



***iLib: RFID & Fondera Robot***



# Hacking Group Como

## Website and informations:

<http://www.hgcomo.org>  
[info@hgcomo.org](mailto:info@hgcomo.org)

## Relators and contacts:

**Ulisse**  
**Matteo**  
**Otacon22**

[ulisse@hgcomo.org](mailto:ulisse@hgcomo.org)  
[matteo@hgcomo.org](mailto:matteo@hgcomo.org)  
[otacon22@email.it](mailto:otacon22@email.it)

# WWWD (what we will do)

---

## Di cosa parliamo:

- Panoramica sulla Remote Frequency IDentification
- Architettura di un tag RFID
- Lettura / Scrittura di tags RFID tramite lettore transponder
- Potenzialità (anche distruttive) della tecnologia.
- Implementazione degli RFID nel Robot iLib
- Panoramica del robot HW/SW
- Approfondimento sul router "la fonera" utilizzato per la comunicazione wifi
- Internet degli oggetti

## Cosa Facciamo:

- Analisi di un tag RFID nel dettaglio
- Creazione di alcuni tags RFID personalizzati
- Dimostrazione pratica delle possibilità del router sociale "la Fonera"
- Dimostrazione pratica di iLib robot

# WWW!D

---

- **Di cosa NON parliamo ma che comunque è possibile chiedere su richiesta:**
- **Microcontrollori:** Argomento troppo vasto, disponibilità di esempi di programmazione, codice, applicaizoni, etc etc su richiesta (per PIC e AVR).
- **Software iLib:** Scritto in delphi liberissimi di chiedere sorgenti / spiegazioni
- **Firmware iLib:** " " Bascom AVR e Mikrobasic PIC " "
- Gestione della **comunicazione seriale:** Il lettore RFID e i vari moduli del robot comunicano via RS232, è possibile cheiderci informazioni su come utilizzare la comunicaizone seriale sotto linux in C e Python e sotto W32 in Delphi, VisualBasic e gli altri linguaggi che supportano i controlli ActiveX.
- **Cosa NON Facciamo:**
- Creazione di tags personalizzati per tutti: i tags su cui lavoriamo sono limitati e ottenuti come campioni da diverse case produttrici con cui siamo in contatto.

# I – Remote Frequency Identification

...In PostOrder...

**Identification** -> Codice univoco che viene associato ad un oggetto (o anche persona) reale

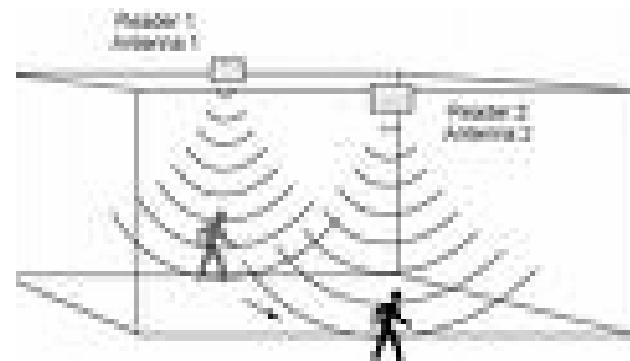
**Frequency** -> Comunicato utilizzando una trasmissione dati in frequenza (ASK / FSK)

