

FX-S50

Sensor de obstáculos Infravermelho

1. Descrição

Sensor digital de obstáculos compacto, rápido e configurável. Possui uma saída digital para detecção imediata, e um pino para configuração ou leitura via protocolo FoxWire, permitindo a conexão simultânea de até 32 sensores em paralelo (compartilhando o mesmo fio). A configuração pode ser realizada por meio de leitura e escrita de registradores via protocolo FoxWire ou pelo modo Shell, que utiliza comandos de texto.



2. Principais Características

- Faixa de medição: 50cm (*)
- Tensão de Alimentação: 3.3V a 5V
- Corrente de operação: 12mA a 16mA
- Dimensões: 11,4 x 12,4 x 16,2 mm
- Comunicação: Saída digital e Fox Wire
- Peso: 4,9 g
- Alcance ajustável

- Suporte ao modo "Shell"

3. Aplicações Típicas

- Robótica móvel
- Sumô de robôs (Mini ou Micro)
- Animatrônicos
- Contagem de objetos
- Detecção de movimento

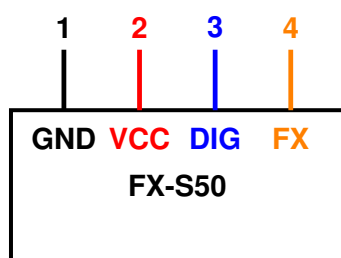


Figura 3.1: Pinagem

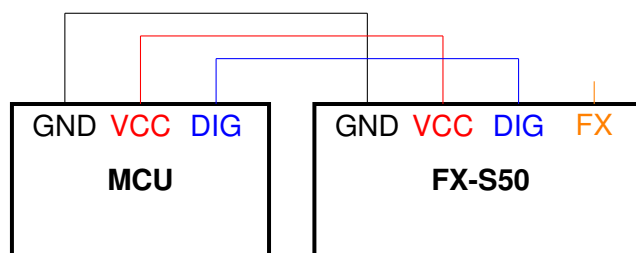


Figura 3.2: Circuito usando pino digital

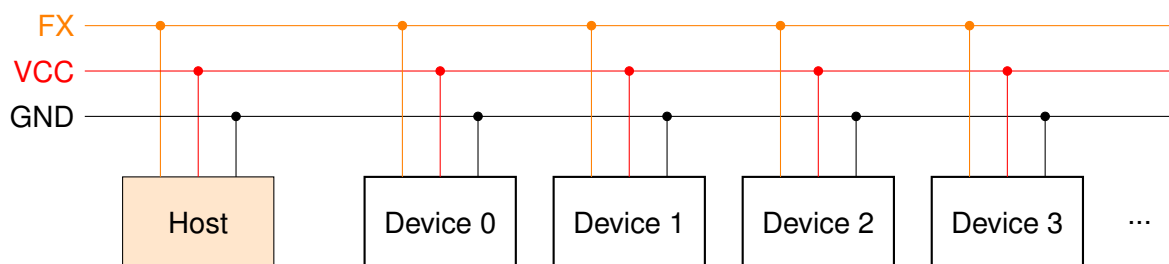


Figura 3.3: Circuito usando Fox Wire

4. Pinagem

Tabela 4.1: Pinagem

Simbolo	Pino	Tipo ⁽¹⁾	Descrição
GND	1	PWR	Pino de Ground (0V)
Vcc	2	PWR	Pino de Alimentação
DIG (S)	3	I	Saída digital. Obstaculo detectado = HIGH, não detectado = LOW
FX	4	I/O	Fox Wire. Pino de configuração ou leitura.

(1) Tipos de sinais: I = Input (entrada), O = Output (saída), I/O = entrada ou saída, PWR = Power.

5. Características Elétricas

Tabela 5.2: Características elétricas

Parâmetros		Mínimo	Típico	Máximo	Unidade
V_{CC}	Tensão de Alimentação	3.3	5	-	V
I_{PWR}	Corrente de Alimentação	12	-	16	mA
V_{OD}	Tensão de saída do pino digital	0	-	3,3	V
I_{OD}	Corrente de saída do pino digital	-	-	30	mA
F_{aq}	Frequência de aquisição	-	-	10	kHz

6. Descrição detalhada

O FX-S50 é um sensor digital de obstáculos infravermelho. O sensor emite luz infravermelha com características específicas que a diferenciam de outros sinais. A detecção é realizada com base na luminosidade refletida pelo obstáculo, o que faz com que o alcance varie conforme o tamanho, a cor e a geometria do objeto. Na configuração padrão, o sensor pode detectar até aproximadamente 50cm um objeto branco plano com dimensões de 7x7cm e até cerca de 17cm um objeto preto plano com dimensões similares.

O sensor dispõem de registradores que armazenam suas configurações, permitindo ajustar o alcance de detecção, os parâmetros de comunicação e a filtragem de ruído.

