IOT Robots

2016

ISS Robots

Controllori, Attuatori, Sensori

MICROCOMPUTER

- Raspberry Pi

MICROCONTROLLERS

- Arduino
- ESP8266

- Motor driver
- Batterie

Sensori

- Ultrasonic Sensor
- Infrared line sensor
- Infrared distance sensor
- Camera

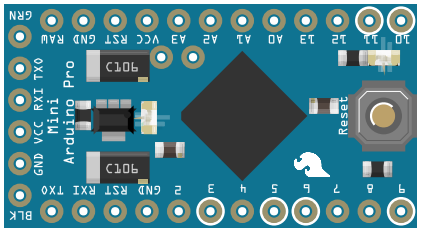
Attuatori

- DC Motor
- Servo
- Stepper Motor

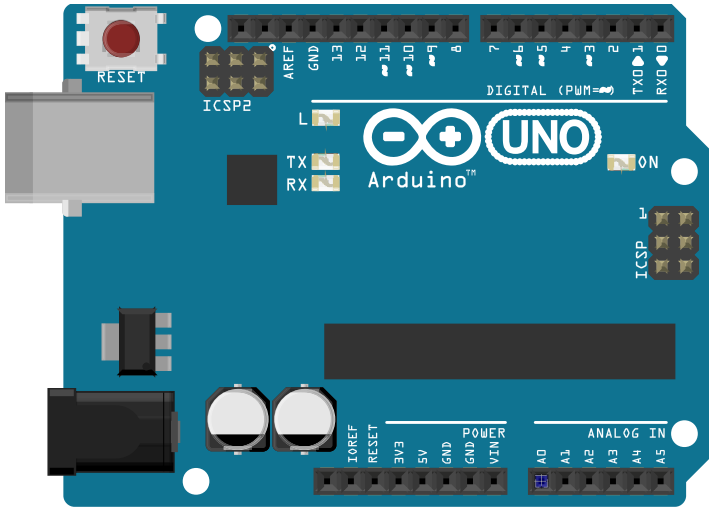
Arduino

Arduino

- ProMini



- Uno (Rev 3)



ATmega 328

RAM 2K

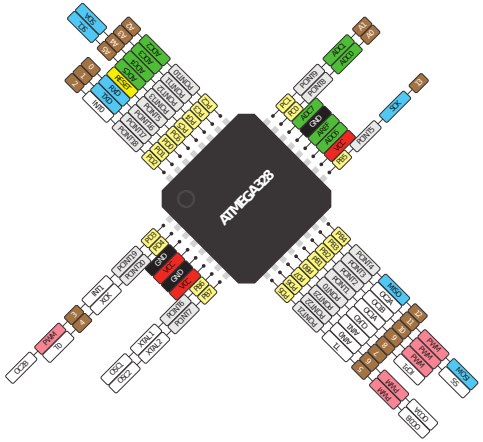
Flash 32K

Timers

UART

I2C

SPI

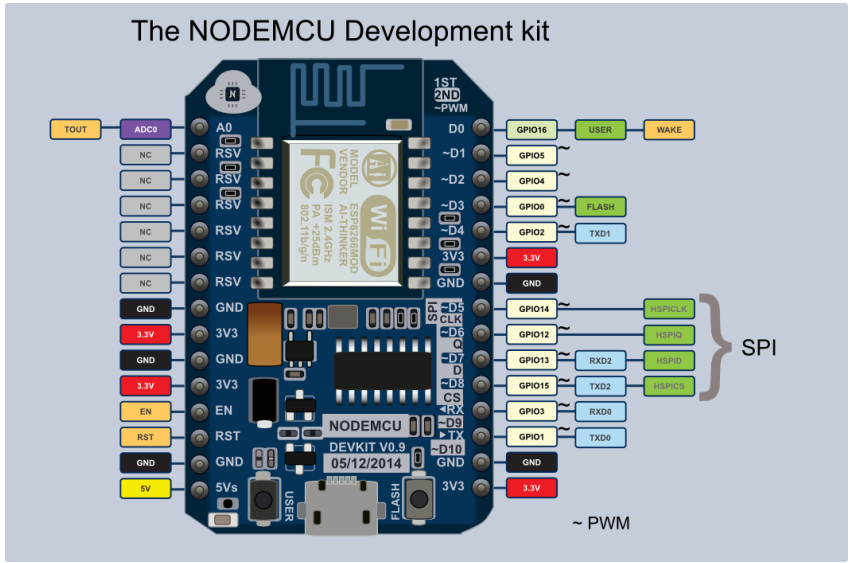


ESP8266

NodeMCU, Adafruit HUZZAH, Wemos D1

WiFi SoC ESP8266

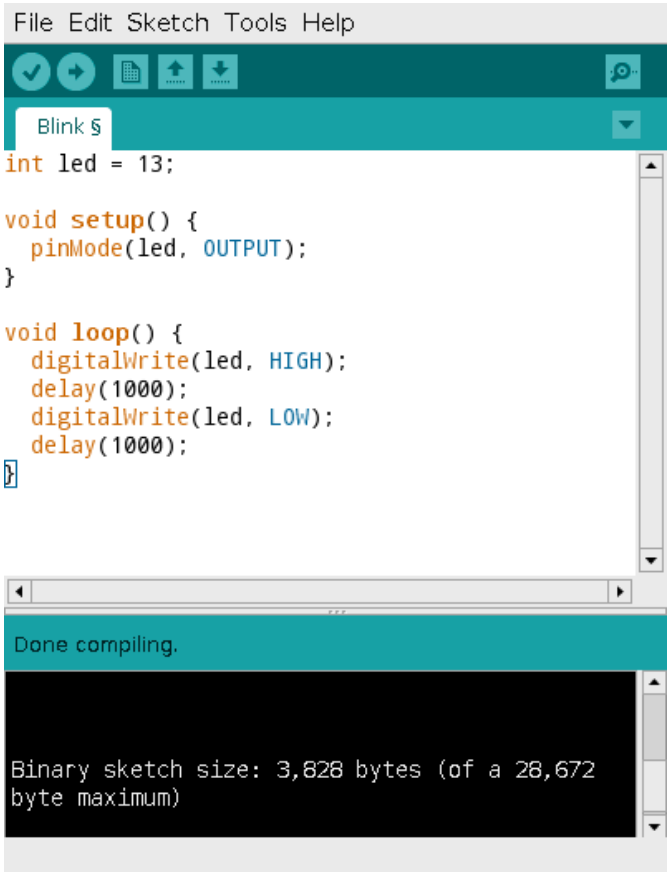
CPU	80MHz 160MHz
RAM	160K
BOOT ROM	64K
Flash	4M
Timers	
UART	
I2C	
SPI	
Wi-Fi	Direct (P2P), soft-AP
802.11	b/g/n
TCP/IP	Integrated protocol stack

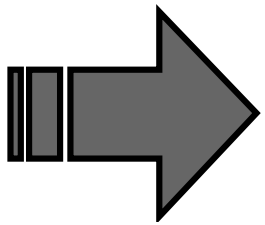


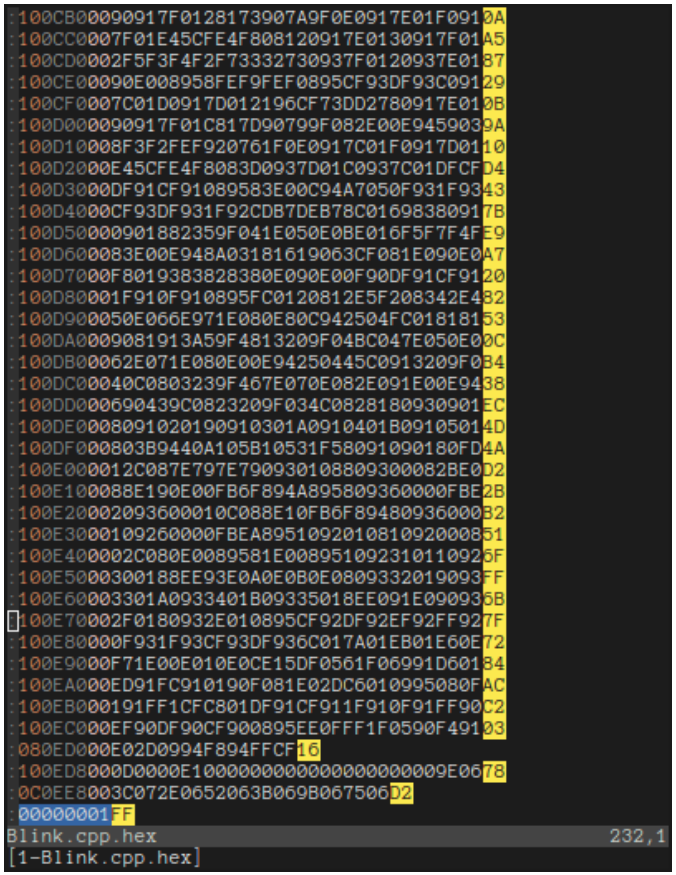
Come programmare?

