

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 prezzi¹	• 16GB € 729	• 16GB € 839
	• 64GB € 839	• 64GB € 949
	• 128GB € 949	• 128GB € 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

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:

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 TS0P1738 
- 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 *v) {
// 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 -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 -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 -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
-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 -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 neccessary 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
,550,1650,550,550,550,1650,500,550,550,1650,550,550,550,500,60
0,500,600,550,550,550,550,1650,500,550,550,600,500,1700,500,55
0,550,550,550,550,550,600,500,550,550,550,550,550,550,550,
1650,550,1650,550,1650,500,1650,550,1650,550,550,550,550,550,5
50,550,550,550,1650,550,1650,550,500,550,550,550,1700,500,1650
,550,550,550,500,600,550,550,550,550,550,550,550,550,550,550,1
650,500,1700,500,550,550,550,550,550,550,550,550,550,550,550,550
,550,550,550,550,550,550,550,550,550,550,550,550,550,550,550
,550,1650,550,500,550,1700,500,550,550,550,550,550,550,1650,55
0};

// OFF the A/C

unsigned          OFF[]          =
{3100,2900,2950,4400,700,1500,700,400,700,1500,700,400,650,450
,550,1650,550,550,650,1500,600,500,600,1600,600,500,600,500,60
0,450,650,450,600,550,550,550,550,550,550,600,500,1600,600,500
,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,
```


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: