

Una tensione vera: il DAC MCP4725 a 12 bit

Il modulo Adafruit 935 dà alla FireBeetle 2 ESP32-E, o ad Arduino UNO, un'uscita analogica vera: 4096 livelli comandati in I²C, da misurare con il tester e da guardare sul Plotter seriale.

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



Guida online,
sempre
aggiornata

IN BREVE

- L'MCP4725 di Microchip trasforma un numero da 0 a 4095 in una tensione ferma, non a impulsi come il PWM.
- Il riferimento è l'alimentazione: $V_{OUT} = V_{DD} \times \text{codice} \div 4096$. A 3,3 V un passo vale circa 0,8 mV.
- Si comanda in I²C all'indirizzo 0x62 con la libreria Adafruit MCP4725; con l'ESP32 il modulo va al 3V3.
- Con `setVoltage(codice, true)` il valore resta nella EEPROM e il chip lo ritrova a ogni accensione.

Tempo: un'ora

Livello: un passo avanti

Prima: guide n. 8, 14 e 22

Per leggere una tensione le schede hanno l'ADC; per produrne una, quasi sempre no. Il PWM della guida n. 37 accende e spegne il pin molto in fretta e il tester ci vede una media, ma la tensione vera salta sempre tra zero e il massimo. Un **DAC** invece tiene ferma sull'uscita la tensione che gli chiedi. L'MCP4725 di Microchip lo fa con **4096 livelli**, riceve gli ordini in I²C e si ricorda l'ultimo valore salvato anche da spento.

1 Il materiale

- il modulo **Adafruit 935** con l'MCP4725: ha un pettine a sei pin sciolto, da saldare. Adafruit avverte che appoggiarlo nei fori senza saldatura non funziona (guida n. 70);
- la **FireBeetle 2 ESP32-E** con i pin già saldati, cinque cavetti femmina-femmina;
- un **tester** (guida n. 8). In alternativa alla FireBeetle va bene **Arduino UNO**, passo 8.

Parola nuova: DAC

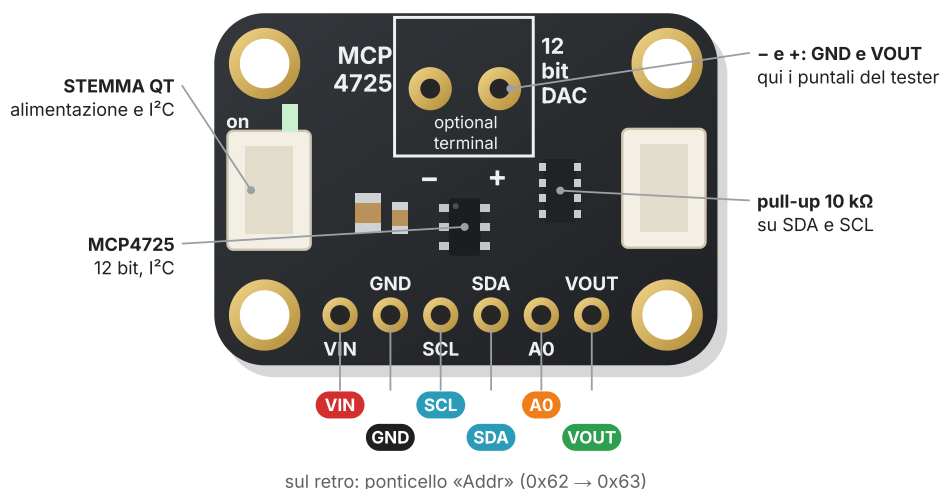
Digital-to-Analog Converter, convertitore da digitale ad analogico: riceve un numero e lo trasforma in una tensione. Fa il contrario dell'ADC, che misura una tensione e la trasforma in un numero (guida n. 13).

2 Il modulo

- **VIN** e **GND**: alimentazione, da 2,7 a 5,5 V per il chip. Adafruit consiglia la stessa tensione della scheda che lo comanda: le resistenze di pull-up da 10 kΩ portano SDA e SCL proprio a VIN. Con l'ESP32 quindi **3,3 V**.
- **SCL** e **SDA**: il bus I²C (guida n. 22). L'indirizzo è **0x62**; diventa 0x63 portando **A0** a VIN o chiudendo il ponticello «Addr» sul retro. Così sullo stesso bus ci stanno due moduli.

