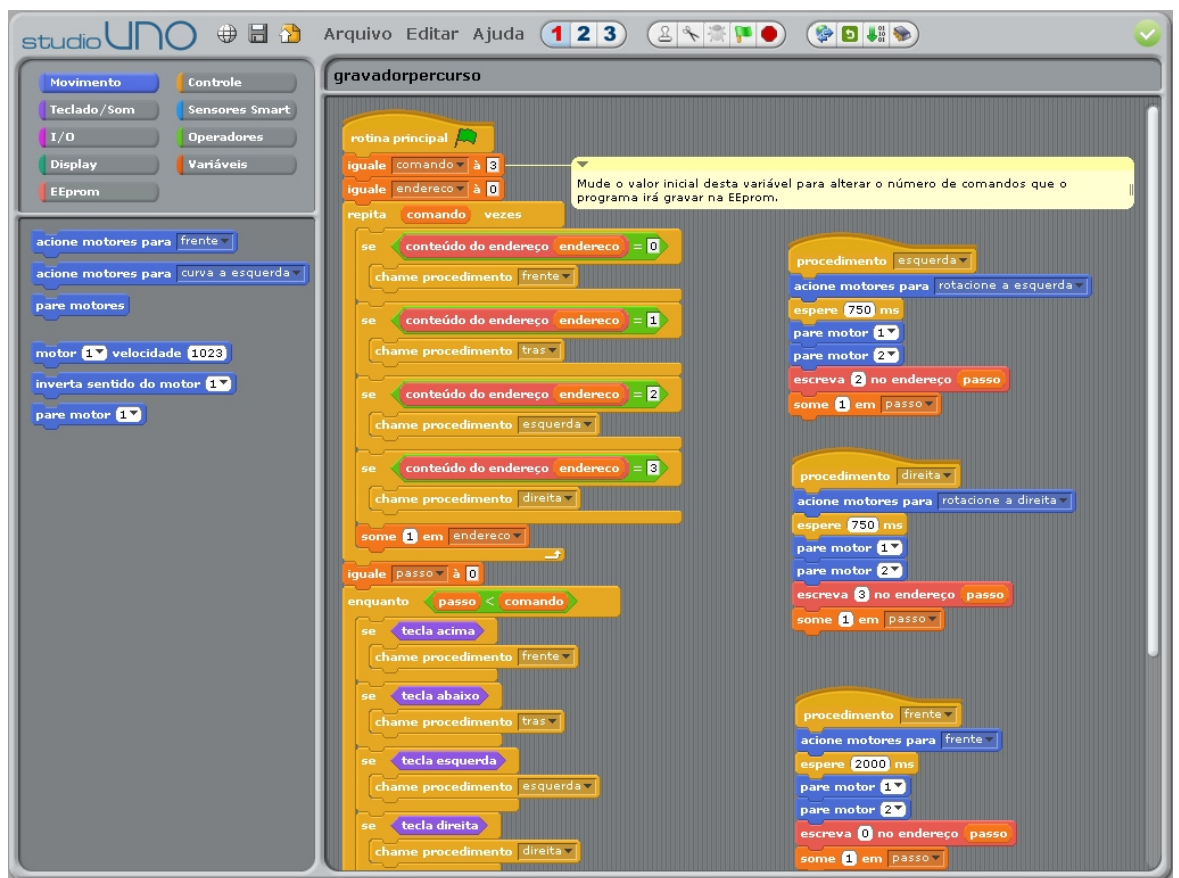


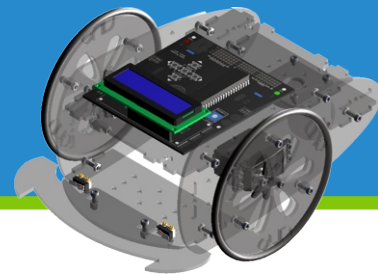
UNO

Manual Studio Uno 2.0

Guia de Referência



REVISÃO 1011.24



Manual Studio Uno 2.0

1. Introdução

O Studio Uno é um ambiente gratuito, que torna fácil a programação do seu Robô Uno. Criado a partir da linguagem de programação Scratch, desenvolvida pelo Lifelong Kindergarten Group no MIT Media Lab, se baseia no modelo de programação por blocos, onde o usuário "monta" seu programa ao arrastar os blocos de comando para a área de programação.

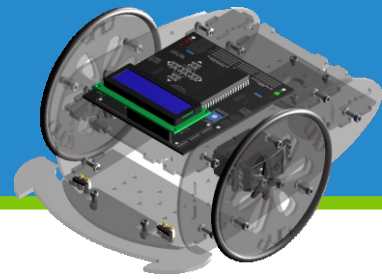
Este **Guia de Referência** apresenta em detalhes, todas as funções e recursos do Studio UNO. Se você quer iniciar a utilização do Studio UNO de uma maneira mais geral, recomendamos a leitura do documento **Primeiros Passos**, disponível na seção Studio UNO do site www.robouno.com.br. E então, se você precisar informações mais detalhadas, leia este Guia de Referência.

O site do robô UNO possui diversos outros recursos para ajudar você a aprender a programar seu robô usando o Studio UNO: tutoriais, vídeos, exemplos de programas, FAQ's e um fórum. Para mais detalhes, Por favor acesse o link <http://www.robouno/studiouno>.