- ESP-open-sdk (C)
- NodeMCU (Lua)
- Arduino (C++)
- Sming (C++)

Programmazione dei microcontrollori







INTERPRETER **FIRMWARE** **OPERATION SYSTEM**

Programmazione dei microcontrollori

Esempio

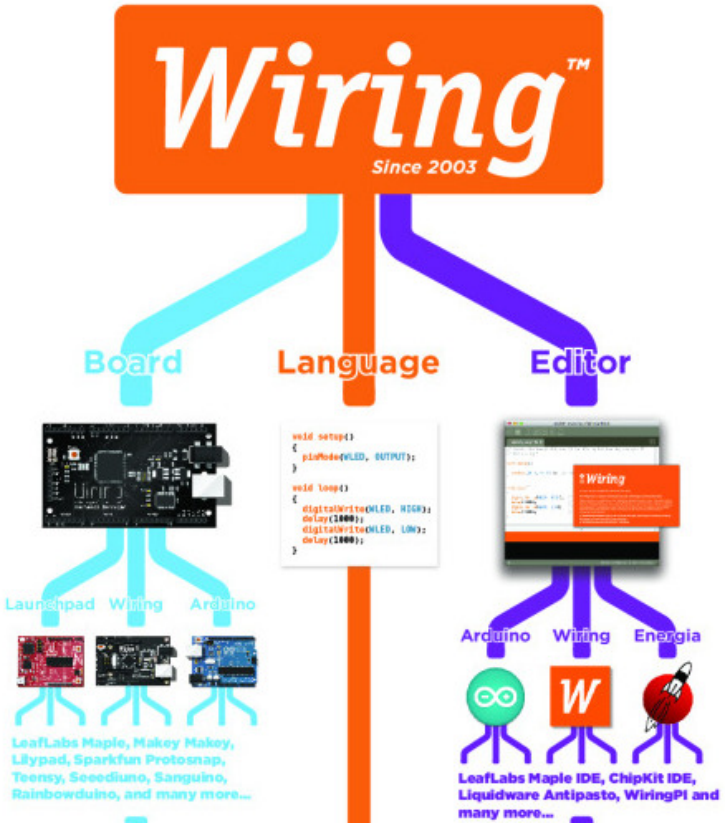
```
1  #include "arduino.h"
2
3  void setup() {
4      DDRD = DDRD | B00100000;
5  }
6  void loop() {
7      PORTD = PORTD | B00100000;
8      delay(1000);
9      PORTD = PORTD & B11011111;
10     delay(1000);
11 }
12
```

```
1  #include "arduino.h"
2
3  void setup() {
4      DDRB |= _BV(DDB5);
5  }
6
7  void loop() {
8      PORTB |= _BV(PORTB5);
9      _delay_ms(1000);
10     PORTB &= ~_BV(PORTB5);
11     _delay_ms(1000);
12 }
```

Wiring

Arduino, WiringPi, Sming basati su Wiring

- Open Source
- Open Hardware
- Linguaggio o framework accessibile
- IDE semplice
- ...



Wiring

Struttura del programma

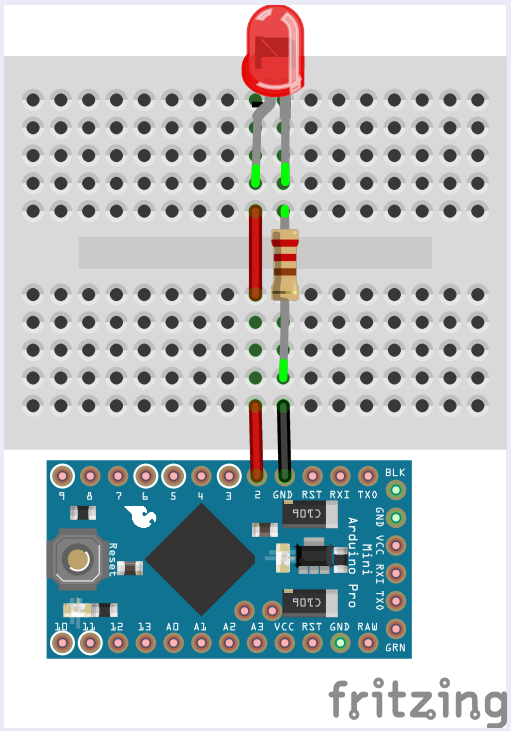
File Edit Sketch Tools Help

Blink

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

Done compiling.

```
Binary sketch size: 3,828 bytes (of a 28,672 byte maximum)
```



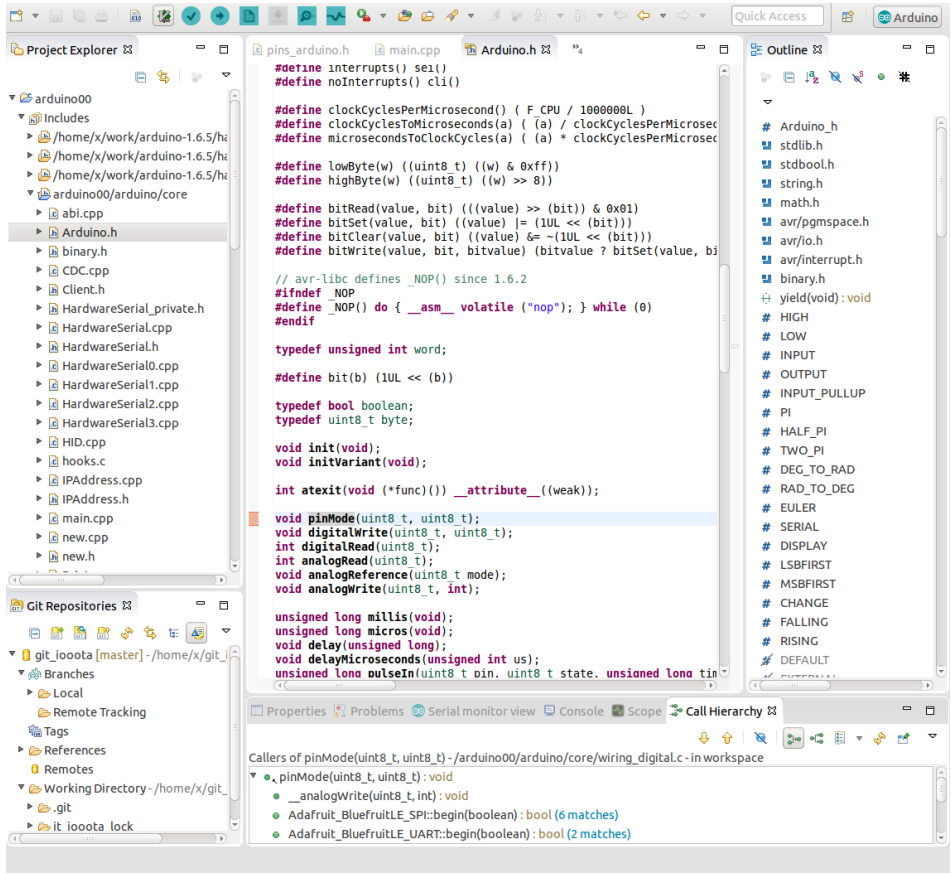
IDE per sviluppo

Eclipse plugin AVR-eclipse

completely free and open Arduino IDE alternative

PlatformIO

cross platform code builder and library manager



Arduino Button

polling

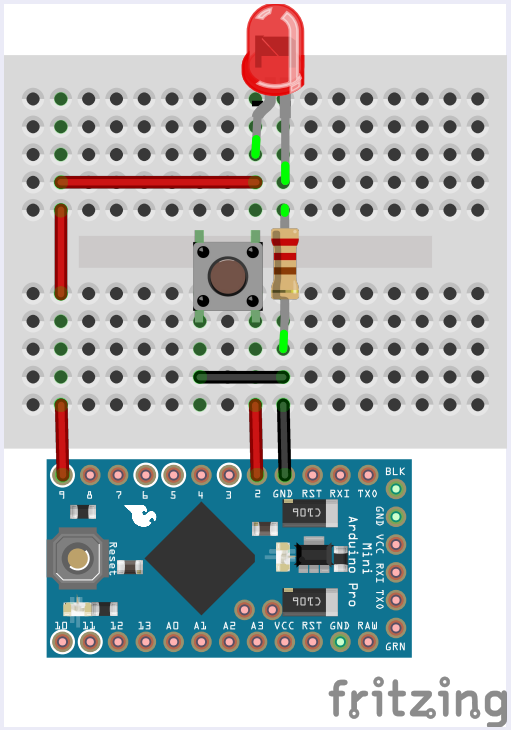
File Edit Sketch Tools Help

sketch_may15b §

```
void setup(){
  Serial.begin(9600);
  pinMode(2, INPUT_PULLUP);
  pinMode(13, OUTPUT);
}

void loop(){
  int sensorVal = digitalRead(2);
  Serial.println(sensorVal);
  if (sensorVal == HIGH) {
    digitalWrite(13, LOW);
  }
  else {
    digitalWrite(13, HIGH);
  }
}
```

Done compiling.

```
Binary sketch size: 4,476 bytes (of a 28,672
byte maximum)
```



