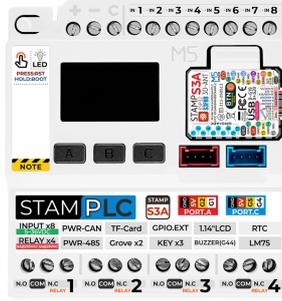
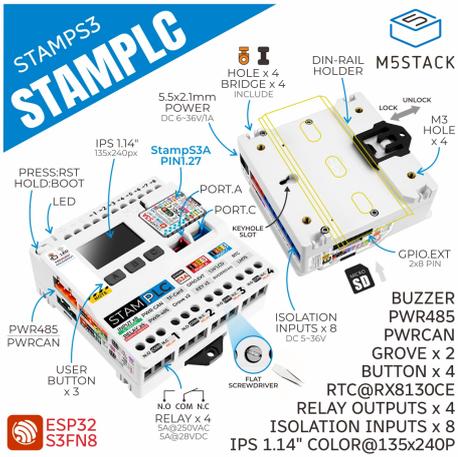
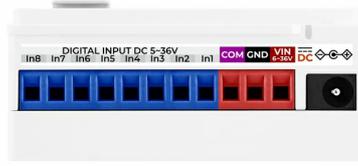


# StamPLC

SKU:K141





## Description

**StamPLC** is an IoT Programmable Logic Controller designed for industrial automation and remote monitoring. The product uses the **Stamp-S3A** control module, offering strong processing power and efficient **wireless** connectivity. In terms of **control**, StamPLC provides 8-channel optocoupler isolated digital inputs and 4-channel relay outputs (supporting AC / DC loads), along with GPIO.EXT and 2 HY2.0-4P ports, making the connection of various sensors and actuators simple and reliable. With onboard **PWR-CAN** and **PWR-485** interfaces, the device can be seamlessly integrated into industrial field bus networks to achieve remote data transmission and centralized control. For human-machine interaction, the product features a 1.14-inch color **display**, RESET/BOOT keys, 3 user buttons, and a **buzzer**, enabling users to configure parameters and monitor status in real-time, and promptly alert in case of anomalies. To suit harsh industrial environments, StamPLC supports DC 6 ~ 36V wide-voltage power supply and is designed for **DIN** rail installation to ensure sturdy mounting; the built-in **microSD** slot facilitates data storage and firmware upgrading. Its environmental monitoring system integrates the LM75 temperature sensor and INA226 voltage/current sensor to provide real-time feedback on operating status, while the RTC (RX8130CE) module ensures accurate time synchronization and log recording. The factory firmware by default uploads data to M5's **EZData** cloud platform, automatically generating monitoring pages, offering users a convenient remote cloud **access and control** method. This product is suitable for industrial automation, remote monitoring, and smart manufacturing applications.

## Tutorial



## Quick Start

StamPLC built-in firmware usage tutorial.



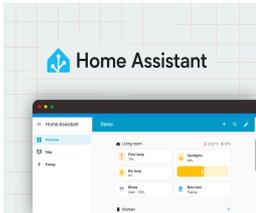
## Arduino IDE

This tutorial will show you how to program and control StamPLC devices using Arduino IDE.



## UiFlow2

This tutorial will show you how to control StamPLC devices using the UiFlow2 graphical programming platform.



## Home Assistant

This demo will introduce how to add the device into Home Assistant

## Note

### Power Notes

The PWR-CAN and PWR-485 output power pins of StamPLC are directly connected to the main input power, so please ensure that the input voltage matches the power requirements of the external expansion device during use. For example, when using StamPLC to control Unit Roller485 or Unit RollerCAN, it is recommended to keep the input voltage between DC 6 ~ 16V to prevent damage caused by voltage mismatch.

