

Università degli Studi di Salerno  
DIEM, a.a. 2021-22  
Corso: SISTEMI EMBEDDED  
Professore: Vincenzo Carletti



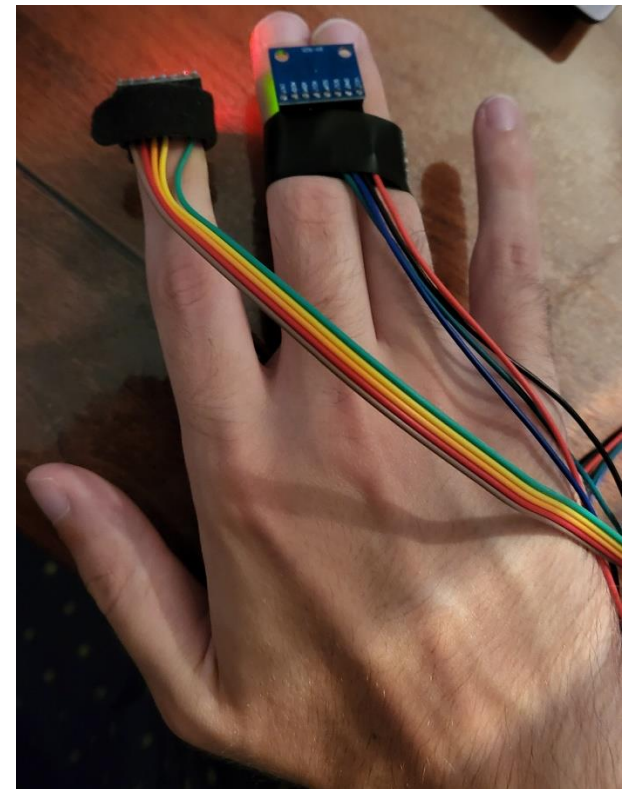
# AIR-MOUSE

BY GROUP 6

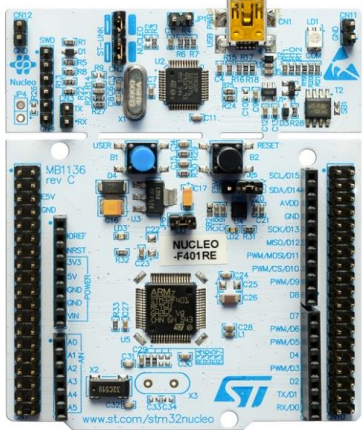
# INTRODUZIONE

***Air-Mouse è un mouse indossabile intelligente, capace di inviare input HID ad un computer attraverso una tecnologia motion sensing***

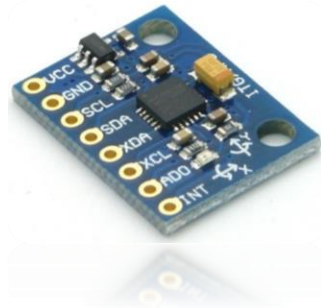
- ✓ Riconoscimento di **pattern aerei** (gesture) per riprodurre click, doppio click, zoom-in, zoom-out, ecc.
- ✓ **Calibrazione** smart e customizzabile
- ✓ Conservazione in un **log** di tutti gli eventi
- ✓ Adozione di un **protocollo di comunicazione** appositamente progettato per l'interfacciamento con il PC
- ✓ Utilizzo di diversi **task** per compiere molteplici attività



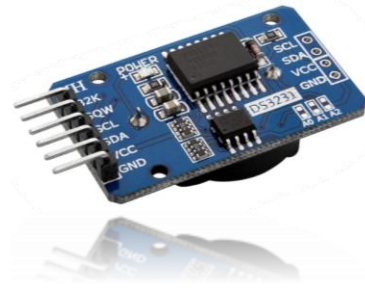
# COMPONENTI PRINCIPALI



Scheda  
**STM32F401RE**



Giroscopio-  
Accelerometro  
**GY-521 MPU6050**  
(X2)