Arduino Button

interrupt

File Edit Sketch Tools Help

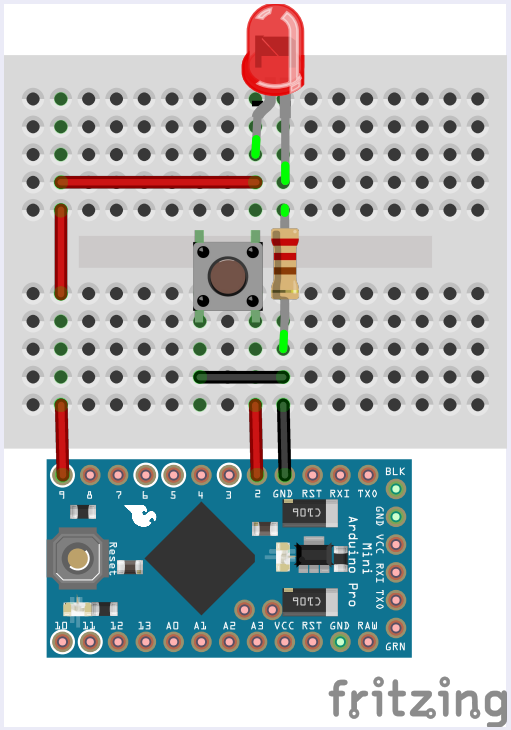
sketch_may15a §

```
int pin = 13;
volatile int state = LOW;

void setup(){
  pinMode(pin, OUTPUT);
  attachInterrupt(0, blink, CHANGE);
}

void loop(){
  digitalWrite(pin, state);
}

void blink(){
  state = !state;
}
```

Done compiling.

```
Binary sketch size: 4,446 bytes (of a 28,672
byte maximum)
```

1 SparkFun Pro Micro 5V/16MHz on /dev/ttyACM0



Raspberry Pi

Single Board Computer

- CPU ARM11 700Mhz
- 512 RAM
- 2 USB
- Ethernet port
- HDMI
- 3.5 mm audio
- GPIO
  - **Digital I/O**
  - PWM pin
  - Serial
  - I2C

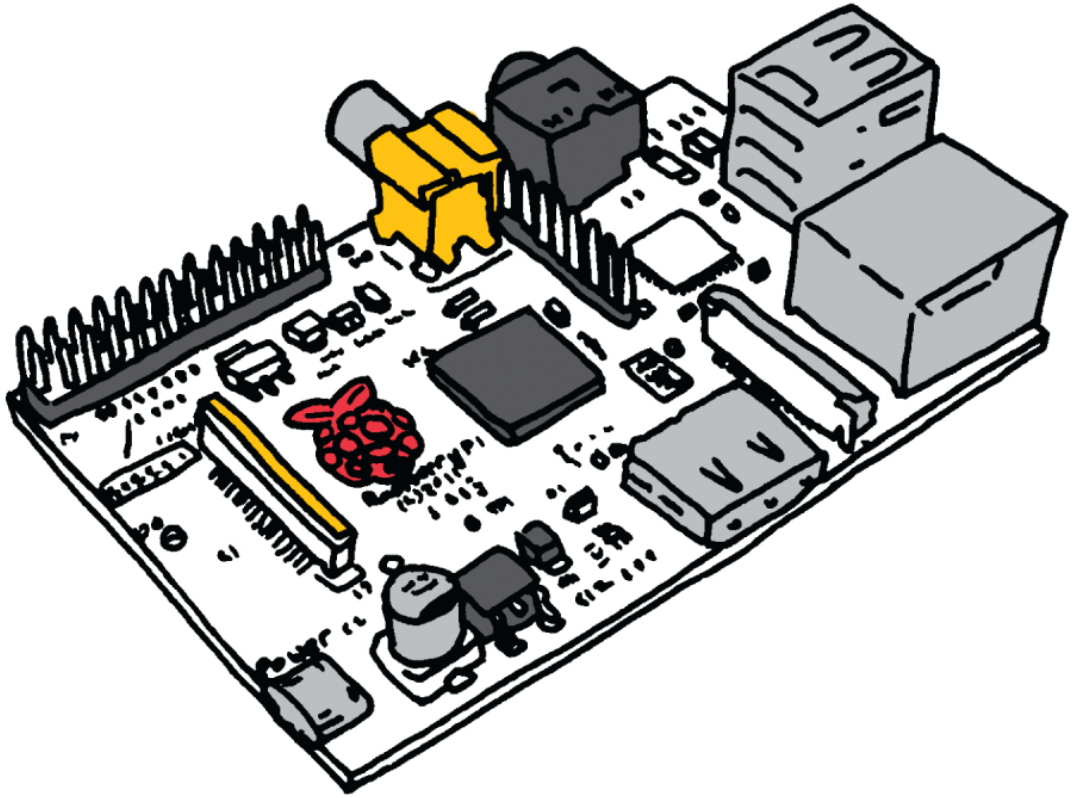


Figura : RASPBERRY PI MODEL B

Raspberry Pi

Software

Sistemi operativi

- Raspbian(Debian)
- Pidora (Fedora Remix)
- Arch Linux
- RISC OS
- E molti altri sistemi

E' possibile installare Android, anche se non e' supportato ufficialmente.

Linguaggi di programmazione

- Python
- Java
- Bash
- C / C++
- Ruby
- Perl
- ...

Raspbian

Versione custom per il corso

Software Aggiuntivo

- Server DHCP
- Tool e librerie per la progettazione
  - wiringPI
  - RaspiCam
  - Java
  - pi4j
  - vim
- Server VNC accessibile al 192.168.137.2:1
- Trasferimento del file tramite protocollo SMB
- Configurazione wifi facilitata
- `https://137.204.107.21/www-files/2016-02-26-raspbian-jessie-iss.img.zip`