A leitura do sensor pode ser realizada de duas formas:

1. Usando o *pino digital (DIG)*, que apresenta nível lógico alto (3,3V) quando detecta um obstáculo e nível lógico baixo (0V) na ausência de detecção.
2. Usando o *pino FX*, que utiliza o protocolo FoxWire, que permite a comunicação com um ou vários sensores usando o mesmo fio.

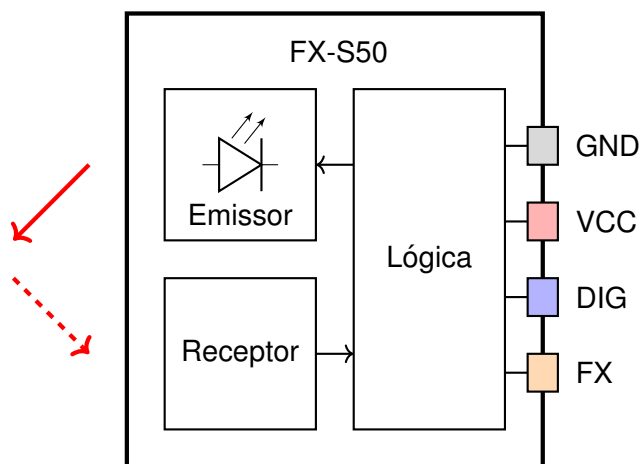


Figura 6.1: Diagrama funcional

6.1 Saída digital

O pino de saída digital (DIG) é continuamente atualizado pelo sensor. Enquanto um obstáculo é detectado o pino assume nível lógico alto (3,3V), e retorna ao nível lógico baixo (0V) na ausência de detecção.

Este pino pode ser conectado diretamente a uma entrada digital de um microcontrolador. Em geral, microcontroladores operando a 5V conseguem identificar sinais digitais de 3,3V, eliminando a necessidade de um conversor de nível, o que simplifica a integração.

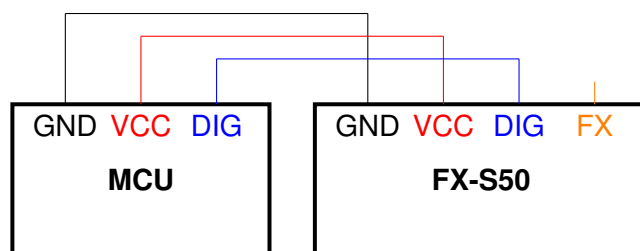


Figura 6.2: Circuito usando pino digital

6.2 Tratamento do sinal de saída

O sensor realiza o tratamento computacional do sinal medido para reduzir o ruído na saída. Esse tratamento é implementado por meio de um integrador digital com saturação. Os limites do integrador e os valores de transição da porta são parametrizados por dois registradores: `FILTER_LEN` e `FILTER_TRIG`, cujas funções são descritas a seguir:

- **FILTER_SIZE:** Define o comprimento do integrador. Valores maiores resultam em uma taxa de atualização mais lenta no sinal de saída, mas com menor suscetibilidade a ruídos.
- **FILTER_TRIG:** Define a sensibilidade do sensor a transições rápidas. Valores mais altos tornam o sensor menos sensível a mudanças bruscas, sem alterar a taxa de atualização. Deve ser configurado dentro do intervalo de **0** a **FILTER_SIZE-1**.

7. Opções de configuração e seus Registradores

O sensor possui alguns registradores de 1 byte que armazenam suas configurações. Os valores são salvos em uma memória não volátil após a execução do comando "save". Os registradores disponíveis são descritos na tabela a seguir.

Tabela 7.3: Tabela de Registradores

Endereço	Nome	Tipo ⁽¹⁾	Descrição	Valor Padrão
0x00	FX_ADDR	RW	Endereço do dispositivo no protocolo Fox Wire	0x00
0x01	RESERVADO	-	-	-
0x02	LED_FREQ	RW	Controle da frequência do emissor. Quanto mais próximo de 100 mais sensível. De 0 a 255.	120
0x03	LED_POWER	RW	Controle da potência do emissor. Quanto maior o valor mais sensível. De 10 a 100.	15
0x04	FILTER_SIZE	RW	Tamanho do integrador usando para filtragem	6
0x05	FILTER_TRIG	RW	Valor de transição após integrador	4
0x06	RESERVADO	-	-	-
0x07	RESERVADO	-	-	-
0x10 - 0x2F	DEVICE_NAME	RW	Nome do dispositivo	"FX-S50"
0x30 - 0x7F	RESERVADO	-	-	0x00

(1) Tipos de registradores: R = Read (Somente Leitura), W = Write (Somente Escrita), RW = Leitura ou Escrita.

8. Comunicação usando Fox Wire

O protocolo de comunicação opera via USART em modo *half-duplex*, permitindo que um *host* se conecte a até 32 dispositivos utilizando apenas um único fio para transmissão e recepção de dados. A comunicação segue a configuração padrão USART 8N1: 8 bits de dados, sem bit de paridade e 1 bit de parada. Nesse modo, os dados fluem em apenas uma direção por vez: ou o *host* ou um dos dispositivos transmite, enquanto os demais permanecem em espera. Essa abordagem previne colisões no barramento compartilhado, garantindo uma comunicação eficiente e coordenada entre todos os dispositivos conectados. Para mais informações sobre o protocolo FoxWire, consulte o manual.

