Servo Controller for X-Y Table

Author: Joaquín Juan Montalvo
Tutor: Geert Calu
Degree: Bachelor in Electrical Engineering
Institution: Katholieke Hogeschool VIVES Zuid
Department: Applied Engineering and Technology
Location: Kortrijk (Belgium)

30 May 2016

Dedication

I dedicate this final project to my parents Joaquín and Guadalupe and especially to my Sister Natalia for their support.

My thanks to VIVES for giving me this opportunity especially my tutor Mr. Geert Calu for sharing his knowledge and for his patience.

Acknowledgements

This final project, "**Servo Controller for X-Y Table**" has been realized by Joaquín Juan Montalvo, in Katholieke Hogeschool VIVES Zuid in Kortrijk (Belgium) as ERASMUS student. Execution period between the dates 01.09.2015 and 30.05.2016.

Mr. Geert Calu, professor of Industriéle Wetenschappen Technologie (W & T), has been the Tutor of the Thesis, in Vives Hogeschool.

Rafael Montoya Villena, professor and Academic Director of Electrical Engineering degree in Escuela Politécnica Superior de Alcoy (Spain), has been the Tutor of the Thesis, in Spain.

Abstract

The purpose of the final project "**Servo Controller for X-Y Table**" consist in performing of a ServoControl for two axes using SIGMATEK automation system.

For this project are available an XY table manufactured by VIVES University College for which has designed a control and visualization system that allows operations for different applications in industry.

For this purpose two servomotors YASKAWA from SGMP-01V314T series, two Servopacks OMRON from R88D-UP04V series, the ETV 0551 (Control Panel) and the modules CCP 521 (processor module) , CST 022 (used as a control for servo motors), CTO 163 (digital output module) and CDI 161 (digital input module) has been used. All modules belong to the C-DIAS series from SIGMATEK.

Also a printed circuit board has been designed to facilitate the sevopack connection with SIGMATEK modules.

An appendix is included in this project with the title **"Basic SIGMATEK Servo Control"**. The purpose of this appendix is provide the basics knowledge to build a basic servo control using SIGMATEK software and hardware.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Servo Systems

1.1 XY Table

An XY table is a surface with 2 perpendicular axes, anchored one on another. Each axes is moved by one servomotor. Both axes can be moved very accurately to a certain position, with XY coordinates, inside the table surface. These tables are used for verification PCBs, milling or cutting pieces in a pattern.

In Figure 1.1 is shown all the hardware that makes up the machine and the structure.

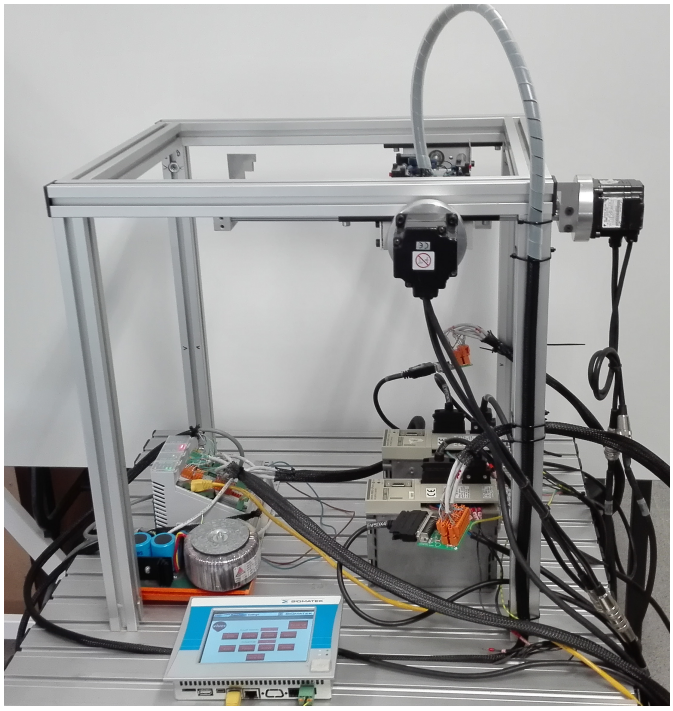


Figure 1.1: XY Table

1.2 Servo Control System

Servo System or **Servo Control System** is a mechanism that, detects the machine position (output data), feeds back the data to the input side, compares it with the specified position (input data), and moves the machine by the difference between the compared data.

The Figure 1.2 illustrates a close loop control for a Servo System.

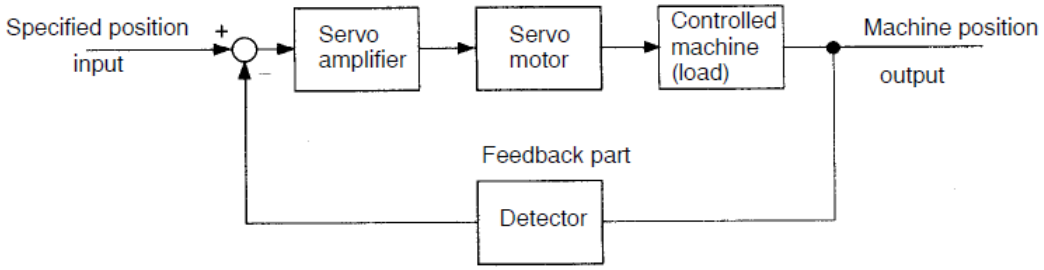


Figure 1.2: Close Loop Servo Control

A Servo Control System is composed by: **Host Controller** that send a position reference (pulse train) to the Servopack, **Servopack** (Comparator + Power Amplifier) that processes an error and operates the servomotor, **Servomotor** (motor + encoder (Position or Speed Detector)).

The Figure 1.7 illustrates a Servo control System configuration for position control.

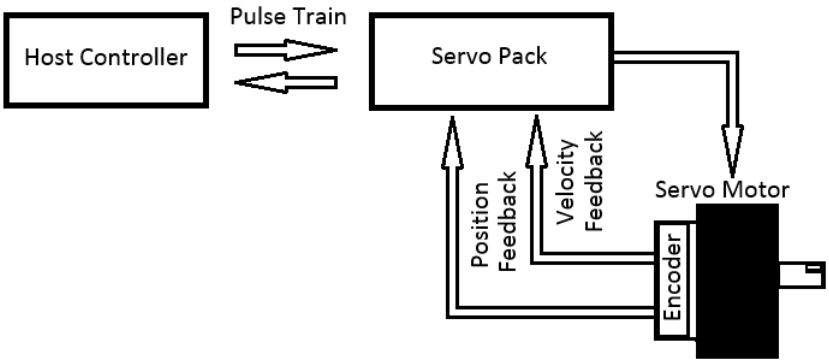


Figure 1.3: Servo Control System diagram

1.3 Host Controller

The host controller send a position reference (pulse train) to the Servopack to perform positioning or interpolation. This type of Servopack contains a position control loop.

SIGMATEK PLC belong to the **C-DIAS** series is used as a Host Controller in this project.

The **CST 022** module send a position reference (pulse train) to the Servopack (OMRON R88D-UP04V) by the **Power Stack Control** and receives a position reference (pulse train) from the Servopack by the **Incremental Encoder Input**.

User constants can be used to select either of the following pulse trains:

1.3.1 Power Stack Control signals

1. **Pulse with a 90°phase shift (Mode 0 = Mode 2).**

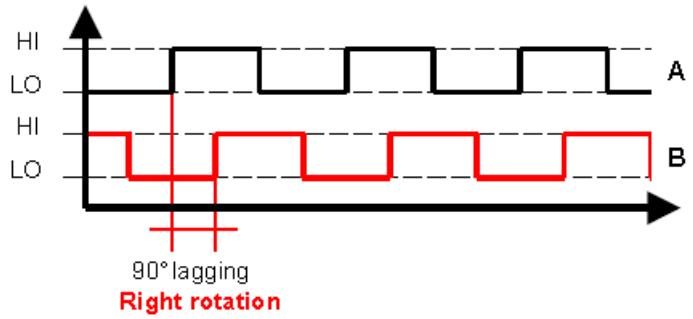


Figure 1.4: Pulses for Right Rotation

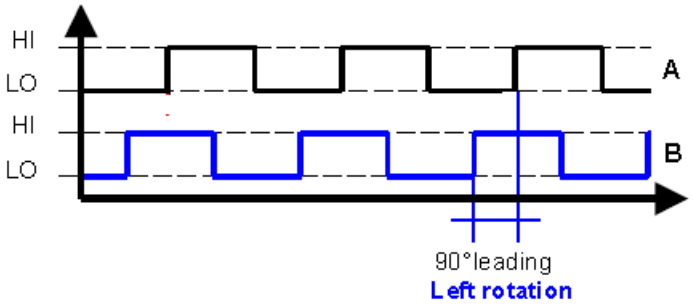


Figure 1.5: Pulses for Left Rotation

The two output wave forms are 90°out of phase. These signals are decoded to produce a count up pulse or a count down pulse.

2. **CW (clockwise) / CCW (counterclockwise) Pulse Command (Mode 1)** .

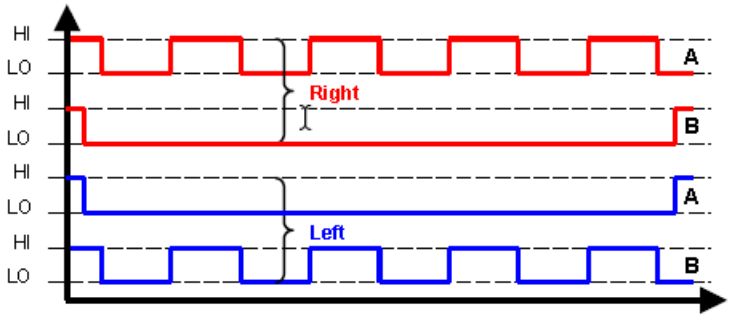


Figure 1.6: Pulses for CW / CCW Pulse Command

On mode 1 there are pulses only in one channel the other channel remains at low level. When the channel B remains at low level the direction of movement is clockwise. When the channel A remains at low level the direction of movement is counterclockwise.

3. **Pulse / Sign Command (Mode 3)**.

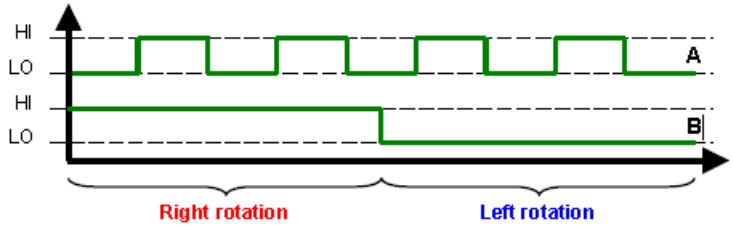


Figure 1.7: Pulses for Pulse / Sign Command

On mode 3 the direction of movement depends on the level of channel B (high level = right movement, low level= left movement)

On this project Pulse with a 90°phase shift (Mode 0 = Mode 2) is used for the host controller to send a position reference.

1.3.2 Incremental Encoder Input signals

The host controller receives a pulse train (position information) from the Servopack and uses it to monitor the position.

For more information about Incremental Encoder signals see section 1.5.1

1.4 Servopack

A Servopack consist of **Servo Amplifier** that processes an error signal to correct the difference between a reference (send by Host Controller) and feedback data (send by Host Servo Motor) and a **Power Amplifier**, which operates the servomotor accordingly. A servo amplifier is required to operate an AC servomotors. An Omron ServoPack belong R88D-UP04V series is used in this project.

1. **Power Amplifier**

   A power amplifier runs the servomotor at a speed or torque proportional to the output of the comparator. In other words, from the commercial power supply of 50/60 Hz, it generates alternating current with a frequency proportional to the reference speed and runs the servomotor with this current.

2. **Comparator**

   A comparator consists of a **Comparison Function** and a **Control Function**.

   - The **Comparison Function** compares reference input (position or speed) with a feedback signal and generates a differential signal.
   - The **Control Function** amplifies and transforms the differential signal. In other words, it performs proportional (P) control or proportional/integral (PI) control.

1.5 Servomotor

AC servomotors are AC motors that incorporate encoders. This motors providing feedback to operate in closed-loop control so that can be positioned to high accuracy.

For a synchronous servomotors, motor speed is controlled by changing the frequency of alternating current this type of servomotors are ideal when precise positioning is required.

A servomotor must have "instantaneous power" so that it can start as soon as a start reference is received. The term "power rating (kW/s)" is used to represent instantaneous power. It refers to the electric power (kW) that a servomotor generates per second. The greater the power rating, the more powerful the servomotor.

The following Figure 1.8 illustrates the structure of a synchronous servomotor:

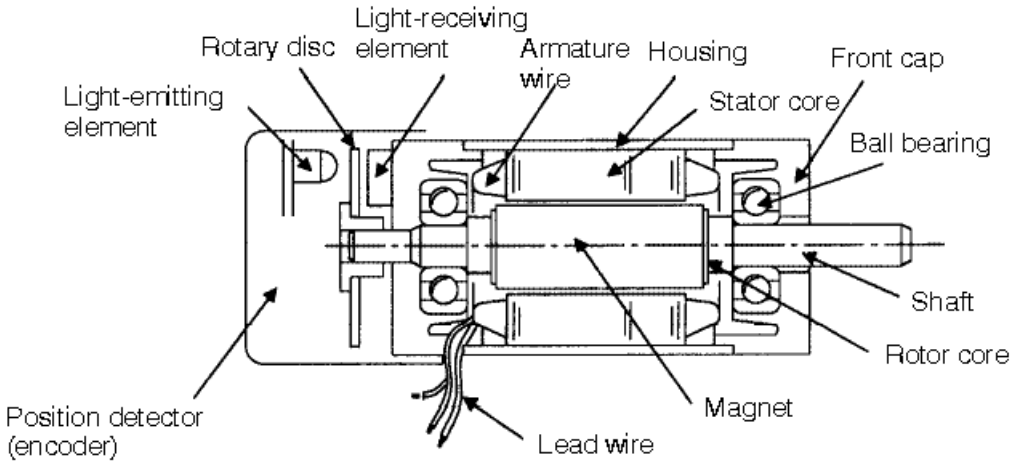


Figure 1.8: Synchronous Servomotor Structure

1.5.1 Encoder

A servo System requires a position or speed detector. It uses an encoder mounted on a servomotor for this purpose.

A rotary encoder, also called a shaft encoder, is an electro-mechanical device that converts the angular position or motion of a shaft to an analog or digital code.

An incremental encoder outputs incremental changes from a pre-defined home position. As a result, an incremental encoder requires additional electronics (typically a PLC) to count pulses and convert the data into speed or motion.

The encoder consists of an electronic circuit in which a diode laser emits a beam which is interrupted by the grooves of a metal sheet, resulting in an intermittent signal.

The following Figure 1.9 illustrates the encoder structure:

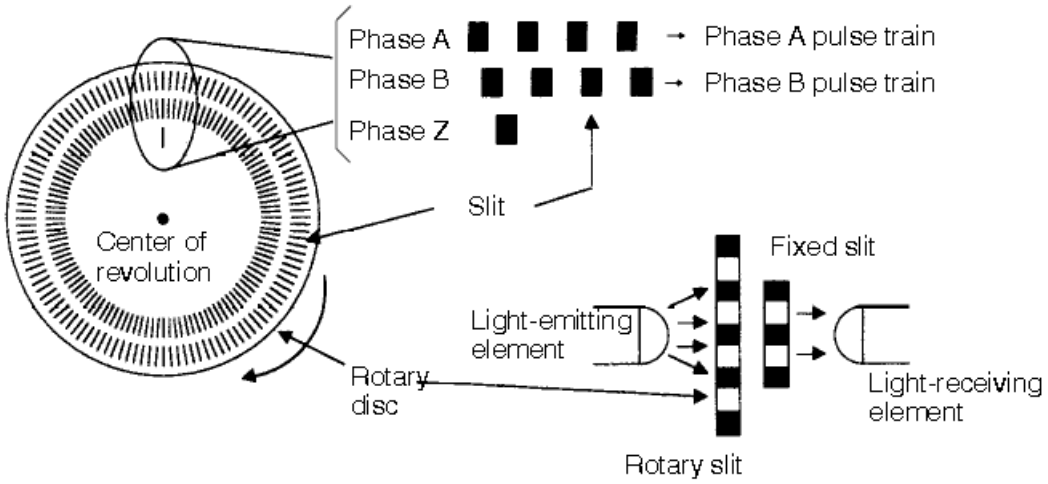


Figure 1.9: Encoder Structure

When an incremental encoder moves, it generates a stream of binary pulses proportional to the rotation of the shaft (rotary encoder). However, this encoder does not detect an absolute position and merely outputs a pulse train. Hence zero return operation must be performed before positioning. The incremental encoder provides two square waveforms with 90 °electrical phase shift , usually Channel A and Channel B. Reading a single channel (channel A) information of the rotation speed is obtained. whether the channel signal B is also read is possible to know the rotation direction based on the sequence data produced by both signals Incremental Encoder also have a “R” or ”Z” signal. Once every rotation, this ”R” signal is rising on exactly the same position.This signal is represented as a square pulse with phase

and amplitude centered on channel A. This "R" signal can be used as an accurate reference point. The additional differential signals, called "/A", "/B" and "/R" are inverted "A", "B" and "R" signals. Controllers can compare each pair ("A" must be equal to inverted "/A") to ensure that there is no error during the transmission. An incremental encoder is incapable of follow the motor shaft position if system power is lost and some motion occurs during that period of time.

The following figures(Figure 3.2, Figure 1.11 & Figure 1.12) illustrates the Incremental Encoder signals:

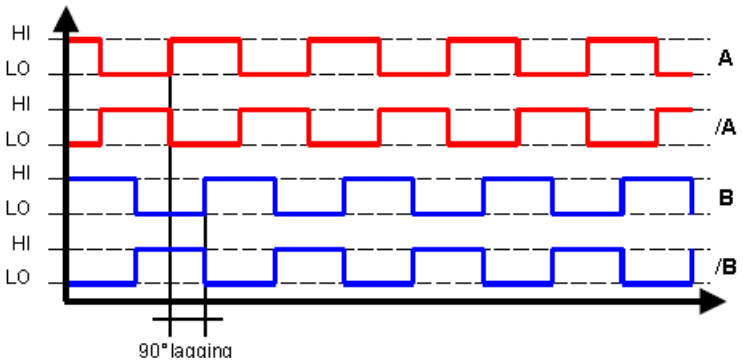


Figure 1.10: Count Up signal

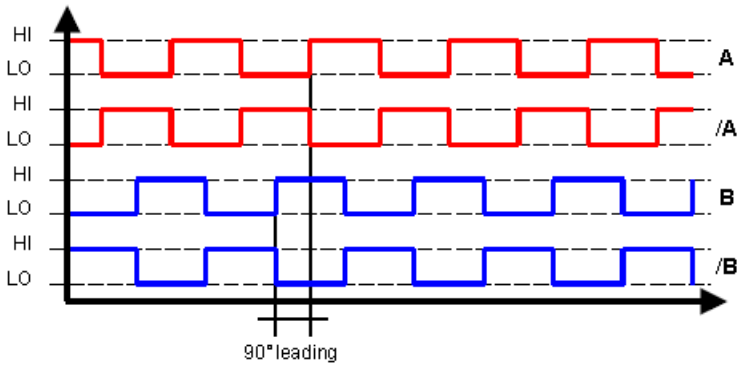


Figure 1.11: Count Down signal

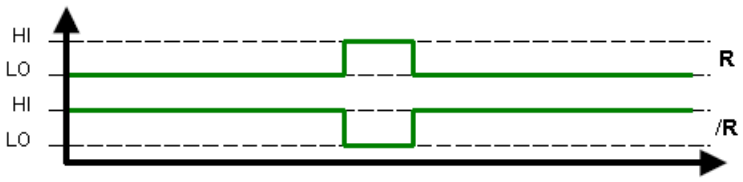


Figure 1.12: Reference Pulse(Zero Position)

Chapter 2

Host Controller

The Host Controller used in this project consists of following modules:

1. **ETV 0551** (Control Panel).
2. **CIV 512** (VARAN Control Module)
3. **CST 022** (Control for Servo Motors).
4. **CTO 163** (Digital Output Module).
5. **CDI 161** (Digital Input Module)
6. **CMB 082** (C-DIAS Module Carrier)

2.1 C-DIAS Control Panel VARAN ETV 0551

Figure 2.1: Control Panel VARAN ETV 0551

The control panel is an intelligent terminal for programming and visualization of automated processes. A touch-screen serves as the input medium for process data and parameters; the output is shown on a 5.7“ VGA TFT color display.

The interface connections are used to exchange process data and configure the terminal. A micro SD card serves as the storage medium for the operating system, application and application data. The integrated VARAN manager operate for communication modules (VARAN is based on standard Ethernet physics and the protocol is implemented in the hardware).The Ethernet interface is used as the online interface.

2.1.1 Technical Data

In the following tables some ETV 0551 features are shown. For more information see the product information [6].

Table 2.1: **ETV 0551 Technical Data**

Performance Data	
Processor	EDGE Technology X86 compatible
Internal cache	32-kbyte L1 cache 256-kbyte L2 cache
BIOS	AMI
Internal program and data memory (DDR2 RAM)	64-Mbyte
Internal remnant data memory	512-kbyte
Internal storage device (IDE)	microSD card
Interfaces	2x USB Type A 2.0 (full speed 12 Mbit/s) 1x USB Type Mini B 1.1 1x Ethernet 1x VARAN Out (Manager) 1x CAN bus
Internal interface connections and devices	1x TFT LCD color display 1x Touch
Control panel	4-wire touch screen (analog resistive)
Display	5.7” TFT color display. 640 x 480 Pixel
Data buffer	yes
Real-time clock	yes
Dimensions	180mm/135mm/40.4mm (W x H x D)

Table 2.2: **ETV 0551 Electrical Requirements**

Electrical Requirements	
Supply voltage	typically +24 V DC minimum +18 V DC maximum +30 V DC
Current consumption Power supply +24 V	typically 335 mA (without externally connected devices) maximum 610 mA (with externally connected devices)
Inrush current	maximum 28 A for 20 μ s

Table 2.3: **ETV 0551 Digital Outputs**

Digital outputs	
Number	8
Short-circuit proof	yes
Maximum continuous current load allowed per channel	2A
Maximum total current (all 8 channels)	6 A (100% of on time)
Voltage drop over power supply (output active)	$\leqslant$ 1V
Residual current (output inactive)	$\leqslant$ 12 μ A
Digital Inputs	
Number	8
Input voltage	typically +24 V maximum +30 V
Signal level	low: ¡ +4.5 V high: ¿ +14 V
Switching threshold	typically +11 V
Input current	typically 5 mA at +24 V
Input delay	typically 5 ms

2.1.2 Connections

The Control Panel ETV 00551 is connected over the Ethernet bus system VARAN-Out (X5) to the CIV 512 Interface Module (VARAN-Input(X1)). Through The Ethernet plug (X4) the Control Panel is connected to the Personal Computer. Connector (X1) should be connected to the power supply (+24V DC). Into connector (X8) should be introduced the microSD card with the operating system installed.

In Figure 2.2 the connector layout used in this project are shown.

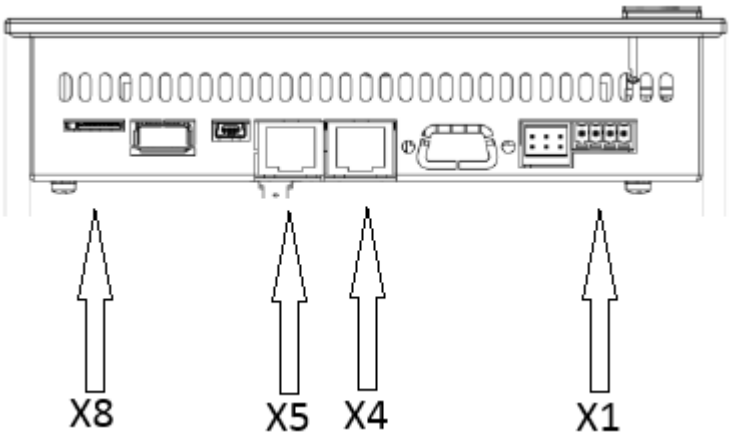


Figure 2.2: ETV 0551 connectors

Table 2.4: **ETV 00551 Connections**

Connections						
ETV 0551 Plug	**Pin**	**Function**	**Device Connected**	**Plug**	**Pin**	**Function**
X1	2	+24V Input	Power Supply	+24V	—	+24V Output
	3	GND	Power Supply	GND	—	Ground
X4	—	Ethernet	PC	RJ-45	—	Ethernet
X5	—	VARAN-Out	CIV 512	X1		VARAN-Input
X8	—	Storage	microSD Card	—	—	OS

Table 2.5: **ETV 00551 Cable Specifications**

Cable Specifications			
ETV 0551 Plug	**Cable №** №	**Type**	**Wire Colour**
X1-2	7	Cu/1(mm^2)/PVC	Red
X1-3	7	Cu/1(mm^2)/PVC	Black
X4	10	ETHERNET Cable RJ45/UTP/CAT6	—
X5	11	VARAN Cable flexible system cable, 4-pin, 2x2xAWG22/7, double shielding	—
X8	—	—	—

2.2 C-DIAS VARAN Control Module CIV 512

Figure 2.3: VARAN Control Module CIV 512

The C-DIAS CIV 512 module serves as the power supply and con-nection for decentralized C-DIAS module groups with a CPU over the VARAN bus.

A module group consists of a module carrier and the C-DIAS mod-ules mounted on it.

Depending on the module carrier, up to 8 mod-ules can be mounted.

The VARAN-Out port allows the construction of the VARAN bus in a line structure.

The VARAN-Out port has automatic Ethernet recognition. If the VARAN-Out is connected to an Ethernet participant, it is automati-cally changed to an Ethernet port.

Incoming Ethernet packets are, similar to using a HUB, distributed to all other Ethernet ports in the VARAN bus system and the VARAN manager (and there-with the CPU) with VtE.

2.2.1 Technical Data

In the following tables some CIV 512 features are shown. For more information see the product information [3]

Table 2.6: **CIV 512 Performance data**

Performance data	
Performance data	1 x VARAN-In (RJ45) 1 x VARAN-Out (Optional Ethernet (VtE)) (RJ45) (maximum length: 100 m)

Table 2.7: **CIV 512 Electrical requirements**

Electrical requirements	
Voltage supply	18 – 30 V DC
Current consumption of power supply	The current consumption depends on the connected load (max. 1.7 A)
C-DIAS bus supply	Through the CIV 512
Current load on the C-DIAS bus (I/O/P module supply)	+5V / +24V
	Maximum 1.2 A

2.2.2 Connections

The CIV 512 module is connected over the Ethernet bus system VARAN-In (X1) to the Control Panel ETV 00551 via VARAN-Output (X5).

Connector (X3) should be connected to the power supply (+24V DC).

In Figure 2.4 the CIV 512 module connector layout is shown.

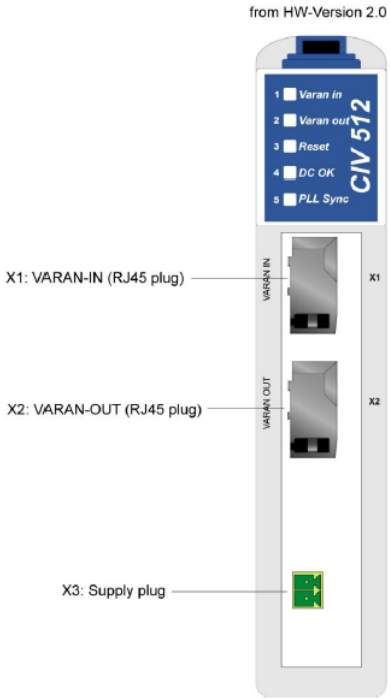


Figure 2.4: CIV 512 Connections

Table 2.8: **CIV 512 Connections**

Connections						
CIV 512 Plug	**Pin №**	**Function**	**Device Connected**	**Plug**	**Pin №**	**Function**
X1	—	VARAN Input	ETV 00551	X5	—	VARAN Output
X3	1	+24V In	Power Supply	+24V	—	+24V Out
X3	2	Gnd	Power Supply	GND	—	Ground

Table 2.9: **CIV 512 Cable Specifications**

Cable Specifications				
Plug	**Pin №**	**Cable №**	**Type**	**Wire Colour**
X1	—	11	VARAN Cable flexible system cable, 4-pin, 2x2xAWG22/7, double shielding	—
X3	1	6	UTP 8x	Orange
X3	2	6	UTP 8x	White Orange

2.3 C-DIAS Steepeer Module CST 022

Figure 2.5: Module CST 022

The CST 022 module is used as a control for the two Servo motors. One module can control two motors from two independent channels.

An RS 422 connection with the corresponding signal level servers as the output for the control.

An incremental encoder connection with A/B/R analysis and the corresponding +5V incremental encoder supply is available for each channel.

The digital inputs are designed for the reference path and monitoring the end position of Step motors control. For servo motors control this inputs are not used.

In addition, the CST 022 has three digital differential outputs per channel. This Outputs are not used in this protect.

2.3.1 Technical Data

In the following tables some CST 022 features are shown. For more information see the product information [4]

Table 2.10: **CST 022 Incremental Encoder Input**

Incremental Encoder Input	
Number of Channels	2
Input Signals	Incremental encoder signal (A, /A, B, /B, R, /R) RS422–level 150W termination 330W spreading
Input Frequency	Max. 125kHz
Count frequency	Max. 500kHz
Signal analysis	4x
Counter frequency	16Bit
Encoder supply	+5V / 0,5A short circuit protected (per channel)
Status display (A and B Signals)	green LEDs

Table 2.11: **CST 022 Power Stack Control Outputs**

Power Stack Control Outputs	
Number of channels	2
Output signals	Control signals (A, /A, B, /B, C, /C) RS422 level
Output frequency	Max. 500kHz
Max. allowable continuous current.	40mA
Status display	Yellow LEDs

Table 2.12: **CST 022 Electrical requirements**

Electrical requirements	
Supply voltage +24 V1	18 – 30V DC
Current consumption Voltage supply +24 V1	¡350mA (incl. Supply of the Incremental encoder)
Supply from C-DIAS-Bus	+5V
Current consumption of CDIAS- Bus (+5 V-supply)	Typically 400mA /Maximum 480mA
Status display	None

Figure 2.6: CST 022 Status Displays

Table 2.13: **CST 022 Status Display**

Status Display Channel 1		
LED №	LED Colour	Description
1	YELLOW	Output „RS422 – A“
2	YELLOW	Output „RS422 – C“
3	YELLOW	Output „RS422 – B“
4	GREEN	Input „INC – A“
5	GREEN	Input „INC – B“
6	GREEN	Input 1
7	GREEN	Input 2
8	YELLOW	Output 1
9	YELLOW	Output 2
10	YELLOW	Output 3

Table 2.14: **CST 022 Status Display**

Status Display Channel 2		
LED №	LED Colour	Description
11	YELLOW	Output „RS422 – A“
12	YELLOW	Output „RS422 – C“
13	YELLOW	Output „RS422 – B“
14	GREEN	Input „INC – A“
15	GREEN	Input „INC – B“
16	GREEN	Input 1
17	GREEN	Input 2
18	YELLOW	Output 1
19	YELLOW	Output 2
20	YELLOW	Output 3

2.3.2 Connections

Plug **X1** (Incremental Encoder Input) and **X2** (Power Stack Control) are connected over the **Printed Circuit Board** (PCB) to the **ServoPack** connector **CN1**. The Plug X7 is not connected because the Digital Outputs are not used in this module.

In Figure 2.7 the CST 022 terminal assignment are shown.

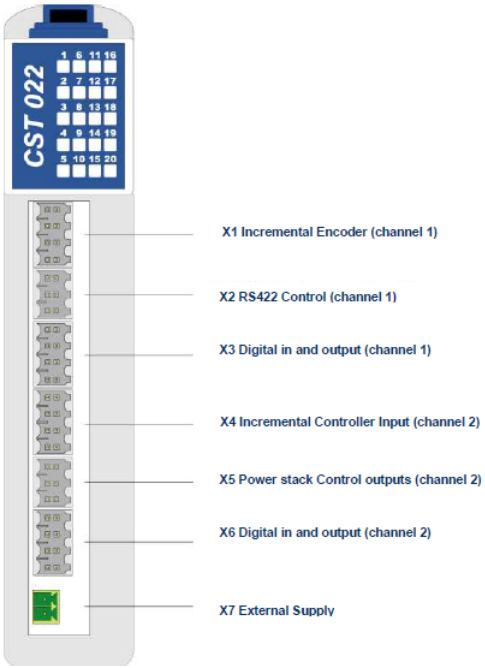


Figure 2.7: Pin assignment Module CST 022

Table 2.15: **CST 022 Connections**

Connections						
CST 022 Plug	**Pin №**	**Function**	**Device Connected**	**Plug**	**Pin №**	**Function**
X1	1	Incremental Encoder Input (A-)	PCB	X2	3	A-
X1	2	Incremental Encoder Input (A+)	PCB	X2	2	A+
X1	3	Incremental Encoder Input (B-)	PCB	X2	4	B-
X1	4	Incremental Encoder Input (B+)	PCB	X2	5	B+
X1	5	Incremental Encoder Input (R-)	PCB	X2	7	R-
X1	6	Incremental Encoder Input (R+)	PCB	X2	6	R+
X1	7	Incremental Encoder Input (GND)	PCB	X2	1	GND
X2	1	Power Stack Control (A-)	PCB	X1	9	A-
X2	2	Power Stack Control (A+)	PCB	X1	10	A+
X2	3	Power Stack Control (B-)	PCB	X1	7	B-
X2	4	Power Stack Control (B+)	PCB	X1	8	B+

Table 2.16: **CST 022 Cable Specifications**

Cable Specifications				
Plug	**Pin №**	**Cable №**	**Type**	**Wire Colour**
X1	1	1	UTP 8x	White Blue
X1	2	1	UTP 8x	Blue
X1	3	1	UTP 8x	White Green
X1	4	1	UTP 8x	Green
X1	5	1	UTP 8x	White Brown
X1	6	1	UTP 8x	Brown
X1	7	1	UTP 8x	White Orange
X2	1	2	UTP 8x	Green
X2	2	2	UTP 8x	Blue
X2	3	2	UTP 8x	Orange
X2	4	2	UTP 8x	Brown

2.4 C-DIAS Digital Output Module CTO 163

Figure 2.8: Module CTO 163

The module CTO 163 has 16 digital outputs +24V/ 2A (switching positive). These outputs are protected against short-circuiting. The supply voltage of each channel group is checked on undervoltage.

2.4.1 Technical Data

In the following tables some CTO 163 features are shown. For more information see the product information [5]

Table 2.17: **CTO 163 Digital Outputs**

Digital Outputs	
Number of outputs	16
Protection against short circuiting	10
Maximum permissible constant current / channel	2 A
Maximum total current (per 4 channels)	6A (100% operating time)
Maximum total current (whole module)	24A (100% operating time)
Maximum breaking energy of the output (inductive load)	With additional internal protective circuit (HW V2.0 or higher) Maximum 0,65 Joule / channel maximum 1,95 Joule / 4 channels
Voltage drop across the supply (output switched on)	$\leqslant$ 1V
Residual current (output switched off) Switch-on delay time	$\leqslant 12\mu$ A $<200\mu$s
Switch-off delay time	$<200\mu$s
Status display	Optional (LEDs: outputs yellow, voltage surveillance red)

Table 2.18: **CTO 163 Electrical requirements**

Electrical requirements	
Supply voltage +24V / 1-4	18 – 30V DC
Supply voltage current consumption +24V / 1-4	Corresponds to the load on the digital outputs (max. 6A/ group of 4)
Supply of the C-DIAS bus	+5V
Current consumption on the C-DIAS bus (+5V supply)	Typically 5mA / Maximum 20mA

Table 2.19: **CTO 163 Voltage surveillance**

Voltage surveillance	
Supply voltage +24V / 1-4	Supply voltage <18V (Error-LED illuminates red)

Figure 2.9: CTO 163 Status Displays

Table 2.20: **CTO 163 Status Displays**

Status Displays		
LED №	**Led Colour**	**Meaning**
1 – 16	yellow	Outputs 1 – 16
G1 – G4	red	Error voltage surveillance

2.4.2 Connections

The Outputs (**X1 & X2**) are connected over the **Printed Circuit Board** (PCB) to the **ServoPack** connector **CN1**. The Plug **X5** is the Plus supply.

