

Con Q8 si può pagare il rifornimento direttamente dallo smartphone

Ormai lo smartphone è diventato un oggetto con cui fare tantissime cose; mandare messaggi, scattare foto, fare ricerche sul web e tanto altro. Nell'ultimo periodo, sta avendo molto successo anche per quanto riguarda i pagamenti, sfruttando la tecnologia NFC. Tuttavia, per i dispositivi che non dispongono di tale tecnologia, è possibile sfruttare altre risorse, come ad esempio una connessione ad Internet ed un account Paypal.

Quest'ultimo meccanismo viene sfruttato da una nuova applicazione Q8easy CLUB, con la quale è possibile pagare il rifornimento del carburante, direttamente dal nostro smartphone, sfruttando l'account di Paypal.



L'applicazione è davvero semplice da usare, in quanto necessita solamente di una connessione ad Internet e del GPS attivo.



I passi da seguire sono i seguenti:

1. Si scarica l'applicazione e si effettua l'iscrizione a Q8Easy Club
2. Si seleziona Paypal come metodo di pagamento, selezionando l'importo del rifornimento
3. Dopo qualche secondo verrà generato un codice, che ha validità di 10 minuti, che dovrà essere inserito nella macchinetta presente alla stazione di rifornimento
4. Una volta verificata la validità del codice, sarà possibile fare rifornimento

Questa soluzione è davvero comoda e sicura!

Come l'approccio mobile sta cambiando il settore turistico

Come l'approccio mobile sta cambiando il settore turistico

Il mondo mobile e lo sviluppo di applicazioni mobile hanno già cambiato il

modo in cui tutti noi cerchiamo e divoriamo le informazioni a livello globale; i prossimi cambiamenti saranno certo percepiti anche a livello più prettamente locale.

Giusto per fare un esempio, la tecnologia mobile ha completamente cambiato il settore dei **viaggi**, già fortemente “colpito” dall’avvento di internet e del marketing digitale.

Pensiamo solamente a **come oggi prenotiamo voli, alberghi, appartamenti, trasferimenti, visite guidate** o a quanto ghiotti siamo di **recensioni** altrui e di **opinioni** condivise in erte.

Per di più, il mercato del turismo mobile sembra essere in crescita esponenziale.

La tendenza è oramai dilagante: gli utenti pretendono di organizzare e gestire i dettagli mentre sono già in viaggio. Per molti il sogno è quello di poter fare il check-in nell’hotel prenotato a chilometri di distanza, mentre si sta aspettando il volo in aeroporto.

In altre parole, l’umanità o, meglio, una fetta di essa, non ha più pazienza. E questo vale per tutti, non solo per gli uomini d’affari che viaggiano per lavoro e sono, per definizione, impazienti ed esigenti.

Pensiamo invece a una famiglia che viaggia con più figli al seguito: per uno c’è bisogno della culla, per l’altro serve un letto pieghevole, il più grande è allergico al glutine e ha bisogno di un menù speciale.

Riuscite a immaginare che comodità poter **coordinare tutto mentre si è in viaggio**, senza dover aspettare di arrivare all’albergo per accertarsi che tutte le richieste siano state rispettate?

Mamma e papà affronterebbero il viaggio a cuor leggero, consapevoli che al loro arrivo dovranno semplicemente prendere un taxi, arrivare all’hotel e salire nella camera prenotata per lasciare i bagagli.

È proprio qui che entra in gioco il mondo mobile.

I genitori potranno fare tutto comodamente mentre sono in volo: prenotare il trasporto dall’aeroporto all’hotel ed effettuare il check-in assicurandosi che tutte le loro esigenze siano state rispettate.

Nulla di tutto questo, tuttavia, potrebbe avverarsi se:

- 1) mamma e papà non avessero uno smart-phone
- 2) aeroporto e aereo non offrirono il Wi-Fi
- 3) né l’hotel né il servizio taxi disponessero di un App o di un sito mobile

Spesso, però, scaricare un’applicazione non è un gioco da ragazzi, nel senso

che capita ancora, troppe volte, che lo [sviluppo di applicazioni mobile](#) non segua le regole dell'usabilità, obbligando l'utente, per esempio, a creare un nuovo account, senza offrirgli la comodità di effettuare il sign-up con le sue credenziali sociali (Facebook, Twitter o altro).

Nei casi peggiori, il percorso è ancora più ingarbugliato: alcune App, addirittura, richiedono la registrazione al sito prima, poi da lì la possibilità di fare il log-in sulla App. In un periodo storico in cui tutti abbiamo fretta e zero pazienza, "perdere" 5 minuti per registrarsi a un sito per accedere a una applicazione sembra impensabile.

Al contrario, per cancellare un'applicazione, basta un attimo.

È quindi indispensabile che le aziende che sviluppano applicazioni per il mobile prendano sempre più in considerazione l'**importanza dell'usabilità**: il segreto deve essere quello di offrire all'utente un'esperienza piacevole, facendolo arrivare dritto all'obiettivo senza intoppi o digressioni.

Si tratta della stessa evoluzione che ha investito i siti web, la cui "facilità d'uso" o *usability* è divenuta una caratteristica essenziale per garantirne il successo. Tutt'oggi, tuttavia, ancora troppi siti e pagine web sembrano ignorare questa semplice quanto indispensabile regola.

Stiamo a vedere che cosa accadrà al settore delle App per mobile.

Articolo a cura di Cinzia Greco.

[Recensione del nuovo iPhone 6 e 6 plus](#)

Lo scorso Venerdì Apple ha ufficialmente reso disponibile l'acquisto agli utenti Italiani del nuovo melafonino iPhone 6 e 6 Plus.

Lo stesso prodotto era stata già presente l'otto Settembre durante un keynote, nel quale è stato anche mostrato l'iWatch che verrà lanciato sul mercato nei primi mesi del 2015.



I prezzi del nuovo dispositivo sono:

	iPhone 6	iPhone 6 Plus
Capacità e prezzi ¹	•	•
	16GB	16GB
	€ 729	€ 839
	•	•
	64GB	64GB
	€ 839	€ 949
•	•	
128GB	128GB	
€ 949	€ 1.059	

Le migliorie introdotte in questo dispositivo sono diverse; la più evidente riguarda il design completamente rifatto rispetto ai predecessori.

