La livella elettronica: inclinazione e urti con l'ADXL345

Guida online, sempre aggiornata

Wemos D1 mini e accelerometro ADXL345 sul modulo GY-291: l'inclinazione ricavata dalla gravità, una livella a bolla sul display OLED, il tocco e la caduta libera con l'interrupt, i valori nel plotter seriale.

A cura di **Ada Intelisano** · redazione tecnica di informaticait.it

IN BREVE

- Fermo, l'ADXL345 sente solo la gravità: da come 1 g si divide tra X, Y e Z si ricava l'inclinazione (Analog Devices, nota AN-1057).
- Tutto a 3,3 V dal pin 3V3 della D1 mini: sensore e display su D6 (SDA) e D5 (SCL), INT1 su D7.
- Sul display una livella a bolla con i due angoli; la prova del giro corregge lo zero di sensore e montaggio.
- Tocco e caduta libera li riconosce il sensore e li segnala alzando INT1; gli angoli vanno nel plotter seriale.

Tempo: un pomeriggio | Livello: un passo avanti, con una saldatura | Prima: guide n. 9 e 22

Un accelerometro fermo sul tavolo sente una cosa sola: la gravità. Se lo inclini, quella spinta si divide tra i suoi tre assi, e dal modo in cui si divide si capisce di quanti gradi è inclinato. In questa guida lo usiamo così: l'ADXL345, una D1 mini e un piccolo display diventano una livella a bolla. Poi le insegniamo a sentire un colpetto e a capire quando cade, con l'interrupt. Tutto sta su una breadboard, che fa da piano della livella.

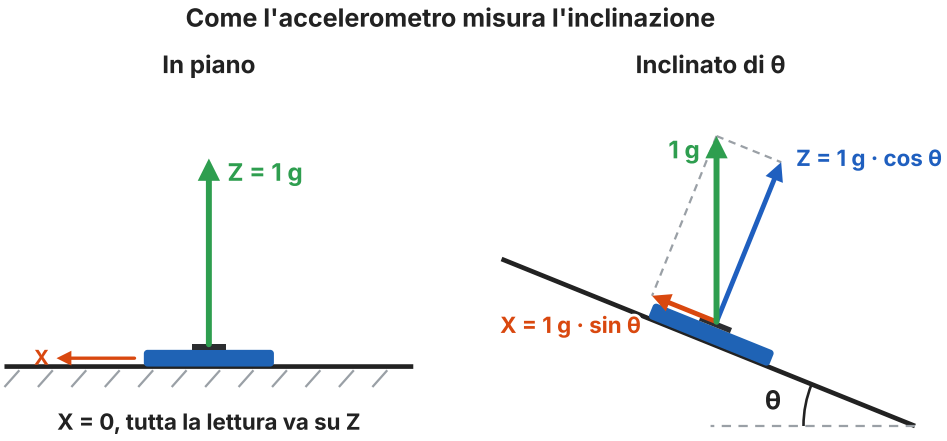
1 Il materiale

Pezzo	Quanti	A cosa serve
Wemos D1 mini (ESP8266, chip USB CH340), con i pettini da saldare	1	la scheda
Modulo **GY-291** con l'accelerometro **ADXL345** di Analog Devices, con il pettine da 8 pin da saldare	1	il sensore
Display OLED 0,96", 128 × 64 pixel, I²C, con il pettine già saldato	1	la livella
Breadboard da 830 punti	1	il piano su cui sta tutto
Cavetti maschio-maschio da 20 cm	14	i collegamenti
Cavo USB-A / micro USB	1	programma e alimentazione

Ti servono anche un computer e un saldatore. Per la prova della caduta, un cuscino.

Parola nuova: accelerometro
Un sensore che misura l'accelerazione lungo tre direzioni, gli assi X, Y e Z. Dentro ci sono parti minuscole libere di muoversi, come dice Analog Devices. L'ADXL345 misura sia l'accelerazione costante della gravità, che serve per l'inclinazione, sia quella dei movimenti e degli urti. L'unità è il **g**: 1 g è l'accelerazione di gravità, circa 9,8 m/s².

2 Come si misura l'inclinazione



Fermo, il sensore segna 1 g verso l'alto: è così che «sente» la gravità.
Con tre assi: θ = arctan(X ÷ √(Y² + Z²)) (Analog Devices, AN-1057)

