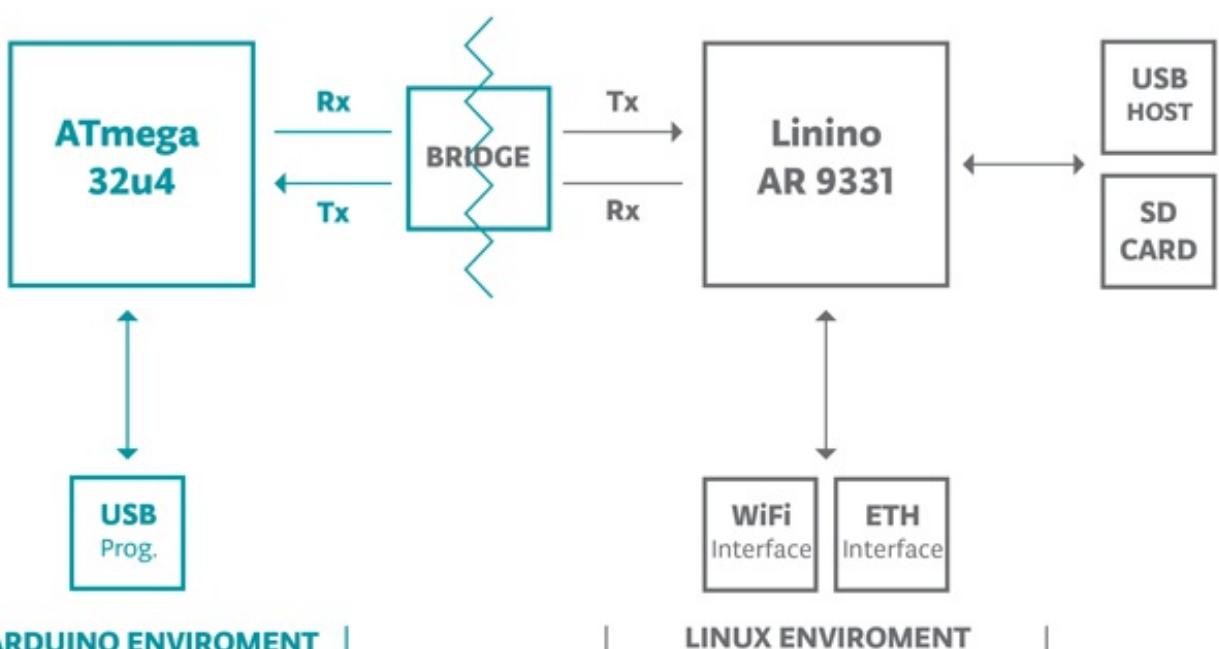


Capitolo	Pagina
Indice	2
Arduino YÙN	3
Inizializzare Arduino	4
Lato arduino	5
Lato lilino	6

Arduino Yun

L'Arduino Yun è una scheda con incoroparti due sistemi indipendenti ma collegati fra loro. Uno è l'Arduino, con il chip ATmega 32u4 collegata alla porta MicroUSB ed hai pin, mentre il secondo chip è un Lilino AR 9331, il quale esegue una distro di linux molto leggera per sistemi embedded, OpenWRT. Il chip Lilino è collegato alla porta USB Type A, allo slot per la scheda MicroSD, alla porta Ethernet ed al modulo Wi-Fi.

I due moduli sono collegati fra solo tramite un bridge. Il quale permette di strasferire dei dati fra i due chip.



Inizializzare Arduino

Per iniziare a configurare l'Arduino Yun bisogna accedere alla shell, per configurare la password di root e vedere l'indirizzo IP. Per eseguire queste operazioni bisogna caricare sull'Arduino un programma che emuli la porta seriale tramite la porta seriale dell'Arduino, quindi caricare il seguente programma tramite l'IDE di Arduino.

```
long linuxBaud = 250000;

void setup() {
    SERIAL_PORT_USBVIRTUAL.begin(115200); // open serial connection via USB-Serial
    SERIAL_PORT_HARDWARE.begin(linuxBaud); // open serial connection to Linux
}

boolean commandMode = false;

void loop() {
    // copy from USB-CDC to UART
    int c = SERIAL_PORT_USBVIRTUAL.read(); // read from USB-CDC
    if (c != -1) { // got anything?
        if (commandMode == false) { // if we aren't in command mode...
            if (c == '~') { // Tilde '~' key pressed?
                commandMode = true; // enter in command mode
            } else {
                SERIAL_PORT_HARDWARE.write(c); // otherwise write char to UART
            }
        } else { // if we are in command mode...
            if (c == '0') { // '0' key pressed?
                SERIAL_PORT_HARDWARE.begin(57600); // set speed to 57600
                SERIAL_PORT_USBVIRTUAL.println("Speed set to 57600");
            } else if (c == '1') { // '1' key pressed?
                SERIAL_PORT_HARDWARE.begin(115200); // set speed to 115200
                SERIAL_PORT_USBVIRTUAL.println("Speed set to 115200");
            } else if (c == '2') { // '2' key pressed?
                SERIAL_PORT_HARDWARE.begin(250000); // set speed to 250000
                SERIAL_PORT_USBVIRTUAL.println("Speed set to 250000");
            } else if (c == '3') { // '3' key pressed?
                SERIAL_PORT_HARDWARE.begin(500000); // set speed to 500000
                SERIAL_PORT_USBVIRTUAL.println("Speed set to 500000");
            } else if (c == '~') { // '~' key pressed?
                SERIAL_PORT_HARDWARE.write((uint8_t *)"\xff\0\0\x05XXXX\x7f\xf9", 11); // send "bridge shutdown" command
                SERIAL_PORT_USBVIRTUAL.println("Sending bridge's shutdown command");
            } else { // any other key pressed?
                SERIAL_PORT_HARDWARE.write('~'); // write '~' to UART
                SERIAL_PORT_HARDWARE.write(c); // write char to UART
            }
            commandMode = false; // in all cases exit from command mode
        }
    }
}

// copy from UART to USB-CDC
c = SERIAL_PORT_HARDWARE.read(); // read from UART
if (c != -1) { // got anything?
    SERIAL_PORT_USBVIRTUAL.write(c); // write to USB-CDC
}
```

Quando il programma è caricato avviare il monitor seriale dell'IDE, con il bottone in alto a destra. Inviare un comando qualunque per controllare se la seriale funziona correttamente. Dopo di che inviare il comando `passwd`, dopo di che inserire due volte la stessa password, che sarà la password dell'utente root sul chip Lilino dell'Arduino Yun. Dopo di che eseguire il comando `ip addr show`, per potersi collegare in remoto con l'Arduino Yun,

Dopo di che collegarsi via SSH all'arduino utilizzando l'indirizzo IP prima letto, lo username `root` e la password prima inserita.

Questo per utilizzare Lilino, mentre per utilizzare la scheda Arduino, caricare tramite l'IDE ufficiale di Arduino i programmi che si desidera.

Dal lato Lilino invece bisogna utilizzare python. Per comunicare fra le due parti, Arduino ha sviluppato un sistema di bridge, che permette trasferire dei valori tramite delle variabili accessibili da entrambe le parti.

Lato arduino

Per capire come funziona, creare un esempio di blick sul pin 13, comandato tramite il bridge.
Utilizzare il seguente codice:

```
// importare libreria bridge.
#include <Bridge.h>
// importare libreria stdio per i metodi delle stringhe e caratteri
#include <stdio.h>

/** 
 * Arduino blink on pin 13 sketch.
 *
 * @author giuliobosco (giuliobva@gmail.com)
 * @version 1.0 (2019-03-20 - 2019-03-20)
 */

// crea variabile in cui salvare i valori del bridge
char D13value[2];

void setup() {
    // setta la dimensione della variabile in cui salvare i valori del bridge
    memset(D13value, 0, 2);

    // setta il pin 13 come output
    pinMode(13, OUTPUT);

    // avvia il bridge
    Bridge.begin();
}

void loop() {
    // leggere il valore di D13 sul bridge e salvarlo nella variabile prima istanziata
    Bridge.get("D13", D13value, 2);
    // trasformare da int a stringa
    int D13int = atoi(D13value);
    // scrivere il valore della variabile sul pin 13
    digitalWrite(13, D13int);

    // ferma il programma per 10 ms per non sovraccaricare troppo il micro controllore
    delay(10);
}
```

Lato Lilino

Mentre dal lato di Lilino, bisogna creare un programma in python, che setti in maniera alternata il valore di D13 ad 1 e 0 ogni secondo, tramite il bridge.