La seconda, in ordine di importanza è la nuova fotocamera, migliorata che permette di scattare foto anche più nitide, oltre che alla possibilità di girare filmati fino a 240 fps.

Potete ammirare questa caratteristica in questo filmato girato con l'iPhone 6

<http://youtu.be/44J0ibZVaDc>

In generale nel dispositivo sono state migliorate le prestazioni, grazie ad un processore A8 davvero veloce.

Dalle prove che ho potuto fare, ho notato che l'iPhone 6 ha un design davvero accattivante, prestazioni ottime ma ha alcuni difetti, software e hardware.

iOS è sistema operativo che ha introdotto nuove features, ma risulta avere ancora troppi bugs; spesso capita che le applicazioni si chiudono improvvisamente.

Mentre dal punto di vista hardware, la batteria risulta essere maggiorata, ma il consumo è ancora un fattore da migliorare, che ne compromette un po' utilità.

Confrontando l'iPhone 6 e iPhone 6 plus posso dire che quest'ultimo non mi ha fatto impazzire, soprattutto per via delle dimensioni troppo eccessive, che ne compromettono

la maneggevolezza; è davvero difficile utilizzare tale dispositivo con una sola mano !

Per una recensione video completa, potete guardare questo filmato:

<http://youtu.be/wkYQnx4c3uw>

[L'iPhone parla, grazie a Speak Notification !](#)

A chi è capitato di sentirsi recapitare dei messaggi mentre si è alla guida, oppure mentre si sta già facendo qualcosa che non permette di poter prendere in mano il cellulare ? D'ora in poi, grazie ad un tweak presente in Cydia, l'iPhone "parlerà", dicendo il testo del messaggio che abbiamo ricevuto ! Il suo nome è Speak Notification ed è disponibile nella repo di BigBoss.



Speak Notification è davvero semplice da usare, quanto geniale ! Infatti, inserendo la lingua tra quelle presente nella lista (tra cui c'è anche l'Italiano), potremmo farci dettare dall'iPhone, il testo del messaggio che abbiamo ricevuto, oppure informarci di un post che è stato pubblicato su Facebook e tanto altro. Inoltre, attraverso le relative impostazioni, sarà possibile attivare Speak Notification solo quando lo desideriamo, ad esempio quando il bluetooth con il relativo auricolare è attivo, per indicare che siamo alla guida. Il prezzo di soli 1.49 \$, è più che accettabile, per il tweak davvero utilissimo, per evitare anche possibili incidenti stradali, dovuti alle possibile distrazioni con l'iPhone.


[Come comandare il condizionatore di casa con Arduino, anche da remoto !](#)

[Arduino](#) permette davvero di fare tantissime cose riguardo [all'Internet of things](#), basta avere un po' di voglia e intuizione, per realizzare qualsiasi tipo di progetto.

In questo post vedremo come comandare da remoto il proprio condizionatore di casa, attraverso Arduino, che è connesso ad Internet. Si avete proprio letto bene !



Requisiti hardware

- 1 x Arduino Uno + 1 x Arduino Mega oppure 2 x Arduino Uno
- Ricevitore ad infrarossi TSOP1738 
- IR Led
- 1 x Resistenza da 330 Ohm
- Qualche cavo flessibile

Requisiti software

- [Libreria IRemote Arduino](#)
- [Programmi disponibili nella mia repo di Github](#)

Come funziona il sistema ?

Come molti sapranno, il meccanismo di regolazione delle varie impostazioni tra utente e condizionatore, avviene attraverso un telecomando, che è basato sull'invio di segnali ad infrarosso. Per approfondire il funzionamento di questi segnali, vi consiglio di guardare il post in riferimento al [Capitolo 10 di Arduino progettare diventa semplice](#).

Ovviamente, i codici di un condizionatore, variano in base al modello e in base al costruttore. Per questo motivo, come prima fase dal lavoro, è necessario fare il procedimento che prende il nome di [reverse engineering](#), ovvero di ottenere i codici per il proprio dispositivo, sfruttando il trasduttore TSOP1738.

Una volta che sono stati ottenuti i codici per il proprio condizionatore, ci siamo quasi a completare "l'opera". Occorre utilizzare un piccolo programma che semplifichi l'impostazione del codice da inserire in Arduino e sarà possibile inviare le varie codifiche. Queste codifiche potranno essere impostate dall'utente, attraverso una semplice pagina web, con o senza protezione.

Come avete potuto leggere nella sezione "requisiti hardware", ho scritto che sono necessari due Arduino; il motivo principale è quello che le sequenze di bit da inviare ai condizionatori occupano troppo spazio per poter essere contenute in un solo dispositivo, che tra l'altro dovrà occuparsi anche della parte di gestione dei comandi che verranno trasmessi via Internet.

In pratica ci sarà un dispositivo che si occuperà della parte webserver e l'altro di inviare i segnali IR. L'altro motivo che mi ha spinto a realizzare questo progetto con due dispositivi, è che si può utilizzare tranquillamente Raspberry PI per la gestione del webserver, in modo da migliorare sicuramente l'aspetto grafico della pagina web.

Come impostare i vari dati

Come specificato nelle righe precedenti, è necessario ottenere le codifiche del nostro telecomando, per poterle ritrasmettere da Arduino. Per farlo faremo uso della libreria IRemote per Arduino, in particolare di questo codice:


```

12345
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91

