



Istituto di Scienza e Tecnologie dell'Informazione
“A Faedo” (ISTI)
Laboratorio di domotica

Introduzione al software open source per la domotica

Dario Russo
(dario.russo@isti.cnr.it)





Cosa è l'open source

(termine inglese che significa sorgente aperto) indica un software rilasciato con un tipo di licenza per la quale il codice sorgente è lasciato alla disponibilità di eventuali sviluppatori, in modo che con la collaborazione (in genere libera e spontanea) il prodotto finale possa raggiungere una complessità maggiore di quanto potrebbe ottenere un singolo gruppo di programmazione e che la conoscenza sia messa a disposizione di tutti.



- Richard Matthew Stallman (Manhattan, New York, 16 marzo 1953) è un informatico statunitense. È uno dei padri del concetto del copyleft e un pioniere nel concetto di software libero.
- Nel 1989 ideò il concetto di copyleft (riutilizzo, modificabilità e diffusione sotto certe regole dipendenti dalla licenza).





Nasce la GNU General Public License (GPL), il preambolo del cui manifesto comincia con:

« Le licenze per la maggioranza dei programmi hanno lo scopo di togliere all'utente la libertà di dividerlo e di modificarlo. Al contrario, la GPL è intesa a garantire la libertà di condividere e modificare il free software, al fine di assicurare che i programmi siano "liberi" per tutti i loro utenti. »



Konnex - Engineering Tool Software

ETS è un software per disegnare e configurare le installazioni domotiche basate sul sistema Konnex;

- ETS gira esclusivamente su ambiente Windows ed è un software proprietario (necessita di licenza a pagamento);
- si basa su librerie non libere (Falcon);
- ad oggi è l'unica possibilità reale per poter configurare il sistema.



- Institute of Computer Aided Automation di Vienna ha realizzato una serie di software per la gestione del bus e dei dispositivi konnex.
- Il software è realizzato usando principalmente Java e ha pubblicato i sorgenti su sourceforge.
- <https://www.auto.tuwien.ac.at/lab/software.html>



- Software Development Kit che permette di scrivere le propri applicativi da scaricare (attraverso EIBD, network access server) sulle BCU M68HC05;
- permette una programmazione ad oggetti senza usare assembler (GCC/binutils + BCU headers/libraries);
- GCC permette il cross - compiling.



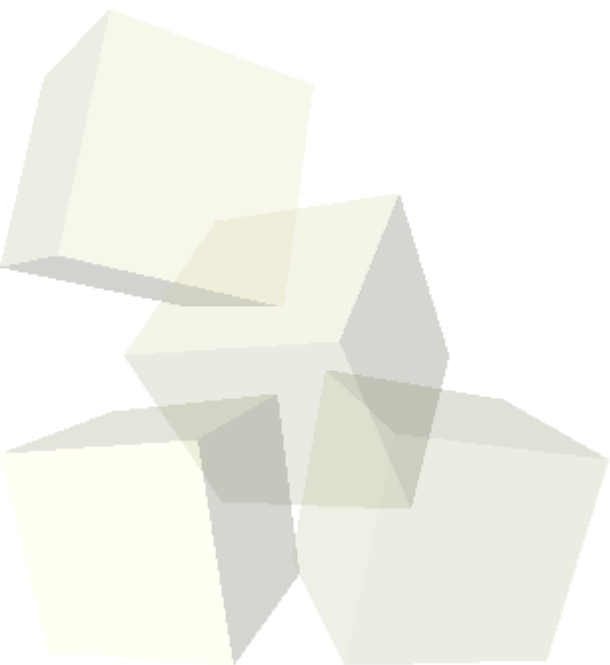
- GCC (GNU Compiler Collection) non è solo un compilatore per il linguaggio C;
- GCC offre frontend per i linguaggi:
 - ◆ C;
 - ◆ C++;
 - ◆ Objective-C;
 - ◆ Fortran;
 - ◆ Java;
 - ◆ Ada.