- **VOUT**: l'uscita. Gli stessi segnali, VOUT escluso, passano anche dai due connettori STEMMA QT.

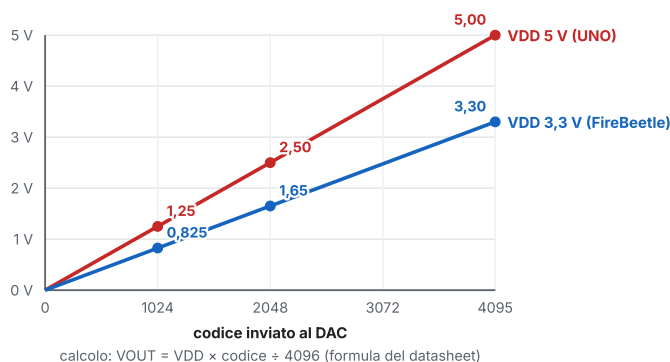
Il modulo Adafruit 935: sei pin e due connettori STEMMA QT



informaticait.it

3 Quanto vale un passo

La tensione d'uscita segue l'alimentazione



informaticait.it

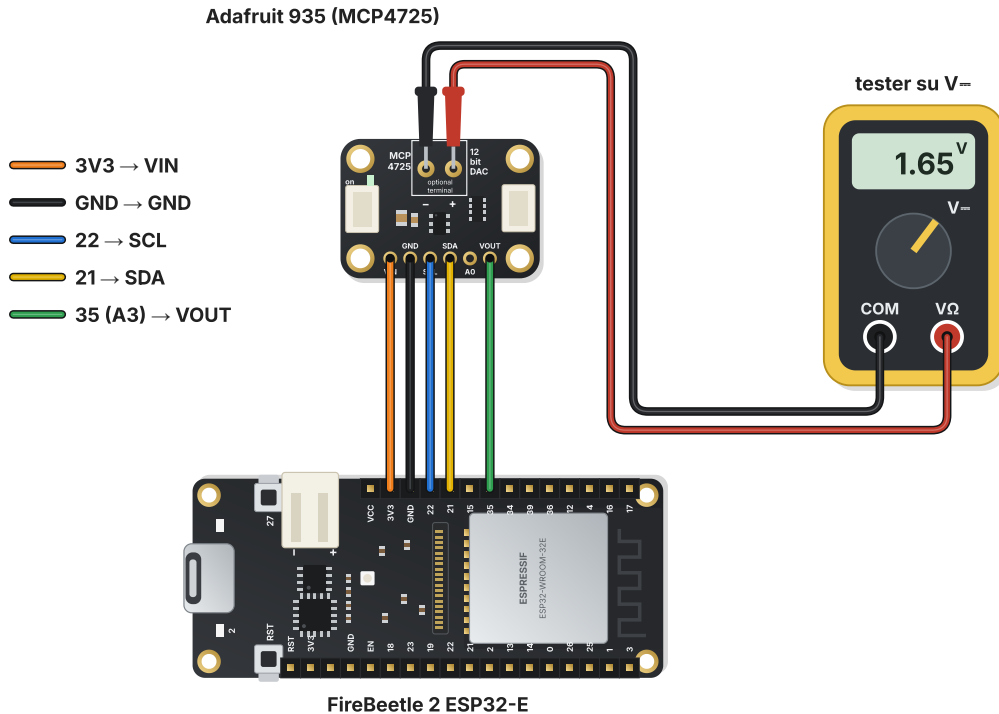
Il chip non ha un riferimento suo: usa l'alimentazione. La formula del datasheet è $VOUT = VDD \times \text{codice} \div 4096$, con il codice da 0 a 4095. Se VDD cambia, cambia anche l'uscita.

- A 5 V un passo vale **1,22 mV** (tabella del datasheet); a 3,3 V circa 0,8 mV (calcolo nostro).
- Con 4095 l'uscita arriva a un passo sotto VDD, non a VDD.
- Vicino agli estremi la precisione cala: Microchip misura la linearità tra i codici **100 e 4000** e l'errore di zero non è tarato in fabbrica.

