



GUIA RÁPIDO PARA COMUNICAÇÃO MODBUS RS485 COM A REMOTA MA-8X8YT XINJE



Sumário

1. INTRODUÇÃO	3
2. CONFIGURAÇÃO DO HARDWARE	4
2.1 Ligação Física entre CLP e Expansão Remota (RS485).....	4
2.2 Parâmetros de Comunicação – Configuração do Protocolo.....	5
3. CONFIGURAÇÃO DO SOFTWARE	9
3.1 Endereço Modbus das ENTRADAS DIGITAIS.....	9
3.2 Endereço Modbus das SAÍDAS DIGITAIS.....	10
3.3 Blocos Modbus para o CLP Xinje.....	10
4. LEITURA DAS ENTRADAS DA EXPANSÃO	12
4.1 BLOCO COLR – Lendo as Entradas do Módulo.....	12
5. ESCRITA DAS SAÍDAS DA EXPANSÃO	14
5.1 Configuração do Bloco COLW – Single Coil Write.....	14
5.2 Configuração do Bloco MCLW – Multi-Coil Read	15
6. CONSIDERAÇÕES FINAIS	16

1. INTRODUÇÃO

Nesse guia rápido, vamos criar um passo a passo de como configurar e acionar as entradas/saídas da expansão Remota MA-8X8YT utilizando um CLP XD5-32T4-C, ambos da Xinje.

Sabemos que as expansões remotas são diferentes dos módulos que geralmente acoplamos ao CLP. Essas expansões são acionadas via PROTOCOLO, e nesse caso, utilizamos o protocolo Modbus RS485 para fazer a comunicação entre CLP e expansão.

Temos duas grandes vantagens de utilizar expansões remotas para algumas aplicações. A primeira delas, que na maioria dos casos, é quando utilizamos IHM com CLP incorporado, onde o número de expansões é limitado, ou até mesmo em alguns modelos não são permitidos. Porém, se essa IHM com CLP conter o protocolo Modbus RS485, você pode acionar essas I/O's via protocolo, viabilizando assim uma "expansão" para a IHM com CLP.

Outra vantagem seria que a expansão física não precisa ser acoplada ao CLP para ter seu funcionamento. Em aplicações onde o painel de controle já está montado e não tenha espaço, ou até mesmo em uma estação grande de processo, onde não é viável a expansão ao lado do CLP, nós conseguimos apenas com um cabo blindado "ligar" o CLP à expansão, respeitando sempre a distância máxima permitida pelo protocolo que são de 1200 metros.

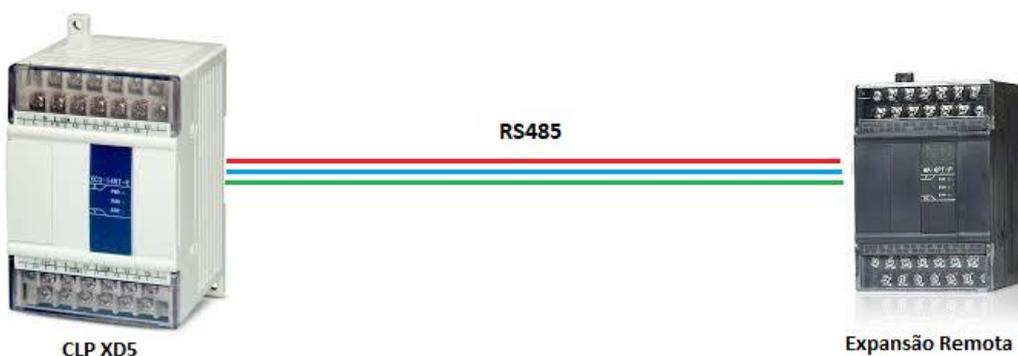
Essa expansão remota permite comunicação com o protocolo puro Modbus RS485, isso quer dizer que outros fabricantes de CLP's que possuam o mesmo protocolo podem se comunicar com ela, uma outra vantagem em utilizar essa formatação.

O empecilho, na maioria das vezes, é a velocidade de comunicação. Justamente por não ser comandada diretamente pela CPU do CLP, uma das desvantagens (e muito importante frisar isso) é a questão do tempo de resposta ao acionamento. Caso você tenha algum sensor rápido, que precise de um processamento e de uma resposta rápida, tanto de leitura quanto de escrita, essa topologia não é adequada para você. Em alguns processos críticos, o tempo de acionamento para o tempo de resposta no CLP pode ter um gap de até 1 segundo ou mais, dependendo dos periféricos que você está comunicando e da configuração da sua rede Modbus. Por isso é importante a conversa com algum Técnico da Kalatec antes de especificar e usar esse módulo.

