LINUX DAY 2016

LINUX
D A Y
I T A L I A

Programmare Raspberry Pi

LINUX DAY 2016

Il talk si divide in due parti:

- Prima Parte descrittiva
- Seconda Parte Laboratorio

LINUX DAY 2016

Prima Parte

Introduzione su Raspberry Pi
e le porte GPIO

LINUX DAY 2016

Perché ?

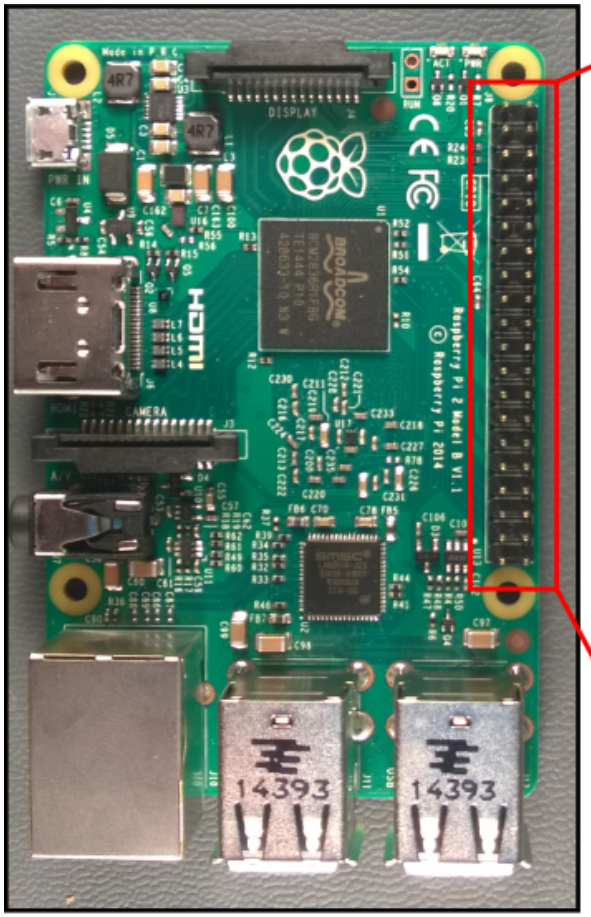
LINUX DAY 2016

Ma cosa posso fare ?

LINUX DAY 2016

LINUX DAY ITALIA

Il connettore GPIO



3.3V PWR	1		2	5V PWR
I2C1 SDA	3		4	5V PWR
I2C1 SCL	5		6	GND
GPIO 4	7		8	Reserved
GND	9		10	Reserved
SPI1 CS0	11		12	GPIO 18
GPIO 27	13		14	GND
GPIO 22	15		16	GPIO 23
3.3V PWR	17		18	GPIO 24
SPI0 MOSI	19		20	GND
SPI0 MISO	21		22	GPIO 25
SPI0 SCLK	23		24	SPI0 CS0
GND	25		26	SPI0 CS1
Reserved	27		28	Reserved
GPIO 5	29		30	GND
GPIO 6	31		32	GPIO 12
GPIO 13	33		34	GND
SPI1 MISO	35		36	GPIO 16
GPIO 26	37		38	SPI1 MOSI
GND	39		40	SPI1 SCLK

LINUX DAY 2016

La retro compatibilità

wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin
–	–	3.3v	1 | 2	5v	–	–
8	R1:0/R2:2	SDA0	3 | 4	5v	–	–
9	R1:1/R2:3	SCL0	5 | 6	0V	–	–
7	4	GPIO7	7 | 8	TXD	14	15
–	–	0V	9 | 10	RXD	15	16
0	17	GPIO0	11 | 12	GPIO1	18	1
2	R1:21/R2:27	GPIO2	13 | 14	0V	–	–
3	22	GPIO3	15 | 16	GPIO4	23	4
–	–	3.3v	17 | 18	GPIO5	24	5
12	10	MOSI	19 | 20	0V	–	–
13	9	MISO	21 | 22	GPIO6	25	6
14	11	SCLK	23 | 24	CE0	8	10
–	–	0V	25 | 26	CE1	7	11
30	0	SDA. 0	27 | 28	SCL. 0	1	31
21	5	GPIO. 21	29 | 30	0V	–	–
22	6	GPIO. 22	31 | 32	GPIO. 26	12	26
23	13	GPIO. 23	33 | 34	0V	–	–
24	19	GPIO. 24	35 | 36	GPIO. 27	16	27
25	26	GPIO. 25	37 | 38	GPIO. 28	20	28
		0V	39 | 40	GPIO. 29	21	29
wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin

For RPi B

For RPi B+/2 model B

Prima di cominciare…

...carichiamo la funzionalità wiringPi altrimenti non facciamo nulla.

Scarichiamo il codice

```
# git clone git://git.dragon.net/wiringPi
```

Installaimo wiringPi

```
# cd wiringPi
# git pull origin
# ./build
```

LINUX DAY 2016

Verifichiamo che sia andato tutto bene…

```
root@raspberrypi:/home/pi# gpio -v
gpio version: 2.32
Copyright (c) 2012-2015 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
  Type: Pi 2, Revision: 01, Memory: 1024MB, Maker: Sony
  * Device tree is enabled.
  * This Raspberry Pi supports user-level GPIO access.
    -> See the man-page for more details
    -> ie. export WIRINGPI_GPIOMEM=1
root@raspberrypi:/home/pi#
```

LINUX DAY 2016

...e che le porte GPIO siano operative

```
root@raspberrypi:/home/pi# gpio readall
 +-----+-----+---------+------+---+---Pi 2---+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 |     |     |    3.3v |      |   |  1 || 2  |   |      | 5v      |     |     |
 |   2 |   8 |   SDA.1 |   IN | 1 |  3 || 4  |   |      | 5V      |     |     |
 |   3 |   9 |   SCL.1 |   IN | 1 |  5 || 6  |   |      | 0v      |     |     |
 |   4 |   7 | GPIO. 7 |   IN | 1 |  7 || 8  | 1 | ALT0 | TxD     | 15  | 14  |
 |     |     |      0v |      |   |  9 || 10 | 1 | ALT0 | RxD     | 16  | 15  |
 |  17 |   0 | GPIO. 0 |  OUT | 1 | 11 || 12 | 0 | OUT  | GPIO. 1 | 1   | 18  |
 |  27 |   2 | GPIO. 2 |  OUT | 0 | 13 || 14 |   |      | 0v      |     |     |
 |  22 |   3 | GPIO. 3 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO. 4 | 4   | 23  |
 |     |     |    3.3v |      |   | 17 || 18 | 0 | IN   | GPIO. 5 | 5   | 24  |
 |  10 |  12 |    MOSI |   IN | 0 | 19 || 20 |   |      | 0v      |     |     |
 |   9 |  13 |    MISO |   IN | 0 | 21 || 22 | 0 | IN   | GPIO. 6 | 6   | 25  |
 |  11 |  14 |    SCLK |   IN | 0 | 23 || 24 | 1 | IN   | CE0     | 10  | 8   |
 |     |     |      0v |      |   | 25 || 26 | 1 | IN   | CE1     | 11  | 7   |
 |   0 |  30 |   SDA.0 |   IN | 1 | 27 || 28 | 1 | IN   | SCL.0   | 31  | 1   |
 |   5 |  21 | GPIO.21 |   IN | 1 | 29 || 30 |   |      | 0v      |     |     |
 |   6 |  22 | GPIO.22 |   IN | 1 | 31 || 32 | 0 | IN   | GPIO.26 | 26  | 12  |
 |  13 |  23 | GPIO.23 |   IN | 0 | 33 || 34 |   |      | 0v      |     |     |
 |  19 |  24 | GPIO.24 |   IN | 0 | 35 || 36 | 0 | IN   | GPIO.27 | 27  | 16  |
 |  26 |  25 | GPIO.25 |   IN | 0 | 37 || 38 | 0 | IN   | GPIO.28 | 28  | 20  |
 |     |     |      0v |      |   | 39 || 40 | 0 | IN   | GPIO.29 | 29  | 21  |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+---Pi 2---+---+------+---------+-----+-----+
```

