

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA
RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE



Département électrotechnique

Final Project Thesis

For obtaining the state engineering diploma in power engineering

Design and development of an industrial programmable logic controller (PLC) with its programming software and SCADA system, integrating AI, in compliance with the IEC 61131-3 industrial standard (Startup Final Year Project).

Souhaib Cherbal

Under the supervision of **Mr. Berkouk Abdelmajid** ENP

Presented and publicly defended on (July 03, 2025)

Jury composition:

President:	Mr. Boughrara Kamel	ENP
Supervisor:	Mr. Berkouk Abdelmajid	ENP
Examiner:	Mr. Belkacemi Rabie	ENP

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA
RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE



Département électrotechnique

Final Project Thesis

For obtaining the state engineering diploma in power engineering

Design and development of an industrial programmable logic controller (PLC) with its programming software and SCADA system, integrating AI, in compliance with the IEC 61131-3 industrial standard (Startup Final Year Project).

Souhaib Cherbal

Under the supervision of **Mr. Berkouk Abdelmajid** ENP

Presented and publicly defended on (July 03, 2025)

Jury composition:

President:	Mr. Boughrara Kamel	ENP
Supervisor:	Mr. Berkouk Abdelmajid	ENP
Examiner:	Mr. Belkacemi Rabie	ENP

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA
RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE



Département Électrotechnique

Mémoire de Projet de Fin d'Études

Pour l'obtention du diplôme d'ingénieur d'État en génie électrique

Conception et développement d'un automate programmable industriel (API) avec son logiciel de programmation et système SCADA, intégrant l'IA, en conformité avec la norme industrielle IEC 61131-3 (Projet de Fin d'Études Startup).

Souhaib Cherbal

Sous la direction de **M. Berkouk Abdelmajid ENP**

Présenté et soutenu publiquement le (03 juillet 2025)

Composition du jury :

Président :	M. Boughrara Kamel	ENP
Directeur :	M. Berkouk Abdelmajid	ENP
Examineur :	M. Belkacemi Rabie	ENP

الحمد لله
الذي بنعمته تم الصالحات

We would first like to thank Allah the Almighty and Merciful, for having given us the strength and patience to successfully complete this modest work.

ملخص

يعرض هذا المشروع نظام **PLC A1 CA**، وهو جهاز تحكم منطقي قابل للبرمجة من الجيل الجديد يتميز بمعالجة ثنائية النواة، وهندسة لوحات إلكترونية مطبوعة (PCB) معيارية، وواجهات اتصال شاملة تشمل USB-C و RS-485 وبروتوكولات لاسلكية لتطبيقات الصناعة 0.4. يتكامل **A1 CA** مع نظام ذكي يضم **IDE CA**، وهي بيئة برمجة مدعومة بالذكاء الاصطناعي. كما توفر منصة **SCADA CA** إشرافاً على مستوى المؤسسة مع عرض مرئي في الوقت الحقيقي. يتيح **Monitor Register CA** تشخيصاً دقيقاً في الوقت الفعلي وتحسين أداء النظام من خلال مراقبة مفصلة لسجلات وحدة التحكم. تشكل هذه المكونات معاً حلاً موحداً للأتمتة يلبي التحديات الصناعية الحالية، مع وضع الأسس لتطبيقات المصانع الذكية. يُظهر النظام إنتاجية محسّنة، وقدرات صيانة تنبؤية، واندماجاً سلساً في تطبيقات التصنيع، وإدارة الطاقة، والروبوتات.

الكلمات المفتاحية: أجهزة التحكم المنطقية القابلة للبرمجة، الأتمتة الصناعية، الصناعة 0.4، أنظمة المراقبة والتحكم والحصول على البيانات، البرمجة المدعومة بالذكاء الاصطناعي، تكامل إنترنت الأشياء، التصنيع الذكي

Résumé

Ce projet présente le système CA A1 PLC, un automate programmable de nouvelle génération doté d'un traitement bicur, d'une architecture PCB modulaire et d'interfaces de communication complètes incluant USB-C, RS-485 et protocoles sans fil pour les applications Industrie 4.0. Le CA A1 s'intègre à un écosystème intelligent comprenant le CA IDE, un environnement de programmation enrichi par l'IA. La plateforme CA SCADA fournit une supervision de niveau entreprise avec visualisation en temps réel. Le CA Register Monitor permet un diagnostic granulaire en temps réel et l'optimisation du système grâce à la surveillance détaillée des registres de l'automate. Ensemble, ces composants créent une solution d'automatisation unifiée qui répond aux défis industriels actuels tout en établissant les fondations pour les implémentations d'usines intelligentes. Le système démontre une productivité améliorée, des capacités de maintenance prédictive et une intégration transparente dans les applications de fabrication, gestion énergétique et robotique.

Mots clés: API, Automatisation Industrielle, Industrie 4.0, Systèmes SCADA, Programmation Assistée par IA, Intégration IoT, Fabrication Intelligemnte

Abstract

This project presents the CA A1 PLC system, a next-generation programmable logic controller featuring dual-core processing, modular PCB architecture, and comprehensive communication interfaces including USB-C, RS-485, and wireless protocols for Industry 4.0 applications. The CA A1 integrates with an intelligent ecosystem comprising the CA IDE, an AI-enhanced programming environment. The CA SCADA platform provides enterprise-grade supervisory control with real-time visualization. The CA Register Monitor enables granular real-time diagnostics and system optimization through detailed PLC register monitoring. Together, these components create a unified automation solution that addresses current industrial challenges while establishing foundations for smart factory implementations. The system demonstrates enhanced productivity, predictive maintenance capabilities, and seamless integration across manufacturing, energy management, and robotics applications.

Keywords: Programmable Logic Controllers, Industrial Automation, Industry 4.0, SCADA Systems, AI-Enhanced Programming, IoT Integration, Smart Manufacturing

Acknowledgments

I render thanks to Almighty God, the source of all wisdom and strength, who has granted me the patience, perseverance, and health necessary to successfully complete this work. His divine guidance has illuminated my path throughout this academic endeavor.

First and foremost, I would like to express my deepest gratitude to my beloved family, whose unwavering love, sacrifices, and endless support have been the cornerstone of my academic journey. To my parents, who instilled in me the values of perseverance and dedication, and who believed in my dreams and talents. Your faith in me has been my greatest source of strength. To my siblings, whose encouragement and understanding during the countless hours of study and research have meant the world to me. Without your constant presence and emotional support, this achievement would not have been possible.

I extend my heartfelt appreciation to my supervisor, **Mr. El Madjid BERKOUK**, for his exceptional mentorship, invaluable guidance, and unwavering support throughout this research. His expertise, constructive feedback, availability, and genuine care have been instrumental in shaping this work.

I owe a special debt of gratitude to **Mr. BOUSBAI M'hamed** and **Mr. LAHLAH Abdelkarim**, the directors of the ENP Incubator and ENP FabLab respectively. Their exceptional leadership and visionary support have been fundamental to the success of this project. Beyond providing crucial financial backing, they generously offered access to state-of-the-art facilities and resources within both the incubator and FabLab environments. Their commitment to fostering innovation and their belief in this project created the perfect ecosystem for research and development. The dedicated workspace they provided became the birthplace of countless hours of experimentation and discovery. Without their trust, resources, and unwavering support, this project would have remained merely an ambitious idea. Their contribution extends far beyond material support; they have been true enablers of academic and technological advancement.

My sincere gratitude goes to my professors and the distinguished members of the jury **Mr. BOUGHRARA Kamel** and **Mr. BELKACEMI Rabie** for their insightful comments, constructive criticism, and continuous support. Their expertise and commitment to academic rigor have significantly contributed to the refinement of this work.

I would also like to acknowledge all the individuals who have contributed, directly or indirectly, to the realization of this thesis. Their assistance, whether through sharing knowledge, providing resources, or offering moral support, has been invaluable to the completion of this research.

To my cherished friends and colleagues, thank you for your camaraderie, understanding, and encouragement throughout my studies. Your friendship has made this journey not only bearable but enjoyable, and your belief in my abilities has motivated me to strive for excellence.

Finally, I extend my appreciation to everyone who has been part of this academic adventure. Each contribution, no matter how small, has played a meaningful role in bringing this work to fruition.

To all of you, thank you from the bottom of my heart.

CHERBAL Souhaib.

Contents

List of Figures

List of Tables

List of acronyms

General Introduction	16
1 Technical Structure of PLCs	17
1.1 Microcontroller (MCU)	18
1.2 Memory	19
1.3 Memory Access	21
1.4 Digital Input Interface	21
1.5 Digital Output Interface	23
1.5.1 Electrical Isolation	23
1.5.2 Voltage Level Adaptation and Signal Amplification	23
1.5.3 Relay Integration for High-Power Switching	23
1.5.4 Protection Mechanisms	24
1.5.5 Operational Modes and Considerations	24
1.5.6 Conclusion	25
1.6 Analog Input and Output Interface	25
1.7 Power Supply	26
1.7.1 Voltage Conversion and Regulation	26
1.7.2 Isolation and Safety Considerations	26
1.7.3 Efficiency and Thermal Management	26
1.7.4 Reliability and Compliance with Standards	27
1.7.5 Modular and Scalable Designs	27
1.7.6 Conclusion	27
1.8 Communication Interface	27
1.9 Internal/External Buses	28
1.10 Conclusion	29
2 Design of Printed Circuit Boards (PCBs)	30
2.1 Introduction	31
2.2 Printed Circuit Boards (PCBs)	31
2.2.1 Structure of a Printed Circuit Board	31
2.2.2 Concept of PCB Layer Stacking	31
2.2.3 Types of Printed Circuit Boards	34
2.2.4 Single-Layer PCB Design Process	35

2.2.5	Modular PCB Design Process	36
2.2.6	Conclusion	37
3	PLC Design: Multimedia Applications, Electrical Schematics, PCB	38
3.1	Introduction	39
3.2	Communication interface	39
3.2.1	USB	39
3.2.2	USB-C	40
3.2.3	RS-485 Interface	44
3.2.4	UART Communication	48
3.2.5	Power Supply	52
3.3	MCU of the CA A1: ESP32 WROVER E-N8R8 Module Overview	53
3.4	ESP32-WROVER-E-N8R8 Module Characteristics	57
3.5	SIMATIC S7-1200 CPU Characteristics	58
3.6	Feature Comparison	58
3.6.1	ESP32 GPIO Pin Behavior at Boot	59
3.7	3D view of CA A1	60
3.8	Conclusion	61
4	CA IDE: Intelligent PLC Programming Software	63
4.1	Introduction	64
4.2	Software Architecture	64
4.2.1	Core Framework and Design Philosophy	64
4.2.2	Database Structure	65
4.3	Programming Features	65
4.3.1	IEC 61131-3 Language Support	65
4.3.2	AI-Assisted Programming	66
4.3.3	Library Management	66
4.4	Hardware Integration	66
4.4.1	CA A1 Device Configuration	66
4.4.2	Communication Protocols	67
4.5	Debugging and Simulation	67
4.5.1	Integrated Simulation Engine	67
4.5.2	Runtime Debugging	67
4.6	AI Integration	68
4.6.1	Intelligent Documentation Generation	68
4.6.2	LLM-Based Code Generation	68
4.6.3	Future AI Development	68
4.7	Project Management	69
4.7.1	Basic Project Organization	69
4.7.2	Single-User Focus	69
4.8	User Interface and Experience	69
4.8.1	Branded Design	69
4.9	Example of using the CA IDE	70
4.9.1	Opening the Software	70
4.9.2	Creating a New Project	70
4.9.3	Creating a New Program Organization Unit (POU)	71
4.9.4	Creating the Ladder Diagram and Variables Table	72
4.9.5	Creating a ST program	73

4.9.6	Creating a FBD program in CA IDE	74
4.9.7	Creating a SFC program	74
4.9.8	Launching PLC Simulation	74
4.9.9	Monitoring Program Execution	75
4.9.10	Debugging and Fine-tuning	76
4.9.11	Deploying to Hardware	76
4.10	CA Register Monitor	76
4.11	Introduction	76
4.12	Key Functional Roles	77
4.12.1	Connection Management	77
4.12.2	Register Visualization	77
4.12.3	Real-time Monitoring	77
4.12.4	Control Capabilities	78
4.13	Architectural Components	78
4.13.1	Data Management	78
4.14	CA A1 PLC Integration	78
4.15	Key Features	78
4.16	Conclusion	78
4.17	CA SCADA System: Integrated Supervisory Control and Data Acquisition Platform	79
4.17.1	Executive Summary	79
4.17.2	System Overview	79
4.17.3	Core Capabilities	80
4.17.4	Integration Benefits	83
4.17.5	Technical Specifications	85
4.17.6	Deployment Scenarios	86
4.18	Conclusion	87
General Conclusion		88
Bibliography		89
Appendix 1		93
Appendix 2		95
Appendix 3		102
Appendix 4		117

List of Figures

1.1	Structure of a PLC	18
1.2	A digital input interface	22
1.3	A digital output interface	24
1.4	Example of connecting a motor to a PLC digital output	25
1.5	An example of PLC to PLC Ethernet Communication	28
1.6	RS Port	28
1.7	Ethernet Port	28
1.8	RS and Ethernet Ports in a PLC	28
2.1	Structure of Single-Layer PCB	32
2.2	Structure of Multilayer PCB	33
2.3	Rigid PCB	34
2.4	Smart watch - Flexible PCB	34
2.5	Flex-Rigid PCB	35
2.6	Metal Core PCB	35
2.7	PCB Design Flow [1].	36
3.1	USB-C Pinout	40
3.2	Vbus and GND Pins	41
3.3	Data Pins	41
3.4	Super Speed Data Pins	41
3.5	CC Pins	42
3.6	USB Connection	44
3.7	Differences Between RS-232 / RS-422 / RS-485	45
3.8	RS-485 Network	45
3.9	How RS485 works?	46
3.10	2 Wire Half Duplex System	46
3.11	4 Wire Full Duplex System	47
3.12	Simultaneously transmitted and received between devices on RS-485 Protocol	47
3.13	UART Communication between two devices	49
3.14	send the capital letter "S" with UART	51
3.15	Parity bit example	52
3.16	ESP32 WROVER E-N8R8 Module	53
3.17	ESP32 Pin Layout	54
3.18	3D View of the CA A1	60
4.1	the splash image of the CA IDE	70
4.2	Creating new project interface	71
4.3	Interface of choosing the language	72

4.4	creating the ladder diagram	73
4.5	Creating a ST program in CA IDE	73
4.6	FBD program in CA IDE	74
4.7	SFC program for traffic lights	74
4.8	CA IDE Simulator	75
4.9	CA Rigistre Monitor window	77
4.10	CA SCADA station example	79
4.11	CA SCADA Editor	80
4.12	CA SCADA alarm example	82
4.13	Multi-PLC CA SCADA system	84
4.14	EasyEDA main page	103
4.15	Empty workspace	103
4.16	Schematic drawing	105
4.17	OpenPLC interface	118

List of Tables

1.1	Memory Sizes Expressed in Bits, Bytes, Words, and Long Words	19
1.2	Special Memory Byte SMB0 (SM0.0 to SM0.7)	20
1.3	Special Memory Byte SMB1 (SM1.0 to SM1.7)	20
2.1	Advantages and disadvantages of single-layer PCBs [2].	32
2.2	Advantages and disadvantages of multilayer PCBs [2]	33
3.1	Detailed Specifications of RS485	48
3.2	ESP32 Module Pinout	54
3.3	Key Characteristics and Advantages: ESP32-WROVER-E-N8R8 vs. SIMATIC S7-1200 CPU	58
3.4	ESP32 GPIO Pin Behavior at Boot	60
4.1	CA SCADA Technical Specifications	83

List of acronyms

- **AC** : Alternating Current
- **ADC** : Analog-to-Digital Converter
- **AI** : Artificial Intelligence
- **AS-Interface** : Actuator-Sensor Interface
- **BLE** : Bluetooth Low Energy
- **BOM** : Bill of Materials
- **CA A1** : Cherbal Automation's PLC model
- **CA IDE** : Cherbal Automation Integrated Development Environment
- **CAD** : Computer-Aided Design
- **CAN** : Controller Area Network
- **CNC** : Computer Numerical Control
- **CPU** : Central Processing Unit
- **DAC** : Digital-to-Analog Converter
- **DC** : Direct Current
- **DFA** : Design for Assembly
- **DFM** : Design for Manufacturing
- **DFMA** : Design for Manufacturing and Assembly
- **DRC** : Design Rule Check
- **ECAD** : Electronic Computer-Aided Design
- **EEPROM** : Electrically Erasable Programmable Read-Only Memory
- **EMI** : Electromagnetic Interference
- **EOC** : End of Conversion
- **ERC** : Electrical Rule Check
- **ESD** : Electrostatic Discharge

- **FBD** : Function Block Diagram
- **FLASH** : Non-volatile Flash Memory
- **FPC** : Flexible Printed Circuit
- **FR4** : Flame Retardant 4
- **GM** : General Motors
- **GPIO** : General-Purpose Input/Output
- **HDI** : High-Density Interconnect
- **HMI** : Human-Machine Interface
- **HNP** : Host Negotiation Protocol
- **IEC** : International Electrotechnical Commission
- **IEC 61010-1** : Safety standard for electrical equipment
- **IEC 61131-2** : Standard for PLC programming
- **IEC 61131-3** : Standard for PLC programming languages
- **IL** : Instruction List
- **I/O** : Input/Output
- **IoT** : Internet of Things
- **IT** : Information Technology
- **LD** : Ladder Diagram
- **LED** : Light-Emitting Diode
- **LLM** : Large Language Model
- **MCAD** : Mechanical Computer-Aided Design
- **MCPCB** : Metal Core PCB
- **MCU** : Microcontroller Unit
- **ML** : Machine Learning
- **Modbus** : Industrial communication protocol
- **Modicon** : Manufacturer of the first PLC
- **NPN** : Negative-Positive-Negative (transistor type)
- **OT** : Operational Technology
- **PCB** : Printed Circuit Board

-
- **PDN** : Power Distribution Network
 - **PLC** : Programmable Logic Controller
 - **PNP** : Positive-Negative-Positive (transistor type)
 - **PROFINET** : Industrial Ethernet standard
 - **PROM** : Programmable Read-Only Memory
 - **PSRAM** : Pseudo Static RAM
 - **PTC** : Positive Temperature Coefficient
 - **PTH** : Plated Through Hole
 - **RAM** : Random Access Memory
 - **ROM** : Read-Only Memory
 - **RS-232** : Recommended Standard 232
 - **RS-485** : Recommended Standard 485
 - **RTC** : Real-Time Clock
 - **SAR** : Successive Approximation Register
 - **SCADA** : Supervisory Control and Data Acquisition
 - **SFC** : Sequential Function Chart
 - **SMB** : Special Memory Byte
 - **SMT** : Surface Mount Technology
 - **SRP** : Session Request Protocol
 - **ST** : Structured Text
 - **THT** : Through-Hole Technology
 - **TWAI** : Two-Wire Automotive Interface
 - **TVS** : Transient Voltage Suppressor
 - **UART** : Universal Asynchronous Receiver/Transmitter
 - **USB** : Universal Serial Bus
 - **USB-C** : USB Type-C
 - **USB OTG** : USB On-The-Go

General Introduction

In an era defined by Industry 4.0, industrial automation systems are increasingly expected to deliver not only precision and productivity, but also intelligence and adaptability. This final year project, titled *“Design and Development of an Industrial Programmable Logic Controller (PLC) with its Programming Software and SCADA System, Integrating AI, in Compliance with the IEC 61131-3 Standard”*, aims to address this need by developing a next-generation PLC platform.

The core objective of the project is to create a complete automation ecosystem that includes:

- A modular, dual-core industrial PLC called **CA A1**;
- An intelligent, AI-enhanced programming environment (**CA IDE**);
- A full-featured SCADA system (**CA SCADA**) for real-time visualization and supervisory control;
- And a **Register Monitor** tool for granular diagnostics and performance optimization.

Designed to comply with the **IEC 61131-3** industrial standard, this project bridges the gap between traditional automation and emerging technologies such as artificial intelligence and the Internet of Things (IoT). By leveraging modern communication protocols (USB-C, RS-485, wireless) and integrating AI for code generation and predictive maintenance, the system demonstrates the potential for smarter, safer, and more efficient industrial operations.

This project not only meets current industrial challenges but also lays the groundwork for future smart factory applications, making it a pioneering step toward **Industry 5.0**.

Chapter 1

Technical Structure of PLCs

Programmable Logic Controllers (PLCs) are designed to receive signals from sensors and other field devices, process these inputs based on a user-defined control program, and send the appropriate commands to output devices. This streamlined process enables precise, automated control over a wide range of industrial operations [3, 4].

To better understand how a PLC accomplishes these tasks, Figure 1.1 presents a simplified block diagram that highlights the PLC's main constituent modules:

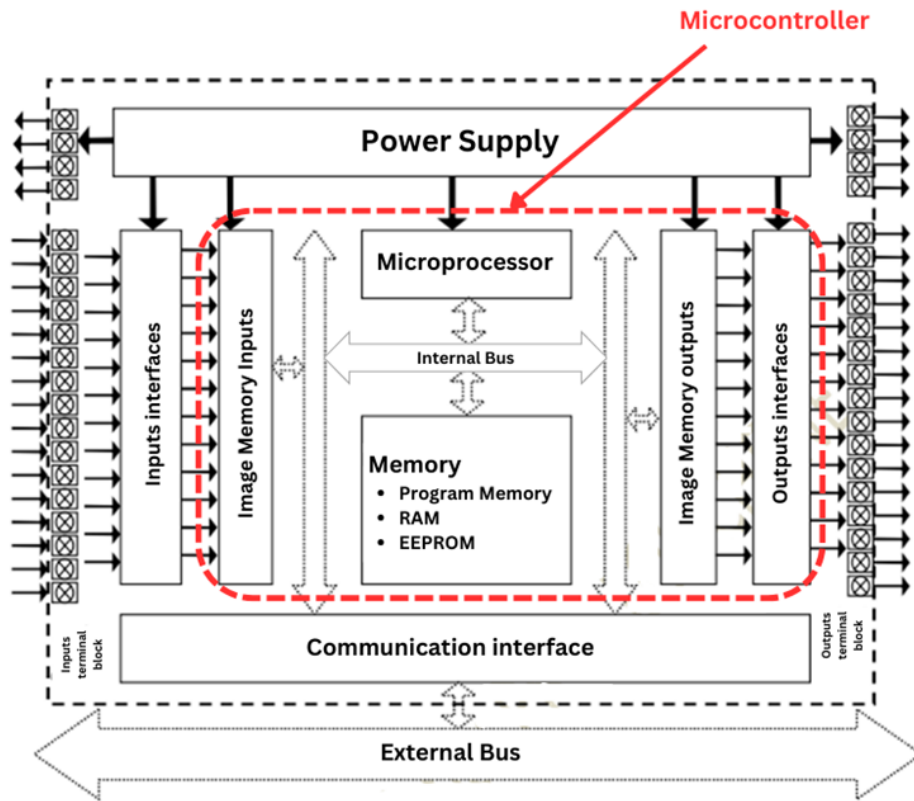


Figure 1.1: Structure of a PLC

1.1 Microcontroller (MCU)

The processing unit or central unit forms the core of the programmable controller [5]. The microprocessor is the main component of this central unit, responsible for processing and decision-making based on interpreting and executing a user-written program.

In general, program execution follows a sequential instruction-by-instruction process, from the first to the last command. For a program to run continuously, the last instruction must be of a type that loops back to the start of the main program. The execution time of each instruction is determined by the number of system clock cycles required.

The operations performed by this unit can be classified into two categories:

- **Logical operations:** AND, OR, XOR, NOT, SHIFT, and ROTATE...
- **Arithmetic operations:** Addition, subtraction, multiplication, and division.

Since PLCs are designed for industrial applications, they are built for continuous operation [4]. As a result, program execution follows a cyclical process, where the last instruction typically restarts the execution cycle from the beginning of the program.

1.2 Memory

The memory section is intended for storing programs or data. In the case of programmable logic controllers, the memory (often RAM) is typically divided into distinct areas within data memory or program memory, used to store all variables necessary for programming and operation. Consequently, the result of any operation is frequently stored in a dedicated register, sometimes referred to as the **NEGATIVE** result register.

A memory divided into registers serves as a real-time "image" of internal inputs and outputs. This approach enables the system to latch or lock the state of inputs or outputs during the execution of a cyclic program. When the microprocessor directly accesses this memory through variables, it is commonly called addressable memory.

Access to this memory is performed by Bit, Byte, Word, or Long Word. This pertains to the memory capacity, which can be expressed in bits, bytes, words, and long words. For instance, "kilo" typically corresponds to 1000, but in binary notation, the closest value is 1024.

Bit	Byte (8 bits)	Word (16 bits)	Long Word (32 bits)
1024	128	64	32
256K	32K	16K	8K
1M	128K	64K	32K

Table 1.1: Memory Sizes Expressed in Bits, Bytes, Words, and Long Words

The memory of a Programmable Logic Controller (PLC) is divided into different types based on its function:

- **ROM or PROM:** Reserved for the operating system. This is the first set of instructions executed when the PLC is powered on.
- **EEPROM (FLASH):** Known as program memory, it stores the user-written program. This program is loaded into the PLC and executed when the RUN button is activated.
- **RAM:** A volatile memory that does not retain its content after a power loss.

Unlike general-purpose microprocessor systems, certain PLC memory sections are specifically dedicated to predefined functions:

- **Flags (Mementos):** A memory section where bits are used as status flags to store intermediate results. These bits can also serve as condition states for logic operations.

- **Special Memory:** Each bit in this memory represents a specific system state during program execution. For example, in Siemens S7-200 PLCs, some special memory bits are assigned predefined roles, as illustrated in Tables 2 and 3.

SM Bit	Description
SM0.0	This bit is always set to 1.
SM0.1	This bit is set to 1 during the first scan cycle. It can be used, for example, to call an initialization subroutine.
SM0.2	This bit is set to 1 for one scan cycle if data is received in the communication buffer.
SM0.3	This bit is set to 1 for one scan cycle if the PLC enters RUN mode immediately after power-up. For instance, it can trigger final setup tasks before normal operation.
SM0.4	This bit is set to 1 for one scan cycle over a 30-second period, providing a delay after powering up the PLC for a gradual startup.
SM0.5	This bit functions as a 0.5-second clock pulse, producing a half-second blinking signal.
SM0.6	This bit functions as a 1-second clock pulse, producing a one-second blinking signal.
SM0.7	This bit indicates the current position of the mode switch (0 = TERM, 1 = RUN). If communication is enabled via programming, the bit toggles to show the PLC is in RUN mode even if the physical switch remains in TERM.

Table 1.2: Special Memory Byte SMB0 (SM0.0 to SM0.7)

Bit (SM1.x)	Description
SM1.0	This bit is set to 1 during certain operations if their result is zero.
SM1.1	This bit is set to 1 during certain operations if the result is negative or an invalid numeric value occurs.
SM1.2	This bit is set to 1 when an arithmetic operation triggers a specific condition (possibly overflow or carry, depending on the operation).
SM1.3	This bit is set to 1 when a division by zero is attempted.
SM1.4	This bit is set to 1 when the "Write to Table" operation causes a table overflow.
SM1.5	This bit is set to 1 when LIFO or FIFO operations attempt to read from an empty table.
SM1.6	This bit is set to 1 when an attempt is made to convert a non-DCB (Decimal Coded Binary) value into a binary value.
SM1.7	This bit is set to 1 when an ASCII value cannot be converted into a valid hexadecimal value.

Table 1.3: Special Memory Byte SMB1 (SM1.0 to SM1.7)

Input Image Memory: During program execution, the PLC first copies the states of the signals from sensors connected to its various inputs into the corresponding bits of the input image memory. The user program then manipulates this input image rather than directly interacting with the physical inputs.

Output Image Memory: The outputs generated during program execution are stored in the corresponding bits of the output image memory. At the end of each program execution cycle, the values saved in the output image memory are transferred through the PLC's output interface to the external terminals.

1.3 Memory Access

Memory management in PLCs is highly flexible, a PLC can access memory by bit, byte, word, or long word. Although this feature can exist in many computing systems, PLCs allow words and long words to overlap in memory.

To illustrate this concept, we can look at the Siemens S7 PLC family, which is widely used in industrial environments. The figures demonstrating memory access are adapted from Siemens' manuals.

- **Byte (octet)** = 8 bits
- **Word (mot)** = 2 bytes
- **Long Word (long mot)** or **Double Word (double mot)** = 2 words

1.4 Digital Input Interface

Digital input interfaces are essential components of Programmable Logic Controllers (PLCs), enabling communication between external industrial sensors or switches and the PLC's internal processing unit [6]. These interfaces must address two primary challenges:

- **Electrical isolation:** Protecting sensitive PLC electronics from voltage spikes, electromagnetic interference (EMI), and ground loops.
- **Voltage level adaptation:** Ensuring compatibility between the PLC's low-voltage logic (e.g., 3.3V) and higher-voltage industrial signals (e.g., 24V DC).

Opto-couplers (or opto-isolators) commonly provide electrical isolation, as illustrated in Figure 1.2. When a 24V DC signal activates the infrared LED within the opto-coupler, the emitted light switches a phototransistor on the PLC side, forming a galvanic isolation barrier [6]. This prevents high-voltage transients or noise from propagating into the PLC's logic circuits. Opto-couplers also offer high noise immunity and meet international safety standards such as IEC 61010-1, which mandates robust voltage isolation in industrial equipment.

