Radonix

cnc controller

User Guide

radonix.com

User Guide

Radonix CNC Controller

For router CNC machines

Prepared and written by: Sara Yarmohammadi, Shayan Turk

Headquarters

TÜRKİYE
RADONİX OTOMASYON SİSTEMLERİ VE TİC. LTD. ŞTİ

Halil Rıfat Paşa Mah. Yüzer Havuz Sok. Perpa Ticaret Merkezi B Blok No. 1668 Okmeydanı / İSTANBU
Telefon: +90-212-2229522| Fax: +90-212-22295

Chapter 1

Introduction

- Please read this book before using the product to ensure proper use of the controller and its related software.

This book provides information on how to install and operate Radonics PC-Pro LAN and PC-Smart controllers for router machines. Having basic knowledge of industrial electricity is essential to make use of this information. Information about the drive motors used in the machine and their wiring is also required as supplementary information for the necessary electrical panel setup.

By installing the interface, default settings are installed that can be used as a wiring template by accessing the program's settings. Inputs, outputs, and axes are present in these settings. However, based on the conditions, different wiring templates can be applied, and by changing the program's settings, the hardware can be synchronized with the software.

Radonics offers several interfaces for router machine startup, with the XYZ-Router interface for simple 3-axis routers and the XYZA-Router interface for 4-axis rotary routers being the most commonly used examples. By installing them on the computer, pre-configured settings for inputs and outputs are installed.

Who can benefit from using this book?

- This book is intended for the following users
- People responsible for installing or wiring Radonix controllers.
- People responsible for setting up Radonix controllers.
- People responsible for setting up Radonix controller software.
- People responsible for troubleshooting and maintaining CNC machines with Radonix controllers.
- People who intend to present a project based on CNC machines.

- Indicates a hazardous situation that, if not avoided, will result in serious injury or death.

- Indicates a hazardous situation that, if not avoided, could result in minor or serious injury to the controller.death.

- Indicates an operation that is not recommended and may cause a disruption in the controller's performance.

Wiring

- Do not connect the controller power supply to the city power or 220V, incorrect connection can cause serious damage to the controller.

- To prevent noise, use shielded cable and foil for connecting to the encoders of the drives.
- Before connecting the controller to the power supply, check the wiring again and make sure the connections are correct. n

- Make sure to correctly ground the electrical panel and computer case, otherwise the generated noise can cause incorrect operation of the computer and software.
- Pay attention to the wire size during wiring, otherwise the resistance created in the wire can cause incorrect data transmission.

Software

- Before using the controller and software, set the software parameters correctly, otherwise it may cause serious damage to the mechanical system.

Maintenance and Repair

- Please do not touch the controller while using it as it may cause an electric shock.
- Only people with electronics knowledge should be responsible for maintenance and repair.

- To facilitate easier understanding of the text, some commonly used terms that may need explanation are briefly described below:

Interface

The main interface of the Radonix Cam-Pro program with the user, which is actually what the user sees on the computer and communicates with the controller through.

Variable

All numerical and string data available to the user to create changes in the user interface display, motion, and response are called variables.

Function

Programs that allow the controller and/or the user to execute specific commands and instructions, and enable communication between the controller and the user.

Parameter

Refers to internal variables of the controller and/or interface that allow the user to create changes in the program execution process and/or changes in the motion of the system.

Key

Can include both hardware and software keys, which can be defined based on the type of application.

Chapter 2

Checking the packaging and describing the types of models

User Guide

Check Package

Please check the following items after receiving the package:
Make sure that the product is the same as what you have ordered, and the serial number on the controller matches.

Make sure that the controller is not damaged and the controller board is not broken or any piece is damaged or detached.

If you notice any damage, please inform the sales representative or Radonix company before starting up the controller.

A complete and executable controller package should include the following:

Radonix PC-Pro LAN 2A

1. Controller
2. DB15 male connector (2 pieces)
3. DB15 connector cover (2 pieces)
4. Phoenix 3-pin terminal (1 piece)
5. Phoenix 6-pin terminal (1 piece)
6. Phoenix 8-pin terminal (3 pieces)
7. Phoenix 9-pin terminal (1 piece)
8. Radonix software CD

Radonix PC-Pro LAN 4A

1. Controller
2. DB15 male connector (4 pieces)
3. DB15 connector cover (4 pieces)
4. Phoenix 3-pin terminal (1 piece)
5. Phoenix 6-pin terminal (1 piece)
6. Phoenix 8-pin terminal (5 pieces)
7. Phoenix 9-pin terminal (1 piece)
8. Radonix software CD

Radonix PC-Pro LAN 6A

1. Controller
2. DB15 male connector (6 pieces)
3. DB15 connector cover (6 pieces)
4. Phoenix 3-pin terminal (1 piece)
5. Phoenix 6-pin terminal (1 piece)
6. Phoenix 8-pin terminal (8 pieces)
7. Phoenix 9-pin terminal (1 piece)
8. Radonix software CD

Radonix PC-Pro LAN 3AS

1. Controller
2. Phoenix 3-pin terminal (1 piece)
3. Phoenix 4-pin terminal (4 pieces)
4. Phoenix 6-pin terminal (1 piece)
5. Phoenix 8-pin terminal (1 piece)
6. Radonix software CD

Radonix PC-Smart 4A

1. Controller
2. DB15 male connector (4 pieces)
3. DB15 connector cover (4 pieces)
4. Phoenix 3-pin terminal (2 pieces)
5. Phoenix 4-pin terminal (2 pieces)
6. Phoenix 5-pin terminal (4 pieces)
7. Phoenix 6-pin terminal (1 piece)
8. Radonix software CD

Radonix PC-Smart 3AS

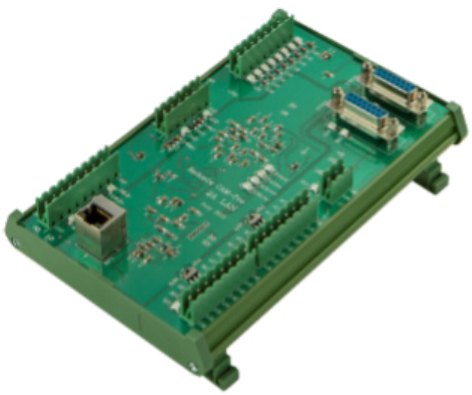
1. Controller
2. Phoenix 3-pin terminal (2 pieces)
3. Phoenix 4-pin terminal (4 pieces)
4. Phoenix 5-pin terminal (3 pieces)
5. Radonix software CD

Controller Models Features

Radonix PC-Pro LAN 2A

Flexible High Performance Position Control

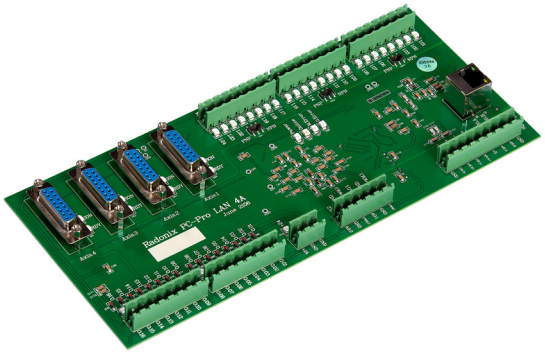
Axes	2 independent and interpolated axes
Digital Inputs	16 Isolated NPN inputs
Digital Outputs	8 (Negative protected outputs)
Analog Outputs	2 (Protected outputs)
PWM Outputs	2 PWM outputs with adjustable frequency
Axis Pulse Rate	500,000 pulses per second
Axis Pulse Type	Directional pulse
Acceleration Time	From 50 mm/s to 30,000 mm/s
Speed Profile	S-Curve
Hardware Buffer Size	2,000 block FIFO
PC-Controller Data exchange time	20 milliseconds
Isolation Type	Optocoupler
Communication Type	LAN with 100 Mbps transfer rate (TCP/IP)
Communication Length	More than 20 meters with UTP/More than 50 meters with SFUTP
Hardware Lock	24 time locks with internal clock
Power Consumption	16 to 32 volts - 300 milliamperes
Dimensions	19*13 centimeters
Controller Type	Loop-open control
Operating System	Windows 7, 8, 10, 11
Supported Equipment	Joystick, Remote control, Handwheel
Active Axis Count	2



Radonix PC-Pro LAN 4A

Flexible High Performance Position Control

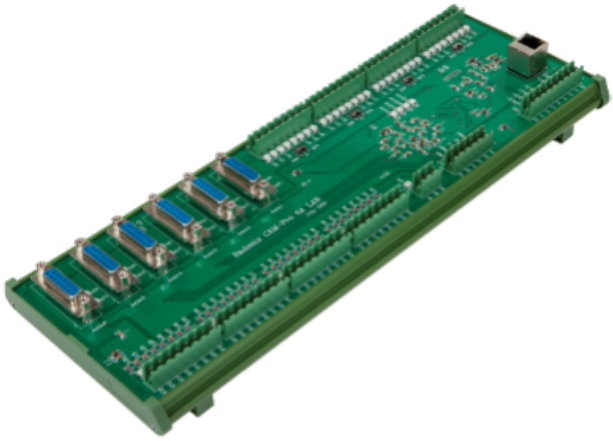
Axes	4 independent and interpolated axes
Digital Inputs	24 Isolated NPN inputs
Digital Outputs	16 open collector outputs with short circuit protection circuit
Analog Outputs	2
PWM Outputs	2 PWM outputs with adjustable frequency
Axis Pulse Rate	500,000 pulses per second
Axis Pulse Type	Directional pulse
Acceleration Time	From 50 mm/s to 30,000 mm/s
Speed Profile	S-Curve
Hardware Buffer Size	2,000 block FIFO
PC-Controller Data exchange time	20 milliseconds
Isolation Type	Optocoupler
Communication Type	LAN with 100 Mbps transfer rate (TCP/IP)
Communication Length	More than 20 meters with UTP/More than 50 meters with SFUTP
Hardware Lock	24 time locks with internal clock
Power Consumption	16 to 32 volts - 300 milliamperes
Dimensions	25*13 centimeters
Controller Type	Loop-open control
Operating System	Windows 7, 8, 10, 11
Supported Equipment	Joystick, Remote control, Handwheel
Active Axis Count	4,3,2



Radonix PC-Pro LAN 6A

Flexible High Performance Position Control

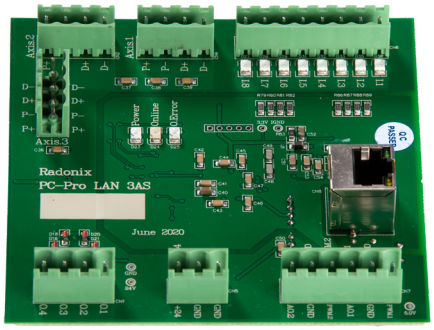
Axes	6 independent and interpolated axes
Digital Inputs	32 PNP and NPN inputs
Digital Outputs	32 (Negative protected outputs)
Analog Outputs	2
PWM Outputs	2 PWM outputs with adjustable frequency
Axis Pulse Rate	500,000 pulses per second
Axis Pulse Type	Directional pulse
Acceleration Time	From 50 mm/s to 30,000 mm/s
Speed Profile	S-Curve
Hardware Buffer Size	2,000 block FIFO
PC-Controller Data exchange time	20 milliseconds
Isolation Type	Optocoupler
Communication Type	LAN with 100 Mbps transfer rate (TCP/IP)
Communication Length	More than 20 meters with UTP/More than 50 meters with SFUTP
Hardware Lock	24 time locks with internal clock
Power Consumption	16 to 32 volts - 300 milliamperes
Dimensions	34*13 centimeters
Controller Type	Loop-open control
Operating System	Windows 7, 8, 10, 11
Supported Equipment	Joystick, Remote control, Handwheel
Active Axis Count	6,5,4,3,2



Radonix PC-Pro LAN 3AS

Smart Position Control - Suitable for steppers

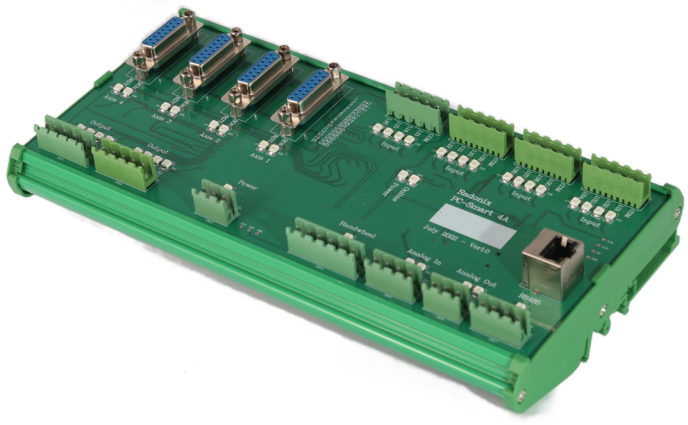
Axes	3 independent and interpolated axes
Digital Inputs	8 Isolated NPN inputs
Digital Outputs	4 (Negative protected outputs)
Analog Outputs	2
PWM Outputs	2 PWM outputs with adjustable frequency
Axis Pulse Rate	500,000 pulses per second
Axis Pulse Type	Directional pulse
Acceleration Time	From 50 mm/s to 30,000 mm/s
Speed Profile	S-Curve
Hardware Buffer Size	2,000 block FIFO
PC-Controller Data exchange time	20 milliseconds
Isolation Type	Optocoupler
Communication Type	LAN with 100 Mbps transfer rate (TCP/IP)
Communication Length	More than 20 meters with UTP/More than 50 meters with SFUTP
Hardware Lock	24 time locks with internal clock
Power Consumption	16 to 32 volts - 200 milliamperes
Dimensions	11*13 centimeters
Controller Type	Loop-open control
Operating System	Windows 7, 8, 10, 11
Supported Equipment	Joystick, Remote control, Handwheel
Active Axis Count	3,2



Radonix PC-Smart 4A

Smart Position Control - Suitable for steppers

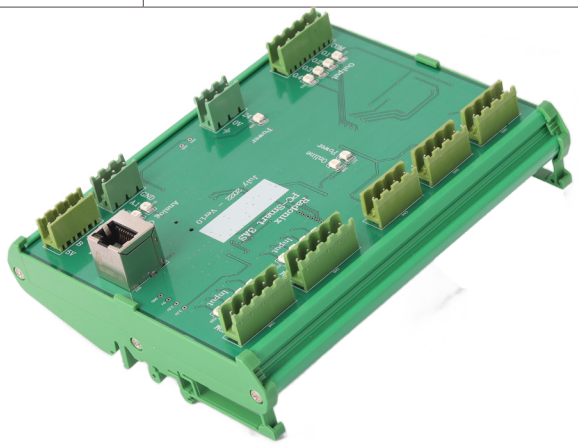
Axes	4 independent and interpolated axes
Digital Inputs	16 PNP and NPN inputs
Digital Outputs	8 PNP and NPN output
Analog Outputs	2 (10-0 Volts)
Analog Inputs	2 (10-0 Volts)
Axis Pulse Rate	500,000 pulses per second
Axis Pulse Type	Directional pulse
Acceleration Time	From 50 mm/s to 30,000 mm/s
Speed Profile	S-Curve
Hardware Buffer Size	2,000 block FIFO
PC-Controller Data exchange time	20 milliseconds
Isolation Type	Optocoupler
Communication Type	LAN with 100 Mbps transfer rate (TCP/IP)
Communication Length	More than 20 meters with UTP/More than 50 meters with SFUTP
Hardware Lock	24 time locks with internal clock
Power Consumption	12 to 24 volts - 300 milliamperes
Dimensions	25*13 centimeters
Controller Type	Loop-open control
Operating System	Windows 7, 8, 10, 11
Supported Equipment	Joystick, Remote control, Handwheel
Active Axis Count	4,3,2



Radonix PC-Smart 3AS

Smart Position Control - Suitable for steppers

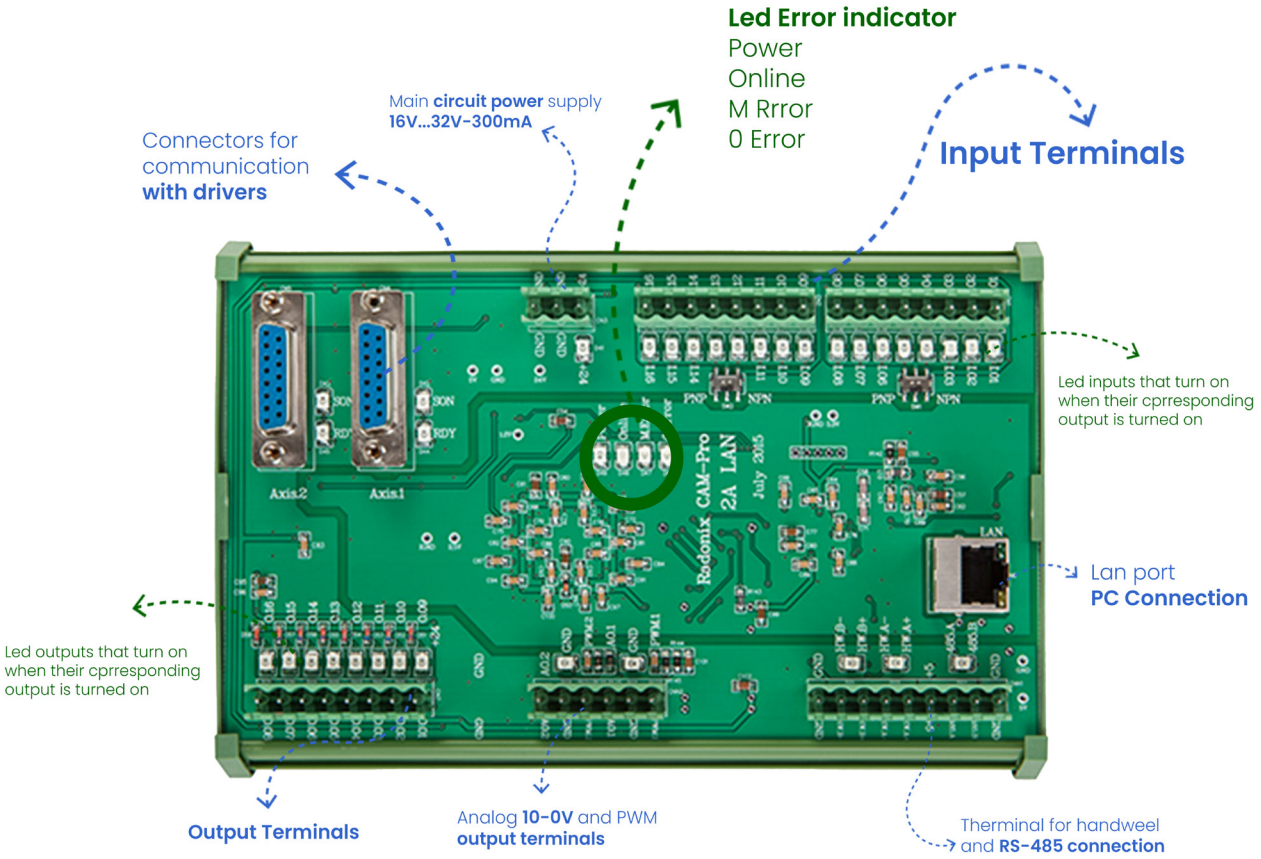
Axes	3 independent and interpolated axes
Digital Inputs	8 PNP and NPN inputs
Digital Outputs	4 PNP and NPN output
Analog Outputs	2 (10-0 Volts)
Axis Pulse Rate	100,000 pulses per second
Axis Pulse Type	Directional pulse
Acceleration Time	From 50 mm/s to 30,000 mm/s
Speed Profile	S-Curve
Hardware Buffer Size	2,000 block FIFO
PC-Controller Data exchange time	20 milliseconds
Isolation Type	Optocoupler
Communication Type	LAN with 100 Mbps transfer rate (TCP/IP)
Communication Length	More than 20 meters with UTP/More than 50 meters with SFUTP
Hardware Lock	24 time locks with internal clock
Power Consumption	12 to 24 volts - 300 milliamperes
Dimensions	15*13 centimeters
Controller Type	Loop-open controller
Operating System	Windows 7, 8, 10, 11
Supported Equipment	Joystick, Remote control, Handwheel
Active Axis Count	3,2



Identifying Different Parts of the Board

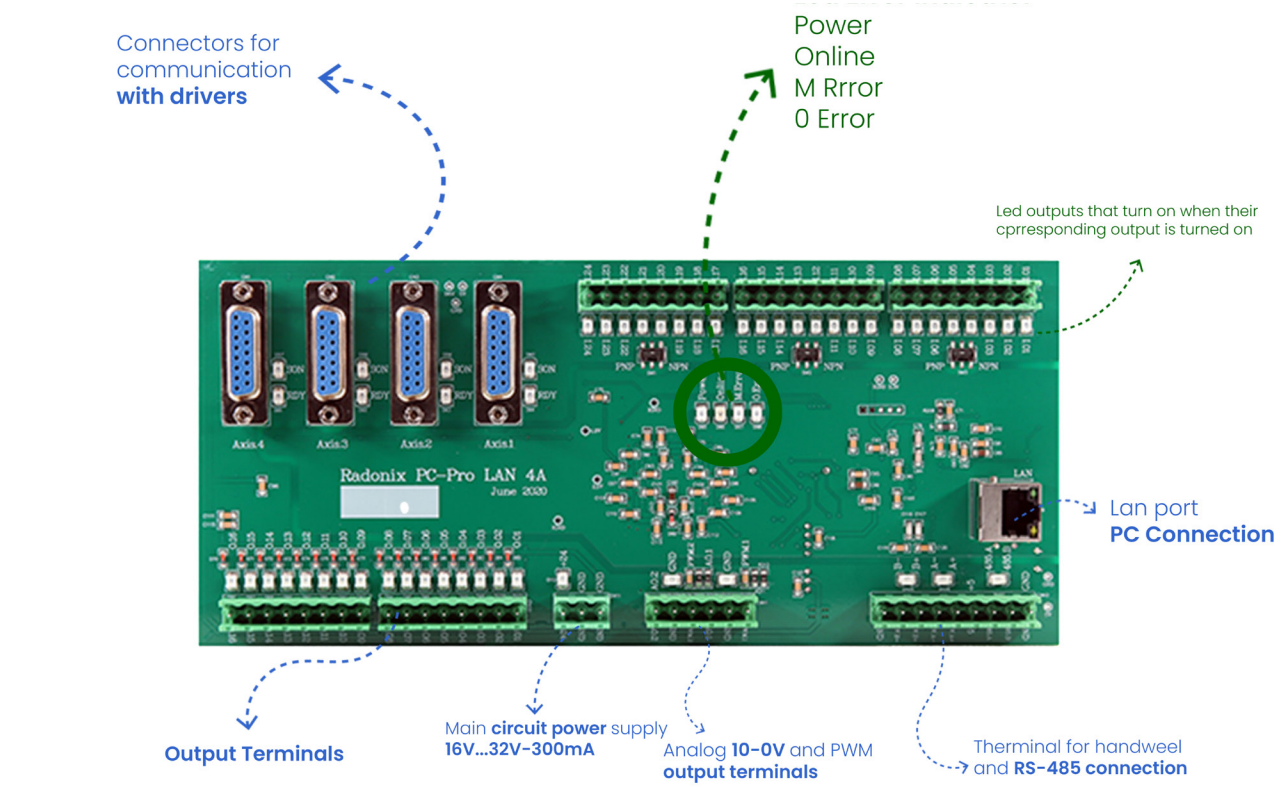
Radonix PC-Pro LAN 2A

Flexible High Performance Position Control



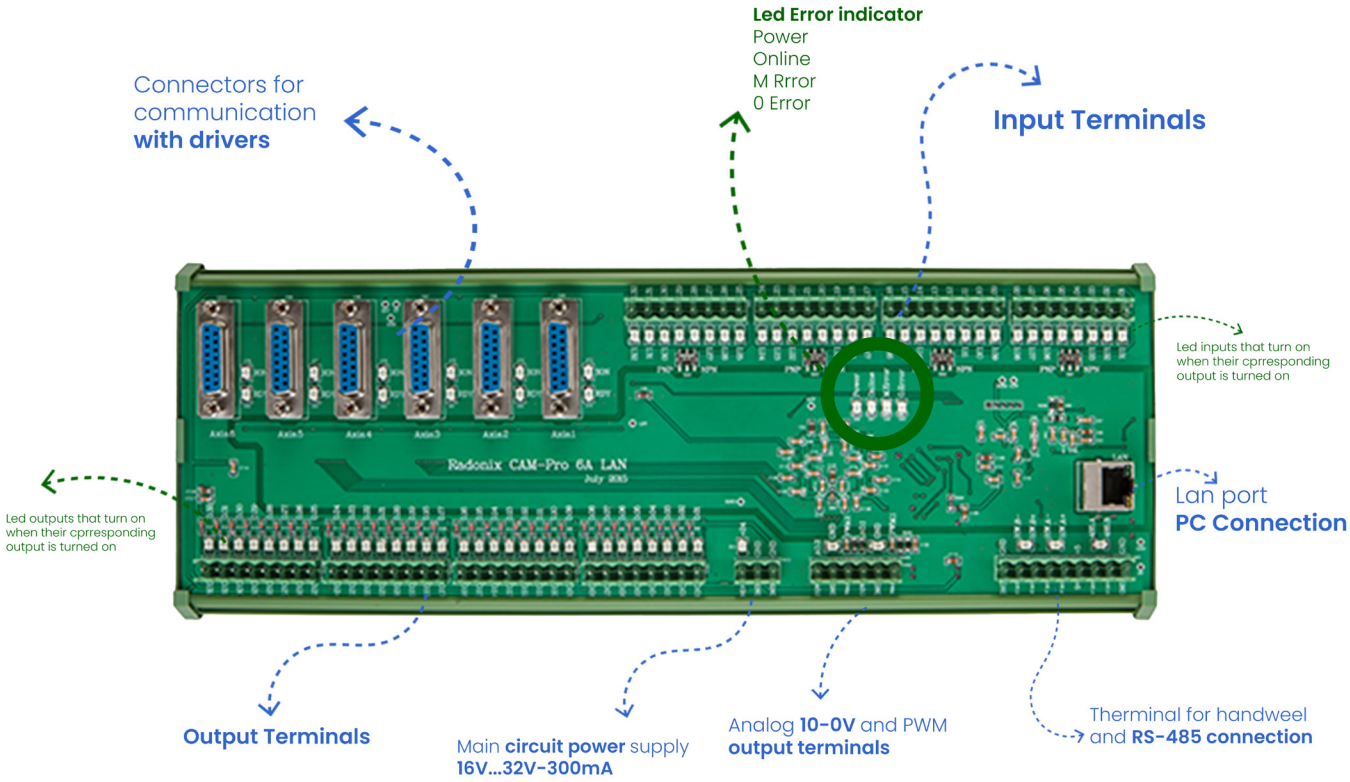
Radonix PC-Pro LAN 4A

Flexible High Performance Position Control



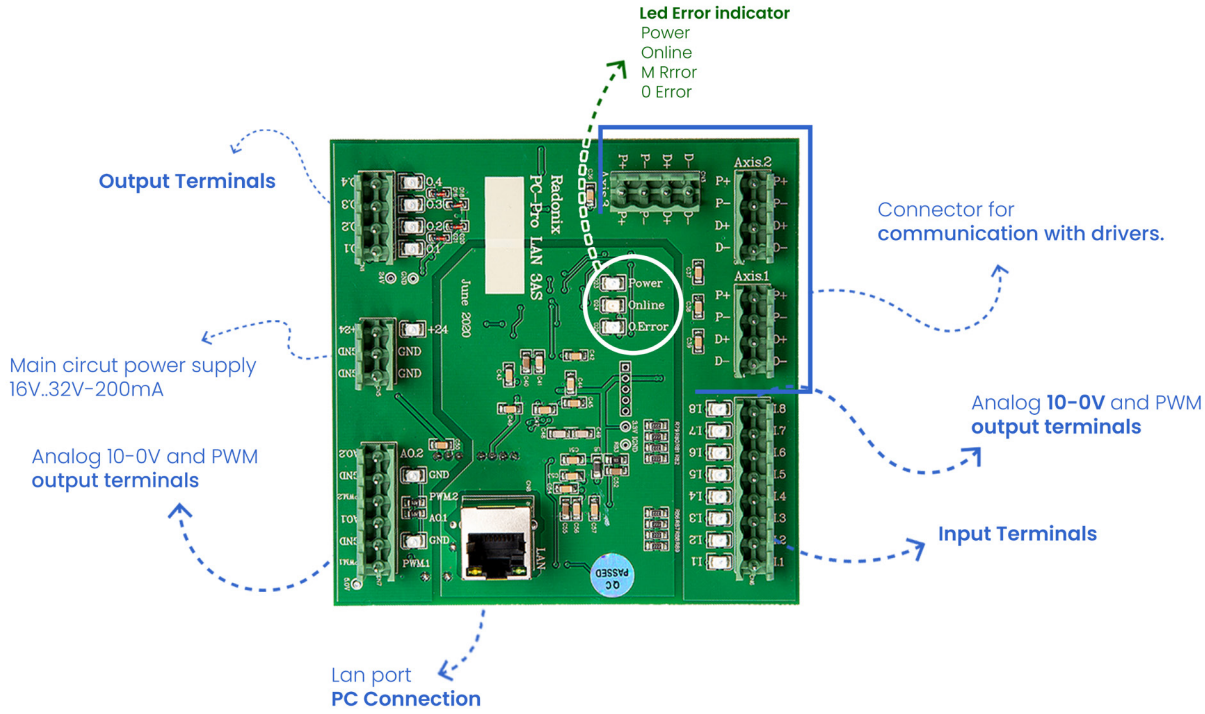
Radonix PC-Pro LAN 6A

Flexible High Performance Position Control



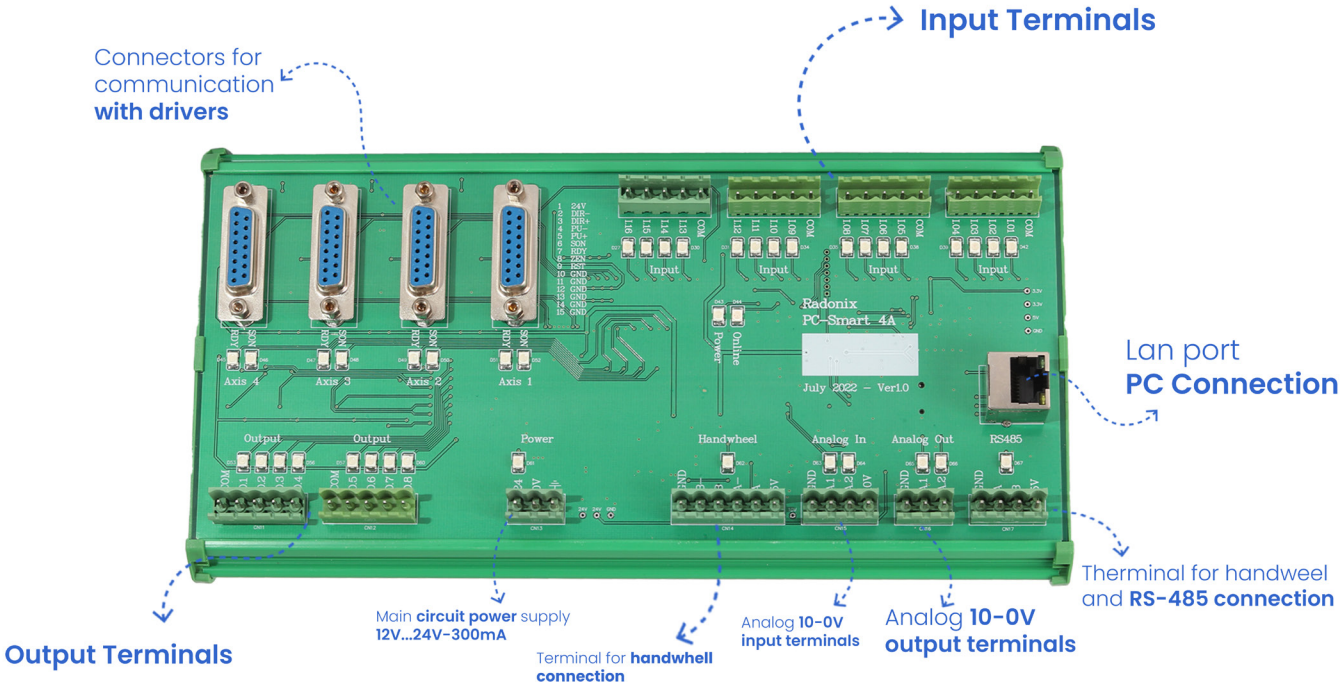
Radonix PC-Pro LAN 3AS

Flexible High Performance Position Control



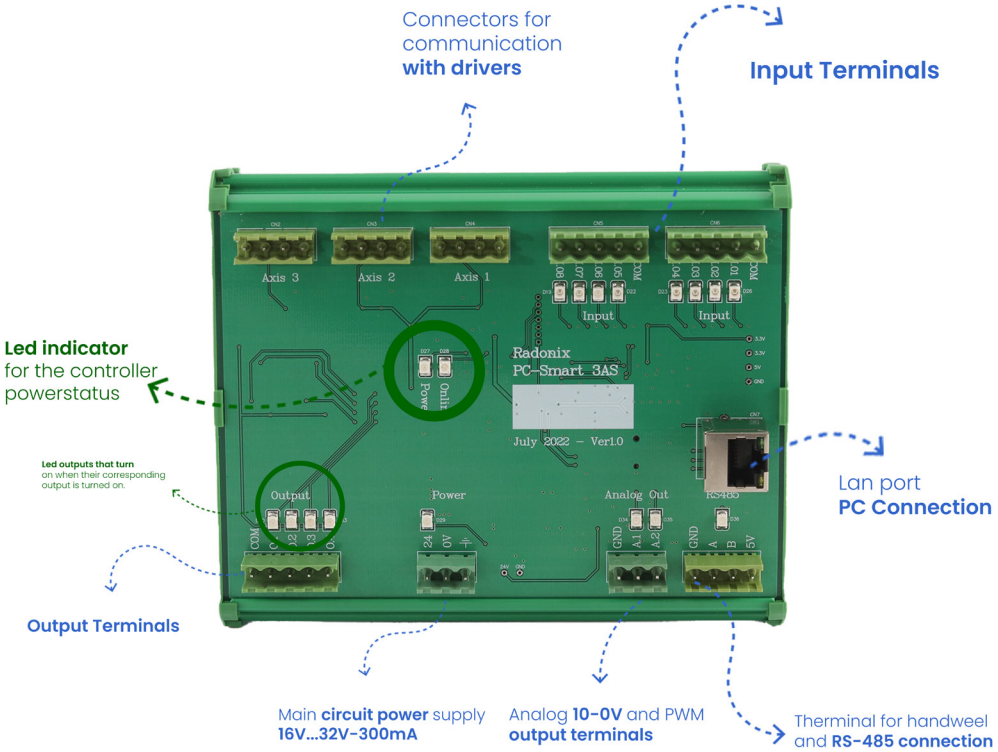
Radonix PC-Smart 4A

Flexible High Performance Position Control



Radonix PC-Smart 3AS

Flexible High Performance Position Control



Chapter 3

Installing the Controller in the Electrical Panel and Wiring

Installation of the Radonix Controller in the Electrical Penel

The Radonix controllers have a strong plastic protective cover that is supported by a base. They have been designed to be installed on the rails of the electrical panel.

Location of the Controller

The location of the controller is crucial in relation to the other elements inside the electrical panel. Inverters, drivers, and high-voltage cables can generate noise in the electrical panel. Therefore, it is important to place the controller at a sufficient distance from these sources, particularly the path of the communication cable to the computer. The controller should be at least 5 cm away from the other components (such as ducts and cables inside the panel) on each side. (Refer to Figure 1)

Switchboard Design

It is recommended that the switchboard designer allocate a separate area for the low-voltage elements and the controller, and respect the distance between the high-voltage components of the panel. The high-voltage cables should have the shortest path in the panel and should not pass near low-voltage elements or be at a greater distance from these elements.

Wiring and internal circuit of the Controller

Table (1) summarizes the internal circuitry of the Radonix controller board.

Terminals	Terminal Identification	Description
Ethernet connection	Computer Connection	To communicate with a computer under TCP/IP protocol, please refer to section 2.2.1 for more information.
DC24V, DC0V	Power Supply	For the main power supply of the controller, please refer to section 2.2.2 for more information.
Input connector	Digital Inputs	For communication with external controllers, please refer to section 2.2.3 for more information.
Output connector	Digital Outputs	For communication with external controlled devices, please refer to section 2.2.4 for more information.
Output Analog & PWM	Analog and PWM Outputs	For communication with external controlled devices, these outputs create a continuous value between 0 and 10 volts. Please refer to section 2.2.5 for more information.
Axis	Axis	For communication between the controller's axes and servo and stepper motors, please refer to section 2.2.7 for more information.
Handwheel	Handwheel	It is designed for communication with a handwheel, please refer to section 2.2.7 for more information.
RS485 communication connector	RS485	For communication with RS_485, please refer to section 2.2.7 for more information.

Table (1) - Wiring and Internal Circuit of the Controller

Communication with a Computer

The PC-Pro-Lan and PC-Smart series Radonix controllers communicate with a computer using the TCP/IP protocol and are connected through an Ethernet connection in a standard manner. (Refer to Figure 2) Standard CAT5 and CAT6 cables, up to 30 meters in length, can be used for communication. The wiring on both sides of the cables is depicted in Figure 3.

- Keep in Mind that it is advisable for the path of the communication cable to not be in close proximity to high voltage cables and to not be exposed to pressure or impact along its route.

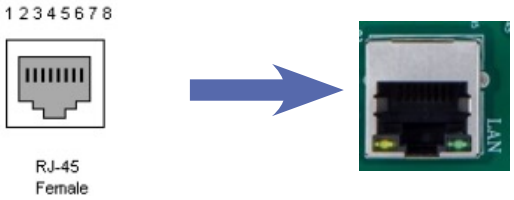


Figure (2) - LAN cable

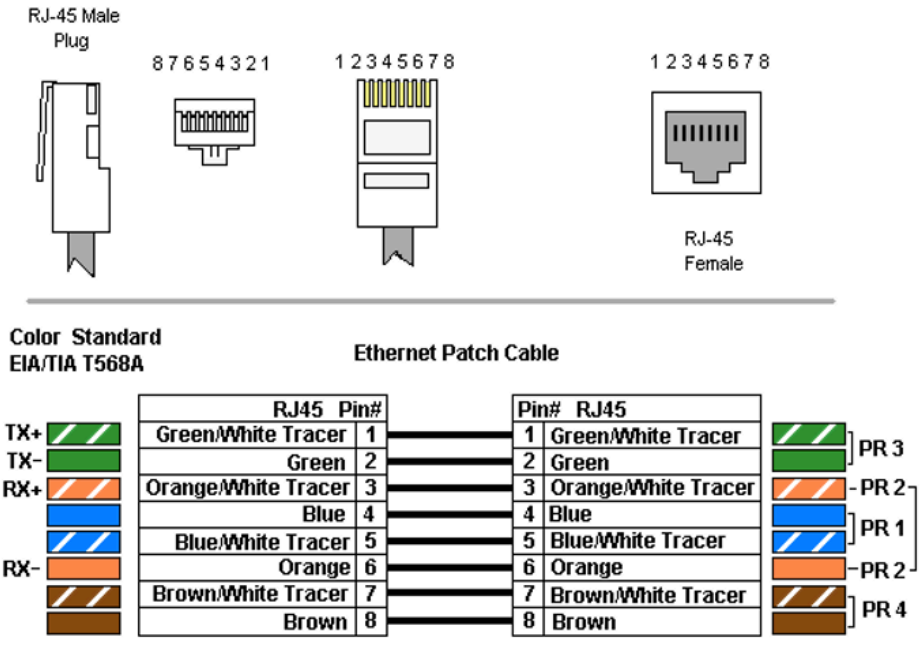
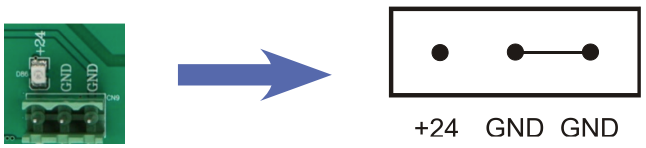


Figure (3) - Wiring of LAN cable

Power Supply

Power Supply The Radonix controllers are connected to the power supply via a 3-pin terminal. The positive terminal (24+) is connected to the positive pole of the power source, and the negative terminal (GND) is connected to the negative pole. (Refer to Figure 4 for the GND connection)

PC-Pro Models:



PC-Smart Models:

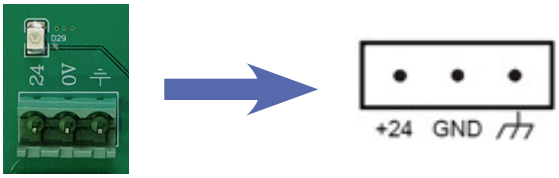
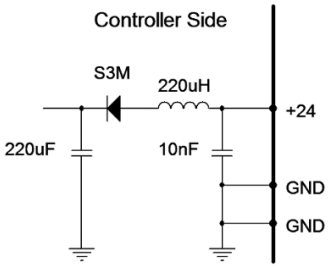


Figure (4) - Power Supply Connector

- Power Connector The positive pole of the controller is protected by a diode, which prevents damage to the controller if the poles are switched. The electronic circuit diagram of the controller is depicted in Figure 5.

Schematic of Power Supply in PC-Pro Models:



Schematic of Power Supply in PC-Smart Models:

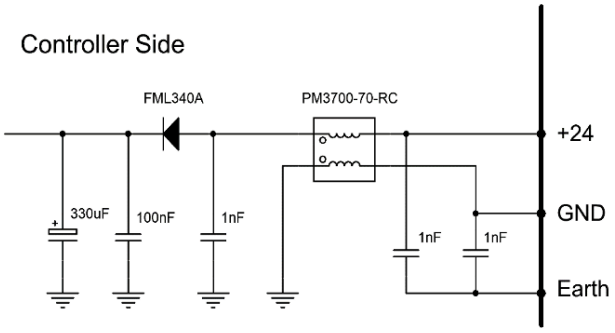


Figure (5) - Schematic of the Electronic Power Supply Circuit

The power supply voltage for PC-Pro models should be between 18 and 28 volts, and for PC-Smart models it should be between 12 and 24 volts, requiring a current of fewer than 0.5 amps. Switching power supplies are the most suitable type for the Radonix controllers because they are not significantly affected by changes in network voltage. It is recommended to use 24V switching power supplies to power the Radonix controllers. Calculating Power Supply Current To determine the required power supply current in the electrical panel, the current consumption of each element connected to the power supply must be calculated. For example, if there are 4 relays, 3 pneumatic solenoid valves, and a controller in a switchboard, the current consumption can be determined using Ohm's law (V=I*R).

The electric currents should be calculated and their sum considered as the current of the power supply, along with a confidence factor of a few percent. For example, if the current consumption of the relays is 0.1A and that of the solenoid valves is 0.25A, the total current can be calculated as follows:
Total current = controller current + relay current + solenoid valve current IT = 0.5 + 4 * 0.1 + 3 * 0.25 = 1.65A

Therefore, by choosing a 2A pow`er supply, a good reliability factor will be maintained.

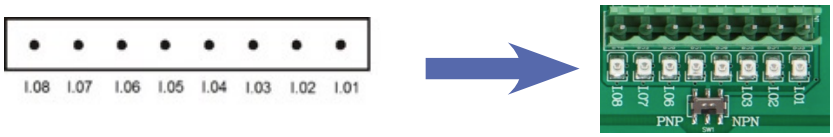
Power Consumption of Significant Elements

It is important to calculate the current consumption of elements that impose a significant load on the power supply. Elements such as brakes, servo motors, and pneumatic valves are examples of such elements. The power consumption of these elements can be calculated using an ammeter or by using Ohm's law (V=I*R). The ohmic resistance of each element can be measured using an ohmmeter and then, considering the source voltage, the equivalent current consumed by each element (I=24R) can be calculated.

Digital Inputs

The digital inputs of Radonix controllers are named as I.[n], where n is a number greater than zero and represents the input number. These inputs are isolated by optical couplers and have low noise tolerance due to their low impedance. (Figure 6)

PC-Pro Models:

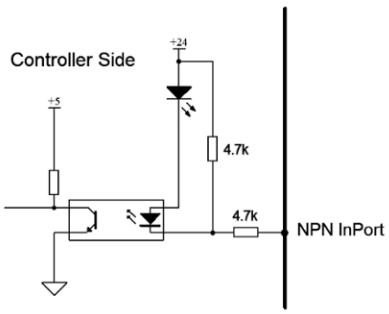


PC-Smart Models:

Figure (6) - Digital Inputs

The digital inputs on Radonix controllers have the capability to switch between NPN and PNP modes depending on the output of the device or sensor. The mode changeover switch must be set to PNP position for devices with PNP output and NPN position for devices with NPN output. The internal circuit schematics for both NPN and PNP modes on PC-Pro and PC-Smart models can be seen in Figures 7 and 8.

Digital input schematic in NPN mode for PC-Pro models:



Digital input schematic in PNP mode for PC-Pro models:

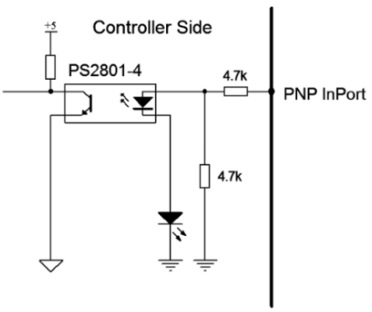


Figure (7) - Schematic of Digital Inputs in NPN and PNP Mode in PC-Pro Models

Digital Input Schematic in NPN and PNP modes in PC-Smart models:

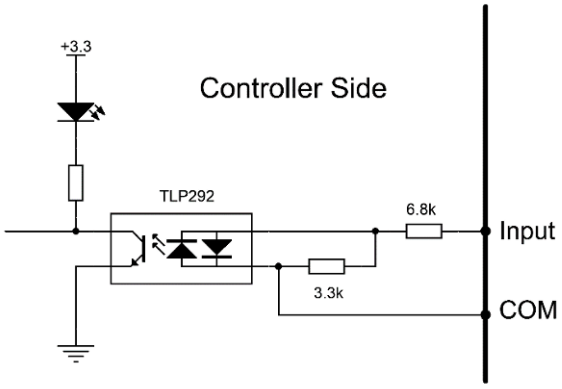


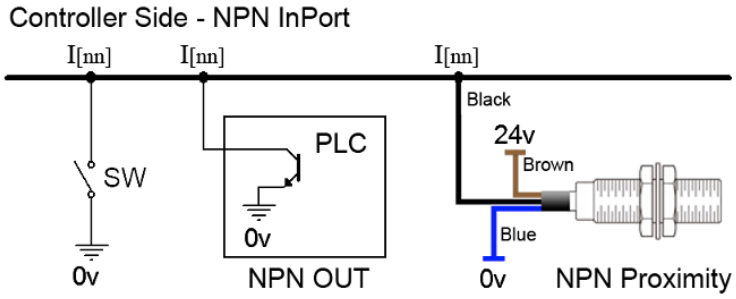
Figure (8) - Schematic of digital inputs in NPN and PNP modes in PC-Smart models.

In various branches of industrial automation, there are equivalent terms for PNP and NPN. Some examples of these terms include:

NPN = Sink = Low Active
PNP = Source = High Active

Is done by connecting the positive terminal of the device to the input terminal and the negative terminal to the common terminal (GND). In PNP mode, the connection is done by connecting the negative terminal of the device to the input terminal and the positive terminal to the positive power supply (+Vcc). The wiring diagrams for these connections are shown in Figures 9 and 10.

Connection of digital inputs with switchboard elements in NPN mode in PC-Pro models:



Connection of digital inputs with switchboard elements in PNP mode in PC-Pro models:

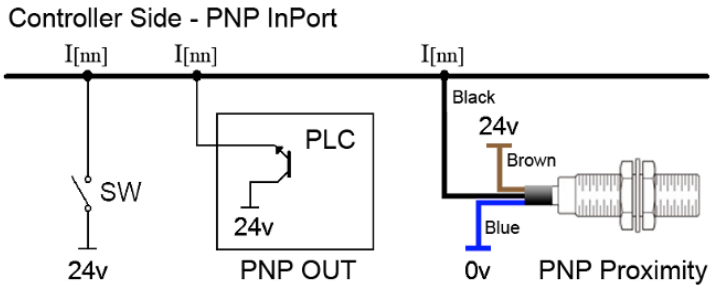
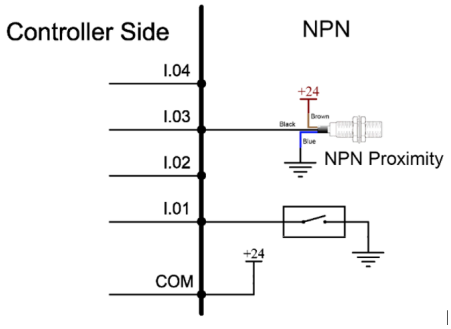


Figure (9) - Connection of digital inputs with electrical panel elements in NPN and PNP mode in PC-Pro models

Connection of digital inputs with electrical panel elements in NPN mode in PC-Smart models



Connection of digital inputs with electrical panel elements in PNP mode in PC-Smart models

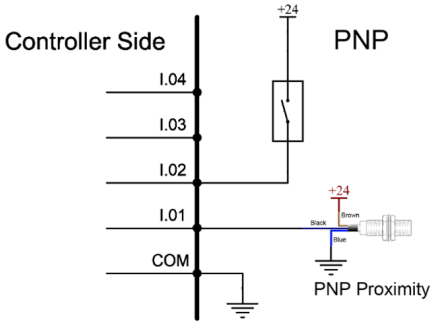
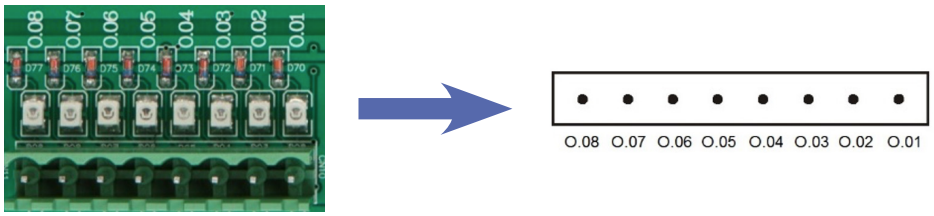


Figure (10) - Connection of digital inputs with electrical panel elements in NPN and PNP mode in PC-Smart models

Digital Outputs

The digital outputs of radial Radonix controllers are named O.[n], where n is a numerical identifier greater than zero and represents the output number (Figure 11). These outputs are protected in PC-Pro models against short-circuits and heat generated by high currents. If any of these events occur, an O-Error error is indicated on the board by the red LED, and all outputs are disabled by the controller. In PC-Smart models, the outputs are relays with a 1-amp contact and 24-volt bipolar zener diodes are taken into consideration for protection against overloading.

PC-Pro Models:



PC-Smart Models:

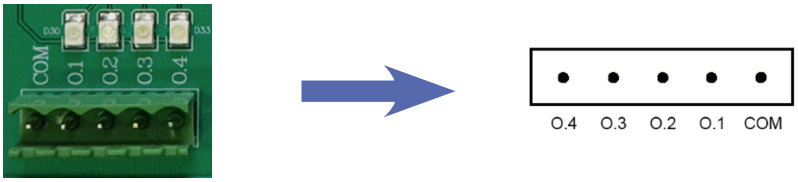


Figure (11) – Digital Outputs

- In PC-Pro models, under no circumstances should the outputs be connected to each other for high current flow or to create circuit paths, as the controller considers this as an error and will immediately turn off all outputs.

The maximum current output for PC-Pro models is around mA300, and for PC-Smart models is approximately 1A. Therefore, if we plan to set up a device with a higher current, we must use a relay.
The digital outputs of the Radonix controllers in PC-Pro models are all NPN, which means they are Low Active or Sink, so when activated, the output is connected to zero voltage or GND.
The electrical schematic of the Radonix controller outputs is shown in Figure 12

The internal circuit of the digital outputs in PC-Pro models:

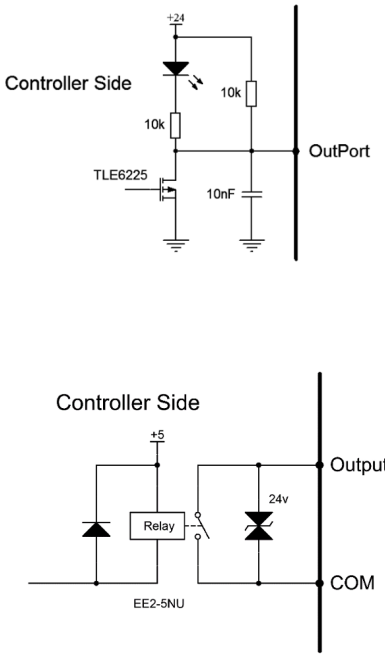
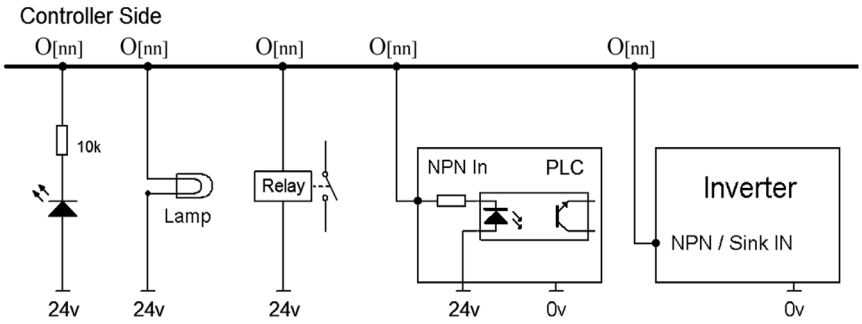


Figure (12) - Internal circuit of digital outputs.

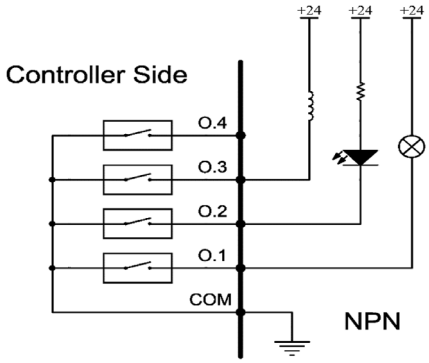
In the connection to relays, a diode is not needed to eliminate back current. The way of connecting the outputs to multiple devices is shown in Figure 13.

- Please note that all peripheral devices such as PLCs and inverters must be connected to a common or mutually powered ground in the connection. In fact, even if multiple power supplies are used, the ground of the power supplies must be connected to each other.

Connection of digital outputs with electrical panel elements in PC-Pro models:



Connection of digital outputs with electrical panel elements in NPN mode in PC-Smart models:



Connection of digital outputs with electrical panel elements in PNP mode in PC-Smart models:

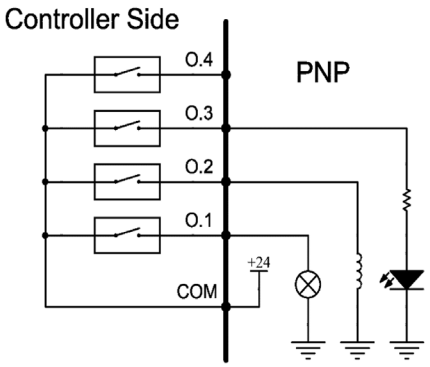
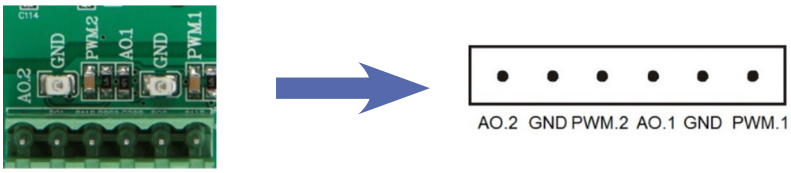


Figure (13) - Connection of digital outputs with electrical panel elements

Analog and Bandwidth Modulation Outputs

Radonix controllers in PC-Pro models have analog and PWM outputs which are shown as AO.[n] and PWM.[n] respectively, while PC-Smart models only have analog outputs shown as A[n], where n is a number greater than zero.
The analog outputs produce a continuous value between 0 and 10 volts. (Figure 14)

PC-Pro models:



PC-Smart Models:

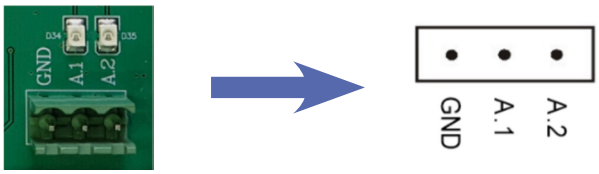
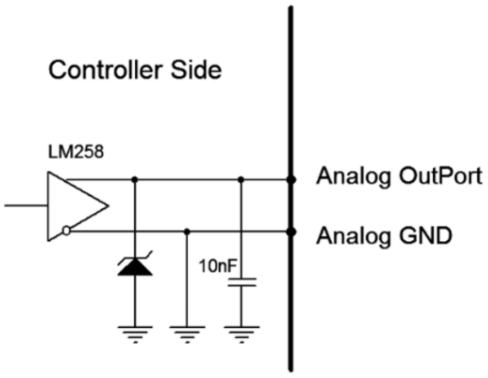


Figure (14) - Analog and PWM outputs

- These outputs are protected against a short connection to the negative power supply terminal and grounded electricity, but a direct connection to a voltage higher than 10 volts will cause damage to them.

