# Interazione Fisica e Pensiero Computazionale

Gionata Massi*
*IIS "Savoia Benincasa", Ancona*

Emanuele Lorenzoni



ITADINFO 2026 · Padova · 09/10/2026

[ITADINFO]

# Percorso della presentazione

1. Introduzione
    1. Obiettivi
    2. Metodo e strumenti
    3. La grande idea
2. La progressione didattica
    1. Incontro 1 — Familiarizzare con Micro:bit
    2. Incontro 2 — Descrivere stati osservabili
    3. Incontro 3 — Conservare informazione
    4. Incontro 4 — Codificare e far evolvere gli stati
    5. Incontro 5 — Gestione del tempo e progetti di gruppo
3. Valutazione e risultati osservati
    1. Criticità
    2. Conclusioni

# Introduzione

## Contesto e destinatari

- Progetto finanziato con fondi del DM 65/2023 (STEM)
- IIS “Savoia Benincasa”, Ancona
- Classe seconda ITE AFM
- 10 ore: cinque incontri da due ore
- Nessuna competenza di programmazione richiesta

## Prerequisiti reali

- Leggere consegne operative
- Distinguere causa ed effetto
- Usare i numeri naturali per contare e misurare intervalli di tempo

# Obiettivi

## Riferimento alle indicazioni CINI

## Traguardi di competenza

**T-S-6.** Definisce, realizza e valida programmi e sistemi che modellano o simulano sistemi fisici o processi familiari del mondo reale o oggetto di studio nelle altre discipline.

**T-S-10.** Sceglie e riconosce nei programmi la rappresentazione dei dati dei problemi, dei risultati che ottiene e degli elementi utili a tener traccia degli stadi intermedi dell'elaborazione.

## Obiettivi di apprendimento

**O-S-P-1.** Riconoscere come le varie parti di un programma contribuiscono al suo funzionamento.

**O-S-P-2.** Predire il risultato di un programma senza farlo eseguire.

**O-S-P-4.** Utilizzare cicli con condizioni per descrivere l'esecuzione di azioni parametriche.

**O-S-N-1.** Realizzare esperienze di raccolta ed analisi dati attraverso sensori e di controllo di dispositivi esterni.

**O-S-D-1.** Valutare vantaggi e svantaggi di rappresentazioni alternative della stessa informazione *(riferimento introduttivo)*.

# Metodo e strumenti

## Strumenti

- Micro:bit: pulsanti e sensori come ingressi; display LED come uscita
- MakeCode: linguaggio a blocchi, simulatore, debugger e caricamento sul dispositivo

## Cornice pedagogica

- Ciclo di apprendimento esperienziale di Kolb

| | |
|---|---|
| Esperienza concreta | FARE → |
| Osservazione riflessiva | DESCRIVERE → |
| Concettualizzazione astratta | MODELLIZZARE → |
| Sperimentazione attiva | MODELLIZZARE → DESCRIVERE → FARE |

# La grande idea

## Un comportamento si può descrivere: la macchina a stati

- **Eventi (ingressi + tempo):** pulsanti, sensori, scadenze temporali
- **Transizioni:** eventi e condizioni determinano il passaggio da uno stato all'altro
- **Uscite osservabili:** azioni come mostrare o riprodurre
- **Stati:** configurazioni del sistema, rappresentate nel programma anche tramite variabili