RASPBERRY PI PINOUT

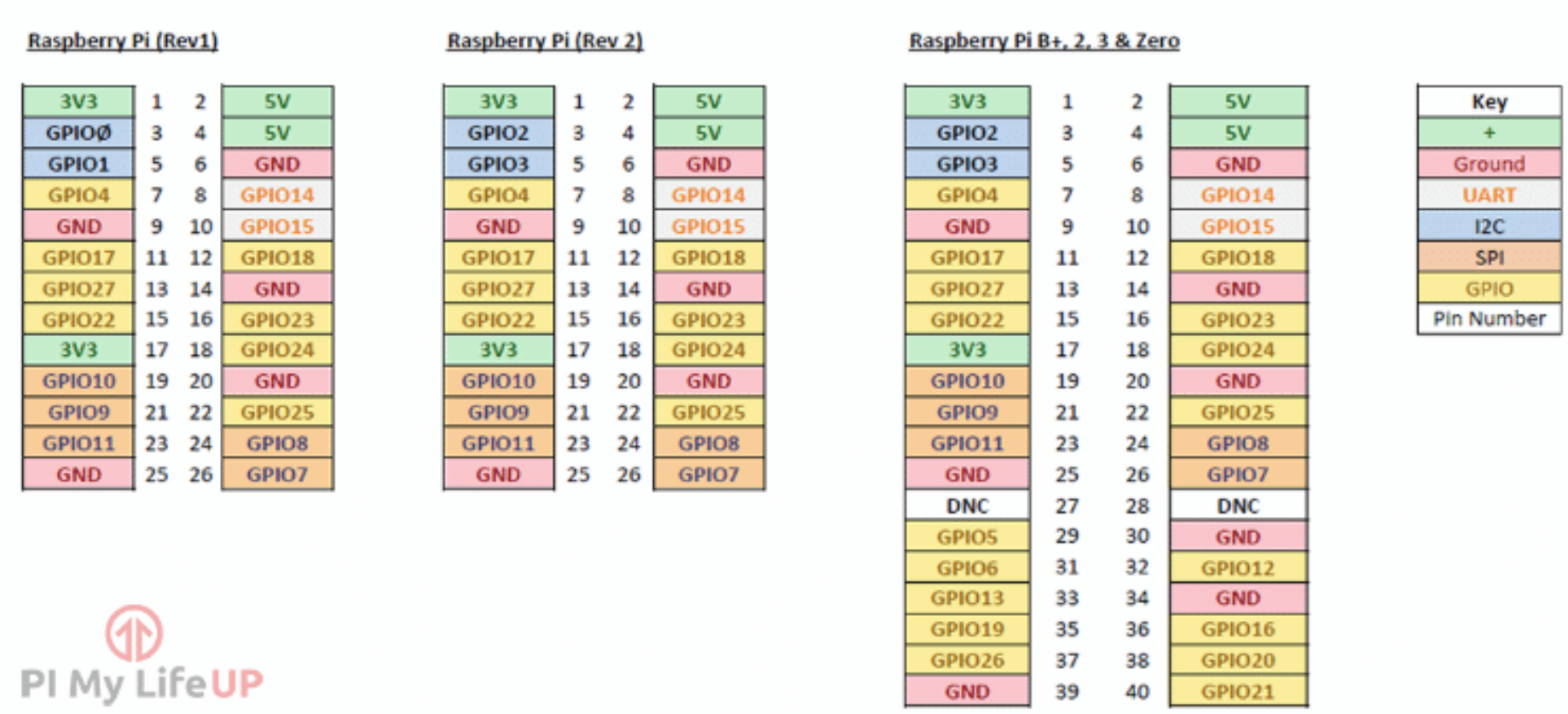


Figura : RASPBERRY PI PINOUT

Programmazione GPIO

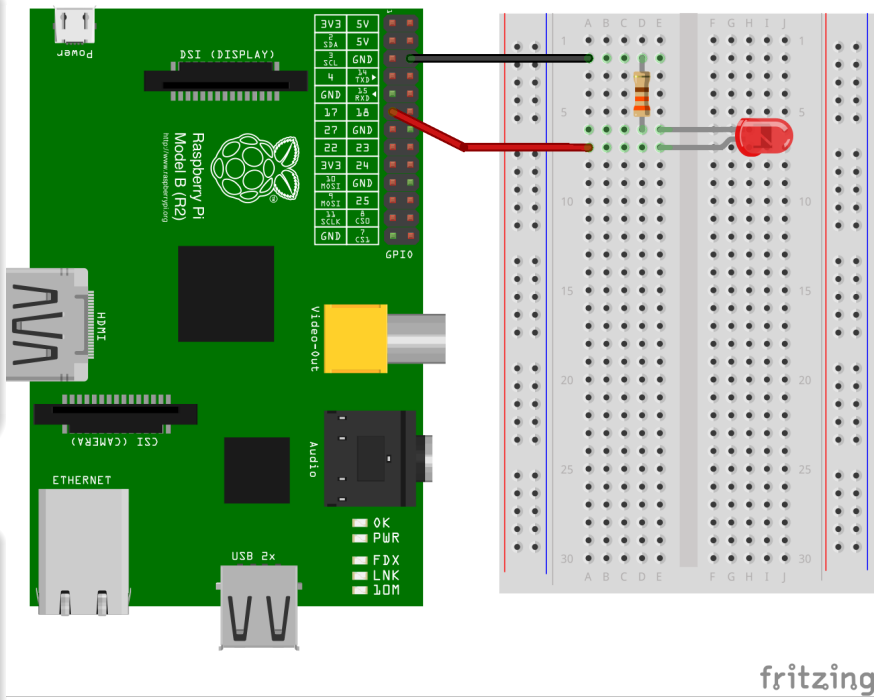
E' possibile gestire GPIO tramite un filesystem virtuale.

- "/sys/class/gpio" è un'interfaccia di controllo in user space
  - "export" permette di esportare il controllo di GPIO in user space.(*echo 19 > export*)
  - "unexport" rimuove la risorsa da user space (*echo 19 > unexport*)
- /sys/class/gpio/gpioN/ è un'interfaccia di controllo di un singolo GPIO
  - "direction" imposta GPIO come Input o Output (*echo in > direction*)
  - "value" legge il valore di GPIO (*cat value*)
  - ...

Raspberry Pi GPIO Output

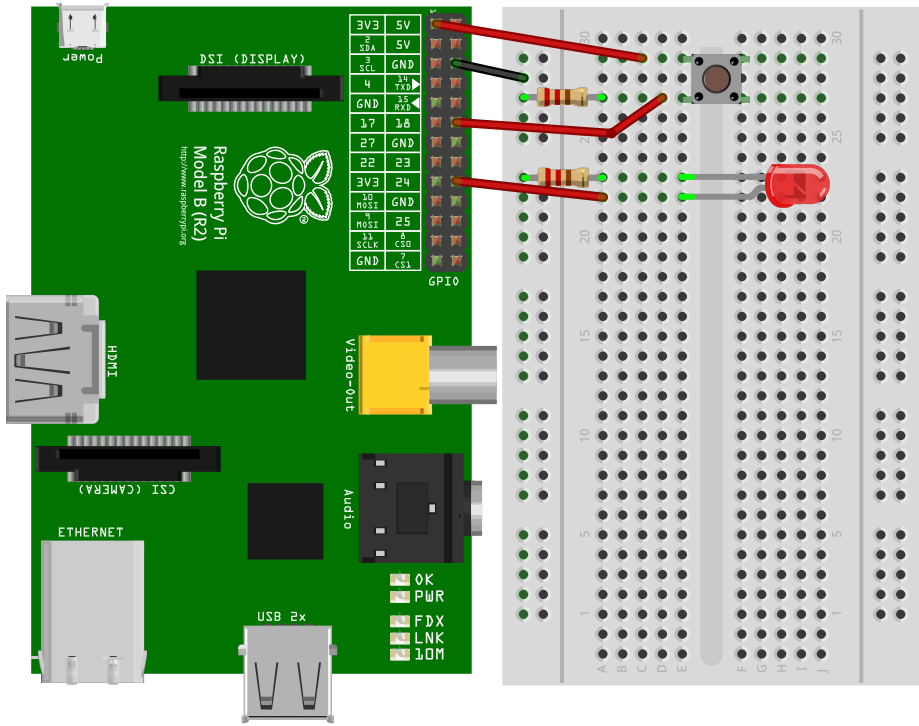
```
#!/bin/bash
echo 17 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio17/direction

while :
do
 echo 0 > /sys/class/gpio/gpio17/value
 sleep 0.2
 echo 1 > /sys/class/gpio/gpio17/value
 sleep 0.2
done
```

```
#!/bin/bash
gpio mode 0 out
gpio write 0 1
sleep 1
gpio write 0 0
```



Raspberry Pi

```
#!/bin/bash
echo 24 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio24/direction
echo 18 > /sys/class/gpio/export
echo in > /sys/class/gpio/gpio18/direction

while :
do
 cat /sys/class/gpio/gpio24/value >\
 /sys/class/gpio/gpio24/value
 sleep 0.2
done
```



Differenza tra Arduino e Raspberry Pi

	Arduino	Raspberry Pi
Processor	ATMega 328	ARM11
Clock Speed	16 MHz	700 MHz
Flash	32 KB	SD
RAM	2 KB	512 MB
I/O Current Max	40 mA	10 mA
Min Power	42 mA	700 mA
Register Width	8-bit	32-bit
Operating System	None	Linux and Other

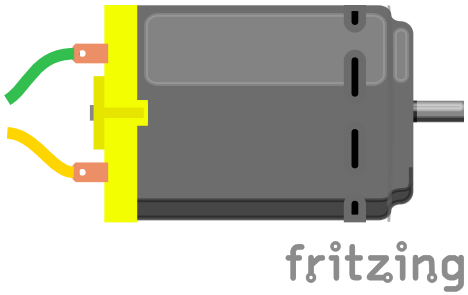
Tabella : Arduino Uno vs Raspberry

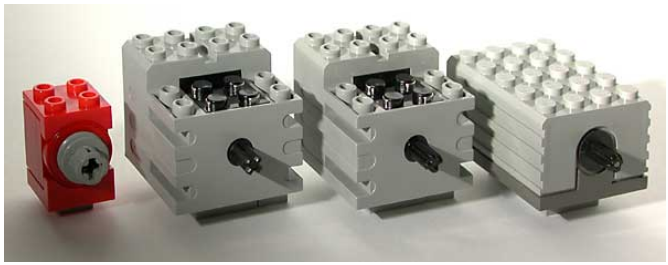
Microcontroller vs Microprocessor

DC Motor

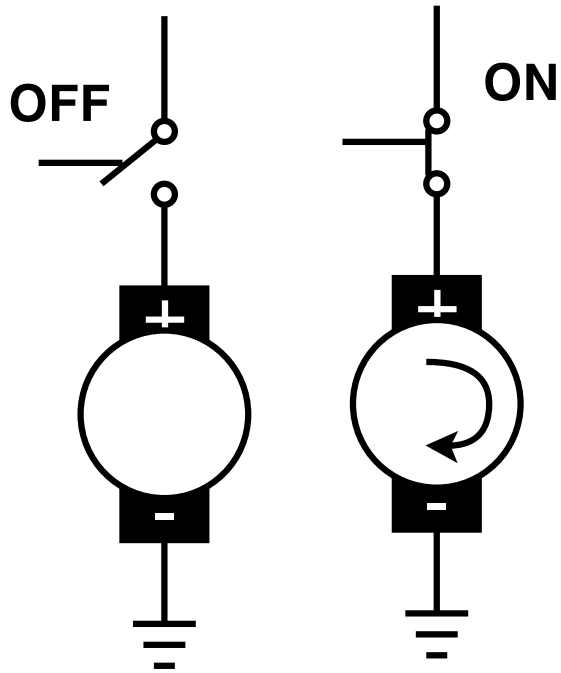
Controllo ON-OFF di motori DC

- DC Motors





Lego RCX DC Motors



Quando l'interruttore è aperto il motore è fermo, quando è chiuso gira alla massima velocità

Attuatori

HBridge

- H Bridge (Motor Driver)