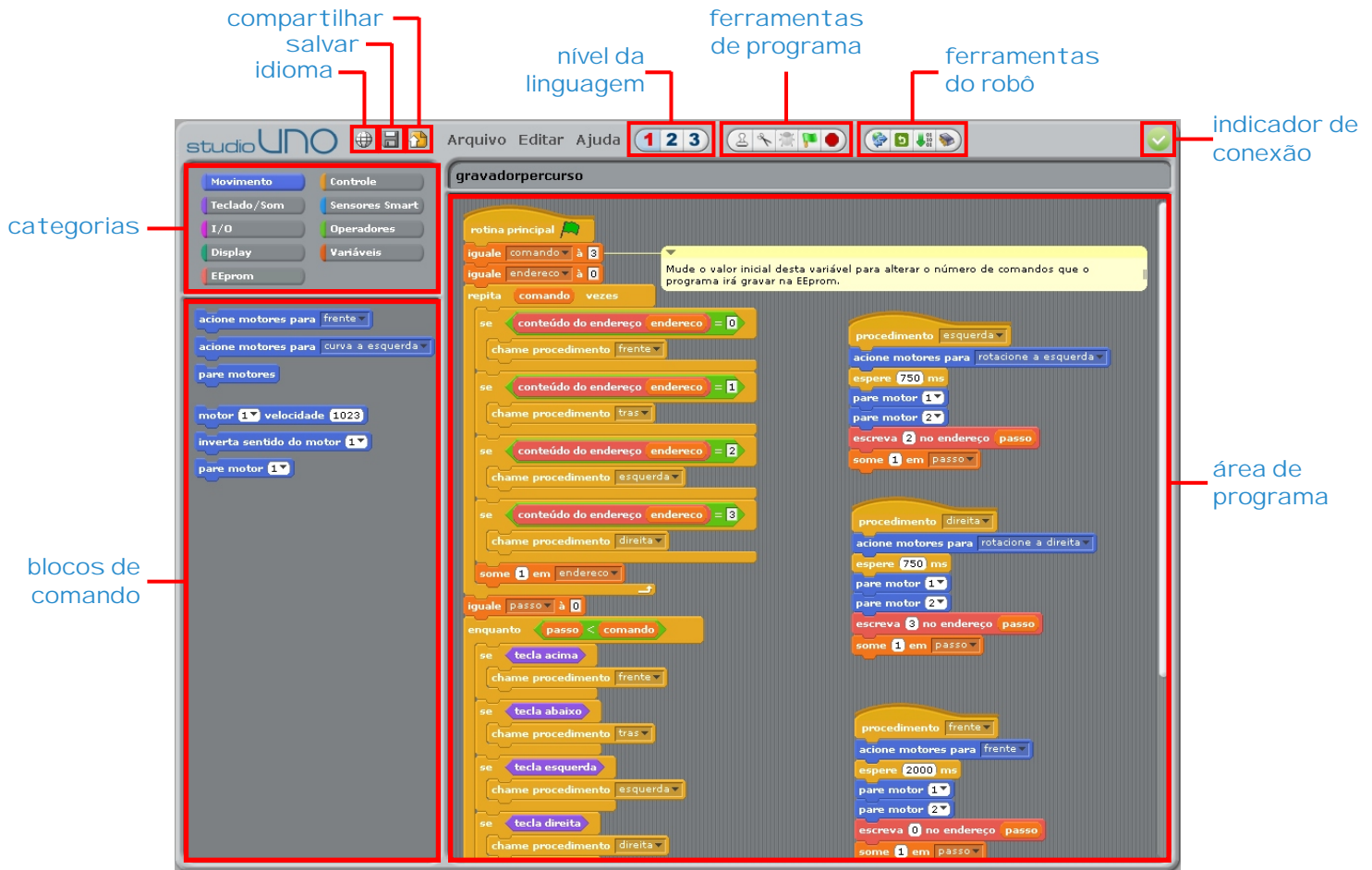
Este guia se baseia na versão 2.0 do Studio UNO. Para versões mais atualizadas deste Guia de Referência acesse o link acima.

2. Instalação

Para obter e instalar a versão mais atualizada do Studio Uno, acesse o site do robô UNO e siga os passos do **Manual de Instalação Studio UNO 2.0**.

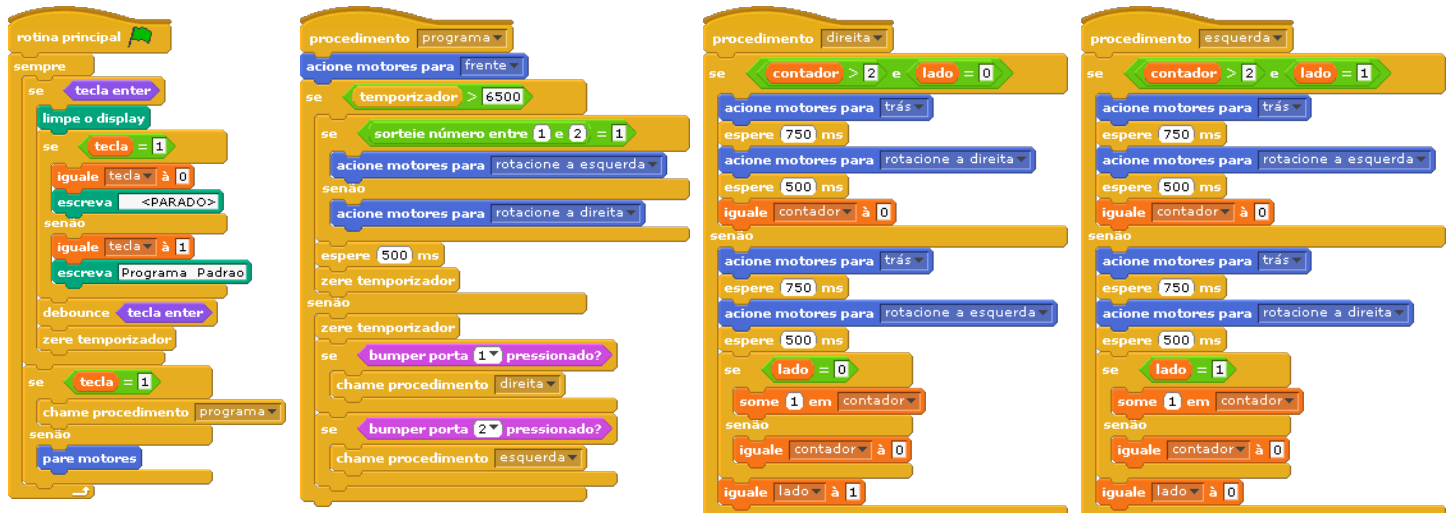


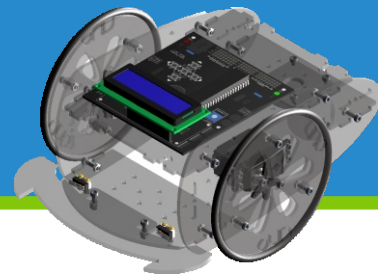
3. Interface Studio UNO



Área de Programa

A **Área de Programa** é o local onde você cria seu programa arrastando **Blocos de Comando** de diversas **Categorias**, encaixando-os uns nos outros, formando *pilhas* para criar a seqüência de execução do seu programa.





Quando você move um bloco pela **Área de Programa**, uma barra branca indica onde você pode largar o bloco de comando para formar uma conexão válida com outro bloco.

