

# Vending Machine IoT Monitor (Full-Stack Project)

Platform STM32 Nucleo | Android

Lang C++ | Kotlin

Connectivity Bluetooth Low Energy

Status Completed

Un sistema IoT completo che trasforma una scheda **STM32 Nucleo** in una **Vending Machine intelligente**, controllata e monitorata in tempo reale tramite un'**App Android** dedicata.

Il progetto dimostra la comunicazione bidirezionale via **BLE (Bluetooth Low Energy)**: l'App non solo legge i dati dei sensori, ma invia comandi per selezionare prodotti e gestire il sistema.

## ◆ Funzionalità Principali

### ⌚ Hardware & Firmware (STM32)

- **Smart Dispensing:** Gestione di 4 prodotti con prezzi differenziati e feedback visivo RGB dedicato.
- **Macchina a Stati (FSM):** Logica robusta con stati di *Eco-Mode, Attesa, Erogazione, Resto ed Errore*.
- **Sensori:**
  - **Monete:** Simulazione inserimento tramite sensore di luce (LDR).
  - **Ambiente:** Monitoraggio Temperatura e Umidità (DHT11).
  - **Presenza:** Attivazione automatica display all'avvicinarsi dell'utente (Ultrasuoni HC-SR04).
- **Attuatori:** Display LCD I2C, Servo Motore, Buzzer e LED RGB.
- **Automazione:** Timeout automatico (30s) per il resto se l'utente non completa l'acquisto.

### 📱 Android App (Kotlin + Compose)

- **Dashboard "Dark Tech":** Interfaccia moderna divisa in *Area Clienti* (Acquisto) e *Area Diagnostica* (Sensori).
- **Controllo Remoto:** Selezione prodotti e richiesta rimborso/annullamento direttamente da smartphone.
- **BLE Management:** Scansione automatica, gestione permessi (Android 12+) e riconnessione fluida.

## 🛠️ Architettura Hardware

Pinout (Nucleo F401RE + IDB05A2 Shield)

Componente	Pin Nucleo	Funzione
BLE Shield	Morpho	X-NUCLEO-IDB05A2 (SPI)
LCD I2C	D14 (SDA), D15 (SCL)	Interfaccia Utente (Indirizzo 0x4E/0x27)
Sensore Ultrasuoni	A1 (Trig), D9 (Echo)	Rilevamento presenza (Interrupt Safe)
Sensore LDR	A2 (Analog)	Rilevamento passaggio monete

Componente	Pin Nucleo	Funzione
<b>Sensore DHT11</b>	D4	Temperatura & Umidità
<b>Servo Motore</b>	D5 (PWM)	Meccanismo erogazione
<b>Buzzer</b>	D2	Feedback acustico
<b>LED RGB</b>	D6 (R), D8 (G), A3 (B)	Stato e Colore Prodotto
<b>Tasto Utente</b>	PC_13 (Button Blu)	Annullamento manuale locale

## 🔧 Architettura Hardware

Il sistema utilizza una scheda Nucleo F401RE con shield BLE IDB05A2.

Tutti i sensori e attuatori sono collegati tramite breadboard.

👉 [Clicca qui per lo Schema Elettrico Completo e Guida al Cablaggio \(WIRING.md\)](#)

### Componenti Principali

- **Display:** LCD 16x2 I2C
- **Sensori:** DHT11, HC-SR04, LDR
- **Attuatori:** Servo SG90, LED RGB, Buzzer

## 📡 Protocollo di Comunicazione (GATT)

Il sistema espone un **Custom Service** con UUID: **0000A000-0000-1000-8000-00805f9b34fb**.

### Caratteristiche (Characteristics)

Nome	UUID	Tipo	Descrizione
<b>Temperatura</b>	<b>0xA001</b>	<b>NOTIFY</b>	Invia la temperatura in °C (Int32 LittleEndian).
<b>Stato Sistema</b>	<b>0xA002</b>	<b>NOTIFY</b>	Byte Array [2]: [0]= <b>Credito</b> , [1]= <b>ID_Stato</b> .
<b>Umidità</b>	<b>0xA003</b>	<b>NOTIFY</b>	Invia l'umidità in % (Int32 LittleEndian).
<b>Comandi</b>	<b>0xA004</b>	<b>WRITE_NO_RESP</b>	Canale per inviare comandi dall'App alla Scheda.