**Remote** -> Al suo interrogatore remoto

**From barcode....**

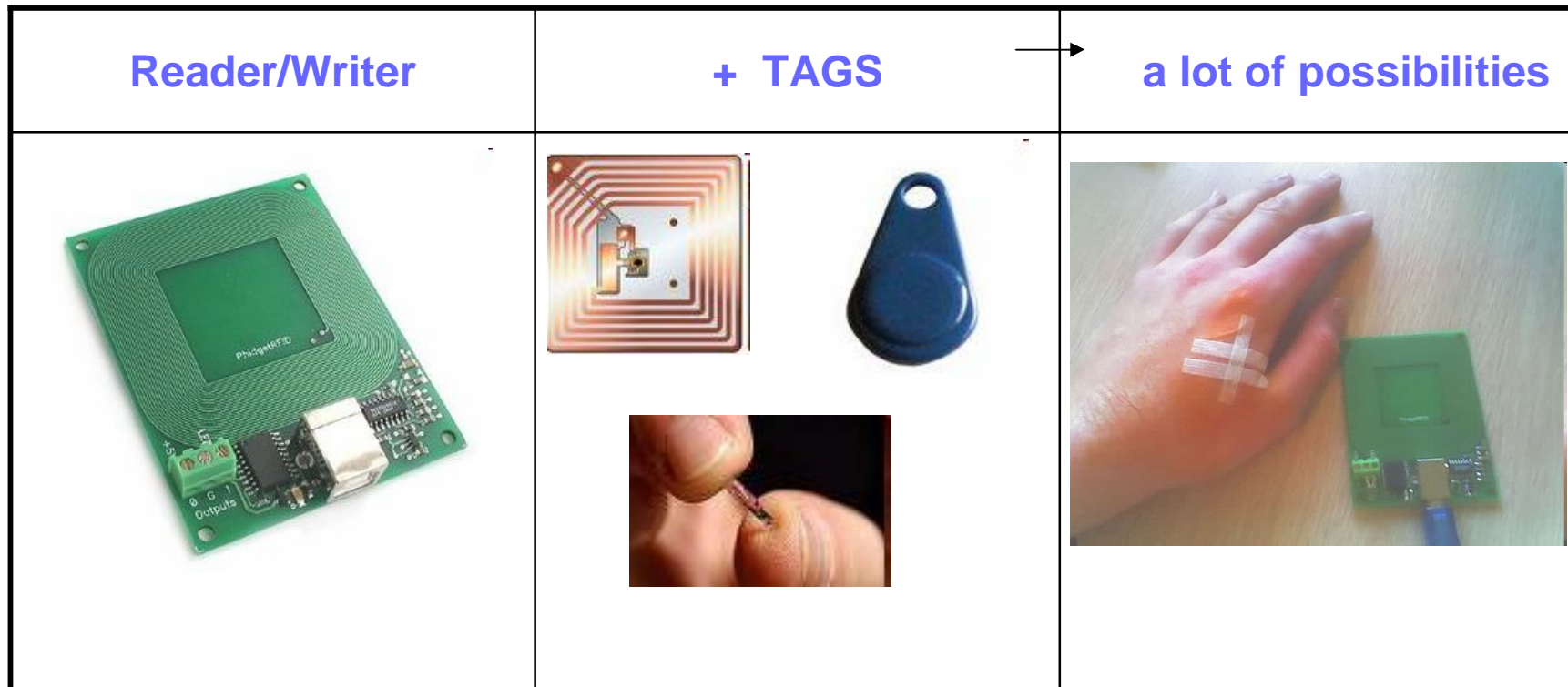


**.... To RFID**



La tecnologia RFID sta soppiantando l'uso dei codici a barre e degli altri sistemi di lettura ottica, non solo per la maggiore capacità di memoria, ma per la possibilità di gestire i dati implementando crittografia, algoritmi specifici per la comunicazione etc.

## II – RFID Architecture



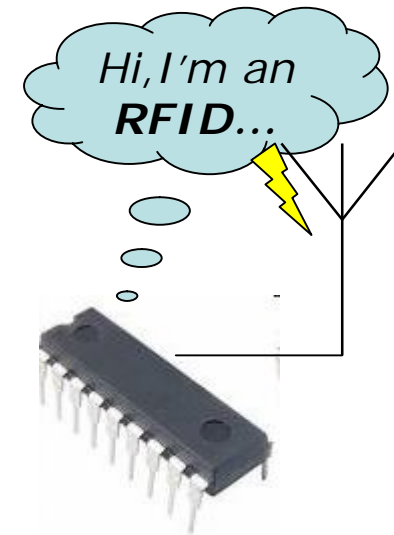
Tra le principali applicazioni si trovano **antitaccheggio**, **logistica**, **controllo accessi** e **riconoscimento** degli animali e **documenti personali** (passaporto etc etc.).

# III RFID Tags

Un **Tag RFID** è un **uController** dotato di **antenna**.

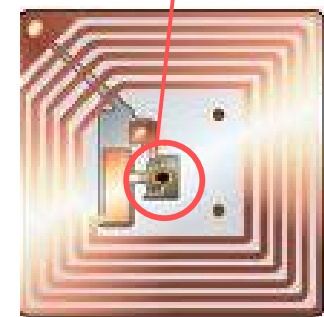
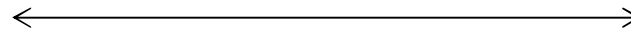
uController: Un microprocessore dotato di memoria dati e codice che esegue un programma (firmware).

Nel caso di un TAG RFID, il firmware gestisce la **comunicazione con il lettore** e i **dati memorizzati** sulla memoria interna consentendo la lettura e la scrittura.