In Figure 2.10 the CTO 163 terminal assignment are shown.

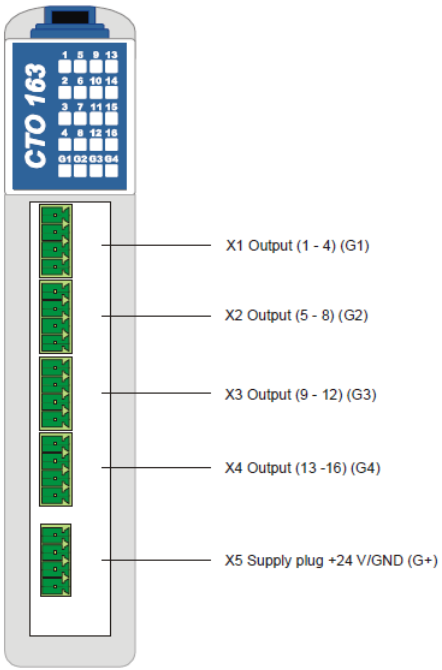


Figure 2.10: Terminal assignment Module CTO 163

Table 2.21: **CTO 163 Connections**

Connections						
CTO 163 Plug	**Pin №**	**Function**	**Device Connected**	**Plug**	**Pin №**	**Function**
X1	1	Output 1	PCB	X1	1	RUN
X2	1	Output 5	PCB	X2	10	POT
X2	3	Output 7	PCB	X2	9	NOT
X5	1	+24V/1 (output 1-4)	Power Supply	+24V	—	+24V Output
X5	2	+24V/1 (output 5-8)	Power Supply	+24V	—	+24V Output

Table 2.22: **CTO 163 Cable Specifications**

Cable Specifications				
Plug	**Pin №**	**Cable №**	**Type**	**Wire Colour**
X1	1	4	UTP 8x	Blue
X2	1	4	UTP 8x	Green
X2	3	4	UTP 8x	Brown
X5	1	6	UTP 8x	Blue
X5	2	6	UTP 8x	Brown

2.5 C-DIAS Digital Input Module CDI 161

The CDI 161 is equipped with 16 inputs and a +24V level for reading the signal states “0“ and “1“. Input filters are available to suppress noise impulses occurring in the signal leads.

Figure 2.11: Module CDI-161

2.5.1 Technical Data

In the following tables some CDI 161 features are shown. For more information see the product information [2]

Table 2.23: **CDI 161 Digital Inputs**

Digital Inputs	
Number of inputs	16
Input voltage	Typically +24V / Maximum +30V
Signal levels	Low: <+8V / High: >+14V
Switching threshold	Typically +11V
Input current	5mA at +24V
Input delay	Typically 5ms
Status display	Optional

Table 2.24: **CDI 163 Electrical requirements**

Electrical requirements	
Supply of the C-DIAS bus	+5V
Current consumption on the C-DIAS bus (+5V supply)	Typically 1mA / Maximum 3mA

Figure 2.12: Module CDI-161 Status display

Table 2.25: **CDI Status Displays**

Status Displays		
LED №	**Led Colour**	**Meaning**
1 – 16	green	Inputs 1 – 16
G1 – G4	—	not used

2.5.2 Connections

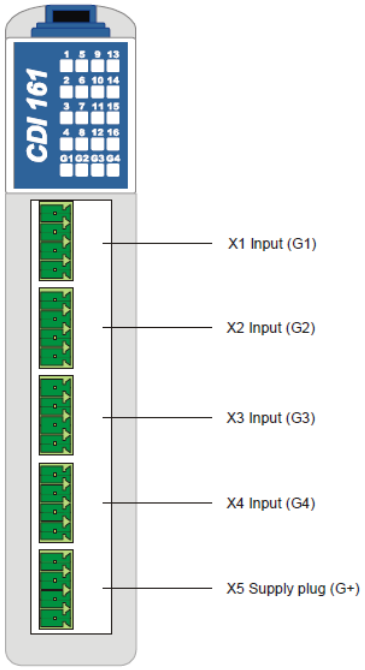


Figure 2.13: Pin assignment Module CDI-161

Table 2.26: **CDI 161 Connections**

Connections						
CDI 161 Plug	**Pin №**	**Function**	**Device Connected**	**Plug**	**Pin №**	**Function**
X1	1	Input 1	Micro-Switch	N.O.	—	Reference Switch
X1	3	Input 3	PCB	X1	2	$\overline{ALM}$
X2	1	Input 5	Micro-Switch	N.O.	—	End Switch Forward
X2	3	Input 7	Micro-Switch	N.O.	—	End Switch Reverse
X3	1	Input 9	PCB	X1	5	INP
X3	3	Input 11	PCB	X1	4	TGON
X4	1	Input 13	PCB	X1	6	BKIR

Table 2.27: **CDI 161 Cable Specifications**

Cable Specifications				
Plug	**Pin №**	**Cable №**	**Type**	**Wire Colour**
X1	1	8	UTP 8x	Orange
X1	3	3	UTP 8x	White Blue
X2	1	8	UTP 8x	Green
X2	3	8	UTP 8x	White-Green
X3	1	3	UTP 8x	White Green
X3	3	3	UTP 8x	White Brown
X4	1	3	UTP 8x	White Orange

2.6 C-DIAS Module Carrier CMB 082

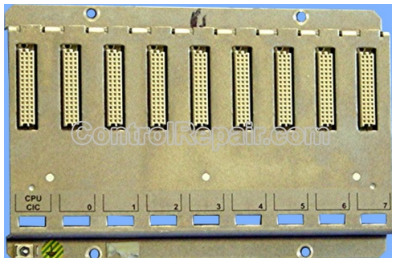


Figure 2.14: Carrier CMB 082

Each module carrier consists of an aluminium carrier profile for mechanically connecting the modules and a bus board for the electrical connection.

The bus board is located on the rear of the module carrier and consists of bus lines over which data is exchanged (periphery bus).

To mount, the modules are hooked on to the bottom of the carrier and simply snapped at the top.

The slots are, starting with the directly next to the control module or central unit, number in ascending order from left to right.

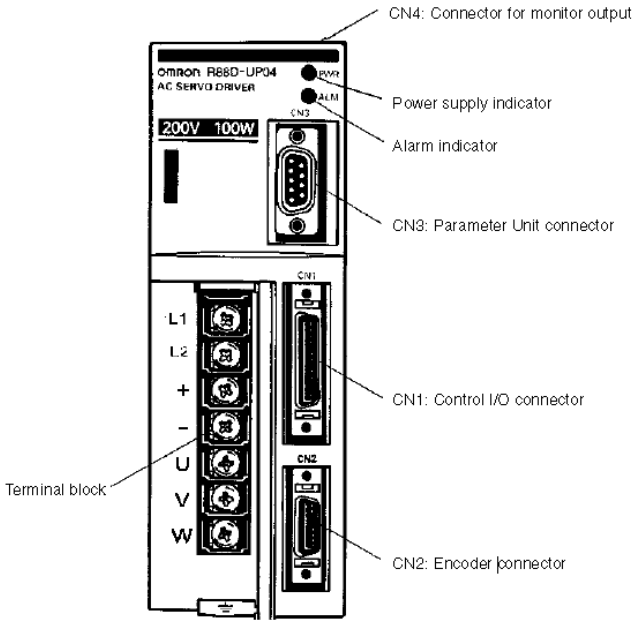
Chapter 3

Servopack & Servomotor

3.1 ServoPack OMRON R88D-UP04V

For this project an OMRON Servopack belong to the R88D-UP04V series as shown in Figure 3.1a is used.

(a) OMRON R88-UP04V ServoPack



(b) OMRON R88-UP04V Terminals

Control Functions

Any one of the following 4 control modes can be selected in the parameter settings.

For this project 90°differential phase (A/B phases) signals (mode 1) is used.

This mode has been chosen because has greater number of pulses per revolution 8192 Pulses/Rev (2048 Pulses x 4(Cn-02 Parameter)). This pulses correspond to a linear displacement of 5mm per motor revolution getting a very accurate position control (moving unit = 0.61x10-3 mm) (see subsection 5.1.3 and **??**)

1. **Position Control (Factory Setting)**

   Controls the position and speed of the Servomotor very precisely with pulse-train input signals. Any one of the following 3 pulse trains can be selected:

   (a) 90°differential phase (A/B phases) signals

   (b) Feed pulses/Directional signals

   (c) Forward/Reverse Pulses

2. **Position Control with Pulse Stop Input Enabled**

   Turning ON the Pulse Stop Input (IPG) prevents the control signals from being read by the Unit during position control.

3. **Internal Speed Control Settings**

   The speed of the motor is controlled with the three speeds (No. 1, No. 2, and No. 3 internal speed settings) set in the parameters. This mode is effective for simple position control or speedswitching operation.

4. **Internal Speed Control Setting + Position Control**

   Speed control can be performed with the internal speed settings and position control can be performed with pulse-train inputs.

3.1.1 ServoPack Specifications

Table 3.1: **General Specifications**

General Specifications	
Item	**Features**
Continuous output current (0-P)	1.2 A
Momentary max. output current (0-P)	4.0 A
Input power supply	Single-phase 200/230 VAC (170 to 253 V) 50/60 Hz
Control method	All-digital servo
Speed feedback	Optical encoder, 2,048 pulses/revolution
Applicable load inertia	Maximum of 30 times motor's rotor inertia
Inverter method	PWM method based on IGBT
PWM frequency	11 kHz
Applicable Servomotor wattage	100 W
Cable length between motor and driver	20 m max.
Weight (approximate)	0.9 kg
Heating value	20 W
Maximum pulse frequency	200 kpps
Position loop gain	0 to 500 (1/s)
Electronic gear	Electronic gear ratio setting range: $0.01 \leqslant (G1/G2) \leqslant 100$ (G1, G2 = 1 to 65,535)
Positioning completed range	0 to 250 command units
Feed-forward compensation	0 % to 100% of speed command amount (pulse frequency)
Bias setting	0 to 450 r/min
Position acceleration/deceleration time constant	0 to 64.0 ms (The same setting is used for acceleration and deceleration.)
External regeneration processing	Required for regeneration of more than 30 times the motor's rotor inertia.

The input pulse width must meet the following conditions:

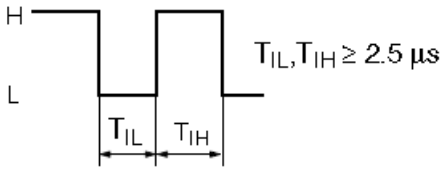


Figure 3.2: Input Pulses Condition

Table 3.2: **Input signals specifications**

Input signals	
Item	**Features**
Position command pulse input	TTL, line driver input with photoisolation, input current: 6 mA at 3 V Feed pulse and direction signal, forward pulse and reverse pulse, or 90°differential phase (A and B phases) signal (set via parameter). Pulse width: See note.
Deviation counter reset	TTL, line driver input with photoisolation, input current: 6 mA at 3 V
Sequence input	24-VDC, 5-mA photocoupler input, external power supply: 24±1 VDC, 50 mA min.

Table 3.3: **Output signals specifications**

Output signals	
Item	**Features**
Position feedback output	A-, B-, Z-phase line driver output (EIA RS-422A) A-phase and B-phase (dividing rate setting): 16 to 2,048 pulses/revolution Z-phase: 1 pulse/revolution
Speed monitor output	0.5 V/1,000 r/min
Current monitor output	0.5 V/rated torque
Sequence output	Alarm output, motor rotation detection, brake interlock, positioning completion; open-collector outputs: 30 VDC, 50 mA (except for alarm code output, which is 30 VDC, 20 mA)

3.1.2 Terminal Block Specifications

Table 3.4: **Output signals specifications**

Terminal Block Specifications		
Signal	**Function**	**Condition**
L1, L2	Power supply input	(200-VAC Units): Single-phase 200/230 VAC (170 to 253 VAC) 50/60 Hz
+, -	Main circuit DC output	When using multiple axes and there is excessive regenerative energy, the + terminals can be connected together and the – terminals can be connected together to increase the regeneration absorption capacity.
U	Servomotor U phase output	Terminal for outputs to the Servomotor.
V	Servomotor V phase output	Terminal for outputs to the Servomotor
W	Servomotor W phase output	Terminal for outputs to the Servomotor
⏚	Protective earth terminal	This is the connection terminal. Use a 100Ω or less (class-3) or better ground.

Table 3.5: **Terminal Block Current and Wire Sizes**

Terminal Block Current and Wire Sizes	
Power supply input current (L1, L2)	2.5 A
Motor output current (U, V, W)	0.87 A
Power supply input terminal wire size	0.75 mm2 or AWG 18 min.
Motor output terminal wire size	0.5 mm^2 or AWG 20. SGMP B4B MOTOR Power Cable.
Ground terminal wire size	2.0-mm2 external ground wires. Use the same wire as used for the motor output.

3.1.3 Terminal and Connectors functions

Below tables (Table 3.6 and Table 3.7) shown Terminal and Connector functions used for positioning control with 90°differential phase (A/B phases) signals (mode 1). In Figure 3.3 CN1 Signal Connections and External Signal Processing are shown.

1. **CN1 Control Input**

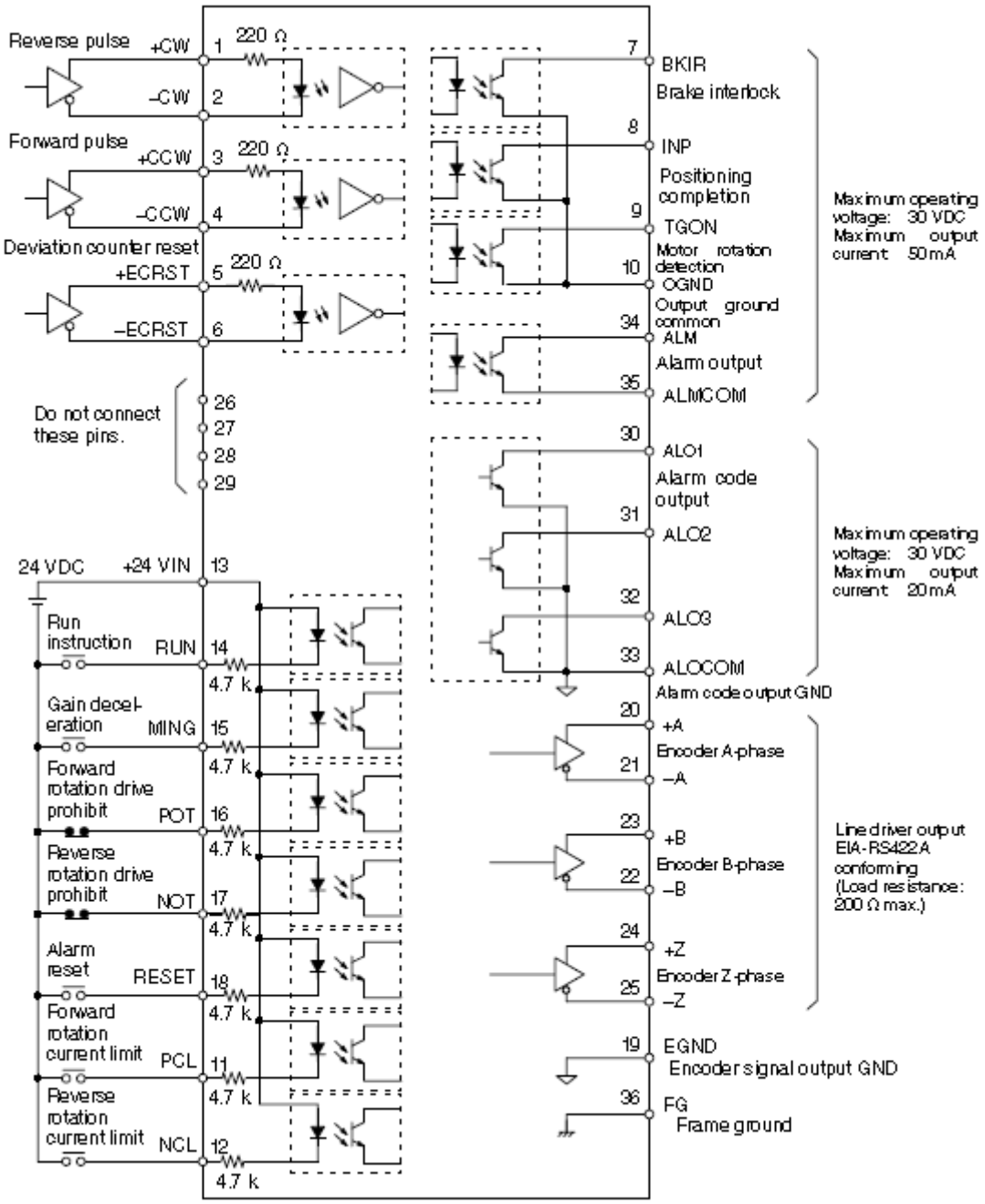


Figure 3.3: CN1 Control I/O Signal Connections and External Signal Processing

Table 3.6: **CN1: Control Input (Pulse Train Input)**

CN1: Control Input (Pulse Train Input)			
Pin №	**Signal №**	**Name**	**Function, Interface**
1	+A	90°differential phase pulse (A phase)	Line driver input: 6 mA at 3V Open collector input: 15 mA at –5V Switched between 90°differential phase pulse (A and B phases) using bits 3, 4, and 5 of the Cn-02 setup parameter. Maximum frequency: 200 kpps
2	-A	90°differential phase pulse (A phase)	
3	+B	90°differential phase pulse (B phase)	
4	-B	90°differential phase pulse (B phase)	
13	+24 VIN	+24V power supply input for control DC	Power supply for pin nos. 11, 12, 14, 15, 16, 17, 18; +24-V input
14	RUN	Run command input	ON: Servo ON, when setup parameter Cn-01 bit no. 0 = 0.When setup parameter Cn-01 bit no. 0 = 1, this signal is not used. (Automatically set to Servo ON.)
16	POT	Forward drive prohibit input	Forward rotation overtravel input (OFF when prohibited). When setup parameter Cn-01 bit no. 2 = 1, this signal is not used.
17	NOT	Reverse drive prohibit input	Reverse rotation overtravel input (OFF when prohibited). When setup parameter Cn-01 bit no. 3 = 1, this signal is not used.

2. **CN1 Control Output**

Table 3.7: **CN1: Control Output (Pulse Train Input)**

CN1: Control Output (Pulse Train Input)			
Pin №	**Signal №**	**Name**	**Function, Interface**
7	BKIR	Brake interlock output	Outputs external brake interlock signal.
8	INP	Positioning competed output	Turned ON when the pulse count remaining in the deviation counter is equal to or less than the positioning completed range set in user parameter Cn-1b.
9	TGON	Servomotor rotation detection output	When setup parameter Cn-01 bit no. 4 = 0, this turns ON if the Servomotor rotational speed exceeds the value set for the Servomotor rotation detection speed (Cn-0b).
10	OGND	Output ground common	Output ground common for BKIR, VCMP, INP, TGON/CLIMT
19	EGND	Encoder signal output GND	This is the ground for encoder signal outputs.
20	+A	Encoder A-phase + output	Outputs encoder pulses divided according to user parameter Cn-0A Line driver output (conforming to RS-422A).
21	-A	Encoder A-phase – output	
22	-B	Encoder B-phase – output	Outputs encoder pulses divided according to user parameter Cn-0A Line driver output (conforming to RS-422A).
23	+B	Encoder B-phase + output	
24	+Z	Encoder Z-phase + output	Encoder Z-phase output (1 pulse/revolution). Line driver output conforming to RS-422A).
25	-Z	Encoder Z-phase – output	
34	$\overline{ALM}$	Alarm output	When the alarm is generated for the Servo Driver, the output is OFF, Open collector output (30 V/50mA max.)
35	ALMCOM	Alarm output GND	
36	FG	Frame ground	Ground terminal for shield wire of cable and FG line.

3. CN2 Encoder Input

Table 3.8: **CN2: Encoder Input**

CN2: Encoder Input			
Pin №	**Signal Name**	**Function**	**Interface**
1, 2, 3	E0V	Encoder power supply GND	Power supply outlet for encoder: 5 V, 120 mA
4, 5, 6	E5V	Encoder power supply +5 V	Power supply outlet for encoder: 5 V, 120 mA
7	DIR	Rotation direction switch input	Connects to E0V when reverse rotation is executed by + input
8, 9	NC	Not used	Do not connect.
10, 11	NC	Not used	Do not connect.
12, 13	NC	Not used	Do not connect.
14	S+	Encoder + S-phase input	Line driver input (conforming to EIA-RS422A)(Input inpedance:220 Ω)
15	S-	Encoder – S-phase input	
16	A+	Encoder + A-phase input	
17	A-	Encoder – A-phase input	Line driver input (conforming to EIA-RS422A)(Input inpedance:220 Ω)
18	B+	Encoder + B-phase input	
19	B-	Encoder – B-phase input	
20	FG	Shielded ground	Cable shielded GROUND

4. CN3 Parameter Unit Input

Table 3.9: **CN3: Parameter Unit Input**

CN3: Parameter Unit Input			
Pin №	**Signal Name**	**Function**	**Interface**
1	TXD+	Transmission data +	This is the send data line-driver output to the Parameter Unit (or a personal computer).
2	TXD-	Transmission data -	
3	RXD+	Reception data +	
4	RXD-	Reception data -	
5	PRMU	Unit switching	This is the switching terminal for a Parameter Unit or personal computer. If the pin is open, it is for a personal computer. If connected to +5V, it is for a Parameter Unit.
6	RT1	Termination resistance enabled/disabled	This is the termination resistance terminal for the line receiver. For 1-to-1 communications or for the final Servo Driver, short-circuit RT1-RT2.
7	RT2	Termination resistance enabled/disabled	
8	+5V	+5 V output	This is the +5 V output to the Parameter Unit
9	GND	Ground	

3.1.4 Wiring Servopack

1. Wiring Terminal Block

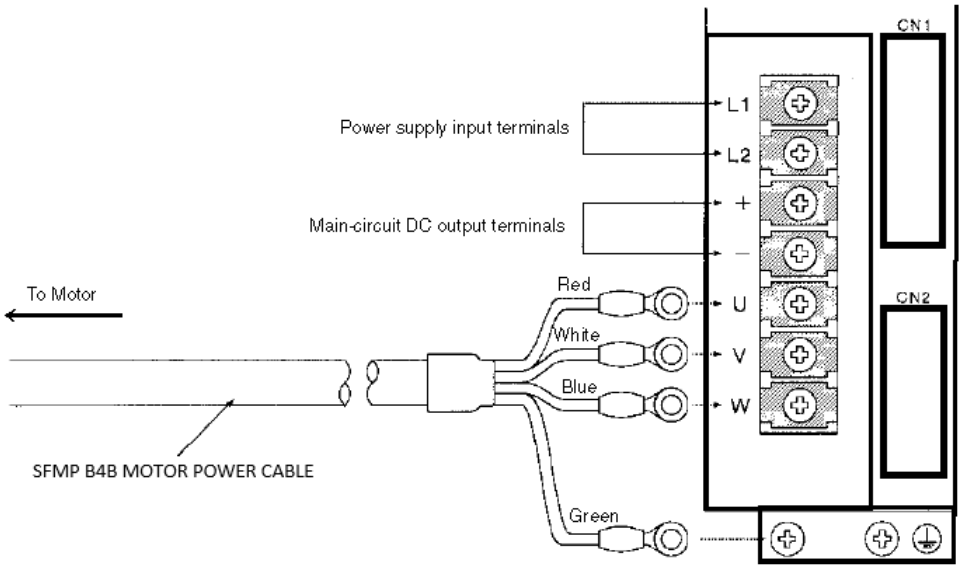


Figure 3.4: Wiring Terminal Block

Table 3.10: **Wiring Terminal Block**

Wiring Terminal Block		
ServoPack Connector	**Cable Type**	**Destination**
Terminal Block	SGMP B4B Motor Power Cable	Servomotor Power Connector

2. Wiring Control I/O Terminal (CN1)

Table 3.11: **Wiring Control I/O Terminal (CN1)**

Wiring Control I/O Terminal (CN1)		
ServoPack Connector	**Cable Type**	**Destination**
CN1	SGMP DE9404859 I/O Cable	PCB (Micro Sub D Connector)

3. Wiring Encoder Input Terminal (CN2)

Table 3.12: **Encoder Input Terminal (CN2)**

Encoder Input Terminal (CN2)		
ServoPack Connector	**Cable Type**	**Destination**
CN2	SGMP DE9407236 Encoder cable	Servomotor Encoder Connector

4. Wiring Parameter Unit Input Terminal (CN3)

Table 3.13: **Parameter Unit Input Terminal (CN3)**

Parameter Unit Input Terminal (CN3)		
ServoPack Connector	**Cable Type**	**Destination**
CN3	D-SUB 9-Pin CONNECTOR Cable	PC

3.2 Servomotor Yaskawa SGMP-01V314T

SGMP Servomotors are synchronous type servomotors. For a synchronous type servomotor, motor speed is controlled by changing the frequency of alternating current.

Figure 3.5: YASKAWA SGMP-01V314T

The Yaskawa SGMP Servomotors have the following features: To get more information see the manual [1].

3.2.1 ServoMotor Features

Table 3.14: **Servo motor features**

FEATURES	
Manufacturer	YASKAWA Electric Corporation
Serie code	SGMP-01V314T
Type	AC synchronous servomotor
Power Supply	200V CE European Specification
Rated Current	0.89 A
Rated Power	100W (0.13HP)
Rated Torque	0.318 N·m
Rated Rotation Speed	3000 rpm
Shaft Specifications	straight with key
Encoder Specifications	2048 P/R incremental encoder

3.2.2 Wiring ServoMotor

Table 3.15: **ServoMotor Connections**

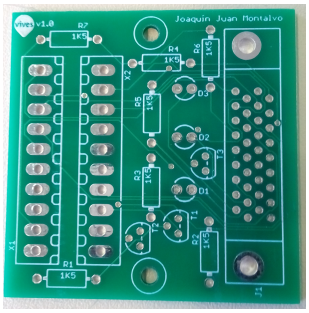
ServoMotor		
ServoMotor Connector	**Cable Type**	**Destination**
Encoder Connector	SGMP DE9407236 Encoder Cable	ServoPack (CN 2)
Power Connector	SGMP B4B Motor Power Cable	ServoPack Terminal Block

Chapter 4

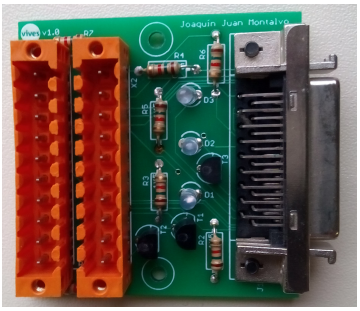
Extra Hardware

4.1 Printed Circuit Board

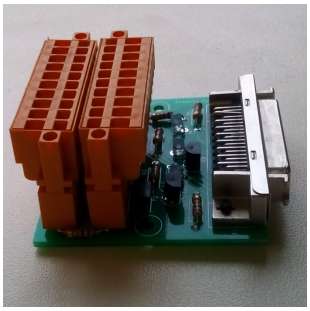
The PCB shown in Figure 4.1a has been designed to facilitate the connections of ServoPack R88D-UP04V with SIGMATEK modules. An SL 3.5 Series connector with the BLZF 3.5/10 Pluggable Terminal Block allows connection of 20 wires with the host controller. A Micro D Sub Connector allows the PCB connection with the servopack connector CN1 via SGMP DE9407236 Encoder cable.



(a) PCB



(b) PCB with components soldered



(c) PCB with pluggable terminal

4.1.1 PCB Components

1. CONNECTORS

Table 4.1: **MICRO D SUB CONNECTOR**

MICRO D SUB CONNECTOR					
Image	**Items**	**Description**	№ Contacts	Contact Type	Current Rate
	1	3M 10236-5212PL Micro D Sub Connector, Right Angle, 102 Series, Receptacle, PCB Mount	36	300V / Through Hole	1A

Table 4.2: **WEIDMUELLER CLAMP CONNECTOR**

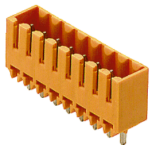
WEIDMUELLER CLAMP CONNECTOR					
Image	**Items**	**Description**	№ Contacts	Contact Type	Pitch Spacing
	2	WEIDMULLER SL 3.5/10/180G Wire-To-Board Connector, Top Entry, SL 3.5 Series, Plug	10	Through Hole	3.5mm

Table 4.3: **WEIDMUELLER PLUGGABLE TERMINAL**

WEIDMUELLER CLAMP CONNECTOR					
Image	**Items**	**Description**	№ Positions	Rated Voltage / Current	Conductor
	2	WEIDMULLER BLZF 3.5/10 Pluggable Terminal Block, 22 AWG, 14 AWG	10	300V / 10A	$1.5mm^2$

2. TRANSISTORS

Table 4.4: **TRANSISTOR NPN**

TRANSISTOR NPN							
Image	**Items**	**Description**	Case Style	V_{CEO}	V_{CBO}	Freq.	I_C
	3	ON SEMICONDUCTOR 2N5550G. Bipolar (BJT) Single Transistor, General Purpose, 625 mW,60 hFE	TO-92	140 Vdc	160 Vdc	300 MHz	600mA

3. LEDs

Table 4.5: **LEDs**

LEDs						
Image	**Quantity**	**Description**	Packag.	Size	Ford. Voltage	Current If
	3	VAOL-3LSBY1 LED, 3MM, BLUE, 1.2CD, 470NM	TO-92	140 Vdc	160 Vdc	300 MHz

4. RESISTORS

Table 4.6: **RESISTORS**

TRANSISTOR NPN							
Image	**Quantity**	**Description**	Type	Resist.	Toler.	Power	Volt. Rat.
	7	TE CONNECTIVITY CFR50J8K2 Through Hole Resistor, CFR Series, Axial Leaded	Carbon Film	8K2Ω	±5%	500 mW	350 V

4.1.2 Pin Arrangement

Table 4.7: **Pin Arrangement**

Pin Arrangement				
Micro D Sub Pin	**SL-3.5 Pin**	**Function**	**Destination**	**Pin**
1	X1-10	Puls +A	Power Stack Control	X2-2/X5-2
2	X1-09	Puls -A	Power Stack Control	X2-1
3	X1-08	Puls +B	Power Stack Control	X2-4
4	X1-07	Puls -B	Power Stack Control	X2-3
7	X1-06	BKIR	CDI 161	X4-1
8	X1-05	INP	CDI 161	X1-1
9	X1-04	TGON	CDI 161	X3-3
13	X1-03	+24V	Power Supply	+24V
34	X1-02	ALM	CDI 161	X1-3
14	X1-01	RUN	CTO 163	X1-1
16	X2-10	POT	CTO 163	X2-1
17	X2-09	NOT	CTO 163	X2-3
10	X2-08	OGND	Power Supply	GND
35	X2-08	ALMCOM	Power Supply	GND

25	X2-07	Puls -Z	Incremental Encoder Input	X1-5
24	X2-06	Puls +Z	Incremental Encoder Input	X1-6
23	X2-05	Puls +B	Incremental Encoder Input	X1-4
22	X2-04	Puls -B	Incremental Encoder Input	X1-3
21	X2-03	Puls -A	Incremental Encoder Input	X1-1
20	X2-02	Puls +A	Incremental Encoder Input	X1-2
19	X2-01	EGND	Incremental Encoder Input	X1-7
5, 6, 11, 12, 15, 18, 26, 27, 28, 29, 30, 31 ,32, 33, 34, 36	—	NOT USED	NOT USED	—

4.1.3 Wiring SL 3.5 Connectors

Table 4.8: **Wiring SL 3.5 Connector**

SL 3.5 Connections				
PCB Connector	**UTP Cable №**	**Wire Color**	**Destination**	**Pin**
X1-1	4	Blue	CTO 163	X1-1
X1-2	3	White-Blue	CDI 163	X1-3
X1-3	5	Orange	Power Supply	+24V
X1-4	3	White-Brown	CDI 163	X3-3
X1-5	3	White-Green	CDI 163	X3-1
X1-6	3	White-Orange	CDI 163	X4-1
X1-7	2	Green	CST 022	X2-3
X1-8	2	White-Green	CST 022	X2-4
X1-9	2	Blue	CST 022	X2-1
X1-10	2	White-Blue	CST 022	X2-2

X2-1	1	White-Orange	CST 022	X1-7
X2-2	1	Blue	CST 022	X1-2
X2-3	1	White-Blue	CST 022	X1-1
X2-4	1	White-Green	CST 022	X1-3
X2-5	1	Green	CST 022	X1-4
X2-6	1	Brown	CST 022	X1-6
X2-7	1	White-Brown	CST 022	X1-5
X2-8	5	Blue	Power Supply	GND
X2-9	4	Brown	CTO 163	X2-3
X2-10	4	Green	CTO 163	X2-1

4.2 Microswitch

Three switches per motor can be used. Two as End Switches and one Reference Switch (the reference program couldn't be add in this project because of lack of time although tests were performed successfully).

4.2.1 Microswitch Features

Table 4.9: **Microswitch Features**

Microswitch						
Image	**Items**	**Description**	Type	Oper. Force	I Max.	Volt. AC Nom.
	3	OMRON ELECTRONIC COMPO-NENTS D3V-16-3C25 Microswitch, D3V Series, SPST-NO, Quick Connect, Solder	Miniature	1.96N	16A	250V AC

4.2.2 Pin Arrangement

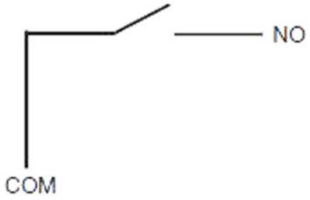


Figure 4.2: Microswitch D3V-16-3C25 Schematic

Table 4.10: **Microswitch 1 Pin Arrangement**

Pin Arrangement Microswitch 1			
Microinterruptor Pin	**Function**	**Destination**	**Pin**
NO	Reference	CDI 163	X1-1
COM	+24V	Power Supply	+24V DC

Table 4.11: **Microswitch 2 Pin Arrangement**

Pin Arrangement Microswitch 2			
Microinterruptor Pin	**Function**	**Destination**	**Pin**
NO	End Switch Forward	CDI 163	X2-1
COM	+24V	Power Supply	+24V DC

Table 4.12: **Microswitch 3 Pin Arrangement**

Pin Arrangement Microswitch 3			
Microinterruptor Pin	**Function**	**Destination**	**Pin**
NO	End Switch Reverse	CDI 163	X2-3
COM	+24V	Power Supply	+24V DC

4.2.3 Wiring Microswitch

Table 4.13: **Wiring Microinterruptor 1**

Microswitch 1 Connections				
Microswitch Connector	**Wire Section**	**Wire Color**	**Destination**	**Pin**
ON	$1mm^2$	Yellow	CDI 163	X1-1

Table 4.14: **Wiring Microinterruptor 2**

Microswitch 2 Connections				
Microswitch Connector	**Wire Section**	**Wire Color**	**Destination**	**Pin**
ON	$1mm^2$	Blue	CDI 163	X2-1

Table 4.15: **Wiring Microinterruptor 3**

Microswitch 3 Connections				
Microswitch Connector	**Wire Section**	**Wire Color**	**Destination**	**Pin**
ON	$1mm^2$	Grey	CDI 163	X2-3

Table 4.16: **Wiring Microinterruptor**

Microswitch Connections				
Microswitch 1,2 & 3 Connector	**Wire Section**	**Wire Color**	**Destination**	**Pin**
COM	$1mm^2$	Brown	Power Supply	+24V DC

Chapter 5

Setting ServoPack Parameters

The servo pack Omron R88D-UP04V should be correctly confiuged for proper system operation.

The configuration used allows to operate with positioning control by pulse-train inputs with 90°differential phase (A/B phase) pulses and signal (4×) as shown in subsection 5.1.3

To proceed with the configuration you need to connect the Servopack to a personal computer via RS-232C or RS-422A cable and use the program WMon.

To send the parameters to the SERVOPACK with WMON Software see section **??**

To get more information about Wmon see the WMon Win U serie manual [7].

The program WMON, the configuration file with the parameters defined in the sections subsection 5.1.1,subsection 5.1.2 and subsection 5.1.4 and the Wmon manual can be found in the CD attached to this document.

