John Lopez, Sabbia Erik, Pietro farci, Cristian Ferrando , Valerio Ameglio4AEA
ROBOTICA .A.S.2013-14

RELAZIONE DI ROBOTICA

TITOLO:

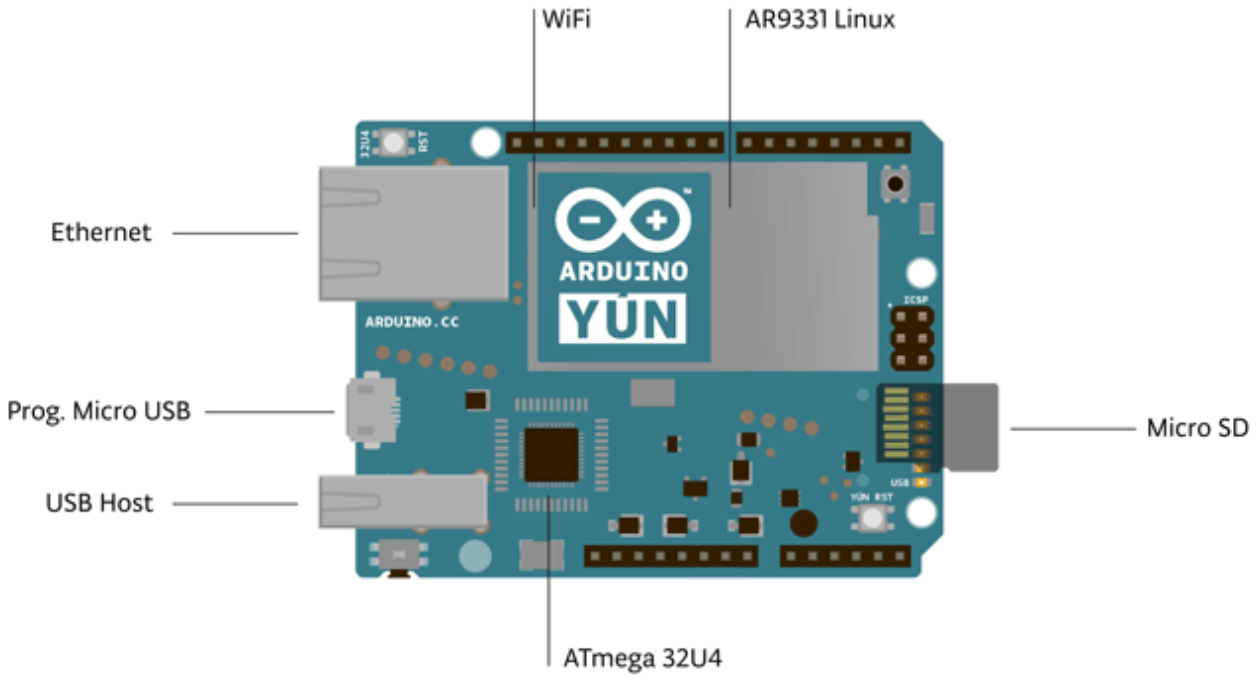
STUDIO DI ARDUINO YUN UTILIZZIATO COME SISTEMA DI SICUREZZA PHOTO-FRAME

SCOPO:

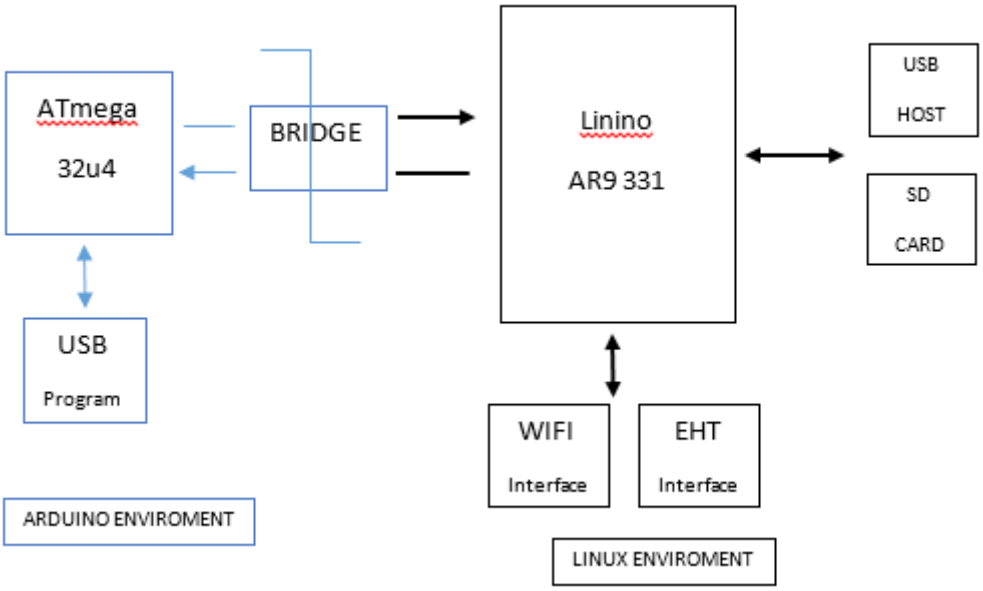
Fotografare tramite una webcam collegata ad Arduino Yùn tramite USB, e visualizzare le foto fatte tramite il browser.