Piccola parentesi – Java open source

- Non esiste solo il Java versione Sun;
- Il Java versione Sun non era open source (si potevano scaricare i binari ma non i sorgenti);
- Nascita di progetti Java open source:
 - ◆ Kaffe (macchina virtuale);
 - ◆ gcj (compilatore Java GCC);
 - ◆ ecj (eclipse java compiler);
- Ultimamente Sun ha deciso di rendere disponibili parti del codice Java (<http://openjdk.java.net>).



- Nuovo approccio per pianificare e configurare bus domotici (attualmente solo Konnex);
- ancora in versione beta.





- Libreria per gestire in maniera semplice (lettura e scrittura) l'accesso al bus Konnex;
- si connette al bus konnex attraverso un adattatore IP;
- si basa su Tweety (KNXnet/IP Tunnelling server).





```
public static void main(String[] args) {  
    try {  
/** Creates a new connection to the specified  
    socket address and port number */  
        CEMI_Connection tunnel =  
        new CEMI_Connection(new  
            InetAddress(  
                "tunnelservice.somewhere.net",  
                EIBNETIP_Constants.EIBNETIP_PORT_NUMBE),  
            new TunnellingConnectionType());  
    }  
}
```



```
/** Creates the datatype using a major and  
minor structures. */  
PointPDUXlator dimVal =  
    PDUXlatorList.getPointPDUXlator(  
        PDUXlatorList.TYPE_8BIT_UNSIGNED[0],  
        PointPDUXlator_8BitUnsigned.  
            DPT_SCALING[0]);
```

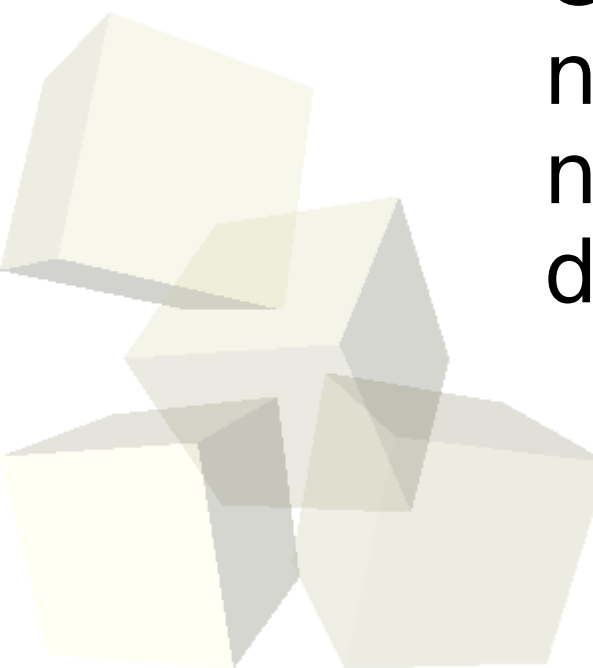


```
/** Setting to write mode and assigning a  
value. */  
dimVal.setServiceType(PointPDUXlator.  
                        A_GROUPVALUE_WRITE);  
dimVal.setASDUfromString("75");
```



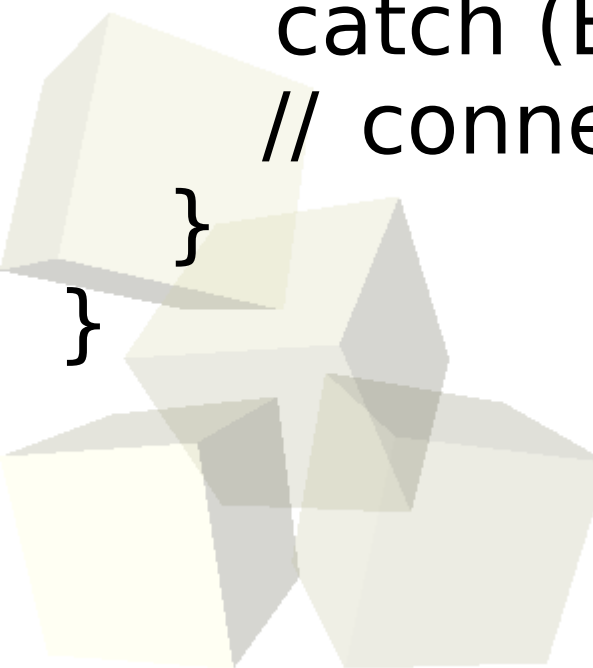


```
/** Creating the message to be sent. */  
CEMI_L_DATA message =  
    new CEMI_L_DATA(  
        CEMI_L_DATA.MC_L_DATAREQ,  
        new EIB_Address(),  
        new EIB_Address("0/0/1"),  
        dimVal.getAPDUByteArray());
```





```
/** Sending frame waiting for an ack. */  
tunnel.sendFrame(message,  
CEMI_Connection.WAIT_FOR_CONFIRM);  
}  
catch (EICLException ex) { }  
// connection error
```





