

DuinoBlocks4Kids: Instruções para Instalação, Configuração e Uso

1. DB4K-CS e DB4K-SA

O DuinoBlok4Kids possui duas versões, o [DB4K-CS \(Client Side\)](#) e o [DB4K-SA \(Satandalone\)](#). O DB4K-CS é multiplataforma, no entanto, exige a instalação do Python (ferramenta Livre) e de um navegador de internet, preferencialmente o Google Chrome (Navegador Gratuito). Já o DB4K-SA (*Satandalone*), está disponível apenas para o Windows 64 Bits, e dispensa a instalação do Python e o uso do navegador. Ambas as versões possuem exatamente as mesmas funcionalidades e é importante mencionar que ambas rodam localmente sem necessidade de acesso à Internet. Ambas as versões necessitam que o usuário tenha o software Arduino instalado na sua máquina.

O ambiente de programação DB4K possui uma [versão de demonstração online](#), no entanto, não é possível, nesta versão de demonstração, enviar os programas feitos no DB4K diretamente para uma placa Arduino conectada ao computador. Porém, pode-se baixar o programa em blocos construído no ambiente, no formato “.ino” (formato dos programas Arduino) e, então, abrir este programa no software Arduino e enviá-lo para uma placa Arduino conectada ao computador.

2. Instruções de Instalação, Configuração e Uso

2.1 Instalando o Ambiente

2.1.1 DB4K-SA (Windows 64)

1. Baixe e instale o **Software Arduino 1.6.8**, <https://www.arduino.cc/en/Main/OldSoftwareReleases#00xx>, na pasta C:\Program Files (x86)\Arduino (Destino padrão de instalação do Arduino)
2. Baixe e descompacte o arquivo **DB4K-SA-Beta-Win.zip**, (<http://www.nce.ufrj.br/ginape/livre/paginas/db4k/downloads/DB4K-SA-Beta-Win.zip>)
3. Copie as pastas **Ultrasonic** e **Thermistor** presentes na pasta Bibliotecas Arduino do pacote de instalação do DB4K para a pasta **C:\Program Files (x86)\Arduino\libraries**. Essas pastas contêm as bibliotecas necessárias para se utilizar os sensores de distância e de temperatura.
4. Execute o arquivo **DB4K-SA-B3.0.0-Win-Setup.exe**. Após a instalação, um ícone para execução do DB4K- SA ficará disponível na área de trabalho.

2.1.2 DB4K-CS (Multiplataforma)

1. Baixe e instale o Interpretador **Python 3.4.4** (<https://www.python.org/downloads/release/python-344/>) (No Ubuntu o Software Arduino Já vem previamente instalado)
2. Baixe e instale o **Software Arduino 1.6.8**, <https://www.arduino.cc/en/Main/OldSoftwareReleases#00xx>.
3. Baixe e descompacte o arquivo **DB4K-CS-Beta.zip** (<http://www.nce.ufrj.br/ginape/livre/paginas/db4k/downloads/DB4K-CS-Beta.zip>)
4. Copie as pastas **Ultrasonic** e **Thermistor** presentes na pasta Bibliotecas Arduino do pacote de instalação do DB4K para a pasta **libraries** localizada dentro da **pasta do Software Arduino**. Essas pastas contêm as bibliotecas necessárias para se utilizar os sensores de distância e de temperatura.
5. Copie a Pasta **DB4K-CS** (descompactada no item 3) para um local de sua preferência.
6. A aplicação é disparada por meio do arquivo **start.py** presente na pasta **DB4K-CS**. O DB4K irá abrir no navegador padrão. Sugerimos o uso do **Google Chrome**, mas, outros navegadores, em princípio, podem ser utilizados.

2.2 Configurando os Pinos de Entra e Saída

O DB4K vem com os pinos de entrada e saída utilizados pelos blocos previamente configurados para o uso com a "caixinha mágica" (apresentada no vídeo pitch e com instruções para construção presentes neste vídeo: https://youtu.be/dHW5nmfMs_g), que faz uso de uma placa Arduino Mega, e os demais materiais de robótica desenvolvidos e sugeridos para uso com o DB4K, que fazem uso de placas Arduino Uno. No entanto, é possível utilizar o DB4K para o controle de qualquer circuito que faça uso dos sensores e atuadores contemplados pelo DB4K. Assim, caso se deseje modificar os pinos para o controle de circuitos montados com outra configuração de pinos e de entrada e saída, os números dos pinos podem ser reconfigurados. Toda essa configuração das pinagens, evidentemente, é feita pelo professor, ela é transparente para as crianças, ao menos em um primeiro momento.

Caso deseje modificar os números dos pinos, edite os valores da seguinte maneira:

2.2.1 Modificando os pinos de entrada e saída no DB4K-SA

Os números dos pinos devem ser modificados no arquivo **DB4K-SA\ardublockly\db4k_setup.js** (Este arquivo pode ser editado utilizando-se programas como o Notepad++ ou o Bloco de Notas do Windows).

IMPORTANTE: Após modificar os valores dos pinos e salvar o arquivo de configuração, ao rodar o DB4K, é necessário fazer um "reload" da aplicação, de maneira que essas modificações sejam refletidas no código Arduino gerado pelo ambiente. Para tanto, entre no menu <Configurações> e selecione a opção <atualizar pinos>

2.2.2 Modificando os pinos de entrada e saída no DB4K-CS

Os números dos pinos devem ser modificados no arquivo **DB4K-SA\DB4K\db4k_setup.js** (Este arquivo pode ser editado utilizando-se programas como o Bloco de Notas, o Notepad++ ou o Gedit).