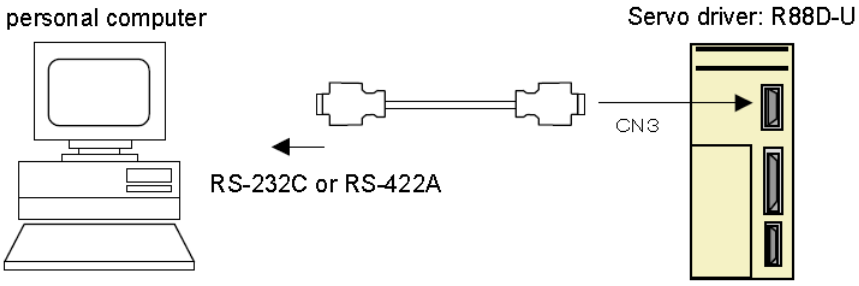


Figure 5.1: PC Connection

5.1 Setup parameters

Setup parameters are parameters that are essential for starting up the system. They include I/O signal function changes, selection of processing for momentary stops and errors, command pulse modes, and so on.

5.1.1 Setup Parameter No. 1

Table 5.1: **Setup Parameter No. 1 (Cn-01)**

Cn-01			
Item	**Bit №**	**Setting**	**Explanation**
Sequence input signal switching	0	0	Servo turned ON or OFF by Run command (externally input).
	1	0 (Factory setting)	Not used
	2	0	Enables forward drive prohibit input (POT).
	3	0	Enables reverse drive prohibit input (NOT).
Sequence output signal switching	4	0	Takes TGON/CLIMT signal as motor rotation detection output.
Processing at time of recovery from momentary stop	5	0	Servo alarm set at time of recovery from momentary stop.
Abnormal stop	6	0	Motor stopped by dynamic brake.
	7	0	Dynamic brake OFF after motor stopped.
	8	0	Method for stopping when over-travel occurs depends on bit no. 6 setting.
	9	0	When over-travel occurs, motor comes to deceleration stop and servo turns OFF.
Deviation counter with servo OFF	A	0	Clear counter for alarms occurring while Servo is OFF
P control switch selection	B	0	Switch control according to bits C and d.
P control switch conditions	D,C	0,0	The torque command value (Cn-0C) is taken as the condition.

—	E	0 (Factory setting)	Not used
Pulse stop switching	F	1	Position Control (when bit 2 of Cn-02 is 0) Enables the pulse stop input. (CN1-15 is the pulse stop input (IPG).) Internal speed control settings (when bit 2 of Cn-02 is 1) Command pulses aren't received when PCL and NCL are OFF. (Position control is performed with the internal speed control settings and the pulse-train input.)

5.1.2 Setup Parameter No. 2

Table 5.2: **Setup Parameter No. 2 (Cn-02)**

Cn-02			
Item	**Bit №**	**Setting**	**Explanation**
Reverse rotation mode	0	0	Rotates in the CCW direction with a + command.
	1	0 (Factory setting)	Not used
Input command mode	2	0	Position control with pulse-train input: CN1-11 and 12 are used as forward and reverse current limit inputs (PCL, NCL). CN1-15 will be the gain reduction (MING) if Cn-01 bit F is set to "0" or the pulse stop input (IPG) if Cn-01 bit F is set to "1."
Command pulse mode	5,4,3	1,0,0	90° phase difference (A/B phase) signal (4X)
—	6	0 (Factory setting)	Not used
—	7	1	Not used
—	8	1	Not used

—	9	0 (Factory setting)	Not used
Deviation counter clear	A	0	Clears the deviation counter when the signal is high level
Speed integration constant's units	B	0	1 ms
Torque command filter time constant	C	0	Primary filter
Command pulse logic reversal	D	0	Positive logic
Parameter Unit monitor output lever change	E	0	Position deviation monitor set for 1 command.
—	F	0 (Factory setting)	Not used

5.1.3 Important Setup Parameters (Cn-01 & Cn-02)

This section explains the particularly important setup parameters. If these parameters aren't set properly, the motor might not operate or might operate unpredictably.

1. Control Mode Settings

   The control mode is determined by the following setup parameters:

   (a) Input command mode: Cn-02 bit 2 (position control by pulse-train input/internal speed control settings.

   (b) Pulse stop switch: Cn-01 bit F (The function of this bit depends on the setting of Cn-02 bit 2.)

Table 5.3: **Control Mode Settings**

Control Mode Settings		
Cn-02 bit 2	**Cn-01 bit**	**Control** Mode
0	1	Position control by pulse-train inputs (pulse stop input (IPG) enabled)

2. Command Pulses in Position Control

   Bits 3, 4, and 5 of Cn-02 specify the kind of command pulse mode used for position control.

Table 5.4: **Command Pulses in Position Control**

Command Pulses in Position Control			
Cn-02 bit 3	**Cn-02 bit 4**	**Cn-02 bit 5**	**Selected command pulse mode**
0	0	1	90°differential phase (A/B phase) signal (4×)

By selecting 4× multiple the input pulses are multiplied by a factor of 4, so the number of motor revolutions (speed and angle) are 4 times the number when the 1× multiple is selected.

5.1.4 User Parameters for Positioning Control

Table Table 5.5 show parameters used in this project for positioning control with 90°differential phase (A/B phases) signals (mode 1).

Table 5.5: **User Parameters**

User Parameter Chart						
Cn.№	Parameter Name	Value	Units	Min	Max	Explanation
Cn-04	Speed loop gain	30	Hz	1	2000	Adjusts speed loop response.
Cn-05	Speed loop integration constant	300	ms	2	10000	Speed loop integration const. The units can be set with bit b of Cn-02. Bit b=0: 1-ms units Bit b=1: 0.01-ms units
Cn-06	Emergency stop torque	200	%	0	322	Deceleration torque when abnormality occurs (compared to rated torque).
Cn-07	Software start acceleration time	0	ms	0	10000	Acceleration time setting for software start.

Cn-08	Forward torque limit	200	%	0	322	Output torque for rotation in forward direction (compared to rated torque).
Cn-09	Reverse torque limit	200	%	0	322	Output torque for rotation in reverse direction (compared to rated torque).
Cn-0A	Encoder divider rate	2048	Pulses /rev.	16	2048	Setting for number of output pulses from Servo Driver.
Cn-0b	Rotational speed for motor rotation detection	1	r/min	1	4500	Setting for rotational speed for motor rotor detection output.
Cn-0C	P control switching (torque commands)	200	%	0	322	If a torque command exceeds this value, the mode switches from PI to P control.
Cn-0d	P control switching (speed commands)	0	r/min	0	4500	If a speed command exceeds this value, the mode switches from PI to P control.
Cn-0E	P control switching (acceleration commands)	0	10 (r/min) /s	0	3000	If an acceleration command exceeds this value, the mode switches from PI to P control.
Cn-0F	P control switching (deviation pulse)	20	Command units	0	10000	If the deviation pulse exceeds this value, the mode switches from PI to P control.
Cn-10	Jog speed	300	r/min	0	4500	Setting for manual rotational speed
Cn-011	Number of encoder pulses	2048	Pulses / rev.	2048	2048	Setting for number of pulses for encoder used.

Cn-12	Brake timing 1	0	10 ms	0	50	Delay time setting from brake command until servo turns off.
Cn-15	Brake com- mand speed	100	r/min	0	4500	Sets rotational speed for outputting brake commands.
Cn-16	Brake timing 2	10	10 ms	10	100	Waiting time from servo-off to brake command output.
Cn-17	Torque command filter time constant	50	100 μ s	0	250	Setting for torque command filter time constant (6.4 to 398 Hz).
Cn-18	Forward rotation external current limit	100	%	0	322	Output torque for when forward rotation current limit is input (compared to rated torque).
Cn-19	Reverse rotation external current limit	100	%	0	322	Output torque for when reverse rotation current limit is input (compared to rated torque).
Cn-1A	Position loop gain	20	1/s	1	500	For position loop response adjustment.
Cn-1b	Positioning completion range	3	Command units	0	250	Sets the range for the positioning completion signal output.
Cn-1C	Bias rotational speed	0	r/min	0	450	Sets the bias for position control.
Cn-1d	Feed-forward amount	50	%	0	100	Position control feed-forward compensation.
Cn-1E	Deviation counter overflow level	1024	x 256 commands	1	32767	Sets the level for detection of deviation counter overflow.

Cn-1F	No. 1 internal speed setting	100	r/min	0	4500	Rotational speed, no. 1 internal setting
Cn-20	No. 2 internal speed setting	200	r/min	0	4500	Rotational speed, no. 2 internal setting
Cn-21	No. 3 internal speed setting	300	r/min	0	4500	Rotational speed, no. 3 internal setting
Cn-23	Software start deceleration time	0	ms	0	10000	Sets the deceleration time for software starts.
Cn-24	Electronic gear ratio G1 (numerator)	1	—	1	65535	Setting range $1/100 \leqslant G1/G2 \leqslant 100$
Cn-25	Electronic gear ratio G2 (denominator) (see note 2)	1	—	1	65535	Setting range $1/100 \leqslant G1/G2 \leqslant 100$
Cn-26	Position command acceleration/ deceleration time constant	0	x 0.1 ms	0	640	Sets the time constant for smoothing.
Cn-27	Feed-forward command filter	100	x 0.1 ms	0	640	Sets the feed-forward command filter.
Cn-28	Compensating gain	50	—	0	100	Adjustment gain during position control
Cn-29	Unit number setting	0	—	0	14	Unit number setting used during multi-axis communications

Chapter 6

SIGMATEK Software

6.1 Lassal package

Lasal package is used for creating projects and visualizations for machine processes.

This package is compose by:

- **LASAL CLASS:** Programming tool to create application projects
- **LASAL MOTION:** Software tool for control of axes
- **LASAL SCREEN:** Visualization tool for graphic terminals
- **LASAL LARS:** Software tool for Simulations.

For this project LASAL CLASS and LASAL MOTION software have been used for create the application project, LASAL SCREEN for create the visualization project and LARS for simulations during programming process.

6.1.1 Lasal Class 2 project tool

LASAL CLASS (**Control Logic Application Software System**) is the engineering tool from SIGMATEK to solve automations tasks.

Lasal Class 2 Integrate Object Oriented Programming (OOP) with graphic representation and client/server communication into automation technology to prepare programs for machine process control. With object oriented programming, the various components of a machine or system are represented in the form of objects.

This tool allows placement and visualization of objects in a network showing the interaction between the individual objects. Behind each object stands a class contain-

ing the program code and the corresponding data elements. Programs are defined under classes and objects work with the program of a class.

The various components of a machine or system are represented in the form of objects Behind each object stands a class containing the program code and the corresponding data elements.

Lasal Class 2 has a library collection components. Components imported from the library can be linked in the project and used to implement applications.

Lasal Class 2 as a multilanguage Programming tool can create applications in different program languages Structured Text (**ST**), Instruction list (**AWL**), Sequential Function Chart (**Interpreter**) and **C**.

6.1.2 Lasal Motion

LASAL Motion is a library of LASAL classes for implementing positioning tasks with various drive manufacturers. With the LASAL MOTION packet, axes with different motion sequences such as absolute, relative or infinite positioning as well as coupling functions and coordinated movements can be controlled.

6.1.3 Lasal Screen

LASAL SCREEN is an HMI tool (Human Machine Interface) for all graphic terminals and industrial PCs manufactured by SIGMATEK. The Software LASAL Screen Editor offers all the Tools needed for the construction of visualization projects.

6.1.4 Lasal Lars

LASAL Runtime Server (LARS) provides a Windows based simulation of control programs and visualizations, with which LASAL applications can be run on a limited basis.

This can be uses for various applications:

- Developing visualization projects on the PC when the actual destination hardware is not available
- Demo applications for presentation purposes
- Visualize values from LASAL applications for remote maintenance.

Chapter 7

Lasal Class Project

In this chapter are shown the classes used in Lasal Class 2 project and their functions and connections.

7.1 User Classes

User classes are classes created specifically for this project. Here two User classes has been created (MainControllerClass & ControllerClass).

7.1.1 MainControllerClass

MainControllerClass is composed by 10 clients, 45 Servers, 7 Types and 22 Variables. This class is used to select the different states of clients and variables and display values. The object MainControllerClassObj belongs to this class. It has scheduled two methods on MainControllerClass an Init and a CyWork method. The CyWork method have a Default Cyclic time of 10 ms.

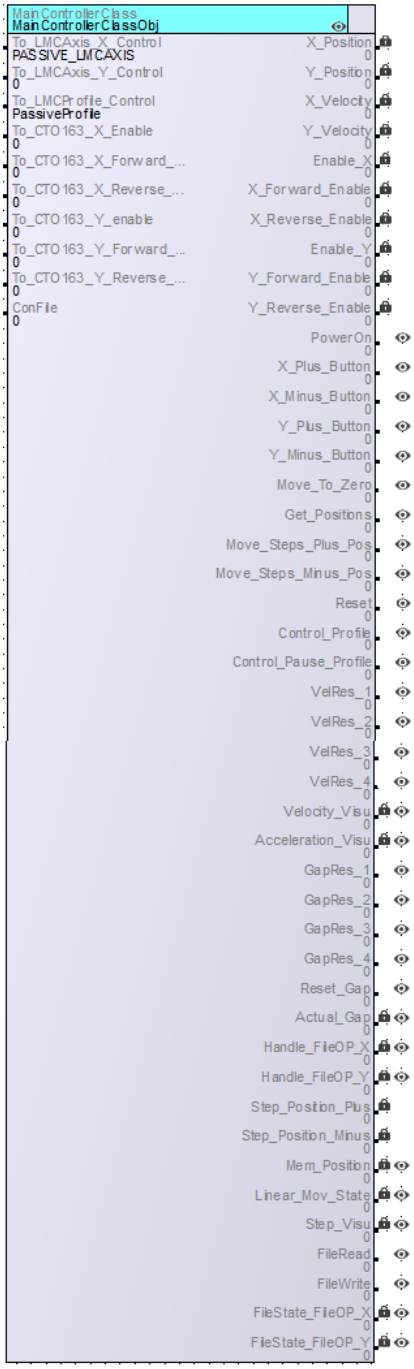


Figure 7.1: MainControllerClass

1. **Types**

   On MainControllerClass are scheduled seven Public Enum types with USINT data Types. Its features are shown in the table below

Table 7.1: **MainControllerClass Types**

MainControllerClass Types			
Type Name	**№Items**	**Function**	**Table**
_STEPS_PROFILE	6	Steps used for linear movement control	a
_STEPS_PAUSE_PROFILE	4	Steps to produce a pause during automatic movement	b
_Switch_1	3	Public Type with three steps. Several uses (see Servers Table).	c
_Switch_2	2	Public Type with two steps. Several uses (see Servers Table).	d
FILESTEPS_Read	4	Steps used to read positions from files and store positions on the array fields	e
FILESTEPS_Write	4	Steps used to create files and save positions in a file	f
_GETNUMBER	2	Steps to store positions in arrays fields	g

a: **_STEPS_PROFILE Items**

_STEPS_PROFILE Items	
Name	**Function**
_AXISSTEP_NO_OPERATION	No operation
_AXISSTEP_CONTROLLER_ON	Initialization of the basic parameters for profile movement
_AXISSTEP_LOCK_PROFILE	Defines which axes are coupled
_AXISSTEP_CONTROLLER_OFF	Shutdown of the axis
_AXISSTEP_MOV_LINEAR	Lineal motion started
_AXISSTEP_STOP_MOVE	Lineal motion stopped

b: **_STEPS_PAUSE_PROFILE Items**

_STEPS_PAUSE_PROFILE Items	
Name	**Function**
_AXISSTEP_MOVE_PAUSE_MODE	Lineal motion paused
_AXISSTEP_MOVE_PAUSE_LINEAR	Lineal motion restarted
_AXISSTEP_STOP_PAUSE_MODE	Lineal motion stopped
_AXISSTEP_NO_OPERATION_PAUSE	No operation

c: **_Switch_1 Items**

_Switch_1 Items	
Name	**Function**
RUN	Activates the defined states
STOP	Activates the defined states
IDLE	No operation

d: **_Switch_2 Items**

_Switch_2 Items	
Name	**Function**
_ON	Activates the defined states
_OFF	No operation

e: **FILESTEPS_Read Items**

FILESTEPS_Read Items	
Name	**Function**
No_Operation_Read	No operation
OpenRead	File is opened
ReadFile	Read values from the file and write this values in the arrays fields
CloseFileRead	File is closed

f: **FILESTEPS_Write Items**

FILESTEPS_Write Items	
Name	**Function**
No_Operation_Write	No operation
CreateFile	Create a file
OpenWrite	Open the file
WriteFile	Write values in the file
CloseFileWrite	Close the file

g: **_GETNUMBER Items**

_Switch_2 Items	
Name	**Function**
PAUSE	No operation
GET	Writes the given position

2. **Variables**

Table 7.2: **MainControllerClass Variables**

Variables		
Name	**Type**	**Function**
Index	DINT	Counter used to write the given positions into array fields
Index_Reset	DINT	Counter used to delete positions on memory
Index_Counter	DINT	Counter increases when the array field is written
Index_Profile	DINT	Counter used to load 9 commands in the LMC-Buffer
Gap	DINT	Used to set relative distances
Acceleration	DINT	Used to set acceleration value
Velocity	DINT	Used to set velocity value
MAESURES_X	ARRAY OF DINT	Used to store X axis positions
MAESURES_Y	ARRAY OF DINT	Used to store Y axis positions
END_POSITIONS	_LMCPROF_POS	Used to Load commands in the LMC-Buffer
Handle_X	DINT	Return whether the file was opened or the error code
Handle_Y	DINT	Return whether the file was opened or the error code
FileState_X	DINT	Return the _Filesys state for X axis
FileState_Y	DINT	Return the _Filesys state for Y axis
nulpunt	_LMCPROF_POS	Set null point offset for axis
Loop_Positions	DINT	Activates the states for the linear movement

Loop_Positions_Pause	DINT	Activates the states for pause linear movement and restart
Step_Control_X_PLUS	DINT	Used as Condition to start relative movement for X axis on forward direction
Step_Control_X_MINUS	DINT	Used as Condition to start relative movement for X axis on backward direction
Step_Control_Y_PLUS	DINT	Used as Condition to start relative movement for Y axis on forward direction
Step_Control_Y_MINUS	DINT	Used as Condition to start relative movement for Y axis on backward direction
Counter	DINT	Increases when the arrays fields are filled

3. **Servers**

   MainControllerClass is composed by 44 Servers. All servers have the following characteristics in common:

   (a) Type = Data Channel
   (b) World = False
   (c) Initialize = False
   (d) Retentive = False

   Other features are shown in the following tables.

Table 7.3: **MainControllerClass Servers**

Performance Data				
Name	**Data Type**	**Write Protected**	**Visual.**	**Function**
X_Position	DINT	True	False	Show actual position for X axis
Y_Position	DINT	True	False	Show actual position for Y axis
X_Velocity	DINT	True	False	Show actual velocity for X axis
Y_Velocity	DINT	True	False	Show actual velocity for Y axis
Velocity_Visu	DINT	True	True	Used to visualize actual velocity in Lasal Screen
Acceleration_Visu	DINT	True	True	Visualize actual Acceleration in Lasal Screen (Not used)
Enable_X	DINT	True	False	Shows the status (value 0 or 1)

X_Forward_Enable	DINT	True	False	Shows the status (value 0 or 1)
X_Reverse_Enable	DINT	True	False	Shows the status (value 0 or 1)
Enable_Y	DINT	True	False	Shows the status (value 0 or 1)
Y_Forward_Enable	DINT	True	False	Shows the status (value 0 or 1)
Y_Reverse_Enable	DINT	True	False	Shows the status (value 0 or 1)
PowerOn	_Switch_1	False	True	Activates the states for Power On and Power Off the system
X_Plus_button	_Switch_1	False	True	Activates the states to move the X axis in forward direction
X_Minus_button	_Switch_1	False	True	Activates the states to move the X axis in reverse direction
Y_Plus_button	_Switch_1	False	True	Activates the states to move the Y axis on forward direction
Y_Minus_button	_Switch_1	False	True	Activates the states to move the Y axis on reverse direction
Move_To_Zero	_Switch_1	False	True	Activates the states to move the X and Y axis to zero position

Get_Positions	_GETNUMBER	False	True	Activates the states for memorize X and Y positions in the arrays
Move_Steps_Plus_Pos	_Switch_2	False	True	Two axis go over the memory locations (stopping after each position) on forward direction
Move_Steps_Minus_Pos	_Switch_2	False	True	Two axis go over the memory locations (stopping after each position) on reverse direction
Reset	_Switch_2	False	True	Delete positions on Arrays
Control_Profile	_STEPS _PROFILE	False	True	Activates the states to control _LMCProfile lineal movement
Control_Pause_Profile	_STEPS _PAUSE _PROFILE	False	True	Activates the states to control _LMCProfile Pause/Restart movement

VelRes_1	_Switch_2	False	True	Activates the states to set Velocity for two axis to 5000 (Internal Units/s)
VelRes_2	_Switch_2	False	True	Activates the states to set Velocity for two axis to 10000 (Internal Units/s)
VelRes_3	_Switch_2	False	True	Activates the states to set Velocity for two axis to 20000 (Internal Units/s)
VelRes_4	_Switch_2	False	True	Activates the states to set Velocity for two axis to 30000 (Internal Units/s)
GapRes_1	_Switch_2	False	True	Activates the states to set relative distance to 1000 (Internal Units)
GapRes_2	_Switch_2	False	True	Activates the states to set relative distance to 10000 (Internal Units)
GapRes_3	_Switch_2	False	True	Activates the states to set relative distance to 20000 (Internal Units)

GapRes_4	_Switch_2	False	True	Activates the states to set relative distance to 30000 (Internal Units)
Reset_Gap	_Switch_2	False	True	Activates the states to set relative distance to 0
Actual_Gap	DINT	True	True	Visualize relative distance in Lasal Screen
Handle_FileOP_X	DINT	True	True	Show whether the file for X axis was opened or the error code when an error occurs
Handle_FileOP_Y	DINT	True	True	Show whether the file for Y axis was opened or the error code when an error occurs
Step_Position_Plus	DINT	True	False	Used as a counter for relative movement of two axis on forward direction
Step_Position_Minus	DINT	True	False	Used as a counter for relative movement of two axis on reverse direction
Mem_Position	DINT	True	True	Used to display the memory position on Lasal Screen

Linear_Mov_State	DINT	True	True	Used on Lasal Screen to block buttons during automatic movement (State Schemmes). Value 0 or 1.
Step_Visu	DINT	True	True	Used to display the memorized position on Screen
FileRead	FILESTEPS _Read	False	True	Activates the states to read positions from files and store positions on the array fields
FileWrite	FILESTEPS _Write	False	True	Activates the states to create files and save positions in a file
FileState_FileOP_X	DINT	True	True	Show error code when an error occurred Operating File System for X axis
FileState_FileOP_Y	DINT	True	True	Show error code when an error occurred Operating File System for Y axis

4. **Clients**

Table 7.4: **MainControllerClass Clients**

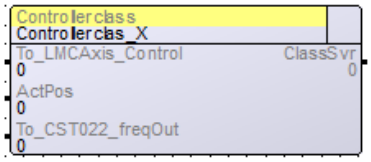
Clients		
Name	**Class**	**Type**
To_LMCAxis_X_Control	_LMCAxis	Object Channel
To_LMCAxis_Y_Control	_LMCAxis	Object Channel
To_LMCProfile_Control	_LMCProfile	Object Channel
To_CTO163_X_Enable	—	Data Channel
To_CTO163_X_Forward_Enable	—	Data Channel
To_CTO163_X_Reverse_Enable	—	Data Channel
To_CTO163_Y_enable	—	Data Channel
To_CTO163_Y_Forward_Enable	—	Data Channel
To_CTO163_Y_Reverse_Enable	—	Data Channel
ConFile	_FileSys	Object Channel

Table 7.5: **MainControllerClass Clients Connections**

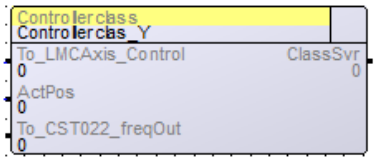
Connections		
Clients Name	**Destination**	**Server Connected**
To_LMCAxis_X_Control	X_LMCAxis	Control
To_LMCAxis_Y_Control	Y_LMCAxis	Control
To_LMCProfile_Control	_LMCProfile1	Control
To_CTO163_X_Enable	CTO163_IM1	Output1
To_CTO163_X_Forward_Enable	CTO163_IM1	Output5
To_CTO163_X_Reverse_Enable	CTO163_IM1	Output7
To_CTO163_Y_enable	CTO163_IM1	Output2
To_CTO163_Y_Forward_Enable	CTO163_IM1	Output6
To_CTO163_Y_Reverse_Enable	CTO163_IM1	Output8
ConFile	CTO163_IM1	Not connected

7.1.2 ControllerClass

ControllerClass is composed by 3 clients and 1 Server. This class used read and write methods to send data to perform the motor's movements to the CST022_IM1 hardware class . Two objects belongs to this class Controllerclas_X and Controllerclas_Y. It has scheduled a RtWork method on ControllerClass.



(a) ControllerClass_X



(b) ControllerClass_Y

Table 7.6: **ControllerClass_X Servers**

ControllerClass_X Servers				
Name	**Data Type**	**Write Protected**	**Visual.**	**Function**
ClassSvr	DINT	True	False	Connection with X_LMCAxis

Table 7.7: **ControllerClass_X Clients**

ControllerClass_X Clients		
Name	**Class**	**Type**
To_LMCAxis_X_Control	_LMCAxis	Object Channel
ActPos	—	Data Channel
To_CST_freqOut	—	Data Channel

Table 7.8: **ControllerClass_X Client Connections**

ControllerClass_X Client Connections		
Clients Name	**Destination**	**Server Connected**
To_LMCAxis_X_Control	X_LMCAxis	Control
ActPos	X_LMCAxis	ActPos
To_CST_freqOut	CST022_IM1	M1_FreqOut

Table 7.9: **ControllerClass_X Server Connections**

ControllerClass_X Server Connections		
Server Name	**Destination**	**Client Connected**
ClassSvr	X_LMCAxis	LMCPostRtWorkTrigger

Table 7.10: **ControllerClass_Y Server**

ControllerClass_Y Servers				
Name	**Data Type**	**Write Protected**	**Visual.**	**Function**
ClassSvr	DINT	True	False	Connection with Y_LMCAxis

Table 7.11: **ControllerClass_Y Clients**

ControllerClass_Y Clients		
Name	**Class**	**Type**
To_LMCAxis_Y_Control	_LMCAxis	Object Channel
ActPos	—	Data Channel
To_CST_freqOut	—	Data Channel

Table 7.12: **ControllerClass_Y Client Connections**

ControllerClass_Y Client Connections		
Clients Name	**Destination**	**Server Connected**
To_LMCAxis_Y_Control	Y_LMCAxis	Control
ActPos	Y_LMCAxis	ActPos
To_CST_freqOut	CST022_IM1	M1_FreqOut

Table 7.13: **ControllerClass_Y Server Connections**

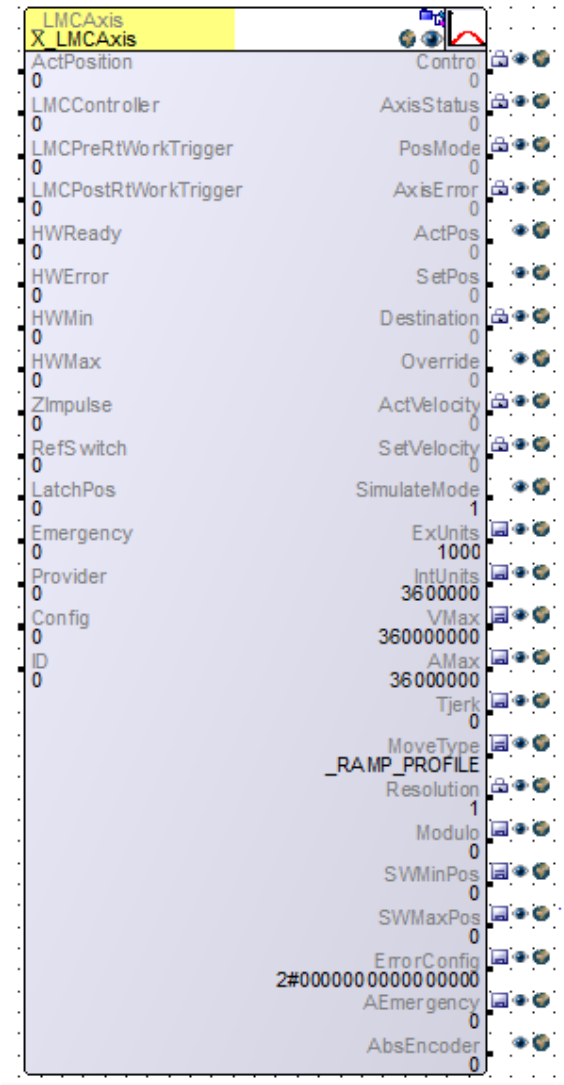
ControllerClass_Y Server Connections		
Server Name	**Destination**	**Client Connected**
ClassSvr	Y_LMCAxis	LMCPostRtWorkTrigger

7.2 Standard Classes

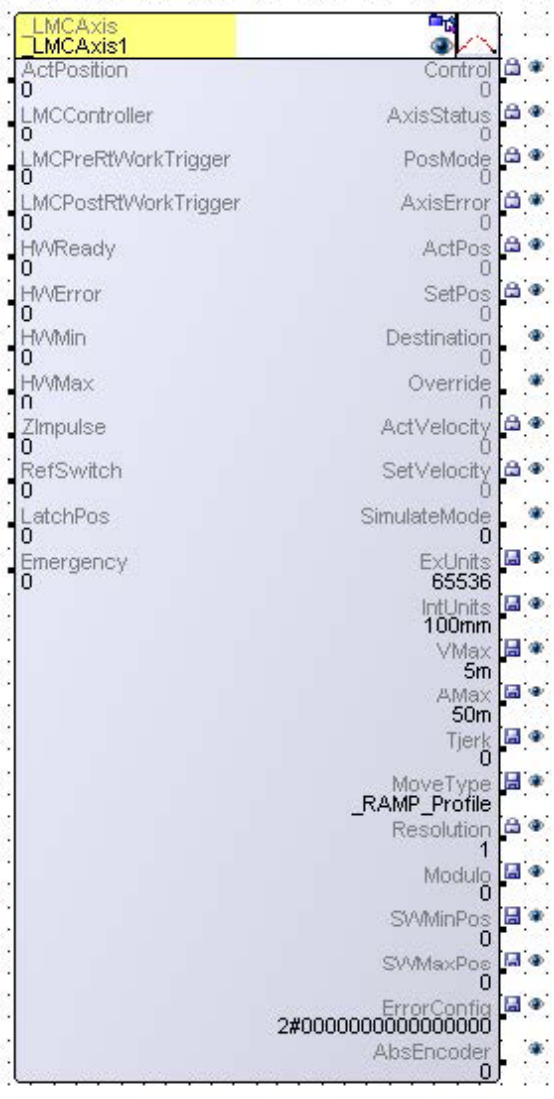
Standard classes are classes created by SIGMATEK for axis control and has been imported into the project from the motion library. For this project two standard classes are used (_LMCAxis (2 objects) & _LMCProfile).

7.2.1 _LMCAxis Class

The _LMCAxis class represents the base class for the profile generator for 1 axis. It calculates the actual position setting and provides with the set and actual values to the controller class. The _LMCAxis class contains all the functions and interfaces needed to drive an axis. Two _LMCAxis objects are placed in MainControllerNet Network, X_LMCAxis & Y_LMCAxis.



(a) X_LMCAxis



(b) Y_LMCAxis

Table 7.14: **X_LMCAxis Client Connections**

X_LMCAxis Client Connections		
Client Name	**Destination**	**Server Connected**
LMCPostRtWorkTrigger	ControllerClass_X	ClassSvr

Table 7.15: **X_LMCAxis Server Connections**

X_LMCAxis Server Connections		
Server Name	**Destination**	**Client Connected**
Control	_LMCProfile	LMCAxis1
Control	MainControllerClass	To_LMCAxis_X_Control
ActPos	Controllerclass_X	ActPos

Table 7.16: **Y_LMCAxis Clients Connections**

Y_LMCAxis Clients Connections		
Client Name	**Destination**	**Server Connected**
LMCPostRtWorkTrigger	ControllerClass_Y	ClassSvr

Table 7.17: **Y_LMCAxis Servers Connections**

Y_LMCAxis Servers Connections		
Server Name	**Destination**	**Client Connected**
Control	_LMCProfile	LMCAxis2
Control	MainControllerClass	To_LMCAxis_Y_Control
ActPos	Controllerclass_Y	ActPos

7.2.2 _LMCProfile Class

The _LMCProfile class allows up to nine axes to be coupled and individually moved at the same time. Using various interfaces, the _LMCProfile class can perform linear and circular movement and complete profiles.

For this project one _LMCProfile object has been placed in MainControllerNet Network for control the two axis performing a linear movement.

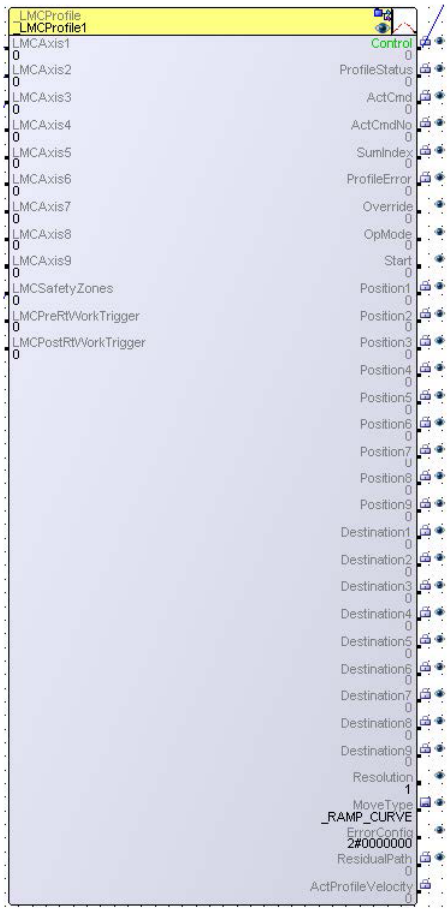


Figure 7.4: _LMCProfile

Table 7.18: **_LMCProfile Clients Connections**

_LMCProfile Connections		
Client Name	**Destination**	**Server Connected**
LMCaxis1	X_LMCAxis	Control
LMCaxis1	Y_LMCAxis	Control

Table 7.19: **_LMCProfile Server Connections**

_LMCProfile Server Connections		
Server Name	**Destination**	**Client Connected**
Control	MainControllerClass	TO_LMCProfile_Control

7.3 Visualization Class

The visualization Class _Lse serves as connection between Lasal Class 2 project and Lasal Screen project.

To build a visualization project, both LASAL Class and a LASAL Screen projects are necessary._Lse Class is a Kernel prepared by SIGMATEK that interprets the lists from LASAL Screen Editor in which all picture, text, language and other information are contained.

_Lse Class is placed in the LasalScreen Network and no connections are made in this class.

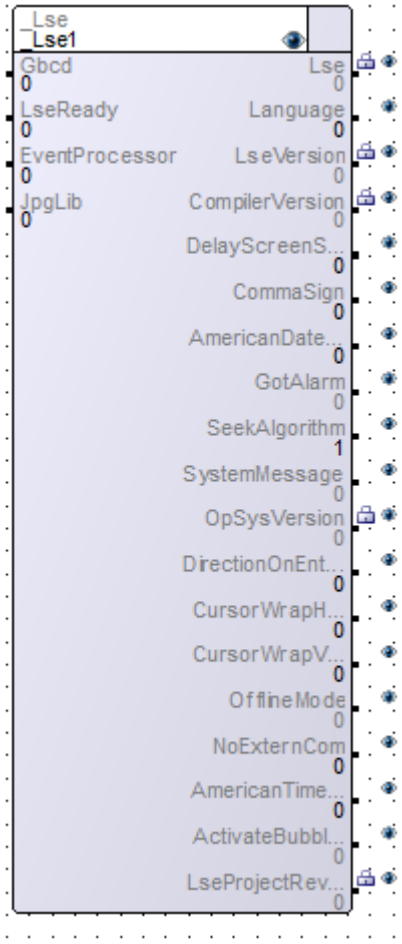


Figure 7.5: _Lse

7.4 Hardware Classes

Hardware classes are classes created by SIGMATEK for harware control. These classes are upload from PLC. For this projects six harware classes are used (HwControl, VaranManager_1,CIV512, CST022_IM, CTO163_IM & CDI161_IM).