Esta conexão indica a seqüência em que os blocos serão executados pelo robô UNO.

Os blocos podem ser inseridos e conectados no início, no meio ou no fim da pilha, em qualquer ordem, mas sua execução se dará pela ordem sequencial final.

Para mover uma pilha inteira, arraste-a clicando no primeiro bloco da pilha. Se você clicar em um bloco no meio da pilha e o arrastar, todos os bloco conectados a partir dele serão movidos juntos.

Alguns blocos possuem um campo de texto editável, como por exemplo o bloco **espere 1000 ms**. Para mudar o seu valor, clique no campo e digite o valor desejado. Você também pode arrastar blocos arredondados, como o bloco **bateria**, para dentro do campo de texto.

Também existem blocos que possuem um menu de opções, como por exemplo o bloco **pare motor 1**. Clique em para ver o menu e escolher uma opção.

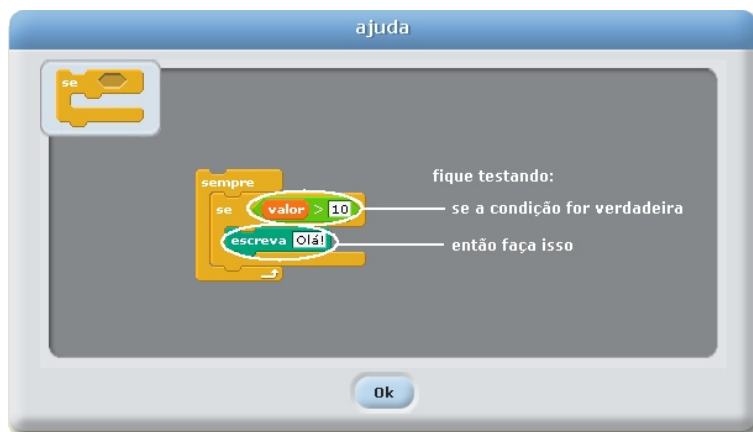
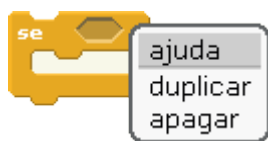
Para organizar automaticamente a **Área de Programa**, clique com o botão direito do mouse em um espaço vazio da **Área de Programa** e selecione a opção **organize área de programa** no menu de contexto. Para exportar uma imagem da **Área de Programa**, selecione a opção **salvar uma imagem da área de programa** no menu de contexto.

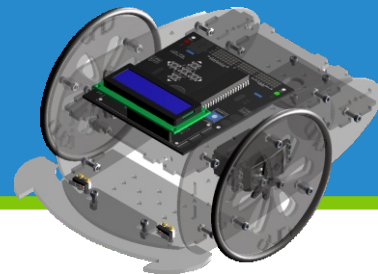
Para adicionar comentários sobre o programa na **Área de Programa**, selecione a opção **adiciona comentário** do menu de contexto. Uma área amarela de comentário surgirá, e você pode digitar seus comentários nela.

Use a barra na lateral direita para redimensionar a largura da área de comentário. Clique no triângulo no topo para ocultar ou mostrar o comentário. Um comentário pode ser adicionado em qualquer lugar da **Área de Programa**, e você pode movê-los e reposicioná-los simplesmente os arrastando.

Para ligar um comentário a um bloco específico de uma pilha, arraste e solte o comentário em cima do bloco. Para desconectar, arraste o comentário para longe do bloco.

Para obter ajuda sobre o que o bloco de comando faz, clique no bloco com o botão direito do mouse, e selecione a opção **Ajuda** no menu que irá surgir.





Nível da Linguagem

Para tornar mais fácil o aprendizado da utilização do Studio UNO, os blocos de comando foram agrupados por níveis:



3 - Profissional: mostra todos os blocos de comando do Studio UNO, e uma nova categoria: Registradores. Os Sensores Smart podem ser configurados para responderem a eventos.

2 - Avançado: neste nível, além dos blocos do nível 1, estão à disposição blocos de comando para controle de servo motores, recepção de comandos infra vermelhos, blocos de leitura dos Sensores Smart e operadores de bits. Também é disponibilizada a ferramenta de Depuração

1 - Básico: ideal para iniciantes, aparecem apenas blocos de comando básicos. Todos os blocos de comando necessários para programar o robô UNO na sua configuração básica estão neste nível.

Ferramentas de Programa

A barra de Ferramentas de Programa oferece acesso rápido às opções de duplicar, apagar blocos e depurar o programa:



Parar: encerra a a execução simulada do programa, fechando a ferramenta de depuração.

Executar: simula a execução do programa, destacando cada bloco.

Depurar: inicia a execução passo-a-passo do programa. É possível verificar o fluxo de execução do programa e inspecionar o valor das variáveis, em um ambiente de simulação.

Apagar: clique nesta ferramenta e mova o mouse sobre um bloco. Ao clicar, o bloco ou os blocos marcados com uma borda vermelha serão apagados.

Duplicar: clique nesta ferramenta e depois no bloco que deseja duplicar. Todos os blocos conectados a partir do bloco selecionado serão duplicados também. Para retornar o cursor do mouse ao normal, clique em qualquer lugar vazio da **Área de Programa**.

Ferramentas do Robô

Já a barra de Ferramentas do Robô oferece acesso rápido às opções tela on-line, reset, compilar e programar o robô:



Compilar e Gravar: compila o programa e o grava na memória do controlador do robô UNO para ser executado de forma independente do computador.

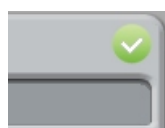
Compilar: transforma o programa formado por blocos em uma linguagem que será entendida e interpretada pelo controlador do robô UNO.

Reinicializa: envia para o robô UNO conectado na USB um comando para ele reinicializar.

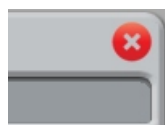
Tela On-Line: abre uma janela onde é possível visualizar o estado e as configurações de todas as teclas e portas e outras informações do controlador.

Indicador de Conexão

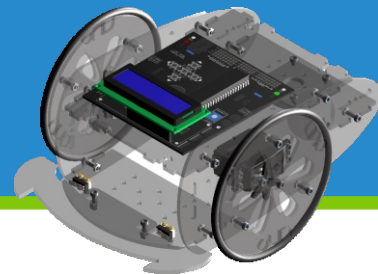
Este indicador mostra se o Robô UNO está corretamente conectado ao computador e comunicando com o Studio UNO.



Se o indicador estiver na cor verde, significa que o robô UNO está comunicando com o Studio UNO.



Se o indicador estiver na cor vermelha, significa que o robô UNO não está comunicando com o Studio UNO. Conecte o robô na porta USB ou consulte www.robouno.com.br/forum.



Menu Principal



Para mudar o idioma da interface do **Studio UNO**, clique no botão **Selecionar Idioma**: . Selecione então o idioma desejado no menu. Os idiomas atualmente suportados são o português e o inglês. Para incluir outros idiomas acesse a área *Documentos* do endereço www.robouno.com.br/studiouno.

Clique no botão para salvar seu programa.

Para compartilhar seu programa no site do robô UNO clique no botão . Você tem a opção de tornar público seu programa ou não. Assim você tem acesso aos seus programas em qualquer lugar e ainda pode compartilhar seu conhecimento com outros usuários do robô UNO.

No menu **Arquivo**, você pode criar um novo programa, abrir um programa salvo anteriormente e salvar o programa atual na pasta Projetos UNO ou em outros locais.

Importar Programa possibilita que seja um programa ou trecho de programa seja reutilizado, sem a necessidade de reescrever o código.

Exportar Código C cria um arquivo, na mesma pasta do seu projeto com a extensão `.c`, com a tradução do seu programa para a linguagem de programação C e o mostra no seu navegador.

Notas do Programa permite que você escreva e guarde anotações sobre o seu programa, tais como instruções sobre como usá-lo.

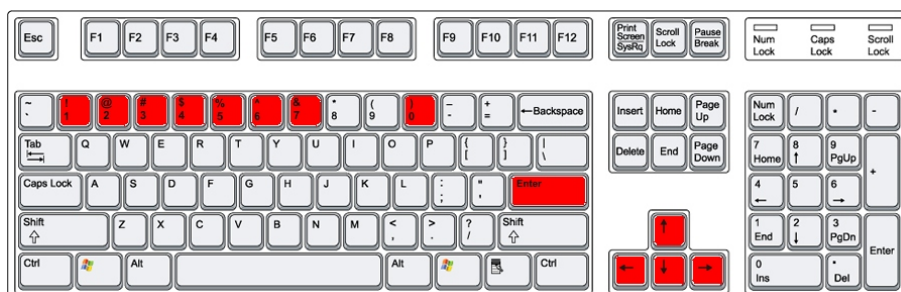
Sair fecha o Studio UNO.

O menu **Editar** oferece vários recursos para edição do seu programa.

Desfazer permite recuperar o último bloco ou pilha de blocos que você tenha apagado.

Executar programa faz com que o Studio UNO simule a execução do seu programa, passo a passo. Cada bloco é destacado ao ser executado. Esta opção pode ser útil para encontrar erros no programa e para ajudar novos programadores a entenderem o fluxo de execução do programa. Use a opção *Parar programa* para terminar a execução passo a passo.

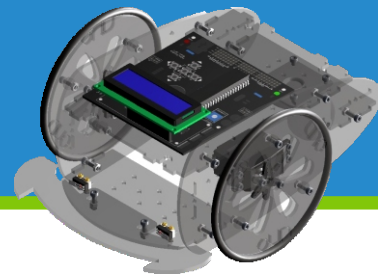
Para simular o estado das portas digitais do seu robô use as teclas 0 a 7 do teclado do seu computador. As teclas direcionais e a tecla enter simulam o teclado do robô UNO. Consulte a seção Ferramenta de Depuração para aprender mais sobre os recursos de depuração do Studio UNO.



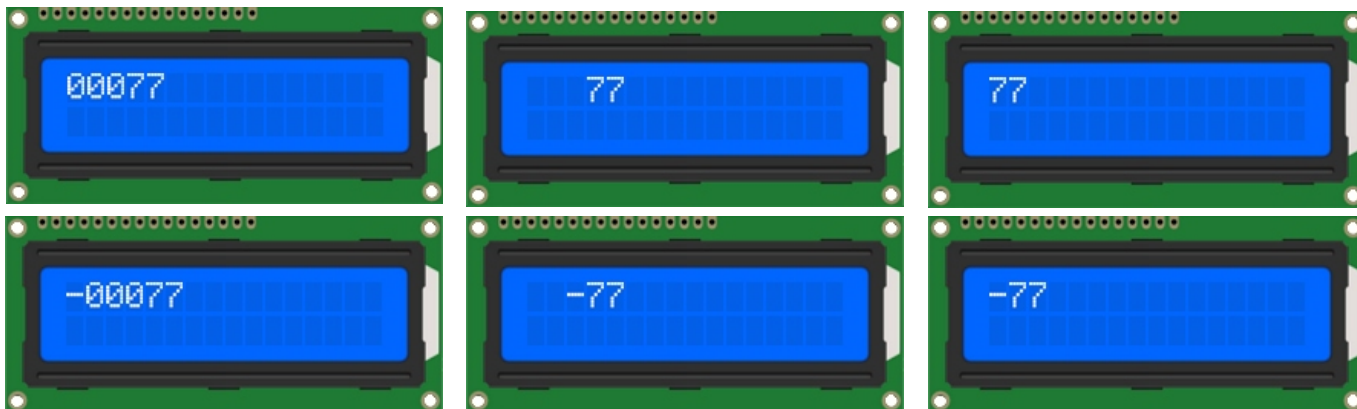
Mostrar Controlador UNO permite observar virtualmente o controlador, enquanto o programa é executado no depurador.

Velocidade do debug permite alterar velocidade com que os blocos são executados durante a depuração: normal, rápido ou devagar. A opção normal não mostra nenhum destaque no bloco em execução. As opções rápido e devagar mostram o bloco em execução destacado por 30ms e 200ms, respectivamente.

Nome do Robô define o nome que aparecerá na tela de inicialização do seu robô UNO. Você pode alterar quantas vezes quiser o nome do seu robô. O nome pode ter até 10 caracteres incluindo letras, números e espaços. Evite usar acentos e ç pois os displays não os mostram corretamente.



Modo do display permite configurar como o bloco **escreva** da categoria Display irá alinhar os caracteres no display: preenchido com zeros, com espaços ou sem preenchimento, de acordo com o tipo da variável ou da saída:



Preenchido com zeros

Preenchido com espaços

Sem preenchimento

Esta configuração é global, ou seja, todas as saídas do bloco **escreva** no programa usarão a mesma configuração.