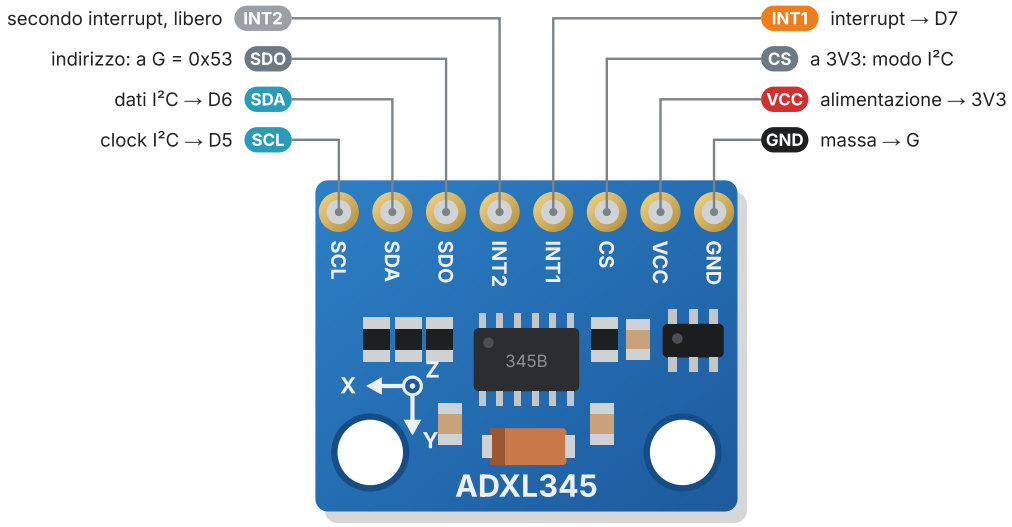
informaticait.it

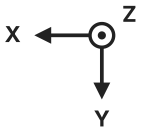
- Appoggiato in piano, con il chip in alto, l'ADXL345 segna **Z = 1 g** e zero su X e Y: lo mostra il datasheet di Analog Devices. Per un accelerometro fermo, la gravità è una spinta di 1 g verso l'alto.
- Inclinandolo, quel g si divide: sull'asse X va **1 g · sin θ**, su Z resta **1 g · cos θ**. Dalle tre letture l'angolo si ricava con la formula della nota applicativa AN-1057 di Analog Devices, **θ = arctan(X ÷ √(Y² + Z²))**, e allo stesso modo quello di Y.
- Perché tre assi e non uno solo? Con un asse solo, verso i 90° la lettura cambia sempre meno e la misura perde sensibilità. Usando il rapporto tra gli assi, spiega la stessa nota, la sensibilità resta circa costante. E se gli assi hanno la stessa sensibilità, gran parte del loro errore si annulla nel rapporto.

Da sapere
- La livella funziona da ferma. Se la muovi, il sensore somma alla gravità il tuo movimento e l'angolo sballa: la nota AN-1057 parte proprio dall'ipotesi che l'unica accelerazione sia quella di gravità.

3 Il modulo GY-291

Il modulo GY-291 visto dal lato dei componenti





Gli assi come sono stampati sul modulo:
X verso sinistra, Y verso il basso,
Z esce dal modulo (verso l'alto, se è in piano).

informaticait.it

- Il chip ADXL345 lavora da **2,0 a 3,6 V** e nessun pin deve superare 3,9 V. Secondo la nostra scheda prodotto il modulo ha un regolatore e si può alimentare da 3,3 a 5 V. Noi lo alimentiamo dal **3V3** della D1 mini: così ogni pin resta a 3,3 V, comunque sia fatto il modulo.
- La scheda prodotto dice anche che sulle linee I²C ci sono già le resistenze di pull-up da 4,7 kΩ (guida n. 20). Sulle nostre foto si vedono tre resistenze con la sigla 472, cioè 4,7 kΩ. Controlla sul tuo modulo.
- **CS** va al 3V3 e **SDO** a massa. Il datasheet è chiaro: dentro il chip non ci sono resistenze che li tengano a un livello, quindi in I²C vanno collegati tutti e due. Con SDO a massa l'indirizzo è **0x53**, con SDO al positivo sarebbe 0x1D.
- **INT1** e **INT2** sono uscite push-pull, alte quando c'è un avviso: vanno dritte a un pin della scheda, senza resistenze. Qui usiamo solo INT1.

> **Parola nuova: interrupt**
> Un filo con cui il sensore avvisa la scheda: «è successo qualcosa». La scheda non deve chiedere di continuo; quando il pin sale, interrompe un attimo quello che sta facendo e prende nota.