In questo caso esegue questa operazione 100 volte, dopo di che stampa *I hope you enjoyed the light show*.

```
# importa sys per aggiungere la libreria bridge
import sys
# aggiungo libreria bridge
sys.path.insert(0, '/usr/lib/python2.7/bridge')

# importo la funzione sleep
from time import sleep
# importo libreria bridge
from bridgeclient import BridgeClient as bridgeclient

# inizializzo bridge
value = bridgeclient()

# per 100 volte
for idx in range(0, 100):
    # setta D13 a 1 (pin acceso)
    value.put('D13','1')
    # aspetta 1 secondo
    sleep(1)
    # setta D13 a 0 (pin spento)
    value.put('D13','0')
    # aspetta 1 secondo
    sleep(1)

# stampa messaggio
print("I hope you enjoyed the light show\n")
```

Guida Modulo back-end

Capitolo	Pagina
Indice	2
Introduzione	3
Design	3
LDAP	3
JDBC	4
Modelli database	4
ACC-Server	6
Dati per front-end (JSON)	7
Implementazione	8
LDAP	8
JDBC	8
Modelli database	8
ACC-Server	9
Dati per front-end (JSON)	9

back-end

L'unione di tutti i moduli del progetto avviene nel back-end, il quale deve collegare il front-end, cioè la web-app, il database di domotics, e i micro controller (nel nostro caso Arduino), tramite l'ACC.

Il back-end è scritto in java, e verrà servito da tomcat, come web server.

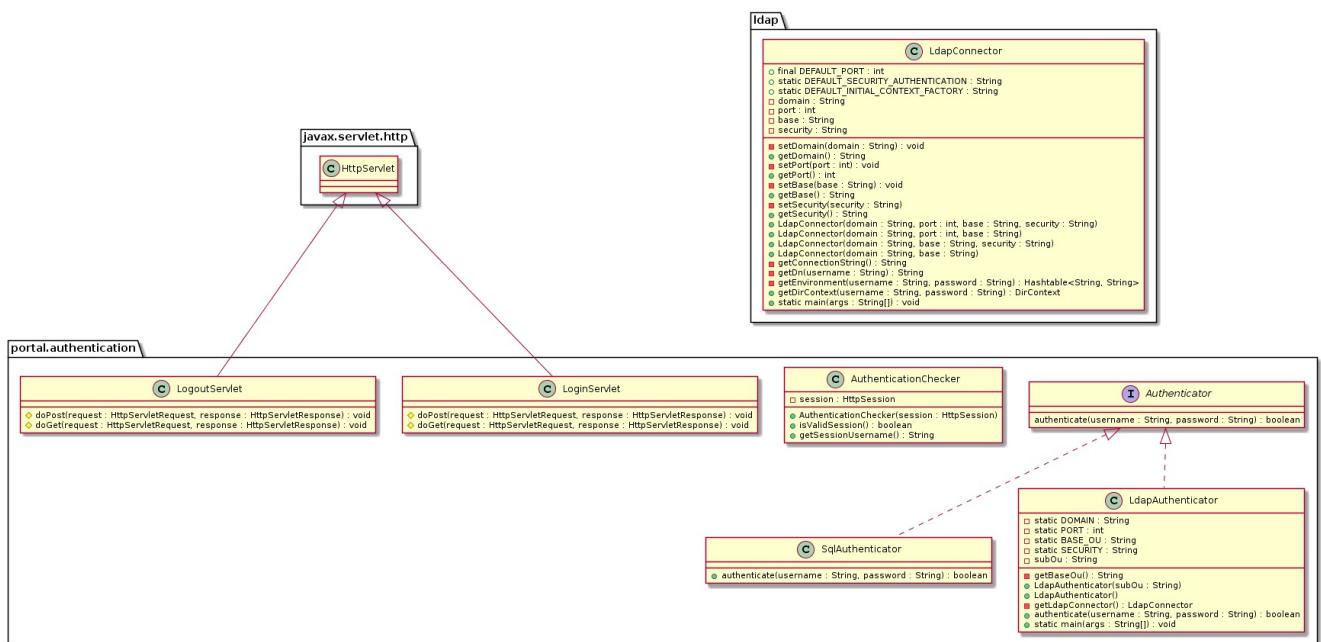
Design

Il back-end, è stato progettato in più fasi, modulo per modulo.

LDAP

Il primo modulo ad essere stato progettato è stato quello di `ldap` e dell'autenticazione, che si compone delle classi:

- `LdapConnector`, gestisce le connessioni con il server ldap.
- `Authenticator`, è un interfaccia creata per poter utilizzare diversi tipi di autenticazione con facilità
- `LdapAuthenticator`, classe per autenticarsi con ldap, implementa `Authenticator`.
- `AuthenticationChecker`, una classe utilizzata per convalidare le sessioni HTTP.
- `LoginServlet`, servlet per eseguire il login della sessione HTTP.
- `LogoutServlet`, servlet per invalidare la sessione HTTP.

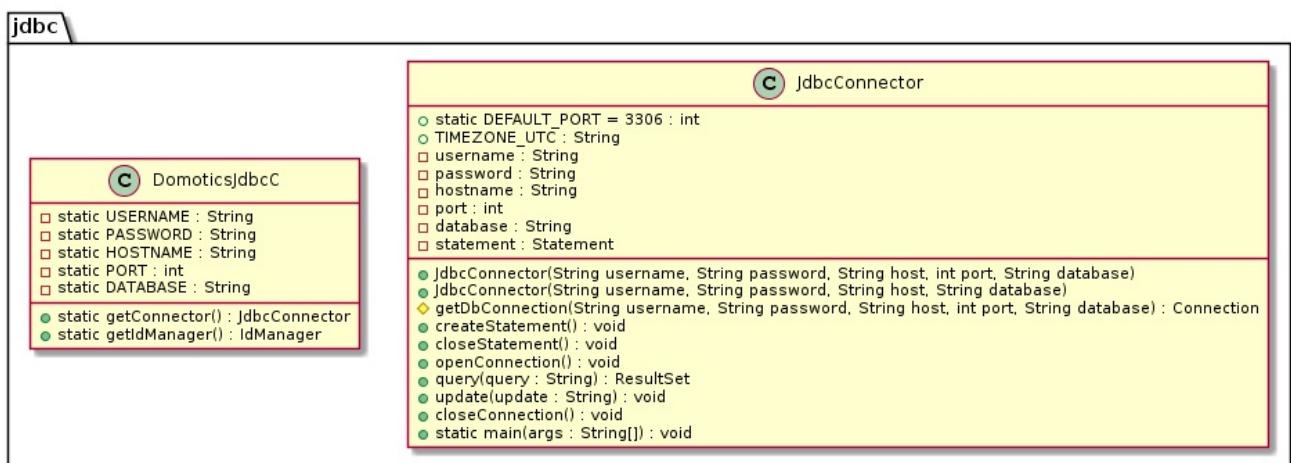


JDBC

Dopo di che è stato progettato il modulo relativo al database, quindi l'utilizzo di JDBC, il driver per connettersi ai database MySQL con Java.

Il quale è composto delle seguenti classi:

- `JdbcConnector`, gestisce le connessioni al server MySQL, per poter funzionare necessita che vi sia presente la libreria `mysql-connector-java-8.0.13.jar`.
- `DomoticsJdbcC`, è una classe che istanzia `JdbcConnector` nella maniera corretta per questo progetto.

