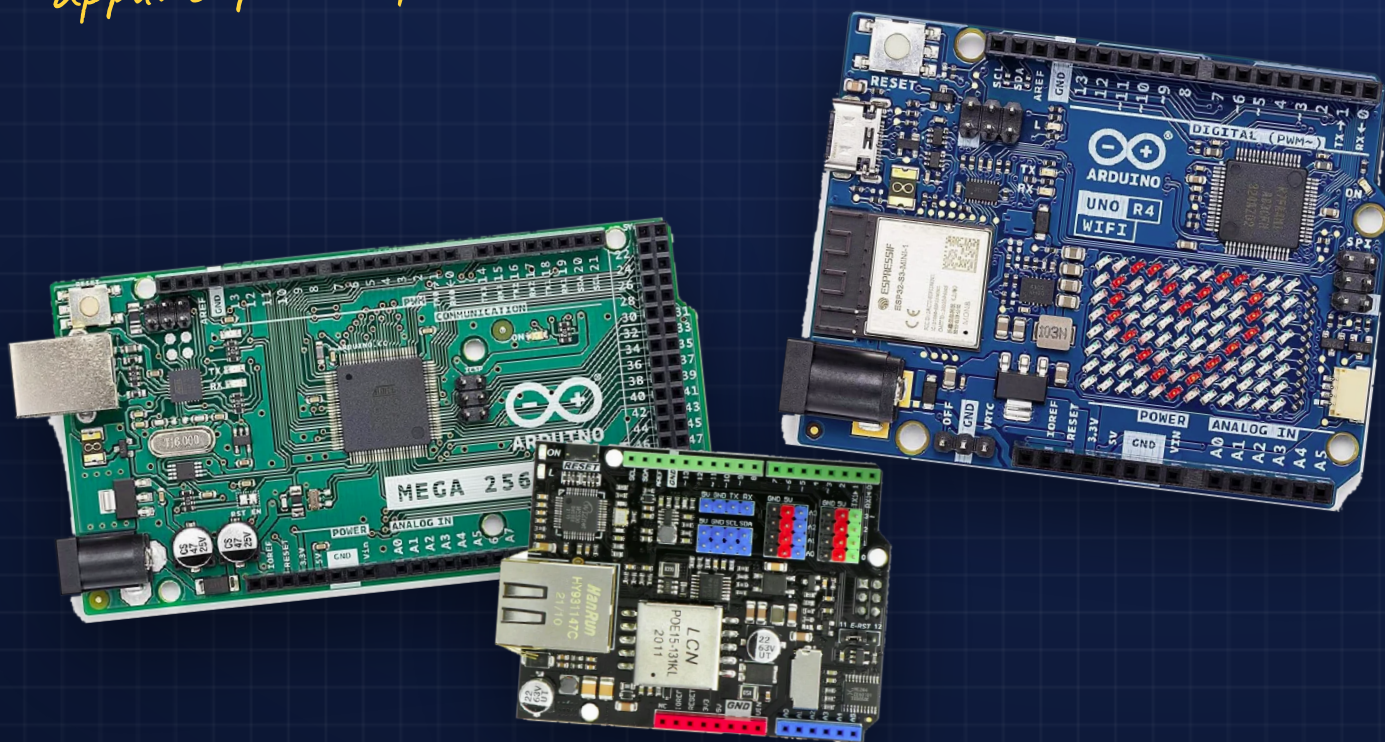


Manuale del docente

Dalla prima accensione alla lezione in laboratorio:
configurazione, linguaggi, esercizi e limiti

appunti pratici per chi insegna automazione ✎



Indice

1	Che cos'è ArduLearn	l'idea, cosa serve, come è fatto
2	Le due schede	UNO R4 WiFi e Mega 2560: differenze e pin
3	L'app ArduLearn per Windows	le quattro schede dell'app
4	Prima configurazione, passo passo	firmware, rete, prima apertura, PIN
5	La pagina di ArduLearn	giro guidato, docente e studenti
6	Come ragiona un PLC	il ciclo, i segnali, i numeri, il tempo
7	Pin e variabili	ingressi, uscite, pull-up, indirizzi
8	Programmare a blocchi (FBD)	logica, uso dell'editor, catalogo dei blocchi
9	Programmare in LADDER	contatti, bobine, box, rami
10	Programmare in GRAFCET	tappe, transizioni, azioni, sicurezze
11	Script ST e funzioni C	quando il disegno non basta
12	Display, matrice LED e immagini	OLED, LCD, pagine, animazioni
13	Moduli: sensori e attuatori	temperatura, distanza, servo, LED...
14	Collegamenti elettrici	schemi e regole di sicurezza
15	Salvare e organizzare i programmi	file, slot, esempi, bozza
16	Aggiornamenti	UNO R4 WiFi dalla rete, Mega con l'app
17	Esercizi guidati	sette esercitazioni con soluzioni
18	Limiti del prodotto	cosa non fa e perché
19	Problemi e soluzioni	le domande più frequenti
A	Appendice	comandi del monitor seriale, glossario, contatti

*Come leggere questo manuale: i riquadri **rossi** sono attenzioni, quelli **blu** informazioni utili, i foglietti gialli sono consigli "da collega". Sulle schermate le frecce e le scritte a mano indicano dove cliccare.*

1 Che cos'è ArduLearn

Un PLC da laboratorio, costruito su una scheda Arduino, che si programma dal browser.

Un **PLC** (controllore logico programmabile) è il "cervello" delle macchine automatiche: legge pulsanti e sensori, decide cosa fare secondo un programma e comanda lampade, motori, valvole. In azienda si programma con software dedicati e linguaggi standard.

ArduLearn trasforma una scheda Arduino in un piccolo PLC didattico: gli studenti scrivono il programma in una pagina web, con gli stessi linguaggi dell'industria, lo caricano sulla scheda e lo vedono funzionare **dal vivo**, con i collegamenti che si colorano quando sono attivi.

Cosa si può insegnare

- **Logica combinatoria:** AND, OR, NOT, XOR con pulsanti e lampade.
- **Memorie e autoritenuta:** il classico marcia/arresto di un motore.
- **Temporizzatori e contatori:** luce delle scale, contapezzi, lampeggi.
- **Sequenze:** semafori, carrelli, macchine a fasi con il GRAFCET.
- **Grandezze analogiche:** potenziometri, temperature, soglie con isteresi, regolazione PWM.
- **Programmazione testuale** (facoltativa): testo strutturato ST e piccole funzioni in C.

Per gli studenti non c'è niente da installare: serve solo un browser (Chrome, Edge, Firefox) su un PC della stessa rete della scheda.

Cosa serve

Oggetto	A cosa serve
Una scheda: Arduino UNO R4 WiFi oppure Arduino Mega 2560 con la shield DFRobot Ethernet & PoE (DFR0850)	È il PLC: esegue il programma e comanda ingressi e uscite.
Un PC Windows con l'app ArduLearn.exe	Solo per installare ArduLearn sulla scheda (e ogni tanto per aggiornarla o diagnosticarla). Poi non serve più.
Un cavo USB (USB-C per la UNO R4, USB-B per la Mega)	Per la prima installazione e per alimentare la scheda.
La rete: Wi-Fi della scuola (UNO R4) o cavo di rete (Mega)	I PC degli studenti aprono la pagina della scheda attraverso la rete.
Facoltativi: pulsanti, LED, relè, sensori, display OLED o LCD	Per costruire impianti veri sul banco. Si può anche iniziare senza nulla, con i pulsanti virtuali della pagina.

Come è fatto il sistema



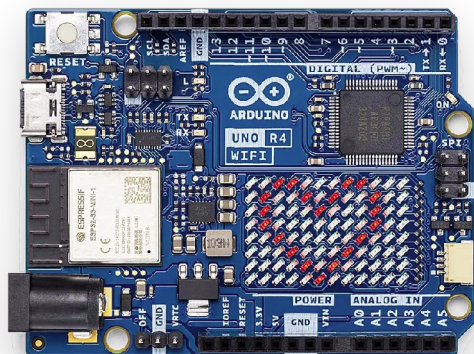
La scheda contiene sia il PLC sia un piccolo **sito web**: la pagina di ArduLearn. Chi apre l'indirizzo della scheda trova l'editor, scrive o modifica il programma e, se ha il **PIN del docente**, lo carica. Da quel momento la scheda lavora da sola: anche se si chiude il browser o si spegne il PC, il programma continua a girare, e resta in memoria anche togliendo l'alimentazione.

Tre linguaggi, uno per volta

Ogni progetto si scrive in **uno** dei tre linguaggi: **Blocchi (FBD)**, **LADDER** (a contatti, stile Siemens TIA Portal) o **GRAFCET** (sequenze a tappe, con segmenti LADDER). Un programma non si converte da un linguaggio all'altro: si sceglie all'inizio, con .

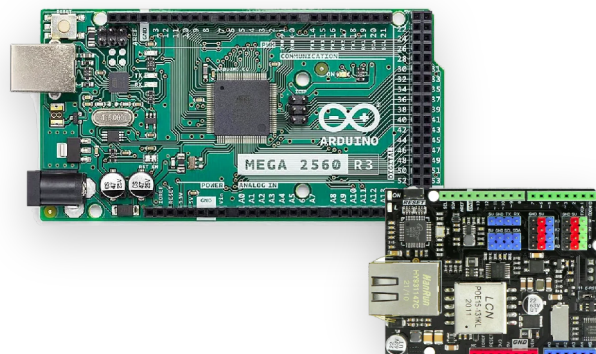
2 Le due schede

Il programma e la pagina sono gli stessi: cambiano il collegamento alla rete e il numero di pin.



Arduino UNO R4 WiFi

Si collega al **Wi-Fi** della scuola, senza schede aggiuntive. Ha a bordo una **matrice di 12×8 LED rossi** che mostra l'indirizzo della scheda e, se il programma lo chiede, numeri, scritte e icone. Si aggiorna **dalla pagina**, anche da Internet.



Arduino Mega 2560 + shield Ethernet PoE

Si collega con il **cavo di rete** (e, con uno switch PoE, prende anche l'alimentazione dal cavo). Ha **molti più pin**, adatti a impianti grandi. I programmi si conservano sulla **microSD** della shield.

Confronto rapido

	UNO R4 WiFi	Mega 2560 + shield DFR0850
Rete	Wi-Fi 2,4 GHz (rete della scuola o rete propria "ArduLearn-xxxx")	Cavo Ethernet (DHCP, oppure indirizzo automatico 169.254.x.y con cavo diretto al PC)
Pin digitali	D0–D13	D0–D53 (alcuni riservati, vedi sotto)
Ingressi analogici	A0–A5 (A4 e A5 riservati se si usano display o orologio)	A0–A15
Uscite PWM (~)	3, 5, 6, 9, 10, 11	2–13, 44–46
Programma attivo	fino a 7.680 byte, nella memoria interna	fino a 8.192 byte sulla microSD (3.072 byte senza microSD)
Programmi salvati	99 posti nella memoria del modulo Wi-Fi	99 posti sulla microSD (senza microSD: nessuno)
Matrice LED	sì (12×8)	no
Aggiornamenti	dalla pagina, anche da Internet, oppure con l'app	solo con l'app, via USB
Studenti collegati insieme	fino a circa 7 collegamenti contemporanei	una richiesta alla volta: con molti PC la pagina può rallentare

Pin riservati

Alcuni pin sono già usati dalla scheda stessa e nella pagina compaiono bloccati, con il motivo scritto accanto:

- **Mega:** D0 e D1 (collegamento USB), D4 (microSD), D10 (rete), D20 e D21 (display e orologio), D50–D53 (rete e microSD).
- **UNO R4 WiFi:** solo A4 e A5 (display e orologio). D13 è collegato al LED "L" della scheda: comodo per le prime prove.
- Un **display LCD collegato in parallelo** occupa altri 6 pin (quelli scelti nelle Impostazioni).

Reti Wi-Fi che non funzionano

La UNO R4 WiFi non si collega alle reti che chiedono **nome utente e password personali** (per esempio eduroam o reti "WPA2-Enterprise") né alle reti a 5 GHz. Alcune reti scolastiche inoltre **isolano i dispositivi** fra loro: i PC non riescono a vedere la scheda. In questi casi chiedi al tecnico una rete dedicata al laboratorio (anche un semplice router), oppure usa la Mega con il cavo.

*Quale scegliere? Per iniziare e per le aule con buon Wi-Fi: **UNO R4 WiFi**. Per banchi con molti ingressi/uscite o reti Wi-Fi difficili: **Mega con cavo di rete**.*

Tornare a usare la scheda come un Arduino normale

UNO R4 WiFi: ArduLearn sostituisce anche il firmware del **modulo Wi-Fi** della scheda. Gli sketch normali si caricano ancora dall'IDE di Arduino, ma le librerie Wi-Fi di Arduino (e gli esempi che usano la rete) funzionano solo dopo aver **riscritto il firmware originale del modulo Wi-Fi**: nell'IDE Arduino apri **Strumenti** → **Aggiornamento firmware** (Firmware Updater), scegli la UNO R4 WiFi e installa la versione proposta. Per rimettere ArduLearn basta di nuovo l'app.

Mega 2560: basta caricare un qualsiasi sketch dall'IDE; per tornare ad ArduLearn si ricarica il firmware con l'app.

3 L'app ArduLearn per Windows

Il programma da installare una sola volta sul PC del docente: prepara le schede, le trova in rete, fa diagnosi.

L'app si scarica da GitHub ed è un singolo file, **ArduLearn.exe**, che non richiede installazione: basta avviarlo.

github.com/danielemichelotti/ArduLearn/releases/latest/download/ArduLearn.exe

Avviso di Windows

Al primo avvio Windows può mostrare "Windows ha protetto il PC", perché l'app non è firmata da un'azienda. Clicca **Ulteriori informazioni** e poi **Esegui comunque**. Se il PC è gestito dalla scuola, potrebbe servire il tecnico.

L'app ha quattro schede in alto: **Trova le schede**, **Carica il firmware**, **Monitor seriale** e **Guida**.

Carica il firmware



La scheda "Carica il firmware" con una UNO R4 WiFi collegata.

1 Porta USB: la porta a cui è collegata la scheda (la lista si aggiorna da sola).

2 Scheda: si imposta da sola riconoscendo la scheda; va scelta a mano solo per le schede compatibili non originali.

3 Firmware incluso: le versioni che l'app installerà.

4 Carica il firmware: installa ArduLearn. Per la UNO R4 WiFi prepara prima anche il modulo Wi-Fi (fino a un minuto).

I programmi salvati sulla scheda **non vengono cancellati** da un nuovo caricamento del firmware. Durante il caricamento non scollegare il cavo e non chiudere l'app.

Trova le schede

ArduLearn il PLC da laboratorio

Trova le schede

Carica il firmware

Monitor seriale

Guida

ArduLearn è gratuito: se ti è utile, offrirmi un caffè

[Fai una donazione](#)

ideato e realizzato da Daniele Michelotti

Schede ArduLearn trovate nella rete

Nome	Indirizzo IP	Scheda	Rete	Stato	Programma	Pagina
ardulearn-2b28	192.168.254.47	UNO R4 WiFi	Wi-Fi Scuola (segnale 78%)	RUN	Nuovo progetto	aggiornata

la tua scheda: nome, indirizzo, stato

doppio clic sulla riga = apre la pagina

1 Apri nel browser

2 Prepara la microSD / Aggiorna la pagina

3 Cerca di nuovo

1 scheda trovata (ultimo aggiornamento 16:08:30)

Pagina per la microSD inclusa nel programma: versione 3544555

Le schede ArduLearn accese nella stessa rete del PC compaiono da sole.

1 **Apri nel browser** : apre la pagina della scheda selezionata (anche con doppio clic).