7.4.1 HwControl Class

This hardware class controls the flow between VaranManager hardware class and the DiasMasterC hardware class. This class is required by the VaranManager.

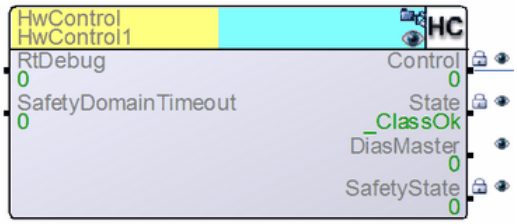


Figure 7.6: HWControl Hardware Class

- Interface Connections:

Table 7.20: **HwControl Connections**

Connections		
HwControl Server Name	**Destination**	**Client Name Connected**
Control	VaranManager	To_HwControl

7.4.2 VaranManager_1 Class

This hardware class establishes the communication with the operating system and the VaranMaster. It controls all VARAN hardware classes.This class is the base class for the following classes:

VaranManager_1 contain 1 VaranOut to which Varan participants can be connected in the object network.

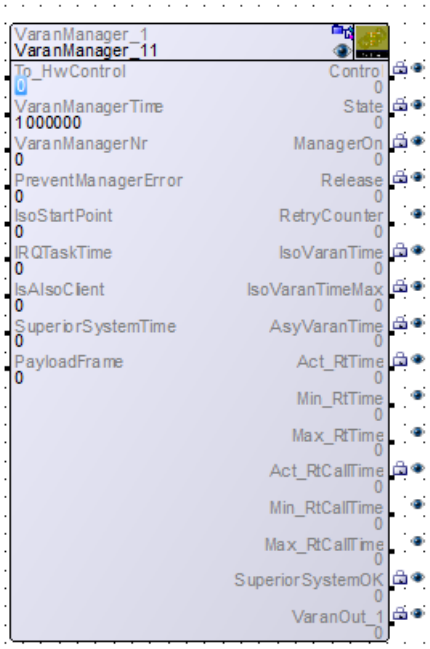


Figure 7.7: HWControl Hardware Class

- Interface Connections:

Table 7.21: **VaranManager_1 Connections**

Connections		
VaranManager_1 Server Name	**Destination**	**Client Name Connected**
VaranOut_1	CIV5121	VaranIn

7.4.3 CIV512 Class

This hardware class is used to control the CIV512 hardware module

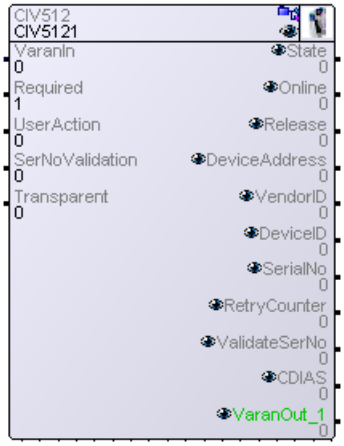


Figure 7.8: CIV512 Hardware Class

- Interface Connections:

Table 7.22: **CIV512 Connections**

Connections		
CIV512 Server Name	**Destination**	**Client Name Connected**
CDIAS	CST022_IM	MasterConnect
CDIAS	CTO163_IM	MasterConnect
CDIAS	CDI161_IM	MasterConnect

7.4.4 CST022_IM Class

This hardware class is used to control the CST022 stepper motor module.

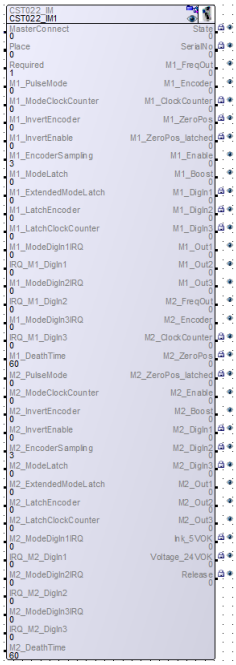


Figure 7.9: CST022_IM Hardware Class

- Interface Connections:

Table 7.23: **CST022_IM Connections**

Connections		
CST022_IM Server Name	**Destination**	**Client Name Connected**
M1_FreqOut	ControllerClass_X	To_CST_freqOut
M2_FreqOut	ControllerClass_Y	To_CST_freqOut
M1_ZeroPos	X_LMCAxis	ZImpulse
M2_ZeroPos	Y_LMCAxis	ZImpulse

7.4.5 CTO163_IM Class

This hardware class is used to control the CTO163 module with 16 digital outputs.

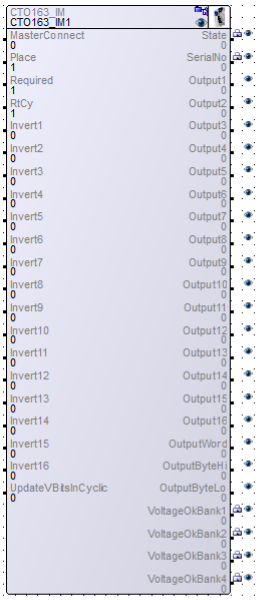


Figure 7.10: CTO162_IM Hardware Class

- Interface Connections:

Table 7.24: **CTO163_IM Connections**

Connections		
CTO163 Server Name	**Destination**	**Client Name Connected**
Output1	MainControllerClass	To_CTO163_X_Enable
Output5	MainControllerClass	To_CTO163_X_Forward_Enable
Output7	MainControllerClass	To_CTO163_X_Reverse_Enable
Output2	MainControllerClass	To_CTO163_Y_Enable
Output6	MainControllerClass	To_CTO163_Y_Forward_Enable
Output8	MainControllerClass	To_CTO163_Y_Reverse_Enable

7.4.6 CDI161_IM Class

This hardware class is used to control the CDI161 hardware module with 16 digital inputs.

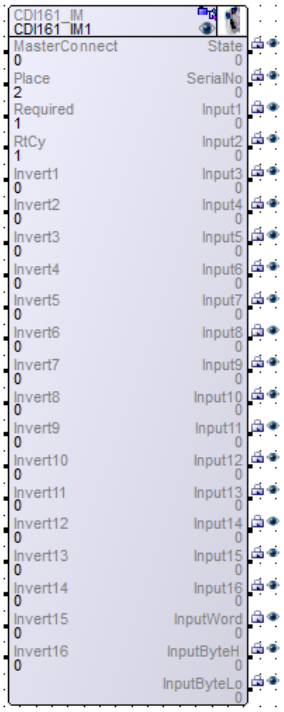


Figure 7.11: CDI161_IM Hardware Class

- Interface Connections:

Table 7.25: **CDI161_IM Connections**

Connections		
CDI161 Server Name	**Destination**	**Client Name Connected**
Input1	X_LMCAxis	RefSwitch
Input5	X_LMCAxis	HWMax
Input7	X_LMCAxis	HWMin
Input2	Y_LMCAxis	RefSwitch
Input6	Y_LMCAxis	HWMax
Input8	Y_LMCAxis	HWMin

Chapter 8

Lasal Screen Project

In this chapter is shown the Lasal Screen project features. For the Lasal Screen project the Sigma640x480.Ipr file has been used as a template. This template provide the settings for the Control Panel ETV 0551 with a resolution of 640x480 pixels and 16 Bit color. Also provide 8 buttons at the bottom of each screen and a Alarm screen. The Alarm screen (screen 5) and alarm buttons are not used in this project. Numbers within a white box at the bottom-left of the buttons in figures 8.1, 8.22, 8.41, 8.46 indicate the button number.

8.1 Screen1 features (Power On/Movement Settings)

The Figure 8.1 shows how looks the first screen of Lasal Screen project.

The upper right button (button 1) is used to start the system (Power on / off). Any button works if the Power On button (button 1) is not active. The four buttons below the text "Actual Velocity" are used to select the motors speed in internal units (buttons from 2 to 5). On the right side of the text "Actual Velocity" the selected speed for the servo motors is shown.

The four buttons below the text "Actual Gap" (buttons from 6 to 9) are used to select the distance to travel by the servo motors before stop the movement. If any of these buttons is pressed, the motor will stop after travelling the selected distance.

On the right side of the text "Actual Gap" the selected distance for the servo motors movement is shown.

Actual gap should be 0 to enable automatic movement (screen 2 buttons) or to use step buttons (Screen 1).

The button below the distance selection buttons (button 10) resets this distance.

The blue buttons on the bottom are used to move through the screens. The characteristics of these buttons are explained in the section 8.6.

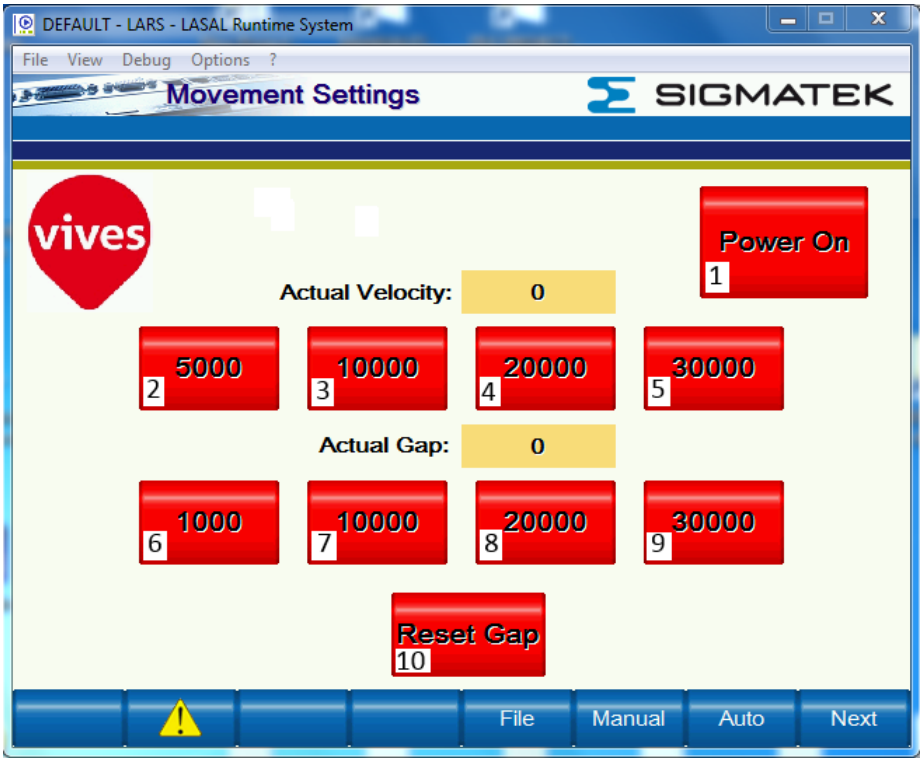


Figure 8.1: Power On/Movement Settings Screen

8.1.1 Power On Button

- Button 1 Component Properties

1. Data Properties

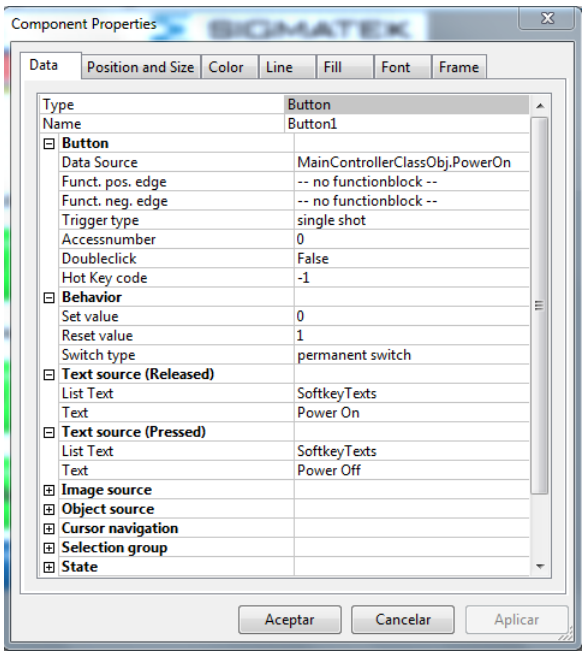


Figure 8.2: Button 1 Data Properties

2. Color Properties

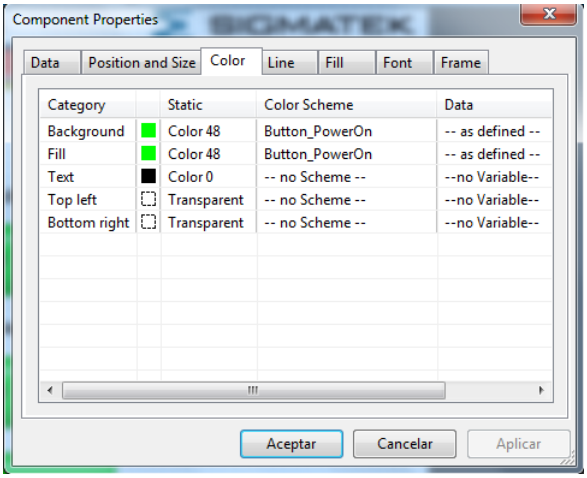


Figure 8.3: Button 1 Color Properties

8.1.2 Speed Selection Buttons

- Button 2 Component Properties

1. Data Properties

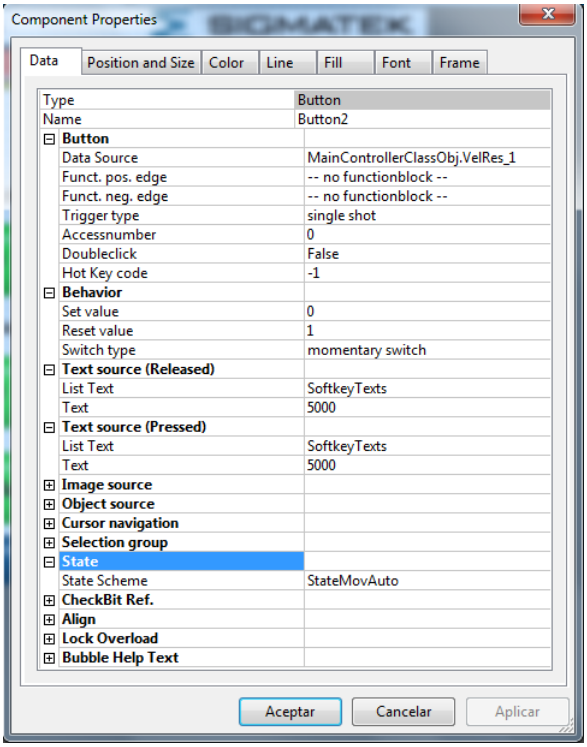


Figure 8.4: Button 2 Data Properties

2. Color Properties

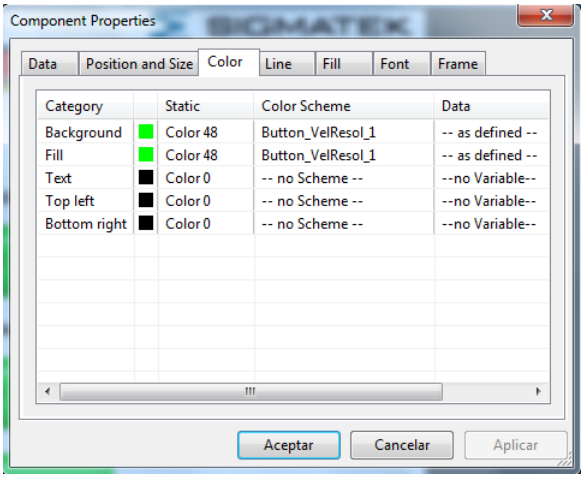


Figure 8.5: Button 2 Color Properties

- Button 3 Component Properties

1. Data Properties

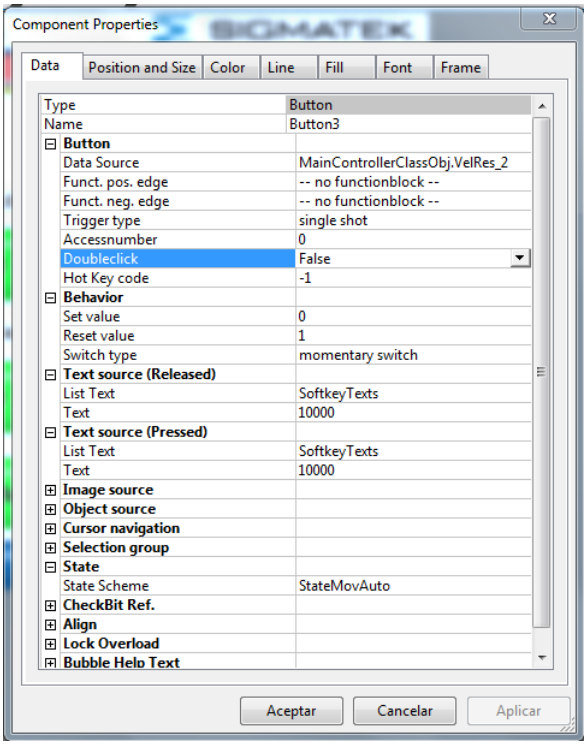


Figure 8.6: Button 3 Data Properties

2. Color Properties

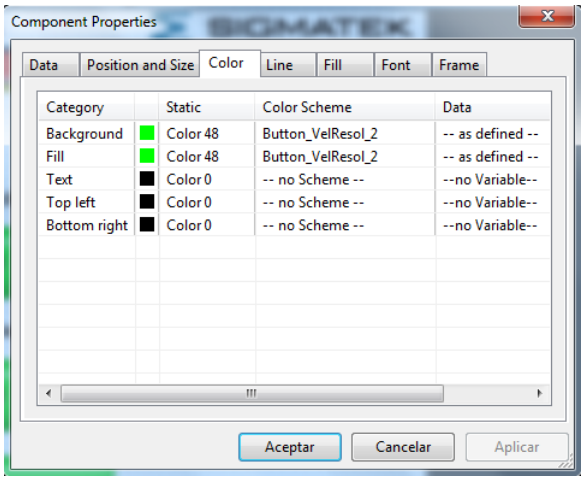


Figure 8.7: Button 3 Color Properties

- Button 4 Component Properties

1. Data Properties

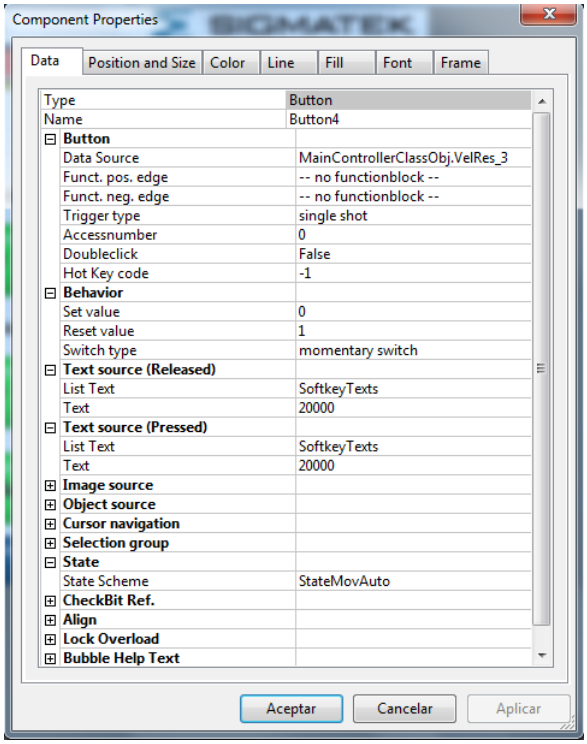


Figure 8.8: Button 4 Data Properties

2. Color Properties

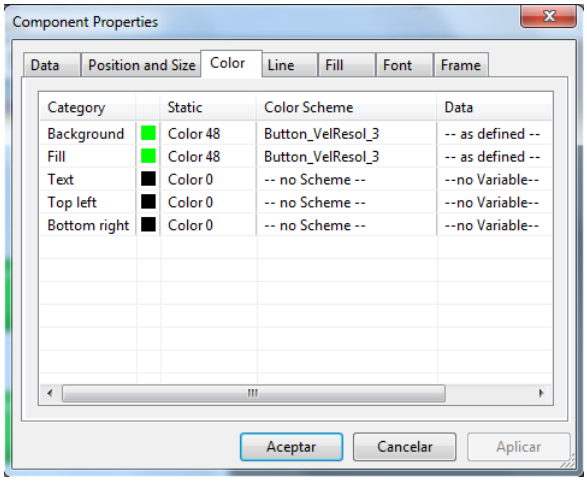


Figure 8.9: Button 4 Color Properties

- Button 5 Component Properties

1. Data Properties

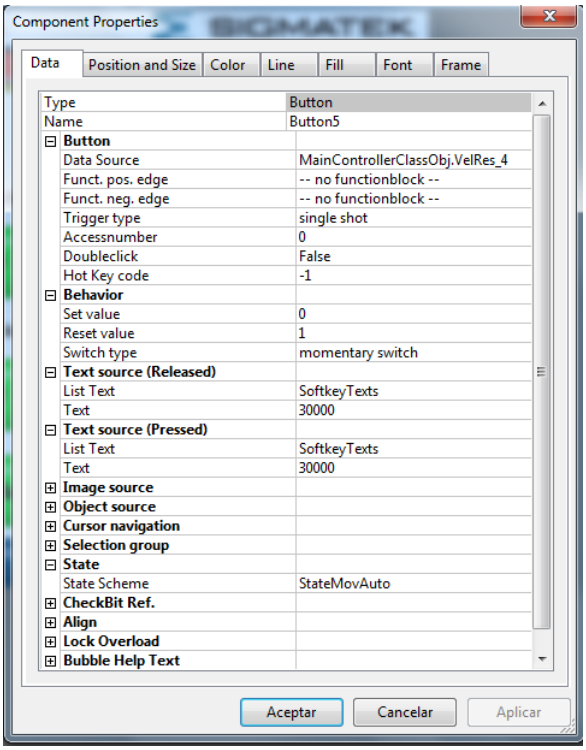


Figure 8.10: Button 5 Data Properties

2. Color Properties

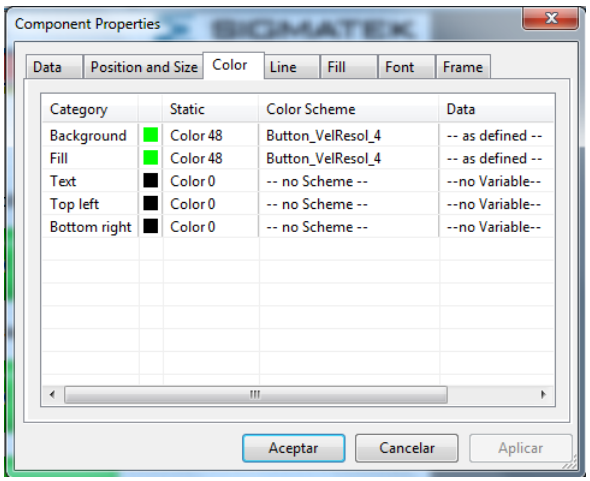


Figure 8.11: Button 5 Color Properties

8.1.3 Distance Selection Buttons

- Button 6 Component Properties

1. Data Properties

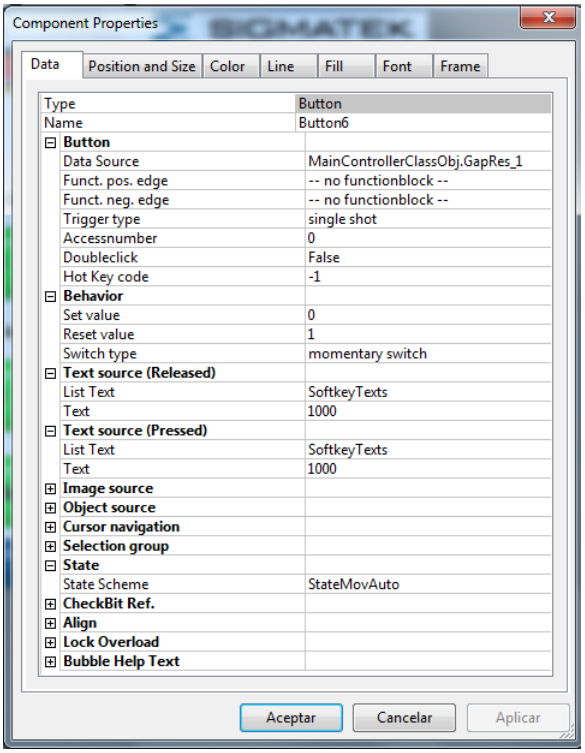


Figure 8.12: Button 6 Data Properties

2. Color Properties

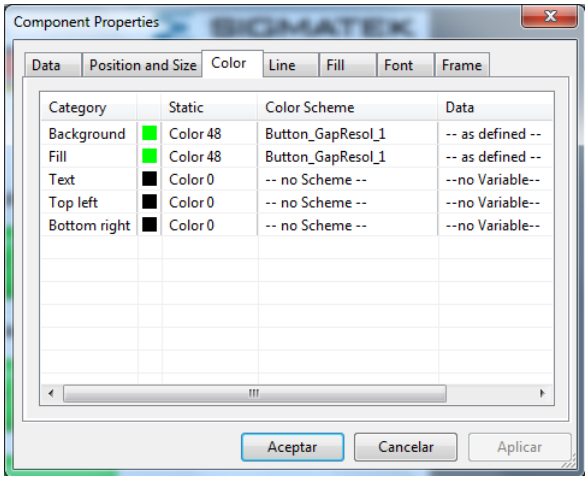


Figure 8.13: Button 6 Color Properties

- Button 7 Component Properties

1. Data Properties

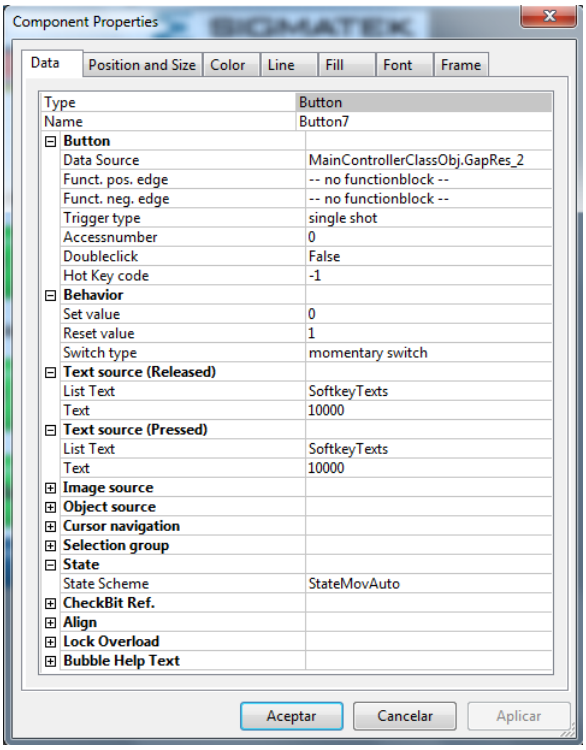


Figure 8.14: Button 7 Data Properties

2. Color Properties

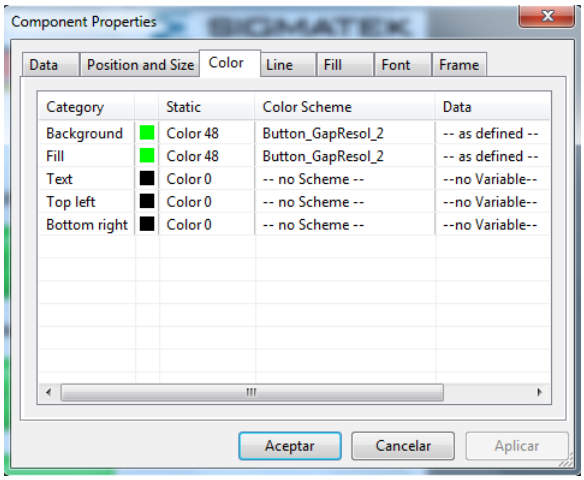


Figure 8.15: Button 7 Color Properties

- Button 8 Component Properties

1. Data Properties

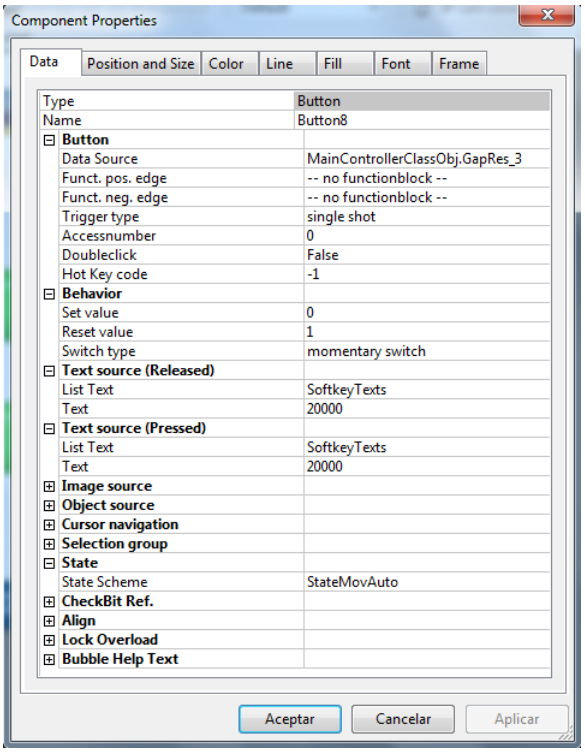


Figure 8.16: Button 8 Data Properties

2. Color Properties

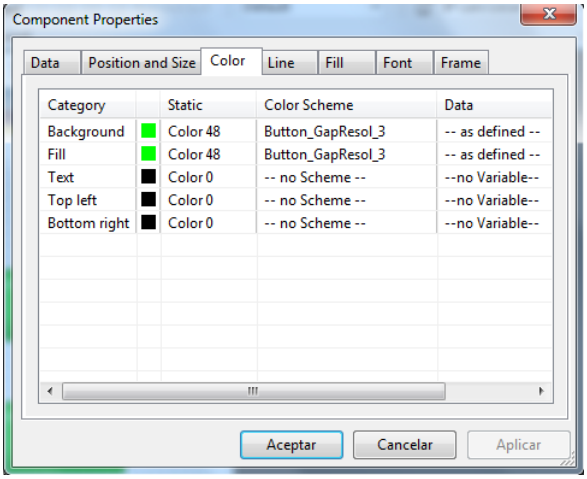


Figure 8.17: Button 8 Color Properties

- Button 9 Component Properties

1. Data Properties

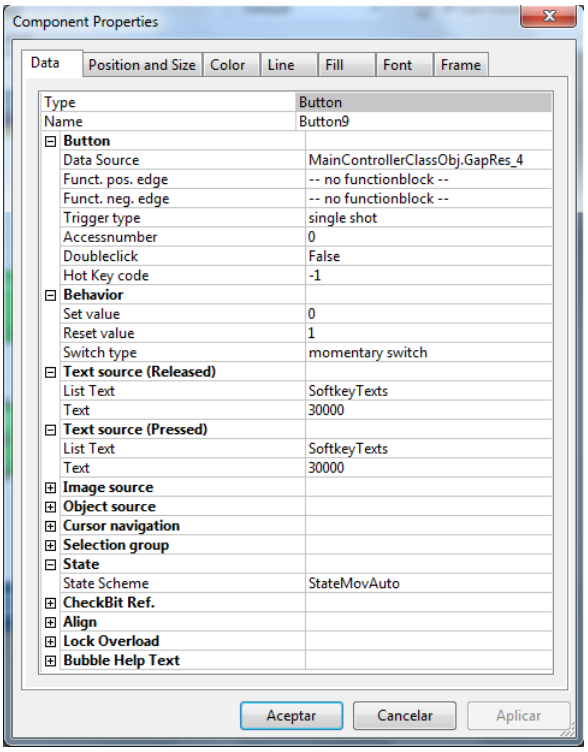


Figure 8.18: Button 9 Data Properties

2. Color Properties

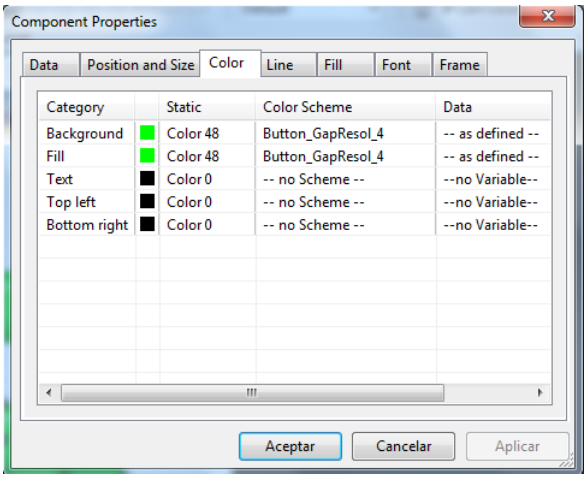


Figure 8.19: Button 9 Color Properties

- Button 10 Component Properties

1. Data Properties

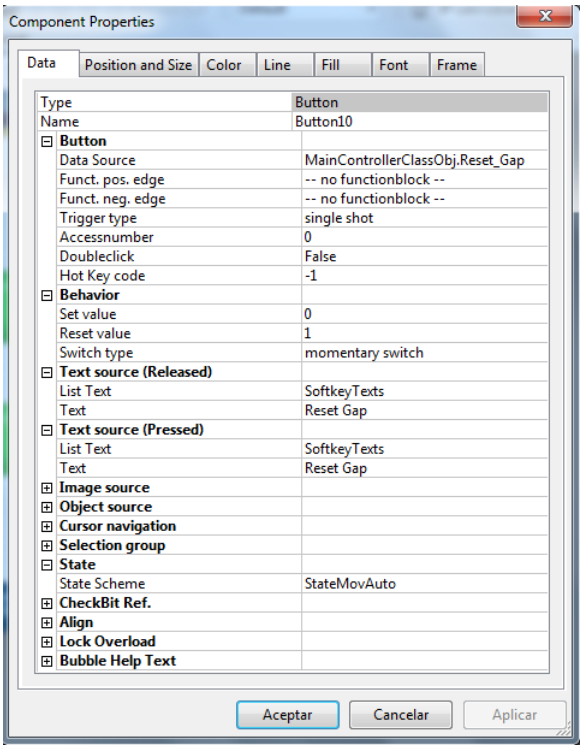


Figure 8.20: Button 9 Data Properties

2. Color Properties

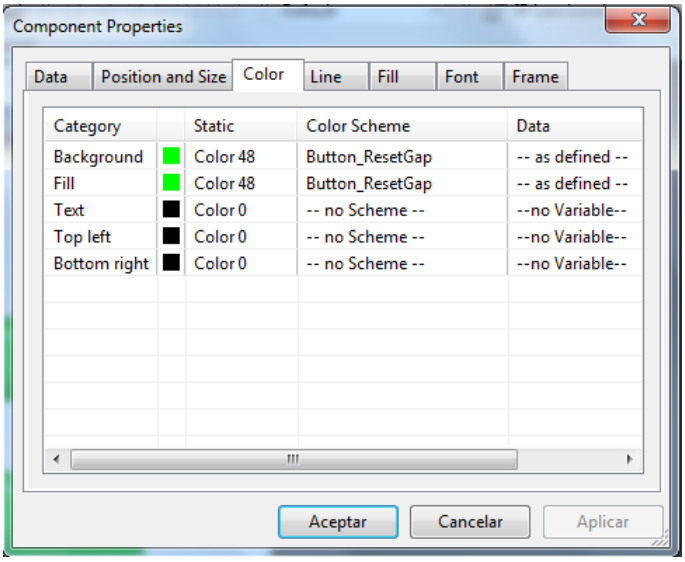


Figure 8.21: Button 9 Color Properties

8.2 Screen2 features (Move Manual & Get Positions)

The Figure 8.22 shows how looks the second screen of Lasal Screen project.

The second screen is used to move the servo motors manually to a certain position, memorise positions and explore these positions. Also memorized positions can be deleted.