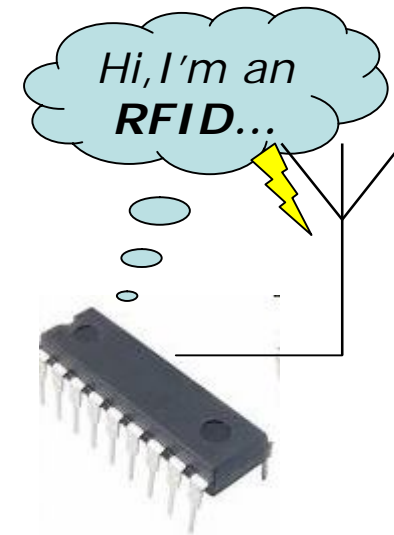
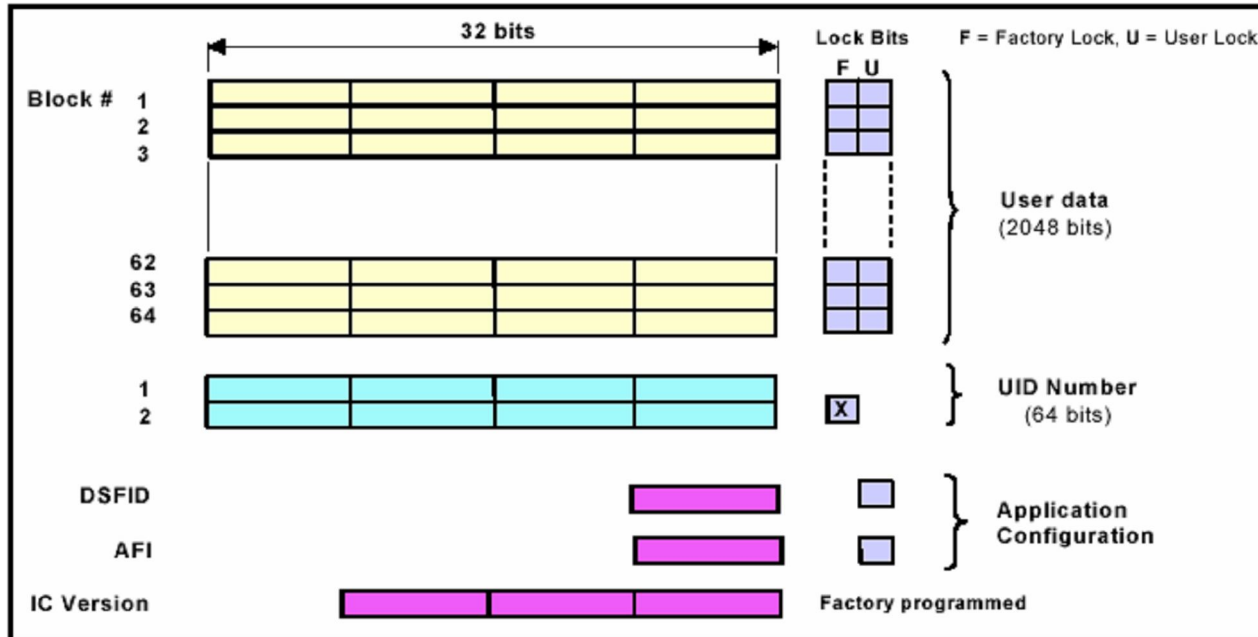


0x52	0x00
------	------

0x30	0x00
------	------

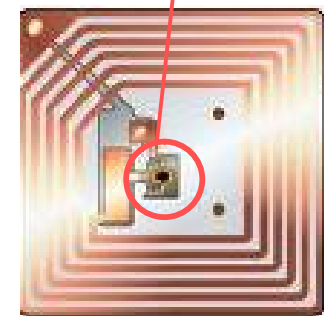
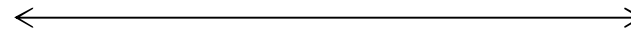