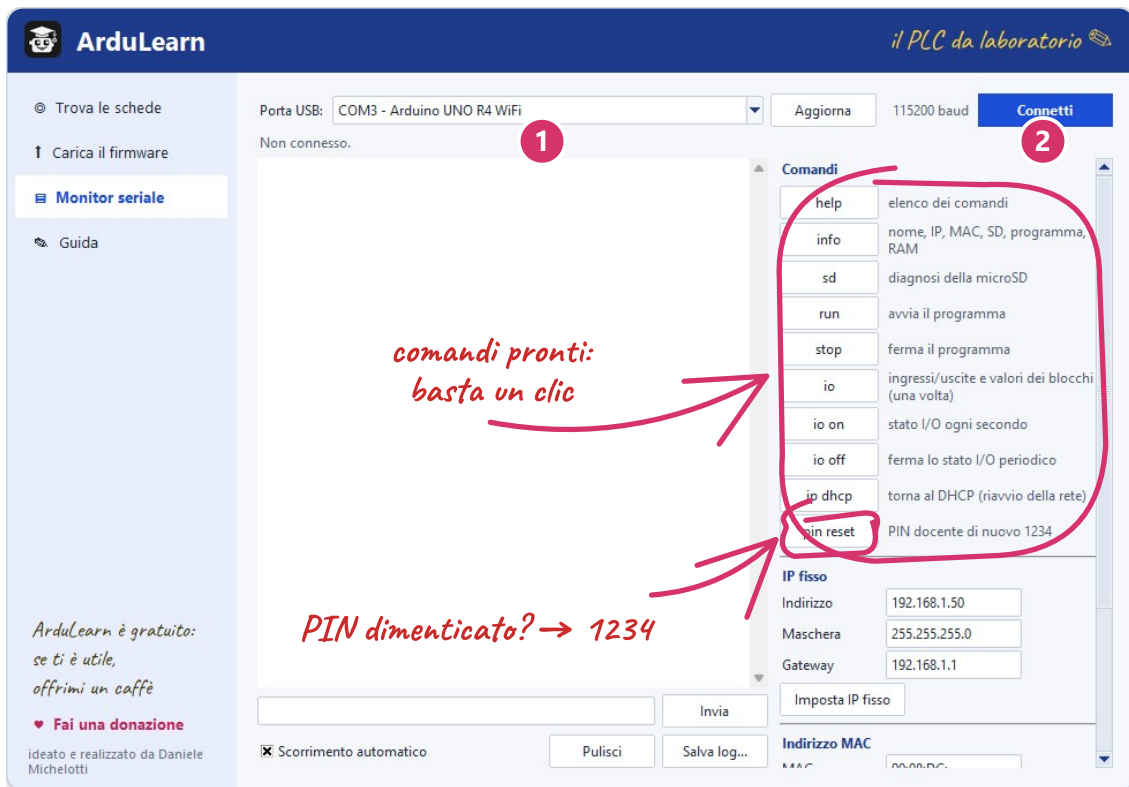
2 **Prepara la microSD / Aggiorna la pagina** : copia la pagina web sulla microSD della Mega (o nella memoria della UNO R4). Chiede il PIN del docente.

3 **Cerca di nuovo** : ripete subito la ricerca.

Le colonne mostrano nome, indirizzo, tipo di scheda, rete, stato (RUN/STOP), nome del programma e se la pagina è *aggiornata* o *da aggiornare*. Se non compare nulla, l'app stessa elenca le cause più probabili (stessa rete, firewall, reti che isolano i dispositivi).

Scrivi il nome di ogni scheda su un'etichetta attaccata alla scheda stessa: ardulearn-2b28, banco3... Così in classe nessuno sbaglia indirizzo.

Monitor seriale



Il monitor seriale, per parlare con la scheda attraverso il cavo USB.

Serve per la **diagnosi**: scegli la porta (1), premi **Connetti** (2) e usa i pulsanti a destra. I più utili sono **info** (nome, indirizzo, programma), **run** / **stop** e soprattutto **pin reset**, che riporta il PIN del docente a **1234**. Per la Mega ci sono anche le impostazioni di **IP fisso** e **indirizzo MAC**; per la UNO R4 i comandi **wifi** (stato della rete) e **wifi clear** (dimentica la rete Wi-Fi salvata). L'elenco completo è in Appendice.

Aprendo il monitor seriale la **Mega si riavvia**: è normale, il programma riparte da capo dopo pochi secondi.

Guida

La quarta scheda contiene un riassunto di queste istruzioni, sempre a portata di mano anche senza il manuale.

4 Prima configurazione, passo passo

Da scheda nuova a scheda pronta per la lezione: circa dieci minuti, una volta sola.

4.1 Installare ArduLearn sulla scheda

- 1 Collega la scheda al PC con il cavo USB. Se è la prima volta, Windows installa da solo il driver (qualche secondo).
- 2 Avvia **ArduLearn.exe** e apri la scheda **Carica il firmware**. Controlla che in **Porta USB** compaia la tua scheda (es. "COM3 - Arduino UNO R4 WiFi").
- 3 Chiudi ogni altro programma che usa la scheda (per esempio l'IDE di Arduino), poi premi **Carica il firmware** e conferma.
- 4 Aspetta il messaggio "**Firmware caricato correttamente. La scheda si riavvia da sola.**" Con la UNO R4 WiFi la prima volta ci vuole un po' di più, perché l'app prepara anche il modulo Wi-Fi.

Se il caricamento non riesce

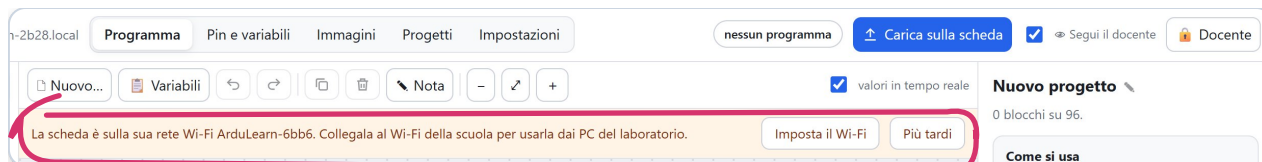
L'app scrive sotto il motivo e un suggerimento. I casi più comuni: la porta è occupata da un altro programma (chiudilo); la UNO R4 non risponde (premi **due volte velocemente** il tasto RESET della scheda e riprova); il cavo USB è solo di ricarica (cambialo).

4.2 UNO R4 WiFi: collegarla al Wi-Fi della scuola

Una scheda appena installata non conosce ancora nessuna rete Wi-Fi. Allora crea **una sua rete**, chiamata **ArduLearn-xxxx** (le ultime quattro lettere cambiano da scheda a scheda), con password **ardulearn**. La si usa solo per la configurazione.



- 1 Con la scheda accesa, sul PC (o su un portatile) apri l'elenco delle reti Wi-Fi e collegati a **ArduLearn-xxxx**, password **ardulearn**. Poi apri il browser e scrivi **http://192.168.4.1**. Se Windows o il telefono avvertono "nessuna connessione a Internet", è normale: resta collegato. Sulla matrice LED scorre la scritta **AP 192.168.4.1**.
- 2 Nella pagina compare un avviso giallo: premi **Imposta il Wi-Fi** (oppure vai in **Impostazioni → Rete Wi-Fi**).



L'avviso che compare finché la scheda è sulla sua rete.

- 3 Premi **Cerca reti** e scegli il Wi-Fi della scuola (1), scrivi la password (2) e premi **Salva e collega** (3). Ti verrà chiesto il PIN del docente: all'inizio è **1234**.

Impostazioni → Rete Wi-Fi.

- 4 La scheda si collega alla rete della scuola e sulla matrice LED fa **scorrere il suo nuovo indirizzo** (per esempio 192.168.1.57). Ricollega il PC al Wi-Fi della scuola e apri quell'indirizzo, oppure trova la scheda con l'app (**Trova le schede**).

E se la password è sbagliata?

Dopo 3 tentativi non riusciti la scheda **torna da sola sulla sua rete** ArduLearn-xxxx: ricollegati e riprova. Se la rete della scuola cade, la scheda si ricollega da sola quando torna. Per far dimenticare la rete:

Dimentica la rete nelle Impostazioni, oppure il comando `wifi clear` dal monitor seriale dell'app.

La matrice LED ripete l'indirizzo ogni volta che la rete cambia e, se il programma non la usa, ogni 2 minuti. Un display OLED o LCD collegato mostra sempre l'indirizzo.

Il nome in rete della scheda è del tipo **ardulearn-2b28**: sui PC che lo supportano la pagina si apre anche con **http://ardulearn-2b28.local**, comodo perché non cambia anche se cambia l'indirizzo numerico.

4.3 Mega: rete via cavo e microSD

- 1 Monta la shield DFR0850 sulla Mega e inserisci una **microSD formattata FAT32** (non exFAT: le schede da 64 GB o più vanno riformattate). La microSD serve per conservare i programmi; senza, la Mega funziona ma tiene un solo programma, più piccolo.
- 2 Collega il **cavo di rete** a una presa della rete del laboratorio. La scheda chiede un indirizzo alla rete (DHCP) e lo mostra sull'eventuale display OLED/LCD.
- 3 Con l'app, **Trova le schede** → seleziona la Mega → [Prepara la microSD / Aggiorna la pagina](#) (PIN 1234). Così la pagina sulla microSD è sempre la più recente.
- 4 Apri la pagina con [Apri nel browser](#).

Senza rete? Cavo diretto al PC

Collegando la Mega direttamente alla presa di rete del PC, senza router, la scheda si dà da sola un indirizzo del tipo **169.254.x.y** dopo circa 10 secondi. L'app la trova lo stesso. Con uno switch o un iniettore **PoE** la shield alimenta anche la scheda dal cavo di rete.

Indirizzo	simulazione
Rete	Collegata (DHCP)
MAC	02:00:00:00:00:01
Scheda	Arduino Mega 2560
microSD	presente
Firmware	1.0
Nome in rete	simulatore .local Salva

Dai a ogni scheda un nome diverso (es. **plc1**, **plc2**) e scrivilo su un'etichetta: gli studenti apriranno **http://plc1.local**. Se il nome non funziona sulla rete della scuola, usa lo script **trova_plc.py** o l'app **ArduLearn** (Trova le schede) dal PC del docente.

Impostazioni → Scheda, sulla Mega: indirizzo, stato della rete, microSD, nome in rete.

Checklist prima della lezione:

- ☐ scheda accesa e in rete
- ☐ indirizzo alla lavagna
- ☐ PIN cambiato (e ricordato!)
- ☐ programma di prova caricato

4.4 Le prime impostazioni (da fare subito)

1

Accesso docente
Accesso attivo. Il PIN è ancora 1234: **cambialo**.

PIN dimenticato? Collega la scheda al PC e scrivi `pin reset` nel monitor seriale (115200 baud); torna a 1234.

2

Scheda
Indirizzo **simulazione**
Rete **Wi-Fi: collegata a CasaMia**
MAC **02:00:00:00:00:01**
Scheda **Arduino UNO R4 WiFi**
Programmi **programma attivo nella scheda, 99 slot e bozza nella memoria del modulo Wi-Fi**
Firmware **1.0 · modulo Wi-Fi simulato**

Dai a ogni scheda un nome diverso (es. **plc1**, **plc2**) e scrivilo su un'etichetta: gli studenti apriranno **http://plc1.local**. Se il nome non funziona sulla rete della scuola, usa lo script **trova_plc.py** o l'app ArduLearn (Trova le schede) dal PC del docente.

3

Rete Wi-Fi
Collegata a **CasaMia** · segnale 78% · indirizzo **127.0.0.1**

Password ☐ mostra

Serve il PIN docente. La password attuale non viene mai mostrata. Se il collegamento non riesce per 3 volte, la scheda torna sulla sua rete "ArduLearn-6bb6" (password ardulearn, pagina <http://192.168.4.1>).

4

Aggiornamento del firmware
PLC (RA4M1) **1.0**
Modulo Wi-Fi (ESP32) **simulato**
Pagina web **-**
Ultima versione **nel simulatore non si controlla**

Durante l'aggiornamento il programma si ferma per qualche secondo e poi riparte da solo.

5

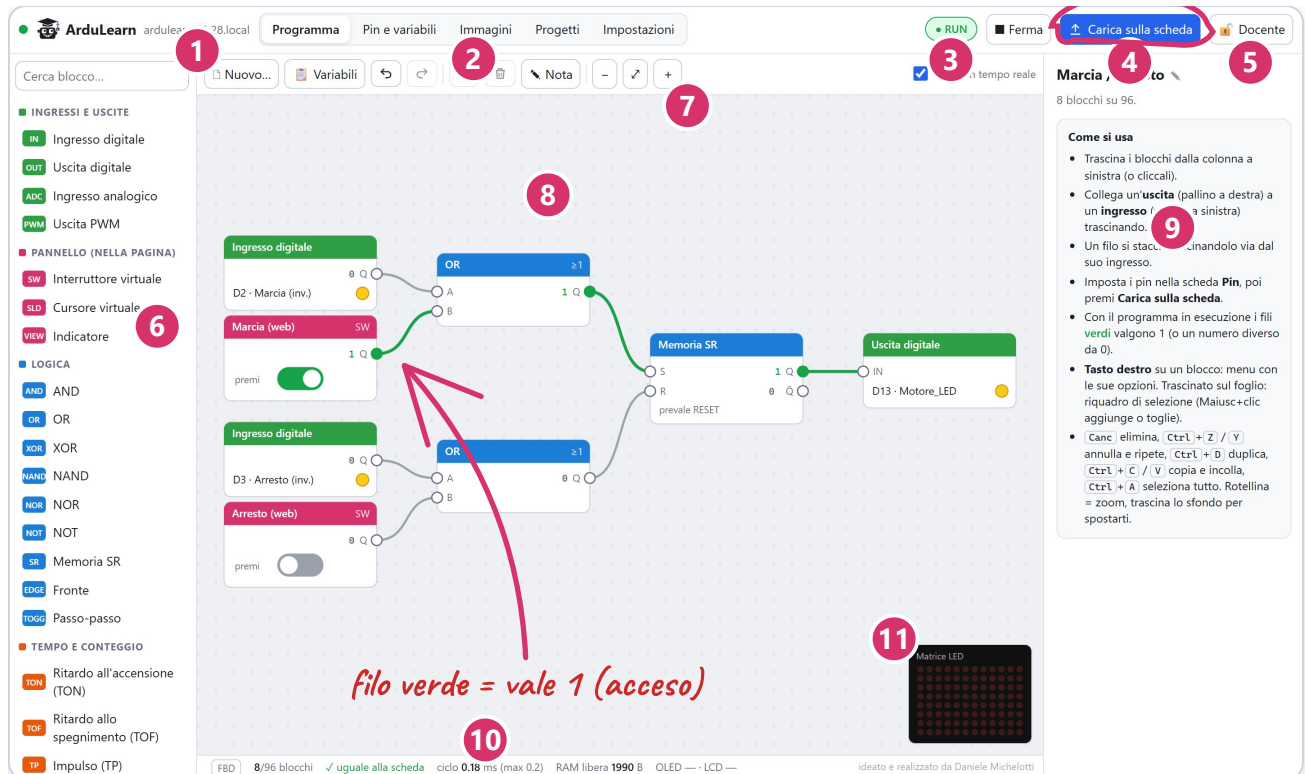
Display OLED (SSD1306 128×64, I²C)
 non rilevato
Se il display mostra pixel accesi a caso o il testo solo in una striscia, scegli l'altro controller.
Il display è facoltativo e si può collegare anche a scheda accesa (viene cercato ogni 3 secondi). Senza blocchi "Scritta su OLED" mostra nome, indirizzo IP e stato della scheda.

La scheda "Impostazioni" della pagina (UNO R4 WiFi).

- 1 Accesso docente:** premi , scrivi 1234 e poi **cambia il PIN** (da 4 a 8 caratteri). Finché resta 1234 la pagina te lo ricorda.
- 2 Scheda:** indirizzo, rete, versione. In **Nome in rete** puoi dare un nome facile, diverso per ogni scheda (es. **banco1**, **banco2**): la pagina diventa <http://banco1.local>.
- 3 Rete Wi-Fi** (solo UNO R4): rete collegata e intensità del segnale; si cambia qui.
- 4 Aggiornamento del firmware** (solo UNO R4): versioni installate e aggiornamenti disponibili (capitolo 16).
- 5 Display OLED e LCD:** tipo e indirizzo del display, pulsante per cercarli. Più sotto: **Orologio** (modulo RTC), **Aspetto** (tema chiaro/scuro).

5 La pagina di ArduLearn

Un giro guidato dell'editor: cosa c'è, cosa fa il docente, cosa possono fare gli studenti.



