

Driver BCM BCM1_MasterSlave

Nome do Arquivo	BCM1_MasterSlave.dll
Fabricante	BCM Automação
Equipamentos	CLPs BCM modelos GP3000, 1086A, 1088 e 2085B (CPU4-A, CPU6)
Protocolo	BCM1
Versão	3.0.3
Última Atualização	10/08/2026
Plataforma	Win32
Dependências	IOKit versão 2.0 ou superior
Leitura com Superblocos	Não
Nível	0

Introdução

Este Driver implementa o protocolo BCM1, que permite aos produtos da **Elipse Software** comunicar com os CLPs BCM modelos GP3000, 1086A, 1088 e 2085B (CPU4-A, CPU6).

Preparação do Equipamento

Para comunicação via RS485 no Windows NT, Windows 2000, Windows XP, Windows Vista ou Windows 7, utilize o conversor RS232/485 BCM515 Turbo. O conversor BCM515 deve ser usado somente no Windows 95, Windows 98 ou Windows ME.

Parâmetros de Configuração do Driver

Os parâmetros **[P]** de configuração deste Driver não são utilizados. O Driver BCM BCM1_MasterSlave foi desenvolvido em conjunto com a biblioteca **IOKit** da **Elipse Software**. Esta biblioteca, após ser configurada pelo usuário, é responsável por implementar o acesso ao meio físico desejado, Serial, Ethernet, Modem ou RAS.

Todas as configurações da biblioteca **IOKit**, e as específicas deste Driver, devem ser efetuadas na janela de propriedades deste Driver ou usando a opção **Extra Settings** no **Elipse SCADA**.

Nesta janela de propriedades, a aba **BCM1_MasterSlave** contém as configurações específicas deste Driver. As demais abas se referem às configurações de comunicação da biblioteca **IOKit** da **Elipse Software**.

Para mais informações sobre a configuração da biblioteca **IOKit**, consulte o tópico Documentação das Interfaces de Comunicação.

Configuração das Propriedades

Este tópico contém informações sobre as propriedades disponíveis na aba **BCM1_MasterSlave**, já incluindo o valor das **Strings** de propriedades *offline*, que podem ser programadas pelo usuário ao iniciar uma aplicação em modo **Offline**.

Para aplicações **E3**, **Power** ou **Water**, o valor destas configurações também pode ser definido em tempo de execução. Para isto, selecione a opção **Start driver OFFLINE** na aba **Setup** da janela de propriedades para iniciar este Driver em modo **Offline**.

As opções de configuração deste Driver estão listadas na tabela a seguir.

Opções de configuração do Driver BCM BCM1_MasterSlave

ABA	PARÂMETRO	STRING OFFLINE	TIPO DE DADOS	DESCRIÇÃO
Setup	Physical Layer	IO.Type	Texto	Seleciona o meio físico de comunicação deste Driver. Por padrão, utilize a opção Serial
	Timeout	IO.TimeoutMs	Número	Limite de tempo, em milissegundos, para recepção dos dados na resposta do equipamento. Por exemplo, o valor 1000 define um limite de 1 (um) segundo

ABA	PARÂMETRO	STRING OFFLINE	TIPO DE DADOS	DESCRIÇÃO
Serial	Port	IO.Serial.Port	Número	Por padrão, utilize o mesmo valor configurado no dispositivo. Os valores possíveis são 1 : COM1, 2 : COM2, 3 : COM3, ..., n : COMn
	Baud rate	IO.Serial.Baudrate	Número	Por padrão, utilize o mesmo valor configurado no dispositivo. Por exemplo, o valor 9600 define a taxa de <i>baud rate</i> da comunicação serial
	Data bits	IO.Serial.DataBits	Número	Por padrão, utilize o mesmo valor configurado no dispositivo. Os valores possíveis são 7 : 7 bits ou 8 : 8 bits
	Parity	IO.Serial.Parity	Número	Por padrão, utilize o mesmo valor configurado no dispositivo. Os valores possíveis são 0 : No Parity, 1 : Odd Parity, 2 : Even Parity, 3 : Mark Parity ou 4 : Space Parity
	Stop bits	IO.Serial.StopBits	Número	Por padrão, utilize o mesmo valor configurado no dispositivo. Os valores possíveis são 0 : Stop Bit 1, 1 : Stop Bit 1.5 ou 2 : Stop Bit 2
BCM1_MasterSlave	Max Events	BCM1_MasterSlave.MaxEvents	Número	Número de eventos na fila do CLP. Se este valor é configurado como 0 (zero), utiliza-se o valor 700 (padrão)

Todas as propriedades *offline* devem ser configuradas via Tags de Comunicação no formato **String**, usando os parâmetros *N1* igual a -1 (menos um), *N2* igual a 0 (zero), *N3* igual a 0 (zero) e *N4* igual a 3 (três). Para mais detalhes e exemplos, consulte o tópico Documentação das Interfaces de Comunicação.