Voltage level adaptation involves a combination of passive and active components [7]. Current-limiting resistors (R1) in series with the opto-coupler LED restrict current flow (typically 510 mA) to safe levels, while a voltage divider or Zener diode steps the 24V signal down to the PLC's logic voltage. Additional signal conditioning, such as a Schmitt

trigger, may be incorporated to filter noise and stabilize transitions (debouncing).

As shown in Figure 1.2, a typical digital input interface includes a 24V DC sensor terminal (e.g., a proximity switch), an opto-coupler (e.g., PC817) for isolation, a current-limiting resistor (R1) to protect the LED, and a pull-up resistor (R2) to define a logic high (3.3V) when the phototransistor is off. When the sensor is active, current flows through R1 and the LED, turning on the phototransistor and pulling the output to ground (logic low). When inactive, the phototransistor remains off, and R2 holds the output at 3.3V (logic high). This binary signal is then processed by the PLC's microcontroller.

External wiring depends on whether the PLC input is sinking or sourcing. In sinking configurations, the PLC provides a ground path while the sensor supplies 24V; in sourcing configurations, the PLC supplies 24V and the sensor completes the ground path. Compatibility with sensor types ranging from mechanical relays to solid-state (NPN/PNP) sensors ensures flexibility in industrial systems.

Compliance with IEC 61131-2 guarantees that digital inputs meet standardized electrical characteristics, including ON/OFF thresholds and input impedance [8]. Protection elements, such as transient voltage suppressors (TVS) and RC filters, further improve reliability by clamping high-voltage spikes and mitigating high-frequency noise. Shielded cables, twisted-pair wiring, and careful component selection help manage signal degradation over long runs and control power dissipation in resistors and opto-couplers.

In summary, the digital input interface is a cornerstone of industrial automation, ensuring safe, robust communication between field devices and PLCs. By integrating opto-isolation, voltage adaptation, and noise suppression for 3.3V logic, it meets the demanding electrical and environmental requirements of industrial settings. The design exemplified in Figure 1.2 balances simplicity, reliability, and adherence to international safety standards, enabling seamless operation across a wide range of applications.

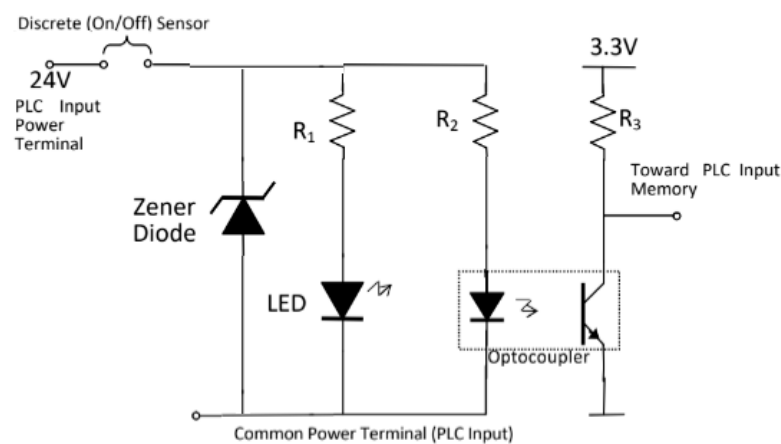


Figure 1.2: A digital input interface

1.5 Digital Output Interface

The digital output interface is a critical component of Programmable Logic Controllers (PLCs), enabling the transmission of control signals from the PLC to external actuators, relays, and industrial devices [7]. This interface ensures safe and reliable communication between the low-voltage PLC logic circuits (e.g., 3.3V or 5V DC) and high-power industrial loads operating at 24V DC or higher. The design of a digital output interface must address key challenges, including electrical isolation, voltage level shifting, and protection against electrical transients.

1.5.1 Electrical Isolation

Electrical isolation is essential in industrial environments to prevent high-voltage transients, electromagnetic interference (EMI), and ground loops from affecting the PLC's internal circuitry [6]. Optical isolation using opto-couplers (also known as opto-isolators) is a common technique to achieve this. As illustrated in Figure 1.3, the opto-coupler consists of an infrared LED and a phototransistor encapsulated in a single package. When the PLC output signal activates the LED, the emitted infrared light switches the phototransistor, enabling current flow in the external circuit without a direct electrical connection. This provides galvanic isolation, enhancing system safety and noise immunity.

1.5.2 Voltage Level Adaptation and Signal Amplification

PLC output circuits typically operate at low voltages (e.g., 3.3V DC or 5V DC), whereas industrial actuators, solenoids, and relays often require 24V DC or more. To bridge this voltage gap, a transistor-based switching stage is introduced. Figure ?? demonstrates a two-stage transistor driver circuit, where:

- The PLC output signal controls a small-signal transistor (T1), which acts as a low-power switch.
- The activation of T1 allows current to flow through the opto-coupler's LED, turning on the phototransistor.
- The phototransistor, in turn, controls a power transistor (T2), which switches the high-power load.

By using this two-stage configuration, the PLC output effectively controls large current loads without exceeding its limited current-driving capability. Additionally, the pull-up resistor (R3) ensures proper biasing of the phototransistor, while R4 and R5 limit the current through the power transistor to prevent excessive dissipation.

1.5.3 Relay Integration for High-Power Switching

For applications requiring the control of high-power devices, such as industrial motors or heavy-duty solenoids, a relay is integrated into the circuit. The relay coil is energized by the switching transistor (T2), allowing the PLC to control high-voltage AC or DC loads through its low-voltage digital output. A flyback diode (D) is placed in parallel with the relay coil to suppress voltage spikes generated by the inductive load when switching off, preventing damage to the circuit components.

1.5.4 Protection Mechanisms

The digital output interface must incorporate protection mechanisms to ensure long-term reliability and operational safety. The following techniques are employed:

1. **Transient Voltage Suppression (TVS):** Protects against voltage surges and spikes induced by inductive loads.
2. **Flyback Diode:** Clamps the back electromotive force (EMF) generated by the relay coil to protect the switching transistor.
3. **Current-Limiting Resistors:** Prevent excessive current draw through the opto-coupler LED and transistors, ensuring stable operation.
4. **Heat Dissipation Management:** Power transistors and relays generate heat; adequate heat sinking or thermal management strategies are necessary for high-current applications.

1.5.5 Operational Modes and Considerations

PLC digital outputs can be configured in different operational modes, including:

- **Sourcing (PNP) Outputs:** The PLC provides a positive voltage to activate the load.
- **Sinking (NPN) Outputs:** The PLC provides a ground path to activate the load.

The selection of sourcing or sinking configuration depends on the external device's compatibility and wiring standards in the industrial setting. Compliance with IEC 61131-2 ensures that digital outputs adhere to standard electrical characteristics, including voltage levels, switching speeds, and output current ratings [8].

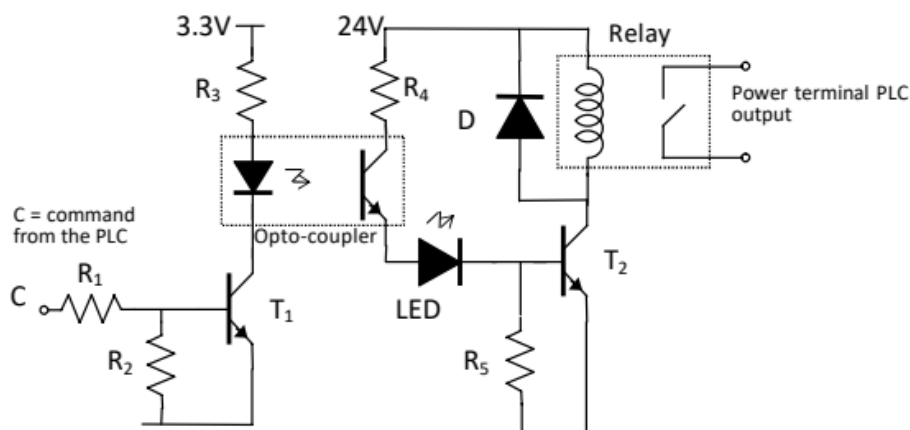


Figure 1.3: A digital output interface

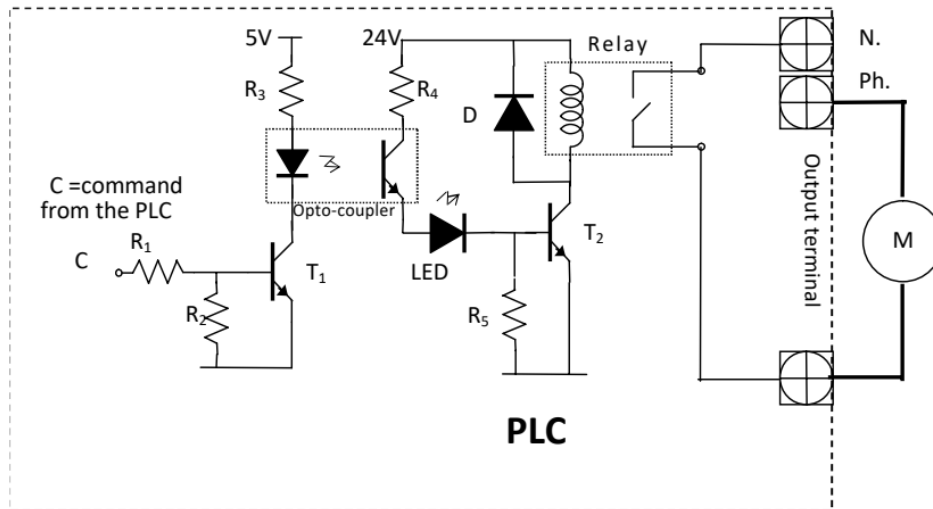


Figure 1.4: Example of connecting a motor to a PLC digital output

1.5.6 Conclusion

The digital output interface is a fundamental component in industrial automation, enabling the PLC to control external devices with precision and safety. By incorporating opto-isolation, transistor switching, and relay integration, the interface ensures compatibility with high-power industrial loads while protecting sensitive PLC electronics. The design exemplified in Figure ?? balances robustness, efficiency, and compliance with international safety standards, making it suitable for a wide range of automation applications, from simple relay control to complex industrial machinery operations.

1.6 Analog Input and Output Interface

In modern control systems, computer-based automation plays a crucial role in replacing human operators. These systems are capable of reading, processing, and making real-time decisions based on the current state of the controlled process. However, most physical parameters in industrial processes such as temperature, pressure, and flow rate are measured by sensors that generate analog electrical signals [9]. Since computer systems operate in a digital environment, capable only of processing binary data (1s and 0s), an essential step in automation is converting these continuous analog signals into discrete digital values.

Once the analog signal has been conditioned and is ready for processing, it must be transformed into a sequence of numerical representations that a digital system can interpret. The electronic circuits responsible for this transformation are known as Analog-to-Digital Converters (ADC). These converters play a fundamental role in bridging the gap between the analog and digital domains, ensuring accurate data representation within digital controllers such as Programmable Logic Controllers (PLCs).

A wide variety of ADCs are available on the market, ranging from basic models to high-performance versions designed for seamless integration with microprocessors. Some advanced ADCs are capable of direct interfacing with digital computing systems, allowing efficient and precise signal conversion.

1.7 Power Supply

Power supplies in Programmable Logic Controllers (PLCs) are essential components that ensure stable and reliable operation in diverse industrial environments [10, 11]. They are responsible for converting and regulating input voltage to meet the specific needs of the PLC's internal circuitry and any connected modules or field devices. Many modern PLCs are designed with integrated 24V DC power supplies, which are particularly popular in industrial settings.

This section details the key aspects of PLC power supplies, including voltage conversion and regulation, isolation and safety, efficiency and thermal management, reliability standards, and modularity.

1.7.1 Voltage Conversion and Regulation

Many PLCs operate with a dedicated 24V DC power supply, which is especially prevalent in industrial settings. However, whether a system uses a direct 24V DC input or requires conversion from AC mains (e.g., 110 or 240V AC) or higher DC voltages, the power supply must convert and regulate the input voltage to a stable level [10].

The conversion process typically involves an AC-to-DC rectifier circuit (when applicable), followed by voltage regulation to maintain a consistent output commonly 24V DC for powering industrial modules, while lower voltages (such as 3.3V or 5V) may be used for internal logic circuits [11]. Filtering elements like capacitors and inductors are also integrated to reduce ripple and suppress electromagnetic interference, ensuring a clean and stable power source [12].

1.7.2 Isolation and Safety Considerations

Electrical isolation is a critical feature of PLC power supplies, as it protects sensitive internal circuitry from voltage surges, electrical noise, and transient disturbances. Isolation is often achieved through the use of transformers, which provide galvanic separation between the input and output circuits in AC-to-DC converters [6].

Additional surge protection devices, such as varistors and transient voltage suppressors (TVS), further safeguard the system by mitigating the impact of voltage spikes [13]. Compliance with international safety standards (e.g., IEC 60950 and IEC 61010) is essential to ensure that the power supply meets rigorous performance, safety, and electromagnetic compatibility requirements [14].

1.7.3 Efficiency and Thermal Management

Efficiency is a vital consideration in the design of PLC power supplies, as high efficiency minimizes energy losses and reduces heat generation a key factor in compact industrial environments. Modern power supplies often employ high-frequency switching regulators, which offer improved efficiency compared to traditional linear regulators [10].

Effective thermal management is also crucial; design strategies may include the use of heat sinks, thermal pads, and forced-air cooling systems to dissipate excess heat and maintain safe operating temperatures [11]. These measures contribute to the long-term reliability of the power supply under continuous load conditions.

1.7.4 Reliability and Compliance with Standards

Given the critical role of PLCs in industrial automation, their power supplies must be robust and reliable. They are engineered to withstand variations in input voltage, transient disturbances, and thermal fluctuations [4].

Compliance with international standards ensures that the power supply consistently performs under harsh industrial conditions, contributing to the overall safety and stability of the automation system [14]. A robust design minimizes downtime and supports the continuous operation of the PLC and connected field devices [3].

1.7.5 Modular and Scalable Designs

Modern PLC systems increasingly utilize modular power supply designs to enhance flexibility and scalability [5]. Such designs allow for easy maintenance, as individual power supply modules can be replaced or upgraded without disrupting the entire system [15].

In critical applications, redundancy can be implemented by incorporating multiple power supplies to ensure continuous operation, even if one unit fails [16]. This modular approach not only simplifies system maintenance but also provides a scalable solution that can adapt to varying power requirements [17].

1.7.6 Conclusion

PLC power supplies are a fundamental element of industrial automation, ensuring that the control system operates reliably and efficiently. Whether operating with a dedicated 24V DC input or converting from other voltage sources, these power supplies are responsible for voltage conversion, regulation, isolation, and thermal management.

Advances in power supply technology particularly in efficiency and modular design continue to improve the performance, safety, and dependability of PLC systems in demanding industrial environments.

1.8 Communication Interface

The communication interface in a PLC is designed to establish robust connections with a wide range of devices and networks, such as Ethernet, RS-232/485, and various industrial fieldbuses [18]. This connectivity is essential for the seamless exchange of data between the PLC and peripheral devices, including sensors, actuators, and other control systems [7]. By supporting multiple communication protocols, the PLC can integrate into heterogeneous industrial environments, enabling interoperability and flexibility in system design [5].

In addition to facilitating local device communication, the communication interface plays a critical role in remote programming and diagnostics [19]. It allows engineers and technicians to access the PLC from remote locations, thereby enabling real-time monitoring, troubleshooting, and configuration adjustments without the need for physical presence on site [20]. This capability not only reduces maintenance costs and downtime but also supports the integration of the PLC into broader automation systems [4].

As a result, the communication interface is a cornerstone in modern industrial automation, ensuring that control systems are both scalable and adaptable to evolving network infrastructures [21].



Figure 1.5: An example of PLC to PLC Ethernet Communication



Figure 1.6: RS Port



Figure 1.7: Ethernet Port

Figure 1.8: RS and Ethernet Ports in a PLC

1.9 Internal/External Buses

Internal bus architectures in PLCs are designed to enable rapid data exchange between the CPU, memory, and I/O modules [22]. This high-speed communication is crucial for real-time processing, ensuring that control commands and sensor data are transmitted efficiently and without delay [23]. By providing a dedicated pathway for data transfer, the internal bus facilitates synchronization and seamless coordination among various PLC components, which is essential for the accurate execution of control algorithms [24].

In modular PLC systems, an external bus often implemented as a backplane serves to connect expansion modules [16]. This arrangement not only enhances the system's scalability by allowing additional I/O or peripheral devices to be easily integrated, but also ensures that synchronization and data integrity are maintained across the entire system [15]. The external bus supports robust and reliable communication between modules, contributing to the overall stability and performance of industrial automation systems [5].

1.10 Conclusion

The structure of a PLC is a finely integrated system composed of several interdependent modules, each critical to its overall performance in industrial automation. At its core, the microcontroller (CPU) executes control programs that continuously process inputs and command outputs, while the memory architecture ensures efficient storage and rapid retrieval of operational data. Digital interfaces both input and output facilitate seamless communication with external sensors and actuators, ensuring signal integrity through techniques like opto-isolation and voltage level adaptation. Additionally, analog-to-digital converters play a pivotal role in bridging the analog and digital domains, converting continuous sensor signals into discrete data that the PLC can process.

The power supply, often integrated as a dedicated 24V DC source in modern PLCs, provides stable and regulated energy essential for uninterrupted operation, while incorporating safety features such as electrical isolation and surge protection. Communication interfaces further enhance system capabilities by enabling remote programming, diagnostics, and integration into broader automation networks. Finally, internal and external buses ensure rapid data exchange and maintain synchronization among the CPU, memory, and I/O modules, which is particularly critical in modular systems.

In summary, the comprehensive design of a PLC encompassing the CPU, memory, I/O interfaces, ADCs, power supplies, and communication networks ensures robust, efficient, and reliable operation in demanding industrial environments. This modular and scalable architecture not only meets current automation needs but also provides a platform for future technological advancements in industrial control systems.

Chapter 2

Design of Printed Circuit Boards (PCBs)

2.1 Introduction

The world of printed circuit board (PCB) design has undergone rapid, continuous, and exciting evolution [25]. Before PCBs were invented, electronic systems were assembled manually by connecting components point-to-point with individual wires on a metal chassis. This method resulted in bulky, expensive, and fragile assemblies that failed to meet a wide range of performance requirements.

After World War II, and following the invention of transistors and printed circuit boards, major technological advancements were achieved. These inventions revolutionized the manufacturing of electronic systems and led to the development of several innovative methods [1].

They paved the way for modern products such as multilayer and multi-board (modular) PCBs, and enabled the mass production of electronic circuits [15].

The objective of this chapter is to delve into the core of PCB design and explore the fundamental principles, methods, steps, and tools used to transform a collection of separate electronic components into a complete and functional system whether as a monolithic unit or a modular structure [16].

2.2 Printed Circuit Boards (PCBs)

Printed circuit boards are fundamental elements of modern electronics, found in almost all electronic devices and adapted for various applications, such as industrial equipment, the medical and aerospace fields, and automotive systems [25]. The purpose of using PCBs is to connect electronic components using conductive tracks. In general, designing a PCB project requires hard and diligent work, along with the necessary knowledge and skills [1].

2.2.1 Structure of a Printed Circuit Board

A printed circuit board generally has a rigid structure containing electrical components with integrated metallic surfaces called traces, and metallic zones representing planes [2]. The components are soldered and connected via metallic pads. PCBs are made from low-conductivity dielectric materials, interspersed with additional metallic and dielectric layers as needed to ensure clean circuit transmission [26]. The most commonly used dielectric is FR4 (Flame Retardant 4), a fire-retardant composite made of woven fiberglass and epoxy [25].

2.2.2 Concept of PCB Layer Stacking

A printed circuit board typically consists of a complex blend of copper and a dielectric substrate [2]. A small amount of resin is incorporated to establish a reliable connection among various types of electronic equipment. In PCBs, the greater the number of successive layers, the higher the functionality, performance, and power capacity [26, 24].

Single-Layer PCB

This type of printed circuit board is produced with a single layer of conductive material (usually copper) on one side, while the other side is used to mount various electronic

components [2]. It consists of a substrate layer, a metallic conductive layer, a protective solder mask, and a silkscreen layer. These are used to produce a complete board consisting of pads, vias, mounting holes, wires, components, connectors, fill zones, and electrical boundaries [26].

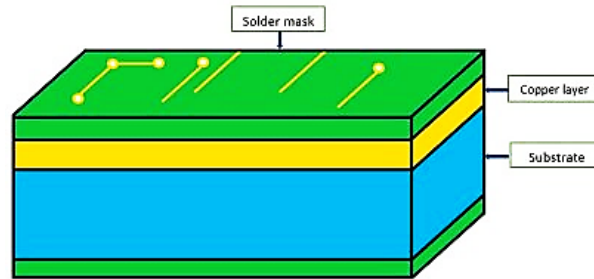


Figure 2.1: Structure of Single-Layer PCB

Single-layer PCBs are used in simple household appliances such as coffee machines, radios, calculators, and basic camera systems. [25].

Advantages	Disadvantages
- Low manufacturing cost. - Faster and easier to design. - Lower risk of errors. - Simpler manufacturing steps. - Better choice for high-volume rapid production. - Cost-effective for mass-market devices. - Ideal for low-density designs.	- Not enough space and connection points for PCBs requiring many components. - Slower speed and lower performance capacity. - Adding functionality results in a larger size and increased weight.

Table 2.1: Advantages and disadvantages of single-layer PCBs [2].

Multilayer PCBs

Multilayer printed circuit boards (PCBs) are created using a process that integrates several layers of insulating material between two conductive surfaces [24]. A thin layer of dielectric material is placed between each pair of layers. To form a single PCB, a lamination process is applied to all layers that make up the circuit board. This is done by exposing the layers to high pressure and temperature [26]. An odd number of layers could cause warping or deformation after soldering, so it is recommended to use an even number of layers [2].

The types of electrical connections between the layers of a multilayer PCB are [1]:

- **Through-hole vias** : Another term used is plated through holes (PTH). This type is used to connect the two external layers.
- **Blind vias** : Used to connect an external layer (either top or bottom) with an internal layer.
- **Buried vias** : Used to connect two internal layers.

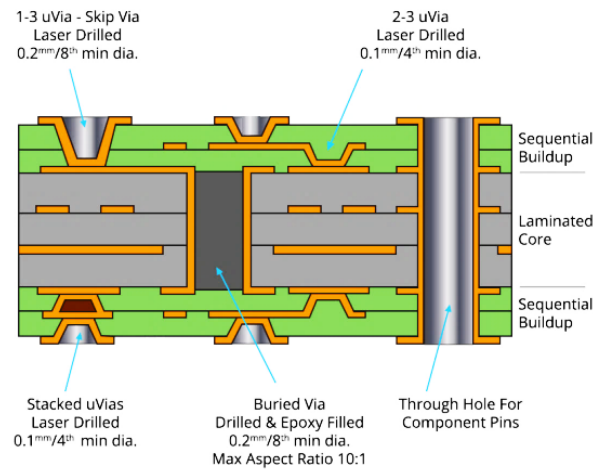


Figure 2.2: Structure of Multilayer PCB

The applications of multilayer PCBs include computers, GPS (Global Positioning System) technology, satellite systems, portable devices, medical equipment such as X-ray machines, heart monitors, signal transmission, and a wide range of other high-tech devices [25, 24].

Advantages	Disadvantages
• Provides more space and additional connections • Makes the board suitable for complex devices requiring additional circuits and components • Withstands delicate handling and lasts longer • Additional density makes multilayer PCBs useful for high-power devices	• Requires additional materials, expertise, and time for fabrication • More expensive than single-layer PCBs • Complex lamination process requirements make multilayer boards take longer to develop • Repair difficulties: internal layers not visible from outside make problem source identification challenging

Table 2.2: Advantages and disadvantages of multilayer PCBs [2]

2.2.3 Types of Printed Circuit Boards

Rigid PCBs

Representing the vast majority of printed circuit boards, the board layout is contained within a rigid substrate [25]. The dielectric material used is FR-4, which can be modified to achieve specific performance improvements in the board [26].

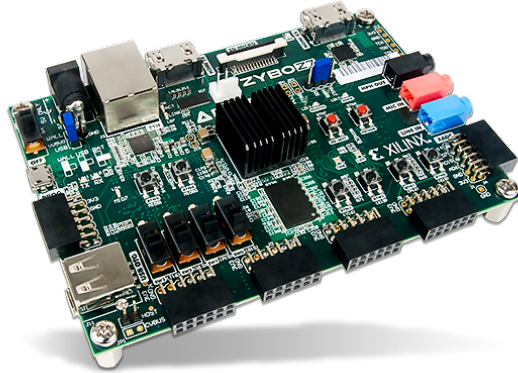


Figure 2.3: Rigid PCB

Flexible PCBs

Composed of a less rigid material allowing for a multilayer circuit made of Rolled Annealed (RA) copper foil and flexible plastic film [25]. This type enables the PCB to conform to specific shapes while maintaining a thinner profile than standard rigid PCBs [26].



Figure 2.4: Smart watch - Flexible PCB

Flex-Rigid PCBs

An innovative design combining properties of two distinct PCB types through a hybrid rigid-flex configuration [25]. Used in applications requiring both structural stability and flexibility [26].

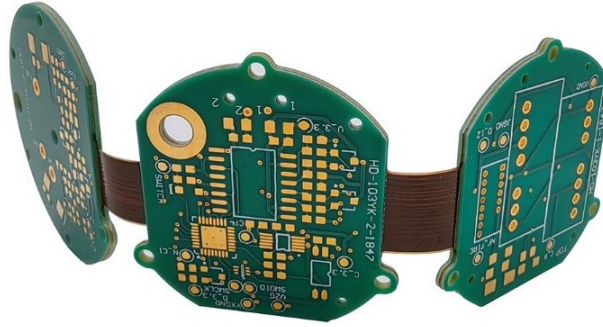


Figure 2.5: Flex-Rigid PCB

Metal Core PCBs

A specialized variant of rigid board designs featuring enhanced thermal conductivity [11]. These incorporate a metal substrate (typically aluminum) to dissipate heat across the board, protecting sensitive circuits. Commonly used in high-current designs requiring thermal management and high-temperature operation [10].

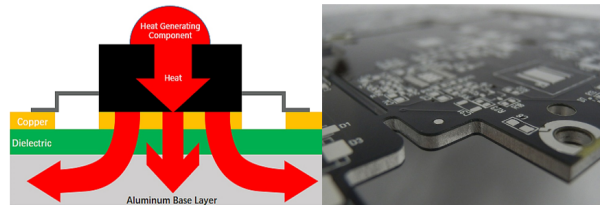


Figure 2.6: Metal Core PCB

2.2.4 Single-Layer PCB Design Process

The PCB design process can be generalized into schematic capture, component placement, routing, fabrication, and assembly, though the details of each step mentioned below are highly complex [1].

- Library development [1]
- Configuration of CAD tools [1]
- Electrical schematic creation [27]

At the end of this stage, an ERC test will be carried out to check for errors in the electrical connections [1].

1. Layout : This is the placement of component footprints while considering the shape and dimensions of the board [28, 29]. All factors that may influence performance must be taken into account, such as areas of excessive heat, connectivity, noise, and electrical or electromagnetic disturbances [30].
2. Routing: This step consists of connecting the component footprints using traces [27]. Several parameters must be considered: the width and thickness of the traces, via diameters, spacing between each pad of the components, etc. [23].

3. Final verification [1]: A DRC test will be applied at this stage as a final check of the design rules. The placement of various texts and markings will also be done; these will be silkscreened onto the outer layers of the PCB.
4. Circuit simulation [1]
5. Generation of design and output data files [1]
6. Manufacturing [14] : This process involves etching the traces and planes onto the metal layers, then compressing them together to obtain a bare board, which will be populated with the components listed in the BOM. It then goes through various soldering processes depending on the component package used [31].
7. DFMA (Design for Manufacturing and Assembly) [14] : DFMA is a design methodology aimed at reducing manufacturing costs and production time [31]. DFMA is based on two distinct methodologies: DFM (Design for Manufacturing) and DFA (Design for Assembly) [14].

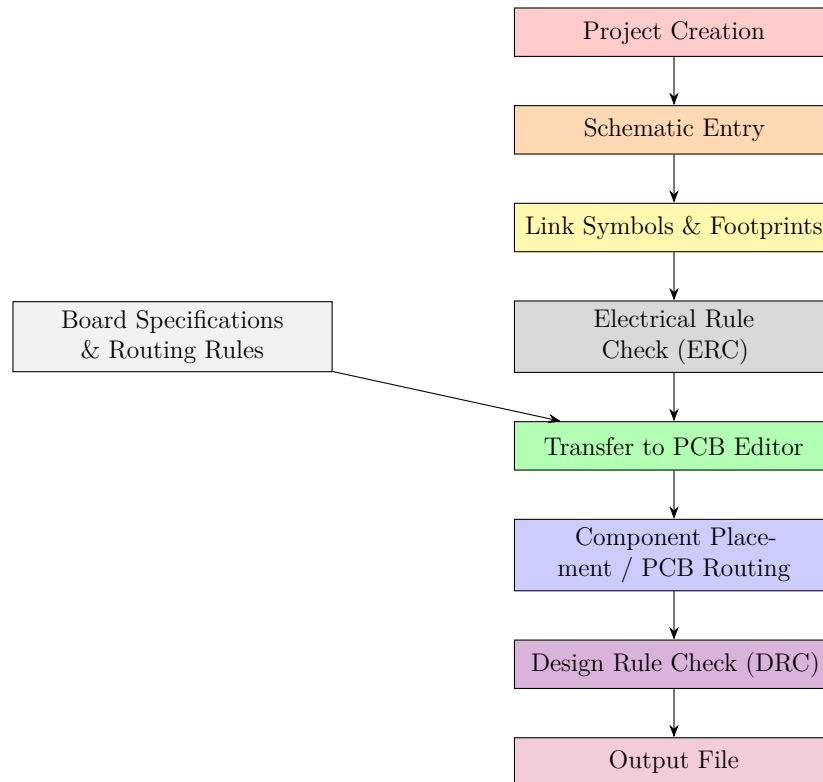


Figure 2.7: PCB Design Flow [1].