L'editor con l'esempio "Marcia / arresto" in esecuzione.

- 1 Indirizzo** della scheda. Il pallino a sinistra è verde se la scheda risponde.
- 2 Schede:** Programma, Pin e variabili, Immagini, Progetti, Impostazioni.
- 3 Stato:** ● RUN (in esecuzione), ■ STOP, nessun programma. Accanto il pulsante **Ferma** / **Avvia**.
- 4 Carica sulla scheda:** invia il programma. **Lampeggia** quando il disegno è diverso da quello sulla scheda.
- 5 Docente:** accesso con il PIN (lucchetto aperto = accesso attivo; un clic lo chiude).
- 6 Colonna dei blocchi** (o degli elementi LADDER) con la casella di ricerca.

- 7 Barra degli strumenti:** nuovo progetto, variabili, annulla/ripeti, duplica, elimina, nota, zoom, "valori in tempo reale".
- 8 Foglio di lavoro** con il programma.
- 9 Pannello delle proprietà:** nome del progetto (cliccalo per cambiarlo) o proprietà dell'oggetto selezionato; se non selezioni nulla, un promemoria "Come si usa".
- 10 Barra di stato:** blocchi usati su 96, "✓ uguale alla scheda" o "● modifiche non ancora caricate", durata del ciclo, memoria libera, display collegati.
- 11 Anteprima della matrice LED** (solo UNO R4 WiFi).

Docente e studenti

Chiunque apra la pagina può **guardare, modificare e provare**. Alcune operazioni però cambiano la scheda per tutti e richiedono il **PIN del docente**:

Senza PIN (studenti)	Con il PIN (docente)
Modificare il programma nell'editor, aprire esempi, slot e file, scaricare il progetto come file. Usare interruttori e cursori virtuali, provare le uscite a mano. Vedere tutto dal vivo.	Caricare un programma sulla scheda, avviarlo e fermarlo. Salvare, rinominare ed eliminare i programmi salvati. Cambiare le impostazioni (rete, nome, display, PIN). Aggiornare la scheda.

Il PIN si inserisce premendo **Docente**, oppure viene chiesto al momento del bisogno. Resta valido finché non chiudi la scheda del browser. Se lo dimentichi: app → **Monitor seriale** → **pin reset** (torna a 1234).

Attenzione

Non lasciare l'accesso docente aperto su un PC usato dagli studenti: premi il lucchetto per chiuderlo.



"Segui il docente"

Quando il docente ha fatto l'accesso, ogni sua modifica al progetto viene salvata sulla scheda come **bozza condivisa** (circa un secondo dopo). Gli studenti con la casella **Segui il docente** attiva (lo è di serie) vedono comparire le modifiche in tempo reale sul proprio schermo: la lezione alla lavagna diventa condivisa. Se uno studente ha fatto anche lui delle modifiche, la pagina chiede se **mostrare quello del docente** o **tenere il proprio**.

Sulla Mega la bozza condivisa richiede la microSD.

Il lavoro non si perde

- Il browser ricorda le modifiche non ancora caricate: riaprendo la pagina compare "Ho ripreso le modifiche che non avevi ancora caricato sulla scheda".
- Se qualcun altro carica un programma diverso, la pagina avvisa e fa scegliere se aprire quello della scheda o tenere il proprio.
- Per conservare un lavoro a lungo: **Progetti** → **Scarica file** (un file sul PC) oppure salvalo in uno **slot** della scheda (capitolo 15).

*Per un compito individuale
fai togliere la spunta a
"Segui il docente": ognuno
lavora sul suo progetto
senza essere sovrascritto.*

Caricare, avviare, fermare

- 1 Premi **Carica sulla scheda**. La pagina controlla prima che non ci siano errori (i blocchi con problemi sono segnalati in rosso, con l'elenco nel pannello a destra).
- 2 Il programma parte subito: lo stato diventa **● RUN** e la barra di stato dice "✓ uguale alla scheda".
- 3 Con **■ Ferma** il programma si ferma e le uscite si spengono; **▶ Avvia** lo fa ripartire. Lo stato RUN/STOP viene ricordato anche dopo lo spegnimento.

Durante il caricamento il programma precedente si ferma per qualche secondo. Sulla UNO R4 WiFi il nuovo programma arriva prima nella memoria del modulo Wi-Fi e solo dopo nel PLC: se la rete cade a metà, il programma in esecuzione non si rovina.

6 Come ragiona un PLC

Quattro idee da spiegare prima di qualsiasi linguaggio.

Il ciclo di scansione



Il PLC non "aspetta" gli eventi: ripete in continuazione lo stesso giro. **Legge** tutti gli ingressi, **calcola** il programma dall'inizio alla fine, **aggiorna** le uscite e ricomincia. Un giro si chiama **ciclo** e la sua durata compare nella barra di stato ("ciclo 0.18 ms"): in un programma normale meno di un millisecondo.

Segnali: 0 e 1, e i numeri

- Un segnale **digitale** vale **1** (acceso, vero, 5 volt) oppure **0** (spento, falso). Nella pagina i fili che valgono 1 diventano **verdi**.
- Un valore **analogico** è un numero: un ingresso analogico va da **0** (0 V) a **1023** (5 V); un'uscita PWM da **0** a **255**.
- ArduLearn lavora con **numeri interi**. Per i decimali si usano i **decimi**: 235 significa 23,5 °C (e sui display si scrive `{1.1}` per vedere "23.5").
- Ogni numero diverso da 0 vale "vero" quando entra in un blocco logico.

Il tempo si conta in millisecondi

Tutti i tempi sono in **millisecondi**: 1000 = 1 secondo, 5000 = 5 secondi, 60000 = 1 minuto. In LADDER e GRAFCET si può scrivere anche nel formato dei PLC industriali: `T#5S`, `T#500MS`, `T#1M30S`.

Il fronte: un attimo lungo un ciclo

Molte cose devono succedere **nell'istante in cui** un segnale cambia, non per tutto il tempo in cui resta acceso: contare una pressione, far avanzare una sequenza. Il **fronte di salita** è un impulso che dura **un solo ciclo** quando il segnale passa da 0 a 1 (quello di discesa, da 1 a 0). È troppo breve per vederlo a occhio, ma è quello che fa contare 1 anche se il pulsante resta premuto a lungo.

Esperimento da fare in classe: collega un pulsante a un contatore, prima direttamente e poi attraverso un fronte. Tenendo premuto... cosa cambia?

7 Pin e variabili

Dire alla scheda cosa è collegato a ogni pin, e dare un nome a ingressi, uscite e memorie.

Configurazione dei pin — fa parte del progetto e viene inviata con "Carica sulla scheda"

☐ mostra solo i pin usati Ingresso pull-up: il pin legge 1 a riposo e 0 quando il pulsante lo collega a GND. ~ = supporta PWM.

Modalità simulazione: clicca sul valore di un ingresso per cambiarne lo stato, trascina il cursore degli analogici.

Tabella degli ingressi, delle uscite e delle variabili — come la tabella dei tag di TIA Portal

Nome	Tipo	Indirizzo	Pin	Descrizione	Valore
Marcia	Bool	%I0.0	D2	Ingresso pull-up	TRUE
Arresto	Bool	%I0.1	D3	Ingresso pull-up	TRUE
Motore_LED	Bool	%Q0.0	D13	Uscita	FALSE

I nomi degli ingressi e delle uscite sono le etichette dei pin (modificare qui sotto); gli indirizzi %I/%Q/%IW/%QW vengono assegnati in ordine di pin.

Digitali

D0 Non usato

D1 Non usato

D2 Ingresso pull-up Marcia

D3 Ingresso pull-up Arresto

D4 Non usato

D5 Non usato

D6 Non usato

D7 Non usato

D8 Non usato

D9 Non usato

D10 Non usato

D11 Non usato

D12 Non usato

D13 LED Uscita Motore_LED Prova uscita

Indirizzo per LADDER e GRAFCET

stato dal vivo

= PWM

modo del pin: pull-up per i pulsanti

nome (etichetta)

prova l'uscita a mano

La scheda "Pin e variabili" con l'esempio "Marcia / arresto" (UNO R4 WiFi).

La configurazione dei pin **fa parte del progetto** e viene inviata con "Carica sulla scheda". Di solito non serve toccarla: quando in un blocco scegli un pin libero, la pagina lo configura da sola (e te lo dice con un messaggio).

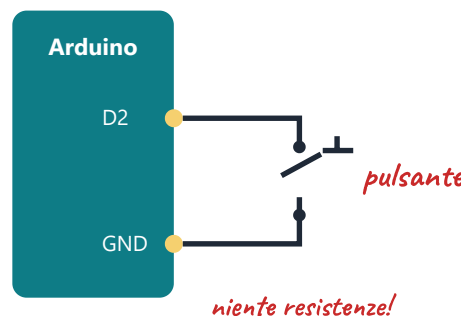
I modi di un pin

Modo	Quando si usa
Non usato	Pin libero.
Ingresso	Sensori che danno già 0 V / 5 V (moduli con uscita digitale).
Ingresso pull-up	Pulsanti, interruttori e finecorsa collegati tra il pin e GND: è il modo consigliato, non servono resistenze.
Uscita	LED, moduli relè, cicalini attivi.
Analogico	Solo pin A: potenziometri, fotoresistenze, sensori analogici.
Modulo	Impostato da solo quando un blocco modulo (sensore, servo...) usa il pin.

Il pull-up e "Inverti"

Con l'ingresso **pull-up** il pin vale **1 a riposo** e **0 quando il pulsante è premuto** (perché lo collega a GND). È il contrario di quello che ci si aspetta! Per questo il blocco "Ingresso digitale" ha l'opzione **Inverti** (in LADDER e GRAFCET: la casella **logica inversa** nella scheda Pin): così **premuto = 1**. Quando la pagina configura un pulsante da sola, attiva già Inverti.

Un pulsante **normalmente chiuso** (come un arresto di emergenza) collegato a GND con pull-up vale 0 a riposo e 1 quando viene premuto o se il filo si rompe: è il collegamento più sicuro per gli arresti.



Nomi e indirizzi

Ogni pin usato può avere un **nome** (es. *Marcia, Motore*): solo lettere, cifre e "_", senza spazi, iniziando con una lettera. I nomi compaiono nei blocchi e sono gli operandi del LADDER.

Nei progetti LADDER e GRAFCET ogni pin riceve anche un **indirizzo** in stile industriale, assegnato in ordine di pin:

Indirizzo	Cosa indica	Esempio
%I0.0 , %I0.1 ...	ingressi digitali	il primo pulsante configurato
%Q0.0 , %Q0.1 ...	uscite digitali	la prima lampada
%IW64 , %IW66 ...	ingressi analogici (numero 0–1023)	un potenziometro
%QW64 ...	uscite PWM (numero 0–255)	un LED regolabile
%M0.0 ... %M7.7	merker: memorie di un bit (vero/falso)	"ciclo avviato"
%MW0 ... %MW30	memorie numeriche (interi)	"pezzi prodotti"

La **Tabella degli ingressi, delle uscite e delle variabili** (come la tabella dei tag di TIA Portal) riassume nome, tipo, indirizzo, pin e **valore dal vivo**. Nei progetti LADDER e GRAFCET, sotto, si aggiungono i merker con **+ Bool (%M)** e **+ Int (%MW)**. Scrivendo in LADDER un nome nuovo, il merker viene creato da solo.

Provare le uscite a mano

Il pulsante **Prova uscita** sotto un'uscita accende e spegne il pin a mano: utilissimo per controllare i collegamenti prima di scrivere il programma. Funziona solo se la configurazione dei pin è già sulla scheda e se quell'uscita non è comandata dal programma in esecuzione (altrimenti prima premi **Ferma**).

Sulla Mega i pin riservati (rete, microSD, display) compaiono bloccati, con il motivo scritto accanto.

8 Programmare a blocchi (FBD)

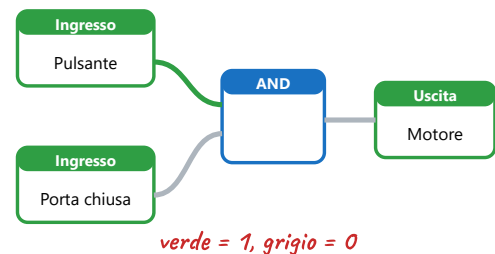
Blocchi collegati da fili, come uno schema funzionale: il modo più intuitivo per cominciare.

La logica

Ogni blocco fa **una cosa sola**: legge un pulsante, fa un AND, aspetta, conta, accende un'uscita. Ha gli **ingressi a sinistra** e le **uscite a destra**. Un **filo** porta il valore di un'uscita nell'ingresso di un altro blocco: il segnale scorre da sinistra verso destra.

La scheda esegue i blocchi **nell'ordine giusto da sola**: prima quelli che forniscono i dati, poi quelli che li usano. Non importa dove li metti sul foglio.

Un ingresso **non collegato** vale 0, oppure il **valore fisso** scritto nelle proprietà (per esempio il tempo PT di un temporizzatore).



Usare l'editor

- 1 **Aggiungere un blocco**: trascinalo dalla colonna di sinistra sul foglio, oppure cliccalo (compare al centro). La casella **Cerca blocco...** trova i blocchi per nome.
- 2 **Collegare**: trascina dal pallino di un'uscita (a destra del blocco) al pallino di un ingresso (a sinistra). I pallini compatibili si evidenziano. Per staccare un filo trascinalo via dal suo ingresso.
- 3 **Impostare**: clicca il blocco e compila le proprietà nel pannello a destra (pin, tempi, valori, etichetta).
- 4 **Caricare** con **Carica sulla scheda** e osservare i fili che si colorano.

Screenshot dell'editor di programmazione a blocchi. In alto a sinistra c'è una barra degli strumenti con icone per "Nuovo...", "Variabili", "Annulla", "Ripristina", "Copia", "Incolla", "Nota", "Zoom in", "Zoom out" e "Reset". In alto a destra c'è un checkbox "valori in tempo reale" che è attivo. Al centro del workspace c'è un blocco "Impulso (TP)" selezionato, con un bordo blu. A sinistra del blocco ci sono due ingressi: "D2 - Pulsante (inv.)" e "Pulsante (web)". A destra c'è un'uscita: "D13 - Luce_LED". A destra del workspace c'è un pannello di proprietà per il blocco "Impulso (TP)". In questo pannello c'è un campo "Ingresso PT (ms)" con il valore "5000" e pulsanti "Duplica" e "Elimina".

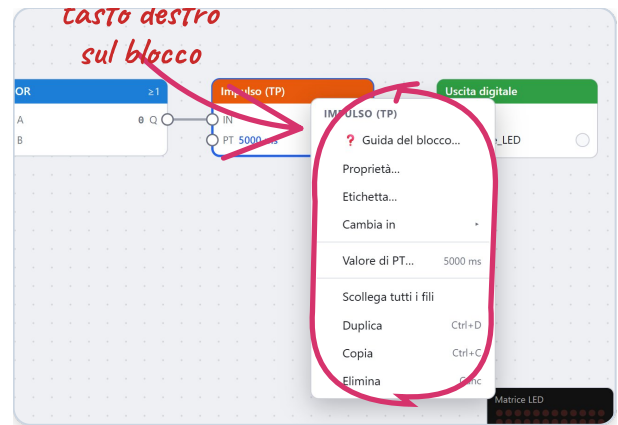
Le annotazioni in rosso indicano:

- clic sul blocco = lo selezioni* (punta al blocco "Impulso (TP)").
- guida del blocco* (punta al pannello di proprietà).
- tempo in millisecondi* (punta al campo "5000").
- 5000 = 5 secondi* (punta al campo "5000").

Il blocco selezionato ha il bordo blu; a destra le sue proprietà.