RISORSE HW E SW:

HW ARDUINO Yùn: Si basa su un circuito stampato che integra un microprocessore ATmega32u4(16MHz di clock e 32kB di RAM) insieme a Atheros AR9331(400MHz, per la gestione di Linino Open WRT), a quest'ultimo è collegato il modulo WIFI che fornisce l'accessibilità wireless all'intero sistema oltre alla possibilità di collegare una antenna esterna.



Inoltre arduino yun dispone di interfacce I/O come:
-USB HOST
-Micro USB(per l'alimentazione e caricamento dati)
-SD CARD
-13 PORTE DIGITALI
-6 PORTE ANALOGICHE
-PORTA Ethernet



HW SHIELD: Questa motorshield per arduino è basata sull'L298: un completo ponte H doppio, progettato per il pilotaggio di carichi induttivi come relè, solenoidi e motori. La shield permette di controllare 2 motori tramite arduino in velocità e direzione, in maniera indipendente l'uno dall'altro. E' anche possibile misurare l'assorbimento in corrente di ogni motore.

SW ARDUINO: L'ambiente di sviluppo integrale (IDE) di Arduino è un'applicazione multi piattaforma scritta in Java, ed è derivata dall'IDE creato per il linguaggio di programmazione. Per permettere la stesura del codice sorgente, l'IDE include un editore di testo dotato inoltre di alcune particolarità, come il syntax highlig hing, il controllo delle parentesi, e l'indentazione automatica. L'editor è inoltre in grado di compilare e lanciare il programma eseguibile in una sola passata e con un solo click. In genere non vi è bisogno di creare dei Makefile o far girare programmi dalla riga di comando.

SW PUTTY: Per poter comunicare con Yùn tramite computer abbiamo utilizzato un programma chiamato PuTTY. Esso è un **client SSH**, emulatore di terminale e client Telnet e SSH per Microsoft Windows e sistemi GNU/Linux.

I passaggi per scattare una foto tramite PuTTY sono i seguenti:
-collegamento ad arduino yun tramite il suo IP
entrati in linino si inseriscono i seguenti comandi:
--cd/www/sd
---fswebcam -d/dev/video0 -r 320x240 test.jpeg
così facendo si scatterà una foto con una risoluzione di 320x240 e di nome test.jpeg, e verrà salvata nella directory SD.

----------TERMINI----------

Client- Il termine client indica anche il software usato sul computer client per accedere alle

funzionalità offerte dal server.

SSH- (Secure SHell) è un protocollo di rete che permette di stabilire una sessione remota cifrata tramite **interfaccia a riga di comando** con un altro **host** di una **rete informatica**.