```

```

/** Modified by Chris Targett* Now includes more
protocols* Novemeber 2011

* IRremote: IRrecvDump - dump details of IR codes
with IRrecv
* An IR detector/demodulator must be connected to
the input RECV_PIN.
* Version 0.1 July, 2009
* Copyright 2009 Ken Shirriff
* http://arcfn.com
*
* Modified by Chris Targett to speed up the
process of collecting
* IR (HEX and DEC) codes from a remote (to put
into and .h file)
*/

#include <IRremote.h>

int RECV_PIN = 11;

IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}

// Dumps out the decode_results structure.
// Call this after IRrecv::decode()
// void * to work around compiler issue
//void dump(void *) {
// decode_results *results = (decode_results *)v
void dump(decode_results *results) {
  int count = results->rawlen;
  if (results->decode_type == UNKNOWN) {
    Serial.print("Unknown encoding: ");
  }
  else if (results->decode_type == NEC) {
    Serial.print("Decoded NEC: ");
  }
  else if (results->decode_type == SONY) {
    Serial.print("Decoded SONY: ");
  }
  else if (results->decode_type == RC5) {
    Serial.print("Decoded RC5: ");
  }
  else if (results->decode_type == RC6) {
    Serial.print("Decoded RC6: ");
  }
  else if (results->decode_type == SAMSUNG) {
    Serial.print("Decoded SAMSUNG: ");
  }
  else if (results->decode_type == JVC) {
    Serial.print("Decoded JVC: ");
  }
  else if (results->decode_type == PANASONIC) {
    Serial.print("Decoded Panasonic: ");
  }
  Serial.print(results->value, HEX);
  Serial.print("");
  Serial.print(results->bits, DEC);
  Serial.println(" bits");
  Serial.print("#define Something_DEC ");
  Serial.println(results->value, DEC);
  Serial.print("#define Something_HEX ");
  Serial.println(results->value, HEX);
  Serial.print("Raw (");
  Serial.print(count, DEC);
  Serial.print("): ");
  for (int i = 0; i < count; i++) {
    if ((i % 2) == 1) {
      Serial.print(results->rawbuf[i]*USECPERTICK, DEC);
    }
    else {
      Serial.print(-(int)results->rawbuf[i]*USECPERTICK,
DEC);
    }
    Serial.print(" ");
  }
  Serial.println("");
}

void loop() {
  if (irrecv.decode(&results)) {
    dump(&results);
    irrecv.resume(); // Receive the next value
  }
}

```

Il collegamento è riportato nell'immagine qui sottostante:



Dopo aver caricato il codice su Arduino, dovremmo premere i vari pulsanti presenti sul telecomando e copiare da qualche parte i risultati mostrati dal seriale.

Nel mio caso, ho ottenuto la seguente sequenza di codifiche:

Accensione 20 C° Caldo

Raw (150): -5320 3000 -3000 3000 -4400 550 -1600 600 -550 550 -1650 550 -550
550 -550 550 -1650 550 -550 550 -1650 500 -550 550 -1650 550 -550 550 -500
600 -500 600 -550 550 -550 550 -1650 500 -550 550 -600 500 -1700 500 -550 550
-550 550 -550 550 -600 500 -550 550 -550 550 -550 550 -550 550 -1650 550
-1650 550 -1650 500 -1650 550 -1650 550 -550 550 -550 550 -550 550 -550 550
-1650 550 -1650 550 -500 550 -550 550 -1700 500 -1650 550 -550 550 -500 600
-550 550 -550 550 -550 550 -550 550 -550 550 -1650 500 -1700 500 -550 550
-550 550 -550 550 -550 550 -550 550 -600 500 -550 550 -550 550 -550 550 -550
550 -550 550 -550 550 -550 550 -550 550 -1650 550 -500 550 -1700 500 -550 550
-550 550 -550 550 -1650 550

Spegnimento

Raw (150): -3276 3100 -2900 2950 -4400 700 -1500 700 -400 700 -1500 700 -400
650 -450 550 -1650 550 -550 650 -1500 600 -500 600 -1600 600 -500 600 -500
600 -450 650 -450 600 -550 550 -550 550 -550 550 -600 500 -1600 600 -500 600
-500 600 -550 550 -500 600 -500 600 -550 550 -550 550 -1600 600 -500 600 -500
600 -500 550 -1650 550 -1600 600 -500 600 -500 600 -550 550 -550 550 -1600
600 -1600 550 -550 550 -550 550 -1650 550 -1600 600 -550 550 -500 600 -500
600 -550 550 -550 550 -500 600 -500 600 -1600 600 -1600 550 -550 550 -550 550
-550 550 -550 550 -550 550 -550 550 -550 550 -550 550 -500 600 -550 550 -550
550 -500 600 -500 600 -500 600 -1600 600 -500 600 -1600 550 -550 550 -1600
600 -550 550 -550 550

Accensione 23 C° Freddo 2 FAN

Raw (150): -16044 3050 -3000 3000 -4400 550 -1600 600 -550 550 -1650 550 -550
550 -550 550 -1650 550 -500 600 -1600 550 -550 550 -1650 550 -1650 550 -1650
550 -550 550 -550 550 -500 600 -1600 550 -550 550 -550 550 -1650 550 -550 550
-550 550 -550 550 -550 550 -550 550 -550 550 -500 600 -1650 500 -550 550 -600
500 -1700 500 -550 550 -550 550 -550 550 -550 550 -550 550 -550 550 -1650 550
-1650 550 -550 550 -500 550 -1650 550 -550 550 -550 550 -550 550 -550 550
-550 550 -550 550 -550 550 -550 550 -550 550 -1650 550 -550 550 -550 550 -550
550 -550 550 -500 550 -550 550 -550 550 -550 550 -550 550 -550 550 -550 550
-550 550 -550 550 -550 550 -550 550 -550 550 -550 550 -550 550 -1650 550
-1600 600 -550 550

Accensione 20 C° Caldo e Swing

Raw (150): 18566 3000 -3000 3000 -4400 550 -1650 550 -500 600 -1600 600 -500
600 -500 600 -1600 550 -550 550 -1650 550 -500 600 -1600 600 -550 550 -550
550 -550 550 -1600 600 -500 600 -500 550 -550 550 -550 550 -550 550 -550 550
-550 550 -550 550 -550 550 -550 550 -550 550 -550 550 -1650 550 -550 550
-1600 550 -550 550 -600 500 -550 550 -550 550 -550 550 -550 550 -550 550
-1650 550 -1650 550 -550 550 -550 550 -1600 550 -1650 550 -550 550 -550 550
-550 550 -550 550 -550 550 -550 550 -550 550 -1650 550 -1650 550 -550 550

-550 550 -550 550 -550 500 -550 550 -550 550 -550 550 -550 550 -550 550 -550
550 -550 550 -550 550 -550 550 -550 550 -550 550 -1650 550 -1650 550 -1650
500 -1650 550 -550 550 -1700 500