# VI RFID Tags memory structure



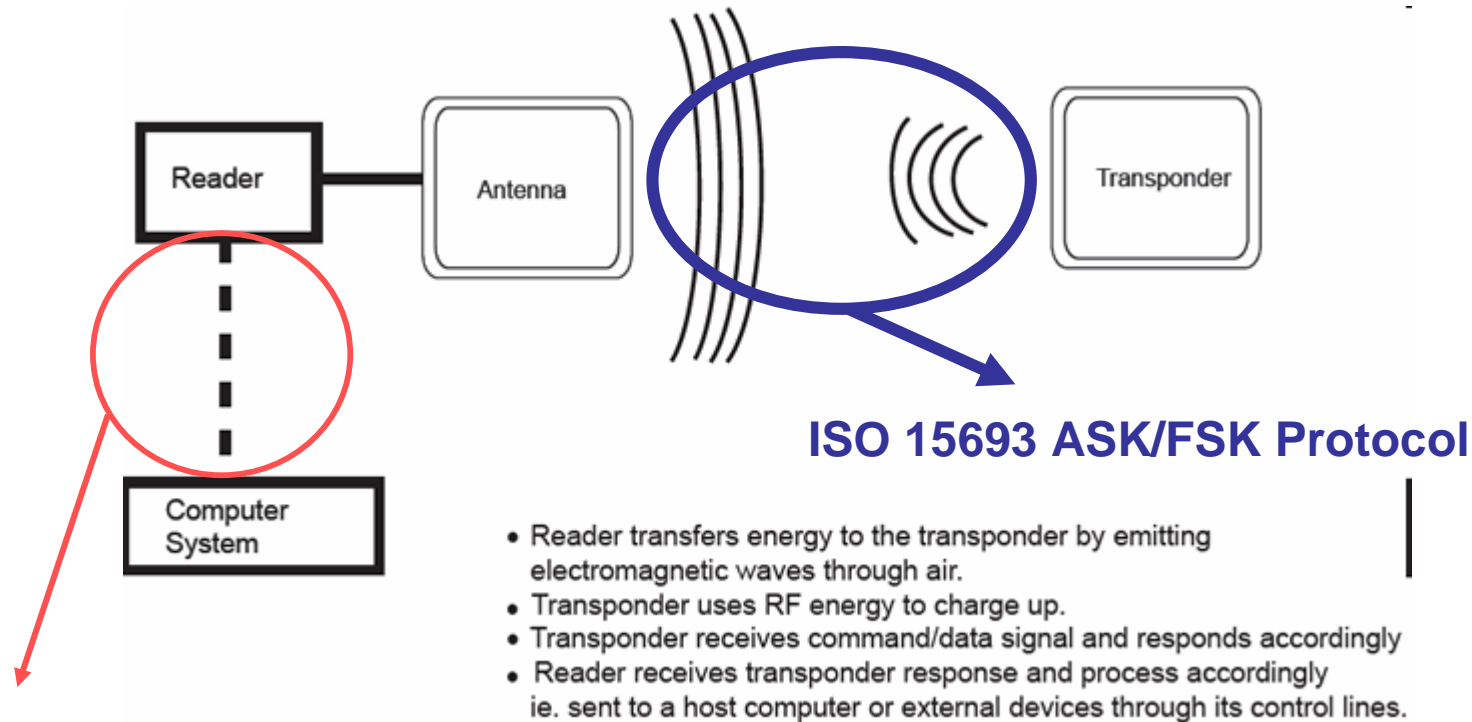
0x52 0x00

0x30 0x00





# V ISO 15693 UHF RFID Systems



In iLib serale RS232C  
(TTL)