### Tabella Comandi (App -> Nucleo)

Scrivendo un byte sulla caratteristica **0xA004**, si controlla la macchina:

Byte	Azione	Prezzo	Colore LED
<b>0x01</b>	Selezione <b>ACQUA</b>	1 €	Ciano (█ █)
<b>0x02</b>	Selezione <b>SNACK</b>	2 €	Magenta (█ █)
<b>0x03</b>	Selezione <b>CAFFÈ</b>	1 €	Giallo (█ █)
<b>0x04</b>	Selezione <b>THE</b>	2 €	Verde (█)

Byte	Azione	Prezzo	Colore LED
0x09	<b>ANNULLA / RESTO</b>	-	Viola (Reset)
0x0A (10)	<b>CONFERMA ACQUISTO</b>	-	Avvia Erogazione
0x0B (11)	<b>RIFORNIMENTO</b>	-	Reset Scorte Max

 **Nota:** A partire dalla v8.4, l'erogazione richiede **SEMPRE conferma esplicita** (comando 10). Non c'è più erogazione automatica dopo inserimento credito.

---

## Screenshot e Demo

 **Galleria Completa:** [Visualizza tutte le foto e schermate su Google Photos](#)

La galleria include:

-  **App Android:** Screenshot interfaccia completa, selezione prodotti, connessione BLE
  -  **Display LCD:** Messaggi sistema, stati FSM, feedback utente
  -  **Hardware Setup:** Breadboard, cablaggio sensori, componenti montati
  -  **Demo Live:** Funzionalità in azione (erogazione, sensori, LED RGB)
- 

## Guida all'Uso

- Avvio:** Accendi la Nucleo. Il display mostra "ECO MODE BLE OK". Il LED è Verde.
  - Connessione:** Apri l'App Android e premi "**CONNETTI DISPOSITIVO**".
  - Selezione:** Dall'App, tocca un prodotto (es. "SNACK").
    - La Nucleo cambia LED (Magenta) e mostra "Ins.Mon x SNACK".
    - Il prezzo target viene impostato a 2€.
  - Inserimento:** Copri il sensore LDR per simulare l'inserimento di monete.
    - L'App aggiorna il credito in tempo reale.
  - Erogazione:** Raggiunti i 2€, il servo si attiva, il buzzer suona e il credito viene scalato.
  - Ripensamento:** Se hai inserito monete ma vuoi annullare, premi "**ANNULLA / RESTO**" sull'App o il tasto Blu sulla scheda.
- 

## Requisiti & Installazione

 Firmware STM32

- Versione Corrente:** v8.14 (LCD Refill Feedback + Sonar Stable)
- IDE:** Keil Studio Cloud / Mbed Studio / Mbed CLI
- Librerie:** [mbed-os](#) (v6+), [TextLCD](#), [X\\_NUCLEO\\_IDB05A1](#)
- File:** [firmware/main.cpp](#) ← **Usa questa versione!**
- Guida Completa:** Vedi [firmware/README.md](#)

 **Novità v8.14 (Versione Finale):**

- **LCD Feedback Rifornimento:** Mostra "RIFORNIMENTO..." → "RIFORNIMENTO OK!" + "Scorte: 5/5/5"
- **Sonar Stabile:** Timing ottimale (echoDuration=0 fix), funzionamento verificato
- **LDR Spike Detection:** Baseline adattivo EMA con soglie relative (+20%/+5%)
- **BLE Feedback LCD:** Notifiche connessione/disconnessione su display
- **Auto-Refund:** Resto immediato alla disconnessione BLE (3s vs 30s)
- **LED RGB Configurabile:** Supporto common cathode/anode
- **Log Compatto:** Monitor seriale ottimizzato (output pulito)

## Android App

- **IDE:** Android Studio Koala (o superiore)
- **Min SDK:** 24 (Android 7.0). Target SDK: 34 (Android 14)
- **Permessi:** Bluetooth Scan, Connect e Location (gestiti a runtime)
- **File:** [app/src/main/java/com/example/vendingmonitor/MainActivity.kt](#)

## Documentazione Aggiuntiva

- **Bug Fixes & Changelog:** [BUGFIXES.md](#)
- **Wiring Hardware:** [WIRING.md](#)
- **Android App Guide:** [ANDROID\\_APP.md](#)
- **Screenshot & Demo:** [Google Photos Gallery](#)

## Struttura Repository

```
VendingMonitor/
├── app/                                # App Android (Kotlin + Jetpack Compose)
│   └── src/main/java/com/example/vendingmonitor/
│       └── MainActivity.kt      # Activity principale con gestione BLE
├── firmware/                            # Firmware STM32 (C++ Mbed OS)
│   ├── main.cpp                         # ← v8.14
│   └── README.md                        # Guida compilazione firmware
├── BUGFIXES.md                          # Documentazione bug fix v7.2
├── WIRING.md                            # Schema elettrico e cablaggio
└── README.md                            # Questo file
```

## Licenza

Progetto Open Source sviluppato a scopo didattico per dimostrare l'integrazione Full-Stack IoT.

*Developed with by Marco Santoro*