cout<<Controllo di LED da remoto<<endl;

Federico M.

4°A 2016/2017

INTRODUZIONE

Le possibilità di una scheda a microcontrollore come la RedBoard, collegata ad una scheda a microprocessore come Raspberry Pi, sono infinite.

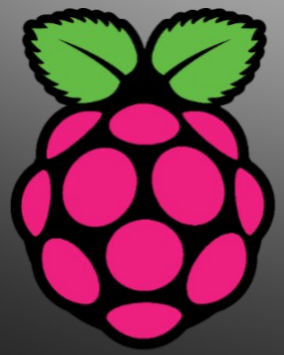
Nel nostro istituto ITIS B. Pascal, grazie alla collaborazione della CISCO e delle loro piattaforme online (PAAS) le possibilità sono ancor di più espandibili.

Amalgamando varie materie come telecomunicazioni, informatica, reti e tecnologie, mi è stata data la possiblità di usare il Prototyping-Lab Cloud (PL-Cloud) che semplifica il tutto rendendo comprensibili decenni di sviluppo tecnologico anche alle new entry nel mondo dell'informatica e di tutti i rami che ne conseguono.

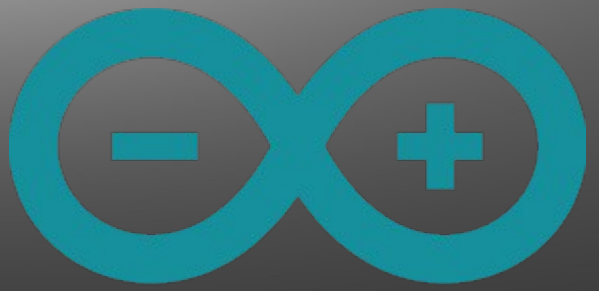
L'unico limite che ho, che abbiamo, è la fantasia.

Il progetto da me prototipato è un mix completo delle tecnologie (per ora) a noi dell'istituto disponibili.

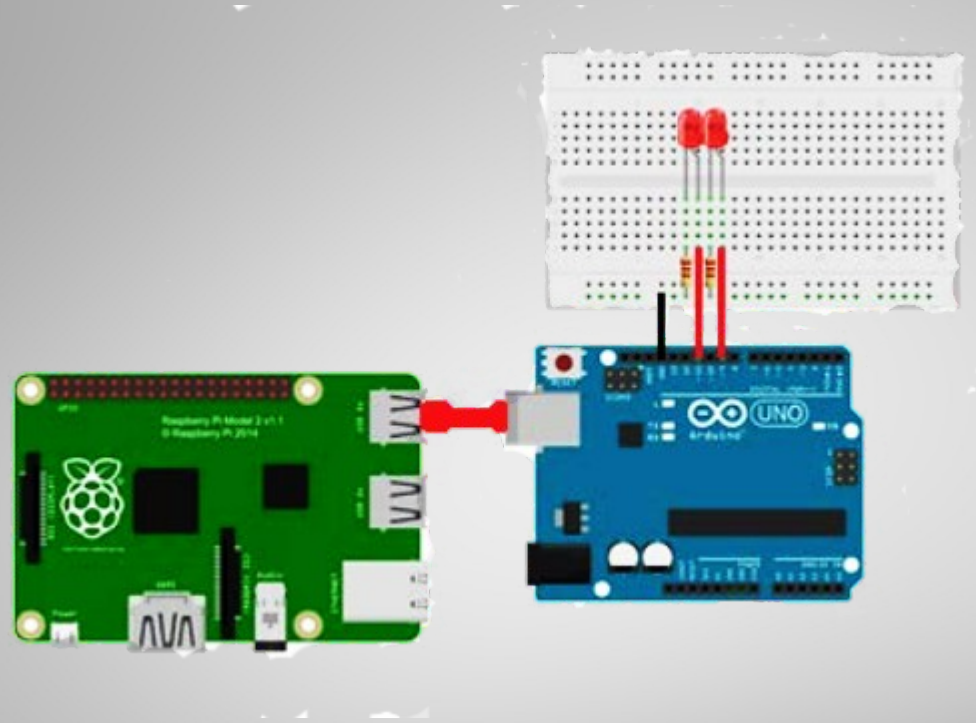
Si tratta di un circuito molto semplice ma che, grazie ai servizi Cloud dalla CISCO, diventa tutto molto più "magico", aprendo la mente a infinite possiblità.



RaspberryPi



ARDUINO



Il progetto serve a controllare dei semplici LED su BreadBoard che, collegata alla scheda a microcontrollore (RedBoard) che a sua volta è collegata alla scheda a microprocessore (RaspBerry Pi 3), possono essere controllati da remoto grazie al PL-Cloud offerto dalla CISCO.