Schema a blocchi di un sistema ad RFID

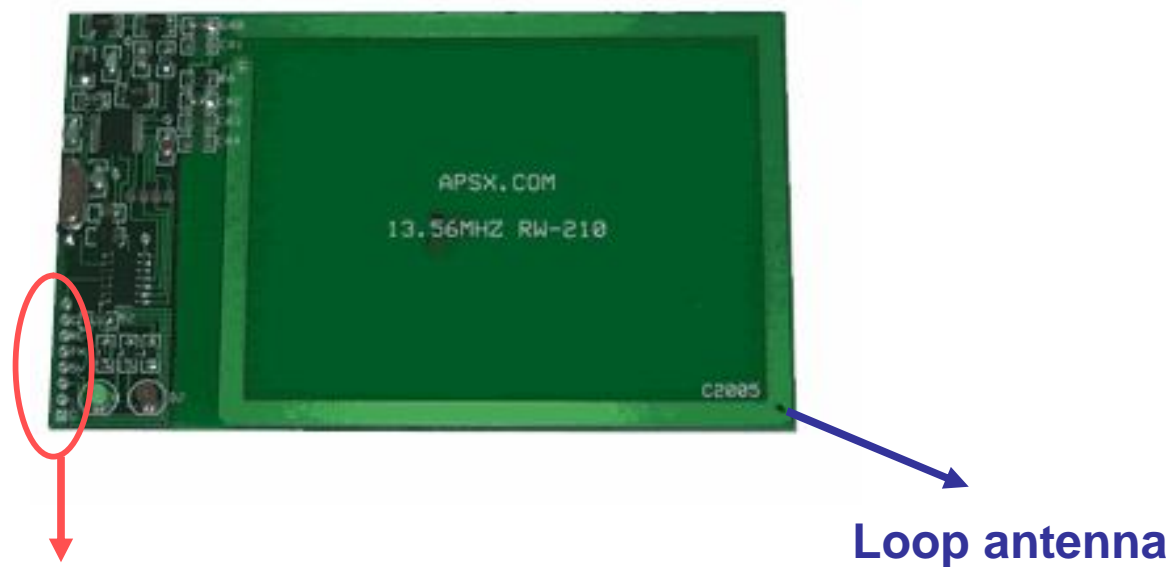
## VI RFID Reader/writer overview

---

In particolare vedremo il lettore RFID APSX RW210

Che grazie al suo semplice protocollo di comunicazione

Si presta bene per applicazioni didattiche e dimostrative



**Interfaccia seriale TTL  
+ alimentazione**

## VII RFID Reader/writer protocol

---

Setting Commands		
<b>FB</b>	<b>Switch in FAST MODE</b>	<b>Resta in listening per il passaggio dei tags</b>
<b>FA</b>	<b>Read UID</b>	<b>Consente di leggere la UNIQUE ID da un tag</b>
<b>READ BLOCK</b>	<b>legge un qualsiasi blocco</b>	<b>il blocco nel range 0-64, 4 byte per blocco</b>
<b>WRITE BLOCK</b>	<b>Scrivere su un blocco</b>	<b>il blocco nel range 0-64, 4 byte per blocco</b>
<b>READ/WRITE</b>	<b>8 bites</b>	<b>numero di bytes da inviare</b>
	<b>8 bites</b>	<b>numero di bytes aspettati</b>
	<b>8 bites</b>	<b>flag</b>
		<b>comando</b>
	<b>16 bites</b>	<b>CRC</b>

## VIII Creazione tags RFID

---



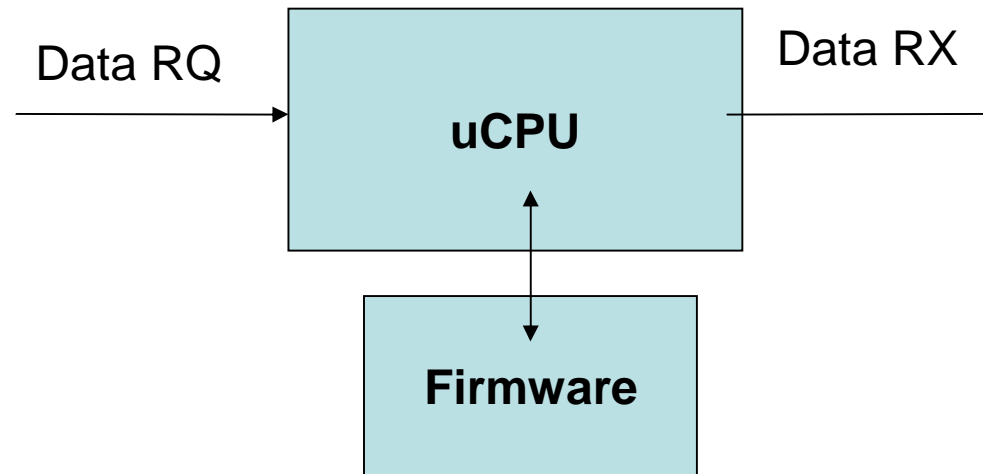
**A volunteer please**

Dimostrazione pratica di come utilizzare un TAG RFID per contenere delle informazioni

(scriveremo il nickname / altre cose a scelta fino a 2kbit in un TAG RFID HF)

# IX RFID [Tfdvsjuz]<sub>26, 1</sub> and not

---



- L'Architettura a microprocessore consente di manipolare il flusso di dati implementando variati algoritmi, purtroppo al momento accessibili solo alle case produttrici.
- E' possibile implementare crittografia dei dati affidandola all'applicazione server,

