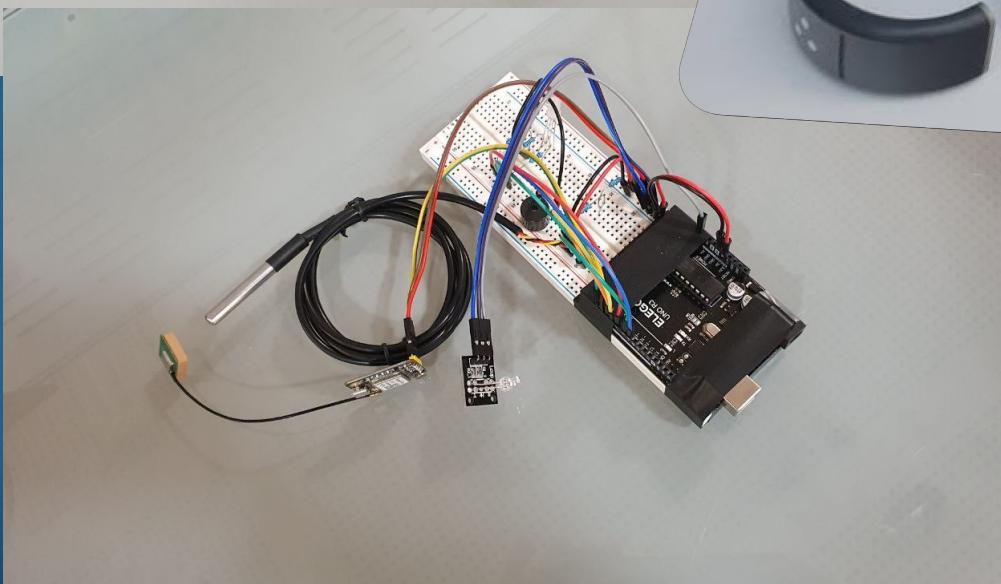
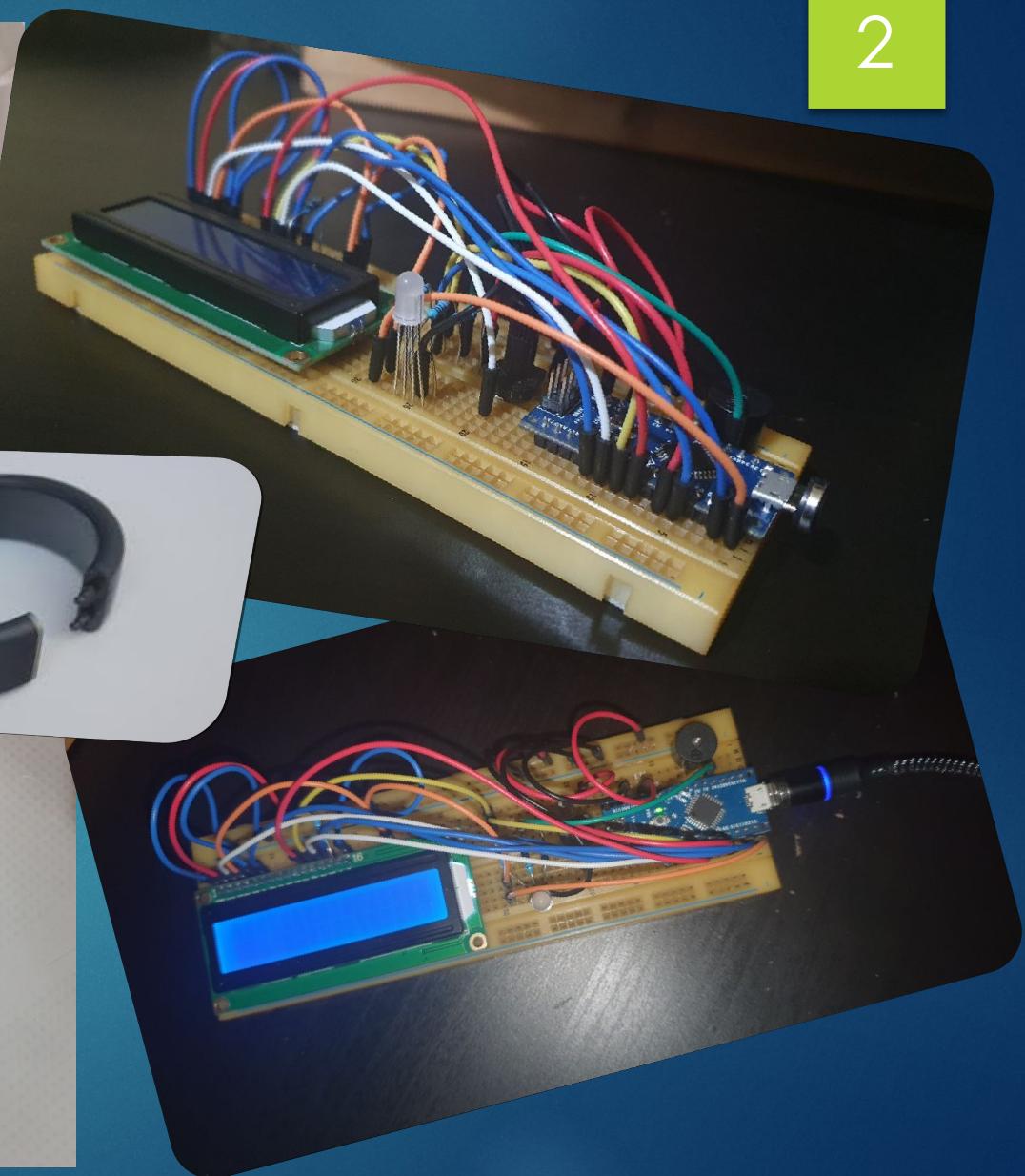
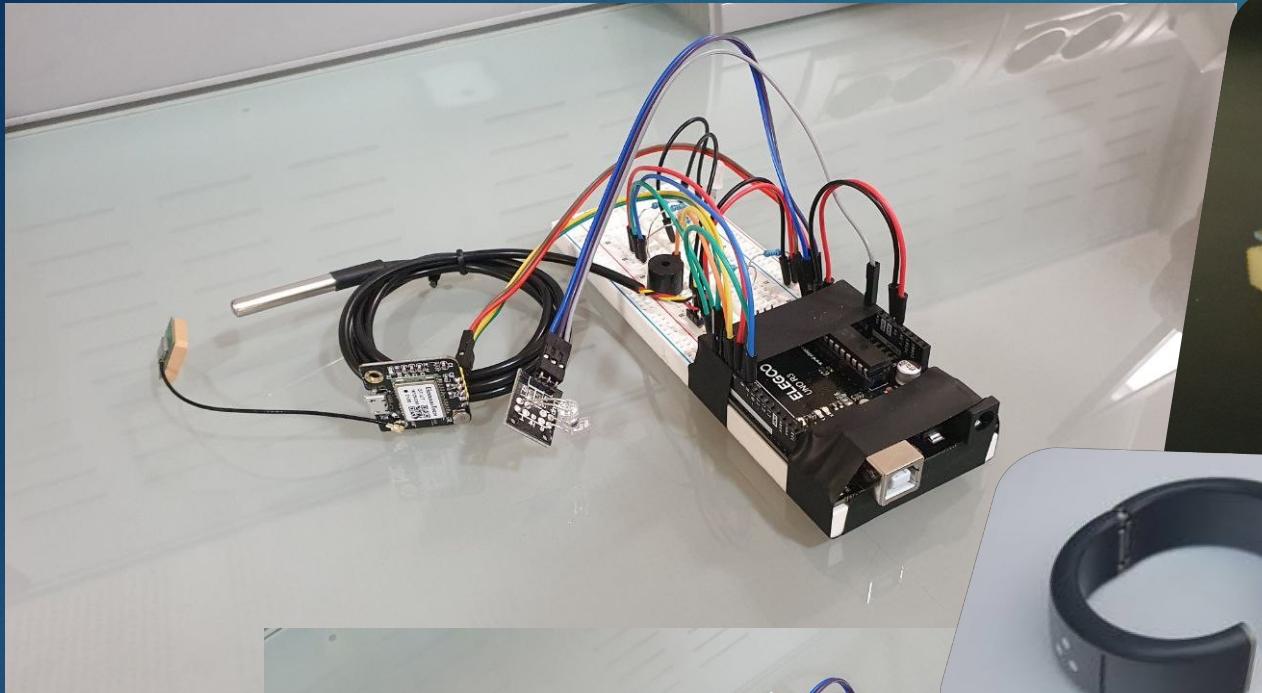


SBAM

SMART
BRACELET FOR
ANIMALS
MANAGEMENT

Progetto realizzato da:
Caputo Danilo
Catalano Luca
Corrado Luca Eugenio

2



Presentazione del prototipo

- ▶ Cos'è SBAM
- ▶ Componenti del SBAM
- ▶ Come lavora SBAM
- ▶ Perché utilizzare SBAM

Cos'è SBAM

- ▶ SBAM è un bracciale intelligente per il controllo e la gestione degli animali di uno zoo
- ▶ Garantisce localizzazione GPS, controllo della temperatura e del battito cardiaco dell'animale, inviando opportune notifiche al guardiano in presenza di un problema