Real Time Clock  
**DS3231**



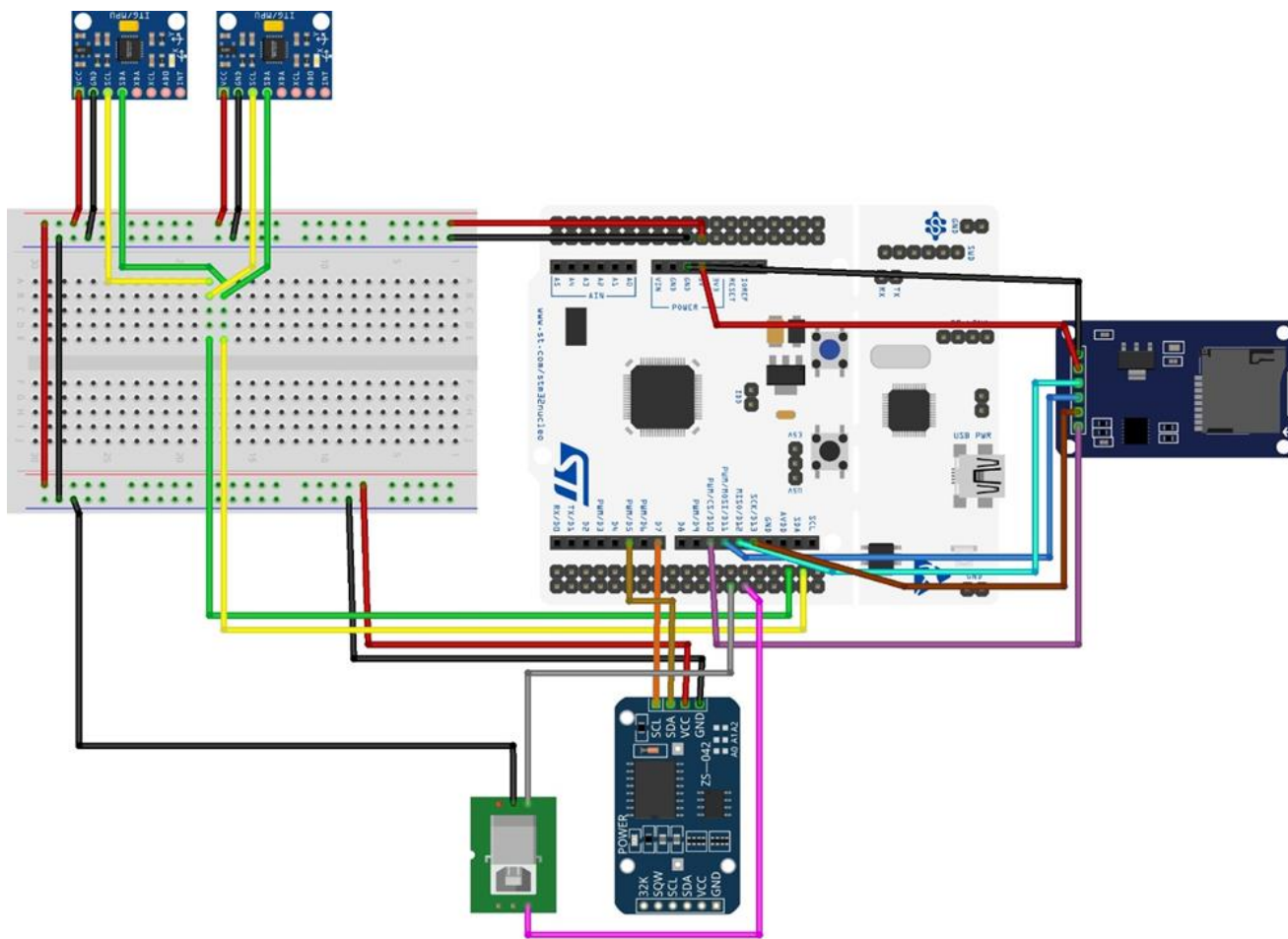
Micro SD  
Reader

# SCELTE PROGETTUALI CHIAVE

1. Calibrazione **flessibile** e intuitiva attraverso apposito **monitor seriale** e connessione **USB** con protocollo **UART**
2. Realizzazione di una **configurazione di default** direttamente utilizzabile dall'utente
3. Adozione di **librerie standard di STM per l'HID**
4. Entrambi i giroscopi-accelerometri vengono adoperati per rilevare le **gesture**, ma uno solo di essi comanda lo **spostamento** del cursore sullo schermo
5. La **velocità** e la **posizione** del cursore sono regolate dall'**inclinazione della mano**
6. Adozione di un **sistema di posizionamento relativo**
7. Utilizzo di un **SO multi-threaded** per garantire elevata **responsività** e contemporanea gestione di più attività
8. Scrittura di tutti gli eventi in un **unico file di log**