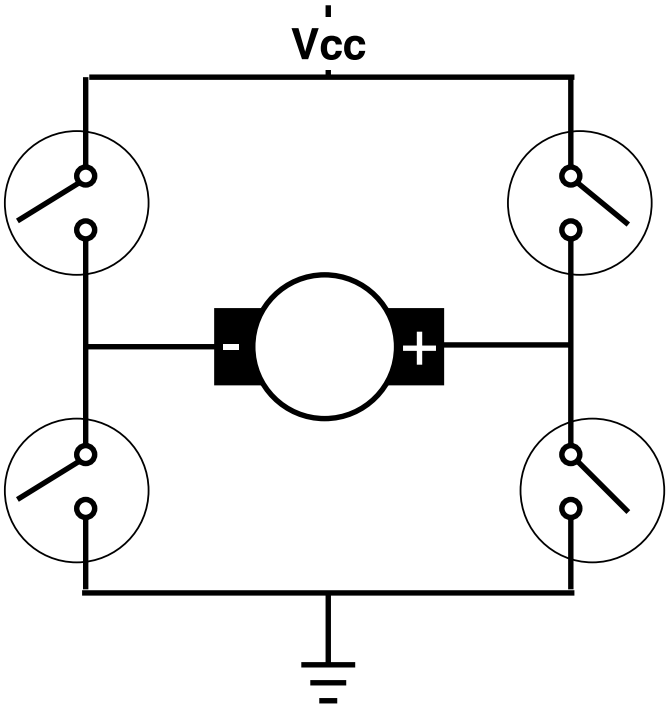
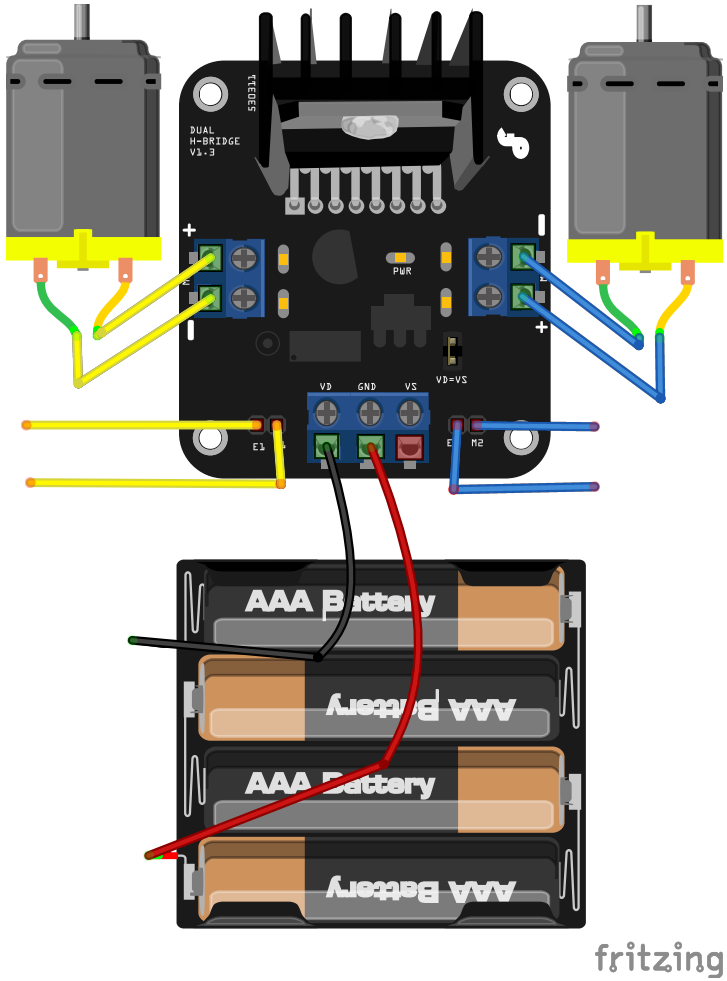


Figura : schema semplificato di un ponte H



Controllo di velocità

PWM Pulse Width Modulation

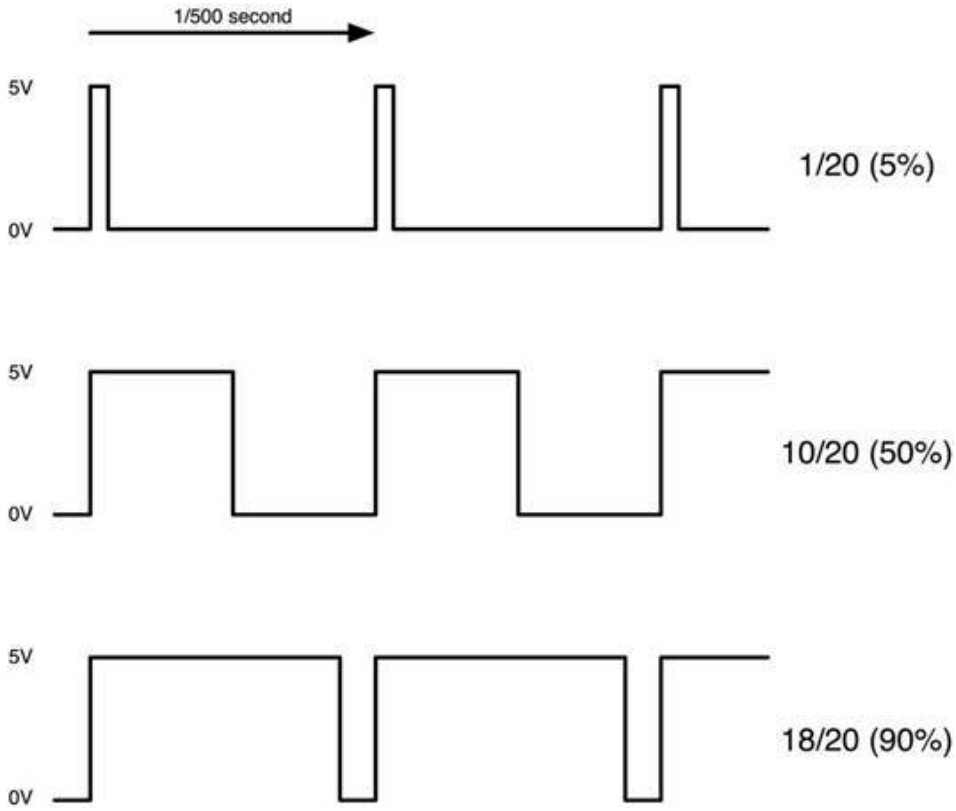


Figura : modulazione di larghezza di impulso

Arduino UNO 5 pin predisposti per PWM

Raspberry Pi 1/2 pin supportano Hardware PWM.

Direct PWM control

Input		Output
FIN	RIN	
PWM	L	Forward
L	PWM	Reverse
H	H	Brake
L	L	Idle

Servomotore

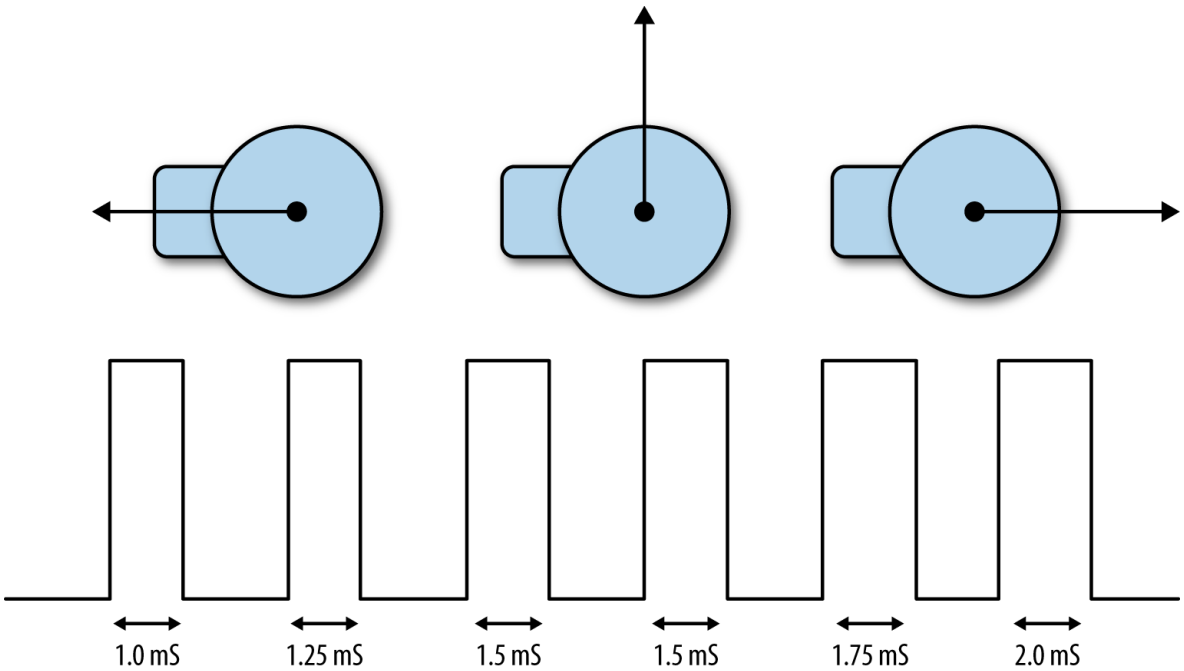
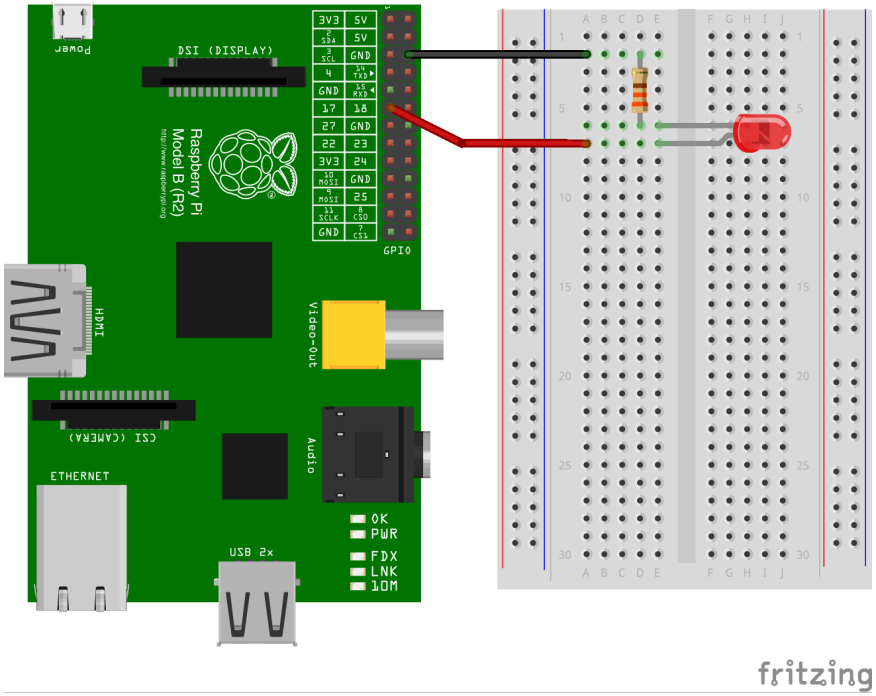