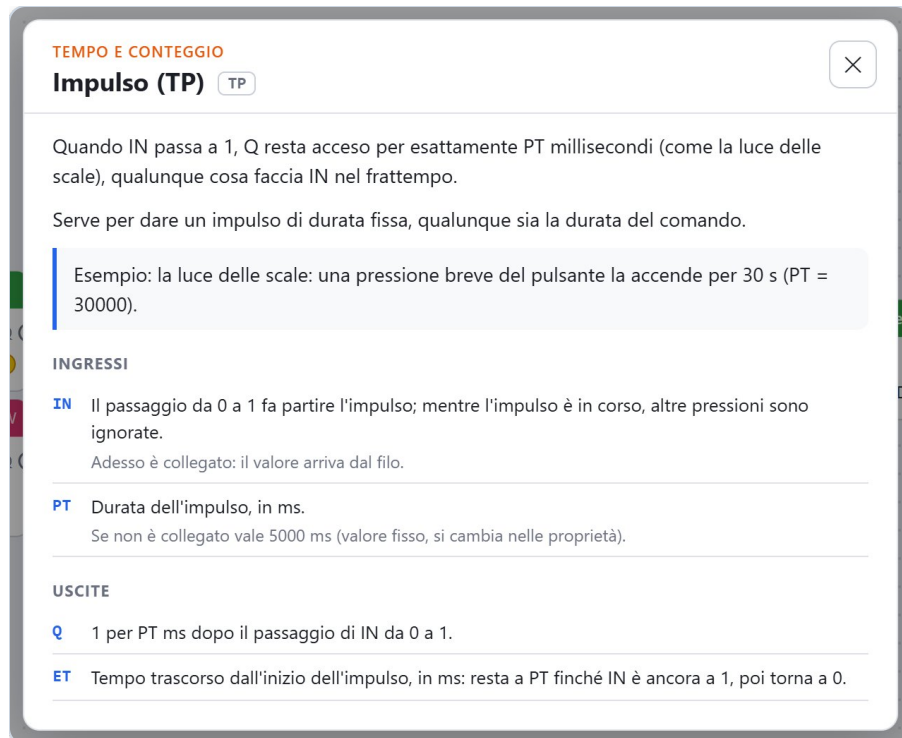
Il tasto destro

Sul blocco apre un menu con: **Guida del blocco**, proprietà, etichetta, **Cambia in** (per esempio da AND a OR o da TON a TOF senza rifare i fili), valori degli ingressi, scollega fili, duplica, copia, elimina. Sul foglio vuoto: incolla, aggiungi nota, seleziona tutto. Trascinando con il tasto destro si seleziona un'area.



Scorciatoie

Ctrl+Z / Ctrl+Y annulla/ripeti · Ctrl+C / Ctrl+V copia/incolla · Ctrl+D duplica · Canc elimina · Ctrl+A tutto · rotellina = zoom · trascina lo sfondo per spostarti.



La "Guida del blocco": a cosa serve, un esempio, ingressi, uscite e parametri, con i valori attuali.

Il pannello virtuale

Tre blocchi esistono solo nella pagina e permettono di provare un programma **senza cablare niente**:

- **Interruttore virtuale**: si clicca sul blocco. Tipo "Interruttore (resta)" oppure "Pulsante (finché premuto)".
- **Cursore virtuale**: si trascina; dà un numero fra Minimo e Massimo.
- **Indicatore**: mostra un valore come numero, lampada o barra.

Funzionano quando il programma è caricato e non ci sono modifiche in sospeso. Anche gli studenti senza PIN possono usarli: **attenzione**, agiscono davvero sulla scheda.

Le note gialle (Nota) sono perfette per scrivere la consegna dell'esercizio direttamente sul foglio.

Catalogo dei blocchi

Numeri fra parentesi = valori proposti. Tutti i blocchi hanno una guida completa con il tasto destro.

Ingressi e uscite

Blocco	Cosa fa	Esempio
Ingresso digitale (IN)	Legge un pin: 1 o 0. Opzione Inverti per i pulsanti con pull-up.	pulsante di marcia su D2
Uscita digitale (OUT)	Accende il pin se l'ingresso è diverso da 0.	LED o relè su D13
Ingresso analogico (ADC)	Legge un pin A: numero da 0 a 1023.	potenziometro su A0
Uscita PWM (PWM)	Uscita "regolabile" da 0 a 255, solo pin con ~.	luminosità di un LED su D9

Pannello (nella pagina)

Interruttore virtuale (SW)	Interruttore o pulsante cliccabile nella pagina.	marcia per le prove
Cursore virtuale (SLD)	Numero regolabile (0...100).	soglia o tempo da provare
Indicatore (VIEW)	Mostra un valore: numero, lampada o barra.	tempo che passa in un TON

Logica

AND · OR · XOR · NAND · NOR	Porte logiche da 2 a 8 ingressi. AND: tutti veri; OR: almeno uno; XOR: diversi; NAND e NOR: il contrario di AND e OR.	luce delle scale con due interruttori (XOR)
NOT	Inverte: 1 diventa 0 e viceversa.	spia "porta chiusa"
Memoria SR	S accende e ricorda, R spegne. Se S e R sono insieme prevale il RESET (scelta consigliata per sicurezza) o il SET.	marcia/arresto del motore
Fronte (EDGE)	Impulso di un ciclo quando l'ingresso sale, scende o entrambi.	contare le pressioni
Passo-passo (TOGGLE)	Ogni pressione su T cambia stato; R spegne.	un solo pulsante accende e spegne

Tempo e conteggio

Ritardo all'accensione (TON)	Q si accende quando IN è rimasto a 1 per PT ms (1000). ET = tempo trascorso.	luce che si accende dopo 5 s
Ritardo allo spegnimento (TOF)	Q resta acceso ancora PT ms dopo che IN torna a 0.	ventola del bagno che gira ancora 60 s
Impulso (TP)	A ogni fronte di IN, Q resta acceso esattamente PT ms.	luce delle scale 30 s
Lampeggiatore (BLINK)	Con EN a 1 lampeggia: ON ms acceso, OFF ms spento (500/500).	spia di allarme
Contatore (CTUD)	CU conta +1, CD -1, R azzera; Q si accende a CV $\geq$ PV (10). Può contare anche all'indietro.	6 bottiglie per cassa

Numeri e confronti

Costante (K)	Un numero fisso da usare in più blocchi.	lo stesso tempo per tre timer
Operazione (MATH)	A+B, A-B, A×B, A÷B (intera), resto, minimo, massimo. Diviso 0 dà 0.	$235 \div 10 = 23$
Confronto (CMP)	Q = 1 se A >, ≥, <, ≤, =, ≠ B.	è buio se luce < 300
Scala (MAP)	Converte un intervallo in un altro (0...1023 → 0...100).	percentuale di un potenziometro
Soglia con isteresi (HYST)	Si accende sopra una soglia e si spegne solo sotto un'altra, più bassa.	termostato: 28 °C / 26 °C
Selettore (SEL)	Passa A se SEL = 0, altrimenti B.	manuale/automatico
Limita (LIM)	Tiene un valore fra un minimo e un massimo.	0...255 prima di un PWM

Display, sequenze e script

Scritta su OLED / LCD	Scrivi un testo con valori ({1}, {2}) in una riga e colonna; può lampeggiare o scorrere.	"Temp: {1.1} C"
Immagine su OLED	Mostra un'immagine o un'animazione disegnata nella scheda Immagini.	icona della ventola
Pagine display	Più schermate sullo stesso display, sfogliabili con pulsanti.	temperatura / umidità / ora
Immagine / Testo su matrice LED	Solo UNO R4 WiFi: icone 12×8 e testi che scorrono.	"23C" che scorre
Passo (sequenza)	Un passo di una sequenza: si attiva con IN, passa al successivo quando T è vero ed è trascorso almeno MIN ms. Uno solo è "passo iniziale".	semaforo a passi
Ripeti N volte	Con START dà N impulsi (ON/OFF ms).	cicalino che suona 3 volte
Script ST / Script C	Un piccolo programma scritto a testo dentro un blocco (capitolo 11).	somma con un ciclo FOR

I blocchi dei **moduli** (sensori e attuatori) sono nel capitolo 13.

Limite: 96 blocchi

Un programma può avere al massimo 96 blocchi. Le porte con più di 4 ingressi e alcuni script usano blocchi "d'appoggio" invisibili che contano anch'essi: il conteggio è sempre nella barra di stato.

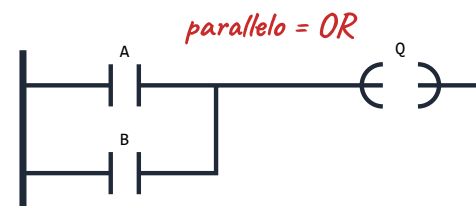
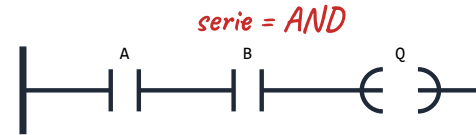
9 Programmare in LADDER

Il linguaggio a contatti, nello stile di Siemens TIA Portal: il programma si disegna come uno schema elettrico.

La logica: la corrente scorre da sinistra a destra

Un programma LADDER è fatto di **segmenti** (righe di schema). A sinistra c'è la **barra di alimentazione**, sempre "in tensione". La corrente scorre verso destra attraversando i **contatti** e arriva alle **bobine**.

- **Contatto NA** (normalmente aperto) : fa passare la corrente se il suo operando vale 1.
- **Contatto NC** (normalmente chiuso) : fa passare la corrente se l'operando vale 0.
- **Bobina** : il suo operando diventa 1 se arriva corrente, 0 se non arriva.
- Contatti **in serie** = AND; rami **in parallelo** = OR.

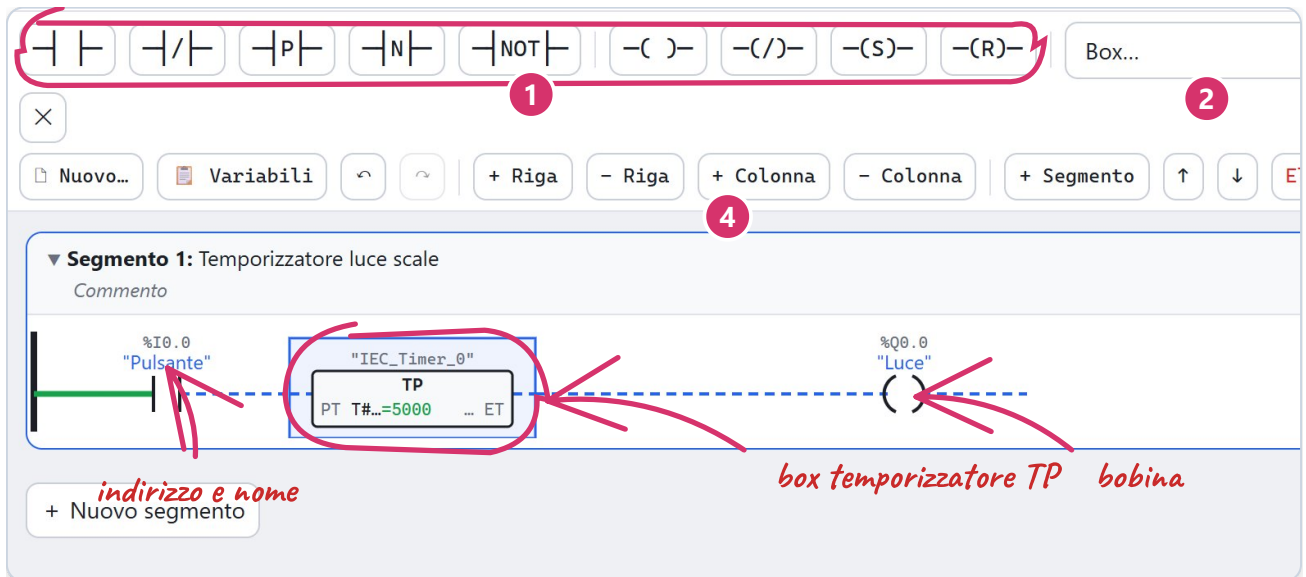


L'autoritenuta: il circuito da conoscere

Il pulsante di **Marcia** accende il **Motore**; un contatto dello stesso **Motore**, in parallelo alla Marcia, lo "tiene" acceso quando si rilascia il pulsante. Il contatto NC di **Arresto** in serie interrompe tutto. È l'esempio "Marcia / arresto (LADDER)".



L'editor LADDER



L'esempio "Luce delle scale (LADDER)" con il box TP selezionato.

1 Elementi: contatti NA, NC, fronti P e N, NOT, bobine normale, negata, S e R.

2 Box...: temporizzatori, contatori, confronti, calcoli, display, moduli.

Più a destra: **Apri ramo** e **Chiudi ramo** per i collegamenti in parallelo.

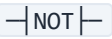
4 Righe, colonne e segmenti: aggiungi, toglì, sposta.

A destra, **controllo online**: colori dal vivo (verde pieno = passa corrente; blu tratteggiato = non passa).

- 1** Clicca una **cella** del segmento, poi l'elemento nella barra (o trascinalo dall'albero "Istruzioni" a destra).
- 2** Scrivi l'**operando**: il nome (es. *Pulsante*), l'indirizzo (`%I0.0`) o un numero. Si può cliccare direttamente `<??.?>` sul disegno. Un nome nuovo diventa un merker.
- 3** Per un **parallelo**: vai sulla riga sotto, **Apri ramo**, inserisci i contatti e chiudi con **Chiudi ramo** nel punto in cui il ramo si ricongiunge.
- 4** Dai un titolo al segmento e un commento: aiuta a leggere il programma.

*Anche il LADDER diventa blocchi dentro la scheda: il limite è di **96 blocchi interni**, indicato nel pannello ("5/96 blocchi interni").*

Elementi e box

Elemento	Cosa fa
	Fronte di salita / discesa: passa corrente per un solo ciclo quando l'operando cambia.
	Inverte la corrente che arriva da sinistra.
	Bobina negata: l'operando è il contrario della corrente.
 / 	Imposta a 1 (e resta 1) / riporta a 0. Coppia tipica per memorizzare.
TON · TOF · TP	Ritardo all'inserzione, alla disinserzione, impulso. PT = tempo (es.  , ET = tempo trascorso (facoltativo). Nomi automatici "IEC_Timer_0"...
CTU · CTD · CTUD	Contatori avanti, indietro, avanti/indietro, con PV (valore da raggiungere) e CV (valore attuale).
CMP	Contatto di confronto: ==, <>, >, >=, <, <= fra due valori.
MOVE · ADD · SUB · MUL · DIV · MOD · SCALE	Copiano o calcolano valori quando arriva corrente (divisione intera).
SR · RS	Flip-flop con reset prevalente / set prevalente.
OLED · LCD · IMG · PAGE · MTX · MTXT	Scritte, immagini e pagine sui display e sulla matrice (con corrente).
SCL	Script ST eseguito solo quando arriva corrente.

Operandi

Si può scrivere	Esempio
nome della tabella variabili o etichetta del pin	 , 
indirizzo	 ,  ,  ,  , 
costante o tempo	 ,  ,  , 
pin diretto	 , 

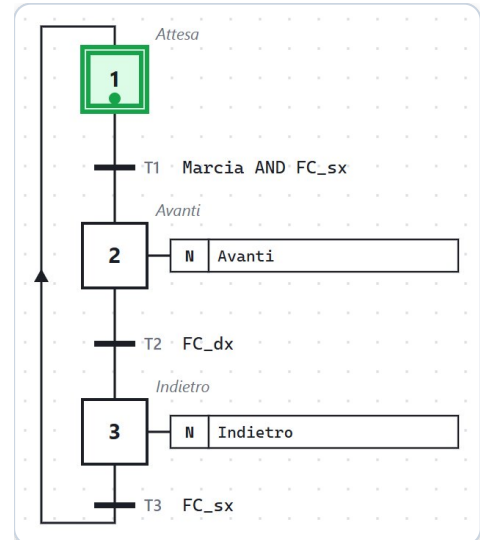
Le uscite vengono scritte sui pin alla **fine** di ogni ciclo, come nei PLC veri: se due segmenti comandano la stessa bobina, vince l'ultimo (la pagina lo segnala).

10 Programmare in GRAFCET

Per le macchine che lavorano a fasi: una tappa dopo l'altra, si avanza quando succede qualcosa.

La logica

- Il grafico si legge **dall'alto verso il basso**: tappa, transizione, tappa...
- Una **tappa** (quadrato numerato) è una fase del lavoro; quando è **attiva** esegue le sue **azioni** (accendere un motore, scrivere un messaggio).
- La **tappa iniziale** ha il doppio quadrato: è attiva all'accensione.
- Fra due tappe c'è una **transizione** (trattino orizzontale) con la sua **recettività**: la condizione per andare avanti (es. `FC_dx`, il finecorsa di destra).
- Una transizione **scatta** quando la tappa sopra è attiva e la recettività è vera: la tappa sopra si spegne, quella sotto si accende.
- L'ultima tappa di solito **torna alla prima** con una linea di rinvio.