Nível da linguagem seleciona quais blocos e ferramentas devem ser mostrados, dependendo do nível de conhecimento do usuário: Básico, Avançado ou Profissional.

Tela On-Line abre uma janela onde é possível visualizar o estado e as configurações de todas as teclas e portas e outras informações do controlador.

Reinicializa o Robô UNO é usado para reinicializar o robô e voltar ao menu inicial no robô. Esta opção equivale a retirar e recolocar a bateria no robô.

Compilar transforma o programa formado por blocos em uma linguagem que será entendida e interpretada pelo controlador do robô UNO.

Compilar e Gravar o programa compila e grava na memória do controlador do robô UNO o programa corrente, para ser executado de forma independente do computador.

Usando o menu **Ajuda** você pode acessar a página de ajuda com links para manuais de referência, tutoriais e outros. Também pode acessar a página de ajuda com todas as tela de ajuda do Studio UNO.

4. Blocos de Comando

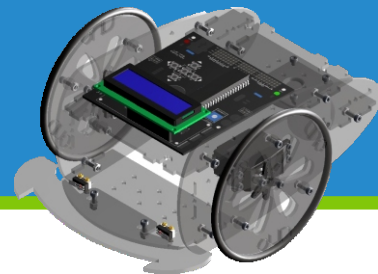
Tipos de Blocos

Existem três tipos principais de **Blocos de Comando**:

Blocos de Pilha: Estes blocos de comando possuem encaixes na parte superior e/ou na parte inferior, como por exemplo o bloco **pare motores**. Você pode encaixar estes blocos entre si, formando pilhas. Alguns Blocos de Pilha possuem uma área para entrada de dados, onde você pode digitar um número (como o número 1000 no bloco **espere 1000 ms**), escolher uma opção no menu do bloco (como a opção "trás" do bloco **selecione motores para trás**) ou encaixar um Bloco de Valor. Alguns Blocos de Pilha possuem a forma de um "C", como o bloco **enquanto**, que permitem que pilhas de blocos sejam encaixadas no seu interior.

Blocos de Início: Estes blocos possuem seu topo arredondado, como o bloco **procedimento**. Estes blocos são colocados no topo de pilhas de blocos de comandos, indicando seu início. Eles indicam um evento que o programa espera que aconteça, como uma interrupção ou chamada, e então, os blocos da pilha são executados.

Blocos de Valor: São blocos como o **temporizador** e o **tecla enter**, que podem ser inseridos na área de entrada de outros blocos. Blocos de Valor com cantos arredondados retornam números ou letras, e podem ser encaixados dentro de blocos com áreas de entrada de formato arredondado ou quadrado, como os blocos **arredonde** ou **escreva UNO**. Blocos de Valor que possuem cantos angulares (**entrada digital 0**) retornam valores *booleanos* (verdadeiro ou falso) e podem ser encaixados dentro de blocos com áreas de entrada angulares ou quadradas, como os blocos **debounce** ou **escreva UNO**.



Variáveis

São espaços de memória no controlador reservados para guardarem valores durante a execução do programa. Cada variável corresponde a uma posição de memória, cujo conteúdo pode ser alterado pelo programa.

Para criarmos uma variável no Studio UNO, selecionamos a categoria **Variáveis** e clicamos no botão **Criar uma variável**.



Na janela que surge, escolha o tipo da variável. Ao criarmos uma variável, o compilador reserva um espaço na memória do controlador. Por isso, precisamos determinar o tipo de informação que a variável trabalhará de modo a se ter espaço suficiente para a alocação de qualquer dado do tipo declarado. Escolha o tipo que melhor se aplique a informação que sua variável irá armazenar.

Informe também o nome da sua variável. O nome deve iniciar com uma letra ou o caracter "_" (sublinha) e não pode conter espaços, acentos ou "ç".

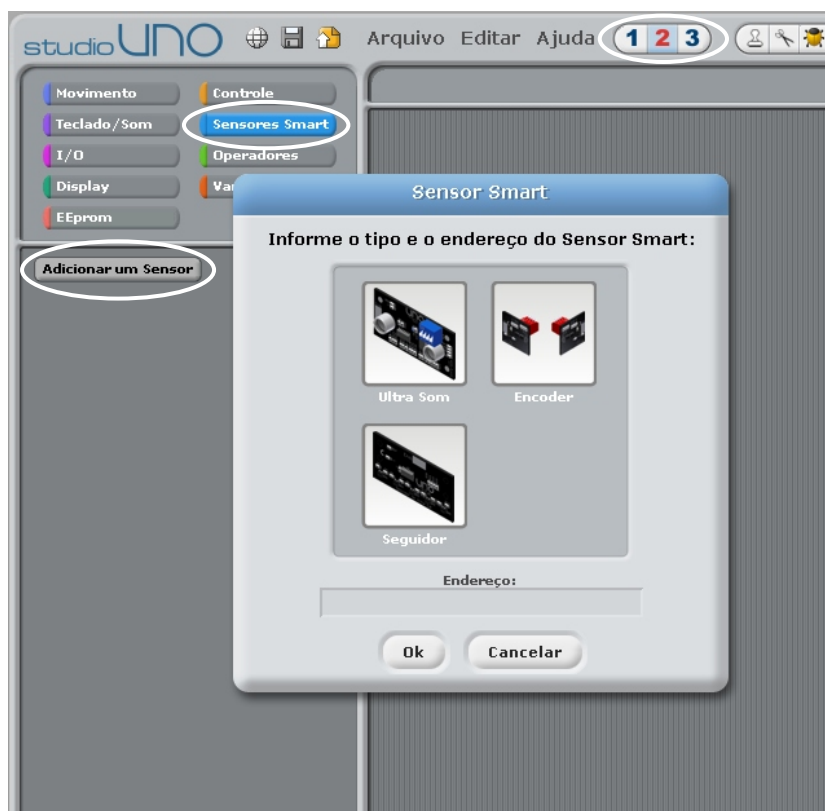
Quando se cria uma variável, diversos blocos aparecem na categoria **Variáveis**. As funções destes blocos são explicadas na seção Descrição dos Blocos.