The keypad on the center with five buttons is used to move the motors forward, behind and move the two servomotors to the start position (buttons from 11 to 15 . This five buttons make up an object named "KeyPad".

The button below to the right (button 16) is used to store the servomotors current position.

The upper right button erases the stored positions (button 17).

The top left button (button 18) after get positions is used to cross the memorized positions one at a time stopping at each position on forward direction. The button below left (button 19) has the same function in reverse direction. This two buttons. Step buttons (button 18 & 19) doesn't work if no more than two positions are stored on memory.

The blue buttons on the bottom are used to move through the screens. The characteristics of these buttons are explained in the section 8.6.

In the information windows we can see:

- On the top right the actual position and actual velocity of the two servomotors. This window make up the "ShowPositionVelocity" object.

- On the top left the current memory position to stored "Stored Position button" and the current position when the Step buttons are used "Step Position button". This window make up the "ShowPositionStep" object.

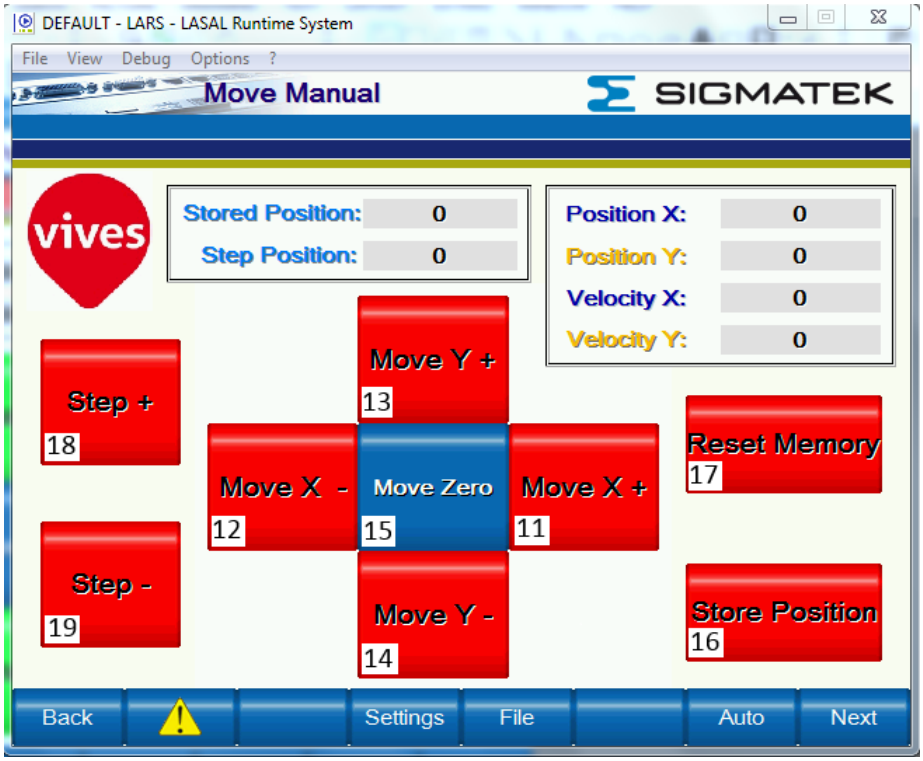


Figure 8.22: Move Manual & Get Positions Screen

8.2.1 Movement Control Buttons

- Button 11 Component Properties

1. Data Properties

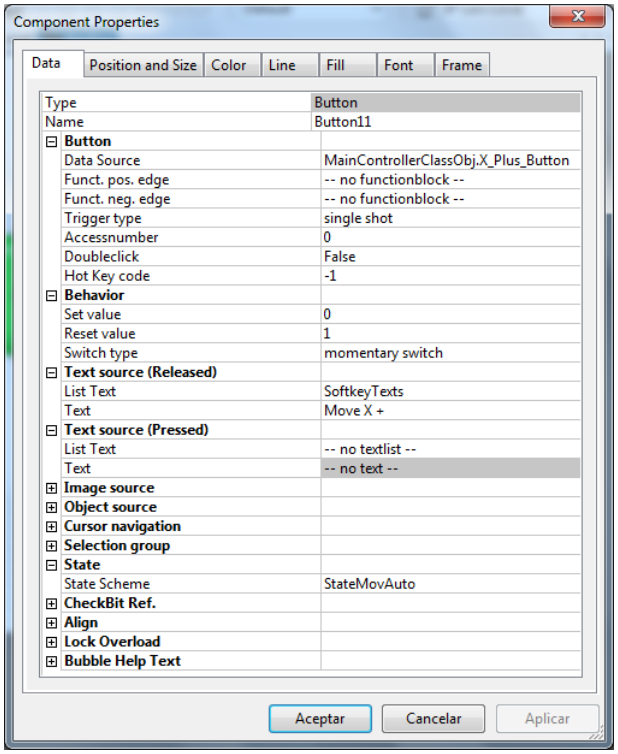


Figure 8.23: Button 11 Data Properties

2. Color Properties

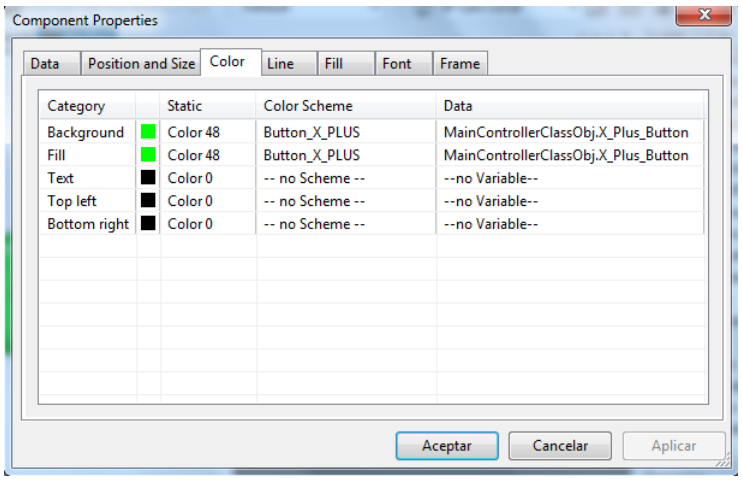


Figure 8.24: Button 11 Color Properties

- Button 12 Component Properties

1. Data Properties

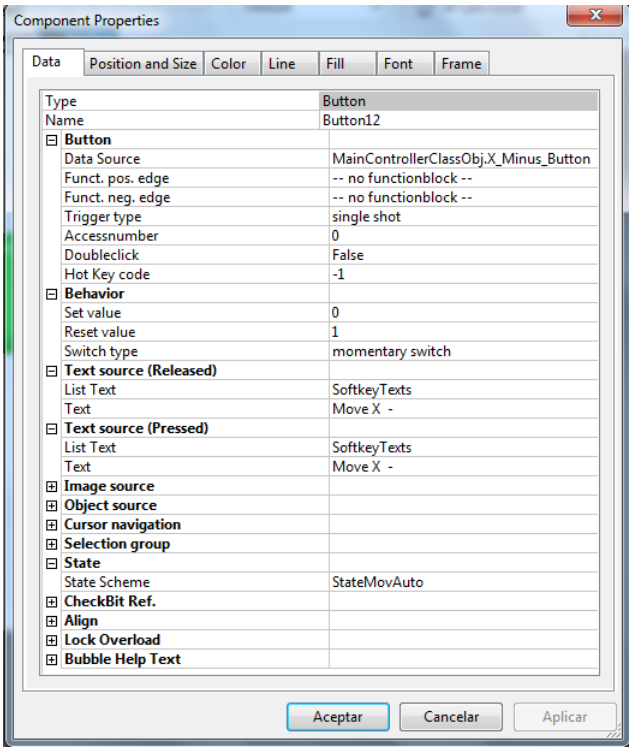


Figure 8.25: Button 12 Data Properties

2. Color Properties

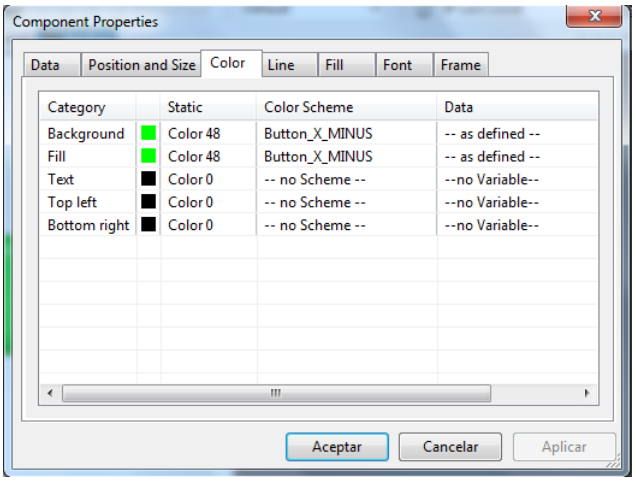


Figure 8.26: Button 12 Color Properties

- Button 13 Component Properties

1. Data Properties

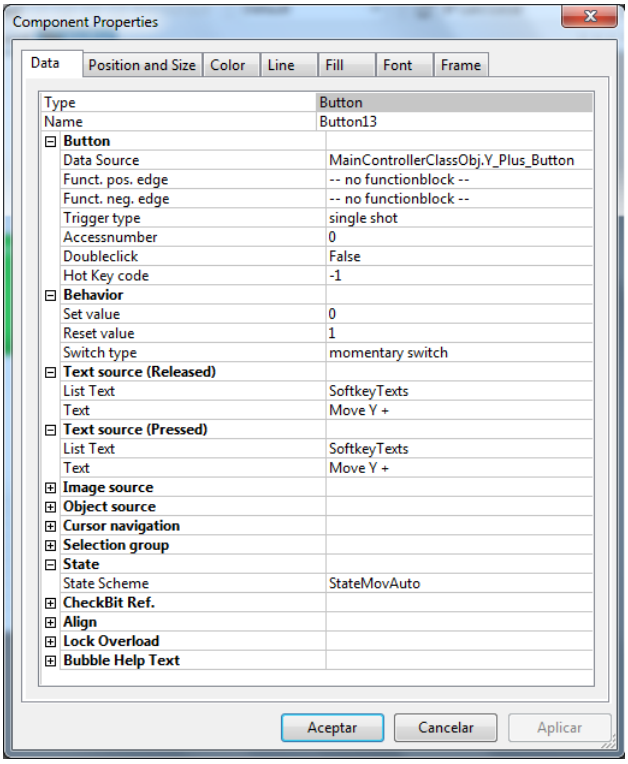


Figure 8.27: Button 13 Data Properties

2. Color Properties

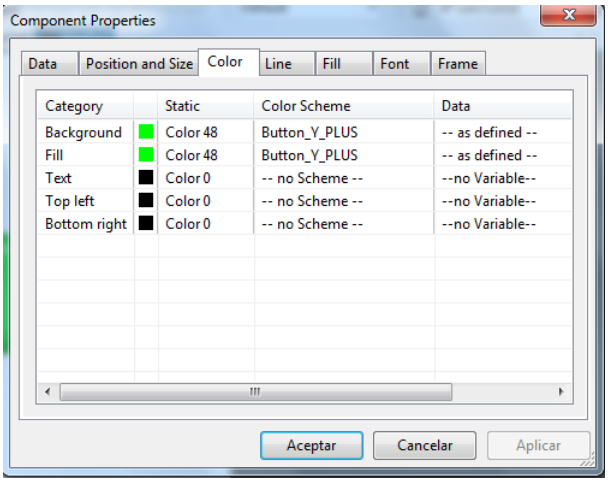


Figure 8.28: Button 13 Color Properties

- Button 14 Component Properties

1. Data Properties

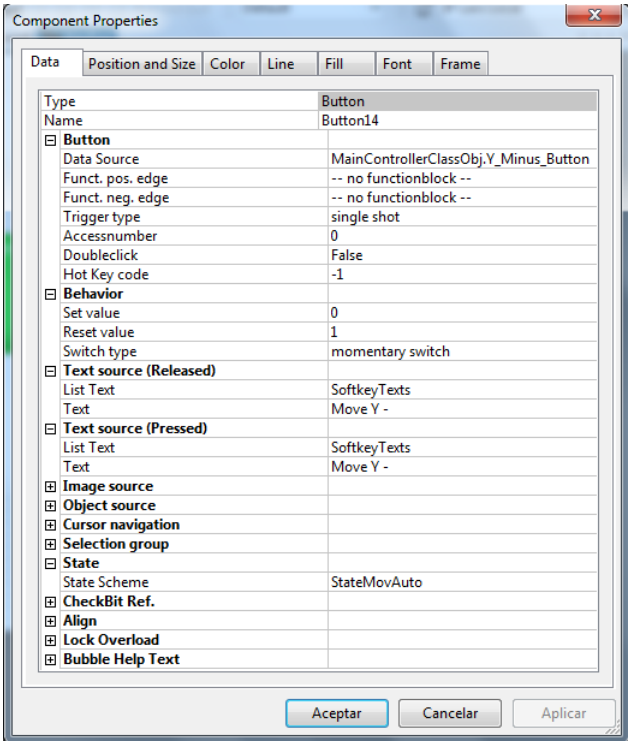


Figure 8.29: Button 14 Data Properties

2. Color Properties

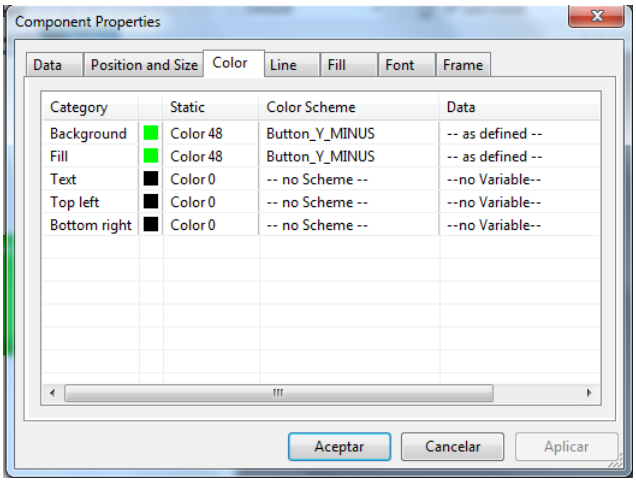


Figure 8.30: Button 14 Color Properties

- Button 15 Component Properties

1. Data Properties

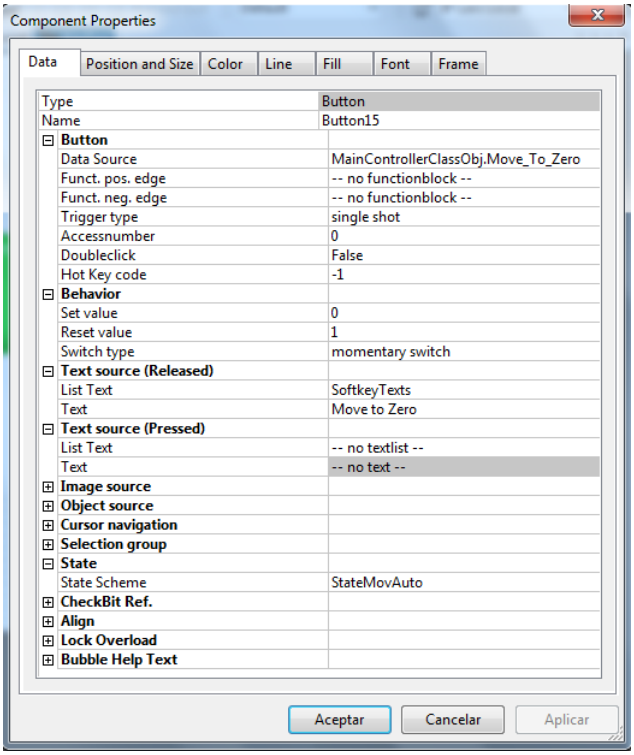


Figure 8.31: Button 15 Data Properties

2. Color Properties

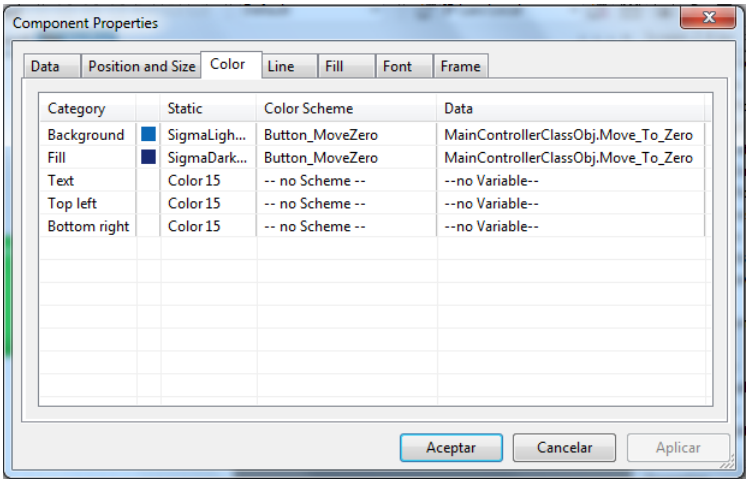


Figure 8.32: Button 15 Color Properties

8.2.2 Store Position Button

- Button 16 Component Properties

1. Data Properties

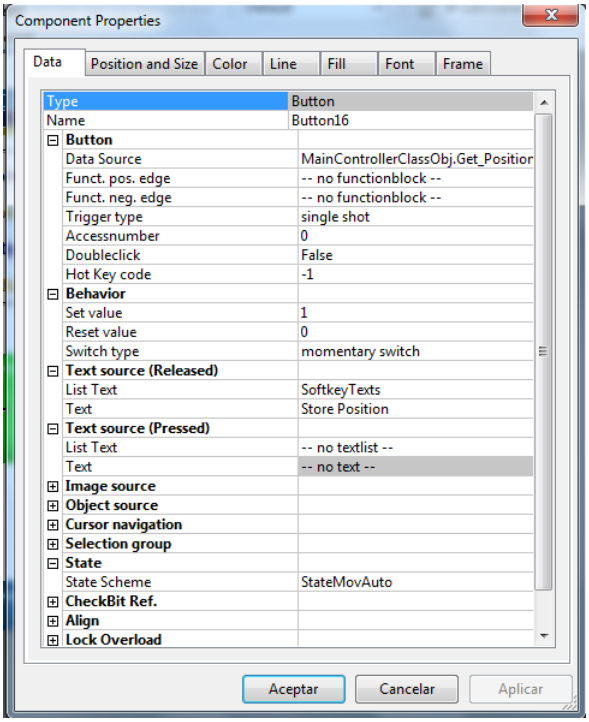


Figure 8.33: Button 16 Data Properties

2. Color Properties

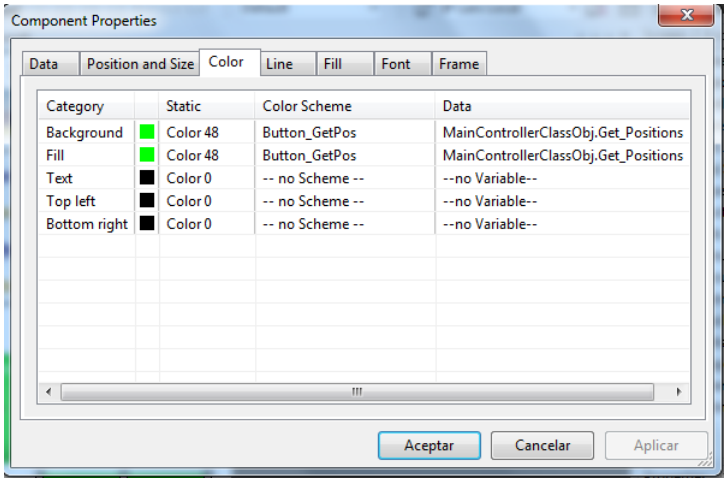


Figure 8.34: Button 16 Color Properties

8.2.3 Reset Memory Button

- Button 17 Component Properties

1. Data Properties

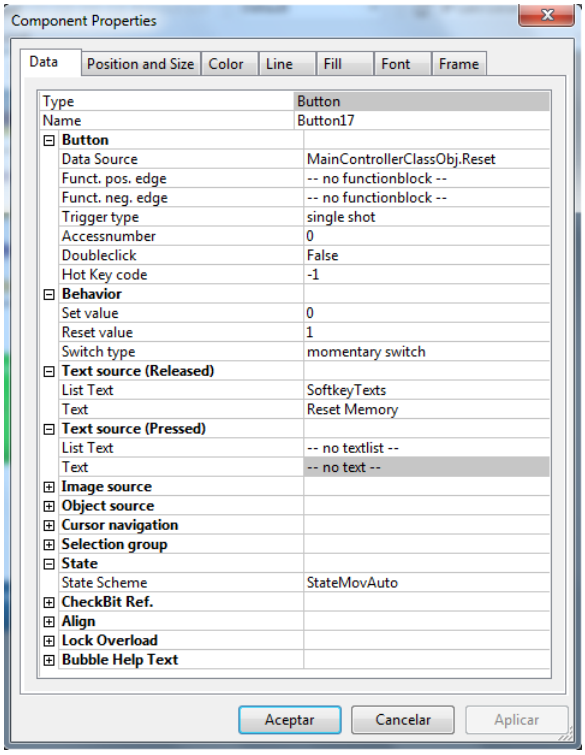


Figure 8.35: Button 17 Data Properties

2. Color Properties

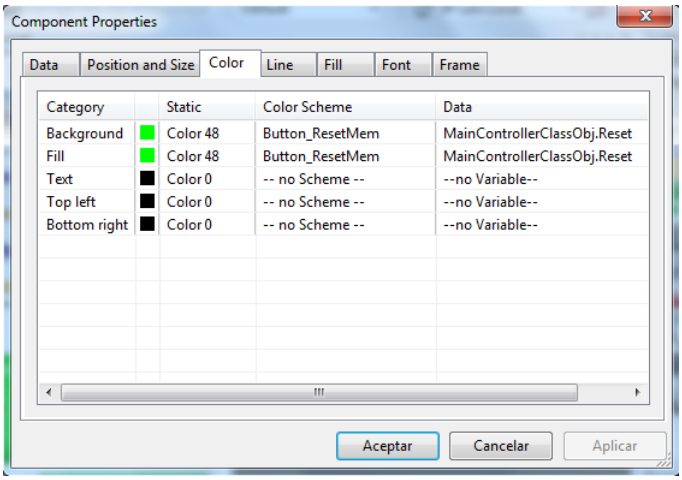


Figure 8.36: Button 17 Color Properties

8.2.4 Stepper movements Buttons

- Button 18 Component Properties

1. Data Properties

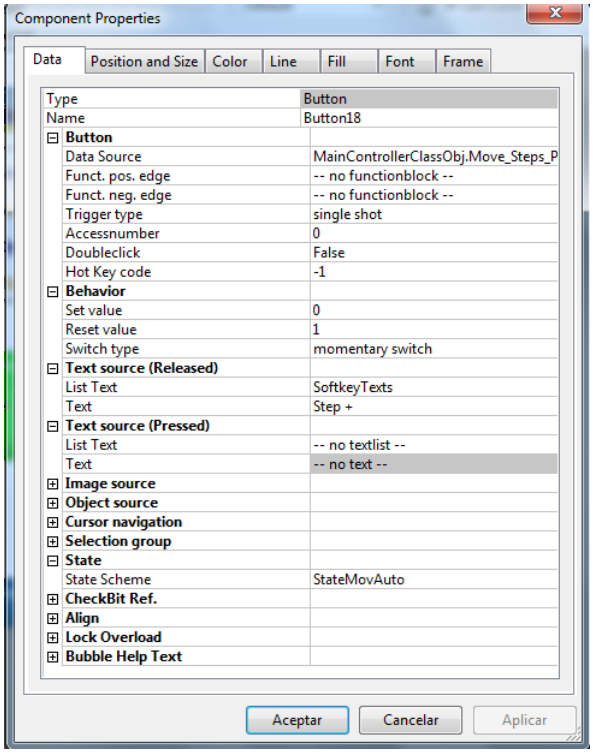


Figure 8.37: Button 18 Data Properties

2. Color Properties

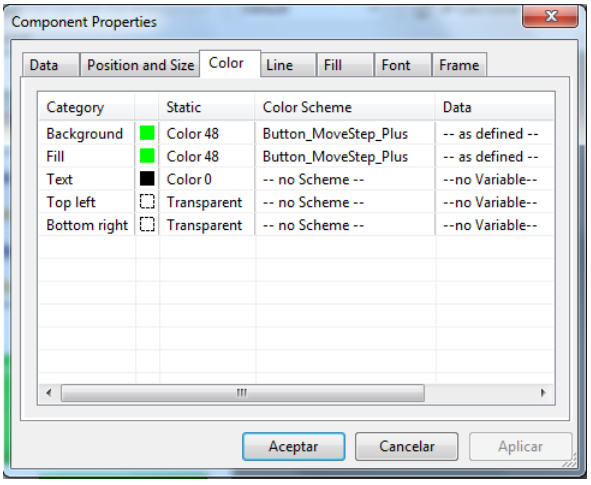


Figure 8.38: Button 18 Color Properties

- Button 19 Component Properties

1. Data Properties

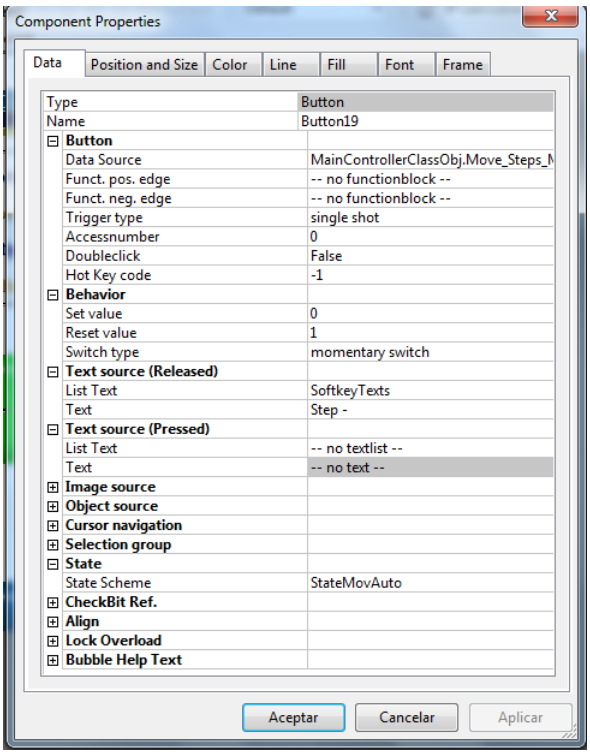


Figure 8.39: Button 19 Data Properties

2. Color Properties

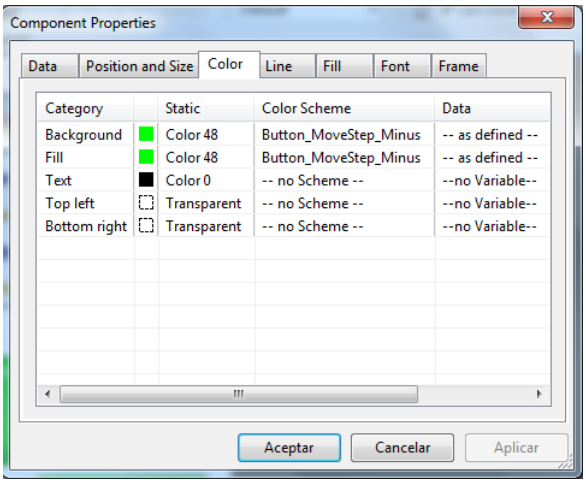


Figure 8.40: Button 19 Color Properties

8.3 Screen3 features (Automatic Movement)

The Figure 8.41 shows how looks the third screen of Lasal Screen project.

In the third window are placed the buttons for the automatic movement. The left button (button 20) start or stop the automatic movement in loop when there are more than two positions stored on memory. Each time it is activated the movement starts from the first memory location. When Button 20 is active buttons on screens 1 (except the Power On button), screen 2 and screen 4 are blocked using the state scheme "AtateMoveAuto".

Right button (button 21) pauses the automatic movement and restart the movement from the current position.

The blue buttons on the bottom are used to move through the screens. The characteristics of these buttons are explained in the section 8.6.

In the information windows we can see:

- On the top right the actual position and actual velocity of the two servomotors. This window make up the "ShowPositionVelocity" object.

- On the top left are shown the destinations to reach by the servo motors during the automatic movement. This window make up the "Destinations" object.

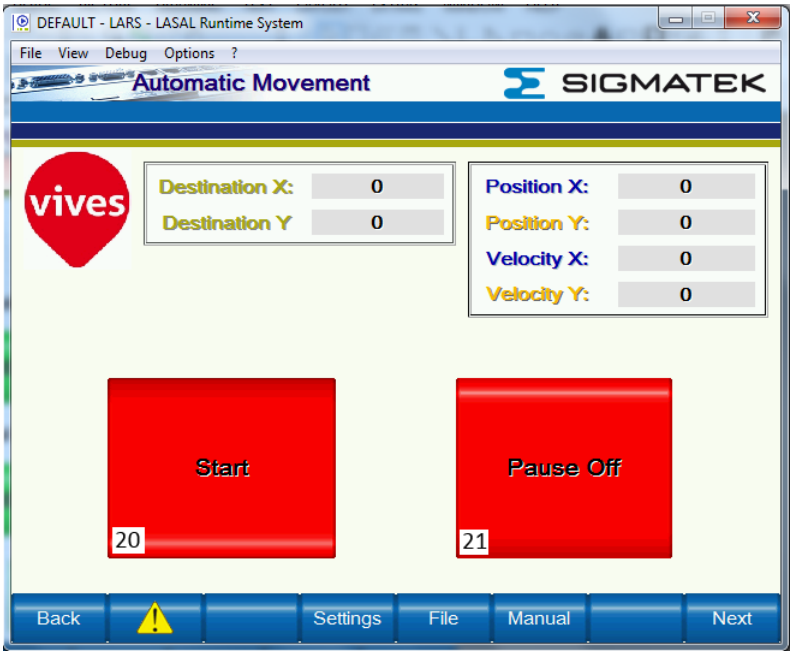


Figure 8.41: Automatic Movement Screen

8.3.1 Automatic Movement Buttons

- Button 20 Component Properties

1. Data Properties

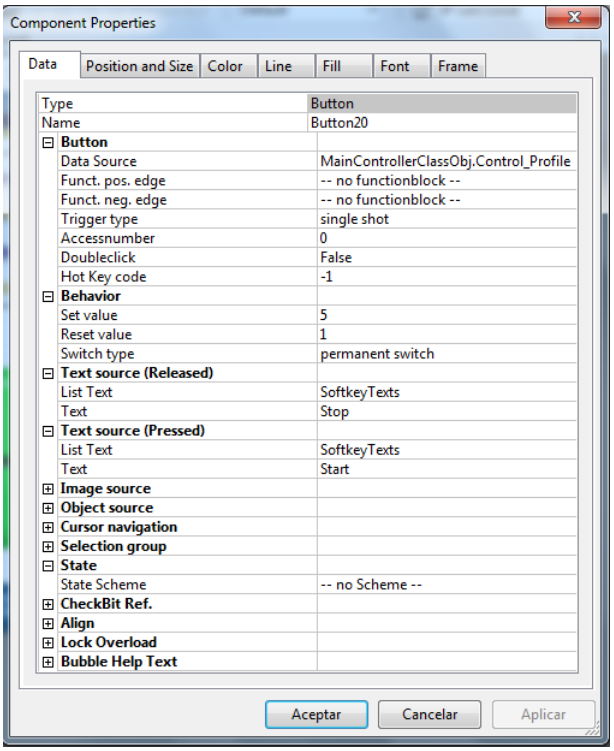


Figure 8.42: Button 20 Data Properties

2. Color Properties

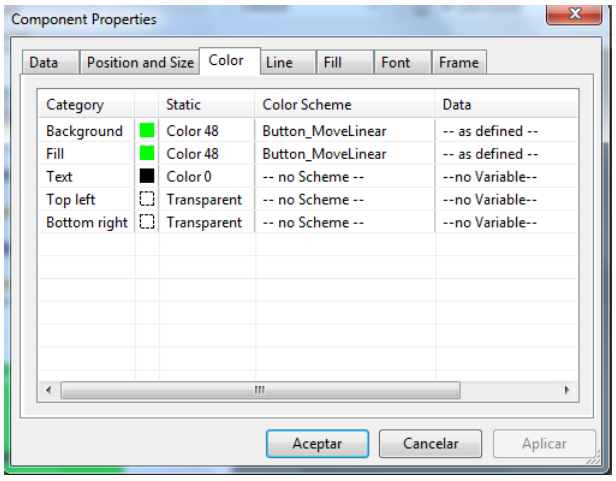


Figure 8.43: Button 20 Color Properties

- Button 21 Component Properties

1. Data Properties

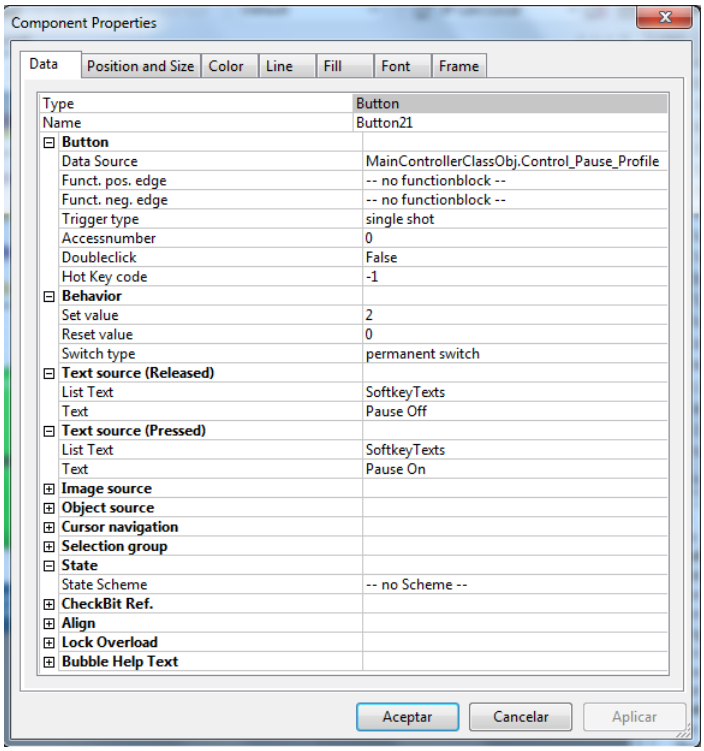


Figure 8.44: Button 21 Data Properties

2. Color Properties

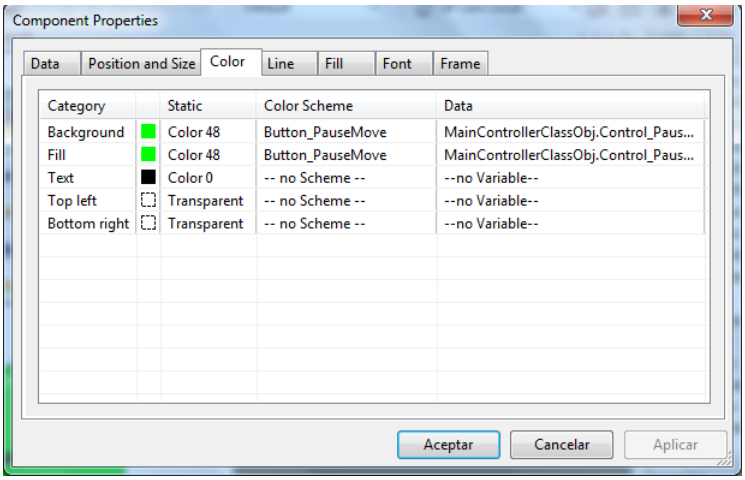


Figure 8.45: Button 21 Color Properties

8.4 Screen4 features (File Options)

On the fourth screen are placed the file options. Right button store the memorized positions in a file on the intelligent terminal memory. The left button read positions from this file to perform the automatic movement through these positions. The blue buttons on the bottom are used to move through the screens. The characteristics of these buttons are explained in the section 8.6.

In the information windows we can see:

- On the top the code error when error occurred creating or reading the file.
- On the bottom the code error when error occurred writing the file.

This window make up the "ShowErrorFileOP" object.