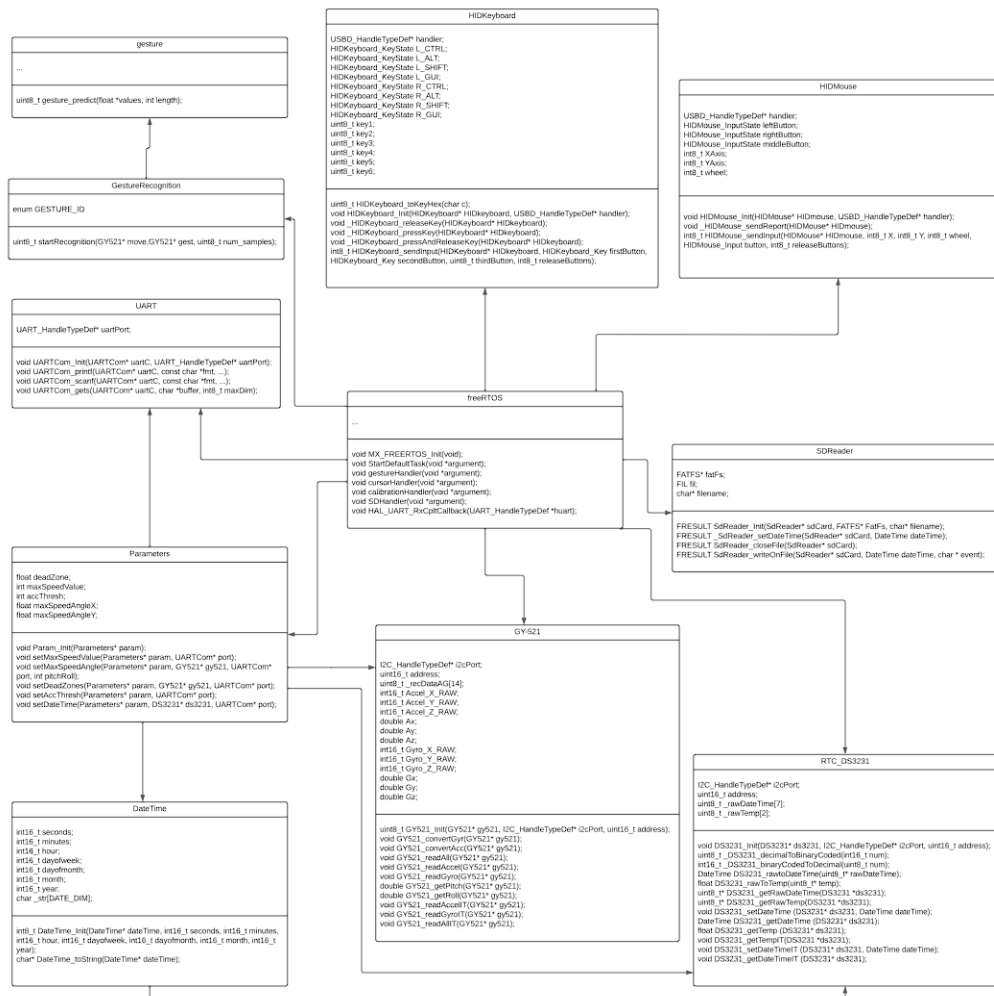


# ARCHITETTURA HARDWARE



- I due **GY-521** sono connessi allo stesso bus **I2C1** tramite la breadboard
- Il **DS3231** è connesso al bus **I2C3** direttamente sulla STM32
- Il **lettore SD** è connesso direttamente ai pin **MOSI** e **MISO** della scheda per il bus **SPI**

