

Tutorial: 3-LED Finite-State Machine (FSM) controlled from Serial & Serial1

Control three LEDs with a simple state machine from either serial port. You'll type commands like S0, S1, S2, NEXT, PREV, STATUS. Both ports stay in sync and the active state lights exactly one LED.

Full Sketch

```
// --- Simple 3-state FSM with 3 LEDs, controlled by Serial and Serial1 ---
current = static_cast<State>((current + 2) % 3); // -1 mod 3
applyOutputs(); return true;
}
if (cmd.equalsIgnoreCase("STATUS")) {
announceState(); return false;
}
// Unknown command
Serial.println("Unknown cmd. Use: S0 S1 S2 NEXT PREV STATUS");
return false;
}

void tryReadLine(Stream& port, String& buffer) {
// Non-blocking line reader (ends on
). Trims CRLF.
while (port.available()) {
char c = (char)port.read();
if (c == '
') continue;
if (c == '
') {
buffer.trim();
if (buffer.length() > 0) {
bool changed = handleCommand(buffer);
if (changed) announceState();
}
buffer = "";
} else {
// limit growth to avoid runaway if no newline ever comes
}
```

```

if (buffer.length() < 64) buffer += c;
}
}
}

void setup() {
for (uint8_t i = 0; i < 3; ++i) {
pinMode(LED_PINS[i], OUTPUT);
digitalWrite(LED_PINS[i], LOW);
}

Serial.begin(9600); // Laptop USB
Serial1.begin(9600); // Adafruit TTL

delay(300);
applyOutputs();
Serial.println("FSM ready. Type: S0 S1 S2 NEXT PREV STATUS");
Serial1.println("FSM ready. Type: S0 S1 S2 NEXT PREV STATUS");
announceState();
}

void loop() {
// Read from both ports
tryReadLine(Serial, bufUSB);
tryReadLine(Serial1, bufTTL);
}

```

1) Hardware: LED wiring

The LED's long leg = anode (+) to the Arduino pin; short leg = cathode (-) through the resistor to GND.

Use three LEDs with resistors (220–1kΩ). One LED per state.

LED	Arduino	Wiring
	pin	
LED 0	D8	Anode to D8; Cathode to resistor(+); resistor(-) to GND

LED 1	D9	Anode to D9; Cathode to resistor(+); resistor(-) to GND
LED 2	D10	Anode to D10; Cathode to resistor(+); resistor(-) to GND

-Hardware: Serial1 (TTL) wiring

Cable wire	Function	Connect to Mega
Black	GND	GND
White	RX	TX1 (pin 18)
Green	TX	RX1 (pin 19)
Red	+5 V	<i>Not needed</i> (leave unconnected if Mega is USB-powered)

3) How to use

1. Upload the sketch via the Mega's USB port (select /dev/ttyACM*). Ensure pins 0/1 are free.
2. Open two serial monitors at 9600 baud:
 - /dev/ttyACM0 : main USB Serial
 - /dev/ttyUSB0 : Adafruit USB-TTL (Serial1)
3. Type commands in either window (press Enter):

Command	Effect
S0	Turn on LED0 (D8), others off
S1	Turn on LED1 (D9), others off
S2	Turn on LED2 (D10), others off
NEXT	Cycle forward (S0, S1, S2, S0...)

PREV Cycle backward

STATUS Print current state on both ports

You'll see STATE=S0/S1/S2 echoed to both ports whenever the state changes.

4) How it works

- Finite-State Variable: current tracks which of the three states is active.
- Outputs: applyOutputs() ensures only the active state's LED is HIGH.
- Parsing: tryReadLine() reads newline-terminated commands from each port without blocking.
- Control: handleCommand() changes state or reports status; it returns whether the state changed so announceState() can broadcast updates.

5) Troubleshooting

- No LED activity: Check LED orientation and resistor to GND; verify pin numbers 8/9/10.
- Only one monitor shows text: Open both /dev/ttyACM* and /dev/ttyUSB* at 9600; ensure the TTL cable uses pins 19/18 (not 0/1).
- Upload timeouts: Close all monitors, select the ACM port for upload, disconnect anything from pins 0/1, retry.
- Commands ignored: Make sure you press Enter.