No protocolo FoxWire, o sensor FX-S50 atua como um dispositivo (equivalente a *slave*), ou seja, ele responde às requisições de um *host* (equivalente a **mestre**), que pode ser, por exemplo, um microcontrolador. A comunicação ocorre por meio da troca de pacotes com tamanhos previamente definidos. Os pacotes disponíveis são:

- **CHECK**: 2 bytes.
- **READ**: 3 bytes.
- **WRITE**: 4 bytes.
- **SPECIAL**: de 3 a 34 bytes.

8.1 Cálculo do checksum

O valor do *checksum* é calculado pela fórmula a seguir, que representa a soma dos cinco bits menos significativos ($R0, R1, \dots, R4$). A soma é limitada ao valor máximo de 3, uma vez que o *checksum* ocupa apenas dois bits.

$$\text{checksum} = 3 \ \& \ (R0 + R1 + R2 + R3 + R4) \quad (1)$$

8.2 Pacote CHECK

Quando o endereço enviado pelo *host* é compatível, o sensor FX-S50 responde ao comando **CHECK** com um byte contendo seu próprio endereço e o estado atual do sensor. Este pacote é útil para o *host* realizar uma leitura rápida do sensor.

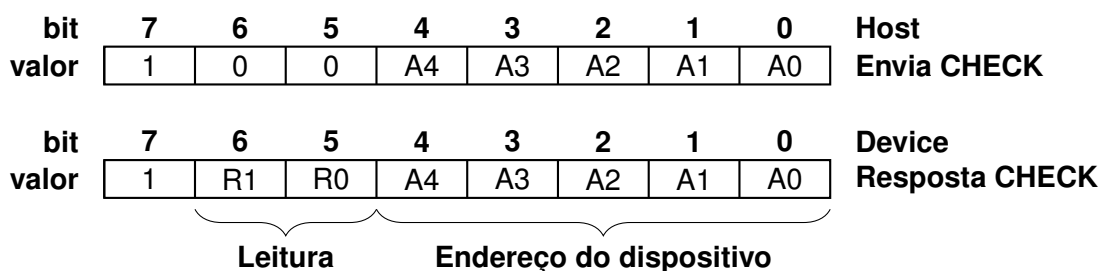


Figura 8.1: Byte de resposta ao CHECK

8.3 Leitura de registrador

A leitura de registradores é realizada por meio do pacote **READ**. Esse pacote permite ao *host* acessar o conteúdo de registradores específicos do sensor, enviando o endereço do registrador e recebendo o valor armazenado como resposta.



Figura 8.2: Leitura de registrador

8.4 Escrita de registrador

A escrita de registradores é realizada utilizando o pacote **WRITE**. Esse pacote permite ao *host* enviar um valor específico para um registrador do sensor, alterando suas configurações ou parâmetros de operação conforme necessário.

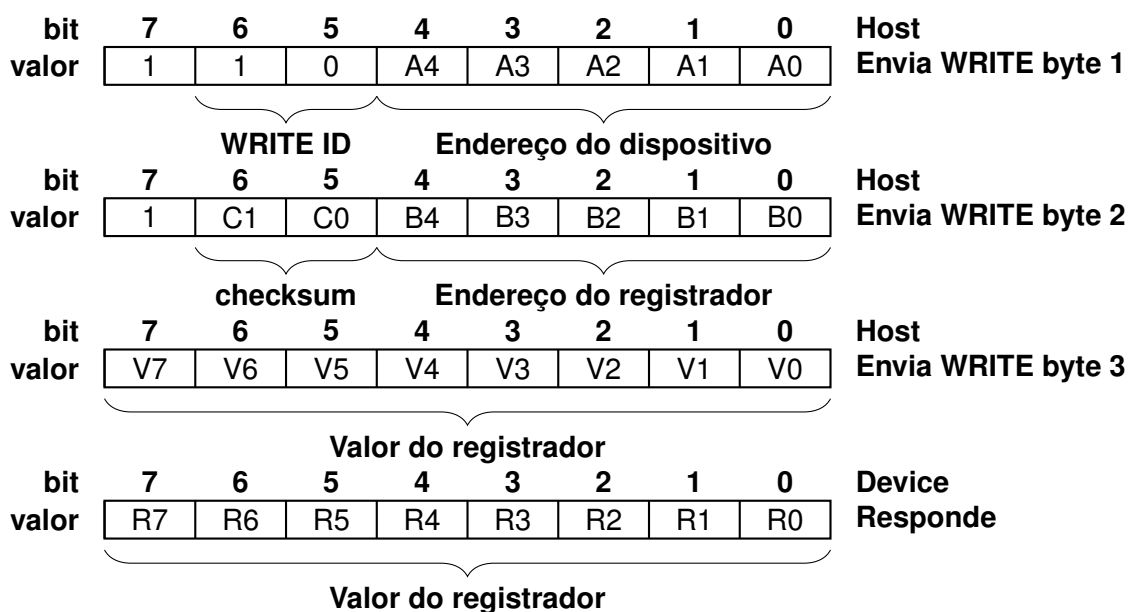


Figura 8.3: Escrita de registrador

8.5 Comandos

O sensor oferece diversos comandos via protocolo FoxWire, permitindo operações como reiniciar, restaurar ou salvar configurações, entre outras. Os comandos estão organizados em dois tipos principais:

- **Comandos Simples:** Executados com apenas uma troca de pacote, utilizando **READ** ou **WRITE**.
- **Comandos com Chave:** Executados em duas etapas, envolvendo um pacote **READ** seguido de um **WRITE**, adicionando uma camada de segurança para evitar execuções acidentais.

8.6 Lista de Comandos

A Tabela a seguir descreve os comandos disponíveis da ultima versão do firmware.

Tabela 8.4: Tabela de Comandos especiais

Valor	Nome	Tipo	Descrição
0x00	DEVICE_ID	READ	Lê o identificador do dispositivo
0x01	FOXWIRE_VERSION	READ	Lê a versão do protocolo Fox Wire
0x02	FIRMWARE_VERSION	READ	Lê a versão do firmware
0x03	READ	READ	Lê o estado do sensor
0x04	RESET	READ	Reinicia o sensor
0x05	KEY_REQUEST	READ	Requisição de uma chave
0x00	RESTORE	Chave	Restaura as configurações para os valores padrão
0x01	RESTORE_KEEP_ADDR	Chave	Restaura as configurações para os valores padrão
0x02	SAVE	Chave	Salva as alterações na flash

8.7 Comandos Simples

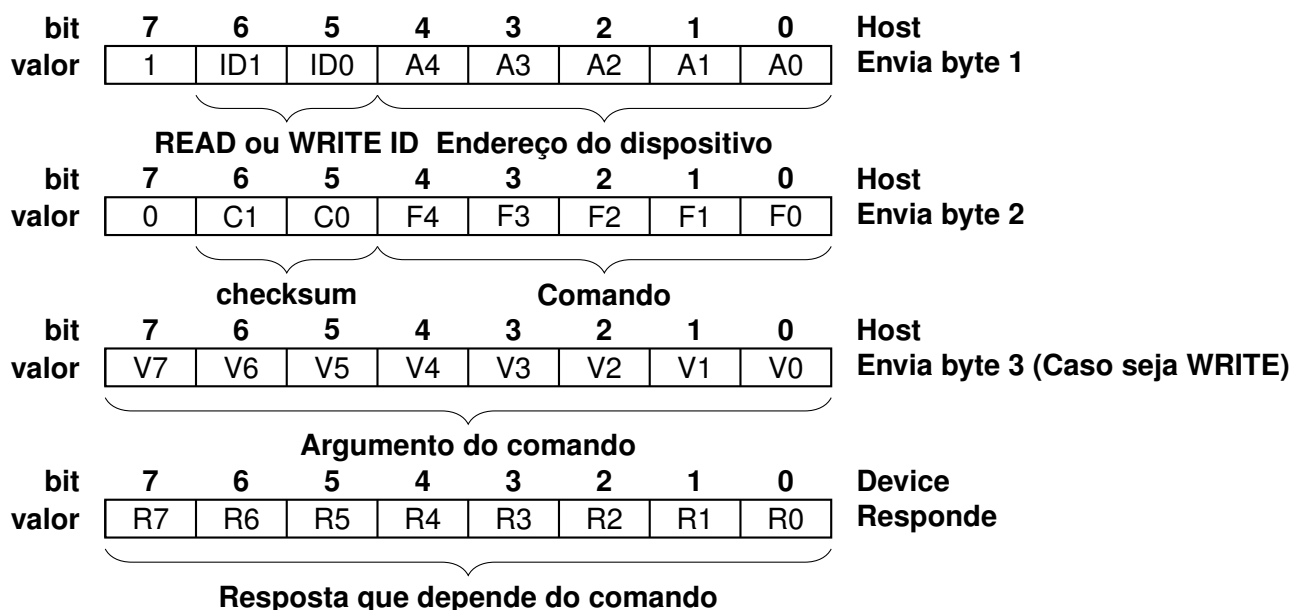


Figura 8.4: Comandos simples

8.8 Comandos com Chave

O comando com chave é executado em duas etapas para aumentar a segurança. Na primeira etapa, o *host* envia uma requisição utilizando um pacote **READ**, e o sensor responde com uma chave temporária. Na segunda etapa, o *host* envia o comando desejado usando um pacote **WRITE**, inserindo no valor do pacote a chave temporária recebida, porém invertida bit a bit. O sensor então verifica o valor recebido e, se ele for compatível com a chave temporária, o comando é executado. Esse processo pode ser resumido como dois **comandos simples**: uma requisição de chave com

um pacote **READ** e a execução do comando com a chave, utilizando um pacote **WRITE**. Caso a chave recebida no Pacote 1 seja nula isso indica que houve uma falha de comunicação.