2.2.5 Modular PCB Design Process

With current challenges in the supply chain, a modular system can be an interesting option [15]. The concept of modularity aims to divide an entire system into several modules connected via connectors or cables [16]. This trend is exemplified by platforms such as Arduino, Raspberry Pi, and BeagleBoard, which allow independent modules to be connected into complete systems [17].

To achieve a successful modular design, the following points must be considered [16, 17]:

- **Circuit Partitioning** [32]
Before design begins, decide how the system will be divided into modules, define the function of each module, and determine how they will be interconnected.
- **Module Interconnection** [22]
Choose the type of connection between modules. This can be done by stacking modules with boardto board connectors or by linking them with cables.
- **Scalability** [15]
During schematic creation, consider future expandability and applications. For example, when wiring a System on Chip (SoC) with at least 150 pins, most I/O pins remain unused. It is preferable to route these pins to external connectors for future use.
- **Dimensional Accuracy** [16]
Accurate placement and alignment are critical in modular design. Even a few millimeters of misalignment can prevent modules from fitting together or cause the overall assembly to exceed its enclosure. To avoid such issues, use a design tool that supports ECAD/MCAD integration (Electronics ComputerAided Design with Mechanical ComputerAided Design) and verify mechanical constraints with a 3D visualization.

2.2.6 Conclusion

Printedcircuitboard design is both an art and a science, and it is a complex, critical discipline for developing electronic products. A wellexecuted PCB ensures optimal performance, high reliability, and manufacturability. Key prerequisites and best practices include:

- A solid understanding of fundamental electronics principles.
- Familiarity with the basic concepts of PCB design.
- Proficiency in CAD tools and simulation/test software for PCB development.
- Execution of all design stages, from schematic capture through generation of output files.
- Adherence to PCB design rules to ensure signal integrity and prevent failures.
- Compliance with DFMA guidelines to deliver a final product that meets requirements.
- Indepth exploration of core PCB design concepts to inform sound design decisions.

Various design methodologies exist in PCB development. Among these, the modular approach offers enhanced flexibility and efficiency, enabling not only better time management but also improved budget control.

Chapter 3

PLC Design: Multimedia Applications, Electrical Schematics, PCB

3.1 Introduction

3.2 Communication interface

Communication interfaces play a crucial role in multimedia applications by enabling all kinds of transmission operations, connectivity, and audiovisual content sharing. Modern industrial control systems, particularly programmable logic controllers (PLCs), rely heavily on robust communication protocols to ensure reliable data exchange and system integration [5].

3.2.1 USB

USB ports are ubiquitous in modern electronic systems, offering a versatile communication interface between various devices and peripherals. They enable not only data transfer but also the powering of external devices, making them a convenient solution for embedded systems and industrial equipment [33].

In our programmable logic controller (PLC) design, the USB interface has been integrated as a means of communication and configuration. The use of the USB-C protocol provides reversible connectivity, high data transfer rates, and compatibility with a wide range of peripherals [34].

USB to UART/JTAG Communication

In electronic systems, serial communication protocols such as UART (Universal Asynchronous Receiver/Transmitter) and JTAG (Joint Test Action Group) are essential for programming, debugging, and testing printed circuit boards. The JTAG protocol enables, among other features, boundary-scan testing, providing access to the core of components without intrusive intervention [35].

To address these needs, a USB to UART/JTAG interface has been implemented using a USB 2.0 converter such as the FT232H [36]. This component supports USB-to-serial conversion for multiple interface types including UART, JTAG, SPI, and I²C, while integrating a multiprotocol synchronous serial engine (MPSSE). It offers the following features:

- Compatibility with high-speed USB 2.0 (480 Mbps).
- Full internal management of the USB protocol, without requiring external firmware.
- Two independent configurable channels for serial/parallel interfaces (ADBUS[0:7] and BDBUS[0:7]).
- Native support for JTAG (TCK, TDI, TDO, TMS) and UART (TXD, RXD).

This solution enables easy system configuration and debugging through a single USB connection, facilitating both development and maintenance of the PLC [7].

USB OTG (On-The-Go)

USB OTG allows direct communication between two USB devices without the need for a host such as a computer [37]. It introduces the concept of *dual-role devices*, which can alternate between host and peripheral roles depending on the connection context. Two protocols are defined in this standard:

- **HNP (Host Negotiation Protocol):** Enables two OTG devices to dynamically negotiate which will act as the host.
- **SRP (Session Request Protocol):** Allows a device (B-device) to request VBUS power activation from the host (A-device).

Micro-AB, Micro-A, and Micro-B connectors are used to define device roles via the identification (ID) pin. When two OTG devices are connected, the ID pin determines the host based on whether it is grounded (Micro-A) or left floating (Micro-B). The role can also be dynamically switched via the OTG controller.

To ensure this functionality, high-speed USB transceivers such as Microchip’s USB3320 are commonly used [38, 39]. This integrated circuit, compliant with the UTMI+ Low Pin Interface (ULPI) standard, provides a high-performance physical (PHY) solution for host, device, or OTG applications. It meets all electrical requirements and allows flexible integration into various embedded systems.

Thus, the integration of USB into our PLC system ensures ease of interfacing, extended debugging capabilities, and enhanced compatibility with modern equipment, all while optimizing usability through the USB-C connector.

3.2.2 USB-C

USB-C Pinout

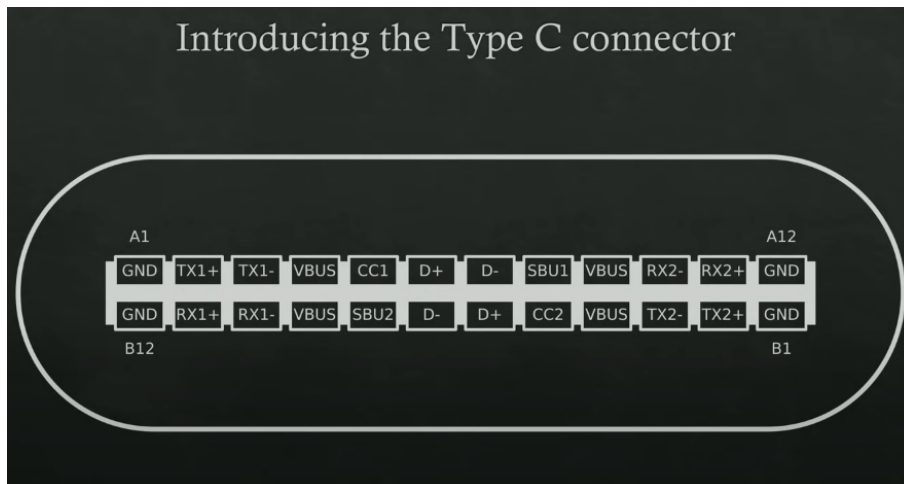


Figure 3.1: USB-C Pinout

Vbus & GND

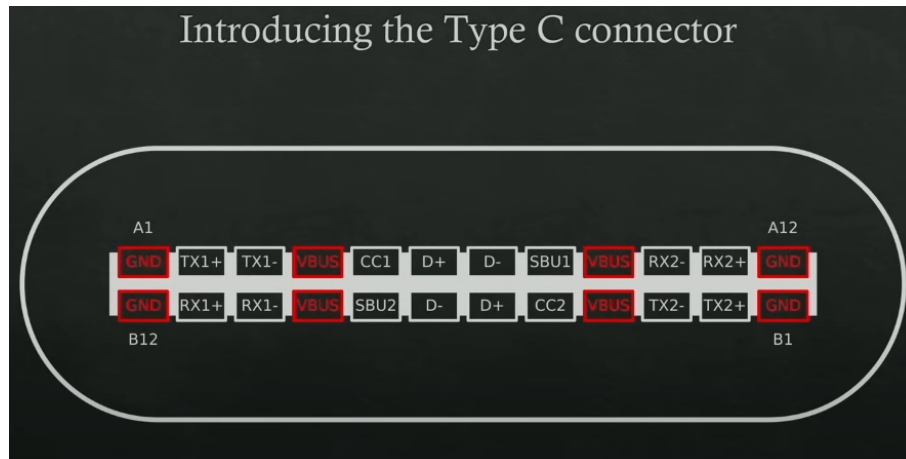


Figure 3.2: Vbus and GND Pins

Data Pins

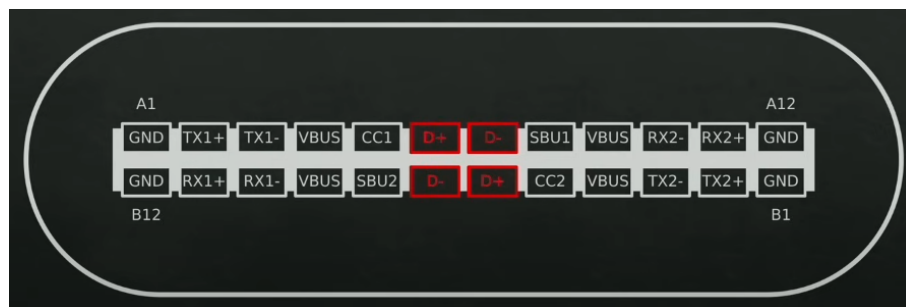


Figure 3.3: Data Pins

Super Speed Data Pins

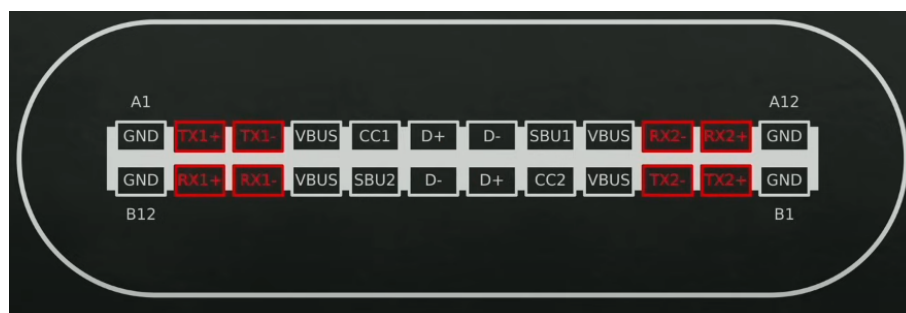


Figure 3.4: Super Speed Data Pins

CC Pins

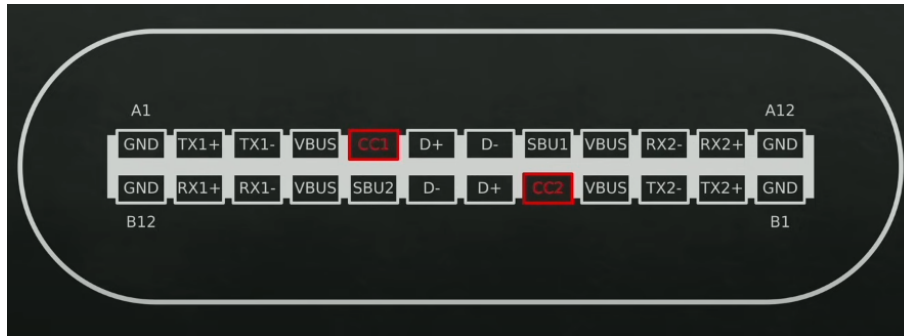


Figure 3.5: CC Pins

Role of CC1 and CC2 in USB-C

The CC1 (Configuration Channel 1) and CC2 (Configuration Channel 2) pins serve multiple purposes according to the USB Type-C specification [33]:

- Detecting the plug orientation
- Determining the role of the device (host or peripheral)
- Power Delivery (PD) communication [40]

Cable Orientation Detection (Reversible Plug)

Unlike previous USB connectors, USB-C is reversible, meaning you can insert the plug either way. However, the USB-C controller needs to know which orientation the plug is inserted in to activate the correct data lines.

Example 1: How orientation is detected:

- The USB host (e.g., a laptop) has two CC pins: CC1 and CC2.
- The USB device (e.g., a phone) also has two CC pins.
- Upon inserting the USB-C cable, only one CC pin (either CC1 or CC2) connects to the host's CC pin.
- The system detects which CC pin is connected and routes the data lines accordingly.

Case 1: Normal Insertion:

- CC1 is connected The host detects this and enables the data lines accordingly.

Case 2: Flipped Insertion:

- CC2 is connected The host detects this and swaps the data lines internally.

This ensures that no matter how you insert the cable, it always works correctly.

Power Role Determination (Host vs Device)

USB-C can dynamically decide which device is the power source and which is the power sink.

Example 2: Charging a Phone from a Laptop

- When a phone is plugged into a laptop using USB-C, the CC pin is used to determine the power role.
- The laptop (usually a power source) pulls the CC line HIGH (via a resistor).
- The phone (usually a power sink) detects this and configures itself to accept power.

If you reverse the connection:

- If a power bank is plugged into the phone, the CC negotiation can make the phone the power receiver and the power bank the source.

USB Power Delivery (PD) Communication

USB-C can deliver different power levels (5V, 9V, 12V, 20V, etc.) based on Power Delivery (PD) communication [40].

Example 3: Fast Charging a Laptop

- A USB-C charger supports multiple voltages (e.g., 5V, 9V, 12V, 20V).
- The charger communicates over the CC pin with the laptop to negotiate the highest supported voltage.
- If the laptop supports 20V fast charging, the charger will supply 20V instead of the default 5V.

Without the CC pin:

- The charger would not know what voltage the laptop needs and would default to only 5V, causing slow charging.

Summary of CC1 and CC2 Functions

Function	Role of CC1/CC2
Detect Plug Orientation	Determines which way the USB-C plug is inserted
Identify Host vs Device	Decides which device provides power (source) and which consumes
Power Delivery Communication	Negotiates higher power (e.g., fast charging for laptops)

USB Connection

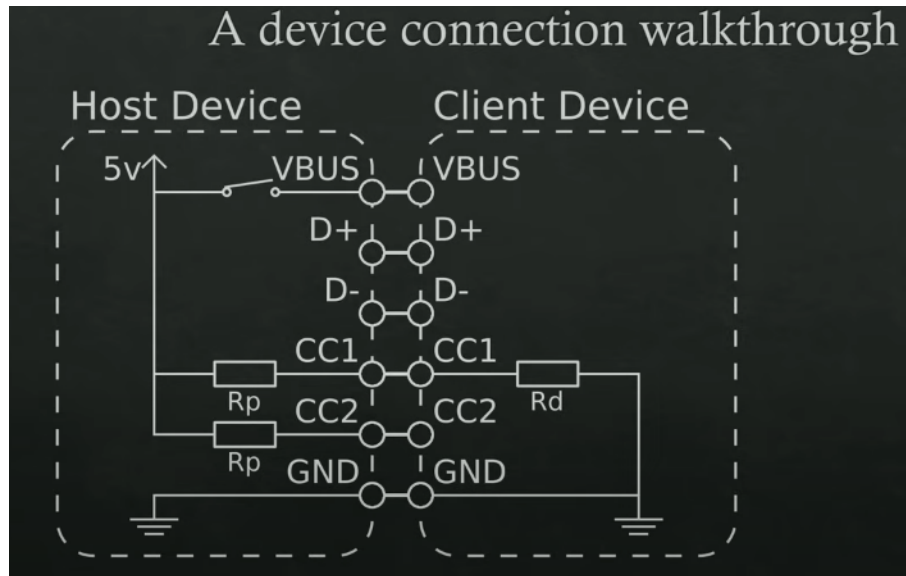


Figure 3.6: USB Connection

3.2.3 RS-485 Interface

RS-485, also known as TIA-485 or EIA-485, is an industrial specification that defines the electrical interface and the physical layer for point-to-point communication between devices [41]. Industrial automation systems extensively utilize RS-485 due to its robustness and multi-point capabilities [18].

It is a multipoint communication standard established by the *Electronics Industry Alliance* (EIA) and the *Telecommunications Industry Association* (TIA). It supports several types of connections, including DB-9 and DB-37.

RS-485 is an update of the serial protocol known as **RS-232**. While the RS-232 standard allowed connection of two devices via a serial link, RS-485 enables serial communication between more than two devices on a networked system [42].

The RS-485 standard defines the electrical characteristics of drivers and receivers for the communication protocol. The Open Systems Interconnection (OSI) model attempts to characterize the different layers of a communication system, from the end application down to the electrical layers, including the physical layer.

An RS-485 compatible network is a multipoint communication network. The RS-485 standard specifies up to 32 drivers and 32 receivers on a single two-wire bus. Since then, new technology has introduced "automatic" repeaters and high-impedance drivers and receivers so that the number of drivers and receivers can be extended to hundreds of nodes on a network. RS-485 transceivers are now even capable of withstanding bus contention issues and bus fault conditions.

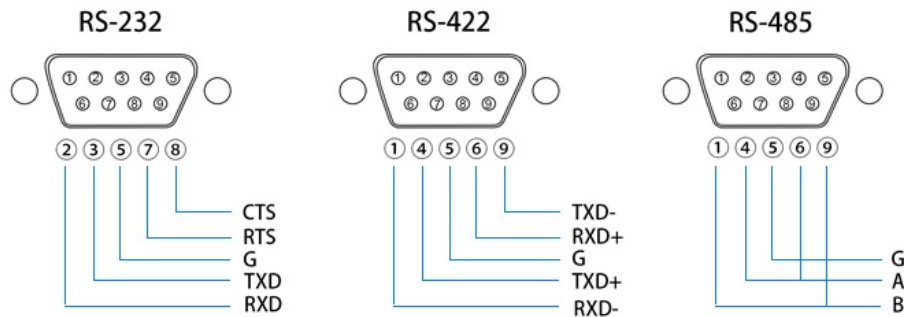


Figure 3.7: Differences Between RS-232 / RS-422 / RS-485

Real-World Applications of RS-485

RS-485 is used in many computer and automation systems. Some examples include robotics, base stations, motor drives, video surveillance, and even home appliances. In computer systems, RS-485 is employed for data transmission between the controller and a disk drive. Commercial aircraft cabins also use RS-485 for low-speed data communications due to the minimal wiring required by its wiring configuration.

RS-485 is, however, most commonly used in programmable logic controllers (PLCs) and factory floors where electrical noise is prevalent [6]. RS-485 serves as the physical layer for many standard and proprietary automation protocols used to implement control systems most notably, **Modbus** [43].

Modbus is the world's most widely adopted automation protocol. Developed by Modicon, Modbus enables integration of devices from different manufacturers into a unified system [44]. Most Modbus implementations use RS-485 because it allows for longer distances, higher transmission speeds, and support for multiple devices on a single network.

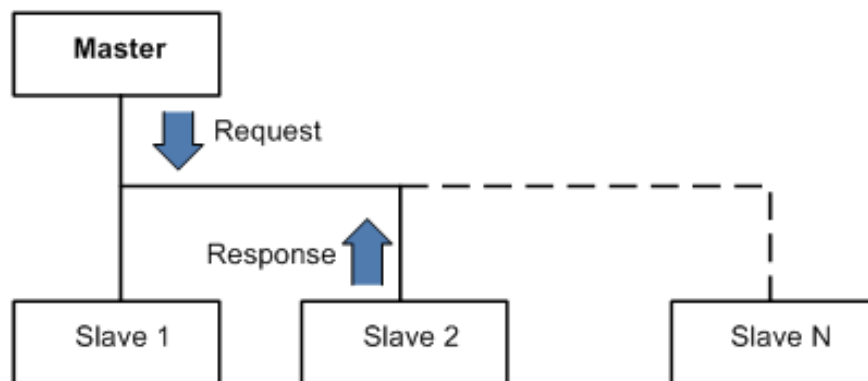


Figure 3.8: RS-485 Network

Modbus devices communicate using a **Master-Slave** technique, where only one device (*The Master*) can initiate transactions (also known as queries). The other devices, called *Slaves*, respond by either providing the requested data to the Master or performing the action specified in the query. This system enables manufacturing facilities to remotely control devices and implement automation.

RS-485 is also widely used in **building automation**, as its simple wiring configuration and support for long cable lengths make it ideal for connecting remote devices. It is commonly used to control video surveillance systems and to connect security control panels with devices such as access control card readers.

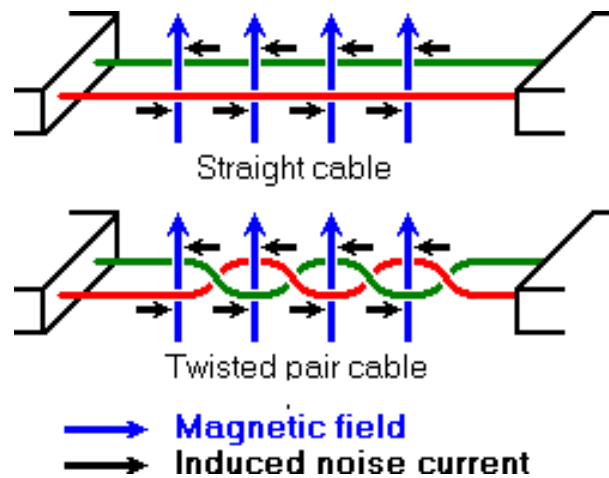


Figure 3.9: How RS485 works?

In the RS-485 standard, data is transmitted via two wires twisted together, commonly referred to as a *Twisted Pair Cable*. The use of twisted pairs in RS-485 provides strong immunity against electrical noise, making RS-485 suitable for electrically noisy environments.

RS-485, at its core with two wires, supports **half-duplex** data transmission. This means data can flow in both directions between devices but only one direction at a time. By adding another two wires, resulting in a four-wire system, **full-duplex** communication becomes possible. In full-duplex mode, data can be transmitted in both directions simultaneously. However, such a configuration is still limited to master-slave communication, where slaves cannot communicate directly with each other.

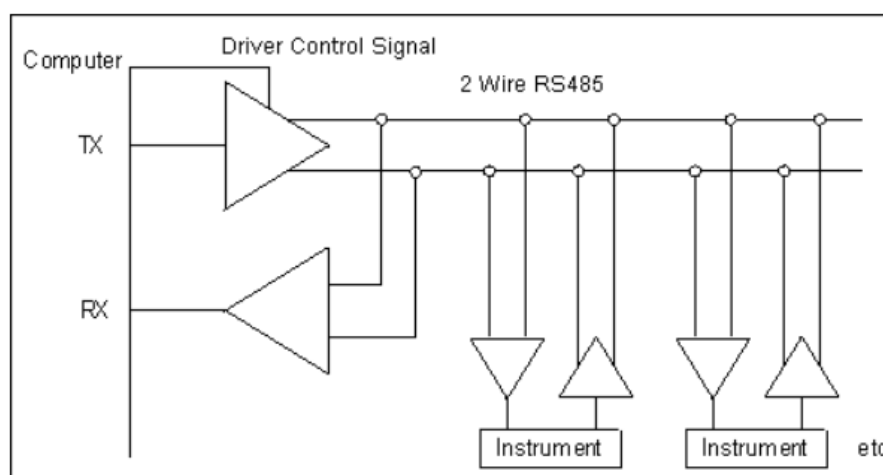


Figure 3.10: 2 Wire Half Duplex System

TX and **RX** refer to the transmitted signal and received signal, respectively. These are often represented by the orientation of the triangles in circuit diagrams.

In this setup, both TX and RX share a single pair of wires. As a result, the communication is limited to **half-duplex** data transmission, meaning data cannot be transmitted simultaneously in both directions between devices.

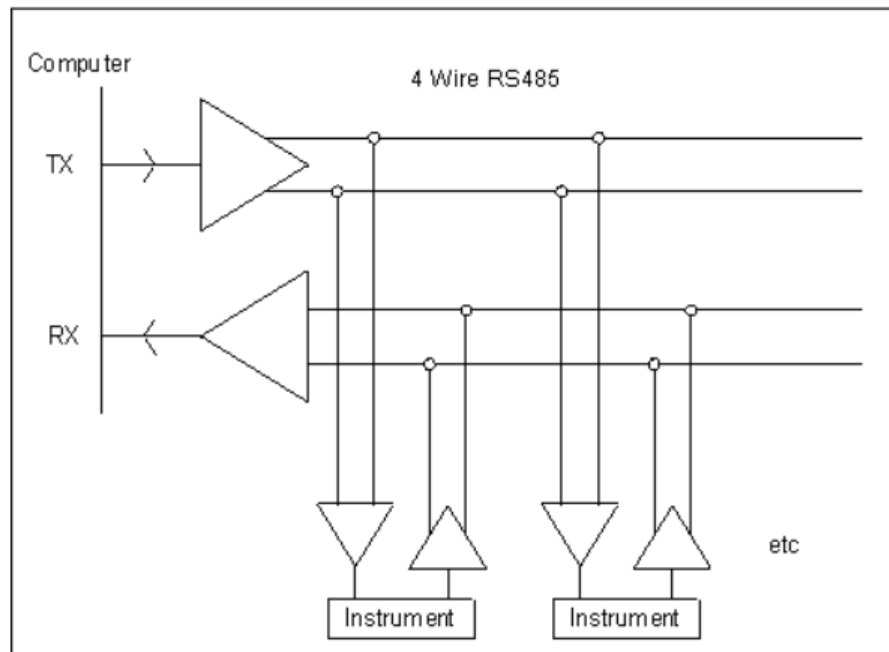


Figure 3.11: 4 Wire Full Duplex System

Similarly, TX and RX refer to the transmitted signal and received signal respectively. It is also represented by the orientation of the triangles.

Variably, this time both TX and RX have their own 2 wires for data transmission. This means that data can be simultaneously transmitted and received between devices.

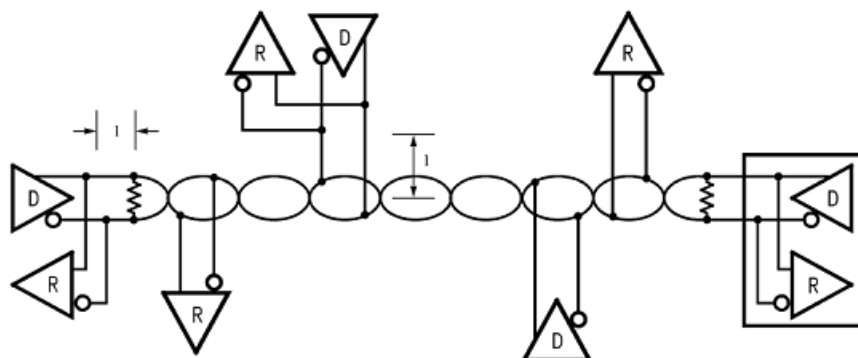


Figure 3.12: Simultaneously transmitted and received between devices on RS-485 Protocol

Devices will be linked together via a bus configuration as shown in the picture above. Here are the detailed specifications of RS485.

Advantages of RS-485

RS-485 has several key advantages compared to other serial communication standards, including tolerance to electrical noise, support for long cable runs, ability to connect multiple slave devices, and fast data transmission speeds [42].

RS485	
Voltage System	Differential
Total Drivers and Receivers on One Line	32 Drivers, 32 Receivers(One Driver active at a time)
Line Configuration	Multidrop
Maximum Operational Distance	1,200M / 3000FT
Maximum Data Transmission Rate	10MBit/s
Duplex Mode	Half Duplex or Full Duplex
Maximum Driver Output Voltage	-7V to +12V
Receiver Input Resistance	12 k Ω
Receiver Input Voltage Range	-7V to +12V
Receiver Sensitivity	$\pm 200\text{mV}$

Table 3.1: Detailed Specifications of RS485

One of RS-485's primary strengths is its robustness in electrically noisy industrial environments. Its design, particularly the use of twisted pair cabling, provides high noise immunity and supports cable lengths of up to **1,200 meters** (approximately **4,000 feet**).

Another major advantage is the support for **multiple slaves** in a single connection up to **32 slave devices** can be connected on one RS-485 bus. This is particularly useful in Supervisory Control and Data Acquisition (SCADA) systems, where many devices need to be monitored and controlled. Additionally, RS-485 systems are cost-effective and easy to implement.

RS-485 also supports data transmission rates of up to **10 Mbit/s**, which is significantly faster than many other serial communication standards.

These advantages have made RS-485 a popular and widely used standard across a variety of industrial and commercial applications [4].

3.2.4 UART Communication

UART stands for **Universal Asynchronous Receiver / Transmitter** and defines a protocols set of rules for exchanging serial data between two devices [35]. UART is very simple and only uses two wires between the transmitter and receiver to allow data transmission and reception in both directions. Both devices must also share a common ground connection.

Communication using UART can operate in three modes:

- **Simplex:** Data is sent in one direction only.
- **Half-duplex:** Each side can transmit, but only one at a time.
- **Full-duplex:** Both sides can transmit simultaneously.

