

SCADA in Domestic Power System and Automation

Ishwarya.M¹, Nandhu Nivedha.J², Chidambaram.S³

UG Scholar, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India^{1,2}

Associate Professor, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India³

ABSTRACT: This project describes the visualization of electrical units consumed by home appliances using Twin CAT3 PLC(Programmable Logic Controller). The software used is Ladder logic language program. The tariff for the consumed power by individual appliance per hour and the overall tariff for all the appliances per day and per month are calculated and monitored. The visualization of current status and controlling of home appliances is made possible with the use of SCADA screen which is interfaced to PLC through communication protocols. The automation of home appliance is made from remote or from any centralized location.

KEYWORDS: TWINCAT3, PLC, ladder logic, SCADA, tariff

I. INTRODUCTION

The SCADA automation project used to calculate the electricity bill for home appliances. This is done by using PLC(Programmable Logic Control). The software used is Twin CAT 3. The software is a BECKHOFF New Automation Technology. The system of automation is applicable for all home appliances to monitor the working hours of each appliances, their units consumed and the overall tariff. This method of monitoring is used nowadays in industries, substations and in the areas where automation is mandatory.

II. OBJECTIVE

The main objective of this project is to control the home appliances using scada dashboard and to calculate the electricity bill for home appliances through PLC(Programmable Logic Control).

III. SCADA

Supervisory control and data acquisition (SCADA) is a control system architecture that uses computers, networked data communications and graphical user interfaces for high-level process supervisory management, but uses other peripheral devices such as programmable logic controllers and discrete PID controllers to interface to the process plant or machinery. The operator interfaces which enable monitoring and the issuing of process commands, such as controller set point changes, are handled through the SCADA supervisory computer system. However, the real-time control logic or controller calculations are performed by networked modules which connect to the field sensors and actuators.

The SCADA concept was developed as a universal means of remote access to a variety of local control modules, which could be from different manufacturers allowing access through standard automation protocols. In practice, large SCADA systems have grown to become very similar to distributed control systems in function, but using multiple means of interfacing with the plant. They can control large-scale processes that can include multiple sites, and work over large distances. It is one of the most commonly-used types of industrial control systems, however there are concerns about SCADA systems being vulnerable to cyber warfare/cyber terrorism attacks.

IV. PROGRAMMABLE LOGIC CONTROLLER

A programmable logic controller (PLC), or programmable controller is an industrial digital computer which has been ruggedized and adapted for the control of manufacturing processes, such as assembly lines, or robotic devices, or any activity that requires high reliability control and ease of programming and process

fault diagnosis. They were first developed in the automobile industry to provide flexible, ruggedized and easily programmable controllers to replace hard-wired relays and timers. Since then they have been widely adopted as high-reliability automation controllers suitable for harsh environments. A PLC is an example of a "hard" real-time system since output results must be produced in response to input conditions within a limited time, otherwise unintended operation will result. PLCs can range from small "building brick" devices with tens of I/O in a housing integral with the processor, to large rack-mounted modular devices with a count of thousands of I/O, and which are often networked to other PLC and SCADA systems. They can be designed for multiple arrangements of digital and analog inputs and outputs (I/O), extended temperature ranges, immunity to electrical noise, and resistance to vibration and impact. Programs to control machine operation are typically stored in battery-backed-up or non-volatile memory. It was from the automotive industry in the USA that the PLC was born. Before the PLC, control, sequencing, and safety interlock logic for manufacturing automobiles was mainly composed of relays, cam timers, drum sequencers, and dedicated closed-loop controllers. Since these could number in the hundreds or even thousands, the process for updating such facilities for the yearly model change-over was very time consuming and expensive, as electricians needed to individually rewire the relays to change their operational characteristics. When digital computers became available, being general-purpose programmable devices, they were soon applied to control sequential and combinatorial logic in industrial processes. However these early computers required specialist programmers and stringent operating environmental control for temperature, cleanliness, and power quality. To meet these challenges the PLC was developed with several key attributes. It would tolerate the shop-floor environment, it would support discrete (bit-form) input and output in an easily extensible manner, it would not require years of training to use, and it would permit its operation to be monitored. Since many industrial processes have timescales easily addressed by millisecond response times, modern (fast, small, reliable) electronics greatly facilitate building reliable controllers, and performance could be traded off for reliability.