2. CONFIGURAÇÃO DO HARDWARE

2.1 Ligação Física entre CLP e Expansão Remota (RS485)

Abaixo segue uma topologia de rede para a configuração dessa remota:

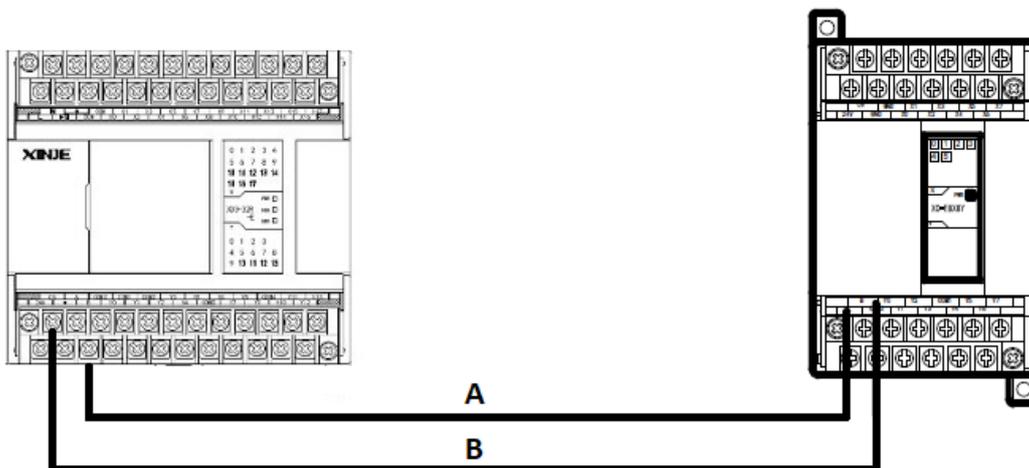


O protocolo Modbus tem algumas características e variáveis de rede que precisam ser configuradas da mesma maneira em AMBOS os periféricos (CLP e Expansão). AS configurações de rede são:

- Baud Rate
- Data Bit
- Stop Bit
- Parity (Paridade)
- Station (Nó)

Essas 5 configurações são imprescindíveis em uma rede Modbus. Alguns CLP's possuem sua configuração de fábrica, o que não quer dizer que todos os modelos terão essa mesma configuração. Por isso que devemos configurá-lo antes de começarmos a programação.

Outro fator importante é a nomenclatura de ligação física dos periféricos. Alguns fabricantes colocam os pinos do Modbus RS485 como Positivo (+) e Negativo (-), outros (no caso da Xinje) identificam como A (+) e B (-). Não existe uma forma correta, ambas as formas estão certas, é somente a maneira de identificá-los. Portanto, a ligação física fica da seguinte maneira:



Com isso, percebemos que devemos conectar o borne A do CLP no borne A da expansão, e mesma coisa com o B. Como dito antes, outros fabricantes podem ser que utilizem o + e - para identificar, isso não tem problema, só devemos nos atentar que: A (+) e B (-), e realizar a ligação física da mesma maneira.

2.2 Parâmetros de Comunicação – Configuração do Protocolo

Voltando a configuração do protocolo, retirei do manual a configuração padrão dos CLP's da Xinje:

Communication Parameters

Station	Modbus station number: 1~254
Baud Rate	300bps~9Mbps
Data Bit	5, 6, 7, 8, 9
Stop Bit	1, 1.5, 2
Parity	Even, Odd, even, empty, mask

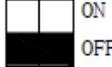
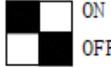
The default parameters: Station number is 1, baud rate is 19200bps, 8 data bits, 1 stop bit, even parity.

Partindo dessa configuração padrão, temos agora que configurar o módulo remoto da mesma maneira, usando EXATAMENTE a mesma configuração do CLP acima.

Essa configuração pode ser alterada conforme necessidade do usuário, porém não indicamos, pois, a programação se torna um pouco mais complexa quando fugimos do padrão. Isso não quer dizer que não vá funcionar, é apenas para fins de facilidade na programação.

Na expansão temos Switches configuráveis na parte frontal, onde justamente temos vamos configurar o protocolo. Segue abaixo a descrição sobre o que cada um faz.

