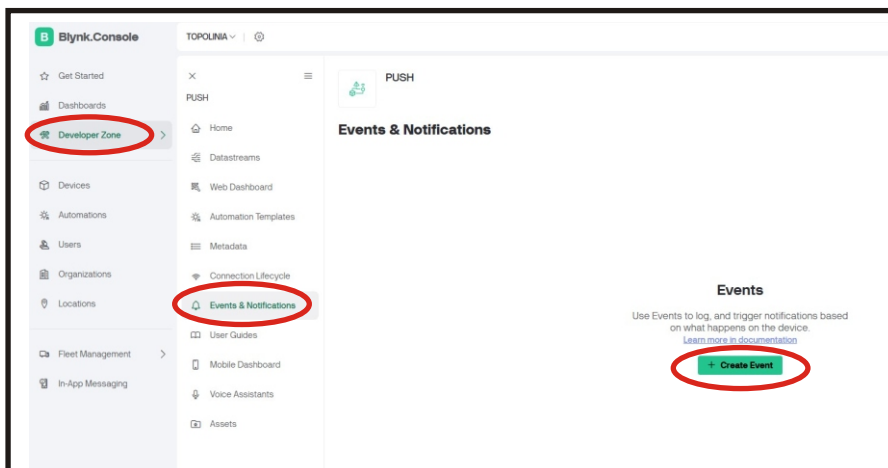


# Gli approfondimenti di L@BOROBOTICA

## ESP 32: progetto per un allarme con notifiche push

Lo smartphone da un bel po' di tempo può essere usato non solo per telefonare; si possono reperire informazioni, chattare oppure comunicare in mille modi differenti. Di solito, giovani e adulti tengono sempre sotto controllo il dispositivo per essere aggiornati sulle ultime novità (la cosa è ormai diventata patologica).

Le notifiche push sono un sistema di messaggistica istantanea che raggiunge l'utente senza che questo debba effettuare alcun download: il testo appare sullo schermo dello smartphone (o del tablet) a condizione che il client sia attivo in background e connesso alla rete. La notifica push è un annuncio pop-up utile quando si vuole informare rapidamente una persona. Vedremo in queste schede, di progettare un allarme con PIR, che rileverà il movimento in una stanza, comunicandocelo con una notifica push sullo schermo dello smartphone (per un massimo di 20 notifiche per ora).



Come già visto con l'uso di Blynk, creiamo un template di nome PUSH, poi selezioniamo **Developer Zone > Event & Notifications > Create Events**

The image shows a screenshot of the Blynk Console configuration form for an event named 'INTRUSIONE'. The form has tabs for 'General', 'Notifications', 'Settings', and 'Expose to Automations'. Under 'General', there are fields for 'EVENT NAME' (filled with 'INTRUSIONE') and 'EVENT CODE' (filled with 'allarme'). Below these are radio buttons for 'TYPE' with options 'Info', 'Warning', 'Critical', and 'Content'. There is a 'DESCRIPTION (OPTIONAL)' field with a placeholder 'Enter...'. At the bottom right, there are 'Cancel' and 'Save' buttons.

Si apriranno una serie di schede. Su General daremo il nome all'evento: questo nome comparirà sulla notifica push. Daremo poi un codice all'evento, in questo caso "allarme", il quale andrà inserito come parametro nella funzione `Blynk.logEvent()` dello sketch che caricheremo sull'ESP32.

## INTRUSIONE

General **Notifications** Settings Expose to Automations

Enable notifications

**Recipients**

Users can set up the notification preferences by going to the Device > Notifications & Events > Notification Settings in Console, or they can also disable them from the Left sidebar > Settings in the mobile app.

E-MAIL TO  
Select contact

PUSH NOTIFICATIONS TO  
Device Owner

SMS TO  
Select contact

**Settings**

Enable notifications management  
When turned ON, end-users will access advanced notification management for this event

Deliver push notifications as alerts  
When turned on, push notifications will use critical alert sounds. End-users will need to turn this setting on in their app settings. They can also change a sound.

Cancel

Nella sezione Notifications decideremo che tipo di notifica ricevere; in questo caso Device Owner, cioè sul nostro smartphone. **Abilitiamo anche tutti gli interruttori.**

## INTRUSIONE

General Notifications **Settings** Expose to Automations

**Limit**

Every 1 message will trigger the event

Event will be sent to user only once per 1 second

**Show in Notifications & Events section**

Device page (app and console)

Home screen (app only)

**Other**

Apply tag when this event is recorded + Choose Tag

Allow users with Edit Timeline permission to archive this event

Cancel

Nella sezione Settings faremo in modo che nel caso di trigger (stato alto sul pin 27 ) ogni secondo venga inviato un messaggio.

```
#define BLYNK_PRINT Serial
#define BLYNK_TEMPLATE_ID "TMPL4MI9dCcyL"
#define BLYNK_TEMPLATE_NAME "PUSH"
#define BLYNK_AUTH_TOKEN "eaQyPFIkni1Vb_Nso0Gjvak4qEfqb5-5"
```

I dati forniti da Blynk alla creazione del Device.