## Features

- Stamp-S3A control module (ESP32-S3FN8)
- EZData cloud monitoring
- 8-channel optocoupler isolated digital input
- 4-channel relay output (AC / DC)
- 1.14-inch color screen (ST7789v2)
- PWR-CAN & PWR-485 interface
- Wide voltage input (DC 6 ~ 36V)
- Integrated INA226 voltage / current sensor, capable of measuring the voltage and current of the VIN input on the DC power interface and the EXT expansion interface
- Internal temperature sensor
- User buttons
- microSD slot
- DIN rail installation
- Buzzer
- HY2.0-4P port

- RTC module
- Development Platform
  - UiFlow2
  - Arduino IDE
  - ESP-IDF
  - PlatformIO

## | Includes

---

- 1 x StamPLC
- 4 x Mounting accessories
- 4 x Screw clamp mounting parts

## | Applications

---

- Industrial automation and remote control
- Distributed control systems
- Smart manufacturing

## | Specifications

---

Specification	Parameter
Module Model	Stamp-S3A Control Module
SoC	ESP32-S3FN8@Xtensa LX7 dual-core, main frequency up to 240MHz
Flash	8MB
Wi-Fi	2.4 GHz Wi-Fi
Digital Input	8-channel optocoupler isolated digital input, input voltage range: DC 5 ~ 36V
Digital Output	4-channel relay output
Relay	AC 5A@250V DC 5A@28V
DC Power Supply	Supports DC 6 ~ 36V @ 1A wide-voltage supply DC power interface: DC5521 female 5.5 x 2.1mm (inner positive outer negative)
Expansion Port	GPIO.EXT port, 2 HY2.0-4P ports
Communication Port	Onboard PWR-CAN and PWR-485 interfaces
PWR-CAN Port	XT30 (2+2) PW-M
PWR-485 Port	HT3.96-4P
Display	1.14-inch color display (135×240 resolution), driver chip ST7789v2
Interaction	1 RESET/BOOT button, 3 user buttons, buzzer
Data Storage	Built-in microSD slot
Sensor	LM75 temperature sensor, INA226 voltage/current sensor, RTC (RX8130CE)
IO Capacity	2x8 expansion port max output: DC 4.76V@700mA HY2.0-4P port load capacity test: DC 4.81V@700mA
Power Consumption	Standby current: (5V supply) DC 5V@21.60mA, (12V supply) DC 12V@15.22mA Working current: (5V supply) DC 5V@93.89mA, (12V supply) DC 12V@47.84mA
Installation	DIN rail installation
Operating Temp	0 ~ 40°C
Product Size	72.0 × 80.0 × 33.4mm
Product Weight	140.0g
Package Size	102.0 x 94.0 x 37.0mm

Specification	Parameter
Gross Weight	163.7g

## Learn

### Download Mode

#### Download Mode

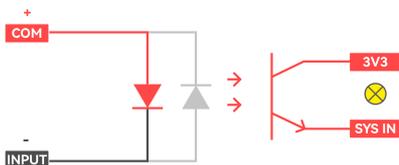
To enter download mode, after connecting the data cable, hold down the Boot button until the red indicator light turns on, then release the Boot button to enter download mode.



### Signal Input Sensor Wiring Method

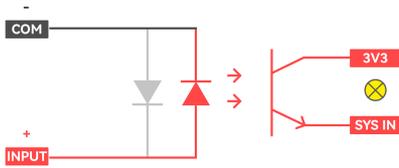
The input channel integrates optocoupler isolation, supporting DC 5 ~ 36V high/low level signal input acquisition to adapt to different sensor input types.

- Low-level signal input wiring:
  - COM connects to the positive terminal of sensor power supply
  - INPUT connects to sensor input signal



- High-level signal input wiring:
  - COM connects to negative terminal of sensor power supply