4 Collega e misura

- Cinque fili dritti: **3V3** → **VIN**, **GND** → **GND**, **22** → **SCL**, **21** → **SDA**, **35** → **VOUT**. Il 35 è anche **A3**, un ingresso analogico: così la FireBeetle rilegge la tensione che ha chiesto.
- VIN al **3V3**, mai al VCC: lì arrivano i circa 5 V dell'USB, le pull-up porterebbero a 5 V anche SDA e SCL, e i pin dell'ESP32 non li reggono (guida n. 21).
- Tester su **V**: puntale nero nel foro **-** e rosso nel foro **+** del morsetto opzionale (sono GND e VOUT), oppure direttamente sui pin GND e VOUT.

I collegamenti: cinque fili dritti e il tester



informaticait.it

✓ Una prima prova

Il tester segna qualcosa già prima dello sketch: all'accensione il chip riprende il codice salvato nella EEPROM, che di fabbrica è la metà scala (datasheet). Se nessuno l'ha cambiato, vedi circa 1,65 V.

5 Quattro tensioni

```
// Il DAC MCP4725: quattro tensioni, una ogni 4 secondi. Misurale con il tester
// tra VOUT e GND; A3 le rilegge. Libreria: Adafruit MCP4725.
#include <Arduino.h>
#include <Adafruit_MCP4725.h>

const byte INDIRIZZO = 0x62; // Adafruit 935: 0x62 (0x63 con il ponticello chiuso)
#ifdef ESP32
const float VDD = 3.3; // FireBeetle 2 ESP32-E: modulo al 3V3
#else
const float VDD = 5.0; // Arduino UNO: modulo al 5V
#endif
const int CODICI[] = {0, 1024, 2048, 4095}; // da 0 a 4095: 12 bit

Adafruit_MCP4725 dac;

void setup() {
  Serial.begin(115200);
  if (!dac.begin(INDIRIZZO)) {
    Serial.println(F("MCP4725 non risponde: controlla i fili e l'indirizzo"));
    while (true) delay(1000);
  }
}
```

```

    }
}

void loop() {
    for (int codice : CODICI) {
        dac.setVoltage(codice, false);           // false: la EEPROM non si tocca
        delay(50);
        Serial.print(F("codice "));
        Serial.print(codice);
        Serial.print(F(" circa "));
        Serial.print(codice * VDD / 4096, 2);    // la formula del datasheet
        Serial.print(F(" V A3: "));
#ifdef ESP32
        Serial.print(analogReadMilliVolts(A3)); // millivolt
        Serial.println(F(" mV"));
#else
        Serial.println(analogRead(A3));          // da 0 a 1023: circa codice / 4
#endif
        delay(4000);
    }
}

```

Installa la libreria **Adafruit MCP4725**. `dac.begin(0x62)` risponde false se il chip non si fa sentire; `setVoltage(codice, false)` cambia l'uscita senza toccare la EEPROM. Apri il Monitor seriale a 115200 baud: per ogni codice c'è la tensione attesa e quella letta da A3.

✓ Cosa devi vedere

Ogni quattro secondi il tester passa da quasi 0 a circa 0,83, 1,65 e 3,3 V (calcolo con VDD a 3,3 V), e il Monitor seriale scrive numeri vicini.

Se non funziona

- «MCP4725 non risponde»: SDA e SCL scambiati, VIN non collegato, oppure l'indirizzo è 0x63 perché il ponticello «Addr» è chiuso.
- Il tester segna un po' meno del previsto: misura il 3V3 della scheda e rifai il conto con quel valore al posto di 3,3.
- A3 e il tester non coincidono: l'ADC dell'ESP32 è meno preciso di un tester (guida n. 14). Fidati del tester.

6 Un'onda sul Plotter

```

// Un'onda dal DAC MCP4725 e quello che rilegge A3, insieme sul Plotter seriale.
#include <Arduino.h>
#include <Adafruit_MCP4725.h>

const byte INDIRIZZO = 0x62; // Adafruit 935
const bool SEN0 = false;     // true: onda sinusoidale; false: triangolare
const int PUNTI = 32;        // punti in ogni periodo

Adafruit_MCP4725 dac;
int punto = 0;