Os primeiros dois switches são para configurar o Baud Rate.

DIP switch	1 2	1 2	1 2	1 2
				
Baud rate	19200 bps	4800bps	9600bps	57600bps

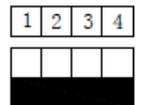
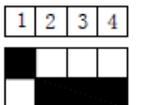
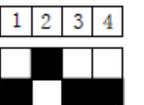
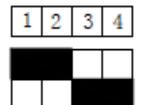
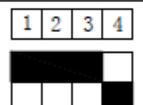
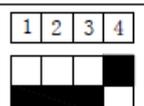
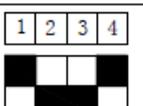
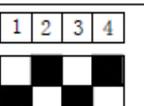
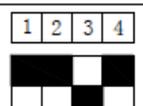
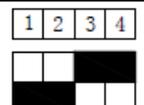
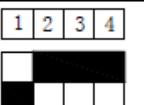
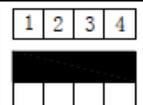
Os quatro últimos switches são para configurar o NÓ da expansão. O nó significa o endereço em que a expansão será identificada na rede, é o único parâmetro da configuração que precisa ser diferente em todos os periféricos. Por exemplo, se eu tenho 1 CLP comunicando com 4 inversores, eu tenho que dar um “nome” para cada componente, esse nome nós chamamos de NÓ. Segue exemplo abaixo:

MODBUS RS485 (Nó)

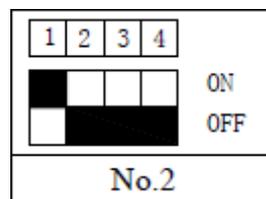


Nessa configuração exemplo, podemos ver que cada periférico possui um nó diferente na rede.

No módulo MA-8X8YT, nós configuramos o nó através dos 4 switches presentes na parte frontal. Segue descritivo:

DIP switch				
Station number	No.1	No.2	No.3	No.4
DIP switch				
Station number	No.5	No.6	No.7	No.8
DIP switch				
Station number	No.9	No.10	No.11	No.12
DIP switch				
Station number	No.13	No.14	No.15	No.16

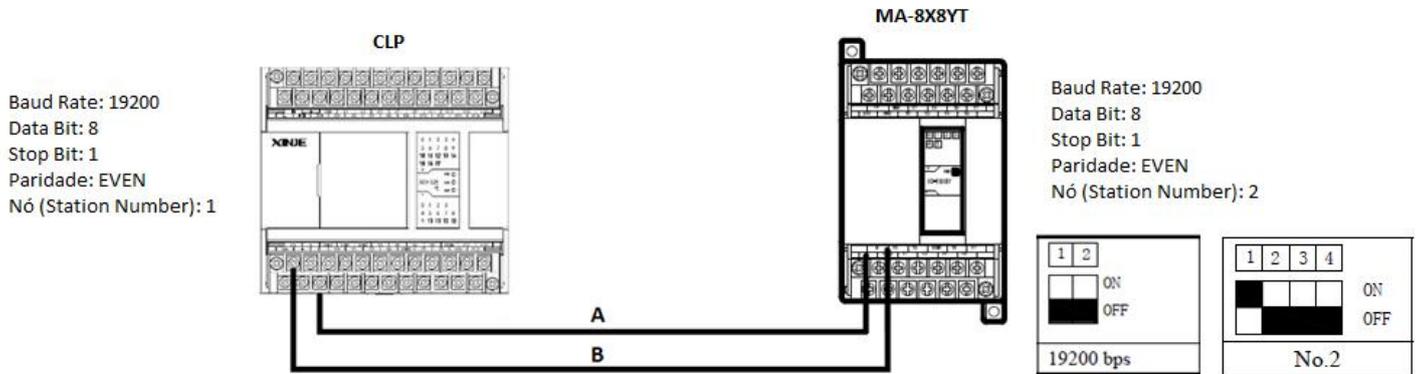
Como o CLP vem de fábrica com nó = 1, vamos configurar a expansão para nó = 2. Ou seja, vamos deixar em “ON” apenas o switch 1, conforme mostrado abaixo:



Com isso, temos configurado no nosso módulo a mesma configuração de fábrica do CLP, que seria:

- Baud Rate: **19200**
- Data Bit: **8**
- Stop Bit: **1**
- Parity (Paridade): **EVEN**
- Station (Nó): **2 (Único parâmetro que deve ser diferente)**

Temos a seguinte topologia de rede:



3. CONFIGURAÇÃO DO SOFTWARE

Feita toda ligação e configuração física dos periféricos, vamos agora acionar as entradas e saídas pelo software XDPPro.