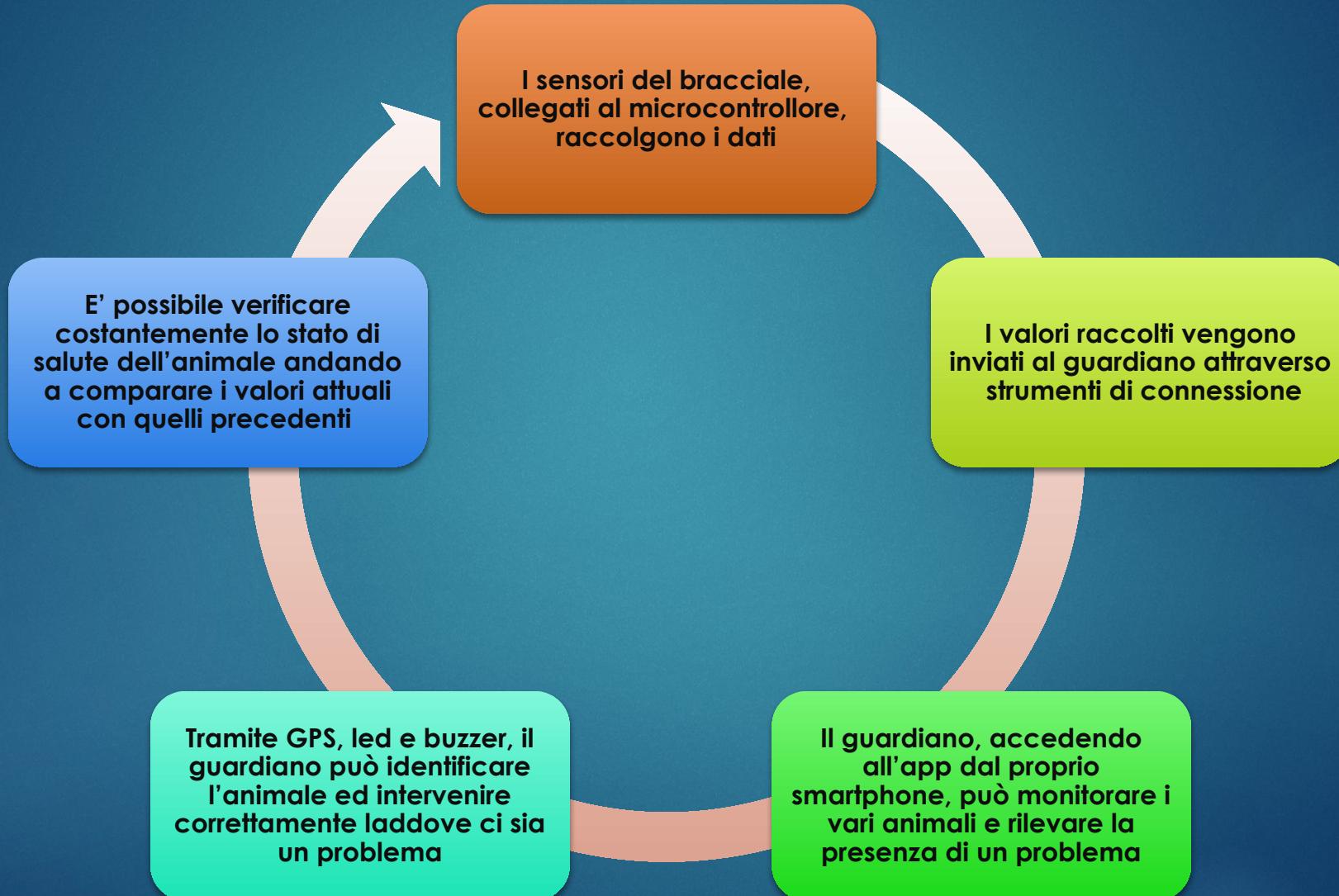
Componenti del **SBAM**

- ▶ **Sensori e attuatori**: Sensore GPS, di temperatura e di battito cardiaco insieme con led, buzzer e schermo LCD collegati al microcontrollore.
- ▶ **Database** : per lo storage dei valori.
- ▶ **Server** : per smistamento ed elaborazione dati.
- ▶ **Mobile app** : utilizzata dal guardiano per il controllo dei valori riportati dai sensori in tempo reale e per un confronto con quelli precedenti memorizzati nel database.

Come lavora

SBAM

6



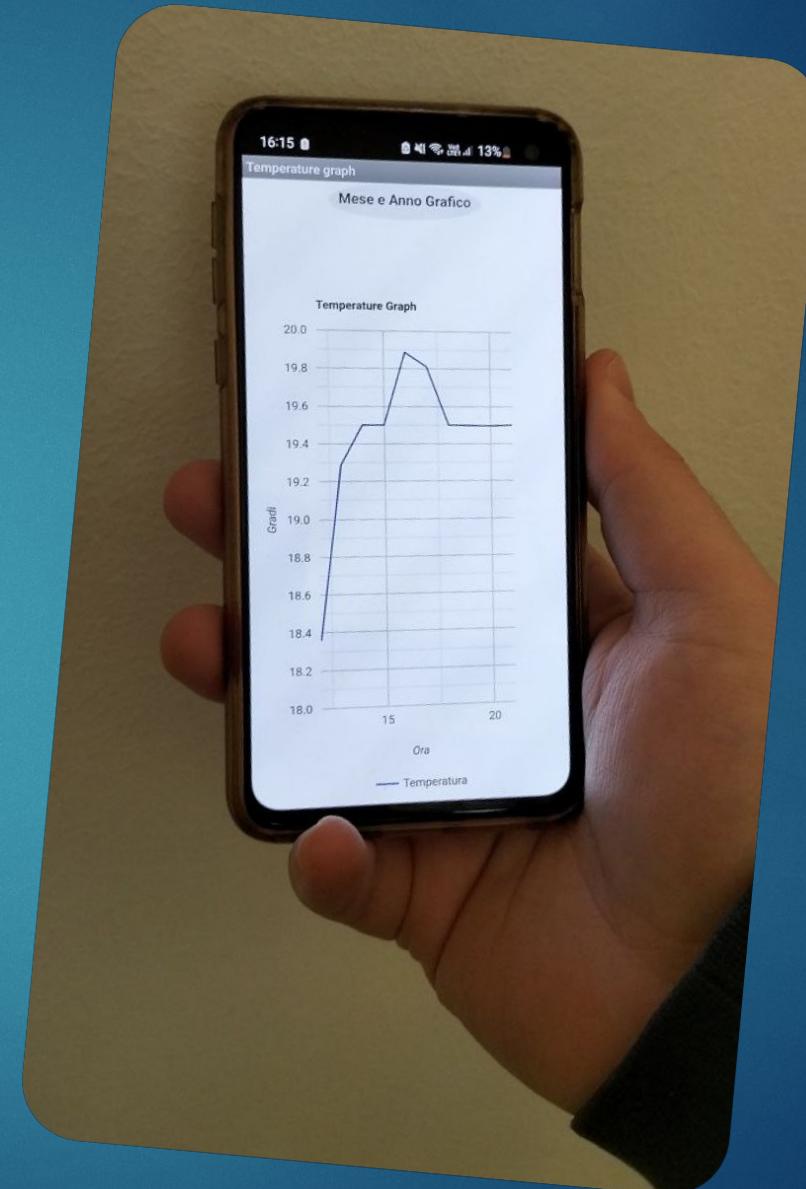
Perché utilizzare **SBAM**

- ▶ Facilità d'utilizzo per obiettivi precisi
- ▶ Mobile
- ▶ Risposta istantanea
- ▶ Storage dei valori ricevuti

Facilità d'utilizzo...

Pochi e semplici passi

- ▶ Sullo smartphone si ricevono i dati raccolti dai sensori inseriti sul bracciale.
- ▶ Selezionando l'animale a cui si è interessati si accede all'area riservata a quello specifico animale.
- ▶ A questo punto è possibile visualizzare i dati ricevuti in tempo reale così come quelli precedentemente memorizzati attraverso dei grafici.



...per obiettivi precisi

Utilità concreta del SBAM

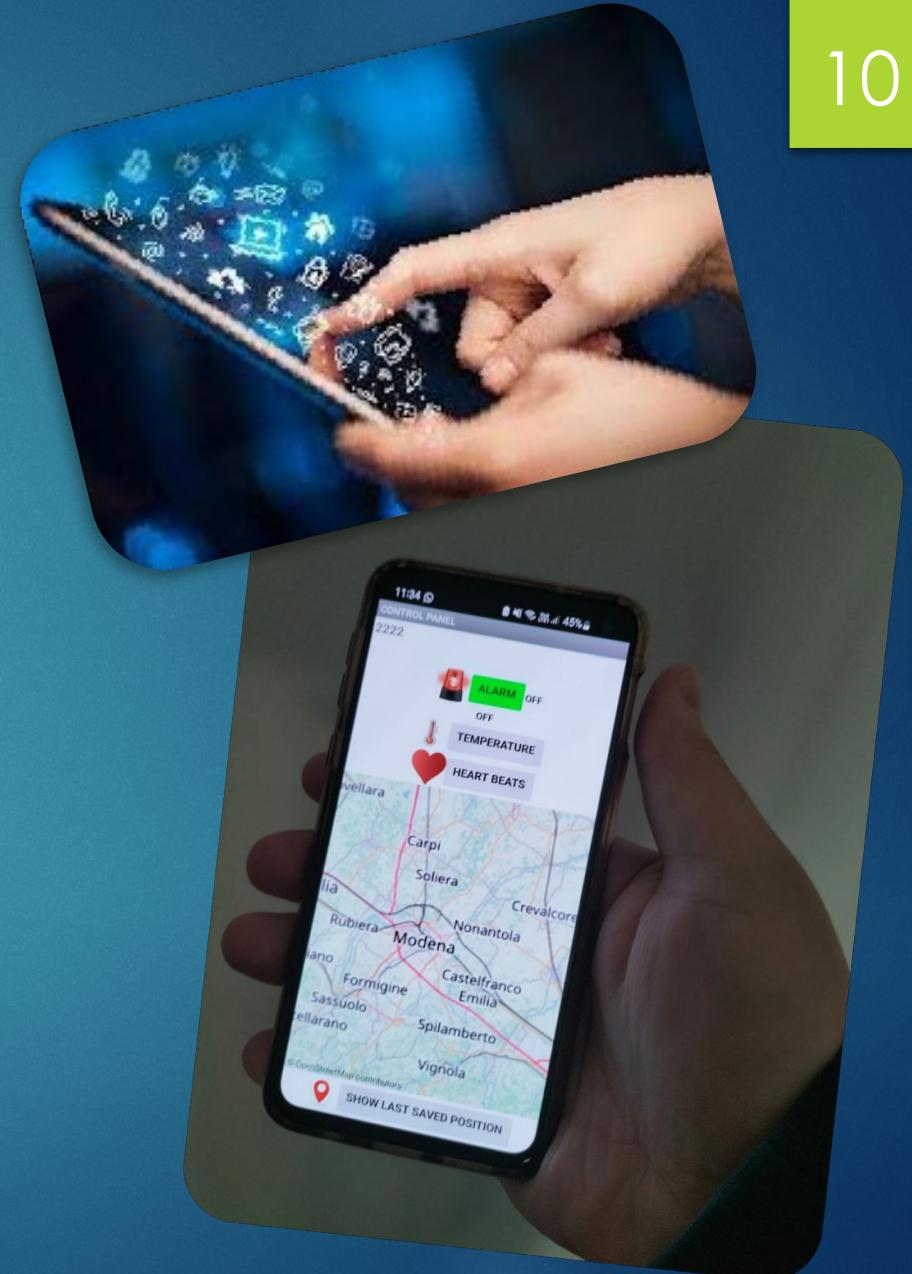
- ▶ Il bracciale intelligente permette all'utente di avere un controllo diretto e completo sull'area *in cui il SBAM è applicato*.
- ▶ Ogni animale dotato di dispositivo può essere supervisionato a distanza in modo efficiente, grazie alla visualizzazione in tempo reale dei dati raccolti dai sensori.
- ▶ La gestione efficiente permette di individuare immediatamente situazioni di pericolo per l'animale quando vengono oltrepassati valori soglia di riferimento (per temperatura o battito) o, più in generale se vengono riscontrate anomalie particolari.
- ▶ In caso di problemi, è possibile intervenire localizzando l'animale grazie al led, al rilevatore sonoro e al GPS presenti nel bracciale anche in punti ostili o nascosti.