LINUX DAY 2016

Ora siamo operativi e pronti

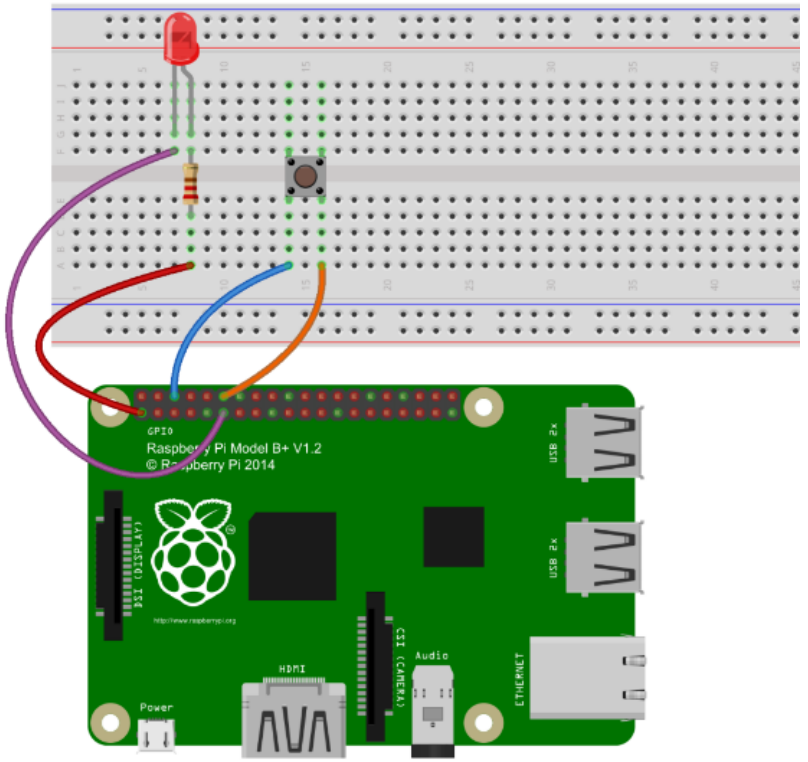
LINUX DAY 2016

Seconda Parte

Laboratorio con tre esempi pratici

LINUX DAY 2016

1# LED comandato da pulsante



LINUX DAY 2016

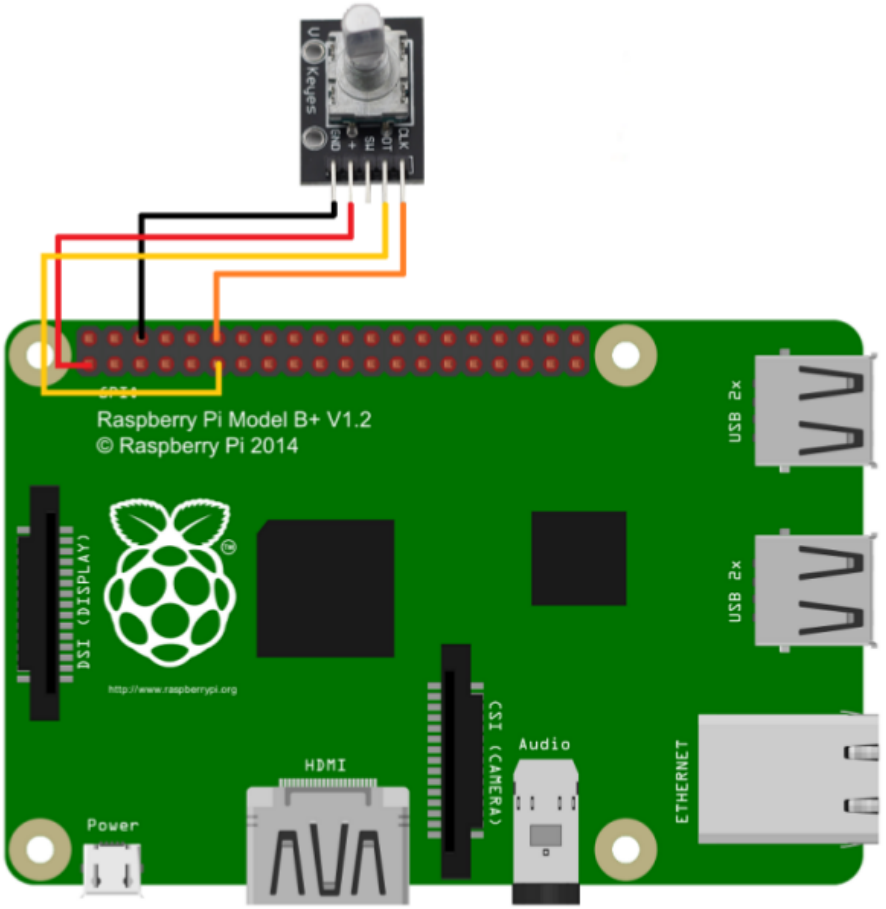
1# LED comandato da pulsante

```
#include <wiringPi.h>
#include <stdio.h>

#define LedPin    0
#define ButtonPin 1

int main(void)
{
        if(wiringPiSetup() == -1){ //when initialize wiring failed,print message to screen
                printf("setup wiringPi failed !");
                return 1;
        }

        pinMode(LedPin, OUTPUT);
        pinMode(ButtonPin, INPUT);

        pullUpDnControl(ButtonPin, PUD_UP);  //pull up to 3.3V,make GPIO1 a stable level
        while(1){
                digitalWrite(LedPin, HIGH);
                if(digitalRead(ButtonPin) == 0){ //indicate that button has pressed down
                        digitalWrite(LedPin, LOW);   //led on
                }
        }

        return 0;
}
```

