Documentation SynPLC

An OpenSource PLC system with STM32 Controllers

Made with LaTeX
Stand: April 13, 2023

Marco Müller

gitlab.synthron.de
synthron@synthron.de

Contents

I. List of Figures

II. List of Tables

1. About

This document will describe the workings of the SynPLC.

The SynPLC is meant to be an OpenSource PLC system for home automation purposes. It can supply digital and analog IO as well as different modules for a variety of different applications, e.g. Phase-Fired Control of AC voltages.

1.1. The Origins

This project was created of the idea that in theory one could build a LOGO!-type PLC system from Arduinos. After some research and a few failed attempts it evolved into the current state, using STM32 Microcontrollers.

1.2. License

2. Summary

The SynPLC project is a PLC system using STM32 microcontrollers and off-the-shelf hardware. For the current version the following modules are available:

- CPU module
- Digital Output module
- Digital Input module
- Analog Output module
- Analog Input module
- Triac PFC module

The following modules are planned, but not yet under development:

- PWM module
- Relais module
- Thermocouple module

For further information and technical specs, please look into the section ”Modules”.

3. Busses

Every module has at least one common bus connector connecting via ribbon cable to the other modules. Alongside the 5V and GND voltage supply there also is the communication bus lines. On the modules there are two different busses possible: RS485 and CAN.

In order to maintain a stable communication, consider the use of twisted-pair ribbon cables.

Both are working on a custom protocol, which is easy to implement on other controllers too.

3.1. RS485

3.1.1. Baudrate and Transmission speed

The RS485 bus is transceiving at 250kBaud. At a maximum of 10 Bytes per frame, this results in a minimum rate of around 3000 frames per second.

3.1.2. Frame Structure

The frame structure of the protocol is as follows:

Byte Number	Data
1	Startbyte (0x5A)
2	Slave Adress
3	Instruction and Register
4	Data (Not used when Instruction == Scan or Read)
5	Data (only used at 12- or 16bit data blocks)
6	Checksum
7	Answer from Slave (ACK or NACK with Error code)
8	Answer from Slave (data, LSB at 12- or 16bit data blocks)
9	Answer from Slave (data, MSB at 12- or 16bit data blocks)
10	Answer from Slave (Checksum)

As you can see, most of the frame consists of the CPU talking to the slaves. Startbyte and Checksums are in place to make sure that an error free communication is possible.

The Checksum itself is easily calculated. This is done by XOR-ing all sent bytes with the constant value 0xC5. In summary, the equation is as follows:

$$\text{Checksum} = 0\text{xC5} \oplus \text{Byte\_1} \oplus \text{Byte\_2} \oplus \text{Byte\_3}(\oplus\text{Byte\_4} \oplus \text{Byte\_5}) \tag{3.1}$$

Equally, the slaves are calculating their checksum with the same method, XOR-ing all their bytes with 0xC5.

If all the bytes are XOR-ed with each other and with the checksum itself and 0xC5 is still left, then the communication was error free and the data is uncompromized.

3.1.3. Slave Addresses

The slave addresses are an 8bit number which consist of a 3-bit module identifier and a 5-bit address settable via dip-switches on the respective module.

The module identifiers are definied as follows:

Identifier	Modul
000	Digital Input
001	Digital Output
010	Analog Input
011	Analog Output
100	PWM Modul
101	Thyristor Modul
110	Relais Modul
111	*reserved*

3.1.4. Instructions

In Byte 3 of the communication protocol are the instruction and the corresponding register specified. For more information on the registers, please refer to the corresponing module section in this document.

The instructions are at the 4 MSBs of the byte. They are as follows:

Instruction-Code	Instruction
0000 xxxx	Bus Scanning (is slave available?) - no data byte
0001 (reg)	Read 8bit-Register
0010 (reg)	Write 8bit-Register
0101 (reg)	Read 16bit-Register
0110 (reg)	Write 16bit-Register

3.1.5. ACK, NACK and Error Codes

In order to know weather the communication was successful, there is a byte used specifically for ACK/NACK and Error codes. This byte is divided into 2 nibbles, where the first nibble is the ACK/NACK and the second nibble is the error code. If everything is fine, the ACK nibble will be 0xC0 (1100). If there is an error present, it will display the NACK status with 0x03 (0011).