# ARCHITETTURA SOFTWARE



- ✓ **Il codice** caricato sulla scheda è organizzato in maniera **modulare**
- ✓ A ciascun dispositivo è associata una **libreria** per la gestione delle sue **funzioni elementari** e la sua **comunicazione** con la scheda

# PRINCIPALI LIBRERIE

## **Libreria RTC**

- *gestisce le operazioni di lettura/scrittura del Real Time Clock DS3231*
- *presenta funzioni apposite sia per polling sia per interrupt*
- *converte dati grezzi*

## **Libreria Parametri**

- *fornisce supporto alla fase di calibrazione*
- *presenta una struttura dati in cui viene memorizzata una serie di parametri utili*

## **Libreria SDCard**

- *contiene le funzioni atte alla scrittura dei log sulla microSD*

## **Libreria HID**

- *permette alla scheda STM di comunicare con il PC cui è connessa così da essere riconosciuta come dispositivo HID*
- *completa e adatta ad essere riutilizzata*

## **Libreria Giroscopio-Accelerometro**

- *gestisce la configurazione e l'uso di GY-521*
- *converte dati grezzi*
- *presenta funzioni apposite sia per polling sia per interrupt*
- *consente di settare opportuni parametri*

## **Modulo FATFS**

- *libreria leggera per MC che implementa il supporto al filesystem FAT/exFAT*

# GESTIONE DEI THREADS

## **Thread di acquisizione del movimento**

- priorità Real Time
- legge costantemente dal GY-521 dedicato al movimento e modifica la posizione del cursore

## **Thread di calibrazione**

- priorità Normale
- comunica con il monitor seriale
- invia e riceve testo per i comandi

## **Thread di rilevazione delle gesture**

- priorità Real Time
- rileva le gesture attraverso letture continue dei due GY-521

## **Thread di log**

- priorità Normale
- scrive sul file di log quando viene rilevata una gesture

Inoltre:

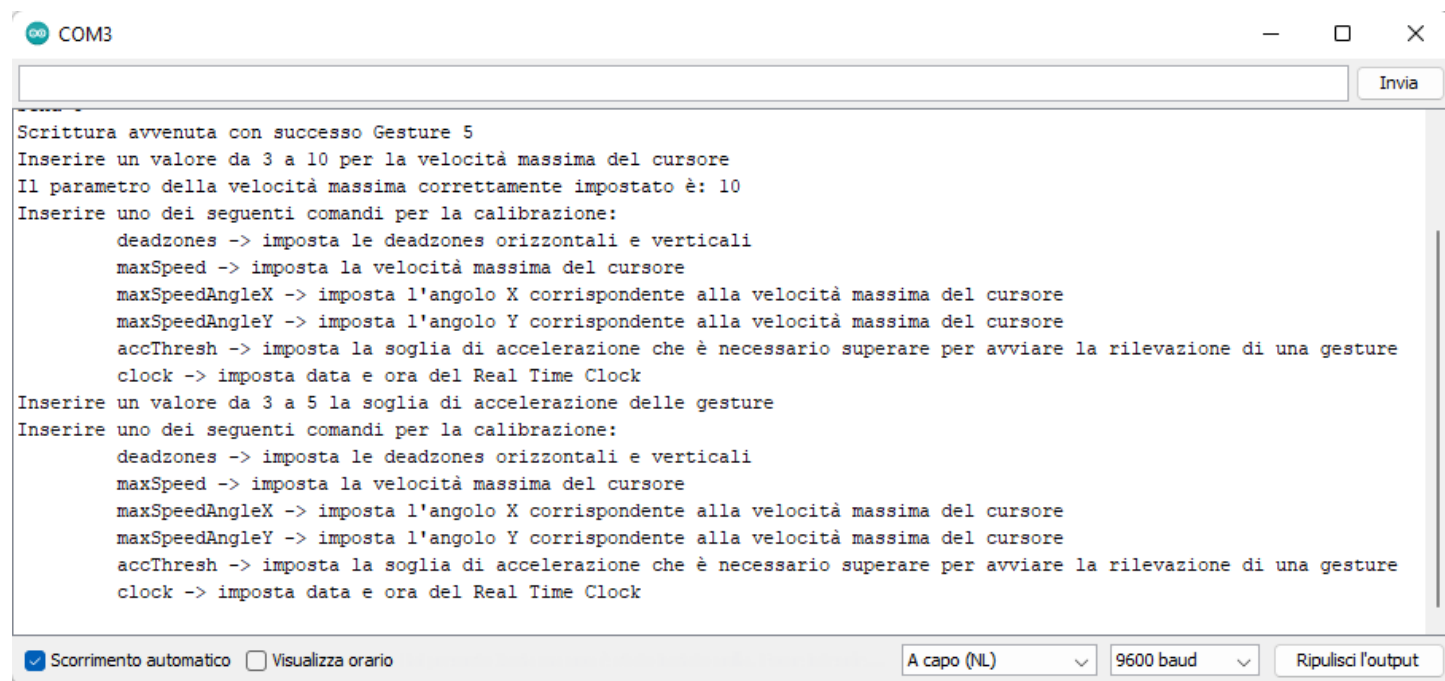
1. È presente il **default thread di RTOS** che viene eseguito all'avvio e quando la CPU è in idle
2. La sincronizzazione tra thread è gestita mediante 4 **mutex**
3. Viene adoperata una **coda** per regolare la comunicazione tra il thread delle gesture e quello dei log