4 Salda i pettini

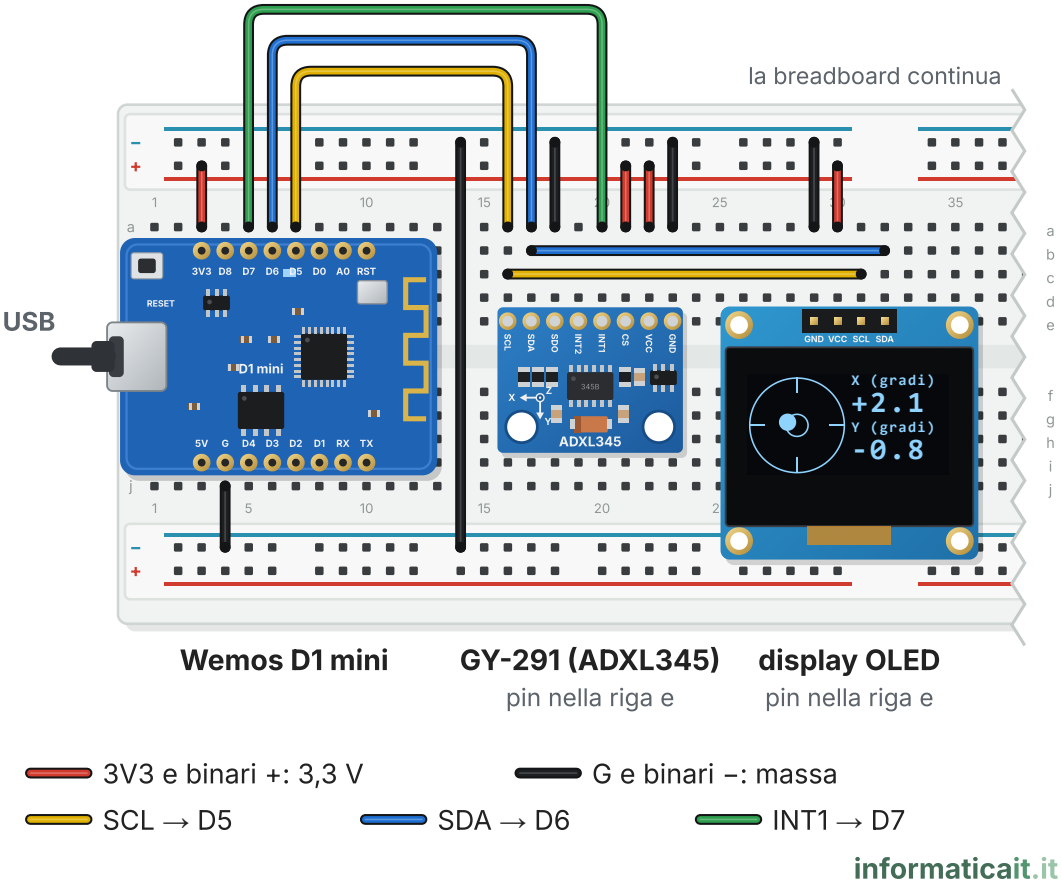
- Sulla D1 mini salda i due pettini **maschi** da 8 pin, con le punte verso il basso; i connettori femmina qui non servono. Il trucco della guida n. 9: infila i pettini nella breadboard, appoggia sopra la scheda e salda, così restano dritti.
- Sul GY-291 fai lo stesso con il pettine da 8: la parte nera sotto il modulo, dal lato senza componenti, le punte lunghe nella breadboard, le saldature dal lato del chip. In tutto sono 24 punti di stagno; come si fa una buona saldatura lo spiega la guida n. 70.
- Il display arriva con il pettine già saldato, con i pin che escono dal retro: si infila nella breadboard con il vetro in alto. Così lo mostrano le nostre foto; se il tuo avesse il pettine sciolto, saldalo nello stesso verso.

Attenzione

- Un pettine saldato storto fa partire la livella già inclinata. La prova del passo 10 corregge anche questo, ma conviene saldare il modulo bene in piano.

5 Monta il circuito

I collegamenti sulla breadboard



- **Stacca la USB** mentre colleghi i fili.

- La D1 mini va a sinistra, con la presa USB fuori dalla breadboard: i pin entrano nelle righe **b** e **i**, colonne da 3 a 10. Nella colonna 3 ci sono 3V3 (in alto) e 5V (in basso). Per ogni pin resta libero un foro: nella riga a per quelli di sopra, nella riga j per quelli di sotto.
- Il GY-291 entra con i pin nella riga **e**, colonne da 16 a 23, chip in alto: SCL nella 16, GND nella 23. Il corpo del modulo resta sospeso sopra la scanalatura centrale.
- Il display entra anche lui nella riga **e**, colonne da 29 a 32 (GND nella 29, SDA nella 32), con il vetro in alto e il corpo verso di te.
- I binari laterali + e – di questa breadboard si interrompono a metà: usiamo solo la metà sinistra.

Cavetto	Da	A
rosso	a3 (3V3)	binario + in alto, colonna 3
nero	j4 (G)	binario – in basso, colonna 4
nero	binario – in basso, colonna 14	binario – in alto, colonna 14
giallo	a7 (D5)	a16 (SCL del GY-291)
blu	a6 (D6)	a17 (SDA del GY-291)
verde	a5 (D7)	a20 (INT1 del GY-291)
nero	a18 (SDO) e a23 (GND)	binario – in alto, stesse colonne
rosso	a21 (CS) e a22 (VCC)	binario + in alto, stesse colonne
giallo	c16	c31 (SCL del display)
blu	b17	b32 (SDA del display)
nero	a29 (GND del display)	binario – in alto, colonna 29
rosso	a30 (VCC del display)	binario + in alto, colonna 30

✓ Controlla prima di andare avanti
Ripassa i 14 fili con il dito sul disegno. Il pin 5V della D1 mini resta libero: qui tutto va a 3,3 V.