A. PROGRAMMING

PLC programs are typically written in a special application on a personal computer, then downloaded by a direct-connection cable or over a network to the PLC. The program is stored in the PLC either in battery-backed-up RAM or some other non-volatile flash memory. Often, a single PLC can be programmed to replace thousands of relays. Under the IEC 61131-3 standard, PLCs can be programmed using standards-based programming languages. The most commonly used programming language is Ladder diagram (LD) also known as Ladder logic. It uses Contact-Coil logic to make programs like an electrical control diagram. A graphical programming notation called Sequential Function Charts is available on certain programmable controllers. A model which emulated electromechanical control panel devices (such as the contact and coils of relays) which PLCs replaced. This model remains common today. IEC 61131-3 currently defines five programming languages for programmable control systems: function block diagram (FBD), ladder diagram (LD), structured text (ST; similar to the Pascal programming language), instruction list (IL; similar to assembly language), and sequential function chart (SFC). These techniques emphasize logical organization of operations. While the fundamental concepts of PLC programming are common to all manufacturers, differences in I/O addressing, memory organization, and instruction sets mean that PLC programs are never perfectly interchangeable between different makers. Even within the same product line of a single manufacturer, different models may not be directly compatible.

B. LADDER LOGIC

The most common method used for programming PLCs is based on the ladder diagrams. Writing a program is then equivalent or resembles to drawing a switching circuit. The ladder diagram consists of two vertical lines on either side representing the power rails which are positive and neutral. Circuits are connected in the rungs of the ladder which are horizontal, between these two rails. Ladder logic was originally a written method to document the design and construction of relay racks as used in manufacturing and process control. Each device in the relay rack would be represented by a symbol on the ladder diagram with connections between those devices shown. In addition, other items external to the relay rack such as pumps, heaters, and so forth would also be shown on the ladder diagram. Although the diagrams themselves have been used since the days when logic could only be implemented using switches and electromechanical relays, the term 'ladder logic' was only latterly adopted with

the advent of solid state programmable logic. Ladder logic acts as a programming language that represents a program in the form of graphical diagram based on the circuit diagrams of relay logic hardware and used in industrial control applications. The name Ladder Logic is appropriate as it resembles a ladder with two vertical rails on either side with a series of horizontally connected rungs between them. The system in the ladder diagram form will be programmed into the PLC. Once the programs have been downloaded into PLC, it can be monitored in the Diagram Workspace during execution. The Logo PLC provide the easy user interface to download the program, to upload the program, and to go back at online mode to see program desirable state. In this paper, the each home appliance is controlled by using PLC as shown in the flow charts.

Ladder diagrams are specialized schematics commonly used to document industrial control logic systems. They are called "ladder" diagrams because they resemble a ladder, with two vertical rails (supply power) and as many "rungs" (horizontal lines) as there are control circuits to represent. Ladder logic has evolved into a programming language that represents a program by a graphical diagram based on the circuit diagrams of relay logic hardware. Ladder logic is used to develop software for programmable logic controllers (PLCs) used in industrial control applications.