Referência de Tags

Esta seção contém informações sobre a configuração dos Tags **[N/B]** deste Driver.

Este Driver também implementa recursos de leitura de estampa de tempo fornecidos pelo CLP de duas formas. Uma destas formas consiste no *download* de eventos do CLP através de um *buffer* circular de eventos e um conjunto de variáveis que deve ser mantido por este Driver. A outra forma é através da leitura de Tags Bloco contendo uma estampa de tempo que corresponde a uma estrutura montada na própria memória do CLP. Para mais informações sobre estas funcionalidades, consulte os tópicos Eventos do CLP e Tags Bloco com Estampa de Tempo Fornecida pelo CLP.

Parâmetros [N] para Tags de Comunicação

Leitura e Escrita

N1	Número do CLP, maior que 0 (zero)
N2	Tipo de variável. Para mais informações, consulte a tabela Tipos de Variáveis
N3	Endereço, que varia conforme o parâmetro <i>N2</i> . Para mais informações, consulte a tabela Tipos de Variáveis versus Endereçamento
N4	0 (zero, não utilizado)

Tipos de Variáveis

TIPO DE VARIÁVEL (N2)	SIGNIFICADO
0	Endereço absoluto
1	Display 1 (um)
2	Display 2 (dois)
3	Display 3 (três)
4	Display 4 (quatro)

Tipos de Variáveis versus Endereçamento

TIPO DE VARIÁVEL (N2)	ENDEREÇO (N3)
0	Endereço da variável
1, 2, 3 ou 4	Posição do Display

Parâmetros [B] para Blocos de Comunicação

Leitura e Escrita

B1	Número do CLP, maior que 0 (zero)
B2	Tipo de variável. Para mais informações, consulte a tabela Tipos de Variáveis
B3	Endereço, que varia conforme o parâmetro B2. Para mais informações, consulte a tabela Tipos de Variáveis versus Endereçamento
B4	Endereço da primeira palavra de controle do gerenciamento da pilha. Para mais informações, consulte a nota mais adiante neste tópico

Tipos de Variáveis

TIPO DE VARIÁVEL (B2)	SIGNIFICADO
0	Endereço absoluto de memória
1	Display 1 (um)
2	Display 2 (dois)
3	Display 3 (três)
4	Display 4 (quatro)
5	Endereço absoluto de memória com estampa de tempo vinda do CLP

Tipos de Variáveis versus Endereçamento

TIPO DE VARIÁVEL (B2)	ENDEREÇO (B3)
0	Endereço da variável
1, 2, 3 ou 4	Posição do Display
5	Endereço da variável

NOTA

Caso o parâmetro *B2* seja igual a 5 (cinco), ou seja, endereço absoluto de memória com estampa de tempo vinda do CLP, o parâmetro *B4* deve conter o endereço da primeira palavra (*Word*) de controle do gerenciamento da pilha. Nos demais casos, o parâmetro *B4* deve ser igual a 0 (zero). Além disto, no caso do parâmetro *B2* igual a 5 (cinco), este Tag Bloco é somente de leitura. Para mais informações, consulte o tópico Tags Bloco com Estampa de Tempo Fornecida pelo CLP.

Tags Bloco com Estampa de Tempo Fornecida pelo CLP

Para transferir automaticamente valores com estampa de tempo própria para uma aplicação, deve-se criar uma fila, em uma área de memória qualquer do CLP, com as seguintes características:

- Deve ser uma área contínua formada por células no formato a seguir, para um Tag Bloco de tamanho *N*:
- $\text{Mês} \times 100 + \text{Dia}$
- $\text{Hora} \times 100 + \text{Minuto}$
- $\text{Ano} \times 100 + \text{Segundo}$
- Valor do Elemento 0 (zero)
- Valor do Elemento 1 (um)
- ...
- Valor do Elemento *N*
- Soma de verificação (*checksum*)
- A soma de verificação (*checksum*) é calculada como a soma simples de todos os Elementos daquela célula.
- Duas palavras de controle devem ser colocadas em dois endereços consecutivos de memória. A primeira destas palavras é um status de leitura. Ao iniciar a leitura de uma pilha, uma aplicação escreve o valor 1 (um) nesta posição de memória. Ao terminar a leitura, escreve o valor 2 (dois), indicando que os dados foram lidos com sucesso, podendo ser descartados desta pilha. Ao zerar esta pilha, o CLP deve configurar esta variável novamente em 0 (zero). A segunda palavra de controle contém o tamanho de uma pilha em valores do tipo **Word**. Assim, para um Tag Bloco de tamanho *N*, cada célula deve ter o tamanho $N + 4$, ou seja, $N \text{ Elementos} + \text{Mês} + \text{Hora} + \text{Ano} + \text{Soma}$, e o tamanho total desta pilha é, portanto, o número de células multiplicado por $N + 4$.
- Ao configurar a primeira variável de controle em 1 (um), uma aplicação indica ao CLP que iniciou um processo de leitura. Enquanto durar esta operação, novos dados não devem ser colocados nesta pilha. Neste caso, os dados devem ser armazenados em outra área e transferidos para esta pilha após o término desta operação, ou seja, depois de uma aplicação configurar a variável em 2 (dois) e o CLP ter zerado esta pilha.

- Para evitar problemas em caso de erro na leitura, o CLP deve voltar a variável de controle para a posição inicial 0 (zero) após três minutos do início desta operação.

Eventos do CLP

Esta seção contém informações necessárias para a correta utilização dos eventos de um CLP com este Driver.

Requisitos de Utilização

Para o correto funcionamento deste recurso neste Driver, o programa do CLP deve manter as seguintes variáveis:

- **W03, no endereço 8206h:** Apontador de escrita de um *buffer* circular. Deve ser atualizado pelo programa do CLP sempre que um novo evento é adicionado a este *buffer*.
- **W04, no endereço 8208h:** Apontador de leitura de um *buffer* circular. Deve ser zerado pelo CLP ao ser ligado. Sempre que o CLP adiciona um evento a este *buffer*, o contador de *overflow* deve ser incrementado caso este *buffer* tenha estourado, ou seja, se **W03** é igual a **W04**. Antes de realizar o *download* deste *buffer*, a aplicação consulta o contador de *overflow*. Se este contador é diferente de 0 (zero), então o apontador de leitura é desconsiderado e o *buffer* inteiro é considerado válido.
- **W05, no endereço 820Ah:** Contador de *overflow* de um *buffer* circular. Deve ser zerado pelo CLP ao ser ligado. Sempre que o CLP adiciona um evento a este *buffer*, deve-se incrementar o contador de *overflow* caso o *buffer* tenha estourado, ou seja, se **W03** é igual a **W04**. Antes de realizar o *download* deste *buffer*, a aplicação consulta o contador de *overflow*. Se este contador é diferente de 0 (zero), então o apontador de leitura é desconsiderado e o *buffer* inteiro é considerado válido.
- **W06 ou W07, no endereço 820Ch ou 820Eh:** Armazena a última base de tempo utilizada por uma aplicação, um **DWord** de 32 bits. Deve ser zerada pelo CLP ao ser ligado. A cada *download* deste *buffer*, a aplicação atualiza esta variável. A variável **W06** contém os 16 bits menos significativos e a variável **W07** contém os 14 bits mais significativos. Esta base é desconsiderada por uma aplicação caso haja *overflow* do *buffer* circular.
- **W08 ou W09, no endereço 8210h ou 8212h:** Próxima base de tempo a ser utilizada, um **DWord** de 32 bits. Deve ser zerada pelo CLP ao ser ligado. A aplicação atualiza esta variável antes de enviar um comando de sincronia de relógio, que consiste na ativação de uma saída de um dos CLPs, que deve estar conectada a uma entrada específica em todos os outros CLPs. A ativação desta entrada deve ser interpretada por uma aplicação como um comando de sincronia de relógio. Quando o CLP recebe um comando de sincronia, deve adicionar ao *buffer* um evento de troca de base de tempo, o evento de sincronia, contendo o valor destas memórias. A variável **W08** contém os 16 bits menos significativos e a variável **W09** contém os 14 bits mais significativos. Este comando deve ser acionado periodicamente por uma aplicação, devido à imprecisão do relógio dos controladores.
- **Buffer, no endereço 8A00h em diante:** *Buffer* de armazenamento de eventos. Trata-se de um *buffer* circular, com os índices de leitura e escrita retornando 0 (zero) após ultrapassar o final. O espaço a ser alocado para este *buffer* depende do número de eventos definidos na janela de configuração deste Driver.