Carrello avanti e indietro.

L'editor GRAFCET

GRAFCET
Transizione T1

Numero: 1

Recettività scritta come: Espressione

Recettività: Marcia AND FC_sx

COLLEGAMENTI

	a monte	a valle
Tappa 1 - Attesa	☒	☐
Tappa 2 - Avanti	☐	☒

A sinistra il grafico; a destra le proprietà della transizione selezionata.

- 1 Con **SFC / LADDER** in alto si passa dal grafico ai segmenti LADDER del progetto.
- 2 **+ Tappa** (o doppio clic sul foglio) aggiunge una tappa. Passando sopra una tappa compare un **pallino blu**: trascinalo su un'altra tappa per collegarle (lasciato sul foglio vuoto crea una tappa nuova già collegata).
- 3 Sulla nuova linea c'è un **+**: cliccalo per mettere la transizione e scrivere la recettività (doppio clic su una transizione per cambiarla).
- 4 Seleziona una tappa per aggiungere le **azioni** nel pannello a destra. I pulsanti **↓ Tappa successiva** e **↶ Torna all'iniziale** costruiscono la sequenza più in fretta.
- 5 Con il programma in esecuzione e **controllo online** attivo, la tappa attiva diventa **verde** con il "gettone" e le recettività vere si colorano.

Scrivere la recettività

Scrivi	Significa
Marcia AND FC_sx (anche · o *)	entrambe vere
A OR B (anche +)	almeno una vera
NOT FC1 oppure /FC1	FC1 falso
RE(Pulsante) / FE(Pulsante)	fronte di salita / discesa del pulsante
X3	la tappa 3 è attiva
X2.T > T#5S	la tappa 2 è attiva da più di 5 secondi (tappa a tempo)
Livello >= 500	confronto fra una variabile e un numero (= <> < <= > >=)
1 oppure TRUE	sempre vera (si passa subito)
parentesi	(A OR B) AND NOT C

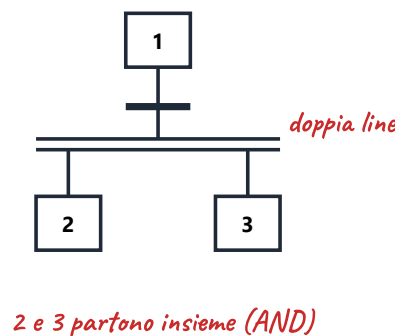
In alternativa, in **Recettività scritta come** si può scegliere **Segmento LADDER**: la condizione è vera quando la corrente arriva in fondo a destra del segmento.

Le azioni delle tappe

Tipo	Cosa fa	Esempio
N	uscita accesa finché la tappa è attiva (la più usata)	N Avanti
S / R	accende e lascia acceso / spegne	S Allarme ... R Allarme
D	accesa dopo un ritardo dall'attivazione della tappa	D Sirena T#2S
L	accesa solo per un tempo limitato	L Cicalino T#1S
V	scrive un numero in una variabile finché la tappa è attiva	Angolo := 45
P1	un calcolo eseguito una volta, all'attivazione	Pezzi := Pezzi + 1
MSG	messaggio su OLED o LCD	"Carrello in avanti"

Scelte e rami simultanei

- **Scelta (OR)**: più transizioni che partono dalla stessa tappa. Si prende il ramo la cui recettività diventa vera: conviene scriverle in modo che non possano essere vere insieme.
- **Rami simultanei (AND)**: una transizione che attiva più tappe insieme (doppia linea). Si riuniscono con una transizione che aspetta che tutte le tappe siano finite.
- Si impostano con il tasto destro sulla transizione → **Divergenza / Convergenza**, oppure con i pulsanti **OR scelta / AND simultanea** nel pannello.



Sicurezze ed emergenza

Nel pannello del grafico (senza nulla selezionato) c'è la **Condizione di reinizializzazione**: quando è vera (es. Emergenza) tutte le tappe si spengono e si torna alla situazione iniziale. Nella vista **LADDER** del progetto si possono aggiungere **segmenti di sicurezza** che leggono lo stato delle tappe (X1 , X2 ...) e, per esempio, resettano le uscite con l'emergenza: così fa l'esempio "Carrello avanti e indietro (GRAFCET)".

Il GRAFCET è ideale per insegnare a scomporre un problema: prima si disegnano le fasi alla lavagna, poi si programmano. Funziona benissimo in coppia: uno disegna le tappe, l'altro scrive le recettività.

11 Script ST e funzioni C

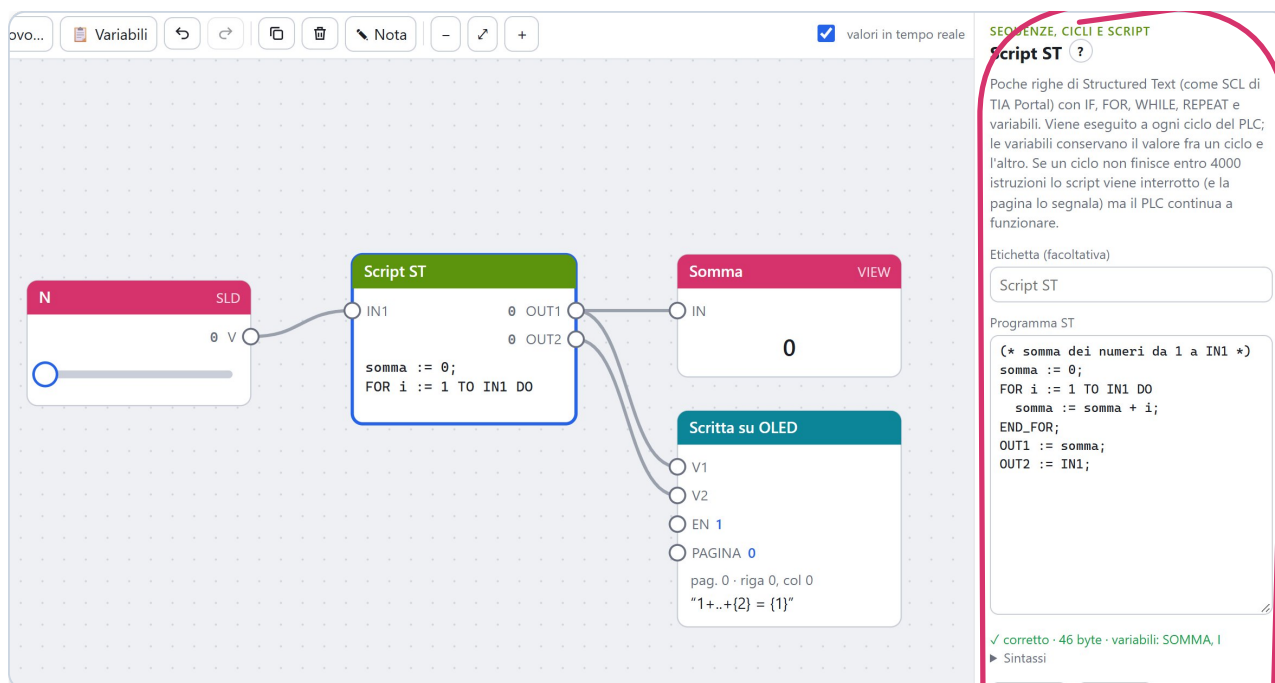
Per calcoli, cicli e scelte fra molti casi: un piccolo programma a testo dentro un blocco.

Gli script sono un'estensione facoltativa, utile nelle classi che studiano già la programmazione. Il blocco **Script ST** (testo strutturato, il linguaggio testuale dei PLC) e il blocco **Script C** (una funzione nello stile di Arduino) si usano nei progetti a blocchi; in LADDER c'è il box **SCL**.

Più spazio per scrivere

Accanto al codice c'è il pulsante  **Ingrandisci**: apre il codice in un riquadro grande, con i **numeri di riga**, il controllo degli **errori mentre scrivi** (con il numero di riga) e la **guida del linguaggio** a lato. Si chiude con **Chiudi** o con Esc; quello che scrivi è già nel blocco. Vale anche per il box SCL del LADDER.

Script ST



Esempio "Somma con ciclo FOR (script)": il codice si scrive nel pannello a destra.

- **Ingressi** IN1 ... IN4 (solo lettura), **uscite** OUT1 , OUT2 : il blocco mostra solo quelli usati.
- Le altre variabili nascono assegnandole (`x := 5;`) e **ricordano il valore** da un ciclo all'altro (massimo 16 per script).
- Istruzioni: `IF ... THEN ... ELIF ... ELSE ... END_IF;` , `FOR ... TO ... DO ... END_FOR;` , `WHILE` , `REPEAT ... UNTIL` , `EXIT` .
- Operatori: `+` `-` `*` `/` `MOD` , confronti, `AND` `OR` `XOR` `NOT` . Funzioni: `ABS` `MIN` `MAX` `LIMIT` `MILLIS` .
- Commenti `(* ... *)` o `//` . Solo numeri interi; niente `CASE` .

```
(* somma da 1 a N *)
somma := 0;
FOR i := 1 TO IN1 DO
    somma := somma + i;
END_FOR;
OUT1 := somma;
OUT2 := IN1;
```

Funzioni C

La screenshot mostra l'interfaccia di un IDE per Arduino. In alto a sinistra c'è una barra degli strumenti con icone per variabili, undo, redo, copia, incolla, nota, zoom in, zoom out e reset. In alto a destra c'è un checkbox "valori in tempo reale" che è attivo. Al centro dello workspace c'è un blocco funzione "media()" di tipo "C". A sinistra del blocco ci sono due blocchi "Valore A" e "Valore B" di tipo "SLD", entrambi con un slider. A destra del blocco "media()" ci sono tre blocchi: "Media" di tipo "VIEW" con un display a 7 segmenti che mostra "0", "Uscita digitale" di tipo "C" con un display a 7 segmenti che mostra "0", e "Allarmi" di tipo "VIEW" con un display a 7 segmenti che mostra "0". Sotto il blocco "media()" c'è un tooltip giallo che dice: "Clicca sul blocco per vedere il codice: la funzione è nella libreria del progetto (colonna a sinistra, Funzioni C)". A destra dello workspace c'è un pannello "SEQUENZE, CICLI E SCRIPT" con un tab "Script C" selezionato. Sotto il tab c'è un campo "Etichetta (facoltativa)" con il valore "Script C". Sotto quello c'è un menu a tendina "Funzione della libreria del progetto" con "media()" selezionato. In basso c'è un riquadro "Codice C della funzione" che mostra il seguente codice:

```
// Media di due valori, allarme sopra
// e conteggio degli allarmi.
int media(int a, int b, bool &allarme)
static bool prima = false; // stato
int m = (a + b) / 2; // media
allarme = m > 500;
if (allarme && !prima) {
    quanti++; // conteggio
}
prima = allarme;
```

Esempio "Media e allarme (funzione C)".

- Si scrive **una funzione**: i parametri normali sono gli **ingressi** (fino a 8), il valore restituito e i parametri con `&` sono le **uscite** (fino a 4).
- Le variabili normali ripartono a ogni ciclo; quelle `static` ricordano il valore.
- Si possono usare `if/else` , `switch` , `for` , `while` , e funzioni come `min` `max` `abs` `constrain` `map` `millis` .
- **Non** si possono usare `digitalRead` , `delay` , `Serial` , numeri con la virgola, array o stringhe: ingressi e uscite passano sempre dai fili del blocco. La pagina spiega l'errore se li si scrive.
- Le funzioni del progetto compaiono nella colonna "Funzioni C del progetto" e si possono riusare in più blocchi.

Niente cicli infiniti

Uno script che supera 4000 istruzioni in un ciclo viene interrotto (il resto del PLC continua a funzionare) e la barra di stato mostra "ciclo infinito?". Il PLC non deve mai "aspettare" dentro uno script: per i tempi si usano i temporizzatori.

12 Display, matrice LED e immagini

Far "parlare" l'impianto: messaggi, valori, icone e animazioni.

I display che si possono collegare

Display	Collegamento	Note
OLED 128×64 (0,96" o 1,3")	4 fili: GND, 5V, SDA, SCL (Mega: pin 20 e 21; UNO R4: A4 e A5)	21 caratteri × 8 righe. Controller SSD1306 o SH1106 (se vedi pixel a caso, scegli l'altro nelle Impostazioni).
LCD 16×2 / 20×4 con modulo I ² C	come l'OLED (indirizzo di solito 0x27 o 0x3F)	Si imposta in Impostazioni → Display LCD .
LCD collegato in parallelo	6 fili sui pin scelti nelle Impostazioni	Quei pin diventano riservati.
Matrice LED 12×8	già sulla UNO R4 WiFi	Numeri brevi, scritte che scorrono, icone.

I display sono facoltativi e si possono collegare anche a scheda accesa: la scheda li cerca ogni 3 secondi. Se nessun blocco scrive sul display, la scheda mostra da sola nome, indirizzo e stato: comodo per sapere l'indirizzo senza PC. In **Impostazioni** il pulsante **Cerca dispositivi I²C** dice cosa è collegato.

Scrivere testi e valori

I blocchi **Scritta su OLED** e **Scritta su LCD** (in LADDER i box OLED e LCD, in GRAFCET l'azione MSG) scrivono un messaggio di massimo 40 caratteri in una riga e colonna, quando EN vale 1. Nel testo i **segnaposto** vengono sostituiti dai valori dei fili V1 e V2:

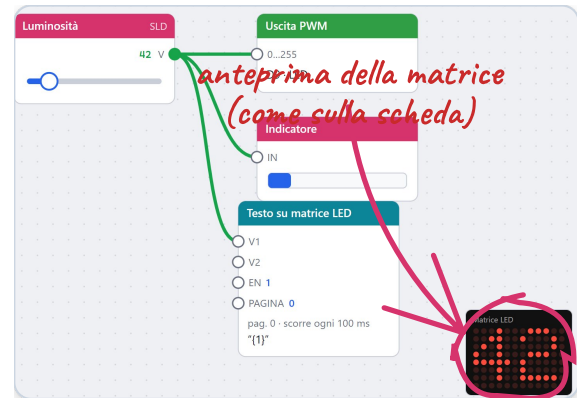
Segnaposto	Risultato	Esempio (V1 = 235)
{1} {2}	il valore di V1 o V2	"Pezzi: {1}" → Pezzi: 235
{1.1} {1.2}	con uno o due decimali	"T: {1.1} C" → T: 23.5 C
{1:3}	almeno 3 cifre, con zeri davanti	V1 = 7 → 007
{1:3.1}	le due cose insieme	V1 = 57 → 05.7

L'effetto può essere **Lampeggia** o **Scorre** (se il testo è più lungo della riga). Con il blocco **Pagine display** si preparano fino a 16 schermate diverse e si sfogliano con dei pulsanti: ogni scritta ha l'ingresso PAGINA che dice in quale schermata compare.

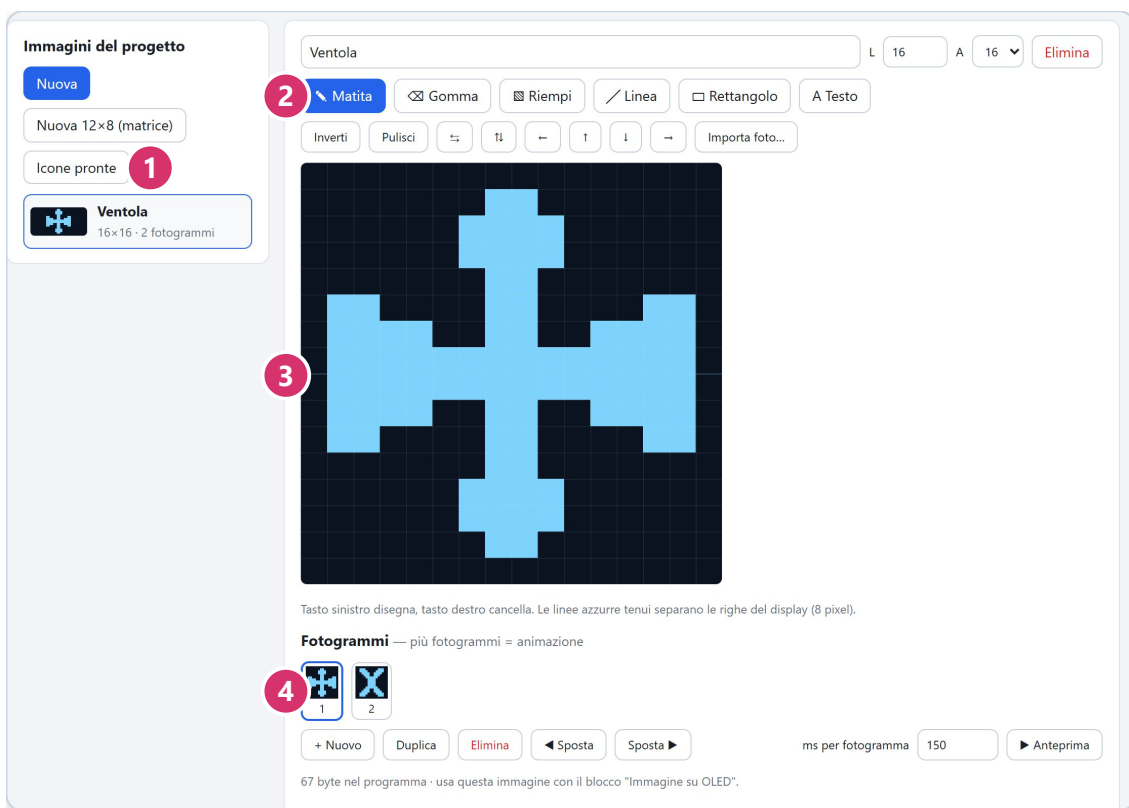
La matrice LED della UNO R4 WiFi

Con il blocco **Testo su matrice LED** il testo resta fermo se entra (circa 2 caratteri, per esempio "42"), altrimenti **scorre**. Il blocco **Immagine su matrice LED** mostra un'immagine 12×8, anche animata. La pagina mostra in basso a destra un'**anteprima** identica a quella della scheda.

Quando nessun programma usa la matrice, la scheda la usa per sé: un puntino che si muove = RUN, un quadrato = STOP, un trattino = nessun programma, "!" = errore nel programma, e l'indirizzo che scorre quando la rete cambia.



Disegnare immagini e animazioni



La scheda "Immagini" con l'icona pronta "Ventola (animata)".

- 1 **Nuova**, **Nuova 12×8 (matrice)**, **Icone pronte** (lampadina, ventola, termometro, allarme, frecce, motore...).
- 2 Strumenti: matita, gomma, riempi, linea, rettangolo, testo, inverti, specchia, sposta, **Importa foto...**

- 3 Il disegno: tasto sinistro disegna, tasto destro cancella.
- 4 **Fotogrammi**: con più fotogrammi l'immagine diventa un'animazione (con la velocità in ms).

Le immagini fanno parte del progetto e occupano spazio nel programma (la pagina dice quanti byte). Si mostrano con i blocchi **Immagine su OLED** e **Immagine su matrice LED**.

13 Moduli: sensori e attuatori

Blocchi pronti per i componenti più usati nei laboratori, senza scrivere codice.

I blocchi dei moduli sono nella categoria **Moduli (sensori e attuatori)** della colonna di sinistra (in LADDER, nel menu **Box...**). Scegliendo i pin nel blocco, la pagina li riserva al modulo.

Modulo	Cosa dà / cosa fa	Collegamento e note
Suono (tone)	Fa suonare un cicalino passivo alla frequenza FREQ (Hz) finché EN = 1.	Cicalino passivo o piccolo altoparlante con resistenza da 100 Ω. Sulla Mega mentre suona non funziona il PWM sui pin 9 e 10.
DHT11 / DHT22	Temperatura e umidità in decimi (235 = 23,5 °C), aggiornate ogni 2 s.	Un pin dati; resistenza da 10 kΩ verso 5 V (i moduli su basetta ce l'hanno già).
Sonda DS18B20	Temperatura in decimi, circa una volta al secondo; uscita OK se la sonda risponde.	Un pin dati con resistenza da 4,7 kΩ fra dati e 5 V. Una sonda per pin.
Distanza HC-SR04	Distanza in millimetri (fino a MAX cm), circa 16 volte al secondo.	Pin TRIG e ECHO.
Encoder rotativo	Posizione (fra MIN e MAX) e verso di rotazione della manopola.	Pin A: Mega 2, 3, 18 o 19; UNO R4 solo 2 o 3 . Massimo 2 encoder. Il pulsante della manopola è un ingresso normale.
Servomotore	Porta il servo all'ANGOLO (0–180°), con velocità VEL facoltativa.	Un pin di segnale. Massimo 4 servo. Sulla Mega disattiva il PWM sui pin 44–46. Più servo: alimentazione esterna a 5 V.
Motore passo-passo	Velocità o posizione (TARGET in passi; 2048 = un giro del 28BYJ-48).	28BYJ-48 con scheda ULN2003 (4 pin) oppure driver STEP/DIR. Velocità entro qualche centinaio di passi/s.
Striscia LED RGB (WS2812) + Colore LED	Fino a 60 LED colorati: si sceglie il colore di ogni LED (o di tutti) e la luminosità.	Pin DIN; alimentazione a 5 V, separata se i LED sono molti.
Orologio RTC DS3231	Ore, minuti, giorno... anche come HHMM (es. 1030) per gli orari.	Sugli stessi fili del display (SDA/SCL). Si regola in Impostazioni → Orologio con l'ora del PC.

Alcuni moduli rallentano un po' il ciclo

Il DHT ferma il ciclo per circa 25 ms ogni 2 s, l'HC-SR04 per qualche millisecondo per metro misurato. Per un impianto didattico è irrilevante, ma spiega perché la "durata massima del ciclo" nella barra di stato sale.

Tutte le misure sono numeri interi: temperature e umidità in decimi, distanze in mm. Per confrontarle si usano direttamente questi numeri (28,0 °C → 280).

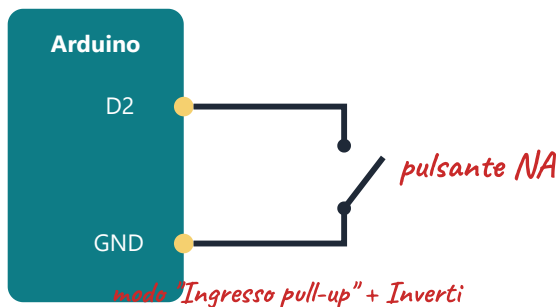
14 Collegamenti elettrici

Schemi semplici per il banco di lavoro, e le regole per non rovinare le schede (e nessuno si faccia male).

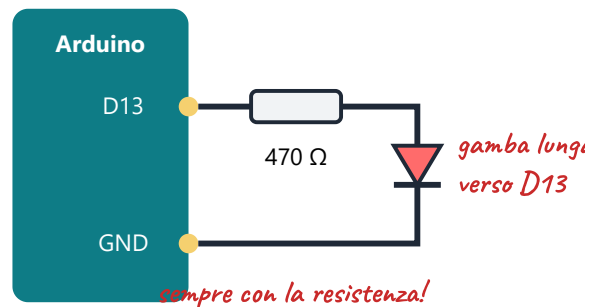
Regole di sicurezza

- Le schede lavorano a **5 volt**. **Mai** collegare direttamente la tensione di rete (230 V) né far cablare agli studenti circuiti a 230 V: per comandare carichi di potenza si usano **moduli relè** con contatti isolati, cablati e controllati dal docente, in bassa tensione quando possibile (12/24 V).
- Un pin dà pochissima corrente: **circa 20 mA sulla Mega, circa 8 mA sulla UNO R4 WiFi**. Direttamente si accende solo un LED con la sua resistenza: motori, lampade, elettrovalvole passano da un relè o da un transistor.
- Motori, servo e strisce LED vanno alimentati a parte; il **GND** dell'alimentatore va unito a quello della scheda.
- Cablare sempre a scheda **scollegata**; ricontrollare prima di alimentare.
- I sensori industriali a 24 V non si collegano direttamente: serve un modulo di interfaccia (optoisolatore).

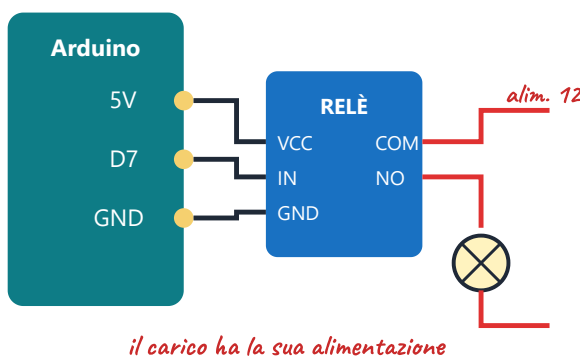
① Pulsante o finecorsa



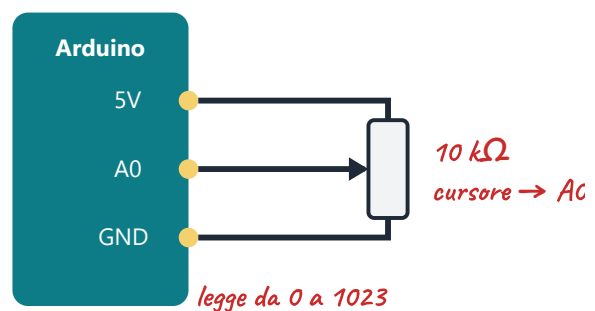
② LED



③ Modulo relè (carichi più grandi)



④ Potenzziometro (analogico)

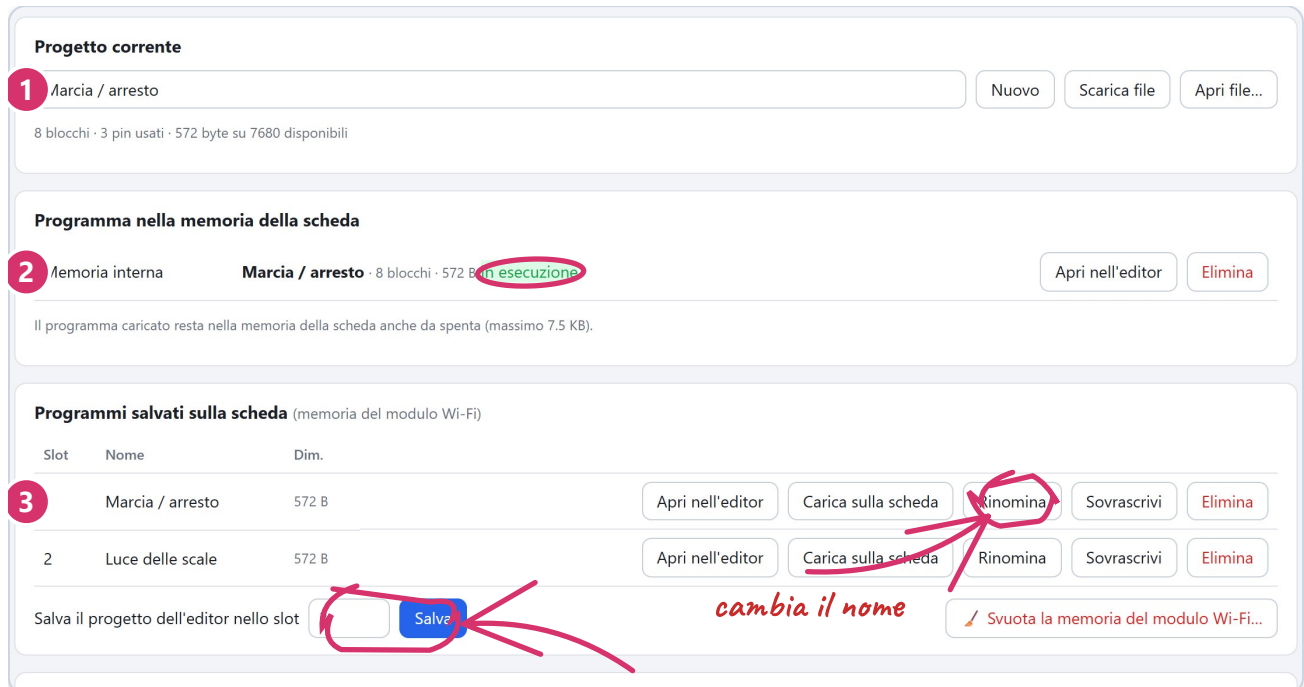


Valori utili

Componente	Collegamento consigliato
LED	resistenza da $470\ \Omega$ – $1\ k\Omega$ in serie (va bene per entrambe le schede)
Pulsante, interruttore, finecorsa	fra pin e GND, modo "Ingresso pull-up" con Inverti; gli arresti con contatto NC
Cicalino attivo (suona da solo)	come un LED, su un'uscita digitale (se assorbe poco) o tramite transistor/relè
Cicalino passivo	blocco "Suono", resistenza da $100\ \Omega$
Motore in corrente continua	mai diretto: relè o driver per motori, alimentazione separata
Alimentazione della scheda	cavo USB, alimentatore sul connettore della scheda, oppure PoE (solo Mega con la shield)

15 Salvare e organizzare i programmi

Il programma in esecuzione, i programmi salvati sulla scheda, i file sul PC e gli esempi.



La scheda "Progetti".

- 1 Progetto corrente:** il nome (si cambia anche cliccando il titolo nell'editor), **Nuovo**, **Scarica file** (salva un file .p1cb sul PC), **Apri file...**. Sotto: blocchi, pin usati e spazio occupato ("572 byte su 7680 disponibili").
- 2 Programma nella memoria della scheda:** quello che la scheda esegue, anche dopo lo spegnimento. **Apri nell'editor** lo riporta nell'editor; **Elimina** lo cancella (la scheda si ferma).
- 3 Programmi salvati sulla scheda:** fino a **99 posti** (slot). Per ogni slot: Apri nell'editor, Carica sulla scheda, **Rinomina**, Sovrascrivi, Elimina. In fondo si salva il progetto dell'editor nello slot scelto.

- Salvare in uno slot, rinominarlo ed eliminarlo richiede il PIN del docente; aprire uno slot nell'editor no.
- **Svuota...** cancella definitivamente tutti gli slot e la bozza condivisa.
- Sulla Mega gli slot sono sulla microSD: senza microSD non ci sono slot. Si può anche fare una **copia di riserva** del programma nella memoria interna (EEPROM), usata se la microSD manca.
- I file .p1cb si aprono su qualsiasi scheda ArduLearn. Un progetto scritto per la Mega aperto sulla UNO R4 viene adattato ai pin disponibili (la pagina avvisa di controllarli).

Un'idea: assegna a ogni classe un gruppo di slot (1-20 la 3ªA, 21-40 la 4ªB...). Oppure fai scaricare a ogni studente il proprio file: è il modo più sicuro di conservare i lavori.

Gli esempi

In fondo alla scheda Progetti ci sono esempi pronti in tutti i linguaggi: **Primo programma**, **Marcia / arresto**, **Luce delle scale**, **Termostato**, **Contapezzi**, **Dimmer PWM**, **Semaforo a passi**, **Somma con ciclo FOR** (ST), **Media e allarme** (C), **Carrello avanti e indietro** e **Semaforo con tempi di tappa** (GRAFCET), e le versioni LADDER di marcia/arresto, luce delle scale, lampeggio con due timer, contapezzi con display. Si aprono con [Apri nell'editor](#) ; poi basta "Carica sulla scheda". Molti funzionano anche senza collegare nulla, con i pulsanti virtuali.

16 Aggiornamenti

ArduLearn migliora nel tempo: ecco come portare le schede all'ultima versione.

UNO R4 WiFi: dalla pagina

Quando esce una nuova versione, nella barra in alto compare la pastiglia lampeggiante **↑ Aggiornamento disponibile**. È il PC a controllare su Internet (ogni 6 ore): la scheda non ha bisogno di uscire in Internet.