- INPUT connects to sensor input signal

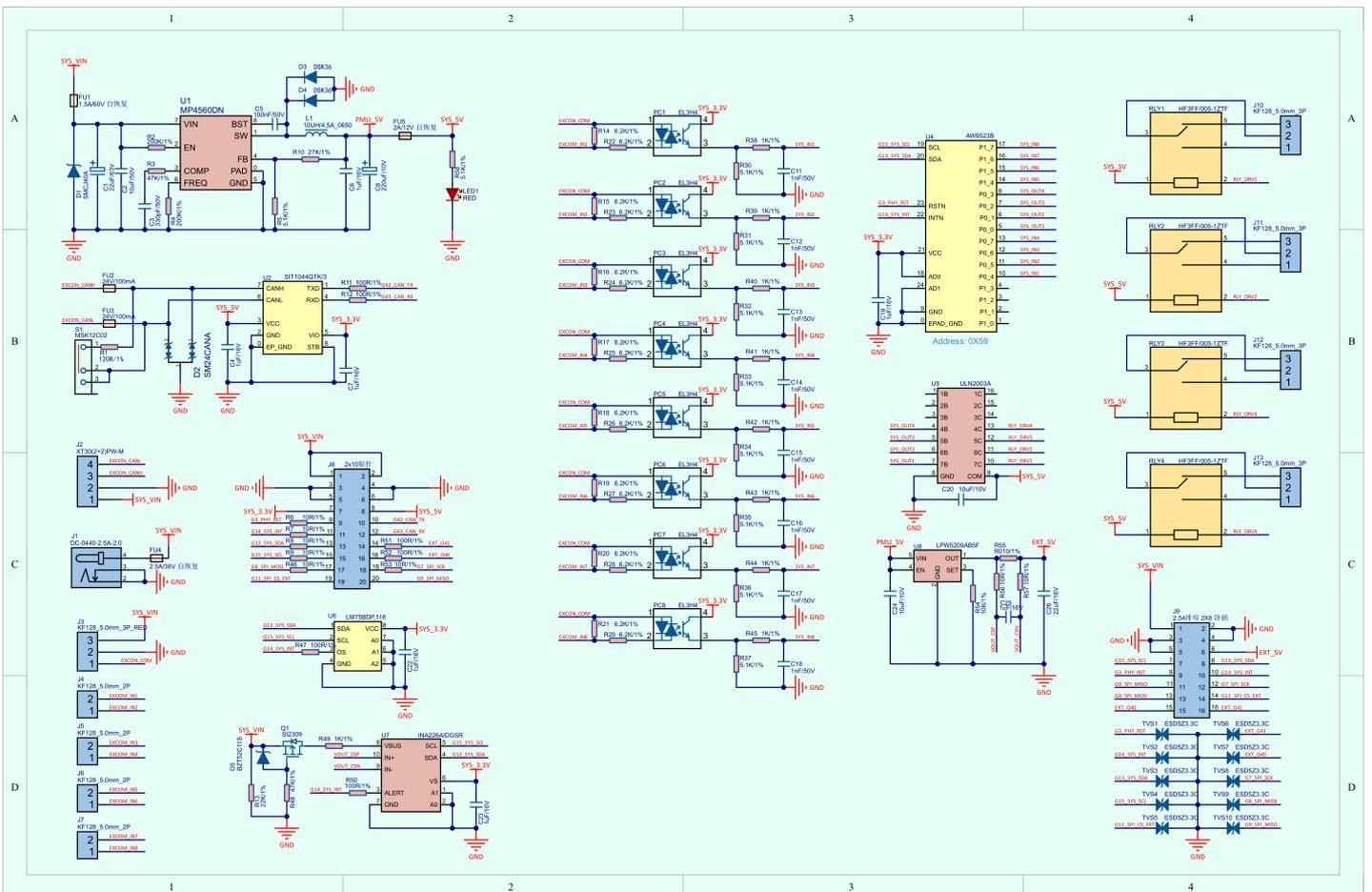
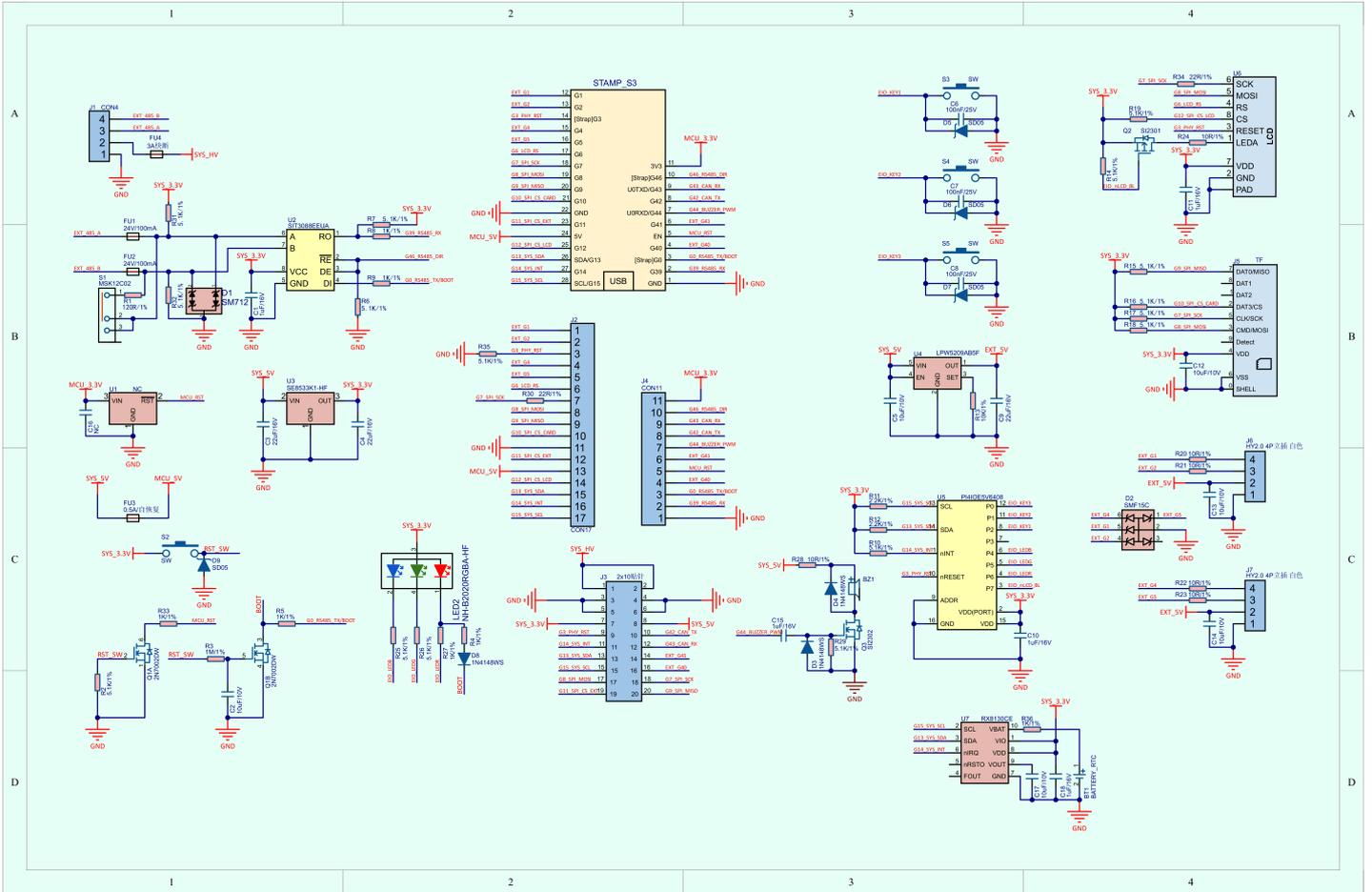


### RGB LED Control Instructions

NH-B2020RGBA-HF is an RGB LED device that can only achieve color combinations by separately controlling the On/Off state of R, G, and B LEDs, and does not support PWM dimming for brightness control or gradient effects.

## Schematics

- [Stamp-S3A Main Board Schematics PDF](#)
- [StamPLC\\_IO Schematics PDF](#)



# PinMap

## PWR-485 & PWR-CAN

ESP32-S3FN8	G42	G43	G0	G39	G46
PWR-CAN	CAN_TX	CAN_RX			
PWR-485			RS485_TX/BOOT	RS485_RX	RS485_DIR

## RGB & User Button A/B/C

◦ Controlled via PI4IOE5V6408

ESP32-S3FN8	G15	G13	G14	G3
PI4IOE5V6408(0x43)	SCL	SDA	INT	RST

PI4IOE5V6408	P6	P5	P4	P2	P1	P0
RGB	R	G	B			
Button				KEYA	KEYB	KEYC

## microSD Card Slot

ESP32-S3FN8	G9	G10	G7	G8
microSD	MISO	CS	SCK	MOSI

## Sensor

ESP32-S3FN8	G15	G13	G14
INA226AIDGSR (0x40)	SCL	SDA	INT
LM75BDP (0x48)	SCL	SDA	INT
RX8130CE (0x32)	SCL	SDA	INT

## LCD

ESP32-S3FN8	G8	G7	G6	G12	G3
LCD	MOSI	SCK	RS	CS	RST

PI4IOE5V6408	P7
LCD	LCD_BL

## Buzzer

ESP32-S3FN8	G44
Buzzer	BUZZER_PWM

## Relay & Optocoupler

- Controlled via AW9523B (0x59)