Mobile, un grande vantaggio

In relazione agli aspetti precedentemente analizzati, cioè che rende ideale e di pratica utilità questo dispositivo è la presenza di un'app con cui visualizzare i dati provenienti dal bracciale.

Punti cardine:

- ▶ Aggiornamento della lista di animali
- ▶ Selezione dell'animale da gestire
- ▶ Visualizzazione dei valori in tempo reale
- ▶ Visualizzazione grafica dei dati precedenti



► Aggiornamento della lista di animali

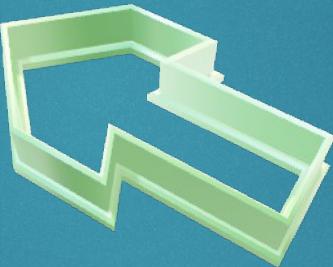
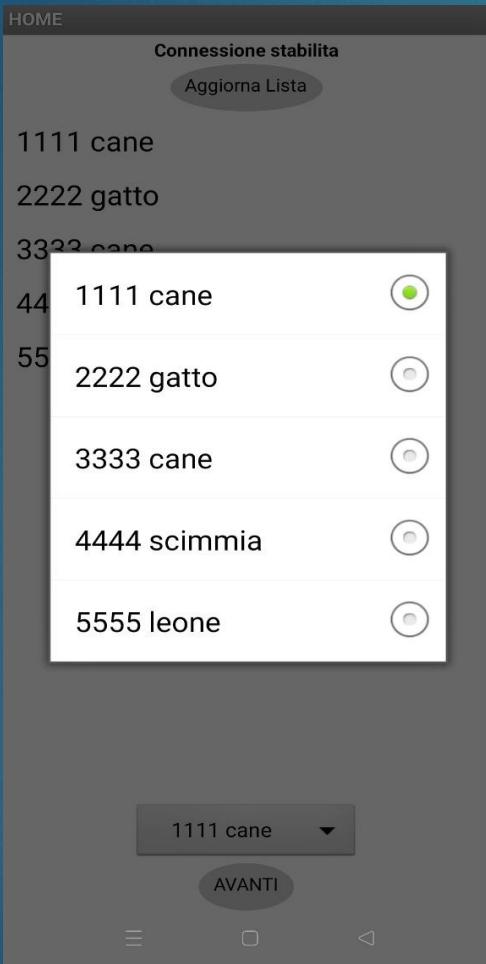


Una volta entrati nell'app, **la connessione con il bracciale dell'animale è già stabilita** e ciò risulta ben evidente dal messaggio situato in alto.

Pigliando sul pulsante "Aggiorna Lista", si ottiene **la lista completa di tutti gli animali** che, dotati del bracciale SBAM, possono essere supervisionati.



► Selezione dell'animale da gestire



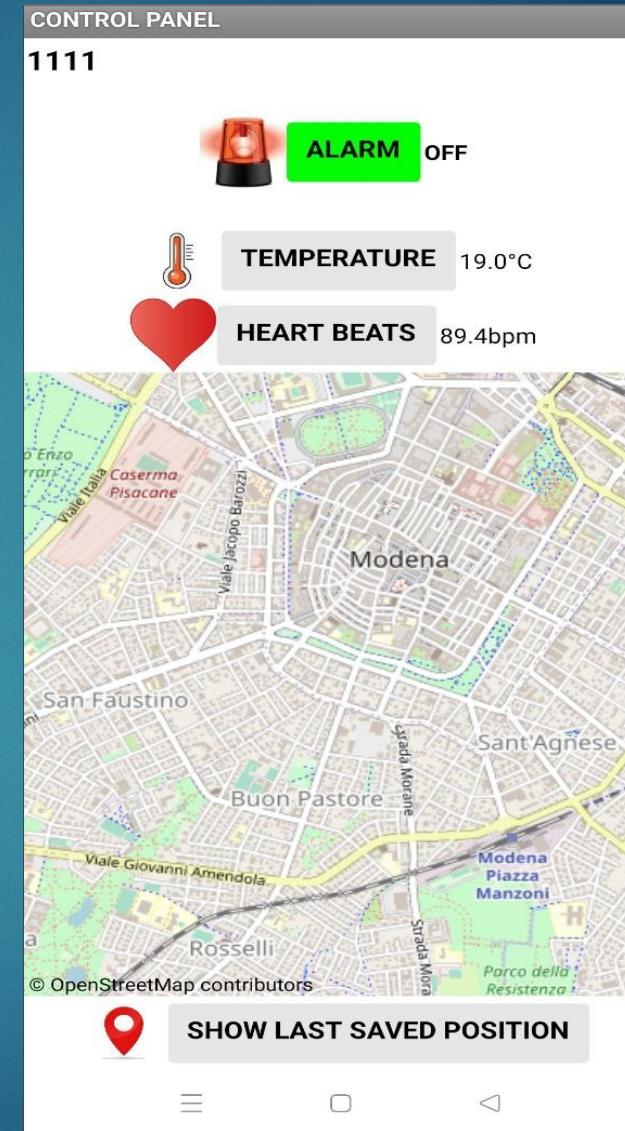
Dal menù a tendina in basso è possibile **selezionare un animale di cui si vogliono visualizzare i valori.**

Nella fase di scelta l'identificativo è dato da un codice di 4 cifre affiancato dal tipo di animale a cui fa riferimento.

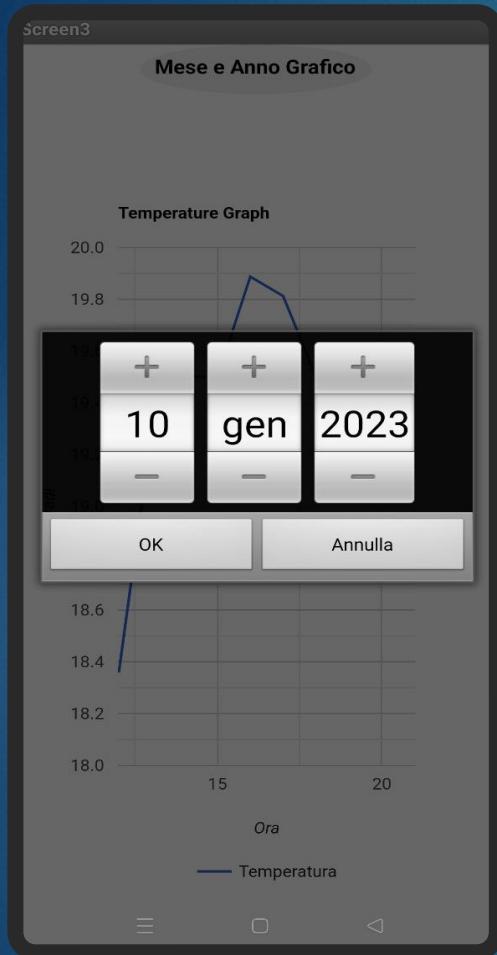
► Visualizzazione dei valori in tempo reale

- Una volta acceduti al pannello di controllo dell'animale selezionato, è possibile visualizzare i valori in tempo reale di temperatura corporea e battito cardiaco.
- Attraverso la mappa poi, è possibile vedere la posizione dell'animale, in modo da poter intervenire in caso di necessità.

Cliccando sui parametri di temperatura o battito cardiaco, si ha la visualizzazione grafica dei dati nel tempo