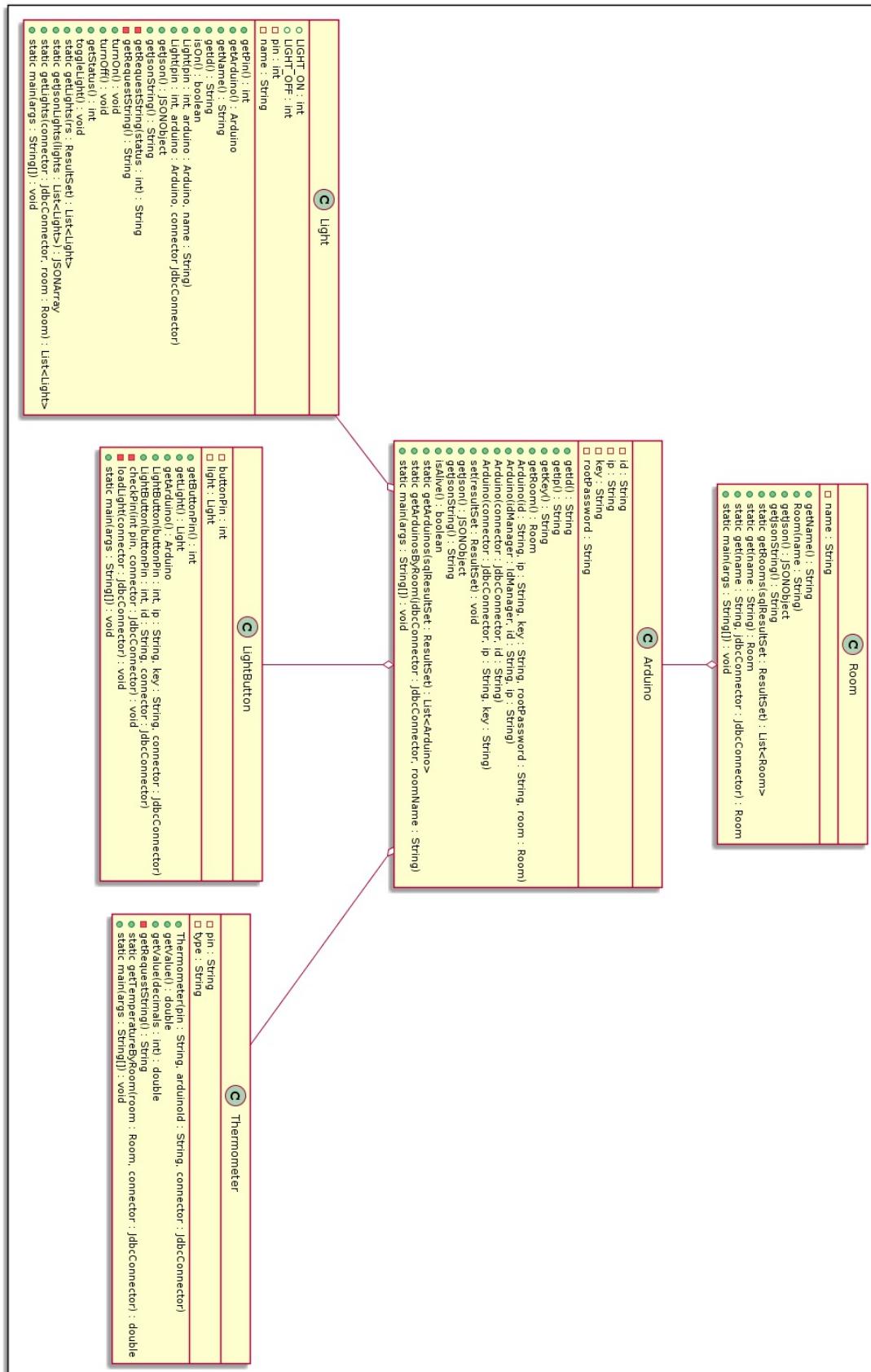


Modelli database

Dopo aver progettato il modulo della connessione al database, sono state progettate le classi che rappresentano le istanze dei database e che ci interaggiscono direttamente.

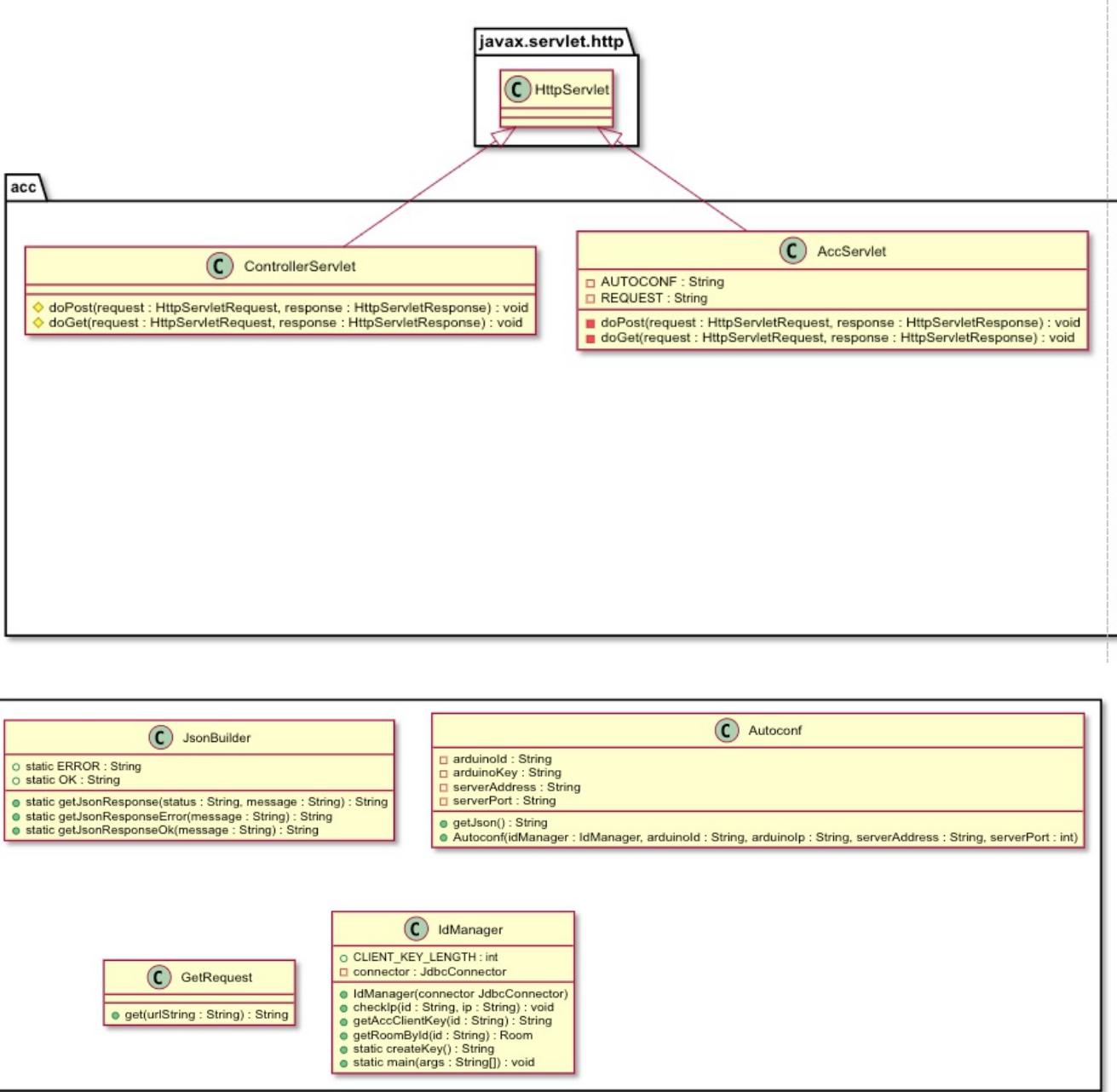
Per ogni tabella del database, è stata creata una classe. La quale servirà per aiutare l'interazione con il database e gli altri moduli. Le classi sono:

models



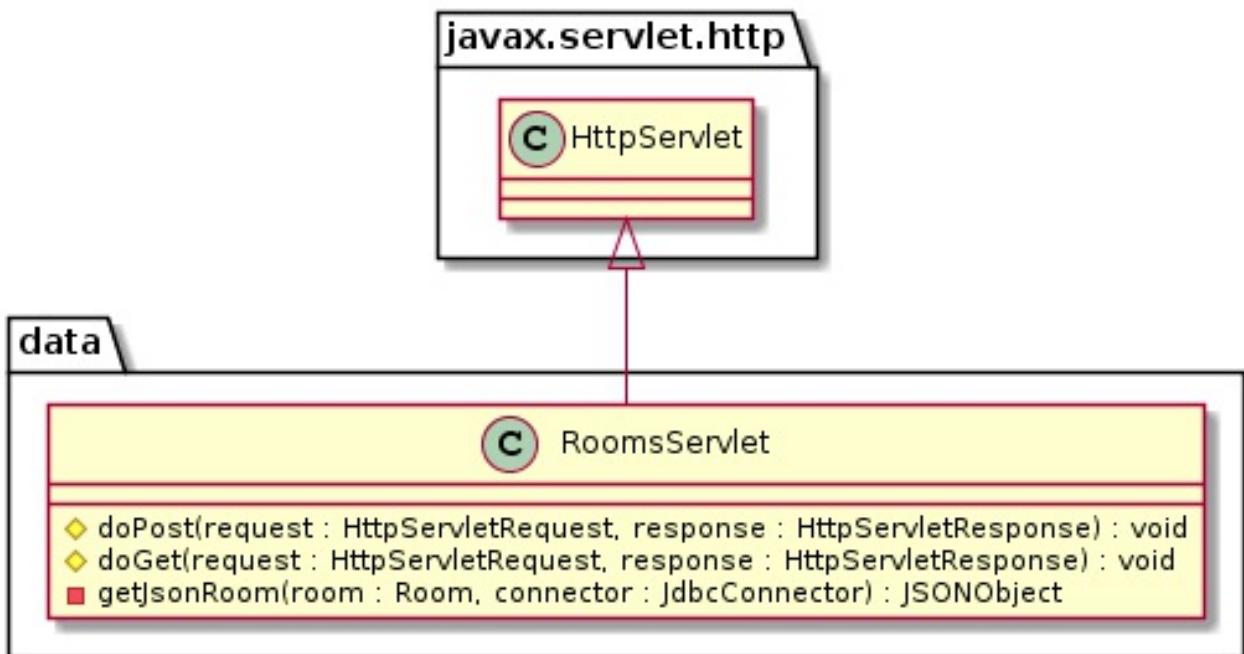
ACC-Server

Dopo aver progettato le classi modello per il database, è stato progettato il modulo dell'ACC lato server, sono state progettate le classi di cui avrebbe necessitato, per poter funzionare correttamente con gli elementi già esistenti. Queste classi sono state studiate, in maniera da mantenere i vari elementi più separati possibile, così da poter sostituire o modificare i vari elementi più facilmente possibile. Infatti le richieste vengono interpretate dalla servlet (`AccServlet`), i controlli sui micro controllori vengono eseguiti dalla classe preposta (`IdManager`), le richieste HTTP ai microcontrollori vengono eseguite tramite la classe `GetRequest` e le configurazioni per i microcontrollori vengono generate dal `Autoconf`.



Dati per front-end (JSON)

Da ultimo è stato progettato il modulo relativo ai dati, i dati da inviare al front-end. Che è composto dalla classe `RoomsServlet`, in futuro potrebbe venir anche ingrandito. Questa classe, semplicemente richiede al database (tramite le classi modello), tutte le `Room`, con le relative luci. E trasforma tutto in un file JSON, che viene inviato come risposta.



Implementazione

Per quanto riguarda l'Implementazione, è stato un processo, che ha preso tempo, ed è stato fatto durante la progettazione degli altri moduli, quando un modulo veniva progettato, veniva subito implementato e nel frattempo si progettava l'altro modulo. Questo perchè l'obbiettivo era mantenerli indipendenti, utilizzando questo approccio eravamo anche più liberi di scegliere l'attività da fare potendo spaziare tra progettazione ed implementazione.

LDAP

Per il modulo di LDAP la parte complicata è quella di costruire la stringa di connessione tramite la `Hashtable`.

```
/**  
 * Get the hashtable environment of the connection.  
 *  
 * @param username Username of the connection.  
 * @param password Password of the connection.  
 * @return Hashtable Environment of the connection.  
 */  
private Hashtable<String, String> getEnvironment(String username, String password) {  
    Hashtable<String, String> environment = new Hashtable<String, String>();  
  
    environment.put(Context.INITIAL_CONTEXT_FACTORY, DEFAULT_INITIAL_CONTEXT_FACTORY);  
    environment.put(Context.PROVIDER_URL, getConnectionString());  
    environment.put(Context.SECURITY_AUTHENTICATION, getSecurity());  
    environment.put(Context.SECURITY_PRINCIPAL, getDn(username));  
    environment.put(Context.SECURITY_CREDENTIALS, password);  
  
    return environment;  
}
```

La quale viene creata in un metodo dedicato.

JDBC

Nell'implementazione del modulo di JDBC, i punti complicati, sono:

- inserire la libreria in maniera che il sistema ne possa fare uso
- creare la giusta stringa di connessione, per la quale abbiamo utilizzato un trucchetto per evitare problemi con gli orari non sincronizzati fra web-server, e server MySQL.

Il driver di JDBC (libreria) va inserito come libreria esterna del progetto (quando si lavora su IntelliJ IDEA), mentre per il server nelle librerie di tomcat.

Mentre per creare la stringa di connessione, bisogna inserire il driver (`jdbc`), il protocollo (`mysql`), l'host, la porta ed il nome del database. Infine aggiungere un `?` con la stringa dell'orario UTC.

```
final String TIMEZONE_UTC =  
"useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC";  
String connectionString = "jdbc:mysql://" + host + ":" + port + "/" + database + "?";  
connectionString += TIMEZONE_UTC;
```

Modelli database

Per i modelli del database, sono semplicemente state riprese le tabelle del database, e scritti i relativi oggetti. Dopo di che, sono stati aggiunti alcuni metodi per eseguire più facilmente determinate operazioni.

ACC-Server

Per l'ACC-Server sono state implementate le varie classi, le sono principalmente `IdManager`, che si occupa di controllare che siano autentici gli Id, gli Ip e le key dei microcontrollori, e la classe `AccServlet`, che si occupa soddisfare le richieste provenienti dall'ACC-Client.

Dati per il front-end (JSON)

Per la generazione del file JSON da ritornare al front-end come dati, la maggior parte di questo processo avviene nel metodo `getJsonRoom()` della classe `RoomsServlet`, il quale richiede lo stato di tutte le luci, la temperatura della stanza, il nome della stanza e crea un oggetto JSON, il quale verrà aggiunto ad un'array di oggetti JSON, che infine verrà ritornato come stringa al front-end.

```
private JSONObject getJsonRoom(Room room, JdbcConnector jdbc) throws SQLException,  
ClassNotFoundException, IOException {  
    JSONObject roomJson = new JSONObject();  
  
    JSONArray lights = Light.getJsonLights(Light.getLights(jdbc, new Room(room.getName())));  
  
    roomJson.put("lights", lights);  
    roomJson.put("name", room.getName());  
    roomJson.put("temp", Thermometer.getTemperatureByRoom(room, jdbc) + "");  
  
    return roomJson;  
}
```