Strumenti utilizzati per il progetto:

RedBoard/Arduino

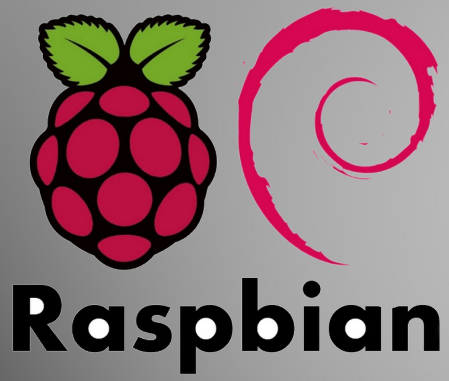
Raspberry Pi 3

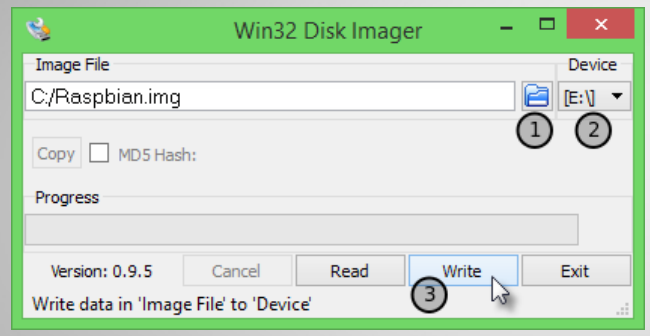
BreadBoard

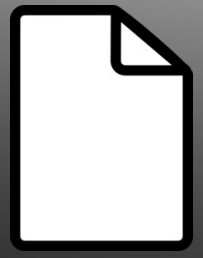
a cui sono collegati..:

- 2 LED
- 2 resistenze da 330 kΩ
- 3 cavi (PIN maschio-maschio)