Se for necessário mudar o tipo da variável após sua criação, selecione a categoria **Variáveis** e clique sobre o bloco da variável logo abaixo com o botão direito do mouse. No menu, selecione a opção **mudar tipo** e escolha o novo tipo da variável.

Sensores Smart

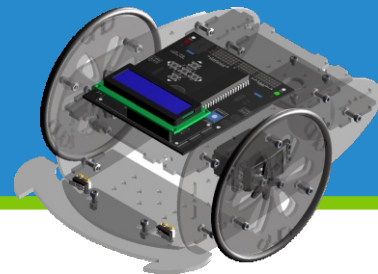
Sensores Smart são sensores que possuem processamento próprio e podem ser configurados para detectarem determinados eventos de maneira independente, liberando o controlador de processar o reconhecimento destes eventos.

Para adicionarmos um Sensor Smart no Studio UNO, selecionamos a categoria **Sensores Smart** e clicamos no botão **Adicionar um Sensor**. Este botão só aparece nos níveis de linguagem 2 e 3.



Na janela que surge, escolha o tipo do Sensor Smart e o seu endereço. Este endereço é numérico e pode variar de 0 a 255. O endereço deve corresponder ao endereço configurado no Sensor Smart. Para obter maiores informações sobre como configurar o endereço do Sensor Smart, consulte o manual específico do sensor.

Quando adicionamos um Sensor Smart, diversos blocos específicos do tipo do sensor aparecem na categoria **Sensores Smart**. As funções destes blocos são explicadas na seção Descrição dos Blocos.



5. Ferramenta de Depuração

Encontrar e eliminar possíveis erros em um programa chama-se depurar. Estes erros também são conhecidos como *bugs* e o processo de encontrá-los chama-se *debug*.

A ferramenta de depuração do Studio UNO é um recurso que pode ser usado para verificar o funcionamento do programa. Enquanto o programa está sendo simulado, podemos usar o depurador para inspecionar o conteúdo das variáveis, simular as entradas e saídas, acompanhar passo a passo a execução de cada bloco de comando, parar o processamento em um determinado bloco e visualizar o estado do controlador no simulador.

O Studio UNO permite simular e depurar o seu programa de duas maneiras: execução simulada ou passo a passo.

Execução Simulada


Para iniciar a execução simulada do seu programa clique no botão  da barra de ferramentas de programa. Cada bloco ao ser executado é destacado e é possível simular os blocos de entrada com o teclado. Consulte a ajuda de cada bloco para saber como simular seu valor de retorno. Também é possível visualizar o conteúdo das variáveis clicando na caixa de seleção próxima ao bloco da variável.

Diagram illustrating the simulated execution of a program. The script includes a 'rotina principal' block, an 'iguale contador à 0' block, a 'sempre' loop containing 'led 2', 'espere 1000 ms', and 'se tecla acima' with an 'incremente contador' block. The 'tecla acima' variable is set to 'falso' and the 'contador' variable is set to '47'. An arrow points to the 'led 2' block with the label 'Bloco em execução'.

Execução Passo a Passo



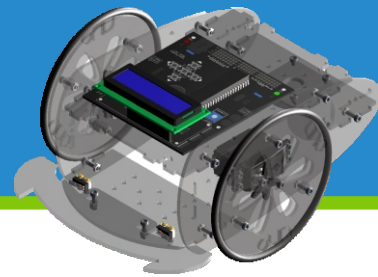
Para iniciar a execução passo a passo do seu programa clique no botão  da barra de ferramentas de programa. Este modo só estará disponível nos níveis de linguagem 2 e 3. Cada bloco ao ser executado é destacado e a execução do programa é parada. Use os botões da janela de depuração para controlar o processo:

Diagram illustrating the step-by-step execution of a program. The script is shown in the 'Depurador' window. The 'iguale contador à 0' block is highlighted. The debugger window includes the following controls:

- Bloco destacado:** Points to the highlighted 'iguale contador à 0' block.
- Parar programa:** termina a depuração, parando a execução simulada do programa. Fecha a janela do depurador.
- Executar programa:** continua a depuração no modo execução simulada, fechando a janela do depurador.
- Inspeccionar bloco:** permite verificar o valor que o bloco irá retornar.
- Passar bloco:** executa os blocos internos ao bloco em depuração se a condição retornar verdadeiro, sem parar em cada um deles. A execução é parada no primeiro bloco abaixo na pilha.
- Próximo bloco:** executa o bloco destacado e pára no próximo bloco da pilha.

Em programas muito grandes, a execução passo a passo pode se tornar muito cansativa. Para evitar processos de depuração muito demorados, pode-se usar o bloco  da categoria **Controle**.

Ao simularmos a execução do programa, e o bloco **ponto de parada** for executado, o programa será parado e a janela de depuração aparecerá, permitindo continuar com a depuração passo a passo.



6. Descrição dos Blocos

Os blocos de comando do Studio UNO são organizados em dez categorias codificadas por cores: **Movimento**, **Controle**, **Teclado/Som**, **Sensores Smart**, **I/O**, **Operadores**, **Display**, **Variáveis**, **Eeprom** e **Registradores**.

Movimento

acione motores para frente

Liga os dois motores do robô UNO fazendo com que o robô se mova para frente, para trás ou inverta o sentido do movimento atual.

acione motores para curva a esquerda

Aciona os motores de forma que o robô faça o movimento selecionado.

pare motores

Para os dois motores do robô.

motor 1 velocidade 1023

Aciona o motor selecionado no sentido e velocidade especificada.
Motor: 1 ou 2 (esquerdo/direito) Velocidade/Sentido: -1023 (trás) a 1023 (frente) (motor parado: 0)

inverta sentido do motor 1

Muda a direção do movimento do motor sem alterar sua velocidade.
Motor: 1 ou 2 (esquerdo/direito)

pare motor 1

Para o motor especificado.
Motor: 1 ou 2 (esquerdo/direito)

Os blocos abaixo só são mostrados nos níveis da linguagem 2 e 3 (Avançado e Profissional):

habilita servo na porta 0

Configura a porta digital indicada como saída PWM para controle de servo motor.
Porta: 0 a 6 (porta 7 não pode ser usada para controle de servos)

desabilita servo na porta 0

Desabilita o controle de servo motores na porta indicada.
Porta: 0 a 6

servo porta 0 angulo 90°

Posiciona o eixo do servo motor da porta digital indicada na posição informada.
Porta: 0 a 6 Ângulo: 1 a 180°