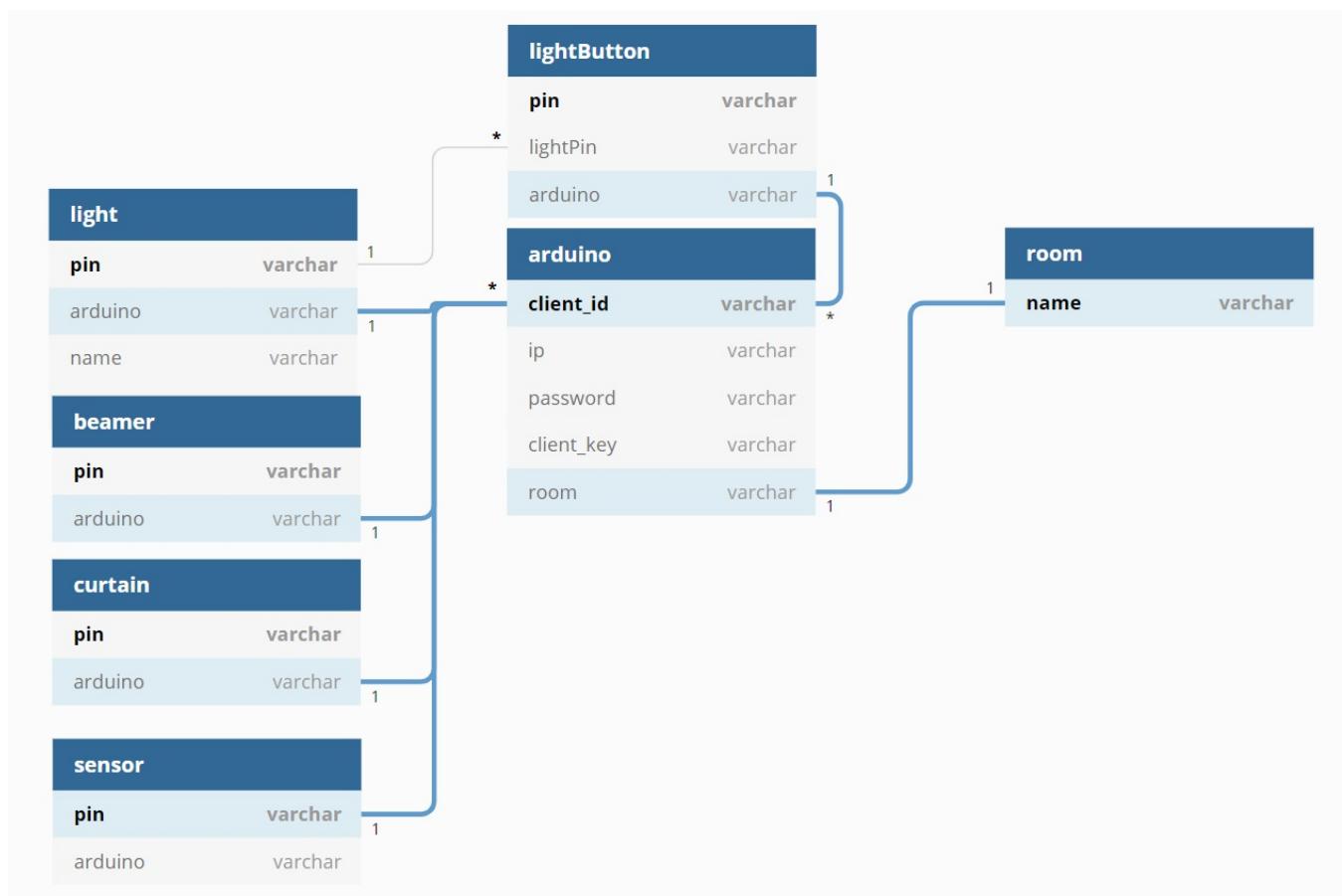
Guida Modulo domotics database



Capitolo	Pagina
Indice	2
Funzionamento	3
Implementazione	4

domotics database

Funzionamento



Questo è il design del database, che è formato da sette tabelle. La tabella room rappresenta le aule di cui viene salvato il loro identificatore, per ogni aula viene associato un arduino, di esso si vuole sapere l'indirizzo ip, la password e la chiave per il client. Ogni arduino gestisce i moduli che devono essere controllati all'interno dell'aula che poi potranno essere aggiunti e implementati ma che al momento sono solamente gli interruttori delle luci, di cui vengono memorizzati i suoi pin e le luci a cui sono collegati. Delle luci si memorizzano i loro pin e stessa cosa per il beamer, le tende e i sensori. In questo modo grazie alla chiave che ogni arduino ha, il server riesce a riconoscerli e identificarli per poi ricavare lo stato di tutti i suoi moduli, quindi luci, tende, beamer e sensori andando a ricercare le varie informazione nel database. Grazie al database si possono tenere tutte le informazioni in modo ordinato, pulito e reperibile per quando si ha bisogno.

Implementazione

Per mettere sul il seguente data base bisogna seguire i seguenti passaggi.

Innanzitutto bisogna creare la tabella principale che rappresenta le aule che vengono identificate dal loro nome.

```
CREATE TABLE domotics.room (
    name VARCHAR(255) PRIMARY KEY
);
```

Poi si crea una tabella che rappresenta l'arduino che sarà posizionato all'interno dell'aula. L'arduino viene identificato da un id e si vuole sapere il suo indirizzo ip, la password della root, la chiave del client che servirà al server per riconoscerlo e l'aula in cui sono.

```
CREATE TABLE domotics.arduino (
    client_id      VARCHAR(255) PRIMARY KEY,
    ip              VARCHAR(255),
    root_password  VARCHAR(255),
    client_key     VARCHAR(255),
    room            VARCHAR(255),

    FOREIGN KEY (room)
        REFERENCES domotics.room (name)
);
```

Poi si andranno a creare le tabelle che rappresentano i moduli all'interno delle aule che vengono controllati dall'arduino.

I valori da memorizzare sono molto simili per i moduli tranne qualche eccezione.

Delle luci si vogliono sapere i loro pin, l'arduino alla quale appartengono e quindi da cui vengono controllate e il loro stato quindi se sono accese o spente.

```
CREATE TABLE domotics.light (
    pin      VARCHAR(255),
    arduino VARCHAR(255),
    name    VARCHAR(255),
    status   INT(1),
    PRIMARY KEY (pin, arduino),
    FOREIGN KEY (arduino)
        REFERENCES domotics.arduino (client_id)
);

CREATE TABLE domotics.beamer (
    pin      VARCHAR(255),
    arduino VARCHAR(255),
    name    VARCHAR(255),
    PRIMARY KEY (pin, arduino),
    FOREIGN KEY (arduino)
        REFERENCES domotics.arduino (client_id)
);
```

Il tipo di tenda permette di riconoscere se è la tenda di sinistra o di destra.

```
CREATE TABLE domotics.curtain (
    pin      VARCHAR(255),
    arduino VARCHAR(255),
    name    VARCHAR(255),
    PRIMARY KEY (pin, arduino),
    FOREIGN KEY (arduino)
        REFERENCES domotics.arduino (client_id)
);
```

Dei sensori si vogliono sapere anche i pin del arduino a cui appartengono grazie al quale poi sarà facile ricavare tutte le informazioni necessarie. SensorType serve a evitare che ci siano sensori uguali con nomi diversi.

```
CREATE TABLE domotics.sensor (
    pin      VARCHAR(255),
    arduino  VARCHAR(255),
    type     VARCHAR(255),
    name     VARCHAR(255),
    PRIMARY KEY (pin, arduino),
    FOREIGN KEY (arduino)
        REFERENCES domotics.arduino (client_id),
    FOREIGN KEY (type)
        REFERENCES domotics.sensorType (name)
);
```

```
CREATE TABLE domotics.sensorType (
    name VARCHAR(255) PRIMARY KEY
);
```

Le luci possono essere gestite anche tramite dei pulsanti di cui si salva il pin, il pin della luce che gestisce, l'arduino a cui è collegato e il nome.

```
CREATE TABLE domotics.lightButton (
    pin      VARCHAR(255),
    lightPin VARCHAR(255),
    arduino  VARCHAR(255),
    name     VARCHAR(255),
    PRIMARY KEY (pin, arduino),
    FOREIGN KEY (arduino)
        REFERENCES domotics.arduino (client_id),
    FOREIGN KEY (lightPin)
        REFERENCES domotics.light (pin)
);
```