Tuwien – esempio di Calimero

- L'esempio riporta una operazione di scrittura su bus ma si possono effettuare anche letture sullo stato dei dispositivi.
- La lettura, nonostante sia permesso dalla libreria, va contro la filosofia Konnex (i cambiamenti di stato ed i dati sono noti in quanto transitano sul bus) ma è possibile farla programmando opportunamente i dispositivi.
- Soluzione corretta: uso di un “database”.



- Distribuzione linux live basata su Knoppix 5.0.1;
- software preinstallato e pronto all'uso;

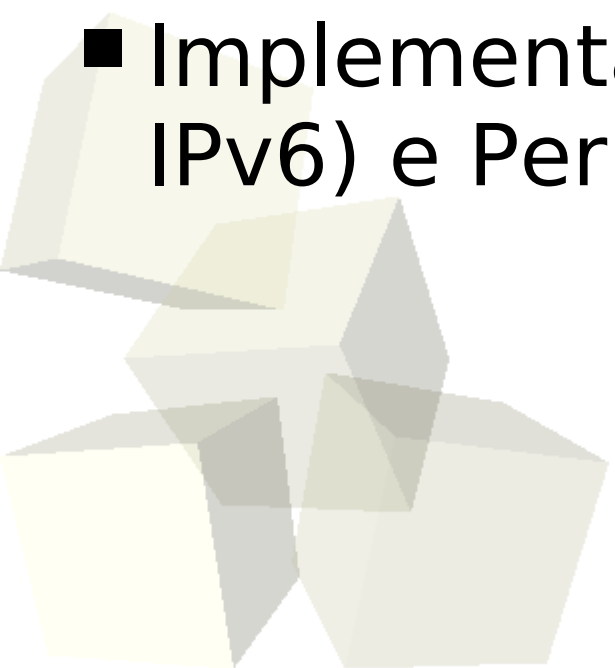




- Satoshi Konno - Tokyo, Japan;
- si occupa principalmente di sistemi di realtà virtuale;
- ha scritto librerie in Java e C++ per la gestione di UpnP.
- <http://www.cybergarage.org>



- E' un pacchetto di sviluppo per UPnP. Permette di creare e gestire dispositivi e control point facilmente.
- Implementato in C, C++, Java (con supporto IPv6) e Perl.





CyberGarage – esempio di CyberLink

```
/**
```

- * The listener for the UPnP devices.
 - * It captures every action performed in the
 - * UPnP network.
- ```
*/
```