LINUX DAY 2016

LINUX
D A Y
I T A L I A

2# Encoder Rotativo



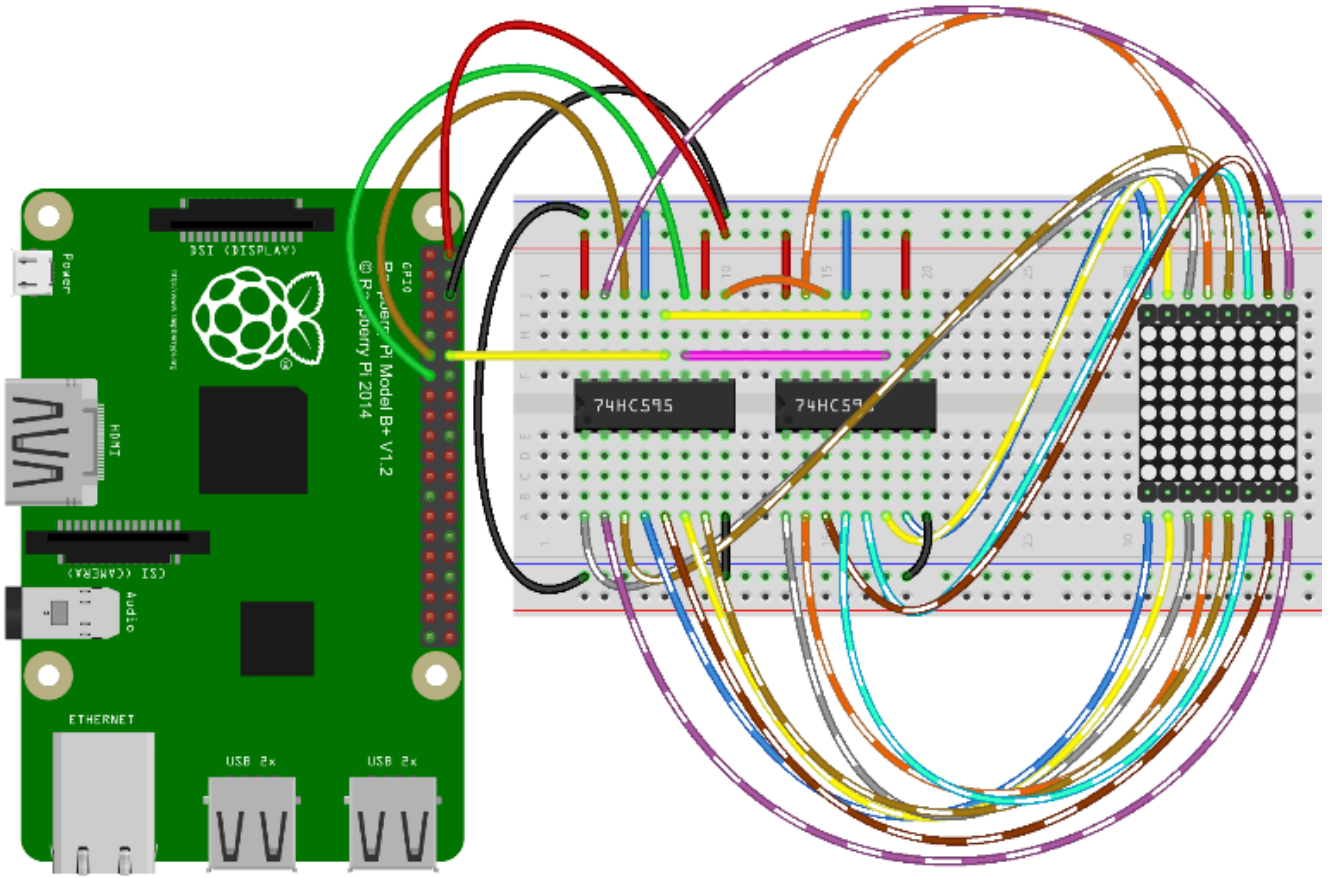
2# Encoder Rotativo

```
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <stdlib.h>
#include <wiringPi.h>

#define  RoAPin    0
#define  RoBPin    1

static volatile int globalCounter = 0 ;

unsigned char flag;
unsigned char Last_RoB_Status;
unsigned char Current_RoB_Status;

void rotaryDeal(void)
{
        Last_RoB_Status = digitalRead(RoBPin);

        while(!digitalRead(RoAPin)){
                Current_RoB_Status = digitalRead(RoBPin);
                flag = 1;
        }
…..
```

LINUX DAY 2016

LINUX DAY ITALIA

3# Matrice a LED



LINUX DAY 2016

3# Matrice a LED

```
#include <wiringPi.h>
#include <stdio.h>

#define   SDI   0   //serial data input
#define   RCLK  1   //memory clock input(STCP)
#define   SRCLK 2   //shift register clock input(SHCP)

unsigned char code_H[20] =
{0x01,0xff,0x80,0xff,0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff};
unsigned char code_L[20] =
{0x00,0x7f,0x00,0xfe,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0x7f};

//unsigned char code_L[8] = {0x00,0x00,0x3c,0x42,0x42,0x3c,0x00,0x00};
//unsigned char code_H[8] = {0xff,0xe7,0xdb,0xdb,0xdb,0xdb,0xe7,0xff};

//unsigned char code_L[8] = {0xff,0xff,0xc3,0xbd,0xbd,0xc3,0xff,0xff};
//unsigned char code_H[8] = {0x00,0x18,0x24,0x24,0x24,0x24,0x18,0x00};

void init(void)
{
        pinMode(SDI, OUTPUT); //make P0 output
        pinMode(RCLK, OUTPUT); //make P0 output
        pinMode(SRCLK, OUTPUT); //make P0 output

        digitalWrite(SDI, 0);
        digitalWrite(RCLK, 0);
        digitalWrite(SRCLK, 0);
}
…..
```

LINUX DAY 2016

Grazie

damiano.conte@gmail.com