Data in UART is transmitted in the form of *frames*. Each frame includes structured information that defines how the data is packaged for transmission. The format and content of these frames will be briefly described and explained in the following sections.

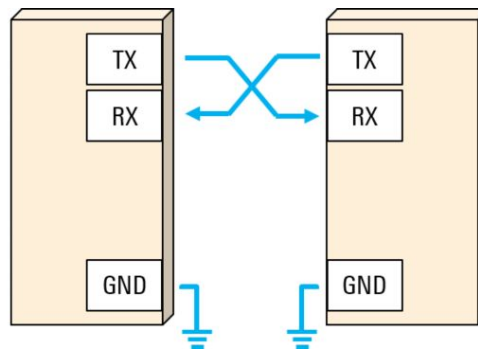


Figure 3.13: UART Communication between two devices

Where is UART Used?

UART was one of the earliest serial communication protocols. The once-ubiquitous serial ports were almost always UART-based. Devices that used RS-232 interfaces, such as external modems, are common examples of UART applications.

In recent years, the popularity of UART has declined. Communication protocols like SPI and I2C are increasingly used for interfacing between chips and components. Additionally, modern computers and peripherals have largely adopted technologies such as Ethernet and USB instead of serial ports.

Despite this decline, UART remains in use for low-speed and low-throughput applications. It is still favored in many embedded systems because it is simple, cost-effective, and easy to implement [45].

Timing and Synchronization in UART Protocols

One of the main advantages of UART is its **asynchronous** nature; the transmitter and receiver do not share a common clock signal. This simplicity, however, introduces timing requirements for both ends of the communication.

Since there is no shared clock, both the transmitter and the receiver must operate at the same, pre-determined baud rate to ensure synchronized communication. Common UART baud rates include:

- 4800 bps
- 9600 bps
- 19.2K bps
- 57.6K bps
- 115.2K bps

In addition to matching baud rates, both devices must also agree on the same frame format and parameters. Understanding this requirement becomes clearer when examining the structure of a UART frame, which will be discussed in the next section.

UART Frame Format

As with most digital systems, a "high" voltage level represents a logical "1" and a "low" voltage level represents a logical "0". Since UART does not define exact voltage levels, the high state is sometimes referred to as *mark* and the low state as *space*. When no data is being transmitted (idle state), the line is held high. This default state allows easy detection of disconnected or malfunctioning transmitters.

Start and Stop Bits

Because UART is asynchronous, the transmitter needs to signal that data bits are coming. This is accomplished by using the start bit. The start bit is a transition from the idle high state to a low state, and immediately followed by user data bits.

After the data bits are finished, the stop bit indicates the end of user data. The stop bit is either a transition back to the high or idle state or remaining at the high state for an additional bit time. A second (optional) stop bit can be configured, usually to give the receiver time to get ready for the next frame, but this is uncommon in practice.

Data Bits

The data bits represent the actual user data. UART frames can include 5 to 9 data bits, but 7 or 8 bits are most commonly used. Data is usually transmitted with the least significant bit (LSB) first.

Example:

The ASCII code for the capital letter 'S' is 0x53, which corresponds to the binary value:

- 7-bit ASCII 'S' = 1 0 1 0 0 1 1 (reading left to right as we normally would)
- Reversed to LSB first: 1 1 0 0 1 0 1 (the order gets flipped for transmission)

This bit sequence will be transmitted after the start bit, and followed by the stop bit(s). The complete UART frame for transmitting 'S' using 7-bit data and 1 stop bit would look like this:

Start Bit (0) 1 1 0 0 1 0 1 Stop Bit (1)

This returns the line to the idle state (high) until the next transmission begins.



Figure 3.14: send the capital letter "S" with UART

There are several practical engineering reasons why UART and many other serial communication protocols use LSB-first transmission:

First, Hardware Simplicity : Shift Register Efficiency: Most microcontrollers and UART hardware use shift registers to handle serial data. When transmitting, it's more natural for the hardware to shift out the LSB first because:

1. Data can be loaded into the shift register as-is.
2. Each clock cycle shifts the register right, naturally exposing the next LSB
3. No bit reversal logic is needed in hardware

Second, Receiver Processing : When receiving data, the receiving device can:

1. Process each bit as it arrives without waiting for the complete byte
2. Build up the value incrementally (bit 0, then bit 1, then bit 2, etc.)
3. Start making decisions about the data before the full byte is received

Parity bit

A UART frame can optionally include a **parity bit** for simple error detection. This bit is inserted between the end of the data bits and the stop bit. Its value depends on the chosen parity scheme:

- **Even parity:** The parity bit is set so that the total number of logical "1"s in the frame (data bits plus parity bit) is even.
- **Odd parity:** The parity bit is set so that the total number of logical "1"s in the frame (data bits plus parity bit) is odd.

Example The 7-bit ASCII code for the capital letter 'S' is 1 0 1 0 0 1 1, which contains four "1"s:

- With even parity, the parity bit is 0, since there are already an even number of "1"s.
- With odd parity, the parity bit is 1, to make the total count of "1"s odd.

Note that a single parity bit can only detect one bit error. If two or more bits flip, the parity check may still pass, so it provides only limited error-detection capability.

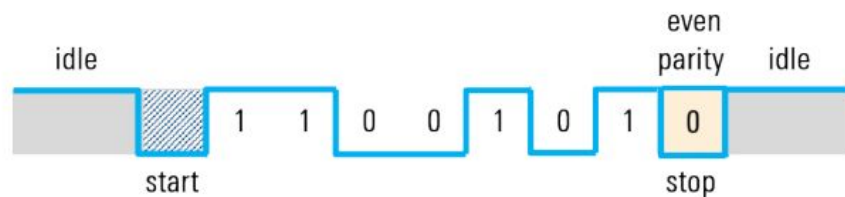


Figure 3.15: Parity bit example

3.2.5 Power Supply

PDN: Power Distribution Network

The design of a Power Distribution Network (PDN) is a critical step in the development of a high-speed digital system [46]. On a Printed Circuit Board (PCB), the PDN refers to the way electrical power is distributed to the components and circuits of the system.

The main objective of a PDN is to provide stable and low-noise power to the integrated circuits, while limiting peak voltage ripple to ensure proper system operation [10, 11]. Voltage disturbances can propagate through the PDN and interfere with other power or signal networks. This can lead to:

- Electromagnetic Interference (EMI),
- Signal Integrity (SI) issues,
- Voltage drop,
- Current fluctuations.

To achieve an efficient power distribution network, the following key elements must be considered [24]:

- Power and ground planes (Stack-Up, Build-Up),

- Grounding,
- Decoupling capacitors,
- Power trace routing (width and length),
- VIAs,
- Isolation of sensitive components,
- Electrostatic Discharge (ESD) events and transient overvoltages,
- PDN protection: ESD protection circuits, eFuse, PTC fuse (Positive Temperature Coefficient), etc.

3.3 MCU of the CA A1: ESP32 WROVER E-N8R8 Module Overview



Figure 3.16: ESP32 WROVER E-N8R8 Module

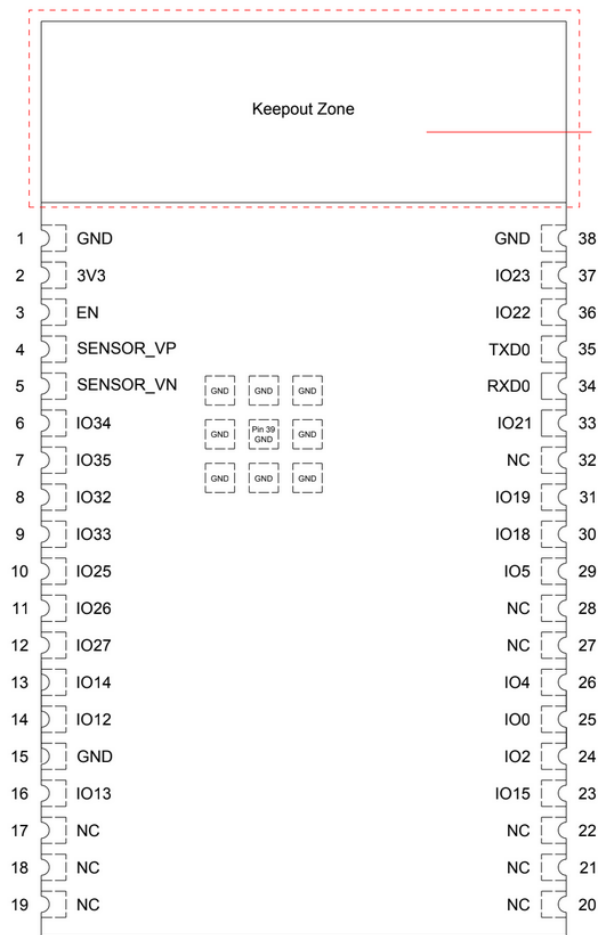


Figure 3.17: ESP32 Pin Layout

P: power supply; I: input; O: output.

Pins GPIO6 to GPIO11 on the ESP32-D0WD-V3/ESP32-D0WDR2-V3 chip are connected to the SPI flash integrated on the module and are not led out.

Table 3.2: ESP32 Module Pinout

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply
EN	3	I	Module-enable signal. Active high.
SENSOR_VP	4	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, ADC1_CH3, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
<i>continued on next page</i>			

Name	No.	Type	Function
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal output), ADC1_CH5, TOUCH8, RTC_GPIO8
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPI_CLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPI_Q, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
NC	17	–	See note
NC	18	–	See note
NC	19	–	See note
NC	20	–	See note
NC	21	–	See note
NC	22	–	See note
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO3, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
IO0	25	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
<i>continued on next page</i>			

Name	No.	Type	Function
IO4	26	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
NC	27	–	
NC	28	–	
IO5	29	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	I/O	GPIO18, VSPICLK, HS1_DATA7
IO19	31	I/O	GPIO19, VSPIQ, UOCTS, EMAC_TXD0
NC	32	–	
IO21	33	I/O	GPIO21, VSPIHD, EMAC_TX_EN
RXD0	34	I/O	GPIO3, U0RXD, CLK_OUT2
TXD0	35	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
IO23	37	I/O	GPIO23, VSPID, HS1_STROBE
GND	38	P	Ground

ADC (Analog-to-Digital Converter)

Hardware component that converts continuous analog signals (like voltage, current, temperature) into discrete digital values that can be processed by the ESP32 micro-controller [9]. The ESP32 features multiple 12-bit ADC channels used for sensing analog inputs in the PLC.

DAC (Digital-to-Analog Converter)

Hardware component that performs the reverse function of an ADC, converting digital values into analog signals. The ESP32 includes two 8-bit DAC channels, used in this project for generating analog output signals.

RTC (Real-Time Clock)

A circuit that maintains accurate time tracking even when the main system power is off. In this project, the ESP32's RTC functionality is leveraged for timestamping events and scheduling time-dependent control operations.

GPIO (General-Purpose Input/Output)

Configurable digital pins that can be programmed to serve either as inputs or outputs. These form the foundation of the digital interfacing capabilities of the PLC.

SPI (Serial Peripheral Interface)

A synchronous serial communication protocol used for short-distance communication in embedded systems. The ESP32 provides multiple hardware SPI interfaces used for connecting to external components like display modules and expanded I/O.

I2C (Inter-Integrated Circuit)

A synchronous, multi-master, multi-slave, packet-switched, single-ended, serial com-

munication bus. In this PLC design, I2C is used for connecting to sensor modules and expansion boards.

UART (Universal Asynchronous Receiver/Transmitter)

Hardware for asynchronous serial communication, used in this project for debugging, programming, and connecting to external serial devices [35].

PWM (Pulse Width Modulation)

A technique for encoding a message into a pulsing signal, widely used for motor control and LED dimming. The ESP32's PWM capabilities are utilized for proportional control outputs.

HSPI/VSPI (Hardware/Virtual SPI)

Dedicated hardware implementations of the SPI protocol on the ESP32, providing high-speed communications with external devices.

EMAC (Ethernet Media Access Controller)

The hardware component that handles Ethernet communications, providing network connectivity for the PLC.

TWAI/CAN (Two-Wire Automotive Interface/Controller Area Network)

A robust vehicle bus standard designed to allow microcontrollers and devices to communicate without a host computer. Implemented for industrial fieldbus communication in this PLC.

PSRAM (Pseudo Static RAM)

A type of dynamic RAM with a self-refresh circuit that behaves similarly to static RAM but with higher densities. The 8MB PSRAM in the ESP32-WROVER-E-N8R8 provides expanded data storage capabilities for the PLC.

BLE (Bluetooth Low Energy)

A wireless personal area network technology designed for applications requiring low power consumption. Used in this project for wireless configuration and monitoring of the PLC.

3.4 ESP32-WROVER-E-N8R8 Module Characteristics

The ESP32-WROVER-E-N8R8 is built around the dual-core Xtensa® 32-bit LX6 micro-processor, each core operating at up to 240 MHz [47, 45]. It incorporates 448 KB of on-chip ROM, 520 KB of SRAM, plus 16 KB of RTC SRAM, and features an external 8 MB Quad-SPI flash alongside 8 MB of PSRAM for high-performance buffering and dynamic data storage. Operating voltage ranges from 3.0 V to 3.6 V and ambient temperatures from -40°C to $+85^{\circ}\text{C}$. Wireless connectivity includes 802.11 b/g/n Wi-Fi with up to 150 Mbps aggregate throughput and Bluetooth 4.2 BR/EDR BLE support. A rich peripheral set (SPI, I2C, UART, SDIO, I2S, PWM, ADC, DAC, capacitive touch, TWAI/CAN, Ethernet MAC) is exposed on up to 24 GPIOs, making it ideal for industrial automation tasks requiring both high compute performance and versatile I/O capabilities [5].

3.5 SIMATIC S7-1200 CPU Characteristics

The SIMATIC S7-1200 family comprises compact PLC CPUs with a single processing core optimized for deterministic control tasks [4]. Typical CPU models (e.g. CPU 1214C) offer up to 125 KB of work memory, 2 MB of built-in program memory (expandable to 24 MB via external storage), and execute bitlevel operations in a minimum of 0.1 μ s per instruction [8]. Integrated I/O (e.g. 14 digital inputs and 10 digital outputs), an onboard PROFINET port (10/100 Mbps), and optional communication modules (Profibus, AS-Interface) enable straightforward fieldbus integration. Power is supplied at 24 V DC, and the operating temperature range spans -20°C to $+60^{\circ}\text{C}$, ensuring reliable performance in industrial environments.

3.6 Feature Comparison

Table 3.3: Key Characteristics and Advantages: ESP32-WROVER-E-N8R8 vs. SIMATIC S7-1200 CPU

Characteristic	ESP32-WROVER-E-N8R8	SIMATIC S7-1200 CPU
<i>Processor</i>	Dual-core Xtensa LX6 @ up to 240 MHz	Single core PLC CPU, optimized for logic scans
<i>Volatile Memory</i>	520 KB SRAM + 16 KB RTC SRAM + 8 MB PSRAM	Up to 125 KB work memory
<i>Non-volatile Memory</i>	8 MB Quad-SPI flash	2 MB built-in program memory (expandable to 24 MB)
<i>I/O and Peripherals</i>	Up to 24 GPIOs; SPI, I2C, UART, SDIO, I2S, ADC/-DAC, PWM, touch, CAN, Ethernet MAC	Integrated digital I/O (e.g. 14 DI/10 DO); PROFINET; optional fieldbus modules
<i>Connectivity</i>	Wi-Fi 802.11 b/g/n (150 Mbps), Bluetooth 4.2 BR/EDR & BLE	PROFINET 10/100 Mbps; Profibus, AS-Interface (optional)
<i>Power Supply</i>	3.0 to 3.6V DC	24 V DC
<i>Operating Temp.</i>	-40°C to $+85^{\circ}\text{C}$	-20°C to $+60^{\circ}\text{C}$
<i>Advantages</i>	Highspeed wireless, rich multimedia and sensor interfaces, low-power coprocessor for ULP tasks	Deterministic scan times, integrated safety and machine-level I/O, IEC 61131-3 compliance

3.6.1 ESP32 GPIO Pin Behavior at Boot

The table below summarizes the behavior of each ESP32 GPIO pin at boot. Some GPIOs have specific boot functions or are connected to flash memory or internal peripherals. It's crucial to avoid using pins with special boot behavior or integrated flash connections as general-purpose I/O without proper caution. Use green-labeled (“OK”) pins for general-purpose input/output whenever possible.

GPIO	Input	Output	Notes
0	pulled up	OK	outputs PWM signal at boot, must be LOW to enter flashing mode
1	TX pin	OK	debug output at boot
2	OK	OK	connected to on-board LED, must be left floating or LOW to enter flashing mode
3	OK	RX pin	HIGH at boot
4	OK	OK	
5	OK	OK	outputs PWM signal at boot, strapping pin
6	x	x	connected to the integrated SPI flash
7	x	x	connected to the integrated SPI flash
8	x	x	connected to the integrated SPI flash
9	x	x	connected to the integrated SPI flash
10	x	x	connected to the integrated SPI flash
11	x	x	connected to the integrated SPI flash
12	OK	OK	boot fails if pulled high, strapping pin
13	OK	OK	
14	OK	OK	outputs PWM signal at boot
15	OK	OK	outputs PWM signal at boot, strapping pin
16	OK	OK	
17	OK	OK	
18	OK	OK	
19	OK	OK	
21	OK	OK	
22	OK	OK	
23	OK	OK	
25	OK	OK	
26	OK	OK	
27	OK	OK	

32	OK	OK	
33	OK	OK	
34	OK	input only	
35	OK	input only	
36	OK	input only	
39	OK	input only	

Table 3.4: ESP32 GPIO Pin Behavior at Boot

3.7 3D view of CA A1

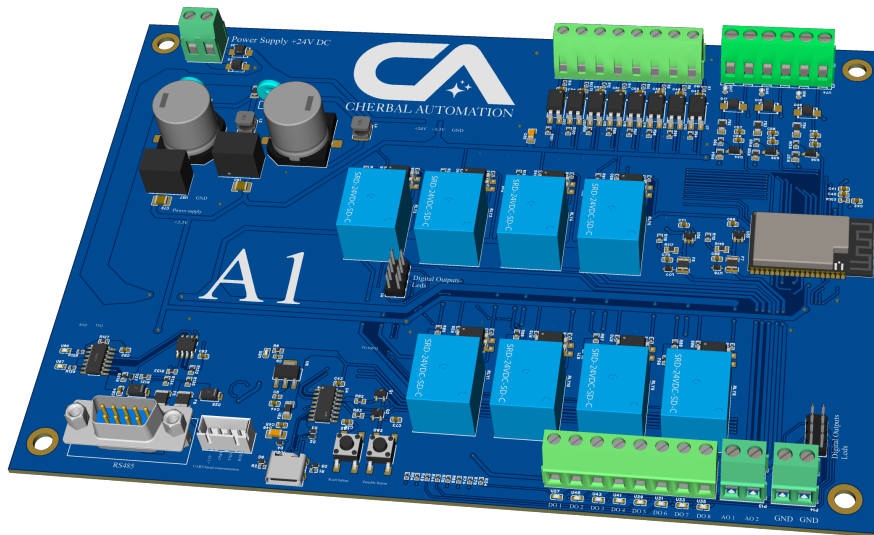


Figure 3.18: 3D View of the CA A1

The Cherbal Automation A1 Programmable Logic Controller represents a significant advancement in compact industrial automation technology, combining robust hardware architecture with versatile connectivity options in a professionally engineered package. The controller's striking blue PCB serves as both functional substrate and visual brand statement, with the prominent "CA" logo and "A1" designation establishing its identity within Cherbal Automation's product ecosystem. The board layout reveals meticulous attention to signal integrity principles, with clear isolation between power domains, analog circuitry, and digital switching sections - a critical consideration in electrically noisy industrial environments where reliable operation is paramount.

The power management section positioned in the upper left quadrant demonstrates thoughtful engineering with its dual electrolytic capacitors providing substantial energy storage and filtering capabilities. This design choice suggests optimization for handling momentary power fluctuations common in industrial settings, while the adjacent power entry terminal block with protective components ensures compliance with electromagnetic compatibility standards. The "+24V DC" labeling indicates adherence to standard industrial voltage levels, facilitating seamless integration with existing factory infrastructure

and sensor networks without requiring additional power conversion stages.

Eight electromechanical relays, distinguished by their bright blue housings and arranged in symmetrical rows of four, constitute the primary output stage of the controller. This relay-based approach provides galvanic isolation between the controller's sensitive electronics and potentially harsh field conditions, while offering versatility in switching both AC and DC loads. The relays are strategically positioned to optimize thermal dissipation during switching operations and minimize electromagnetic interference with adjacent signal processing circuitry. Their prominent placement suggests a design philosophy prioritizing serviceability and visual confirmation of switching states through integrated indicator LEDs.

Communication capabilities are comprehensive, with the RS485 serial interface prominently positioned in the lower left corner providing industrial standard networking capabilities for Modbus RTU or proprietary protocols. Adjacent to this is what appears to be a USB interface for programming and diagnostics, while the wireless module in the lower right corner suggests IoT or local network connectivity options - positioning the A1 as a bridge between traditional industrial control systems and modern networked environments. This multi-protocol approach enables the controller to function effectively in both retrofitted legacy systems and forward-looking Industry 4.0 implementations.

The generous array of screw terminal blocks with clear labeling for digital and analog inputs/outputs demonstrates a user-centric design approach focused on installation simplicity and maintenance accessibility. The terminals' substantial construction suggests compliance with vibration resistance standards essential for industrial applications. Input protection circuitry visible near these connection points indicates defensive design practices against electrical anomalies, enhancing the controller's reliability in unpredictable field conditions. The systematic labeling of I/O points (DO1-DO8 for digital outputs, AI1-AI2 for analog inputs) reflects adherence to programming conventions familiar to automation professionals, reducing commissioning time and configuration errors.

The PCB's four mounting holes reinforced with brass inserts ensure secure installation in control cabinets while providing potential grounding points, reflecting attention to both mechanical stability and electromagnetic compatibility considerations. The overall component density and trace routing patterns suggest a multi-layer board design optimized for manufacturability while maintaining signal integrity at all operating temperatures. This sophisticated engineering approach positions the CA A1 PLC as a versatile control solution capable of addressing diverse automation challenges across manufacturing, process control, building automation, and infrastructure management applications.

3.8 Conclusion

The CA A1 PLC represents a next-generation industrial automation system that successfully integrates advanced communication protocols (USB-C, RS-485, UART) with powerful processing capabilities through the ESP32-WROVER-E module. This system bridges traditional industrial control with modern IoT technology, featuring robust electrical design including isolation barriers, dual-channel power conversion, and comprehensive I/O interfaces. Key achievements include exceptional noise immunity for harsh industrial

environments, versatile connectivity options, and a professional PCB layout optimized for performance and maintainability. The ESP32's dual-core 240 MHz processor with 8MB PSRAM and flash memory enables both traditional control tasks and advanced multimedia applications. The CA A1 establishes a versatile platform ready for Industry 4.0 demands, combining industrial robustness with modern connectivity requirements while positioning Cherbal Automation competitively in the evolving industrial control market.

Chapter 4

CA IDE: Intelligent PLC Programming Software

4.1 Introduction

The development of advanced industrial automation solutions requires not only robust hardware but also sophisticated programming tools that empower engineers to efficiently implement control logic [3]. The Cherbal Automation Integrated Development Environment (CA IDE) represents a significant advancement in PLC programming software, Based on OpenPLC thechnologie and designed specifically to complement the capabilities of the CA A1 PLC system. This purpose-built programming environment combines traditional PLC programming paradigms with modern software engineering practices and cutting-edge artificial intelligence integration, creating a comprehensive solution for industrial automation professionals [5].

CA IDE was developed with a clear focus on accessibility, efficiency, and integration with the CA A1 hardware ecosystem. By providing a unified environment for programming, simulation, deployment, and monitoring, CA IDE streamlines the entire automation development lifecycle [19]. The software supports industry-standard IEC 61131-3 programming languages, ensuring compatibility with established practices while introducing innovative features that accelerate development and reduce engineering overhead [8].

This chapter explores the architecture, features, and capabilities of CA IDE, highlighting its role as an essential component of the Cherbal Automation ecosystem. We will examine how the software's integrated debugging tools collectively enhance programmer productivity [48]. Additionally, we will explore the groundbreaking AI integration that generates comprehensive project documentation and ST code, representing a paradigm shift in industrial automation development tools [49].

4.2 Software Architecture

4.2.1 Core Framework and Design Philosophy

CA IDE is built on a modular, extensible architecture that prioritizes performance, reliability, and user experience [50]. The software is designed for Linux operating systems, providing a consistent experience across various distributions. This architecture is organized into several key layers:

- **User Interface Layer:** Provides an intuitive interface with a distinctive blue theme matching the Cherbal Automation brand identity.
- **Project Management Layer:** Handles project organization and file management.
- **Programming Languages Layer:** Implements the IEC 61131-3 programming languages with syntax highlighting [20].
- **Runtime Layer:** Manages compilation, simulation, and deployment to target hardware.
- **Communication Layer:** Facilitates data exchange with PLC hardware for programming, monitoring, and diagnostics [18].
- **AI Integration Layer:** Provides intelligent assistance through LLM-based algorithms for code generation and electrical cabinet design guidance [51].

The design philosophy behind CA IDE emphasizes accessibility and functionality, particularly for users transitioning from traditional PLC programming methods to more advanced automation development [21]. While maintaining compatibility with established programming methodologies, the software introduces AI-assisted development capabilities that help bridge knowledge gaps and accelerate project implementation.

4.2.2 Database Structure

CA IDE employs a streamlined database structure to maintain project integrity and facilitate essential operations. The database architecture includes:

- **Symbol Database:** Manages variables, data types, and memory allocations.
- **Project Configuration Database:** Stores hardware configurations, communication settings, and I/O mappings.
- **Code Repository:** Maintains program logic for current projects.
- **Documentation Database:** Stores AI-generated documentation and guidance.

This structured approach ensures data consistency across different tools within the IDE, while enabling the core features required for effective PLC programming.

4.3 Programming Features

4.3.1 IEC 61131-3 Language Support

CA IDE provides support for all five IEC 61131-3 standard programming languages, ensuring compatibility with existing programming knowledge and practices [8]:

- **Ladder Diagram (LD):** Graphical programming language resembling electrical relay logic, familiar to electricians and maintenance personnel.
- **Function Block Diagram (FBD):** Graphical language representing data and signal flow through function blocks.
- **Structured Text (ST):** Text-based high-level language similar to Pascal or C for complex algorithms.
- **Instruction List (IL):** Low-level text language similar to assembly programming.
- **Sequential Function Chart (SFC):** Graphical language for sequential control processes based on GRAFCET.

Each language implementation includes specialized editors with features tailored to the specific language paradigm [20]. For graphical languages, the IDE offers intuitive interfaces for creating control logic. Text-based languages benefit from basic editing capabilities including syntax highlighting.

4.3.2 AI-Assisted Programming

A standout feature of CA IDE is its LLM-based AI capabilities that assist with programming tasks:

- **ST Code Generation:** AI-powered generation of Structured Text code based on user requirements and descriptions.
- **Electrical Cabinet Design Guidance:** Intelligent assistance with electrical cabinet layout and wiring for automation projects.
- **Context-Aware Help:** Basic guidance on programming concepts and best practices.

These AI features are designed to make programming more accessible to users with varying levels of experience, providing valuable guidance that accelerates the learning curve and improves project outcomes.

4.3.3 Library Management

The software includes a basic library management system that facilitates code reuse:

- **Built-in Standard Libraries:** Pre-configured function blocks for common operations like PID control, timers, counters, and mathematical operations.
- **User-Defined Functions:** Capability to create and reuse custom function blocks within projects.

The library system provides essential components needed for typical automation projects, allowing developers to leverage pre-built functionality rather than recreating common control elements.

4.4 Hardware Integration

4.4.1 CA A1 Device Configuration

CA IDE features integration with the CA A1 PLC hardware, providing tools for device configuration:

- **I/O Mapping:** Interface for assigning variables to physical I/O points.
- **Parameter Configuration:** Editors for device parameters like communication settings and startup behavior.
- **Hardware Diagnostics:** Basic monitoring of hardware status.

The configuration interface is designed to simplify the process of setting up CA A1 hardware for specific applications, providing the essential tools needed to establish connections between software variables and physical I/O points.

4.4.2 Communication Protocols

The software supports common industrial communication protocols, enabling integration with standard automation components:

- **Modbus RTU/TCP:** Configuration for both serial and Ethernet implementations of this widely used protocol.
- **Custom Serial Protocols:** Framework for implementing simple custom communication schemes.

These protocol support options ensure that CA A1 systems can communicate effectively with other industrial equipment and systems.

4.5 Debugging and Simulation

4.5.1 Integrated Simulation Engine

CA IDE includes a simulation engine that allows testing control logic without physical hardware:

- **I/O Simulation:** Virtual inputs and outputs that can be manually controlled for testing.
- **Example Projects:** Pre-configured examples like LED blinking projects that demonstrate simulation capabilities.
- **Basic Visualization:** Simple interfaces for observing program behavior during simulation.

The simulation environment provides visibility into program execution, helping developers identify and resolve logical errors before deployment to physical hardware.