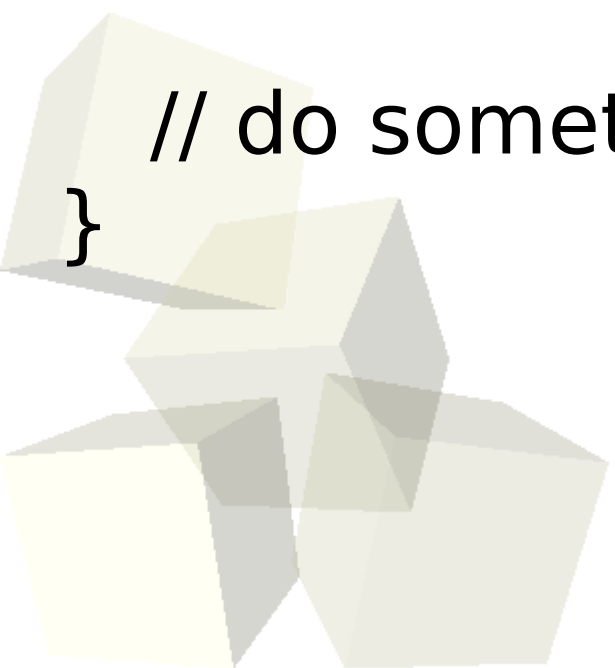
```
public class UPNPManagerPoint extends
 ControlPoint
 implements NotifyListener,
 EventListener, SearchResponseListener,
 DeviceChangeListener,
 HTTPRequestListener {
```

```
...
}
```

```
/** Executes an action. */
Action action =
 device.getAction(actionName);
action.setArgumentValue(argumentName,
 agumentValue);
```

```
/** Executes the action and get the result. */
if (action.postControlAction()) {
 // operation executed successfully.
 // getting output argument list
 outArgList =
 action.getOutputArgumentList();
}
```

```
/** Takes output values */
for (int n = 0; n < nArgs; n++) {
 Argument arg =
 outArgList.getArgument(n);
 // do something
}
```





E' una implementazione basata sulla libreria CyberLink che implementa un UpnP AV MediaServer. Può supportare i file system locali e alcuni progetti PVR (Personal Video Recorder) come MythTV (<http://www.mythtv.org>).





Sul sito di UpnP (<http://www.upnp.org>) è possibile trovare le specifiche dello standard e dei dispositivi (Basic Device, MediaServer, MediaRenderer, Stampanti, Scanner, Controllo Luci ecc.).





# Cidero – Software UPnP per A/V

- CyberLink offre il protocollo di base per la comunicazione “di base” coi dispositivi domotici.
- L'audio / video ha bisogno di un livello implementativo maggiore (implementazione delle funzionalità del MediaServer e MediaRenderer) in quanto prevede l'uso di protocolli (per la sincronizzazione tra i dispositivi).



- Cidero (<http://www.cidero.com/>) offre una soluzione in Java “pronta all'uso”, ma non troppo, per gestire dispositivi A / V implementando i protocolli necessari per la comunicazione tra MediaServer e MediaRenderer.
- Si basa sulla libreria CyberLink.
- Il codice da scrivere è analogo a quello visto per CyberLink.



Intel offre gratuitamente una suite di tool per UPnP sotto piattaforma Windows tra cui uno sniffer, un MediaServer e MediaRenderer utili per testare e sviluppare le proprie applicazioni.



- Tecnologia domotica su powerline ormai obsoleta rispetto a Konnex;
- funzionalità e dinamicità limitate (ogni dispositivo può avere al più un indirizzo) anche se con l'ausilio di alcuni apparati (CM-11) si possono definire macro;
- facile reperibilità, installazione, programmazione ad un costo contenuto.



- Controllo luci attraverso pulsantiere, telecomandi, sensori di presenza ad onde radio;
- impianti di allarme, di riscaldamento;
- controllo tapparelle attraverso sensori crepuscolari;
- installazione esterna e / o interna ai muri;
- il vero telecomando universale.



- Miriade di librerie per sviluppare applicativi per X-10 disponibili in molti linguaggi di programmazione (java, c, c++, perl).
- Miriade di progetti software esistenti che usano X-10





- <http://misterhouse.sourceforge.net/>;
- gestione della casa tramite interfaccia web altamente configurabile e decisamente accattivante;
- possibilità di creare script perl per estendere le funzionalità X-10, aggiungere logica, programmare il comportamento della casa in base agli eventi in corso.



- <http://www.linuxmce.org/>;
- distribuzione linux basata su KUbuntu;
- gestione multimediale (include anche MythTV) e della domotica (X-10);
- studiato appositamente per essere visto sul televisore di casa;
- può partire direttamente in modalità Media Center con dei menù intuitivi e completi;
- wizard efficaci per la configurazione della macchina e dell'ambiente domestico.

# Costruirsi il proprio Linux Media Center

Sulla rete si trovano migliaia di tutorial più o meno esaustivi e dettagliati sul come costruirsi, configurarsi e mettere in opera un proprio linux media center a costi bassissimi e “riciclando” il vecchio (ma non troppo) hardware.