- 1 Clicca la pastiglia (o **Impostazioni** → **Aggiornamento del firmware**).
- 2 Leggi le **Novità**: l'elenco dei miglioramenti della nuova versione.
- 3 Premi **Aggiorna** e inserisci il PIN. Non spegnere la scheda e non chiudere la pagina: il programma si ferma per qualche secondo (fino a un minuto) e riparte da solo.

Programmi salvati, rete Wi-Fi e impostazioni restano. Alla fine la pagina si ricarica e mostra le versioni installate.

Aggiornare pagina web, firmware del PLC (RA4M1)?

Novità

- Il nome del progetto si cambia anche dall'editor: basta cliccare il titolo nel pannello a destra.
- I programmi salvati sulla scheda si possono rinominare con il pulsante "Rinomina" nella scheda Progetti.

Il programma si ferma per qualche secondo (fino a un minuto) e poi riparte da solo. Non spegnere la scheda durante l'aggiornamento.

AnnullaAggiorna

Senza Internet sul PC l'aggiornamento risulta "non verificabile". Si può anche aggiornare da file (**Aggiorna da file...**) oppure con l'app via USB.

Mega 2560: con l'app

La Mega si aggiorna con l'app: scarica l'ultima versione di **ArduLearn.exe** da GitHub, collega la scheda via USB e usa **Carica il firmware**; poi **Trova le schede** → **Prepara la microSD / Aggiorna la pagina** per la pagina. Anche la UNO R4 WiFi si può aggiornare così. I programmi salvati restano.

17 Esercizi guidati

Sette esercitazioni, dalla più semplice alla più complessa. Ognuna ha la soluzione pronta negli Esempi della pagina.

Ogni esercizio indica l'obiettivo, cosa collegare (tutti si possono fare anche solo con i comandi virtuali), la soluzione commentata e qualche variante per differenziare. Le soluzioni si aprono da **Progetti** → **Esempi**: si può far lavorare la classe da zero e mostrare la soluzione alla fine, oppure partire dall'esempio e chiedere le varianti.

Esercizio 1 · Il primo lampeggio

OBIETTIVO

Prendere confidenza con la pagina: blocchi, fili, caricamento, comandi virtuali.

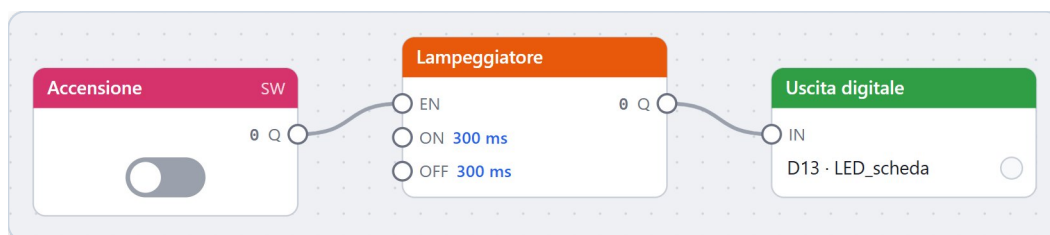
CONSEGNA

Un interruttore nella pagina fa lampeggiare il LED della scheda (D13): 0,3 s acceso e 0,3 s spento.

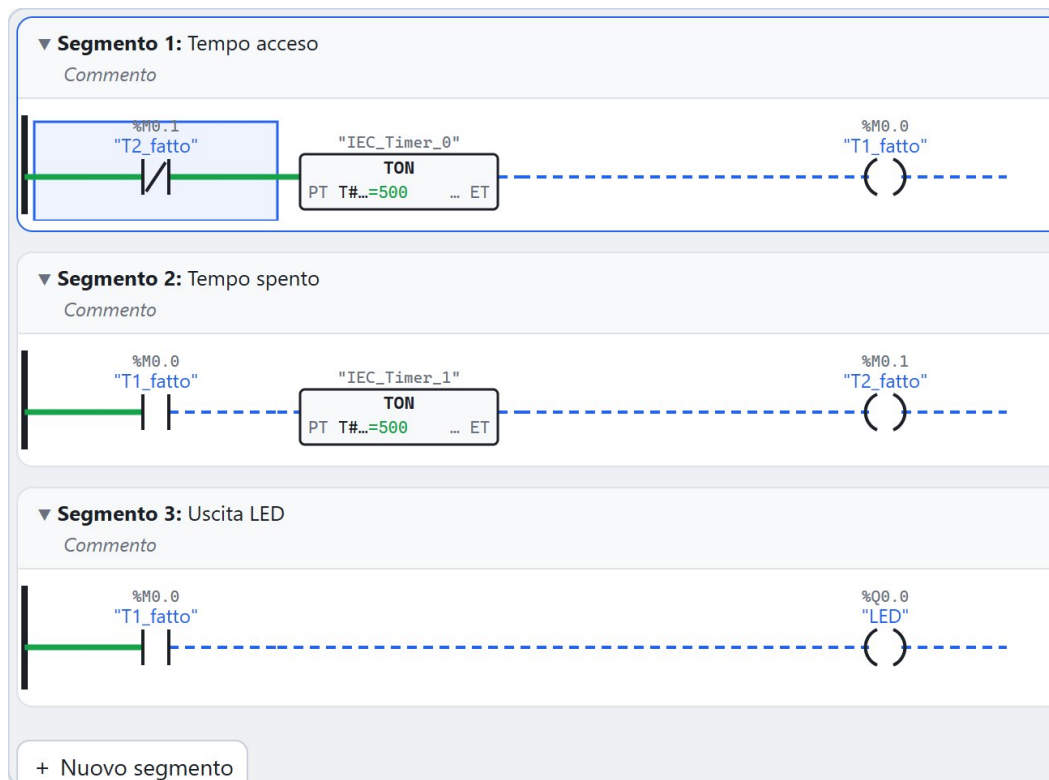
MATERIALE

Nessuno: il LED "L" è già sulla scheda.

SOLUZIONE FBD · esempio "Primo programma"



Interruttore virtuale → **Lampeggiatore** (EN) con ON = OFF = 300 ms → Uscita digitale D13. Quando l'interruttore è spento, EN vale 0 e il LED resta spento.



Due TON da 0,5 s si resettano a vicenda: quando il primo finisce accende l'uscita e avvia il secondo, che a sua volta resetta il primo. È lo schema classico dei PLC senza blocco lampeggiatore.

VARIANTI

- Cambiare i tempi con un **Cursore virtuale** collegato a ON.
- Aggiungere un secondo LED che lampeggia in opposizione (NOT).

Esercizio 2 · Marcia e arresto di un motore

OBIETTIVO

Capire la memoria (autoritenuta) e la priorità dell'arresto.

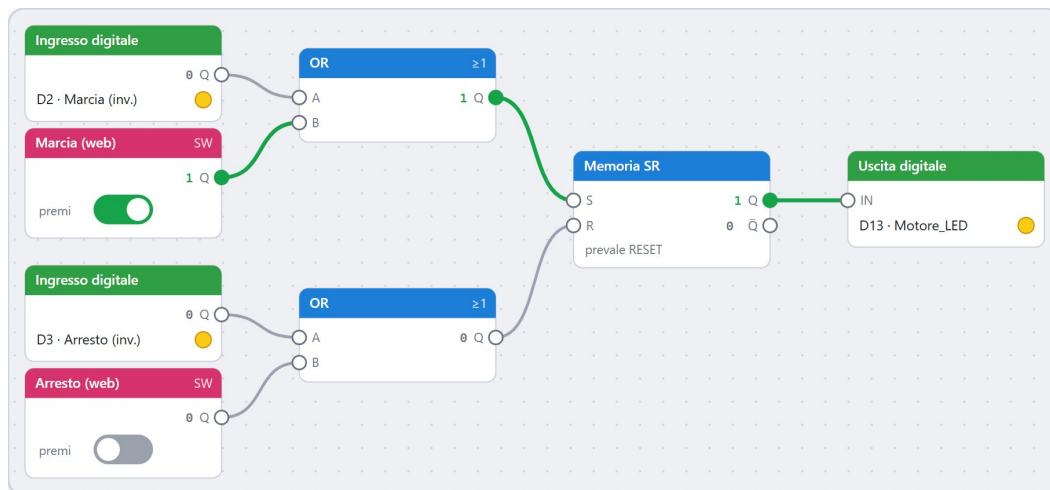
CONSEGNA

Il pulsante MARCIA accende il motore, che resta acceso anche rilasciando il pulsante; il pulsante ARRESTO lo spegne. Se si premono insieme, deve vincere l'arresto.

MATERIALE

Due pulsanti su D2 (Marcia) e D3 (Arresto) verso GND; un LED su D13 al posto del motore. Oppure solo pulsanti virtuali.

SOLUZIONE FBD · esempio "Marcia / arresto"



Il pulsante vero e quello virtuale entrano in un OR (così funzionano entrambi) e vanno in S e R di una **Memoria SR** con "prevale RESET". L'uscita Q comanda il motore.

SOLUZIONE LADDER · esempio "Marcia / arresto (LADDER)"



Contatto Marcia in parallelo al contatto del Motore (autoritenuta), in serie al contatto NC di Arresto, e la bobina Motore.

DOMANDE PER LA CLASSE

- Cosa succede se il filo dell'arresto si rompe? Come si collega un arresto "sicuro"? (contatto NC, capitolo 7)
- Rifare la versione LADDER con le bobine S e R.

Esercizio 3 · La luce delle scale

OBIETTIVO

Usare un temporizzatore a impulso.

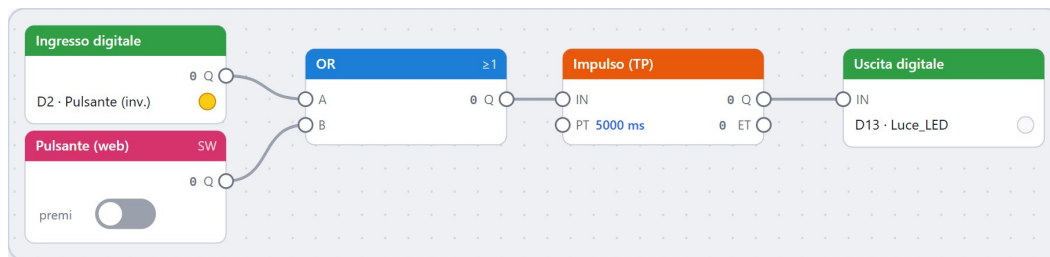
CONSEGNA

Premendo il pulsante la luce si accende e resta accesa 5 secondi, poi si spegne da sola.

MATERIALE

Un pulsante su D2, un LED su D13 (o pulsante virtuale).

SOLUZIONE FBD · esempio "Luce delle scale"



Pulsante (vero o virtuale) → **Impulso (TP)** con PT = 5000 ms → luce. Durante l'impulso altre pressioni non allungano il tempo.

SOLUZIONE LADDER · esempio "Luce delle scale (LADDER)"



VARIANTI

- Con un **TOF** invece del TP: che differenza c'è se si tiene premuto il pulsante?
- Due pulsanti (piano terra e primo piano) in OR.
- Far lampeggiare la luce negli ultimi 2 secondi (confronto sul tempo ET).

Esercizio 4 · Il contapezzi

OBIETTIVO

Contatori, fronti e display.

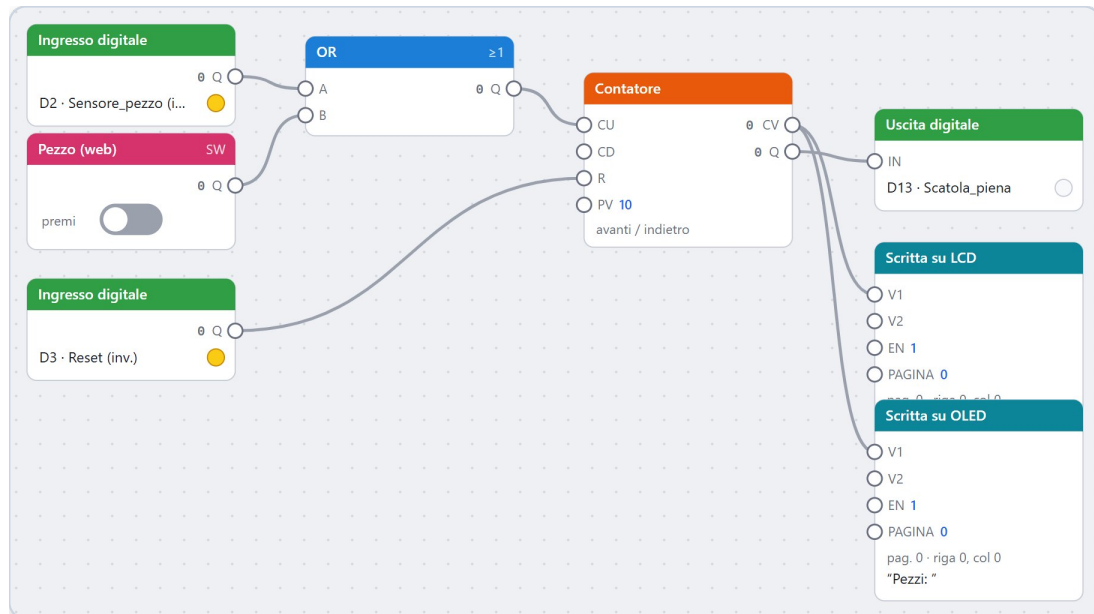
CONSEGNA

Un sensore conta i pezzi che passano; a 10 pezzi si accende la spia "scatola piena" e sul display compare il conteggio. Un pulsante azzerà.

MATERIALE

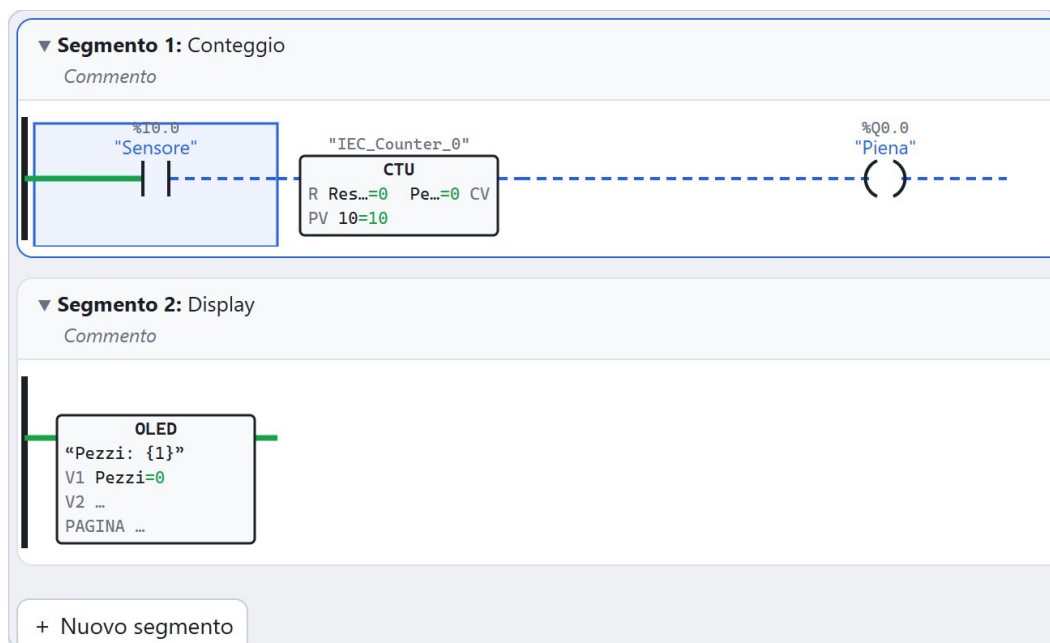
Pulsante o sensore su un ingresso, pulsante di reset, LED; facoltativi display OLED o LCD.

SOLUZIONE FBD · esempio "Contapezzi"



Contatore con PV = 10: CU dal sensore, R dal reset, Q alla spia, CV al display ("Pezzi: {1}").

SOLUZIONE LADDER · esempio "Contapezzi con display (LADDER)"



VARIANTI

- Mostrare anche le scatole completate (secondo contatore).
- Scrivere "{1:3}" per vedere 007 invece di 7.

Esercizio 5 · Il semaforo

OBIETTIVO

Le sequenze: prima con i blocchi Passo, poi con il GRAFCET.