Controle

rotina principal

Indica o início do programa. Sua utilização é obrigatória. Só pode haver um bloco deste em cada programa.

sempre

Repete a execução dos blocos internos permanentemente.

enquanto

Verifica se a condição é verdadeira. Se for, executa os blocos internos e verifica a condição novamente. Quando a condição retornar falso, segue a execução dos blocos posicionados abaixo do bloco.

repita 10 vezes

Repete a execução dos blocos internos o número de vezes especificado

se

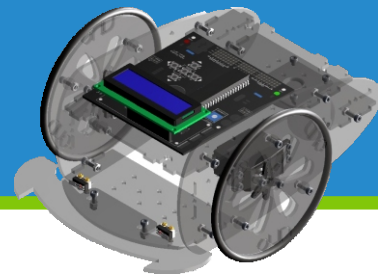
Se a condição retornar verdadeiro, executa os blocos internos.

se
senão

Se a condição retornar verdadeiro, executa os blocos interno à parte **se** do bloco. Se retornar falso, executa os blocos da parte **senão**.

procedimento

Dá um nome a uma pilha específica de blocos, a fim de organizar o programa e seu fluxo, além de permitir a reutilização das pilhas sem duplicá-las.



chame procedimento	Executa a pilha do bloco Procedimento especificado no menu. Quando a execução do procedimento terminar, o bloco seguinte a este é executado.
temporizador	Retorna o valor do temporizador. O valor deste bloco é incrementado a cada 1 milissegundo pelo controlador. O tipo da variável deste bloco é <i>unsigned long</i> .
zere temporizador	Zera o valor do temporizador reiniciando sua contagem.
espere 1000 ms	Pausa a execução do programa pelo número de milissegundos especificados e então executa o próximo bloco.
espere até	Pausa a execução do programa até que a condição retorne verdadeiro e então executa o próximo bloco.
pare tudo	Termina a execução do programa.
reinicializa	Reinicializa o robô. Este bloco equivale a retirar e recolocar a bateria do robô.

Os blocos abaixo só são mostrados nos níveis da linguagem 2 e 3 (Avançado e Profissional):

ponto de parada	Interrompe a execução simulada e abre a janela do Depurador que permite a execução passo a passo do programa.
------------------------	---

O bloco abaixo só é mostrado no nível da linguagem 3 (Profissional):

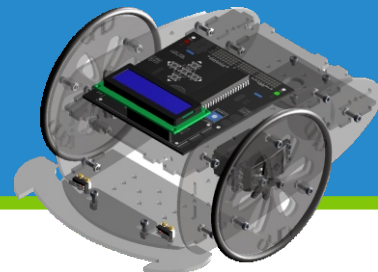
comando C	Permite incluir comandos da linguagem C no programa. A utilização deste bloco sem o conhecimento necessário, pode danificar permanentemente o controlador UNO.
------------------	--

Teclado/Som

tecla acima	Retorna verdadeiro se a tecla acima do teclado do robô estiver pressionada.
tecla abaixo	Retorna verdadeiro se a tecla abaixo do teclado do robô estiver pressionada.
tecla esquerda	Retorna verdadeiro se a tecla esquerda do teclado do robô estiver pressionada.
tecla direita	Retorna verdadeiro se a tecla direita do teclado do robô estiver pressionada.
tecla enter	Retorna verdadeiro se a tecla enter do teclado do robô estiver pressionada.
debounce tecla direita	Elimina o efeito de repique gerado pelo pressionamento das teclas do robô, aguardando a liberação da tecla indicada.
toque a nota 60 por 500 ms	Emite a nota selecionada pelo número de milissegundos indicado.
beep	Emite o sinal sonoro padrão.

Os blocos abaixo só são mostrados nos níveis da linguagem 2 e 3 (Avançado e Profissional):

Recepção Infra Vermelho ligada	Habilita a recepção dos sinais do controle remoto infra vermelho conectado na porta Digital 4
IR bit 0 à IR bit 6	Retorna verdadeiro se a tecla correspondente do controle remoto infra vermelho estiver pressionada.
Código IR	Retorna o código IR de um sinal enviado por um controle remoto. <i>unsigned char</i>
Endereço IR	Retorna o endereço IR correspondente ao sinal recebido de um controle remoto. <i>unsigned char</i>



Sensores Smart

Os blocos desta categoria variam de acordo com o sensor adicionado e não são mostrados no nível da linguagem 1. Com o nível 2 selecionado são mostrados apenas os blocos de acesso aos valores de retorno específicos de cada tipo de sensor. Já no nível da linguagem 3, além dos blocos de acesso específicos, são mostrados os seguintes blocos:

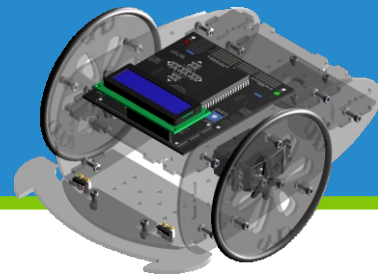
	Este bloco indica o início da pilha que será executada se ocorrer o evento programado no sensor indicado no parâmetro do bloco.
	Retorna um valor referente a qual evento o sensor indicado no parâmetro está programado para detectar. Para a lista dos valores retornados, consulte o manual específico do sensor.
	Verifica se algum evento esperado foi detectado por algum sensor e executa a pilha de tratamento deste evento. É automaticamente colocado dentro de comandos de laço no código C a ser compilado.

I/O

	Retorna o valor lido na porta digital selecionada. Porta: 0 a 7 Retorna: verdadeiro ou falso
	Retorna o valor lido na porta analógica selecionada. Porta: 0 a 5 Retorna: 0 a 1023
	Retorna o valor analógico (0 a 1023) da tensão da bateria. Para transformar em volts use a fórmula: valor * 23,5 / 1023 (aproximado)
	Liga, desliga ou muda o estado da saída digital selecionada. Saída: 0 a 7
	Retorna verdadeiro se o bumper conectado à porta selecionada estiver acionado. Porta: 0 a 7
	Liga, desliga ou muda o estado do led conectado à porta Digital 0.