The electrical circuit diagram of the analog outputs is shown in figure (15).

Electronic circuit schematic of analog outputs in PC-Pro models:



Electronic circuit schematic of analog outputs in PC-Smart models:

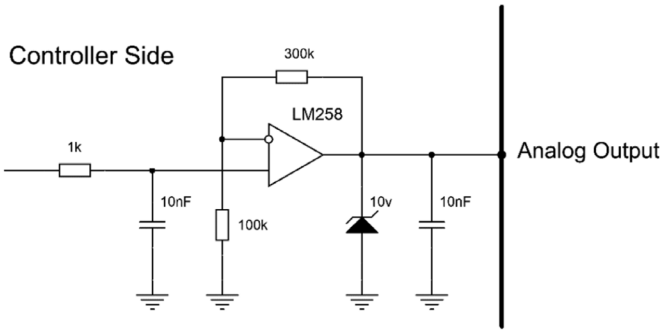


Figure (15) - Electronic Circuit Diagram of Analog Outputs

How to connect the analog outputs is shown in Figure 16.

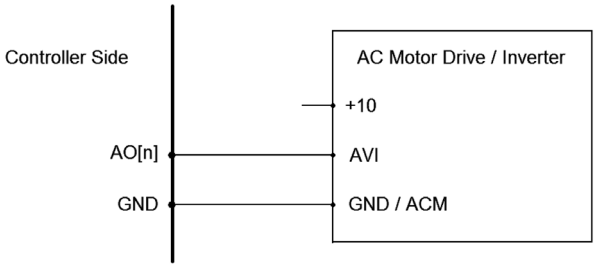


Figure (16) – Analog Output Connection

- Please note that the analog inputs and outputs are highly sensitive to noise due to their high impedance. Therefore, it is better to keep the analog communication cables away from strong voltage cables and noisy components. Use a twisted pair cable to establish communication and avoid using single-wire cables for analog communication, preferably use multi-wire or twisted pair cables.

Analog Inputs

Radonix controllers in the PC-Smart models have analog inputs named A[n], where n represents a number greater than zero. The voltage range of these analog inputs is 0 to 10 volts, as shown in Figure 17.

PC-Smart Models:

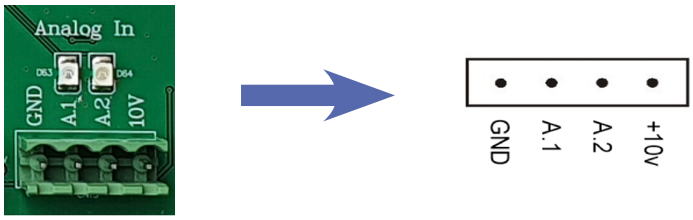


Figure (17) – Analog Inputs

The electronic circuit schematic of analog inputs in PC-Smart models

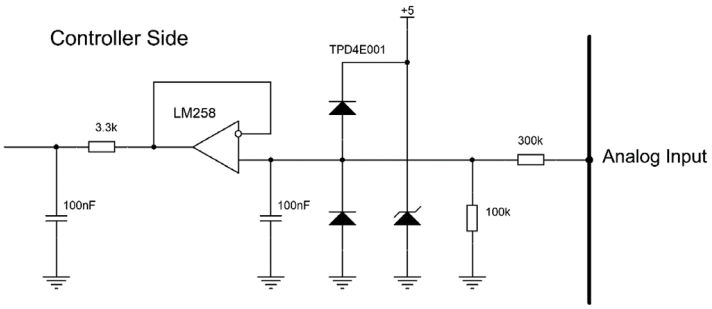


Figure (18) – Analog Inputs

Axes

Axes in Radonix controllers are numbered and named Axis[n]. Only active axes are allowed and inactive axes are not. The connection between the controller and the DB servo motors or stepper motors for the axes is established through connector 15, as shown in Table 2 and Figure 19.

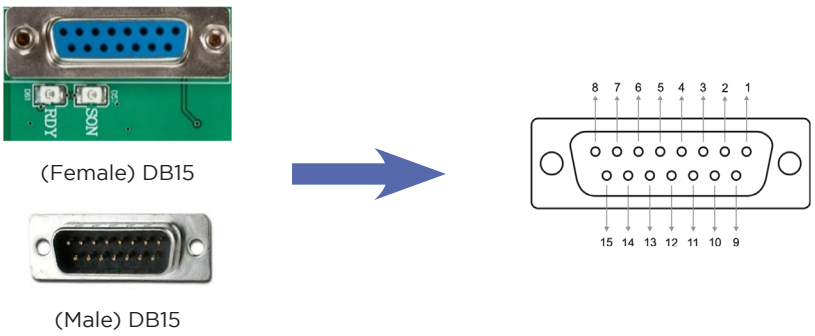


Figure (19) – Connector

1	+VCC
2	Direction -
3	Direction +
4	Pulse -
5	Pulse +
6	Servo On
7	Servo Ready
8	Encoder Zero

9	Alarm Reset
10	GND
11	GND
12	GND
13	GND
14	GND
15	GND

Table (2) – Pinouts

+VCC

This pin is the source voltage output pin through the controller connector. It has a maximum current capacity of 100mA and is used to power the COM+.

Direction+ & Direction-

These two pins determine the direction of the motor. The connection type is line drive and they have a maximum current capacity of 25mA. They can control up to two motors simultaneously and in parallel.

Pulse+ & Pulse-

These two pins determine the amount of movement and the speed of the motor through a digital pulse with line drive output. They have a maximum current capacity of 25mA and can control up to two motors at the same time. The electronic circuit schematic for Pulse and Direction outputs is shown in Figure 20.

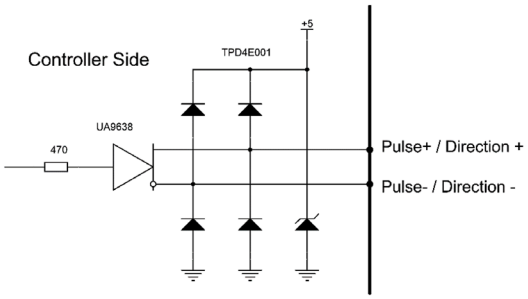


Figure (20) - Internal Circuit of Pulse and Direction Pins.

Servo On

Servo On Pin The Servo On pin is an output pin that is used to turn on the servo motor when the CNC controller is activated. If activated, the SON LED, which is located on the controller near each axis connector, will illuminate. The internal circuit of the Servo On pin is shown in Figure 21. It is important to note that the Servo On pin is not used in stepper motor systems.

PC-Pro Models :

PC-Smart Models:

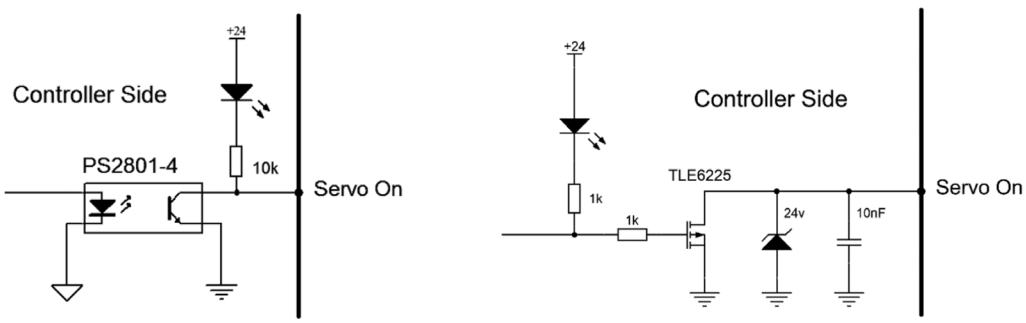
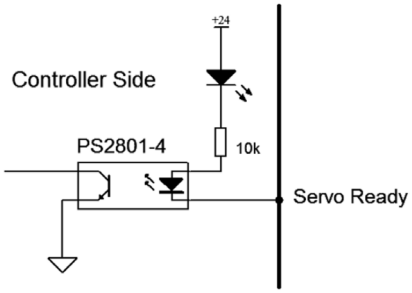


Figure (21) - Internal Circuit of the Servo On Pin.

Servo Ready

Servo-Ready Pin The Servo Ready pin is an input pin isolated by a photo coupler that checks the readiness of the servo motor. In the case of an error or problem with the servo motor's power supply, connection, or encoder, the pin reports the status to the controller, preventing the axes from continuing to move, and an axis error can be displayed on the interface (Figure 22). The Servo Ready pin is not used in stepper motors.

PC-Pro Models :



PC-Smart Models:

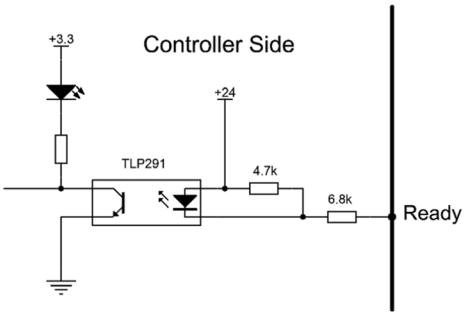


Figure (22) - Internal Circuit of the Servo Ready Pin.

Encoder Zero

This pin is an input pin isolated by Photo Coupler, which is only available in PC-Smart models. And the zero point of the encoder in the servo motor is announced to the controller as a pulse. The use of this pin to find the Home point.
It is accurate on some devices. (Figure 23)
Encoder Zero pin is not used in steppers.

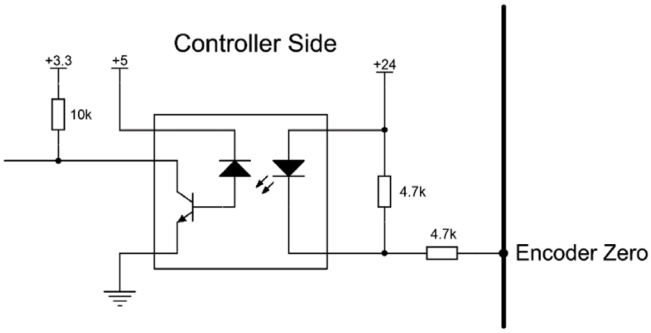
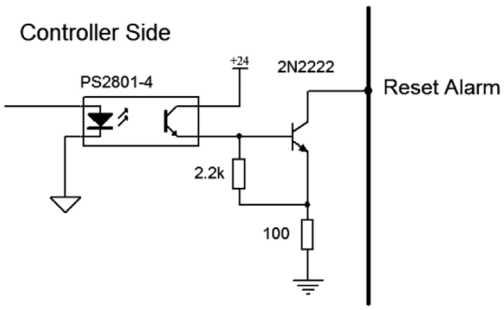


Figure (23) - Internal Circuit of the Encoder Zero Pin.

Alarm Reset

This pin is an output pin used to clear the drive error. If the drive reports an error and it is resolved, the error can be cleared without the need to disconnect the electrical circuit, by using this output pin, and the drive can be put back into normal operation. (Figure 24)

PC-Pro Models :



PC-Smart Models:

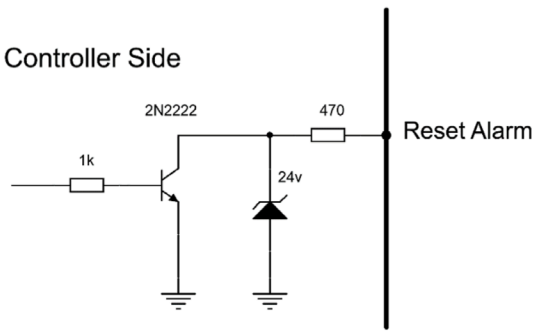


Figure (24) - Internal alarm reset circuit.

GND

The Ground (GND) of the CNC controller is connected to six pins of the axis connector. These pins provide the ground connection required to connect to the drive, ensuring proper electrical signaling and stability of the system.

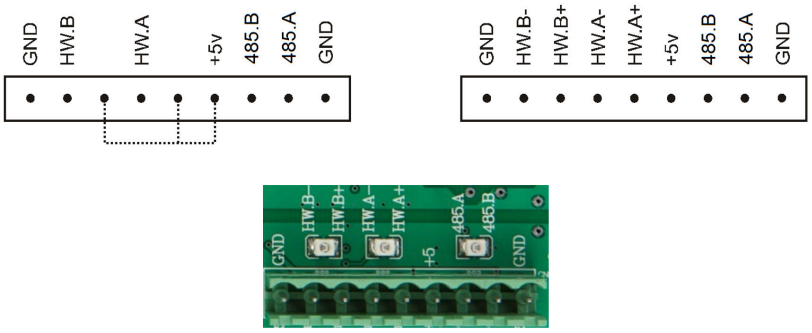
- Please note that the Direction and Pulse output pins are digital and be careful when connecting them to other pins, especially pin 1.

- The Reset and Servo On output pins of the Radonix controller are open collector and NPN, meaning that when activated, Ground is switched to the pin. The driver connected to these pins must be NPN.
- The Ready and Encoder Zero input pins are also NPN, and the servo drive connected to these pins must connect Ground to these pins, making it NPN as well.
- Only the Direction+, Direction-, Pulse+, and Pulse- pins are used to connect the controller to the stepper motors. No other pins are needed. If the cable is disconnected, the axis error may appear as the controller sees this as an error. You can check pins 6 and 7 in the DB15 connector to detect any errors
- For the user's convenience, wiring between Radonix controllers and servo motors is provided in Appendix 1."

Handwheel and serial communication

The 9-pin connector in the Radonix controllers (PC-Pro models, excluding the Pro LAN 3AS model) is used for communication with the handwheel and RS485 connection. In the PC-Smart models, there are two separate connectors (6 and 4 pins respectively) for handwheel and RS485 communication (Figure 25) Handwheels with 5V encoder and differential outputs are compatible for communication with the Radonix controller. The schematic pin connection with the handwheel is shown in Figure 27. RS485 communication, a type of serial communication, is transmitted through pins A.485 and B.485.

PC-Smart Models :



PC-Pro Models :

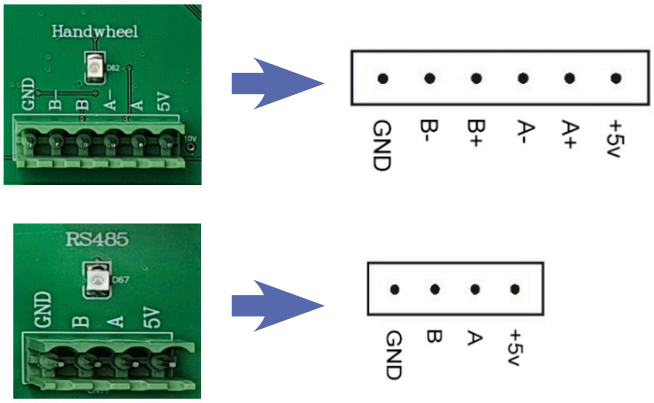


Figure (25) - Communication pins with the Encoder and RS485

The internal circuit of the RS485 communication pins in the PC-Smart and PC-Pro models:

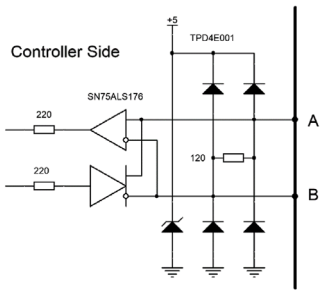
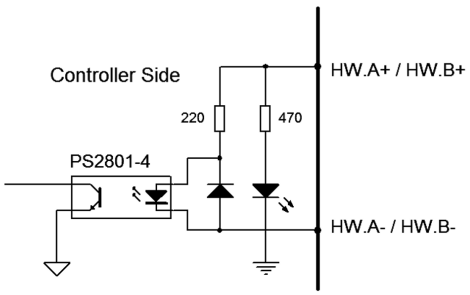


Figure (26) - Internal circuit of RS485 communication pins

The internal circuit of the handwheel communication pins in the PC-Pro models:



The internal circuit of the handwheel communication pins in the PC-Smart models:

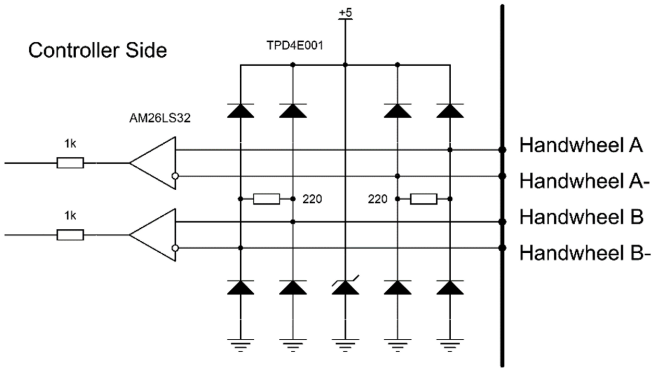


Figure (27) - Internal circuit for communication pins with handheld.

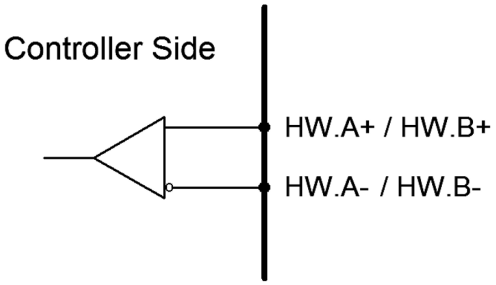


Figure (28) - The schematic equivalent of the internal circuit of the controller for communication with the handwheel.

Figure 29 shows the handwheel encoder communication pins on the controller in two differential and open collector modes. In the PC-Pro models, if the handwheel encoder is an open collector, the HW.A+ and HW.B+ pins should be connected to the +5V pin of the 9-pin connector. The other switches of the handwheel should be connected to the digital inputs of the controller, which is shown in Figure 29.

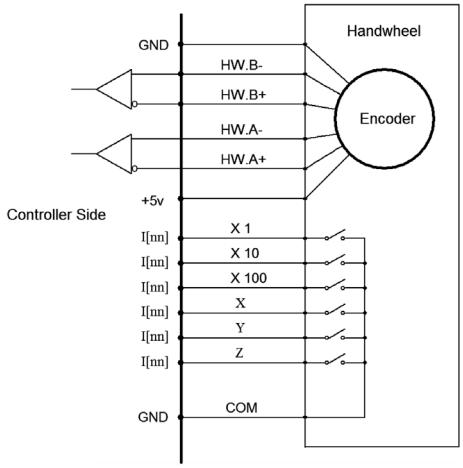


Figure (29) - Connection of handwheel with Radonix controller

- Note that the analog input is only available in the PC-Smart model controllers.

Chapter 4

Software Installation

User Guide

Requirements and comprehensive view for software installation

In this section, we take a general look at the steps of installing Radonix software. First, to install and use Radonix software, you need a computer with the following minimum specifications.

Operating System	Windows 7, Windows 8, Windows 10, Windows 11
CPU	1GHz or higher
RAM	2GByte or higher
Hard Disk	More than 10Gbyte of free space
Graphics Card	The graphics card must be a model that does not encounter an error during installation of XNA and is identified by XNA.
Motherboard:	The computer motherboard must have at least one LAN connection.

Table (1)

The internal circuit of the RS485 communication pins in the PC-Smart and PC-Pro models:

1	.NET Framework 4	dotNetFx40_Client_x86_x64
2	XNA version 4	xnafx40_redist
3	RadonixCAM-Pro software	RadonixCAM-Pro X.X.X
4	Interface	Interface

Table (2)

And after installing the above software, the controller is ready to start.

Installation Steps

- Make sure that the computer is compatible, as explained in the second chapter.
- Ensure that the computer does not have a virus, as a computer virus can cause disruptions in the computer's performance and installed software, including the Radonics software..

After making sure that the computer operating system is healthy, install the software in order.

Installation of Microsoft .NET Framework 4 Client Profile software

To download Microsoft .NET Framework 4 Client Profile, scan the barcode below

After downloading the .NET Framework 4 software, double-click on the downloaded file and follow these steps.

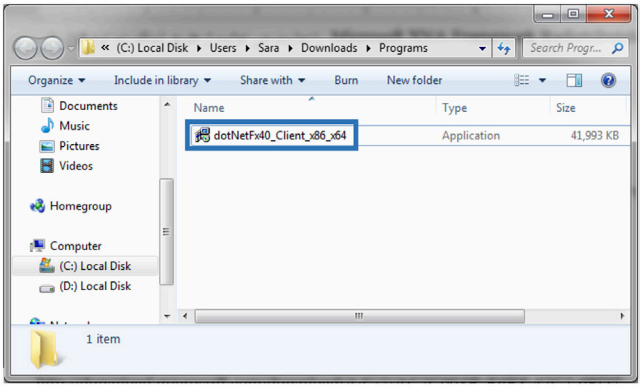


Figure (1)

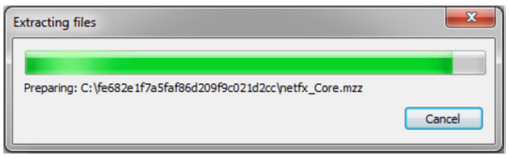


Figure (2)

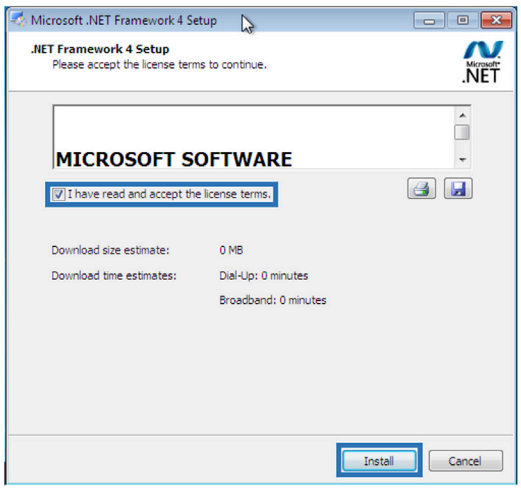


Figure (3)

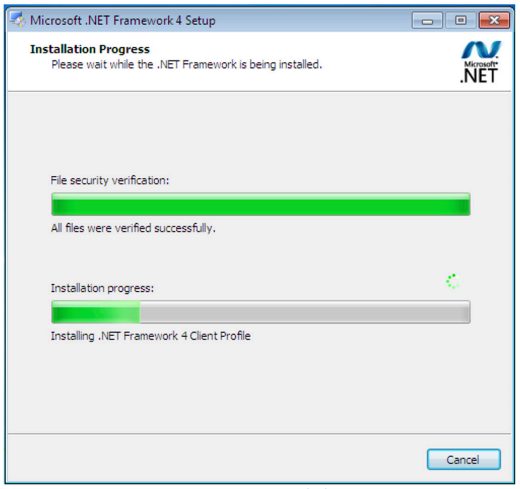


Figure (4)

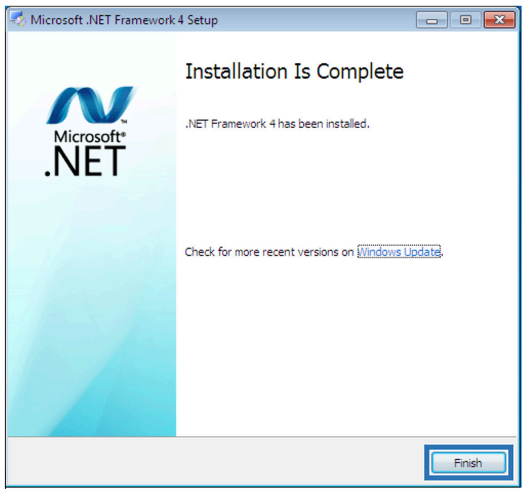
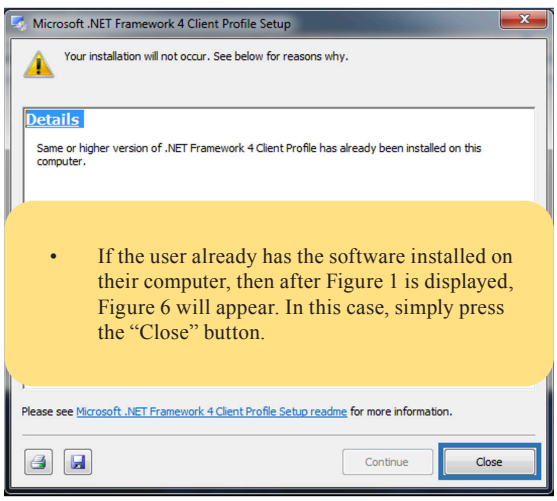


Figure (5)



- If the user already has the software installed on their computer, then after Figure 1 is displayed, Figure 6 will appear. In this case, simply press the “Close” button.

Figure (6)

Installing the Microsoft XNA Framework Redistributable 4.0 software

If you do not have the Microsoft XNA Framework Redistributable 4.0 software, you can download it using the QR code provided

After downloading the Microsoft XNA Framework Redistributable 4.0 software, double-click on the downloaded file and follow the on-screen instructions to install it.

Figure (7)

Figure (8)

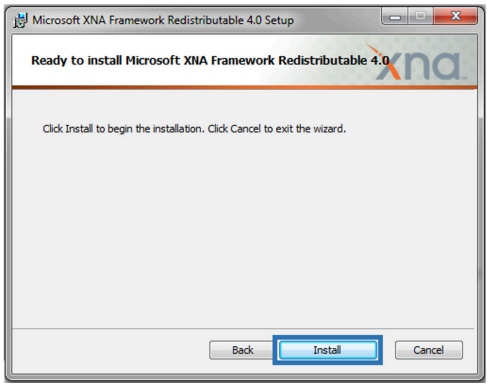


Figure (9)

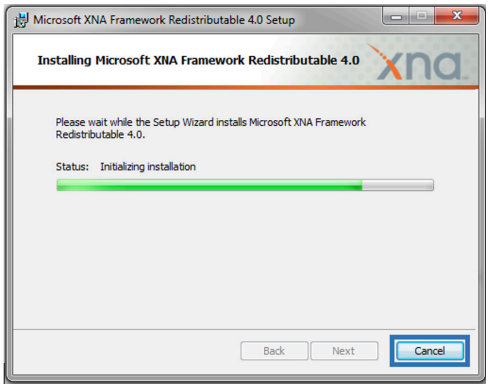


Figure (10)

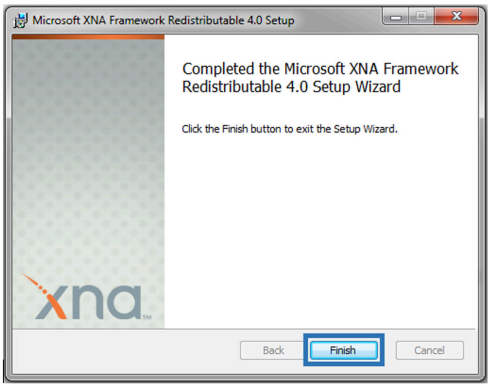


Figure (11)

Installing Radonix CAM-Pro X.X.X software

If you do not have the latest version of the Radonix CAM-Pro X.X.X software (where X represents the latest version number), you can download it by using the link or QR code provided.

After downloading the Radonix CAM-Pro X.X.X software, double-click on the downloaded file and follow the on-screen instructions to install it.

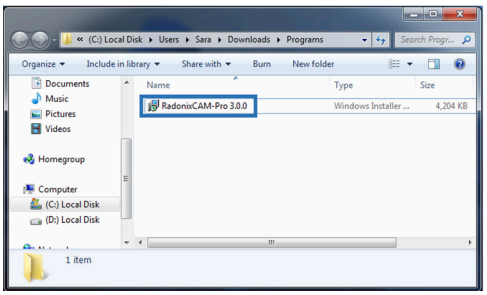


Figure (12)

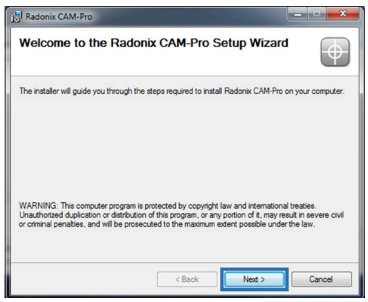


Figure (13)

- To change the default installation and save the location for the Radonix CAM-Pro software, you need to follow the installation steps outlined in Figures 14 and 15. If you do not wish to change the default location, simply proceed with the installation steps starting from Figure 15 or Step 5 written next to the figure.

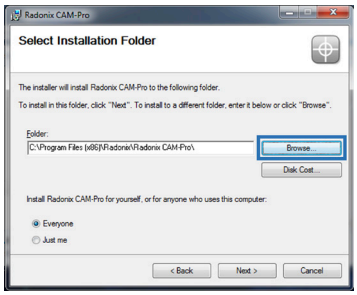


Figure (14)

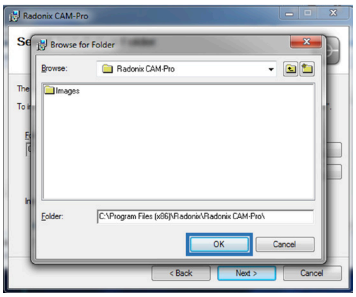


Figure (15)

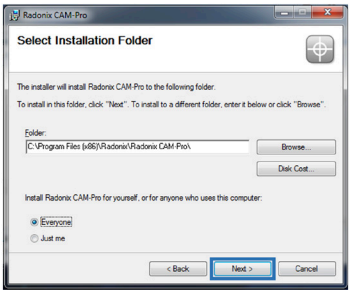


Figure (16)

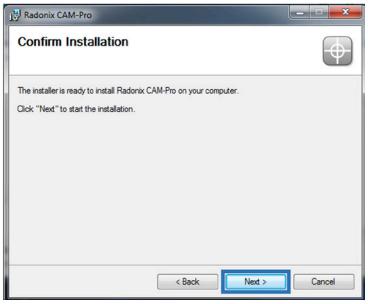


Figure (17)

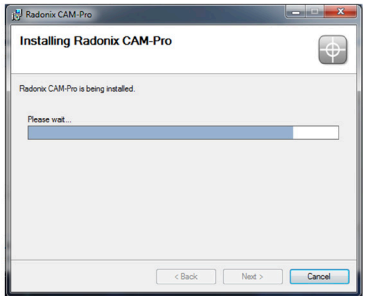


Figure (18)

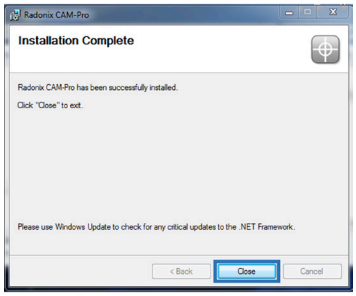


Figure (19)

- If the user already has the software installed on their computer, they can continue by following the steps starting from Figure 20 in order to prevent the installation process from repeating

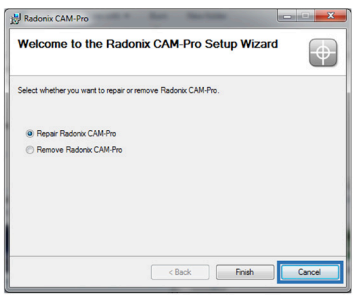


Figure (20)

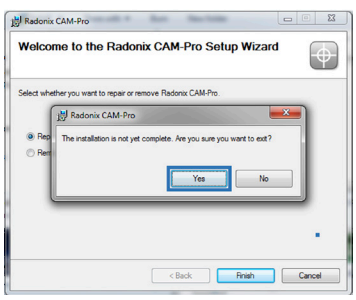


Figure (21)

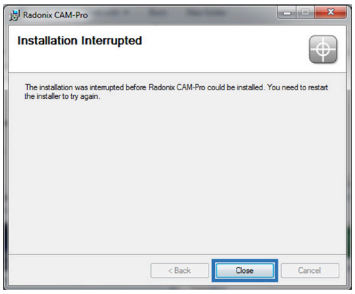


Figure (22)

- After successfully completing the installation of the Radonix CAM-Pro software, the user should select the appropriate interface for their device based on the Radonix controller being used. The different interfaces and their features are thoroughly explained in Chapter 5.

Interface installation

As stated in the introduction, the CNC (Computer Numeric Control) controller is the main interface between the program and the user. Everything the user sees and interacts with on the computer is through the Radonix program. To ensure the proper interface, the user can access the folder of different router interfaces through the barcode and download the desired one.

After downloading the interface, double-click on the downloaded file and follow the steps below.

- Note that due to the variety of interfaces, they are represented as x-x-x in this text. The user should use the appropriate interface based on the type of their device. For example, one of the router interfaces is shown in Figure 23 (for more information, refer to Chapter 5).

Figure (23)

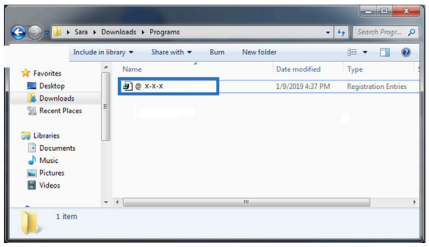


Figure (24)

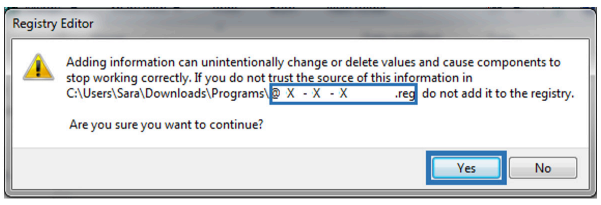


Figure (25)

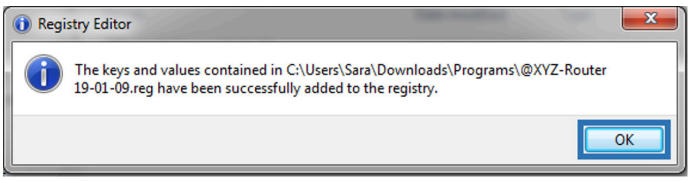


Figure (26)

The interface has been installed correctly and the software is ready to launch.
To access the installed software, go to the Start menu in Windows and search for “Radonix” in the search field. You will see three Radonix icons
(refer to Table 3 and Figure 27) as follows:

	Radonix CAM-Pro	The main Radonix software
	Radonix CAM-Pro Calibrator	It is used for axis calibration. For more information, please refer to Chapter 6.
	Radonix CAM-Pro Test	It is used to test the controller. For more information, please refer to Chapter 7.

Table (3)

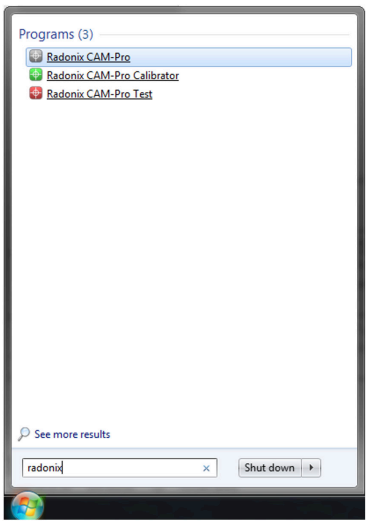


Figure (27)

Connecting the controller to the computer

As explained in detail in Chapter 2 regarding the wiring and voltage supply of the controller, turn on the controller and make sure the Power LED lights up. Then, connect one side of the LAN cable to the LAN port of the controller and the other side to the computer. To establish the connection between the controller and the computer, perform the following basic settings in order. Note that these settings only need to be done once and are saved after the controller and computer are connected.

IP definition

- To set the IP, after connecting the controller to the computer, open the “Not Pro” program.

To set the IP, follow these steps in order:

1. Go to the Start menu in Windows.
2. Go to the Control Panel
3. Follow the steps shown in Figures 28 to 35 in order.

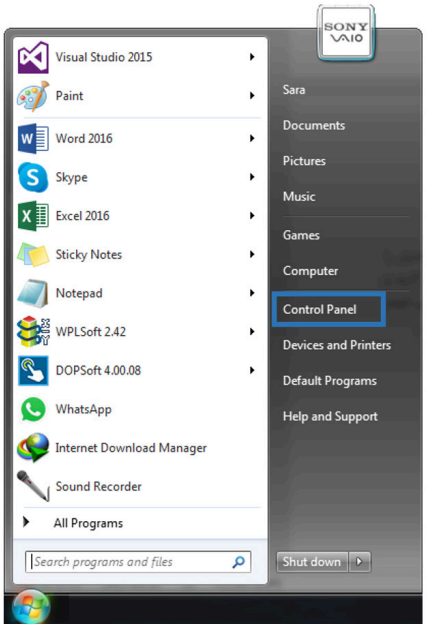


Figure (28)

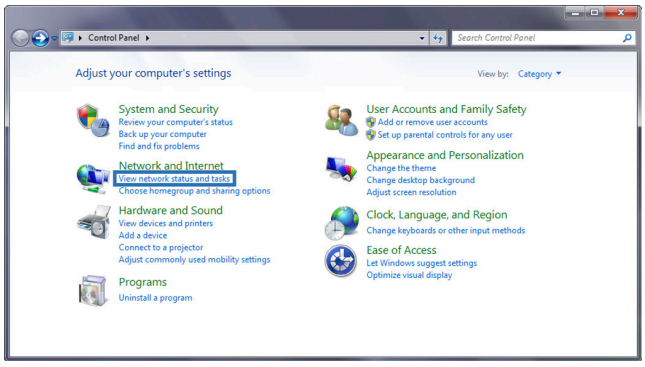


Figure (29)

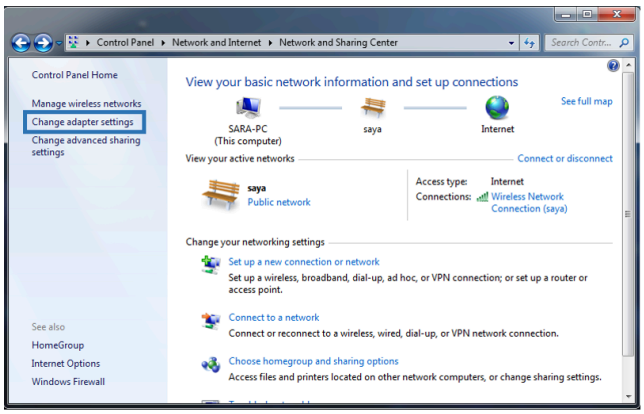


Figure (30)

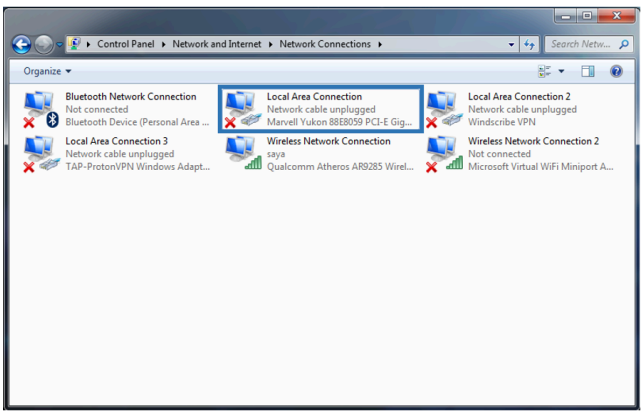


Figure (31)

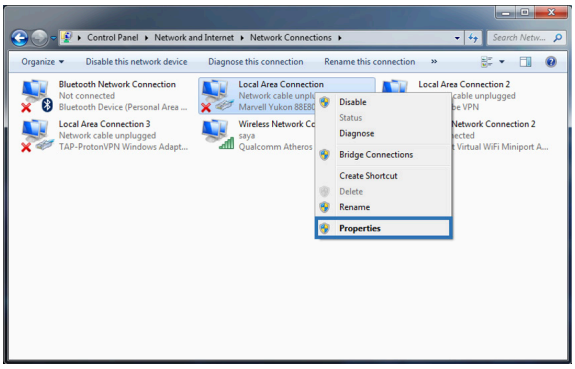


Figure (32)

- According to step 6, first click on Internet Protocol Version 4 (TCP/IPv4) to make it blue, then according to step 7, click on Properties. (Figure 33)

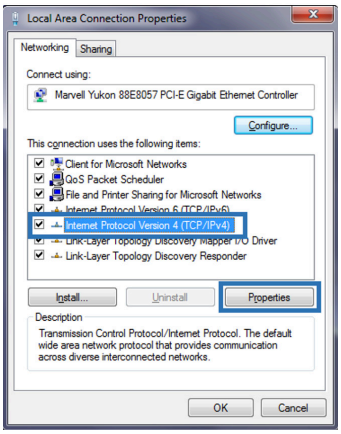


Figure (33)

- According to step 8, select “Use the following IP address”. Then, enter the IP address used in the controller, which is 192.168.11.100, and the Subnet mask, which is 255.255.255.0. Finally, click OK (refer to Figure 34). Close any remaining open windows.

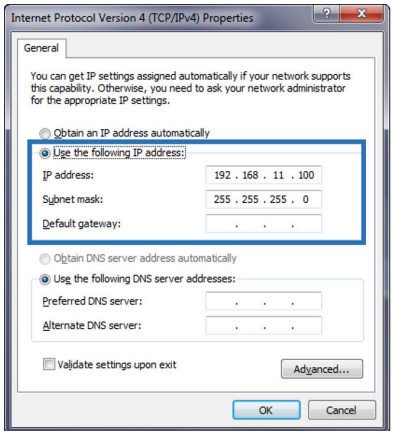


Figure (34)

- After setting the IP, a warning may appear from the Windows Firewall after running each of the Cam Pro programs (Cam Pro, Cam Pro Test, and Cam Pro Calibrator), indicating that the connection is being blocked. In this window, you are asked to allow communication through the network port. To do so, follow the steps in 9 and 10, tick the two options, and finally click the “Allow Access” button. This will allow Windows to communicate with the Cam Pro programs through the network port. (Refer to Figure 35)

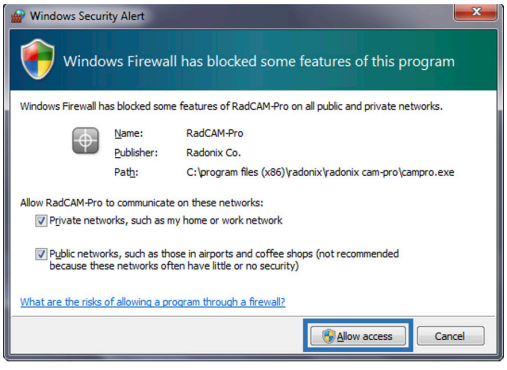


Figure (35)

- If you did not correctly perform steps 9, 10, and 11, and the IP is blocked by the firewall or if an antivirus is installed on the computer and is blocking the connection between the controller and the computer, then follow the steps shown in Figures 36 to 45 in order to resolve the issue.

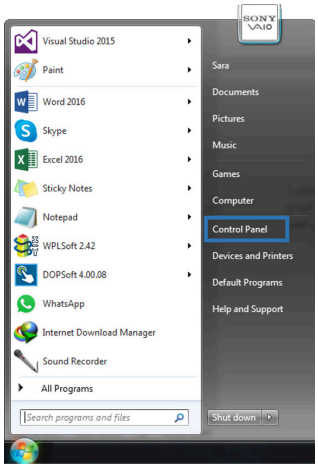


Figure (36)

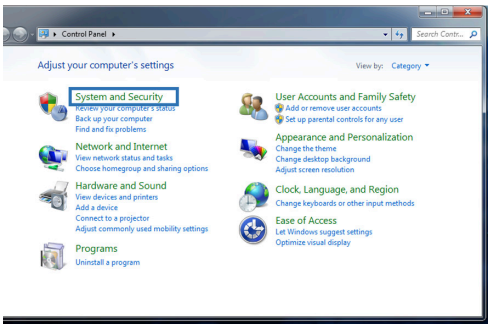


Figure (37)

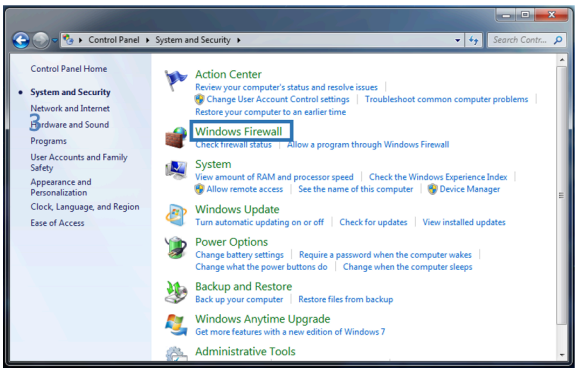


Figure (38)

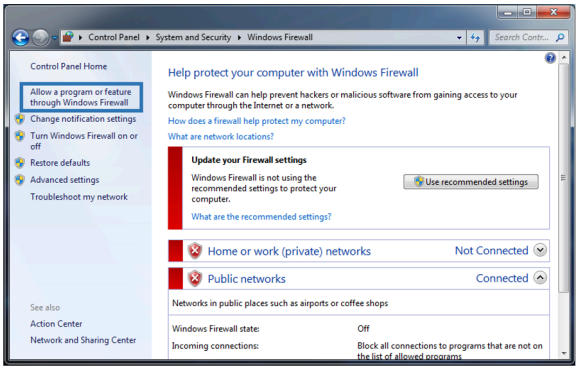


Figure (39)

- If the software appears in gray color in the “Allow a program or feature through Windows Firewall” environment, you will not be able to make changes. To be able to make changes, click on “Change settings” as shown in Figure 40. If the software appears in black color, this means you have the ability to make changes, so you can skip this step and do not need to follow step 5. (Refer to Figure 40)

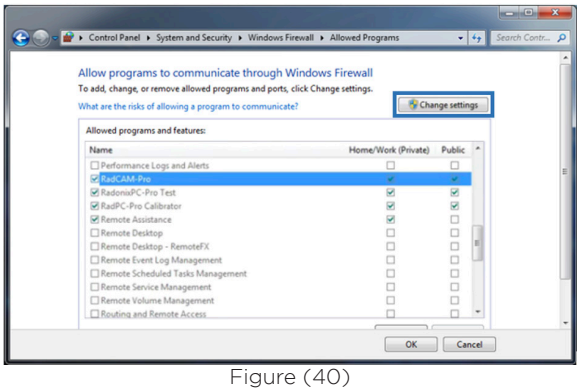


Figure (40)

- If the Radonix software (RadCAM-Pro, RadonixPC-Pro Test, RadPC-Pro Calibrator) is not listed in the “Allow a program or feature through Windows Firewall” environment (refer to Figure 41), you need to follow steps 6 to 8 as shown in figures 41 to 43. However, if the software is already listed, you can skip these steps and do not need to perform steps 6 to 8.

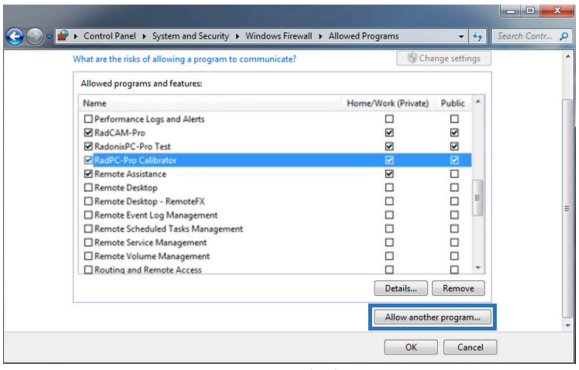


Figure (41)

- In step 7, you should find Radonix files from the Browser section and add them to the list. (Figure 42)

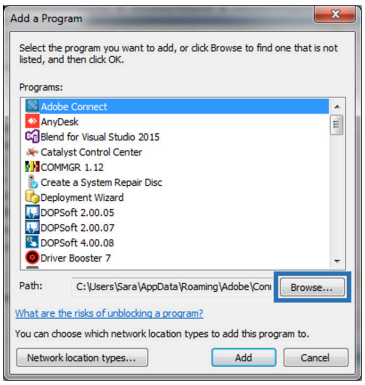


Figure (42)

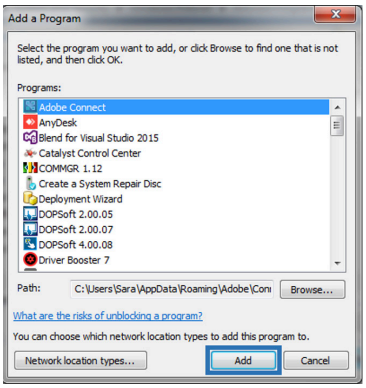


Figure (43)

- According to step 9, all squares within the box should be checked (ticked) and if a checkmark is missing, it should be added by clicking within the check box. (Refer to Figure 44 for visual reference).

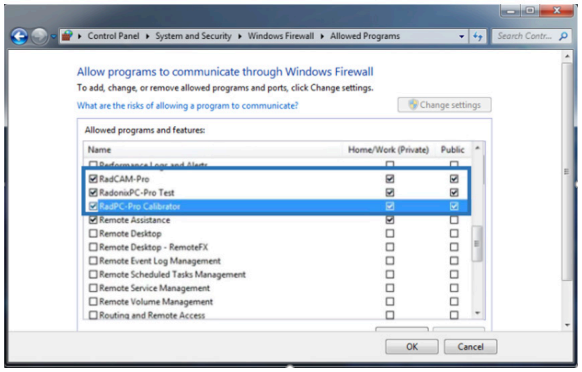


Figure (44)

- Now you are able to launch the Radonix software.

- If the CNC controller and computer are properly connected and the necessary power supply voltage is being supplied to the controller, the LAN connector LEDs on the controller should be either blinking or on. If these LEDs are off, it can be concluded that the cable connection between the controller and the computer is disconnected or there is an issue with the connector.

Chapter 5

Interfaces and Router Parameters

User Guide

Interfaces and Router Parameters

In this chapter, we will examine the different interfaces utilized to operate the router device. We will also delve into the various parameters found in these interfaces, and learn how to use the software and elements associated with them.
To aid in understanding the material, some technical terms that may require clarification will be defined briefly. As previously stated in the introduction, this will help make the text easier to understand.

- Interface: The main interface of the Radonix Cam Pro program that the user interacts with and communicates with the CNC controller through. This is any part of the program that the user can see on the computer screen.
- Variable: A variable refers to numbers and strings of letters that can be changed by the user in the user interface, affecting the display, movement, and reaction of the program.
- Function: A function is a set of commands and instructions that the CNC controller or user can execute through the program, enabling communication between the controller and the user.
- Parameter: A parameter refers to the internal variables of the CNC controller or interface that the user can modify to change the execution process of the program or the movement of the machine.
- Port: A port refers to the set of inputs and outputs of the CNC controller.
- Tabs: Tabs are used to categorize and separate different sections of the interface, preventing overcrowding on the screen.
- Element: An element refers to each display tool used in the user interface. The user communicates with functions and parameters through these elements.
- Key: A key can refer to either a hardware or software key, depending on the type of application. These are defined based on the specific needs of the program.

All of the variables associated with the CNC controller, including information on the interface, movement settings, software-hardware relationships, and any other configurable variables, are stored in a tree diagram in the operating system registry. The user can access these variables through the settings window. The variables are organized into branches based on their specific application or device type, with descriptive names, making it easier for the user to locate them.

Interface

In CNC router machines and all other CNC machines, it is necessary for the machine to have knowledge of all the points within the limited machining space so that the movement of the tool on the workpiece can be defined in terms of known coordinates. This is achieved by having control axes embedded in the machine that control movement in linear paths and/or rotations. These axes include the main axes, auxiliary axes, and rotary axes, which control the direction of movement of the table or tool.
The primary axes are typically designated as X, Y, and Z, and they must be perpendicular to each other. The auxiliary axes, which are aligned with the primary X, Y, and Z axes, are named U, V, and W. These axes are not considered independent axes and are used to supplement the limited movement of the primary axes. In addition to these axes, there is also a rotary axis that is built around the X, Y, and Z axes, and it is designated as A, B, and C, respectively. (Figure 1)

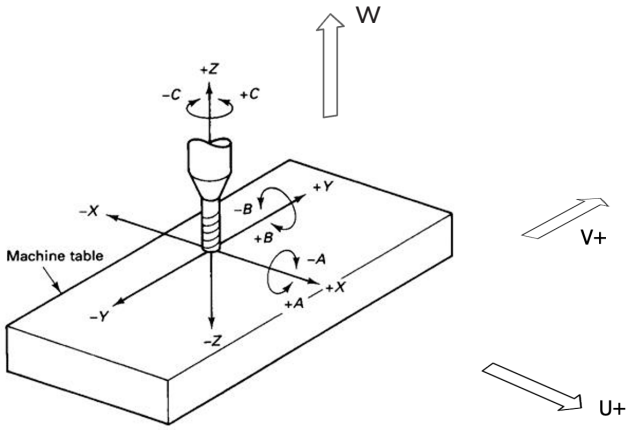


Figure (1)

How to determine coordinate axes

The way to determine the direction of the axes of the device is based on the right-hand rule that if we hold the three middle, index, and thumb fingers together perpendicularly, the middle finger is the Z+ axis, the index finger is the Y+ axis, and the thumb is the X+ axis. Obviously, the negative directions of X, Y, and Z are opposite to the direction shown. It should be noted that the Z axis is always aligned with the rotary axis, and the positive direction of Z is from the workpiece to the tool.

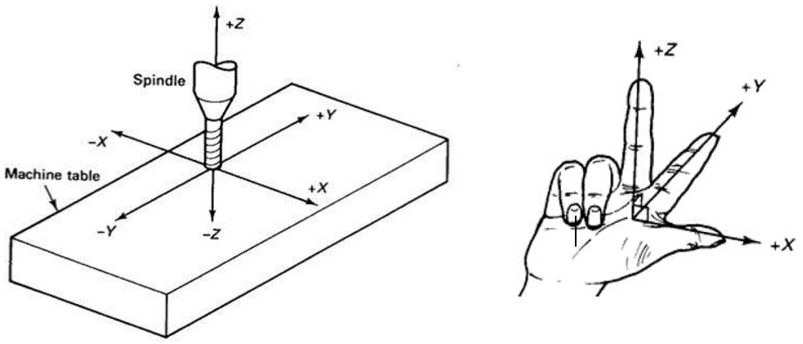


Figure (2)

To determine the direction of the rotation axes A, B, C, we hold the right hand so that the thumb points in the positive direction of the X, Y, Z axes. The direction of rotation indicated by the other four fingers, when they are closed, shows the positive direction of the A, B, C axes. It is important to note that the negative directions of A, B, C are opposite to the direction indicated by the fingers.

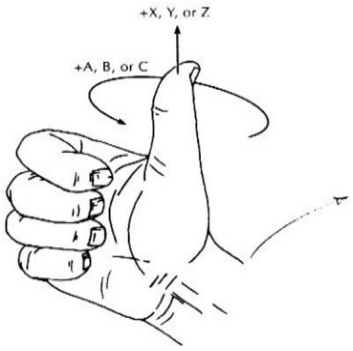


Figure (3)

Types of interface

Different interfaces have been developed based on the coordinates used in the router. As explained in the previous section, which discussed the types of coordinate axes, it is easy to understand the interfaces and their usage. You can reach out to the interfaces via QR provided in Figure(5) from Radonix Github

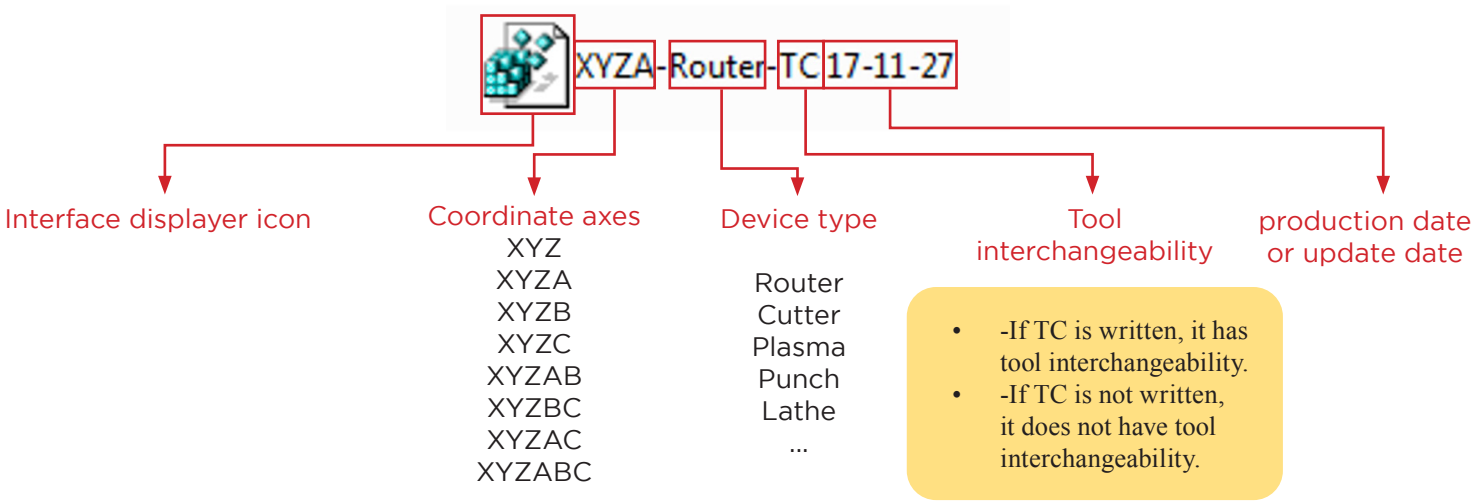


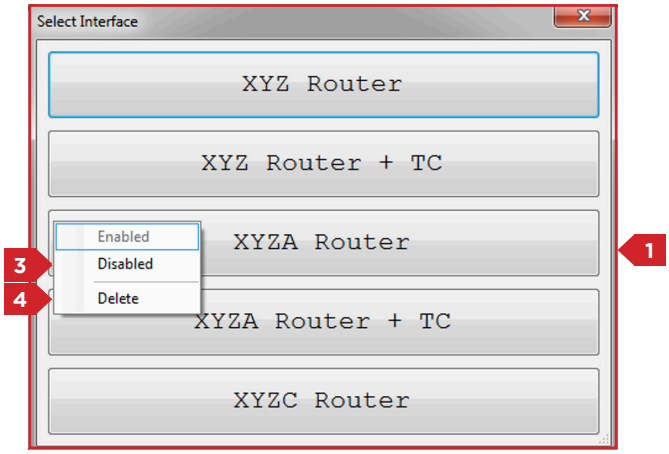
Figure (4)

or you can click here for download directly

Figure(5) - Interfaces download link

Softwere

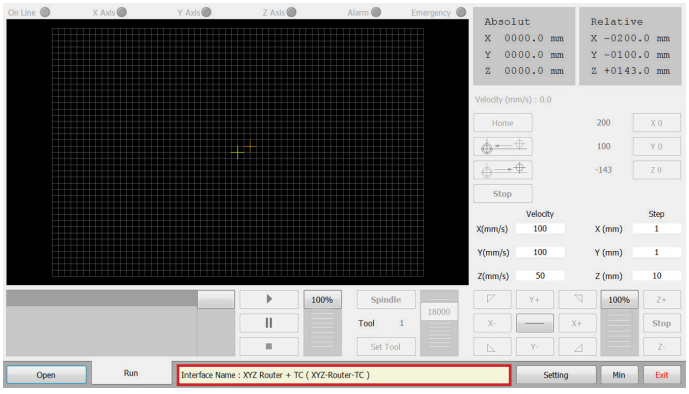
After installing the software, as described in Chapter 3, and the appropriate interface, you can now use the Radonix software. This section will explain all of the elements used in the software. Figure 6 shows a list of all the installed interfaces. If multiple interfaces are installed on one computer, the user can enable or disable each interface by right-clicking on it and selecting the appropriate option from the menu. The user can also remove the interface from the computer.



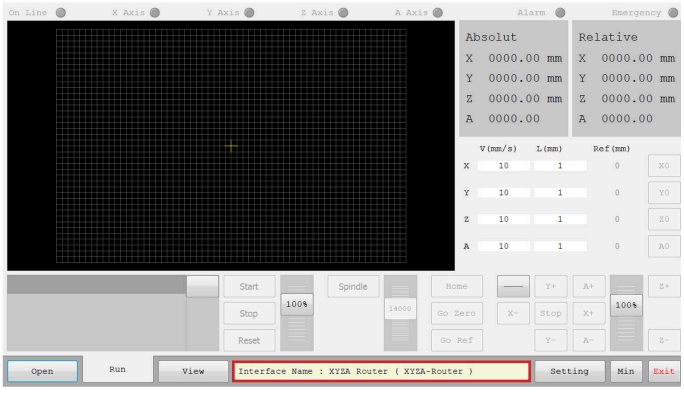
Figure(6)

1. Activate the interfaces
2. Reactivate the interfaces
3. Disable the interface, which is indicated by dimming the caption of the interface
4. Completely remove the desired interface from the computer

To open an interface, the user can left-click on any of the software interfaces. Figures 7 and 8 show two examples of interfaces. Each interface displays different icons, depending on its characteristics. For example, Figure 7 includes the X, Y, Z axes, and the ability to change tools, while Figure 8 includes only the X, Y, Z, and A axes, and does not include the ability to change tools. This clearly indicates that Figure 8 is an A-axis controller interface, and the tool-changing feature has been removed in this form. Similarly, other differences can be observed.



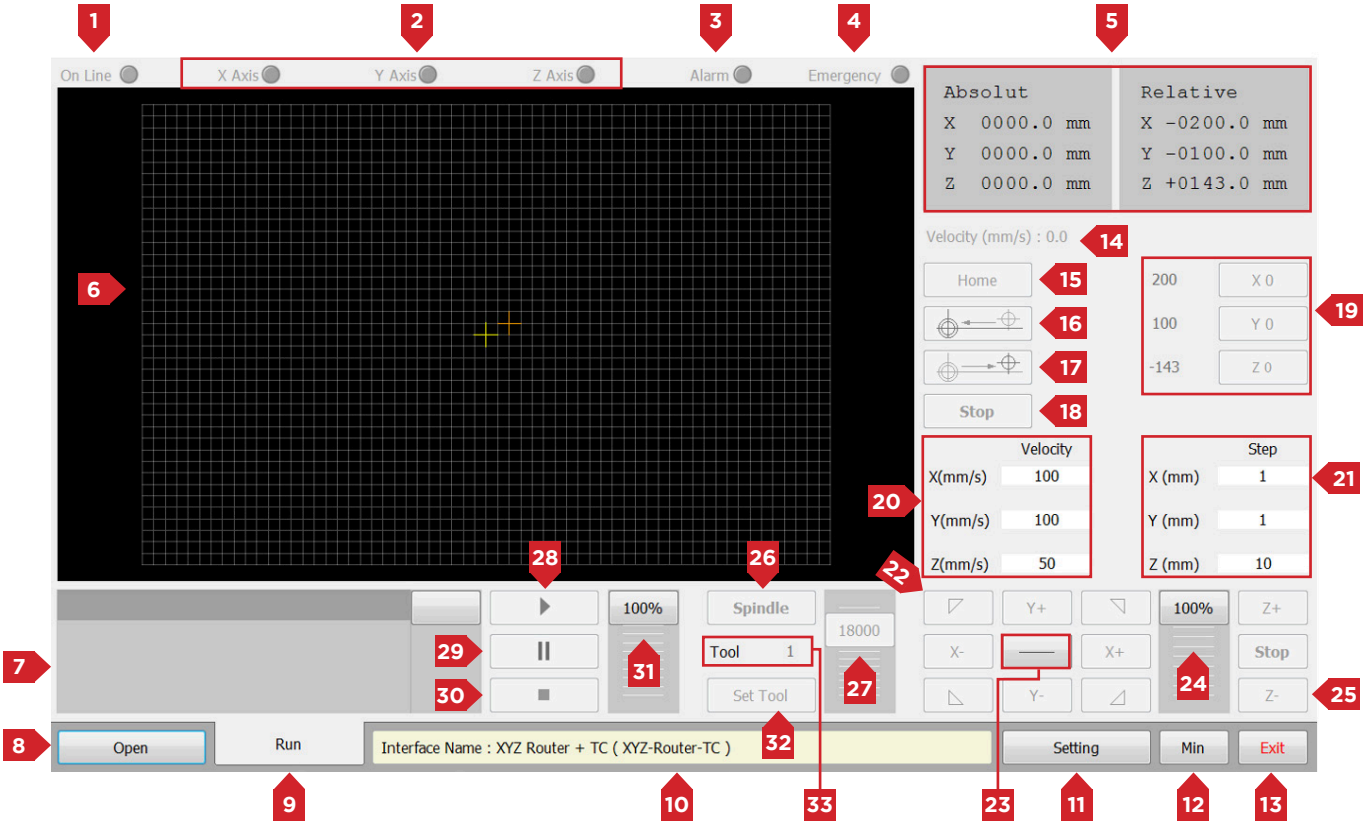
Figure(7)



Figure(8)

Introduction of menus and keys on the software page

Understanding the Functionality of Each Button and Component



Figure(10)

Number	Name of the Element	Description	Shortcut
1	Online	If the controller and software are connected, it is on. If there is no communication between the controller and the software, it will be turned off or displayed in gray color (Figure 9)	
2	X Axis, Y Axis, Z Axis ...	When each axis is active, it will be displayed with a light indicating its activation. If an axis is inactive, it will be turned off and displayed in gray color (as shown in Figure 9). **Note that the display of the axes may vary in different interfaces. The interface may display all available axes or a different number depending on the interface's capabilities.**	
3	Alarm	**If there is an error or issue with the connection between the controller and the motors and drives, the indicator will turn on, otherwise, it will remain turned off**	
4	Emergency	In the event of an error in the emergency settings section, the indicator can be adjusted to turn on. **Note that the number of emergency device inputs can be defined.**	
5	Absolut & Relative	Absolute: Displays the coordinates of the axes in reference to the home point. Relative: Shows the coordinates of the axes relative to the reference point. **Note that the display of the coordinates may vary in different interfaces. The interface may display all available axes or a different number depending on the interface's capabilities.**	

6	Tool Path (Movement path indicator Tool)	Once a file has been loaded, the movement path will be displayed in this section. Both in simulation mode and during actual operation, you can observe the device's movements and the sequence of program execution. In the upper left corner, you can see the file's upload location and the Work Size or Work Length, which displays the sizes in different axes. In the lower right corner, you can view the file from different perspectives and zoom in or out. By holding the right mouse button, you can change the viewing angle of the design.	
7	G-Code display	Once you have loaded the desired file using the Open button, the Code-G and program lines will be displayed in this section. During program execution, each executed GCode will be highlighted with a different color.	
8	Open	It is used to load G-Code, dxf, and Autocad files.	CTRL + SHIFT + ALT + O
9	Run	Tab is program execution. Different interfaces may have more than one tab.	
10	Text box	In normal mode, this section displays the name of the interface. When a CNC operation is being performed, this section displays the approximate time elapsed and the remaining time of the operation. When an alarm or emergency occurs, this section changes to red. By double-clicking with the left mouse button, the Form Debug window will open. Here, you can view the performance report of the controller and software, including the Radonix software version, interface type, and alarms. By pressing a shortcut key twice, a general performance report of the device will be displayed in Notepad.	CTRL + SHIFT + ALT +PrtSc
11	Setting	All device parameter settings can be configured in this window. (For more information, refer to section 4.2.2).	
12	Min	In order to minimize the software window, press the minimize button.	
13	Exit	To exit the software.	
14	Velocity	This feature is available in some interfaces and displays the movement speed in ToolPath. If needed, it can be added to interfaces that do not have this feature by default. (For more information, refer to section 4.2.7).	
15	Home	This feature is available in some interfaces and displays the movement speed in ToolPath. If needed, it can be added to interfaces that do not have this feature by default. (For more information, refer to section 4.2.7).	Alt + Home
16	Go Zero	By pressing this button, the device will quickly move to the zero point (Home) of the device. The difference between this case and case 15 is that in this case, the device moves quickly to the zero point without the need to slowly approach it to detect the sensors	
17	Go Ref	In order to move the axes to the zero point of the workpiece or reference point.	
18	Stop	In order to stop the process of device operation.	
19	X0,Y0,Z0,...	To determine the reference point, simply click on one of the buttons. The reference point value will be displayed in the box next to the button. **Note: The display of available axes may vary depending on the interface and the number of axes.**	Alt+X => X0 Alt+Y =>Y0 Alt+Z =>Z0

20	Velocity text box of axes X(mm/s), Y(mm/s), Z(mm/s),...	In these text boxes, the movement speed of the axes can be entered, and these values are applied in Continuous mode or  if the Jog key is held on the desired axis. In Continuous mode, the value entered in the text box for each axis is represented as a percentage of manual movement speed (in relation to it). The Jog term is multiplied, and by pressing the Jog key, each axis moves at the speed resulting from the product of the Jog term and the value entered in the Velocity text box. For example, if the number entered in the Velocity text box for the Z axis (Figure 11) is 10mm/s, and the Scroll percentage is 100%, the product of these two numbers is 10mm/s. By pressing the Z+ or Z- key, the Z axis moves in the positive or negative direction at a speed of 10mm/s, respectively	
21	Step text box of axes X(mm), Y(mm), Z(mm),... which are located next to the axes	In these text boxes, the movement value of the axes is entered, and these values are applied in Incremental mode or when the corresponding Jog key for each axis is pressed. In Incremental mode, the value entered in the text box of each axis is multiplied by the Scroll percentage related to manual movement (also known as Jog), and when the Jog key of each axis is pressed, it moves by the product of the above value. For example, if the value entered in the Step text box for the Z axis (Figure 11) is 50mm and the Scroll percentage is 100%, the product of these two values is 50mm. Thus, each time the Z+ and -Z axis buttons are pressed, it moves 50mm in the positive and negative direction, respectively.	
22	Two-axis movement Simultane-ous	The button for simultaneous movement of two axes is used to move two axes at the same time. It works in the same way as pressing two Jog buttons at the same time, but each axis will move based on its own set parameters and speed. The function is similar to the one described in item 25 of the same table.	
23	Continous and incremental	This button is a toggle switch, which means that each time you press it, the mode alternates between Continuous and Incremental. In Continuous mode or  if you hold the Jog key, the selected axis moves at the set speed. (For more information, see item 20 in the same table) In Incremental mode or  by pressing the Jog key each time, the selected axis moves to the specified distance. (For more information, see item 21 in this table).	
24	Scroll	In fact, the percentage of this scroll is calculated by multiplying the values in the text boxes in manual or jog mode (refer to numbers 20 and 21 in the table for more information).	
25	X+, X-, Y+, Y-...	To manually move the axes, the Jog axes can be used in two modes: Continuous and Incremental. In Continuous mode, the desired axis moves at the set speed if the Jog key is held. (Refer to number 20 in the same table for more information.) In Incremental mode, the desired axis moves by the set distance each time the Jog key is pressed. (Refer to number 21 in the same table for more information.)	
26	Spindle	It is to turn the spindle on or off.	
27	Scroll	**It is used to determine the spindle rotation speed.**	
28	Play	It is for the purpose of running the program.	
29	Pause	To halt the execution of a program, it's important to note that pressing the Pause button will stop the program on the same G-Code line. The program can be resumed from that line by pressing the Play button again.	
30	Stop	To stop and reset the program.	

31	Scroll	In order to control the execution speed of the program.	
32	(Text box) Tool	To call a specific tool, the tool number must be entered. When changing tools, if the spindle is holding a tool, it will be placed in its designated location and the tool specified by the number entered in the text box will be taken when the Enter button is pressed.	
33	Set Tool	It is to set the tool height.	

Table(1)

<table>
<tr><td>1</td><td>ConveyorSensorPin</td><td>This is the input of a canoview sensor. When this input is activated, the canoview will turn off.</td></tr>
<tr><td>2</td><td>Axis name ,HomeFastPin</td><td>This is used for a sensor that is located before the Home sensor and moves the machine at a high speed during the Home operation. After activating this input, the machine moves at the speed that was set for the Home operation.
<table>
<tr><td>Axis name, HomeFastPin</td><td>The axis name can be any of the X, Y, Z, A, B, or C axes.</td></tr>
<tr><td>Axis name, HomeFastPin, -</td><td>In some machines that use two motors for one axis (gantry axis), two HomeFastPin inputs can be used for both sides of the machine, in which case a negative sign is used for the second input. The axis name can be any of the X, Y, Z, A, B, or C axes.</td></tr>
</table></td></tr>
<tr><td>3</td><td>Axis name, HomePin</td><td>When this input is activated, the home point of the machine is determined.
<table>
<tr><td>Axis name, HometPin</td><td>The axis name can be any of the X, Y, Z, A, B, or C axes.</td></tr>
<tr><td>Axis name, Hometpin</td><td>In some machines that use two motors for one axis (gantry axis), two HomePin inputs can be used for both sides of the machine, in which case a negative sign is used for the second input. The axis name can be any of the X, Y, Z, A, B, or C axes.</td></tr>
</table></td></tr>
<tr><td>4</td><td>InterruptEndSensorPin number, InterruptEndSensorPin</td><td>InterruptEndSensorPin is used for the following two cases:
1. The entire process is executed if a single M code or function command is given.
2. The entire process is executed if two M codes or function commands are given.

Note that if InterruptStartSensorPin and InterruptEndSensorPin are not defined for the inputs, the Interrupt process will be executed without the need for the input sensors to be triggered or True. As the Interrupt process is fully related to the input, output, and function links, the complete process performance is described here, and it is sufficient to refer to the corresponding table and define the links based on the pin or function name when defining the process.

In addition, in the diagram drawn for better clarity, the sequence of operations and the activation of inputs, outputs, and functions are shown. The links are displayed with abbreviation symbols
(Input Link -- I), (Output Link -- O), and (Function Link -- F)

• In the case where the entire process is executed if a single M code or function command is given:
When the Interrupt command (Function Link) is executed, it first waits for the InterruptStartSensorPin input (Input Link) to be triggered, and when the InterruptStartSensorPin input (Input Link) is triggered or True, the command to turn on the InterruptPin output (Output Link) is issued. It then waits for the InterruptEndSensorPin input (Input Link), and if the InterruptEndSensorPin input (Input Link) is triggered or True, the InterruptPin output (Output Link) is turned off, and the process ends.
(please refer to appendix 6 - Figure (1))

• In the event that the entire process is executed with two M codes or two functions under command:
When the InterruptStart command (function link) is executed, it first waits for the InterruptStartSensorPin input link and when the InterruptStartSensorPin input link is triggered or, in other words, becomes True, the command to turn on the InterruptPin output link and InterruptStartPin output link is issued and the first stage of the process is completed.
(please refer to appendix 6 - Figure (2))</td></tr>
</table>

<table>
<tr><td></td><td></td><td>When the InterruptEnd command (function link) is executed, first the InterruptPin (output link) and InterruptStartPin (output link) are turned off, and InterruptEndPin (output link) is turned on. Then, it waits for the input of InterruptEndSensorPin (input link), and if InterruptEndSensorPin (input link) is triggered, or in other words, becomes True, InterruptEndPin (output link) is turned off and the final step of the process is completed.
(please refer to appendix 6 - Figure (3))</td></tr>
<tr><td>5</td><td>InterruptStart-SensorPin Number, Inter-ruptStartSensor-Pin</td><td>InterruptStartSensorPin is used for the following two conditions:
1. The entire process is executed when a single M-code or a single function command is given.
2. The entire process is executed when two M-codes or two function commands are given.
Please note that if InterruptStartSensorPin and InterruptEndSensorPin are not defined for the inputs, the Interrupt process will be executed without the need for the input sensor to be triggered or in other words, become True.
Since the Interrupt process is completely related to input, output, and function links, the entire process is explained in this section, and when defining the process, the links used are mentioned, and it is sufficient to refer to the relevant table and define the links based on the pin or function name.
Additionally, in the diagram that has been drawn for better clarity, the order of operations and the activation of inputs, outputs, and functions are shown.
Links are displayed with abbreviations.
(Input Link -- I), (Output Link -- O), and (Function Link -- F)

<table><tr><td>InterruptStart-SensorPin</td><td>By executing this function, only one Interrupt process is performed.
In fact, this function does not have any argument as input, and the program considers the argument number as 0.</td></tr><tr><td>InterruptStart-SensorPin, Number</td><td>The number entered as an argument is actually used to define different Interrupts for performing different processes.
In this case, all Interrupt links (input, output, and function links) that are used for a process must have the same arguments. For example, the argument for all of them should be 4.
The number can be between 0 and 15.</td></tr></table>
• In the case where the entire process is executed when a single M-code or a single function command is given:

When the Interrupt command (function link) is executed, it first waits for the input of InterruptStartSensorPin (input link), and when the input of InterruptStartSensorPin (input link) is triggered, or in other words, becomes True, the command to turn on the InterruptPin (output link) is issued. Then, it waits for the input of InterruptEndSensor-Pin (input link), and if InterruptEndSensorPin (input link) is triggered, or becomes True, the InterruptPin (output link) is turned off and the process is completed.
(please refer to appendix 6 - Figure (4))

• In the case where the entire process is executed when two M-codes or two function commands are given:

When the Interrupt command (function link) is executed, it first waits for the input of both InterruptStartSensorPin1 and InterruptStartSensorPin2 (input links), and when both inputs are triggered, or in other words, become True, the command to turn on the InterruptPin (output link) is issued.
Then, it waits for the input of both InterruptEndSensorPin1 and InterruptEndSensorPin2 (input links), and when both inputs are triggered, or become True, the InterruptPin (output link) is turned off and the process is completed.
(please refer to appendix 6 - Figure (5))</td></tr>
</table>

<table>
<tr><td>5</td><td></td><td>When the InterruptEnd command (function link) is executed, first the InterruptPin (output link) and InterruptStartPin (output link) are turned off, and InterruptEndPin (output link) is turned on. Then, it waits for the input of InterruptEndSensorPin (input link), and if InterruptEndSensorPin (input link) is triggered, or in other words, becomes True, InterruptEndPin (output link) is turned off and the final step of the process is completed.
(please refer to appendix 6 - Figure (6))</td></tr>
<tr><td>6</td><td>Axis name, LimitPin</td><td>The input is a limit switch for the axes. When this input is activated, movement is not allowed on the determined axis.
<table>
<tr><td>Axis name, LimitPin</td><td>It is for the limit switches related to the positive direction of the axes. The name of the axis can be any of the X, Y, Z, A, B, C axes.</td></tr>
<tr><td>Axis name -, LimitPin</td><td>The negative sign is for the limit switches related to the negative direction of the axes. The name of the axis can be any of the X, Y, Z, A, B, C axes</td></tr>
<tr><td>Axis name, LimitPin, H</td><td>In addition to limiting the axis, this input is considered as the Home position for that axis in the positive direction. The name of the axis can be any of the X, Y, Z, A, B, C axes.</td></tr>
<tr><td>Axis name, LimitPin, H-</td><td>In addition to executing the command related to the limit switch input, this input is considered as the Home position for that axis in the positive direction. The name of the axis can be any of the X, Y, Z, A, B, C axes.</td></tr>
</table></td></tr>
<tr><td>7</td><td>R-CornerJack-SensorPin</td><td>The input is the jack sensor of the table. When this input is activated, it determines whether the jack is on or off.</td></tr>
<tr><td>8</td><td>SimpleCom-mandSensorPin Number,</td><td>Since the SimpleCommand process is fully related to input, output, and function links, the complete process functionality is explained in this section. To fully clarify the process, the used link is mentioned when defining the process, and it is enough to refer to the related table and define the links based on the pin or function name. In the diagram, for better clarity, the sequence of operations and the activation of inputs, outputs, and functions are shown. The links are displayed with abbreviations.
(Input Link -- I), (Output Link -- O), and (Function Link -- F)

When the SimpleCommand (function link) command is executed, first the SimpleCommandPin (output link) is turned on, and then it waits for the SimpleCommandSensorPin (input link) to be activated, or in other words, to become True. Once this happens, the SimpleCommandPin is turned off and the process ends.
(please refer to appendix 6 - Figure (7))</td></tr>
<tr><td>9</td><td>SpindleCover-SensorPin</td><td>The input is the spindle cover jack sensor. When this input is activated, the on/off status of the spindle cover is determined.</td></tr>
<tr><td>10</td><td>SurfaceDetec-torPin</td><td>The input is the surface detection sensor. When this input is activated, the workpiece level is determined. This input is related to the Z axis of the rotary machine.
The SurfaceDetector sensor is usually placed on the workpiece so that after the tool contacts the sensor, the workpiece level is determined.</td></tr>
<tr><td>11</td><td>SurfaceScanSen-sorPin</td><td>The input is the surface scan sensor. This input is related to the SurfaceScan function. For more information, please refer to table 4.</td></tr>
<tr><td>12</td><td>SurfaceScan-JackSensorPin</td><td>The input is the scan jack sensor. This input detects the activation of the jack.
This input is related to the SurfaceScan function. For more information, please refer to table 4.</td></tr>
</table>

<table>
<tr>
<td>13</td>
<td>T-CC-S1Pin
T-HSD-S1Pin
T-TEKNO-S2Pin
T-ToolSensorPin</td>
<td>The input is the tool presence sensor in the spindle collet. When this input is activated, the presence or absence of the tool in the spindle collet is determined.
<table>
<tr><td>T-CC-S1Pin</td><td>It is used for CC brand spindles</td></tr>
<tr><td>T-HSD-S1Pin</td><td>It is used for HSD brand spindles</td></tr>
<tr><td>T-TEKNO-S2Pin</td><td>It is used for TEKNO brand spindles</td></tr>
<tr><td>T-ToolSensorPin</td><td>It is used for spindles other than CC, HSD, and TEKNO brands.</td></tr>
</table>
</td>
</tr>
<tr>
<td>14</td>
<td>T-CC-S2Pin
T-HSD-S2Pin
T-TEKNO-S1Pin
T-ToolHolderSen-sorPin</td>
<td>This input is the spindle clamp sensor which detects the open or closed state of the spindle clamp. When this input is activated, it determines whether the spindle clamp is open or closed.
<table>
<tr><td>T-CC-S2Pin</td><td>It is used for CC brand spindles</td></tr>
<tr><td>T-HSD-S2Pin</td><td>It is used for HSD brand spindles</td></tr>
<tr><td>T-TEKNO-S1Pin</td><td>It is used for TEKNO brand spindles</td></tr>
<tr><td>T-ToolHolderSensorPin</td><td>It is used for spindles other than CC, HSD, and TEKNO brands.</td></tr>
</table>
</td>
</tr>
<tr>
<td>15</td>
<td>T-CC-S3Pin
T-HSD-S3Pin
T-TEKNO-S3Pin
T-InverterStop-Pin
T-SpindleRota-tionSensorPin</td>
<td>he input is a sensor for detecting the rotation of the spindle. When this input is activated, the status of the spindle rotation is determined. In other words, when the spindle is rotating, this input turns on and off once for each revolution of the spindle.
<table>
<tr><td>T-CC-S3Pin</td><td>It is used for CC brand spindles</td></tr>
<tr><td>T-HSD-S3Pin</td><td>It is used for HSD brand spindles</td></tr>
<tr><td>T-TEKNO-S3Pin</td><td>It is used for TEKNO brand spindles</td></tr>
<tr><td>T-InverterStopPin</td><td>When the outputs of some inverters are used to determine the rotation status of the spindle.</td></tr>
<tr><td>T-SpindleRotationSensorPin</td><td>It is used for spindles other than CC, HSD, and TEKNO brands.</td></tr>
</table>
</td>
</tr>
<tr>
<td>16</td>
<td>Numberm,
T-Jack3Sensor-Pin</td>
<td>The input T-Jack3SensorPin is used during tool change. When this input is activated, the status of the T-Jack3Pin output is displayed (on or off). Using this input increases the safety of the machine and reduces the possibility of errors. If T-Jack3SensorPin is not used, a parameter called Jack3Delay is activated in the Settings window under the ToolChanger branch, where the delay time when turning on or off the T-Jack3Pin output can be entered. This input is related to the T-Jack3Pin output. Please refer to Table 3 for more information.
<table>
<tr><td>T-Jack3SensorPin, 0</td><td>If this input is defined using the argument zero, when the T-Jack3Pin output is turned off, it can be confirmed whether this output is off or not, and the tool change process will continue until feedback is received through the T-Jack3SensorPin input.</td></tr>
<tr><td>T-Jack3SensorPin, 1</td><td>If this input is defined using argument 1, when the T-Jack3Pin output is turned on, it is possible to confirm the on/off status of this output and the tool change process will be stopped until feedback is received through the T-Jack3SensorPin input.</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td>17</td><td>Number, T-JackSensorPin</td><td>The input is a jack sensor used during tool changing. When this input is activated, the status of the T-JackPin output is displayed as on or off. Using this input increases the safety of the device and reduces the likelihood of errors.
If this input is not used, a parameter called JackDelay is activated in the ToolChanger section under the Settings window. It is possible to enter the delay time for turning on or off the T-JackPin output.
This input is related to the T-JackPin output. Please refer to Table 3 for more information.<table><tr><td>T-JackSensorPin, 0</td><td>If this input is defined using the argument 0, when the T-JackPin output is turned off, it is possible to confirm the on/off status of this output and the tool change process will be stopped until feedback is received through the T-JackSensorPin input.</td></tr><tr><td>T-JackSensorPin, 2</td><td>If this input is defined using argument 1, when the T-JackPin output is turned on, you can confirm the on/off status of this output, and the tool change process will continue until feedback is received through the T-JackSensorPin input.</td></tr></table></td></tr>
<tr><td>18</td><td>T-SpindleHomePin</td><td>The input is used to determine the Home position of the spindle tool. In some spindle machines that have tool changing capability, the tool must be placed in a specific position or the Home position before continuing the tool change process. This input is used to ensure that the tool is first placed in the correct position before continuing the tool change process. Typically, the T-SpindleHomePin sensor is connected to the spindle structure.
This input is related to the T-SpindleLockPin output. Please refer to table 3 for more information.</td></tr>
<tr><td>19</td><td>T-ToolHeightSensorPin</td><td>When this input is executed, the height of the tool is determined. This input is related to the output called T-ToolHeightPin. Please refer to table 3 for more information on this output.

This input is related to the T-SetToolHeight function. Please refer to table 4 for more information on this function.</td></tr>
</table>

Table(2)

<table>
<tr><th></th><th>Name of
the Element</th><th>Descriptions</th></tr>
<tr><td>1</td><td>AlarmPin</td><td>If the ClearAlarm function is executed, this output will be turned on for 0.5 seconds.</td></tr>
<tr><td>2</td><td>Axis name,
AlarmStatusPin</td><td>In case of an alarm occurring in the designated axis, the output is turned on.<table><tr><td>Axis name, AlarmStatus-Pin</td><td>The name of the axis can be any of the X, Y, Z, A, B, or C axes.</td></tr></table></td></tr>
<tr><td>3</td><td>Axis name,
AlarmStatusPin</td><td>If the direction of movement for the designated axis is positive, this output turns on, and if the movement is in the negative direction, the output turns off.<table><tr><td>Axis name, AxisDirection-Pin</td><td>The name of the axis can be any of the X, Y, Z, A, B, or C axes.</td></tr></table></td></tr>
<tr><td>4</td><td>Number, Axis name, AlarmStatusPin</td><td>X4 is a hardware designed by the company Radonics, which has the capability of switching a single axis to four axes and includes 4 input channels. By connecting the digital outputs of a controller to the X4 inputs, it enables the control of up to 4 motors completely separately on a single axis. For example, on the Z-axis, it is possible to connect up to 4 motors and turn them on or off separately.
(please refer to appendix 6 - Figure (8))

Note that these four outputs can also be used in combination. In other words, any combination of them can be turned on or off simultaneously. To do this, simply turn on the output connected to the desired X4 input.
Note that this output is related to the AxisSwitch function. Please refer to table 4, row 2 for more information.<table><tr><td>Number, Axis name, AxisSwitchPin,</td><td>The name of the axis connected to the X4 input can be any of the X, Y, Z, A, B, or C axes.
The number can be from 1 to 4, each of which corresponds to the input number of the designated axis.</td></tr></table></td></tr>
<tr><td>5</td><td>BinaryOutPort-Pin, Number</td><td>By executing the command, the output is turned on in binary format. For example, it can be used to control the spindle speed in binary format. Suppose an inverter has four inputs for speed control, and it is sufficient to define four outputs with the links BinaryOutPortPin, 0, BinaryOutPortPin, 1, BinaryOutPortPin, 2, and BinaryOutPortPin, 3, and connect them to the inputs of the inverter in order.<table><tr><td>Number, BinaryOutPortPin</td><td>The number can be from 1 to 32</td></tr></table></td></tr>
<tr><td>6</td><td>BrakePin</td><td>After the controller goes online and the motors are powered, executing the BrakePin command will turn on the output connected to the brake relay, releasing the brake 1 second after the motors are powered on.</td></tr>
<tr><td>7</td><td>ControlStatusPin, Number</td><td>Various states of the controller and software can be viewed<table><tr><td>ControlStatusPin, Number</td><td>The number can be from 1 to 16</td></tr><tr><td>ControlStatusPin, 1</td><td>When the controller goes online, the output turns on.</td></tr><tr><td>ControlStatusPin, 2</td><td>When a file is opened, the output turns on.</td></tr><tr><td>ControlStatusPin, 3</td><td>During the process of executing the Home command, the output turns on.</td></tr><tr><td>ControlStatusPin, 4</td><td>During program execution, the output turns on.</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<tr><td>ControlStatusPin, 5</td><td>When the Emergency input is turned on, the output turns on.</td></tr>
<tr><td>ControlStatusPin, 6</td><td>When the axes have an alarm, the output turns on.</td></tr>
<tr><td>ControlStatusPin, 7</td><td>When the registry code is entered, the output turns on.</td></tr>
<tr><td>ControlStatusPin, 8</td><td>When the software is in simulation mode, the output turns on.</td></tr>
<tr><td>ControlStatusPin, 9</td><td>When the Home operation is complete, the output turns on.</td></tr>
<tr><td>ControlStatusPin, 10</td><td>When G-Code is executed in forward order, the output turns on.</td></tr>
<tr><td>ControlStatusPin, 11</td><td>When G-Code is executed in backward order, the output turns on.</td></tr>
<tr><td>ControlStatusPin, 12</td><td>When the Hold function is enabled, the output turns on. Please refer to table 4, row 27 for more information.</td></tr>
<tr><td>ControlStatusPin, 13</td><td>When Remote is enabled, the output turns on.</td></tr>
<tr><td>ControlStatusPin, 14</td><td>When the limit input of any of the axes is activated, the output turns on.</td></tr>
<tr><td>ControlStatusPin, 15</td><td>When an axis is in Jog operation, the output turns on.</td></tr>
<tr><td>ControlStatusPin, 16</td><td>When program execution is complete, the output turns on.</td></tr>
</table>
</td></tr>
<tr><td>8</td><td>ConveyorPin</td><td>By executing the command to turn on or off the conveyor motor, this output will turn on or off.
When the conveyor sensor (ConveyorSensorPin) in the inputs is activated, this output will be automatically turned off.
There is also a function related to turning on the Conveyor, please refer to row 10 in table 4 for more information.</td></tr>
<tr><td>9</td><td>CoolantPin</td><td>By executing the command to turn on or off the cooling system of the machine, this output will turn on or off.
Usually, this output can be controlled through the M code (M8).
For the operation of this pin, there is a function called Coolant. For more information, please refer to row 11 in table 4.</td></tr>
<tr><td>10</td><td>Axis name, EnabledStatusPin</td><td>When the designated axis is enabled, this output is turned on.
<table>
<tr><td>Axis name, EnabledStatusPin</td><td>The axis name can be any of X, Y, Z, A, B, or C.</td></tr>
</table>
</td></tr>
<tr><td>11</td><td>Axis name, ErrorStatusPin</td><td>When the specified axis has an alarm or limit, this output turns on.
<table>
<tr><td>Axis name, ErrorStatusPin</td><td>The axis name can be any of the axes X, Y, Z, A, B, C.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr>
<td>12</td>
<td>InterruptEndPin Number, InterruptEndPin</td>
<td>
InterruptEndPin is only used in a situation where the entire process is executed in case of two M codes or two functions. Note that if InterruptStartSensorPin and InterruptEndSensorPin are not defined for inputs, the Interrupt process is executed without the need for the input sensor to be triggered or an expression to be True.

Considering that the Interrupt process is completely related to the input, output, and function links, the complete operation of the process is explained in this section, and in order to be fully clear, the used pin is mentioned when defining the process, and it is enough to define the links according to the pin name or function and refer to the corresponding table.

In addition, the order of operations and the activation of inputs, outputs, and functions are shown in a diagram for greater clarity. The links are displayed with abbreviations: (Input Link -- I), (Output Link -- O), (Function Link -- F).

<table>
<tr><td>InterruptEndPin</td><td>By executing this function, only one Interrupt process is performed.
In fact, in this function, which does not have any arguments, the program considers the argument number as zero.</td></tr>
<tr><td>InterruptEndPin, Number</td><td>By executing this function, only one interrupt process is performed. In fact, in this function that does not have any number as an argument, the program considers the argument number as zero.

The entered number as an argument is actually for defining different interrupt processes to perform different operations. In this case, all Interrupt links (input link, output link, function) used for a process must have the same arguments. For example, all their arguments are 4.

The number can be from 0 to 15.</td></tr>
</table>
In a case where the entire process is executed in response to two M codes or two functions:
When the InterruptStart command (function link) is executed, it first waits for the InterruptStartSensorPin input link to be triggered or in other words become True, and when that happens, it issues the command to turn on the InterruptPin (output link) and InterruptStartPin (output link), and the first stage of the process is completed.
(please refer to appendix 6 - Figure (9))

When the InterruptEnd (function link) command is executed, first the InterruptPin (output link) and InterruptStartPin (output link) are turned off, and then the InterruptEndPin (output link) is turned on and waits for the InterruptEndSensorPin (input link) to be triggered or in other words become true. When the InterruptEndSensorPin (input link) is triggered, the InterruptEndPin (output link) is turned off and the final stage of the process ends.
(please refer to appendix 6 - Figure (10))
</td>
</tr>
</table>

<table>
<tr>
<td>13</td>
<td>InterruptPin
Number, Inter-
ruptPin</td>
<td>
InterruptPin is used for the following two cases:

When the entire process is executed in case of a single M code or a single function.
When the entire process is executed in case of two M codes or two functions.
Note that if InterruptStartSensorPin and InterruptEndSensorPin are not defined for the inputs, the Interrupt process will be executed without the need for the input sensor to be triggered or True.

As the Interrupt process in the input, output, and function links are fully related to each other, the entire process is explained in detail in this section. To make it clear, the link used is mentioned when defining the process, and it is sufficient to refer to the corresponding table and define the links according to the pin name or function.
Also, in the diagram that is drawn for better clarity, the order of operations and the activation of inputs, outputs, and functions are shown. The links are displayed with abbreviation symbols.
(I: input link), (O: output link), (F: function link)
<table>
<tr><td>InterruptPin</td><td>Executing this function only performs one Interrupt process. In fact, in this function, which has no arguments, the program assumes the argument number is zero.</td></tr>
<tr><td>Number, Interrupt-Pin</td><td>Executing this function only performs one Interrupt process. In fact, in this function, which has no arguments, the program assumes the argument number is zero.</td></tr>
</table>
In the case where the entire process is executed upon a command of an M-code or a function, when the Interrupt command (function link) is executed, it first waits for the input of InterruptStartSensorPin (input link), and when the input of InterruptStartSensorPin (input link) is triggered or True, the command to turn on InterruptPin (output link) is issued, and then it waits for the input of InterruptEndSensorPin (input link), and if InterruptEndSensorPin (input link) is triggered or True, InterruptPin (output link) is turned off and the process ends.
(please refer to appendix 6 - Figure (11))
In the case where the entire process is executed upon a command of two M-codes or two functions, when the InterruptStart command (function link) is executed, it first waits for the input of InterruptStartSensorPin (input link), and when the input of InterruptStartSensorPin (input link) is triggered or True, the commands to turn on InterruptPin (output link) and InterruptStartPin (output link) are issued, and the first stage of the process ends.
(please refer to appendix 6 - Figure (12))

When the InterruptEnd command (function link) is executed, first the InterruptPin (output link) and InterruptStartPin (output link) are turned off, and the InterruptEndPin (output link) is turned on, then it waits for the input of InterruptEndSensorPin (input link), and if InterruptEndSensorPin (input link) is triggered or True, InterruptEndPin (output link) is turned off and the final stage of the process ends.
(please refer to appendix 6 - Figure (13))
</td>
</tr>
<tr>
<td>14</td>
<td>InterruptStratPin
Number, Inter-
ruptStratPin</td>
<td>
InterruptStartPin is only used in the case where the entire process is executed upon a command of two M-codes or two functions.
Note that if InterruptStartSensorPin and InterruptEndSensorPin are not defined as inputs, the Interrupt process will be executed without the need for the input sensor to be triggered or True.
Since the Interrupt process in the input, output, and function links are all interrelated, the complete operation of the process is explained in this section. To make it completely clear, the links used are mentioned when defining the process, and it is sufficient to refer to the corresponding table and define the links based on the pin or function name.
In the diagram, drawn for more clarity, the sequence of operations and the activation of inputs, outputs, and functions are shown.
Links are represented by abbreviations
(input link -- I), (output link -- O), (function link -- F)
</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<tr><td>InterruptStratPin</td><td>By running this function, only one Interrupt process is executed. In fact, in this function, which does not have any argument, the program considers the argument number as zero.</td></tr>
<tr><td>Number, InterruptStratPin</td><td>The number entered as an argument is actually used to define different Interrupts for different processes.
In this case, all Interrupt links (input link, output link, function link) used for a process must have the same arguments. For example, all of them should have the argument 4.
The number can be from 0 to 15.</td></tr>
</table>
In the case where the entire process is executed by the command of two M codes or two functions:

When the InterruptStart command (function link) is executed, it first waits for the InterruptStartSensorPin input (input link) to be triggered, or in other words, to become True. Once the InterruptStartSensorPin input is triggered, the command to turn on the InterruptPin output (output link) and InterruptStartPin output (output link) is issued and the first stage of the process is completed.
(please refer to appendix 6 - Figure (14))

In the next stage, when the InterruptEnd command (function link) is executed, the InterruptPin output (output link) and InterruptStartPin output (output link) are first turned off, and the InterruptEndPin output (output link) is turned on. Then, it waits for the InterruptEndSensorPin input (input link) to be triggered, and if the Interrupt-EndSensorPin input is triggered or, in other words, becomes True, the InterruptEndPin output (output link) is turned off, and the final stage of the process is completed.
(please refer to appendix 6 - Figure (15))
</td></tr>
<tr><td>15</td><td>Number, Jog-ModePin</td><td>
To display the Jog mode, if the entered argument is proportional to the Jog mode, the output will be turned on. For example, if the argument is zero and the Jog mode is Continuous, the output will be turned on.
<table>
<tr><td>JogMode-Pin,0</td><td>When Jog mode is set to Continuous, the output will be turned on.3</td></tr>
<tr><td>JogMode-Pin,1</td><td>When Jog mode is set to Incremental, the output will be turned on.</td></tr>
</table>
</td></tr>
<tr><td>16</td><td>Axis name, Jog-StatusPin</td><td>
If the specified axis is in Jog mode or being jogged, the output will be turned on.
<table>
<tr><td>Axis name, JogStatusPin</td><td>The axis name can be any of the X, Y, Z, A, B, or C axes.</td></tr>
</table>
</td></tr>
<tr><td>17</td><td>LaserPointerPin</td><td>In a device equipped with a pointer laser, this output is used. When this output is turned on, the pointer laser is turned on as well.</td></tr>
<tr><td>18</td><td>Axis name, Lim-itNStatusPin</td><td>
When the specified axis is at the negative limit, this output is turned on.
<table>
<tr><td>Axis name, LimitNStatus-Pin</td><td>The axis name can be any of the X, Y, Z, A, B, or C axes.</td></tr>
</table>
</td></tr>
<tr><td>19</td><td>Axis name, Limit-PStatusPin</td><td>
When the specified axis is at the positive limit, this output is turned on.
<table>
<tr><td>Axis name, LimitPStatus-Pin</td><td>The axis name can be any of the X, Y, Z, A, B, or C axes.</td></tr>
</table>
</td></tr>
<tr><td>20</td><td>Axis name, Limit-StatusPin</td><td>When the limit is taken from the specified axis, this output becomes clear.</td></tr>
<tr><td>21</td><td>Lubrication-PumpPin</td><td>This output is activated when the oiling command is executed.</td></tr>
</table>

<table>
<tr><td>22</td><td>R-CornerJackPin</td><td>This output is activated when the spindle orientation command is executed.</td></tr>
<tr><td>23</td><td>ReadyPin</td><td>When the controller goes online, this output is turned on.</td></tr>
<tr><td>24</td><td>Axis name, ReadyStatusPin</td><td>When the specified axis is in Ready state and there is no active Alarm condition, this output is turned on.<table><tr><td>Axis name, ReadyStatusPin</td><td>The axis name can be any of the X, Y, Z, A, B, or C axes.</td></tr></table></td></tr>
<tr><td>25</td><td>Axis name, RunStatusPin</td><td>When the specified axis is in Run state, this output is turned on.<table><tr><td>Axis name, RunStatusPin</td><td>The axis name can be any of the X, Y, Z, A, B, or C axes.</td></tr></table></td></tr>
<tr><td>26</td><td>SelectPin, Number</td><td>It is possible to define any number of outputs as needed. When one output is turned on, the other defined outputs are turned off. In fact, only one of the outputs can be turned on in that unit at a time.
This output is associated with a function called “Select”, which allows the user to define Toggle keys as desired (up to a maximum of 32) and by pressing this key, the corresponding output defined by the SelectPin is turned on, and if another output has been defined by the SelectPin, it will be automatically turned off. For example, by defining three Toggle keys with links Select,1, Select,2, and Select,3 and connecting them to three outputs with links SelectPin,1, SelectPin,2, and SelectPin,3, when the Select,1 key is pressed, output 1 is turned on, and outputs 2 and 3 are turned off automatically.<table><tr><td>SelectPin, Number</td><td>The number can be between zero and 32.</td></tr></table></td></tr>
<tr><td>27</td><td>SimpleCommandPin, Number</td><td>Due to the complete relationship between the SimpleCommand process in the input, output, and function links, the operation of the entire process is fully explained in this section. To be clear, the link used is mentioned when defining the process, and it is sufficient to refer to the corresponding table and define the links based on the name of the pin or function. In the diagram, which shows the order of operations and the activation of inputs, outputs, and functions more clearly, the links are displayed with abbreviations.

((Input link -- I), (Output link -- O), (Function link -- F))

When the SimpleCommand command (Function link) is executed, first the SimpleCommandPin (Output link) is turned on, and then it waits for the SimpleCommandSensorPin input (Input link) to be triggered, and if the SimpleCommandSensorPin input is triggered, or in other words, becomes True, the SimpleCommandPin (Output link) is turned off, and the process is completed.
(please refer to appendix 6 - Figure (16))</td></tr>
<tr><td>28</td><td>SimpleDelayPin</td><td>As the SimpleDelayPin command is fully related to the SimpleDelay function, the operation of the entire process is explained in this section. To be clear, the link used is mentioned when defining the process, and it is sufficient to refer to the corresponding table and define the links based on the name of the pin or function.
When the SimpleDelay command (Function link) is executed, based on the argument entered in the SimpleDelay function, the SimpleDelayPin output (Output link) is turned on or off, and then a delay is created for the amount of time set in the SimpleDelay parameter in the General branch in the Setting window.</td></tr>
<tr><td>29</td><td>SpindleCCWPin</td><td>The Spindle direction of movement is determined using it. This output is similar to the output called SpindleCWPin, with the difference that its direction of movement is opposite to the direction of rotation of SpindleCWPin. In fact, their functions are opposite to each other.</td></tr>
</table>

30	SpindleCWPin	The Spindle direction of movement is determined using it. This output is similar to the output called SpindleCCWPin, with the difference that its direction of movement is opposite to the direction of rotation of SpindleCCWPin. In fact, their functions are opposite to each other.
31	SpindleCooler-Pin	The output is turned on or off with the turning on or off of the Spindle fan
32	SpindleCoverPin	The output is turned on or off with the opening or closing of the Spindle cover.
33	SurfaceScan-JackPin	When the Surface Scan command is issued, the SurfaceScanJackPin output is turned on, and when the surface scanning process is finished, the SurfaceScanJackPin output is turned off. This output is associated with an input called SurfaceScanJackSensorPin. For more information, please refer to Table 2.
34	T-Jack3Pin	Machines use a jack as part of the tool changing process, such that when the tool change command is issued and the machine is in the desired position, the jack becomes active and moves the tool holder forward, thereby changing the tool. This is used, for example, to change rotary tools installed on the Z and Y axes. This output is associated with an input called T-Jack3SensorPin. For more information, please refer to Table 2.
35	T-JackPin	When the tool change command is issued, the T-JackPin output is turned on, and when the tool change process is complete, the T-JackPin output is turned off. For example, the tools may be held in a collet, and a jack may be needed to move the collet forward to access the spindle. Depending on the mechanical structure of the machine, the jack can be used to move the tool holders, tool covers, or the entire tool changing system forward and backward by turning the jack on or off. This output is associated with an input called T-JackSensorPin. For more information, please refer to Table 2.
36	T-NormalTool-ChangingPin	When the tool change command is issued, the T-NormalToolChangingPin output is turned on, and when the tool change process is complete, the T-NormalToolChanging-Pin output is turned off. This output is similar to the output called T-ToolChangingPin, but is used for some tool changes that use clamps.
37	T-SpindleLock-Pin	The Spindle Lock output is used in some tool-changing spindles where the type of collet used is important for positioning and rotating the tool. In this case, the spindles must first position the tool in a special position or the tool home point. To do this, a sensor called T-SpindleHomePin is defined, which positions the tool in a special position, and then, if the T-SpindleLockPin output is defined and the tool is in the desired position, the output is turned on. This is done to lock the tool in that special position, so that it does not move from that position. Usually, the T-SpindleHomePin sensor is connected to the spindle structure. This output is associated with an input called T-SpindleHomePin. For more information, please refer to Table 2.
38	T-ToolChang-ingPin	When the tool change command is executed, this output is turned on, and when the tool change process is complete, it is turned off. This output is similar to the T-NormalToolChangingPin output, but can be used for both normal and Head Changer tools.
39	T-ToolCleaner-Pin	This output is used to activate an electric valve connected to the tool cleaner. During a tool change, before the machine removes the selected tool, this output is turned on to clean the tool, and then the machine proceeds to remove the tool.
40	T-ToolHeightPin	This output is turned on when the T-SetToolHeight process starts and turned off when the T-SetToolHeight process is complete. For example, a tool height sensor can be connected to a jack, and this output can be used to move the sensor in or out of a compartment by turning the jack on or off, in order to prevent the sensor from getting dirty or damaged. This output is associated with an input called T-ToolHeightSensorPin. For more information, please refer to Table 2. This output is related to a function called T-SetTool-Height. For more information, please refer to Table 4.

<table>
<tr><td>41</td><td>T-ToolHolderPin</td><td>When the command to hold or release a tool inside the spindle is executed, this output turns on or off. This output is similar to the output called T-ToolLockPin, with the difference that T-ToolHolderPin releases the tool when turned on, while T-ToolLockPin holds the tool in place and prevents it from being released when turned on. In fact, their functions are opposite to each other.
This output is related to the T-ToolHolder function. Please refer to Table 4 for more information.</td></tr>
<tr><td>42</td><td>T-ToolLockPin</td><td>When the command to hold or release the tool inside the spindle is executed, this output is turned on or off. This output is similar to the output called T-ToolHolderPin, but the T-ToolLockPin keeps the tool inside the spindle when it is turned on, while the T-ToolHolderPin releases the tool when it is turned on. In fact, their functions are opposite to each other.</td></tr>
<tr><td>43</td><td>T-ToolPin, Tool number

T-ToolPin, Tool number, Tool number</td><td>If the tool number matches the value entered as an argument, the defined output with T-ToolPin link will be turned on.<table><tr><td>T-ToolPin, Tool number</td><td>When the tool being used matches the tool number entered as an argument, the defined output is turned on.</td></tr><tr><td>T-ToolPin, Tool number, Tool number</td><td>In this link, multiple arguments can be entered simultaneously. When the tool being used matches any of the tool numbers entered as arguments, the defined output is turned on.</td></tr></table></td></tr>
<tr><td>44</td><td>VaccumPin</td><td>By executing the command to turn on or off the Vaccum, this output is turned on or off.</td></tr>
<tr><td>45</td><td>Zone</td><td>When the axes are within the desired range, the output is turned on or off. And it is possible to define any number of outputs.<table><tr><td>Zone, Axis Name, Number, Number</td><td>The name of the axis can be any of the X, Y, Z, A, B, or C axes. The first number specifies the lower limit and the second number specifies the upper limit of the designated axis. For example, Zone, X, 10, 100 means that if the X axis is between coordinates 10 and 100, the output will turn on.</td></tr><tr><td></td><td>The name of the axis can be any of the X, Y, Z, A, B, or C axes. Also, there is no need for a comma when writing the axes. The first number specifies the lower limit of the first axis, the second number specifies the upper limit of the first axis, the third number specifies the lower limit of the second axis, the fourth number specifies the upper limit of the second axis, and so on for the third, fourth, fifth, and sixth axes. When writing the numbers, a comma is needed after each number.
For example, Zone XY 10 100 20 200 means that if the X axis is between coordinates 10 and 100, and the Y axis is between coordinates 20 and 200, the output will turn on.</td></tr><tr><td>Zone, Axis Name!, Number, Number</td><td>The exclamation mark causes the output to turn off at the specified coordinates. The name of the axis can be any of the X, Y, Z, A, B, or C axes, and there is no need for a comma when writing the axes. The first number specifies the lower limit of the first axis, the second number specifies the upper limit of the first axis, the third number specifies the lower limit of the second axis, the fourth number specifies the upper limit of the second axis, and so on for the third, fourth, fifth, and sixth axes. When writing the numbers, a comma is needed after each number.

For example, Zone, XY, 10, 100, 20, 200 means that if the X axis is between coordinates 10 and 100 and the Y axis is between coordinates 20 and 200, the output will turn off.</td></tr></table></td></tr>
</table>

Table(3)

<table>
<tr><th></th><th>Name of
the Element</th><th>Descriptions</th></tr>
<tr><td>1</td><td>AJog, Axis name</td><td>Absolute Jog is an abbreviation for manually moving to a specific coordinate, also known as jogging. In this function, we only specify the desired axis for manual movement. The AJog link is related to the parameter link, and all you need to do is enter the coordinates in the Value field, and by executing this function, the desired axis will be moved to the specified coordinates.<table><tr><td>AJog, Axis name</td><td>The name of the axis can be any of the X, Y, Z, A, B, or C axes</td></tr></table></td></tr>
<tr><td>2</td><td>AxisSwitch,
Axis name,
Number</td><td>A hardware called X4 is embedded, which includes four input channels. By connecting the outputs to the inputs of X4, up to four motors in a single axis can be controlled separately. For example, on the Z-axis, up to four motors can be connected and turned on or off separately. With the motors turned on, the corresponding output pulse for the Z-axis is transmitted to those motors.
(please refer to appendix 6 - Figure (17))

Please note that these four outputs are also used in combination. In fact, each one can be turned on or off simultaneously, and it is sufficient to turn on or off the output connected to the desired X4 input. By executing this function, the outputs connected to the X4 hardware inputs can be turned on or off. Please note that this function is related to the AxisSwitchPin output link (Table 3).<table><tr><td>AxisSwitch, Axis name, Number</td><td>The name of the axis can be any of the X, Y, Z, A, B, or C axes. The number can be from 1 to 4, and each number represents the input number of the X4 hardware on the specified axis.</td></tr></table></td></tr>
<tr><td>3</td><td>BlockNumber</td><td>By executing this function, the program is moved to the line number specified in the G-Code. To enter the line number, a Value field can be used.</td></tr>
<tr><td>4</td><td>Brake</td><td>By executing this function, the motor brake is manually turned on or off.<table><tr><td>Brake</td><td>By selecting the Toggle key and executing this function, the output of the Brake is turned on if it is off, and turned off if it is on.</td></tr><tr><td>Brake, 0</td><td>By executing this function, the output related to this function is turned off.</td></tr><tr><td>Brake, 1</td><td>By executing this function, the output related to this function is turned on.</td></tr><tr><td>Brake, 2</td><td>By executing this function, the output of the Brake is turned on if it is off, and turned off if it is on.</td></tr></table></td></tr>
<tr><td>5</td><td>CJog, Axis name</td><td>To define Jog or Continuous manual motion mode.<table><tr><td>CJog, Axis name</td><td>The name of the axis can be any of the X, Y, Z, A, B, or C axes.</td></tr><tr><td>CJog, Axis name -</td><td>The negative sign is used to move in the negative direction of the axes. The name of the axis can be any of the X, Y, Z, A, B, or C axes.</td></tr></table></td></tr>
<tr><td>6</td><td>Capture</td><td>Reserve</td></tr>
<tr><td>7</td><td>ClearAlarm</td><td>By executing this function, the motor alarm can be cleared. Note that to use this function, the motor in question must have hardware capability to clear the alarm.</td></tr>
</table>

<table>
<tr><td>8</td><td>ClearPoints</td><td>By executing this function, all points saved by the SavePoint function can be cleared. The SavePoint function is described in row 68 of the same table. Please refer to it for more information,</td></tr>
<tr><td>9</td><td>ClearToolPath</td><td>By executing this function, you can delete the file that has been displayed in ToolPath,</td></tr>
<tr><td>10</td><td>Conveyor, Axis name</td><td>By executing this function, you can turn the conveyor on or off. If the conveyor sensor (ConveyorSensorPin) defined in the input links is available, the conveyor output can be automatically turned off.</td></tr>
<tr><td></td><td></td><td><table>
<tr><td>Conveyor</td><td>This function turns the conveyor output on or off when executed.</td></tr>
<tr><td>Conveyor, Axis name</td><td>By executing this function, the conveyor output in the specified axis can be turned on or off. The axis name can be any of the X, Y, Z, A, B, or C axes.</td></tr>
</table></td></tr>
<tr><td>11</td><td>Coolant, Number</td><td>This function turns on or off the output related to the cooling system of the machine. In a router machine, the air power switch is connected to it, and with the CoolantPin output, the power switch can be turned on or off. Additionally, the output can be controlled through the M code (M8).<table>
<tr><td>Coolant, 0</td><td>By running this function, the output related to this function is turned off.</td></tr>
<tr><td>Coolant, 1</td><td>By running this function, the output related to this function is turned off.</td></tr>
<tr><td>Coolant, 2</td><td>This function turns the output related to this function on if it is off and turns it off if it is on. The output in this case is the coolant output.</td></tr>
</table></td></tr>
<tr><td>12</td><td>Displace, Number</td><td>By executing this function, without running the program, the axes are moved to a specific point on the machine according to the entered argument.<table>
<tr><td>Displace, -1</td><td>By running this function, the axes are moved to the point where the program should be executed without actually running the program, based on the input argument.</td></tr>
<tr><td>Displace, 0</td><td>By executing this function, the axes are moved to the Home position</td></tr>
<tr><td>Displace, 1</td><td>By executing this function, the axes are moved to the zero point of the workpiece or the reference point.</td></tr>
<tr><td>Displace, 2</td><td>By executing this function, the axes are moved to the coordinates defined in the AJog function. Please refer to row 1 of this table for more information. In this case, all axes are moved simultaneously to the defined coordinates.</td></tr>
<tr><td>Displace, 3</td><td>By running this function, the axes are moved to the coordinates of Park which are specified in the General section of the Setting window. Note that if Park is enabled, the axes will automatically move to the Park point after the program is finished. Otherwise, this command can be used to move the axes to the Park point.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>13</td><td>DisplaceReference, Axis Name</td><td>By defining this function, the RealTimeReferenceDisplace parameter is added in the General section of the Setting window, and if the user enables or, in other words, sets it to True, they can use this function during program execution, and if it is False, they must first stop program execution and then use the DisplaceReference function.

During program execution, by running this function, the Reference point of the workpiece can be moved. In other words, it is used to correct errors that the user needs to correct during program execution. For example, during program execution, the user realizes that the design has an error at a specific point and can correct the error by changing the Reference at that moment.<table><tr><td>DisplaceReference,
Axis Name</td><td>In fact, the reference point of the workpiece is displaced by the amount specified in the ReferenceDisplace parameter under the axis branch (X, Y, Z, A, B, C) in the General branch of the Setting window. The name of the axis can be any of X, Y, Z, A, B, or C.</td></tr></table></td></tr>
<tr><td>14</td><td>EditGCode</td><td>By running this function, a window will open that allows you to edit G-code.</td></tr>
<tr><td>15</td><td>Emergency, Text</td><td>The text that the user wants to be displayed in case of an Emergency can be added.<table><tr><td>Emergency, Text</td><td>The text that the user wants to be displayed in case of an Emergency can be added.</td></tr></table></td></tr>
<tr><td>16</td><td>Exit</td><td>This function is used to close the program. Before exiting the program, it checks the Dictionary section to see if any text has been written for “Exit”. If there is any text, it displays that message before exiting the program. If no text is written, and if the program is not running or the machine is not homing or moving manually, it will exit the program without any message.</td></tr>
<tr><td>17</td><td>ExportSetting</td><td>By executing this function, the program settings can be exported.</td></tr>
<tr><td>18</td><td>Feed, Number</td><td>By executing this function, the program execution speed can be changed. The Scroll function uses the Feed function to change the program execution speed, and the user can set the speed value using the Value parameter. Additionally, by providing an argument, the number of steps to increase or decrease the program execution speed can be specified. In other words, by running this function, the program speed is increased or decreased by multiplying the value of the argument with the number of steps determined.<table><tr><td>Feed, Number</td><td>The number entered actually represents the number of steps in the Scroll in the positive direction for increasing the speed.</td></tr><tr><td>Feed, Number-</td><td>The entered number actually represents the steps in the negative direction in the Scroll for decreasing the speed.</td></tr></table></td></tr>
<tr><td>19</td><td>FindCode</td><td>By running this function, you can search for a specific code inside the G-Code program. Essentially, this function opens a page where you can type in the desired code to search for. For example, if the user searches for the code “M30,” it will display in which line of the G-Code program the code “M30” exists.</td></tr>
<tr><td>20</td><td>Flip, Axis Name</td><td>By executing this function, the symmetry of the design is created.<table><tr><td>Flip, X</td><td>It creates symmetry of the design with respect to the X-axis.</td></tr><tr><td>Flip, Y</td><td>It creates symmetry of the design with respect to the Y-axis.</td></tr></table></td></tr>
</table>

<table>
<tr><td>21</td><td>Floor, Number</td><td>By executing this function, you can set a coordinate on the Z-axis that the tip of the tool should not move lower than. Essentially, the Z-axis is moved to the specified point for setting the Floor coordinate, and then with this function, that point is saved. This setting will also be applied to the G-Code by opening the desired G-Code.
If the Reference or G-Code is lower than the Floor level, the values will be cut by the amount specified for the Floor setting.

Note that any changes to the Floor setting will require reopening the file for the command to be applied to the G-Code.<table><tr><td>Floor, 1</td><td>Sets the lowest level of the Z-axis.</td></tr><tr><td>Floor, 0</td><td>This function is used to delete the stored Floor value.</td></tr></table></td></tr>
<tr><td>22</td><td>Frame, Number</td><td>This function draws a rectangle around the design, and the axes move according to the coordinates of the rectangle. It is used to check if the cutting design is included in the workpiece.

The user needs to open the G-Code before applying this function.

By executing this function, the parameter FrameOverReference is added to the General branch, and any value entered in the FrameOverReference parameter moves from the Reference point to the rectangle. For example, if the FrameOverReference parameter is 10 millimeters, it moves 10 millimeters above the Reference point on the path.<table><tr><td>Frame</td><td>It moves away from the Reference point by the amount entered in the FrameOverReference parameter and moves along the path.</td></tr><tr><td>Frame, -1</td><td>This function moves along the path to the lowest point on the Z-axis.</td></tr><tr><td>Frame, 0</td><td>It moves along the path at a distance from the reference point based on the value entered in the FrameOverReference parameter.</td></tr><tr><td>Frame, 1</td><td>It moves on the path at the highest point of the Z-axis.</td></tr></table></td></tr>
<tr><td>23</td><td>GenerateG-Code</td><td>By executing this function, G-Code is generated from the points saved in SavePoint. The SavePoint function is explained in row 68 of this table, please refer to that section for more information.</td></tr>
<tr><td>24</td><td>HandwheelA, Axis Name</td><td>HandwheelA is an abbreviation for Handwheel Axis, which is used to determine the axes on which the handwheel is active. It is possible to create a toggle key for each axis, and by pressing each key, the handwheel of that axis is selected.<table><tr><td>HandwheelA, Axis Name</td><td>The axis name can be any of the X, Y, Z, A, B, or C axes.</td></tr></table></td></tr>
<tr><td>25</td><td>HandwheelX, Number</td><td>By executing this function, the Planck’s constant can be determined. Normally, the Planck’s constant is set to 1, 10, 100, or 1000, but it can be set to any value.<table><tr><td>HandwheelX, Number</td><td>The number is actually the Planck’s constant.</td></tr></table></td></tr>
<tr><td>26</td><td>Hold, Text</td><td>By executing this function, all axis movements (Jog and Run) are halted. In fact, the axes are released from any movement until the cause of Hold (which can be an input sensor or a key defined with the Hold function) is resolved. After the removal of the cause, the axes are re-enabled again.<table><tr><td>Hold, Text</td><td>When executing this function, the text that is written as an argument will be displayed.</td></tr></table></td></tr>
</table>

<table>
<tr><td>27</td><td>HoldJog, Text</td><td>By executing this function, manual movement or Jogging of the axes is disabled. In fact, the axes are released from any Jog movement until the cause of HoldJog (which can be an input sensor or a key defined with the HoldJog function) is resolved. After the removal of the cause, the axes are re-enabled again.<table><tr><td>HoldJog, Text</td><td>When executing this function, the text that is written as an argument will be displayed.</td></tr></table></td></tr>
<tr><td>28</td><td>HoldPreRun, Text</td><td>By executing this function, the axis movements during program execution are halted. In fact, the axes are released from any movement during program execution until the cause of HoldPreRun (which can be an input sensor or a key defined with the HoldPreRun function) is resolved. After the removal of the cause, the axes are re-enabled again.

In this case, if the program is running and the cause of the HoldPreRun command is activated, the program will continue to run until the end of execution and allows the work to be completed, but it does not allow the program to execute the next Run command until the cause of HoldPreRun is resolved.
In this case, if the program is running and the cause of the HoldPreRun command is activated, and the Stop button is pressed, it does not allow the program to execute the Run command.<table><tr><td>HoldPreRun, Text</td><td>When executing this function, the text that is written as an argument will be displayed.</td></tr></table></td></tr>
<tr><td>29</td><td>HoldRun, Text</td><td>By executing this function, the axis movements during program execution are halted. In fact, the axes are released from any movement during program execution until the cause of HoldRun (which can be an input sensor or a key defined with the HoldRun function) is resolved. After the removal of the cause, the axes are re-enabled again.<table><tr><td>HoldRun, Text</td><td>When executing this function, the text that is written as an argument will be displayed.</td></tr></table></td></tr>
<tr><td>30</td><td>Home
Home, Number
Home, Axis name</td><td><table><tr><td>Home</td><td>By executing this function, the axes move in the homing direction to find the home position.</td></tr><tr><td>Home, -1</td><td>Regardless of the home point detection sensor, issuing the Home command with this function is possible at any point it is used.</td></tr><tr><td>Home, -2</td><td>After the program execution is finished, the machine automatically performs the homing operation.</td></tr><tr><td>Home, Axis name</td><td>The axis specified as the argument for this function does not home and only zeroes that axis. The axis name can be any of the X, Y, Z, A, B, or C axes.</td></tr></table></td></tr>
<tr><td>31</td><td>InPort, Number</td><td>By executing this function, the inputs can be turned on or off.<table><tr><td>InPort, Number</td><td>The number written as an argument is actually the input pin number</td></tr></table></td></tr>
<tr><td>32</td><td>Interrupt
Interrupt, Number</td><td>Interrupt is only used in a situation where the entire process is executed when a command for an M code or a function is issued. Note that if InterruptStartSensorPin and InterruptEndSensorPin are not defined for the inputs, the Interrupt process will be executed without the need for the input sensor to be triggered, or in other words, to be true.</td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td>
Considering that the Interrupt process in input, output, and function links are fully related, the operation of the entire process is fully explained in this section, and for clarity, the used link is mentioned when defining the process, and the user needs to refer to the corresponding table based on the pin name or function and define the links. Additionally, in the diagram drawn for better clarity, the order of input, output, and function operations and activations are shown.

Links are displayed with abbreviations.
((Input Link -- I), (Output Link -- O), (Function Link -- F))
<table>
<tr><td>Interrupt</td><td>By executing this function, only one Interrupt process is performed. In fact, in this function which has no argument, the program considers the argument number as zero.</td></tr>
<tr><td>Interrupt, Number</td><td>The number entered as an argument is actually used to define different Interrupts for performing different processes. In this case, all Interrupt links (input link, output link, function link) used for one process must have the same arguments. For example, all of their arguments can be 4.

The number can be from 0 to 15.</td></tr>
</table>
When the entire process is executed in response to an M code or a function command: When the Interrupt command (function link) is executed, the program waits for the InterruptStartSensorPin input link to be triggered or become True. Once this happens, it issues a command to turn on the InterruptPin output link. It then waits for the InterruptEndSensorPin input link to be triggered or become True. If this happens, it turns off the InterruptPin output link, and the process ends.
(please refer to appendix 6 - Figure (18))
</td>
</tr>
<tr>
<td>33</td>
<td>InterruptEnd
InterruptEnd, Number</td>
<td>
InterruptEnd is only used in the case where the entire process is executed upon the command of two M codes or two functions.
Note that if InterruptStartSensorPin and InterruptEndSensorPin are not defined for inputs, the Interrupt process will be executed without the need for input sensors to be triggered or become True.
Since the Interrupt process is fully related to the input, output, and function links, the complete operation of the process is explained in this section, and in order to be completely clear, the links used are mentioned when defining the process, and the user can refer to the relevant table based on the name of the pin or function and define the links. Also, in the diagram drawn for clarity, the order of operations and the activation of inputs, outputs, and functions are shown.

The links are displayed with abbreviated symbols.
((Input Link I), (Output Link O), (Function Link F))
<table>
<tr><td>InterruptEnd</td><td>By running this function, only one Interrupt process is performed. In fact, in this function, which does not have any argument, the program assumes the argument number to be zero.</td></tr>
<tr><td>InterruptEnd, Number</td><td>The number entered as the argument is actually used to define different Interrupts, for performing different processes. In this case, all Interrupt links (input link, output link, function link) used for a process should have the same argument. For example, all of them have 4 as the argument.
The number can be from 0 to 15.</td></tr>
</table>
When the entire process is executed in case of two M codes or two functions command:
</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td>When the InterruptStart (function link) command is executed, it waits for the InterruptStartSensorPin input link to be triggered or in other words become True. Then, the command to turn on the InterruptPin and InterruptStartPin output links is issued, and the first stage of the process is completed.
(please refer to appendix 6 - Figure (19))

When the InterruptEnd (F) command is executed in the case that the entire process is executed by two M codes or two functions, it first waits for the InterruptEndSensorPin (I) input and when the InterruptEndSensorPin input is triggered or in other words set to True, it turns off the InterruptPin and InterruptStartPin (O) outputs and turns on the InterruptEndPin output. Then it waits for the InterruptEndSensorPin input again and when it is triggered or set to True, it turns off the InterruptEndPin output, and the final stage of the process ends.
(please refer to appendix 6 - Figure (20))</td></tr>
<tr><td>34</td><td>InterruptManual</td><td>By executing this function, you can enable or disable the Interrupt process. If InterruptManual is True, the Interrupt process will be completely disabled, and if it is False, it will be enabled.</td></tr>
<tr><td>35</td><td>InterruptStart
InterruptStart, Number</td><td>InterruptStart is only used in cases where the entire Interrupt process is executed by two M codes or two functions commands.
Please note that if InterruptStartSensorPin and InterruptEndSensorPin are not defined as inputs, the Interrupt process will be executed without requiring the input sensor to be triggered or True.
Since the Interrupt process is fully interconnected in input, output, and function links, the entire process operation is fully explained in this section. To make it clear, the used link is mentioned when defining the process, and the user can define the links according to the pin name or function. Also, in the diagram that has been drawn for more clarity, the order of operations and the activation of inputs, outputs, and functions have been illustrated.
Links are displayed with abbreviations.
((Input Link -- I), (Output Link -- O), (Function Link -- F))
<table>
<tr><td>InterruptStart</td><td>The execution of this function only performs one Interrupt process. In fact, in this function, which does not have any argument, the program considers the argument number as zero.</td></tr>
<tr><td>InterruptStart, Number</td><td>If an argument is passed to the function, it is used to define different Interrupt processes for performing different tasks. In this case, all the Interrupt links (input, output, and function) used for a process must have the same arguments. For example, if the argument is 4, all Interrupt links for that process should have an argument of 4.
The argument can range from 0 to 15.</td></tr>
</table>
In the case where the entire process is executed in the event of two M codes or two functions being commanded:
When the InterruptStart command (function link) is executed, it waits for the InterruptStartSensorPin input link, and when the InterruptStartSensorPin input link is triggered or becomes True, the command to turn on the InterruptPin (output link) and InterruptStartPin (output link) is issued, and the first stage of the process is completed.
(please refer to appendix 6 - Figure (21))

When the InterruptEnd command (function link) is executed in the case where the entire process is executed based on two M codes or two functions, first the InterruptPin (output link) and InterruptStartPin (output link) are turned off, and the InterruptEndPin (output link) is turned on. Then, it waits for the InterruptEndSensorPin (input link) and if the InterruptEndSensorPin is triggered or becomes true, it turns off the InterruptEndPin (output link) and the final step of the process is completed.
(please refer to appendix 6 - Figure (22))</td></tr>
</table>

<table>
<tr><td>36</td><td>Jog, Axis name</td><td>To execute this function, a manual movement command, also known as Jog, is given.<table>
<tr><td>Jog, Axis name</td><td>By executing this function, the axis specified as an argument will be moved. The axis name can be any of the X, Y, Z, A, B, or C axes.</td></tr>
<tr><td>Jog, Axis name -</td><td>By executing this function, the axis specified as the argument is moved in the negative direction. The name of the axis can be any of the X, Y, Z, A, B, or C axes.</td></tr>
<tr><td>Jog, Axis name, Axis name</td><td>This function can take multiple arguments simultaneously. By executing this function, the specified axes as arguments will be moved. If a negative sign is also added for each axis, that axis will move in the negative direction of the machine. The name of the axes can be any of X, Y, Z, A, B, or C.</td></tr>
</table></td></tr>
<tr><td>37</td><td>JogFeed, Number</td><td>By executing this function, the speed of manual or jog movement is changed.<table>
<tr><td>JogFeed, Number</td><td>By running this function, the Jog speed is increased by multiplying the Step value by the number written as an argument. For example, if JogFeed,1 is written, the Jog speed is increased by one Step.</td></tr>
<tr><td>JogFeed, Number -</td><td>By executing this function, the Jog speed is decreased by the number specified in the argument multiplied by the Step size. For example, if JogFeed,-1 is written, the Jog speed is decreased by one Step.</td></tr>
</table></td></tr>
<tr><td>38</td><td>JogMod, Number</td><td>By running this function, you can set the state of Jog or manual movement command.<table>
<tr><td>JogMod, 0</td><td>In this mode, Jog is in continuous mode.</td></tr>
<tr><td>JogMod, 1</td><td>In this mode, Jog is in incremental mode.</td></tr>
</table><table>
<tr><td>JogMod, 2</td><td>In this mode, Jog switches from continuous to incremental mode and vice versa with each execution of this function.</td></tr>
<tr><td>JogMod, 3</td><td>With each execution of this function, the Jog speed is increased by a factor of 10.</td></tr>
<tr><td>JogMod, 4</td><td>With each execution of this function, the Jog speed is decreased by a factor of 0.1.</td></tr>
</table></td></tr>
<tr><td>39</td><td>LaserPointer, Number</td><td>By executing this function, the laser pointer can be turned on or off. There is also a folder called “LaserPointer” in the “References” branch, which is used to apply an offset in the X and Y directions for the laser pointer.<table>
<tr><td>LaserPointer, 0</td><td>To turn off the laser pointer, you can execute this function.</td></tr>
<tr><td>LaserPointer, 1</td><td>To turn on the laser pointer, you can execute this function.</td></tr>
<tr><td>LaserPointer, 2</td><td>To toggle the state of the laser pointer (turn on if it’s off, turn off if it’s on), you can execute this function</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>40</td><td>LimitJog, Axis name</td><td>By executing this function, it is possible to limit the manual movement of the axes.<table><tr><td>LimitJog, Axis name</td><td>By executing this function, it is possible to limit the manual movement of the axes.</td></tr><tr><td>LimitJog, Axis name -</td><td>It limits the manual motion course of the specified axis in the negative direction. The axis name can be any of X, Y, or Z.</td></tr></table></td></tr>
<tr><td>41</td><td>Lubricate</td><td>This function performs a one-time oiling process.</td></tr>
<tr><td>42</td><td>Minimize</td><td>This function minimizes or makes the software window smaller.</td></tr>
<tr><td>43</td><td>MoveToEnd, Axis name</td><td>By executing this function, the axis specified as the argument is moved to the end point of the axis.<table><tr><td>MoveToEnd, Axis name</td><td>Moves the specified axis to the positive end of the axis. The name of the axis can be any of X, Y, or Z.</td></tr><tr><td>MoveToEnd, Axis name -</td><td>Moves the specified axis to the negative end of the axis.
The name of the axis can be any of X, Y, or Z.</td></tr></table></td></tr>
<tr><td>44</td><td>MoveToZero, Axis name</td><td>By running this function, the axis specified as an argument will be moved to the zero point.<table><tr><td>MoveToZero, Axis name</td><td>The name of the axis can be any of X, Y, Z, A, B, or C.</td></tr></table></td></tr>
<tr><td>45</td><td>NextFile, Number</td><td>In the GCode folder, there is a parameter called MultiFileEnable which, when enabled, adds a feature to the software that allows opening multiple files simultaneously. In this case, a list of selected files is available to the user, and the user can move between them. This function is used to move between files and control the execution of each selected file. The name of the function is not provided in the given text.<table><tr><td>NextFile, 1</td><td>Running this function selects the files in the list in order, from the first file to the last file.</td></tr><tr><td>NextFile, -1</td><td>Running this function selects the files in the list in reverse order, from the last file to the first file.</td></tr><tr><td>NextFile, -2</td><td>Running this function resets the list, which means that with this argument, the list selects files from the first file again, regardless of the current file.</td></tr></table></td></tr>
<tr><td>46</td><td>OpenFile
OpenFile, File address</td><td>Executing this function opens the file selection window.<table><tr><td>OpenFile</td><td>This function opens the file selection window.</td></tr><tr><td>OpenFile, File address</td><td>It opens the file located at the address specified as an argument.</td></tr></table></td></tr>
<tr><td>47</td><td>OpenMachine, Interface name</td><td>This function allows direct access to the interface specified in the argument without the need to exit the current interface and enter another interface. It is used for devices that have various uses and require different interfaces.<table><tr><td>OpenMachine, Interface name</td><td>The name of the interface that the user wants to switch to can be entered as an argument. By pressing Ctrl+Alt+Shift+F5 while the software is open, the name of the interface can be viewed in the Name section.</td></tr></table></td></tr>
</table>

<table>
<tr><td>48</td><td>OpenRecentFile</td><td>By running this function, the last file opened by the user will be reopened. This function can be used even if a portion of the program has been executed and the user has exited the software. The program will reopen from the exact point where the user exited and can continue the program execution.
In the General branch, there is a parameter called OpenRecentFile which, if enabled, will automatically execute this function when the program is opened.</td></tr>
<tr><td>49</td><td>OpenReport</td><td>Upon running this function, the following window opens, providing the user with information about the project.
(please refer to appendix 6 - Figure (23))
<table><tr><td>Total Dimention</td><td>The dimensions of the workpiece are displayed in each axis.</td></tr><tr><td>Work Dimention</td><td>The dimensions of the tool path are displayed in each axis.</td></tr><tr><td>ToolPath</td><td>The total length of the motion path and the tool path displayed in the ToolPath section is shown in general.</td></tr><tr><td>ToolChanger</td><td>If the interface has a ToolChanger capability, the tool numbers used in the program are displayed.</td></tr><tr><td>Total Price</td><td>After entering the price in the Unit Price section and selecting the calculation method in the Base of Price section, which is listed in millimeters, centimeters, meters, seconds, minutes, and hours respectively, the total price is calculated.</td></tr></table></td></tr>
<tr><td>50</td><td>OpenSetting</td><td>By executing this function, the Setting window opens.</td></tr>
<tr><td>51</td><td>OpenSimple-CAM</td><td>Reserve</td></tr>
<tr><td>52</td><td>OpenTab, Number</td><td>By running this function, the desired tab can be opened. The number of the tab can be found in the registry section of the Radonix software in the system.<table><tr><td>OpenTab, Number</td><td>The number written as an argument is the desired Tab number.</td></tr></table></td></tr>
<tr><td>53</td><td>OutPort, Number, Number, ...
OutPort, Number</td><td>By executing this function, the outputs can be turned on or off. It can be controlled manually using the Momenty key or the Toggle key. If the Momenty key is used, the operation is performed as soon as the function is executed and is stopped when the key is released. If the Toggle key is used, the operation is performed once by pressing the key and to stop the execution, the key must be pressed again.<table><tr><td>OutPort, Number</td><td>The number written as an argument is actually the output pin number, and by executing the function, the output is turned on.</td></tr><tr><td>OutPort, Number -</td><td>The number written as an argument is actually the output pin number, and by executing the function, the output is turned off.</td></tr><tr><td>OutPort, Number, Number, ...</td><td>In this function, multiple arguments can be entered simultaneously.
It is enough to separate the numbers with a comma between them, according to the number of outputs in the hardware.
If the numbers are positive, by executing the function, those outputs will be turned on.
If the numbers are negative, by executing the function, those outputs will be turned off.
Positive and negative numbers can be added simultaneously to this function, and it is not necessary for all numbers to be positive or negative.</td></tr></table></td></tr>
</table>

<table>
<tr><td>54</td><td>R-CornerJack,
Number</td><td>By executing this function, you can turn on or off the spindle.<table><tr><td>R-CornerJack,
0</td><td>By executing this function, the R-CornerJack can be turned on or off.</td></tr><tr><td>R-CornerJack, 1</td><td>By executing this function, the R-CornerJack is turned on.</td></tr><tr><td>R-CornerJack, 2</td><td>By selecting the Toggle key and executing this function, the R-CornerJack is turned on if it is off, and turned off if it is on.</td></tr></table></td></tr>
<tr><td>55</td><td>ReferenceAB-Centre
ReferenceAB-Centre, Axis name</td><td>By running this function, the average of two points A and B selected by the SavePoint function is calculated and the resulting number is considered as a reference.
The SavePoint function is explained in row 69 of the same table, please refer to it for more information.<table><tr><td>ReferenceABCentre</td><td>In this case, the reference is calculated for all axes according to the method described.</td></tr><tr><td>ReferenceABCentre,
Axis name</td><td>In this case, only the reference for the selected axis as the argument is calculated according to the method described.
The axis name can be any of the X, Y, Z, A, B, C axes.</td></tr></table></td></tr>
<tr><td>56</td><td>RemoteFunc-tionRun</td><td>Reserve</td></tr>
<tr><td>57</td><td>RemoteFunc-tionSelect</td><td>Reserve</td></tr>
<tr><td>58</td><td>ReplaceLocation</td><td>By executing this function, the ToolPath is moved to a selected point. For example, if a program is running and the user pauses it to move the axes to a new position, this function can be used to move the ToolPath to the new position. Then, by pressing the Run button, the program continues from the new point.</td></tr>
<tr><td>59</td><td>Rerun
Rerun, Axis name</td><td>By executing this function, the program will be repeated without creating any stop. Additionally, this function can be applied to an M-code and that M-code can be added to the G-code program, in which case the program will be repeated.<table><tr><td>Rerun</td><td>This function runs the program again</td></tr><tr><td>Rerun, Axis name</td><td>By executing this function, the last coordinate of the axis selected as the argument is considered as the Home point, and when the program is executed again, it will continue from this point. The axis name can be any of X, Y, Z, A, B, or C.</td></tr></table></td></tr>
<tr><td>60</td><td>Reset
Reset, Axis name</td><td>By executing this function, the tool is moved to the beginning of GCode and ToolPath is reset. Additionally, this function can be applied to an M code and added to the GCode program.<table><tr><td>Reset</td><td>Resets the program.</td></tr><tr><td>Reset, -1</td><td>In addition to resetting the program, it considers the axis coordinates as Reference.</td></tr><tr><td>Reset, Axis name</td><td>In addition to resetting the program, it considers the selected axis coordinate as the Home point for that axis.
The axis name can be any of X, Y, Z, A, B, C.</td></tr></table></td></tr>
</table>

<table>
<tr><td>61</td><td>RestoreInterface</td><td>By executing this function, the program's Backup is used. For example, if the program's settings have been mistakenly changed by the user, this function can be used to revert to the program's Backup.

Ctrl+Alt+Shift+R is actually the shortcut for this function.
By pressing Ctrl+Alt+Shift+F12, the interface settings can be reset. In other words, unused parameters are removed, the interface structure is updated, and the program's Backup is used for the settings.</td></tr>
<tr><td>62</td><td>Rotate</td><td>By executing this function, a window will open and by entering the angle in the Angle (Degree) box and pressing the Ok button, the design can be rotated around the Z-axis. (please refer to appendix 6 - Figure (24))</td></tr>
<tr><td>63</td><td>Run, Number</td><td>By running this function, the program is executed.
<table>
<tr><td>Run, 1</td><td>When G-Code is executed sequentially from the first line to the last line, it is referred to as Forward execution. In this mode, the program is executed in a forward direction.</td></tr>
<tr><td>Run, -1</td><td>When G-Code is executed sequentially from the last line to the first line, it is referred to as Backward execution. In this mode, the program is executed in a backward direction.</td></tr>
<tr><td>Run, 2</td><td>In this mode, the behavior changes between Forward execution and Pause mode. If this function is defined while the program is being executed, it will stop. If the machine is in Pause or Stop mode, pressing the Toggle button will start the program in Forward mode.</td></tr>
<tr><td>Run, -2</td><td>In this mode, the behavior changes between Backward execution and Pause mode. If this function is defined while the program is being executed, it will stop. If the machine is in Pause or Stop mode, pressing the Toggle button will start the program in Backward mode.</td></tr>
</table></td></tr>
<tr><td>64</td><td>RunApplication, File address
RunApplication, File address, B</td><td>By executing this function, the desired application is opened. In fact, the function takes the address of an application file and opens the application from that address.
<table>
<tr><td>RunApplication, File address</td><td>The function takes the address of an application file and opens the application from that address.</td></tr>
<tr><td>RunApplication, File address, B</td><td>It takes the file path of an application and opens the application accordingly. However, the control does not return to the Radonix software until the application is closed.</td></tr>
</table></td></tr>
<tr><td>65</td><td>RunTimer, Timer name</td><td>This function is used to call the desired timer. By executing the function, the timer starts to function according to the settings made in the Timer section of the Seeting window.
In the Timer section of the Seeting window, users can create any number of timers they need.
<table>
<tr><td>RunTimer, Timer name</td><td>By executing this function, the timer specified by the name entered as an argument is called.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>66</td><td>SJog, Axis name
SJog, Axis name, Axis name
SJog, Number</td><td>By executing the SJog function, it is possible to move the desired axis in the specified direction incrementally.<table>
<tr><td>SJog, Axis name</td><td>The axis name can be any of X, Y, Z, A, B, or C.</td></tr>
<tr><td>SJog, Axis name -</td><td>The negative sign is used to move in the negative direction of the axes.
The axis name can be any of X, Y, Z, A, B, or C.</td></tr>
<tr><td>SJog, Axis name, Axis name</td><td>Multiple arguments can be entered simultaneously in this function.
By executing this function, all selected axes move incrementally at the same time.
The axis name can be any of X, Y, Z, A, B, or C.</td></tr>
<tr><td>SJog, Axis name, Number</td><td>The selected axis moves in the specified direction by the numerical value entered as an argument in millimeters.
The axis name can be any of X, Y, Z, A, B, or C.</td></tr>
</table></td></tr>
<tr><td>67</td><td>SaveFile</td><td>By executing this function, the GCode file is opened and saved. For example, the user may have resized the file, changed the file's angle, or made other changes that require the modified file to be saved, and this function can be used for that purpose.</td></tr>
<tr><td>68</td><td>SavePoint</td><td><table>
<tr><td>SavePoint</td><td>By executing this function, the coordinates of the axes are stored in a list in order.</td></tr>
<tr><td>SavePoint, A</td><td>By executing this function, the coordinates of the axes are stored in a variable named A.</td></tr>
</table></td></tr>
<tr><td></td><td></td><td><table>
<tr><td>SavePoint, B</td><td>By executing this function, the coordinates of the axes are stored in a variable named B.</td></tr>
<tr><td>SavePoint, O</td><td>By executing this function, the coordinates of the axes are stored in a variable named O.</td></tr>
<tr><td>SavePoint, Number</td><td>By executing this function, the coordinates of the axes are stored in a list, and the order of storage in the list is determined by the number entered as an argument. For example, SavePoint,4 actually stores the coordinates of the axes in the fourth item of the list.

If the number entered as an argument is greater than the items that exist in the list, it stores the previous coordinates as empty. For example, if SavePoint,4 is entered, but no other coordinates have been saved in the list, it stores empty previous coordinates until it reaches the fourth item and then stores the coordinates in the fourth item.</td></tr>
</table></td></tr>
<tr><td>69</td><td>SaveSecureFile</td><td>This function is used in cases where the user wants to use a specific file and for some reasons doesn't want that file to be executable on a system other than their own. For example, by executing this function, it asks the user for the device serial number and encodes the file with that serial number. The encoded file can only be executed on the device with the same serial number. The extension of the encoded file is RCSF.
(please refer to appendix 6 - Figure (25))</td></tr>
<tr><td>70</td><td>Scale</td><td>By running this function, a window is opened and you can scale or resize the X, Y, and Z axes. By default, the scale of the axes is 100 percent. A negative value on any axis reverses that axis.
(please refer to appendix 6 - Figure (26))</td></tr>
</table>

<table>
<tr><td>71</td><td>Select, Number</td><td>By executing this function, multiple outputs are defined in a way that each output, when activated, will deactivate the others automatically. This function is related to a link named “SelectPin” which allows the user to define up to 32 Toggle keys, and by activating each key, the corresponding output defined by SelectPin will be turned on, while if another output has been defined by SelectPin, it will be turned off automatically. For example, we define three Toggle keys with links Select,1, Select,2, and Select,3, and then we define three outputs with links SelectPin,1, SelectPin,2, and SelectPin,3. By activating the key Select,1, the output 1 will turn on and outputs 2 and 3 will turn off automatically in order. This function is only applicable to the Toggle key.<table><tr><td>Select, Number</td><td>The number can be from zero to 32.</td></tr></table></td></tr>
<tr><td>72</td><td>SelectFeed, Number</td><td>Based on the given text, this function is used to detect the type of input selector connected to the input, whether it is binary or decimal, because different input selectors can have different types. If SelectBinaryFeed is set to True in the General tab, the connected selector is considered as binary, and if it’s set to False, the connected selector is considered as decimal. The function adds a parameter called SelectBinaryFeed to the General tab.<table><tr><td>SelectFeed, Number</td><td>If SelectBinaryFeed is enabled or True, the argument passed to the SelectFeed function should be a binary number. For example, by setting the SelectFeed links to 1, 2, 4, and 8, four input pins are defined, and by turning on each of these functions, their binary number is ORed together.

If SelectBinaryFeed is disabled or False, the argument passed to the SelectFeed function is a percentage. For example, by setting the SelectFeed links to 10, 20, 30, and 40, four input pins are defined, and by turning on each of these functions, the Feed percentage is determined.

The percentage passed as an argument to the SelectFeed function can be up to the maximum value set in the Feed parameter in the General branch.</td></tr></table></td></tr>
<tr><td>73</td><td>SelectReference, Refereance number</td><td><table><tr><td>SelectReference, Refereance number</td><td>By running this function, the reference whose number is passed as an argument will be selected as the active reference. For example, the reference number can be G54 to G59, and by selecting any of them, the coordinates of that reference will be the basis for the reference.

To enter the reference number, simply enter the number and there is no need to write the letter G. For example, SelectReference,54.</td></tr></table></td></tr>
<tr><td>74</td><td>SetG54</td><td>By executing this function, the current coordinates with their axis will be considered as G54 reference.</td></tr>
<tr><td>75</td><td>SetG55</td><td>By executing this function, the current coordinates with their axis will be considered as G55 reference.</td></tr>
<tr><td>76</td><td>SetG56</td><td>By executing this function, the current coordinates with their axis will be considered as G56 reference.</td></tr>
<tr><td>77</td><td>SetG57</td><td>By executing this function, the current coordinates with their axis will be considered as G57 reference.</td></tr>
<tr><td>78</td><td>SetG58</td><td>By executing this function, the current coordinates with their axis will be considered as G58 reference.</td></tr>
</table>

<table>
<tr><td>79</td><td>SetG59</td><td>By executing this function, the current coordinates with their axis will be considered as G59 reference.</td></tr>
<tr><td>80</td><td>SetOutPort, Number</td><td>By executing this function, it is possible to turn on or off outputs. This function is usually not used on a toggle key.<table><tr><td>SetOutPort, Number</td><td>By executing this function, the output is turned on. The argument passed to the function represents the output pin number.</td></tr><tr><td>SetOutPort, Number -</td><td>By running this function, the output is turned off. The number entered as the argument is actually the output pin number.</td></tr></table></td></tr>
<tr><td>81</td><td>SetPark</td><td>Executing this function sets the current coordinates of the axes as the Park coordinates.</td></tr>
<tr><td>82</td><td>SetReference
SetReference, Axis name
SetReference, Axis name, Axis name ...</td><td>By executing this function, the current coordinates with the axes in it are considered as the reference coordinates. If the machine has a Pointer and the Pointer is active, the user first specifies the point that they want the machine to be referenced to using the Pointer, and then the Pointer automatically calculates the offset and calculates the actual reference coordinates, considering them as the reference coordinates. For more information, please refer to rows 6 and 40 of this table.<table><tr><td>SetReference</td><td>By executing this function, the current coordinates with axes specified in the arguments are considered as the reference coordinates.</td></tr><tr><td>SetReference, Axis name</td><td>By executing this function, the current coordinates of the axis specified as the argument are considered as the reference coordinates.
The axis name can be any of X, Y, Z, A, B, C.</td></tr><tr><td>SetReference, Axis name, Axis name ...</td><td>Multiple arguments can be passed to this function simultaneously. By executing this function, the current coordinates of all the axes specified as the arguments are considered as the reference coordinates.
The axis name can be any of X, Y, Z, A, B, C.</td></tr></table></td></tr>
<tr><td>83</td><td>ShapeScanner</td><td>By executing this function, the machine starts to scan the workpiece to display an approximate shape of the workpiece.</td></tr>
<tr><td>84</td><td>SimpleCommand, Number</td><td>In SimpleCommand process, since the input links, output links, and function are closely related to each other, the complete process is explained in this section, and in order to be completely clear, the used link is mentioned during the process definition. Users can refer to the corresponding table and define the links based on the pin name or function. Also, in the diagram, which is drawn for more clarity, the order of operations and the activation of inputs, outputs, and functions are shown.
The links are represented by the following abbreviations:
((Input Link -- I), (Output Link -- O), (Function Link -- F))<table><tr><td>SimpleCommand</td><td>By executing this function, if the output is off, it will be turned on, and if it is on, it will be turned off.</td></tr><tr><td>SimpleCommand, 0</td><td>By executing this function, it turns off the SimpleCommand output.</td></tr><tr><td>SimpleCommand, 1</td><td>By executing this function, it turns on the SimpleCommand output.</td></tr><tr><td>SimpleCommand, 2</td><td>By executing this function, if the output is off, it will be turned on, and if it is on, it will be turned off for the SimpleCommand output,</td></tr></table>When the SimpleCommand command (function link) is executed, it first turns on the SimpleCommandPin (output link) and then waits for the SimpleCommandSensorPin (input link) input and if the SimpleCommandSensorPin (input link) is triggered or in other words becomes True, it turns off the SimpleCommandPin (output link) and the process ends.
(please refer to appendix 6 - Figure (27))</td></tr>
</table>

<table>
<tr><td>85</td><td>SimpleDelay, Number</td><td>When the SimpleDelay command (function link) is executed, depending on the entered argument, it turns on or off the SimpleDelayPin output link and then waits for the amount of time set in the SimpleDelay parameter in the Setting window under the General branch.<table><tr><td>SimpleDelay, 0</td><td>By executing this function, it turns off the SimpleDelayPin output (output link).</td></tr><tr><td>SimpleDelay, 1</td><td>By executing this function, it turns on the SimpleDelayPin output (output link).</td></tr></table></td></tr>
<tr><td>86</td><td>Simulation, Number</td><td>This function restarts the program and allows switching the program mode to simulation mode or exiting simulation mode and returning to normal mode. In simulation mode, the program can be executed without the need for a controller or hardware.<table><tr><td>Simulation, 0</td><td>It exits simulation mode and returns to normal mode.</td></tr><tr><td>Simulation, 1</td><td>It enters simulation mode.</td></tr></table></td></tr>
<tr><td>87</td><td>Spindle, Number</td><td>This function can be used to turn on or off a spindle.<table><tr><td>Spindle, 1</td><td>The SpindleCWPin output is turned on for rotating the spindle in a clockwise (CW) direction.</td></tr><tr><td>Spindle, -1</td><td>The SpindleCCWPin output is turned on for rotating the spindle in a counterclockwise (CCW) direction.</td></tr><tr><td>Spindle, 2</td><td>In this case, the SpindleCWPin output (rotating the spindle in a CW direction) is turned off if it was on, and turned on if it was off.</td></tr><tr><td>Spindle, -2</td><td>In this case, the SpindleCCWPin output (rotating the spindle in a CCW direction) is turned off if it was on, and turned on if it was off.</td></tr></table></td></tr>
<tr><td>88</td><td>SpindleCooler, Number</td><td>By running this function, the spindle cooling output can be turned on or off.<table><tr><td>SpindleCooler, 0</td><td>The output of the spindle cooling fan is turned off by running this function.</td></tr><tr><td>SpindleCooler, 1</td><td>The output of the spindle cooling fan is turned on by running this function.</td></tr><tr><td>SpindleCooler, 2</td><td>The output of the spindle cooling fan is turned off if it’s on, and turned on if it’s off by running this function.</td></tr></table></td></tr>
<tr><td>89</td><td>SpindleCover, Number</td><td>By running this function, the spindle cover output can be turned on or off.<table><tr><td>SpindleCover, 0</td><td>The output for spindle cover is turned off with the execution of this function.</td></tr><tr><td>SpindleCover, 1</td><td>The output for spindle cover is turned on with the execution of this function.</td></tr><tr><td>SpindleCover, 2</td><td>The output for spindle cover is turned off if it was on and turned on if it was off with the execution of this function.</td></tr></table></td></tr>
</table>

<table>
<tr><td>90</td><td>SpindleSpeed, Number</td><td>By executing this function, the spindle speed can be changed. In fact, the Scroll function that changes the spindle speed uses this function, and by using the Value link, the speed value of the program can be determined. This function can also be defined for a button or an input, and depending on the argument entered, the Step factor can be specified.

In the Router branch in the Setting window, there are several parameters that are related to this function (explained in detail in the Router branch description), including SpindleMaxSpeed, SpindleMinSpeed, SpindleSpeedStep, and so on.

<table><tr><td>SpindleSpeed, Number</td><td>The number specified as an argument is actually a multiplier of the value entered by the user in the SpindleSpeedStep parameter in the Router branch of the Setting window.</td></tr><tr><td>SpindleSpeed, Number -</td><td>The number specified as an argument is actually a negative multiplier of the value entered by the user in the SpindleSpeedStep parameter in the Router branch of the Setting window.</td></tr></table></td></tr>
<tr><td>91</td><td>Stop</td><td>By executing this function, the operation of the device is paused during program execution, jog movement, and homing operation.</td></tr>
<tr><td>92</td><td>StopHome, Axis name
StopHome, Axis name, Axis name</td><td>By executing this function, the operation of the axis or axes specified as an argument is paused during Homing operation.<table><tr><td>StopHome, Axis name</td><td>This function pauses the specified axis or axes during homing operation.
The axis name can be any of X, Y, Z, A, B, or C.
Multiple axes can be specified as arguments to the function.</td></tr><tr><td>StopHome, Axis name, Axis name</td><td>Multiple arguments can be entered simultaneously in this function. The axis name can be any of X, Y, Z, A, B, or C.</td></tr></table></td></tr>
<tr><td>93</td><td>StopJog, Axis name
StopJog, Axis name, Axis name ...</td><td>By executing this function, the movement of the axis or axes specified as arguments is paused during manual jogging.<table><tr><td>StopJog, Axis name</td><td>It pauses the movement of the axis or axes specified as the argument during manual jogging.
The axis name can be any of X, Y, Z, A, B, or C</td></tr><tr><td>StopJog, Axis name, Axis name ...</td><td>Multiple arguments can be entered simultaneously in this function.
The axis name can be any of X, Y, Z, A, B, or C.</td></tr></table></td></tr>
<tr><td>94</td><td>StopRun</td><td>This function pauses the machine operation during program execution.</td></tr>
<tr><td>95</td><td>StopRunReset</td><td>By executing this function, the machine is paused during program execution and is also returned to the first line of the program (G-Code). In case of running the program again, it will start the program from the beginning.</td></tr>
<tr><td>96</td><td>StopTimer, Timer name</td><td>By running this function, the timer is disabled. In the Timer branch of the Setting window, the user can create as many timers as needed.<table><tr><td>StopTimer, Timer name</td><td>The user enters the name of the timer they want to deactivate as an argument to run this function.</td></tr></table></td></tr>
</table>

<table>
<tr><td>97</td><td>SurfaceDetector, Axis name</td><td>This function sets the workpiece level. There is also an input link named SurfaceDetectorPin in the table of input links, which is used to define the input or sensor named SurfaceDetector, and by executing the SurfaceDetector function, the workpiece level is determined.<table><tr><td>SurfaceDetector, Axis name</td><td>In a router machine, Z-axis is usually defined but there is the ability to define other axes as well. The name of the axis can be any of the X, Y, Z, A, B, C axes.</td></tr></table></td></tr>
<tr><td>98</td><td>SurfaceScan</td><td>By running this function, the workpiece surface is scanned, and if there are any protrusions or depressions on the workpiece surface, the function applies changes to the G-Code to level the surface.</td></tr>
<tr><td>99</td><td>T-SelectTool
T-ToolNo</td><td>This function is used to display the current tool number of the machine, and also allows the tool to be changed by changing the tool number.</td></tr>
<tr><td>100</td><td>T-SetToolHeight</td><td>By executing this function, the tool height of the machine can be set.
This function is linked to an output called T-ToolHeightPin. Please refer to table 3 for more information.
This function is also linked to an input called T-ToolHeightSensorPin and an input called T-SetToolHeight. Please refer to table 2 for more information.</td></tr>
<tr><td>101</td><td>T-ToolHolder, Number</td><td>This function is used to open and close the tool holder inside the spindle. By defining the T-ToolHolder function and turning it on, the T-ToolHolderPin output, which is connected to the electrical valve inside the spindle, is turned on and holds or releases the tool inside the spindle according to the specified argument. This function is related to the T-ToolHolderPin output. Please refer to table 3 for more information. Note that if the spindle is on or the machine is running during the program, this function is disabled for safety reasons.<table><tr><td>T-ToolHolder,1</td><td>This function can be defined for a button or an input, so that when it is activated, it releases the tool and otherwise holds it.</td></tr></table></td></tr>
<tr><td>102</td><td>ToggleOutPort, Number
ToggleOutPort, Number, Number, ...</td><td>This function turns on the output pin or pins specified as arguments if they are off, and turns them off if they are on.<table><tr><td>ToggleOutPort, Number</td><td>The number entered as the argument is actually the output pin number. If a negative sign is used before the number, the on/off state of that pin is reversed.</td></tr><tr><td>ToggleOutPort, Number, Number, ...</td><td>Multiple arguments can be entered simultaneously in this function.</td></tr></table></td></tr>
<tr><td>103</td><td>TransformXY</td><td>By executing this function, a window opens that allows scaling and rotating a design around the Z axis. The scaling factor for each axis can be set using the X, Y, and Z fields, and the design can be rotated around the Z axis by entering an angle in degrees in the Angle field. In the default state, the scale of each axis is set to 100%. Negative values can be used for the scale to create symmetry.
(please refer to appendix 6 - Figure (28))</td></tr>
<tr><td>104</td><td>Vaccum, Number</td><td>By executing this function, you can turn on or off a vacuum output.<table><tr><td>Vaccum, 0</td><td>By running this function, the vacuum output will be turned off.</td></tr><tr><td>Vaccum, 1</td><td>By running this function, the vacuum output will be turned on.</td></tr><tr><td>Vaccum, 2</td><td>By running this function, the vacuum output will be turned on if it is off, and turned off if it is on.</td></tr></table></td></tr>
</table>

<table>
<tr>
<td>105</td>
<td>Zoom, Number</td>
<td>By executing this function, the Zoom operation is applied to the file displayed in ToolPath.
<table>
<tr><td>Zoom, -1</td><td>In this mode, each time the command is executed, the Zoom Out operation is performed on the file displayed in the ToolPath.</td></tr>
<tr><td>Zoom, 0</td><td>In this mode, each time the command is executed, the Fit operation is performed and the display size of the entire pattern is made the same size as the ToolPath dimensions.</td></tr>
<tr><td>Zoom, 1</td><td>In this mode, each time the command is executed, the Zoom In operation or magnification of the display of the pattern being executed is performed.</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><th></th><th>Name of the Element</th><th>Descriptions</th></tr>
<tr><td>1</td><td>AnalogFeed</td><td>By running AnalogFeed, it is possible to change the execution speed of the program or feed in an analog manner.</td></tr>
<tr><td>2</td><td>AnalogJog, Axis name</td><td>By running AnalogJog, the manual or jog motion command is given in an analog manner via the execution of the JoyStick command, and the speed of the manual motion can also be controlled. The AnalogJog command is effective in JoySticks because some of the JoyStick buttons are analog. Also, by changing the pressure applied to the button, the analog level in the JoyStick button is decreased or increased, and with the change in the analog level, the Jog speed is also changed.<table><tr><td>AnalogJog, Axis name</td><td>By changing the analog value, the Jog speed in the selected axis is changed. The axis name can be any of the X, Y, Z, A, B, C axes.</td></tr></table></td></tr>
<tr><td>3</td><td>AnalogJogFeed</td><td>It is used as a coefficient for AnalogJog and by changing the analog level, the manual or jog motion speed can be more accurately adjusted.</td></tr>
<tr><td>4</td><td>AnalogJogPedal</td><td>It is used as a coefficient (the coefficient in AnalogJogPedal is exponential) for AnalogJog, and by changing the analog level, the manual or jog motion speed can be more accurately adjusted. The AnalogJogPedal command is effective in JoySticks because some of the JoyStick buttons are analog. Also, by changing the pressure applied to the button, the analog level in the JoyStick button is decreased or increased, and with the change in the analog level, the Jog speed is also changed.</td></tr>
<tr><td>5</td><td>AnalogSpindle-Speed</td><td>By running AnalogSpindleSpeed, the spindle performance speed is changed in an analog manner.</td></tr>
</table>

Table(5)

<table>
<tr><th></th><th>Pin name</th><th>Descriptions</th></tr>
<tr><td>1</td><td>Analog, Number</td><td>This parameter is used to create a voltage from 0 to 10 volts on analog outputs and is also used to create a pulse with a density of 0 to 100 percent on PWM outputs. In fact, it is used to manually change the analog.
<table>
<tr><td>Analog,1</td><td>The argument entered represents the analog number embedded on the Radonix controller hardware. For example, by selecting the number 1 as the argument, changes will be applied to analog number 1.</td></tr>
<tr><td>Analog,1</td><td>The number entered as an argument is actually the identifier of the analog that is embedded in the Radonix controller hardware. For example, by selecting the number 2 as an argument, the changes will be applied to analog number 2.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button ☐</td><td>Toggle ☐</td><td>Check Box ☐</td><td>Label ■</td></tr>
<tr><td>Value ■</td><td>Signal ☐</td><td>Vertical Scroll ■</td><td></td></tr>
</table></td></tr>
<tr><td>2</td><td>BinaryOutPort</td><td>This parameter is used for binary control of spindle speed. For example, an inverter connected to the spindle can be controlled, and depending on the inputs provided by the inverter, the speed can be adjusted.

Note that this parameter is related to the BinaryOutPortPin output and to use this parameter, the corresponding output should also be defined according to Table 3.
<table>
<tr><td rowspan="2">Element</td><td>Button ☐</td><td>Toggle ☐</td><td>Check Box ☐</td><td>Label ☐</td></tr>
<tr><td>Value ☐</td><td>Signal ☐</td><td>Vertical Scroll ■</td><td></td></tr>
</table></td></tr>
<tr><td>3</td><td>BlockNumber</td><td>This parameter can be used to view the G-Code line number and enter the desired line number.
<table>
<tr><td rowspan="2">Element</td><td>Button ■</td><td>Toggle ☐</td><td>Check Box ☐</td><td>Label ■</td></tr>
<tr><td>Value ■</td><td>Signal ☐</td><td>Vertical Scroll ☐</td><td></td></tr>
</table>
In Value, the line number value can be viewed and entered. To do this, first the G-Code line number is entered in Value, and by pressing Enter, it is transferred to the entered line number in the G-Code.
In Label, the line number value can be viewed.
In Button, by executing this parameter, a window opens where the line number can be entered, and by pressing the Ok button, it is transferred to the desired line number entered in the G-Code.
(please refer to appendix 6 - Figure (29))</td></tr>
</table>

<table>
<tr>
<td>4</td>
<td>ConveyorDisplacement, Axis name</td>
<td>
Using this parameter, the amount of conveyor movement can be determined in millimeters. By entering the amount of conveyor movement in millimeters and executing the conveyor command, the conveyor will move by the specified amount and then automatically turn off.

• Note that this parameter is related to the ConveyorSensorPin input. For more information, please refer to Table 2.
• Note that this parameter is related to the ConveyorPin output and to use it, the corresponding output must be defined according to Table 3.
• Note that this parameter is related to the Conveyor function and to use it, the corresponding output must be defined according to Table 4.
• Note that this parameter is related to the ConveyorVelocity parameter. For more information, please refer to Row 5 in the same table.
<table>
<tr><td>ConveyorDisplacement, Axis name</td><td>The name of the axis can be any of X, Y, or Z axes.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button
☐</td><td>Toggle
☐</td><td>Check Box
☐</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>
• In Value, the amount of conveyor movement can be viewed and entered.
• In Label, the amount of conveyor movement can be viewed.
</td>
</tr>
<tr>
<td>5</td>
<td>ConveyorVelocity, Axis name</td>
<td>
• This parameter is used to set the speed of conveyor movement.
• Note that this parameter is related to the input ConveyorSensorPin. For more information, please refer to Table 2.
• Note that this parameter is related to the output ConveyorPin. For more information, please refer to Table 3.
• Note that this parameter is related to the function Conveyor. For more information, please refer to Table 4.
• Note that this parameter is related to the ConveyorDisplacement parameter. For more information, please refer to row 4 in this table.
<table>
<tr><td>ConveyorVelocity, Axis name</td><td>The name of the axis can be any of the X, Y, or Z axes.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button
☐</td><td>Toggle
☐</td><td>Check Box
☐</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>
• In Value, the amount of conveyor movement can be viewed and entered.
• In Label, the amount of conveyor movement can be viewed.
</td>
</tr>
<tr>
<td>6</td>
<td>DisplaceReference, Axis name</td>
<td>
This parameter is used to create partial changes in the Reference. For example, during program execution, if the user realizes that a design is skewed at a point and has an error, they can immediately compensate for it by changing the Reference using this parameter.
Note that this parameter is related to the DisplaceReference function. For more information, please refer to Table 4.
Please note that if this parameter is used, the RealTimeReferenceDisplace parameter is added to the General branch in the Settings window and the ReferenceDisplace parameter is added to the General and Axis sub-branches. If the RealTimeReferenceDisplace parameter is enabled or, in other words, set to True, the DisplaceReference parameter can be used during program execution. If it is False, program execution must be stopped first and then the DisplaceReference parameter can be used. Also, the Reference value is shifted by the amount stored in the ReferenceDisplace parameter.
</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<tr><td>DisplaceReference, Axis name</td><td>The Reference point of the workpiece on the specified axis as the argument is displaced by the entered value. The axis name can be any of X, Y, Z, A, B, or C.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button
■</td><td>Toggle
□</td><td>Check Box
□</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
□</td><td>Vertical Scroll
□</td><td></td></tr>
</table>

In Value, the numerical value of the Reference point in the specified axis can be viewed and entered.
In Lable, the numerical value of the Reference point in the specified axis can be viewed.
In Button, by executing this parameter, the Reference point in the specified axis is displaced by the value entered in the ReferenceDisplace parameter under the axes (X, Y, Z, A, B, C) in the General branch of the Setting window.