► Visualizzazione grafica dei dati memorizzati



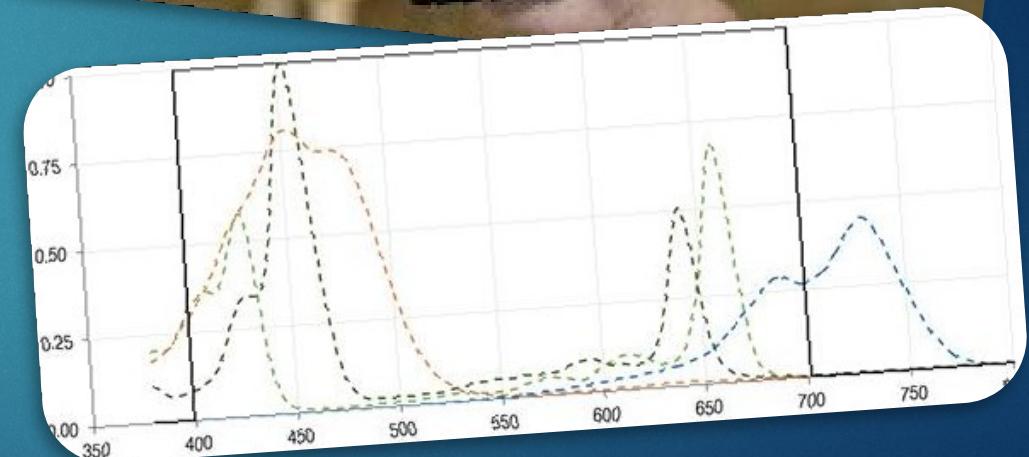
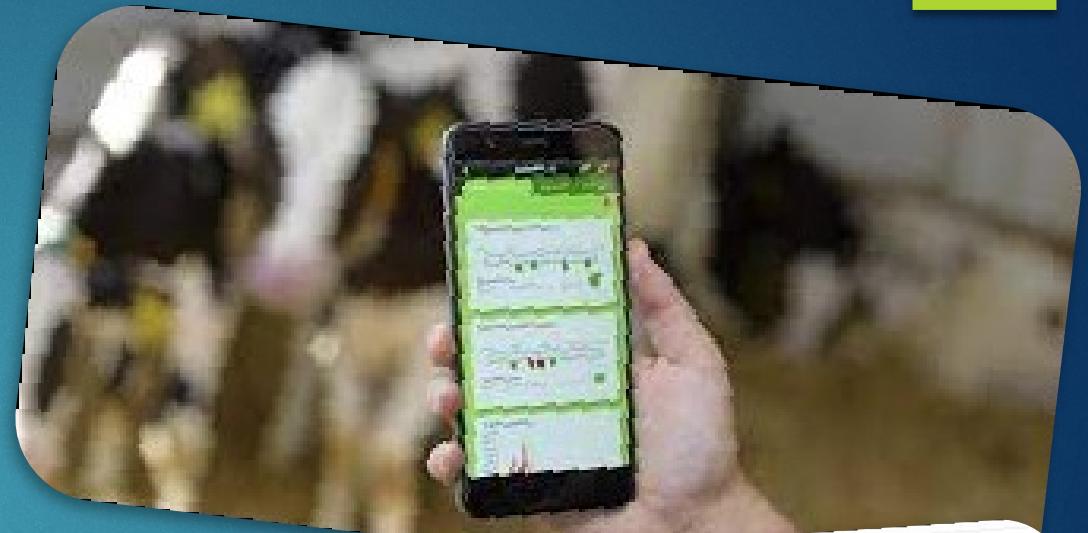
Su questa nuova schermata è possibile **selezionare la data** di cui si vuole avere una **visualizzazione grafica della media oraria dei valori.**



Risposta istantanea

L'efficienza di un sistema IoT sta anche nella capacità di inviare i dati raccolti in tempo reale al soggetto richiedente.

È possibile comparare i valori riportati in tempo reale con la media orari di quelli ricevuti in precedenza per valutare la presenza di eventuali anomalie nei valori in modo da intervenire correttamente.



Storage dei valori ricevuti

La presenza di un database nel sistema IoT è di fondamentale importanza perché permette il salvataggio dei dati ricevuti (funzione di storage).

In questo modo è possibile effettuare un monitoraggio completo sullo stato di salute dell'animale.



Installazione del sistema

Punti cardine

- ▶ Ogni animale presente nell'area di riferimento (ad esempio un'area di uno zoo) è provvisto del **bracciale SBAM** in modo da rilevare i valori di temperatura, battito cardiaco e la localizzazione GPS.
- ▶ Il sistema vede la presenza di vari **moduli Wi-Fi** sparsi nelle varie zone. Questi moduli dovranno essere di **piccole dimensioni e poco invasivi**, nel pieno rispetto dell'ambiente circostante e, nello specifico, della fauna che li circondano.
- ▶ La postazione del guardiano prevede un sistema dotato di **monitor, luce di stato e avviso sonoro** dal quale viene ricevuta la notifica in caso di malfunzionamenti o problemi. E' prevista poi **l'installazione dell'applicazione** sul proprio smartphone per visualizzare i valori dei parametri nel tempo.

Scelte di progetto

- ▶ Livello hardware
- ▶ Livello software
- ▶ Connessione
- ▶ Struttura del prototipo

Livello hardware

Bracciale lato animale

Sensori

Attuatori

Microcontrollore

Livello hardware

Postazione lato guardiano

Attuatori

Microcontrollore

Livello software

Vari strati per diverse funzionalità



**Bridge
lato
animale**



**Bridge
lato
guardiano**



Server



Database



**Mobile
app**

Connessione

Interfacciamento tra le varie componenti

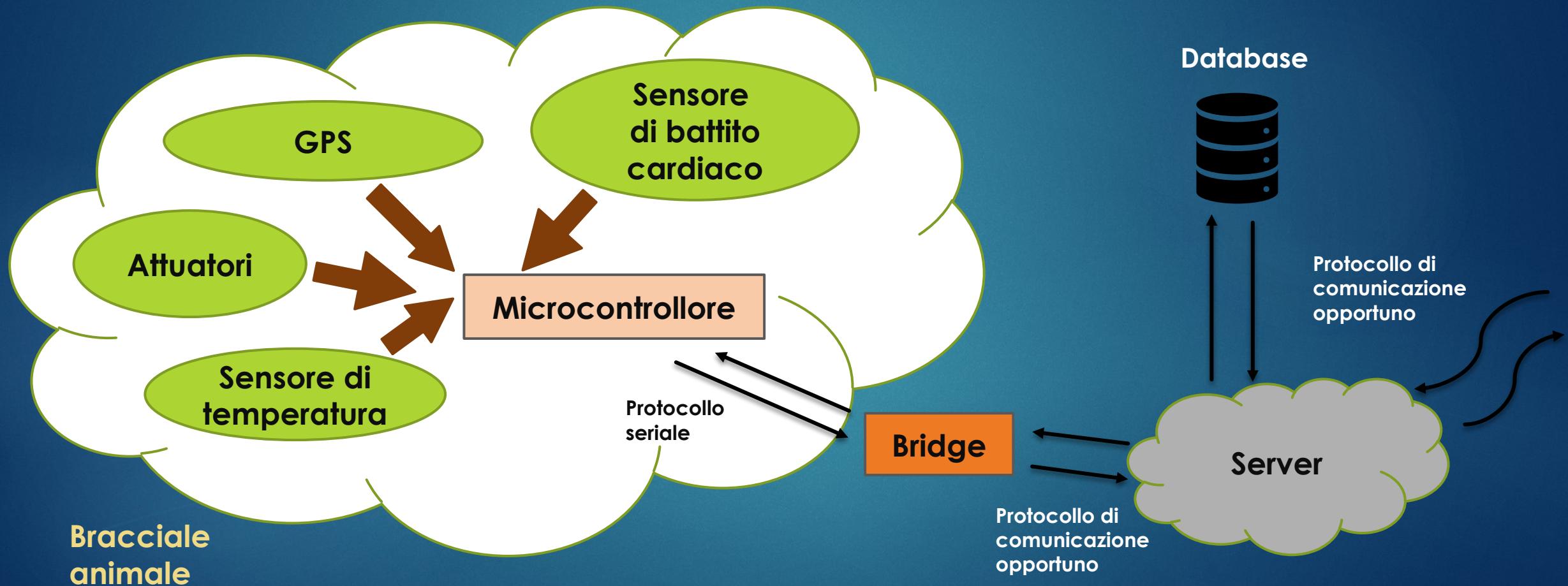
Dal
microcontrollore
al bridge

Dal bridge
Al server

Dal server
all'app o al db

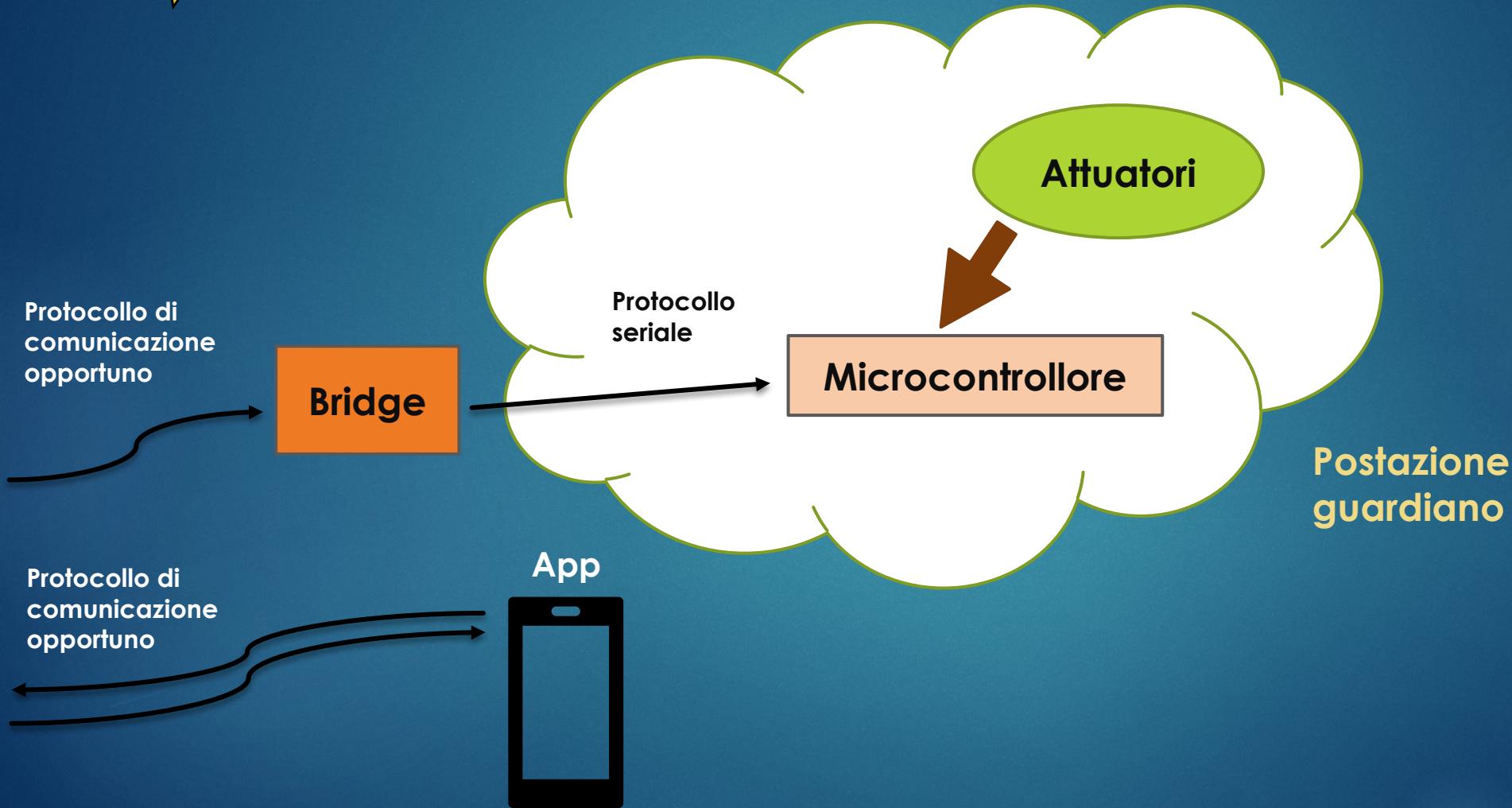
Struttura del prototipo

Dal lato animale...