Parola nuova: D5 e D6 per l'I²C
Le guide n. 9 e 22 usano D2 e D1, i pin predefiniti. Qui i dati del sensore e del display passano da **D6** (SDA) e **D5** (SCL), come nella guida n. 92: così tutti i fili partono dalla fila di pin di sopra, accanto a D7 e al 3V3. LOLIN dichiara l'I²C su tutti i pin tranne D0, e il programma lo dice con `Wire.begin(D6, D5)`. Per INT1 va bene D7: D3, D4 e D8 decidono come parte la scheda (guida n. 9) e D0 non accetta interrupt.

6 Prepara il computer

- Arduino IDE 2, il driver del chip CH340, il pacchetto per l'ESP8266 e la scheda **LOLIN(WEMOS) D1 R2 & mini**: segui la guida n. 9, capitoli 3 e 4, con la prima prova della scheda da sola.
- Nel **Gestore librerie** installa **Adafruit ADXL345** e **Adafruit SSD1306**. Quando ti propone le librerie che servono a queste (Adafruit Unified Sensor, Adafruit BusIO, Adafruit GFX Library), installale tutte. Noi abbiamo compilato con Adafruit ADXL345 1.3.4 e SSD1306 2.5.17.

7 La prima prova: i numeri nel plotter

Prima della livella, un programma corto per vedere che cosa misura il sensore. Venti volte al secondo manda al computer le tre accelerazioni in m/s².

```
// Prima prova dell'accelerometro ADXL345 (modulo GY-291) con la Wemos D1 mini.
// Scrive venti volte al secondo le tre accelerazioni, in m/s², per il plotter seriale.
// Collegamenti: SDA -> D6, SCL -> D5, modulo dal 3V3, SDO a GND (indirizzo 0x53).
#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>

Adafruit_ADXL345_Unified accel(345);        // il numero è solo un nome per il sensore

void setup() {
  Serial.begin(115200);
  Wire.begin(D6, D5);                       // SDA su D6, SCL su D5
  if (!accel.begin(0x53)) {                 // 0x53: l'indirizzo con SDO collegato a GND
    Serial.println("ADXL345 non trovato: controlla i fili");
    while (true) delay(1000);
  }
  accel.setRange(ADXL345_RANGE_16_G);       // fino a 16 g, sempre con 4 mg per passo
  accel.setDataRate(ADXL345_DATARATE_100_HZ);
}

void loop() {
  sensors_event_t misura;
  accel.getEvent(&misura);                  // le tre accelerazioni in m/s²
  Serial.print("x:");                       // "nome:valore" dà il nome alla linea
  Serial.print(misura.acceleration.x);
  Serial.print(",y:");
  Serial.print(misura.acceleration.y);
  Serial.print(",z:");
  Serial.println(misura.acceleration.z);
  delay(50);
}
```

- Il campo ±16 g non toglie precisione: la libreria accende la piena risoluzione, e in quel modo il passo resta di circa 4 mg in tutti i campi, dice il datasheet. Nel campo più alto c'è posto anche per i colpi.

- Carica il programma, poi apri il **Plotter seriale** con il pulsante in alto a destra dell'IDE, a 115200 baud. Il pezzo `x:` prima del numero dà il nome alla linea.

✓ Cosa devi vedere
Con la breadboard in piano, x e y restano vicino a 0 e z intorno a 9,8. Alza il lato sinistro della breadboard: x sale. Alza il lato verso di te: sale y. Girala sottosopra e z diventa negativa.

Se non funziona
- «ADXL345 non trovato»: SDA e SCL scambiati (giallo e blu), CS o SDO non collegati, o il modulo non ha il 3,3 V.
- x scende quando alzi il lato sinistro: il tuo modulo è montato girato. Nel programma della livella lo sistemi con `VERSO_X` e `VERSO_Y`.
- Un colpetto sul tavolo si vede poco: il programma legge venti volte al secondo e un urto secco può cadere tra due letture. Al tocco ci pensa l'interrupt, al passo 11.