**Modello informale**: abbastanza rigoroso da discutere il programma ma senza introdurre la teoria degli automi

Esempi MakeCode: eventi · stato · ciclo · timer

# La progressione didattica

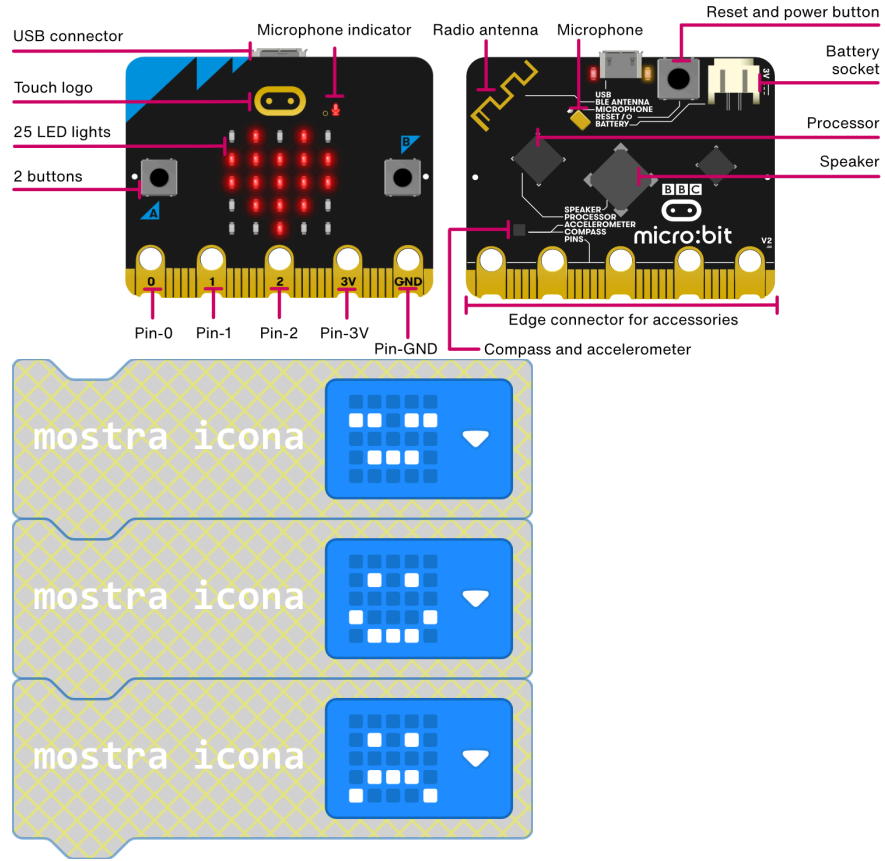
| Incontro | Attività |
|---|---|
| 1 | Familiarizzazione con Micro:bit e tutorial introduttivi |
| 2 | Transizioni guidate dagli input |
| 3 | Variabili di stato |
| 4 | Sequenze cicliche, codifica degli stati e selezione |
| 5 | Transizioni temporizzate e mini-progetti |
| - | Test e questionario |

# Incontro 1 — Familiarizzare con Micro:bit

- Conoscere il display LED e i pulsanti
- Esplorare i tutorial: *Pet Hamster*, *Flashing Heart*, *Name Tag* e *Smiley Buttons*
- Osservare output visivi e testuali; sperimentare gli input dei pulsanti
- Confrontare simulatore e dispositivo

**Domanda guida:** come cambia l'effetto osservabile quando modifico il programma?

**Sperimentazione attiva:** personalizzare un tutorial a partire dal comportamento osservato



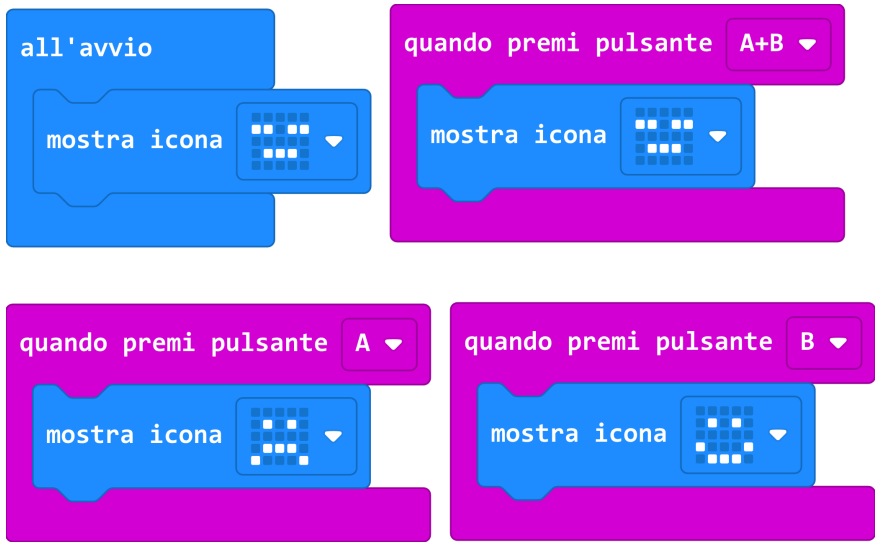
# Incontro 2 — Descrivere stati osservabili