Struttura del prototipo

...al lato guardiano



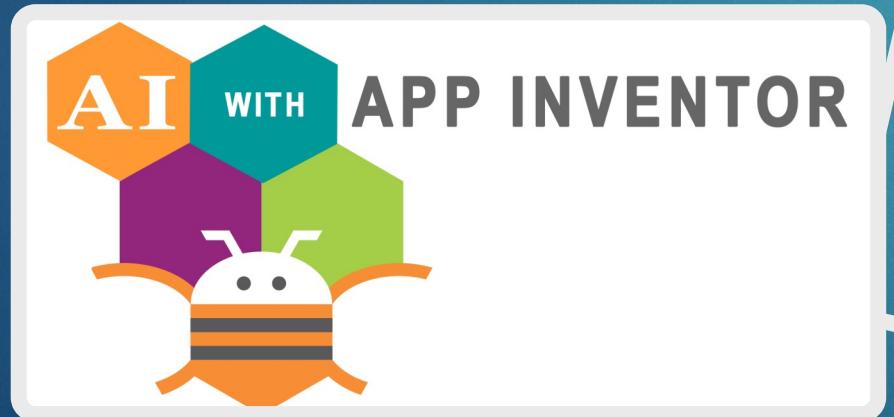
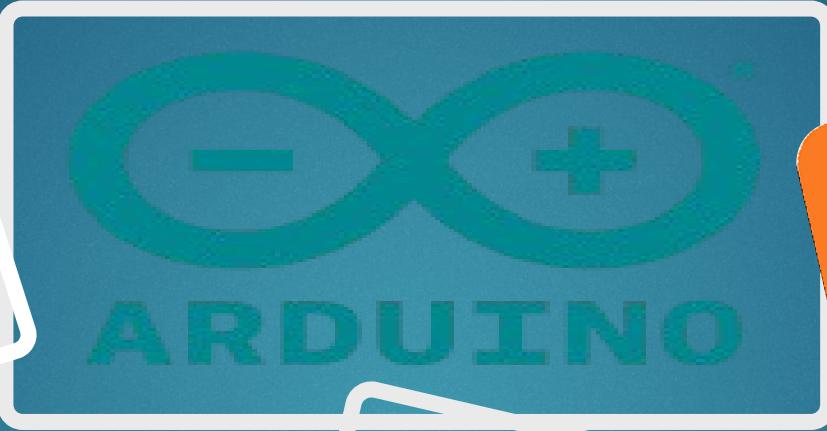
Architettura HW & SW

Caratteristiche tecniche

- ▶ 1. Microcontrollore e sensori
- ▶ 2. Bridge
- ▶ 3. Server
- ▶ 4. Storage
- ▶ 5. Client App

SW utilizzati per il prototipo

26



HW utilizzati per il prototipo

<u>Hardware utilizzato</u>	<u>Tipologia</u>
Mainboard- microcontrollore	Arduino UNO Rev.3
Sensori	GPS GT-U7, Sonda temperatura DS18B20, rilevatore battito cardiaco KY-039
Attuatori	Led, buzzer, schermo LCD
Smartphone (client)	Smartphone android (MIT App Inventor)

Microcontrollore e Sensori



Mainboard: ARDUINO UNO Rev.3



Caratteristiche:

Ottimo per prototipi
Economico
Non ingombrante
Ricco di documentazione
Bassa potenza e basso consumo
Presenza dell'Arduino IDE

Funziona all'interno del prototipo:

Acquisire i valori dai sensori
Elabora I dati
Comunica i dati al bridge
Ascolta dal bridge eventuali comandi

Mainboard: ARDUINO UNO Rev.3

30

Caratteristiche tecniche:

- Microcontrollore ATmega328
- Tensione di ingresso 7-12 V
- 14 pin I / O digitali (6 uscite PWM)
- 6 ingressi analogici
- 32kb di memoria Flash
- Velocità di 16 MHz

Detttagli tecnici:

Arduino Uno usa ATmega16U2 per gestire USB.

Ciò consente elevata velocità di trasferimento e memoria maggiore.

Non sono necessari driver per Linux o Mac (il file inf per Windows è necessario ed è incluso nell'IDE di Arduino).



Sensore di localizzazione: Modulo ricevitore GPS GT-U7

31



**Alta sensibilità e
posizionamento ad alta
precisione**



Basso consumo energetico



Miniaturizzazione



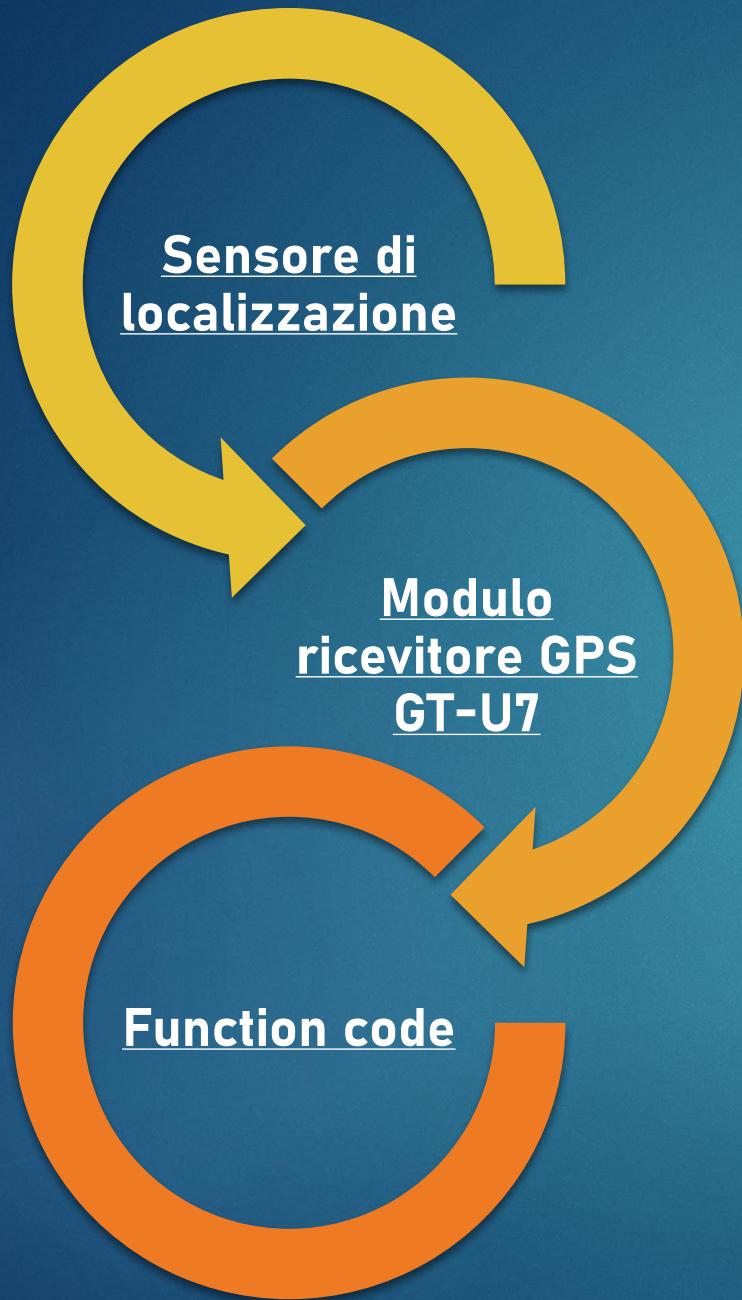
**Buona localizzazione, anche
in luoghi poco aperti**

Sensore di localizzazione: Modulo ricevitore GPS GT-U7



Specifiche:

- **Tensione di esercizio:** 3,6 V-5 V (o alimentazione USB diretta)
- **Baund rate operativo:** 9600 (modificabile)
- **Batteria a bottone ricaricabile integrata**
- **E2PROM integrato può salvare i dati dei parametri**
- **Dimensioni:** 27,6 mm * 26,6 mm

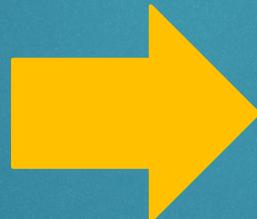


```
void gpsFunction() {  
  
    if (gps.location.isValid()) {  
        lat.floatante= gps.location.lat();  
        lon.floatante= gps.location.lng();  
    }  
    //qui si dovrebbe far mandare l'errore per  
    else {  
        lat.floatante= 44.62914473592921;  
        lon.floatante= 10.948843084508601;  
    }  
}
```