Temos que lembrar que estamos trabalhando com o protocolo Modbus para acionamentos das entradas e saídas, com isso, não podemos identificar essas I/O's no software como normalmente fazemos X0, X1, Y0, Y1. Esses endereços são apenas da própria CPU do CLP, os endereços em Modbus são identificados de outra maneira. Para isso, o manual nos fornece uma tabela com o nome da variável e seu respectivo endereço Modbus para utilizarmos na programação. Segue a tabela de entradas e saídas com seus respectivos endereços Modbus:

3.1 Endereço Modbus das ENTRADAS DIGITAIS

Modbus address (decimal)	Description	Remark
0x0	X0 Input point	8X8YR , 16X
0x1	X1 Input point	8X8YR , 16X
0x2	X2 Input point	8X8YR , 16X
0x3	X3 Input point	8X8YR , 16X
0x4	X4 Input point	8X8YR , 16X
0x5	X5 Input point	8X8YR , 16X
0x6	X6 Input point	8X8YR , 16X
0x7	X7 Input point	8X8YR , 16X
0x8	X8 Input point	16X
0x9	X9 Input point	16X
0x10	X10 Input point	16X
0x11	X11 Input point	16X
0x12	X12 Input point	16X
0x13	X13 Input point	16X
0x14	X14 Input point	16X
0x15	X15 Input point	16X

3.2 Endereço Modbus das SAÍDAS DIGITAIS

Modbus address (decimal)	Description	Remark
0x128	Y0 Output point	8X8YR , 16YR/T
0x129	Y1 Output point	8X8YR , 16YR/T
0x130	Y2 Output point	8X8YR , 16YR/T
0x131	Y3 Output point	8X8YR , 16YR/T
0x132	Y4 Output point	8X8YR , 16YR/T
0x133	Y5 Output point	8X8YR , 16YR/T
0x134	Y6 Output point	8X8YR , 16YR/T
0x135	Y7 Output point	8X8YR , 16YR/T
0x136	Y8 Output point	16YR/T
0x137	Y9 Output point	16YR/T
0x138	Y10 Output point	16YR/T
0x139	Y11 Output point	16YR/T
0x140	Y12 Output point	16YR/T
0x141	Y13 Output point	16YR/T
0x142	Y14 Output point	16YR/T
0x143	Y15 Output point	16YR/T

Nessas duas tabelas retiradas do manual do módulo temos a informação dos endereços. Podemos pegar como exemplo o Y0, onde ele possui o endereço 128 em decimal de acordo com a primeira coluna da tabela. O Y1 possui o endereço 129 em decimal e assim por diante. Ou seja, agora que temos todos os endereços mapeados, podemos configurar o bloco de leitura e escrita Modbus no software XDPPro.

3.3 Blocos Modbus para o CLP Xinje

Outro detalhe que temos que nos atentar é em relação a qual bloco Modbus vamos utilizar na programação do CLP. Abaixo deixo uma lista com alguns blocos disponíveis:

Mnemonic	Function	Circuit and soft components
MODBUS Communication		
COLR	Coil Read	— — COLR S1 S2 S3 D1 D2
INPR	Input coil read	— — INPR S1 S2 S3 D1 D2
COLW	Single coil write	— — COLW D1 D2 S1 S2
MCLW	Multi-coil write	— — MCLW D1 D2 D3 S1 S2
REGR	Register read	— — REGR S1 S2 S3 D1 D2
INRR	Input register read	— — INRR S1 S2 S3 D1 D2
REGW	Single register write	— — REGW D1 D2 S1 S2
MRGW	Multi-register write	— — MRGW D1 D2 D3 S1 S2

Cada bloco desse tem sua particularidade e sua função. Para fins de demonstração e exemplo, vamos utilizar apenas 2 blocos, um para escrita Modbus (para ativar as saídas da remota) e outro para leitura Modbus (para ler as entradas da remota). Os blocos que utilizaremos são: **COLW** (Single Coil Write) e **COLR** (Coil Read).

Coil – São memórias e acionamentos booleanos, exemplo: M, X, Y.