4.2.1 Rung

A type of line diagram that uses the input and output symbols used by PLC ladder logic. Line diagrams are converted to programming diagrams before being entered into a PLC. One of multiple horizontal programming lines in a ladder logic diagram. Each rung controls one output. Ladder logic was originally a written method to document the design and construction of relay racks as used in manufacturing and process control. Each device in the relay rack would be represented by a symbol on the ladder diagram with connections between those devices shown. In addition, other items external to the relay rack such as pumps, heaters, and so forth would also be shown on the ladder diagram. Ladder logic has evolved into a programming language that represents a program by a graphical diagram based on the circuit diagrams of relay logic hardware. Ladder logic is used to develop software for programmable logic controllers (PLCs) used in industrial control applications. The name is based on the observation that programs in this language resemble ladders, with two vertical rails and a series of horizontal rungs between them. While ladder diagrams were once the only available notation for recording programmable controller programs, today other forms are standardized in IEC 61131-3 (For example, as an alternative to the graphical ladder logic form, there is also a more assembly language like format called Instruction list within the IEC 61131-3 standard.). Ladder logic is widely used to program PLCs, where sequential control of a process or manufacturing operation is required. Ladder logic is useful for simple but critical control systems or for reworking old hardwired relay circuits. As programmable logic controllers became more sophisticated it has also been used in very complex automation systems. Often the ladder logic program is used in conjunction with an HMI program operating on a computer workstation. The motivation for representing sequential control logic in a ladder diagram was to allow factory engineers and technicians to develop software without additional training to learn a language such as FORTRAN or other general purpose computer language. Development, and maintenance, was simplified because of the resemblance to familiar relay hardware systems. Implementations of ladder logic have characteristics, such as sequential execution and support for control flow features, that make the analogy to hardware somewhat inaccurate. This argument has become less relevant given that most ladder logic programmers have a software background in more conventional programming languages.

Ladder logic can be thought of as a rule-based language rather than a procedural language. A "rung" in the ladder represents a rule. When implemented with relays and other electromechanical devices, the various rules "execute" simultaneously and immediately. When implemented in a programmable logic controller, the rules are typically executed sequentially by software, in a continuous loop (scan). By executing the loop fast enough, typically many times per second, the effect of simultaneous and immediate execution is achieved, if considering intervals greater than the "scan time" required to execute all the rungs of the program. Proper use of programmable controllers requires understanding the limitations of the execution order of rungs.

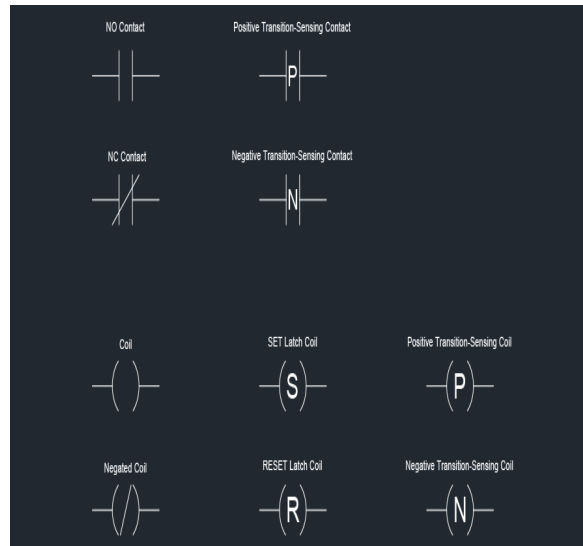


Figure 1. Ladder Logic Symbols

—[]— Normally open contact, closed whenever its corresponding coil or an input which controls it is energized. (Open contact at rest)

—[/]— Normally closed ("not") contact, closed whenever its corresponding coil or an input which controls it is not energized. (Closed contact at rest) Rung Output: Actuators (coils)

—()— Normally inactive coil, energized whenever its rung is closed. (Inactive at rest)

(/)— Normally active ("not") coil, energized whenever its rung is open. (Active at rest) The "coil" (output of a rung) may represent a physical output which operates some device elsewhere in the program. A way to recall these is to imagine the checkers (contacts) as a push button input, and the actuators (coils) as a light bulb output. The presence of a slash within the checkers or actuators would indicate the default state of the device at rest.