```

```

void setup() {
  Serial.begin(115200);
  if (!dac.begin(INDIRIZZO)) {
    Serial.println(F("MCP4725 non risponde: controlla i fili e l'indirizzo"));
    while (true) delay(1000);
  }
}

void loop() {
  int codice;
  if (SENO) {
    codice = 2048 + 2047 * sin(TWO_PI * punto / PUNTI);
  } else {
    // sale per metà periodo, poi scende
    int meta = PUNTI / 2;
    int salita = punto < meta ? punto : PUNTI - punto;
    codice = salita * 4095L / meta;
  }
  dac.setVoltage(codice, false);
  delay(2);
#ifdef ESP32
  Serial.print(F("Inviato_mV:"));
  Serial.print(codice * 3300L / 4096);           // modulo al 3V3
  Serial.print(F(",A3_mV:"));
  Serial.println(analogReadMilliVolts(A3));
#else
  Serial.print(F("Inviato:"));
  Serial.print(codice / 4);                     // in punti di A3: 1023 = 5 V
  Serial.print(F(",A3:"));
  Serial.println(analogRead(A3));
#endif
  punto = (punto + 1) % PUNTI;
  delay(40);
}

```

Ogni 40 millesimi di secondo circa il DAC fa un passo: 32 punti per salire e scendere. Apri il **Plotter seriale** a 115200 baud. Con `SENO = true` l'onda diventa una sinusoide, centrata su metà alimentazione. Il Plotter dell'IDE 2 mostra le ultime 50 righe, poco più di un'onda e mezza.

✓ Cosa devi vedere

Due triangoli uno sull'altro: `Inviato_mV` è la tensione chiesta, `A3_mV` quella letta. Il tester, che è lento, segue l'onda a fatica: è normale.

7 Un valore per l'accensione

Con `dac.setVoltage(2048, true)` il codice va anche nella **EEPROM**, e il chip lo ricarica a ogni accensione: l'uscita parte subito da quel valore, prima ancora che lo sketch giri. Comodo per una tensione di riferimento che deve esserci sempre.

Attenzione

- Una scrittura nella EEPROM dura fino a **50 ms** (datasheet) e le scritture non sono infinite: falla una volta, nel setup o alla pressione di un pulsante, mai nel loop.

8 Con Arduino UNO

- VIN al **5V**, GND al GND, SCL e SDA ai pin **SCL** e **SDA** accanto ad AREF, VOUT ad **A3**. L'uscita va da 0 a 5 V.
- Lo sketch è lo stesso: si compila per l'UNO e stampa la lettura di A3 da 0 a 1023. DAC e ADC dell'UNO prendono come riferimento gli stessi 5 V, quindi la lettura vale circa **codice ÷ 4** anche se la porta USB dà un po' meno di 5 V (calcolo nostro).
- Senza cavetti: il cavo **STEMMA QT** porta alimentazione e I²C, con spinotti maschio per l'UNO e con prese femmina per la FireBeetle. Rosso V+, nero GND, blu SDA, giallo SCL; VOUT resta sul pettine.

Prova tu

- Una rampa lenta, da 0 a 4095 in dieci secondi, da seguire con il tester.
- Leggi un potenziometro (guida n. 36) e copia la sua posizione sul DAC.
- Misura l'uscita con 0 e con 4095: quanto manca da 0 V e da VDD?

✓ Cosa hai imparato

- un DAC dà una tensione ferma, non a impulsi;
- $VOUT = VDD \times \text{codice} \div 4096$: il riferimento è l'alimentazione;
- con `true` il valore resta nella EEPROM, ma va scritto di rado.

Trovi i componenti su informaticait.it

- | | |
|---|--|
| • Convertitore digitale-analogico DAC MCP4725 12 bit, I2C | • Multimetro digitale UNI-T UT131B |
| • Scheda FireBeetle 2 ESP32-E con pin saldati | • Cavo STEMMA QT/Qwiic JST SH 4 pin a connettori femmina passo 2,54 mm |
| • Scheda Arduino Uno Rev3 originale | • Cavo STEMMA QT/Qwiic JST SH 4 pin a connettori maschio passo 2,54 mm |

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

Fonti: Microchip, MCP4725 12-Bit Digital-to-Analog Converter with EEPROM Memory, datasheet DS20002039F (2023; valori uguali alla DS22039D) · Adafruit, MCP4725 Breakout Board (935): pagina prodotto, guida «MCP4725 12-Bit DAC Tutorial», schema e file Eagle della versione STEMMA QT; libreria Adafruit_MCP4725 2.0.2 · DFRobot, FireBeetle 2 ESP32-E (DFR0654): wiki e disegno quotato · Arduino, riferimento del linguaggio: `analogRead()` e `analogWrite()`; Arduino IDE 2, Plotter seriale · Guide informaticait n. 8, 13, 14, 22, 36, 37 e 70