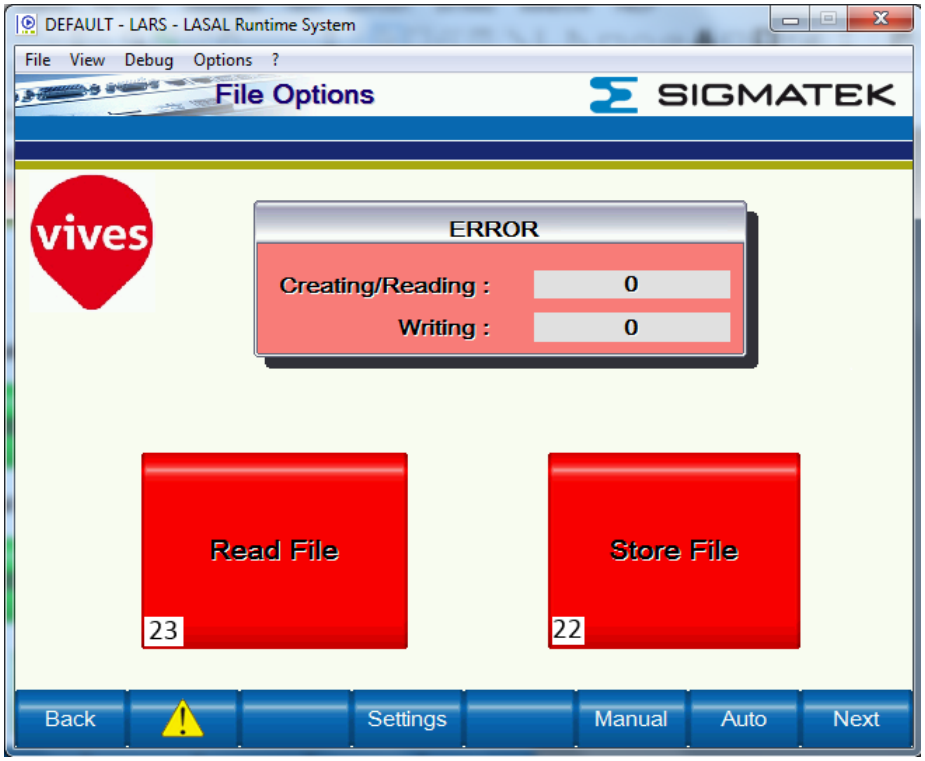


Figure 8.46: File Options Screen

8.4.1 File Options Buttons

- Button 22 Component Properties

1. Data Properties

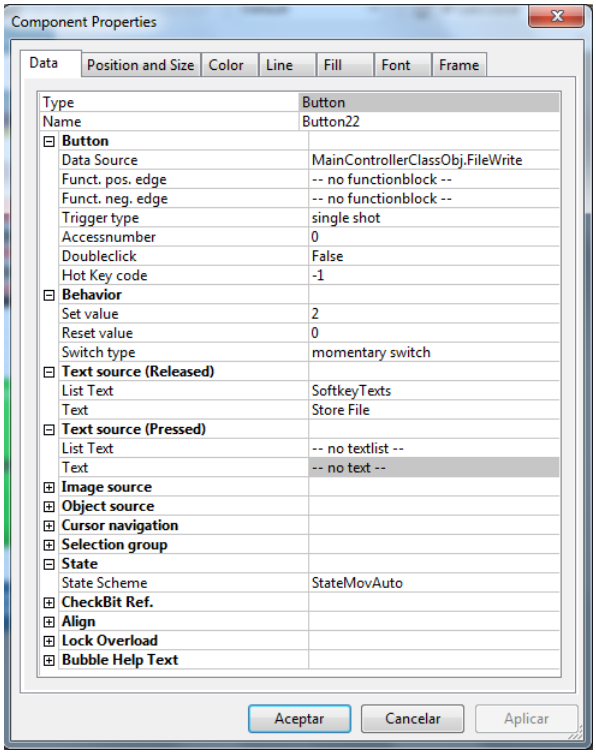


Figure 8.47: Button 22 Data Properties

2. Color Properties

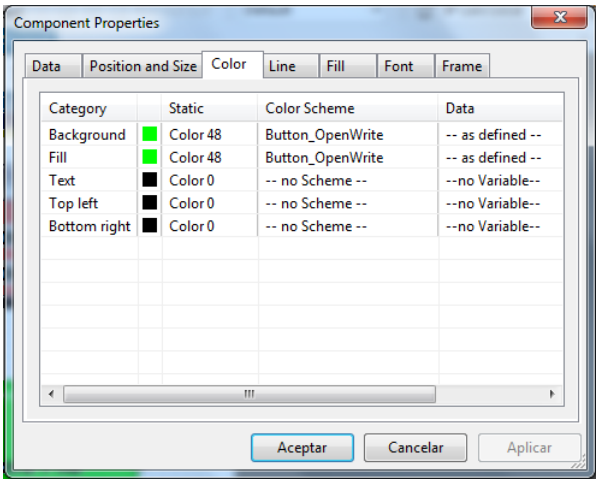


Figure 8.48: Button 22 Color Properties

- Button 23 Component Properties

1. Data Properties

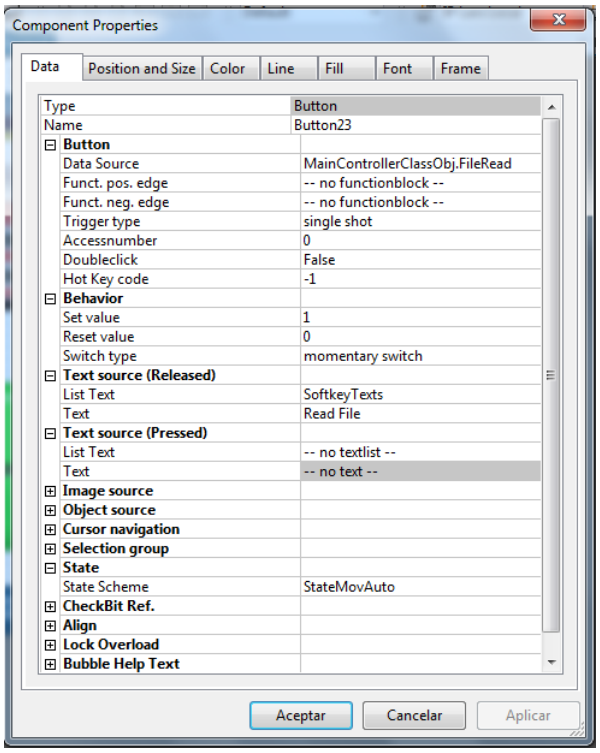


Figure 8.49: Button 23 Data Properties

2. Color Properties

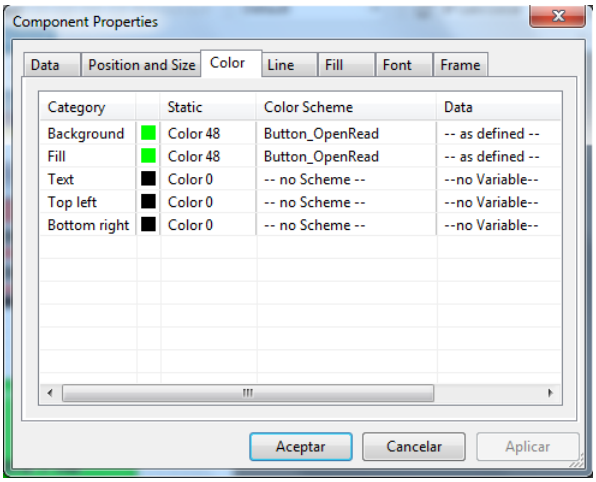


Figure 8.50: Button 23 Color Properties

8.5 Objects

8.5.1 "KeyPad" Object

KeyPad Object is composed by five buttons (Button 11, Button 12, Button 13, Button 14 & Button 15). The features of this buttons are shown on subsection 8.2.1

The keypad object is used to move the motors forward, behind and move the two servomotors to the start position.

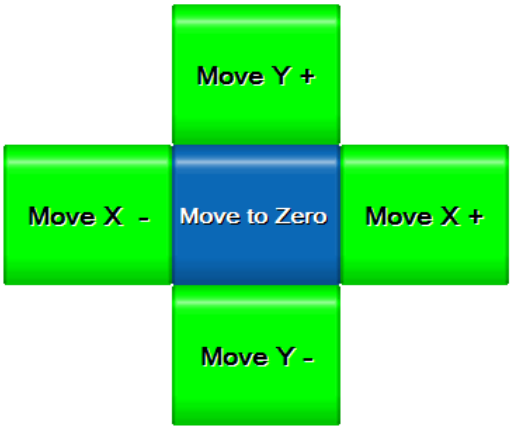


Figure 8.51: "KeyPad" Object

8.5.2 "ShowPositionVelocity" Object

"ShowPositionVelocity" Object show the actual position and actual velocity of the two servomotors.The source of the information is collected from LMCAxis object.

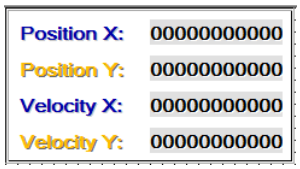


Figure 8.52: "ShowPositionVelocity" Object

- Position X Data

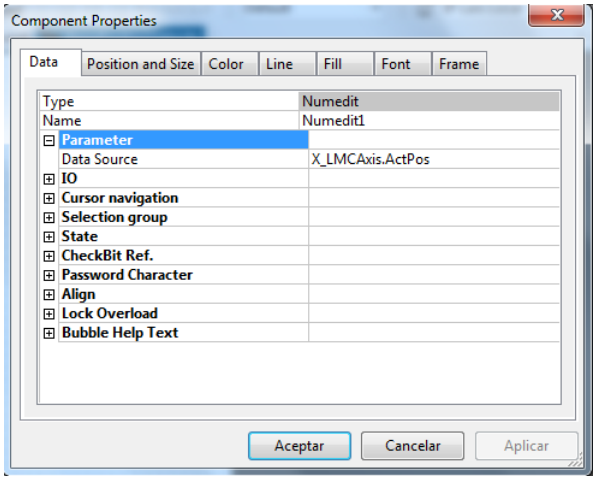


Figure 8.53: Position X Data

- Position Y Data

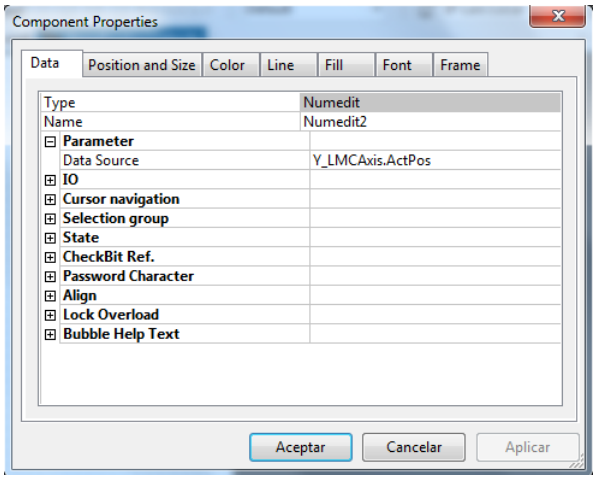


Figure 8.54: Position Y Data

- Velocity X Data

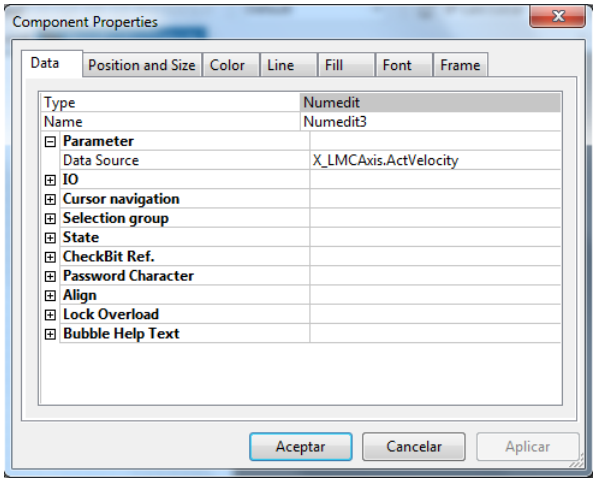


Figure 8.55: Velocity X Data

- Velocity Y Data

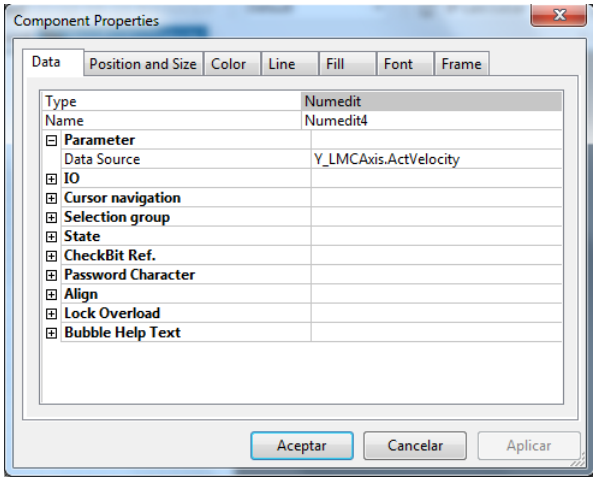


Figure 8.56: Velocity Y Data

8.5.3 "Destinations" Object

Destinations object show the destinations to reach by the servo motors during the automatic movement. The source of the information is collected from LMCProfile object.

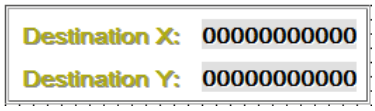


Figure 8.57: "Destinations" Object

- Destination X Data

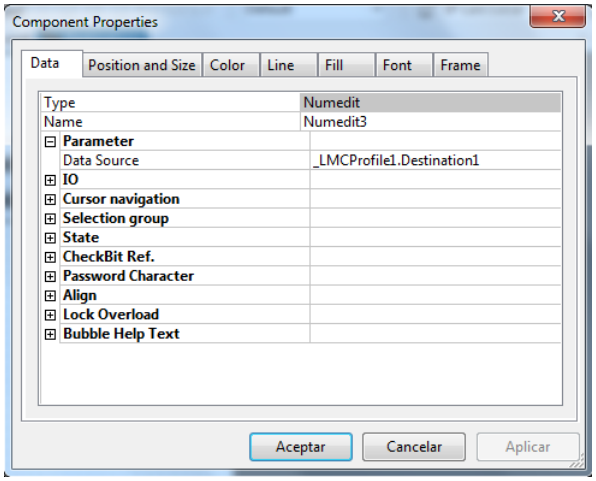


Figure 8.58: Destination X Data

- Destination Y Data

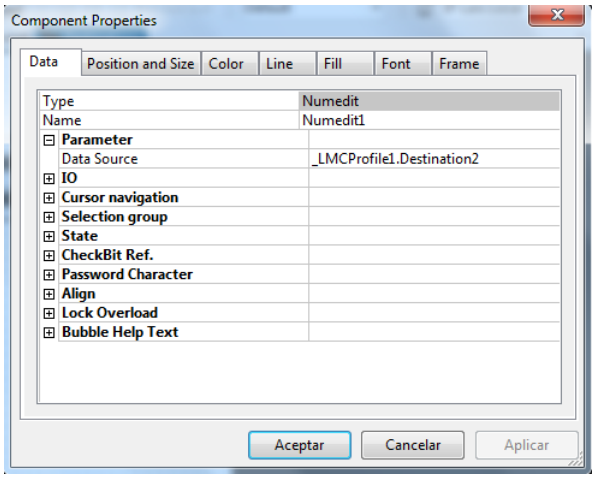


Figure 8.59: Destination Y Data

8.5.4 "ShowPositionStep" Object

ShowPositionStep object show the current memory position to stored (StoredPosition Data) and the current position when the Step buttons are used (Step Position Data). The source of the information is collected from MainControllerClass object.

Figure 8.60: "ShowPositionStep" Object

- Stored Position Data

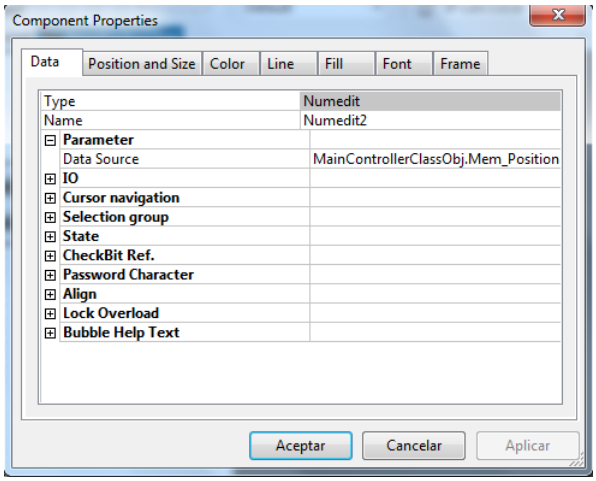


Figure 8.61: Stored Position Data

- Move Step Data

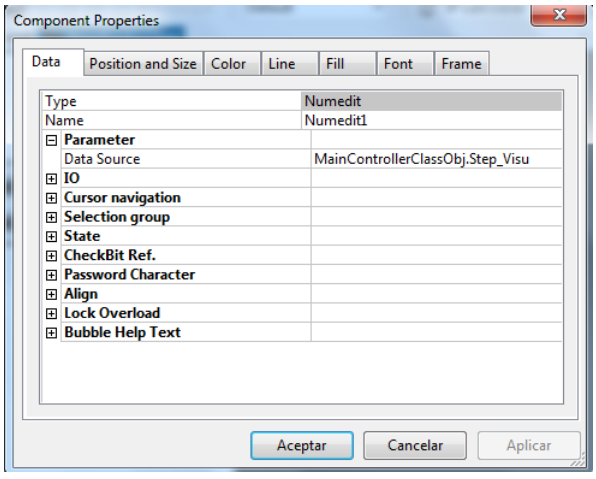


Figure 8.62: Move Step Data

8.5.5 "ShowErrorFileOP" Object

ShowErrorFileOP object show the code error when error occurred creating or reading the file (Creating/Reading Data) and the code error when error occurred writing the file (Writing Data) The source of the information is collected from MainControllerClass object.

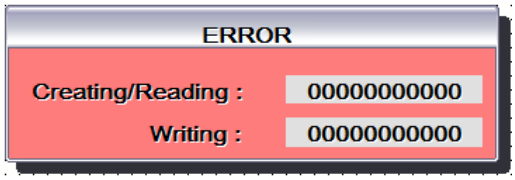


Figure 8.63: "ShowErrorFileOP" Object

- Creating/Reading Error Data

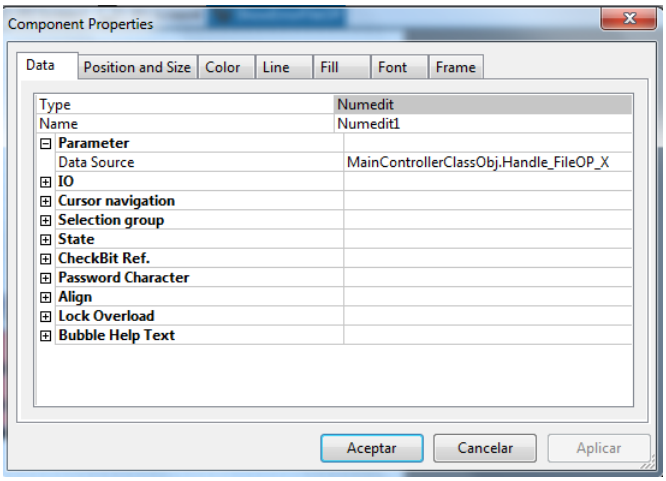


Figure 8.64: Creating/Reading Error Data

- Writing Error Data

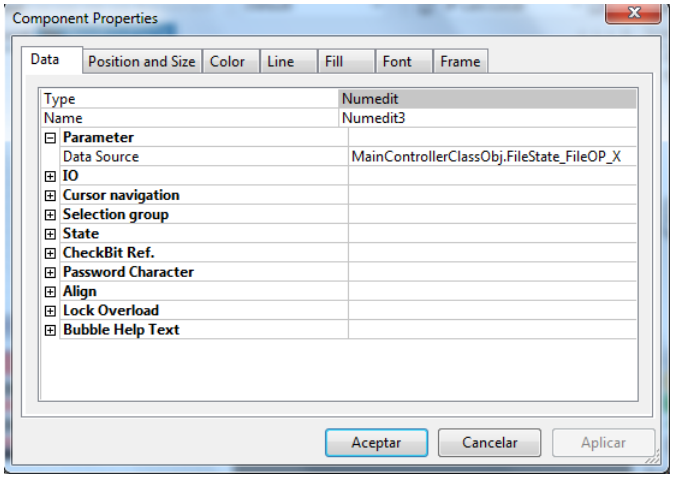


Figure 8.65: Writing Error Data

8.6 Screen Change Buttons

The blue buttons on the bottom of each screen are used to move through the different screens. This buttons use the function blocks shown in section 8.7 and are numbered from left to right from 0 to 7.

8.6.1 Screen 1

Figure 8.66: Screen Change Buttons (Screen 1)

1. **Softkey0 Properties**(Not used)

2. **Softkey1 Properties**

   Softkey1 active the alarm screen. This screen is not used in this program.

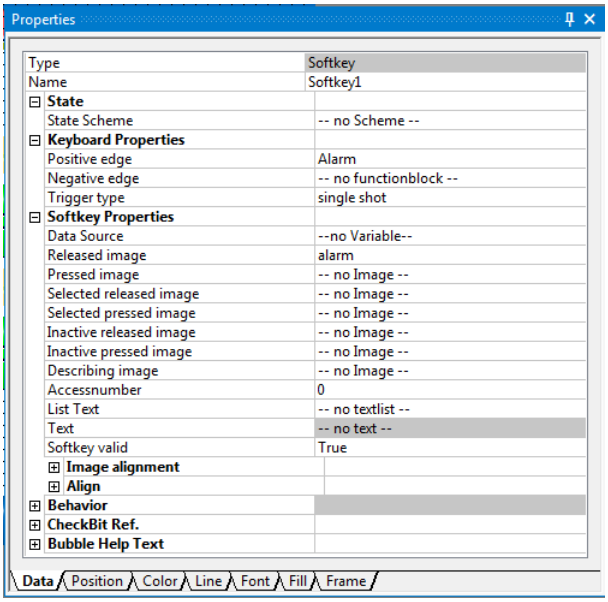


Figure 8.67: Softkey 1 Data

3. **Softkey2 Properties** (Not used)
4. **Softkey3 Properties** (Not used)
5. **Softkey4 Properties**

   Softkey4 active the "File Options Screen" (screen 4).

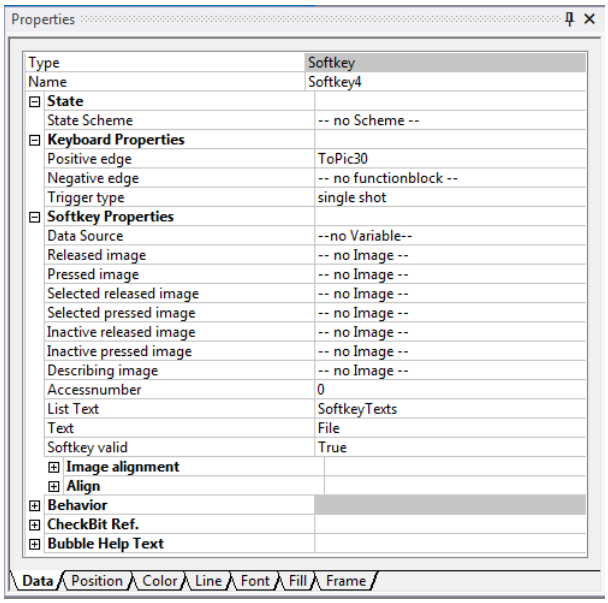


Figure 8.68: Softkey 4 Data

6. **Softkey5 Properties**

Softkey5 active the "Move Manual & Get Positions" (Screen 2).

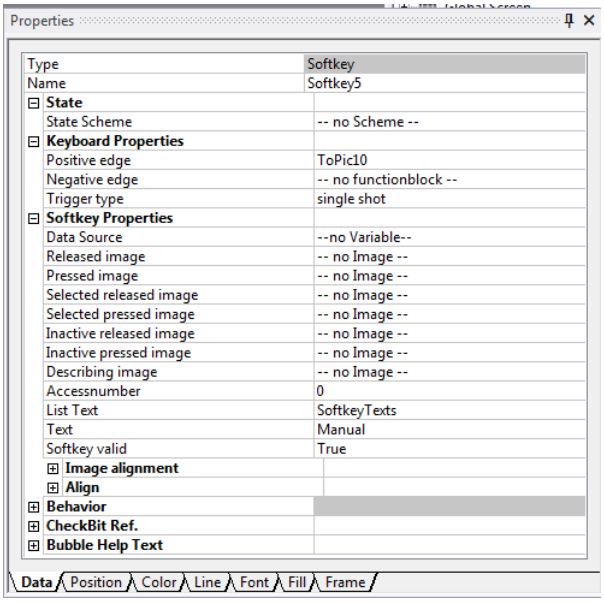


Figure 8.69: Softkey 5 Data

7. **Softkey6 Properties**

Softkey6 active the "Automatic Movement" screen (screen 3).

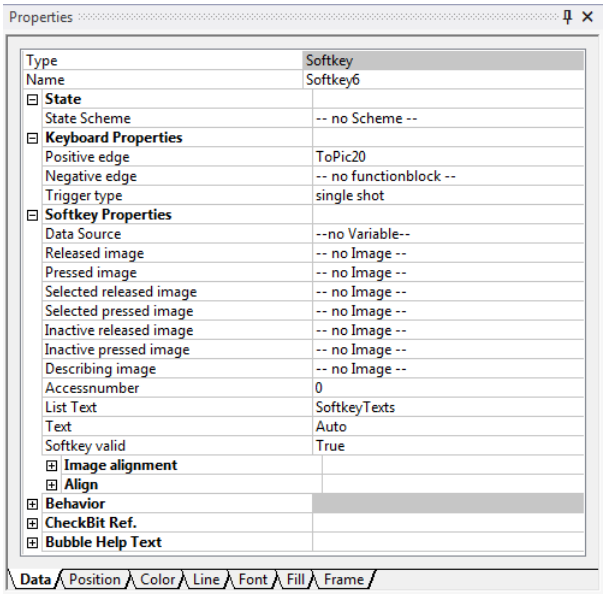


Figure 8.70: Softkey 6 Data

8. **Softkey7 Properties**

   Softkey7 activate the next screen (screen 2).

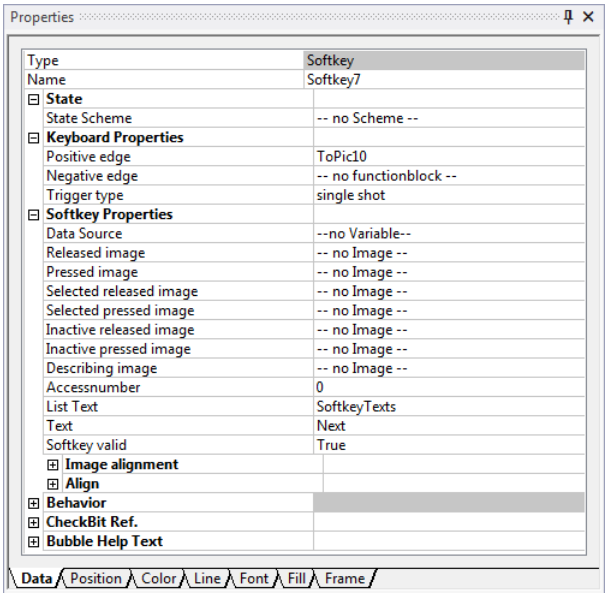


Figure 8.71: Softkey 7 Data

8.6.2 Screen 2

Figure 8.72: Screen Change Buttons (Screen 2)

1. **Softkey0 Properties**

   Softkey0 activate the previous screen (screen 1)

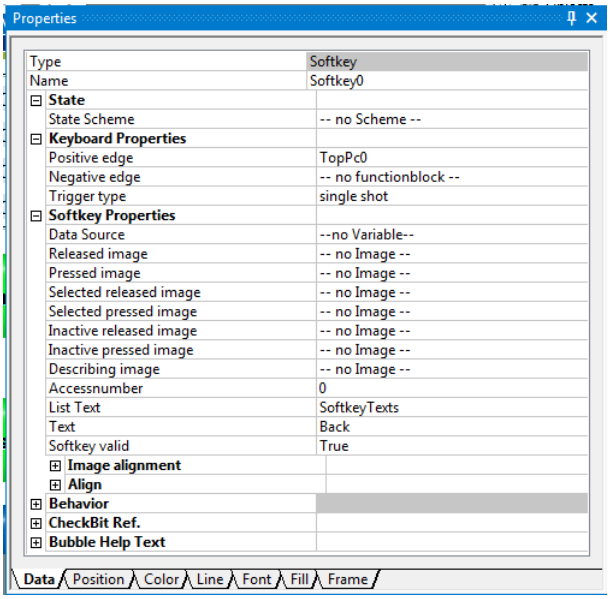


Figure 8.73: Softkey 0 Data

2. **Softkey1 Properties**

   Softkey1 active the alarm screen. This screen is not used in this program.

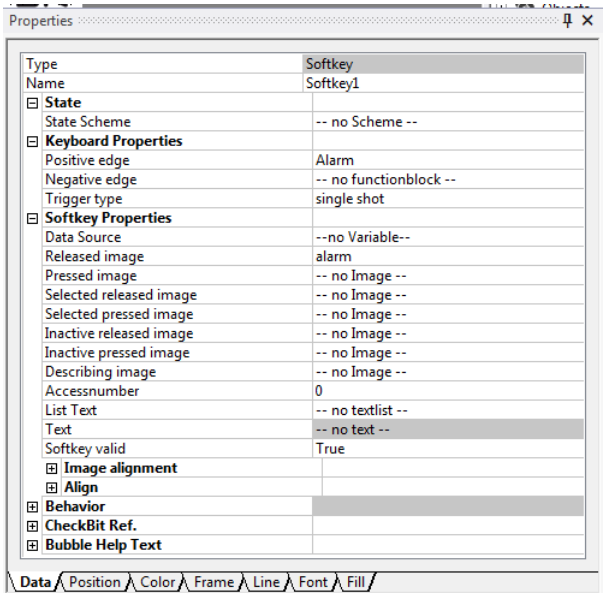


Figure 8.74: Softkey 1 Data

3. **Softkey2 Properties** (Not used)

4. **Softkey3 Properties**

   Softkey3 active the "Movement Settings" screen (screen 1).

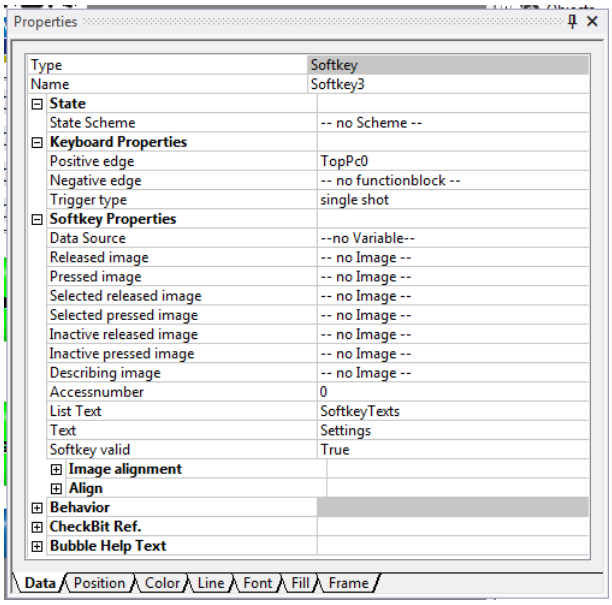


Figure 8.75: Softkey 3 Data

5. **Softkey4 Properties**

   Softkey4 active the "File Options" screen (screen 4).

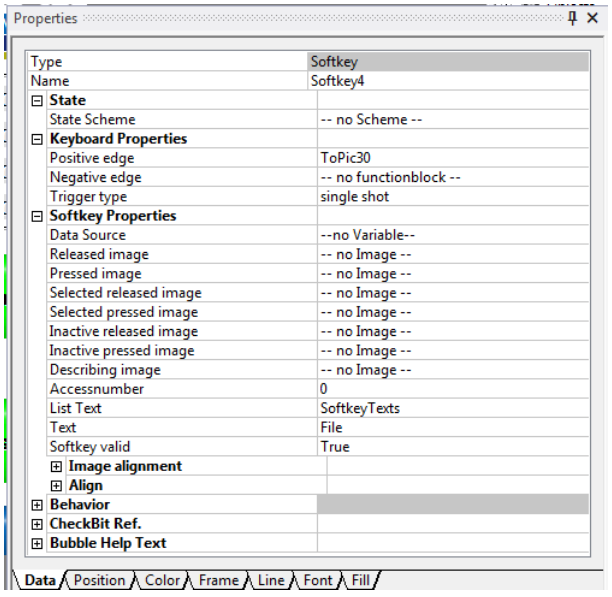


Figure 8.76: Softkey 4 Data

6. **Softkey5 Properties** (Not used)

7. **Softkey6 Properties**

Softkey6 active the "Automatic Movement" screen (screen 3).

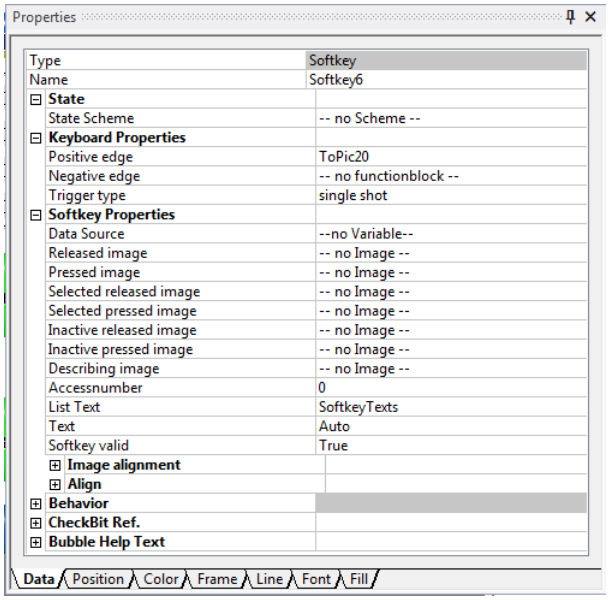


Figure 8.77: Softkey 6 Data

8. **Softkey7 Properties**

Softkey7 activate the next screen (screen 3)

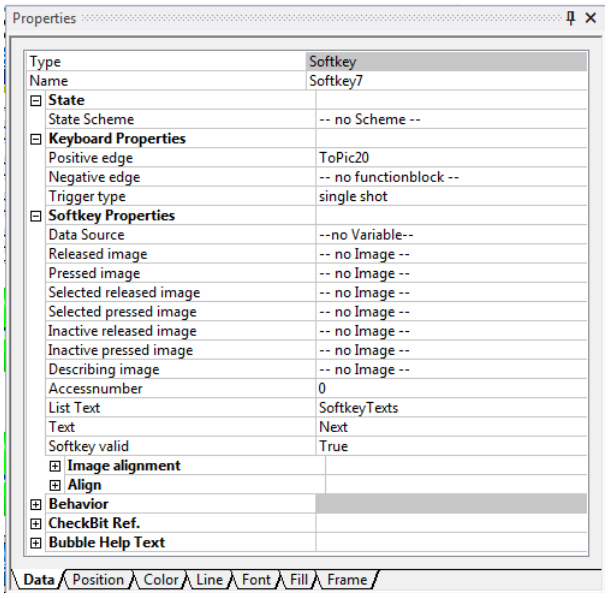


Figure 8.78: Softkey 7 Data

8.6.3 Screen 3

Figure 8.79: Screen Change Buttons (Screen 3)

1. **Softkey0 Properties**

   Softkey0 activate the previous screen (screen 2)

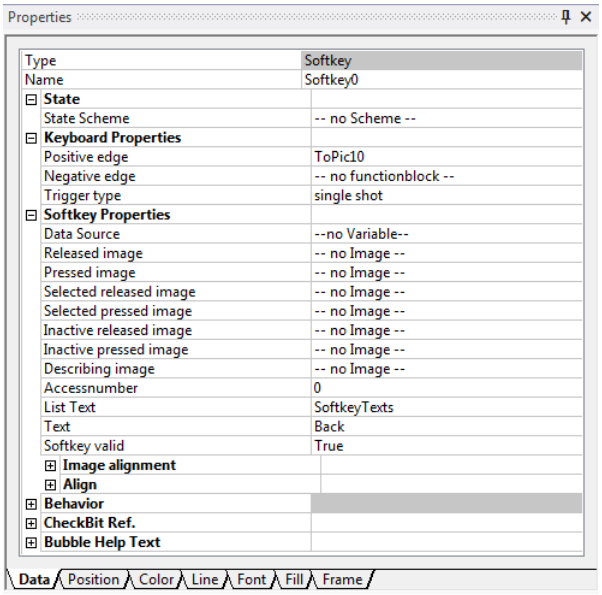


Figure 8.80: Softkey 0 Data

2. **Softkey1 Properties**

   Softkey1 active the alarm screen. This screen is not used in this program.