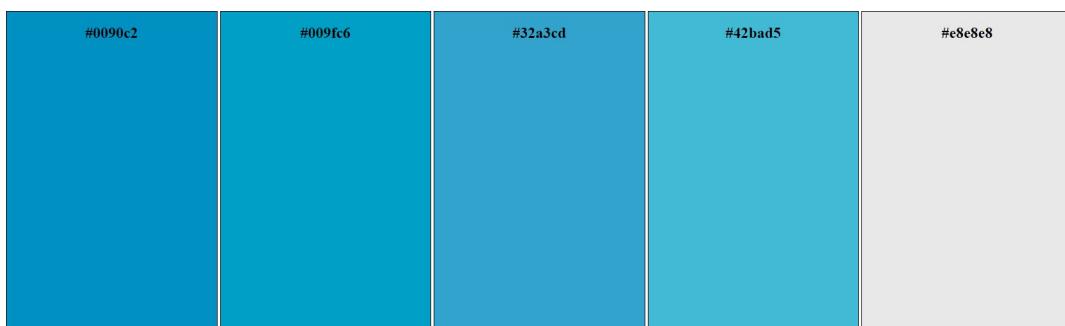
Guida Modulo front-end

Capitolo	Pagina
Indice	2
Design	3
Implementazione	5
WEB-APP	5
moduli presenti	5
AngularJS	5
Creare un controller	5
Creare un service	6
views	5
stile	7

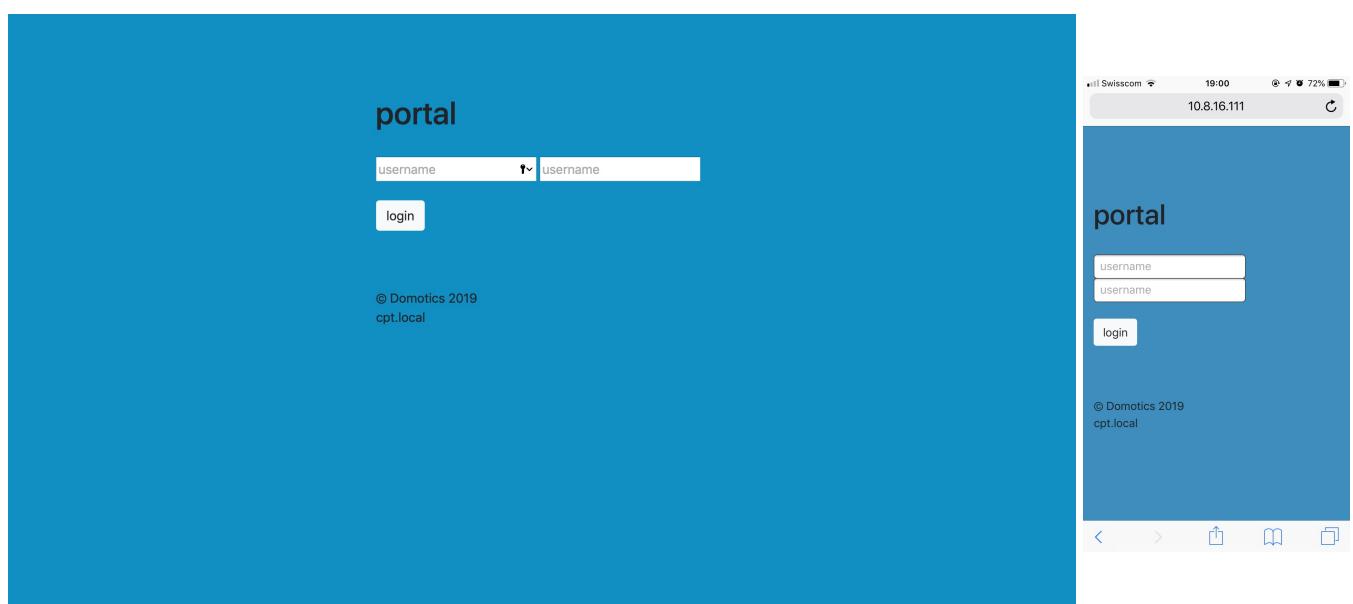
front-end

Design

Per il frontend abbiamo scelto una paletta di colori di base per tutti i siti.



La pagina di login contiene un semplice form composto da un campo testo, un campo password e un pulsante di submit.



La pagina principale è composta da due parti, la dashboard e il corpo principale contenente le aule. La dashboard contiene le pagine del sito (al momento solo una) e la gestione del account. Invece il corpo della pagina permette di gestire le aule, vedere la temperatura e spegnere/accendere le luci.

The screenshot displays the SAMT domotics front-end interface. On the left, a vertical sidebar titled "portal" contains a "rooms" section. The main area is titled "rooms" and shows four rooms: A100, A101, A102, and A103. Each room has a temperature reading (23.5°C, 22°C, 22°C, 22°C) and two light control buttons: "window light" and "door light". The "window light" button for A100 is off, while the "door light" is on. For A101, the "window light" is on, and the "door light" is off. For A102 and A103, both "window light" and "door light" are on. Below the rooms section, there is a "rooms" button followed by three icons: a left arrow, an upward arrow, and a right arrow. At the bottom of the sidebar, there are "gear" and "refresh" icons.

The main content area shows the same "rooms" dashboard. Above it, a header bar indicates the IP address 10.8.16.111 and the time 19:00. The header bar also includes Swisscom signal strength, battery level at 72%, and a refresh icon.

Below the header, the "rooms" section is repeated, showing the same room configurations. At the bottom of this section, there are five icons: a left arrow, a right arrow, an upward arrow, a book icon, and a square icon.

To the right of the "rooms" section, a large blue area titled "portal" contains the word "rooms". At the bottom of this area, there are "gear" and "refresh" icons, along with the same set of five bottom icons.

Implementazione

WEB-APP

Il front-end di domotics, è stato pensato come una web, app costruita su moduli, quindi per facilitare questo metodo di sviluppo è stato deciso di basarlo su AngularJS, un framework, che permette di aggiungere e togliere moduli indipendenti senza andare ad intaccare gli altri. Questo è stato pensato per poter usare questo sistema come base per un portale al quale si possono aggiungere altri moduli.

moduli presenti

Al momento sono presenti solamente due moduli, quelli di base per il l'interazione con il modulo domotics:

- Login: modulo per il login basato su LDAP.
- Rooms: Controllo delle luci di domotics.

AngularJS

La web-app basata su AngularJS, è formata dalla pagina index.html, la quale carica tutte le librerie e i framework utilizzati dal progetto, per esempio AngularJS, jQuery e bootstrap. Inoltre carica l'applicazione Angular, configura le routes delle pagine ed infine carica i controller ed i services.

Per ogni pagina bisogna creare un controller, il quale richiede i services di cui necessita ed inserisca i valori nella view.

Per inizializzare la web app:

```
var app = angular.module('ViewsAPP', ['ngRoute', 'ngSanitize']);
```

Questa stringa di codice inizializza l'applicazione angular, con il nome `ViewsAPP` e le configura le librerie `ngRoute`, che serve per caricare le giuste routes, richiede al server la giusta view, dato l'url. Mentre `ngSanitize`, serve per stampare del codice html scaricato tramite un service, per esempio all'interno di un file JSON.

creare un controller

I controller servono, per trasferire i dati dal servizio alla view (e nel caso in cui necessario eseguire delle operazioni su di essi). Un controller si crea come segue:

```
app.controller('Controller', ['$scope', '$sce', 'Service', function ($scope, $sce, service) {
    service.getFromService().then(function (data) {
        $scope.data = data;
    });
}]);
```

Creando un controller, bisogna inserire il suo nome, poi un array contenente gli elementi che necessita il controller, ed infine la funzione del controller, con i parametri richiesti precedentemente.

Dopo di che eseguire le operazioni che si necessitano nel controller, la variabile `$scope`, viene utilizzata per passare i valori fra i controller e le view.

creare un service

I service servono per eseguire le richieste al server, queste possono essere per esempio richieste HTTP.

```
app.factory('Service', ['$http', function($http) {
    var service = [];
    var urlBase = "/data/rooms";

    service.getFromService = function () {
        return $http({
            method: 'GET',
            url: url
        }).then(function (response){
            return response.data;
        },function (error){
            return error;
        });
    };

    return service;
}]);
```

Un service richiede il nome, ed un array, con gli oggetti di angular di cui necessita, quindi per esempio `$http`, che sarebbe la libreria, per eseguire le richieste HTTP. In ogni service solitamente si mette una sola richiesta, che può essere eseguita in modalità diverse, quindi per ogni modalità si crea una funzione per eseguire la richiesta.