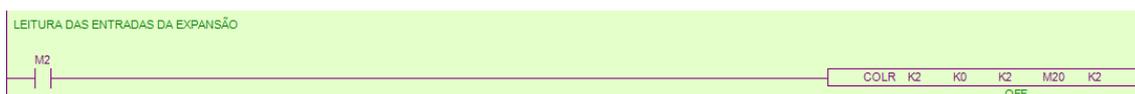
Register – São registradores que armazenam algum valor, exemplo: D, C, T.

4. LEITURA DAS ENTRADAS DA EXPANSÃO

4.1 BLOCO COLR – Lendo as Entradas do Módulo

O bloco COLR (Coil Read) é usado para ler um determinado range de coils (entradas) na expansão remota. Abaixo segue a configuração do bloco:

OPERADOR	FUNÇÃO	EXEMPLO
S1	Endereço (Nó) da Expansão Remota.	K2
S2	Endereço da Primeira Entrada da Expansão Remota.	K0
S3	Quantidade de Endereços para serem lidos.	K8
D1	Endereço Inicial de Memória para serem lidas as Entradas.	M20
D2	Endereço da Porta Serial que está sendo Usada.	K2



Ou seja, essa aplicação representa a leitura das entradas remotas via Modbus, onde temos o primeiro endereço das entradas (S2 do bloco) que seria o valor 0 (zero) em decimal:

Modbus address (decimal)	Description	Remark
0x0	X0 Input point	8X8YR , 16X
0x1	X1 Input point	8X8YR , 16X

E com o valor da quantidade de endereços a serem lidos (S3 do bloco), temos as 8 entradas lidas em um só bloco, com o endereço de leitura visível a partir do M20. Logo, temos a seguinte tabela de acionamentos:

- Quando a entrada X0 for acionada, a leitura será dada na memória M20.
- Quando a entrada X1 for acionada, a leitura será dada na memória M21.
- Quando a entrada X2 for acionada, a leitura será dada na memória M22.

E assim por diante, oito vezes, que foi o número que colocamos no S3 do bloco e contadas a partir do M20 que colocamos no D1 do bloco.

A variável D2 que é referente à porta de comunicação, sempre utilizaremos o valor 2, pois representa a porta Modbus RS485 dos CLP's da Xinje. Segue abaixo uma tabela com o exemplo das portas de comunicação:

PORTA DE COMUNICAÇÃO	Valor	Descrição
Porta 0	K0	RS232
Porta 1	K1	RS232
Porta 2	K2	RS485
Porta 3	K3	Porta de Extensão do Lado Esquerdo
Porta 4	K4	Acima da Porta de Extensão 1
Porta 5	K5	Acima da Porta de Extensão 2

Portanto, sempre utilizamos o K2 (RS485) nos blocos Modbus tanto de leitura, quanto de escrita.

5. ESCRITA DAS SAÍDAS DA EXPANSÃO.

Para o acionamento das saídas do módulo, temos dois blocos que podemos utilizar, o bloco COLW (Single Coil Write) e o bloco MCLW (Multi-Coil Write). A principal diferença é:

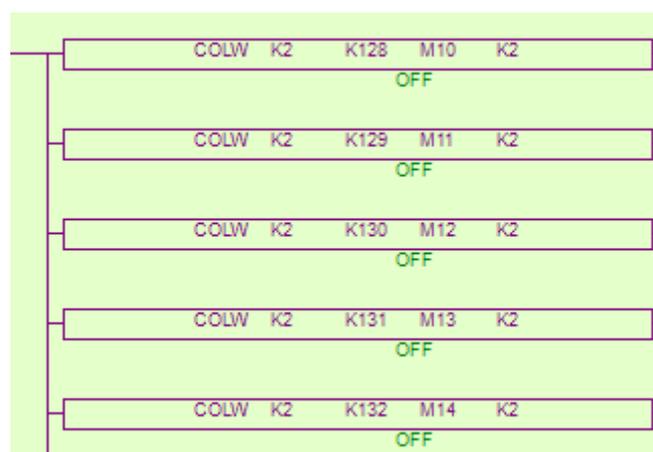
- COLW (Single Coil Write) – O bloco faz a escrita de apenas 1 saída por vez;
- MCLW (Multi-Coil Write) – O bloco faz a escrita de mais de uma 1 saída por vez;

Ou seja, caso eu queira acionar apenas uma saída, eu posso utilizar o bloco COLW que faz essa função. Mas se eu quiser utilizar as 08 saídas presentes no módulo eu utilizo o bloco MCLW.