# PROCEDURA DI CALIBRAZIONE

La calibrazione viene effettuata attraverso un **monitor seriale** che riceve e manda **dati testuali** tramite la connessione UART della scheda, su USB. La procedura è intuitiva, coinvolgente e suddivisibile in fasi

L'utente può **visualizzare messaggi** provenienti dalla scheda e inviare a sua volta delle stringhe per selezionare dei **comandi**:



```
COM3
Invia

Scrittura avvenuta con successo Gesture 5
Inserire un valore da 3 a 10 per la velocità massima del cursore
Il parametro della velocità massima correttamente impostato è: 10
Inserire uno dei seguenti comandi per la calibrazione:
    deadzones -> imposta le deadzones orizzontali e verticali
    maxSpeed -> imposta la velocità massima del cursore
    maxSpeedAngleX -> imposta l'angolo X corrispondente alla velocità massima del cursore
    maxSpeedAngleY -> imposta l'angolo Y corrispondente alla velocità massima del cursore
    accThresh -> imposta la soglia di accelerazione che è necessario superare per avviare la rilevazione di una gesture
    clock -> imposta data e ora del Real Time Clock
Inserire un valore da 3 a 5 la soglia di accelerazione delle gesture
Inserire uno dei seguenti comandi per la calibrazione:
    deadzones -> imposta le deadzones orizzontali e verticali
    maxSpeed -> imposta la velocità massima del cursore
    maxSpeedAngleX -> imposta l'angolo X corrispondente alla velocità massima del cursore
    maxSpeedAngleY -> imposta l'angolo Y corrispondente alla velocità massima del cursore
    accThresh -> imposta la soglia di accelerazione che è necessario superare per avviare la rilevazione di una gesture
    clock -> imposta data e ora del Real Time Clock

☒ Scorrimento automatico ☐ Visualizza orario
A capo (NL) 9600 baud Ripulisci l'output
```

1. **maxSpeed**: max velocità cursore
2. **deadzones**: angoli entro i quali la mano si considera 'ferma'
3. **accThresh**: soglia di accelerazione per pattern recognition
4. **maxSpeedAngleX, maxSpeedAngleY**: angoli a cui corrisponde la max velocità del cursore
5. **Clock**: data e ora

# RETE NEURALE



- ✓ Per il **riconoscimento delle gesture** è stato utilizzato il classificatore **ExtraTreesClassifier**, messo a disposizione dal framework **EMLearn**
- ✓ La scelta fatta è il **compromesso migliore** tra performance, velocità e memoria occupata
- ✓ Il **training set** è stato costruito manualmente registrando 100 letture da entrambi i GY-52 I: ciascuna lettura è a sua volta costituita da 50 samples



GRAZIE PER  
L'ATTENZIONE