# X AND NOT at all – RFID Explosion

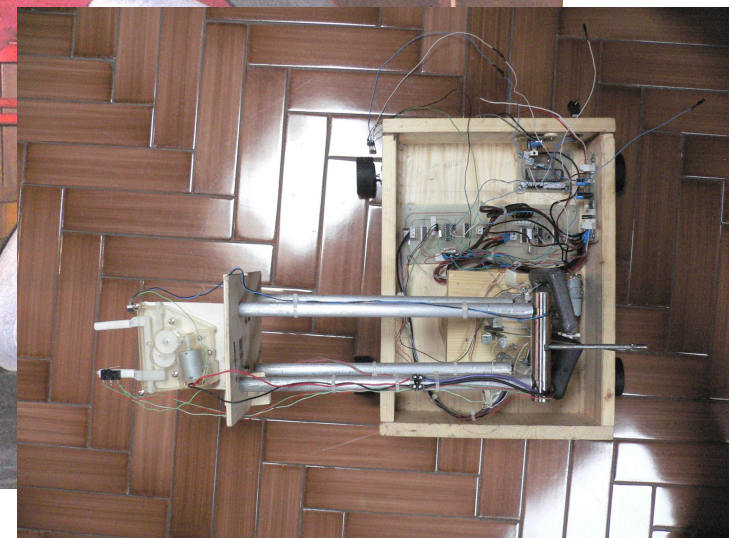
---



**Come back volunteers..**

**Dimostrazione pratica delle potenzialità anche pericolose della tecnologia RFID**

# XI RFID can be a wonderful thing - iLib



## XII il Progetto

---

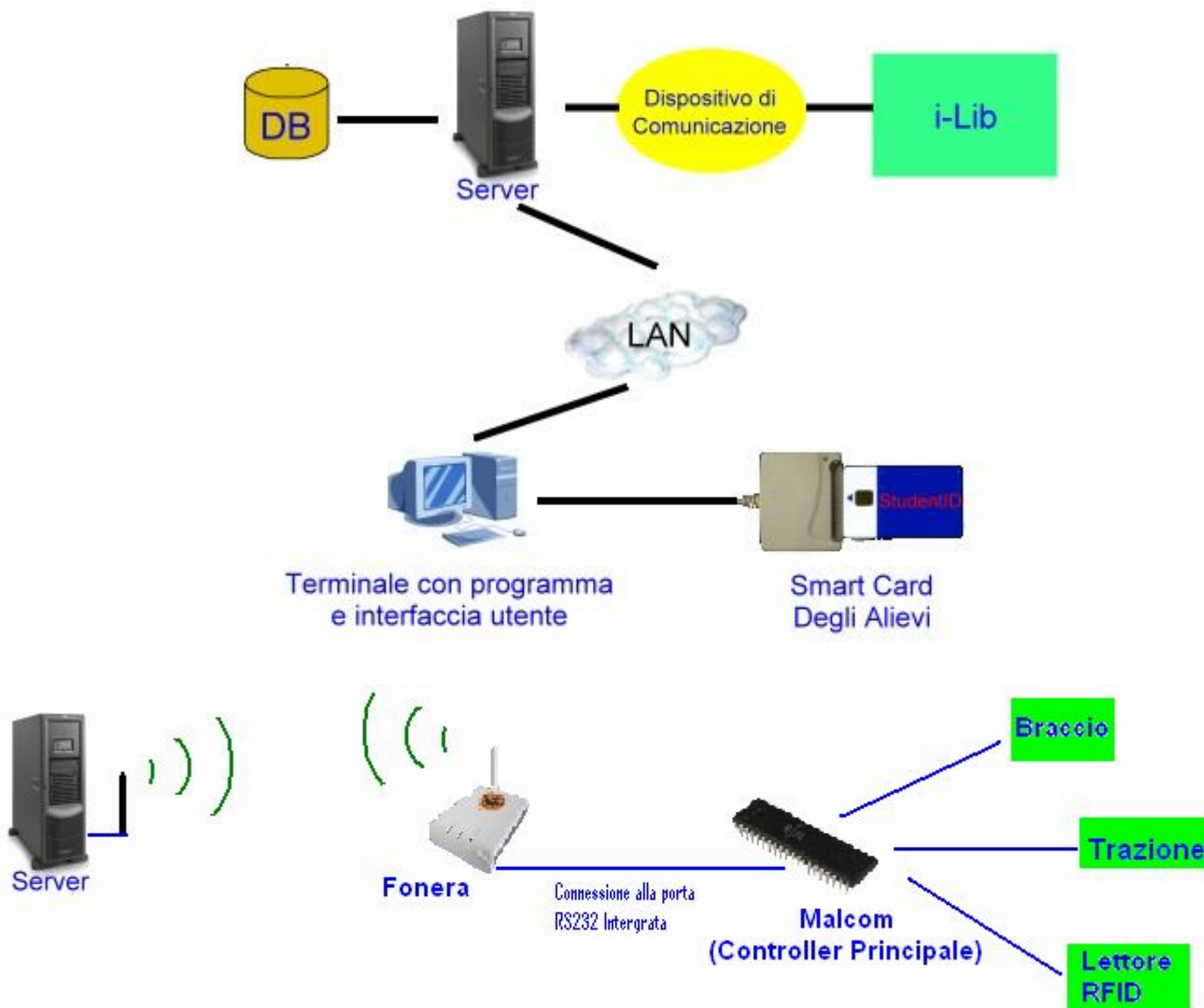
**Il Progetto nasce con l'idea di creare un sistema di gestione automatica per una biblioteca.**

**Questo sia per minimizzare l'intervento umano e quindi, furti e problematiche riguardo al prestito dei libri non correttamente registrato.**