Tutte le richieste vanno inserite in un oggetto, il quale verrà poi ritornato.

Views

Per gestire le pagine vengono usate delle views che vengono caricate da `app.js` nel body della pagina `index.html`, queste vengono gestite dal modulo angular `ngRoute`.

```
app.config(function ($routeProvider) {
    // index
    $routeProvider.when('/', {
        templateUrl: 'views/index.html'
    });

    // login
    $routeProvider.when('/login', {
        templateUrl: 'views/login.html'
    });

    // rooms
    $routeProvider.when('/rooms', {
        templateUrl: 'views/rooms.html'
    });

    // settings
    $routeProvider.when('/settings', {
        templateUrl: 'views/settings.html'
    });

    // else
    $routeProvider.otherwise({
        redirectTo: '/'
    });
}).run(function ($rootScope, $route) {
    $rootScope.$route = $route;
});
```

Per configuare `ngRoute`, bisogna congiurarlo con una funziona da inserire nel metodo `config()` dell'app. Al quale viene passato l'oggetto `$routeProvider`, il quale ha un metodo `when()`, questo metodo permette di settare ad uno specifico url (dopo il simbolo `#`, per esempio `localhost#/test`), una view da caricare. Ed utilizzare anche un metodo `otherwise()`, che viene utilizzato nel caso in cui viene inserito un url non specificato prima.

Infine avviare la web app con le routes configurate precedentemente.

Stile

Per lo stile delle pagine usiamo bootstrap (<https://getbootstrap.com>) e fontawesome (<https://fontawesome.com>). Abbiamo usato Fontawesome per aggiungere delle icone al sito, abbiamo usato Fontawesome al posto delle immagini perché le icone che contiene sono dei caratteri che quindi sono più leggeri e più facili da usare delle immagini.

Ecco un esempio di view implementata usando bootstrap e fontawesome.

```
<div class="navbar">
    <i class="fa fa-bars fa-2x"></i>
</div>
<div class="sidebar">
    <div class="col-md-12">
        <h2>portal<i class="pull-right fa fa-times"></i></h2>
        <ul class="a-white">
            <li><a href="#!/rooms">rooms</a></li>
        </ul>
    </div>
    <div class="bottom col-md-12 a-white">
        <a href="#!/settings"><i class="fa fa-cog fa-2x"></i></a>
        <a href="/Logout"><i class="fa fa-sign-out pull-right fa-2x"></i></a>
    </div>
</div>
<script src="assets/js/scripts/sidebar.js"></script>
```

Fontawesome usa il tag *per inserire le icone*. Per definire l'icona bisogna inserire nel attributo *class*: *fa* che definisce l'uso di fontawesome, *fa-[nome icona]* che definisce quale icona si vuole usare, e se si vuole la grandezza dell'icona

```
<i class="fa fa-cog fa-2x">
```

Guida Modulo LDAP



Capitolo	Pagina
Indice	2
Progettazione	3
Implementazione	4

Guida Lightweight Directory Access Protocol

Progettazione

Per connetterci al database della scuola abbiamo deciso di utilizzare LDAP Connector perché era la soluzione più facile e sicura da implementare dato che passi le credenziali per entrare nel database alla scuola, gli dici cosa vuoi sapere e fa tutto lui così nessuno accede direttamente sul database per evitare problemi.

TestLdapCtp

```
+static main(args[]): String: void
```

LdapConnector

```
+static DEFAULT_PORT: int
+static DEFAULT_SECURITY_AUTHENTICATION: String
+static DEFAULT_INITIAL_CONTEXT_FACTORY: String
-domain: String
-port: int
-base: String
-security: String

+setDomain(domain: String): void
+getDomain(): String
+setPort(port: int): void
+getPort(): int
+setBase(base: String): void
+getBase(): String
+setSecurity(security: String): void
+getSecurity(): String
+LdapConnector(domain: String, port: int, base: String, security: String)
+LdapConnector(domain: String, port: int, base: String)
+LdapConnector(domain: String, base: String)
+getConnectionString(): String
+getDn(username: String): String
+getEnvironment(username: String, password: String): DirContext
+getDirContext(username: String, password: String): DirContext
```

La classe LdapConnector viene utilizzata nel login, infatti quando l'utente si loggerà le credenziali che mette vengono prese e vengono confrontate con le credenziali del database della scuola dove vede se l'utente è un docente e quindi ha i permessi per accederci o è un allievo e quindi non ha i permessi. LDAP permette di mantenere anche una certa sicurezza essendo che la comunicazione è criptata. Nella classe ci saranno i seguenti attributi statici: la porta di default del server, la chiave di autenticazione, e una variabile che rappresenta il contesto iniziale del LDAP. Poi nelle variabili domain viene salvato il dominio del server LDAP, nella variabile port la porta del server se è diversa da quella di default, la variabile base rappresenta il livello del server nelle

unità organizzative dove deve andare a controllare le credenziali e security rappresenta il tipo di sicurezza che viene utilizzata per connettersi. Nella classe vengono implementati poi tutti i vari get e set per settare o ritornare i valori delle variabili, ci sono tre costruttori, uno a cui viene passato il dominio, la porta, l'unità organizzativa e il tipo di sicurezza, uno a cui non viene passato il tipo di sicurezza e nel terzo viene passato solo il dominio e l'unità organizzativa. Il metodo getEnvironment ritorna l'ambiente hashtable della connessione, getConnectionstring ritorna la stringa di connessione, getDN ritorna una stringa con le credenziali e l'unità organizzativa da inviare nel metodo getEnvironment per creare la connessione, e getContext ritorna se l'utente ha i permessi o no.

Implementazione

Nella nostra classe sono presenti cinque costruttori, uno a cui gli viene la stringa rappresentante il dominio, la porta, la stringa rappresentante il percorso dell'unità organizzativa dove andare a confrontare gli account presenti con quello del login e la stringa security che contiene il livello di sicurezza dell'autentificazione. Il secondo costruttore è uguale ma non gli viene passata la stringa di autentificazione e viene usato il valore di default "simple", nel terzo gli vengono passati solo dominio, percorso e porta, nel quarto costruttore solo il dominio, il percorso e il livello di sicurezza dell'autentificazione e nell'ultimo gli viene passato solo il dominio e il percorso.

```
public LdapConnector(String domain, int port, String base, String security) throws IOException {
    this.setDomain(domain);
    this.setBase(base);
    this.setPort(port);
    this.setSecurity(security);
}

public LdapConnector(String domain, int port, String base) throws IOException {
    this(domain, port, base, DEFAULT_SECURITY_AUTHENTICATION);
}

/**
 * Create the LDAP connector.
 *
 * @param domain    LDAP server address.
 * @param base      LDAP base ou.
 * @param security LDAP Security type.
 */
public LdapConnector(String domain, String base, String security) {
    try {
        this.setDomain(domain);
        this.setBase(base);
        this.setPort(DEFAULT_PORT);
        this.setSecurity(security);
    } catch (IOException ignored) {
        // ignored because the default port is in the range (and it's a constant).
    }
}

/**
 * Create the LDAP connector.
 *
 * @param domain LDAP server address.
 * @param base   LDAP base ou.
 */
public LdapConnector(String domain, String base) {
    this(domain, base, DEFAULT_SECURITY_AUTHENTICATION);
}
```

Il metodo getConnectionstring come dice il nome ritorna la stringa di connessione, molto semplicemente prende il percorso di ldap nella rete e ci aggiunge il domionio e la porta.

```
private String getConnectionstring() {
    return "ldap://" + getDomain() + ":" + getPort();
}
```

Questo metodo viene utilizzato per creare e poi ritornare DN, è simile a un percorso assoluto solo che invece di scendere l'albero da sinistra scende da destra.