Operadores

	Soma dois valores.
	Subtrai dois valores.
	Multiplica dois valores.
	Divide dois valores.
	Retorna um número aleatório entre os dois valores.
	Retorna verdadeiro se o primeiro valor for menor que o segundo valor.
	Retorna verdadeiro se o primeiro valor for igual ao segundo valor.
	Retorna verdadeiro se o primeiro valor for maior que o segundo valor.
	Retorna verdadeiro se o primeiro valor for diferente do segundo valor.
	Retorna verdadeiro se o primeiro valor for menor ou igual do que o segundo valor.
	Retorna verdadeiro se o primeiro valor for maior ou igual do que o segundo valor.
	Retorna verdadeiro se ambas condições forem verdadeiras.



ou	Retorna verdadeiro se uma das condições for verdadeira.
não	Retorna verdadeiro se a condição for falsa, e falso se a condição for verdadeira.
resto da divisão de <input type="text" value=""/> por <input type="text" value=""/>	Retorna o resto da divisão do primeiro número pelo segundo número
elevado a <input type="text" value=""/>	Retorna o resultado do primeiro número elevado ao segundo número.
arredonde <input type="text" value=""/>	Retorna o número informado arredondado para o inteiro mais próximo.
raiz quadrada de <input type="text" value="10"/>	Retorna o resultado da função escolhida aplicada ao número informado.

Os blocos abaixo só são mostrados nos níveis da linguagem 2 e 3 (Avançado e Profissional):

bit <input type="text" value="0"/> de <input type="text" value=""/>	Retorna verdadeiro se o bit indicado do bloco de valor for 1. Caso o bit seja 0, o bloco retorna falso
seta bit <input type="text" value="0"/> de <input type="text" value=""/>	Muda o bit indicado do bloco de valor para 1.
limpa bit <input type="text" value="0"/> de <input type="text" value=""/>	Muda o bit indicado do bloco de valor para 0.
muda bit <input type="text" value="0"/> de <input type="text" value=""/>	Muda o bit indicado do bloco de valor para 0 case seja 1 ou para 1 caso seja 0.

Display

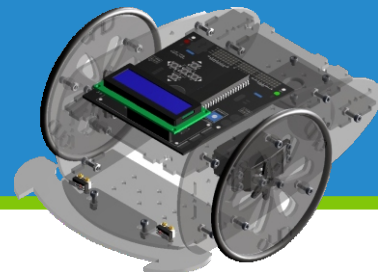
escreva <input type="text" value="UNO"/>	Escreve no display do robô o texto informado ou conteúdo de uma variável.
limpe o display	Apaga o conteúdo do display e posiciona o ponto de inserção na primeira linha e primeira coluna.
vá para linha <input type="text" value="1"/> coluna <input type="text" value="1"/>	Posiciona o ponto de inserção na linha e coluna indicadas.
luz de fundo <input type="text" value="ligada"/>	Liga e desliga a luz de fundo do display (<i>backlight</i>).

Variáveis

Criar uma variável	Clique no botão para criar uma nova variável. Informe o tipo e o nome da variável. Quando a primeira variável é criada, os blocos da categoria Variável são mostrados.
Apagar uma variável	Apaga uma variável.
contador	Retorna o valor que a variável contém.
iguale <input type="text" value="contador"/> à <input type="text" value="0"/>	Coloca o valor informado na variável selecionada.
some <input type="text" value="1"/> em <input type="text" value="contador"/>	Adiciona o valor informado ao valor atual da variável selecionada.
subtraia <input type="text" value="1"/> de <input type="text" value="contador"/>	Diminui o valor informado do valor atual da variável selecionada.
incremente <input type="text" value="contador"/>	Adiciona 1 ao valor atual da variável selecionada.
decremente <input type="text" value="contador"/>	Diminui 1 do valor atual da variável selecionada.

EEProm

conteúdo do endereço <input type="text" value="0"/>	Retorna o conteúdo gravado no endereço informado da memória EEPROM.
escreva <input type="text" value="0"/> no endereço <input type="text" value="0"/>	Grava no endereço solicitado da memória EEPROM o valor informado.



Registadores

Os blocos desta categoria só são mostrados no nível da linguagem 3 (Profissional)

Estes blocos de valor dão acesso a todas as configurações de hardware do microcontrolador do controlador UNO. A utilização destes blocos sem o conhecimento necessário, pode causar erros na execução dos programas e, até mesmo, danificar permanentemente o controlador UNO.

Para obter a descrição de cada bloco desta categoria acesse a ajuda do bloco no Studio UNO ou consulte o datasheet do microcontrolador.

ADCON0	HLVDCON	POSTDEC1	TOCON	UADDR
ADCON1		POSTDEC2	T1CON	UCFG
ADCON2	INDF0	POSTINC0	T2CON	UCON
ADRES	INDF1	POSTINC1	T3CON	UEIE
ADRESH	INDF2	POSTINC2	TABLAT	UEIR
ADRESL	INTCON	PR2	TBLPTR	UEP0
	INTCON2	PREINC0	TBLPTRH	UEP1
BAUDCON	INTCON3	PREINC1	TBLTPRL	UEP2
BSR	IPR1	PREINC2	TBLTPRU	UEP3
	IPR2	PROD	TMR0H	UEP4
CCP1CON	LATA	PRODH	TMR0L	UEP5
CCP2CON	LATB	PRODL	TMR1H	UEP6
CCPR1	LATC		TMR1L	UEP7
CCPR1H	LATD	RCON	TMR2	UEP8
CCPR1L	LATE	RREG	TMR3H	UEP9
CCPR2		RCSTA	TMR3L	UEP10
CCPR2H	DSCCON	SPBRG	TOS	UEP11
CCPR2L	DSCUNE	SPBRGH	TOSH	UEP12
CMCON		SPPCFG	TOSL	UEP13
CVRCON	PCL	SPPCON	TOSU	UEP14
	PCLATH	SPPDATA	TRISA	UEP15
ECCP1AS	PCLATU	SPPEPS	TRISB	UFRM
ECCP1DEL	PIE1	SPPADD	TRISC	UFRMH
EEADR	PIE2	SPPBUF	TRISD	UFRML
EECON1	PIR1	SPPCON1	TRISE	UIE
EECON2	PIR2	SPPCON2	TXREG	UIR
EEDATA	PLUSW0	SPPSTAT	TXSTA	USTAT
	PLUSW1	STATUS		WDTCN
FSR0H	PLUSW2	STKPTR		WREG
FSR0L	PORTA			
FSR1H	PORTB			
FSR1L	PORTC			
FSR2H	PORTD			
FSR2L	PORTE			
	POSTDEC0			