Accensione 23 C° Freddo 2 FAN e Swing

Raw (150): 2084 2950 -3050 2950 -4400 550 -1600 600 -550 550 -1650 550 -550
550 -550 550 -1600 600 -500 600 -1600 550 -500 600 -1650 550 -1650 550 -1600
600 -500 600 -1600 600 -500 600 -500 550 -550 550 -550 550 -550 550 -550 550
-550 550 -550 550 -550 550 -500 600 -550 550 -500 600 -1650 550 -550 550
-1600 600 -500 600 -1600 550 -1600 600 -500 600 -500 600 -550 550 -500 600
-1650 550 -1600 600 -500 600 -500 550 -1650 550 -550 550 -550 550 -550 550
-550 550 -550 550 -550 550 -550 550 -550 550 -1650 550 -500 600 -500
600 -500 600 -500 600 -500 550 -550 550 -550 550 -550 550 -550 550 -550 550
-550 550 -550 550 -550 550 -1650 550 -1650 550 -1600 600 -1600 550 -550 600
-500 550 -550 550 -550 550

Come riportato nella libreria IRemote, per poter ritrasmettere questi dati, occorre effettuare alcune modifiche; in particolare occorre sostituire i segni meno, eliminare il primo valore e porre, tra i vari valori, una virgola, dal momento che nel codice di Arduino, verranno inseriti in un vettore di interi.

Per facilitare questa operazione, ho realizzato un piccolo programma in Java, in grado di fare queste operazioni in automatico:

Anche questo programma, è disponibile nella repo di Github e si chiama AcToolArduino. In questa prima versione, il programma automatizza la procedura, inserendo i dati nella TextField. Nella prossima versione, sarà possibile caricare da file, i risultati ottenuti dal seriale di Arduino.

Attraverso il prossimo programma, potremo verificare il corretto funzionamento dell'invio di segnali IR da Arduino:



```
#include "IRremote.h"
```

```
IRsend irsend;
```

```
void setup()
```

```
{
```

```
Serial.begin(9600);
```

```
}
```

```
int khz=38; //NB Change this default value as necessary to the correct  
modulation frequency
```

```
// ON and 20 C° with 1 FAN heat
```

```
unsigned heat[] =  
{3000,3000,3000,4400,550,1600,600,550,550,1650,550,550,550,550,550,1650,550,5  
50,550,1650,500,550,550,1650,550,550,550,500,600,500,600,550,550,550,550,1650  
,500,550,550,600,500,1700,500,550,550,550,550,550,550,600,500,550,550,550,550  
,550,550,550,550,1650,550,1650,550,1650,500,1650,550,1650,550,550,550,550  
,550,550,550,550,1650,550,1650,550,500,550,550,550,1700,500,1650,550,550,550,  
500,600,550,550,550,550,550,550,550,550,550,550,1650,500,1700,500,550,550,550  
,550,550,550,550,550,550,550,600,500,550,550,550,550,550,550,550,550,550,  
550,550,550,550,550,550,1650,550,500,550,1700,500,550,550,550,550,550,165  
0,550};
```

```
// OFF the A/C
```

```
unsigned OFF[] =  
{3100,2900,2950,4400,700,1500,700,400,700,1500,700,400,650,450,550,1650,550,5  
50,650,1500,600,500,600,1600,600,500,600,500,600,450,650,450,600,550,550,550,  
550,550,550,600,500,1600,600,500,600,500,600,550,550,500,600,500,600,550,550,  
550,550,1600,600,500,600,500,600,500,550,1650,550,1600,600,500,600,500,600,55  
0,550,550,550,1600,600,1600,550,550,550,550,550,1650,550,1600,600,550,550,500  
,600,500,600,550,550,550,550,500,600,500,600,1600,600,1600,550,550,550,550,55  
0,550,550,550,550,550,550,550,550,550,550,550,500,600,550,550,550,550,500  
,600,500,600,500,600,1600,600,500,600,1600,550,550,550,1600,600,550,550,550,5  
50};
```

```
// ON and 23° with 2 FAN cold
```

```
unsigned cold[] =  
{3050,3000,3000,4400,550,1600,600,550,550,1650,550,550,550,550,550,1650,550,5  
00,600,1600,550,550,550,1650,550,1650,550,1650,550,550,550,550,550,500,600,16  
00,550,550,550,550,550,1650,550,550,550,550,550,550,550,550,550,550,550,550,5  
50,500,600,1650,500,550,550,600,500,1700,500,550,550,550,550,550,550,550,  
550,550,550,550,1650,550,1650,550,550,550,500,550,1650,550,550,550,550,550,55  
0,550,550,550,550,550,550,550,550,550,550,550,550,1650,550,550,550,550,55  
0,550,550,550,550,500,550,550,550,550,550,550,550,550,550,550,550,550,550,  
,550,550,550,550,550,550,550,550,550,550,550,550,1650,550,1600,600,550,55  
0};
```

```
// ON and 20 C° with 1 FAN heat and SWING
```

```
unsigned heat_with_swing[] =  
{3000,3000,3000,4400,550,1650,550,500,600,1600,600,500,600,500,600,1600,550,5  
50,550,1650,550,500,600,1600,600,550,550,550,550,550,550,1600,600,500,600,500  
,550,550,550,550,550,550,550,550,550,550,550,550,550,550,550,550,550,550,  
550,550,1650,550,550,550,1600,550,550,550,600,500,550,550,550,550,550,550
```

```
, 550, 550, 550, 1650, 550, 1650, 550, 550, 550, 550, 550, 1600, 550, 1650, 550, 550, 550, 550, 550,
550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 1650, 550, 1650, 550, 550, 550, 550, 550,
550, 550, 550, 500, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550,
550, 550, 550, 550, 550, 550, 550, 1650, 550, 1650, 550, 1650, 500, 1650, 550, 550, 550, 1700,
500};
```

```
// ON and 23° with 2 FAN cold and SWING
```

```
unsigned cold_with_sqwing[] =
{2950, 3050, 2950, 4400, 550, 1600, 600, 550, 550, 1650, 550, 550, 550, 550, 550, 1600, 600, 5
00, 600, 1600, 550, 500, 600, 1650, 550, 1650, 550, 1600, 600, 500, 600, 1600, 600, 500, 600, 5
00, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 500, 600, 550, 55
0, 500, 600, 1650, 550, 550, 550, 1600, 600, 500, 600, 1600, 550, 1600, 600, 500, 600, 500, 600
, 550, 550, 500, 600, 1650, 550, 1600, 600, 500, 600, 500, 550, 1650, 550, 550, 550, 550, 550, 5
50, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 55
0, 550, 600, 500, 600, 500, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 550, 55
0, 550, 550, 550, 1650, 550, 1650, 550, 1600, 600, 1600, 550, 550, 600, 500, 550, 550, 550, 550
, 550};
```

```
void loop() {

  irsend.sendRaw(heat, sizeof(heat)/sizeof(int), khz);

  delay(10000);

  irsend.sendRaw(OFF, sizeof(OFF)/sizeof(int), khz);

  delay(10000);

}
```