</td></tr>
<tr><td>7</td><td>Feed</td><td>
This parameter allows the user to change the speed during program execution by applying a factor to the execution speed. Note that this parameter is related to the Feed function, for more information please refer to table 3.
<table>
<tr><td rowspan="2">Element</td><td>Button
■</td><td>Toggle
■</td><td>Check Box
□</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
□</td><td>Vertical Scroll
■</td><td></td></tr>
</table>

In VerticalScroll, the program execution speed factor can be determined and the Feed function is used. For more information, please refer to Table 3.
In Value, the factor of the program execution speed can be viewed and set.
In Lable, the factor of the program execution speed can be viewed.
In Button, by running the Feed function and using the argument entered, the program execution speed factor can be specified. For more information, please refer to Table 3.

</td></tr>
<tr><td>8</td><td>FitDimension,
Axis name</td><td>
This parameter is used to change the dimensions of the opened file. For example, if the design of the opened file is 150 millimeters along the X-axis and the user wants the design to be 200 millimeters along the X-axis, they use this parameter and by entering the number 200 in the Value field, the design is enlarged to 200 millimeters along the X-axis. In fact, this function is similar to the Scale function, but with the difference that it is enough to enter the final value in each axis and the design changes to that size. Also, depending on the axis chosen as the argument, changes can be made in different axes.
The user can scale the design up or down.
By entering a negative value, in addition to changing the dimensions of that axis, symmetry can be created.
If no value is entered, the file will be opened without any changes.
<table>
<tr><td>FitDimension, Axis name</td><td>The axis name can be any of the X, Y, Z axes.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button
□</td><td>Toggle
□</td><td>Check Box
□</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
□</td><td>Vertical Scroll
□</td><td></td></tr>
</table>

In Value, you can view and enter the desired dimensions. Also, if no value is entered, the file will be opened without any changes.
In Label, you can see the desired dimensions.

</td></tr>
</table>

<table>
<tr>
<td>9</td>
<td>Floor</td>
<td>
This parameter is used to set a coordinate on the Z-axis such that the tip of the tool will not be allowed to move below that coordinate. Essentially, movement on the Z-axis is limited up to the coordinate specified in the Value field, and this coordinate is also applied to the relevant G-Code. If the Reference point or any point in the G-Code is below the level set in the Floor parameter, all those points will be brought up to the Floor level. Note that any changes to the Floor parameter require reopening the G-Code file for the value to be applied.
Please note that this parameter is related to the Floor function, and for more information, please refer to table number 3.
<table>
<tr><td rowspan="2">Element</td><td>Button
■</td><td>Toggle
☐</td><td>Check Box
☐</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>

In Value, the desired coordinates can be viewed and entered.
In Label, the desired coordinates can be viewed.
In Button, by running the Floor function and using the entered argument, it can be used. For more information, please refer to Table 3.

</td>
</tr>
<tr>
<td>10</td>
<td>SetG54, Axis name</td>
<td>
This parameter is used to set a numerical value as a reference for G54. Additionally, the numerical value stored in the G54 reference can be viewed.
<table>
<tr><td>SetG54, Axis name</td><td>The numerical value entered is saved as the reference in G54. The axis name can be any of X, Y, Z, A, B, or C.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button
■</td><td>Toggle
☐</td><td>Check Box
☐</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>

In Value, the numerical value saved as the reference in G54 can be viewed and entered.
In Label, the numerical value saved as the reference in G54 can be viewed.

</td>
</tr>
<tr>
<td>11</td>
<td>SetG55, Axis name</td>
<td>
This parameter is used to set a numerical value as a reference in G55. It is also possible to view the numerical value that is stored as the reference in G55.
<table>
<tr><td>SetG55, Axis name</td><td>The numerical value entered is stored as a reference in G55. The axis name can be any of X, Y, Z, A, B, or C.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button
■</td><td>Toggle
☐</td><td>Check Box
☐</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>

In Value, you can view and enter the numerical value to be used as the reference for G55.
In Label, you can view the numerical value that has been stored as the reference for G55.

</td>
</tr>
</table>

<table>
<tr>
<td>12</td>
<td>SetG56, Axis name</td>
<td>
This parameter is used to set a numerical value as a reference in G56. It is also possible to view the numerical value that is stored as the reference in G56.
<table>
<tr><td>SetG56, Axis name</td><td>The numerical value entered is stored as a reference in G56. The axis name can be any of X, Y, Z, A, B, or C.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button ■</td><td>Toggle □</td><td>Check Box □</td><td>Label ■</td></tr>
<tr><td>Value ■</td><td>Signal □</td><td>Vertical Scroll □</td><td></td></tr>
</table>

In Value, you can view and enter the numerical value to be used as the reference for G56.
In Label, you can view the numerical value that has been stored as the reference for G56.

</td>
</tr>
<tr>
<td>13</td>
<td>SetG57, Axis name</td>
<td>
This parameter is used to set a numerical value as a reference in G57. It is also possible to view the numerical value that is stored as the reference in G57.
<table>
<tr><td>SetG57, Axis name</td><td>The numerical value entered is stored as a reference in G57. The axis name can be any of X, Y, Z, A, B, or C.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button ■</td><td>Toggle □</td><td>Check Box □</td><td>Label ■</td></tr>
<tr><td>Value ■</td><td>Signal □</td><td>Vertical Scroll □</td><td></td></tr>
</table>

In Value, you can view and enter the numerical value to be used as the reference for G57.
In Label, you can view the numerical value that has been stored as the reference for G57.

</td>
</tr>
<tr>
<td>14</td>
<td>SetG58, Axis name</td>
<td>
This parameter is used to set a numerical value as a reference in G58. It is also possible to view the numerical value that is stored as the reference in G58.
<table>
<tr><td>SetG58, Axis name</td><td>The numerical value entered is stored as a reference in G58. The axis name can be any of X, Y, Z, A, B, or C.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button ■</td><td>Toggle □</td><td>Check Box □</td><td>Label ■</td></tr>
<tr><td>Value ■</td><td>Signal □</td><td>Vertical Scroll □</td><td></td></tr>
</table>

In Value, you can view and enter the numerical value to be used as the reference for G58.
In Label, you can view the numerical value that has been stored as the reference for G58.

</td>
</tr>
</table>

<table>
<tr>
<td>15</td>
<td>SetG59, Axis name</td>
<td>
This parameter is used to set a numerical value as a reference in G59. It is also possible to view the numerical value that is stored as the reference in G59.
<table>
<tr><td>SetG59, Axis name</td><td>The numerical value entered is stored as a reference in G59. The axis name can be any of X, Y, Z, A, B, or C.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button ■</td><td>Toggle □</td><td>Check Box □</td><td>Label ■</td></tr>
<tr><td>Value ■</td><td>Signal □</td><td>Vertical Scroll □</td><td></td></tr>
</table>

In Value, you can view and enter the numerical value to be used as the reference for G59.
In Label, you can view the numerical value that has been stored as the reference for G59.

</td>
</tr>
<tr>
<td>16</td>
<td>HandwheelA, Axis name</td>
<td>
HandwheelA is short for Handwheel Axis, and is used to determine the axis in which the handwheel is active. Note that this parameter can be used in inputs.
<table>
<tr><td>HandwheelA, Axis name</td><td>The axis name can be any of the X, Y, or Z axes.</td></tr>
<tr><td>HandwheelA, -1</td><td>You can change the Feed rate factor.</td></tr>
<tr><td>HandwheelA, -2</td><td>You can change the Spindle speed.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button □</td><td>Toggle ■</td><td>Check Box □</td><td>Label □</td></tr>
<tr><td>Value □</td><td>Signal □</td><td>Vertical Scroll □</td><td></td></tr>
</table>

In Toggle, by executing it, the handwheel of the desired axis can be activated.

</td>
</tr>
<tr>
<td>17</td>
<td>HandwheelStep, Axis name</td>
<td>
This parameter is for setting the step value of the handwheel for each axis, and with each rotation of the handwheel, the axis will move by the product of the handwheel coefficient and the step value.
<table>
<tr><td>HandwheelStep, Axis name</td><td>The name of the axis can be any of X, Y, or Z.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button □</td><td>Toggle □</td><td>Check Box □</td><td>Label ■</td></tr>
<tr><td>Value ■</td><td>Signal □</td><td>Vertical Scroll □</td><td></td></tr>
</table>

In Value, you can view and enter the value of each Step.
In Label, you can view the value of each Step.

</td>
</tr>
<tr>
<td>18</td>
<td>HandwheelX, Number</td>
<td>
This parameter is for setting the handwheel ratio.
Typically, the handwheel ratio is 1, 10, 100, or 1000, but it can be set to any value.
Note that this parameter can be used in inputs.
<table>
<tr><td>HandwheelX, Number</td><td>The number is actually the handwheel multiplier.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button □</td><td>Toggle ■</td><td>Check Box □</td><td>Label □</td></tr>
<tr><td>Value □</td><td>Signal □</td><td>Vertical Scroll □</td><td></td></tr>
</table>

In Toggle, by executing it, you can select the desired handwheel factor.

</td>
</tr>
</table>

<table>
<tr><td>19</td><td>JogFeed, Number</td><td>This parameter is used to set the speed factor for manual jog movements, also known as JogFeed. Note that this parameter is related to the JogFeed function, for more information please refer to table 3.
<table>
<tr><td rowspan="2">Element</td><td>Button
■</td><td>Toggle
■</td><td>Check Box
□</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
□</td><td>Vertical Scroll
■</td><td></td></tr>
</table>
In VerticalScroll, the speed of manual movement or Jog can be changed.
In Value, the value of the speed of manual movement or Jog can be entered.
In Lable, the value of the speed of manual movement or Jog can be viewed.
In Button, by executing the JogFeed function and using the entered argument, it can be used. For more information, please refer to Table 3.
In Toggle, by executing the JogFeed function and using the entered argument, it can be used. For more information, please refer to Table 3.</td></tr>
<tr><td>20</td><td>JogLocation, Axis name</td><td>This parameter is used to set the destination coordinates for manual movement or Jogging. For example, if the value of JogLocation is entered as 100 millimeters, by executing the AJog function with the specified argument, the axis will be moved to that coordinate.
Note that this parameter is related to the AJog function. For more information, please refer to table 3.
<table>
<tr><td>JogLocation, Axis name</td><td>The specified axis as an argument is actually the axis to be moved manually to the specified coordinates. The axis name can be any of X, Y, Z, A, B, or C.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button
□</td><td>Toggle
□</td><td>Check Box
□</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
□</td><td>Vertical Scroll
□</td><td></td></tr>
</table>
In Value, the coordinate value for manual or Jog movement can be viewed and entered.
In Label, the coordinate value for manual or Jog movement can be viewed.</td></tr>
<tr><td>21</td><td>JogMode, Number</td><td>This parameter is used to determine the status of Jog command or manual movement. Note that this parameter is related to JogMode function. For more information, please refer to table number 3.
<table>
<tr><td rowspan="2">Element</td><td>Button
□</td><td>Toggle
□</td><td>Check Box
□</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
□</td><td>Vertical Scroll
□</td><td></td></tr>
</table>
In Value, the Jog command status can be set.
If the number 0 is entered, Jog is continuous.
If the number 1 is entered, Jog is incremental
In Lable, the Jog command status can be viewed.
In Toggle, by running the JogMode function and using the entered argument, the Jog command status can be set. For more information, please refer to table 3.</td></tr>
<tr><td>22</td><td>JogStep, Axis name</td><td>This parameter is used to set the step value for Incremental Jog motion.
<table>
<tr><td>JogStep, Axis name</td><td>The axis name can be any of X, Y, or Z.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<tr><td rowspan="2">Element</td><td>Button
☐</td><td>Toggle
☐</td><td>Check Box
☐</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>
In Value, the value of the steps for manual jogging can be viewed and entered.
In Lable, the value of the steps for manual jogging can be viewed.
</td></tr>
<tr><td>23</td><td>JogVelocity</td><td>
This parameter is used to set the speed of motion during manual movement or Jog in both Incremental and Continuous modes.
<table>
<tr><td>JogVelocity, Axis name</td><td>The axis name can be any of X, Y, or Z.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button
☐</td><td>Toggle
☐</td><td>Check Box
☐</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>
In Value, you can view and enter the speed of movement during manual movement or Jog.
In Label, you can view the speed of movement during manual movement or Jog.
</td></tr>
<tr><td>24</td><td>NextFile, Number</td><td>
The GCode branch includes a parameter called MultiFileEnable, which when activated, adds the ability to open multiple files simultaneously to the software. In this case, a list of selected files is made available to the user, which can be rearranged in this list. In fact, this parameter is used to move and control the execution of each of the selected files.
<table>
<tr><td>NextFile</td><td>The parameter displays the currently running file and the number of open files.</td></tr>
<tr><td>NextFile, 1</td><td>In this mode, files are selected in the list from the first file to the last file.</td></tr>
<tr><td>NextFile, -1</td><td>In this mode, files are selected in the list from the last file to the first file.</td></tr>
<tr><td>NextFile, -2</td><td>In this mode, the list is reset, which means that files are selected from the first file again for any file.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button
■</td><td>Toggle
☐</td><td>Check Box
☐</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>
In Button, by executing it and using the input argument, selected files can be viewed.
</td></tr>
</table>

<table>
<tr>
<td>25</td>
<td>Park, Axis name</td>
<td>
This parameter is used to specify the Park coordinates and also allows the user to view the coordinates set for Park according to the specified axis as an argument.
<table>
<tr><td>Park, Axis name</td><td>The axis name can be any of X, Y, or Z.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button ☐</td><td>Toggle ☐</td><td>Check Box ☐</td><td>Label ■</td></tr>
<tr><td>Value ■</td><td>Signal ☐</td><td>Vertical Scroll ☐</td><td></td></tr>
</table>
In Value, you can view and set the Park coordinates.
In Label, you can view the Park coordinates.
</td>
</tr>
<tr>
<td>26</td>
<td>Reference, Axis name</td>
<td>
This parameter is used to activate the Reference coordinates, and also the coordinates set for the Reference can be viewed based on the specified axis as an argument.
<table>
<tr><td>Reference, Axis name</td><td>The axis name can be any of X, Y, or Z.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button ☐</td><td>Toggle ☐</td><td>Check Box ☐</td><td>Label ■</td></tr>
<tr><td>Value ■</td><td>Signal ☐</td><td>Vertical Scroll ☐</td><td></td></tr>
</table>
In Value, the active Reference coordinates can be viewed and set.
In Label, the active Reference coordinates can be viewed.
</td>
</tr>
<tr>
<td>27</td>
<td>SafeZ</td>
<td>
This parameter is used to set the SafeZ coordinate, and you can also view the SafeZ coordinate. In fact, when moving the Z-axis to different positions, this parameter is used to avoid traveling the full course on the Z-axis. In this case, a lower coordinate can be defined for SafeZ, which will be moved to that coordinate during movement, resulting in less time spent traveling between positions.
<table>
<tr><td rowspan="2">Element</td><td>Button ☐</td><td>Toggle ☐</td><td>Check Box ☐</td><td>Label ■</td></tr>
<tr><td>Value ■</td><td>Signal ☐</td><td>Vertical Scroll ☐</td><td></td></tr>
</table>
In Value, you can view and set the SafeZ coordinates.
In Label, you can view the SafeZ coordinates.
</td>
</tr>
<tr>
<td>28</td>
<td>SelectFeed, Number</td>
<td>
This parameter is used to connect binary and decimal keys for setting the speed factor. If this parameter is used, a parameter called SelectBinaryFeed is added to the General branch. If SelectBinaryFeed is active or True, the defined key is considered binary and if SelectBinaryFeed is inactive or False, the defined key is considered decimal.
<table>
<tr><td>Select-Feed, Number</td><td>If SelectBinaryFeed is enabled, or in other words, True, the number entered as an argument in the SelectFeed parameter is actually the desired binary number. For example, you can define 4 key numbers with links SelectFeed,1, SelectFeed,2, SelectFeed,4, and SelectFeed,8, and by turning on each of these keys, their binary number will be ORed together.
The numbers entered as arguments in the case where SelectBinaryFeed is active, or in other words, True, are between 0 and 31.</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td>If SelectBinaryFeed is disabled, or in other words, False, the number entered as an argument in the SelectFeed parameter is actually the percentage value. For example, you can define 4 key numbers with links SelectFeed,10, SelectFeed,20, SelectFeed,30, and SelectFeed,40, and by turning on each of these keys, the feed percentage will be determined.

The percentage entered as an argument in the SelectFeed parameter is between 0 and 200.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button
☐</td><td>Toggle
■</td><td>Check Box
☐</td><td>Label
☐</td></tr>
<tr><td>Value
☐</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>
In Toggle, by executing the SelectFeed parameter and using the entered argument, the feed rate coefficient can be determined.
</td></tr>
<tr><td>29</td><td>SelectReference
SelectReference, Reference Number</td><td>
This parameter is used to activate the Reference and it’s also possible to view the active Reference.
Note that this parameter is similar to the SelectReference function.
<table>
<tr><td>SelectReference</td><td>is used in Value and Lable.</td></tr>
<tr><td>SelectReference, Reference Number</td><td>This parameter is used in Button and Toggle, and by executing this parameter, the number entered as an argument represents the reference number. For example, the reference number can be G54 to G59, and by selecting each one, the coordinates of that reference are set as the reference base. To enter the reference number, simply enter the number and there is no need to write the letter G. For example, SelectReference,54.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button
■</td><td>Toggle
■</td><td>Check Box
☐</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>
In Value, the active Reference can be viewed and set.
In Lable, the active Reference can be viewed.
In Button, by executing this parameter and using the entered argument, the active Reference can be set.
In Toggle, by executing this parameter and using the entered argument, the active Reference can be set.
</td></tr>
<tr><td>30</td><td>ShowGroup / ShowGroup, Group number</td><td>This parameter is used to determine the group number and the used group number can also be viewed. In other words, there is a capability on the edit page of the interface to group elements. For example, three keys can be placed in group 1 and four other keys in group 2. Now, using this parameter, the contents of the called group can be viewed and used, in fact, it is a kind of Tab.</td></tr>
</table>

		ShowGroup: In Value and Label, this parameter is used to display the active group number. ShowGroup, Group number: In Button and Toggle, this parameter is used to specify the group number with the argument provided in the ShowGroup parameter. This allows you to access and activate the contents of a specific group, similar to a tab in a user interface. Element: Button ■, Toggle ■, Check Box □, Label ■; Value ■, Signal □, Vertical Scroll □ In Value, the used group number can be viewed and set. In Label, the used group number can be viewed. In Button, by executing this parameter and using the entered argument, the desired group number can be activated. In Toggle, by executing this parameter and using the entered argument, the desired group number can be activated.
31	SimpleCommand, Number	In SimpleCommand, since the input links, output links, and function are all closely related, the entire process is fully explained in this section. To ensure clarity, the links used are mentioned when defining the process, and the user can refer to the relevant table based on the pin name or function to define the links. Additionally, a diagram is provided that depicts the order of operations and the activation of inputs, outputs, and functions for greater clarity. The links are represented by the following shorthand symbols: ((Input Link -- I), (Output Link -- O), (Function Link -- F)) SimpleCommand: This parameter is used in Lable and Toggle. SimpleCommand,0: By executing this parameter, the output of SimpleCommandPin will turn off. It is used in Lable and Button. SimpleCommand,1: By executing this parameter, the SimpleCommandPin output will turn on. It is used in Lable and Button SimpleCommand,2: This parameter is used in Lable and Button. By executing this parameter, if SimpleCommandPin output is off, it will turn on and if it is already on, it will turn off. When the SimpleCommand (parameter link) command is executed, first the SimpleCommandPin (output link) is turned on, and then it waits for the input of SimpleCommandSensorPin (input link) and if the SimpleCommandSensorPin (input link) is triggered or, in other words, becomes True, it turns off the SimpleCommandPin (output link) and the process ends. **(please refer to appendix 6 - Figure (30))** Element: Button ■, Toggle ■, Check Box □, Label ■; Value □, Signal □, Vertical Scroll □ In Lable, you can see the on/off state of the SimpleCommand. In Button, by executing this parameter with the specified argument, the SimpleCommand process is executed. In Toggle, by executing this parameter, the SimpleCommand process is executed.

<table>
<tr><td>32</td><td>SimpleCom-
mandManual</td><td>This parameter is used to enable or disable the SimpleCommand process. If Simple-CommandManual is set to True, the SimpleCommand process is completely disabled, and if it’s set to False, the SimpleCommand process is enabled.
<table>
<tr><td rowspan="2">Element</td><td>Button
☐</td><td>Toggle
■</td><td>Check Box
☐</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>
In Lable, you can see if SimpleCommand is enabled or disabled.
In Toggle, by executing this parameter, you can enable or disable SimpleCommand-Manual.</td></tr>
<tr><td>33</td><td>Smoothness</td><td>This parameter is used to set the motion smoothness and also to view the set motion smoothness.
There is a parameter in ToolPath in the Settings window that is related to motion smoothness.
Motion smoothness can be defined from 0 to 250 milliseconds. 250 has the highest smoothness value.
Note that if the machine acceleration is above 1000, it is better to reduce the motion smoothness, and if the machine acceleration is low, the smoothness can be increased up to 250 milliseconds.
High acceleration and smoothness can have an undesirable effect on the coordination between axes.
For a router machine, a value of about 50 milliseconds is recommended.
<table>
<tr><td rowspan="2">Element</td><td>Button
☐</td><td>Toggle
☐</td><td>Check Box
☐</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>
In Value, you can both view and set the value of Smoothness.
In Lable, you can view the value of Smoothness.</td></tr>
<tr><td>34</td><td>SpindleAutoOff</td><td>This parameter is used to turn off the Spindle in G-Codes that do not have the M code for turning off the Spindle.
<table>
<tr><td rowspan="2">Element</td><td>Button
■</td><td>Toggle
■</td><td>Check Box
☐</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>
In Toggle, this parameter can be used to turn off the spindle.
In Lable, you can see the status of the spindle being off.
In Button, by executing this parameter, you can turn off the spindle.</td></tr>
<tr><td>35</td><td>SpindleAutoOn</td><td>This parameter is used to turn on the Spindle in G-Codes where there is no M-code to turn it on.
<table>
<tr><td rowspan="2">Element</td><td>Button
■</td><td>Toggle
■</td><td>Check Box
☐</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>
In Toggle, by executing this parameter, you can turn on the spindle.
In Lable, you can see if the spindle is on or not.
In Button, by executing this parameter, you can turn on the spindle.</td></tr>
</table>

<table>
<tr>
<td>36</td>
<td>SpindleSpeed,
Number</td>
<td>This parameter is used to change the spindle speed and can also display the spindle speed. In the Router branch of the Setting window, there are several parameters related to this parameter (which are fully explained in the section on the Router branch). These include SpindleMaxSpeed, SpindleMinSpeed, SpindleSpeedStep, and more. Note that this parameter is related to the SpindleSpeed function. For more information, please refer to Table 3.
<table>
<tr><td rowspan="2">Element</td><td>Button
■</td><td>Toggle
□</td><td>Check Box
□</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
□</td><td>Vertical Scroll
■</td><td></td></tr>
</table>
In Value, the spindle speed can be viewed and set.
In Lable, the spindle speed can be viewed.
In VerticalScroll, the spindle speed can be changed and the SpindleSpeed function (function link) can be used with the argument entered. For more information, please refer to Table 3.
In Button, the spindle speed can be changed and the SpindleSpeed function (function link) can be used with the argument entered. For more information, please refer to Table 3.</td>
</tr>
<tr>
<td>37</td>
<td>T-SelectTool
T-ToolNo
T-SelectTool,
Tool number
T-ToolNo, Tool
number</td>
<td>This parameter is used to select the tool in machines that have a tool changing capability (Toolchange).
Both T-SelectTool and T-ToolNo parameters are similar.
<table>
<tr><td>T-SelectTool
T-ToolNo</td><td>This parameter is used in Value and Lable.</td></tr>
<tr><td>T-SelectTool, Tool number
T-ToolNo, Tool number</td><td>It is used in Button and Toggle, and by executing this parameter, the number entered as an argument is actually the tool number.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button
■</td><td>Toggle
■</td><td>Check Box
□</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
□</td><td>Vertical Scroll
□</td><td></td></tr>
</table>
In Value, you can view and set the tool number.
In Lable, you can view the tool number.
In Toggle, by executing this parameter, you can select the desired tool number.
In Button, by executing this parameter, you can select the desired tool number.</td>
</tr>
<tr>
<td>38</td>
<td>T-ToolHeight</td>
<td>This parameter is used to set and display the tool height value in devices that have the ability to change tools (Toolchange).
<table>
<tr><td rowspan="2">Element</td><td>Button
□</td><td>Toggle
□</td><td>Check Box
□</td><td>Label
■</td></tr>
<tr><td>Value
■</td><td>Signal
□</td><td>Vertical Scroll
□</td><td></td></tr>
</table>
In Value, you can view and set the tool height.
In Lable, you can view the tool height.</td>
</tr>
<tr>
<td>39</td>
<td>ToolPathReport,
Number</td>
<td>After loading a file and displaying it in ToolPath, in the top left corner of ToolPath, the address of the loaded file is displayed, and the Work Size or the workpiece size in different axes, and the Work Length or the total length of the work can be viewed. To display these items in Value or Lable, this parameter is used.</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td>
<table>
<tr><td>ToolPathReport,0</td><td>The file address that has been loaded is displayed.</td></tr>
<tr><td>ToolPathReport,1</td><td>Work Size or dimensions in different axes are displayed</td></tr>
<tr><td>ToolPathReport,2</td><td>Work Length or the total length of the work (cutting and non-cutting lines) is displayed</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button
☐</td><td>Toggle
☐</td><td>Check Box
☐</td><td>Label
■</td></tr>
<tr><td>Value
☐</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>
In Lable, you can view a report of the file
</td>
</tr>
<tr>
<td>40</td>
<td>Velocity
Velocity, Axis name
Velocity, Axis name, Axis name ...</td>
<td>
This parameter is used to display the program execution speed.
<table>
<tr><td>Velocity</td><td>The speed of all axes is displayed.</td></tr>
<tr><td>Velocity, Axis name</td><td>The speed of the axis selected as an argument is displayed. The name of the axis can be any of X, Y, Z, A, B, C.</td></tr>
<tr><td>Velocity, Axis name, Axis name ...</td><td>The parameter is used to display the program execution speed on the axes that have been selected as arguments. The axis name can be any of the X, Y, Z, A, B, or C axes.</td></tr>
</table>
<table>
<tr><td rowspan="2">Element</td><td>Button
☐</td><td>Toggle
☐</td><td>Check Box
☐</td><td>Label
■</td></tr>
<tr><td>Value
☐</td><td>Signal
☐</td><td>Vertical Scroll
☐</td><td></td></tr>
</table>
In Lable, the program execution speed can be viewed.
</td>
</tr>
</table>

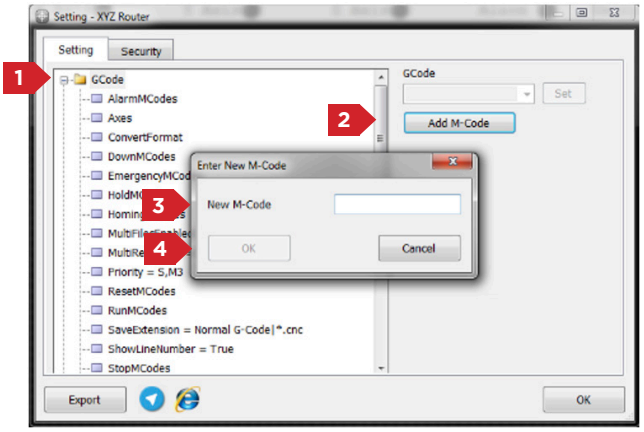
Table(6)

<table>
<tr><th></th><th>Name of Element</th><th>Descriptions</th></tr>
<tr><td>1</td><td>ControlStatus, Number</td><td>Different statuses of the controller and software can be viewed.
<table>
<tr><td>ControlStatus, Number</td><td>The number can be from 1 to 16.</td></tr>
<tr><td>ControlStatus, 1</td><td>When the controller goes online, the signal element turns on.</td></tr>
<tr><td>ControlStatus, 2</td><td>When a file is opened, the signal element turns on.</td></tr>
<tr><td>ControlStatus, 3</td><td>During the process of executing the Home command, the signal element turns on.</td></tr>
<tr><td>ControlStatus, 4</td><td>During the program execution, the signal element turns on.</td></tr>
<tr><td>ControlStatus, 5</td><td>When the Emergency input is turned on, the signal element turns on</td></tr>
<tr><td>ControlStatus, 6</td><td>When the axes have alarms, the signal element turns on.</td></tr>
<tr><td>ControlStatus, 7</td><td>When the registration code is entered, the signal element turns on.</td></tr>
<tr><td>ControlStatus, 8</td><td>When the software is in simulation mode, the signal element turns on.</td></tr>
<tr><td>ControlStatus, 9</td><td>When the Home operation is completed, the signal element turns on.</td></tr>
<tr><td>ControlStatus, 10</td><td>When the G-Code is executed in the forward direction (i.e. from the first line to the last line), the signal element turns on.</td></tr>
<tr><td>ControlStatus, 11</td><td>When the G-Code is executed in the backward direction (i.e. from the last line to the first line), the signal element turns on.</td></tr>
<tr><td>ControlStatus, 12</td><td>When the Hold function is activated, the signal element turns on. Please refer to row 26 of table 4 for more information.</td></tr>
<tr><td>ControlStatus, 13</td><td>When the Remote is activated, the signal element turns on.</td></tr>
<tr><td>ControlStatus, 14</td><td>When the limit input of any axis is activated, the signal element turns on.</td></tr>
<tr><td>ControlStatus, 15</td><td>When an axis is in Jog operation, the signal element turns on.</td></tr>
<tr><td>ControlStatus, 16</td><td>When the program execution is completed, the signal element turns on.</td></tr>
</table></td></tr>
<tr><td>2</td><td>EnabledStatus, Axis name</td><td>When the selected axis is active or enabled, the signal element turns on.
<table>
<tr><td>EnabledStatus, Axis name</td><td>The axis name can be any of X, Y, Z, A, B, or C.</td></tr>
</table></td></tr>
<tr><td>3</td><td>JogStatus, Axis name</td><td>When the selected axis is in Jog or manual movement, the signal element is turned on.
<table>
<tr><td>JogStatus, Axis name</td><td>The axis name can be any of X, Y, Z, A, B, C.</td></tr>
</table></td></tr>
<tr><td>4</td><td>RunStatus, Axis name</td><td>When the selected axis is in the Run mode, the signal element is turned on.
<table>
<tr><td>RunStatus, Axis name</td><td>The axis name can be any of X, Y, Z, A, B, C.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>5</td><td>ReadyStatus, Axis name</td><td>When the selected axis is in the Ready state and has no alarms, the signal element will turn on.<table><tr><td>ReadyStatus, Axis name</td><td>The axis name can be any of X, Y, Z, A, B, C.</td></tr></table></td></tr>
<tr><td>6</td><td>AlarmStatus, Axis name</td><td>If an alarm occurs in the designated axis, the signal element turns on.<table><tr><td>AlarmStatus, Axis name</td><td>The axis name can be any of X, Y, Z, A, B, C.</td></tr></table></td></tr>
<tr><td>7</td><td>LimitNStatus, Axis name</td><td>When the selected axis has reached the negative limit, the signal element turns on.<table><tr><td>LimitNStatus, Axis name</td><td>The axis name can be any of X, Y, Z, A, B, C.</td></tr></table></td></tr>
<tr><td>8</td><td>LimitPStatus, Axis name</td><td>When the selected axis is taken from the positive limit, the signal element turns on.<table><tr><td>LimitPStatus, Axis name</td><td>The axis name can be any of X, Y, Z, A, B, C.</td></tr></table></td></tr>
<tr><td>9</td><td>LimitStatus, Axis name</td><td>When the specified axis is taken to its limit, the signal element turns on.<table><tr><td>LimitStatus, Axis name</td><td>The axis name can be any of X, Y, Z, A, B, C.</td></tr></table></td></tr>
<tr><td>10</td><td>ErrorStatus, Axis name</td><td>When the specified axis has an Alarm or Limit, the signal element turns on.<table><tr><td>ErrorStatus, Axis name</td><td>The axis name can be any of X, Y, Z, A, B, C.</td></tr></table></td></tr>
</table>

Table(7)

GCode



Following the steps in the above image, a new M-code can be added. In other words, according to step 1, first, right-click on the GCode branch and then, according to step 2, by clicking on “Add M-Code”, a window called “Enter new M-Code” is opened. Then, according to step 3, the desired M-code name can be added to the window. For example, by typing M8 and pressing the OK button (according to step 4), M8 is added to the GCode branches.

GCode\AlarmMCodes

During an alarm, this variable is checked and if one or more M codes are stored in this variable, they will be executed.

- Note that to save an M code, simply type the desired M code and then press the “Set” button.
- Keep in mind that the M code must be defined under the GCode branch.
- To enter multiple M codes, simply separate them with commas. For example, enter “M8,M5” and then press the “Set” button.

GCode\Axes

This subtree is used to specify the axes from the ToolPath that the user wants to be considered in the GCode. For example, if the ToolPath includes the XYZ axes, by writing XY in this variable, only the X and Y axes will be considered in the GCode.

- By default, the software matches the axes that are read from the GCode exactly with the axes that exist in the ToolPath.

GCode\ConvertFormat

When converting a DXF file to GCode, the number of decimal places displayed in the GCode can be determined. For example, by entering 0.0000 in the relevant box, four decimal places are considered for all numbers in the DXF file. If the number of decimal places is greater, only four decimal places are considered, and if the number of decimal places is less or there are no decimals, zeros are added up to four decimal places.

- Note that the number of zeros before the decimal point determines the number of decimal places. For example, by entering 0.00 in the relevant box, two decimal places are considered for all numbers in the DXF file. If the number of decimal places is greater, only two decimal places are considered, and if the number of decimal places is less or there are no decimals, zeros are added up to two decimal places.
- Note that if # is used to determine the number of decimal places, and the actual number of decimal places is greater, the decimal places are reduced by the number of # signs. Otherwise, if the number of decimal places is less or there are no decimals, the same number is displayed. For example, by entering ###.0 in the relevant box, if the numbers in the DXF file have more decimal places, only three decimal places are considered, and if the number of decimal places is less or there are no decimals, the same number is displayed in the GCode without any change.

GCode\DownMCodes

When opening the GCode, if Tangent is active, this M code is used to move the Z-axis downwards.

- Note that to define an M code, simply type the M code and then press the Set button.
- Note that the M code must be defined under the GCode branch.
- If a jack is used to move the Z-axis, this command is used.
- To enter multiple M codes, simply separate them with a comma. For example, M78, M77 and then press the Set button.

GCode\EmergencyMCodes

If an M code or multiple M codes are defined in this section, they will be executed in case of an Emergency.

- To define an M code, simply type the code and press the Set button.
- Note that the M code must be defined under the GCode branch.
- To enter multiple M codes, simply separate them with commas. For example, enter “M8,M5” and then press the Set button.

GCode\HoldMCodes

This appears to be a duplicate of the previous message you asked me to translate. Here is the translation again:

- If an M code or codes are defined in this section, they will be executed during Hold functions.
- Note that to define an M code, you just need to write the code and press the Set button.
- Also note that the M code must be defined in the GCode subcategory.
- To enter multiple M codes, simply separate them with a comma. For example, enter “M78, M77” and then press Set.

GCode\HomingMCodes

If M codes are defined in this section, they will be executed during the execution of Home functions.

- Note that to define an M code, simply write the desired code and then press the Set button.
- Note that the M code must be defined under the GCode branch.
- To enter multiple M codes, simply separate them with a comma. For example, M8, M7, and then press the Set button.

GCode\InitializeMCodes

If an M code or multiple M codes are defined in this section, they will be executed at the beginning of the program.

- Note that to define an M code, you simply need to write the M code and then press the Set button.
- Keep in mind that the M code must be defined under the GCode branch.
- To enter multiple M codes, simply separate them with commas. For example, M8, M7, and then press the Set button.

GCode\MultiFileEnabled

The variable “MultiFileEnabled” is defined as True or False. If it is True, multiple files can be opened simultaneously. In other words, after selecting the desired files, they are opened in order and the next file is automatically opened after each file is finished, and it is only necessary to press the Run button to execute each file.

GCode\MultiReference

The variable MultiReference is defined as True or False.
If it is False and no reference is specified in the GCode, the reference set in the software page will be used as the criterion and the GCode will be executed based on that reference.

- If it is False and a reference is specified in the GCode, the reference specified in the GCode will be used as the criterion and the entire GCode will be executed based on that reference.
- If it is True, the reference can be changed within the GCode. For example, it can be specified that a part of the GCode should be executed using reference G54 and another part using reference G59. In this case, the machine will be initially positioned at reference G54 and the GCode will be executed, and when it reaches the GCode line that specifies reference G59, the machine will be moved to reference G59 and the remaining GCode will be executed based on that reference. (Note that G54 and G59 are mentioned as an example.)

GCode\Priority

The Priority variable is used when the ShortCut variable is in the True, General branch or is determined using the RunFromHere command to specify that the program is executed from a specific line.

- Please note that the ShortCut variable is used as a program shortcut, meaning that if the False, ShortCut variable is used, the machine will first move to the reference point or zero position and the program will be executed from the beginning. If the ShortCut variable is True, the machine is moved to the first cutting and executable line, and the program is executed from there, taking into account the last state the machine must have for the program to run, such as the last tool used, spindle speed, or any M code specified for program execution. The order of execution for M, T, P, and S codes can be determined in the Priority section. For example, if T is written first in the Priority section, tool change is performed first, and if S is written, spindle speed is determined. In fact, the order of execution for M, T, P, and S codes is determined in this section.
- If the RunFromHere command is executed to specify that the program is executed from a specific line, the last state the machine must have for the program to run is taken into account. This includes the last tool used, spindle speed, or any M code specified for program execution at the specified line. The order of execution for M, T, P, and S codes can be determined in the Priority section. For example, if T is written first in the Priority section, tool change is performed first, and if S is written, spindle speed is determined. In fact, the order of execution for M, T, P, and S codes is determined in this section.

GCode\ResetMCodes

If M codes are defined in this section, they will be executed when the Reset functions are run.

- Please note that in order to define an M code, you simply need to write the desired M code and then press the Set button.
- Also, please note that the M code must be defined under the GCode branch. To enter multiple M codes, simply place a comma between the desired M codes, such as M10, M11, and then press the Set button.

GCode\RunMCodes

If M codes are defined in this section, they will be executed during the execution of the Run functions.

- Note that to define an M code, simply write the desired M code and then press the Set button.
- Note that the M code must be defined under the GCode branch.
- To enter multiple M codes, simply separate them with a comma. For example, M10, M11 and then press the Set button.

GCode\SaveExtension

In this section, you can specify the file extension for the GCode format.

- Note that there are several useful extensions in the list provided in this section. Alternatively, you can enter or change the GCode extension as needed by right-clicking on one of the items in the list, typing the desired extension, and then pressing the Set button.

GCode\ShowLineNumber

The ShowLineNumber variable is defined as True or False.
If this variable is False, line numbers will not be displayed in the GCode display section (Figure 9, section 7).
If this variable is True, line numbers will be displayed in the GCode display section (Figure 9, section 7).

GCode\StopMCodes

If M codes are defined in this section, they will be executed during the execution of the Stop functions.

- Note that to define an M code, simply write the desired M code and then press the Set button.
- Note that the M code must be defined under the GCode branch.
- To enter multiple M codes, simply separate them with a comma. For example, M10, M11 and then press the Set button.

GCode\ToolChangerMCodes

If M codes are defined in this section, they will be executed during the tool change operation.

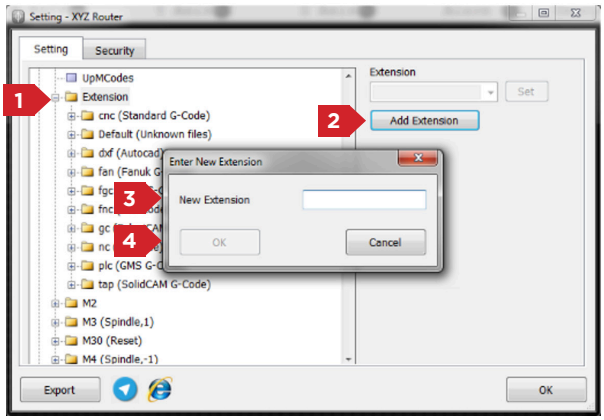
- Note that to define an M code, simply write the desired M code and then press the Set button.
- Note that the M code must be defined under the GCode branch.
- To enter multiple M codes, simply separate them with a comma. For example, M10, M11 and then press the Set button.

GCode\UpMCodes

When opening GCode, if the tangent is active, this M code is used to move the Z-axis upward.

- Note that to define an M code, simply write the desired M code and then press the Set button.
- Note that the M code must be defined under the GCode branch.
- If a jack is used to move the Z-axis, this command is used.
- To enter multiple M codes, simply separate them with a comma. For example, M78, M77 and then press the Set button.

GCode\Extension



According to the steps in the figure above, a new extension can be added. That is, according to step 1, right-click on the Extension branch, and then according to step 2, by pressing the Add Extension button, a window called “Enter new Extension” opens, and according to step 3, the desired Extension name can be added to the window. For example, by typing “tab” and pressing OK (step 4), “tab” is added to the sub-branches of Extension.

GCode\Extension\[ext]

[ext] can be any of the CNC formats (Standard G-Code), Default (Unknown files), fan (Fanuk G-Code), fgc (FGC G-Code), gc (FelexiCAM G-Code), fnc (F G-Code), Nc (NC Code), plc (GMS G-Code), tap (SolidCAM G-Code), or any other format defined by the user.
If a format is not specified in the GCode, it can be opened using this format. In other words, using the Default (Unknown files) format, all GCode formats can be opened.

Absolute IJk	The variable “Absolute IJk” is defined as True or False. If this variable is False and there is a arc in the GCode, the center of the arc is considered as incremental, which means that the center of the arc is adjusted relative to the last position of the machine. If this variable is True and there is an arc in the GCode, the center of the arc is considered as absolute, which means that the coordinates of the center of the arc are adjusted relative to the zero point of the workpiece or the set reference.

AxesCoefficient	This variable can be used to change the dimensions or movement direction of GCode for each axis using a multiplier. For example, if the ToolPath is XYZ and “1-,1,1” is typed in the corresponding box, then the X-axis is multiplied by 1, the Y-axis is multiplied by 1, and the Z-axis is multiplied by -1, which means that the direction of the Z-axis is reversed. Similarly, if the GCode unit is in meters and “100,100,100” is typed in the box, the GCode unit is converted to centimeters. • To understand which number corresponds to which axis multiplier, you can refer to the Axes section under the ToolPath branch in the System section. The multipliers are applied to the axes in the order they are listed in that section.
Change	To change GCode in the entire program, the program’s Replace function is used. For example, to change all M4 codes in the GCode to M8, simply type M4=M8 in the corresponding field and then press the Set button. • To apply multiple changes simultaneously to the GCode, simply separate the changes with commas. For example, type M4=M8,B=C and then press the Set button. • Note that the case sensitivity of characters is important, and the entered characters must be exactly the same as those written in the GCode in order for the changes to be applied correctly.
Tangent	The variable “Tangent” is defined as True or False. • If this variable is False, the tangent of the C-axis is not calculated. • If this variable is True, the tangent of the C-axis is calculated. • If the GCode has a C-axis, the Tangent variable must be set to False so that the tangent of the C-axis is not calculated and the C-axis inside the GCode is used.
Text	The text that represents the file format is entered into this variable.

GCode\Extension\dxf (Autocad)

AtTheEndGoZero	The variable “AtTheEndGoZero” is defined as True or False. • If it is False, no code will be added at the end of the GCode to move to the zero point of the workpiece or reference. • If it is True, code or codes will be added at the end of the GCode to move to the zero point of the workpiece or reference.
CurvePrecision	Parts of a DXF file containing curves are converted into broken lines by the software. To change the precision in creating these broken lines, the number in the “CurvePrecision” variable can be adjusted. For example, if the user requires high precision, the value of 0.0001 can be entered into the variable.
Grind	In some interfaces, large lines must be divided into smaller lines, and the length of these lines is defined in this variable.
JointPrecision	The JointPrecision variable is used to determine whether disconnected lines of shapes in dxf files should be connected to each other. If the distance between the connection points of the lines is smaller than the value of this variable, the lines are connected to each other. If the distance is greater than this value, the lines are treated separately. • Note that the JointPrecision variable can only be used if the Optimize variable is set to True.
MultiColor	The variable MultiColor is defined as True and False. • If this variable is False, all lines and cutting lines are considered regardless of the color they were designed with in the dxf file. • If this variable is True, the program distinguishes between the colors present in the dxf file and considers variables for each color. • To define variables for each color, you can apply the desired settings in the address GCode\Extension\dxf (Autocad).

<table>
<tr><td>Optimize</td><td>The variable “Optimize” is defined as True and False.
• If this variable is False, all codes are executed in the same order they were designed in the dxf file.
• If this variable is True, the shortest path is automatically considered, and the value set in the “JointPrecision” variable is taken into account.</td></tr>
<tr><td>PolylineOnly</td><td>When the order of executing elements inside the dxf file is important, by selecting the order of the elements and converting them to a Polyline, the dxf file is saved in the same order.
The variable “PolylineOnly” is defined as True and False.
• If this variable is False, all elements of the dxf file are considered.
• If this variable is True, only parts of the dxf file that are Polyline are considered.</td></tr>
<tr><td>Reverse</td><td>The variable “Reverse” is defined as True and False.
• If this variable is False, the order of executing dxf elements is not reversed, and the program’s elements are executed from the beginning to the end.
• If this variable is True, the order of executing dxf elements is reversed, and the program’s elements are executed from the end to the beginning.</td></tr>
<tr><td>ReverseObjects</td><td>variable “ReverseObjects” is defined as True or False.
• If the variable is False, the direction of execution for dxf objects is not reversed.
• If the variable is True, the direction of execution for dxf objects is reversed. For example, if a line is normally executed from point A to point B, with the “ReverseObjects” variable enabled, the line will be executed from point B to point A.</td></tr>
<tr><td>StartPoint</td><td>When a dxf file is opened, a frame is automatically created around the design, and all design elements are placed within that frame. The origin point of the frame can be defined in the following modes:
<table>
<tr><td>First Point</td><td>The first point of the DXF file is considered as the origin (0,0) point of the frame.</td></tr>
<tr><td>LeftBottom</td><td>The origin point is set to the bottom left corner of the frame.</td></tr>
<tr><td>RightBottom</td><td>The origin point is set to the bottom right corner of the frame.</td></tr>
<tr><td>LeftTop</td><td>The origin point is set to the top left corner of the frame.</td></tr>
<tr><td>RightTop</td><td>The origin point is set to the top right corner of the frame.</td></tr>
<tr><td>Middle</td><td>The origin point is set to the middle of the frame.</td></tr>
<tr><td>None</td><td>No change is made to the coordinate system of the DXF file.</td></tr>
</table></td></tr>
<tr><td>Tangent</td><td>The variable “Tangent” is defined as True or False. If this variable is False, the tangent of the C axis will not be calculated. If it is True, the tangent of the C axis will be calculated.</td></tr>
<tr><td>Text</td><td>The text that is displayed as the file format is defined in this variable.</td></tr>
</table>

GCode\Extension\dxf (Autocad)\[color]

This can refer to any of the colors Black, Blue, Cyan, Gray, Green, Magenta, Red, Silver, White, and Yellow.

Code	During the conversion from DXF to GCode, sections of the DXF file containing elements with the specified color ([color]) are added to the GCode according to the code written in the Code variable. For example, if the code M8 is typed in the Code variable, for each element with the [color] color, the code M8 is added to the GCode. • Note that the M code must be defined under GCode branches.

Enabled	The Enabled variable is defined as True and False. If this variable is False, the [color] element is not considered as a cutting element. In fact, this code is considered as the framework and basis for the DXF, but not as a cutting element. If this variable is True, the [color] element is considered as a cutting element.
FeedFactor	This variable is not active in the router interface. The speed of execution for elements with the [color] color can be determined by using the colors of the elements. For example, if the FeedFactor variable is set to 2.0, the speed of execution for the elements with the [color] color is multiplied by 2.0.
Offset	This variable is not active in the router interface. For each color, an offset can be considered, which is applied to the elements that have that color. For example, if a tool with a diameter of 6 millimeters is used during program execution, you can enter 3 millimeters in the Offset box. During the conversion from DXF to GCode, when the program reaches the lines with the [color] color, it realizes that the design is executed with a 6-millimeter tool, and automatically shifts the piece by 3 millimeters, so that the size of the tool does not affect the actual size. By placing a positive number, the piece shifts to the right of the path of movement, and by placing a negative number, the piece shifts to the left of the path of movement.
Traverse	The Traverse variable is defined as True and False. If this variable is False, elements with the specified color are considered as cutting elements. If this variable is True, elements with the specified color are considered as displacement elements and not as cutting elements.

GCode\M[n]

[n] can be any of the numbers 30, 5, 4, 3, 2, or 99.
In M codes, the number written next to M is the code number.

Enabled	The variable "Enabled" is defined as True or False. If it is False, M[n] will be disabled and ignored. If it is True, M[n] will be enabled.				
Extra	Reserve				
Link	☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

GCode\P

Enabled	The number written next to P represents the percentage of function performance. For example, P60 means that 60% of the function is performed, meaning that if the function is speed, it will be executed at 60% of the set speed. The variable "Enabled" is defined as True or False. If this variable is False, P is disabled in general and ignored. If this variable is True, P is enabled.				
Link	☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

GCode\S

Enabled	The variable "Enabled" is defined as True or False. If this variable is False, S is generally disabled and ignored. If this variable is True, S is enabled. • Note that the code S is usually used for spindle speed, and in this case, the number written next to S can be used to determine the spindle speed. For example, S1800.				
Link	☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

GCode\T

Enabled	The variable "Enabled" is defined as True or False. If it is False, the T code is generally disabled and ignored. If it is True, the T code is enabled. • Note that the T code is usually used for tool changing, and in this case, the number next to T specifies the tool number. For example, T1 indicates tool number 1.				
Link	☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Table(8)

General

General\DefaultReference

During software startup, it is determined which reference will be the default. For example, if DefaultReference = 54, G54 reference will be used.
The last reference saved through the SetReference function is stored in the DefaultReference variable.

General\FeedStep

This variable represents the step size for changes in the Scroll feed. By changing the FeedStep value, the amount of change per step can be determined, which is how much the Feed speed is increased or decreased for each unit of movement in the Scroll. For example, if FeedStep = 10, the Feed speed is increased or decreased by 10% for each unit of movement in the Scroll.

General\FirstGoHome

The FirstGoHome variable is defined as True or False.
If this variable is False, FirstGoHome is generally disabled and ignored.
If this variable is True, the machine is automatically moved to the Home coordinates after software startup.

General\JogFeedStep

This variable represents the step size for changes in the Scroll jog feed. By changing the JogFeedStep value, the amount of change per step can be determined, which is how much the JogFeed speed is increased or decreased for each unit of movement in the Scroll. For example, if JogFeedStep = 10, the JogFeed speed is increased or decreased by 10% for each unit of movement in the Scroll.

General\KeepLocation

KeepLocation variable is defined as True or False.
If this variable is False, KeepLocation is generally disabled and ignored.
If this variable is True, the last coordinates are saved if anything happens that causes the user to exit the Radonix software. When the software is used again, it will be able to execute the program from those coordinates. For example, if there is a power outage, the last position is saved and the program can be resumed from that position after the power is restored and the software is reopened.

General\MaxFeed

This variable is used to determine the maximum value of Scroll and Feed. By changing the value of MaxFeed, the maximum amount of Scroll and Feed can be determined. For example, if MaxFeed=100, the Scroll and Feed can be changed up to a maximum of 100%

General\MaxJogFeed

This variable is used to determine the maximum value of Scroll and JogFeed. By changing the value of MaxJog-Feed, the maximum amount of Scroll and JogFeed can be determined. For example, if MaxJogFeed=100, the JogFeed can be changed up to a maximum of 100% in the Scroll.

General\OpenRecentFile

The variable OpenRecentFile is defined as True or False. If this variable is False, OpenRecentFile is generally disabled and ignored. If this variable is True, the last used file is automatically opened after starting the software.

General\ParkEnabled

The variable ParkEnabled is defined as True or False. If this variable is False, ParkEnabled is generally disabled and ignored. If this variable is True, the machine is automatically moved to the Park coordinates after completing the program.

- Note that the Park coordinates are fully explained in the General[axis]\Park address in this table, please refer to it for more information.

General\ShortCut

The variable ShortCut is defined as True or False. The ShortCut variable is used for program shortcuts. If this variable is False, the machine is moved to the reference point or zero point before the program starts and all codes are executed from the beginning. If this variable is True, the machine is moved to the first cutting and executable line, and the program starts. In this case, codes such as P, T, S, and M that must be used to execute the program may exist before the program starts and are checked, and the last state that the machine must have for executing the program is taken into account. For example, if the T code is placed first in the Priority section, the tool is first replaced, and if it is the S code, the spindle speed is determined, and so on, the order of execution of the P, T, S, and M codes is determined.

- Note that if the ShortCut variable is True in the GCode section, the Priority variable is used.

General\SimpleDelay

the SimpleDelay variable is completely related to the SimpleDelay function (Table 4) and the SimpleDelayPin output (Table 3), the complete process operation is explained in this section. To make it clear and understandable during the process definition, the link used is mentioned, and it is sufficient to refer to the relevant table according to the name of the pin or function and define the links. When the SimpleDelay command (link to the function) is executed, depending on the argument entered in the SimpleDelay function, the SimpleDelayPin output is turned on or off, and then a delay is created for the time set by this variable.

- Note that the SimpleDelay function has an argument, and when defining the SimpleDelay function, refer to Table 4.

General\[axis]

Please note that based on the installed interface, [axis] can be any of the X, Y, Z, A, B, C axes. All settings are the same for all axes, with the difference that the desired settings are created on the selected axis.

General\[axis]\JogLocation

Given that the JogLocation variable is completely related to the JogLocation parameter (Table 6) and the AJog function (Table 4), the complete functionality of the process is explained in this section. To make it clear and complete, the link used is mentioned when defining the process, and it is sufficient to refer to the relevant table and links based on the name of the parameter or function.
The JogLocation variable is the destination coordinates when moving manually or, in other words, Jogging. For example, if the JogLocation value is entered as 100, the [axis] axis will be moved to coordinate 100 by running the AJog function.

- Note that this variable is similar to the JogLocation parameter (Table 6) with the difference that the JogLocation parameter (Table 6) has an argument for specifying the axes.
- Also note that, based on the installed interface, [axis] can be any of the X, Y, Z, A, B, or C axes.

General\[axis]\JogMaxVelocity

This variable is used to determine the maximum speed of Jog movement in the [axis] axis, and by changing the value of JogMaxVelocity, the maximum speed of Jog movement in the [axis] axis can be determined. For example, if JogMaxVelocity=100, the maximum Jog movement speed in the [axis] axis can be increased up to 100 millimeters per second.

- Note that according to the installed interface, [axis] can be any of the X, Y, Z, A, B, or C axes.

General\[axis]\JogStep
This variable is used to determine the step size in the [axis] axis during manual or Jog movement in an incremental manner. • Note that this variable is similar to the JogStep parameter (Table 6), with the difference being that the JogStep parameter has an argument for determining the axes. • Also note that according to the installed interface, [axis] can be any of the X, Y, Z, A, B, or C axes.
General\[axis]\JogVelocity
This variable is used to set the movement speed during manual or Jog movement on the [axis] axis for both Incremental and continuous modes. • Note that this variable is similar to the JogVelocity parameter (Table 6) with the difference that JogVelocity parameter (Table 6) has arguments to determine the axes. • Note that according to the installed interface, [axis] can be any of the X, Y, Z, A, B, or C axes.
General\[axis]\Park
variable is used to determine the Park coordinates. For example, if the value of Park is set to 100 and the variable ParkEnabled is active or True, after the program execution is completed, the [axis] axis will be automatically moved to coordinate 100. • Note that this variable is similar to the Park parameter in Table 6, with the difference being that the Park parameter in Table 6 has arguments to determine the axes. • Also note that according to the installed interface, [axis] can be any of the X, Y, Z, A, B, or C axes.
General\[axis]\StopDisplacement
After executing the Stop function during program execution, the [axis] axis will be displaced by the amount stored in the StopDisplacement parameter. For example, if the value of StopDisplacement is set to 10 millimeters, after executing the Stop function and the program is stopped, the [axis] axis will be automatically displaced by 10 millimeters in the positive direction. • Note that according to the installed interface, [axis] can be any of the X, Y, Z, A, B, or C axes.