5.1 Configuração do Bloco COLW – Single Coil Write

OPERADOR	FUNÇÃO	EXEMPLO
S1	Endereço (Nó) da Expansão Remota.	K2
S2	Endereço da Saída da Expansão Remota.	K128
D1	Endereço de Memória para serem Escritas as Saídas.	M10
D2	Endereço da Porta Serial que está sendo Usada.	K2

Exemplo de Programação com o bloco COLW:



Com isso, temos o seguinte cenário:

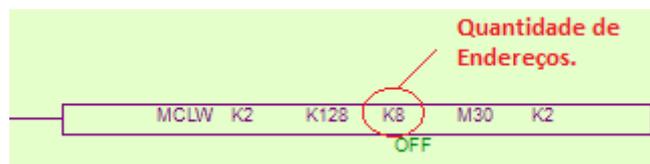
- Quando a memória M10 for acionada, a saída acionada será a Y0.
- Quando a memória M11 for acionada, a saída acionada será a Y1.
- Quando a memória M12 for acionada, a saída acionada será a Y2.

Portanto, enxergamos que a configuração do bloco COLW fica mais “manual” quando utilizamos mais de uma saída, onde tenho que implementar o bloco diversas vezes. Para facilitar o nosso trabalho, segue abaixo a configuração do bloco MCLW que permite a configuração apenas uma vez.

5.2 Configuração do Bloco MCLW – Multi-Coil Read

OPERADOR	FUNÇÃO	EXEMPLO
S1	Endereço (Nó) da Expansão Remota.	K2
S2	Endereço da Primeira Saída da Expansão Remota.	K128
S3	Quantidade de Endereços para serem Escritos.	K8
D1	Endereço Inicial de Memória para serem Escritas as Saídas.	M30
D2	Endereço da Porta Serial que está sendo Usada.	K2

Exemplo de Programação com o bloco MCLW:



Podemos ver no exemplo que o bloco MCLW, colocamos apenas uma vez no programa e identificamos a quantidade de endereços que eu quero ler, no caso do exemplo, oito endereços. Portanto, ficamos com o seguinte cenário, que é exatamente igual ao cenário anterior com o bloco COLW, com a diferença de ser colocado apenas uma vez no programa:

- Quando a memória M30 for acionada, a saída acionada será a Y0.
- Quando a memória M31 for acionada, a saída acionada será a Y1.
- Quando a memória M32 for acionada, a saída acionada será a Y2.

Assim, finalizamos o conceito e o funcionamento do módulo de entradas e saídas digitais remotas. Abordamos as aplicações e sua configuração no hardware e no software.

6. CONSIDERAÇÕES FINAIS

Com a introdução sobre o conceito de módulos remotos, conseguimos entender um pouco mais sobre a aplicação e o funcionamento dessa expansão. Para ter um bom desenvolvimento é necessário um conhecimento básico no protocolo Modbus, porém vimos que não é difícil a comunicação.

Outra vantagem excelente dessa expansão MA-8X8YT é que conseguimos agrega-la com qualquer outro fabricante de CLP, seja Delta, Siemens, Rockwell, Schneider, entre outros, desde que o CLP possua o protocolo Modbus RS485. Com isso, temos algumas vantagens quando falamos em entradas e saídas. Conseguimos expandir nossa aplicação apenas implementando esse módulo.

Qualquer dúvida pertinente entre em contato com o time técnico da Kalatec para solucioná-las. Para terem acesso a programação usada nesse documento, vídeo-aulas e manuais, entre em contato conosco também que fornecemos essa documentação. Abaixo segue os contatos em relação a cada região:

Matriz Campinas – SP

Segunda à Quinta das 07h40 às 17h30
Sexta das 08h00 às 17h00
Telefone: (19) 3045-4900
Atende Brasil inteiro e Interior de São Paulo.

Filial São Paulo – SP

Segunda à Quinta das 07h40 às 17h30
Sexta das 08h00 às 17h00
Telefone: (11) 5514-7680
Atende Grande São Paulo e São Paulo Capital.

Filial Joinville – SC

Segunda à Quinta das 07h40 às 17h30
Sexta das 08h00 às 17h00
Telefone: (47) 3425-0042
Atende a Região Sul do Brasil.

SITE KALATEC AUTOMAÇÃO