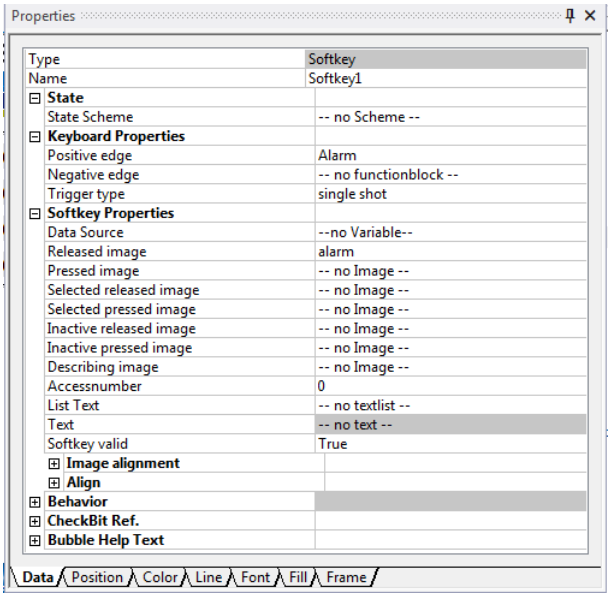


Figure 8.81: Softkey 1 Data

3. **Softkey2 Properties** (Not used)

4. **Softkey3 Properties**

   Softkey3 active the "Movement Settings" screen (screen 1).

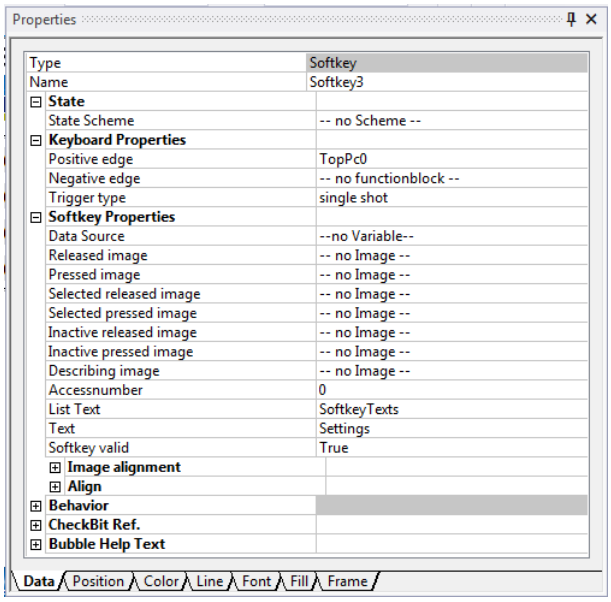


Figure 8.82: Softkey 3 Data

5. **Softkey4 Properties**

   Softkey4 active the "File Options" screen (screen 4).

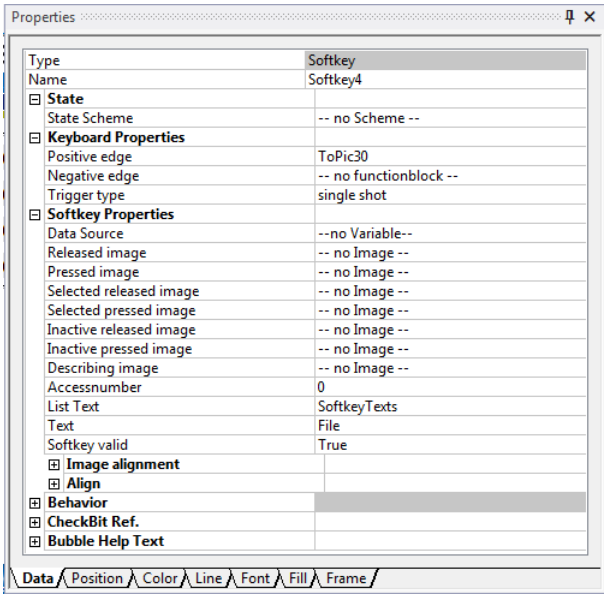


Figure 8.83: Softkey 4 Data

6. **Softkey5 Properties**

   Softkey5 active the "Move Manual & Get Positions" (Screen 2).

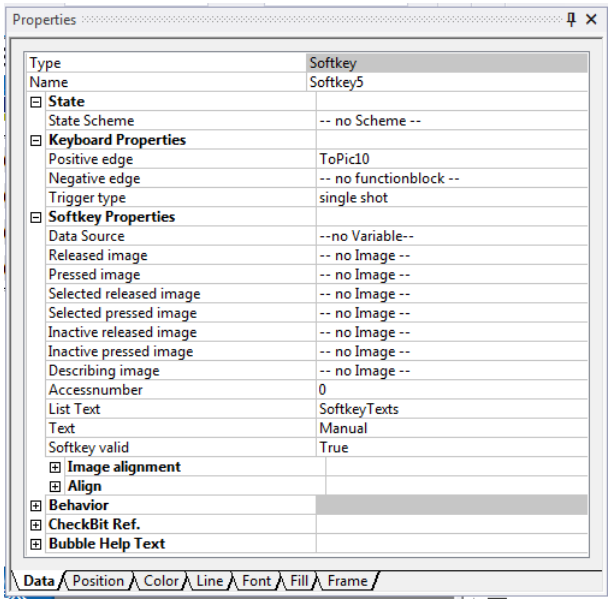


Figure 8.84: Softkey 5 Data

7. **Softkey6 Properties** (Not used)

8. **Softkey7 Properties**

   Softkey7 activate the next screen (screen 4)

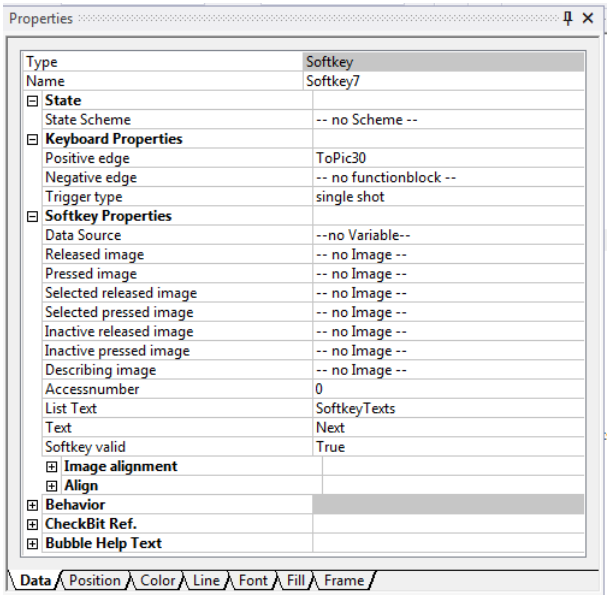


Figure 8.85: Softkey 7 Data

8.6.4 Screen 4

Figure 8.86: Screen Change Buttons (Screen 4)

1. **Softkey0 Properties** Softkey0 activate the previous screen (screen 3)

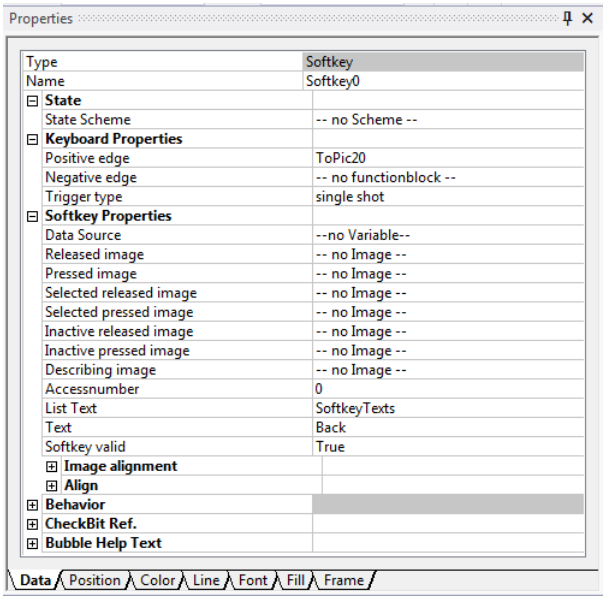


Figure 8.87: Softkey 0 Data

2. **Softkey1 Properties**

   Softkey1 active the alarm screen. This screen is not used in this program.

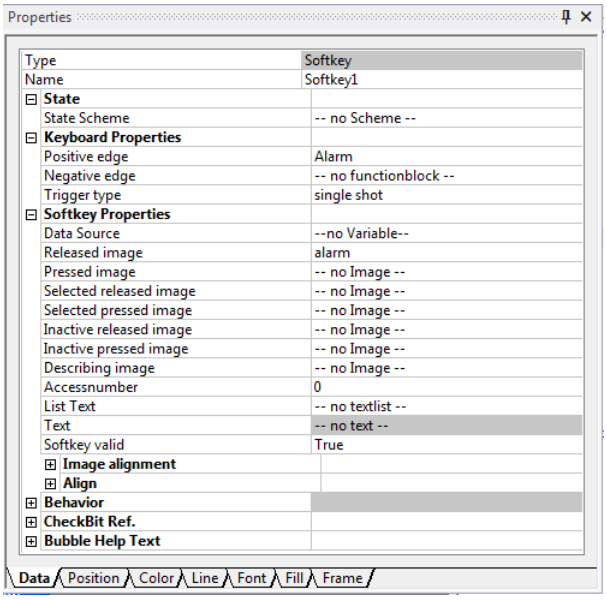


Figure 8.88: Softkey 1 Data

3. **Softkey2 Properties** (Not used)

4. **Softkey3 Properties**

   Softkey3 active the "Movement Settings" screen (screen 1).

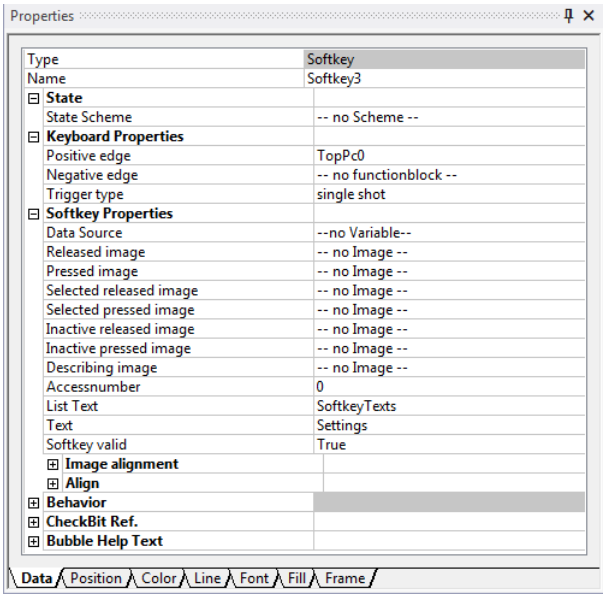


Figure 8.89: Softkey 3 Data

5. **Softkey4 Properties** (Not used)

6. **Softkey5 Properties**

   Softkey5 active the "Move Manual & Get Positions" (Screen 2).

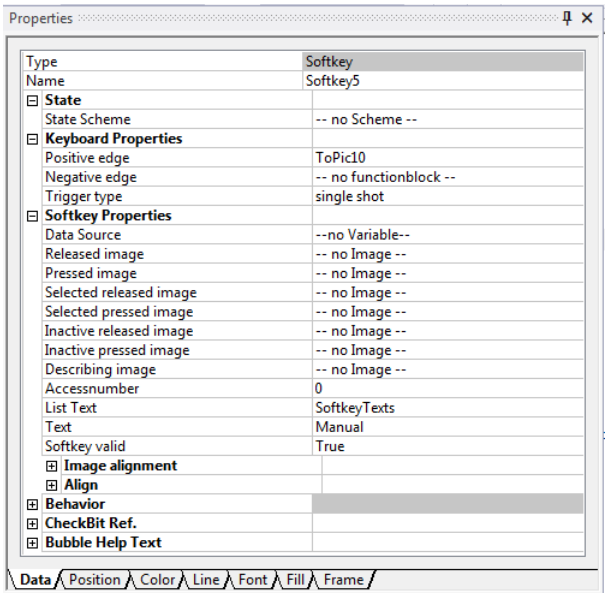


Figure 8.90: Softkey 5 Data

7. **Softkey6 Properties**

Softkey6 active the "Automatic Movement" screen (screen 3).

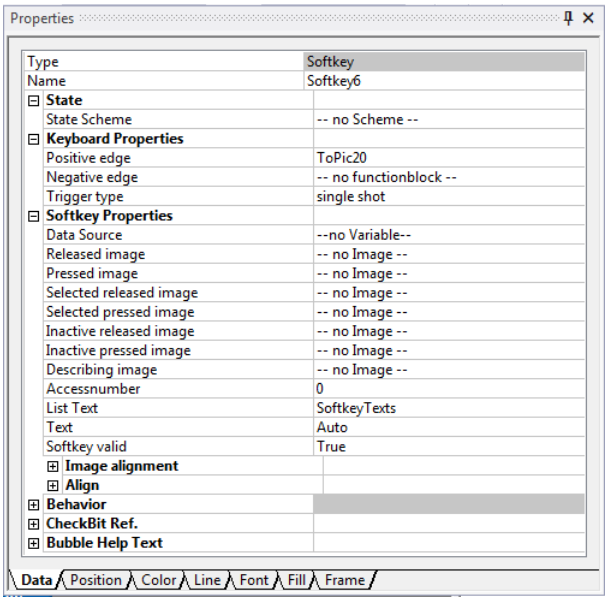


Figure 8.91: Softkey 6 Data

8. **Softkey7 Properties**

Softkey7 activate the next screen (Alarm screen)

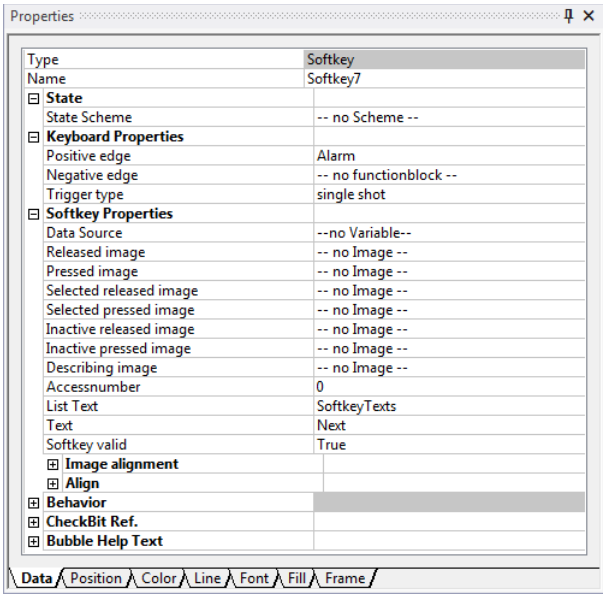


Figure 8.92: Softkey 7 Data

8.6.5 Alarm Screen

Figure 8.93: Screen Change Buttons (Alarm Screen)

1. **Softkey0 Properties**

   Softkey0 activate the previous screen (screen 4).

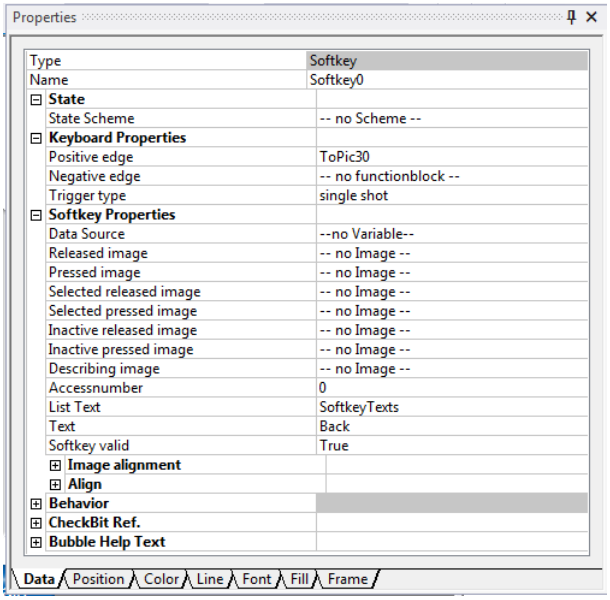


Figure 8.94: Softkey 0 Data

2. **Softkey1 Properties**

   Softkey1 deactivate the alarm.

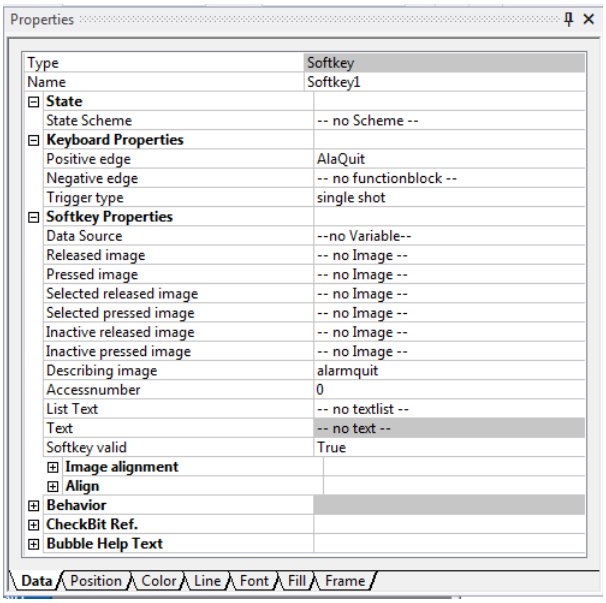


Figure 8.95: Softkey 1 Data

3. **Softkey2 Properties**

   Softkey2 delete the alarm.

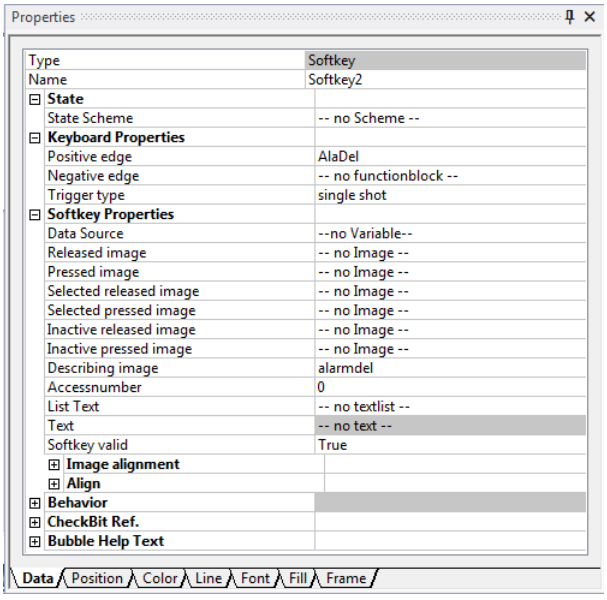


Figure 8.96: Softkey 2 Data

4. **Softkey3 Properties**

   Softkey3 active the "Movement Settings" screen (screen 1).

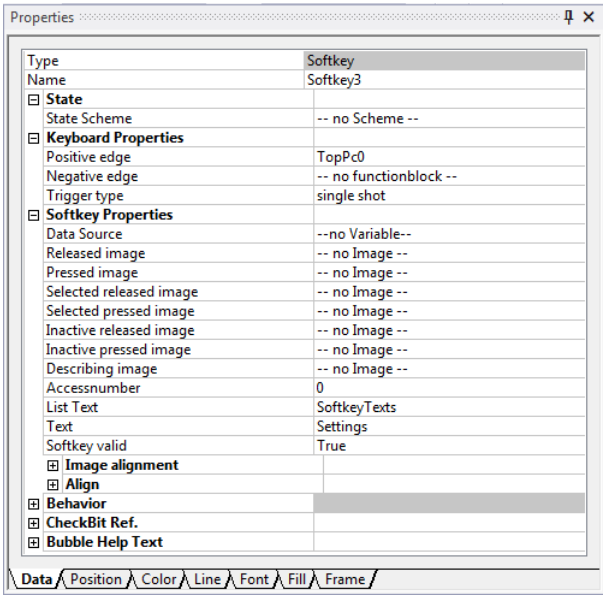


Figure 8.97: Softkey 3 Data

5. **Softkey4 Properties**

   Softkey4 active the "File Options" screen (screen 4).

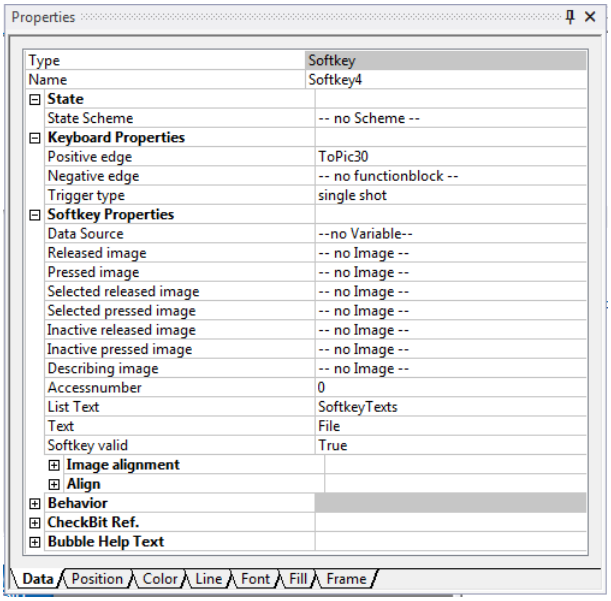


Figure 8.98: Softkey 4 Data

6. **Softkey5 Properties**

   Softkey5 active the "Move Manual & Get Positions" (Screen 2).

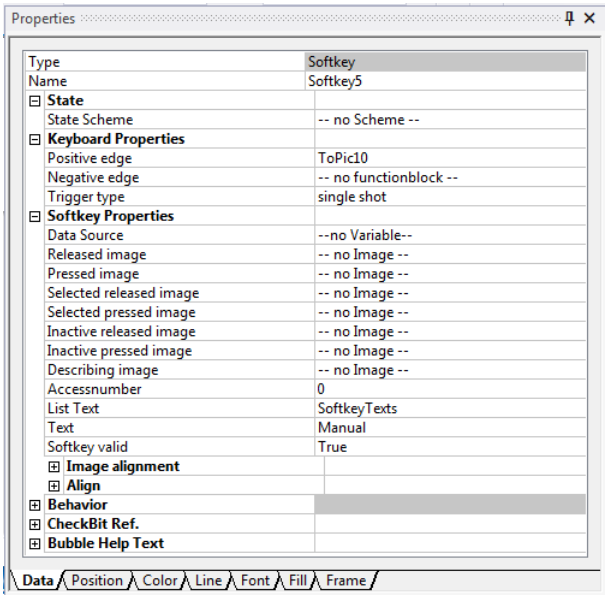


Figure 8.99: Softkey 5 Data

7. **Softkey6 Properties**

   Softkey6 active the "Automatic Movement" screen (screen 3).

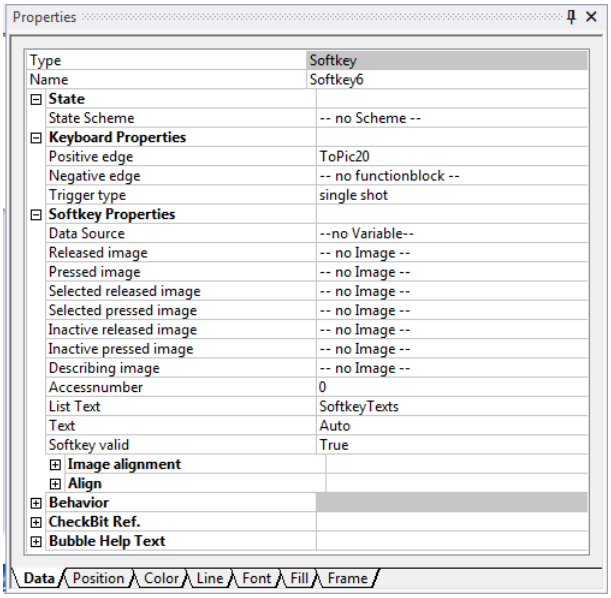


Figure 8.100: Softkey 6 Data

8. **Softkey7 Properties**

Softkey7 active the "Movement Settings" screen (screen 1).

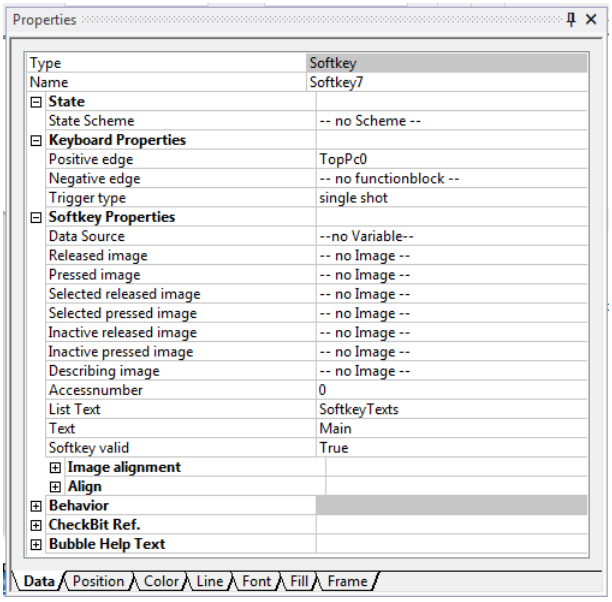


Figure 8.101: Softkey 7 Data

8.7 Function Blocks

The function blocks below are used to changing screens with softkeys.

1. "ToPic0" Features

   "ToPic0" function block activates the screen 1.

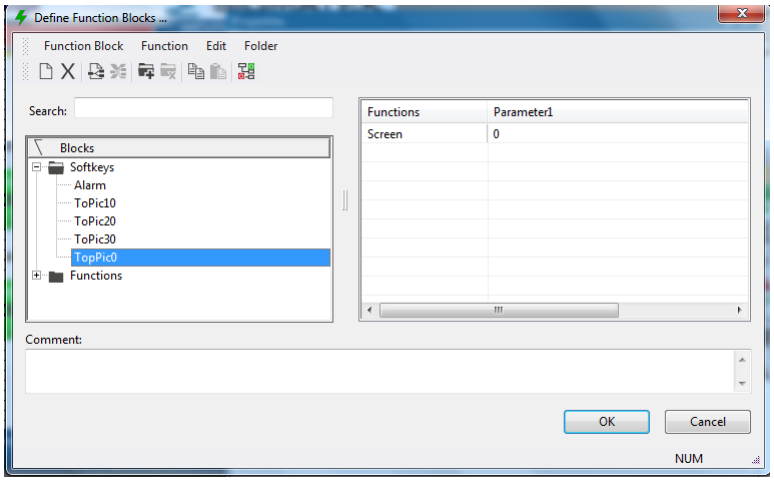


Figure 8.102: "ToPic0" Function Block

2. "ToPic10" Features

   "ToPic10" function block activates the screen 2.

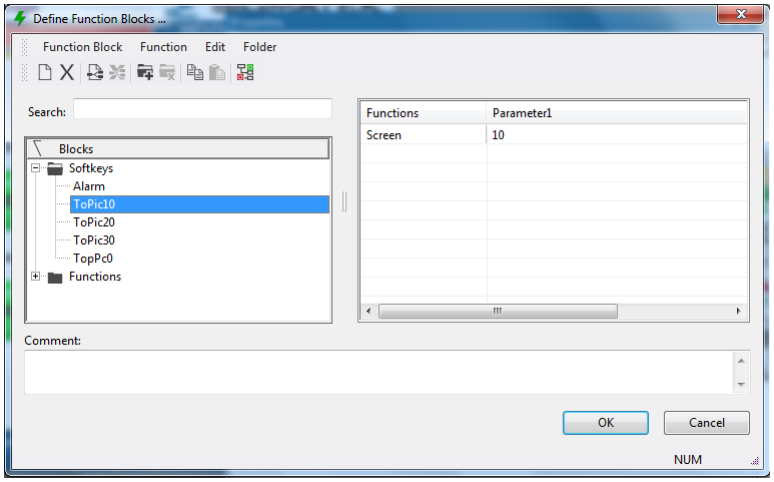


Figure 8.103: "ToPic10" Function Block

3. "ToPic20" Features

   "ToPic20" function block activates the screen 3.

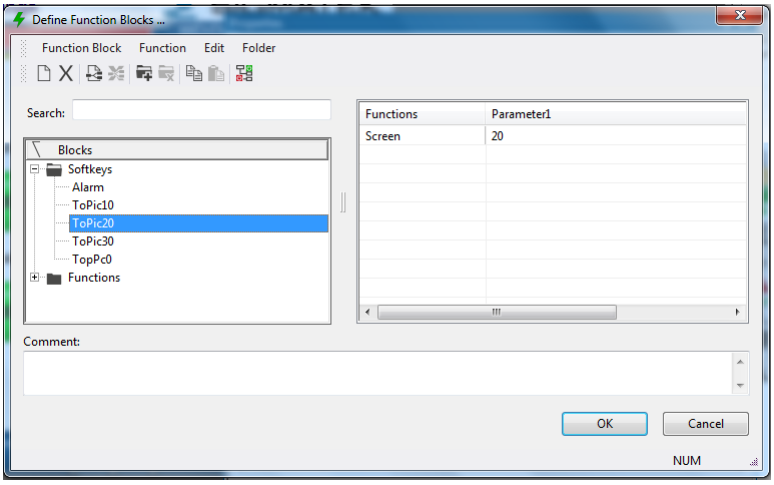


Figure 8.104: "ToPic20" Function Block

4. "ToPic30" Features

   "ToPic30" function block activates the screen 4.

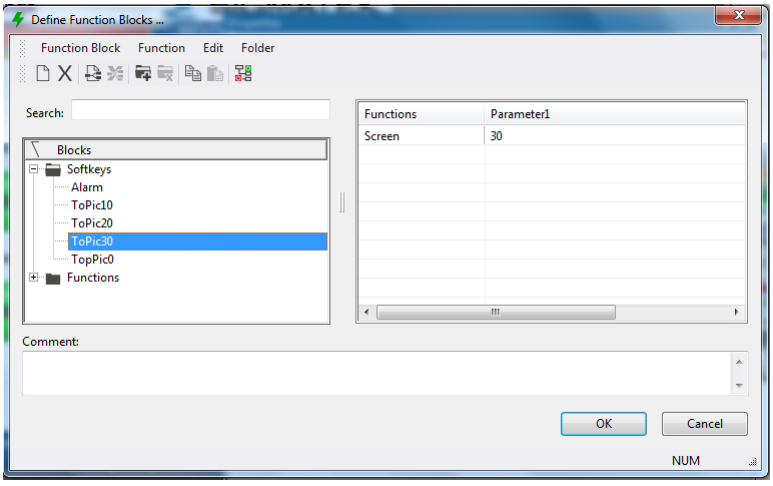


Figure 8.105: "ToPic30" Function Block

5. "AlaDel" Features

   "AlaDel" is part of the staff and has not been used in this project.

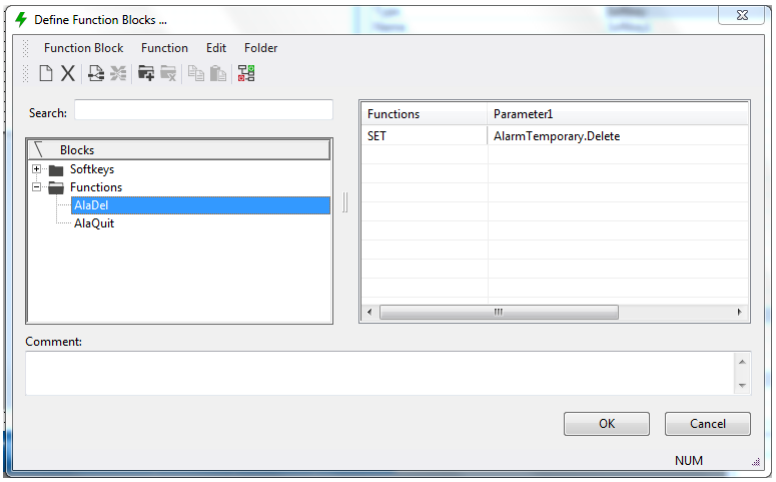


Figure 8.106: "AlaDel" Function Block

6. "AlaQuit" Features

   "AlaQuit" is part of the staff and has not been used in this project.

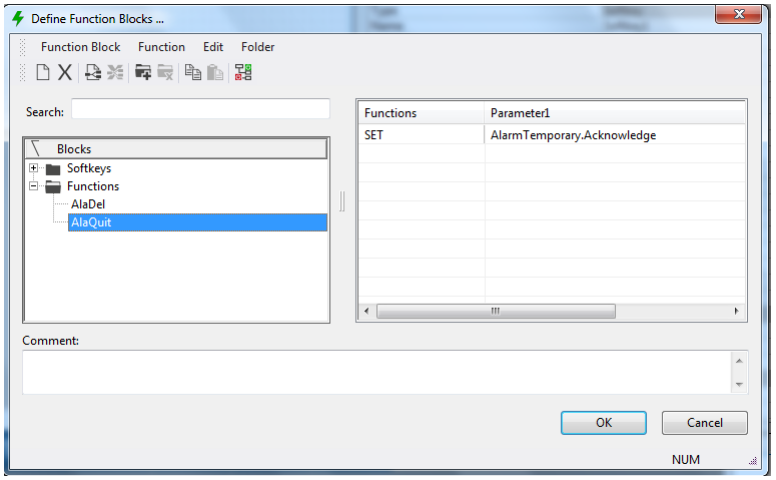


Figure 8.107: "AlaQuit" Function Block

8.8 Schemes

8.8.1 State Schemes

StateMoveAuto scheme is used to block buttons on screens 1 (except the Power On button), screen 2 and screen 4 when button 20 is active. In Figure 8.108 State-

MoveAuto scheme features are shown.