IMPORTANTE: Após modificar os valores dos pinos e salvar o arquivo de configuração, ao rodar o DB4K, é necessário fazer um "reload" da aplicação, de maneira que essas modificações sejam refletidas no código Arduino gerado pelo ambiente. Para tanto, entre no menu <Configurações> e selecione a opção <atualizar pinos>. Caso o reload da página não seja feito e os números dos pinos não sejam atualizados, aperte ao mesmo tempo as teclas Ctrl e F5, para forçar a atualização do cache do navegador.

2.2.3 Tabela com a configuração inicial dos Pinos do DB4K

A tabela abaixo apresenta os pinos previamente configurados no DB4K.

Componenete		Pino	Componenete	LED	Pino	Componenete	Sinal	Pino
LED	VERMELHO	9	Display_7_SEG.	A-7	24	Display_LCD	RS	31
LED	AMARELO	10	Display_7_SEG.	B-6	22	Display_LCD	RW	33
LED	VERDE	11	Display_7_SEG.	C-4	27	Display_LCD	EN	35
LED	AZUL	12	Display_7_SEG.	D-2	25	Display_LCD	D4	30
LED_RGB	R	2	Display_7_SEG.	E-1	23	Display_LCD	D5	32
LED_RGB	G	4	Display_7_SEG.	F-9	26	Display_LCD	D6	34
LED_RGB	B	5	Display_7_SEG.	G-10	28	Display_LCD	D7	36
LED BRANCO	LOAD	13	Display_7_SEG.	PD-5	29			
DISTANCIA	TRIG	6	Componenete		Pino			
DISTANCIA	ECHO	7	MOTOR DC		3			
TERMISTOR		A0	SERVO		1			
LDR		A1	BUZZER		8			

2.3 Criando um programa e enviando para a placa Arduino

2.3.1 Configurações Iniciais do Ambiente.

Entre no menu <Configurações> e selecione a opção <Preferências>. Na Janela que abre, especifique:

1. A localização do compilador Arduino.
 - a. DB4K-SA - **C:\Program Files (x86)\Arduino\arduino_debug.exe** (Caso você tenha instalado o Software Arduino em outra pasta, especifique aqui a localização do mesmo)
 - b. DB4K-CS - Você deve digitar o caminho completo até o arquivo **arduino_debug** (windows) ou **Arduino** (Linux), presente da pasta onde o Software Arduino foi instalado. Exemplo: Windows: **C:\Program Files (x86)\Arduino\arduino_debug.exe** | Linux: **/opt/arduino-1.6.8/Arduino**

2. A pasta do Projeto.
 - a. DB4K-SA - Por padrão, a pasta do projeto é a pasta raiz do DB4K, ou seja: **C:\DB4K-SA**, mas você pode especificar outra pasta de sua preferência.
 - b. DB4K-CS - Você deve digitar o caminho completo até a pasta DB4K-CS. Ex: Windows: **C:\DB4K-CS** | Linux: **/home/nomedousuario/DB4K-CS**

Estas configurações precisam ser feitas apenas na primeira vez que o ambiente for utilizado

2.3.2 configurações da placa Arduino e da porta COM

Entre no menu <Configurações> e selecione a opção <Preferências>. Na Janela que abre, especifique:

1. A **Placa Arduino** desejada
2. A **Porta COM** a ser utilizada

Estas configurações têm que ser feitas sempre que você conectar uma placa Arduino no computador.

Obs. Para se certificar de que o DB4K configurou a porta COM adequadamente, clique sobre o nome da porta mesmo que ela já tenha sido definida automaticamente pelo DB4K.

2.3.3 Enviando um programa para a placa Arduino

Para construir um programa, escolha os blocos desejados no menu de blocos e os posicione na área de trabalho, encaixando-os na ordem desejada. Os blocos de estruturas condicionais usam os blocos de sensores como parâmetros condicionantes. Os códigos construídos com o DB4K rodam dentro de um loop infinito, ou seja, a sequência de comandos será executada indefinidamente (a não ser que se utilize o bloco “parar programa”, que é utilizado apenas em algumas poucas atividades, muito específicas).

Para carregar um programa na placa Arduino selecionada você deve montar o programa em blocos e apertar o botão laranja presente na parte superior da área de montagem de programas. Esse botão envia o programa diretamente para a placa Arduino configurada anteriormente.

Caso você deseje parar a execução do programa aperte o botão vermelho presente na parte superior da área de montagem de programas. Este botão envia um “programa em branco” para a placa Arduino.

Enquanto o programa estiver sendo enviado, o botão de envio, ou parada, ficará cinza. É possível que o programa leve, em algumas situações, cerca de um minuto ou mais para ser carregado. Essa demora, em algumas máquinas, é característica do compilador Arduino, responsável por enviar o programa do ambiente de programação em blocos Arduino para a placa.

Existe uma área na parte inferior da janela do programa, chamada: Mensagens IDE Arduino, onde podem ser vistas as mensagens enviadas pela IDE.

2.3.4 Possíveis Problemas

1. Em algumas situações, é necessário que a configuração da Placa Arduino e da porta COM seja feita antes no Software Arduino para que as configurações dessas informações, dentro do DB4K, tenham efeito.
2. Caso, por alguma razão, o DB4K não consiga enviar o programa para a placa, pode-se utilizar a opção de menu <Código Arduino> <Abrir código na IDE Arduino> para abrir o código construído no Software Arduino e então enviá-lo para a Placa Arduino.
3. Caso ocorra um erro de versão de biblioteca quando do uso do sensor de distância, deve-se, no Windows, apagar a biblioteca Ultrasonic presente nesta pasta “documentos\Arduino\libraries”, caso contrário, o Arduino utilizará esta biblioteca, não compatível, por enquanto, com o DB4K, ao invés de utilizar a presente na pasta C:\Program Files (x86)\Arduino\libraries.