4.5.2 Runtime Debugging

When connected to physical CA A1 hardware, CA IDE offers basic debugging capabilities:

- **Live Monitoring:** Real-time visualization of variable values.
- **Force Values:** Capability to override I/O and internal variable values for testing specific conditions.
- **Diagnostic Logging:** Basic recording of system events and errors.

These tools provide the essential visibility needed to diagnose issues during commissioning and maintenance of automation systems.

4.6 AI Integration

4.6.1 Intelligent Documentation Generation

One of the innovative features of CA IDE is its AI-powered documentation system:

- **Project Documentation:** AI-assisted creation of basic project documentation.
- **Code Comments:** Suggestions for documenting program logic.
- **Electrical Design Documentation:** Guidance for creating electrical schematics and cabinet layouts.

The documentation capabilities help ensure that projects are properly documented, facilitating maintenance and knowledge transfer throughout the system lifecycle.

4.6.2 LLM-Based Code Generation

The current AI integration focuses primarily on code generation capabilities:

- **ST Code Generation:** Creation of Structured Text programs based on natural language descriptions.
- **Basic Logic Implementation:** Conversion of process requirements into executable control logic.
- **Starter Templates:** Generation of foundational code structures for common automation tasks.

These capabilities significantly accelerate the initial phases of development, particularly for users who may be less experienced with programming but can clearly describe the desired system behavior.

4.6.3 Future AI Development

The CA IDE roadmap includes plans to expand AI capabilities in future versions:

- **Ladder Diagram Generation:** Extending AI capabilities to generate graphical LD programs.
- **Enhanced Code Optimization:** More sophisticated analysis and improvement of user-created logic.
- **Advanced Electrical Design:** More comprehensive assistance with electrical engineering aspects.

These planned enhancements reflect Cherbal Automation's commitment to continuously improving the AI integration within CA IDE to provide even greater value to users.

4.7 Project Management

4.7.1 Basic Project Organization

CA IDE provides essential project management capabilities:

- **Project Templates:** Pre-configured starting points for common automation scenarios.
- **File Organization:** Structured storage of program files, configurations, and documentation.
- **Export/Import:** Tools for backing up and transferring projects.

The project management features are designed to be straightforward and accessible, providing the organizational structure needed without unnecessary complexity.

4.7.2 Single-User Focus

The current version (v1.0) of CA IDE is designed primarily for single-user operation:

- **Local Project Storage:** Projects are stored locally on the developer's machine.
- **Manual Backup:** Users are responsible for creating backups of important projects.
- **Simplified Management:** Streamlined project handling without the complexity of multi-user controls.

This approach simplifies the user experience for individual developers while providing the essential capabilities needed for effective project management.

4.8 User Interface and Experience

4.8.1 Branded Design

CA IDE features a distinctive visual design that reinforces the Cherbal Automation brand identity:

- **Blue Theme:** Consistent blue color scheme matching the Cherbal Automation logo.
- **Clean Layout:** Organized workspace with logical grouping of related tools.
- **Visual Consistency:** Unified design language across all parts of the application.

The visual design not only creates an appealing work environment but also strengthens brand recognition and professional appearance.

4.9 Example of using the CA IDE

4.9.1 Opening the Software

When we click on the CA IDE software icon in the applications gallery, the software starts with the characteristic blue-themed interface reflecting the Cherbal Automation brand identity. The interface presents a clean, professional layout with the main programming area, project navigation panel, and function libraries clearly organized.



Figure 4.1: the splash image of the CA IDE

4.9.2 Creating a New Project

To begin developing a new automation project, we follow these steps :

- Click on "File" in the top menu bar, then select "New" to create a new project
- A project setup dialog appears, prompting for the project name and location
- After configuring the basic project settings, the main development interface is presented with an empty project structure

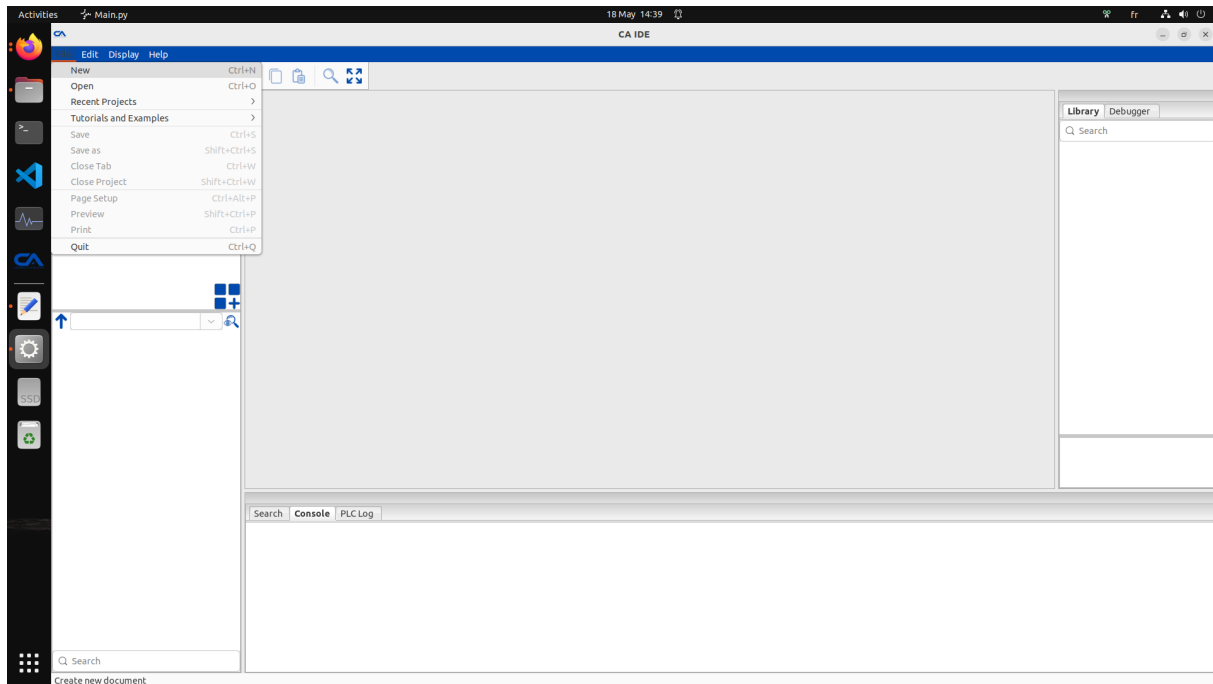


Figure 4.2: Creating new project interface

4.9.3 Creating a New Program Organization Unit (POU)

To add a new program to our project, we:

- Right-click on the project in the navigation tree or use the appropriate toolbar button
- In the dialog that appears (as shown in Image 1), we enter "Blink" as the POU name
- Select "program" as the POU Type
- Choose the programming language from the dropdown menu showing all five IEC 61131-3 languages (IL, ST, LD, FBD, SFC) [8]

The language selection dialog clearly shows CA IDE's support for all standard industrial programming languages, giving developers flexibility to use the most appropriate language for each task.

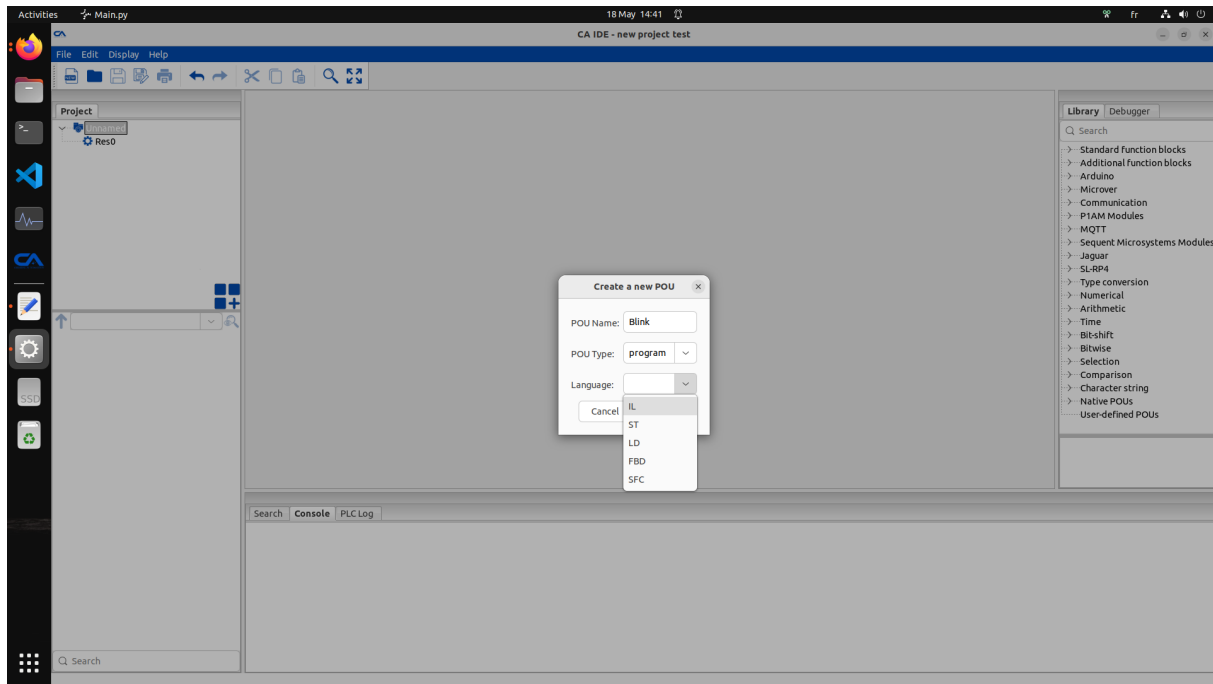


Figure 4.3: Interface of choosing the language

4.9.4 Creating the Ladder Diagram and Variables Table

For our example, we create a simple LED blinking program using Ladder Diagram (LD) language. As shown in Image 2, the development interface provides:

- A variables table at the top where we define the required variables:
 - LED (BOOL type) - mapped to physical address %QX0.0 on the CA A1 PLC
 - TON0 (TON type) - an on-delay timer
 - TOF0 (TOF type) - an off-delay timer
- Documentation field showing "The led is connected to the Digital Output 1 of the CA A1"
- Ladder diagram editor below for creating the control logic

The ladder diagram implements a simple oscillator circuit using two timers:

- The first timer (TON0) activates after a 500ms delay
- The second timer (TOF0) deactivates after another 500ms delay
- This creates a continuous blinking pattern with the LED on for 500ms and off for 500ms

The ladder diagram is drawn with clear graphical elements representing contacts, coils, and function blocks, making the program logic easy to understand visually.

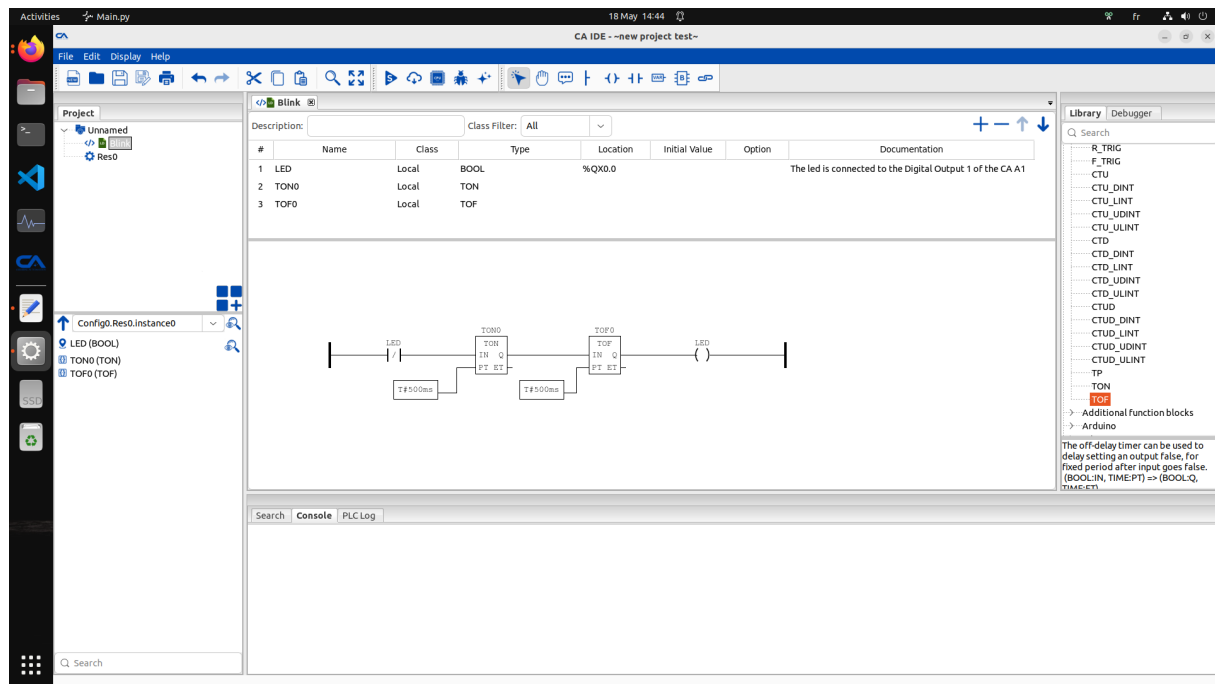


Figure 4.4: creating the ladder diagram

4.9.5 Creating a ST program

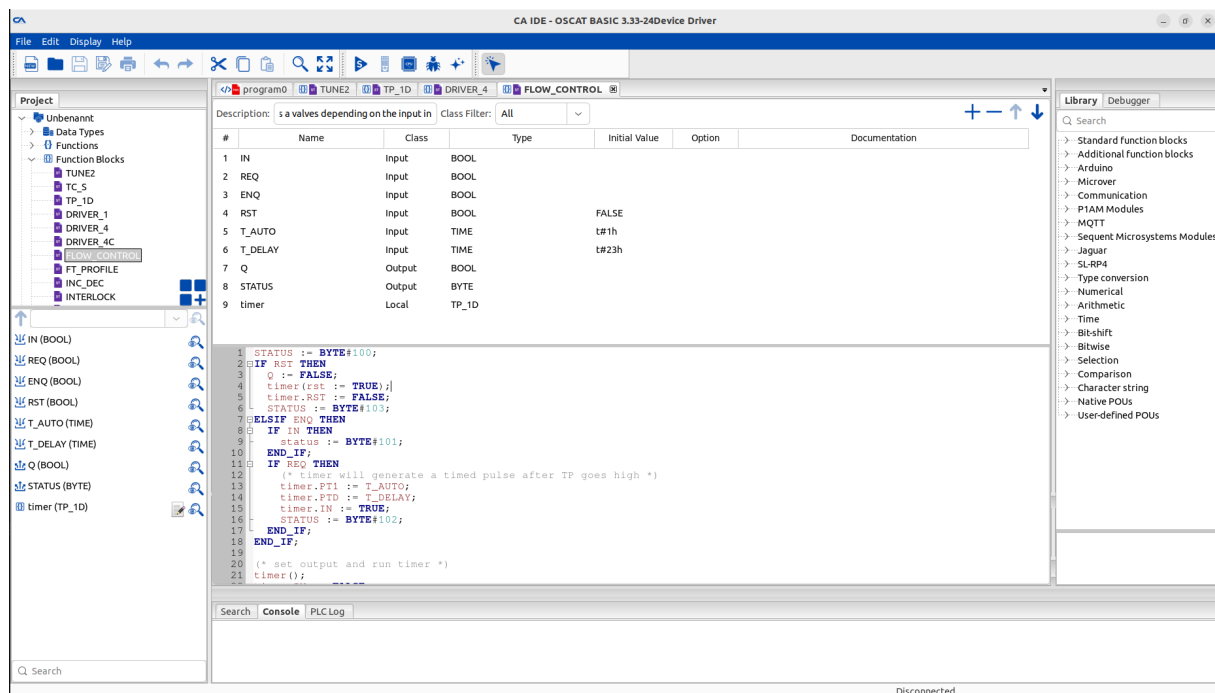


Figure 4.5: Creating a ST program in CA IDE

4.9.6 Creating a FBD program in CA IDE

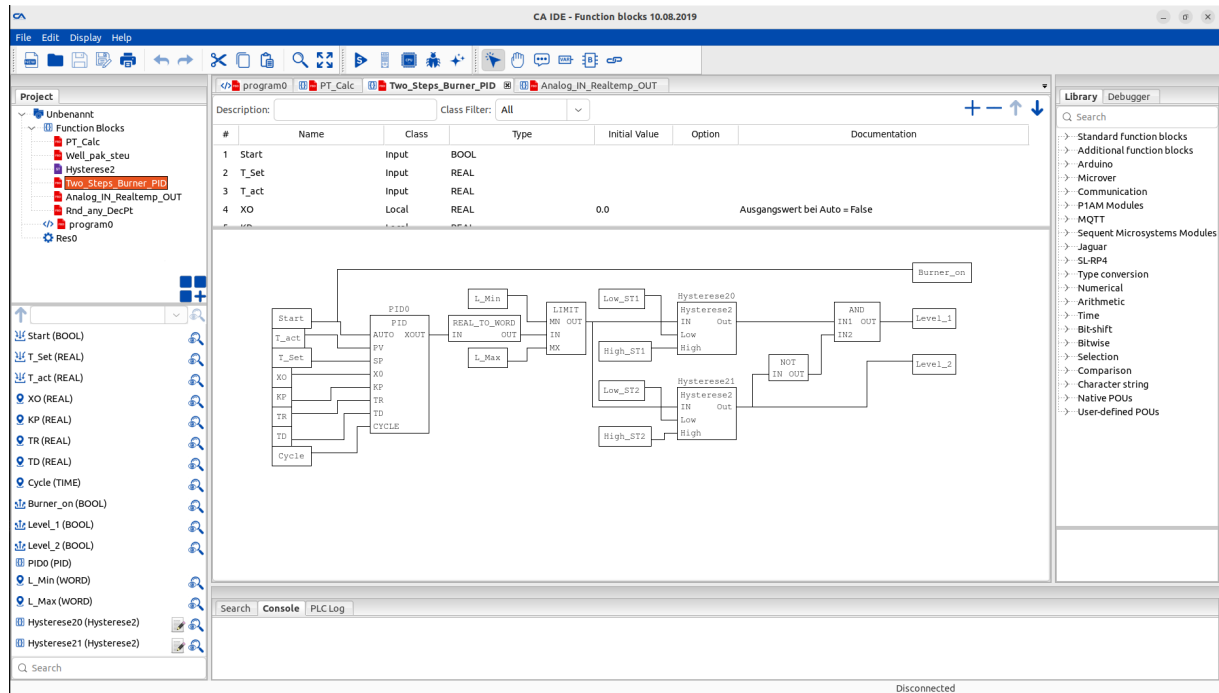


Figure 4.6: FBD program in CA IDE

4.9.7 Creating a SFC program

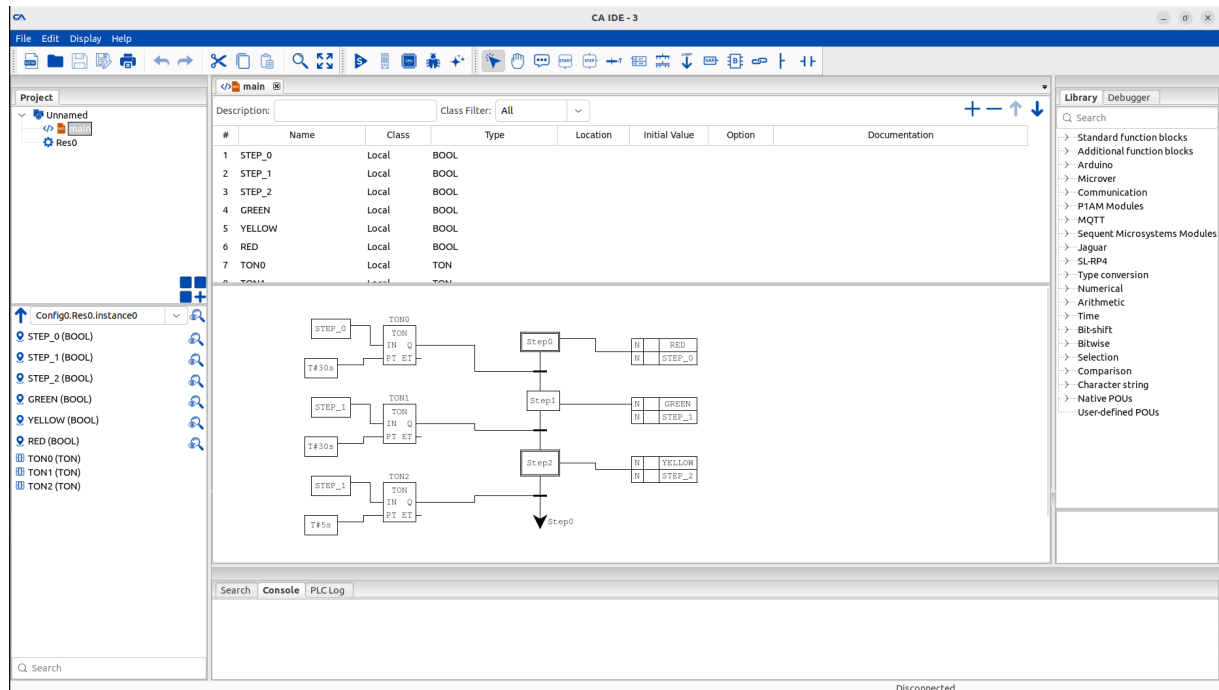


Figure 4.7: SFC program for traffic lights

4.9.8 Launching PLC Simulation

Once the program is complete, we can test it using the built-in simulation capabilities:

- Click on the "Start PLC Simulation" button in the toolbar as shown in Image 3
- The simulator initializes the virtual PLC environment
- The ladder diagram becomes active, showing the current state of the program execution

4.9.9 Monitoring Program Execution

During simulation, CA IDE provides comprehensive monitoring capabilities:

- Active elements in the ladder diagram are highlighted in different colors (green, yellow) showing the current signal flow
- The timer function blocks display their current status visually
- The PLC Log tab at the bottom shows system messages, including "Python extensions started" and "PLC started"
- A real-time graph on the right side displays the LED state over time, showing the regular on/off pattern (as visible in Image 4)

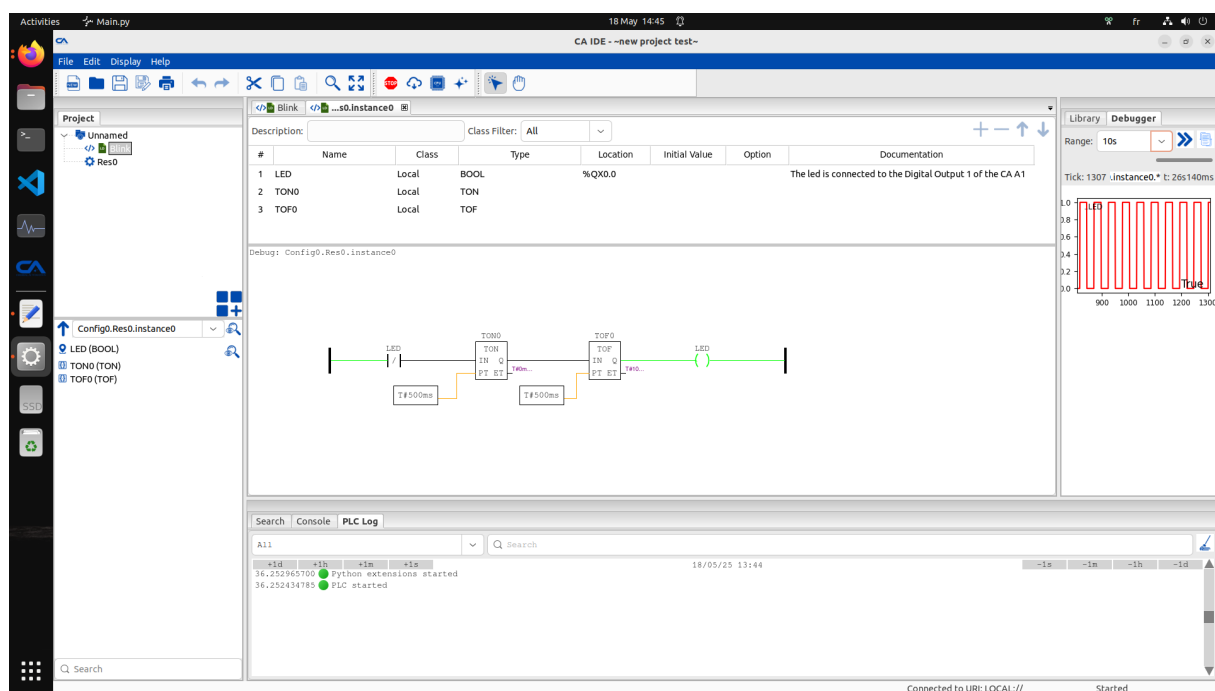


Figure 4.8: CA IDE Simulator

The image above shows the simulation in progress with:

- The LED signal graph showing a regular square wave pattern oscillating between 0 and 1
- The ladder diagram with active signal paths highlighted
- Timer function blocks showing their current timing status

- Monitoring tools providing detailed runtime information, including the current tick count

The simulation interface includes configurable options like the range setting (10s) that controls the timeframe displayed in the monitoring graph. This allows developers to observe the program behavior over different time periods to verify correct operation.

4.9.10 Debugging and Fine-tuning

CA IDE provides several debugging capabilities to help identify and resolve issues :

- Real-time variable monitoring in the variables table
- Signal state visualization in the ladder diagram with clear color coding
- Execution logs in the PLC Log panel
- Timing diagrams showing signal changes over time

The debug information panel (visible in Image 4) shows "Debug: Config0.Res0.instance0", providing clear identification of the current execution context.

4.9.11 Deploying to Hardware

After verifying correct operation in the simulator, the program can be deployed to physical CA A1 hardware :

- Connect to the PLC hardware using the appropriate connection method
- Use the deployment tools to transfer the program to the device
- Monitor the real-world execution using the same visualization tools used during simulation

This seamless transition from simulation to hardware deployment makes the development process efficient and reduces commissioning time.

4.10 CA Register Monitor

4.11 Introduction

The window serves as the central monitoring and control interface for the CA A1 PLC system. It provides real-time visualization and interaction with all CA A1 registers through Modbus communication.

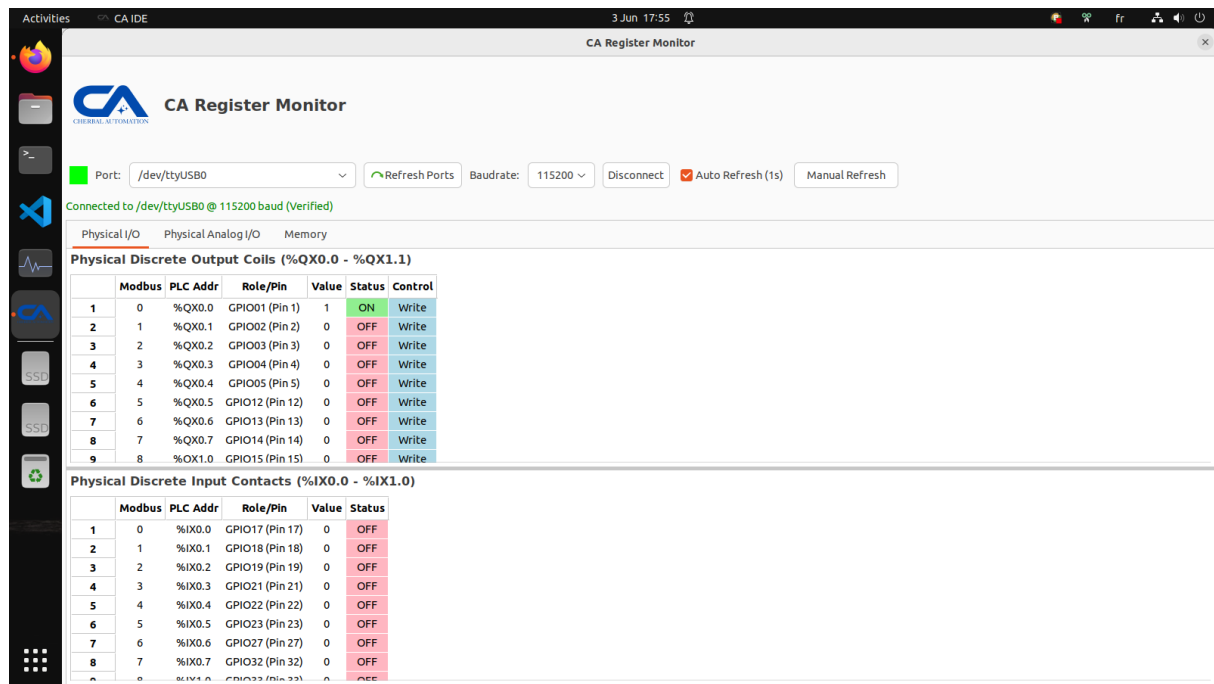


Figure 4.9: CA Rregistre Monitor window

4.12 Key Functional Roles

4.12.1 Connection Management

- Serial port detection and selection
- Modbus connection establishment
- Communication status visualization (LED indicator)
- Auto/manual refresh modes

4.12.2 Register Visualization

The window organizes registers into categorized tabs:

Tab	Content
Physical I/O	Discrete inputs/outputs with pin mapping
Physical Analog I/O	Analog inputs/outputs with pin mapping
Memory	1-bit, 16-bit, 32-bit and 64-bit memory registers

4.12.3 Real-time Monitoring

- Background thread for periodic register polling
- Visual status indicators (color-coded cells)
- Multiple data representations (decimal, hex, binary)

4.12.4 Control Capabilities

- Write access to output registers
- Value editing via double-click interface
- Input validation for write operations

4.13 Architectural Components

4.13.1 Data Management

- Register address mapping to PLC conventions
- Physical pin assignment tracking
- Thread-safe data updates

4.14 CA A1 PLC Integration

The window directly interfaces with the PLC through:

- Modbus RTU protocol implementation
- Register mapping aligned with CA A1 memory architecture
- Physical pin assignments matching CA A1 hardware
- Error handling for communication failures

4.15 Key Features

Feature	Benefit
Multi-tab Interface	Logical register organization
Live Status Coloring	Immediate state recognition
Control Column	Direct register manipulation
Auto-refresh	Continuous monitoring without interaction
LED Indicator	Instant connection status

4.16 Conclusion

The monitoring window serves as the primary human-machine interface (HMI) for the CA A1 PLC system, providing:

1. Real-time visualization of all I/O states
2. Direct control of output registers
3. Diagnostic information for troubleshooting

4. Intuitive access to physical pin mappings

Its design specifically addresses the requirements of industrial automation systems, combining robust communication capabilities with an operator-friendly interface.