Sensore di temperatura digitale: Sonda DS18B20

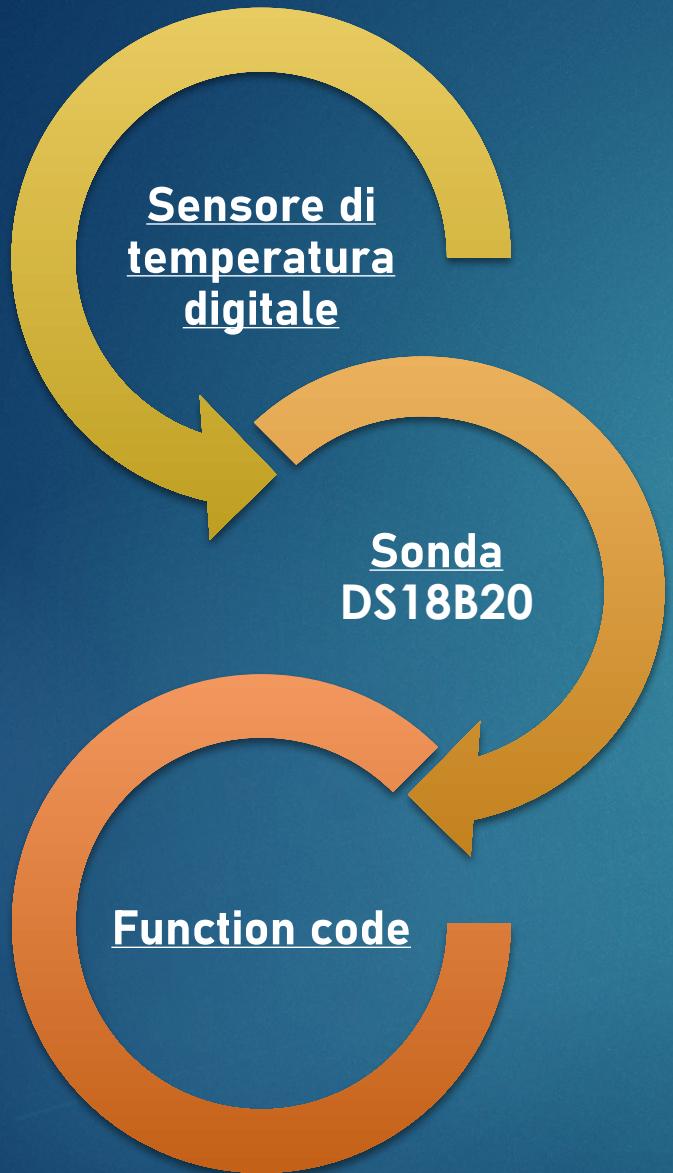


Sensore di temperatura digitale: Sonda DS18B20



Caratteristiche:

- Grazie all'interfaccia unica a 1 filo, il sensore richiede solo un pin di porta per comunicare con un microcontrollore.
- Ogni sensore ha un indirizzo seriale unico a 64 bit, quindi più sensori possono operare sullo stesso bus a un filo.
- Adatto anche per uso esterno grazie al cavo impermeabile.



```
void printTemperature() {  
  
    sensors.requestTemperatures() ;  
    float temperature= sensors.getTempCByIndex(0) ;  
    temp.floattante=temperature;  
  
    if(temp.floattante != DEVICE_DISCONNECTED_C) {  
        t = false ;  
        if(allarme == true && aUt== false)  
            allarme = false;  
        cod = 0 ;  
    }  
    else {  
        //allarme e led rosso  
        allarme = true;  
        aUt = false ;  
        if(state && !t) {  
            state = false ;  
            t = true ;  
        }  
        cod=99;  
    }  
}
```

Sensore heart-beat: Rilevatore di battito cardiaco KY-039

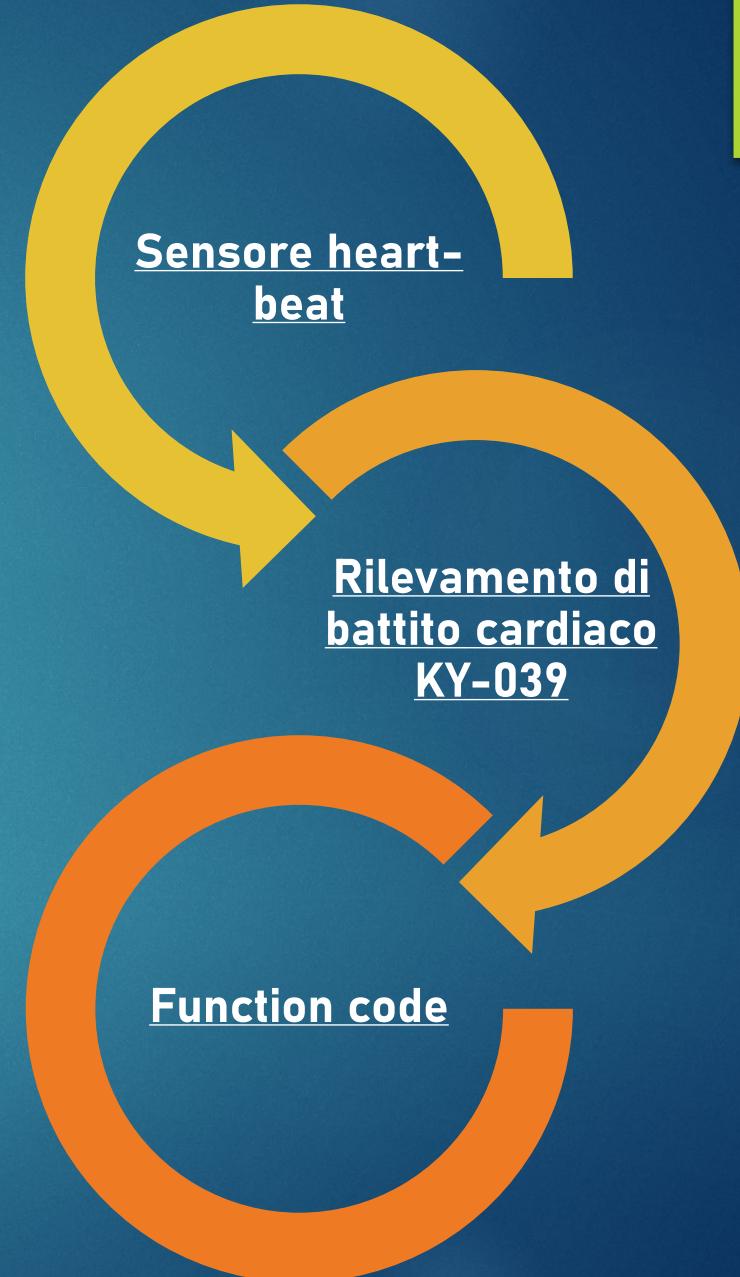


Sensore di
battito
cardiaco su
una scheda
di breakout

Tensione di
esercizio:
3.3V-5V

Sensore heart-beat: Rilevatore di battito cardiaco KY-039

```
void heartRate() {  
  
    int sum = 0;  
    for (int i = 0; i < 50; i++)  
        sum += analogRead(A0);  
    pulse.flottante = sum / 50.00 ;  
}
```



Bridge

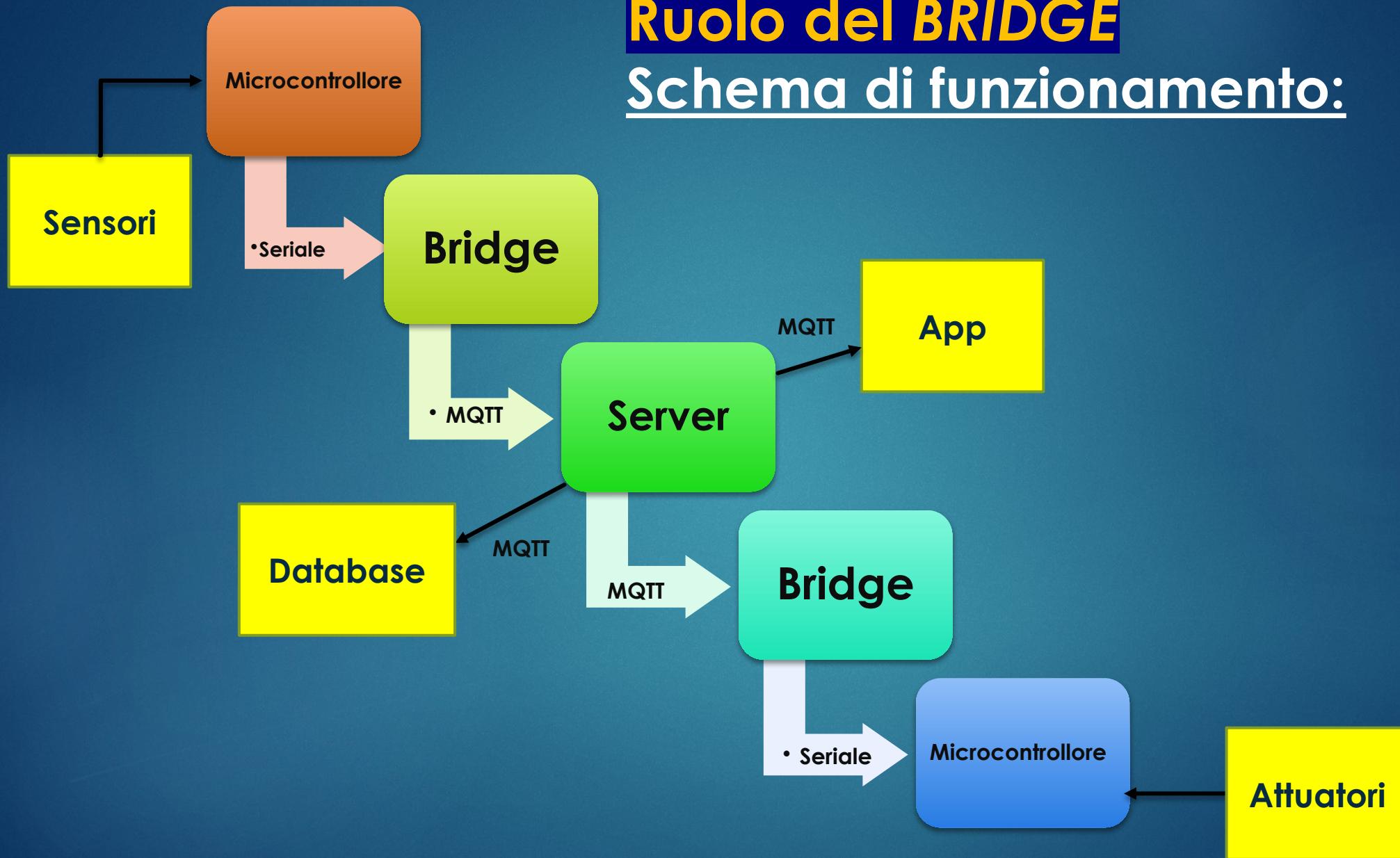


Ruolo del **BRIDGE**

- ▶ Il microcontrollore non può collegarsi direttamente alla rete internet, sarebbe irragionevole in quanto non può fare tutta la parte di trasmissione, protezione dati etc.
- ▶ E' ragionevole avere un **collegamento dati** aperto con un dispositivo detto **BRIDGE** e tale collegamento può essere anche non necessariamente standard.
- ▶ Sarà il bridge a comunicare col cloud utilizzando una serie di protocolli più standardizzati.

Ruolo del *BRIDGE*

Schema di funzionamento:



Interfacciamento microcontrollore-bridge: Protocollo seriale

42

A livello di collegamento, si distingue tra trasmissione sincrona e asincrona.