Se siamo usciti ad azionare il condizionatore di casa, siamo a buon punto per poter completare il progetto !

Come indicato precedentemente, è necessario utilizzare due Arduino, uno che gestisce la parte webserver, l'altro che gestisce la comunicazione con il condizionatore.

Per collegare i due Arduino in maniera corretta, occorre prestare attenzione all'immagine successiva.



Sul dispositivo collegato con la Shield Ethernet è necessario caricare il programma che prende il nome di AC_Webpage_Controller.ino, mentre nell'Arduino che invierà i comandi IR, occorre caricare il programma che prende il nome di AC_Sender_Code.ino.

Ora che abbiamo completato la procedura di caricamento, non ci resta che fare alcune operazioni per perfezionare il progetto; quella più utile, è quella di

rendere accessibile il webserver anche da remoto. Per farlo dovremmo “aprire le porte” del nostro router, per instradare correttamente le richieste che vengono fatte al di fuori della nostra rete locale.

Occorre precisare, che nella versione del codice riportata su Github, non è presente una protezione a livello di password, per semplificare un po’ il progetto. Tuttavia, con alcune modifiche al codice AC_Webpage_Controller.ino, sarà possibile implementare anche questa caratteristica.

L’operazione di inoltro delle porte, varia da router a router; tuttavia, in generale, si può procedere così:

- Individuare l’IP del webserver di Arduino
- Individuare l’IP del gateway
- Aprire il browser e digitare l’IP del nostro router e inserire le credenziali
- Aprire la sezione di inoltro delle porte e inserire come dati, l’IP di Arduino e come porta la 80.

Ora potremmo accedere anche al di fuori della nostra rete locale, semplicemente aprendo una pagina web, con l’IP pubblico della propria connessione ADSL (per recuperare questo valore, basterà semplicemente digitare su Google.com “my ip”).

Ora potremmo finalmente testare il nostro progetto !



Spero che questo progetto, sia utile per capire come è possibile realizzare qualcosa di davvero carino con Arduino, con un po’ di pazienza e soprattutto con un po’ di teoria !

Ecco il video del risultato finale:

<http://youtu.be/i9Jo3F9GyFo>

Come emulare sull'iPhone e iPad il vecchio Game Boy

Prima che venissero introdotti sul mercato gli smartphone, il principale strumento di intrattenimento era il Game Boy; con questo dispositivo, inserendo opportune "schede" era possibile giocare con Super Mario, con i Pokemon e con tantissimi altri giochi. Ora su iPhone, iPad e Android è possibile trovare tantissimi giochi sui vari store, con una grafica sicuramente migliore rispetto a quella dei vecchi Game Boy. Ma per chi è nostalgico dei vecchi dispositivi, come può farli funzionare sui dispositivi iOS ? Con GB4iOS è possibile farlo, anche per i dispositivi non sbloccati !



Per installare questa semplice applicazione, è necessario aprire dal dispositivo la seguente pagina web <http://gba4iosapp.com/download/>. Occorre ricordare, che per effettuare l'installazione correttamente, per i dispositivi non sbloccati, è necessario impostare la data sul proprio dispositivo, prima del 18 Febbraio 2014:



Una volta scaricata l'applicazione, sarà possibile aggiungere, in maniera semplice, i vari giochi presenti nella repo ufficiale direttamente dalla schermata principale. Sarà necessario, tuttavia, installare ulteriori pacchetti per far funzionare alcuni giochi. Per esempio, per i dispositivi sbloccati, è necessario installare gratuitamente da Cydia il pacchetto gpSPhone. Inoltre, una grande funzionalità, presente in GBA4IOS è quella di poter salvare i risultati ottenuti nei vari giochi.

Questa applicazione è davvero utilissima per chi vuole riassaporare i vecchi giochi che funzionavano sul Game Boy; inoltre, dal momento che l'installazione è davvero facile, è ancora più interessante il suo utilizzo !
Buon divertimento !

GoalShouter: l'app per commentare le partite nel WEB

Si sa che gli italiani amano vedere il calcio e anche giocarlo. La Domenica tipica di un tifoso, è quella di guardare la partita della propria squadra in TV. Tuttavia, vengono trasmesse soltanto un numero esiguo di partite sullo schermo di casa, perché l'emittenti TV non hanno motivi economici per acquistare i diritti TV. Allora come è possibile far partecipare ad un numero maggiore di persone, alla partita ? La soluzione [GoalShouter](#) !



[GoalShouter](#) è un servizio made in Italy, davvero geniale e semplice da usare; lo scopo principale di questo sito è quello di condividere statistiche, telecronaca scritta, immagini e tanto altro, riguardo alle partite di calcio di qualsiasi livello, dalla Serie A, alla Terza Categoria, passando per le partite di calcetto della domenica tra amici.



Il funzionamento è davvero semplice; occorre scaricare l'applicazione gratuita per Android e per iOS dai vari store e procedere alla registrazione, inserendo nome utente, email e password. Tuttavia la procedura di registrazione è possibile farla dal seguente link <http://www.goalshouter.com/register> .