Pacote 1: READ / requisição de chave



Pacote 2: WRITE / comando com chave



Figura 8.5: Comando com chave

9. Modo Shell

O modo Shell oferece uma forma prática de configurar o sensor sem a necessidade de códigos ou bibliotecas específicas. É necessário apenas um programa de comunicação serial, como o *Putty* ou o *Serial Monitor* do Arduino. A conexão é simples e pode ser realizada com um conversor USB-Serial e um resistor para configurar a comunicação como UART em modo *Half-Duplex*, vide Figura 9.1.

Importante: O modo Shell permite a configuração de apenas um sensor por conexão serial. Caso outros sensores estejam conectados em paralelo, é necessário desconectá-los antes da configuração. Para configurar vários sensores simultaneamente, recomenda-se o uso da biblioteca Python *FoxWire*.

Após estabelecer a conexão serial com o monitor, digite **FOX-SHELL** para ativar o modo Shell. A Tabela 9. lista todos os comandos disponíveis.

Comandos relacionados a parâmetros do sensor, como `led_hz`, `led_power` e `address`, quando utilizados sem argumentos, são interpretados como comandos de leitura, retornando o respectivo valor configurado.

Tabela 9.5: Tabela de comandos *Shell*

Comandos	Descrição
FOX-SHELL	Palavra chave para iniciar o modo Shell
exit	Encerra o modo Shell
reset	Reinicia o sensor, consequentemente sai do modo Shell
help	Lista os comandos disponíveis
dump	Lista as configurações do sensor
save	Salva na memória não volátil as configurações que foram alteradas. Sem esse comando as alterações realizadas são perdidas após o desligamento.
register <addr >	Lê o registrador no endereço <addr >
address <addr >	Retorna ou altera ADDR para <addr > limitado de 0 a 31.
led_freq <val >	Retorna ou altera LED_FREQ (frequência normalizada do emissor) para <val > limitado de 0 a 255.
led_power <val >	Altera LED_POWER (potência do emissor) para <val > limitado de 10 a 100.
filter_size <val >	Altera FILTER_SIZE (comprimento do integrador) para <val > limitado de 1 a 255.
filter_trigger <val >	Altera FILTER_TRIG (sensibilidade a variação de sinal) para <val > limitado de 1 a FILTER_SIZE-1 .
name	Retorna o nome do sensor
restore	Restaura as configurações para os valores padrão
restore_keep_addr	Restaura as configurações para os valores padrão sem alterar o valor do endereço
read	Lê o estado do sensor

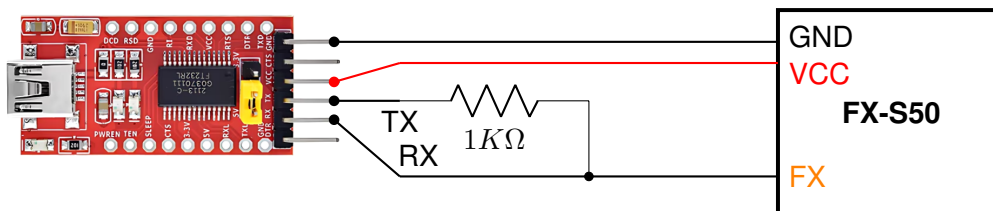


Figura 9.1: Usando um conversor USB serial como FoxWire

9.1 Instruções de uso

1. Conecte o sensor ao computador utilizando um conversor USB-Serial, vide figura 9.1, ou um Arduino.
2. Abra um aplicativo de comunicação serial, como *Putty* ou o *Serial Monitor* do Arduino. Configure a comunicação com uma taxa de baud (*baudrate*) de 115200. Na IDE Arduino, selecione a porta COM correta e escolha qualquer placa, por simplicidade, é selecionado um Arduino.
3. Digite **FOX-SHELL** para o sensor entrar no modo Shell. O sensor responderá com a mensagem "FOX-SHELL INIT!".
4. Com o modo Shell iniciado, é possível configurar o sensor ou realizar medições. Digite o comando **help** para exibir a lista de comandos disponíveis.
5. Utilize o comando **dump** para exibir os valores de configuração do sensor.