NOTA

As bases de tempo, **DWords** de 32 bits, contém o número de segundos desde 1/1/1998 00:00:00. Apenas os 30 bits menos significativos são utilizados, o que resulta em um intervalo de datas representáveis entre 01/01/1998 00:00:00 e 10/01/2032 13:37:03. Um CLP não precisa manipular estes números, apenas copiá-los quando necessário.

Tipos de Eventos

Cada evento é composto por uma variável **DWord** de 32 bits. Os dois bits mais significativos indicam o tipo de evento:

- **0**: Canal em 0 (zero). Os 9 (nove) bits menos significativos da primeira variável **Word** indicam o endereço da entrada que gerou este evento. A segunda variável **Word** contém o valor do contador de centésimos no momento deste evento.
- **1**: Canal em 1 (um). Os 9 (nove) bits menos significativos da primeira variável **Word** indicam o endereço da entrada que gerou este evento. A segunda variável **Word** contém o valor do contador de centésimos no momento deste evento.
- **10**: *Overflow* no contador de centésimos.
- **11**: Sincronia e mudança na base de tempo. Os 14 bits menos significativos da primeira variável **Word** contém os bits de 16 a 30 do novo tempo base. A segunda variável **Word** contém os 16 bits menos significativos, de 0 (zero) a 15, do novo tempo base. Este evento é ativado periodicamente por uma aplicação, conforme descrito no tópico Requisitos de Utilização.

Tarefas do CLP

O programa do CLP deve ser responsável por executar as seguintes tarefas:

- **Ao ligar**: Zerar as memórias das variáveis de **W04** a **W09**
- **Após adicionar um evento no buffer de memória**: Verificar se houve *overflow*, ou seja, se a variável **W03** é igual à variável **W04**. Caso positivo, incrementar a variável **W05**

O apontador de leitura permite ao CLP detectar e sinalizar a uma aplicação sobre o *overflow* do *buffer* circular, informação crucial na validação dos dados.

O CLP registra o banco de dados para uma aplicação junto com os eventos de sincronia. Isto permite que uma aplicação desligada possa interpretar corretamente o contador de centésimos dos eventos do *buffer*.

O armazenamento da última base de tempo utilizada por uma aplicação, variáveis **W06** ou **W07**, permite que esta aplicação seja interrompida e reiniciada a qualquer momento. Desta forma, os eventos não são perdidos e os registros de tempo são completamente confiáveis. Só ocorre perda se houver *overflow* no *buffer* do CLP.

A indicação de *overflow* da memória permite que uma aplicação tente recuperar um número maior de eventos, pois quando isto ocorre, o contador de leitura é desconsiderado e todo o *buffer* é lido. Esta indicação também valida ou não a última base de tempo utilizada por este Driver, nas variáveis **W06** ou **W07**.

Quando mais de uma aplicação está presente, deve ser criado um mecanismo de comunicação entre os CLPs, de forma que quando uma das aplicações sincroniza o relógio, este possa passar para um dos CLPs a nova base de tempo, e este CLP informa aos outros CLPs na rede a nova base de tempo, copiando o conteúdo das variáveis **W08** ou **W09**. Isto permite que uma aplicação atualize a base de tempo de todos os CLPs, sem que para isto todas as aplicações estejam ativas.

Tags da Tabela de Eventos

Esta seção contém informações sobre os Tags especiais para a leitura de eventos de um CLP.

NOTA

É possível ler e escrever em Elementos de Tags Bloco, com exceção de Tags que utilizem estampa de tempo, ou seja, com o parâmetro *B2* igual a 5 (cinco) ou o parâmetro *B2* igual a 999, que são somente de leitura.