TEORIA COINVOLTA





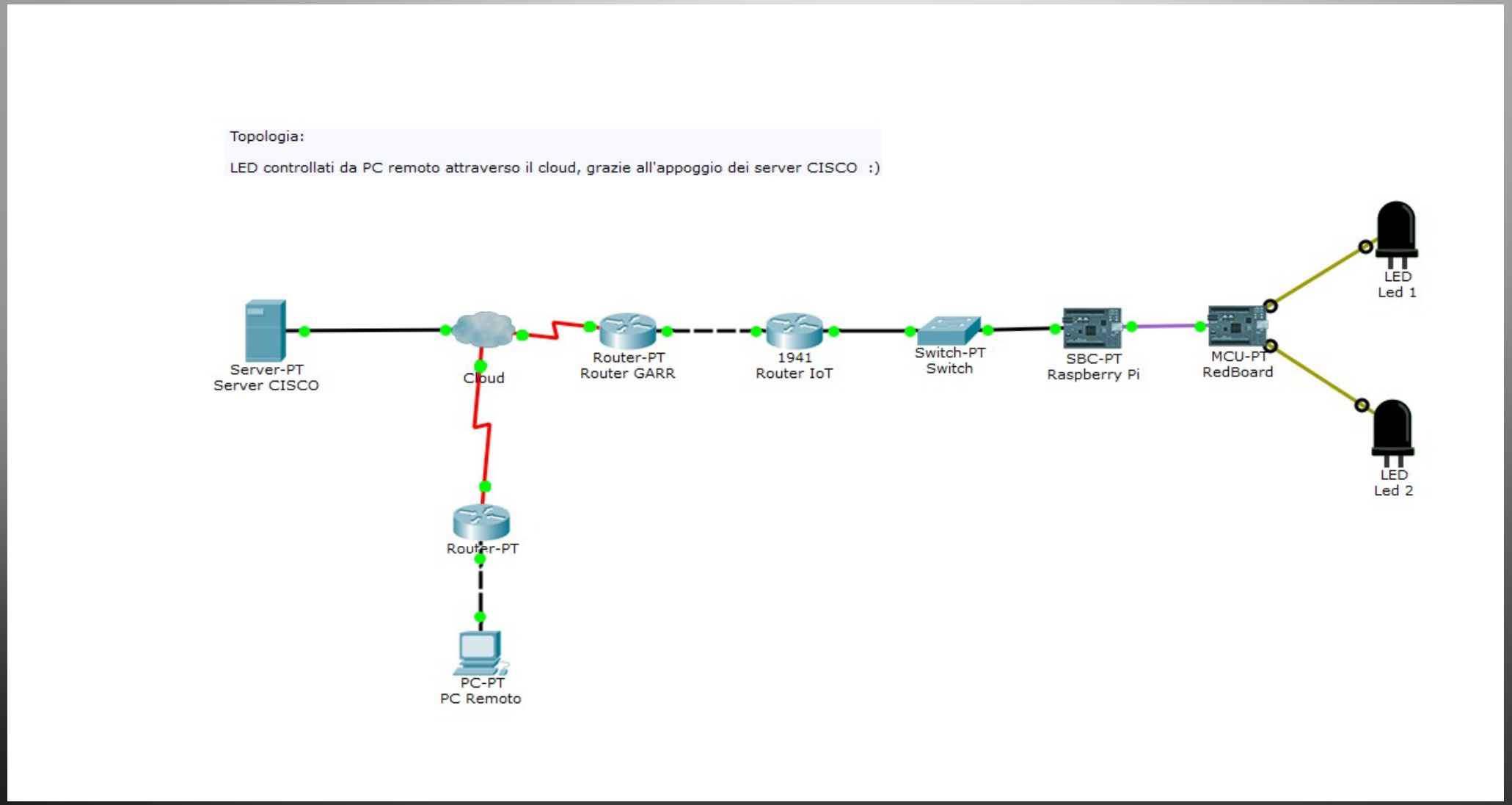


Il RaspBerry Pi e' un vero e proprio micro-computer con una scheda di rete integrata, e grazie a questo è collegabile ad Internet ed è possibile controllare i LED da remoto tramite il PL-Cloud.

Il primo passaggio che ho svolto, dopo aver configurato la rete in DHCP nella sala Internet of Things, è stato installare il Sistema Operativo Raspbian (gioco di parole Raspberry+Debian/Linux) tramite un software chiamato Win32DiskImager che rende una pendrive USB o, in questo caso, una Micro SD "bootable", inserita poi nello slot del RaspBerry Pi e installato automaticamente.

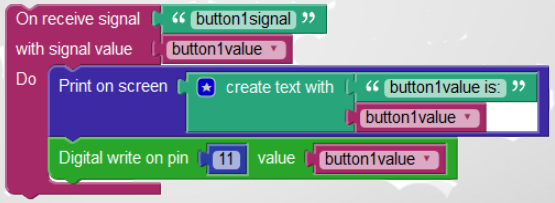
Oltre al Sistema Operativo, nella root della Micro SD, ho inserito un file che contiente un ID univoco del dispositivo, per garantire un miglior sistema di sicurezza.

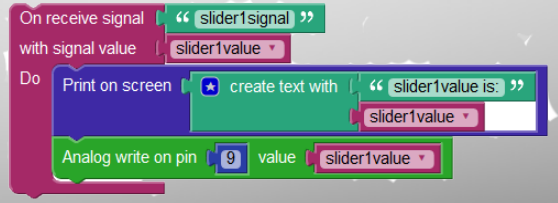
- Per avere un idea più chiara del progetto, ho simulato una topologia di rete grazie al software Packet Tracer 7.0:

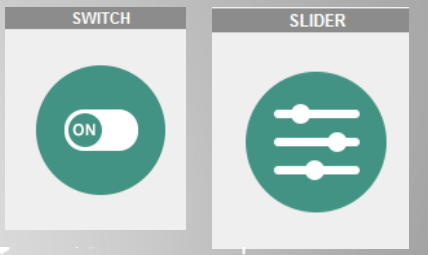


LINGUAGGIO COINVOLTO

Tramite il PL-Cloud, ho scritto un programma in linguaggio Visual Google Blockly automaticamente tradotto in linguaggio Python, il quale controlla tramite due attuatori virtuali (uno Slider e uno Switch-Button ON/OFF) i LED collegati all'Arduino UNO, spengendo e accendendo il primo LED con lo Switch e aumentando e abbassando la luminosità del secondo LED tramite lo slider.








```
from pyfirmata import Arduino
from pyfirmata import util

from wyliodrin import *

from time import *

import json

button1value = None
slider1value = None

def setBoard(boardType, port):
  if boardType == 'arduino':
    board = Arduino(port)
  else:
    board = ArduinoMega(port)
  return board
board=setBoard('arduino', '/dev/ttyUSB0')
reader = util.Iterator(board)
reader.start()

pin_var = board.get_pin("d:11:p")

pin_var2 = board.get_pin("d:9:p")


print('Hello from PL-Cloud')
sleep (2)
print('Ready')

def myFunction(__sender, __channel, __error, __message):
  global button1value
  button1value = int(json.loads(__message))
  print(str('button1value is:') + str(button1value))
  pin_var.write(button1value)

openConnection("signal:" + 'button1signal', myFunction)

def myFunction2(__sender, __channel, __error, __message):
  global slider1value
  slider1value = int(json.loads(__message))
  print(str('slider1value is:') + str(slider1value))
  pin_var2.write(slider1value/255.0)

openConnection("signal:" + 'slider1signal', myFunction2)
```

LABORATORIO