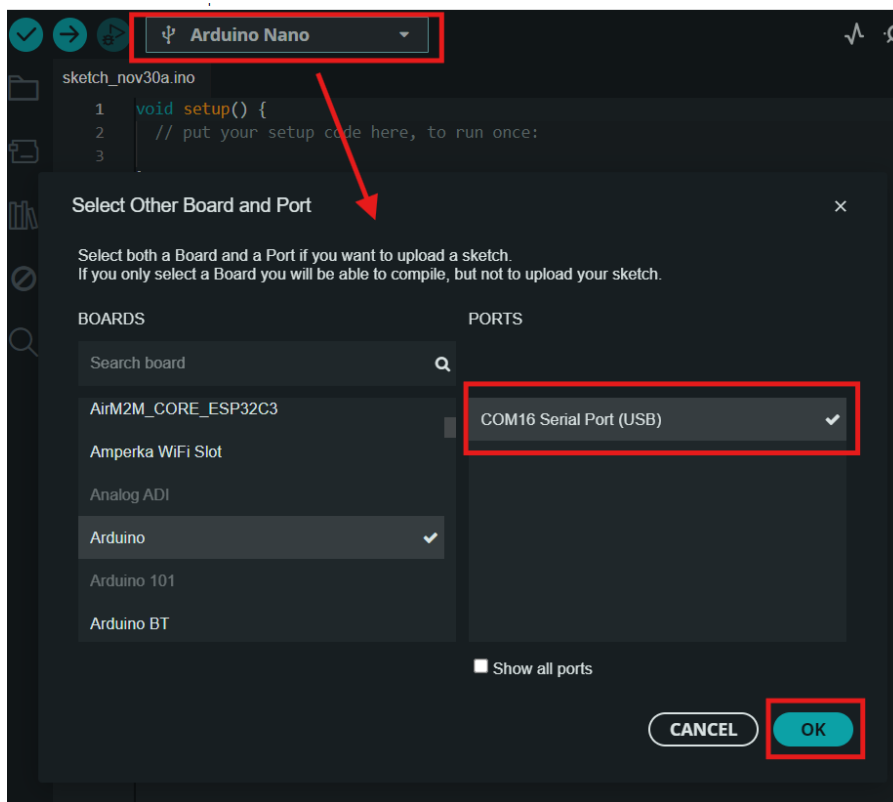


Figura 9.2: Configuração no *Seleção da COM no Arduino IDE*.

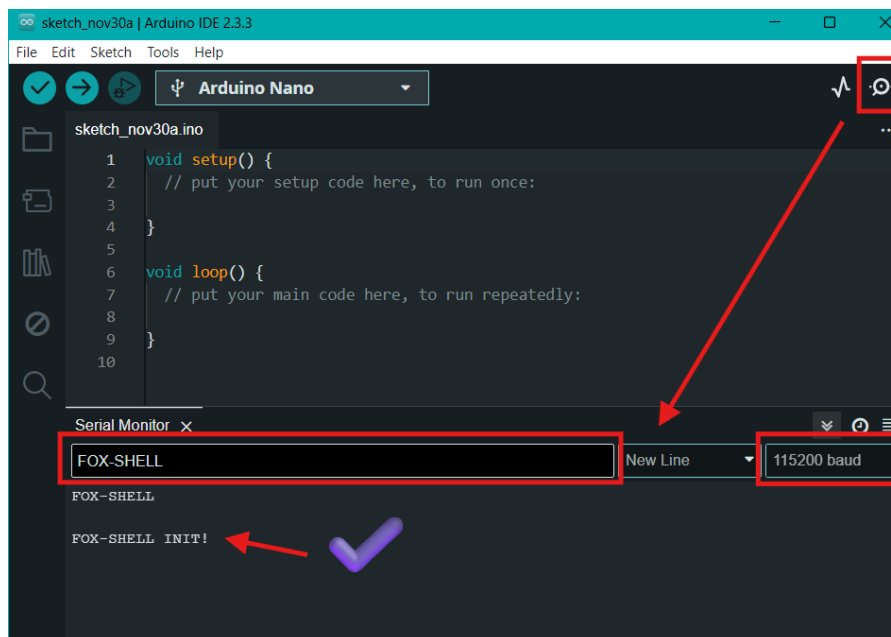


Figura 9.3: Entrada no modo Shell.

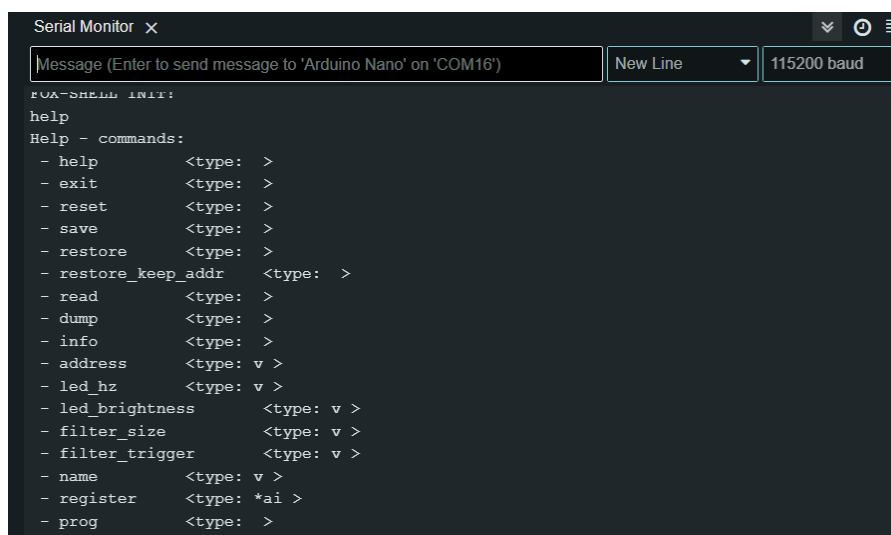


Figura 9.4: Exibição dos comandos disponíveis no modo Shell.