Ecco un esempio di DN:

CN=john.doe,OU=People,DC=example,DC=com

CN è il nome utente, OU è l'unità organizzativa a cui deve puntare (che possono essere più di una) mentre il primo DC rappresenta le componenti del dominio.

Per creare il DN prendiamo l'username e lo uniamo con il percorso dell'unità organizzativa che ricaviamo con getBase();

```
private String getDn(String username) {  
    return "CN=" + username + "," + getBase();  
}
```

Il metodo sottostante si occupa di creare una Hashtable che contiene tutti i parametri che poi verranno inviati.

I parametri che andranno inviati sono:

- DEFAULT_INITIAL_CONTEXT_FACTORY, Connessione iniziale predefinita del contesto iniziale di fabbrica.
- getConnectionstring(), stringa di connessione.
- getDn(username), percorso a cui deve puntare.
- password, con cui si è tentato di accedere e che deve essere controllata.

```
private Hashtable<String, String> getEnvironment(String username, String password) {  
    Hashtable<String, String> environment = new Hashtable<String, String>();  
  
    environment.put(Context.INITIAL_CONTEXT_FACTORY, DEFAULT_INITIAL_CONTEXT_FACTORY);  
    environment.put(Context.PROVIDER_URL, getConnectionstring());  
    environment.put(Context.SECURITY_AUTHENTICATION, getSecurity());  
    environment.put(Context.SECURITY_PRINCIPAL, getDn(username));  
    environment.put(Context.SECURITY_CREDENTIALS, password);  
  
    return environment;  
}
```

Quest'ultimo metodo invece utilizza l'Hashtable per collegarsi e controllare se l'utente è presente all'interno dell'unità organizzativa a cui gli è stato detto di andare a controllare poiché al metodo gli viene passato sia l'username che la password che poi richiama il metodo per fare l'Hashtable.

Se l'autenticazione è valida continua altrimenti richiama un eccezione.

```
public DirContext getDirContext(String username, String password) throws NamingException {  
    return new InitialDirContext(getEnvironment(username, password));  
}
```

Guida installazione Apache Tomcat 9



Capitolo	Pagina
Indice	2
Introduzione	3
Install Java	3
Install Tomcat	5
Download Tomcat	5
Configure firewall	6
Test tomcat	6
enable service	7
conf tomcat user	8
configure Apache Reverse Proxy	9

Install Tomcat 9.0.13

Introduction

This guide explains all the step-by-step procedures to install a web server in Tomcat on a centOS 7 machine.

OS: CentOS 7

Install Java

To use Tomcat is required java, before you install java update the system. Install the Java JRE (Java Runtime Environment) and the JDK (Java Development Kit).

```
yum update  
yum install java-1.8.0-openjdk.x86_64 java-1.8.0-openjdk-devel.x86_64
```

Check if java is installed, checking the version.

```
java -version  
javac -version
```

Set the last version as the usual Java version. Use the following command and select the last java version.

```
update-alternatives --config java
```

Set the java environment, editing the following file.

```
/etc/environment
```

Insert in the file the variable `JAVA_HOME` as follows:

```
JAVA_HOME="/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.111-1.b15.el7_2.x86_64/jre"  
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.191.b12-1.el7_6.x86_64/jre
```

then edit the profile bash script (`~/.bash_profile`) and insert the following lines.

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.111-1.b15.el7_2.x86_64/jre  
export PATH=$JAVA_HOME/bin:$PATH
```

reload the profile bash script

```
source ~/.bash_profile
```

check if the variable works correctly displaying it

```
echo $JAVA_HOME
```

It should return something like this

```
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.111-1.b15.el7_2.x86_64/jre
```

Install Tomcat

Create the user and group for tomcat:

```
groupadd tomcat  
useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```

The `-s /bin/false` means that this user has no shell.

Download Tomcat

Move to the directory where you want to install tomcat.

```
cd /opt
```

Install wget, it's a small software to download files by http.

```
yum install wget
```

Download tomcat, with the software you just installed.

```
http://www-eu.apache.org/dist/tomcat/tomcat-9/v9.0.13/bin/apache-tomcat-9.0.13.tar.gz
```

Decompress tomcats files

```
tar -xzvf apache-tomcat-9.0.13.tar.gz
```

Move tomcats files to the tomcat user home directory

```
mv apache-tomcat-9.0.13/* tomcat/
```

change owner of all the files under the tomcat home directory

```
chwon -hR tomcat:tomcat tomcat
```

Configure firewall

Install, start and enable the firewall

```
yum install firewalld  
systemctl start firewalld  
systemctl enable firewalld
```

Add to the firewall tables the port 8080 used by Tomcat.

```
firewall-cmd --zone=public --permanent --add-port=8080/tcp  
firewall-cmd --reload
```

Check the ports and the services in the firewall tables.

```
firewall-cmd --list-ports  
firewall-cmd --list-services
```

Test tomcat

To test tomcat, start it.

```
cd /opt/tomcat/bin  
. ./startup.sh
```

Open the server ip, with the port on the **8080** on a Browser, like **80.19.194.99:8080**

If the reserver shows the page, shut it down

```
./shutdown.sh
```

enable service

edit file:

```
/etc/systemd/system/tomcat.service
```

write in the file:

```
[Unit]
Description=Apache Tomcat 8 Servlet Container
After=syslog.target network.target

[Service]
User=tomcat
Group=tomcat
Type=forking
Environment=CATALINA_PID=/opt/tomcat/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

reload services, start service and enable it

```
systemctl start tomcat
systemctl enable tomcat
```

check service status

```
systemctl status tomcat
```

conf tomcat user

edit file:

```
/opt/tomcat/conf/tomcat-users.xml
```

add rows:

```
<role rolename="manager-gui"/>
<user username="admin" password="password" roles="manager-gui,admin-gui"/>
```

then edit the file:

```
/opt/tomcat/webapps/manager/META-INF/context.xml
```

comment the rows 19, 20:

```
<Context antiResourceLocking="false" privileged="true" >
    <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="127\\.\\d+\\.\\d+\\.\\d+|::1|0:0:0:0:0:0:1" /> -->
</Context>
```

edit file:

```
/opt/tomcat/webapps/host-manager/META-INF/context.xml
```

comment the rows 19,20:

```
<Context antiResourceLocking="false" privileged="true" >
    <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="127\\.\\d+\\.\\d+\\.\\d+|::1|0:0:0:0:0:0:1" /> -->
</Context>
```

reboot the server and try to connect again on **ip:8080**

configure Apache Reverse Proxy

Apache will be used as Reverse Proxy, to redirect the requests from the apache web server to Apache Tomcat.

Install Apache on CentOS, enable it and start it.

```
yum install httpd      # install  
systemctl enable httpd # enable  
systemctl start httpd  # start
```

Then create the virtual host. Create the file `/etc/httpd/conf.d/tomcat_rev_proxy.conf`.

```
<VirtualHost example.domain.com:80>  
  ServerName domotics.domain.com  
  
  ProxyRequests Off  
  ProxyPass /examples ajp://localhost:8009/examples  
  ProxyPassReverse /examples ajp://localhost:8009/examples  
</VirtualHost>
```

With this it will be displayed the the Apache Tomcat web-app `/example` on the request to the Proxy `example.domain.com/examples`.

After you configured the reverse proxy, check the configuration:

```
apachectl configtest
```

It should return something like `Syntax OK`, if it returns it, restart Apache httpd, with:

```
systemctl restart httpd
```

Now configure tomcat to accept connections from the Apache httpd reverse proxy.

Edit configuration tomcats file, `/opt/tomcat/conf/server.xml`, and go to the line that contains `Connector` and edit it as follows:

```
<Connector address="127.0.0.1" port="8009" ...
```

Then restart tomcat:

```
/opt/tomcat/bin/shutdown.sh  
/opt/tomcat/bin/startup.sh
```



Before you test if everything works open the port 80 on the firewall, with the command:

```
firewall-cmd --permanent --add-port=80/tcp  
firewall-cmd --reload
```

Now open your browser and open the page `example.domain.com/example`.