8 La livella

Ora il programma vero: angoli, bolla sul display, tocco e caduta. Nelle prime righe ci sono i valori che puoi cambiare.

```
// La livella elettronica: Wemos D1 mini, accelerometro ADXL345 (GY-291) e display OLED.
// Ricava dalla gravità l'inclinazione sui due assi, disegna una livella a bolla e, con
// l'interrupt INT1, riconosce un tocco e la caduta libera.
// Collegamenti: SDA -> D6, SCL -> D5, INT1 -> D7; moduli alimentati dal 3V3.
#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

const float ZERO_X = 0.0;          // correzioni in gradi: le trovi con la prova del giro
const float ZERO_Y = 0.0;
const int VERSO_X = -1;            // la X del GY-291 punta a sinistra: bolla al contrario
const int VERSO_Y = 1;             // la Y punta verso di te
const float PIXEL_GRADO = 3.0;     // un grado sposta la bolla di 3 pixel
const float LISCIA = 0.2;          // più è piccolo, più la bolla è calma (e lenta)
const float IN_BOLLA = 0.5;        // sotto mezzo grado sui due assi: "in bolla"

Adafruit_ADXL345_Unified accel(345);
Adafruit_SSD1306 display(128, 64, &Wire, -1);

volatile bool avviso = false;      // lo mette a true l'interrupt
float angoloX = 0, angoloY = 0;
int tocchi = 0, cadute = 0;
String messaggio = "";
unsigned long fineMessaggio = 0;
```

```
void IRAM_ATTR suInt1() {          // gira quando INT1 sale: alza solo una bandierina
  avviso = true;
}

void preparaInterrupt() {
  accel.writeRegister(ADXL345_REG_INT_ENABLE, 0);     // prima tutto spento
  accel.writeRegister(ADXL345_REG_THRESH_TAP, 0x38);  // tocco: oltre 3,5 g (62,5 mg x 56)
  accel.writeRegister(ADXL345_REG_DUR, 0x20);         // per meno di 20 ms (625 µs x 32)
  accel.writeRegister(ADXL345_REG_TAP_AXES, 0x07);    // su tutti e tre gli assi
  accel.writeRegister(ADXL345_REG_THRESH_FF, 0x07);   // caduta: sotto 0,44 g (62,5 x 7)
  accel.writeRegister(ADXL345_REG_TIME_FF, 0x14);     // per almeno 100 ms (5 ms x 20)
  accel.writeRegister(ADXL345_REG_INT_MAP, 0);        // tutti gli avvisi sul pin INT1
  accel.writeRegister(ADXL345_REG_INT_ENABLE, 0x44);  // tocco (0x40) e caduta (0x04)
}

void scrivi(int x, int y, int grandezza, String testo) {
  display.setTextSize(grandezza);
  display.setCursor(x, y);
  display.print(testo);
}

String gradi(float a) {            // +1.2 oppure -0.4: sempre con il segno
  return (a >= 0 ? "+" : "") + String(a, 1);
}

void mostra(String testo) {        // una scritta grande per un secondo e mezzo
  messaggio = testo;
  fineMessaggio = millis() + 1500;
}

void disegna() {
  display.clearDisplay();
  if (millis() < fineMessaggio) {
    scrivi((128 - messaggio.length() * 12) / 2, 12, 2, messaggio);
    scrivi(10, 44, 1, "tocchi " + String(tocchi) + "  cadute " + String(cadute));
  } else {
    display.drawCircle(31, 32, 30, SSD1306_WHITE);          // il vetro della livella
    display.drawCircle(31, 32, 7, SSD1306_WHITE);           // il cerchio del centro
    display.drawFastHLine(1, 32, 10, SSD1306_WHITE);        // quattro tacche
    display.drawFastHLine(52, 32, 10, SSD1306_WHITE);
    display.drawFastVLine(31, 2, 10, SSD1306_WHITE);
    display.drawFastVLine(31, 53, 10, SSD1306_WHITE);
    float dx = constrain(VERSO_X * angoloX * PIXEL_GRADO, -24.0f, 24.0f);
    float dy = constrain(VERSO_Y * angoloY * PIXEL_GRADO, -24.0f, 24.0f);
    display.fillCircle(31 + (int)dx, 32 + (int)dy, 5, SSD1306_WHITE);   // la bolla
    scrivi(66, 0, 1, "X (gradi)");
    scrivi(66, 10, 2, gradi(angoloX));
    scrivi(66, 30, 1, "Y (gradi)");
    scrivi(66, 40, 2, gradi(angoloY));
    if (fabs(angoloX) < IN_BOLLA && fabs(angoloY) < IN_BOLLA) {
      scrivi(66, 56, 1, "in bolla");
    }
  }
  display.display();               // manda la schermata al display
}

void setup() {
```

```
  Serial.begin(115200);
  Wire.begin(D6, D5);              // SDA su D6, SCL su D5
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.setTextColor(SSD1306_WHITE);
  if (!accel.begin(0x53)) {        // 0x53: l'indirizzo con SDO collegato a GND
    Serial.println("ADXL345 non trovato: controlla i fili");
    display.clearDisplay();
    scrivi(0, 0, 1, "ADXL345 non trovato");
    display.display();
    while (true) delay(1000);
  }
  accel.setRange(ADXL345_RANGE_16_G);         // fino a 16 g, sempre con 4 mg per passo
  accel.setDataRate(ADXL345_DATARATE_100_HZ);
  preparaInterrupt();
  pinMode(D7, INPUT);
  attachInterrupt(digitalPinToInterrupt(D7), suInt1, RISING);
  accel.readRegister(ADXL345_REG_INT_SOURCE); // cancella gli avvisi rimasti in sospeso
}

void loop() {
  sensors_event_t m;
  accel.getEvent(&m);                          // le tre accelerazioni in m/s²
  float x = m.acceleration.x, y = m.acceleration.y, z = m.acceleration.z;
  // angoli rispetto all'orizzonte: Analog Devices AN-1057, equazioni 11 e 12
  float ax = degrees(atan2(x, sqrt(y * y + z * z))) - ZERO_X;
  float ay = degrees(atan2(y, sqrt(x * x + z * z))) - ZERO_Y;
  angoloX += LISCIA * (ax - angoloX);          // un passo verso il valore nuovo
  angoloY += LISCIA * (ay - angoloY);

  int evento = 0;
  if (avviso) {                                // l'ADXL345 ha alzato INT1
    avviso = false;
    uint8_t fonte = accel.readRegister(ADXL345_REG_INT_SOURCE);   // leggerlo spegne INT1
    if (fonte & 0x04) {
      cadute++;
      evento = -5;
      mostra("CADUTA!");
    } else if ((fonte & 0x40) && millis() > fineMessaggio) {
      tocchi++;
      evento = 5;
      mostra("TOCCO!");
    }
  }
  disegna();
  Serial.printf("angoloX:%.2f,angoloY:%.2f,evento:%d\n", angoloX, angoloY, evento);
  delay(20);
}
```