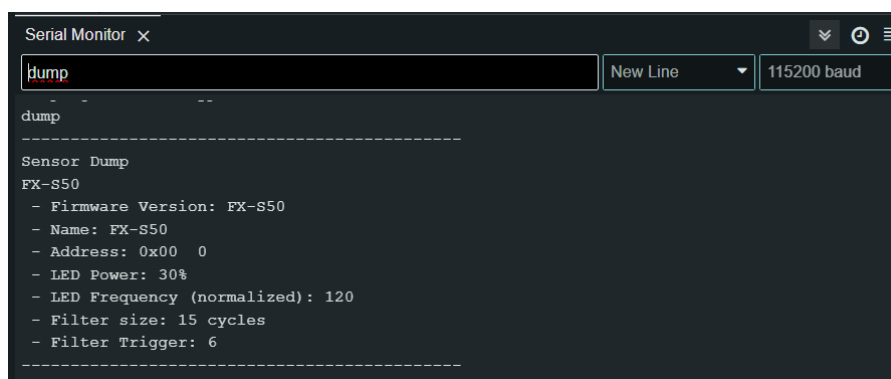


Figura 9.5: Valores de configuração exibidos pelo comando **dump**.

10. Esquemático de Aplicação

10.1 Circuito com Arduino UNO usando pino Digital

A Figura 10.1 apresenta o esquemático típico de conexão com alguns sensores usando o pino digital. Nesse circuito é utilizado um Arduino UNO, mas podem ser utilizados outros microcontroladores com entradas digitais.

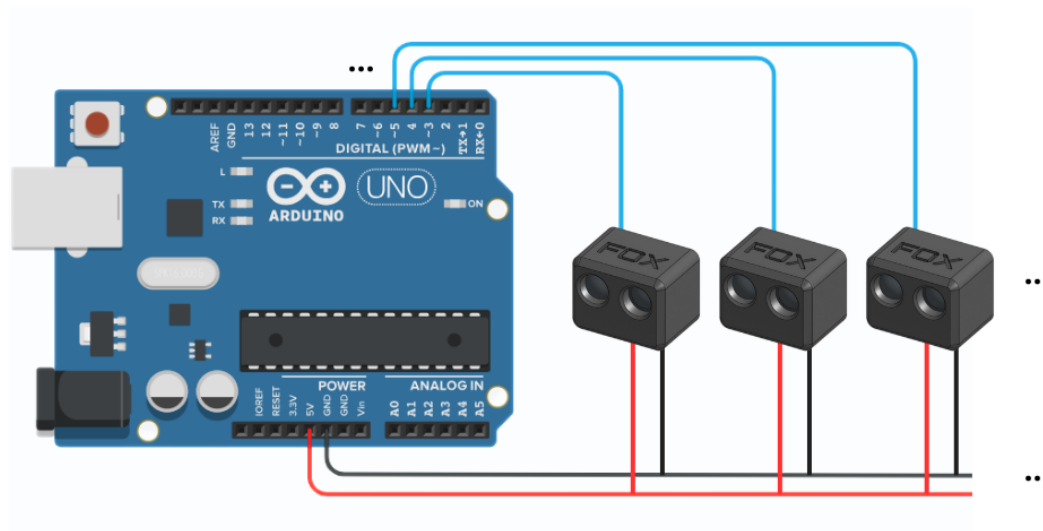


Figura 10.1: Circuito com Arduino UNO usando pinos digitais.

10.2 Circuito com Arduino UNO usando FoxWire

A Figura 10.2 apresenta um exemplo de circuito usando FoxWire e alguns sensores. O código está disponível no repositório FoxWire (<https://github.com/luisf18/FoxWire>). Consulte também o repositório do sensor FX-S50 (https://github.com/luisf18/FXDevices/tree/main/Sensor_FXS50).