Table(9)

JoyStick

1. Left Trigger { LeftTriggerA, LeftTrigger+D }
2. Right Trigger { RightTriggerA, RightTrigger+D }
3. Left Bumper { LeftBumperD }
4. Right Bumper { RightBumperD }
5. Back Button { BackButtonD }
6. Start Button { StartButtonD }
7. X Button { XButtonD }
8. Y Button { YButtonD }
9. Left Stick { LeftStickButtonD, LeftStickXA, LeftStickX-D, LeftStickX+D, LeftStickYA, LeftStickY-D, LeftStickY+D }
10. B Button { BButtonD }
11. A Button { AButtonD }
12. Directional Pad { DPadLeftD, DPadRightD, DPadUpD, DPadDownD }
13. Right Stick { RightStickButtonD, RightStickXA, RightStickX-D, RightStickX+D, RightStickYA, RightStickY-D, RightStickY+D }
14. Note that (A: Analog, D: Digital).

JoyStick/Enabled

The variable “Enabled” is defined as True or False.
If it is False, the Joystick is generally disabled and ignored.
If it is True, the Joystick is enabled.

JoyStick/ExponentialTrigger

The variable “ExponentialTrigger” is defined as True or False.
- If it is False, changes are linear. For example, when increasing speed, the speed increases linearly from 10 to 20 and then to 30.
- If it is True, changes are exponential. For example, when increasing speed, the speed increases exponentially from 10 to 25 and then to 80.

JoyStick/AButton/Enabled

The variable “Enabled” is defined as True or False.
- If it is False, the AButton is generally disabled and ignored.
- If it is True, the AButton is enabled according to the defined link.

JoyStick/AButton/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/BackButton/Enabled

The variable “Enabled” is defined as a Boolean (True or False).
- If this variable is False, the BackButton is disabled and ignored.
- If this variable is True, the BackButton is enabled according to the specified link.

JoyStick/BackButton/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/DPadLeft/Enabled

The variable “Enabled” is defined as True or False.
- If this variable is False, DPadLeft is completely disabled and ignored.
- If this variable is True, DPadLeft is enabled based on the defined link

JoyStick/DPadLeft/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/DPadRight/Enabled

The variable Enabled is defined as True or False.
- If this variable is False, DPadRight is generally disabled and ignored.
- If this variable is True, DPadRight is enabled according to the specified link.

JoyStick/DPadRight/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/DPadUp/Enabled

The variable “Enabled” is defined as a Boolean (True/False) value.
- If this variable is False, DPadUp is disabled and ignored.
- If this variable is True, DPadUp is enabled according to the specified link.

JoyStick/DPadUp/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/LeftBumper/Enabled

The variable “Enabled” is defined as True or False.
- If this variable is False, the LeftBumper is generally disabled and ignored.
- If this variable is True, the LeftBumper is enabled based on the specified link.

JoyStick/LeftBumper/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/LeftStickButton/Enabled

The variable Enabled is defined as True and False.
- If this variable is False, LeftStickButton is generally disabled and ignored.
- If this variable is True, LeftStickButton is activated according to the specified link.

JoyStick/LeftStickButton/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/LeftStickX/Enabled

The variable “Enabled” is defined as True or False.
- If this variable is False, “LeftStickX” is generally disabled and ignored.
- If this variable is True, “LeftStickX” is enabled according to the specified link.

JoyStick/LeftStickX/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/LeftStickX-/Enabled

The variable Enabled is defined as True and False.
- If this variable is False, LeftStickX- is generally disabled and ignored.
- If this variable is True, LeftStickX- is enabled according to the specified link.

JoyStick/LeftStickX-/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/LeftStickX+/Enabled

The variable Enabled is defined as True or False.
- If this variable is False, LeftStickX+ is generally disabled and ignored.
- If this variable is True, LeftStickX+ is activated according to the specified link.

JoyStick/LeftStickX+/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/LeftStickY/Enabled

The variable “Enabled” is defined as a Boolean variable, with a value of either True or False.
- If the variable is False, the LeftStickY is generally disabled and ignored.
- If the variable is True, LeftStickY is enabled and active according to the designated link.

JoyStick/LeftStickY/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/LeftStickY-/Enabled

The variable Enabled is defined as True or False.
- If this variable is False, LeftStickY- is disabled and ignored.
- If this variable is True, LeftStickY- is activated based on the specified link.

JoyStick/LeftStickY-/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/LeftStickY+/Enabled

The variable Enabled is defined as True or False. If this variable is False, LeftStickY+ is completely disabled and ignored. If this variable is True, LeftStickY+ is activated according to the specified link.

JoyStick/LeftStickY+/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/LeftTrigger/Enabled

The variable Enabled is defined as True and False.
If this variable is False, LeftTrriger is disabled entirely and ignored.
If this variable is True, LeftTrriger is activated based on the assigned link.

JoyStick/LeftTrigger/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/LeftTrigger+/Enabled

The variable “Enabled” has been defined as true and false.
If this variable is false, LeftTrigger+ is generally disabled and ignored.
If this variable is true, LeftTrigger+ is activated according to the specified link.

JoyStick/LeftTrigger+/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/RightBumper/Enabled

The variable “Enabled” is defined as True or False.
- If this variable is False, the “RightBumper” is completely disabled and ignored.
- If this variable is True, the “RightBumper” is activated according to the specified link.

JoyStick/RightBumper/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/RightStickButton/Enabled

Variable Enabled is defined as True or False.
- If this variable is False, RightStickButton is generally disabled and ignored.
- If this variable is True, RightStickButton is enabled based on the assigned link.

JoyStick/RightStickButton/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/RightStickX/Enabled

Variable Enabled is defined as True or False.
- If this variable is False, RightStickX is generally disabled and ignored.
- If this variable is True, RightStickX is enabled based on the assigned link.

JoyStick/RightStickX/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/RightStickX-/Enabled

Variable Enabled is defined as True or False.
- If this variable is False, RightStickX- is generally disabled and ignored.
- If this variable is True, RightStickX- is enabled based on the assigned link.

JoyStick/RightStickX-/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/RightStickX+/Enabled

Variable Enabled is defined as True or False.
- If this variable is False, RightStickX+ is generally disabled and ignored.
- If this variable is True, RightStickX+ is enabled based on the assigned link.

JoyStick/RightStickX+/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/RightStickY/Enabled

Variable Enabled is defined as True or False.
- If this variable is False, RightStickY is generally disabled and ignored.
- If this variable is True, RightStickY is enabled based on the assigned link.

JoyStick/RightStickY/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/RightStickY-/Enabled

Variable Enabled is defined as True or False.
- If this variable is False, RightStickY- is generally disabled and ignored.
- If this variable is True, RightStickY- is enabled based on the assigned link.

JoyStick/RightStickY-/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/RightStickY+/Enabled

Variable Enabled is defined as True or False.
- If this variable is False, RightStickY+ is generally disabled and ignored.
- If this variable is True, RightStickY+ is enabled based on the assigned link.

JoyStick/RightStickY+/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/RightTrigger/Enabled

Variable Enabled is defined as True or False.
- If this variable is False, RightTrigger is generally disabled and ignored.
- If this variable is True, RightTrigger is enabled based on the assigned link.

JoyStick/RightTrigger/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/RightTrigger+/Enabled

Variable Enabled is defined as True or False.
- If this variable is False, RightTrigger+ is generally disabled and ignored.
- If this variable is True, RightTrigger+ is enabled based on the assigned link.

JoyStick/RightTrigger+/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/StartButton/Enabled

Variable Enabled is defined as True or False.
- If this variable is False, StartButton is generally disabled and ignored.
- If this variable is True, StartButton is enabled based on the assigned link.

JoyStick/StartButton/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/XButton/Enabled

Variable Enabled is defined as True or False.
- If this variable is False, XButton is generally disabled and ignored.
- If this variable is True, XButton is enabled based on the assigned link.

JoyStick/XButton/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

JoyStick/YButton/Enabled

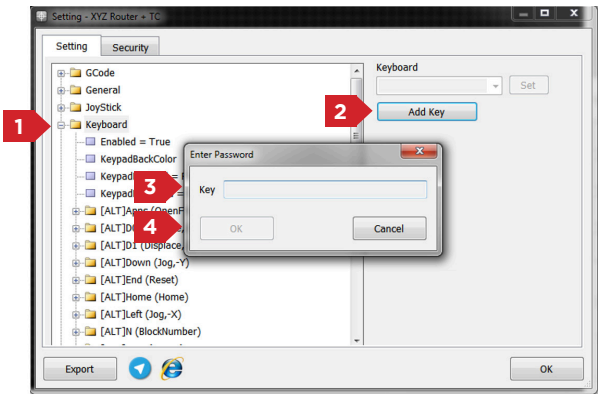
Variable Enabled is defined as True or False.
- If this variable is False, YButton is generally disabled and ignored.
- If this variable is True, YButton is enabled based on the assigned link.

JoyStick/YButton/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Table(10)

Keyboard



As shown in the steps illustrated in the above figure, a new Shortcut can be added. First, as per step 1, click the Keyboard branch with the left mouse button, then, as per step 2, by pressing Add Key, the Enter Key window opens, and then, as per step 3, by pressing the desired key or keys, the Shortcut is added. For example, by pressing Ctrl, Alt, and G, and then pressing the OK button (as per step 4), the new Shortcut is added to the Keyboard branch.
Note that by default, several Shortcuts are defined, and by referring to the table related to the function links (Table 4), you can become familiar with the functionality of each defined Shortcut. Additionally, according to the user's needs, a new Shortcut can be defined by selecting the desired link (Table 4)

Keyboard/Enabled

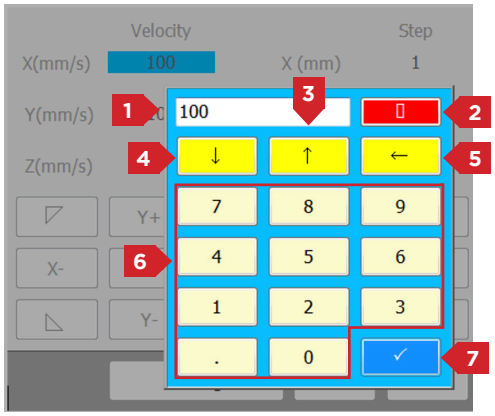
The Enabled variable is defined as True or False. If this variable is False, the keyboard is disabled and ignored. If it is True, the keyboard is enabled.

Keyboard/KeypadBackColor

This variable is used to change the background color of the Keypad window.

Keyboard/KeypadEnabled

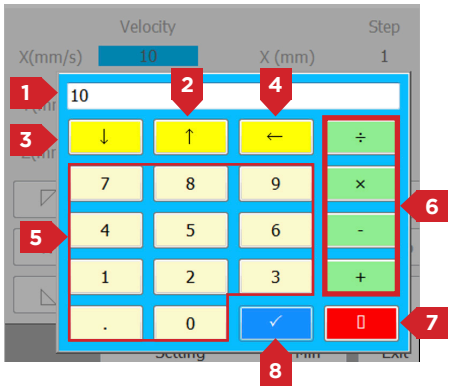
The KeypadEnabled variable is defined as True or False. If this variable is False, the Keypad is disabled and ignored. If it is True, the Keypad is enabled, and you can enter or modify a number in the Value element using this window.



1. Used to enter the desired number.
2. Used to close the opened window.
3. Used to add one unit to the number in section 1.
4. Used to subtract one unit from the number in section 1.
5. Used to clear the number in section 1.
6. The desired number can be entered using the available digits. Additionally, if decimal numbers are needed, the . key can be used.
7. Used to confirm and enter the number.

Keyboard/KeypadExtended

The variable KeypadExtended is defined as True or False. If this variable is False, KeypadExtended is generally disabled and ignored. If this variable is True and the KeypadEnabled variable is also True, KeypadExtended is enabled, and you can enter the desired number and perform the four basic arithmetic operations (addition, subtraction, multiplication, and division) using this window.



1. Used to enter the desired number.
2. Used to add one unit to the number in section 1.
3. Used to subtract one unit from the number in section 1.
4. Used to clear the number in section 1.
5. The desired number can be entered using the available digits, and if decimal numbers are needed, the button . can be used.
6. Used to perform the four basic arithmetic operations (addition, subtraction, multiplication, and division).
7. Used to confirm and enter the number.
8. Used to close the opened window.

Keyboard/[ALT]Apps (The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True or False.
If this variable is False, the corresponding shortcut will be disabled and ignored.
If it is True and the selected link is valid, the corresponding shortcut (which involves pressing Alt and Apps keys simultaneously) will be activated based on the selected link.

- According to the fact that all links in the shortcuts are fully explained in the table related to the functions (Table 4), for more information, refer to Table 4.

Keyboard/[ALT]Apps (The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]D0(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False. If this variable is False, the desired shortcut is generally disabled and ignored. If this variable is True, the desired shortcut, which is pressing both the D0 and Apps buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information.

Keyboard/[ALT]D0(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]D1(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and D1 buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information.

Keyboard/[ALT]D1(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]Down(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and Down buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information.

Keyboard/[ALT]Down(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]End(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and End buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information.

Keyboard/[ALT]End(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]Home(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and Home buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information

Keyboard/[ALT]Home(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]Left(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and Left buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information

Keyboard/[ALT]Left(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]N(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and N buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information

Keyboard/[ALT]N(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]Next(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and Next (Page down) buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information

Keyboard/[ALT]Next(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]OemMinus(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and OemMinus (-) buttons simultaneously, will be enabled based on the selected link.

,

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information

Keyboard/[ALT]OemMinus(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]Oemplus(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and Oemplus (+) buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information

Keyboard/[ALT]Oemplus(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]Pageup(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and Page down buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information

Keyboard/[ALT]Pageup(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]Return(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and Return (Enter) buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information

Keyboard/[ALT]Return(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]Right(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and Right buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information

Keyboard/[ALT]Right(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]Space(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and Space buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information

Keyboard/[ALT]Space(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]T(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and T buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information

Keyboard/[ALT]T(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]Up(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and Up buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information

Keyboard/[ALT]Up(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]X(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and X buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information

Keyboard/[ALT]X(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]Y(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and Y buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information

Keyboard/[ALT]Y(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Keyboard/[ALT]Z(The link defined in the Link section is displayed.) / Enabled

The variable “Enabled” is defined as True and False.
If this variable is False, the desired shortcut is generally disabled and ignored.
If this variable is True, the desired shortcut, which is pressing both the Alt and Z buttons simultaneously, will be enabled based on the selected link.

- Considering that all the links in the shortcuts are fully described in the table related to the functions (Table 4), please refer to Table 4 for more information

Keyboard/[ALT]Z(The link defined in the Link section is displayed.) / Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Table(11)

Subcategory References

References
References\G54
This variable is used to determine and store the coordinates of the G54 reference point.
References\G55
This variable is used to determine and store the coordinates of the G55 reference point.
References\G56
This variable is used to determine and store the coordinates of the G56 reference point.
References\G57
This variable is used to determine and store the coordinates of the G57 reference point.
References\G58
This variable is used to determine and store the coordinates of the G58 reference point.
References\G59
This variable is used to determine and store the coordinates of the G59 reference point.

Table(12)

Setting up the remote

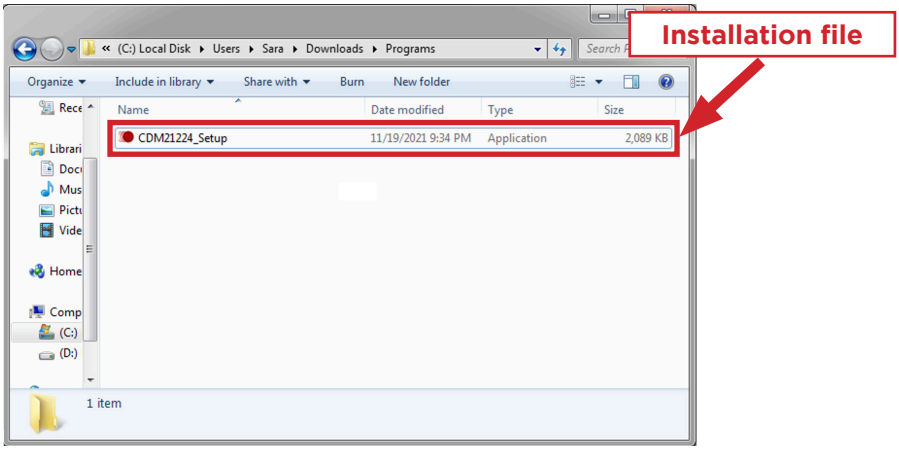
To set up the remote, first connect the remote receiver to the computer via the cable. This will turn on the receiver's internal light. Then, some settings must be made in the software in the Remote section under Settings.

Note that before connecting the cable to the computer, the USB driver with the name CDM21XXX must be installed so that the receiver can be identified by the computer after connecting the cable to the computer.

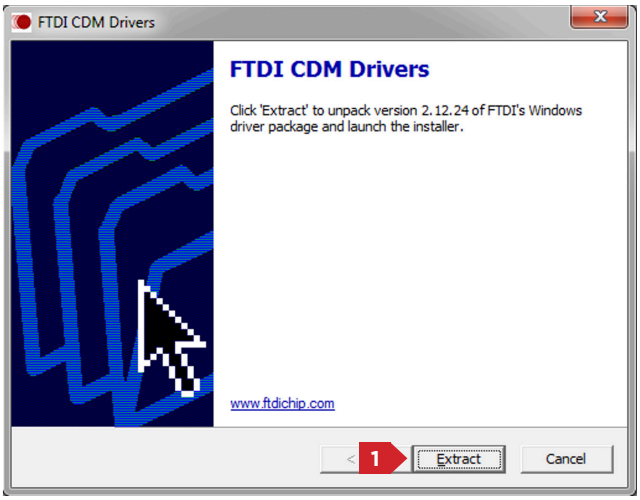
Figure(1) - CDM21XXX download link

CMD21XXX installation

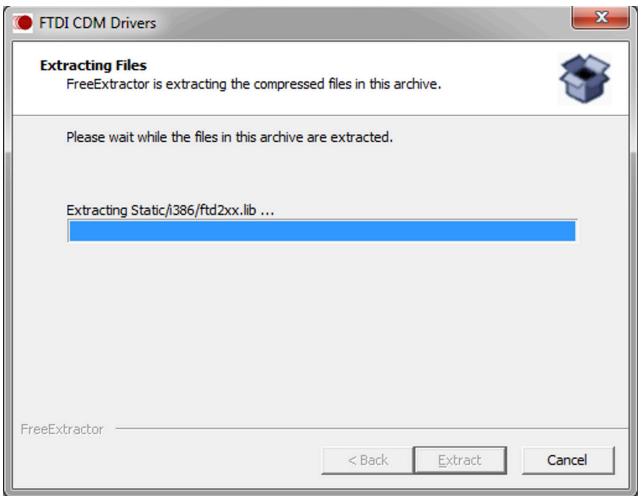
After downloading the CDM21228 software, double-click on the downloaded file and follow the steps below.



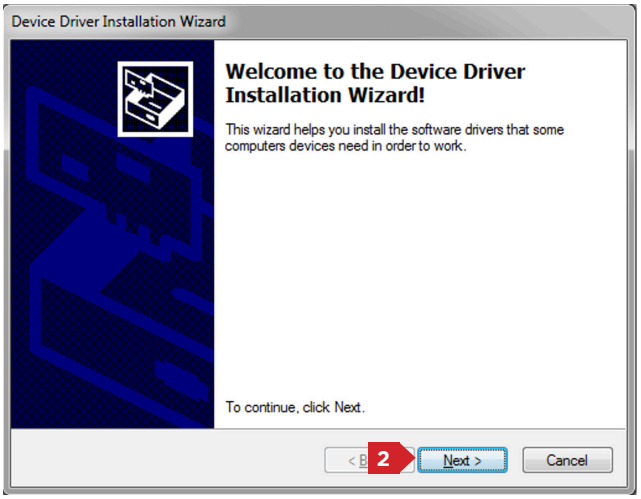
Figure(2)



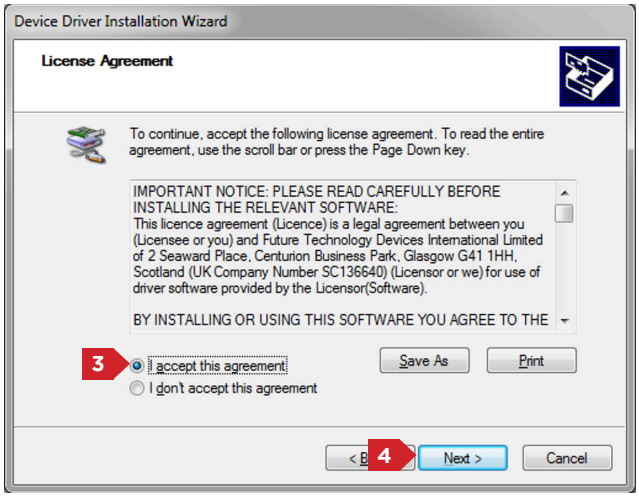
Figure(4)



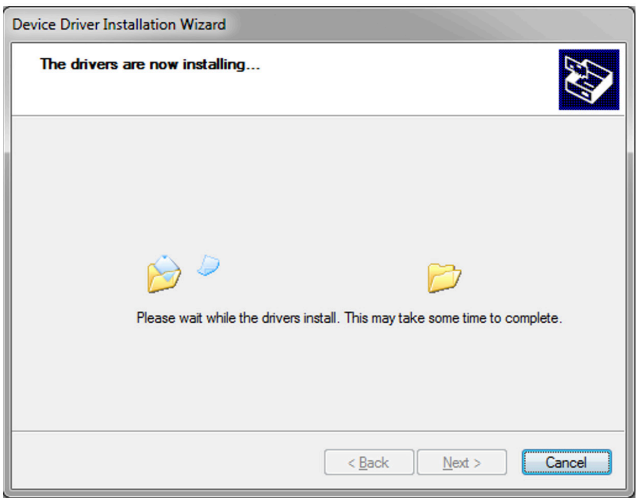
Figure(5)



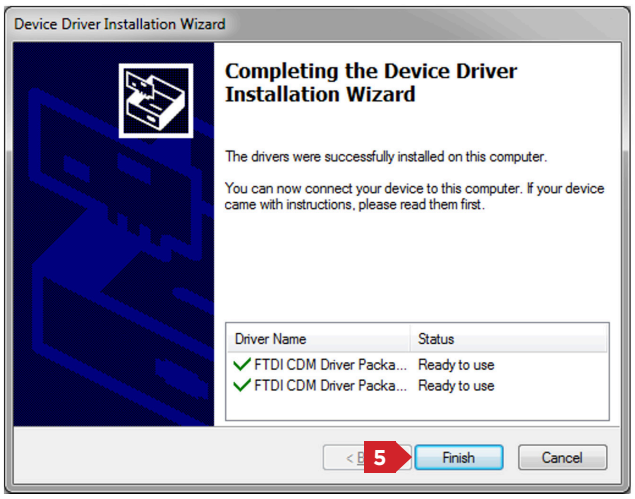
Figure(6)



Figure(7)



Figure(8)

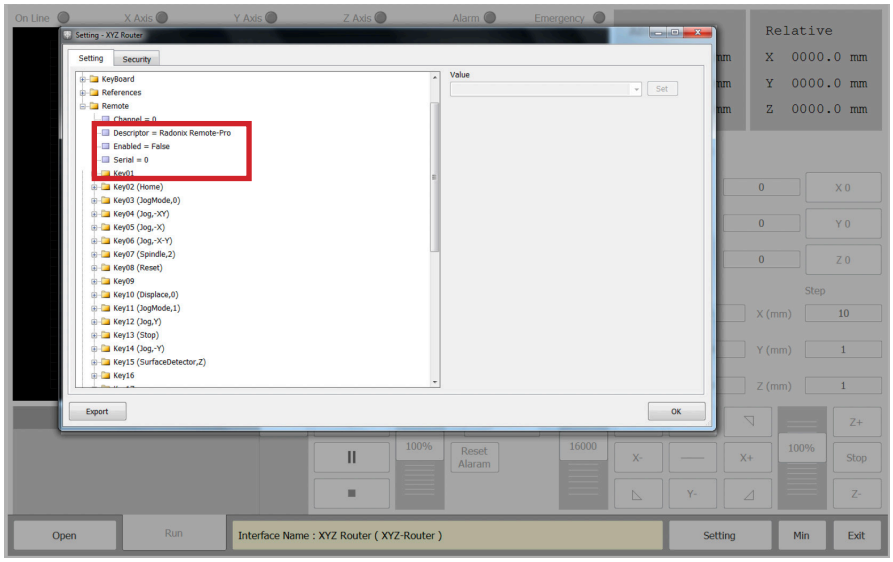


Figure(9)

The software has been installed correctly.

Configuration

To set up the remote, a series of software settings must be performed. In the CAM-Pro software, enter the Remote section from the Settings menu, open the Remote branch, and you will see 4 sub-branches. (Figure 10)



Figure(10)

Remote

Remote/Channel

This variable is used to determine the remote channel. The remote channel is a number between 1 and 32 that needs to be entered in this section.

How to change the remote transmitter channel:
There are 32 buttons on the transmitter with corresponding numbers, as shown in the figure below, that the operator can use to select the transmitter channel. The method for doing this is as follows:

First, simultaneously press the two buttons named Reset and Pause (marked with the abbreviations RST and ||) in the last row of the transmitter buttons. The red light on the transmitter will start flashing continuously, and after a few seconds, the light will turn on uniformly. Then, release the buttons, and by pressing any of the keyboard buttons with the corresponding number, it will be saved as the new remote channel. (The number corresponding to each button is written next to that button in the figure. For example, F12 corresponds to button number 9.) In the last step, the selected new number should be entered in this Channel section to set the new channel completely.

- The Reset and Pause buttons are used to change the channel.

Remote/Descriptor

This variable is used to determine the type of remote. In fact, through this section, it can be specified whether the remote is related to the Pro or Plus controller model.

Remote/Enabled

The Enabled variable is defined as True and False. If this variable is False, the remote is disabled completely and ignored. If this variable is True, the remote is activated.

Remote/Serial

This variable is used to determine the remote serial number. In fact, the remote serial is a 5-digit code that is located on the back of each remote.

- Each remote has a different serial number.

Remote/Key[n]

Since all Remote sub-branches from Key01 to Key32 have a common definition, Key[n] is used to provide a comprehensive definition for all Remote sub-branches. In fact, [n] can be any number from 1 to 32.
There are 32 buttons on the transmitter, each of which can be defined based on its button number. (The button numbers are provided in this table in the explanations of the Channel sub-branch). Therefore, if any number from 1 to 32 is used instead of [n], it will be the identifier of the Remote sub-branches and button number.

Remote/Key[n]/Enabled

The Enabled variable is defined as True and False. If this variable is False, the remote button is disabled completely and ignored. If this variable is True, the button on the remote is activated based on the specified link. For example, if Key02 is True in the Enabled section, button number 2 will be activated.

Remote/Key[n]/Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

- You should check the user guide for learn more about related tables

Table(13)

Subcategory Router

Router
Router\KeepZUp
This variable is used to determine the distance of the Z-axis from the reference point when the machine axes are moved to the reference point. For example, if the entered number is 0.0, the Z-axis will be located at the reference point. If the entered number is 20, the Z-axis will move 20 units in the positive direction from the reference point.
Router\SafeZ
Generally, when moving the axes, the Z-axis is moved to its highest point and after moving the other axes, it moves to the desired point. With this variable, instead of moving the Z-axis to its highest point, it moves up or down from the reference point by the SafeZ value For example, if the highest point of the Z-axis is 0 and the lowest point is 250 and this variable is set to 100, the machine will move the Z-axis 100 units up from the reference point instead of moving it to its highest point.
Router\SpindleAutoOff
This variable is used for GCodes that do not have an M-code to turn off the spindle. The SpindleAutoOff variable is defined as True or False. If this variable is False, SpindleAutoOff is generally disabled and ignored. If this variable is True, SpindleAutoOff is enabled, and at the end of executing the GCode, the spindle is automatically turned off.
Router\SpindleAutoOn
This variable is used for GCodes that do not have an M-code to turn on the spindle. The SpindleAutoOn variable is defined as True or False. If this variable is False, SpindleAutoOn is generally disabled and ignored. If this variable is True, SpindleAutoOn is enabled, and during the execution of the GCode, the spindle is automatically turned on.
Router\SpindleDelay
This variable is used to set the delay time for the spindle to reach the desired speed or stop. The unit for this variable is milliseconds.
Router\SpindleMaxSpeed
This variable is used to determine the maximum speed that can be set for the spindle.
Router\ SpindleMinSpeed
This variable is used to determine the minimum speed that can be set for the spindle.
Router\ SpindleSpeed
This variable is used to store the speed value set for the spindle. For example, if the user sets the Scroll for spindle speed to 18000, the number 18000 is stored in this variable.
Router\SpindleSpeedStep
This variable determines the step change for the SpindleSpeed when using the Scroll. By changing the value of SpindleSpeedStep, the amount of change for each step can be determined. For example, if SpindleSpeedStep is set to 2000, the spindle speed will change by 2000 for each unit of movement in the Scroll.

Table(14)

Subcategory RS485

RS485

This branch can be used if an analog module with four analog inputs is used. In fact, by connecting the analog module to the RS485 port of the controller, all four analog sub-branches can be used in the RS485 branch.

RS485\Enabled

The Enabled variable is defined as True and False.
If this variable is False, the RS485 branch is generally disabled and ignored.
If this variable is True, the RS485 branch is activated.

RS485\AnalogInput

Note that if the link defined for the analog module input is similar to the link for the Scroll on the software page, the Scroll is deactivated. For example, if a volume is connected to the analog module and a link is defined for it so that the spindle speed can be controlled through that link, then the Scroll defined for changing the spindle speed is automatically deactivated.

RS485\AnalogInput\Analog[n]\Enabled

There are four analog inputs on the analog module, so instead of [n], any number from 1 to 4 can be used. In fact, because all the AnalogInput sub-branches from Analog1 to Analog4 have a common definition, Analog[n] is used.
The AnalogX variable is defined as True and False.
If this variable is False, the analog input is generally disabled and ignored.
If this variable is True, the analog input is activated according to the defined link. For example, if Analog2 is True in the Enabled section, Analog 2 is activated.

RS485\AnalogInput\Analog[n]\Link

☐ InPort Table(2)	☐ OutPort Table(3)	☐ Function Table(4)	☑ Analog Table(5)	☐ Value Table(6)

Table(15)

Subcategory System

- All fundamental system variables are located in this branch. Therefore, this section is highly sensitive and illogical changes may disrupt program execution or damage the machine's mechanics.

System

System\AlarmEnabled

The variable AlarmEnabled is defined as True or False. If it is False, AlarmEnabled is generally inactive and ignored. If it is True and the wiring for the alarm motors is done correctly (as described in chapter two), then if the motors have an alarm, the alarm is triggered. Additionally, if the wiring for the alarm section is incorrect or not done, the alarm is triggered again.
Note that in the event of an error in any axis, all axes will stop.
Also note that in the motor stop command, as there is no alarm output, the variable AlarmEnabled must be set to False.

System\Connection

This variable is used to determine the type of connection between the controller and the computer and can be defined as LAN or USB depending on the type of controller used.

System\EmergencyType

This variable is used to determine the operation that occurs when the Emergency button is pressed.

None	When Emergency is triggered, the device stops. It is similar to the Pause command, except that it cannot be resumed.
ClearOutPorts	When Emergency is triggered, in addition to stopping the axes, the digital output ports of the controller are turned off.
ClearAnalog	When Emergency is triggered, in addition to stopping the axes, the analog outputs are turned off.
DisableAxes	When Emergency is triggered, the axes are stopped and become inactive, and in this case, the motors become free.
ResetToolPath	When Emergency is triggered, in addition to stopping the axes, the ToolPath is reset and returns to the first coordinate in the program.
ClearOutPorts+ClearAnalog	When Emergency is triggered, both the digital output ports and the analog outputs are turned off, the axes are stopped and become inactive, the motors become free, and the ToolPath is reset and returns to the first coordinate in the program.
All	When Emergency is triggered, the digital output ports and analog outputs are turned off, the axes are stopped and become inactive, the motors become free, the ToolPath is reset and returns to the first coordinate in the program.

System\HomeNecessary

This variable is used to determine the importance of having a Home position for the device.

0	In this case, whether or not the device has a Home position is not important, and both the axes can be moved manually and the program can be executed.

0	In this case, whether or not the device has a Home position is not important, and both the axes can be moved manually and the program can be executed.
1	In this case, if the device does not have a Home position, the axes can still be moved manually but the program cannot be executed.
2	In this case, if the device does not have a Home position, neither the axes can be moved manually nor can the program be executed, and the device must first have a Home position so that both the axes can be moved manually and the program can be executed.

System\HomeOrder

This variable is used to determine the order in which the axes are homed. For example, Z,XY, where in this case the Z axis is homed first, and after the Z axis homing is complete, the X and Y axes are homed simultaneously.

If there is a comma between the axes, the axes are homed separately and in the order specified. For example, Z,X,Y, where in this case the Z axis is homed first, and after the Z axis homing is complete, the X axis is homed, and finally, the Y axis is homed.
If there is no comma between the axes, the axes are homed simultaneously. For example, ZXY, where in this case the Z, X, and Y axes are homed simultaneously.

System\PulseDivider

This variable is used to divide the output pulses of the controller into specific numbers. It is used to create a proportion between the pulse rate of the controller and the motors. In other words, the pulse rate of the controller is 500 kilopulses, and there are motors that have a lower pulse rate, which can cause problems. For example, if the pulse rate of a servo motor is 200 kilopulses, the pulses should be divided by 4 so that the pulse rate of the controller becomes 125 kilopulses. Another example is when using stepper motors with a pulse rate of 100 kilopulses, the value of 8 should be selected for PulseDivider so that the stepper motor can read the pulse rate of the controller.

System\Analogs\Analog[n]\Enabeld

[n] refers to the number 1 and 2, which determine the output pins of the analog.
The Enabled variable is defined as True or False.
If this variable is False, the Analog[n] port is generally disabled and ignored.
If this variable is True, the Analog[n] port is enabled and can be used according to the specified link.

System\Analogs\Analogs[n]\Link

☐ InPort Table(2)	☐ OutPort Table(3)	☐ Function Table(4)	☑ Analog Table(5)	☐ Value Table(6)

System\Axes\Axes[n]\Acceleration

[n] is an identifier number between 1 to 6 that determines the axis number. Note that it is not allowed to have an inactive axis between active axes.
This variable is used to determine the axis movement acceleration in terms of units/s2. The value of this variable does not affect the acceleration of the program execution by ToolPath.
The appropriate value for this variable is determined by considering the stiffness, axis weight, and motor power. In general, it can be said that stiffness and axis weight have an inverse relationship, and motor power has a direct relationship with the value of this variable.
The value that can be defined for this variable is a number between 10 and 30,000.

System\Axes\Axes[n]\AlarmEnabled

n] refers to a number between 1 and 6 that determines the axis number. Note that a disabled axis cannot be present among the active axes.
If the AlarmEnabled variable is active in the System\AlarmEnabled address in this table, this variable will be taken into account.
The AlarmEnabled variable is defined as True or False.

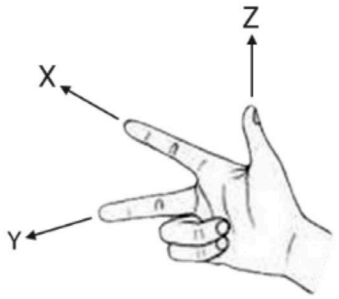
If this variable is False, the AlarmEnabled of the corresponding axis is disabled and ignored.
If this variable is True, the AlarmEnabled of the corresponding axis is enabled, and if the motor alarm part of the axis is correctly wired (according to section two), if the axis has an alarm, it will be displayed. Also, if wiring of the alarm section is incorrect or not wired, the alarm of the corresponding axis will turn on again.
Note that in stepper motors, since there is no alarm output, the AlarmEnabled variable must be set to False.

System\Axes\Axes[n]\Backlash

[n] is a number between 1 and 6, which determines the axis number. Note that there cannot be any inactive axes among the active ones.
If the axis has mechanical backlash, this variable can be used to approximately neutralize its effect. This value should be the average backlash over the length of the axis, in terms of the unit used for that axis.
The value that can be defined for this variable is a number between 0 and 1.

System\Axes\Axes[n]\Direction

[n] is a number between 1 and 6 which specifies the axis number. Note that there should not be any inactive axes among the active ones.
This variable is used to determine the direction of axis movement. It is better for axis movement directions to comply with international standards. The right-hand rule can be used to determine the correct direction of axes.



The value that can be defined for this variable is Negative or Positive.

System\Axes\Axes[n]\Enabled

[n] is a number between 1 and 6 that identifies the axis number. Note that it is not allowed to have a disabled axis among the active ones.
The variable Enabled is defined as True or False. If it is False, the axis will generally be disabled and ignored. If it is True, the axis will be enabled.

Note that the axes must be enabled in order. For example, you cannot enable axes 1 and 3 while disabling axis 2; instead, you must enable axes 1 and 2 and disable axis 3.

System\Axes\Axes[n]\HomeDetectVelocity

[n] is a number between 1 and 6 that identifies the axis number. Note that it is not allowed to have a disabled axis among the active ones.
This variable is used to determine the speed of the axis after the home sensor is detected. The value of this variable should not be too high to give the sensor enough time to accurately determine the home point. After the home sensor is detected, the axis moves in the opposite direction of the home sensor until the sensor is released.

System\Axes\Axes[n]\HomeDirection

[n] is a number between 1 and 6 that identifies the axis number. Note that it is not allowed to have a disabled axis among the active ones.
This variable is used to determine the direction of axis movement towards the home sensor. Therefore, using this variable, the location of the home sensor can be defined.

The value that can be defined for this variable is Negative or Positive.

System\Axes\Axes[n]\HomeDisplace

[n] is a number between 1 and 6 that identifies the axis number. Note that it is not allowed to have a disabled axis among the active ones.
This variable determines the displacement of the axis home point after homing.

System\Axes\Axes[n]\HomeFastVelocity

[n] is a number between 1 and 6 that determines the axis number. Note that among the active axes, the presence of an inactive axis is not allowed.
If high speed is required when homing, the speed value is determined by this variable.

Please note that adding a sensor is required to execute this variable. In fact, the first home sensor is for Home-FastVelocity and the second home sensor is for HomeVelocity.

System\Axes\Axes[n]\HomeLocation

[n] is a number between 1 and 6 that determines the axis number. Note that among the active axes, the presence of an inactive axis is not allowed.
This variable is used to determine the coordinates of the home point after the home sensor is activated. In other words, if the position of the home point is something other than MinCourse or MaxCourse, the numerical value of the home point can be determined by this variable.

System\Axes\Axes[n]\HomeVelocity

[n] is a number between 1 and 6 that determines the axis number. Note that among the active axes, the presence of an inactive axis is not allowed.
The homing speed can be determined by this variable.
109

System\Axes\Axes[n]\MaxCourse

[n] is a number between 1 and 6 that determines the axis number. Note that among the active axes, the presence of an inactive axis is not allowed.
This variable is used to determine the maximum course that the axis can have. In other words, after homing, this value will be the upper limit of the axis and the axis will not go beyond this limit.

The value that can be defined for this variable is a number between -109 and 109.

System\Axes\Axes[n]\MaxCourseGrid

[n] is a number between 1 and 6 that determines the axis number. Note that among the active axes, the presence of an inactive axis is not allowed.
This variable is used to determine the display of the maximum course of guide lines in ToolPath before homing or when the device does not need to be homed. In other words, the upper limit of the guide line display is determined by this variable before homing or when the device does not need to be homed.

The value that can be defined for this variable is a number between -109 and 109.

System\Axes\Axes[n]\MaxVelocity

[n] is a number between 1 and 6 that determines the axis number. Note that among the active axes, the presence of an inactive axis is not allowed.
This variable determines the maximum allowed speed for axis movement. This value affects the speed of the axis under all conditions.

If no value is considered for this variable, the maximum possible speed for this axis will be considered.

System\Axes\Axes[n]\MinCourse

[n] is a number between 1 and 6 that determines the axis number. Note that among the active axes, the presence of an inactive axis is not allowed.
This variable is used to determine the minimum course that the axis can have. In other words, after homing, this value will be the lower limit of the axis and the axis will not go below this limit.

The value that can be defined for this variable is a number between -109 and 109.

System\Axes\Axes[n]\MinCourseGrid

[n] is a number between 1 and 6 that specifies the axis number. Note that there is no permissible inactive axis among the active axes.

This variable is used to determine the minimum display range of guide lines in ToolPath before homing or in cases where the device does not need to be homed. In other words, this variable determines the lower limit of the guide line display range before homing or in cases where the device does not need to be homed.

The value that can be defined for this variable is a number between -109 and 109

System\Axes\Axes[n]\Name

[n] is a number between 1 and 6 that specifies the axis number. Note that there is no permissible inactive axis among the active axes.
This variable determines the name of the axis.
Note that the axis name can be one of the letters X, Y, Z, U, V, W, A, B, and C.

System\Axes\Axes[n]\Rotary

[n] is a number between 1 and 6 that specifies the axis number. Note that there is no permissible inactive axis among the active axes.
The Rotary variable is defined as True or False.
If this variable is False, the movement of the axis is considered to be linear.
If this variable is True, the movement of the axis is considered to be rotary.
Note that if a rotary motion axis is selected, the movement of the axes is not limited by MinCourse and Max-Course, and in displacements, values outside this range are transferred to within the range while maintaining the position.

System\Axes\Axes[n]\RundTrip

[n] is a number between 1 and 6 that specifies the axis number. Note that there is no permissible inactive axis among the active axes.
The RundTrip variable is defined as True or False.
If this variable is False, RundTrip is generally disabled and ignored.
If this variable is True, when moving, the axis is first moved to the highest defined point, then the other axes are moved to the specified position, and finally the axis is moved to the specified position.
For example, in router devices, if the RundTrip variable is True in Axes3, when moving the axes, the Z-axis is first moved to the highest point, then the X and Y axes are moved to the specified position, and finally the Z-axis is moved to the specified position.

System\Axes\Axes[n]\SameAs

[n] is a number identifier between 1 and 6, determining the axis number. Note that inactive axes are not allowed among active axes.
This variable can be used to connect the pulse output of an axis to a specific axis and create identical axes. The value of this variable is equal to the name or number of the axis to which this axis is to be connected.

If movement in the opposite direction of the desired axis is necessary, a negative sign is used before the name and number of that axis.
It is usually used when there is more than one motor in an axis.
Note that the axis for which the SameAs parameter is defined must be the last axis or Axis.

System\Axes\Axes[n]\Step

[n] is a number identifier between 1 and 6, determining the axis number. Note that inactive axes are not allowed among active axes.
This variable indicates the movement value of the axis per pulse in the desired unit of measure. In fact, the adaptation of axis movement sizes to movement sizes based on one of the standard units is done by this variable, which is also called calibration. For example, it can be determined that 1 unit of movement of the desired axis is equal to 1 millimeter, 1 centimeter, 1 inch, etc.
This variable is obtained either through measurement and proportion or by calculation. Calculation is certainly a more accurate and reliable method that requires information about the drives and motion transfer system.
A software called Radonix Calibrator is provided for measuring this variable, which easily provides calibration. (Please refer to Chapter 5 for more information.)

System\Axes\Axes[n]\Unit

[n] is a number between 1 and 6 which determines the axis number. Note that inactive axes are not allowed among active axes.
This variable is a string used to display the measurement unit. This variable does not affect internal interface calculations and is only used for display purposes.

System\InPorts\InPort[n]\Enabled

[n] is the identifier for the pin number of the digital inputs of the controller. The variable “Enabled” is defined as True or False. If this variable is False, the input port in question is generally deactivated and ignored. If this variable is True, the input port in question is activated and can be used according to the designated link.

System\InPorts\InPort[nn]\Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

System\InPorts\InPort[n]\NC

[n] refers to the pin number of the digital inputs/outputs of the controller.

The variable “Enabled” is defined as either True or False.
If this variable is False or undefined, the input port is considered normally open.
If this variable is True, the input port is considered normally closed.

System\OutPorts\OutPort[n]\Enabled

[n] refers to the pin number of the digital outputs of the controller.

The variable “Enabled” is defined as either True or False.
If this variable is False, the output port is generally disabled and ignored.
If this variable is True, the output port is activated and can be used according to the specified link.

System\OutPorts\OutPort[n]\Link

☐ InPort Table(2)	☐ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

System\OutPorts\OutPort[n]\NC

[n] refers to the pin number of digital input/output ports of the controller.
The variable “Enabled” is defined as True or False. If this variable is False or undefined, the input/output port is considered as Normally Open. If this variable is True, the input/output port is considered as Normally Closed.

System\OutPorts\OutPort[n]\NC

[n] refers to the pin number of digital input/output ports of the controller.
The variable “Enabled” is defined as True or False. If this variable is False or undefined, the input/output port is considered as Normally Open. If this variable is True, the input/output port is considered as Normally Closed.

System\ToolPath\Acceleration

This variable is used to determine the acceleration of the axes simultaneously during program execution in units of unit/s. The appropriate value for this variable is determined taking into account the inertia, axis weight, and motor power. This value can be calculated mechanically and approximately. Generally, it can be said that there is an inverse relationship between inertia and axis weight and a direct relationship between motor power and the value of this variable.

The value that can be defined for this variable is a number between 10 and 30000.

System\ToolPath\Axes

This variable determines the axes used in ToolPath that have simultaneous movement (Interpolate).

Note that the names of the axes must be separated from each other by commas (,).

System\ToolPath\FeedVelocity

When the Feed or F value is not defined in the GCode, the value specified in this variable is used as the basis for the movement speed.

System\ToolPath\FeedVector

If this variable has no value, all axes in the ToolPath with Interpolate motion will calculate in the movement speed. The axes listed in this variable are used in calculating the movement speed for Interpolate motion.

System\ToolPath\Focus

The Focus variable is defined as True or False.
If this variable is False or has no value, the display is centered on the desktop and the layout is displayed on the entire screen when the executable file is opened.
If this variable is True, the display is centered on the layout and the layout is displayed on the entire screen when the executable file is opened.

System\ToolPath\JunctionFactor

Software programs that convert motion paths into G-code usually simulate curved paths using line segments. There are angles between these lines, and based on these angles, the length of the lines, and the motion acceleration defined for the ToolPath, the maximum speed allowed for passing these intersections is calculated. This variable is actually a coefficient that is multiplied by the calculated maximum speed.

Values greater than 1 are used in cases where the motion acceleration is low or the device is capable of passing intersections at a higher speed.
The value that can be defined for this variable is a number between 1 and 100.

System\ToolPath\LastView

There are three general structures for displaying LastView, as follows:

Default	Displays the ToolPath view at the top of the screen when the program is opened. Read_ Data
Read_Data	Displays the ToolPath view that was previously saved using the Ctrl+Shift+Alt+V shortcut when the program is opened
Read_Write_ Data	Displays the last view that existed in the program when it was last opened.

System\ToolPath\MinLineLength

When this parameter is zero, the averages of the steps are merged and a vector is obtained. In other words, if this parameter is zero, 10 times the merging of steps is considered as the smallest linear value that should be accepted, and by doing so, smaller lines than that value are ignored and connected to the next line. For example, if the steps are 1 millimeter and this parameter is zero, 10 times the steps are considered as the smallest value, meaning lines smaller than 0.1 millimeters are connected to the next line, while lines 0.1 millimeters or larger are executed without any change.

System\ToolPath\Smoothness

This variable is used to create a smoother and more uniform motion. In fact, the velocity curve changes from linear to an S-curve. As the value of this variable increases, the curvature of the S-curve also increases.

If the speed or acceleration of the device is high, it is better not to set the value of this variable too high. The value that can be defined for this variable is a number between 1 and 250 milliseconds.

System\ToolPath\TraverseAcceleration

This variable is used to determine the axis displacement acceleration in units of unit/s without running the program. The appropriate value for this variable is determined based on the stiffness, axis weight, and motor power. This value can be mechanically and approximately calculated. Generally, it can be said that stiffness and axis weight have an inverse relationship, and motor power has a direct relationship with the value of this variable.

The value that can be defined for this variable is a number between 10 and 30000.

System\ToolPath\TraverseVelocity

This variable is used to determine the axis displacement speed without running the program, as well as during displacement using G0.

System\ToolPath\View

This variable determines the axes that are visible in the ToolPath. Only the combination of the display in the interface is determined in this variable, and the names X, Y, Z, C, A, and B must be used in these combinations.

The axes used in the ToolPath are displayed in the order of the axes written in this variable.
The number of axes in this variable should not exceed the number of axes used in the ToolPath.

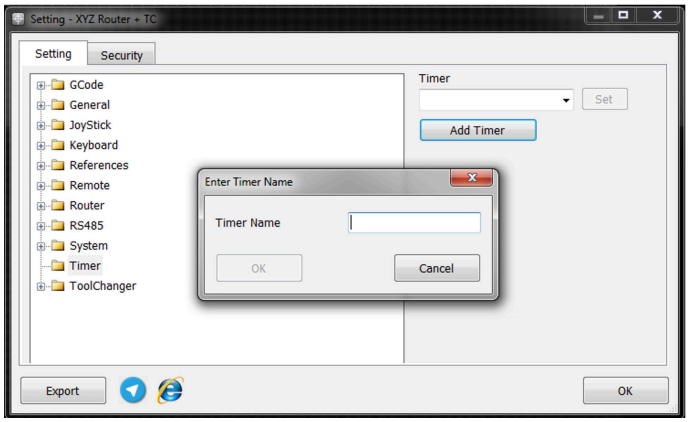
Three general structures, along with the axis order in the interface, have been considered for displaying ToolPath, as follows:

2-axis structure	In this structure, the first axis is from left to right on the screen, and the second axis is from bottom to top. For example, the value X, Y means displaying the plane and normal of two axes X and Y, where X is displayed from left to right and Y is displayed from bottom to top, and the value Y, X is the reverse of the previous state. Also, X, Z is a two-axis planar state with X from left to right and Z from bottom to top.
3-axis structure	This structure provides users with a variety of combinations for display. These combinations include Cartesian combinations such as X, Y, Z and their displacement, as well as combinations for displaying the tangent axis to the motion path, such as X, Y, C, and possible displacement for the first two axes. The first axis is from left to right, the second axis is from bottom to top, and the third axis is displayed tangent to the path if the C axis is selected, otherwise it is displayed outward from the depth of the monitor.
4-axis structure:	This structure includes combinations of X, Y, Z, A axes, X, Y, Z, B axes, and combinations of the displacement of the first three axes. These combinations are used to display four-axis devices that have a rotary axis. If the rotary axis is in the direction of the first axis, combinations of X, Y, Z, A axes are used, and if the rotary axis is in the direction of the second axis, combinations of X, Y, Z, B axes are used.

Table(16)

Subcategory Timer

Timer



According to the steps shown in the above figure, a new timer can be added. In other words, in step 1, the left-click of the mouse is pressed on the Timer branch, and then in step 2, by pressing Add Timer, a window called Enter Timer Name opens, which in step 3, the desired name can be selected for the new timer, for example, by writing T1 and pressing the OK button (according to step 4), the new timer is added to the Timer sub-branches.

Timer\[n]\Automatic

The variable “Automatic” is defined as True or False. If this variable is False, the timer command is issued using the RunTimer and StopTimer functions in the function link (Table No. 4), and after executing the timer command, the timer is executed according to the link specified in the Link section. If this variable is True, the timer is automatically executed during program execution according to the specified link.
The RunTimer and StopTimer functions are fully explained in the function link table (Table No. 4). Please refer to Table No. 4 for more information.

Timer\[n]\Continuouse

The variable “Continuous” is defined as True or False.
If this variable is False, the timer is executed only once.
If this variable is True, the timer is repeated multiple times as long as the condition specified in the ControlStatusCondition variable is met.

Timer\[n]\ControlStatusCondition

There is a variable called ControlStatusCondition which is used to determine the condition for executing the timer. By entering one of the ControlStatus arguments (parameter link) in the ControlStatusCondition variable, it can be determined under what conditions the timer should be executed. For example, if ControlStatusCondition is set to 1, the timer will only run when the software is online.
The ControlStatus parameter is fully explained in the parameter link table. For more information, please refer to table (6).

Note that only one argument or condition can be defined for the ControlStatusCondition variable.

Timer\[n]\Duration

The variable “Duration” is used to determine the length of time for the timer in milliseconds. In other words, after issuing the command to run the timer (either automatically or manually), the timer will run for the time set in the “Interval” variable and then for the time set in the “Duration” variable. For example, if the “Duration” variable is set to 5000, after issuing the command to run the timer and after the time set in the “Interval” variable has passed, the timer will run for 5 seconds and then turn off.

Timer\[n]\Enabled

The variable “Enabled” is defined as True and False.
If it is False, the timer is generally disabled and ignored.
If it is True, the timer is enabled and can be used based on the defined variables and links.

Timer\[n]\Interval

This variable is used to determine the start time of the timer in milliseconds. In other words, after issuing the command to run the timer (either automatically or manually), the timer will run after the time set in the Interval parameter has elapsed. For example, if Interval = 60000, the timer will run after 60 seconds have passed following the issuance of the command to run the timer.

If the Continuouse variable in the Timer[n] address in the same table is active, this process will be repeated.

Timer\[n]\Link

☐ InPort Table(2)	☑ OutPort Table(3)	☑ Function Table(4)	☐ Analog Table(5)	☐ Value Table(6)

Table(17)

ToolChanger

ToolChanger\AutoSetToolHeight

The variable “AutoSetToolHeight” is defined as True or False. If this variable is False, AutoSetToolHeight is generally disabled and ignored. If it is True, AutoSetToolHeight is activated, and when the tool is changed, the tool height is first registered using a height detection sensor, and then the tool is changed. This process is performed every time the tool change command is issued.

ToolChanger\ClampDepth

This variable is used to determine the height of the tool holder’s collet. In fact, if the tool holder’s collet is not of the pressure type and the tool is placed in the collet, the value of the collet’s height is determined by this variable.

The value that can be defined for this variable is a number between 0 to 100 millimeters.

ToolChanger\CurrentTool

This variable stores the current tool number.

ToolChanger\DefaultTool

This variable is used to determine the active tool as default. In other words, if for structural reasons, a specific tool must be considered as the active tool when opening the program, this variable is used to specify the number of the active tool.
Note that this variable is specific to certain devices and usually has no value, meaning it should be empty.

ToolChanger\Enabled

The Enabled variable is defined as True or False.
If this variable is False or undefined, the ToolChanger or automatic tool change is generally disabled and ignored.
If this variable is True, the ToolChanger or automatic tool change is enabled.

ToolChanger\GoSafeLocation

The variable “GoSafeLocation” is defined as True or False. If this variable is False, “GoSafeLocation” is generally disabled and ignored. If this variable is True, “GoSafeLocation” is enabled. This variable is used when it is necessary to move the machine to a safe coordinate before changing the tool. After moving to the safe point, the tool change will be executed. This variable is usually used for machines that have a parking cover.

ToolChanger\Jack3Delay

In devices that use a jack for tool changing, this variable is defined. In fact, if the start, end, or both ends of the jack have not been used with the end-of-path detection sensor input T-Jack3SensorPin, this variable is used to create the required delay to complete the jack movement. In other words, if the input T-Jack3SensorPin has not been used, the Jack3Delay variable is activated, which can determine the delay time for turning on and off the T-Jack3Pin output.
Note that this variable is used when the end-of-path detection sensor and input T-Jack3SensorPin are not used in the device because the sensor takes priority over the Jack3Delay variable if it exists.
This variable is related to the input T-Jack3SensorPin and output T-Jack3Pin. For more information, refer to tables 2 and 3.

ToolChanger\JackDelay

In devices that use a jack for tool changing, this variable is defined. In fact, if the start, end, or both ends of the jack have not been used with the end-of-path detection sensor input T-JackSensorPin, this variable is used to create the required delay to complete the jack movement. In other words, if the input T-JackSensorPin has not been used, the JackDelay variable is activated, which can determine the delay time for turning on and off the T-JackPin output.