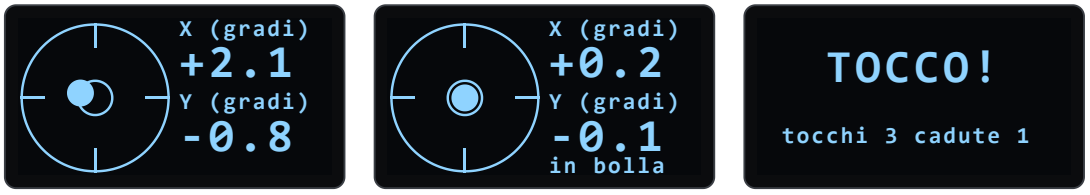
9 Come funziona la livella

- A ogni giro `loop()` legge le tre accelerazioni e calcola i due angoli con le equazioni 11 e 12 della nota AN-1057. `degrees()` li porta da radianti a gradi.
- Le letture tremano un poco. `LISCIA` fa fare all'angolo mostrato solo un quinto della strada verso il valore nuovo: la bolla si calma, ma risponde un po' più piano.

- La bolla si sposta di 3 pixel per grado, quindi arriva al bordo a 8° (calcolo nostro: 24 pixel diviso 3). Sul GY-291 la X punta a sinistra: se alzi il lato sinistro l'angolo X diventa positivo e la bolla, che va sempre verso il lato più alto, deve andare a sinistra. Per questo `VERSO_X` vale -1.
- «in bolla» compare sotto mezzo grado su tutti e due gli assi: è una soglia nostra, la cambi in `IN_BOLLA`.
- Al computer arrivano i due angoli e un terzo numero, `evento`: vale 5 nella riga di un tocco, -5 in quella di una caduta, 0 nelle altre. Nel plotter seriale diventa un picco ben visibile.

Il display della livella

valori di esempio



1 · il lato sinistro è più alto
la bolla va a sinistra

2 · in piano
meno di mezzo grado

3 · un tocco sul piano
la scritta resta 1,5 s

informaticait.it

✓ Cosa devi vedere

Sul display c'è il cerchio con la bolla e a destra i due angoli. Inclina la breadboard: la bolla corre verso il lato più alto. Nel plotter seriale vedi le linee `angoloX` e `angoloY`.

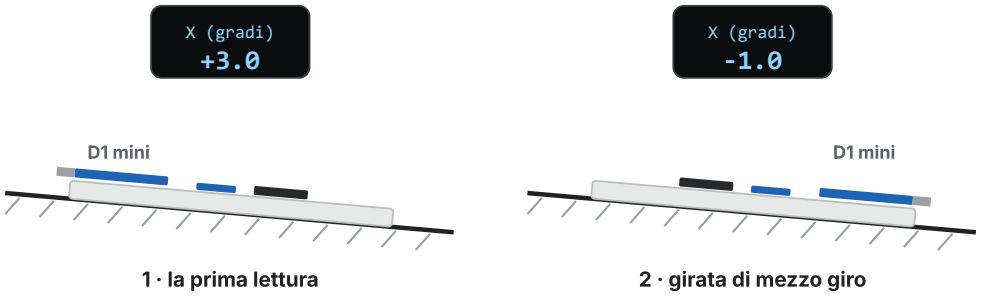
Se non funziona

- Display nero: il display risponde a 0x3C? Alcuni rispondono a 0x3D (guida n. 9). Controlla anche VCC e GND nelle colonne 29 e 30.
- La bolla scappa dalla parte sbagliata: cambia il segno di `VERSO_X` o di `VERSO_Y`.
- In piano non segna zero: è normale, lo sistema la prova del passo 10.

