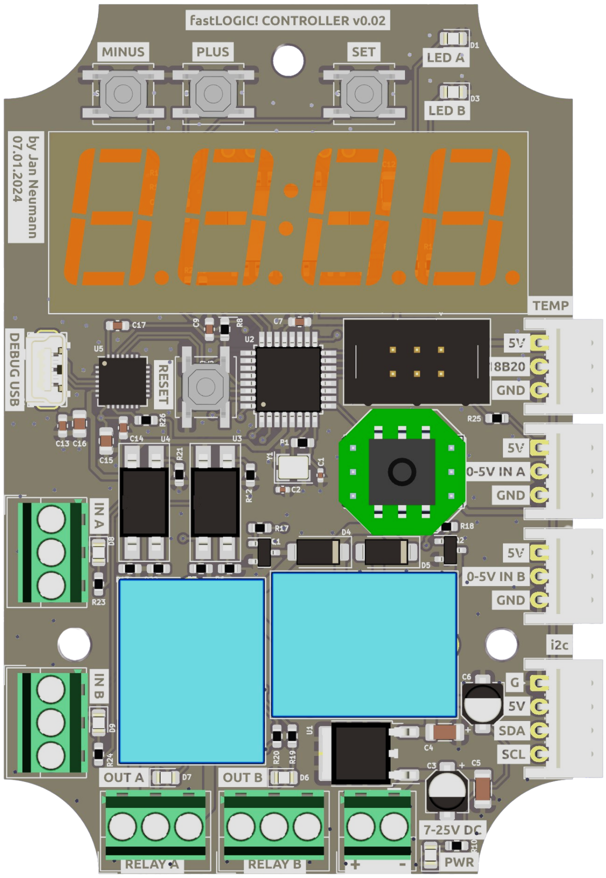
FastLOGIC! industrial controller
hardware revision 0.02_07.01.2024
document version 1.0



Description

FastLOGIC! is a development board to get started with simple automation tasks. It could be used for experimentation and education but also for small permanent automation installations. A number of standard and easy-to-use in- and outputs give the user fast access to the world of automation.

Use cases could range from a simple sensor data display to complex logic operations and decision-making based on sensor inputs to switch outputs.

Target users

Makers, Hobbyists, Engineers

Features

- **ATMega328p microcontroller**
    - AVR CPU at 16MHz
    - 32KB Flash
    - 2KB SRAM
    - 1KB EEPROM
- **Power**
    - 7-25V DC supply
    - 5V logic and sensor voltage
    - 34mA idle current at 12V
- **Peripherals I/O**
    - USB / UART
    - 2x digital input
    - 3x button input
    - 2x orange LED output
    - 4-Bit 7-Segment red display
    - DS18b20 temperature sensor
    - XGZP6847A pressure sensor socket
    - 2x 10-Bit 0-5V analog input
    - I2C interface
    - Serial Peripheral Interface (SPI)
    - 2x relay output
    - Watchdog timer
- **Electrical connectors**
    - Mini USB vertical
    - 3.5mm screw terminal for relays and digital inputs (3-pin)
    - 3.5mm screw terminal for power (2-pin)
    - JST EH sensor inputs (3-pin)
    - JST EH i2C (4-pin)
    - XGZP6847A pressure sensor socket
    - 6-pin ICSP programming header
- **Physical properties**
    - PCB only: 90 * 60 * 19 mm (L * W * H), 43g
    - With Housing: 100 * 68 * 50 mm (L * W * H), 98g
    - M3 mounting: triangle 45 * 62 mm (A * h)
- **Software**
    - Programmable via Arduino IDE
    - FastLOGIC! library

1 Application Examples and Use Cases

For **educational, hobby, and professional users** FastLOGIC! provides an easy start to automation as the well-known programming language C++ and the Arduino IDE can be used out of the box. Everything that runs on an Arduino UNO runs on FastLOGIC! as well.
Standard sensors and interfaces make FastLOGIC! easily adaptable and expandable.