1. StateMoveAuto Features

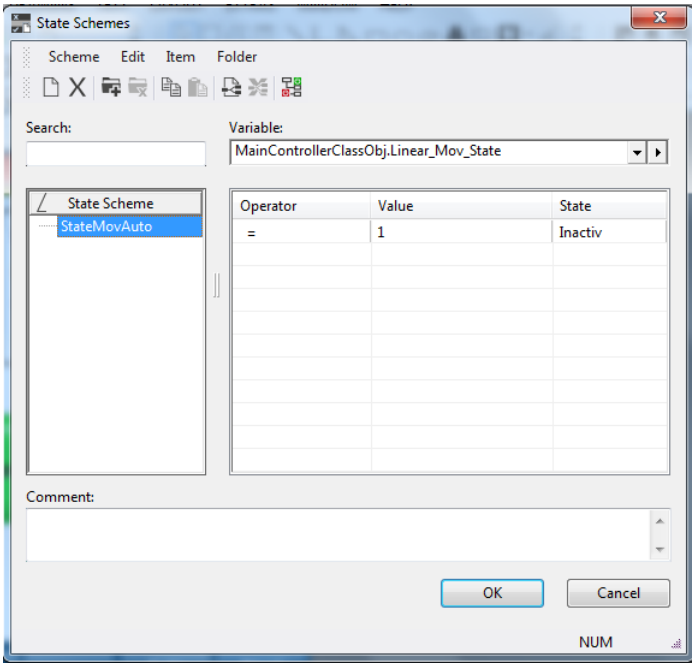


Figure 8.108: "StateMoveAuto" State Scheme Features

8.8.2 Color Schemes

1. Button 1 Color Scheme

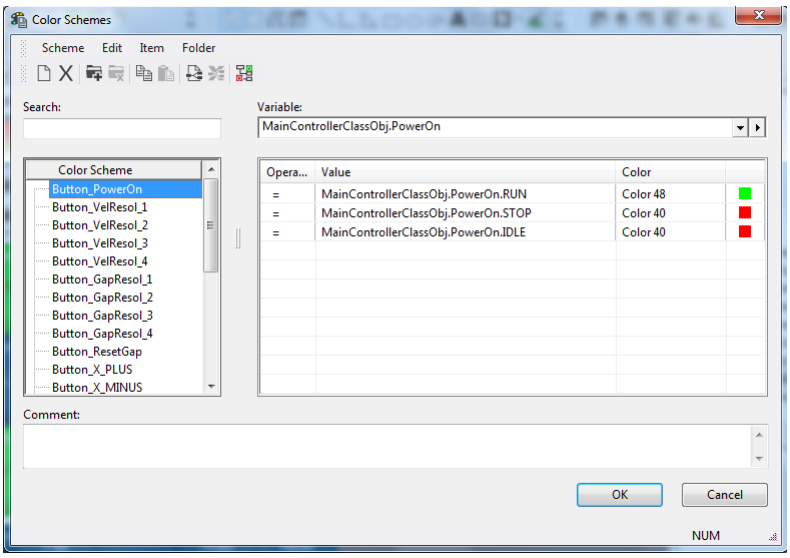


Figure 8.109: Button 1 Color Scheme

2. Button 2 Color Scheme

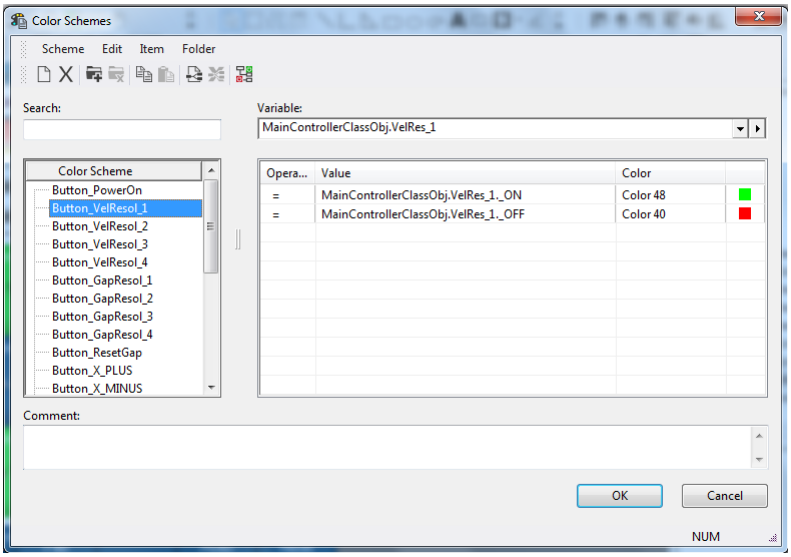


Figure 8.110: Button 2 Color Scheme

3. Button 3 Color Scheme

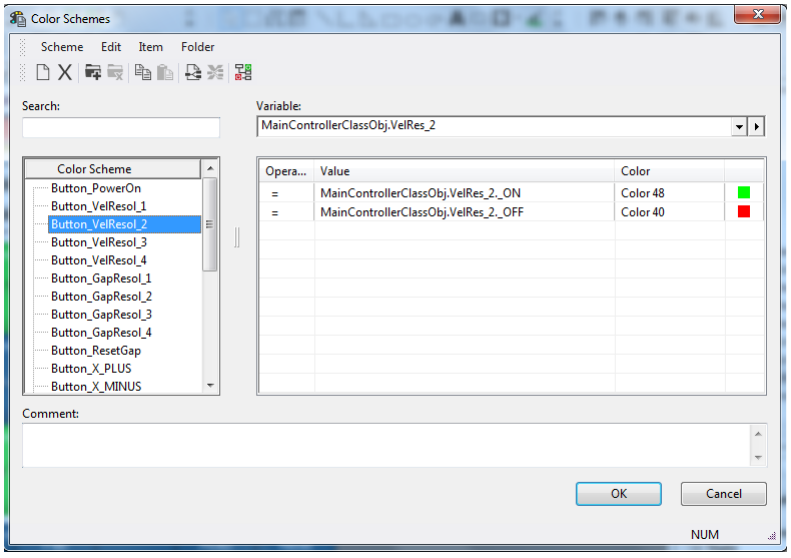


Figure 8.111: Button 3 Color Scheme

4. Button 4 Color Scheme

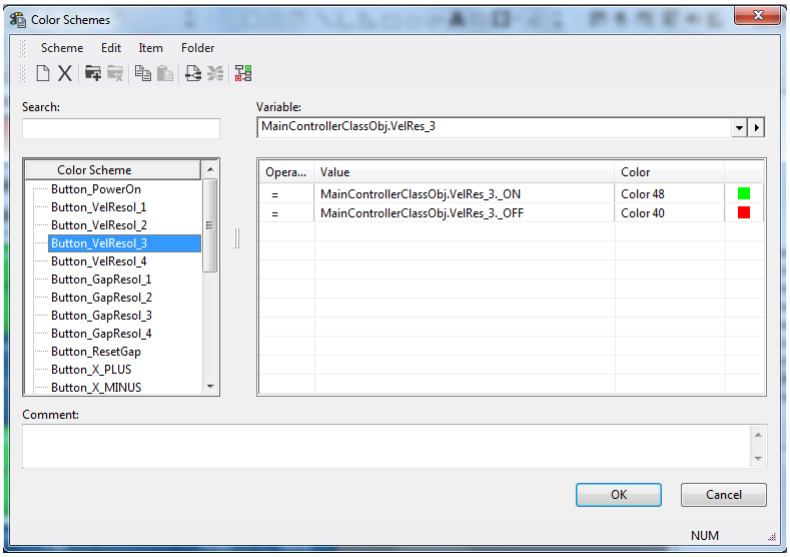


Figure 8.112: Button 4 Color Scheme

5. Button 5 Color Scheme

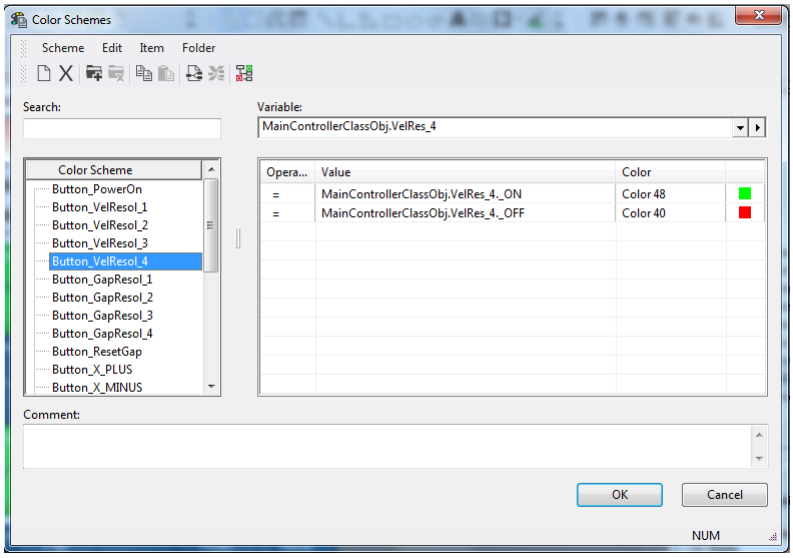


Figure 8.113: Button 5 Color Scheme

6. Button 6 Color Scheme

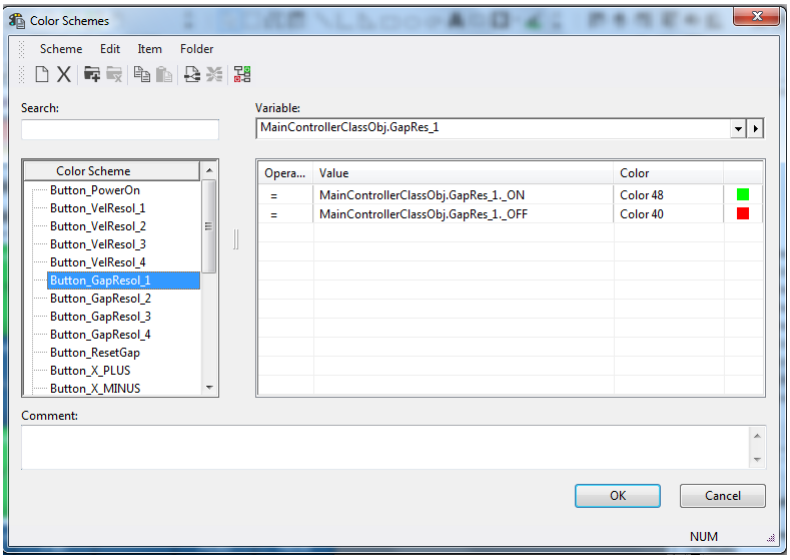


Figure 8.114: Button 6 Color Scheme

7. Button 7 Color Scheme

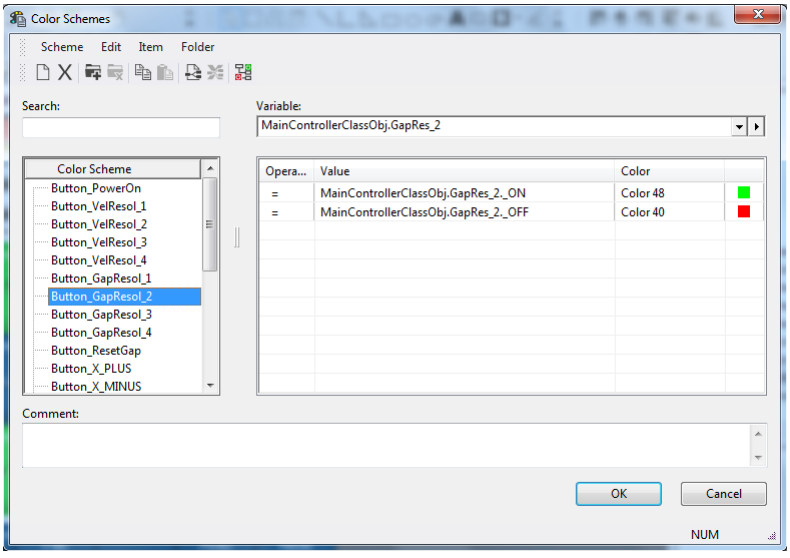


Figure 8.115: Button 7 Color Scheme

8. Button 8 Color Scheme

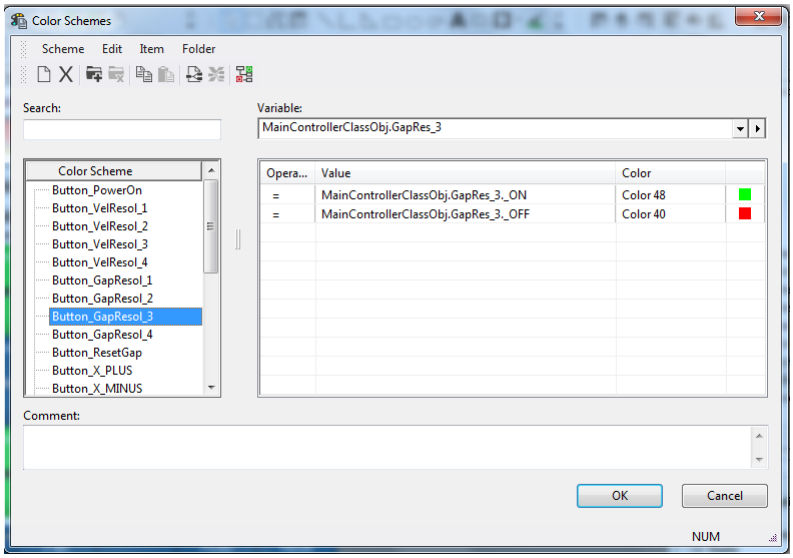


Figure 8.116: Button 8 Color Scheme

9. Button 9 Color Scheme

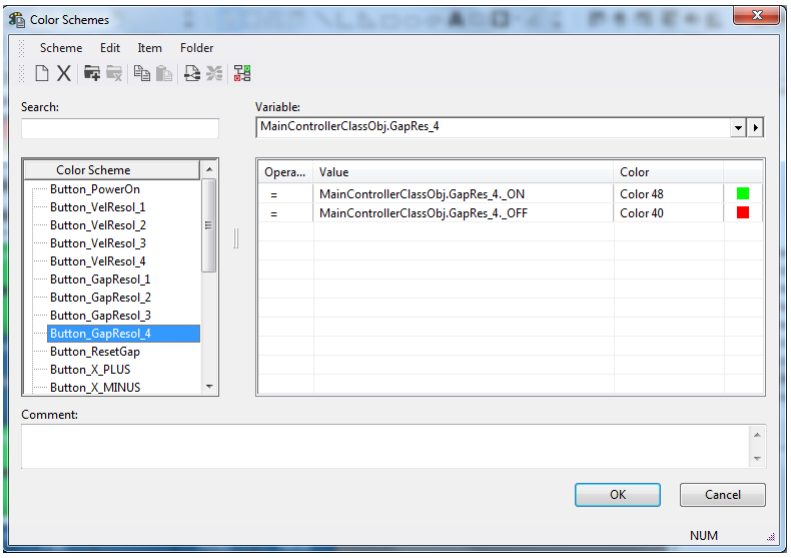


Figure 8.117: Button 9 Color Scheme

10. Button 10 Color Scheme

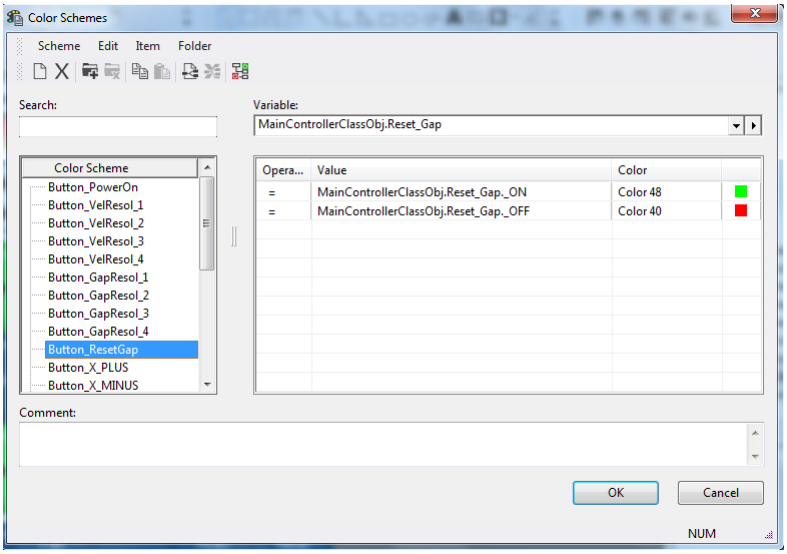


Figure 8.118: Button 10 Color Scheme

11. Button 11 Color Scheme

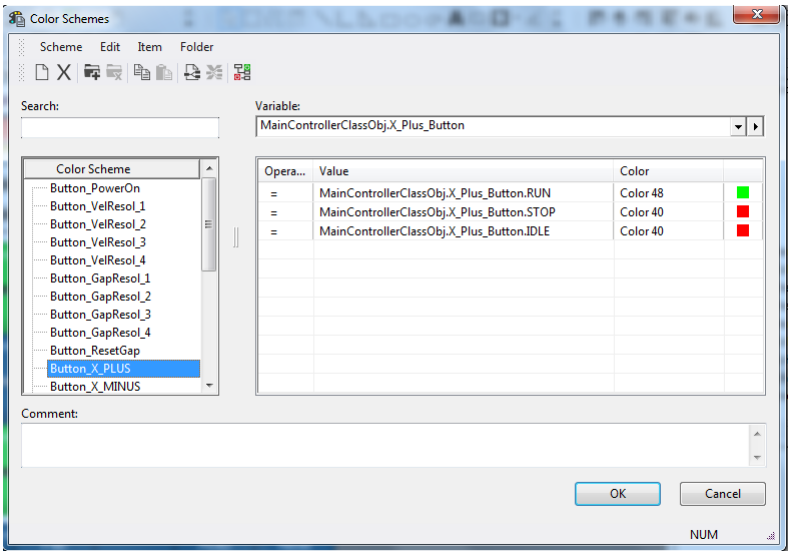


Figure 8.119: Button 11 Color Scheme

12. Button 12 Color Scheme

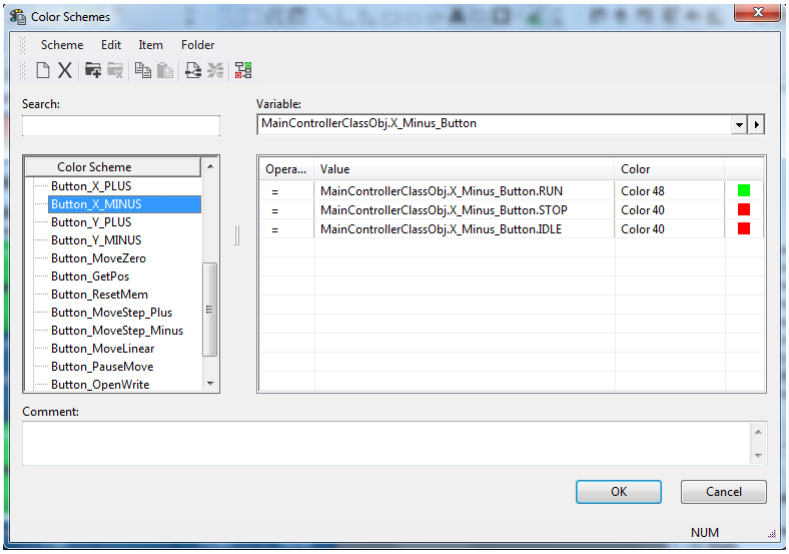


Figure 8.120: Button 12 Color Scheme

13. Button 13 Color Scheme

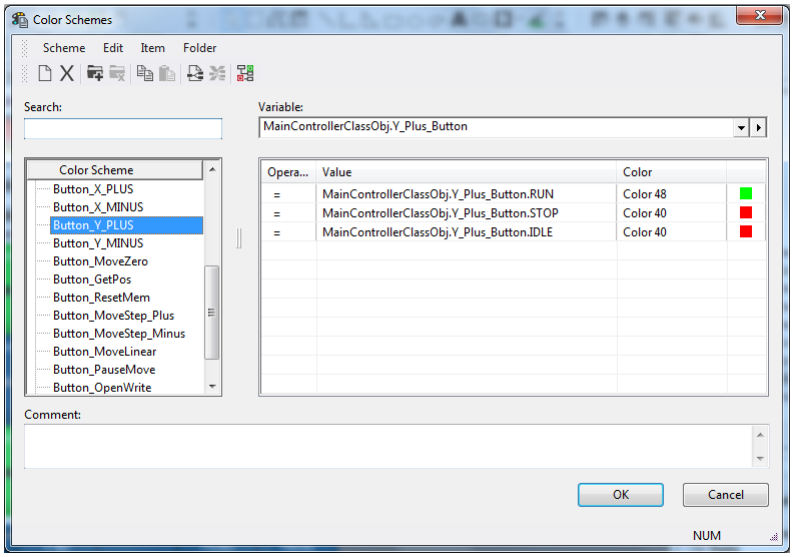


Figure 8.121: Button 13 Color Scheme

14. Button 14 Color Scheme

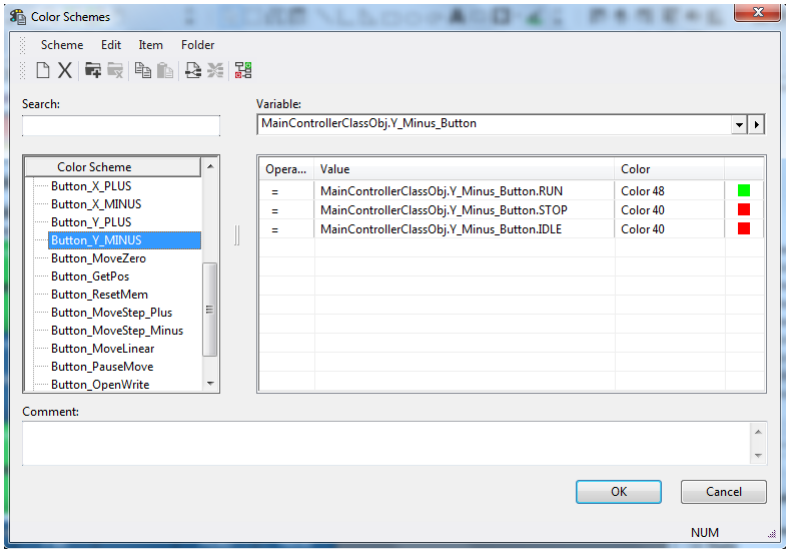


Figure 8.122: Button 14 Color Scheme

15. Button 15 Color Scheme

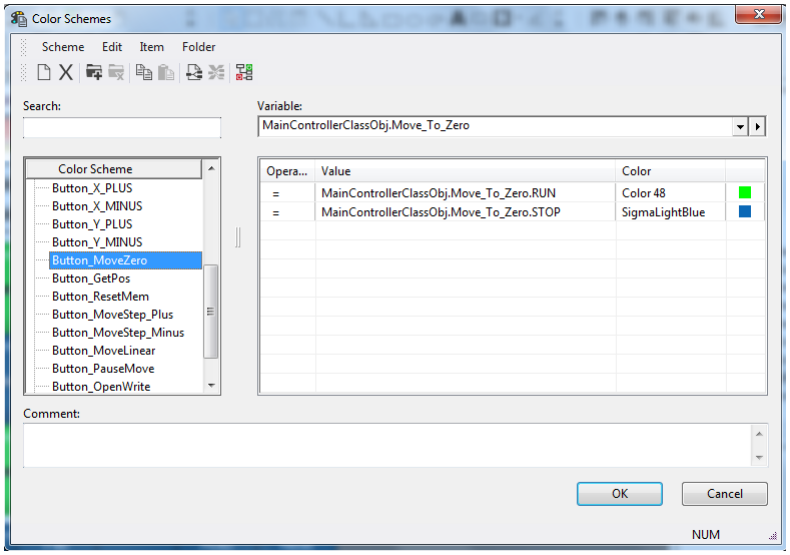


Figure 8.123: Button 15 Color Scheme

16. Button 16 Color Scheme

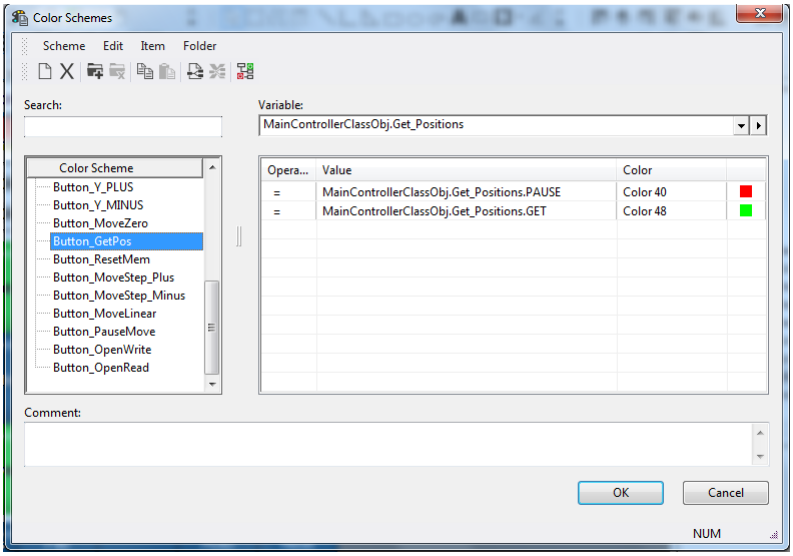


Figure 8.124: Button 16 Color Scheme

17. Button 17 Color Scheme

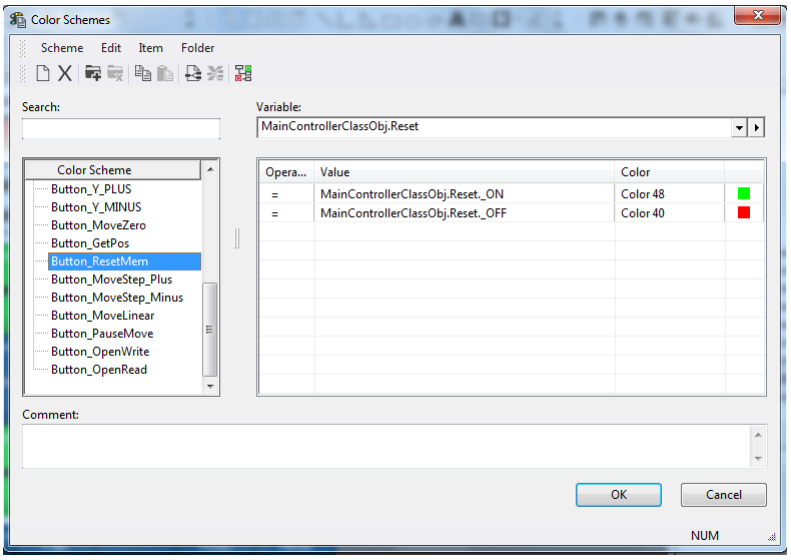


Figure 8.125: Button 17 Color Scheme

18. Button 18 Color Scheme

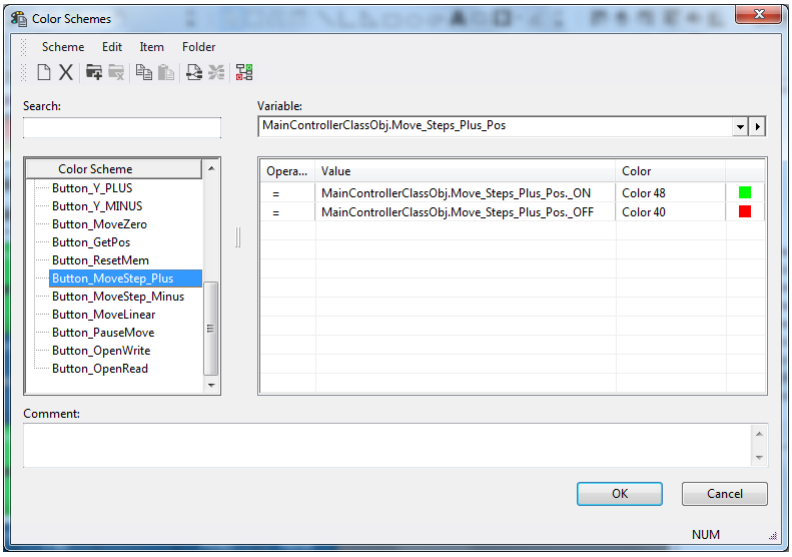


Figure 8.126: Button 18 Color Scheme

19. Button 19 Color Scheme

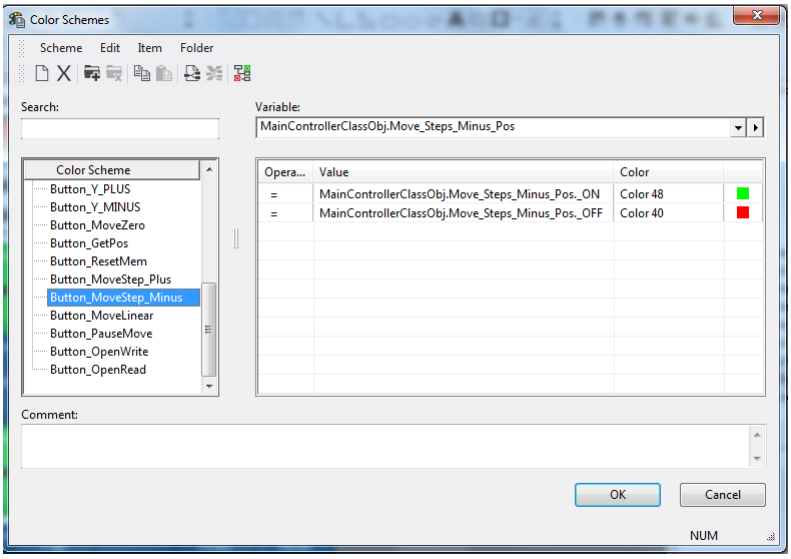


Figure 8.127: Button 19 Color Scheme

20. Button 20 Color Scheme

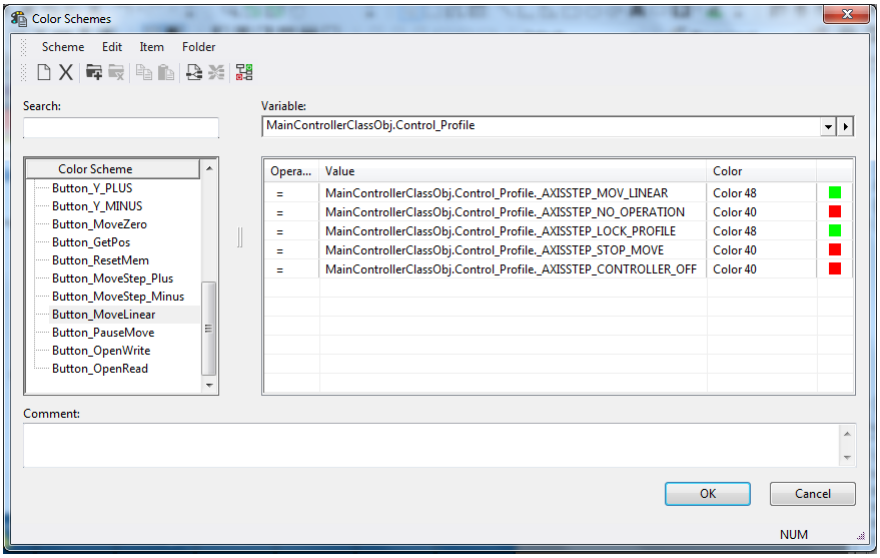


Figure 8.128: Button 20 Color Scheme

21. Button 21 Color Scheme

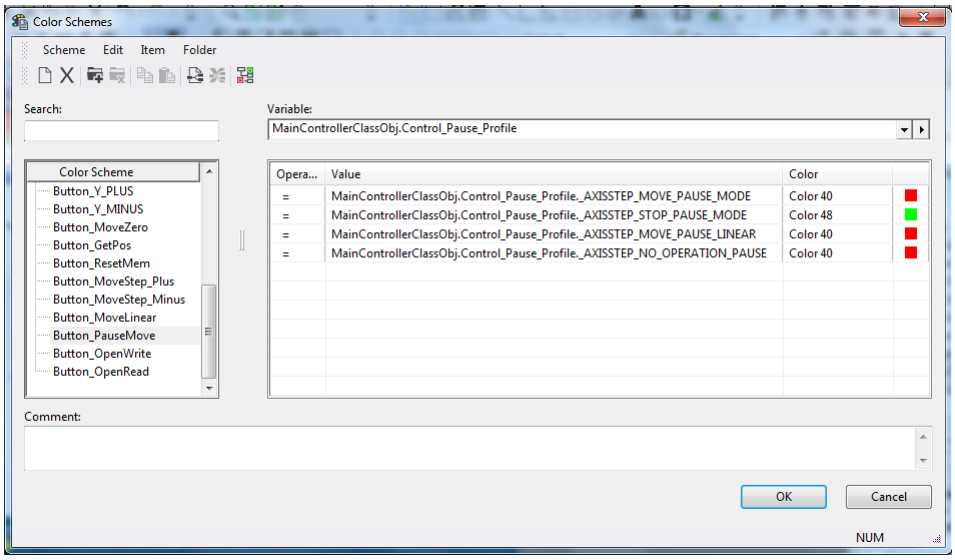


Figure 8.129: Button 21 Color Scheme

22. Button 22 Color Scheme

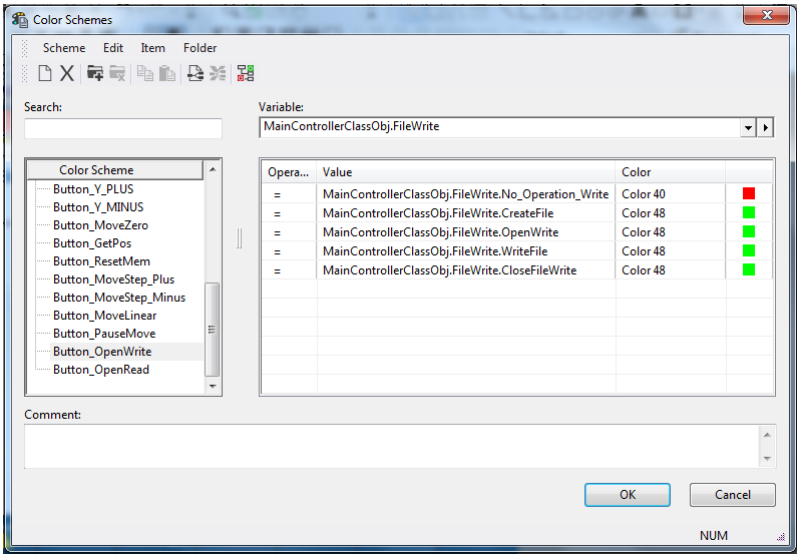


Figure 8.130: Button 22 Color Scheme

23. Button 23 Color Scheme

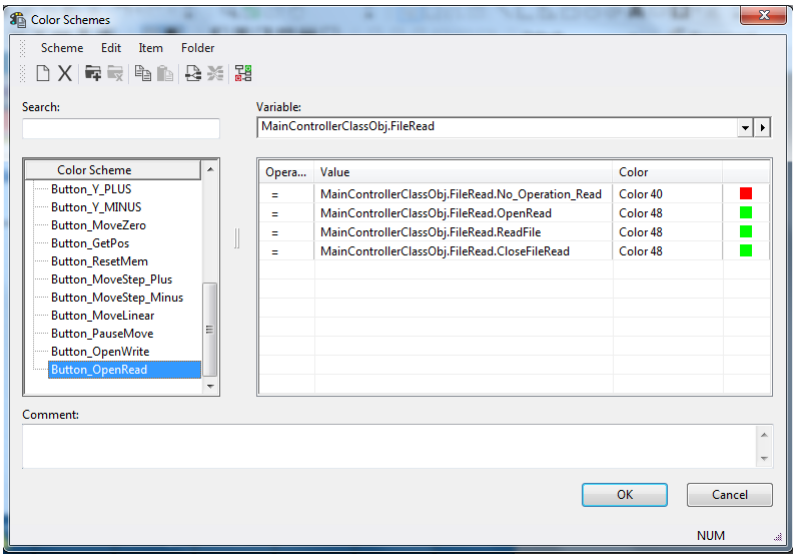


Figure 8.131: Button 23 Color Scheme

Bibliography

[1] YASKAWA ELECTRIC CORPORATION. *series SGM-SGMP-SGDA User's Manual*. Version TSE-S800-15C. 1995. DOI: https://www.yaskawa.com/pycprd/download/home#tab1.

[2] SIGMATEK GmbH & Co KG. *CDI 161 Product Information*. Version Unknown. 2014. DOI: http://www.sigmatek-automation.com/fileadmin/user_upload/downloads/CDI-161-eng.pdf.

[3] SIGMATEK GmbH & Co KG. *CIV 512 Product Information*. Version Unknown. 2015. DOI: http://www.sigmatek-automation.com/en/search-results/?q=CIV+512.

[4] SIGMATEK GmbH & Co KG. *CST 022 Product Information*. Version Unknown. 2014. DOI: http://www.sigmatek-automation.com/fileadmin/user_upload/downloads/CST-022-eng.pdf.

[5] SIGMATEK GmbH & Co KG. *CTO 163 Product Information*. Version Unknown. 2014. DOI: http://www.sigmatek-automation.com/fileadmin/user_upload/downloads/CTO-163-eng.pdf.

[6] SIGMATEK GmbH & Co KG. *ETV 0551 Product Information*. Version Unknown. 2016. DOI: http://www.sigmatek-automation.com/en/search-results/?q=ETV-0551.

[7] OMRON. *OMNUC U series User's Manual*. Version 1.01. 1996. DOI: http://www.fa.omron.com.cn/upload/doc/data/20080415154531863.pdf.

Chapter 9

Schematics