4.2.2 Basic ladder logic symbol

Normally open contact Passes power (ON) if coil driving the contact is ON (closed) Allen-Bradley calls it XIC- eXamine If Closed Normally closed contact Passes power (ON) if coil driving the contact is off (open) Allen-Bradley calls it XIO - examine If Open Output or coil if any left-to-right path of inputs passes power, output is energized Allen-Bradley calls it OTE – Output Energize Not Output or coil if any left-to-right path of inputs passes power, output is de-energized

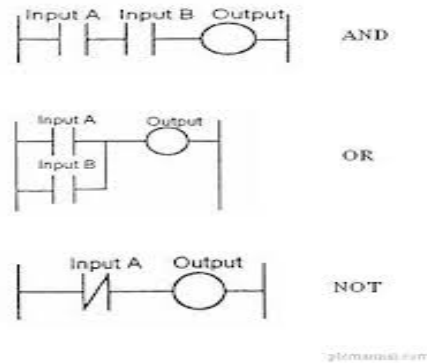


Figure 2. Ladder logic symbols for AND, OR and NOT gates

V. BECKHOFF AUTOMATION

Beckhoff Automation GmbH & Co. KG is a medium-sized, internationally acting company located in Verl, Westphalia, Germany. It was established in 1980 as a business unit of the ElektroBeckhoff GmbH, an electrical installation and retail company which was founded in 1953 by Arnold Beckhoff Senior. Until 2005 the business unit "New Automation Technology" operated under the name BeckhoffIndustrieElektronik (Beckhoff Industrial Electronics). In 2005 the company ElektroBeckhoff GmbH has been restructured into three companies:

- Beckhoff Automation GmbH & Co. KG (PC-based automation technology)
- ElektroBeckhoff GmbH (Building services engineering)
- Beckhoff Technique and Design GmbH (Retail of high-end entertainment electronics, light systems, home appliances)

A. TWINCAT3

- TwinCAT3 is a new era for PC based control software and is another milestone in the company history of the Beckhoff automation GmbH & co. KG.
- In terms of modular and flexible software architecture. Especially in the field of efficient engineering TwinCAT3 will set the pattern
- Nearly every kind of control applications is possible with TwinCAT3.
- The user can access different programming languages to realize such applications.
- Toolboxes are available with graphic languages editors or the visualization editor and provide graphic programming elements which can be inserted into the editor via drag & drop.
- In home automation layout the elements such as switches, indicators and textboxes are
- taken from the toolbox and their attributes are chosen from properties.

VI. TARIFF

The price at which electricity is sold is known as tariff for electricity and several factors come into play while determining the tariff. Normally the tariff of electricity is mentioned in rate per kilowatt hour of power consumed or rate per kWh.

The tariff calculation for the home appliances can be monitored via HMI. The PLC program for the tariff calculation is fed through Beckhoff TWINCAT 3 software.

A. TARIFF CALCULATION

The tariff is the rate at which the electrical energy is sold. There are various types of tariffs followed in the market. Electrical energy produced by the power system is delivered to a large number of customers. The tariff

becomes the attention for the electric supply company. The company has to ensure that the tariff such that it not only recovers total cost of producing electrical energy but also earns profit on the capital investment.

FORMULA

Cost = Σ (units * cost) //As per the above table

1 unit=1000w*1hour=1kwh

Electricity Unit = $(30 \times n \times W \times H) / 1000$

Where,

W = Energy Consumption Unit (watts)