4.17 CA SCADA System: Integrated Supervisory Control and Data Acquisition Platform

4.17.1 Executive Summary

The CA SCADA system represents a comprehensive supervisory control and data acquisition solution seamlessly integrated within the CA IDE development environment for the CA A1 programmable logic controller. This integrated approach provides engineers and operators with a unified platform for both PLC programming and real-time process monitoring, eliminating the need for separate SCADA applications and streamlining industrial automation workflows.

4.17.2 System Overview

CA SCADA is a platform that operates as an integral component of the CA IDE software suite. The system provides real-time visualization, data acquisition, alarm management, and control capabilities specifically designed for industrial automation applications utilizing the CA A1 PLC platform.

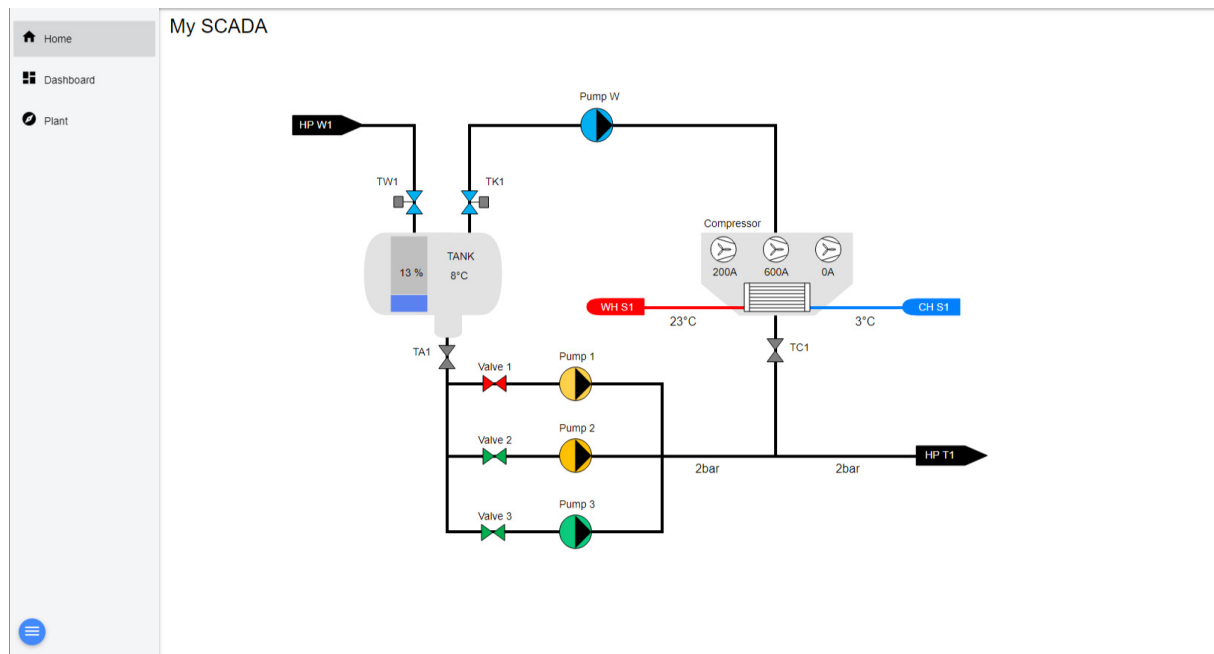


Figure 4.10: CA SCADA station example

Architecture

The CA SCADA system employs a modern, scalable architecture that supports:

- Client-server communication protocols

- Real-time data processing engines
- Distributed data collection mechanisms
- Multi-user concurrent access capabilities
- Cross-platform compatibility through web-based interfaces

4.17.3 Core Capabilities

Real-Time Data Visualization

CA SCADA provides comprehensive visualization capabilities including:

Graphical User Interface Elements:

- Dynamic process graphics with customizable layouts
- Interactive dashboards with real-time data updates
- Trend charts and historical data visualization
- Gauge displays, bar charts, and numerical indicators
- Animated graphics reflecting system status changes

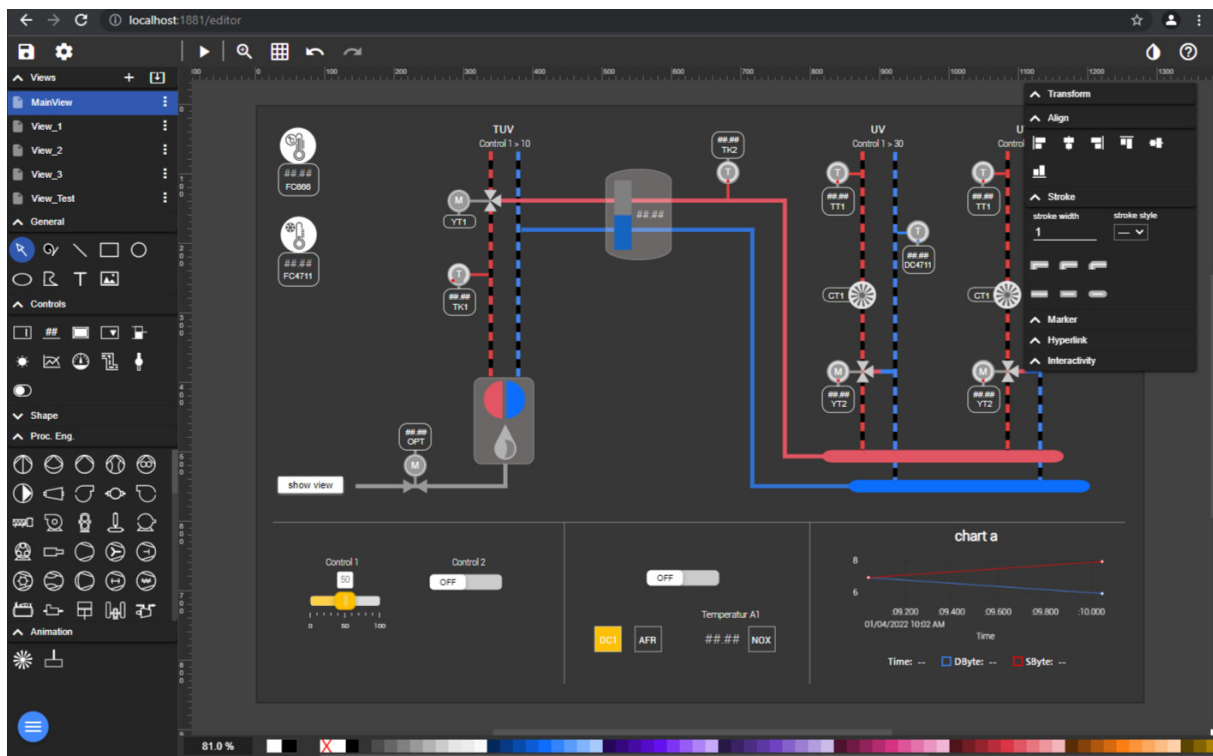


Figure 4.11: CA SCADA Editor

Display Management:

- Multi-screen support for complex process monitoring
- Responsive design adapting to various screen resolutions
- Layer-based graphics organization
- Custom color schemes and branding options

Data Acquisition and Processing

The system implements robust data acquisition mechanisms:

Communication Protocols:

- Native integration with CA A1 PLC communication protocols
- Support for standard industrial protocols (Modbus)
- Configurable polling rates and data sampling intervals
- Automatic reconnection and error handling mechanisms

Data Management:

- Real-time data buffering and caching
- Historical data logging with configurable retention periods
- Data compression and archival capabilities
- Export functionality for data analysis and reporting

Alarm and Event Management

CA SCADA incorporates a comprehensive alarm system featuring:

Alarm Configuration:

- Configurable alarm thresholds and conditions
- Priority-based alarm classification
- Custom alarm messages and descriptions
- Escalation procedures for critical alarms

Notification Systems:

- Visual alarm indicators on process graphics
- Audio alarm notifications
- Email and SMS alert capabilities
- Alarm acknowledgment and logging mechanisms

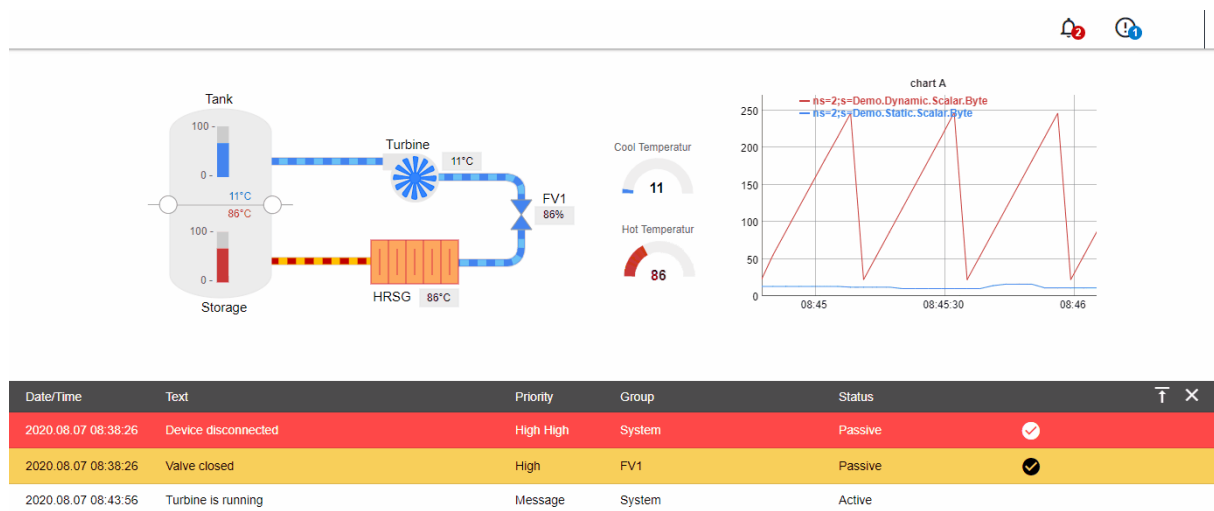


Figure 4.12: CA SCADA alarm example

Control Operations

The platform enables secure control operations through:

Control Interface:

- Interactive control elements (buttons, sliders, input fields)
- Setpoint adjustment capabilities
- Manual/automatic mode switching
- Batch operation controls

Security Features:

- User authentication and role-based access control
- Control permission levels and restrictions
- Audit trails for all control actions
- Secure communication protocols

4.17.4 Integration Benefits

The integration of CA SCADA within the CA IDE environment provides several key advantages:

Development Efficiency

- Unified development environment eliminating context switching
- Shared variable mapping between PLC programs and SCADA displays
- Integrated testing and simulation capabilities
- Streamlined deployment processes

Operational Advantages

- Reduced training requirements for operators familiar with CA IDE
- Consistent user interface across programming and monitoring functions
- Simplified maintenance through single software platform
- Enhanced troubleshooting with integrated diagnostic tools

4.17.5 Technical Specifications

Specification	Details
Platform Compatibility	Linux, Web-based access
Maximum I/O Points	10,000+ tags
Update Rate	100ms minimum
Concurrent Users	Up to 50 simultaneous connections
Historical Storage	Configurable (days to years)
Communication Protocols	Modbus TCP/RTU (Native CA A1)
Database Support	SQLite, MySQL, PostgreSQL

Table 4.1: CA SCADA Technical Specifications

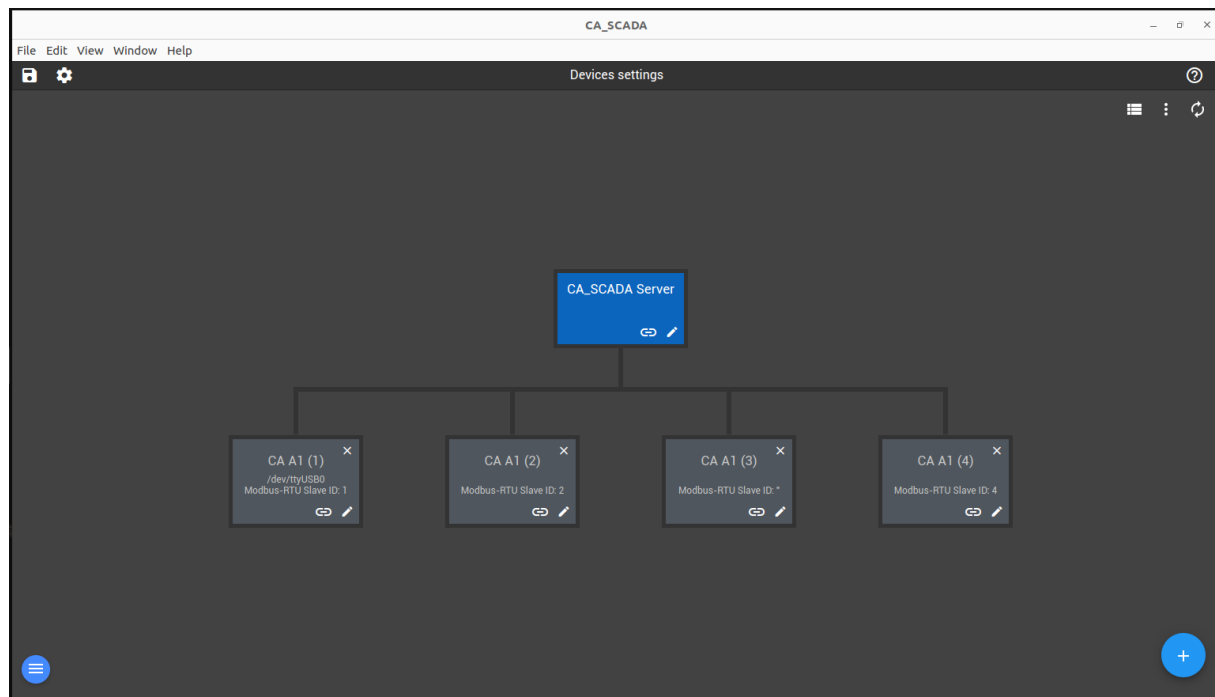


Figure 4.13: Multi-PLC CA SCADA system

4.17.6 Deployment Scenarios

CA SCADA is suitable for various industrial applications:

Manufacturing Operations:

- Production line monitoring and control
- Quality control systems
- Equipment performance tracking

Process Industries:

- Chemical process monitoring
- Water treatment facilities
- Energy management systems

Building Automation:

- HVAC system control
- Lighting management
- Security system integration

4.18 Conclusion

The Cherbal Automation Integrated Development Environment (CA IDE) v1.0 and its integrated SCADA platform represent a transformative leap in industrial automation tooling. Together, they deliver a unified ecosystem that bridges programming, simulation, real-time monitoring, and control—addressing the full lifecycle of automation projects for the CA A1 PLC platform.

Key advancements include:

- **Comprehensive Development Suite:** CA IDE’s core programming capabilities and AI-assisted tools (Structured Text generation, electrical design guidance) accelerate development, while the natively integrated CA SCADA system eliminates toolchain fragmentation. This unified environment enables seamless variable mapping and shared workflows between PLC code and supervisory interfaces.
- **Operational Excellence:** The SCADA platform’s real-time visualization (dynamic graphics, multi-screen dashboards), robust data acquisition (10,000+ tags, industrial protocols), and sophisticated alarm management empower operators. Integrated security features like role-based access control and audit trails ensure secure operations within the same environment used for development.
- **Cross-Functional Efficiency:**
 - *For Engineers:* Unified testing/simulation reduces context switching
 - *For Operators:* Consistent interface lowers training overhead
 - *For Maintenance:* Integrated diagnostics simplify troubleshooting
- **Scalable Architecture:** The web-based SCADA platform supports 50+ concurrent users across Linux systems, with configurable historical storage (SQLite/MySQL/-PostgreSQL) and deployment flexibility for manufacturing, process industries, and building automation.

Forward-Looking Innovation:

Cherbal Automation’s roadmap extends beyond current AI-assisted programming (including planned Ladder Diagram generation) to SCADA enhancements: advanced analytics, expanded protocol support, and enriched visualization capabilities. This dual-track evolution positions CA IDE as a future-ready platform for Industry 4.0 applications.

Final Perspective:

By converging AI-enhanced development tools with enterprise-grade SCADA functionality in a single environment, Cherbal Automation redefines industrial productivity. The solution not only solves immediate pain points in automation workflows but demonstrates how strategic integration of emerging technologies can create truly cohesive, accessible, and powerful engineering ecosystems for next-generation industrial systems.

General Conclusion

The evolution of Programmable Logic Controllers (PLCs) represents a cornerstone in the advancement of industrial automation, revolutionizing how industries optimize efficiency, safety, and adaptability. From their inception in the late 1960s as replacements for cumbersome relay systems to their current role as sophisticated, modular control units, PLCs have consistently addressed the growing demands of modern manufacturing, energy management, robotics, and smart infrastructure. This project underscores the transformative impact of PLC technology, emphasizing its architectural robustness, versatility, and integration with emerging technologies.

The structural design of PLCs comprising a microcontroller, memory systems, I/O interfaces, and communication modules ensures reliable performance in harsh industrial environments. By transitioning from hardwired relay logic to programmable, software-driven control, PLCs have introduced unprecedented flexibility, enabling rapid reconfiguration of processes without physical rewiring. The modular architecture of modern PLCs, exemplified by the CA A1 system, further enhances scalability, allowing seamless integration of additional I/O modules, communication protocols (e.g., USB-C, RS-485), and advanced functionalities such as real-time data analytics and IoT connectivity.

A key highlight of this project is the design and implementation of the CA A1 PLC, which embodies the principles of modern industrial automation. Its compact, multi-layered PCB design integrates robust power management, noise-immune signal conditioning, and dual-core processing capabilities, ensuring high-speed operation and reliability. The CA A1's support for diverse communication interfaces including USB-C for programming, RS-485 for industrial networking, and wireless protocols for IoT positions it as a versatile solution for both legacy systems and Industry 4.0 applications.

Complementing this hardware is the CA IDE, an intelligent programming environment that leverages AI to streamline code generation, simulation, and documentation. By bridging traditional ladder logic programming with AI-assisted Structured Text generation, the CA IDE democratizes automation development, reducing barriers for engineers and technicians.

The comprehensive automation ecosystem is further enhanced by the integrated CA SCADA platform and CA Register Monitor, which provide complete supervisory control and data acquisition capabilities. The CA SCADA system delivers real-time visualization through dynamic graphics and multi-screen dashboards, supporting over 10,000 tags with robust industrial protocol compatibility. Its web-based architecture enables concurrent access for up to 50+ users while maintaining enterprise-grade security through role-based access control and comprehensive audit trails. The CA Register Monitor complements

this by offering granular real-time monitoring of PLC registers and variables, enabling operators and maintenance personnel to perform detailed diagnostics and system optimization. Together, these tools create a unified development-to-operations workflow that eliminates toolchain fragmentation and accelerates project deployment from programming through commissioning to ongoing maintenance.

PLCs have become indispensable across industries due to their ability to enhance productivity, ensure consistent quality, and adapt to hazardous environments. In robotics, PLCs enable precise motion control and real-time feedback; in energy systems, they manage complex grid interactions; in manufacturing, they orchestrate synchronized assembly lines. The CA A1 extends these capabilities further, offering predictive maintenance through integrated sensors and facilitating seamless human-machine interaction via its modular design.

Looking ahead, the integration of AI and machine learning into PLC systems promises to unlock new frontiers in automation. Future PLCs will likely feature enhanced edge computing capabilities, self-optimizing control algorithms, and deeper convergence with cyber-physical systems. The CA IDE's AI-driven tools, combined with the advanced analytics capabilities of CA SCADA and the detailed monitoring features of CA Register Monitor, foreshadow this trend, illustrating how intelligent software can accelerate development cycles, improve system resilience, and provide comprehensive operational insights.

In conclusion, this project demonstrates that PLCs are not merely tools for automation but enablers of industrial innovation. The CA A1 PLC and its comprehensive ecosystem—encompassing the AI-enhanced CA IDE, enterprise-grade CA SCADA platform, and detailed CA Register Monitor—exemplify how cutting-edge hardware design, coupled with intelligent software and comprehensive monitoring solutions, can address today's challenges while laying the groundwork for tomorrow's smart factories. As industries continue to evolve, PLC technology will remain at the heart of this transformation, driving efficiency, sustainability, and adaptability in an increasingly connected world.

Bibliography

- [1] 10 étapes faciles pour le plan de conception dun projet de fabrication de circuit imprimé dans altium designer. <https://resources.altium.com/fr/p/10-easy-steps-comprehensively-designing-circuit-board-altium-designer>, 2018.
- [2] Everything about printed circuit board(pcb) layers. <https://www.pcbastore.com/blogs/printed-circuit-board-layers-explained.html>, 2023.
- [3] Richard E. Morley. The birth of the programmable logic controller: A technical revolution in industrial automation. In *Proceedings of the IEEE Conference on Industrial Control Systems*. IEEE, 1968.
- [4] International Society of Automation. Evolution of industrial control systems: From relays to smart plcs. <https://www.isa.org/automation-week/plc-evolution>, 2022.
- [5] Minghao Chen and Wei Li. Comparative analysis of plc architectures for robotic control systems. In *International Conference on Advanced Mechatronic Systems*. IEEE, 2021.
- [6] Q. Wang and P. Dubois. Galvanic isolation techniques in modern plc architectures. In *International Symposium on Industrial Electronics*. IEEE, 2019.
- [7] Y. Liu and T. Müller. Advanced signal conditioning techniques for mixed-signal industrial controllers. In *Proceedings of the 35th International Conference on Electronics Manufacturing*. IEEE, 2020.
- [8] Iec 61131-3 programming language specifications. https://en.wikipedia.org/wiki/Programmable_logic_controller, 2023.
- [9] Silicon Labs. Analog input signal conditioning for microcontrollers. <https://www.silabs.com/documents/public/application-notes/an1234-analog-conditioning.pdf>, 2023.
- [10] H. Zhang, R. Patel, and W. Schmidt. High-reliability dc-dc conversion for industrial automation systems. In *IEEE Transactions on Power Electronics*. IEEE, 2022.
- [11] Power supply design guidelines for industrial pcbs. <https://resources.altium.com/p/power-supply-design-industrial-pcbs>, 2023.
- [12] Silergy. An_sy8003/sy8003a high efficiency 5.5v, 3a continuous, 1mhz synchronous step down regulator. <https://www.silergy.com/download/downloadFile?id=3766&type=product&ftype=note>, 2018.

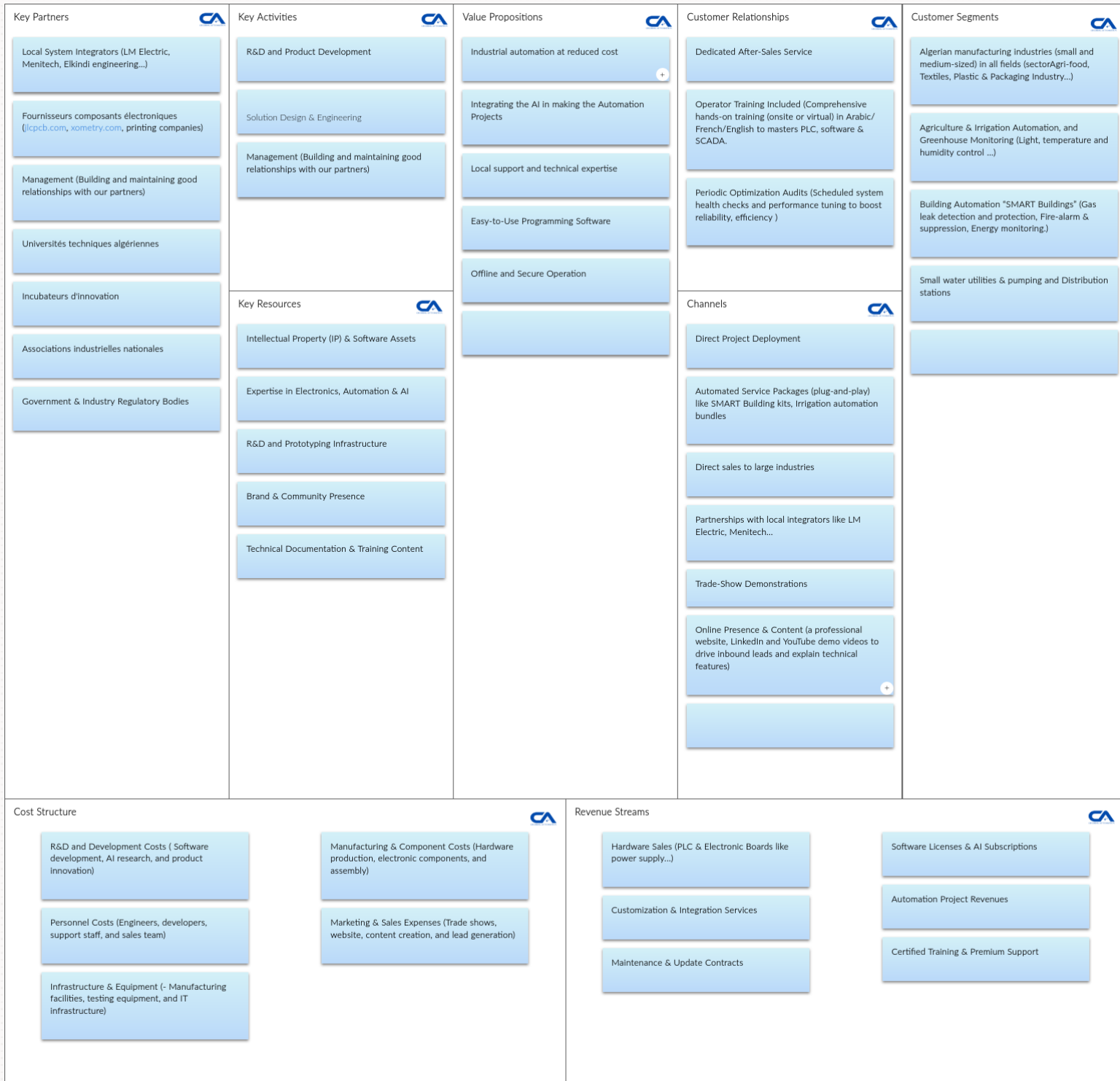
- [13] Texas Instruments. Tps25940evm-635: Evaluation module for tps25940x. Technical Report SLVUA44B, Texas Instruments, 2017.
- [14] Design for manufacturing and assembly (dfma) | siemens software. <https://www.plm.automation.siemens.com/global/en/our-story/glossary/design-for-manufacturing-and-assembly-dfma/53982>, 2023.
- [15] What is modular hardware? <https://resources.pcb.cadence.com/view-all/2020-what-is-modular-hardware>, 2020.
- [16] Taking a modular pcb design approach in altium designer. <https://resources.altium.com/p/taking-a-modular-pcb-design-approach-in-altium-designer>, 2021.
- [17] RayPCB. How to achieve a successful modular pcb design. <https://www.raypcb.com/modular-pcb-design/>, 2023.
- [18] Texas Instruments. Rs-485 design guide for industrial communication systems. <https://www.ti.com/lit/an/slla701a/slla701a.pdf>, 2021.
- [19] Plc programming pedagogy: Bridging theory and industrial practice. <https://www.industrial-automation-edu.org/plc-best-practices>, 2021.
- [20] Iec 61131-3 compliance in modern plc programming environments. <https://ca.idec.com/idec-ca/en/CAD/Software/WindLDR-PLC-Software/c/WindLDR>, 2023.
- [21] Phoenix Contact. Modular software architectures for industrial automation. <https://www.phoenixcontact.com/en-us/products/software/plc-programming>, 2024.
- [22] S. Agelis, S. Jacobsson, M. Jonsson, A. Alping, and P. Ligander. Modular interconnection system for optical pcb and backplane communication. In *Proceedings of the 16th International Parallel and Distributed Processing Symposium*. IEEE, 2002.
- [23] What is signal propagation delay in a pcb? <https://www.protoexpress.com/blog/signal-propagation-delay-pcb/>, 2020.
- [24] K. Shringarpure, S. Pan, J. Kim, and ... Formulation and network model reduction for analysis of the power distribution network in a production-level multilayered printed circuit board. *IEEE Transactions on Electromagnetic Compatibility*, 58(3):849–858, 2016.
- [25] An introduction to printed circuit boards. <https://resources.pcb.cadence.com/blog/2022-an-introduction-to-printed-circuit-boards>, 2022.
- [26] Pcb layers: Everything you need to know. <https://www.wevolver.com/article/pcb-layers-everything-you-need-to-know>, 2023.
- [27] Résoudre les problèmes de routage de circuits imprimés modernes. <https://resources.altium.com/fr/p/solving-the-routing-problems-of-modern-printed-circuits>, 2017.
- [28] NXP Semiconductors. Pcb layout guidelines for nxp mcus in bga packages. Rev.2, Apr. 2011, 2011.