Note that this variable is used when the end-of-path detection sensor and input T-JackSensorPin are not used in the device because the sensor takes priority over the JackDelay variable if it exists.
This variable is related to the input T-JackSensorPin and output T-JackPin. For more information, refer to tables 2 and 3.

ToolChanger\RotationSensorPeriod

This variable is used to detect spindle rotation stoppage. In fact, a time period for checking spindle rotation is determined by this variable. For example, if the value of this variable is 1000, if no signal is sent from the spindle rotation sensor input for one second, it is considered as spindle rotation stoppage.

ToolChanger\SpindleOffDelay

This variable is used to determine the delay time until the spindle stops during a tool change. If this variable does not have a value, its value is taken from the SpindleDelay variable in the Router branch.
Note that if the spindle has one of the sensors (T-HSD-S3Pin, T-CC-S3Pin, T-TEKNO-S3Pin, T-SpindleRotationSensorPin, T-InverterStopPin), this variable will be ignored

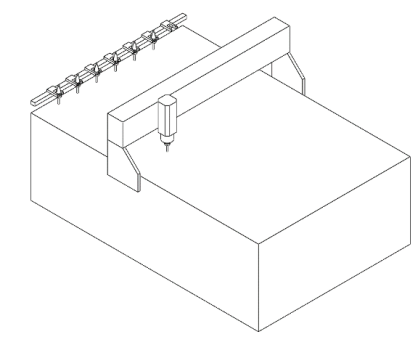
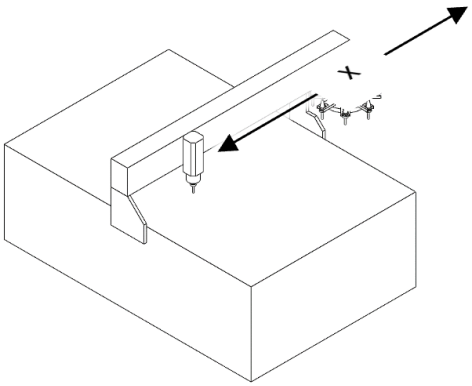
ToolChanger\ToolHolderDelay

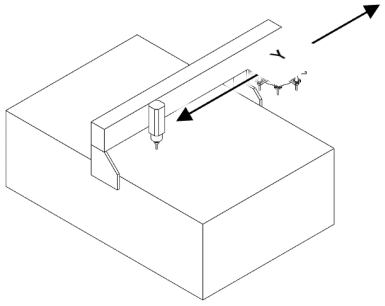
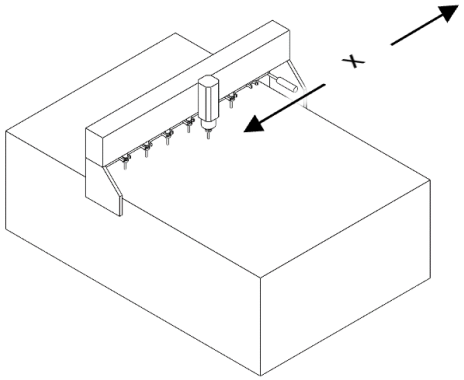
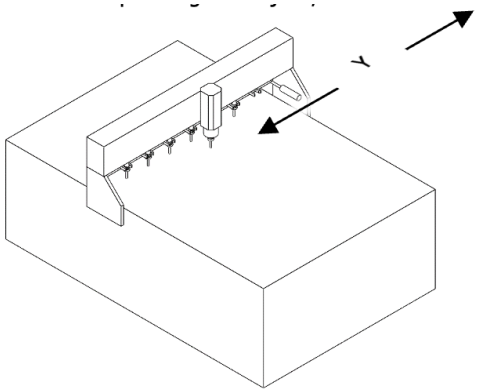
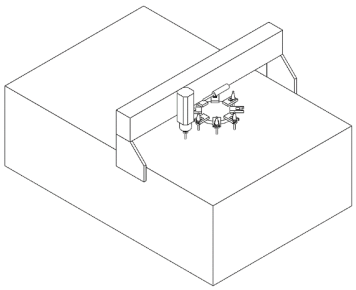
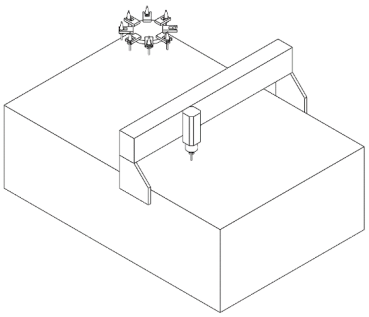
A delay is set to allow the internal jack of the spindle enough time to grip or release the tool. This time is determined by the ToolHolderDelay variable.

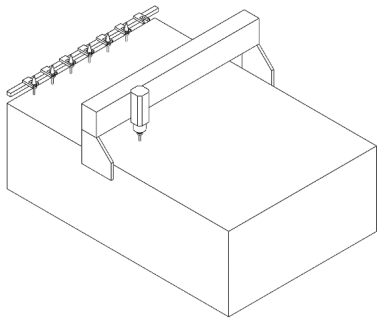
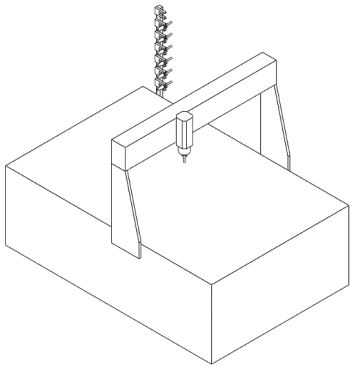
ToolChanger\ToolsInCourse

The ToolsInCourse variable is defined as True or False.
If this variable is False, ToolsInCourse is disabled.
If this variable is True, ToolsInCourse is enabled and the operator can exit after taking Home and determining the maximum and minimum courses during the Home operation. This is used when the machine parking is outside the Home coordinates and table limits, and the user wants to obtain the coordinates of each tool for automatic tool change.
Note that after the settings are done, this variable should be set to False.

ToolChanger\Type

XYZ_YZ	Reserve	
XYZ	Tools are held linearly on a part of the table in the X or Y axis using clamps.	
XCZ	Tools are held rotationally on the X-axis using clamps that move along the X-axis.	

YCZ	This is the most common rotational parking where tools are held rotationally on the Y-axis using clamps that move along the Y-axis.	
XZ	Tools are held linearly on the X-axis using clamps and are inserted or removed from the clamps using an air jack, which moves along the X-axis.	
YZ	Tools are held linearly on the Y-axis using clamps and are inserted or removed from the clamps using an air jack, which moves along the Y-axis.	
CZ	Tools are held rotationally on the Z-axis using clamps and are inserted or removed from the clamps using an air jack, which moves along the Z-axis.	
XYZB	Reserve	
XYZC	Tools are held rotationally on the table using clamps.	

XYZ_All	This is used for machines that have more than three axes and the remaining axes must be moved to the zero coordinates for tool changing. Tools are held linearly on the table using clamps.	
XYZAB	Tools are held linearly on the table using clamps. In this case, the tools can have an angle relative to the surface of the table.	
XYZA_MH	Reserve	
XYZB_MH	Reserve	
XZB_Vertical	Reserve	

ToolChanger\Velocity

This variable is used to determine the speed of tool changing. In fact, the entry and exit speed of the parking area can be specified.

ToolChanger\Displacement\X

The displacement value of the X-axis is determined by these variables for extracting the tool from the clamp. The reverse process is also done when releasing the tool inside the clamp.
Note that the value entered in this variable should be relative to the parking area. ToolChanger\Displacement\Y

ToolChanger\Displacement\Y

The displacement value of the Y-axis is determined by these variables for extracting the tool from the clamp. The reverse process is also done when releasing the tool inside the clamp.
Note that the value entered in this variable should be relative to the parking area.

ToolChanger\Displacement\Z

After releasing the tool inside the clamp, the Z-axis rises by the value set for this variable and then moves towards the next tool. The reverse process is also done when grabbing the tool inside the clamp.
The value of spindle rise should be such that there is a logical distance from the other tools.

ToolChanger\SafeLocation\X

If the GoSafeLocation variable is True, a sub-branch called SafeLocation is added to the ToolChanger branch. When using this variable, before changing the tool, the device is moved to the point specified by this variable along the X axis and after moving to the safe point, the tool change is performed.
Note that this variable is defined with respect to the home point of the device.

ToolChanger\Displacement\Z

After releasing the tool inside the clamp, the Z-axis rises by the value set for this variable and then moves towards the next tool. The reverse process is also done when grabbing the tool inside the clamp.
The value of spindle rise should be such that there is a logical distance from the other tools.

ToolChanger\SafeLocation\X

If the GoSafeLocation variable is True, a sub-branch called SafeLocation is added to the ToolChanger branch. When using this variable, before changing the tool, the device is moved to the point specified by this variable along the X axis and after moving to the safe point, the tool change is performed.
Note that this variable is defined with respect to the home point of the device.

ToolChanger\SafeLocation\Y

ToolChanger\SafeLocation\Y If the GoSafeLocation variable is True, a sub-branch called SafeLocation is added to the ToolChanger branch. When using this variable, before changing the tool, the device is moved to the point specified by this variable along the Y axis and after moving to the safe point, the tool change is performed.

Note that this variable is defined with respect to the home point of the device.

ToolChanger\ToolHeightSensor\Velocity

This variable is used to determine the speed of movement towards the tool height sensor.
The device moves at the speed set in the TraverseVelocity variable (located in the System branch and ToolPath sub-branch) towards the coordinates of the tool height sensor. In fact, at the beginning of the tool height measurement process, the device is moved at the TraverseVelocity speed to the height specified by the Z variable in the ToolChanger branch and ToolHeightSensor sub-branch and then moves at the speed specified in this variable. A lower value of this speed increases accuracy but also increases measurement time.

ToolChanger\ToolHeightSensor\X

This variable determines the coordinates of the installation location of the height measurement sensor along the X axis.

ToolChanger\ToolHeightSensor\Y

This variable determines the coordinates of the installation location of the height measurement sensor along the Y axis.

ToolChanger\ToolHeightSensor\Z

This variable determines the starting coordinates of movement towards the sensor along the Z axis, after which the Z axis moves at the speed specified in the Velocity parameter at the ToolChanger\ToolHeightSensor address.

ToolChanger\Tools\ToolReference

The coordinates on which the tool height sensor measures the tool height.
If the tool height measurement process is performed with the empty collet and tool number zero, this variable is automatically set and in this case, the tool height is the actual height from the collet surface.
The default value of this variable is zero and if the height sensor is used, there is no need to set this variable.

ToolChanger\Tools\Tool[n]\Enabled

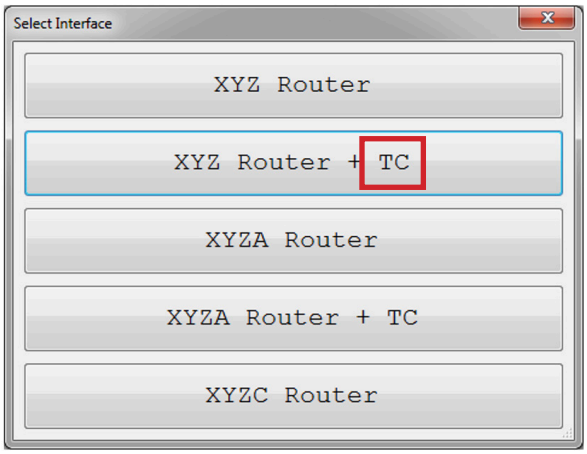
n refers to the tool number, which is a number between 1 and 64.
The variable Enabled is defined as True or False.
If this variable is False, the tool is disabled and ignored in general.
If this variable is True, the tool is enabled and can be used in automatic tool change.

ToolChanger\Tools\Tool[n]\FadeCoefficient
n refers to the tool number, which is a number between 1 and 64. This variable is used to determine the amount of tool wear per meter of travel. For example, if FadeCoefficient = 0.1, then for every meter of travel, 0.1 millimeters of wear is considered for the length of the tool. Note that only some router interfaces have this feature.
ToolChanger\Tools\Tool[n]\Head
n refers to the tool number, which is a number between 1 and 64. The variable Head is defined as True or False. If this variable is False, the tool is not considered as a head tool. If this variable is True, the tool is considered as a head tool. In such tools, the tool is usually activated by a pneumatic jack or a similar mechanism.
ToolChanger\Tools\Tool[n]\Height
n refers to the tool number, which is a number between 1 and 64. This variable is used to determine the desired height of the tool. Note that it is not necessary for this height to be the actual height of the tool, but rather the relative height of the desired tool. The value of this variable is obtained either automatically using the SetToolHeight method or manually by measuring the distance between the tool tip and a fixed point.
ToolChanger\Tools\Tool[n]\X
n refers to the tool number, which is a number between 1 and 64. The behavior of this parameter is different in two different modes: 1- If the Head variable is False, the tool change is performed by the spindle, and this variable is the coordinates of the tool holding location in the clamp along the X-axis. 2- If the Head variable is True, the tool is separate from the spindle. In this case, the coordinates of the tool offset or the tool offset relative to the reference tool along the X-axis are taken into account.
ToolChanger\Tools\Tool[n]\Y
n refers to the number of the tool, which is a number between 1 and 64. The function of this parameter varies in two different modes: 1- If the Head variable is False, the tool is actually exchanged by the spindle and this variable represents the coordinates of the tool holding location in the clamp along the Y axis. 2- If the Head variable is True, the tool is actually a separate head. In this case, the Y coordinate represents the offset of the tool relative to the reference tool or in other words, the tool offset relative to the reference tool along the Y axis.
ToolChanger\Tools\Tool[n]\Z
n refers to the tool number, which is a number between 1 and 64. The function of this parameter varies in 2 different ways: 1- If the Head variable is False, the tool change is actually done by the spindle, and this variable represents the coordinates of the tool holder location in the Z-axis direction. 2- If the Head variable is True, the tool is separate and this variable is ignored.

Table(18)

Automatic Tool Changing (Tool Changer)

In this section, we will explain automatic tool changers or Tool Changers and how to set them up. First, based on the types of axes in your machine, an appropriate interface that supports automatic tool changing should be installed. The interface that supports automatic tool changing usually has the abbreviation TC at the end of its name. (Figure 23)



Figure(23)

After installing the interface and opening the Radonics software, the necessary settings for using the Tool Changer must be applied. To do this, first, the Settings window should be opened and the ToolChanger branch, where all the settings related to the Tool Changer exist, should be accessed. In this section, we will explain the essential settings for setting up the Tool Changer.

- Note that in the table related to the ToolChanger section (Table 18) and also in the tables related to different types of links, all settings and features related to automatic tool changing in the controller are fully explained, and the user can use other Tool Changer features according to their needs.

In the next step, based on the variety of tool changing methods (which are determined based on the type of tool parking on the machine), there are different tool changing options in the Radonics software that the user should choose according to the tool changing method available on their machine. To do this, go to the ToolChanger branch and the Type variable to determine the type of tool changer on the machine (for more information on the types of tool changing methods, please refer to table number 18 in the ToolChanger branch).

After determining the type of tool changer, we need to set the necessary digital inputs and outputs to issue the essential commands for tool changing. To do this, first go to the System branch and the InPorts sub-branch, and select the desired input. Now, by opening one of the desired inputs, you will encounter three variables for the settings on each of the inputs, which are Enabled, Link, and NC, respectively (for more information on how each of the 3 digital input variables works, please refer to the table in the System branch under the InPorts sub-branch). To perform the tool changing operation, you need 5 inputs, and you should set the links T-ToolHolder,1 (Function link in Table 4), and T-ToolHeightSensorPin (Input link in Table 2), T-HSD-S1Pin or (T-ToolSensorPin, T-TEKNO-S2Pin, T-CC-S1Pin) (Input link in Table 2), T-HSD-S2Pin or (T-CC-S2Pin, T-TEKNO-S1Pin, T-ToolHolderSensorPin) (Input link in Table 2), and T-HSD-S3Pin or (T-CC-S3Pin, T-TEKNO-S3Pin, T-SpindleRotationSensorPin, T-InverterStopPin) (Input link in Table 2) for them (for more information, please refer to Tables 2 and 4). After that, go to the System branch and the OutPorts sub-branch and select the desired output. Now, by opening one of the desired outputs, you will encounter three variables for setting each of the outputs, which are Enabled, Link, and NC, respectively (for more information on how each of the digital output variables works, please refer to the table in the System branch under the OutPorts sub-branch). To perform the tool changing operation, you need 2 outputs, and you should define the links

T-ToolHolderPin and T-ToolCleanerPin for them (for more information, please refer to Table 3).

- Note that the initial settings for setting up the tool changer in the input and output sections have been done by default.

After setting up the inputs and outputs, go back to the ToolChanger branch and adjust the remaining essential variables for the tool changing process. (For more information, refer to the ToolChanger table (Table 18) in the ToolChanger branch.)

After setting the ToolChanger variables, the coordinates of all the tool parking locations must be determined under the Tools sub-branch in the ToolChanger branch. To do this, first home the machine and then place each tool with its ISO in the desired parking number.

- ISO is one of the taper standards widely used in router machines in various sizes such as ISO30, ISO40, and ISO50 according to the specific requirements.

Now, by manually moving or jogging the spindle, it is moved to the position of each tool in the parking area. At this stage, it must be ensured that the gripper is open when the ISO is placed on the spindle (the gripper being open depends on the output that was previously set using the T-ToolHolderPin link (output link in table 3)). A momentary switch connected to the T-ToolHolder,1 digital input link can be used to issue the command, or a virtual button can be added to the software with the same T-ToolHolder,1 link to toggle the output. Now, with great precision, the spindle is brought up to the top of each tool so that the ISO, along with the tool, is fully seated in the tool holder clamps. After that, on the main screen of the software, the Absolute coordinate values of each axis, which indicate the position relative to the home point, are entered separately for the coordinates of each parking space, and all parking spaces must be defined in the same way (Figure 24).

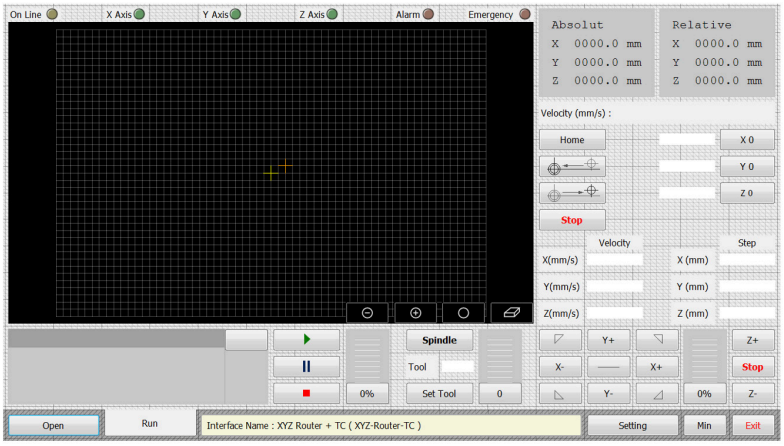
Figure(24)

In the final step, the tool height measurement, which is one of the most essential steps in the tool changing process, is performed. In this step, first make sure that the input with the T-ToolHeightSensorPin link (input link in table 2) is defined, and then enter the tool height sensor coordinates in the ToolChanger branch and ToolHeightSensor sub-branch in order of axes (for more information, please refer to the ToolChanger table (table number 18) in the ToolChanger branch).

- Please note that in general applications, we set the ToolReference value to zero. In cases where automatic tool height measurement is not performed and the user wants to measure the tool height using a caliper, the ToolReference value should be set equal to the spindle collet offset value relative to the Z-axis zero point.
- Please note that tool height measurement should be performed after defining the parking positions.

Introduction to the Interface Editing Environment

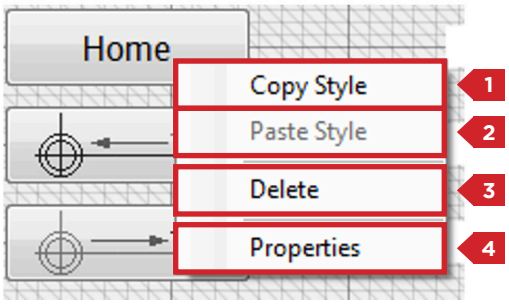
In this section, we will provide a comprehensive explanation of the interface editing page and all its elements. By simultaneously pressing the CTRL+SHIFT+ALT+Home shortcut key, the interface editing environment opens, which provides the user with the ability to customize and make visual changes to the interface. In fact, users can add or remove different elements in the interface according to their needs, as well as determine or change parameters and links of the elements. (Figure 25)



Figure(25)

Making Changes to Elements

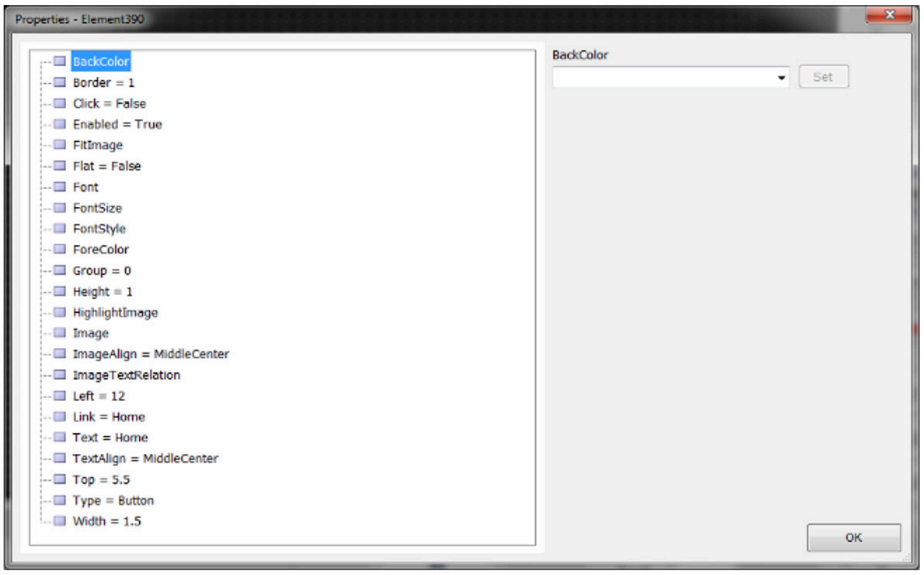
Right-clicking on any of the elements on the page opens a window where changes can be made to the selected element using the available parameters. Additionally, the style of the element can be copied, pasted, or deleted in this window. (Figure 26)



Figure(26)

- Note that the numbers in the table below are based on the numbering of elements in Figure 26.

	Name of the Element	Descriptions
1	Copy Style	In order to copy all the specifications related to the desired element, including size and the settings in the Properties section (please refer to row number 4 in this table), use the Copy Style item. Note that after copying the specifications related to the desired element, you can transfer the specifications to another element of the same type using the Paste Style item. For example, if the specifications related to a Button element have been copied, you can only transfer those specifications to another element of the same type (i.e. another Button).
2	Past Style	To transfer all the specifications related to the desired element, including size and the settings in the Properties section (please refer to row number 4 in this table) that have been copied using Copy Style, use the Paste Style item.

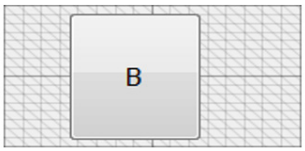
		Note that after copying the specifications related to the desired element, you can transfer the specifications to another element of the same type using the Paste Style item. For example, if the specifications related to a Button element have been copied, you can only transfer those specifications to another element of the same type (i.e. another Button).
3	Delete	Clears the desired element entirely.
4	Properties	By pressing the Properties item, a window opens that allows you to view or change the variables related to the desired element. Note that the number and types of variables in the window are different depending on the type of element. For example, the Properties window for a Button element is shown in the following figure:  All the variables in the Properties window are explained separately in table number 21.

Table(19)

Adding an Element

By right-clicking on the interface editing environment, a window shown in figure 27 will open at points where no element exists.

Figure (27)

<table>
<tr><th></th><th>Name of
the Element</th><th>Descriptions</th></tr>
<tr><td>1</td><td>Add Button</td><td>The appearance of the Button element on the interface editing page:
 

By right-clicking on the existing Button element on the page and then pressing the Properties item, a window opens where you can adjust the variables related to the Button element.
All the variables available in the Properties window are explained in table number 21.
Note that since the Link variable in the Properties window is different for each element, the table for it is written separately for each element.
<table><tr><th colspan="5">Link related to</th></tr><tr><td>☐ Input pins</td><td>☐ Output pins</td><td>☑ Function</td><td>☐ Analog</td><td>☐ Parameters</td></tr></table></td></tr>
<tr><td>2</td><td>Add Toggle</td><td>It is used for 2-state commands. In fact, by pressing the Toggle, the corresponding link is turned on if it is off, and turned off if it is already on. For example, the Toggle key is used to turn the spindle on and off.
The appearance of the Toggle element on the interface editing page:
 

By right-clicking on the existing Toggle element on the page and then pressing the Properties item, a window opens where you can adjust the variables related to the Toggle element.
All the variables available in the Properties window are explained in table number 21.
Note that since the Link variable in the Properties window is different for each element, the table for it is written separately for each element.
<table><tr><th colspan="5">Link related to</th></tr><tr><td>☐ Input pins</td><td>☐ Output pins</td><td>☑ Function</td><td>☐ Analog</td><td>☐ Parameters</td></tr></table></td></tr>
<tr><td>3</td><td>Add Check Box</td><td>It is used for 2-state commands. In fact, when the Check Box is checked, the output is turned on, and when it is unchecked, the output is turned off. For example, the Check Box can be used to turn vacuum on and off.
The appearance of the Check Box element on the interface editing page:
 

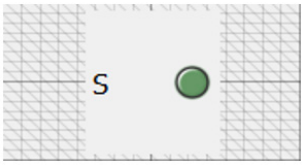
By right-clicking on the existing Check Box element on the page and then pressing the Properties item, a window opens where you can adjust the variables related to the Check Box element.
All the variables available in the Properties window are explained in table number 21.
Note that since the Link variable in the Properties window is different for each element, the table for it is written separately for each element.
<table><tr><th colspan="5">Link related to</th></tr><tr><td>☐ Input pins</td><td>☑ Output pins</td><td>☐ Function</td><td>☐ Analog</td><td>☐ Parameters</td></tr></table></td></tr>
</table>

<table>
<tr><td>4</td><td>Add Lable</td><td>
The Label element is used to display text, title, or the desired variable value to the user. For example, the speed of the machine can be displayed using this element.

Note that this element only allows you to view the desired variable or entered text, and it does not have the ability to edit or change the variable using this element.

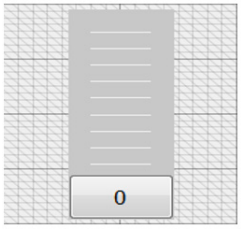
The appearance of the Label element on the interface editing page:


By right-clicking on the existing Label element on the page and then pressing the Properties item, a window opens where you can adjust the variables related to the Label element.
All the variables available in the Properties window are explained in table number 21.
Note that since the Link variable in the Properties window is different for each element, the table for it is written separately for each element.
<table>
<tr><th colspan="5">Link related to</th></tr>
<tr><td>☐ Input pins</td><td>☐ Output pins</td><td>☐ Function</td><td>☐ Analog</td><td>☑ Parameters</td></tr>
</table>
</td></tr>
<tr><td>5</td><td>Add Value</td><td>
The Value element is used to view and modify variables.
The Value element can be used to both view and modify a variable.


To access the Properties window and adjust the variables related to the Value element, right-click on the Value element on the interface editing page. All of the available variables in the Properties window are explained in Table 21.
Please note that the Link variable in the Properties window is different for each element, so the table is written separately for each element..
<table>
<tr><th colspan="5">Link related to</th></tr>
<tr><td>☐ Input pins</td><td>☐ Output pins</td><td>☐ Function</td><td>☐ Analog</td><td>☑ Parameters</td></tr>
</table>
</td></tr>
<tr><td>6</td><td>Add Signal</td><td>
The Signal element is used to display variables that include two states of True and False. This element comes in three different colors.
View of the Signal element in the interface editor:


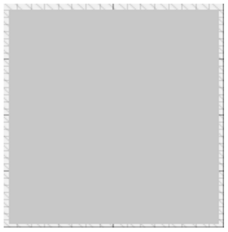
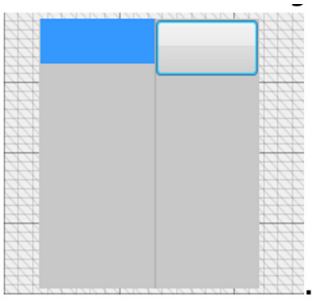
By right-clicking on the Signal element that exists on the page and then pressing the Properties item, a window opens in which the variables related to the Signal element can be adjusted. All the variables available in the Properties window are explained in table number 21.

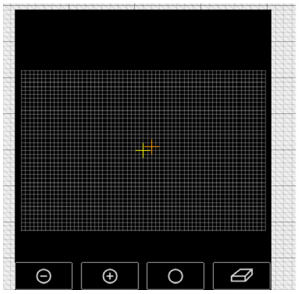
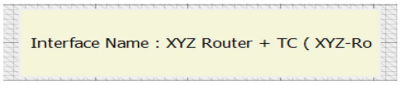
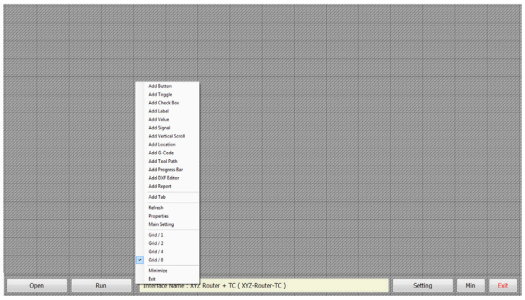
Note that, since the Link variable in the Properties window is different for each of the elements, the table is written separately for each element.
<table>
<tr><th colspan="5">Link related to</th></tr>
<tr><td>☐ Input pins</td><td>☐ Output pins</td><td>☐ Function</td><td>☐ Analog</td><td>☑ Parameters</td></tr>
</table>
</td></tr>
</table>

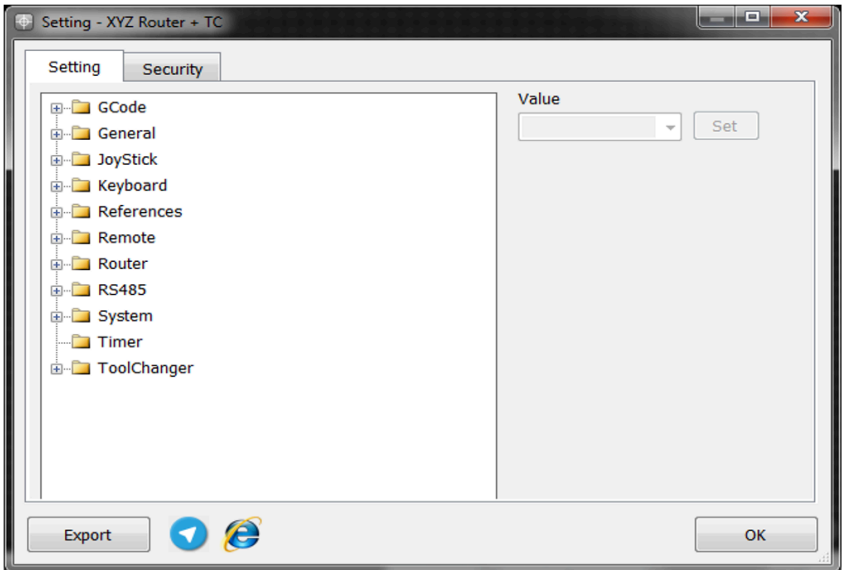
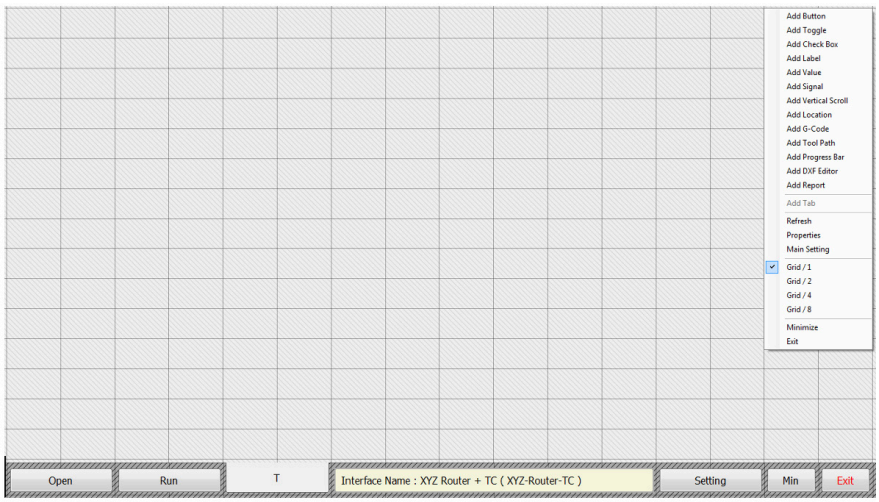
<table>
<tr><td>7</td><td>Add Vertical Scroll</td><td>It is used to change some variables. For example, Scroll has been used for Feed variables.
Appearance of Vertical Scroll element on the editing page of the interface:
 

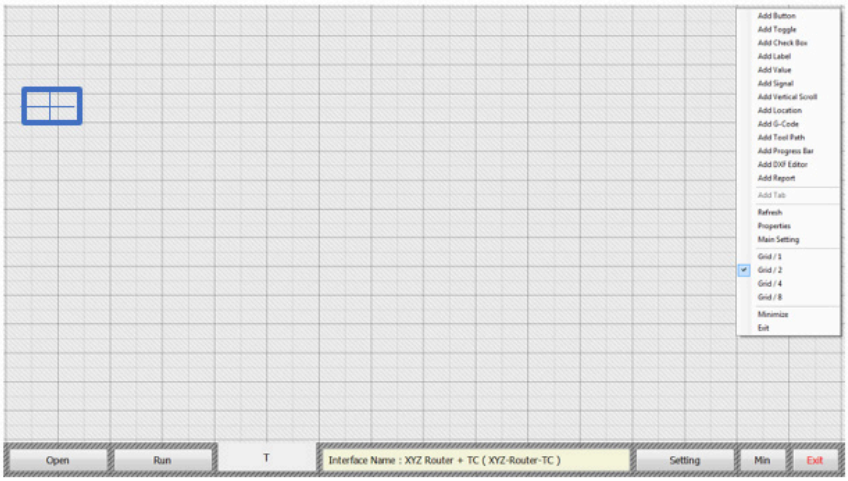
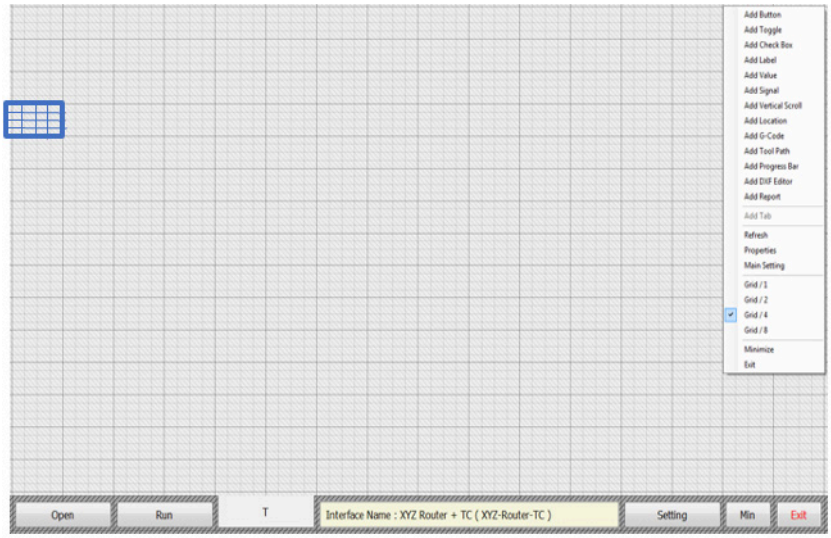
When right-clicking on the Vertical Scroll element on the page and selecting the Properties item, a window opens in which you can adjust the variables related to the Vertical Scroll element. All the variables available in the Properties window are explained in Table 21.

Please note that since the Link variable in the Properties window is different for each element, the table is written separately for each element.
<table><tr><td colspan="5">Link related to</td></tr><tr><td>☐ Input pins</td><td>☐ Output pins</td><td>☐ Function</td><td>☐ Analog</td><td>☑ Parameters</td></tr></table></td></tr>
<tr><td>8</td><td>Add Location</td><td>It is used to display the axis coordinates in either Absolute or Relative form:

Absolute displays the axis coordinates relative to the Home.
Relative displays the axis coordinates relative to the Reference.
Appearance of Location element on the editing page of the interface:
 
When right-clicking on the Location element on the page and selecting the Properties item, a window opens in which you can adjust the variables related to the Location element. All the variables available in the Properties window are explained in Table 21.</td></tr>
<tr><td>9</td><td>Add G-Code</td><td>After loading the desired file using the Open element, in this section, G-Code and program lines are displayed. Additionally, during program execution, the executed parts are identified with a different color.
Appearance of G-Code element on the editing page of the interface:
 
When right-clicking on the G-Code element on the page and selecting the Properties item, a window opens in which you can adjust the variables related to the G-Code element. All the variables available in the Properties window are explained in Table 21.</td></tr>
<tr><td>10</td><td>Add Tool Path</td><td>The loaded file is displayed in this section, and you can observe the machine's operation and the order of program code execution, both in simulation and during operation. Additionally:</td></tr>
</table>

		In the upper-left part of this section, the address of the file that the user has loaded and the Work Size, which displays the dimensions in different axes, and the Work Length, which displays the total length of the workpiece, are visible. In the lower-right part of this section, you can view the file from different perspectives and zoom in or out on the design. Holding down the right mouse button allows you to view the design from different angles. Appearance of Tool Path element on the editing page of the interface:  When right-clicking on the Tool Path element on the page and selecting the Properties item, a window opens in which you can adjust the variables related to the Tool Path element. All the variables available in the Properties window are explained in Table 21.
11	Add Progress Bar	It is used to display information such as: In normal mode, it displays the interface name. During operation, it displays the estimated time and the remaining time for the operation. In case of any alarm or emergency, this section changes to red and displays the created alarm. Double-clicking opens the Debug Form window, where you can view the performance report of the controller and software, the Radonics software version, the interface type, and alarms. Pressing the CTRL+SHIFT+ALT+PrtSc shortcut key opens the device performance report in Notepad. Appearance of Progress Bar element on the editing page of the interface:  When right-clicking on the Progress Bar element on the page and selecting the Properties item, a window opens in which you can adjust the variables related to the Progress Bar element. All the variables available in the Properties window are explained in Table 21.
12	Add Tab	It is used to add a new section or Tab. Please note that this element is not active in all parts of the editing page of the interface and can only be viewed by right-clicking on the lower toolbar of the editing page of the interface. (The active range of this element is shown in the figure below.) When right-clicking, a new Tab is created, as shown in the figure below. Pressing Add Tab adds a new Tab to the interface. Appearance of Tab element on the editing page of the interface:  When right-clicking on the Tab element on the page and selecting the Properties item, a window opens in which you can adjust the variables related to the Tab element. All the variables available in the Properties window are explained in Table 21.

13	Refresh	Refresh or Reload is a common term used to update or reload the displayed or saved content. For example, if a user changes the size of the Location element but the variables in this element do not change, then Refresh is used in this case.
14	Properties	By clicking on the Properties item, a window is opened to view and apply changes to the variables related to the desired element.
15	Main Setting	Initially, by clicking on the Main Setting item, the Setting window with two different tabs, Setting and Security, is opened. In the Setting section, all the main settings of the Radonix software are included, and in the Security section, all the settings related to activation code and time locks are included, as well as the Export button to get a backup file of the applied settings up to that moment. In this page, there are also links to the Radonix Telegram channel and website, which by clicking on them, you can easily enter the official Telegram channel and website of the Radonix company. 
16	Grid / 1	Elements can be moved within the rectangles specified in the figure below. As the rectangle is not divided into smaller units, the movement is carried out with a low precision. 
17	Grid / 2	Elements can be moved within the rectangles defined in the image below. In this case, since the rectangle is divided by 2, the movement is done with a precision of 2 times the Grid/1.

		
18	Grid / 4	Elements can be moved within the rectangles defined in the figure below. Since the rectangle is divided by 4, movement is done with a precision of 4 times the Grid/1. 
19	Grid / 8	Elements can be moved within the rectangles specified in the figure below. Since the rectangle is divided by 8, movement is done with an accuracy of 8 times Grid / 1. 
20	Minimize	It is used to minimize or reduce the size of the interface editing window.
21	Exit	It is used to exit the interface editing window.

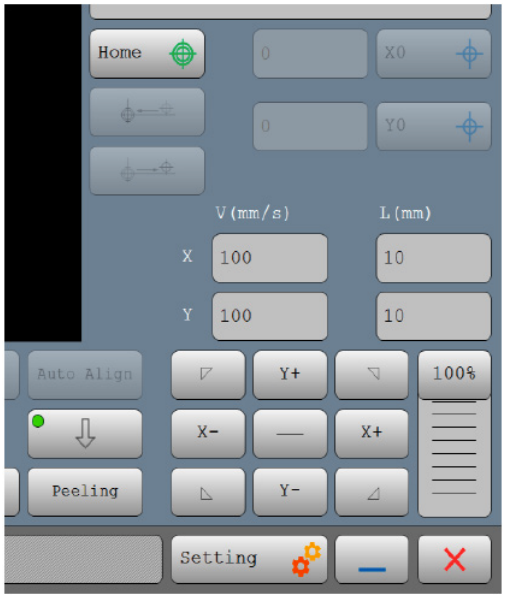
Table(20)

In the table below, all variables available in Properties are fully explained.

Table(20)

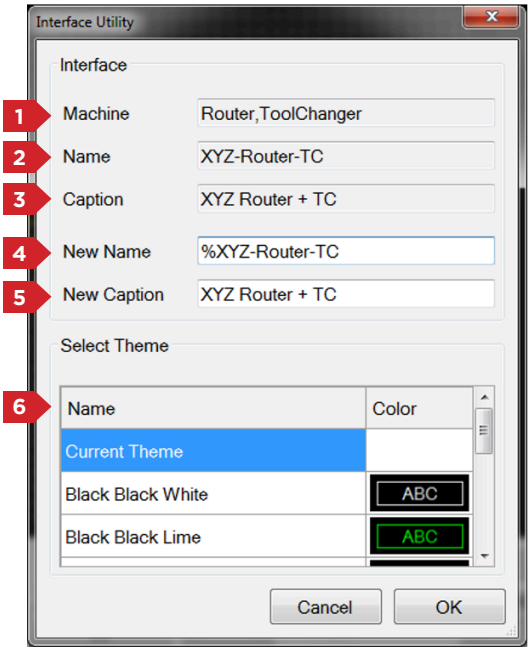
Introduction to Visual Effects

In this section, we will provide a comprehensive explanation of Visual Effects and its capabilities. Initially, by pressing the Ctrl+Shift+Alt+F5 shortcut keys simultaneously, the Visual Effects window can be opened. This window enables the user to activate Visual Effects in the current interface and select a new theme. Using these themes, the user can change the overall appearance of the program and also create a new interface with a custom name and caption while preserving the previous interface. Additionally, the display method of elements within the page can be modified. For example, the border color, corner rounding of elements, color gradient, and the presence or absence of a light in Toggle elements can be changed. (Figure 28)



Figure(28)

After pressing the shortcut key Ctrl+Shift+Alt+F5 simultaneously, the Interface Utility window (Figure(29)) opens.



Figure(29)

	Descriptions
1	This displays the type of interface (device type).
2	This is the current name of the interface. It is the name with which the interface has been saved.
3	This is the current caption of the interface. It is visible on the interface selection page in case there are multiple interfaces.
4	The new name for the interface is defined in this section.
5	The new caption for the interface is defined in this section.
6	A list of pre-defined themes is provided in this section. If you do not want to change the interface theme, select Current Theme.

Table (22)

Chapter 6

Calibration

User Guide

Concept of calibration

In the Iranian national standard, calibration refers to the comparison of a measuring device with a standard and determining the amount of error in the device relative to the standard, and if necessary, adjusting the calibration in comparison with the relevant standards. In other words, calibration refers to the process of aligning the motion value of an axis with one of the standard units or with a specific value. Common units in industrial devices are usually millimeters, centimeters, meters, inches, feet, degrees, etc., which are available for use based on the need. Therefore, the concept of calibration for the millimeter unit can be expressed as the alignment of the motion value of the desired axis with the millimeter unit. That is, in the case of measuring with measuring tools, the numerical value of the motion of the axis should be consistent with the measured value in millimeters.

In Radonix controller, each axis has the ability to calibrate separately, and even the axes can be calibrated in different units. The variable representing the calibration factor in this controller is the Step variable, which to access it, you need to open the Radonix software and after opening the Setting window, go to the System branch and then the Axis{n} sub-branch ({n} represents the number between 1 to 6, which specifies the axis number.), where the Step variable is located, and each axis has its own specific Step variable. (For more information, please refer to chapter (5) and table (16)

The types of calibration methods are:

- Computational method
- Measurement method

Computational method

To adjust the axis motion value to the desired unit, we use the calibration coefficient called “Step.” To access this variable, open the software and go to the “System” branch and then the “Axis{n}” sub-branch (where {n} is a number between 1 and 6 representing the axis number). The Step variable is located in each of these sub-branches, and each axis has its own Step variable.

If the unit of axis motion is millimeters, the unit of this variable will be millimeters per pulse. If the unit is centimeters, inches, or any other unit, the Step variable will have the same unit per pulse. It represents the amount of axis motion per pulse sent to the motor driver. Therefore, to calculate this variable, we need to know the relationship between the motor motion value and the input pulse to the driver, as well as the conversion ratios of gearbox, pulleys, and any other motion transmission.

For better understanding of these calculations, consider the following example:

Example 1: Consider a linear axis of a CNC machine with a measurement unit in millimeters, which uses Panasonic A5 motors with a 10/1 gearbox and is connected to a shaft gear with an effective diameter of 66 millimeters via a pinion. Now, calculate the Step value for the given axis.

- Note that each of these pieces of information plays a fundamental role in these calculations, and by using this information, it is possible to calculate the axis step for calibration. For example, knowing the type of motor driver and its specifications determines the relationship between the input pulse value to the driver and the amount of motor shaft movement.

In Panasonic A5 motors, at a rate of 500,000 pulses per second, the motor reaches a speed of 3,000 RPM. In the first step, we calculate the motor’s rotational speed per second. In this case, the number 3,000 means the number of revolutions per minute. To convert it to seconds, we divide it by 60, which is equal to 50 revolutions per second. Therefore, the equation is written as follows:

Motor speed per second = 3000 (r/m)/60(s)=50 (r/s)

In the second step, we need to calculate the linear displacement for one rotation. In this example, the distance traveled will be equal to the circumference of the circle, which is equal to the diameter of the circle multiplied by pi (3.1415). As mentioned in Example 1, a pinion with a diameter of 66 millimeters and a 10/1

gearbox are used in the mechanics. Therefore, we have:
Linear displacement for one rotation of the motor = (1/10) * 66(mm) * 3.1415 = 20.7339(mm)

The result is the linear displacement for one rotation of the motor.

The final step is to write the final equation and find the unknown variable, which is the Step variable. As we have seen, all calculations are based on 500,000 pulses sent by the controller and the motor speed of 3,000 RPM. Therefore, to calculate the displacement or distance traveled for one pulse, we need to divide the calculated number by 500,000. However, as mentioned, this motor has 50 revolutions per second. So, we need to multiply the result obtained in the second step by 50 to find the total distance traveled per second of motor rotation. Now we will rewrite the general equation.

$$\text{Step(mm/pulse)} = \frac{50(\text{r/s}) * 20.7339(\text{mm})}{500000(\text{pulse/s})} = 0.00207339\ (\text{pulse/mm})$$

The step value is calculated from this equation, which means that the motor moves 0.00207339/0 millimeters by taking 1 pulse.

- Note that the Computational method used in Example 1 is for the calibration of linear axes.

The general equation for linear axis calibration is as follows:

$$\text{Step(unit/pulse)} = \frac{\text{The amount of axis movement per one revolution of the motor} * \text{The number of motor rotations per second}}{\textit{The number of controller pulses sent per second.}}$$

Example 2: Consider the rotary axis of a CNC machine whose measuring unit is in degrees, and Panasonic A5 motors with a 25:1 gearbox connected to a 12-tooth pulley, which is then connected to a 44-tooth pulley via a belt. Now, calculate the step value for the axis in question using computational method 2.

- Note that one revolution of the shaft is equivalent to 360 degrees.

In Panasonic A5 motors, at a rate of 500,000 pulses per second, the motor reaches a speed of 3000 rpm. Now, in the first step, we calculate the motor rotation speed per second. In this case, the number 3000, which represents the number of motor revolutions per minute, is divided by 60 to convert it to seconds, resulting in the number 50. This means that this motor rotates 50 times per second. Therefore, the equation can be written as follows:

The motor rotation speed per second = 3000(r/m)/60(s)=50(r/s)

In the final step, we need to multiply the effective values such as gearbox ratio, pulley ratio, and motor rotation speed per second by the number 360. As we have seen, all calculations were based on 500,000 controller pulses and 3000 rpm motor speed. Therefore, to calculate the motion value for 1 pulse, we need to divide the obtained number by 500,000, which will result in the Step value with this simple estimate.

$$\text{Step} = \frac{50(\text{r/s}) * (1/25) * (12/44) * 360}{500000(\text{pulse/s})} = 0.000392727272$$

From this equation, the Step value is calculated.

The general equation for linear axis calibration is as follows:

$$\text{Step(unit/pulse)} = \frac{360° * \text{Effective values} * \text{Motor rotation speed per second}}{\text{The number of controller pulses sent per second}}$$

- Note that in both linear and rotary axis calculations, using more significant figures will result in more accurate calculations.
- The pulse divisions in steppers have a direct effect on these calculations, so for the sake of simplicity, the motor rotation value for a specific number of pulses should be calculated first, and then the result should be applied to the above equations.
- In drives where the number of pulses required to reach the maximum motor speed is greater than the number of pulses produced by the controller, increasing the electronic coefficient of the drive by a specific ratio can allow the motor to reach maximum speed. For example, if a drive requires 4 million pulses per second to reach the maximum motor speed and considering the production of 500,000 pulses per second by Radonics, the electronic coefficient of the drive should be increased by a ratio of 8 to reach the maximum motor rotation speed.

Measurement Method

In this calibration method, the basis is measurement with measurement tools and equipment. Therefore, the accuracy of the measuring tool, measurement accuracy, and mechanical errors of the device directly affect the quality of calibration. Although this method is not as precise as computational methods, it is a widely used method due to the lack of information on motors, gearboxes, pulleys, and gear wheels, which are sometimes not available. The basis of this method is measuring the displacement value for a specific number of pulses.

To facilitate, improve accuracy, and speed up calibration using the measurement method, Radonics has provided a free software called CAM-Pro Calibrator, which is automatically installed with the main Radonics software. Based on the physical displacement measurement of the axis and the required number of pulses for this displacement, the software calculates the step value. (Figure 1)

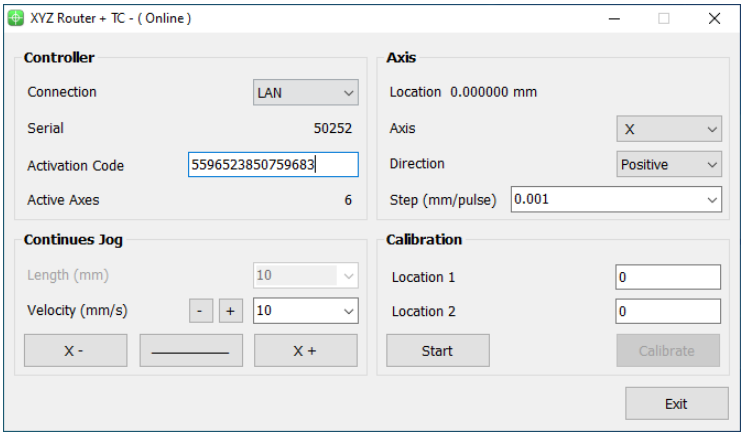


Figure (1)

Since pulse counting in this method is the responsibility of the controller, there is no error in pulse measurement. Therefore, the measurement accuracy in this method is directly related to the physical measurement accuracy. In addition, the greater the distance between the two measured points, the larger the pulse value will be, which means the denominator will increase, and according to the constant error, the overall step value obtained will be more accurate. Therefore, calibrating based on two points will result in more accurate results.

- Note that you should never run two Radonix software at the same time because in this case, one of the software will not go online. Therefore, make sure to close the other Radonics software when running the desired software.

Introduction to CAM-Pro Calibrator

As explained in the previous section, the CAM-Pro Calibrator software is installed along with the CAM-Pro software. There are two ways to find the Calibrator software after installation. According to the first method, simply go to the installation location of the software, which is usually in the C drive of Windows by default, then go to the (x86) Program Files folder, open the Radonix folder, and the CAMProCalibrator software is visible in the Radonix CAM-Pro file. According to the second method, simply go to the Start menu and search for Radonix CAM-Pro Calibrator in the Search option (Step 2 in Figure 2).

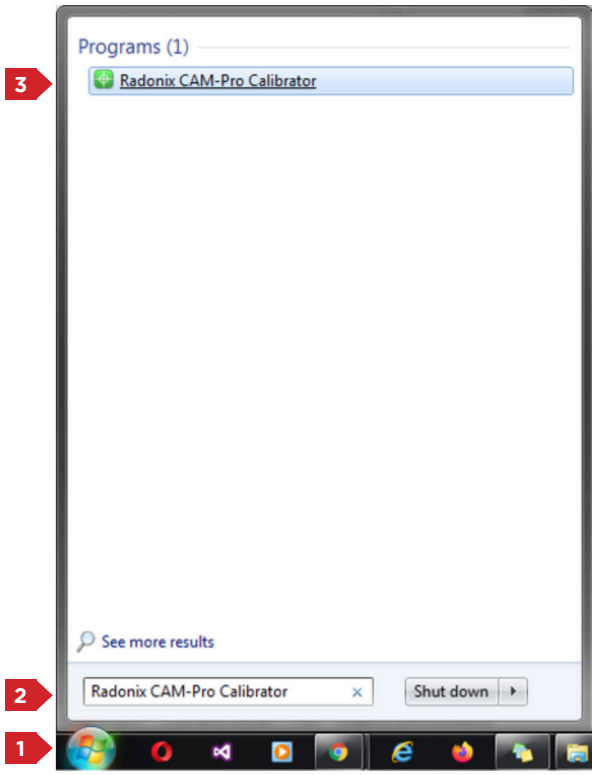


Figure (2)

This program consists of two windows. The first window only runs when multiple interfaces are active on a computer, as shown in Figure (3), and it allows the user to select the desired interface and execute it by clicking the Ok button. If there is only one active interface on the computer, the interface selection window will not appear, and the main window will open directly.