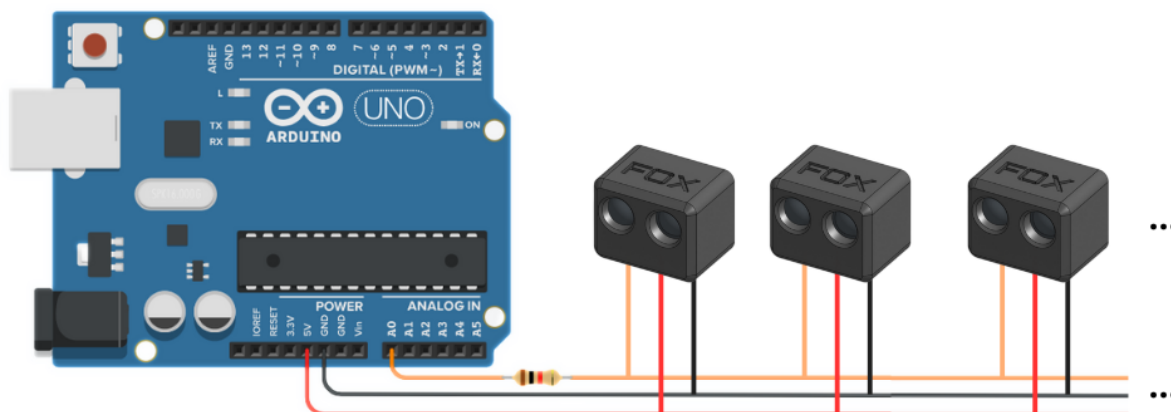


Figura 10.2: Circuito com Arduino UNO usando FoxWire.

11. Desenho Técnico e Dimensões

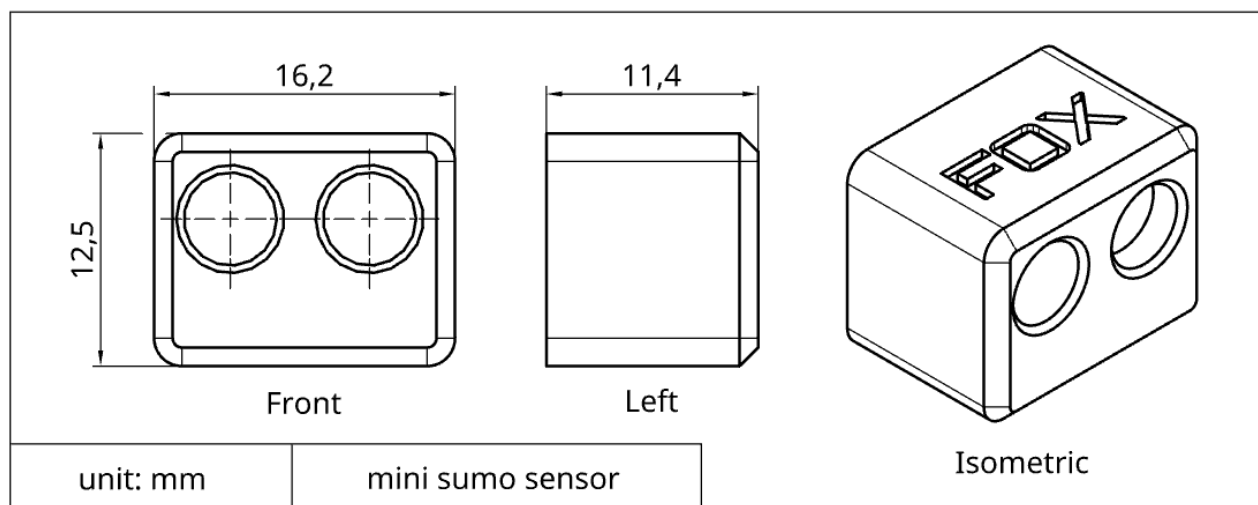


Figura 11.1: Desenho técnico com as dimensões do sensor. Todas as medidas estão em milímetros (mm).

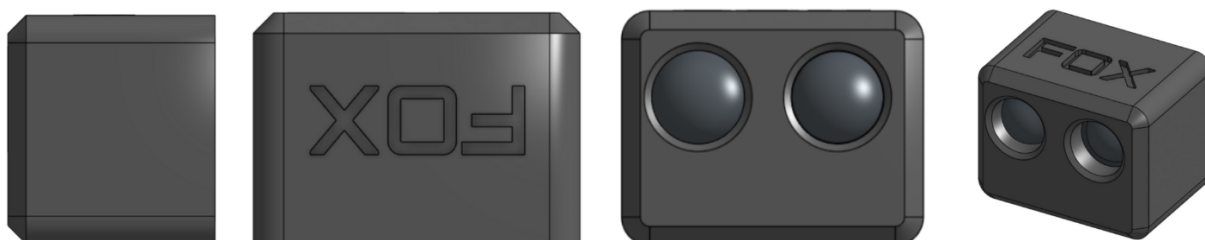


Figura 11.2: Vistas do sensor.

Links Relacionados:

- [Desenho Técnico em PDF](#)
- [Modelo 3D em arquivo STEP](#)

12. Repositórios

- github do sensor: https://github.com/luisf18/FXDevices/tree/main/Sensor_FXS50
- github FoxWire: <https://github.com/luisf18/FoxWire>

13. Informações de Contato

- E-mail: foxdynamicsteam@gmail.com
- Instagram: [@FoxDynamicsTeam](#)
- YouTube: [Fox Dynamics Team](#)