- Rappresentare stati con comportamenti e icone riconoscibili
- Provare input diversi e osservare le transizioni
- Passare dalla sequenza di azioni a una descrizione del sistema

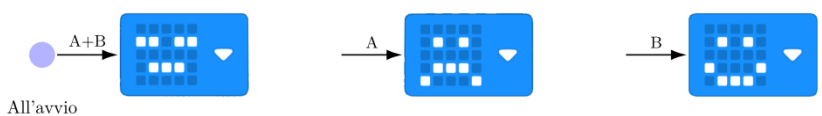
**Non solo “che cosa fa”: anche “in quale stato è?”**

**Domanda guida:** lo stato cambia anche senza un nuovo input?

**Sperimentazione attiva:** un primo Tamagotchi

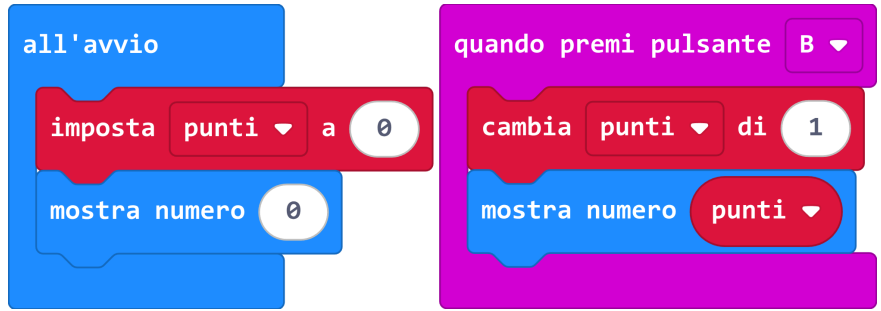



## Macchina a stati

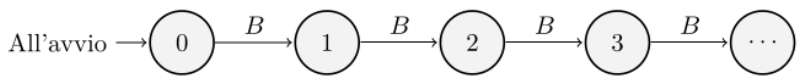


# Incontro 3 — Conservare informazione

- Il programma deve ricordare un valore tra un evento e il successivo
- La **variabile** conserva una parte dello stato del sistema
- **Input:** pulsante · **Stato:** punteggio · **Output:** valore mostrato




## Macchina a stati




**Sperimentazione attiva:** segnapunti; estensione: punteggi di due squadre o differenza reti

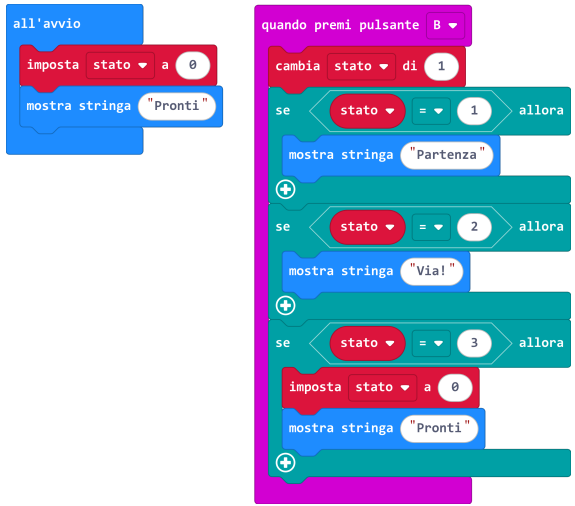
# Incontro 4 — Codificare e far evolvere gli stati

- Rappresentare gli stati con valori numerici
- Usare condizioni per scegliere l'azione successiva
- Esplorare una sequenza ciclica: "Pronti, partenza, via!"

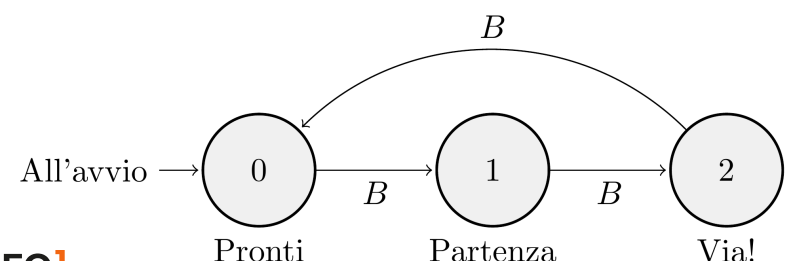
**Previsione:** se lo stato è 2 e si preme B, che cosa succede?