ESP32-S3FN8	G15	G13	G14	G3
AW9523B(0x59)	SCL	SDA	INT	RST

- Relay Control

AW9523B	P0_0	P0_1	P0_2	P0_3
Relay	RLY_DRV1	RLY_DRV2	RLY_DRV3	RLY_DRV4

- Optocoupler Output

AW9523B	P0_4	P0_5	P0_6	P0_7	P1_4	P1_5	P1_6	P1_7
EL3H4	SYS_IN1	SYS_IN2	SYS_IN3	SYS_IN4	SYS_IN5	SYS_IN6	SYS_IN7	SYS_IN8

- Optocoupler Input

EL3H4	Function Description
EXCOM_IN1	External Input Signal 1
EXCOM_IN2	External Input Signal 2
EXCOM_IN3	External Input Signal 3
EXCOM_IN4	External Input Signal 4
EXCOM_IN5	External Input Signal 5
EXCOM_IN6	External Input Signal 6
EXCOM_IN7	External Input Signal 7
EXCOM_IN8	External Input Signal 8
EXCOM_COM	Common Terminal

## StamPLC-Bus

PIN	LEFT	RIGHT	PIN
VIN	1	2	GND
GND	3	4	GND
GND	5	6	5V_OUT
SCL	7	8	SDA
PHY_RST	9	10	INT
MISO	11	12	SCK
MOSI	13	14	CS
Custom	15	16	Custom

## HY2.0-4P

○ PORT.A

HY2.0-4P	Black	Red	Yellow	White
PORT.A	GND	5V	G2	G1

○ PORT.C



- [StamPLC Structure Files](#)

## Softwares

---

### Quick Start

- [StamPLC Quick Start](#)

### Arduino

- [StamPLC Arduino Quick Start](#)
- [StamPLC Arduino Library](#)

### UiFlow2

- [StamPLC UiFlow2 Quick Start](#)

### ESP-IDF

- [StamPLC Factory Firmware](#)

### PlatformIO

```
[env:m5stack-stamp-s3]
platform = espressif32
board = esp32-s3-devkitc-1
framework = arduino
upload_speed = 1500000
build_flags =
  -DESP32S3
  -DCORE_DEBUG_LEVEL=5
  -DARDUINO_USB_CDC_ON_BOOT=1
  -DARDUINO_USB_MODE=1

lib_deps =
  M5StamPLC=https://github.com/m5stack/M5StamPLC
  M5Unified=https://github.com/m5stack/M5Unified
```

### Protocol

- Modbus

By default, StamPLC firmware initializes the Modbus slave after startup. External devices can control the StamPLC via the PWR-485 interface using Modbus RTU protocol, with the following register mapping.

Register Map:

1. Coils (Read/Write)
  - Address 0: Relay 1 output (true/false)
  - Address 1: Relay 2 output (true/false)
  - Address 2: Relay 3 output (true/false)

- Address 3: Relay 4 output (true/false)

## 2. Input Registers (Read-only)

- Address 0-7: Inputs (true/false) - 8 registers
- Address 8-9: Temperature (FLOAT32) - 2 registers
- Address 10-11: Bus Voltage (FLOAT32) - 2 registers
- Address 12-13: Shunt Current (FLOAT32) - 2 registers

## Easyloader

Easyloader	Download	Note
<a href="#">StamPLC User Demo Easyloader</a>	<a href="#">download</a>	/

## Other

- [StamPLC Restore Factory Firmware Tutorial](#)

## Video

- [StamPLC product introduction and case demo](#)

[K141\\_video.mp4](#)