- [29] W. Little. Pcb design guidelines for qfn and dqfn packages. *Microchip Technology Inc. Application Note*, AN1935, 2014.
- [30] Infineon Technologies. Board assembly recommendations for qfp package. https://www.infineon.com/dgdl/Infineon-Board_Assembly_Recommendations-QFP-Package-v06_00-EN.pdf?fileId=5546d46275b79adb0175bd38e18f609, 2020.
- [31] Conception pour la fabrication (dfm) principes expliqués. https://fractory.com/design-for-manufacturing-dfm/#What_Is_DFM, 2023.
- [32] O. Kerbrat. Méthodologie de conception d'outillages modulaires hybrides basée sur l'évaluation quantitative de la complexité de fabrication. In *Thèse de doctorat, École Centrale de Nantes*, 2009.
- [33] USB Implementers Forum. Usb type-c cable and connector specification. In *Release 2.3*, 2023.
- [34] Usb-c interface design for industrial embedded systems. <https://www.protoexpress.com/blog/usb-c-design-industrial-pcb/>, 2022.
- [35] Uart basics and hardware flow control. <https://www.ti.com/lit/an/s11a324a/s11a324a.pdf>, 2022.
- [36] FTDI Chip. Ft2232h datasheet - dual high-speed usb to multipurpose uart/fifo ic. https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT2232H.pdf, 2020.
- [37] Z. Lin, D. Meng, and Q. Zhong. Design and implementation of otg communication system terminal based on usb. In *2011 International Conference on Control, Automation and Systems Engineering (CASE)*, pages 1–4, 2011.
- [38] Usb3320c - hi-speed usb ulpi transceiver. <https://ww1.microchip.com/downloads/aemDocuments/documents/UNG/ProductDocuments/DataSheets/USB3320-Data-Sheet-DS00002293B.pdf>, 2019.
- [39] Microchip. Usb332x transceiver layout guidelines. <https://ww1.microchip.com/downloads/en/DeviceDoc/00001692c.pdf>, 2018.
- [40] USB Implementers Forum. Usb power delivery specification. In *Revision 3.2*, 2021.
- [41] Telecommunications Industry Association. Electrical characteristics of generators and receivers for use in balanced digital multipoint systems (tia-485-a). In *ANSI/TIA-485-A Standard*. TIA, 2015.
- [42] Rs-485 basics: When and where to use rs-485. <https://www.analog.com/en/resources/technical-articles/rs485-basics-when-and-where-to-use-rs485.html>, 2022.
- [43] Modbus Organization. Modbus application protocol specification. https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf, 2012.
- [44] Modicon 084: The first programmable logic controller. <https://www.modicon.com/history/084-plc>, 2022.

- [45] Espressif Systems. Esp32-wrover-e technical reference manual. https://www.espressif.com/sites/default/files/documentation/esp32-wrover-e_technical_reference_manual_en.pdf, 2023.
- [46] Eric Bogatin. Power distribution network design for high-speed pcbs. In *Signal Integrity Journal*, 2020.
- [47] Espressif Systems. Esp32-wrover-e esp32-wrover-ie datasheet. https://www.espressif.com/sites/default/files/documentation/esp32-wrover-e_esp32-wrover-ie_datasheet_en.pdf, 2023.
- [48] Y. Liu and T. Müller. Unified development environments for next-gen industrial automation. In *Proceedings of the 36th International Conference on Automation Technology*. IEEE, 2024.
- [49] BrainHub Research Team. Ai-assisted programming in industrial ides: Current trends and future directions. <https://brainhub.eu/library/software-developer-age-of-ai>, 2025.
- [50] Open architecture design patterns in modern plc software. <https://www.automationdirect.com/ls/software>, 2024.
- [51] A. Schmidt and T. Wagner. Modular ai integration in industrial control systems. In *IEEE Conference on Industrial Cyber-Physical Systems*. IEEE, 2024.
- [52] autonomy. openplc thechnology. <https://autonomylogic.com/download/>, 2025.

Appendix 1

CA Business Model Canvas



Appendix 2

Introduction

This annex provides a comprehensive declaration table for all variables used in the PLC program. The table follows IEC 61131-3 standards and includes detailed specifications for each variable including memory addressing, data types, and initialization parameters.

PLC System Specifications

The target PLC system is a CE A1 compact controller with the following specifications:

- **Digital Inputs:** 8 channels (%IX0.0 to %IX0.7) - 24V DC
- **Digital Outputs:** 8 channels (%QX0.0 to %QX0.7) - 24V DC
- **Analog Inputs:** 3 channels (%IW0 to %IW2) - 0-10V DC
- **Analog Outputs:** 2 channels (%QW0 to %QW1) - 0-10V DC
- **Power Supply:** 24V DC
- **Design:** Compact unit with no expansion capabilities

Variable Column Specifications

Name of Variable

Variable naming follows a structured approach for clarity and maintainability:

- Use device-function-location pattern (e.g., MTR1_RUN_STATUS)
- Ensure self-documenting names with clear prefixes indicating function
- Use consistent capitalization (PascalCase or snake_case) throughout the project

Class

Precisely categorize variables using standard IEC classes:

- **Input** - Physical signals into the PLC (sensors, switches, feedback signals)
- **Output** - Physical signals from the PLC (actuators, motors, indicators)
- **InOut** - Bidirectional parameters for function blocks or programs
- **External** - Imported variables from other program organization units
- **Local** - Internal variables with limited scope to the current POU
- **Temp** - Temporary variables that exist only during execution cycles

Type

Implement strict data typing using standard IEC 61131-3 data types:

- **Boolean:** BOOL (single bit TRUE/FALSE)
- **Integer:** SINT (8-bit), INT (16-bit), DINT (32-bit), LINT (64-bit) and unsigned variants USINT, UINT, UDINT, ULINT
- **Floating Point:** REAL (32-bit), LREAL (64-bit IEEE 754)
- **String:** STRING[n], WSTRING[n] with explicit length declaration
- **Time:** TIME, DATE, TOD (Time of Day), DT (Date and Time)
- **Bit Strings:** BYTE, WORD, DWORD, LWORD for bit manipulation
- **Complex:** ARRAY[x..y] OF DataType, structured types, enumerations

Memory Address

Use manufacturer-specific addressing with proper syntax:

- Digital Inputs: %IXa.b (a=byte address, b=bit position)
- Digital Outputs: %QXa.b (a=byte address, b=bit position)
- Analog Inputs: %IWa (W=word/16-bit)
- Analog Outputs: %QWa

Initial Value

Specify startup values with proper formatting:

- **BOOL:** TRUE or FALSE (uppercase per IEC standard)
- **Numeric:** Include units where applicable (e.g., 0.0_degC, 100_mm)
- **String:** In double quotes with escape characters when needed
- **Array:** Use initialization list format {val1, val2, ...}

Option

Specify memory retention behavior:

- **Constant** - Immutable value protected from runtime modification
- **Retain** - Value preserved across power cycles in non-volatile memory
- **NonRetain** - Value reinitialized on power cycle or warm restart
- **ReadOnly** - Value can be read but not modified by the program

Documentation

Provide comprehensive variable documentation including:

- Physical connection details (terminal block, pin number)
- Engineering units with standard notation
- Valid operating ranges with limits
- Failure behavior and default states

Variable Declaration Table

The following table contains all variables used in the PLC program, organized according to IEC 61131-3 standards:

Variable Name	Class	Type	Memory Address	Initial Value	Option	Documentation
Start_Btn	Input	BOOL	%IX0.0	FALSE	NonRetain	Start button for process initiation
Stop_Btn	Input	BOOL	%IX0.1	FALSE	NonRetain	Emergency stop button - normally closed contact
Auto_Mode	Input	BOOL	%IX0.2	FALSE	NonRetain	Automatic mode selector switch
Manual_Mode	Input	BOOL	%IX0.3	FALSE	NonRetain	Manual mode selector switch
Level_Sensor	Input	BOOL	%IX0.4	FALSE	NonRetain	High level sensor - proximity switch
Pressure_OK	Input	BOOL	%IX0.5	FALSE	NonRetain	Pressure switch - system pressure OK
Safety_Guard	Input	BOOL	%IX0.6	FALSE	NonRetain	Safety guard position sensor
Spare_Input	Input	BOOL	%IX0.7	FALSE	NonRetain	Reserved for future expansion
Main_Motor	Output	BOOL	%QX0.0	FALSE	NonRetain	Main drive motor contactor
Pump_Motor	Output	BOOL	%QX0.1	FALSE	NonRetain	Hydraulic pump motor
Valve_Solenoid	Output	BOOL	%QX0.2	FALSE	NonRetain	Main control valve solenoid
Status_Light	Output	BOOL	%QX0.3	FALSE	NonRetain	System running indicator light
Continued on next page...						

Variable Name	Class	Type	Memory Address	Initial Value	Option	Documentation
Alarm_Light	Output	BOOL	%QX0.4	FALSE	NonRetain	Fault condition alarm light
Buzzer	Output	BOOL	%QX0.5	FALSE	NonRetain	Audible alarm buzzer
Cooling_Fan	Output	BOOL	%QX0.6	FALSE	NonRetain	Cooling system fan
Spare_Output	Output	BOOL	%QX0.7	FALSE	NonRetain	Reserved for future expansion
Temp_Sensor	Input	INT	%IW0	0	NonRetain	Temperature sensor input (0-1000 = 0-100°C)
Pressure_Analog	Input	INT	%IW1	0	NonRetain	Pressure transducer (0-1000 = 0-10 bar)
Flow_Rate	Input	INT	%IW2	0	NonRetain	Flow rate sensor (0-1000 = 0-100 L/min)
Speed_Control	Output	INT	%QW0	0	NonRetain	Variable speed drive reference (0-1000 = 0-100%)
Valve_Position	Output	INT	%QW1	0	NonRetain	Proportional valve position (0-1000 = 0-100%)
System_State	Local	INT	N/A	0	NonRetain	Current system state machine value
Cycle_Counter	Local	DINT	N/A	0	Retain	Total production cycle counter
Fault_Code	Local	INT	N/A	0	Retain	Last fault code for diagnostics
Process_Timer	Local	TIME	N/A	T#0s	NonRetain	Process timing control variable
Delay_Timer	Local	TIME	N/A	T#0s	NonRetain	General purpose delay timer
Temp_Setpoint	Local	REAL	N/A	25.0	Retain	Temperature setpoint in degrees Celsius
Press_Setpoint	Local	REAL	N/A	5.0	Retain	Pressure setpoint in bar
System_Ready	Local	BOOL	N/A	FALSE	NonRetain	System ready to start flag
Auto_Running	Local	BOOL	N/A	FALSE	NonRetain	Automatic mode running status
Continued on next page...						

Variable Name	Class	Type	Memory Address	Initial Value	Option	Documentation
Fault_Active	Local	BOOL	N/A	FALSE	NonRetain	System fault condition flag

Data Type Specifications

Boolean Types

- **BOOL:** Single bit value (TRUE/FALSE)
- Used for digital inputs, outputs, and status flags
- Memory allocation: 1 bit per variable

Integer Types

- **INT:** 16-bit signed integer (-32,768 to 32,767)
- **DINT:** 32-bit signed integer (-2,147,483,648 to 2,147,483,647)
- Used for analog values, counters, and state machines

Real Types

- **REAL:** 32-bit floating point (IEEE 754)
 - Used for precise calculations and setpoints
- Range: $\pm 1.18 \times 10^{-38}$ to $\pm 3.40 \times 10^{38}$

Time Types

- **TIME:** Duration value in milliseconds
- Format: T#1s500ms or T#1500ms
- Used for timers and delays

Memory Retention Options

Retain

Variables marked as **Retain** preserve their values across power cycles and are stored in non-volatile memory. These are typically used for:

- Production counters
- Setpoint values
- Configuration parameters
- Fault codes

Non-Retain

Variables marked as **Non-Retain** are reinitialized to their default values on power-up or warm restart. These include:

- I/O signal states
- Temporary calculations
- Timer states
- Process status flags

Safety Considerations

All safety-critical variables follow these guidelines:

- Emergency stop inputs use normally closed contacts
- Safety interlocks default to safe state (FALSE)
- Critical outputs are fail-safe (power loss = safe state)
- Diagnostic variables retain fault information

Appendix 3

EasyEDA for designing PCBs

Create an EasyEDA Account and Create a New Project

Welcome to EasyEDA, a great web based on EDA (Electronics Design Automation) tool for electronics engineers, educators, students, makers and enthusiasts. There is no need to install any software. Just open EasyEDA in any HTML5 capable, standards compliant web browser. <https://easyeda.com>



Figure 4.14: EasyEDA main page

1. Register to create a new account on EasyEDA.
2. After successful login, click on EasyEDA Designer button

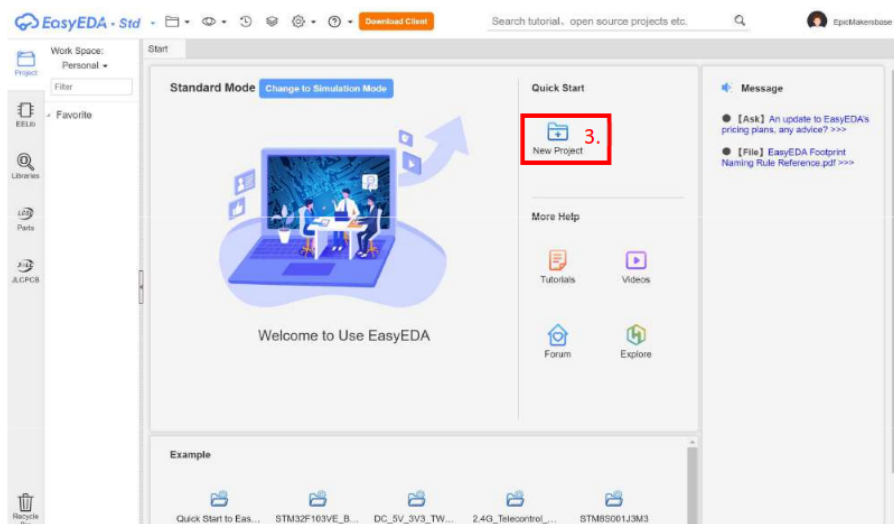
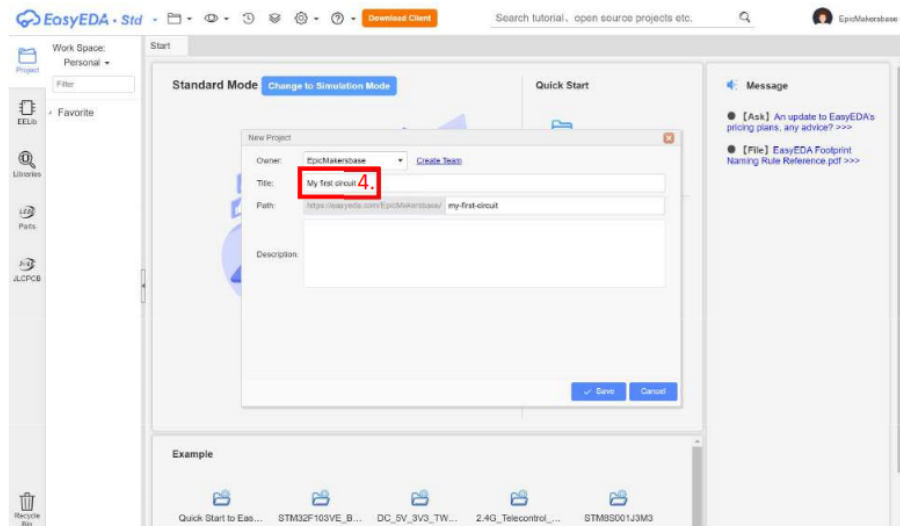


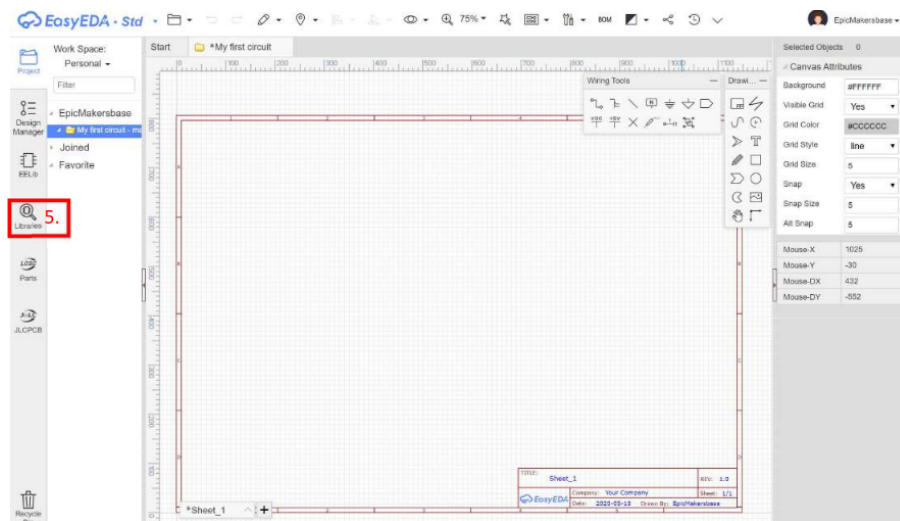
Figure 4.15: Empty workspace

3. Create new project

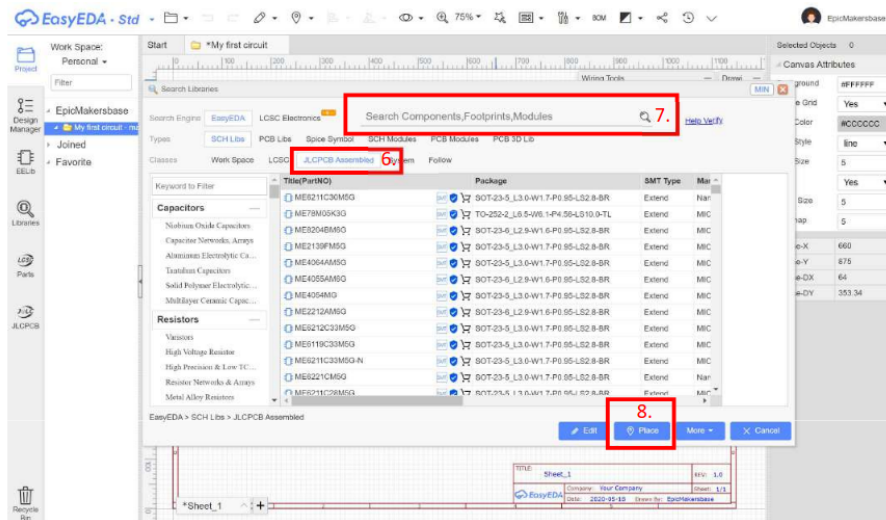


4. Name your project and click save.

Create New Schematic and PCB



5. Click on library to select components to add.



6. Select JLCPCB Assembly (JLCPCB are manufacturers of professional PCB) so we can use their SMT (Assembly) service.
7. Search the component by part number or value, select the component from the list.
8. Click place and place it on the sheet.

Components list for making blinking LED using 555 timer IC:

#	Component Name	Part No.	Quantity
1	555 Timer IC	LMC555CMX	1
2	CR2032 cell holder	CR2032-BS-6-1	1
3	10uF Capacitor	CA45A-B-25V-10uF-K	1
4	LED lights	17-21/T1D-CP2R1TY/3T	2
5	100K resistor	RTAN0805BKE100K	2
6	100 Ω resistor	TC0225F1000TCE	2
7	Button switch	TS-1187A-C-G-A	1

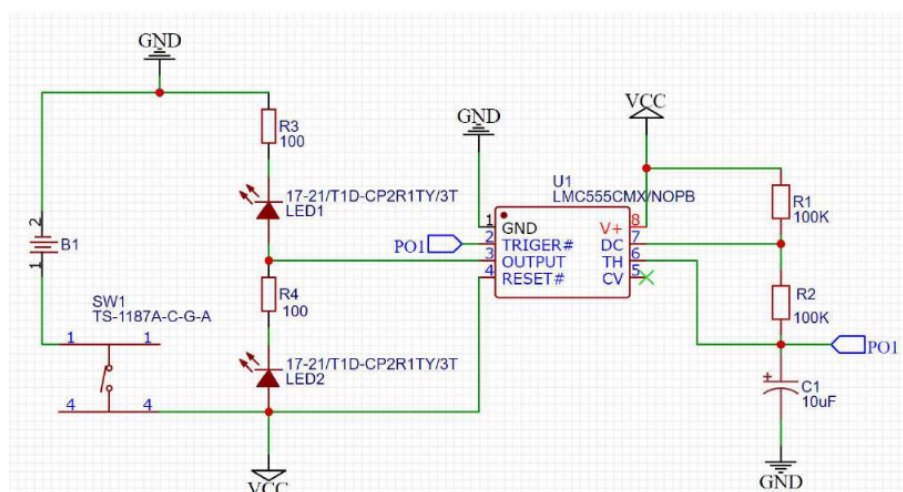
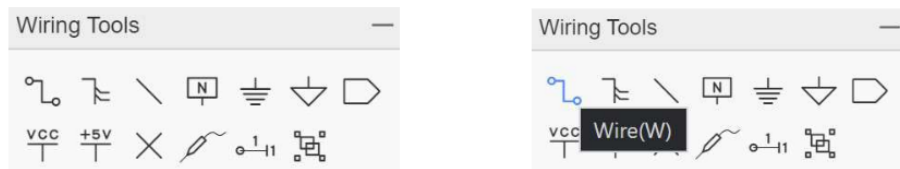
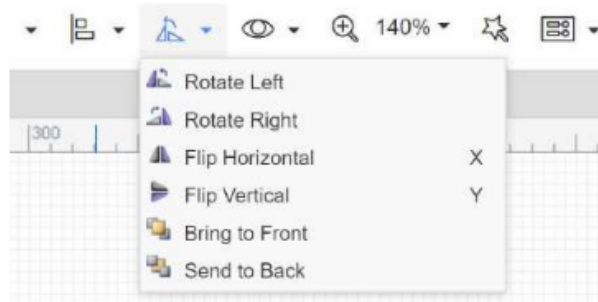


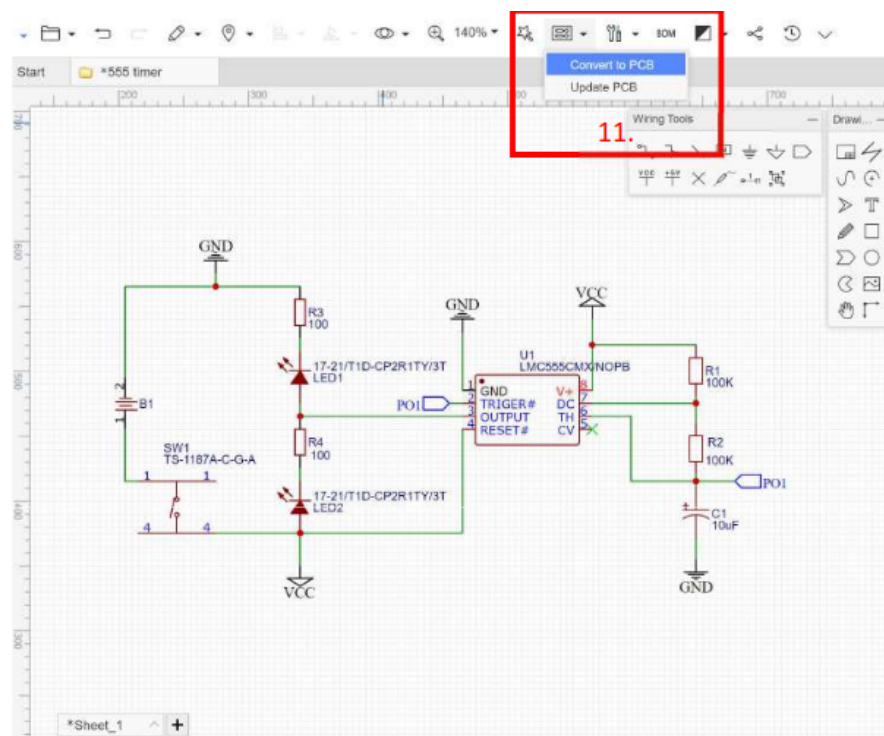
Figure 4.16: Schematic drawing



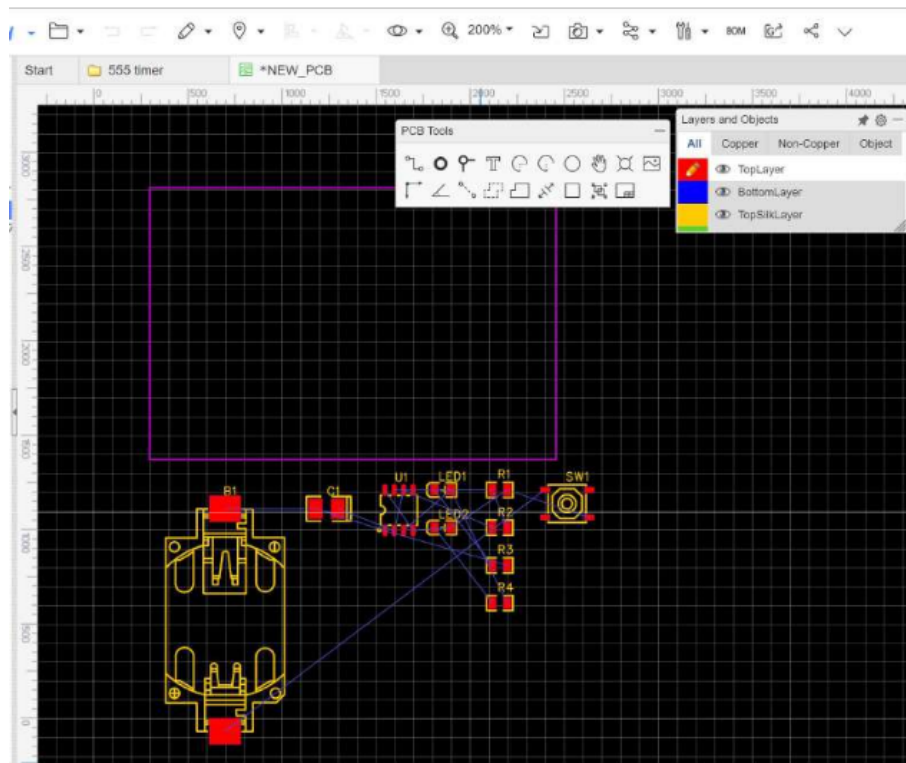
9. Create the above schematic using the wiring tools to join all the components. Components can be placed according to user as long as all the connections are made.