H = Time (hours)

n = Number of Appliances

VII.OUTPUT SCREENSHOT

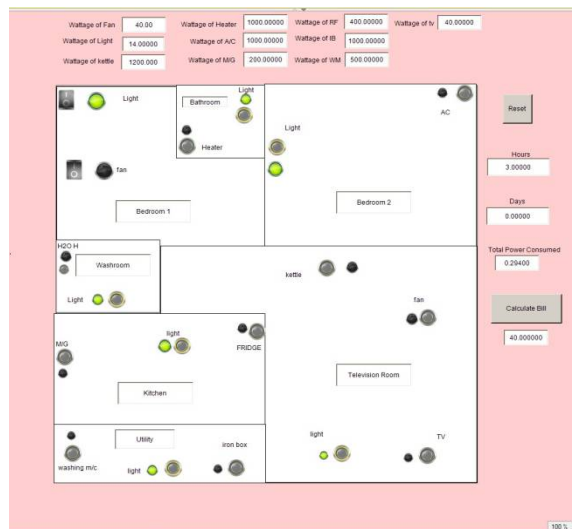


Figure 3. wattage and tariff calculation screen

VIII. ADVANTAGES

A. ENERGY EFFICIENCY

Increase your home's energy efficiency by remotely powering off systems and appliances when they aren't in use. In addition to the standard home automation products that give you active control, some products actively monitor systems and arm the homeowner with knowledge, insight and guidance to achieve greater control and energy efficiency. Comfort Guard is a 24-hour monitoring service for heating and cooling equipment that can predict and even prevent breakdowns. It also includes personalized monthly reports to help homeowners track equipment performance, reduce excess energy usage and improve efficiency.

B. SAVING

Home automation literally pays off. When you are able to use home systems and appliances only when needed, the savings will be apparent in the first utility bill. No more wasting money on lights left on when you aren't home, or spending money on gas to drive home because you forgot to lock the door. Monetary savings are apparent, but you'll also be saving time. No wasted trips home, no running through the house turning everything off, no time spent worrying about what was or wasn't turned off.

C. REMOTE CONTROL

A connected home product like the Wi-Fi Programmable Thermostat lets you conveniently adjust your home temperature from the mobile app so your family is always comfortable. Don't underestimate the power of being able to control your home's functions from a distance. On an exceptionally hot day, you can order your house to become cooler in just enough time before you

get home from work. If you're in a hurry to get dinner started but you're still at the store, you can have your oven start to preheat while you're still on your way home. You can even check to see if you left the lights on, who is at your front door, or make sure you turned off all your media while you're away

XI. APPLICATIONS

The uses of SCADA technology are vast. Many industries require the many monitoring and control capabilities that SCADA offers. In most uses, SCADA is used to manage a physical process (manufacturing and water processing are common). In other uses of the word "SCADA", a telecom or IT system of communications is being managed instead.

Here are a few of the most common uses for SCADA systems:

- Manufacturing
- Food Production
- Electric and Gas Utilities
- Waste Water Treatment
- Telecom and Information Technology

X. RESULTS AND DISCUSSION

The visualized appliances can be controlled from any remote location through HMI. The wattage for the running appliance per hour can be monitored. When the input is given i.e., the required appliances are switched ON the power consumed in units for overall appliances can be visualized. The tariff for the power units consumed by the running appliances can be monitored.

XI. CONCLUSION

When you incorporate security and surveillance features in your smart home network, your home security can skyrocket. There are tons of options here-only a few dozen of which are currently being explored. For example, home automation systems can connect motion detectors, surveillance cameras, automated door locks, and other tangible security measures throughout your home so you can activate them from one mobile device before heading to bed. You can also choose to receive security alerts on your various devices depending on the time of day an alert goes off. And monitor activities in real-time whether you're in the house or halfway around the globe.

REFERENCES

- [1] PLC principle and application by John W. Webb
- [2] Smart home automation's security system design implementation
- [3] <http://www.modicon.com/techpubs/toc7.htm>
- [4] <https://inductiveautomation.com/scada>
- [5] www.iraj.in/up_proc/pdf/54-13958134761-5.pdf
- [6] wikipedia.org/wiki/Programmable_logic_controller.
- [7] http://www.thelearningpit/TheAdvantages_of_Using_Programmable_Controllers_How.html.
- [8] An overview of continuous monitoring and control system for three phase induction motor based on Programmable Logical Control & SCADA Technology .volume 4,Issue4, July- August (2013) pp.188-1966