The following nibble will show which exactly went wrong:

Error Code	Meaning
0000	No Error
0001	Target Register invalid
0010	Instruction invalid
1100	Processing Error - read Error Register
1101	Checksum Error - Send again
1111	System Error - read Error Register

4. Modules V1

All V1 modules are now discontinued. The successors are the V2 modules.

Further information:

V1 was meant to be a Proof of Concept system using easy-to-program ATmega328P with Arduino framework.

The Hardware was not as refined as wanted and a lot of compromises were made resulting in high production costs and hard-to-source parts.

5. Modules V2

These are the current module revisions. If not stated otherwise, every module is powered by a STM32F103 microcontroller.

Every module is capable of both CAN and RS485, but not simultaneously. To select one of the two bus systems, set the Jumpers on the boards accodingly.

V2 was designed with the mindset of more generic circuitry without losing accuracy or stability. Even tho not every module was improved (talking about you, DO8...) major steps were made on the analog modules. In addition, the triac module (XP) is in testing now.

Also all V1 modules got discontinued and decommissioned.

5.1. CPU

The CPU is the heart of the PLC system and necessary for normal operation. It connects to all IO-modules and processes all the data.

The CPU has the following interfaces:

- RS485 and CAN Businterface
- USB-Interface for serial monitor
- ESP01-Modul as Webinterface (read-only as of now)
- SD-card for structured text (WIP) and log files
- DC/DC-Converter as power supply for the whole system

The DC/DC-Converter uses an ESD-safe 24V DC-input to generate 5V DC at 10W output power. This voltage is available at the bus connector and providing power to all modules in the system.

It has no seperate ID since it is a one-of-a-kind module. In later revisions, there will be measures for redundancy available, but not for now.

5.2. DO8

The DO8 (Digital Output 8-channel) is providing 8 channels of 24V compatible outputs with a current rating of 0.5A each. The channels are galvanically isolated from the system voltage via optoisolators.

The 24V Output voltage has to be supplied on the connector.

To prevent major faults caused by short circuits, the output channels are measured back and compared to their set value. If a channel differs from its value, the module sets an error bit and shuts off the faulty channel.

On the front are 8 LEDs, indicating the status of the output channels.

5.3. DI

The DI8 (Digital Input 8-channel) is providing 8 channels of 24V compatiple inputs, divided into two sets of 4 Inputs where both sets are galvanically isolated from each

other. The maximum input current is limited to 2mA. The channels are galvanically isolated from the system voltage via optoisolators.

For normal operation, the system ground fromt the inputs need to be connected.

The channels are ESD safe and can tolerate up to 33V DC. Reverse-Polarity and over-current protections are in place.

A digital Low-Level is defined as 0..5 VDC.

A digital HIGH-Level is defined as 10..33 VDC.

5.4. AO

The AO4 (Analog Output 4-channel) is providing 4 channels of analog output, each capable of supplying either 0..10V CV or 0..20mA CC at a resolution of 12bits. The switch between CV and CC is managed by a relay.

The output resolution is set at 16bits, giving a 0.15mV or 0.30μA resolution.

The channels are electrically isolated from the system, so at least 15V with common analog grond have to be provided.

On the front are 4 LEDs indicating if the channel is actively being used.

5.5. AI

The AI4 (Analog Input 4-channel) is providing 4 channels of analog inputs, capable of meauring 0..10V or 0..20mA each.

The input resolution is set at 12bits. At a maximum input volatage of 10V or input current of 20mA, this is giving a resolution of 2.4mV or 4.8mA.

The channels are electrically isolated from the system.

5.6. XT

The XT4 (Interface Triac 4-channel) is providing 4 triacs on two seperate voltage domains. The channels can be used to either switch AC voltage to a supply or even using them in a PFC (phase-fired controller) mode.

For PFC mode each voltage domain has a Zero-Crossing-Detection (ZCD) and can control the triacs with PWM. The ZCD also calculates the frequency of the AC voltage automatically and compensates for frequency drift.

Each channel is rated for up to 230V and 1A. Keep in mind that the triacs can become quite hot.

All channels are electrically isolated from the system. The two voltage domains are also isolated from each other.

Appendix