Possible use cases could be:

- display to read sensor data
- a timer that switches an output for a defined time after a button press
- RPM counter using an induction sensor
- water level sensor using the XGZP6847A pressure sensor
- temperature controller using DS18b20 sensors with EEPROM set value data memory

2 Ratings

Symbol	Description	Min	Typ	Max	Unit
Vin	Input supply voltage on screw terminal	7.0	12.0	25.0	V
U	Current draw at 12V input voltage	0.04	-	0.3	A
P	Power consumption at 12V	0.4	-	3.6	W
Vusb	USB supply voltage	4.5	5.0	5.5	V
Urel	Switchable relay current	-	-	10	A
Irel	Switchable relay voltage	-	-	30	V
Vtrig	Digital input A and B trigger voltage	-	9.5	-	V
Vanalog	Analog input voltage	-0.5	-	5.5	V
Tmax	Safe temperature range	-40	-	+85	°C

NOTE: In extreme temperatures, EEPROM, voltage regulator, analog reference, and the crystal oscillator, might not work as expected!

3 Functional Overview

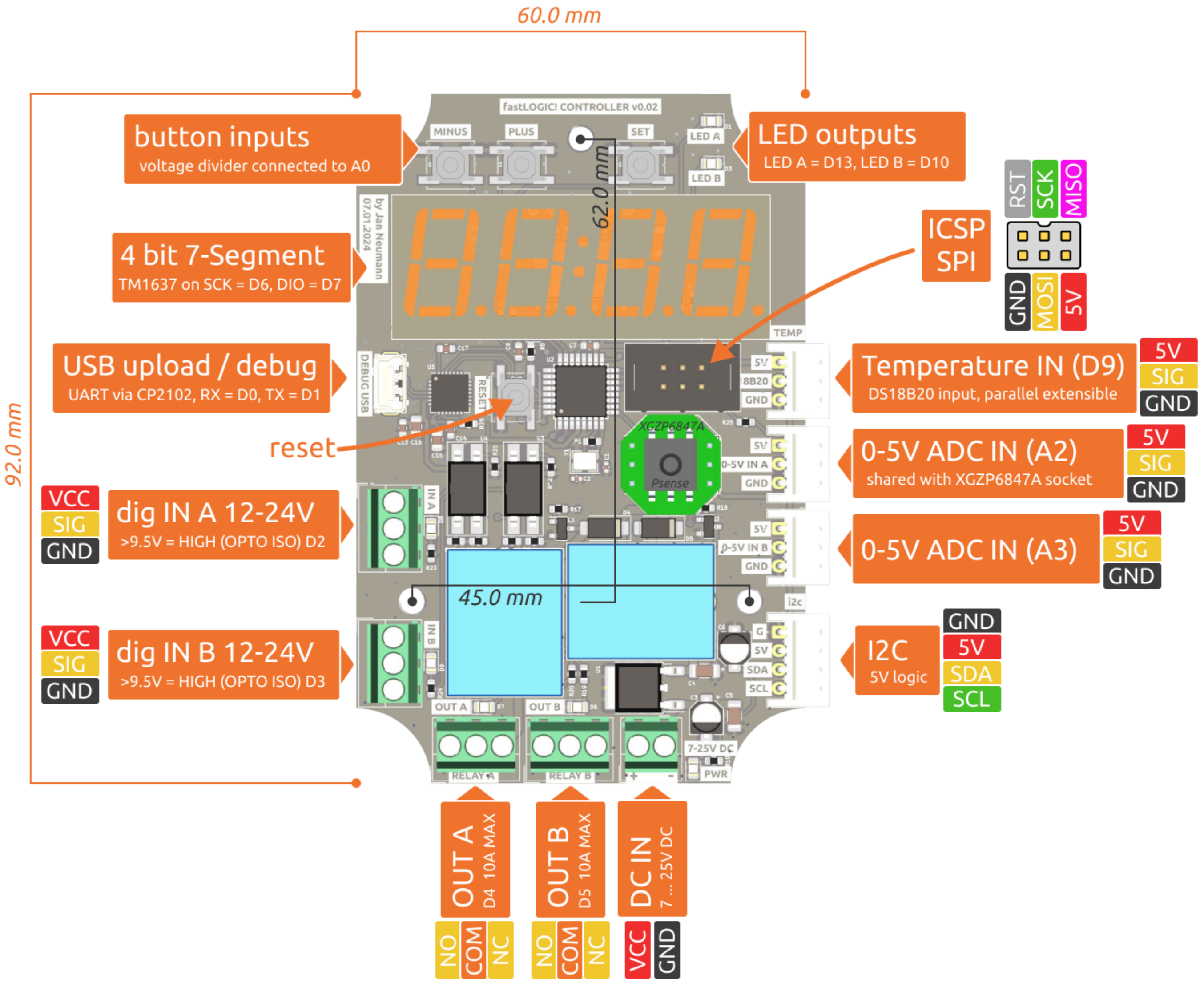


Figure 1: Functional overview of inputs, outputs, dimensions, and pinouts

4 Getting Started

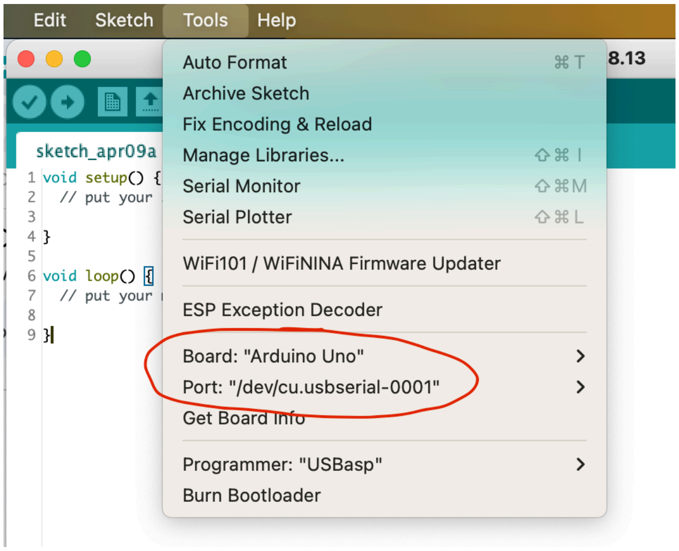
Connect

To program the board using the Arduino IDE, use a Mini USB cable to connect it to the computer. The cable also provides 5V power to run the board.

NOTE: The computer must provide enough current to power all connected sensors!

IDE Setup

The board behaves just like an Arduino UNO board. Therefore select Ardunino UNO as the development platform. The board has to be "Arduino Uno". Select the corresponding serial port for your board.



```
void setup() {
  // put your
 
}
 
void loop() {
  // put your
 
}
```

Board: "Arduino Uno"

Port: "/dev/cu.usbserial-0001"