Figura : controllo del servomotore con PWM

I servomotori sono attuatori muniti di un sistema di retroazione che permette di controllarne la posizione angolare attraverso il segnale di ingresso.

Software PWM

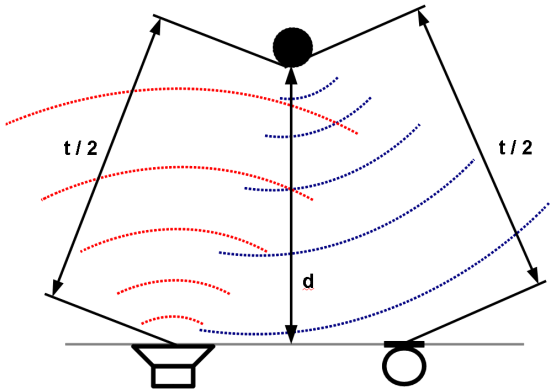
Esempio




```
35      com.pi4j.wiringpi.Gpio.wiringPiSetup();
36
37      // create soft-pwm pins (min=0 ; max=100)
38      SoftPwm.softPwmCreate(1, 0, 100);
39
40      // continuous loop
41      while (true) {
42          // fade LED to fully ON
43          for (int i = 0; i <= 100; i++) {
44              SoftPwm.softPwmWrite(1, i);
45              Thread.sleep(100);
46          }
47
48          // fade LED to fully OFF
49          for (int i = 100; i >= 0; i--) {
50              SoftPwm.softPwmWrite(1, i);
51              Thread.sleep(100);
52          }
53      }
```

Ultrasonic Sensor

Principio di funzionamento



Working Voltage 5V

Output Signal 0-5V

Sentry Angle max 15 degree

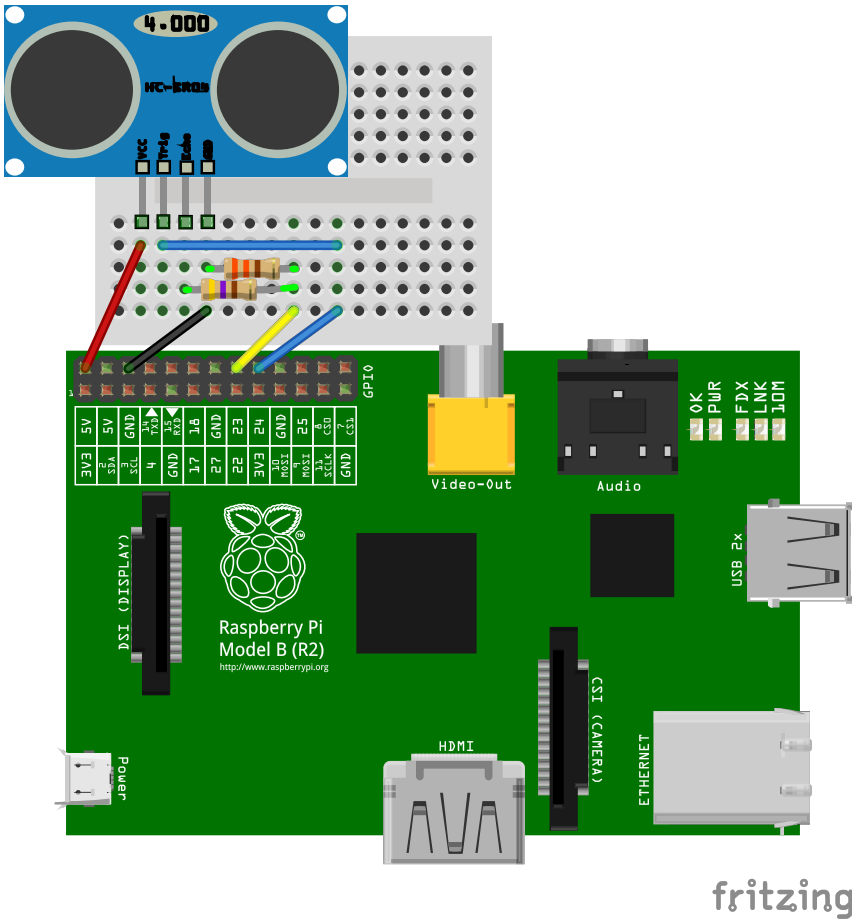
Working Current 15 mA

Working frequency 40Hz

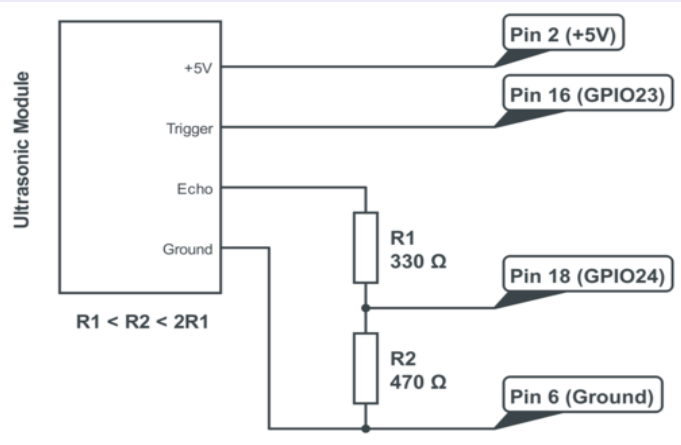
$$\textbf{distance} = \frac{\textbf{speed of sound} \times \textbf{time taken}}{2}$$

Ultrasonic sensor

Utilizzo con Raspberry Pi



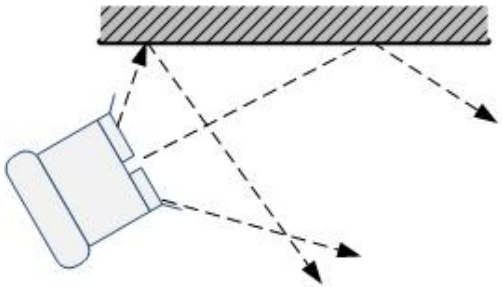
Partitore di tensione



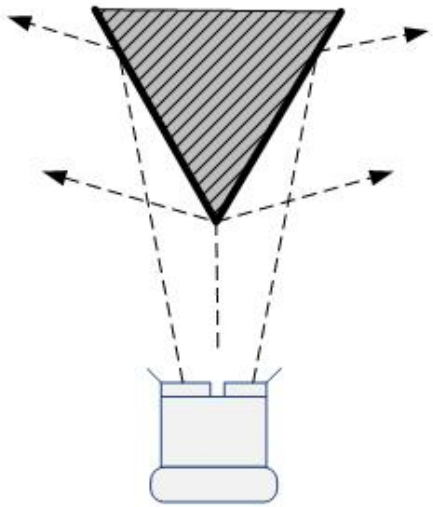

```
21 //Send trig pulse
22 digitalWrite(TRIG, HIGH);
23 delayMicroseconds(20);
24 digitalWrite(TRIG, LOW);
25 //Wait for echo start
26 while(digitalRead(ECHO) == LOW);
27 //Wait for echo end
28 long startTime = micros();
29 while(digitalRead(ECHO) == HIGH);
30 long travelTime = micros() - startTime;
31 //Get distance in cm
32 int distance = travelTime / 58;
33 return distance;
```