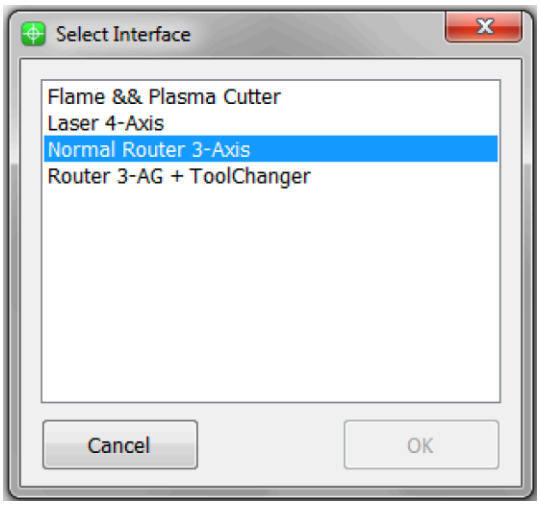


Figure (3)

When opening the CAM-Pro Calibrator software, you will be faced with Figure (4) which includes the following components in order:

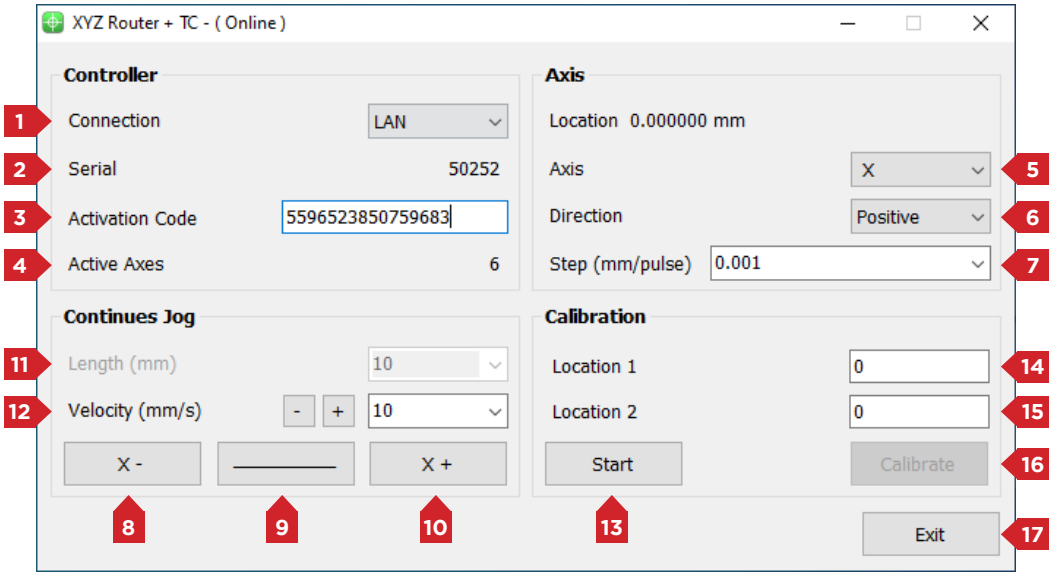
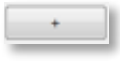


Figure (4)

Controller		
1	Connection	In ProLan controllers, it should be in LAN mode. In ProUSB controllers, it should be in USB mode.
2	Serial	Displays the controller's serial number.
3	Activation Code	The 16-digit activation code that activates the axes.
4	Active Axes	Displays the number of axes activated by the activation code.
Axis		
5	Axis	Axis is used to select the axis for calibration.
6	Direction	Selecting the appropriate direction of movement to align with standard axes. Negative represents movement towards the negative direction and Positive represents movement towards the positive direction.
7	Step	Step or calibration coefficient is the output of the software that is automatically recorded in the interface after the calibration operation.
Continues Jog		
8		Manual movement towards the negative axis.
9		Used to select the type of manual movement (continuous/discrete). If the sign is it  represents continuous movement, and if it is  it represents incremental movement.
10		Represents manual movement towards the positive axis.
11	Length	Displays the value of incremental movement, which becomes active when selecting incremental movement.
12	Velocity	Used to set the speed of manual movement and the + and - keys next to Velocity are used to increase and decrease the speed of manual movement.

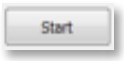
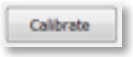
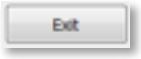
Calibration		
13		key to start the calibration process.
14	Location 1	Used to determine the initial position for calibration. This position should be measured relative to a fixed point on the measurement axis.
15	Location 2	Used to determine the end position for calibration. This position should be measured relative to a fixed point on the measurement axis.
16		Finish Calibration and Calculate Step or Calibration Coefficient is the key to end the calibration process.
17		Exit Program is the key to exit the program.

Table (1)

For example, to calibrate one of the axes, we must follow the steps below:
First, select the axis to be calibrated from the Axis section. Then, by selecting a low speed and moving the axis, we ensure the direction of the axis movement is correct. If the direction is incorrect, select the appropriate direction from the Direction section. Then, move the axis to one of the two end points of its motion range. Since the starting point is not important in this method, it doesn't matter which end point is chosen. After positioning the axis in the appropriate place, a point on the axis is measured with a suitable measuring tool relative to a fixed point on the machine and recorded in Location 1. Then, press the Start button. Note that the selection of the unit is entirely optional. Therefore, if the device needs to be calibrated with a specific unit, all measurements must be made in that unit. For example, the device can be calibrated in inches, meters, centimeters, millimeters, or even micrometers.

Move the axis in the opposite direction. If measuring at greater distances is possible and the measuring tool has sufficient accuracy, measurements at greater distances result in more accurate results. After moving the axis to the second point, measure it relative to the fixed point on the measuring machine like the first point, and record it in Location 2. Then, press the Calibrate button. Using these measurements, the calibration step or coefficient is calculated and automatically recorded in the desired interface. Calibration can be performed for any number of axes as needed.

Note that if the activation code is not entered, an error message will be displayed when opening the CAM-Pro Calibrator software.

Chapter 7

Controller Test Software, Remote, and Handwheel

User Guide

Introduction To CAM-Pro Test

CAM-Pro Test software is installed along with CAM-Pro software and there are two ways to find it after installation. According to the first method, simply go to the location of the software installation, which is by default in the C drive of Windows, and then go to the (x86) Program Files folder and open the Radonix folder. The CAM-ProTest software can be found in the Radonix CAM-Pro file. According to the second method, simply search for the Radonix CAM-Pro Test software in the Start menu search bar (step 2 in figure 1) and then run it. (Figure 1)

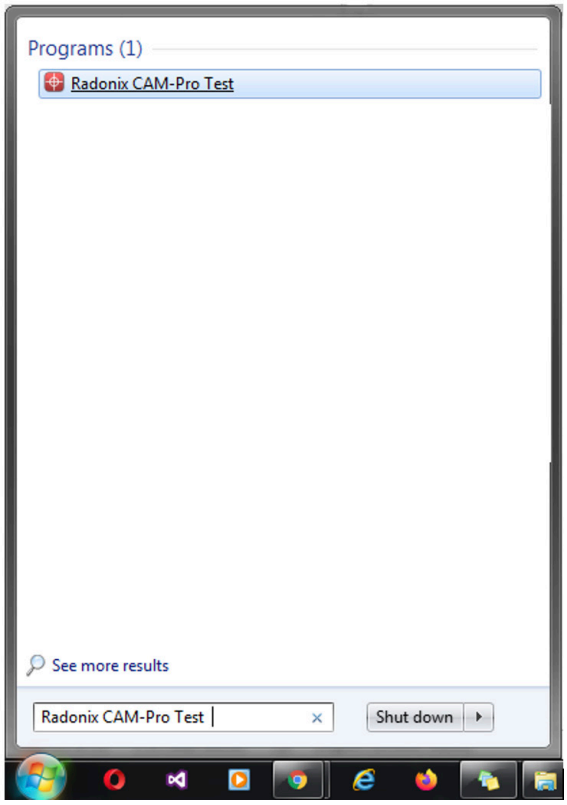


Figure (1)

When running the CAM-Pro Test software, you will be faced with Figure (2), which includes groups of Connection, Activation, Axes, InPort, OutPort, Analog, Clock, Remote, and Handwheel, each of which contains various components. In Table (1), we provide separate definitions for all of these components.

- Note that you should never run two Radonix software at the same time, as this will cause one of the software to not go online. Therefore, be sure to close the other Radonix software when running the desired software.

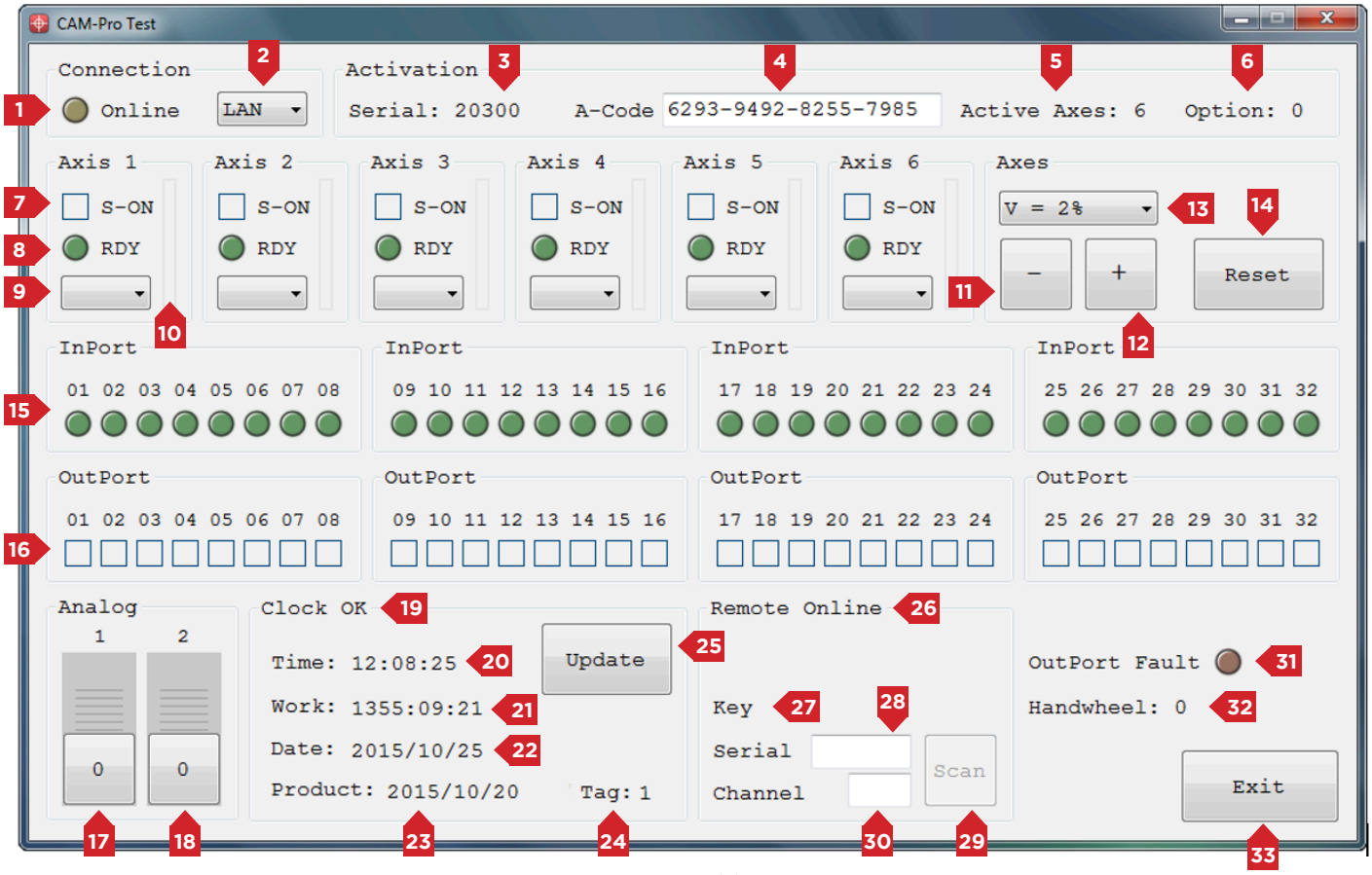


Figure (2)

	Connection
1	It displays the connection between the controller and the computer. If the connection between the controller and the computer is established correctly and the IP is set correctly (please refer to Chapter 3.3.1, section “Defining IP” for IP configuration), the light turns on.
2	It is used to select the type of connection between the controller and the computer (USB/LAN). In Pro USB controllers, it must be set to USB mode, and in PC-Pro LAN and PC-Smart controllers, “LAN” must be selected.
	Activation
3	The controller serial number is displayed in this section after the controller goes online.
4	This is the place to enter the activation code or 16-digit code. By entering the activation code, the required axes will be activated based on the type of code and the type of purchased controller. After entering and confirming the activation code by pressing the Enter button, the axes will change from gray and off state to colorful and on state.
5	Displays the number of activated axes.
6	Displays the purchased option code.
	Axis
7	By clicking on the box, the corresponding axis will be activated (the motor related to that axis will turn on).
8	This displays the feedback “Ready” for the axes. If the wiring between the controller and drivers is correct, and the axis is activated by clicking on the S-ON box (which stands for Servo ON), the RDY box (which stands for Ready) will turn on. The on state of the RDY box indicates the normal state of the corresponding axis. Note that the Radonix board also has LEDs for Servo ON and Ready, and the same thing happens to those LEDs as well.

9	This option is used to change the axis movement.
10	This displays the pulse rate for the axis, which has a direct relationship with the speed of the axis movement.
11	This is used to move the motor in the negative direction of the corresponding axis. In other words, if the corresponding axis is active and its RDY light is on, clicking and holding on the - box will cause the motor to turn clockwise or counterclockwise. Note that if the axis moves in the positive direction due to wiring or driver logic, the movement direction can be changed using section 9 of this table.
12	This is used to move the motor in the positive direction of the corresponding axis. In other words, if the corresponding axis is active and its RDY light is on, clicking and holding on the + box will cause the motor to turn clockwise or counterclockwise. Note that if the axis moves in the negative direction due to wiring or driver logic, the movement direction can be changed using section 9 of this table.
• Please note that if the motor rotates in one direction by pressing both + and - buttons, there may be a mistake in the wiring.	
13	This is used to change the pulse speed in percentage. In other words, motor speed can be changed in percentage using this element.
14	If any of the motors have an alarm and the corresponding axis is disabled, this element is used to execute a reset command to clear the motor alarms.
Please refer to Chapter 3, section 3.2.6 (Axes) for more information.	
InPort	
15	This displays the status of the digital inputs on the Radonix controller, and if any of the inputs are turned on, the corresponding LED in the software will also turn on. For example, if input number 2 on the controller is turned on, LED 02 will turn on in the software. Please refer to Chapter 2, Section 2.2.3 (Digital Inputs) for a complete understanding of the inputs' operation and the concepts of NPN and PNP.
OutPort	
16	The toggle buttons are used to turn the digital outputs on and off on the Radonics controller, and by clicking on each of the squares in the Out Port section, the corresponding LED for that output should turn on on the Radonics board. For example, if you click on the square for output number 02, the LED number 00.02 on the controller should turn on. Please refer to Chapter 2.2.4 (Digital Outputs) for a complete understanding of the output functions.
Analog	
17	This is used to test analog output 1 and PWM1.
18	This is used to test analog output 2 and PWM2.
These scroll bars range from 0 to 100 and correspond to variable voltage output. The number 0 corresponds to 0 volts and the number 100 corresponds to 10 volts. To ensure the health of the analog outputs, it is sufficient to measure the analog voltage with a multimeter. The black probe should be placed on GND and the red probe should be placed on AO1 or AO2. Now by moving the scroll bar, the displayed voltage on the multimeter should change in the same proportion. Additionally, to test the health of the PWM outputs on the board, the PWM voltage value can also be viewed. Note that the PWM voltage value is always half of the AO voltage value. For example, if AO1 is 10 volts, PWM1 is 5 volts. For a complete understanding of the analog and PWM output function, please refer to section 3.2.5 (Analog and PWM Outputs) of chapter 3.	
Clock	
This section is designed to test the functionality of the embedded controller clock, and to activate this section, the T-Code must be entered in the Activation group and in the rectangle where the 16-digit Activation Code is entered. By entering the T-Code, this section will be activated and will change from the off and gray state to the on and colorful state, and the Update option will also be activated. T-Code stands for Time Keeper Code, which is a 16-digit code similar to A-Code that is responsible for activating the controller clock and the settings related to the time lock.	

19	After verifying the entered T-Code, the status of the controller's internal clock can be viewed from this section. (OK/Error) If the internal clock is healthy and the T-Code is correct, OK will be displayed.
20	It is used to display the controller's internal clock. If the Update button is pressed, the displayed time is synchronized with the computer's clock, and the internal clock of the controller can be adjusted in this way.
21	It is used to display the device's operating time.
22	It is used to display the controller's internal date.
23	It is used to display the date of the controller's update.
24	It is used to display the controller's tag.
25	The command button for updating the internal clock of the controller.
• Note that in order to update the controller's date, the computer's clock must be up to date. Otherwise, the time locks will not work properly.	
Remote	
• Please note that in order for this section to be activated, the CDM21XXX driver must be installed. If this driver is not installed, the remote section in CAM-Pro Test software will be grayed out and offline. For instructions on installing the CDM21XXX software, please refer to Chapter 5, section 5.2.2.6.7 (Remote).	
• Please refer to Chapter 5, section 5.2.2.6.7 (Remote) for a complete understanding of the remote functionality before proceeding with the explanations in this section.	
26	This section is used to display the status of the remote (Online/Offline).
27	This section is used to display the number of the button pressed on the remote transmitter, which is used to test the remote. In fact, if the remote serial number and channel are entered correctly, pressing any button on the remote transmitter will display a number corresponding to that button in front of the Key.
28	The serial number of the remote transmitter must be entered in this section. The serial number of the remote transmitter is a 5-digit number that is printed on the back of the remote case and must be entered in the rectangle in front of Serial, and then the Enter button must be pressed.
29	The Automatic Scan command is used to automatically scan the remote, which allows you to find the channel number by holding down one of the transmitter buttons.
30	The frequency channel number of the remote transmitter is entered in this section. If the user is not aware of their remote channel number, the method of changing the remote transmitter channel is described in chapter 4, section 4.2.2.6.7 (Remote), and the corresponding table is provided. Please refer to that section.
Handwheel	
31	This is used for counting Hall sensor pulses. Please refer to Chapter 2, section 2.2.7 (Hall sensor and RS-485) for information on the structure of the Hall sensor terminal on the Radonix board and how it operates.
OutPort Fault	
32	As the outputs are protected against short circuits and high current-induced heat, in case of any of these occurrences, an error will be displayed by the red LED labeled O-Error on the board. In case of an error, all outputs will be deactivated by the controller. Also, a red OutPort Fault light is embedded on the software, which lights up in case of such errors.
Exit	
33	By pressing this button, the CAM-Pro Test software will be closed.

Table (1)

Appendix 1

Connection between Radonix controller (DB15) and types of drives

User Guide

Radonix DB15 Connector														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

DELTA ASDA-B2 Connector														
11	37	39	41	43	9	7	44	33	21	22	25	23	30 31 32	6 14

DELTA ASDA-B2 دو درایور Connector														
11(1,2)	37(1,2)	39(1,2)	41(1,2)	43(1,2)	9(1,2)	7(1)	44(1,2)	33(1,2)	32(1,2)	31(1,2)	30(1,2)	19(1,2)	6(2)	14(1,2)

DELTA ASDA-A2 Connector														
11	37	36	41	43	9	7	5	33	21	22	25	23	4 13 34 47	10

ADTECH QS7 Connector														
9	26	27	24	25	10	5	28	14	-	6	16	29	-	-

ALPHA5 Smart Connector														
1	21	20	8	7	2	9	25	3	-	10	14	26	-	-

BONMET SA series Connector														
18	35	34	33	32	10	8	7	11	-	27	9	-	-	-

DELTA ASDA-A Connector														
11	37	36	41	43	9	7	48	33	13	6	30	31	32	49

DELTA ASDA-A2 Connector														
11	37	36	41	43	9	7	48	33	13	6	30	31	32	49

DELTA ASDA-B Connector														
4	20	19	22	21	17	16	25	18	-	3	8	13	14	15

DORNA-EPS-B1 Connector														
47	12	11	8	7	40	31		45			32			

ESTUN EDB Connector														
9	26	27	24	25	10	4		14		5	3	12	13	

Radonix DB15 Connector														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

ESTUN EDC Connector														
16	14	13	12	11	15	4		6		5				

ESTUN ETS Connector														
9	26	27	24	25	10	6		14		5	12	13	36	

ESTUN PRONET Connector														
13	33	32	31	30	14	9		39		10		16	17	

Fuji-ALPHA5 Smart series Connector														
1	21	20	8	7	2	15	25	3			14	26		

HIGEN FDA7000 Connector														
49	12	9	10	11	18	22		42		8	15	24	25	40

Inovance IS620P Connector														
11	39	37	43	41	33	7	44	8		9	10	14	29	6

JMC JASD Connector														
11	37	36	41	43	9	7	48	10			6	44		

Kinco-CD420 Connector														
3	14	13	12	11	4	19		5		20	8	9		

Kinco-FD Connector														
2	33	31	29	27	4	1		6		3	12	24	14	

Kinco-JD Connector														
1 3 8	15	10	5	4	4	18		5		2	9	10	19	

Kingservo KSDG Connector														
7	6	5	4	3	29	35	19	31				34	13	25

Kollmorgen Connector														
X8-3	X7-10	X7-1	X7-9	-	X8-4					X8-5	X8-6			

Leadshine EL5 Connector														
1	6	5	4	3	2	32	-	-		7	8	9	31	

Leadshine ES-D808 Connector														
ENA+	DIR-	DIR+	PUL-	PUL+	ENA-	ALM+					ALM_			

Radonix DB15 Connector														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Lenze EVS9324-ET Connector														
X5(E1,E2,28)	X9(9)	X9(1)	X9(2)	X9(3)	Radonix InPorts-PNP						X5(39)	X9(5)		

LS XDL-S Connector														
50	12	11	10	9	47	40		17		36	41	18	19	20

Maxsine EP2 Connector														
1	6	19	7	20	14	4	22	2		3	10	15	18	

MECAPION APD-VS Connector														
50	12	11	10	9	47	40	3	17		36	41	18	19	20

Mitsubishi MR-E Connector														
1	24	25	22	23	4	11		3		6	7	8	13	

Mitsubishi MR-J2S Connector														
13 (CN1B)	2 (CN1A)	12 (CN1A)	3 (CN1A)	13 (CN1A)	5 (CN1B)	18 (CN1B)	14 (CN1A)	14 (CN1B)		10 (CN1B)	15 (CN1B)	16 (CN1B)	17 (CN1B)	1 (CN1A)

Mitsubishi MR-JE Connector														
20 21	35	36	10	11	15	49	33	41		34	42	46	47	

NG100H Connector														
16	22	6	21	5	20	31	27	4		3	32	19	11	

Omicron-MK6 series Connector														
10	14	4	13	3	12	19	6	2		1	11	15	20	

OMRON sigma-II Connector														
47	12	11	8	7	40	29		44				30	42	43

Panasonic MINAS A4 Connector														
7	6	5	4	3	29	35	19	31		8	9	13	25	34

Panasonic MINAS A5E Connector														
7	6	5	4	3	29	35	19	31		8	9	13	25	34

Panasonic Minas-LIQI Series Connector														
1	23	22	21	20	2	10	19	3		6	7	11	12	

Radonix DB15 Connector														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

SANMOTION-RS1 Series Connector														
49 50	27	26	29	28	37	41	11	15		12	24	25	5	6

SANYODENKI-Q-S Connector														
50	27	26	29	28	37	41				24			48	

SIEMENS-SINAMICS Connector														
4 28	27	26	2	1	5	3	17	6		7	8	13	49	50

Stepper Connector														
	DIR-	DIR+	PUL-	PUL+	*	*								

TECO JSDA Series Connector														
47	17	16	15	14	1	18	43	2			4	5	48	49

TECO-JSDE Connector														
17	7	6	5	4	1	18	11	14						24

TECO SR, TSDA and SD Series Connector														
45	17	16	15	14	1	18	43	2				4	5	48

TECO TSTA Series Connector														
47	17	16	15	14	1	18	43	2				4	5	48

TETA D2 Connector														
6	15	2	14	1	19	16				3	8	20		

WEIDE B2 series Connector														
20	11	12	14	15	17	24		2			38	8		

YASKAWA Σ-V Series Connector (SGMJV / SGMAV / SGMPS / SGMGV / SGMCS Servomotors and SGDV SERVOPACK)														
47	12	11	8	7	40	29		44				42	43	30 47

YASKAWA-SGD7S Connector														
47	12	11	8	7	40	38	29	44	1	30	42	43	37	39

YASKAWA-SGDM Connector														
47	12	11	8	7	40	29		44				42	43	30

YL-VPV Series Connector														
	29	28	27	26	6	10		7				5		

Radonix DB15 Connector														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

YL-VPV Series Connector														
8	7	19	6	18	21	1		23			3			

YUHAI SG-AS Connector														
24	3	18	4	19	7	23		25		9	38	39		

Radonix PC and Delta VFD-B Connection

- Note that in Delta VFD-B, switch SW1 should be set to Sink mode.

Radonix PC Pins	Description	VFD-B Pins
AO1/AO2	6 Pin Connector	AVI
GND	6 Pin Connector	ACM
Onn	OutPort 8-Pins Connectors	FWD
Onn	OutPort 8-Pins Connectors	REV
GND	Power Ground	DCM

Appendix 2

Shortcut keys

User Guide

	Shortcut	Description
1	CTRL+SHIFT+ALT + B	It saves a backup version within the interface. This feature can be used to maintain factory settings along with current settings. After pressing this shortcut key, a window will open to confirm the operation, and by pressing YES, the operation will be performed correctly.
2	CTRL+SHIFT+ALT + C	It is used to clear the Tool Path.
3	CTRL+SHIFT+ALT + E	Displays the G-code editing window. With this feature, the user can type G-code or edit the existing G-code.
4	CTRL+SHIFT+ALT + F	It is used to search for a code or text in the G-code list. Also, with this command, you can identify the line that contains that code or text.
5	CTRL+SHIFT+ALT + K	It is used to activate the virtual keyboard. To enter a number using a mouse or touch screens in interfaces that require numeric input, this command can be used to enable or disable the keyboard. The virtual keyboard can also be enabled or disabled in the Keyboard folder in the settings.
6	CTRL+SHIFT+ALT + L	It is used to open the Condition key page or the password entry page for timed locks.
7	CTRL+SHIFT+ALT + M	It is used to execute special codecs. With this command, if a special codec software window is available in the interface, it can be executed. Not all interfaces have this feature.
8	CTRL+SHIFT+ALT + N	It is used in G-code to move to a specific line number. In fact, this option makes it easier to go to a specific position in simple G-code.
9	CTRL+SHIFT+ALT + O	It is used to open a file. With this command, the window for opening input files can be executed. This command is equivalent to the OpenFile function, table number 4 in chapter 5.
10	CTRL+SHIFT+ALT + R	It is used to recover a backup version from within the interface. This feature can be used to return to factory settings.
11	CTRL+SHIFT+ALT + S	It is used to save G-code after editing it and also to open a DXF file and convert it to G-code and save it.
12	CTRL+SHIFT+ALT + T	It is used to open the Transform window. In fact, this shortcut is used when there is a need for rotation or dimension change in G-code, and changes are made only in the X and Y axes.
13	CTRL+SHIFT+ALT + X	It is used to lock G-code files. With this command, the G-code will only run with a specific serial control, and this feature is suitable for designers who do not want their designs to be available to the public.
14	CTRL+SHIFT+ALT + 0	It is used to send a clear error command to the motors. In fact, in case of axis errors in motors, this shortcut can be used to clear the error.
15	CTRL+SHIFT+ALT + ?	It is used to open the Report and Cost Calculation window. In addition to providing a report on the ToolPath, this window allows the user to calculate the cost of the work in several different ways.
16	CTRL+SHIFT+ALT + Backspace	It is used to display installed interfaces. In fact, this window is used to manage installed interfaces on the computer and allows the user to activate or delete
17	CTRL+SHIFT+ALT + Delete	It is used to display the status of internal variables of the controller, axes, inputs, and outputs of the controller. In fact, this feature can be used for device and controller input and output troubleshooting. Please refer to Appendix 4 for more information.

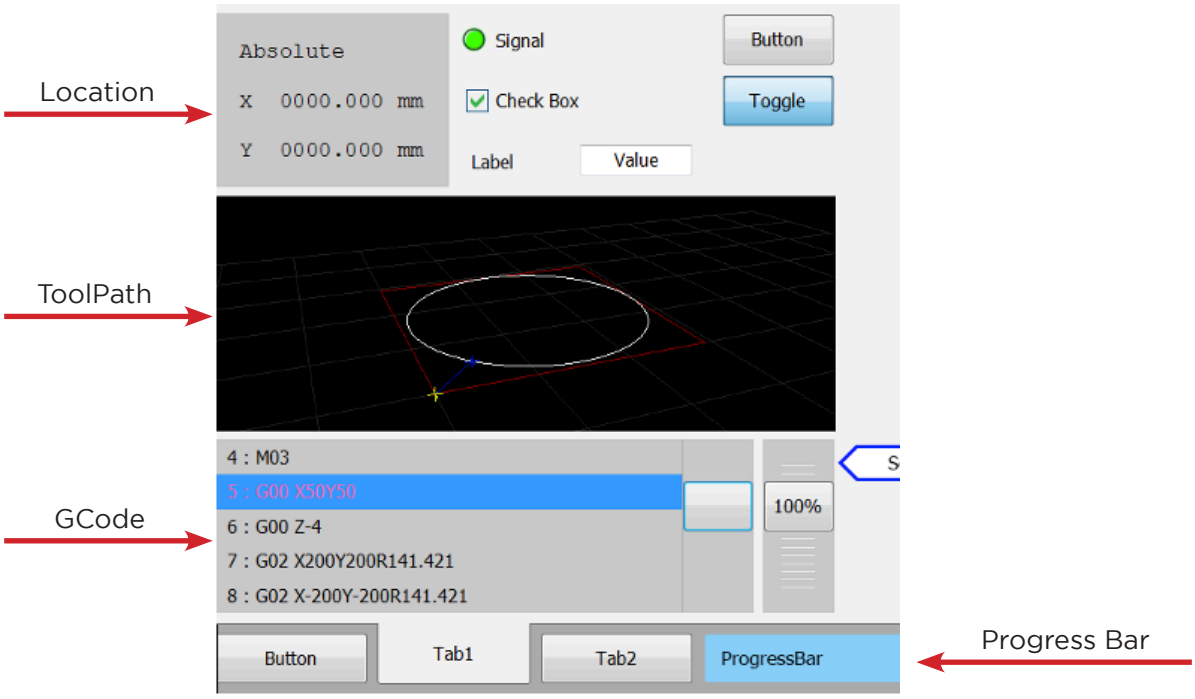
18	CTRL+SHIFT+ALT + End	It is used to open and close the program again, to save settings, and to restart the program.
19	CTRL+SHIFT+ALT + Enter	It is used to display the Settings window and to allow changes to be made in the settings.
20	CTRL+SHIFT+ALT + Home	It is used to enter the interface editing mode. In fact, this feature allows the user to customize and modify the interface appearance
21	CTRL+SHIFT+ALT + PrtSc	It is used to display the program's dialog box, program version, and active interface. Also, if this shortcut is repeated, the daily report page will be opened, which helps to monitor the daily device process and troubleshoot it.
22	CTRL+SHIFT+ALT + Space	It is used to enter or exit simulation mode. In fact, in this mode, the program execution process can be simulated without the need for a controller and device.
23	CTRL+SHIFT+ALT + F3	It is used to send a search command in G-code. In fact, with this command, the search command in G-code can be repeated.
24	CTRL+SHIFT+ALT + F4	When using an interface with a Tool changer, it is used to set the tool position by first moving the tool to the desired position and then using this command to set the position for the tool.
25	CTRL+SHIFT+ALT + F5	It is used to open the Personalization window of the interface. In fact, this tool allows changes to be made in the name, caption, visual theme of the interface, and also to create a new copy of the interface. In addition, this tool also cleans and updates the interface, just like the CTRL+SHIFT+ALT+F12 command.
26	CTRL+SHIFT+ALT + F8	By sending this command, a window containing the coordinates of all axes with 12 decimal places of precision is displayed.
27	CTRL+SHIFT+ALT + F9	By sending this command, a window containing the coordinates of all axes in terms of pulses is displayed.
28	CTRL+SHIFT+ALT + F11	It is used for cleaning up the interface. With this command, it is possible to clean up the interface settings, remove unused variables, and add new ones. If the interface is old, this command removes variables that are no longer functional in the interface.
29	CTRL+SHIFT+ALT + F12	It is used for cleaning up and updating the interface. With this command, it is possible to clean up the interface settings, remove unused variables, and add new ones, and also update the structure of the interface. If the interface is old, this command removes variables that are no longer functional in the interface and transforms parts of the interface into a more structured form with classification. Note that installing an old interface and an updated interface on one device will cause conflicts between variables.
30	CTRL+SHIFT+ALT + Right Click	It is used to display the links defined for the elements of the interface. By right-clicking on the active elements of the interface, their links can be viewed.
31	CTRL+SHIFT+ALT + V	With this command, the last status of the Tool Path display is saved.
32	CTRL+SHIFT+ALT + Up	With this command, the Tool Path display angle around the positive X axis is rotated
33	CTRL+SHIFT+ALT + Down	With this command, the Tool Path display angle around the negative X axis is rotated.

34	CTRL+SHIFT+ALT + Left	With this command, the Tool Path display angle around the negative Z axis is rotated.
35	CTRL+SHIFT+ALT + Right	With this command, the Tool Path display angle around the positive Z axis is rotated.
36	CTRL+SHIFT+ALT + (-)	With this command, the Tool Path view is zoomed out.
37	CTRL+SHIFT+ALT +(+)	With this command, the Tool Path view is zoomed in.

Appendix 3

Shortcut keys for interface elements

User Guide

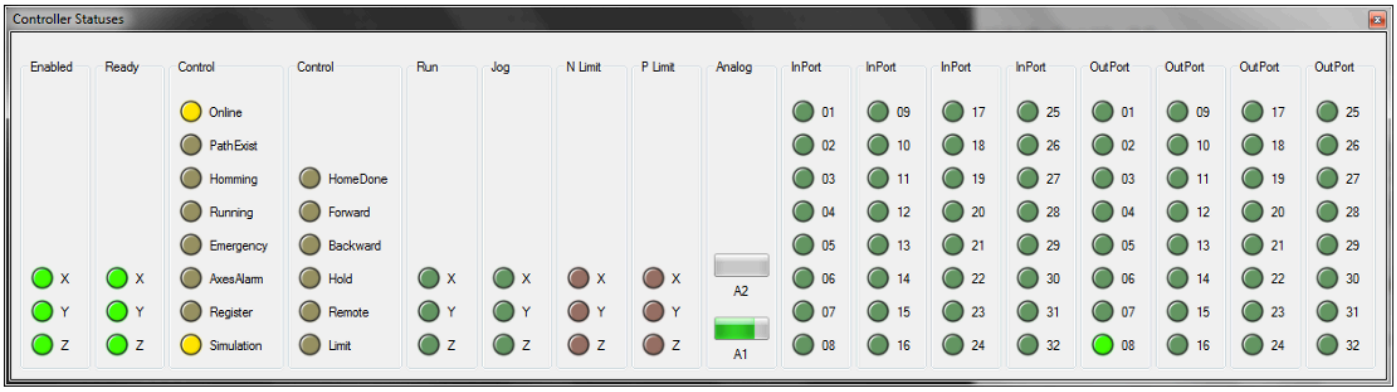


	Shortcut	Element	Description
1	Double Click	Progress Bar	To display the program's diag, program version, and active interface, this is used. To do this, double-click the left mouse button on the status bar element.
2	G-Code	Double Click	This is used to prepare for running the program from the line clicked. To do this, double-click the left mouse button on the line of interest in the G-code list.
3	G-Code	Right Click	To display a full line from the G-code list that has been clicked on, this is used. To do this, right-click the mouse button on the line of interest in the G-code list.
4	Tool Path	Double Click	This is used to change the zoom status between the full desktop view and the piece display view. To do this, double-click the left mouse button in the path display section.
5	Location	Double Click	This is used to display the precise coordinates of the axes with 12 decimal places. To do this, double-click the left mouse button on the position display section.

Appendix 4

Shortcut keys for interface elements

User Guide



	Shortcut	Description
1	CTRL+SHIFT+ALT + Delete	To display the status of internal variables of the controller, axes, controller inputs, and outputs, it is used. In fact, this feature can be used for troubleshooting the device and controller inputs and outputs.

- By clicking on the output signals (OutPort), during program execution, you can turn on or off the desired output to test the outputs.
- Note that using this shortcut without knowledge of the outputs' operation may damage the device.

Appendix 5

Codes

User Guide

Common codes used in Radonix interfaces

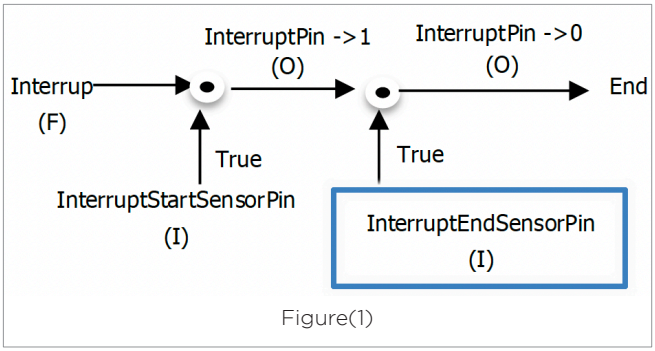
	Code	Description
1	G0	Positioning (Rapid traverse)
2	G1	Linear interpolation (Feed)
3	G2	Circular interpolation (CW)
4	G3	Circular interpolation (CCW)
5	G4	Dwell
6	G54...G59	Work coordinate system select
7	G90	Absolute programming
8	G91	Incremental programming
9	M2	End of program (No rewind)
10	M30	End Program (Rewind stop)
11	M99	End program and repeat
12	P	Dwell interval

Common codes used in Router interfaces

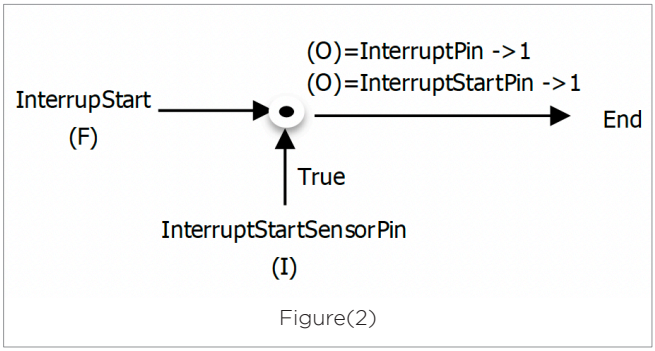
	Code	Description
1	M3	Spindle CW
2	M4	Spindle CCW
3	M5	Spindle stop
4	M6	Tool change
5	M8	Flood coolant ON
6	M9	Flood coolant OFF
7	S	Spindle speed
8	T	Tool number/ Tool change

Appendix 6

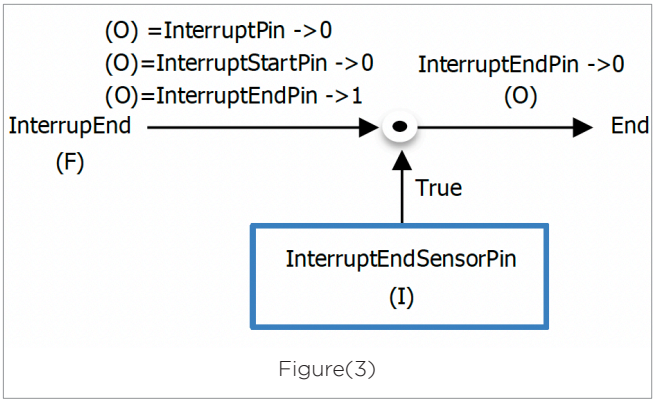
User Guide



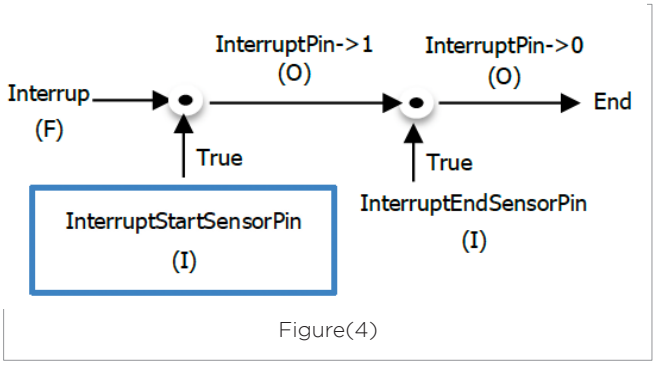
Figure(1)



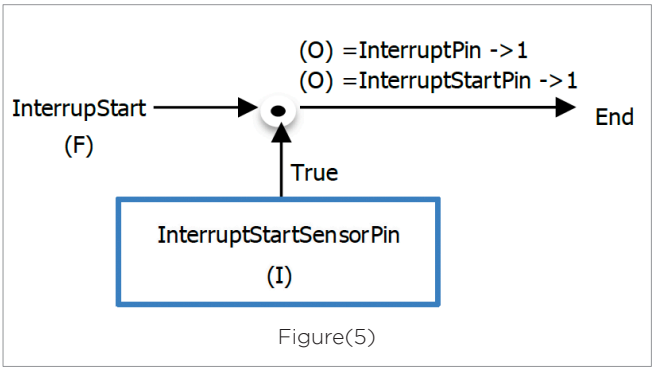
Figure(2)



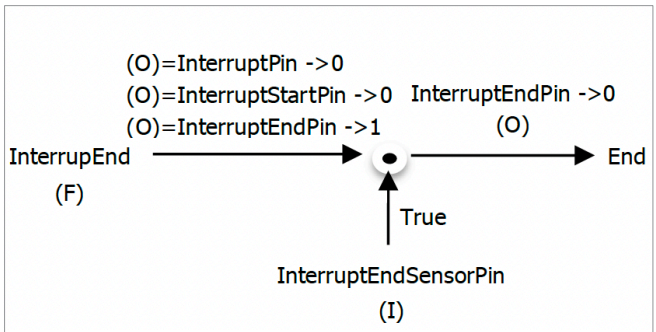
Figure(3)



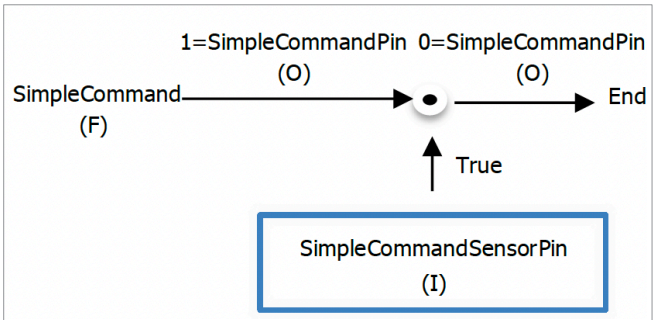
Figure(4)



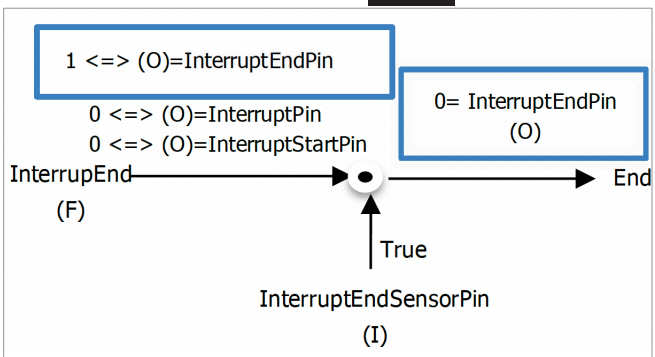
Figure(5)



Figure(6)

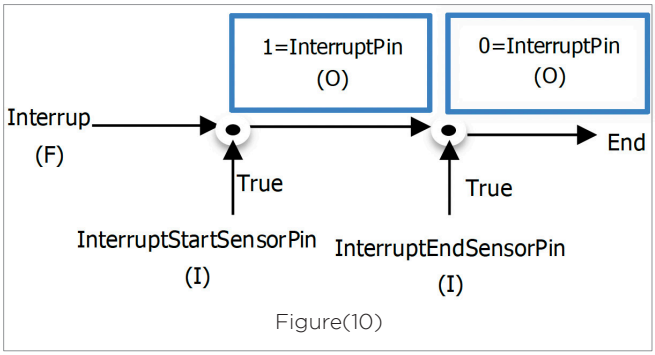


Figure(7)

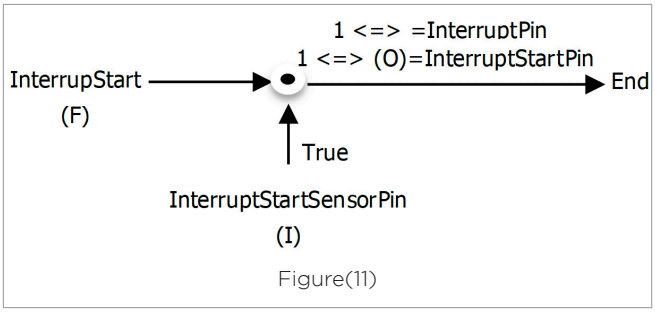


Figure(8)

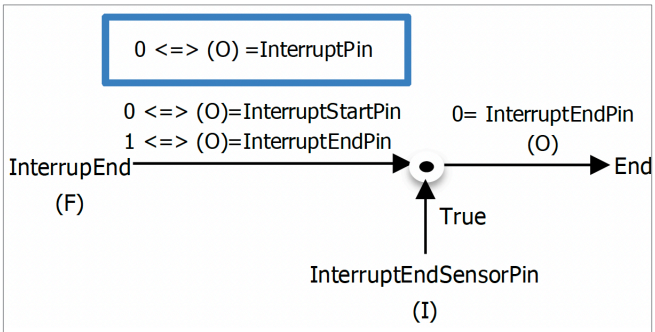
Figure(9)



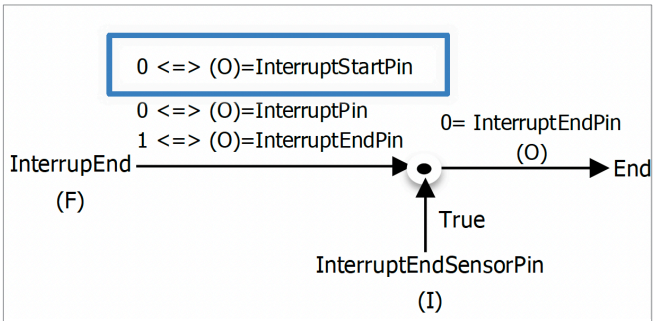
Figure(10)



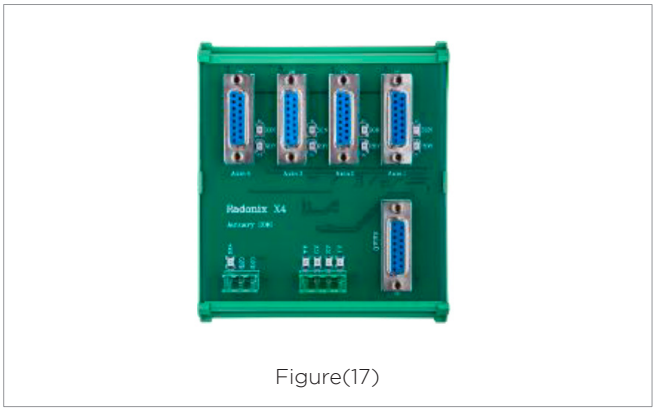
Figure(11)



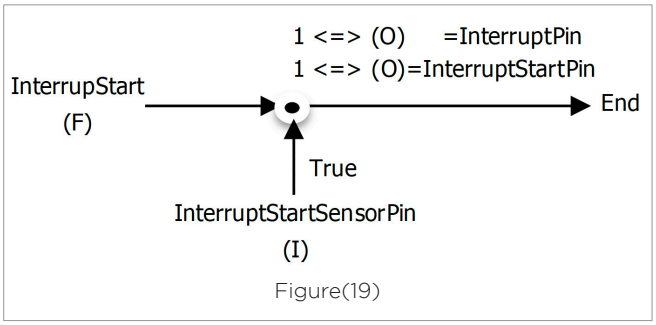
Figure(13)



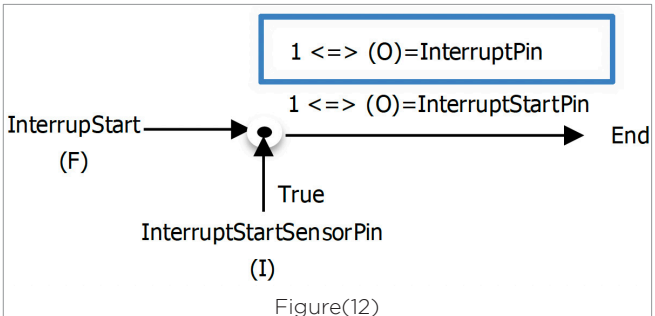
Figure(15)



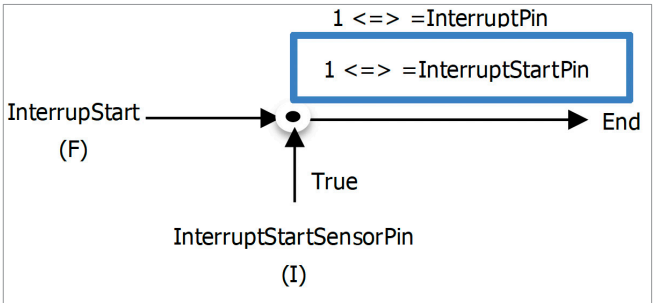
Figure(17)



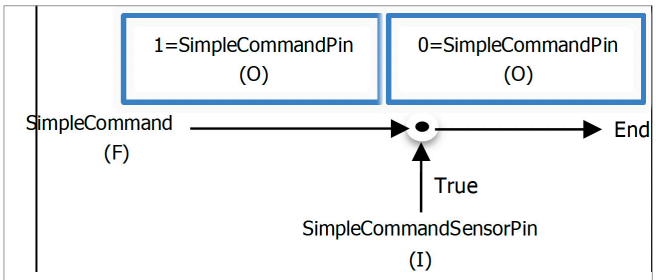
Figure(19)



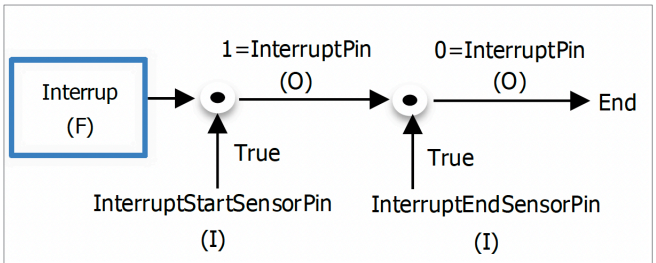
Figure(12)



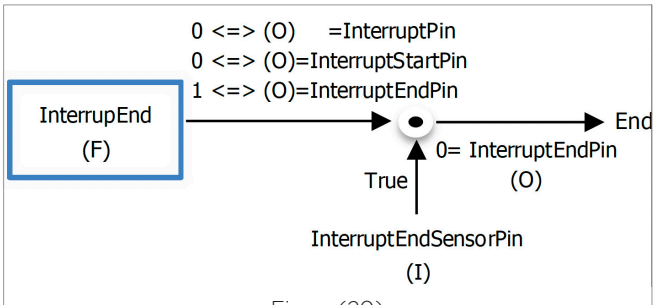
Figure(14)



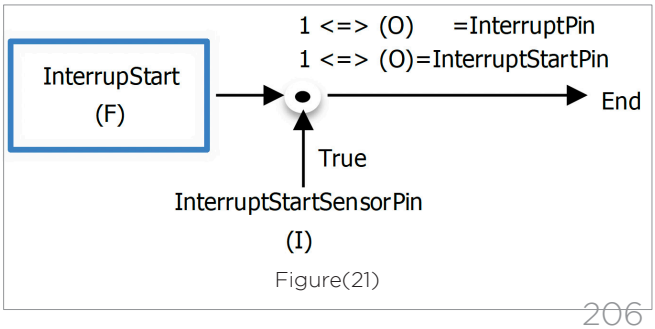
Figure(16)



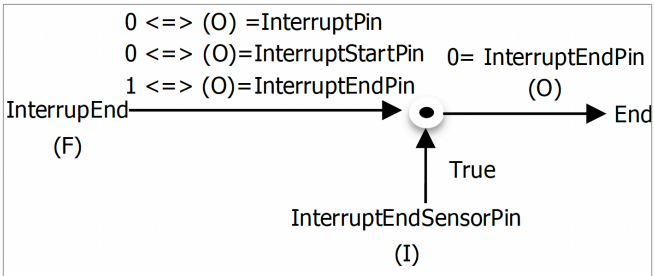
Figure(18)



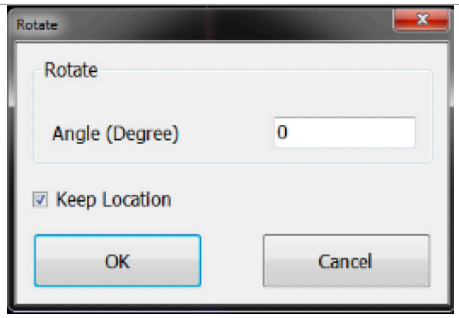
Figure(20)



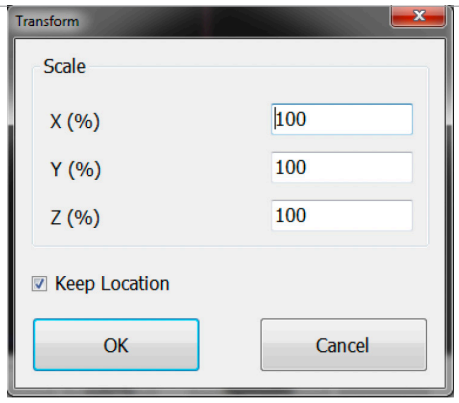
Figure(21)



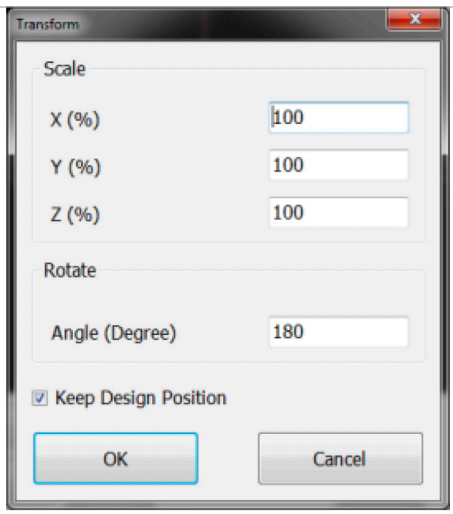
Figure(22)



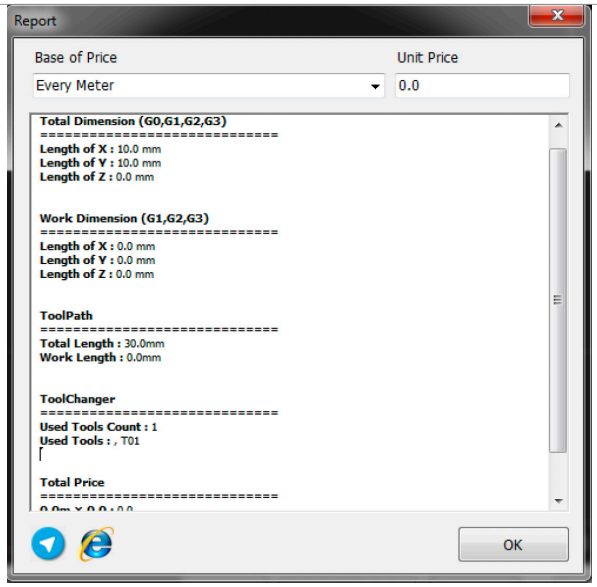
Figure(24)



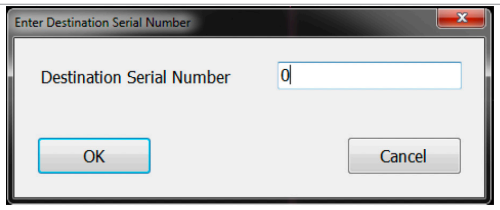
Figure(26)



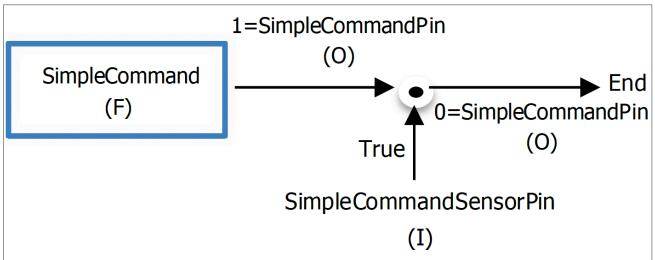
Figure(28)



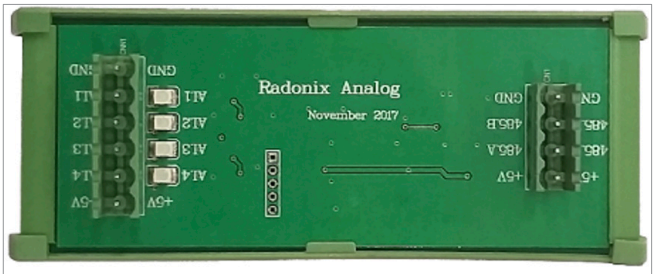
Figure(23)



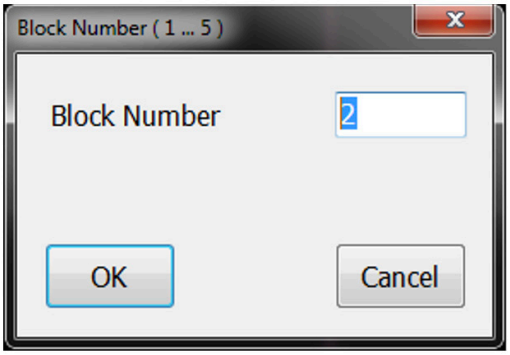
Figure(25)



Figure(27)



Figure(29)



Figure(30)