10 La prova del giro: lo zero

Appena montata, la livella quasi mai segna zero su un piano perfetto. Il datasheet dà per l'uscita a 0 g degli assi X e Y un valore tra -150 e +150 mg, cioè fino a circa 8,6° (calcolo nostro: arcoseno di 0,15). Analog Devices aggiunge che saldatura e montaggio spostano lo zero e consiglia di tararlo dopo il montaggio. E poi c'è il pettine, che non è mai saldato perfettamente in piano.

La prova del giro: due letture, mezzo giro in mezzo



1 · la prima lettura **2 · girata di mezzo giro**

inclinazione della mensola = (prima − seconda) ÷ 2 = (3,0 − (−1,0)) ÷ 2 = 2,0°
errore della livella = (prima + seconda) ÷ 2 = (3,0 + (−1,0)) ÷ 2 = 1,0° → ZERO_X = 1.0

Valori di esempio; la mensola è disegnata più inclinata del vero.

informaticait.it

- Appoggia la livella su una mensola o sul tavolo e segna con il nastro di carta dove sta. Aspetta che i numeri si fermino e scrivi X e Y.
- Girala di mezzo giro nello stesso posto, con la presa USB dall'altra parte, e scrivi di nuovo X e Y.
- Per ogni asse fai la somma delle due letture e dividi per 2: è l'errore della livella. Scrivilo in `ZERO_X` e `ZERO_Y` e ricarica il programma. La differenza diviso 2 è invece l'inclinazione vera della mensola.

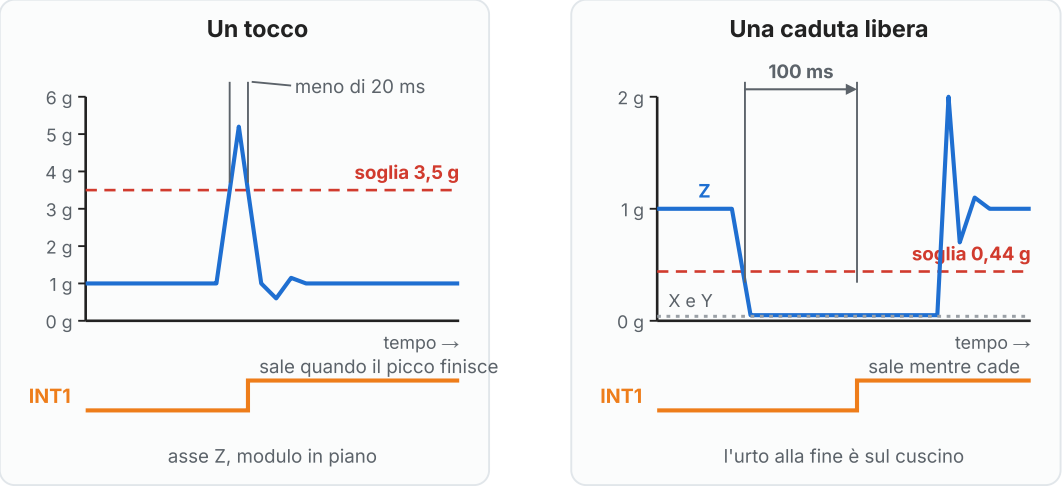
È un metodo nostro, con la stessa idea della taratura a due punti della nota AN-1057: l'errore resta uguale, l'inclinazione cambia segno, e la media delle due letture lascia solo l'errore. Vale per inclinazioni piccole, di qualche grado.

11 Tocco e caduta libera con l'interrupt

L'ADXL345 sa riconoscere da solo un colpetto e la caduta libera. Si impostano le soglie nei suoi registri, e quando succede qualcosa alza il pin INT1. La funzione `preparaInterrupt()` scrive questi valori; le unità sono quelle del datasheet.