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
```

```
char auth[] = BLYNK_AUTH_TOKEN;
```

```
char ssid[] = "Vodafone-***";
char pass[] = "****";
```

Le credenziali di accesso alla rete WiFi.

```
#define PIR 27
BlynkTimer timer;
```

Il pin a cui è connesso il sensore PIR.

```
void intrusione()
{
  int stato = digitalRead(PIR);
```

La sub intrusione() viene richiamata ogni qualvolta il pin 27 va a stato alto (PIR eccitato).

```
  if (stato==1)
  {
    Blynk.logEvent("allarme","allarme");
    delay(5000);
```

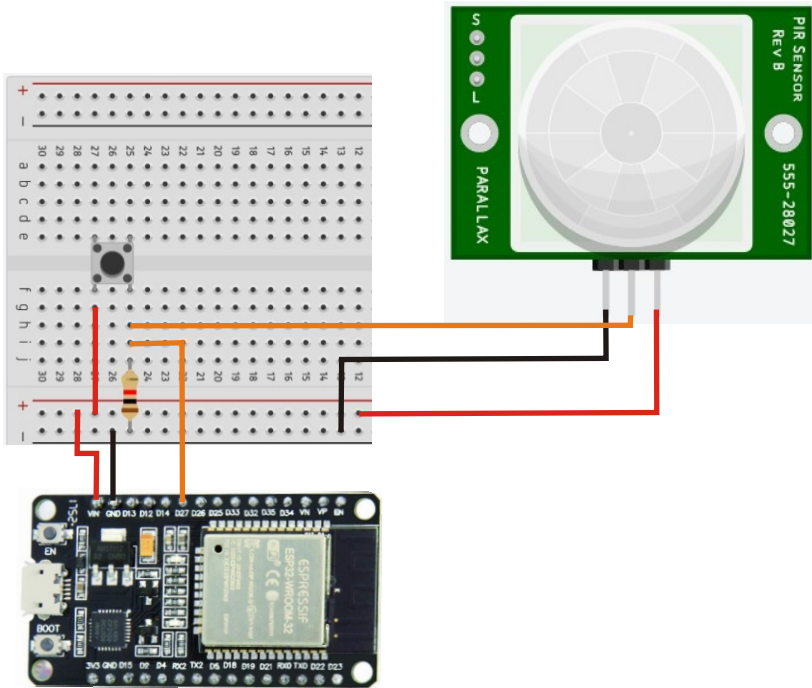
Questo delay impedisce la sovrapposizione di messaggi

```
//
  }
  else if (stato==0)
  {
  }
}
```

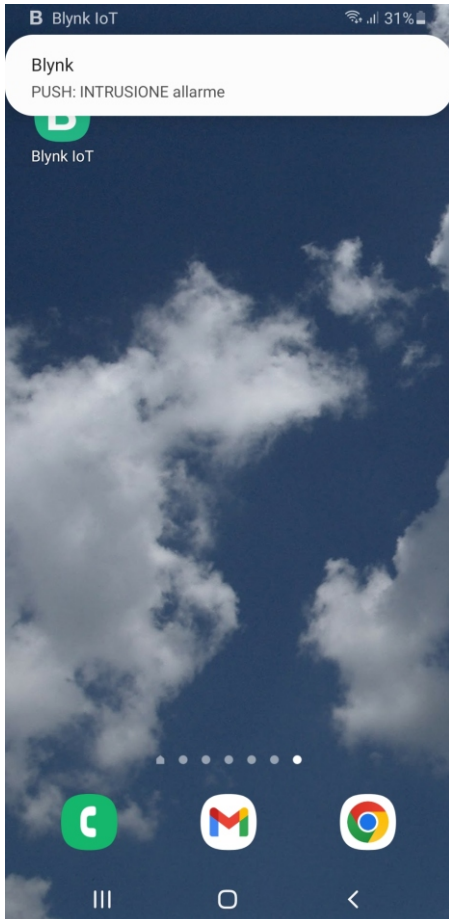
```
void setup(){
  pinMode(PIR, INPUT);
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  timer.setInterval(1000L, intrusione);
}
```

Ogni secondo verrà chiamata la sub intrusione().

```
void loop(){
  Blynk.run();
  timer.run();
}
```



Il circuito di cablaggio.



Ecco come si presenta la notifica PUSH all'arrivo sullo smartphone.  
 Nome del template: PUSH  
 Nome dell'evento: INTRUSIONE  
 Messaggio: "allarme".

Attenzione che sul nostro smartphone le notifiche devono essere abilitate:  
 Impostazioni > Notifiche.