Ultrasonic sensor

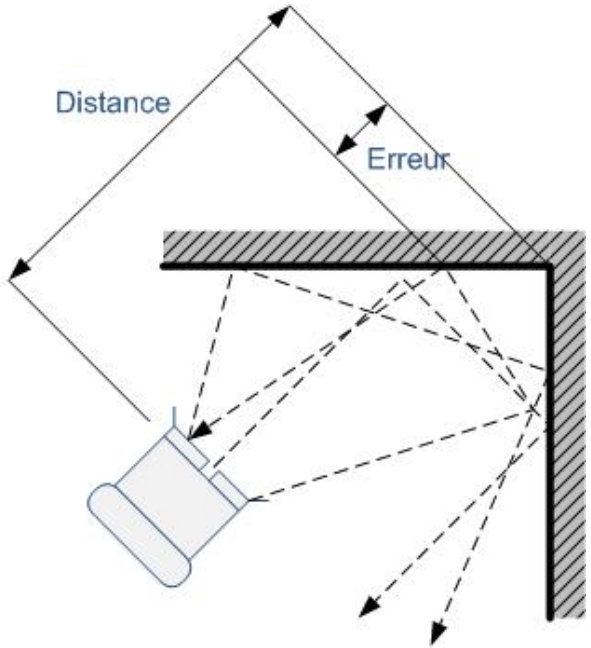
Problemi e limiti



Cas n°4

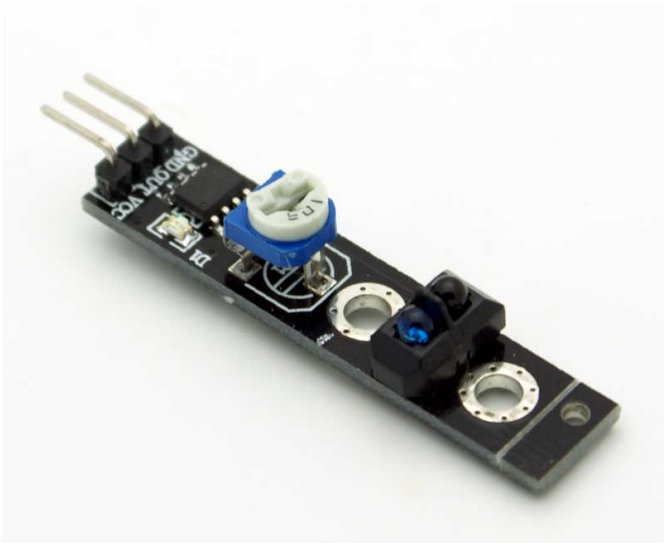


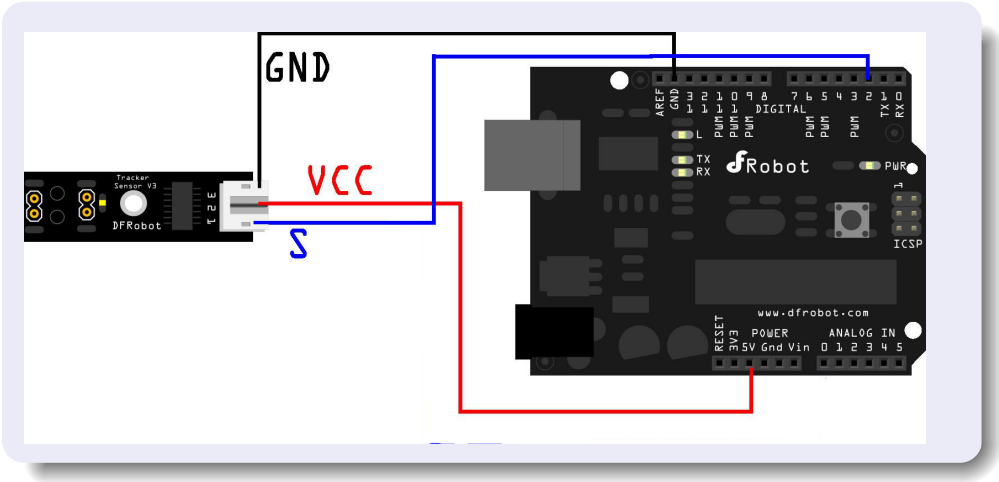
Cas n°5



Cas n°6

Line Sensor





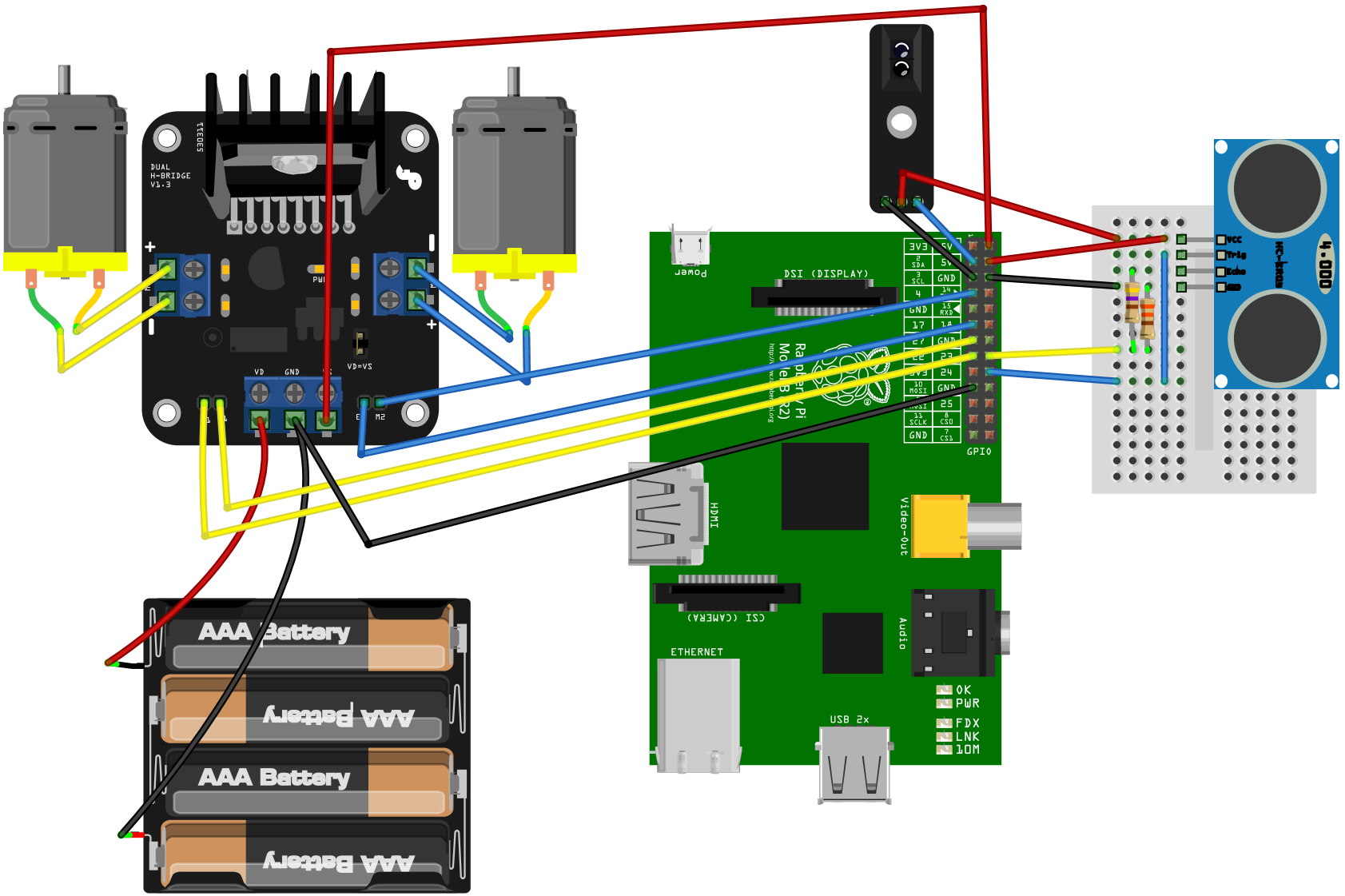
Power supply	+5V
Operating current	10mA

3-wire interface

1. signal
2. power
3. power supply negative

```
///Arduino Sample Code
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  Serial.println(digitalRead(2)); // print the data from the sensor
  delay(500);
}
```