Modalità sincrona



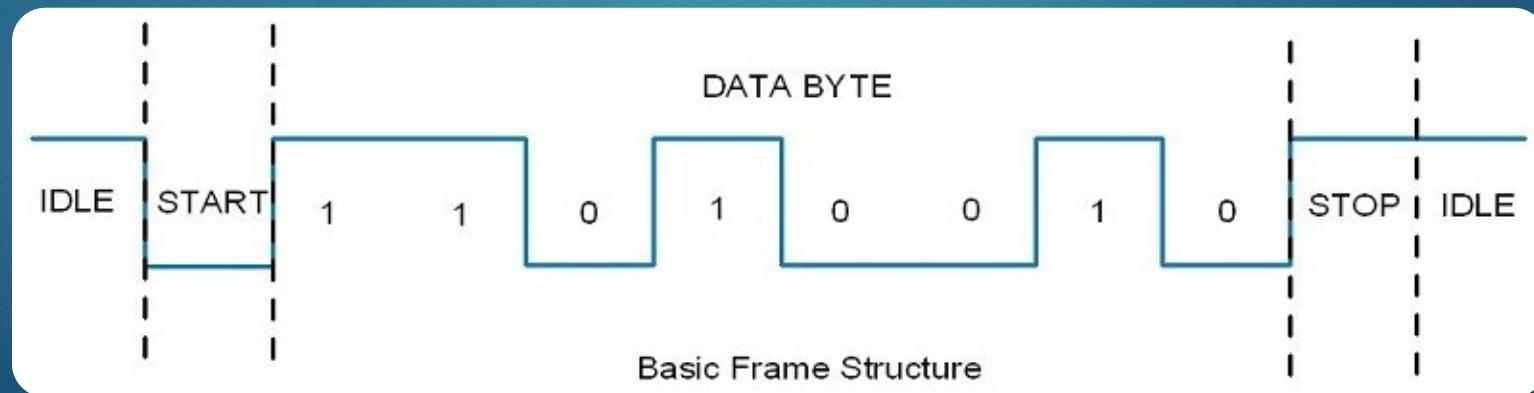
Modalità asincrona



Modalità asincrona: UART

43

- ▶ UART (''Universal Asynchronous Receiver-Transmitter'') definisce un primo layer di trasmissione e può essere implementato su un filo di trasmissione, senza specificarne il tipo.
- ▶ In UART ci sono due sottosistemi cioè ricezione e trasmissione ed entrambi hanno un clock a se, cioè sono indipendenti
- ▶ Con le comunicazioni UART va stabilito a priori il valore della frequenza, cioè quanto sarà lungo lo START iniziale
- ▶ Il modo in cui questi livelli sono mappati secondo segnali di tensione, si ha in base a diversi standard (ad esempio RS-232)



Server



Funzionamento del server

Operazioni principali nel sistema **SBAM**

- ▶ **Smistamento** → Identificazione e reindirizzamento dei dati e delle richieste
- ▶ **Temporizzazione** → Salvataggio temporizzato dei dati
- ▶ **Elaborazione** → Funzione di sviluppo dei dati raccolti

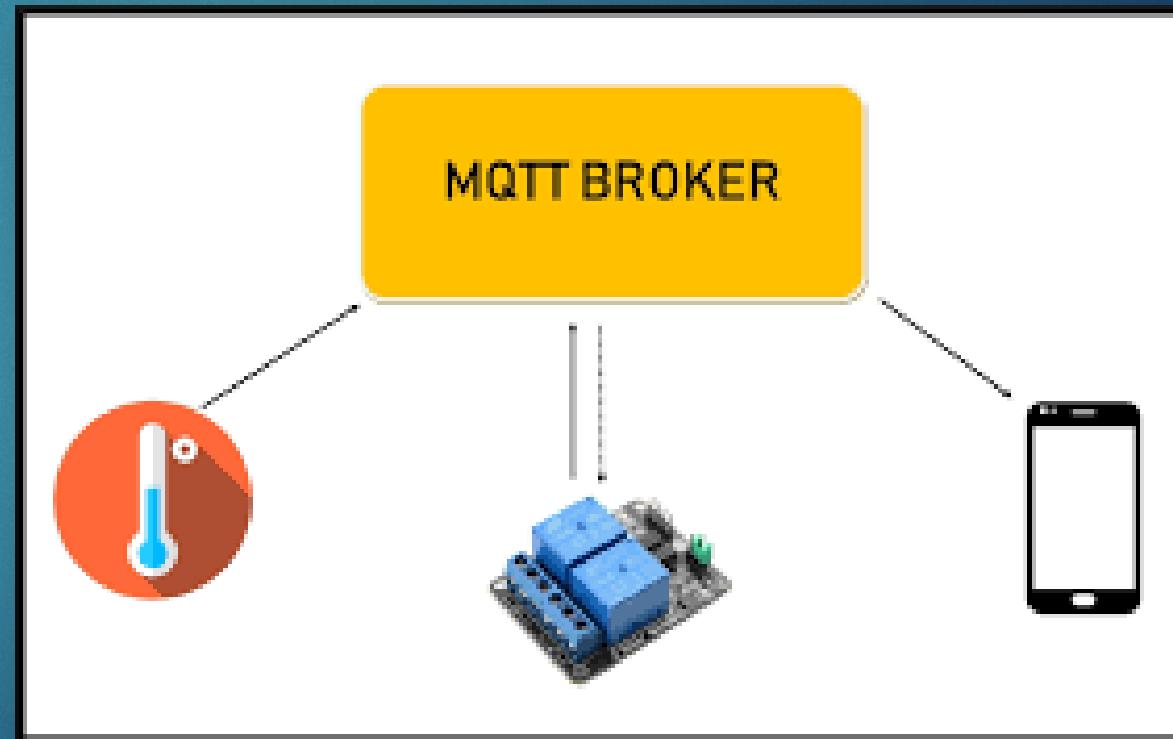
Protocollo di comunicazione **MQTT**

46

Interfacciamento bridge-server

Dal punto di vista di connessione, si basa su tre attori:

- ▶ **Broker**, lato server raggiungibile dai vari client
- ▶ **Device side**, parte bridge (con sensori e attuatori collegati)
- ▶ **Application Side**, ad esempio un cellulare

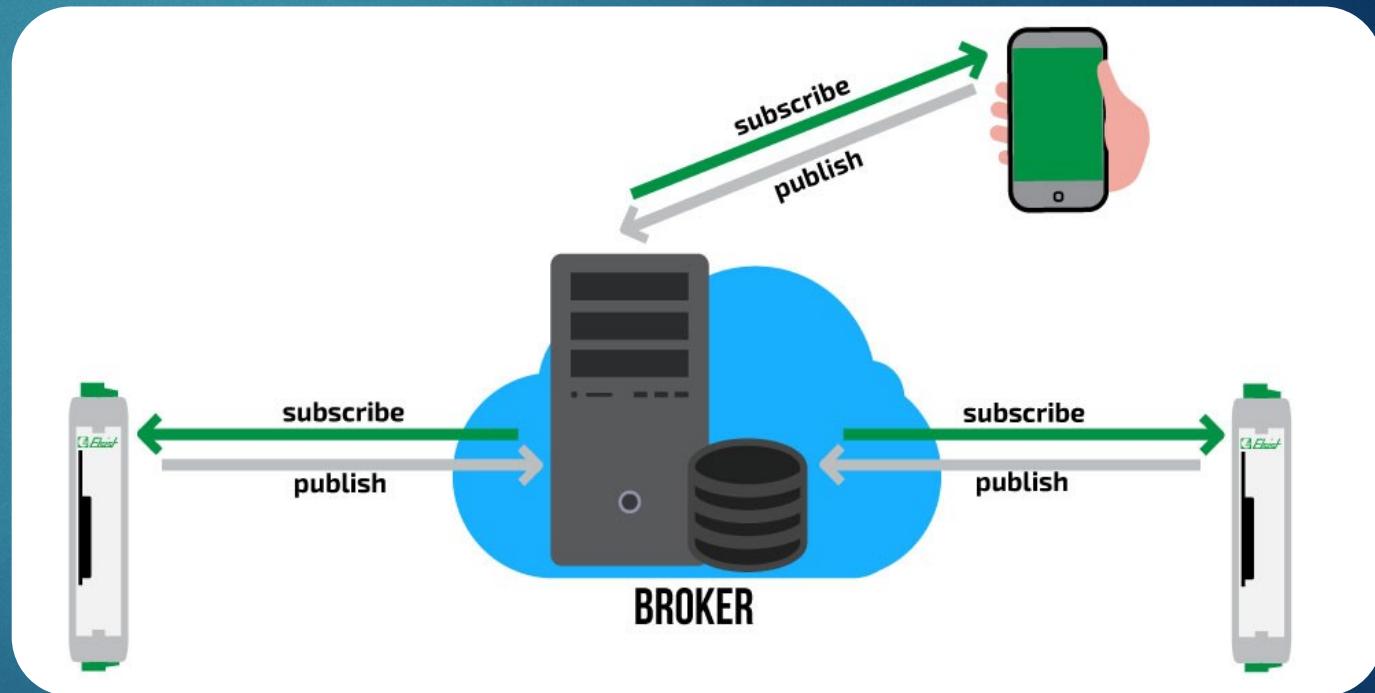


Protocollo di comunicazione **MQTT**

Interfacciamento bridge-server

Punti fondamentali:

- ▶ **Concetto di Pub/Sub:** tutti i client possono pubblicare un messaggio o sottoscrivere un argomento.
- ▶ **Topic:** la sezione in cui verranno inviati i messaggi, è un bene stabilire una gerarchia di topic.
- ▶ **QoS (Quality of service):** tre tipologie (0,1,2). Definisce la garanzia dell'effettiva avvenuta consegna dei messaggi.