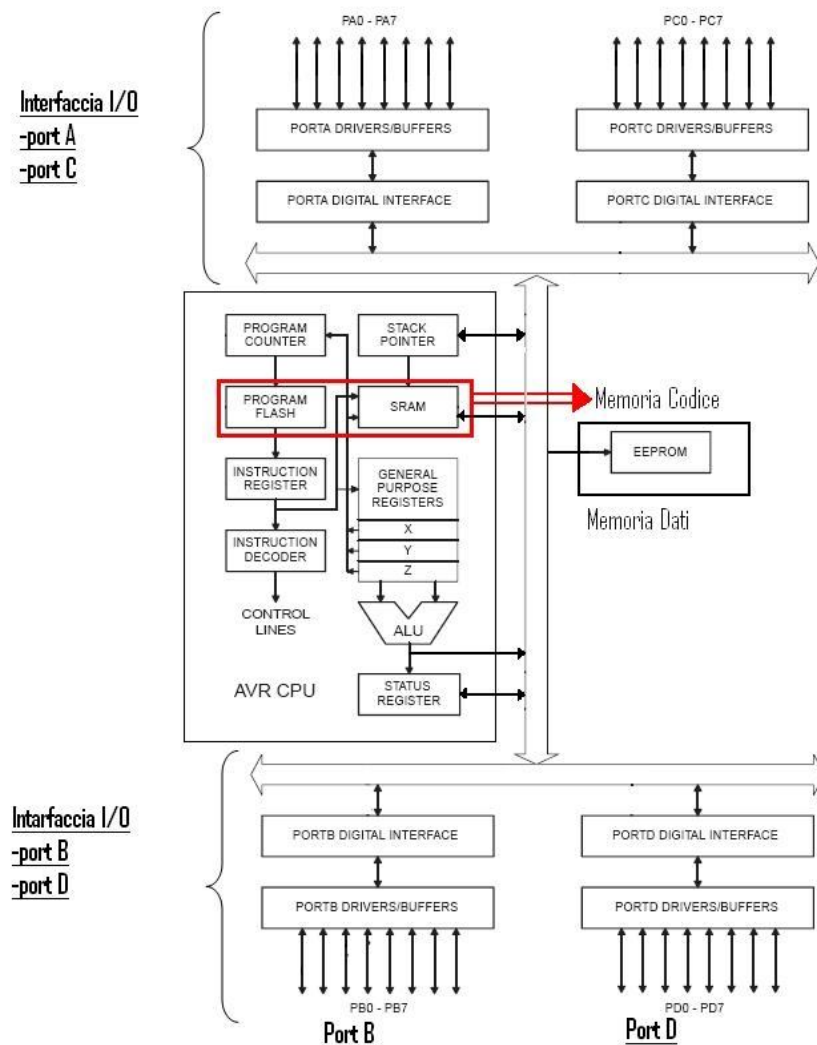
**Questo è stato possibile realizzando due parti distinte, una Software ed una Hardware, la quale è composta dal nostro robot "Malcom".**



# XIII iLib system overview



# XIV Microcontrollers architecture overview



I microcontroller costituiscono uno dei punti chiave sui quali verte il progetto.

Essi infatti offrono la possibilità di eseguire continuamente un programma “IL FIRMWARE” che hanno salvato nella loro memoria FLASH (Memoria di codice).

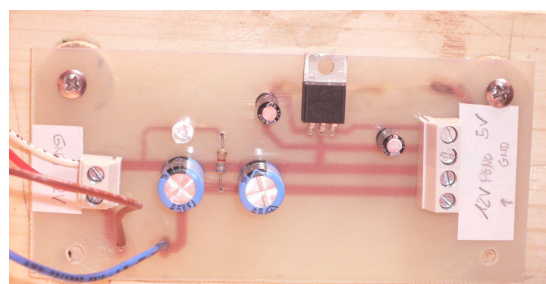
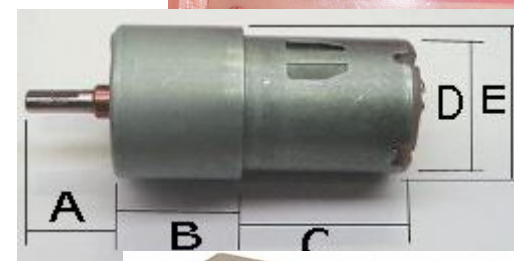
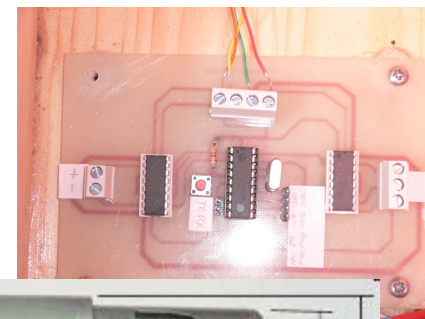
Abbiamo deciso di usare queste architetture, perché ormai hanno raggiunto bassi costi, i microcontroller che usiamo noi oscillano dai 5 ai 10€ l'uno, e hanno moltissime funzionalità e potenzialità.