Robot

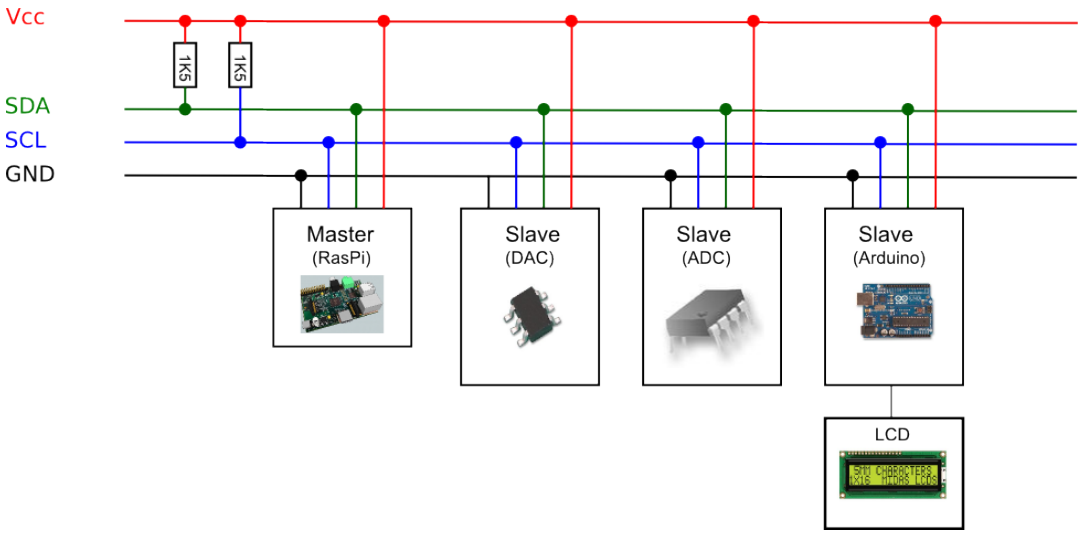


fritzing

I2C Bus

Inter Integrated Circuit

- Due linee seriali di comunicazione
  - SDA (Serial DAta) per i dati
  - SCL SCL (Serial CLock) per il clock
- Ogni dispositivo é identificato dall'indirizzo univoco, e puó trasmettere o ricevere
- Master - dispositivo che emette il segnale di clock
- Slave - nodo che si sincronizza sul segnale di clock



Bussola digitale

HMC5883l

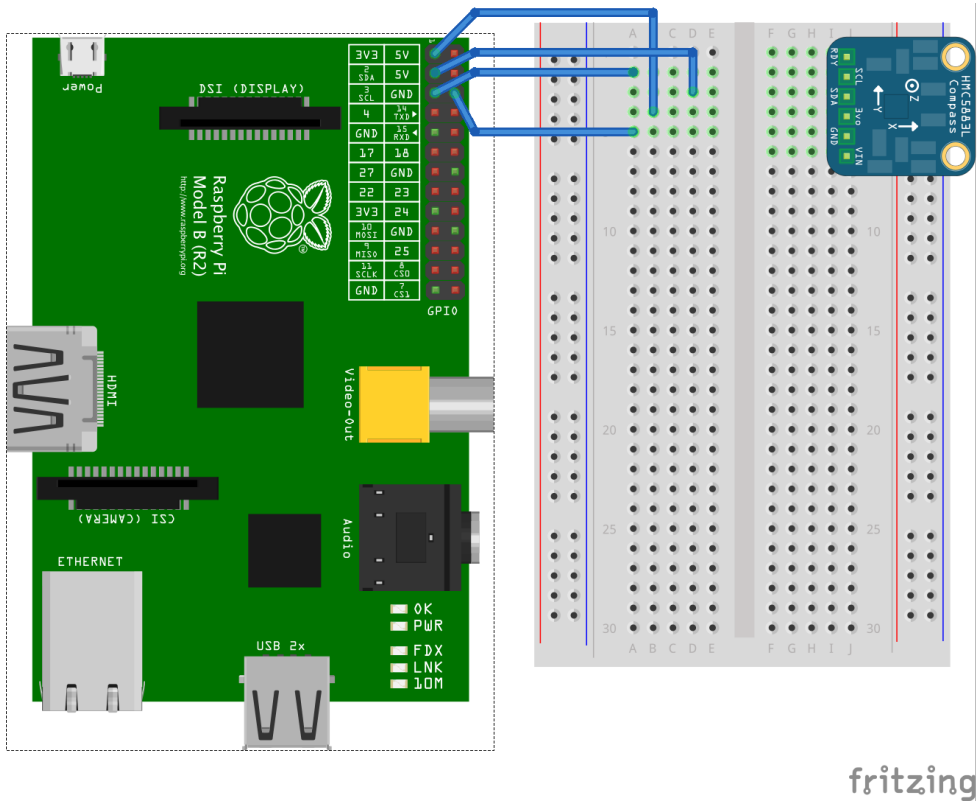
- Leggere i dati grezzi

```
int r = i2cDevice.read(address, data, 0, 6);

x=((data[0]&0xff)«8) + (data[1]&0xff);

y=((data[2]&0xff)«8) + (data[3]&0xff);

z=((data[4]&0xff)«8) + (data[5]&0xff);
```

- Applicare le correzioni
- Calcolare la direzione

```
Math.toDegrees(Math.atan2(compassx, compassy)
```

```
pi@raspberrypi ~ $ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- 1e --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- 53 -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- 77
pi@raspberrypi ~ $
```



Raspberry Pi Camera

http://www.raspberrypi.org/documentation/usage/camera/README.md

Still resolution	5 Megapixel
Video modes	1080p30, 720p60 and 640x480p60 90
Picture formats	JPEG (accelerated) , JPEG + RAW , GIF , BMP , PNG , YUV420 , RGB888
Video formats	raw h.264 (accelerated)

```
#acquisire immagine i salvare come imge.jpg
raspistill -o image.jpg
#salvare immagine come png (-e png)
raspistill -e png -o image.png
#cambiare la risoluzione
raspistill -w 640 -h 360 -o image.jpg
#acquisire video di 5 secondi
raspivid -t 5000 -o video.h264
#video con bitrate 3.5MBits/s
raspivid -t 5000 -o video.h264 -b 3500000
#cambiare la risoluzione del video
raspivid -w 640 -h 360 -o video.h264
```

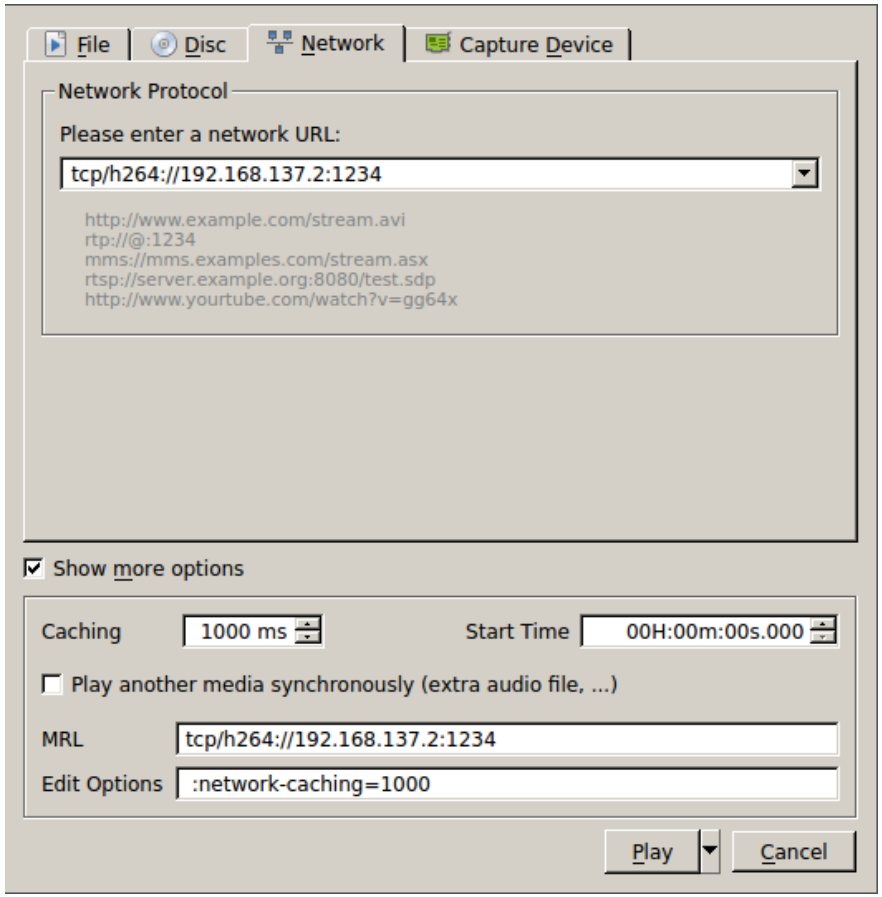
Raspberry Camera

Streaming video

- Raspberry Pi command

```
nc.traditional -l -p 1234 -c "exec raspivid -t 600000 -n -w 640 -h 360 -o -"
```

- Client side VLC setup



Link utili

Streaming video

Arduino Eclipse IDE	Eclipse with WinAVR and the Eclipse plugin AVR-eclipse http://eclipse.baeyens.it/
Platform IO	open source ecosystem for IoT development http://platformio.org/platformio-ide
NodeMCU	A lua based firmware for wifi-soc esp8266 https://github.com/nodemcu/nodemcu-firmware
Sming	framework for ESP8266 native development with C++ https://github.com/SmingHub/Sming
Wiring	http://wiring.org.co/
Wiring Pi	GPIO Interface library for the Raspberry Pi http://wiringpi.com/
The Untold History of Arduino	https://arduinohistory.github.io/
Raspberry pi custom distribution	http://www.raspberrypi.org/documentation/usage/camera/README.md