Una volta completata la registrazione, confermando l'email ricevuta al propria account di posta elettronica, possiamo creare la nostra squadra, la quale sarà così in grado di poter partecipare alle varie partite. L'inserimento dei dati è davvero semplice, dal momento che basterà inserire alcune informazioni, riguardo alla squadra. Per creare la partita è necessario fare tap alla voce "Crea partita". Nella schermata successiva, sarà impossibile inserire le due squadra sfidanti. Per l'inserimento dei nomi dei giocatori, basterà fare tap su "Inserisci le formazioni", dopo aver inserire i nomi delle due squadre. Attraverso la voce "Modifica impostazioni", sarà possibile modificare la durate dei tempi di gioco, e varie altre opzioni, come ad esempio la condivisione su Facebook e Twitter. Infatti, oltre alla generazione di una pagina web dedicata alla partita di calcio, è possibile condividere le statistiche riguardo alla partita anche sui due social network. Inoltre, sarà semplice commentare, per il singolo utente, più partite alla volta, grazie al widget presente nell'applicazione.



L'inserimento delle statistiche, delle foto e dei commenti, come già detto in precedenza, è possibile farlo direttamente dall'applicazione disponibile per iOS e Android.



GoalShouter è davvero un servizio geniale, perché racchiude nella sua semplicità di utilizzo, un servizio davvero professionale per seguire le partite di calcio di qualsiasi livello. Il risultato delle telecronache è davvero più che buono e l'idea di condividere momenti di sport vero con i propri amici e con il web, è davvero geniale. Inoltre il servizio funziona sia per i semplici utenti che vanno a vedere all stadio le partite di calcio, oppure per i giocatori di calcetto della domenica, sia per le società di calcio di leghe inferiori a quella di Serie A, che vogliono condividere le proprie partite. Per ulteriori informazioni ecco alcuni esempi di telecronache live <http://urli.st/fj0-GoalShouter-examples-wwwgoalshoutercom>.

Ed ecco il video di presentazione del servizio:

<http://youtu.be/lqX3kLoP52s>

—

[Le migliori applicazioni per caricare le proprio foto su Instagram](#)

Per che non lo conoscesse ancora Instagram è un social network basato sul caricamento di immagini dal proprio smartphone. Per esempio è possibile inserire per esempio, l'immagine di un tramonto, di una partita di calcio etc... Il meccanismo della condivisione delle proprie foto è simile a quello di Facebook, nel senso che è possibile aggiungere i propri amici e vedere gli aggiornamenti di quest'ultimi nella pagina principale.



Inoltre per facilitare la ricerca di immagini di altri utenti, sono stati inseriti gli hashtag, molto verosimilmente a quanto accade in Twitter.

Per esempio digitando la parola #sunset, sarà possibile vedere i tramonti più belli caricati su Instagram.



Nell'applicazione ufficiale disponibile per [iOs](#) e [Android](#) sono disponibili alcuni effetti fotografici, specialmente i filtri, per rendere le proprie foto uniche.



Tuttavia nei vari stores, sono presenti tantissime applicazioni, che permettono di aggiungere nuovi effetti fotografici alle proprie foto, come ad esempio i collage, effetto fumetto e tanti altri. Le prossime applicazioni oltre che essere gratuite, sono disponibili sia per iOs che per Android, attraverso i relativi stores.

Vediamo quali sono le migliori:

PhotoGrid

Questa applicazione permette di creare ottimi collage di foto, in maniera facile e veloce. Selezionando le foto dal rullino fotografico dello smartphone, sarà possibile selezionare vari layout, filtri etc... sfruttando il tab in basso.



Il risultato sarà davvero ottimo, dal momento che permetterà di inserire più foto, in un'unica immagine, già adattata al formato di Instagram.



InstaWeather

Questa applicazione permette di scattare delle foto, inserendo la situazione del meteo della posizione in cui ci troviamo. Per esempio vogliamo condividere una bella giornata con una temperatura piacevole, possiamo farlo senza troppe difficoltà, oppure dire ai nostri amici che in città nevica etc..



Retrica

Questa applicazione disponibile sia per iOs che per Android, permette di aggiungere tantissimi filtri alle proprie fotografie, per renderle davvero perfette. Inoltre sono disponibili altri effetti fotografici, che non sono

presenti nell'applicazione ufficiale di Instagram



Lezione 8: Utilizziamo il KeyPad con Arduino

In questa ottava puntata su Arduino, introdurremo un nuovo strumento, utile per tanti diversi progetti: il KeyPad.



Il meccanismo di funzionamento è davvero semplice. Esso è formato da 7 cavi, che rappresentano 4 righe e 3 colonne. Attraverso la pressione di un tasto, c'è una variazione di tensione sul Pin collegato ad Arduino e quest'ultimo è in grado di rilevare il relativo valore.

Attraverso questo device è davvero possibile fare tantissimi progetti, come ad esempio una simulazione di un piccolo Bancomat, proteggere una porta d'entrata e tanto altro. Vediamo qualche esempio per Arduino.

Esempi di programmazione del KeyPad

Requisiti Hardware:

- Arduino
- KeyPad
- LED
- Relay
- Servo

Requisiti Software:

- Libreria [keypad](#)

1° Programma: Hello KeyPad

In questo piccolo programma, vedremo come mostrare su seriale il tasto premuto sul KeyPad

```
/* @file HelloKeypad.pde
|| @version 1.0
|| @author Alexander Brevig
|| @contact alexanderbrevig@gmail.com
||
|| @description
```

```

|| | Demonstrates the simplest use of the matrix Keypad library.
|| #
*/
#include <Keypad.h>

const byte ROWS = 4; //four rows
const byte COLS = 3; //three columns
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};
byte rowPins[ROWS] = {8,7,6,5}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {4,3,2}; //connect to the column pinouts of the keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup(){
  Serial.begin(9600);
}

void loop(){
  char key = keypad.getKey();

  if (key){
    Serial.println(key);
  }
}

```

Il funzionamento del programma dal punto di vista hardware è già stato spiegato precedentemente. Dal punto di vista software, la situazione è davvero semplice. Sfruttando le librerie già presenti sul sito di Arduino, il loop() è davvero immediato da capire: una volta che viene premuto qualcosa, si entra in if, che stampa sul seriale il relativo valore, semplice no ? Il collegamento del KeyPad con Arduino è semplice, basta guardare la seguente foto.



In pratica, prendendo il KeyPad dal lato frontale, basta collegare i cavi in modo parallelo, partendo dal Pin 8 fino al Pin 2.

Ecco il video che mostra il KeyPad in azione con Arduino:

<http://youtu.be/XX7wQ8GrFVo>

2° Programma: KeyPad che permette di proteggere qualcosa

Questo secondo programma può essere modificato dall'utente. In pratica, questo codice, non fa altro che permette di inserire un codice con il KeyPad, verificarlo tramite Arduino e se tale codice è corretto, accende un LED. In

realtà è possibile fare qualsiasi cosa, come ad esempio accendere una lampada, aprire una porta con un servo. Lascio questo codice, in modo che chiunque possa implementarlo a suo piacimento.

```
/**
Questo programma permette di far inserire da un utente un codice con il
KeyPad. Qualora il codice inserito sia corretto, viene acceso un LED.
Per inviare il codice, una volta digitato interamente è necessario premere *.
Mentre per spegnere il LED, oppure re-digitare il codice è necessario premere #
#
Autore Giacomo Bellazzi
Versione 1.0
*/
#include <Keypad.h>
#define LED 13
const byte ROWS = 4; //quattro righe
const byte COLS = 3; //tre colonne
char keyInsert[6];
// Queste variabili servono come verifica del corretto inserimento del codice
int i = 0;
int j = 0;
int s = 0;
int x = 0;
// Codice segreto
char code[7]= "112233";
char keys[ROWS][COLS] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'*','0','#'}
};
byte rowPins[ROWS] = {8,7,6,5}; //i Pin a cui sono connesse le righe del
KeyPad
byte colPins[COLS] = {4,3,2}; // i Pin a cui sono connesse le colonne del
KeyPad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup(){
    Serial.begin(9600);
    pinMode(LED,OUTPUT);
}

void loop(){
    char key = keypad.getKey();
    if (i==0){
        Serial.println("Insert PIN to verify...");
        i++;
    }
    if (key != NO_KEY && j<6){
        Serial.print("*");
    }
}
```

```

    //Serial.println(key);
    keyInsert[j]=key;
    j++;
}
if(key == '*') {
    Serial.println();
    Serial.println("Verifyng the code...");
    delay(1000);
    for(s=0; s<6;s++){
        if(keyInsert[s]==code[s]){
            x++;
        }
    }
    if(x==6){
        Serial.println("The code is correct");
        digitalWrite(LED,HIGH);
        //TODO possibili ulteriori implementazioni
    }else{
        Serial.println("The code is incorrect, please retry");
        delay(2000);
        x=0;
        i=0;
        j=0;
    }
}
if(key == '#'){
    x=0;
    i=0;
    j=0;
    digitalWrite(LED,LOW);
}
}

```

Il codice risultato essere leggermente un po' più complesso del primo, tuttavia attraverso i commenti presenti è facile intuire il meccanismo di funzionamento dei vari blocchi di codice. In pratica vengono letti i valori inseriti attraverso il KeyPad. Una volta terminata l'immissione dei 6 caratteri, si preme il comando * e Arduino valuta se il codice inserito è lo stesso di quello presente in memoria. Questo avviene attraverso un ciclo for e un if che verificano se i 6 valori sono uguali.

Ecco il video che mostra il funzionamento del programma :

<http://youtu.be/5J6lcA3D9nE>

3°Programma: Utilizziamo il OTP e il KeyPad

Questo programma è stato realizzato da [Luca Dentella](#), un appassionato di Arduino. Questo programma si basa sull'utilizzo del metodo di sicurezza OTP,

che viene usato da chi ha un Online Banking. Per chi non sapesse cosa sia l'OTP, ecco una definizione tratta da Wikiepdia:

Una **One-Time Password** ([password](#) usata una sola volta) è una password che è valida solo per una singola sessione di accesso o una transazione. La OTP evita una serie di carenze associate all'uso della tradizionale password (statica). Il più importante problema che viene risolto da OTP è che, al contrario della password statica, esso non è vulnerabile agli [attacchi con replica](#). Ciò significa che, se un potenziale intruso riesce ad intercettare una OTP che è stata già utilizzata per accedere a un servizio o eseguire una transazione, non sarà in grado di riutilizzarla, in quanto non sarà più valida. D'altra parte, una OTP non può essere memorizzata da una persona. Essa richiede quindi una tecnologia supplementare per poter essere utilizzata.^[1]

Come vengono generate le password OTP ?

Gli algoritmi di generazione delle OTP in genere fanno uso di numeri casuali. Ciò è necessario perché altrimenti sarebbe facile prevedere l'OTP futuro osservando i precedenti. Gli algoritmi OTP che sono stati realizzati sono abbastanza diversi tra loro. I vari approcci per la generazione di OTP sono elencati di seguito.

- Algoritmi basati sulla **sincronizzazione temporale** tra server di autenticazione e client che fornisce la password (le OTP sono valide solo per un breve periodo di tempo)
- Algoritmi matematici che generano una nuova **password in base alla password precedente** (le OTP sono, di fatto, una catena di password legate tra loro, e devono essere utilizzate in un ordine predefinito)
- Algoritmi matematici dove la password è **basata su una sfida** (per esempio, un numero casuale scelto dal server di autenticazione o dai dettagli della transazione) e/o su un contatore.

Ci sono anche diversi modi per rendere note all'utente le successive OTP da usare. Alcuni sistemi elettronici prevedono l'uso di speciali *token* che l'utente porta con sé, che generano le OTP e le mostrano utilizzando un piccolo display. Altri sistemi sono costituiti da un software che gira sul telefono cellulare dell'utente. Altri sistemi generano le OTP sul lato server e le trasmettono all'utente su un canale [fuori banda](#), come ad esempio un canale di messaggistica SMS. Infine, in alcuni sistemi le OTP sono stampate su carta, che l'utente è tenuto a portare con sé.

In pratica l'OTP si basa su un calcolo di codice, che è formato da una password one time e da un algoritmo che proviene dall'attuale ora.