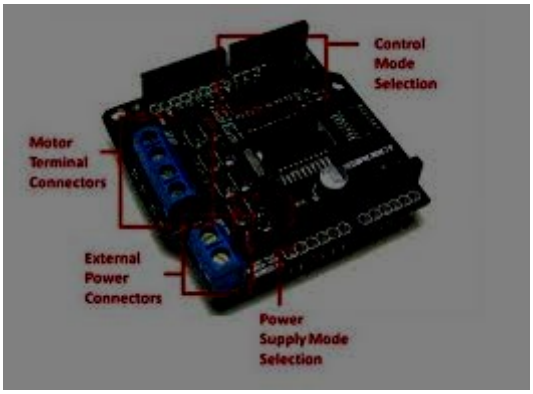
interfaccia a riga di comando-- indica una tipologia di interfaccia utente caratterizzata da un'interazione di tipo testuale tra utente ed elaboratore: l'utente impartisce comandi testuali in input mediante tastiera alfanumerica e riceve risposte testuali in output dall'elaboratore mediante display o stampante alfanumerici. (in breve, comunichiamo con il computer scrivendo tramite una shell).

rete informatica-- è un sistema o un particolare tipo di rete di telecomunicazioni che permette lo scambio o condivisione di dati informativi e risorse (hardware e/o software) tra diversi calcolatori. (es. Internet).

host-- O nodo ospite, indica ogni terminale collegato, attraverso link di comunicazione, ad una rete informatica (es. Internet).

terminale virtuale--- Dall'avvento e conseguente crescita di popolarità del personal computer, l'uso di terminali veri e propri per interfacciarsi con i computer è andato scomparendo. Usando il monitor e la tastiera, i sistemi operativi moderni come Linux e i derivati di BSD(La **Berkeley Software Distribution**, sigla **BSD**, è la variante originaria di Unix) implementano terminali virtuali, che sono in massima parte indipendenti dall'hardware usato. Durante l'uso di un'interfaccia grafica basata sull'X Window System(è un gestore grafico), come ad esempio GNOME(**GNOME** (**G**NU **N**etwork **O**bject **M**odel **E**nvironment) è un ambiente desktop) o KDE(Ambiente desktop Unix), non è possibile accedere alle periferiche che consentirebbero l'uso di un terminale virtuale. In questo caso viene normalmente usato un emulatore di terminale, un'applicazione che "finge" di essere un terminale (Il Terminale è un dispositivohardware elettronico o elettromeccanico che viene usato per inserire dati ingresso ad un computer o di un sistema di elaborazione e riceverli in uscita per la loro visualizzazione.) e fa sì che l'utente possa continuare ad accedere al computer nel modo in cui è abituato.

CRITERI DI PROGETTO: Abbiamo cercato in internet come poter far scattare foto ad arduino autonomamente , e dopo aver trovato un programma che potevamo utilizzare lo abbiamo modificato e testato ,nel programma c'erano funzioni di cui non conoscevamo nulla , ma eseguendo altre ricerche siamo riusciti a capirne la loro utilità e a utilizzarle successivamente nel programma per il video



ALIMENTAZIONE	4-12V
Motor controller	L298P, per pilotare 2 motori in continua
Max corrente	2A per motore

Funzione	Motore A	Motore B
Dir	D12	D13
PWM	D3	D11
BRAKE	D9	D8

PRODOTTI HW e SW:

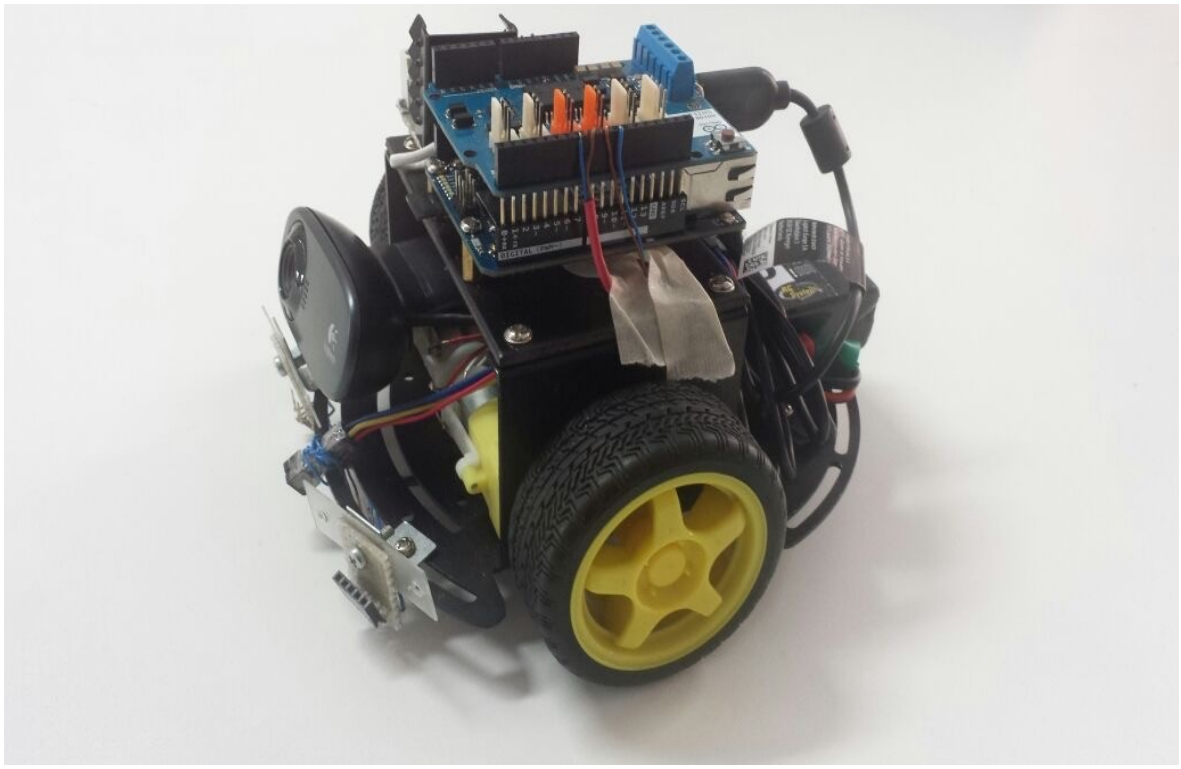
-SW

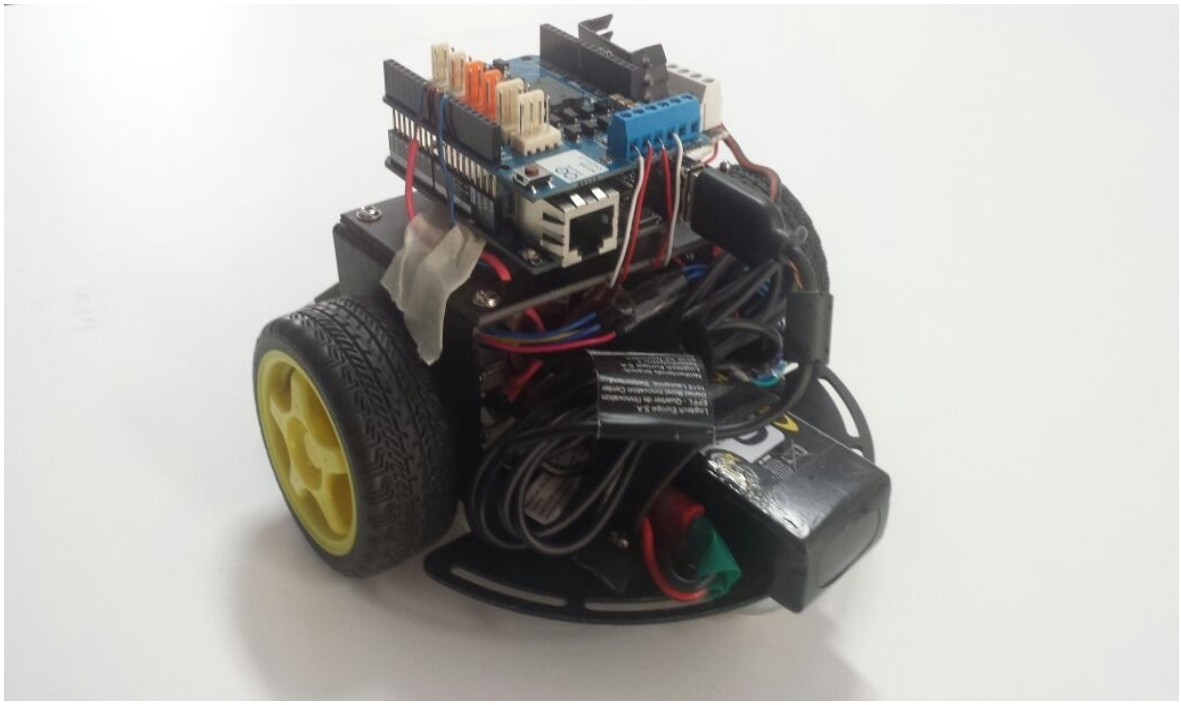
```
#include <Bridge.h>
#include <Process.h>


int pwm_a = 3;  //PWM control for motor outputs 1 and 2
int pwm_b = 9;  //PWM control for motor outputs 3 and 4
int dir_a = 2;  //direction control for motor outputs 1 and 2
int dir_b = 8;  //direction control for motor outputs 3 and 4


Process picture;


String filename = "Z";

String path = "-d/dev/video0";  //"/mnt/sda1/";


String p_SD = "/www/sd/";

int k=0;

void setup()
{

  Bridge.begin();

  pinMode(pwm_a, OUTPUT);  //Set control pins to be outputs
  pinMode(pwm_b, OUTPUT);
  pinMode(dir_a, OUTPUT);
  pinMode(dir_b, OUTPUT);

  analogWrite(pwm_a, 100);  //set both motors to run at (100/255 = 39)% duty cycle (slow)
  analogWrite(pwm_b, 100);

}

void loop(void)
{

    digitalWrite(dir_a, LOW);
    digitalWrite(dir_b, LOW);


    analogWrite(pwm_a, 20);
    analogWrite(pwm_b, 175);

    delay(1000);
```