I microcontroller integrano anche dei convertitori analogico digitali, e questo li rende dei dispositivi utilizzabili per realizzare controlli anche per qualsiasi applicazione elettronica.

# XV iLib components

Malcolm, è costituito principalmente da:

- Schede in vetronite, con a bordo microcontrollori, driver di controllo di potenza, etc.
- La Fonera, la quale è un router wireless che abbiamo flashato installandoci sopra OpenWRT 7.07, una particolare distro di Linux specializzata per target embedded.
- 6 Motoriduttori da 12kg/cm, per coordinare lo spostamento di Malcolm ed il movimento del braccio.
- Una batteria 12V9Ah per alimentare il tutto
- Un lettore RFID per identificare i Libri



## XVI iLib firmware overview

---

Il firmware per i microcontrollori ATMEL è stato scritto in “bascom AVR” un linguaggio che eredita la sintassi da Visual Basic, si tratta come Visual Basic di un linguaggio ad alto livello, che ci permette di scrivere i programmi per il microcontrollore astraendoci a livello umano.

Per quanto riguarda i PIC è stato usato mikrobasic della mikroelektronika ([www.mikroe.com](http://www.mikroe.com)), anch'esso eredita da BASIC la sintassi.

Una nota importante sui firmware è che girano in maniera sequenziale, è quindi necessario sempre creare un ciclo infinito, altrimenti dopo che ha eseguito le istruzioni il microcontroller smette di funzionare fino al prossimo reset.

Per attendere l'arrivo di un comando o l'attesa della risposta di un fine corsa, sono

stati introdotti dei cicli a vuoto che rimanevano in attesa attiva su quel comando/risposta.

Questo rappresenta il limite dei microcontrollori i quali necessitano di attese attive e quindi non sono in grado di fare altro se aspettano qualcosa.



# XVII Social router by fon.com "La Fonera"



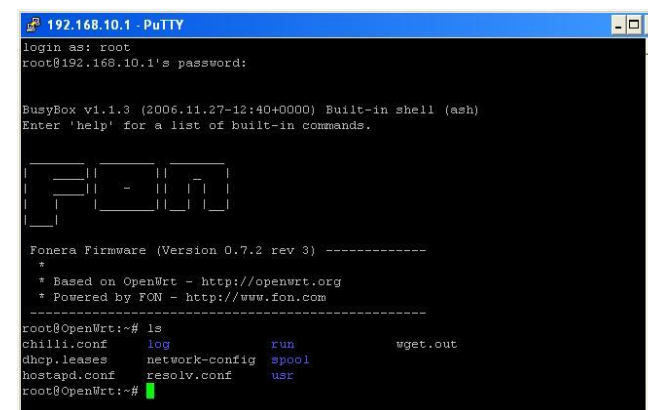
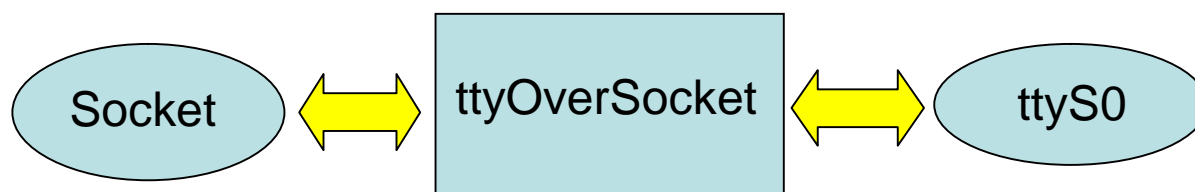
LA FONERA nasce come router wireless per la distribuzione di internet gratis per tutti, secondo la filosofia del movimento FON ([www.fon.com](http://www.fon.com)).

Noi abbiamo approfittato dell' offerta FON per poi installare sul router OPEN-WRT (v. kamikaze 7.07).



Subito dopo abbiamo pensato di utilizzare la porta seriale della fonera /dev/ttyS0 che prima usava per la shell, in una porta dedicata per comunicare con il nostro robot.

È stato quindi scritto un programma in C che sfrutta le socket in linux per poter scrivere sulla seriale e quindi mandare messaggi a malcom.



```
192.168.10.1 - PuTTY
login as: root
root@192.168.10.1's password:

BusyBox v1.1.3 (2006.11.27-12:40+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

FON

Fonera Firmware (Version 0.7.2 rev 3) -----
*
* Based on OpenWrt - http://openwrt.org
* Powered by FON - http://www.fon.com
-----

root@OpenWrt:~# ls
chilli.conf  log          run          wget.out
dhcp.leases  network-conf spool
hostapd.conf resolv.conf  usr
root@OpenWrt:~#
```

## XVIII iLib demonstration

---



**Dimostrazione pratica delle funzionalità di iLib**