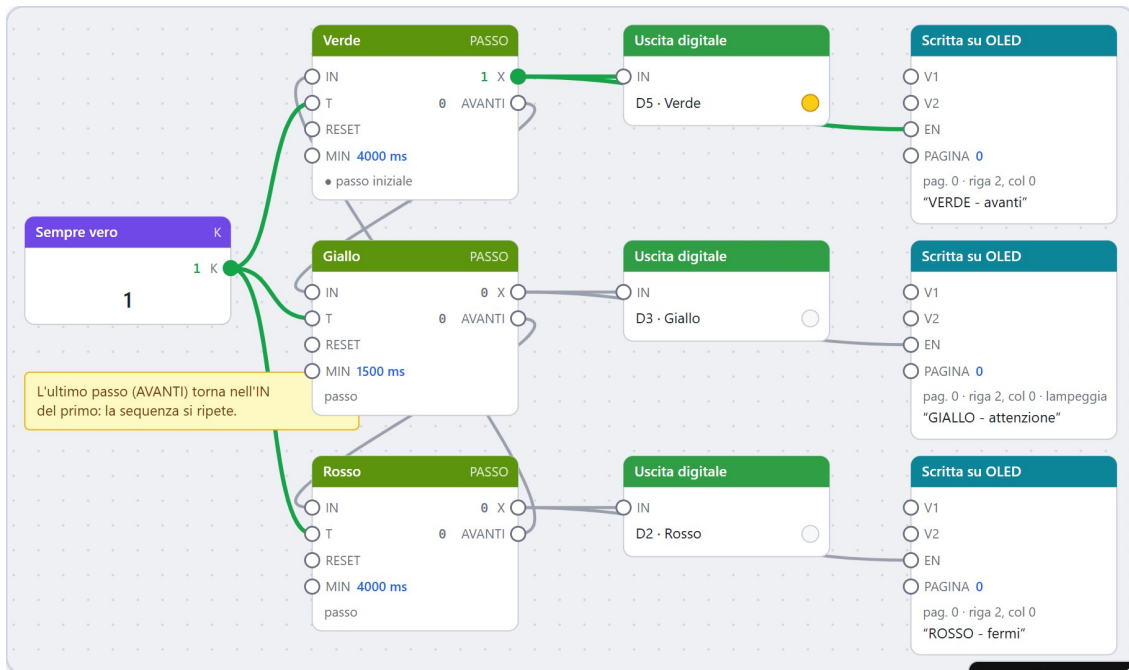
CONSEGNA

Verde 4 s, giallo 1,5 s, rosso 4 s, e poi di nuovo verde, all'infinito.

MATERIALE

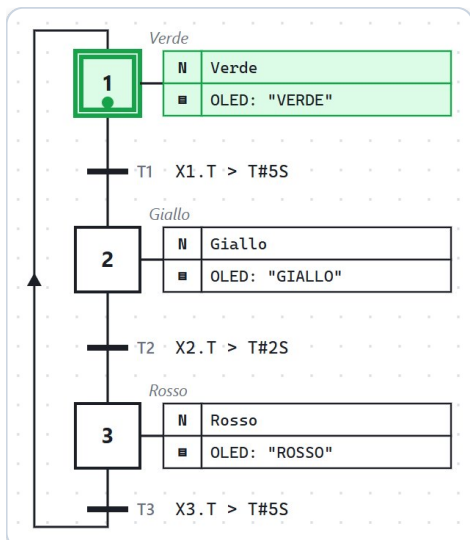
Tre LED (verde, giallo, rosso) su tre uscite, con resistenze; facoltativo il display.

SOLUZIONE FBD · esempio "Semaforo a passi"



Tre blocchi **Passo** in catena: il verde è il passo iniziale; T = 1 (costante "Sempre vero") e MIN = durata della luce. L'AVANTI dell'ultimo torna all'IN del primo.

SOLUZIONE GRAFCET · esempio "Semaforo con tempi di tappa"



Tre tappe con azione **N** sulle lampade (e MSG sul display). Le transizioni usano il tempo di tappa: X1.T > T#5S (verde), X2.T > T#2S (giallo), X3.T > T#5S (rosso, e si torna alla tappa 1). Nell'esempio i tempi sono un po' diversi da quelli della consegna: un buon primo esercizio è correggerli.

Confronta le due soluzioni: quale si legge meglio?

VARIANTI

- Pulsante pedonale: rosso anticipato alla richiesta.
- Lampeggio del giallo di notte (interruttore "notte").

Esercizio 6 · Il carrello avanti e indietro

OBIETTIVO

Un vero esercizio di automazione con finecorsa, marcia ed emergenza.

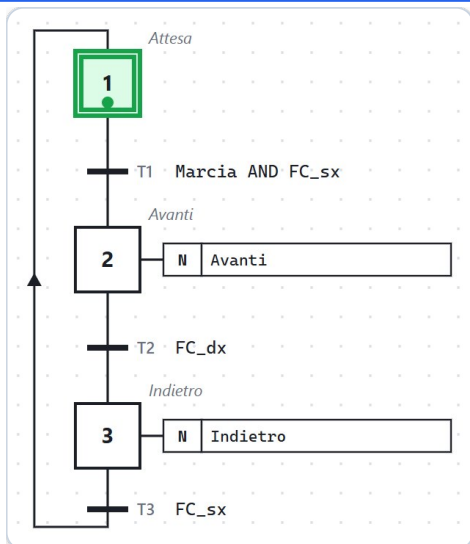
CONSEGNA

Con il carrello a sinistra (FC_sx) e il pulsante Marcia, il carrello va avanti fino al finecorsa destro (FC_dx), poi torna indietro fino a FC_sx e si ferma. L'emergenza ferma tutto e riporta la sequenza all'inizio.

MATERIALE

Pulsanti per Marcia, Emergenza (NC), due finecorsa; due LED o due relè per Avanti e Indietro.

SOLUZIONE GRAFCET · esempio "Carrello avanti e indietro (GRAFCET)"



Tappa 1 *Attesa*; T1 = `Marcia AND FC_sx`; tappa 2 *Avanti* (N Avanti); T2 = `FC_dx`; tappa 3 *Indietro* (N Indietro); T3 = `FC_sx` torna alla 1.

Condizione di reinizializzazione: `Emergenza`. Un segmento LADDER di sicurezza resetta Avanti e Indietro con l'emergenza.

VARIANTI

- Pausa di 3 s a destra prima di tornare (tappa con `X4.T > T#3S`).
- Contare i cicli e fermarsi dopo 5 (azione P1 con un contatore).
- Ciclo continuo con un selettore automatico/singolo.

Esercizio 7 · Il termostato

OBIETTIVO

Grandezze analogiche, conversione e isteresi.

CONSEGNA

Un potenziometro simula un sensore di temperatura da 0 a 50 °C. La ventola si accende sopra 28 °C e si spegne solo sotto 26 °C; il display mostra la temperatura con un decimale.

MATERIALE

Potenziometro su A0, LED o relè per la ventola, display OLED facoltativo.

SOLUZIONE FBD · esempio "Termostato"



Ingresso analogico → **Scala** 0...1023 → 0...500 (decimi di grado) → **Soglia con isteresi** 280 / 260 → ventola. Il valore va anche al display OLED ("Temp:" con un decimale), insieme allo stato della ventola.

DOMANDE PER LA CLASSE

- Perché serve l'isteresi? Provare con un semplice Confronto: cosa succede vicino a 28 °C?
- Sostituire il potenziometro con un sensore DS18B20 (capitolo 13).

18 Limiti del prodotto

ArduLearn è uno strumento didattico: ecco cosa non fa, per scegliere gli esercizi giusti.

Limite	Dettaglio
Non è un PLC di sicurezza	Non va usato per comandare macchine reali o proteggere persone. Gli arresti di emergenza veri devono agire direttamente sull'alimentazione, non passare dal programma.
Dimensione del programma	Massimo 96 blocchi (anche LADDER e GRAFCET diventano blocchi interni). Spazio: 7.680 byte (UNO R4), 8.192 byte (Mega con microSD), 3.072 byte (Mega senza microSD). Testi dei display e script: 1.024 byte (UNO R4), 512 byte (Mega).
Solo numeri interi	Niente numeri con la virgola: si lavora in decimi o centesimi. La divisione è intera ($7 \div 2 = 3$).
Memorie	Merker %M0.0–%M7.7 e parole %MW0–%MW30. I loro valori, i contatori e i temporizzatori ripartono da zero a ogni accensione: non esistono memorie "ritentive".
Script	ST: 16 variabili per script, codice fino a 400 byte, niente CASE. C: una funzione per blocco, niente virgola, array, stringhe, accesso diretto ai pin. Massimo 4000 istruzioni per ciclo.
Velocità	Adatto a segnali "umani" (pulsanti, finecorsa, sensori lenti). Non legge impulsi velocissimi (encoder industriali ad alta risoluzione, contatori veloci).
Corrente dei pin	Circa 20 mA (Mega) e 8 mA (UNO R4 WiFi) per pin, a 5 V: ogni carico più grande di un LED richiede relè o driver.
Rete Wi-Fi (UNO R4)	Solo 2,4 GHz, reti con password semplice (non WPA2-Enterprise/eduroam). Reti che isolano i dispositivi impediscono di vedere la scheda. Circa 7 collegamenti contemporanei: con classi numerose alcuni PC possono dover ricaricare la pagina.
Rete via cavo (Mega)	La scheda risponde a una richiesta alla volta: con molti studenti collegati la pagina può essere lenta. Mentre cerca l'indirizzo in rete (DHCP) il ciclo può fermarsi qualche secondo.
Pagina	Pensata per PC con mouse e schermo grande: su telefono non è utilizzabile.
Linguaggi	Un progetto usa un solo linguaggio (il GRAFCET include segmenti LADDER); non c'è conversione automatica fra FBD, LADDER e GRAFCET.
Aggiornamenti	UNO R4 dalla pagina (serve Internet sul PC); Mega solo via USB con l'app.
Schede supportate	Solo UNO R4 WiFi e Mega 2560 con shield DFR0850. Non Arduino Uno classico, UNO R4 Minima o altre schede.
Uso come Arduino normale	Sulla UNO R4 WiFi ArduLearn sostituisce il firmware del modulo Wi-Fi: per tornare a usare le librerie Wi-Fi di Arduino si deve riscrivere il firmware originale del modulo Wi-Fi dall'IDE Arduino (Strumenti → Aggiornamento firmware). Sulla Mega basta caricare un altro sketch.

19 Problemi e soluzioni

Le situazioni più frequenti in laboratorio, e cosa fare.

Problema	Cosa fare
L'app non trova la scheda	PC e scheda devono essere sulla stessa rete. Controlla il firewall di Windows (deve permettere l'app). Sulla UNO R4 guarda l'indirizzo che scorre sulla matrice e aprilo a mano. Alcune reti Wi-Fi scolastiche isolano i dispositivi: serve una rete dedicata.
La pagina non si apre con il nome .local	Usa l'indirizzo numerico (dall'app, dalla matrice o dal display). Il nome .local non funziona su tutte le reti.
La UNO R4 non si collega al Wi-Fi	Dopo 3 tentativi torna sulla sua rete ArduLearn-xxxx: ricollegati e ricontrolla nome e password. Reti a 5 GHz o con nome utente personale non sono supportate.
"PIN errato" / PIN dimenticato	App → Monitor seriale → <code>pin reset</code> : il PIN torna 1234.
"Carica sulla scheda" dice di correggere i problemi	Leggi l'elenco nel pannello a destra (clic su un problema = porta al blocco): pin non scelto, pin riservato, blocchi senza collegamenti necessari, troppi blocchi.
I pulsanti virtuali non fanno nulla	Il programma deve essere caricato e senza modifiche in sospeso ("✓ uguale alla scheda").
Il pulsante funziona "al contrario"	Con l'ingresso pull-up attiva Inverti (o "logica inversa" in LADDER/GRAFCET).
Un'uscita non si accende	Usa Prova uscita nella scheda Pin (a programma fermo) per separare i problemi di cablaggio da quelli di programma. Ricorda il limite di corrente del pin.
Uno studente ha "perso" il lavoro	Riaprendo la pagina sullo stesso PC le modifiche non caricate vengono riprese. Per il futuro: Scarica file o uno slot.
Le modifiche di uno studente vengono sostituite	È attivo "Segui il docente": toglie la spunta per lavorare in autonomia.
Il display non mostra niente	Impostazioni → Cerca dispositivi I ² C; controlla indirizzo e tipo (SSD1306/SH1106). Sulla Mega i fili vanno a 20/21, sulla UNO R4 ad A4/A5.
La Mega si riavvia aprendo il monitor seriale	È normale: il programma riparte dopo qualche secondo.
Il caricamento del firmware fallisce	Chiudi IDE Arduino e altri programmi che usano la porta, cambia cavo USB; sulla UNO R4 premi due volte il tasto RESET e riprova.
Sulla matrice LED compare una X	Il PLC non riesce a comunicare con il modulo Wi-Fi: scollega e ricollega l'alimentazione; se continua, ricarica il firmware con l'app.
"Aggiornamento non verificabile"	Il PC non raggiunge Internet (GitHub): aggiorna da un PC collegato o con l'app.

A Appendice

Comandi del monitor seriale, glossario e contatti.

Comandi del monitor seriale (115200 baud)

Comando	Cosa fa	Schede
<code>help</code>	elenco dei comandi	entrambe
<code>info</code>	versione, nome, rete, indirizzo, programma, memoria libera	entrambe
<code>run</code> / <code>stop</code>	avvia / ferma il programma (ricordato)	entrambe
<code>io</code> , <code>io on</code> , <code>io off</code>	stato di ingressi, uscite e blocchi (una volta / ogni secondo / basta)	entrambe
<code>pin reset</code>	PIN docente di nuovo 1234	entrambe
<code>sd</code>	diagnosi della microSD	Mega
<code>ip dhcp</code> · <code>ip A M G</code>	indirizzo automatico · indirizzo fisso (anche dai campi dell'app)	Mega
<code>mac ...</code>	cambia l'indirizzo MAC (ogni scheda della rete deve averne uno diverso)	Mega
<code>wifi</code>	stato del Wi-Fi	UNO R4
<code>wifi clear</code>	dimentica la rete e riattiva la rete ArduLearn-xxxx	UNO R4
<code>wifi ap</code>	passa alla rete della scheda senza dimenticare quella salvata	UNO R4

Glossario

PLC	Controllore logico programmabile: il "cervello" delle macchine automatiche.
Ciclo (scansione)	Un giro completo: lettura ingressi, esecuzione del programma, scrittura uscite.
FBD	Function Block Diagram: programma a blocchi collegati.
LADDER (KOP)	Linguaggio a contatti, simile a uno schema elettrico.
GRAFCET / SFC	Rappresentazione a tappe e transizioni delle sequenze.
Recettività	La condizione scritta su una transizione del GRAFCET.
Merker	Memoria interna di un bit (%M), usata come "relè ausiliario".
Pull-up	Modo di un ingresso che lo tiene a 1 a riposo: il pulsante verso GND lo porta a 0.
PWM	Uscita che si accende e spegne velocissima per regolare luminosità o velocità (0–255).
Fronte	L'istante in cui un segnale cambia; in ArduLearn dura un ciclo.
Firmware	Il programma "di base" installato sulla scheda con l'app: ArduLearn stesso.
Slot	Uno dei 99 posti in cui la scheda conserva i programmi salvati.
I²C	Il collegamento a 4 fili (GND, 5V, SDA, SCL) usato da display e orologio.
DHCP / .local	Il modo in cui la rete dà un indirizzo alla scheda / il nome facile della scheda (es. ardulearn-2b28.local).

Contatti e aggiornamenti

ArduLearn è stato ideato e realizzato da **Daniele Michelotti**. È software libero (licenza GPL-3.0): si può usare, copiare e migliorare liberamente.

E-mail: daniele.michelotti@gmail.com

Progetto, app e nuove versioni: github.com/danielemichelotti/ArduLearn

ArduLearn è gratuito: se ti è utile, puoi sostenerlo con una donazione su PayPal: paypal.me/danistn

Arduino, UNO e Mega sono marchi di Arduino S.r.l.; la shield Ethernet & PoE DFR0850 è un prodotto DFRobot. ArduLearn è un progetto indipendente, non affiliato a questi produttori.