Tag de Baixa de Eventos

Leitura

N1	Número do CLP, maior que 0 (zero)
N2	999
N3	0 (zero)
N4	0 (zero)

Este Tag executa o *download* dos dados da tabela de eventos do CLP para a memória interna deste Driver. Os dados permanecem nesta memória até que uma leitura seja solicitada. Para um funcionamento satisfatório, recomenda-se que este Tag seja lido periodicamente.

Tag de Leitura de Eventos

Leitura

N1	Número do CLP, maior que 0 (zero)
N2	999
N3	2 (dois)
N4	0 (zero)

A cada leitura deste Tag, este Driver retira o próximo evento armazenado na memória interna e retorna este evento neste Tag. O valor retornado neste Tag indica o tipo do evento e a estampa de tempo deste Tag indica a data e a hora em que este evento ocorreu.

Tag de Última Base de Tempo

Leitura

N1	Número do CLP, maior que 0 (zero)
N2	999
N3	3 (três)

N4	0 (zero)
----	----------

Esta base de tempo é utilizada internamente por este Driver para calcular a data e a hora dos eventos.

Tag de Próxima Base de Tempo

Leitura

N1	Número do CLP, maior que 0 (zero)
N2	999
N3	4 (quatro)
N4	0 (zero)

Uma aplicação deve programar a próxima base de tempo antes de enviar um comando de sincronia de relógio. Este comando deve ser enviado através do acionamento de uma determinada saída de um dos CLPs, que está ligada fisicamente a uma determinada entrada em todos os outros CLPs da rede, entrada esta que deve ser monitorada pela aplicação dos CLPs.

Quando o CLP recebe o comando de sincronia, através do acionamento da entrada mencionada, este deve adicionar ao *buffer* um evento de sincronia contendo o valor escrito por este Tag, **W08** ou **W09**. Para mais informações, consulte o tópico Requisitos de Utilização.

Tag Bloco de Leitura de Eventos

Leitura

B1	Número do CLP, maior que 0 (zero)
B2	999
B3	1 (um)
B4	0 (zero)

Verificação do primeiro evento da memória interna deste Driver. Preencha este Tag Bloco com informações sobre o próximo evento armazenado na memória interna deste Driver:

- **Elemento 0:** Estampa de tempo de um evento
- **Elemento 1:** Endereço da entrada que causou este evento
- **Elemento 2:** Tipo deste evento

Para um funcionamento satisfatório, deve ser executado um *polling* periódico neste Tag Bloco para verificar se existem eventos na memória deste Driver. Se a memória deste Driver está vazia, todos os Elementos deste Tag Bloco retornam com o valor -1 (menos um). Caso contrário, este Tag Bloco retorna o primeiro evento da memória interna deste Driver. Da mesma forma, se há eventos, o Tag de Leitura de Eventos, com o parâmetro *N2* igual a 999 e o parâmetro *N3* igual a 2 (dois), pode ser lido normalmente. Esta leitura retira o primeiro evento da memória interna deste Driver.

Documentação das Interfaces de Comunicação

Esta seção contém a documentação das Interfaces de Comunicação referentes ao Driver **BCM1_MasterSlave**.

Esta documentação é compartilhada e mantida no manual do IOKit:

- Configurações de um Driver
- Configurações Gerais
- Configuração de Estatísticas
- Configuração da Interface Ethernet
- Configuração da Interface Modem
- Configuração da Interface RAS
- Configuração da Interface Serial

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
3.0.3	10/08/2026	M. Ludwig	• Driver atualizado para a biblioteca IOKit versão 3 e Visual Studio 2022 (<i>Case 37944</i>).
3.0.2	31/08/2021	C. Mello	• Ajustes no processo de coleta de Tags Bloco com estampa de tempo (<i>Case 31183</i>).
3.0.1	11/09/2014	G. Taschetto	• Driver migrado para a biblioteca IOKit versão 2 (<i>Case 16993</i>). • Corrigido um problema na leitura de Blocos de Words (<i>Case 16988</i>).
2.0.1	09/04/2012	G. Taschetto	• Driver migrado para a biblioteca IOKit versão 1.15 .
1.1.1	27/04/2009	A. Quites	• Corrigido um GPF (<i>General Protection Failure</i>) na escrita de Blocos com mais de 25 Elementos (<i>Case 10385</i>).
1.0.1			• Todas as publicações anteriores ao controle de revisões.