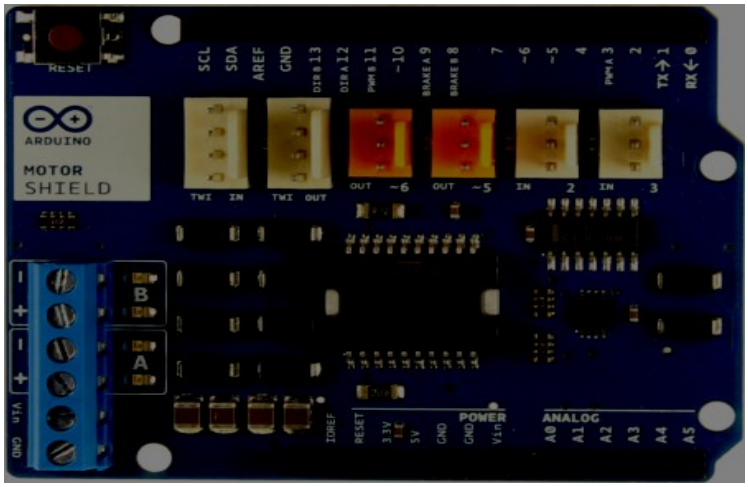
```
  picture.runShellCommand(p_SD);
  picture.runShellCommand("fswebcam " + path + " -r 1280x720 " + p_SD + filename + k++ + ".jpg");
  while(picture.running());

}
```

-HW

Ecco alcune foto del robot e delle singole schede.



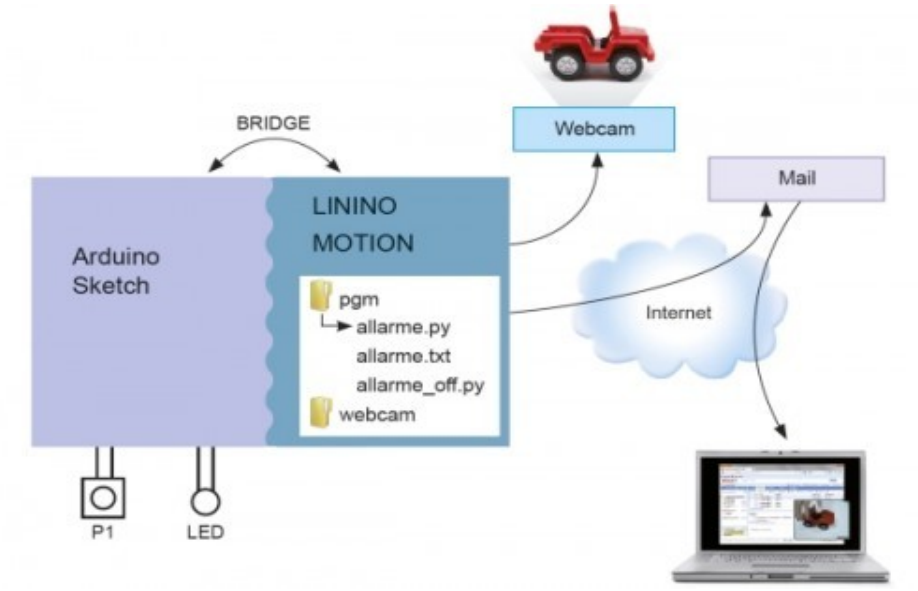




COLLAUDO:

Utilizzando l'ambiente di sviluppo Arduino e le nuove funzioni apprese dalle nuove librerie *"bridge"* e *"process"*, otteniamo come risultato la possibilità di entrare nella runshell di Linino e configurare la porta USB HOST alla quale è collegata la camera logitech HD tramite cavo USB, in questo modo tramite il comando *"fswebcam"* riusciamo a aprire l'otturatore della camera, impostare la risoluzione dell'immagine e dove salvare l'immagine ottenuta, che nel nostro caso è indirizzata alla scheda micro SD all'interno del lettore di cui

è provvisto Arduino Yùn per evitare il sovraccarico della memoria RAM del sistema.



Arduino Yùn inoltre ci permette, impostato come server, di visualizzare le immagini dalla rete internet collegandoci tramite l'indirizzo IP di Arduino (premesso che Yùn abbia l'accesso ad internet) immesso nel browser di qualsiasi computer del mondo.
Dovevamo creare un programma che ci permettesse di fare foto in movimento, ma non siamo riusciti a completarlo siccome durante la programmazione arduino yun ha smesso di funzionare "buttando" così due ore per provare a farlo comunicare di nuovo.
Inoltre siamo riusciti a installare un programma su Linino (tramite comandi "VI") che ci permette di trasmettere in streaming un video e visualizzarlo tramite la porta "(ip yun):8080".
Le librerie ed i comandi per programmare arduino in modo che riuscisse a comunicare con yun sono i seguenti:

Classi Bridge e Process

La libreria Bridge semplifica la comunicazione tra l' ATmega32U4 e l'AR9331. I comandi del Bridge, dal 32U4 vengono interpretati(ovvero viene analizzata e poi "lanciata" una riga di codice alla volta) da Python sul AR9331.

Bridge: Bridge è la classe base per tutte le chiamate Bridge. Non è chiamata direttamente, ma viene chiamata ogni volta che si utilizza una funzione che si basa su di essa.

(Es.: Bridge.begin())

.begin()- facilita la comunicazione tra l' AR ed il processore di linux. Deve essere chiamata una sola volta nel setup().

begin() è un blocco di funzioni. Una volta chiamata" Bridge.begin()", non succederà nulla nello sketch finchè non sarà completata. Questo processo dura all'incirca tre secondi.

Process: è usato per lanciare processi sul processore di linux , e altre cose come shell scripts. E' la classe base per tutti i bridge basati sulla comunicazione della shell di yun. Non è chiamata direttamente ma richiamata quando si una una funzione che si basa su di essa.

runShellCommand()

Fa partire una shell di comandi in linux.

runShellCommand() è una funzione "bloccante". Cioè, quando chiami Process.runShellCommand(), non succederà niente nel tuo sketch finchè non ha finito. il tempo dipende dalla natura del comando che si sta inviando.

(Dato che questa funzione interrompe lo sketch ("intrappola in un ciclo") e non ci permette di fare niente nel frattempo, meglio usare in comando runShellCommandAsynchronously()).

runShellCommandAsyncronously()

Fa partire una shell di comandi in linux.

Al contrario di runShellCommand(), runShellCommandAsynchronously() non blocca il programma. lo sketch continuerà a girare finche il processo di linux girerà in background.

(Permette di comunicare conla shell, mentre fa dell'altro. Es.: Fare video mentre si muove).

OSSERVAZIONI:

Molti comandi che abbiamo appreso durante lo studio di Yun, ci sono stati suggeriti dal professore di laboratorio o da siti in internet, ma siamo andati a vedere le loro funzioni in modo da non rimanerne all'oscuro e a capire il loro funzionamento, dopo svariati tentativi di programmazione abbiamo fatto funzionare YUN. dopo varie prove siamo riusciti a soddisfare in parte gli obbiettivi ,perchè riusciamo a fargli scattare autonomamente foto e a visualizzare in streaming dei video fatti dal robot ma senza che esso si muova , questo è causato soprattutto dalla mancanza di tempo per rifinire il programma, e da inconvenienti tecnici con il robot.