Programmer: "USBasp"

Figure 2: Arduino IDE board settings for uploading code to FastLOGIC!

Use the upload button to upload the code.

Note: Press the reset button on FastLOGIC! while the code is uploading. **Otherwise, the upload will fail.**

Programming

Use the examples from the GitHub repository to get started. It helps you use the ports, sensors, and relays with simple OOP methods.
You can use all the standard functionalities, the Arduino UNO supports like millis, sleep, EEPROM, etc. Inputs can be read using digitalRead and analogRead. The relays can be switched using digitalWrite on the corresponding pins.

Pin mapping

Each functionality is connected to a certain pin on the ATMega328. The pin mapping shows each pin and its associated function.

Pin name	Pin number	Description
RX	D0	directly connected to CP2102 USB / Serial chip
TX	D1	directly connected to CP2102 USB / Serial chip
DIGITAL_IN_A	D2	reads HIGH when input A HIGH (>9.5V)
DIGITAL_IN_B	D3	reads HIGH when input B HIGH (>9.5V)
RELAY_B_OUT	D4	output HIGH drives the relay B
RELAY_A_OUT	D5	output HIGH drives the relay A
TM1637_SCK	D6	TM1637 display driver clock pin
TM1637_DIO	D7	TM1637 display driver data pin
NC	D8	not connected
DS18B20_DATA	D9	DS18B20 temperature sensor data pin (pulled HIGH by 4.7k)
LED_B_OUT	D10	HIGH enables LED B
LED_A_OUT	D13	HIGH enables LED A
VOLTAGE_SENS_IN	A0	reads Vin via 1 : 4.45 voltage divider **(Vin 25V MAX!)**
BUTTONS_IN	A1	reads buttons: SET: <20; MINUS: >25 & <45; PLUS: >320 & <470
ANALOG_IN_A	A2	reads analog IN A and XGZP6847A pressure sensor
ANALOG_IN_B	A3	reads analog IN B 0-5V **(5.5V MAX!)**
I2C_SDA	A4	I2C bus data pin SDA
I2C_SCL	A5	I2C bus clock pin SCL