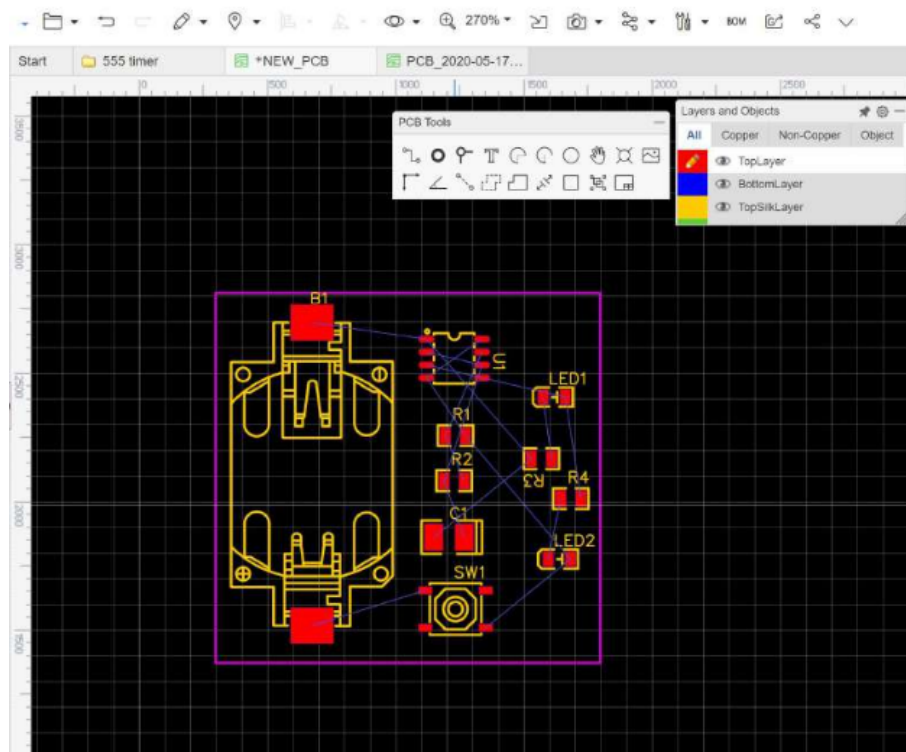
10. Rotate components as per the schematic given



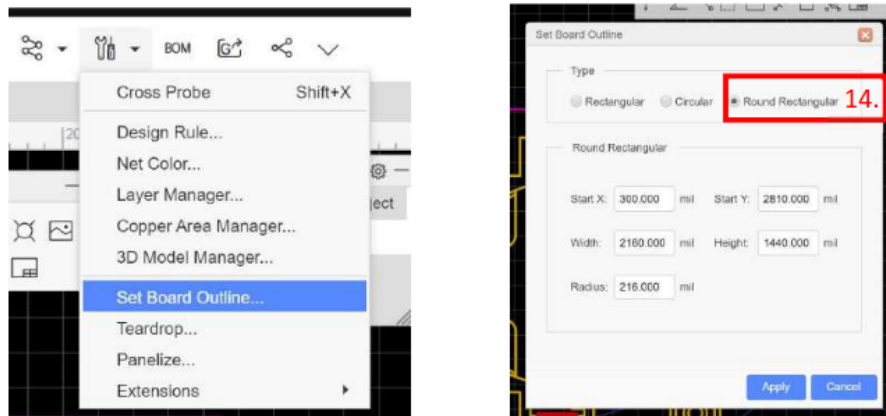
11. Select convert to PCB option as shown in figure



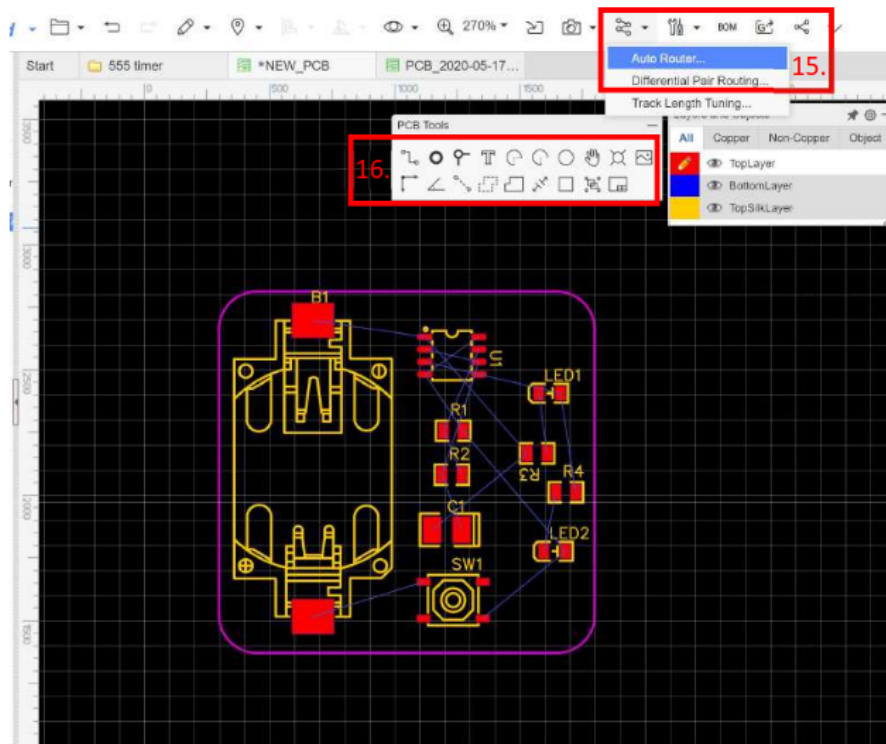
12. Arrange all the components as per requirements and shrink the pink outer border



13. Rearrange the outer border of PCB.

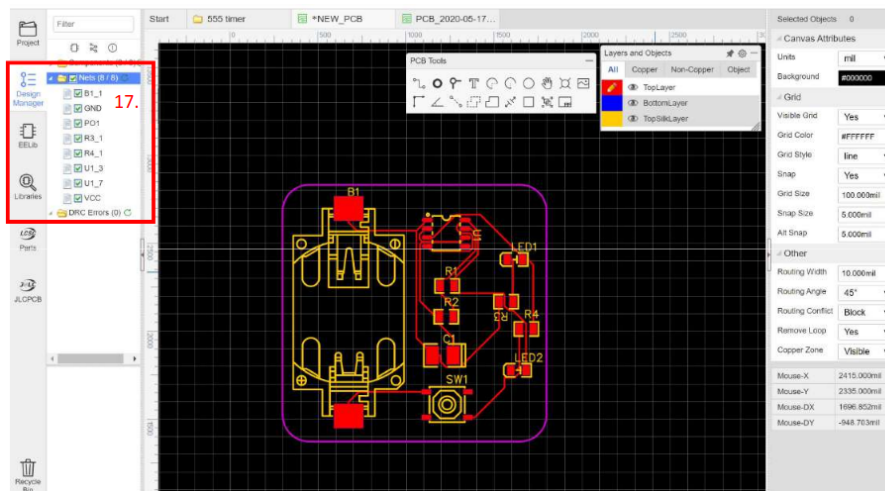


14. Select Set Board Outline, Select Round Rectangular and click apply.

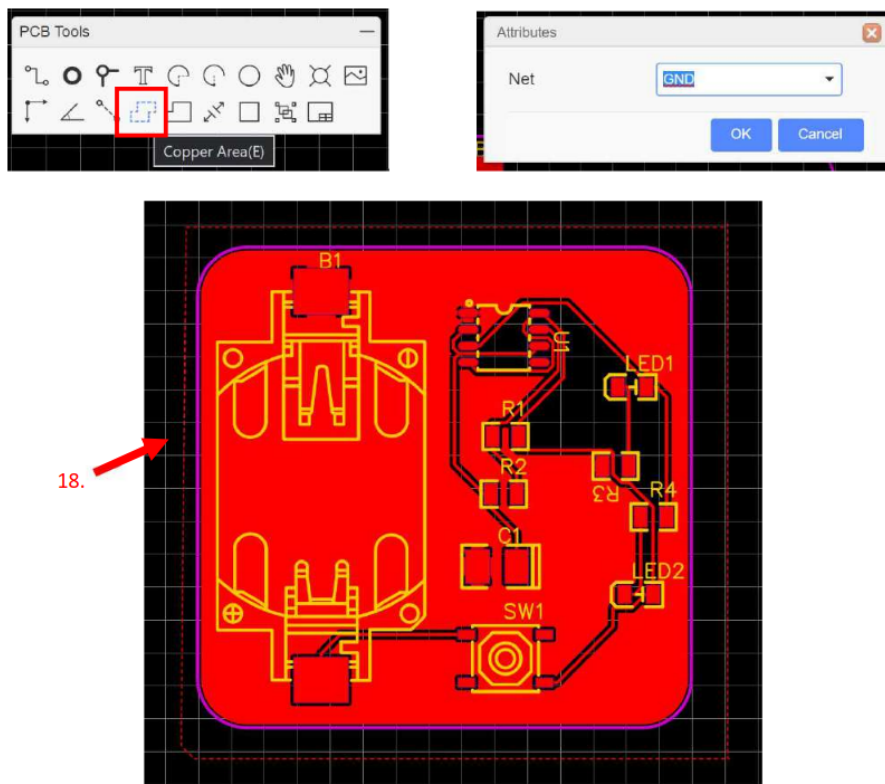


15. Use Auto router feature to make the traces automatically.

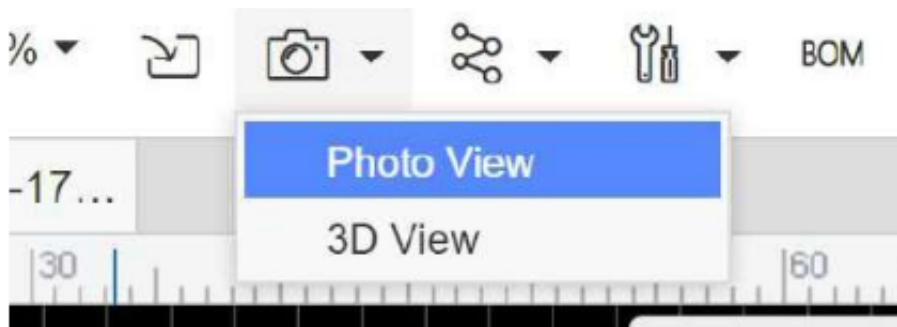
16. The PCB can be manually routed as well with PCB Tools.



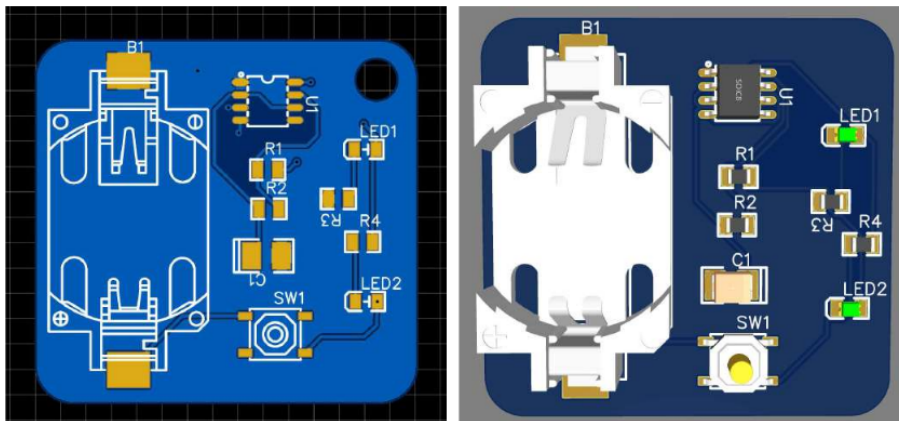
17. Check in Design Manager tab if all the Net points are routed without error. If the nets have problem connecting, try to increase the space between components or move the component to different location.



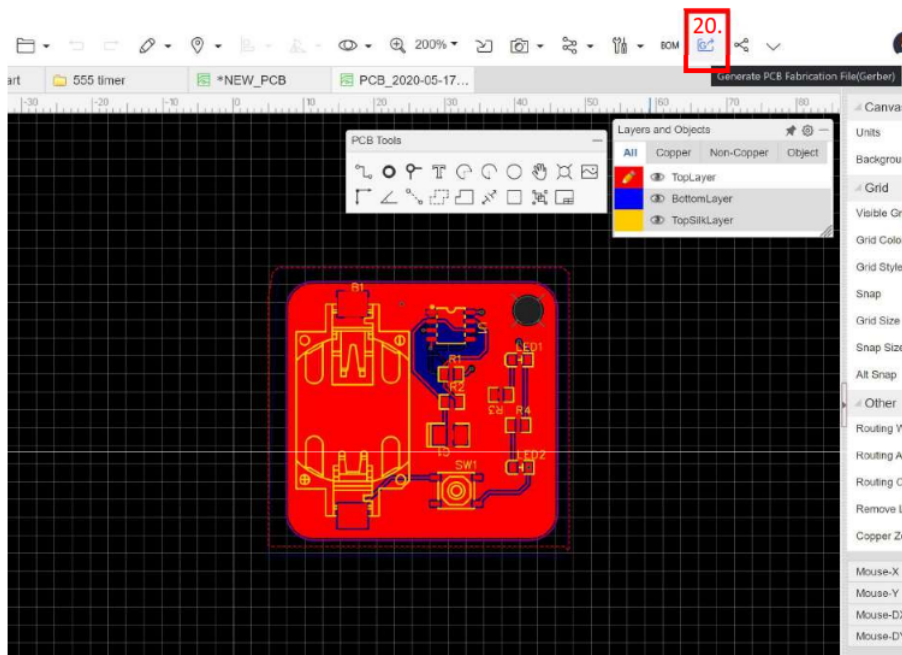
18. Select the copper area tool, select Net as GND and create outer border line around the board. This will fill the unused area with GND. You can also do this for the bottom area.



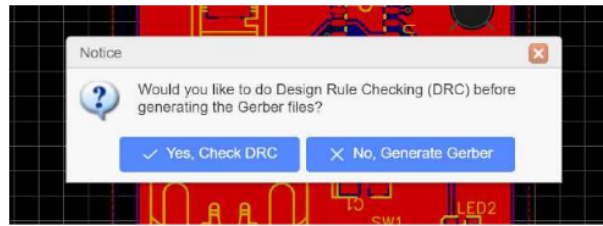
19. Choose Photoview or 3D view to see how the pcb will actually look like



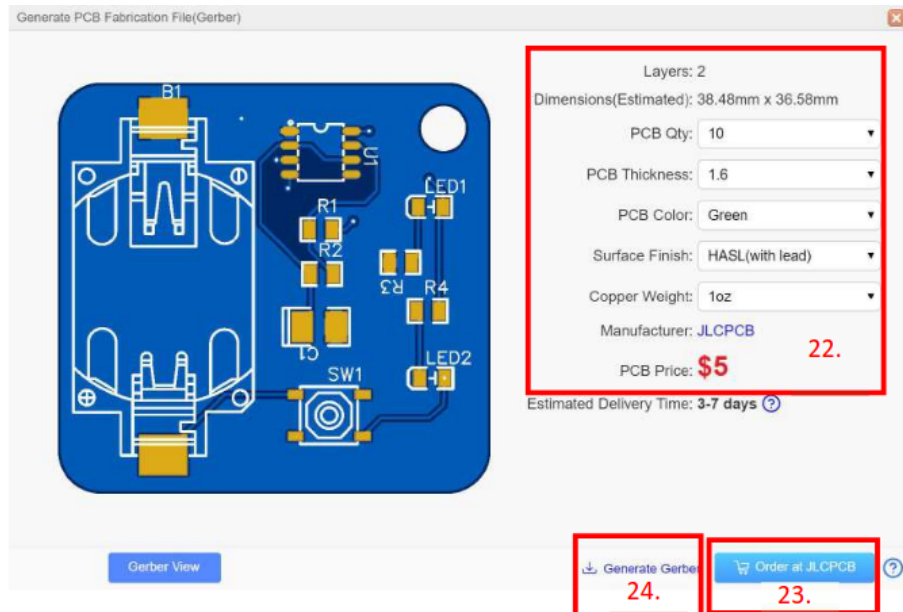
Ordering PCB



20. Click Generate PCB Fabrication File button.



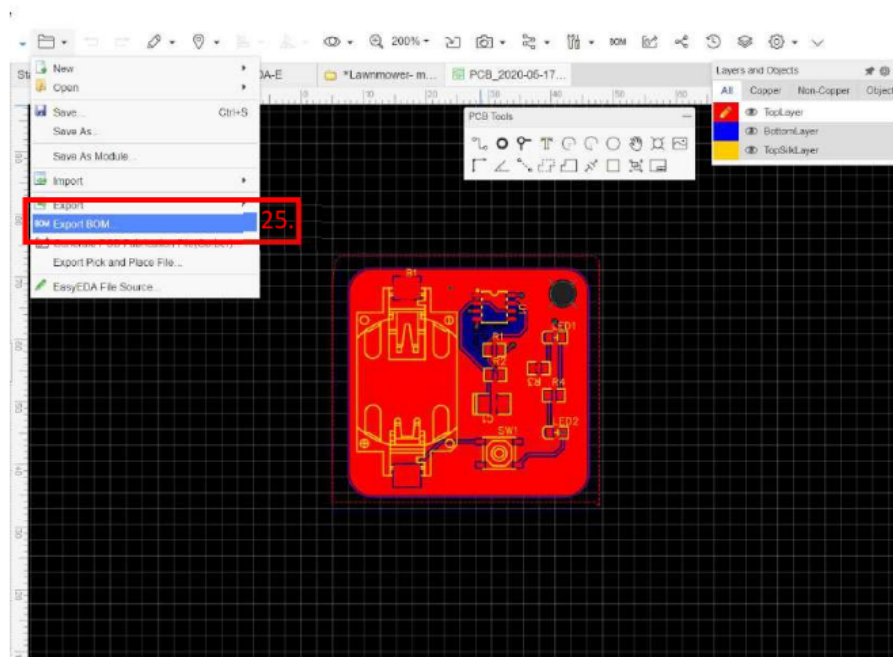
21. Select No to only Generate Gerber.



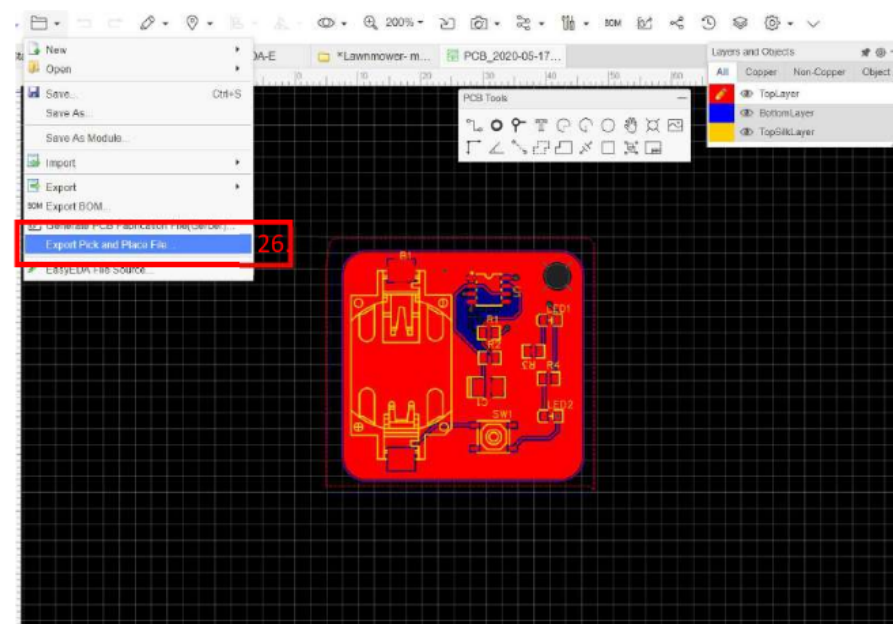
22. This shows the quantity, price for total quantity and the estimated delivery time.

23. The PCB can be directly ordered from JLCPCB.

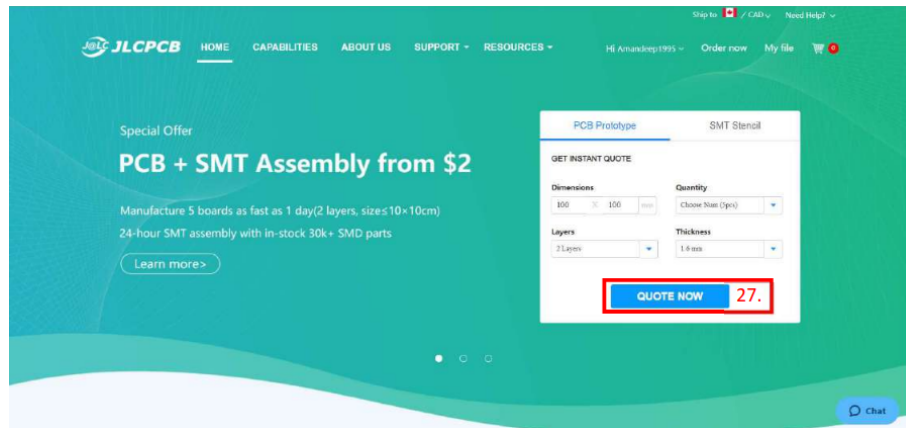
24. Gerber files can also be downloaded for manually uploading files on JLCPCB. If you would like fully assembled PCB with all components on it (SMT Service), follow further steps.



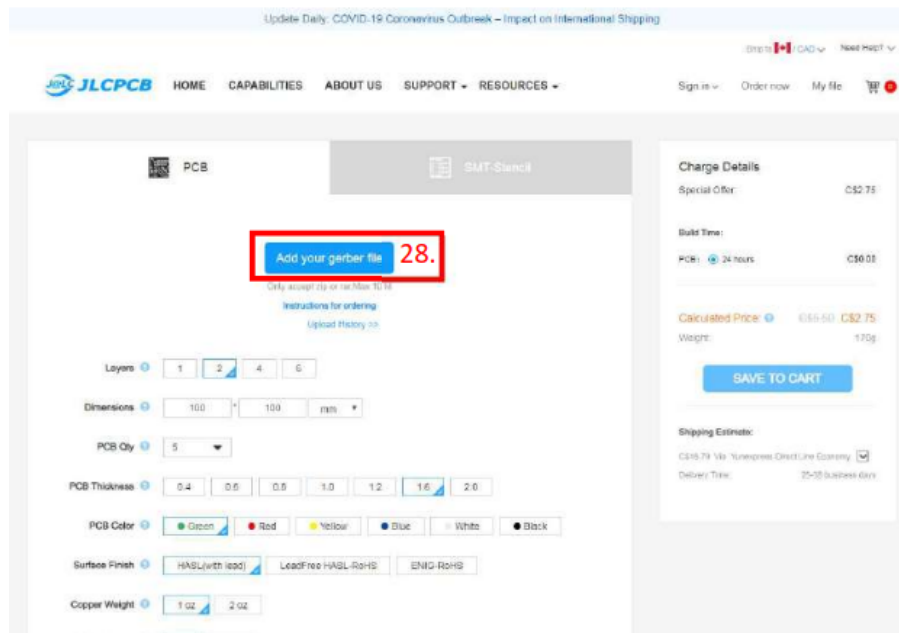
25. Export BOM (Bill of Material) and save for using assembly service (SMT service) of JLCPCB.



26. Export Pick and Place file and save it for SMT services



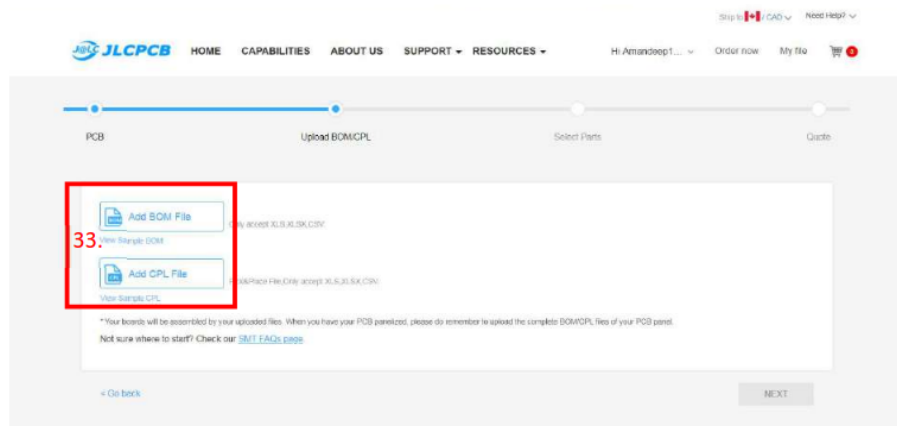
27. Open <https://jlcpcb.com/> and click on Quote now.



28. Upload gerber files downloaded in step 24.

29. 2D image of the PCB can be seen and reviewed, you can order only PCB without any components soldered on it at this point.

30. Click on SMT Assembly
31. Choose Assemble on top side.
32. Click confirm to go further



32. Upload files downloaded in step 25 and 26 and click next

Total 7 parts detected, of which 7 parts to be confirmed, 0 parts not selected.

Uploaded BOM Data			Review Matched Parts			
Designator	Name	Footprint	Matched Part Detail	Qty	Unit Price	Operate
SW1	TS-1187A.C.G.A	SW-SMD_4P-L5.1. WB-1-PB-70-L56.4	TS-1187A.C.G.B SMD-SW-4_P-150.5-150.5 C1010006	1	C\$0.0950	confirmed
B1	CR2032-85-6-1	BA7-TH_CR2032-85-6-1	CR2032-85-6-1 SMD-BAT-CR2032-85-6-1 C1010177	1	C\$0.1770	confirmed
R3,R4	100	R0805	00-1HG(2532) ±1% 0805 C1101733	2	C\$0.0002	confirmed
R1,R2	100K	R0805	100K(11030) ±0.1% ±25P... 0805 C1047036	2	C\$0.4159	confirmed
LED1,LED2	17-21MT10-CP2R1TY3T	LED0805-RD	LED_XT_BLV-57-140 LED_0805 C1010660	2	C\$0.0967	confirmed
C1	10µF	CAP-SMD_L5-3-W2.5-R-RO	10µF(105) ±10% 25V C48E-B_3525 C1010302	1	C\$0.1745	confirmed
U1	LM3555CI/NOPB	SOIC-8_L4.5-W3.9-P1.27-L36.0-5L	LM3555CI/NOPB SOIC-8_2.264,3x1.27P C1010703	1	C\$0.4020	confirmed

< Go back NEXT

34. Confirm all the components and click next.

PCB

Upload BOM/CPL

Select Parts

Review Parts Placement

The parts placement is for reference purpose only. If you are sure the rotation and polarity of your design are correct, you can omit the preview and continue placing your order.

Selected Parts (7 items)

Part Detail	Selected By	Designator	Price
Unimyal Elec. 010505P10032105			

Charge Details

Special Offer: C\$2.75

SMT Price:

Setup fee:	C\$9.63
Stencil:	C\$2.00
Panel:	C\$0.00
Components:	C\$15.04
Extended components fee:	C\$28.90
Assembly:	C\$0.00

Build Time:

PCB:	1-2 days	C\$0.00
SMT:	48 hours	

Total Price: C\$58.39

Weight: 39.35g

[SAVE TO CART](#)

By placing your order, pls read the [Terms and Conditions of JLCPCB SMT Service](#).

35. A 3D view will be displayed of the PCB, orientation of components in the display might be wrong but it will be correct in the actual ordered PCB. Save to cart and order it with your shipping details.

Appendix 4

OpenPLC Overview [52]

OpenPLC is an open-source industrial control platform that transforms popular microcontrollers into programmable logic controllers (PLCs). Built on the IEC 61131-3 standard, it supports five programming languages: Structured Text (ST), Instruction List (IL), Ladder Diagram (LD), Sequential Function Chart (SFC), and Function Block Diagram (FBD). Ladder Diagram is the most commonly used language in industrial applications.

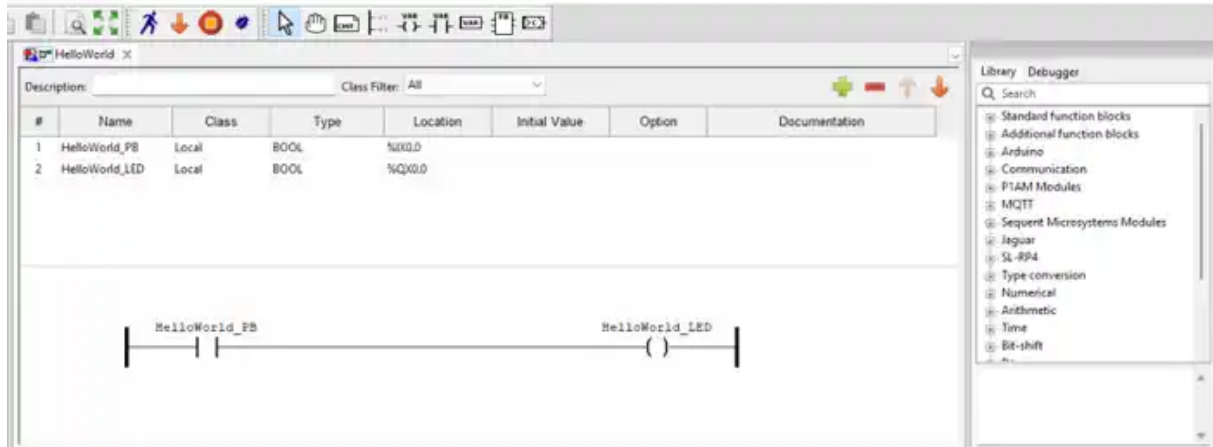


Figure 4.17: OpenPLC interface

Key Features

Hardware Support

OpenPLC version 3 supports multiple platforms:

- Arduino boards (ATMEGA328 8-bit microcontroller)
- ESP32 and ESP8266 modules
- Raspberry Pi (BCM 2711 for Pi 4B, BCM 2937 for Pi 3B)

Technical Capabilities

- Runs ST code on target microcontrollers
- C-code installation on Raspberry Pi
- Runtime dashboard for monitoring I/O devices
- Simulation environment for program testing
- Standard PLC bit instructions (XIC, OTE)

Implementation Process

Development Steps

1. Download and install OpenPLC software
2. Create project directory and folder
3. Wire hardware components (switches, LEDs, resistors)
4. Define I/O tags with BOOL data types
5. Build ladder logic program using XIC and OTE instructions
6. Test program through simulation
7. Upload compiled code to target microcontroller

Advantages

- **Cost-effective:** Uses affordable microcontroller hardware
- **Open-source:** No licensing fees, customizable code
- **Standards-compliant:** Based on IEC 61131-3 international standard
- **Educational:** Excellent learning platform for automation concepts
- **Flexible:** Supports multiple programming languages and hardware platforms
- **Accessible:** Intuitive graphical programming interface

Applications

OpenPLC is suitable for:

- Educational institutions teaching industrial automation
- Small-scale industrial control systems
- Prototyping and proof-of-concept projects
- Cost-sensitive automation applications
- Custom control system development

Conclusion

OpenPLC democratizes industrial automation by providing a professional-grade, standards-compliant PLC platform at minimal cost. Its combination of open-source accessibility, comprehensive hardware support, and adherence to international standards makes it an attractive alternative to proprietary PLC systems. The platform successfully bridges the gap between educational learning tools and industrial-grade control systems, making advanced automation technology accessible to a broader audience.