Per il calcolo del codice, viene incontro una semplice applicazione di Google. Per il download della parte di Arduino, trovato tutto il materiale sul post dell'autore del codice <http://www.lucadentella.it/2013/09/14/serratura-otp/2/>.

Per prima cosa, è necessario inserire hmackey all'interno del codice di Arduino; questo valore di ricava inserendo la password di default,

all'interno di questo script, realizzato sempre da Luca Dentella <http://www.lucadentella.it/OTP>



Una volta copiato il codice nello Sketch per Arduino, basterà compilare e per quanto riguarda la parte legata ad Arduino, abbiamo finito. Ora è necessario installare il programma per il calcolo del codice OTP sul nostro Smartphone: [Google Authenticator](#).

Ora che abbiamo installato il programma lo apriamo e attraverso la fotocamera, puntiamo lo smartphone sul codice QR che è stato generato in precedenza. Ora abbiamo finito e possiamo testare il programma, attraverso il seriale con Arduino.

Ecco un video che mostra l'utilizzo del programma:

<http://youtu.be/AQuTeuIpuTg>

KeyPad 4x4



Sul web è possibile trovare una versione del KeyPad con 16 caratteri, cioè con i numeri dallo zero al nove e le lettere AB,C,D. Il meccanismo di funzionamento è lo stesso, il codice varia leggermente. Ecco due esempi come mostrano l'utilizzo del KeyPad nella sua versione "più grande":

4° Programma con KeyPad 4x4: HelloKeyPad 4x4

```
/*Questo programma mostra come usare la tastiera KeyPad 4x4 con Arduino Uno.
Il testo che viene premuto sulla KeyPad viene mostrato nel seriale
*/
```

```
#include <Keypad.h>
```

```
const byte ROWS = 4; //quattro righe
const byte COLS = 4; //quattro colonne
byte colPins[4] = {5,4,3,2}; // Pin a cui sono connesse le colonne
byte rowPins[4] = {9,8,7,6}; // Pin a cui sono connesse le righe
char Keys[4][4]= //creo la matrice dei tasti della tastiera.
```

```
{
{'1','2','3','A'} ,
{'4','5','6','B'},
{'7','8','9','C'},
{'*','0','#','D'}
};
```

```
Keypad keyp = Keypad(makeKeymap(Keys), rowPins, colPins,4,4);
```

```
void setup(){
```

```

    Serial.begin(9600);
}

void loop(){
    char key = keyp.getKey();
    if (key){
        Serial.println(key);
    }
}

```

5°Programma con Keypad 4×4: Codice Segreto con Lettere

```

/**
Questo programma permette di far inserire da un utente un codice con il
Keypad. Qualora il codice inserito sia corretto, viene acceso un LED.
Per inviare il codice, una volta digitato interamente è necessario premere *.
Mentre per spegnere il LED, oppure re-digitare il codice è necessario premere #
Autore Giacomo Bellazzi
Versione 1.0
*/
#include <Keypad.h>
#define LED 13
const byte ROWS = 4; //quattro righe
const byte COLS = 4; //quattro colonne
char keyInsert[6];
// Queste variabili servono come verifica del corretto inserimento del codice
int i = 0;
int j = 0;
int s = 0;
int x = 0;
// Codice segreto
char code[7]= "11ABCD";
char Keys[ROWS][COLS]= //creo la matrice dei tasti della tastiera.
{
{'1','2','3','A'} ,
{'4','5','6','B'},
{'7','8','9','C'},
{'*','0','#','D'}
};
byte colPins[4] = {5,4,3,2}; // Pin a cui sono connesse le colonne
byte rowPins[4] = {9,8,7,6}; // Pin a cui sono connesse le righe

Keypad keypad = Keypad( makeKeymap(Keys), rowPins, colPins, ROWS, COLS);

void setup(){
    Serial.begin(9600);
    pinMode(LED,OUTPUT);
}

void loop(){

```



```

char key = keypad.getKey();
if (i==0){
  Serial.println("Insert PIN to verify...");
  i++;
}
if (key != NO_KEY && j<6){
  Serial.print("*");
  //Serial.println(key);
  keyInsert[j]=key;
  j++;
}
if(key == '*') {
  Serial.println();
  Serial.println("Verifyng the code...");
  delay(1000);
  for(s=0; s<6;s++){
    if(keyInsert[s]==code[s]){
      x++;
    }
  }
  if(x==6){
    Serial.println("The code is correct");
    digitalWrite(LED,HIGH);
    //TODO possibili ulteriori implementazioni
  }else{
    Serial.println("The code is incorrect, please retry");
    delay(2000);
    x=0;
    i=0;
    j=0;
  }
}
if(key == '#'){
  x=0;
  i=0;
  j=0;
  digitalWrite(LED,LOW);
}
}

```

Conclusione

Spero che i programmi presentati in questo post, siano di facile lettura e soprattutto, siano possibili di ulteriori implementazioni.

Chad2Win: l'applicazione che permette di chattare gratuitamente e guadagnare soldi

Vi ricordati fino a poco tempo fa, quanto si pagava per inviare SMS ai propri amici con il telefonino ? Poi, grazie all'avvento di Internet e la possibilità di utilizzare quest'ultimo sul proprio Smartphone, sono spuntate come funghi applicazioni che permettono di inviare messaggi, come ad esempio:

- WhatsApp 
- Viber 
- Line 
- WeChat 
- iMessage 

Ora è disponibile presso i vari stores, una nuova applicazione che non solo permette di inviare messaggi gratuiti ai propri amici, ma permette anche di guadagnare soldi ! Non ho sbagliato a scrivere ! Siamo passati dal dover pagare gli SMS a ricevere soldi per inviarli ! Il servizio si chiama [Chad2Win](#).



Il meccanismo di guadagno funziona attraverso i banner pubblicitari, che sono presenti nell'applicazione; essi sono presenti nella pagina dove stiamo scrivendo il messaggio e non sono intrusivi come verrebbe naturale pensare. Con la semplice operazione di tap su di essi, si potranno guadagnare dei crediti, che possono essere convertiti in denaro, da poter essere trasferito sul nostro conto di Paypal, con pochi passaggi. Semplice no ? Quindi più si inviano messaggi, più sarà possibile guadagnare soldi veri ! Provare per credere !