**Sperimentazione attiva:** operatore resto e debugger

**Sperimentazione attiva:** se/allora/altrimenti



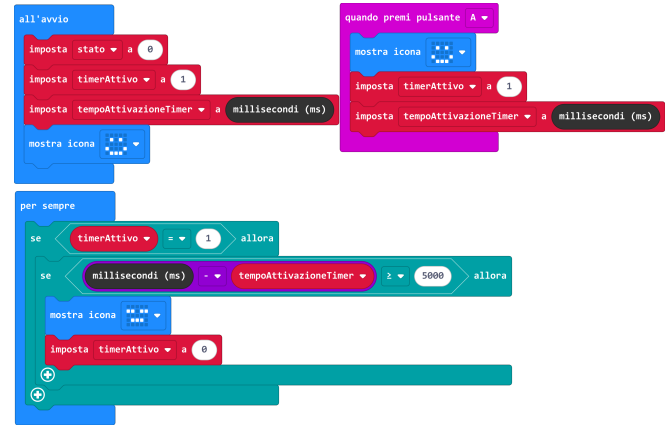

## Macchina a stati



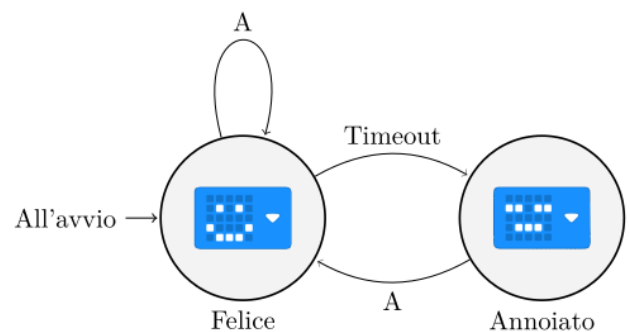

[ITADINFO]

# Incontro 5 — Gestione del tempo e progetti di gruppo

- L'animale si sveglia felice ma, dopo un certo tempo senza stimoli... «all'avvio», stato iniziale, timer attivo, istante
- ...si annoia. timer disattivato
- Uno stimolo lo rende temporaneamente felice timer attivo, istante
- Il tempo deve essere gestito! «Per sempre»




## Macchina a stati




[ITADINFO]

# Valutazione e risultati osservati

## Test finale

Lettura dei blocchi · input, output e stato · «all'avvio» e «per sempre» · previsione delle transizioni

## Questionario di gradimento

- Il corso è stato apprezzato, ma non è stato percepito come pienamente rispondente alle esigenze degli studenti
- Non si è realizzato l'atteso aumento delle iscrizioni all'indirizzo **Sistemi Informativi Aziendali (SIA)**

## Esito del test

- Risultati mediamente buoni su variabili, struttura dei programmi e diagrammi di stato
- Restano difficoltà nel classificare input/output e nel ragionare sul ciclo temporizzato

# Criticità

## Fare non significa ancora saper spiegare

- La competenza pratica non sempre si accompagna a una spiegazione formale
- Codificare una macchina a stati temporizzata non è banale!
- Variabili di stato, *scan loop*, strutture di controllo annidate
- **Gradimento != Orientamento**

# Conclusioni

- La Micro:bit rende osservabile il legame tra programma e mondo fisico
- Eventi, variabili e transizioni rendono descrivibile il comportamento
- Il valore formativo emerge quando il ragionamento è esplicitato
- Non solo far funzionare un programma: descriverlo, prevederlo e discuterlo

# Sviluppi per una nuova edizione

- Più tempo per consolidare concetti e lessico
- Una ricetta di macchina a stati senza temporizzazione
- Verbalizzare input, stato, output e transizioni durante le attività
- Esercitare il passaggio diagramma → programma

# Interazione Fisica e Pensiero Computazionale

Gionata Massi*
*IIS "Savoia Benincasa", Ancona*

Emanuele Lorenzoni

## Grazie per l'attenzione!

Domande?