Storage



Storage dei dati: Schema E/R

49

Lo storage dei dati nel SBAM permette il salvataggio delle informazioni ricevute nel tempo, garantendo un monitoraggio continuo sullo stato di salute dell'animale

Animals(ID, type, last_latitude, last_longitude)

Parameters(REF_ID, DATE, temperature, heart-rate)

ANIMALS		PARAMETERS	
Id	Integer[4] NOT NULL PK	1	N
Type	Varchar[20] NOT NULL		
Last_longitude	Float		
Last_latitude	Float		
		Ref_id	Integer[4] NOT NULL PK FK
		Date	Datetime NOT NULL PK
		Temperature	Float NOT NULL
		Heart-rate	Float NOT NULL

Storage dei dati: Database

50

phpMyAdmin

Server: 127.0.0.1 > Database: dati animali > Tabella: parameters

Opzioni extra

	ref_id	date	temperature	heart_rate
<input type="checkbox"/> Modifica	1111	2022-12-27 18:41:44	21.3958	79.1492
<input type="checkbox"/> Modifica	1111	2022-12-27 20:06:51	20.5833	56.7311
<input type="checkbox"/> Modifica	1111	2022-12-27 21:15:36	19.4958	51.1361
<input type="checkbox"/> Modifica	1111	2022-12-27 22:16:17	19.5	49.5142
<input type="checkbox"/> Modifica	1111	2022-12-28 12:25:36	18.3583	81.3403
<input type="checkbox"/> Modifica	1111	2022-12-28 13:26:29	19.2875	56.1458
<input type="checkbox"/> Modifica	1111	2022-12-28 14:27:15	19.5	41.575
<input type="checkbox"/> Modifica	1111	2022-12-28 15:27:59	19.5	41.5235
<input type="checkbox"/> Modifica	1111	2022-12-28 16:29:05	19.8875	52.3058
<input type="checkbox"/> Modifica	1111	2022-12-28 17:29:53	19.8125	55.5109
<input type="checkbox"/> Modifica	1111	2022-12-28 18:30:38	19.5	60.0067
<input type="checkbox"/> Modifica	1111	2022-12-28 20:08:31	19.4958	46.7175
<input type="checkbox"/> Modifica	1111	2022-12-28 21:09:15	19.5	56.2342
<input type="checkbox"/> Modifica	1111	2023-01-08 21:20:16	19.1708	77.4219
<input type="checkbox"/> Modifica	1111	2023-01-08 22:21:02	19.8167	80.75

Nuovo

Nuova

animals

parameters

information_schema

mysql

performance_schema

phpmyadmin

test

Tabella animals

Tabella parameters

Operazioni

Mostra

Struttura

SQL

Cerca

Inserisci

Esporta

Importa

Privilegi

Monitoraggio

Trigger

Opzioni extra

Server: 127.0.0.1 > Database: dati animali > Tabella: animals

14 secondi.)

Codice PHP] [Aggiorna]

Cerca nella tabella

Ordina per chiave: Nessuno

Modifica

Copia

Elimina

Esporta

Importa

Privilegi

Operazioni

Monitoraggio

Trigger

Opzioni extra

Mostra tutti

Numero di righe: 25

Filtra righe: Cerca nella tabella

Ordina per chiave: Nessuno

Operazioni sui risultati della query

Console

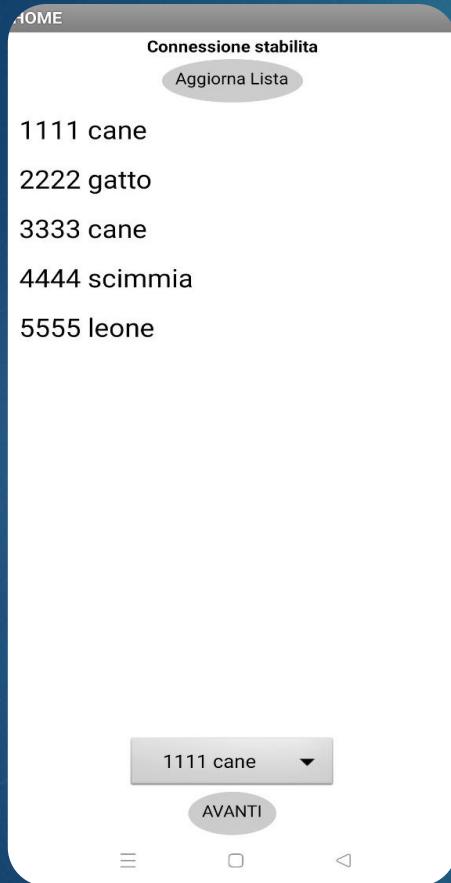
	id	type	last_latitude	last_longitude
<input type="checkbox"/> Modifica	1111	cane	38.1013	13.3345
<input type="checkbox"/> Modifica	2222	gatto	44.0002	43
<input type="checkbox"/> Modifica	3333	cane	44.0003	43.0121
<input type="checkbox"/> Modifica	4444	scimmia	44.0011	43.011
<input type="checkbox"/> Modifica	5555	leone	43	43

Client App



Client App: SBAM App

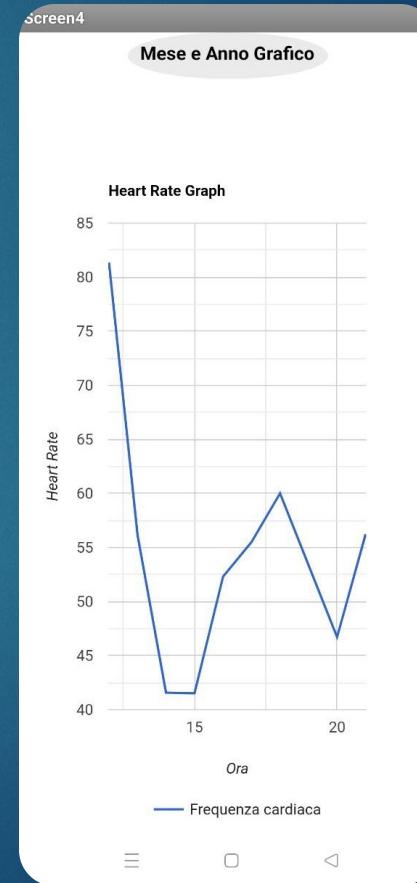
Home



Control panel



Parameter graph



Client App: SBAM App

UrsAI2MQTT Extension → Utilizzo

- ▶ Metodo **Connect**: stabilisce la connessione tra client e broker

L'evento **"ConnectionStateChanged"** segnala qualsiasi modifica nello stato della connessione.

- ▶ Metodo **Disconnect**: cancella la connessione tra client e broker

- ▶ Metodo **Publish**: consente l'invio di messaggi al broker

- ▶ Metodo **Subscribe**: determina a quali argomenti il client desidera ricevere i messaggi

- ▶ Metodo **Unsubscribe**: annulla l'iscrizione agli argomenti

L'evento **"PublishReceived"** segnala la ricezione di un messaggio sull'argomento sottoscritto.

Client App: SBAM App

Lista dei topic utilizzati

In questo caso, l'identificativo 1111 fa riferimento ad uno specifico animale in possesso del bracciale.

In generale, perciò, è possibile ottenere i valori riportati dai sensori per ciascun animale in possesso del SBAM, modificando semplicemente l'identificativo univoco nei topic utilizzati.

- ▶ **alarm/send/1111** (server-bracciale) → accensione o spegnimento allarme
- ▶ **animals/+** (server-database) → richiesta della liste dei dispositivi
- ▶ **animals/temperature/+** (server-database) → richiesta temperatura animale in un mese particolare
- ▶ **animals/heartRate/+** (server-database) → richiesta frequenza cardiaca in un mese particolare
- ▶ **animals/last_position/+** (server-database) → richiesta dell'ultima posizione di un dispositivo
- ▶ **average/+** (server-database) → salvataggio dati di un dispositivo
- ▶ **last_position/+** (server-database) → salvataggio ultima posizione di un dispositivo
- ▶ **server/animals/** (app-server) → richiesta della lista dei dispositivi
- ▶ **server/animals/response/** (database-server) → risposta della lista dei dispositivi

Client App: SBAM App

Lista dei topic utilizzati

- ▶ **server/temperature/+** (bracciale-server) → temperatura in tempo reale
- ▶ **server/heart_rate/+** (bracciale-server) → frequenza cardiaca in tempo reale
- ▶ **server/gps/+/+** (bracciale-server) → posizione in tempo reale (latitudine e longitudine)
- ▶ **server/alarm/+** (bracciale-server) → errori per allarmare in tempo reale
- ▶ **server/alarm/rec/+** (bracciale-server) → stato allarme in tempo reale
- ▶ **server/alarm/send/+** (app-server) → accensione o spegnimento allarme
- ▶ **server/animals/last_position/** (app-server) → richiesta ultima posizione
- ▶ **server/animals/last_position/+** (database-server) → risposta ultima posizione
- ▶ **server/animals/temperature/** (app-server) → richiesta temperatura in un particolare mese
- ▶ **server/animals/temperature/+** (db-server) → risposta temperatura in un particolare mese
- ▶ **server/animals/heartRate** (app-server) → richiesta frequenza cardiaca in un mese
- ▶ **server/animals/heartRate/+** (db-server) → risposta frequenza cardiaca in un mese
- ▶ **custode/alarm/+** (server-custode) → eventuali errori per allarmare in tempo reale

Client App: SBAM App

Lista dei topic utilizzati

- ▶ **custode/alarm/rec/+** (server-custode) → stato allarme in tempo reale
- ▶ **custode/alarm/send/+** (server-custode) → spegnimento allarme
- ▶ **animals/response/** (server-app) → risposta della lista dei dispositivi
- ▶ **temperature/1111** (server-app) → temperatura in tempo reale del dispositivo
- ▶ **heart_rate/1111** (server-app) → frequenza cardiaca in tempo reale del dispositivo
- ▶ **alarm/1111** (server-app) → invio allarme utente
- ▶ **alarm/rec/1111** (server-app) → stato allarme in tempo reale
- ▶ **animals/last_position/1111** (server-app) → risposta dell'ultima posizione di un dispositivo
- ▶ **gps/lon/1111** (server-app) → longitudine in tempo reale di un dispositivo
- ▶ **gps/lat/1111** (server-app) → latitudine in tempo reale di un dispositivo
- ▶ **notify/alarm/rec** (server-app) → accensione allarme dei vari dispositivi
- ▶ **animals/temperature/1111** (server-app) → risposta temperatura in un mese di un dispositivo
- ▶ **animals/heartRate/1111** (server-app) → risposta frequenza cardiaca in un mese di un dispositivo

GRAZIE PER L'ATTENZIONE