Registro	Valore	Che cosa vuol dire
THRESH_TAP	0x38	soglia del tocco: 56 × 62,5 mg = 3,5 g
DUR	0x20	il tocco deve durare meno di 32 × 625 µs = 20 ms
TAP_AXES	0x07	il tocco vale su X, Y e Z
THRESH_FF	0x07	soglia della caduta: 7 × 62,5 mg ≈ 0,44 g
TIME_FF	0x14	sotto soglia per almeno 20 × 5 ms = 100 ms
INT_MAP	0x00	tutti gli avvisi sul pin INT1
INT_ENABLE	0x44	accesi il tocco singolo (0x40) e la caduta libera (0x04)

Quando l'ADXL345 alza INT1



Disegno schematico, non una misura: le soglie sono quelle dello sketch.

informaticait.it

- Per la caduta Analog Devices consiglia una soglia tra 300 e 600 mg e un tempo tra 100 e 350 ms: siamo dentro. Per il tocco indica come punto di partenza una durata oltre 10 ms e una soglia oltre 3 g, e avverte che ogni oggetto risponde a modo suo: qualche prova va fatta.
- La funzione `suInt1()` parte da sola quando INT1 sale. Il core ESP8266 vuole `IRAM_ATTR` davanti; lei alza solo una bandierina, `avviso`, e il resto lo fa `loop()`.
- `loop()` legge il registro **INT_SOURCE**, che dice che cosa è successo. Leggerlo rimette INT1 a riposo, pronto per il prossimo avviso.

✓ Cosa devi vedere

Dai un colpetto secco sulla breadboard accanto al sensore: per un secondo e mezzo compare «TOCCO!». Poi tieni la breadboard ferma sopra un cuscino, a 20 o 30 cm, con il cavo USB lento, e lasciala andare: compare «CADUTA!». Da 20 cm la caduta dura circa 0,2 secondi (calcolo nostro, t = √(2h ÷ g)), più dei 100 ms richiesti.

Se non funziona

- Il tocco non viene contato: batti più deciso. Puoi abbassare la soglia, ma con valori molto bassi, dice il datasheet, il sensore prende anche gli echi del colpo.
- La caduta non viene vista: il cavo USB tirava, o la caduta era troppo corta. Dopo la prova controlla che i moduli siano ancora ben infilati.
- Non succede niente in nessun caso: il filo verde va da a5 (D7) ad a20 (INT1).

Prova tu

- Metti `PIXEL_GRADO` a 10: la bolla arriva al bordo a 2,4° e la livella diventa più fine.
- Abbassa `LISCIA` a 0.05 e guarda nel plotter come cambia la risposta.
- Accendi anche il doppio tocco (0x20 in INT_ENABLE): servono i registri Latent e Window, e il datasheet suggerisce di partire da oltre 20 ms e oltre 80 ms.

Parola nuova: MPU-6050

Nel nostro catalogo c'è anche il modulo GY-521 (127) con l'MPU-6050 di InvenSense: un accelerometro a tre assi più un giroscopio, che misura quanto velocemente il modulo ruota, da ±250 a ±2000 gradi al secondo. Il chip lavora da 2,375 a 3,46 V. Per una livella ferma basta l'accelerometro; il giroscopio serve quando l'oggetto si muove, come un robot o un drone.

✓ Cosa hai imparato

- fermo, un accelerometro misura la gravità, e da come si divide tra gli assi si ricava l'inclinazione;
- con tre assi la formula di AN-1057 resta sensibile a tutti gli angoli;
- lo zero si corregge con due letture e mezzo giro in mezzo;
- il sensore riconosce tocco e caduta da solo e avvisa la scheda con l'interrupt.

Trovi i componenti su informaticait.it

- Modulo GY-291 ADXL345 accelerometro digitale di precisione a 3 assi I2C SPI
- Wemos d1 mini v3.0.0 wifi ch340g 4mb compatibile con IDE Arduino Nodemcu...
- Display oled lcd led 0.96" 128×64 12864 I2C IIC 4 pin ssd1306 BLU Arduino
- Piastra sperimentale breadboard MB-102 830 punti 16,5×5,5cm trasparente
- 40 ponticelli cavi cavetti flessibili dupont maschio maschio 20 cm per...
- Cavo USB-A / micro USB 1 metro in nylon intrecciato per ricarica e dati

Clicca sul nome per aprire la scheda prodotto. Disponibilità e prezzi aggiornati sul sito.

Fonti: Analog Devices, datasheet ADXL345 (caratteristiche, interfaccia I²C, interrupt, registri, rilevamento del tocco, taratura dell'offset, assi di sensibilità) · Analog Devices, nota applicativa AN-1057 «Using an Accelerometer for Inclination Sensing» (C. J. Fisher) · LOLIN (Wemos), D1 mini v3.1.0: tabella dei pin e caratteristiche · ESP8266 Arduino Core, documentazione: riferimento (interrupt sui pin) e librerie (Wire) · Adafruit, librerie Adafruit ADXL345 1.3.4, Unified Sensor, SSD1306 2.5.17 e GFX · Arduino, Arduino IDE 2: Serial Plotter · InvenSense, MPU-6000/MPU-6050 Product Specification rev. 3.4 · Schede prodotto informaticait 400, 2190, 314 e 127; guide n. 9, 20, 22, 70 e 92